





КӘСІПТІК БІЛІМ

Деректер қоры

1305000 – Ақпараттық жүйелер
1304000 – Есептеу техникасы және бағдарламамен
қамтамасыз ету

Оқулық

Қазақстан Республикасы Білім және ғылым министрлігі
техникалық және кәсіптік білім беру ұйымдарына ұсынады

FOLIANT
БАСПАСЫ

Нұр-Сұлтан
2021

«Турани-Астана» университеті

КІТАПХАНА

Исп.№

65620

УДК 004 (075)
ББК 32.973.202я73
Д 28

Пікір жазғандар:

Шаңғытбаева Г. А. – PhD докторы, Қ. Жұбанов атындағы Ақтөбе өңірлік мемлекеттік университеті

Нурина Ж.К. – арнайы пәндер оқытушысы, Ақтөбе жоғары политехникалық колледжі

Марат Г.М. – өндірістік оқыту шебері, Ақтөбе жоғары политехникалық колледжі

Құрастырғандар:

Эльмар Делер, Дирк Харди Лаупхайм, Оберхаузен
Хуберт Троссман, Буртенбах

Д 28 Деректер қоры: *Оқулық* / Неміс тілінен ауд. – Нұр-Сұлтан: Фолиант, 2021. – 232 бет.

ISBN 978-601-338-375-0

Кітапта деректер қорын пайдалану, жүйелік архитектуралар, деректер қорының модельдері, деректер қорын әзірлеу және қалыпқа келтіру, DB-Desinger, Microsoft VISIO, Access ортасында деректер қорын әзірлеу, MySQL клиенттерімен жұмыс жасау, редакциялау, Java көмегімен деректер қорына кіру, т.б. тақырыптар қарастырылып, студенттердің өзіндік жұмысын ұйымдастыруға мүмкіндік беретін тәжірибелік тапсырмалар берілген.

Оқулық техникалық және кәсіптік білім беру мекемелерінің 1305000 «Ақпараттық жүйелер», 1304000 «Есептеу техникасы және бағдарламамен қамтамасыз ету» мамандықтары бойынша білім алушы студенттеріне арналған.

УДК 004 (075)
ББК 32.973.202я73

ISBN ISBN 978-601-338-375-0

© 2018 VERLAG EUROPA-LEHRMITTEL,
Nourney, Vollmer GmbH & Co. KG
42781 Haan-Gruiten (Germany)
© «Фолиант» баспасы, аударма, 2021

АЛҒЫ СӨЗ

Информатика және ақпараттық технологиялар өміріміздің барлық әлеуметтік салаларына әсер етеді. Барлық дерлік кәсіби іс-әрекеттер мен процестер айтарлықтай дәрежеде ақпараттық технологиялардың арқасында жүзеге асырылады. Деректер қорының жүйелері маңызды компонент болып табылады, өйткені компанияның қабілеттілігі сақталатын деректердің қолжетімділігіне, толықтығына және дәлдігіне байланысты.

«Деректер қоры» кітабы қазіргі заманғы бағдарламалық жүйелерді пайдалана отырып, деректер қорын жоспарлау, іске асыру және бағдарламалаудағы теориялық және практикалық негіздерді ұсынады. Бұл ретте өзара байланысты түсіндіруге үлкен мән беріледі.

Пәндік салаға деректер қоры технологиялары бойынша кіріспе бола отырып, бұл кітап кәсіптік колледждердің, кәсіптік академиялардың, гимназиялардың, арнайы жоғары оқу орындарының студенттеріне арналған.

Жекелеген тарауларда көптеген мысалдармен қатар тақырыптық салаларды бекіту және тереңдетуге арналған сараланған практикалық жаттығулар да бар.

2018 ж., күз

Авторлар мен баспагерлер

МАЗМҰНЫ

АЛҒЫ СӨЗ	3
1 Деректер қоры жайлы негізгі түсінік	8
1.1 ДЕРЕКТЕР ҚОРЫН ПАЙДАЛАНУ	8
1.1.1 Деректер қорын пайдалану мысалдары	8
1.1.2 Деректер қорында деректерді сақтау мәселелері	10
1.1.3 ДҚБЖ міндеттері	12
1.2 ЖҮЙЕЛІК АРХИТЕКТУРАЛАР	14
1.2.1 Үстел ДК деректер қоры (бір пайдаланушыға қолжетімді)	14
1.2.2 Бірнеше пайдаланушыға арналған үстел ДК деректер қоры (бірнеше пайдаланушыға қолжетімділік)	14
1.2.3 Клиент-серверлердің деректер қоры	15
1.3 ДЕРЕКТЕР ҚОРЫНЫҢ МОДЕЛЬДЕРІ	15
1.3.1 Реляциялық деректер қоры	15
1.3.2 Объектілі-бағытталған деректер қоры	16
1.3.3 Иерархиялық және желілік деректер қоры	16
1.4 ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕСІНІҢ АРХИТЕКТУРАСЫ (ДҚБЖ)	18
1.4.1 Үш деңгейлі архитектура (үш қабатты архитектура)	18
1.5 ДЕРЕКТЕР ҚОРЫН ЖОБАЛАУ КЕЗЕҢДЕРІ	20
2 Деректер қорының реляциялық жүйелері	22
2.1 ДЕРЕКТЕР ҚОРЫНЫҢ РЕЛЯЦИЯЛЫҚ ЖҮЙЕЛЕРІ	22
2.1.1 Кестелер мен байланыстар	22
2.1.2 Кілттер мен байланыстар	23
2.2 «МӘН – БАЙЛАНЫС» ТИПТЕГІ ҮЛГІ	26
2.3 «МӘН-БАЙЛАНЫС» ТИПТЕГІ ҮЛГІСІНЕ АРНАЛҒАН ШЕШІМІ БАР МЫСАЛДАР:	31
2.3.1 Тапсырысты өңдеу	31
2.3.2 Жеткізушілер мен тауарлар	31
2.4 2-БӨЛІМГЕ АРНАЛҒАН ЖАТТЫҒУЛАР	32
3 Деректер қорын өзірлеу және қалыпқа келтіру	38
3.1 ДЕРЕКТЕР ҚОРЫН ӨЗІРЛЕУ	38
3.1.1 Бағдарламалық жасақтаманы өзірлеу рәсімдері	39
3.2 ҚАЛЫПҚА КЕЛТІРУ	40
3.2.1 Қалыпты формалар	40

3.2.2 Қалыпқа келтіру үшін мысалдар: қашықтықтан жасалатын сауда	44
3.2.3 Басқа да қалыпты формалар	48
3.2.4 Біртұтастылық шарттары	48
3.3 3-БӨЛІМГЕ АРНАЛҒАН ЖАТТЫҒУЛАР	49
4 Деректер қорын модельдеу үшін бағдарламалық қамтамасыз ету	54
4.1 DB-DESIGNER.....	54
4.1.1 Жүктеу және орнату.....	54
4.1.2 Кестелерді құру	57
4.1.3 Кестелерді реляциялық байланыстыру	59
4.1.4 Деректер қорының жолдарын жазу	60
4.1.5 ER диаграммасын құру.....	60
4.1.6 Forward Engineering (Тікелей жобалау).....	64
4.2 MICROSOFT VISIO	67
4.2.1 Деректер қоры моделінің диаграммасын іске қосу	67
4.2.2 Кестелер жасау	68
4.2.3 Бағандар қосу	69
4.2.4 Байланыстар қосу	70
4.2.5 Реверсивті инжиниринг.....	71
4.2.6 Индекстерді құру	74
4.2.7 Көріністер құру (Views)	76
4.2.8 Өрістерді тексеру шарттарын жасау	78
5 Access ортасында деректер қорын әзірлеу	79
5.1 КЕСТЕЛЕР ҚҰРУ	79
5.2 БАЙЛАНЫС ЖӘНЕ СІЛТЕМЕЛІК ТҰТАСТЫҚТЫ АНЫҚТАУ	82
5.3 НЫСАНДАР	84
5.3.1 Нысан жасау.....	84
5.3.2 Бағынышты нысандар	86
5.3.3 Батырмалар арқылы деректер қорын басқару	88
5.4 МАКРОСТАР.....	90
5.5 ЕСЕП ҚҰРУ	91
5.6 ДЕРЕКТЕР ҚОРЫНА СҰРАНЫС ЖАСАУ	94
5.7 5-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР	96
6 SQL деректер қорының тілі	98
6.1 SQL СТАНДАРТТАРЫ.....	98
6.2 КЕСТЕЛЕРДІ ҚҰРУ, ӨЗГЕРТУ ЖӘНЕ ЖОЮ.....	99
6.3 SELECT АРҚЫЛЫ ТАҢДАУ СҰРАНЫСТАРЫ	100

6.3.1	Шарттары бар таңдауға сұраныстарды шектеу.....	101
6.3.2	DISTINCT	102
6.3.3	WHERE шартындағы өріс мазмұнын көрсету.....	102
6.3.4	BETWEEN операторы.....	103
6.3.5	IN операторы.....	104
6.3.6	Нөлдік мәндермен жұмыс	105
6.3.7	Деректерді сұрыптау.....	105
6.3.8	Сұраныс нәтижелерін шектеу	106
6.3.9	SELECT-сұраныстарындағы функциялар	107
6.3.10	Деректерді топтастыру	112
6.3.11	Бірнеше кесте бойынша сұраныстар.....	113
6.3.12	ҚОСЫМША СҰРАНЫСТАР.....	119
6.4	SQL АРҚЫЛЫ ДЕРЕКТЕРДІ ӨҢДЕУ	120
6.4.1	Деректер қорының жолдарын кірістіру.....	120
6.4.2	Деректер қорының жолдарын жою	122
6.4.3	Деректерді жаңарту	122
6.5	ДЕРЕКТЕР ҚОРЫНЫҢ КЕЛІСІЛУІ.....	123
6.6	ТРАНЗАКЦИЯЛАР	124
6.7	6-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР	126
7	LibreOffice Base	129
7.1	ДЕРЕКТЕР ҚОРЫН ҚҰРУ	129
7.2	КЕСТЕЛЕР АРАСЫНДА БАЙЛАНЫС ЖАСАУ	140
7.3	ДЕРЕКТЕР ҚОРЫНЫҢ ЖОЛДАРЫН ЖАЗУ	142
7.4	БАСҚА ДЕРЕКТЕР ҚОРЫНА ҚОСЫЛУ	143
7.5	СҰРАНЫСТАР ЖАСАУ.....	146
7.6	ҚАЛЫПТАР	150
8	Интернет желісіндегі деректер қоры	155
8.1	ХАМРР ӨЗІРЛЕУ ОРТАСЫ.....	155
8.2	КОМПОНЕНТТЕРДІҢ ЖҰМЫС ІСТЕУІ	155
8.2.1	Веб-сервер	155
8.2.2	ХАМРР өзірлеу ортасын орнату	156
8.2.3	Компоненттерді іске қосу.....	157
8.3	PHP СЦЕНАРИЙ ТІЛІ	157
8.3.1	Кіріспе	157
8.3.2	PHP сценарийін жазу.....	158
8.3.3	PHP-дағы айналымдар.....	159
8.3.4	Массивтер	159
8.3.5	Массивтермен жұмыс.....	164

8.3.6 Символдық жолдарды өңдеу	165
8.3.7 PHP арқылы файлдармен операциялар орындау	167
8.3.8 Файлдарға кіру құқығы.....	169
8.3.9 Нысандармен жұмыс	171
8.4 MYSQL ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕСІ	172
8.4.1 MySQL клиенттерімен MySQL жұмыс істейді	173
8.4.2 Қолжетімділік құқықтарын беру және жою	176
8.4.3 PHP көмегімен MySQL деректер қорын өңдеу	179
8.5 ODBC ИНТЕРФЕЙСТЕРІ АРҚЫЛЫ ДЕРЕКТЕР АЛМАСУ	181
9 JAVA арқылы деректер қорына кіру.....	186
9.1 JAVA АРҚЫЛЫ ДЕРЕКТЕР ҚОРЫНА КІРУ	186
9.1.1 JDBC көмегімен деректер қорын байланыстыру.....	186
9.1.2 JDBC драйверлерін жүктеу және байланыс орнату.....	187
9.1.3 SQLite деректер қорына кіру.....	187
9.1.4 SELECT-тен ерекшеленетін командаларды жою	191
9.1.5 Метадеректерді алу	193
9.2 БАСҚА ДЕРЕКТЕР ҚОРЫНА ЖҮГІНУ	195
9.2.1 Драйверді қосу	195
9.2.2 Басқа деректер қорының драйверлері	196
9.3 9-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР	196
10 NET және C# көмегімен деректер қорына кіру#	199
10.1 NET және C көмегімен деректер қорына кіру #	199
10.1.1 NET Framework астында деректер қорын қосу.....	199
10.1.2 Провайдерлерді пайдалану және байланыс орнату	200
10.1.3 ACCESS деректер қорына кіру мысалы.....	201
10.1.4 SELECT-ке ұқсамайтын командаларды жою	204
10.1.5 DataAdapter және деректер жиынтығы	206
10.2 VISUAL C ДЕРЕКТЕР ҚОРЫНЫҢ ШЕБЕРІН ПАЙДАЛАНУ	210
10.2.1 Деректер қорын қосу.....	210
10.2.2 Windows Forms басқару элементтерін автоматты түрде қосу	214
10.2.3 WPF басқару элементтерін автоматты түрде қосу	216
10.3 10-ТАРАУҒА ТАПСЫРМАЛАР	221

1 Деректер қоры жайлы негізгі түсінік

1.1 ДЕРЕКТЕР ҚОРЫН ПАЙДАЛАНУ

Деректер қоры – бұл логикалық байланыстағы және **деректер қорын басқару жүйесін (ДҚБЖ)** басқаратын сақталған деректер жиынтығы. Бұл деректерді, мысалы, қолданбалы бағдарламалар мен кәсіпорын пайдаланушылары қолданады.

Ескерту

Деректер қоры логикалық байланысты деректер жиынтығы болып табылады.

1.1.1 Деректер қорын пайдалану мысалдары

Деректер қоры кәсіпорынның компьютерлік технологияларды қолдануында жиі орталық рөл атқарады. Яғни жұмыс процестері компьютердің көмегімен еңделетін, үлкен көлемді деректерді сақтау қажет ететін барлық жерлерде пайдаланылады.

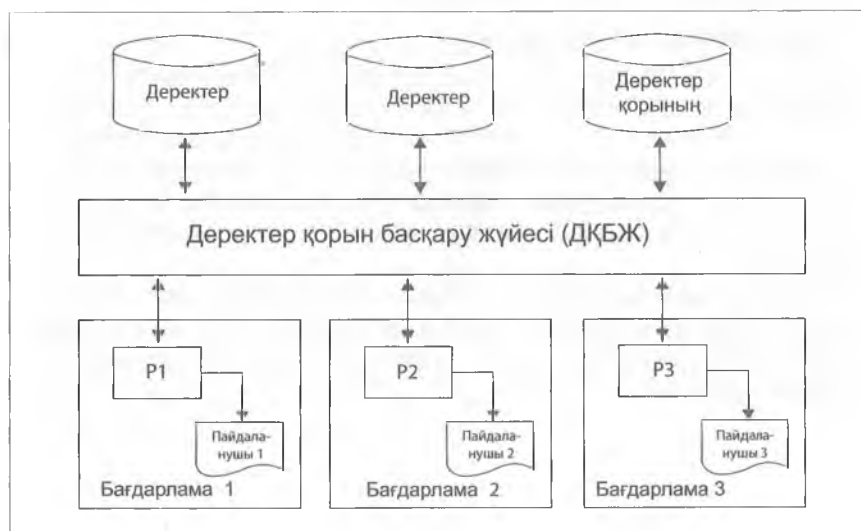
Мысалдар

- Банктер мен сақтандыру компаниялары деректер қорын басқару жүйелерімен жұмыс істейді. Деректер қорында шоттар, жазбалар және клиенттер туралы барлық ақпарат құрылымдалған. Деректердің құпиялылығы мен қауіпсіздігі аса басымдыққа ие.
- Ресурстарды жоспарлау жүйесі (ERP, Enterprise Resource Planning) бар кез келген келемдегі және салалардағы компанияларда, мысалы, клиенттер, қызметкерлер немесе тауарлар туралы деректер қоры басқару жүйесінің көмегімен сақталады және еңделеді.
- Қойманы автоматты басқару да деректер қорын пайдалануды талап етеді. Қойманың деректер қоры көптеген жеткізушілер, тауарлар және олардың қорлары туралы реттелген ақпаратты қамтиды.
- Интернеттегі ақпараттық жүйелер (мысалы, Wikipedia) деректер қорын пайдалана отырып, өз мақалаларын басқарып отырады.

- Компаниялар деректердің қоймаларында (Data Warehouses) деректерді талдау және бизнестің түрлі салаларында шешімдер қабылдауға көмектесетін түрлі ақпаратты сақтайды. Мысалы, нарықтарды зерттеу институты оларды одан әрі өңдеу үшін өздерінің деректерін және үшінші тұлғалардың деректерін сақтайды.



Қолданбалы бағдарламалар, мысалы, қойманы немесе персоналды басқаруға арналған бағдарламалар ДҚБЖ арқылы жалпы деректерге қатар жүгіне алады.



1.1.2 Деректер қорында деректерді сақтау мәселелері

Деректер қорының көмегімен деректерді сақтау кезінде көптеген проблемалар туындауы мүмкін:

- **Деректер артықтығы**

Деректер бірнеше рет сақталады (қайталанады), бұл деректерге өзгерістер енгізуді қымбат етеді, ал қателер ықтималдығы артады. Бір ғана дерек әр жерде бірнеше рет өзгертілуі тиіс. Мысалы, бірнеше рет сақталған деректерді өзгерту тек бір жерде ғана жүзеге асырылса, деректер ауқымы қате болады.

- **Сәйкессіздіктер**

Егер деректерді бірнеше пайдаланушы немесе бағдарламалар бір уақытта өзгертсе, онда бұл деректердің келісілмеген жағдайына әкелуі мүмкін. Деректерге кіру қадамдастырылмаған, мысалы, егер ағымдағы шотты бір уақытта екі пайдаланушы өңдесе, онда екеуі де бірдей сальдоны көреді: 2000 еуро. Ал егер пайдаланушы А шотынан 1000 еуро алса және осы операцияны сақтаса, ал пайдаланушы В бір мезгілде 500 еуро енгізсе және осы операцияны сақтаса, онда екі мәндер де деректер қорындағы: 1000 және 2500 еуро қарама-қайшы және дұрыс емес болады.

- **Деректерді қорғау**

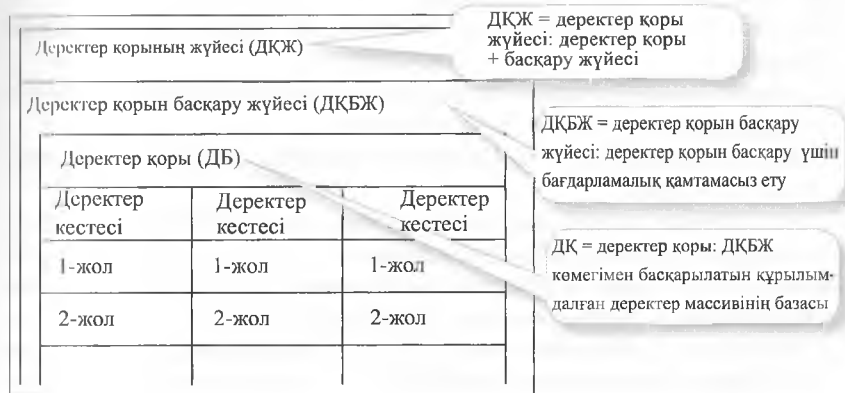
Ақпаратты оқуға және жазуға барлық деректер көлемінде қол жеткізуге болады. Деректерді қорғау – қолданылатын операциялық жүйеге байланысты: қол жеткізу құқықтарын теңшеу немесе шифрлау арқылы іске асырылуы мүмкін.

- **Деректердің тәуелсіз болмауы**

Деректерді басқару әдетте тиісті қолданбалы бағдарламалық қамтамасыз ету арқылы ғана мүмкін. Егер деректер құрылымын өзгерту қажет болса, қолданбалы бағдарламалық қамтамасыз етуде де, файлдарды қайта құрылымдау бағдарламасында да өзгеріс енгізу қажет. Сондай-ақ басқа қолданбаның бір файлдарын бағалау үшін осы жаңа бағдарлама үшін жеке деректерді басқаруды жасау керек.

Пайдаланушы деректерді оңай және анық басқара алуы үшін оған деректер қорын басқару жүйесі – ДҚБЖ қажет. Осылайша, **деректер қорының жүйесі (ДҚЖ)** деректер қорынан (ДБ) және басқару жүйесінен тұрады.

Ең көп тараған ДҚБЖ: Microsoft Access, LibreOffice Base, MySQL, Paradox, Oracle және MS SQL Server.



Ескерту:

ДҚБЖ деректерді артық, қажетті қауіпсіздік жағдайымен және деректерді қорғау деңгейімен сақтайды және ұйымдастырады. ДҚБЖ деректерге қол жеткізетін қолданбалы бағдарламаларға (қосымшаларға) байланысты емес.

Қолданбалы бағдарламалар деректерге тікелей қол жеткізе алмайды, тек сұраныстарды деректер қорын басқару жүйесіне жібереді. Деректер қоры белгілі бір пәндік салада логикалық байланысқан деректер жиынтығы болып табылады, мысалы, клиент мәліметтері және тапсырыс мәліметтері. ДҚБЖ деректер қоры мен оларды пайдаланушылар арасында деректер алмасуды жүзеге асырады, мысалы, қолданбалы бағдарламаларды.

Ол орталықтандырылған басқару мен бақылауды қамтамасыз ете отырып, деректерге қол жеткізуді қамтамасыз етеді. ДҚБЖ пайдаланушыларды, олардың деректер қорына кіруін және пайдаланушылардың кіру құқықтарын басқарады. Сонымен қатар ДҚБЖ көмегімен аппараттық бөлік пен бағдарламалық қамтамасыз ету қатесінен қорғау қамтамасыз етіледі, сондықтан, мысалы, жүйелік істен шығу кезінде деректер жоғалмайды немесе қалпына келтірілуі мүмкін. Деректер қорының құрылымына өзгерістер енгізу қолданбалы бағдарламаларды өзгертуді талап етпейді.

1.1.3 ДҚБЖ міндеттері

1982 жылы британдық математик Эдгар Франк Кодд ДҚБЖ талаптарының 9 тармағын әзірледі:

1. Деректер интеграциясы = барлық қажетті деректерді бірыңғай басқару.
2. Деректер операциялары = деректер ауқымы деректерді іздеуге, өзгертуге және сақтауға мүмкіндік береді.
3. Деректер каталогы = (Data Dictionary) деректер қорының сипаттамасын қамтиды.
4. Пайдаланушы көріністері = әрбір қолданба деректер ауқымының әртүрлі көріністерін (Views) талап етеді.
5. Келісу мониторингі = деректер тұтастығының мониторингі ДБ-дағы деректердің дұрыстығына кепілдік береді.
6. Рұқсат ету = деректерге қолжетімділік бақылануы мүмкін және қажет болған жағдайда пайдаланушыға қол жеткізуден бас тартылуы мүмкін.
7. Транзакциялар = ДБ-дағы өзгерістер бірлік ретінде топтастырылуы мүмкін.
8. Синхрондау = бәсекелес транзакциялар кезінде жалпы деректер синхрондалуы тиіс.
9. Деректердің сақтық көшірмесі = жүйе ақаулығы сияқты, қақтығыстан кейін деректер ауқымын қалпына келтіруге мүмкіндік береді.

Деректер қоры жүйесі мен жеке файлдарды жинақтау арасындағы айырмашылық – деректер қоры жүйесінде деректер қорларын басқару жүйесі ДҚБЖ орталықтандырылған түрде басқарылады. Қолданбалы бағдарламалар ДҚБЖ арқылы қатар жалпы деректерге қол жеткізуге болады.



ДҚБЖ архитектурасы

Деректер каталогі (Data Dictionary) деректерді сақтау ішкі деңгейде қалай жүзеге асырылатынын сипаттайды. Бұл – деректерді басқарудағы маңызды барлық ақпараттың орталық каталогі. Атап айтқанда келесі компоненттер:

- Файлдық ұйымдарды анықтау, файлдарға кіру жолдарын анықтау.
- Тұжырымдамалық деректерді анықтау, пайдаланушы түсініктерін анықтау, деректер қорының дескрипторларын оңтайландыру.
- Сұраныстар мен өзгерістерді бағалау және дискке қатынауды бақылау.

Жекелеген транзакциялар деректер ауқымына дербес сұраныс жасау болып табылады. Мысалы, ақша қаражатын аудару кезінде 100 € А шотынан алынады және В шотына енгізіледі. Транзакциялар журналы транзакцияның басталуы мен аяқталуы туралы, сондай-ақ транзакцияға дейін және одан кейін деректердің өзгертілген ауқымы туралы ақпараттан тұрады. Журнал деректері негізінде транзакциялар бақылануы немесе түзетілуі мүмкін (күші жойылған).

Транзакцияларды басқаруға арналған бағдарламалық қамтамасыз ету деректерге бір мезгілде қатынауды қамтамасыз етеді. Параллель транзакциялар деректер қорының тұтастығын қамтамасыз ету үшін синхрондалады.

Мысал

Штутгарт-Берлин әуе рейсіне 1 орын брондалған. Ульм қаласындағы А клиенті Штутгарт қаласындағы В клиентімен бір мезгілде брондау үшін бағдарламалық қамтамасыз етуге кіреді және сол бос орынды көреді. Енді А клиенті орынды брондаған кезде, брондау батырмасын басу деректер қорының жолына жазбаға тең болады. Осы сәтте деректер қоры жолы тек транзакция үшін А клиентіне қолжетімді. Егер брондау және транзакция сәтті аяқталса, В клиенті бұл орынды брондай алмайды. Клиент В орынның бос емес екенін көреді.

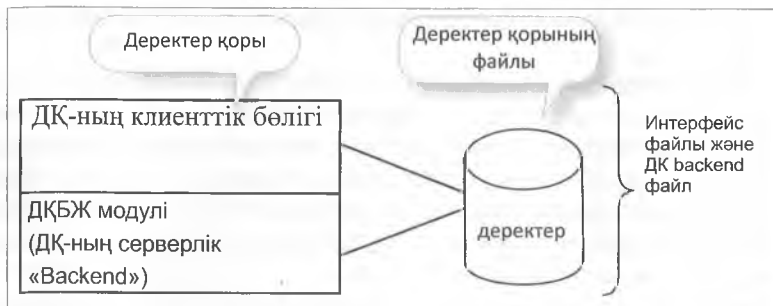
Келесідей артықшылықтар бар:

- Барлық бағдарламалар бір деректер қорымен жұмыс істейді, яғни барлық деректер бірдей өзекті.
- Барлық қолданбалар үшін деректерді бір рет сақтау.
- Орталық басқарудағы жалпы деректерге тәуелсіз бір мезгілде қолжетімділік.

1.2 ЖҮЙЕЛІК АРХИТЕКТУРАЛАР

1.2.1 Үстел ДҚ деректер қоры (бір пайдаланушыға қолжетімді)

Жұмыс үстел үсті компьютерлеріне арналған деректер қоры жағдайында, ДҚБЖ, мысалы, Access немесе Вазе пайдаланушының ДҚ-дегі нақты деректер қоры (немесе қорлары) жұмыс істейді.

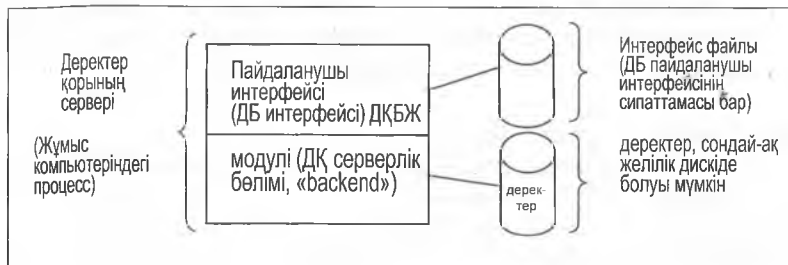


Бір пайдаланушы қол жеткізуге арналған үстелдік деректер қоры

1.2.2 Бірнеше пайдаланушыға арналған үстел ДҚ деректер қоры (бірнеше пайдаланушыға қолжетімді)

Егер деректер қорының мазмұны (backend-файл) желілік дискіде интражеліде немесе интернетте болса, онда бірнеше пайдаланушы деректер қорының мазмұнына параллель қатынаса алады.

Барлық деректер іздеу немесе сұрыптау сияқты өңдеу үшін желі арқылы берілуі тиіс.

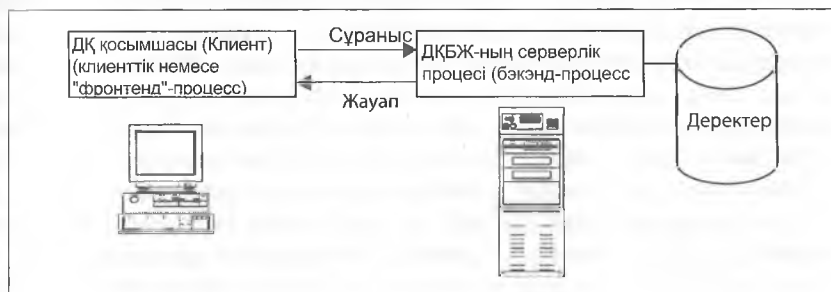


Бірнеше пайдаланушы қол жеткізуге арналған үстелдік деректер қоры

1.2.3 Клиент-серверлердің деректер қоры

Деректер қорын басқару жүйесі желіде серверлік ДК-де жұмыс істейді және деректер қорының файлдарына эксклюзивті қатынау мүмкіндігі бар. Сонымен қатар, SQL сервері («**Structured Query Language**» – сұраныстардың құрылымдық тілі) деп аталады. Клиенттік бағдарламалар деректерді алады және сақтайды.

Бұл ретте желімен тек сұраныс жасалуы (SQL, Query операторы) және жауап берілуі тиіс. Деректер қорының қосымшасы клиенттік немесе серверлік бөлікте жұмыс істей алады немесе екі бөлікке бөлінеді.



Клиент / Деректер қорының сервері

1.3 ДЕРЕКТЕР ҚОРЫНЫҢ МОДЕЛЬДЕРІ

Жалпы деректер қорының **төрт үлгісі** бар. Олар – реляциялық деректер қоры, объектілі-бағытталған, иерархиялық және желілік деректер қоры. Бұл төрт модельдің айырмашылықтары деректер қорын логикалық құру тәсілінен тұрады.

1.3.1 Реляциялық деректер қоры

Реляциялық деректер қоры тек кестелерден тұрады. Кіру осы кестелер арқылы жүзеге асырылады. Жаңа кестелерді қосу немесе жою оңай болғандықтан, деректер қорының логикалық құрылымын одан әрі өзгерту салыстырмалы түрде оңай жасалады. Кестелерге қол жеткізу оңай бағдарламалық жүзеге асырылады, бұл деректер қорының осы моделінің үлкен танымалдығына әкелді.

Әрбір жеке кесте арасындағы өзара байланыс – байланыс арқылы құрылады. Бұл байланыстар кестелерде сақталады. Кестелердің көмегімен мәліметтер массивін құру және олардың бір-бірімен байланыстары математикалық негізделген (реляциялық алгебра).

Бірақ реляциялық деректер қорлары да кемшіліктерге ие: қол жетімділік бірнеше кестеден кейін жиі жүзеге асырылады, бұл ұзақ уақыт жауап беру және енгізу/шығару санының көптігіне әкелуі мүмкін.

1.3.2 Объектілі-бағытталған деректер қоры

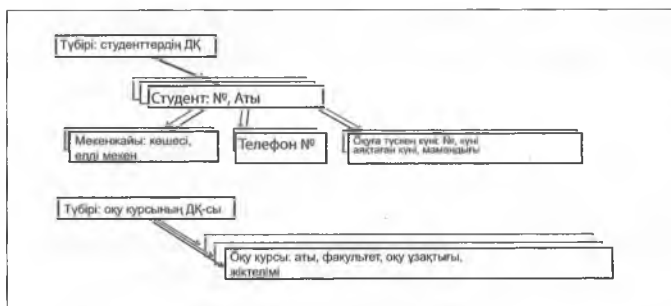
Объектілі-бағытталған деректер қоры тек объектілерден тұрады. Нысан нақты нәрсе болып табылады, мысалы, рейс, адам немесе жалпы дерексіз зат: мекенжай, шот, операция немесе компания бөлімі.

Көптеген объектілер кестелік түрде сақталатындықтан, объектілі-бағытталған ДҚ жиі реляциялық деректер қорының кеңеюі болып саналады. Алайда бұлай деп тек ішінара атауға болады. Объектілі-бағытталған деректер қоры кластар, деректер инкапсуляциясы немесе мұралық ету сияқты объектілі-бағытталған тәсілдерді қамтиды.

Объектілі-бағытталған және объектілі-реляциялық деректер қорларының реляциялық деректер қорларына қарағанда анағұрлым күрделі құрылымы бар (себебі қарапайым кестелер емес, іс жүзінде кез келген объектілер болуы мүмкін). Нәтижесінде, деректер қорының архитекторлары мен қосымшалардың бағдарламашылары жобалау мен бағдарламалауға көп күш салулары тиіс. Сонымен қатар, деректер қорын ішкі басқару кеңірек болып келеді. Сондай-ақ техникалық және мультимедиялық қосымшаларда артықшылық ретінде неғұрлым көрнекі құрылымды алуға болады (кестедегі күрделі нысандарды мәжбүрлі түрде көрсету қажет емес). Бұл сұранысты орындау уақытын айтарлықтай азайтуы мүмкін.

1.3.3 Иерархиялық және желілік деректер қоры

Ең ескі деректер қоры – иерархиялық деректер қоры, ДҚ-да қолданылатын әдеттегі файлдық ұйымның келесі даму сатысы. Бұл ДҚ логикалық құрылымы төңкерілген ағаш тәрізді құрылымға сәйкес келеді. Қажетті түйінге кіру әрқашан тамыр арқылы жүзеге асырылады. Бұл жағдайда объект тек бір тамырға байланысты болуы мүмкін; бұл моноиерархия деп аталады. Мұндай әдіс минималды артықшылықты қамтамасыз етеді, өйткені қолжетімділік тікелей ағаш тәрізді құрылым арқылы жүзеге асырылады және қолжетімділіктің ең қысқа уақытына кепілдік береді.



Иерархиялық деректер қорының үлгісі

Иерархиялық деректер қоры деректерді тіркелген байланыстар арқылы байланыстырады және де бір жазба келесіге сілтеме жасайды.

ДҚБЖ міндеттерінің бірі – нақты әлемді модельдеу – иерархиялық модельде өте шектеулі. Жоғарғы және төменгі байланыстарды көрсетуге шектеу – таза иерархиялық байланыстарды сирек бақылауға болатын күнделікті жағдайларды шынайы көрсетуге жарамсыз.

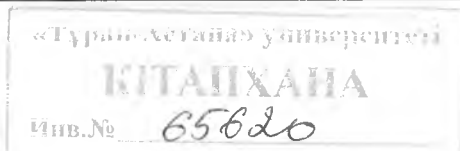
Иерархиялық деректер қорын басқару жүйесінің мысалы – IBM-нен IMS.

Желілік деректер қорының логикалық құрылымы таза иерархиялық емес, ерікті құрылған байланыс желісі арқылы бір-бірімен байланысты деректерден тұрады. Бұл икемділікті айтарлықтай арттырады, бірақ ДБ модельдеудің күрделілігін арттырады.

Бұл екі модель де ДБ-ның бүгінгі талаптарына сәйкес келмейді.

Желілік деректер қорының негізгі өкілдері IDMS (Computer Associates) және UDS (Siemens-Nixdorf) болып табылады.

Деректер қорының әртүрлі үлгілерінің артықшылықтары мен кемшіліктері		
Үлгі	Артықшылықтары	Кемшіліктері
Реляциялық деректер қоры	деректер қорының құрылымына өзгерістер енгізу жеңілдігі, математикалық негізделген, оңай бағдарламаланатын, қарапайым басқару.	жиі көптеген енгізу / шығару қажет, үлкен деректер көлемі үшін компьютердің жоғары өнімділігі талап етіледі.
Объектілі-бағытталған деректер қоры	әмбебап объектілі-бағытталған конструкция, қарапайым бағдарламалау процесі, басқару оңай.	салыстырмалы түрде көп енгізу / шығару қажет, күрделі құрылым, компьютердің салыстырмалы түрде жоғары өнімділігі қажет.



Иерархиялық және желілік деректер қоры	Төмен жауап беру уақыты, төмен артықтылық	Құрылымның өзгеруі іс жүзінде мүмкін емес, күрделі бағдарламалау.
---	---	---

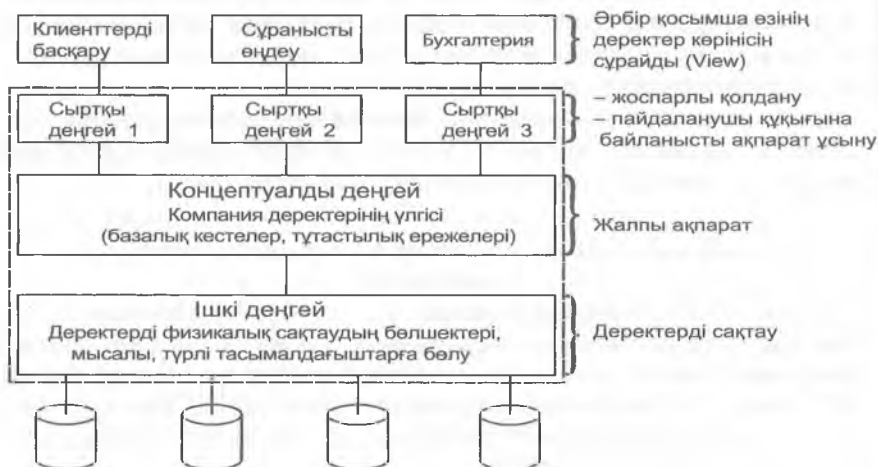
1.4 ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕСІНІҢ АРХИТЕКТУРАСЫ (ДҚБЖ)

Деректер қорында бұрын сипатталған тәсілдің маңызды ерекшеліктері:

- Бағдарламалар мен деректерді оқшаулау (бағдарламалардың, деректер мен операциялардың тәуелсіздігі)
- Қолданушылардың бірнеше көріністерін қолдау (Views)
- Деректер қорын (сызбаларды) сипаттау үшін каталогты пайдалану.

1.4.1 Үш деңгейлі архитектура (үш қабатты архитектура)

Үш деңгейлі архитектура теңшелетін қосымшалар мен сақтау бөлшектерін (физикалық қасиеттері) бір-бірінен бөледі.



Келесідей үш деңгей анықталды:

1. Ішкі деңгей

Ішкі деңгей деректер қоры жадының физикалық құрылымын сипаттайды. Ішкі схема деректердің физикалық моделін пайдалана-

ды және деректерді сақтау кезінде деректерге қол жеткізу, деректер қорына қол жеткізу жолдары және файлдарды ұйымдастыру туралы мәліметтерді сипаттайды.

2. Тұжырымдамалық деңгей

Тұжырымдамалық деңгей деректер қорының барлық пайдаланушылары үшін барлық деректер қорының құрылымын сипаттайды. Тұжырымдамалық схема сақтаудың физикалық құрылымдарының бөлшектерін жасырады және бірліктердің, деректердің типтерінің, байланыстардың, пайдаланушы операциялары мен шектеулердің сипаттамасына фокусталады. Бұл деңгейде жады құрылымынан тәуелсіз деректердің логикалық моделі қолданылады.

3. Сыртқы қабат немесе көріністер қабаты (View)

Сыртқы қабат сыртқы таңдамалы көріністерді (Views) қамтиды. Әрбір ұсыным пайдаланушылардың белгілі бір тобы мүдделі деректер қорының бөлігін сипаттайды және пайдаланушылардың осы тобынан деректер қорының қалған деректерін жасырады. Бұл деңгейде (логикалық) жад құрылымына байланысты емес деректер моделін де пайдалануға болады.

Деректер қорының барлық заманауи жүйелерінің негізінде жатқан деңгей архитектурасы қолданбалы бағдарламалар мен деректердің ішкі құрылымы арасындағы тәуелсіздікке елеулі үлес қосады.

Деректерді физикалық сақтауға қатысты өзгерістер ішкі схемадан жоғары деңгейлер үшін ДҚБЖ үшін елеулі дәрежеде керінбейтін болады. Тұжырымдамалық схемадағы өзгерістер (практикада мүмкіндігінше сирек жүргізілуі тиіс) сыртқы схемалардың көмегімен ақпаратты ұсыну арқылы бірнеше рет жүзеге асырылуы және қолданбалы бағдарламалар үшін ашық болуы мүмкін.

Ескерту:

Базалық кестелерді әдетте деректер қорының әкімшісі ғана өңдейді.

Сыртқы деңгейлер, ең алдымен, жеке қосымшалар тек олар болуы мүмкін және болуы тиіс ақпаратты алатынына кепілдік береді.

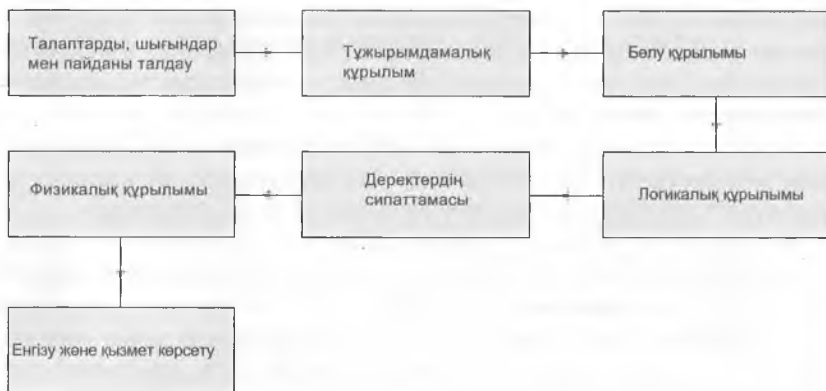
Ескерту:

Пайдаланушылар немесе қолданбалы бағдарламалар ДБ контентін сыртқы деңгей тұрғысынан ғана көреді: «сыртқы көрініс» (external view).

1.5 ДЕРЕКТЕР ҚОРЫН ЖОБАЛАУ КЕЗЕҢДЕРІ

Деректер қоры үшін қолданбалы бағдарламаны іске асыру бірнеше кезеңге бөлінуі мүмкін. Деректер қорын жобалау кезеңдері басқа бағдарламалық өнімдерді әзірлеуге де қолданылады.

1. Жаңа деректер қорына сұраныстарды жинау және талдау.
2. Қосымша функцияларына сәйкес деректер қорын жүйелі-тәуелсіз жобалау.
3. Таратылған деректер қоры үшін: жүйелік-тәуелсіз белінген жобалау жүйесі.
4. Деректер қорының моделін таңдау және деректер қорының моделінің тұжырымдамалық дизайнын көрсету.
5. Деректер анықтамалары, яғни оларды ДҚБЖ қолдану арқылы кодтау және бағдарламалау, пайдаланушы түсініктерін анықтау.
6. ДБ физикалық моделінде қол жетімділік құрылымдарын анықтау.
7. Деректер қорының қолданбасын орнату, тестілеу.



1.6 1-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР

1. Деректер қорын басқару жүйесі дегеніміз не?
2. Деректер қоры жүйесі дегеніміз не?
3. Деректер қорын пайдалану мысалдарын атаңыз.
4. Деректер қорын пайдалану кезінде қандай проблемалар туындайды?
5. АТМ (банкомат) көмегімен ақша алу мысалында деректер қорындағы сәйкессіздіктерді сипаттаңыз.
6. ДҚБЖ-ның қандай міндеттері бар?
7. ДҚБЖ-ның сыртқы деңгейінде қандай міндеттер орындалады?

8. Бір пайдалану рұқсаты бар үстел үсті ДК үшін деректер қорын сипаттаңыз.
9. Көп пайдаланушы қол жетімділігі бар үстел үсті ДК үшін деректер қорын сипаттаңыз.
10. Клиент-сервер түріндегі деректер қорын сипаттаңыз.
11. Деректер қорының әртүрлі модельдерін атаңыз.
12. Реляциялық модельді сипаттаңыз.
13. Иерархиялық деректер қоры қандай артықшылықтар береді?
14. Үш деңгейлі сәулетті сипаттаңыз.
15. Үш қабатты құрылымның әрбір деңгейінде тапсырманы көрсетіңіз.
16. Объектілі бағытталған деректер қорының қандай артықшылығы бар?
17. Деректер қоры жүйесі мен ДК-де деректерді сақтау арасындағы айырмашылықты атаңыз.
18. Деректер каталогі (Data Dictionary) көмегімен қандай есептер шешіледі?

2 Деректер қорының реляциялық жүйелері

2.1 ДЕРЕКТЕР ҚОРЫНЫҢ РЕЛЯЦИЯЛЫҚ ЖҮЙЕЛЕРІ

Реляциялық деректер қорындағы деректер, мысалы, клиенттердің деректері немесе енімнің деректері бір-бірімен байланысты болатын кестелер түрінде сақталады.

2.1.1 Кестелер мен байланыстар

Бір-бірімен байланысты **кестелер реляциялық кестелер** деп аталады. Кестенің әрбір ұяшығында **деректер қорының** бір жазбасы болады.

Ескерту:

Деректер қорының реляциялық жүйелері кестелердегі бір-бірімен байланысты деректерді басқарады.

	Аты	Тегі
K001	Walter	Hans
K002	Christ	Reiner
K003	Dehler	Elmar
K004	Schiemann	Bernd
K005	Trossmann	Hubert

Кесте бағандарында әр жазба үшін салыстырылатын мәліметтер, мысалы, клиенттің аты болады. Бағанның мәндері **деректер типі өрісінде** (деректер типі) сақталады. Деректер қорының жүйелерінде әртүрлі деректер типі болады.

Деректер қоры жүйелерінің деректер типі		
Деректер типі	Сипаттама	Мысал
Integer	Тұтас сан	5
Numeric (x.y)	Үтірге дейін X сандары бар және үтірден кейін Y сандары бар ондық сан	53,27
Decimal (x.y)	Үтірге дейін X сандары бар және үтірден кейін Y сандары бар минимум ондық сан	53,2768
Float	Қалқымалы үтірі бар сан	8,3E13
Character	Символдық жол	CH80653
Date	Күн	31.05.1966
Time	Уақыт	17:55

2.1.2 Кілттер мен байланыстар

Деректер қоры жолы кестесінің жеке жолдары. Деректер қорының жолдары мысалы, адамды, объекті немесе оқиғаны сипаттайды. Деректер қорының жолдарын, сондай-ақ **мән** (ағылш. «entity») немесе объектілер деп атайды.

Ескерту:

Мән, объект, деректер қорының жолы деректер қоры жолы кестесіндегі жолға сәйкес келеді.

Объекттер егер аттары бірдей болса, кестедегі басқа объекттермен шатасып кетуі мүмкін. Клиенттер кестесінде ол екі клиент болуы мүмкін. Ретсіздікті болдырмау үшін деректер қорының әрбір жолы үшін бірегей өрістердің көмегімен сәйкестендіргіш орнату қажет. Ол өрістер **алғашқы кілт (Primary Key)** деп аталады. Олар тиісті жазбаны бір мағыналы теңестіруге мүмкіндік жасайды. Клиенттер кестесінде ол – негізінен клиент номері.

Ескерту:

Алғашқы кілт кестедегі объекті бірегей белгілеу мен сәйкестендіру үшін пайдаланылады

Байланыс 1:м

Егер клиент тауар сатып алса, клиенттің барлық деректерін басқа «Клиенттер» деген кестеге қайтадан енгізу орынсыз. Тапсырыстар кестесінде тек клиент номері сақталады.

Графиктік көрсетілімде, мысалы, клиент деректері бар кестеден байланыс желісі тапсырыс деректері бар кестеге нұсқайды.



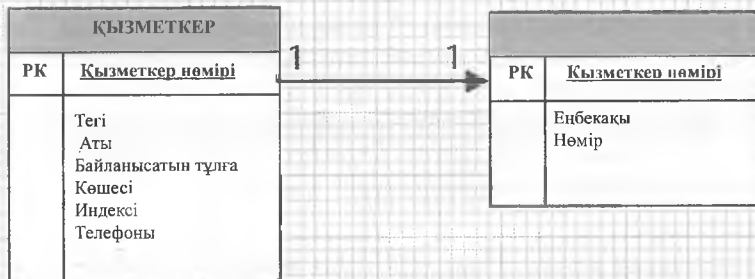
Клиенттер және Тапсырыстар кестелері бір-бірімен Клиенттер кестесінің алғашқы кілтінің көмегімен байланысқан. Деректер қорының әрбір жолының Тапсырыстар кестесінде клиент номері болады, ол Клиенттер кестесінен «1» клиентіне байланысқан болады. Және керісінше, Клиенттер кестесінен әрбір клиентке етініш жасау Тапсырыстар кестесінің «М» бірнеше жолдарынан жүргізілуі мүмкін. Сонымен, Клиенттер және Тапсырыстар кестелерінің 1:m байланысы болады. Клиент Номері алғашқы кілті бойынша Клиенттер кестесінде бір мағыналы сәйкестендірілетін деректер қорының жолдары клиенттің осы номері бойынша Тапсырыстар кестесінде анықталады. Клиент номерінің өрісі Клиенттер кестесінің тиісті өрісіне нұсқайды және, сондықтан, **сілтемелі кілт немесе сыртқы кілт** деп аталады.

Ескерту:

Сыртқы кілт – басқа кестенің алғашқы кілтінің өрісіне нұсқайтын өріс.

1:1 байланысы

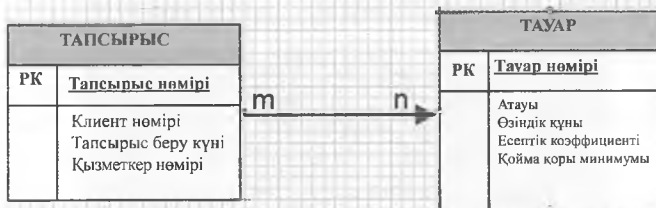
Деректердің дербес кестесінде, мысалы, еңбекақы сияқты деректер әрбір пайдаланушыға берілмеуі тиіс және еркін қолжетімділікте болмауы тиіс. Сонымен Қызметкерлер_құпия жаңа кестесі құрылады. Өйткені осы кестеде жазбаларды қызметкер Номері алғашқылық кілті бойынша табуға болады, онда бірінші кестеде деректер қорының бір жолы екінші кестедегі деректер қорының бір жолына сәйкес келеді. Мұндай байланыс 1:1 байланыс деп аталады.



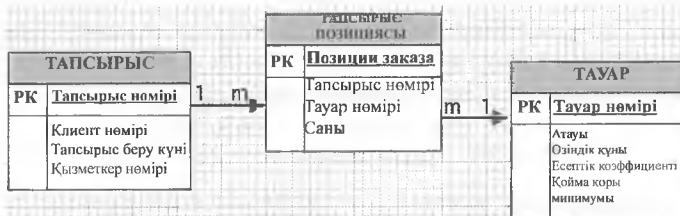
Тәжірибеде 1:1 байланыстар сирек кездеседі.

m:n байланысы

Тапсырыстар және Тауарлар кестелері жағдайында тапсырыстың «m», «n» немесе «∞» тауарлары болуы мүмкін, екінші жағынан, әрбір тауар кез келген «n», «m» немесе «∞» тапсырыстар мөлшерінде болуы мүмкін (∞ шексіздікті білдіреді).



Онда позициялар кестесінің көмегімен 1:n-екі байланысы құрылады.



Тапсырыс Позициясы жазбасында көрсетілетін Тауар Тауарлар кестесінде бір мәнді сәйкестендірілуі мүмкін. Және керісінше, кез келген тауар тапсырыстың бірнеше позицияларында көрсетілуі мүмкін. Сонымен, Тауарлар кестелері мен тапсырыс Позициялары кестелері

арасында 1:m байланысы болады. Тапсырыс позициялары мен Тапсырыстар кестелері сондай-ақ 1:m байланысқа ие болады. Тапсырыс қаншалықты қажет болатындай тауарлық позицияларға ие болады, бірақ тауардың әрбір позициясы тапсырыспен дәл байланысуы тиіс.

Екі сыртқы кесте басты кесте (сондай-ақ күшті объект, ағылш. strong entity), ал ортаңғы кестені туынды кесте деп аталады (қосылыстар кестесі, әлсіз объект, ағылш. weak entity).

Ескерту:

Реляциялық деректер қоры ортасында m:n байланысы сәйкес келетін туынды кесте арқылы 1: m және m:1 түрлендірілуі тиіс. Туынды кестеде негізгі кестелердің алғашқы кілттері сыртқы кілттер ретінде енгізілуі тиіс.

Байланыс қуаты деректер қоры жолдарының бір-бірімен байланыс түрін сипаттайды. Мүмкін болатын 16 қуатты ерекшелеп көрсетуге болады.

		тиіс	мүмкін		
		1	n	c	nc
тиіс	1	1:1	1:n	1:c	1:nc
	m	m:1	m:n	m:c	m:nc
мүмкін	c	c:1	c:n	c:c	c:nc
		mc:1	mc:n	mc:c	mc:nc

m, n бір немесе бірнеше (1-ге тең немесе одан көп), c 1 немесе 0 білдіреді

Ескерту: «мән – байланыс» типті үлгісін құру үшін міндетті емес (c) байланысты, сондай-ақ min, max белгілерін пайдаланудың мәні болмайды, бірақ, әйтсе де, автор оларды осында ескеруді жөн көреді. Диаграммаларды бейнелейтін қысқартуға бола міндетті емес байланыстардан толығымен бас тартатын номенклатуралар көп.

2.2 «МӘН – БАЙЛАНЫС» ТИПТЕГІ ҮЛГІ

«Мән – байланыс» типтегі үлгі (ERM) – ол деректерді үлгілеудің стандартталған әдісі. ERM деректердің құрылымын бағдарламалық жасақтаманың көмегімен олардың жүзеге асырылуын бейнелейді. ERM әзірлеу кезінде пайдаланушылар (клиент) мен әзірлеушілер (деректер қорының құрылымдаушылары) нақты жағдайды барынша дәл көрсету үшін тұрақты байланыс ұстайды.



ERM шыққан жері мән болып табылады. Мән – ол заттардың, адамдардың, түсініктердің немесе шынайы әлемнің абстрактілі идеяларының жеке және сәйкестендірілетін нұсқасы.

Мәндер мысалдары	
Белгіленуі	Мысал
Жеке тұлғалар	Қызметкер Харди, студент Троссманн, клиент Делер
Шынайы объектілер	Машина 2, белме 7, тауар 4711
Оқиғалар	төлем, брондау, басталуы, отырғызу
Абстрактілі түсініктер	сабақ, дәрістер, қызметтер

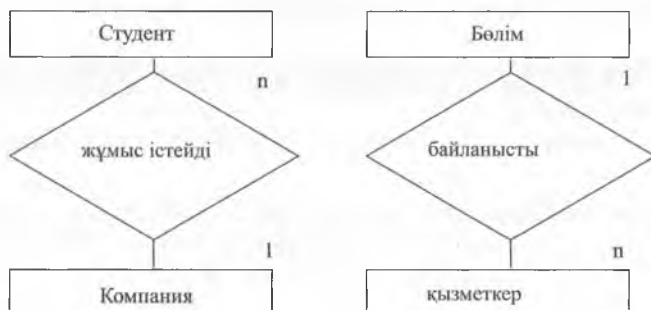
Мәнді сондай-ақ жиі объект деп атайды.

Құрылымы бірдей және бірдей сипаты болатын мәндер, мысалы, клиенттер, топтарға топтастырылады (**мәндер типтері**). Реляциялық үлгіде мәндің әрбір типі деректер қорының кестесі арқылы көрсетіледі, мысалы, Клиенттер кестесінде.

«мән-байланыс» типтегі үлгі сақталған деректерді және олардың өзара қатынастарын көзбен шолу және сипаттау үшін арналған. Ол үлгілеу деп аталады. Үлгілеу нәтижесі **ERD диаграммасы** болып табылады. Ол символдармен көрсетілген.

Чен бойынша көзбен шолуға арналған «мән-байланыс» типтегі үлгі символдары	
<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">Клиент</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">Есеп</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">Тапсырыс</div>	Бірдей қасиеттері бар мәндер ақпары біртұтас ортақ терминмен ER-үлгіде тік төртбұрышпен көрсетілген. Тік төртбұрышта мән атауы болады.
 алады	Мәндер арасындағы өзара байланыстар мен тәуелділіктер байланыстармен (relationships) берілген. Байланыстар негізінен етістіктермен сипатталады. Мысал: <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-right: 10px;">Клиент</div> — Алады — <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 10px;">№ 2 шотты алады</div>
	Өзара байланыстар жалғаушы жолақтармен берілген.

«Мән-байланыс» үлгісінің диаграммалары жоғарыдан төменге немесе солдан оңға қарай оқылады..



Атрибуттар

Объекттер атрибуттармен сипатталады. Мысалы, клиенттің нөмірі, аты, әкесінің аты, мекенжайы және тұрғылықты жері сияқты атрибуттары болады.

Ажыратады:

Түйінді атрибуттар немесе сәйкестендіру атрибуттары, мысалы, клиент номері, фирма номері, адам номері және сипаттама атрибуттары, аты-жөні, мақала атауы, құны сияқты.

Егер кестеде сәйкес келетін атрибуттар табылмаса, жасанды түйінді атрибут енгізіледі, мысалы, автоматты түрде артып отыратын сериялық номері.

Алғашқы кілттер жеке сызықпен, ал сыртқы кілттер пунктирлі сызықпен асты сызылады.

Ескерту:

Атрибуттар атомарлық болып табылмайды, яғни тым ұсақ бірліктерге «ажыратылуы» мүмкін емес.

Байланыстар

Мәндер арасында **байланыстар** болады. Мысалы, Тағылым-дамадан өтуші (1) компаниялардың бірінде жұмыс істейді. Және керісінше: бір компания бірнеше тағылымдамадан өтушілердің еңбегін пайдаланады. Бір бөлімге (1) бірнеше қызметкер тағайындалған, және керісінше, бір қызметкер тек бір бөлімге ғана қатысты болады. «Мән-байланыс» типтегі үлгіде 1:n байланысы мәндермен қатар көрсетіледі.



m:n байланысы

Объекттер арасындағы m:n байланысы әрбір А объектісіне бірнеше В объекттердің тәуелді болатынын білдіреді және керісінше әрбір В объектісіне бірнеше А объект қатысты болады. Мысалы, оқушы бірнеше курсты тыңдауы керек. Және керісінше, бір курста әрқашан бірнеше оқушы болуы тиіс. «Мән-байланыс» типтегі үлгіде оқушы мен курс арасындағы өзара байланыс көрсетілген.



ER-үлгі немесе ER-диаграмма үшін m:n байланыстары жоғарыда көрсетілген тәсілмен берілуі мүмкін.

m:n байланысын кестелік түрде бейнелеу мүмкін емес. Сондықтан мұндай байланысты екі 1:m байланысқа бөлу керек, яғни байланыс жеке кесте арқылы шығарылады (туынды кесте немесе байланысу объектісі).

Жаңа туынды кестеде кем дегенде екі сыртқы кестенің (негізгі кестелер) алғашқы кілті болады, қосымша атрибуттар әдетте ойға қонымды. Туынды кестенің аты заттардың шынайы қалпынан келіп шығады, мысалы, Жұмысбастылық. Егер мазмұнды атауын табу мүмкін болмаса, онда негізінен сыртқы атының үйлесімі таңдалады, мысалы, Студенттік курс.

Ескерту:

m:n байланысын орнату үшін байланысу объектісі қажет (туынды кесте).

Туынды кестені бейнелейтін кесте (мұнда: Жұмысбастылық) сыртқы кілт ретінде іске қосылған кестелердің (мұнда: Студенттер мен Курстар) алғашқы кілтін қабылдайды.



Әдетте жаңа алғашы кілт беріледі (мұнда: п-п №). Ерекшелік ретінде екі кілт те жаңа кестеде кейде алғашқы кілтті қалыптастыра алады. Қосымша атрибуттар осы кестеде пайдаланылуы мүмкін (мұнда: ескерту).

Питер Чен (Шен) бойынша «мән-байланыс» үлгісі:

1. Мәндер мен байланыстарды анықтау.
2. Мәндерге арналған сәйкестендіруші кілттерді анықтау.
3. Мәндер мен байланыстар типтерін анықтау.
4. Байланыстар параметрлерін баптау.
5. Атрибуттарды және олардың мәндерінің диапазонын (домейндер) анықтау.
6. Мәндер мен байланыстар жинағы үшін мән-байланыс диаграммасын (ERD) тұрғызу.
7. Алғашқы және сыртқы кілттерді анықтау.
8. Кестелерге атрибуттарды және олардың мәндерінің диапазондарын қосу.

2.3 «МӘН-БАЙЛАНЫС» ТИПТЕГІ ҮЛГІСІНЕ АРНАЛҒАН ШЕШІМІ БАР МЫСАЛДАР:

2.3.1 Тапсырысты өңдеу

Ұйымдардың бірінде деректер қорын пайдалана отырып, тапсырыстардың өңделуін ұйымдастыру керек. Әрбір тапсырыс үшін бір қызметкер жауапты (=персонал).

- а) Осы кестелер арасындағы байланысты анықтаңыз.
- б) Клиент, Тапсырыс және Персонал мәндері үшін тиісті атрибуттары бар кестелерді жобалаңыз.
- с) Алғашқы кілттің сәйкес келетін өрістерін анықтаңыз.

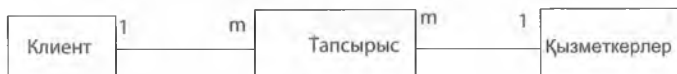
Шешімі:

Дизайн нұсқалары

А мәні	Байланыс	Б мәні
Клиент 1	береді m	Тапсырыс
Тапсырыс m	өңделуде	1 Қызметкерлер

Клиент: клиент номері, клиент тегі, клиент аты, мекенжайы, пошта индексі, клиент телефоны

Тапсырыс: тапсырыс номері, клиент номері, қызметкер номері, қызметкердің деректері: қызметкер номері, қызметкер фамилиясы, қызметкер аты



2.3.2 Жеткізушілер мен тауарлар

Компания бірнеше жеткізушілерден тауарлар алады, бір жеткізуші компанияға бірнеше тауар жеткізеді. Әрбір жеткізілімге бір тауар тең болады.

- а) Клиент пен Тауар мәндері үшін тиісті атрибуттары бар кестелерді жобалаңыз. ТауарЖетк m:n байланысын орнатуға арналған.
- б) Алғашқы кілттің сәйкес келетін өрістерін анықтаңыз.
- с) Осы кестелер арасындағы байланыстарды анықтаңыз.

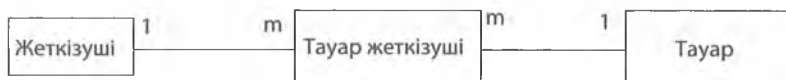
Дизайн нұсқалары

Мән А	Байланыс	Мән Б
Жеткізуші m	жеткізеді n	Тауар
Жеткізуші 1	қатысты m	ТауарЖетк
Тауар 1	таратылады m	ТауарЖетк

Жеткізуші: жеткізуші номері, жеткізуші фамилиясы, жеткізуші аты, жеткізуші мекенжайы, пошта индексі, телефоны

ТауарЖетк: жеткізілім номері, жеткізуші номері, тауар номері, күні, мөлшері

Тауар: тауар номері, тауар атауы, құны, қоймадағы бірліктер, қаптау бірліктері



2.4 2 БӨЛІМГЕ АРНАЛҒАН ЖАТТЫҒУЛАР

1. Оқытатын фирмада бір қатысушы бірнеше курстардан өте алады, сол кезде курстар кем дегенде 5 қатысушы болғанда өткізіледі.

Қатысушы және Курс мәндері үшін тиісті атрибуттары бар кестелерді жобалаңыз. Курстың Толымдылығы $m:n$ байланыстарын орнатуға арналған.

Дизайн нұсқалары

Мән А		Байланыс		Мән Б

2. Жалпы медициналық тәжірибеде «Medico» бірнеше дәрігер жұмыс істейді. Сонымен әрбір пациент әртүрлі дәрігерлерден ем қабылдай алады.

Дәрігерлер Пациент мәндері үшін тиісті атрибуттары бар кестелерді жобалаңыз. Емделу кестесі $m:n$ байланысын орнатуға арналған.

Дизайн нұсқалары

Мән А		Байланыс		Мән Б

3. Деректерді электрондық өңдеу жүйесі

Сіз ДК көп мөлшерде пайдаланатын кәсіпорында жұмыс жасайсыз. Сізге техникалық қызмет көрсету үшін және ДК-ден құжат айналымын жүзеге асыру үшін шектеулі деректер қорының жүйесін әзірлеу қажет. Талап етілетін ақпарат кәсіпорын компьютерлерінен сұраныс жасау үшін қолжетімді болуы тиіс.

Деректер қорында келесідей ақпарат болуы тиіс:

- Жүйелік әкімшілердің (көпшілігі) қайсысы әрбір нақты ДК үшін жауапты болып табылады.
 - Әкімшілердің әрқайсысының телефон номері мен кеңсесінің номері қолжетімді болуы тиіс.
 - Жүйелік әкімші езі жауап беретін компьютерлер қайда орналасқанын (кеңсе номері) білуі үшін деректер қорын пайдалана алады.
 - Әрбір ДК үшін қатқыл дисктің сыйымдылығын, сондай-ақ қандай операциялық жүйе мен желілік карта орнатылғандығын анықтап білу үшін конфигурацияны сақтау керек.
 - Жетекші әрбір нақты жүйелік әкімші нақты бір ДК-де қанша сағат жұмыс істегенін біліп алуы мүмкін.
- a) Көрсетілген деректер қоры үшін барлық қажетті ақпараты болатын ER-үлгі (мән-байланыс) құрыңыз.
 - b) Жаңа ER-үлгідегі (1:n.) туындайтын m:n байланыстарын орнатыңыз.
 - c) Деректер қорының нақты сипаттамасын байланыстардың сипаттамасы түрінде құрыңыз. Байланыстарда барлық алғашқы және сыртқы кілттерді нақты ерекшеленген тәсілмен көрсетіңіз.

4. Faradiso велосипед жалға алу

Faradiso компаниясы өз клиенттеріне велосипедтерді жалға береді. Велосипедті жалға беруді тікелей фирмада рәсімдеуге болады немесе алдын ала брондауға болады. Сонымен қатар әртүрлі бағыттар бойынша велокезеңдер ұйымдастырады. Туроператорға турдың бағыттары кесте түрінде беріледі (тур, сипаттамасы, турдың ұзақтығы, күрделілік дәрежесі, жөнелту пункті). Faradiso тек турдың рәсімделуіне және күндерді таңдауға ғана жауапты. Сізге техникалық қызмет көрсету үшін және компаниядағы бизнес-процестерді жүзеге асыру үшін деректер қоры жүйесін әзірлеу қажет. Сұраныс жасау үшін клиенттер мен велосипедтер туралы ақпарат қолжетімді болуы тиіс.

Деректер қорында келесідей ақпарат болуы тиіс:

- Деректер қорының көмегімен қандай велосипедтер, клиенттердің қайсысына және қандай мерзімге жалға берілгендігі туралы ақпаратты алуға болады?
 - Деректер қорының көмегімен қандай велосипедтер брондалғаны және клиенттердің қайсысымен брондалғаны туралы ақпаратты алуға болады?
 - Турдың белгіленген мерзіміне жазылып қойылған клиенттер саны.
 - Әрбір велосипед үшін: дайындаушы туралы ақпарат, атауы, типі (мысалы, туристік велосипед, тау велосипеді, және т.б.) рамасының өлшемі, сатып алу күні, сатып алу құны мен соңғы техникалық қызмет көрсетілген күні.
 - Әрбір велосипед үшін: әрбір жалдау күні үшін құны. Велосипедтерді бағалық категориялар бойынша жіктеу қажет.
- a) Көрсетілген деректер қоры үшін барлық қажетті ақпараты болатын ER-үлгі (мән-байланыс) құрыңыз.
 - b) Жаңа ER-үлгіде (1:n.) туындайтын m:n байланыстарды орнатыңыз.
 - c) Байланыстар сипаттамасы түрінде деректер қорының кестелерінің дәл сипаттамасын құрыңыз. Байланыстарда барлық алғашқы және сыртқы кілттерді нақты ерекшеленген тәсілмен көрсетіңіз.

5. Аурухананың бөлімшесін басқаруға арналған деректер қоры

Ауруханалардың бірінің бөлімшесін басқару үшін деректер қорын құрыңыз. Сұраныс жасау үшін пациенттер, дәрігерлер және кіші медициналық персонал туралы ақпарат қолжетімді болуы тиіс.

Деректер қорында келесідей ақпарат болуы керек:

- Қай дәрігерлер қандай пациенттерді емдейді?
- Палаталардың (бөлмелердің) қайсысы жаңа пациентке берілуі мүмкін?
- Хирургиялық бөлімшесінде бүгін қанша бос палата қолжетімді?

- ЛОР-бөлімшесінде бүгін қанша бос палата қолжетімді?
- Қандай палаталарға аға медбике Хильда жауап береді?

Ескерту:

Палаталардың әрқайсысы белгілі бөлімшелерге бекітілген. Дәрігерлер мен кіші медициналық персонал мүшелерінің (медбике/мейіргер) әрқайсысы белгілі бір бөлімшелерге бекітілген.

Мәндер: пациент, дәрігер, медбике/мейіраға, палата, бөлімше

6. Worldfly әуе компаниясы

Әуе компаниясына арналған деректер қорын құру қажет. Деректер қорында белгілі бір аэропорттарға қандай ұшақтар келіп қонатыны туралы ақпарат болуы тиіс. Сонымен қатар белгілі бір аэропорттарда ұшақ жасап шығарушылардың қайсысының техникалық қызмет көрсету пункті бар екендігі туралы ақпарат алу қажет. Әуе кемелеріне тиісті жасап шығарушылардың техникалық қызмет көрсету пункттерінде қызмет көрсетіле алады. Worldfly әуекомпаниясы әртүрлі жасап шығарушылардың ұшақтарын пайдаланады.

Деректер қорында келесідей ақпарат болуы тиіс:

- Нақты бір ұшаққа қайда қызмет көрсетіле алады?
- Нақты бір ұшақ қайсы әуежайларда қонады?
- Нақты бір ұшақ үшін әртүрлі әуежайларда қону ақысы қанша?

7. Кітапхана

Монастырлық кітапхананың кітап қоры туралы барлық ақпарат бір ғана деректер қорына жиналуы керек.

- Кітапханада белгілі бір кітаптардың бірнеше данасы ғана бар.
 - Әрбір кітап үшін деректер қорында келесідей ақпарат сақталуы тиіс: авторы, атауы, баспа, шығарылған жылы.
 - Халықаралық стандарттық кітап номері (ISBN) болмайтын салыстырмалы түрде ескі кітаптар бар.
 - Бір немесе бірнеше авторлардан бөлек кейбір кітаптарда басып шығарушысы да болады.
- a) ER-диаграммасы арқылы кестелердің байланыстарының графиктік көрсетілімін құрастырыңыз және әрбір кесте арасындағы байланыс типін көрсетіңіз.
 - b) Алғашқы және сыртқы кілттерді нақты белгілей отырып, маңызды атрибуттары бар кестелерді көрсетіңіз.

8. Дәрігерлік машық

Бір дәрігерлік машықта 8 дәрігер жұмыс жасайды. Олардың әрқайсысының қабылдауына тіркелу деректер қорының көмегімен іске асырылады. Әрбір қабылдау үшін тек бір дәрігер және бір ғана пациент белгіленеді. Дәрігерлер кестесінде дәрігердің фамилиясы мен телефон номері сақталған. Деректер қорында сондай-ақ пациенттер туралы ақпарат сақталуы тиіс: фамилиясы, аты мен туған күні. Деректер қорында нақты қабылдау күні, басталу уақыты мен аяқталу уақыты болуы тиіс.

- a) ER-диаграмманың көмегімен дәрігер мен пациент мәндерінің арасындағы байланысты бейнелеңіз.
- b) Бар байланыстарды көрсетіңіз.
- c) 1:n байланыстары арқылы жалғау объектісінің көмегімен қабылдау байланыс орнатыңыз.

9. BurgerLand жылдам тамақтану мейрамханасы

BurgerLand жылдам тамақтану мейрамханасының жеке филиалдарынан келіп түсетін деректер қорының көмегімен ұйымдастырылған. Қорда филиалдардың әрқайсысының аты, сонымен қатар позицияларының атауы, сатылым және сатып алу бағасы сақталады. Филиалдардың әрқайсысының деректер қорында күні көрсетілген сатылған позициялардың саны көрсетілуі тиіс.

- a) ER-диаграммасының көмегімен Филиал және Позиция мәндері арасындағы байланысты бейнелеңіз.
- b) Бар байланыстарды көрсетіңіз.
- c) Сатылым жалғау объектісі арқылы байланыс орнатыңыз.
- d) Барлық алғашқы және сыртқы кілттерді нақты ерекшеленген тәсілмен көрсетіңіз.

10. Бөлшектер қоймасы

Құрастырушы кәсіпорындардың бірінде белгілі бір монтаждау түйіндерін орнату жүргізіледі. Орнатуға жарамды құрамдас бөлшектері қоймада сақталады. Деректер қорының көмегімен бөлшектер қоймасын басқаруды ұйымдастыру қажет.

Әрбір бөлшектің атрибуттары бар: атауы, салмағы, қоймадағы ең аз қоры мен сипаттамасы.

Бірдей бөлшектер белгілі бір қоймалық орында сақталады. Өртүрлі қоймалық орындар өртүрлі рұқсат етілетін ортақ салмаққа есептелінген.

Әрбір бөлшек әртүрлі жеткізушілерден алынуы мүмкін. Деректер қорында жеткізушінің аты, оның мекенжайы мен телефон номері сақталады.

Жеткізушілер мерзіммен ағымдағы бағасымен және әрбір бөлшек үшін жеткізу уақытын көрсете отырып ұсыныстар жібереді.

Кәсіпорында Германияның барлық елді-мекендері бар кесте болады. Келесідей атрибуттар белгілі: елді-мекеннің сәйкестендіргіші, пошта индексі, елді-мекеннің атауы.

- a) ER-диаграммасының көмегімен мәндер арасындағы байланыстарды бейнелеңіз.
- b) Бар байланыстарды көрсетіңіз.
- c) Туындаушы M:N қатынастарын орнатыңыз.

11. Футболшылардың деректер қоры

Футболшылардың аттары бар және олардың ойын тәжірибесі бар деректер қорын құру керек. Деректер қорында ойыншының келесідей мәліметтері сақталады: тегі, аты, туған күні. Сонымен қатар деректер қорында келесідей ақпарат болуы тиіс:

Ойыншы қай ойындарда қатысты, ойыншы жалпы қанша уақыт (бастап, дейін) ойнады, қандай позицияда және қай команда үшін ойнады?

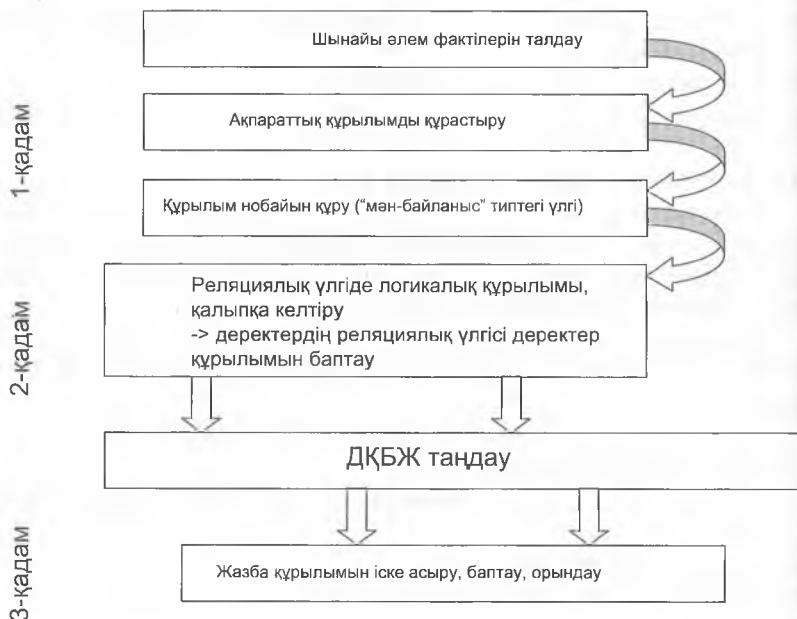
Деректер қорында команда атауын, жаттықтырушының аты мен жасын сақтау мүмкіндігі болуы тиіс.

- a) ER-диаграммасының көмегімен мәндер арасындағы байланыстарды бейнелеңіз.
- b) Бар байланыстарды көрсетіңіз.
- c) Туындаушы m:n қатынастарын орнатыңыз.

3 Деректер қорын әзірлеу және қалыпқа келтіру

3.1 ДЕРЕКТЕР ҚОРЫН ӘЗІРЛЕУ

Деректер қорын әзірлеу бірнеше кезеңмен орындалады.



Деректер қорын әзірлеу бойынша жұмыс кезеңдері

1-қадам: ақпараттық құрылымды баптау

Ең алдымен ақпаратқа нәтиже шығартушы талаптарды құрастыра отырып, шынайы әлем фактілерін талдауды жүргізу керек. Пайдаланушы деректер қоры жүйесінен қандай ақпаратты күтеді? Содан соң өзара байланыстардың графикалық көрсетілімі мақсатында «мән-байланыс» типтегі үлгі құрылады.

2-қадам: ақпараттық құрылымды баптау

Берілген кезеңде деректер мен кестелер арасында қандай байланыстар бар деген сұраққа жауап беруіңіз қажет. Содан кейін объектілерді (мәндерді) және олардың бір-бірімен қатынастарын жобалау жүзеге асырылады.

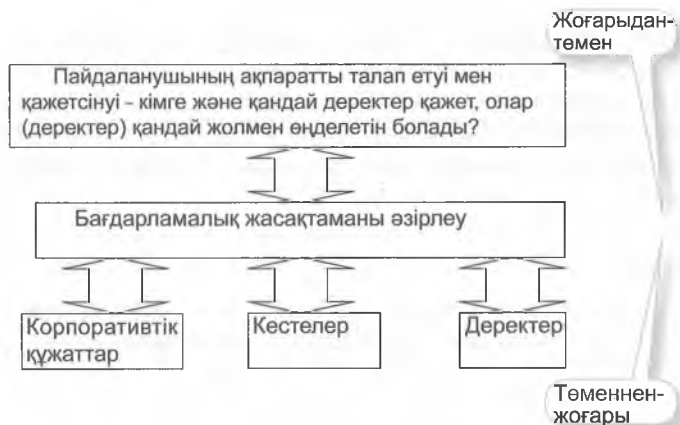
1 және 2-қадамдардың нәтижесі болып деректердің реляциялық үлгісі табылады, ол үшінші қалыпты формада көрініс табады (келесі бөлімді қараңыз).

3-қадам. Жазба құрылымын баптау

Деректердің типін анықтау үшін енді деректердің жеке кестелері мен өрістері анықталатын болады. Сәйкес келетін ДҚБЖ таңдалған соң бағдарламалық жасақтаманың көмегімен болашақ деректер қорының құрылуы жүреді.

1-қадамға сәйкес деректер қорының ақпараттық құрылымын тұрғызу төменнен-жоғары немесе жоғарыдан-төмен әдісімен жүзеге асырылады. Тәжірибеде, әдеттегідей, осы екі әдіс біріктіріледі.

3.1.1 Бағдарламалық жасақтаманы әзірлеу рәсімдері



«Төменнен-жоғары» әдісі:

«Төменнен-жоғары» әдісін қолдану кезінде деректер қорына енгізу керек ақпаратты шығару үшін кәсіпорында бар барлық құжаттар мен деректерді зерттеу керек. Жобалаудың осы әдісі кептеген шешімдерге әкелуі мүмкін. Өртүрлі тапсырмалардың шешімдері бір-біріне қабаттасу қаупі болады және бірдей деректер әртүрлі кестелерде бірнеше рет (артықшылық) сақталуы мүмкін. Артықшылықтан қашу керек. Қандай да бір мәнді өзгерткен кезде, мысалы, мекенжайды, деректер қорының жеке жолын редакциялау деректер қорында сәйкес өзгерістердің енгізілуін қамтамасыз етуі тиіс.

Ескерту:

Деректер қорының артық, міндетті емес жолдарының болуы **артықшылық** деп аталады. Реляциялық деректер қорында артықшылық туындамауы тиіс.

Жоғарыдан-төмен әдісі

Берілген әдісте деректердің үлгісі белек қосымшамен анықталмайды, алдымен деректер қорын кезекті барлық пайдаланушылары үшін ақпаратқа қойылатын талаптар анықталады. Осы әртүрлі талаптар негізінде деректер қорына қандай ақпарат, мысалы, клиенттер туралы ақпарат кіруі керектігі туралы қорытынды жасалады. Сонымен деректер қорының барлық қосымшалары үшін тек бір ғана клиенттер деген кесте болады.

3.2 ҚАЛЫПҚА КЕЛТІРУ

Ақпараттық құрылыммен берілген деректердің өрістері кезекті алдын ала жұмыс жасаусыз кестеге ауыстырылуы мүмкін болмайды. Көп реттік жазба жүргізілмеуіне кез жеткізу керек (артықшылықтың болмауы). Деректер кезең-кезеңімен әртүрлі кестелерге белінетін процесс **қалыпқа келтіру** деп аталады. Сонымен қатар әртүрлі қалыпты формалар сипатталады.

Ескерту:

Қалыпқа келтіру кезінде деректер өрісі логика ережелеріне сәйкес түрлі кестелер бойынша біртіндеп таратылады.

3.2.1 Қалыпты формалар

Бірінші қалыпты форма

Егер Персонал кестесінде Тұрғылықты жері өрісінде пошта индексімен қатар, елді мекен атауы да сақталса, онда бірнеше мәселе туындайды. Мысалы, Ульм қ. сияқты үлкен қалаларда бірнеше пошта индекстері болады, екіншіден, ол елді мекендердің атауы бойынша сұрыпталуын және мекенжайларды іздеуді күрделендіреді.

Сондықтан, алдымен, әртүрлі өрістерде мәндерді бөлу үшін қарапайым, қолжетімсіз ретінде құрамдас ақпаратты сақтау керек. Осындай мәндер атомарлық мәндер деп аталады.

Кесте: қызметкер			
жолаушының нөмірі	Тегі	Аты	Тұрғылықты орны
1	Палмер	Карло	89348 Йеттинген
2	Мюллер	Герта	86416 Крумбах
3	Май	Ханс	89077 Ульм
4	Шульц	Анна	88471 Лаупхайм
5	Винтер	Сюзанна	89312 Гюнцбург

Өрістерде 2 мәні болады

Кесте: қызметкер				
жолаушының нөмірі	Тегі	Аты	Тұрғылықты орны	Пошта индексі
1	Палмер	Карло	Йеттинген	89348
2	Мюллер	Герта	Крумбах	86416
3	Май	Ханс	Ульм	89077
4	Шульц	Анна	Лаупхайм	88471
5	Винтер	Сюзанна	Гюнцбург	89312

Өрістері құрамы бөлінген

Қызметкер деген кесте қалыпқа келтірілмеген және бірінші қалыпты формасында

Ескерту:

Егер деректер өрісінің бірде-біреуінде бөлуге болатын мәндерге ие болмаса, кесте бірінші қалыпты формасында болады.

Клиенттер, қоймадағы белшектер, сонымен қоса басқа да объектілер мен операциялар алғашқы кілттің өрісінің көмегімен нақты анықталуы мүмкін. Дербес номері бойынша іздеу кезінде қызметкерлердің барлық сақталған деректерін алуға болады, мысалы, аты мен мекенжайын. Тегі, аты, тұрғылықты жері атрибуттары функционалды түрде жеке номеріне тәуелді болады.

Екінші қалыпты форма

Кестенің алғашқы кілті, сондай-ақ бірнеше өрістерден құрастырылуы мүмкін. Сонымен, Тапсырыс позициялары кестесі тапсырыстың номері, тауар номеріне ие болады. Осы екі номерлердің комбинациясы мұнда алғашқы кілт ретінде орнатылуы мүмкін. Тауар туралы қосымша мәліметтер енді талап етілмейді, өйткені тауар № сыртқы кілті Тауарлар алғашқы кілт ретінде кестесінде деректер қорының жолында бірегей жазбасын білдіреді.

Кесте: Тапсырыс Позициялары		
	Өріс атауы	Деректер өрісінің типі
↔	Тапсырыс номері	Мәтін
↔	Тапсырыс номері	Мәтін
	Партия көлемі	Тұтас
	Тапсырыс құны	Валюта

Құрамалы алғашқы кілт: тапсырыс номері және тауар номері

Кілт белгісі бойынша танылатын құрамалы алғашқы кілті бар тапсырыс Позициялары кестесі

Ескерту:

Кесте екінші қалыпты формада болады, егер:

- Ол алғашқы қалыпты формада болса және
- Әрбір атрибутты тек алғашқы кілттің белігі ғана деп емес, тұтас алғашқы кілт деп сәйкестендіруге болатын болса.

Кілтке деген осындай тәуелділік толығымен функционалды тәуелділік деп аталады.

Құрамалы алғашқы кілті бар кестелер екінші қалыпты формасын әзірлеу кезінде ескеріледі.

Ескерту:

Екінші қалыпты форма құрамалы алғашқы кілттері бар кестелер үшін өте маңызды.

Тапсырыстың Позициялары кестесі екінші қалыпты формаға ауыстырылуы тиіс. Ол үшін алғашқы кілттерге тәуелсіз өрістер басқа кестеде сақталады.

Тапсырыс Позициялары кестесі мысалында қос кілтті өрістерге «Саны» атрибуты ғана тәуелді. Тауар туралы барлық деректер кілт бөлігімен берілген, дәлірек тауар Номерімен. Олар әрі қарай Тауарлар кестесіне беріледі.

Кесте: Тапсырыс позициялары	
	Өріс аты
↔	Тапсырыс номері
↔	Тауар №
	Тауар атауы
	Тауар құны
	Тауар мөлшері

Кесте: Позиция	
	Өріс аты
↔	Тапсырыс номері
↔	Тауар №
	Тауар мөлшері

Тапсырыс позициялары кестесін 2 кестеге бөлу, мысалы, Позициялар мен Тауар. Жаңа кестелердің екеуі де 2-ші қалыпты формада болады.

Кесте: Тауарлар	
	Өріс аты
↔	Тауар №
	Тауар атауы
	Тауар құны

Тапсырыс Позициялары мысалында 2-қалыпты форма.

Үшінші қалыпты форма

Егер кесте екінші қалыпты формада болса, онда оны үшінші қалыпты формаға келтіруін бастауға болады.

Ескерту:

Кесте мына шарт болғанда ғана үшінші қалыпты формада болуы мүмкін: 1) кесте екінші қалыпқа келтірілген және 2) алғашқы кілт құрамайтын байланыстар жоқ өрістер арасында

Ол өтпелі тәуелділік деп аталады, негізгі емес атрибуттар функционалды түрде басқа негізгі емес атрибуттардан ерекшеленбейді.

Осы талап қарапайым алғашқы кілті бар кестелерге таралады. Өзге негізгі емес өрістерге тәуелді деректердің өрістері жеке кестеге ерекшеленіп бөлінеді.

Ол Персонал кестесінде мүмкін. Ондағы «пошта индексі» атрибуты «елді мекен» атрибутына тәуелді. Осы тұжырым кішігірім елді мекендер үшін дұрыс, Гамбург немесе Штутгарт сияқты үлкен қалаларға ол таралмайды. Үлкен қалаларда пошта индексі көшеге немесе клиенттің пошта жөшігіне тәуелді. Дегенмен кері тәуелділік те болады: елді мекен пошта индексіне тәуелді, әрбір пошта индексі елді мекеннің атауымен дәл байланысты. Егер бірнеше елді мекен бір топқа біріктірілген болса, онда бұл атау біріккен қауымның атауы. Сонымен Елді мекен өрісі осы кестеден ерекшеленіп және өзінің Елді мекен кестесінде сақталуы мүмкін.

Кесте: Елді мекен		
	Өрістің аты	Өріс деректерінің типі
→	Пошта индексі	Мәтін
	Елді-мекен	Мәтін

Елді мекен кестесі қалыпты үшінші нысанда. Әрбір пошта индексіне тек бір елді мекен байланған.

Егер барлық кестелер қалыпқа келтіру ережелеріне сәйкес құрылып және редакцияланса, онда артықшылық болмайды. Қалыпқа келтіру деректер қорының логикалық құрылымының дұрыстығына кепілдік береді.

Қалыпқа келтірулер және ER-диаграммалар

ER-диаграммалар

- «Жоғарыдан-төмен» әдісі
- Жылдам
- Талаптарды тексеру
- Корпоративтік білім



Қалыпқа келтіру

- «Төменнен-жоғары» әдісі:
- Өте баяу
- Деректердің толық ауқымын өңдеу
- Математикалық негіздер

- «Жоғарыдан-төмен» әдісімен құру – «төменнен-жоғары» тексеру

НФ қалыпты формалары		
	Мәні	Ескерту
1-ші НФ	Барлық мәндер атомарлық (бөлінбейді)	Тізімдер болмайды
2-ші НФ	Әрбір негізсіз белгі алғашқы кілтке тәуелді	Бірінші қалыпты формаға қол жеткізілді, тек құрамалы алғашқы кілттерге назар аудару қажет
3-ші НФ	Әрбір негізсіз атрибут тікелей алғашқы кілтке тәуелді	Екінші қалыпты формаға қол жеткізілді, артықшылық кестелерде болмайды

Жеке деректердің кестелерге белінуі кемшілігі бірнеше кестені қамтитын ДҚБЖ арқылы іздеу үшін көп уақыт қажет ететіндігінде. Сондықтан ДҚБЖ өнімділігін арттыру мақсатында белгілі құрамдас бөліктерін ерекшелеуден бас тарту қажеттігі пайда болуы мүмкін, мысалы, пошта индекстерін. Осы процесс **денормализация** деп аталады.

3.2.2 Қалыпқа келтіру үшін мысалдар: қашықтықтан жасалатын сауда

Қашықтықтан сауда жасайтын компания шот деректерін электрондық кестелер көмегімен басқарады, мысалы, Excel. Осылайша жасалатын шот-фактураларда барлық маңызды деректер болады. Осы бағдарламалық жасақтама деректер қорымен алмастырылуы мүмкін. Берілген жағдайда қалыпқа келтіру 3-қалыпты формаға дейін жүргізіледі.

АЗЫҚ-ТҮЛІК ЖЕТКІЗУ

П/ж 1234, 88471 Лаупхайм, Тел +49739233333, Факс +49739233334

Ральф Бэр
Ан дер Хайде 7
83262 Н қаласы

Шот-фактура
Клиент номері
110

Күні: 20.12.2015
Шот номері
342

Саны	Қаптама	Тауар №	Белгіленуі	Қоймадағы саны	Құны	Бағасы	
2	0,75 л	G2	Қызыл шарап	3	3,99	7,98	€
3	6 x 0,5 л	G1	Сыра	2	6,59	19,77	€

Сомасы
ҚҚС-ты есептегенде

27,75 €
5,27 €

Шот-фактура бланкі

Қалыпқа келтірілмеген кесте

Алдымен шот бланкіндегі барлық деректер қалыпқа келтірілмеген кестеге ауыстырылады.

Шот бланкі		Клиенттер					Тауар						Саны
№	Күні	№	Тегі	Көше	Пошта индексі	Елді- мекен	№	Қап- тама.	Ата- уы	Қой- ма- дағы саны	Құны	ҚҚС	
342	20.12.2015	110	Бэр, Ральф	Ан дер Хайде 7	83262	Н қала- сы	G2, G1	0,75 л 6 x 0,5 л	Қы- зыл ша- рап сыра	3, 2	3,99 6,59	0,19 0,19	2, 3

Бірінші қалыпты форма

Енді қалыпқа келтірілмеген кестені алғашқы қалыпты формаға келтіру үшін атомарлық емес деректері бар бағандар жаңа кестеге Позиция деректері кестесіне жүктеледі. Нәтижесі болып Тапсырыс деректері кестелері мен бірінші ҚФ позициялар деректері табылады.

Шот номері	Шот күні	Клиент номері	Клиент аты	Көше	Клиенттің пошта индексі	Клиенттің елді-мекені
342	20.12.2015	110	Бэр, Ральф	Ан дер Хайде 7	83262	Н қаласы

Тапсырыс деректері
кестесі мен позиция
деректері 1-ші ҚФ-та

Шот номері	Тауар №	Тауар қаптамасы	Тауар атауы	Тауардың қоймада болуы	Тауардың құны	ҚҚС	Саны
342	G2	0,75 л	Қызыл шарап	3	3,99	0,19	2
342	G2	6 x 0,5 л	Сыра	2	6,59	0,19	3
...

Тапсырыс деректері кестесінде шот-фактура Номері бағаны алғашқы кілтті қалыптастырады. Позиция Деректері кестесінде алғашқы кілт шот Номері мен тауар Номері бағанында, ол құрамдас алғашқы кілт деп аталады.

Екінші қалыпты форма

Позициялар деректері кестесінде тауарлардың атрибуттары құрамдас бөліктердің бөлігінен, дәлірек, тауар номерінен тұрады. Сонымен бұл кесте екінші қалыпты формада болмайды. Қалыпқа келтіру мен кестені 2-ші ҚФ келтіру үшін тауар номері, сондай-ақ оған тәуелді барлық атрибуттар тауар Номері кілті бар жеке кестесіне біріктіріледі. Сөйтіп Тауар және Позиция кестелері қалыптастырылады.

Тауар №	Тауар қаптамасы	Тауар атауы	Тауардың қоймада болуы	Тауар құны	ҚҚС
G2	0,75 л	Қызыл шарап	3	3,99	0,19
G3	6 x 0,5 л	Сыра	72	6,59	0,19
...

Тауарлар мен позициялар кестелері 2-қалыпты формада

Шот номері	Тауар №	Саны
342	G2	2
342	G3	3
...

Ескерту:

2-қалыпты форма құрамдас алғашқы кілтті бар кесте үшін ғана маңызды.

Үшінші қалыпты форма

Тапсырыс Деректері кестесі 3-қалыпты формада болмайды, өйткені атрибуттар қатары клиент Номеріне тәуелді. Клиенттер кестесінде кілт ретінде пайдаланатын клиент номері Клиент Номеріне тәуелді атрибуттар мәліметі осы мәселені шешеді. Үшінші қалыпты формада тапсырыс туралы Мәліметтер кестесінде номер, шот шығару және клиент номері атрибуттары қалады.

Клиент номері	Клиенттің аты	Көше	Клиенттің пошта индексі	Клиенттің елді мекені
110	Бэр, Ральф	Ан дер Хай-де 7	83262	Н қаласы
***	***	***	***	***

Клиенттер кестесі үшінші қалыпты формада, барлық негізгі емес атрибуттар клиент номері негізгі атрибутына тәуелді.

Қашықтықтан жасалатын сауда кәсіпорны мысалындағы деректер қорын қалыпқа келтіру нәтижесі болып 5 кестенің пайда болуы табылады, олардың әрқайсысы үшінші қалыпты формада болады. Өрбір кестенің негізгі емес атрибуттары толықтай алғашқы кілтке тәуелді болады. Тиісті функционалдық тәуелділіктер бағдарлармен көрсетілген. Атрибуттар төменгі керсеткіште көрсетілген. Бірталай биік қалыпты формалар да болады. Алайда олар тәжірибе жүзінде қолданылмайды.

Ескерту:

3-қалыпты формада болатын кестелерде кілттерге немесе қатынастарға тәуелділіктердің барлығы сыртқы кілттерге келтіріледі. 1. Қадам:

1-қадам:

Excel кестесінен қалыпқа келтірілмеген деректерді ауыстыру

Қашықтықтан жасалатын сауда		
шот ном		
шот күні		
клиент ном		
клиент аты		
клиент көше		
клиент индекс		
клиент ө/мекен		
тауар ном	тауар ном	тауар ном
тауар атауы	тауар атауы	тауар атауы
тауар қойма	тауар қойма	тауар қойма
тауар құны	тауар құны	тауар құны
тауар ҚҚС	тауар ҚҚС	тауар ҚҚС
саны	саны	саны

Қалыпқа келтірілмеген деректер

2-қадам:

1-ші ҚФ келтіру

Тапсырыс деректері	Позициялар туралы деректер
шот ном	шот ном
шот күні	тауар ном
клиент ном	тауар атауы
клиент аты	тауар қойма
клиент көше	тауар құны
клиент индекс	тауар ҚҚС
клиент ө/мекен	саны

1-ші қалыпты формадағы деректер

3-қадам:

2-ші ҚФ келтіру

Тапсырыс деректері	Позициялар туралы деректер	Тауар
шот ном	шот ном	тауар ном
шот күні	тауар ном	тауар атауы
клиент ном	саны	тауар қойма
клиент аты		тауар құны
клиент көше		тауар ҚҚС
клиент индекс		
клиент ө/мекен		

2-ші қалыпты формадағы деректер

4-қадам:

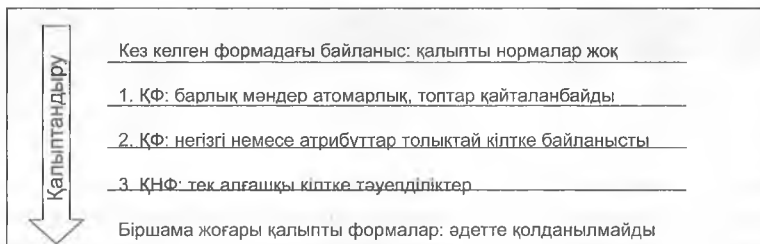
Бес қалыпқа келтірілмеген кестені

3-ші ҚФ келтіру

Тапсырыс деректері	Позициялар туралы деректер	Клиент	Тауар	Елді-мекен
шот ном	шот ном	клиент ном	тауар ном	ө/мекен индекс
шот күні	тауар ном	клиент аты	тауар атауы	ө/мекен атауы
клиент ном	тауар атауы	клиент көше	тауар қойма	
клиент аты	тауар құны	клиент индекс	тауар құны	
клиент көше	тауар ҚҚС	клиент ө/мекен		
клиент индекс				
клиент ө/мекен				

3-ші қалыпты формадағы деректер

Қалыпқа келтіру процесінің сызбасы



3.2.3 Басқа да қалыпты формалар

Сондай-ақ төртінші және бесінші қалыпты формалар болады, бірақ олар белгілі емес, өйткені мұндай жағдайларда қалыпты формалардың бұзылуы күтілу мүмкіндігі төмен. Ең алдымен мұндай бұзушылықтарды болдырмауды интуитивтік түрде қол жеткізуге болады. Атап айтқанда, егер біз кейін реляциялық жобаға түрленетін ER-үлгі құрылатын әдісті таңдасақ, онда үшінші қалыпты формада тестілеу жүргізу жеткілікті. Екінші және үшінші қалыпты форма деңгейіндегі бұзушылықтар мүмкін, бірақ оларды түптің-түбіне келгенде тексеру жолымен болдырмауға болады. Осыған сәйкес жағдайларда төртінші және бесінші қалыпты формалардың шарттары бұзылмауы мүмкін екенін болжауға болады.

3.2.4 Біртұтастылық шарттары

Деректер қорының тұтастылығы талабы сақталған деректердің бейнесі шындыққа толық сәйкес келетіндігіне бағытталған. Қате жазбалар, қызмет көрсетуді (жаңартылуларды) ескермеушілік деректер қорының пайдалылығына күмән келтіреді.

- Деректер типі мен мәндердің диапазондары деректердің қателерін анықтау мен болдырмау үшін жеке атрибуттар деңгейінде маңызды құрал болып табылады.
- Мүмкін қателердің басқа класы кілттің және сыртқы кілттердің (сілтемелік біртұтастылық) қасиеттерін тексеру үйлесімділігінде деректер қорының құрылымдық қасиеттерімен болдырылмайды.

Сілтемелік біртұтастылық

Деректердің сілтемелік біртұтастығы – ол ережелер жинағы, олардың кемегімен деректер қорын басқару жүйесі, мысалы, MS Access, егжей-тегжейлі кестелерде жазбалар арасындағы байланыстар шынайы екендігіне және байланысқан деректер кездейсоқ жойылған немесе

өзгертілгендігіне «көз жеткізіледі». Келесідей шарттардың барлығы орындалған жағдайда сілтемелік біртұтастылыққа қол жеткізіледі:

- Басты кестеден сәйкес өріс алғашқы кілт болып табылады немесе бірегей индексі болады.
- Екі кесте де бір деректер қорына жатқызылады. Сілтемелік біртұтастылықты орнату үшін кестелер сақталатын деректер қоры ашылады. Сілтемелік біртұтастылық басқа форматтағы деректер қорына байланысты кестелерге қолданылуы мүмкін емес.

Сілтемелік біртұтастылықты сақтаған жағдайда келесідей ережелер қолданылады:

- Егжей-тегжейлі кестенің сыртқы кілтінің өрісінде басты кестенің алғашқы кілтінің өрісінде болмайтын мәнді енгізуге болмайды. Бірақ сыртқы кілт өрісіне жазбалар бір-бірімен байланыспағанын көрсете отырып, нольдік мәнді енгізуге болады. Сонымен, мысалы, бар клиенттермен байланысты емес тапсырыс құруға болмайды. Алайда клиент номері ерісі нольдік мәнге ие болатын болса, «ешкіммен» байланысқан тапсырыс құрылуы мүмкін.
- Деректер қорының тиісті жолдары егжей-тегжейлі кестеге кіргізілген болса, басты кестедегі жазба жойылуы мүмкін болмайды. Мысалы, егер тапсырыс Деректері кестесінде деректер қорының жолы байланысты болса, Персонал кестесінен қызметкер туралы жазбаны жоюға болмайды.
- Егер осы жазбаға мәліметтер жазбасы болса, басты кестедегі алғашқы кілттің мәні өзгертілуі мүмкін емес. Мысалы, егер осы қызметкермен Тапсырыстар кестесінде жазбалар байланысты болса, Персонал кестесінде қызметкер номерін өзгерту мүмкін емес.

3.3 3-БӨЛІМГЕ АРНАЛҒАН ЖАТТЫҒУЛАР

1.

1-кестесімен танысыңыз:

- а) Бұзушылықтарда қалыпты формалардың қайсысы бар?
- б) Қалыпқа келтірілген формада баламалы жүйесін құрыңыз.

1-кесте

ISBN	Авторлар	Атауы	Шығарылған жыл	Беті
0-201-14192-2	Күні	Деректер қорын басқаруға арналған реляциялық үлгі: Басылым 2	1990	538
3-89319-117-8	Финкенцеллер Х., Краке У., Унтерштайн М.	SQL-Oracle жүйелі қолдану	1989	494
1-55860-245-3	Мелтон Й., Саймон А.	Жаңа SQL түсіну	1993	536

2.

2-кестемен танысыңыз:

- а) Бұзушылықтарда қалыпты формалардың қайсысы бар?
б) Қалыпқа келтірілген формада баламалы жүйесін құрыңыз.

2-кесте

Инд. студент №	Студент	К у р с №	Курстың атауы	Бағалау
30321	Й. Мейер	70656	Деректер қорлары жүйелері	1,0
30321	Й. Мейер	71554	Бағдарламалық жасақтаманы әзірлеу	1,7
30346	Х. Аренс	71554	Бағдарламалық жасақтаманы әзірлеу	3,0
30346	Х. Аренс	70656	Деректер қорлары жүйелері	2,0
30346	Х. Аренс	71355	Реляциялық және процедуралық бағдарламалау	1,7
30378	К. Кнудсен	70656	Деректер қорлары жүйелері	2,0

3. Жабдықты есепке алу

Келесі бейне жабдықты түгендеу деректерін көрсетеді (мысалы, мектептердің бірін). Деректер реляциялық деректер қорында қалыпқа келтірілген формада сақталуы тиіс. (Порт деректері келесі мәнге тең: M1 = IDE Port 1 Master, S1 IDE Port1 Slave, ...)

Жүйелік блок		Жауапты			Қатқыл дисктер			IDE-
Инвент. №	Орналасқан жері	Инд. №	Тегі	Телефон	Жасап шығарушы	Өнім №	Сыйымдылық	
L1001	B246	F100	М. Мейер	8975	Western Digital	102BA	1TB	M1
					Western Digital	102BA	1TB	S1
					Fujitsu	MPF3204 AT	2TB	M2
					Fujitsu	MPF3204 AT	2TB	S2
L1003	B251	F101	Ф. Биндер	5635	IBM	DTLA-305020	2TB	M1
					IBM	DTLA-307045	4TB	M2
					Western Digital	102BA	1TB	S1

Көрсетілген деректер ауқымына арналған реляциялық деректер қоры үшін қалыпқа келтірілген кестелерді әзірлеу керек. Келесідей әрекеттер алгоритмін ұстаныңыз:

- 1-қадам:** деректер ауқымын 2 кестеге бөліп, қалыпқа келтірілген бірінші форманы құрыңыз. Нәтиже ретінде олардың атрибуттармен қатынасын көрсетіңіз және алғашқы кілт пен сыртқы кілтін белгілеңіз.

2-қадам: егер қажет болса, 1-қадамдағы қатынастар үшін екінші қалыпты форманы құрыңыз. Нәтиже ретінде қайтадан олардың атрибуттарымен қатынастарын көрсетіңіз және алғашқы кілт пен сыртқы кілтін белгілеңіз.

3-қадам: қажет болған жағдайда 2-қадамдағы қатынастар үшін 3-қалыпты форманы құрыңыз және ортақ нәтиже ретінде қалыпқа келтірілген формада бастапқы деректер ауқымына енді сәйкес келетін олардың атрибуттары бар барлық қатынастарды көрсетіңіз.

Сұрақтар:

Қалыпқа келтіру жолымен қандай артықшылықтардың алдын алуы мүмкін еді?

2-ші емес 1-ші ҚФ сәйкес келетін кестелерде (1-қадамнан кейін) жазбаларды қойған кезде, езгерткен кезде және жойған кезде мәселелер туындайды. Осы кестелер негізінде формалардың әрқайсысына мысалдар келтіріңіз.

4. Әуе компаниясының багаж тізімі

Әуекомпаниясының багаж тізімін қадамдық қалыпқа келтіруін жүргізіңіз және оны үшінші қалыпты формаға келтіріңіз..

Әуе компаниясының номері:	WA876	Ұшақ типі	B 747	
Қаладан:	Мюнхен	КВС	Эбелинг	
Қала:	Гонolulu			
Әуе компаниясының күні:	01.04.2015			
Багаж тізімі				
Жолаушы				
Жіке куәлік номері	Тегі	Аты	Кешесі	Пошта индексі
00021238	Бреценху-бер	Ральф	Бекерштрассе 9	81669
				Мюнхен
74921209	Хиппхалер	Ханс	Бройштрассе 6	85049
				Ингольштадт
00554367	Майя	Маттиас	Брайнштрассе 12	85049
				Ингольштадт
Багаж				
Багаж номері	Типі	Салмағы (кг)		
1	Шабдан	20,5		
2	Серфингке арналған тақтай	6,5		
3	Жолдорба	7		
4	Тау велосипеді	35		
5	Шабдан	12		
6	Ангельруте	1,2		

Деректердің нормаланбаған массиві

№	из:	в:	Күні:	FT	FK	Жолаушы номері	N	VN	Кеше	Пошта индексі	Елді-мекен	Әлемдік атының тіркеу номері	A	G
WA876	MUC	HON	1.4.15	B747	Эбелинг	89...	Бреценху-бер	Ральф	Бекерштрассе 9	81669	Мюнхен	1	Шабдан	20,5
												2	Серфингке арналған тақтай	6,5

											3	Ж о л - 7	
				74...	Хипп	Ханс	Бройш- трассе 6	85049	И н - голь- штадт	4		Тау ве- лоси- пед	35
										5		Шаба- дан	12
				96...	Майя	Мат.	Б р а - й н ш - трассе 12	85049	И н - голь- штадт	6		А н - гельру- те	1,2

5. Қалыпқа келтірілмеген кесте берілген. Қадамдық қалыпқа келтіруді орындаңыз және кестені үшінші қалыпқа келтірілген формаға келтіріңіз.

Жеке номер	Тегі	Аты	Бөлім №	Бөлім	Ж о б а №	Сипаттамасы
1	Лоренц	София	1	Персонал	2	Сатуларды жылжыту
2	Холь	Татьяна	2	Сатып алу	3	Бәсекелестік талдау
3	Вильшрайм	Тео	1	Персонал	1,2,3	Клиенттерден сұрау алу, сатуларды жылжыту, Бәсекелестік талдау
4	Рихтер	Ханс	3	Сату	2	Сатуларды жылжыту
5	Визенланд	Вернер	2	Сатып алу	1	Клиенттерден сұрау алу

6. Қызметкерлер және олардың бөлімдерге қатыстылығы туралы деректері бар келесідей қалыпқа келтірілмеген байланыстар берілген. Сонымен қатар бүкіл компанияның жобаларына жеке қызметкерлердің қатысуы, сондай-ақ белгілі бір жобаларға қай бөлім жауапты екендігі белгіленген.

а) Қадамдық қалыпқа келтіруді орындаңыз және деректер ауқымын үшінші қалыпты формаға келтіріңіз.

Ескерту. Туынды байланыстарды бейнелеу кезінде атрибуттардың тиісті аттарымен шектеліңіз.

б) Маңызды алғашқы кілттерді енгізіңіз.

с) Қалыпты формаларды қалыптастыру ережелері негізінде өз қадамдарыңызды түсіндіріңіз.

	АТЫ	ТУҒАН КҮНІ	ЖҰМЫСҚА ҚАБЫЛДАҢУ МЕРЗІМІ	БӨЛІМ АТАУЫ	БӨЛІМ ЖЕТЕКШІСІ	ЖОБА АТЫ	ЖОБАНЫ ІСКЕ ҚОСУ	ЖОБАЛЫҚ- КОНСТРУКТО- РЛЫҚ БӨЛІМІ
Қиы- ғару	Зигги	04.05.65	01.06.90	Бақы- лау	Ж о - спар	Толық концерт үшін теңдестірілген көрсеткіштер жүйесі, ақпаратты басқару жүйесін әзірлеу	0 1 . 0 0 04.01	Информати- каны басқару
Ж о - спар	Руди	21.12.60	01.06.87	Ин - фор - мати - ка	Маус	Ақпаратты басқару жүйесін әзірлеу	04.01	Информа- тика
Бюкі	Петер	09.03.61	01.03.84	Ба - қылау	Ж о с - пар	Дайындық акцияларды біріншілік орналастыру	12.99	Бақылау
Түзету	Эрика	22.07.69	01.09.98	Ба - қылау	Ж о с - пар	BSC дайындық акцияларды біріншілік орналастыру	0 1 . 0 0 12.99	Бақылау Бақылау
Маус	Курт	30.01.55	01 01 72	Қаржы	Түзету	Дайындық акцияларды біріншілік орналастыру	12.99	Бақылау
Бюкі	Миха- эль	09.10.50	01.01.90	Ин - фор - мати - ка	Маус	Дайындық акцияларды біріншілік орналастыру	12.99	Бақылау
Гиро	Эмиль	16.04.68	01.09.97	Ба - қылау	Ж о с - пар	Дайындық акцияларды біріншілік орналастыру Ақпаратты басқару жүйесін әзірлеу Толық концерт үшін теңдестірілген көрсеткіштер жүйесі, Толық концертке арналған ССП	1 2 . 9 9 04.01 01 00	Бақылау Информа- тика Бақылау
Гиро	Герд	03 09 73 01.09.99		Қаржы	Түзету	Акцияларды алдын ала ор- наластыру	12 99	Бақылау

7. Студенттер, сыныптар, оқытушылар мен оқу уақыты туралы деректері бар оқу ұсынысы туралы қалыпқа келтірілмеген деректер массиві берілген.

Қадамдық қалыпқа келтіруді орындаңыз және деректер ауқымын үшінші қалыпты формаға келтіріңіз. Байланыс: оқу ұсынысы

Студент №	Тегі	Аты	Сыныбы	Оқытушысы	Оқу курсының №	Сипаттамасы	Сағат саны
	Юргенс	Ина	11a	Лемпель	2	Би	12
	Шмидт	Том	12a	Брайер	3	Хор	22
	Егер	Франц	11a	Лемпель	1,2,3	Электроника, би, хор	15, 12, 2
	Ольсен	Ина	11b	Зоммер	2	Би	5
	Юргенс	Паула	12a	Брайер	1	Электроника	23

4 Деректер қорын модельдеу үшін бағдарламалық қамтамасыз ету

Деректер қорын модельдеу үшін бағдарламалық қамтамасыз ету деректер қорын әзірлеу процесін қолдауды қамтамасыз етеді. Бағдарламалық қамтамасыз етудің лайықты пакеттерімен барлық өңдеу процесі есептеуіш техниканың көмегімен жүзеге асырылады. Деректер қорын жобалауға арналған типтік бағдарламалар – DB-Designer және Microsoft VISIO.

4.1 DB-DESIGNER

MySQL Workbench бағдарламалық пакетін интернет арқылы алуға болады. Бағдарламалық қамтамасыз етудің көмегімен деректер қорын жобалау және модельдеу, SQL бағдарламалау және MySQL деректер қорын басқару орындалады.

4.1.1 Жүктеу және орнату

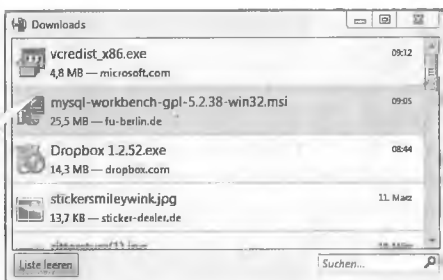
Жүйелік талаптар мен толық құжаттама MySQL үй парақшасында қолжетімді.



Жүктегеннен кейін, мысалы, msi пакет (Windows орнатушысы – Microsoft Software Installer), бұл пакет орнатуға дайын.

Іске қосу үшін орнатушыға екі рет басыңыз

Msi-пакет MySQL-



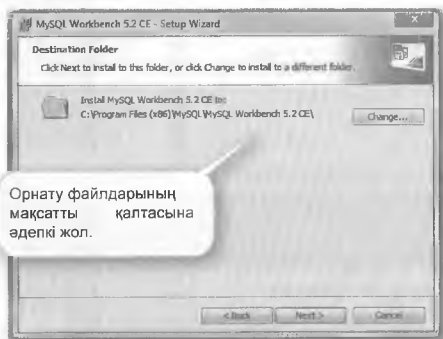
Сәлемдесу экраны

Орнату процесі орнату шеберімен бірге (Setup Wizard) жүреді. Автоматты орнату процесін бастау үшін Next > түймесін басыңыз.



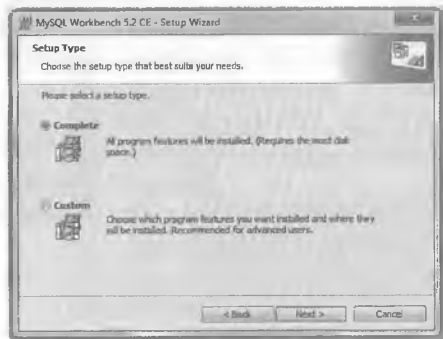
Destination Folder – тағайындау қалтасы

Алдымен орнату үшін қалтаны таңдаңыз. Әдепкі қалтаны өзгерту қажет болса, Change түймесін басу арқылы кез келген қалтаны таңдай аласыз ... таңдау



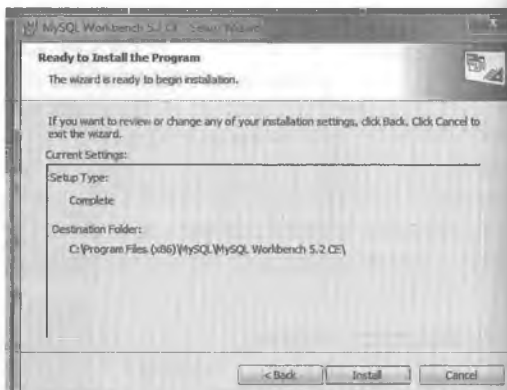
Setup Type – орнату түрі

Бұдан әрі орнату түрін таңдаңыз: толық орнату үшін «Complete» және таңдауда жеке бағдарламалық компоненттерді орнату үшін «Custom».

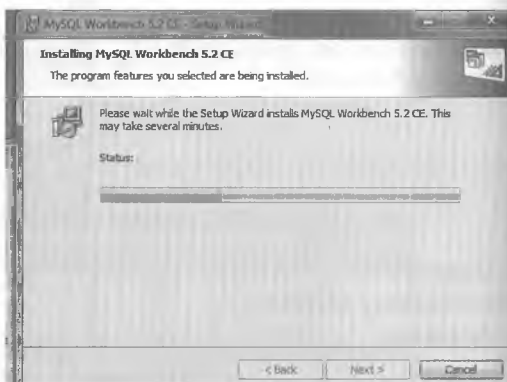


Ready to Install the Program – Бағдарламаны орнатуға дайын

Install түймесін басқанда, файлдарды әдепкі қалтаға немесе сіз таңдаған орынға көшіре бастайды.



Орнату күйі туралы апарат келесі орнату кезеңінде көрсетіледі.



Wizard Completed – орнату аяқталды

Орнату аяқталғаннан кейін шебер орнату аяқталғанын хабарлайды. Launch MySQL Workbench now жалаушасы – «MySQL Workbench іске қосу» – әдепкі жағдайда тұр. Finish түймесін басқаннан кейін MySQL Workbench бастапқы экраны жүктеледі.





SQL көмегімен әзірлеу: қолда бар деректер қорымен байланыс жасау SQL.

EER көмегімен деректерді модельдеу («мән-байланыс» типінің кеңейтілген моделі).

SQL серверін басқару.

SQL деректер қорына жаңа қосылу, кестелерді өңдеу, SQL скрипттерін өңдеу, байланысты басқару.

Жаңа EER-модельді әзірлеу, бар деректер қорынан жаңа EER-модельді әзірлеу, SQL сценарийінен жаңа EER-модельді әзірлеу.

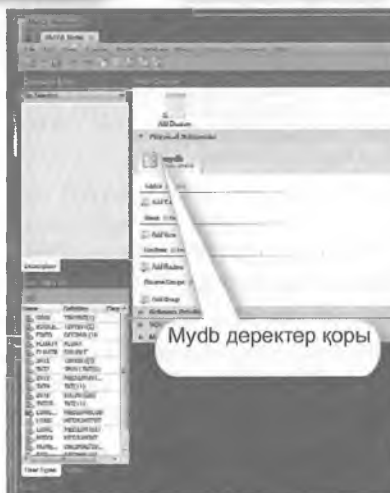
Жаңа сервер данасын жасау, басқару немесе экспорттау, қауіпсіздік

Бағдарламаның толық пакеті жоғарыда сипатталған бағдарламалық компоненттерден тұрады: SQL Development, Data Modeling және Server Administration.

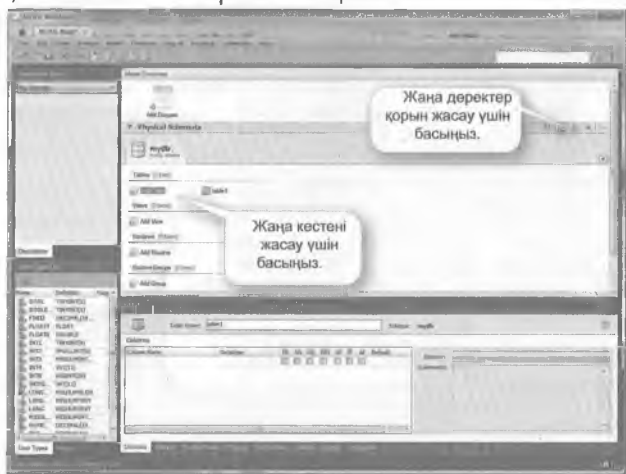
4.1.2 Кестелерді құру

Деректерді модельдеу үшін **Create New EER Model** мәзірін таңдаңыз (жаңа «мән-байланыс» үлгісінің кеңейтілген үлгісін жасау).

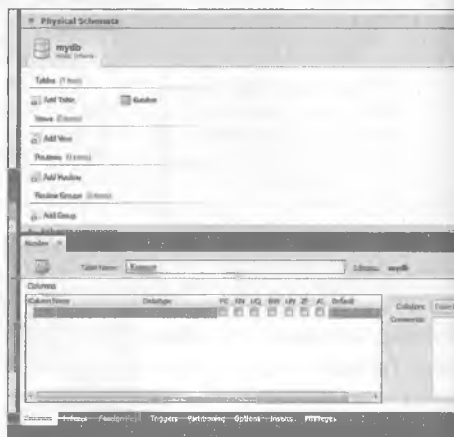
Mydb деректер қоры әдепкі бойынша құрылды.



«+» Таңбасын басу арқылы жаңа деректер қорын жасауға болады. MySQL деректер қорында жаңа кестені жасау үшін, мысалы, Клиенттер кестелерін, Add Table екі рет басыңыз.



Екі рет басқаннан кейін жаңа Table1 кестесі пайда болады; атауы талап етілетінге қарай өзгертілуі мүмкін, мысалы, Клиенттер деп. Columns бағандары баған аттары (Column Name) және деректер түрі (Datatype) бар кесте бағандарын анықтайды. Таңдау өрістері ретінде PK Primary Key, NN Not Null, UQ Unique, BIN, UN, ZF және AI параметрлері бар



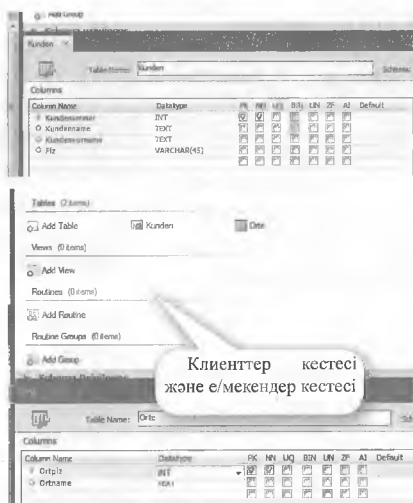
Жүйе автоматты түрде бастапқы кілтпен баған жасайды (IdКлиент). Бастапқы кілт өрісінде PK (Primary Key) және NN (Not Null) жалаушалары бар.



Әдепкі бойынша idКлиент деп аталатын баған аты, мысалы, Клиенттің нөмірі болып өзгертілуі мүмкін.

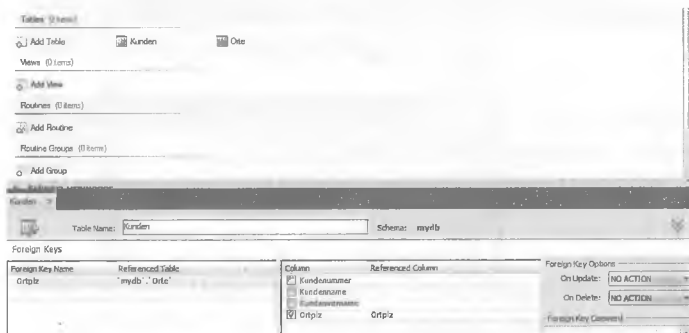
Қосымша өрістер, мысалы, Клиенттің тегі, Клиенттің аты және тиісті деректер типтері бар Пошта индексі жасалуы мүмкін.

Жоғарыда сипатталғандай, екінші кестені жасауға болады, мысалы, Е/мекендер және Пошта индексі және Елді мекен сияқты атрибуттарды орнатуға болады.



4.1.3 Кестелерді реляциялық байланыстыру

Екі кесте – Клиенттер және е / мекен 1: m қатынасында, бір индекс бірнеше Клиентті көрсетеді, бірақ әрбір Клиентке тек бір белгілі индекс (1) бекітілген.



Кестелер арасындағы қарым-қатынас бағдарламалық жасақтама арқылы қолмен және автоматтандырылған болуы мүмкін. Foreign Keys қосымшасында сыртқы кілттерді қолмен орнатуға болады.

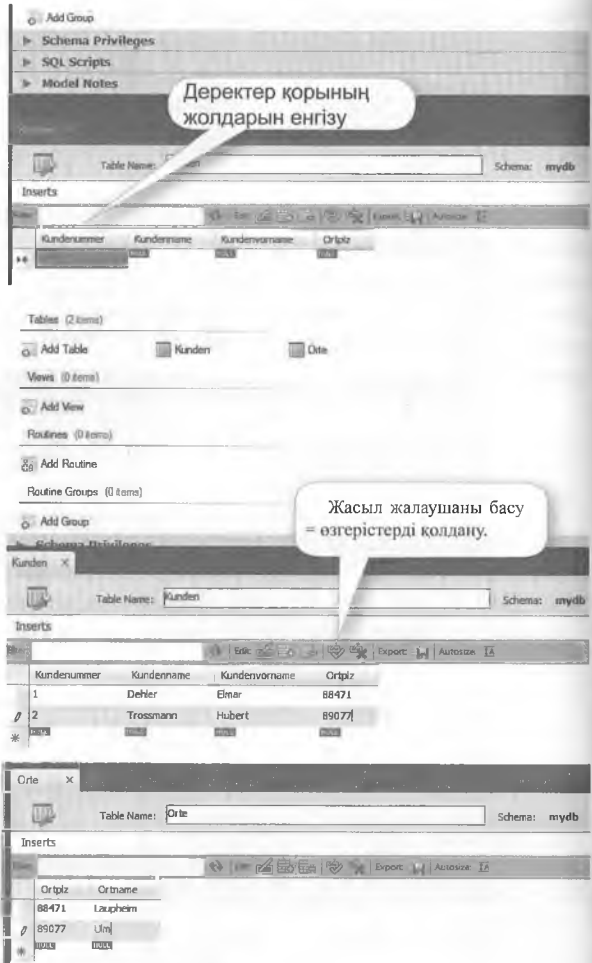
Foreign Keys бөлімінде Foreign Key Name (сыртқы кілттің атауы) ретінде сыртқы кілттің белгісі пайдаланылады, мұнда: Индекс және тиісті кесте (Referenced Table), мұнда: е/мекендер. Оң жақта тиісті Column бағаны таңдалады. Foreign Key Options бөлімінде жазбаларды өзгерту (ON Update) және жою (On Delete) кезінде сілтеме мәндерінің тәртібін орнатуға болады. Мүмкін болатын опциялар: NO ACTION = әрекет қажет емес, CASCADE = одан әрі өту, RESTRICT = тыйым және SET NULL = нелге орнату.

4.1.4 Деректер қорының жолдарын жазу

Деректер қорының жолдарын енгізу Inserts қосымшасын басу арқылы жүзеге асырылады.

Екі жазба енгізілгеннен кейін, Apply changes to data жасыл жалаушаны басу арқылы сақтау жасалады (деректерге өзгерістерді қолдану).

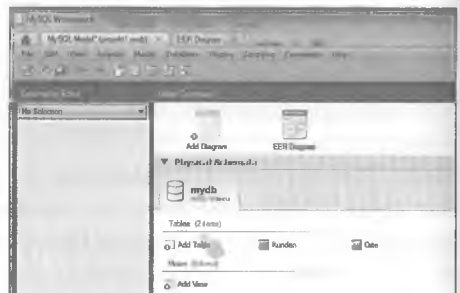
Осылайша, е/мекен жазба енгізуге болады.



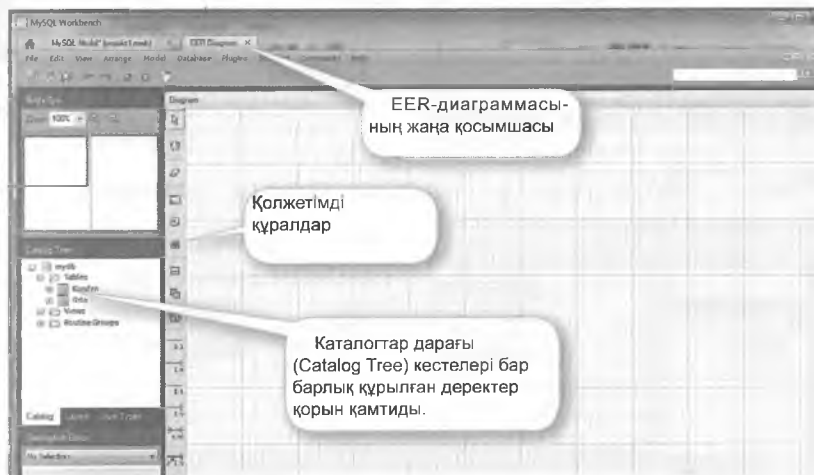
4.1.5 ER диаграммасын құру

Add Diagram командасының көмегімен екі кестеден тұратын деректер қорының диаграммасы жасалады.

Клиенттер және е /мекені Add Diagram екі рет бассаңыз, жаңа еер-диаграмманы белгіше түрінде жасайды және бір уақытта EER Diagram жазуы бар жаңа қосымшасын ашады.



Суретте EER-диаграммаларының редакторының терезесі көрсетілген.



Drag-and-Drop әдісі құрылған Клиенттер кестелері мен е/ мекенін жұмыс парағына апаруға мүмкіндік береді.



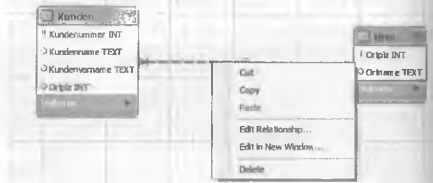
Графикалық көрініс (мұнда: Мартин нотациясы түрінде) екі кесте арасында 1:m қатынасын дұрыс көрсетуге көмектеседі, «қаз аяқтары» М қатынасын білдіреді.

Курсорды кесте арасындағы диаграммадағы қатынас сызығына апарғанда, ол тұтас болып, сыртқы кілт ретінде Клиент/индекс атрибуттарын және қатынасқа қатысатын бастапқы кілт ретінде е/мекен индексті көрсетеді.

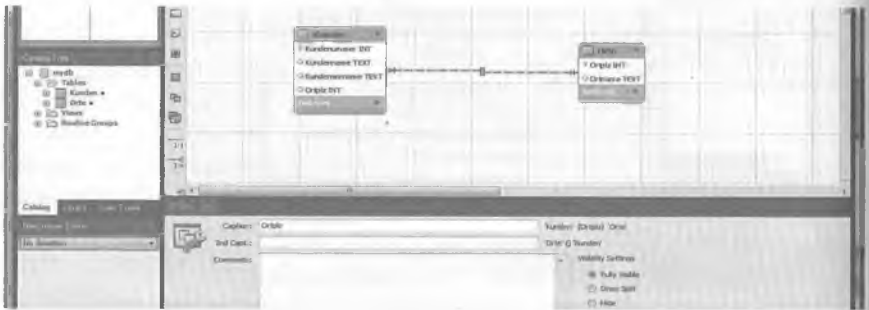


Навигацияны жеңілдету үшін сол жақ аймақта («Құс көзі») тұтас параққа қатысты екі кестенің орналасуы бейнеленеді.

Курсорды кесте арасындағы диаграммада қарым-қатынас сызығына апарғанда және тінтуірдің оң жақ батырмасын басқанда мәнмәтіндік мәзір пайда болады.



Edit Relationship... мәзірінің тармағы қатынасты еңдеуге мүмкіндік береді (жазуы)



Мудб терезесінде көрсетілгендей, EER диаграммасына кестелерді кірістіруге және жасауға болады. Диаграмма терезесінің сол жағында графикалық белгішелер бар. Жаңа кесте жасау үшін кесте белгішесін жұмыс парағына қарай апарыңыз.

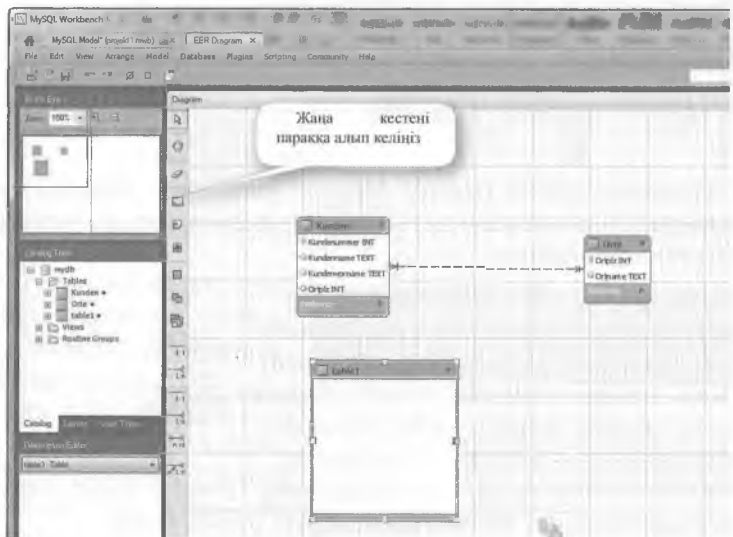
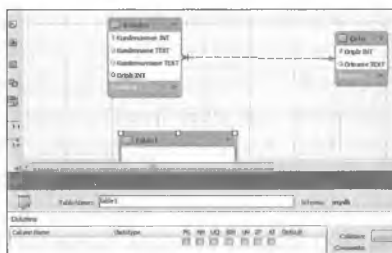
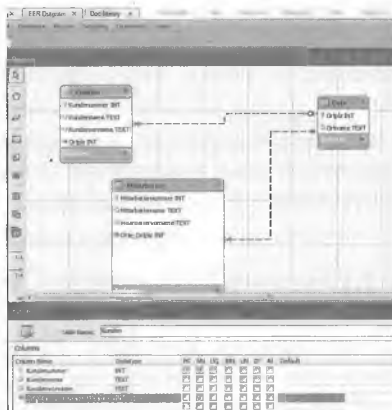


Table1 кестесі еңдеу режимінде өкі рет басқан соң ашылады. Көсте бағандары (атрибуттар) және олардың деректер түрлерін жоғарыда сипатталған тәсілмен өңгізуге болады.



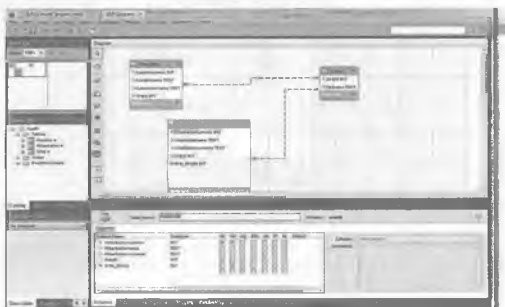
Бұдан әрі Клиенттер кестесінің атрибуттарындағыдай Қызметкерлер кестесі түрінде жаңа Table1 кестесі құрылады.



Қызметкерлер жаңа кестесін диаграмма терезесінің сол жағындағы құралдар тақтасындағы е/мекен кестесіне реляциялық қосу керек. Қызметкерлер кестесінде және е/мекендер кестесінде елді мекендердің индексі көрсетіледі. Нәтижесі суретте көрсетілген. Жүйе сыртқы кілттің жаңа атрибутын дербес жасайды (мұнда: Е/мекен_индекс).

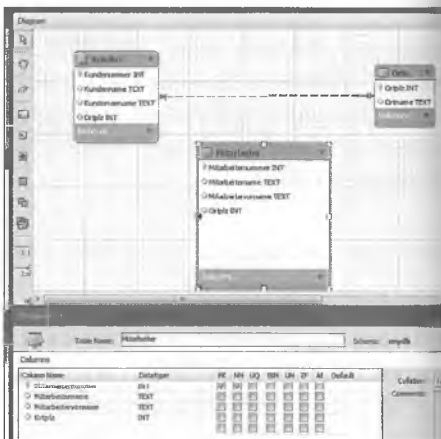
Ескерту:

ЕЕR-диаграмманың көмегімен ДҚ құру кезінде сыртқы кілт атрибуттармен құрылмауы тиіс, атрибуттар реляциялық жалғауда орнатылады.



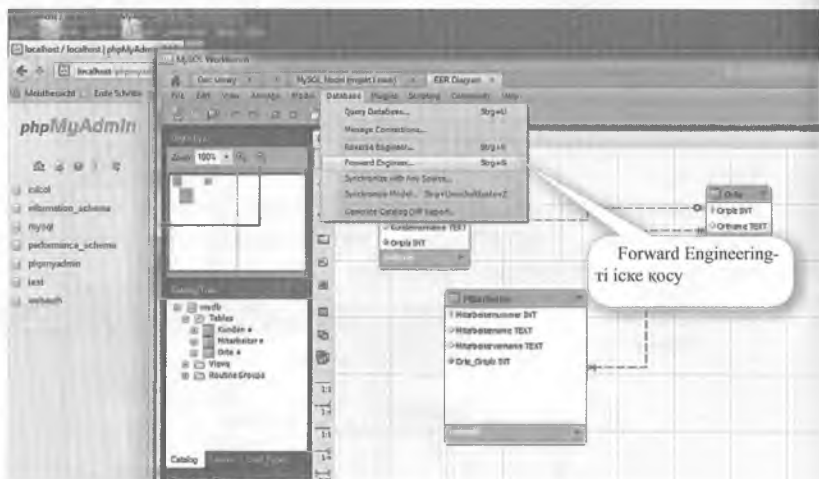
Толық аяқталуы үшін жоғарғы Индекс атрибуты өшіріледі.

Сонымен қатар жаңа сыртқы кілт басқа қаріп түсімен визуалды бөлінеді. Клиенттер кестесінде деректер де дәл осылай жасалуы тиіс. Клиенттер кестесінің белгішесін екі рет бассаңыз, терезенің төменгі жағында редактор іске қосылады. Бағанның (Column Name) атауының алдындағы ромбқа қайта басқанда, ромб түсі өзгереді. Ақ түс – кілтсіз атрибут, қызыл түс – сыртқы кілт (NN Not Null жалаушалары) немесе бастапқы кілтті белгілейтін кілт символы.



4.1.6 Forward Engineering (Тікелей жобалау)

Forward Engineering жүйелер мен бағдарламалық қамтамасыз етуді әзірлеуге арналған әдістер мен құралдарды ұсынады, мысалы: деректер қорының нобайы (моделі) негізінде іске асырылатын деректер қорын құру. Бұл деректер қорын ДҚБЖ-ға имплементациялауды немесе бастапқы SQL-кодты генерациялауды қамтиды.



Деректер қоры мәзірінде Forward Engineer ішкі мәзірі болады.... Forward Engineer немесе Ctrl+G пернелерді қатар басу арқылы шеберді іске қосуға болады.

Құрылатын деректер қорының параметрлері / параметрлерін орнату
Бірінші терезеде деректер қорын жабу үшін параметрлер таңдалатын болады.



Select Objects / Forward Engineer үшін нысандар таңдаңыз

Содан кейін Forward Engineering опциясы арқылы экспортталған нысандар таңдалатын болады.

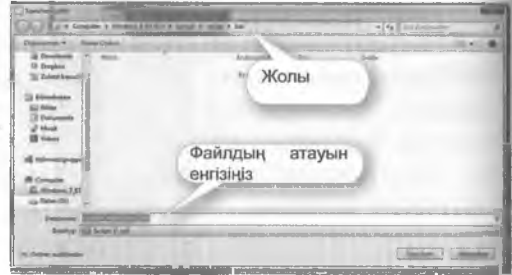


Review SQL Script / Орындалатын SQL-скриптіні қарау

Құрылған SQL-скрипті келесі қадамда көруге болады. Save to File ... «Файлға сақтау» батырмасын басқанда, SQL файлы жасалады, ал Copy to Clipboard «Алмасу буферіне көшіру» командасын таңдағанда... SQL файлы жұмыс үстеліне көшіріледі.

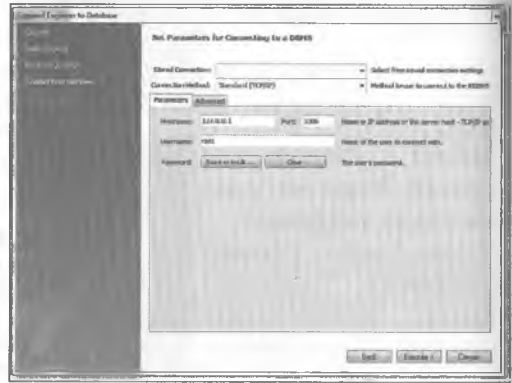


...түрінде сақтау терезесінде файл атауын және сақтау жолын енгізіңіз, мысалы, Клиенттер_Қызметкерлер_e /мекен



Connection Options / ДҚБЖ-ға қосылу үшін параметрлер орнату

Қосылым параметрлері бөлімінде қосылған SQL серверінің параметрлері енгізіледі, мысалы, Хост аты және Порт нөмірі. Орындауға басқан кезде көрсетілген процесс іске қосылады, яғни ДҚБЖ-ға қосылу және SQL сценарийін орындау.



Маңызды: веб-сервер және MySQL іске қосылуы тиіс.

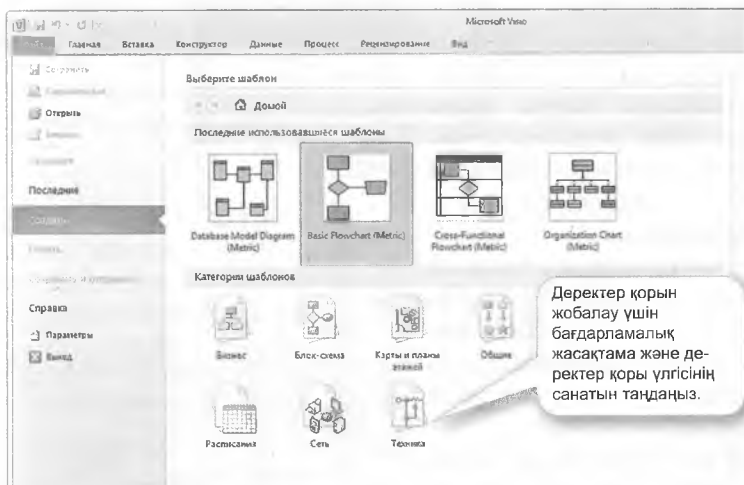
Progress / Forward Engineering операция прогресі

Forward Engineering процедурасының прогресі Forward Engineering Progress экранында көрсетіледі.

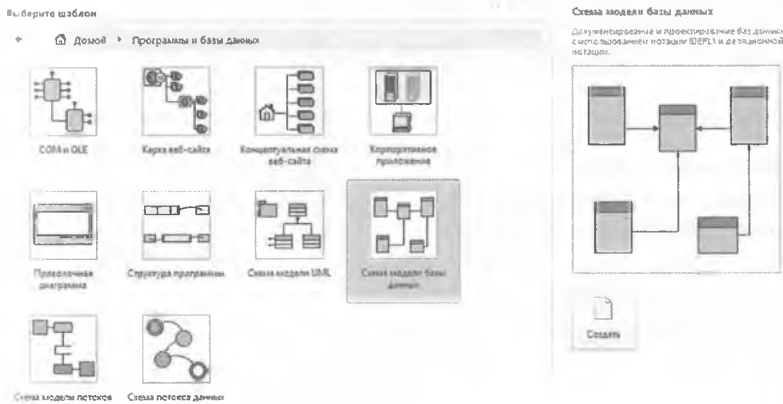


4.2 MICROSOFT VISIO

4.2.1 Деректер қоры моделінің диаграммасын іске қосу
Microsoft Visio 2010 бағдарламасын іске қосқаннан кейін,
бағдарламалық жасақтама белгішесін және деректер қорын таңдаңыз.



ДҚ үлгісіне диаграмма шаблонын таңдау үшін Шаблон таңдауды басып, оны Жасау түймешігін басып растаңыз. Деректер қоры моделіне арналған диаграмма үлгісі кері жобалауды (Reverse Engineering) пайдалана отырып, қолданыстағы деректер қоры негізінде нысан қатынасының жаңа моделін (ER-модель) немесе модельді жасауға мүмкіндік береді.



Сол жақ аймақта – тиісті әзірлеу құралдары бар фигуралар деп аталатын шаблондар, оң жақ облыста

ER моделі жасалатын жұмыс парағы бар.

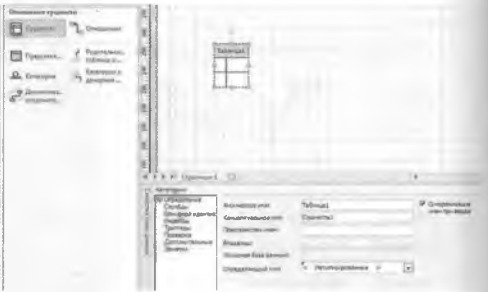
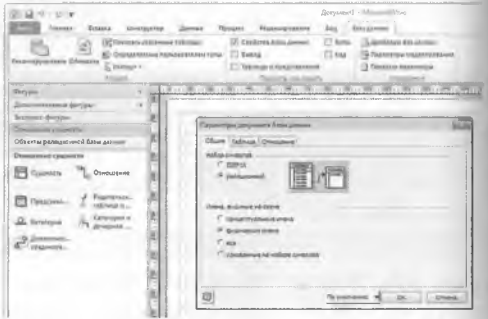
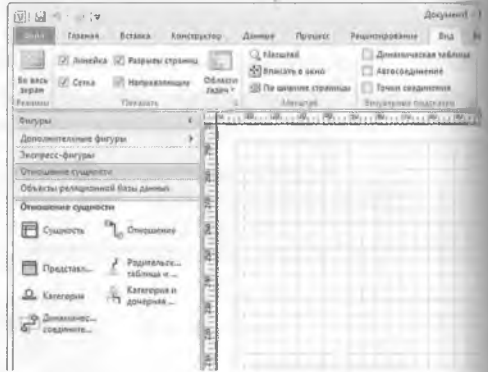
«Мән-байланыс» (Entity Relationship) шаблонны негізінде SQL92 стандарты және алдыңғы стандарттар негізінде деректер базасын модельдеу үшін қолданылады.

Реляциялық-нысандық шаблон SQL99 стандарты және одан кейінгі стандарттар негізінде деректер қорын модельдеу үшін қосымша құралдардан тұрады.

Басқару тобының деректер қоры қосымшасында көрсету параметрлерін таңдауға болады. Ашылған құжат – деректер қоры параметрлері терезесінде, мысалы, реляциялық және басқа да кестелер мен қарым-қатынастар параметрлері сияқты диалогтық терезедегі таңбалар жиынтығын таңдауға болады.

4.2.3 Кестелер жасау

«Мәндер қатынасы» шаблоннан «Мән» белгішесі параққа (Drag and Drop) апаруға болады. Белгішені екі рет шертсеңіз, Деректер қорының сипаттары терезесі ашылады.



Анықтама бөліміндегі сандарда мысалы, Клиенттер деп, кесте атауы (жеке аты) енгізілуі мүмкін. Санаттар / Бағандар бөлімінде атрибуттарды, мысалы, Клиент нөмірін, ал деректер түрін, мысалы, бүтін санды ашылмалы тізімнен таңдауға болады.



Клиент нөмірі бастапқы кілт болып табылатындықтан, Міндетті баған бос мәндердің пайда болуын болдырмайды (NN Not Null сәйкес келеді).

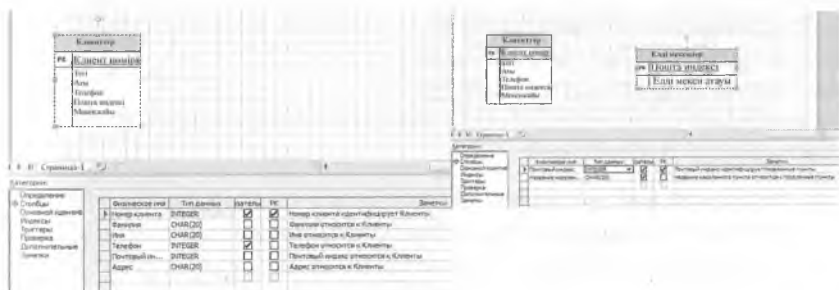
Бағандарға арналған РК (Primary Key = бастапқы кілт) жалаушасын орнатқанда, деректер қоры кестесіндегі жеке жолдар бірегей идентификаторды (UQ Unique сәйкес келеді) алады. Нәтижесі – бастапқы кілтті қысқарту және атын сызу.



4.2.2 Бағандар қосу

Деректер қоры/бағандар ерекшеліктері бөлімінде қосымша атрибуттар мен оларға арналған деректер түрін енгізуге болады. Жалаушаны орнату міндетті түрде нелдік мәндерді енгізбеуге мүмкіндік береді, мысалы, туған күн үшін.

Жоғарыда сипатталғандай, Клиенттер кестесі және қосымша атрибуттары бар Е/мекендер кестесі жасалады.



Файл / ... түрінде сақтау сияқты командасын таңдағанда, Visio үлгісі Kunden.vsd сияқты жаңа файл атауымен сақталады.



Файл атауын енгізіңіз

Сақтау жолы

4.2.4 Байланыстар қосу

Байланыстарда Деректер қорлары кестедегі жолды байланысқан кестедегі (мысалы, customer.plz). жолмен байланыстыратындай бастапқы және сыртқы кілттер қолданылады (мысалы, Orte.plz) Бұл байланыстар диаграммада көрсетілуі мүмкін. Одан басқа, байланыс қуатын (мысалы, 1: n) көрсетуге болады және оны Мартин нотациясын, реляциялық нотацияны немесе IDEF1X нотациясын пайдалана отырып көрсетуге болады.

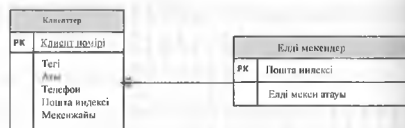
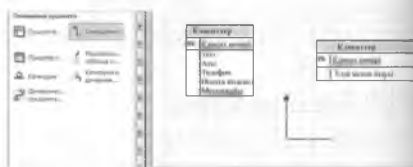
Ескерту:

m: n қатынасы (жиын) Мартин нотациясын, реляциялық нотацияны немесе IDEF1X нотациясын пайдалана отырып, деректер қоры моделінің диаграммасының шаблонында көрсетілуі мүмкін емес.

Мән-байланыс шаблоннан байланыс белгішесі параққа апарылады да, бос орынға орналас-тырылады.

Көрсеткісі бар ұшы еншілес кестемен (бағынышты кестемен) байланысты, мысалы, Клиенттер, ал екінші ұшы – негізгі кестемен (бас кестемен), мысалы, E/ мекендермен.

Егер еншілес кестеде бастапқы кілт сияқты аты бар баған болмаса, Visio ішкі кілтті еншілес кестеге қосады.

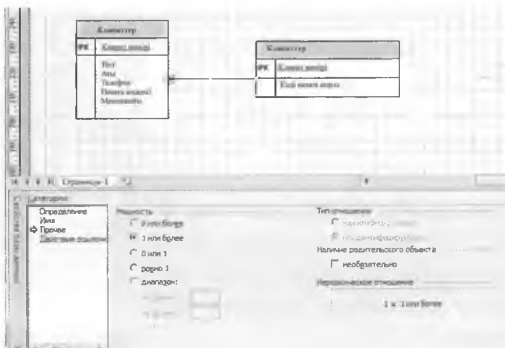


Ескерту

Егер қатынас жолдары бұдан былай көрсетілмесе, «Деректер қоры» қосымшасындағы Басқару тобынан Көрсету параметрлерін таңдаңыз. Байланыс қосымшасындағы «Көрсету» белімінде «Байланыстар» жалаушасын қойыңыз.

Байланыс қуатын орнату:

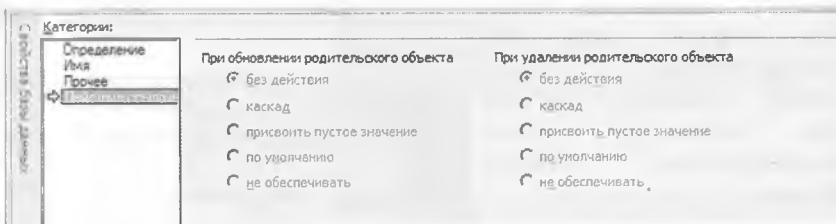
Қатынас жолын екі рет басу арқылы деректер қорының «Деректер қорының типаттары» терезесінде санаттарды таңдаңыз: әртүрлі. Байланыс қуаты параметрінде бір немесе жиынтықты таңдауға болады.



Ескерту

1:n байланыстары үшін нөл немесе жиынтық немесе бір немесе жиынтық тура келеді. 1:1 байланыстары үшін нөл немесе бір, немесе бір.

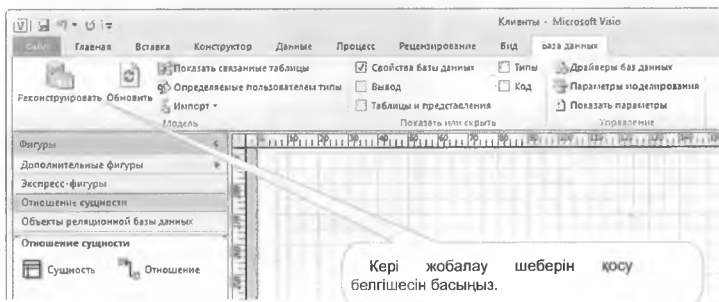
Сілтеме әрекеті бас элементті жою немесе жаңарту кезінде, мысалы, Нөлге орнатуда (SET NULL) жаңарту тәсілін анықтайды немесе әрекет талап етілмейді (NO ACTION).



4.2.5 Реверсивті инжиниринг

Қолда бар деректер қорын Кері жобалау ассистенті (Reverse Engineering) Visio арқылы импорттауға және «мән-байланыс» типінің үлгісі ретінде ұсынуға болады.

Деректер қоры қосымшасында Кері жобалау (Reverse Engineering) белгішесін басыңыз.



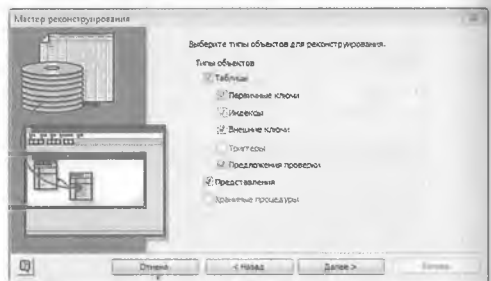
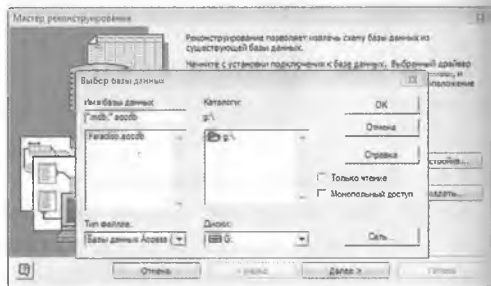
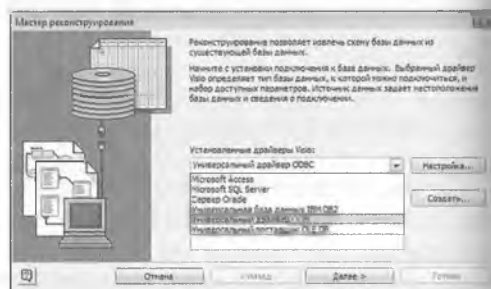
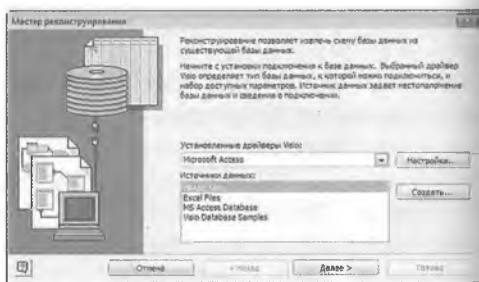
Дереккөздер бөлімінде тиісті дерек көзі таңдап алынады және деректер қорын басқару жүйесі үшін дерекқор драйвері орнатылады (ДҚБЖ). Visio деректер қорының драйвері әлі нақты ODBC деректер көзіне байланысты болмаса, оны орнату бөлімінде жасауға болады.

Егер Excel парағы үшін кері жобалау қажет болса, онда әмбебап ODBC-драйверді таңдау керек.

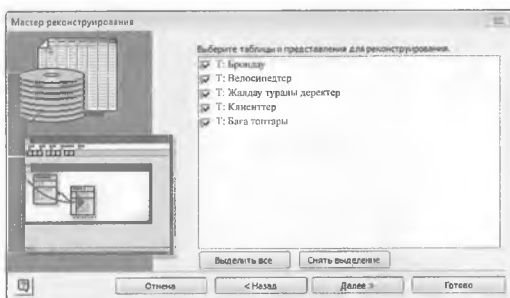
Тиісті Visio драйверін анықтау шебер өз деректер түрлерін дұрыс салыстыратынына және шебер шығарған бүкіл код терезесінде дұрыс көрсетілетініне кепілдік береді.

Келесі терезеде сияқты деректер қорының дереккөзін таңдаңыз, (мысалы, Faradiso.accdb) ол Visio-ға импортталуы тиіс. Егер қолда бар деректер қоры әлі құрылмаған болса, оны Құру батырмасын басу арқылы жасауға болады.

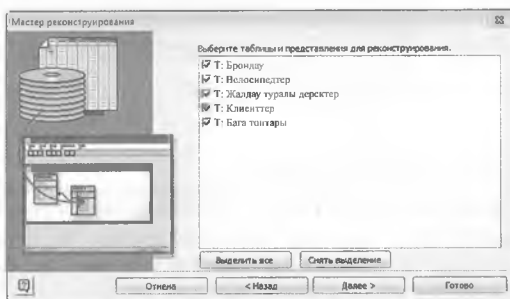
Енді шебер кері жобалау арқылы таңдалуы тиіс нысан түрлерін сұратады.



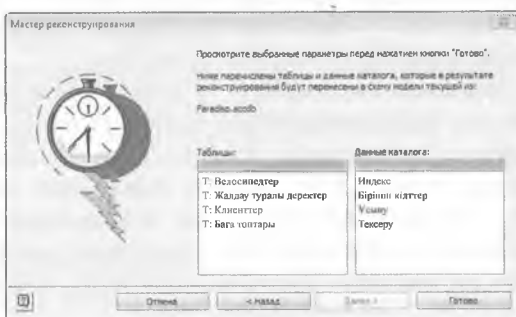
Нысандар түрін таңдағанынан кейін кестелер көрсетіледі. Кейбір нысандарды жеке жалауша қойып немесе Барлығын таңдау опциясымен бірге таңдауға болады.



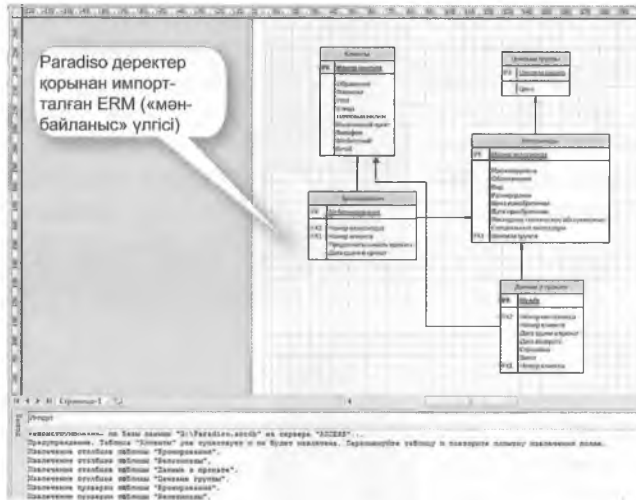
Дайын батырмасын басқаннан кейін, сызба парағын автоматты түрде көрсету керек пе, әлде осы мүмкіндік кейінірек қосыла ма деген шешім қабылдау қажет.



Таңдалған параметрлер туралы жиынтық ақпарат шебердің жұмысын аяқтау терезесінде көрсетіледі.



Шебер таңдалған ақпаратты импорттайды және төменде қорытынды терезесінде импорттау туралы ақпаратты көрсетеді. Біртіндеп барлық кестелер мен байланыстарды тексеріп шығаруға болады.



4.2.6 Индекстерді құру

Индекстерді орнату сұранымды орындағанда деректер қорының өнімділігін немесе жылдамдығын арттырады. Индекс кіру жолы ретінде анықталады, ДҚБЖ әдетте индекс бойынша сұрыпталған

кестені сұранысты жеделдету үшін индекстелген мән, мысалы, пошта индексі арқылы сақтайды. Басты мәндер әрқашан индекстеледі. Теменде faradiso деректер қорының Клиенттер кестесінде жасалған индекстер бар.

Клиенттер	
PK	Клиент нөмірі
	Жүгінді
	Тегі
I2	Аты
	Көшесі
I1	Пошта индексі
	Елді мекен
	Ұялы телефон
	Email

Жасалған индекстерді I1 және I2 әріптері бойынша ER-диаграммада білуге болады. Осылайша, бұл мысалда пошталық индекс және тек индекстеледі. Егер жұмыс парағында клиенттер кестесін басып, Санат аймағындағы Индекстер таңдасаңыз, тағайындалған индекстер көрсетіледі және олар редакциялануы мүмкін.

Бар индекстерді көрсету үшін индекстер жалаушасын қойыңыз.



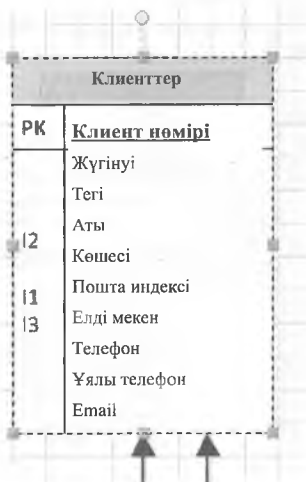
Құру батырмасын басқан кезде жаңа индекс жасалады. Индекс құру диалогтік терезеде индекс атауы енгізіледі, мысалы, Е / мекен. ОК батырмасын басқан кезде Индекс құру диалогтік терезесі жабылады.

Индекс түрі тізімінде бірегей немесе бір мәнді индекс жасау үшін параметрді таңдаңыз. 4 нұсқасы қолжетімді.

Қолжетімді бағандар тізімінен жаңа индекске енгізілетін бағандар аттарын таңдаңыз. Тиісті бағанды таңдау үшін Қосу батырмасын басыңыз.

Индекстелген бағандар тізімінде индексті сұрыптаудың өсу ретімен құру үшін Қосу жалаушасын қойыңыз. Жалаушаны өшіргенде индекс кему бойынша сұрыптау гәртiбiмен құрылады.

Деректер қоры моделінің диаграммасында 13 жаңа индексі кейінірек Е/мекен атрибутында көрінеді.

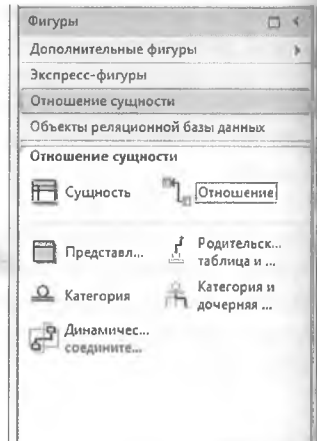


4.2.7 Көріністер құру (Views)

Көрініс (View) сақталған сұраныс болып табылады. Олар әсіресе бір ақпаратты бірнеше кестеден қайта сұрағанда пайдалы. Көріністерді деректер қорының әртүрлі пайдаланушыларын (мысалы, клиенттер үшін деректер) кестелерді нақты өзгерту мүмкіндігінсіз керсету үшін пайдалануға болады.

Көріністі құру үшін Мән-байланыс немесе Реляциялық объект шаблонынан көрініс нысанын сызба бетіне қарай апарыңыз.

Көрініс нысаны



Жаңа 1-кесте көрініс болып табылады

Клиенттер	
PK	Клиент нөмірі
	Жүгінуі
	Тегі
12	Аты
	Көшесі
11	Пошта индексі
13	Елді мекен
	Телефон
	Ұялы телефон
	Email

Көріністі анықтау Санаттар бөлімінде жүреді.

Кесте 1
Мән 1

Категория:

Физическое имя:

Концептуальное имя:

Пространство имен:

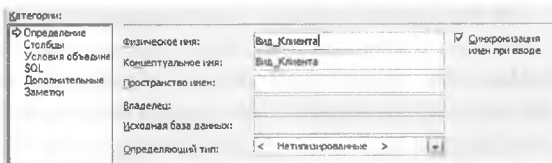
Владелец:

Исходная база данных:

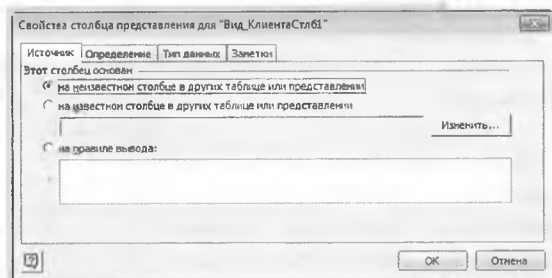
Определяющий тип:

Синонимизация имен при вводе

View_Kunden сипаттары, яғни Клиенттер көстесінің көрінісі енді сипаттарда көрсетілген. Анықтама мәзірінде View_Kunden сияқты атау орнатылады.

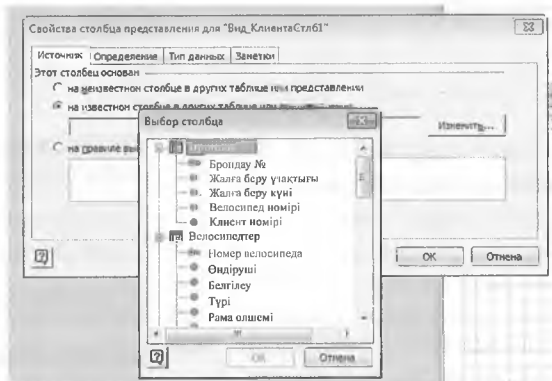


Бағандар бөлімінде жаңа бағанды енгізеді. Қосқаннан кейін жаңа бағанның сипаттары Редакциялау бөлімінде орнатылады.

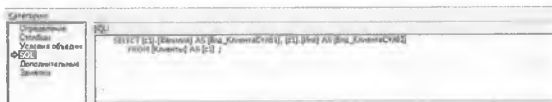


Дереккөзді басқа көстедегі Белгілі бағанды таңдап, немесе Таңдау терезесінде Көру және Өзгерту бағанын таңдау арқылы ашуға болады.

Мұнда, мысалы, Тегі параметрін таңдауға болады. ОК арқылы екі рет растау кезінде баған көрініске көшіріледі. Осылайша, баған артынан баған қосуға болады.



SQL санатында View_Kunden клиенттерінің арасындағы байланыс тағы да SQL-код түрінде көрсетіледі.



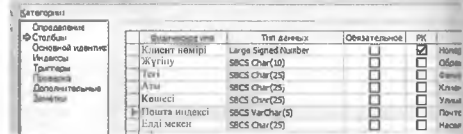
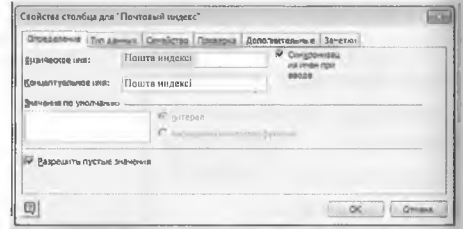
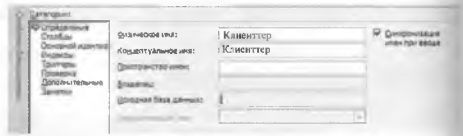
4.2.8 Өрістерді тексеру шарттарын жасау

Өрістерді тексеру шарттары бағанға енгізілген деректер мәндердің белгілі бір ауқымында болуына кепілдік береді. Мысалы, пошта индексінің ұзындығы 5 сан екенін анықтауға болады.

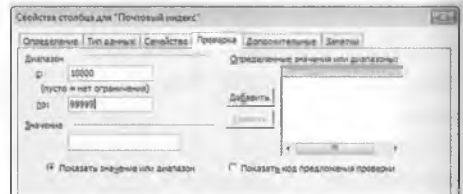
Кестені екі рет шертсеңіз (мысалы, Клиенттер), деректер қорының сипаттар диалогтік терезесін ашады.

Содан кейін Бағандар санынатында Индекс бағанын басыңыз.

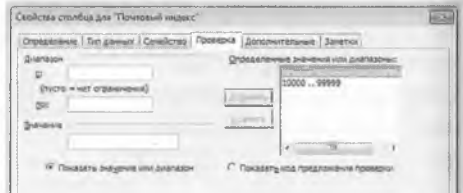
Содан кейін Өзгерту батырмасын басып, Пошта индексі үшін Бағанның сипаттары терезесі ашылады. Талап етілетін шектеу Тексеру қосымшасында енгізіледі.



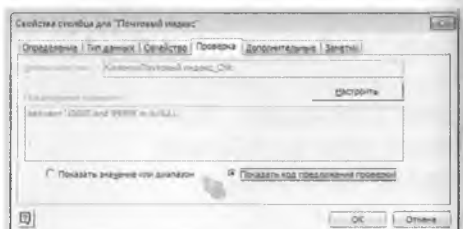
Пошта индексі 10000-9999999 аралығында болуы керек.



Мәндерді енгізгеннен кейін және Қосу батырмасын басқаннан кейін аймақ қосылады.

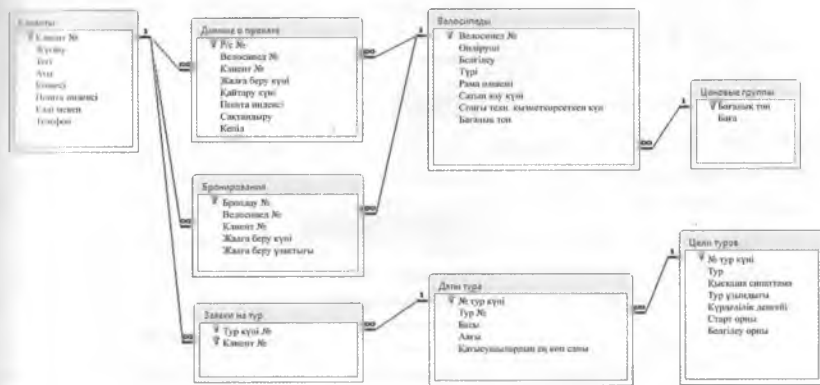


Нәтижелерді тексеру өрістерді тексеру шарттары үшін кодпен орындалуы мүмкін. Мұнда 10000-нан 99999-ға дейін немесе НӨЛ қосылды.



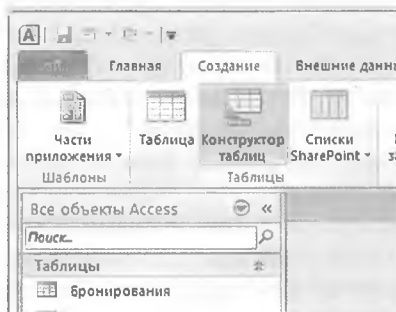
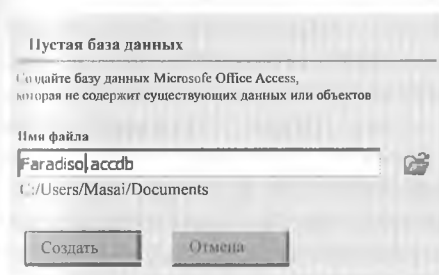
5 Access ортасында деректер қорын әзірлеу

Access деректер қорын басқару жүйесі арқылы Faradiso велосипедін жалға беруді басқару үшін деректер қоры біртіндеп жасалады. Бұл рәтте жалға беру және брондау операциялары тіркеледі. Сондай-ақ жоспарланған велосипед сапарларының тіркелімдерін басқаруға болады. Осылайша, кеңейтілген ER-деректер қорының моделі келесі құрылымды алуы тиіс:



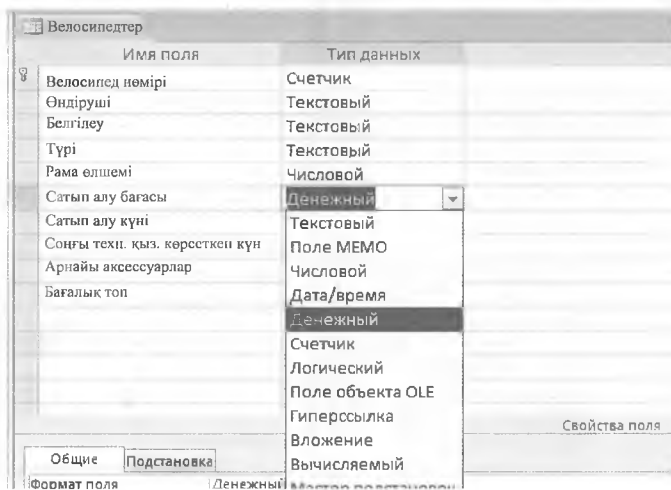
5.1 КЕСТЕЛЕР ҚҰРУ

Access-ті іске қосқан кезде қандай деректер қорымен жұмыс істеу керек деген сұранысы бар терезе ашылады. Жаңа деректер қорын жасау үшін Бос деректер қоры тармағын таңдаңыз. Қажетті сақтау орны мен Paradiso атауын енгізгеннен кейін жаңа деректер қоры Құру батырмасын басу арқылы жасалады.



Деректер реляциялық деректер қорындағы кестелерде сақталады. Деректерді енгізер алдында осы кестелер үшін олардың бағандары мен деректер түрлерінің параметрлері қойылады.

Құру Жаңа қосымшасын таңдағаннан кейін және Кесте макеті ба-тырмасын басқаннан кейін жаңа кестенің конструктор терезесі ашы-лады.



Кесте өрістерін анықтау үшін өрістің аты бағанасында өрістің атау-ын енгізіп, өрістің деректер түрі бағанасында деректер түрін орнатыңыз. Оң жақ бағанда өрістің сипаттамасын көрсетуге болады.

Деректер түрлерін орнату

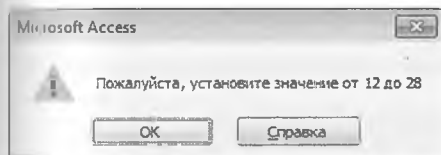
Әрбір кесте бағанына деректер түрін көрсету қажет.

Мысалы, велосипед нөмірі өрісі үшін деректер түрін автоматты түрде таңдауға болады. Жазбаларды енгізгенде, жазба нөмірі автоматты түрде бір рет ұлғайтылады. Өндіруші, Марка және Тип өрістері үшін Мәтін деректер типі таңдалып, өрістің өлшемі белгіленеді, мысалы, 20 таңба. Өрістің мазмұны үшін раманың өлшемі деректер түрі – Сан, ал Байт өрісінің өлшемі (0-ден 255-ке дейін бүтін сан). Сатып алу бағасы өрісінде деректер түрі ретінде валюта, ал валюта бірлігі ретінде еуро беріледі. Өрістерге сатып алу күні және соңғы техникалық қызмет керсету күні/уақыты деректер түрлері тағайындалады. Деректер өрісі үшін арнайы аксессуарлар, Мемо деректер түрі ұзақ сипаттамаларды жазу үшін орнатылады.

Токсеру ережесі

Сандық өріске қате енгізуі үшін ережелерді орнатуға болады. Егер енгізілген мән берілген ережеге сәйкес келмесе, экранда хабар мәтіні бір терезе пайда болады.

Мысалы, жақтау елшемі өрісі үшін 12-ден 28 дюймге дейінгі аймақты анықтайды. 36 мәнін енгізгенде, көрсетілген хабар көрсетіледі.



Общие	Ссылки
Размер поля	Длинное целое
Формат поля	00
Число десятичных знаков	Автоматически
Маска ввода	
Подпись	
Значение по умолчанию	0
Условие на значение	>=12 And <=28
Сообщение об ошибке	Пожалуйста, установите значение от 12 до 28
Обязательное поле	Нет
Индексированное поле	Нет
Смарт-теги	
Выравнивание текста	По умолчанию

Рама өлшемі өрісінде сандық мәнді «дюйм» белгісімен толықтыру үшін пішім анықталады. 00 саны үшін екі толтырғыштан кейін бойнеленетін таңбалар (мұнда таңбалар ") тырнақшаға алынады.

Стандартты мәндерді енгізгенде, мысалы, 0, бағдарлама автоматты түрде жаңа деректер жазбасын енгізгенде мәндерді қоюды ұсынады. Бұл көп жағдайда қажетті мәндерді енгізуді жеңілдетеді.

Бастапқы кілт

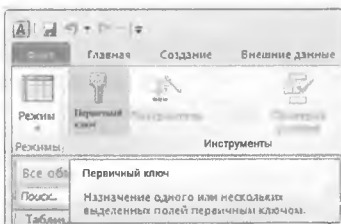
Жеке велосипедтерді бірдей сөйкестендіру үшін бастапқы кілт алаңы ретінде әрбір жазбада мазмұны өзгеше велосипед нөмірі өрісі орнатылады. Жолаққа басқаннан кейін құралдар тақтасындағы велосипед нөмірі бастапқы кілт батырмасы басылады.

Өріс аты алдында кілт белгішесі пайда болады.

Бірнеше өрістен тұратын бастапқы кілт басқару пернесін басқан кезде тиісті жолдарды бөлу және бастапқы кілт батырмасын басқан кезде анықталады.

Кестені сақтау үшін құралдар тақтасындағы диск белгішесін басыңыз. Аты ретінде, мысалы, велосипедтің мәні енгізіледі және Сақтауға басу арқылы расталады.

Деректерді енгізу үшін кестені екі рет шертіп немесе Қарау батырмасын басып, кестені қараңыз. Деректерді енгізгенде, тиісті жолда қауырсын белгісі пайда болады.



Жазбадан шыққан кезде қауырсын жоғалады және жазба автоматты түрде сақталады.

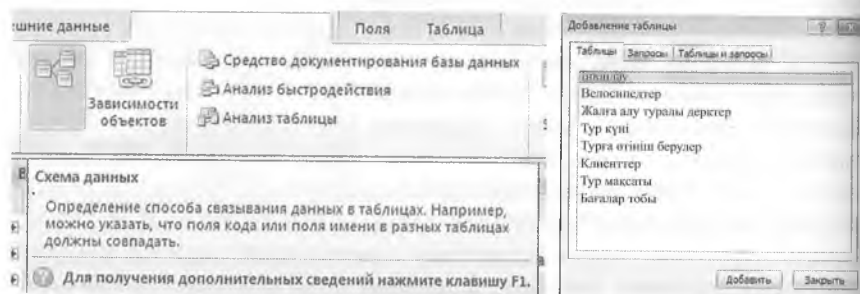
Ескерту:

Деректерді жоғалтуды болдырмау үшін жазбадан шыққаннан кейін деректер дереу қатты дискіде сақталады.

Мәтіндік редакторларда сияқты деректерді енгізуді болдырмау функциясы жоқ.

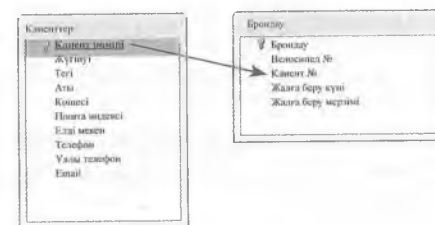
5.2 БАЙЛАНЫС ЖӘНЕ СІЛТЕМЕЛІК ТҰТАСТЫҚТЫ АНЫҚТАУ

Байланыс әртүрлі кесте өрістерінің қатынасын анықтайды. Access ортасында байланыстар графикалық құралдардың көмегімен жасалады. Мысалы, Резервтеу кестесіндегі клиенттердің нөмірлері Клиенттер кестесінің нөмірлерімен байланысты екенін көрсету үшін. Ол үшін мәзір жолағында деректер қорының құралдары қосымшасын таңдап, байланыс батырмасын басыңыз. Кестелерді таңдау үшін Кестені көрсету терезесі ашылады.

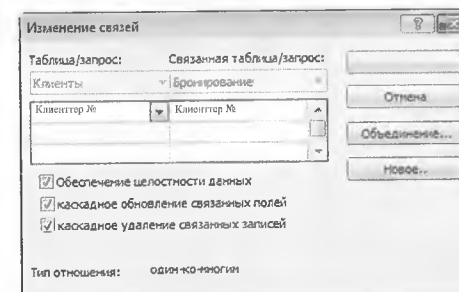


Барлық көрсетілген кестелерді Shift пернесін басып тұрып, тізімдегі бірінші және соңғы кестені басу арқылы таңдап, Қосу пернесін басыңыз. Кестені көрсету терезесі жабық болады.

Байланыс орнату үшін клиент нөмірін клиенттер кестесінде басыңыз және тінтуірдің батырмасын басқанда, клиент нөмірін резервтеу кестесі арқылы апарыңыз.



Байланыс қасиеттерін өңдеу терезесі ашылады.



Сілтеменің бүтіндігін ескеру деректер қоры жүйесін жазбалар арасындағы логикалық қарым-қатынасты тексеруге мәжбүрлейді. Клиенттер кестесіне әлі қосылмаған клиенттің бастапқы кілті Резервтеу кестесіне енгізілуі мүмкін емес. Керісінше, басқа кестедегі деректер жазбаларына сілтеме жасайтын клиенттің деректер жазбасын, мысалы, Резервтеу кестесінде байланысқан деректер жазбаларын жоймай өшіру мүмкін емес.

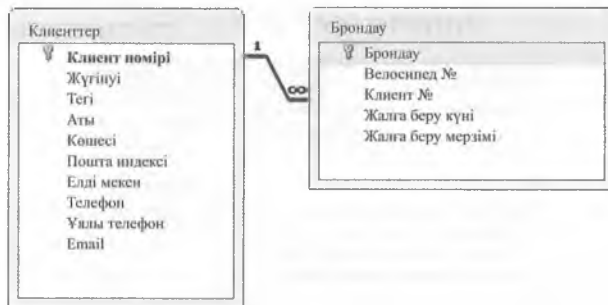
Ескерту:

Сілтемелік тұтастық кестелер арасындағы қатынастардың логикалық тұрғыдан дұрыс болуын және мәліметтер массивінің болмауын қамтамасыз етеді.

Деректерді (мысалы, бұрынғы клиенттер) жою мүмкіндігі болу үшін байланысқан деректер жазбаларына тасымалдау жалаушасын қойыңыз. Егер жазба кестеден алынып тасталса, оның бастапқы кілтінің өрісі, мысалы, клиенттің нөмірі басқа жазбаға сілтеме жасайды, мысалы, клиенттің сол нөмірі бар басқа кестелердегі барлық жазбалар да жойылады. Осылайша, жағдайлардың алдын алады, яғни бастапқы жазбаларды жою кезінде «жетімсіреп қалған» жазбалар қалады, соның нәтижесінде жою ауытқулары туындауы мүмкін.

Өзара байланысты өрістерге Жаңартуды көшіруді іске қосу – сілтеме өрісін, мысалы, Клиенттер кестесіндегі клиент нөмірін өзгерткенде, осы өріске сілтеме жасалған кестенің басқа өрістерінің мазмұны да өзгередінін білдіреді.

Егер Құру батырмасын бассаңыз, екі кесте арасындағы 1: n байланысы 1 және «байланыс» терезесінде сызық пен 1 және ∞ символдары арқылы бейнеленеді.

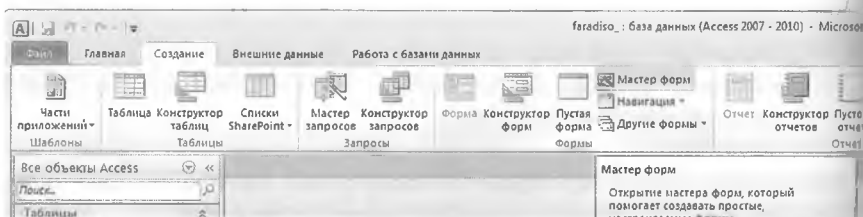


5.3 НЫСАНДАР

Деректерді енгізу үшін нысандар қарастырылған. Олар деректерді енгізу үшін де, көру, өңдеу және шолу жасау үшін де қолданылады.

5.3.1 Нысан жасау

Жаңа нысан жасау үшін Құру қосымшасына өтіп, нысан бөліміндегі Нысан жасау шеберін таңдаңыз.



Келесі диалогтық терезеде нысан керінісі көрсетіледі. Велосипедтер кестесі үшін бір нысан мысалында:

Создание форм

Выберите поля для формы.

Допускается выбор нескольких таблиц или запросов.

Таблицы и запросы

Таблица: Велосипедтер

Доступные поля:

- Велосипед №
- Опдіруші
- Белгіленуі
- Түрі
- Сатып алу бағасы
- Сатып алу күні
- Соңғы тех. көрсеткіші

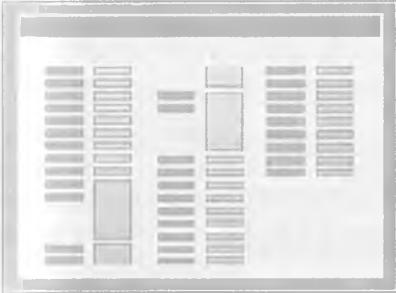
Выбранные поля:

Отмена < Назад Далее > Готово

Велосипедтер кестесін таңдағаннан кейін **>>** батырмасы нысан үшін кестенің барлық өрістерін тасымалдайды. Батырманы басқаннан кейін келесі **>** нысан макеті орнатылады, мысалы, бір бағанды.

Создание форм

Выберите внешний вид формы:



В один столбец

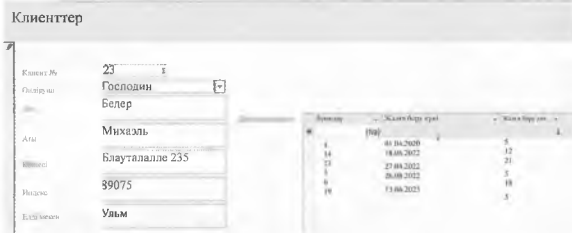
Аяқтау батырмасын басқаннан кейін, нысан жасау процесі аяқталады. Ол, мысалы, деректерді таңдау немесе өзгерту үшін пайдаланылады.



5.3.2 Бағынышты нысандар

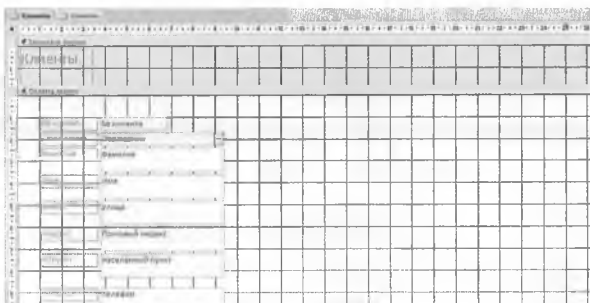
Бағынышты нысандар – 1:n кестелерді нысанда көрсетуге арналған құралдар. Негізгі нысан бастапқы қатынас кілті өрісін қамтитын жазбаны көрсетсе, ал бағынышты нысан жазбалардың арасындағы байланысты егжей-тегжейлі көрсетеді.

Бұдан әрі бағынышты нысан бар Клиенттер нысаны жасалады, онда тиісті көрсетілген клиент үшін байланысқан резервтеу көрсетіледі.



Бағынышты нысан ретінде алдымен резервтеу жазбаларын көрсете алатын нысан жасалады. Ол үшін Құру қосымшасында нысан жасау шебері батырмасын басыңыз. Деректер көзі ретінде резервтеу кестесі орнатылған, ал >> батырмасы арқылы барлық өрістер қосылады. Нысандар, мысалы, қосалқы Резервтеудің__бағынышты__нысаны атауымен сақталады.

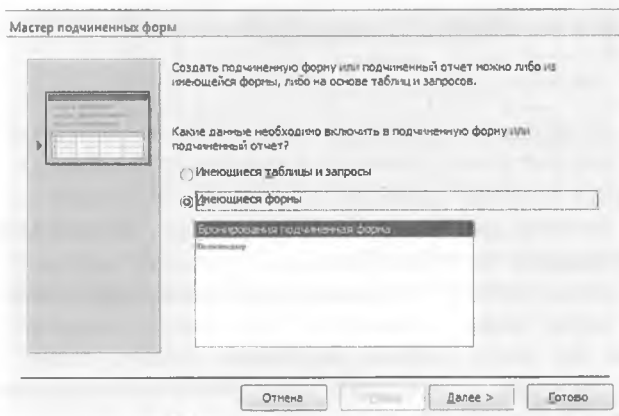
Енді бір бағаны бар клиенттер пішін шебері арқылы жасалады. Бағынышты нысан үшін жеткілікті орын табу үшін нысан бөлшегі тиісінше ұлғайтылуы тиіс. Оны Конструктор режимінде детализация аймақтың оң жақ шетін басып, тінтуірдің көмегімен оңға қарай апаруға болады.



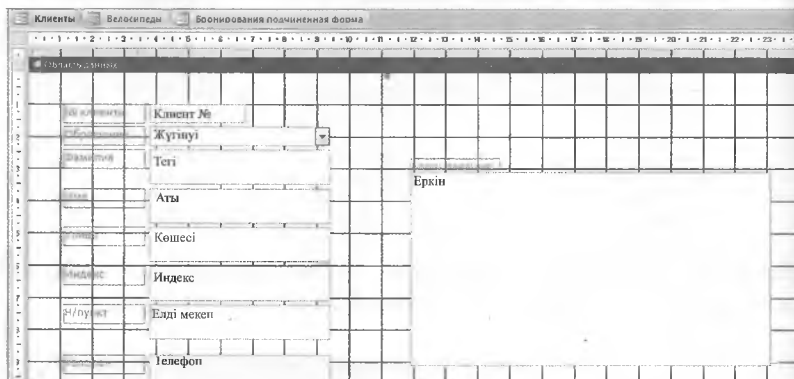
Нысанды жобалау құралдары қосымшасында – «Дизайн» қосымшасында «бағынышты нысан» батырмасы таңдалады.



Енді пішіннің еркін кеңістігінде қолайлы үлкен тік төртбұрыш салынады. Бағынышты пішінді таңдауға мүмкіндік беретін шебер (конструктор) іске қосылады.



Нысанды сақтағаннан кейін әрбір клиент туралы ақпарат тиісті Резервтеумен көрсетіледі. Конструктор келесідей:



5.3.3 Батырмалар арқылы қорын басқару

Нысандағы батырмаларды пайдалана отырып, Пайдаланушы деректер қорының қосымшаларының әртүрлі жұмыс аймақтарына ауыса алады.

Мысал

Paradiso велосипедтерін жалға алу өрісімен және Велосипедтер, Клиенттер және Резервтеу батырмаларымен Орталық бөлім нысаны жасалады. Дайын нысан мысалы былай болады:

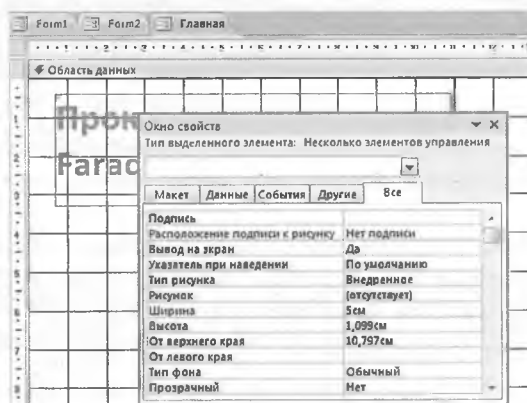


Әртүрлі кестелердің еңдеуін командалық батырмалардың көмегімен басқару мүмкіндігі бар.

Алдымен нақты кестеге байланысты емес жаңа нысан жасалады. Тақырып жасау үшін «Макетті басқару элементтері» көрінісінде Аа белгі өрісінің белгішесін басып, содан кейін тінтуірдің көмегімен пішін мәліметтерінің айналасында жиектерді апарып, тақырыпты енгізіңіз.




Тінтуірдің оң жақ батырмасын басқанда, ол Ерекшеліктер диалогтік терезесін таңдауға болатын мәзірді ашады. Мұнда, мысалы, өрістің ені мен биіктігін және қаріп қасиеттерін анықтауға болады.

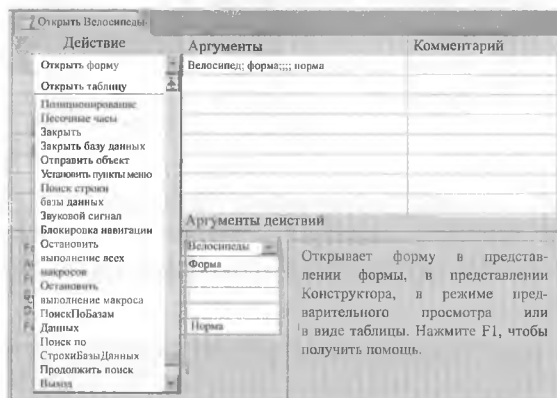


Командалардың кнопкалары нысан макетінің құралдары – «Дизайн» қосымшасындағы командалық батырмасын басу және нысандағы батырманың өлшемін өзгерту жолымен нысанға салынады.

Мысалы, пәрменді батырмалар белгілі бір әрекеттерді орындау үшін макростарда сақталады.

5.4 МАКРОСТАР

Макростар, мысалы, орталық нысандағы Велосипедтерге басу арқылы велосипедтің нысанын ашу және бір мезгілде орталық нысанды жабу үшін жасалады. Егер «Деректер қоры» терезесінде Құру қосымшасын басып, Макрос батырмасын бассаңыз, макростарды жасау  терезесі ашылады.



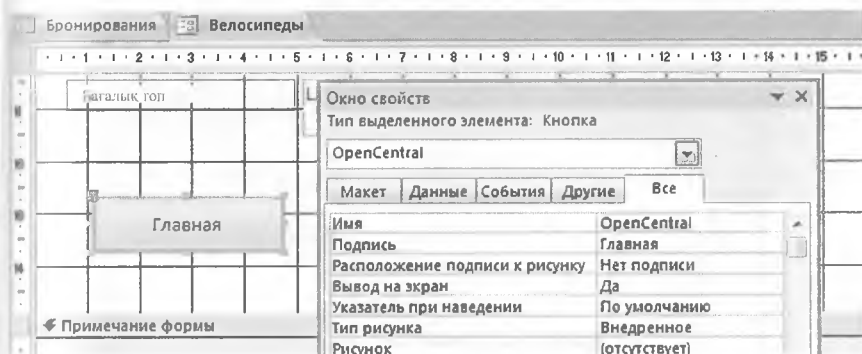
Ол үшін Әрекет ерісінде нысанды ашу әрекетін таңдаңыз. Енді Облыста Велосипедтер нысаны мен нысан көрінісі сияқты аргументтер бар.

Орталық нысан макростың екінші әрекеті арқылы жабылады, мысалы, Жабу және нысанның формасы мен Нысанның атауы Орталық нысан үшін аргументтер қойылады. Макрос терезесін жабыңыз және оны Қолжетімді велосипедтер сияқты маңызды атаумен сақтаңыз.

Ескерту:

Маңызды нысан атаулары мәліметтер қорын басқаруды жеңілдетеді.

Енді OpenWheels макросы орталық нысандағы батырмамен байланысты. Ол үшін орталық нысанды дизайн режимінде ашыңыз және Велосипедтер командасын басу арқылы қасиеттер бетін ашыңыз. Оқиға қосымшасында басылған кезде Ашық оқиға үшін Ашық велосипедтер макросы орнатылады. Егер сіз осы командалық батырманы орталық нысанда шертсеңіз, онда макрос және онда жазылған әрекеттер орындалады.



Ол үшін Велосипедтерді ашу макросымен бірдей етіп, OpenCentral макросын жасаңыз. Орындалатын бірінші әрекет – ағымдағы нысанды жабу үшін ешқандай қосымша аргументтерсіз Жабу. Екінші әрекет «Нысанды ашу» орталық және нысан амалдарының аргументтерімен орнатылады. Пәрмендік батырма «Велосипедтер» нысанының дизайнерінде орналасқан және екі рет басудан туындаған сәйкес қасиеттермен жабдықталған. Тақырыптың орнына сіз кескін орната аласыз (мысалы, ақпараттық нүктелік кескін гуралы ақпарат). Сипатты басқан кезде, Орталық нысанды ашу макросы орнатылады.

5.5 ЕСЕП ҚҰРУ

Есептер басып шығару үшін деректерді еңдеуге арналған. Мысалы, каталогтар немесе шоттар бір есепте жасалады және басып шығарылады. Бір есепте аралық сомалармен және басқа да есептермен ауқымды деректер тобын оңай жүзеге асыруға болады.

Есептерді **Есеп шебері** арқылы жасауға болады.

Мысал

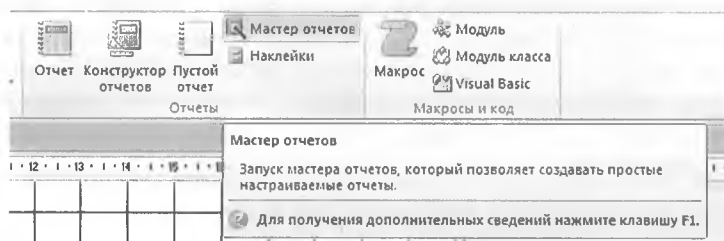
Деректер қорында сақталатын әрбір клиентті резервтеу туралы есеп құрылады. Тегі мен аты, брондау нөмірі, велосипед туралы ақпарат, жалға беру күні мен жоспарланған мерзімі көрсетілуі тиіс. Әрбір клиенттен кейін жалдау кезеңінің барлық күндері үшін сомасы көрсетілуі тиіс.

Есеп келесідей болуы мүмкін:

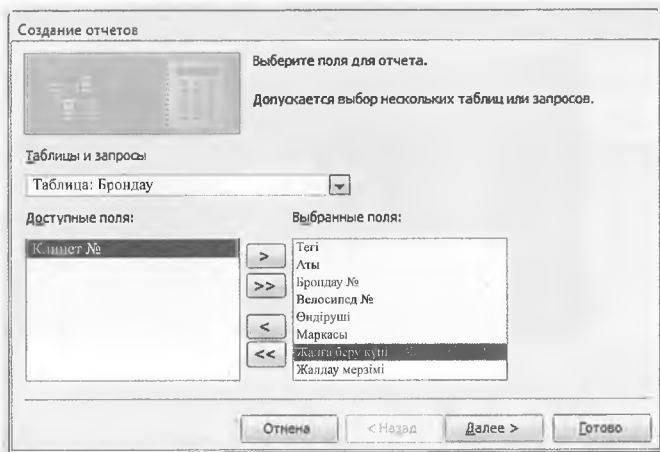
Аты	Аты	Номер/№	Даты/дата	Сыртқы	Матрица	Жалп. бер. күні	Жалп. мерзімі
Сарыбай	Матри		1.1.	ИИШ	4014	23.08.19	8
			1.2.	ИИШ	4014	30.12.18	13

23.08.2019. Стр. 1 из 1

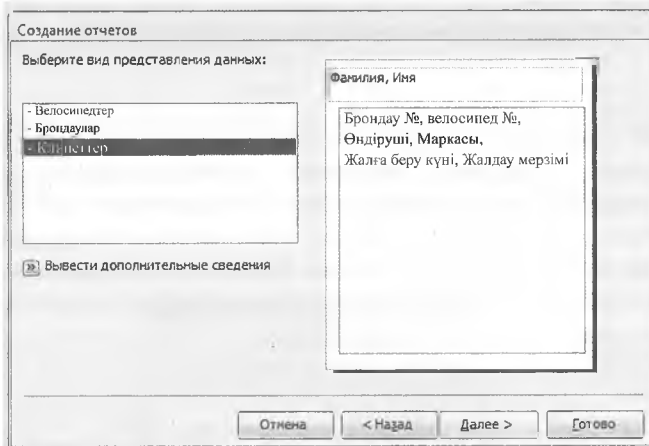
Құру қосымшасын таңдағаннан кейін, Таспадағы есеп аймағында Есеп шебері батырмасын басыңыз.



Келесі терезеде есепке енгізілетін өрістерді анықтауға болады. Клиенттің тегі мен атын қабылдау үшін Кесте / Сұраныстар терезесінде Кесте параметрін таңдаңыз: Клиенттер және тиісті өрістер, > батырмасын басу арқылы суреттің оң жағына жылжыту.



Резервтеу кестесінен рез. №, велосипед №, Жалға беру күні және Жалға беру мерзімі өрістері алынады, ал Өндіруші және Белгі Велосипедтер кестесінен алынады. Әрі қарай батырмасын басқаннан кейін деректер жазбаларын топтастыру жүргізіледі. Бір клиентке қатысты барлық жазбаларды көру үшін клиенттер бойынша жазбаларды топтап, растау үшін Әрі қарай батырмасын > басыңыз.



Әрбір клиент үшін барлық жалдау мерзімінің құнын көру үшін келесі терезеден Жиынтық параметрлері пәрменін таңдап, Сона жалаушасын және ОК батырмасын басыңыз.

Дайын түймесін басқаннан кейін есеп көрсетіледі.

5.6 ДЕРЕКТЕР ҚОРЫНА СҰРАНЫС ЖАСАУ

Сұраныстар деректерді бағалау және өңдеу үшін қолданылады. Мысалы, белгілі бір өлшемдер бойынша кестелерден іздеуге болады.

Мысал

Жалға беру деректерімен және жалға берілген велосипедтердің деректерімен 2019 жылғы барлық клиенттерді көрсететін Жалға беру_2019 сұрауын әзірлеу қажет.

Тегі	АТЫ	Жалға алу күні	Қайтару күні	Велосипед номері	Маркасы
Аржанов	Михаил	15.06.18	18.02.19	2	Strails
Саржанов	Талгат	18.10.18	19.12.18	1	A25

Access ортасының мәзірінің таспасындағы Сұраныстар бөлімінде жасау қосымшасын таңдағаннан кейін Кестені көрсету терезесінде таңдау үшін редакциялау үшін Кесте сұраныстары конструкторы батырмасы таңдалады.

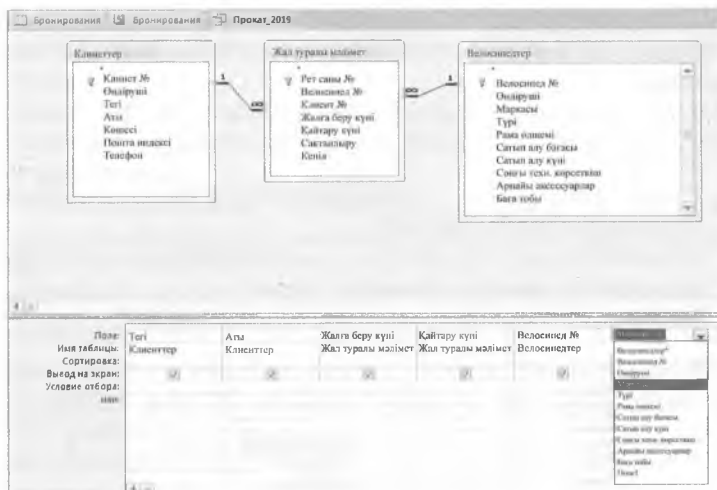
Добавление таблицы

Таблицы | Запросы | Таблицы и запросы

- Брондау
- Велосипедтер
- Жалға алу туралы деректер
- Тур күні
- Турға өтініш берулер
- Клиенттер**
- Тур мақсаты
- Бағалар тобы

Добавить | Закрыть

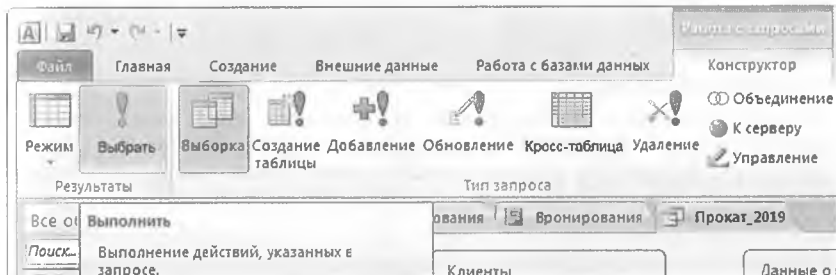
Басқару пернесін басқан кезде Клиенттер кестесі, Жалға беру деректері және Велосипедтер таңдалады. Қосу арқылы расталғаннан кейін сұраныс конструкторының көрінісі ашылады.



Тегі, аты, жалға беру күні сияқты қалаған өрістер ашылатын мәзір немесе тінтуірдің екі рет басылуы арқылы бөлек бағандарда таңдалады.

2019 жылы жалға беру сұранымын шектеу үшін жалға беру күні бағанында критерий ретінде #01.01.2019 бастап #31.12.2019 дейінгі мән енгізіледі. Access-та күн мәндері #таңбаларының арасында орнатылады.

Сұраныс сақталғаннан кейін (мысалы, Verleihe_2019 деп аталатын) батырмасын пайдалануға болады! Іске қосу орындалады.



Ескерту:

Сақталған сұраныстар сұраныс нәтижесі емес, SQL-сұрауды орындау нұсқаулығын қамтиды.

Шығаруды сұрыптау:

Жазбаларды баған бойынша сұрыптау үшін сұраныс конструкторы режимінде тиісті жолда өсу немесе кему бойынша сұрыптау тәртібі қойылады.

Мысалы

Жалға Беру_2019 сұранысы жалға беру күні бойынша өсуі бойынша сұрыпталуы тиіс.

Ол үшін Жалға берілген күн бағанындағы сұрыптау өрісінің ашылмалы мәзірін басыңыз және өсу бойынша дегенін таңдаңыз.

Поле:	Тегі	Аты	Жалға беру күні	Қайтару күні	Велосипед №	Маркасы
Ияя таблиця:	Клиенттер	Клиенттер	Жал туралы мәлімет	Жал туралы мәлімет	Велосипедтер	Велосипедтер
Сортировка:						
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:						
или:						

Содан кейін, сұраныс орындалғаннан кейін деректер қорының жолдары жалға беру күнінің өсу ретімен көрсетіледі:

Тегі	Жалдау	Тапсыру күні	Қайтару күні	Велосипед №	Маркасы
Арканов	Михаил	15.06.18	18.02.19	2	Strails
Саржанов	Талгат	18.10.18	19.12.18	1	A25
*					

5.7 5-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР

1-тапсырма

Жобаларды басқару

Компаниялардың бірінде деректер қорының көмегімен болашақта функцияларды кеңейту мүмкіндігімен жобаларды басқарудың шектеулі жүйесін құру қажет.

Мынадай шарттар сақталуы тиіс:

- Жүйеге әрбір қызметкердің қол жетімділігі құпия сөз арқылы жүзеге асырылуы тиіс. Құпия сөз қызметкерлер кестесінде сақталады.
- Әрбір жоба үшін аты-жөнін, сипаттамасын, басталу және аяқталу күнін, сондай-ақ жоба жетекшісінің атын сақтау қажет.
- Әрбір жоба үшін жобаның жетекшісі ретінде бір қызметкер таңдалады.
- Әрбір қызметкер үшін жұмыс уақытының ұзақтығы (сағат ұзақтығы) және нақты жоба үшін белгілі бір күні орындаған жұмыстың сипаттамасы жазылады.

а) Кесте байланыстарының графикалық көрінісін ER-диаграмма арқылы құрастырыңыз және кестелер арасындағы байланыс түрін көрсетіңіз.

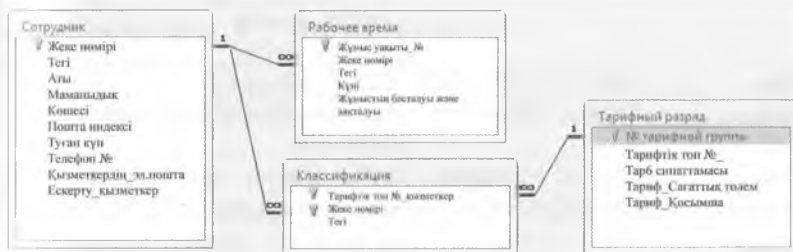
Қажетті төлсипаттары бар кестелерді 3-қалыпты пішінде жасаңыз. Әрбір кесте мен байланыстар үшін бастапқы кілттерді жасаңыз. Байланыс деректерді енгізу кезінде сілтеме тұтастығына тексерілуі тиіс. Сонымен қатар толық жазбаларды жою және жаңарту кеңейтілуі мүмкін.

Сілтеме тұтастығы не үшін қажет?

б) жоба деректерін енгізу үшін пішін жасаңыз.

2-тапсырма

Компаниядағы қызметкерлердің еңбекақысын есептеу үшін төменде келтірілген кеңейтілген ER-диаграммасы бар деректер қоры пайдаланылады.



- Access-те «мөн-байланыс» типінің диаграммасын іске асырыңыз.
- бұл жоба бірінші, екінші және үшінші қалыпты формадағы қалыптау ережелерін қандай уақытта бұзады? Оларды қалай болдырмау керектігін ұсыныңыз.
- қалыптандыру ережелерін бұзбау не үшін орынды және маңызды?

6 SQL деректер қорының тілі

SQL (Structured Query Language = сұраныстардың құрылымдық тілі) – бұл деректер қорының реляциялық деректер қорында құрылымдарын анықтау үшін, сондай-ақ өңдеу (кірістіру, өзгерту, жою) және деректер қоймаларына сұраныс жасауға арналған деректер қорының тілі.

6.1 SQL СТАНДАРТТАРЫ

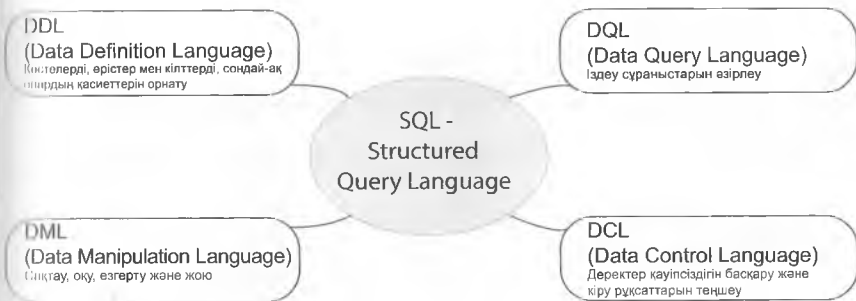
Ең алдымен SQL реляциялық деректер қорының соңғы пайдаланушысының тілі болды. Мұнда кең таралған графикалық интерфейстер ауыстырылды. Бірақ SQL көптеген графикалық жүйелерге қарағанда анағұрлым қуатты болғандықтан, ол деректер қорының әзірлеушілері үшін маңызды бағдарламалық қамтамасыз ету болып табылады. Ол сондай-ақ Интерфейс тілі ретінде және басқа деректер қорының жүйелеріне қол жеткізу тілі ретінде үлкен мәнге ие, өйткені ол іс жүзінде операциялық жүйеге, жұмыс режиміне және интерфейске байланысты емес.

Түрлі стандарттар бар:

Шыққан жылы	Норма	Стандарт
1986 ж.	ANSI-SQL1	SQL-1 стандарты
1989 ж.	ISO SQL-1	SQL-1 тұтастықтың кеңейімінен
1992 ж.	ISO SQL-92	SQL-2 стандарты, үш кезең: – Бастапқы деңгей – Орта деңгей – Тұтас деңгей
1999 ж.	ISO / IEC 9075: 1999	SQL-3 стандарты
2003 ж.	ISO/IEC 9075:2003	XML-мен (белгілеудің кеңейтілген тілі) байланыс
2007 ж.	ISO/IEC 9075:2007	XML деректер түрін кеңейту
2011 ж.	ISO/IEC 9075:2011	Ағымдағы нұсқа

Қазіргі уақытта деректер қоры жүйелерінде, ең алдымен, SQL-2 стандарты іске асырылды, ал жаңа стандарттар тек жеке жүйелерде ғана іске асырылады.

SQL тілі аз командалардан тұрады, бірақ оларды әртүрлі тәсілдермен қолдануға болады. Сонымен қатар, ол кесте мен өрісті анықтау үшін де, сұрау және деректерді өзгерту үшін де қызмет етеді.



6.2 КЕСТЕЛЕРДІ ҚҰРУ, ӨЗГЕРТУ ЖӘНЕ ЖОЮ

Кестелерді құру

CREATE TABLE командасы кестенің аты деректер қорының кестесін жасайды. Жалпы синтаксис:

```
CREATE TABLE кесте аты (
  Spalte_1 Datentyp_für_Spalte_1,
  Spalte_2 Datentyp_für_Spalte_2,
  ...);
```

Мысал

Клиенттер жаңа кестесін құру.

Шешім:

```
CREATE TABLE Клиенттер (
  Клиент № INT NOT NULL auto_increment,
  тері АТЫ(45) NOT NULL,
  АТЫ(45) NULL ,
  көше АТЫ(45) NULL ,
  Пошта индексі VARCHAR (6)
  NULL, телефон VARCHAR(25) NULL ,
  );
```

Ескерту:

CREATE командасымен сондай-ақ жаңа деректер қорын жасауға болады, мысалы, CREATE Database paradiso_neu; paradiso_neu атымен деректер қорын жасайды.

Кестені өзгерту

ALTER командасы

ALTER TABLE кесте аты командасы сипаттарын немесе бағандарын өзгертеді.

Жалпы синтаксис:

```
ALTER TABLE кесте аты  
[ Өзгерістер Ерекшелігі]
```

Ерекшеліктерді өзгерту талап етілетін өзгерістердің сипатына байланысты, мысалы:

- Баған қосу:

```
ADD бағаны_1 деректер түрі_бағандар үшін_1
```
- Бағанды алып тастау:

```
DROP баған_1
```
- Бағанның атын өзгерту :

```
CHANGE бағанның_ескі аты бағанның жаңа аты
```

Деректер түрі жаңа_баған_аты үшін
- Бағанның деректер түрін өзгерту:

```
MODIFY Баған_1 жаңа деректер түрі
```

Мысал

Клиенттер кестесінде Email бағаны қосылады.

Шешім:

```
ALTER TABLE Клиенттер  
ADD Email VARCHAR(30);
```

Кестені жою

DROP TABLE Кесте аты командасы кестені жояды.

Мысал

Мысал

Велосипедтер_ескі кестесін жою қажет.

Шешім:

```
DROP TABLE Велосипедтер_ескі;
```

Ескерту:

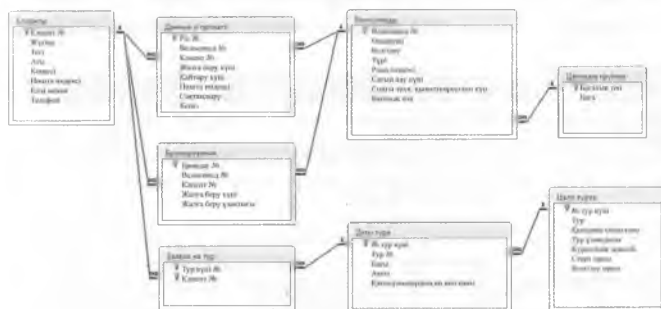
Қашықтағы кесте енді деректер тұтастығының шарттарына байланысты undo командасы арқылы қалпына келтірілуі мүмкін емес.

6.3 SELECT АРҚЫЛЫ ТАҢДАУ СҰРАНЫСТАРЫ

Таңдау сұраныстары үшін SELECT командасы бар. Бұл команда әмбебап ететін кең синтаксиске ие.

Ескертулер:

Келесі мысалдар велосипедтерді жалға беруді көрсететін «Paradiso» деректер қорына жатады



Клиенттер мен велосипедтердің негізгі деректеріне қосымша ретінде ол жалға алу деректерін де, брондау деректерін де қамтиды. Сонымен қатар велосипед турлары клиенттер тіркелуі мүмкін әртүрлі бағыттар бойынша ұсынылады.

0.3.1 Шарттары бар таңдауға сұраныстарды шектеу

Мысал

5-тен аз баға тобы бар барлық велосипедтерді таңдау үшін SQL-сұраныс жасау.

Шешім:

```
SELECT велосипед нөмірі, өндіруші, атауы,  
баға тобы  
FROM Велосипедтер  
WHERE баға тобы < 5;
```

SELECT кілт сөзінен кейін, мысалы, велосипед нөмірі, өндіруші, атауы, баға тобы үтірлермен бөлінген көрсету бағандары көрсетіледі. Қорытынды SELECT операторынан кейін орындалады. Кестелердегі өрістер атауларының жазылуына назар аударыңыз, әйтпесе олар танылмайды. Сондай-ақ «-» дефис сияқты кейбір арнайы таңбалар математикалық «минус» операторы ретінде түсіндіріледі.

Ескерту:

Арнайы таңбалар мен Бос орындар кестелер мен өрістердің аттарында қателерді тудырады.

* Таңбаны барлық кесте өрістері үшін толтырғыш ретінде пайдалануға болады, мысалы, нұсқауды көрсетеді

```
SELECT * from велосипедтер;  
велосипед кестесіндегі барлық жазбалар.
```

Жұмыс істеу қажет кесте, мысалы, велосипедтер from(= с, по) кілт сөзінен кейін көрсетіледі. Егер бірнеше кесте қолданылса, олар үтірмен белінеді. Where (= қайда) сақталған сөзден кейін көрсетілетін жазбаларды таңдауды шектейтін шарт пайда болады. Әдетте ол салыстыру операторы және салыстыру мәні бар бағанның атауынан тұрады.

Оператор	Мысал	Мәні, нәтижесі
=	Клиент № = 24	дәл сол
<	Баға тобы < 5	...қарағанда
>		...қарағанда үлкенірек
<>		тең емес
<=		Кіші немесе тең
>=		Үлкен немесе тең
LIKE	E/мекен LIKE 'U%'	Орналасқан жер ерісін мәтіндік үлгімен салыстырады; u бастапқы латын барлық елді мекендер ескеріледі.
ЖӘНЕ	ПОШТА ИНДЕКСІ = 89077 AND	Деректер қорының жолы егер 1-шарт және 2-шарт орындалған жағдайда ғана ескеріледі
	№ > 200 OR № < 100	Жазба 1-шарт немесе 2-шарт орындалса ескеріледі.
NOT	NOT (ПОШТА ИНДЕКСІ = 89077)	Шарт орындалмаса, жазба ескеріледі

Ескерту:

«WHERE» шарттары әрқашан логикалық салыстырудан тұруы керек, оған «ақиқат» немесе «жалған» деп жауап беруге болады.

Мысалда баға тобы < 5 шарты қолданылады. Кестедегі әрбір жазу үшін велосипедтер, деректер қорын басқару жүйесі ол үшін «ақиқат» немесе «жалған» шартының орындалғанын тексереді. Сұраныс нәтижесінде экранда баға тобы < 5 шынайы болып табылатын жазбалар көрсетіледі.

6.3.2 DISTINCT

Distinct (= әртүрлі) кілт сөз сұраныс жасаудағы жазбалардың дубликаттарын жасырады. Егер, мысалы, велосипед сатып алынған өндірушілердің атын ғана көрсету қажет болса, онда SELECT DISTINCT FROM Велосипедтерді таңдау кезінде әрбір тиісті өндіруші тек бір рет көрсетіледі.

6.3.3 WHERE шартындағы өріс мазмұнын көрсету

Валюта өрістерін қоса алғанда, сандық өрістердің мазмұны қосымша таңбалаусыз қолданылады,

мысалы, бағасы = 150

Мәтін өрістерінің мазмұны WHERE шарты ішіндегі тырнақшаға алынған,

мысалы, тұрғылықты жері = 'Штутгарт'.

Мәтін өрістерін мәтіндік үлгілер бойынша іздеуге болады. Jet-SQL-де (мысалы, Access) плейс-холдерлер ретінде кез келген таңбаны бөлгілеу үшін «?» сұрақ белгісі немесе бір немесе бірнеше таңбаны бөлгілеу үшін * қолданылады.

Мысал	Нәтижесі
'M??er'	'Meier', 'Maier', 'Mayer', 'Meyer'; дәл екі таңба ауыстырылады.
'M*er'	Maier, Mittersberger, Müller ... ; мазмұны 'M' деп басталатын және 'er' деп аяқталатын барлық өрістер.

Мысал

SELECT операторы тұрғылықты жері и әрпінен басталатын барлық клиенттерді шығару үшін қолданылады.

Шешімі:

Тегін, атын, көшені, пошта индексін, қаланы, клиенттен телефонды таңдаңыз.

WHERE елді мекен LIKE 'U*';

Мәтіндік өрісті мәтіндік үлгімен салыстыра отырып, LIKE операторы мәтіндік өрістің мәтіндік үлгімен салыстырылуы тиіс екенін түсініледі және осылайша, мысалда и бастапқы әрпімен барлық орындар шығарылуы тиіс.

Ескерту:

Мәтін өрістерін салыстырғанда = орнына LIKE операторын пайдаланыңыз.

Күн өрістері деректер қоры жүйелерінде түрліше енгізіледі және бейнеленеді.

Деректер қоры	Күн өрісінің пішімі
Oracle, Informix, DB2	'2020/10/03'
Access	#2020/10/03# немесе #03.10.2020#
MySQL	'2020-10-03'

Неміс және американдық емле арасында түрлендіру тәсілдері бар болса да, американдық емле дұрысырақ (жыл /ай/күн), мысалы #2020/10/04#.

6.3.4 BETWEEN операторы

BETWEEN операторы (=арасында) диапазон сұраныстарын өзірлеу үшін қолданылады. Ол мәтіндік өрістер, күн өрістері және сандық өрістер үшін пайдаланылуы мүмкін. BETWEEN төменгі және жоғарғы

шектер арасындағы деректерді таңдайды. Шекті параметрлер іріктемеге қосылады

Мысал

2018 жылы SQL-операторының көмегімен сатып алынған барлық велосипедтерді көрсету қажет.

Шешімі (Access-те):

Select № велосипед, өндіруші, белгілеу, сатып алу күні, сатып алу бағасы

From велосипедтер

WHERE сатып алу күні Between #2018/01 # And # 2018/12/31#;

Нәтижесі:

Велосипед нөмірі	Өндіруші	Белгіленуі	Алынған күні	Алу бағасы
3	Panasonic	FirstClass	17.01.2018	1 960,00 €
4	Miyata	Devant	17.03.2018	2 800,00 €
6	Scott	Executive	18.03.2018	2 350,00 €
19	Hirsch	Klettergemse	16.06.2018	2 400,00 €
24	Ivecu	Stralys	31.12.2018	1 270,00 €

6.3.5 IN операторы

IN операторы өрістің мазмұнын ықтимал мазмұн тізімімен салыстыру үшін пайдаланылады. Штутгарт агломерациясының барлық клиенттерін түсіру үшін, мысалы, келесі Шарт қолданылуы мүмкін:

```
... WHERE E/мекен IN ('Stuttgart', 'Esslingen', 'Fellbach', 'Waiblingen');
```

Мысал

Сұраныс Баден-Вюртемберг жерінің аудан орталықтарынан барлық клиенттерді іздейді.

Шешімі:

SELECT тегі, аты, көшесі, пошта индексі, қаласы
FROM Клиенттер

WHERE E/мекен IN ('Stuttgart', 'Karlsruhe', 'Tübingen', 'Freiburg'); -

Ескерту:

IN операторы мәтінге, күні мен сандық өрістерге қолданылуы мүмкін.

6.3.6 Нөлдік мәндермен жұмыс

Ешқандай мәндер енгізілмеген бағандар NULL (нал деп оқылады). NULL мәні 0 саны немесе бос жол емес. NULL мәні бар өріспен арифметикалық операциялар NULL мәніне әкеледі. NOT NULL сипаты жазу өрісін іздейді.

Мысал

Сұраныс

```
SELECT тегі, аты, көшесі, пошта индексі, қаласы,  
Email FROM Клиенти  
WHERE Email is NOT NULL;  
электрондық пошта мекенжайы бар барлық клиенттерді таңдау.
```

Егер кесте өрісінде NULL мәні болмаса, яғни бос болмаса, бұл кестені жобалау кезінде ескеру қажет. Бастапқы кілттің өрістері, мысалы бос қалуы мүмкін емес. Сондай-ақ тегі немесе пошта индексі сияқты басқа өрістер үшін NULL мәндеріне тыйым салу орынды болуы мүмкін.

6.3.7 Деректерді сұрыптау

ORDER BY нұсқаулығы шығыс деректерін өрістер мазмұны бойынша сұрыптауға мүмкіндік береді. Сұрыптау тәртібі ASC (өсу бойынша) немесе DESC (кему бойынша) аргументін қосумен көрсетіледі. Әдепкі бойынша (дәлелсіз) өсу ретімен сұрыпталады.

Мысал

Клиенттер алфавиттік ретпен көрсетіледі:

```
SELECT *  
FROM Клиенттер  
ORDER BY тегі, аты ASC
```

Клиент №	Қалай керек	Тегі	Аты	Көшесі	Пошта индексі	Елді мекен	Телефон
11	мырза	Адлер	Астрид	Гюнтербургштрассе 56	40213	Дюссель- дорф	0211 543210
10	мырза	Амини	Ханс	Бухенландвег 120	89075	Ульм	0731 4538254
21	мырза	Бадер	Михаэль	Блаутальалле 235	89075	Ульм	4567890
11	мырза	Мюллер	Андреа	Адельштрассе 13	88446	Биберах	07364/894623
1	мырза	Мюллер	Ханс	Фридбергер Ландштрассе 204	89077	Аугсбург	0821/434343
2	мырза	Мюллер	Герта	Рингельштассе 2	86416	Крумбах	08282/465432
6	мырза	Winter	Сюзанна	Ам Вальдранд 1	89077	Ульм	0731/4892
11	мырза	Цвитель	Карп	Хюгельштрассе 123	89123	Эльхинген	07321/745645

Егер сұрыптау өлшемі ретінде ерістің екі мазмұны көрсетілсе, онда сұрыптау бірінші (тегі) кейін, сол ерістің мазмұнымен (мысалы, Мюллер) қосымша екінші ерістен кейін (мұнда аты) жүргізіледі.

Егер SELECT сезінен кейін "*" белгісі бар болса, онда көрсету үшін барлық ерістер таңдалады. Сұрыптау тәртібі екі өлшем бойынша түрлі болуы мүмкін.

Мысал

Велосипедтер ең жаңадан бастап ең ескіге дейін оларды сатып алу күніне сәйкес сұрыпталуы тиіс. Сатып алудың сол күні Тапсырыс өндіруші бойынша есу тәртібімен сұрыпталуы тиіс:

```
SELECT *
FROM Велосипедтер
ORDER BY сатып алынған күн DESC, өндіруші;
```

Велосипед нөмірі	Өндіруші	Белгіленуі	Түрі	Сатып алынған бағасы	Сатып алынған күні	Соңғы техникалық қызмет көрсетілуі
23	Hercules	Davos	Жол велосипеді	890,00 €	18.08.2017	08.09.2018
24	Peugeot	Stralis	Жол велосипеді	1 270,00 €	18.08.2017	01.01.2018
26	Yamaha	Sutra	Жарыс велосипеді	4 560,00 €	18.08.2017	01.01.2018
7	Hercules	GoClimb	Тау велосипеді	850,00 €	26.04.2016	08.01.2018
8	Panasonic	EasyRide	Жол велосипеді	960,00 €	26.04.2018	08.05.2019
18	Hirsch	Schneller Hirsch	Жарыс велосипеді	2 100,00 €	13.01.2018	19.06.2019
20	Panasonic	FirstClass	Жол велосипеді	1 325,00 €	05.10.2009	10.03.2019

6.3.8 Сұраныс нәтижелерін шектеу

TOP кілт сөзі белгілі бір жазба санымен сұрау нәтижелерін шектейді. Мысалы, жоғары 3 тек алғашқы 3 сұрау жазбасын шығарады.

Мысалы

Сатып алынған бағасы ең қымбат 5 велосипедті жалға беру қажет.

Шешімі:

```
SELECT TOP 5 *
FROM Велосипедтер
ORDER BY сатып алу бағасы DESC;
```

Нәтижесі:

Нөлө-сипед нөмірі	Өндіруші	Белгіленуі	Түрі	Сатып алынған бағасы	Сатып алынған күні	Соңғы қызмет көрсетілуі
9	Mercier	Excalibur Ultra Light	Жарыс велосипеді	3 750,00 €	27.04.2016	08.01.2020
16	Techno- bike	Supertandem	Жол велосипеді	3 200,00 €	28.12.2016	03.06.2020
17	Staiger	Supertandem		2 950,00 €	14.01.2016	19.06.2020
18	Systemo	Hurricane	Тау велосипеді	2 665,00 €	30.07.2016	12.04.2019
4	Miyata	Devant	Жол велосипеді	2 585,00 €	17.03.2018	29.11.2020

Кестедегі жазбалар әдетте сұрыпталмаған түрде сақталатындықтан, сұраныс ORDER BY сатып алу DESC бағасы сұрыптау функциясымен толықтырылады.

TOP функциясы белгілі бір жазба пайызын көрсетуі мүмкін. Синтаксис мысалы: SELECT TOP 10 percent ...

Тапсырма

10 пайыз жаңа велосипедтерді жалға беру үшін сұраныс жазыңыз.

Шешімі:

```
SELECT TOP 10 пайыз * FROM Велосипедтер
ORDER BY сатып алынған күні DESC;
```

Сұрыптау операторы да DESC болып табылады, өйткені жаңа күн мәндері алдыңғы мәндерден үлкен болып саналады.

6.3.9 SELECT-сұраныстарындағы функциялар

Агрегаттық функциялар

Агрегаттық функциялар (сондай-ақ топтық функциялар) белгілі бір өлшемдер бойынша кестенің бір немесе бірнеше бағандарын бағалайды. Нәтиже, мысалы, бағанның сомасы еріске шығады.

Функциясы	Нәтижесі
MIN (баған атауы)	Баған өрісі мазмұнының ең аз мәні
MAX (баған атауы)	Максималды мән
COUNT (баған атауы)	Бағандағы бар жазбалардың саны
COUNT (*)	Барлық жазбалар саны
SUM (баған атауы)	Баған өрісінің мазмұн сомасы
AVG (баған атауы)	Өрістің орташа арифметикалық мазмұны, AVG ағылшын тілінен.: average = орташа

Мысал

Барлық велосипедтердің жиынтық сатып алу құнын шығару үшін SQL-сұранысын жасау қажет.

Шешімі:

```
SELECT SUM (бастапқы бағасы) AS sum FROM
Велосипедтер;
```

Нәтиже:

Сомасы
41 337,00 €

Нәтижені беру кезінде қалаған тақырып бағанын шығаруды қамтамасыз ету үшін AS (= қалай) операторы қолданылады, одан кейін жаңа тақырып, мысалы, Сoma.

Статистикалық функцияның нәтижесі әдетте бір өріс болып табылады. Велосипед нөмірі сияқты басқа бағанның мазмұнын көрсету үшін бірдей пәрменді пайдалану әрекеті қате туралы хабарлайды.

Ескертулер:

Select қарапайым операторында агрегаттық функциясы бар бағанның мәндерінен артық көрсетілмейді.

COUNT (баған аты) функциясы SELECT COUNT (Email) FROM нұсқаулығы бос емес бағандағы өрістер санын есептейді. Клиенттер электрондық пошта мекенжайы тіркелген клиенттер санын анықтайды.

SELECT COUNT (*) FROM Клиенттер барлық Клиенттер кестесінің жазбаларын ескереді.

Ескертулер:

Кестедегі барлық жазбаларды қамту үшін, іздеу COUNT арқылы орындалады (*).

SUM () және AVG () функциялары сандық бағанның мазмұнына ғана қолданылуы мүмкін.

MIN () және MAX () функцияларын мәтін және күн өрістерімен пайдалануға болады, сонымен қатар мәтіндік өрістер ASCII:

A «аз» кодына сәйкес z қарағанда, ал кіші әріптер «үлкен», жоғарғы регистрге қарағанда, бірақ «аз», арнайы символдарға қарағанда өңделеді. Кейінгі күн алдыңғы күннен артық.

Мысал:

Келесі SQL нұсқаулығы соңғы велосипедті сатып алу күнін анықтайды.

```
SELECT MAX (сатып алу күні) FROM велосипедтер;
```

Арифметикалық операция

Бағандардың сандық мәндері қосу, азайту, көбейту және бөлу сияқты арифметикалық операцияларды орындау үшін пайдаланылуы мүмкін.

Бұл күн мәндеріне де қатысты, өйткені олар бүтін сандар (бүтін сандар) ретінде сақталады.

Күн функциялары

Күн функциясы күн мөнін өңдеуге мүмкіндік береді. Мысалы, жыл нөмірін күні немесе ағымдағы күні сүзгілеуге болады.

Мысал

100 күн бұрын қызмет көрсету күні өткен барлық велосипедтерді тізімдеу үшін келесі нұсқаулық қажет:

```
SELECT велосипед нөмірі, белгілеу,  
from велосипедтер соңғы қызмет  
WHERE соңғы қызмет көрсету < DATE () -100;
```

DATE () функциясы жүйелік күнді көрсетеді. Бұл мәннен 100 саны алынып, нәтиже соңғы қызмет өрісінің мазмұнымен салыстырылады. Аз мәнді барлық жазбалар шығады.

Ескерту:

Күн функциялары SQL стандартының бір бөлігі болып табылмайды, бірақ олар деректер қорының барлық жүйелерінде іс жүзінде бар. Алайда олар ішінара ерекшеленеді. Олар пайдаланылатын деректер қоры жүйесіне ете тәуелді. Қажет болса, тиісті басшылыққа немесе анықтамалық функцияға хабарласыңыз.

Access жылы, ағымдағы күнді көрсететін date () функциясына қосымша, қосымша күн функциялары бар:

Функциясы	Нәтижесі
DATE ()	Ағымдағы күн
DAY (күні)	1-ден 31-ге дейінгі сан түріндегі күн
MONTH (күні)	1-ден 12-ге дейінгі сан түріндегі ай
YEAR (күні)	Төрт таңбалы сан түріндегі жыл
ISDATE (күні)	Күннің дұрыстығын тексереді
Форматы (өрнегі, 'format')	Өдепкі мәнде күнді пішімдейді: D, DD: күн нөмірі, DDD, DDDD: күн нөмірі апта, M, MM: ай нөмірі, MMM, MMMM: ай аты YY, YYYY: жыл нөмірі

FORMAT операторы (кескін, пішім) күн мәндерін пішімделген түрде шығару үшін пайдаланылуы мүмкін.

Мысал

Жалдау деректерінің кестесінде қайтару жолдарының күні «2019 жылғы 26 сәуір, жұма» нысанында көрсетілуі тиіс.

```
SELECT FORMAT (қайтару күні, 'DDDD, DD. MMMM  
YYYY') AS [шот-фактура күні]
```

```
FROM жалға беру_күндері;
```

Нәтижесі:

Қайтару күні
Сейсенбі, 06 ақпан, 2018 ж.
Сәрсенбі, 04 қазан, 2017 ж.
Сейсенбі, 12 наурыз 2019 ж.
Сенбі, 27 сәуір, 2019 ж.

Символдық жол функциялары

Әртүрлі деректер қорының жүйелері символдық жолдарды өңдеу және түрлендіру үшін әртүрлі функцияларды ұсынады. Алайда тиісті функциялардың әрекеті өте ұқсас. Мысалы, Oracle деректер қоры жүйесінде ASCII (n) функциясы ASCII символының кодтық нөмірін қайтарады, Access функциясы ASC (n) функциясы бірдей етеді.

Функциясы	Әрекеті, мысалдар
ASC (n)	Код нөмірін көрсетеді n, ASC('M') в77.
LCASE (жол)	Таңбаларды төменгі регистрге түрлендіреді, LCASE ('Müller') → 'müller'
LEFT (жол, ұзындығы)	Сол жақ жиегі бойынша түзетілген символдық жол береді, LEFT('Müller', 2) в 'Mü'.
LEN (жол)	Таңбалар санын шығарады, LEN('Müller') 6.
RIGHT (жол, ұзындығы)	Оң жақ шетінде түзетілген символдық жол береді, LEFT () қарау.
UCASE (жол)	Таңбаларды жоғарғы регистрге түрлендіреді, UCASE('text') → 'TEXT'
&	Біріктіруге арналған оператор (Access)'Hans' & '' & 'Müller' → 'Hans Müller'

Мысал

Сақталған емлеге қарамастан, бірінші әріп бас әріп ретінде бейнеленуі үшін, ал келесі әріптер кіші әріптермен бейнеленуі үшін клиенттің тегін шығаратын оператор қажет.

```
SELECT Left(UCase (Nachname), 1) & LCase(Right
(Nachname, Len
) -1) AS [ 'тегі' ]
```

From клиенттер;

«Мюллер» тегін кездейсоқ енгізу осы нұсқаулықпен «Мюллер» болып өзгертіледі.

Түрлендіру функциялары

Мысалы, түрлендіру функциясы арқылы символдық жолдар санға және керісінше түрлендіріледі. Мысалдар:

Функциясы	Әрекеті, мысалдар
CDATE ('күн жолы')	Символдық жол, күн түріндегі сан DATE CDATE ('12 .03.2019 ') - 12.03.2019
STR (сан)	Символдық жолдағы сан, STR(89077) - '89077'
VAL (жол)	Символдық жолды санға, VAL('34') -34
CINT (нөмір)	INTEGER-ға конвертациялайды, CINT-ке дейін дөңгелектейді (45,8) - 46 CINT (-67,4) в -67

Мысалы, CDATE («дата») функциясы сандардың символдық жолын немесе бүтін санды бастапқы түріне байланысты мүмкіндігінше күніне түрлендіреді.

CINT() функциясы келесі жоғары бүтін санға дейін дөңгелектенеді және нәтижесінде көрсетеді.

Мысал

Келесі SQL операторы әр велосипедті жалға алу бағасын 5%-ға ұлғайтылған, еуроға толық сомаға дейін дөңгелектелген баға топтары кестесінде көрсетеді.

```
SELECT CINT (бағасы* 1.05)
FROM Баға топтары;
```

Математикалық және логикалық функциялар

Математикалық есептерді шешу үшін әріс мазмұнының квадрат түбірін қалыптастыру үшін келесі функциялар қолданылады.

Функция	Нәтижесі
ABS (сан)	Абсолюттік мәні (теріс белгісіз)
SQR (сан)	Санның түбірі: SQR (16) = 4
RND ()	Кездейсоқ сан 0-ден 1-ге дейін
ISNULL (көрініс)	Мән = 1, мысалы, өрістің мазмұны бос болса, өрістің мазмұны бос болмаса, = 0 мәні.
EXP (мән)	Эйлер е = 2.71828 нөмірімен е мәнін көрсетеді ... база ретінде.
LOG (мән)	Табиғи логарифм, яғни LN(EXP (мән)) = мән.
LOG10 (мән)	Негіздегі сандық мәнің Логарифмі 10
SIGN (мән)	-1 теріс мәндер үшін және 1 оң мәндер үшін
MOD (мән, белгіш)	Мәннің/белушінің қалдық құны (бүтін саны)

Мысал

Кабельдің көлденең қимасы бар кестеден диаметрін есептеу.

Шешімі:

```
SELECT алаңы, 2*SQR (алаң/(3.1415*2)) AS диаметр
FROM Кабельдік желілер;
```

RND() функциясы кездейсоқ санды өзгермелі үтірлі >0 және <1 санын жасайды, мысалы, 0,3452419.

ISNULL (өрнек) функциясы кескін, мысалы, өріс мәні бос екенін тексереді. Егер бұл болса, функция 1 ('true') мәнін, әйтпесе 0 ('false') мәнін қайтарады.

6.3.10 Деректерді топтастыру

Топтастыру деректерді өрістің бірдей мазмұнымен өріске топтастыруды білдіреді. Бұл SQL-group BY өрнегі арқылы жасалады.

Мысал

Келесі Нұсқаулық Paradiso компаниясының қызмет көрсету аймағын әр елді мекеннен тұратын жері мен клиенттер санын көрсеткенде сәйкестендіреді:

```
SELECT е/мекен, COUNT (клиент№) AS  
саны FROM Клиенттер  
GROUP BY е/мекен;
```

Елді мекен	Саны
Берлин	2
Биберах	2
Йеттинген-Шеппах	2
Дюссельдорф	35
Neckarsulm	1
Хан-Грютен	12
Ульм	11
Цвиккау	1
...	...

Алдымен топ құру үшін GROUP BY Ort өрнегін е/мекен өрісінде бірдей жазбасы бар деректер қорының барлық жолдарынан қолданылады. Кейіннен, осы топтардың әрқайсысында клиенттің № бойынша жазбалар санын есептеу үшін COUNT (клиенттің №) функциясы қолданылады.

Ескерту:

SELECT негізгі сөзінен кейін статистикалық функцияға қосымша көрсетілген барлық өрістер GROUP BY топтастыру операторынан кейін көрсетілуі тиіс, әйтпесе қате туралы хабар пайда болады.

Топтау шарттары – HAVING:

Жеке топтарды көрсету, сондай-ақ HAVING шарттары операторы арқылы шартты түрде жасауға болады.

Мысал

```
SELECT Өндіруші, SUM (Сатып алу бағасы) AS сомасы
FROM Велосипедтер
GROUP BY Өндірушілер
HAVING SUM (бастапқы бағасы) > 4000;
```

Өндіруші	Сома
Hirsch	5 932,00 €
Panasonic	6 095,00 €

Мысалы, HAVING SUM (сатып алу бағасы) > 4000 сатып алудың жалпы бағасы («жалпы сома») 4000€-нан асатын топтардың ғана шығарылуына әкеледі.

Ескерту:

Having кілт сөзінен кейінгі шарт әрқашан бүкіл топқа қолданылуы тиіс.

6.3.11 Бірнеше кесте бойынша сұраныстар

Joins (to join = байлау) бірнеше кесте арасында логикалық байланыстарды жасайды.

Мысалы, клиенттердің тиісті аттарымен барлық жалдау деректерін көрсету үшін екі кесте өрістері мен резервтеу керсетілуі тиіс.

Нұсқаулық осылай:

```
SELECT. Клиенттің №, тегі, аты,
резервтеу, жалға беру күні
from клиенттер, брондау;
```

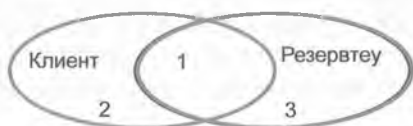
Нәтиже барлық клиенттерді жалға алу деректерімен біріктіретін қорытынды болып табылады. 50 жазбасы бар кестеден және 400 жазбасы бар кестеден 20000 жазбадан нәтижелер жиынтығы жасалады.

Жазбалардың саны қосылатын әрбір кесте жазбаларының санына көбейтіледі.

Бірлестіктер түрлері

Екі кесте мысалында клиенттер мен резервтеу әртүрлі бірлестіктер түсіндіріледі:

- резервке тапсырыс берген клиенттер (1),
- резервтеуге тапсырыс бермеген клиенттер (2) және
- клиенттермен байланысты емес резервтеу бар (3)



Тегі	Көрінісі	Түсініктеме	Синтаксис мысалы
INNER JOIN, EQUI JOIN		Велосипедтерді резервтеуге байланысты клиенттерді көрсету.	SELECT * FROM Клиенттер INNER JOIN Резервтеу ON клиенттер клиент № = резервтеу клиент №;
NATURAL JOIN		Нәтижесі INNER JOIN	SELECT * FROM клиенттер, брондау WHERE клиент № = резервтеу, клиент №;
LEFT JOIN,		Барлық клиенттерді көрсету және (егер бар болса) байланысты резервтеу.	SELECT * FROM Клиенттер LEFT JOIN Резервтеу ON клиенттер клиент № = резервтеу клиент №;
RIGHT JOIN,			SELECT * FROM Клиенттер RIGHT JOIN Резервтеу ON клиенттер клиент № = резервтеу клиент №;
LEFT OUTER JOIN		Резервтеуге тапсырыс бермеген клиенттерді ғана шығару.	SELECT * FROM Клиенттер LEFT JOIN Резервтеу ON клиенттер клиент № = резервтеу клиент №; WHERE Резервтеу. клиент № is NULL;
RIGHT OUTER JOIN		Клиенттердің ешқайсысына байланбаған резервтеуді шығару.	SELECT * FROM Клиенттер RIGHT JOIN Резервтеу ON клиенттер, клиент № = резервтеу клиент №; WHERE Резервтеу. Клиент № is NULL;
FULL OUTER JOIN		Барлық клиенттерді резервтеусіз және клиенттерге байланысты емес барлық резервтеусіз шығару	SELECT * FROM Клиенттер FULL JOIN Резервтеу ON клиенттер клиент № = резервтеу клиент №; WHERE Резервтеу. клиент № is NULL; (Қолдау көрсетілмейді ACCESS-SQL)

Inner-Joins, Equi-Joins, Natural Joins

Бірнеше кестелерді, теңдікті біріктіруді, ішкі қосылыстарды қамтитын сұраныстар үшін. Бұл мысалда нөмірі резервтеу кестесінде көрсетілген клиенттер ғана назарға алынуы мүмкін. Бұл үшін SQL ко-

мандасында бірегей байланыс ережесі белгіленуі тиіс. Мына шартпен іске асырылады:

WHERE клиенттің № = резервтеу. Клиенттің №.

Осылайша, толық нұсқаулық келесідей түрге ие болады:

```
SELECT клиенттің №, Тегі, Аты, Әкесінің аты,
резервтеу, жалға беру күні
FROM клиенттер, резервтеу
WHERE клиенттің № = резервтеу №;
```

өртүрлі кестедегі аттары бірдей ерістерді ажырату үшін баған атауына, мысалы, клиенттің № кесте атының префиксі қосылады.

Бірнеше кестелерді біріктіру SQL сұрауында сипатталады, мысалы, WHERE шарты түрінде.

Мысал

Клиенттің деректері бар барлық резервтеу (клиенттің немірі, тегі, аты, тұрғылықты жері) және велосипедтің деректері (велосипедтің немірі, өндіруші, аты) шығарылады. Нұсқаулық:

```
SELECT К. Клиенттің №, тегі, аты, е/мекені,
жалға беру күні, жалға беру мерзімі AS жалға беру мерзімі, велосипед №, өндіруші
```

```
From клиенттер ASK, велосипедтер ASF, Сақтық көшірме ASR WHERE к. Клиент № = R. Клиент № және F. Велосипед № = R. велосипед №;
```

Жалпы бағада жалға берудің жалпы құны көрсетіледі.

R. баға * L as жалға беру ұзақтығы. Нәтижесі:

```
SELECT К. Клиенттің №, тегі, аты, е/мекені, жалға беру күні, жалға беру ұзақтығы AS ұзақтығы, F. велосипед AS №, өндіруші, P. бағасы * l жалға беру бағасы AS жалпы құны
From ask клиенттері, ASP баға топтары, AS f велосипед, ASr
WHERE к. Kdr = R. Капе AND F. велосипед нөмірі = R. Franc AND P. баға тобы = баға тобы F.;
```

Қорытынды келесідей:

Клиент №	Тегі	Аты	Жалға берілген күні	№	Өндіруші	Жалпы құны
2	Мюллер	Герта	9	23	Hercules	108,00 €
3	Мюллер	Ханс	18	13	Systemo	486,00 €
5	Winter	Сюзанна	12	8	Panasonic	132,00 €
5	Winter	Сюзанна	9	21	Hercules	135,00 €
8	Цвигель	Карл	9	8	Panasonic	99,00 €
10	Хазе	Ханна	12	21	Hercules	180,00 €

Резервтеу кестесі клиенттер кестесіне біріктіру арқылы байланысты болғандықтан, бұл да резервтеу кестелерінің қатынасына сәйкес WHERE шарты ретінде көрсетіледі – циклдер – баға топтары. Сондықтан бірнеше кесте үшін SQL күрделі сұрауын әзірлеу кезінде қатынас табиғатын және қозғалған өрістерді білу қажет.

SQL синтаксисін жеңілдету үшін сұраныстағы кестелер атаулары өзгертілуі мүмкін. Ол үшін from операторында кесте атауына тән әріп беріледі, мысалы, В.К, Р, R, F (яғни F велосипедтері). Бұл кестеге Жаңа (уақытша) атау береді. Бұл жағдай да кесте тек осы хат бойынша сұрау салуда қолжетімді.

Кілт сөз as Шығыс деректеріндегі өрістерді қайта атау үшін пайдаланылады. Бұл, мысалы, ешқандай аты басқаша анықталмаған, есептелетін өрістер үшін маңызды.

Жоғарыда келтірілген мысалда бұл: Р. Бағасы *аз жалдау мерзімі жалпы құны

Ішкі бірлестіктер INNER JOIN кілт сөздерімен сипатталуы мүмкін.

Мысал

Барлық резервтелген велосипедтер (велосипед нөмірі, атауы, өндіруші) резервтеудің маңызды деректерімен (ResNr, LoanDate, RentalDuration) көрсетілуі тиіс. Нұсқаулық:

```
SELECT F. велосипед №, F.Аты, F.Өндіруші, F.соңғы техникалық қызмет көрсетілуі, R.резервтеу №, R. жалға беру күні, R.жалға беру мерзімі
```

```
FROM Велосипедтер F INNER JOIN Резервтеу ON R F. Велосипед № = R. велосипед №;
```

Шығу нәтижелері:

Велосипед нөмірі	Атауы	Өндіруші	Соңғы қызмет көрсетілуі	резервтеу №	Жалға беру күні	Жалға беру ұзақтығы
4	Devant	Miyata	29.08.2018	20	22.01.2021	2
4	Devant	Miyata	29.11.2018	22	01.04.2020	5
8	Galata	Panasonic	08.05.2018	7	18.01.2019	9
13	Hurrican	Systemo	12.04.2019	5	16.06.2020	3
14	DownHill Racer	Hitachi	21.05.2019	8	15.04.2021	9
19	Klettergemse	Hirsch	27.02.2018	12	15.12.2022	6

Ескерту:

INNER-Joins, EQUI-Joins und Naturai Joins, олардың әрқайсысы бірдей нәтиже береді. Олар әртүрлі синтаксиспен ғана сипатталады. Табиғи бірлестіктер үшін where ұсынысындағы кестелердің қосылымы сипатталады, INNER (= EQUI бірлестіктері) қосылымдары үшін бұл FROM шарты жасалады.

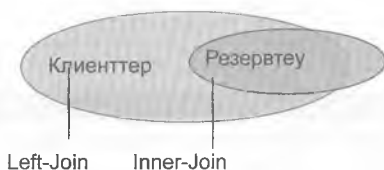
Left-Joins (соңдай-ақ Left-Inner-Joins)

LEFT-JOIN бір кестенің барлық жазбаларын (мысалы, клиенттер) алады және оларды басқа кестедегі (резервтеу) барлық жарамды жазбалармен немесе егер екінші кестеде тиісті жазба табылмаса, NULL-ден біріктіреді. Екі кесте үшін тиісті ER моделін білу маңызды:



Қарым-қатынас әрбір клиенттің бірнеше резервтеу немесе резервтеуінің болмауы мүмкін дегенді білдіреді. Егер резервтеу тағайындалмаған жағдайда, LEFT JOIN нәтижесі бағанасында жалға беру күні NULL мәнін қамтиды.

Егер Сіз брондау сұранысына велосипедін әлі брондамаған клиенттерді де қосқыңыз келсе, онда сілтеме шарты өзгертілуі керек. Equi-Join екі ауқымның қиылысына сәйкес келеді, ал: клиенттер және резервтеу, екі жиынтықтың сол бөлігі қосылуы тиіс.



Мұндай Left Join аргументі жалпы синтаксисімен Access-те қолданылуы мүмкін

```
... FROM таб1 LEFT JOIN таб2 ON таб1.поле = таб2.өрісі. ..
```

Мысал

```
SELECT K. клиент №, тегі, аты, орны, жалға берілген күні AS жалға берілген күні, резервтеу AS L жалға беру мерзімі
```

```
FROM клиенттер K LEFT JOIN Резервтеу ON R K.
```

```
клиент № = R. клиент №;
```

Шығу нәтижелері:

Клиент №	Тегі	Аты	Елді мекен	Жалға беру күні	Жалға беру мерзімі
1	Пальмерт	Карло	Йеттинген		
2	Мюллер	Герта	Крумбах	16.10.2019	9
2	Мюллер	Герта	Крумбах	22.01.2020	3

3	Мюллер	Ханс	Ульм	27.02.2022	18
4	Шульце	Анна	Ульм		
5	Winter	Сюзанна	Ульм	16.10.2021	9
5	Winter	Сюзанна	Ульм	01.05.2021	12
6	Мюллер	Андреа	Биберах		
7	Цвибель	Карл	Эльхинген	18.01.2021	9

Сыртқы Қосылыстар (Outer-Joins)

Сыртқы бірлестіктер (мысалы, Left Outer Join) нәтиже ретінде екінші кестеде анықтамалық деректер жоқ жазбаларды ғана шығарады.

Мысал

әлі резервте болмаған клиенттердің деректерін көрсету.
 SELECT K. Клиенттің №, тегі, аты, әкесінің аты, жалға беру орны, күні AS жалға беру күні, тапсыру ұзақтығы AS тапсыру мерзімі
 From клиенттер үшін LEFT JOIN резервтеу ON к...
 клиенттің № = R мұндағы R. Клиенттің №. IS NULL
 клиенттің №;

Қорытынды:

Клиент №	Тегі	Аты	Елді мекен	Жалға беру күні	Жалға беру мерзімі
1	Пальмерт	Карло	Йеттинген		
4	Шульце	Анна	Ульм		
6	Мюллер	Андреа	Биберах		
9	Уинклер	Клаус	Тальфинген		
10	Хазе	Ханна	Гюнцбург		
11	Хан	Изабель	Зенден		

Ескерту:

Ішкі бірлестік (сондай-ақ тең бірлестік, табиғи бірлестік деп аталады): бірнеше кестелерге сұрау, онда тек деректер қиылысуы ғана көрсетіледі.

Left-Join/Right Join (сондай-ақ Left-Inner-Join және Right-Inner-Join): қиылысқа қосымша ішкі жиындардың сол / оң жақ бөлімі қосылады.

Outer-Join: қиылысуды болдырмайды және қосылған кестеде жазбалары жоқ жазбаларды ғана көрсетеді. Access-те WHERE операторын қосу арқылы қойылады [өріс аты] NULL. операторы WHERE [өріс атауы] NULL.

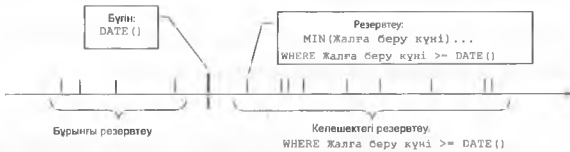
6.3.12 ҚОСЫМША СҰРАНЫСТАР

Іздеу шарты басқа сұраныстың нәтижесіне байланысты болса, қосымша сұраныстар (Subqueries) қажет. WHERE белімінде логикалық бағаланатын жағдайға әкелетін деректер ғана шығарылады, әйтпесе қате туралы хабар пайда болады.

Мысал

Брондалуы келесі тәртіпте орындалуы тиіс клиенттің атын көрсететін сұраныс жасау. Велосипедтің нөмірі мен атауын көрсету.

Алдымен келесі брондау күнін шығаратын сұрау жасау пайдалы. MIN (жалға беру күні) функциясы ең ерте сақталған резервтеу күнін шығарады, содан кейін WHERE > = DATE () шарты сұранысты ең аз күнмен шектейді.



Осылайша, кейінірек қосымша сұраныс болып қызмет ететін қалаған күн мәнін іздеу үшін сұраныс жасау:

```
SELECT MIN (жалға беру күні) FROM резервтеу
Мұндағы жалға беру күні > = Date ();
```

Бұл сұраныстың нәтижесі, мысалы, болашақ тапсырудың ең аз күнін білдіреді

Қосымша сұраныс
16.10.201X

Енді резервтеу кестесіндегі тиісті жазба WHERE Borrow Date = [Subquery] шарты бойынша табылуы керек. Деректер бірнеше кестеден есептелгендіктен, кестелер клиенттер, резервтеу және велосипедтер қажетті теңдеулермен байланысты болуы тиіс.

Барлық сұраныс:

```
SELECT F велосипед №, F. атауы, тегі, жалға беру күні
FROM Велосипедтер AS F, Резервтеу AS R,
Клиенттер AS K
WHERE клиент № = R. клиент №
AND F. велосипед № = R. велосипед № табиғи
байланыс Natural Joins
AND жалға беру күні = (
    SELECT MIN (жалға берілген күні Subquery
    FROM Резервтеу Қосымша сұраныс
```

```

WHERE жалға берілген күні > = DATE()
);
WHERE жалға берілген күні > = DATE()
);

```

Шығатын нәтижелер:

Келесі жалға беру			
№	Белгіленуі	Тегі	Алынған күні
21	StormRide	Winter	16.10.201X

Ескерту:

Қосалқы сұраудың нәтижесі салыстыру өрісімен деректер түріне сәйкес болуы тиіс. Сонымен қатар, егер, мысалы, ол in операторымен шарттың бір бөлігі болып табылса және осылайша, өрістер тізіміне сәйкес келсе, қосымша сұраныс тек бірнеше нәтиже бере алады.

Күрделі сұрау әрқашан модульдік болуы керек. Нәтижесінде қателер уақытында табылуы және оңай жойылуы мүмкін.

Мысал

Велосипед өндірушілердің е/мекенінде тұратын клиенттердің деректерін шығару керек. Алдымен велосипед өндірушілердің деректері бар өндірушілер кестесі құрылатыны міндетті шарт болып табылады.)

```

SELECT *
From клиенттер
WHERE клиенттер. е/мекен IN
(
    SELECT өндіруші. е/мекен
    FROM өндіруші
);

```

6.4 SQL АРҚЫЛЫ ДЕРЕКТЕРДІ ӨНДЕУ

SQL-де деректерді манипуляциялау тілі (DML) жаңа жазбаларды енгізе отырып, қолда бар жазбаларды жою немесе оларды өзгерту арқылы деректерді өңдеуге мүмкіндік береді.

6.4.1 Деректер қорының жолдарын кірістіру

Ағымдағы бизнес процестеріне сәйкес деректерді жаңарту үшін жаңа деректер қосылады және қажетсіз деректер жойылады. Жазбаларды қосу INSERT негізгі операторымен SQL-де орын алады. INSERT жалпы синтаксисі:

INSERT INTO кесте (1 баған, 2 баған, ...) VALUES (мазмұны 1, мазмұны 2, ...);

Егер енгізу деректері бар кестеден таңдалса, тиісті SELECT операторы қосылады. INSERT нұсқаулығының құрылымы:

INSERT INTO кесте (1 баған, 2 баған, ...) SELECT (1 баған, 2 баған,...)

Мысалы

SQL операторының көмегімен Stiegl өндірушінің «Easygo» велосипедіне арналған деректер қорының жаңа жолын және велосипед кестесіндегі велосипед № қосыңыз.

Шешімі:

```
INSERT INTO велосипедтер (велосипед №, өндіруші, атауы, түрі) VALUES (12, 'Stiegl', 'Easygo', 'Tourenrad');
```

Ескерту:

INSERT кейін көрсетілген өрістер реті VALUES арқылы көрсетілген мәндерге сәйкес келеді. Өрістердің деректер түрі тиісті мәнің деректер түріне сәйкес келуі тиіс.

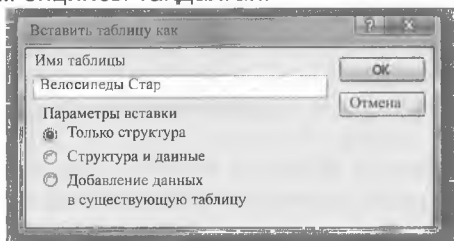
Мысал

2004 жылға дейін сатып алынған велосипедтер пайдаланудан шығарылуы тиіс. Деректерді біржола жоғалтпау үшін велосипедтің деректерін Велосипедшілер кестесіне резервтік көшіру мақсатында енгізетін нұсқаулық қажет.

```
INSERT INTO велосипедші SELECT *  
From велосипедтер  
WHERE сатып алу күні < # 1/1/2004#;
```

Сол құрылымды Fahr-raederAlt кестесі қажет. Оны CREATE TABLE операторы немесе бар велосипед кестесі арқылы жасауға болады.

мысалы, Access бағдарламасында мәтінмәндік мәзір командасын көшіруге болады, содан кейін кірістіруге болады. Тек құрылым опциясы таңдалған.



Бұл мысалда SELECT операторы 2004 жылға дейін сатып алынған велосипедтер туралы барлық жазбаларды іздейді, содан кейін оларды жаңа кестеде сақтайды.

Ескерту:

Деректер түрлері және екі кестенің тиісті ерістерінің аттары осы әдіске сәйкес келуі тиіс.

6.4.2 Деректер қорының жолдарын жою

Түсірілген жазбаларды жою

SQL DELETE пәрменімен жүзеге асырылады. Жалпы синтаксис:

```
Delete төлсипат аты / * FROM кесте WHERE шарты;
```

Мысал

SQL нұсқаулығы 2004 жылға дейін сатып алынған велосипедтерді кестеден шығаруға арналған велосипедтер.

Шешімі:

```
DELETE *  
FROM велосипедтер  
WHERE сатып алу күні < # 1/1/2004#;
```

6.4.3 Деректерді жаңарту

Ерістің мазмұнын өзгерту үшін көптеген жазбалар бір бағытта өзгертілуі керек болса, UPDATE Жаңарту командасын пайдаланыңыз.

Жалпы синтаксис:

```
UPDATE кесте  
SETAttribute name = WHERE шарты мәні;
```

Мысал

Баға топтары бірдей жалға бағасы бар велосипедтерді жинақтау үшін Paradise деректер қоры мысалында пайдаланылады. 2006 жылға дейін сатып алынған велосипедтерді жалдау бағасы бір баға тобына азайтылуы тиіс. Тиісті жазбаларда өзгерістер сақталуы тиіс.

Шешімі:

```
UPDATE велосипедтері  
SET баға топтамасы = баға топтамасы - 1  
WHERE YEAR (сатып алынған күні) < 2006;
```

Мұнда баға тобының мазмұны WHERE шартына сәйкес келетін және 2006 жылға дейін сатып алынған велосипед кестесіндегі барлық деректер жазбаларында 1-ге азаяды.

6.5 ДЕРЕКТЕР ҚОРЫНЫҢ КЕЛІСІЛУІ

Деректер қорының келісілуі. Ішкі сақтау құрылымдары мен қол жеткізу жолдарының дұрыстығын сипаттайды.

Деректер қорын әдетте бірнеше пайдаланушы бір уақытта басқаратындықтан, мысалы, деректерді жою операциялары жойылуы мүмкін емес. Жою кейін, басқа пайдаланушылар өзгертілген деректер массивімен жұмыс істей алады. Жоюдың мүмкін еместігі, осылайша, деректер қорындағы ауытқуларды (логикалық қайшылықтарды) болдырмайды және деректер дәйекті болып қалады (қарама-қайшы емес).

Мысал

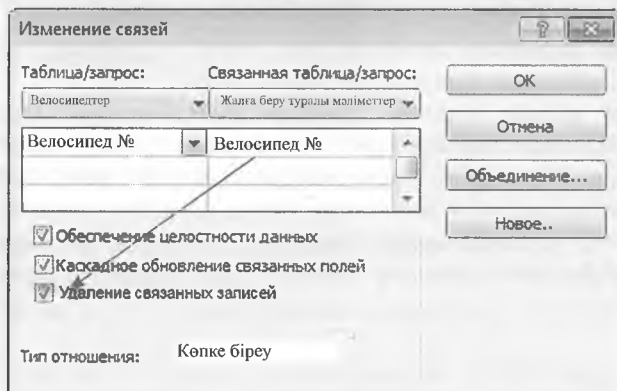
Деректер қорының құрылымы клиенттің сақталған мекенжайы осы клиенттің басқа мекенжайына қайшы келмейтініне кепілдік беруі тиіс. Әйтпесе, шотты жіберу мүмкін емес.

Деректер қорын жоспарлау кезінде кірістіру, өзгерту және кірістіру аномалиясының алдын алу қажет.

Жою ауытқулары бастапқы кілтінің мәні басқа кестеден тәуелді жазба болып табылатын жазба жойылғанда пайда болады. Егер, мысалы, 15 клиентінің нөмірі бар клиент туралы деректерді жазу жойылған болса, онда 15 клиенттің нөміріне қатысты жалдау деректері енді шығарылмайды.

Өзгерістер ауытқулары, мысалы, деректерді бірнеше рет сақтау салдарынан пайда болады. Егер деректер (мысалы, клиенттің мекенжайы) бір кестеде өзгерсе, онда олар басқа кестедегі деректерге қайшы келеді.

Кірістіру ауытқулары деректер қорының жолдарын енгізгеннен кейін мәндер деректер қорында бірнеше рет пайда болып, бір-бірімен қақтығысқанда пайда болады. Бұл жағдайда, байланыс деректер қорын жобалау кезеңінде орнатылады, сондықтан жою процедуралары оңай бақылануы мүмкін. Егер, мысалы, велосипед кестесінде жойылса, ал осы велосипедтің нөміріне жалға беру кестесінде деректер жазылса, онда осы нөмірдің астында велосипедті жалға беру деректерін жазу да жойылуы тиіс. Мысалы, Access-те қатынасты өңдеу тілқатысу терезесіндегі байланысты жазбаларға кетіру жалаушасын орнату арқылы жасауға болады:



6.6 ТРАНЗАКЦИЯЛАР

Деректер қоры әрдайым, тіпті аппараттық немесе бағдарламалық істен кейін де келісілуі тиіс. Екі банктің шоттары арасында аударылған жағдайда, шот сомасына азайтылған және В шотына осы шотқа есептелген жағдайда ғана аударым жарамды болуы мүмкін.

1-операция: Есебі А:		
шоттың ескі балансы	1300 €	
шығын:	250 €	
жаңа баланс:	1050 €	250 €
Операция 2: есебі Б:		
шоттың ескі балансы:	4320 €	
Пайда:	250 €	
жаңа баланс:	4570 €	

Егер 1 операциясын жүргізгеннен кейін қоректену іркілісі орын алса, онда шот қалдықтары бұдан былай келісілмейді, өйткені а шотындағы қалдық 250 еуроға азайған, ал Б шотының қалдығы әлі ұлғайған жоқ.

Жеткізу аяқталғанына көз жеткізу үшін екі операция әрдайым транзакция түрінде орындалады.

Анықтама:

Транзакция – бұл бір-бірімен логикалық байланысқан және деректердің келісімділігін қамтамасыз ететін SQL-операторлардың тізбегі.

Ескерту:

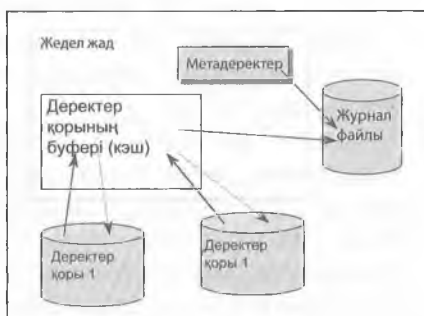
Транзакцияны орындау кезінде транзакция қате болған жағдайда толық немесе толық жойылғанына көз жеткізу қажет. Келесі нұсқаулар алдыңғы жұмыстар сәтті орындалған жағдайда ғана орындалады.

Деректер массивінің жарамдылығын қамтамасыз ету үшін келесі қадамдар жасалады:

Қадам	Тапсырма
1. Деректерді санау	Деректер тасымалдаушыдан есептеледі.
2. Ағымдағы деректер мәнін есте сақтау	Өзгертілуі тиіс деректер журнал файлына (Before-Image) жазылады.
3. Деректерді өзгерту	Жад деректерін өзгерту, басқа пайдаланушылар үшін осы жазбаларды құлыптау (журналдағы жазу).
4. Өзгертілген деректер мәндерін есте сақтау	Өзгертілген деректер журнал файлына (After-Image) жазылады.
5. COMMIT арқылы транзакцияны аяқтау	Барлық бейнелер мен метадеректерді журнал файлына жазу. Транзакцияның аяқталуын журнал файлына белгілеңіз. Бұғаттауды шешу.
6. Транзакциялар ROLLBACK (=Раcтау)	Метадеректі транзакцияны қалпына келтіру. Деректер қорына жазылған өзгертілген деректер жойылады. Бұғаттауды шешу.
7. Өзгерістерді сақтау	Өзгертілген деректер деректер қорына жазылады.

Алдымен деректер есептеледі, яғни А шотының балансы: 1300 €. Әлі өзгертілмеген деректер енді Before-Image (=өзгертілгенге дейінгі сурет) ретінде жазылады. Бұл деректер қоры жүйесіндегі энергияға тәуелді файл.

Енді қажетті командалармен (UPDATE, DELETE, INSERT) жедел жадындағы деректерді өзгерту орындалады. Бұл ретте пайдаланушылар үшін құлыпталған метадеректер (=алдын ала өзгертілген деректер) жасалады. Осылайша, А шотындағы қалдық 250 € жаңартумен азаяды, яғни 1050 € дейін. Бұл өзгертілген баланс журнал файлында After-Image (=өзгертуден кейінгі сурет) ретінде сақталады.



Осылайша, жаңа деректер журнал файлындағы After-Image сияқты қол жетімді.

Егер әлі қате болмаса, COMMIT командасы журнал файлындағы транзакцияның аяқталуын белгілейді. Бұғаттауды алу. Метадеректер деректер қорына жазылады.

COMMIT командасының алдында қате орын алса, ROLLBACK нұсқауы барлық метадеректі транзакцияны тастайды. Алдыңғы деректер журнал файлындағы Before-Images ретінде сақталады. Транзакцияның соңында журнал файлына ескертілгендіктен, оны өзгерту аяқталған транзакцияға жататын жүйе істен шыққан жағдайда кез келген уақытта табуға болады.

Егер журнал файлы деректер қоры орналасқан тасымалдағышта жасалмаған болса, екі дерек файлы да, журнал файлы мен деректер қоры бір уақытта жойылатыны екіталай. Осылайша, соңғы резервтік көшірме мен журнал файлының көмегімен деректер жиыны келісілген деректер жиынтығын алу үшін кез келген уақытта қалпына келтірілуі мүмкін.

6.7 6-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР

1-тапсырма

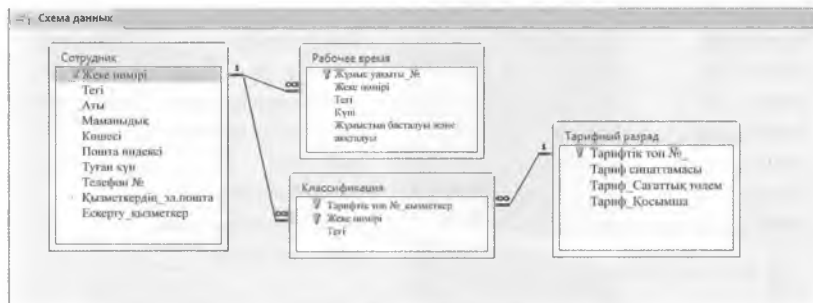
Дәрігерлік практика (дәрігер кабинеті) үшін мынадай талаптармен деректер қорын әзірлеу:

- ▶ пациенттердің жеке процедуралары туралы ақпаратты басқару қажет. Ол үшін күнді, диагнозды және ескертулерді сақтау қажет.
- ▶ әрбір пациент тек бір медициналық сақтандыру қорының қатысушысы болып табылады. Жеке сақтандыруы бар пациенттер есепке алынбайды.
- ▶ кабинетте мынадай қызметтер ұсынылады (қан жинау, қан қысымының мониторингі, зертханалық талдау, егжей-тегжейлі тексеру, бағыттарды жазып беру, денсаулық жағдайы туралы анықтама, рецептер беру). Мұнда кейде емдеу үшін бірнеше тағайындау беріледі. Қызметтер нөмірі, сипаттамасы және бағасы бар каталогқа сәйкес жазылуы тиіс.
- ▶ рецептілерге қатысты тағайындалған препарат, Өндіруші туралы, сондай-ақ баға және негізгі әсер етуші зат туралы деректер сақталуы тиіс.
- ▶ ұсынылатын қызметтерді қоса алғанда, әрбір пациент үшін емдеу құны туралы ақпарат қолжетімді болуы тиіс. (Құнның ықтимал ұлғаюын белгілеу қажет емес.)

Қолайлы деректер қорын жасаңыз. Қажетті байланыстарды орнатыңыз.

2-тапсырма

Қызметкерлердің еңбекақысын есептеу үшін дәрігер кабинетінде «дәрігер кабинеті» деректер қорын келесі кеңейту қолданылады.



- Бұл жоба бірінші, екінші және үшінші қалыпты формадағы қалыптау ережелерін қандай уақытта бұзады? Бұзушылықтарды қалай болдырмау керектігін ұсыныңыз.
Келесі тапсырмаларды орындайтын SQL сұрауларын жасаңыз:
- 46 *** пошта индексі бар барлық қызметкерлер туралы ақпаратты шығару.
- дәрігер кабинетінде қанша қызметкер жұмыс істейді?
- қандай қызметкерлер 23.12.2015 жұмыс істеді?

3-тапсырма

Ұйымның бірінде деректер қорының көмегімен жобаларды басқаруды құру қажет. Жүйеге әрбір қызметкердің қол жетімділігі құпия сөз арқылы жүзеге асырылуы тиіс.

Құпия сөз қызметкерлер кестесінде сақталады. Әрбір жоба үшін аты, сондай-ақ басталған күні, аяқталған күні және жоба менеджері сақталуы тиіс.

Әрбір жоба үшін жоба жетекшісі ретінде бір қызметкер таңдалады.

Әрбір қызметкер үшін жұмыс уақытының ұзақтығы (сағат) және нақты жоба үшін белгілі бір күні орындалған жұмыстың сипаттамасы жазылады деп болжанады.

- кесте байланыстарының графикалық көрінісін ER-диаграмма арқылы құрастырыңыз және кестелер арасындағы байланыс түрін көрсетіңіз.

- b) Жобаларға қанша қызметкер қатысады? Жобаның нөмірі мен атауын, сондай-ақ қызметкерлер санын шығаратын SQL-сұрау жасаңыз.

4-тапсырма

Ұялы байланыс операторының деректер қорының келесі арақатынасы (кестелері) берілді..

Клиенттер	Шарттар	Сөйлесулер	Тарифтер
клиент_№	келісімшарт_№	сөйлесу_№	тариф_№
клиент_тегі	№ клиент_келісімшарты	№ келісімшарт_сөйл	тариф_минуттар
клиент_аты	келісімшарт_арт	сөйл_мақстатты_№	
клиент_көшесі	келісімшарт_басталуы	сөйл_күні	
клиент_индексі	келісімшарт_аяқталуы	сөйл_басталуы	
клиент_туған күні	келісімшарт_бағасы	сөйл_ұзақтығы (мин.)	
		сөйл_тарифі №	

Келесі тапсырмаларды орындайтын SQL сұрауларын жасаңыз:

- № 34890 қандай клиент әңгіме өткізді?
- Тапсырыс беруші Герман Майер 25.03.2015 ж.өткізген барлық сөйлесулер туралы ақпарат шығарылуы тиіс.
- қандай келісімшарт (клиенттің атымен беру) келесімен аяқталады?
- әр клиенттің әңгіме саны туралы ақпаратты шығару.
- «HAVING» SQL-операторы не істейді ...»(мысал)?

5-тапсырма:

Дәрігерлердің бірлескен тәжірибесі пациенттерге онлайн дәрігердің қабылдауына жазылуды ұсынады. Бұдан басқа, келесі қатынастарды қамтитын деректер қорын әзірлеу қажет:

Дәрігер (дәрігер №, аты, мамандығы, телефон №)

Қабылдау (қабылдау №, дәрігердің №, күні, басталу уақыты, аяқталу уақыты, пациенттің №)

Пациент (пациенттің №, тегі, туған күні)

- бастапқы кілт өрісін және сыртқы кілт өрісін белгілеңіз.
- дәрігер көмекшісіне (фельдшерге) бүгінгі күні дәрігер Галленбиттерде хронологиялық тәртіппен барлық қабылдаулардың тізімі қажет.

Қажетті тізім үшін SQL сұрауын реттеңіз.

- дәрігер Мюллер (дәрігер № =2) 17 маусым күні 16: 30

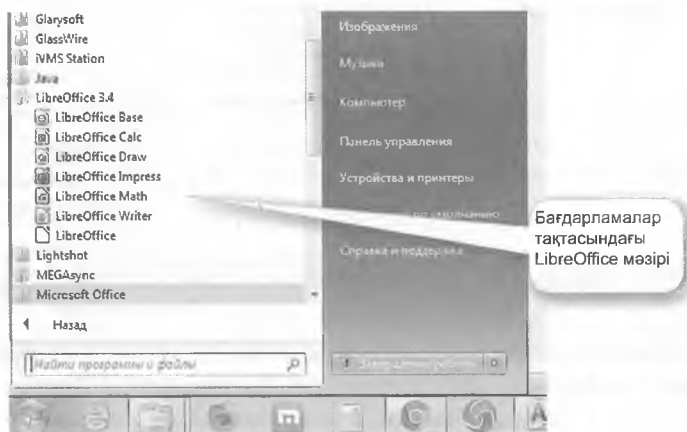
Бұл жазбаны енгізетін SQL нұсқауын жасаңыз.

(қабылдау № автоматты түрде деректер қорын басқару жүйесімен беріледі.)

7 LibreOffice Base

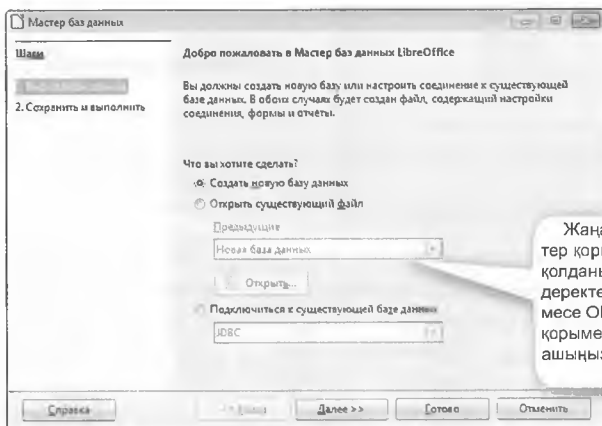
7.1 ДЕРЕКТЕР ҚОРЫН ҚҰРУ

LibreOffice Base деректер қорын жасауға және басқаруға мүмкіндік береді.

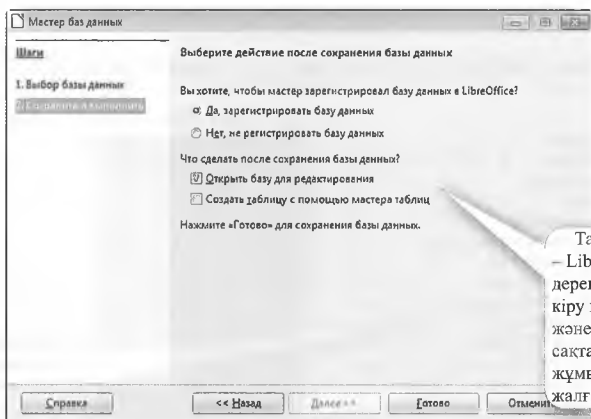


LibreOffice Base іске қосылғанда дерекқор шебері іске қосылады. Дерекқор шебері терезесінде келесі опциялар бар:

- Жаңа деректер базасын құру,
- Қолданыстағы деректер қорын ашу немесе
- Қолданыстағы деректер қорына қосылым жасау.

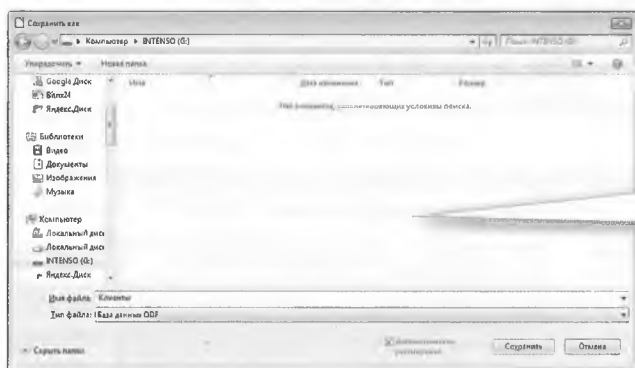


Жаңа деректер қорын Құруды таңдап және Ары қарайды басу арқылы Сіз Аяқтау және Жалғастыру таңдауының 2-терезесіне өтесіз:



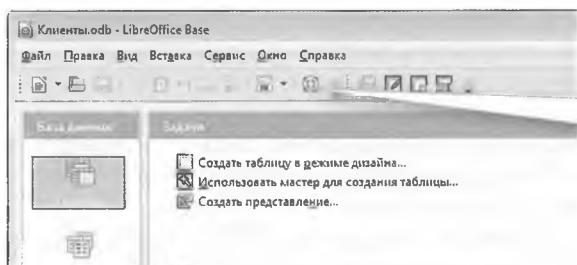
Таңдау мүмкіндігі – LibreOffice деректер қорына кіру немесе жоқ, және деректер қорын сақтағаннан кейін жұмысты қалай жалғастыру керек.

Дайын батырмасын басқаннан кейін деректер қорының аты және оның орналасқан жері көрсетіледі. Бұл мысалда Клиент файлының аты бар деректер базасы көрсетілген. Odbc деректер қоры үшін әдепкі файл түрі.



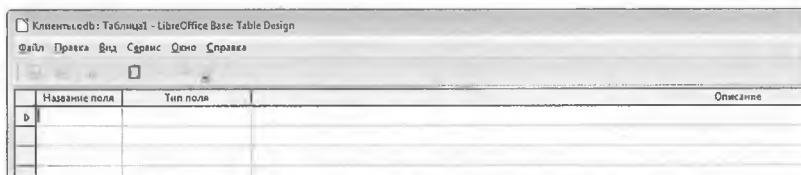
Сақтау жолын көрсетіңіз және файл атауын енгізіңіз.

Деректер қорының терезесі сол жақта кестені, сұрауларды, есеп нысандарын таңдау белгішелері бар деректер қорының аумағын қамтиды. Оң жақ панельде есептер бөлімінде Конструктор режимінде кесте жасау мәзірі тармақтары бар ..., Шебердің көмегімен кестені жасау ... және көрініс жасау ... Құтқару шеңбері бар белгішені басқан кезде, Интернетке қосылған кезде онлайн-анықтама жүктеледі.



Құтқару сақинасына басу онлайн-анықтаманы іске қосады

Түймені басқанда, Конструкторда кесте жасау керек... сонымен қатар, бұл терезеде кесте құрылымы, яғни атрибуттар үшін өрістер аттары, деректер типі үшін өрістер түрі және қажет болған жағдайда сипаттама жасалады.



Өріс атауын (мысалы, индекс) енгізгеннен кейін, ашылмалы мәзір арқылы «өріс түрі» өрісінде кептеген деректер түрлерін таңдауға болады.

Деректер түрі ретінде келесі таңдау опциялары бар:

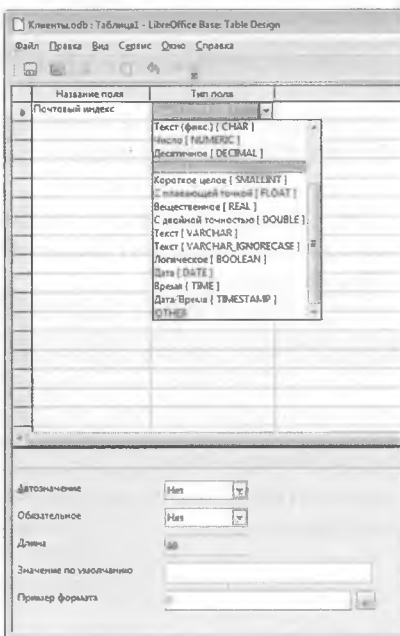
Сандық өрістер

Бүтін сандар

Tiny Integer [TINYINT], 1-байт бүтін сан -128-127-ге дейін, сондай-ақ Byte деп аталады.

Small Integer [SMALLINT], 2-байттық бүтін сан -32 768-ден 32 767-ге дейін, Integer деп аталады.

Жалпы [INTEGER], 2 147 483 647-ден 2 147 483 647-ге дейінгі ауқымдағы 4-байттық бүтін мәндер, сондай-ақ Long Integer деп аталады.



Bigint [BIGINT], 8 - байт бүтін сан + / - 9,2 триллион мәндер ауқымында жиі проприетарлық деректер қоры бағдарламаларында қолдау көрсетілмейді.

Ондық бөлшек

[DECIMAL], ондық бөлшек 10 санға дейін, ондық нәтижелер ондық разрядтардың көрсетілген санына дейін дөңгелектенеді, әсіресе ақша сомасы жағдайында, сондай-ақ Валюта деп аталады.

[NUMERIC] саны, 646 456,993 сандар мен 32 767 ондық разрядтары бар ондық бөлшек, әсіресе үлкен ақша үшін қолайлы.

Өзгермелі үтірлі сандар

[FLOAT], өзгермелі үтірмен Сан, үтірлі кейін (17) және сандарды көрсетуге мүмкіндік береді.

[REAL], 4 - + / -3,4E38 -1,40 E-45 мәндерінің диапазонында қалқымалы үтірлі байт саны, сондай-ақ Single ретінде белгіленеді.

[DOUBLE], 8 - + / -1, 80E308-4,94 E-324-ден 4,94 E-324-ге дейінгі мәндер диапазонында қалқымалы үтірмен, сондай-ақ Double ретінде белгіленеді.

Мәтін өрістері

[VARCHAR], 65 534 таңбаға дейінгі Мәтін Text деп аталады.

[VARCHAR_IGNORE_CASE], 65 534 таңбаға дейінгі мәтін салыстырғанда тіркелім ескерілмейді.

[CHAR], ұзындығы 65 534 таңбаға дейінгі мәтін, енгізу әдепкі мәнге дейінгі бос орындармен толықтырылады.

[LONGVARCHAR], 2 гигабайт таңбасын сақтай алады, сондай-ақ *Мето* ретінде белгіленеді.

Егер пайдаланылатын таңбалар саны белгілі болса, ұзындығы шексіз мәтін өрістерін пайдалану керек.

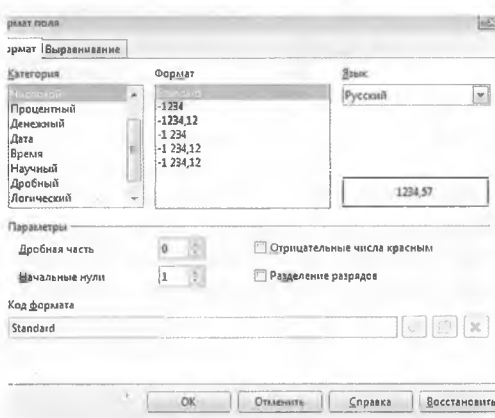
Басқа өрістер

Деректер өрісі	Екілік Өрістер
Иә / жоқ [BOOLEAN], шынайы күндер [DATE], күнді енгізу	Екілік деректер, мысалы, деректер қоры өңделмейтін суреттер немесе дыбыстық деректер.
Уақыты [TIME], уақыты/ уақыты [TIMESTAMP], күні мен уақытын енгізу.	Екілік өріс [VARBINARY], 2 гигабайт екілік өріс (fix) [BINARY], 2 гигабайт сурет [LONGVARBINARY], 2 гигабайт Other [OTHER], тек Java объектілерін сақтау үшін пайдалану керек.

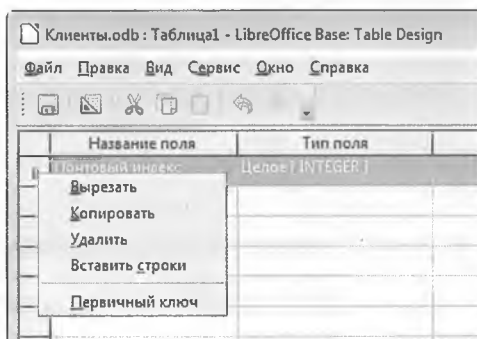
Терезенің төменгі жағында әрбір телсипат (өріс аты) үшін қосымша сипаттарды анықтауға болады. Мысалы, автоматты мән, енгізу қажет (NN = NOT NULL сәйкес келеді), ең үлкен ұзындық, өдепкі мән және пішімдеу пішім үлгісі арқылы көрсетілуі мүмкін.



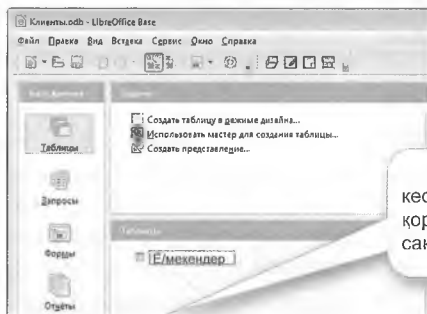
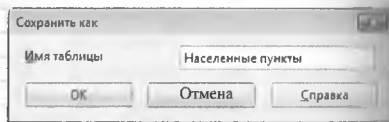
Пішім үлгісі бөлімінде гүймені басы арқылы дайын өрістерді пішімдеуді таңдауға болады Формат бойынша ұсыныстар санаттарда таңдап алынады, дайын форматтар ұсынылады. Өрісті одан өрі пішімдеу «Параметрлер» белімінен таңдауға болады. Өз пішімдеріңізді жасай аласыз.



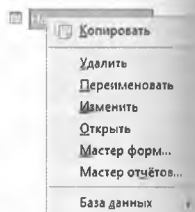
Өріс жолы бойынша гінтуірдің оң жақ батырмасын басып, мәтінмәндік мәзірдегі бастапқы кілттің қасиетін таңдауға болады. Егер бастапқы кілт өрісі көрсетілмесе, Дерекқордың бағдарламалық жасақтамасы бастапқы кілт ретінде жаңа өрісті жасау керектігін сұрайды.



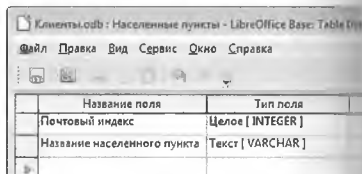
Соңғы қадам ретінде деректер қорындағы Е/мекен атымен кесте сақталады.



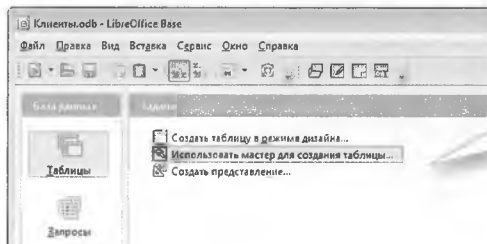
Кез келген кесте бойынша тінтуірдің оң жақ батырмасын басқанда, оны Конструктордағы түзету бөлімінде контекстік мәзірде қайта ашуға және оны тиісті түрде өзгертуге болады.



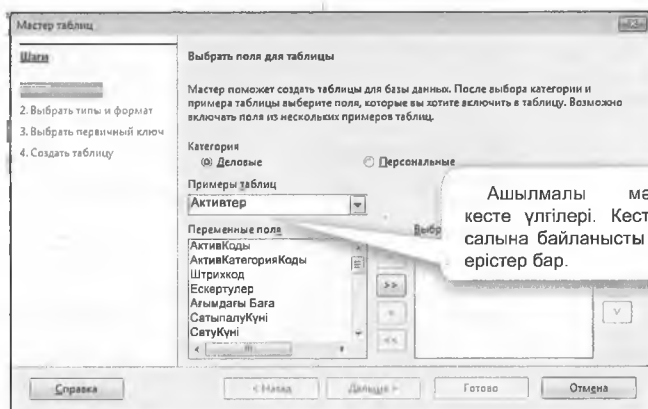
Е/мекен кестесінде басқа телсипат ретінде индекске қосымша Е/мекен атауы бар мәтіндік өріс қойылады. Жобаны аяқтау үшін сақтау белгішесін басыңыз.



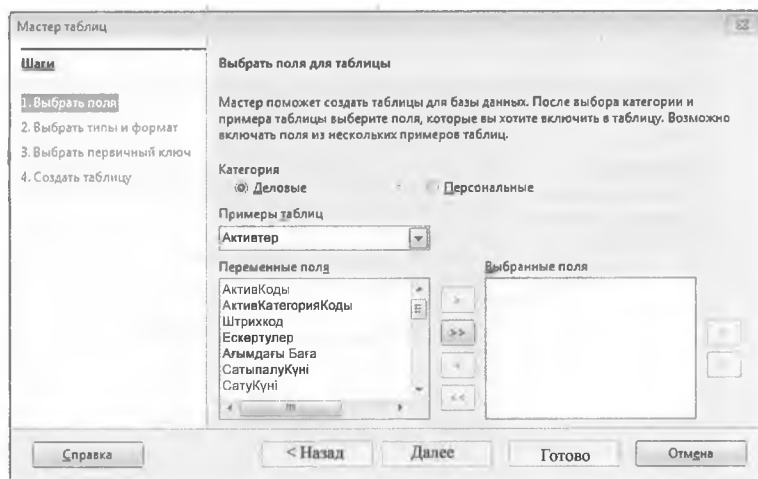
Жаңа кесте клиенттер енді кесте жасау шебері арқылы жасалуы керек.



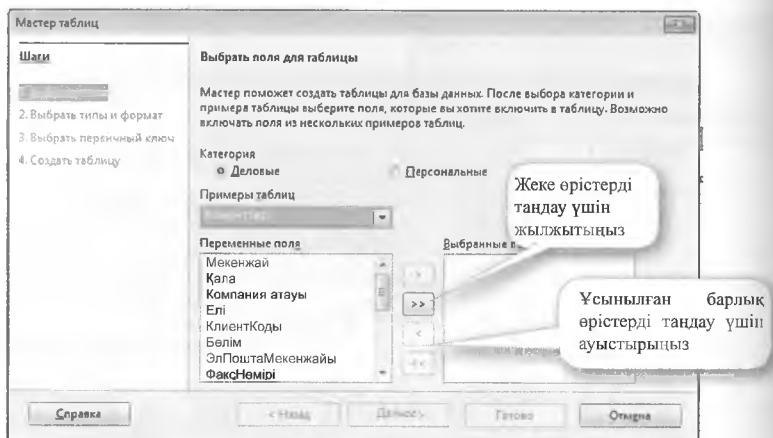
Кесте жасау шеберінің көмегімен кесте төрт қадамда жасалады.



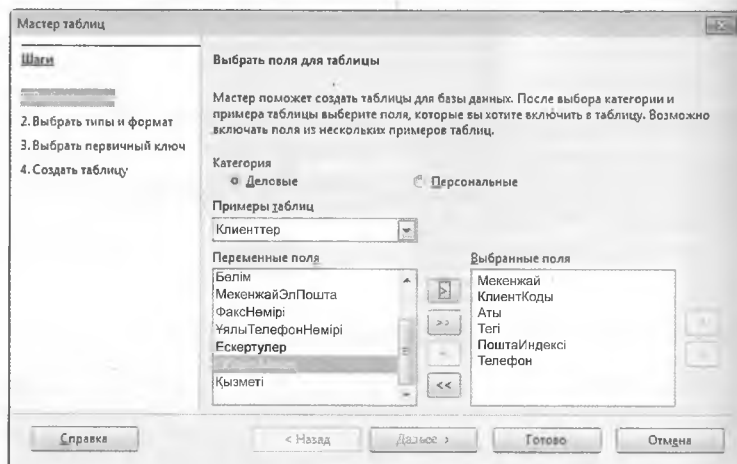
Таңдау батырмаларымен іске қосылатын Бизнес және жеке сәттарында өрістер атаулары бар көптеген үлгілер бар. Кесте үлгілерінің ашылмалы мезірінде клиенттер таңдалады.



Кесте шебері таңдалған кесте үшін қолайлы болуы мүмкін өрістерді ұсынады. Өрістер екі рет басу немесе басу арқылы, содан кейін нұсқар батырманы басу арқылы оң жақ терезеде пайда болады. Барлық ұсынылған өрістерді қос нұсқар пернесі көмегімен таңдалған өрістерге ауыстыруға болады.

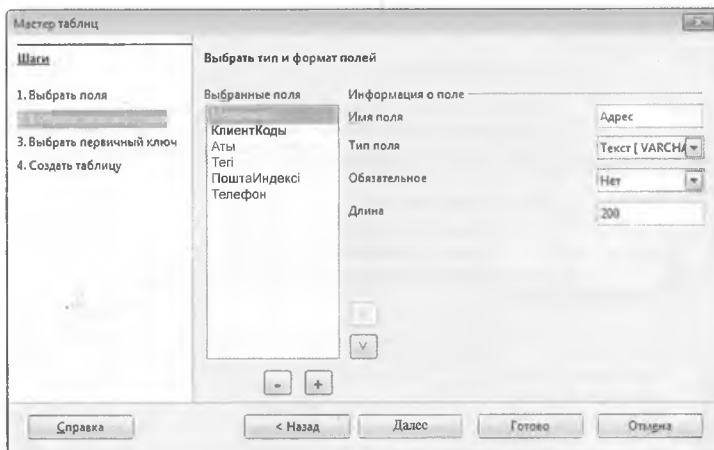


Жоғарыда сипатталғандай, мекенжай, клиенттің №, аты, тегі, пошта индексі және телефон нөмірі өрістері дәйекті түрде таңдалады. Одан әрі батырмасын басып, 2 деректер түрі өрісін таңдау экранына түседі.

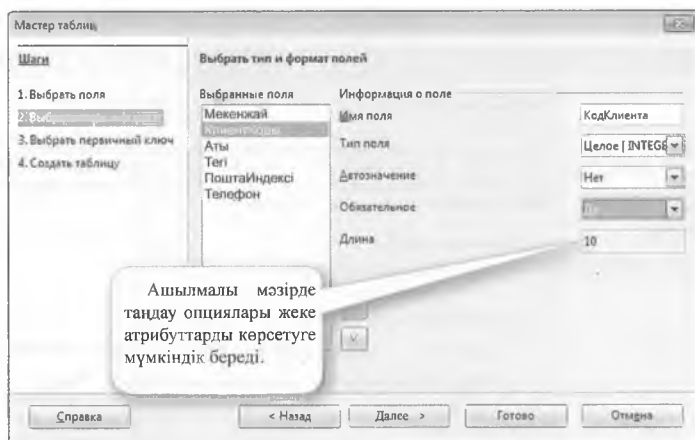


Енді, әрбір төлсіпат (өріс аты) үшін Y/N енгізуді талап ететін өріс деректерінің түрін анықтауға болады.

(NN нөл емес) және жеке өрістің ұзындығына сәйкес келеді.

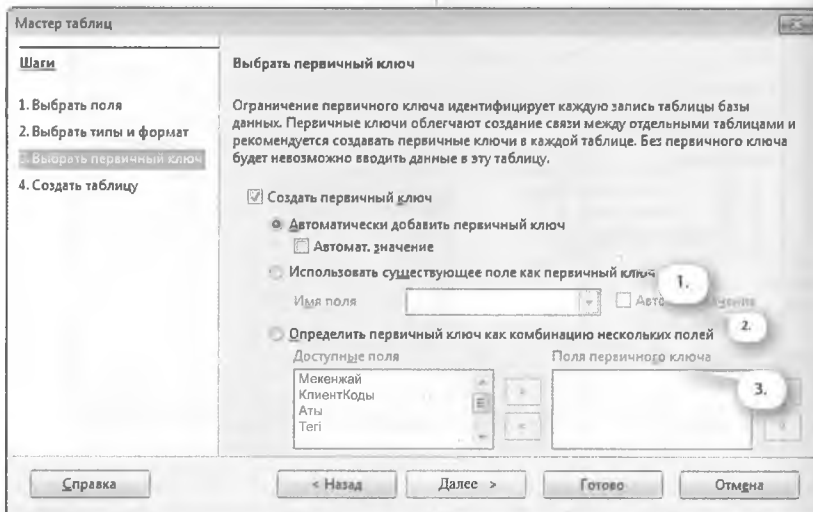


Енді сіз, мысалы, клиенттің № өрісіне енгізу қажет мәнді орнатуға болады; бұл сондай-ақ 3 терезесінде бастапқы кілттерді орнатуға болады.

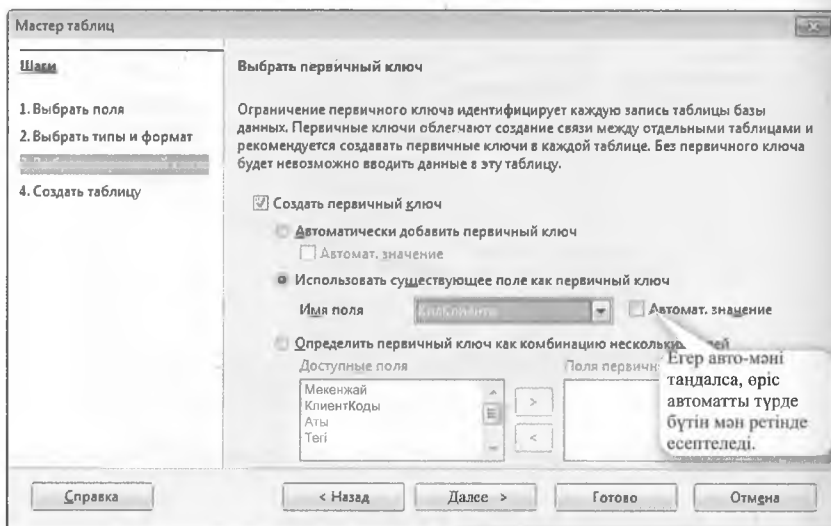


Бастапқы кілт өрісті бастапқы кілт ретінде орнатады. Кілт болуы мүмкін:

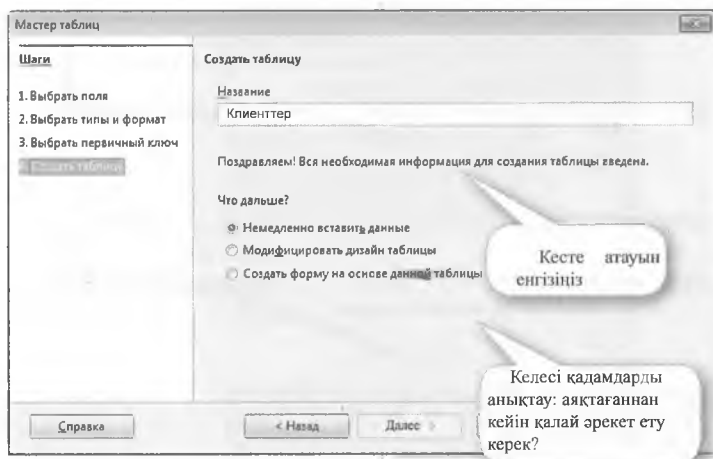
1. авто-мән ретінде автоматты түрде қосылуы,
2. қолданыстағы өріс ретінде таңдалуы, немесе
3. құрамдас бастапқы кілт түрінде бірнеше өрістен жасалған.



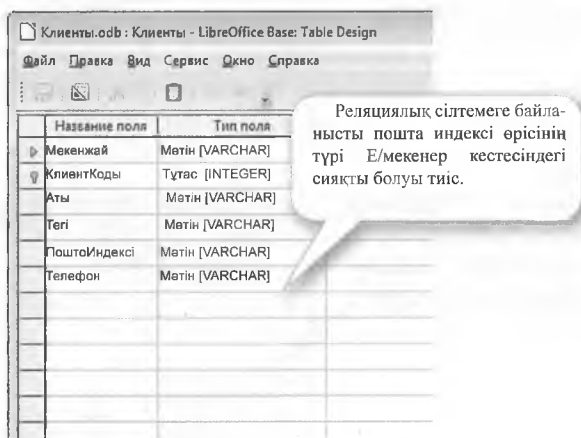
Бұл мысалда, клиент № өрісі екінші параметрмен орнатылады (бастапқы кілт ретінде бар өріс пайдаланылады).



Кесте шеберінің жұмысы аяқталғаннан кейін аты және одан кейінгі әрекеттері көрсетіледі.



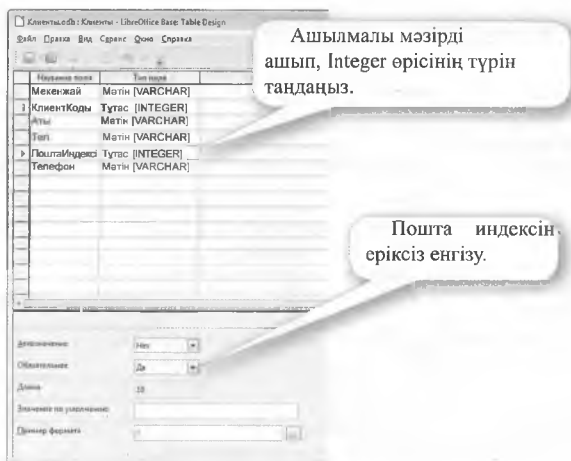
Аяқтауды басқаннан кейін клиенттер кестесі құрылды. Егер кестені өңдеу режимінде ашсаңыз, онда пошта индексі бүтін болуы керек, себебі пошта индексі екі кесте арасындағы реляциялық байланысты іске асырады. Өрістің түрін өзгерту өрістің деректер түрі 2 белімінде шеберлер терезесінде жасауға болады.



Ескерту:

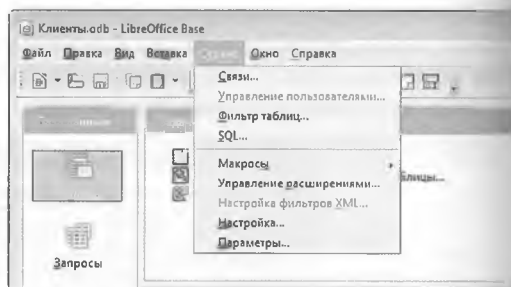
Реляциялық байланысқан негізгі белдіктер өрісінің түрі (бастапқы кілт-сыртқы кілт) бірдей болуы тиіс. Терминдері өріс түрі және өріс деректер түрі LibreOffice бірдей ұғымды білдіреді.

Параметрді таңдау арқылы мән бар, индексті мәжбүрлеп енгізу қажет.

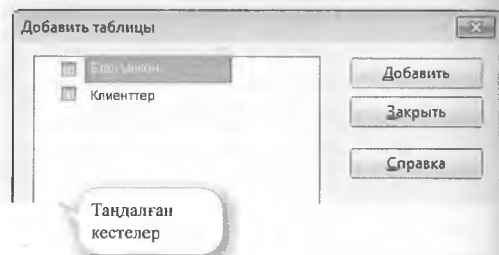


7.2 КЕСТЕЛЕР АРАСЫНДА БАЙЛАНЫС ЖАСАУ

Қосымша Байланыстар бөлімінде жасалған екі кесте арасында реляциялық байланыстарды орнатуға болады.



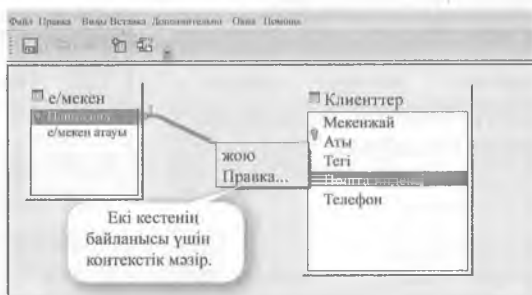
Кестелерді басыңыз және қосу немесе кестелердің аттары бойынша екі рет басу арқылы таңдаңыз.



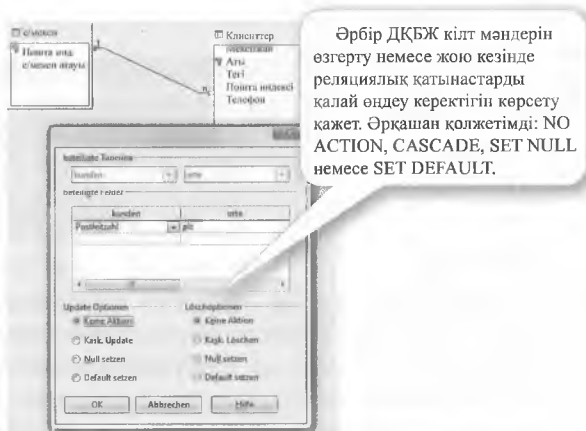
Тінтуірдің сол жақ батырмасын басқанда (сүйреп апарыңыз) элементті Е/мекеннен жылжытыңыз. клиенттер индексі. индекс және қайтадан клиенттерге тінтуір түймешігін босатыңыз. Кесте мен байланыс қуаты арасындағы байланыс желісінің индексі (1: n) автоматты түрде қосылады. Бастапқы кілт ерісі бар кесте әрқашан 1 қуатын алады.



Тінтуірдің оң жақ батырмасын басып, контекстік мәзірде жою немесе өңдеу пункті пайда болады.



Өзгерту таңдасаңыз..., сіз іске қосылған кестелер мен өрістерді көресіз. Сонымен қатар, жаңарту параметрлерін (ON UPDATE) және жою параметрлерін (ON DELETE) орнатуға болады.



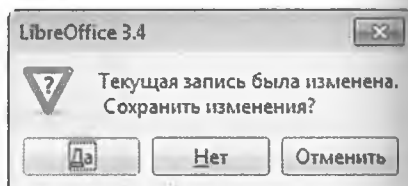
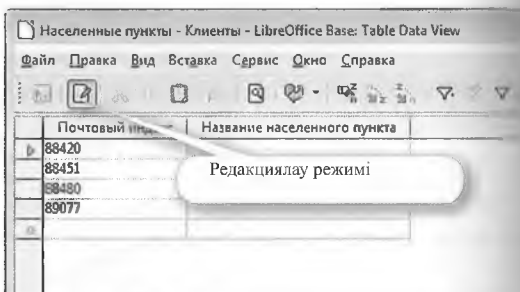
Енгізу арқылы деректер қорының мерзімін ДБЖ-ның негізгі байланысты шын мәнінде орнатқанын тексеру қажет.

7.3 ДЕРЕКТЕР ҚОРЫНЫҢ ЖОЛДАРЫН ЖАЗУ

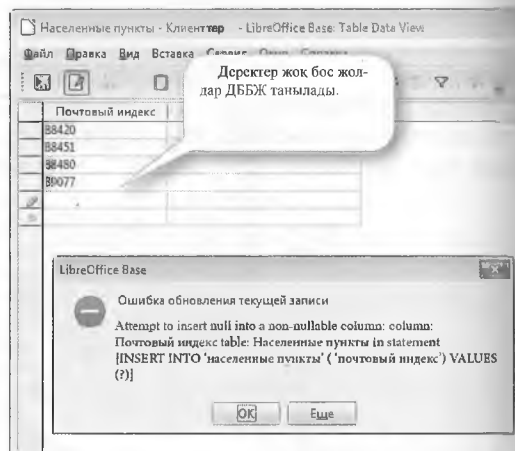
Өңдеу үшін оны ашу үшін `N/тармақтар` кестесін екі рет басыңыз.

Енді кез келген жазбаларды енгізуге болады. Қате жазбалар, мысалы, индексті жазу мәтіндік деректер ретінде ДББЖ ұсталады және әдепкі мәнмен ауыстырылады (мысалы, 0).

Кестені жапқан кезде ДҚБЖ деректерді сақтау керектігін сұрайды.

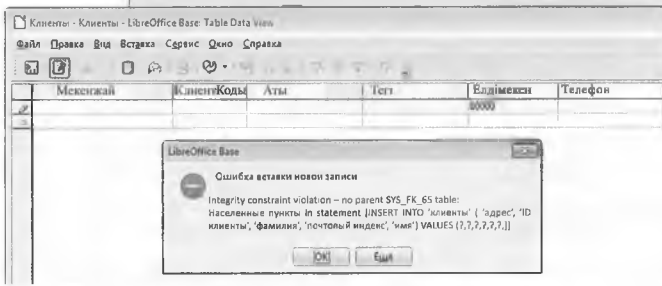


Соңғы жолға ешқандай деректер енгізілмегендіктен, бірақ ерісі `NN = NOT NULL` сипатының индексіні білдіреді, қате туралы хабарлама осыны көрсетеді. Бос жолды жою қатені жоюға және сақтауға мүмкіндік береді.



Деректер жазбаларын негізгі Индекс жазбасының кестесіне өнгізгеннен кейін еншілес кестеге клиенттер енгізіледі. Алайда сілтемелік тұтастықты қамтамасыз ету үшін сілтемелік кестеде Индекс бар Пошта индекстері ғана тағайындалуы мүмкін.

Кестеге деректерді Еркін жазу енгізілгеннен кейін, сілтеме тұтастығына қайшы келетін пошта индексі (**Integrity constraint violation**), қате туралы хабарламамен расталады (**no parent FK = foreign key**).

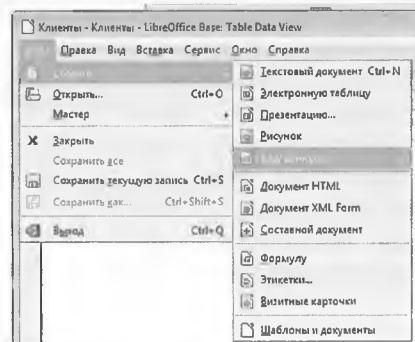


Пошта индексін нақты қолданыстағы, мысалы, 89077-ге өзгерту деректерді сақтауға мүмкіндік береді.

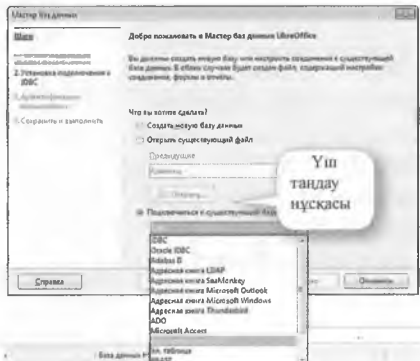
7.4 БАСҚА ДЕРЕКТЕР ҚОРЫНА ҚОСЫЛУ

Access деректер қорына қосылу

Жеке деректер қорын жасауға қосымша, LibreOffice Base басқа қолданбаларда жасалған деректер қорына қосылуға мүмкіндік береді. Жылдам кіру панелін немесе Файл – Жаңа немесе жаңа гүймешіктер мәзірін пайдаланып, Дерекқордың құжат түрі қолданылады.

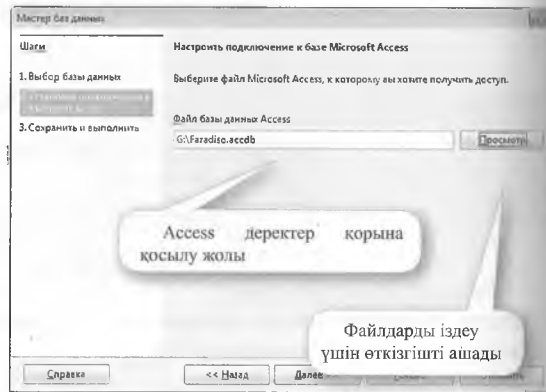


Дерекқор шебері таңдаудың үш түрлі нұсқасын ұсынады. Бірінші қадам – қолданыстағы деректер қорына қосылу. Ашылған мәзір белгілі деректер қоры жүйелері үшін көптеген интерфейстерді ұсынады. Microsoft Access-ке басу кезінде деректер қорымен оған сақтау жолдарын көрсету арқылы байланыс орнатылады.

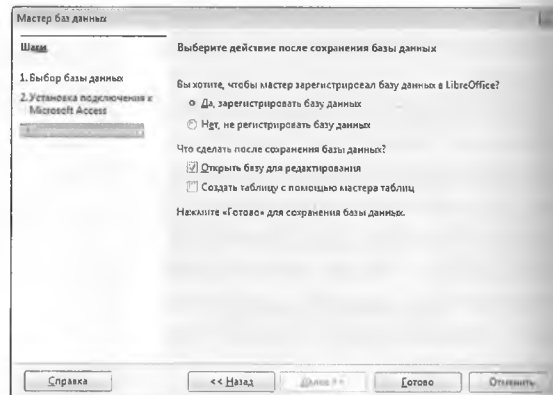


ДКБЖ белгілі интерфейстері.

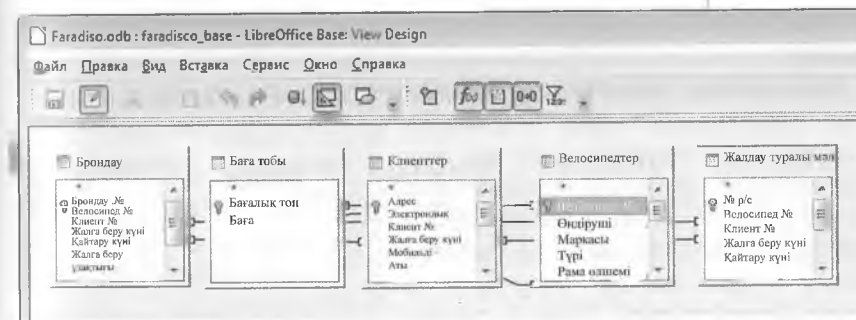
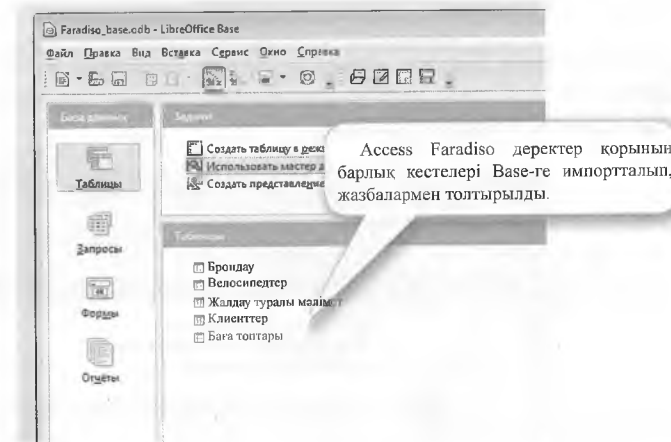
«Бұдан әрі» батырмасын басқаннан кейін деректер қорының шебері қосылуы қажет Access деректер қорына жол сұрайды. Шолу опциясы арқылы сіз ДК-де өткізгіш бағдарламасының көмегімен ұқсас жолмен ДБ жолын таба аласыз.



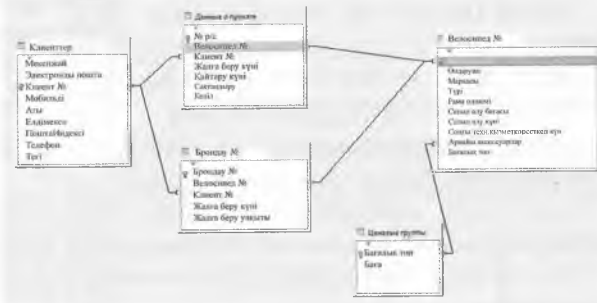
Үшінші және соңғы қадамда сіз «Иә», деректер қорын қосуды таңдайсыз. Кестелерді жасау шеберін қолданбау, өңдеу үшін деректер қорын ашу ұсынылады. Бос деректер қорына қосылу енді кез келген жерде сақталуы мүмкін.



Файл атауын енгізгеннен кейін, мысалы, faradiso_base және Saқтау батырмасын басқаннан кейін, сіз деректер қоры құжат терезесінде боласыз.

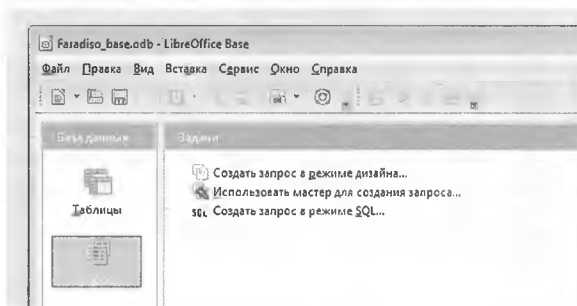


Сондай-ақ, байланыстар импорт арқылы дұрыс тасымалданады, мысалы, Қосымша - > Байланыс. Кестелер реттеліп тұруы керек, мысалы, Клиенттер сол жақта, Велосипедтер оң жақта, қалған кестелер ортада.



7.5 СҰРАНЫСТАР ЖАСАУ

Сұраныстарға басқан кезде сұраныс терезесі ашылады. Мұнда сұраныстарды **Конструктор** режимінде, **Шебер** немесе тікелей **SQL** коды арқылы жасауға болады.



Конструктор режимінде сұраныс жасау

Конструктор режимінде сұраныстар графикалық түрде жасалады. Сұраныстар кестелер үшін де, бар сұраныстар үшін де орындалуы мүмкін. Кесте аттарын таңдап, Қосу батырмасын басу арқылы немесе кесте атауын екі рет басу арқылы тиісті атрибуттарды төменгі таңдау терезесіне апарыңыз.

Таңдау терезесінде келесі жолдар бар:

Лақап аты – бұл өріс сұраныс өрісінің атынан ерекшеленетін атау.

Кесте: мақсатты кестені анықтау, мысалы, атау өрісі бірнеше кестелерде бар болса, әсіресе пайдалы.

Сұрыптау: ascending (ascending) немесе төмендеу (Descending) сұрау нәтижелерін сұрыптау параметрлерін анықтайды.

Көрінетін: белгіленген жалауша жоқ, бұл сұраныс нәтижесінде шықпайды.

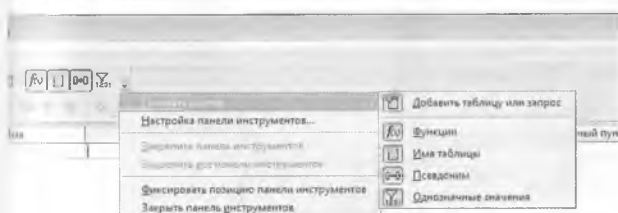
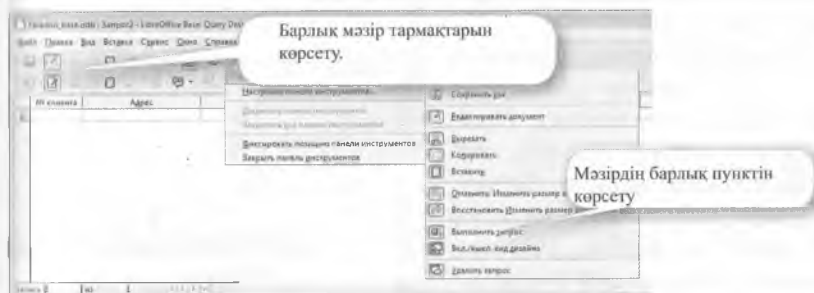
Функциясы: есеп үшін функцияларды ұсынады, мысалы, SUM (сума).

Критерий: сұрау үшін сүзу критерийлерін анықтайды.

Немесе: жеке критерийлердің НЕМЕСЕ қосымшаларын салуға мүмкіндік береді.

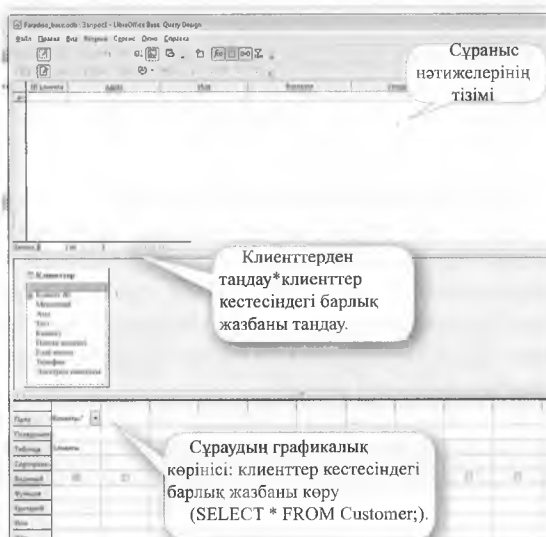


Екі рет басу немесе кесте бойынша шертіп, кейін оны қосу арқылы сұраныста пайдаланылатын кестелерді таңдаңыз.



Керінетін батырмаларды басқан кезде барлық қолжетімді пәрмендер көрсетіледі.

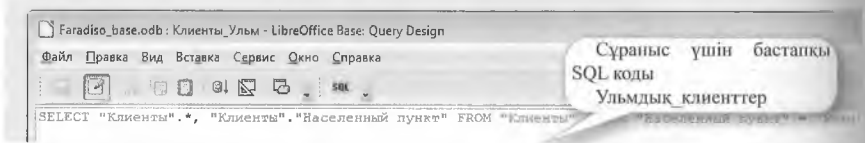
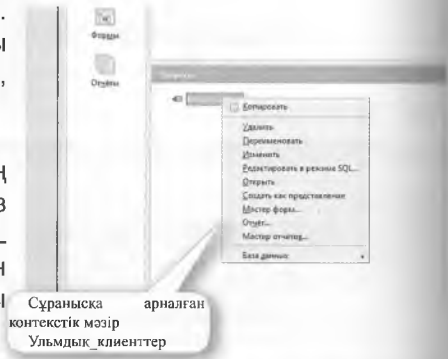
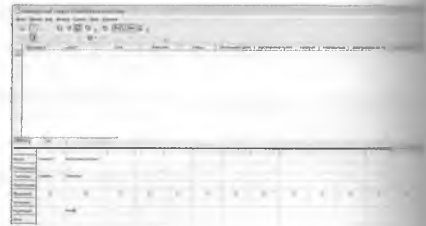
Бірінші сұраныста барлық клиенттердің барлық мәліметтері ұсынылуы тиіс. Жұлдызшаны екі рет басыңыз, Клиенттер* ұяшықтар өрісіне қосылады. Сұрау F5 пернесін басу немесе сұрау салу батырмасын басу арқылы орындалады. Кестеде барлық клиенттер көрсетілген.



Егер, мысалы, Ulm (Ульм қ.) Е/ мекен клиенттерін ғана көрсету қажет болса, онда сұранысқа критерий қосылуы тиіс.

Жол өрісінің екінші бағанына және Ulm өлшеміне Н/ пункт енгізу сұрау нәтижесінде Ульм қаласынан келген барлық клиенттер туралы ақпарат береді. Енді бұл сұраныс сақталуы және жабылуы мүмкін, мысалы, Ульмалық_клиент деп аталады.

Сұраныс бойынша тінтуірдің оң жақ батырмасын басыңыз мәтінмәндік мәзірді ашады. SQL көрінісін өзгерту таңдалғаннан кейін SQL бастапқы коды көрсетіледі.



Сұраныс үшін бастапқы SQL коды Ульмдық клиенттер

Сүзу шарттарын қалыптастыру

Сүзу шарттарын қалыптастыру үшін әртүрлі операторлар мен командалар қолданылады. Сонымен қатар, деректер қоры өрістерінің мазмұнын сұрау үшін SQL-ге тән командалар бар. Бұл командалар Конструктор режимінде графикалық түрде жасалады және SQL синтаксисіне автоматты түрде тасымалданады. SQL командалары дәл тікелей енгізілуі мүмкін.

Келесі кестелерде операторлар мен командаларға шолу ұсынылған:

Оператор	Мәні	Шарт орындалған, егер...
=	тең	... өрістің мазмұны көрсетілген еріске сәйкес келеді. Оператор сұрау өрісінде көрсетілмейді. Егер операторсыз мәнді енгізсеңіз, операторы болжанады.
<>	Тең емес	... өрістің мазмұны көрсетілген мазмұнға сәйкес келмейді.
>	көп	... өрістің мазмұны көрсетілген мазмұннан көп.
<	аз	... өрістің мазмұны көрсетілген мазмұннан аз.
>=	Көп немесе тең	... өрістің мазмұны көрсетілген мазмұннан көп немесе тең.
<=	Аз немесе тең	... өрістің мазмұны көрсетілген мазмұннан аз немесе тең.

OpenOffice.org Команда	SQL- Команда	Мәні	Шарт орындалды, егер...
БҮС	IS NULL	бос	... деректер өрісі бос. Иә/жоқ – үш жай-күйі бар өрістер үшін бұл команда белгісіз күй сұратады (иә, жоқ).
БҮС ЕМЕС	IS NOT NULL	Бос емес	... деректер өрісі бос емес.
БЫЛАЙША (Толтырғыш * таңбалардың көз келген саны үшін Толтырғыш (плейс-холдер ? бір белгі үшін)	LIKE (Кез келген таңбалар үшін % қою белгісі дәл бір белгі үшін плейс-холдер)	Құрамдас белігі болып табылады	... деректер өрісінде көрсетілген өрнек бар. Қою белгілері (*) х өрнегінің ба-сында (x*), соңында (*x) немесе өріс мазмұнының ішінде (*x*) кездесетінін көрсетеді. Толтырғыштар ретінде SQL% таңбасын SQL сұраныстарында енгізуге болады. Интерфейсте OpenOffice.org әдеттегі файлдық жүйе (*) белгілерін пайдалануға болады. Қою белгісі * немесе % таңбалардың кез келген санын білдіреді. Сұрақ белгісі (?) Интерфейстегі бір таңба үшін ғана қолданылады. OpenOffice.org немесе астын сызу () SQL-сұраныстарда толтырғыш ретінде.
ЕМЕС	NOT LIKE	Белігі емес	... деректер өрісінде көрсетілген мазмұн жоқ.
X және Y аралығында	BETWEEN	[x,y] интервалында орналасқан	... деректер өрісінде екі X және Y мәндерінің арасындағы мән бар.
X және Y аралығында емес	NOT BETWEEN x AND y	[x, y] интервалында орналаспаған	...деректер өрісі X және y мәндерінің аралығында жатқан мәнді қамтиды.
B (a; b; c...) Барлық мәндергізіміндегі белгіш ретінде үтір нүктесіне назар аударыңыз!	B (a, b,	a, b, c... тұрады	... деректер жолағында көрсетілгендердің бірі бар ... Кез келген өрнектер санын көрсетуге болады, сұрау нәтижесі НЕМЕСЕ операциямен анықталады. A, b, c өрнектері ... сандар да, символдар да болуы мүмкін.
НЕ B (a; b; c...)	NOT IN (a, b, c...)	a, b, c...тұрмайды	... деректер өрісінде A, b, c, берілген өрнектердің бірі жоқ
= АҚИҚАТ	= TRUE	True мәніне ие	... деректер өрісінде True мәні бар.
= ЖАЛҒАН	= FALSE	False мәніне ие	... деректер өрісінде False мәні бар.

Мысалдар

"Әйел"	"әйел" өрісі бар деректер өрісі.
WIE 'H?llo'	"Hallo" немесе "Hello" сияқты өрістері бар деректер өрісі.
КАК 'S'	"SUN" сияқты өріс мазмұнындағы деректер өрісі.
10-нан 20-ға дейін	10 және 20 мәндері арасындағы өрістің мазмұнындағы деректер өрісі. (Бұл мәтіндік және сандық өрістер болуы мүмкін).
IN (1; 3; 5; 7)	1, 3, 5, 7 мәндері бар деректер өрісі. Мысалы, егер деректер өрісінде мақала нөмірі болса, Сіз көрсетілген нөмірдегі нақты мақалаларды қайтаратын сұрау жасай аласыз.
НЕ ('Müller')	"Müller" жоқ деректер өрісі.

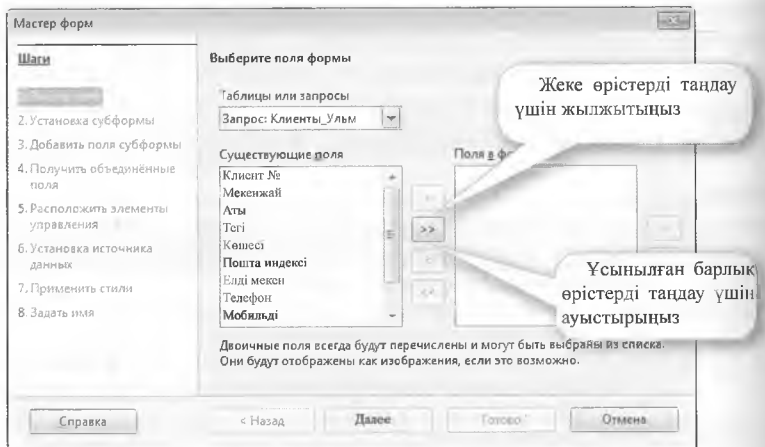
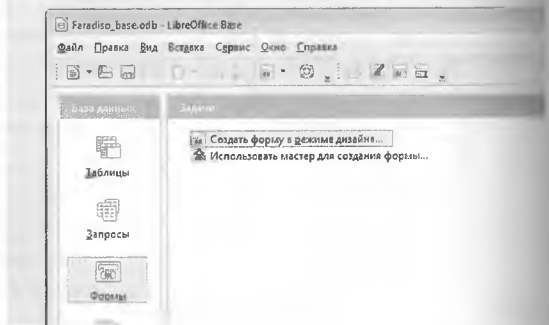
7.6 ҚАЛЫПТАР

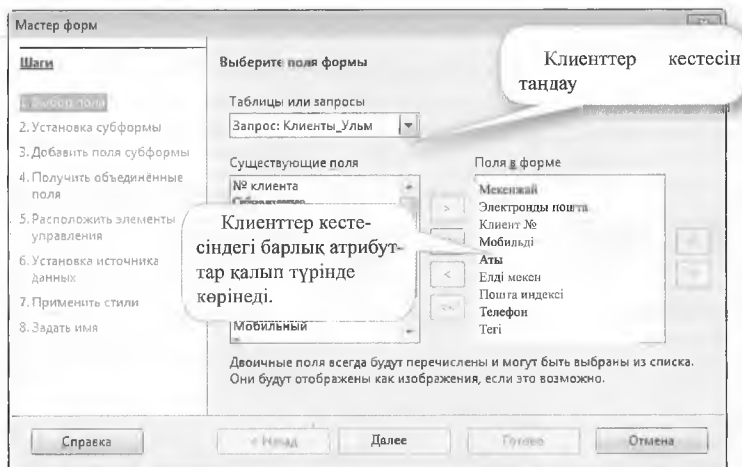
Қалып – бұл арнайы сипаттамалары бар мәтіндік құжат. Ол енгізу өрістері, түймелер, басқару элементтері немесе тізімдер сияқты болуы мүмкін. Sotyre мәзіріндегі элементті басқаннан кейін, пішін жасуу немесе шебер арқылы таңдауға болады.

Шебер енді клиент деректерін енгізу үшін біртіндеп қалып жасайды.

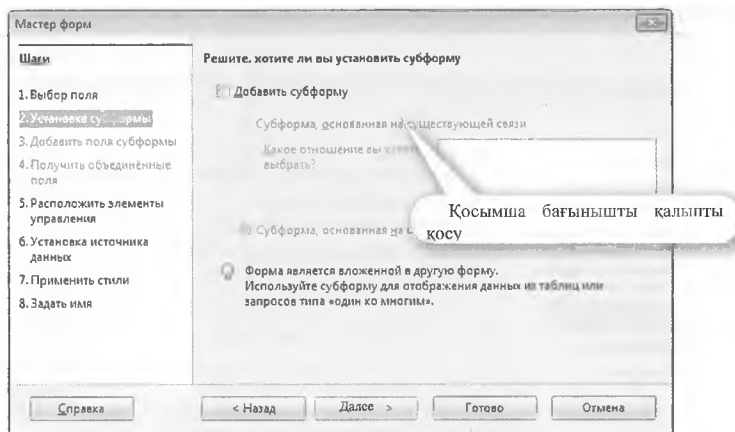
Деректер қоры терезесінің сол жағында форманың мәзірінің пункті қолжетімді. Тапсырмалар батырмасын басқаннан кейін, сіз Конструкторда қалып жасай аласыз және қалып жасау шеберінің көмегімен Қалып таңдау аласыз.

Қалып шеберінде алдымен тиісті кестелер немесе сұраулар, содан кейін олар бар өрістер таңдалады. Бұл көрсеткі түймені немесе қос көрсеткі батырманы басу арқылы жасауға болады. Қос көрсеткі батырмасы арқылы барлық кесте атрибуттары клиенттер қалыпқа көшіріледі.

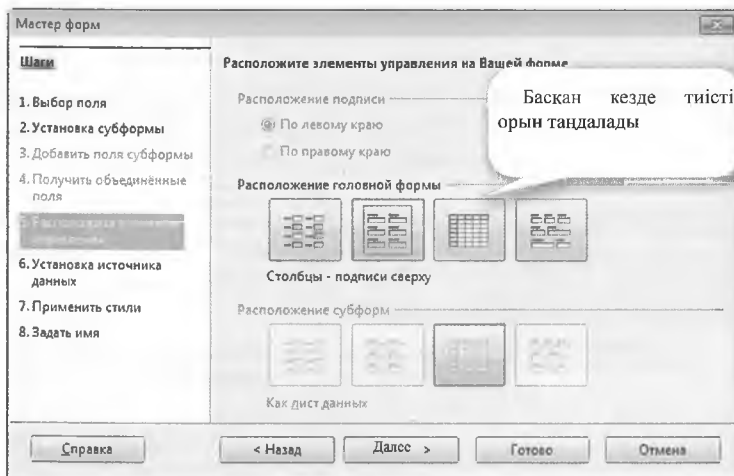




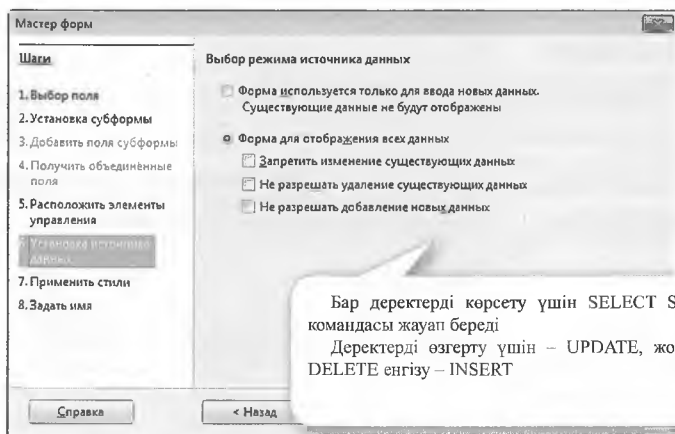
Екінші қадамда міндетті емес бағынышты қалыпты қосуға болады. Мысалы, бағынышты қалып арқылы тапсырыстың барлық байланысқан деректері әрбір клиент үшін көрсетілуі мүмкін. Іске қосылған кестелердің реляциялық байланысы қажет.



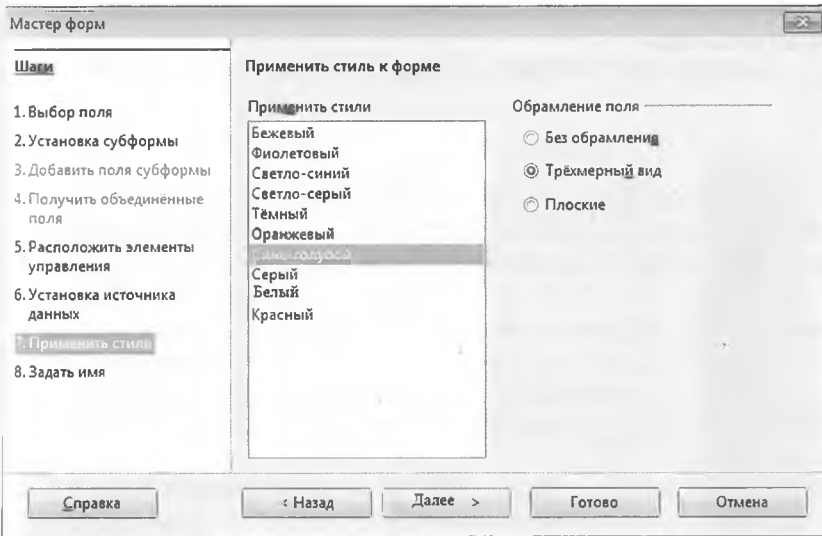
Клиент деректерін енгізу үшін бағынышты қалып талап етілмегендіктен, 3 және 4 қадамдары сұр түспен бейнеленеді және анықтау мүмкін емес. 5-қадам деректер қоры өрістерінің тәртібін анықтайды. Тәртіп кейінірек өзгертілуі мүмкін.



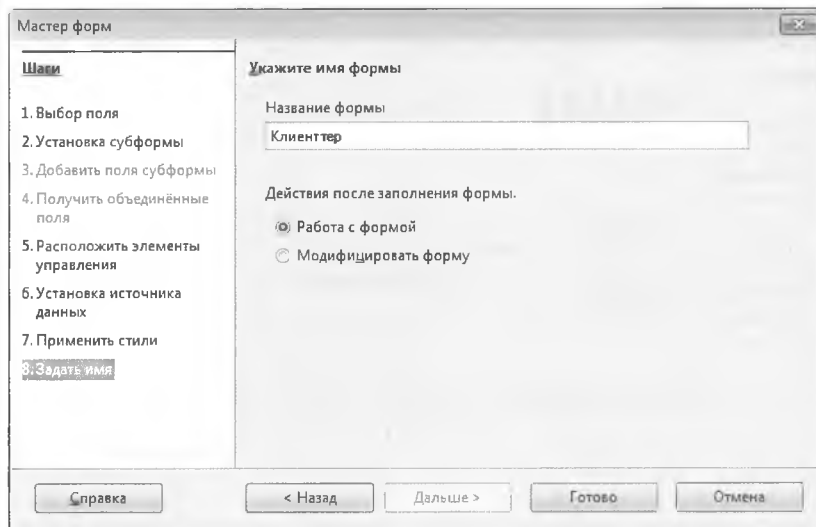
Деректерді енгізгенде, клиенттің бар деректері көрсетілетіндігі анықталады. Сондай-ақ, формадағы бар деректерді өзгерту, жою сияқты деректерді манипуляциялауға рұқсат етілген бе, сол анықталады. Осындай тәсіл арқылы бұның жаңа деректер жазбалары үшін таза енгізу нысаны болып табыла ма, әлде кең өрекеттер болуы мүмкін бе дегенді бақылауға болады.



Нәтижесінде, түрлі түсті стильдер мен түрлі шекаралар қолжетімді болады.



Қалып атауын анықтағаннан кейін, ол әдепкі деректерді енгізу үшін шылады.



Дайын қалып келесідей көрінеді, басқару тақтасы автоматты түрде қосылады

Сілтеме

Телефон

Госномер

002021465432

Электрондық пошта

Тел

Грета

Самант №

2

Әкібаева

Әлі

Мәскеу

Қала

Курорт

Почта/Қонақ

88416

Самант

Республика

Клиентке кіруге арналған қалыпты басқару панелі

Запись 1

Страница 1 / 1

Общая

СТАНД.

Конструктор режимінде форманың сыртқы түрі кейіннен өзгеруі мүмкін.

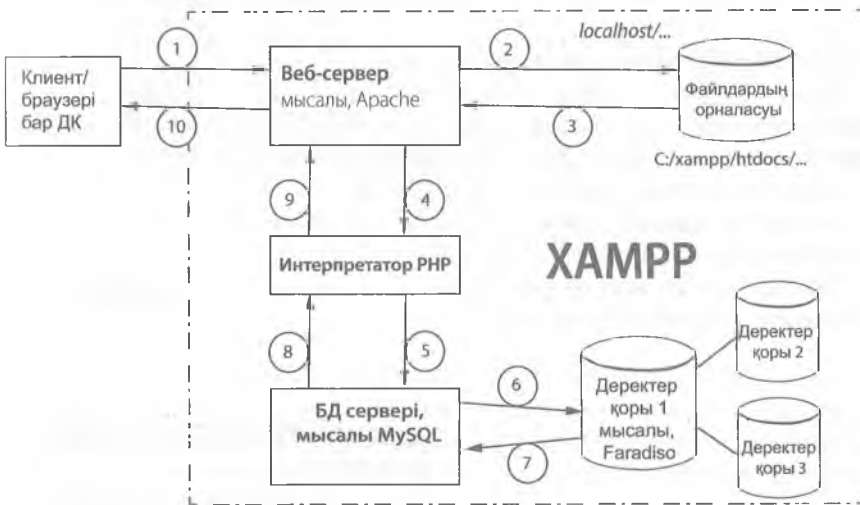
8 Интернет желісіндегі деректер қоры

8.1 ХАМРР ӨЗІРЛЕУ ОРТАСЫ

Веб деректер қоры – бұл Интернетте орналастыру және пайдалану үшін қолайлы деректер қоры. Мысалы, MySQL деректер қоры. Веб-дерекқорды әзірлеу үшін жергілікті өңдеу ортасы қажет. Ол да, мысалы, төмендегілерден тұруы мүмкін:

- Онлайн сұралған деректерді клиентке жіберетін **Apache веб-серверлерінен**
- ДҚБЖ **MySQL** (My Structured Query Language)
- және **PHP** бағдарламалау тілінен (PHP-дан гипермәтіндік препроцессор).

Бұл элементтердің барлығы **XAMPP** әзірлеу ортасында жиналған (суретті қараңыз).



8.2 КОМПОНЕНТТЕРДІҢ ЖҰМЫС ІСТЕУІ

8.2.1 Веб-сервер

Веб-сервер, мысалы, Apache жеке интернет компоненттерінің өзара әрекеттесуін ұйымдастырады, мысалы, деректер сақталатын браузер мен сервер.

Клиент сұраған URL (=ресурстың біріздендірілген көрсеткіші) веб-сервермен мекенжай ретінде сәйкестендіріледі және сақтау орнында алынады.

①Файл кеңейтімі бойынша веб-сервер бұл бет таза HTML-бет екенін немесе PHP файлдары сияқты басқа файл түрлерін сұрайтынын анықтайды. PHP кеңейтілген файлдары алдымен интерпретациялау үшін PHP бағдарламасына веб-сервермен жіберіледі ②. Онда PHP-кодтардың барлық үлестері танылады және түсіндіріледі. Егер, мысалы, деректер қорында сұраныстар болса, онда олар PHP интерпретаторымен деректер қорының жүйесіне жіберіледі ③. Деректер қорының жүйесі SQL операторлары деректер қорында ④ және ⑤ орындайды және деректерді PHP интерпретаторына жібереді ⑥. Бұл нәтижелер (мысалы, SQL сұрау) HTML кодына түрлендіріледі және веб-серверге ⑦ қайтарылады. Сол жерден клиент веб-браузер экранында керінетін таза HTML құжатын ⑧ алады. Mozilla Firefox.

Веб-сервер, бағдарлама-интерпретаторлар және деректер қорының серверлері міндетті түрде бір ДК-де болуы тиіс емес, ал бүкіл әлем бойынша Интернетте әртүрлі серверлерде орналасуы мүмкін.

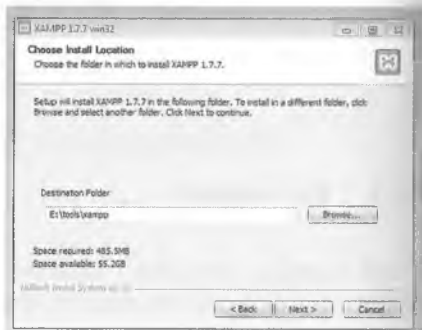
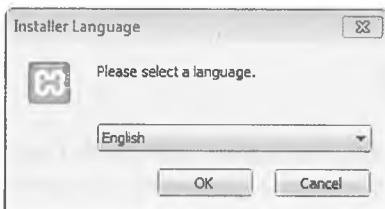
8.2.2 XAMPP әзірлеу ортасын орнату

Тегін әзірлеу ортасын жүктегеннен кейін (мысалы, [www.apachefriends.org / ru / xampp-windows.html](http://www.apachefriends.org/ru/xampp-windows.html)) XAMPP XAMPP-win32-1 файлын іске қосыңыз.x.x-VCxx-installer.exe.

Ашылатын мәзір арқылы тілді таңдаңыз:

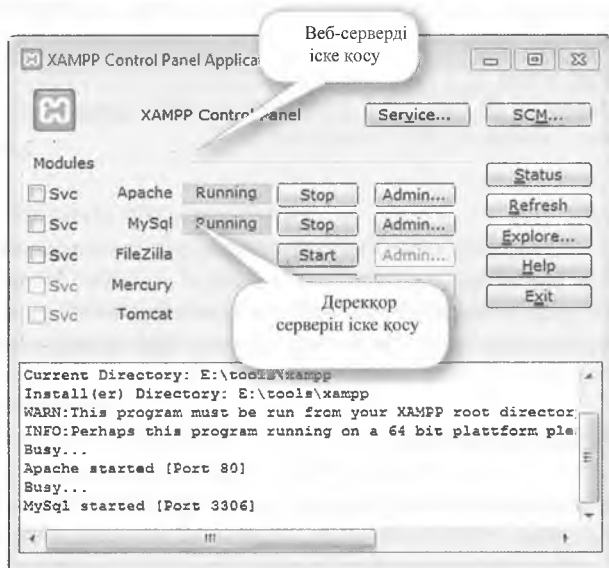
Мақсатты орнату дискі мақсатты қалтаны (каталогты) таңдау терезесінде көрсетіледі.

Осы каталогқа жол орнату үшін `setup_xampp.bat` файлын оны іске қосу үшін екі рет басыңыз.



8.2.3 Компоненттерді іске қосу

Xampp-control файлы.exe XAMPP әзірлеу ортасын іске қосады және XAMPP басқару тақтасы ашылады. Start түймесін басы Apache веб-серверін, сондай-ақ MySQL деректер қорының серверін іске қосады.



8.3 PHP СЦЕНАРИЙ ТІЛІ

8.3.1 Кіріспе

PHP – бұл сценарий тілі (scriptum = document), ол HTML кодына енгізілуі мүмкін. Сценарий тілі алдын ала компиляцияланбайды, сценарий орындалуы кезінде түсіндіріледі және орындалады. Бірақ, JavaScript қарағанда, веб-сервер бұл кодты клиентке жібермейді, оны сервер жағында орындайды.

Клиент бағдарлама нәтижесін HTML-файл түрінде алады, ол бұдан былай бастапқы PHP коды жоқ.

Ескерту:

PHP клиентке бастапқы кодты (скрипт) бермей, оны серверде түсіндіре отырып, кодты ұрлануына жол бермейді.

Клиенттің HTML-браузері Java-кодының көмегімен беттерді өңдеу үшін қолайлы және орнатылған болуы керек, PHP браузерге ешқандай талаптар қоймайды.

XAMPP әзірлеу ортасына PHP интерпретаторы қосылған. XAMPP сынақ ортасында бастапқы PHP кодын іске қосу үшін.

- XAMPP сынақ ортасында бастапқы PHP кодын орындау үшін сценарийлер C:\ішкі каталогында сақталуы тиіс... \XAMPP \ htdocs кеңейтілімі бар .php
- веб-сервер және
- скриптті шақыру IP 127.0.0.1 / бар веб-сервер арқылы жүзеге асырылуы тиіс... немесе URL local-host/...

Ескерту:

Егер test файлы болса.php тікелей браузерден жады мекенжайы бойынша оқылады, мысалы, C:\xampp \ ht-docs ..., онда PHP интерпретаторы елемейді, ал бастапқы PHP коды тікелей браузерде көрсетіледі. Деректер қорына кіру орындалмайды.

8.3.2 PHP сценарийін жазу

PHP тілдік элементтері негізінен C, Java және Perl бағдарламалау тілдерінен тұрады.

Мысал

Скрипт мәтіндік айнымалының мазмұнын көрсету керек, мысалы, «Сәлем, деректер қоры бойынша сарапшылар!».

```
<html>
  <head>
    <title>Жаттығу 1</title>
  </head>
  <body>
    <<?php
      $Data = «Сәлем, деректер қоры
        бойынша сарапшылар!»;
      // Айнымалыны хабарландыру
      echo $daten; // Айнымалыны көрсету
    ?>
  </body>
</html>
```

HTML код

PHP-скрипт

HTML код

Тәсіл:

Мәтін немесе HTML редакторының көмегімен HTML файлының негізін құраңыз.

PHP-сценарий тегтер арасына HTML-скрипт денесіне орнатылады (=маркерлер) `<?php және ?>`.

Нұсқаулықтың әрбір жолы үтірмен жабылуы тиіс. Пікірлер `//` деген символдардан басталады.

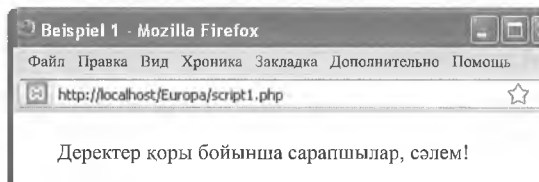
Ескерту

PHP скрипттері әрдайым қателерді дер кезінде анықтау, оқшаулау және түзету үшін біртіндеп жасалуы және тестілеуі керек.

8.3.3 PHP-дағы айнымалылар

PHP айнымалылары `$` префиксімен белгіленеді, мысалы – `$date`. Олар анық жарияланбауы тиіс, бірақ оларға мән беру арқылы тиісті деректер форматы беріледі. `$Daten` айнымалыларын экранға шығару `print $daten` немесе `echo $daten` операторының көмегімен жүзеге асырылады.

PHP сценарийі бар файл `.php` кеңейтілуімен веб-серверді жариялау каталогында сақталуы керек, мысалы, `script1.php`, `C:\` жолы бойынша файлды ашу арқылы, шолғышта желіден тыс режимде емес қоңырау шалуға болады... `\xampp\htdocs...` ал онлайн режимде, мысалы, `http://localhost/script1.php` мекенжайы бойынша.



HTML беттерінде PHP сценарийлерін түсіндіру үшін файлдар, мысалы, кеңейтумен сақталуы керек `.php` және веб-сервер арқылы орындалған.

8.3.4 Массивтер

Массив (=өріс) деректердің бір түрі міндетті емес элементтердің белгілі бір санынан тұрады. Мысалы, өрістерді, мәтіндік өрістерді және сандардың сурет нысандарын массивке біріктіруге болады.

Массивтер әртүрлі деректер түрлері бар өріс элементтерінен тұруы мүмкін.

Сондықтан PHP массивтері деректер қорындағы жазбаларды өңдеу үшін қолайлы.

Массив басқа айнымалылар сияқты, есімге ие, мысалы, \$ Клиенттер. Өрістің жекелеген элементтері сандық массивтерде үздіксіз индексімен ерекшеленеді, мысалы, \$ Клиенттер[1] = «Huber». PHP индексі әдепкі бойынша басталады[0].

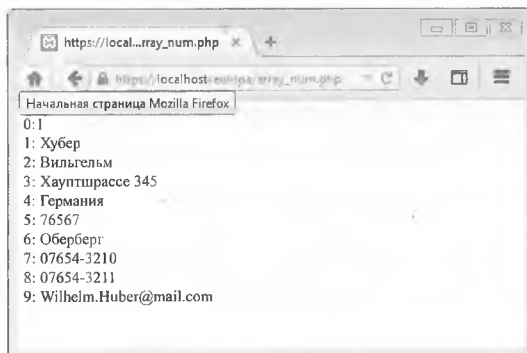
Мысал

Скрипт көмегімен келесі сандық массив құрылады –

№: 1;
Тегі: Хубер; Аты: Вильгельм; Мекенжайы: Хауптштрассе 345; Елі: Германия; Пошта индексі: 76567; город: Оберберг;
Тел.: 07654-3210;
Факс: 07654-3211;
E-Mail: Wilhelm.Huber@mail.com;

```
<?php
$клиенттер[ 0 ] = 1;
$клиенттер[ 1 ] = «Хубер»;
$клиенттер[ 2 ] = «Вильгельм»;
$клиенттер[ 3 ] = «Хауптштрассе 345»;
$клиенттер[ 4 ] = «Германия»;
$клиенттер[ 5 ] = «76567»;
$клиенттер[ 6 ] = «Оберберг»;
$клиент[ 7 ] = «07654-3210»;
$клиент[ 8 ] = «07654-3211»;
$клиенттер[ 9 ] = «Wilhelm.Huber@mail.com»;
?>
```

Экранға массивтің шығуы көрсетеді::



Ассоциативті массивтер

Ассоциативті (көріністермен байланысты) массивтер өз элементтерінің индексі ретінде сан емес, кілттер деп аталатын жолдар немесе сөздер қолданады. Оның орнына \$клиенттер[0] және \$клиенттер[1] массив элементтері \$клиенттер[Nr] және \$клиенттер [аты] .

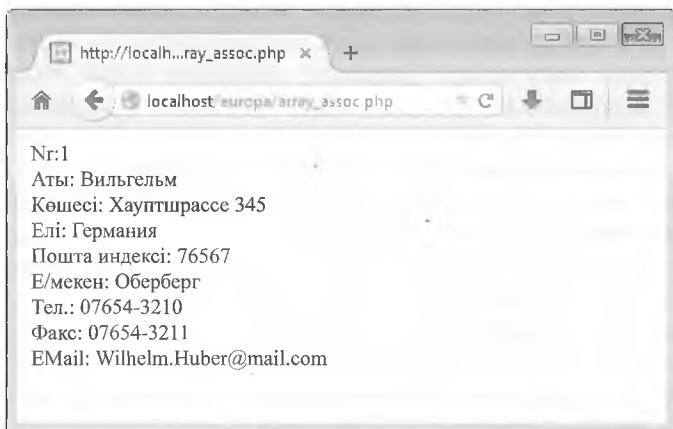
Мысал

PHP-скрипт көмегімен Huber клиентінің деректері қауымдастырылған массивте сақталады және шығарылады.

```
<?php
    $клиенттер[Nr] = 1;
    $клиенттер[ аты] = "Хубер";
    $клиенттер[ аты] = "Вильгельм";
    $клиенттер[ көшесі] = "Хауптштрассе 345";
    $клиенттер[ елі] = "Германия";
    $клиенттер[ пошта индексі] = "76567";
    $клиенттер[ Е/мекен] = "Оберберг";
    $клиенттер[ тел] = "07654-3210";
    $клиент[ факс] = "07654-3211";
    $клиенттер[ EMail] = "Wilhelm.Huber@mail.com";

foreach($kunden as $key => $element)
    echo $key,': ', $element,'<br>';
?>
```

Экранға шығарылуы:



Бұл мысалда өріс мазмұны foreach циклі арқылы шығарылады. Осы циклдің жалпы синтаксисі (ассоциативті массив үшін):

```
foreach ($arrayname as $keyname => $elementname)
Нұсқаулық;
```

Сандық массивтер үшін кілттің спецификациясы түсіріледі. Циклдің жалпы синтаксисі:

```
foreach ($arrayname as $ elementname)
{
    1-Нұсқаулық
    2-Нұсқаулық;
    3-Нұсқаулық
}
```

Бір нұсқаудың орнына, жақшадағы бірнеше фигураларды бір Нұсқаулық блогына біріктіруге болады.

Ассоциативті массивтерді реттеу үшін, массив атауынан бұрын кілт сөзді ағау жеке элементтердің кілттері тырнақшаларда көрсетіледі. мысалы, «Аты» және тағайындау операторынан кейін => мәндермен толтырылады:

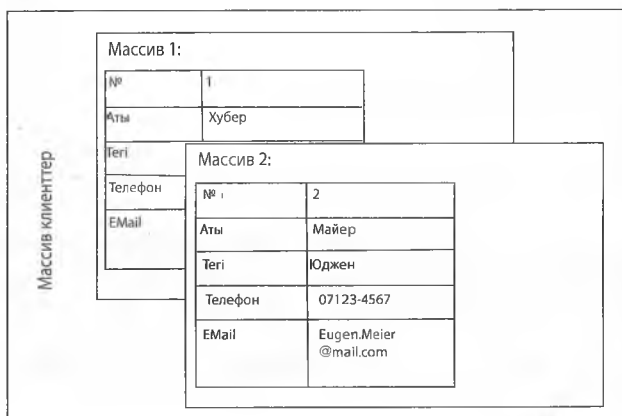
```
$клиенттер=array (
    "№" => 1,
    "Тегі"=> " Хубер",
    "Аты"=> "Вильгельм",
    ...
    "EMail"=>"Wilhelm.Huber@mail.com");
```

Көп өлшемді массивтер

Қарапайым сандық немесе ассоциативті массивтер қалауы бойынша кеңейтілуі мүмкін. Дегенмен, бірнеше клиенттердің шолуын сақтау үшін әрбір клиент үшін жаңа массив жасау ұсынылады. Жеке клиенттердің массивтері (деректер қорының тиісті жазбаларына сәйкес) неғұрлым жоғары деңгейдегі массивке біріктіріледі. келесі түрде елестетуге болады:

Көп өлшемді массивтің принципін келесідей анықтауға болады:

Көп өлшемді массивтер реляциялық деректер қорының барлық кестесінің мазмұнын сақтауға арналған.



Мысал

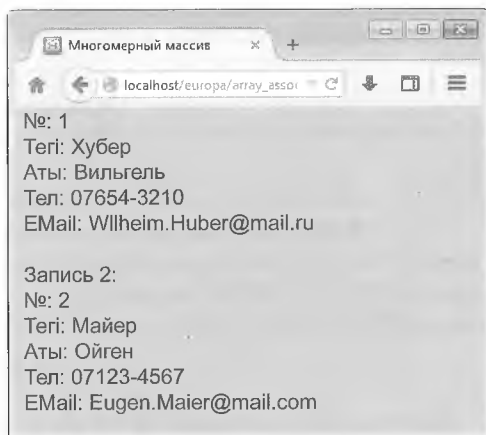
PHP сценарийі екі мекенжайды сақтайды және оларды сурет экранына шығарады.

```
<html>
  <head>
    <title>көпөлшемді массив</title>
  </head>
  <body text=#000000 bgcolor=#00FFFF
    <?php
      $клиент[ 1] = array(
        "№"=> 1,
        "Тегі"=> "Хубер",
        "Аты"=> "Вильгельм",
        "Тел"=> "07654-3210",
        "EMail"=> "Wilhelm.Huber@mail.com"
      );
      $клиент[ 2] = array( "№"=> 2,
        "Тегі"=> "Майер",
        "Аты"=> "Ойген",
        "Тел"=> "07123-4567",
        "EMail"=> "Eugen.Maier@mail.com"
      );
      foreach ( $клиент как $есептеуіш =>
$жазба)
        { echo 'жазба', $есептеуіш, ':' , '<br>' ;
          foreach($жазба as $ индекс=>$ерістің мәні)
            { echo $index, ':' , $ерістің мәні, '<br>' ;
              }
          echo '<br>' ;
        }
    ?>
  </body>
</html>
```

Түсіндірме:

Екі массивті көрсеткен соң \$клиент[1] және \$ клиент [2], деректер қос тіркемесі бар foreach циклі арқылы шығарылады. Сыртқы циклде ішкі есептеуіш \$ есептеуіш алдымен бірінші элементке орнатылады. Бұл бірінші еншілес массив \$ клиент[1] енді айнымалы \$ жазба беріледі. «1:» жазбасын шығарғаннан кейін ішкі цикл басталады. Мұнда \$index есептегіші іске қосылады және белсенді массивтің бірінші элементі \$жазба айнымалы \$F құрайтын өріске көшіріледі. Echo көмегімен... ерістің индексі мен мазмұны экранда көрсетіледі:

Егер \$index алымы деректер жиынтығының барлық элементтері арқылы өтсе, онда \$циклдің есептеуіші жоғарғы деңгейдің есептеуіші 1-ге артады және процесс келесі \$клиент кіші массивімен қайталананды[2].



8.3.5 Массивтермен жұмыс

PHP массивтерді өңдеу үшін арнайы функциялары бар:

Функция	Мысалы	Өсері
array_walk	\$нәтиже = array_walk (\$сандар, 'квадрат');	Әрбір элементке теңшелетін функцияны (мысалы, 'квадрат') қолдана отырып, массив элементтерін өзгертеді.
key	кілт = кілт (\$array);	Массив көрсеткішінің ағымдағы орналасқан жерінің кілтін қайтарады.
array_pop	array_pop (\$клиенттер);	Массивтің соңғы элементін жояды.
array_push	array_push (\$клиенттер, \$elem);	Басқа элемент ретінде массивке \$ elem қосады.

Мысал

Массивте 10-нан 20-ға дейінгі сандар сақталады. Содан кейін бұл массивтің элементтері теңшелетін функцияның көмегімен квадратқа салынады және нәтижелер экранда көрсетіледі.

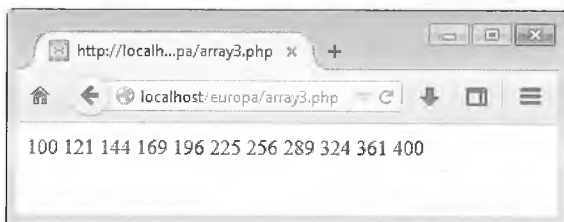
```
<?
function квадрат (&$x) {
    $x=$x* $x;
}
$сандар = массив (10,11,12,13,14,15,16,17,18,19
,20);
$нәтиже = array_walk ($сандар, 'квадрат' );
foreach
```

```
($сандары $элементі ретінде)
    echo $element, '  ' ;
}
```

?>

Алдымен элементтерді квадратқа тұрғызу функциясы анықталады. Бастапқы массивті орнатқаннан кейін \$сандар операторы array_walk (\$сандар, 'square') массив элементтерін квадратқа салады және оларды бастапқы орынға жазады. Қорытынды foreach циклінде нұсқаулар арқылы жүзеге асырылады.

Нәтиже келесідей:



8.3.6 Символдық жолдарды өңдеу

PHP символдық жолдарды өңдеу үшін түрлі функцияларды ұсынады (кестені қараңыз).

Мысалы, тіркелімдерді деректер қорында басқаруға болады:

Функция	Мысал	Сипаттамасы
strtolower()	\$name = strtolower(\$name);	Мәтінді төменгі регистрге түрлендіреді.
strtoupper()	\$name = strtoupper(\$name);	Мәтінді жоғарғы регистрге түрлендіреді.
ucfirst()	\$name = ucfirst(\$name);	Бірінші таңбаны жоғарғы регистрге түрлендіреді.
ucwords()	\$str1 = ucwords(\$str);	Әр сөздің бірінші таңбасын жоғарғы регистрге түрлендіреді.
ord()	\$mən = ord("A")	ASCII таңбаның мәнін қайтарады.

Мысал

Тіркелімге қарамастан, Хубер клиентінің деректері массивте сақталады. Шығару кезінде мәтіндік элементтердің бірінші әрпі бас, қалғандары кіші болуы тиіс.

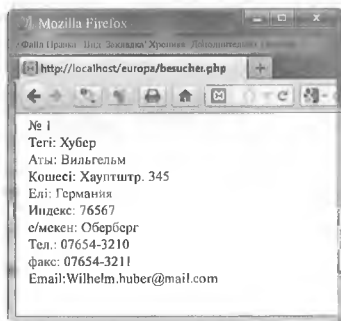
<?

```
$клиенттер = array
(
    "№"=> 1,
    "Тегі"=> "Хубер",
    "Аты"=> "Вильгельм",
    "Көшесі"=> "Хауптштр. 345",
    "Елі"=> "Германия",
    "Пошта индексі" => " 76567",
    "Е/мекен"=> "ОберБерг",
    "Тел"=> "07654-3210",
    "Факс" => "07654-3211",
    "EMail"=> "Wilhelm.Huber@mail.com"
);
foreach ($$клиенттер $есептегіш ретінде => $
    еріс мазмұны) {
    $ еріс мазмұны = strtolower ($еріс мазмұны;K)

    $ еріс мазмұны= ucfirst($еріс мазмұны;K)
    echo $есептегіш,' : ', $($еріс мазмұны;K),'<br>';
}
?>
```

Foreach функциясы strtolower (\$еріс мазмұны) айнымалының мазмұнын төменгі регистрге түрлендіреді және оны сол атауымен қайта сақтайды. Сандық айнымалылар өңделмейді.

Ucfirst() функциясы бірінші әріпті бас әріптерге түрлендіреді, содан кейін ерістің мазмұнын экранға шығарады:



8.3.7 PHP арқылы файлдармен операциялар орындау

Файл операциялары үшін бірнеше мүмкіндіктер бар:

| Функция | Мысал | Мәні |
|---------------|--|---|
| fopen() | <code>\$fp = fopen('файл аты', \$режим);</code> | Белгілі бір кіру түрінің файлын ашады |
| fclose() | <code>fclose(\$fp);</code> | Ашық файлды жабады |
| fgets() | <code>fgets(\$fp, 12);</code> | Файлдың ең көп ұзындығы 12 жолды оқиды. |
| fgetc() | <code>fgetc(\$fp);</code> | Файлдың бір таңбасын оқиды. |
| fwrite() | <code>fwrite(\$fp, \$есептеуіш);</code> | \$Str символдық жолын файлға жазады. |
| file_exists() | <code>if(!file_exists(\$есептеуіш файлы) ...;</code> | Файл бар-жоғын тексереді. |
| Функция | Мысал | Мәні |
| rewind() | <code>rewind(\$fp);</code> | Файлдың басына көрсеткіш орнатады. |
| unlink() | <code>unlink('filename');</code> | Файлды өшіреді. |
| require() | <code>require(\$есептеуіш файлы);</code> | Файлды PHP скриптіне салады. |

Мысал

Үй бетіне келушілерді есептеу үшін counter мәтіндік файлында есептеуіш мәртебесін сақтайтын есептеуіш бағдарламаланған counter.txt оны экранға шығарады.

```
<html>
<head>
  <title>есептеуішPHP</title>
</head>
<body bgcolor="#87ceff">
  <?php
    $zaehldatei="counter.txt";
    if(!file_exists($файл есептеуіш))
    {
        $fp=fopen($zaehldatei, "w+");
        fwrite($fp, "0");
        fclose($fp);
    }
    $fp=fopen($zaehldatei, "r+");
    $zaehler=(int) fgets($fp, 12);
    $zaehler++; rewind($fp);
    fwrite($fp, $zaehler);
    fclose($fp);
```

```

?>
<div align="Center">
    <<қаріп көлемі="+2">
    <? require($есептеуіш файлы); ?>
        . Келуші!!<!/!</font></div>
</body>
</html>

```

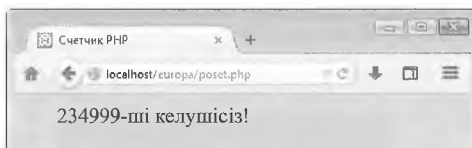
Counter файлында.txt бүтін сан ретінде тек есептегіш көрсеткіші сақталады. Бұл файлдың аты айнымалыда көрсетіледі.

Егер бұл файл жоқ болса (әлі жоқ), if операторы орындалады. Бұл файл w+режимінде жасалады және ашылады. Бұл файлдың жұмыс режимі ол оқу және жазу үшін ашылатынын анықтайды, егер ол жоқ болса, алдымен құрылады. Fwrite () функциясы 0 санын файлға жазады. fclose () файлды жабады.

Fopen() функциясы енді R+ режимінде оқу және жазу үшін бар файлды ашады. Fgets() көмегімен 12 таңбаның ең көп саны жоң ретінде есептеледі және int () көмегімен бүтін санға айналады. Ол \$zaehler және

\$ zaehler++ бір артады. Rewind () функциясы файлдың басына ішкі көрсеткішті орнатады. Ол \$zaehler және \$ zaehler++ бір өзгереді. Rewind () функциясы файлдың басына ішкі көрсеткішті орнатады. Онда сан fwrite () арқылы бірлікке артады және файл қайтадан жабылады.

Шығыс деректері HTML форматында берілген, сонымен қатар есептеуіштің көрсеткіштері require (\$ count) функциясы арқылы файлды біріктіру арқылы көрсетіледі.



Файлдар fopen () функциясы арқылы әртүрлі режимдерде ашылуы мүмкін. A режимі файлды ашады және файлдың ішкі көрсеткішін файлдың соңына орналастырады. Тек жазу операцияларына рұқсат етіледі. A + режимінде бұл шектеу жоқ, оқу мүмкін. Екі режимдерде де таңдалған файл әлі жоқ болса жасалады.

Файлдармен мүмкін болатын жұмыс режимдеріне шолу:

Жұмыс режимі	Түсіндірме
"r"	Файлды тек оқу үшін ашады.
"r+"	Файлды оқу және жазу үшін ашады.
"w"	Файлды тек жазу үшін ашады.
"w+"	Оқу және жазу үшін файлды ашады. Алдымен мазмұны жойылады. Егер файл жоқ болса, ол жасалады.
"a"	Файлды тек жазу үшін ашады және файлдың соңына файл көрсеткішін орналастырады, егер файл жоқ болса, ол жасалады.
"a+"	Оқу және жазу үшін файлды ашады және файлдың көрсеткішін файлдың соңына орналастырады.

8.3.8 Файлдарға кіру құқығы

Файлдармен, әсіресе деректер қорымен жұмыс істеу кезінде маңызды ақпарат қол жеткізу құқығы болып табылады. Сіз осы файлдармен байланысты әртүрлі пайдаланушылардың рұқсаттары туралы біле аласыз.

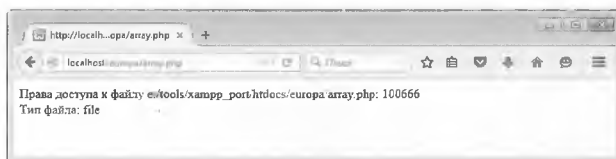
PHP файлдар мен кіру құқықтарын басқару үшін қолайлы функцияларды ұсынады:

Функция	Сипаттамасы
filegroup();	Файл иесінің тобын анықтайды.
fileowner();	Файл иесін анықтайды.
fileperms();	Файлға кіру құқықтарын анықтайды.
filesize();	Файл өлшемін анықтайды.
filetype();	Файл түрін анықтайды.
decoct();	Ондық санды сегізге түрлендіреді.

Мысал

Файл үшін, мысалы, кіру құқықтары мен файл түрі D:\tools\xampp_port\htdocs\europa\array.php, PHP сценарийі көмегімен анықталады.

```
<? php
    $файл = "d:/tools/xampp_port/htdocs/europa/array.
php";
    $serg=fileperms ($файл);
    echo " $datei: файльна кіру құқығы "
    , decoct($serg) , "<br>";
    echo "файл түрі: ", filetype($файл);
?>
```



Файлдың айнымалы жолын анықтағаннан кейін fileperms() функциясы осы файлға кіру құқығын анықтайды. Нәтиже ондық жазбада болғандықтан, бірақ тек сегіздік жазбада дұрыс түсіндірілуі мүмкін, оны decoct() функциясы арқылы алғашқы үш сан, мұнда 100 файл түрі туралы ақпараттандырады. Filetype () функциясы файл түрін мәтіндік айнымалы түрінде шығарады.

Ықтимал файл түрлері:

- block: арнайы құрылғыны құлыптау
- char: символдық арнайы құрылғы
- dir: директория
- fifo: FIFO (атаулы арна)
- file: кәдімгі файл
- link: символдық сілтеме
- unknown: белгісіз файл түрі

666 сандар тізбегі иесінің, топтың және басқа пайдаланушылардың құқықтары туралы хабарлайды. Бірінші 6 деген сан файл иесі оқуға (4), жазуға (2) құқығы бар, бірақ орындай алмайды дегенді білдіреді. Екінші 6 саны топтың тең құқықтары туралы және соңғы 6 саны – барлық қалған пайдаланушылардың құқықтары туралы хабарлайды.

Әр пайдаланушының ықтимал құқықтары:

Иесі			Тобы			Басқа тұтынушы		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

rr от read = оқу, w от write = жазба, X от execute = орындау

Мысал

Файл үшін кіру кілті ретінде 754 саны пайдаланылады. Жеке құқықтар мен өкілеттіктер мынадай түрде айқындалады:

7 = иесі: оқу (4) + жазу (2) + бастау (1);

5 = тобы: оқу (4) + іске қосу (1);

4 = басқа: оқу (4).

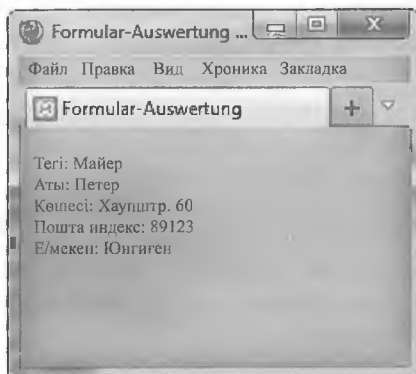
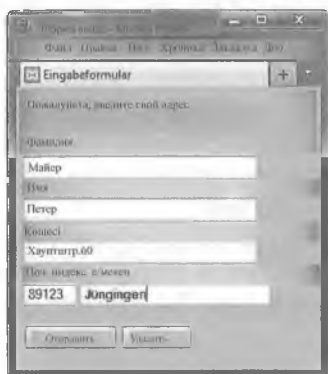
8.3.9 Нысандармен жұмыс

PHP-скрипттер HTML-формасындағы деректерді өңдеуге мүмкіндік береді.

Мысал

HTML-нысанда мекенжай (тегі, аты, көшесі, пошталық индексі, қаласы) енгізіледі. Командалық батырма PHP скриптіне деректерді жіберу және оларды көрсету үшін пайдаланылады. Енгізу нысанының коды келесідей:

```
<html>
<head><title>енризы типі</title></head>
<body text=»#000000» bgcolor=»#00E0FF»>
  <h4>мекенжайыңызды енгізіңіз:< /></h4>
  <form action="form_aus.php" method="POST">
    name:<br><input type="Text" name="Тегі"
      size="35">
    <br> name:<br><input type="Text" name="Аты"
      size="35"><br>
    Көше:<br><input type="Text" name="Көшесі" size="35">
    <br>
    Пошта индексі: Е / мекен: <br><input type="Text"
      name="Индекс" size="5" >
    < input type="Text" name="Е/мекен" size="28"
    ><br><br>
    <input type="submit" name="жазылу" value="жіберу">
    <input type="reset" name="лоесчен" value="жою"
  </form>
</body>
</html>
```



Алдымен HTML кодында пішін жасалады. POST әдісі арқылы бұл нысандар PHP скриптіне жіберіледі, мысалы, form_aus.php. Бұл HTML нұсқаулығымен жасалады

```
<form action="form_aus.php" method="post">.
```

Жіберу түймешігін басқанда, деректер form_aus.php файлына жіберіледі

Form_aus.php файлына код үлгісі

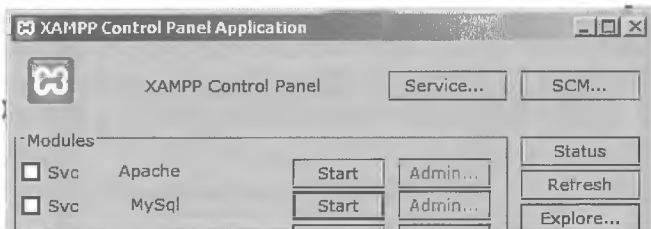
```
<html><head>
    <<title>нысанды бағалай</title>< / head>
<body text="#000000"bgcolor="#00E0FF">
<?php
    $тері =$_POST [ "Аты" ];
    $аты =$_POST[ "Vorname" ];
    $көше =$_POST [ "көше" ];
    $пошта индексі =$_POST [ "пошта индексі" ];
    $е/мекен =$_POST[ "Е/мекен" ];
    echo " <br>Name:", $тері;
    echo " <br>аты:", $аты; echo " <br>көшесі:",
    $көшесі; echo " <br>индекс:", $индекс;
    echo " <br>е / мекен:", $ е/ мекен;
    ?>
</body > </html>
```

\$_POST стандартты айнымалысы [] деректерді массив пішімінде береді. Элементтер, мысалы, \$vorname=\$_POST[«Vorname»] көмегімен есептеледі және жергілікті айнымалымен беріледі. Олар echo пәрменімен көрсетілуі мүмкін...

8.4 MYSQL ДЕРЕКТЕР ҚОРЫН БАСҚАРУ ЖҮЙЕСІ

MySQL-бұл деректер қорының реляциялық жүйесі. Mysqld деректер қорының сервері клиент-сервер архитектурасы қолданылады. ехе сервердегі фондық процесс ретінде жұмыс істейді. Серверге кіру шексіз клиенттер ала алады. Кеп дәлділік тұжырымдамасы серверге әртүрлі сұраныстарды өңдеуге мүмкіндік береді.

MySQL дистрибутиві тегін XAMPP деректер қорын өзірлеу құралына қосылған. Серверді, мысалы, XAMPP Control Panel арқылы іске қосуға болады, MySQL жазбасының жанындағы Бастау батырмасын басыңыз.

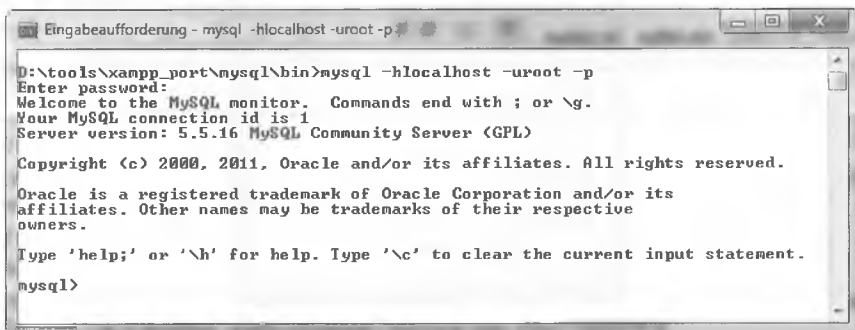


8.4.1 MySQL клиенттерімен MySQL жұмыс істейді

MySQL клиенттік бағдарламасы (MySQL деректер қоры жүйесіне қарағанда кіші әріптермен жазылады)

MySQL серверімен байланысады және командаларды серверге жіберуге мүмкіндік береді.

MySQL клиенті XAMPP орнату каталогында Windows пәрмен жолынан іске қосылады, файл аты `mysql`. Қосылым параметрлері серверге жүгінгенде көрсетіледі. Атрибуттан кейін `-h` деректер қоры серверінің орналасуы көрсетіледі, мысалы: `localhost`, кейін `-u` пайдаланушының орналасуы, мысалы, `root`. Root пайдаланушысы үшін әдепкі құпия сөз орнатылмаған. Қауіпсіздікті арттыру үшін құпия сөзді орнату қажет.



Сәлемдесуден кейін клиентке командалар енгізілуі мүмкін. Монитордың командалары әрқашан `;` белгісімен аяқталады. Анықтаманы шақыру `\h` пәрменімен жүзеге асырылады.

```

mysql> \h
For information about MySQL products and services, visit:
  http://www.mysql.com/
For developer information, including the MySQL Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Enterprise support, training, or other products, visit:
  https://shop.mysql.com

List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
(?) Synonym for 'help'.
clear      (C) Clear the current input statement.
connect    (C) Reconnect to the server. Optional arguments are db and host.
delimiter (d) Set statement delimiter.
escape    (G) Send command to mysql server, display result vertically.
exit      (G) Exit mysql. Same as quit.
go        (G) Send command to mysql server.
help      (h) Display this help.
noisy     (C) Don't write into outfile.
print     (p) Print current command.
prompt    (P) Change your mysql prompt.
quit      (G) Quit mysql.
rehash    (R) Rebuild completion hash.
source    (S) Execute an SQL script file. Takes a file name as an argument.
status    (S) Get status information from the server.
tee       (T) Set outfile [to_outfile]. Append everything into given outfile.
use       (U) Use another database. Takes database name as argument.
charset   (C) Switch to another charset. Might be needed for processing binlog
with multi-byte charsets.
warnings  (W) Show warnings after every statement.
warning   (W) Don't show warnings after every statement.

For server side help, type 'help contents'

mysql>

```

MySQL шақыруы бойынша операторлар деректер қорының серверіне жіберіледі.

Мысалы, show database командасы; барлық деректер қорын серверге шығарады. Суретте бұл 11 дерекқор.

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| donaureisen |
| fahrschule |
| faradiso   |
| mysql      |
| performance_schema |
| phpmyadmin |
| test       |
| wb         |
| webauth    |
+-----+
11 rows in set (0.00 sec)

```

MySQLadmin клиенті

MySQLadmin клиенті деректер қоры серверінде әкімшілік тапсырмаларын орындауға мүмкіндік береді.

Мысал

Root пайдаланушысы үшін hus89adrbsiu құпия сөзі орнатылған.

```

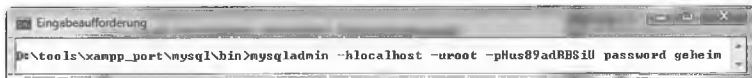
D:\tools\xampp_port\mysql\bin>mysqladmin -localhost -uroot password Hus89adRBSIU

```

MySQL каталогының командалық жолында localhost сервері мөн root пайдаланушысы үшін атрибуттары бар mysqladmin бағдарламасы шақырылады. Password кілт сөзі дерекқор сервері үшін соңғы құпия сезді анықтайды.

Мысал

Құпия сөз құпия түрде жаңа құпия сөзге ауысады.



```
D:\tools\xampp_port\mysql\bin>mysqladmin -hlocalhost -uroot -pplus89adRBSiU password geheim
```

Дерекқор серверіне екінші рет кіру үшін осы құпия сезді өзгерту үшін -p атрибутынан кейін алдын ала анықталған құпия сөз қажет.

Mysqldump клиенті

Mysqldump клиенті деректерді резервтік көшіру үшін пайдаланылады. Бағдарлама Деректер қорының мазмұнын мәтіндік файлға жазады.

Мысал

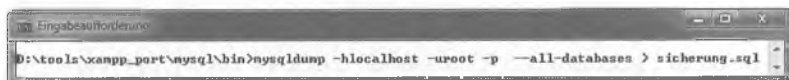
Резервтік көшіру мақсатында Faradiso деректер қорының мазмұны backup_file.sql файлында сақталады.



```
D:\tools\xampp_port\mysql\bin>mysqldump -hlocalhost -uroot -pgeheim faradiso > backup_file.sql
```

Жүйеге кіру атрибуттарынан кейін деректер қоры көрсетіледі және > белгісінен кейін экспорт файлының аты көрсетіледі, мысалы, backup_file.sql. Бұл мәтін файлында кестелерді қайта жасайтын және орындалған кезде деректерді кірістіретін SQL командалары жазылған.

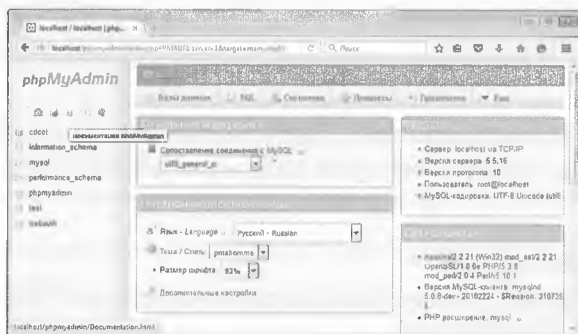
Мысалы, sicherung.sql файлына деректер қоры серверінің барлық деректер қорын резервтік көшіруді орындайды:



```
D:\tools\xampp_port\mysql\bin>mysqldump -hlocalhost -uroot -p --all-databases > sicherung.sql
```

PhpMyAdmin клиенті

XAMPP әзірлеу ортасы әдепкі phpMyAdmin клиентін ұсынады. Ол браузерде localhost / phpmyadmin адресін шақырудан басталады. MySQL деректер қорының серверінде барлық әкімшіліктендіру тапсырмалары осы құралдың көмегімен ыңғайлы орындалуы мүмкін. Сондықтан алдағы уақытта, ең алдымен, осы клиентпен жұмыс істейтін боламыз.

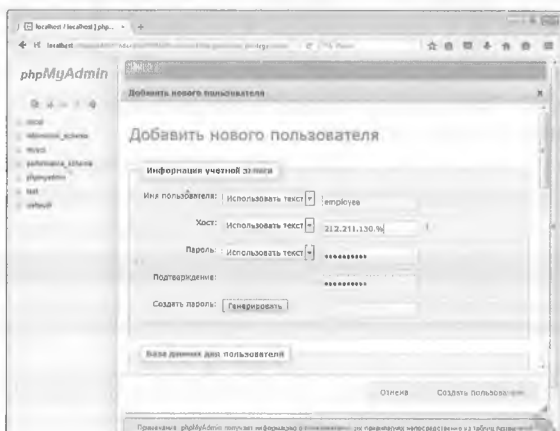


8.4.2 Қолжетімділік құқықтарын беру және жою

MySQL mysqli деректер қорында пайдаланушылардың құқықтарын өте сараланған түрде сақтайды.

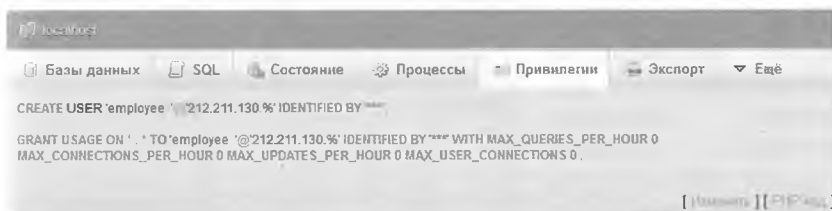
Мысал

PhpMyAdmin клиентінде, мысалы, қызметкер атымен және қызметкер паролі бар жаңа пайдаланушы құрылады. Ол тек 212.211.130.xxx.домен хостымен кіру мүмкіндігіне ие болуы тиіс.



Басты бетте құқық қосымшасында MySQL деректер қорында бар пайдаланушыларды көруге болады. Мұнда сіз жаңа пайдаланушы қосуды таңдасаңыз және жоғарыда көрсетілген деректерді енгізу үлгісін ашасыз. Жүйеге кіру үшін қажетті ақпаратты енгізу жүргізіледі. Хост жолында пайдаланушы дерекқор серверіне кіруге рұқсат етілген нақты хост енгізу үшін ашылған терезеде мәтіндік өрісті пайдалану параметрін орнату қажет. Егер жаңа пайдаланушы белгілі бір желінің барлық компьютерлерінен кіру мүмкіндігі болса, мекенжайдың соңғы байтының орнына %топтырғышын көрсетіңіз.

Жаһандық құқық саласында пайдаланушының құқықтары белгіленеді. Мысалы, деректерді оқу үшін SELECT, бар деректерді өзгерту үшін кестелерге және UPDATE жазбаларын енгізу үшін Insertд. Пайдаланушы құру түймешігін басқанда, жаңа пайдаланушы CREATE операторымен құрылады және оған пайдаланушы құқықтары тағайындалады. Сәтті орындалғаны туралы SQL кодында орындалған команданы көрсету арқылы хабарланады.

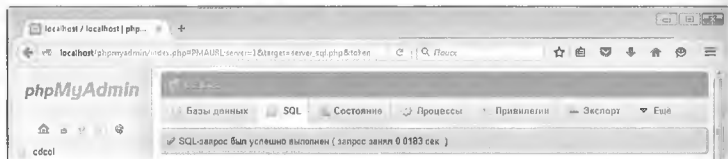


Обзор учетных записей

Пользователь	Хост	Пароль	Глобальные привилегии	GRANT	Действие
employee	212.211.130.%	Да	USAGE	Нет	Редактирование привилегий Экспорт

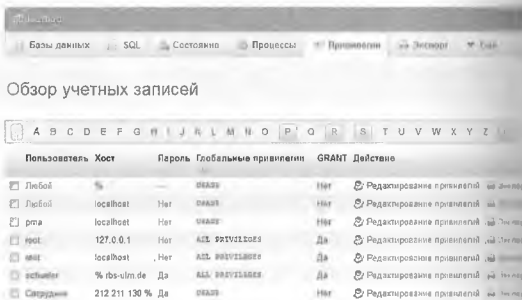
Мысал

SQL пәрменімен Schueler-дің жаңа пайдаланушысын жасаңыз. Ол Домен хостымен ғана кіру мүмкіндігі болуы керек rbs-ulm.de және барлық құқықтармен тек faradiso деректер қорын өңдеу.



GRANT кілт сөзі SQL-пайдаланушы құқығын ұсыну үшін пәрменді іске қосады. ALL сөзі пайдаланушы үшін барлық құқықтарды белгілейді. FARADISO атрибуты бар.* бұл құқықтар faradiso деректер қорының барлық кестелерімен шектеледі. Атрибут *.* пайдаланушы барлық қолданыстағы деректер қорын өзгерте алады дегенді білдіреді. TO атрибуттан кейін, мысалы, schueler жүйесіне кіру үшін жаңа пайдаланушының аты, ал @ - хост таңбасынан кейін немесе пайдаланушы жүйеге кіруге рұқсат етілген желі аймағы бар. Кіру үшін құпия сөз IDENTIFIED BY кілт сөзімен басталады.

MySQL деректер қорын тікелей манипуляциялардан SQL командаларын пайдаланудың артықшылығы – бұл деректер қорындағы кестелердің мазмұнын білу қажет емес және осы кестелердің қайсысында жазбалар үлестірілуі тиіс шешім қабылдау қажет емес.



Әр пайдаланушының құқықтарын phpmyadmin клиентінің пайдаланушыларын шолуда керуге болады.

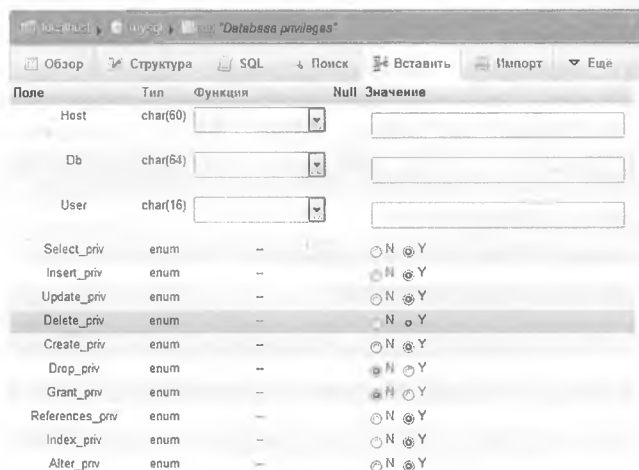
Schueler (оқушы) пайдаланушысының құқықтары мысалда USAGE терминімен берілген. Бұл кілт сөз schueler пайдаланушысының осы дерекқор серверінде жаһандық құқықтары жоқ екенін айтады.

Оның faradiso деректер қорына құқықтары MySQL деректер қорының db кестесінде сақталады. Кестедегі жазбалардың фрагменті:

Host	Db	User	Select_priv	Insert_priv	Update_priv	Delete_priv	Create_priv	Drop_priv	Grant_priv
%	test		Y	Y	Y	Y	Y	Y	N
%	test%		Y	Y	Y	Y	Y	Y	N
localhost	phpmyadmin	pma	Y	Y	Y	Y	N	N	N
%fbs-	faradiso	ученик	Y	Y	Y	Y	Y	Y	N

Мысалы, оқушы Grant_priv (жазба - N) құқығын алмаған, әйтпесе ол басқа пайдаланушыларға құқықтарды тағайындай алады және осылайша өз шектеулерін айналып өтуге болады.

Мысалы, schueler пайдаланушысы кестелерді немесе деректер қорын жою алмауы үшін, бұл пайдаланушыны drop_priv құқығынан айыру керек. Бұл phpmyadmin клиентінде оның құқықтарын өңдеу арқылы жасалады. Онда Drop_priv жолында N түймесін таңдау және растау қажет.



Ескерту:

Ғаламдық рұқсаттар сервердегі барлық деректер қорына қолданылады.

8.4.3 PHP көмегімен MySQL деректер қорын өңдеу

PHP MySQL-мен жұмыс істеу үшін түрлі мүмкіндіктерді ұсынады:

PHP функция-сы	Сипаттамасы	Мысал
mysql_connect()	ДБ серверіне қосылу.	\$serg=mysql_connect (\$host, \$user, \$password);
mysql_db_query()	SQL-сұранымды дерекқор серверіне жібереді.	\$serg=mysql_db_query ('faradiso', 'select * from клиент-ты');
mysql_num_rows()	Сұраныстағы жазбалар саны.	\$саны= MySQL_num_rows(\$serg);
mysql_num_fields()	Сұрау жазу өрістерінің саны.	\$саны= MySQL_num_fields(\$serg);
mysql_close()	Дерекқор серверіне қосылуды жабады.	mysql_close();
mysql_fetch_array()	Жазбаны массив түрінде қайтарады.	\$liste=mysql_fetch_array (\$нәтиже, \$тип)

Деректер қоры серверіне қосылу және деректер сұранысы бірнеше кезеңде орындалады:

1. Пайдаланушының дерекқор серверіне кіруі
2. Деректер қорын теңшеу
3. Дерекқордың серверіне сұранысты жіберу, нәтижені айнымалы күйінде сақтау
4. Қосылымды жабу

Дерекқор серверіне пайдаланушының кіруі `mysql_connect ($host, $user, $passwd)` функциясы арқылы жүзеге асырылады.

SQL-сұрау `mysql_db_query ($db, $sql)` функциясы арқылы жіберіледі. Мысалы, `mysql_num_rows ($erg_sql)` функциясы кестедегі деректер жазбаларының санын есептеп, оларды `$anz` айнымалысында сақтайды. `mysql_close ()` функциясы деректер қоры серверімен байланысты өшіреді.

Жазба санын шығару сценарийі:

```
<?
$db      = mysql_connect('localhost', 'root', '');
$erg_sql = mysql_db_query('faradiso', 'select * из
клиенты' );
$anz = mysql_num_rows($erg_sql);
echo $anz;
mysql_close($db);
?>
```

Деректер қоры серверіне қосылу

SQL нұсқау-лығын жіберу

Көру

SQL-нәтиженің нәтижесін оқу

ДБ қосылуын жабу

Мысал

FARADISO деректер қорының клиенттері кестені HTML кестесі түрінде экранға шығаратын PHP-сценарийін әзірлеу қажет.

Алдымен скрипт қосылымды орнатады және SQL нұсқауын (командасын) жібереді. Қажетті бағандардың саны `PHP mysql_num_fields($erg_sql)` функциясы арқылы анықталады және `$number` айнымалысы қойылады. `For` циклінің көмегімен кесте тақырыбы жасалады, ол `mysql_field_name($erg_sql, $i)` функциясының көмегімен ерістердің аттарымен белгіленеді.

`mysql_fetch_array ($erg_sql, MYSQL_ASSOC)` функциясы `while` циклінде әрбір іске қосу кезінде `$zeile` ассоциативті массивіне сұрау нәтижесін жазуды оқиды. `Foreach` циклы әрбір жеке жазу үшін HTML кестесіндегі жеке жазу өрістеріне және шығаруға қол жеткізуді қамтамасыз етеді. Кестенің соңғы тегтері көрсетілгеннен кейін деректер қоры серверімен байланыс `mysql_close ()` жабылады.

Скрипт мысалы:

```

<?php
$db = mysql_connect('localhost', 'root', '');
$erg_sql = mysql_db_query('faradiso', 'select *
    from kunden');
?>
<table width=80% border=1
  <tr bgcolor=' #CFCFCF' >
    <? $cаны=mysql_num_fields ($erg_sql)
    for ($i=0;$i<$cаны; $i++) {
      ?> <th><?
        echo mysql_field_name($erg_sql,$i);
      ?> </th> <?
    } ?>
  </tr>
  <tr>
    <?
    while ($can = mysql_fetch_array ($erg_sql,
    MYSQL_ASSOC)){
      foreach ($can as $elem) {
    echo «<td bgcolor=' #EFEFEF' ><font size='-1' > $elem </
    font></td>»;
      }
    }
  ?></tr><?
}
</table> <? ?>
mysql_close();
?>

```

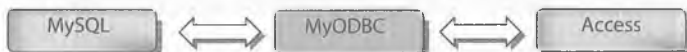
Бағандар тақырыбын шығару

Деректер қоры жолының мазмұнын шығару

Жеке жазбаларды оқу

3.5 ODBC ИНТЕРФЕЙСТЕРІ АРҚЫЛЫ ДЕРЕКТЕР АЛМАСУ

Әртүрлі форматтағы деректер қорынан деректерді тікелей алмасу мүмкін емес. Сондықтан ODBC (Open Database Connectivity = ашық деректер қоры байланысы) стандарты жасалды, ол мәліметтерді ODBC арнайы драйверлері арқылы алмасуға мүмкіндік береді. Сонымен қатар, MySQL мәліметтерді ODBC драйверіне береді, ол оларды Access сияқты екінші ДББЖ пішіміне түрлендіреді.



Мысал:

Әртүрлі деректер қоры жүйелері ODBC интерфейстері арқылы бірге жұмыс істей алады.

Access және MySQL арасында ақпарат алмасу үшін Интернетте Linux және Windows үшін қолайлы драйверлер бар, мысалы, `mysql-connector-odbc - 3.51.23-win32` орнату файлы ретінде `.msi`.

MyODBC драйверін орнату

Орнату орнату файлын іске қосу арқылы орындалады. Сәлемдесу экраны келесі батырмамен расталады>. Содан кейін бағдарлама қалаған орнату түрін сұрайды (суретті қараңыз). Түр жалаушасын қойып, Келесі > түймесін басу арқылы растаңыз. Ашылған терезеде орнату тарихы көрсетіледі және жұмыстың сәтті аяқталғаны расталады.



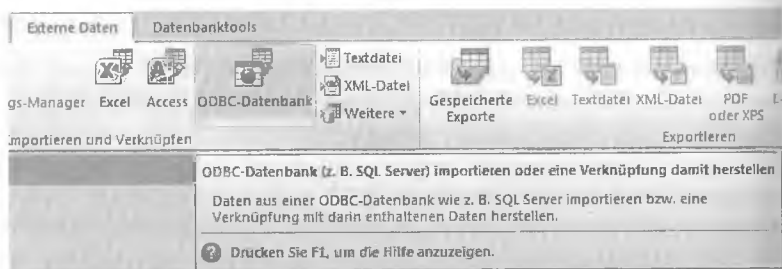
Access арқылы MySQL деректер қорын өңдеу

Access іске қосқаннан кейін

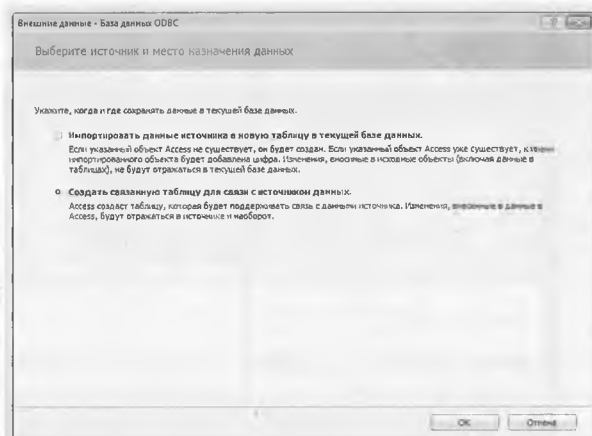
ODBC интерфейсі арқылы жаңа деректер қоры бар MySQL деректер қорының кестелеріне қол жеткізу жүзеге асырылады.

Мысал

Кесте MySQL FARADISO деректер қорының клиенттері Access деректер қорының жүйесімен өңделеді. Бұл үшін Деректер қоры жүйесі іске қосылғаннан кейін Access-ті сыртқы деректер мәзірінен таңдаңыз. ODBC деректер қоры параметрі.

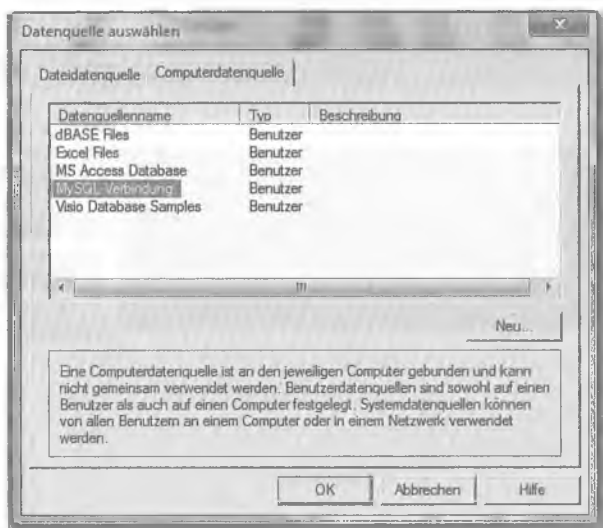


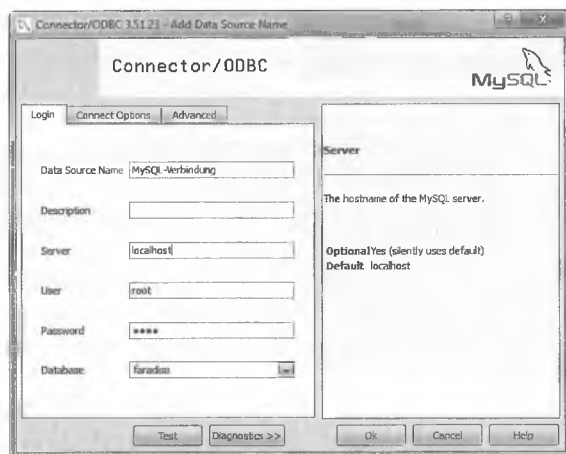
Келесі экранда деректер импорты мен сілтеме жасау арасында таңдай аласыз. Сілтемені жасау батырмасын басып... OK арқылы растаңыз.



Осылайша, деректер Access деректер қорына импортталмайды, бірақ MySQL деректер қорында қалады, бірақ ODBC интерфейсі арқылы Access-тен редакциялануы мүмкін. Импорт параметрі таңдалса... деректер Access деректер қорына тәуелсіз кесте ретінде көшірілетін болады. MySQL деректер қоры өңдеу жолымен өзгертілмейді.

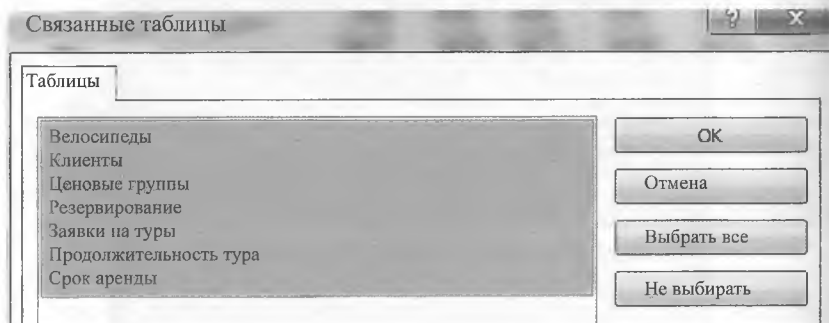
Келесі терезе дерек кезін таңдау үшін диалог береді. Компьютер дерек көзі қосымшада MySQL connection дерек кезінің аты OK басу арқылы таңдалады және расталады.





MySQL серверіне кіру үшін келесі терезеде бірегей байланыс аты, MySQL серверінің аты, FARADISO сияқты бар Дерекқордың аты, сондай-ақ root сияқты Пайдаланушының аты болуы керек. Қажет болған жағдайда тағы бір құпия сөзді көрсету қажет.

OK кемеімен расталғаннан кейін Access MySQL деректер қорының серверінде тіркеледі және MySQL FARADISO деректер қорында қол жетімді кестелерді көрсетеді. OK кемеімен барлық мәліметтерді таңдау және Растау батырмасын басқаннан кейін, MySQL деректер қорының барлық кестелері редакциялау үшін қолжетімді.



Id	Name	Strasse	PLZ	Ort	Telefon	Mobile	Mail
1	Herr Mal Hans	Friedberger La	89077	Ulm	0731/436543	0171 54513676	
2	Frau Winter Susanne	am Waldstrand	89077	Ulm	0731/4892		
3	Frau Sommer Andrea	Adelsriedstr. 1	88446	Oberach	07364/894623	0160 45679161	
4	Frau Holm Mira	Allendorfer Str	88365	Ehningen	07344/733665	0175 3583636	
5	Herr Zwickel Paul	Pögelstr. 123	89123	Ehningen	01421/746445		
6	Herr Winkler Claus	Eisenerhelmer	89333	Thalfingen	07341/7807474		
7	Frau Hase Hanna	Ulrichstr. 43	89332	Gönzburg	08231/5432231		
8	Frau Hales Isabelle	Hinter den Lin	89248	Senden	(0821)453412		
9	Herr Hender Christoph	Rungelstr. 34	88612	Augsburg	(0731)457849		
10	Frau Metz Tatjana	Berger Str. 99	89677	Ulm/Do	(0731)457849		
11	Herr Wiese Markus	Löwenjasse 11	89077	Ulm a d Donau	(0731)457849		

Қазір жасалған өзгерістер, мысалы, Клиенттер кестесінде Access деректер қорында сақталмайды, ал ODBC интерфейсі арқылы MySQL деректер қорына жіберіледі және сонда сақталады. Бұл өзара байланысты MySQL деректер қоры кестелері кесте атының алдында таңбамен белгіленеді.

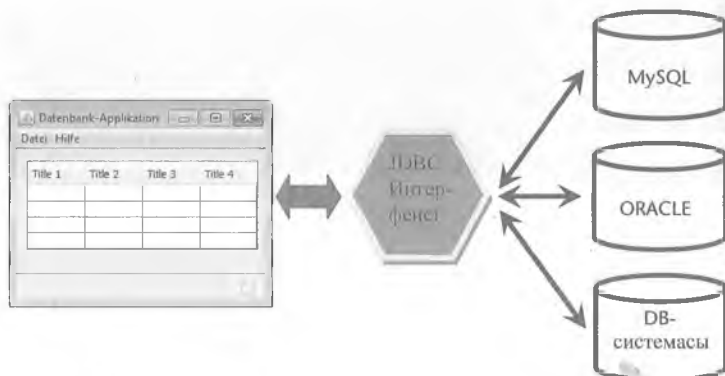
9 Java арқылы деректер қорына кіру

9.1 JAVA АРҚЫЛЫ ДЕРЕКТЕР ҚОРЫНА КІРУ

Деректерді сақтау қолданбалы бағдарламалардың маңызды міндеті болып табылады. Әрине, қолданба деректерді файлдық операцияларды пайдалана отырып дербес сақтай алады. Шағын деректер келемі үшін бұл, бәлкім, ең жақсы таңдау, өйткені салыстырмалы деректер тәуелсіздігі сақталады. Алайда, егер деректердің (немесе жазбалардың) үлкен көлемін сақтау талап етілсе, ал деректердің өздері күрделі құрылымы болса, онда оларды деректер базасында сақтау туралы мәселені қарастыру қажет. Деректер қорына қосылудың үлкен артықшылығы – қосымшаның деректер қоймасын техникалық іске асырудың тәуелсіздігі. Бұл фондық режимде деректер қорының көмегімен жасалады. Деректерді өзгерту немесе жою, сондай-ақ тиісті деректер қорының пәрмендерімен оңай жүзеге асырылады. Егжей-тегжейлі талқыланған **SQL** сұрау тілі мұнда маңызды рөл атқарады.

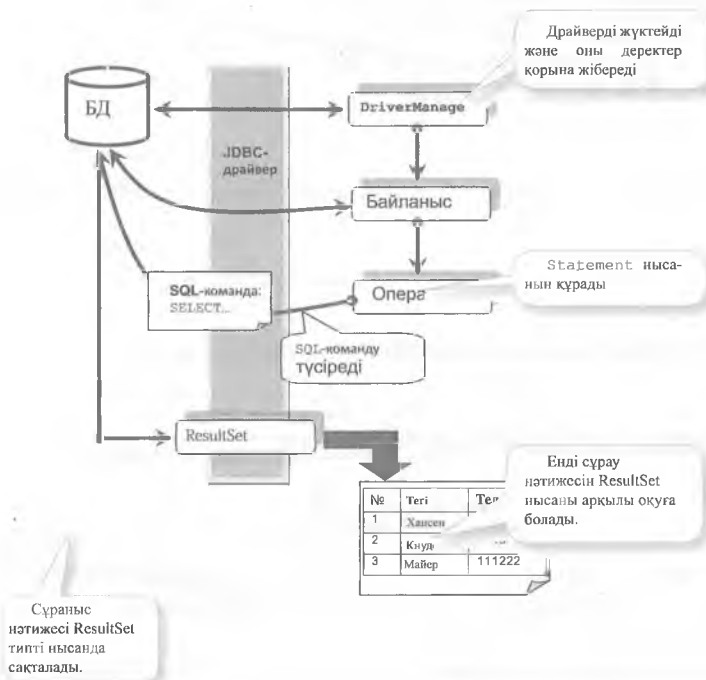
9.1.1 JDBC көмегімен деректер қорын байланыстыру

Java деректер қорына қосылу үшін көптеген кластарды ұсынады. Бұл кластар **JDBC (Java Database Connectivity)** жалпы терминімен жиналған деректер қорының кепшілігінде **Java қосымшасынан** деректер қорына қолжетімділікті тиісті деректер қорының командаларына түрлендіретін JDBC-интерфейстері бар. Жалпы, Java бағдарламашысы үшін фондық режимде қандай деректер базасы қолданылады – қатынау бір тәсілмен іске асырылады. Келесі суретте осы қолжетімділікті іске асырудың негізгі принципі көрсетілген:



9.1.2 JDBC драйверлерін жүктеу және байланыс орнату

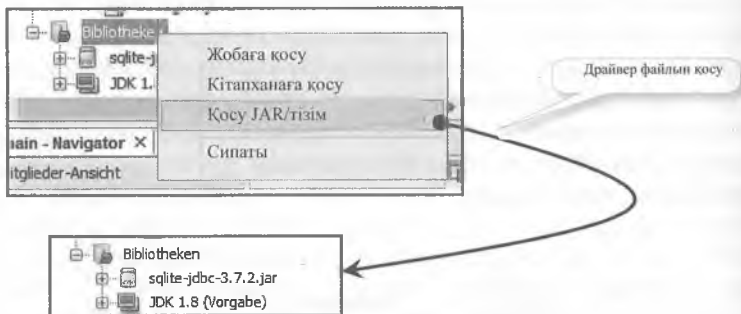
Деректер қорына қосылу үшін тиісті драйверді алдын ала орнату қажет. Содан кейін **DriverManager** класы әдісі арқылы драйверді жадқа жүктеуге болады. Бұл класс **java.sql** пакетінің бір бөлігі болып табылады. **Java** жобасына импорттайтын Сәтті жүктелгеннен кейін, драйвер деректер қорына қосылуы мүмкін. Бұл деректер қорына қосылу үшін **DriverManager** пайдаланылатын **connection** класы арқылы жасалады. Деректер қорына байланысты Пайдаланушының аты мен паролін көрсету қажет. Содан кейін **Statement** нысаны сұрауды іске қосып, оны **ResultSet** нысанының көмегімен оқи алады. Келесі сурет қарым-қатынасты көрсетеді:



9.1.3 SQLite деректер қорына кіру

Келесі бөлімде **SQLite** деректер қорына қатынасу сипатталған. Берілген принцип **MySQL** немесе **Oracle** деректер қоры сияқты басқа реляциялық дерекқорларға да қолдануға болады. Бұл ретте пакет **java.sql** пакеті импортталуы тиіс, бұл пакетте деректер қорына кіру үшін барлық тиісті сыныптар бар.

Келесі мысалдарда **NetBeans** әзірлеу ортасы қолданылады. Бұл принцип басқа ортаға (мысалы, **Eclipse**) көшірілуі мүмкін. Қажетті драйверді жүктегеннен кейін (мысалы, **sqlite-jdbc-XXX.JAR**) **Java** Мұрағат файлы жобаға біріктіріледі:



Драйвер сәтті қосылғаннан кейін, байланыс **Class** класы арқылы жасалуы мүмкін:

Деректер қоры жолы = "jdbc:sqlite:/жол/деректер қоры";

```
Class.forName("org.sqlite.JDBC");
```

Жолды және деректер қорының
файлын көрсетіңіз.

Драйверді жүктеу.

Қосылымды орнату

Connection қосылу;

```
қосылу = DriverManager.getConnection(деректер қоры);
```

Негізгі деректер қоры клиенттер Клиенттер.sqlite «C:\temp» папкасында орналасқан. Бұл кестені қамтиды – ID атрибуттары бар Клиенттер мысалы (varchar түрі) және атауы (varchar түрі):

ID	Name
Filter	Filter
1	Maier
2	Knudsen
3	Kaiser
4	Franzen
5	Knobloch
6	Laufer

Кеңес:

SQLite деректер қоры тегін және портталатын деректер базасы болып табылады, ол деректер қорының толық логикасын және деректерді бір файлға қосады. Бұл Java бағдарламаларын жеке деректер қоры бар жіберуге мүмкіндік береді. Салыстырмалы түрде аз деректер көлемі бар шағын жобалар үшін бұл – Oracle немесе MySQL сияқты ірі ДББЖ-ның тамаша баламасы. «DB-Browser for SQLite» сияқты тегін құралдардың көмегімен бұл деректер қорын оңай басқаруға болады.

```
package db_zugriff_java;
import java.sql.*;
```

```
public class DBKipy {
public static void main(String[] args) {
try {
```

Драйвер және деректер көзі туралы ақпаратпен байланыс жолын анықтаңыз.

```
String ,деректер қоры = "jdbc:sqlite:/c:/temp/клиенттер.sqlite";
```

Драйверді жүктеу

```
Class.forName("org.sqlite.JDBC");
```

Статикалық getConnection әдісін қолдана отырып, байланыс нысанын сұраңыз.

Connection қосылым

```
DriverManager.getConnection(деректер қоры, "", "");
```

Қосылу нысаны SQL командасы үшін объект жасайды

```
Statement sqlкоманда = verbсоединения.createStatement();
```

ExecuteQuery әдісі SQL-сұратуды (немесе select командасын) орындайды және ResultSet түріндегі нысан түрінде нәтиже береді.

ResultSet нәтиже

```
sqlBefehl.executeQuery("SELECT * FROM клиенттер;");
```

ResultSet нәтижелер кестесін сұрау үшін әдістерді ұсынады, ал next әдісі жазбалардың бар-жоғын көрсетеді, ал getString әдісі қажетті бағаннан келесі жазбаны оқиды (мұнда Name)..

```
while (нәтиже.next() == true) {  
    System.out.println («Name: « +  
        нәтиже.getString ("Name"));  
}
```

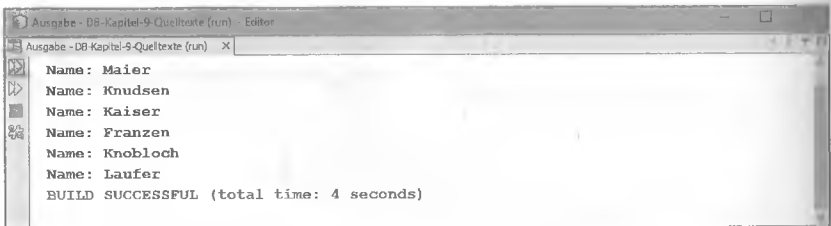
```
қосылу.close ();  
}
```

МАҢЫЗДЫ: деректер қорына қосылу қайтадан жабылуы керек.

```
catch (Exception e) {  
    System.out.println (e.getMessage ());  
}
```

НАЗАР АУДАРЫҢЫЗ: деректер қорына сұрау салу кезінде ерекшеліктерді өңдеу маңызды!

Іске қосылғаннан кейін келесі экран пайда болады:



Деректер қоры қатесіз сұралды, ал клиенттердің барлық аттары оқылып, көрсетілді.

Ескерту:

Қорытынды объектісі бар кесте баған мәндеріне кіру тиісті деректер түріне байланысты. Әрбір деректер түрі үшін бағанның индексі немесе бағанның атауын қабылдайтын қолайлы әдіс бар

- ▶ getString (int баған индексі)
- ▶ getString (String баған аты)

- ▶ `getDouble` (int баған индексі)
- ▶ `getDouble` (String баған аты)
- ▶ `getInt` (int баған индексі)
- ▶ `getInt` (String баған аты)
- ▶ ... қосымша түрлері

Мысалы, кестенің бірінші бағаны клиенттер `getInt` әдісін пайдаланып оқылуы мүмкін, себебі бұл бүтін сандық түрі: `while` (нәтиже.next() == true) { System.out.println(«ID: « + нәтиже.getInt(0)); }

9.1.4 SELECT-тен ерекшеленетін командаларды жою

Кез келген кестені оқу жоғарыда сипатталған нұсқауларды пайдаланып орындауға болады. Дегенмен, бағандарды таңдау емес, қою, өзгерту немесе жою керек болса, онда `executeUpdate` деп аталатын команданы алып тастауға болады. Қажетті SQL-команда символдық жолда бұрын құрылуға тиіс. Келесі мысалда клиенттер кестесіне жаңа жол енгізіледі, бар жол өзгереді және жолдардың бірі жойылады:

```

package db_қолжетімділік_java;
import java.sql.*;

public class DB_қолжетімділік {
    public static void main(String[] args) {
        try {
            Деректер орны жолы= "jdbc:sqlite:/c:/temp/клиенттер.sqlite";
            Class.forName("org.sqlite.JDBC");
            Connection қосылым =
                DriverManager.getConnection(деректер қоры, "", "");
            Statement sqlкоманда = қосылым.createStatement();

            int anzahl = sqlbefehl.executeUpdate («INSERT INTO Kunden
                ("IVALUES (7, 'Koenig' );");

            System.out.println("Енгізілген жолдар саны: "
                +саны);
        }
    }
}

```

SQL командасын болдырмау және қозғалған жолдар санын алу.

SQL-жолды енгізу командасы.

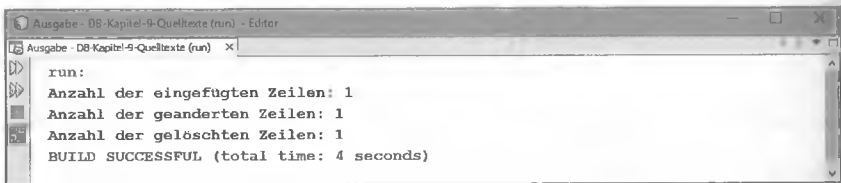
UPDATE командасы

```
саны = sqlBefehl.executeUpdate("Teri =' Клиенттер SET'  
WHERE Name = 'Knobloch;»);  
System.out.println("өзгертілген жолдар саны: "  
+ саны);
```

DELETE командасы

```
саны = sqlBefehl.executeUpdate("Delete = FROM Kunden  
WHERE Name' Kunden `;");  
  
System.out.println("жойылған жолдар саны: "  
+ саны);  
  
        қосылым.close();  
    }  
    catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    }  
}  
}
```

Іске қосылғаннан кейін SQL *No-Select* үш командасы беріледі және қозғалған жолдар саны шығарылады.



```
Ausgabe - DB-Kapitel-9-Quelltexte (run) - Editor  
Ausgabe - DB-Kapitel-9-Quelltexte (run) x |  
run:  
Anzahl der eingefügten Zeilen: 1  
Anzahl der geänderten Zeilen: 1  
Anzahl der gelöschten Zeilen: 1  
BUILD SUCCESSFUL (total time: 4 seconds)
```

Салыстыру үшін: SQL командаларына дейін және кейін Customers (клиенттер) кестесі:

Бұрында:

ID	Name
Filter	Filter
1	Maier
2	Knudsen
3	Kaiser
4	Franzen
5	Knobloch
6	Laufer

Нәтижесінде:

ID	Name
Filter	Filter
1	Maier
3	Kaiser
4	Franzen
5	Knoblauch
6	Laufer
7	Koenig

9.1.5 Метадеректерді алу

Деректер қорына икемді қол жеткізу үшін деректер қоры, драйвері және деректер қорының кестелері туралы ақпаратты алу маңызды. Бұл ақпарат метадеректі деректер қорының кластарын пайдалана отырып алынуы мүмкін. Осы ақпарат негізінде бағдарлама одан әрі шешім қабылдай алады. Мысалы, деректер қорындағы кесте үнемі жаңа бағандарды ала алады. Бұл кестеде қол жетімділік икемді болуы керек, әйтпесе бұл қателікке немесе жұмыста жаңылысуға әкеледі. Екі сынып метадеректерді оқуға маңызды: DatabaseMetaData және ResultSetMetaData.

Келесі мысалда жоғарыда келтірілген мысалдан SQLite деректер қорын оқу үшін екі сыныпты қалай пайдалану керектігі көрсетілген:

```
package db_қолжетімділік_java;
import java.sql.*;

public class DBқолжетімділік{
    public static void main (String[] args) {
        try {
            Деректер қорының жолы = "jdbc:sqlite:/c:/temp/ клиенттер.
sqlite";
            Class.forName ("org.sqlite.JDBC");

            Connection қосылым =
                DriverManager.getConnection (деректер қоры, "", "");
```

Connection нысаны метадеректер дерекқорын сұратады.

```
// Деректер қорының метадеректері DatabaseMetaData
System.out.println("Деректер қорының мета деректері");
System.out.println("Деректер қоры атауы : "
    + dbinfos.getDatabaseProductName());
System.out.println("Драйвер атауы : "
    + dbinfos.getDriverName());
System.out.println();

// Метадеректер кестелері
Statement sqlкоманда = қосылым.createStatement();
ResultSet нәтиже=
    sqlBefehl.executeQuery("SELECT * FROM Клиенттер;");
```

ResultSet нысаны арқылы метадеректі кестелер сұралады.

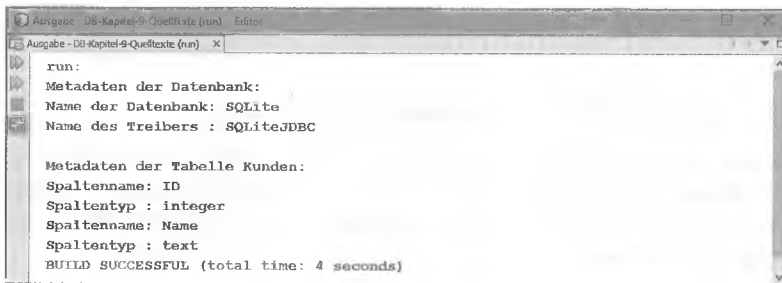
```
ResultSetMetaData tb тbinfos = нәтиже.getMetaData();
```

```
System.out.println("Клиенттер кестесінің метадеректері:");
for ( int i = 1; i <= tbinfos.getColumnCount(); i++)
```

Бағандар санын сұрау.
МАҢЫЗДЫ: индекс 1-ден басталады!

```
System.out.println("Баған аты: " +
    tbinfos.getColumnLabel(i));
System.out.println("Баған түрі: " +
    tbinfos.getColumnTypeName(i));
}
қосылым.close();
}
catch (Exception e) {
    System.out.println(e.getMessage());
}
}
```


NetBeans іске қосылғаннан кейін келесі экран пайда болады:



```
run:
Metadaten der Datenbank:
Name der Datenbank: SQLite
Name des Treibers : SQLiteJDBC

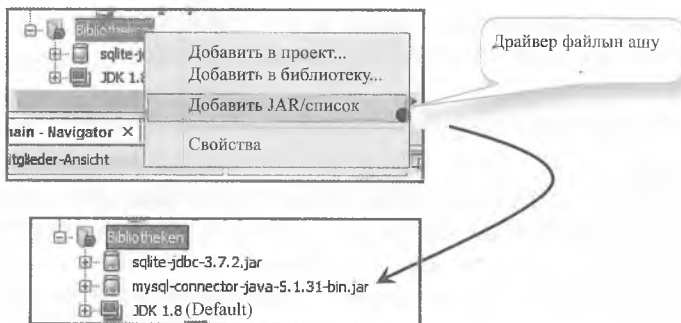
Metadaten der Tabelle Kunden:
Spaltenname: ID
Spaltentyp : integer
Spaltenname: Name
Spaltentyp : text
BUILD SUCCESSFUL (total time: 4 seconds)
```

9.2 БАСҚА ДЕРЕКТЕР ҚОРЫНА ЖҮГІНУ

Көптеген деректер қоры драйверлері бар, сондықтан оларға Java және JDBC арқылы қол жеткізуге болады. Әдетте, драйверді тек жүктеп, жобаға қосу керек. Осыдан кейін драйвер әдетте тіркелуі мүмкін.

9.2.1 Драйверді қосу

Қажетті драйверді жүктегеннен кейін (мысалы, `mysql-connector-java-XXX-bin.jar`) Java-мұрағат файлы жобаға біріктіріледі:



MySQL драйверін сәтті қосқаннан кейін Class класты қосылым орнатылуы мүмкін:

Деректер қоры жолы = `<jdbc:mysql://Servername/Datenbank>`;

```

Class.forName («com.mysql.jdbc.Driver»);
Connection қосылым;
қосылу = DriverManager.getConnection (деректер қоры

```

9.2.2 Басқа деректер қорының драйверлері

Келесі кестеде Жалпы деректер қорын шолу және тиісті Java драйверінің аты көрсетілген. Тиісті драйвер файлы жоғарыда сипатталғандай провайдердің веб-сайтынан жүктелуі тиіс немесе басқаша қолжетімді.

Деректер қоры	Java драйверінің атауы
Firebird (тегін ДК, Borland Interbase ізбасары)	org.firebirdsql.jdbc.FBDriver
DB2 (IBM)	com.ibm.db2.jcc.DB2Driver
Informix (IBM)	com.informix.jdbc.IfxDriver
Microsoft SQL-Server	com.microsoft.jdbc.sqlserver.SQLServerDriver
MySQL	com.mysql.jdbc.Driver
Oracle	oracle.jdbc.OracleDriver
SQLite	org.sqlite.JDBC

Ескерту:

Деректер қорын интеграциялау алдында, әдетте, интернетте жан-жақты іздеу немесе деректер қоры бойынша тиісті әдебиеттерді шолу баламалары қалдырылмайды.

9.3 9-ТАРАУҒА АРНАЛҒАН ЖАТТЫҒУЛАР

1-тапсырма

Компаниялардың бірінде, комиссиялық сауда өкілдері қарапайым деректер қорында (осы мысалда – SQLite деректер қорында) сақталады. Осы деректер қорын кестемен және тиісті мазмұнмен жасаңыз. Кейіннен кестеден келесі статистикалық деректерді оқу қажет болады:

- ▶ ең жоғары комиссиямен сауда өкілі
 - ▶ ең аз комиссиясы бар сауда өкілі
 - ▶ барлық комиссиялардың сомасы
 - ▶ барлық комиссиялардың орташа мәні
- SQLite деректер қорындағы кесте:

Имя	Фамилия
Filter	Filter
Maier	1200.0
Knudsen	800.0
Laufer	600.0
Kaufhold	1400.0
Mager	350.0
Kaiser	1900.0
Lehmberg	950.0
Katernberg	700.0

SQLite деректер қорын ыңғайлы басқару тегін «DB-Browser for SQLite» құралы арқылы орындалуы мүмкін.

Ескерту:

Параметрлерді есептеу тиісті SQL функцияларымен орындалуы мүмкін (мысалы, SUM, AVG, MIN және MAX) немесе Java бағдарламалық логикасы арқылы жүзеге асырылған.

Іске қосылғаннан кейін экранға шығару келесідей болуы мүмкін:

```

Ausgabe - DB Aufgabepool (run) - Editor
Ausgabe - DB Aufgabepool (run) x |
run:
Mitarbeiter mit der höchsten Provision: Kaiser
Mitarbeiter mit der geringsten Provision: Mager
Summe der Provisionen: 7900.0
Durchschnitt der Provisionen: 987.5
BUILD SUCCESSFUL (total time: 7 seconds)

```

2-тапсырма

Бастапқы шарттар:

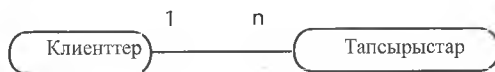
Компанияның бірінде клиенттің тапсырысының деректері деректер қорының екі кестесінде сақталады (мысалы, SQLite). Сонымен қатар, клиент тапсырысының мәліметтерін нақты көрсете алатын, графикалық интерфейсі бар қарапайым Java-қосымшасын әзірлеу қажет.

Базалық кестелер келесідей:

Клиенттер кестесі:

ID	Name
Filter	Filter
1	Maier
2	Knudsen
3	Kaiser
4	Franzen
5	Knobloch
6	Laufer

Кестелердің байланысы:



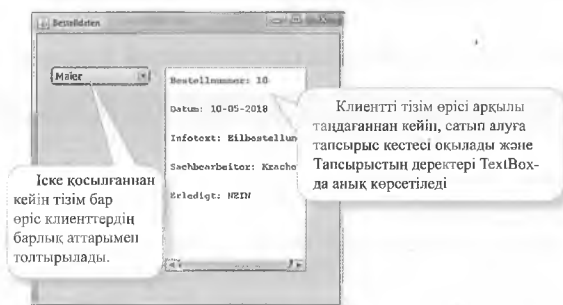
Тапсырыстар кестесі:

Клиенттер	Тапсырыс коды	Күні	Ақпарат	Қызметкер	Орындалуы
Filter	Filter	Filter	Filter	Filter	Filter
1	10	10-05-2018	Eilbestellung	Kracher	true
3	11	10-06-2018	gefährliche Fr...	Klauber	true
4	12	10-07-2018	guter Kunde	Hütter	false

Тапсырыс кестесі клиенттің сыртқы идентификаторы бар, ол екі кестенің «1:n» байланысын жүзеге асырады.

Тапсырманы қою:

Тиісті деректер қорында (мысалы, SQLite) екі кестені жасап, кестені тиісті мәндермен толтырыңыз. Содан кейін Деректер қорына жүгінетін және кестелердің мәліметтерін оқитын Java қосымшасын әзірлеңіз. Қолданба интерфейсі осылай болуы керек:



10 NET және C# көмегімен деректер қорына кіру#

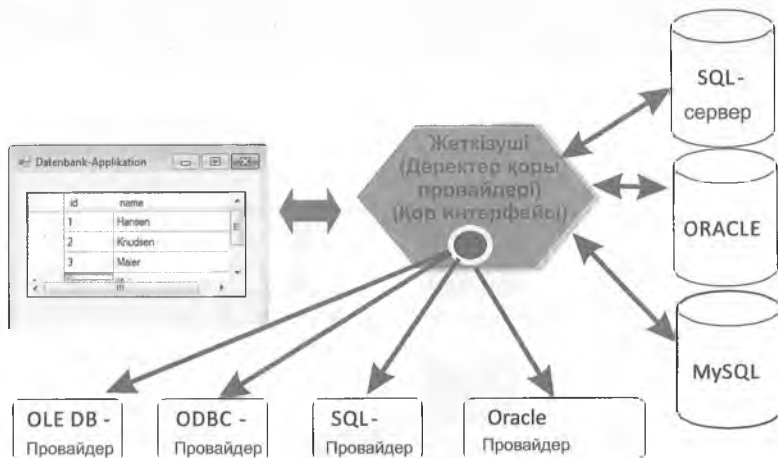
10.1 NET ЖӘНЕ C КӨМЕГІМЕН ДЕРЕКТЕР

БАЗАСЫНА КІРУ

C# бағдарламалау тілі деректерді тұрақты сақтау немесе оқу үшін көптеген файлдық операцияларды ұсынады. Үлкен көлемдегі немесе күрделі құрылымдағы деректер қорында оларды сақтау мүмкіндігін қарастырудың мағынасы бар (алдыңғы тарауда Java бағдарламалау тілінде сияқты). NET Framework жүйесі ыңғайлы нұсқаларды ұсынады. SQL сұрау тілі де маңызды рөл атқарады.

10.1.1 .NET Framework астында деректер қорын қосу

.NET Framework деректер қорына қосылу үшін көптеген кластарды ұсынады. Бұл сыныптар жалпы термин астында жиналған ADO.NET. ADO ActiveX Data Objects деп аталады және қолданыстағы Microsoft технологиясының кеңейтілуі болып табылады. ADO.NET SQL Server немесе OLE DB және ODBC деректер көздері сияқты деректер көздеріне қатынауды қамтамасыз етеді. Келесі суретте ADO.NET негізгі принципі көрсетілген:



Жеке жеткізушілер (провайдерлер) деректер базасына қосудың белгілі бір түріне арналған:

► **OLE DB провайдері:** OLE DB Object Linking and Embedded Database деп аталады және Microsoft Office қосымшаларында

қолданылатын әдіс болып табылады. Мысалы, бастапқы электрондық кестедегі өзгерістер әрқашан Word электрондық кестесінде (және керісінше) көрінуі үшін Excel электрондық кестесін Word құжатына қосуға болады. OLE DB провайдері деректер қоры үшін қол жетімді өнім беруші (мысалы, ACCESS) болған кезде пайдаланылуы мүмкін.

► SQL-дің ең көп саны, яғни SQL-дің ең көп саны, яғни SQL-дің ең көп саны, яғни, SQL-дің ең көп саны, яғни, SQL-дің ең көп саны, яғни, SQL-дің ең көп саны, яғни, SQL-дің ең көп саны, яғни SQL-дің ең көп саны. Әрбір деректер қоры тек бір ODBC интерфейсіні ұсынуы керек, сондықтан Windows қосымшасында қолданылуы мүмкін.

► SQL: бұл өнім беруші Microsoft SQL Server кіру мүмкіндігін береді.

► XORACLE: бұл өнім беруші ORACLE деректер қорына қол жеткізу үшін Функционалдық мүмкіндіктерді ұсынады.

10.1.2 Провайдерлерді пайдалану және байланыс орнату

Деректер қорына қосылу үшін тиісті деректер қорының жеткізушісі болуы тиіс. Кейбір драйверлер Visual C # (мысалы, OLE DB жеткізушісі) стандартты орнатуында бар. Басқа жеткізушілер тиісті өндірушілерден сатып алынуы тиіс. Содан кейін Connection класы нысанымен нақты байланыс жасалады. Деректер қорына байланысты Пайдаланушының аты мен пароль көрсетіледі. Command нысанын пайдаланғанда, сұрау DataReader кемегімен іске қосылуы және оқылуы мүмкін. Келесі суретте байланыс көрсетілген:



10.1.3 ACCESS деректер қорына кіру мысалы

- ▶ келесі бөлімде Access деректер қорына OLE DB жеткізушісімен қатынасу сипатталған. Бұл принцип Microsoft SQL Server немесе MySQL сияқты реляциялық деректер қорының басқа жүйелеріне көшірілуі мүмкін. Деректер қорына қосылу OleDbConnection нысанының көмегімен жүзеге асырылады, SQL-пәрменді беру-OleDbCommand нысанының көмегімен, нәтижені оқу – OleDbDataReader типті нысанның көмегімен жүзеге асырылады.
- ▶ Келесі мысалдарда пайдаланылатын х даму ортасы Microsoft Visual C # 2017 тегін нұсқасы.

Келесі бастапқы кодта қалтада қол жетімді ACCESS Customers. accdb деректер қорына қосылу көрсетіледі (мұнда: C:\temp). Бұл ДБ-да кесте – мысалы бар customers атрибутымен id (TypZahl) және Teri, көшесі, E / мекен, телефон (Text):



The screenshot shows a Microsoft Access table named 'Kunden'. The table has five columns: 'id', 'name', 'strasse', 'ort', and 'telefon'. The data is as follows:

id	name	strasse	ort	telefon
1	Hansen	Baumallee 1	Hamburg	123456
2	Knudsen	Sonnenstr.4	Berlin	654321
3	Albers	Paulistr. 8	Hamburg	111222

At the bottom of the screenshot, there is a status bar showing 'Datensatz: 1 von 3', 'Kein Filter', and a search button labeled 'Suchen'.

```
System ;
System .Data ;
System .Data .OLEDB ;
```

Қажетті атау кеңістігін қосыңыз!

```
DB_Zugriff_CSharp
```

```
CDBZugriff
```

Провайдер мен дерек көз ерекшелігінің қосылымын көрсету

```
Main( [] args)
```

```
verb indungsstring =
"Provider=Microsoft.ACE.OLEDB.12.0 ;
Data Source=C:\Temp\Kunden.accdb" ;
```

Бір OLEDB

қосылымына сілтеме

```
OLEDBConnection dBVerbindung = ; OLEDB коман-
OLEDBCommand befehl = ; дасына сілтеме
OLEDBDataReader datenleser = ;
offen = ;
```

МАҢЫЗДЫ: OLEDB
кателерді өңдеу деректерді оқу
сілтемесі

```
dBVerbindung =
OLEDBConnection(verb indungsstring) ;
dBVerbindung .Open () ;
offen = ; Байланыс  
үлгісі
```

Белгіні ДК-ның
орнатыңыз ашылуы

Команда
нысанын құру

```
befehl = dBVerbindung .CreateCommand () ;
befehl .CommandText = "SELECT * FROM Kunden" ;
```

SQL пәрменінің деректерін оқу
данасының негізінде

SQL командасын
тағайындау (кестенің
барлығы)

```
datenleser = befehl .ExecuteReader () ; Дәйекті оқу
```

```
{ (datenleser .Read ())
```

```
Console .WriteLine ("Name : " + datenleser .GetString (1)) ;
```

GetString әдісі ағымдағы жол мәнін және берілген баған индексін қайтарады


```

catch (Exception исключение)
{
    Console.WriteLine("деректер қоры қатесі: "
        + ерекшелік.Message);
}
finally
{
    if (offen == true) dBҚосылым.Close();
}
}
}

```

Егер `offen` логикалық айнымалы мәні болса, байланыс жабылады.

Іске қосылғаннан кейін клиент кестесі оқылып, 2-бағанның мәні (индексі 1) деректерді оқу құрылғысын пайдалана отырып қадамды көрсетеді:

```

C:\WINDOWS\system32\cmd.exe
Name: Hansen
Name: Knudsen
Name: Albers
Drücken Sie eine beliebige Taste . . .

```

Ескерту:

Деректер оқу құрылғысы арқылы кесте бағандарының мәндеріне кіру тиісті деректер түріне байланысты. Әрбір деректер түрі үшін қолайлы әдіс қолжетімді:

- ▶ `GetDateTime` (баған индексі)
- ▶ `GetString` (баған индексі)
- ▶ `GetInt32` (баған индексі)
- ▶ ... қосымша типтер

Мысалы, `customer` кестесінің бірінші бағаны `GetInt32` () әдісі арқылы оқылуы мүмкін, себебі бұл бүтін сан түрі (`ACCESS` түрі):

```

while (деректерді оқу.Read())
{
    Console.WriteLine («Бірінші Баған: "
+ деректерді есептеуіш.GetInt32(0));
}

```

`ACCESS` саны өрісінің деректерінен бірінші бағанды оқу.

10.1.4 SELECT-ке ұқсамайтын командаларды жою

Кез келген кестені оқу жоғарыда сипатталған нұсқауларды пайдаланып орындауға болады. Дегенмен, бағандарды таңдау емес, қою, өзгерту немесе жою керек болса, ExecuteNonQuery командасын жоюға болады. Қажетті SQL-команда символдық жолда бұрын құрылуға тиіс. Келесі мысалда клиенттер кестесіне жаңа жол енгізіледі, бар жол өзгереді және жолдардың бірі жойылады:

```
using System;
using System.Data;
using System.Data.OleDb;

namespace DB_Қолжетімділік_CSharp
{
    class CDBҚолжетімділік
    {
        static void Main(string[] args)
        {
            string қосылым жолы =
                «Provider=Microsoft.ACE.OLEDB.12.0;
                Data Source=C:\\Temp\\Клиенттер.accdb»;
            OleDbConnection dBҚосылым = null;
            OleDbCommand = null;
            bool offen = false;
            int сан = 0;
            try
            {
                dBҚосылым = new
                    OleDbConnection (қосылым жолы);
                dBҚосылым.Open();
                offen = true;
                команда = dBҚосылым.CreateCommand();

                команда.CommandText = " INSERT INTO Клиенттер
                VALUES (4, 'Кениг', 'Зеештр. 5',
                'Гамбург', '45621' );";
            }
            catch { }
        }
    }
}
```

Провайдердің сипаттамасымен және деректер көзімен байланыс жолын анықтаңыз.

SQL-жолды енгізуге арналған команда.

SQL командасын болдырмаңыз және қозғалған жолдардың санын алыңыз.

```
саны=команда.ExecuteNonQuery();  
Console.WriteLine ("саны  
қосылған жолдар саны: "+ саны);
```

Команда UPDATE

```
команда.CommandText ="UPDATE Клиенттер  
SET Телефон='11111' WHERE name =  
Хансен'";  
саны=команда.ExecuteNonQuery();  
Console.WriteLine ("өзгерген жолдар:  
"+ саны);
```

Команда DELETE

```
команда.CommandText = «DELETE FROM Клиенттер  
WHERE name = 'Кнудсен'";
```

```
саны=команда.ExecuteNonQuery();  
Console.WriteLine ("жойылған  
жолдар: "  
+саны);  
Console.WriteLine();
```

```
}  
catch(Exception шығару)  
{
```

```
Console.WriteLine("деректер қоры қатесі: "  
+ шығару.Message);
```

```
finally
```

Деректер базасына
қосылуды жабу

```
{  
if (offen == true) dBҚосылым.Close();  
}
```

Іске қосылғаннан кейін Select-тен өзгеше үш SQL-команда беріледі және әрбір командадан кейін қозғалған жолдар саны шығарылады:

Салыстыру үшін: SQL командаларына дейін және кейін Customers (клиенттер) кестесі:

Бұрынырақта:

```
C:\WINDOWS\system32\cmd.exe

Anzahl der eingefügten Zeilen: 1
Anzahl der geänderten Zeilen: 1
Anzahl der gelöschten Zeilen: 1

Drücken Sie eine beliebige Taste . . .
```

id	name	strasse	ort	telefon
1	Hansen	Baumallee 1	Hamburg	123456
2	Knudsen	Sonnenstr.4	Berlin	654321
3	Albers	Paulistr. 8	Hamburg	111222

Соңында:

id	name	strasse	ort	telefon
1	Hansen	Baumallee 1	Hamburg	11111
3	Albers	Paulistr. 8	Hamburg	111222
4	König	Seestr. 5	Hamburg	45621

Жаңа жазба автоматты түрде ACCESS ID 4 алады. ACCESS жазбаларды жою және кірістіргеннен кейін толық идентификаторларды елемейді.

10.1.5 DataAdapter және деректер жиынтығы

Бұрын деректер қоры ашылды және SQL тиісті командаларын пайдалану арқылы ретімен немесе өзгертілді. Мұндай тәсіл қолданылады, бірақ ол өте ыңғайлы емес. Осы себепті dataset класындағы арнайы нысанда сұраудың толық нәтижесін сақтауға болады. Бұл нысан де-

рекер қорына тәуелсіз өңделеді және тек содан кейін ғана деректер қорымен синхрондалады. Сондықтан мұндай нысандар әлі де біріктірілмеген деп аталады. Осы нысандар мен деректер қоры арасында деректерді дұрыс алмасу үшін адаптер-нысандар қажет. Келесі сурет осы өзара байланысты бейнелейді:



Келесі мысалда DataSet және DataAdapter пайдалану көрсетілген:

```
using System;
using System.Data;
using System.Data.Ole
```

Қажетті аттар массивтерін біріктіру

```
namespace DB_Ұқсат_CSharp
{
```

```
    КлассCDBҰқсат
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            string қосылу жолы =
            «Provider=Microsoft.ACE.OLEDB.12.0;
            Data Source=C:\\Temp\\Клиенттер.accdb»;
            OleDbConnection dВқосылым= null;
            OleDbCommand = null;
            bool offen = false;
```

Провайдердің сипаттамасымен және деректер көзімен байланыс жолын анықтаңыз

```
            try
```

```
            {
```

```
                dВқосылым =
                new OleDbConnection(қосылу
                жолы);
```

```
dbҚосылым.Open();  
offen = true;
```

```
команда = dbҚосылым.CreateCommand();  
команда.CommandText = "SELECT *  
FROM Клиенттер";
```

OleDbDataAdapter нысанын құру және
OleDbCommand нысанын беру

```
OleDbDataAdapter da = new OleDbData-  
Adapter(команда);
```

DataSet объектісін
құру

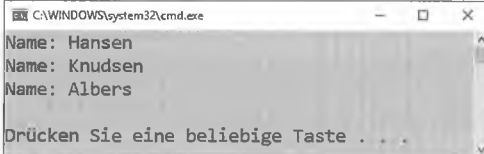
```
DataSet ds = new DataSet();  
da.Fill(ds);
```

Fill() әдісі DataSet нысанын жоғарыдағы
сұрауға сәйкес келетін мәліметтер базасындағы
жазбалармен толтырады.

```
for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
```

```
Console.WriteLine(  
ds.Tables[0].Rows[i]["name"] .  
ToString());
```

Сұрау нәтижесі DataSet нысан кестелерінің массиві арқылы жіберілуі мүмкін. Со-
нымен қатар, қажетті жол мен баған көрсетіледі:



```
C:\WINDOWS\system32\cmd.exe  
Name: Hansen  
Name: Knudsen  
Name: Albers  
Drücken Sie eine beliebige Taste ...
```

```
ds.Tables[0].Rows[0]["name"] = "Laufer";
```

Содан кейін деректерді өзгерту
қарапайым тағайындау арқылы
жүзеге асырылады!

```

DataRow саны = ds.Tables[0]. NewRow();
саны["id"] = 10;
саны["name"] = "Kaiser";
ds.Tables[0].Rows.Add(строка);

```

Жазбаны қосу
DataRow нысаны
мен қосу Add()
әдісі арқылы жүзеге
асырылады.

```
ds.Tables[0].Rows[1].Delete();
```

Жою Delete әдісі
арқылы жүзеге
асырылады!

```

OleDbCommandBuilder cmb =
    new OleDbCommandBuilder(da);
da.Update(ds);

```

Деректер базасымен синхрондау үшін қажетгі Update пәрмендерін ұсынатын
CommandBuilder нысаны қажет.

```

}
catch (Exception шығару)
{
    Console.WriteLine("деректерқорықатесі: "
        + шығару.Message);
}
finally
{
    if (offen == true) dbҚосылым.Close();
}
}
}
}

```

Синхрондалғаннан кейін клиенттер кестесі келесідей болады:

id	name	strasse	ort	telefon
1	Laufer	Baumallee 1	Hamburg	123456
3	Albers	Paulistr. 8	Hamburg	111222
10	Kaiser			

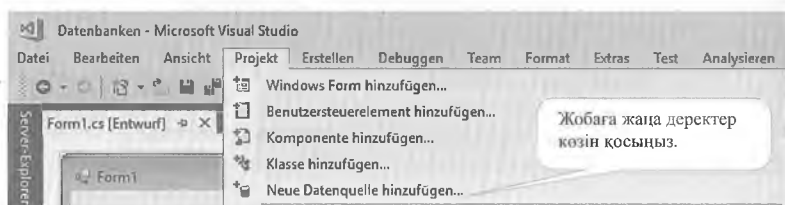
Detensatz: 11 von 3 | Kein Filter | Suchen

10.2 VISUAL C ДЕРЕКТЕР ҚОРЫНЫҢ ШЕБЕРІН ПАЙДАЛАНУ

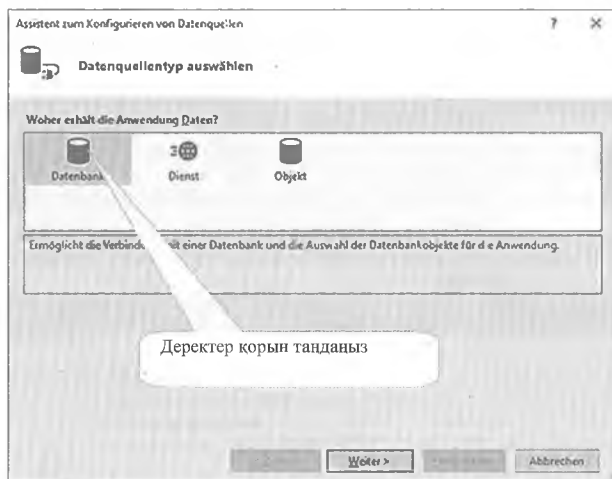
10.2.1 Деректер қорын қосу

Visual C # өңдеу ортасы жобаға деректер қорын автоматты түрде біріктіру үшін пайдалануға болатын шебермен жұмысты ұсынады. Бұл жоғарыда сипатталған әдістермен салыстырғанда әзірлеу уақытын айтарлықтай қысқартады. Дегенмен, әзірлеуші аз еркіндік бар, өйткені шебер автоматты түрде бастапқы кодтың үлкен санын жасайды.

Бірінші кезеңде жобаға деректер көзін қосу қажет. Бұл *"Projekt» мәзірінің → Жаңа дерек көзін қосу арқылы қол жеткізіледі..."*:



Келесі қадамда деректер қоры таңдалады:



Содан кейін деректер қорының моделін таңдаңыз:



Осыдан кейін жаңа қосылымды таңдаңыз:



Деректер көзін қажетті деректер қорына өзгертіңіз (мұнда ACCESS деректер қоры):

Verbindung hinzufügen

Geben Sie Informationen zum Verbinden mit der ausgewählten Datenquelle ein, oder klicken Sie auf "Ändern", um eine andere Datenquelle und/oder einen anderen Anbieter auszuwählen.

Datenquelle:
Microsoft Access-Datenbankdatei (OLE DB)

Name der Datenbankdatei:
C:\Temp\Kunden.accdb

Bei der Datenbank anmelden

Benutzername:

Kennwort:

Kennwort speichern

Microsoft Visual Studio

1 Die Testverbindung war erfolgreich.

Қосылысты тексергеннен кейін оны «OK»-ге басу арқылы растауға болады. Содан кейін бастапқы диалогтың «жалғастыру» түймесі арқылы процесті аяқтауға болады.

Алайда бұған дейін шебер деректер қорын көшірме ретінде пайдалануды сұрайды:

Microsoft Visual Studio

? Die ausgewählte Verbindung verwendet eine lokale Datendatei, die nicht im aktuellen Projekt vorhanden ist. Möchten Sie die Datei in das Projekt kopieren und die Verbindung ändern?

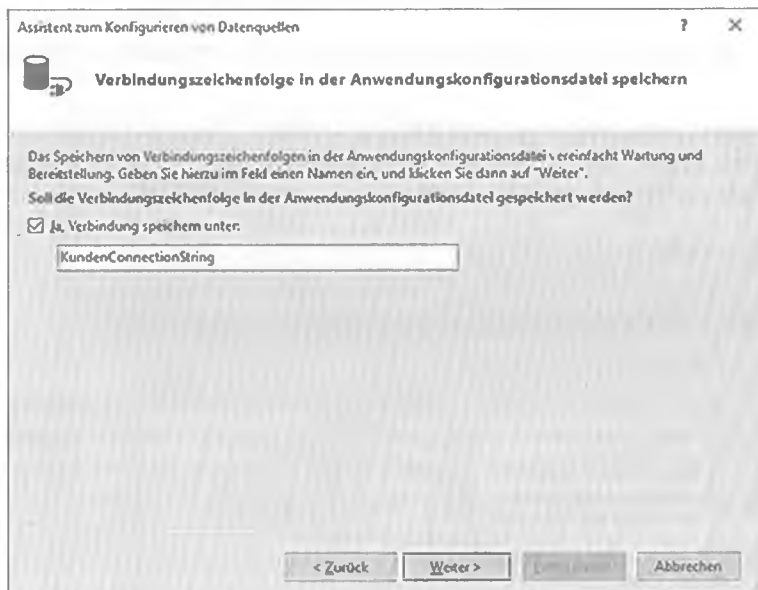
Wenn Sie die Datendatei in Ihr Projekt kopieren, wird sie bei jedem Ausführen der Anwendung in das Ausgabeverzeichnis des Projekts kopiert. Drücken Sie F1, um Informationen darüber zu erhalten, wie Sie dieses Verhalten steuern können.

Microsoft онлайн анықтамасы осы опция туралы ақпарат береді:

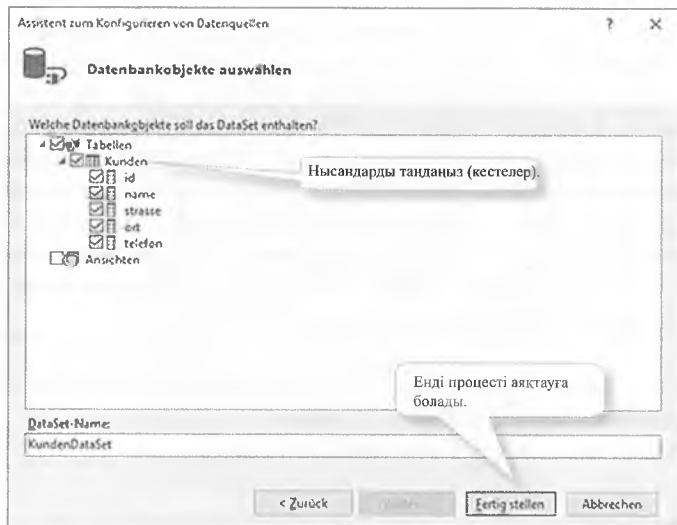
Жергілікті деректер қорының файлы файл түрінде жобаға қосылуы мүмкін. Қолданбаны жергілікті деректер қорының файлына алғаш рет қосқан кезде, жобаңыздағы деректер қорының көшірмесін жасауға немесе оның ағымдағы орналасқан жерінде деректер қорының файлына қосылуға болады. Бар файлға қосылған кезде, байланыс кез келген басқа қашықтағы деректер қорына сияқты орнатылады және деректер қоры файлы өзінің бастапқы орнында қалады. Егер сіз деректер қорын жобаңызға көшіргіңіз келсе, Visual Studio деректер қоры файлының көшірмесін жасайды, оны жобаға қосады және ол деректер қоры файлының бастапқы орнына емес, жобадағы деректер қорына көрсететіндей етіп қосылымды өзгертеді.

Бұл мысалда – «жоқ» таңдалған, бағдарлама бастапқы деректер қорымен (көшірмесіз) жұмыс істейтін болады.

Содан кейін байланыс атауымен сақталуы мүмкін:

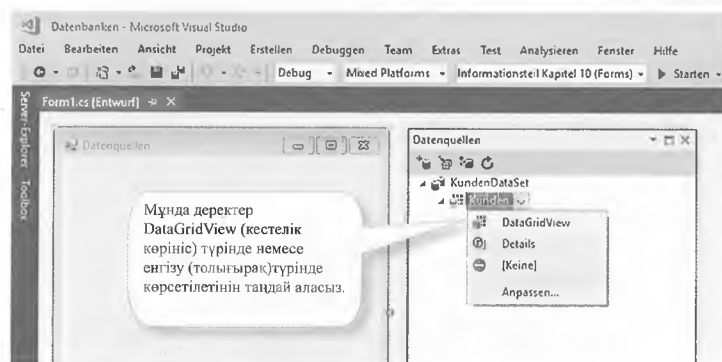


Енді деректер қорының нысандарын таңдауға болады. Бұл жағдайда клиенттердің толық кестесі таңдалады:

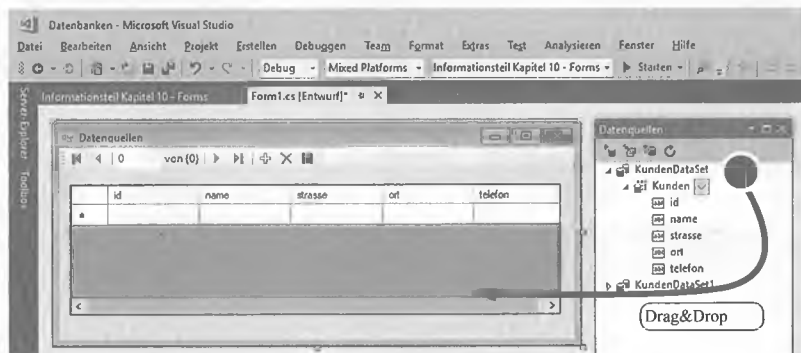


10.2.2 Windows Forms басқару элементтерін автоматты түрде қосу

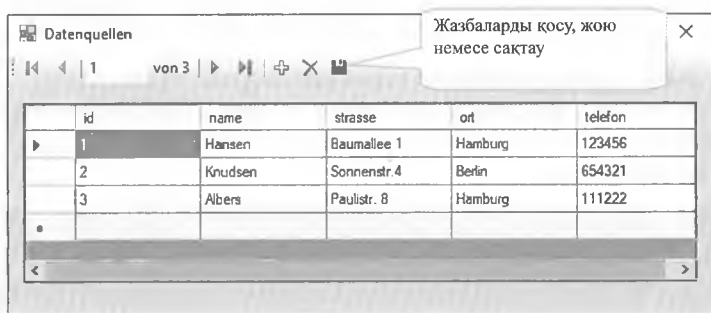
Деректер көзін көрсету үшін Көрініс мәзірін → Басқа терезелер → Деректер көзін пайдаланыңыз, содан кейін оны Forms қосымшасы үшін пайдаланыңыз.



Енді "Клиенттер" кестесін қажетті көріністе (мұнда datagridview) үлгіге Drag&Drop функциясы арқылы жылжытуға болады. Шебер автоматты түрде тиісті басқару элементін және деректер қоры мен кестемен байланысын жасайды:



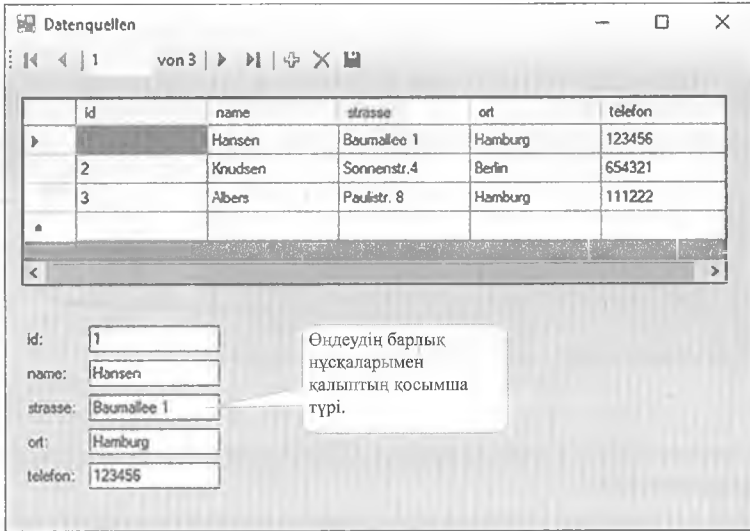
Іске қосылғаннан кейін деректер қоры кестесінің функционалдық көрінісі қолжетімді.



Жазбалар өзгертілуі және жойылуы мүмкін. Сондай-ақ жаңа жазбаларды қосуға болады. Әрине, әзірлеуші элементті бағдарламалау үшін қол жетімді көптеген оқиғалар мен функцияларға ие. Мысалы, ұяшықты екі рет шертіп, ұяшықтың мазмұнын шертуге жауап беретін CellContentClick оқиғасының еңдеушісін жасауға болады.

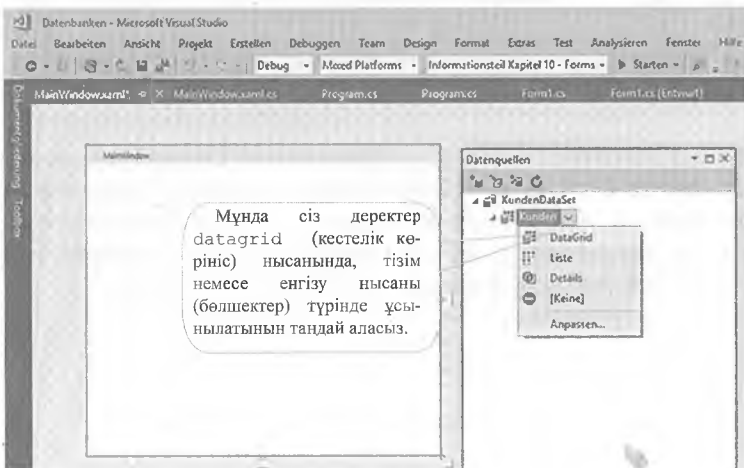
```
private void клиенттерDataGridView_CellContentClick
    (object sender,
    DataGridViewCellEventArgs e)
{
    MessageBox.Show ("ҰЯШЫҚТЫҢ МАЗМҰНЫН БАСУ!");
}
```

GridView көрінісіне қосымша, толық көріністі (*Drag&Drop*) пішінге жылжытуға болады. Бұл пайдаланушыға тек кестені ғана емес, пішімді де көруге мүмкіндік береді. Деректерді көрсету автоматты түрде синхрондалады:

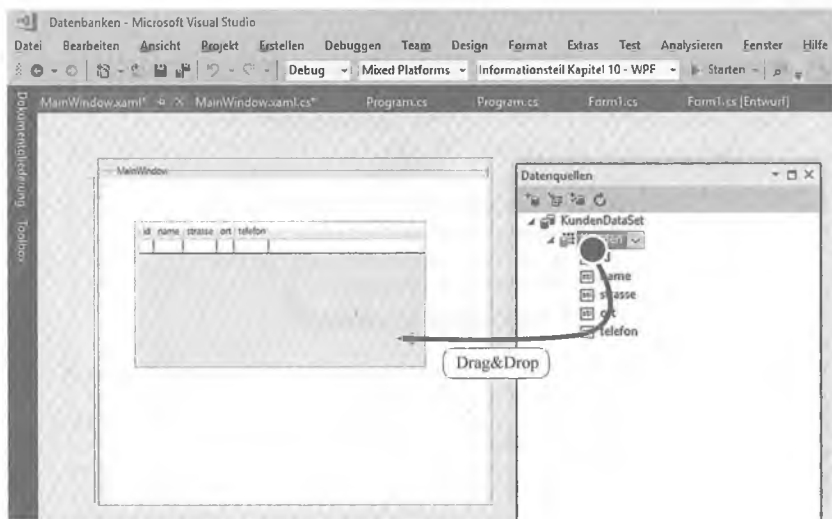


10.2.3 WPF басқару элементтерін автоматты түрде қосу

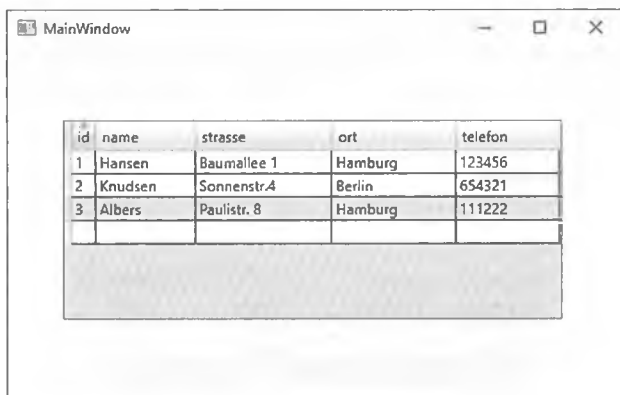
Деректер көзін көрсету үшін *Көрініс мәзірін* → *Басқа терезелер* → *Деректер көзі* пайдаланыңыз, содан кейін оны WPF қосымшасы үшін пайдаланыңыз.



Енді Sustomer (клиенттер) кестесін қалаған түрде (мұнда DataGrid) терезеге жылжытуға болады. Шебер автоматты түрде тиісті басқару элементін және деректер қоры мен кестемен байланысын жасайды:



Жібергеннен кейін деректер қорының көрінісі қолжетімді болады.



Байланыстағы XAML коды келесідей:

```
<DataGrid x:Name="клиенттерDataGrid" AutoGenerateColumns="False"
EnableRowVirtualization="True" ItemsSource="{Binding}" Margin="
45,59,51,64"
RowDetailsVisibilityMode="VisibleWhenSelected">
```

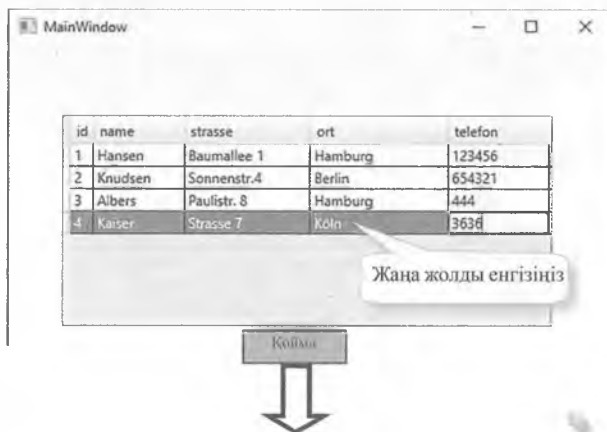
бағандар сәйкес келеді

```
<DataGrid.Columns>
<DataGridTextColumn x:Name="idColumn
Binding="{Binding id}" Header="id"
Width="SizeToHeader"/>
```

WPF деректерін байланыстыру

```
<DataGridTextColumn x:Name="nameColumn" Binding="{ Binding
name}" Header="name" Width="SizeToHeader"/>
<DataGridTextColumn x:Name="улицаColumn" Binding="{ Binding
улица}" Header="көше" Width="SizeToHeader"/>
<DataGridTextColumn x:Name="е.мекенColumn" Binding="{ Binding
орт нас.пункт}" Header="е.мекен" Width="SizeToHeader"/>
<DataGridTextColumn x:Name="телефонColumn" Binding="{ Binding
телефон}" Header="телефон" Width="SizeToHeader"/>
</DataGrid.Columns>
</DataGrid>
```

Кестеге жаңа мәндер қосылуы мүмкін, бірақ автоматты синхрондау орындалмайды. Бұл үшін файл кодында `datenadapterkiasse` класының белгілі бір әдісі туындауы тиіс. Келесі оқиға әдісі синхрондау әдісін көрсетеді.




```

private void Button_Click(object sender, RoutedEventArgs e)
{
    try
    {
        OleDbCommandBuilder cmbКлиенттер = new
        OleDbCommandBuilder Adapter);

        клиентыDataSetКлиенттерTableAdapter.
            Update (КлиенттерDataSet);
        MessageBox.Show("Жаңарту сәтті өтті");
    }
    catch (System.Exception ex)
    {
        MessageBox.Show(«Жаңарту қате:» + ex);
    }
}

```

CommandBuilder нысанының данасын жасағанда, адаптердің данасы жаңарту үшін барлық қажетгі ақпаратты алады!

Жаңартуды орындаңыз!

Алайда, код файлында бастапқы код деректер қорына қосылу сілтемелері атрибуттар ретінде жасалатындай етіп өзгертілуі тиіс. Басқа жағдайда адаптер объектісіне кіру жоғарыда келтірілген әдіс ретінде мүмкін емес.

Деректер қорына .NET және C арқылы кіру#

```

public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }

    private КлиенттерDataSetклиенттерDataSet;

    private КлиенттерDataSetTableAdapters.КлиенттерTableAdapter
        клиенттер DataSetКлиенттерTableAdapter;
}

```

Жеке атрибуттар ретінде сілтемелер жасау.

```

private System.Windows.Data.CollectionViewSource клиенттер ViewSource;

private void Window_Loaded(object sender, RoutedEventArgs)
{
    клиенттерDataSet =
        ((KundenDataSet) (this.FindResource(„kundenDataSet“)));

    клиенттерDataSet.КлиенттерTableAdapter = new
        КлиентDataSetTableAdapters.Клиенттер TableAdapter();

    КлиентDataSet.КлиенттерTableAdapter.Fill(клиентDataSet.Клиент);

    клиентViewSource =
        ((System.Windows.Data.CollectionViewSource)
            (this.FindResource(„kundenViewSource“)));

    клиентViewSource.View.MoveCurrentToFirst();
}
:
:
}

```

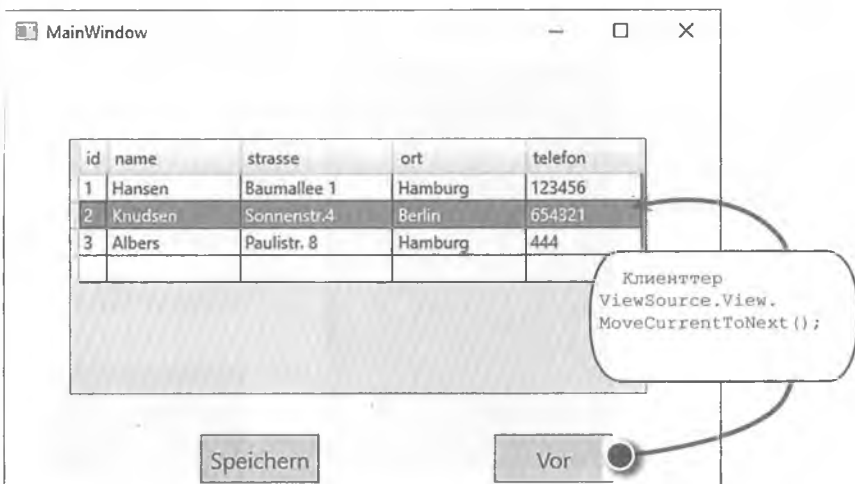
Навигация әдістері

Өзгерістерді бекіту үшін жаңарту әдісіне қосымша, жазбаларда қозғалыстар жасау үшін Viewsource класының Move әдістерін пайдалануға болады:

```

kundenViewSource.View.MoveCurrentToNext();
kundenViewSource.View.MoveCurrentToPrevious();
kundenViewSource.View.MoveCurrentToFirst();
kundenViewSource.View.MoveCurrentToLast();

```



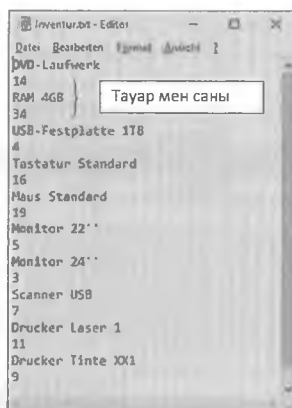
10.3 10-ТАРАУҒА ТАПСЫРМАЛАР

1-тапсырма

Компаниялардың бірінде түгендеу үшін штрих-коды бар барлық тауарларды сканерлейтін мобильді сканер пайдаланылады. Сонымен қатар, штрих-код парағы арқылы сканерленуі мүмкін. Түгендеуден кейін барлық деректер сканер жадының микросхемасында мәтіндік файл түрінде сақталады. Осы мәтіндік файлды оқитын және оны деректер қорының кестесінде сақтайтын қарапайым консоль бағдарламасын жазыңыз. Деректер қорының кестесі бұрын SQL тиісті командаларымен сәйкес келетін деректер қорында (мысалы, ACCESS) жасалған. Кейіннен кестеден келесі статистикалық деректер есептеледі:

- ▶ ең көп мөлшерде қолжетімді үш тауар.
- ▶ ең аз мөлшерде қолжетімді үш тауар.
- ▶ тауарлардың орташа саны.

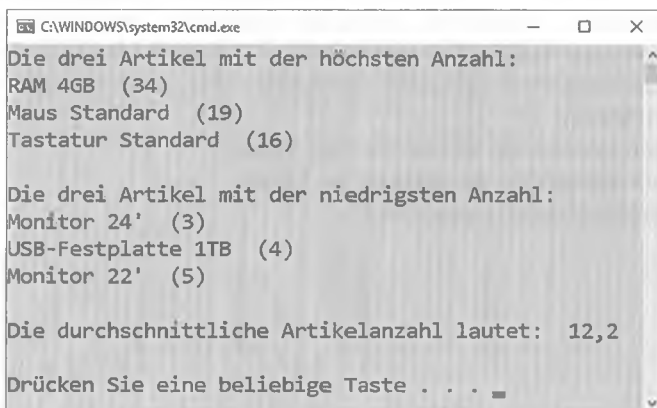
Инвентаризациядан кейін мәтіндік файл былай көрінуі мүмкін:



Ескертулер:

- ▶ жазбаларды енгізу үшін SQL операторын немесе dataset объектісі.
- ▶ деректер есебі SQL (мысалы, AVG) немесе тікелей C#бағдарламасында тиісті функциялардың көмегімен орындалады.

Іске қосылғаннан кейін экранға шығару келесідей болуы мүмкін:



2-тапсырма Бастапқы шарттар:

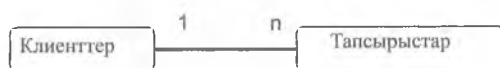
Компанияның бірінде клиенттің тапсырыстары деректер қорының екі кестесінде сақталады (мысалы, Access). Қызметкерлер үшін графикалық интерфейсі бар қарапайым қосымшаны әзірлеу қажет, оның көмегімен әрбір клиенттің тапсырысының деректерін нақты көрсетуге болады.

Базалық кестелер келесідей:

Клиенттер кестесі:

Клиенттер	
ID	Тегі
1	Майер
2	Кнудсен
3	Кайзер
4	Францен
5	Кнеблех

Кестелердің байланысы:



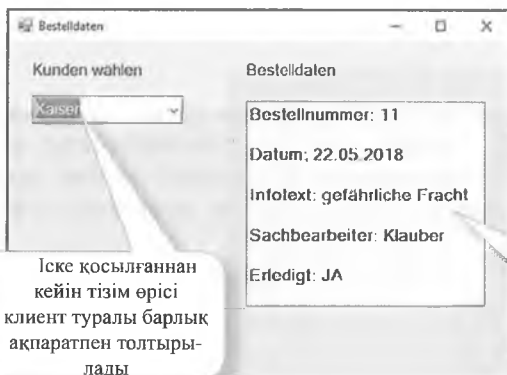
Тапсырыс кестесі:

Bestellungen					
KundenID	Bestellnummer	Datum	Infotext	Sachbearbeiter	erledigt
1	10	10.05.2018	Eilbestellung	Kracher	Ja
3	11	22.05.2018	gefährliche Fracht	Klauber	Ja
4	12	20.05.2018	guter Kunde	Hütter	Nein

Тапсырыстардың кестесі Клиент_ID сыртқы идентификаторы бар, ол екі кесте үшін 1:n байланысын жүзеге асырады.

Тапсырманы қою:

Тиісті деректер қорында (мысалы, ACCESS) екі кестені жасап, кестені тиісті мәндермен толтырыңыз. Содан кейін Windows Forms немесе WPF қолданбасын енгізіңіз, ол деректер қорына жүгінеді және кестелерді оқиды. Қолданба интерфейсі осылай көрінуі керек:



Іске қосылғаннан кейін тізім өрісі клиент туралы барлық ақпаратпен толтырылады

Клиентті тізімнен таңдағаннан кейін тапсырыс кестесі оқылады және Тапсырыс деректері `TextBox` көрсетіледі (`WPF-TextBlock`).

Алфавиттік-пәндік көрсеткіш

– AVG 158, 181
– MAX 158, 181
– SUM 158, 181
\$_POST[] 140
.NET-Framework 161
1:1-байланысы 19
1:m-байланысы 19

A

ABS (сан) 89
ADO.NET 161
ALTER 80
ASC 88
AVG 86

B

BETWEEN 83

C

CASCADE 45
CDATE 88
CINT(сан) 88
Codd 11
Command 162
CommandBuilder 169
COMMIT 100
COUNT 86
CREATE 80
Crows Feet Notation 47

D

DataAdapter 167
Database 9
DataReader 162
DATE() 87
DB-Designer 40
DCL 79
DDL 79
DELETE 97
DISTINCT 82
DML 79, 96
DQL 79
DROP 80

E

EER-модель 43
Entity Relationship Model 21 ERM 21
ER-модельдеу 23
ExecuteNonQuery 165
ExecuteUpdate 154
EXP(мән) 89
External View 16

F

fileperms (); 138
filetype (); 138
fopen() 136
foreach циклі 133
fwrite() 136

G

GetDateTime әдісі 164
– getDouble 153
– getInt 153
GetInt әдісі 32 164
– getString 153
GetString әдісі 163, 164
GRANT 144
GROUP BY 89

H

HAVING 90

I

IN 83
INSERT INTO 96
ISNULL 89

J

JDBC 150

L

LCASE 88
LEFT 88
LEN 88
LibreOffice Base 11, 103

M

m:n-байланыс 20, 22

MAX 86
Microsoft Access 11
Microsoft VISIO 53
– MIN 158
MIN 86
MOD 89
MS-VISIO 53
mydb 44
MyODBC 147
MySQL 11, 40, 128, 140
– Mysql 157
mysql_close() 145
mysql_connect() 145
mysql_db_query() 145
mysql_fetch_array() 145
mysql_num_fields() 145
mysql_num_rows() 145
mysqladmin 142
mysqldump 142
MySQL-Workbench 42
MySQLклиенттері 141

N

NO ACTION 45
NULL44, 84

O

ODBC 147, 161
ODBC драйвері 57
ODBC интерфейсі 116
OLEDB 161
OleDbCommand 162, 168
OleDbDataAdapter 168
OleDbDataReader 162
ON DELETE 114
ON UPDATE 114
Oracle 11, 161
ORDER BY 84

P

PHP 128, 130
phpMyAdmin 142
POST 140
Primary Key 19

R

require () 137
RESTRICT 45
ResultSet
rewind() 137
RIGHT 88
RND() 89
ROLLBACK 100

S

SELECT 81
SET NULL 45
SQL 79
– SQLite 151
SQL-Script 51
SQL-Server 11, 43, 161
SQL-команда13
STR(сан) 88
strtolower() 135
Subqueries 95
SUM 86

T

TextBox 182

U

UCASE 88
ucfirst() 135
UPDATE 97
V
VAL(жол) 88
Visual C# 2010 162

W

WHERE 81
X
XAMPP 128, 129

A

Агрегаттық функция 86
Агрегаттық функциялар 86
Адаптер-нысандар 167
айнымалы 131
Архитектура (ДҚБЖ) 15
Архитектура 12

Ассоциативті массивтер 132
Атрибуттар 22

Б

Баған 18
Баған18
Бағынышты нысандар 70, 124
Базалық кестелер 16
Байланыс 19, 22
Байланыс қуаты 113
Байланыс қуатын анықтауи 57
Байланыстар 67
Байланыстарды қосу 56
Басқару жүйесі 20
Бастапқы кілт 18, 67
Бастапқы кілтті орнату 111
Бірнеше пайдаланушыға арналған режим 13
Біртұтастылық 36
Бэкэнд 13

В

Веб-сервер 128
Веб-Сервер Apache 128

Д

Денормализация 33
Дерекқор шебері 103
Деректер жиынтығы (DataSet) 167
Деректер каталогы 11
Деректер қоры 9
Деректер қоры желісі 14
Деректер қоры жолы 11
Деректер қоры моделінің диаграммасы 53
деректер қорын басқару 72
Деректер қорын басқару жүйесі 9
Деректер қорын басқару жүйесі 9
Деректер қорын өзірлеу 29
Деректер қорын жобалау кезеңдері 17
Деректер қорын модельдеу 40
Деректер қорына кіру 150, 161
Деректер қорының жүйесі 10

Деректер қорының келісілуі 98
Деректер тәуелсіздігі 10
Деректер түрі (Datatype) 44
Деректер түрі 18
Деректерді жаңарту 97
Деректерді жеткізуші 161
Деректерді манипуляциялау тілі 96
Деректерді сақтау 9
ДҚБЖ 9
ДҚБЖ-ға қосылу 52
ДҚЖ 10

Е

Екінші қалыпты форма форма 31
Енгізу масқалары 12
Есептер 74

Ж

Жазбаларды өшіру 97
Жоғарыдан төмен 30
Журнал 12

И

Иерархиялық деректер қоры 14
Индекс 61

І

Ішкі деңгей 16

К

Кері жобалау (Reverse Engineering) 57
Кесте құру 53
Кесте шебері 108
Кестелерді байланыстыру 45
Кілт 19
Клиент/серверлердің деректер қоры 13
Командалық кнопкалар 72
Концептуалды деңгей 16
Көп өлшемді массивтер 133
Көрініс 12, 62
Көріністер 62

Күн функциясы 87

Қ

Қалып шебері 123

Қалыптар 69, 123, 139

Қалыпты форма 30

Қолданбалы бағдарламалар 9

Қолжетімділік құқықтары 143

Қосылмаған объектілер 167

Қосымша сұраулар 95

Құпиялылық 10

Л

Логикалық функция 89

М

Макростар 73

Мартин нотацясы 47

Массивтер 131

Математикалық функциялар 89

Н

Негізсіз атрибут 32

Нормализация 29

О

Объект 19

Объектілі-бағытталған деректер қоры 14

Объектілі-реляциялық деректер қоры 14

Ө

Өзгерістер ауытқулары 98

Өріс түрі 112

Өрістерді тексеру шарттары 64

Өтпелі тәуелділік 32

П

Пайдаланушы көріністері 11

Пакет msi 41

Провайдер 161

Р

Реляциялық деректер қоры 14, 18

С

Сандық массивтер 131

Сәйкессіздіктер 10

Символдық жолдар 135

Сілтеме әрекеті 57

Сілтемелік тұтастық 36, 67

Сүзу шарттары 121

Сұлба 15

Сұраныстар 76

Сұраныстар жасау 119

Сұрыптау 77

Сыртқы кілт 45

Т

Таңдау сұраныстары 81

Текстік блок WPF182

Толық функционалдық тәуелділік 31

Төменнен жоғары әдісі 30

Транзакциялар 99

Туынды кесте 20

Түрлендіру функциясы 88

Ұ

Ұяшық 18

Ф

Файлдық ұйымдар 12

Ш

Шарттар 81

Барлық құқығы қорғалған. Кітапты немесе оның кез келген бөлігін электрондық не механикалық түрде, фотокөшірме түрінде, компьютерлік жадыда жазба түрінде көшіруге, сондай-ақ кез келген ақпараттық жүйеде баспа рұқсатынсыз пайдалануға тыйым салынады. Кітапты немесе оның бөлімдерін баспа рұқсатынсыз көшіру, көбейту және басқа да түрде пайдалану заңсыз болып табылады және өкімшілік, азаматтық жауапкершілікке тартылады.

«Кәсіптік білім» сериясы

Деректер қоры

1305000 – Ақпараттық жүйелер
1304000 – Есептеу техникасы және бағдарламамен
қамтамасыз ету

Оқулық