

Максим Кузнецов  
Игорь Симдянов



# MySQL 5

- Полный список функций MySQL 5
- Язык запросов SQL
- Хранимые процедуры
- Локализация
- Представления
- Тонкости администрирования
- Взаимодействие с C++, PHP и Perl

**Наиболее  
полное  
руководство**

+  cd

**В ПОДЛИННИКЕ** <sup>®</sup>

**Максим Кузнецов  
Игорь Симдянов**

# **MySQL 5**

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.06  
ББК 32.973.26-018.1  
К89

**Кузнецов, М. В.**

К89 MySQL 5 / М. В. Кузнецов, И. В. Симдянов. — СПб.: БХВ-Петербург, 2006. — 1024 с.: ил.

ISBN 5-94157-928-4

Описывается новая версия популярной СУБД MySQL, в том числе вложенные запросы, хранимые процедуры, представления, триггеры, курсоры, информационная схема, пространственные расширения, репликация и другие элементы. Большое внимание уделяется администрированию и настройке MySQL. Приводится большое число примеров самых разнообразных SQL-запросов и программ на языках C/C++, Perl и PHP. На компакт-диске содержатся исходные коды авторских программ, а также дистрибутивы и исходные коды MySQL версий 5.0 и 5.1 для Windows и Linux, графические клиенты для MySQL.

*Для программистов и разработчиков баз данных*

УДК 681.3.06  
ББК 32.973.26-018.1

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 15.08.06.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 82,56.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953 Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ОАО "Техническая книга"

190005, Санкт-Петербург, Измайловский пр., 29.

# Оглавление

<b>Введение</b> .....	<b>1</b>
Для кого и о чем эта книга.....	1
Благодарности.....	2
<b>ЧАСТЬ I. ЗНАКОМСТВО С SQL</b> .....	<b>3</b>
<b>Глава 1. История развития баз данных. Понятие реляционной базы данных</b> .....	<b>5</b>
1.1. История развития СУБД. Реляционные базы данных.....	6
1.1.1. Иерархические базы данных.....	6
1.1.2. Сетевые базы данных.....	8
1.2. Особенности реляционных баз данных.....	9
1.2.1. Первичные ключи.....	11
1.2.2. Нормализация базы данных.....	13
1.3. СУБД и сети.....	14
1.3.1. Централизованная архитектура.....	15
1.3.2. Архитектура "клиент-сервер".....	16
1.3.3. Трехуровневая архитектура Интернета.....	17
1.3.4. Кластерная модель.....	19
1.4. Как работают базы данных и что такое SQL?.....	20
1.5. Версии MySQL.....	21
1.5.1. Что нового в MySQL 4.1?.....	23
1.5.2. Что нового в MySQL 5.0?.....	23
<b>Глава 2. Установка MySQL 5</b> .....	<b>24</b>
2.1. Получение дистрибутива.....	24
2.2. Установка на платформу Windows.....	25
2.3. Установка на платформу Linux.....	39
<b>Глава 3. Работа с утилитами MySQL</b> .....	<b>43</b>
3.1. Утилита <i>mysql</i> .....	44
3.2. Утилита <i>mysqldump</i> .....	63

<b>Глава 4. Создание баз данных и таблиц. Типы данных.....</b>	<b>69</b>
4.1. Создание базы данных.....	69
4.2. Создание таблицы.....	73
4.3. Типы данных.....	76
4.3.1. Числовые данные.....	77
4.3.2. Строковые данные.....	80
4.3.3. Календарные данные.....	83
4.3.4. Тип данных <i>NULL</i> .....	86
4.3.5. Выбор типа данных.....	88
4.4. Учебная база данных.....	89
<b>Глава 5. Индексы .....</b>	<b>93</b>
5.1. Первичный ключ.....	94
5.2. Обычный и уникальный индексы.....	98
<b>Глава 6. Добавление данных.....</b>	<b>103</b>
6.1. Однострочный оператор <i>INSERT</i> .....	103
6.2. Многострочный оператор <i>INSERT</i> .....	110
6.3. Отложенная вставка записей.....	111
6.4. Пакетная загрузка данных.....	111
6.5. Утилита <i>mysqlimport</i> .....	115
<b>Глава 7. Выборка данных.....</b>	<b>118</b>
7.1. Изменение количества и порядка следования столбцов.....	118
7.2. Условия.....	121
7.3. Сортировка.....	124
7.4. Ограничение выборки.....	128
7.5. Использование функций.....	129
7.6. Группировка записей.....	132
7.7. Объединение таблиц.....	138
7.8. Сохранение результатов во внешний файл.....	143
<b>Глава 8. Многотабличные запросы .....</b>	<b>147</b>
8.1. Перекрестное объединение таблиц.....	147
8.2. Объединение таблиц при помощи <i>JOIN</i> .....	159
<b>Глава 9. Удаление данных.....</b>	<b>167</b>
9.1. Оператор <i>DELETE</i> .....	167
9.2. Оператор <i>TRUNCATE TABLE</i> .....	168
9.3. Удаление из нескольких таблиц.....	169
<b>Глава 10. Обновление данных .....</b>	<b>173</b>
10.1. Оператор <i>UPDATE</i> .....	173
10.2. Многотабличный оператор <i>UPDATE</i> .....	178
10.3. Оператор <i>REPLACE</i> .....	181

<b>ЧАСТЬ II. СЛОЖНЫЕ ВОПРОСЫ MYSQL.....</b>	<b>183</b>
<b>Глава 11. Типы и структура таблиц.....</b>	<b>185</b>
11.1. MyISAM .....	185
11.2. MERGE.....	186
11.3. MEMORY (HEAP).....	188
11.4. EXAMPLE.....	189
11.5. BDB (BerkeleyDB).....	190
11.6. InnoDB.....	191
11.7. NDB Cluster.....	192
11.8. ARCHIVE.....	192
11.9. CSV.....	193
11.10. FEDERATED.....	194
11.11. BLACKHOLE.....	196
<b>Глава 12. Создание таблиц и удаление таблиц.....</b>	<b>198</b>
12.1. Оператор <i>CREATE TABLE</i> .....	198
12.1.1. Структура таблицы.....	199
12.1.2. Параметры таблицы .....	206
12.2. Оператор <i>DROP TABLE</i> .....	211
<b>Глава 13. Редактирование структуры таблиц.....</b>	<b>213</b>
13.1. Добавление и удаление столбцов.....	213
13.2. Изменение уже существующих столбцов.....	216
13.3. Добавление и удаление индексов.....	218
13.4. Преобразование параметров таблицы .....	220
<b>Глава 14. Приведение типов .....</b>	<b>225</b>
14.1. Ключевое слово <i>BINARY</i> .....	225
14.2. Функция <i>CAST()</i> .....	226
14.3. Функция <i>CONVERT()</i> .....	228
14.4. Поддержка кодировок.....	229
14.5. Преобразование кодировок .....	237
14.6. Применение ключевого слова <i>COLLATE</i> .....	238
<b>Глава 15. Операторы и математические функции .....</b>	<b>240</b>
15.1. Операторы.....	240
15.1.1. Арифметические операторы.....	240
15.1.2. Операторы сравнения.....	245
15.1.3. Логические операторы .....	257
15.1.4. Битовые операторы .....	259
15.1.5. Приоритет операторов .....	265
15.2. Математические функции.....	266
15.2.1. Функция <i>ABS()</i> .....	266
15.2.2. Функция <i>ACOS()</i> .....	267

15.2.3. Функция <i>ASIN()</i> .....	267
15.2.4. Функция <i>ATAN()</i> .....	267
15.2.5. Функция <i>ATAN2()</i> .....	268
15.2.6. Функция <i>CEILING()</i> .....	268
15.2.7. Функция <i>COS()</i> .....	269
15.2.8. Функция <i>COT()</i> .....	270
15.2.9. Функция <i>CRC32()</i> .....	270
15.2.10. Функция <i>DEGREES()</i> .....	270
15.2.11. Функция <i>EXP()</i> .....	271
15.2.12. Функция <i>FLOOR()</i> .....	271
15.2.13. Функция <i>LOG()</i> .....	272
15.2.14. Функция <i>LOG2()</i> .....	272
15.2.15. Функция <i>LOG10()</i> .....	273
15.2.16. Функция <i>MOD()</i> .....	273
15.2.17. Функция <i>PI()</i> .....	273
15.2.18. Функция <i>POW()</i> .....	273
15.2.19. Функция <i>RADIANS()</i> .....	274
15.2.20. Функция <i>RAND()</i> .....	274
15.2.21. Функция <i>ROUND()</i> .....	275
15.2.22. Функция <i>SIGN()</i> .....	276
15.2.23. Функция <i>SIN()</i> .....	277
15.2.24. Функция <i>SQPT()</i> .....	277
15.2.25. Функция <i>TAN()</i> .....	277
15.2.26. Функция <i>TRUNCATE()</i> .....	278
<b>Глава 16. Функции даты и времени .....</b>	<b>279</b>
16.1. Функция <i>ADDDATE()</i> .....	279
16.2. Функция <i>ADDTIME()</i> .....	285
16.3. Функция <i>CONVERT_TZ()</i> .....	286
16.4. Функция <i>CURDATE()</i> .....	286
16.5. Функция <i>CURTIME()</i> .....	287
16.6. Функция <i>DATE()</i> .....	287
16.7. Функция <i>DATEDIFF()</i> .....	288
16.8. Функция <i>DATE_FORMAT()</i> .....	288
16.9. Функция <i>DAY()</i> .....	291
16.10. Функция <i>DAYNAME()</i> .....	291
16.11. Функция <i>DAYOFMONTH()</i> .....	292
16.12. Функция <i>DAYOFWEEK()</i> .....	293
16.13. Функция <i>DAYOFYEAR()</i> .....	293
16.14. Функция <i>EXTRACT()</i> .....	293
16.15. Функция <i>FROM_DAYS()</i> .....	295
16.16. Функция <i>FROM_UNIXTIME()</i> .....	295
16.17. Функция <i>GET_FORMAT()</i> .....	297
16.18. Функция <i> HOUR()</i> .....	298
16.19. Функция <i>LAST_DAY()</i> .....	299

16.20. Функция <i>MAKEDATE()</i> .....	300
16.21. Функция <i>MAKETIME()</i> .....	301
16.22. Функция <i>MICROSECOND()</i> .....	301
16.23. Функция <i>MINUTE()</i> .....	302
16.24. Функция <i>MONTH()</i> .....	302
16.25. Функция <i>MONTHNAME()</i> .....	302
16.26. Функция <i>NOW()</i> .....	303
16.27. Функция <i>PERIOD_ADD()</i> .....	304
16.28. Функция <i>PERIOD_DIFF()</i> .....	304
16.29. Функция <i>QUARTER()</i> .....	305
16.30. Функция <i>SECOND()</i> .....	305
16.31. Функция <i>SEC_TO_TIME()</i> .....	305
16.32. Функция <i>STR_TO_DATE()</i> .....	306
16.33. Функция <i>SUBDATE()</i> .....	307
16.34. Функция <i>SUBTIME()</i> .....	307
16.35. Функция <i>TIME()</i> .....	308
16.36. Функция <i>TIMEDIFF()</i> .....	309
16.37. Функция <i>TIMESTAMP()</i> .....	309
16.38. Функция <i>TIMESTAMPADD()</i> .....	310
16.39. Функция <i>TIMESTAMPDIFF()</i> .....	311
16.40. Функция <i>TIME_FORMAT()</i> .....	312
16.41. Функция <i>TIME_TO_SEC()</i> .....	312
16.42. Функция <i>TO_DAYS()</i> .....	313
16.43. Функция <i>UNIX_TIMESTAMP()</i> .....	314
16.44. Функция <i>UTC_DATE()</i> .....	314
16.45. Функция <i>UTC_TIME()</i> .....	315
16.46. Функция <i>UTC_TIMESTAMP()</i> .....	315
16.47. Функция <i>WEEK()</i> .....	316
16.48. Функция <i>WEEKDAY()</i> .....	317
16.49. Функция <i>WEEKOFYEAR()</i> .....	317
16.50. Функция <i>YEAR()</i> .....	318
16.51. Функция <i>YEARWEEK()</i> .....	318
<b>Глава 17. Строковые функции .....</b>	<b>320</b>
17.1. Функция <i>ASCII()</i> .....	320
17.2. Функция <i>BIN()</i> .....	321
17.3. Функция <i>BIT_LENGTH()</i> .....	321
17.4. Функция <i>CHAR()</i> .....	321
17.5. Функция <i>CHAR_LENGTH()</i> .....	323
17.6. Функция <i>CHARSET()</i> .....	324
17.7. Функция <i>COLLATION()</i> .....	325
17.8. Функция <i>COMPRESS()</i> .....	325
17.9. Функция <i>CONCAT()</i> .....	327
17.10. Функция <i>CONCAT_WS()</i> .....	328
17.11. Функция <i>CONV()</i> .....	329

17.12. Функция <i>ELT()</i> .....	329
17.13. Функция <i>EXPORT_SET()</i> .....	330
17.14. Функция <i>FIELD()</i> .....	331
17.15. Функция <i>FIND_IN_SET()</i> .....	331
17.16. Функция <i>FORMAT()</i> .....	332
17.17. Функция <i>HEX()</i> .....	333
17.18. Функция <i>INSERT()</i> .....	333
17.19. Функция <i>INSTR()</i> .....	334
17.20. Функция <i>LEFT()</i> .....	335
17.21. Функция <i>LENGTH()</i> .....	335
17.22. Функция <i>LOAD_FILE()</i> .....	335
17.23. Функция <i>LOCATE()</i> .....	336
17.24. Функция <i>LOWER()</i> .....	337
17.25. Функция <i>LPAD()</i> .....	337
17.26. Функция <i>LTRIM()</i> .....	338
17.27. Функция <i>MAKE_SET()</i> .....	338
17.28. Функция <i>MID()</i> .....	338
17.29. Функция <i>OCT()</i> .....	340
17.30. Функция <i>ORD()</i> .....	340
17.31. Функция <i>POSITION()</i> .....	341
17.32. Функция <i>QUOTE()</i> .....	341
17.33. Функция <i>REPEAT()</i> .....	341
17.34. Функция <i>REPLACE()</i> .....	342
17.35. Функция <i>REVERSE()</i> .....	344
17.36. Функция <i>RIGHT()</i> .....	344
17.37. Функция <i>RPAD()</i> .....	345
17.38. Функция <i>RTRIM()</i> .....	345
17.39. Функция <i>SOUNDEX()</i> .....	346
17.40. Функция <i>SPACE()</i> .....	346
17.41. Функция <i>SUBSTRING()</i> .....	346
17.42. Функция <i>SUBSTRING_INDEX()</i> .....	348
17.43. Функция <i>TRIM()</i> .....	348
17.44. Функция <i>UNCOMPRESS()</i> .....	349
17.45. Функция <i>UNCOMPRESSED_LENGTH()</i> .....	350
17.46. Функция <i>UNHEX()</i> .....	351
17.47. Функция <i>UPPER()</i> .....	351
<b>Глава 18. Безопасность и MySQL</b> .....	<b>352</b>
18.1. Функции <i>AES_ENCRYPT()</i> и <i>AES_DECRYPT()</i> .....	352
18.2. Функции <i>ENCODE()</i> и <i>DECODE()</i> .....	355
18.3. Функции <i>DES_ENCRYPT()</i> и <i>DES_DECRYPT()</i> .....	355
18.4. Функция <i>ENCRYPT()</i> .....	356
18.5. Функция <i>MD5()</i> .....	357
18.6. Функция <i>PASSWORD()</i> .....	358
18.7. Функция <i>SHA1()</i> .....	359

<b>Глава 19. Поиск и регулярные выражения.....</b>	<b>361</b>
19.1. Оператор <i>LIKE</i> .....	361
19.2. Оператор <i>NOT LIKE</i> .....	366
19.3. Оператор <i>SOUND LIKE</i> .....	367
19.4. Оператор <i>RLIKE (REGEXP)</i> .....	367
19.5. Оператор <i>NOT RLIKE</i> .....	379
19.6. Функция <i>STRCMP()</i> .....	380
<b>Глава 20. Полнотекстовый поиск.....</b>	<b>381</b>
20.1. Индекс <i>FULLTEXT</i> .....	381
20.2. Конструкция <i>MATCH (...) AGAINST (...)</i> .....	383
20.3. Логический режим.....	388
20.4. Режим расширения запроса.....	395
<b>Глава 21. Функции, применяемые вместе с конструкцией <i>GROUP BY</i>.....</b>	<b>397</b>
21.1. Функция <i>AVG()</i> .....	397
21.2. Функция <i>BIT_AND()</i> .....	400
21.3. Функция <i>BIT_OR()</i> .....	401
21.4. Функция <i>BIT_XOR()</i> .....	402
21.5. Функция <i>COUNT()</i> .....	403
21.6. Функция <i>GROUP_CONCAT()</i> .....	406
21.7. Функция <i>MIN()</i> .....	408
21.8. Функция <i>MAX()</i> .....	409
21.9. Функция <i>STD()</i> .....	410
21.10. Функция <i>STDDEV_SAMP()</i> .....	411
21.11. Функция <i>SUM()</i> .....	412
21.12. Функция <i>VAR_POP()</i> .....	412
21.13. Функция <i>VAR_SAMP()</i> .....	413
21.14. Конструкция <i>WITH ROLLUP()</i> .....	413
<b>Глава 22. Разные функции.....</b>	<b>417</b>
22.1. Функции управления потоком выполнения.....	417
22.1.1. Функция <i>CASE()</i> .....	417
22.1.2. Функция <i>IF()</i> .....	418
22.1.3. Функция <i>IFNULL()</i> .....	420
22.1.4. Функция <i>NULLIF()</i> .....	421
22.2. Информационные функции.....	421
22.2.1. Функция <i>BENCHMARK()</i> .....	422
22.2.2. Функция <i>CONNECTION_ID()</i> .....	422
22.2.3. Функция <i>CURRENT_USER()</i> .....	423
22.2.4. Функция <i>DATABASE()</i> .....	423
22.2.5. Функция <i>FOUND_ROWS()</i> .....	424
22.2.6. Функция <i>LAST_INSERT_ID()</i> .....	425
22.2.7. Функция <i>ROW_COUNT()</i> .....	427

22.2.8. Функция <i>USER()</i> .....	429
22.2.9. Функция <i>VERSION()</i> .....	429
22.3. Разные функции.....	430
22.3.1. Функция <i>DEFAULT()</i> .....	430
22.3.2. Функция <i>GET_LOCK()</i> .....	432
22.3.3. Функция <i>INET_ATON()</i> .....	433
22.3.4. Функция <i>INET_NTOA()</i> .....	433
22.3.5. Функция <i>IS_FREE_LOCK()</i> .....	434
22.3.6. Функция <i>IS_USED_LOCK()</i> .....	435
22.3.7. Функция <i>NAME_CONST()</i> .....	435
22.3.8. Функция <i>RELEASE_LOCK()</i> .....	436
22.3.9. Функция <i>SLEEP()</i> .....	436
22.3.10. Функция <i>UUID()</i> .....	436
<b>Глава 23. Переменные и временные таблицы.....</b>	<b>438</b>
23.1. Переменные SQL.....	438
23.2. Временные таблицы .....	442
<b>Глава 24. Вложенные запросы .....</b>	<b>446</b>
24.1. Вложенный запрос как скалярный операнд.....	449
24.2. Вложенные запросы, возвращающие несколько строк.....	452
24.2.1. Ключевое слово <i>IN</i> .....	453
24.2.2. Ключевое слово <i>ANY (SOME)</i> .....	454
24.2.3. Ключевое слово <i>ALL</i> .....	456
24.3. Проверка на существование .....	458
24.4. Коррелированные запросы .....	461
24.5. Вложенные запросы, возвращающие несколько столбцов.....	461
24.6. Подзапросы в конструкции <i>FROM</i> .....	463
24.7. Вложенные запросы в операторе <i>CREATE TABLE</i> .....	466
24.8. Вложенные запросы в операторе <i>INSERT</i> .....	469
<b>Глава 25. Внешние ключи и ссылочная целостность .....</b>	<b>473</b>
<b>Глава 26. Транзакции и блокировки .....</b>	<b>481</b>
26.1. Транзакции.....	481
26.2. Блокировка таблиц .....	487
<b>Глава 27. Управление учетными записями пользователей .....</b>	<b>490</b>
27.1. Учетные записи СУБД MySQL .....	490
27.2. Оператор <i>CREATE USER</i> .....	493
27.3. Оператор <i>DROP USER</i> .....	495
27.4. Оператор <i>RENAME USER</i> .....	497
27.5. Оператор <i>GRANT</i> .....	498
27.6. Оператор <i>REVOKE</i> .....	508

<b>ЧАСТЬ III. СРЕДСТВА АДМИНИСТРИРОВАНИЯ СУБД MYSQL</b> .....	<b>509</b>
<b>Глава 28. Администрирование СУБД MySQL</b> .....	<b>511</b>
28.1. Параметры запуска сервера MySQL .....	511
28.2. Системные переменные сервера .....	519
28.3. Режим SQL-сервера.....	540
28.4. Журнальные файлы .....	544
28.4.1. Журнальные таблицы .....	544
28.4.2. Журнал ошибок .....	545
28.4.3. Общий журнал запросов .....	546
28.4.4. Бинарный журнал регистраций .....	546
28.4.5. Утилита <i>mysqlbinlog</i> .....	549
28.4.6. Журнал медленных запросов.....	551
28.5. Оператор <i>CACHE INDEX</i> .....	552
28.6. Оператор <i>FLUSH</i> .....	553
28.7. Оператор <i>KILL</i> .....	554
28.8. Оператор <i>LOAD INDEX INTO CACHE</i> .....	555
28.9. Оператор <i>RESET</i> .....	555
28.10. Утилита <i>mysqladmin</i> .....	556
<b>Глава 29. Оператор SET</b> .....	<b>560</b>
<b>Глава 30. Оператор SHOW</b> .....	<b>573</b>
30.1. Оператор <i>SHOW CHARACTER SET</i> .....	574
30.2. Оператор <i>SHOW COLLATION</i> .....	575
30.3. Оператор <i>SHOW COLUMNS</i> .....	576
30.4. Оператор <i>SHOW CREATE DATABASE</i> .....	579
30.5. Оператор <i>SHOW CREATE TABLE</i> .....	579
30.6. Оператор <i>SHOW DATABASES</i> .....	580
30.7. Оператор <i>SHOW ENGINES</i> .....	581
30.8. Оператор <i>SHOW ENGINE</i> .....	582
30.9. Оператор <i>SHOW ERRORS</i> .....	583
30.10. Оператор <i>SHOW GRANTS</i> .....	585
30.11. Оператор <i>SHOW INDEX</i> .....	586
30.12. Оператор <i>SHOW PLUGIN</i> .....	589
30.13. Оператор <i>SHOW PRIVILEGES</i> .....	589
30.14. Оператор <i>SHOW PROCESSLIST</i> .....	593
30.15. Оператор <i>SHOW STATUS</i> .....	596
30.16. Оператор <i>SHOW TABLE STATUS</i> .....	601
30.17. Оператор <i>SHOW TABLES</i> .....	604
30.18. Оператор <i>SHOW VARIABLES</i> .....	606
30.19. Оператор <i>SHOW WARNINGS</i> .....	607
30.20. Утилита <i>mysqlshow</i> .....	610

<b>Глава 31. Предотвращение катастроф и восстановление .....</b>	<b>613</b>
31.1. Оператор <i>CHECK TABLE</i> .....	613
31.2. Оператор <i>ANALYZE TABLE</i> .....	616
31.3. Оператор <i>CHECKSUM TABLE</i> .....	617
31.4. Оператор <i>OPTIMIZE TABLE</i> .....	619
31.5. Оператор <i>REPAIR TABLE</i> .....	620
31.6. Оператор <i>BACKUP TABLE</i> .....	622
31.7. Оператор <i>RESTORE TABLE</i> .....	623
31.8. Резервное копирование .....	625
31.9. Утилита <i>mysqlcheck</i> .....	627
<b>Глава 32. Репликация в MySQL.....</b>	<b>630</b>
32.1. Введение в репликацию .....	630
32.2. Детали реализации процесса репликации .....	631
32.2.1. Состояние потока репликации главного сервера.....	634
32.2.2. Состояние потока ввода/вывода подчиненного сервера.....	634
32.2.3. Состояние потока обработки ретрансляционных журналов подчиненного сервера.....	636
32.2.4. Журнал ретрансляции и файлы состояния .....	636
32.3. Настройка репликации.....	638
32.4. Совместимость репликации между версиями MySQL.....	641
32.5. Параметры запуска репликации.....	641
32.6. Операторы управления главным сервером .....	646
32.6.1. Оператор <i>PURGE MASTER LOGS</i> .....	647
32.6.2. Оператор <i>RESET MASTER</i> .....	647
32.6.3. Оператор <i>SET SQL_BIN_LOG</i> .....	647
32.6.4. Оператор <i>SHOW BINLOG EVENTS</i> .....	648
32.6.5. Оператор <i>SHOW MASTER LOGS</i> .....	648
32.6.6. Оператор <i>SHOW MASTER STATUS</i> .....	648
32.6.7. Оператор <i>SHOW SLAVE HOSTS</i> .....	649
32.7. Операторы управления подчиненными серверами .....	649
32.7.1. Оператор <i>CHANGE MASTER TO</i> .....	649
32.7.2. Оператор <i>LOAD DATA FROM MASTER</i> .....	651
32.7.3. Оператор <i>LOAD TABLE FROM MASTER</i> .....	652
32.7.4. Функция <i>MASTER_POS_WAIT()</i> .....	652
32.7.5. Оператор <i>RESET SLAVE</i> .....	653
32.7.6. Оператор <i>SET GLOBAL SQL_SLAVE_SKIP_COUNTER</i> .....	653
32.7.7. Оператор <i>SHOW SLAVE STATUS</i> .....	653
32.7.8. Оператор <i>START SLAVE</i> .....	656
32.7.9. Оператор <i>STOP SLAVE</i> .....	657

<b>ЧАСТЬ IV. НОВОВВЕДЕНИЯ MYSQL 5.0.....</b>	<b>659</b>
<b>Глава 33. Хранимые процедуры.....</b>	<b>661</b>
33.1. Хранимые процедуры и привилегии.....	662
33.2. Создание хранимой процедуры.....	662
33.2.1. Тело процедуры.....	664
33.2.2. Параметры процедуры.....	668
33.2.3. Работа с таблицами базы данных.....	671
33.2.4. Хранимые функции.....	679
33.3. Группа характеристик хранимых процедур.....	681
33.4. Операторы управления потоком данных.....	684
33.4.1. Оператор <i>IF...THEN...ELSE</i> .....	684
33.4.2. Оператор <i>CASE</i> .....	688
33.4.3. Оператор <i>WHILE</i> .....	691
33.4.4. Оператор <i>REPEAT</i> .....	696
33.4.5. Оператор <i>LOOP</i> .....	697
33.4.6. Оператор <i>GOTO</i> .....	698
33.5. Метаданные.....	700
33.5.1. Оператор <i>SHOW PROCEDURE STATUS</i> .....	700
33.5.2. Оператор <i>SHOW CREATE</i> .....	702
33.5.3. Извлечение информации из таблицы <i>mysql.proc</i> .....	703
33.6. Удаление хранимых процедур.....	706
33.7. Редактирование хранимых процедур.....	707
33.8. Обработчики ошибок.....	708
33.9. Курсоры.....	714
<b>Глава 34. Триггеры.....</b>	<b>718</b>
34.1. Оператор <i>CREATE TRIGGER</i> .....	718
34.2. Оператор <i>DROP TRIGGER</i> .....	722
<b>Глава 35. Представления.....</b>	<b>723</b>
35.1. Создание представлений.....	724
35.2. Удаление представлений.....	738
35.3. Редактирование представлений.....	739
35.4. Оператор <i>SHOW CREATE VIEW</i> .....	739
<b>Глава 36. Информационная схема.....</b>	<b>741</b>
36.1. Представление <i>CHARACTER_SETS</i> .....	742
36.2. Представление <i>COLLATIONS</i> .....	745
36.3. Представление <i>COLLATION_CHARACTER_SET_APPLICABILITY</i> .....	747
36.4. Представление <i>COLUMN_PRIVILEGES</i> .....	748
36.5. Представление <i>COLUMNS</i> .....	751
36.6. Представление <i>KEY_COLUMN_USAGE</i> .....	755
36.7. Представление <i>ROUTINES</i> .....	757

36.8. Представление <i>SCHEMA_PRIVILEGES</i> .....	761
36.9. Представление <i>SCHEMATA</i> .....	763
36.10. Представление <i>STATISTICS</i> .....	765
36.11. Представление <i>TABLE_CONSTRAINTS</i> .....	769
36.12. Представление <i>TABLE_PRIVILEGES</i> .....	770
36.13. Представление <i>TABLES</i> .....	772
36.14. Представление <i>USER_PRIVILEGES</i> .....	775
36.15. Представление <i>VIEWS</i> .....	776

## **ЧАСТЬ V. ВЗАИМОДЕЙСТВИЕ MYSQL С ЯЗЫКАМИ ПРОГРАММИРОВАНИЯ ..... 779**

### **Глава 37. Взаимодействие MySQL и C/C++ ..... 781**

37.1. Взаимодействие с MySQL в Linux .....	781
37.1.1. Типы данных.....	786
37.1.2. Функции интерфейса C .....	791
37.2. Взаимодействие с MySQL в Windows .....	815

### **Глава 38. Взаимодействие MySQL и Perl ..... 838**

### **Глава 39. Взаимодействие MySQL и PHP..... 844**

39.1. Функция <i>mysql_connect()</i> .....	844
39.2. Функция <i>mysql_close()</i> .....	846
39.3. Функция <i>mysql_select_db()</i> .....	847
39.4. Функция <i>mysql_query()</i> .....	850
39.5. Функция <i>mysql_result()</i> .....	851
39.6. Функция <i>mysql_fetch_row()</i> .....	852
39.7. Функция <i>mysql_fetch_assoc()</i> .....	853
39.8. Функция <i>mysql_fetch_array()</i> .....	855
39.9. Функция <i>mysql_fetch_object()</i> .....	858
39.10. Функция <i>mysql_num_rows()</i> .....	859
39.11. Система регистрации .....	862
39.12. Система авторизации .....	871
39.13. Базовая HTTP-авторизация.....	873
39.14. Пользователи online .....	876
39.15. Постраничная навигация.....	879
39.16. Алфавитная навигация .....	883
39.17. Сортировка.....	885
39.18. Двойной выпадающий список.....	888
39.19. Удаление сразу нескольких позиций .....	894
39.20. Хранение MP3-файлов в базе данных .....	897
39.21. Хранение изображений в базе данных.....	901

### **Заключение..... 907**

---

**ПРИЛОЖЕНИЯ .....909**
**Приложение 1. Пространственные расширения MySQL .....911**

П1.1. Геометрическая модель OpenGIS.....	912
П1.1.1. Класс <i>Geometry</i> .....	913
П1.1.2. Класс <i>Point</i> .....	914
П1.1.3. Класс <i>Curve</i> .....	915
П1.1.4. Класс <i>LineString</i> .....	915
П1.1.5. Класс <i>Surface</i> .....	916
П1.1.6. Класс <i>Polygon</i> .....	916
П1.1.7. Класс <i>GeometryCollection</i> .....	917
П1.1.8. Класс <i>MultiPoint</i> .....	917
П1.1.9. Класс <i>MultiCurve</i> .....	917
П1.1.10. Класс <i>MultiString</i> .....	918
П1.1.11. Класс <i>MultiSurface</i> .....	918
П1.1.12. Класс <i>MultiPolygon</i> .....	918
П1.2. Форматы пространственных данных .....	918
П1.2.1. Объект <i>Point</i> .....	919
П1.2.2. Объект <i>LineString</i> .....	921
П1.2.3. Объект <i>Polygon</i> .....	921
П1.2.4. Объект <i>MultiPoint</i> .....	922
П1.2.5. Объект <i>MultiLineString</i> .....	923
П1.2.6. Объект <i>MultiPolygon</i> .....	924
П1.2.7. Объект <i>GeometryCollection</i> .....	925
П1.3. Работа с геометрическими элементами в MySQL .....	926
П1.3.1. Функция <i>PointFromText()</i> .....	927
П1.3.2. Функция <i>PointFromWKB()</i> .....	929
П1.3.3. Функция <i>Point()</i> .....	930
П1.3.4. Функция <i>LineFromText()</i> .....	930
П1.3.5. Функция <i>LineFromWKB()</i> .....	931
П1.3.6. Функция <i>LineString()</i> .....	931
П1.3.7. Функция <i>PolyFromText()</i> .....	933
П1.3.8. Функция <i>PolyFromWKB()</i> .....	933
П1.3.9. Функция <i>Polygon()</i> .....	934
П1.3.10. Функция <i>GeomFromText()</i> .....	935
П1.3.11. Функция <i>GeomFromWKB()</i> .....	936
П1.3.12. Функция <i>MPointFromText()</i> .....	936
П1.3.13. Функция <i>MPointFromWKB()</i> .....	937
П1.3.14. Функция <i>MultiPoint()</i> .....	937
П1.3.15. Функция <i>MLineFromText()</i> .....	938
П1.3.16. Функция <i>MLineFromWKB()</i> .....	939
П1.3.17. Функция <i>MultiLineString()</i> .....	939
П1.3.18. Функция <i>MPolyFromText()</i> .....	940
П1.3.19. Функция <i>MPolyFromWKB()</i> .....	941
П1.3.20. Функция <i>MultiPolygon()</i> .....	941

П1.3.21. Функция <i>GeomCollFromText()</i> .....	942
П1.3.22. Функция <i>GeomCollFromWKB()</i> .....	943
П1.3.23. Функция <i>GeometryCollection()</i> .....	943
П1.4. Общие функции геометрических объектов .....	944
П1.4.1. Функция <i>Dimension()</i> .....	945
П1.4.2. Функция <i>Envelope()</i> .....	945
П1.4.3. Функция <i>GeometryType()</i> .....	948
П1.4.4. Функция <i>SRID()</i> .....	949
П1.5. Функции для работы с объектом <i>Point</i> .....	949
П1.5.1. Функция <i>X()</i> .....	949
П1.5.2. Функция <i>Y()</i> .....	950
П1.6. Функции для работы с объектом <i>LineString</i> .....	951
П1.6.1. Функция <i>EndPoint()</i> .....	951
П1.6.2. Функция <i>StratPoint()</i> .....	951
П1.6.3. Функция <i>PointN()</i> .....	952
П1.6.4. Функция <i>NumPoints()</i> .....	953
П1.6.5. Функция <i>GLength()</i> .....	953
П1.6.6. Функция <i>IsClosed()</i> .....	954
П1.7. Функции для работы с объектом <i>MultiLineString</i> .....	955
П1.7.1. Функция <i>GLenght()</i> .....	955
П1.7.2. Функция <i>IsClosed()</i> .....	956
П1.8. Функции для работы с объектом <i>Polygon</i> .....	957
П1.8.1. Функция <i>Area()</i> .....	957
П1.8.2. Функция <i>ExteriorRing()</i> .....	959
П1.8.3. Функция <i>InteriorRingN()</i> .....	960
П1.8.4. Функция <i>NumInteriorRings()</i> .....	961
П1.9. Функции для работы с объектом <i>MultiPolygon</i> .....	961
П1.9.1. Функция <i>Area()</i> .....	961
П1.10. Функции для работы с объектом <i>GeometryCollection</i> .....	962
П1.10.1. Функция <i>GeometryN()</i> .....	962
П1.10.2. Функция <i>NumGeometries()</i> .....	963
П1.11. Функции для проверки отношения минимальных ограничивающих прямоугольников .....	964
П1.11.1. Функция <i>MBRContains()</i> .....	964
П1.11.2. Функция <i>MBRDisjoint()</i> .....	964
П1.11.3. Функция <i>MBREqual()</i> .....	966
П1.11.4. Функция <i>MBRIntersects()</i> .....	967
П1.11.5. Функция <i>MBROverlaps()</i> .....	968
П1.11.6. Функция <i>MBRTouches()</i> .....	970
П1.11.7. Функция <i>MBRWithin()</i> .....	970
П1.12. Пространственные индексы .....	971
<b>Приложение 2. Описание компакт-диска .....</b>	<b>973</b>
<b>Предметный указатель .....</b>	<b>975</b>

# Введение

## Для кого и о чем эта книга

Книга посвящена описанию СУБД MySQL и особенностям ее диалекта SQL. Популярность реляционных баз данных неуклонно возрастает последние десятилетия. СУБД MySQL представляет собой СУБД с открытым кодом, разработкой которой занимается корпорация MySQL AB. СУБД MySQL успешно работает под управлением свыше 20 различных операционных систем (Windows, Linux, OS/2, Mac OS X, FreeBSD, Solaris, SunOS, OpenBSD, NetBSD, AIX, DEC UNIX, Tru64 UNIX и т. д.), а также предоставляет интерфейс для взаимодействия с множеством языков программирования: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tcl.

Принимая во внимание тот факт, что это одна из немногих СУБД с открытым кодом, и лицензионное соглашение, по которому она распространяется, не требует денежных отчислений (исключением является случай, когда MySQL входит в состав коммерческого продукта), не удивительно, что она получила широкое распространение в Российской Федерации. В частности MySQL является стандартом де-факто у российских хост-провайдеров.

На сегодняшний день MySQL, наряду с СУБД Oracle, является одной из самых быстрых баз данных. В версии MySQL 5.0 была достигнута функциональность, доступная ранее только в коммерческих базах данных. Так введены долгожданные вложенные запросы, хранимые процедуры, курсоры, обработчики ошибок, триггеры, представления и множество новых внутренних функций.

Книга будет полезна как начинающим разработчикам приложений с применением баз данных, которые хотят разобраться во всех тонкостях работы с MySQL, так и профессионалам, использовавшим в своей работе более ранние версии MySQL (3 и 4) и желающим познакомиться с нововведениями MySQL 5.0. Изложение в книге ведется таким образом, чтобы у читателя, не знакомого с языком запросов SQL, материал книги не вызвал затруднений. Сложность материала возрастает последовательно от первой главы до последней. На протяжении всей книги нововведения версий 4.1, 5.0 и 5.1 указываются явно в тексте, поэтому книга может быть полезна разработчикам, использующим более старые версии СУБД MySQL, но планирующим со временем перейти на новые версии.

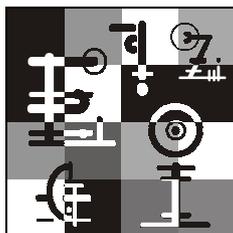
*Часть I* книги посвящена основам SQL и взаимодействию с сервером MySQL. В этой части ориентация идет в основном на начинающих разработчиков. *Часть II* посвящена более сложным вопросам, таким как внутренние функции, полнотекстовый поиск, регулярные выражения в MySQL, временные таблицы и переменные, вложенные и многотабличные запросы, транзакции и блокировки, учетные записи и администрирование. *Часть III* предоставляет сведения об административных средствах СУБД MySQL, утилитах, параметрах настройки, восстановлении данных и т. д. *Часть IV* полностью посвящена нововведениям 5-й версии СУБД MySQL — рассматриваются хранимые процедуры и функции, обработчики ошибок, курсоры, триггеры, представления и информационная схема. Наконец, в заключительной части (*часть V*) рассматривается взаимодействие СУБД MySQL с прикладными программами, разработанными с помощью языков программирования C/C++, Perl и PHP.

На протяжении всей книги используется учебная база данных электронного магазина, которую можно найти в каталоге base прилагаемого к книге компакт-диска, а также загрузить по адресу <http://www.softtime.ru/sql/base.zip>. Помимо этого, компакт-диск содержит дистрибутивы СУБД MySQL версий 5.0 и 5.1 для операционных систем Windows и Linux. Кроме того, на прилагаемом компакт-диске вы сможете найти графические клиенты для обращения к серверу MySQL.

По всем вопросам, которые могут возникнуть по мере чтения материала книги, вы можете обращаться на форум, расположенный на Web-сайте IT-студии SoftTime, сотрудниками которой являются авторы книги ([http://www.softtime.ru/forum/index.php?id\\_forum=3](http://www.softtime.ru/forum/index.php?id_forum=3)). Авторы присутствуют на форуме каждый день и ответят на возникшие у вас вопросы.

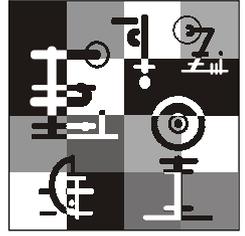
## Благодарности

Авторы выражают признательность сотрудникам издательства "БХВ-Петербург", благодаря которым наша рукопись увидела свет, а также компании MySQL AB и лично директору по связям с общественностью Стиву Кари (Steve Curry) за любезное разрешение поместить дистрибутивы MySQL на компакт-диске, прилагаемом к книге.



# ЧАСТЬ I

# ЗНАКОМСТВО С SQL



## Глава 1

# История развития баз данных. Понятие реляционной базы данных

Компьютерные технологии постоянно эволюционируют. Как следствие, программы становятся все более и более изощренными, а чем сложнее программа, тем больше кода необходимо для ее создания. Исследования показали, что плотность ошибок на 1000 строк кода не зависит от используемой технологии, а зависит лишь от квалификации программиста. Поэтому чем короче программа, тем меньше ошибок она содержит. В связи с этим основные усилия программистов направлены на создание таких языков программирования и технологий, которые позволили бы уменьшить объем программ, а следовательно, и ошибок.

За время существования компьютерной отрасли программисты сменили машинные коды на ассемблер, ассемблер — на языки высокого уровня. Каждый новый виток в технологии программирования обеспечивал одну простую вещь — программист мог создавать более сложные программы, т. к. рутинные операции брали на себя компиляторы и библиотеки.

Однако в компьютерной отрасли сложилась уникальная ситуация: на протяжении 40 лет выполняется закон Мура, каждые 18—24 месяца происходит удвоение производительности микрочипов при неизменном падении цены на них. Это приводит к тому, что компьютеры способны обрабатывать все больший объем данных и выполнять все более сложные программы.

Практически сразу после появления языков высокого уровня их средств стало не хватать для создания ясных и легко сопровождаемых программ, т. к. последние достигли объемов десятков тысяч строк (в одном файле). Ответом на это явилось структурное программирование, позволившее увеличить объем программы до 100 000 строк. Добиться этого удалось за счет особого набора правил и введения функций, позволивших разбить код на более мелкие блоки, оперировать которым много проще, чем огромной монолитной программой. Очень скоро и данного предела оказалось не достаточно, и был предложен объектно-ориентированный подход, который позволил создавать еще более сложные и объемные программы.

Программы — это процессы, которые манипулируют данными. Как же происходила эволюция данных? В далеких 60-х годах прошлого столетия программисты использовали машинные коды и файлы. В настоящий момент ни прикладное, ни даже системное

программное обеспечение в машинных кодах не пишется, однако файлы используются повсеместно. Значит ли это, что эволюция данных не происходила? Ответу на этот вопрос посвящена данная глава.

## 1.1. История развития СУБД. Реляционные базы данных

Задача длительного хранения и обработки информации появилась практически сразу с созданием первых компьютеров. Для решения этой задачи в конце 60-х годов XX века были разработаны специализированные программы, получившие название *систем управления базами данных* (СУБД). СУБД проделали длительный путь эволюции от системы управления файлами через иерархические и сетевые базы данных к реляционным моделям. В конце 80-х годов XX века доминирующей стала *система управления реляционными базами данных* (СУРБД). С этого времени такие СУБД стали стандартом де-факто. Каждая СУБД имела собственный язык запросов, и для того чтобы унифицировать работу с ними, был разработан *язык структурированных запросов* (Structured Query Language, SQL), который представляет собой язык управления реляционными базами данных. Цель данного раздела — рассмотреть реляционную модель данных, сравнив ее с более ранними СУБД: иерархическими и сетевыми.

### ЗАМЕЧАНИЕ

Взаимодействие с базой данных происходит при помощи *системы управления базой данных* (СУБД), которая расшифровывает запросы и производит операции с информацией в базе данных. Поэтому более правильно было бы говорить о запросе к СУБД и о взаимодействии приложения с СУБД. Но так как это несколько усложняет восприятие, далее везде мы будем говорить "база данных", подразумевая при этом СУБД.

Существуют следующие разновидности баз данных:

- система управления файлами;
- иерархические;
- сетевые;
- реляционные;
- объектно-ориентированные;
- гибридные.

### 1.1.1. Иерархические базы данных

Первыми базами данных были иерархические. Иерархическая база данных основана на древовидной структуре хранения информации и в этом смысле напоминает файловую систему компьютера. Наиболее известным и распространенным примером иерархической базы данных является Information Management System (IMS) фирмы IBM, первая версия которой появилась в 1968 году. С точки зрения организации хранения информации иерархическая база данных состоит из упорядоченного набора

деревьев одного типа, т. е. каждая запись в базе данных реализована в виде отношений "предок-потомок". На рис. 1.1 показана схема организации такой иерархической базы данных.

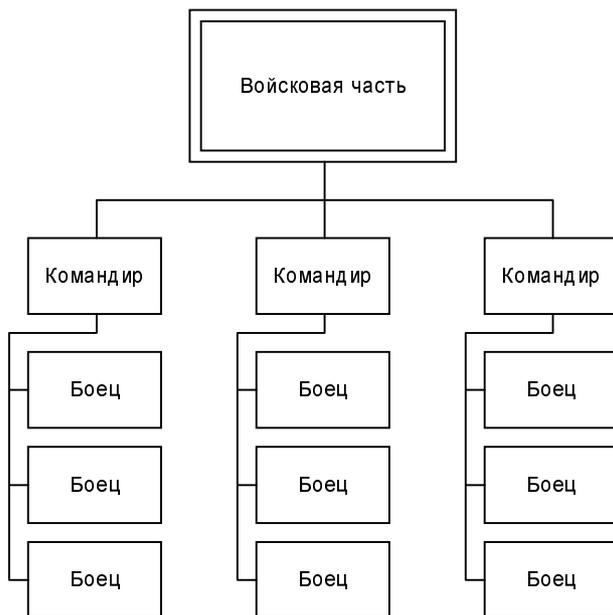


Рис. 1.1. Схема организации иерархической базы данных на примере войсковой части

В этой схеме объекты "Командир" являются потомками для объекта "Войсковая часть", а объекты "Боец" являются потомками для объекта "Командир". Вся база представляет собой иерархическое дерево, используя которое можно ответить на вопросы: сколько бойцов в части, сколько бойцов необходимо перебросить в часть, если она будет укомплектована еще 12 командирами и т. п.

Основной недостаток иерархической структуры базы данных — невозможность реализовать отношения "многие ко многим". Например, если мы создаем каталог книг, то одна книга может относиться сразу к нескольким разделам, в этом случае отношения будут выглядеть так, как это представлено на рис. 1.2.

Иерархические базы данных наиболее пригодны для моделирования структур, являющихся иерархическими по своей природе. Однако они не позволяют организовывать более сложные отношения, например, отнести отдельные книги сразу к нескольким подкаталогам (рис. 1.2), т. к. иерархия подразумевает только одного родителя. Если при моделировании воинской части это условие легко выполнить, поскольку воинские подразделения подразумевают единоначалие, то в остальных сферах деятельности разработчики столкнулись с тем, что иерархические базы данных плохо справляются с моделированием.

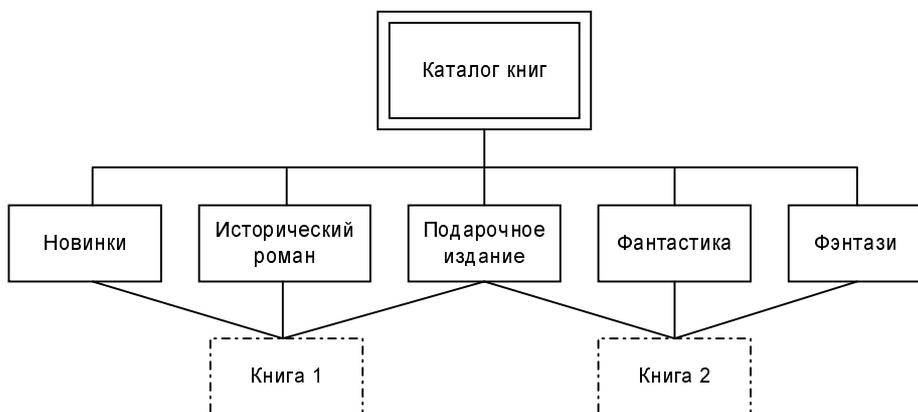


Рис. 1.2. Отношение "многие ко многим"

#### ПРИМЕЧАНИЕ

В настоящее время не разрабатываются СУБД, поддерживающие только иерархическую модель организации хранения данных. В лучшем случае проводится совмещение иерархической и сетевой модели (см. далее), которые допускают связывание древовидных структур между собой и установление связей внутри них. Такие СУБД называются *сетевыми даталогическими СУБД*.

### 1.1.2. Сетевые базы данных

Иерархические базы данных в силу большого количества недостатков просуществовали недолго и были заменены сетевыми базами данных, являющимися, по сути, расширением иерархических. То есть если в иерархических базах данных структура "запись-потомок" должна иметь только одного предка, то в сетевой базе данных потомок может иметь любое количество предков (см. рис. 1.2).

#### ПРИМЕЧАНИЕ

Термин "сетевая" в данном случае не подразумевает никакого намека на локальные сети. Этим словом обозначается модель связей в базе данных, когда каждая запись может находиться в отношениях "многие ко многим" с другими записями, что делает графическую модель базы похожей на рыбацкую сеть (рис. 1.3).

Схема сетевой базы данных на примере организации сети железнодорожных путей показана на рис. 1.3. Как видно из этого рисунка, каждая крупная железнодорожная станция может иметь связи (пути сообщения) с несколькими другими станциями (связь "многие ко многим"). Примерно таким же образом выглядит графическая схема любой сетевой базы данных.

Основным недостатком сетевых баз данных является сложность разработки больших приложений.



На рис. 1.4 приведен пример таблицы `catalogs` базы данных электронного магазина изделий компьютерных комплектующих, которые подразделяются на разделы. Каждая строка этой таблицы представляет собой один вид товарных позиций, для описания которой используется поле `id_catalog` — уникальный номер раздела, `name` — название раздела. Столбцы определяют структуру таблицы, а строки — число записей в таблице. Как правило, одна база данных содержит несколько таблиц, которые могут быть как связаны друг с другом, так и независимы друг от друга.

Таблица catalogs	
id_catalog	name
1	Процессоры
2	Материнские платы
3	Видеоадаптеры
4	Жесткие диски
5	Оперативная память

Строка

Столбец

Рис. 1.4. Таблица реляционной базы данных

Таблица products			Таблица catalogs	
id_product	...	id_catalog	id_catalog	name
1	...	2	1	Процессоры
2	...	5	2	Материнские платы
3	...	1	3	Видеоадаптеры
4	...	1	4	Жесткие диски
5	...	1	5	Оперативная память
6	...	5		
7	...	2		
8	...	4		
9	...	3		

Рис. 1.5. Связанные друг с другом таблицы

На рис. 1.5 приведена структура базы данных, состоящей из двух таблиц, `catalogs` и `products`. Таблица `catalogs` определяет число и названия разделов, а таблица `products` содержит описание товарных позиций. Одной товарной позиции соответствует одна строка таблицы. Последнее поле таблицы `products` содержит значения из поля `id_catalog` таблицы `catalogs`. По этому значению можно однозначно опреде-

лить, из какого раздела товарная позиция. Таким образом, таблицы получаются связанными друг с другом, и хотя эта связь условна, может отсутствовать и в большинстве случаев проявляется только в результате специальных запросов, именно благодаря наличию связей такая форма организации информации получила название *реляционной* (связанной отношениями).

#### ПРИМЕЧАНИЕ

В математике таблица, все строки которой отличаются друг от друга, называется *отношением*, — по-английски *relation*. Именно этому английскому термину реляционные базы данных и обязаны своим названием.

Конечно, при таком подходе теряется наглядность, которая присутствовала в иерархической и сетевой базах данных, зато значительно возрастает гибкость в организации данных.

#### ОПРЕДЕЛЕНИЕ

*Реляционной базой данных* называется такая база данных, в которой все данные организованы в виде таблиц, а все операции над этими данными сводятся к операциям над таблицами.

Сила реляционных баз данных заключается не только в уникальной организации информации в виде таблиц. Запросы к таблицам базы данных сами возвращают таблицы, которые называют результирующими таблицами. Даже если возвращается одно значение, это все равно считается таблицей, состоящей из одного столбца и одной строки. То, что SQL-запрос возвращает таблицу, очень важно. Это означает, что результаты запроса можно записать обратно в базу данных в виде таблицы, а результаты двух или более запросов, которые (т. е. результаты) имеют одинаковую структуру, можно объединить в одну таблицу. И, наконец, это говорит о том, что результаты запроса сами могут стать предметом дальнейших запросов.

### 1.2.1. Первичные ключи

Строки в реляционной базе данных неупорядочены, т. е. в таблице нет "первой", "последней", "тридцать шестой" и "сорок третьей" строк, а есть просто неупорядоченный набор строк. Возникает вопрос: каким же образом выбирать в таблице конкретную строку? Для этого в правильно спроектированной базе данных для каждой таблицы создается один или несколько столбцов, значения которых во всех строках различны. Такой столбец называется *первичным ключом* таблицы. Первичный ключ обычно сокращенно обозначают как РК (*primary key*). Никакие из двух записей таблицы не могут иметь одинаковых значений первичного ключа, т. е. благодаря наличию первичных ключей каждая строка таблицы обладает своим уникальным идентификатором. Так на рис. 1.5 в качестве первичного ключа таблицы *products* выступает поле *id\_product*, а в качестве первичного ключа таблицы *catalogs* — поле *id\_catalog*.

По способу задания первичных ключей различают *логические* (естественные) ключи и *суррогатные* (искусственные).

Для логического задания первичного ключа необходимо выбрать в базе данных то, что естественным образом определяет запись. Примером такого ключа является номер паспорта в базе данных о паспортных данных жителей. К примеру, в таблице базы данных, содержащей паспортные данные, уникальный индекс можно создать для поля "номер паспорта", поскольку каждый такой номер является единственным в своем роде. Однако дата рождения уже не уникальна, поэтому поле "Дата рождения" не может выступать в качестве первичного ключа.

Если подходящих примеров для естественного задания первичного ключа не находится, пользуются суррогатным ключом. Суррогатный ключ представляет собой дополнительное поле в базе данных, предназначенное для обеспечения записей первичным ключом (именно такой подход принят на рис. 1.5).

### СОВЕТ

Даже если в базе данных содержится естественный первичный ключ, лучше использовать суррогатные ключи, поскольку их применение позволяет абстрагировать первичный ключ от реальных данных. В этом случае облегчается работа с таблицами, поскольку суррогатные ключи не связаны ни с какими фактическими данными этой таблицы.

Как уже упоминалось, в реляционных базах данных практически всегда разные таблицы логически связаны друг с другом. Одним из предназначений первичных ключей и является однозначная организация такой связи.

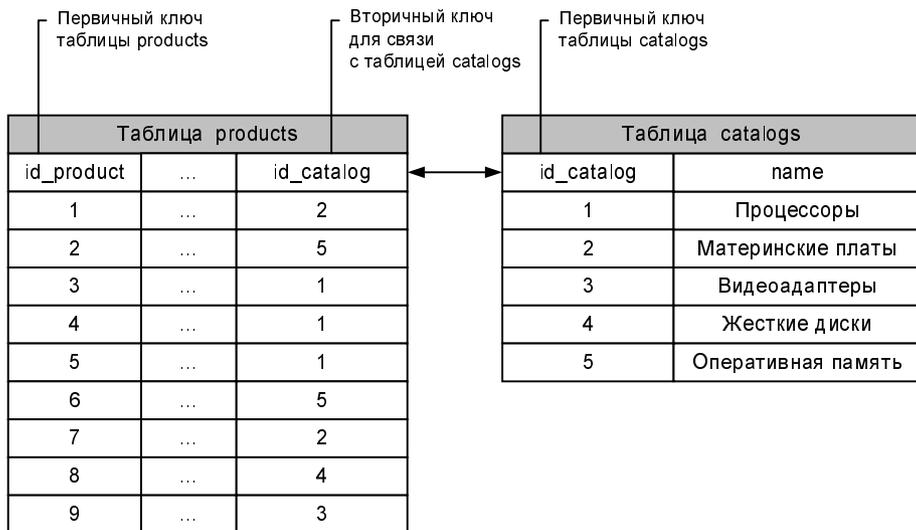


Рис. 1.6. Связь между таблицами products и catalogs

Например, рассмотрим упоминавшуюся базу данных электронного магазина (рис. 1.6). Каждая из таблиц имеет свой первичный ключ, значение которого уникально в пределах таблицы. Еще раз подчеркнем, что таблица не обязана содержать первичные

ключи, но это очень желательно, если данные связаны друг с другом. Столбец `id_catalog` таблицы `products` принимает значения из столбца `id_catalog` таблицы `catalogs`. Благодаря такой связи мы можем выстроить иерархию товарных позиций и определить, к какому разделу они относятся. Поле `id_catalog` таблицы `products` называют внешним или вторичным ключом. Внешний ключ сокращенно обозначают как FK (foreign key).

## 1.2.2. Нормализация базы данных

*Нормализацией* схемы базы данных называется процедура, производимая над базой данных с целью удаления в ней избыточности.

Для того чтобы лучше уяснить приведенное определение нормализации, рассмотрим следующий пример. В электронном магазине, таблицы которого рассматриваются, необходимо учитывать не только товарные позиции, но и покупки. Это может быть полезно для автоматического пополнения запасов на складе, ведения бухгалтерии, отслеживания предпочтений покупателей или для предоставления им скидок. Для учета сделок можно использовать таблицу `orders`, структура которой представлена на рис. 1.7.

Таблица orders				
id_order	user	ordertime	number	id_product
1	Симдянов И. В.	2005-10-04	1	8
2	Корнеев А. А.	2005-02-10	2	10
3	Иванов В. А.	2005-02-18	4	20
4	Симдянов И. В.	2005-03-10	1	20
5	Симдянов И. В.	2005-03-17	1	20

Рис. 1.7. Пример избыточности в таблице базы данных

Представленная на рис. 1.7 таблица содержит первичный ключ `id_order`, фамилию и инициалы покупателя `user`, время покупки `ordertime`, число приобретенных товаров `number` и вторичный ключ `id_product`, по которому в таблице `products` можно определить приобретенную товарную позицию. Эта таблица избыточна — для каждого заказа вводится имя покупателя, что может приводить к ошибкам и замедлять операции, связанные с выбором заказов каждого из покупателей. Для нормализации таблицы ее обычно разбивают на две, как это показано на рис. 1.8.

Теперь в таблице `orders` вместо фамилии покупателя используется вторичный ключ `id_user` для связи с таблицей `users`, в которую вынесена вся информация о каждом покупателе. Как видно из рис. 1.8, разбиение таблицы `orders` на две связанные таблицы позволило добавить контактную информацию для каждого из покупателей. В противном случае потребовалось бы вводить дополнительные поля в таблицу `orders` и дублировать информацию при каждой новой покупке. При этом изменение

контактной информации потребовало бы обновления всех записей таблицы `orders`, соответствующих данному покупателю.

Таблица orders				
id_order	id_user	ordertime	number	id_product
1	1	2005-10-04	1	8
2	2	2005-02-10	2	10
3	3	2005-02-18	4	20
4	1	2005-03-10	1	20
5	1	2005-03-17	1	20

Таблица users			
id_user	name	phone	email
1	Симдянов И. В.	9056666100	
2	Корнеев А. А.	89-78-36	korneev@domen.ru
3	Иванов В. А.	58-98-78	ivanov@mail.ru

Рис. 1.8. Нормализованная база данных

Очевидно, что нормализация несет в себе немало преимуществ: в нормализованной базе данных уменьшается вероятность возникновения ошибок, она занимает меньше места на жестком диске и т. д.

#### **ЗАМЕЧАНИЕ**

Хотя в теории баз данных и говорится о том, что схема базы данных должна быть полностью нормализована, в реальности при работе с полностью нормализованными базами данных необходимо применять весьма сложные SQL-запросы, что приводит к обратному эффекту — замедлению работы базы данных. Поэтому иногда для упрощения запросов даже прибегают к обратной процедуре — *денормализации*.

## **1.3. СУБД и сети**

Развитие компьютерных сетей, а тем более рост популярности Интернета, сыграли серьезную роль в развитии СУБД. Если раньше и приложение, и СУБД находились на одном центральном главном компьютере (мэйнфрейме), то теперь они отделены друг от друга: клиентская часть выполняется на рабочих машинах, а СУБД и собственно сама база данных находятся на сервере. Клиент и сервер взаимодействуют друг с другом посредством специального языка запросов SQL. Рассмотрим основные этапы эволюции СУБД, связанной с развитием компьютерных сетей.