

Содержание

ВВЕДЕНИЕ	13
Часть I ❖ Проектирование базы данных	15
Глава 1 ❖ Немного истории	16
Создание структурированного языка запросов	17
Возможности SQL	18
InterBase	19
Реализация SQL в InterBase	20
Диалекты InterBase SQL.....	22
Резюме	22
Глава 2 ❖ Введение в реляционную модель	23
Сущность и атрибуты.....	24
Реляционная связь	26
Целостность данных	27
Доменные ограничения	27
Целостность сущностей.....	28
Ссылочная целостность.....	28
Корпоративная целостность	29
Модель «сущность–связь»	29
Нормализация данных	30
Резюме	32
Глава 3 ❖ Создание базы данных	33
Создание БД в консоли администрирования	33
Управление БД средствами SQL.....	36
Создание БД, CREATE DATABASE.....	36
Изменение БД, ALTER DATABASE.....	38
Уничтожение БД, DROP DATABASE	39
Резюме	39
Глава 4 ❖ Типы данных и домены	40
Типы данных.....	40
Точные числовые типы	41
Приближённые числовые типы.....	41
Логический тип	42
Строки символов	42
Дата и время	43
Большие бинарные объекты	45
Преобразование типов данных.....	46
Операции с данными	47
Проверка на равенство значений	47
Домены	47
Определение домена в консоли администрирования.....	49

Создание домена CREATE DOMAIN	50
Изменение домена ALTER DOMAIN	54
Удаление домена DROP DOMAIN	54
Как получить информацию о домене?	54
Резюме	55

Глава 5 ❖ Таблицы56

Ключи.....	57
Создание значений ключа с помощью генератора.....	58
Таблицы и консоль администрирования	59
Создание таблицы, CREATE TABLE.....	61
Определение столбцов таблицы.....	62
Первичный ключ.....	66
Внешние ключи и связи между таблицами	66
Изменение таблицы ALTER TABLE.....	69
Удаление таблицы DROP TABLE.....	71
Временные таблицы	71
Резюме	72

Глава 6 ❖ Представления.....73

Представления и консоль администрирования	74
Создание представления, CREATE VIEW	75
Модифицируемые представления	76
Ограничение WITH CHECK OPTION.....	78
Изменение представления	78
Удаление представления, DROP VIEW	78
Резюме	79

Глава 7 ❖ Индексы.....80

Индексы на основе B-деревьев.....	81
Правила назначения пользовательских индексов.....	85
Создание индексов в консоли администрирования	86
Создание индекса, CREATE INDEX.....	87
Подключение/отключение индекса, ALTER INDEX	87
Удаление индекса, DROP INDEX.....	88
Избирательность индекса	88
Резюме	90

Глава 8 ❖ Выборка данных и инструкция SELECT.....91

Порядок сортировки, ORDER BY	93
Ограничение количества строк в результирующем наборе	94
Условие отбора данных, предложение WHERE	94
Сравнение.....	95
Попадание в диапазон, BETWEEN.....	96
Начало строки с подстроки, STARTING WITH.....	96
Проверка на входжение подстроки, CONTAINING.....	97
Соответствие шаблону, LIKE	97
Проверка на неопределённость, IS NULL	98

Вложенные запросы и проверка существования.....	99
Подзапрос IN.....	99
Проверка существования EXISTS.....	100
Множественное сравнение.....	101
Агрегирующие функции.....	102
Группировка данных GROUP BY.....	103
Дополнительная фильтрация группы строк, HAVING.....	104
Резюме.....	104
Глава 9 ❖ Многотабличные запросы SELECT.....	105
Внутреннее соединение с помощью WHERE.....	106
Соединение JOIN.....	106
Соединение нескольких таблиц.....	108
Запросы к иерархическим данным.....	110
Слияние UNION.....	111
Оптимизация запроса.....	112
Резюме.....	114
Глава 10 ❖ Редактирование данных.....	115
Вставка, инструкция INSERT.....	115
Модификация, инструкция UPDATE.....	116
Удаление, инструкция DELETE.....	118
Применение UDF-функций.....	118
Резюме.....	119
Глава 11 ❖ Процедурный SQL.....	120
Хранимая процедура.....	121
Комментарии.....	122
Переменные.....	123
Выборка данных с помощью SELECT...INTO.....	124
Условный оператор IF...THEN...ELSE.....	124
Цикл WHILE...DO.....	125
Цикл выборки данных FOR SELECT...DO.....	126
Оператор SUSPEND.....	127
Оператор EXIT.....	128
Вызов процедуры, EXECUTE PROCEDURE.....	128
Триггер.....	129
Контекстные переменные.....	130
Преобразование данных.....	132
Ввод значений по умолчанию.....	132
Поддержка корпоративной целостности данных.....	133
События.....	134
Что нельзя сделать в процедурном SQL?.....	135
Резюме.....	135
Глава 12 ❖ Внешние функции.....	136
Размещение UDF-библиотеки.....	136
Подключение внешней функции к БД.....	137

Подключение UDF в консоли администрирования.....	139
Вызов UDF.....	140
Разработка UDF-библиотек в Delphi.....	140
Работа со строками.....	141
Особенности разработки в C++ Builder.....	142
Резюме	144
Глава 13 ❖ Обработка исключений	145
Исключения в InterBase	146
Создание и вызов пользовательского исключения	147
Модификация и удаление исключения.....	148
Обработка исключений, выражение WHEN...DO	148
Ведение протокола исключений.....	150
Объектная модель исключений в Delphi.....	151
Обработка исключений в Delphi.....	152
Централизованная обработка исключений в приложении	155
Настройка среды разработки для обработки ИС	157
Резюме	158
Глава 14 ❖ Транзакции и параллельная обработка данных ...	159
Проблемы параллельного доступа к данным.....	160
Управление транзакциями в InterBase.....	161
Подходы к сериализации транзакций	161
Идентификация транзакции.....	162
Состояние транзакции.....	163
Многоверсионная архитектура.....	163
Уровни изоляции транзакций в стандарте SQL.....	164
Уровни изоляции транзакций в InterBase	165
Разрешение блокировок.....	166
Управление транзакцией средствами SQL	166
Определение транзакции, SET TRANSACTION.....	167
Фиксация транзакции, COMMIT WORK.....	168
Откат транзакции, ROLLBACK	169
Точки сохранения	169
Резюме	169
Глава 15 ❖ Аутентификация и авторизация	170
Способы аутентификации в InterBase.....	171
Выбор способа аутентификации	172
Учётные записи.....	174
Предопределённые полномочия.....	174
Создание учётных записей средствами SQL.....	175
Управление наборами привилегий	176
Ролевой доступ	176
Резюме	180
Глава 16 ❖ Шифрование данных	181
Подготовка к шифрованию, действия SYSDBA	181

Действия SYSDSO	182
Создание пароля SEP	182
Создание криптографического ключа.....	183
Полномочия на шифрование и расшифровку	184
Шифрование данных.....	185
Расшифровка данных	186
Резюме	186

Часть II ❖ Разработка клиентского приложения 187

Глава 17 ❖ Доступ к БД из клиентского приложения 188

Структура клиентского приложения InterBase	188
Доступ к базе данных, компонент TIBDatabase.....	189
Выбор диалекта ISQL.....	190
Создание и уничтожение базы данных	190
Соединение с базой данных	191
Разрыв соединения	195
Информирование о составе БД	196
Контроль за транзакциями.....	198
Файл инициализации БД, TIBDatabaseINI	199
Резюме	200

Глава 18 ❖ Транзакции и компонент TIBTransaction 201

Параметры транзакции	201
Автоматическое управление транзакцией	203
Управление транзакцией в ручном режиме.....	203
Точки сохранения	205
Тайм-аут транзакции	205
Управление несколькими соединениями	206
Диагностика состояния транзакции.....	207
Резюме	207

Глава 19 ❖ Работа со скриптами SQL 208

Выполнение скриптов, TIBScript.....	208
Разработка помощника установки БД	210
Получение метаданных, TIBExtract.....	212
Пример генератора скриптов.....	214
Резюме	215

Глава 20 ❖ Общие черты наборов данных IBX 216

Функционал набора данных	217
Подключение к объектам БД.....	218
Открытие и закрытие набора данных	218
Обновление набора данных	219
Перемещение по записям.....	220
Двунаправленный и однонаправленный курсор	222
Закладки	222
Состояние набора данных.....	225

Редактирование записей в наборе.....	227
Отложенные обновления.....	229
Доступ к отдельному столбцу.....	232
Вычисляемые поля.....	234
Поле генератора.....	234
Фильтрация набора данных.....	236
Поиск строки в наборе.....	238
Обработка событий.....	239
Взаимодействие с визуальными элементами управления.....	241
Резюме.....	241

Глава 21 ❖ Поле набора данных 242

Базовый класс TField.....	242
Тип обслуживаемых данных.....	244
Функциональное назначение.....	244
Обращение к полю.....	245
Доступ к значению.....	246
Размер поля.....	249
Значение по умолчанию.....	249
Ограничения на ввод данных.....	250
Маска ввода.....	251
Индексные поля.....	252
Отображение данных.....	252
Обработка событий.....	254
Искусственные поля.....	255
Поля подстановки.....	255
Вычисляемые поля.....	260
Числовые поля.....	261
Поля целых чисел.....	263
Поля вещественных чисел.....	263
Текстовое поле.....	264
Логическое поле.....	264
Поля даты и времени.....	265
Поля BLOB.....	265
Резюме.....	268

Глава 22 ❖ Компонент быстрой разработки TIBTable..... 269

Программирование без кода.....	269
Особенности подключения.....	271
Отношение между главной и подчинённой таблицами.....	271
Очистка, удаление таблицы.....	273
Работа с индексами.....	273
Создание и удаление индексов.....	275
Создание таблицы.....	276
Описание состава полей таблицы, класс TFieldDefs.....	276
Описание состава индексов таблицы, класс TIndexDefs.....	278
Пример использования метода CreateTable.....	280
Резюме.....	281

Глава 23 ❖ Компоненты для работы с SQL.....	282
Инструкция SQL TIBSQL.....	282
Подготовка к работе.....	283
Обслуживание полученного набора данных.....	285
Запрос TIBQuery.....	285
Коллекция TParams и динамический SQL.....	287
Параметр TParam.....	289
Пример запроса с параметром.....	290
Хранимая процедура TIBStoredProc.....	291
Универсальный набор данных TIBDataSet.....	292
Формирование запросов.....	293
Модифицируем данные, TIBUpdateSQL.....	295
Резюме.....	297
Глава 24 ❖ Реакция на события сервера в клиентском приложении.....	299
Компонент TIBEvents.....	299
Пример реакции на события.....	300
Резюме.....	303
Глава 25 ❖ Компоненты Data Controls.....	304
Источник данных – компонент TDataSource.....	304
Общие черты компонентов Data Controls.....	306
Сетка TDBGrid.....	307
Одновременный выбор нескольких строк.....	308
Колонки сетки.....	309
Обработка событий.....	315
Статический текст TDBText.....	319
Строка ввода TDBEdit.....	320
Многострочный редактор TDBMemo.....	321
Изображение TDBImage.....	322
Список строк TDBListBox.....	323
Комбинированный список TDBComboBox.....	324
Группа переключателей TDBRadioGroup.....	324
Кнопка выбора TDBCheckBox.....	324
Компонент TDBCtrlGrid.....	325
Списки с полями подстановки.....	327
Список подстановки TDBLookupListBox.....	329
Комбинированный список подстановки TDBLookupComboBox.....	329
Навигатор TDBNavigator.....	329
Резюме.....	331
Глава 26 ❖ Иерархические данные.....	332
Дерево TTreeView.....	332
Пример работы с рекурсивными таблицами.....	334
Сбор данных.....	335
Новая запись.....	337

Редактирование записи	337
Удаление записи	339
Сортировка узлов	340
Переподчинение узлов.....	341
Резюме	345

Глава 27 ❖ Нестандартное представление данных 346

Компоненты списки	346
Список с флажками выбора TCheckBox.....	348
Сетка строк TStringGrid	349
Пример работы с сеткой строк.....	352
Развитие примера	354
Список просмотра TListView	356
Хранение данных в списке, коллекция TListItems.....	356
Элемент списка TListItem	357
Стиль представления данных.....	358
Особенности работы списка со стилем vsReport.....	358
Выбор элементов списка.....	359
Упорядочивание элементов.....	361
Поиск элементов	361
Группировка элементов	362
Пример работы со списком просмотра	362
Резюме	366

Глава 28 ❖ Библиотеки DLL на стороне клиента 367

Общая характеристика DLL	367
Экспорт функций DLL.....	368
Разработка библиотеки доступа к БД.....	368
Подключение библиотеки к приложению	372
Неявная загрузка DLL.....	372
Явная загрузка DLL.....	373
Резюме	375

Глава 29 ❖ Компоненты для администрирования

InterBase 376

Общие черты компонентов.....	376
Свойства сервера, TIBServerProperties.....	379
Конфигурирование БД, TIBConfigService.....	382
Отключение и перезапуск БД	383
Управление опциями.....	384
Ведение журнала транзакций.....	386
Шифрование данных.....	388
Протокол работы сервера, TIBLogService.....	389
Статистика, TIBStatisticalService.....	390
Валидация и восстановление, TIBValidationService.....	392
Управление учётными записями, TIBSecurityService	394
Резервное копирование и восстановление, TIBBackupService и TIBRestoreService.....	399
Информация о БД, компонент TIBDatabaseInfo.....	402

Монитор SQL, TSQLMonitor	403
Резюме	404
Глава 30 ❖ Многоуровневые БД по технологии DataSnap....	405
Архитектура трёхзвенного проекта БД DataSnap	406
Сервер TDSServer.....	408
Класс сервера TDSServerClass	409
Обмен данными клиент–сервер, TDSTCPServerTransport и TDSHTTPService.....	411
Менеджер аутентификации, TDSAuthenticationManager.....	413
Метод сервера TSqlServerMethod	416
Пример проекта DataSnap	416
Подготовка клиентского приложения.....	420
Подключение сервера приложений к БД	421
Получение данных клиентским приложением	422
Модификация данных	424
Доступ к методу вставки записи на стороне клиента	424
Механизм обратного вызова	425
Резюме	427
Глава 31 ❖ Диаграммы и графики	428
Диаграмма TChart	428
Хранение графиков в диаграмме.....	429
Базовый класс графиков, TChartSeries	431
Легенда диаграммы TChartLegend.....	438
Координатные оси диаграммы TChartAxis	439
Масштабирование	442
Многостраничные диаграммы	443
Экспорт диаграмм	444
Печать диаграммы	445
Упорядочивание графиков внутри диаграммы.....	446
Обеспечение объёмного вида диаграммы	446
Пример работы с TChart	447
Особенности диаграммы TDBChart	451
Пример работы с TDBChart	451
Резюме	452
Глава 32 ❖ Создаём интерфейс руками пользователя	454
Общие требования к интерфейсу.....	454
Стили оформления приложения	455
Менеджер стилей TStyleManager	457
Настройка интерфейса во время выполнения приложения, диалог TCustomizeDlg	459
Настройка горячих клавиш THotKey.....	463
Использование системного реестра, TRegistryIniFile.....	464
Резюме	465
Глава 33 ❖ Электронные отчёты.....	466
Технология автоматизации.....	467
Обращение к серверу автоматизации	469

Понятие коллекции.....	470
Редактор Microsoft Word.....	470
Приложение Word – Application.....	471
Коллекция документов Documents и документ Document.....	472
Параметры страницы, объект PageSetup.....	474
Область документа Range.....	475
Выделанная область Selection.....	477
Абзацы Paragraphs.....	479
Списки Lists.....	480
Коллекция разделов Sections и раздел Section.....	482
Коллекция таблиц Tables и таблица Table.....	484
Внедрение в документ внешних объектов.....	488
Пример универсального генератора отчётов.....	490
Электронные таблицы Microsoft Excel.....	491
Приложение Excel – Application.....	491
Коллекция WorkBooks и книга Workbook.....	493
Листы Excel.....	494
Область ячеек Range.....	499
Пример табличного отчёта.....	501
Пример создания диаграммы.....	502
Резюме.....	503

Глава 34 ❖ Особенности выпуска клиентских приложений FM..... 505

Создание приложения FM.....	506
Выпуск приложения для OS X.....	507
Выпуск приложения InterBase ToGo.....	509
Резюме.....	509

Заключение..... 512

Список использованных источников..... 513

Приложение 1 ❖ Модель демонстрационной БД «Склад» ... 514

Приложение 2 ❖ Встроенные функции..... 516

Приложение 3 ❖ Функции UDF..... 517

Приложение 4 ❖ Листинги примеров..... 518

Предметный указатель..... 519

ВВЕДЕНИЕ

Давным-давно, в середине 90-х гг. прошлого века при появлении на свет самой первой версии Delphi (в те времена ещё принадлежащей компании Borland) о новой среде проектирования в первую очередь говорили как о превосходной платформе для разработки настольных и клиент-серверных приложений для баз данных. Спустя почти 20 лет Delphi не только не утратила своих позиций в этом сегменте компьютерного рынка, но и существенно нарастила свои возможности. Судите сами, Delphi позиционируется как среда проектирования, поддерживающая самые распространённые и успешные системы управления базами данных (СУБД), такие как Oracle, Informix, Microsoft SQL Server, DB2, Sybase, MySQL, Firebird, PostgreSQL, и, конечно же, свой собственный сервер – InterBase.

Если вы только начинаете изучать клиент/серверные технологии реляционных баз данных (БД) и рассчитываете разработать свою первую БД, то самой лучшей платформой для этого станет сервер InterBase. Тому несколько причин. Во-первых, это весьма компактный программный продукт, для инсталляции которого достаточно пары десятков мегабайт на жёстком диске. Во-вторых, в сравнении с большинством современных коммерческих систем управления базами данных (СУБД) система InterBase предъявляет минимальные требования к ресурсам компьютера, на котором она станет функционировать. В-третьих, на работе с InterBase специализируются одни из самых успешных языков программирования – Delphi и C++ Builder. В-четвёртых, для взаимодействия клиентского приложения с сервером InterBase достаточно установить на компьютер всего одну библиотеку. В-пятых, это настоящая кросс-платформа, поддерживающая не только Windows, но и OS X, Linux, Solaris. В-шестых, у InterBase есть очень схожий коллега – бесплатный сервер Firebird, с которым вы без проблем сможете работать, прочитав эту книгу.



Несмотря на то что книга посвящена InterBase, большая часть излагаемого материала вполне подходит для применения при разработке клиент/серверных БД для СУБД Firebird. Это объясняется тем, что бесплатная СУБД Firebird была создана на основе открытого кода InterBase 6 и поэтому хорошо совместима со своим именитым предком.

Сервер InterBase позволяет создавать три варианта клиент-серверных приложений:

- сервер БД и клиент развёрнуты на одном и том же компьютере;
- классическая двухуровневая архитектура, в которой клиент получает доступ к серверу, установленному на другом сетевом компьютере;
- многоуровневая архитектура, когда между сервером и клиентом появляется дополнительный уровень – сервер приложений.

Кроме того, при использовании современной кросс-платформенной библиотеки FM (ранние версии библиотеки FM назывались FireMonkey) вы сможете создавать приложения БД для OS X и мобильных устройств, работающих на основе Android и iOS. Для этих целей разработано упрощённое ядро InterBase – InterBase ToGo и IBLite.

Если заглянуть под «капот» InterBase, то вы наверняка окажетесь впечатлены весьма серьёзными возможностями СУБД. Некоторые показательные технические характеристики последнего на момент написания этих страниц книги сервера InterBase XE3 приведены в табл. 1.

Таблица 1. Основные технические характеристики InterBase XE3

Параметр	Значение
Максимальный размер БД	При стандартном размере страницы в памяти 4 Кб может достигать 8 терабайт, с увеличением страницы до 16 Кб – 32 Тб
Максимум таблиц в БД	32 640
Максимальный размер записи	64 Кб (без учёта BLOB)
Максимум записей в таблице	2 ³²
Максимальное число событий и хранимых процедур	Не лимитируется, ограничение касается лишь размера кода процедур и триггеров – он не должен превышать 48 Кб
Максимальное число строк и колонок в таблице	2 ³²
Максимум индексов для одной таблицы	255
Максимум столбцов таблицы в составном индексе	16
Максимальное количество таблиц, участвующих в операции соединения JOIN	Не ограничено, однако для обеспечения наилучшей производительности рекомендуется одновременно соединять не более 16 таблиц
Максимум индексов для БД	2 ³²
Максимальный размер BLOB-данных	Определяется размером страницы в памяти. При размере страницы 4 Кб может достигать 4 Гб
Максимальное число вложенных вызовов хранимых процедур и триггеров	Для Windows – 750 Для UNIX – 1000
Диапазон хранимых дат	От 1 января 100 г. н. э. до 29 февраля 32 768 г. н. э.

Завершая введение, отметим ещё один важный плюс InterBase – эта СУБД относится к категории продуктов «zero administration», то есть практически не требующих администрирования. Другими словами, чтобы воспользоваться разработанным вами программным комплексом на основе InterBase, руководители предприятий и организаций не должны задумываться о введении в штат дополнительной единицы – администратора БД. Это весьма немаловажный аргумент, который вы сможете положить на чашу весов при продвижении в свет своих БД.

Надеюсь, что приведённые аргументы произвели на читателя должное впечатление и ему уже не терпится приступить к работе – изучению InterBase.

Часть I

**ПРОЕКТИРОВАНИЕ
БАЗЫ ДАННЫХ**

Немного истории

На сегодняшний день наиболее распространённым способом хранения структурированных данных являются реляционные базы данных. Создателем реляционной модели считается математик Эдгар Фрэнк Кодд (Edgar Frank Codd, 1923–2003 гг.). Датой рождения реляционной БД можно считать июнь 1970 г. Именно тогда Кодд (на тот момент времени сотрудник одной из лабораторий корпорации IBM) опубликовал свою знаменитую статью «Реляционная модель данных для больших совместно используемых банков данных», в которой впервые прозвучал столь популярный сегодня термин «реляционная модель».

Первопричиной возникновения нового по тем временам подхода к проектированию баз данных послужили существенные ограничения предыдущих моделей. Ни сетевая, ни иерархическая модели не были способны просто и доступно описывать подлежащие учёту данные. Кодд сумел объединить на первый взгляд несовместимые вещи – с одной стороны, реляционная модель опиралась на математические выкладки, а с другой – была понятна рядовому пользователю, состоящему в конфронтации даже с таблицей умножения.

Работа над реляционной моделью объединяла как теоретические, так и практические изыскания всех предшественников. Первый существенный результат пришёл в 1976 г. В этом году в исследовательской лаборатории корпорации IBM, расположенной в городе Сан-Хосе, штат Калифорния, на свет появился прототип современных реляционных БД – проект System-R. Руководителем проекта был Мортон Астрахан (Morton M. Astrahan).

Этот проект преследовал цель доказать практичность реляционной модели, что достигалось посредством реализации предусмотренных ею структур данных и требуемых функциональных возможностей. На основе этого проекта был разработан структурированный язык запросов (в ту пору названный SEQUEL), который несколько позднее стал стандартом SQL.

На базе System-R впоследствии (в 1979 г.) был создан первый успешный коммерческий реляционный продукт фирмы IBM – DB2. Говоря о DB2, нельзя не упомянуть одного из её авторов – Криса Дж. Дейта (Chris J. Date). На сегодняшний день это ведущий специалист по реляционной модели данных, в России широко известна и многократно переиздавалась его книга «Введение в системы баз данных» [3], на которой выросло не одно поколение разработчиков БД.



Реляционной модели данных посвящено много фундаментальных трудов, в которых подробно изложены все ключевые аспекты модели. Обязательно рекомендую читателю хотя бы в обзорном порядке ознакомиться с работами ведущих специалистов в этой области [3–5]. Задача автора несколько прозаичнее, он предполагает, что читатель знаком с азами реляционной модели.

Создание структурированного языка запросов

В середине 70-х гг. XX в., сразу после появления реляционной модели, специалисты БД приступили к разработке принципиально нового языка, предназначенного для управления данными. Среди огромного количества пожеланий, предъявляемых к делающему первые шаги языку, мы выделим самые ключевые. Перспективный язык реляционных баз данных должен был позволять:

- создавать базы данных, таблицы и другие объекты БД;
- выполнять основные операции редактирования данных в таблицах (вставка, модификация и удаление);
- выполнять запросы пользователя к данным, преобразующие хранящиеся в таблицах данные в выходные отношения.

Ко всему прочему разрабатываемый язык должен был в принципе отличаться от высокоуровневых языков программирования тех лет. Во-первых, базы данных работают в трёхзначной логике. У них наряду с классическими для любого языка понятиями истина/ложь (FALSE/TRUE) предусмотрено третье значение неопределённости UNKNOWN. Во-вторых, новый язык создавался не только в интересах программистов, но и в интересах пользователей, поэтому в идеале он должен быть не процедурным, а декларативным¹. В соответствии с этим пользователь лишь ставит БД задачу (указывает, что ему нужно от БД), а каким образом СУБД станет решать поставленную задачу, пользователя не интересует.

Стандартом SQL (Structured Query Language) стал в 1986 г. благодаря Американскому национальному институту стандартов (American National Standards Institute, ANSI) и Международной организации стандартизации (International Organization for Standardization, ISO). Кстати, первый стандарт SQL иногда называют по имени принявшей его организации – ANSI SQL.

В 1990-х гг. официально действующим и общепризнанным стал считаться стандарт SQL:92, принятый, как вы уже догадались, в 1992 г. Практически любая серьёзная компания, разрабатывающая СУБД, старается поддерживать требования SQL:92.

В 1999 г. заговорили об очередном (третьем по счёту) стандарте SQL. В этом году было опубликовано пять частей стандарта SQL-3 (SQL:99):

- SQL/Framework – концептуальная структура стандарта;
- SQL/Foundation – базисное описание SQL;
- SQL/CLI – уточнения к интерфейсу уровня вызовов;
- SQL/PCM – уточнение описания хранимых процедур;
- SQL/Bindings – определение правил взаимодействия SQL и ряда стандартных языков программирования.

Спустя некоторое время появились ещё три части стандарта:

- SQL/MED – управление внешними данными;
- SQL/OLB – правила взаимодействия с объектно-ориентированными языками;
- SQL/Schemata – информационная схема.

¹ InterBase SQL сочетает черты как декларативного, так и процедурного языка.


Однако многие специалисты вновь скептически отнеслись SQL-3, обвинив его в незавершённости. Во многом по этой причине в 2003 г. к вопросу модернизации SQL вернулись вновь. В обновлённый стандарт с необходимыми изменениями вошли все части прежнего SQL:99 (правда, часть SQL/Bindings в самостоятельном виде существовать перестала и была включена во вторую часть стандарта SQL/Foundation). В дополнение к перечисленным выше частям SQL:2003 приобрёл ещё несколько документов:

- SQL/JRT – взаимодействие с языком Java;
- SQL/XML – работа с XML-документами.

На сегодня последним действующим стандартом считается SQL:2003, этот стандарт постепенно дорабатывается (наиболее существенные изменения вносились в 2006 и 2008 г.).

 Полную спецификацию стандарта SQL вы можете найти в Интернете по адресу <http://www.wiscorp.com/SQLStandards.html>.

История совершенствования SQL отчасти подтверждает один из неписаных законов программирования: лучшее – враг хорошего. С каждым очередным витком развития стандарта все меньше и меньше производителей программного обеспечения могут его поддерживать в полном объёме. У признанного гуру в области реляционной модели данных Криса Дейта на этот счёт есть хорошее высказывание: «...в наши дни ни один программный продукт не поддерживает полностью даже SQL:92; вместо этого такие продукты, как правило, поддерживают то, что можно было бы назвать “надмножеством подмножества” стандарта...» [3]. Как следствие стандарт не поспевает за производителями, а это неминуемо ведёт к появлению различных ветвей языка, что с каждым годом всё более и более минимизирует вероятность появления новой, общепринятой редакции SQL, однозначно поддерживаемой всеми разработчиками ПО.

 В официальной документации, поставляемой вместе с программным обеспечением InterBase [1], утверждается, что InterBase практически полностью поддерживает стандарт SQL:1992.

Возможности SQL

Если читатель только начинает знакомиться с SQL, то необходимо сразу заметить, что это весьма мощный, но далеко не всемогущий язык. В сферу интересов SQL не попали задачи, стоящие перед прикладным и тем более системным программистом. Нет ни реализации низкоуровневых операций ввода-вывода, ни вопросов построения пользовательского интерфейса, ни организации работы с периферийными устройствами и т. п. Одним словом, на SQL не напишешь ни одного, даже самого элементарного приложения для Windows, OS X, Linux или для любой другой ОС.

Язык SQL выступает неотъемлемой частью СУБД (в нашем случае InterBase или Firebird) и применяется только в интересах обработки данных (рис. 1.1):

- *Определение данных.* Реализуется средствами подязыка определения данными (DDL, Data Definition Language). Язык нацелен на решение вопросов

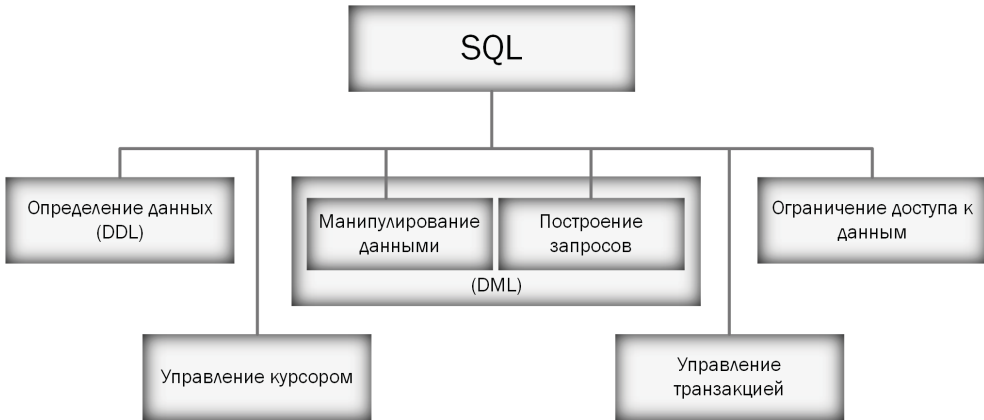


Рис. 1.1 ❖ Основные задачи языка SQL

создания и удаления базы данных и её объектов. Перечень объектов БД достаточно велик, это таблицы, представления, индексы, курсоры, определения доменов. Визитной карточкой DDL выступают операторы `CREATE`, `ALTER` и `DROP`.

- Подъязык манипулирования данными (DML, Data Manipulation Language) обеспечивает проведение операций вставки, редактирования и удаления данных из таблиц БД.
 - *Манипулирование данными.* Для модификации данных в распоряжение DML предоставлены три команды: `INSERT`, `UPDATE` и `DELETE`.
 - *Построение запросов.* Вторая и наиболее востребованная часть DML, основанная на инструкции `SELECT`, позволяет извлекать данные из одной или нескольких таблиц.
- *Ограничение доступа к данным.* Определяет ограничения на права пользователей при работе с объектами БД. В основу подъязыка положены две команды – `GRANT` и `REVOKE`.
- *Управление курсором.* Подъязык позволяет обрабатывать данные построчно. Он опирается на квартет команд: `DECLARE CURSOR`, `OPEN CURSOR`, `FETCH CURSOR`, `CLOSE CURSOR`.
- *Управление транзакцией.* Включает инструкции `SET TRANSACTION`, `BEGIN TRANSACTION`, `COMMIT` и `ROLLBACK`. Язык позволяет определять уровень изоляции транзакции, стартовать, фиксировать или возвращать транзакцию в исходное состояние.

В последующих главах книги мы узнаем, каким образом с помощью InterBase SQL решается большинство из перечисленных выше задач.

InterBase

Триумфальное шествие реляционной модели инициировало многочисленные разработки систем управления баз данных, исповедующих основные принципы

этой модели. Ряд компаний и отдельных профессиональных программистов на рубеже 1970–1980-х гг. активно включились в «гонку вооружений». В их числе оказался и Джим Старки (Jim Starkey), сотрудник на сегодняшний день уже не существующей компании Digital Equipment Corporation (DEC). Работая в DEC над проектом СУБД, основанной на сетевой модели данных, Джим параллельно вёл самостоятельную разработку своей собственной, на этот раз реляционной системы управления БД. История умалчивает о дальнейшей судьбе сетевой СУБД, создаваемой компанией DEC, однако доподлинно известно, что в 1984-м Джим Старки выпустил в свет свою собственную реляционную СУБД, дав ей тенденциозное название Jim's relational database (JRD).

Название JDS продержалось недолго, буквально через пару лет – в 1986 году – доработанный программный продукт Старки получил хорошо знакомое нам название InterBase 2.

После ряда перепродаж прав на программный продукт в 1991 году исходный код InterBase попал в руки Borland, и очередные 10 лет СУБД находилась в исключительном ведении этой весьма успешной в 1990-е гг. компании. Именно благодаря Borland базы данных InterBase стали известны широкой аудитории программистов. Это событие произошло в 1998 г., когда руководство компании приняло историческое решение включить InterBase 5.1 в программный пакет Delphi 4. Программисты Delphi и C++ Builder сразу оценили всё совершенство InterBase и стали активно применять эту СУБД в своих проектах.

Следующее ключевое событие в судьбе InterBase произошло в 2000 г. К этому времени по ряду причин Borland оказалась на перепутье. Из стен компании уволилось несколько ключевых разработчиков InterBase, что привело к тому, что судьба системы повисла на волоске. Всё это привело к беспрецедентному для коммерческой компании решению – Borland в буквальном смысле подарила мировому сообществу СУБД, выпустив очередной релиз сервера InterBase 6.0 в открытых кодах!

Появление открытого кода InterBase вдохновило ряд независимых разработчиков, которые немедленно основали проект Firebird – бесплатной СУБД, базирующейся на открытом коде InterBase 6.0.



Даже сегодня благодаря общему коду InterBase и Firebird очень близки. Конечно, нельзя говорить о полной совместимости, ведь проекты уже 15 лет развиваются самостоятельно, но, изучив основы InterBase, вы при необходимости легко переквалифицируетесь на родственную платформу Firebird, и наоборот.

Начиная с 2008 г. права на InterBase, как, впрочем, и на Delphi, принадлежат компании Embarcadero, которая дала новый импульс развитию этой программной системе. На момент написания данных строк InterBase XE3 входит в состав дистрибутива Embarcadero RAD Studio XE7, в который, кроме InterBase, традиционно входят Delphi и C++ Builder.

Реализация SQL в InterBase

Язык SQL в InterBase почти полностью соответствует стандарту SQL:92 и достаточно близок к стандартам SQL:99 и SQL:2003. Вместе с тем создатели InterBase

не рассматривают стандарт как догму и последовательно совершенствуют своё детище. Поэтому в InterBase внедрены три развитых диалекта SQL (рис. 1.2), способных решать весь диапазон задач, стоящих перед языком реляционных БД.

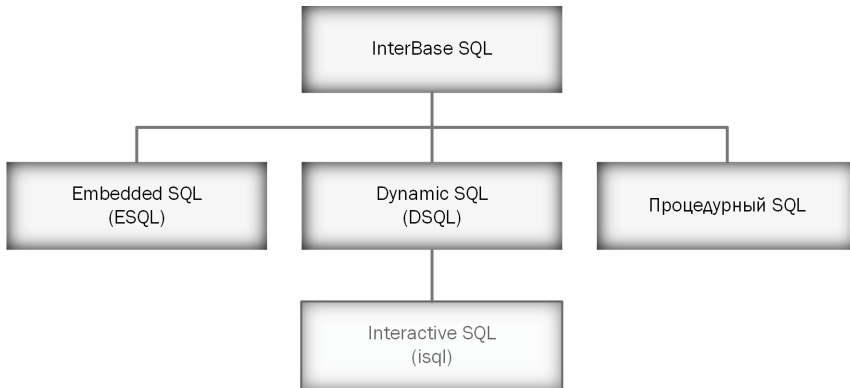


Рис. 1.2 ❖ Реализация языка SQL в InterBase

В состав InterBase SQL включены:

- встраиваемый SQL (Embedded SQL, ESQL) – предназначен для использования в клиентских приложениях БД, написанных на языках классического программирования 3-го поколения, таких как C и Pascal. Фактически ESQL представляет собой сравнительно небольшое подмножество операторов, подобных операторам SQL, и ряд специфичных конструкций. Так как препроцессор включает текст ESQL в исполняемый код приложений, в результате мы получаем исключительно статические запросы, что не позволяет пользователю изменять правила выборки данных во время выполнения приложения. Ещё одним недостатком встраиваемого SQL является то, что его операторы нельзя применять в хранимых процедурах и триггерах;
- динамический SQL (Dynamic SQL, DSQL) – наиболее востребованная реализация SQL, позволяющая создавать запросы во время выполнения программы. Взаимодействие между приложением и сервером осуществляется за счёт обращения к функциям встроеного InterBase API. Если вы разрабатываете клиентские проекты на высокоуровневых языках из состава Embarcadero RAD Studio (например, Delphi и C++ Builder), то о прямом обращении к API можно не задумываться, так как все функции инкапсулированы в многочисленные классы и компоненты библиотек VCL и FM;
- процедурный SQL – вносит в изначально декларативный SQL разнообразные процедурные расширения, такие как переменные, условные операторы, циклы и т. п. Перечисленные возможности применяются в триггерах и хранимых процедурах InterBase. Процедурный SQL поддерживает DML и DQL, но не в состоянии работать с большинством инструкций определения данных DDL;

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru