

Ускоренный курс SQL

Эта небольшая глава предназначена для быстрого ознакомления с базовой терминологией и концепциями SQL.

Что такое база данных

Начнем с основ. *База данных (БД)* — это место для организованного хранения данных. Есть множество способов организации данных и, как следствие, большое количество баз, из которых можно выбирать. Базы данных делятся на две категории: *SQL* и *NoSQL*.

SQL

SQL — это сокращение от *Structured Query Language* (язык структурированных запросов). Представьте, что у вас есть приложение, которое помнит дни рождения ваших друзей и близких. SQL — самый популярный язык, с помощью которого вы можете взаимодействовать с этим приложением.

Русский язык: Привет, приложение. Когда у моего мужа день рождения?

```
SQL: SELECT * FROM birthdays  
WHERE person = 'husband';
```

Базы данных SQL часто называют *реляционными*, поскольку они состоят из отношений, которые чаще всего называют таблицами. Множество таблиц, связанных между собой, составляют БД. На рис. 1.1 изображена таблица в базе данных SQL.

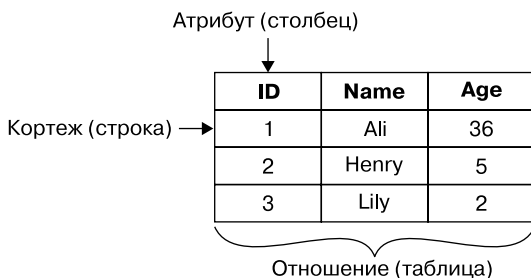


Рис. 1.1. Таблица в базе данных SQL

Главное, что следует отметить в отношении баз данных SQL, — это то, что они требуют наличия predetermined *схем*. Схему можно рассматривать как способ организации или структурирования данных в базе. Допустим, вы хотите создать таблицу. Прежде чем загружать в нее какие-либо данные, необходимо сначала определиться с ее структурой, в том числе с тем, какие столбцы будут в ней, будут ли они содержать целочисленные или десятичные значения и т. д.

Однако наступает момент, когда данные невозможно организовать таким структурированным образом. Ваши данные могут иметь различные поля, или вам может потребоваться более эффективный способ хранения большого объема данных и доступа к ним. Вот здесь и приходит на помощь NoSQL.

NoSQL

NoSQL означает *не только SQL* (not only SQL). В нашей книге эта категория баз подробно не рассматривается, но я хотела обратить на нее внимание, поскольку с 2010-х этот термин стал весьма популярным и важно понимать, что существуют способы хранения данных, выходящие за рамки таблиц.

Базы данных NoSQL часто называются *нереляционными* и бывают разных форм и размеров. В числе их основных характеристик — наличие динамических схем (то есть схема не должна быть зафиксирована заранее) и возможность горизонтального масштабирования (то есть данные могут быть распределены по нескольким машинам).

Наиболее популярной базой данных NoSQL является *MongoDB*, которая в большей степени представляет собой БД документов. На рис. 1.2 показано, как хранятся данные в MongoDB. Вы можете заметить, что данные больше не находятся в структурированной таблице, а количество полей (аналогично столбцам) варьируется для каждого документа (аналогично строкам).

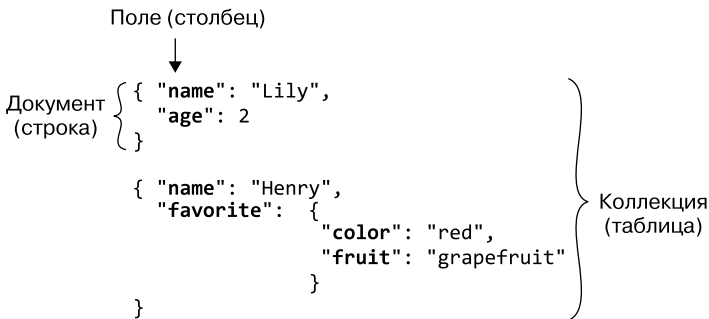


Рис. 1.2. Коллекция (вариант таблицы) в MongoDB, базе данных NoSQL

Тем не менее основное внимание в этой книге уделяется базам данных SQL. Даже после появления NoSQL очень многие компании по-прежнему хранят большую часть своих данных в таблицах реляционных БД.

Системы управления базами данных (СУБД)

Возможно, вы слышали такие термины, как PostgreSQL или SQLite, и задавались вопросом, чем они отличаются от SQL. Это два типа *систем управления базами данных (СУБД)*, то есть программного обеспечения, используемого для работы с базой.

Программное обеспечение СУБД включает в себя такие функции, как определение способов импорта и организации данных, а также управление доступом пользователей и других программ к данным. *Система управления реляционными базами данных (РСУБД)* — программное обеспечение, специально предназначенное для реляционных БД, или баз, состоящих из таблиц.

Каждая РСУБД имеет свою реализацию SQL, а это означает, что синтаксис немного различается в зависимости от программного обеспечения. Например, вот как можно вывести десять строк данных в пяти различных системах:

MySQL, PostgreSQL и SQLite

```
SELECT * FROM birthdays LIMIT 10;
```

Microsoft SQL Server

```
SELECT TOP 10 * FROM birthdays;
```

Oracle Database

```
SELECT * FROM birthdays WHERE ROWNUM <= 10;
```

ПОИСК СИНТАКСИСА SQL В GOOGLE

При поиске синтаксиса SQL в Интернете всегда указывайте в поиске РСУБД, с которой работаете. На заре моего изучения SQL я не могла понять, почему скопированный мною из Интернета код не работает, а причина была именно в этом!

Сделайте так.

Поиск: *создать таблицу datetime postgresql*

→ Результат: timestamp

Поиск: *создать таблицу datetime microsoft sql server*

→ Результат: datetime

А не так.

Поиск: *создать таблицу datetime*

→ Результат: синтаксис может быть для любой РСУБД

В этой книге рассматриваются основы SQL, а также нюансы пяти популярных систем управления базами данных: Microsoft SQL Server, MySQL, Oracle Database, PostgreSQL и SQLite.

Одни из этих систем являются проприетарными, то есть принадлежат компании, и их использование требует денег, а другие имеют открытый исходный код, то есть любой желающий может использовать их бесплатно. В табл. 1.1 подробно описаны различия между РСУБД.

Таблица 1.1. Таблица сравнения РСУБД

РСУБД	Владелец	Основные сведения
Microsoft SQL Server	Microsoft	<ul style="list-style-type: none"> • Популярная проприетарная РСУБД. • Часто используется вместе с другими продуктами Microsoft, в том числе с Microsoft Azure и фреймворком .NET. • Наиболее часто используется на платформе Windows. • Иное название — MSSQL или SQL Server
MySQL	Открытый исходный код	<ul style="list-style-type: none"> • Популярная РСУБД с открытым исходным кодом. • Часто используется вместе с такими языками веб-разработки, как HTML/CSS/Javascript. • Приобретена компанией Oracle, хотя по-прежнему имеет открытый исходный код
Oracle Database	Oracle	<ul style="list-style-type: none"> • Популярная проприетарная РСУБД. • Часто используется в крупных корпорациях благодаря большому количеству доступных функций, инструментов и поддержки. • Иное название — просто Oracle
PostgreSQL	Открытый исходный код	<ul style="list-style-type: none"> • Популярность быстро растет. • Часто используется вместе с технологиями с открытым исходным кодом, такими как Docker и Kubernetes. • Эффективна и отлично подходит для больших наборов данных

Таблица 1.1 (продолжение)

РСУБД	Владелец	Основные сведения
SQLite	Открытый исходный код	<ul style="list-style-type: none">• Самый используемый в мире движок баз данных.• Распространен на платформах iOS и Android.• Легкий и отлично подходит для небольшой БД



Далее в книге:

- Microsoft SQL Server будет называться SQL Server;
- Oracle Database будет называться Oracle.

Инструкции по установке и фрагменты кода для каждой системы приведены в разделе «Программное обеспечение РСУБД» главы 2.

SQL-запрос

В мире SQL часто встречается аббревиатура *CRUD*, которая расшифровывается как Create, Read, Update, Delete (создание, чтение, обновление и удаление). Это четыре основные операции, которые доступны в базе данных.

SQL-операторы

Люди, имеющие *доступ для чтения и записи*, могут выполнять с базой все четыре операции: создавать и удалять таблицы, обновлять в них данные и читать данные из таблиц. Другими словами, эти люди обладают всей полнотой власти.

Они пишут *SQL-операторы* — общий SQL-код, который можно написать для выполнения любой из операций CRUD. Эти люди часто занимают такие должности, как *администратор баз данных* (database administrator, DBA) или *инженер баз данных*.

SQL-запросы

Люди, имеющие *доступ для чтения*, могут выполнять в БД только операцию чтения, то есть могут просматривать данные в таблицах.

Эти люди пишут *SQL-запросы*, которые представляют собой более конкретный тип SQL-операторов. Запросы используются для поиска и отображения данных, иначе называемых *чтением* данных. Иногда это действие называют *запросом к таблицам*. Эти люди часто занимают такие должности, как *аналитик данных* или *специалист по анализу данных*.

Следующие два подраздела представляют собой краткое руководство по написанию SQL-запросов, поскольку это наиболее часто встречающийся тип SQL-кода. Более подробная информация о создании и обновлении таблиц содержится в главе 5.

Оператор SELECT

Самый простой SQL-запрос (который будет работать в любом программном обеспечении SQL) имеет вид:

```
SELECT * FROM my_table;
```

В нем говорится: покажите мне все данные в таблице с именем `my_table` — все столбцы и все строки.

Хотя SQL не чувствителен к регистру (`SELECT` и `select` эквивалентны), можно заметить, что одни слова написаны заглавными буквами, а другие — нет.

- Слова в запросе, записанные заглавными буквами, называются *ключевыми*. Это означает, что SQL зарезервировал их для выполнения той или иной операции над данными.
- Все остальные слова пишутся строчными буквами. Это относится к именам таблиц, столбцов и т. д.

Формат заглавных и строчных букв не является обязательным, но это хорошее условное обозначение, которое следует соблюдать ради удобства чтения. Вернемся к уже знакомому нам запросу:

```
SELECT * FROM my_table;
```

Допустим, вместо того, чтобы возвращать все данные в их текущем состоянии, я хочу их:

- фильтровать;
- сортировать.

Для этого необходимо изменить оператор `SELECT`, добавив в него несколько *предложений (clauses)*, и результат будет выглядеть примерно так:

```
SELECT *  
FROM my_table  
WHERE column1 > 100  
ORDER BY column2;
```

Более подробную информацию обо всех предложениях можно найти в главе 4, но главное, что следует отметить: предложения всегда должны быть перечислены в одном и том же порядке.

ЗАПОМНИТЕ ЭТОТ ПОРЯДОК

Все SQL-запросы будут содержать некую комбинацию этих предложений. Обязательно запомните этот порядок!

SELECT	-- столбцы для отображения
FROM	-- таблица (-ы), из которой (-ых) нужно извлечь данные
WHERE	-- фильтрация строк
GROUP BY	-- разбиение строк на группы
HAVING	-- фильтрация сгруппированных строк
ORDER BY	-- столбцы для сортировки



Знак `--` представляет собой начало комментария в SQL, то есть текст после него предназначен только для документации и код выполняться не будет.

По большей части, из всех этих предложений обязательные лишь `SELECT` и `FROM`. Исключением является случай, когда вы выбираете определенную функцию базы данных, — здесь вам требуется только `SELECT`.

Классическим мнемоническим выражением для запоминания порядка расположения предложений является:

Sweaty feet will give horrible odors.

Если не хотите думать о потных ногах каждый раз, когда пишете запрос, то вот один из вариантов, который я придумала:

Start Fridays with grandma's homemade oatmeal.

Порядок выполнения

Порядок выполнения SQL-кода обычно не изучается в курсах SQL для начинающих, но я добавила данную тему в книгу, поскольку часто получала этот вопрос, когда преподавала SQL студентам, имеющим опыт программирования на Python.

Разумным было бы предположить, что порядок *написания* предложений соответствует порядку их *выполнения* компьютером, но это не так. После выполнения запроса компьютер обрабатывает данные в следующем порядке:

- 1) `FROM`;
- 2) `WHERE`;
- 3) `GROUP BY`;
- 4) `HAVING`;
- 5) `SELECT`;
- 6) `ORDER BY`.

Если сравнивать с порядком, в котором вы фактически записываете предложения, то можно заметить, что оператор `SELECT` переместился на пятую позицию. Главный вывод, который можно сделать, заключается в том, что `SQL` работает в следующем порядке:

- 1) собирает все данные с помощью `FROM`;
- 2) фильтрует строки данных, используя `WHERE`;
- 3) группирует строки с помощью `GROUP BY`;
- 4) фильтрует сгруппированные строки, используя `HAVING`;
- 5) указывает столбцы для отображения с помощью `SELECT`;
- 6) переупорядочивает результаты, используя `ORDER BY`.

Модель данных

В заключительном разделе ускоренного курса я хотела бы рассмотреть простую *модель данных* и обратить ваше внимание на некоторые термины, которые вы часто будете слышать, обсуждая `SQL` в офисе.

Модель данных — это визуализация, которая обобщает информацию о том, как все таблицы в БД связаны друг с другом, а также некоторые сведения о каждой из них. На рис. 1.3 представлена простая модель данных БД об оценках учащихся.

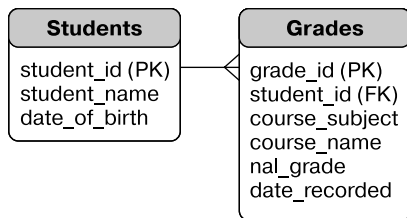


Рис. 1.3. Модель данных об оценках учащихся

В табл. 1.2 перечислены технические термины, описывающие происходящее в модели данных.

Таблица 1.2. Термины, используемые для описания того, что происходит в модели данных

Термин	Определение	Пример
База данных	База данных — место для организованного хранения данных	Эта модель данных отображает все данные в БД об оценках учащихся
Таблица	Таблица состоит из строк и столбцов. В модели данных они представлены прямоугольниками	В базе данных об оценках учащихся имеется две таблицы: <code>Students</code> и <code>Grades</code>
Колонка	Таблица состоит из множества столбцов, которые иногда называют атрибутами или полями. Каждый столбец содержит определенный тип данных. В модели данных все столбцы таблицы перечислены в каждом прямоугольнике	В таблице <code>Students</code> столбцами является <code>student_id</code> , <code>student_name</code> и <code>date_of_birth</code>
Первичный ключ	<i>Первичный ключ</i> однозначно идентифицирует каждую строку данных в таблице. Может состоять из одного или нескольких столбцов таблицы. В модели данных обозначается как <code>pk</code> (от <code>primary key</code>) или значком ключа	В таблице <code>Students</code> в поле первичным ключом является столбец <code>student_id</code> . Это говорит о том, что значение <code>student_id</code> для каждой строки данных различно
Внешний ключ	<i>Внешний ключ</i> в таблице ссылается на первичный ключ в другой таблице. Две таблицы могут быть связаны между собой общим столбцом. Таблица может иметь несколько внешних ключей. В модели данных внешний ключ обозначается как <code>fk</code> (от <code>foreign key</code>)	В таблице <code>Grades</code> столбец <code>student_id</code> является внешним ключом, то есть значения в этом столбце совпадают со значениями в соответствующем столбце первичного ключа в таблице <code>Students</code>

Продолжение ↗

Таблица 1.2 (продолжение)

Термин	Определение	Пример
Отношения	<i>Отношение</i> описывает, как строки одной таблицы соотносятся со строками другой. В модели данных представлено в виде линии с символами в конечных точках. Распространенными типами являются отношения «один к одному» и «один ко многим»	В этой модели данных две таблицы имеют отношение «один ко многим», представленное разветвлением. Один студент может иметь много оценок, или одна строка таблицы Students сопоставляется с несколькими строками таблицы Grades

Более подробную информацию об этих терминах можно найти в разделе «Создание таблиц» главы 5.

Возможно, вам интересно, почему мы тратим столько времени на чтение модели данных вместо того, чтобы уже писать код SQL! Причина в том, что часто приходится писать запросы, связывающие несколько таблиц, поэтому рекомендуется сначала ознакомиться с моделью данных, чтобы знать, как эти таблицы связаны между собой.

Модели данных обычно можно найти в репозитории документации компании. Возможно, вы захотите распечатать модели данных, с которыми часто работаете, — это может облегчить удобство их использования.

Вы также можете писать запросы в РСУБД для поиска информации, содержащейся в модели данных, например, таблиц в БД, столбцов или ограничений таблицы.

ВОТ И ВЕСЬ УСКОРЕННЫЙ КУРС!

Оставшаяся часть книги предназначена для использования в качестве справочного материала, и ее необязательно читать по порядку. Вы можете искать в ней понятия, ключевые слова и стандарты.