Содержание

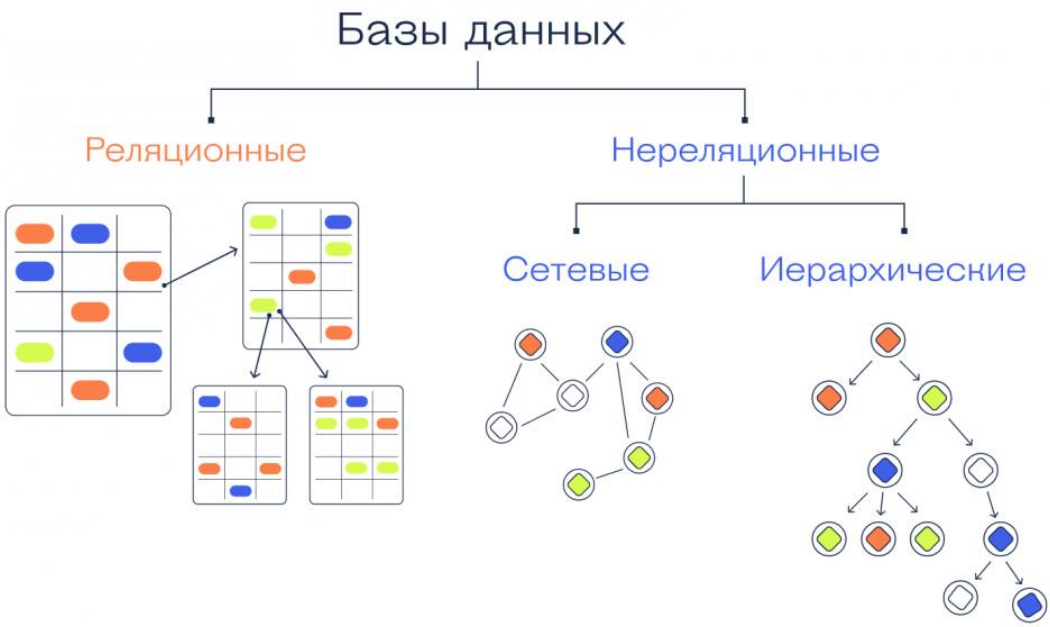
Пункт 1. Типы баз данных 1
ПРОВЕРОЧНЫЙ ТЕСТ 1 3
Пункт 2. Простые операторы языка SQL 4
ПРОВЕРОЧНЫЙ ТЕСТ 2 5
Пункт 3. Агрегатные функции SQL 6
ПРОВЕРОЧНЫЙ ТЕСТ 3 8

SQL — это не совсем язык программирования. Правильно сказать, что это язык запросов к базе данных. Название так и расшифровывается: «язык структурированных запросов» (Structured Query Language). Это значит, что каждый запрос к базе данных формируется по какой-то структуре, а сам язык задаёт правила, как именно сделать такой запрос. Это способ получить данные из базы, а потом обработать их с помощью более гибких и удобных языков: Python, JavaScript, C++ или другого подобного языка.

Пункт 1. Типы баз данных

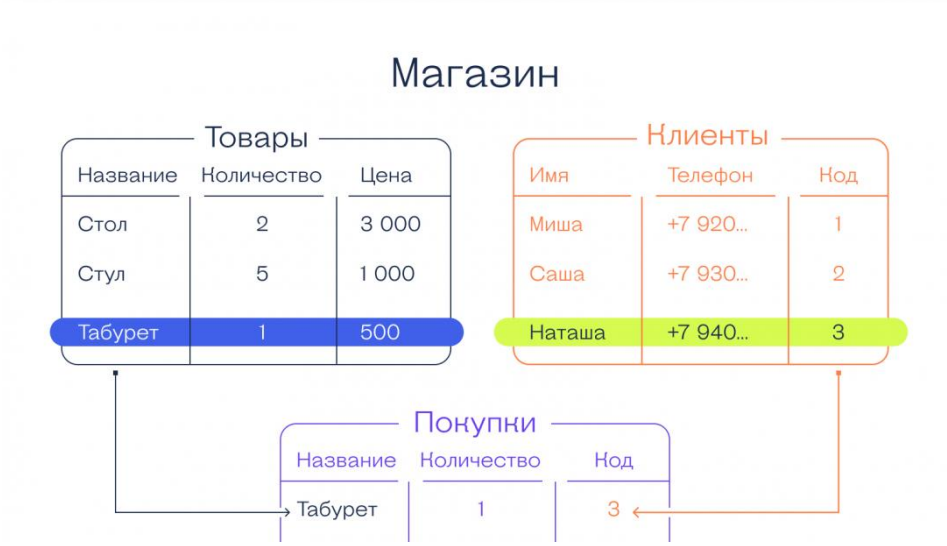
Три основных типа

В зависимости от того, какие данные нужно в ней хранить и как с ними работать, базы делятся на реляционные и нереляционные:



Реляционные

Реляционные базы данных ещё называют табличными, потому что все данные в них можно представить в виде разных таблиц. Одни таблицы связаны с другими, а другие — с третьими. Например, база данных покупок в магазине может выглядеть так:



У магазина есть две таблицы — с товарами и покупателями. Но когда один из них что-то покупает, то данные попадают в третью таблицу. В ней есть своя информация (количество купленных товаров) и ссылки на покупателя и сам товар. Если нужно, можно по этим связям попасть в нужную таблицу и узнать подробности о той или другой записи.

Если у покупателя поменяется номер телефона, то нам достаточно будет поменять это в одной таблице «Клиенты». Благодаря тому, что в «Покупки» записывается только код покупателя, нам не нужно менять имя больше нигде — данные сами обновятся автоматически, когда мы захотим посмотреть, кто именно купил табурет.

Сетевые

В отличие от реляционных баз, в сетевых между таблицами и записями может быть несколько разных связей, каждая из который отвечает за что-то своё.

Если мы возьмём базу данных с сайта Кинопоиска, то она может выглядеть так:

Особенность сетевой базы данных в том, что в ней запоминаются все связи и всё содержимое для каждой связи. Базе не нужно тратить время на поиск нужных данных, потому что вся информация об этом уже есть в специальных индексных файлах. Они показывают, какая запись с какой связана, и быстро выдают результат.

ПРОВЕРОЧНЫЙ ТЕСТ 1

пригласить через игровой код

1.Попросите участников открыть

joinmyquiz.com

2.И введите этот код

0853 1851

Пункт 2. Простые операторы языка SQL

Одно из самых главных слов в SQL — это `SELECT`. С его помощью можно вывести на экран всё, что душе угодно.
Немного потренируйтесь и отобразите на экране строку `'Мы начинаем SQL'`.

SELECT 'Мы начинаем SQL'; - ничего не меняя в коде, нажмите «Проверить»

```
SELECT 'Hello, World!';
SELECT 5;
```

SELECT 'Я начинаю изучать SQL';

SQL вместо калькулятора

SELECT 3+5; -- замените на любое арифметическое выражение

Отобразите на экране сумму чисел 35 и 28. Сделайте так, чтобы программа сама посчитала сумму.SELECT 35+28;

Работаем с таблицами

```
 выгрузим данные из таблицы actor — в ней собраны имена и фамилии актёров и
актрис. Получится такой запрос:
SELECT * FROM actor;
```

Выборка данных из таблицы

Как отобрать данные из определенных столбцов? Например нам нужно получить только имена пользователей и из какого они города. Сначала пишем оператор **SELECT**, затем перечисляем нужные нам столбцы из таблицы *Person*.

```
SELECT CustomerName, City FROM Person;
```

Упорядочить ORDER BY

Сейчас пользователи расположены в таблице по номеру **ID**, мы хотим упорядочить их по стране проживания. Как вы наверное уже заметили, что язык **SQL** очень лаконичный и логичный. Выбрать **SELECT** все столбцы «*» из таблицы **Person** и упорядочить их по странам.

```
 SELECT * FROM Person
ORDER BY Country;
```

Оператор WHERE

Мы хотим отобрать пользователей только из Швеции. Выбираем все столбцы из таблицы **Person**, где **WHERE** страна проживания Швеция.

```
 SELECT * FROM Person
WHERE Country='Sweden';
```

ПРОВЕРОЧНЫЙ ТЕСТ 2

пригласить через игровой код

1.Попросите участников открыть

joinmyquiz.com

2.И введите этот код

2550 5675

Пункт 3. Агрегатные функции SQL

Когда нам нужно получить сумму, максимальное, минимальное или среднее значение хранящихся в базе данных чисел, то мы используем агрегатные функции. Большинство систем управления базами данных поддерживают 5 агрегатных (математических) функций.

SUM рассчитывает сумму

Вычислим сумму полей «Quantity» в таблице «OrderDetails».

Quantity
12
10
5
9
40

Пишем слово SELECT, а затем название агрегатной функции SUM(). В скобках указываем, значения какого столбца мы хотим суммировать. Дальше пишем оператор FROM и имя таблицы.

```
SELECT SUM(Quantity) FROM OrderDetails;
```

MAX рассчитывает максимальное значение

Найдем с помощью оператора MAX() самую большую стоимость продукта.

ProductName	SupplierID	CategoryID	Unit	Price
Chais	1	1	10 boxes x 20 bags	18
Chang	1	1	24 - 12 oz bottles	19
Aniseed Syrup	1	2	12 - 550 ml bottles	10
Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
Chef Anton's Gumbo Mix	2	2	36 boxes	21.35

```
SELECT MAX(Price) FROM Products;
```

MIN рассчитывает минимальное значение

Найдем с помощью оператора MIN() самую маленькую стоимость продукта.

```
 SELECT MIN(Price)
FROM Products;
```

Функция MIN нашла цену самого дешевого продукта в столбце Price из таблицы Products.

COUNT подсчитывает количество значений

Функция COUNT() возвращает количество строк, которое соответствует заданному критерию. Давайте выведем общее количество товаров, хранящихся в базе данных. Проще всего подсчитать количество продуктов по ID.

```
 SELECT COUNT(ProductID) FROM Products;
```

AVG рассчитывает среднее значение

Функция AVG() возвращает среднее значение чисел в указанном столбце.

```
SELECT AVG(Price)

FROM Products;
```

Группировка данных Group By

Ключевое слово Group By чаще всего используется вместе с агрегатными функциями, чтобы группировать столбцы по тому или иному признаку. Например, нам нужно подсчитать количество клиентов в базе из каждой страны по отдельности.

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Пишем оператор SELECT и рассчитываем количество клиентов по столбцу CustomerID для каждой страны и группируем результат расчета по странам. Например из Аргентины у нас 3 клиента, а из Бельгии - 2 клиента.

```
 SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country;
```

Упорядочим наш список и отсортируем клиентов в порядке убывания. Нам нужно знать, в каких странах живет всех больше клиентов.

```
 SELECT COUNT(CustomerID), Country
FROM Customers
GROUP BY Country
ORDER BY COUNT(CustomerID) DESC;
```

Всех больше клиентов проживает в USA.

ПРОВЕРОЧНЫЙ ТЕСТ 3

пригласить через игровой код

1.Попросите участников открыть

joinmyquiz.com

2.И введите этот код

1121 8827