



SQL

27.04.2024



ПЛАН

- СУБД
- SELECT, FROM, LIMIT
- WHERE, ORDER BY
- GROUP BY, HAVING
- JOIN
- Подзапросы



СУБД, SQL



СУБД, SQL

База данных — данные



СУБД, SQL

База данных — данные

СУБД — ПО для работы с данными

SQL — (Structured Query Language) язык запросов для взаимодействия с реляционными базами данных



Типы СУБД

SQL vs NoSQL

Реляционные базы данных – базы данных, записи в которых хранятся в виде набора таблиц и заранее определенных связей между ними.

Нереляционные базы данных – базы данных, которые не используют традиционную реляционную модель.



Типы СУБД

SQL vs NoSQL

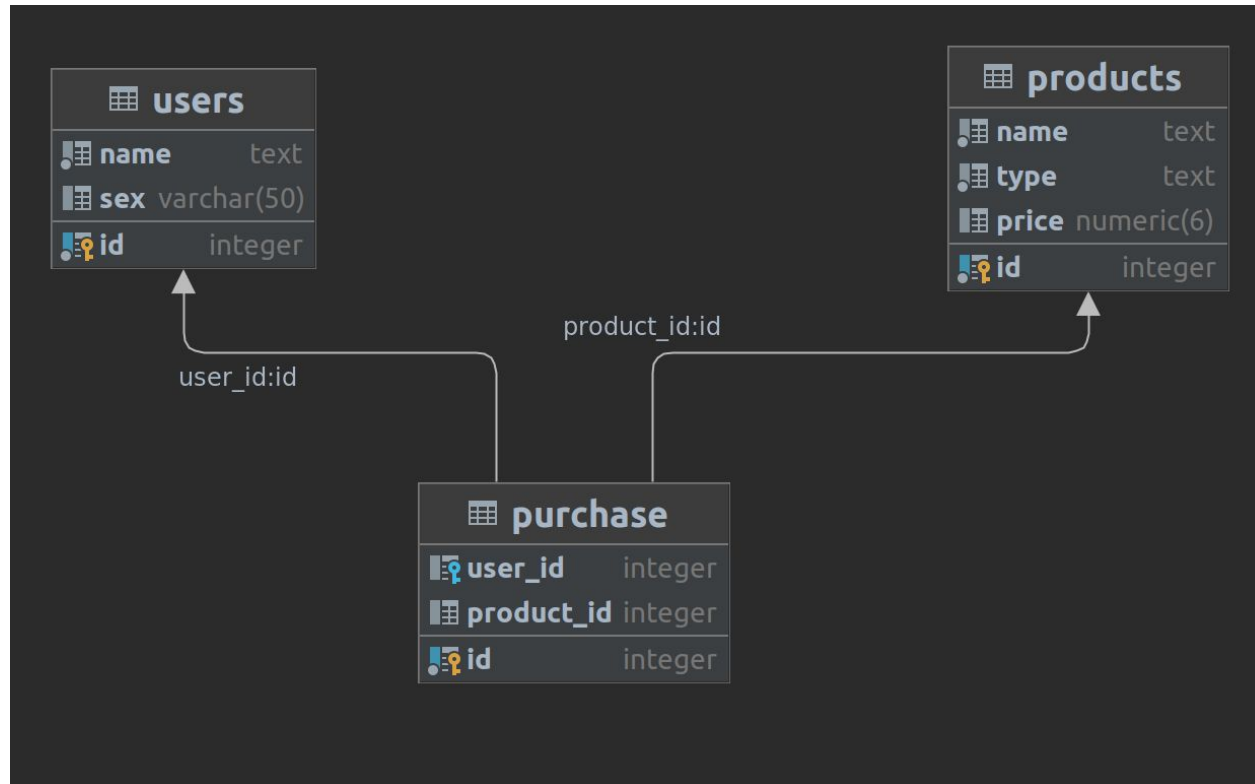
Реляционные базы данных: PostgreSQL, MySQL ...

Нереляционные базы данных:

- **key-value** (Redis) – записи по ключу
- **Документные БД** (MongoDB) – JSON
- **Колоночные БД** (Cassandra, ClickHouse) – набора колонок, не связанных между собой
- **Графовые БД** (Neo4j)
- ...



Таблички и СВЯЗИ





SQL

Типы запросов

- **DDL** (Data Definition Language) – CREATE, ALTER, DROP, TRUNCATE
- **DML** (Data Manipulation Language) – SELECT, INSERT, UPDATE, DELETE
- **DCL** (Data Control Language) – GRANT, REVOKE
- **TCL** (Transaction Control Language) – COMMIT, ROLLBACK, SAVEPOINT

SQL



SELECT -
FROM -
JOIN -
WHERE -
GROUP BY -
HAVING -
ORDER BY -
LIMIT -



SQL

SELECT - выборка данных из таблицы или вычисление выражений

FROM - указание таблицы или представления, из которой выбираются данные

JOIN - объединение данных из таблиц на основе определенного условия

WHERE - фильтрация данных на основе определенного условия

GROUP BY - группировка данных по определенному столбцу или выражению

HAVING - фильтрация данных после группировки

ORDER BY - сортировка результатов по определенному столбцу или выражению

LIMIT - ограничение количества возвращаемых строк



SELECT, FROM, LIMIT

```
SELECT * FROM users;
```

```
SELECT *  
FROM users;
```

Получить всех колонки (**SELECT ***)
из таблицы users (**FROM users**)



SELECT, FROM, LIMIT

```
SELECT * FROM users;
```

```
SELECT *  
FROM users;
```

```
SELECT  
    name,  
    sex  
FROM users;
```

Получить всех колонки (**SELECT ***)
из таблицы users (**FROM users**)

Получить колонки name и sex
(**SELECT name, sex**)
из таблицы users (**FROM users**)



SELECT, FROM, LIMIT

```
SELECT * FROM users;
```

```
SELECT *  
FROM users;
```

```
SELECT  
    name,  
    sex  
FROM users;
```

```
SELECT  
    name AS user_name,  
    sex  
FROM users;
```

Получить всех колонки (**SELECT ***)
из таблицы users (**FROM users**)

Получить колонки name и sex
(**SELECT name, sex**)
из таблицы users (**FROM users**)

Получить колонки name и sex, но
переназвать name в user_name
(**SELECT name AS user_name, sex**)
из таблицы users (**FROM users**)



SELECT, FROM, LIMIT

```
SELECT
    name,
    price / 1.20
FROM products;
```

Цены разделить на 1.2
(**SELECT price / 1.20**)

```
SELECT
    name,
    round(price / 1.20)
FROM products;
```

Цены разделить на 1.2 и округлить
(**SELECT round(price / 1.20)**)

```
SELECT
    name,
    round(price / 1.20) AS price_no_tax
FROM products;
```



SELECT, FROM, LIMIT

```
SELECT DISTINCT id  
FROM products;
```

Получить уникальные id
(**SELECT DISTINCT id**)

```
SELECT COUNT(DISTINCT id)  
FROM products;
```

Посчитать количество уникальных id
(**SELECT COUNT(DISTINCT id)**)

```
SELECT COUNT(*)  
FROM products;
```

Посчитать количество строк
(**SELECT COUNT(*)**)

```
SELECT COUNT(*) AS total  
FROM products;
```

Посчитать количество строк и назвать total
(**SELECT COUNT(*) AS total**)



SELECT, FROM, LIMIT

DEMO



WHERE, ORDER BY

```
SELECT *  
FROM products  
WHERE price > 2000;
```

Получить продукты, у которых цена больше 2000

(WHERE price > 2000)

```
SELECT *  
FROM products  
WHERE price > 2000  
    AND type = 'tech';
```

Получить продукты, у которых цена больше 2000 и тип – tech

(WHERE price > 2000 AND type = 'tech')



WHERE, ORDER BY

```
SELECT *  
FROM products  
ORDER BY price; -- ASC
```

Отсортировать продукты по
возрастанию цены
(**ORDER BY price ASC**)

```
SELECT *  
FROM products  
ORDER BY price DESC;
```

Отсортировать продукты по
убыванию цены
(**ORDER BY price DESC**)



WHERE, ORDER BY

DEMO



GROUP BY, HAVING

```
SELECT ????  
FROM products  
GROUP BY type;
```

```
SELECT * -- Ошибка!  
FROM products  
GROUP BY type;
```

Группируем продукты по типу
(**GROUP BY type**) НО ...



GROUP BY, HAVING

```
SELECT sum(price),  
       count(id),  
       type  
FROM products  
GROUP BY type;
```

Группируем продукты по типу (**GROUP BY type**) и считаем сумму по цене (**sum(price)**) и количество товаров (**count(id)**)



GROUP BY, HAVING

```
SELECT sum(price),  
       count(id),  
       string_agg(name, ', '),  
       type  
FROM products  
GROUP BY type;
```

```
SELECT sum(price),  
       count(id),  
       string_agg(name, ', '),  
       type  
FROM products  
WHERE price < 8000  
GROUP BY type;
```

Группируем продукты по типу (**GROUP BY type**) и считаем сумму по цене (**sum(price)**) и количество товаров (**count(id)**)

Можно также фильтровать!!!



GROUP BY, HAVING

```
SELECT sum(price),  
       count(id),  
       string_agg(name, ', '),  
       type  
FROM products  
WHERE price < 10000  
GROUP BY type  
HAVING count(id) > 1;
```

Группируем продукты по типу (**GROUP BY type**), считаем сумму по цене (**sum(price)**), количество товаров (**count(id)**) и результат фильтруем так, чтобы количество товаров было больше 1 (**HAVING count(id) > 1**)



GROUP BY, HAVING

DEMO



JOIN

Типы джойнов

- **INNER JOIN** – соответствия в обеих таблицах
- **LEFT JOIN** – все строки из левой и соответствующие из правой
- **RIGHT JOIN** – все строки из правой и соответствующие из левой
- **FULL JOIN** – все строки из левой и из правой
- **CROSS JOIN** – декартово произведение таблиц (комбинации строк)



JOIN

user_id	product_id
1	21
2	12341

product_id	name
7	Ноутбук
21	Телефон



JOIN

user_id	product_id
1	21
2	12341

INNER JOIN

product_id	name
7	Ноутбук
21	Телефон

user_id	product_id	name
1	21	Телефон



JOIN

user_id	product_id
1	21
2	12341

LEFT JOIN

product_id	name
7	Ноутбук
21	Телефон

user_id	product_id	name
1	21	Телефон
2	12341	NULL



JOIN

user_id	product_id
1	21
2	12341

RIGHT JOIN

product_id	name
7	Ноутбук
21	Телефон

user_id	product_id	name
NULL	7	Ноутбук
1	21	Телефон



JOIN

user_id	product_id
1	21
2	12341

FULL JOIN

product_id	name
7	Ноутбук
21	Телефон

user_id	product_id	name
NULL	7	Ноутбук
1	21	Телефон
2	12341	NULL



JOIN

```
SELECT *  
FROM purchase  
JOIN products -- INNER  
    ON products.id = purchase.product_id;
```

```
SELECT *  
FROM purchase AS l  
JOIN products AS r -- INNER  
    ON r.id = l.product_id;
```

Левая таблица – purchase
Правая таблица – products



JOIN

```
SELECT pu.user id,  
       pu.product id,  
       pr.name as product,  
       u.name  as user_name  
FROM purchase pu  
FULL JOIN products pr ON pr.id = pu.product_id  
FULL JOIN users u ON u.id = pu.user_id
```

JOIN



DEMO



Подзапросы

Подзапросы — запросы, которые вложены в другие запросы

```
SELECT select_list
FROM table1
WHERE columnA operator (
    SELECT columnB
    FROM table2
    WHERE condition
);
```



Подзапросы

DEMO