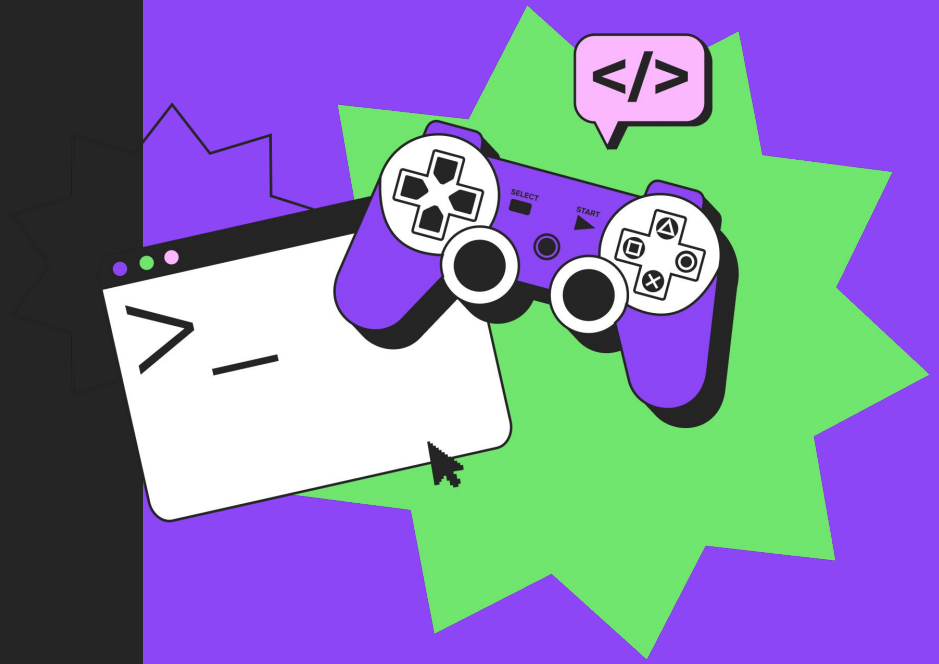
GeekBrains

Базы данных и SQL

Урок 4

SQL – объединение таблиц union, соединение - join, подзапросы



План курса

1. Инструменты для работы с базами данных
2. SQL – создание объектов, изменение данных, логические операторы
3. SQL – выборка данных, сортировка, агрегатные функции
4. SQL – объединение таблиц union, соединение - join, подзапросы
5. Будущий урок
6. Будущий урок
7. Будущий урок
8. Будущий урок
9. Будущий урок
10. Будущий урок
11. Будущий урок
12. Будущий урок

Содержание урока

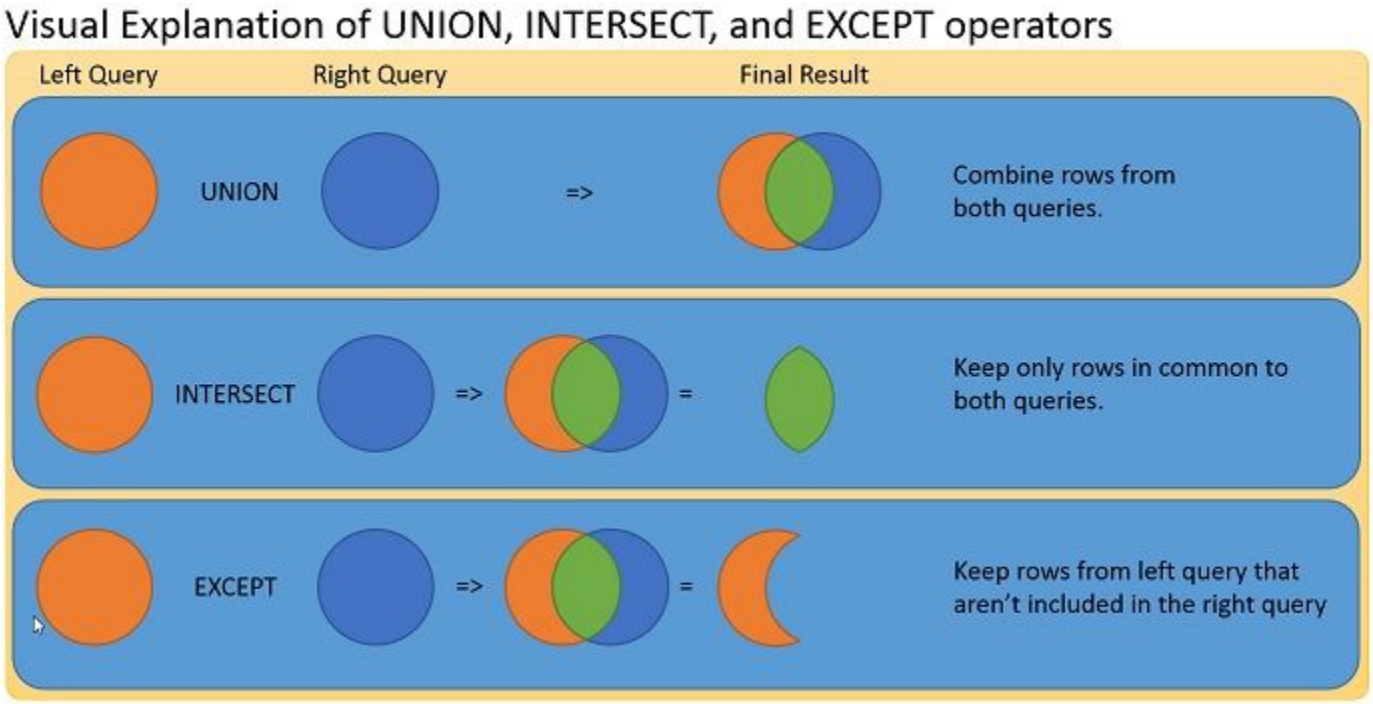
Что будет на уроке сегодня

- Операторы union и union all
- Соединение таблиц — Join (left, right, inner, full, cross)
- Использование подзапросов (join, in, exists, select)
- Использование select into, создание таблиц из результатов запроса
- Порядок выполнения запроса — дополнение схемы

Оператор UNION

```
1 SELECT ...
2 UNION [ALL | DISTINCT] SELECT ...
3 [UNION [ALL | DISTINCT] SELECT ... ]
```

Оператор UNION: общий вид



Оператор UNION: простой пример

```
37 •   SELECT 1, 2 UNION SELECT 'a', 'b';
38
39
40
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	1	2
▶	1	2
	a	b

Оператор UNION: пример

Пусть имеются две таблички: “сотрудники” и “клиенты”:

EmpID	Name	Address	Phone	Salary
1	Nikhil	Delhi	9878906543	8000
2	Divya	Ranchi	8990076543	5000
3	Ravi	Bareilly	7789945765	7000
4	Anna	Noida	9789945760	4000
5	Surbhi	Jaipur	7800541123	5500
NULL	NULL	NULL	NULL	NULL

Id	FirstName	LastName	AccountSum
1	Tom	Smith	2000
2	Sam	Brown	3000
3	Mark	Adams	2500
4	Paul	Ins	4200
5	John	Smith	2800
6	Tim	Cook	2800
7	Tom	Smith	2000
8	Sam	Brown	3000
9	Mark	Adams	2500
10	Paul	Ins	4200
11	John	Smith	2800
12	Tim	Cook	2800
13	Tom	Smith	2000
14	Sam	Brown	3000
15	Mark	Adams	2500
16	Paul	Ins	4200
17	John	Smith	2800
18	Tim	Cook	2800
NULL	NULL	NULL	NULL

Оператор UNION: пример

Выберем сразу всех клиентов банка и его сотрудников из обеих таблиц:

```
1 SELECT FirstName, LastName
2 FROM Customers
3 UNION SELECT FirstName, LastName FROM Employees;
```

Оператор UNION: пример

```
2
3  SELECT FirstName, LastName
4  FROM Customers
5  UNION SELECT FirstName, LastName FROM Employees;
```

Result Grid | Filter Rows: | Export: | Wrap Cell C

	FirstName	LastName
	Tom	Smith
	Sam	Brown
	Mark	Adams
	Paul	Ins
	John	Smith
	Tim	Cook
	Homer	Simpson
	Nick	Svensson

UNION и сортировка с помощью ORDER by

```
1 SELECT FirstName, LastName
2 FROM Customers
3 UNION SELECT FirstName, LastName
4 FROM Employees
5 ORDER BY FirstName DESC;
```

UNION и сортировка с помощью ORDER by

```sql
SELECT FirstName, LastName
FROM Customers
UNION SELECT FirstName, LastName
FROM Employees
ORDER BY FirstName DESC;
```

Result Grid | Filter Rows: | Export:

	FirstName	LastName
▶	Tom	Smith
	Tim	Cook
	Sam	Brown
	Paul	Ins
	Nick	Svensson
	Mark	Adams
	John	Smith
	Homer	Simpson

Ошибка в UNION: если в одной выборке больше столбцов, чем в другой, то они не могут быть объединены

```
1 SELECT FirstName
2 FROM Customers
3 UNION SELECT FirstName, LastName
4 FROM Employees
5 ORDER BY FirstName DESC;
```

UNION ALL:

```
1 SELECT FirstName, LastName
2 FROM Customers
3 UNION ALL SELECT FirstName, LastName
4 FROM Employees
5 ORDER BY FirstName;
```

UNION ALL:

SQL File 3*

Limit to 1000 rows

```sql
SELECT FirstName, LastName
FROM Customers
UNION ALL SELECT FirstName, LastName
FROM Employees
ORDER BY FirstName;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName
Homer	Simpson
Homer	Simpson
John	Smith
Mark	Adams
Mark	Adams
Mark	Adams
Nick	Svensson
Nick	Svensson
Paul	Ins
Sam	Brown
Tim	Cook
Tom	Smith
Tom	Smith
Tom	Smith

UNION ALL и UNION: в чем разница?

1. Оператор **UNION** удаляет повторяющиеся строки.

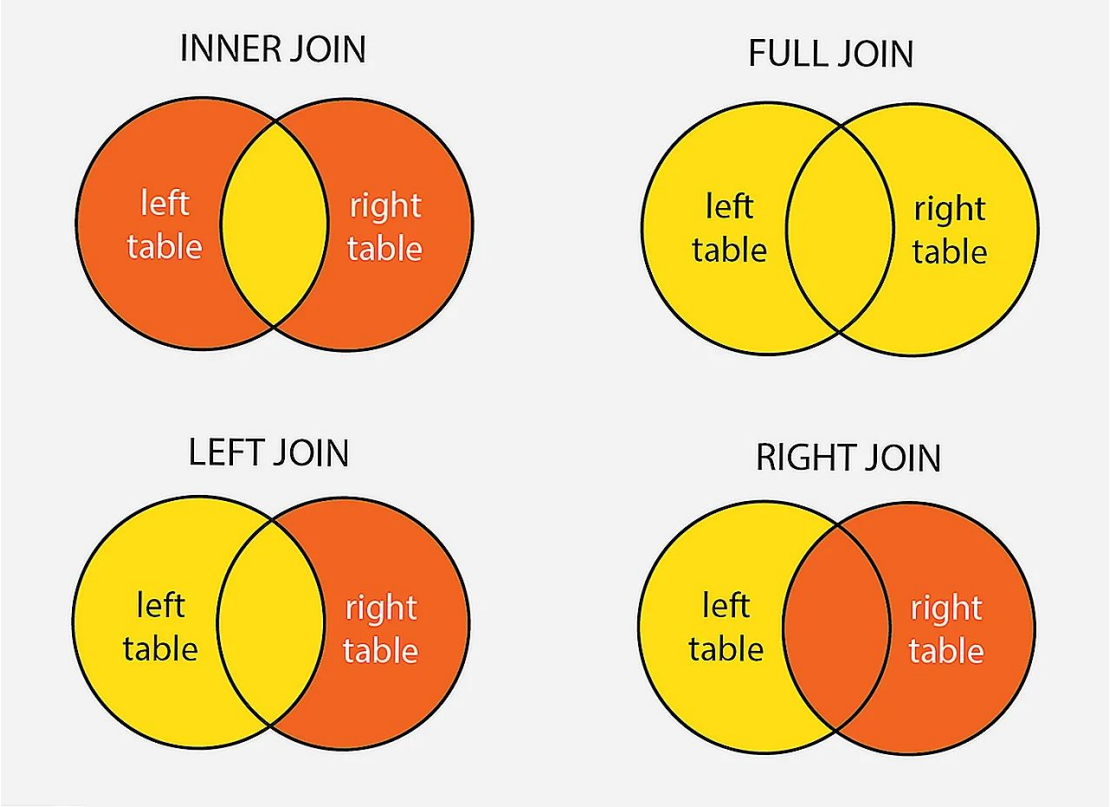
2. **UNION ALL** не удаляет повторяющиеся строки

UNION в пределах одной таблицы. Начисление процентов на вклад:

```
1 SELECT First Name, Last Name, AccountSum + AccountSum * 0.1 AS TotalSum
2 FROM Customers WHERE AccountSum < 3000
3 UNION SELECT FirstName, LastName, AccountSum + AccountSum * 0.3 AS TotalSum
4 FROM Customers WHERE AccountSum ≥ 3000;
```

Если сумма меньше 3000, то начисляются проценты в размере 10% от суммы на счете.
Если на счете больше 3000, то проценты увеличиваются до 30%.

Соединение таблиц — JOIN



Поддерживаемые типы объединений в MySQL

1. INNER JOIN: возвращает записи с совпадающими значениями в обеих таблицах.

2. LEFT JOIN: возвращает все записи из левой таблицы и соответствующие записи из правой таблицы.

3 RIGHT JOIN: возвращает все записи из правой таблицы и соответствующие записи из левой таблицы.

4. CROSS JOIN: возвращает все записи из обеих таблиц.

INNER JOIN:

```
1 SELECT столбцы
2 FROM таблица1
3     [INNER] JOIN таблица2
4     ON условие1
5     [[INNER] JOIN таблица 3
6     ON условие2]
```

INNER JOIN: возьмем БД из прошлых уроков

```sql
CREATE TABLE Products
(
    Id INT AUTO_INCREMENT PRIMARY KEY,
    ProductName VARCHAR(30) NOT NULL,
    Manufacturer VARCHAR(20) NOT NULL,
    ProductCount INT DEFAULT 0,
    Price DECIMAL NOT NULL
);
CREATE TABLE Customers
(
    Id INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(30) NOT NULL
);
CREATE TABLE Orders
(
    Id INT AUTO_INCREMENT PRIMARY KEY,
    ProductId INT NOT NULL,
    CustomerId INT NOT NULL,
    CreatedAt DATE NOT NULL,
    ProductCount INT DEFAULT 1,
    Price DECIMAL NOT NULL,
    FOREIGN KEY (ProductId) REFERENCES Products(Id) ON DELETE CASCADE,
    FOREIGN KEY (CustomerId) REFERENCES Customers(Id) ON DELETE CASCADE
);
```

INNER JOIN: пример

Выберем все заказы и добавим к ним информацию о товарах:

```
1 SELECT Orders.CreatedAt, Orders.ProductCount, Products.ProductName
2 FROM Orders
3 JOIN Products ON Products.Id = Orders.ProductId;
```

```
1  USE productsdb;
2
3  SELECT Orders.CreatedAt, Orders.ProductCount, Products.ProductName
4  FROM Orders
5  JOIN Products ON Products.Id = Orders.ProductId;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CreatedAt	ProductCount	ProductName
2018-05-21	2	Galaxy S8
2018-05-23	1	iPhone X
2018-05-21	1	iPhone X

INNER JOIN: пример с использованием псевдонимов

Выберем все заказы и добавим к ним информацию о товарах:
сократим “Orders” до “O”, “Products” до “P”

```
1 SELECT O.CreatedAt, O.ProductCount, P.ProductName
2 FROM Orders AS O
3 JOIN Products AS P
4 ON P.Id = O.ProductId;
```

OUTER JOIN

```
1 SELECT столбцы
2 FROM таблица1
3     {LEFT|RIGHT} [OUTER] JOIN таблица2 ON условие1
4     [{LEFT|RIGHT} [OUTER] JOIN таблица3 ON условие2]...
```

Перед оператором JOIN указывается одно из ключевых слов LEFT или RIGHT, которые определяют тип соединения:

LEFT: выборка будет содержать все строки из первой или левой таблицы

RIGHT: выборка будет содержать все строки из второй или правой таблицы

LEFT OUTER JOIN: пример

```
1 SELECT FirstName, CreatedAt, ProductCount, Price, ProductId
2 FROM Orders LEFT JOIN Customers
3 ON Orders.CustomerId = Customers.Id
```

```
1  USE productsdb;
2
3  SELECT FirstName, CreatedAt, ProductCount, Price, ProductId
4  FROM Orders LEFT JOIN Customers
5  ON Orders.CustomerId = Customers.Id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	CreatedAt	ProductCount	Price	ProductId
Tom	2018-05-21	2	46000	5
Tom	2018-05-23	1	76000	1
Bob	2018-05-21	1	76000	1

OUTER JOIN и INNER

```
#INNER JOIN
SELECT FirstName, CreatedAt, ProductCount, Price
FROM Customers JOIN Orders
ON Orders.CustomerId = Customers.Id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	CreatedAt	ProductCount	Price
Tom	2018-05-21	2	46000
Tom	2018-05-23	1	76000
Bob	2018-05-21	1	76000

INNER JOIN

```
#LEFT JOIN
SELECT FirstName, CreatedAt, ProductCount, Price
FROM Customers LEFT JOIN Orders
ON Orders.CustomerId = Customers.Id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	CreatedAt	ProductCount	Price
Tom	2018-05-21	2	46000
Tom	2018-05-23	1	76000
Bob	2018-05-21	1	76000
Sam	NULL	NULL	NULL

LEFT JOIN

RIGHT OUTER JOIN: пример

```
1 SELECT FirstName, CreatedAt, ProductCount, Price
2 FROM Customers RIGHT JOIN Orders
3 ON Orders.CustomerId = Customers.Id;
```

```
1  USE productsdb;
2
3  # RIGHT JOIN
4  SELECT FirstName, CreatedAt, ProductCount, Price
5  FROM Customers RIGHT JOIN Orders
6  ON Orders.CustomerId = Customers.Id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	CreatedAt	ProductCount	Price
Tom	2018-05-21	2	46000
Tom	2018-05-23	1	76000
Bob	2018-05-21	1	76000

FULL JOIN: пример замены через UNION

Таблица products

Result Grid | Filter Rows:

product_id	category	product_name
1	1	Системный блок
2	1	Монитор
3	2	Холодильник
4	2	Телевизор
5	NULL	Операционная система

Таблица categories

Result Grid | Filter Rows:

category_id	category_name
1	Комплектующие компьютера
2	Бытовая техника
3	Мобильные устройства

```
1
2  SELECT p.product_name, c.category_name
3  FROM products p
4  LEFT JOIN categories c ON p.category = c.category_id
5
6  UNION
7
8  SELECT p.product_name, c.category_name
9  FROM products p
10 RIGHT JOIN categories c ON p.category = c.category_id;
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	category_name
Системный блок	Комплектующие компьютера
Монитор	Комплектующие компьютера
Холодильник	Бытовая техника
Телевизор	Бытовая техника
Операционная система	NULL
NULL	Мобильные устройства

FULL JOIN: пример замены через UNION ALL

Таблица products

Result Grid | Filter Rows:

product_id	category	product_name
1	1	Системный блок
2	1	Монитор
3	2	Холодильник
4	2	Телевизор
5	NULL	Операционная система

Таблица categories

Result Grid | Filter Rows:

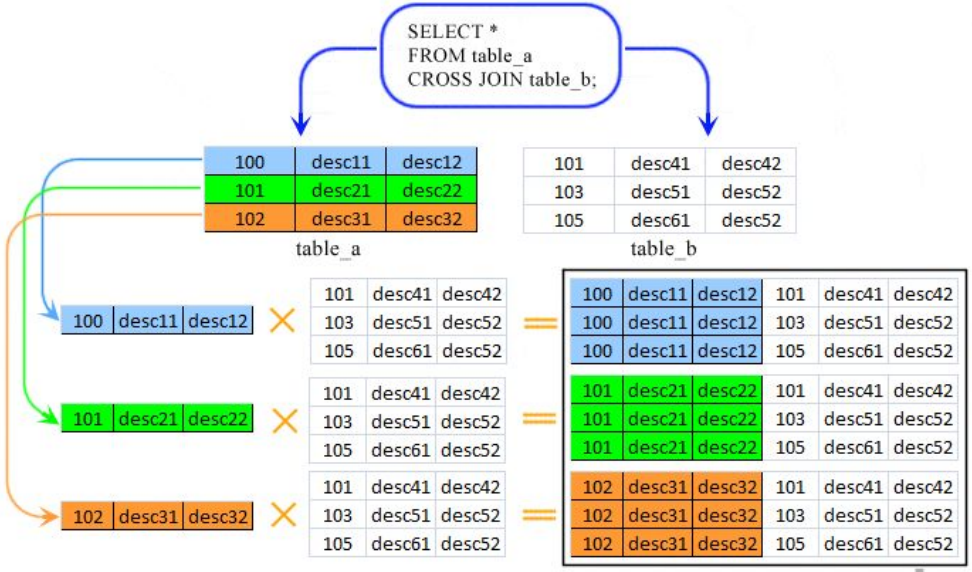
category_id	category_name
1	Комплектующие компьютера
2	Бытовая техника
3	Мобильные устройства

```
1
2   SELECT p.product_name, c.category_name
3   FROM products p
4   LEFT JOIN categories c ON p.category = c.category_id
5
6   UNION ALL
7
8   SELECT p.product_name, c.category_name
9   FROM products p
10  RIGHT JOIN categories c ON p.category = c.category_id
11  WHERE p.category IS NULL;
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

product_name	category_name
Системный блок	Комплектующие компьютера
Монитор	Комплектующие компьютера
Холодильник	Бытовая техника
Телевизор	Бытовая техника
Операционная система	NULL
NULL	Мобильные устройства

SQL CROSS JOIN




```
id         des1       des2       id         des3       des4
---------- ---------- ---------- ---------- ---------- -------
100        desc11     desc12     101        desc41     desc42
100        desc11     desc12     103        desc51     desc52
100        desc11     desc12     105        desc61     desc62
101        desc21     desc22     101        desc41     desc42
101        desc21     desc22     103        desc51     desc52
101        desc21     desc22     105        desc61     desc62
102        desc31     desc32     101        desc41     desc42
102        desc31     desc32     103        desc51     desc52
102        desc31     desc32     105        desc61     desc62
```

Краткий синтаксис	Полный синтаксис	Описание (
JOIN	INNER JOIN	Из строк **левой_таблицы** и **правой_таблицы** объединяются и возвращаются только те строки, по которым выполняются **условия_соединения**.
LEFT JOIN	LEFT OUTER JOIN	Возвращаются все строки **левой_таблицы** (ключевое слово LEFT). Данными **правой_таблицы** дополняются только те строки **левой_таблицы**, для которых выполняются **условия_соединения**. Для недостающих данных вместо строк **правой_таблицы** вставляются NULL-значения.
RIGHT JOIN	RIGHT OUTER JOIN	Возвращаются все строки **правой_таблицы** (ключевое слово RIGHT). Данными **левой_таблицы** дополняются только те строки **правой_таблицы**, для которых выполняются **условия_соединения**. Для недостающих данных вместо строк **левой_таблицы** вставляются NULL-значения.
FULL JOIN	FULL OUTER JOIN	Возвращаются все строки **левой_таблицы** и **правой_таблицы**. Если для строк **левой_таблицы** и **правой_таблицы** выполняются **условия_соединения**, то они объединяются в одну строку. Для строк, для которых не выполняются **условия_соединения**, NULL-значения вставляются на место **левой_таблицы**, либо на место **правой_таблицы**, в зависимости от того данных какой таблицы в строке нет.
CROSS JOIN	-	Объединение каждой строки **левой_таблицы** со всеми строками **правой_таблицы**. Этот вид соединения иногда называют декартовым произведением.

Подзапросы: оператор IN

Выберем все товары из таблицы Products, на которые есть заказы в таблице Orders:

```
1 SELECT * FROM Products
2 WHERE Id IN (SELECT ProductId FROM Orders)
```

Можем выбрать те товары, на которые нет заказов в таблице Orders:

```
1 SELECT * FROM Products
2 WHERE Id NOT IN (SELECT ProductId FROM Orders)
```

Подзапросы: оператор EXISTS

```
1 WHERE [NOT] EXISTS (подзапрос)
```

```
1  USE productsdb;
2
3  SELECT * FROM Products
4  WHERE EXISTS
5  (SELECT * FROM Orders WHERE Orders.ProductId = Products.Id)
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Id	ProductName	Manufacturer	ProductCount	Price
1	iPhone X	Apple	2	76000
2	iPhone 8	Apple	2	51000
5	Galaxy S8	Samsung	1	46000
NULL	NULL	NULL	NULL	NULL

CREATE TABLE SELECT.

```
1 CREATE TABLE yourTableName SELECT *FROM yourOriginalTableName;
```

SQL File 3*

Limit to 1000 rows

```
1  CREATE TABLE SelectIntoDemo
2  (
3      Id INT,
4      FirstName VARCHAR(200)
5  );
6
7  INSERT INTO SelectIntoDemo VALUES(1,'Bob'),(2,'Carol'),(3,'David');
8
9  SELECT * FROM SelectIntoDemo;
10 CREATE TABLE  yourTempTable SELECT * FROM SelectIntoDemo;
11 SELECT * FROM yourTempTable;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Id	FirstName
1	Bob
2	Carol
3	David

Порядок выполнения запроса

```
1 SELECT [DISTINCT | ALL] поля_таблиц
2 FROM список_таблиц
3 [WHERE условия_на_ограничения_строк]
4 [GROUP BY условия_группировки]
5 [HAVING условия_на_ограничения_строк_после_группировки]
6 [ORDER BY порядок_сортировки [ASC | DESC]]
7 [LIMIT ограничение_количества_записей]
```

Итоги занятия:

- 📌 Операторы union и union all
- 📌 Соединение таблиц — Join (left, right, inner, full, cross)
- 📌 Использование подзапросов (join, in, exists, select)
- 📌 Использование select into, создание таблиц из результатов запроса
- 📌 Порядок выполнения запроса — дополнение схемы

Спасибо за внимание!