



Базы данных и SQL

Семинар 4.



Вопросы?



Вопросы?



Вопросы?



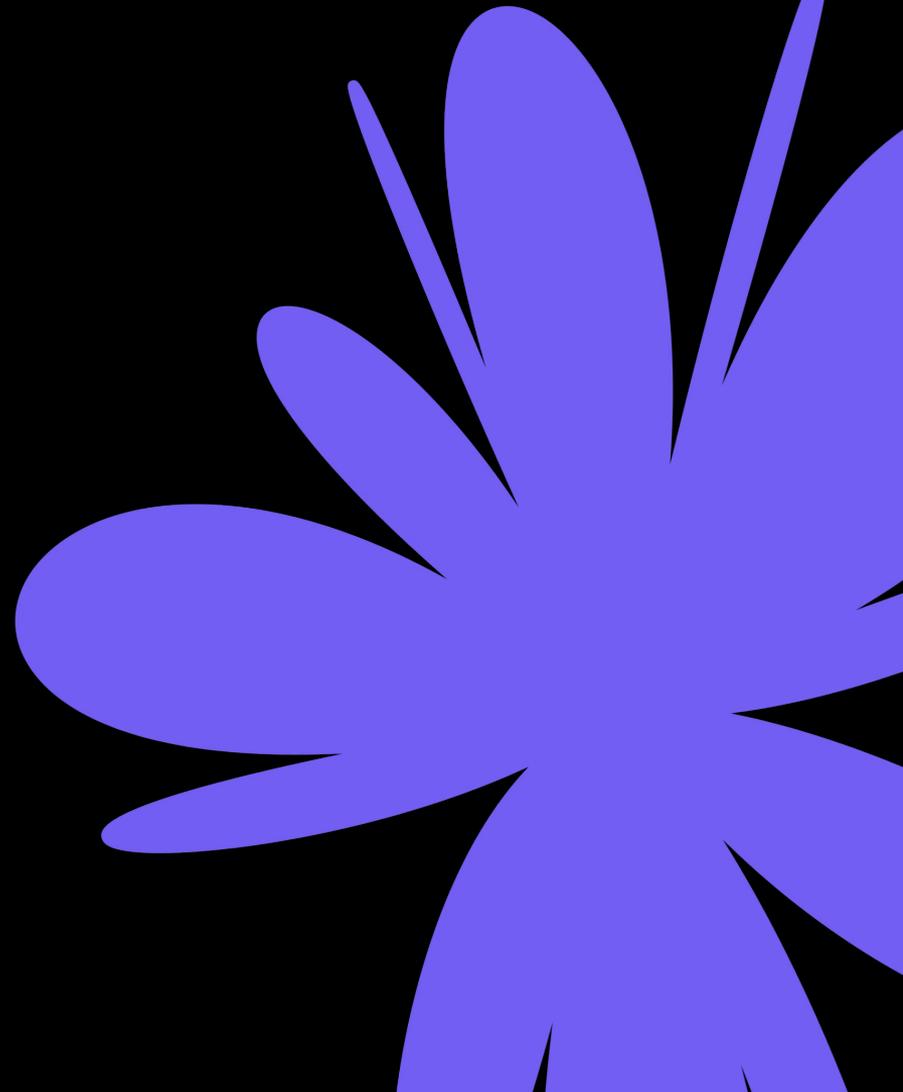
План на сегодня:

- Quiz!
- Объединение таблиц (UNION)
- Соединение таблиц (JOIN)
- Перерыв
- Использование подзапросов
- Домашнее задание





Quiz!



Что такое JOIN?

1. операция объединения
2. операция группировки
3. операция суммирования
4. операция создания



Что такое JOIN?

1. операция объединения
2. операция группировки
3. операция суммирования
4. операция создания



Какого из перечисленных ниже видов JOIN на самом деле не существует?

1. LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL
2. RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL
3. INNER JOIN - который показывает только те записи, для которых нашлись пары
4. TRUE JOIN - который выведет все верные значения



Какого из перечисленных ниже видов JOIN на самом деле не существует?

1. LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL
2. RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL
3. INNER JOIN - который показывает только те записи, для которых нашлись пары
4. TRUE JOIN - который выведет все верные значения



Выберите правильный пример запроса с использованием UNION?

1. `select id, city from Orders order by id union select id, city from Sellers order by city;`
2. `select id, city, seller_id from Orders and select city, id from Select order by id;`
3. `select id, city from Orders union select id, city from Sellers order by id;`
4. Все запросы верные



Выберите правильный пример запроса с использованием UNION?

1. `select id, city from Orders order by id union select id, city from Sellers order by city;`
2. `select id, city, seller_id from Orders and select city, id from Select order by id;`
3. `select id, city from Orders union select id, city from Sellers order by id;`
4. Все запросы верные



Если выборка объединения данных производится из нескольких таблиц, то это может указываться во фразе FROM следующим образом?

1. таблица 1 таблица 2
2. таблица 1 AND таблица2
3. таблица 1 CROSS JOIN таблица2
4. таблица1 INNER JOIN таблица2
5. таблица 1 OUTER JOIN таблица2



Если выборка объединения данных производится из нескольких таблиц, то это может указываться во фразе FROM следующим образом?

1. таблица 1 таблица 2
2. таблица 1 AND таблица2
3. таблица 1 CROSS JOIN таблица2
4. таблица1 INNER JOIN таблица2
5. таблица 1 OUTER JOIN таблица2



Какая команда используется для объединения результатов запроса без удаления дубликатов?

1. UNION
2. UNION ALL



Какая команда используется для объединения результатов запроса без удаления дубликатов?

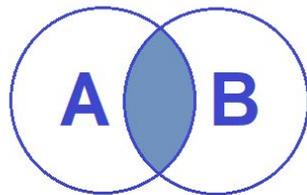
1. UNION
2. UNION ALL



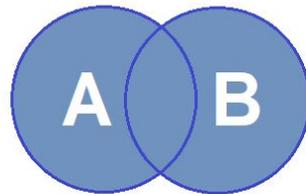
Поддерживаемые типы объединений

Таблицы для работы:

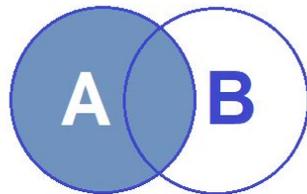
<https://drive.google.com/file/d/1gwhpa1idMipop59KNDb3dc9PBpWVlPCL/view?usp=sharing>



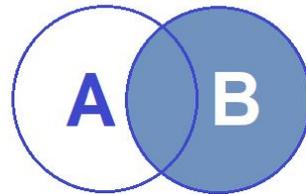
INNER JOIN



FULL OUTER JOIN



LEFT JOIN



RIGHT JOIN

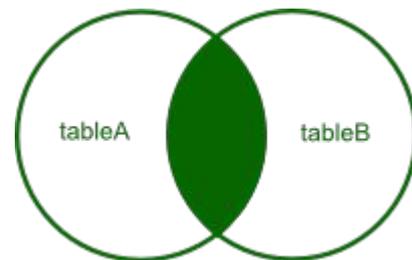


INNER JOIN

15мин

tableA	
id	name
1	Яблоко
2	Ананас
3	Банан
4	Груша

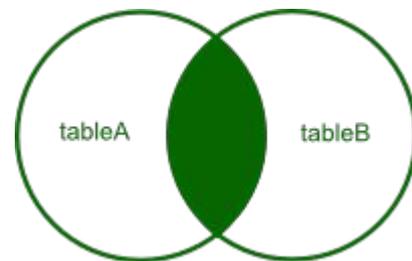
table B	
id	name
1	Тыква
2	Яблоко
3	Дыня
4	Банан



INNER JOIN

15мин

```
SELECT *  
FROM tableA  
INNER JOIN tableB  
ON tableA.name = tableB.name
```



id	name	id	name
1	Яблоко	2	Яблоко
3	Банан	4	Банан



15мин

INNER JOIN

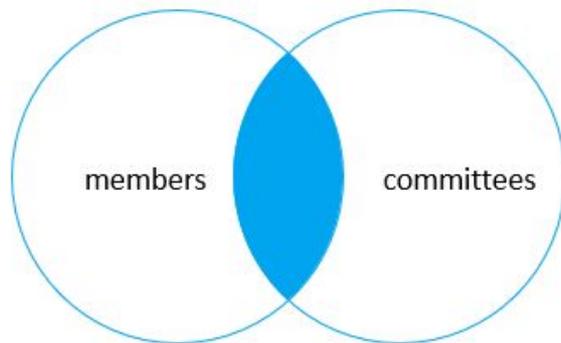
```
SELECT column_list  
FROM table_1  
INNER JOIN table_2 ON join_condition
```

USING:

```
SELECT column_list  
FROM table_1  
INNER JOIN table_2 USING (column_name);
```

Задание:

Выведите участников, которые также являются членами комитета, используйте INNER JOIN (пересечение 2 таблиц по имени)



FULL JOIN

SELECT * FROM A

LEFT JOIN B ON A.key = B.key

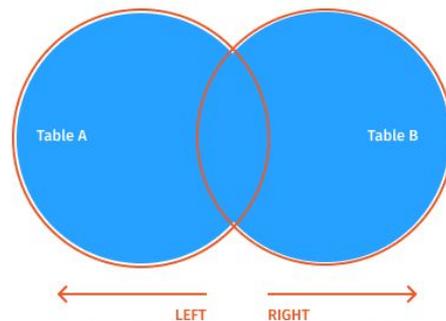
UNION

SELECT * FROM A

RIGHT JOIN B ON A.key = B.key

FULL JOIN

15МИН



Ссылка на материалы:

<https://drive.google.com/file/d/18cpqHW2kKRnWFX1TUiziYUX8YrdVHAPL/view?usp=sharing>



FULL JOIN: UNION

15мин

```
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
```

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

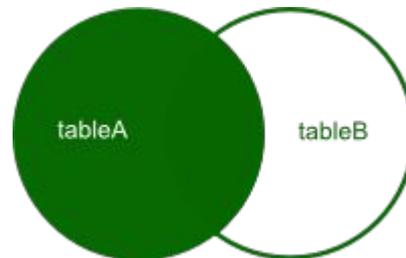


LEFT JOIN

tableA	
id	name
1	Яблоко
2	Ананас
3	Банан
4	Груша

table B	
id	name
1	Тыква
2	Яблоко
3	Дыня
4	Банан

15мин

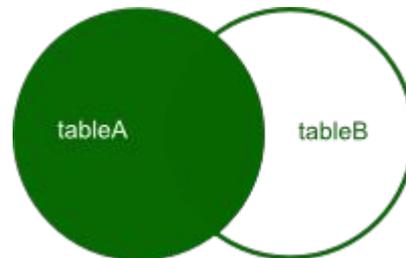


LEFT JOIN

```
SELECT * FROM tableA  
LEFT OUTER JOIN tableB  
ON tableA.name = tableB.name
```

Выбираем все записи из таблицы A и совпавшие записи из таблицы B. Если совпадение отсутствует, то правая сторона будет содержать значение null, т.к. управляющая таблица левая, об этом говорит ключевое слово LEFT.

15мин



LEFT JOIN

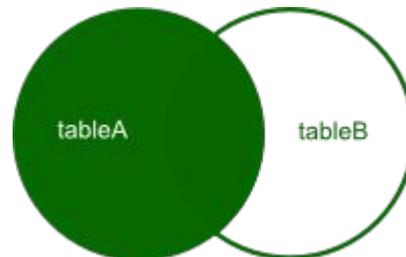
```
SELECT * FROM tableA
```

```
LEFT OUTER JOIN tableB
```

```
ON tableA.name = tableB.name
```

id	name	id	name
--	----	--	----
1	Яблоко	2	Яблоко
2	Ананас	null	null
3	Банан	4	Банан
4	Груша	null	null

15МИН



LEFT JOIN

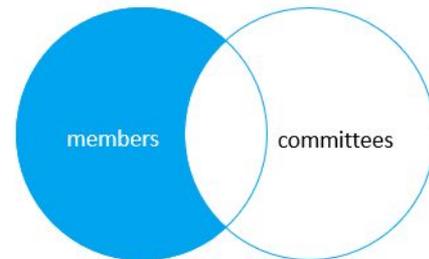
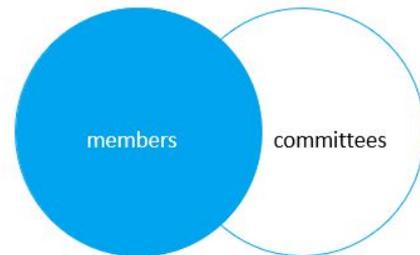
15мин

Таблица:

<https://drive.google.com/file/d/1gwhpa1idMipop59KNDb3dc9PBpWVlPCL/view?usp=sharing>

Задание:

1. Выведите участников, которые являются членами комитета (Используя LEFT JOIN, выполните соединение двух таблиц по полю “Имя”)
- 2.* Выведите участников, которые не являются членами комитета



RIGHT JOIN

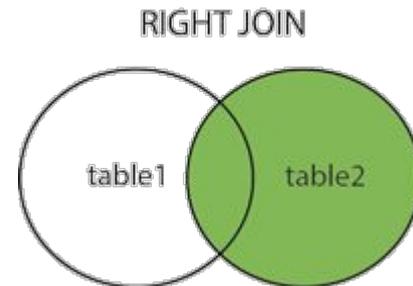
SELECT * FROM tableA

RIGHT OUTER JOIN tableB

ON tableB.name = tableA.name

Выбираем все записи из таблицы B и совпавшие записи из таблицы A. Если совпадение отсутствует, то левая сторона будет содержать значение null, т.к. управляющая таблица правая, об этом говорит ключевое слово RIGHT.

15мин



15мин

RIGHT JOIN: уникальные данные из правой таблицы

```
SELECT * FROM tableA
```

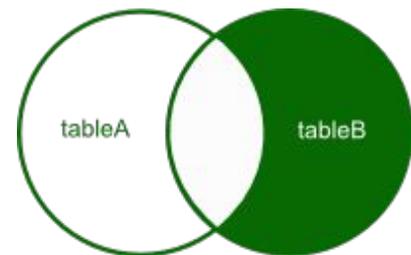
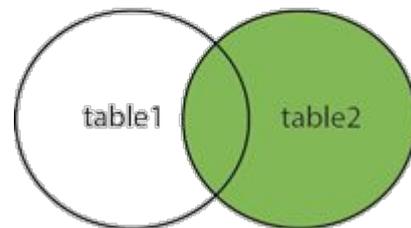
```
RIGHT OUTER JOIN tableB
```

```
ON tableA.name = tableB.name
```

```
WHERE tableA.id IS null
```

id	name	id	name
--	----	--	----
null	null	1	Тыква
null	null	3	Дыня

RIGHT JOIN



RIGHT JOIN:

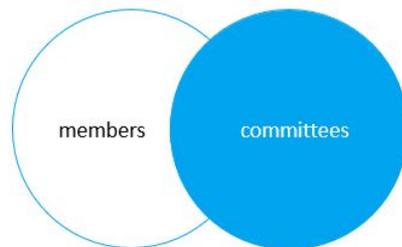
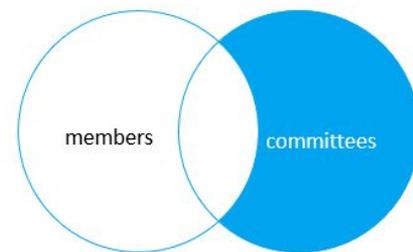
Таблица:

<https://drive.google.com/file/d/1gwhpa1idMipop59KNDb3dc9PBpWVlPCL/view?usp=sharing>

Задание:

1. Найти членов комитета, которых нет в members с помощью RIGHT JOIN
2. Выполнить RIGHT JOIN между 2 таблицами по полю “Имя”

15мин



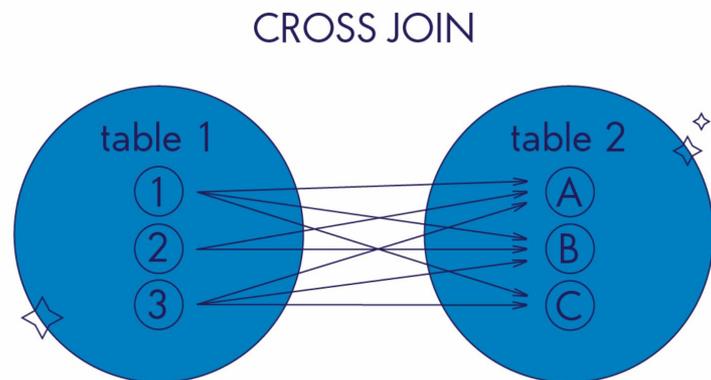
CROSS JOIN

10мин

```
SELECT select_list  
FROM table_1  
CROSS JOIN table_2;
```

Перекрестное соединение создает декартово произведение строк из соединенных таблиц.

Предположим, что в первой таблице есть **n** строк, а во второй таблице есть **m** строк. Перекрестное соединение, которое соединяет таблицы, возвращает **$n \times m$** строк.



```
SELECT *  
FROM LeftTable
```

LCode	LDescr
1	L-1
2	L-2
3	L-3
5	L-5



LCode	LDescr
-------	--------

1	L-1
---	-----

+

2	L-2
---	-----

+

3	L-3
---	-----

+

5	L-5
---	-----

+

```
SELECT *  
FROM RightTable
```

RCode	RDescr
2	B-2
3	B-3
4	B-4



RCode	RDescr
-------	--------

2	B-2
3	B-3
4	B-4

2	B-2
3	B-3
4	B-4

2	B-2
3	B-3
4	B-4

2	B-2
3	B-3
4	B-4



```
SELECT l.*,r.*  
FROM LeftTable l  
CROSS JOIN RightTable r
```

LCode	LDescr	RCode	RDescr
1	L-1	2	B-2
1	L-1	3	B-3
1	L-1	4	B-4
2	L-2	2	B-2
2	L-2	3	B-3
2	L-2	4	B-4
3	L-3	2	B-2
3	L-3	3	B-3
3	L-3	4	B-4
5	L-5	2	B-2
5	L-5	3	B-3
5	L-5	4	B-4

15МИН



15мин

CROSS JOIN

```
SELECT select_list  
FROM table_1  
CROSS JOIN table_2;
```

Таблица:

<https://drive.google.com/file/d/1gwhpa1idMipop59KNDb3dc9PBpWVlPCL/view?usp=sharing>

Задание:

1. Выведите перекрестное произведение таблиц members - committees



15МИН

CROSS JOIN

SELECT

m.member_id,

m.name AS member,

c.committee_id,

c.name AS committee

FROM

members m

CROSS JOIN committees c;

```
+-----+-----+-----+-----+
| member_id | member | committee_id | committee |
+-----+-----+-----+-----+
|         1 | John   |             4 | Joe       |
|         1 | John   |             3 | Amelia    |
|         1 | John   |             2 | Mary      |
|         1 | John   |             1 | John      |
|         2 | Jane   |             4 | Joe       |
|         2 | Jane   |             3 | Amelia    |
|         2 | Jane   |             2 | Mary      |
|         2 | Jane   |             1 | John      |
|         3 | Mary   |             4 | Joe       |
|         3 | Mary   |             3 | Amelia    |
|         3 | Mary   |             2 | Mary      |
|         3 | Mary   |             1 | John      |
|         4 | David  |             4 | Joe       |
|         4 | David  |             3 | Amelia    |
|         4 | David  |             2 | Mary      |
|         4 | David  |             1 | John      |
|         5 | Amelia |             4 | Joe       |
|         5 | Amelia |             3 | Amelia    |
|         5 | Amelia |             2 | Mary      |
|         5 | Amelia |             1 | John      |
+-----+-----+-----+-----+
20 rows in set (0.00 sec)
```



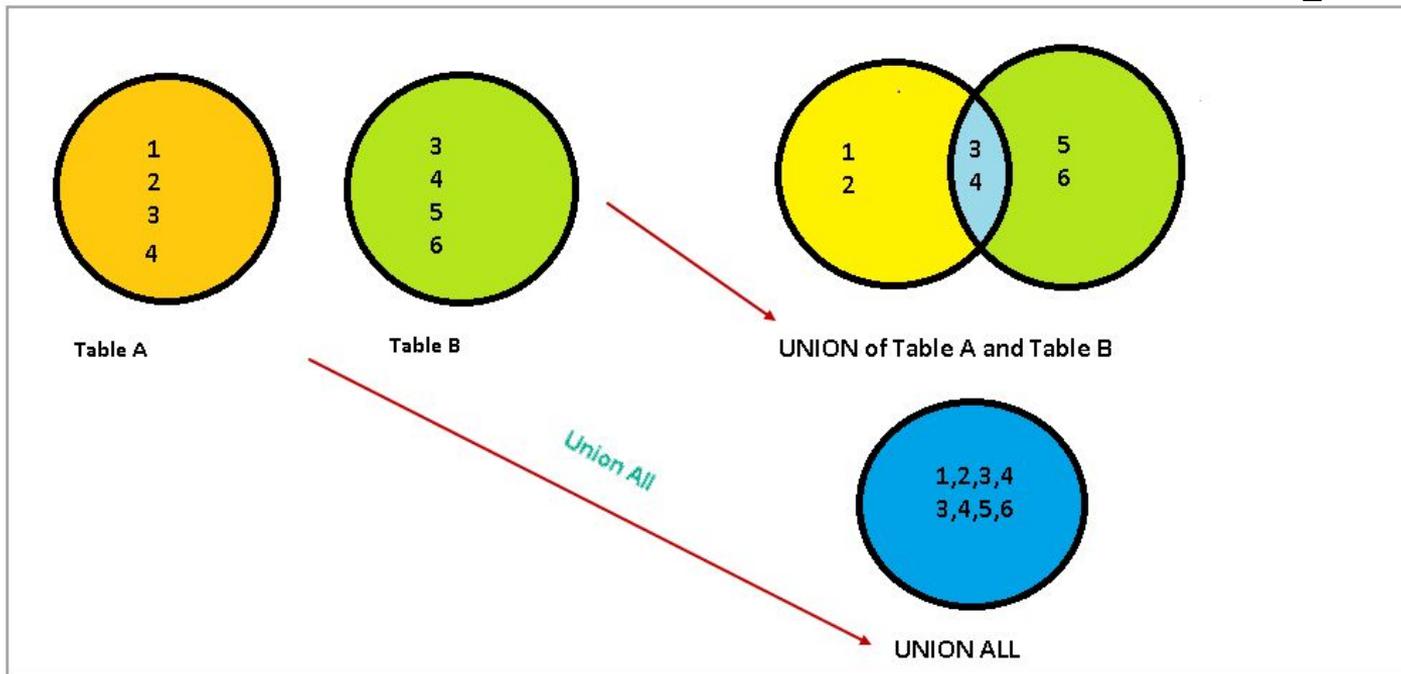
Ваши вопросы?

Перерыв



UNION и UNION ALL

20 МИН

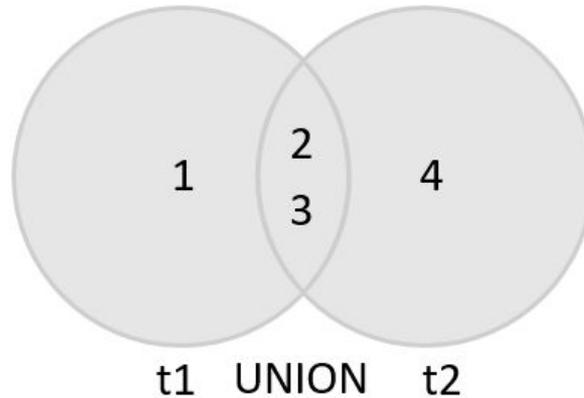


UNION: пример

```
CREATE TABLE t1  
(  
  id INT PRIMARY KEY  
);
```

```
CREATE TABLE t2  
(  
  id INT PRIMARY KEY  
);
```

```
INSERT INTO t1 VALUES (1),(2),(3);  
INSERT INTO t2 VALUES (2),(3),(4);  
-- UNION  
SELECT id FROM t1  
UNION  
SELECT id FROM t2;
```



20 МИН



UNION и UNION ALL

Табличка для работы:

<https://drive.google.com/file/d/1PQn576YVakvlWrIgjSP9YEf5id4cqYs/view?usp=sharing>

Задание:

1. Получить список пользователей и клиентов, удалив одинаковых клиентов и пользователей
2. Получить список пользователей и клиентов. Дубликаты удалять не нужно

20 мин

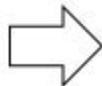


UNION vs JOIN

id
1
2
3

UNION

id
2
3
4



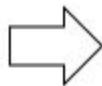
id
1
2
3
4

Append
result sets
vertically

id
1
2
3

INNER
JOIN

id
2
3
4



id	id
2	2
3	3

Append
result sets
horizontally

20 МИН



Оператор IN

SELECT *column_name(s)*

FROM *table_name*

WHERE *column_name* **IN** (*value1*, *value2*, ...);

Таблица:

<https://drive.google.com/file/d/1PQn576YVakvlWrIgjSP9YEf5id4cqYs/view?usp=sharing>

1. Проверьте, присутствует ли буква “A” в последовательности 'A', 'B', 'C', 'D'
2. Проверьте, присутствует ли буква “Z” в последовательности 'A', 'B', 'C', 'D'
3. Получить столбцы из таблицы “clients” , в которых первое имя является набором значений.
4. Выберите все логины из таблицы “users”, кроме “Mikle”.

25 МИН



Оператор EXISTS

Табличка:

<https://drive.google.com/file/d/1PQn576YVakvlWrIgIjSP9Yef5id4cqYs/view?usp=sharing>

25 МИН



Employee

Id	Name	Department	Salary	Gender	Age	City
1001	John Doe	IT	35000	Male	25	London
1002	Mary Smith	HR	45000	Female	27	London
1003	James Brown	Finance	50000	Male	28	London
1004	Mike Walker	Finance	50000	Male	28	London
1005	Linda Jones	HR	75000	Female	26	London
1006	Anurag Mohanty	IT	35000	Male	25	Mumbai
1007	Priyanka Dewangan	HR	45000	Female	27	Mumbai
1008	Sambit Mohanty	IT	50000	Male	28	Mumbai
1009	Pranaya Kumar	IT	50000	Male	28	Mumbai
1010	Hina Sharma	HR	75000	Female	26	Mumbai

Projects

ProjectId	Title	ClientId	EmployeeId	StartDate	EndDate
1	Develop ecommerce website from scratch	1	1003	2021-04-18 16:03:26	2021-05-18 16:03:26
2	WordPress website for our company	1	1002	2021-04-18 16:03:26	2021-06-02 16:03:26
3	Manage our company servers	2	1007	2021-04-18 16:03:26	2021-06-02 16:03:26
4	Hosting account is not working	3	1009	2021-04-18 16:03:26	2021-04-25 16:03:26
5	MySQL database from my desktop application	4	1010	2021-04-18 16:03:26	2021-05-03 16:03:26
6	Develop new WordPress plugin for my business ...	2	NULL	2021-04-18 16:03:26	2021-04-28 16:03:26
7	Migrate web application and database to new s...	2	NULL	2021-04-18 16:03:26	2021-04-23 16:03:26
8	Android Application development	4	1004	2021-04-18 16:03:26	2021-05-18 16:03:26



Оператор EXISTS

1. Получите из таблицы "Сотрудники" только тех сотрудников, которые в данный момент работают над любым из активных проектов.
2. Проверьте, существует ли сотрудник с идентификатором 1004 в таблице сотрудников или нет.

25 мин



EXISTS VS IN

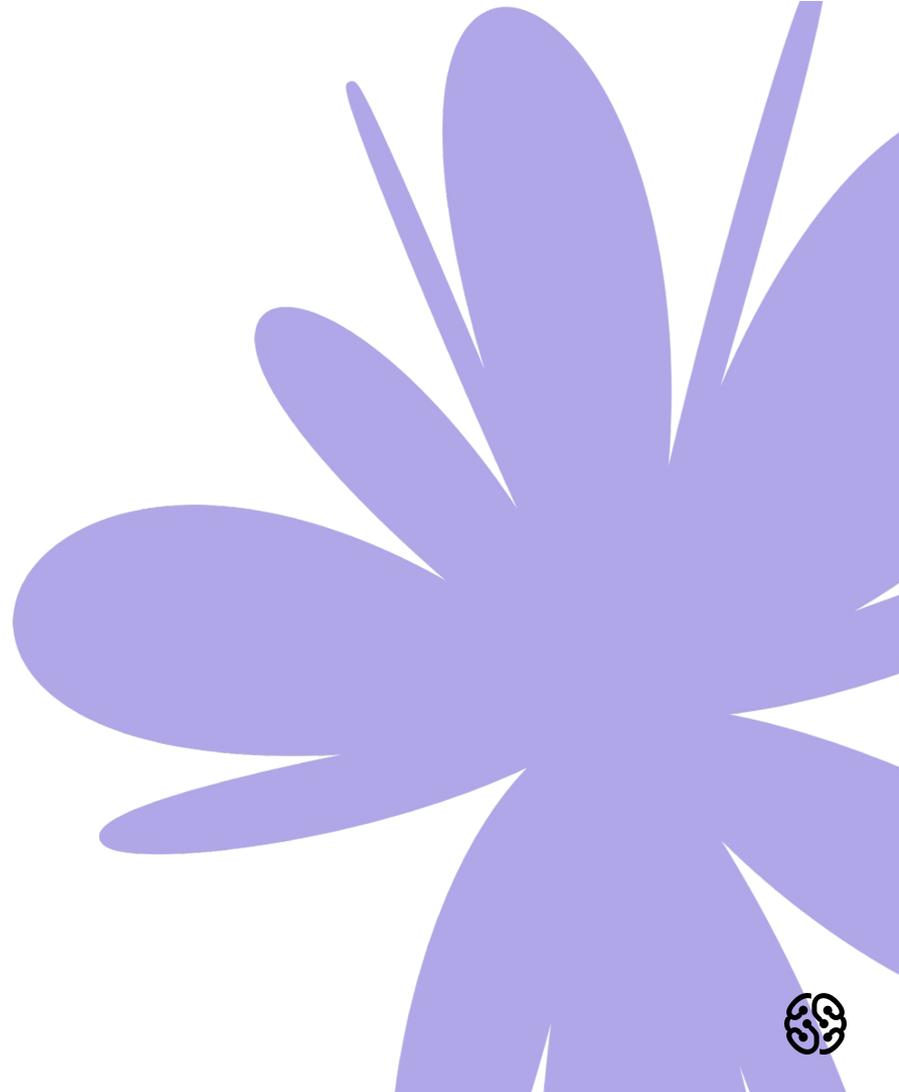
25 мин



1. EXISTS используется для проверки существования данных в подзапросе, тогда как IN используется для минимизации множественных условий ИЛИ.
2. Оператор EXISTS останавливает свое дальнейшее выполнение, как только находит первое истинное вхождение, тогда как оператор IN сравнивает все значения, присутствующие внутри него.
3. Оператор EXISTS не выполняет сравнение между родительским запросом и дочерним запросом, тогда как оператор IN выполняет сравнение между родительским запросом и дочерним запросом.



Ваши вопросы?



Табличка:

<https://drive.google.com/file/d/1PQn576YVakvlWrIgjSP9YEf5id4cqYs/view?usp=sharing>

1. Вывести на экран, сколько машин каждого цвета для машин марок BMW и LADA

2. Вывести на экран марку авто(количество) и количество авто не этой марки.

100 машин, их них 20 - BMW и 80 машин другой марки , AUDI - 30 и 70 машин другой марки, LADA - 15, 85 авто другой марки

Задание №3.

Даны 2 таблицы, созданные следующим образом:

```
create table test_a (id INT, test varchar(10));
```

```
create table test_b (id INT);
```



```
insert into test_a(id, test) values
```

```
(10, 'A'),
```

```
(20, 'A'),
```

```
(30, 'F'),
```

```
(40, 'D'),
```

```
(50, 'C');
```

```
insert into test_b(id) values
```

```
(10),
```

```
(30),
```

```
(50);
```

Напишите запрос, который вернет строки из таблицы test_a, id которых нет в таблице test_b, НЕ используя ключевого слова NOT.



Рефлексия



Был урок полезен вам?



Узнали вы что-то новое?



Что было сложно?



Спасибо
за внимание

