



Базы данных

Лекция 2

SQL



Андрей Каледин

В предыдущих сериях...

- Виды моделей данных
- OLTP vs OLAP
- Основные сущности реляционной модели
- Ключи

План лекции

1. DDL vs DML vs TCL
2. CREATE
3. INSERT, UPDATE, DELETE
4. SELECT
5. GROUP BY
6. WITH
7. Оконные функции
8. JOIN

DDL vs DML vs TCL

DDL vs DML vs TCL

Название	Назначение	Примеры
Data Definition Language (Язык определения данных)	Создание и изменение <u>структуры БД</u>	<ul style="list-style-type: none">• CREATE TABLE• ALTER TABLE• CREATE INDEX

DDL vs DML vs TCL

Название	Назначение	Примеры
Data Definition Language (Язык определения данных)	Создание и изменение <u>структуры БД</u>	<ul style="list-style-type: none">• CREATE TABLE• ALTER TABLE• CREATE INDEX
Data Manipulation Language (Язык управления данными)	Получение, создание, изменение и удаление <u>данных</u>	<ul style="list-style-type: none">• SELECT• INSERT• UPDATE• DELETE








DDL vs DML vs TCL

Название	Назначение	Примеры
Data Definition Language (Язык определения данных)	Создание и изменение <u>структуры БД</u>	<ul style="list-style-type: none">• CREATE TABLE• ALTER TABLE• CREATE INDEX
Data Manipulation Language (Язык управления данными)	Получение, создание, изменение и удаление <u>данных</u>	<ul style="list-style-type: none">• SELECT• INSERT• UPDATE• DELETE
Transaction Control Language (Язык управления транзакциями)	Сохранение и откат изменений	<ul style="list-style-type: none">• COMMIT• ROLLBACK• SAVEPOINT

CREATE TABLE









```
CREATE TABLE users (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(100) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL,  
  password VARCHAR(255) NOT NULL,  
  age INTEGER,  
  avatar VARCHAR(255),  
  created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);
```


CREATE TABLE

Column Name	#	Data type	Not Null	Default
 id	1	serial4	[v]	nextval('users_id_seq'::regclass)
 name	2	varchar(100)	[v]	
 email	3	varchar(255)	[v]	
 password	4	varchar(255)	[v]	
 age	5	int4	[]	
 avatar	6	varchar(255)	[]	
 created_at	7	timestamp	[v]	CURRENT_TIMESTAMP







ALTER TABLE

ALTER TABLE users
ADD COLUMN phone_number **VARCHAR(15)**;

Column Name	#	Data type	Not Null	Default
 id	1	serial4	[v]	nextval('users_id_seq'::regclass)
 name	2	varchar(100)	[v]	
 email	3	varchar(255)	[v]	
 password	4	varchar(255)	[v]	
 age	5	int4	[]	
 avatar	6	varchar(255)	[]	
 created_at	7	timestamp	[v]	CURRENT_TIMESTAMP
 phone_number	8	varchar(15)	[]	

CREATE INDEX

```
CREATE INDEX users_age_idx ON users (age);
```

Column	Index Name	Table	Ascending	Nullable	Unique
>  age	 users_age_idx	users	—	—	[]
>  email	 users_email_key	users	—	—	[v]
>  id	 users_pkey	users	—	—	[v]

INSERT

```
INSERT INTO users (name, email, password, age, avatar)
VALUES
('Иван Иванов', 'ivan@example.com', 'd131dd02c5e6eec4', 31, '/pictures/ivan.jpeg'),
('Мария Петрова', 'maria@example.com', '55ad340609f4b302', null, null),
('Алексей Смирнов', 'alex@example.com', 'd8823e3156348f5b', 15, '/root/alex.png'),
('Кек Лолов', 'kek@example.com', 'f131df02c5e6eec4', null, null),
('Эрик Картман', 'eric@southpark.com', '44ad440609f4b302', null, '/southpark/cartman.png');
```

	123 id 🔼🔽	ABC name 🔼🔽	ABC email 🔼🔽	ABC password 🔼🔽	123 age 🔼🔽	ABC avatar 🔼🔽	🕒 created_at 🔼🔽	ABC phone_number 🔼🔽
1	2	Мария Петрова	maria@example.com	55ad340609f4b302	[NULL]	[NULL]	2025-04-06 22:08:56.643	[NULL]
2	1	Иван Иванов	ivan@example.com	d131dd02c5e6eec4	31	/pictures/ivan.jpeg	2025-04-06 22:08:56.643	[NULL]
3	3	Алексей Смирнов	alex@example.com	d8823e3156348f5b	15	/root/alex.png	2025-04-06 22:08:56.643	[NULL]
4	5	Кек Лолов	kek@example.com	f131df02c5e6eec4	[NULL]	[NULL]	2025-04-06 22:30:11.684	[NULL]
5	6	Эрик Картман	eric@southpark.com	44ad440609f4b302	[NULL]	/southpark/cartman	2025-04-06 22:30:11.684	[NULL]

UPDATE

```
UPDATE users
SET PHONE_NUMBER = '88005553535'
WHERE email = 'eric@southpark.com';
```

	123 id 🔼🔼	ABC name 🔼🔼	ABC email 🔼🔼	ABC password 🔼🔼	123 age 🔼🔼	ABC avatar 🔼🔼	🕒 created_at 🔼🔼	ABC phone_number 🔼🔼
1	2	Мария Петрова	maria@example.com	55ad340609f4b302	[NULL]	[NULL]	2025-04-06 22:08:56.643	[NULL]
2	1	Иван Иванов	ivan@example.com	d131dd02c5e6eec4	31	/pictures/ivan.jpeg	2025-04-06 22:08:56.643	[NULL]
3	3	Алексей Смирнов	alex@example.com	d8823e3156348f5b	15	/root/alex.png	2025-04-06 22:08:56.643	[NULL]
4	5	Кек Лолов	kek@example.com	f131df02c5e6eec4	[NULL]	[NULL]	2025-04-06 22:30:11.684	[NULL]
5	6	Эрик Картман	eric@southpark.com	44ad440609f4b302	[NULL]	/southpark/cartman	2025-04-06 22:30:11.684	88005553535




DELETE

```
DELETE FROM users
WHERE name = 'Алексей Смирнов';
```

	123 id 🔼🔽	ABC name 🔼🔽	ABC email 🔼🔽	ABC password 🔼🔽	123 age 🔼🔽	ABC avatar 🔼🔽	🕒 created_at 🔼🔽	ABC phone_number 🔼🔽
1	2	Мария Петрова	maria@example.com	55ad340609f4b302	[NULL]	[NULL]	2025-04-06 22:08:56.643	[NULL]
2	1	Иван Иванов	ivan@example.com	d131dd02c5e6eec4	31	/pictures/ivan.jpeg	2025-04-06 22:08:56.643	[NULL]
3	5	Кек Лолов	kek@example.com	f131df02c5e6eec4	[NULL]	[NULL]	2025-04-06 22:30:11.684	[NULL]
4	6	Эрик Картман	eric@southpark.com	44ad440609f4b302	[NULL]	/southpark/cartman	2025-04-06 22:30:11.684	8800553535




SELECT

SELECT 1+3;

	123 ?column?  
1	4

SELECT

SELECT id, name, age **FROM** users;

	123 id 	ABC name 	123 age 
1	2	Мария Петрова	[NULL]
2	1	Иван Иванов	31
3	5	Кек Лолов	[NULL]
4	6	Эрик Картман	[NULL]

SELECT

SELECT * FROM users
WHERE avatar IS NOT NULL;

	<div>123id</div>	<div>ABCname</div>	<div>ABCemail</div>	<div>ABCpassword</div>	<div>123age</div>	<div>ABCavatar</div>	<div>🕒created_at</div>	<div>ABCphone_number</div>
1	1	Иван Иванов	ivan@example.com	d131dd02c5e6eec4	31	/pictures/ivan.jpeg	2025-04-06 22:08:56.643	[NULL]
2	6	Эрик Картман	eric@southpark.com	44ad440609f4b302	[NULL]	/southpark/cartman.png	2025-04-06 22:30:11.684	88005553535





SELECT

SELECT * FROM users
WHERE avatar IS NOT NULL AND age > 20;

	123 id 🔼🔼	ABC name 🔼🔼	ABC email 🔼🔼	ABC password 🔼🔼	123 age 🔼🔼	ABC avatar 🔼🔼	🕒 created_at 🔼🔼	ABC phone_number 🔼🔼
1	1	Иван Иванов	ivan@example.com	d131dd02c5e6eec4	31	/pictures/ivan.jpeg	2025-04-06 22:08:56.643	[NULL]

SELECT

```
SELECT id, name, email, created_at FROM users
WHERE email LIKE '%@southpark.com';
```







	123 id 	ABC name 	ABC email 	🕒 created_at 
1	6	Эрик Картман	eric@southpark.com	2025-04-06 22:30:11.684

SELECT

```
SELECT id, name, created_at FROM users  
ORDER BY id  
LIMIT 2 OFFSET 0;
```




SELECT

```
SELECT id, name, created_at FROM users
ORDER BY id
LIMIT 2 OFFSET 0;
```




	 id 	 name 	 created_at 
1	1	Иван Иванов	2025-04-06 22:08:56.643
2	2	Мария Петрова	2025-04-06 22:08:56.643

SELECT




```
SELECT id, name, created_at FROM users
ORDER BY id
LIMIT 2 OFFSET 0;
```

	 id	 name	 created_at
1	1	Иван Иванов	2025-04-06 22:08:56.643
2	2	Мария Петрова	2025-04-06 22:08:56.643

```
SELECT id, name, created_at FROM users
ORDER BY id
LIMIT 2 OFFSET 2;
```

	 id	 name	 created_at
1	5	Кек Лолов	2025-04-06 22:30:11.684
2	6	Эрик Картман	2025-04-06 22:30:11.684

GROUP BY

	123 id 	ABC name 	ABC company 
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark

GROUP BY

	123 id 🔼🔼	ABC name 🔼🔼	ABC company 🔼🔼
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark

```
SELECT company, count(*) FROM users  
GROUP BY company;
```





GROUP BY

	123 id 🔼🔼	ABC name 🔼🔼	ABC company 🔼🔼
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark

SELECT company, **count**(*) **FROM** users
GROUP BY company;

	ABC company 🔼🔼	123 count 🔼🔼
1	meme factory	1
2	southpark	1
3	hse	2

WITH

	123 id 	ABC name 	ABC avatar 	123 age 
1	2	Мария Петрова	[NULL]	[NULL]
2	1	Иван Иванов	/pictures/ivan.jpeg	31
3	5	Кек Лолов	[NULL]	[NULL]
4	6	Эрик Картман	/southpark/cartman.png	[NULL]


WITH

	123 id 	ABC name 	ABC avatar 	123 age 
1	2	Мария Петрова	[NULL]	[NULL]
2	1	Иван Иванов	/pictures/ivan.jpeg	31
3	5	Кек Лолов	[NULL]	[NULL]
4	6	Эрик Картман	/southpark/cartman.png	[NULL]

```
WITH users_with_avatar AS (  
    SELECT id, name, email  
    FROM users  
    WHERE avatar IS NOT NULL  
)  
SELECT * FROM users_with_avatar  
WHERE age > 20;
```

Что будет выведено в
результате запроса?

WITH

	123 id 	ABC name 	ABC avatar 	123 age 
1	2	Мария Петрова	[NULL]	[NULL]
2	1	Иван Иванов	/pictures/ivan.jpeg	31
3	5	Кек Лолов	[NULL]	[NULL]
4	6	Эрик Картман	/southpark/cartman.png	[NULL]

```
WITH users_with_avatar AS (  
    SELECT id, name, email  
    FROM users  
    WHERE avatar IS NOT NULL  
)  
SELECT * FROM users_with_avatar  
WHERE age > 20;
```



SQL Error [42703]: ERROR: column "age" does not exist
Position: 143

WITH

	123 id ↕	ABC name ↕	ABC avatar ↕	123 age ↕
1	2	Мария Петрова	[NULL]	[NULL]
2	1	Иван Иванов	/pictures/ivan.jpeg	31
3	5	Кек Лолов	[NULL]	[NULL]
4	6	Эрик Картман	/southpark/cartman.png	[NULL]

```
WITH users_with_avatar AS (  
    SELECT id, name, age, email  
    FROM users  
    WHERE avatar IS NOT NULL  
)  
SELECT * FROM users_with_avatar  
WHERE age > 20;
```

Что будет выведено в
результате запроса?





WITH

	123 id 🔽↑	ABC name 🔽↑	ABC avatar 🔽↑	123 age 🔽↑
1	2	Мария Петрова	[NULL]	[NULL]
2	1	Иван Иванов	/pictures/ivan.jpeg	31
3	5	Кек Лолов	[NULL]	[NULL]
4	6	Эрик Картман	/southpark/cartman.png	[NULL]

```
WITH users_with_avatar AS (  
    SELECT id, name, age, email  
    FROM users  
    WHERE avatar IS NOT NULL  
)  
SELECT * FROM users_with_avatar  
WHERE age > 20;
```







	123 id 🔽↑	ABC name 🔽↑	123 age 🔽↑	ABC email 🔽↑
1	1	Иван Иванов	31	ivan@example.com

MAX




	123 id 	ABC name 	 created_at 
1	2	Мария Петрова	2025-04-06 22:08:56.643
2	1	Иван Иванов	2025-04-06 22:08:56.643
3	5	Кек Лолов	2025-04-06 22:30:11.684
4	6	Эрик Картман	2025-04-06 22:30:11.684

```
SELECT MAX(created_at) AS last_registered
FROM users;
```

MAX

	 id 	 name 	 created_at 
1	2	Мария Петрова	2025-04-06 22:08:56.643
2	1	Иван Иванов	2025-04-06 22:08:56.643
3	5	Кек Лолов	2025-04-06 22:30:11.684
4	6	Эрик Картман	2025-04-06 22:30:11.684

```
SELECT MAX(created_at) AS last_registered  
FROM users;
```

	 last_registered 
1	2025-04-06 22:30:11.684

Оконные функции: AVG

	123 id 	ABC name 	123 age 	123 company_id 
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1

Оконные функции: AVG

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1

```
SELECT id, name, age, company_id, avg(age) OVER (PARTITION BY company_id)
FROM users;
```

	123 id	ABC name	123 age	123 company_id	123 avg
1	5	Кек Лолов	24	1	40.5
2	2	Мария Петрова	57	1	40.5
3	1	Иван Иванов	31	2	31
4	6	Эрик Картман	10	3	10

JOIN

В чем может быть проблема такой схемы?

	123 id 🔍 ⬆️ ⬆️	ABC name 🔍 ⬆️ ⬆️	ABC company 🔍 ⬆️ ⬆️
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark

В чем может быть проблема такой схемы?

	123 id ↕	ABC name ↕	ABC company ↕
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark



	123 id ↕	ABC name ↕	ABC company ↕
1	2	Мария Петрова	hse
2	1	Иван Иванов	meme factory
3	5	Кек Лолов	hse
4	6	Эрик Картман	southpark

Как исправить?

Как исправить?

```
CREATE TABLE companies (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO companies (name)  
VALUES  
( 'hse' ),  
( 'meme factory' ),  
( 'southpark' );
```

	123 id 	ABC name 
1	1	hse
2	2	meme factory
3	3	southpark

Как исправить?

```
CREATE TABLE companies (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO companies (name)  
VALUES  
( 'hse' ),  
( 'meme factory' ),  
( 'southpark' );
```

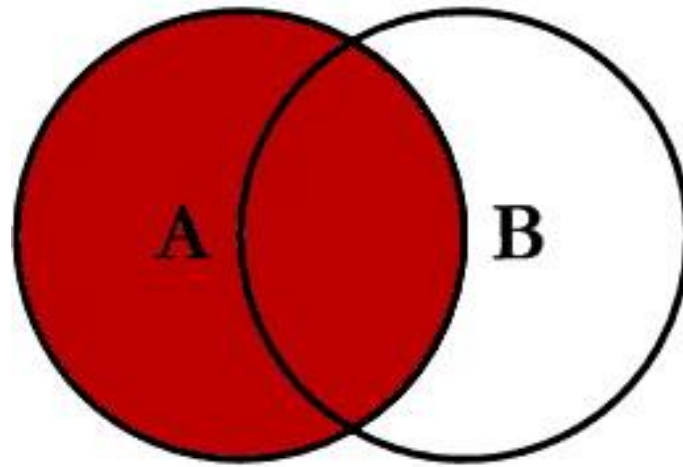
```
ALTER TABLE users  
ADD COLUMN company_id INTEGER;
```

```
ALTER TABLE users  
DROP COLUMN company;
```

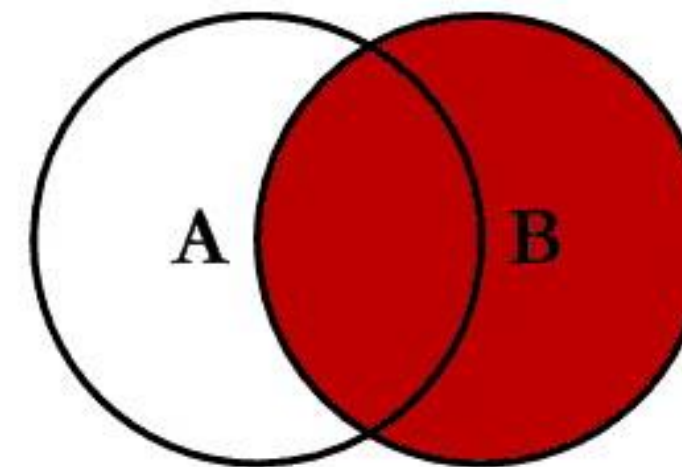
	123 id 🔍 ↕	ABC name 🔍 ↕
1	1	hse
2	2	meme factory
3	3	southpark

	123 id 🔍 ↕	ABC name 🔍 ↕	123 company_id 🔍 ↕
1	2	Мария Петрова	1
2	1	Иван Иванов	2
3	5	Кек Лолов	1
4	6	Эрик Картман	3

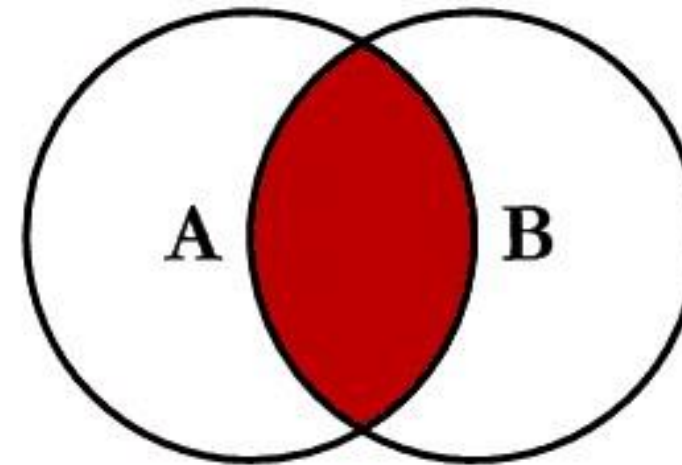
SQL JOINS



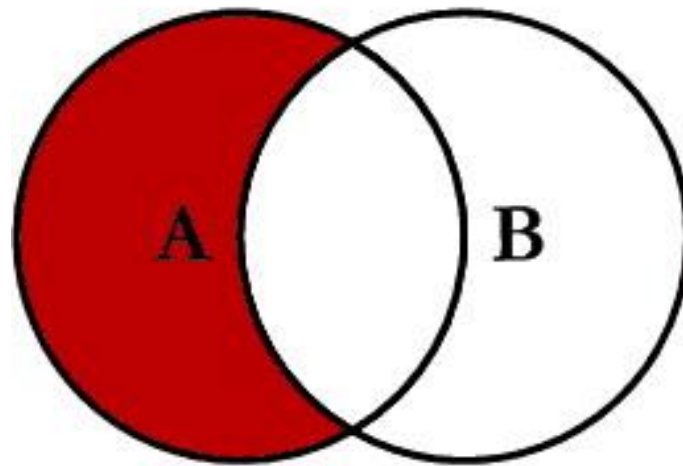
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



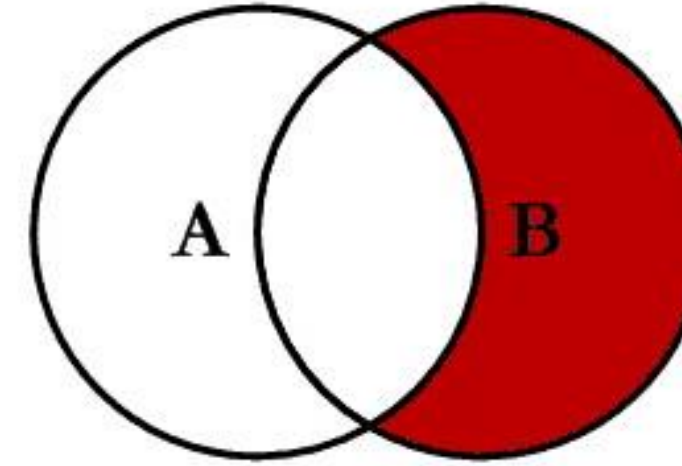
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



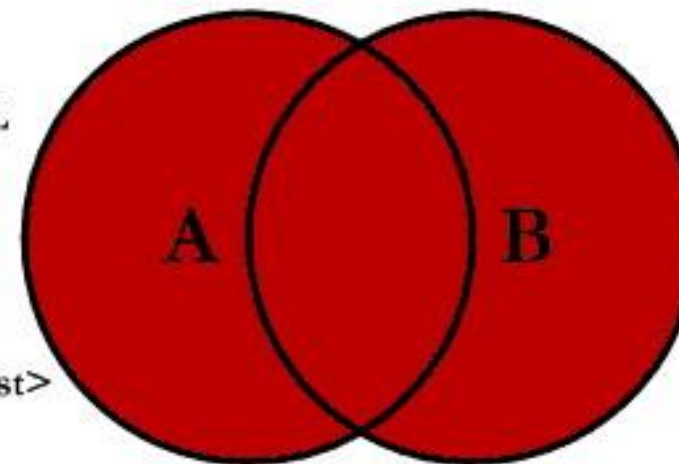
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



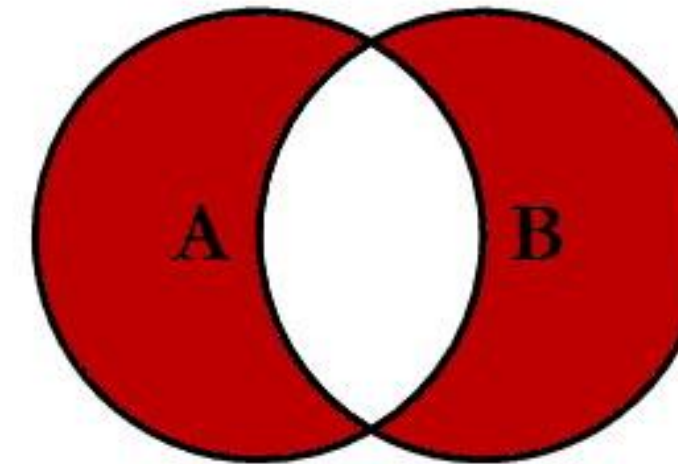
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

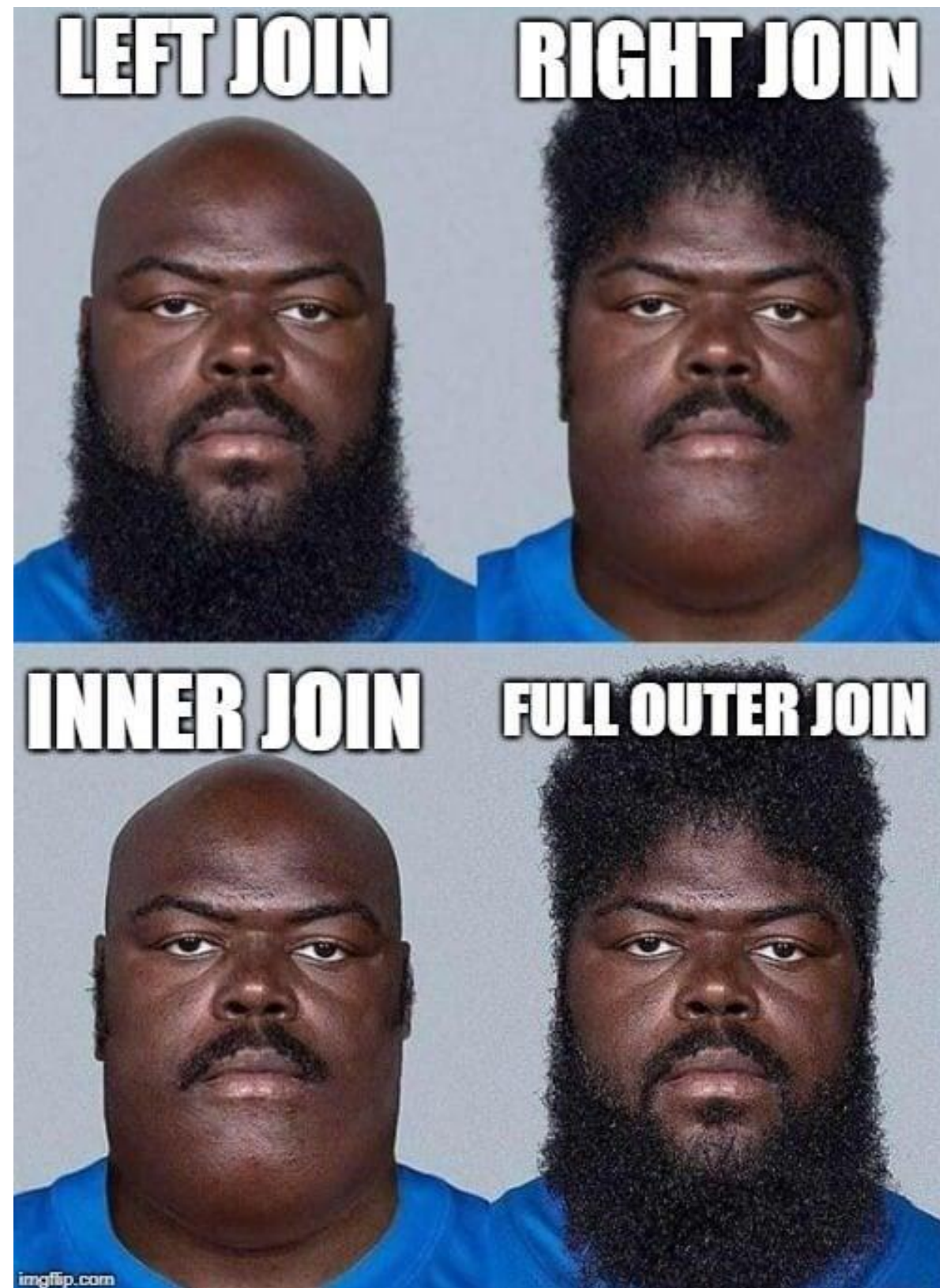


```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

SQL JOINS



Какие таблицы сейчас есть?

	123 id 🔼🔽	ABC name 🔼🔽	123 age 🔼🔽	123 company_id 🔼🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔼🔽	ABC name 🔼🔽
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

INNER JOIN

Показывает пользователей, у которых указана компания (company_id не NULL)

	123 id 🔼🔼	ABC name 🔼🔼	123 age 🔼🔼	123 company_id 🔼🔼
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔼🔼	ABC name 🔼🔼
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

INNER JOIN

Показывает пользователей, у которых указана компания (company_id не NULL)

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
INNER JOIN companies c
ON u.company_id = c.id;
```

	123 id	ABC user_name	ABC company_name
1	2	Мария Петрова	hse
2	5	Кек Лолов	hse
3	1	Иван Иванов	memefactory
4	6	Эрик Картман	southpark

LEFT JOIN

Показывает всех пользователей, даже если у них нет компании

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

LEFT JOIN

Показывает всех пользователей, даже если у них нет компании

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
LEFT JOIN companies c
ON u.company_id = c.id;
```

	123 id	ABC user_name	ABC company_name
1	2	Мария Петрова	hse
2	5	Кек Лолов	hse
3	1	Иван Иванов	memefactory
4	6	Эрик Картман	southpark
5	7	Симпл Димпл	[NULL]

RIGHT JOIN

Показывает все компании, даже если в них нет пользователей

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

RIGHT JOIN

Показывает все компании, даже если в них нет пользователей

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
RIGHT JOIN companies c
ON u.company_id = c.id;
```

	123 id	ABC user_name	ABC company_name
1	2	Мария Петрова	hse
2	5	Кек Лолов	hse
3	1	Иван Иванов	memefactory
4	6	Эрик Картман	southpark
5	[NULL]	[NULL]	school

FULL JOIN

Показывает все компании и всех пользователей (даже если нет связей)

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

FULL JOIN

Показывает все компании и всех пользователей (даже если нет связей)

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
FULL JOIN companies c
ON u.company_id = c.id;
```

	123 id 🔽	ABC user_name 🔽	ABC company_name 🔽
1	2	Мария Петрова	hse
2	5	Кек Лолов	hse
3	1	Иван Иванов	memefactory
4	6	Эрик Картман	southpark
5	[NULL]	[NULL]	school
6	7	Симпл Димпл	[NULL]

CROSS JOIN

Все возможные комбинации пользователей и компаний (редко используется)

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name,  
       c.name AS company_name  
FROM users u  
CROSS JOIN companies c;
```


CROSS JOIN

Все возможные комбинации пользователей и компаний (редко используется)

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name,  
       c.name AS company_name  
FROM users u  
CROSS JOIN companies c;
```

	123 id	ABC user_name	ABC company_name
1	1	Иван Иванов	hse
2	5	Кек Лолов	hse
3	6	Эрик Картман	hse
4	2	Мария Петрова	hse
5	7	Симпл Димпл	hse
6	1	Иван Иванов	meme factory
7	5	Кек Лолов	meme factory
8	6	Эрик Картман	meme factory
9	2	Мария Петрова	meme factory
10	7	Симпл Димпл	meme factory
11	1	Иван Иванов	southpark
12	5	Кек Лолов	southpark
13	6	Эрик Картман	southpark
14	2	Мария Петрова	southpark
15	7	Симпл Димпл	southpark
16	1	Иван Иванов	school
17	5	Кек Лолов	school
18	6	Эрик Картман	school
19	2	Мария Петрова	school
20	7	Симпл Димпл	school

JOIN + WHERE

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
FULL JOIN companies c
  ON u.company_id = c.id
WHERE age > 10;
```


JOIN + WHERE

	123 id 🔽	ABC name 🔽	123 age 🔽	123 company_id 🔽
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔽	ABC name 🔽
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

```
SELECT u.id, u.name AS user_name, c.name AS company_name
FROM users u
FULL JOIN companies c
  ON u.company_id = c.id
WHERE age > 10;
```

	123 id 🔽	ABC user_name 🔽	ABC company_name 🔽
1	1	Иван Иванов	meme factory
2	5	Кек Лолов	hse
3	2	Мария Петрова	hse

Как можно комбинировать функции?

	123 id 🔼🔼	ABC name 🔼🔼	123 age 🔼🔼	123 company_id 🔼🔼
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id 🔼🔼	ABC name 🔼🔼
1	1	hse
2	2	meme factory
3	3	southpark
4	4	school

Вывести компании, количество их сотрудников, средний возраст сотрудников и последнего присоединившегося сотрудника

Как можно комбинировать функции?

	123 id	ABC name	123 age	123 company_id
1	1	Иван Иванов	31	2
2	5	Кек Лолов	24	1
3	6	Эрик Картман	10	3
4	2	Мария Петрова	57	1
5	7	Симпл Димпл	9	[NULL]

	123 id	ABC name
1	1	hse
2	2	memefactory
3	3	southpark
4	4	school

Вывести компании, количество их сотрудников, средний возраст сотрудников и последнего присоединившегося сотрудника

SELECT

```
c.name AS company_name,  
COUNT(u.id) AS total_users,  
ROUND(AVG(u.age)) AS avg_age,  
(  
    SELECT u2.name  
    FROM users u2  
    WHERE u2.company_id = c.id  
    ORDER BY u2.created_at DESC  
    LIMIT 1  
) AS last_registered_user  
FROM companies c  
LEFT JOIN users u ON u.company_id = c.id  
GROUP BY c.id  
HAVING COUNT(u.id) > 0  
ORDER BY total_users DESC;
```

	ABC company_name	123 total_users	123 avg_age	ABC last_registered_user
1	hse	2	41	Кек Лолов
2	memefactory	1	31	Иван Иванов
3	southpark	1	10	Эрик Картман

AND THAT IS HOW YOU WRITE A SQL QUERY

ANY QUESTIONS?

Что мы сегодня узнали?

1. Рассмотрели группы, на которые делятся операции в БД
2. Научились создавать таблицы и наполнять их данными
3. Разобрались, как получать данные из таблиц и фильтровать их
4. Поняли, как комбинировать записи из разных таблиц

Что будет на следующей лекции?

1. Индексы
2. B-деревья
3. Как делать схему данных производительной?

Спасибо за внимание!

