

SELECT (Part 1)

Определение

- **SELECT** — оператор DML языка SQL, возвращающий набор данных (выборку) из базы данных, удовлетворяющих заданному условию;
- Оператор SELECT позволяет реализовать почти все операции реляционной алгебры: Объединение, Пересечение, Вычитание, **Декартово произведение**, **Выборка (ограничение)**, **Проекция**, **Соединение** (Переименование атрибутов → ALTER);
- В большинстве случаев, выборка осуществляется из одной или нескольких таблиц.
- По статистике является самым часто используемым оператором из всего SQL в реальных проектах.

Извлечение всех столбцов

- `SELECT * FROM contact`

	id_category	first_name	middle_name	last_name	age	created	updated	active
1	1	Иван	Иванович	Иванов	20	10/26/2016 11:12:01 AM	Null	<input checked="" type="checkbox"/>
2	1	Петр	Null	Петров	18	10/26/2016 11:12:21 AM	Null	<input checked="" type="checkbox"/>
3	2	Ирина	Викторовна	Николаева	22	10/26/2016 11:28:44 AM	10/26/2016 5:53:21 PM	<input checked="" type="checkbox"/>
4	1	Иван	Иванович	Иванов	21	10/26/2016 5:28:06 PM	Null	<input checked="" type="checkbox"/>

- `SELECT * FROM contact WHERE id = 1` (Выборка по строкам)

	id_category	first_name	middle_name	last_name	age	created	updated	active
1	1	Иван	Иванович	Иванов	20	10/26/2016 11:12:01 AM	Null	<input checked="" type="checkbox"/>

- `SELECT * FROM contact WHERE id < 3` (Выборка по строкам)

	id_category	first_name	middle_name	last_name	age	created	updated	active
1	1	Иван	Иванович	Иванов	20	10/26/2016 11:12:01 AM	Null	<input checked="" type="checkbox"/>
2	1	Петр	Null	Петров	18	10/26/2016 11:12:21 AM	Null	<input checked="" type="checkbox"/>

Проекция

- SELECT id, last_name, first_name, middle_name FROM contact

id	last_name	first_name	middle_name
1	Иванов	Иван	Иванович
2	Петров	Петр	Null
3	Николаева	Ирина	Викторовна

- SELECT id, last_name, first_name, middle_name FROM contact WHERE id = 1
(Проекция + Выборка по строкам)

id	last_name	first_name	middle_name
1	Иванов	Иван	Иванович

- SELECT id, last_name, first_name, middle_name FROM contact WHERE id < 3
(Проекция + Выборка по строкам)

id	last_name	first_name	middle_name
1	Иванов	Иван	Иванович
2	Петров	Петр	Null

Проекция (Дубликаты???)

- SELECT first_name, last_name, middle_name FROM contact

first_name	last_name	middle_name
Иван	Иванов	Иванович
Иван	Иванов	Иванович
Ирина	Николаева	Викторовна
Петр	Петров	Null

- SELECT **DISTINCT** first_name, last_name, middle_name FROM contact

first_name	last_name	middle_name
Иван	Иванов	Иванович
Ирина	Николаева	Викторовна
Петр	Петров	Null

Динамические столбцы

- SELECT first_name, last_name AS surname, 'Hello' AS txt, age + 100 AS age1, age > 18 AS adult FROM contact;

first_name	surname	txt	age1	adult
Иван	Иванов	Hello	120	<input checked="" type="checkbox"/>
Петр	Петров	Hello	118	<input type="checkbox"/>
Иван	Иванов	Hello	121	<input checked="" type="checkbox"/>
Ирина	Николаева	Hello	122	<input checked="" type="checkbox"/>

- SELECT last_name || ' ' || first_name || ' ' || middle_name as fio, length(last_name) as last_length, pow(age, 2) FROM contact

fio	last_length	pow
Иванов Иван Иванович	6	400
Null	6	324
Иванов Иван Иванович	6	441
Николаева Ирина Викторовна	9	484

- SELECT 2+2;

Использование функций

<https://www.postgresql.org/docs/9.5/static/functions.html>

(Эти функции не входят в стандарт SQL и являются специфичными для PostgreSQL сервера СУБД).

- `SELECT exp(2);`
- `SELECT sqrt(2);`
- `SELECT last_name || ' ' || first_name || ' ' || middle_name as fio, length(last_name) as last_length, pow(age, 2) FROM contact;`
- `SELECT age(now(), timestamp '1990-08-24');`
- `SELECT date_trunc('hour', now());`
- `SELECT date_trunc('year', now());`
- `SELECT date_trunc('hour', now()), power(2, 8), random(), repeat('Hello', 3) FROM contact;`

Объединение, Пересечение, Вычитание

```
/* create table copy as select * from contact */
```

- `SELECT * FROM contact UNION SELECT * FROM copy`
- `SELECT * FROM contact INTERSECT SELECT * FROM copy`
- `SELECT * FROM contact EXCEPT SELECT * FROM copy`

WHERE операторы

- `SELECT * FROM contact WHERE id > 0;`
- `SELECT * FROM contact WHERE id = 1;`
- `SELECT * FROM contact WHERE id <> 1;`
- `SELECT * FROM contact WHERE age < 30;`
- `SELECT * FROM contact WHERE id > 0 AND id < 3;`
- `SELECT * FROM contact WHERE id > 0 AND NOT id < 3;`
- `SELECT * FROM contact WHERE id >= 3 OR id <= 1;`
- `SELECT * FROM contact WHERE NOT (id >= 3 OR id <= 1);`
- `SELECT * FROM contact WHERE age BETWEEN 18 AND 20;`
- `SELECT * FROM contact WHERE first_name = 'Иван';`
- `SELECT * FROM contact WHERE created < now();`
- `SELECT * FROM contact WHERE updated IS NULL;`
- `SELECT * FROM contact WHERE updated IS NOT NULL;`
- `SELECT * FROM contact WHERE first_name LIKE 'И%';`
- `SELECT * FROM contact WHERE first_name LIKE '_ри%'`
- `SELECT * FROM contact WHERE first_name ILIKE 'и%';`
- `SELECT * FROM contact WHERE id IN (1, 3, 4, 6);`
- `/* NOT LIKE, NOT BETWEEN, NOT IN */`

Агрегатные функции

- **COUNT** – подсчитывает номера строк или не-NULL значения полей которые выбрал запрос;
 - **SUM** – подсчитывает арифметическую сумму всех выбранных значений данного поля;
 - **AVG** – подсчитывает усреднение всех выбранных значений данного поля;
 - **MAX** – вычисляет наибольшее из всех выбранных значений данного поля;
 - **MIN** – вычисляет наименьшее из всех выбранных значений данного поля.
-
- `SELECT count(id) FROM contact;`
 - `SELECT sum(age) FROM contact;`
 - `SELECT avg(age) FROM contact;`
 - `SELECT max(first_name) FROM contact;`
 - `SELECT min(first_name) FROM contact;`
 - `SELECT count(*) FROM contact;`
 - `SELECT count(DISTINCT first_name) FROM contact;`