

Шпаргалка по базовому SQL. Часть 1



SQL

SQL, or **Structured Query Language**, это язык взаимодействия с базами данных. Он позволяет отбирать данные и строить сложные отчеты. **SQL** используется практически во всех технологиях обработки данных.

ПРИМЕР ДАННЫХ

COUNTRY				
id	name	population	area	
1	France	66600000	640680	
2	Germany	80700000	357000	
...	

CITY				
id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
...

ЗАПРОСЫ К ОДНОЙ ТАБЛИЦЕ

Получить все поля из таблицы **country**:

```
SELECT *
FROM country;
```

Получить поля id и name из таблицы **city**:

```
SELECT id, name
FROM city;
```

Получить названия городов, отсортированных по алфавитному порядку:

```
SELECT name
FROM city
ORDER BY rating [ASC];
```

Получить названия городов, отсортированных в обратном алфавитном порядке:

```
SELECT name
FROM city
ORDER BY rating DESC;
```

АЛИАСЫ

ПОЛЯ

```
SELECT name AS city_name
FROM city;
```

ТАБЛИЦЫ

```
SELECT co.name, ci.name
FROM city AS ci
JOIN country AS co
    ON ci.country_id = co.id;
```

ФИЛЬТРАЦИЯ

ОПЕРАТОРЫ СРАВНЕНИЯ

Получить названия городов с рейтингом более 3:

```
SELECT name
FROM city
WHERE rating > 3;
```

Получить названия всех городов, кроме Berlin и Madrid:

```
SELECT name
FROM city
WHERE name != 'Berlin'
    AND name != 'Madrid';
```

ТЕКСТОВЫЕ ОПЕРАТОРЫ

Получить названия городов, с началом 'P' или окончанием 's':

```
SELECT name
FROM city
WHERE name LIKE 'P%'
    OR name LIKE '%s';
```

Получить названия городов, с началом на любую букву и окончанием на 'ublin' (например Dublin или Lublin):

```
SELECT name
FROM city
WHERE name LIKE '_ublin';
```

ДРУГИЕ ОПЕРАТОРЫ

Получить названия городов с населением от 500K до 5M:

```
SELECT name
FROM city
WHERE population BETWEEN 500000 AND 5000000;
```

Получить название городов, у которых не пропущен рейтинг:

```
SELECT name
FROM city
WHERE rating IS NOT NULL;
```

Получить названия городов, id стран которых равен одному из значений - 1, 4, 7, или 8:

```
SELECT name
FROM city
WHERE country_id IN (1, 4, 7, 8);
```

ЗАПРОСЫ К НЕСКОЛЬКИМ ТАБЛИЦАМ

INNER JOIN

JOIN (или **INNER JOIN**) возвращает строки, имеющие равные значения в обеих таблицах в соединяющих полях:

```
SELECT city.name, country.name
FROM city
[INNER] JOIN country
    ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	3	Iceland

LEFT JOIN

LEFT JOIN возвращает все строки левой таблицы с соответствующими строками правой таблицы. Если не нашлось соответствий, значения полей в правой таблице будут равны null:

```
SELECT city.name, country.name
FROM city
LEFT JOIN country
    ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL

RIGHT JOIN

RIGHT JOIN возвращает все строки правой таблицы с соответствующими строками левой таблицы. Если не нашлось соответствий, значения полей левой таблицы будут равны null:

```
SELECT city.name, country.name
FROM city
RIGHT JOIN country
    ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
NULL	NULL	NULL	3	Iceland

FULL JOIN

FULL JOIN (или **FULL OUTER JOIN**) возвращает все строки обеих таблиц - если соответствий не нашлось, значения в этих полях будут равны null:

```
SELECT city.name, country.name
FROM city
FULL [OUTER] JOIN country
    ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL
NULL	NULL	NULL	3	Iceland

CROSS JOIN

CROSS JOIN возвращает все возможные комбинации строк из обеих таблиц. Ниже приведены варианты синтаксиса:

```
SELECT city.name, country.name
FROM city
CROSS JOIN country;
```

```
SELECT city.name, country.name
FROM city, country;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
1	Paris	1	2	Germany
2	Berlin	2	1	France
2	Berlin	2	2	Germany

NATURAL JOIN

NATURAL JOIN может объединять таблицы используя поля с одинаковыми названиями:

```
SELECT city.name, country.name
FROM city
NATURAL JOIN country;
```

CITY			COUNTRY	
country_id	id	name	name	id
6	6	San Marino	San Marino	6
7	7	Vatican City	Vatican City	7
5	9	Greece	Greece	9
10	11	Monaco	Monaco	10

NATURAL JOIN используется для соотнесения значений полей: **city.id**, **city.name**, **country.id**, **country.name**. **NATURAL JOIN** на практике используется довольно редко.