SQL Queries Cheatsheet

*Learn it all at **academy.vertabelo.com***

Vertabelo ACADEMY

QUERYING SINGLE TABLE

Sample Data

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
...	...	...	...	...

-- all columns from table country

```
SELECT * FROM country;
```

-- columns id and name from table city

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

-- names of cities which...

```
SELECT name FROM city WHERE...
```

-- have a rating above 3

```
        rating > 3;
```

-- start with a 'P' or end with an 's'

```
        name LIKE 'P%' OR name LIKE '%s';
```

-- start with any letter followed by 'ublin'
-- (like Dublin in Ireland and Lublin in Poland)

```
        name LIKE '_ublin';
```

-- are not Berlin or Madrid

```
        name != 'Berlin' AND name != 'Madrid';
```

-- have a population between 1M and 5M

```
        population BETWEEN 1000000 AND 5000000;
```

-- do not have a missing rating value

```
        rating IS NOT NULL;
```

-- are in countries 1, 4, 7 or 8

```
        country_id IN (1,4,7,8);
```

QUERYING MULTIPLE TABLES

-- city names with their country names

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

-- fetch only cities with non-null matching countries

```
city (INNER) JOIN country
```

CITY

id	name	country_id
1	Paris	1
2	Berlin	2
3	Warsaw	4

COUNTRY

id	name
1	France
2	Germany
3	Iceland

-- fetch all cities even if no matching country exists

```
city LEFT (OUTER) JOIN country
```

CITY

id	name	country_id
1	Paris	1
2	Berlin	2
3	Warsaw	4

COUNTRY

id	name
1	France
2	Germany
null	null

-- fetch all countries even if no matching cities exist

```
city RIGHT (OUTER) JOIN country
```

CITY

id	name	country_id
1	Paris	1
2	Berlin	2
null	null	null

COUNTRY

id	name
1	France
2	Germany
3	Iceland

-- fetch all cities and all countries even if no matching
-- values exist in the other table

```
city FULL (OUTER) JOIN country
```

CITY

id	name	country_id
1	Paris	1
2	Berlin	2
3	Warsaw	4
null	null	null

COUNTRY

id	name
1	France
2	Germany
null	null
3	Iceland

SQL Queries Cheatsheet

*Learn it all at **academy.vertabelo.com***

AGGREGATION/GROUPING

-- cities with highest rating first

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

-- number of all cities

```
SELECT COUNT(*) FROM city;
```

-- number of cities with non-null rating

```
SELECT COUNT(rating) FROM city;
```

-- number of distinctive country values

```
SELECT COUNT(DISTINCT country_id)
FROM city;
```

-- smallest and greatest country population

```
SELECT MIN(population), MAX(population)
FROM country;
```

-- total population of cities in respective countries

```
SELECT country_id, SUM(population)
FROM city GROUP BY country_id;
```

-- average rating for cities in
-- respective countries if the average
-- is above 3.0

```
SELECT country_id, AVG(rating) FROM
city GROUP BY country_id
HAVING AVG(rating) > 3;
```

SUBQUERIES

SINGLE VALUE

-- cities with the same rating as Paris

```
SELECT name FROM city
WHERE rating = (SELECT rating
      FROM city
      WHERE name = 'Paris');
```

MULTIPLE VALUES

-- cities in countries that have
-- a population above 20M

```
SELECT name FROM city
WHERE country_id IN
      (SELECT country_id FROM country
      WHERE population > 20000000);
```

CORRELATED

-- cities with population greater than
-- average population in the country

```
SELECT * FROM city main_city
WHERE population >
      (SELECT AVG(population)
      FROM city average_city
      WHERE average_city.country_id =
      main_city.country_id);
```

-- countries that have at least one city

```
SELECT name FROM country
WHERE EXISTS (SELECT * FROM city
      WHERE country_id = country.id);
```

SET OPERATIONS

CYCLING

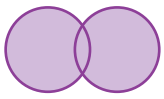
id	name	country
1	YK	DE
2	ZG	DE
3	WT	PL
...	...	...

SKATING

id	name	country
1	YK	DE
2	DF	DE
3	AK	PL
...	...	...

-- German cyclers together with German
-- skaters (ALL shows repetitions)

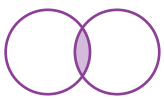
```
SELECT name FROM cycling
WHERE country = 'DE'
UNION (ALL)
SELECT name FROM skating
WHERE country = 'DE';
```



UNION

-- German cyclers that are also German
-- skaters at the same time

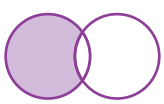
```
SELECT name FROM cycling
WHERE country = 'DE'
INTERSECT
SELECT name FROM skating
WHERE country = 'DE';
```



INTERSECT

-- German cyclers unless they are also
-- German skaters at the same time

```
SELECT name FROM cycling
WHERE country = 'DE'
EXCEPT/MINUS
SELECT name FROM skating
WHERE country = 'DE';
```



EXCEPT