Note: Keep in mind that SQL isn't case sensitive

<u>Database name = Sample Data</u>

Table 1 = Country

id	name	population	area
1	france	66600000	640680
2	Germany	80700000	357000
3	Iceland	376248	103125

Table 2 = City

id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
3	Warsaw	4	1863056	6

Select Operation

→ SELECT * FROM country

id	name	population	area
1	france	66600000	640680
2	Germany	80700000	357000
3	Iceland	376248	103125

 $\rightarrow \ \ \text{SELECT * FROM city}$

id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
3	Warsaw	4	1863056	6

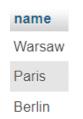
Selecting according to ascending order

- → SELECT name
- → FROM city
- \rightarrow ORDER BY rating ASC;



Selecting according to descending order

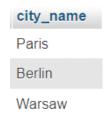
- → SELECT name
- \rightarrow FROM city
- \rightarrow ORDER by DESC;



ALIASES

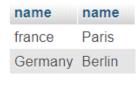
♦ Columns

SELECT name as city_name from city;



♦ Tables

SELECT co.name, ci.name FROM city as ci JOIN country as co on ci.country_id = co.id;



• Filtering the OUTPUT

- > Comparison operators
- fetching the city having rating >4

SELECT name FROM city WHERE rating>4;



• fetching name of cities that are neither Berlin nor Madrid

SELECT name FROM city WHERE name != 'berlin' AND name != 'Madrid';



- > Text Operators
- Fetching the name of cities whose name start with 'P'



• Fetching the name of cities whose name ends with 'N'



• Fetching the name of cities whose name starts with 'P' and ends with 's'

SELECT name FROM city WHERE name LIKE 'p%' or name LIKE '%s';



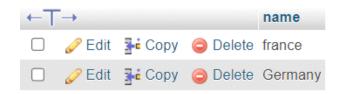
• Fetching the name of cities that start with some letter followed by some (e.g.: _aris in Paris , __rmany, so on)

SELECT name FROM city WHERE name LIKE '__ris';



- Other operators
- Fetching the name of country whose area is greater than 300,000 km^2.

SELECT name FROM country WHERE area > 300000;

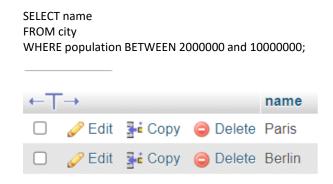


• Fetching the name of country whose population greater than 1,000,000

SELECT name FROM country WHERE population > 1000000;

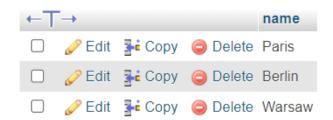


• Fetching the name of cities whose population is between 2,000,000 and 10,000,000.



• Fetching the name of cities whose rating is not null.

SELECT name FROM city WHERE rating is not null;



• fetching the name of cities with the help of IDs

SELECT name FROM city WHERE country_id IN (1,3);

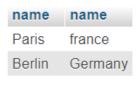


Querying multiple tables

> INNER JOIN

returns the matching values from both the tables

SELECT city.name, country.name from city INNER JOIN country
ON city.country_id = country.id;

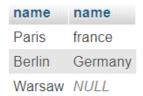


LEFT JOIN

returns all rows from the left table with the corresponding values from right table. if there's extra rows in left, and not in right then in right It will be null.

SELECT city.name, country.name FROM city

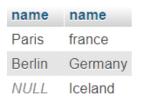
LEFT JOIN country
ON city.country_id = country.id;



RIGHT JOIN

Returns all rows from the right table with corresponding from the left table. If there's extra rows in right, and not in left then in left It will be null.

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



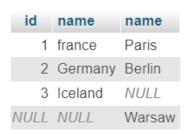
> FULL JOIN

returns all the values from the both the tables and if there's no value for the corresponding cell, it returns null.

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

UNION

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



> CROSS JOIN

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

name	name
Paris	france
Berlin	france
Warsaw	france
Paris	Germany
Berlin	Germany
Warsaw	Germany
Paris	Iceland
Berlin	Iceland
Warsaw	Iceland