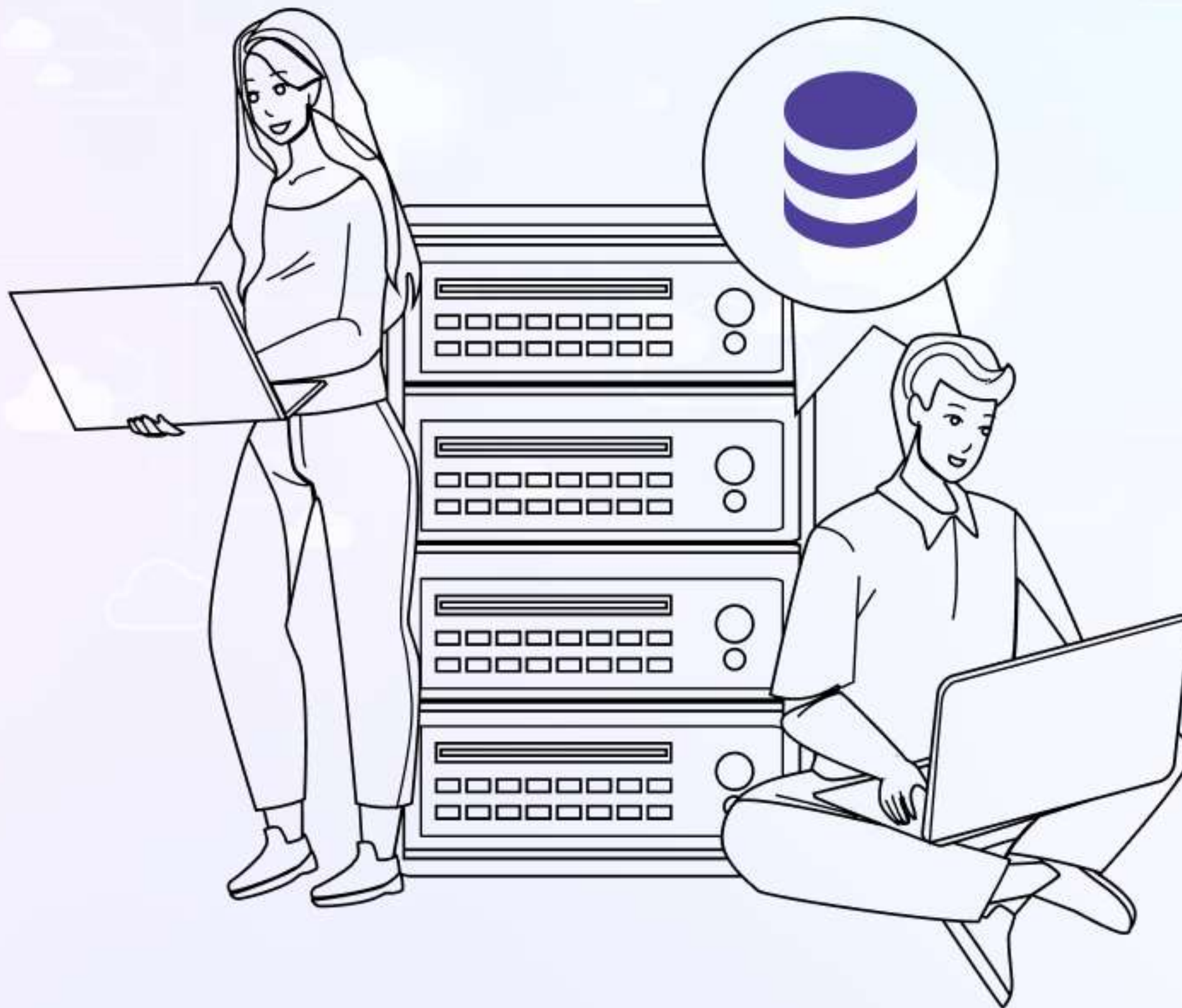


SQL BASICS

Cheat Sheet



SQL

SQL, or Structured Query Language, is a language to talk to databases. It allows you to select specific data and to build complex reports. Today, SQL is a universal language of data. It is used in practically all technologies that process data.

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
...



QUERYING SINGLE TABLE

Fetch all columns from the country table:

```
SELECT *  
FROM country;
```

Fetch id and name columns from the city table:

```
SELECT name  
FROM city;
```

Fetch city names sorted by the rating column in the default ASCending order:

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

Fetch city names sorted by the rating column in the DESCending order:

```
SELECT name  
FROM city  
ORDER BY rating 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

Fetch names of cities that have a rating above 3:

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

Fetch names of cities that are neither Berlin nor Madrid:

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

TEXT OPERATORS

Fetch names of cities that start with a 'P' or end with an 's':

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



Fetch names of cities that start with any letter followed by 'ublin' (like Dublin in Ireland or Lublin in Poland):

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

TEXT OPERATORS

Fetch names of cities that have a population between 500K and 5M:

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

Fetch names of cities that don't miss a rating value:

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

Fetch names of cities that are in countries with IDs 1, 4, 7, or 8:

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



QUERYING MULTIPLE TABLES

INNER JOIN

JOIN (or explicitly **INNER JOIN**) returns rows that have matching values in both tables.

```
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 return all rows from the left table with corresponding rows from the right table. if there's no matching row, NULLS are returned as values from the second table.




```
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 return all rows from the right table with corresponding rows from the left table. if there's no matching row, NULLS are returned as values from the left table.

```
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



**FOLLOW US
FOR MORE SUCH CONTENT**

