datacamp

SQL for Data Science

SQL Basics Cheat Sheet

Learn SQL online at www.DataCamp.com

What is SQL?

SQL stands for "structured query language". It is a language used to query, analyze, and manipulate data from databases. Today, SQL is one of the most widely used tools in data.

> The different dialects of SQL

Although SQL languages all share a basic structure, some of the specific commands and styles can differ slightly. Popular dialects include MySQL, SQLite, SQL Server, Oracle SQL, and more. PostgreSQL is a good place to start —since it's close to standard SQL syntax and is easily adapted to other dialects.

> Sample Data

Throughout this cheat sheet, we'll use the columns listed in this sample table of `airbnb_listings`

airbnb_listings				
id	city	country	number_of_rooms	year_listed
1	Paris	France	5	2018
2	Tokyo	Japan	2	2017
3	New York	USA	2	2022

> Querying tables

1. Get all the columns from a table

```
SELECT *
FROM airbnb_listings;
```

2. Return the `city` column from the table

```
SELECT city
FROM airbnb_listings;
```

3. Get the `city` and `year_listed` columns from the table

```
SELECT city, year_listed
FROM airbnb_listings;
```

4. Get the listing `id`, `city`, ordered by the `number_of_rooms` in ascending order

```
SELECT id, city
FROM airbnb_listings
ORDER BY number_of_rooms ASC;
```

5. Get the listing `id`, `city`, ordered by the `number_of_rooms` in descending order

```
SELECT id, city
FROM airbnb_listings
ORDER BY number_of_rooms DESC;
```

6. Get the first 5 rows from the `airbnb_listings` table

```
SELECT *
FROM airbnb_listings
LIMIT 5;
```

7. Get a unique list of cities where there are listings

```
SELECT DISTINCT city
FROM airbnb_lisitings;
```

> Filtering Data

Filtering on numeric columns

1. Get all the listings where `number_of_rooms` is more or equal to 3

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms >= 3;
```

2. Get all the listings where `number_of_rooms` is more than 3

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms > 3;
```

3. Get all the listings where `number_of_rooms` is exactly equal to 3

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms = 3;
```

4. Get all the listings where `number_of_rooms` is lower or equal to 3

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms <= 3;
```

5. Get all the listings where `number_of_rooms` is lower than 3

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms < 3;
```

6. Get all the listings with 3 to 6 rooms

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms BETWEEN 3 AND 6;
```

Filtering on text columns

7. Get all the listings that are based in 'Paris'

```
SELECT *
FROM airbnb_listings
WHERE city = 'Paris';
```

8. Get the listings based in the 'USA' and in 'France'

```
SELECT *
FROM airbnb_listings
WHERE country IN ('USA', 'France');
```

9. Get all the listings where the city starts with 'j' and where the city does not end in 't'

```
SELECT *
FROM airbnb_listings
WHERE city LIKE 'j%' AND city NOT LIKE '%t';
```

Filtering on multiple columns

10. Get all the listings in `Paris` where `number_of_rooms` is bigger than 3

```
SELECT *
FROM airbnb_listings
WHERE city = 'Paris' AND number_of_rooms > 3;
```

11. Get all the listings in `Paris` OR the ones that were listed after 2012

```
SELECT *
FROM airbnb_listings
WHERE city = 'Paris' OR year_listed > 2012;
```

Filtering on missing data

12. Return the listings where `number_of_rooms` is missing

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms IS NULL;
```

13. Return the listings where `number_of_rooms` is not missing

```
SELECT *
FROM airbnb_listings
WHERE number_of_rooms IS NOT NULL;
```

> Aggregating Data

Simple aggregations

1. Get the total number of rooms available across all listings

```
SELECT SUM(number_of_rooms)
FROM airbnb_listings;
```

2. Get the average number of rooms per listing across all listings

```
SELECT AVG(number_of_rooms)
FROM airbnb_listings;
```

3. Get the listing with the highest number of rooms across all listings

```
SELECT MAX(number_of_rooms)
FROM airbnb_listings;
```

4. Get the listing with the lowest number of rooms across all listings

```
SELECT MIN(number_of_rooms)
FROM airbnb_listings;
```

Grouping, filtering, and sorting

5. Get the total number of rooms for each country

```
SELECT country, SUM(number_of_rooms)
FROM airbnb_listings
GROUP BY country;
```

6. Get the average number of rooms for each country

```
SELECT country, AVG(number_of_rooms)
FROM airbnb_listings
GROUP BY country;
```

7. Get the listing with the maximum number of rooms per country

```
SELECT country, MAX(number_of_rooms)
FROM airbnb_listings
GROUP BY country;
```

8. Get the listing with the lowest amount of rooms per country

```
SELECT country, MIN(number_of_rooms)
FROM airbnb_listings
GROUP BY country;
```

9. For each country, get the average number of rooms per listing, sorted by ascending order

```
SELECT country, AVG(number_of_rooms) AS avg_rooms
FROM airbnb_listings
GROUP BY country
ORDER BY avg_rooms ASC;
```

10. For Japan and the USA, get the average number of rooms per listing in each country

```
SELECT country, AVG(number_of_rooms)
FROM airbnb_listings
WHERE country IN ('USA','Japan');
GROUP BY country;
```

11. Get the number of cities per country, where there are listings

```
SELECT country, COUNT(city) AS number_of_cities
FROM airbnb_listings
GROUP BY country;
```

12. Get all the years where there were more than 100 listings per year

```
SELECT year_listed
FROM airbnb_listings
GROUP BY year_listed
HAVING COUNT(id) > 100;
```