

SQL Cheat Sheet



DATA with BARAA
datawithbaraa.com
[SQL Youtube Tutorials](#)

QUERYING DATA

SELECT

Retrieve all data and columns from customers

```
SELECT *  
FROM customers
```

Retrieve ONLY first name and country of all customers

```
SELECT  
    first_name,  
    country  
FROM customers
```

DISTINCT

List all countries of all customers without duplicates

```
SELECT DISTINCT country  
FROM customers
```

ORDER BY

Retrieve all customers where the result is sorted by score (smallest first)

```
SELECT *  
FROM customers  
ORDER BY score ASC
```

```
SELECT *  
FROM customers  
ORDER BY score
```

Retrieve all customers where the result is sorted by score (highest first)

```
SELECT *  
FROM customers  
ORDER BY score DESC
```

Retrieve all customers, sorting the result by country (alphabetically) and then by score (highest first)

```
SELECT *  
FROM customers  
ORDER BY country ASC, score DESC
```

FILTERING DATA

WHERE

List only german customers

```
SELECT *  
FROM customers  
WHERE customers = 'Germany'
```

Comparison Operators

=	Equal
!= or <>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Find all customers whose score is greater than 500

```
SELECT *  
FROM customers  
WHERE score > 500
```

Find all customers whose score is less than 500

```
SELECT *  
FROM customers  
WHERE score < 500
```

Find all customers whose score is less than or equal to 500

```
SELECT *  
FROM customers  
WHERE score <= 500
```

Find all customers whose score is higher than or equal to 500

```
SELECT *  
FROM customers  
WHERE score >= 500
```

Find all non-german customers

```
SELECT *  
FROM customers  
WHERE country != 'Germany'
```

Logical Operators

AND	Return True if both conditions are True
OR	Return True if one of conditions is True
NOT	Reverse the result of any Boolean operator
IN	Return True if a value is in a set of values
BETWEEN	Return True if a value falls within a specific range
LIKE	Return True if a value matches a pattern

AND

Find all customers who come from Germany AND whose score is less than 400

```
SELECT *  
FROM customers  
WHERE country = 'Germany'  
AND score <= 500
```

SQL Cheat Sheet



DATA with BARAA
datawithbaraa.com
[SQL Youtube Tutorials](#)

OR

Find all customers who come from Germany OR whose score is less than 400

```
SELECT *  
FROM customers  
WHERE country = 'Germany'  
OR score < 400
```

NOT

Find all customers whose score is NOT less than 400

```
SELECT *  
FROM customers  
WHERE NOT score < 400
```

BETWEEN

Find all customers whose score falls in the range between 100 and 500

```
SELECT *  
FROM customers  
WHERE score BETWEEN 100 AND 500
```

Or we can solve the same task with the following SQL

```
SELECT *  
FROM customers  
WHERE score >= 100 AND score <= 500
```

IN

Find all customers whose ID is equal to 1, 2 or 5

```
SELECT *  
FROM customers  
WHERE customer_id IN (1,2,5)
```

LIKE

Find all customers whose first name starts with M

```
SELECT *  
FROM customers  
WHERE first_name LIKE 'M%'
```

Find all customers whose first name ends with n

```
SELECT *  
FROM customers  
WHERE first_name LIKE '%n'
```

Find all customers whose first name contains r

```
SELECT *  
FROM customers  
WHERE first_name LIKE '%r%'
```

Find all customers whose first name contains r in 3d position

```
SELECT *  
FROM customers  
WHERE first_name LIKE '__r%'
```

ALIASES

Columns

```
SELECT customer_id AS cid  
FROM customers
```

Tables

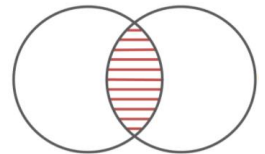
```
SELECT c.customer_id AS cid  
FROM customers AS c
```

JOINING TABLES

INNER JOIN

List customer ID, first name, order ID, quantity. Exclude the customers who have not placed any orders

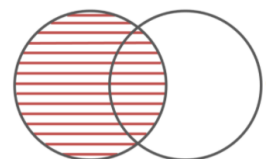
```
SELECT  
    c.customer_id,  
    c.first_name,  
    o.order_id,  
    o.quantity  
FROM customers AS c  
INNER JOIN orders AS o  
ON c.customer_id = o.customer_id
```



LEFT JOIN

List customer ID, first name, order ID, quantity. Include the customers who have not placed any orders

```
SELECT  
    c.customer_id,  
    c.first_name,  
    o.order_id,  
    o.quantity  
FROM customers AS c  
LEFT JOIN orders AS o  
ON c.customer_id = o.customer_id
```



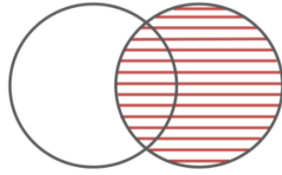


RIGHT JOIN

List customer ID, first name, order ID, quantity. Include all orders, regardless of whether there is a matching customer

SELECT

```
c.customer_id,  
c.first_name,  
o.order_id,  
o.quantity
```



FROM customers **AS** c

RIGHT JOIN orders **AS** o

ON c.customer_id = o.customer_id

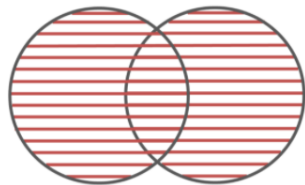
FULL JOIN

List customer ID, first name, order ID, quantity. Include all customers and all orders.

All databases that support FULL JOIN (MySQL doesn't support it)

SELECT

```
c.customer_id,  
c.first_name,  
o.order_id,  
o.quantity
```



FROM customers **AS** c

FULL JOIN orders **AS** o

ON c.customer_id = o.customer_id

Workaround for databases that don't support FULL JOIN
Like MySQL

SELECT

```
c.customer_id,  
c.first_name,  
o.order_id,  
o.quantity
```

FROM customers **AS** c

LEFT JOIN orders **AS** o

ON c.customer_id = o.customer_id

UNION

SELECT

```
c.customer_id,  
c.first_name,  
o.order_id,  
o.quantity
```

FROM customers **AS** c

RIGHT JOIN orders **AS** o

ON c.customer_id = o.customer_id

UNION

List first name, last name and country of all persons from customers and employees

SELECT

```
first_name,  
last_name,  
country
```

FROM customers

UNION ALL

SELECT

```
first_name,  
last_name,  
emp_country
```

FROM orders

List first name, last name and country of all persons from customers and employees without duplicates

SELECT

```
first_name,  
last_name,  
country
```

FROM customers

UNION

SELECT

```
first_name,  
last_name,  
emp_country
```

FROM orders

SQL Cheat Sheet



DATA with BARAA
datawithbaraa.com
[SQL Youtube Tutorials](#)

Aggregate Functions

COUNT()	Returns the number of items in a set
SUM()	Returns the sum of all or distinct values in a set
AVG()	Returns the average of a set
MAX()	Returns the minimum value in a set
MIN()	Returns the maximum value in a set

COUNT()

Find the total number of customers

```
SELECT COUNT(*) AS total_customers
FROM customers
```

SUM()

Find the total quantity of all orders

```
SELECT SUM(quantity) AS sum_quantity
FROM orders
```

AVG()

Find the average score of all customers

```
SELECT AVG(score) AS avg_score
FROM orders
```

MAX() MIN()

Find the highest score of all customers

```
SELECT MAX(score) AS max_score
FROM customers
```

Find the lowest score of all customers

```
SELECT LOW(score) AS low_score
FROM customers
```

GROUP BY

Find the total number of customers for each country

```
SELECT
    COUNT(*) AS total_customers,
    country
FROM Customers
GROUP BY country
```

HAVING

Find the total number of customers for each country and only include countries that have more than 1 customer

```
SELECT
    COUNT(*) AS total_customers,
    country
FROM Customers
GROUP BY Country
HAVING COUNT(*) > 1
```

String Functions

CONCAT()	Concatenating two or more string values
LOWER()	Converts a string to lowercase
UPPER()	Converts a string to uppercase
TRIM()	Removes leading and trailing spaces
LENGTH()	Returns the length of a string
SUBSTRING()	Returns a substring from string

CONCAT()

List all customer names, where the name is combination of first name and last name

```
SELECT
    CONCAT(first_name, '-', lastname) AS
customer_name
FROM customers
```

LOWER()

List the first name of all customers in lowercase

```
SELECT
    LOWER(first_name) AS low_first_name
FROM customers
```

UPPER()

List the first name of all customers in lowercase

```
SELECT
    UPPER(first_name) AS up_first_name
FROM customers
```

TRIM()

List the last name of all customers and remove all the white spaces in the names

```
SELECT
    TRIM(last_name) AS trim_last_name
FROM customers
```

LENGTH()

Find the length of the last name of all customers

```
SELECT
    LENGTH(last_name) AS len_last_name
FROM customers
```

SUBSTRING()

Subtract 3 characters from the last name of all customers, starting from the 2d position

```
SELECT
    SUBSTRING(last_name, 2, 3) AS
sub_last_name
FROM customers
```

SQL Cheat Sheet



DATA with BARAA
datawithbaraa.com
[SQL Youtube Tutorials](#)

Modifying Data

INSERT

Insert new customer Anna Nixon from UK

```
INSERT INTO customers  
(first_name, last_name, country)  
VALUES ('Anna', 'Nixon', 'UK')
```

UPDATE

Change the country of customer ID 7 to Germany

```
UPDATE customers  
SET country = 'Germany'  
WHERE customer_id = 7
```

Change the score of the customer Anna to 100 and change her country from UK to USA

```
UPDATE customers  
SET country = 'USA',  
    country = 100  
WHERE customer_id = 6
```

DELETE

Delete both customers Anna and Max from our database

```
DELETE FROM customers  
WHERE customer_id IN (6,7)
```

TRUNCATE

Delete all customers from our database

```
TRUNCATE customers
```

Defining Data

CREATE

Create new SQL table called Persons with 4 columns: ID, person name, birth date, and phone

```
CREATE TABLE persons (  
id INT PRIMARY_KEY AUTO_INCREMENT,  
Person_name VARCHAR(50) NOT NULL,  
birth_date DATE,  
phone VARCHAR(15) NOT NULL UNIQUE  
)
```

ALTER

Add new column called email to table Persons

```
ALTER TABLE persons  
ADD email VARCHAR(15) NOT NULL
```

DROP

Delete the new table Persons from our database

```
DROP TABLE persons
```

Subqueries

IN

Find all orders placed from customers whose score higher than 500 using customer_id

```
SELECT *  
FROM orders  
WHERE customer_id IN (  
    SELECT customer_id  
    FROM customers  
    WHERE score > 500)
```

IN

Find all orders placed from customers whose score higher than 500 using customer_id

```
SELECT *  
FROM orders AS o  
WHERE EXISTS (  
    SELECT 1  
    FROM customers AS c  
    WHERE c.customer_id = o.customer_id  
    AND score > 500)
```