
SQL Cheat Sheet

Sir. Rohail Qamar



1st Edition

<i>Basics</i>	5
<i>Comments</i>	5
<i>SELECT Clause</i>	5
<i>WHERE Clause</i>	6
<i>Logical Operators</i>	6
<i>IN Operator</i>	7
<i>BETWEEN Operator</i>	7
<i>LIKE Operator</i>	7
<i>REGEXP Operator</i>	7
<i>IS NULL Operator</i>	8
<i>ORDER BY Clause</i>	8
<i>LIMIT Clause</i>	8
<i>Inner Joins</i>	9
<i>Outer Joins</i>	9
<i>USING Clause</i>	9
<i>Cross Joins</i>	9
<i>Unions</i>	10
<i>Inserting Data</i>	10
<i>Want to Become a SQL Expert?</i>	10

Basics

```
USE sql_store;
```

```
SELECT *  
FROM customers  
WHERE state = 'CA'  
ORDER BY first_name  
LIMIT 3;
```

- SQL is **not** a case-sensitive language.
- In MySQL, every statement must be terminated with a semicolon.

Comments

We use comments to add notes to our code.

```
-- This is a comment and it won't get executed.
```

SELECT Clause

```
-- Using expressions
```

```
SELECT (points * 10 + 20) AS discount_factor  
FROM customers
```

Order of operations:

- Parenthesis
- Multiplication / division
- Addition / subtraction

```
-- Removing duplicates
```

```
SELECT DISTINCT state  
FROM customers
```

WHERE Clause

We use the WHERE clause to filter data.

Comparison operators:

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

Logical Operators

```
-- AND (both conditions must be True)
SELECT *
FROM customers
WHERE birthdate > '1990-01-01' AND points > 1000
```

```
-- OR (at least one condition must be True)
SELECT *
FROM customers
WHERE birthdate > '1990-01-01' OR points > 1000
```

```
-- NOT (to negate a condition)
SELECT *
FROM customers
WHERE NOT (birthdate > '1990-01-01')
```

IN Operator

```
-- Returns customers in any of these states: VA, NY, CA
SELECT *
FROM customers
WHERE state IN ('VA', 'NY', 'CA')
```

BETWEEN Operator

```
SELECT *
FROM customers
WHERE points BETWEEN 100 AND 200
```

LIKE Operator

```
-- Returns customers whose first name starts with b
SELECT *
FROM customers
WHERE first_name LIKE 'b%'
```

- %: any number of characters
- _: exactly one character

REGEXP Operator

```
-- Returns customers whose first name starts with a
SELECT *
FROM customers
WHERE first_name REGEXP '^a'
```

- ^: beginning of a string
- \$: end of a string
- |: logical OR
- [abc]: match any single characters
- [a-d]: any characters from a to d

More Examples

```
-- Returns customers whose first name ends with EY or ON
WHERE first_name REGEXP 'ey$|on$'

-- Returns customers whose first name starts with MY
-- or contains SE
WHERE first_name REGEXP '^my|se'

-- Returns customers whose first name contains B followed by
-- R or U
WHERE first_name REGEXP 'b[ru]'
```

IS NULL Operator

```
-- Returns customers who don't have a phone number
SELECT *
FROM customers
WHERE phone IS NULL
```

ORDER BY Clause

```
-- Sort customers by state (in ascending order), and then
-- by their first name (in descending order)
SELECT *
FROM customers
ORDER BY state, first_name DESC
```

LIMIT Clause

```
-- Return only 3 customers
SELECT *
FROM customers
LIMIT 3
```



```
-- Skip 6 customers and return 3
SELECT *
FROM customers
LIMIT 6, 3
```

Inner Joins

```
SELECT *
FROM customers c
JOIN orders o
    ON c.customer_id = o.customer_id
```

Outer Joins

```
-- Return all customers whether they have any orders or not
SELECT *
FROM customers c
LEFT JOIN orders o
    ON c.customer_id = o.customer_id
```

USING Clause

If column names are exactly the same, you can simplify the join with the USING clause.

```
SELECT *
FROM customers c
JOIN orders o
    USING (customer_id)
```

Cross Joins

```
-- Combine every color with every size
SELECT *
FROM colors
CROSS JOIN sizes
```

Unions

-- Combine records from multiple result sets

```
SELECT name, address
```

```
FROM customers
```

```
UNION
```

```
SELECT name, address
```

```
FROM clients
```

Inserting Data

-- Insert a single record

```
INSERT INTO customers(first_name, phone, points)
```

```
VALUES ('Mosh', NULL, DEFAULT)
```

-- Insert multiple single records

```
INSERT INTO customers(first_name, phone, points)
```

```
VALUES
```

```
    ('Mosh', NULL, DEFAULT),
```

```
    ('Bob', '1234', 10)
```

