TCL(Transaction Control Language)

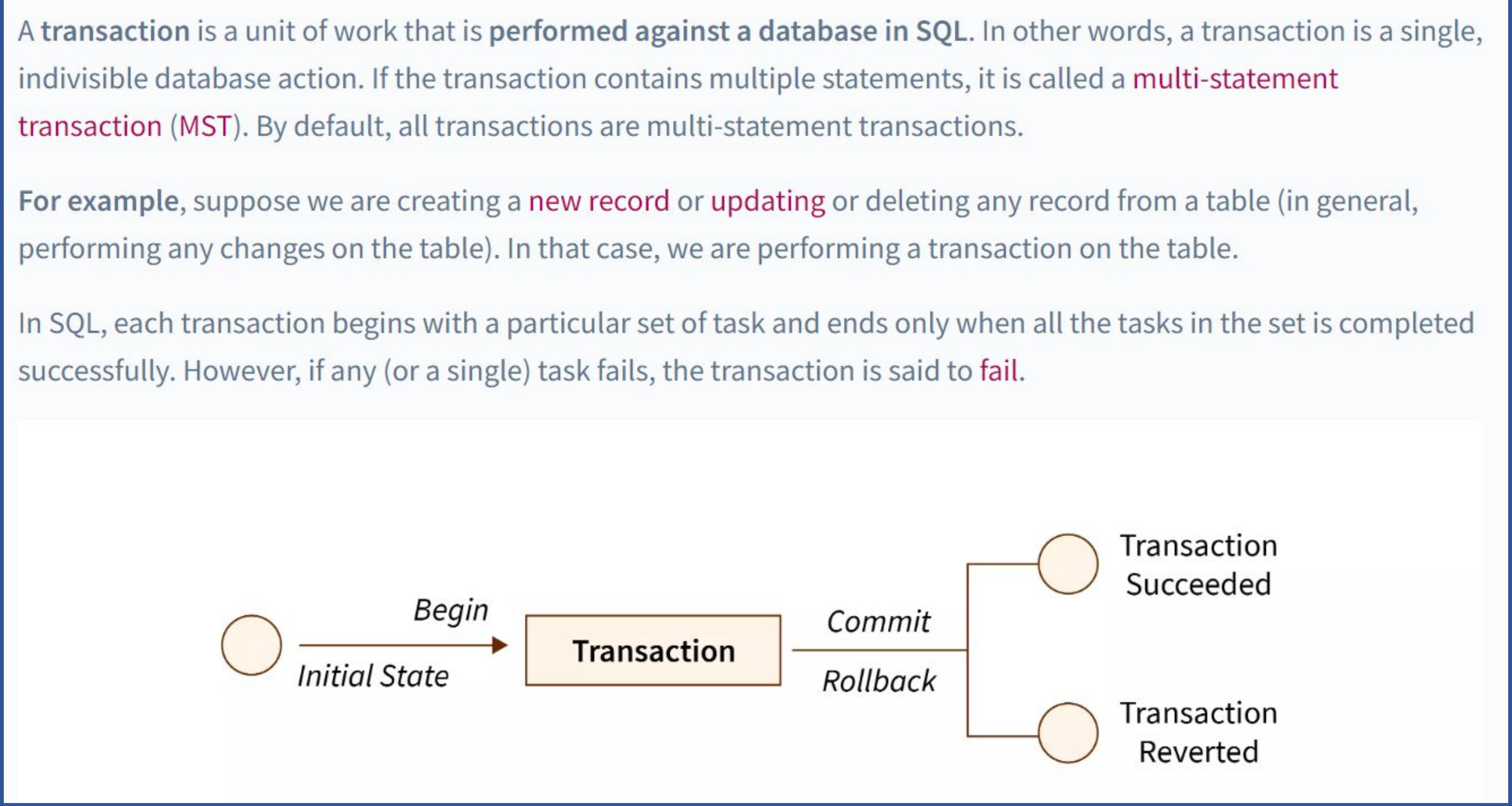
- Transactions are sequences of one or more SQL statements that are executed as a single unit of work

- Each transaction must follow the principles of being consistent, isolated, durable, and atomic—collectively known as the ACID properties.

- TCL contains 3 commands:

  ❑ COMMIT
  ❑ ROLEBACK
  ❑ SAVEPOINT

TCL(Transaction Control Language)



A **transaction** is a unit of work that is **performed against a database in SQL**. In other words, a transaction is a single, indivisible database action. If the transaction contains multiple statements, it is called a multi-statement transaction (MST). By default, all transactions are multi-statement transactions.

For example, suppose we are creating a new record or updating or deleting any record from a table (in general, performing any changes on the table). In that case, we are performing a transaction on the table.

In SQL, each transaction begins with a particular set of task and ends only when all the tasks in the set is completed successfully. However, if any (or a single) task fails, the transaction is said to fail.

TCL(Transaction Control Language)

- **COMMIT**: It is used to permanently save the changes made within a transaction to the database. It indicates that the changes are complete and should be made permanent.

- **ROLLBACK**: It is used to undo all the changes made within a transaction and restore the database to the state it was in before the transaction started. It is typically used when something goes wrong during the transaction or when you want to discard the changes for any reason.

- **SAVEPOINT**: It is used to define a point within a transaction to which you can later roll back. This allows you to undo changes only up to a specific point within the transaction, rather than rolling back the entire transaction.

Things to remember!!

- SQL keywords are NOT case sensitive:

`select` is the same as `SELECT`

- Some database systems require a semicolon at the end of each SQL statement.
- Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.
- In this tutorial, we will use semicolon at the end of each SQL statement.

Delete VS Truncate

Delete VS Truncate

DELETE	TRUNCATE
It is used to remove rows from table. A Where Clause can be used to remove only some rows	It removes all rows from a table. It is faster & does not use as much undo space as a Delete
It is a DML command, it only removes rows from a table, leaving the table structure untouched.	It is a DDL command,it involves managing the table's storage at a higher level than individual data manipulation.
In DELETE, you can rollback	You cannot rollback in TRUNCATE

SQL CONSTRAINTS

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

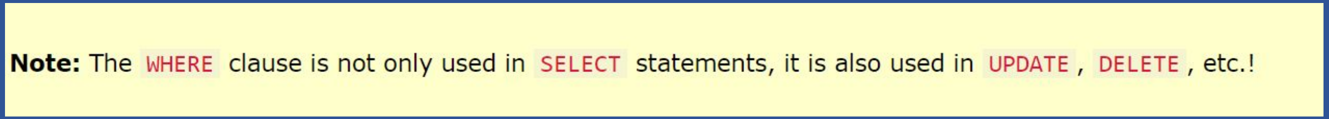
Constraints can be column level or table level.

The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified

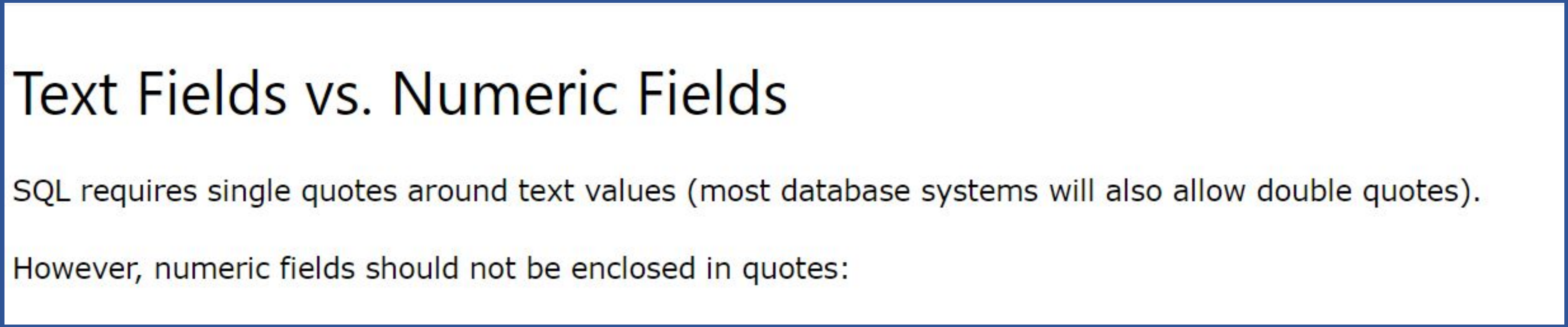
SQL WHERE CLAUSE

The WHERE clause is used to filter records.
It is used to extract only those records that fulfill a specified condition.



Note: The `WHERE` clause is not only used in `SELECT` statements, it is also used in `UPDATE`, `DELETE`, etc.!

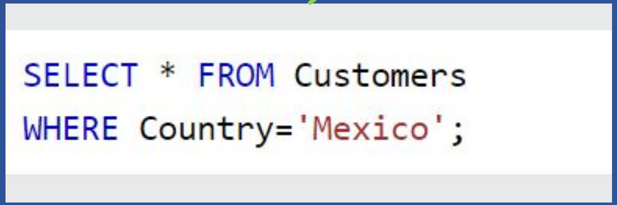
SQL WHERE CLAUSE



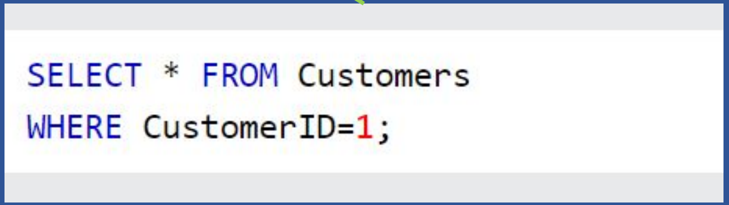
Text Fields vs. Numeric Fields

SQL requires single quotes around text values (most database systems will also allow double quotes).

However, numeric fields should not be enclosed in quotes:

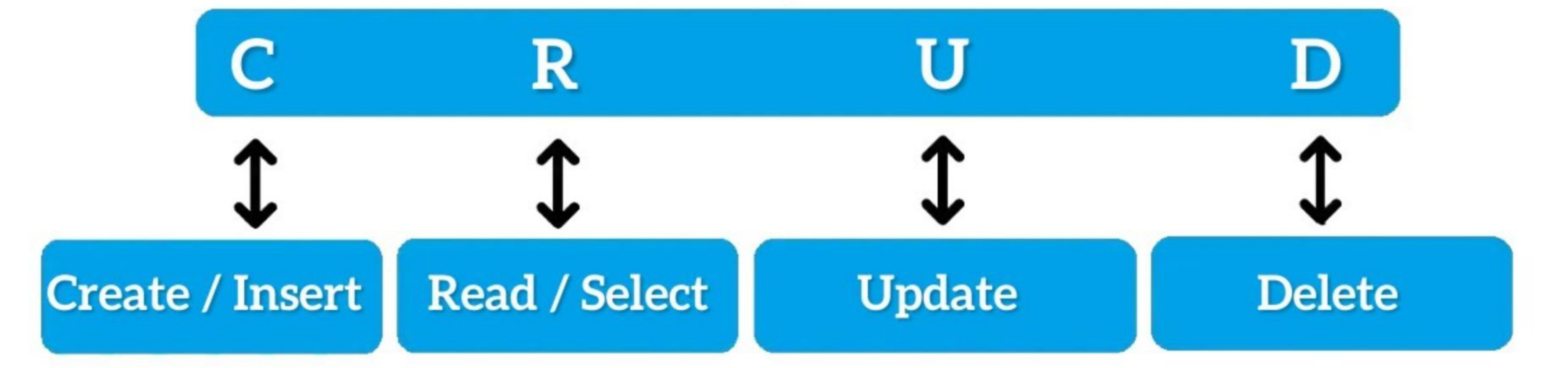


```
SELECT * FROM Customers
WHERE Country='Mexico';
```



```
SELECT * FROM Customers
WHERE CustomerID=1;
```

CRUD OPERATIONS IN SQL



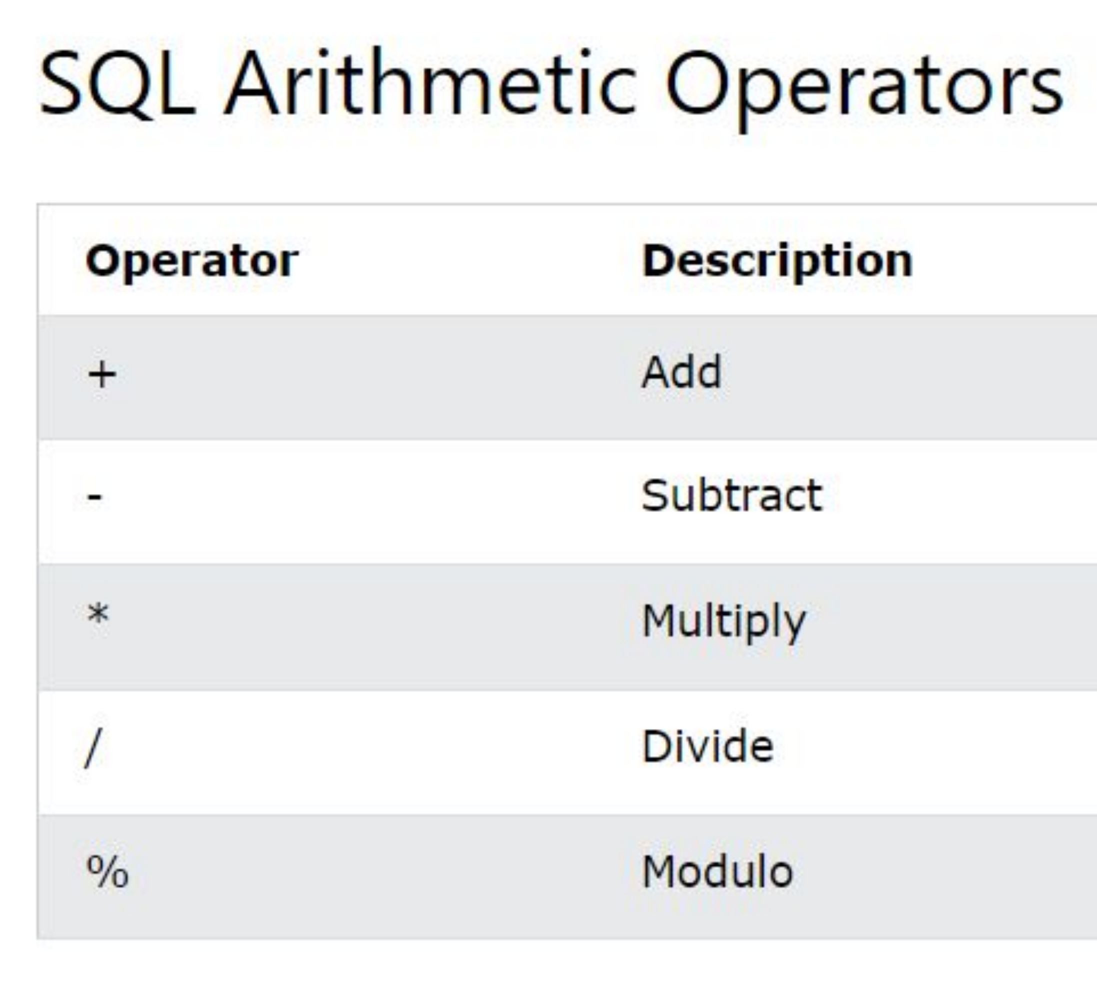
create --- Insert statements

read --- select statements

update --- update statements

delete ---- delete statements

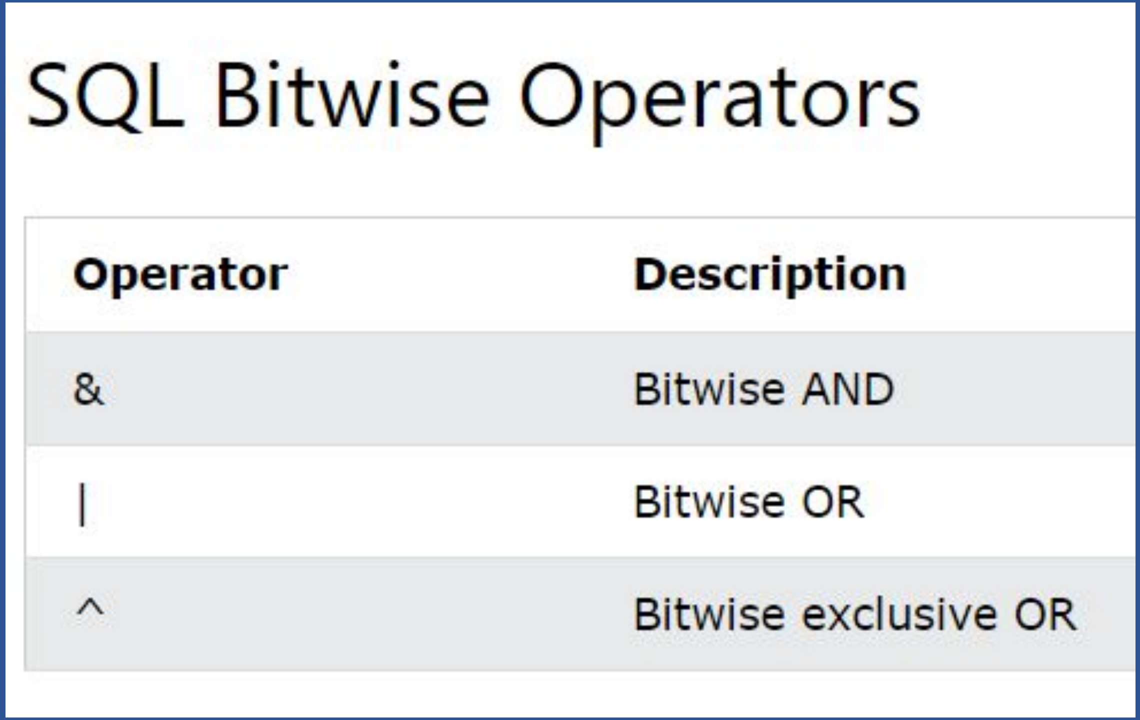
OPERATORS IN MYSQL



SQL Arithmetic Operators

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide
%	Modulo

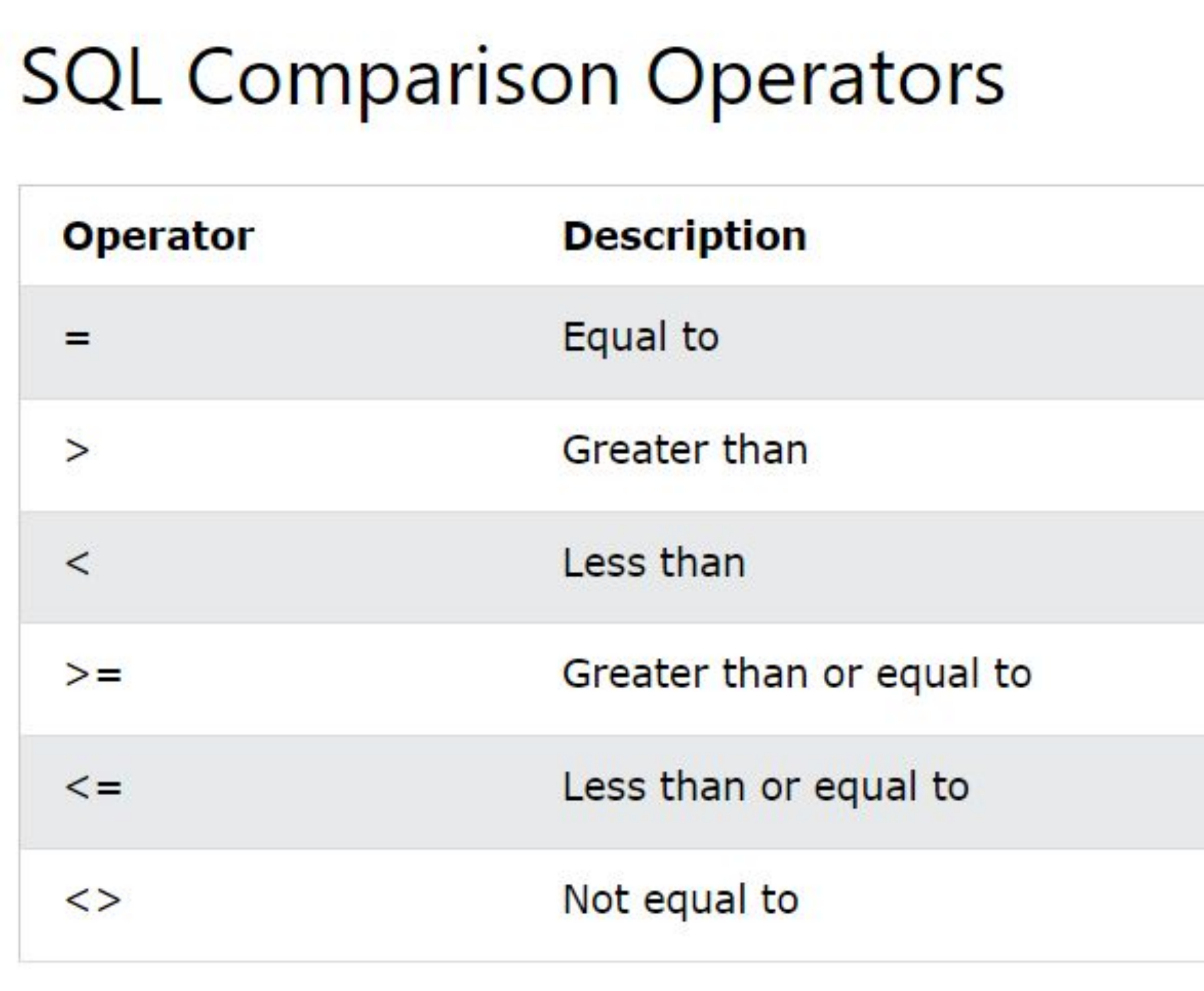
OPERATORS IN MYSQL



SQL Bitwise Operators

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR

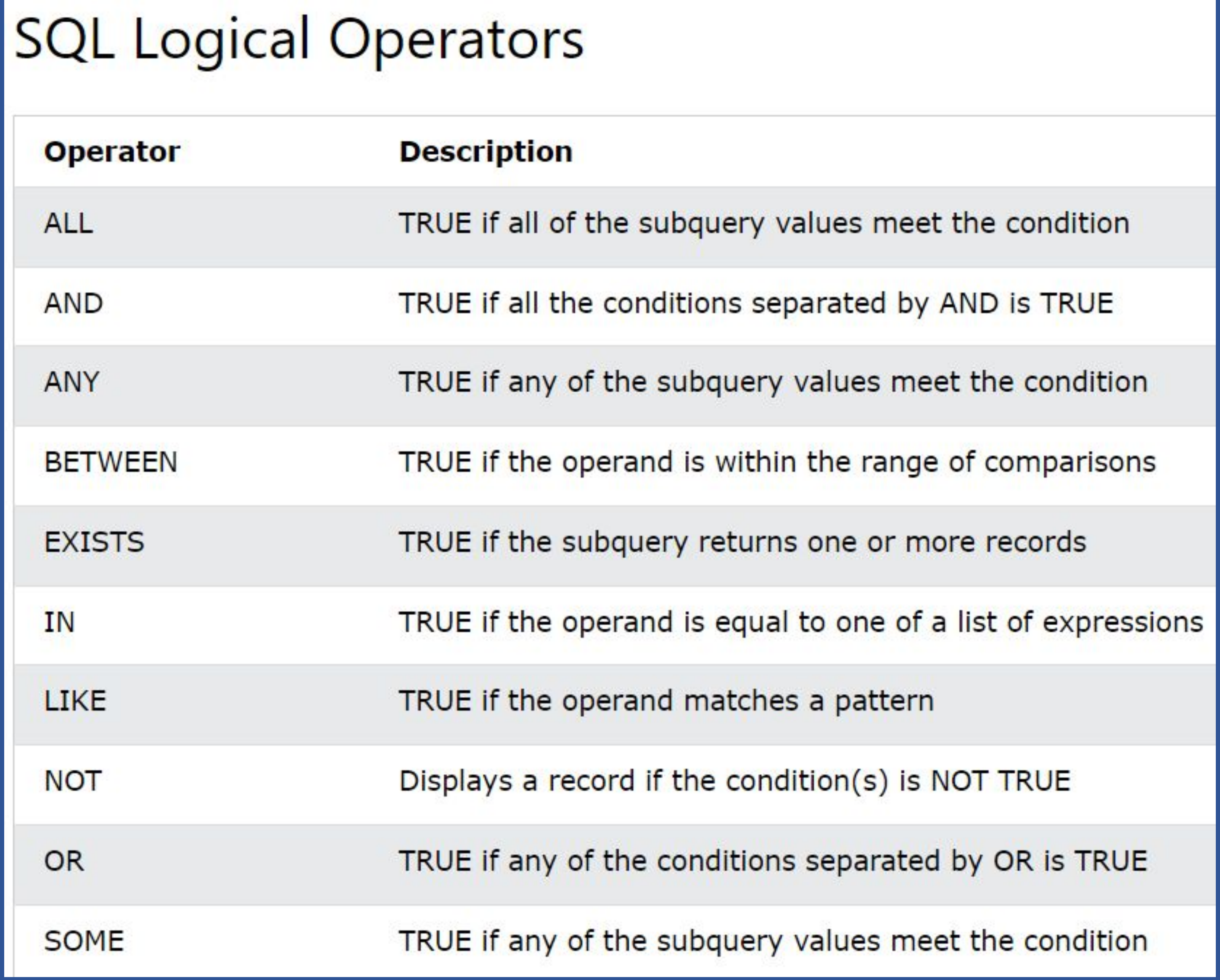
OPERATORS IN MYSQL



SQL Comparison Operators

Operator	Description
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<>	Not equal to

OPERATORS IN MYSQL



SQL Logical Operators

Operator	Description
ALL	TRUE if all of the subquery values meet the condition
AND	TRUE if all the conditions separated by AND is TRUE
ANY	TRUE if any of the subquery values meet the condition
BETWEEN	TRUE if the operand is within the range of comparisons
EXISTS	TRUE if the subquery returns one or more records
IN	TRUE if the operand is equal to one of a list of expressions
LIKE	TRUE if the operand matches a pattern
NOT	Displays a record if the condition(s) is NOT TRUE
OR	TRUE if any of the conditions separated by OR is TRUE
SOME	TRUE if any of the subquery values meet the condition

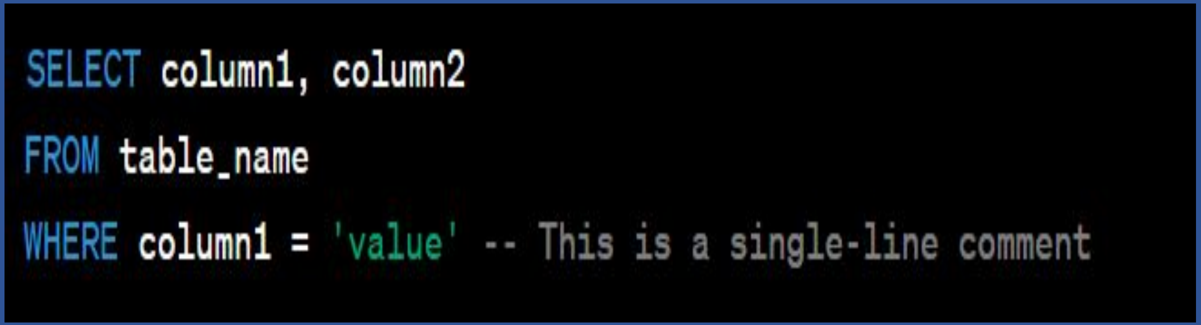
COMMENTS IN MYSQL

SINGLE LINE COMMENTS

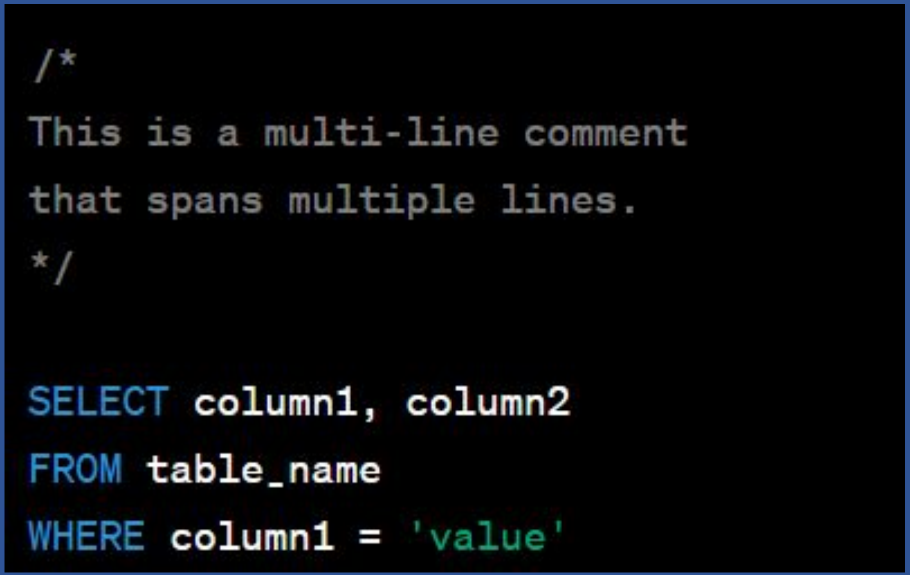
- They are used for adding remarks to a single line of code
- The 2 hyphens – are used to indicate its beginning

MULTI LINE COMMENTS

- They are used for adding remarks which spans multiple lines
- They are enclosed within ‘/*’ and ‘*/’



```
SELECT column1, column2
FROM table_name
WHERE column1 = 'value' -- This is a single-line comment
```



```
/*
This is a multi-line comment
that spans multiple lines.
*/

SELECT column1, column2
FROM table_name
WHERE column1 = 'value'
```