

Warning: The provided syntax is not common for MySQL and PostgreSQL, but variations may exist in keywords or options. Always refer to the official documentation of your specific database system for accurate commands and details.

Below is some example:

1. Rename column

MySQL: `ALTER TABLE student_info CHANGE COLUMN mobile phone INT;`

PostgreSQL: `ALTER TABLE student_info RENAME COLUMN mobile TO phone;`

2. Modify column

MySQL: `ALTER TABLE student_info MODIFY COLUMN mobile INT;`

PostgreSQL: `ALTER TABLE student_info ALTER COLUMN mobile TYPE INT;`

3. Changing the table name (varies between databases):

MySQL: `RENAME TABLE oldTableName TO newTableName;`

PostgreSQL: `ALTER TABLE oldTableName RENAME TO newTableName;`

Different possibles of Alter(add) statement

1. Adding a Single Column:

Eg 1 `ALTER TABLE tableName ADD COLUMN newColumn1 datatype;`

2. Adding Multiple Columns:

Eg 1 `ALTER TABLE tableName ADD COLUMN newColumn1 datatype, ADD COLUMN newColumn2 datatype;`

3. Adding a Column with a Default Value:

Eg 1 `ALTER TABLE tableName ADD COLUMN newColumn1 datatype DEFAULT defaultValue;`

4. Adding a Column at a Specific Position:

Eg 1 `ALTER TABLE tableName ADD COLUMN newColumn1 datatype AFTER existingColumn;`

5. Adding a Column with FOREIGN KEY Constraint:

Eg 1 `ALTER TABLE tableName ADD COLUMN newColumn1 datatype REFERENCES otherTable(otherColumn);`

Different possibles of Alter(rename/change) statement

1. Changing the data type of a column:

Eg 1 `ALTER TABLE tableName ALTER COLUMN columnName newDataType;`

2. Changing the size or precision of a column (for character or numeric types):

Eg 1 ALTER TABLE tableName ALTER COLUMN columnName newDataType(size);

3. Changing the position of a column:

Eg 1 ALTER TABLE tableName MODIFY COLUMN columnName newDataType
AFTER existingColumn;

4. Changing the table name (varies between databases):

Eg 1 RENAME TABLE oldTableName TO newTableName; -- MySQL

Eg 2 ALTER TABLE oldTableName RENAME TO newTableName; -- PostgreSQL

Different possibilities of **Select** statement

1. Selecting **all columns** from a table:

Eg 1 SELECT * FROM tableName;

2. Selecting **specific columns** from a table:

Eg 1 SELECT column1, column2 FROM tableName;

3. **Aliasing** column names:

Eg 1 SELECT column1 AS alias1, column2 AS alias2 FROM tableName;

4. Selecting **distinct** values:

Eg 1 SELECT DISTINCT column1 FROM tableName;

5. Filtering rows using **WHERE** clause: Combining **multiple conditions**:

Eg 1 SELECT column1, column2 FROM tableName WHERE condition;

Eg 2 SELECT column1, column2 FROM tableName WHERE condition1 AND condition2;

6. Sorting results using **ORDER BY**:

Eg 1 SELECT column1, column2 FROM tableName ORDER BY column1 ASC;

7. **Limiting** the number of rows returned:

Eg 1 SELECT column1, column2 FROM tableName LIMIT 10;

8. Using **aggregate functions** (e.g., SUM, AVG, COUNT):

Eg 1 SELECT AVG(column1) FROM tableName;

9. **Joining** tables:

Eg 1 SELECT t1.column1, t2.column2 FROM table1 t1 INNER JOIN table2 t2 ON t1.id =
t2.id;

10. Grouping data with **GROUP BY: HAVING**:

Eg 1 SELECT column1, COUNT(column2) FROM tableName GROUP BY column1;

Eg 2 SELECT column1, COUNT(column2) FROM tableName GROUP BY column1 HAVING
COUNT(column2) > 5;

11. Subqueries in SELECT clause:

Eg 1 SELECT column1, (SELECT MAX(column2) FROM anotherTable) AS max_value FROM tableName;

Different possibilities of Insert statement

1. Inserting values into all columns:

Eg 1 INSERT INTO tableName VALUES (value1, value2, value3, ...);

2. Inserting values into specific columns:

Eg 1 INSERT INTO tableName (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

3. Inserting multiple rows at once:

Eg 1 INSERT INTO tableName (column1, column2) VALUES (value1a, value2a), (value1b, value2b), (value1c, value2c);

4. Inserting data from another table:

Eg 1 INSERT INTO destinationTable (column1, column2) SELECT sourceColumn1, sourceColumn2 FROM sourceTable WHERE condition;

5. Inserting with a subquery:

Eg 1 INSERT INTO tableName (column1, column2) SELECT value1, value2 FROM anotherTable WHERE condition;

6. Inserting default values:

Eg 1 INSERT INTO tableName DEFAULT VALUES;

Different possibilities of Insert statement

1. Updating all rows in a table:

Eg 1 UPDATE tableName SET column1 = value1, column2 = value2, ...;

2. Updating specific rows based on a condition:

Eg 1 UPDATE tableName SET column1 = value1 WHERE condition;

3. Updating multiple columns:

Eg 1 UPDATE tableName SET column1 = value1, column2 = value2 WHERE condition;

4. Incrementing or Decrementing a Numeric Column:

Eg 1 UPDATE tableName SET numericColumn = numericColumn + 1 WHERE condition;

5. Updating with a Subquery:

Eg 1 `UPDATE tableName SET column1 = (SELECT new_value FROM anotherTable WHERE condition) WHERE condition;`

6. Updating using JOIN (MySQL, PostgreSQL, SQL Server):

Eg 1 `UPDATE table1 SET column1 = value1 FROM table2 WHERE table1.id = table2.id;`

Different possibilities of Delete statement

1. Deleting all rows from a table:

Eg 1 `DELETE FROM tableName;`

2. Deleting specific rows based on a condition:

Eg 1 `DELETE FROM tableName WHERE condition;`

3. Deleting rows from multiple tables using JOIN (MySQL, PostgreSQL, SQL Server):

Eg 1 `DELETE t1 FROM table1 t1 JOIN table2 t2 ON t1.id = t2.id WHERE condition;`