1. Quick Reference

If we are new to SQL, Quick Reference will help us understand all basics within a short time. All 25 patterns of syntax in the Quick Reference will cover most of the basics in SQL. All examples in this article can be executed in both SQL Server and MySQL.

Please have a look at this “SQL Quick Reference Guide” before we try the examples.

No.	Keyword(s)	Syntax(s)
1	CREATE DATABASE	`CREATE DATABASE Database1`
2	DROP DATABASE	`DROP DATABASE Database1`

3	CREATE TABLE	`CREATE TABLE Table1`
		`(`
		`Id DataType1`
		`, Column2 DataType2`
		`, Column3 DataType3`
		`, ...`
		`PRIMARY KEY (Id)`
		`)`

4	ALTER TABLE	`ALTER TABLE Table1`
		`ADD Column4 DataType4`
		
		`ALTER TABLE Table1`
		`DROP COLUMN Column4`

5	DROP TABLE	`DROP TABLE Table1`
6	INSERT INTO	`INSERT INTO Table1 (Id, Column2, ...)` `VALUES (Value1, Value2, ...)`
7	SELECT * INTO	`SELECT * INTO Table2` `FROM Table1`

8	UPDATE	`UPDATE Table1`
		`SET Column2 = Value2`
		`, Column3 = Value3`
		`, ...`
		`WHERE Id = Value1`

9	SELECT * FROM	`SELECT *` `FROM Table1`
10	SELECT FROM	`SELECT Id, Column2, ...`

		`FROM Table1`

11	WHERE AND/OR	`SELECT *`
		`FROM Table1`
		`WHERE Condition1`
		`AND Condition2`

12	BETWEEN	`SELECT *`
		`FROM Table1`
		`WHERE Column2`
		`BETWEEN Value1`
		`AND Value2`

13	AS	`SELECT Id AS Column4` `FROM Table1`

14	ORDER BY ASC / DESC	`SELECT *`
		`FROM Table1`
		`ORDER BY Column2 DESC`

15	DISTINCT	`SELECT DISTINCT Column2` `FROM Table1`
16	DELETE	`DELETE FROM Table1` `WHERE Id = Value1`

17	INNER JOIN	`SELECT Table1.*, Table2.*`
		`FROM Table1`
		`INNER JOIN Table2`
		`ON Table1.Column2 = Table2.Id`

18	LEFT JOIN	`SELECT Table1.*, Table2.*`
		`FROM Table1`
		`LEFT JOIN Table2`
		`ON Table1.Column2 = Table2.Id`

19	RIGHT JOIN	`SELECT Table1.*, Table2.*`
		`FROM Table1`
		`RIGHT JOIN Table2`
		`ON Table1.Column2 = Table2.Id`

20	FULL JOIN	`SELECT Table1.*, Table2.*`
		`FROM Table1`
		`FULL JOIN Table2`
		`ON Table1.Column2 = Table2.Id`

21	GROUP BY	`SELECT`
		`FROM Table1`
		`WHERE Condition1`
		`GROUP BY Column2`

22	HAVING	`SELECT Column2`
		`FROM Table1`
		`WHERE Condition1`
		`GROUP BY Column2`
		`HAVING Condition2`
		`ORDER BY Column2;`

23	MIN / MAX	`SELECT MIN(Column2)`
		`FROM Table1`
		`WHERE Condition1;`

24	LIKE	`SELECT *`
		`FROM Table1`
		`WHERE Column3 LIKE '%Name1%'`

25	UNION / UNION ALL	`SELECT Id FROM Table1`
		`UNION`
		`SELECT Id FROM Table2`

How to Create Tables and Insert Data

We need a place to keep all our data. By using the "Quick Reference" we can find a suitable command to create a database and store data. We can use SQL Server or MySQL for our examples. There we have the facility to create our database even graphically.

Our data can be represented in several classes. To store data for a specific class we need a mapping table in the database. We may need several tables to store our data. Table creation and data insertion statements can be put in a single query called a SQL script.

Example 1. Create Tables with Sample Data

By using the following SQL script we can create the "Employees" table & the "Departments" table with sample data. Please refer to 3 and 6 in "Quick Reference" before starting this example.

SQL Script:

```
CREATE TABLE Departments(
    Id INT PRIMARY KEY,
    Name VARCHAR (10) NOT NULL
```

```
);

CREATE TABLE Employees(
    Id INT PRIMARY KEY,
    DepartmentId INT NULL,
    Salary DECIMAL(12,2) NOT NULL,
    IsActive BIT DEFAULT 1,
    FOREIGN KEY (DepartmentId) REFERENCES Departments(Id)
);

INSERT Departments(Id, Name) VALUES(1, 'Production'), (2, 'HR')
, (3, 'Marketing'), (4, 'IT'), (5, 'Accounting');

INSERT Employees(Id, DepartmentId, Salary, IsActive) VALUES(1, 1, 7000, 1)
, (2, 2, 8000, 1), (3, 1, 8000, 0), (4, NULL, 8000, 1), (5, 3, 9000, 1)
, (6, NULL, 7000, 1);
```

How to Get Data from a Table with a Condition

We need a query to get data from a table. Sometimes, we don't need all the data from a table and only want to see certain data. In this case, we should include some conditions in the script to filter only required data.

Example 2. How to Get Employee Data

This example shows how to get active employees having a salary equal to or greater than 8000. Please refer to 9, 01, 11, and 12 in "SQL Quick Reference Guide" before starting this example.

SQL Script:

```
SELECT Employees.Id, Salary
FROM Employees
WHERE Salary >= 8000 AND IsActive = 1;
```

The query will filter active employees having a salary equal to or greater than 8000.

Result:

Id	Salary
2	8000.00
4	8000.00
5	9000.00

How to Create a Join

To retrieve data from two or more tables at the same time, we should combine these tables using a relation called a join.

Example 3. Create a Left Join

This example shows how to create a LEFT JOIN with the Employees table and Department table. Please refer to 18 in "SQL Quick Reference Guide" before starting this example.

SQL Script:

```
SELECT Employees.Id, Departments.Name AS Department
FROM Employees LEFT JOIN Departments
ON Employees.DepartmentId = Departments.Id
```

The query will filter all records from the Employees table. Some records in the department field have NULL values as there are no matching fields.

Result:

Id	Department
1	Production
2	HR
3	Production
4	NULL
5	Marketing
6	NULL

How to Summarize Data

We may need summarized data for our requirements. That can be done using a SQL script.

Example 4. Employee Count by Department

The following example will retrieve the employee count per department using an INNER JOIN. Employees who haven't any department are excluded from the list as the INNER JOIN select only matching records. Please refer to 14, 17, 21, 22, and 23 in "SQL Quick Reference Guide" before starting this example.

SQL Script:

```sql
SELECT Departments.Name AS Department
, COUNT(Employees.Id) AS 'Employees'
FROM Departments INNER JOIN Employees
ON Departments.Id = Employees.DepartmentId
GROUP BY Departments.Name
HAVING COUNT(Employees.Id) > 0
ORDER BY COUNT(Employees.Id) DESC
```

The query will list the employee count per department in descending order.

Result:

Department	Employees
Production	2
HR	1
Marketing	1