3. Syntax

If you are new to SQL this article will help you understand the main syntax within a short time. All 6 patterns of syntax will cover most of the basics in MySQL. All examples in this article can be executed in both SQL Server and MySQL.

Six patterns of syntax are shown in the table

No.	Syntax(s)
1	CREATE DATABASE, ALTER DATABASE, DROP DATABASE
2	CREATE TABLE, ALTER TABLE, DROP TABLE
3	INSERT, UPDATE, DELETE
4	SELECT FROM, WHERE, AND/OR
5	INNER JOIN, LEFT JOIN
6	GROUP BY, HAVING, ORDER BY

Table 1. Syntax Patterns in SQL

Create/Drop a Database

"CREATE DATABASE" command allows us to create a database. "DROP DATABASE" allows us to delete a created database.

Syntax
CREATE DATABASE Database1
DROP DATABASE Database1

Table 2. Create/Drop a Database

Example 1. Create Hr Database

The following query shows how to create a database.

SQL Script:

```
CREATE DATABASE Hr
```

Script 1. Create a Database

Create/Alter/Drop a Table

Following syntax shows how to create a table, modify a table and drop a table. It is the primary key and it is the first column.

Syntax
CREATE TABLE Table1 (Id DataType1 , Column2 DataType2 , Column3 DataType3 , … PRIMARY KEY (Id))
ALTER TABLE Table1 ADD Column4 DataType4 ALTER TABLE Table1 DROP COLUMN Column4
DROP TABLE Table1

Table 3. Create/Alter/Drop a Table

Example 2. Create Tables

The following query shows how to create the Employees table & Departments tables.

SQL Script:

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

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

Script 2. Create Tables

Using INSERT/UPDATE/DELETE

Following syntax shows how to insert, modify, and delete a record.

Syntax

INSERT INTO Table1 (Id, Column2, ...)
VALUES (Value1, Value2, ...)

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

DELETE FROM Table1
WHERE Id = Value1

Table 4. Using INSERT/UPDATE/DELETE

Example 3. Data Insert

This example shows how to insert data into Employees and Departments tables.

SQL Script:

```
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);

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

Script 3. Data Insert

Using SELECT, FROM, WHERE, AND/OR

The following syntax shows how to retrieve data from a table.

Syntax

SELECT *
FROM Table1

SELECT Id, Column2, ...
FROM Table1
WHERE Condition1
AND Condition2

Table 5. Using SELECT, FROM, WHERE, AND/OR

Example 4. How to Filter Employee Data

This example shows how to get active employees having a salary equal to or greater than 8000.

SQL Script:

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

Script 4. Filter Employee Data

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

Table 6. Employee Salary

Using INNER/LEFT/RIGHT/FULL JOIN

INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN allows you to create a join with two tables. INNER JOIN selects all records from both tables. But the LEFT JOIN selects all records from the left table.

Syntax
SELECT Table1.*, Table2.*

```
FROM Table1
INNER JOIN Table2
ON Table1.Column2 = Table2.Id
```

Table 7. Using INNER/LEFT/RIGHT/FULL JOIN

Example 5. Create a Left Join

This example shows how to create a LEFT JOIN Employees table with the Departments table.

SQL Script:

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

Script 5. Using the LEFT JOIN

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

Table 8. LEFT JOIN Result

Using GROUP BY, HAVING, ORDER BY

We use GROUP BY command to summarize data. We have to select the GROUP BY Column and it should be included in the SELECT statement. We can give a condition to the GROUP BY statement by using the HAVING command. Result data can be ordered by using the ORDER BY statement.

Syntax

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

Table 9. GROUP BY/HAVING/ORDER BY

Example 6. Employee Count by Department

The following example will retrieve 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.

SQL Script:

```
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
```

Script 6. Employee Count

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

Result:

Department	Employees
Production	2
HR	1
Marketing	1

Table 10. Employee Count