SQL TASK DOCUMENT

This document covers all essential SQL concepts and tasks from basic to advanced levels, including DDL, DML, DCL, TCL, functions, joins, subqueries, views, procedures, triggers, and even integration with machine learning concepts.

1. Introduction to SQL

Structured Query Language (SQL) is used to store, manipulate, and retrieve data in databases. It is essential for managing relational databases.

2. Data Definition Language (DDL)

Commands: CREATE, ALTER, DROP, RENAME, TRUNCATE.

Example:

```
CREATE TABLE Students (ID INT PRIMARY KEY, Name VARCHAR(50), Age INT, Department
VARCHAR(30));
```

3. Data Manipulation Language (DML)

Commands: INSERT, UPDATE, DELETE.

Example:

```
INSERT INTO Students VALUES (1, 'Maria', 21, 'Computer Science');
```

4. Data Control Language (DCL)

Commands: GRANT, REVOKE.

Example:

```
GRANT SELECT ON Students TO user_name;
```

5. Transaction Control Language (TCL)

Commands: COMMIT, ROLLBACK, SAVEPOINT.

Example:

```
SAVEPOINT sp1;
ROLLBACK TO sp1;
```

6. Constraints in SQL

Types: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT.

Example:

```
ALTER TABLE Students ADD CONSTRAINT chk_age CHECK (Age > 17);
```

7. SQL Joins

Types: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL OUTER JOIN.

Example:

```
SELECT Students.Name, Department.Name FROM Students INNER JOIN Department ON Students.DepartmentID = Department.ID;
```

8. Subqueries and Nested Queries

Subqueries are queries within another query.

Example:

```
SELECT Name FROM Students WHERE ID IN (SELECT StudentID FROM Results WHERE Marks > 90);
```

9. SQL Views

A View is a virtual table based on the result of a SELECT statement.

Example:

```
CREATE VIEW HighScorers AS SELECT Name, Marks FROM Students WHERE Marks > 85;
```

10. SQL Functions

Aggregate: COUNT(), SUM(), AVG(), MAX(), MIN()
Scalar: UCASE(), LCASE(), NOW(), LEN()

Example:

```
SELECT COUNT(*) FROM Students;
```

11. Grouping and HAVING Clause

Example:

```
SELECT Department, AVG(Marks) FROM Students GROUP BY Department HAVING AVG(Marks) > 75;
```

12. Stored Procedures

Procedures are saved SQL code that can be reused.

Example:

```
CREATE PROCEDURE GetStudents AS SELECT * FROM Students;
```

13. Triggers

Triggers automatically perform actions when events occur.

Example:

```
CREATE TRIGGER trg_insert AFTER INSERT ON Students FOR EACH ROW BEGIN INSERT INTO Log VALUES('Record Added'); END;
```

14. Cursors

Cursors are used to fetch rows one by one.

Example:

```
DECLARE cur CURSOR FOR SELECT Name FROM Students;
OPEN cur;
FETCH NEXT FROM cur;
```

15. Advanced SQL Concepts

Topics: Window Functions, CTE (Common Table Expressions), JSON Functions.

Example:

```
WITH HighScorers AS (SELECT Name, Marks FROM Students WHERE Marks > 90) SELECT * FROM HighScorers;
```

16. SQL and Machine Learning Integration

Modern databases like SQL Server, PostgreSQL, and BigQuery support ML integration.

Example:

```
CREATE MODEL StudentPerformanceModel OPTIONS(model_type='linear_reg') AS SELECT Hours_Studied, Marks FROM StudentData;
```

17. Optimization and Indexing

Indexes speed up query performance.

Example:

```
CREATE INDEX idx_name ON Students(Name);
```

18. Backup and Restore

To secure data, use backup commands.

Example:

```
BACKUP DATABASE CollegeDB TO DISK = 'D:\backup\CollegeDB.bak';
```

19. Security and User Management

Example:

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost';
```

20. Conclusion

SQL is the foundation of all modern data handling, from basic data operations to integrating machine learning models. Understanding SQL thoroughly enables efficient data-driven decision-making.