



SQL fundamentals - Introduction, data definition in SQL

SQL

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

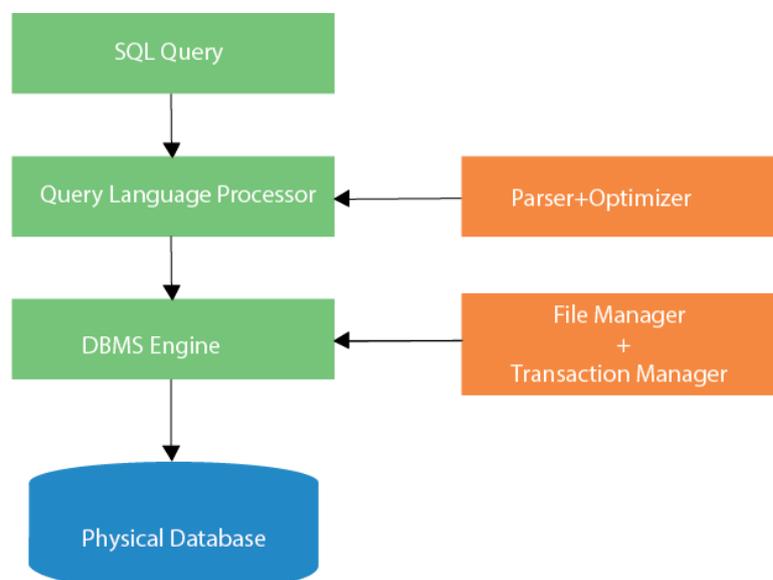
Rules:

SQL follows the following rules:

- Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase.
- Statements of SQL are dependent on text lines. We can use a single SQL statement on one or multiple text line.
- Using the SQL statements, you can perform most of the actions in a database.
- SQL depends on tuple relational calculus and relational algebra.

SQL process:

- When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task.
- In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc.
- All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.





Data Definition Language Commands

The Data Definition Language is used to create and destroy databases and database objects. These commands are primarily used by database administrators during the setup and removal phases of a database project. DDL revolves around four primary commands—**create**, **use**, **alter**, and **drop**.

Create

The **create** command establishes databases, tables, or queries on your platform. For example, the command:

```
CREATE DATABASE employees;
```

creates an empty database named *employees* on your DBMS. After creating the database, the next step is to create tables that contain data. Another variant of the **create** command accomplishes this purpose. The command:

```
CREATE TABLE personal_info (first_name char(20) not null, last_name char(20) not null, employee_id int not null);
```

establishes a table titled *personal_info* in the current database. In the example, the table contains three attributes: *first_name*, *last_name*, and *employee_id* along with some additional information.

Use

The **use** command specifies the active database. For example, if you're currently working in the sales database and want to issue some commands that will affect the employee database, preface them with the following SQL command:

```
USE employees;
```

Double-check the database you're working in before issuing SQL commands that manipulate data.

Alter

After you've created a table within a database, modify its definition through the **alter** command, which changes to the structure of a table without deleting and recreating it. Take a look at the following command:

```
ALTER TABLE personal_info ADD salary money null;
```



SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35 (An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This example adds a new attribute to the `personal_info` table—an employee's salary. The *money* argument specifies that an employee's salary stores using a dollars and cents format. Finally, the *null* keyword tells the database that it's OK for this field to contain no value for any given employee.

Drop

The final command of the Data Definition Language, **drop**, removes entire database objects from our DBMS. For example, to permanently remove the `personal_info` table that we created, use the following command:

```
DROP TABLE personal_info;
```

Similarly, the command below would be used to remove the entire employee database:

```
DROP DATABASE employees;
```

Use this command with care. The **drop** command removes entire data structures from your database. If you want to remove individual records, use the **delete** command of the Data Manipulation Language.