Structure Query Language (SQL)

1. Introduction SQL
2. Data Definition Language (DDL)
3. Data Manipulation Language (DML)
4. Data Control Language (DCL)

Structured Query Language(SQL)

6.1 Introduction

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms.

6.2 What can SQL do?

1. SQL can execute queries against a database .
2. SQL can retrieve data from a database .
3. SQL can insert records in a data base .
4. SQL can update records in a database.
5. SQL can delete records from a database.
6. SQL can create new data bases .
7. SQL can create new tables in a database.
8. SQL can create stored procedures in a database.
9. SQL can create views in a data base .
10. SQL can set permissions on tables, procedures, and views .

6.3 Database Languages

A database system provides a **data definition language** to specify the database schema and a **data manipulation language** to express database queries and updates and a **data control language** to configure security access to relational databases . In practice, the data definition and data manipulation languages are not two separate languages; instead they simply form parts of a single database language, such as the widely used SQL language.

6.3.1 Data Definition Language (DDL)

The SQL DDL allows specification of not only a set of relations, but also information about each relation, including
• The schema for each relation
• The domain of values associated with each attribute
• The integrity constraints
• The set of indices to be maintained for each relation
• The security and authorization information for each relation
• The physical storage structure of each relation on disk

Data Definition Language (DDL): statements are used to define the database structure or schema. Some examples:

- CREATE - to create objects in the database.
- ALTER - alters the structure of the database.
- DROP - delete objects from the database.
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed.
- COMMENT - add comments to the data dictionary.
- RENAME - rename an object.

6.3.2 Data Manipulation Language(DML)

A **data-manipulation language (DML)** is a language that enables users to access or manipulate data as organized by the appropriate data model.
Data manipulation is

• The retrieval of information stored in the database
• The insertion of new information into the database
• The deletion of information from the database
• The modification of information stored in the database
There are basically two types:

• **Procedural DMLs** require a user to specify *what* data are needed and *how* to get those data.

• **Declarative DMLs** (also referred to as **nonprocedural** DMLs) require a user to specify *what* data are needed *without* specifying how to get those data.

Declarative DMLs are usually easier to learn and use than are procedural DMLs.

However, since a user does not have to specify how to get the data, the database system has to figure out an efficient means of accessing data. The DML component of the SQL language is nonprocedural.

A **query** is a statement requesting the retrieval of information. The portion of a DML that involves information retrieval is called a **query language**. Although technically incorrect, it is common practice to use the terms *query language* and *data manipulation language* synonymously.

Data Manipulation Language (DML) statements are used for managing data within schema objects. Some examples:

- **SELECT - retrieve data from the a database.**
- **INSERT - insert data into a table.**
- **UPDATE - updates existing data within a table.**
- **DELETE - deletes all records from a table, the space for the records remain.**
- **MERGE - UPSERT operation (insert or update).**
- **CALL - call a PL/SQL or Java subprogram.**
- **EXPLAIN PLAN - explain access path to data.**
- **LOCK TABLE - control concurrency.**

6.3.3 Data Control Language:

Data Control Language (DCL) :statement is a subset of the Structured Query Language (SQL) that allows database administrators to configure security access to relational databases. Some examples:

- **GRANT - gives user's access privileges to database.**
- **REVOKE - withdraw access privileges given with the GRANT command.**

6.4 SQL - Data Types

SQL data type is an attribute that specifies type of data of any object. Each column, variable and expression has related data type in SQL. You would use these data types while creating your tables. You would choose a particular data type for a table column based on your requirement.

The SQL standard supports a variety of built-in domain types, including:

• **char**(*n*): A fixed-length character string with user-specified length *n*. The full form, **character**, can be used instead.
• **varchar**(*n*): A variable-length character string with user-specified maximum length *n*. The full form, **character varying**, is equivalent.
• **int**: An integer (a finite subset of the integers that is machine dependent). The full form, **integer**, is equivalent.
• **smallint**: A small integer (a machine-dependent subset of the integer domain type).
• **numeric**(*p, d*): A fixed-point number with user-specified precision. The number consists of *p* digits (plus a sign), and *d* of the *p* digits are to the right of the decimal point. Thus, **numeric**(3,1) allows 44.5 to be stored exactly, but neither 444.5 or 0.32 can be stored exactly in a field of this type.
• **real, double precision**: Floating-point and double-precision floating-point numbers with machine-dependent precision.
• **float**(*n*): A floating-point number, with precision of at least *n* digits.
• **date**: A calendar date containing a (four-digit) year, month, and day of the month.
• **time**: The time of day, in hours, minutes, and seconds. A variant, **time**(*p*), can be used to specify the number of fractional digits for seconds (the default being 0). It is also possible to store time zone information along with the time.
• **timestamp**: A combination of **date** and **time**. A variant, **timestamp**(*p*), can be used to specify the number of fractional digits for seconds (the default here being 6).
Date and time values can be specified like this:

date ’2001-04-25’
time ’09:30:00’
time stamp ’2001-04-25 10:29:01.45’

Dates must be specified in the format year followed by month followed by day, as shown. The second's field of time or timestamp can have a fractional part, as in the timestamp above. We can use an expression of the form caste as to convert a character string (or string valued expression) e to the type t, where t is one of date, time, or timestamp. The string must be in the appropriate format as illustrated at the beginning of this paragraph.
To extract individual fields of a date or time value d, we can use extract (field from d), where field can be one of year, month, day, hour, minute, or second.

6.4.1 SQL Data Type Quick Reference

However, different databases offer different choices for the data type definition. The following table shows some of the common names of data types between the various database platforms:

Data type	**Access**	**SQL Server**	**Oracle**	**MySQL**	**PostgreSQL**
Boolean	Yes/No	Bit	Byte	N/A	Boolean
Integer	Number (integer)	Int	Number	Int Integer	Int Integer
Float	Number (single)	Float Real	Number	Float	Numeric
Currency	Currency	Money	N/A	N/A	Money
string (fixed)	N/A	Char	Char	Char	Char
string (variable)	Text (<256) Memo (65k+)	Varchar	Varchar Varchar2	Varchar	Varchar
binary object	OLE Object Memo	Binary (fixed up to 8K) Varbinary (<8K) Image (<2GB)	Long Raw	Blob Text	Binary Varbinary

Note: Data types might have different names in different database. And even if the name is the same, the size and other details may be different! **Always check the documentation!**

6.5 Data Definition Language (DDL) command

6.5.1. Create Command

Create is a DDL command used to create a table or a database.

6.5.1.1 Creating a Database

To create a database in RDBMS, *create* command is uses. Following is the Syntax,

```
create database database-name;
```

Example for Creating Database

```
Create database Test;
```

The above command will create a database named **Test**.

6.5.1.2 Creating a Table

Create command is also used to create a table. We can specify names and data types of various columns along. Following is the Syntax,

```
Create table table-name

{

 column-name1 datatype1,

 column-name2 datatype2,

 column-name3 datatype3,

 column-name4 datatype4

};
```

Create table command will tell the database system to create a new table with given table name and column information.

Example for creating Table

```
Create table Student(id int, name varchar, age int);
```

The above command will create a new table **Student** in database system with 3 columns, namely id, name and age.

6.5.2 Alter command

Alter command is used for alteration of table structures. There are various uses of *alter* command, such as,

- to add a column to existing table
- to rename any existing column
- to change data type of any column or to modify its size.
- *Alter* is also used to drop a column.

6.5.2.1 To Add Column to existing Table

Using alter command we can add a column to an existing table. Following is the Syntax,

```
Alter table table-name add(column-name datatype);
```

Here is an Example for this,

```
alter table Student add(address char);
```

The above command will add a new column *address* to the **Student** table

6.5.2.2 To Add Multiple Column to existing Table

Using alter command we can even add multiple columns to an existing table. Following is the Syntax,

```
Alter table table-name add(column-name1 datatype1, column-name2 d
atatype2, column-name3 datatype3);
```

Here is an Example for this, alter table Student add(father-name varchar(
60), mother-name varchar(60), dob date);

The above command will add three new columns to the **Student** table