SQL

Structured Query Language

Ms. J R Shruti
Assistant lecturer,
Department of Information
technology
NITK, Surathkal

Introduction

SQL is a database computer language designed for the retrieval and management of data in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Applications of SQL

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

RDBMS

RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

MySQL Installation in Ubuntu

- sudo apt update
- sudo apt install mysql-server
- sudo mysql_secure_installation
- sudo mysql
- MySQL should have started running automatically. To test this, check its status.

  systemctl status mysql.service

SQL

SQL commands are mainly categorized into four categories as:

1. DDL – Data Definition Language
2. DQl – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language

SQL is categorized into four categories

DDL (Data Definition Language): define the schema of the database.

DML (Data Manipulation Language): provides commands to manipulate the database (query, insert, update, delete).

DQL (Data Query Language) :

The purpose of DQL Command is to get some schema relation based on the query passed to it. Ex: SELECT

DCL(Data Control Language) : DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

TCL(transaction Control Language) : TCL commands deals with the transaction within the database. Ex: COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION

Basic DDL Commands in SQL

CREATE: to define new tables (to define relation schemas)

DROP: to delete table definitions (to delete relation schemas)

ALTER: to change the definitions of existing tables (to change relation schema)

SQL-
Data Definition Language

DDL-Introduction

- Create
- Drop
- Truncate
- Alter
- Rename

DDL-Creating a Database

- **Syntax:**

  CREATE DATABASE database_name;

  Example:

  CREATE DATABASE COMPANY;

DDL-Creating a Table

- **Syntax**

```
CREATE TABLE table_name
(Column_namedatatype[(size)],
Column_namedatatype[(size)], );
```

- **Example**

```
CREATE TABLE books
 (ISBN          char(20),
 Title          char(50),
 AuthorID       Integer,
 Price          float);
```

Modifying Records

Truncate Statement

- **Truncate Statement:**
  - used to delete all the rows of a table. Delete can also be used to delete all the rows from the table. The difference is that delete performs a delete operation on each row in the table and the database performs all attendant tasks on the way. On the other had the Truncate statement simply throws away all the rows at once and is much quicker. The note of caution is that truncate does not do integrity checks on the way which can lead to inconsistencies on the way. If there are dependencies requiring integrity checks we should use delete.
- **Syntax:** TRUNCATE TABLE table_name;
- **Example:**

```
TRUNCATE TABLE studios;
```

- This deletes all the rows of the table studios

Modifying Records

Drop Statement

- **Drop Statement:**
  - _ used to remove elements from a database, such as tables, indexes or even users and databases. Drop command is used with a variety of keywords based on the need.

- **Drop Table Syntax:** DROP TABLE table_name;
- **Drop Table Example:** DROP TABLE studios;

Modifying Records

Alter Statement

- **Alter Statement:**
  - used to make changes to the schema of the table. Columns can be added and the data type of the columns changed as long as the data in those columns conforms to the data type specified.
- **Syntax:**

```
ALTER TABLE table_name
[ADD|MODIFY|DROP] column column_name datatype;
```

- **Example:**

```
ALTER TABLE studios
ADD column revenue int;
ALTER TABLE studios
modify column revenue double;

ALTER TABLE studios  drop column revenue int;
```

DDL:RENAME

RENAME command is used to set a new name for any existing table. Following is the syntax,

```
RENAME TABLE old_table_name to
new_table_name
```

Here is an example explaining it.

```
RENAME TABLE Product to Product_info;
```

The above query will rename the table **Product** to **Product_info.**

DDL-Data Types

- Following broad categories of data types exist in most databases:
  - String Data
  - Numeric Data
  - Temporal Data

DDL-String Data

- **Fixed Length**:
- Occupies the same length of space in memory no matter how much data is stored in them.
- **Syntax:**

char(n) where n is the length of the String

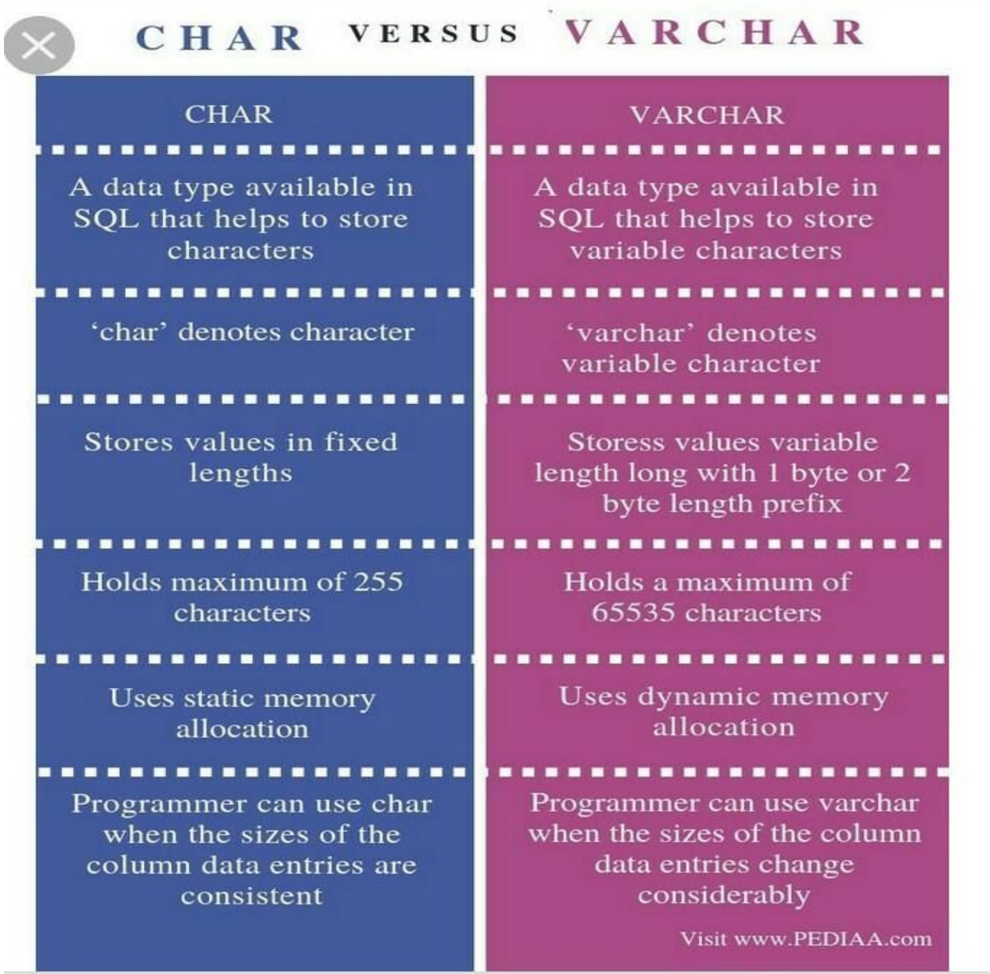
- e.g. name char(50)

If the variable stored for name is ‘Santhosh’ the extra 43 fields are padded with blanks

DDL-String Data

- **Variable Length** string is specified with maximum length of characters possible in the string, however, the allocation is sized to the size of the data stored in memory.
- **Syntax:**
- Varchar(n) – n is the maximum length of data possible for the type
- There may be a restriction in the maximum length of the data that you can specify in the declaration which will vary according to the database.

All character data has to be enclosed in single quotes during specification.



CHAR VERSUS VARCHAR

CHAR	VARCHAR
A data type available in SQL that helps to store characters	A data type available in SQL that helps to store variable characters
‘char’ denotes character	‘varchar’ denotes variable character
Stores values in fixed lengths	Storess values variable length long with 1 byte or 2 byte length prefix
Holds maximum of 255 characters	Holds a maximum of 65535 characters
Uses static memory allocation	Uses dynamic memory allocation
Programmer can use char when the sizes of the column data entries are consistent	Programmer can use varchar when the sizes of the column data entries change considerably

Visit www.PEDIAA.com

DDL-Numeric Data Types

- Store all the data related to purely numeric data.
- Some numeric data may also be stored as a character field
  e.g. zip codes
- **Common Numeric Types**:

– Decimal	Floating point number
– Float	Floating point number
– Integer(size)	Integer of specified length
– Money	A number which contains exactly two digits after the decimal point
– Number	A standard number field that can hold a floating point data

DDL-Temporal Data Types

- **These represent the dates and time:**
- Three basic types are supported:
  - Dates Times
  - Date-Time Combinations

DDL-
Constraints- Disallowing Null Values

Disallowing Null Values:

Null values entered into a column means that the data in not known. These can cause problems in Querying the database.

- Not Null clause is used in preventing null values from being entered in a column.
- **Example:**

```
CREATE TABLE Studios
( studio_id     number, name char(20) NOT NULL,
  city  varchar(50) NOT NULL,  state     char(2)    NOT NULL);
```

- Null clause can be used to explicitly allow null values in a column also

DDL-Constraints- Default Value

Default Value:

- A default value can be inserted in any column by using the Default keyword.

- **Example:**

```
CREATE TABLE Movies (
movie_title varchar(40) NOT NULL,
release_date date        DEFAULT sysdate,
genre varchar(20)       DEFAULT ‘Comedy’ ) ;
```

- release_date defaults to the current date.

DATA MANIPULATION LANGUAGE

(Modifying Records)

Basic DML Commands in SQL

INSERT: to add new rows to table

UPDATE: to change the “state” (the value) of rows.

DELETE: to remove rows

SELECT: a query command that uses relation algebra *like* expressions

Modifying Records

Insert Statement

- **Insert:**
  - Allows you to add new records to the Table
- **Syntax:**
  - Insert into table_name[(column_list)] values (value_list)
- **Example:**

INSERT INTO studios VALUES (1, ‘Giant’, ‘Los Angeles’, ‘CA’)

INSERT INTO studios (studio_city, studio_state, studio_name,
	studio_id) VALUES (‘Burbank’, ‘CA’, ‘MPM’, 2)

Modifying Records

Delete Statement

- **Delete Statement:**
  - is used to remove records from a table of the database. The where clause in the syntax is used to restrict the rows deleted from the table otherwise all the rows from the table are deleted.
- **Syntax:** DELETE FROM table_name [WHERE Condition]
- **Example:**

```
DELETE FROM City_State
WHERE state = ‘TX’
```

- Deletes all the rows where the state is Texas keeps all the other rows.

Modifying Records

Update Statement

- **Update Statement:**
  - used to make changes to existing rows of the table. It has three parts. First, you ,must specify which table is going to be updated. The second part of the statement is the set clause, in which you should specify the columns that will be updated as well as the values that will be inserted. Finally, the where clause is used to specify which rows will be updated.
- **Syntax:**

```
UPDATE table_name
   SET column_name1 = value1,  column_name2 = value2, …..
   [WHERE Condition]
```

- **Example:**

```
UPDATE studios
SET studio_city = ‘New York’, studio_state = ‘NY’  WHERE studio_id = 1
```