**4.2 DATA DEFINITION LANGUAGE**

Data Definition Language (DDL) is a standard for commands that define the different structures in a database. DDL statements create, modify, and remove database objects such as tables, indexes, and users. Common DDL statements are CREATE, ALTER, and DROP.

The present database industry incorporates DDL into any formal language describing data. However, it is considered to be a subset of SQL (Structured Query Language). SQL often uses imperative verbs with normal English such as sentences to implement database modifications. Hence, DDL does not show up as a different language in an SQL database, but does define changes in the database schema.

Commonly used DDL in SQL querying are:

**CREATE:** This command builds a new table and has a predefined syntax. The CREATE statement syntax is CREATE TABLE [table name] ([column definitions]) [table parameters]. CREATE TABLE Employee (Employee Id INTEGER PRIMARY KEY, First name CHAR (50) NULL, Last name CHAR (75) NOT NULL).

**ALTER:** An alter command modifies an existing database table. This command can add up additional column, drop existing columns and even change the data type of columns involved in a database table. An alter command syntax is ALTER object type object name parameters. ALTER TABLE Employee ADD DOB Date.

**DROP:** A drop command deletes a table, index or view. Drop statement syntax is DROP object type object name. DROP TABLE Employee.

**4.2.1 DOMIAN TYPE IN SQL**

1. The SQL-92 standard supports a variety of built-in domain types:
   * **char**(n) (or **character**(n)): fixed-length character string, with user-specified length.
   * **varchar**(n) (or **character varying**): variable-length character string, with user-specified maximum length.
   * **int** or **integer**: an integer (length is machine-dependent).
   * **smallint**: a small integer (length is machine-dependent).
   * **numeric**(*p, d*): a fixed-point number with user-specified precision, consists of *p* digits (plus a sign) and *d* of *p* digits are to the right of the decimal point. E.g., **numeric**(*3, 1*) allows 44.5 to be stored exactly but not 444.5.
   * **real** or **double precision**: floating-point or double-precision floating-point numbers, with machine-dependent precision.
   * **float**(n): floating-point, with user-specified precision of at least *n* digits.
   * **date**: a calendar date, containing four digit year, month, and day of the month.
   * **time**: the time of the day in hours, minutes, and seconds.
2. SQL-92 allows arithmetic and comparison operations on various numeric domains, including, **interval** and *cast* (*type coercion*) such as transforming between *smallint* and *int*. It considers strings with different length are compatible types as well.
3. SQL-92 allows **create domain** statement.

**4.2.2 SCHEMA DEFINITION IN SQL**

**4.3 DATA MANIPULATION LANGUAGE**

A data manipulation language (DML) is a family of syntax elements similar to a computer programming language used for selecting, inserting, deleting and updating data in a database. A popular data manipulation language is that of Structured Query Language (SQL), which is used to retrieve and manipulate data in a relational database.Other forms of DML are those used by IMS/DLI, CODASYL databases, such as IDMS and others.

Data manipulation language comprises the SQL data change statements,which modify stored data but not the schema or database objects. Manipulation of persistent database objects, e.g., tables or stored procedures, via the SQL schema statements, rather than the data stored within them, is considered to be part of a separate data definition language. In SQL these two categories are similar in their detailed syntax, data types, expressions etc., but distinct in their overall function.

Data manipulation languages have their functional capability organized by the initial word in a statement, which is almost always a verb.

**4.3.1 THE SELECT CLAUSE**

The SELECT statement is used to select data from a database. The result is stored in a result table, called the result-set.

**SQL SELECT Syntax**

SELECT *column\_name(s)*

FROM *table\_name;*

And

SELECT \* FROM *table\_name;*

**4.3.2 THE WHERE CLAUSE**

The WHERE clause (optional) specifies which data values or rows will be returned or displayed, based on the criteria described after the keyword where.

**SQL WHERE Syntax**

SELECT *column\_name(s)*

FROM *table\_name*

WHERE *column\_name*

**4.3.3 THE FROM CLAUSE**

The table name that follows the keyword FROM specifies the table that will be queried to retrieve the desired results.

**SQL FROM Syntax**

SELECT *column\_name(s)*

FROM *table\_name*

**4.3.4 THE RENAME OPERATION**

Rename clause is used to rename the table. for example

rename table *old\_name* to *new\_name*;