**DataTypes:-**

**#. Datatype defines or determine the type of data which a memory location can accept.**

**1. CHAR**

**2. VARCHAR/VARCHAR2**

**3. DATE**

**4. NUMBER**

**5. LARGE OBJECT**

**-CLOB ( Character Large Object)**

**-BLOB ( Binary Large Object )**

**1. CHAR**

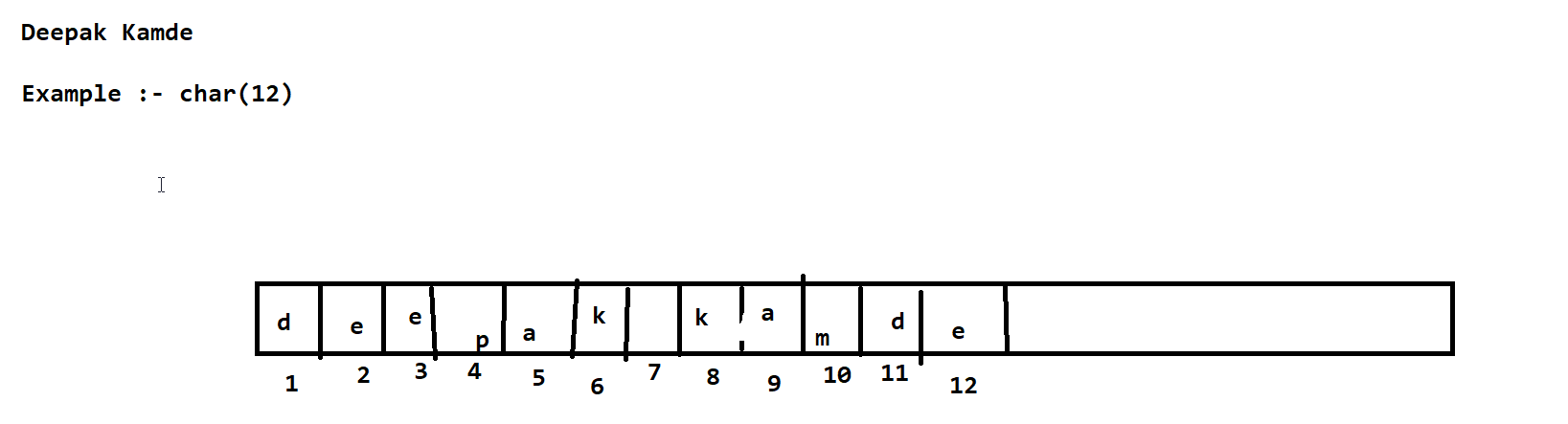
**#. This Datatype can accept characters such as A-Z , a-z, 0-9, and all special character.**

**#. Syntax:- Char(size)**

**#. size :- It specify the maximum number of characters can accept.**

**#. It is a type of fixed length memory allocation.**

**#. We can store upto 2000 characters in Char Datatype.**

****

**VARCHAR**

**#. This Datatype can accept characters such as a-z,A-Z, 0-9, andd all special characters.**

**#. varchar(size)**

**Example:- varchar(2000)**

**#. It is a type of variable length memory allocation.**

**#. We can store upto '2000' characters in varchar datatypes.**

**VARCHAR2**

**Syntax:- Varchar2(size)**

**Example:- Varchar2(4000)**

**DATE:-**

**#. The Standard format is DD-MM-YY**

**#. DD-MM-YYYY**

**Example:- 12-JUN-25 and 12-JUN-2025**

**NULL:-**

**#. Null is a keyword which is used to represent Empty cell or nothing.**

**Characterstics of NULL:-**

**#. NULL does not represent '0' or 'space'.**

**#. NULL does not occupy memory.**

**#. Any Arithmetic Operation on NULL will result in NULL itself.**

**#. Example :- 80+NULL= NULL**

**#. 10\*NULL =NULL**

**CONSTRAINTS:-**

**#. Constraints are the rules which are given to the column for validating the data.**

**Types of Constraints:-**

**#. UNIQUE**

**#. NOT NULL**

**#. CHECK**

**#. PRIMARY KEY**

**#. FOREIGN KEY**

**1. UNIQUE:-**

**#. It is a constraint which is used to avoid duplicate values entering into the table.**

**Syntax:- UNIQUE(COLOUMN\_NAME)**

**2. NOT NULL :-**

**#. It is used for a particular column which can not be NULL.**

**#. It specify the particular column is mandatory.**

**Syntax:- NOT NULL ( COLPUMN\_NAME)**

**3. CHECK :-**

**#. CHECK is a Extra Validation which is given for a particular column.**

**#. if the condition is satisfied the value is accepted or rejected.**

**Syntax:- CHECK(CONDITION)**

**CHECK ( SALARY > 0)**

**CHECK ( LENGTH(PHONE\_NUMBER) =10)**

**4. PRIMARY KEY:-**

**#. Primary Key is a Constraint with which we can uniquely identify a record/row from a table.**

**#. In a table we can have only 1 primary key (PK)**

**#. PK cannot accept NULL values.**

**#. PK cannot accept 'DUPLICATE' values**

**#. PK is always combination of NOT NULL and UNIQUE**

**#. PK is not mandatory but prefer to have 1.**

CREATE DATABASE COLLAGE;

USE COLLAGE;

CREATE TABLE STUDENT( ID VARCHAR(10) PRIMARY KEY ,

NAME VARCHAR(50) NOT NULL, CITY VARCHAR(50) NOT NULL);

SELECT \* FROM STUDENT;

INSERT INTO STUDENT ( ID, NAME , CITY )

VALUES ( 102, 'BALAJI','PUNE'),

( 103, 'SANKET','NAGPUR'),

( 104, 'SUMIT','WARDHA'),

( 105, 'NIKHIL','NAGPUR'),

( 106, 'SACHIN','MUMBAI'),

( 107, 'VIRAT','BANGLORE'),

( 108, 'VINAY','NAGPUR'),

( 109, 'SIMON','HYDERABAD'),

( 110, 'PETER','CHENNAI');

**SQL Commands / Statements:-**

**1. DDL ( Data Definition Language )**

**2. DML ( Data Manipulation Language )**

**3. TCL ( Transaction Control Langauge )**

**4. DCL ( Data Control Language )**

**5. DQL ( Data Query Language )**

**SQL Commands / Statements:-**

**1. DDL ( Data Definition Language )**

**1. CREATE**

**2. RENAME**

**3. ALTER**

**4. TRUNCATE**

**5. DROP**

CREATE DATABASE COLLAGE;

USE COLLAGE;

CREATE TABLE STUDENT( ID VARCHAR(10) PRIMARY KEY ,

NAME VARCHAR(50) NOT NULL, CITY VARCHAR(50) NOT NULL);

SELECT \* FROM STUDENT;

INSERT INTO STUDENT ( ID, NAME , CITY )

VALUES ( 102, 'BALAJI','PUNE'),

( 103, 'SANKET','NAGPUR'),

( 104, 'SUMIT','WARDHA'),

( 105, 'NIKHIL','NAGPUR'),

( 106, 'SACHIN','MUMBAI'),

( 107, 'VIRAT','BANGLORE'),student

( 108, 'VINAY','NAGPUR'),

( 109, 'SIMON','HYDERABAD'),

( 110, 'PETER','CHENNAI');

**SQL Commands / Statements:-**

**1. DDL ( Data Definition Language )**

**1. CREATE :-**

**#. This statement is used to create an object in the database.**

**Example:- We can create a table.**

**2. RENAME**

**#. This Statement is used to change the name of the object/table which is already existing.**

**Syntax:- RENAME TABLE OLD\_TABLE\_NAME TO NEW\_TABLE\_NAME;**

**Example:- RENAME TABLE STUDENT TO STUDENTINFO;**

**ALTER**

**#. This statement is used to modify the table structure.**

**#. Add, Modify and Drop Columns.**

**#. Change the Datatypes**

**#. Add Constratins like Primary Key etc.**

**ALTER TABLE STUDENTINFO ADD MobileNumber VARCHAR(20);**

**DESC STUDENT;**

**SELECT \* FROM STUDENT ;**

**ALTER TABLE STUDENT MODIFY MOBILENUMBER VARCHAR(20) UNIQUE;**