**Day-01**

**24-02-2025**

**Datatypes**

**===================**

SQL Syntax:

create table Employee(

empId number,

empName varchar(30),

salary number

);

**Datatypes:**

**=======**

-> The main important data formats in SQL are:

1) Number format

2) Text Format

3) Date-Time format

When we have n-number bits, then the range of values to store are:

-2^(n-1) to 2^(n-1) - 1

**1) Number format**

**============**

-> number format includes:

1) Integral types

------------------

-> The numbers with no decimal point

-> Four types:

i) Tiny int

------------

-> 1-byte => 8-bits

-> range ==> -2^7 to 2^7 - 1 ==> -128 to 127

ii) small int

------------

-> 2-bytes

-> range ==> -2^15 to 2^15 - 1 ==> -32768 to 32767

iii) int

--------

-> 4-bytes

-> range ==> -2^31 to 2^31 - 1

iv) big int

-----------

-> 8-bytes

-> range ==> -2^63 to 2^63 - 1

Ex:

create table employee(

sno tinyint,

age tinyint,

empId smallint,

pin int,

mobile bigint

);

2) Floating-point types

--------------------------

-> numbers with decimal point.

-> three types:

1) float

--------

-> when a floating-point number needs to define with up to 16 places before and after the decimal point, we can use "float".

Ex: salary float;

56000.7910

2) double

---------

-> when a floating-point number needs to define with up to 32 places before and after the decimal point, we can use "double".

Ex: bankBalance double;

3) decimal

----------

-> when a floating-point number needs to define with up to 32 places before and after the decimal point, we can use "decimal".

-> for the decimal value, we cannot apply the approximation.

round()/approx()

round(9.7)/approx.(9.7) ==> 10

round(9.2)/approx.(9.2) ==> 9

**2) Text Format:**

**===========**

-> Can always allowed to define with single quote or double quotes.

-> can define in three ways:

1) char()

---------

-> When we can create a column with char type,

each character holds the memory of 1-byte.

Syntax:

column-name char(size);

Here:

size ==> positive (>0)

When we have specified the size of the column data, according to the specified size the memory can be created.

Ex: empName char(30);

empName = "Ravi Kumar";

Here:

the specified size ==> 30

So, the memory can be created as: 30-bytes

into this, only 9 characters (9-bytes) can be used to store and remaining are to be wasted.

-> modifications of the data bit faster.

2) varchar()

------------

-> When we can create a column with varchar type,

each character holds the memory of 1-byte.

Syntax:

column-name varchar(size);

Here:

size ==> positive (>0)

The varchar is not based on the memory which has specified. It is always based on the value what we have assigned.

Ex: empName varchar(30);

empName = "Ravi Kumar";

-> From the above:

created memory ==> 9-bytes for total 9-chars

-> modification of the data is slower.

3) text

=======

-> huge amount of text data can be represented with "text" type.

**Day-02**

**25-02-2025**

**==============**

user-name char(30);

user-name = "Ashok IT";

user-name varchar(30);

user-name = "Ashok IT";

**Date-time format:**

**============**

-> can be used to store the column fields with date formatted value or time formatted value or both.

-> can be defined with three datatypes:

1) date

2) time

3) datetime

ex:

create table Student(

stuid smallInt,

stuname varchar(30),

gender char(6),

dob date,

examTime time,

resultedOn datetime,

);

dob = 'yyyy-mm-dd';

ex: dob = '1993-06-20';

examTime = 'HH:MM:SS';

examTime = '15:00:00';

resultedOn = 'YYYY-MM-DD HH:mm:SS';

Ex: resultedOn = '2025-02-20 17:10:10';

---------------------------------------------------------

**Binary format**

**=========**

-> Normally the files can be categorized into two types:

1) Text Files

--------------

-> can store the data in the text format

Ex: .txt files

2) Binary Files

---------------

-> the data other than the text in the files are called as "Binary files".

-> The binary data includes: images, zip files, audio files, video files etc.,

-> The binary formatted data can be represented using two different datatypes:

1) binary() -> fixed length

2) varbinary() -> variable length

Ex:

create table Files(

fid smallInt;

fileData binary(16);

fileData1 varbinary(521);

);

SQL Comments

=============

-> SQL comments can be used for documentation purpose.

-> SQL comments can increase the readability of the program.

-> comments can allow to write in any where of the program.

-> There are two ways for commenting:

1) Single line commenting -> --

2) Multi-line commenting --> /\* \*/