**#Assignment23.2**

**Que-1) Explain Primary and complex data types in HIVE with example.**

**PRIMARY DATA TYPES:**

Primary Data Types are further classified into four categories. They are:

**1**. Numeric Type

**2**. String Type

**3**. Date/Time Type

**4**. Miscellaneous Types

**Numeric Data Types**

1. Integral types -int ,bigint,tinyint,smallint
2. DECIMAL (5, 2) states out of five digits 2 are decimals.
3. Equivalent to Java’s byte, short, int, and long primitive types
4. Floating types are double, decimal and float
5. Equivalent to Java’s float and double, and SQL’s Decimal .

**String Data Types**

String is expressed with single quotes (') or double quotes (")

1. VARCHAR

Varchar type created with a length specifier that defines the maximum characters to be allowed in the string which is 1 to 65355.

2. CHAR

Char types is very similar to Varchar but they are fixed-length meaning that values shorter than the defined value are coupled with spaces but trailing spaces are not significant.

**Date/Time Types**

Hive provides DATE and TIMESTAMP data types in traditional UNIX time stamp format for date/time related fields in hive.

1. DATE is represented in the form YYYY-MM-DD. Example: DATE ‘2017-12-05’.

The allowed range of date is 0000-01-01 to 9999-12-31.

2. TIMESTAMP use the format yyyy-mm-dd hh:mm:ss [.f…].

Example- 2017-10-11 23:12:55PM

**Miscellaneous Types**

Hive supports primitive data types like Boolean and binary which are very much similar to JAVAs data type but used to store true and false values in hive.

**COMPLEX DATA TYPES:**

**MAP**

MAP is a collection of key value pair and is represented by

<Primitive type, datatype>. To access the fields we use array notations of key.

**STRUCT**

• It is very much similar to STRUCT in c, it is basically a record which combines a set of named field of primitive data type. The elements are accessed using dot (.)

STRUCT<col\_name: datatype [COMMENT col\_comment],>

Example – For a column z of type STRUCT {x INT; y INT} the a field is accessed by the expression z.x

**UNIONTYPE**

It is similar to union in c, it can hold any one data type from its specified data types.

UNIONTYPE<datatype, datatype>