**Assignment 23.2**

**DATATYPES IN HIVE**

There are 2 types of Datatypes in Hive

1. Primary Datatype
2. Complex Dataype

**Primary Datatype:**

There are 4 types of primary datatypes in hive

1. Numeric Datatype
2. String Types
3. Date/Time type
4. Miscellaneous Types

**Numeric Datatype**

2 TYPES

1. **Integral types**

It includes TINYINT, SMALLINT, INT & BIGINT which is equivalent to byte, short, int and long in java respectively.

1. **Floating types**

It includes FLOAT, DOUBLE & DECIMAL which is equivalent to float double and Decimal in Java

**String Data Types**

It is a sequence of characters enclosed within single or double quotes

**Eg :”somanath”,’somanath’**

**VARCHAR**

Varchar types are created with a length specifier (between 1 and 65355), which defines the maximum number of characters allowed in the character string.

**CHAR**

Char types are similar to Varchar but they are fixed-length meaning that values shorter than the specified length value are padded with spaces but trailing spaces are not important during comparisons.

For example if we specify the max length of the field (example: name VARCHAR(64)). If the values are less than the max length specified then the remaining space will be freed out.

But the maximum length of CHAR is 255

**Date/Time type:**

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

DATE values are represented in the form YYYY-MM-DD. Example: DATE ‘2014-12-07’.

Date ranges allowed are 0000-01-01 to 9999-12-31.

TIMESTAMP use the format yyyy-mm-dd hh:mm:ss[.f...] from day of 1970 Jan 1

We can also cast the String, Time-stamp values to Date format if they match format.

**Miscellaneous Types:**

**2 types of miscellaneous types**

1. Boolean
2. Binary

**Boolean:**

It is similar to Boolean in java and supports true or false value

**Binary:**

BINARY is an array of Bytes and similar to VARBINARY in many RDBMSs

**COMPLEX DATA TYPES:**

There are 4 types of complex data types in hive

1. ARRAY
2. MAP
3. STRUCT
4. UNIONTYPE

**ARRAY:**

It is an ordered collection of elements of the same datatype

It is similar to array in java

Syntax:

**Age array<int>** - which represents Age is an array OF integer datatype

It can accessed by using index

Eg: for accessing 1 value of age give **Age[0]**

**MAP:**

It is an unordered collection of key-value pairs.

Keys must be of primitive types.

Values can be of any type.

**eg:** feature map<string,boolean>-where feature is of map datatype whose key is string and value is boolean

Maps are accessed by using their keys.

**STRUCT:**

It is a collection of elements of different types. We can use any data type to specify this struct data type.

Elements in STRUCT type are accessed using the DOT (.) notation.

**UNIONTYPE:**

UNIONTYPE is collection of Heterogeneous data types.

It is similar to Unions in C.

At any point of time, a Union Type can hold any one (exactly one) data type from its specified data types

Example UNIONTYPE<int, double, array<string>, struct<a:int,b:string>>