1. Explain Primary data types and complex data types in Hive with an example in brief.

The primary data types supported by Hive are listed below:

a. Primary Data type

1. Numeric Types

* TINYINT (1-byte signed integer, from -128 to 127)
* SMALLINT (2-byte signed integer, from -32,768 to 32,767)
* INT (4-byte signed integer, from -2,147,483,648 to 2,147,483,647)
* BIGINT (8-byte signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
* FLOAT (4-byte single precision floating point number)
* DOUBLE (8-byte double precision floating point number)
* DECIMAL (Hive 0.13.0 introduced user definable precision and scale)

2. Date/Time Types

* TIMESTAMP
* DATE

3. String Types

* STRING
* VARCHAR
* CHAR

4. Misc Types

* BOOLEAN
* BINARY

Apart from these primitive data types Hive offers some complex data types which are listed below: 

b. Complex Types

* arrays: ARRAY<data\_type>
* maps: MAP<primitive\_type, data\_type>
* structs: STRUCT<col\_name : data\_type [COMMENT col\_comment], ...>
* union: UNIONTYPE<data\_type, data\_type, ...>

Hive Presently supported 4 complex data types

* ARRAY
* MAP
* STRUCT
* UNIONTYPE

**ARRAY Data Type**

Same as Array in java,An Ordered sequences of similar type elements that are index using zero-based integers

**Example                      array(‘John’, ‘Doe’)**

The second value is accused by using array[1]

**MAP Data Type**

Collection of key-value pairs. Fields are accessed using array notation of keys (e.g., [‘key’]).

**Example**                      map(‘first’, ‘John’,’last’, ‘Doe’)

if a column  name is of type MAP with key→value pairs  ‘first’→’John’ and ‘last’→’Doe’, then the last  name can be referenced using name[‘last’].

**STRUCT Data Type**

* It is similar to STRUCT in C language. It is a record type which encapsulates a set of named fields that can be any primitive data type. Elements in STRUCT type are accessed using the DOT (.) notation.

**Example** – For a column **c** of type STRUCT {a INT; b INT} the **a** field is accessed by the expression **c.a**

**UNIONTYPE**

UNIONTYPE is collection of Heterogeneous data types.It is similar to Unions in C. At any point of time, an 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>>