Describe complex columns

Author: Shreelekhya Gampa

Version: v1

Background: Currently describe formatted displays the column information of a table and some additional information. When complex types such as ARRAY, STRUCT, and MAP types are present in the table, column definition can be long and it's difficult to read in nested format.

Solution:

For complex types available, the DESCRIBE output can be formatted to avoid long lines for multiple fields. We can pass the complex field name to the command and visualize its structure as if it were a table.

[DESCRIBE | DESC] column fieldname[.nestedFieldNames] ON [db_name.]table_name;

In order to get the high level view of column datatypes of a table, add option 'short' to the command. This does not display all nested levels and datatypes, shows only the outermost level.

[DESCRIBE | DESC] short [db_name.]table_name;

Example:

The below table is 'complexcarbontable' which consists of different complex columns.

```
create table complexcarbontable
(deviceInformationId int,
channelsId string,
ROMSize string,
purchasedate string,
mobile struct<imei:string, imsi:string>,
MAC array<string>,
```

locationinfo array<struct<ActiveAreaId:int, ActiveCountry:string, ActiveProvince:string, Activecity:string, ActiveDistrict:string, ActiveStreet:string, ActiveNeighbourhoodPlaces:array<string>>>,
proddate struct<productionDate:string,activeDeactivedate:array<string>>,
gamePointId map<string,struct<a:int, b:struct<c:string>>>,
contractNumber double,
decimalColumn map<string,struct<d:decimal(10,3)>>)
STORED AS carbondata;

When DESCRIBE specifies the table name.

describe complexcarbontable;

col_name	data_type
deviceinformationid	int
channelsid	string
romsize	string
purchasedate	string
mobile	struct<imei:string,imsi:string>
mac	array<string>
locationinfo	array<struct<activeareaid:int,activecountry:string,activeprovince:string,activecity:string,activedi
proddate	struct<productiondate:string,activedeactivedate:array<string>>
gamepointid	double
contractnumber	double
decimalcolumn	map<string,struct<d:decimal(10,3)>>

1. Column having struct with some primitive child elements: struct<>

   To read a simple structure of mobile column,

   Describe column mobile on complexcarbontable;

```
+-----------------------------+-----------------------------+
|col_name                     |data_type |comment|          |
+-----------------------------+-----------------------------+
|mobile                       |struct       |null    |
|## Children of channelsId:   |             |
|imei                         |string       |null    |
|imsi                         |string       |null    |
+-----------------+-------------------------------------+
```

2. Column having 2 nested levels: map<key, struct<>>

When DESCRIBE specifies the name of the complex column, *decimalcolumn*, can show how an MAP is represented with columns key and value. Here STRUCT is present in nested level of MAP .

Describe column decimalcolumn on complexcarbontable;

```
+-------------------------------+---------------------------------------
|col_name                       |data_type                      |comment|
+-------------------------------+---------------------------------------
|decimalcolumn                  |map                            |null    |
|## Children of decimalcolumn:  |                               |        |
|key                            |string                         |null    |
|value                          |struct<d:decimal(10,3)>        |null    |
+-------------------------------+---------------------------------------
```

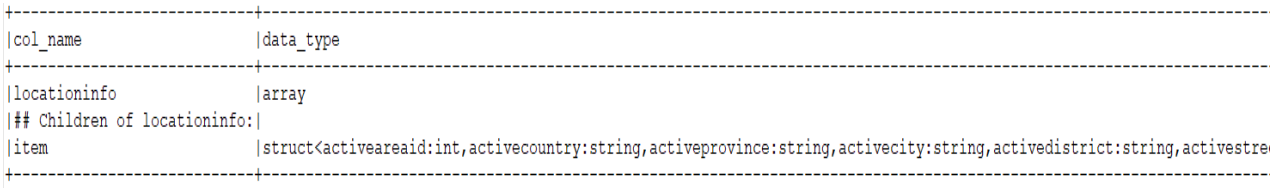
Describe column decimalcolumn.value on complexcarbontable;

```
+-----------------------------------+-----------------------------------------
|col_name                           |data_type                      |comment|
+-----------------------------------+-----------------------------------------
|decimalcolumn.value                |struct                         |null    |
|## Children of decimalcolumn.value:|                               |        |
|d                                  |decimal(10,3)                  |null    |
+-----------------------------------+-----------------------------------------
```

3. Columns with 3 nested levels :
    a. array<struct<array<>>>

When DESCRIBE specifies the name of the complex column, *locationinfo*, can show how an ARRAY is represented with column ITEM.

Describe column locationinfo on complexcarbontable;



col_name	data_type
locationinfo	array
## Children of locationinfo:	
item	struct<activeareaid:int,activecountry:string,activeprovince:string,activecity:string,activedistrict:string,activestre

To see the further nested structure of column, when DESCRIBE specifies the name of the ITEM of the ARRAY column, as the ARRAY contains a STRUCT, the layout of the STRUCT is shown.

Describe column locationinfo.item on complexcarbontable;

col_name	data_type	comment
locationinfo.item	srtuct	null
## Children of locationinfo.item:		
activeareaid	int	null
activecountry	string	null
activeprovince	string	null
activecity	string	null
activedistrict	string	null
activestreet	string	null
activeneighbourhoodplaces	array<string>	null

To read 3rd nested level of locationInfo column, we can use

Describe locationinfo.item.activeneighbourhoodplaces on complexcarbontable;

col_name	data_type	comment
locationinfo.item.activeneighbourhoodplaces	array	null
## Children of locationinfo.item.activeneighbourhoodplaces:		
item	string	null

b. Column having 3 nested levels: map<struct<struct<>>>

When DESCRIBE specifies the name of the complex column, *gamePointId*, can show how an MAP is represented with columns key and value.

Describe column gamepointid on complexcarbontable;

col_name	data_type	comment
key	string	null
value	struct<a:int, b:struct<c:string>	null

Describe column gamepointid.value on complexcarbontable;

col_name	data_type	comment
a	int	null
b	struct<c:string>	null

Describe column gamepointid.value.b on complexcarbontable;

col_name	data_type	comment
c	string	null

4. Display short version of table columns.

Describe short complexcarbontable;

col_name	data_type	comment
deviceinformationid	int	null
channelsid	string	null
romsize	string	null
purchasedate	string	null
mobile	struct<..>	null
mac	array<..>	null
locationinfo	array<..>	null
proddate	struct<..>	null
gamepointid	double	null
contractnumber	double	null
decimalcolumn	map<..>	null