

Capstone Project: Instant Health Alert System - Mid-Project Submission

Building an External Hive Table for Threshold Data

Introduction:

This document provides an explanation of the Hive script used to build an external table for storing threshold data. The table integrates with HBase, allowing seamless access to real-time data updates and facilitating health monitoring system alerts.

1. Create and Use Database

The code starts by creating a new database named `health` and setting it as the active database. This ensures that the subsequent table creation occurs within the `health` database, keeping all health-related data organized.

```
create database health;  
use health;
```

```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false  
hive> create database health;  
OK  
Time taken: 0.48 seconds  
hive> use health;  
OK  
Time taken: 0.02 seconds  
hive> |
```

2. Create External Table `threshold_reference_table`

This section creates an external Hive table named `threshold_reference_table`. External tables are used when data is stored outside of Hive's own data warehouse; in this case, the table is linked to an HBase table, which facilitates real-time access and modifications to the data.

```
CREATE EXTERNAL TABLE threshold_reference_table (  
    key int,  
    Attribute string,  
    low_age_limit int,  
    high_age_limit int,
```

```

Low_Range_Value int,
High_Range_Value int,
Alert_Flag int,
Alert_Message string
]

STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH SERDEPROPERTIES (
'hbase.columns.mapping' = ':key, attribute:attribute, limit:low_age_limit, limit:high_age_limit,
limit:low_value,
limit:high_value, alert:alert_flag, alert:alert_message',
'hbase.table.name' = 'threshold_ref'
)

TBLPROPERTIES ('hbase.mapred.output.outputtable' = 'threshold_ref');

```

```

hive> CREATE EXTERNAL TABLE threshold_reference_table (
>   key int,
>   Attribute string,
>   low_age_limit int,
>   high_age_limit int,
>   Low_Range_Value int,
>   High_Range_Value int,
>   Alert_Flag int,
>   Alert_Message string
> )
> STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
> WITH SERDEPROPERTIES (
>   'hbase.columns.mapping' = ':key, attribute:attribute, limit:low_age_limit, limit:high_age_limit, limit:low_va
> limit:high_value, alert:alert_flag, alert:alert_message',
>   'hbase.table.name' = 'threshold_ref'
> )
> TBLPROPERTIES ('hbase.mapred.output.outputtable' = 'threshold_ref');
OK
Time taken: 2.085 seconds

```

3. HBase Storage Properties

This section specifies that the table data will be managed using the HBase storage handler, `org.apache.hadoop.hive.hbase.HBaseStorageHandler`. The handler provides integration between Hive and HBase, allowing Hive to query and manage HBase-stored data seamlessly.

```
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
```

4. SERDE Properties - Column Mappings

SERDE properties define how Hive columns map to HBase columns. The `hbase.columns.mapping` SERDE property specifies the mapping of each Hive column to its respective HBase column. For instance, `ref_id` is mapped as the HBase row key (`:key`), while other columns are mapped to HBase column families `attribute`, `limit`, and `alert`.

```
WITH SERDEPROPERTIES (  
  "hbase.columns.mapping" = ":key,  
  attribute:attribute,  
  limit:low_age_limit,  
  limit:high_age_limit,  
  limit:low_range_value,  
  limit:high_range_value,  
  alert:alert_flag,  
  alert:alert_message"  
)
```

5. Table Properties - HBase Table Linkage

The TBLPROPERTIES section links the Hive table to an existing HBase table named `threshold_reference_table`. This setup directs Hive to use the HBase table as the underlying data source for the `threshold_reference_table` Hive table.

```
TBLPROPERTIES ("hbase.table.name" = "threshold_reference_table");
```

```
hive> select * from health.threshold_reference_table order by key;
Query ID = root_20241101090720_019620f5-5216-4ea2-9690-e6c4477e52ef
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1730449667648_0007)
```

VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	1	1	0	0	0	0
Reducer 2	container	SUCCEEDED	1	1	0	0	0	0

```
VERTICES: 02/02 [=====] 100% ELAPSED TIME: 7.33 s
```

```
OK
1 heartbeat 0 40 0 69 1 Low Heart Rate than Normal
2 heartbeat 0 40 70 78 0 Normal
3 heartbeat 0 40 79 9999 1 Higher Heart Rate than Normal
4 bp 0 40 0 160 1 Low BP than Normal
5 bp 0 40 161 220 0 Normal
6 bp 0 40 221 9999 1 Higher BP than Normal
7 heartbeat 41 100 0 65 1 Low Heart Rate than Normal
8 heartbeat 41 100 66 73 0 Normal
9 heartbeat 41 100 74 9999 1 Higher Heart Rate than Normal
10 bp 41 100 0 150 1 Low BP than Normal
11 bp 41 100 151 180 0 Normal
12 bp 41 100 181 9999 1 Higher BP than Normal
Time taken: 13.131 seconds, Fetched: 12 row(s)
```