Database Design Document

Introduction:

This document will give you the overall information on the queries being used in ETL process.

Database:

Database (MySQL): **8.0.23**

I have used MySQL because it offers high throughput in terms of Insert/Read Operations.

Queries:

**(At) API Layer:**

Table Name: **EVENTS** (this table stores event data at API end)

Table Structure:

CREATE TABLE **EVENTS** (

EVENT VARCHAR(16) NOT NULL,

PROPERTIES\_TIME TIMESTAMP NOT NULL,

PROPERTIES\_UNIQUE\_VISITOR\_ID VARCHAR(36),

PROPERTIES\_BROWSER VARCHAR(16),

PROPERTIES\_OS VARCHAR(16),

PROPERTIES\_HA\_USER\_ID VARCHAR(16),

PROPERTIES\_COUNTRY\_CODE VARCHAR(16)

)

Query: SELECT \* FROM EVENTS WHERE EVENT='event\_1' AND PROPERTIES\_TIME BETWEEN '2020-10-21 06:36:00' AND '2020-10-22 09:36:00'

**(At) Pipeline Layer:**

The JSON response from API is store in json file at pipeline layer at data lake. The spark jobs gets trigger as soon as the data is ready and read the json data and store in to spark SQL (temp SQL tables). That is transformation of Semi Structured data to Structured data. The same task will write the data from temp table to main table.

DF Table Name(Temp): **df\_events**

Main Table Name: events\_incremental

-- main table structure

CREATE TABLE EVENTS\_INCREMENTAL (

EVENT VARCHAR(16) NOT NULL,

PROPERTIES\_TIME TIMESTAMP NOT NULL,

PROPERTIES\_UNIQUE\_VISITOR\_ID VARCHAR(36),

PROPERTIES\_BROWSER VARCHAR(16),

PROPERTIES\_OS VARCHAR(16),

PROPERTIES\_HA\_USER\_ID VARCHAR(16),

PROPERTIES\_COUNTRY\_CODE VARCHAR(16)

);

## getting the file from data lake, reading using spark json module and creating a spark df ##

file\_path = os.path.join(os.path.join(os.getcwd(),"plugins/datalake/etl\_api"),input\_file\_name)

df = spark.read.json(file\_path)

df.createOrReplaceTempView(DF\_table\_name)

Query: """SELECT

event,

properties.time as properties\_time,

properties.unique\_visitor\_id as properties\_unique\_visitor\_id,

properties.browser as properties\_browser,

properties.os as properties\_os,

properties.ha\_user\_id as properties\_ha\_user\_id,

properties.country\_code as properties\_country\_code

FROM

{table}"""

**Reading the data from spark temp table**

df\_data = spark.sql(query)

**Writing data to load table** df\_data.write.format("jdbc").option("url",jdbc\_url).option("dbtable",main\_table\_name).mode('append').save()

Data Insights Queries

1. how many users have visited the website in a day (distinct users)? (Website considering housinganywhere.com)

*SELECT COUNT (DISTINCT PROPERTIES\_HA\_USER\_ID) NO\_OF\_USERS, PROPERTIES\_TIME FROM EVENTS GROUP BY DATE(PROPERTIES\_TIME);*

1. how many different browsers accessing our website in a day?

*SELECT DISTINCT (PROPERTIES\_BROWSER), DATE(PROPERTIES\_TIME) FROM EVENTS ORDER BY PROPERTIES\_TIME;*

1. Popular browsers/OS in country?

*SELECT PROPERTIES\_BROWSER, PROPERTIES\_COUNTRY\_CODE, COUNT(1) FROM EVENTS*

*WHERE PROPERTIES\_BROWSER IS NOT NULL*

*GROUP BY PROPERTIES\_BROWSER, PROPERTIES\_COUNTRY\_CODE*

*ORDER BY COUNT(1) DESC, PROPERTIES\_BROWSER, PROPERTIES\_COUNTRY\_CODE;*

1. How many times does a customer visit the website in a day/throughout?

*SELECT DATE(PROPERTIES\_TIME), PROPERTIES\_HA\_USER\_ID, COUNT(1) FROM EVENTS*

*GROUP BY DATE(PROPERTIES\_TIME), PROPERTIES\_HA\_USER\_ID*

*ORDER BY DATE(PROPERTIES\_TIME), COUNT(1) DESC;*

1. The number of active user’s country wise per day?

*SELECT PROPERTIES\_COUNTRY\_CODE, COUNT(1) FROM EVENTS*

*GROUP BY PROPERTIES\_COUNTRY\_CODE*

*ORDER BY COUNT(1) DESC;*

There are so many insights queries can be designed but for now I am mentioning a few above.