UDACITY

‹ Return to Classroom

Data Modeling with Cassandra

REVIEW

HISTORY

Meets Specifications

Congratulations

You have completing this project! Excellent work updating the project as mentioned by the previous reviewer. Your description of queries look good. You should be very proud of your accomplishments in building a Data Model with Cassandra. You'll find feedback and some tips to help you continue to improve on your project.

You have built your foundations in the SQL and NOSQL Databases. In the next few projects in the nanodegree, you will focus more on using these concepts on Cloud. Every organisation is utilising the power of cloud and learning the same will benefit in the long run. Happy learning!

Additional Read:

- Difference between partition key, composite key and clustering key in Cassandra?.
- DataStax documentation you can refer for clustering columns

ETL Pipeline Processing

Student creates `event_data_new.csv` file.

Nice job! I see the `event_data_new.csv` file, which indicates you followed the ETL pipeline to create the csv file.

(The file was generated after running your code in the Udacity workspace)

Student uses the appropriate datatype within the `CREATE` statement.

Nice work! You have used appropriate datatype within the Create Statement.

- You can check this link for cql data types.

Data Modeling

Student creates the correct Apache Cassandra tables for each of the three queries. The `CREATE TABLE` statement should include the appropriate table.

Great job! You followed the `one table per query` rule of Apache Cassandra. You are not replicating the same table for all three queries, which defies that rule. You have `three distinct tables` with `unique tables names` and uses appropriate CREATE table statements.

Student demonstrates good understanding of data modeling by generating correct SELECT statements to generate the result being asked for in the question.

The SELECT statement should NOT use `ALLOW FILTERING` to generate the results.

Awesome work! Your data is modeled correctly to generate the exact responses posed in the questions.

Student should use table names that reflect the query and the result it will generate. Table names should include alphanumeric characters and underscores, and table names must start with a letter.

We are looking for table names that provide a good general sense of what this query will generate. For e.g., for query 2, an appropriate table name should reflect song playlist in session (e.g., name could be song_playlist_session). You should not be using table names like query 1 or project 1, etc.

You have used table names that reflect the query and the result it will generate. Table names include alphanumeric characters and underscores, and table names start with a letter.

We are always concerned about correct table schemas. It becomes a lot difficult to manage and work on tables if the names are like table1,table2 and table3. One of my day-to-day work also includes creating multiple tables in production. Having thousands of tables, it's really handy to give them some name that we can understand and that relates to the query/business requirement.

The sequence in which columns appear should reflect how the data is partitioned and the order of the data within the partitions.

This is probably one of the most important learnings that I want you to walk away with from this lesson. Apache Cassandra is a partition row store, which means the partition key determines which node a particular row is stored on. With the Primary key (which includes the Partition Key and any clustering columns), the partitions are distributed across the nodes of the cluster. It determines how data are chunked for write purposes. Any clustering column(s) would determine the order in which the data is sorted within the partition.

The sequence of the columns in the CREATE/INSERT/SELECT statements should follow the order of the COMPOSITE PRIMARY KEY and CLUSTERING columns. The data should be inserted and retrieved in the same order as how the COMPOSITE PRIMARY KEY is set up.

Here is some DataStax documentation you can refer:

- Datastax Documentation on clustering

PRIMARY KEYS

The combination of the PARTITION KEY alone or with the addition of CLUSTERING COLUMNS should be used appropriately to uniquely identify each row.

The combination of the PARTITION KEY alone or with the addition of CLUSTERING COLUMNS looks good. Great!

Presentation

The notebooks should include a description of the query the data is modeled after.

Good work including descriptions of the query! It was clear how each query was being addressed. You can provide heading as well.

Code should be organized well into the different queries. Any in-line comments that were clearly part of the project instructions should be removed so the notebook provides a professional look.

TODO's have been removed. Great!

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START