CMPU4003 Advanced Databases

Continuous Assessment

Part II (Deadline Thursday 14th December 2023 @ 16.00)

Task Overview

1. In CouchDB:
    - Set up replication and partitioning.
    - Port a subset of your data from MariaDB into linked documents.
    - Implement a global query against the linked documents.
    - Implement a partitioned query against one type of document.
2. In Cassandra:
    - Setup a Cassandra cluster implementing replication and partitioning.
    - Port a subset of your data from MariaDB into Cassandra.
    - Implement and tune the performance of query against this data.
    - Create a new table including a collection datatype.
    - Implement and tune the performance of materialized view against this table.
3. You will be required to demonstrate some aspects of your submission.
4. You can demonstrate these at any time before the final deadline.
5. Refer to Task Detail for specifics of what is required for each aspect and how it will be assessed.

CouchDB Task Details

1. Create a new partitioned database.
    - Include your student number in the name of the database.
2. Using the MariaDB dimensional model you implemented for Part I of the CA:
    - Port data from your dimensional model implemented in MariaDB into CouchDB to the master database in the form of linked documents.
        - You need to port the fact data for two counties to a fact document.
        - You also need to port the data for one of the dimensions to another document.
        - The fact document must be linked to the dimensional document.
    - You must divide the data so that some documents are allocated to different partitions.
        - You are expected to have at least TWO partitions.
        - For example:
            - Documents from county 1 on one partition, documents from county 2 on different partition.
            - You need to decide where the associated dimensional data is stored.
    - You are expected to provide a JSON file containing an export of this data from CouchDB so that your submission can be validated.
3. Implement master slave replication on this database.
    - The replica must be partitioned.
    - The replica should also include your student number in the name of the database.
    - You should use a selector in the replication documents.
        - This should ensure that only a subset of the fact and dimension documents are replicated.
    - You must demonstrate that replication is working.
    - Provide the replication document that implements the replication including the selector.
4. Create a design document and view to execute a global query against this database to access data in both the fact and dimension documents.
    - Your query must access content from the fact document plus data from one of the linked dimension documents.
    - Provide the design document.
5. Create a design document and view to execute a query against a partition to access data in one of the document types.
    - Provide the design document.

Cassandra Task Details

1. Setup a Cassandra cluster with three nodes.
2. Create a keyspace in that cluster named with your student number using simplestrategy replication.
   - Provide the CQL to create the keyspace.
3. Using the MariaDB dimensional model you implemented for Part I of the CA:
   - Port the fact data for two counties plus the associated dimensional data from MariaDB into Cassandra implementing appropriate partitioning and clustering.
   - You are expected to provide a CSV file containing an export of this data from Cassandra.
   - Provide the CQL needed to create the table so that your submission can be validated.
4. Implement a query against this data and tune the performance of this query.
   - Tuning should involve implementing indexes.
   - Be able to demonstrate the impact of performance tuning.
   - Provide CQL to implement the query, tune the performance.
   - Provide CQL/additional code needed demonstrate the performance enhancement.
5. Create and populate a new table including a column of a collection datatype.
   - You are expected to provide a CSV file containing an export of this data from Cassandra plus the CQL needed to create the table so that your submission can be validated.
   - Provide CQL to create the table.
6. Implement and tune the performance of materialized view against this table.
   - Tuning should target the column of collection datatype.
   - Primary key and where clause should be targeted.
   - Provide CQL to implement the view, tune the performance.
   - Provide CQL/additional code needed to demonstrate performance.

Demonstration

You can demonstrate aspects of your work at any time in the labs between now and Thursday 14th December @ 16.00.

- **You can demonstrate aspects of the submission as and when you have completed it).**

What needs to be demonstrated:

- CouchDB
  - A correctly named partitioned database has been created and populated with linked documents.
  - Master-Slave replication has been implemented using a correctly named replication database using a selector and is working.
  - A global query has been implemented and is working retrieving data from linked documents.
  - A partition-based query has been implemented and is working.
- Cassandra
  - A Cassandra cluster with three nodes has been created.
    - Data has been ported to the cluster and replication is working (verified by accessing data on multiple nodes).
    - A query has been implemented against this data and performance has been tuned.
    - A second table has been implemented using the collection datatype.
    - A materialized view has been implemented using the collection datatype performance has been tuned.

Submission

- Submissions must be made by the deadline of Thursday 16th December @ 16:00.
- You need to submit using Brightspace using the assignment **CA Part II Submission.**
- You need to **SUBMIT A SINGLE ARCHIVE (.ZIP, .RAR, .7Z)** named with your student number**,** e.g. D123456.zip A single archive file (.zip, .rar etc) named with your student number (e.g. D123456.zip) containing the following:
  - CouchDB
    - A JSON file containing the data from CouchDB.
    - The replication document including the selector.
    - The design document for the global query.
    - The design document for the partition-based query.
    - A brief Readme file outlining what your global query and your partition-based query aim to achieve plus any additional information you consider necessary for your submission to be assessed.
  - Cassandra
    - A CSV file containing the base data from Cassandra.
    - A CSV file containing the data from the table containing a column of datatype collection from Cassandra.
    - A CQL file containing:
      - The CQL to create the required keyspace.
      - The CQL needed to create the base table.
      - The CQL to create the query against the data.
      - The CQL used to create the indexes used to tune performance of this query.
      - The CQL used to create the second table.
      - The CQL used to create the materialized view.
      - The CQL used to tune the performance of the materialized view.
      - Any CQL/other commands used to demonstrate the performance enhancement.
    - A brief Readme file outlining what your query and your materialized view aim to achieve plus any additional information you consider necessary for your submission to be assessed.

MARKING SCHEME

This part of the assessment will be marked out of 100 but weighted to 20% of the module marks.

Marking Breakdown	**Assessed by Demo**	**Assessed by Submission**
Setup and populate CouchDB implementing replication and partitioning	5 marks	5 marks (for data extract plus replication document with selector)
Implement a global query against the linked documents.	3 marks	12 marks
Implement a partitioned query against one type of document.	2 marks	8 marks
Setup and populate a Cassandra cluster implementing replication and partitioning.	3 marks	12 marks (for data extract plus CQL to create keyspace and base table)
Implement and tune the performance of query against this data using indexes.	5 marks	15 marks
Create a new table including a collection datatype.	2 marks	8 marks
Implement and tune the performance of materialized view against this table using primary key and where clause.	5 marks	15 marks
Total Marks	**25 marks**	**75 marks**