CS C352 Database Systems

Assignment on No-SQL Database

Due Date: 26th April

Assignment Objective:

To expose students to a new class of databases called No-SQL databases. These databases have become very popular lately mainly because of their ability to handle Big Data. Through the assignment, students will be able to compare them with the conventional yet most prevalent RDBMSs.

1. About No-SQL Databases

What is NoSQL?

NoSQL encompasses a wide variety of different database technologies and were developed in response to a rise in the volume of data stored about users, objects and products, the frequency in which this data is accessed, and performance and processing needs. Relational databases, on the other hand, were not designed to cope with the scale and agility challenges that face modern applications, nor were they built to take advantage of the cheap storage and processing power available today.

NoSQL Database Types

Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Graph stores are used to store information about networks, such as social connections. Graph stores include Neo4J and HyperGraphDB.

Key-value stores are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or "key"), together with its value. Examples of key-value stores are Riak and Voldemort. Some key-value stores, such as Redis, allow each value to have a type, such as "integer", which adds functionality.

Wide-column stores such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows.

The Benefits of NoSQL

When compared to relational databases, NoSQL databases are more scalable and provide superior performance, and their data model addresses several issues that the relational model is not designed to address:

- Large volumes of structured, semi-structured, and unstructured data
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture

Further References:

NoSQL: http://www.mongodb.com/learn/nosql

Bigtable:http://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf

http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2010-08-2/NET-2010-08-2_06.pdf

HBase: https://hbase.apache.org/

HDFS: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

Hadoop: http://hadoop.apache.org/

Mapreduce:
http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf

2. Setting HBase

Install Hbase on at least two nodes for implementation of problem given in section 4 for this assignment.

Installing instructions of Apache HBase in distributed Mode

http://jayatiatblogs.blogspot.in/2013/01/hbase-installation-fully-distributed.html

http://hbase.apache.org/configuration.html

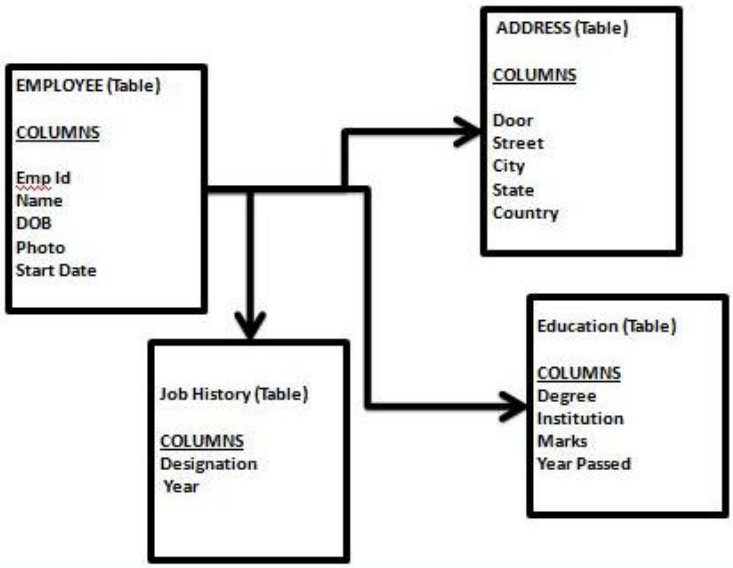
3. Report Submission

Please prepare a handwritten report on differences between SQL and NoSQL databases w.r.t HBase and Oracle RDBMS. Compare on the basis of following attributes:

1. Data access time/ Performance
2. Scalability

3. Physical Storage
4. Complexity
5. Data Availability and Integrity

4. Implementation Problem



Relational Database Design

Convert above ER Diagram in HBase model with following features:

Row key will be represented by Employee ID

Column Family 1 : Basic with Columns (Name, DOB, Photo, Start Date)

Column Family 2 : Address With Columns (Door, Street, City, State, Country)

Column Family 3 : Education With Variable Set of Columns

§ (High School Degree, High School Institution, High School Marks, High School Passed Year)

§ Variable Columns (Graduate Degree, Graduate Institution, Graduate Marks, Graduate Passed Year)

§ Variable Columns (Masters Degree, Masters Institution, Masters Marks, Masters Passed Year)

Column Family 4 : History with Column Job Title and which is Multi Versioned (stored in descending order of timestamp, latest timestamp first)

Logically whole table looks likes the table given below:

Row Key	Time Stamp	Column Family : Basic				Column Family : History	Column Family : Education			Column Family : Address				
		Name	DOB	Photo	Start Date	Job Title	High School	Bachelor	Masters	Door	Street	City	State	Zip
1	T1	John	DD/MM/YY	YY	DD/MM/YY									
1	T3					Sr. System Analyst								
1	T2					Analyst								
1	T2					Programmer								
1	T1						Science							
1	T1							Computers						
1	T1									XX	XX	XX	XX	53353

Key-Values for one column family are stored physically in separate files only. Columns (each column family) are grouped because most of queries are based on one column family only; so we need to read from one file only. Maximum size of file can be 500KB. If while inserting a record file size becomes more than 500 KB, partition the file i.e. store in another file on a different system in your distributed environment (HBase cluster).

Key-value pair will be:

(Empid || Column Family|| Column Name || Timestamp)- (Value)

If timestamp is not entered in key then it should read latest record.

Write programs:

1. To insert records in files.
2. Retrieve records based on key(defined above)
3. Update a value for given key. (Do not update at same place, insert a new value for the same key with latest timestamp.

Group Information:

Maximum 3 students per group.