Group A: Assignment No.1

Title of Assignment: Group A: Introduction to Databases:
1.Study and design a database with suitable example using followingdatabase systems:
• Relational: SQL/ PostgreSQL /MySQL
• Key-value: Riak/ Redis
• Columnar: Hbase
• Document: MongoDB/ CouchDB
• Graph: Neo4J
Compare the different database systems based on points like efficiency, scalability, characteristics and performance.
Objective: To learn and understand different database systems.
Theory:
1.Relational: SQL/ PostgreSQL/ MySQL
RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.
The data in an RDBMS is stored in database objects which are called as tables. This table is basically a collection of related data entries and it consists of numerous columns and rows.
Advantages of SQL :
SQL has many advantages which makes it popular and highly demanded. It is a reliable and efficient language used for communicating with the database. Some advantages of SQL are as follows:

1. Faster Query Processing –
Large amount of data is retrieved quickly and efficiently. Operations like Insertion, deletion, manipulation of data is also done in almost no time.
2. No Coding Skills –
For data retrieval, large number of lines of code is not required. All basic keywords such as SELECT, INSERT INTO, UPDATE, etc are used and also the syntactical rules are not complex in SQL, which makes it a user-friendly language.
3. Standardised Language –
Due to documentation and long establishment over years, it provides a uniform platform worldwide to all its users.
4. Portable –
It can be used in programs in PCs, server, laptops independent of any platform (Operating System, etc). Also, it can be embedded with other applications as per need/requirement/use.
5. Interactive Language –
Easy to learn and understand, answers to complex queries can be received in seconds.
6. Multiple data views –
Disadvantages of SQL :

Although SQL has many advantages, still there are a few disadvantages.
Various Disadvantages of SQL are as follows:
1. Complex Interface –
SQL has a difficult interface that makes few users uncomfortablewhile dealing with the database.
2. Cost –
Some versions are costly and hence, programmers cannot access it.
3. Partial Control –
Due to hidden business rules, complete control is not given to the database.
Applications of SQL :
• SQL is used by developers and DBAs (Database Administrators) inwriting Data Integration Scripts.
• It is used to deal with analytical queries to analyze the data and get instincts from it.
• Retrieving Information
• Modification/Manipulation of data and database table such asInsertion, Deletion and Updation.
2. Key-value: Riak/ Redis

Riak KV is a distributed NoSQL database designed to deliver maximum data availability by distributing data across multiple servers. As long as your Riak KV client can reach one Riak server, it should be able to write data.
Advantages:
Riak's key/value architecture enables it to be more performant than relational databases in many scenarios because Riak doesn't need to perform lock, join, union, or other operations when working with objects. Instead, it interacts with objects on a one-by-one basis, using primary key lookups.
Primary key lookups store and fetch objects in Riak on the basis of three basic locators:
The object's key, which can be anything you want as long as it is Unicode compliant
The bucket which houses the object and its key (bucket names are also Unicode compliant)
The bucket type that determines the bucket's replication and other properties
It may be useful to think of this system as analogous to a nested key/value hash as you would find in most programming languages
Disadvantages:
There are a lot of key-value databases. Bellow there are some general limitations which are in line with the general idea of this database type. For a given database some of them may not be true.
1. The only way to look up values is by key.
2. Range queries are not supported.
3. There is no standard query language comparable to SQL forrelational databases. In consequence queries from one key-value database may not be portable to the other.
Difference between Relational and key-value
1. Simplicity is a key word associated with key-value databases where everything is simple. Unlike in relational databases, there are no tables, so there are no features associated with tables, such as columns and

constraints on columns. If there are no tables, there is no need for joins.
In consequence foreign keys do not exists and so key-value databases do

not support a rich query language such as SQL. Saying the truth, their
query language is very primitive.
The only extra feature supported by some key-value databases are
buckets, or collections. We use them for creating separate namespaces
within a database. Keys from one namespace do not collides with keys
from other so we can use the same keys for more than one namespace.
This can be used to implement something analogous to a relational
schema.
1. Contrary to relational database where meaningless keys are used,the
keys in key-value databases are meaningful
2. While in relational database we avoid duplicating data, in key-value (or in
NoSQL in general) databases it is a common practice.
3.Columnar: Hbase
HBase is a distributed column-oriented database built on top of the Hadoop file
system. It is an open-source project and is horizontally scalable.
HBase is a data model that is similar to Google's big table designed to provide
quick random access to huge amounts of structured data. It leverages the fault
tolerance provided by the Hadoop File System (HDFS).
It is a part of the Hadoop ecosystem that provides random real-time read/write
access to data in the Hadoop File System.
One can store the data in HDFS either directly orthrough HBase. Data consumer
reads/accesses the data in HDFS randomly using HBase. HBase sits on top of
the Hadoop File System and provides read and write access.

HBase and HDFS

HDFS HBase

HDFS is a distributed file
system suitable for storing
large files.

HBase is a database built on top of the HDFS.

HDFS does not support fast
individual record lookups.

HBase provides fast lookups for larger tables.

It provides high latency batch
processing; no concept of
batch processing.

It provides low latency access to single rows
from billions of records (Random access).

It provides only sequential

access of data.

HBase internally uses Hash tables and
provides random access, and it stores the
data in indexed HDFS files for faster lookups.

HBase is a column-oriented database and the tables in it are sorted by row.
The table schema defines only column families, which are the key value pairs.
A table have multiple column families and each column family can have any
number of columns. Subsequent column values are stored contiguously on the
disk. Each cell value of the table has a timestamp. In short, in anHBase:
• Table is a collection of rows.
• Row is a collection of column families.
• Column family is a collection of columns.
• Column is a collection of key value pairs.
Column Oriented and Row Oriented
Column-oriented databases are those that store data tables as sections of
columns of data, rather than as rows of data. Shortly, they will have column
families.

Row-Oriented Database Column-Oriented Database

It is suitable for Online Transaction
Process (OLTP).

It is suitable for Online Analytical
Processing (OLAP).

Such databases are designed for small
number of rows and columns.

Column-oriented databases are
designed for huge tables.

The following image shows column families in a column-oriented database:

HBase and RDBMS

HBase RDBMS

HBase is schema-less, it doesn't have the
concept of fixed columns schema; defines
only column families.//

An RDBMS is governed by its schema,
which describes the whole structure
of tables.

It is built for wide tables. HBase is
horizontally scalable.

It is thin and built for small tables.
Hard to scale.

No transactions are there in HBase. RDBMS is transactional.

It has de-normalized data. It will have normalized data.

It is good for semi-structured as well as
structured data.

It is good for structured data.

Features of HBase
• HBase is linearly scalable.
• It has automatic failure support.
• It provides consistent read and writes.
• It integrates with Hadoop, both as a source and a destination.
• It has easy java API for client.
• It provides data replication across clusters.
Where to Use HBase
• Apache HBase is used to have random, real-time read/write access to Big Data.
• It hosts very large tables on top of clusters of commodity hardware.
• Apache HBase is a non-relational database modeled after Google's Bigtable. Bigtable acts up on Google File System, likewise Apache HBase works on top of Hadoop and HDFS.
Applications of HBase
• It is used whenever there is a need to write heavy applications.
• HBase is used whenever we need to provide fast random accessto available data.
• Companies such as Facebook, Twitter, Yahoo, and Adobe useHBase internally.
4. Document: MongoDB

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.
MongoDB is a cross-platform, documentoriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.
Database
Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases.
Collection
Collection is a group of MongoDB documents. It is the equivalent of anRDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all

documents in a collection are of similar or related purpose.
Document
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.
The following table shows the relationship of RDBMS terminology with MongoDB.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)
Database Server and Client	
mysqld/Oracle	mongod
mysql/sqlplus	mongo

Advantages of MongoDB over RDBMS
• Schema less − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
• Structure of a single object is clear.
• No complex joins.
• Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
• Tuning.
• Ease of scale-out − MongoDB is easy to scale.
• Conversion/mapping of application objects to database objects not needed.
• Uses internal memory for storing the (windowed) working set, enabling faster access of data.
Why Use MongoDB?
• Document Oriented Storage − Data is stored in the form of JSON style documents.

• Index on any attribute

• Replication and high availability
• Auto-Sharding
• Rich queries
• Fast in-place updates
• Professional support by MongoDB
Where to Use MongoDB?
• Big Data
• Content Management and Delivery
• Mobile and Social Infrastructure
• User Data Management
• Data Hub
5.Graph: Neo4J
Neo4j is the world's leading open source Graph Database which is developed using Java technology. It is highly scalable and schema free (NoSQL).
What is a Graph Database?
A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. It is composed of two elements - nodes (vertices) and relationships (edges).
Graph database is a database used to model the data in the form of graph. In here, the nodes of a graph depict the entities while the relationships depict the association of these nodes.
Popular Graph Databases
Neo4j is a popular Graph Database. Other Graph Databases are Oracle NoSQL Database, OrientDB, HypherGraphDB, GraphBase, InfiniteGraph, and AllegroGraph.
Why Graph Databases?
Nowadays, most of the data exists in the form of the relationship between different objects and more often, the relationship between the data is more valuable than the data itself.

Relational databases store highly structured data which have several records storing the same type of data so they can be used to store structured data and, they do not store the relationships between the data.
Unlike other databases, graph databases store relationships and connections as first-class entities.
The data model for graph databases is simpler compared to other databases and, they can be used with OLTP systems. They provide features like transactional integrity and operational availability.
RDBMS Vs Graph Database
Following is the table which compares Relational databases and Graph databases.

Sr.No	RDBMS	Graph Database
1	Tables	Graphs
2	Rows	Nodes
3	Columns and Data	Properties and its values
4	Constraints	Relationships

5 Joins Traversal

Advantages of Neo4J
Following are the advantages of Neo4j.
• Flexible data model − Neo4j provides a flexible simple and yet powerful data model, which can be easily changed according to the applications and industries.
• Real-time insights − Neo4j provides results based on real-time data.
• High availability − Neo4j is highly available for large enterprise real-time applications with transactional guarantees.

• Connected and semi structures data − Using Neo4j, you can easily represent connected and semi-structured data.
• Easy retrieval − Using Neo4j, you can not only represent but also easily retrieve (traverse/navigate) connected data faster when compared to other databases.
• Cypher query language − Neo4j provides a declarative query language to represent the graph visually, using an ascii-art syntax. The commands of this language are in human readable format and very easy to learn.
• No joins − Using Neo4j, it does NOT require complex joins to retrieve connected/related data as it is very easy to retrieve its adjacent node or relationship details without joins or indexes.
Features of Neo4j
Following are the notable features of Neo4j −
• Data model (flexible schema) − Neo4j follows a data model named native property graph model. Here, the graph contains nodes (entities) and these nodes are connected with each other (depicted by relationships). Nodes and relationships store data in key-value pairs known as properties.
In Neo4j, there is no need to follow a fixed schema. You can add or remove properties as per requirement. It also provides schema constraints.
• ACID properties − Neo4j supports full ACID (Atomicity, Consistency, Isolation, and Durability) rules.
• Scalability and reliability − You can scale the database by increasing the number of reads/writes, and the volume without effecting the query processing speed and data integrity. Neo4j also provides support for replication for data safety and reliability.
• Cypher Query Language − Neo4j provides a powerful declarative query language known as Cypher. It uses ASCII-art for depicting graphs. Cypher is easy to learn and can be used to create and retrieve relations between data without using the complex queries like Joins.
• Built-in web application − Neo4j provides a built-in Neo4j Browser web application. Using this, you can create and query your graph data.
• Drivers − Neo4j can work with −

o REST APIto work with programming languages such asJava, Spring, Scala etc.
o Java Script to work with UI MVC frameworks such as Node JS.

o It supports two kinds of Java API: Cypher API and Native Java API
to develop Java applications. In addition to these, you can also
work with other databases such as MongoDB, Cassandra, etc.
• Indexing − Neo4j supports Indexes by using Apache Lucence.
Conclusion: From this assignment we understand the different types of
databases there uses, advantages and disadvantages. Where and how these
databases are used is known. And also understood the different types of
comparisons done on different criteria.