Mobile Application Development

LECTURE 16

What is NoSQL Database

- NoSQL stands for “Not Only SQL”.

- NoSQL database environment is a non relational, schema-less and largely distributed database.

- Developed in late 2000s to deal with limitations of SQL databases, especially scalability, multi-structured data.

NoSQL vs SQL Databases

- NoSQL is free from joins and relationships.

- No Entity Relationship Diagrams (ERD).

- RDBMS (Relational Database Management Systems) use expensive joins.

- Joins slower the speed of queries.

NoSQL vs SQL Databases

- NoSQL has a much lower maintenance cost compared to RDBMS.

- Structure and data types are fixed in advance in RDBMS. To store information about a new data

item, the entire database must be altered.

List of SQL and NoSQL Databases

SQL

- MySQL, MariaDB, Postgresql, Oracle Database, MS SQL, AmazonRDS

NoSQL

- MongoDB, Redis, CouchDB, Cloudant, Cassandra, Neo4j, ArangoDB, OrientDB, DynamoDB, FireStore, Firebase Real-time

Scaling Databases

- Scaling Databases mean increasing the size of database once the storage requirement increases.

- There are 2 ways.

1. Add additional Servers (Horizontal Scaling).

2. Increase the size of existing server (Vertical Scaling).

Scaling Databases

- **Vertical Scaling (SQL)**

We keep increasing RAM, Storage capacities of the single server. This becomes very expensive as it grows. 1 large server is more costly than 3,4 small servers.

Scaling vertically meaning a single server must be made increasingly powerful in order to deal with increased demand.

Scaling Databases

- **Horizontal Scaling (NOSQL)**

Horizontal scaling means to add more commodity servers or cloud instances.

The solution to support rapidly growing applications is to scale horizontally, by adding servers

instead of concentrating more capacity in a single server.

$$(1.00)^{365} = 1.00$$

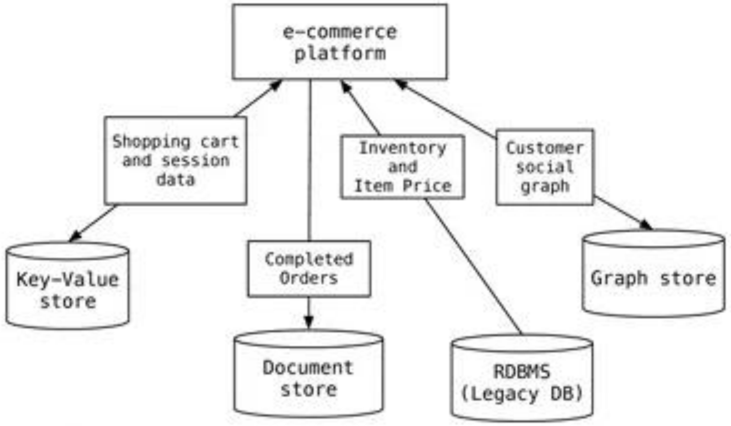
$$(1.01)^{365} = 37.7$$

Why NoSQL ?

- **Scaling is easier and less costly.**

- **PolyGlot Persistance**

Polyglot Persistence means that when storing data, it is best to use multiple data storage technologies, chosen based upon the way data is being used



Why NoSQL ?

- Ensuring global availability is difficult for relational databases where separate add-ons are required – which increases complexity.
- mobile games can reach tens of millions of users in a matter of months e.g. Pokemon GO, Clash of Clans etc
- With a distributed, scale-out database, mobile applications can start with a small deployment and expand as the user base grows, rather than deploying an expensive, large relational database server from the beginning.

Why NoSQL ?

- Analytics and Business Intelligence (Big Data). Cassandra etc

- Caching

A number of products provide a caching tier for SQL database systems. any NoSQL database technologies have excellent integrated caching capabilities. Caching means we store some data we frequently need in a small cache and retrieve from that instead of directly from Database. **Redis**.

Types of NoSQL Databases

- Document Store (**MongoDB**, **FireStore**, CouchDB etc).
- Key Value (Redis, Amazon DynamoDB, **Local Storage**, Riak, **Firebase Realtime**).
- Graph Database (Neo4j, Giraph)
- Column Oriented (Cassandra, Hbase)

Link to resources

- http://slides.com/alamgirqazi/nosql
- https://github.com/alamgirqazi/IntroToNoSQL (in detail)