Module 6 – Assignment 5

James Wells

jwells52@jh.edu

1. **Name two use cases for key-value databases**

A typical use case for a key-value database is for storage of user data and configurations for applications. This type of data is usually user preferences that are loaded when an application is accessed.

Another use case for key-value databases is for caching data from relational databases. If there is data in a relational database that is accessed frequently, a common practice for improved loading times is to cache the data a key-value database like Redis.

1. **Describe two reasons for choosing a key-value database for your application.**

A criterion for selecting a key-value database for an application is when there are frequent small reads and writes. Relational databases can lack in performance when there are many small changes happening at a high interval.  
  
The data models for an application can drive the decision for choosing a key-value database. If the application is handling scalar data such as integers and booleans then a key-value database can provide easy lookup queries for this data by searching on a key.

1. **Name two use cases for document databases.**

A use case for document databases is providing a data storage for when data has a varying number of attributes. For example, an application that sells products.  
  
Another use case for document databases is when the application is handling JSON structured data. Comparatively, document databases can embed JSONs which allow them to be stored/retrieved easily.

1. **Describe two reasons for choosing a document database for your application.**

If your application is dealing with unstructured data like text documents. Then a document database is the most practical solution for storing that data due to its ease-of-use and flexibility with different data types.

When the application is handling large amounts of data that have varying attributes, then a document database is a good option.

1. **Name two use cases for column family databases.**

Column family databases are typically used when an application is geographically distributed across multiple datacenters.

When there is an application dealing with Big Data (i.e. Terabytes or even Petabytes) then column family databases are ideal for use.

1. **Describe two reasons for choosing a column family database for your application.**

When there is a need for high availability of an application (i.e no downtime ever) then the database needs to be highly available as well. Column family databases can be distributed across many servers, and this will allow for high availability of data for the application.

Another reason for choosing a column family database is related to one of the use cases. If the application is dealing with a substantially high amount of data, then column family databases are a good choice.

1. **Name two use cases for graph databases.**

A use case of an application using graph databases are social networks. Profiles can be modeled as entities in a graph database. Traversing the graph database for social networks can help with identifying recommendations on products to buy, people to follow, etc.  
  
IT Infrastructure management is another use case for graph databases. Being able to model IT infrastructure as a graph can help remediate issues with failure modes such as outages and bottlenecks.

1. **Describe two reasons for choosing a graph database in your application.**

Problem domains where the problem is commonly represented as a network of connected entities are suited for graph databases.

When there are direct relationships between entities that need to be modeled, using a graph database to store the entities and model the connections is a standard approach.

1. **Name two types of applications well suited for relational databases.**

E-commerce applications are well suited for relational databases. Customer information and transaction history can be stored in a relational database. These types of databases can make it trivial for e-commerce applications to pull customer information if requested.

Banking applications which require high integrity data can be well-suited for relational databases.

1. **Discuss the need for both NoSQL and relational databases in enterprise data.**

Enterprises have a high demographic of data types, so it is a necessity to use both NoSQL and relational databases. With enterprises, there exists “old” data that follow data models for relational databases. So, enterprises still need relational databases in order to store and use this data. However, now in the age of information, there is a massive influx of unstructured data. For enterprises to create applications that can bring value out of unstructured data, they need NoSQL databases.