Describe when you should use a relational database and why. Describe when you should use a non-relational database and why. In which scenarios should you use one vs the other? Give at least two examples of each.

Non-relational database does not store data in tabular form. Flexible schemas and scale easily with large amounts of data and high user loads. Data partitioned across multiple machines to scale. Relational databases assume data stays on one machine. CAP theorem, less mismatch between objects in programming language and a table in a relational database sense. Mechanisms for storage and retrieval of data which are modeled different than tabular. According to CAP theorem, with partitioning, there is choice between consistency and availability; MongoDB only recently gained ACID transaction support. A car object can be stored in one document in MongoDB, but in relational database, the information for the car object may be spread throughout multiple tables.

Use non-relational databases when working with data that changes frequently, cloud computing, and promoting developer productivity. Document model is well suited to polymorphic data that can change frequently. Social media and real-time analytics

Use relational database when data integrity and consistency is crucial. Structured, tabular data.

Relational databases are best utilized when the data is structured and tabular because they are suitable for storing data organized in fixed schema that includes relationships between entities. Relational databases are ACID compliant, which ensures data consistency and integrity. The fixed schema, in which primary and foreign keys can be defined to enforce integrity constraints and maintain relationships between tables, makes relational databases ideal for users most concerned with data consistency and accuracy. Also, relational databases better handle complex queries, like JOINs, and allow the DBA to create Views specific for user needs.

Non-relational databases have flexible schemas that allow for large amounts of data and high user loads, can be easily updated, and scale horizontally. Non-relational databases are ideal for scenarios in which the user requires access to high volumes of data quickly. Additionally, they can handle a diverse set of datatypes.

Two examples where a relational database would be better than a NoSQL database include a banking system and a customer sales database. In the banking system, ACID transactions would play a crucial role in ensuring that data such as account balances and loan payment records are consistent and accurate, which would both be highly prioritized database qualities in a system that handles such sensitive information. In customer sales database, business analysts would want to perform complex queries to extract information that can inform marketing and other strategies, which would not be as easy in a NoSQL database.

Two examples where a non-relational database would be better than a relational database include social media, and fraud detection. Social media apps and websites contain tons of information that is accessed and updated by millions of users at a time. It also must keep track of a lot of complex relationships between accounts and posts. For these reasons, a NoSQL database would be more efficient to manage the data. In fraud detection, data must be analyzed in real time to quickly identify fraud and alert the victim, thus the models that identify fraud need access to data very quickly. Fast access to data is a strength of non-relational databases over relational databases.