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.

Relational databases and non-relational databases, also known as NoSQL databases, have different strengths and are suitable for different scenarios.

When to use a Relational Database:

1. Structured Data: Relational databases are best suited for structured data that can be organized into tables with predefined relationships. They are ideal for scenarios where data consistency and integrity are crucial (Relational vs non-relational database: Which one should you use?).
2. Examples:
    a. E-commerce: A relational database can be used to store customer information, order details, and product data. The relationships between customers, orders, and products can be easily managed using primary and foreign keys, allowing for efficient querying and reporting (What Is A Relational Database (RDBMS)? | Google Cloud).
    b. Financial Systems: Relational databases are commonly used in financial systems to store transactional data, customer accounts, and financial reports. The ability to enforce data integrity and perform complex queries makes them well-suited for handling financial data (What is a relational database? | IBM).

When to use a Non-Relational Database:

1. Unstructured or Semi-Structured Data: Non-relational databases excel at handling unstructured or semi-structured data, such as JSON documents or key-value pairs. They provide flexibility and scalability for scenarios where data models may evolve over time (Relational vs non-relational database: Which one should you use?).
2. Examples:
    a. Social Media Analytics: Non-relational databases can be used to store and analyze social media data, which often comes in unstructured formats like tweets or posts. The flexible schema of NoSQL databases allows for easy storage and retrieval of this type of data (What Is A Relational Database (RDBMS)? | Google Cloud).
    b. Internet of Things (IoT): IoT applications generate large volumes of data from various sources. Non-relational databases can handle the high velocity and variety of IoT data, making them suitable for storing sensor data, device logs, and real-time analytics (https://www.ibm.com/topics/relational-databases).

In summary, relational databases are ideal for structured data with predefined relationships, while non-relational databases are better suited for unstructured or semi-structured data. The choice between the two depends on the specific requirements of your application and the nature of your data.