BASIC TASKS

a) Read through the list of “Real World Database Examples” and choose several that you want to research.

1.Social Media – Entertainment.

2.Banking – Finance.

b) For each scenario that you have chosen the aim is to produce a recommended database solution. This involves deciding what type of database (Relational, Network, Hierarchical, Object-Oriented, NoSQL[Key Value, Column-Based, Graph, Document]) is most suitable for each scenario as well as appropriate database software tools that could be used to implement .

Social media：

Because it has to deal with unstructured data such as text, images, videos, etc., and because it needs to process a large number of users and data, scalability is also very important. So use a flexible and scalable NoSQL database. To support high-throughput and low-latency access to social media, a distributed storage and computing framework is employed. And to protect user data from unauthorized access and leakage, encryption, controlled access, and anonymization of the clerk are commonly used. The use of open-source databases and distributed systems can reduce software licensing costs, while optimizing the use of hardware resources and the energy efficiency of data centers can reduce operational costs. Due to the sheer volume of read and write operations on social media data, full strong consistency can impact performance and availability. As a result, many social media platforms employ eventual consistency models to guarantee that data is consistent within a certain time window.

Performance is another key metric for social media platforms that directly impacts the user experience. The use of caching, load balancing, and distributed query processing technologies can improve the responsiveness and throughput of the system.

Banking:

Oracle, Microsoft SQL Server, IBM DB2, etc. These database systems are widely used in banks to store and manage important business data such as customer information, account data, transaction records, etc.

But Oracle is the preferred choice for many banks for its powerful data processing capabilities and high levels of security, which effectively preventing data breaches and unauthorized access.

Oracle database also supports multiple data types and complex query operations, enabling banks to manage and analyze data more efficiently. For example, by using PL/SQL (Oracle's procedural language extension), banks can write complex stored procedures and triggers to automate operations and data validation, thereby improving operational efficiency. In addition, Oracle also provides comprehensive optimization tools such as automatic tuning and performance monitoring to help banks continuously optimize system performance.

MEDIUM TASKS

Work in a small group to produce a comprehensive comparison of Relational Vs NoSQL Databases. Do research and identify definitions, benefits and limitations of Relational and NoSQL database systems.

Comparison Table: Relational vs NoSQL Databases

Feature	Relational Databases	NoSQL Databases
Definitions	Structured data stored in tables with predefined schemas.	Non-relational, schema-less, designed for unstructured data.
Benefits	- Strong ACID compliance - Data integrity - Mature, well-established technology	- High scalability - Flexibility - Handles large volumes of data efficiently - Good for distributed systems
Limitations	- Scalability issues - Not suitable for unstructured data - Complex queries can be inefficient	- Lack of standardization - No ACID compliance - Less mature technology - Limited query capabilities
Examples of Software	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis
Use Cases	- Traditional applications - Financial systems - Applications requiring complex queries	- Real-time analytics - Content management - IoT applications

Feature	Relational	NoSQL
Database structures	structured data stored in rows and columns of a table	unstructured data stored in datasets such as documents, key value pairs, column storage, and graph structures
Data storage	smaller volume	larger volume
Do they support ACID transactions	yes	normally no, or partly support
Is Normalisation supported	yes	no
Integrity constraints; Data Accuracy	more stringent and comprehensive	looser
Scalability	focusing on improving processing power by increasing hardware resources or the number of servers	having more advantages in flexibility and scalability
Simplicity	If structured data storage and complex queries are required, relational databases may be simpler and easier to use	If dealing with large-scale data and high concurrency access is required, non-relational databases may be simpler for usage
Complexity Cost	In order to maintain data consistency and integrity, more management and optimization work is required, which may increase the complexity cost	The lack of a unified structure may lead to complexity in data management. Besides it usually doesn't support transaction processing, which is a challenge for applications that require transaction rollback
Reliability	performing better in data consistency and integrity	having more advantages in scalability and performance
Schema Flexibility	Suitable for scenarios that require highly accurate and consistent data processing	Adopting flexible document or key value pairs, capable of storing and processing various types of data, including

		unstructured data
Performance	Relational database typically emphasizes data consistency in terms of read and write performance, which may lead to performance degradation when faced with massive amounts of data and high concurrency read and write.	NoSQL is typically optimized for specific data models and access patterns to achieve higher performance, especially when dealing with large-scale data.