Conger Yang, Yue Ma & Yungheng Li

Basic Task:

a) 1)Online and Social Gaming-Entertainment
   2)Weather.
   3)Library Management System – Document Management.

b) **(1)Database type:** Relational,Hierarchical, Object-Oriented, NoSQL[KeyValue,Column-Based,Document,Graph]

In the online and social gaming space, we use relational databases, which are used to track your scores, your inventory, and your game state. Next, we will use the web database to enable players to interact with other players. The hierarchical database is mainly used by players to view their equipment library or click on skill points. The key value in NoSQL is applied to the player's game settings and game preference recommendation; Column-Based for game logs; Documents are used for in-game profiles and character attributes; Graph is used to handle complex data on social networks (in terms of scalability ,data consistency and performance)

Database Software: MySQL,PostgreSQL,MongoDB,Redis,SQLite

(2)Database type:

Relational,NoSQL[Key Value,Column-Based,Graph]

In the weather field, we will use a relational database, which stores weather-related data through tables, and queries and manages the data through SQL. We also use NoSQL Key Values to store simple key pair data; Column-based to store and analyze large amounts of structured data, such as time series of meteorological observation data; Graphs are used to represent complex relationships between entities, such as spatial relationships between meteorological sites or interactions between weather systems. (From the perspective of data consistency)

Database Software: MySQL,Oracle,SQL Server,Mongo DB,Cassandra,Influx DB,

(3)Database type： Relational,NoSQL[Document]

Like the application of the weather system, the library management system uses a relational database to store the data of books and students through tables, and uses SQL to query and manage the data. In addition, library management systems also use documents in NoSQL to store semi-structured document data, such as book details or users' personal files (for data consistency reasons)

Database Software: MySQL,Oracle,SQL Server,Mongo DB,PostgreSQL

MEDIUM TASKS：

a)

Feature	Relational Databases	NoSQL Databases
Definition	Relational databases are based on the relational model, where data is stored in tables that are related to each other through relationships (such as foreign keys).	NoSQL databases are designed to handle large volumes of data with a flexible schema, supporting various data formats such as key-value pairs, documents, wide-column stores, and graph databases.
Advantages	Strong consistency Transaction support Complex query capabilities Strict data structure	High scalability High performance Flexible data models Suitable for big data applications
Limitations	Limited scalability High maintenance costs Not suitable for unstructured data	Lack of a unified query language Limited transaction support Data consistency issues
Software Examples	MySQL PostgreSQL Oracle SQL Server	MongoDB Cassandra Redis Couchbase
Use Cases	Financial transactions Customer relationship management Inventory management	Social media Real-time analytics Big data storage

b)

Feature	Relational Databases (SQL)	NoSQL Databases
Data Storage – Volumes of Data	Moderate volumes, structured data	Large volumes, unstructured/semi-structured data
Normalization Supported?	Yes, normalization is common	Typically denormalized for performance
Simplicity (Ease of Use)	More complex, but strong support available	Simpler, with flexible structure and developer-friendly
Complexity & Cost	High due to structure and maintenance	Lower, flexible structure
Database Structure	Table-based (rows and columns)	Key-value, document, column-family, graph-based
Integrity Constraints	Strong integrity enforcement (foreign keys, etc.)	Flexible, weaker consistency (eventual or BASE consistency)
ACID Transactions	Full ACID support	Some support ACID, but others follow BASE
Performance (Read/Write)	Slower on writes, optimized for complex queries	Faster writes, optimized for simple operations
Scalability	Vertical scaling (more powerful hardware)	Horizontal scaling (adding more nodes)
Schema Flexibility	Rigid, predefined schema	Flexible schema or schema-less
Reliability	High, due to strong consistency and ACID transactions	Varies, eventual consistency may impact reliability
Storage Requirements	Higher due to normalization	Lower due to denormalized structure

Relational Database System (RDBMS)

Benifit:

High data consistency: Ensure data consistency and integrity through transaction and locking mechanisms
Restrictions:

Features:

Data structuring: Data is stored in tables, each containing multiple columns and rows.

Limitations:

When the amount of data is very large, the scalability of relational databases may be limited, requiring complex sharding or clustering techniques.

Example: MySQL

MySQL supports large database applications and provides high performance, reliability, and scalability.

Non-relational Database System (NoSQL)

Benifit:

High flexibility: Supports a variety of data models, such as key-value pairs, documents, column families and graphs, suitable for storing semi-structured and unstructured data.

Limitaion：

Data consistency is difficult to guarantee: In non-relational databases, data consistency often requires application maintenance.

Features:

No fixed table structure: Fields can be dynamically added or removed as needed.

Example: MongoDB

MongoDB is a document-based NoSQL database that supports dynamic queries and indexes and is suitable for storing complex data structures. It is widely used in areas such as big data, content management and real-time analytics.