DBAS6211 ASSIGNMENT 1

ST10368647

Shiara Phekoo

QUESTION 1 (Attached as separate pdf)

Sources Used:
https://www.canva.com/ (Midori Nediger, 2024)

QUESTION 2

2.1. To store the data of the courses such as the course name, the name of the creator of the course and the category that the course belongs to, a relational database will be the most suitable due to the following reasons:

Structured Data: Relational databases are good when handling well-structured data with defined relationships.The data regarding the courses can be divided among multiple tables and connected through a relationship.(Ben Lutkevich, 2021) For example, this can be done by having one ‘course' table with fields such as course ID, title, description and having separate tables for creators and categories.

Relationships: Relational databases are able to support relationships between entities by using foreign keys. Connections can be easily established between courses, creators and categories. For example, each course could have a foreign key that references the ID of its creator and the ID of which category it belongs to. This ensures that only valid creators and categories are associated with every course. (Ben Lutkevich, 2021)

Complex Queries: Relational databases are appropriate for complex queries such as joins, aggregations, and filtering among multiple tables.(Ben Lutkevich, 2021) In this scenario, queries like locating all courses in a specific category, finding courses created by a particular instructor, or displaying courses sorted by popularity within a category may need to be carried out. Relational databases are able to handle these types of queries efficiently. (Ben Lutkevich, 2021)

Data Consistency and Integrity: Relational databases ensure data consistency and integrity by the use of constraints such as unique constraints, foreign key constraints and data validation rules.(Ben Lutkevich, 2021) In this case we can use rules to prevent duplicates of course titles.

Adaptability to Changes: Relational databases generally have a fixed outline but are still flexible enough to adjust to changes in data requirements.(Ben Lutkevich, 2021) If the structure of the data develops over time such as an increase of users in the course marketplace app, the relational database will be able to support these changes without affecting the existing data.

2.2. To store the data of videos, pictures and files that course creators want to make available to those purchasing their courses, a NoSQL database will be the most suitable due to the following reasons:

Schema Flexibility: NoSQL databases offer schema flexibility,(Hossein Ashtari, 2022), which will be useful in this case since the structure and information of multimedia items such as images, text and videos can change.

Binary Data Support: NoSQL databases offer great support for the storage of binary data such as images and videos within the database documents.(Hossein Ashtari, 2022) This simplifies managing the multimedia by allowing them to be stored with other course related data in a single document.

Scalability: NoSQL databases are designed for horizontal scalability, which makes them very useful at handling large amounts of multimedia and being able to adjust to increases in user activity or data storage requirements.(Hossein Ashtari, 2022) This scalability is useful for this course marketplace app since many users may be uploading or accessing multimedia files regularly.

Performance: NoSQL databases offer efficient performance when regarding multimedia content where data needs to be distributed across many servers or data centers. This helps to ensure smooth playback and fast loading times for the multimedia,(Hossein Ashtari, 2022), which ultimately improves the users experience while using the course marketplace app.

Ease of Integration with Application Frameworks: NoSQL databases have tools and libraries that specialize in integrating with well-known app frameworks and Content Delivery Networks (CDNs). (Hossein Ashtari, 2022) This allows seamless integration of multimedia content in the app and simplifies tasks like video streaming or loading images and will therefore apply appropriately with the course marketplace app.

QUESTION 3

Sources Used:
https://app.diagrams.net/ (Gliffy, 2020)

QUESTION 4

After analyzing the rules and the given diagram, here are the following recommendations and changes I would advise to be made:

1. Firstly, I would recommend making use of a digital software to create the UML ERD diagram. This ensures that the structure of the tables are consistent and the fields and relationships can be clearly located and seen, giving it an overall professional and precise impression. (Gliffy, 2020) By using digital software, changes can also be easily reviewed and carried out with ease.
2. In the Item table, I would change the field name ‘Name’ to ‘Item Name’. This simple change will prevent confusion if other tables have a ‘Name’ field as well.
3. In the Component table, I would add a primary key field named ‘Component ID’ and change the ‘Name’ field to ‘Component Name’. A ‘Component ID’ field would make an appropriate primary key for the table and changing the field ‘Name‘ to ‘Component Name’ gives us more clarity and makes the table easier to understand. (Gliffy, 2020)
4. I would change the Colour table to a Production Line table with ‘Line ID’ as the primary key and ‘Line Color’ as a field. Making these changes will make more sense than just having a table named ‘Colour’ with no fields. And this will allow the Production Lines to be distinguished by their color field.
5. For the Supplier table, I would add ‘Supplier ID’ as the primary key and ‘Supplier Name’ and ‘Supplier Address’ as the other fields. This applies more appropriately to the scenario rules as it stores the supplier's details in the allocated fields. (Gliffy, 2020)
6. I would recommend adding an Item Component table with the foreign keys ‘Item ID’ and ‘Component ID’ alongside an additional field ‘Quantity’. This table will keep track of the quantity of each component.
7. I would also recommend adding a Supplier Component table with the foreign keys ‘Supplier ID’ and ‘Component ID’. This way the Supplier Component table acts as a bridge between the Supplier and Component tables. This allows us to track which component is being sent by which supplier since a supplier can supply many different components.
8. I would add the types of relationships to each table to show how they relate in a relational database. This makes the diagram easier to read and understand. (Gliffy, 2020)
9. Add verbs to the relationships so that we can further understand how the tables and fields interact with each other. (Gliffy, 2020)
10. Show a field is a primary or foreign key in a separate column to the left of its name to increase the overall look and professionality of the diagram. (Gliffy, 2020)

Sources Used:

1. Midori Nediger (Jan 11, 2024) How to Make an Infographic in 5 Easy Steps — Fast (2024 Guide) [Online] Available at: https://venngage.com/blog/how-to-make-an-infographic-in-5-steps/ Accessed at: 20/03/2024
2. Ben Lutkevich (June 2021) Relational Database [Online] Available at:https://www.techtarget.com/searchdatamanagement/definition/relational-database Accessed at: 20/03/2024
3. Hossein Ashtari (October 18, 2022) What Is NoSQL? Features, Types, and Examples [Online] Available at: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-nosql/ Accessed at: 20/03/2024
4. Gliffy (October 2, 2020) How to Draw Entity Relationship Diagrams (ERDs) [Online] Available at: https://www.gliffy.com/blog/how-to-draw-an-entity-relationship-diagram#top Accessed at: 20/03/2024