Advanced Databases 2023-24 Exam Paper

# Answers



(a)

1. Denormalization is when you add extra data to your database tables to make them faster and easier to read. You usually do this after you have normalized your database, which means you have organized your data into separate tables with no duplicates.
2. **1.** When you have a lot of read operations and few write operations. For example, if you have a database that stores historical data, such as weather records or stock prices, you might want to denormalize it to make it easier to query and analyse. You don’t have to worry too much about updating or inserting new data, because the data is mostly static and historical.

**2.** When you have a lot of complex queries that involve multiple joins. For example, if you have a database that stores information about online orders, such as customers, products, and transactions, you might want to denormalize it to reduce the number of joins required to retrieve data. You can add some redundant data, such as customer name or product price, to the transactions table, so that you can get all the information you need from one table. This can improve the performance and simplicity of your queries.

1. **Improved query performance:** Denormalization can improve query performance by reducing the number of joins required to retrieve data. Joins are expensive operations that can slow down your queries, especially if you have to join many tables or large tables. By adding redundant data to one or more tables, you can avoid some joins and make your queries faster.

**Reduced complexity:** Denormalization can reduce complexity by combining related data into fewer tables. This can make your database schema simpler and easier to manage. You can also write simpler queries that involve fewer tables and less logic.

**Easier maintenance and updates:** Denormalization can make it easier to maintain and update your database by reducing the number of tables. You can also use triggers, stored procedures, or application logic to ensure that the redundant data is updated consistently and correctly.

**Improved read performance:** Denormalization can improve read performance by making data more accessible to the queries. You can also use indexes, partitioning, or caching to optimize the data retrieval and reduce the disk I/O.

(b)

1. **1.** SQL databases store data in tables with fixed columns and rows, and each row has a unique identifier. NoSQL databases store data in different ways, such as documents, key-value pairs, graphs, or wide-columns, and they do not have a fixed schema for the data.

**2.** SQL databases can only handle more load by adding more resources to a single server. NoSQL databases can handle more load by adding more servers to a cluster. This makes NoSQL databases cheaper and easier to scale horizontally.

**3.** SQL databases have a rigid schema that must be defined before storing any data. This means that any changes to the data structure or the relationships between tables require modifying the schema and updating the database accordingly. NoSQL databases have a flexible schema that allows adding new attributes and fields without changing the existing ones. This means that any changes to the data structure or the relationships between documents are easier to implement.

1. A. This scenario requires a high level of consistency, accuracy, security, and reliability of the data. It also involves complex queries that need to access multiple tables with different schemas. Therefore, I would choose an SQL solution for this scenario.

B. This scenario requires a high level of availability, scalability, speed, and flexibility of the data. It also involves real-time communication that needs to preserve order and visibility of messages among participants. Therefore, I would choose a NoSQL solution for this scenario.

C. This scenario requires a high level of diversity, adaptability, exploration, and analysis of the data. It also involves unstructured or semi-structured data from various sources such as sensors, maps, social media, etc. Therefore, I would choose a NoSQL solution for this scenario.

Q2

* + - * 1. A database is a system that organizes and stores data in a structured way, so that it can be accessed and manipulated by different applications. A database usually has a central location where all the data is stored, and a set of rules and commands that define how the data is organized and retrieved. A database can store different types of data, such as numbers, text, images, or audio files. Some examples of databases are relational databases (such as MySQL or Oracle), which use tables and rows to store data, and NoSQL databases (such as MongoDB or Cassandra), which use different formats to store data.

A data warehouse is a system that collects and integrates data from various sources into a single repository for analysis and reporting purposes. A data warehouse usually has a three-tier architecture, which consists of a bottom tier that stores the raw data from different sources, a middle tier that performs fast query operations on the data using online analytical processing (OLAP) tools, and a top tier that provides user interfaces for accessing and visualizing the data. A data warehouse can store both structured and unstructured data, such as sales records, customer feedback, or social media posts. Some examples of data warehouses are Amazon Redshift or Google BigQuery, which are cloud-based services that offer scalability and performance for big data analytics.

A data lake is a system that stores large volumes of raw or semi-structured data in its original form, without imposing any predefined schema or structure on the data. A data lake usually has an open architecture that allows users to access and process the data using various tools and frameworks, such as Apache Hadoop or Apache Spark. A data lake can store any type of unstructured or semi-structured data, such as images, audio files, tweets, or web pages. Some examples of data lakes are Azure Data Lake Storage or AWS S3 Glacier Deep Archive Storage Service , which are cloud-based services that offer low-cost storage for long-term retention of big data

| **System** | **Type of data stored** | **Preparing data for use** | **Data Freshness** | **Typical users** |
| --- | --- | --- | --- | --- |
| **Database** | Structured data in tables and rows | Data is organized and validated before storing | Data is updated frequently and reflects the current state | Application developers, business analysts, end users |
| **Data Warehouse** | Structured and unstructured data from various sources | Data is transformed and integrated before storing | Data is updated periodically and reflects the historical state | Data analysts, data scientists, business intelligence professionals |
| **Data Lake** | Unstructured and semi-structured data in original form | Data is processed and analysed after storing | Data is updated continuously and reflects the real-time state | Data engineers, data scientists, machine learning engineers |
|  |  |  |  |  |

(c)

a. An insurance company wants to analyse data patterns across different geographical regions, analyse customer trends, and to track market movements quickly.

A data warehouse is a good choice for this purpose. It can store and query large amounts of structured and semi-structured data from various sources and support business intelligence and data discovery/storage1.

b. A telecommunications company wants to store details of customers, their call records, their data usage and generate monthly bills and an annual report.

A database is a good choice for this purpose. It can store and manage customer information in a consistent and secure way using online transaction processing (OLTP)1.

c. Environmental scientists working for a city transport authority want to use a range of different sources to explore alternate traffic planning solutions, predicting expected reductions in car traffic, potential carbon emission reductions and potential health improvements.

A data lake is a good choice for this purpose. It can access and analyse diverse sources of data from various domains using advanced analytics techniques such as machine learning and artificial intelligence2.

d. Departmental managers in a retail company want to identify buying patterns of individual customers and different types of customers, analyse the impact of special sales promotions and determining future pricing policy for different products.

Both a database and a data warehouse can be used for this purpose, depending on the volume, variety, velocity, veracity (V-V-V-V) of the data32. For example:

If they need large amounts of historical sales transactions with detailed information about individual customers or different types of customers, they may use a database.

If they need aggregated or summarized sales transactions with high-level information about customer segments or product categories, they may use a warehouse.

If they need both types of information but with different levels of granularity or frequency (such as daily vs weekly vs monthly), they may use both a database and a warehouse, but with different schemas or models (such as relational vs dimensional).

e. A large oil and gas production company was an early adopter of multiple disruptive technologies, in particular cloud computing and Internet of Things. The company wishes to use large volumes of historical data to optimise drilling, lower operating costs, stay compliant with regulatory requirements and investigate the impact more stringent economic and regulatory requirements would have on their operating costs.

A cloud-based platform that integrates both a database and a warehouse is the best choice for this purpose. It can access and process large volumes of historical data from various sources using cloud computing and Internet of Things (IoT). It can also offer scalability, flexibility, cost-effectiveness, security, etc., for big data applications.

Q3