Chapter 01

Practice Exercises

1.1 This chapter has described several major advantages of a database system. What are two disadvantages?

Answer :

- **Complexity and Cost** – Database systems require expertise to design and maintain, and they can be expensive.
- **Performance Overhead** – Additional processing for security, concurrency, and transactions can slow down performance.

1.2 List five ways in which the type declaration system of a language such as Java or C++ differs from the data definition language used in a database.

Answer :

- **Purpose** – Java/C++ type declarations define variables and objects in memory, while a database DDL (Data Definition Language) defines structures for persistent data storage.
- **Persistence** – Java/C++ types exist during program execution, whereas database schema definitions persist beyond program execution.
- **Relationships** – DDL supports constraints and relationships (e.g., foreign keys), while Java/C++ types do not inherently manage relationships between objects.
- **Normalization** – Databases use normalization to minimize redundancy, while Java/C++ structures often allow duplication of data for flexibility.
- **Querying** – DDL supports declarative querying via SQL, while Java/C++ requires imperative programming to manipulate data.

1.3 List six major steps that you would take in setting up a database for a particularenterprise.

Answer :

1. **Requirement Analysis** – Identify business needs, data requirements, and system constraints.
2. **Database Design** – Create an Entity-Relationship (ER) model and normalize the schema.
3. **DBMS Selection** – Choose an appropriate database management system (e.g., MySQL, PostgreSQL, MongoDB).

4. **Implementation** – Define tables, relationships, constraints, and indexes using DDL.
5. **Data Insertion & Testing** – Populate the database with sample data and validate its integrity and performance.
6. **Security & Maintenance** – Implement user roles, backups, and indexing optimizations for long-term reliability.

1.4 Suppose you want to build a video site similar to YouTube. Consider each of the points listed in Section 1.2 as disadvantages of keeping data in a file-processing system. Discuss the relevance of each of these points to the storage of actual video data, and to metadata about the video, such as title, the user who uploaded it, tags, and which users viewed it.

Answer :

A file-processing system has several disadvantages compared to a database system when managing a video site like YouTube:

1. **Data Redundancy & Inconsistency** – Storing video metadata separately for each file can lead to duplication and inconsistencies in user-uploaded data.
2. **Lack of Data Integrity & Security** – Ensuring data integrity (e.g., preventing duplicate video entries) is harder in a file-based system, and access control is more difficult to enforce.
3. **Difficulty in Searching & Retrieval** – Searching for videos by tags, title, or uploader is inefficient in a file system compared to a database with indexed queries.
4. **Concurrency Issues** – Multiple users trying to update metadata (e.g., adding comments or likes) can lead to conflicts without proper transaction management.
5. **Backup & Recovery Challenges** – A file-based approach lacks built-in backup and recovery mechanisms, making it harder to restore lost data.
6. **Scalability Issues** – As the platform grows, managing millions of videos and metadata without a structured database becomes increasingly inefficient.

1.5 Keyword queries used in web search are quite different from database queries. List key differences between the two, in terms of the way the queries are specified and in terms of what is the result of a query.

Answer :

Keyword queries in web search and database queries differ in several ways:

1. **Query Format** – Web search queries are simple keywords or phrases (e.g., *"best smartphones 2024"*), while database queries use structured SQL statements (e.g., `SELECT * FROM products WHERE category = 'smartphone'`).

2. **Precision** – Web search results are approximate and ranked by relevance, whereas database queries return exact matches based on specified conditions.
3. **Data Source** – Web search queries retrieve information from unstructured or semi-structured web pages, while database queries work with structured, relational data.
4. **Ranking** – Search engines use algorithms like PageRank to rank results, whereas database queries return data as stored without ranking.
5. **Flexibility** – Web search allows vague or incomplete queries, while database queries require a well-defined structure, including table and column names.
6. **Response Format** – Web searches return a list of relevant web pages or documents, while database queries return structured rows and columns of data.

Exercises

1.6 List four applications you have used that most likely employed a database system to store persistent data.

Answer :

Four Applications That Likely Use a Database System

1. **Social Media Apps (e.g., Facebook, Twitter)** – Store user profiles, posts, comments, and likes.
2. **E-commerce Platforms (e.g., Amazon, eBay)** – Manage product catalogs, orders, and customer details.
3. **Banking Systems (e.g., PayPal, Online Banking Apps)** – Store transaction records, account balances, and user credentials.
4. **Streaming Services (e.g., Netflix, Spotify)** – Maintain media libraries, user preferences, and watch history.

1.7 List four significant differences between a file-processing system and a DBMS.

Answer :

Four Significant Differences Between a File-Processing System and a DBMS

1. **Data Redundancy** – A file-processing system may store duplicate data in multiple files, while a DBMS minimizes redundancy through normalization.
2. **Data Consistency** – DBMS enforces integrity constraints to maintain consistency, whereas a file system lacks centralized control.
3. **Concurrency Control** – DBMS supports multiple users accessing and modifying data simultaneously, while file systems struggle with concurrent access management.
4. **Querying & Retrieval** – DBMS provides powerful query languages like SQL, whereas file systems require manual programming for data retrieval.

1.8 Explain the concept of physical data independence and its importance in database systems.

Answer :

Physical Data Independence and Its Importance

Physical data independence refers to the ability of a database system to change the physical storage structure without affecting how data is accessed at the logical level.

Importance:

- Allows optimization of storage and indexing without modifying application queries.
- Enhances system performance and scalability.
- Reduces maintenance costs and effort when upgrading hardware or storage techniques.

1.9 List five responsibilities of a database-management system. For each responsi- bility, explain the problems that would arise if the responsibility were not dis- charged.

Answer :

Five Responsibilities of a DBMS and Problems If Not Fulfilled

1. **Data Storage and Retrieval** – Ensures efficient storage and access to data.
    - *Problem:* Without it, data retrieval would be slow and unstructured, requiring manual file handling.
2. **Concurrency Control** – Manages simultaneous access by multiple users.
    - *Problem:* Without concurrency control, users may overwrite each other's changes, leading to data inconsistency.
3. **Backup and Recovery** – Protects data from failures and restores it when needed.
    - *Problem:* Data loss due to system crashes or corruption would be irreversible without backup mechanisms.
4. **Security and Authorization** – Restricts unauthorized access to sensitive data.
    - *Problem:* Without security controls, any user could modify or delete critical data, leading to breaches.
5. **Integrity Constraints Enforcement** – Ensures data validity and consistency.
    - *Problem:* Without constraints, duplicate, inconsistent, or incorrect data could be stored, reducing reliability

1.10 List at least two reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a library of C or C++ functions to carry out data manipulation.

Answer :

Two Reasons for Using SQL Instead of Only C/C++ Functions

1. **Simplicity and Productivity** – SQL allows users to specify *what* data they need without writing complex procedural code, making data manipulation more efficient.
2. **Optimization and Performance** – SQL queries are automatically optimized by the DBMS, whereas C/C++ functions require manual optimization, which can be complex and inefficient.

1.11 Assume that two students are trying to register for a course in which there is only one open seat. What component of a database system prevents both students from being given that last seat?

Answer :

Component Preventing Both Students from Registering for the Last Seat

The **Concurrency Control and Transaction Management** component ensures that only one student can successfully register. The system uses **ACID (Atomicity, Consistency, Isolation, Durability)** properties to prevent race conditions, often implementing a **locking mechanism** or **atomic transactions** to ensure that once one student registers, the seat count updates before another transaction can proceed.

1.12 Explain the difference between two-tier and three-tier application architectures. Which is better suited for web applications? Why?

Answer :

Difference Between Two-Tier and Three-Tier Architectures & Best for Web Apps

- **Two-Tier Architecture** – The client communicates directly with the database, typically using an application layer that interacts with the DBMS (e.g., a desktop application connecting to a database).
- **Three-Tier Architecture** – Includes a middle-tier (application server) between the client and the database, handling business logic, security, and request processing before interacting with the DBMS.

Best for Web Applications:
Three-tier architecture is better suited for web applications because it enhances **scalability, security, and performance** by offloading processing from clients and databases while supporting load balancing and distributed systems.

1.13 List two features developed in the 2000s and that help database systems handle data-analytics workloads.

Answer :

Two Features Developed in the 2000s for Data Analytics in Databases

1. **Columnar Storage** – Optimized storage for analytical queries by storing data in columns instead of rows, improving query performance in data warehouses.
2. **Distributed Computing Frameworks (e.g., Hadoop, Spark)** – Enabled large-scale data processing and analytics by leveraging parallel computing across multiple machines.

1.14 Explain why NoSQL systems emerged in the 2000s, and briefly contrast their features with traditional database systems.

Answer :

Emergence of NoSQL:
NoSQL databases emerged in the 2000s due to the need for **scalability, flexibility, and handling large volumes of unstructured or semi-structured data** (e.g., social media, IoT, big data).

Differences from Traditional Databases:

- **Schema Flexibility** – NoSQL databases do not require a fixed schema, unlike relational databases.
- **Horizontal Scalability** – NoSQL scales across multiple servers, whereas traditional DBMSs primarily scale vertically.
- **Data Model Differences** – NoSQL supports key-value, document, column-family, and graph data models, whereas relational databases use structured tables.
- **Eventual Consistency** – NoSQL prioritizes availability and partition tolerance over strict consistency (CAP theorem), whereas traditional DBMSs ensure ACID compliance.

1.15 Describe at least three tables that might be used to store information in a social- networking system such as Facebook.

Answer :

Three Tables in a Social Networking System (e.g., Facebook)

1. **Users Table** – Stores user details (e.g., `user_id`, `name`, `email`, `date_of_birth`).
2. **Posts Table** – Stores user-generated posts (e.g., `post_id`, `user_id`, `content`, `timestamp`).
3. **Friends Table** – Manages relationships between users (e.g., `user_id1`, `user_id2`, `friendship_status`, `created_at`).