Content Challenge

Mahir Hashimov
12695120
Unit 12
Deciphering Big Data
University of Essex Online
5 July 2024

Contents

Question 1

What are the disadvantages of the file-based system and how are these disadvantages addressed by the DBMS approach?

The file-based method has a number of serious drawbacks that make effective data management difficult. First of all, there is a lot of data redundancy and inconsistency since having several files frequently results in data duplication, which leads to inconsistencies. Inconsistent data might result, for instance, from several files storing different addresses for the same client.

Second, because complicated searches in a file-based system need specific scripting, data access is difficult. When extracting data from several files, complex code may be needed to manually connect the data, which would make the procedure time-consuming.

Thirdly, because data is dispersed over many files and formats and is challenging to integrate and retrieve, data isolation is a major problem in file-based systems. For example, intricate processes are required to correlate sales data from one file with customer data from another.

Fourthly, file-based systems have difficulty enforcing connections and limitations among files, which makes preserving data integrity difficult. For example, it's difficult to maintain referential consistency between orders and customers.

File-based systems are especially prone to atomicity problems since they cannot implement transactions, which puts incomplete changes at risk. Reliability of the data may be jeopardised if there was a power outage during an update procedure and files were left inconsistent.

Last but not least, file-based systems frequently have concurrent access anomalies as managing simultaneous access and alterations can be challenging. When two people update the same file at the same time, inconsistent results may result.

The Database Management System (DBMS) method, on the other hand, successfully handles these drawbacks. DBMS ensures consistent data by reducing data redundancy and inconsistency through centralised management and normalisation. SQL makes efficient data access possible by supporting complicated queries and making it simple to connect numerous tables.

In order to guarantee smooth data retrieval, DBMS further unifies data via a single database structure. Data integrity is preserved by DBMSs enforcing integrity requirements like primary and foreign key restrictions.

All-or-nothing modifications are guaranteed by atomic transactions in database management systems (DBMSs) that follow the ACID characteristics. Database management systems (DBMS) offer concurrency control technologies, such locking and multi-version concurrency control, to manage concurrent access and updates and guarantee data consistency.

Question 2

The consistency and reliability aspects of transactions are due to the "ACIDity" properties of transactions. Discuss each of these properties and how they relate to the concurrency control and recovery mechanisms. Give examples to illustrate your answer.

The ACID properties—Atomicity, Consistency, Isolation, and Durability—are responsible for the consistency and dependability of transactions.

A transaction is deemed atomic if every activity within it is finished; otherwise, it is cancelled. When money is transferred across accounts, for instance, it should either totally debit and credit both accounts, or not at all.

The database will always be in the same condition both before and after the transaction, thanks to consistency. Every transaction will exit the database in a legitimate state, abiding by all rules, including keeping correct balances following a transaction.

Concurrent transactions are kept apart from one another by isolation. Intermediate states are shielded from exposure by the fact that changes made to one transaction are not apparent to others until they are committed.

In the case of a system failure, durability makes guarantee that a transaction stays committed. Data permanence is ensured by the recording of committed transactions in non-volatile memory.

Data objects utilised in a transaction are locked by concurrency control techniques like locking, which guarantee isolation. Resilience is guaranteed by logging, which keeps track of every transaction for future reference. Commit protocols verify that every stage of a transaction is finished before concluding, ensuring atomicity and consistency.

Question 3

What are the privileges commonly granted to database users?

Users of databases are frequently given a range of rights to manage their access and activities inside the database. Users with the SELECT privilege can read data from tables. Users that have the INSERT privilege can contribute new data to tables. Users with the DELETE access can delete data from tables, whereas those with the UPDATE privilege can edit already-existing data.
The EXECUTE privilege also allows stored procedures to be executed. Users with the CREATE privilege can make new databases and tables. Table structure can be altered using the ALTER privilege, while databases and tables can be deleted with the DROP privilege.

Question 4

What restrictions are necessary to ensure that a view is updatable?

There are requirements that must be met in order to guarantee that a perspective is manageable. To guarantee a straight link between the view and the table, the view can only refer to one base table. Since aggregate functions, GROUP BY, and HAVING clauses prevent direct alteration of underlying table data, views using these constructs are not updatable.
Additionally, since these actions produce a result set that does not directly map to the underlying database, the view should not utilise DISTINCT, UNION, or UNION ALL. Finally, in order to preserve unique row identification and enable updates, the base table has to have a primary key, and the view also needs to have the primary key.

Question 5

Consider the following table:

Part (partNo, contract, partCost).

Which represents the cost negotiated under each contract for a part (a part may have a different price under each contract). Now consider the following view

ExpensiveParts, which contains the distinct part numbers for parts that cost more than £1,000:

CREATE VIEW ExpensiveParts (partNo).

ASSELECT DISTINCT partNo.

FROM Part.

WHERE partCost > 1000;

How you would maintain this as a materialised view and under what circumstances you would be able to maintain the view without having to access the underlying base table Part?

The ExpensiveParts view must first be first populated with data from the base table in order to remain a materialised view. Triggers or scheduled tasks can be used to update the materialised view whenever changes are made to the base database, hence achieving incremental maintenance.
Transaction logs can be used to record changes made to the view so that it can occasionally be kept current without requiring access to the base database. In order to guarantee instantaneous changes to the materialised view, triggers should be implemented on the base table. When a part's cost is modified, for instance, the trigger may determine whether the new cost satisfies the requirements and update the display appropriately.