

Lecture 25.

Overview of implementation issues

Objectives

Students should have a general awareness of the issues involved since they are important in evaluating DBMS and for knowing when to study more detail.

Students should have a good sense of security issues (read Elmasri, Chapter 22, handed out)

Query processing and optimization (Elmasri, Chapter 18)

See Elmasri Figure 18.1, p. 586

Problem of query optimization: Find the sequence of steps that produces the answer in the most efficient manner, given the database structure. (Note: Remember good structuring of the database for queries to be expected.)

Considerations:

- Narrow down intermediate result sets quickly. SELECT before JOIN
- Use access structures (indexes).

Some procedures/heuristics

- Start with the appropriate base table (N:1 from fields/columns in base table to other tables).
- Apply search criteria in base table, using first the criteria that use indexed fields.
- Do first those joins where the join field is indexed in the target table.

Transaction processing (Rob, Chapter 9)

Transactions and database state

Desirable properties of transactions (9.1.2, p. 462)

Transaction support in SQL (9.1.3, p. 463)

Database recovery and audit trail (9.1.4, p. 463)

System log

Rollback points

Concurrency control

Multiuser databases

Problems that can arise

- The lost update problem (9.2.1, p. 465, Elmasri Fig. 19.3(a), p. 634)
- The temporary update problem (Elmasri Fig. 19.3(b), p. 634)
- The uncommitted update problem (9.2.2, p.466)
- The incorrect summary problem (9.2.3, p.467, Elmasri Fig. 19.3©), p. 635)

Locking and other concurrency control techniques (9.3, p. 469)

Definition of locking. Granularity of locks

Binary locks vs Read/Write locks

Deadlock

Multiversion concurrency control

Database security and authorization (Elmasri, Chapter 22, read entire chapter)

Ease of access requires more security to prevent unwanted access.

General issues

Clearance for accessing computer in the first place; keeping hackers out (firewall)

Adequacy of the security access scheme (usually user name and password)

User-changeable passwords, but disallow passwords whose structure is too simple (require at least 6 characters, require at least one special character, etc.).

Security of data transmission

Audit trail, reports of unsuccessful access attempts

Access to data.

The possibilities for stipulating security definitions in a DBMS can be described along two dimensions:

	Protecting data integrity			Restricting data distribution
Function	Data definition	Data entry	Data modification	Data use: viewing printing
Level				
Access to system				
Data base				
Table				
Data field				
Record or data field selection based on conditions				

For each of these, there is the "meta-stipulation" of authorizing users to perform the function

To what extent can security stipulations be differentiated?

In data use, one can further distinguish between using individual data (with individuals identified or anonymized) and aggregate data. Data use includes operations, such as using a data field for selecting data.

Type of security definition

Directly by user name and password

By access codes assigned to users on the one hand and each level/function combination on the other

Example:

Access codes are a - z

User 1 has assigned three access codes: **akl** User 2 has **blt**

File 1 has read access for these codes: **gkm**

User 1 may read File 1, since she has access cod **k**, but user 2 may not

Protection of files from access outside the program (e.g., encryption)

Distributed databases (Chapter 10)

Computer networks. “Network of equals” vs. client-server architecture.

Objective: Most efficient use of computing and communication resources

Distributed database: Data are distributed over a computer network yet entire database can be accessed with the distribution transparent to the user.

Integrated multi-location database vs federated databases.

Aspects of homogeneity/diversity

- Differences in data models
- Differences in constraints
- Differences in query languages
- Semantic differences (naming, definitions)

Need a system catalog for the distributed database.

Client-server architecture: Database on central computer that does query and transaction processing. Users’ computers (clients) do GUI, possibly query parsing and optimization, perhaps have auxiliary local data.

How to distribute data.

General idea: Put data on the computer that serves most of the users of the data.

Methods of “data fragmentation” (see Elmasri, Figures 24.4 and 24.5, p. 776, for examples)

- By whole table
- One table distributed horizontally
- One table distributed vertically

Deliberate redundancy - update problem

- Store some tables or portions of tables in several places
- Store indexes in several places

Query processing in distributed databases

Concurrency control in distributed databases