

Motivation for Using Databases

Additional Material

Benefits of Using Databases

- Database application development can be a time-consuming, costly, and disruptive process. Why bother?
- The ultimate reason is that the benefits outweigh the costs.
- All computer projects should begin with a cost-benefit analysis.
 - Should be reviewed from time-to-time, or when requirements change

Benefits: Data Sharing

- When applications are developed using flat files, each application's files are designed for that specific application.
 - Files cannot be easily shared across applications.
 - The data can only be presented in the exact way the records are stored.
- Database systems have the unique property of being able to present multiple views of the data to different users.
 - This ability is enabled by the layers of abstraction provided by the DBMS, specifically the logical and external layers presented in Chapter 1.

Benefits: Controlled Redundancy

- The ability to produce views of the data tailored to individual applications eliminates the need for each application system to create master files tailored to its individual data needs.
- Through the design process known as normalization, DBMS objects (tables in relational technology) can be designed in such a way that they hold data about a single business entity.
- Many people believe that one of the goals of database technology, particular relational database technology, is to eliminate redundancy. This is untrue – foreign keys are redundant!

Benefits: Data Consistency

- By controlling redundancy, we obtain data consistency.
- Note, however, in some circumstances, databases are redundantly stored for recovery purposes. However, this does not pose a consistency issues for two reasons:
 - Database backups taken for the purposes of fast recovery from failures should not be available to database users.
 - Using database replication technology, the backup copy can be automatically updated very shortly after updates take place.

Benefits: Data Integrity Enforced

- With file systems, it is up to the applications to enforce data integrity. There are two reliability problems at play.
 - Application programs can contain errors that allow incorrect data to sneak through.
 - When loading data directly into a database or flat file,, the integrity controls in the applications are bypassed.
- Database Integrity Controls:
 - Are independent of application programs
 - Cannot be bypassed when they are in force

Benefits: Data Security

- ***Data security*** is the protection of the database from unauthorized users.
- Modern DBMS products have a built-in security subsystem that allows administrators to control access to the data contained in the database, giving each database user exactly the data required for its work, nothing more, and nothing less.
- The very fact that databases consolidate so much data in one location makes security controls essential because different users require different levels of access to the consolidated database.
- Database security is presented in detail in Chapter 10.

Benefits: Data Privacy

- ***Data privacy***: the protection of sensitive data, such as healthcare and financial data, from unauthorized disclosure.
- DBMS security subsystems usually support data encryption, which scrambles the data into a form that makes it useless to individuals unless they have the key required to decrypt (unscramble) the data.

Other Benefits

- **Intellectual investment:** Existing programs and logical data structures will not have to be redone when changes are made to the database.
- **Ease of use:** Users can gain access to data in a simple fashion since complexity is hidden by the DBMS.
- **Unanticipated requests:** Spontaneous requests for data can be handled by high-level query or report-generation languages.
- **Change:** The database can grow and change without interfering with established ways of using the data. This is possible because of the physical and logical data independence that databases offer when they provide multiple views of the stored data.

Costs: Complexity

- Creating integrated database designs that can serve the needs of an entire organization while allowing for growth and change that does not disrupt existing uses of the database is a complex undertaking.
- The additional complexity of consolidated database systems requires that the designers, developers, and users of these databases be trained in the structure of the database and the capabilities of the DBMS.

Costs: Software Costs

- The cost of DBMS software varies significantly depending on the required features, and maintenance costs.
 - Even so-called freeware DBMS products are not completely free if you read the fine print regarding license restrictions for commercial use and pay-as-you-go technical support plans.
- The DBMS is likely not the only software you will need.
 - Tools for data modeling and design, interactive database queries (SQL clients), report-writing, and the like will likely also be required.

Costs: Hardware Costs

- Databases usually require higher speed hardware than their flat file system counterparts in order to process queries efficiently.
- Databases generally require redundant hardware to mitigate the risk of a centralized storage system.
- DBMS software can require substantial amounts of computer memory and higher speed processors and networks than flat file systems.
 - These costs are at least somewhat offset by the consolidation of many application-specific data file servers into a centralized database server.