

First Lecture

Introduction

Data and Information

Data: It is an object used in computer science includes numbers, characters, symbols, videos, and images. There are three types of data:

1. Input Data
2. Output Data
3. Operational Data

Information: It is the data that have been processed in such a way of the person who use the Data.

Database Concept

A **database** is a collection of data and a set of rules that organize the data by specifying certain relationships among the data.

Through these rules, the user describes a logical format for the data. The data items are stored in a file, but the precise physical format of the file is of no concern to the user. A database administrator is a person who defines the rules that organize the data and also controls who should have access to what parts of the data. The user interacts with the database through a program called a database manager or a database management system (DBMS), informally known as a front end.

Databases Advantages

The logical idea behind a database is this: A database is a single collection of data, stored and maintained at one central location, to which many people have access as needed. However, the actual implementation may involve some other physical storage arrangement or access. The essence of a good database is that the users are unaware of the physical arrangements; the unified logical arrangement is all they see. As a result, a database offers many advantages over a simple file system:

1. shared access, so that many users can use one common, centralized set of data
2. minimal redundancy, so that individual users do not have to collect and maintain their own sets of data
3. data consistency, so that a change to a data value affects all users of the data value

4. data integrity, so that data values are protected against accidental or malicious undesirable changes
5. controlled access, so that only authorized users are allowed to view or to modify data values

A DBMS is designed to provide these advantages efficiently. However, as often happens, the objectives can conflict with each other. As we shall see, security interests can conflict with performance. This clash is not surprising because measures taken to enforce security often increase the computing system's size or complexity. What is surprising, though, is that security interests may also reduce the system's ability to provide data to users by limiting certain queries that would otherwise seem innocuous.

Database Management System DBMS

It consists of collection of interrelated data and set of programs to access the data. DBMS contains information about a particular enterprise and provides an environment that is both *convenient* and *efficient* to use.

Database Applications:

1. Banking: all transactions
2. Airlines: reservations, schedules
3. Universities: registration, grades
4. Sales: customers, products, purchases
5. Manufacturing: production, inventory, orders, supply chain
6. Human resources: employee records, salaries, tax deductions

((Databases touch all aspects of our lives))

Database Systems & File Systems

In the early days, database applications were built on top of file systems. There are some drawbacks of using file systems to store data:

1. Data redundancy and inconsistency
 - Multiple file formats, duplication of information in different files
2. Difficulty in accessing data
 - Need to write a new program to carry out each new task
3. Data isolation — multiple files and formats
4. Integrity problems
 - Integrity constraints (e.g. account balance > 0) become part of program code*
 - Hard to add new constraints or change existing ones*
5. Atomicity of updates
 - Failures may leave database in an inconsistent state with partial updates carried out*
 - E.g. transfer of funds from one account to another should either complete or not happen at all*
6. Concurrent access by multiple users

Concurrent accessed needed for performance

Uncontrolled concurrent accesses can lead to inconsistencies

E.g. two people reading a balance and updating it at the same time

7. Security problems

((Database systems offer solutions to all the above problems))

Database Levels Abstraction

1. Physical level describes how a record (e.g., customer) is stored.
2. Logical level: describes data stored in database, and the relationships among the data.

type customer = record

name: string; *street*: string; *city*: integer;

end;

3. View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

Schemas & Instances

Similar to types and variables in programming languages

1. Schema: is the logical structure of the database. Analogous to type information of a variable in a program (e.g., X: Integer). Applications depend on the logical schema. The student database system consists of information about a set of students (e.g., STD (S_Id, S_Name, S_Addr)).
2. Instance: is the actual content of the database at a particular point in time. Analogous to the value of a variable (e.g., X=4). Instances of students database system are (1, "Ali", "Mosul"), (2, "Muna", "Baghdad").

Data Models

A collection of tools for describing:

1. data
2. data relationships
3. data semantics
4. data constraints

Data Models Types

1. Entity-Relationship ER Model (we will explain it in Chapter 2)
2. Enhanced Entity Relationship EER Model (we will explain it in Chapter 3)
3. Relational Model RM (we will explain it in Chapter 4)
4. Object-Oriented Model
5. Semi-Structured Data Models
6. Network model
7. Hierarchical model

Structured Query Language SQL

Widely used non-procedural language. It has four parts:

Data Definition Language DDL

It is a specification notation for defining the database schema.

E.g. create table *account* (*account-number* char(10), *balance* integer)

Data Storage and Definition Language DSDL

It is a language in which the storage structure and access methods used by the database system are specified.

Data Manipulation Language DML

It is a language for accessing and manipulating the data organized by the appropriate data model. The most important commands in DML are *insert into*, *delete from*, *update*, *select*.

Data Control Language DCL

It is a language in which grant and revoke privileges specified.

The following are some examples of SQL commands:

□ E.g. find the name of the customer with customer-id 192

```
select customer.customer-name
```

```
from customer
```

```
where customer.customer-id = 192
```

□ E.g. find the balances of all accounts held by the customer with customer-id 192

```
select account.balance
```

```
from depositor, account
```

```
where depositor.customer-id = 192 and
```

```
depositor.account-number = account.account-number
```

□ Application programs generally access databases through one of language extensions to allow embedded SQL or application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database.

Database Users

Users are differentiated by the way they expect to interact with the system

1. Application programmers – interact with system through DML calls.
2. Sophisticated (analyst) users – form requests in a database query language.
3. Specialized users – write specialized database applications that do not fit into the traditional data processing framework.
4. Naïve users – invoke one of the permanent application programs that have been written previously.

Database Administrator

Is the person who coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.

■ **Database administrator's duties**

1. Schema definition
2. Storage structure and access method definition
3. Schema and physical organization modification
4. Granting user authority to access the database
5. Specifying integrity constraints
6. Acting as liaison with users
7. Monitoring performance and responding to changes in requirements

Transaction Management

Transaction: is a collection of operations that performs a single logical function in a database application.

- **Transaction Management Component** ensures that the database remains in a consistent (correct) state despite system failures.
- **Concurrency Control Manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

Storage Management

Storage Manager: is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

■ **The storage manager is responsible of the following tasks:**

1. Interaction with the file manager
2. Efficient storing, retrieving, and updating of data