Database Chapter One Outline

A **database** is a collection of interrelated data items

Related to some enterprise, organizations,…

Interrelated data items

Gracefully organized

Easy to access

**A DBMS (DataBase Management System) provide a way to store and retrieve information about a particular enterprise**

**Collection of interrelated data**

**Set of programs to access the data**

**An environment that is both *convenient* , *efficient,* *credible,* and *secure* to use**

**A DBS (DataBase System) contains the following components:**

**Database**

**DBMS**

**Hardware to support DBMS**

**Operators**

Drawbacks of using file systems to store data:

Data redundancy and inconsistency

Difficulty in accessing data

Integrity problems

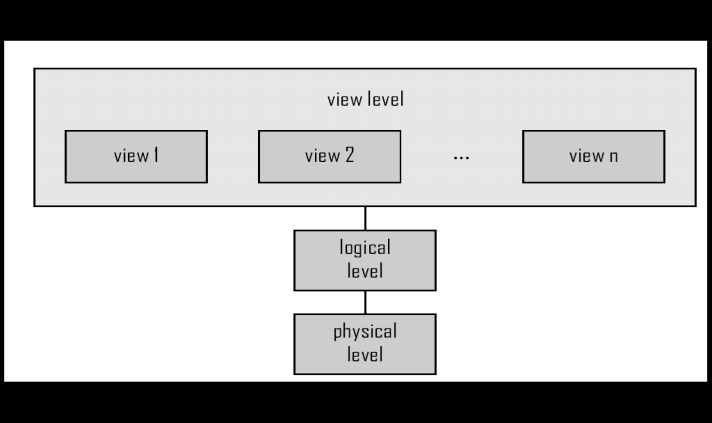
Data isolation — multiple files and formats

Data independence is terrible

Atomicity of updates

Concurrent access by multiple users

Security problems



**Physical level:** describes *how* a record (e.g., customer) is stored.

**Logical level:** describes *what* data are stored in database, and what relationships exist among those data.

**View level:** application programs hide details of data types. Views can also hide information (such as an employee’s salary) for security purposes.

**Instances and Schemas**

**Schema** – the logical structure of the database

Physical schema(Internal schema): database design at the physical level

Logical schema(Conceptual schema): database design at the logical level

Subschema(external schema): describe different views of the database

**Instance** – the actual content of the database at a particular point in time

**Physical Data Independence** – the ability to modify the physical schema without changing the logical schema

**Logical Data Independence** – the ability to modify the logical schema without updating application

**Data Model**: A collection of conceptual tools for describing

Data

Data relationships

Data semantics

Data constraints

Relational model

Entity-Relationship data model (mainly for database design)

Object-based data models (Object-oriented and Object-relational)

Semistructured data model (XML) 半结构化模型

Other older models:

Network model 网络模型

Hierarchical model 层次模型

**Data Definition Language (DDL):**

Specification notation for defining the database schema

DDL compiler generates a set of tables stored in a *data dictionary*

Data dictionary contains metadata (i.e., data about data)

Database schema

Data *storage and definition* language

Specifies the storage structure and access methods used

Integrity constraints

Domain constraints

Referential integrity (**references** constraint in SQL)

Assertions

Authorization

**Data Manipulation Language (DML)**

Language for accessing and manipulating the data organized by the appropriate data model

Query

Insert

Delete

Update

Two classes of languages

**Procedural** – user specifies what data is required and how to get those data

**Declarative (nonprocedural)** – user specifies what data is required without specifying how to get those data

SQL is the most widely used query language

Application programs generally access databases through one of

Language extensions to allow embedded SQL

Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

**Database Design**

**Logical Design** – Deciding on the database schema. Database design requires that we find a “good” collection of relation schemas.

Business decision – What attributes should we record in the database?

Computer Science decision – What relation schemas should we have and how should the attributes be distributed among the various relation schemas?

**Physical Design** – Deciding on the physical layout of the database

**The Entity-Relationship Model**

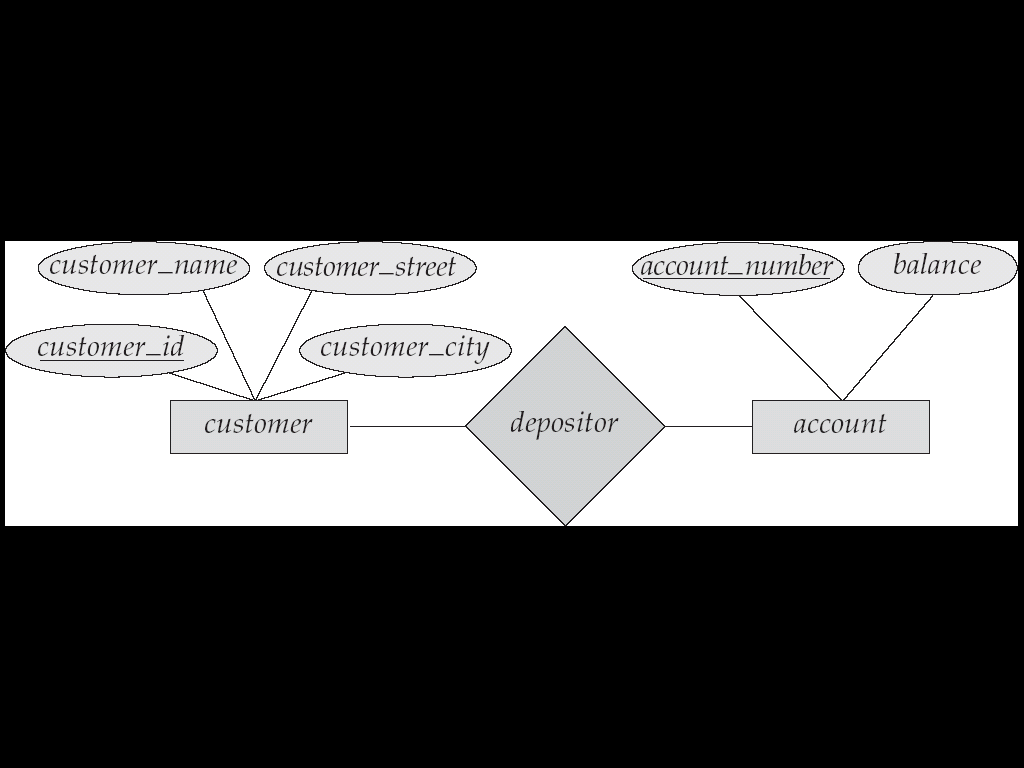
Models an enterprise as a collection of *entities* and *relationships*

Entity: a “thing” or “object” in the enterprise that is distinguishable from other objects

* + - Described by a set of *attributes*

Relationship: an association among several entities

Represented diagrammatically by an *entity-relationship diagram:*



Rectangles: Entity Sets

Ellipses: Attributes

Diamonds: Sets of relationship

Lines: Link attributes to entity sets and entity sets to relationships

**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 to the following tasks:

Interaction with the file manager

Efficient storing, retrieving and updating of data

Issues:

Storage access

File organization

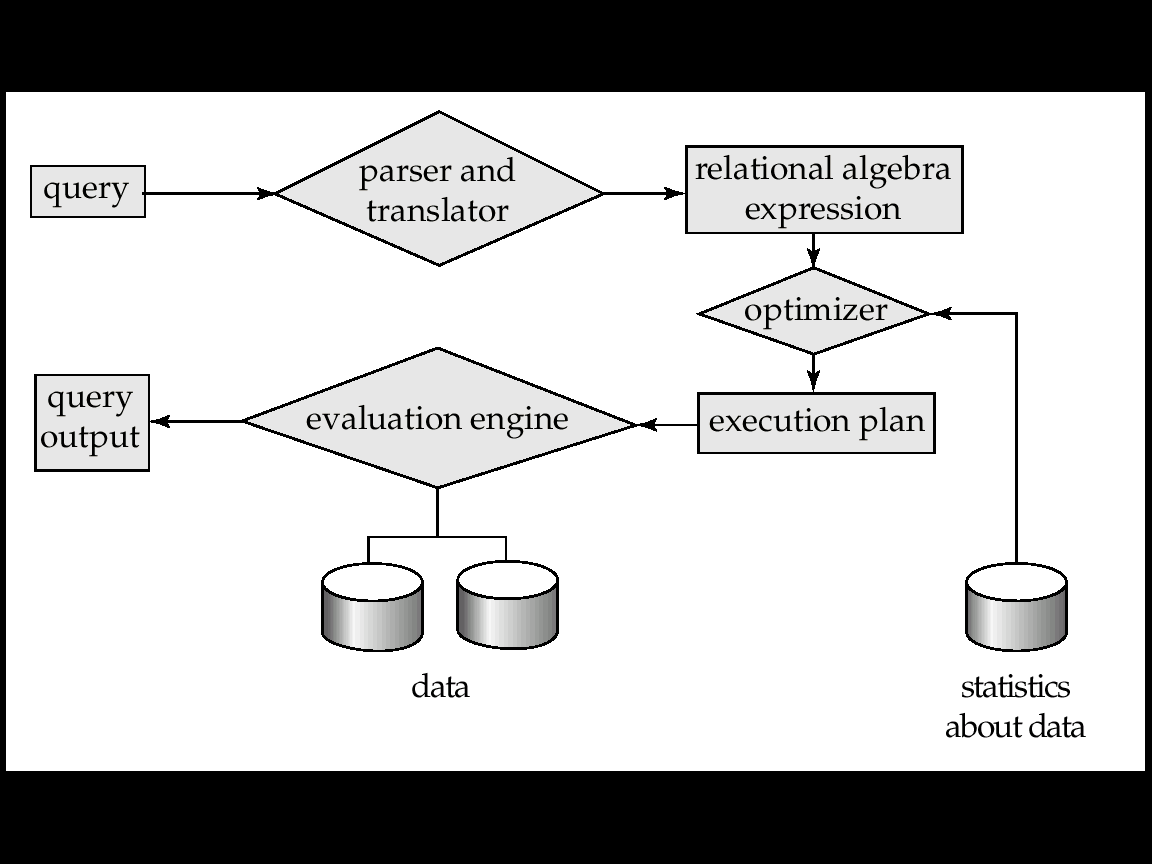
Indexing and hashing

**Query Processing**

1. Parsing and translation

2. Optimization

3. Evaluation



**Transaction Management**

A **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 (e.g., power failures and operating system crashes) and transaction failures.

**Concurrency-control manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.

**Database Architecture**

The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running:

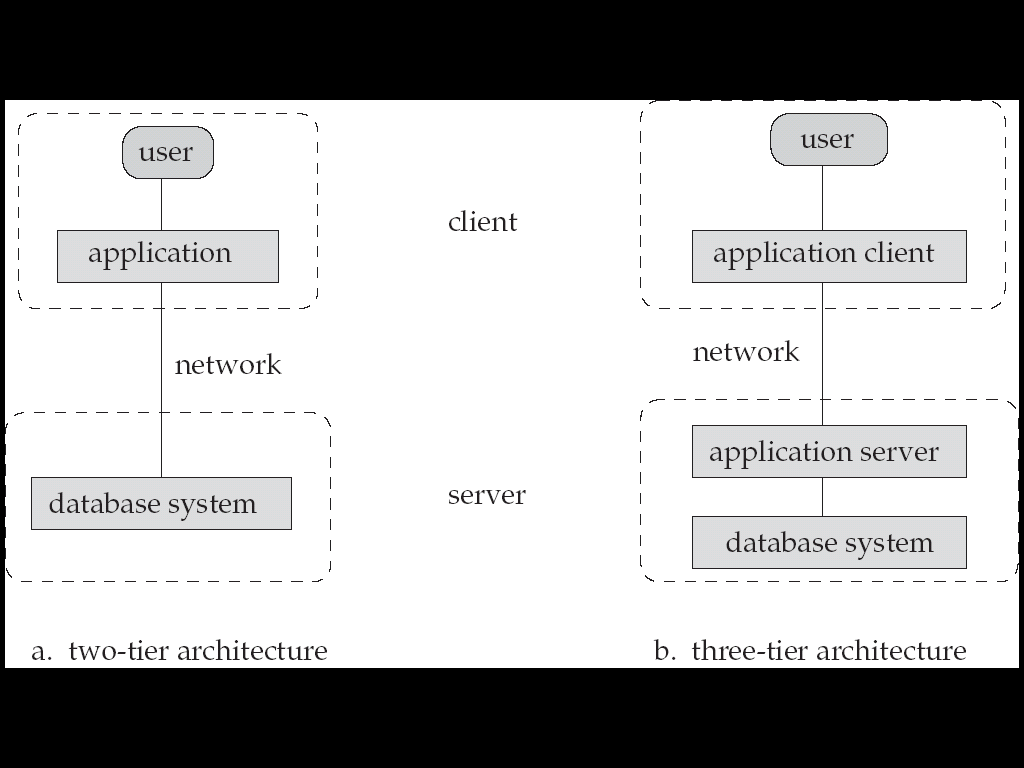
Centralized 集中式

Client-server 客户/服务

Parallel (multi-processor) 并行

Distributed 分布式

Two-tier Architecture And Three-tier Architecture



**Database Users**

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

**Application programmers – interact with system through DML calls** 应用程序员

**Sophisticated users – form requests in a database query language**富有经验的用户

**Specialized users – write specialized database applications that do not fit into the traditional data processing framework**专业用户

**Naïve users – invoke one of the permanent application programs that have been written previously**无经验用户

**Database Administrator**

Database administrator's duties include:

Schema definition

Storage structure and access method definition

Schema and physical organization modification

Granting user authority to access the database

Specifying integrity constraints

Acting as liaison with users

Monitoring performance and responding to changes in requirements