CptS 451- Introduction to Database Systems

Overview of Database Systems

(DMS ch-1)

Instructor: Sakire Arslan Ay

WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Database Management Systems

Introduction

Database

- What is a database?
  - A very large, integrated collection of data.
- Examples of databases.
- What information databases capture?
  - Entities (e.g., students, courses)
  - Relationships (e.g., Jack is taking CptS451)

Database Management System

- What is a Database Management System (DBMS)?
  - DBMS is a software package designed to store and manage databases.
- Examples of DBMSs:
  - Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
  - Open source: MySQL (Sun/Oracle), PostgreSQL, CouchDB, SQLite (library)

Traditional DBMS Goals

- Efficient *management* of
  massive amounts of (terabytes)
  persistent (outlasts creator),
  reliable (outlasts crashes)
  shared information (multiple users).

Database Management Systems

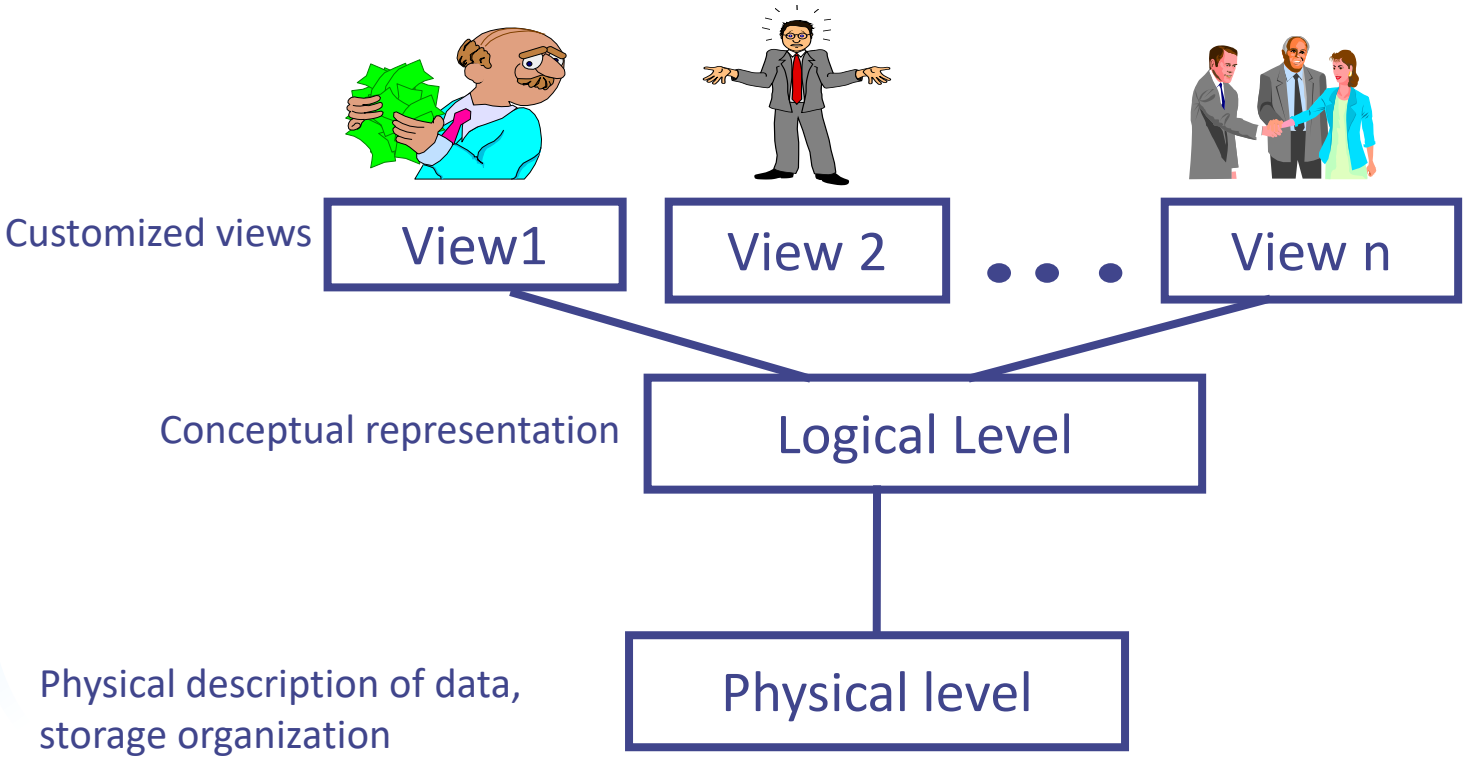
- Massive
- Efficient
- Persistent
- Reliable and safe
- Multi-user
- Convenient

Databases and File Systems

- DBMSs evolved from file systems.
- DBMSs provide many features that traditional file systems do not.
  - Data consistency in presence of concurrency
  - Reliability in presence of failures and system crashes.
  - Efficient associative access to very large amounts of data
  - A high level Query language (SQL) to define, create, access, and manipulate data.
  - Security and authorization
  - Prevention of data redundancy and inconsistencies
  - Data abstraction and support for multiple data views

Levels of Abstraction

- Hiding system complexity, physical storage details from users and application programs



Physical data independence: Physical description of data can be changed easily without affecting application programs

This course: Core Database Issues

- **Data Models**
  - Relational Model
  - ER Model
- **DBMS Languages**
  - Relational Algebra (formal); SQL (commercial),
    - Data Definition Language (DDL)
    - Data Manipulation Language (DML)
- **Schema Design**
  - Normal forms : BCNF, 3NF
- **DBMS Internals**
  - Storage Management
  - Indexing : B+Tree and Hash indexes
  - Query Optimization and Processing (if time permits)
- **Transaction Processing Techniques**
  - to support concurrent access and reliability in the presence of failures

Data Model

- A *data model* is a collection of concepts for describing data.
- The *relational model of data* is the most widely used model today.
  - Main concept: *relation*, basically a table with rows and columns.
  - Every relation has a *schema*, which describes the columns, or fields.

- Other Data Model Examples:
  - Entity-Relationship (ER) Model
  - Semi-structured Model (XML, JSON),
  - Object-Oriented Model (e.g., ODL), etc.

Schemas and Instances

- Schema:
  - overall design, structure, and constraints over the database
  - referred to as metadata
- Instance:
  - set of data currently instantiated in database

Example:

Schema:

Tables

Emp (ename, dep#)
Dept(dep#, dname, mgr)

Constraints

each department has a single manager

Instance:

Emp

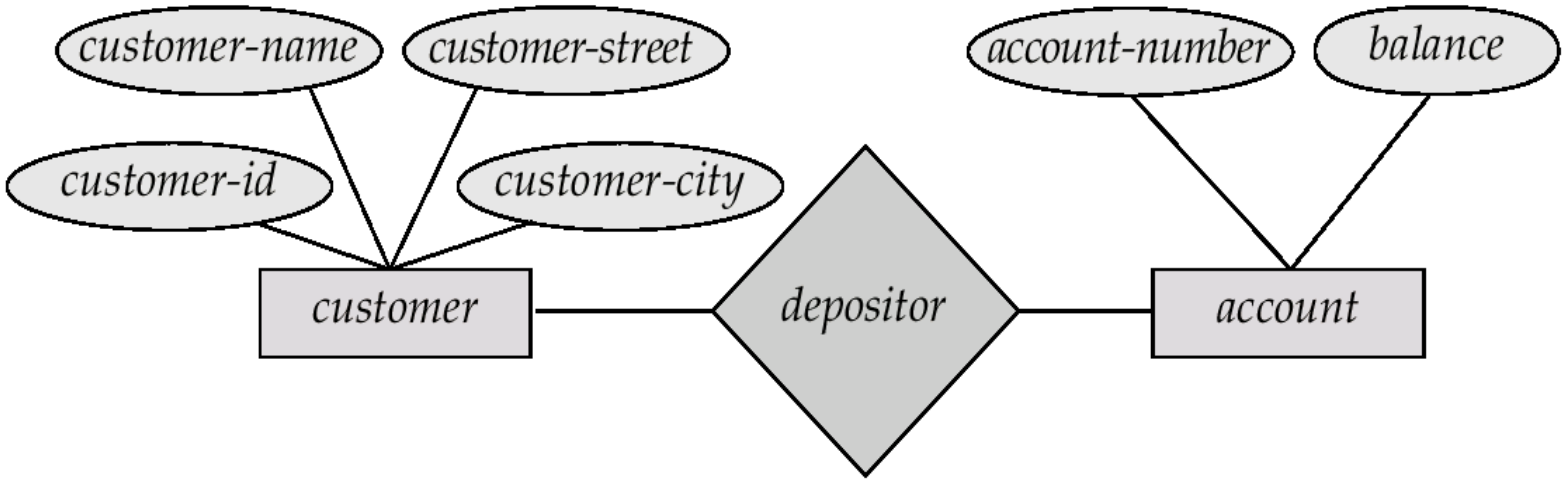
ename	dept
John	10
Cindy	15
Martha	10

Dept

dept	dname	mgr
10	toy	John
15	sales	Cindy

Entity-Relationship Model

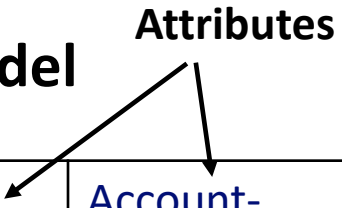
- A example schema in the entity-relationship model



Relational Model

- Uses a collection of relations (tables) to represent data and relationships among data

Example of tabular data in the relational model



Customer-id	Customer-name	Customer-street	Customer-city	Account-number
192-83-7465	Johnson	Alma	Seattle	A-101
019-28-3746	Smith	North	Portland	A-203
192-83-7465	Johnson	Alma	Seattle	A-201
321-12-3123	Jones	Main	Pullman	A-217
019-28-3746	Smith	North	Portland	A-201

A Sample Relational Database

customer-id	*customer-name*	*customer-street*	*customer-city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

account-number	*balance*
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

customer-id	*account-number*
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Semi-structured Data Model

- Allows the specification of data where individual data items (instances) of the same type may have different set of attributes.
- XML (Extensible Markup Language) is widely used to represent semi-structured data
  - XML has become the basis for all new generation data interchange formats.
  - A wide variety of tools is available for parsing, browsing and querying XML documents/data

SQL

- **SQL**: widely used non-procedural DBMS language for relational databases
  - Example DML Query in SQL:

    find the name of the customer with customer-id 192-83-7465

    select customer.customer-name
    from customer
    where customer.customer-id = '192-83-7465'

- Basic SQL has limited expressability
  - cannot implement any arbitrary function in SQL

DBMS Languages

- Data Definition Language (DDL)
  - DDL = the language used to describe a schema
  - Data dictionary/directory = a compiled description of a schema
- Data Manipulation Language (DML)
  - DML= Language users use to ask questions about (query) the database, and to change the data in the database.

Data Definition Language (DDL)

- Specification notation for defining the database schema and integrity constraints

  SQL Example: **create table** *customer*(
  customer-id **char**(11),
  customer-*name* **varchar**(20)**,**
  customer-street **varchar**(20),
  customer-city **varchar**(20) ,
  primarykey (customer-id)
)

- DDL compiler generates a set of table templates stored in a ***data dictionary***
  - Data dictionary contains
    - Database schema
    - Integrity constraints
- The data values stored in a database must satisfy the schema and constraints defined in the data dictionary

Data Definition Language (DDL) (cont.)

- DDL provides facilities to specify such constraints
  - Domain Constraints
    - Constraint on the value an attribute can take
  - Referential integrity
    - A value that appears in one relation for a set of attributes should appear in a certain set of attributes in another relation
  - Assertions
    - Conditions that database must always satisfy
      - E.g. every department must offer at least five courses every semester
    - Domain and referential integrity are special forms of assertions
  - Authorization
    - Users may have different access rights on various data values
      - Read, insert, update, delete, authorization

Transaction Management

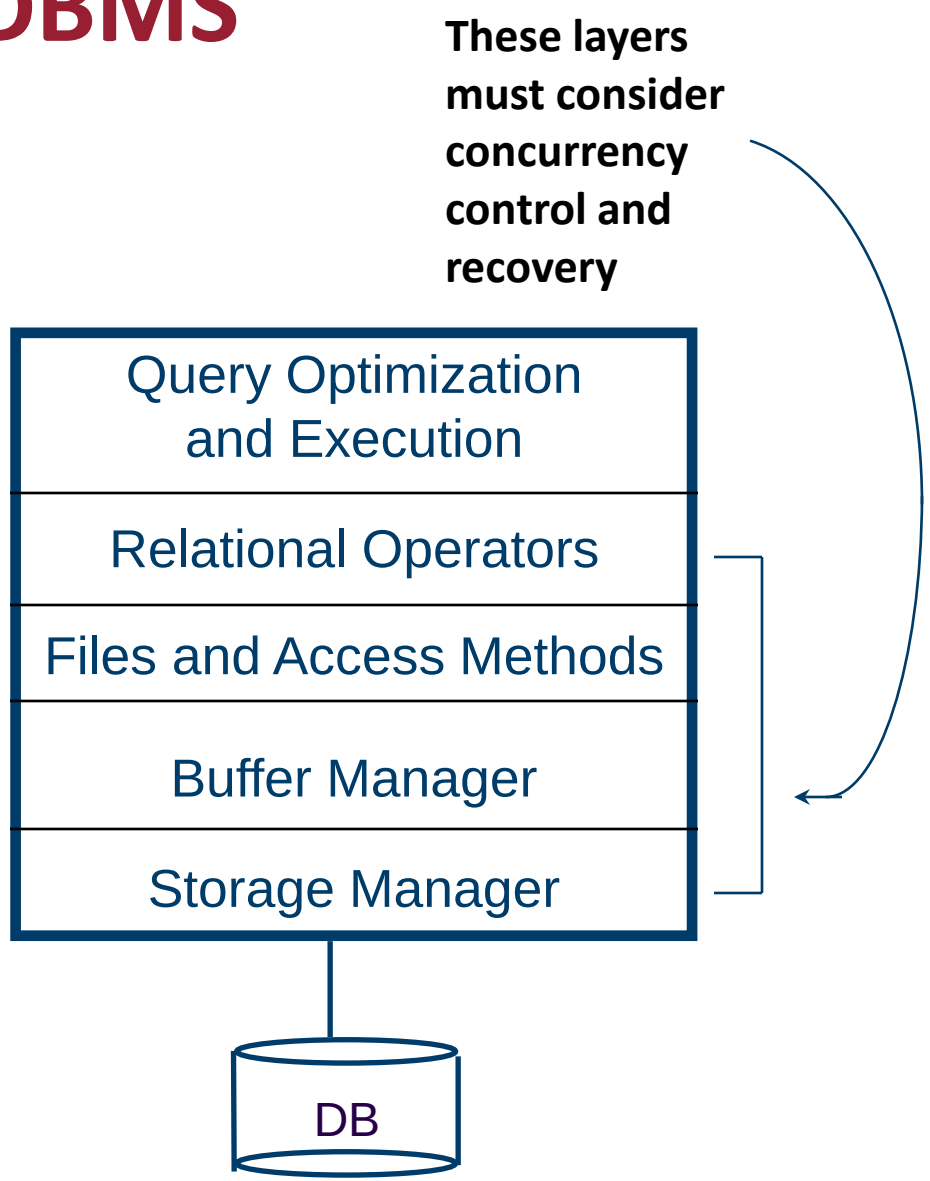
- What if the system fails?
- What if more than one user is concurrently updating the same data?
- A **transaction** is a collection of operations that performs a single logical function in a database application
- **Transaction-management component** ensures that the database remains in a consistent (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.

Transaction Concept

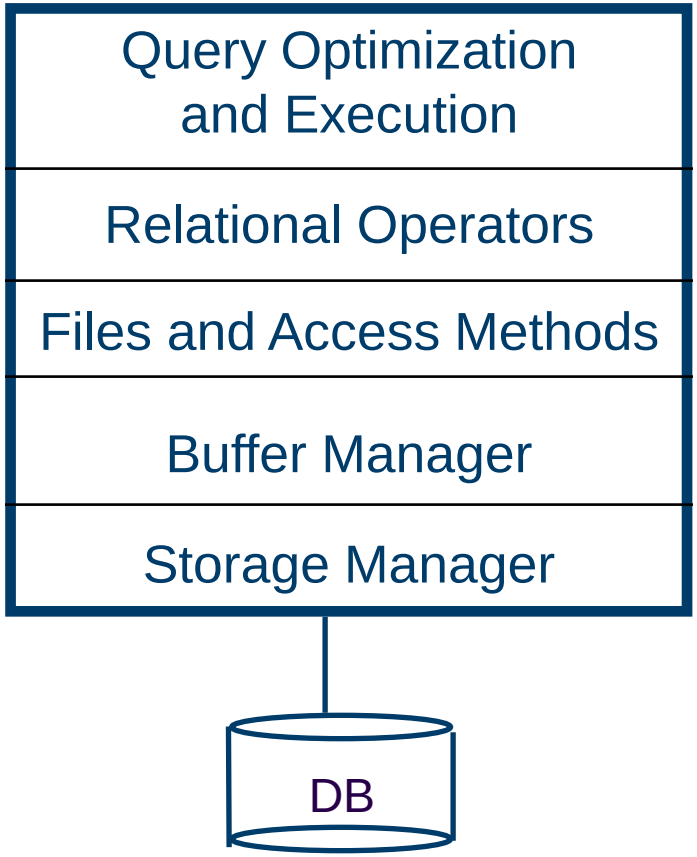
- ***Atomicity:***
  - all or nothing execution.
- ***Consistency:***
  - execution of a transaction leaves system state as well as the state of the real world consistent.
- ***Isolation:***
  - partial effects of a transaction are hidden from each other.
- ***Durability:***
  - transactions that have committed will survive permanently.

Structure of a DBMS

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- This is one of several possible architectures; each system has its own variations.



Structure of a DBMS

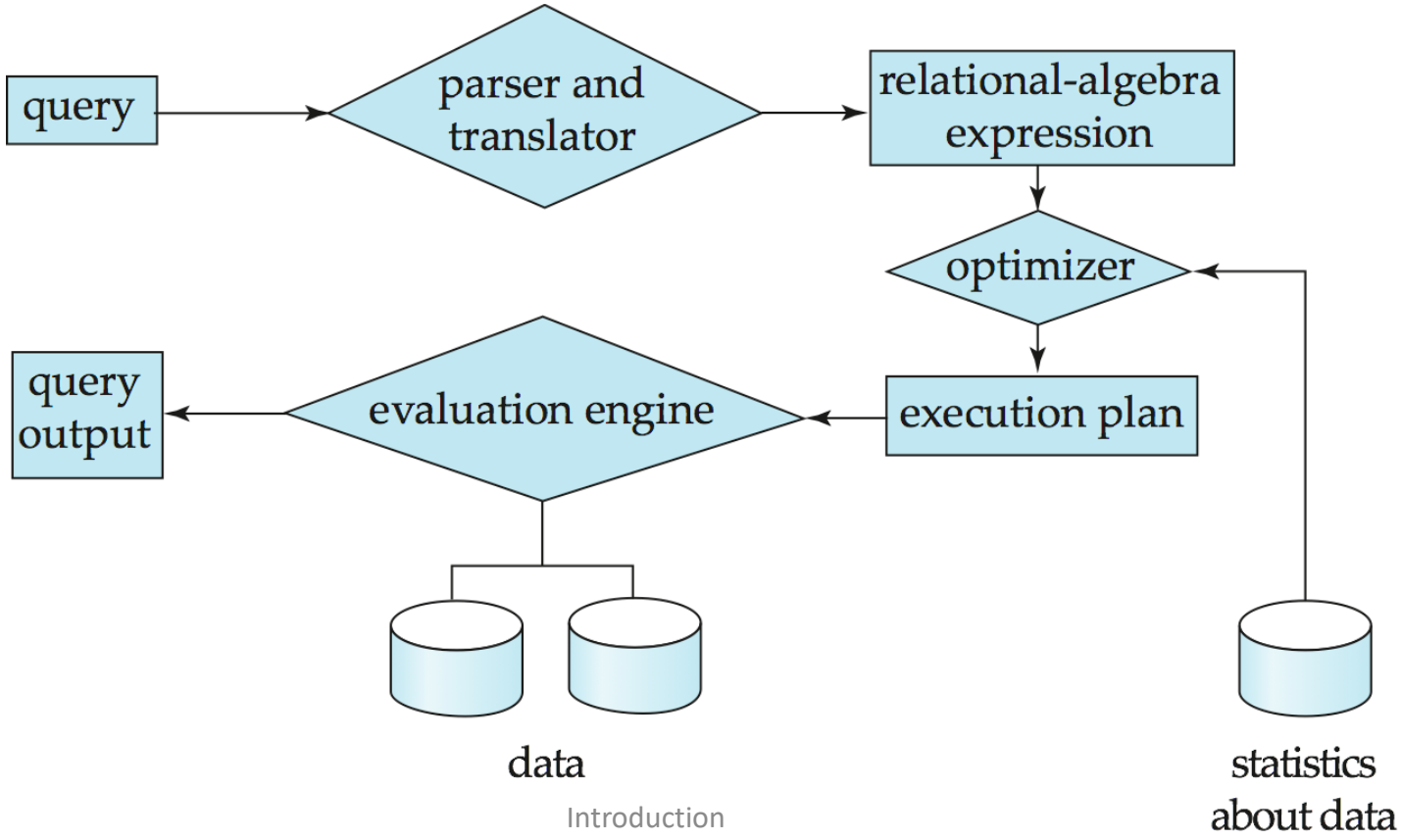


Storage Management

- **Storage manager** is a program module that provides the interface between the low-level data stored in the database and the 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



People Involved with DBMSs

- Database designer
  - Establishes schema
- Application programmers
  - Write programs that operate on database
- Database administrator (DBA)
  - DBA = 'super-user' for a database, similar to a system administrator.
  - DBA can define schemas, views, authorization, indexes, tuning parameters, etc. They make sure the database keeps running smoothly.
- End users

Large number of jobs available for each of the above tasks!!