

CptS 451- Introduction to Database Systems

Overview of Database Systems (DMS ch-1)

Instructor: Sakire Arslan Ay



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, reliable, convenient, and safe *management of massive amounts of* (terabytes) *persistent* (outlasts creator), *reliable* (outlasts crashes) *shared information* (multiple users).

Database Management Systems

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

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

Key Database Technologies

- **Data Models**
 - Relational Model
 - ER Model
- **DBMS Languages**
 - SQL, Relational Algebra
 - Data Definition Language (DDL)
 - Data Manipulation Language (DML)
- **Transaction Processing Techniques**
 - to support concurrent access and reliability in the presence of failures
- **DBMS Internals**
 - Storage Management
 - Query Optimization and Processing

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),
 - 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

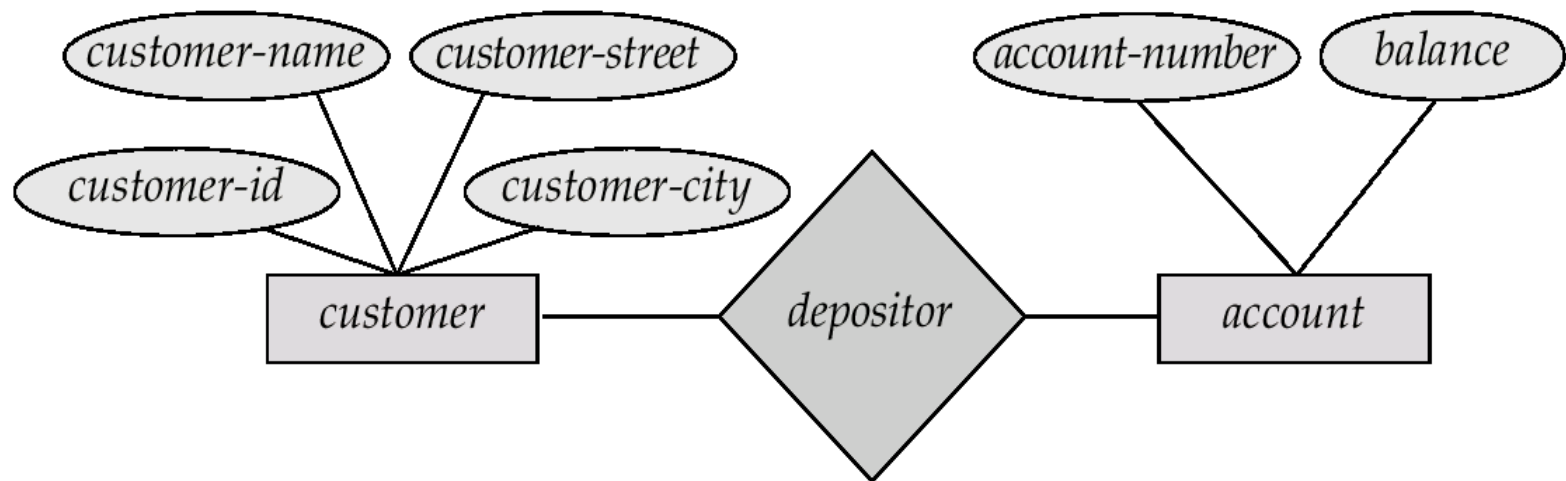
ename	dept
John	10
Cindy	15
Martha	10

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

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.

Key Database Technologies

- **Data Models**
 - Relational Model
 - ER Model
- **DBMS Languages**
 - SQL, Relational Algebra
 - Data Definition Language (DDL)
 - Data Manipulation Language (DML)
- **Transaction Processing Techniques**
 - to support concurrent access and reliability in the presence of failures
- **DBMS Internals**
 - Storage Management
 - Query Optimization and Processing

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

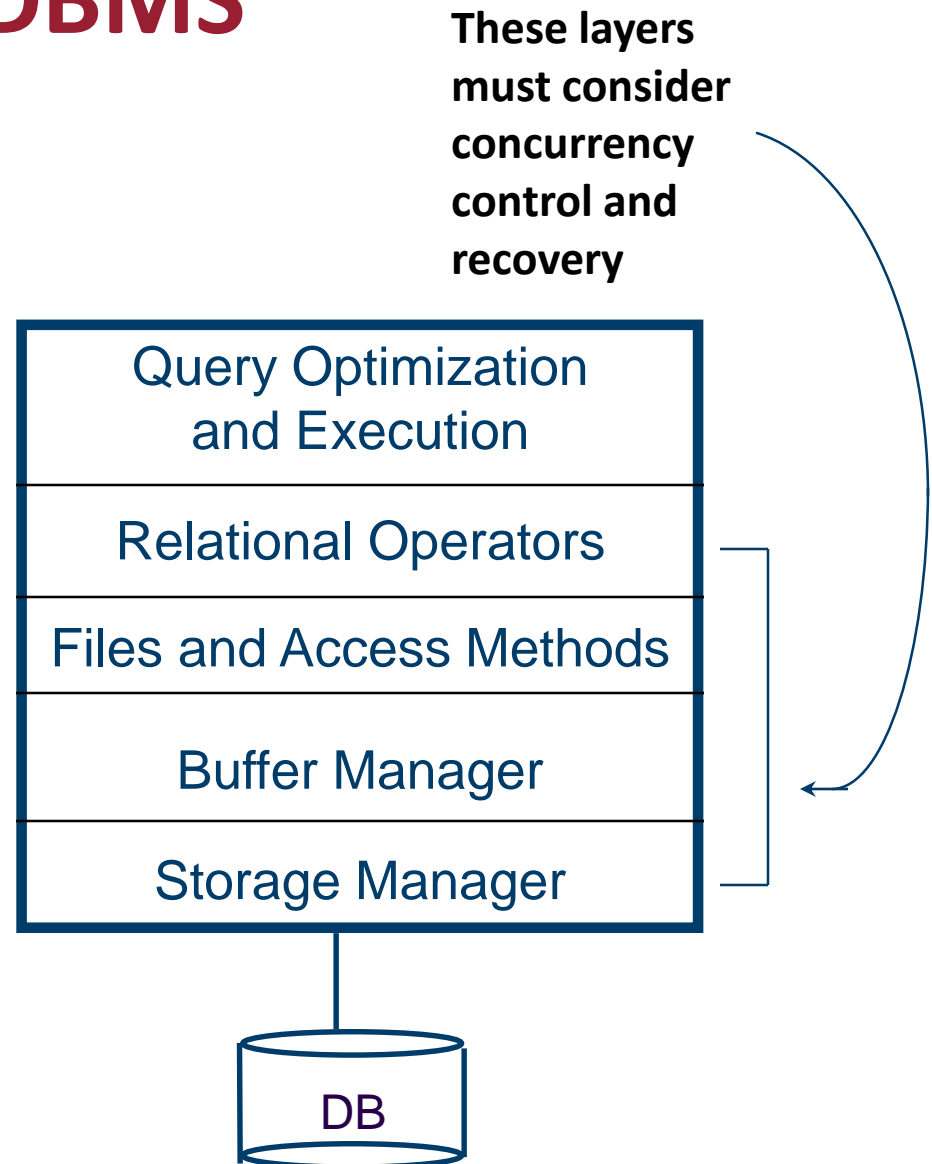
- **Atomicity:**
 - all or nothing execution.
- **Consistency:**
 - execution of a transaction leaves system state as well as the state of the real world consistent.
- **Isolation:**
 - partial effects of a transaction are hidden from each other.
- **Durability:**
 - transactions that have committed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

Key Database Technologies

- **Data Models**
 - Relational Model
 - ER Model
- **DBMS Languages**
 - SQL, Relational Algebra
 - Data Definition Language (DDL)
 - Data Manipulation Language (DML)
- **Transaction Processing Techniques**
 - to support concurrent access and reliability in the presence of failures
- **DBMS Internals**
 - Storage Management
 - Query Optimization and Processing

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.

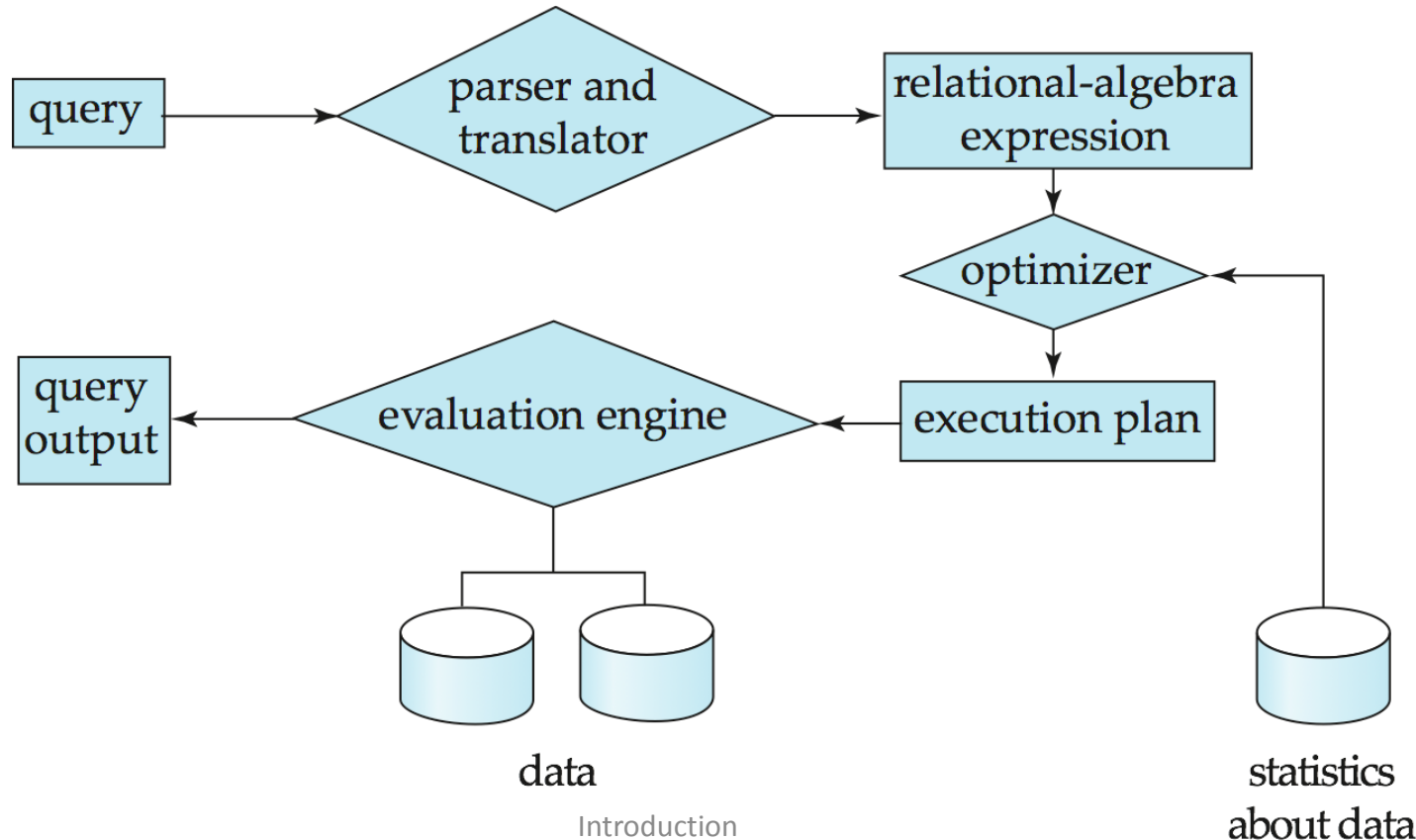


Storage Management

- **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!!