Data Base Management Systems

Sanjay Moulik

IIIT Guwahati

What is a Database System?

- A Database System is essentially a computerized record-keeping system.
- A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access those data.
- Database systems are designed to manage large volume of information efficiently and correctly.
- The primary goal of DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing database information.

Types of Databases

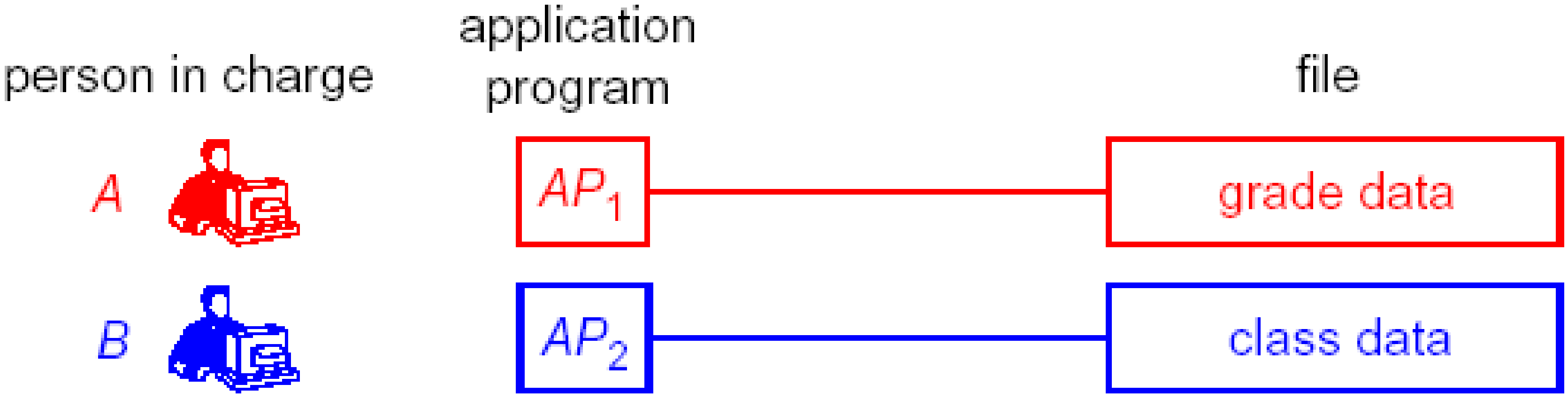
- Numeric and Textual Databases
- Multimedia Databases
- Geographic Information Systems (GIS)
- Data Warehouses
- Real-time and Active Databases

Database applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives

Traditional Data Management using file systems

- A time when there were no database systems…
- Scenario "university administration"



Problems in File environment

- Data redundancy & inconsistency
  - Various files, likely to have different format and program written in different languages
  - Some information may be duplicated in several files
    - Data redundancy
    - Data inconsistency
- Program & data dependency
  - If new request (not anticipated, when designed)
    - Have to write a new program
    - Very difficult, as data are scattered in various files in various formats

Problems in File environment

- Lack of flexibility
  - Analysis of data as well as the realization of the new application is problematic
  - Data from several files can only be combined with very high costs
- Poor security
- Lack of data-sharing and availability
- No concurrency control

Requirements & advantages of DBMS

- Data independence
  - Independence of the application programs from the detail of the data representations and data storage.
  - DBMS can provide an abstract view of the data.
- Efficient data access: utilizes a variety of sophisticated techniques to store and retrieve data efficiently.
- Common databases for all current and future application programs.
- Concurrent data access
  - Simultaneous access to the same data by different users
- Controlling Redundancy
  - Avoiding copies of same data by an integrated view on data
  - Control redundancy for improving performance

Requirements & advantages of DBMS

- Consistency of data
  - Must ensure consistency for controlled redundancy
- Integrity of data
  - Correctness and completeness of data
    - Formulation of integrity constraints
- Data security
  - Protection of databases against unauthorized access
  - Access control with authentication
- Back up & Recovery
  - Protections against the consequences of system errors

    - **Providing persistent storage for program objects and data structures**
    - **Providing multiple user interface**
    - **Representing complex relationships among data**
    - **Enforcing integrity constraints and providing backup and recovery**

Functions of DBMS

- Data definition: **Specifies content and structure of database and defines each data element**
- Data manipulation: **Manipulates data in a database**
- Data security and integrity: **Monitors user requests and rejects any unauthorized attempts**
- Data recovery and concurrency: **Enforces certain controls for recovery and concurrency**
- Data dictionary: Stores definitions of data elements, and data characteristics
- Performance: Functions should be performed efficiently

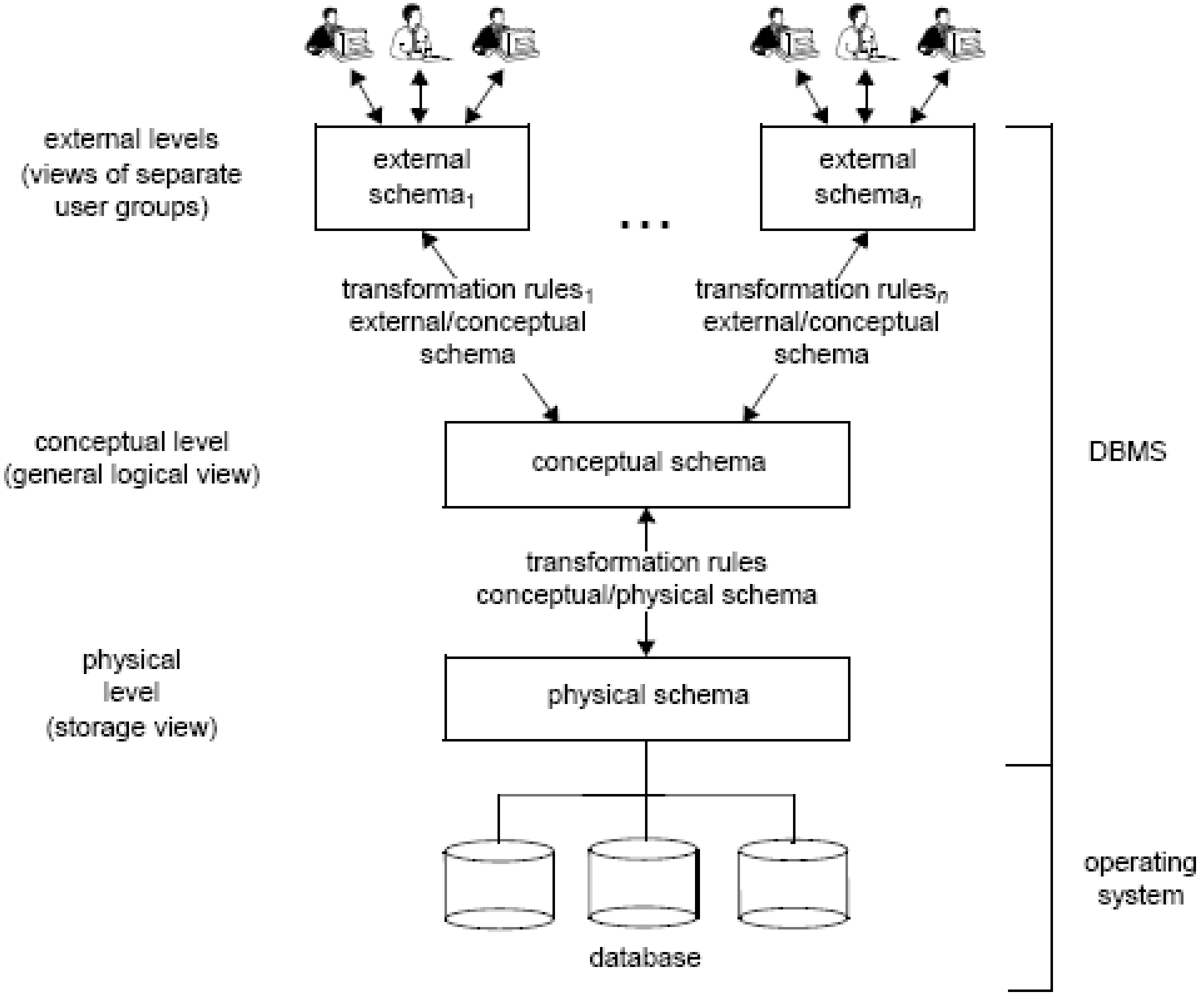
DBMS Architecture

- To handle the data efficiently and to hides the internal complexity of a system, several level of abstractions are defined.
  - Physical / Internal level
  - Logical / Conceptual level
  - View / External level
- The description of a database is called the database schema.
  - Internal Schema
  - Conceptual Schema
  - External Schema

DBMS Architecture

- Internal Schema : **Describes physical storage structure of database**
- Conceptual Schema : **Describes structure of whole database for a community of users. [i.e., what data are stored, and what relationship exist among those data]**
- External Schema : **Each view describes that part of database that a particular user requires, and hides the rest.**

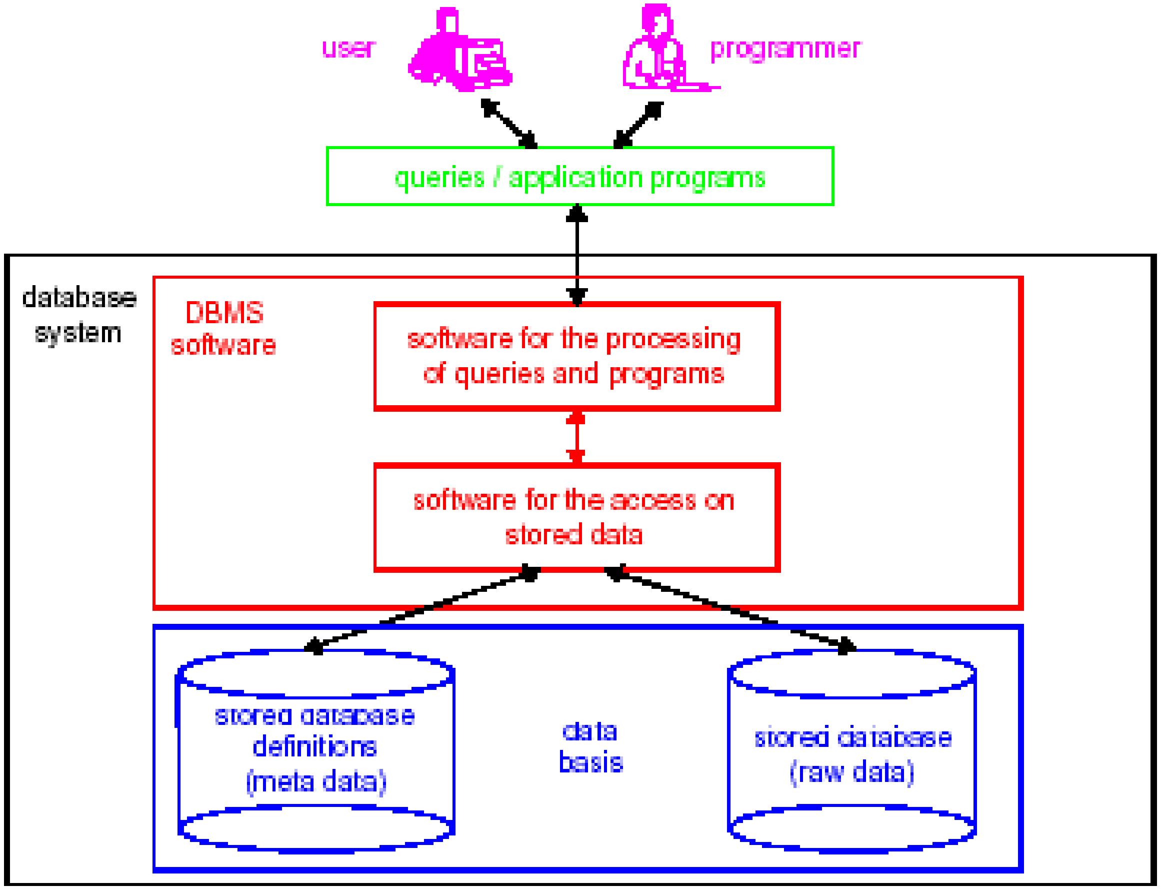
Three level model



DBMS Architecture

- Data Independence: denotes the property that higher levels of the model are not influenced by changes of lower levels
- **Logical data independence** : capacity to change conceptual schema without having to change external schema.
- **Physical data independence :** capacity to change internal schema without changing conceptual schema.

Database system


user
programmer
queries / application programs
database system
DBMS software
software for the processing of queries and programs
software for the access on stored data
stored database definitions (meta data)
data basis
stored database (raw data)

Data Models

- Data Model is to a Database what a Building plan or a blueprint is to a Building
- A Data Model is the conceptual design of a database
- Mathematical formalism consisting of a notation for describing the data of interest and of a set of operations for manipulating these data.
- A set of concepts to describe the *structure* of a database, and certain *constraints* that the database should obey.

Data Models

- Description of structure of a database
  - Data
  - Data relationships
  - Data constraints

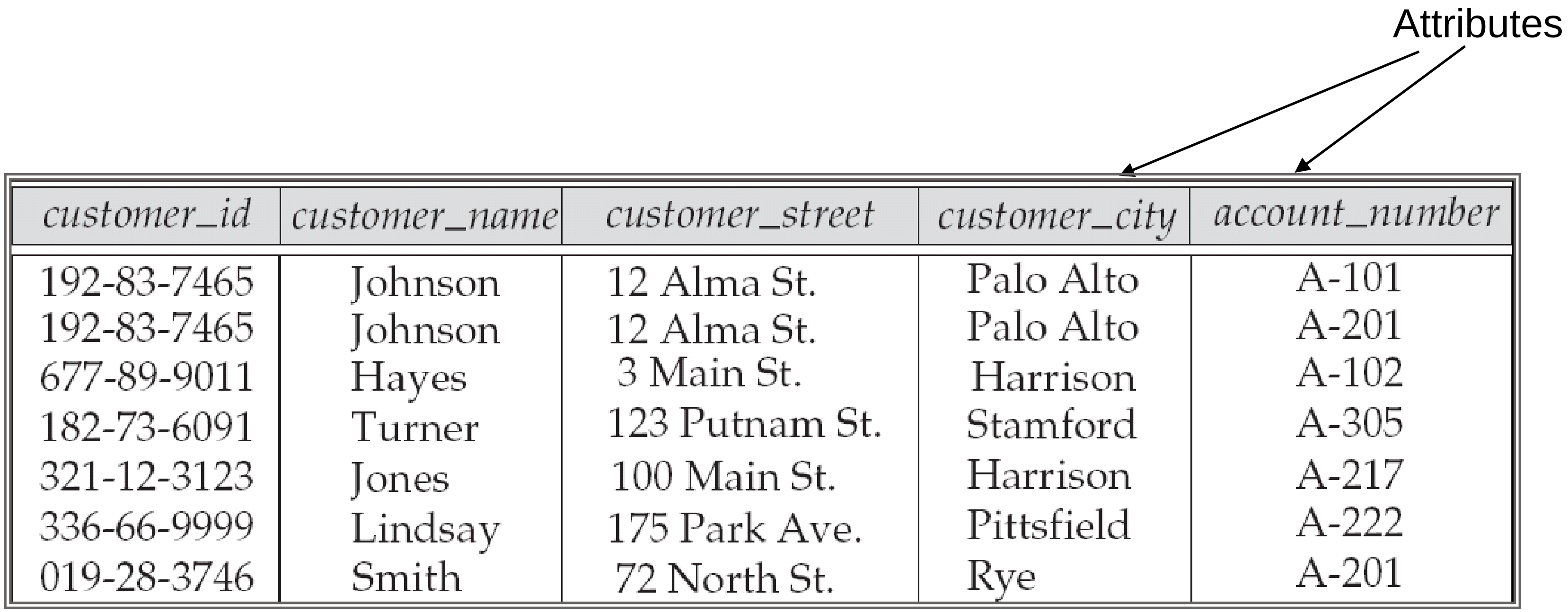
- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semistructured data model (XML)

- Other older models:
  - Network model
  - Hierarchical model
- Why Data Modeling is important?

Cannot build a good system without knowing what data needs to be captured and how it needs to be organized

Relational Model

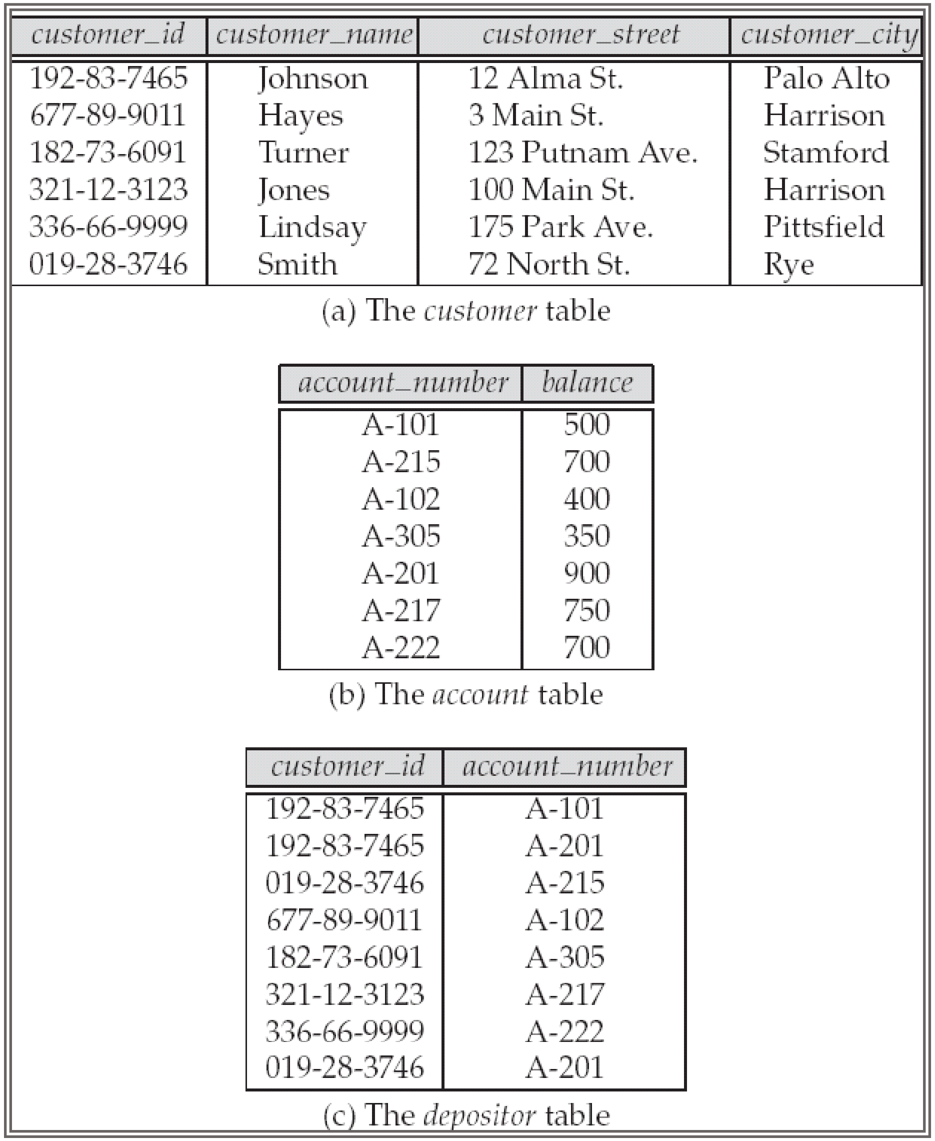
- Example of tabular data in the relational model



customer_id	*customer_name*	*customer_street*	*customer_city*	*account_number*
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-101
192-83-7465	Johnson	12 Alma St.	Palo Alto	A-201
677-89-9011	Hayes	3 Main St.	Harrison	A-102
182-73-6091	Turner	123 Putnam St.	Stamford	A-305
321-12-3123	Jones	100 Main St.	Harrison	A-217
336-66-9999	Lindsay	175 Park Ave.	Pittsfield	A-222
019-28-3746	Smith	72 North St.	Rye	A-201

Attributes

A Sample Relational Database



customer_id	*customer_name*	*customer_street*	*customer_city*
192-83-7465	Johnson	12 Alma St.	Palo Alto
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

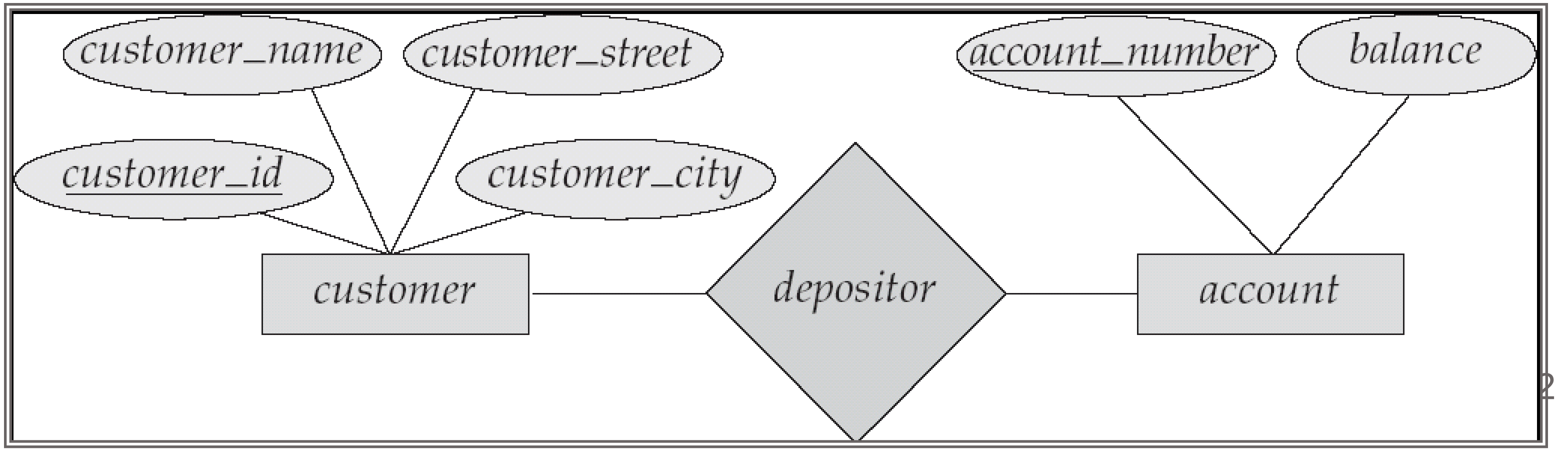
(b) The *account* table

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

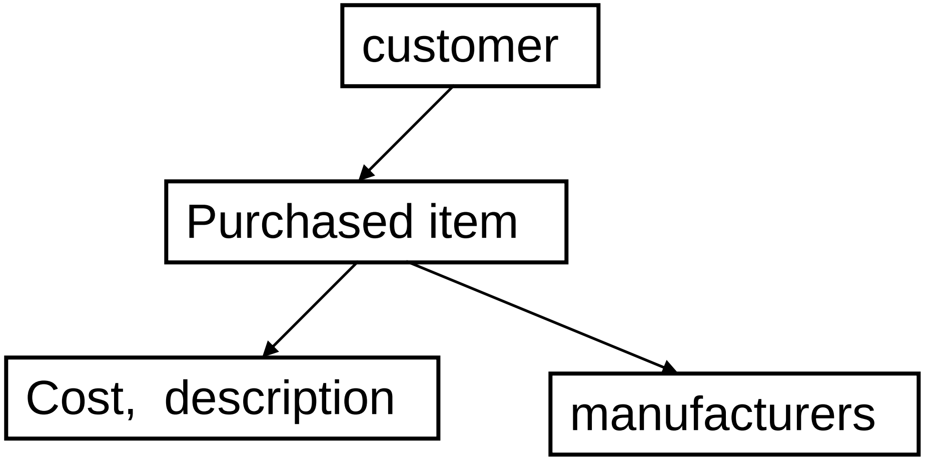
The Entity-Relationship Model

- 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:*



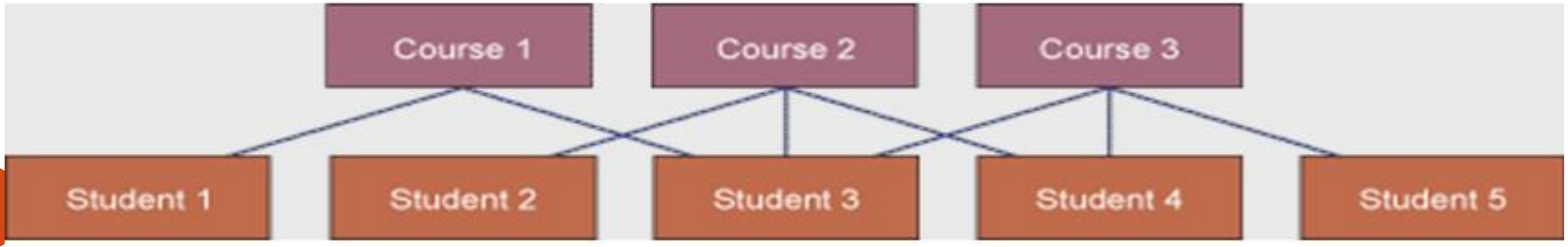
Hierarchical Database Model

- It is a pointer based model
- Organizes data in a tree-like structure
- Stores data in tables and views relationships as links
- Supports one-to-many parent-child relationships, and inflexible due to this



Network DBMS

- Depicts data logically as many-to-many relationships
- Organizes data in tables and views relationships as links
- It is also a pointer based model
- Organizes data in arbitrary graphs



Hierarchical and Network DBMS

Some of the Disadvantages

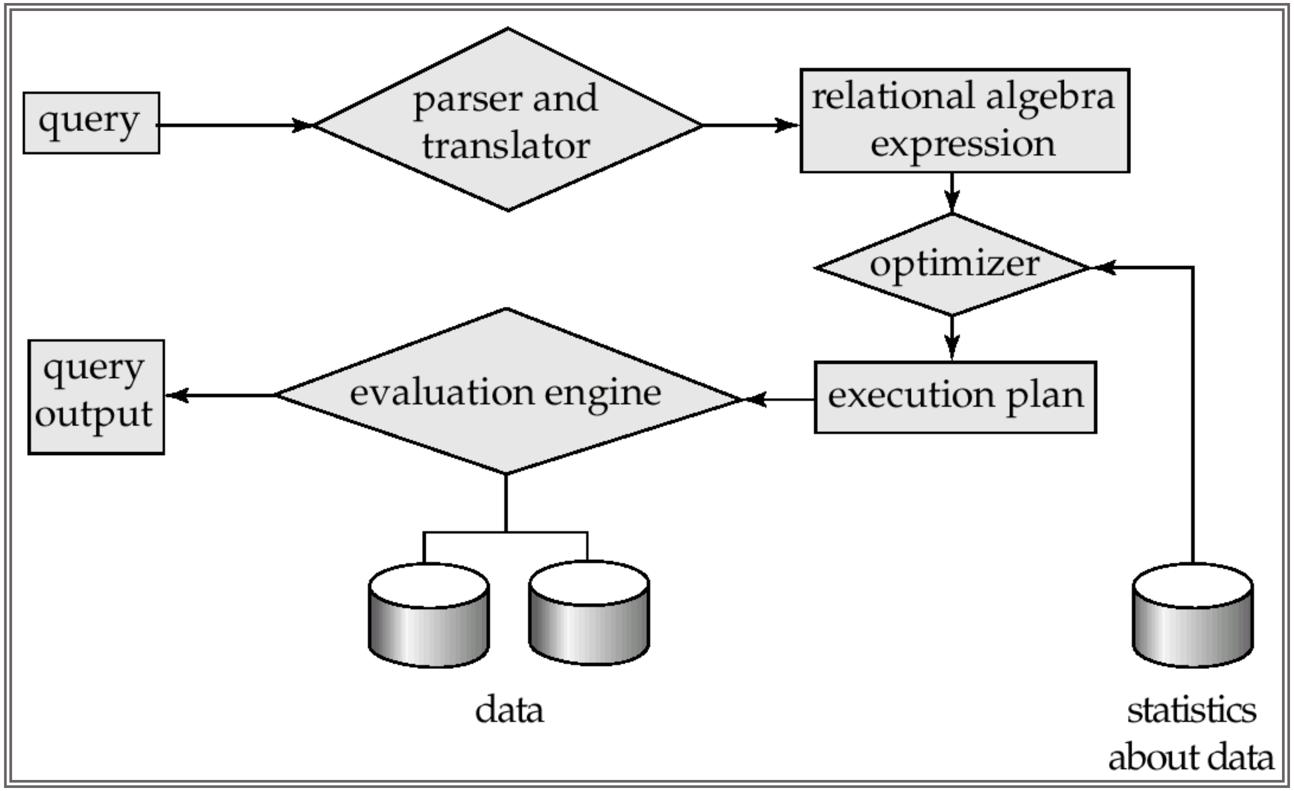
- **Outdated**
- **Complex pointer based organization**
- **Less flexible compared to RDBMS**
- **Lack support for ad-hoc and English language-like queries**

Object-Oriented Databases

- Object-oriented DBMS: Stores data and procedures as objects that can be retrieved and shared automatically
- Object-relational DBMS: Provides capabilities of both object-oriented and relational DBMS

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 Languages

- Data definition language (DDL)
  - Language to manipulate a database schema
  - Permits the specification of implementation details
  - DDL SQL command includes
    - Create - To create a new database, table, etc.
    - Drop – To destroy an existing database, table, view, etc.
    - Alter – modify an existing database object
    - Truncate – irreversibly clear a table.

Database Languages

- Data Dictionary
  - Results of compilation of DDL statements affects some set of tables which are stored in a file called data dictionary.
- Data manipulation language (DML)
  - DML is a language that enables users to access or manipulate data
    - Insertion, deletion, updation, etc.
- DML
  - Procedural [user specifies: what data are needed and how to get them]
  - Non-procedural [user specifies: what data are needed without specifying how to get them]

Database Users

Users are differentiated by the way they expect to interact with

the system

- **Application programmers** – interact with system through DML calls
- **Specialized users** – write specialized database applications that do not fit into the traditional data processing framework
- **Naïve users** – invoke one of the permanent application programs that have been written previously
  - Examples, people accessing database over the web, bank tellers, clerical staff

Database Administrator

- The person having central control of both data and program accessing that data over the system
  - Coordinates all the activities of the database system
- 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
  - Monitoring performance and responding to changes in requirements

When not to use a DBMS

- **When a DBMS may be unnecessary:**
  - If the database and applications are simple, well defined, and not expected to change.
  - If there are real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required.
- **When no DBMS may suffice:**
  - If the database system is not able to handle the complexity of data because of modeling limitations
  - If the database users need special operations not supported by the DBMS.