Database Systems...contd.

- **Database:**

Database is an interrelated data which related to a particular enterprise.

or

Database is an collection of data organized in a particular way.

or

In simple terms we can say database is an interrelated data.

- **DBMS is a collection of interrelated data and set of programs to access those data.**

- **The primary goal of DBMS is to provide a way to store and retrieve database formation that is both convenient and efficient.**

- **The database system must ensure the safety of information stored, despite system crashes or attempts unauthorized access.**

Database Systems...contd.

- The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

- **Defining:**

- Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database.

- The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data.

- **Constructing**

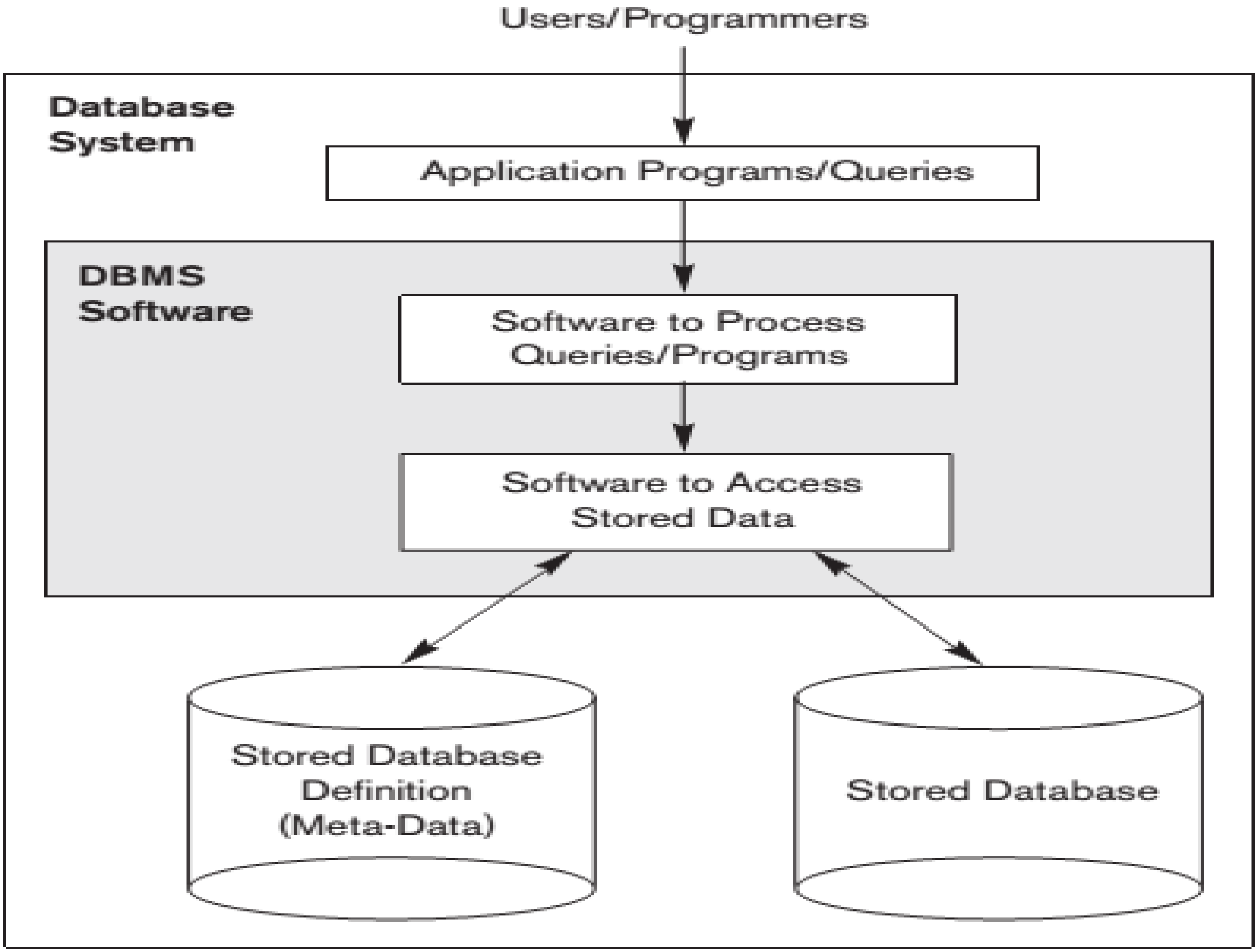
  Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS.

- **Manipulating:**

  Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini-world.

Database Systems…contd.

Database System Environment



Users/Programmers

Database System

Application Programs/Queries

DBMS Software

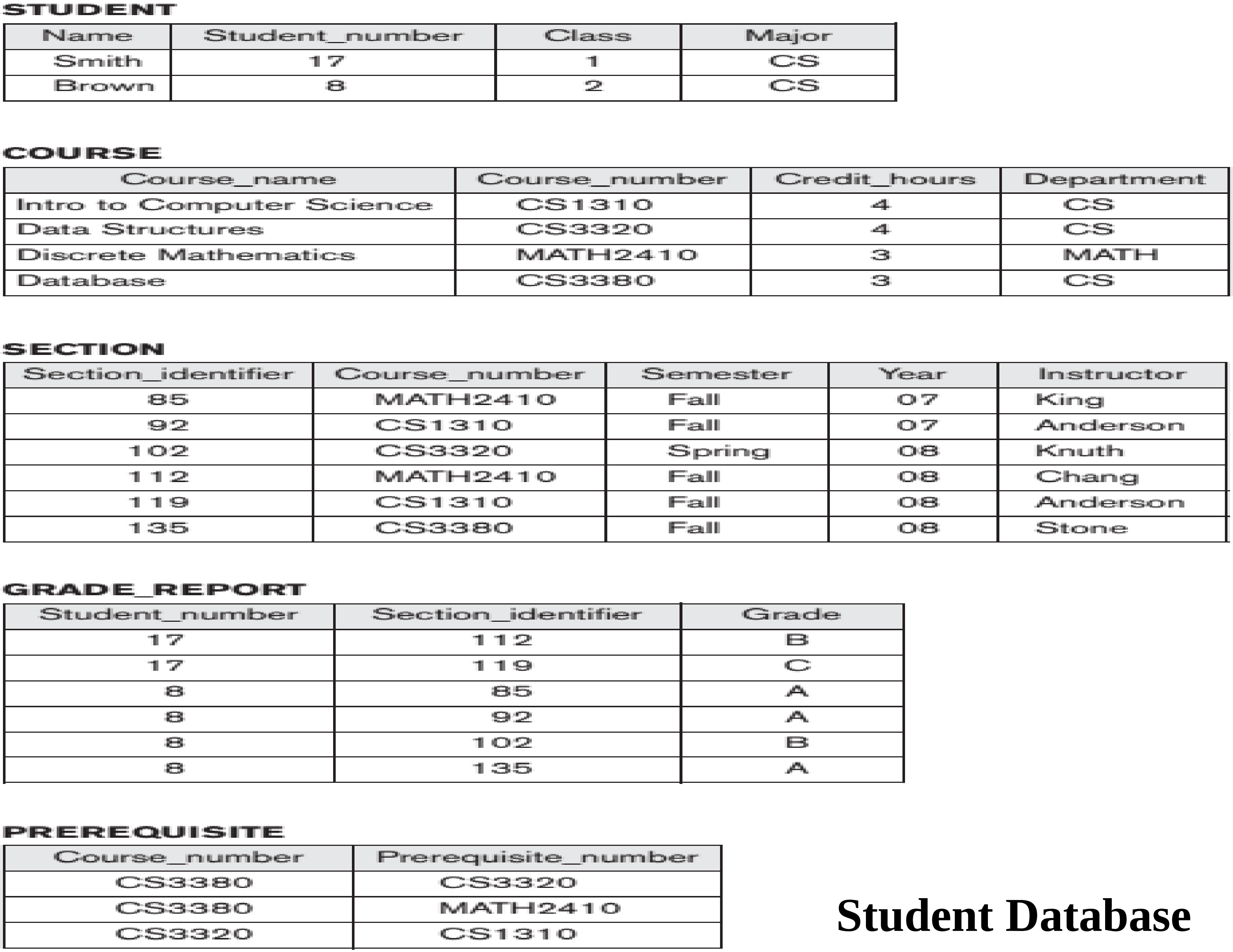
Software to Process Queries/Programs

Software to Access Stored Data

Stored Database Definition (Meta-Data)

Stored Database

Database Systems…contd.



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Student Database

Database Systems...contd.

- In the early days, database applications were built directly on top of file systems

- Drawbacks of using file systems to store data

  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files

  - Difficulty in accessing data
    - Need to write a new program to carry out each new task

  - Data isolation -Multiple files and formats

  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become “buried” in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones

Database Systems...contd.

- **Atomicity of updates**
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example:

    Transfer of funds from one account to another should either complete or not happen at all

- **Concurrent access by multiple users**
  - Concurrent accessed needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
  - Example:

    Two people reading a balance and updating it at the same time

- **Security problems**
  - Hard to provide user access to some, but not all, data
  - Database systems offer solutions to all the above problems

Characteristics of the Database Approach

- **Self-describing nature of a database system**

  Database system contains not only the database itself but also a complete definition or description of the database structure and constraints (Meta Data).

- **Insulation between programs and data, and data abstraction**

  The structure of data files is stored in the DBMS catalog separately from the access programs.

- **Support of multiple views of the data**

  - **A database typically has many users, each of whom may require a different perspective or view of the database.**

  - **A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.**

- **Sharing of data and multiuser transaction processing**

  - **Data for multiple applications (online transaction processing (OLTP)) is to be integrated and maintained in a single database.**

  -The DBMS must include concurrency control software to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.

Database Systems...contd.

The characteristic that allows program-data independence and program-operation independence is called data abstraction.

or

Generally refers to the suppression of details of the data organization and storage, and the highlighting of the essential features for improved understanding of data

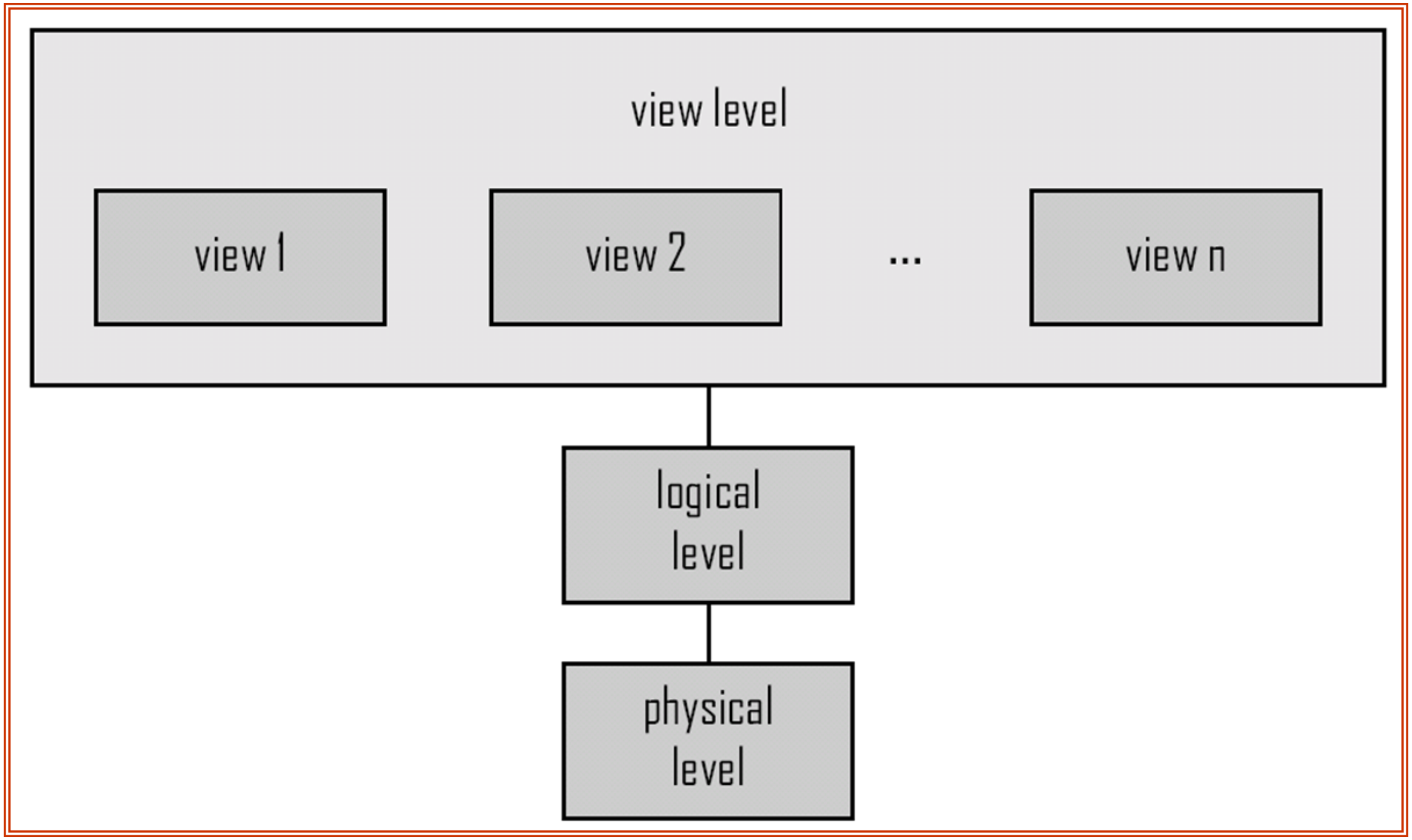
- **Levels of abstraction:**
  - **Physical level:** Describes how a record (e.g., customer) is stored.
  - **Logical level:** Describes data stored in database, and the relationships among the data.

    Example:

    ```
    type customer = record
                        customer_id:string;
                         customer_name:string;
                         customer_street:string;
                       customer_city : string;
         end;
    ```
  - **View level:** Application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

View of Data

An architecture for a database system



Database Systems...contd.

Basic Terminologies

Schema: The logical structure of the database

- **Analogous to type information of a variable in a program**
- **Physical schema: Database design at the physical level**
- **Logical schema: Database design at the logical level**

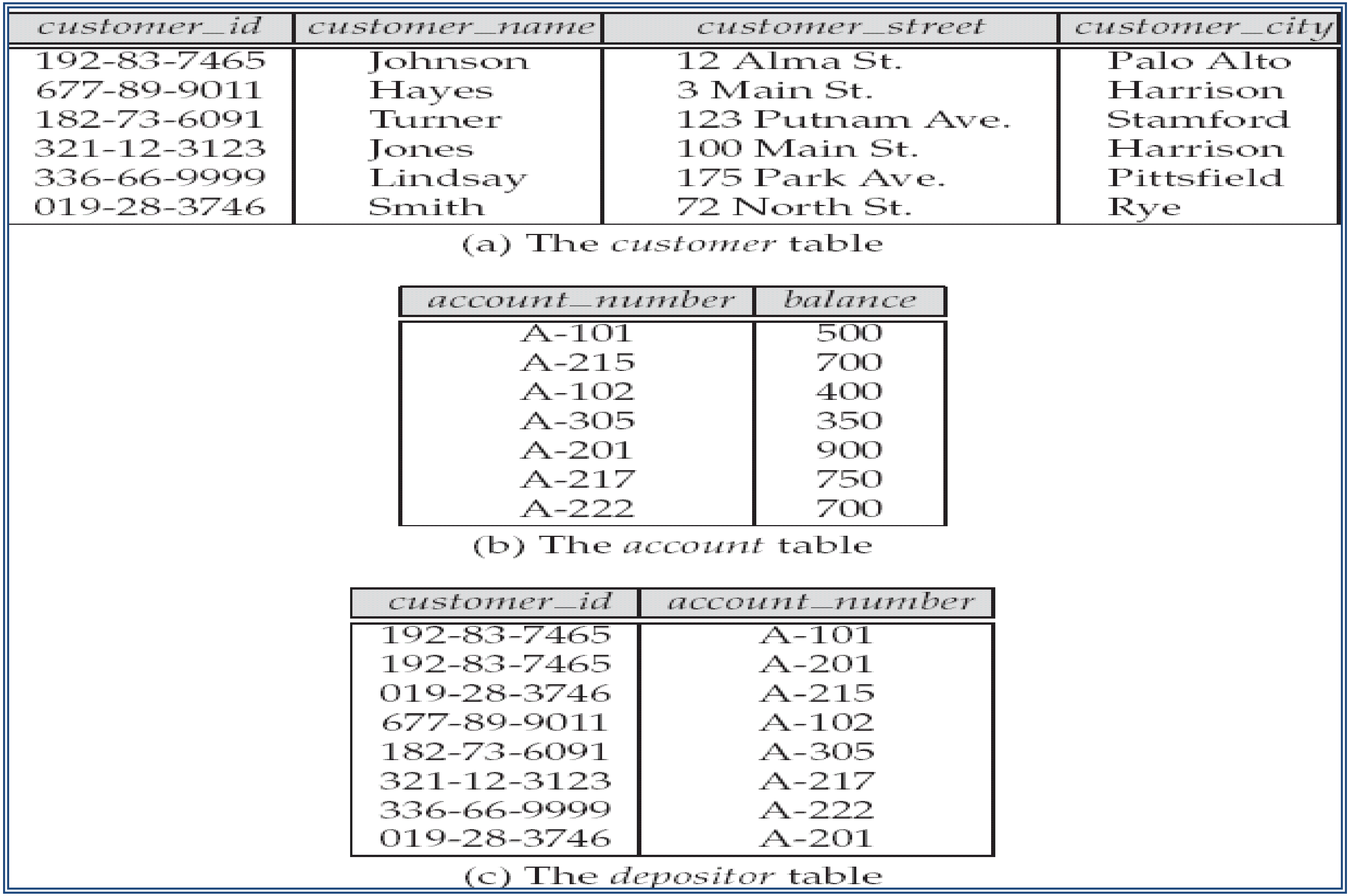
- **Instance: The actual content of the database at a particular point in time**
- **- Analogous to the value of a variable**

A data model

- **A collection of concepts that can be used to describe the structure (data types, relationships, and constraints that apply to the data) of a database**

- **Most data models also include a set of basic operations for specifying retrievals and updates on the database**

Database Systems…contd.



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

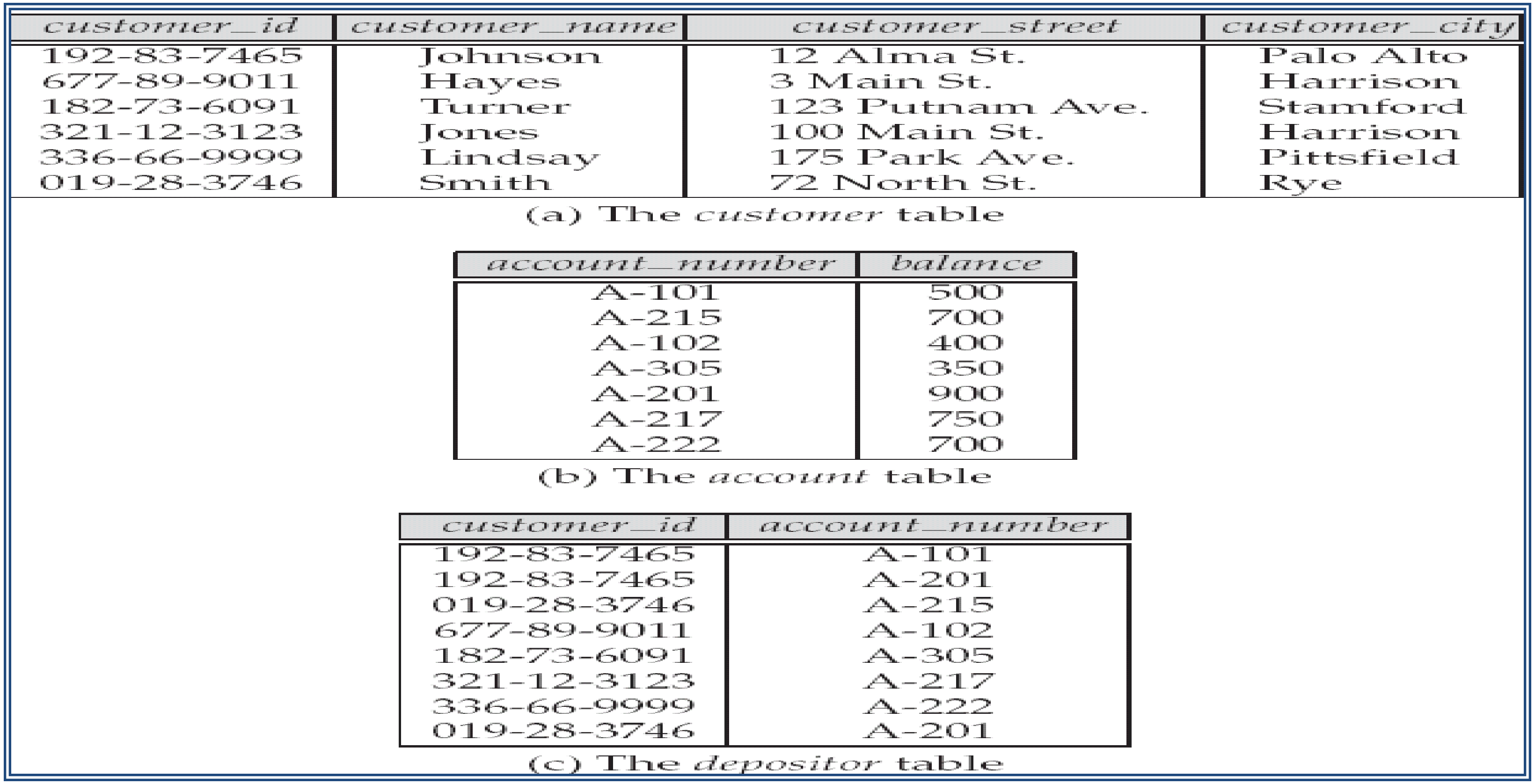
(c) The *depositor* table

Instance of Database

Data Models

Relational Model

- The relational model uses a collection of tables to represent both data & the relationships among those data
- Each table has multiple columns, and column has a unique name
- It is also called as record based models-database is structured in fixed-format records of several types.
- E.g



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

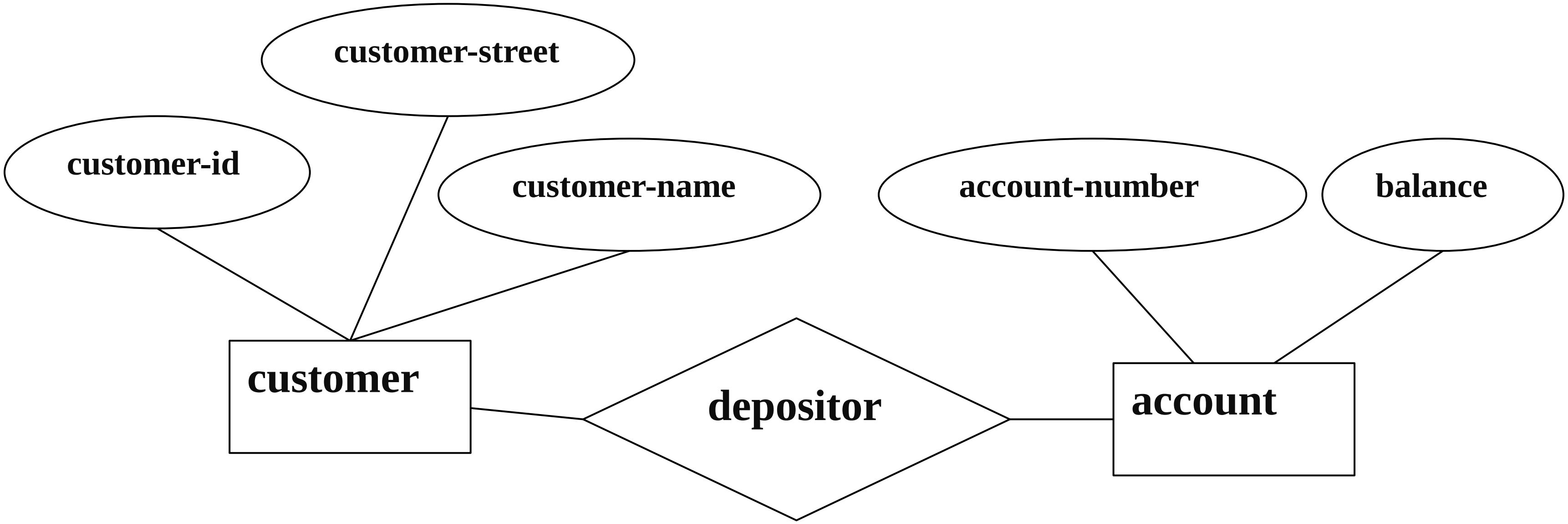
(c) The *depositor* table

The Entity–Relationship Model (ER Model)

- It is based on the perception that real world consists of collection of basic objects called entity and of relationship among those objects

- Entity is an thing or object in the real world and it is distinguishable from other objects

- It is widely used in the database design
- E.g



Database Systems...contd.

The Object-Based Model

- It combines the features of object-oriented data model and relational data model.

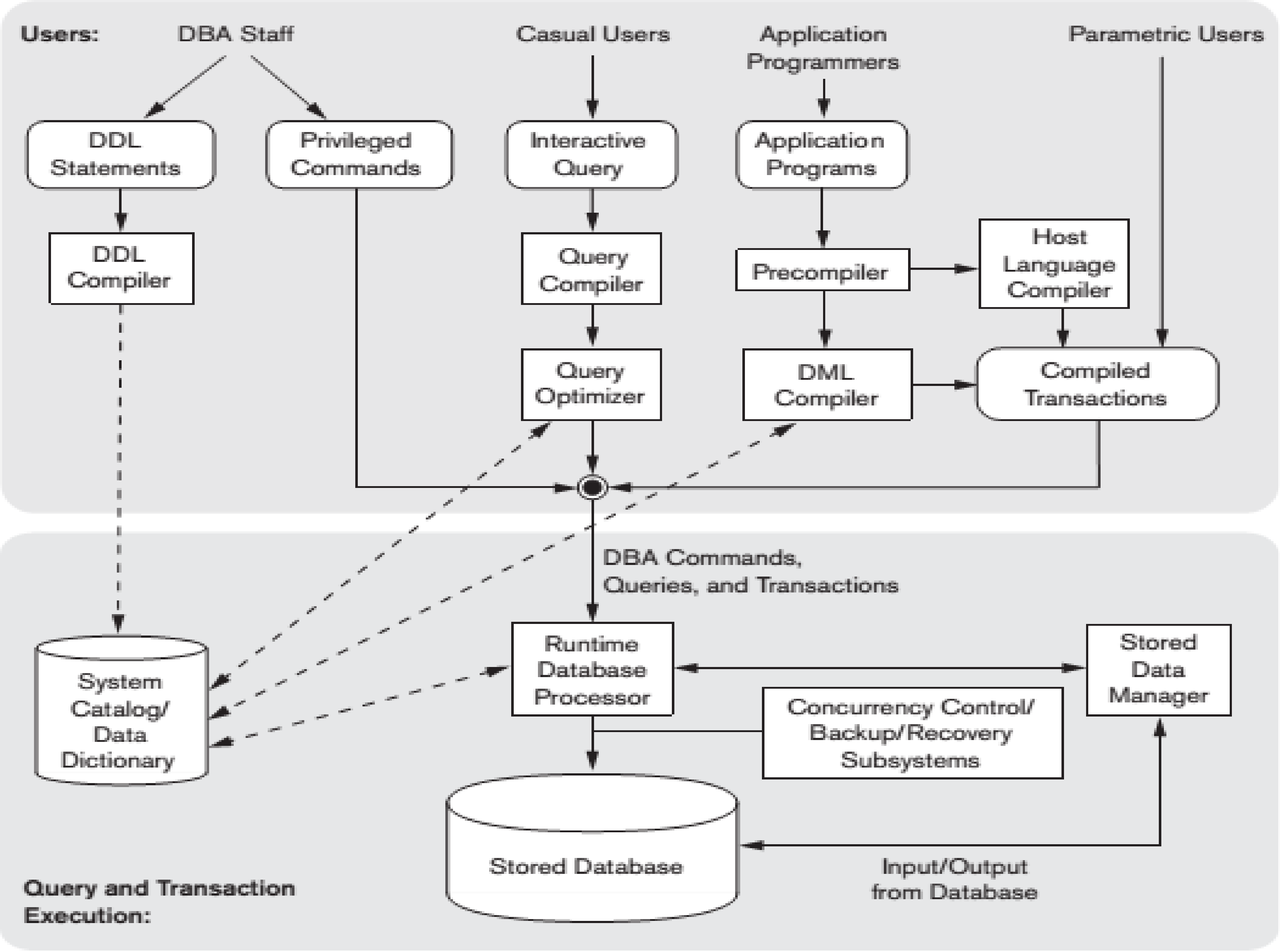
- The object-relational data model extends the relational data model by providing a richer type system including complex data types and object orientation oriented

- The database systems based on the object –relational model provide a convenient migration path for users of relational databases who wish use object-orientated features.

Semistructured Data model

- The semistructured data model permits the specification of data where individual data items of the same type may have different sets of attributes

-The extensible markup Language (XML) is widely used to represent semistructured data



Component Models of a DBMS and their interfaces

Different Actors of Database System

Database Administrators:

- Schema definition: Executing Data Definition Language(DDL)
- Storage Structures and access method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance:
    - Periodically backing the database to prevent from loss of data
    - Monitoring jobs running on the database and ensuring performance
    - Ensuring free space availability for normal operations

Different Actors of Database System

Database Designers:

- Database are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data.

- These tasks are mostly undertaken before the database is actually implemented and populated with data.

- It is the responsibility of database designers to communicate with all prospective database users in order to understand their requirements and to create a design that meets these requirements

Database Systems...contd.

End User: End users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use.

- **Casual end users:** Occasionally access the database, but they may need different information each time.

  - They use a sophisticated database query language to specify their requests and are typically middle- or high-level managers or other occasional browsers.

- **Naïve or parametric end users:**

  - Unsophisticated user who interact with the system by invoking one of the application that have been written previously. Make up a sizable portion of database end users.

  - Their main job function revolves around constantly querying and updating the database, using standard types of queries and updates—called canned transactions—that have been carefully programmed and tested.

  - The tasks that such users perform are varied:
    - Bank tellers check account balances and post withdrawals and deposits.
    - Reservation agents for airlines, hotels, and car rental companies check a availability for a given request and make reservations

Database Systems...contd.

Sophisticated end users: Include engineers, scientists, business analysts, and others who thoroughly familiarize themselves with the facilities of the DBMS in order to implement their own applications to meet their complex requirements.

Standalone users: Maintain personal databases by using ready-made program packages that provide easy-to-use menu-based or graphics-based interfaces.

- **An example is the user of a tax package that stores a variety of personal financial data for tax purposes.**

▪System Analysts and Application Programmer:

- **System analysts determine the requirements of end users, especially naive and parametric end users.**
- **They develop specifications for standard canned transactions that meet these requirements.**
- **Canned Transaction is the process of constantly querying and updating the database using standard types of queries and updates**
- **Application programmers implement these specifications as programs; then they test, debug, document, and maintain these canned transactions.**

Database Systems...contd.

Data Manipulation Language:

- Language for accessing and manipulating the data organized by the appropriate data model. DML also known as query language

-The types of access are:

- Retrieval of information stored in the database
- Insertion of new information into the database
- Deletion of information from the database
- Modification of information stored in the database

Two classes of languages

- Procedural DML– User specifies what data is required and how to get those data
- Declarative (nonprocedural)DML–User specifies what data is required without specifying how to get those data

Database Systems...contd.

- **General DML statements are:**

SELECT - retrieve data from the a database

INSERT - insert data into a table

UPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain

MERGE - UPSERT operation (insert or update)

CALL - call a PL/SQL or Java subprogram

EXPLAIN PLAN - explain access path to data

LOCK TABLE - control concurrency

Database Systems...contd.

- **Data Definition Language**
- **Specification notation for defining the database schema.**

- **DDL also used to specify:**
  - **Domain constraints: Domain of possible value must be associated with every attribute**
  - **Referential integrity: The value that appears in one relation for a given set of attributes should also appears in the another relation for a certain set of attributes**
  - **Assertion: Assertion is any condition that the database must always satisfy.**
  - **Authorization: Read authorization, write authorization, update authorization, delete authorization.**
  - **Example:**

```
create table account (
                         account_number   char(10),
                          branch_name      char(10),
                          balance          integer)
```

- **Referential integrity – A foreign key must have a matching primary key or it must be null**

-This constraint is specified between two tables (parent and child); it maintains the correspondence between rows in these tables. It means the reference from a row in one table to other table must be valid.

- **Examples**

  - In the Customer/Order database: Customer(**custid**, custname)

  Order(**orderID**, custid, OrderDate)

  - To ensure that there are no orphan records, we need to enforce referential integrity.

  - An orphan record is one whose foreign key value is not found in the corresponding entity – the entity where the PK is located.

  - Recall that a typical join is between a PK and FK.

  - **The referential integrity constraint states that the CustID in the Order table must match a valid CustiD in the Customer table. Most relational databases have declarative referential integrity.**

- When the tables are created the referential integrity constraints are set up.

Database Management System Internals

- Storage management
- Query processing
- Transaction processing

Storage Management

- **The main components of storage manager are:**
  - **Authorization and Integrity Manager: Checks integrity constraints & authority of user**
  - **Transaction Manager: Ensures database remains in constant state despite system failure**
  - **File Manager: Manages allocation of free space**
  - **Buffer Manager: Responsible for fetching data from storage into main memory**

- **Storage manager implements several data structures as a part of physical implementation**
  - **Data Files: Stores database itself**
  - **Data Dictionary: Stores metadata**
  - **Indices: Which provides fast access to data items**

Query Processor: **Query processing components includes:**

- **DDL Interpreter:** **Which interprets DDL statements and records the definition in the data dictionary**

- **DML Compiler:** **Which translates DML statements in a query language into a evaluation plan consisting of low-level instructions that query evaluation engine understand.**

  -DML compiler can also perform query optimization

- **Query Evaluation Engine:** **Which executes low-level instructions generated by the DML compiler**

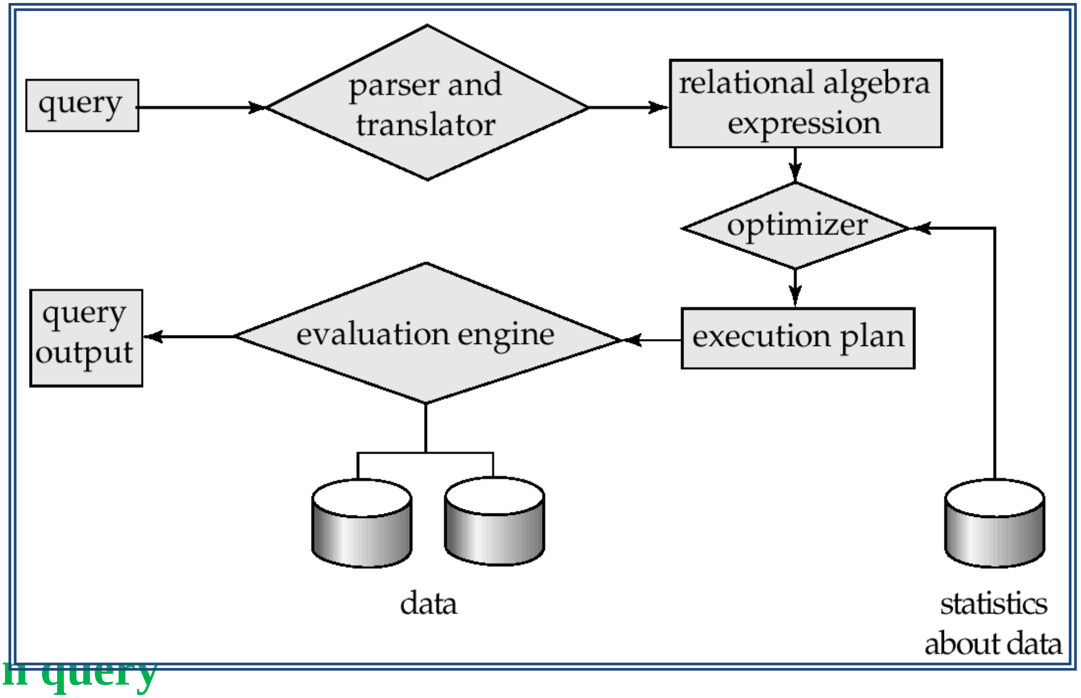
Query processing

It includes following steps:

1. Storage Parsing and Translation

2. Optimization

3. Evaluation



- Alternative ways of evaluating a given query
  - Equivalent expressions
  - Different algorithms for each operation
- Cost difference between a good and a bad way of evaluating a query can be enormous
- Need to estimate the cost of operations
  - Depends critically on statistical information about relations which the database must maintain
  - Need to estimate statistics for intermediate results to compute cost of complex expressions

Database Systems...contd.

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: 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.

DBMS Interfaces

- Menu-Based Interfaces for web Clients or Browsing
- Forms-Based Interfaces for Naïve users
- Graphical User Interface
- Natural Language Interface
- Interface for DBA
- Interface for Parametric User

Database Systems…contd.

- **Main inhibitors (costs) of using a DBMS:**
  - **High initial investment and possible need for additional hardware.**
  - **Overhead for providing generality, security, concurrency control, recovery, and integrity functions.**

- **When a DBMS may be unnecessary:**
  - **If the database and applications are simple, well defined, and not expected to change.**
  - **If there are stringent 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.**