CMPE343 Introduction to Database Management Systems and Programming I

Prof Dr Melike Şah Direkoğlu
msah@ciu.edu.tr

SLIDE CREDIT - YASEMİN BAY AYZEREN,
HTTPS://WWW.LEARNCOMPUTERSCIENCEONLINE.COM/RELATIONAL-DATABASE/

Database Management System (DMBS)

DBMS is software that provides an interface for users and applications to interact with a **database**.

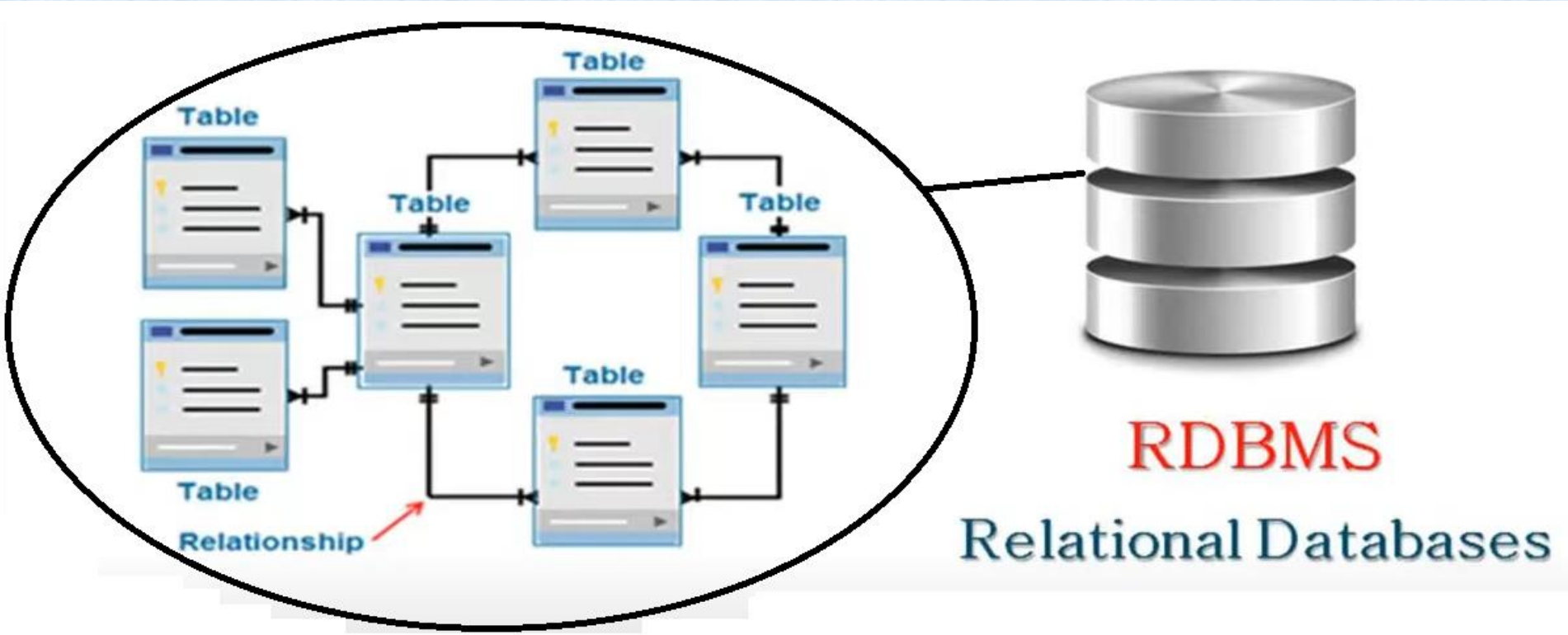
A database is a **structured collection of data** that is organized and stored in a way that allows for efficient retrieval, management, and manipulation of that data.

The role of DBMS is to efficiently store, retrieve, delete, add and modify data.

- Collection of interrelated data (in tables),
- Set of programs to access the data,
- DMBS contains information about a particular enterprise/domain,
- DBMS provides an environment that it both convenient and efficient to use.

DBMS Structure

- DBMS consists of a set of tables and relationships that is called **relational database model**!



What is Relational Database Model?

- A collection of tools for describing
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- **Relational model;** The relational model describes data in a table as attributes and tuples (or records). We call such DBMS as **Relational DBMS (RDBMS).**
- **Entity-Relationship data model (mainly for database design)**; Entity-Relationship Model shows mapping cardinalities between different tables (entities) Will be discussed it later on.

Relational Model (Table specific information)

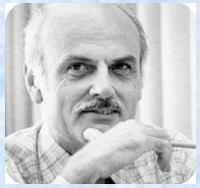
- All the data is stored in various tables.
- Example of tabular data in the relational model

Columns (attributes)

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Rows (records/tuples). Each row has a primary key

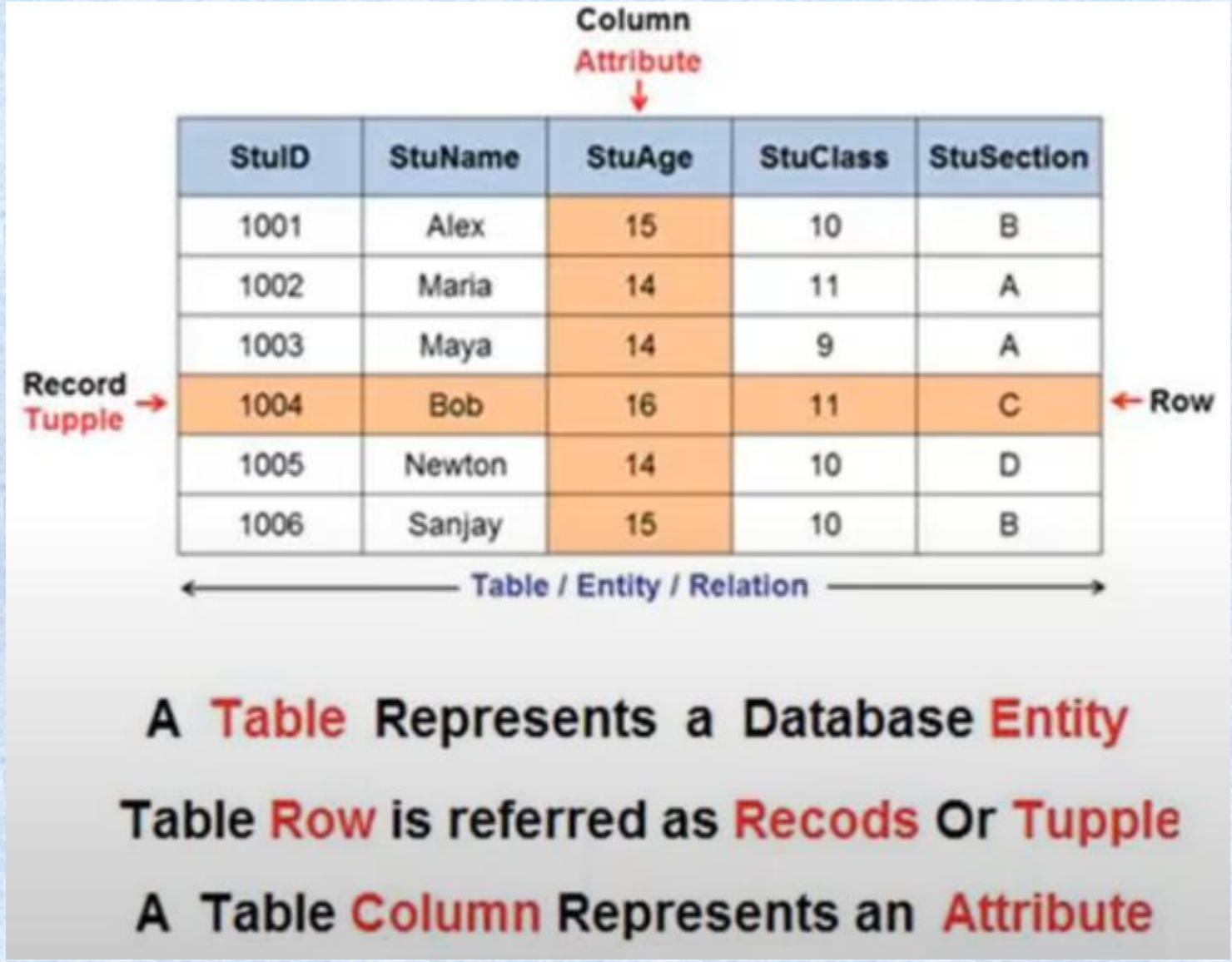


Ted Codd

(while working for IBM, invented the relational model for database management**)**

Turing Award 1981

Relational Model



StuID	StuName	StuAge	StuClass	StuSection
1001	Alex	15	10	B
1002	Maria	14	11	A
1003	Maya	14	9	A
1004	Bob	16	11	C
1005	Newton	14	10	D
1006	Sanjay	15	10	B

A Table Represents a Database Entity

Table Row is referred as Recods Or Tupple

A Table Column Represents an Attribute

A Sample Relational Database

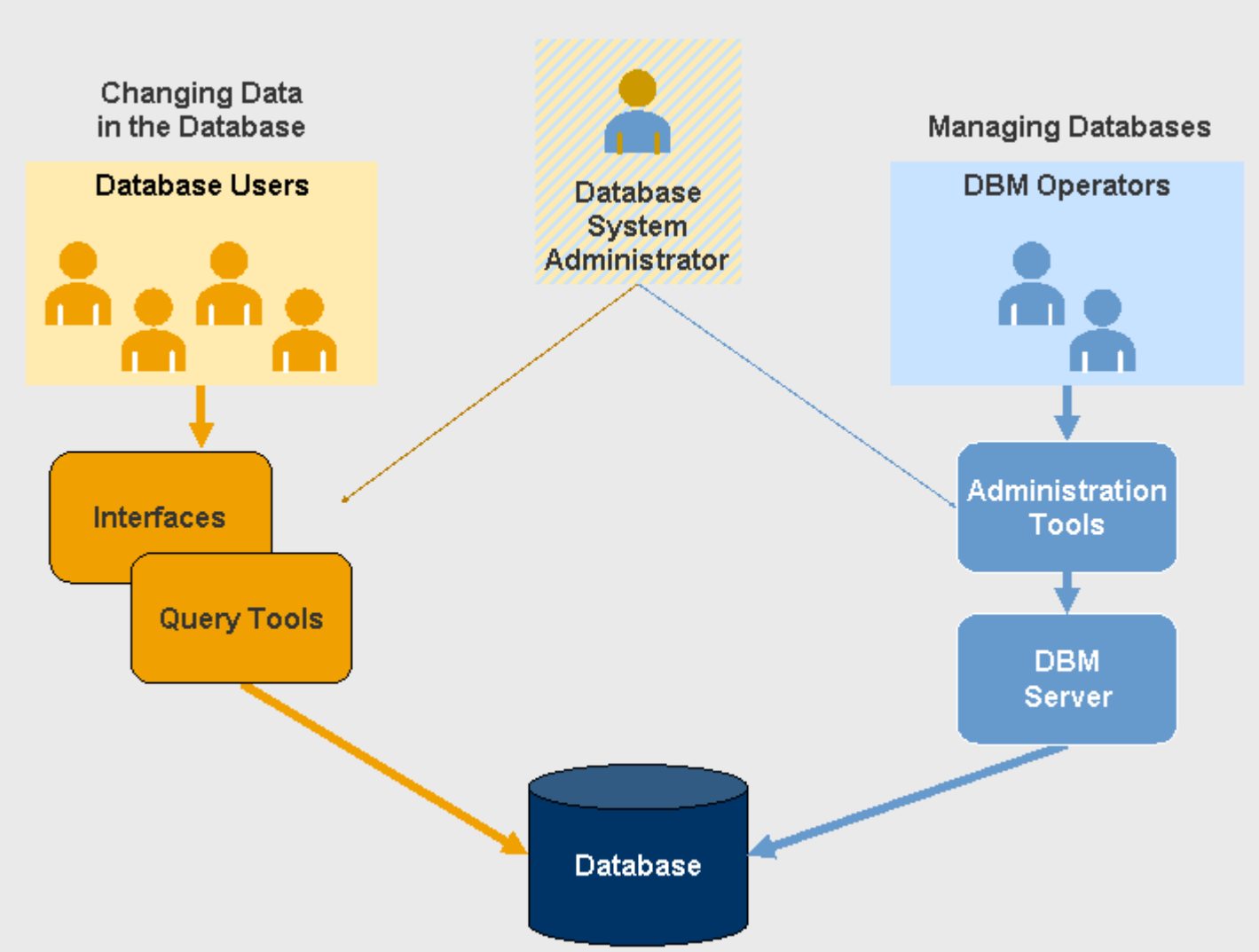
ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

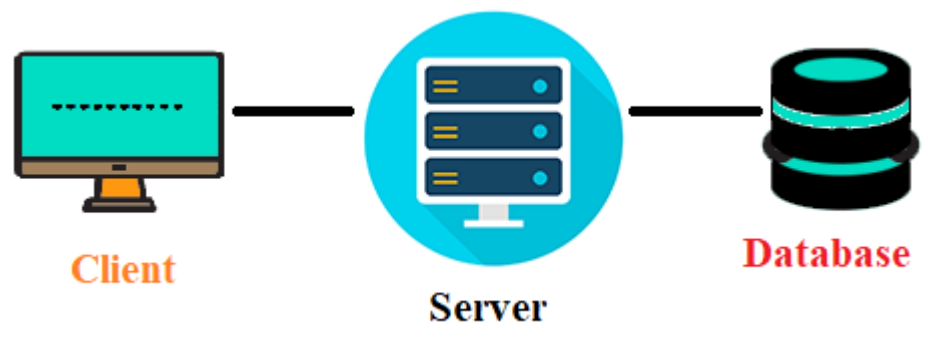
(b) The *department* table

User interactions between the database and the end users



Three Tier Architecture

- Generally three tier architecture is adopted by many applications when accessing to the database.



Presentation Tier

Middle Tier

Data Tier

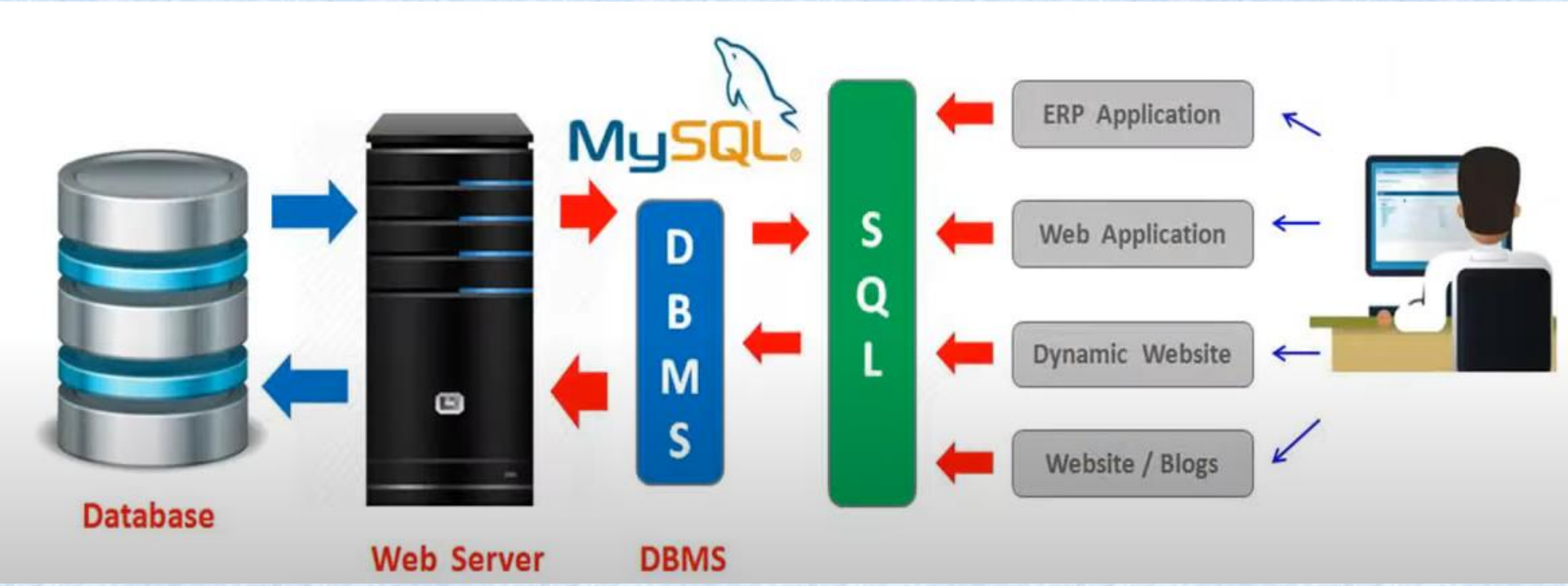
The Three Layers

- Presentation tier
  - Primary interface to the user
  - Needs to adapt different displays (PC, cell, tablet, etc)

- Middle tier (Server)
  - Implements business logic (implements complex actions, maintains state between different steps of workflow)
  - Access different data management systems

- Data management tier
  - One or more standard database management system

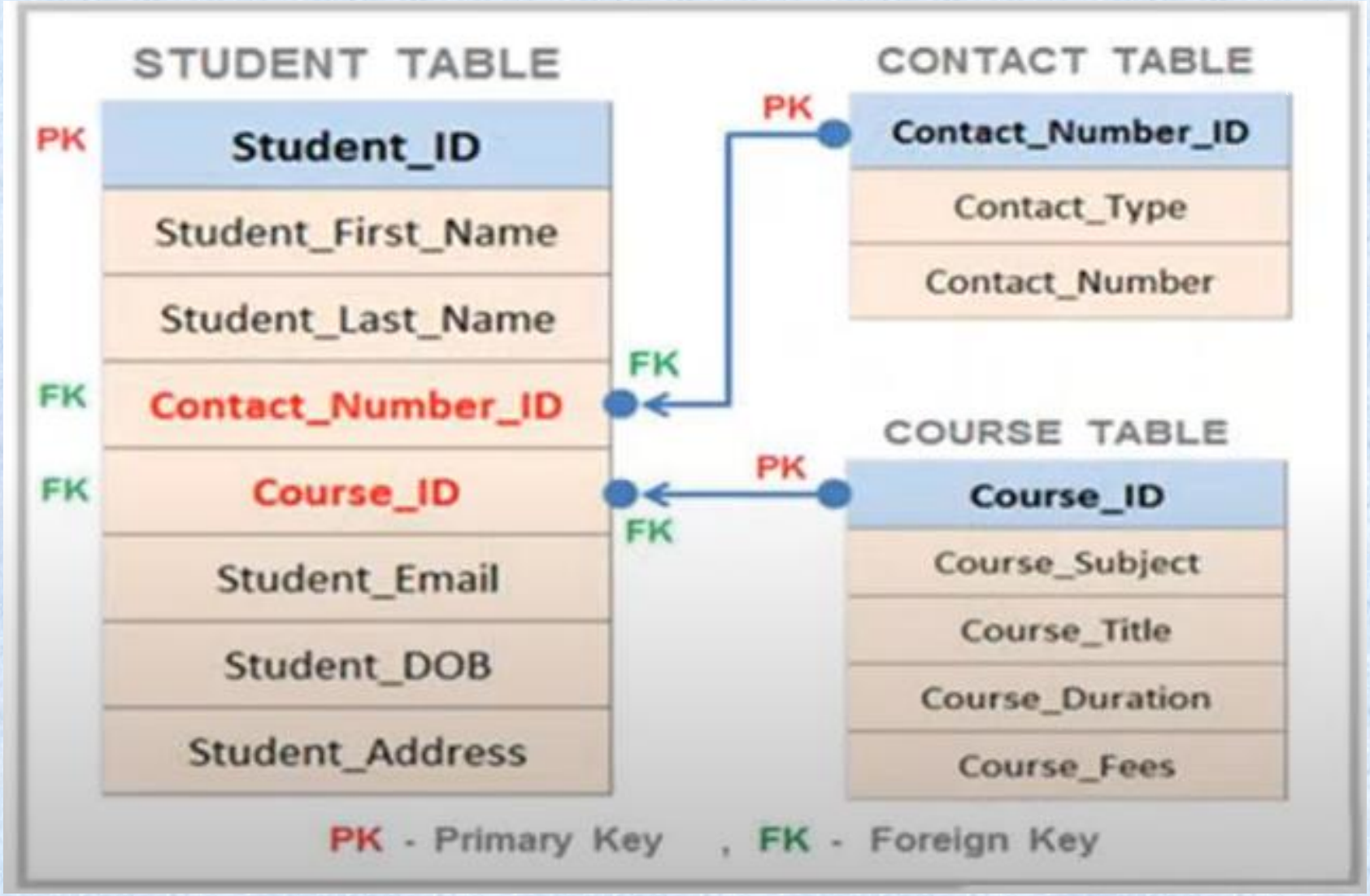
A Three Tier Architecture Example



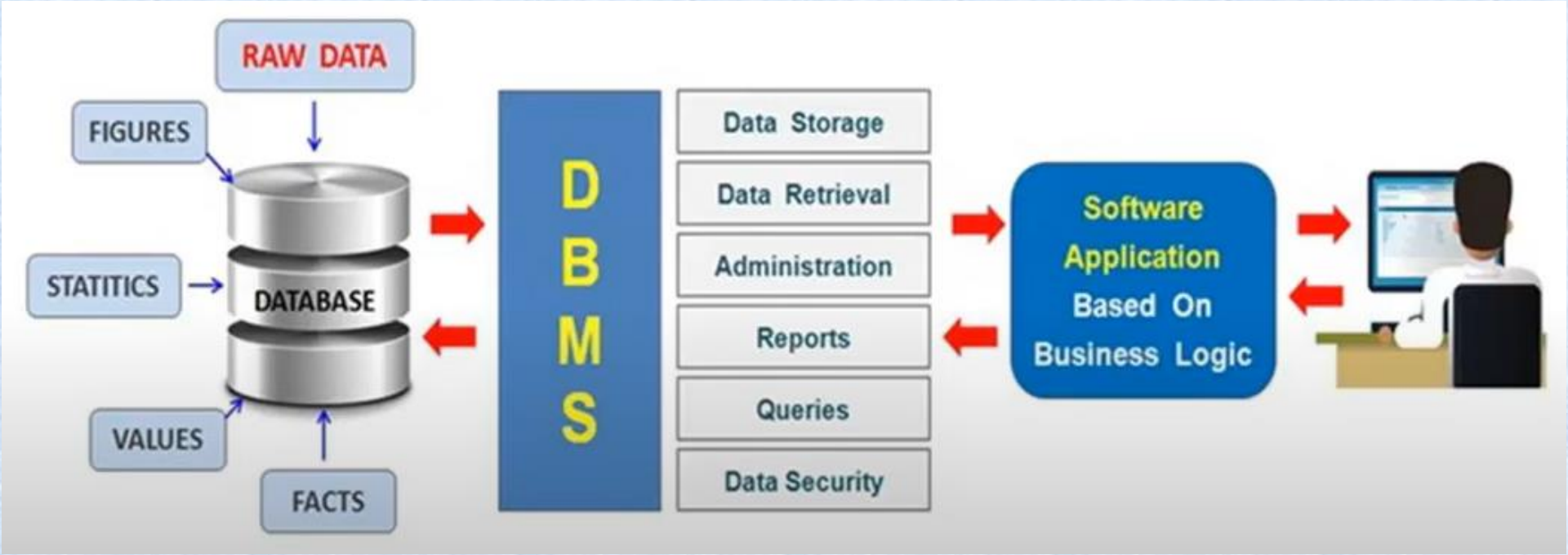
Common Database Applications

- **Banking:** customer information, accounts, credits, debits, …
- **Airlines:** reservations, schedules,…
- **Universities:** student information, registration, courses,…
- **Manufacturing:** production, inventory, customer, sales,…
- **Google ???**

University DBMS Example



What Can We do With a DBMS?



Purpose of Database Systems

DBMS were developed to handle the following difficulties of typical file-processing systems supported by conventional operating systems:

- Data redundancy and inconsistency
- Difficulty in accessing data – time, convenience
- Data isolation – multiple files and formats
- Integrity problems – forms with compulsory areas
- Atomicity - failures during updates
- Concurrent access by multiple users
- Security problems

Advantages of a DBMS

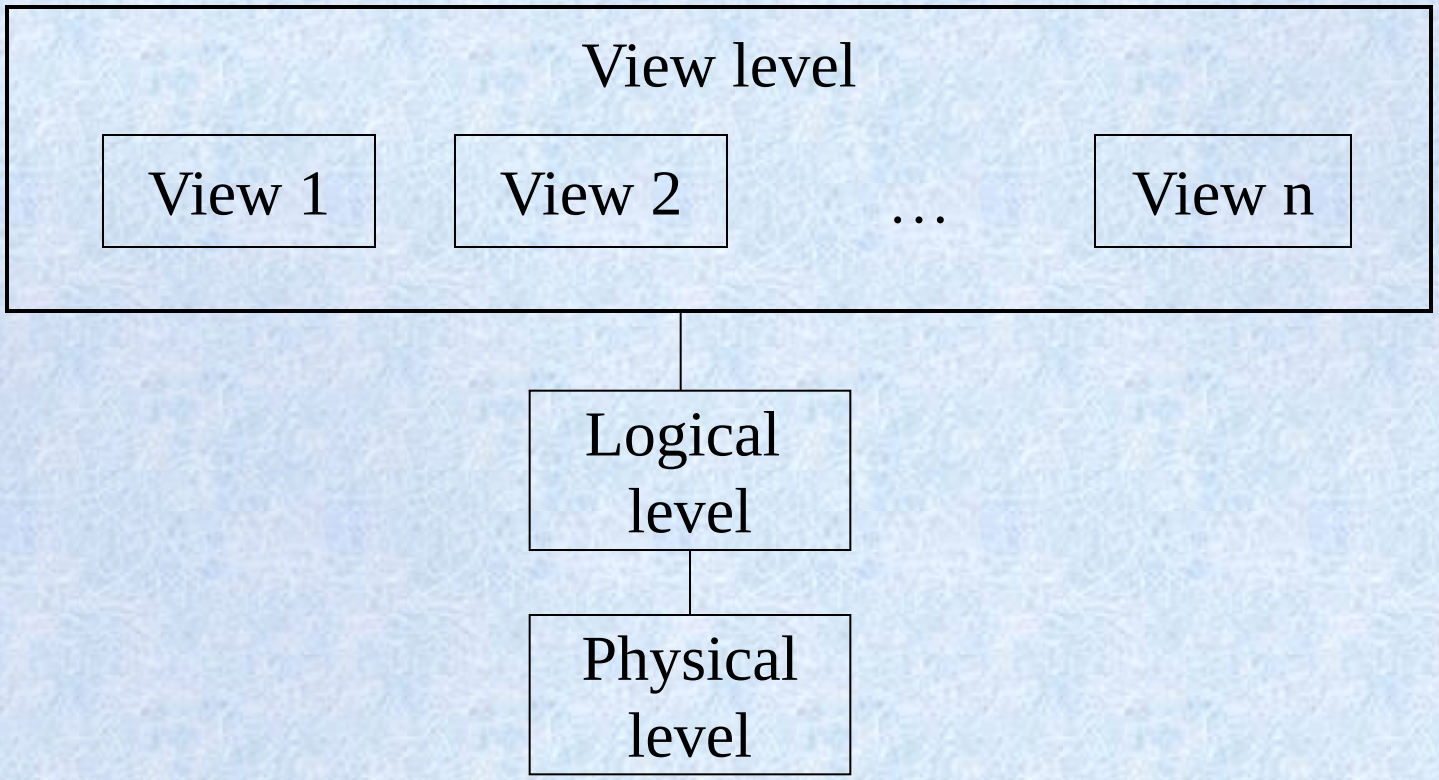
- Redundancy is controlled
- Unauthorized access is restricted
- Providing multiple user interfaces
- Enforcing integrity constraints
- Providing backup and recovery
- Reliability

Popular DBMSs

DBMSs is a software or group of programs designed to perform database operations like creating a new database, administering, managing the database to perform user specified database operations.

View of Data in a DBMS

An architecture for a database system



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.

```
type customer = record
        name: string;
        street: string;
        city: integer;
    end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g. salary) for security purposes.

Data Models

- A collection of tools for describing:
  - Data
  - Data relationships
  - Data semantics
  - Data constraints
- Object-based logical models
  - Entity-relationship model
  - Object-oriented model
  - Semantic model
  - Functional model
- Record-based logical models
  - Relational model (e.g., SQL/DS, DB2)
  - Network model
  - Hierarchical model (e.g., IMS)

Relational Data Model

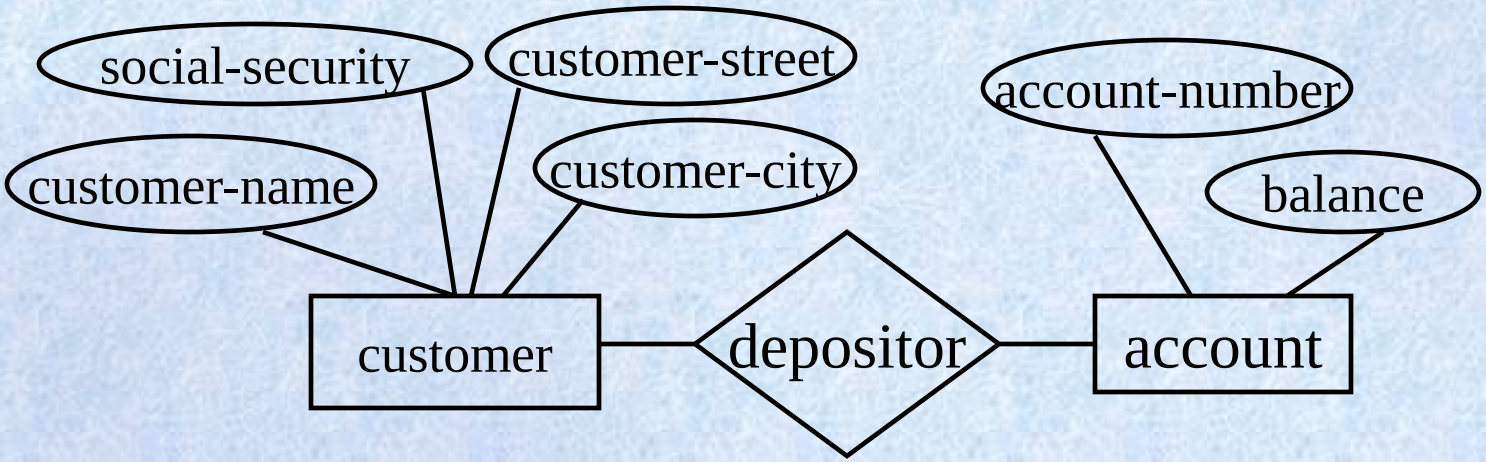
More Examples of tabular data in the relational model:

name	ssn	street	city	account-number
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-215
Johnson	192-83-7465	Alma	Palo Alto	A-201
Jones	321-12-3123	Main	Harrison	A-217
Smith	019-28-3746	North	Rye	A-201

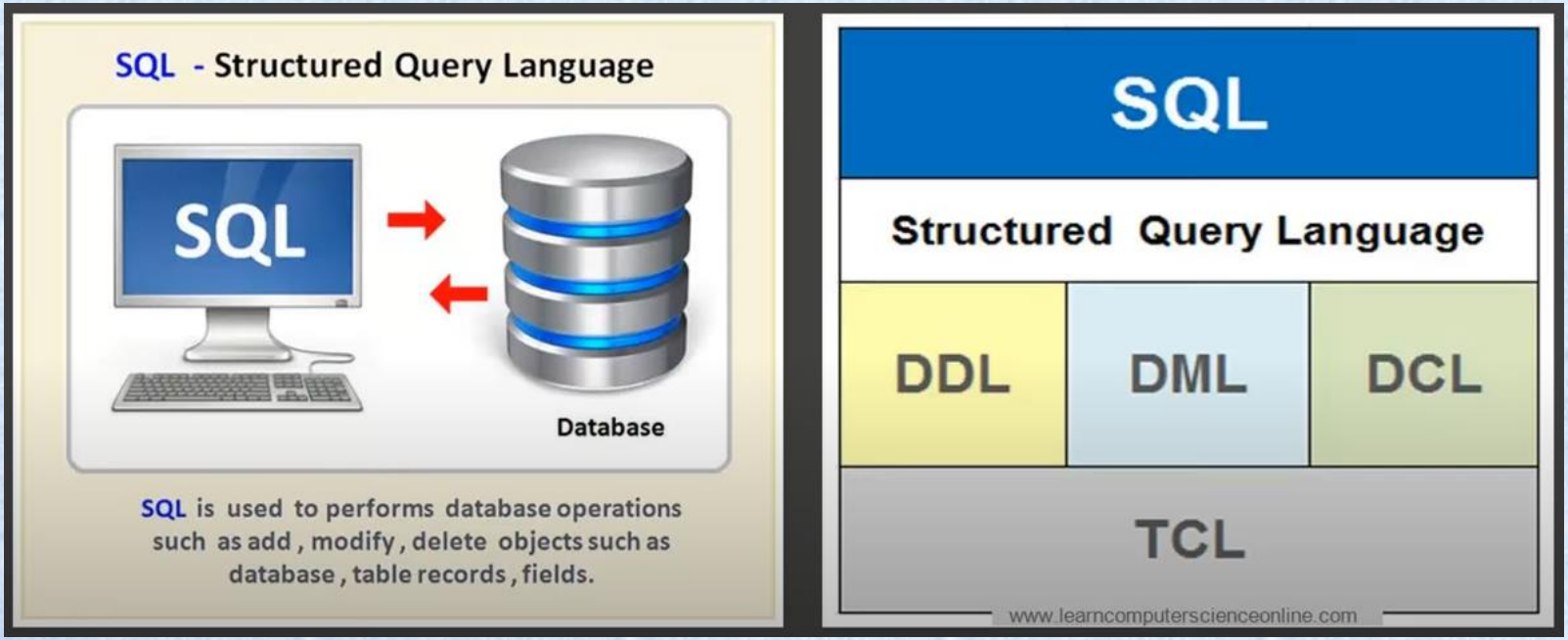
account-number	balance
A-101	500
A-201	900
A-215	700
A-217	750

Entity-Relationship Model

Entity-relationship model describes **how data is stored** in different tables in a relational data model



Structured Query Language (SQL)



DDL – Data Definition Language

DQL – Data Query Language

DML – Data Manipulation Language

DCL – Data Control Language

TCL – Transaction Control Language

Data Definition Language (DDL)

- Used for defining the database schema
- DDL compiler generates a set of tables stored in a data dictionary
- Data dictionary contains ***metadata*** (data about data)
- Ex. **Create table** department(

  department_no number,

  name varchar2(20),

  building varchar2(35),

  primary key (department_no))

Data Query Language (DQL)

- Used for querying the database
- Queries can range from simple select queries to complex nested queries with join operations.
- EX: **select** *ID, name, salary*
  from *instructor*

Data Manipulation Language (DML)

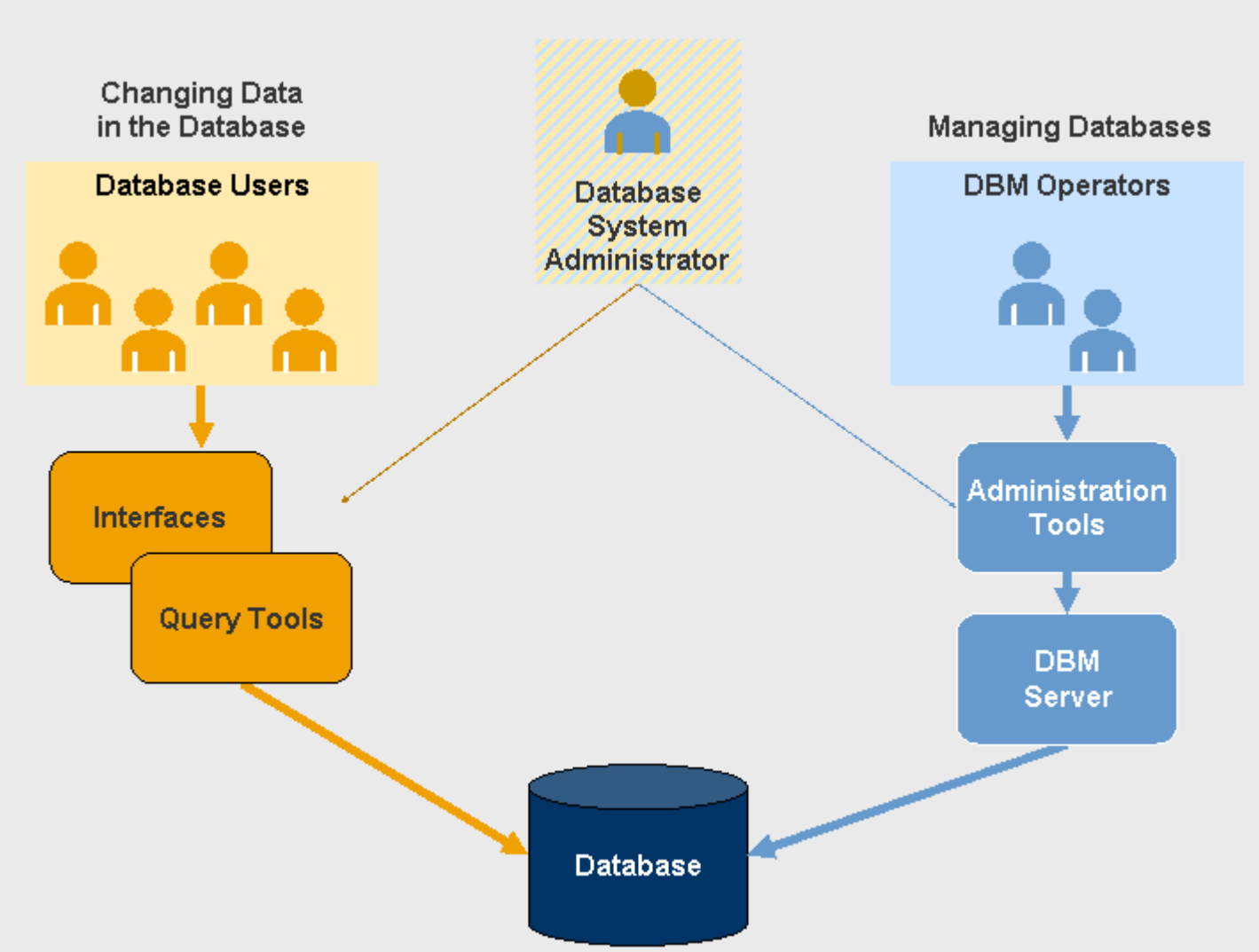
- Language for accessing and manipulating the data organized by the appropriate data model

Ex: **INSERT INTO** clients(first_name,last_name, email, company_name, phone)

VALUES('Sherly','Snider','christene.snider@abc.com', 'ABC Inc', '408-875-6076');

Manipulation → insert, delete, update

Back to User interactions with database and the end users



Database Administrator

- Coordinates all the activities of the database system; the **Database Administrator (DBA)** has a good understanding of the enterprise's information resources and needs:
- 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

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
- **Sophisticated users**: form requests in a database query language.
- **Naive users**: invoke one of the permanent application programs that have been written previously