# Lecture2:

## Ch2. Database System **Concepts** and **Architecture**

## part1

Dr. Alaa eldin Abdallah Yassin

# Outline*

- Database **Users**
- Data Models, Schemas, and Instances.
- Three-Schema Architecture and Data Independence
- Database Languages and Interfaces
- Introduction to SQL

# Database **Users**

# Database **Users**

Users may be divided into

1. Those who actually **use and control the database** content, and those who design, develop and maintain database applications (called "Actors on the Scene").
2. Those who **work to maintain the database** system environment but who are not actively interested in the database itself (called "Workers Behind the Scene").

# Database **Users** … Administrator

**Actors on the scene** ☺

- Database **administrators**: **DBA**
  - Responsible for :
    - authorizing access to the database,
    - for coordinating and monitoring its use,
    - acquiring software and hardware resources as needed.
    - security and poor response time and monitoring efficiency of operations.

# Database **Users** …
## Designer

**Actors on the scene** ☺

- Database **Designers**:
  - Responsible for:
    - identifying data to be stored in database,
    - the structure, the constraints, and functions or transactions against the database.
    - They must communicate with the end-users and understand their needs.
    - Develop different views.

# Database **Users** …
## End users

**Actors on the scene** (continued) ☺

- **End-user's** people whose jobs require access to the database to for querying, updating and generating reports; the database is primarily existing for their use.
- There are several categories for end users.

# Database **Users** …

- System **Analysts** & App **Programmers** **(software engineers):**
  - **S.A** determine the requirements of end users develop spec for canned transaction to meet these requirements.
  - **A.P** implement these spec as program.
    - They test, debug, document and maintain these canned transaction.
  - S.A and A.P commonly referred to as software developer or software engineers.

# Database **Users** …
## Workers Behind the Scene

- DBMS system designer & implementers **(ORACLE, SQL):**
  - Implementing the catalog, processing query language, processing the interface, buffering data, controlling concurrency, data recovery & security.
- Tool developer:
  - Develop package for database modeling & design, database system design, improve performance.
  - Tools are optional to purchase.
- Operator & maintenance personnel:
  - Responsible for the actual running & maintenance of the h/w & s/w environment for the database.

# Data Models

- **Data Model**:
  - **Data abstraction** : remember ( lec. 1)
  - **Data Model**: "A **set of concepts** to **describe** the **structure*** of a database, and certain **constraints** that the database should obey ".
- **Data Model basic Operations:**
  - Operations for specifying database **retrievals** and **updates** by referring to the concepts of the data model.
  - Operations on the data model may include basic operations and **user-defined** operations.

# **Categories** of data models*

- **Conceptual** (High-level, Semantic) data models*:
  - Provide concepts that are close to the way many users perceive data. (Also called ***entity-based*** or ***object-based*** data models.)
- **Physical** (Low-level, internal) data models:
  - Provide concepts that **describe details** of how data is stored in the computer ( this category meant for computer specialists, not for end users).
- **Implementation** (representational) data models:
  - Provide concepts that **fall between the above two**, balancing user views with some computer storage details.
  - This category include the Widely used **relational data model**.

# Schemas, Instances, and Database State

- **Database Schema**: The "***description***" of a database. Includes descriptions of the database structure and the constraints that should hold on the database.
  - **Schema Diagram**: A diagrammatic display of (some aspects of) a database schema.
  - **Schema Construct**: A component of the schema or an **object within the schema**, e.g., STUDENT, COURSE.
- **Database Instance**: The ***actual data stored*** in a database at a particular moment in time. Also called **database state** (or occurrence).

# Schema Diagram*

**STUDENT**

| Name | Student_number | Class | Major |
|---|---|---|---|

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|

# Database State ☺

- **Database State**: Refers to the **content** of a database **at a moment in time.**
- **Initial Database State**: Refers to the database when it is loaded.
- **Valid State**: A state that satisfies the structure and constraints of the database.
- **Distinction**
  - The database **schema** *changes very* ***infrequently*** (rarely).
  - The database **state** *changes every time the database is updated.*
  - **Schema** is also called **intension***, whereas **state** is called **extension**

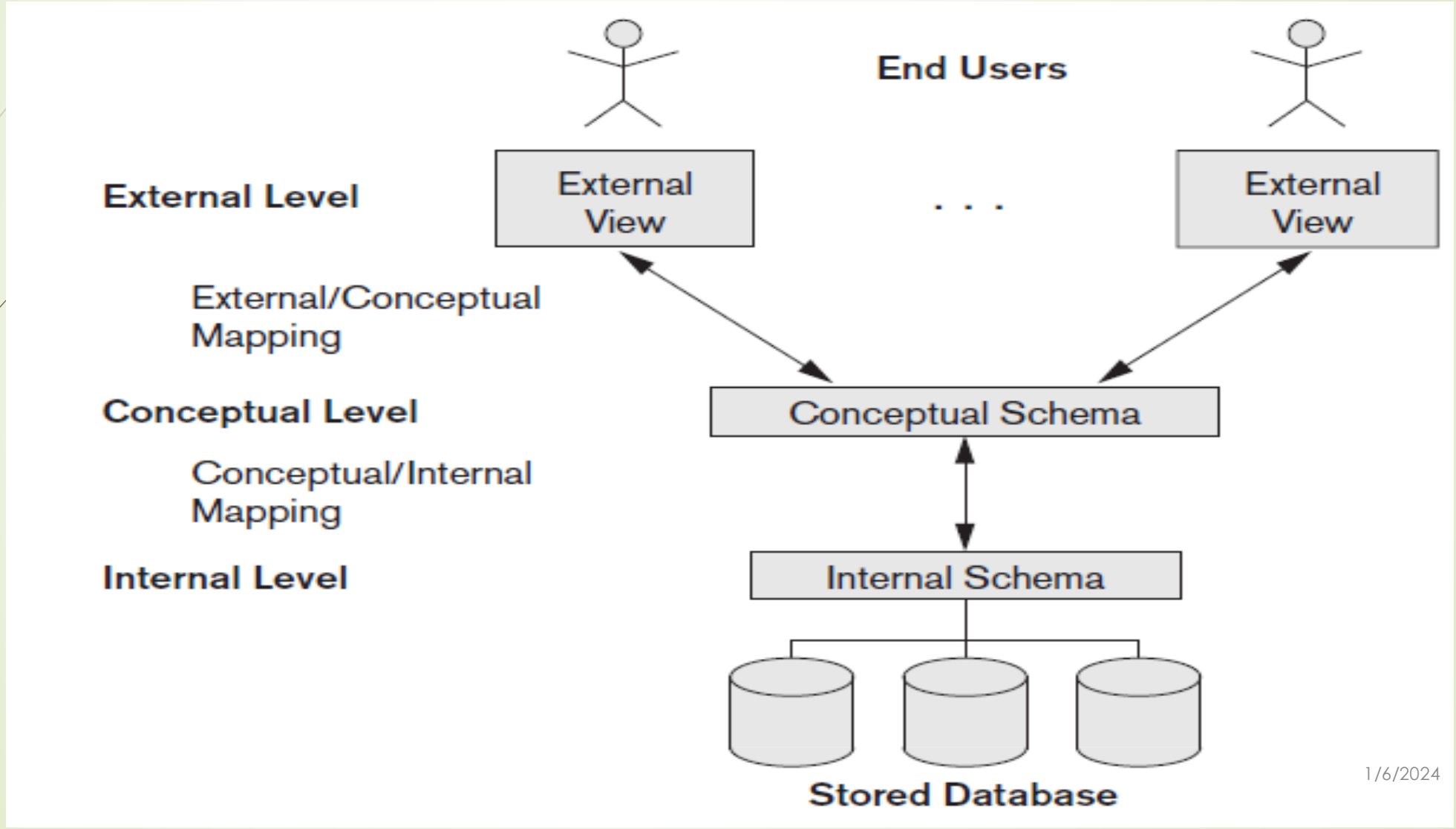
# Three-Schema Architecture

- The **goal** of the three-schema architecture, is "to separate the user applications from the physical database"*.
- So, its Proposed to **support DBMS** characteristics of:
  - Program-data **independence**.
  - Support of **multiple views** of the data.

# DBMS schemas at three levels ☺

- **Internal schema:** at the internal level
  - **to describe physical storage** structures and **access paths**.
  - Typically uses a physical data model.
- **Conceptual schema** at the conceptual level:
  - **to describe the structure and constraints for the whole database** for a community of users.
  - Its hide details of physical storage.
  - Uses a conceptual or representational data model.
- **External schemas** at the external level (view level):
  - **to describe the various user views**.
  - each external schema is typically implemented using a representational data model

# Three-Schema Architecture ☺



# Data Independence

- The three-schema architecture can be used to further explain the concept of **data independence**, which can be defined as "***the capacity to change the schema at one level of a database system without having to change the schema at the next higher level***". We can define two types of data independence:
  - **Logical Data Independence**: The capacity to **change the conceptual schema** without having to change the external schemas and their application programs*.
  - **Physical Data Independence**: The capacity to **change the internal schema** without having to change the conceptual schema.

# DBMS Languages ☺

- **Data Definition Language (DDL):**
  - Used by the DBA and database designers to specify the conceptual schema of a database.
  - In many DBMSs, the DDL is also used to define internal and external schemas (views).
- **Storage definition language (SDL):**
  - used to define internal schemas
- **View definition language (VDL):**
  - used to define external schemas(user view)/ and mappings to conceptual schema.
- **Data Manipulation Language (DML):**
  - Used to specify database retrievals and updates ( insertion, deletion , modification )

# DBMS Interfaces

**User-friendly interfaces** provided by a DBMS may include the following:

- **Menu-based**, popular for browsing on the web
- **Forms-based**, designed for naïve users
- **Graphics-based** (Point and Click, Drag and Drop etc.)
- **Natural language**: requests in written English or some other lang.
- **Combinations** of the above.
- ….

# SQL

# What is SQL?

- SQL stands for **S**tructured **Q**uery **L**anguage.
- SQL lets you access and manipulate databases.
- SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

# What Can SQL do?

- SQL can execute queries against a database.
- SQL can retrieve data from a database.
- SQL can insert records in a database.
- SQL can update records in a database.
- SQL can delete records from a database.
- SQL can create new databases.
- SQL can create new tables in a database.
- SQL can create stored procedures in a database.
- SQL can create views in a database.
- SQL can set permissions on tables, procedures, and views.

# Is SQL Standard ?

- SQL is a Standard - BUT....
- Although SQL is an ANSI/ISO standard, there are different versions of the SQL language.
- However, to be compliant with the ANSI standard, they all support at least the major commands (such as SELECT, UPDATE, DELETE, INSERT, WHERE) in a similar manner.

# Using SQL in Your Web Site

- To build a web site that shows data from a database, you will need:
- An RDBMS database program (i.e., MS Access, SQL Server, MySQL):
  - To use a server-side scripting language, like PHP or ASP.
  - To use SQL to get the data you want.
  - To use HTML / CSS to style the page.
- What is RDBMS?
  - RDBMS stands for Relational Database Management System.
  - RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
  - The data in RDBMS is stored in database objects called tables.
  - A table is a collection of related data entries, and it consists of columns and rows.

# SQL Syntax

## Database Tables

- A database most often contains one or more tables.
- Each table is identified by a name (e.g., "Customers" or "Orders").
- Tables contain records (rows) with data.

| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
|---|---|---|---|---|---|---|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |

- The table above contains five records (one for each customer) and seven columns (CustomerID, CustomerName, ContactName, Address, City, PostalCode, and Country).

# SQL Statements

- Most of the actions you need to perform on a database are done with SQL statements.
- The following SQL statement selects all the records in the "Customers" table:
- `SELECT * FROM Customers;`
- Now ; we will teach you all about the different SQL statements.
- Keep in Mind That...
  - SQL keywords are NOT case sensitive: **select** is the same as **SELECT**
- Semicolon after SQL Statements?
  - Some database systems require a semicolon at the end of each SQL statement.
  - Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

# The Most Important SQL Commands

- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index


SELECT

# SQL SELECT Statement

- The SELECT statement is used to select data from a database.
- The data returned is stored in a result table, called the result-set.
- **SELECT Syntax** :
  - `SELECT column1, column2, ...`
  - `FROM table_name;`
- Here, column1, column2, ... are the field names of the table you want to select data from.
- If you want to select all the fields available in the table, use the following syntax:
  - `SELECT * FROM table_name;`

# SQL SELECT Statement…

## example:

- Ex1: write the select statement which select the Customer Name and its City from the "Customers" table

```
SELECT CustomerName, City FROM Customers;
```

| CustomerID | CustomerName | ContactName | Address | City | PostalCode | Country |
|---|---|---|---|---|---|---|
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |

- Ex2: write the select statement which select all data about Customer stored in "Customers" table

```
SELECT * FROM Customers;
```