

Database Management System

IT 261

Bachelor of Information Technology

Prepared by Dr. Khaled Eskaf

Lecture 1





Learning Objectives

This course is centered on the comprehensive study and application of relational database management systems (RDBMS), with a specific focus on analysis, design, normalization, and implementation techniques. It's tailored to provide a robust understanding of how relational databases function and how they can be effectively utilized and managed using MySQL, a popular open-source relational database management system.

GRADING

Quiz /week = 8 Marks = 8X10 weeks = 80%

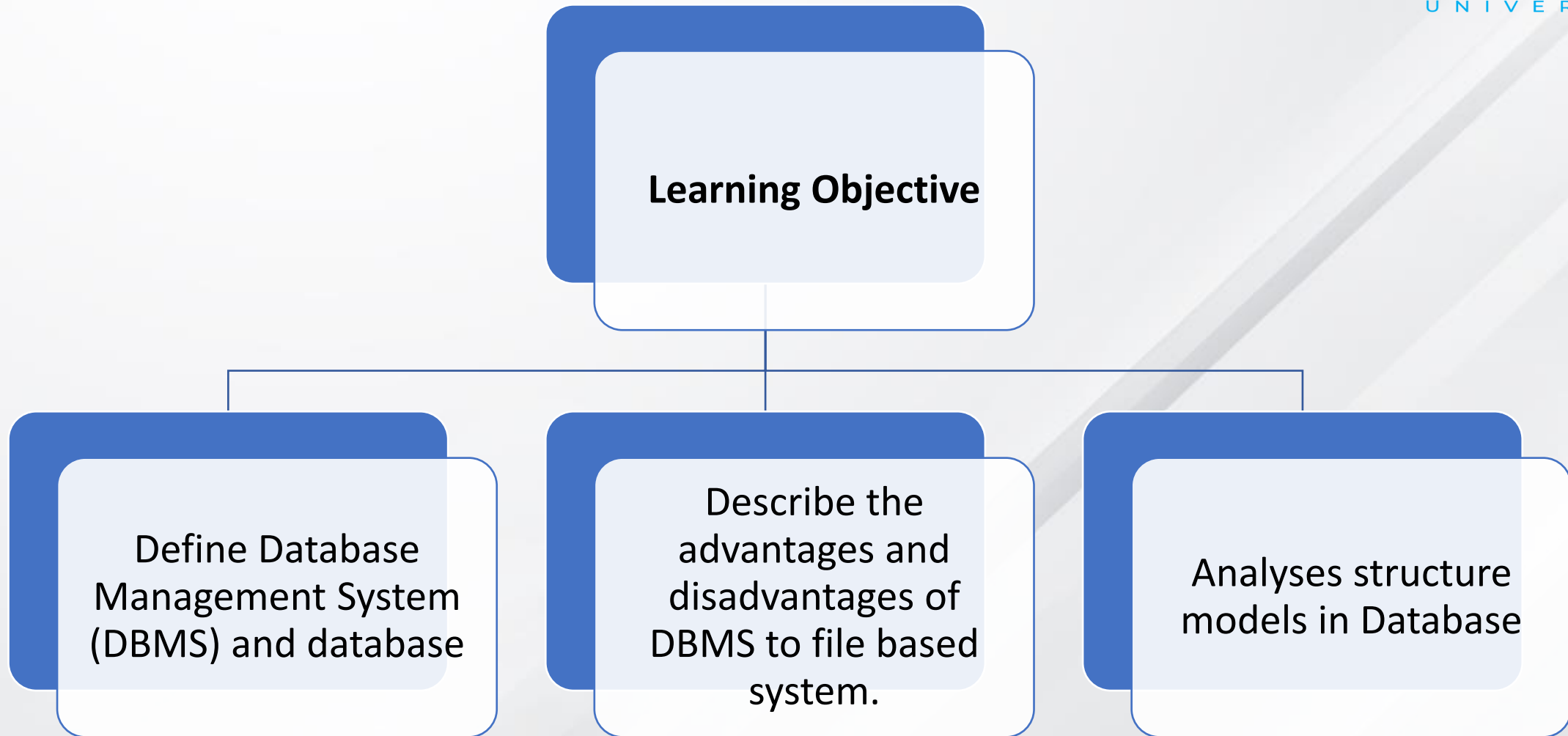
Final Exam: 20%

By the end of this course, you will not only understand the theoretical aspects of database management but will have also navigated the practical challenges and solutions in MySQL.

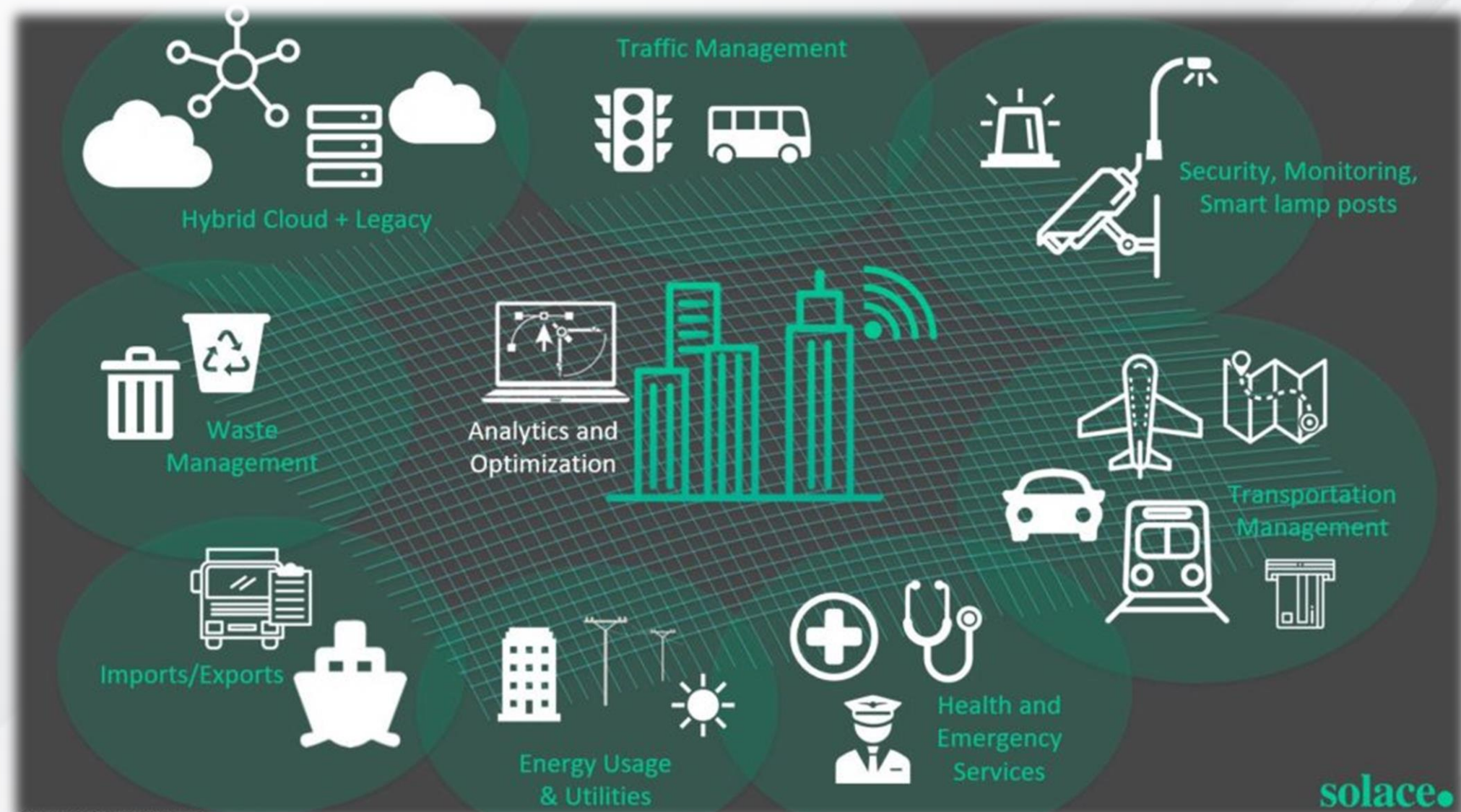


Course Outlines

Week	Tonic	Obiective	MySQL Practical Activities
1	Introduction to Database Concepts	Understand basic database concepts.	Install MySQL, explore the interface, create a basic database.
2	Relational Data Model	Learn about the relational data model.	Create relational tables in MySQL.
3	Relational Integrity Rules	Understand relational integrity constraints.	Implement primary and foreign key constraints in MySQL.
4	Relational Algebra and Basic SQL	Explore relational algebra and basic SQL commands.	Use basic SQL queries in MySQL.
5	Advanced SQL Queries	Delve into complex SQL operations.	Execute advanced SQL commands in MySQL.
6	Introduction to Database Normalization	Begin learning database normalization principles.	Apply the first normal form (1NF) in MySQL.
7	Advanced Database Normalization	Explore advanced normalization techniques.	Normalize a database in MySQL to 2NF and 3NF.
8	Data Modeling and ER Diagrams	Understand data modeling and ER diagrams.	Create an ER diagram and implement it in MySQL.
9	Database Design for Real-World Apps	Apply database design to real-world scenarios.	Develop a database schema based on a real-world example.
10	Project and Review	Apply all learned concepts in a comprehensive project.	Complete a project in MySQL incorporating all learned skills.



Smart City Data Base Management



Database Management System



Definitions

Database:

- A very large, integrated collection of data.. مجموعة كبيرة جدًا ومتكاملة من البيانات
- Models real-world enterprise نماذج المشاريع في العالم الحقيقي
 - Entities (e.g., students, courses)
 - Relationships (Ex., Eva is taking Biology)

Roll No.	Name	Course	
CS08	Steive	Comp. Sci.	→ Entity 1
EE54	Jhson	Electronics	→ Entity 2
B12	Eva	Biology	→ Entity 3
F32	Jhson	Finance	→ Entity 4
M26	Erica	Maths	→ Entity 5

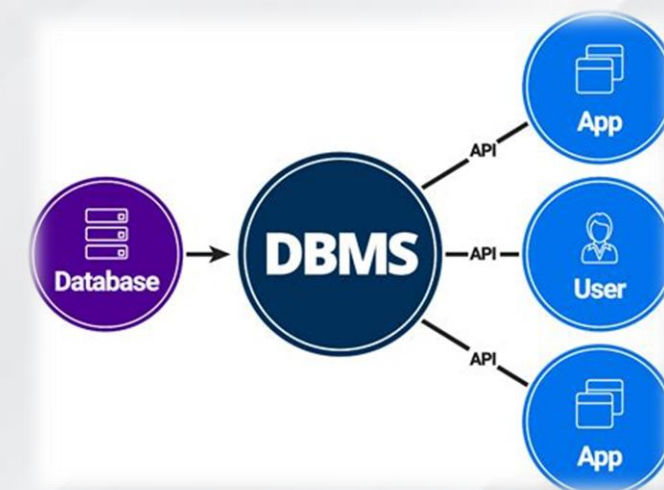
Student Table

Database Management System (DBMS) نظام إدارة قواعد البيانات

a software package designed to store and manage databases.

• Examples of Database Applications:

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades



Purpose of Database Systems

❑ 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
تكرار البيانات وعدم اتساقها
 - Multiple file formats, duplication of information in different files
تنسيقات ملفات متعددة، وتكرار المعلومات في ملفات مختلفة
- Difficulty in accessing data
صعوبة الوصول إلى البيانات
 - Need to write a new program to carry out each new task
الحاجة إلى كتابة برنامج جديد لتنفيذ كل مهمة جديدة
- Data isolation — multiple files and formats
عزل البيانات – ملفات وتنسيقات متعددة



Drawbacks of using file systems (cont.)

❖ Integrity problems

□ Integrity constraints (e.g. account balance > 0) become تصبح قيود النزاهة (على سبيل المثال، رصيد الحساب < 0). "مدفونة" في كود

البرنامج بدلاً من ذكرها صراحةً

“buried” in program code rather than being stated explicitly

□ Hard to add new constraints or change existing ones من الصعب إضافة قيود جديدة أو تغيير القيود الحالية

❖ Atomicity of updates

□ Failures may leave database in an inconsistent state with partial updates carried out قد تؤدي

حالات الفشل إلى ترك قاعدة البيانات في حالة غير متناسقة مع إجراء التحديثات الجزئية

□ E.g. transfer of funds from one account to another should either complete or not

happen at all على سبيل المثال. يجب أن يكتمل تحويل الأموال من حساب إلى آخر أو لا يحدث على الإطلاق

Drawbacks of File System

- = Data redundancy
- = Require more primary memory
- = Processor has to do lot of work
- = More Computations

Why Use a DBMS?



نستخدم نظام إدارة قواعد البيانات (DBMS) لعدة أسباب رئيسية:

- Separation of the Data definition and the Program
- Abstraction into a simple model
- Data independence and efficient access.
- Reduced application development time – ad-hoc queries
- Data integrity and security.
- Uniform data administration.
- Concurrent access, recovery from crashes.
- Support for multiple different views

- فصل تعريف البيانات عن البرنامج: يتيح فصل البيانات عن البرامج، مما يسهل التعامل مع البيانات وتحديثها دون الحاجة لتغيير البرامج.
 - التجريد إلى نموذج بسيط: يقدم نماذج بسيطة وفعالة لتمثيل وتخزين البيانات.
 - استقلالية البيانات والوصول الفعال: يضمن الوصول السريع والفعال إلى البيانات، مع المحافظة على استقلاليتها.
 - تقليل وقت تطوير التطبيق: يسرع من تطوير التطبيقات من خلال توفير الاستعلامات المخصصة.
 - سلامة البيانات وأمنها: يضمن حماية البيانات وسلامتها.
 - إدارة موحدة للبيانات: يسهل إدارة البيانات وتنظيمها في مكان مركزي.
 - الوصول المتزامن والتعافي من الأعطال: يدعم الوصول المتعدد في نفس الوقت ويوفر آليات للتعافي من الأعطال.
 - تقديم دعم لوجهات نظر مختلفة: يتيح تمثيل البيانات بطرق متعددة لتلبية احتياجات مختلف المستخدمين.
- هذه الخصائص تجعل نظام إدارة قواعد البيانات أداة فعالة وأساسية في تنظيم وإدارة البيانات في مختلف التطبيقات.



دراسة قواعد البيانات مهمة لأسباب عدة:

❑ Shift from computation to information:

- at the "low end": scramble to webspace (a mess!)
- at the "high end": scientific applications

❑ Datasets increasing in diversity and volume.

- Digital libraries, interactive video, Human Genome project, EOS project.
- ... need for DBMS exploding

❑ DBMS encompasses most of CS

- OS, languages, theory, "AI", multimedia, logic

❑ تسهم في الانتقال من مجرد التعامل مع الأرقام إلى التعامل مع المعلومات، مما يوسع نطاق استخدامات الحاسوب.
-في النطاقات "المنخفضة"، مثل مساحات الويب، تساعد في تنظيم المعلومات وتجنب الفوضى.

-في التطبيقات "العليا"، مثل الأبحاث العلمية، تلعب دوراً مهماً في تنظيم ومعالجة البيانات الكبيرة.

❑ تتعامل مع مجموعات بيانات متزايدة في الحجم والتنوع، مثل المكتبات الرقمية والفيديو التفاعلي.

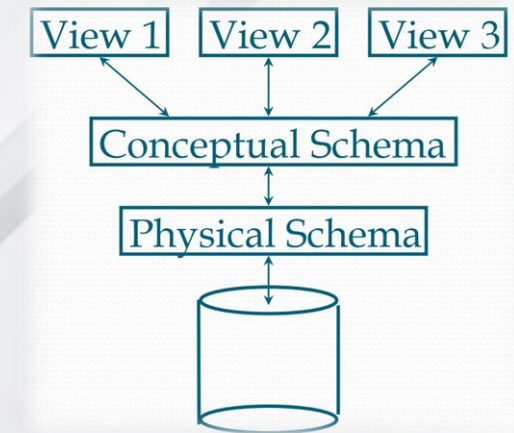
❑ تشمل العديد من مجالات علوم الكمبيوتر مثل نظم التشغيل، اللغات البرمجية، النظرية، الذكاء الاصطناعي، والوسائط المتعددة.

- فهم ودراسة قواعد البيانات يفتح الباب أمام فهم أعمق لعلوم الكمبيوتر وتطبيقاتها في مجالات متعددة.

Levels of Abstraction

Many **views**, single **conceptual (logical) schema** and **physical schema**.

- Views describe how users see the data.
- Conceptual schema defines logical structure. Sometime we separate between conceptual level and logical level
- Physical schema describes the files and indexes used
- *Schemas are defined using DDL (Data Definition Language)*
- *data is modified/queried using DML (Data Manipulation Language)*



مستويات التجريد في قواعد البيانات تتضمن عدة طبقات:

- **المستوى المفاهيمي:** يحدد البنية المنطقية لقاعدة البيانات بأكملها. يُظهر النظام بأكمله كوحدة واحدة دون الخوض في التفاصيل الفنية.
 - **المستوى المنطقي (مخطط منطقي):** يصف كيف يرى المستخدمون البيانات. يحدد أنواع البيانات والعلاقات بينها.
 - **المستوى الفيزيائي (مخطط فيزيائي):** يصف البيانات في مستوى أكثر تفصيلاً مثل الملفات والفهارس المستخدمة في تخزين البيانات.
- لغات قواعد البيانات تشمل:

- **DDL (لغة تعريف البيانات):** تستخدم لتعريف وتعديل البنية الأساسية لقاعدة البيانات.
 - **DML (لغة معالجة البيانات):** تستخدم للاستعلام وتعديل البيانات ضمن قاعدة البيانات.
- هذه المستويات تسمح بفهم وتصميم قاعدة البيانات بطريقة منظمة وفعالة.

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**

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

في سياق قواعد البيانات، تشير مستويات التجريد إلى الطرق المختلفة التي يتم من خلالها عرض وإدارة البيانات:

• **المستوى الفيزيائي:** يتعلق هذا المستوى بكيفية تخزين وتنظيم البيانات فعلياً في قاعدة البيانات. يتعامل مع تفاصيل تخزين البيانات، مثل هيكل السجلات والملفات.

• **المستوى المنطقي:** في هذا المستوى، يتركز الاهتمام على البيانات نفسها والعلاقات بين العناصر البيانية المختلفة. يصف ما هي البيانات المخزنة وكيفية ارتباطها ببعضها البعض، دون التطرق إلى تفاصيل التخزين الفيزيائي.

• **المستوى الظاهري:** هذا هو أعلى مستوى من التجريد ويتعلق بكيفية تفاعل المستخدمين النهائيين مع البيانات. يمكن أن يشمل تقديم البيانات بتنسيقات مختلفة أو إخفاء بعض التفاصيل (مثل المعلومات الحساسة) لأغراض الأمان. يسمح هذا المستوى بعرض مختلف لقاعدة البيانات، مصمماً لتلبية احتياجات المستخدمين أو تطبيقات مختلفة. كل مستوى يجرد ويخفي التفاصيل من المستويات الأدنى، مما يجعل التفاعل مع قاعدة البيانات وإدارتها أسهل.

Instances and Schemas:

- **Schema** – the logical structure of the database
 - Example: The database consists of information about a set of customers and accounts and the relationship between them)
 - 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

النماذج والمثيلات في قواعد البيانات:

• **النموذج Schema**: يشير إلى الهيكل المنطقي لقاعدة البيانات.

- مثال: قاعدة البيانات تتكون من معلومات حول مجموعة من العملاء والحسابات والعلاقة بينهم.
- يشبه نوع المعلومات المتعلقة بمتغير في برنامج.
- **النموذج الفيزيائي**: تصميم قاعدة البيانات على المستوى الفيزيائي.
- **النموذج المنطقي**: تصميم قاعدة البيانات على المستوى المنطقي.

• **المثيل Instance** المحتوى الفعلي لقاعدة البيانات في لحظة زمنية محددة.

- يشبه قيمة متغير في لحظة زمنية معينة.

- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
 - Applications depend on the logical schema
 - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

استقلالية البيانات الفيزيائية :

- هي القدرة على تعديل النموذج الفيزيائي دون تغيير النموذج المنطقي لقاعدة البيانات. هذا يعني أن التطبيقات تعتمد على النموذج المنطقي، وبشكل عام، ينبغي أن تكون الواجهات بين المستويات والمكونات المختلفة محددة بشكل جيد بحيث لا تؤثر التغييرات في جزء ما بشكل كبير على الأجزاء الأخرى.

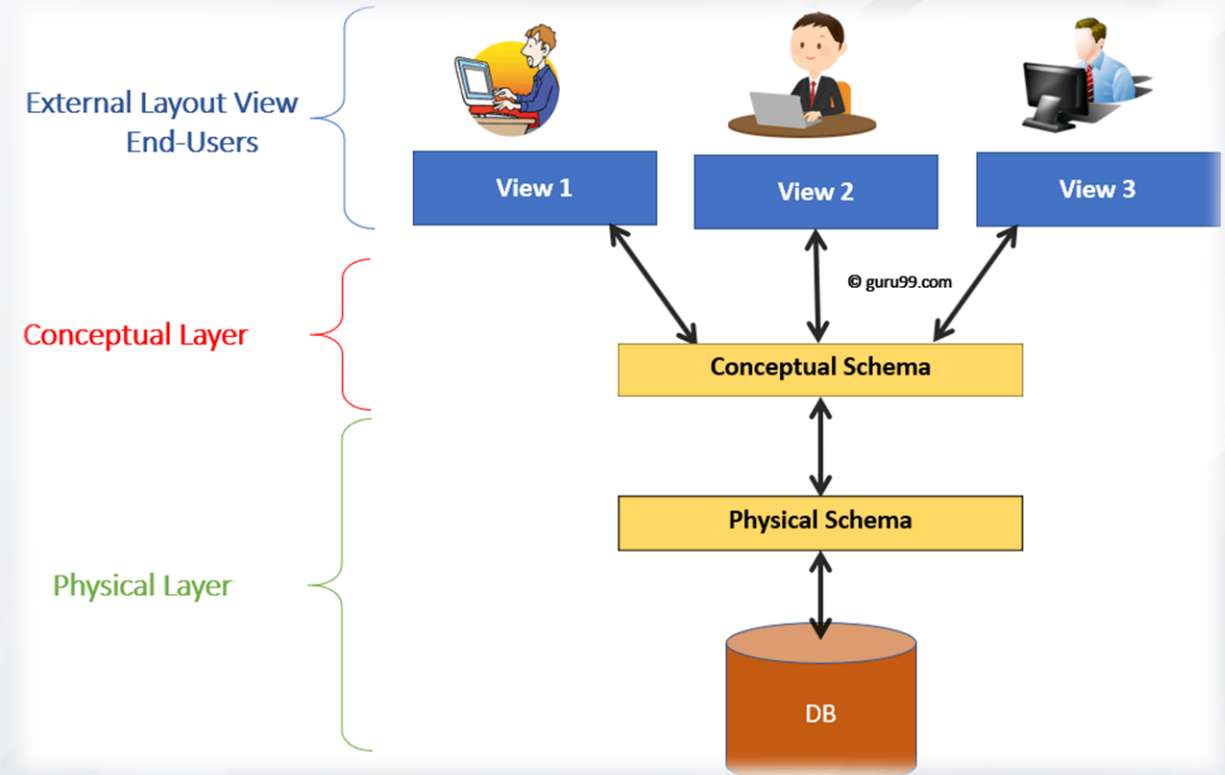
Schemas

- ☆ The **overall design** of the database.
- ☆ Database schema = Variable declaration.
- ☆ **Type** definitions.
- ☆ **No frequent changes** or not at all.
- ☆ Example:

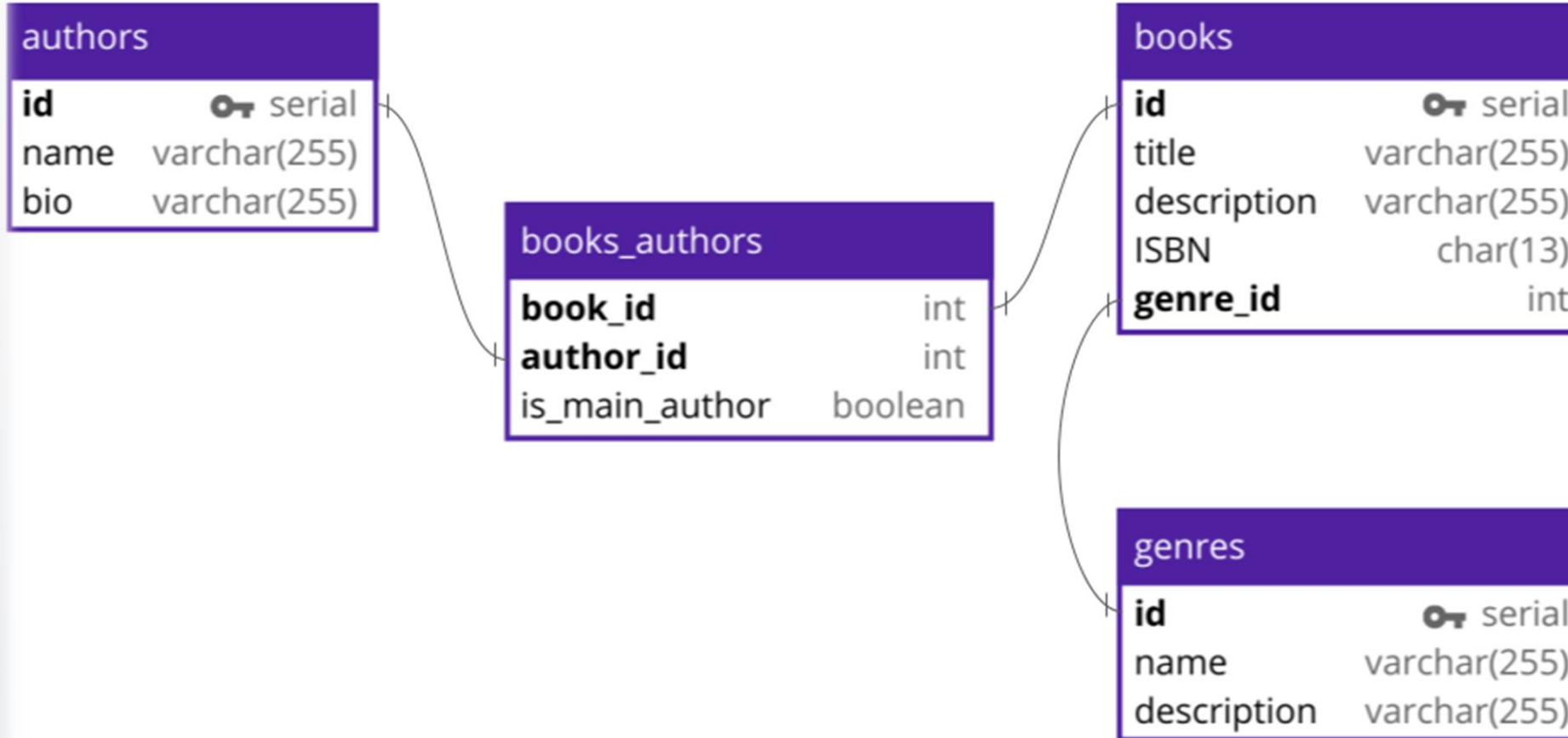
```
type Student = record
    Rollno : numeric (5);
    Name   : char (25);
    Class  : char (10);
end;
```

Instances

- ☆ The collection of information stored at a **particular moment**.
- ☆ **Frequent** changes.
- ☆ Database instant = Value of the variable.



Example of Database Schema.







مجموعة جامعة
الجميع الذكية
للتعليم القابضة

MIDOCEAN
UNIVERSITY



Thank You

