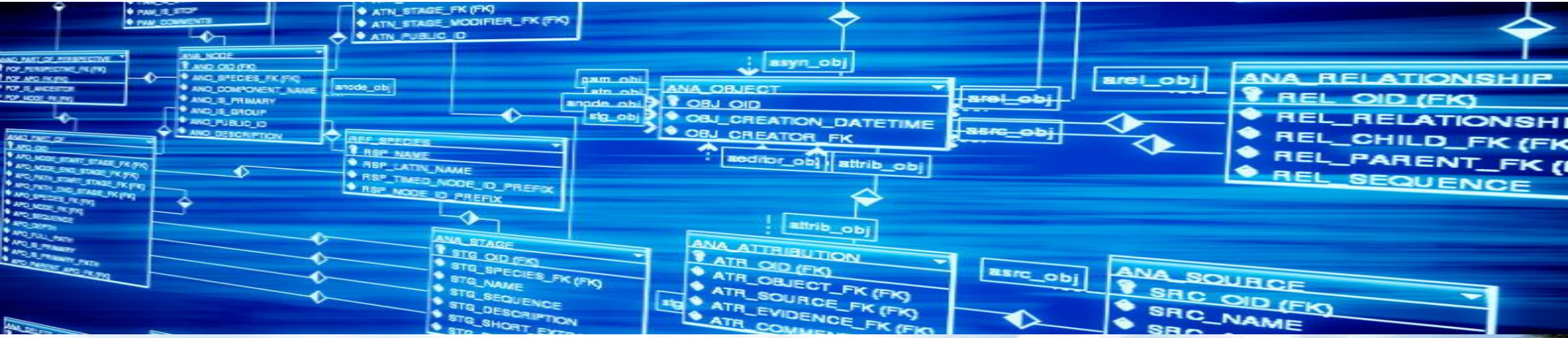
SOF202 Database



Subashini Ganapathy
subashini.ganapathy@xmu.edu.my

Chapter 1

A general introduction
(Part 2)

Learning outcomes

1. Describe data models.
2. Database system structure (Level of Data abstraction)

Data model : how to describe data?

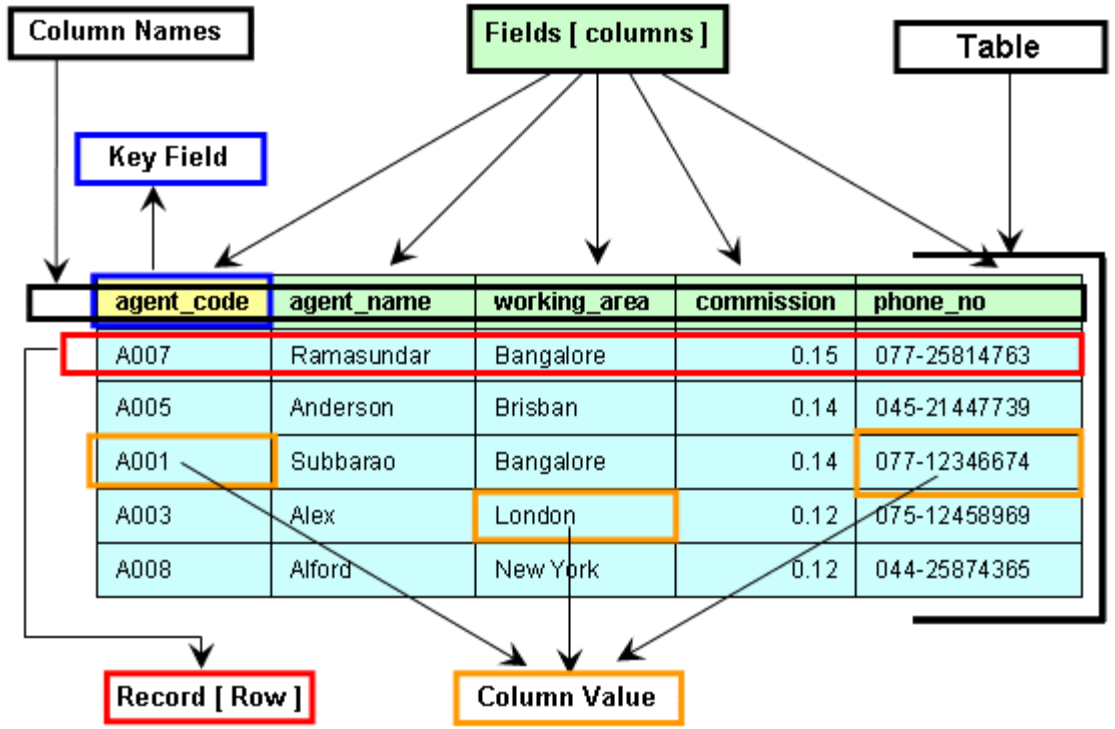
- A **data model** is a set of concepts for **display data**.
- A **schema** is a **description of a specific set of data** using a specific data model.
- Data model is used to describe:
  - data
  - data relationships
  - data semantics
  - data constraints

Types of data model:

- Entity-Relationship model
- Relational model
- Other models:
  - Object-oriented model
  - Semi-structured data models
  - Older models: network model and hierarchical model

Data model : How to describe data?

- The most widely used data model today is the relational model.
- **Relationship** is the main concept, which is essence a **table** with **rows** and **columns**.
- A **schema** is a description of the columns (fields, attributes) and keys in a relation.



Example: University Database

Schema:

- Students(sid text,
  name text,
  login text,
  age integer,
  gpa float)

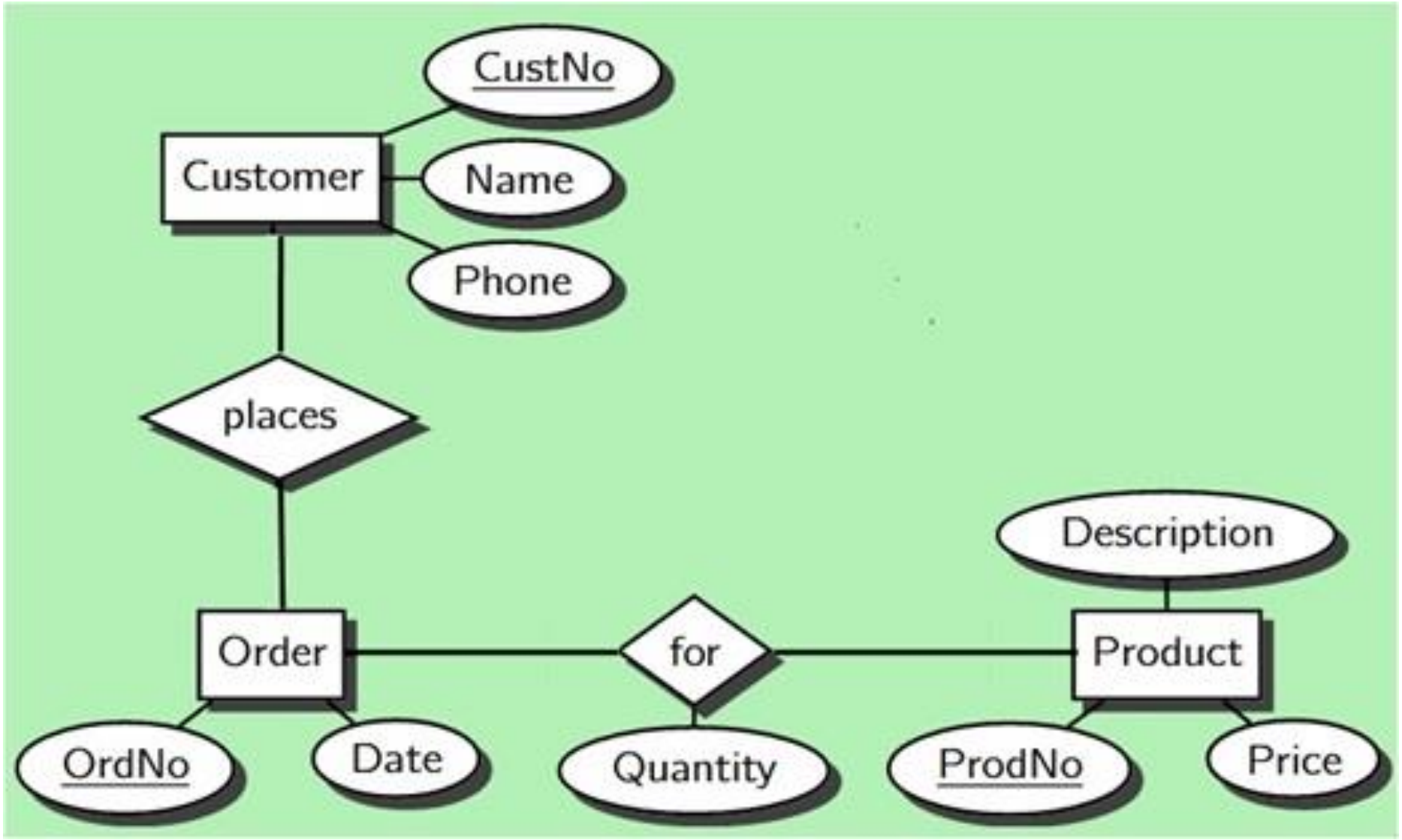
- Courses(cid text,
  cname text,
  credits integer)

- Enrolled(sid text,
  cid text,
  grade text)

Entity-Relationship Model

- ER models are readily translated to relations.
- ER models, also called an ER schema, are represented by ER diagrams.
  - Entities are referred to as tables that contain specific data (data)
  - Attributes – shows the characteristic of an entity.
  - Relationships are characterized as the connections or exchanges between different entities.

Entity-Relationship Model



Entity-Relationship Model (cont.)

- E-R model of real world
  - **Entities**
    - E.g. customer, order, product
  - **Relationships between entities**
    - E.g. **Customer** **place** an **order** (1:M, or 1:1)
    - **Order** **for** **product**.
- Widely used for database design
  - Database design in E-R model usually converted to design in the relational model (coming up next in chapter 2) which is used for storage and processing.

Data Abstraction in dB

- Abstraction is one of the **main features** of database systems
- Keeping **irrelevant information hidden** from users and giving them an abstract view of the data makes user-database interaction simple and effective.
- These constraints are hidden from users so that they can simply access the data, and just the relevant portion of the database is made visible to them through data abstraction.
- There are basically three level:
  - External Level / View Level
  - Conceptual Level/ Logical Level
  - View or External Level

Level of Data Abstraction

Views level

Data abstraction at its **highest** level.

The **user's interaction** with the **database** system is described at this level.

Conceptual level

In a three-level data abstraction architecture, this is the **middle level**.

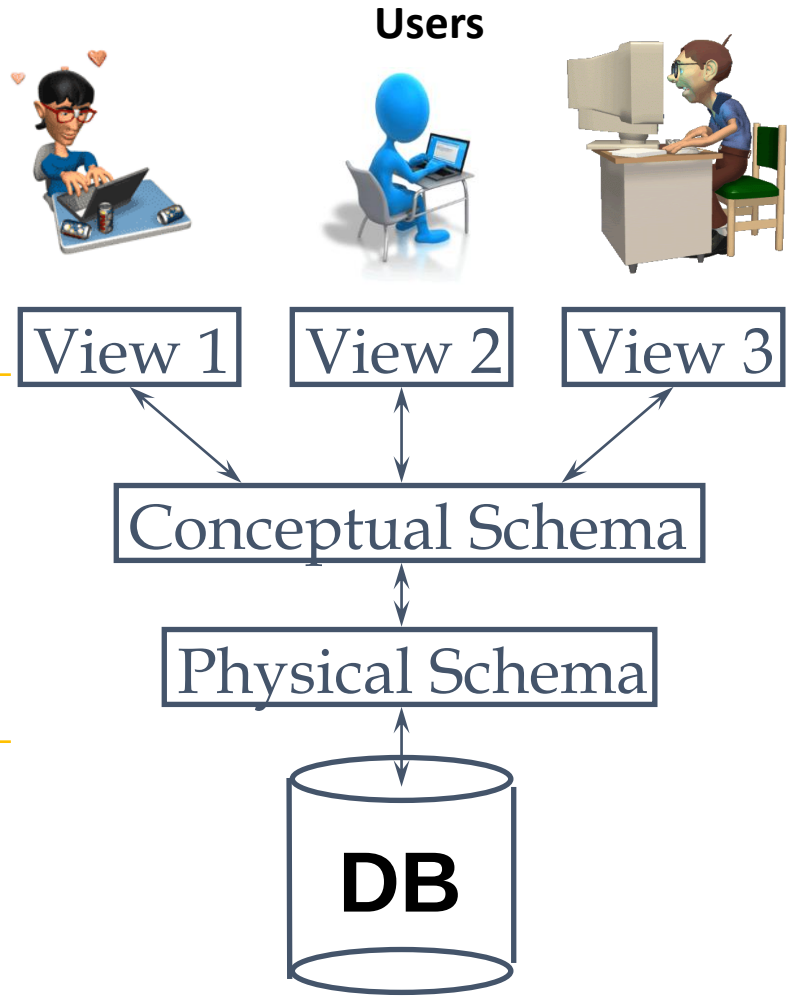
It explains **what information** is **kept** in a database.

Physical level

The **lowest** data abstraction level is this one.

It explains **how information** is actually **kept** in databases.

At this level, you can obtain details about **complex data structures**.



Physical level

- Being the lowest level of abstraction, the physical level can be better understood with an example.
- For example, data is saved in the form of blocks but customer information is kept in tables.

- Sequential file organization—which is a result of continuous record storage—is another illustration of physical-level abstraction.
- Indexes enable us to retrieve records from indexed file organizations.

Logical Level

- The next higher level, or intermediate level, is the logical level.
- It describes the types of data kept in the database and their relationships.
- By outlining how tables should be built and how they should be connected, it aims to convey the full or complete set of data.
- Compared to the physical level, it is less complex.

View Level

- Using a graphical user interface (GUI) to interact with a system, for instance, to access an application's functionality.
- Here, the GUI is the view level, and the user is kept in the dark about the specifics, so they are unaware of how and what data is saved.

Instance and schema in DBMS

- Definition of schema:
  - **Design of a database** is called the **schema**.
  - Schema is of three types:

Physical schema | Logical schema | view schema

- **Physical schema** refers to the design of a database at the level of **how the data stored in storage blocks is described**.
- Just as its name implies, the Physical schema provides information about the actual location of the data that the user is storing.
- The Database Administrators (DBA) make decisions about what data needs to be stored where, how to be fragmented, and which hard drive to keep it on. Whether the data must be disseminated or centralized is decided by them.
- Even while the data is displayed to us as tables at the view level, it is actually only saved as files.
- Depends on DBA, how he/she manages the data in overall.

Instance and schema in DBMS

Physical schema | **Logical schema** | View schema

- Example: Let's look at an instance where the data is stored using the relational model.
- A student's data must be stored; the columns in the student table will include student name, age, mail id, roll no, and other information. While building the database, we have to specify each of these at this point.
- Even if the data is kept in a database, the logical schema defines the structure of the tables, such as the student, teacher, books, and so on. This defines the relationships between the tables as well.
- In general, we might state that, at the logical schema, we are building a blueprint of the data.

Instance and schema in DBMS

- If we have a login-id and password in a university system, then as a student, we can view our marks, attendance, fee structure, etc. But the faculty of the university will have a different view.
- He will have options like salary, edit marks of a student, enter attendance of the students, etc. So, both the student and the faculty have a different view. By doing so, the security of the system also increases.
- In this example, the student can't edit his marks but the faculty who is authorized to edit the marks can edit the student's marks.
- Similarly, the dean of the college or university will have some more authorization and accordingly, he will has his view. So, different users will have a different view according to the authorization they have.

Revise the following article regarding Database Models

https://www.studytonight.com/dbms/database-model.php

Revise the following article regarding Level of Data Abstraction in Database

- https://afteracademy.com/blog/what-is-data-abstraction-in-dbms-and-what-are-its-three-levels

Revise the following article regarding Semi Structured Data Model

https://www.geeksforgeeks.org/what-is-semi-structured-data/

Tutorial 2

- What are the advantages of using DBMS?
- What are the main reason of abstraction in Database?