FACULTY OF TECHNOLOGY
UNIVERSITY OF COLOMBO
SRI LANKA

Database Management Systems -II

Lecture 1: Introduction of Database management System (DBMS II)

Department of Information and Communication Technology

Outline of the syllabus

- Week 1 – Introduction, Revision of Database Design Process and ER
- Week 2 – Enhanced Entity Relationship (EER) -1
- Week 3 – Enhanced Entity Relationship Mapping
- Week 4 – Relational Algebra
- Week 5 – Transact -SQL
- Week 6 – Stored Procedures and Functions
- Week 7 – Data Storages – Disks and Files

Outline of the syllabus

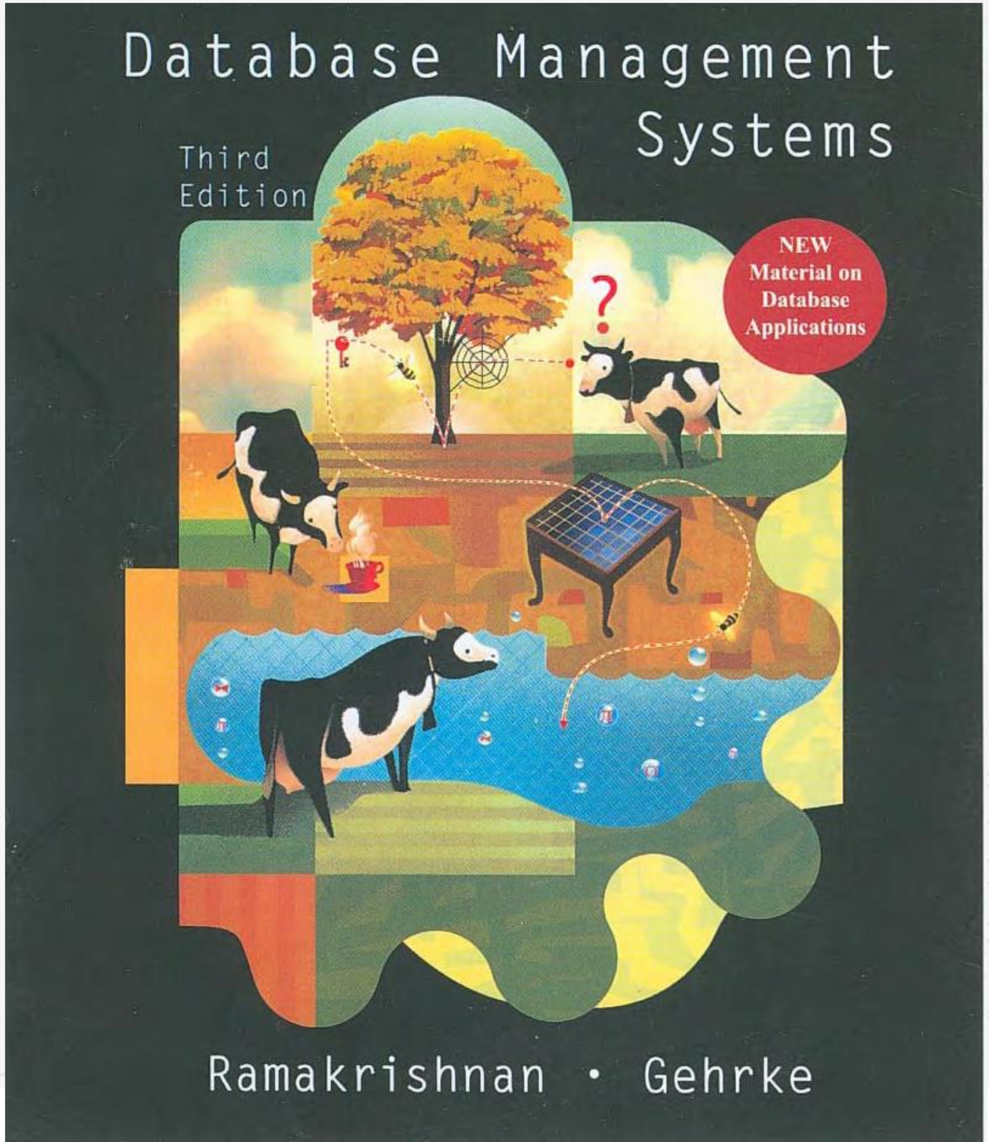
- Week 8 – Database Connectivity
- Week 9 – Mid Semester Exam
- Week 10 – File Organization and Indexes
- Week 11 – Query processing
- Week 12 - Physical Database Design and Turning Overview
- Week 13– Transaction and Concurrency Control
- Week 14 – Crash Recovery

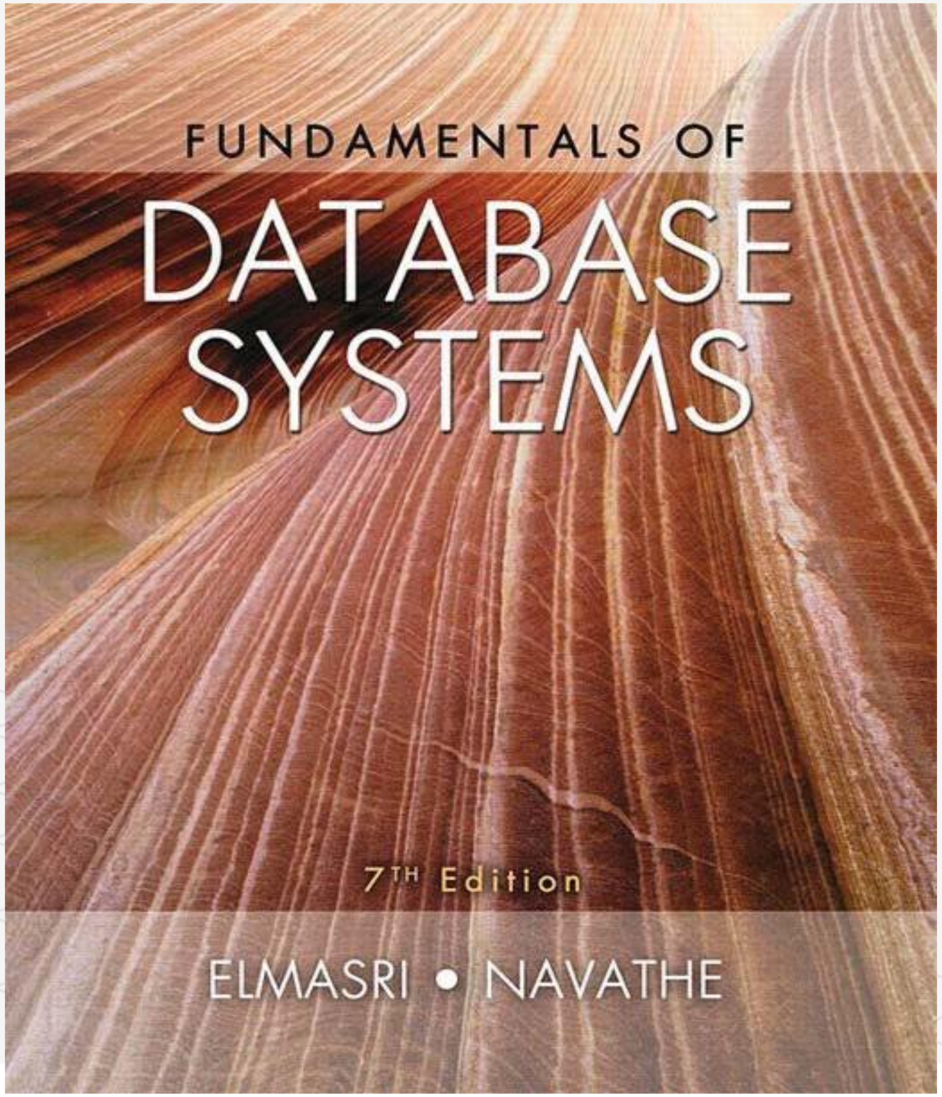
Learning Objectives

After completing this module, you will be able to:

- ❖ get the expert knowledge in Relational Database design for any application
- ❖ good knowledge of implementing commercially available RDBMS .
- ❖ Knowledge of the Database Theory and its features
  - ✓ Data Storage
  - ✓ Indexes
  - ✓ Query Optimization
  - ✓ Transaction Processing
  - ✓ Concurrency Control
  - ✓ Crash Decory

Reference Book and chapters for this Course Module


Database Management Systems
Third Edition
NEW Material on Database Applications
Ramakrishnan • Gehrke



Evaluation Criteria

- Assignments – 10%
- Mid Semester Exam – 20%
- Final Exam – 70%

The examination will be a comprehensive exam based on the lecture materials and assignments covered during the semester.

Lecture -1 Outline

- What is Database and DBMS?
- Characteristics of DBMS
-

What is a Database

Database is a collection of related data.

Examples : Student Database of the University, Customer database of a Bank

What is a DBMS (**Database Management System)**

Set of programs to access the data.

Is a software package designed to create and maintain databases.

What is a DBMS (Database Management System)

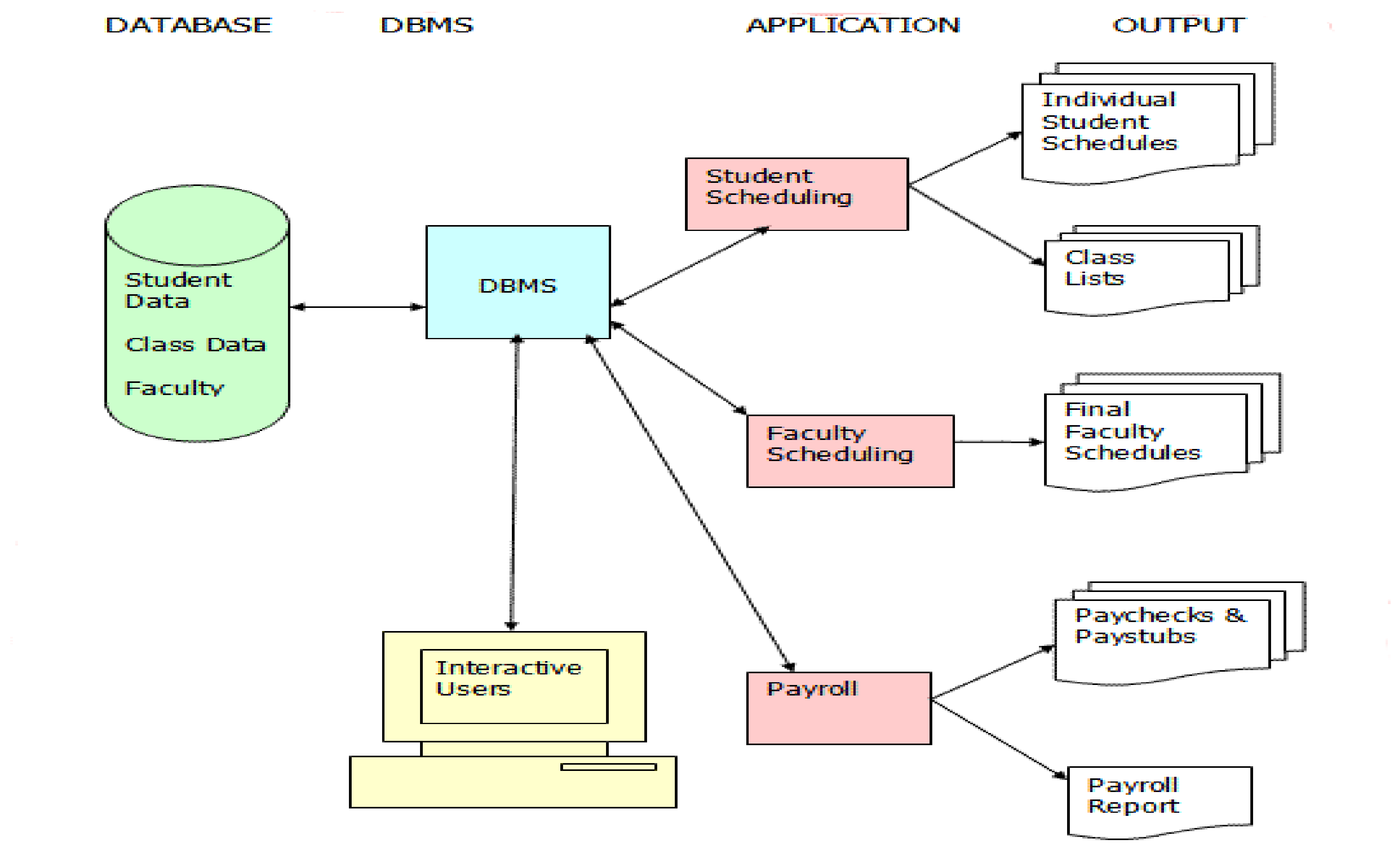


Figure 1.4 The Integrated Database Environment

The Evolution of the Database Modelling Techniques

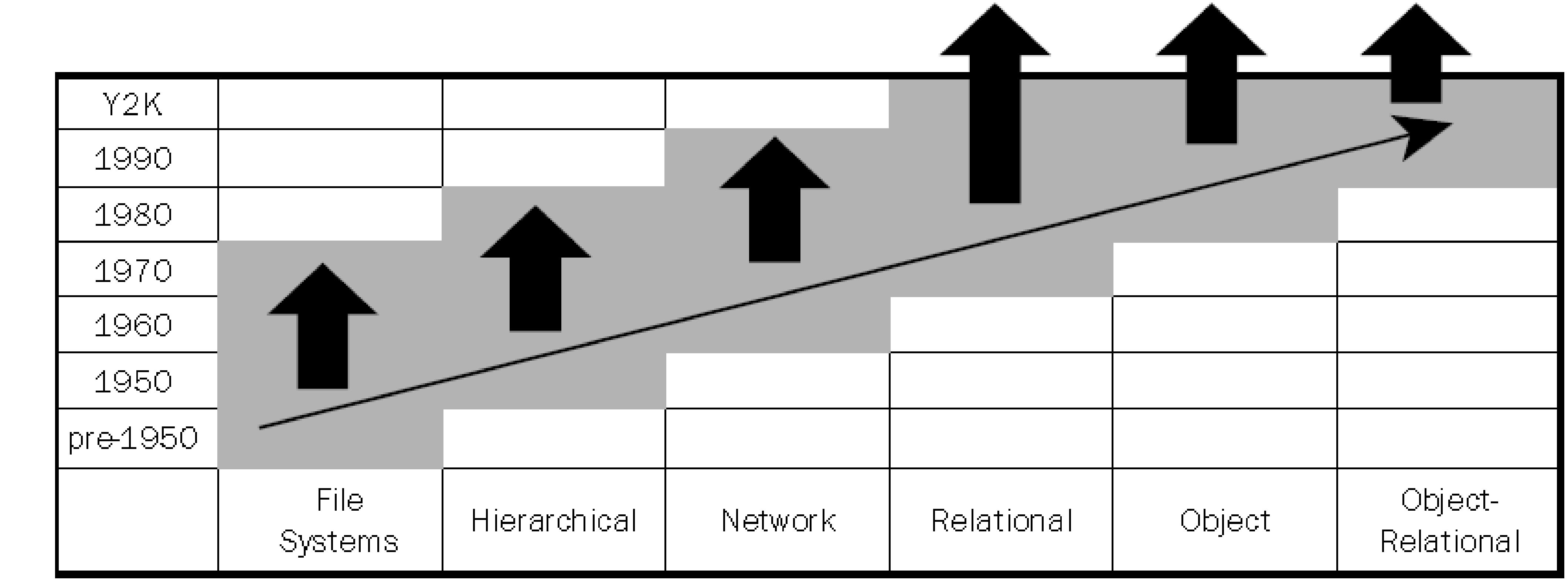
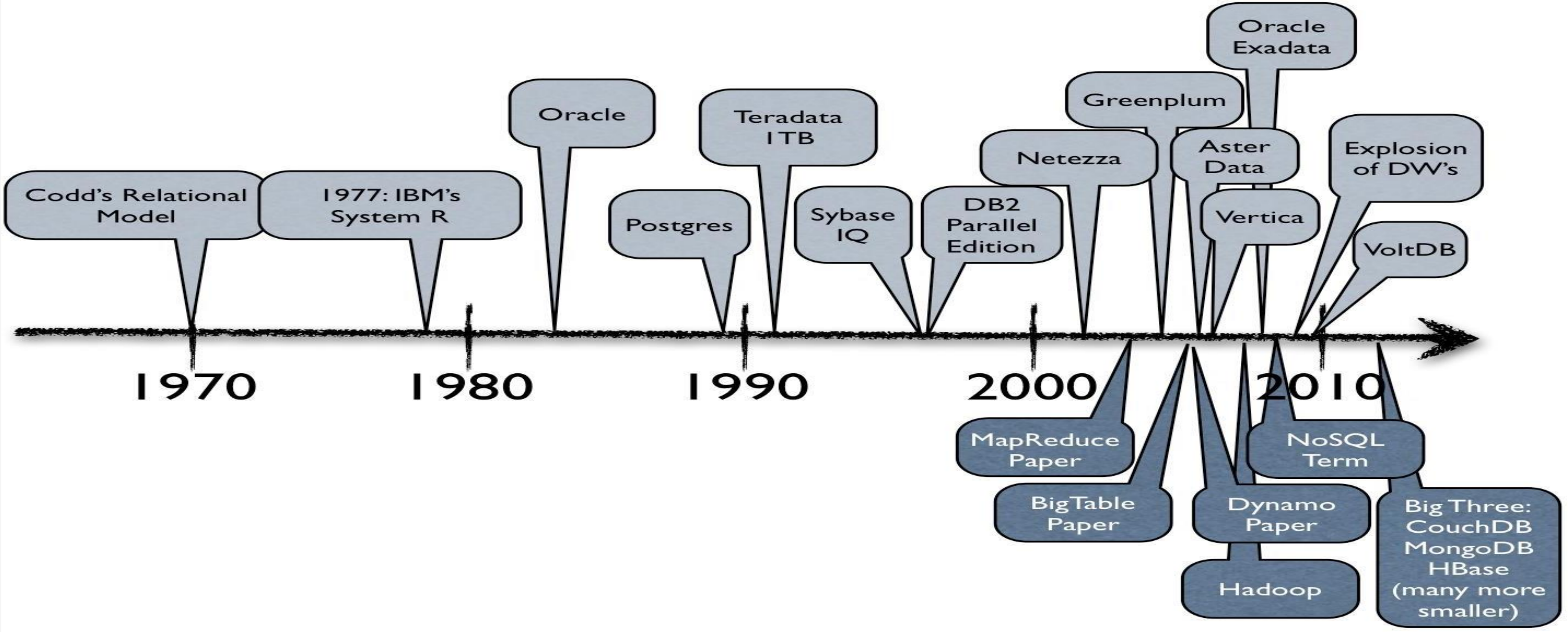


Figure 1-3: The evolution of database modeling techniques

The Evolution of the Database Modelling Techniques



SQL Databases

Evolution of DBMSs… (contd.)

- DBMSs are now an essential part of any Enterprise Information System
- Today, extending DBMSs in a variety of ways…
  - ✓ Storing and querying complex data types
  - ✓ Programmability
  - ✓ Etc.

Why a DBMS?

- Data independence and efficient access
- Reduced application development time
- Data integrity and security
- Uniform data administration
- Concurrent access and recovery from crashes

Why a DBMS?

- Today, large amounts of information are stored & accessed within Information Systems
  - ✓ E.g. : Airline Reservation System, Banking System
- The applications mentioned above require
  - ✓ Storing large amounts of data
  - ✓ Fast access to data
  - ✓ Concurrent access to multiple users
  - ✓ Security
  - ✓ Reliability & Transaction Processing

Why a DBMS? (Contd.)

- Operating System's files provide the basic means of storing information.
  - ✓ Files provide sequential access with only basic security capabilities (read, write, execute)
- They are sufficient for applications requiring small amounts of data & mainly single-user applications.
- DBMS provide advanced features desirable for such applications:
  - ✓ Data independence
    - What is meant by data independence?
  - ✓ Efficient Data Access
    - Why is it efficient to use a DBMS?
  - ✓ Data Integrity
    - What is meant by data integrity? How is it maintained by a DBMS?

Why a DBMS? (Contd.)

- Reduce data redundancy and inconsistency
  - ✓ Same information may be duplicated in several places
  - ✓ All copies may not be updated properly
- Difficulty in new programs to carryout new tasks
- Provide the maximum security to the data
  - ✓ Can set the security levels and only oratorized users   can access the database
  - ✓ Keep the data integrity

Characteristics of Modern DBMS

- Sharing of data
- Redundancy control
- Security control
- Persistent storage
- Database inference
- Multiple user interfaces
- Represent complex relationships
- Integrity enforcement
- Backup and recovery
- Standards enforcement
- Flexibility
- Availability of the data
- Economies of scale
- Self describing: a catalog of meta-data
- Program-data independence
- Multiple views
- Concurrency, distributed, on-line
- Heterogonous HW and SW

Characteristics of Modern DBMS (Contd.)

- System requirements of high availability, reliability, throughput, response time, lifetime, security
- Varied content: numbers, strings, multimedia, GIS, etc.
- Very large databases (DNA sequences, GIS data, Internet of Things IoT, Google, Twitter, etc.)

Database Design Process

1. Requirements Analysis

   What does the user want?

2. Conceptual Database Design

   Defining the entities and attributes, and the relationships between them --> The ER model

3. Logical Database Design

   Map ER to Relational Schema

4. Schema Refinement

   Fine tune

5. Physical Database Design

   Implementation of the design using a Database Management System

6. Security Design

   Implement Controls to ensure security and integrity

1. User Requirement Analysis

Requirements Analysis is the process of determining what the database is to be used for. It involves interviews with user groups and other stakeholders to identify what functionality they require from the database, what kinds of data they wish to process and the most frequently performed operations.

1. User Requirements Analysis

What users want from the database?

- What is going to be stored in the database?
- What applications are going to be built on top the database?
- What are the most frequently asked queries?

1. Requirement Analysis (Example -1)

- **What does the user want ?**

❖ In this process, the data requirements are obtained (may be through series of interviews with the customer) and documented

❖ For example, let us consider a database for Faculty of Technology (FOT)

  ✓ Students want to register to the degree programs

  ✓ Faculty wants to store & retrieve grades of students

  ✓ Administration wants to print out transcripts, organize timetables for students and categorize the students according to the scholarship type (Mahapola or any other)

1. Requirement Analysis (Contd.)

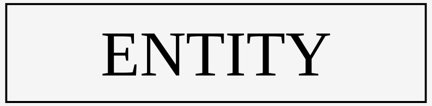
- What are the data to be stored?
  - ✓ Students, grades, Lecturers
- What are the frequent queries?
  - ✓ Grades of a student
- What are the frequent updates?
  - ✓ Updating Cumulative GPA
- What are the different data different users need to access?.....

2. Conceptual Design

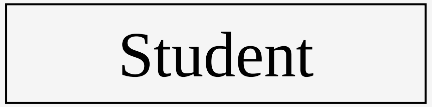
- In this step, a high-level conceptual model is designed encapsulating the data requirements
- **Entity-Relationship (ER)** Model is a very popular & widely used conceptual model
- ER Model is enhanced with object-oriented features & known as the **Enhanced Entity-Relationship (E-ER) Model**

ER Model - Entity

- An **entity** is an object in the real world
- A collection of similar entities is called an **entity set**.
- Graphically,

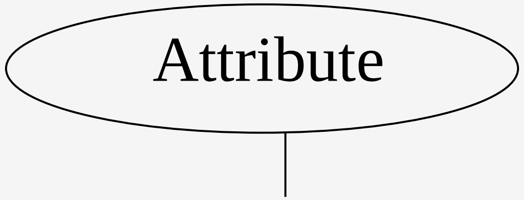


Eg.

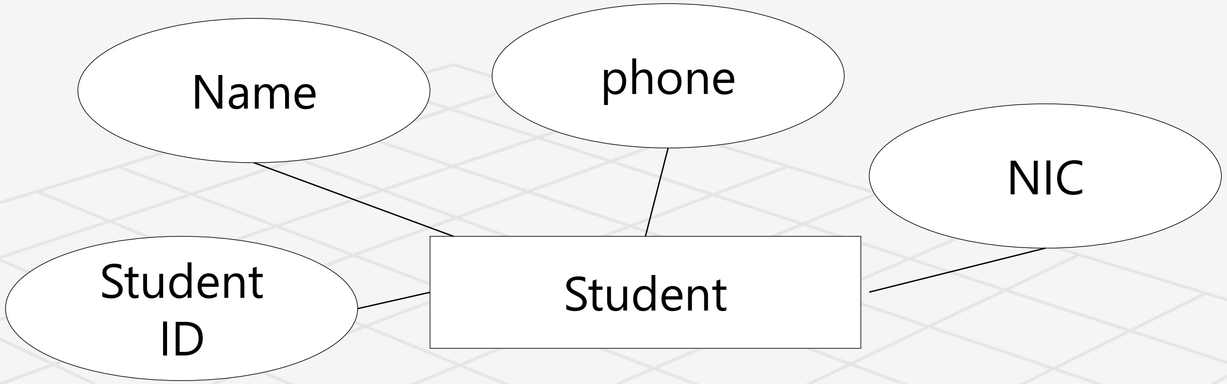


ER Model - Attributes

- An entity is described using a set of **attributes**.
- Graphically represented as

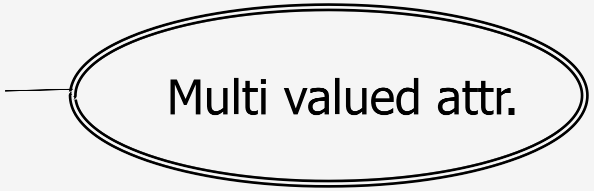


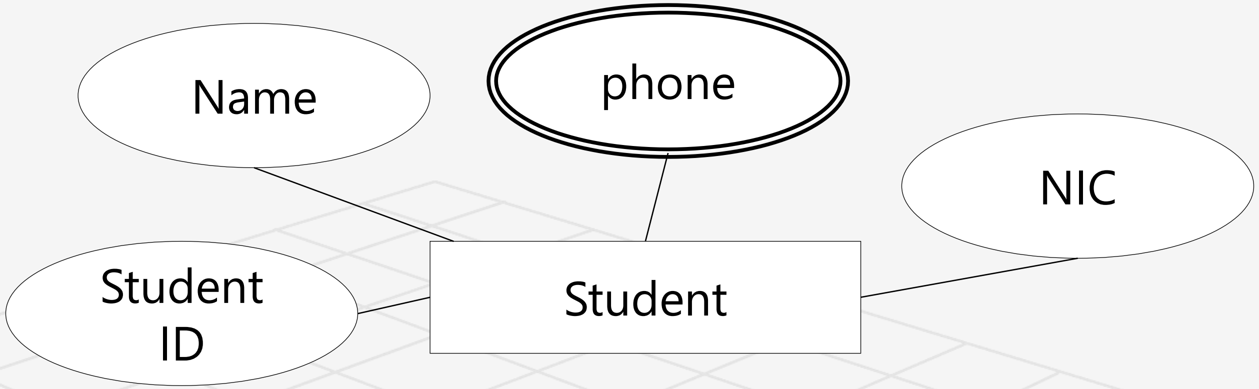
- For each attribute, a **domain** of possible values for the attribute needs to be defined.



ER Model – Multivalued Attributes

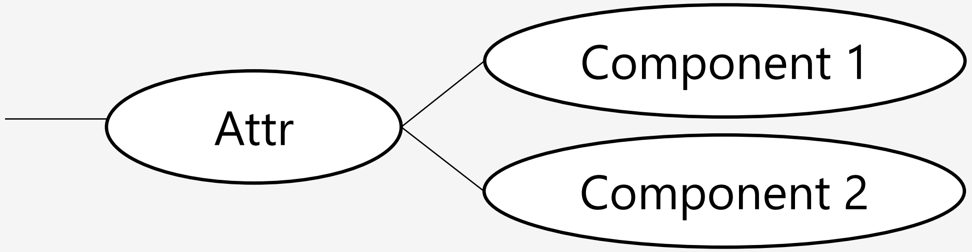
- Attributes containing multiple values are called **multivalued attributes.**
- Graphically,

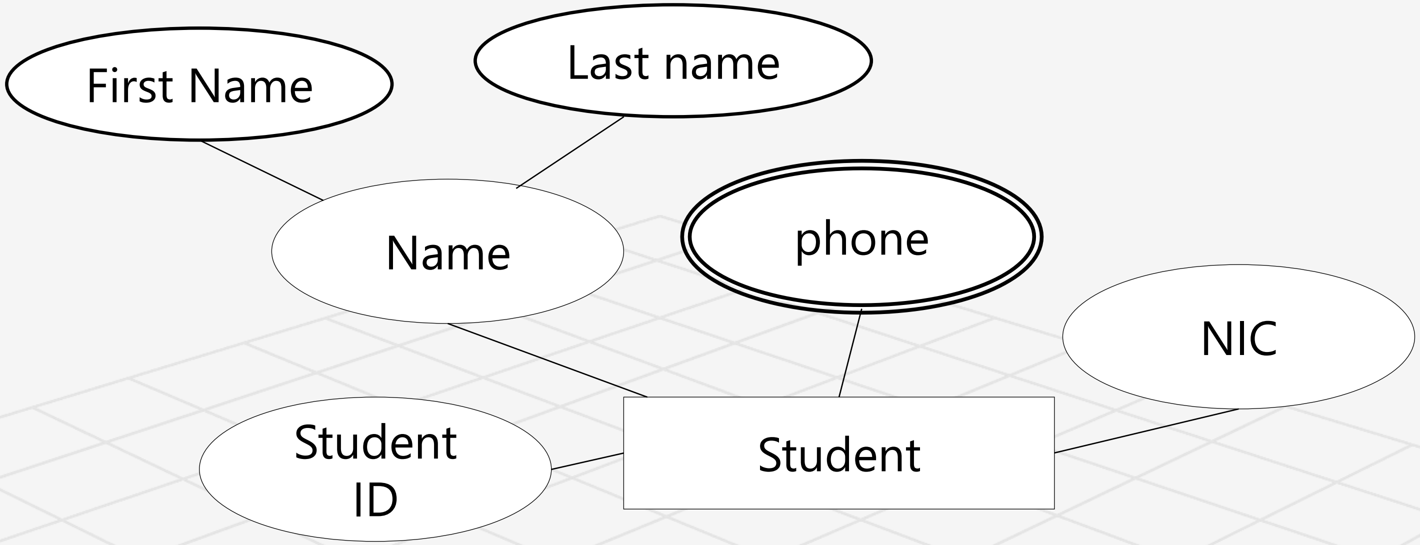




ER Model – Composite Attribute

- An attribute considered composite if it comprises **multiple components.**
- Graphically



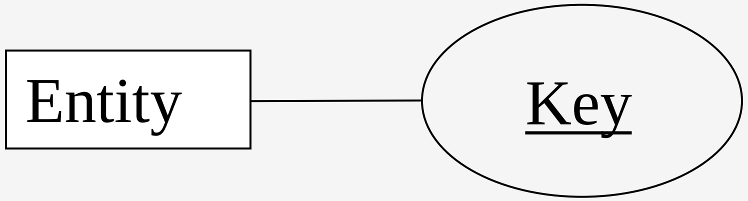


ER Model - Keys

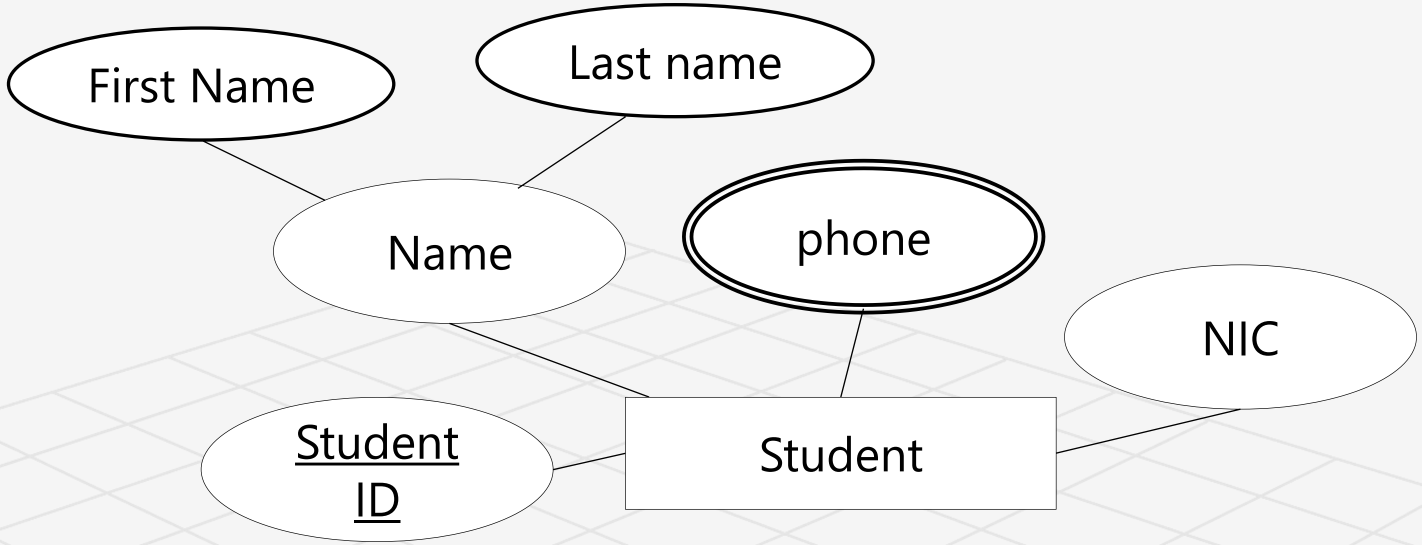
- A **key** is a minimal set of attributes whose values uniquely identify an entity in the set.
- There can be multiple key attributes called **candidate keys** in a single entity (eg. Student No, NIC)
- A single candidate key is designated as the **primary key.**

ER Model – Key (Contd.)

- Graphically,
- Primary key is underlined

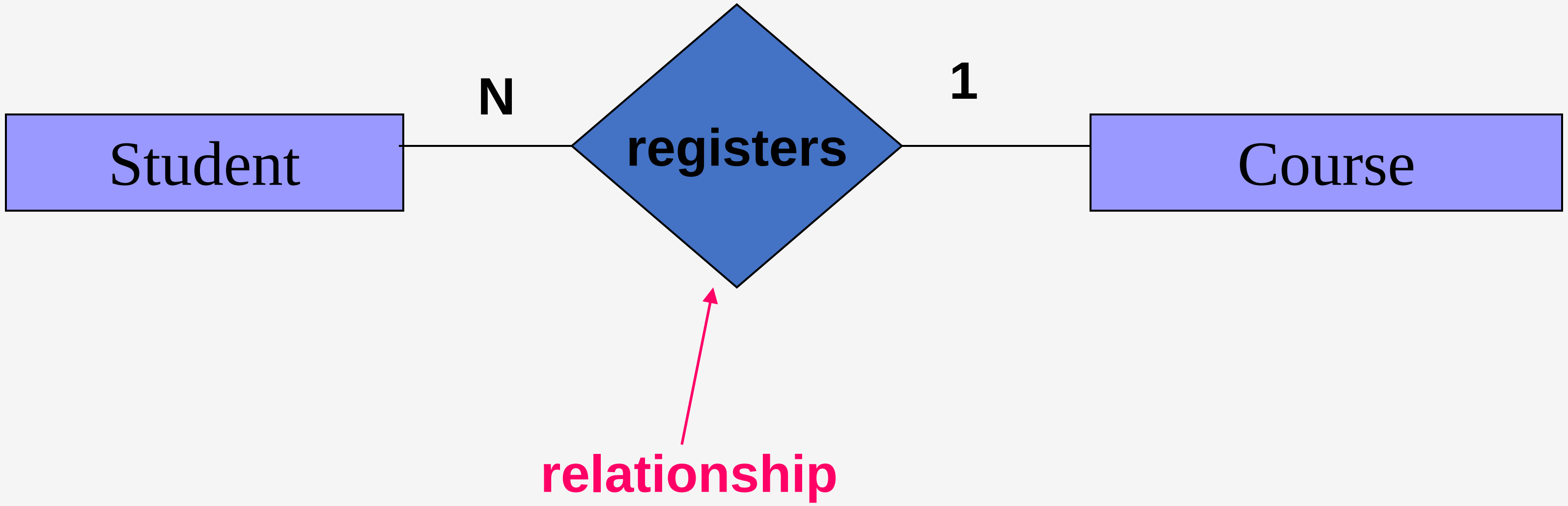


- Sometimes, a group of attributes make up the key. This is called a **composite key**.

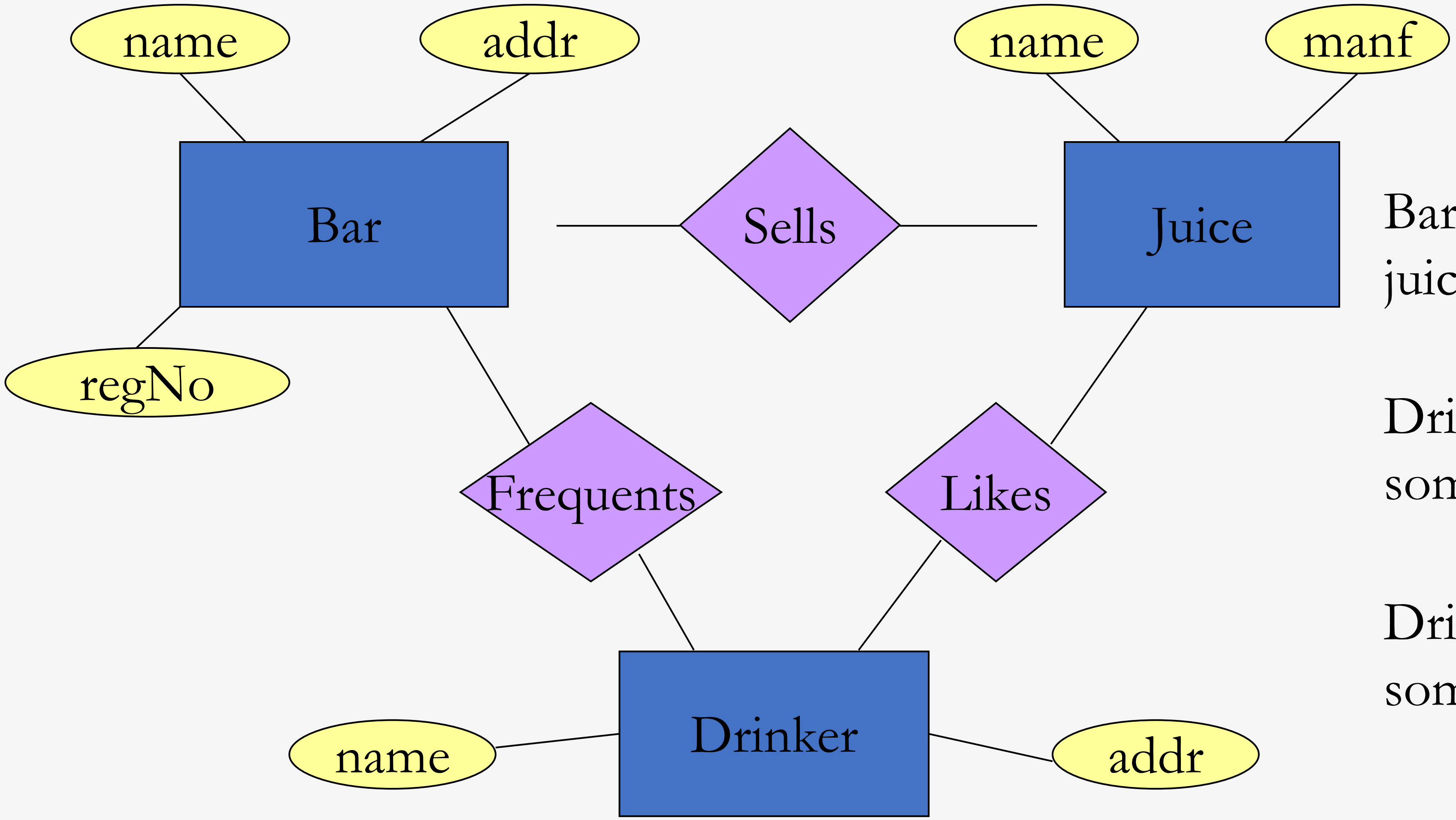


Relationship

- **Relationship** is an association among two or more entities.
- Collection of similar relationships - **relationship set** .
- Shown as a line connecting the associated entities, labelled with the name of the relationship.
  - ✓ Normally a relationship is named using a ‘verb’ or ‘verb group’.



Example: Relationships



Bars sell some juices.

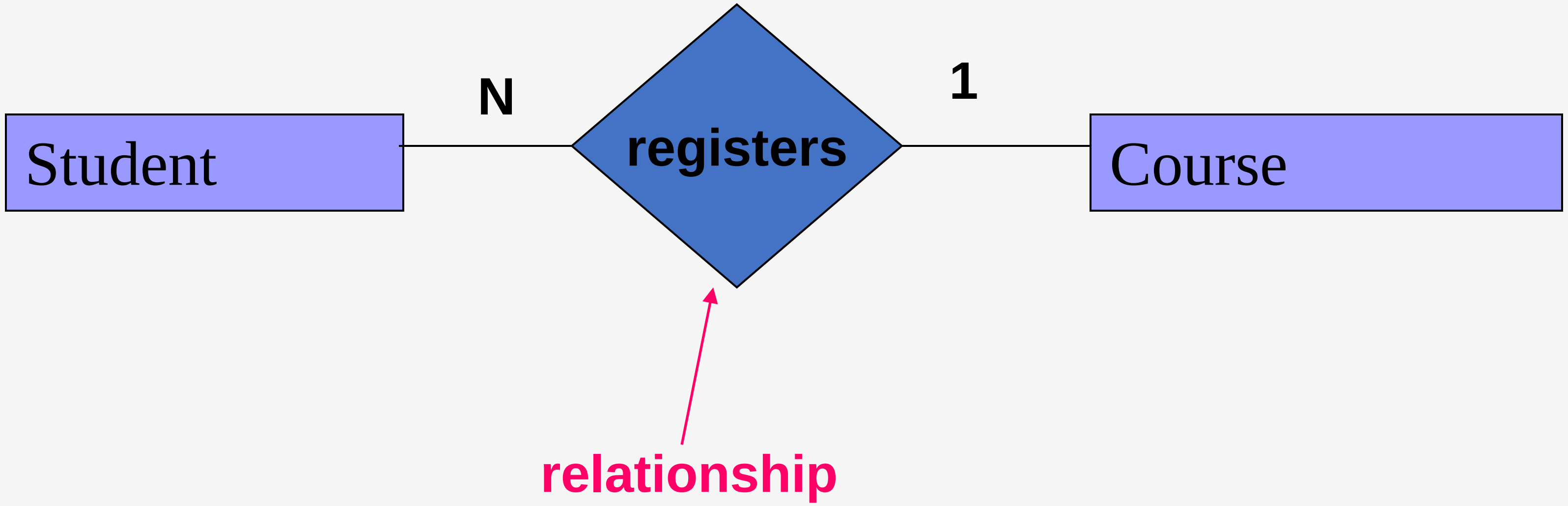
Drinkers like some juices.

Drinkers frequent some bars.

Degree of a Relationship

Degree of a relationship is the number of participating entities in the relationship.

- Degree / Number of Entities = 2 Or binary



Degree of Relationship

- ❖ No of participating entities
- ❖ Relationships can be classified based on their degree into

✓**Binary – relationship with <u>two</u> participants-**

Degree/No of Entities = 2

✓**Ternary – relationship with <u>three</u> participants**

Degree/No of Entities = 3

✓**Quaternary – relationship with <u>four</u> participants**

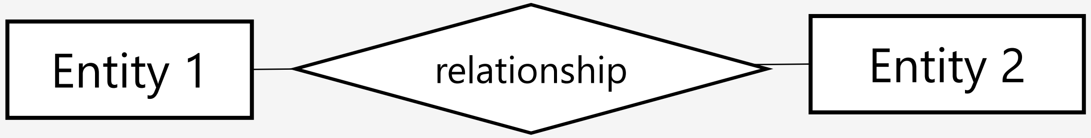
Degree/No of Entities = 4

Ternary Relationships

- Sometimes, we need a relationship that connects more than two entity sets.
- Suppose that drinkers will only drink certain juices at certain bars.
  - ✓ Our three binary relationships Likes, Sells, and Frequents do not allow us to make this distinction.
  - ✓ But a 3-way/ ternary relationship would.

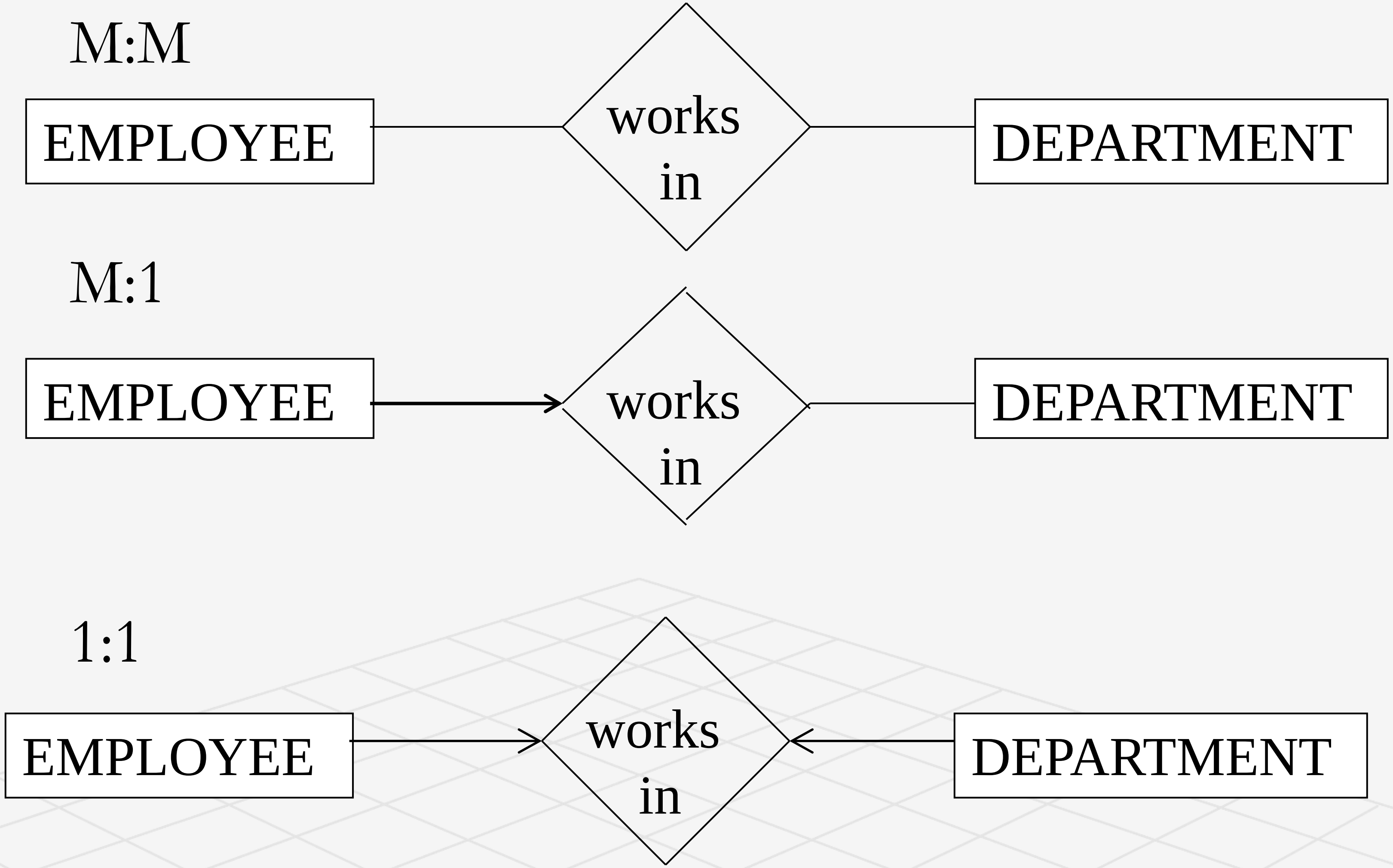
ER Model - Relationships

- A relationship is an **association** among two or more entities
- Graphically,



- The **cardinality ratio** for a binary relationship specifies the number of relationship instances that an entity can participate in.
- There are three types of cardinality ratios for binary relationships. They are **one-to-one, one-to-many (many-to-one)** and **many-to-many.**

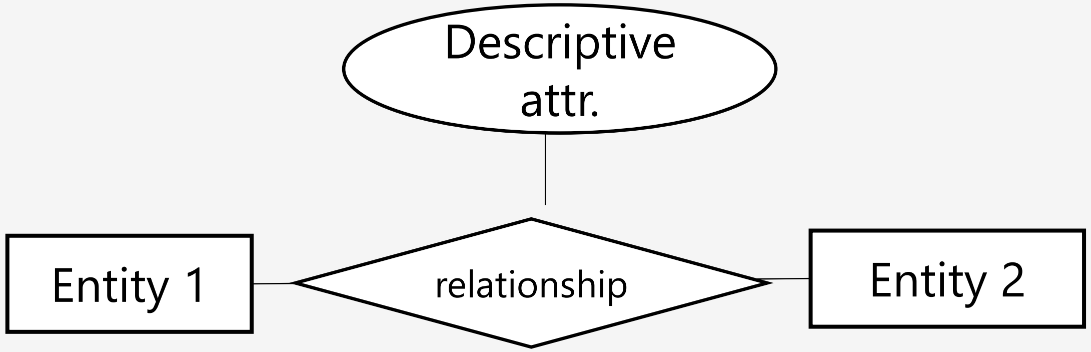
ER Model – Relationships (contd.)



ER Model – Relationships (contd.)

- A relationship can have **descriptive attributes** – used to record information about the relationship.

- Graphically,



Exercise 1

- A Library is organized into several sections such as fiction, children and technology. Each section has a name and a number(unique) and its headed by a head librarian.

- Each book belong to a section and has a title, authors, ISBN, year and a publisher.

- A book may have several copies. Each copy id identified by an access number.

- For each copy rented, current borrower and due date should be tracked.

- Members have a membership number, an address and a phone number.

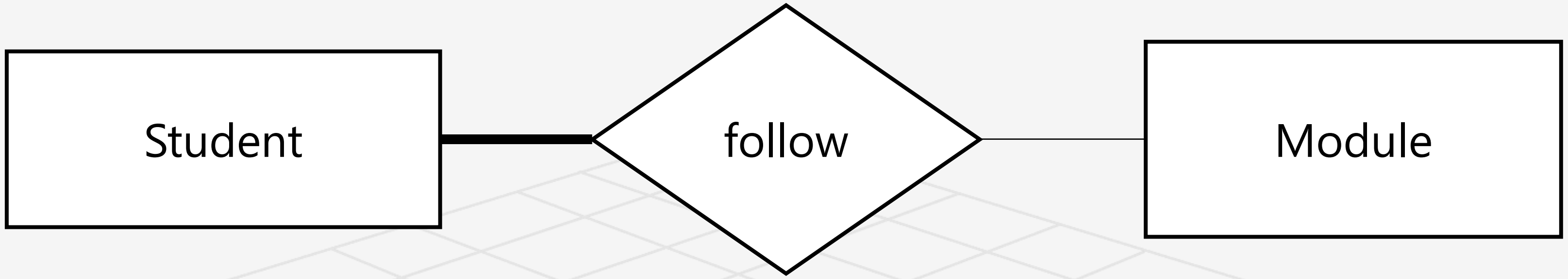
- Members can borrow 5 books and could put hold request on a book

- Librarian has a name, id number, phone and an address.

- Draw an ER diagram for the above scenario.

ER Model – Participation Constraints

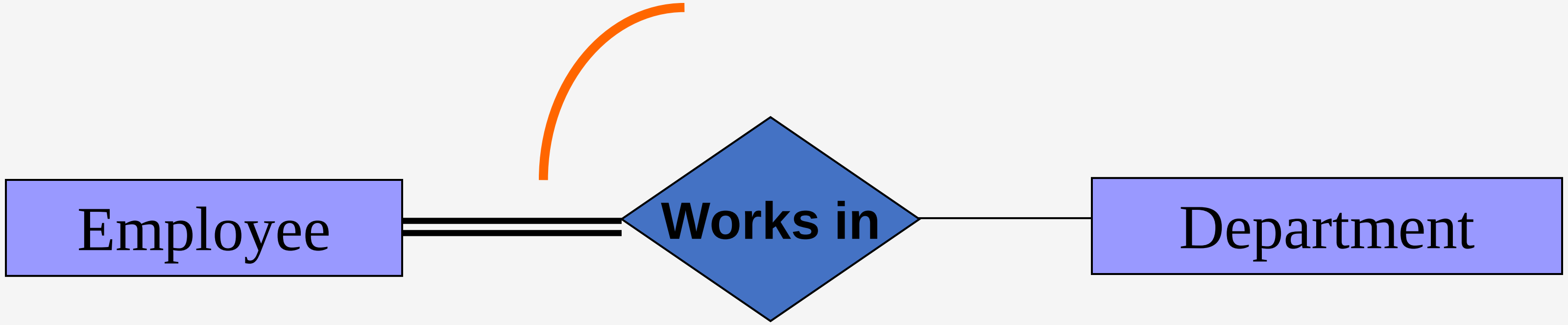
- Participation constraint specifies the fact that the relationship is required for every entity of the entity set.
- The relationship set has either total or partial participation
- Graphically, bold line represents a total participation



Participation Constraints

- For example, if we specify that an employee **must** always work for a department.
- Then we say that the relationship "works in" is in **total participation** from employee entity to department entity.

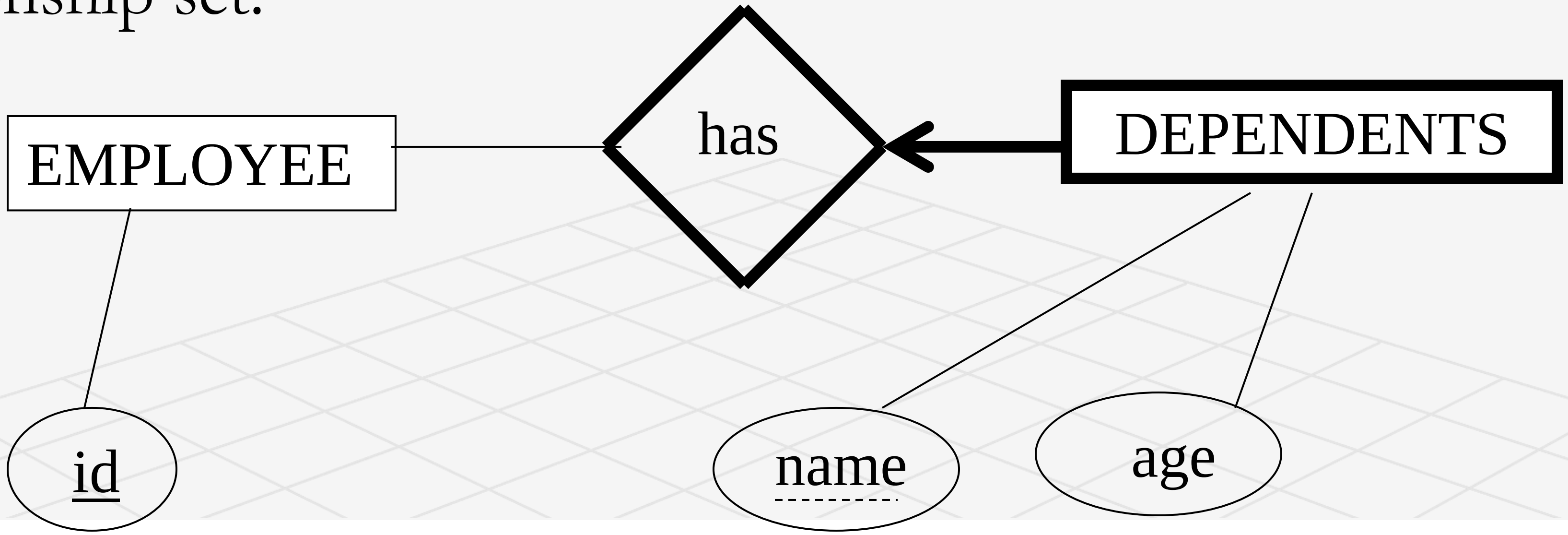
(double lines) → total participation



Employee must work in a department
Department may or may not have employees

ER Model – Weak Entity

- An entity is weak if the entity does not have a key and satisfies the following conditions:
  - ✓ The owner entity set and the weak entity set must participate in a one-to-many relationship
  - ✓ The weak entity set must have total participation in the identifying relationship set.



What are the difficulties in ER Model

Find some difficulties you face when you are going to design ER Model

End of the Lecture – 1
Thank You