ICT 134-2 Database Management Systems

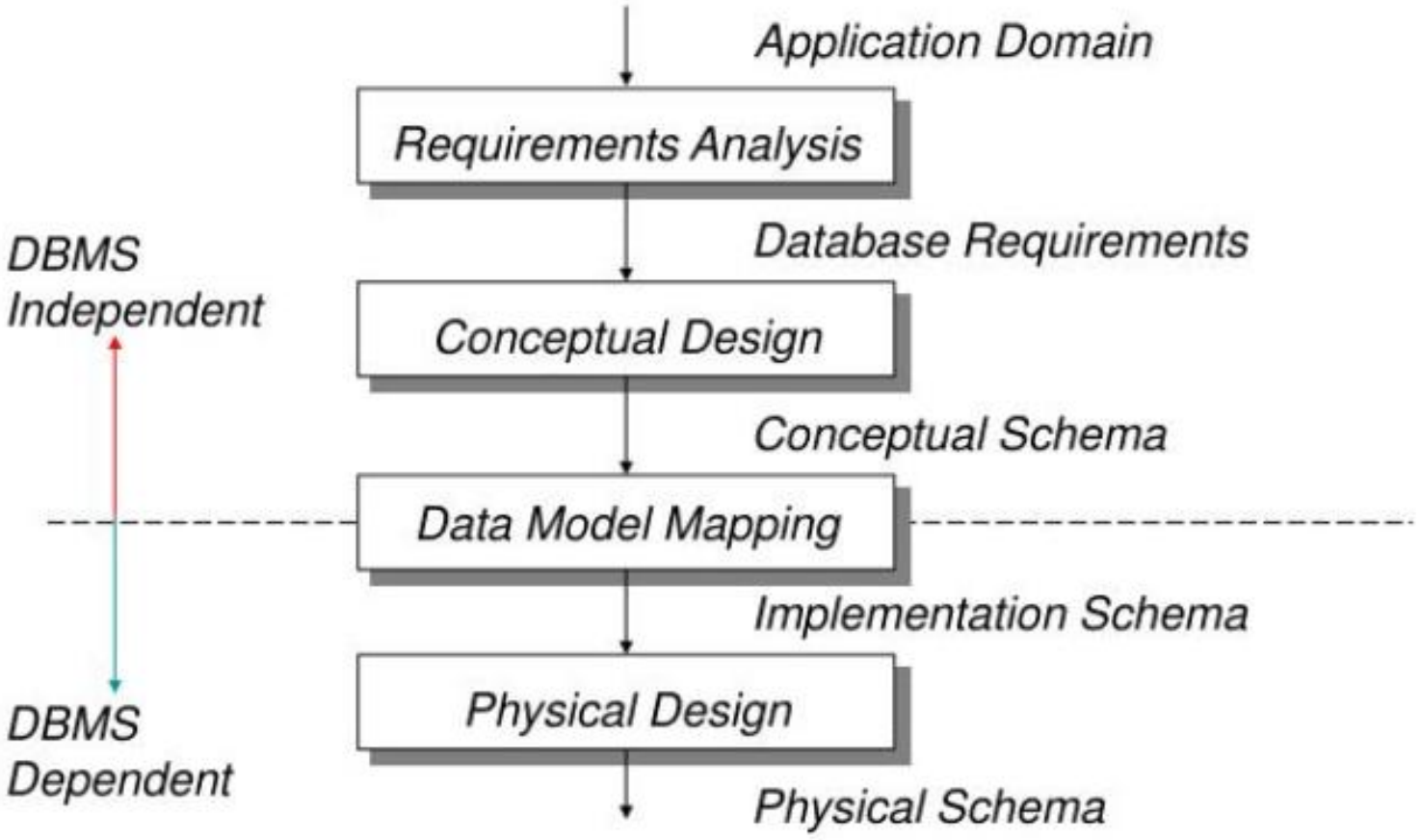
Mrs. W.B.M.S.C. Wijayakoon
B.Sc., M.Eng., D.Eng.(Reading)
Lecturer (Probationary)/ BICT Degree Program
Faculty of Technological Studies, UWU

Lecture 4 – Data Modelling using Entity-Relationship (ER) Model – Part 1

Lecture 4 - Structure

- Phases of Database Design
- Conceptual Modelling
- Abstractions in Conceptual Design
- Example Database Requirements
- ER Diagram
  - Entities, Attributes and Relationships
  - Participation, Cardinality and Keys

Phases of Database Design



Phases of Database Design

The database design process can be broken down into four phases

Phase 1 – Requirements Analysis

Phase 2 – Conceptual Design

Phase 3 – Data Model Mapping/ Logical Design

Phase 4 – Physical Design

A Data Modelling Process

- Steps in the data modelling process
  - Plan project
  - Determine requirements
  - Specify entities
  - Specify relationships
  - Determine identifiers
  - Specify attributes
  - Specify domains
  - Validate model

Planning the Project

- Obtaining project authorization and budget
- Building the project team
- Planning the team's activities
- Establishing tools, techniques, and standards for consistent results
- Defining the project's scope

Determining System Requirements

- Sources for data modelling requirements
  - User interviews and user activity observations
  - Existing forms and reports
  - New forms and reports
  - Existing manual files
  - Existing computer files/ databases
  - Formally defined interfaces (XML)
  - Domain expertise
- The result of the requirements determination will be a repository of notes, diagram, forms, reports, files, etc., that can be used to develop the data model

Specifying Entities

- An entity is something that the users want to track; something the users want to keep data about
- Entities
  - Can be physical things or logical concepts
  - Are identifiable; you can tell one from another
  - Are things described by nouns, not characteristics described by adjectives

Specifying Relationships

- Includes:
  - Identity of the parent and child entities
  - Relationship type
  - Minimum and maximum cardinalities
  - Name of the relationships
- Two techniques:
  - Examine whether a relationship exists between every combination of two entities
  - Locate relationships from requirement documentations
- A combination of two approaches may be used

Determining Identifiers

Identifier is an attribute or group of attributes that uniquely identifies an entity instance

If there is difficulty specifying an identifier, maybe:

- It should be part of a different entity
- It is a subtype or category of a common entity
- It needs one or more identifying relationships

Specifying Attributes and Domains

- Find attributes on forms, reports, existing files, etc…, and add them to entities
- Determine whether the attribute has already defined a domain
  - If so, the attribute is based upon that domain
  - If not, a new domain is defined
- Review the domains and make adjustments as necessary
- Domain property inheritance: when the domain properties change, all the attribute properties change as well
- Domain may be enforce data standards promoting compatible data types and systems
- Once all attributes have been specified the model should be reviewed for missing entities

Validate Model

- Data model is a model of humans' model, not a model of reality
- A data model is wrong if it does not accurately reflect the ways the users think about their world
- Data models are validated through a series of reviews
  - Normally, a team review is followed by user reviews
- ER model as well as prototypes of forms and reports may be used to communicate to users features of the data model

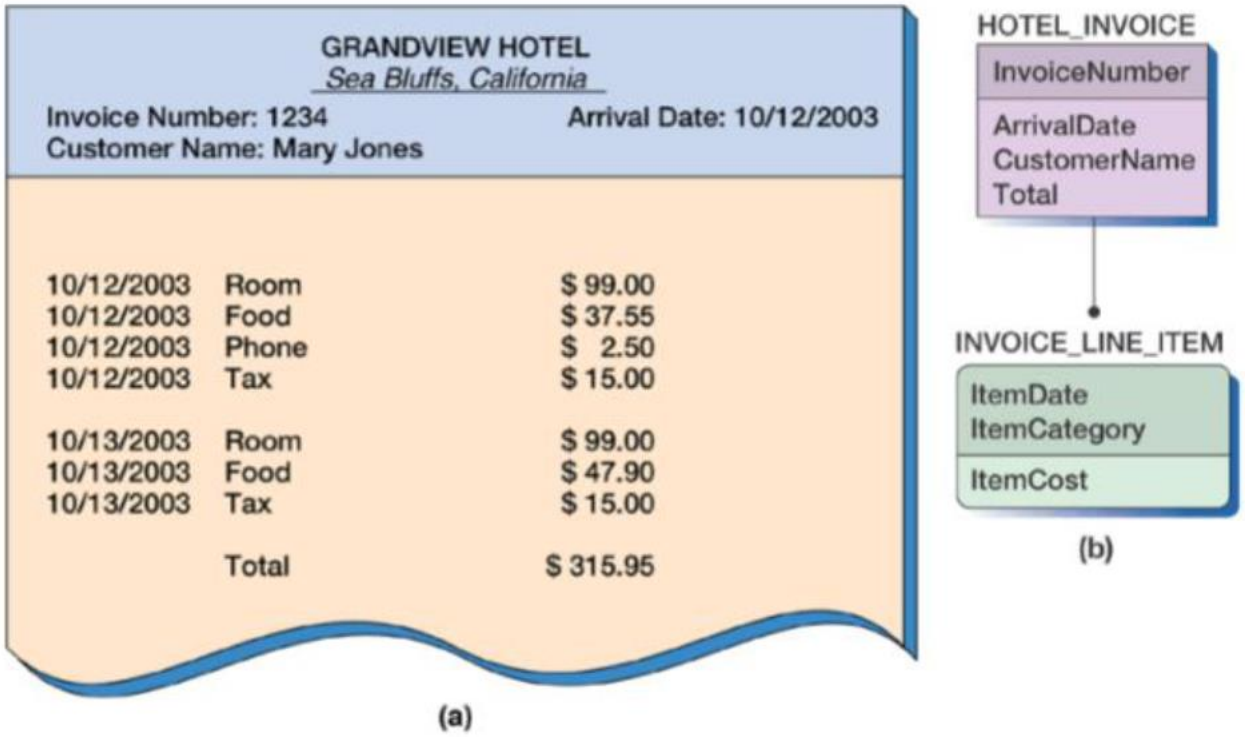
Create Data Models from Forms and Reports

- Example:
- Single entities

(a) Report Showing Need for Single Entity and
(b) Entity for the Report in (a)

Example: Identifying connection relationships

(a) Sample Invoice and (b) Identifying Connection Relationship



Example: Repeating groups

- Hotel Invoice with two repeating groups
- Two Identifying Connection Relationships

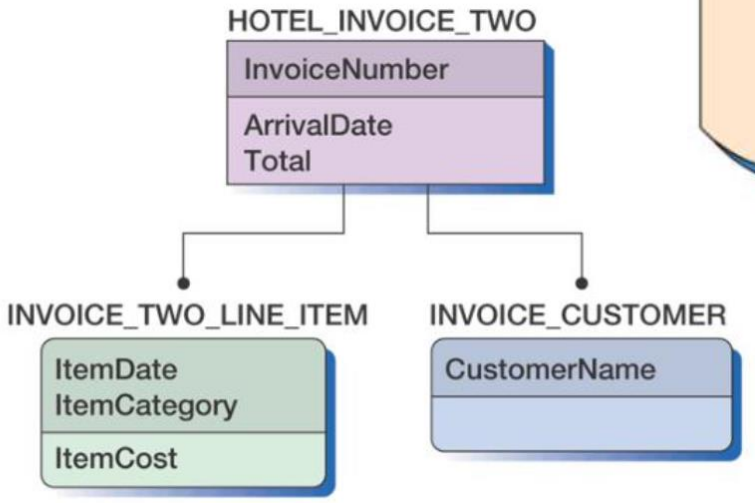
Sample Invoice

GRANDVIEW HOTEL
Sea Bluffs, California

Invoice Number: 1234 Arrival Date: 10/12/2003
Customer Name: Mary Jones
Fred Jones
Sally Jones

10/12/2003	Room	$ 99.00
10/12/2003	Food	$ 37.55
10/12/2003	Phone	$ 2.50
10/12/2003	Tax	$ 15.00
10/13/2003	Room	$ 99.00
10/13/2003	Food	$ 47.90
10/13/2003	Tax	$ 15.00
	Total	$ 315.95

(a)

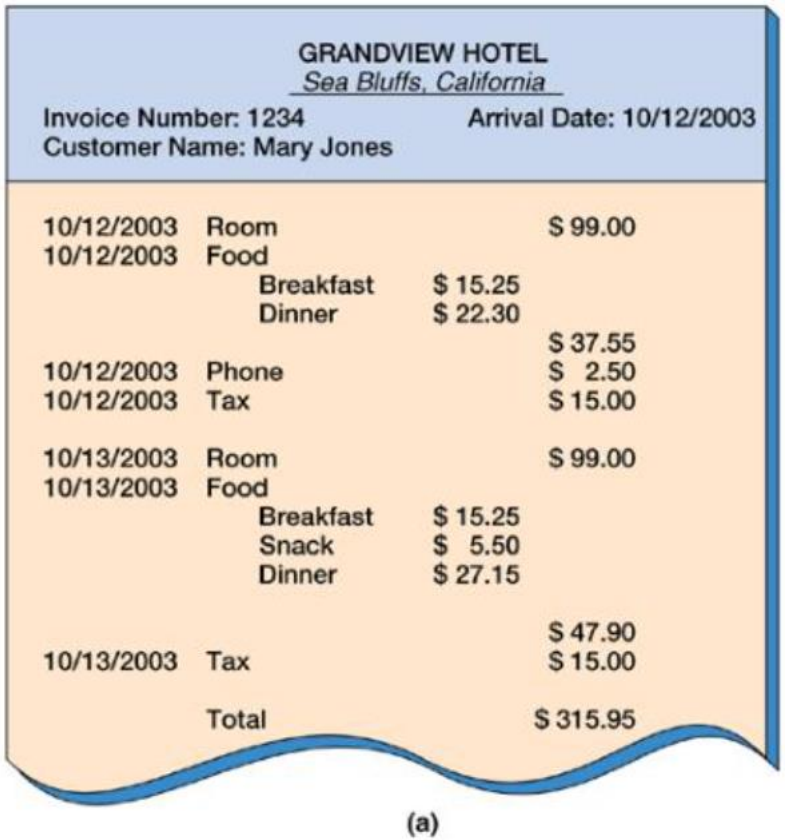


(b)

Example: Nested groups

- Hotel Invoice with nested groups

(a) Sample Invoice and (b) Nested Identifying Connection Relationships



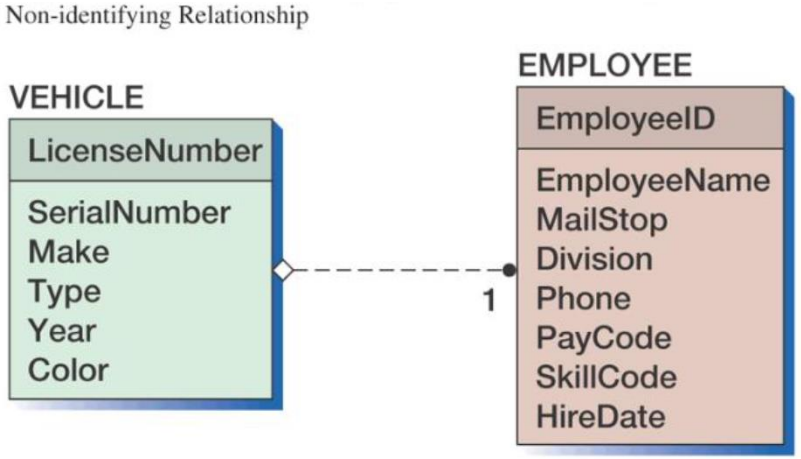
Example: Non-Identifying Connection Relationships

- 1:1 Non-Identifying Connection Relationships - Sample Forms

VEHICLE DATA			
License number	Serial number		
Make	Type	Year	Color
Employee assignment			

EMPLOYEE WORK DATA			
Employee name		Employee ID	
MailStop		Division	Phone
Pay code	Skill code	Hire date	Auto assigned

(a)



(b)

Example: 1:N

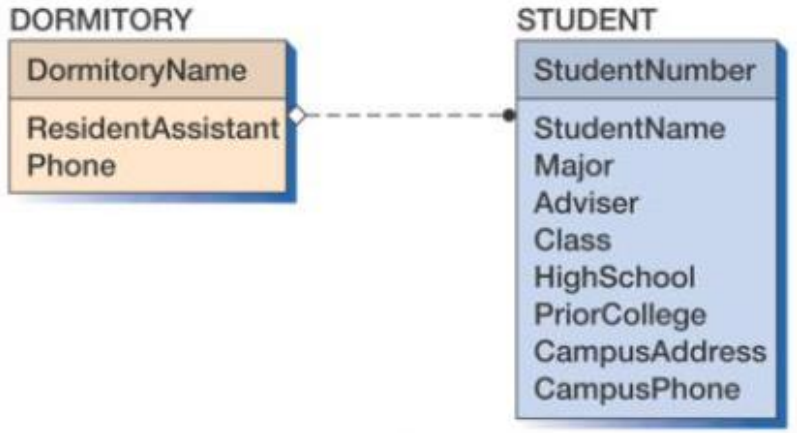
- 1:N Non-Identifying Connection Relationships - Sample Forms

- 1:N Non-Identifying Connection Relationships –Non-identifying Relationship

DORMITORY OCCUPANCY REPORT

Dormitory	Resident Assistant	Phone
Ingersoll	Sarah and Allen French	3-5567

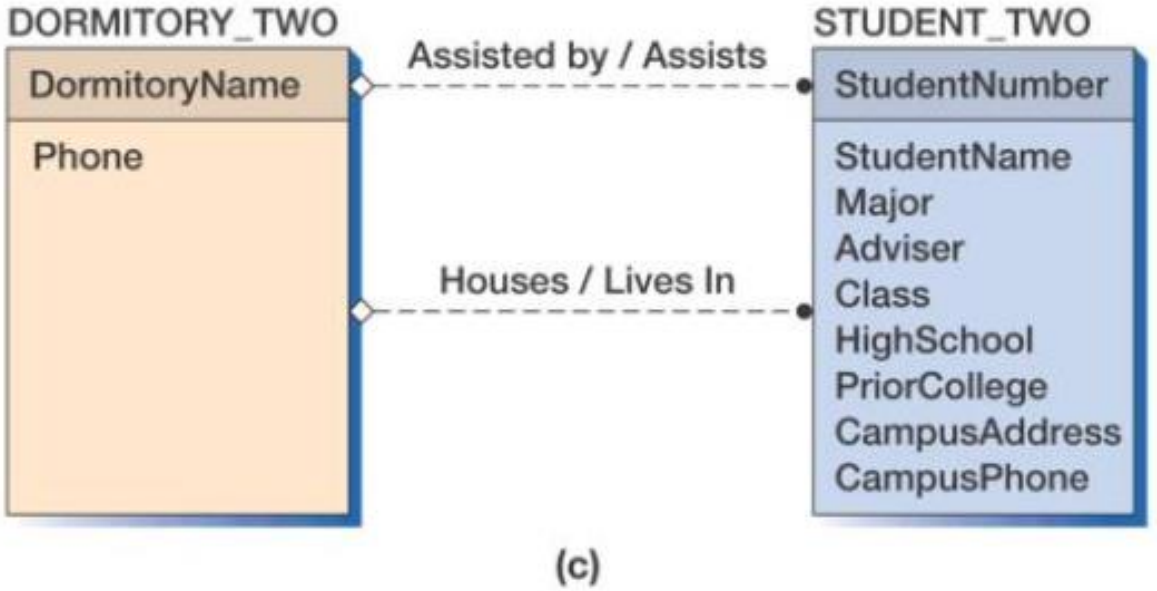
Student name	Student Number	Class
Adams, Elizabeth	710	SO
Baker, Rex	104	FR
Baker, Brydie	744	JN
Charles, Stewart	319	SO
Scott, Sally	447	SO
Taylor, Lynne	810	FR



(b)

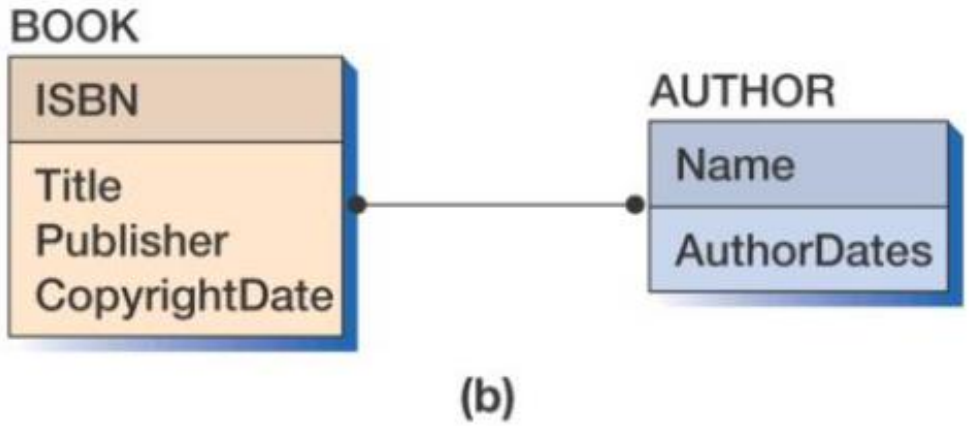
Example: 1:N

- 1:N Non-Identifying Connection Relationships – Using Relationship for Resident Assistant



Example: N:M

- N:M Relationship – Non-Specific Relationship



Conceptual Design

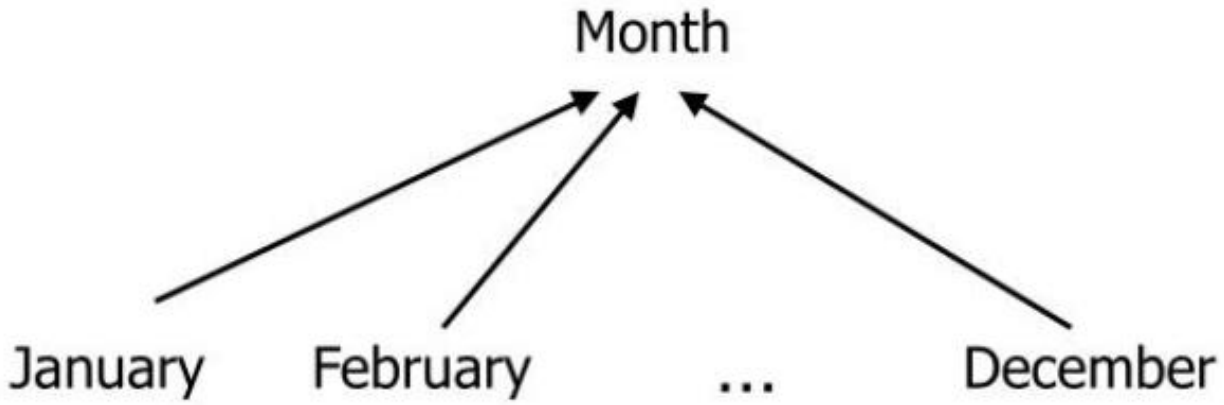
- This is a high level description of the structure of a database
- Concise description of the data requirements of the users and includes detailed descriptions of the data, relationships of the data, relationships and constraints

Abstraction in Conceptual Design

- An abstraction is a mental process where we select some set of properties of an object and exclude others
- 3 types of abstractions
  - Classification
  - Aggregation
  - generalization

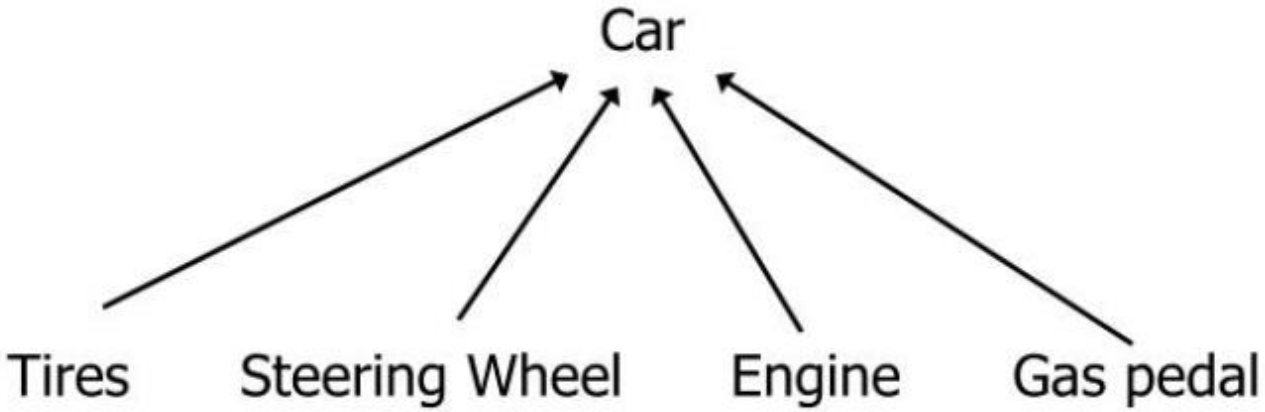
Classification

- Define a class of real-world objects with common properties



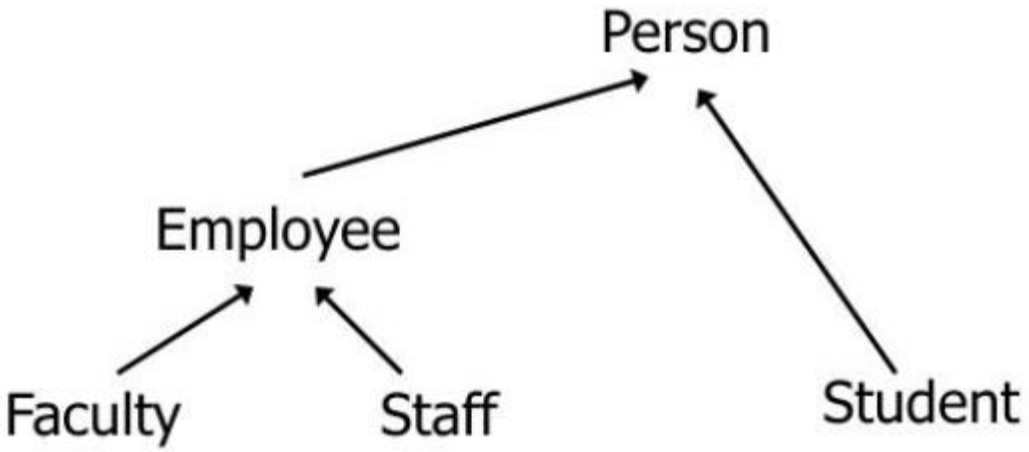
Aggregation

- Define a new class from a set of other classes that represent component parts



Generalization

- Define a subset relationship between elements of two or more classes



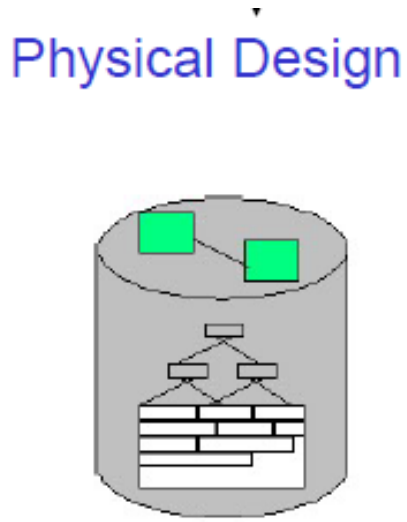
- Will discuss in detail later…

Logical Design

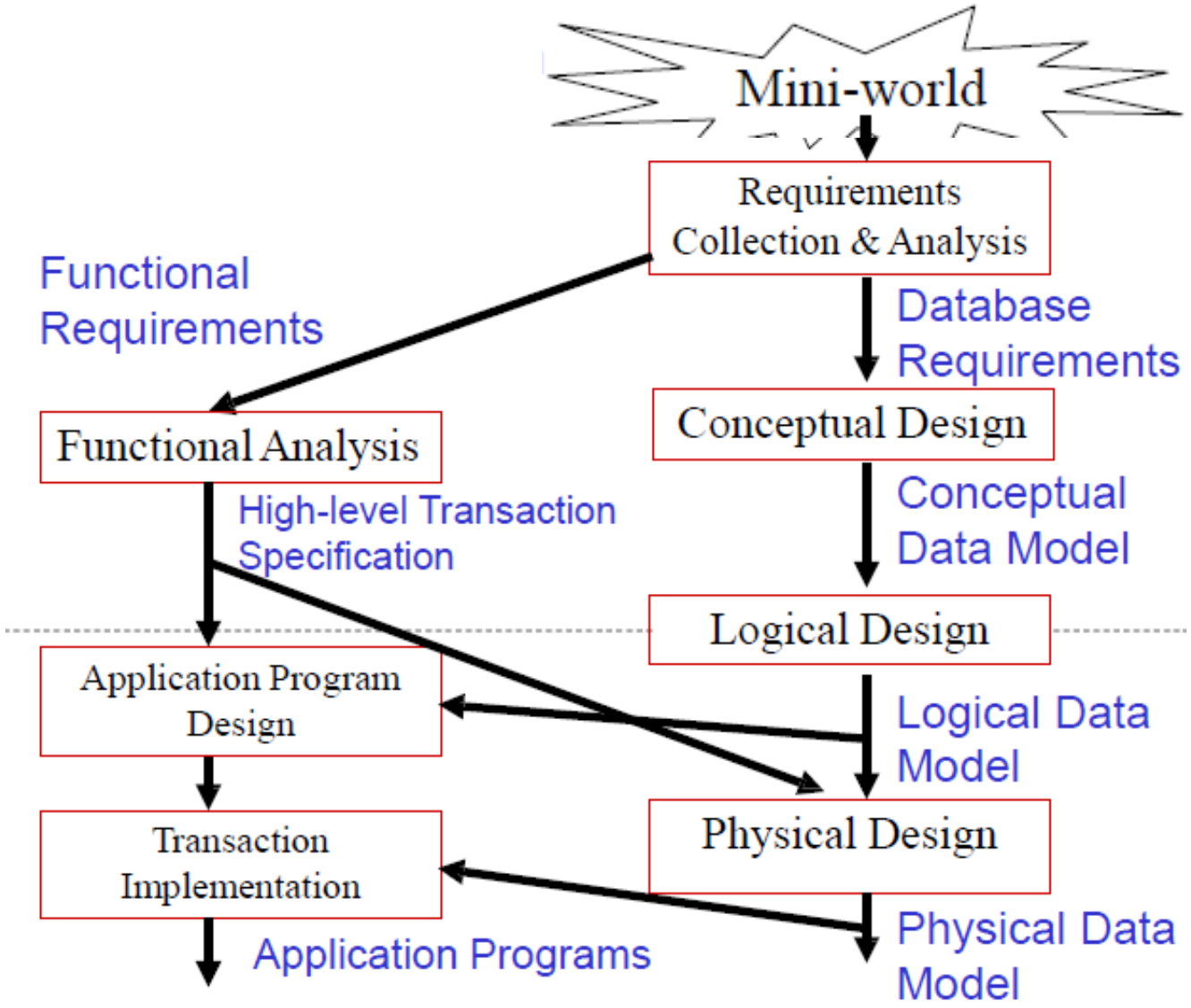
- This is a process of mapping the database structure developed in the previous phase to a particular database model
- E.g.: map ER model to relational
- Specific to a database model, but independent of a particular DBMS (Product)

Physical Design

- This is the process of defining structure that enables the database to be queried in an efficient manner.
- E.g.: index and hash file design, data partition

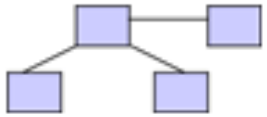


Summary: Phases of Database Design

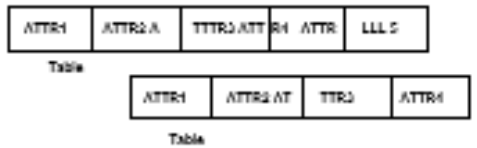


Types of Data Models

- Conceptual Data Model



- Logical Data Model



- Physical Data Model

Conceptual Data Model

- A data model representing the objects and business rules that govern the operation of an organisation
  - Done by a Business Analyst
  - Not constrained by access requirement and technology

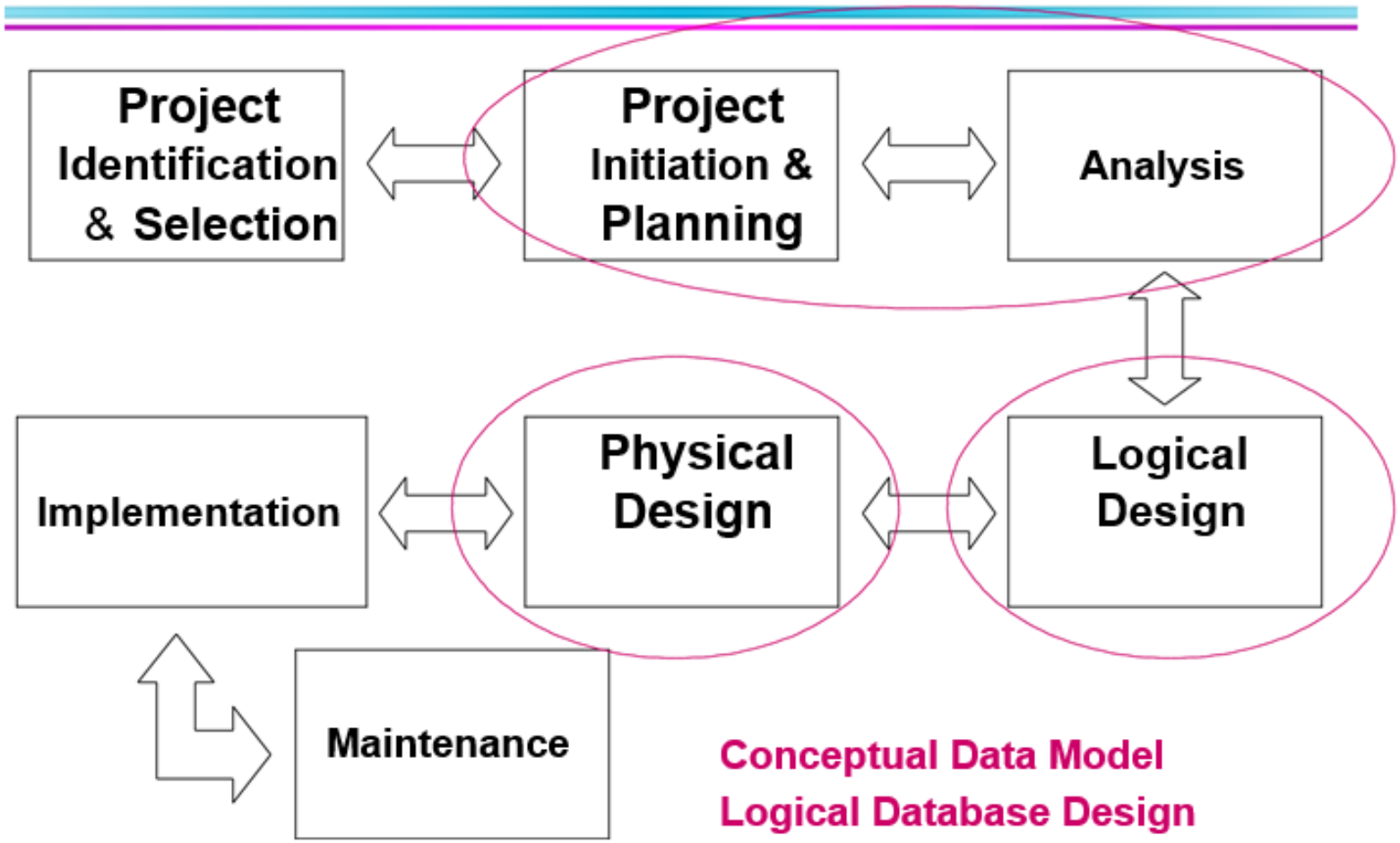
Logical Data Model

- A set of data structures assembled following rules that describe the processing requirements (access paths) of the data in terms of a logical database model
  - Done by a Data Analyst
  - Not constrained by technology

Physical Data Model

- A model prepared for the purpose of implementing a database that runs under the control of a particular DBMS (product)
  - Done by a DBA
  - Constrained by Technology

Systems Development Life Cycle (SDLC)



Conceptual Data Model
Logical Database Design
Physical Database Design

Database Development Activities

- Enterprise Modelling
- Conceptual Data Modelling
- Logical Database Design
- Physical Database Design and Creation
- Database Implementation
- Database Maintenance

Enterprise Modelling

- Analyse current data processing
- Analyse the general business functions and their database needs
- Justify need for new and databases in support of business

Project Identification & Selection

Conceptual Data Modelling

- Identify **scope of database requirements** for the proposed information system
- **Analyse overall data requirements** for business function(s) supported by database
- **Develop preliminary conceptual data model** including entities and relationships
- **Compare** preliminary conceptual data model with enterprise data model

Conceptual Data Modelling

- **Develop detailed conceptual data model,** including all entities, relationships, attributes and business rules
- Make conceptual data model consistent with other models of information system
- Populate repository with all conceptual database specifications

Analysis

Logical Database Design

- Analyze in detail the transactions, forms, displays and inquires (data views) required
- Design by the business functions supported by the database
- Integrate database views into conceptual data model
- Identify data integrity and security requirements, and populate repository

Physical Database Design

- Define database to DBMS (often generated from repository)
- Decide on physical organization of data
- Design database processing programs

Database Implementation

- Code and test database processing programs
- Complete database documentation and training materials
- Install database and convert data from prior systems

Database Maintenance

- Analyse database and database applications to ensure that evolving information requirements are met
- Tune database for improved performance
- Fix errors in database and database applications and recover database when it is contaminated

Entity-Relationship Model

- Most popular conceptual model for database design
- Basic for many other models
- Describes the data in a system and how that data is related
- Describes data as entities, attributes and relationships

Database requirements

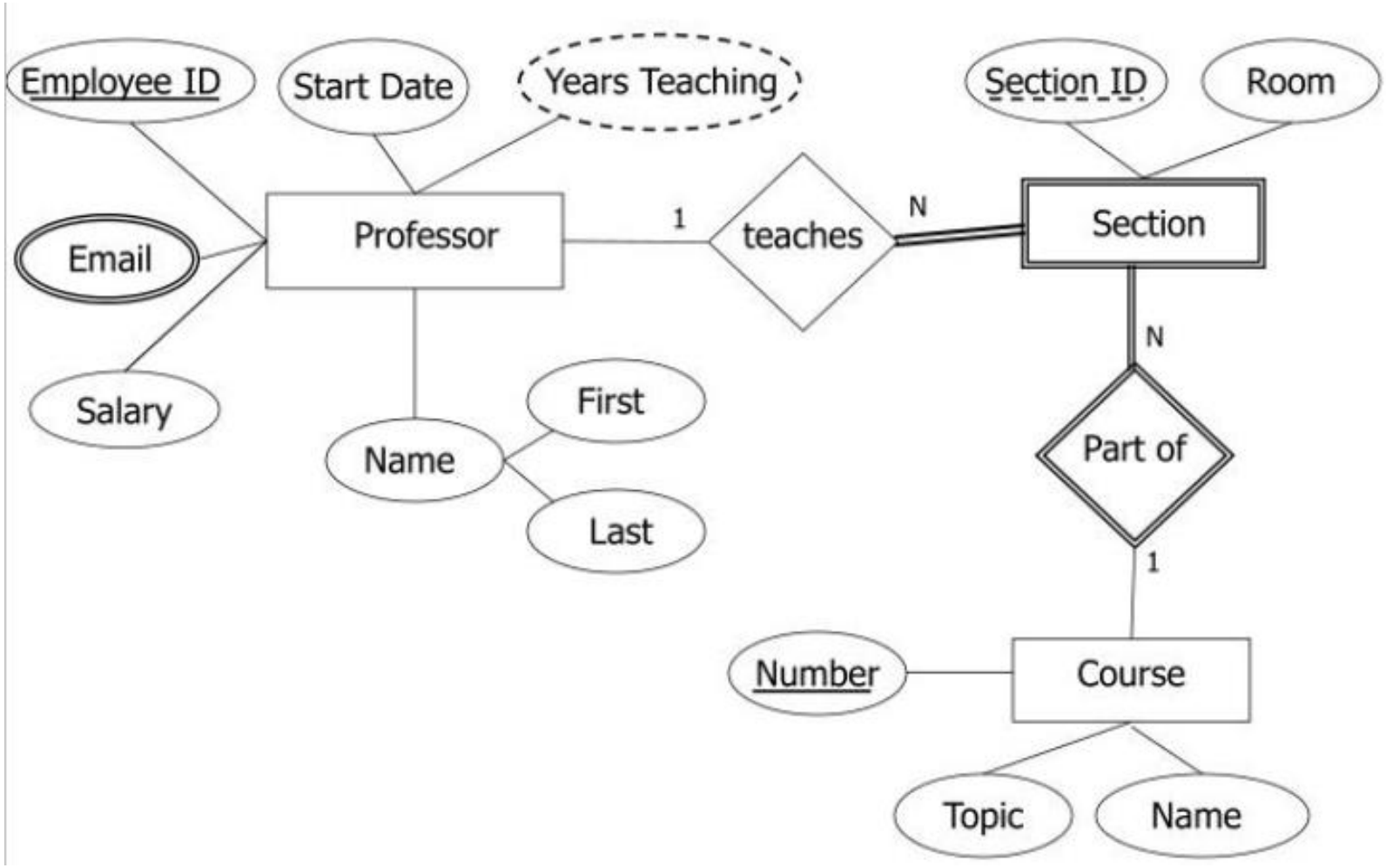
- We must convert the written database requirements into an E-R diagram
- Need to determine the entities, attributes and relationships
  - **Nouns = entities**
  - **Adjectives = attributes**
  - **Verbs = relationships**

Academic Teaching Database

Design an E-R schema for a database to store info about professors, courses and course sections indicating the following:

- The name and employee ID number of each professor
- The salary and email address(es) for each professor
- How long each professor has been at the university
- The course sections each professor teaches
- The name, number and topic for each course offered
- The section and room number for each course section
- Each course section must have only one professor
- Each course can have multiple sections

Visual View of the Database



The Pieces

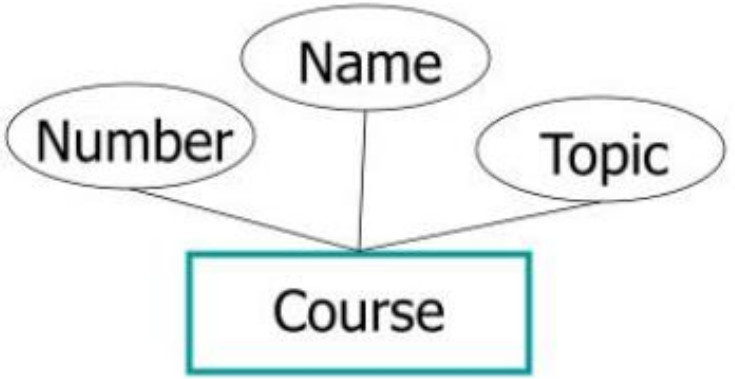
- Objects
  - Entity (Including weak entities)
  - Attribute
  - Relationship
- Structural Constraints
  - Cardinality
  - Participation

Entities

- Entity – Basic object of the ER model
  - Represents a “thing” with an independent existence
  - Can exist physically or conceptual
    - A professor, a student, a course
- Entity type – used to define a set of entities with the same properties

Entity and Entity Types

Entity Type



Entity

Number: 3753
Name: Database Management Systems
Topic: Introduction to DBMSs

Attributes

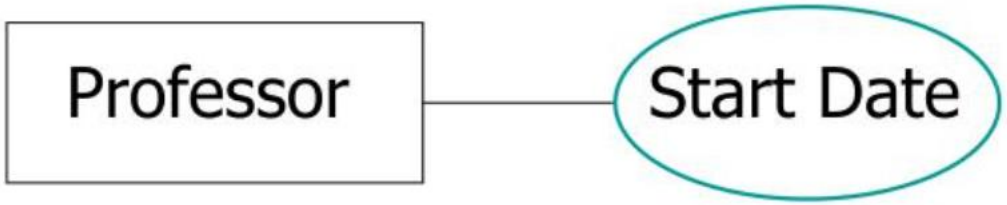
- Each entity has a set of associated properties that describes the entity. These properties are knows as attributes
- Attributes can be:
  - Simple or Composite
  - Single or Multi-valued
  - Stored or Derived
  - NULL

Attributes

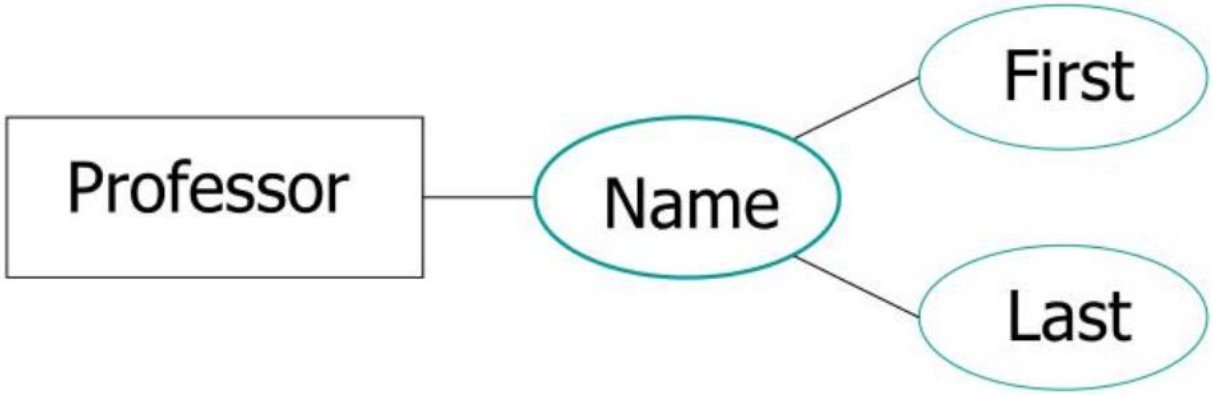
- Each entity has a set of associated properties that describes the entity. These properties are knows as attributes
- Attributes can be:
  - Simple or Composite
  - Single or Multi-valued
  - Stored or Derived
  - NULL

Attributes

Simple

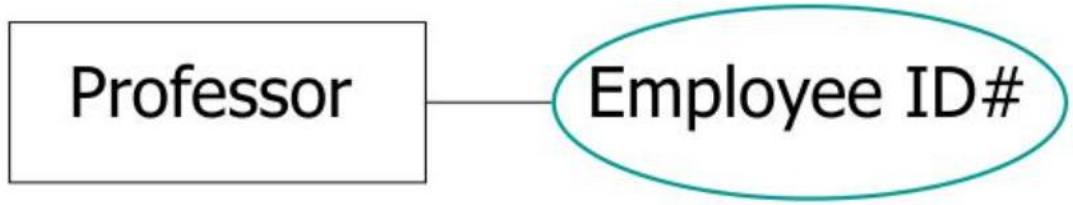


Composite

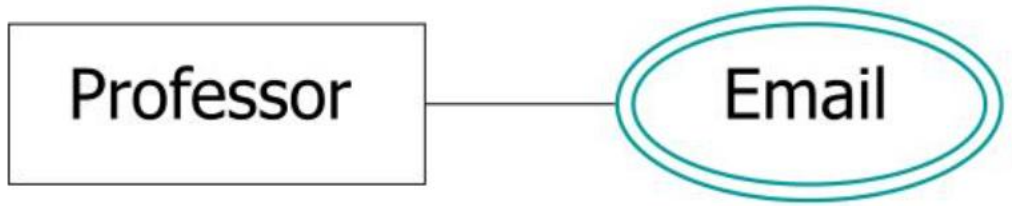


Attributes

Single

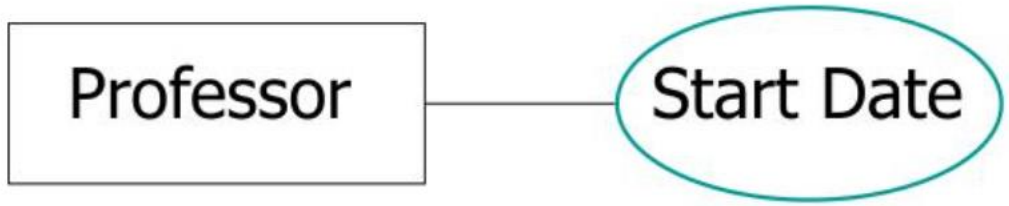


Multi-Valued



Attributes

Stored



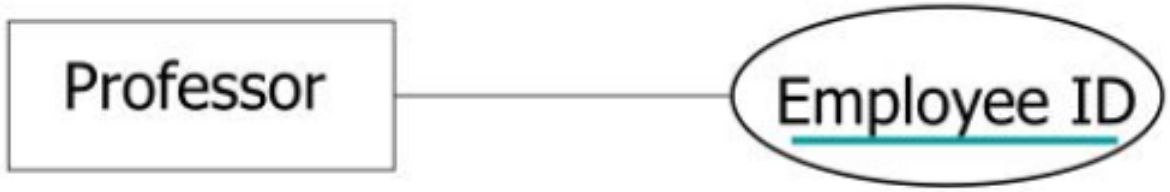
Derived

Attributes

- NULL attributes have no values
  - Not 0 (zero)
  - Not a blank string
- Attributes can be “nullable” where a null value is allowed, or “not nullable” where they must have a value

Primary Keys

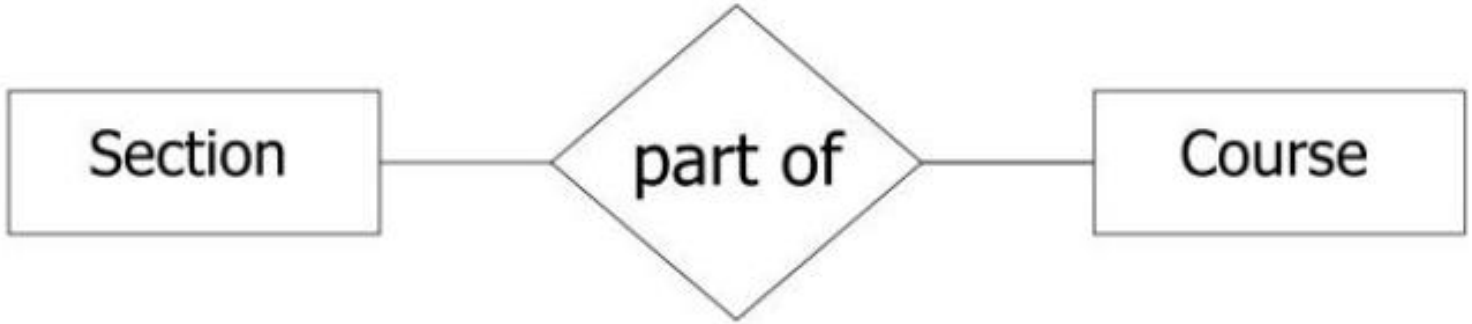
- Employee ID is the primary key
- Primary keys must be unique for the entity in question



Relationships

- Defines a set of associations between various entities
- Can have attributes to define them
- These are limited by:
  - Participation
  - Cardinality Ratio

Relationships



Participation

- Defines if the existence of an entity depends on it being related to another entity with a relationship type
  - Partial
  - Total

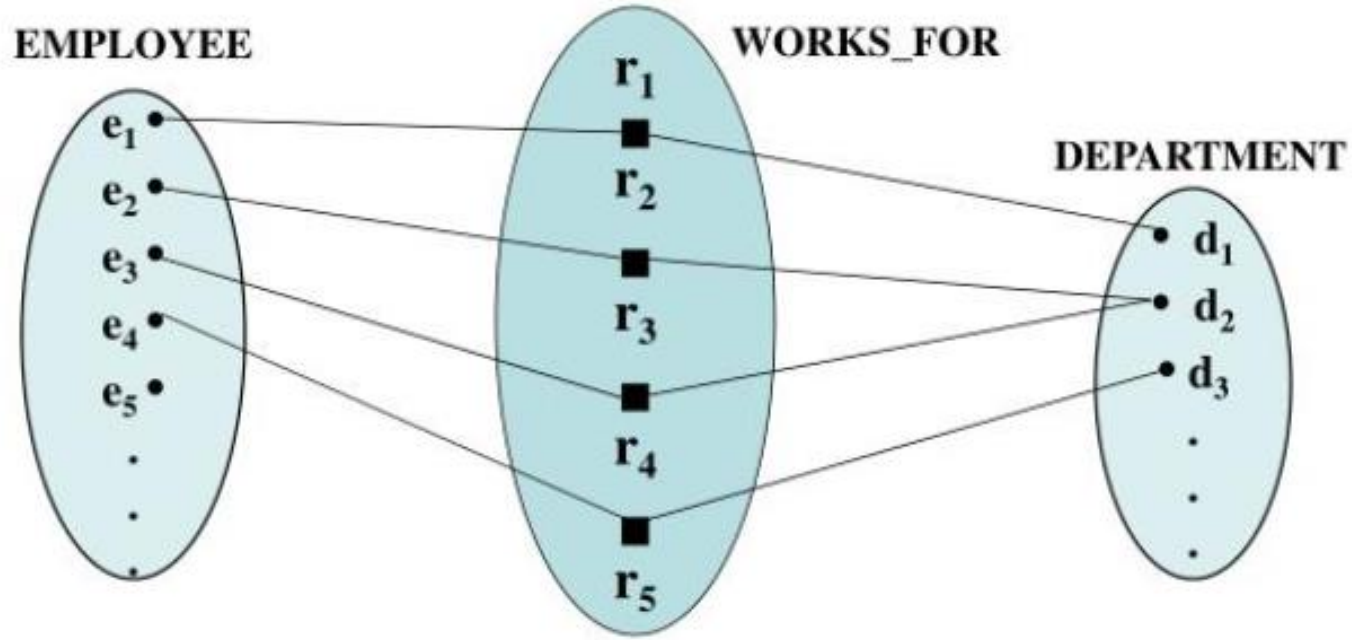
Concepts and Notations

Relationship

- A relationship R among n entity types E1,E2, .. En defines a set of associations (relationship set) among entities from these types.
- A relationship has its type and set (instances).
- Each instance in R is an association of entities, where the association includes exactly one entity from each participating entity type.
- The reference from entity to another should be represented in the ER model as a relationship not as attributes.

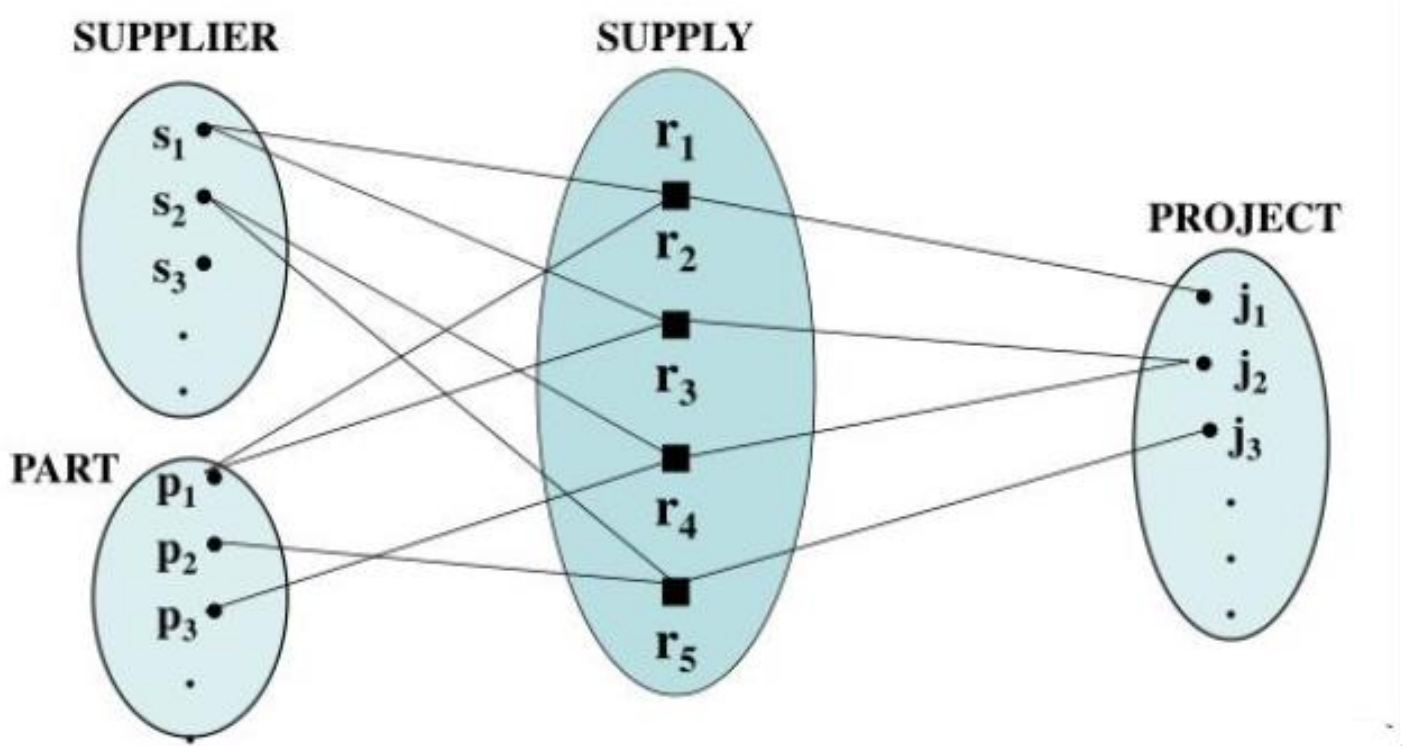
Concepts and Notations

In ER diagram, relationship types are displayed as diamond-shape boxes, which are connected by straight lines to the rectangular boxes representing the participating entity types.



Concepts and Notations

Degree of relationship type: it is the number of participating entity type(binary, ternary,.)



Constraints on Relationship Types

Cardinality Ratio: is specifies the number of relationship instances that an entity can participate in

There are three types:

1:1 (one-to-one) Relationship

1:N (one-to-many) Relationship

M:N (Many-to-many) Relationship

Continue in Part 2…