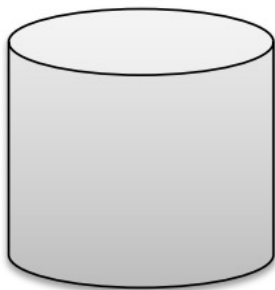


# Database Technologies

MS-218

Semester 1

Session 2021-22



# What is a DBMS?

- ❖ A large, integrated collection of data
- ❖ Models a real-world enterprise
  - *Entities* (e.g., Students, Courses)
  - *Relationships* (e.g., Asad is enrolled in CS-242)


A Database Management System (DBMS) is a piece of software designed to store and manage databases

# What do you want from a DBMS?

- ❖ Keep data around (**Persistent**)
- ❖ Answer questions (**Queries**) about data
- ❖ **Update** data

# Attributes

- ❖ Each entity would have its own attributes/features/characteristics

- Students
  - Courses
  - Professors
- 
- Entities*

Attributes of the students entity:

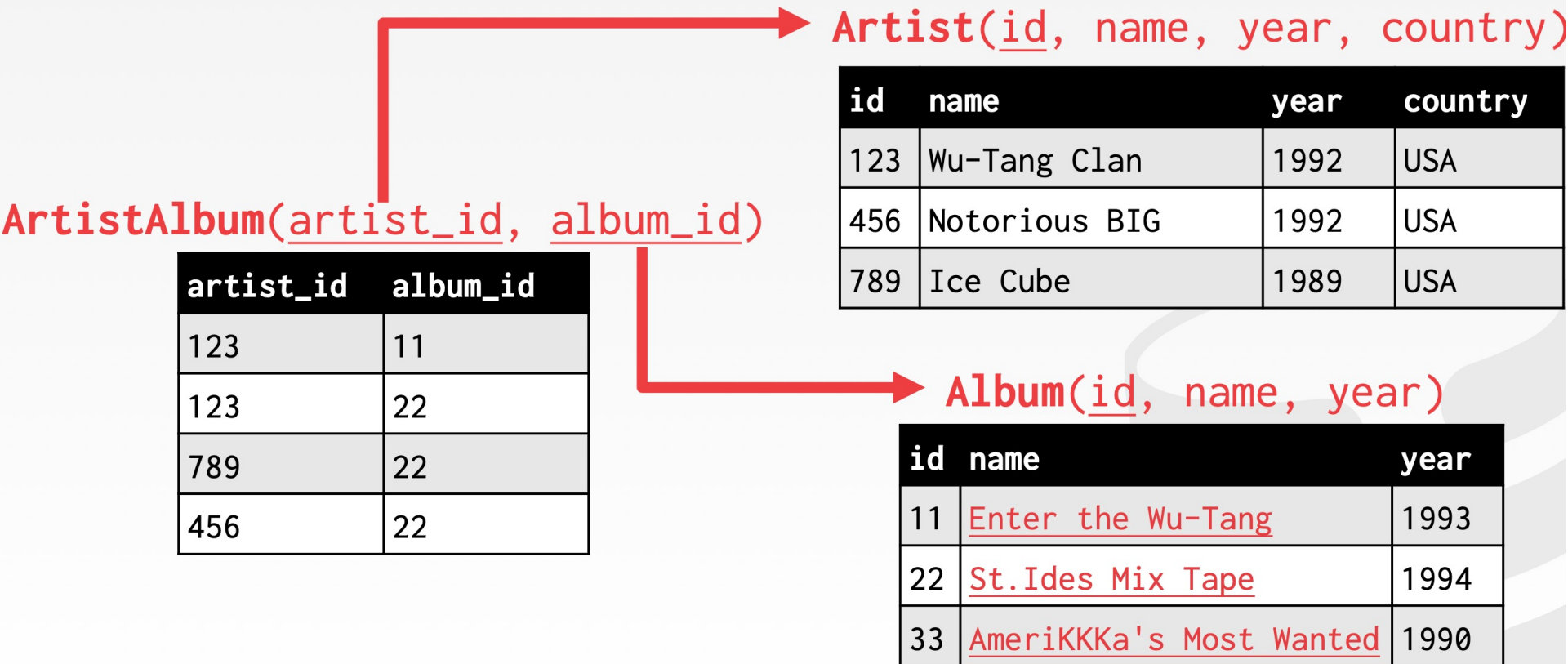
- Student Name
  - Registration#
- 
- Attributes of the Students entity*

Relationships

- Who takes what
  - Who teaches what
- 
- Relationships*

# Relational Model

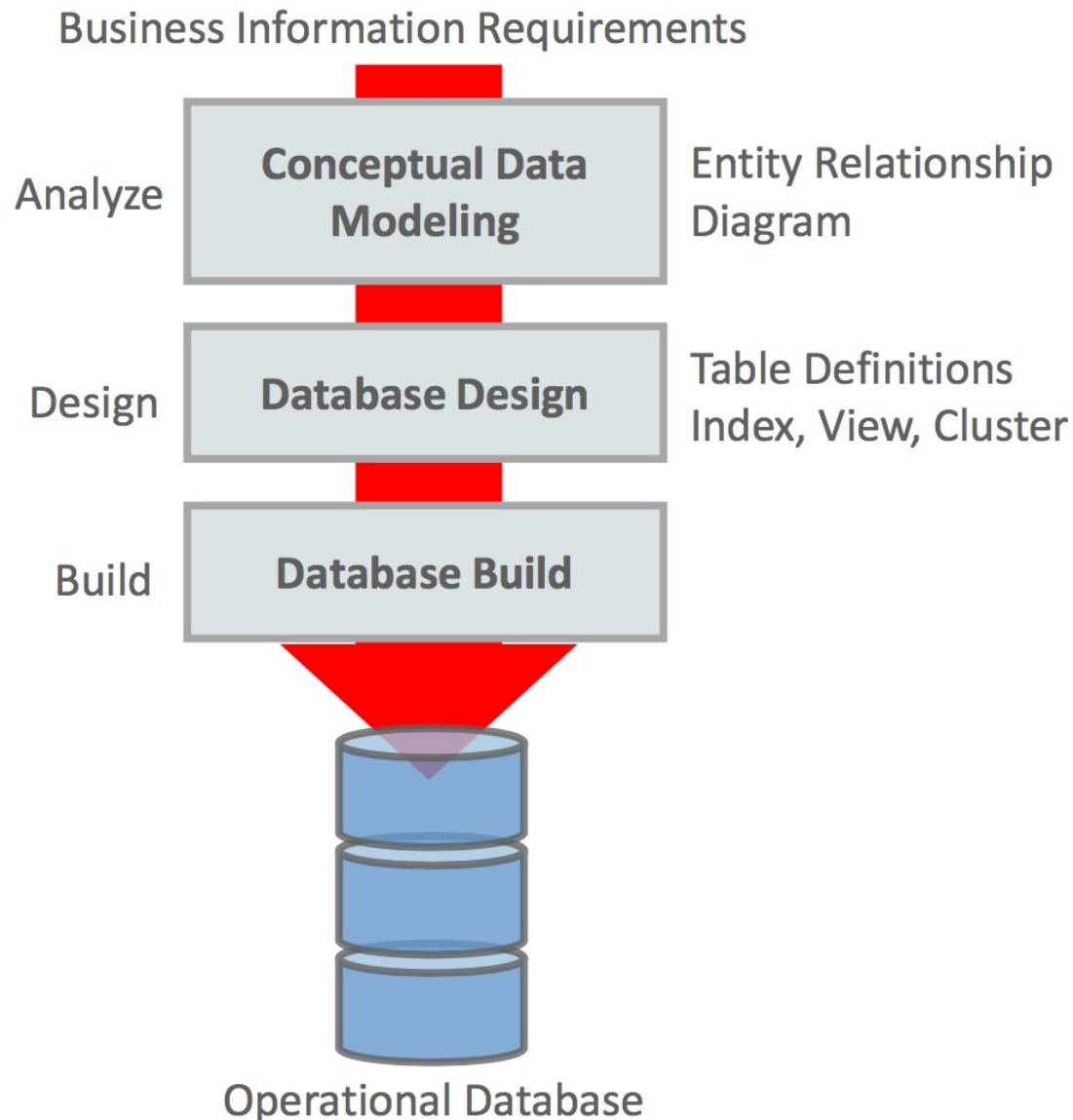
- ❖ Relations, Tuples, Primary keys, Foreign keys



# Physical data independence

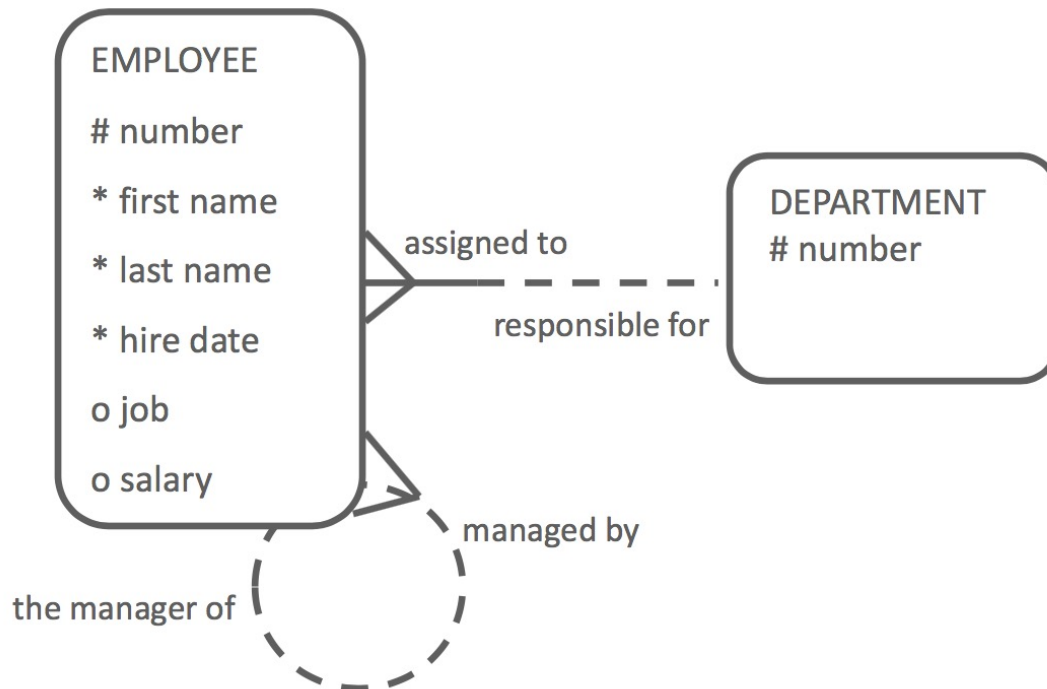
- ❖ Applications should not need to worry about how data is physically structured and stored
- ❖ Applications should work with a logical data model and declarative query language

# Database Development Process



# Step-1 Conceptual Data Modeling

- ❖ An entity relationship diagram (ERD) should completely capture and accurately model the organization's information needs and support the functions of the business.





# Step-2 Database Design (Logical View)

The table instance chart lists:

- ❖ Table name
- ❖ Column(Attributes) names
- ❖ Keys: a primary key (PK) is the unique identifier for each row of data; a foreign key (FK) links data in one table to the data in a second table by referring to the PK column in the second table
- ❖ Nulls: indicates if a column must contain a value
- ❖ (mandatory)
- ❖ Unique: indicates if the value contained in a column is unique within the table
- ❖ Data type: identifies the definition and format of the data stored in each column

# Step-3 Build Database (Physical View)

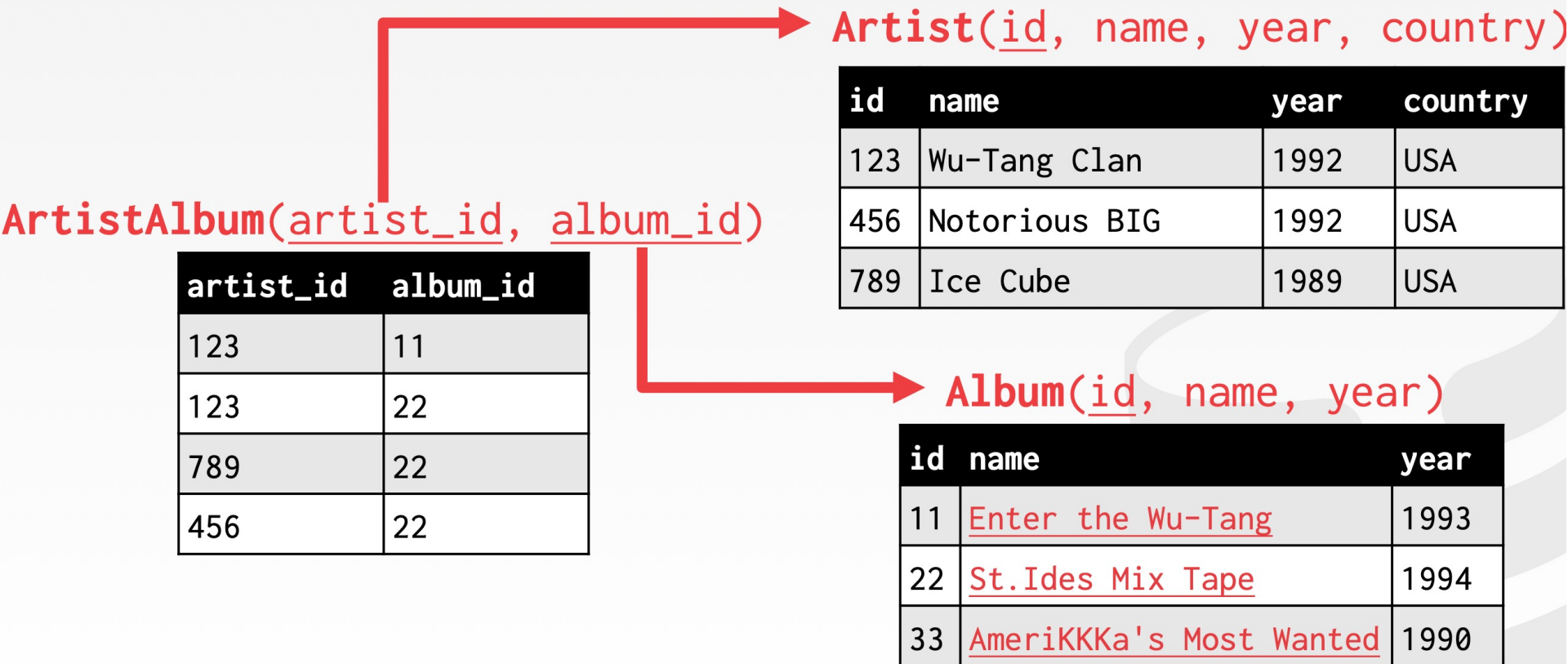
- ❖ Structured Query Language (SQL) commands are used to build the physical structure of the database.

```
CREATE TABLE departments
(deptno NUMBER(5) CONSTRAINT depts_deptno_PK PRIMARY KEY,
 name VARCHAR2(25) CONSTRAINT depts_name_NN NOT NULL,
 loc VARCHAR2(30) CONSTRAINT depts_loc_NN NOT NULL);
```

```
CREATE TABLE employees
(empno NUMBER(9) CONSTRAINT emps_empno_PK PRIMARY KEY,
 fname VARCHAR2(15) CONSTRAINT emps_fname_NN NOT NULL,
 lname VARCHAR2(20) CONSTRAINT emps_lname_NN NOT NULL,
 hiredate DATE CONSTRAINT emps_hiredt_NN NOT NULL,
 salary NUMBER(9,2),
 commission NUMBER(9,2),
 mgr NUMBER(9) CONSTRAINT emps_mgr_FK
      REFERENCES employees(empno),
 deptno NUMBER(5) CONSTRAINT emps_deptno_FK
      REFERENCES departments(deptno));
```

# Step-3 Build Database (Physical View)

- ❖ Structured Query Language (SQL) commands are used to build the physical structure of the database.



# Data Modeling

- ❖ The process of capturing the important concepts and rules that shape a business and depicting them visually on a conceptual model

# What is a Conceptual Model?

A conceptual model:

- ❖ Captures the functional and informational needs of a business
- ❖ Is based on current needs but it may reflect future needs
- ❖ Addresses the needs of a business (what is conceptually ideal), but does not address its implementation (what is physically possible)

# What is a Conceptual Model?

A conceptual model:

- ❖ Is called an “Entity Relationship Model”
- ❖ Is illustrated using an “Entity Relationship Diagram” (ERD)
- ❖ Is the result of completing the Data Modeling process


# Importance of a Conceptual Model?


A conceptual model is important to a business because it:

- ❖ Documents the processes (also known as the “business rules”) of the business
- ❖ Takes into account regulations and laws governing this industry

# A Motivating, Running Example

- ❖ Consider building a course management system (**CMS**):

- Students
  - Courses
  - Professors
- 
- Entities*

- Who takes what
  - Who teaches what
- 
- Relationships*



# Purpose of Entities

- ❖ It is important to learn about entities because they are the things about which we store data.
- ❖ For Example: Entities in School Management System?
- ❖ A school needs to store data about (as a minimum): STUDENTs, TEACHERs, COURSEs, ROOMs, GRADEs.

# Purpose of Attributes

- ❖ It is important to learn about attributes because they provide more specific information about an entity.
- ❖ Attributes help you distinguish between one instance and another by providing greater detail for the entity.
- ❖ For example:
  - In a restaurant, you need to list the individual items on a customer's order so that you can calculate the bill.

# Purpose of Unique Identifiers

- ❖ It is important to learn about unique identifiers because they distinguish one instance of an entity from another.
- ❖ Example-1: In a classroom, you need to distinguish between one student and another.
- ❖ Example-2: When classifying your CD collection, you need to distinguish between one CD and another.
- ❖ Example-3: When listing transactions on a financial statement, you need to distinguish between one transaction and another.

# Entity Defined

An entity is:

- ❖ “Something” of significance to the business about which data must be known
- ❖ A name for a set of similar things that you can list
- ❖ Usually a noun
- ❖ Examples: objects, events, people
- ❖ Entities have instances.
- ❖ An instance is a single occurrence of an entity.

# Entities and Instances

## Entities

PERSON

PRODUCT TYPE

JOB

SKILL LEVEL

ANIMAL

CAR

## Instances

William, Emma

Shoe, Video Game

Electrician, IT Technician

Beginner, Expert

Dog, Cat

Honda Civic, Toyota Corolla

# Entities

- ❖ Some entities have many instances, some have few
- ❖ Example?
- ❖ Example: Student-Faculty
- ❖ Entities can be: Tangible, like PERSON or PRODUCT
- ❖ Entities can be: Intangible, like SKILL LEVEL
- ❖ Entities can be: An event, like CONCERT

# What is an Attribute?

- ❖ Like an entity, an attribute represents something of significance to the business.
- ❖ An attribute is a specific piece of information that helps:
  - Describe an entity
  - Quantify an entity
  - Qualify an entity
  - Classify an entity
  - Specify an entity
- ❖ An attribute has a single value.

# Attributes

- ❖ Attributes have values. An attribute value can be a number, a character string, a date, an image, a sound, etc.
- ❖ These are called "data types" or "formats." Every attribute stores one piece of data of one specific data type.



# Attributes: Examples

Entities	Attributes
CUSTOMER	family name, age, shoe size, town of residence, email
CAR	model, weight, catalog price
ORDER	order date, ship date
JOB	title, description
TRANSACTION	amount, transaction date
EMPLOYMENT CONTRACT	start date, salary

# Attributes: Examples

- ❖ What is the data type of each attribute in CUSTOMER?
- ❖ For example: family name is a character string.  
Attributes are single-valued. Each attribute can have only one value (at any point in time) for each instance of the entity.

Entities	Attributes
CUSTOMER	family name, age, shoe size, town of residence, email
CAR	model, weight, catalog price
ORDER	order date, ship date
JOB	title, description
TRANSACTION	amount, transaction date
EMPLOYMENT CONTRACT	start date, salary

# Attributes

- ❖ Some attributes (such as age) have values that constantly change.
- ❖ These are called volatile attributes.
- ❖ Other attributes (such as order date) will rarely change, if ever.
- ❖ These are nonvolatile attributes.
- ❖ If given a choice, select the nonvolatile attribute.
- ❖ For example, use birth date instead of age.

# Attributes

- ❖ Some attributes must contain a value—these are mandatory attributes.
- ❖ For example: in most businesses that track personal information, name is required.
- ❖ Other attributes may either contain a value or be left null— these are optional attributes.
- ❖ For example: cell phone number is often optional except in mobile or wireless applications.

# Identifiers

- ❖ An EMPLOYEE has a unique identifier (UID).
- ❖ A UID is either a single attribute or a combination of multiple attributes that distinguishes one employee from another.
- ❖ How do you find a specific employee that works for the company?
- ❖ What information uniquely identifies one EMPLOYEE?

# Identifiers

- ❖ Think about all the students in the classroom.
- ❖ Each student is described by several traits or attributes.
- ❖ Which attribute or attributes allow you to pick a single student from the rest of the class?
- ❖ That is the student's UID.

# Attendance

[Settings](#)

Take Attendance

[Attendance Record](#)

## TAKE ATTENDANCE

### 2122-MS218 Database Technologies

Check-In is currently running .  
Students can check in until check in period closes or is ended.

1 8 6 0

End Check In

# Project Groups

- ❖ 3 students in each group
- ❖ Select a group leader
- ❖ Group leader will email me group info, like names and registration numbers of other students. Please keep other members in CC
- ❖ If you have any difficulty in finding group members, then please email me, [jamal.nasir@nuigalway.ie](mailto:jamal.nasir@nuigalway.ie)



# Entity Relationship Diagram(ERD)

- ❖ An entity relationship diagram (ERD) is a consistent tool that can be used to represent the data requirements of a business regardless of the type of database that is used, and even in the absence of one!

# What is an Entity Relationship Model?

An Entity Relationship Model:

- ❖ Is a list of all entities and attributes as well as all relationships between the entities that are of importance.
- ❖ Provides background information such as entity descriptions, data types, and constraints.
- ❖ Note: The model does not require a diagram, but the diagram is typically a very useful tool.

# ER Drawing Conventions

- ❖ Entities are represented by softboxes.
- ❖ Entity names go in the softboxes.
- ❖ Entity names are always singular and written with all capital letters.

**EMPLOYEE**

**JOB**

**DEPARTMENT**

# ER Drawing Conventions

- ❖ Attributes are listed under the entity names.
- ❖ Mandatory attributes are marked with an asterisk: “\*”
- ❖ Optional attributes are marked with a circle: “o”
- ❖ Unique identifiers are marked with a hash sign: “#”



# Relationships in Data Models

Relationships:

- ❖ Represent something of significance or importance to the business
- ❖ Show how entities are related to each other
- ❖ Exist only between entities (or one entity and itself)
- ❖ Are bi-directional
- ❖ Are named at both ends
- ❖ Have optionality
- ❖ Have cardinality

# What is Optionality in a Relationship?

- ❖ Relationships are either mandatory or optional.
- ❖ Consider the two entities EMPLOYEE and JOB.
- ❖ Based on what you know about instances of the entities, you can determine optionality by answering two questions:
  - ❖ Must every employee have a job?
    - In other words, is this a mandatory or optional relationship for an employee?
  - ❖ Must every job be assigned to an employee?
    - In other words, is this a mandatory or optional relationship for a job?

# What is Cardinality in a Relationship?

- ❖ Cardinality measures the quantity of something.
- ❖ In a relationship, it determines the degree to which one entity is related to another by answering the question, “How many?”
- ❖ For example:
  - How many jobs can one employee hold? One job only? Or more than one job?
  - How many employees can hold one specific job? One employee only? Or more than one employee?
- ❖ • Note: The cardinality of a relationship only answers whether the number is singular or plural; it does not answer with a specific plural number.

# Optionality and Cardinality

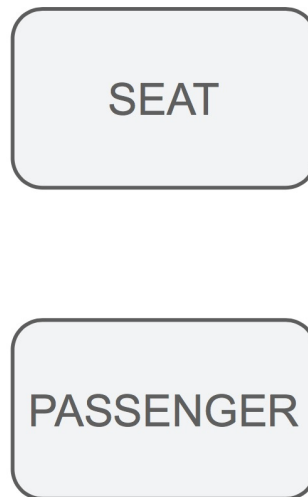
Examples:

- ❖ Each EMPLOYEE must hold one and only one JOB
- ❖ Each JOB may be held by one or more EMPLOYEEs
- ❖ Each PRODUCT must be classified by one and only one PRODUCT TYPE
- ❖ Each PRODUCT TYPE may classify one or more PRODUCTS



# Relationships

- ❖ Each SEAT may be sold to one or more PASSENGERs
- ❖ Each PASSENGER may purchase one SEAT
- ❖ SEAT is sold to a PASSENGER (or PASSENGERs -- hence, overbooking)
- ❖ PASSENGER purchases or books a SEAT



# Business Scenario 1

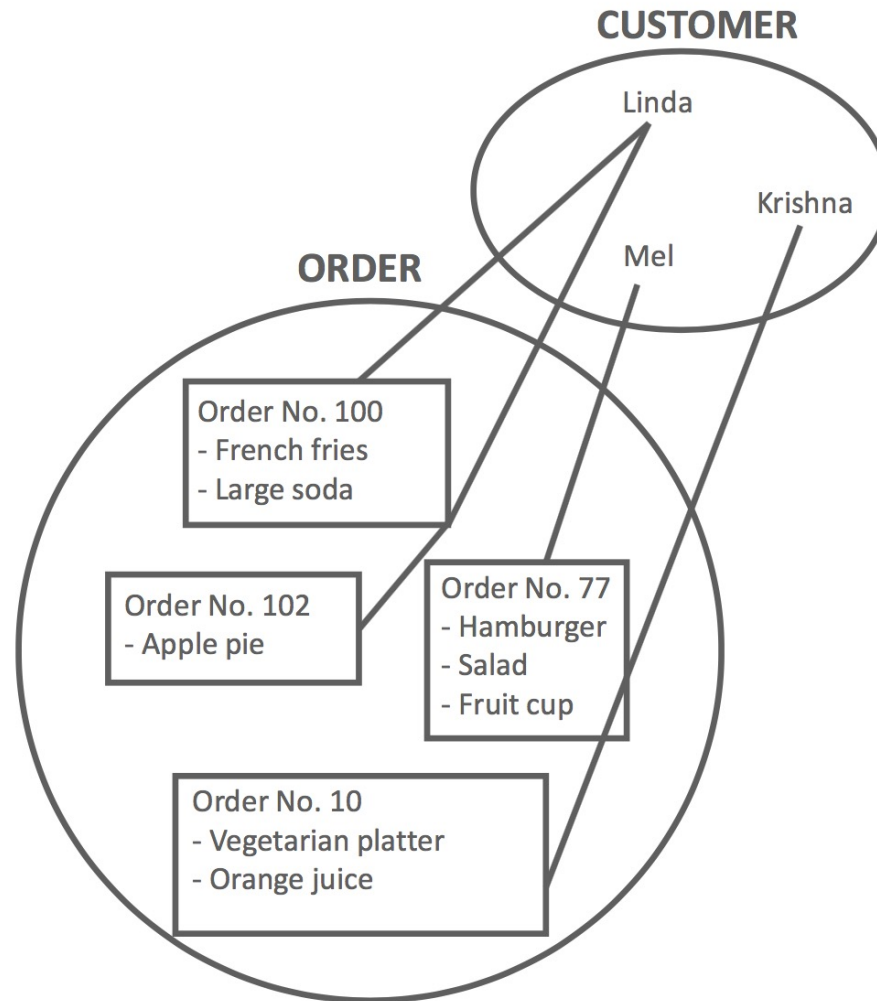
What are the relationships in the following business scenario?

- ❖ “In our restaurant, a customer walks up to the counter and places their order. A customer can order for him or herself only, or for him/herself and others. For example, a mother orders for herself and her children.
- ❖ We consider the mother to be the customer who owns the order and is responsible for payment. Over a period of time, a customer can place as many orders as she wants.”

# Business Scenario 1

- ❖ Optionality = Must or may?
- ❖ Cardinality = How many?
- ❖ Each ORDER must be placed by one and only one CUSTOMER.
- ❖ Each CUSTOMER must place one or more ORDERS.

# Business Scenario 1



# Business Scenario 2

- ❖ A relationship can join one entity to itself.
- ❖ Examine the following scenario:
- ❖ “We need to keep track of our employees and their managers. Every employee has one manager, including the managing director who manages him/herself. Each manager can manage several employees.”

# Business Scenario 2

- ❖ Since managers are also employees, both are listed in the same entity: **EMPLOYEE**.
- ❖ Relationships?
- ❖ Each **EMPLOYEE** may be managed by one and only one **EMPLOYEE**
- ❖ Each **EMPLOYEE** may manage one or more **EMPLOYEEs**

# Drawing Conventions for Relationships

- ❖ Relationships are lines that connect entities.
- ❖ These lines are either solid or dashed.
- ❖ These lines terminate in either a “single toe” or a “crow’s foot” at the end of each entity.

