DATA MODELLING

DATA MODELLING

- Data modeling is important because it specifies the data structure, which can impact all aspects of data usage.
- For example, it can have a significant impact on performance. This is particularly true with data warehousing.
- And, the data warehouse is the primary structural element in business intelligence.

DATA MODELLING

- Generally speaking, a model is an abstraction and reflection of the real world.
- Modeling gives us the ability to visualize what we cannot yet realize. It is the same with data modeling.
- The primary aim of a data model is to make sure that all data objects required by the business are accurately and fully represented.

Data Modeling

- **WHAT IS A DATA MODEL?**

  A data model is an abstraction of some aspect of the real world (system).

- **NEED OF DATA MODEL?**
  - **Helps to visualize the business**
  - **A model is a means of communication.**
  - **Models help elicit and document requirements.**
  - **Models reduce the cost of change.**
  - **Model is the essence of DW architecture based on which DW will be implemented**

What do we want to do with the data?

Model depends on what kind of data analysis we want to do:

- **Different Data Analysis Techniques**
  - ✓**Query and reporting**
    - **Display Query Results**
  - ✓**Multidimensional analysis**
    - **Analyze data content by looking at it in different perspectives**
  - ✓**Data mining**
    - **Discover patterns and clustering attributes in data**

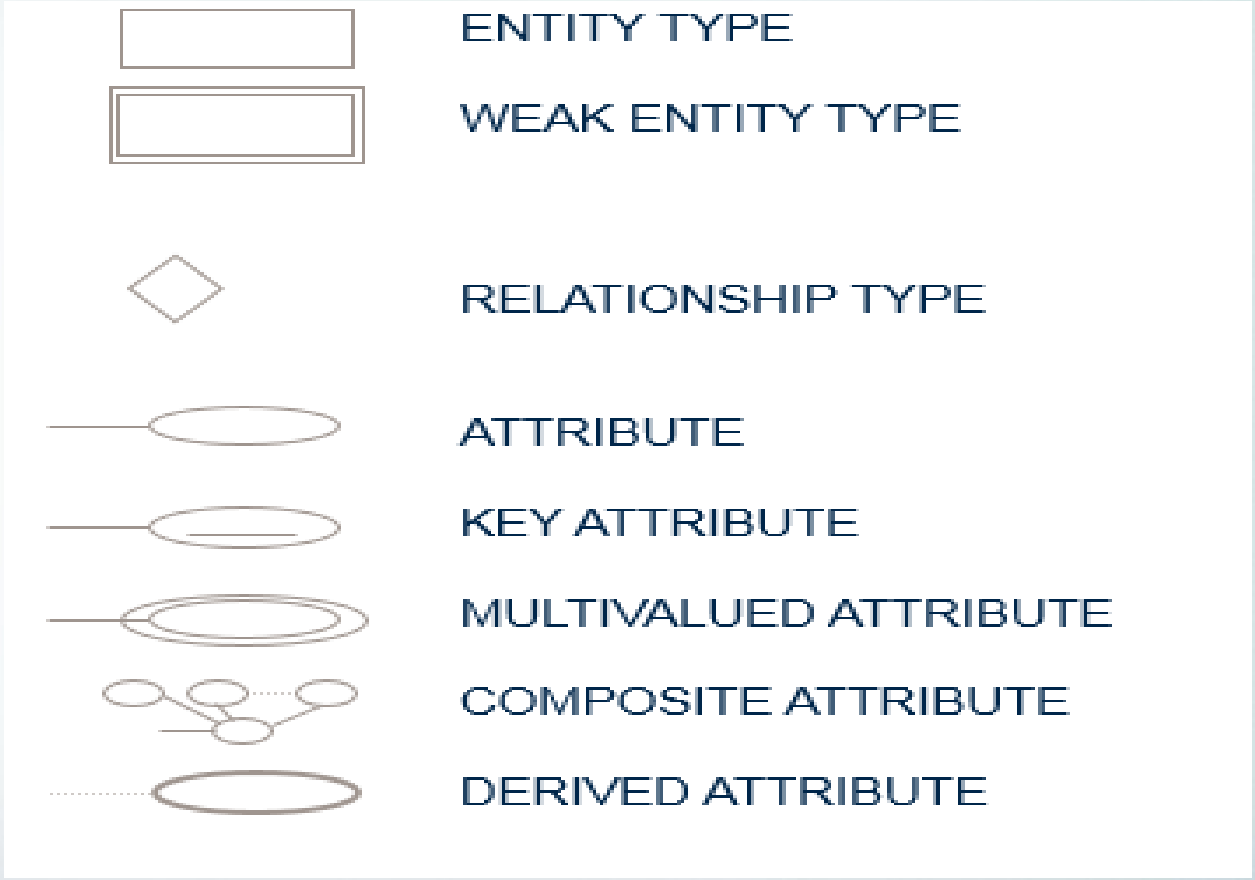
DATA MODELING TECHNIQUES:

- **There are 2 techniques of Data Modeling :**
  - **E-R Modeling**
  - **Dimensional Modeling**

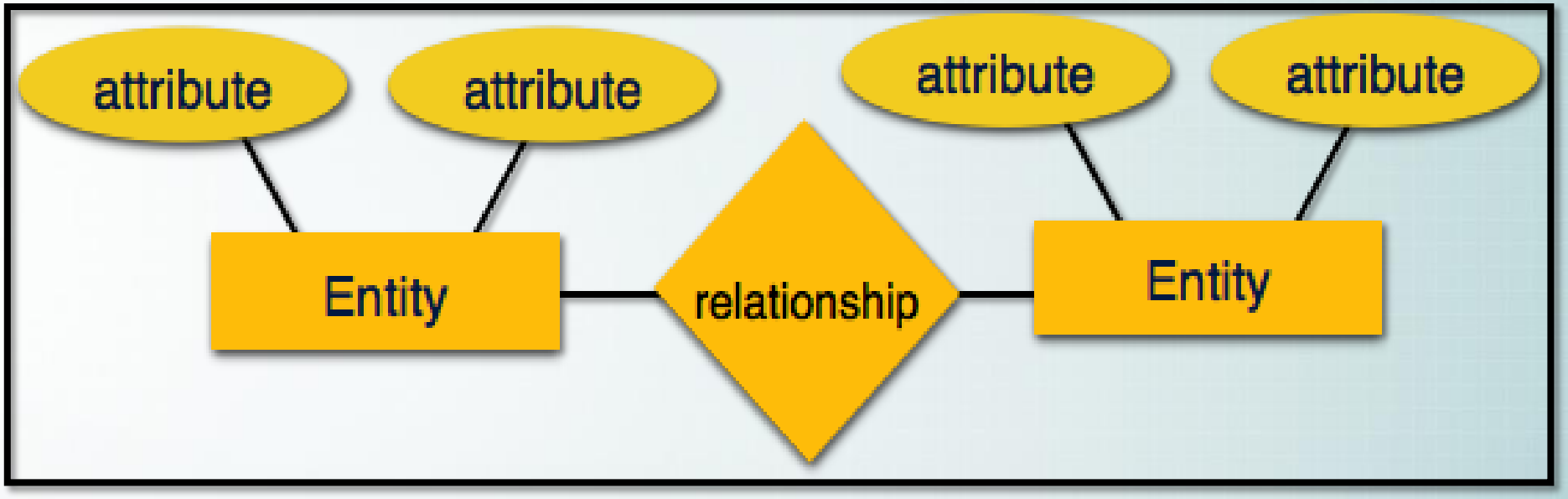
Entity-Relationship Model

- **Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them.**
- **While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.**
- **ER Model is best used for the conceptual design of a database.ER Model is based on the following:**
  - **Entities** and their ***attributes***.
  - **Relationships** among **entities**.

ER NOTATIONS



ER Model - Example Notation



ENTITY and RELATIONSHIP

- **Entity** – An entity in an ER Model is a real-world entity having properties called **attributes**.
  - Every **attribute** is defined by its set of values called **domain**.
  - **For example**, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.

- **Relationship** – The logical association among entities is called ***relationship***.
  - Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

TYPES OF DATAMODEL

- **There are 3 types of Data Models as listed below:**
  - **CONCEPTUAL DATA MODEL**
  - **LOGICAL DATA MODEL**
  - **PHYSICAL DATA MODEL**

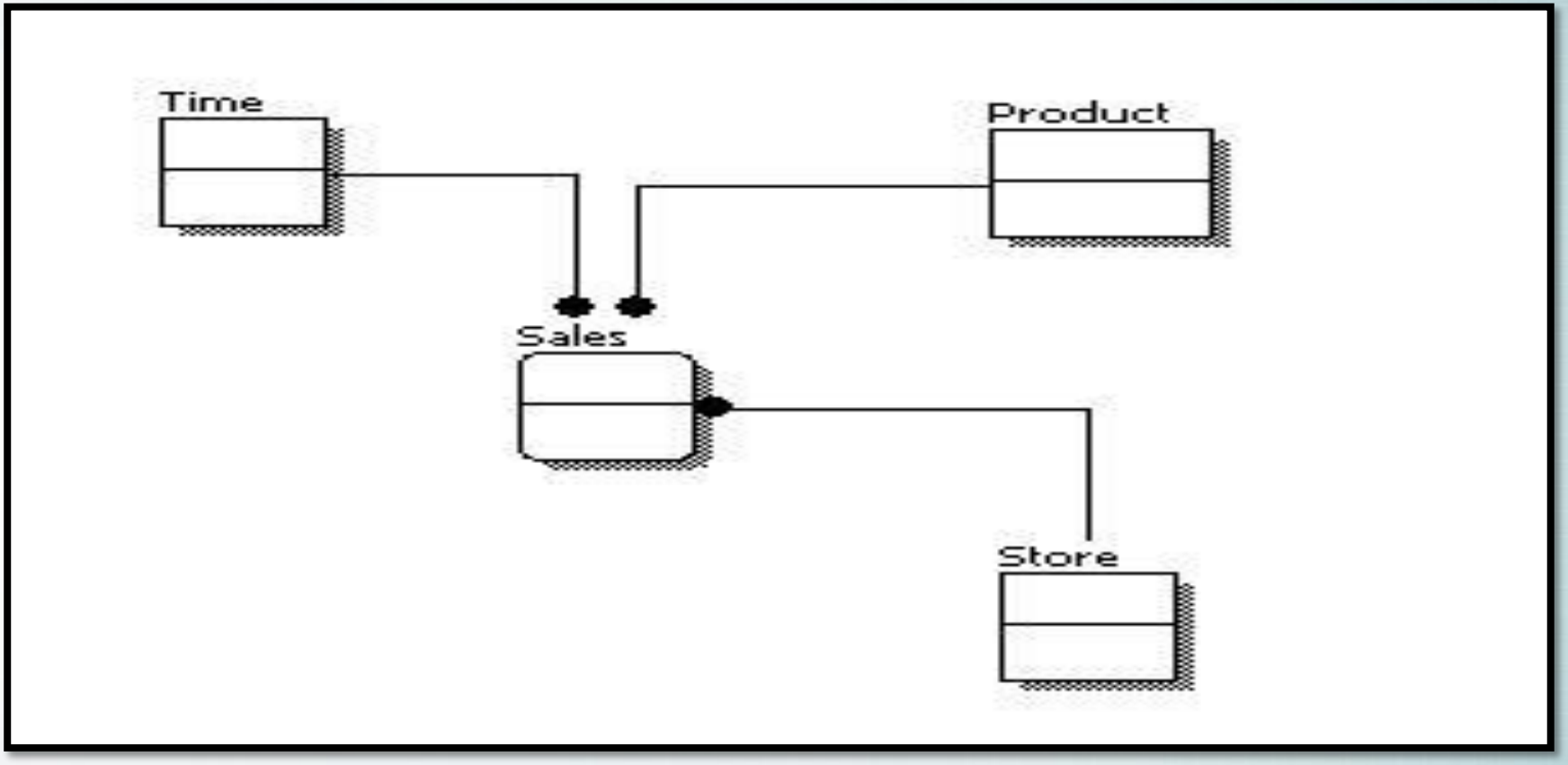
CONCEPTUAL DATA MODEL

- **A conceptual data model identifies the highest-level relationships between the different entities.**

- **Features of conceptual data model include:**

  - Includes the important entities and the relationships among them.
  - No attribute is specified.
  - No primary key is specified.

CONCEPTUAL DATA MODEL



From the figure above, we can see that the only information shown via the conceptual data model is the entities that describe the data and the relationships between those entities. No other information is shown through the conceptual data model.

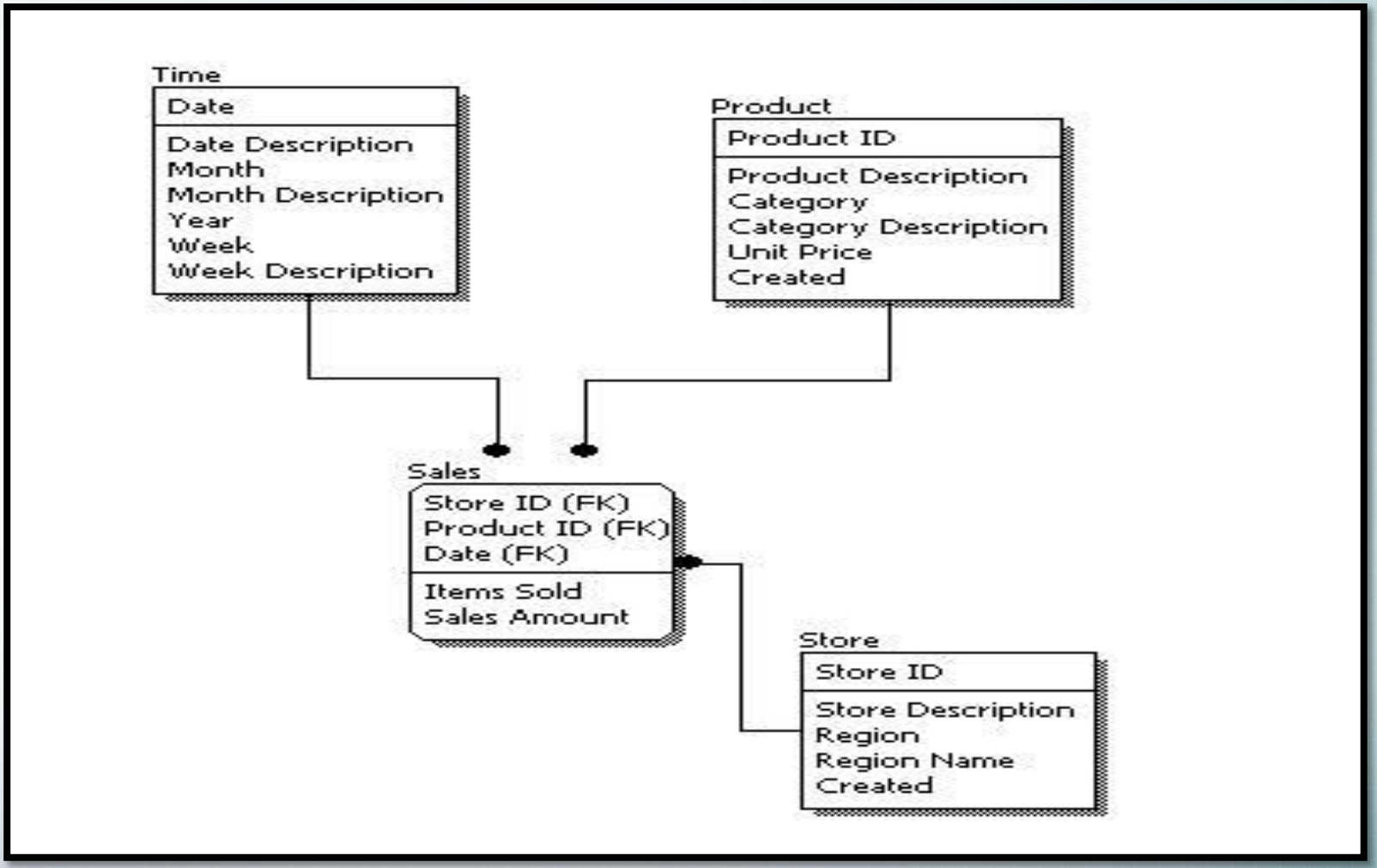
LOGICAL DATA MODEL

- **A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database.**
- **Features of a logical data model include:**
  - Includes all entities and relationships among them.
  - All attributes for each entity are specified.
  - The primary key for each entity is specified.
  - Foreign keys (keys identifying the relationship between different entities) are specified.
  - Normalization occurs at this level.

Steps for Logical Data Model Design

- The steps for designing the logical data model are as follows:
  - ***Specify primary keys for all entities.***
  - ***Find the relationships between different entities.***
  - ***Find all attributes for each entity.***
  - ***Resolve many-to-many relationships.***
  - ***Normalization.***

LOGICAL DATA MODEL



COMPARISON LOGICAL '&' CONCEPTUAL

- ***Comparing the logical data model shown above with the conceptual data model diagram, we see the main differences between the two:***

- **In a logical data model, primary keys are present, whereas in a conceptual data model, no primary key is present.**

- **In a logical data model, all attributes are specified within an entity. No attributes are specified in a conceptual data model.**

COMPARISON LOGICAL '&' CONCEPTUAL

- **Relationships between entities are specified using primary keys and foreign keys in a logical data model.**

- **In a conceptual data model, the relationships are simply stated, not specified, so we simply know that two entities are related, but we do not specify what attributes are used for this relationship.**

PHYSICAL DATA MODEL

- **Physical data model represents how the model will be built in the database.**
- **A physical database model shows all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.**

- **Features of a physical data model include:**
  - Specification all tables and columns.
  - Foreign keys are used to identify relationships between tables.

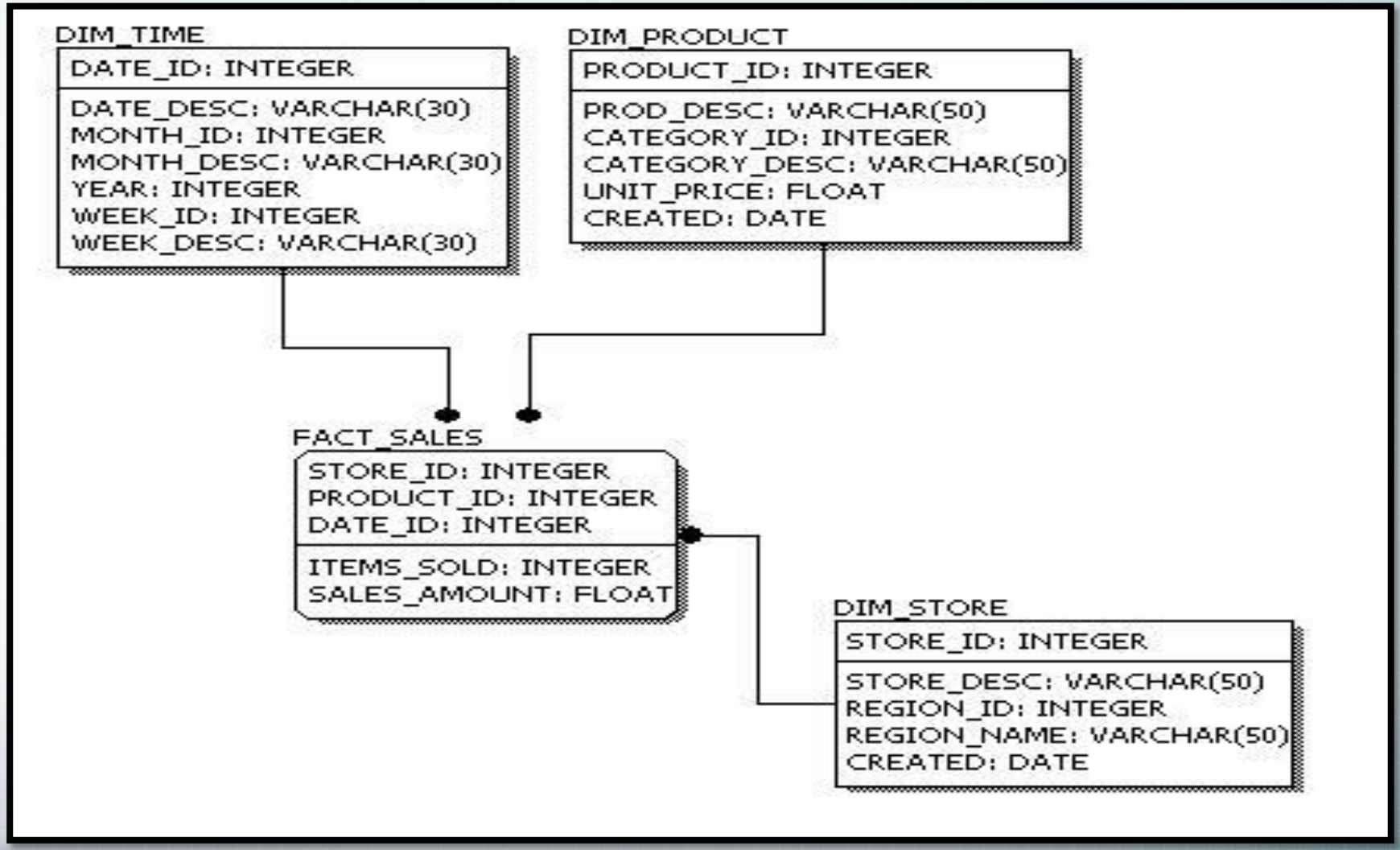
PHYSICAL DATA MODEL

- Denormalization may occur based on user requirements.
- Physical considerations may cause the physical data model to be quite different from the logical data model.
- Physical data model will be different for different RDBMS. For example, data type for a column may be different between MySQL and SQL Server.

STEPS FOR DESIGNING PHYSICAL DM

- **The steps for physical data model design are as follows:**
  - **Convert entities into tables.**
  - **Convert relationships into foreign keys.**
  - **Convert attributes into columns.**
  - **Modify the physical data model based on physical constraints / requirements.**

PHYSICAL DATA MODEL



CONCEPTS OF ER - MODELLING

CONCEPTS OF THE ER MODEL

- **Entity types**
- **Relationship types**
- **Attributes**

ENTITY TYPE

- **Entity type**
  - Group of objects with same properties, identified by enterprise as having an independent existence.
  - For example, in a school database, students, teachers, classes, and courses offered can be considered as entities.
- **Entity occurrence**
  - Uniquely identifiable object of an entity type.

Example: For an ENTITY

ER - MODELLING SYMBOLS

– ENTITY SET

– WEAK ENTITY SET

– RELATIONSHIP SET

– IDENTIFYING RELATIONSHIP FOR A WEAK ENTITY SET

ER - MODELLING SYMBOLS

– ATTRIBUTE

– MULTI-VALUED ATTRIBUTE

– DERIVED ATTRIBUTE

– PRIMARY KEY ATTRIBUTE

ER - MODELLING SYMBOLS

– ONE-TO-ONE RELATIONSHIP

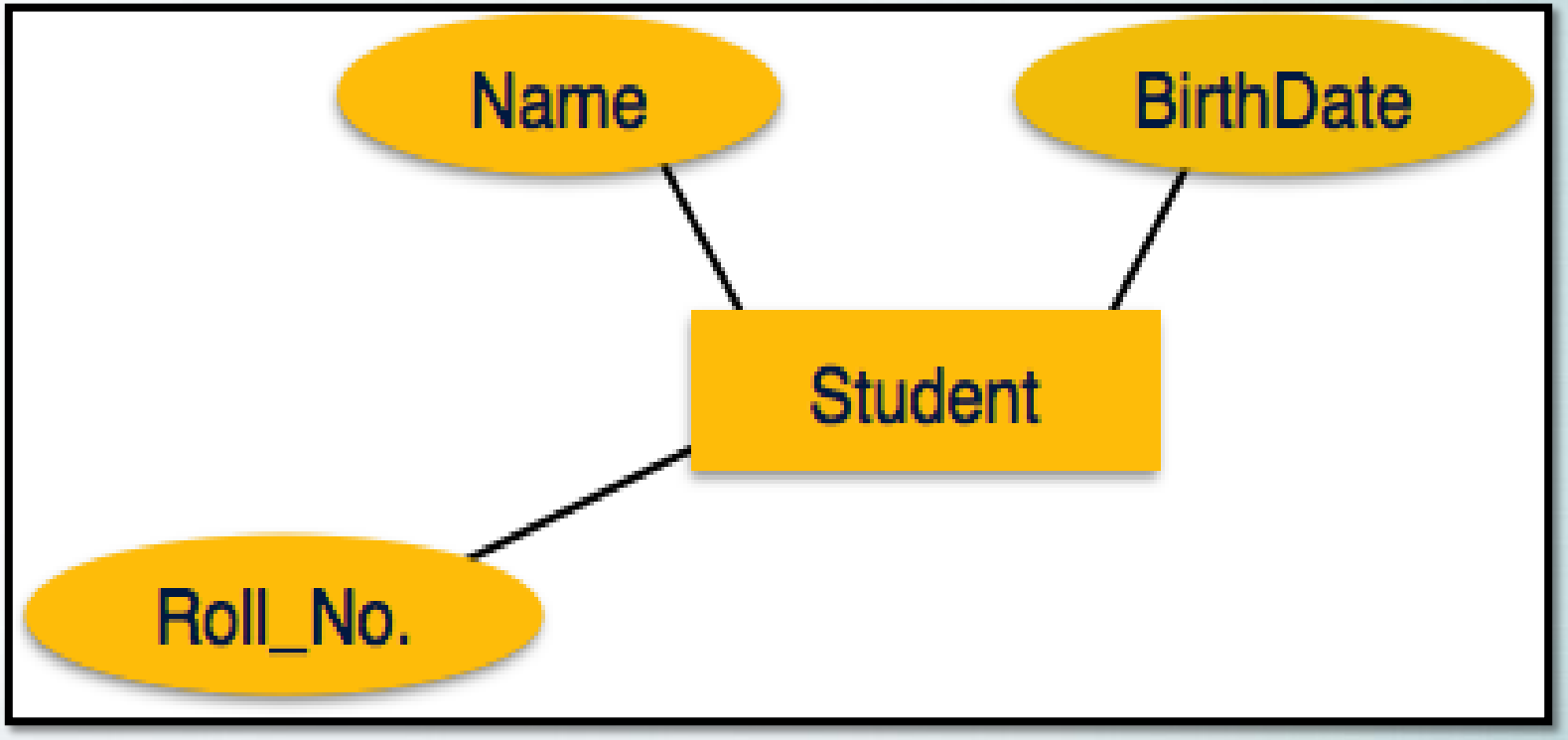
– MANY-TO-ONE RELATIONSHIP

– MANY-TO-MANY RELATIONSHIP

ATTRIBUTES

- **Attribute**
  - Property of an entity or a relationship type.
  - For example, a student entity may have name, class, and age as attributes.
- **Attribute Domain**
  - Set of allowable values for one or more attributes.

EXAMPLE: For an ATTRIBUTE

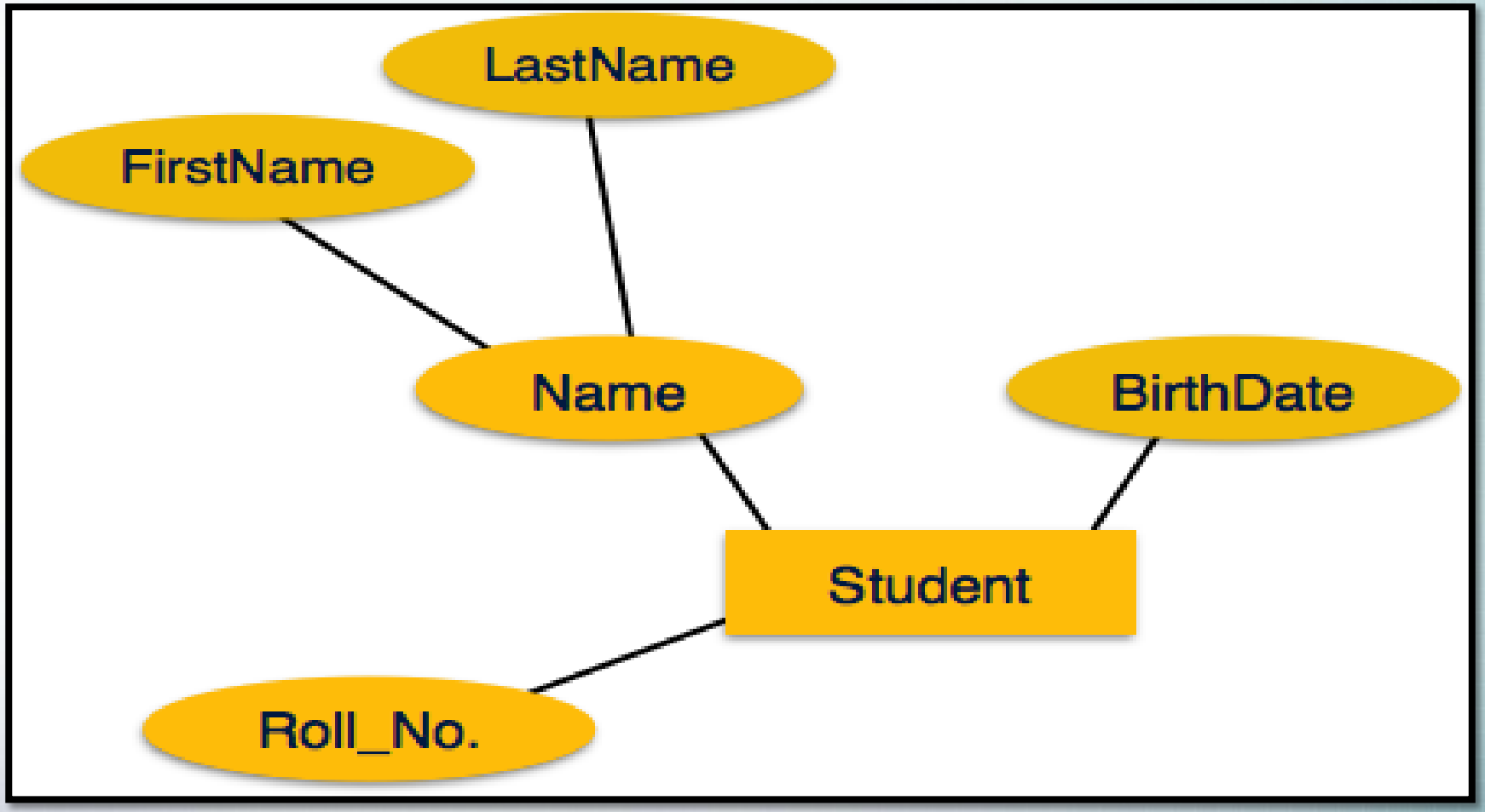


ATTRIBUTES TYPES

- **Simple Attribute**
  - Attribute composed of a single component with an independent existence.
  - For example, a student's phone number is an atomic value of 10 digits.

- **Composite Attribute**
  - Attribute composed of multiple components, each with an independent existence.
  - For example, a student's complete name may have first_name and last_name.

EXAMPLE: COMPOSITE ATTRIBUTE

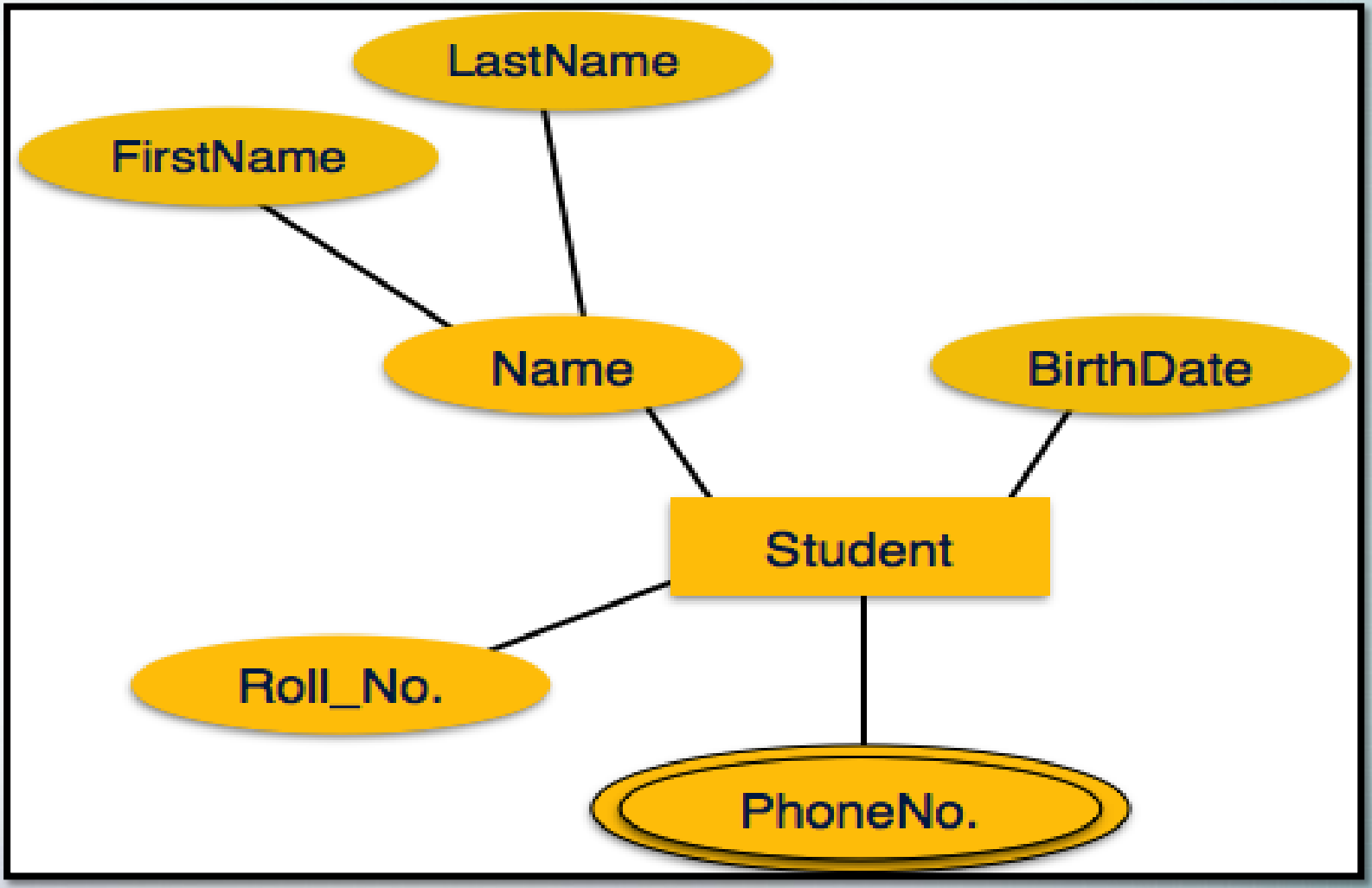


ATTRIBUTES TYPES

- **Single-valued Attribute**
  - Attribute that holds a single value for each occurrence of an entity type.
  - For example – Social_Security_Number.

- **Multi-valued Attribute**
  - Attribute that holds multiple values for each occurrence of an entity type.
  - For example, a person can have more than one phone number, email_address, etc.

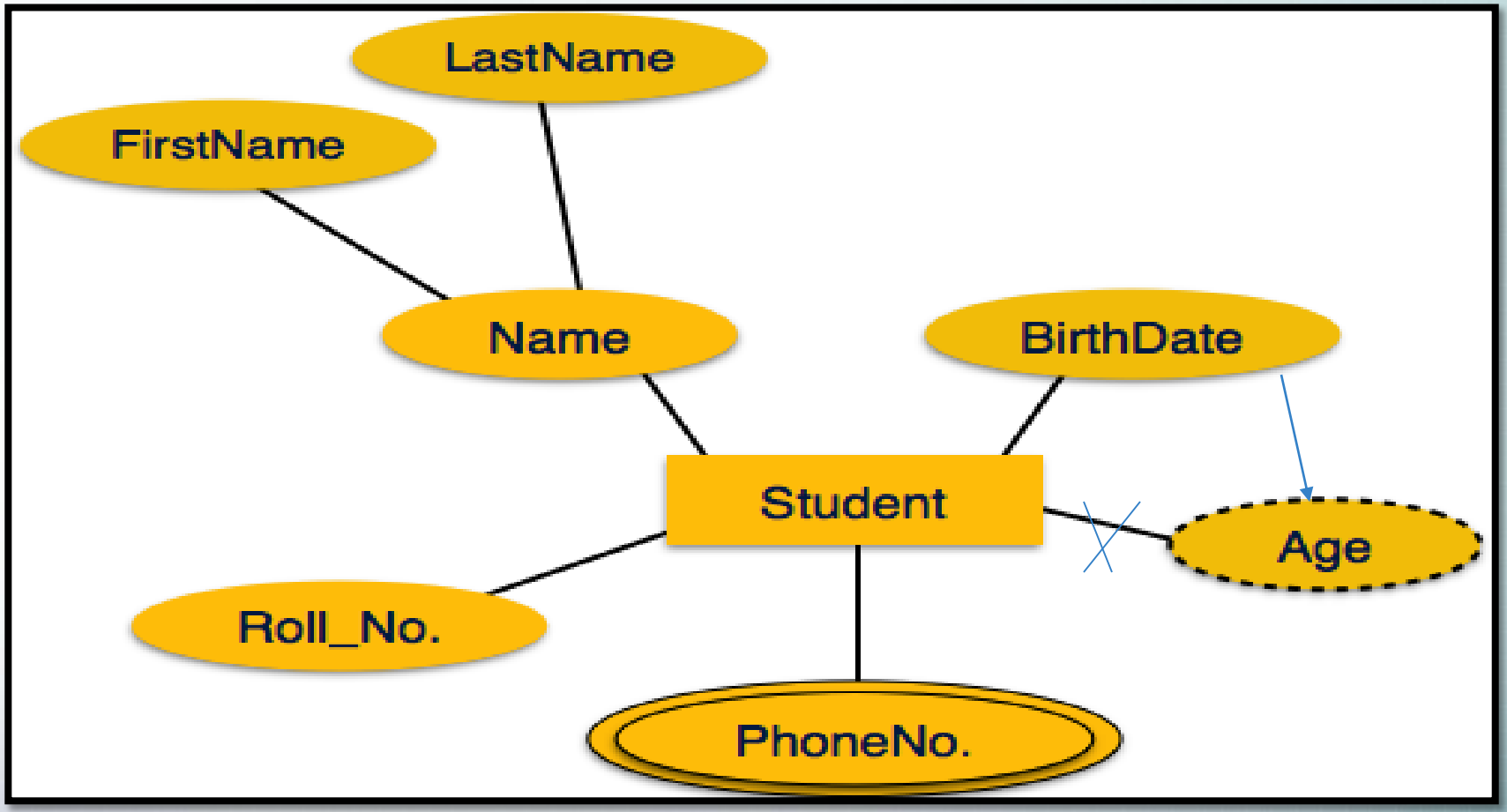
EXAMPLE: MULTIVALUED ATTRIBUTE



ATTRIBUTES TYPES

- **Derived Attribute**
  - Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.
  - For example, age can be derived from data_of_birth.

EXAMPLE: DERIVED ATTRIBUTE



Entity-Set and Keys

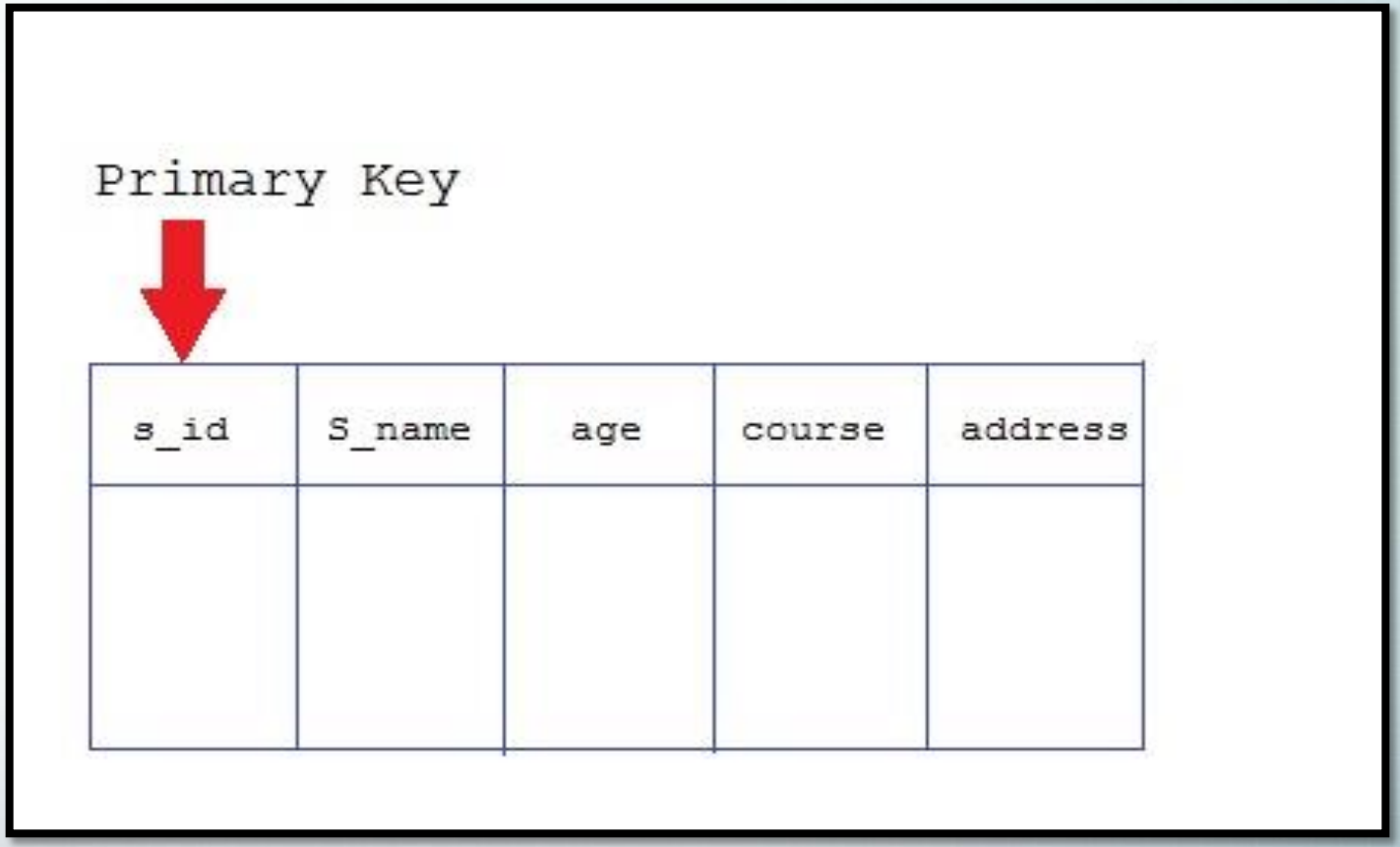
- **Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.**
- **For example, the roll number of a student makes him/her identifiable among students.**
- **Super Key** – is defined as a set of attributes within a table that uniquely identifies each record within a table.
- **Candidate Key** – Candidate keys are defined as the set of fields from which primary key can be selected. It is an attribute or set of attribute that can act as a primary key for a table to uniquely identify each record in that table.

Entity-Set and Keys

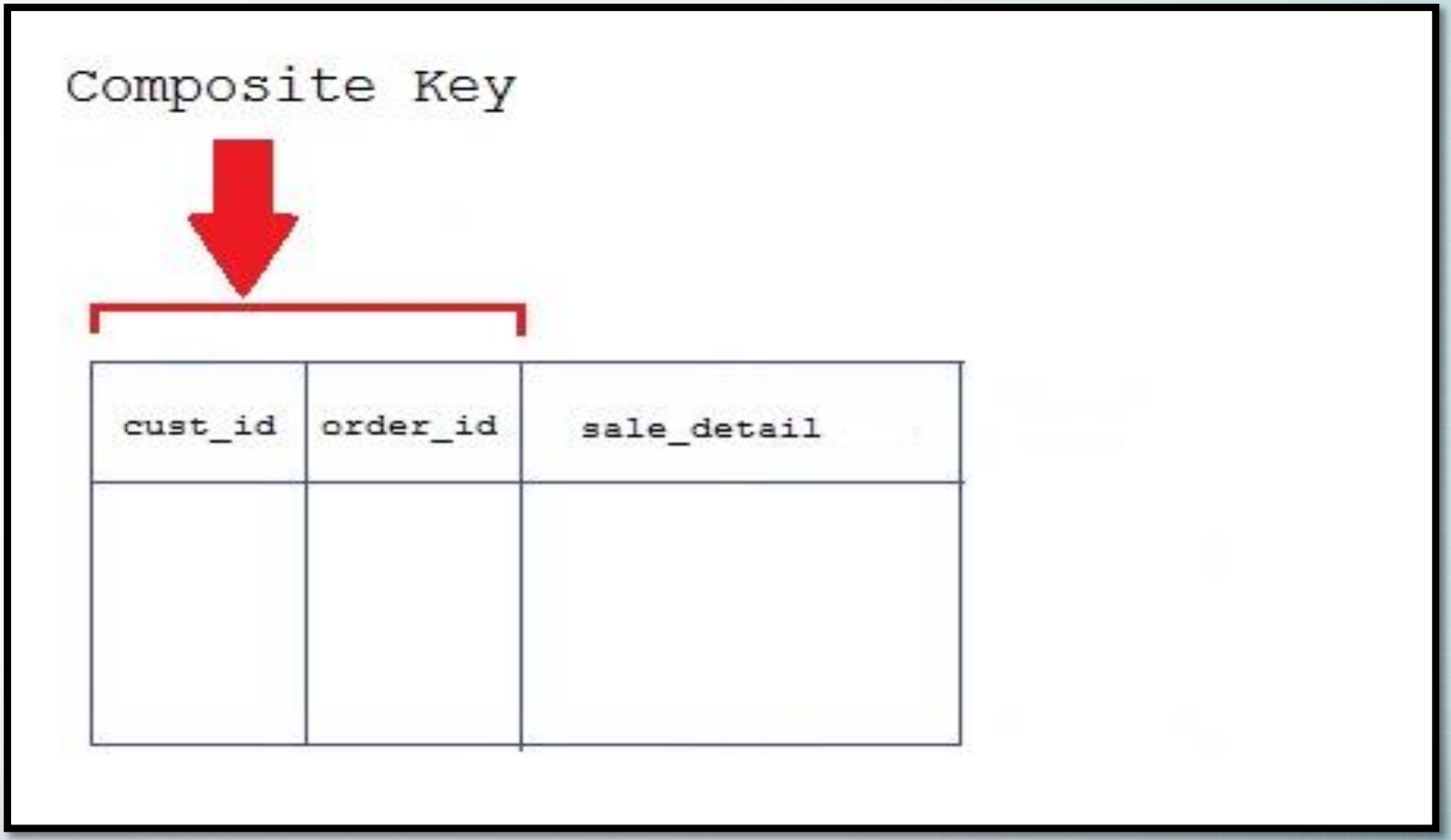
- **Primary Key** – A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

- **Composite Key** Key that consist of two or more attributes that uniquely identify an entity occurrence is called **Composite key**.

EXAMPLE: PRIMARY KEY



EXAMPLE: COMPOSITE KEY



Relationship

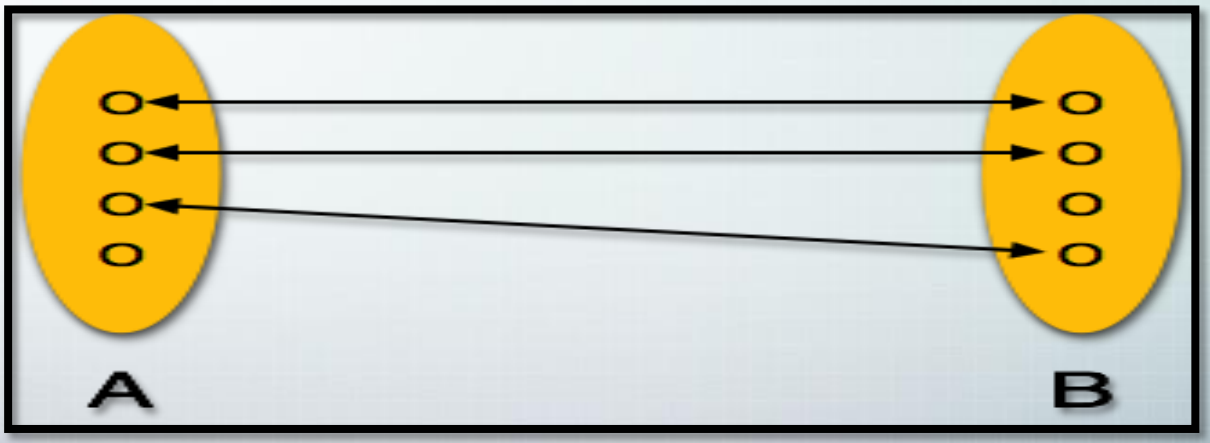
- The association among entities is called a relationship. For example, an employee **works_at** a department, a student **enrolls** in a course. Here, Works_at and Enrolls are called relationships.

- **Relationship Set:**
  - A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. *These attributes are called* ***descriptive attributes.***

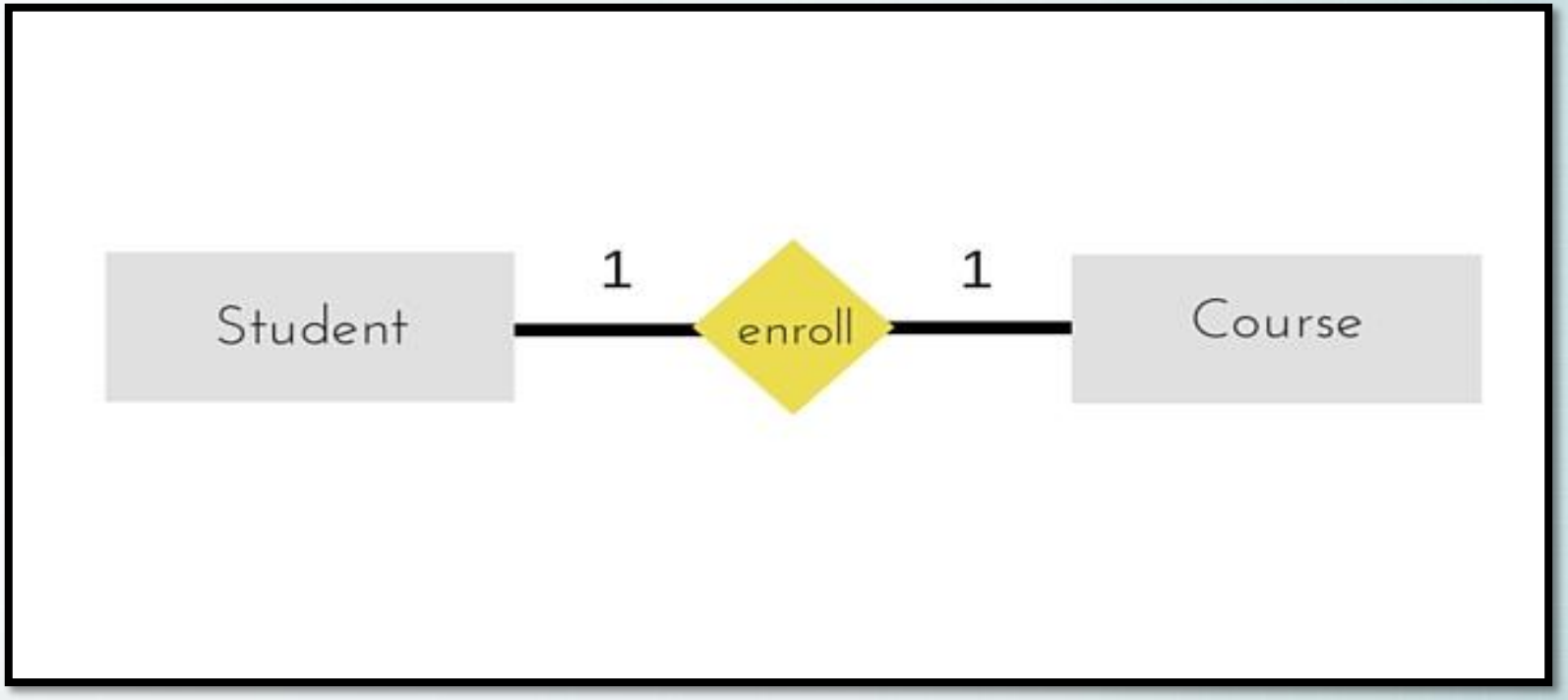
Mapping Cardinalities

- **Cardinality** *defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.*

- **One-to-one** – *One entity from entity set A can be associated with at most one entity of entity set B and vice versa.*

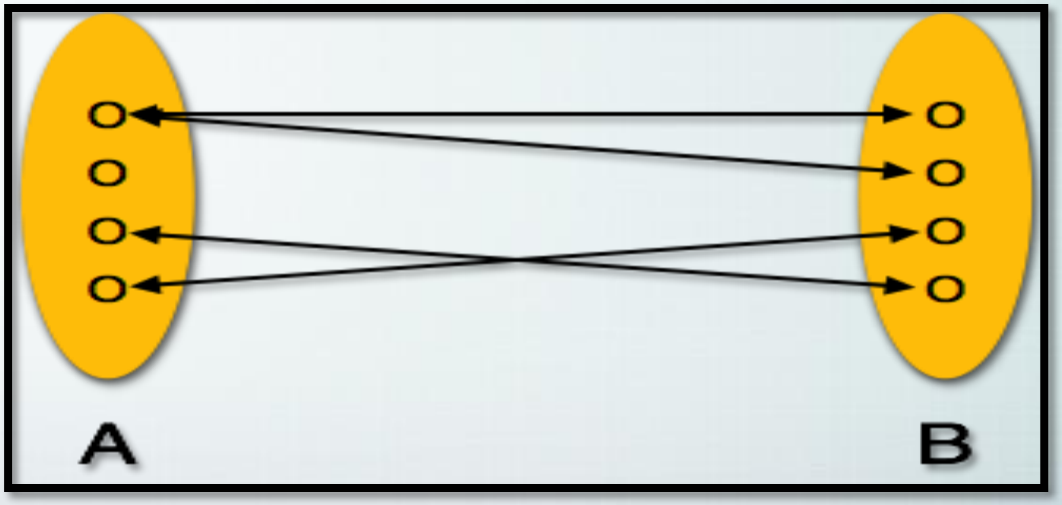


EXAMPLE: 1 – 1

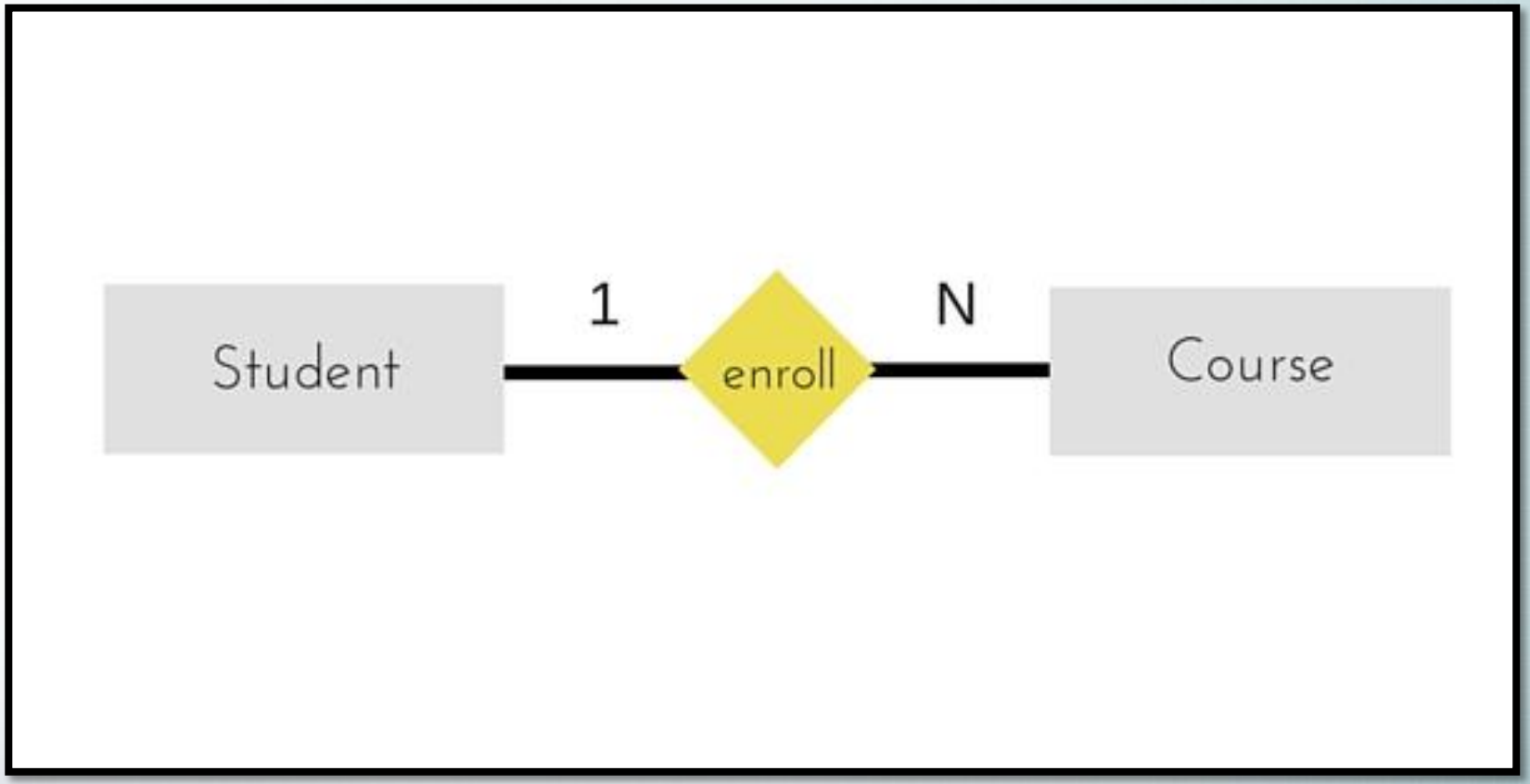


Mapping Cardinalities

- **One-to-many** – One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

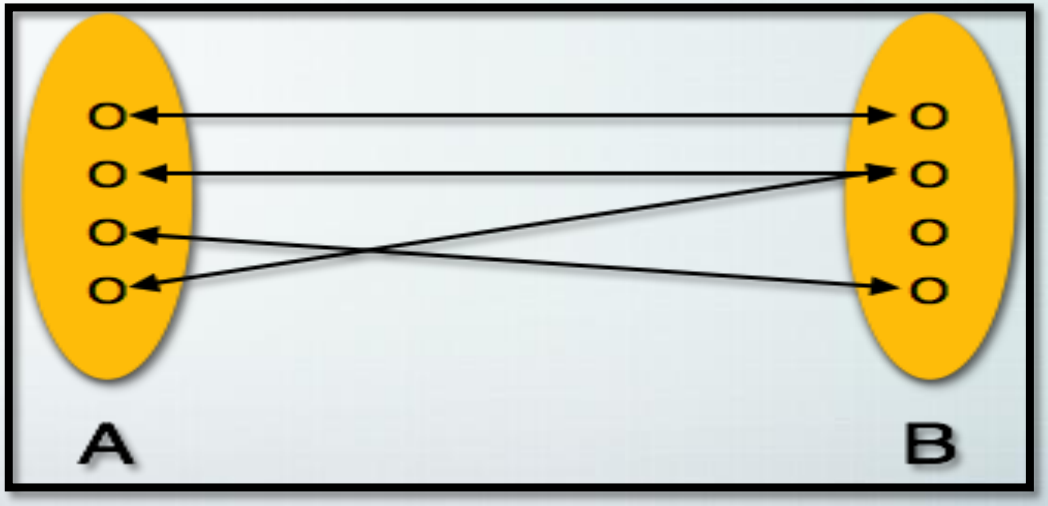


EXAMPLE: 1 - N

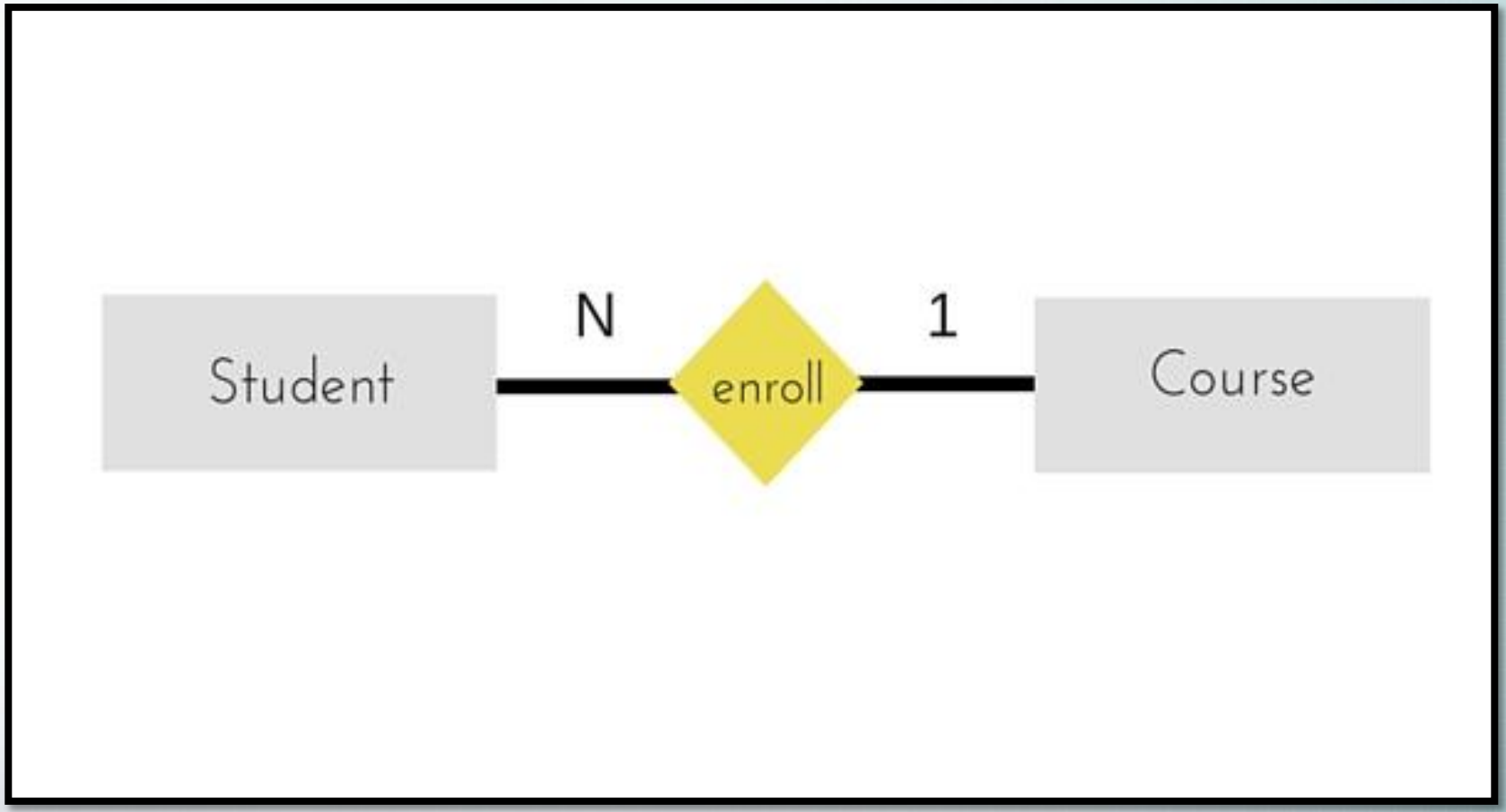


Mapping Cardinalities

- **Many-to-one** – More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

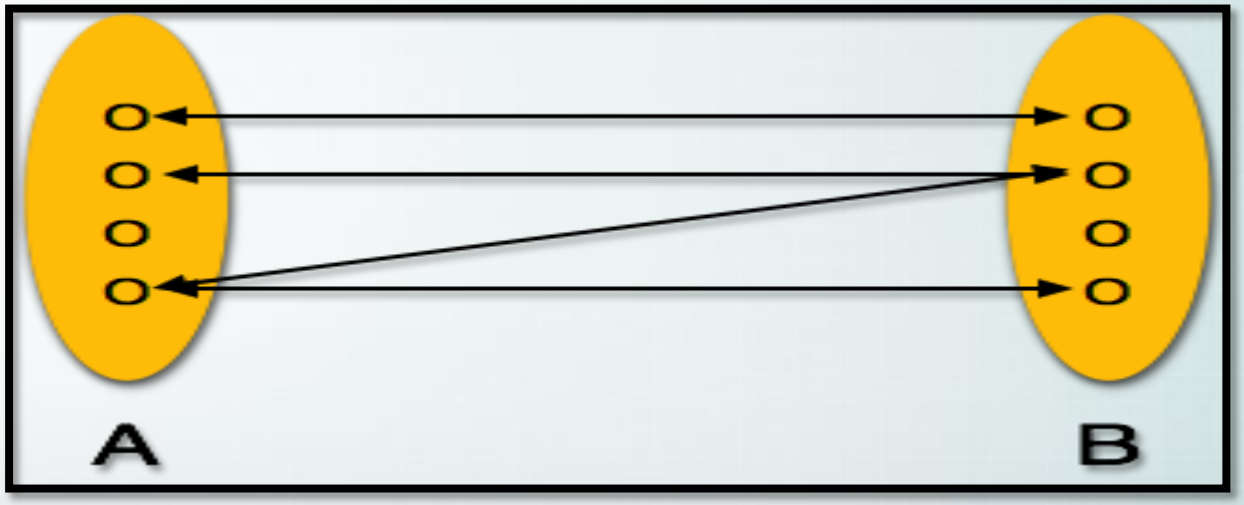


EXAMPLE: N - 1

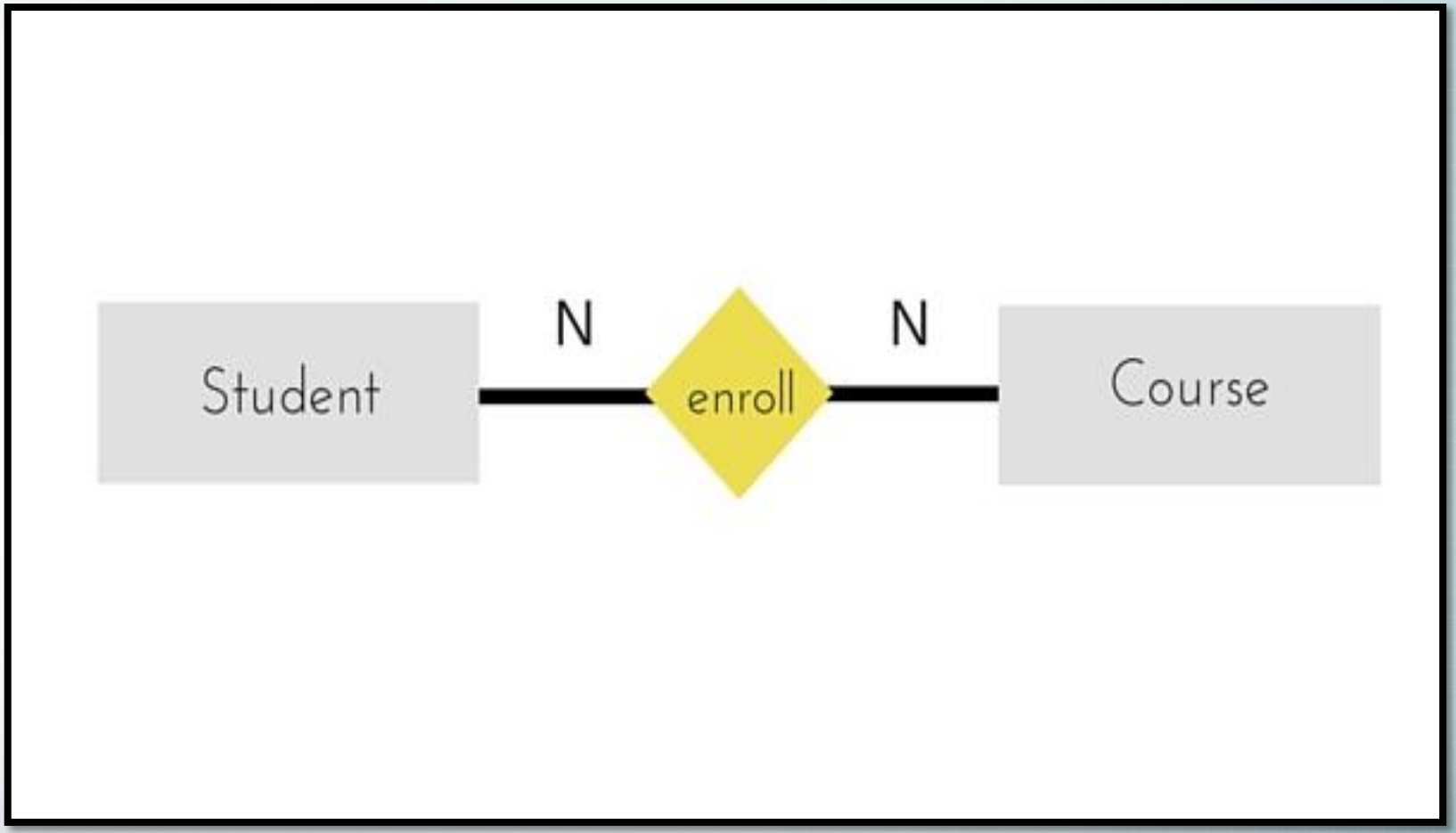


Mapping Cardinalities

- **Many-to-many** – One entity from A can be associated with more than one entity from B and vice versa.



EXAMPLE: N - N



Normalisation

- Normalisation is a 'fancy' term for a set of rules, designed to make sure that a database is organised in the best way possible
- This allows the data to be processed more efficiently and any query to be processed.
- These rules depend on relationships being established between the entities to create a functional dependency between them.

The normalisation process involves:

- **Finding and grouping together all the entities and their attributes.**
- **Removing repeating groups of data.**
- **Providing unique keys for each entity in the database system.**

The Three Major Stages of Normalisation:

- **First Normal Form**
  - 1NF is the first level of normalisation. An entity (table) is in First Normal form if it contains no repeating attributes (fields) or groups of attributes.
- **Second Normal Form**
  - An entity is in 2NF if no attribute (not part of the primary key) is dependent on only part of the primary key. This only applies to entities with concatenated primary keys.
- **Third Normal Form**
  - An entity is in 3NF if all attributes are entirely dependent on the primary key and not on any attribute that is not part of the primary key.
  - In a relational schema, each tuple is divided into fields called Domains.

Functional Dependency

Functional dependency means that there must be only a one-to-one dependency for each attribute mapped from a primary key to that attribute.

It defines a relationship in which the existence of one entity/attribute is entirely dependent on the existence of another (one-to-one).

Functional Dependency - Practical Example

SALES
Order Number
Acc.No.
Customer
Address
Date
Item
Quantity
Item Price
Total Cost

Order Number is the primary key.

The value for each attribute of SALES, except Item Price, depends upon the value of the primary key.

All attributes of SALES, except Item Price, are Functionally Dependent on the primary key

Item Price is Functionally Dependent on the the attribute Item

To produce a set of entities in First Normal Form (1NF):

- Remove repeating (multiple) groups within the primary entities (tables) so that each record (row) within the entity is the same length.
- Repeating groups then become new entities, linked together by a one-to-many relationship.
- Relationships are created by including a primary key from one entity as a foreign key in another entity

Al's Baker Shop

Order No.	Acc. No.	Customer	Address	Date	Item	Qty.	Item Price	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	Bakewell Tart	20	0.15	12.35
					Danish Pastry	13	0.20	
					Apple Pie	45	0.15	
4633	526	Smiths	12 Dee View, Aberdeen	19/7	Butteries	120	0.20	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Apple Pie	130	0.15	56.50
					Cherry Pie	100	0.18	
					Steak Pie	30	0.50	
					Meringue Pie	20	0.20	
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	Apple Pie	15	0.15	7.50
					Danish Pastry	50	0.20	

Al's Baker Shop

Order No.	Acc. No.	Customer	Address	Date	Item	Qty.	Item Price	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	Bakewell Tart	20	0.15	12.35
					Danish Pastry	13	0.20	
					Apple Pie	45	0.15	
4633	526	Smiths	12 Dee View, Aberdeen	19/7	Butteries	120	0.20	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	Apple Pie	130	0.15	56.50
					Cherry Pie	100	0.18	
					Steak Pie	30	0.50	
					Meringue Pie	20	0.20	
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	Apple Pie	15	0.15	7.50
					Danish Pastry	50	0.20	

Al's Baker Shop

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Orders

Item	Qty.	Item Price
Bakewell Tart	20	0.15
Danish Pastry	13	0.20
Apple Pie	45	0.15
Butteries	120	0.20
Apple Pie	130	0.15
Cherry Pie	100	0.18
Steak Pie	30	0.50
Meringue Pie	20	0.20
Apple Pie	15	0.15
Danish Pastry	50	0.20

Items Purchased

Al's Baker Shop

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Orders

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Items Purchased

Al's Baker Shop

Orders Table:

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Order No. can be used to uniquely identify each record and can therefore be made the primary key.

Orders (Order No.
Acc. No.
Customer
Address
Date
Total Cost)

Al's Baker Shop

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Items Purchased Table:

No one attribute can be used to uniquely identify a record.

Concatenated key is required

Order No. and Item together can uniquely identify a record.

Items Purchased

(*Order No.
Item
Quantity
Item Price)

Al's Baker Shop
First Normal Form

Orders	(Order No. Acc. No. Customer Address Date Total Cost)

Items Purchased	(*Order No. Item Quantity Item Price

Q & A

1. What is Normalisation?

2. What does Normalisation process involves?

3. What is Ist Normal Form?

4. In a relational schema, each tuple is divided into fields called Domains. T/F

The Three Major Stages of Normalisation:

- **First Normal Form**
  - 1NF is the first level of normalisation. An entity (table) is in First Normal form if it contains no repeating attributes (fields) or groups of attributes.
- **Second Normal Form**
  - An entity is in 2NF if no attribute (not part of the primary key) is dependent on only part of the primary key. This only applies to entries with concatenated primary keys.
- **Third Normal Form**
  - An entity is in 3NF if all attributes are entirely dependent on the primary key and not on any attribute that is not part of the primary key.

To produce a set of entities in Second Normal Form (2NF):

- Test for dependency by testing each particular attribute in turn to check that it can be uniquely identified by making use of all the primary key. This test need not be completed unless you have at least one table which requires a concatenated Primary Key

- Remove all partially dependent attributes to a new entity.

N.B. – A concatenated key occurs when you need two fields together in order to uniquely identify a record

Al's Baker Shop
First Normal Form

Orders	(Order No. Acc. No. Customer Address Date Total Cost)

Items Purchased	(*Order No. Item Quantity Item Price)

Al's Baker Shop

Orders Table:

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Because this entity has a single attribute as the primary key there can be no partial dependencies and therefore the entity is already in 2NF.

Al's Baker Shop
(2NF Step 1)

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Items Purchased Table:
Primary key is Order No. and Item

Test for dependency by testing each particular attribute.

Primary Key	Attribute	Functionally Dependent?

Al's Baker Shop
(2NF Step 1)

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Items Purchased Table:
Primary key is Order No. and Item

Test for dependency by testing each particular attribute.

Primary Key	Attribute	Functionally Dependent?
Order No Item	Quantity	YES Quantity is functionally dependent on Order No. and Item.

Al's Baker Shop
(2NF Step 1)

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Items Purchased Table:
Primary key is Order No. and Item

Test for dependency by testing each particular attribute.

Primary Key	Attribute	Functionally Dependent?
Order No Item	Quantity	YES Quantity is functionally dependent on Order No. and Item.
Order No Item	Item Price	NO Item price is functionally dependent on Item, but not on Order No. and Item

Al's Baker Shop
(2NF Step 2)

Remove any partially dependent attributes to a new entity

Order No.	Item	Qty.	Item Price
7823	Bakewell Tart	20	0.15
7823	Danish Pastry	13	0.20
7823	Apple Pie	45	0.15
4633	Butteries	120	0.20
2276	Apple Pie	130	0.15
2276	Cherry Pie	100	0.18
2276	Steak Pie	30	0.50
2276	Meringue Pie	20	0.20
1788	Apple Pie	15	0.15
1788	Danish Pastry	50	0.20

Item Price
0.15
0.20
0.15
0.20
0.15
0.18
0.50
0.20
0.15
0.20

Al's Baker Shop
(2NF Step 2)

Remove any partially dependent attributes to a new entity

Order No.	Item	Qty.
7823	Bakewell Tart	20
7823	Danish Pastry	13
7823	Apple Pie	45
4633	Butteries	120
2276	Apple Pie	130
2276	Cherry Pie	100
2276	Steak Pie	30
2276	Meringue Pie	20
1788	Apple Pie	15
1788	Danish Pastry	50

Part Order

Item Price
0.15
0.15
0.20
0.18
0.20
0.20
0.50

Price List

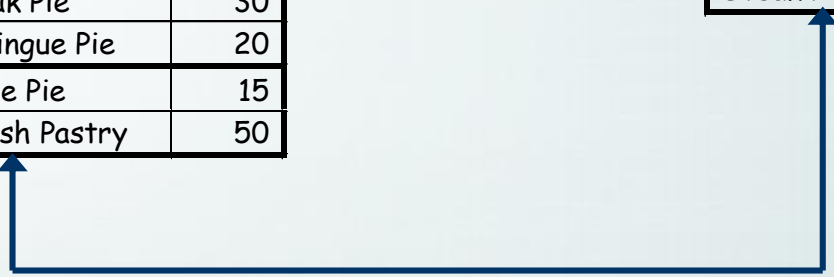
Al's Baker Shop
(2NF Step 2)

Create a relationship between the tables

and assign Primary Keys

Order No.	Item	Qty.
7823	Bakewell Tart	20
7823	Danish Pastry	13
7823	Apple Pie	45
4633	Butteries	120
2276	Apple Pie	130
2276	Cherry Pie	100
2276	Steak Pie	30
2276	Meringue Pie	20
1788	Apple Pie	15
1788	Danish Pastry	50

Item	Item Price
Apple Pie	0.15
Bakewell Tart	0.15
Butteries	0.20
Cherry Pie	0.18
Danish Pastry	0.20
Meringue Pie	0.20
Steak Pie	0.50



Part Order

Primary Key: Order No.
and *Item

Price List

Primary Key: Item

Al's Baker Shop
Second Normal Form

Orders	(Order No. Acc. No. Customer Address Date Total Cost)

Price List	(Item Item Price)

Part Order	(*Order No. *Item Quantity)

The Three Major Stages of Normalisation:

First Normal Form

- 1NF is the first level of normalisation. An entity (table) is in First Normal form if it contains no repeating attributes (fields) or groups of attributes.

Second Normal Form

- An entity is in 2NF if no attribute (not part of the primary key) is dependent on only part of the primary key. This only applies to entries with concatenated primary keys.

Third Normal Form

- An entity is in 3NF if all attributes are entirely dependent on the primary key and not on any attribute that is not part of the primary key.

To produce a set of entities in Third Normal Form (3NF):

- **Test each attribute in turn to check for dependency on the primary key.**
- **Remove all transitive dependencies to a new entity.**
  - A transitive dependency is where an attribute is dependent on another attribute (or attributes) that is (are) NOT the primary key

Al's Baker Shop
Second Normal Form

Orders	(Order No. Acc. No. Customer Address Date Total Cost)

Price List	(Item Item Price)

Part Order	(*Order No. *Item Quantity)

Al's Baker Shop - 3NF Step1

Test for dependency

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency

Al's Baker Shop - 3NF Step1

Test for dependency

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency
Order No.	Acc.No.	YES: **Acc.No** can be found if we know either **Customer** or **Address**

Al's Baker Shop - 3NF Step1

Test for dependency

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	16/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency
Order No.	Acc.No.	YES: **Customer** can be found if we know either **Acc.No.** or **Address**
Order No.	**Customer**	**YES** **Customer** can be found if we know either **Acc.No.** or **Address**

Al's Baker Shop - 3NF Step1

Test for dependency

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	16/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency
Order No.	Acc.No.	YES: **Customer** can be found if we know either **Acc.No.** or **Address**
Order No.	Customer	YES **Acc.No.** can be found if we know either **Customer** or **Address**
Order No.	**Address**	**YES** **Address** can be found if we know either **Customer** or **Acc.No**

Al's Baker Shop - 3NF Step1

Test for dependency

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency
Order No.	Acc.No.	YES: **Customer** can be found if we know either **Acc.No.** or **Address**
Order No.	Customer	YES **Acc.No.** can be found if we know either **Customer** or **Address**
Order No.	Address	YES **Address** can be found if we know either **Customer** or **Acc.No**
Order No.	**Date**	**NO** Dependent on Order No.

Al's Baker Shop - 3NF Step1

Test for dependency

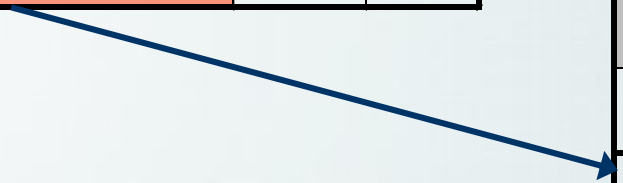
Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	19/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50

Primary Key	Attribute	Trasnsitive Dependency	
Order No.	Acc.No.	YES:	**Customer** can be found if we know either **Acc.No.** or **Address**
Order No.	Customer	YES	**Acc.No.** can be found if we know either **Customer** or **Address**
Order No.	Address	YES	**Address** can be found if we know either **Customer** or **Acc.No**
Order No.	Date	NO	Dependent on Order No.
Order No.	**Total Cost**	**NO**	Dependent on Order No.

Al's Baker Shop - 3NF Step2

Remove transitive dependencies to a new entity

Order No.	Acc. No.	Customer	Address	Date	Total Cost
7823	178	Daisy's Café	27 Bay Drive, Cove	16/7	12.35
4633	526	Smiths	12 Dee View, Aberdeen	16/7	24.00
2276	167	Sally's Snacks	3 High Street, Banchory	17/7	56.50
1788	032	Tasty Bite	17 Wood Place, Insch	18/7	7.50



Acc. No.	Customer	Address
178	Daisy's Café	27 Bay Drive, Cove
526	Smiths	12 Dee View, Aberdeen
167	Sally's Snacks	3 High Street, Banchory
032	Tasty Bite	17 Wood Place, Insch

Al's Baker Shop - 3NF Step2

Remove transitive dependencies to a new entity

Orders

Order No.	Date	Total Cost
7823	16/7	12.35
4633	16/7	24.00
2276	17/7	56.50
1788	18/7	7.50

Customers

Acc. No.	Customer	Address
178	Daisy's Café	27 Bay Drive, Cove
526	Smiths	12 Dee View, Aberdeen
167	Sally's Snacks	3 High Street, Banchory
032	Tasty Bite	17 Wood Place, Insch

Al's Baker Shop - 3NF Step2

Create a relationship between the tables and assign Primary Keys

Orders

Order No.	Acc. No.	Date	To Cos
7823	178	16/7	
4633	526	16/7	
2276	167	17/7	
1788	032	18/7	

Customers

Acc. No.	Customer	Address
178	Daisy's Café	27 Bay Drive, Cove
526	Smiths	12 Dee View, Aberdeen
167	Sally's Snacks	3 High Street, Banchory
032	Tasty Bite	17 Wood Place, Insch

Primary Key: Order No.

Primary Key: Acc.No.

Al's Baker Shop
Third Normal Form

Customers	(Acc. No. Customer Address)

Orders	(Order No. *Acc. No. Date Total Cost)

Price List	(Item Item Price)

Part Order	(*Order No. *Item Quantity)

Normalisation Complete

The Six Point Plan

Normalisation -

1NF

1. Remove repeating groups to create a new entity
2. Create a relationship using one of the attributes that are left [Usually the primary key]

2NF

3. 'Check out' entities with concatenated keys. If any attribute is not fully dependent on both parts of the primary key remove it to create a new entity.
4. Create a relationship using one of the attributes that are left [Usually the primary key]

3NF

5. 'Check out' every entity. If any attribute is dependent on any attribute other than the primary key, remove it into a new entity.
6. Create a relationship using one of the attributes

Q & A

1. What is 2nd Normal Form?

2. In 2nd Normal Form which dependency is tested?

3. What is 3rd Normal Form?

4. What is Transitive Dependency?