Database Normalization and Entity Relationship (ER) Model

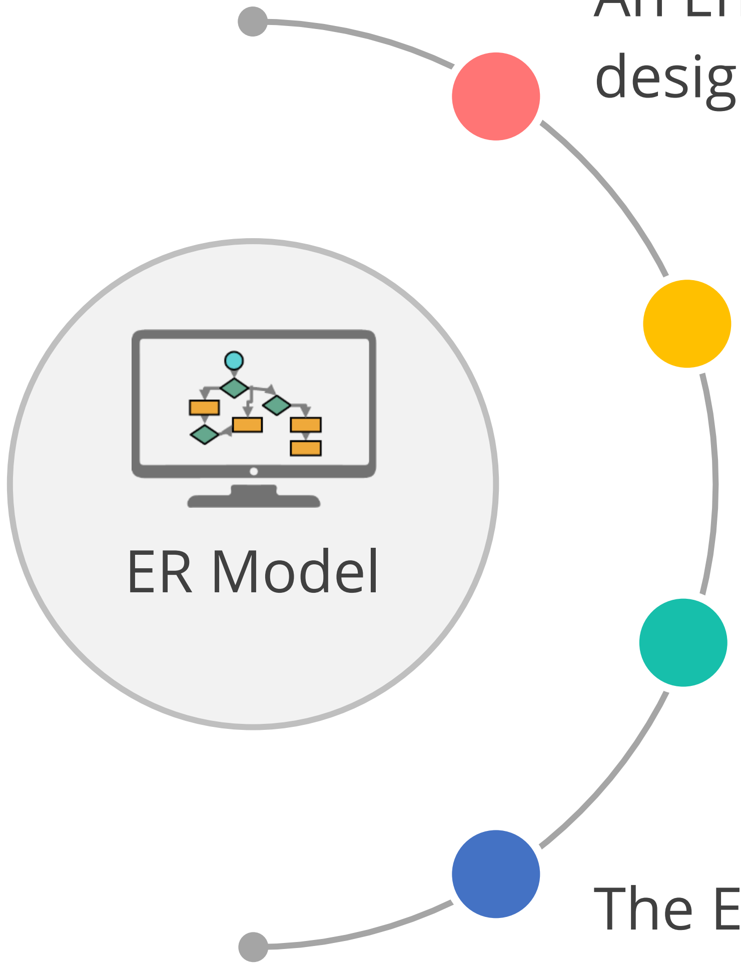
Entity-Relationship (ER) Model

Learning Objectives

By the end of this lesson, you will be able to:

- Interpret the Entity-Relationship Model(ER)
- List down the components of ER Diagram
- Create relationship sets
- Outline relationship degree

Entity-Relationship (ER) Model



An Entity–relationship model (ER model) is a data model which describes the design of database with the help of Entity-relationship Diagram.

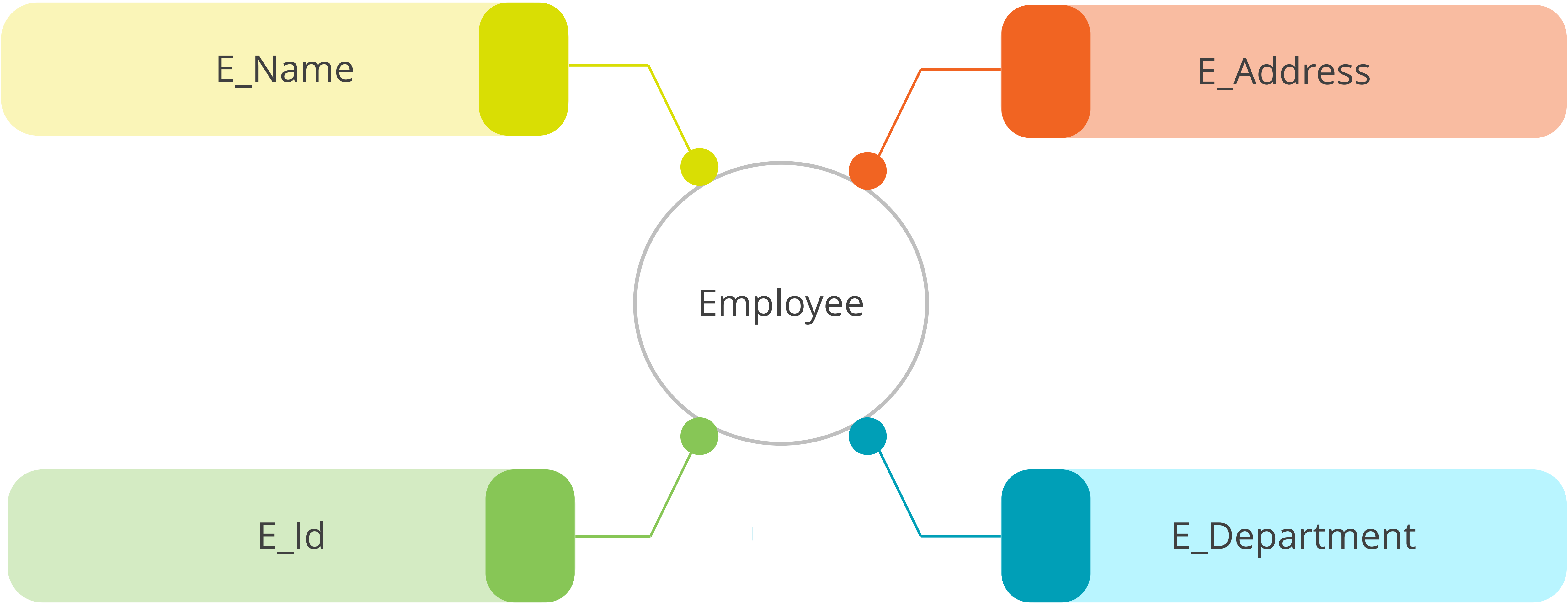
It is an architecture or blueprint of a database that may be implemented later as a database.

Entity set and relationship set are the two primary components of the E-R model.

The ER-Diagram is a graphical representation of data that shows how data is distributed and related to each other.

Entity-Relationship (ER) Model

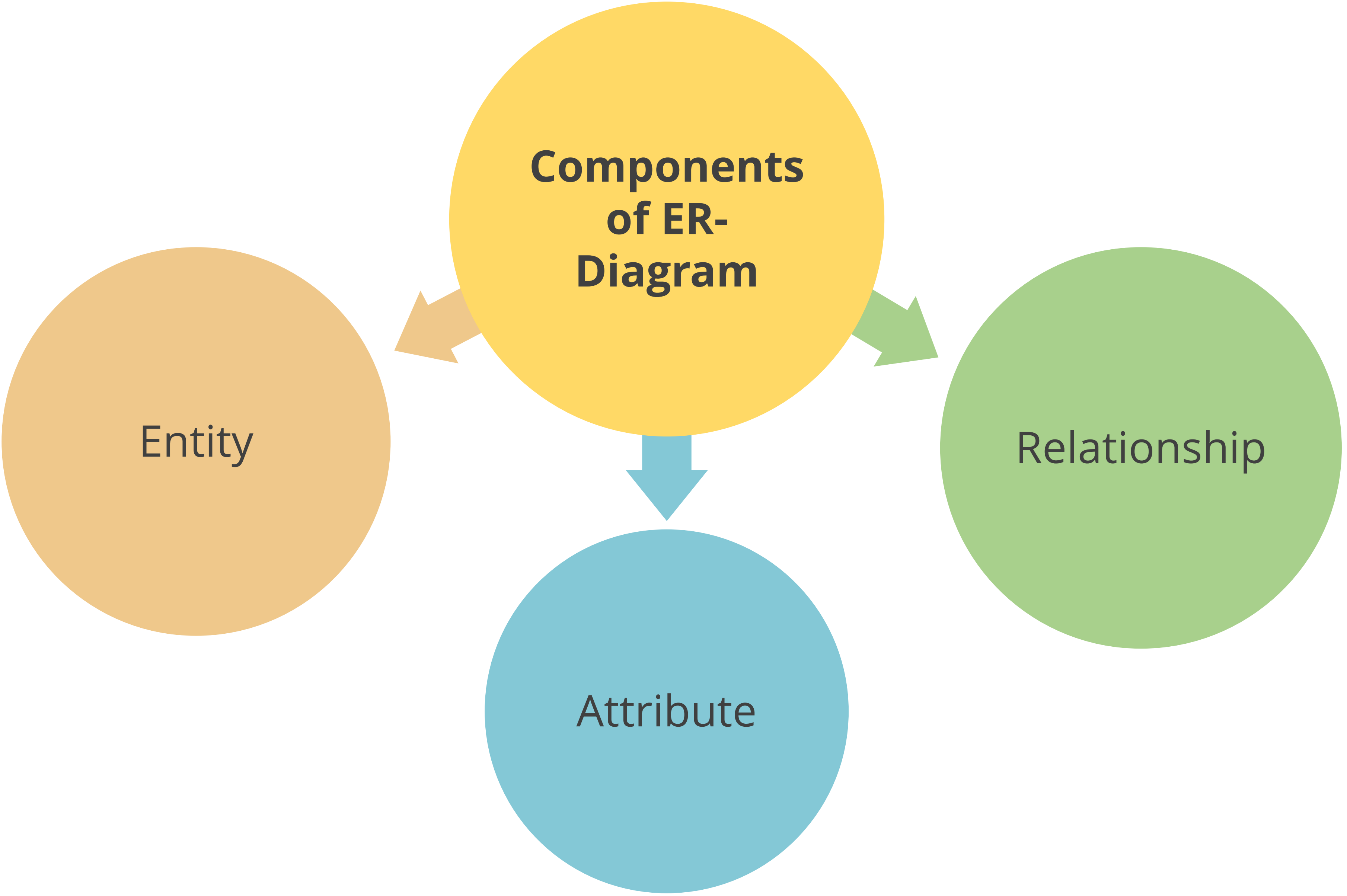
For example: Suppose you design an HR database, the employee will be an entity with the following attributes:



The address can be another entity with attributes, such as country, city, landmark, street name, and pin code, and there will be a relationship between them.

Components of ER Diagram

Components of (ER) Diagram



Components of ER-Diagram

Entity

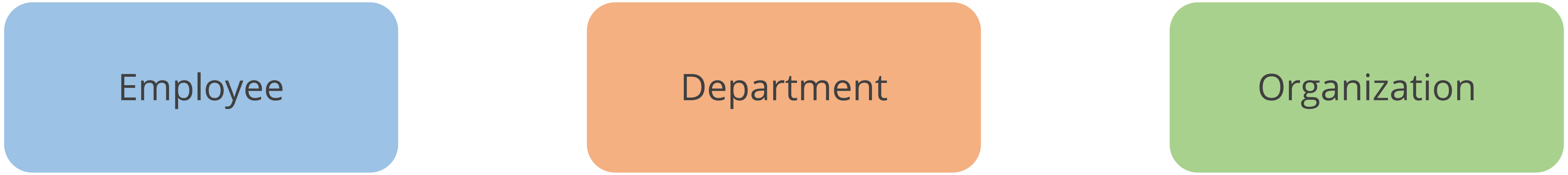
Attribute

Relationship

Entity

Entity

Any object, class, or component of data can be considered an entity. In an ER diagram, entities are represented by a rectangle.



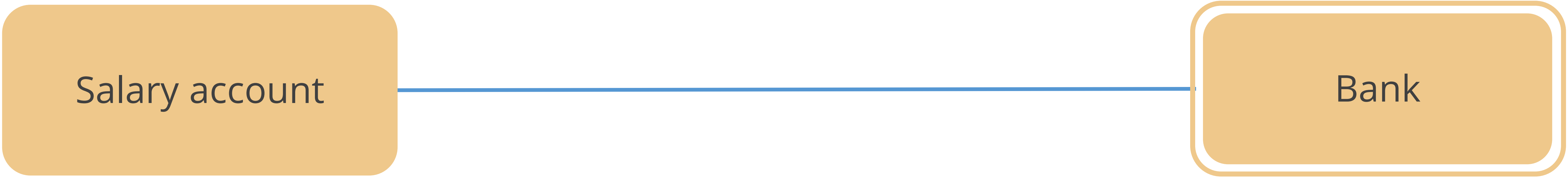
Employee

Department

Organization

Entity

Weak Entity: A weak entity is one that is reliant on another entity. There are no key attributes in the weak entity. A weak entity is represented by a double rectangle.

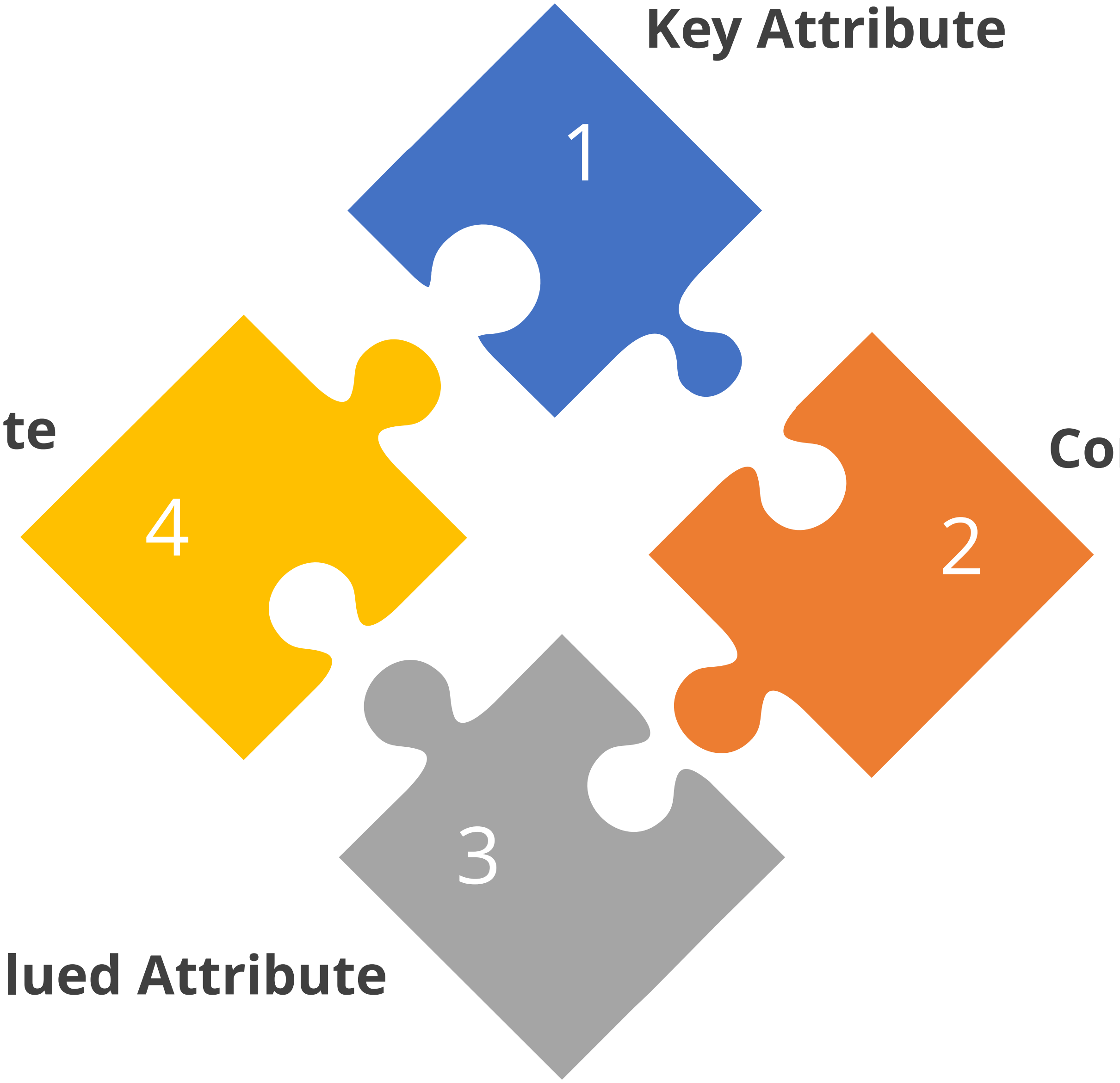


Attributes

Attributes

The properties of entities are known as attributes and an attribute is represented by ellipses.

There are four types of attributes:



Key Attribute

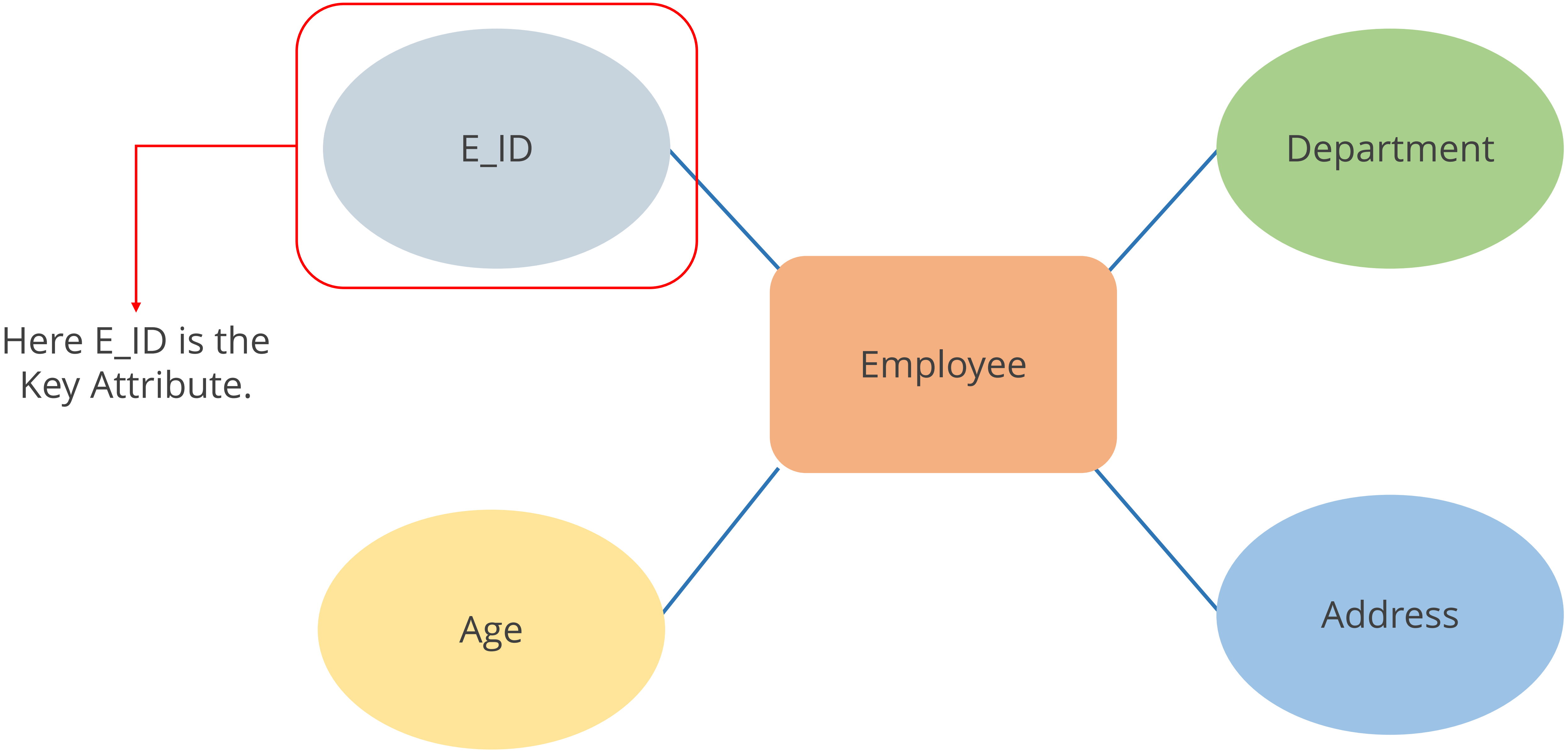
Composite Attribute

Multivalued Attribute

Derived Attribute

Key Attributes

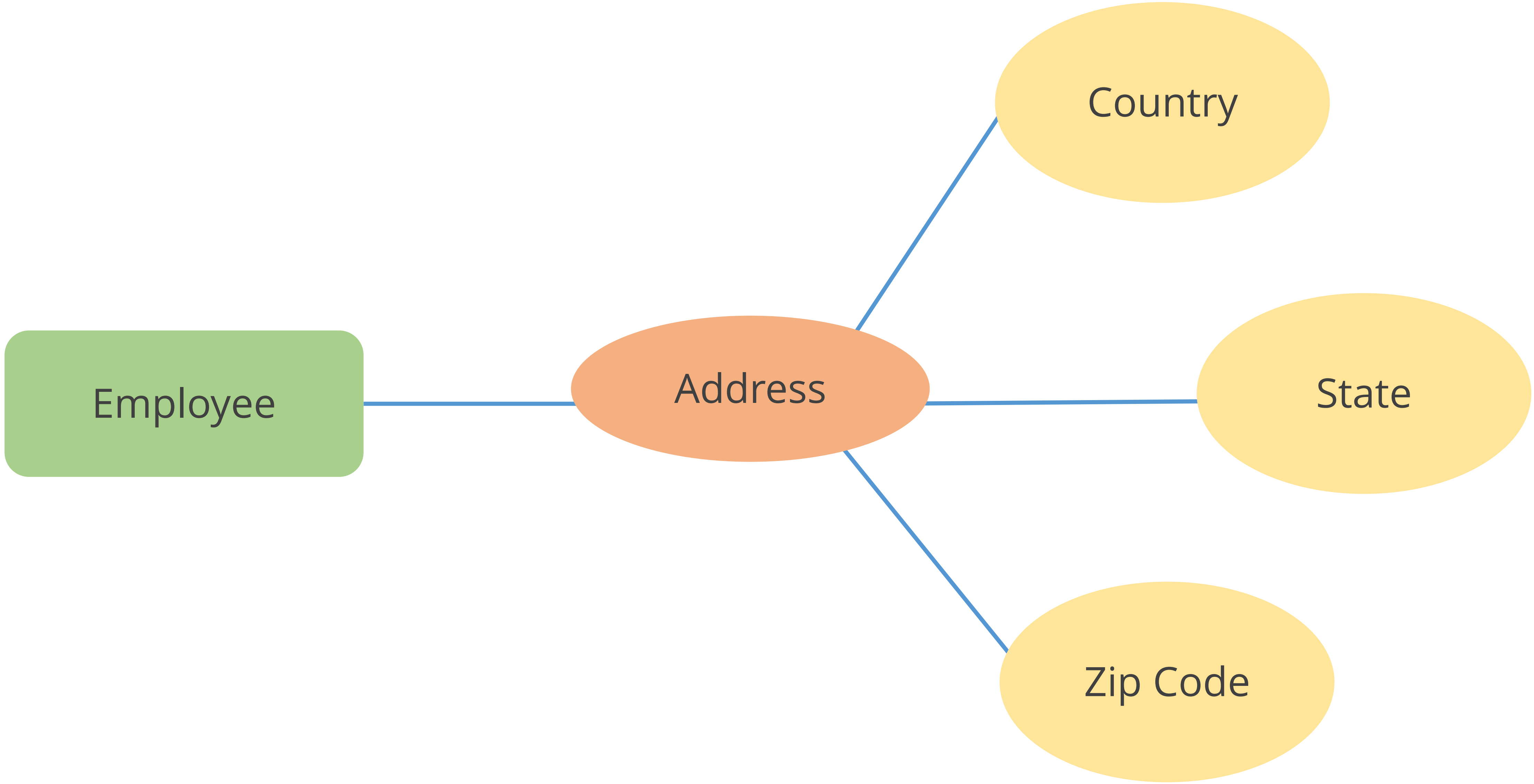
A key attribute can be used to identify one entity from a group of entities.



Here E_ID is the Key Attribute.

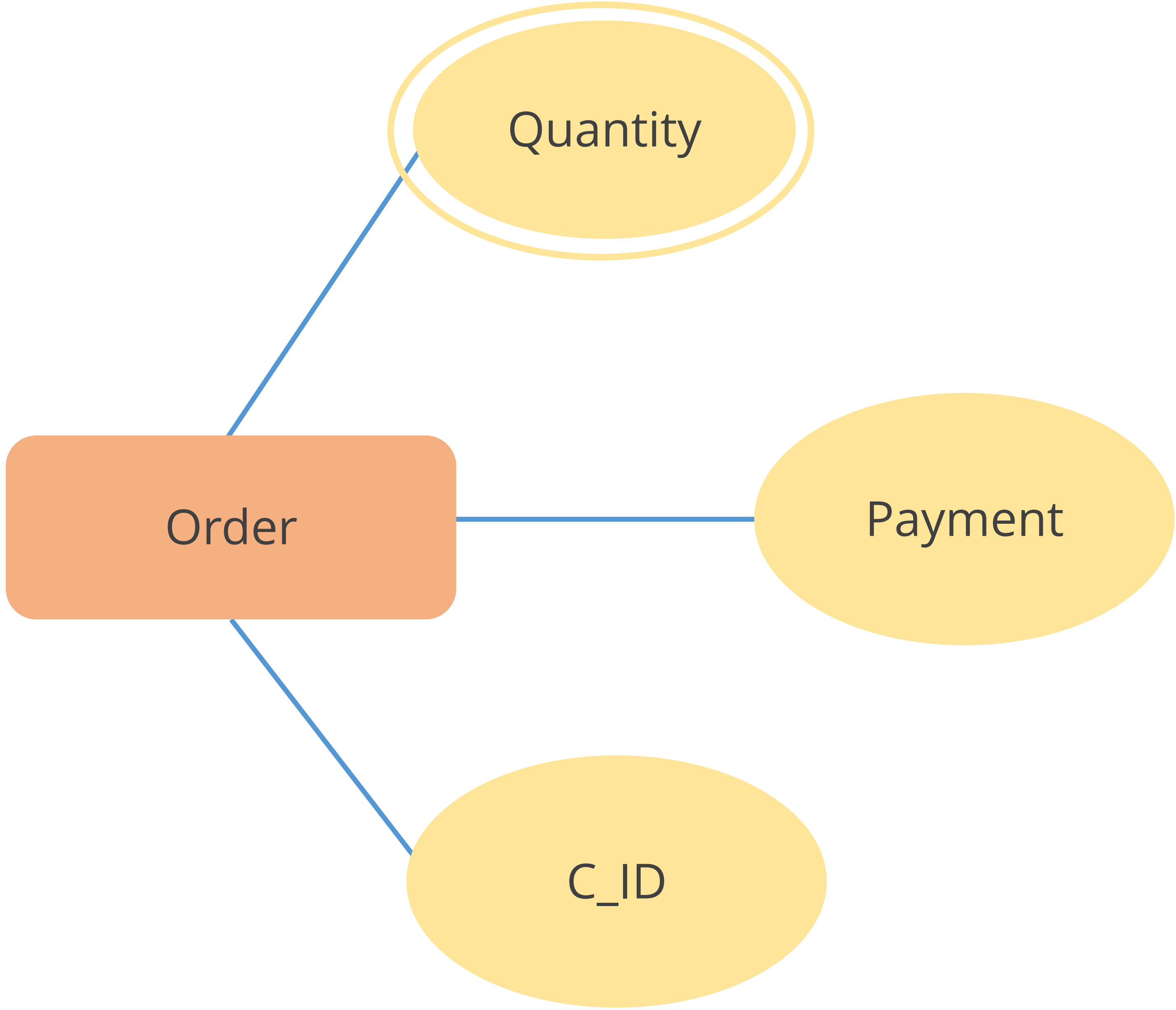
Composite Attributes

A composite attribute is an attribute that is composed of several other attributes.



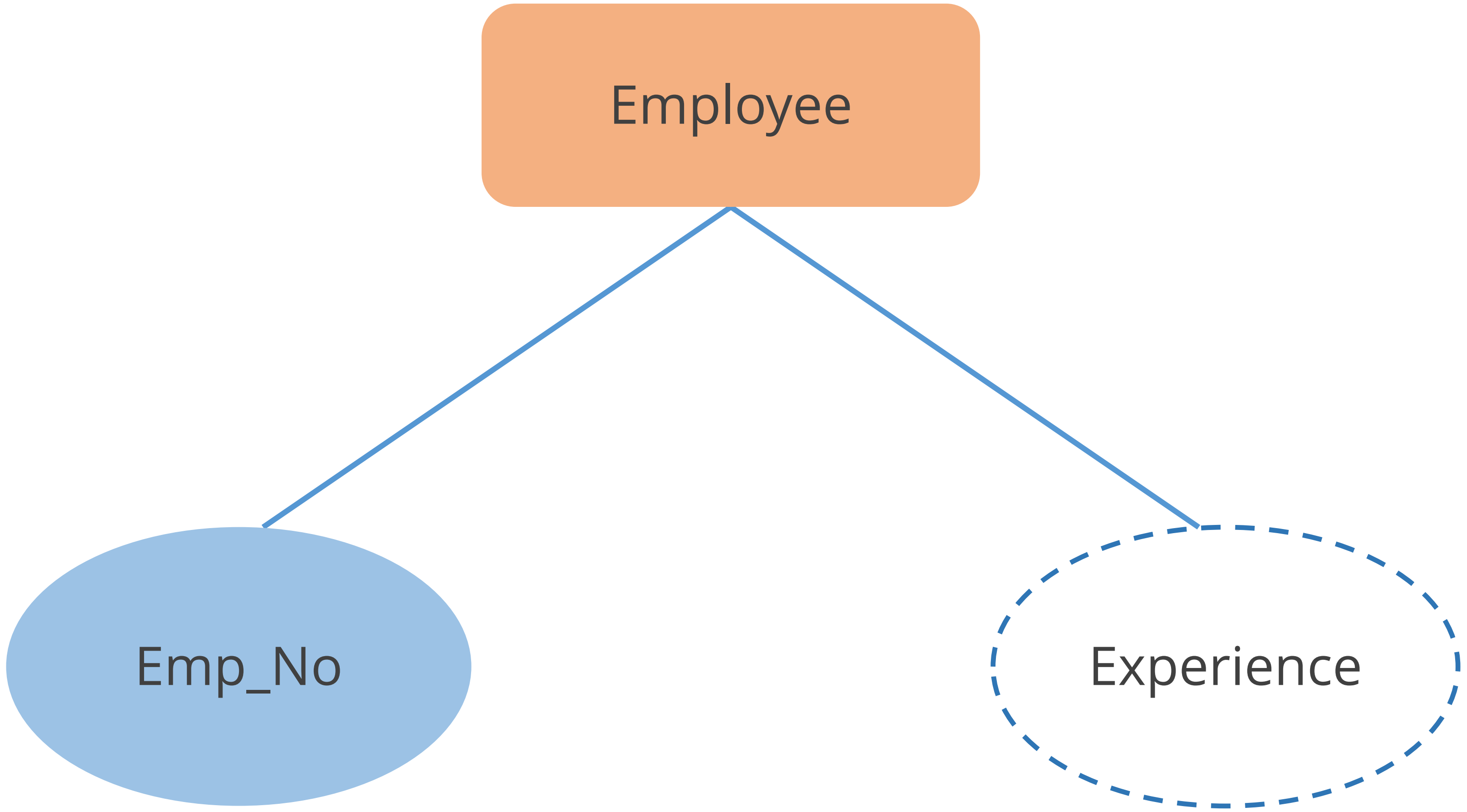
Multivalued Attributes

A multivalued attribute is a kind of attribute that has multiple values and is represented by a double oval.



Derived Attributes

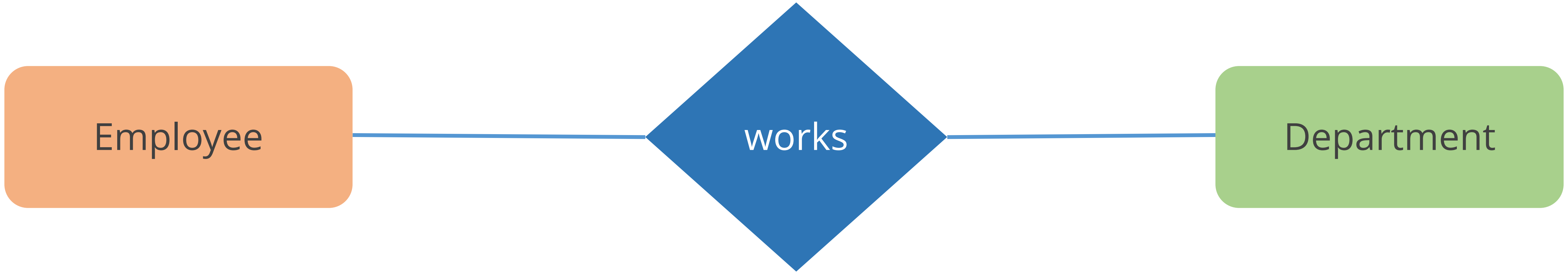
Derivative attributes are attributes that can be derived from other attributes. A dashed ellipse can be used to illustrate this.



Relationship

Relationship

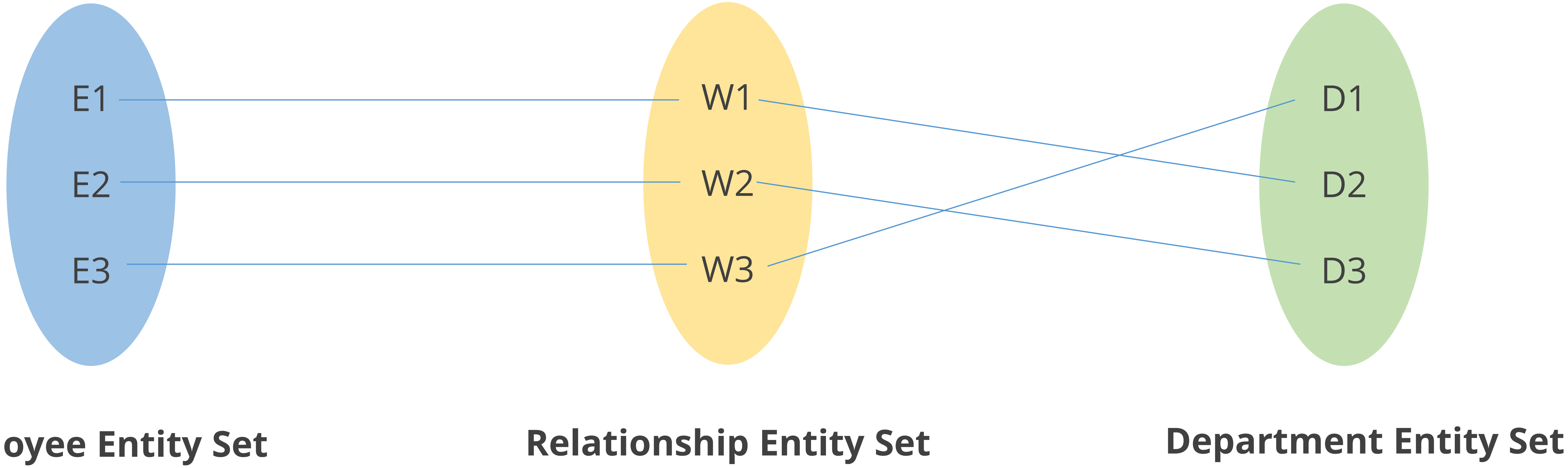
Relationship: A relationship is a term used to describe the connection between two or more entities. The relationship is represented as a diamond or rhombus.



Relationship Sets

Relationship Set

Relationship set: A relationship set is a group of similar types of relationships.



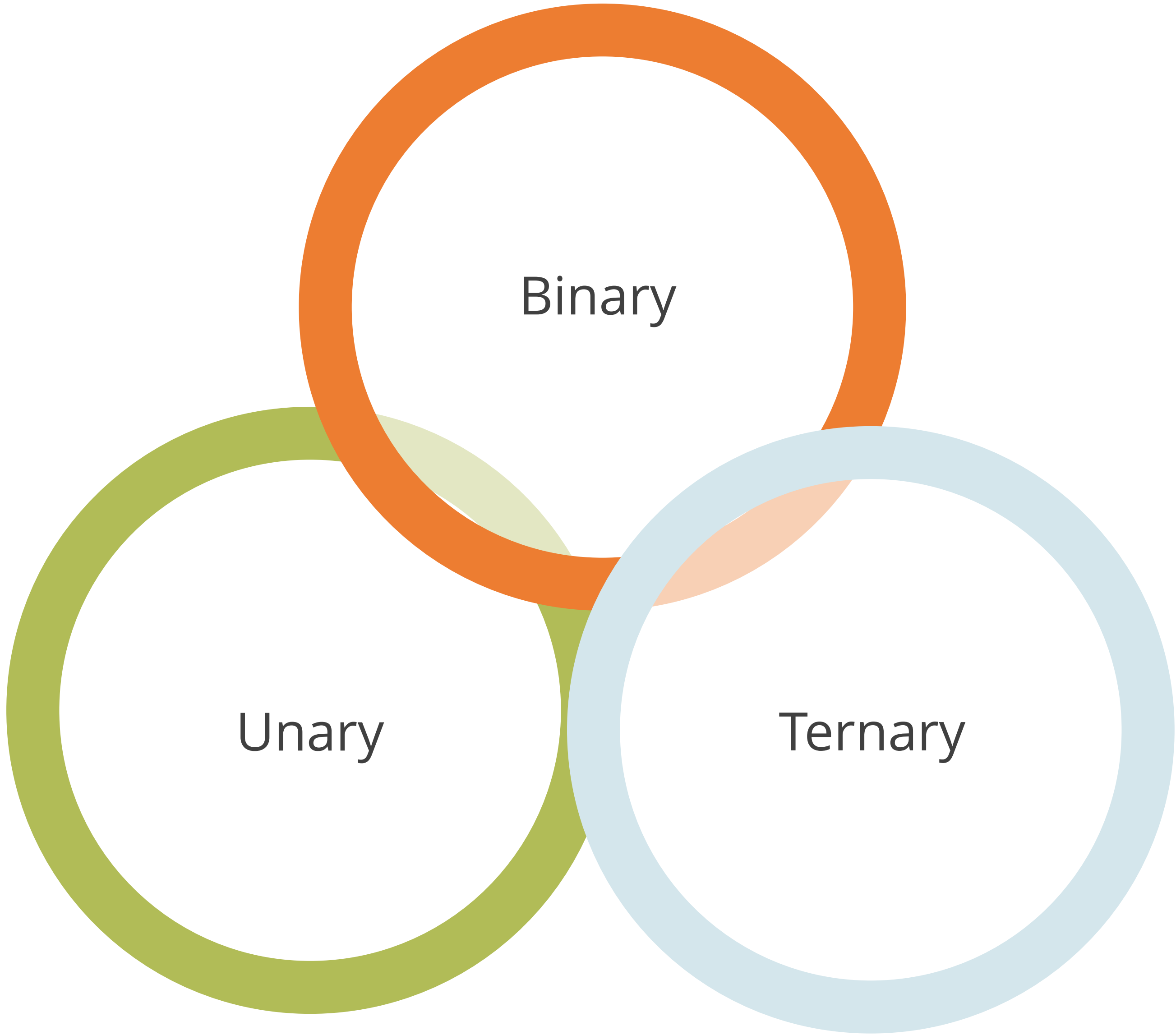
- The above relationship set depicts that E1 works in D2, E2 works in D3, and E3 works in D1.
- W1, W2, and W3 represent the relationship between employees and departments, indicating that each employee is associated with a department.

Relationship Degree

Relationship Degree

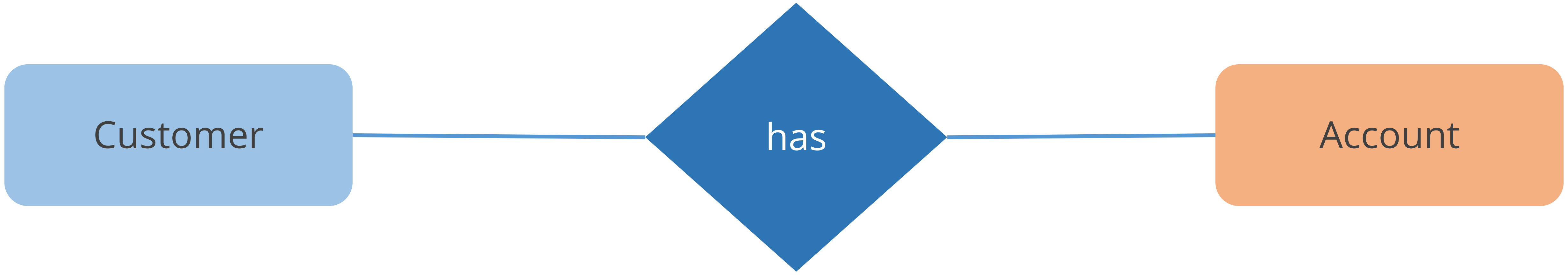
Relationship Degree: The degree of a relationship is determined by the number of entities that participate in it.

The three most common degrees of relationships in ER models are :



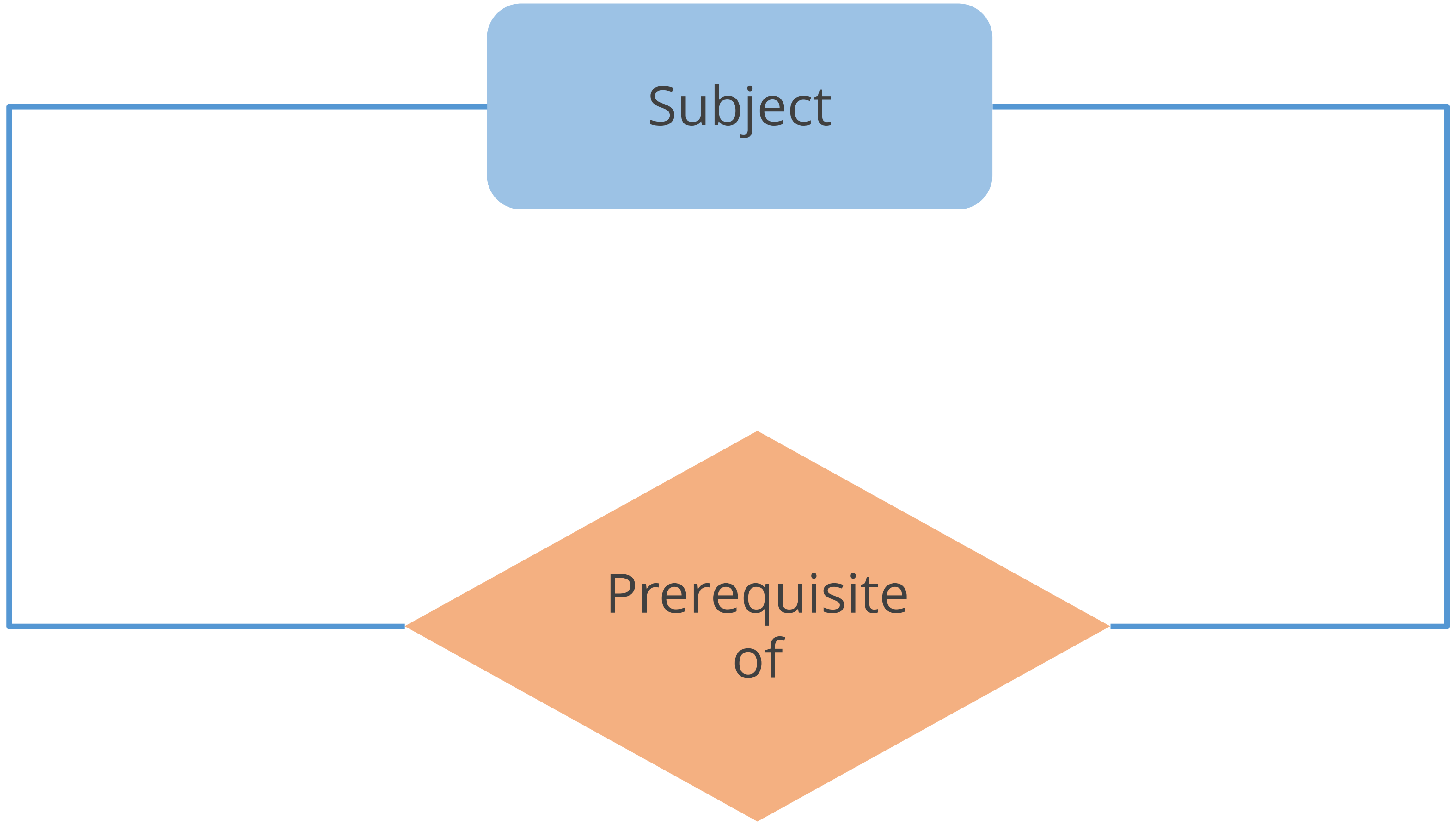
Binary Relationship

The most common type of relationship is a binary relationship, which involves two entities.



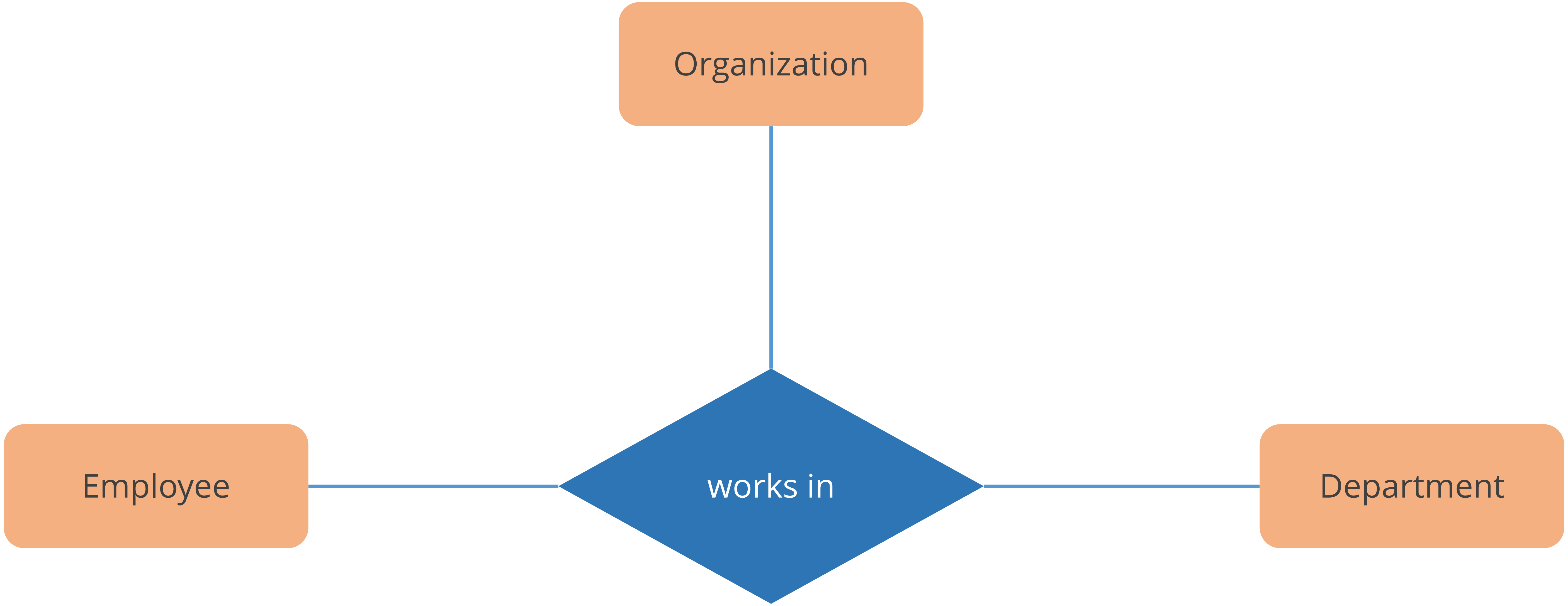
Unary Relationship

When both partners in a relationship are the same entity, the relationship is said to be unary.



Ternary Relationship

A ternary relationship is one where three entities are involved.



Types of Relationships

Types of Relationships

There are four types of relationships:

One-to-one (1:1)

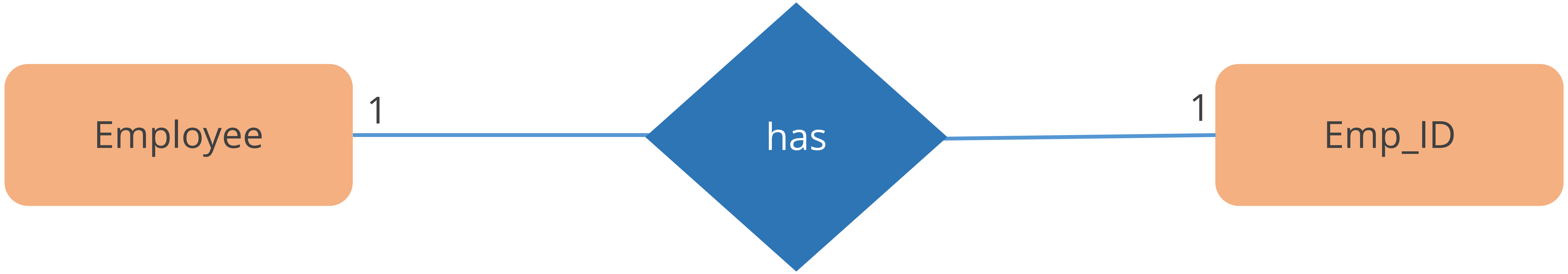
One-to-many (1:M)

Many-to-one (M:1)

Many-to-many (M:M)

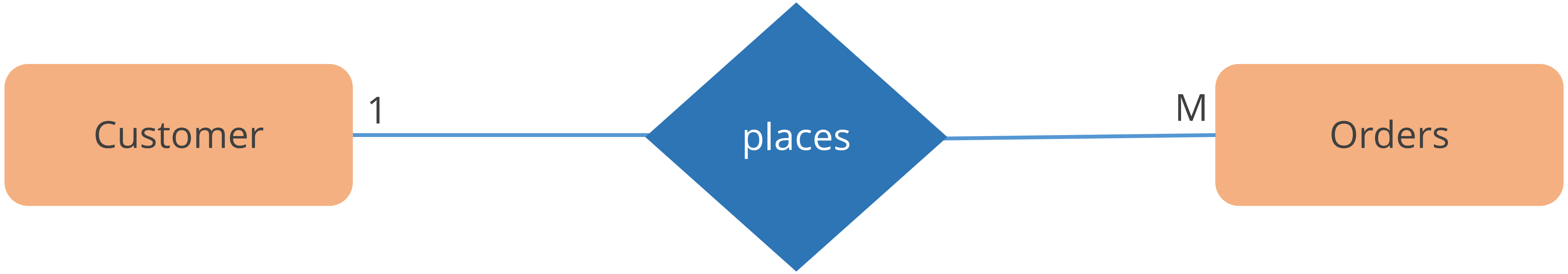
One-to-One (1:1)

A one-to-one relationship exists when a single instance of one entity is connected with a single instance of another entity.



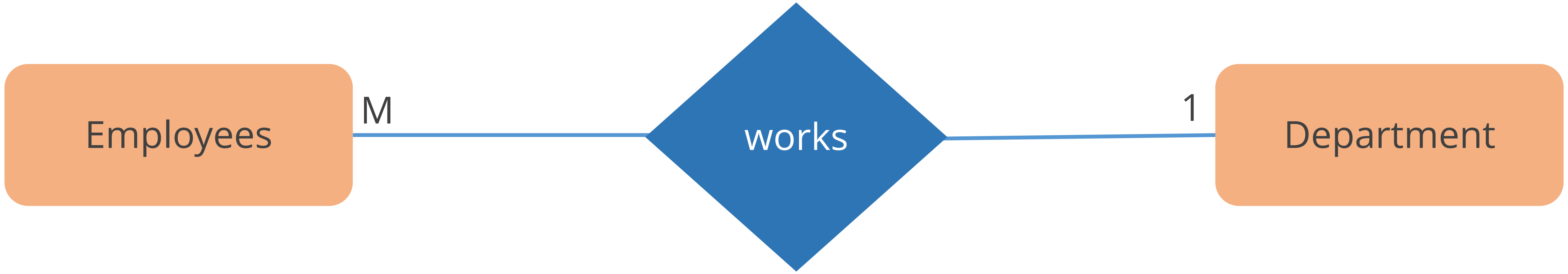
One-to-Many (1:M)

When a single instance of one entity is connected to several instances of another entity, this is referred to as a one-to-many relationship.



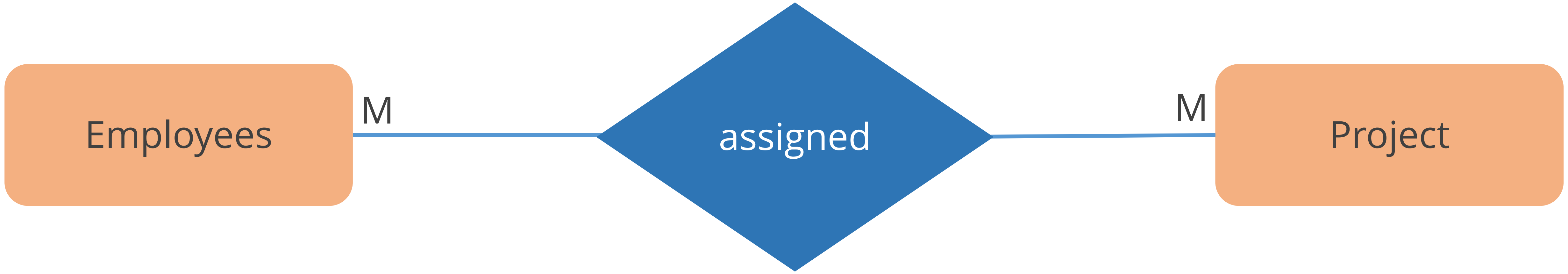
Many-to-One (M:N)

A many-to-one relationship exists when several instances of one entity are connected to a single instance of another entity.



Many-to-Many (M:M)

A many-to-many relationship occurs when more than one instance of an entity is connected to several instances of another entity.



Assisted Practice: ER Diagram

Duration: 20 mins

Problem statement:

E-commerce is one of the emerging fields and is widely accepted across the globe. Design an ER model for a start-up named "Sell in the Sale" based on the below business rules so that it is easy for the management to understand the design of their company's database. A salesperson can manage many customers. A customer, however, is managed by only one salesperson. One customer can place multiple orders, but each order will always belong to single customer. An order lists many products, and a single product can be present in multiple orders. An order will have the columns orderID, orderDate, noOfProducts, and productName. Make sure to follow proper notations to represent the entities, attributes, and relationships with their types (1:1, 1:M, M:1, and M:M).

Objective:

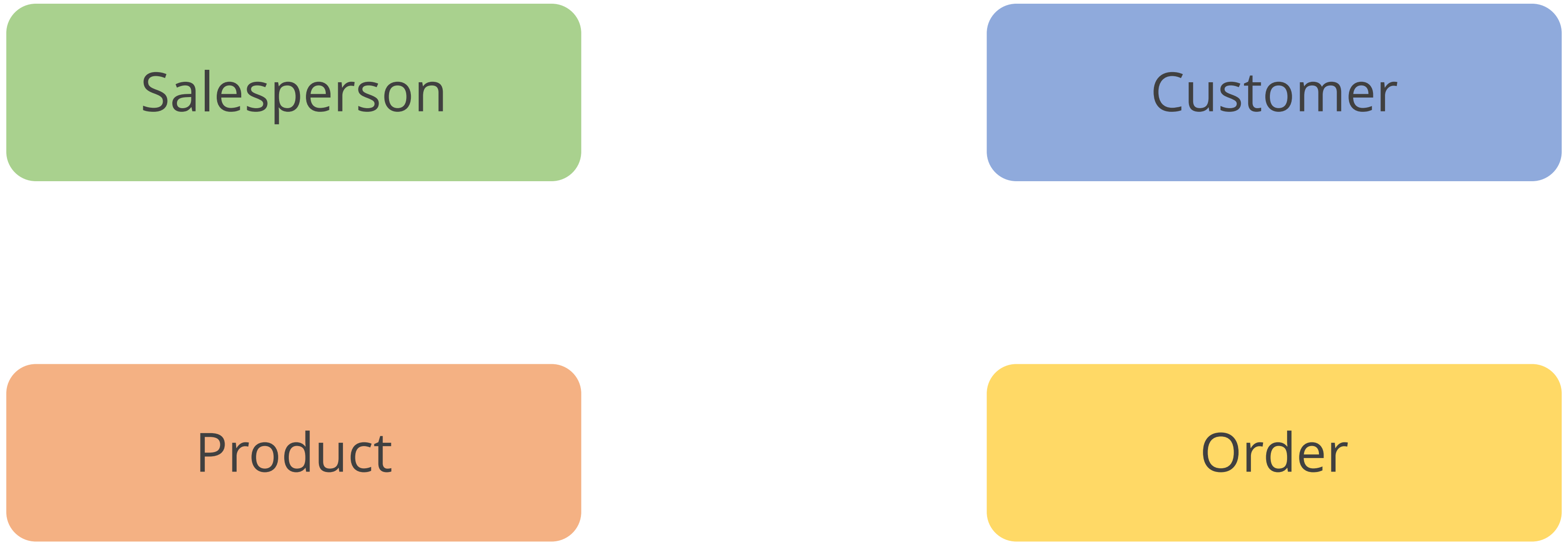
Identify the entities, attributes, and relationships in order to solve this problem

Assisted Practice: ER Diagram

Steps to be performed:

Step 01: Identify the entities

An entity is a real-world object that represents the data. It is represented as a rectangle. Here, Salesperson, Customer, Order, and Product represent the entities.



Salesperson

Customer

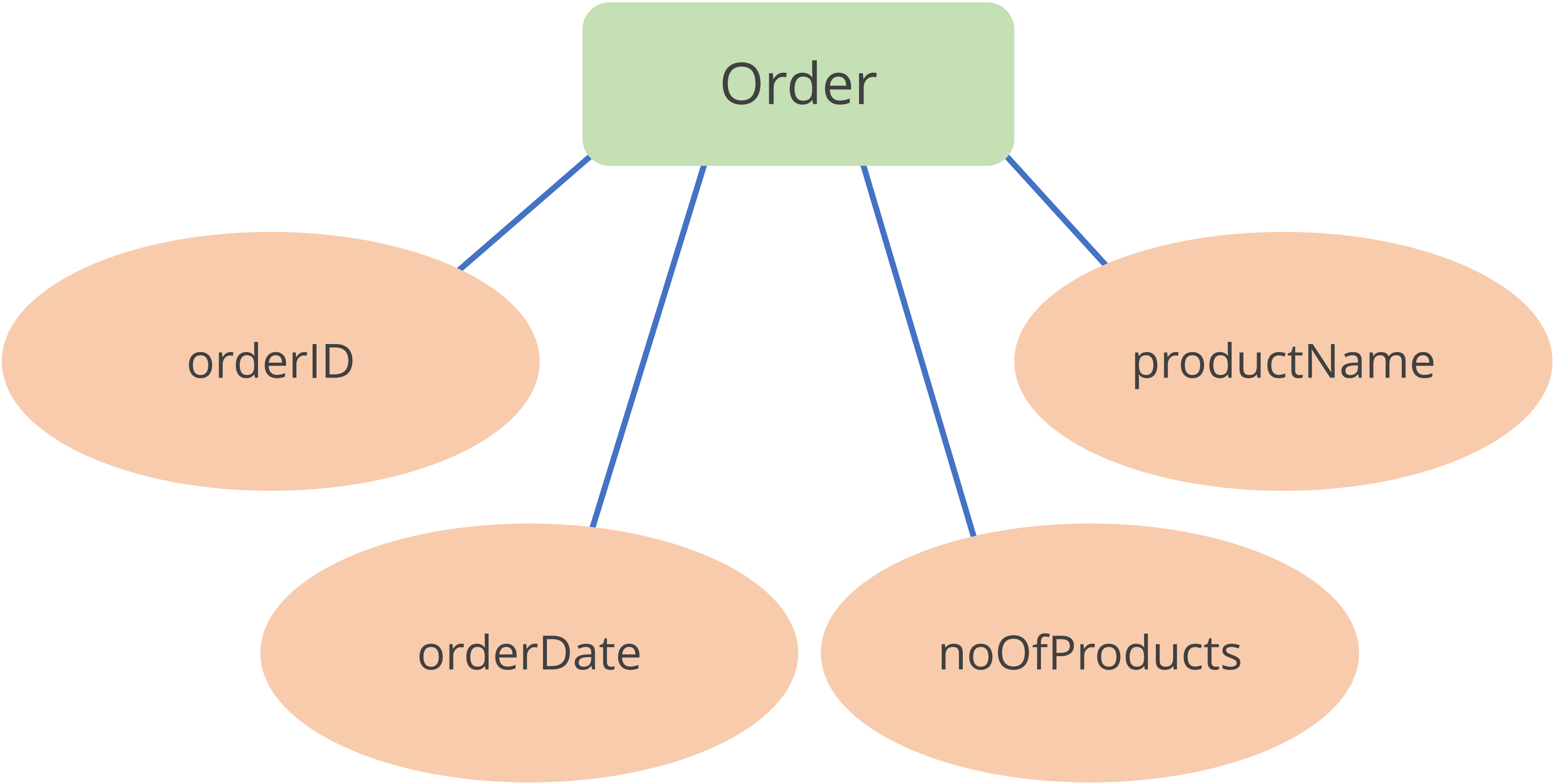
Product

Order

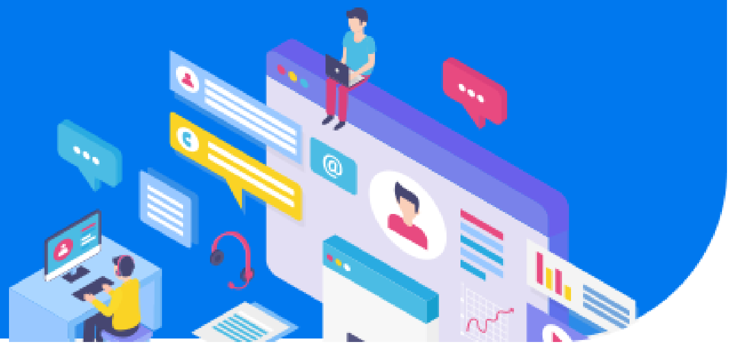
Assisted Practice: ER Diagram

Step 02: Identify the attributes

An attribute is a property that describes an entity. It is represented as an ellipse. orderID, orderDate, noOfProducts, and productName describe the entity Order. Hence, they are attributes.



Assisted Practice: ER Diagram



Step 03: Identify the relationships

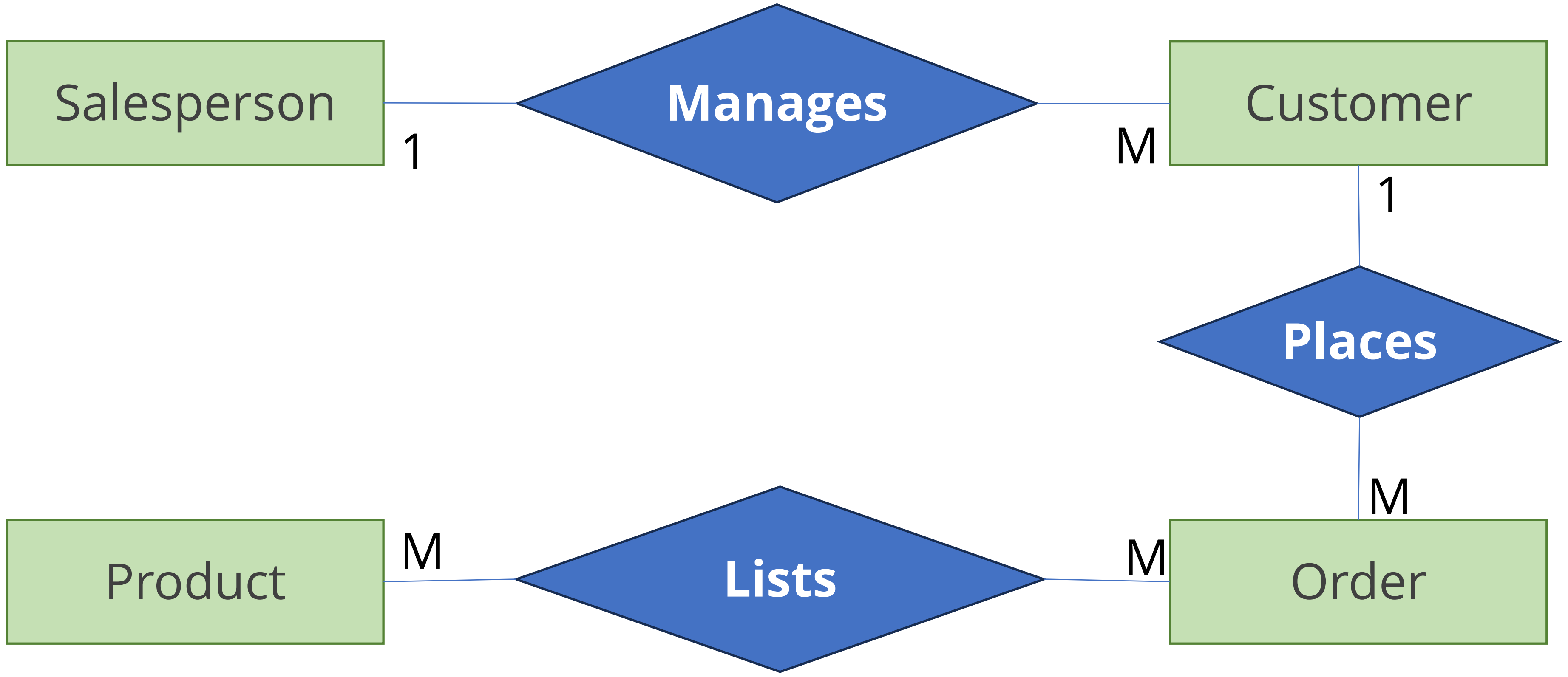
Relationships are used to document the interaction between the entities. It is represented as a diamond.

A salesperson can manage many customers. A customer is managed by only one salesperson. Hence, the relationship here is Manages and one-to-many (1:M).

Multiple orders can be placed by a customer, but an order will always belong to a single customer. Hence the relationship here is Places and one-to-many (1:M).

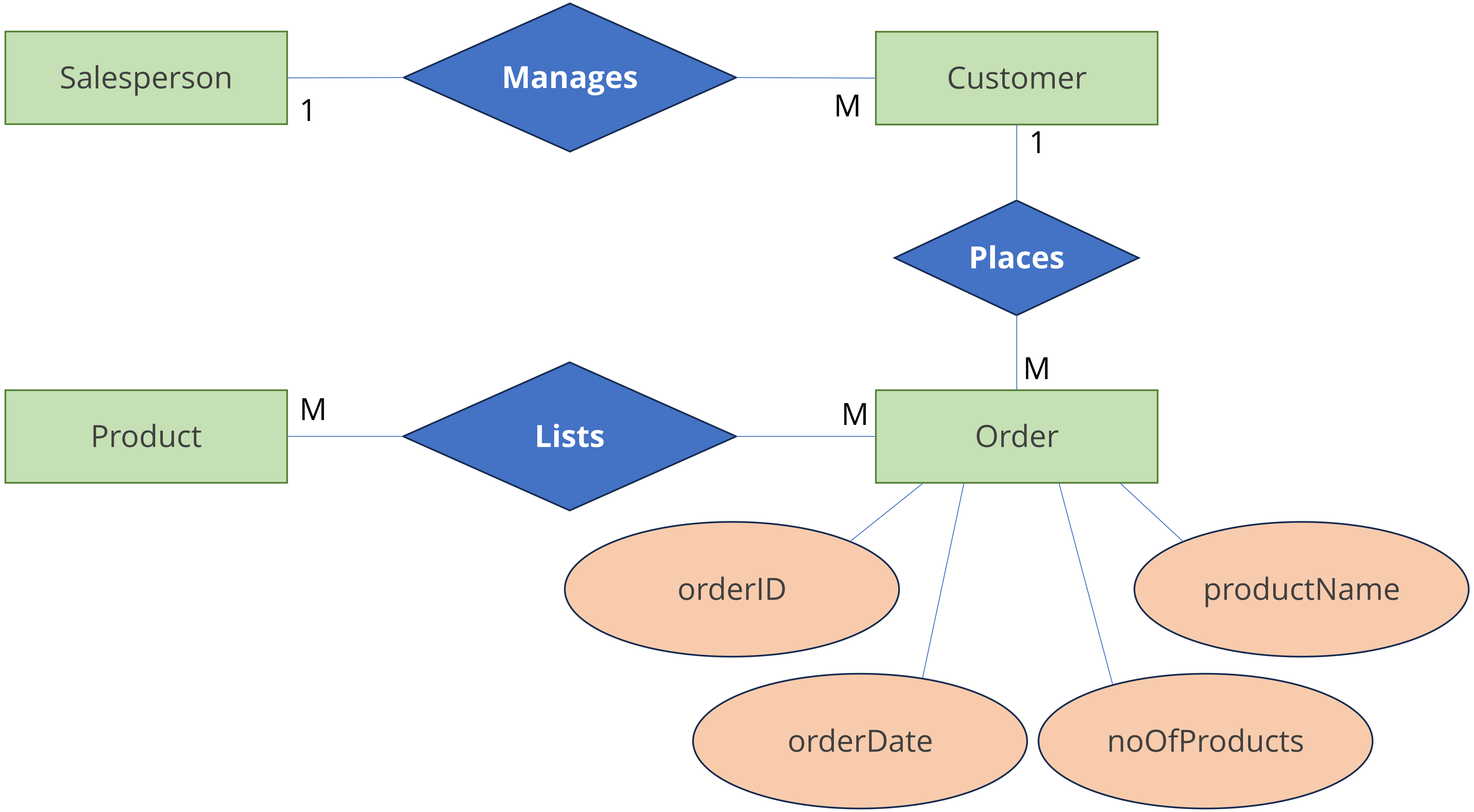
An order lists many products, and a single product can be present in multiple orders. Hence the relationship here is Lists and many-to-many (M:M).

Assisted Practice: ER Diagram



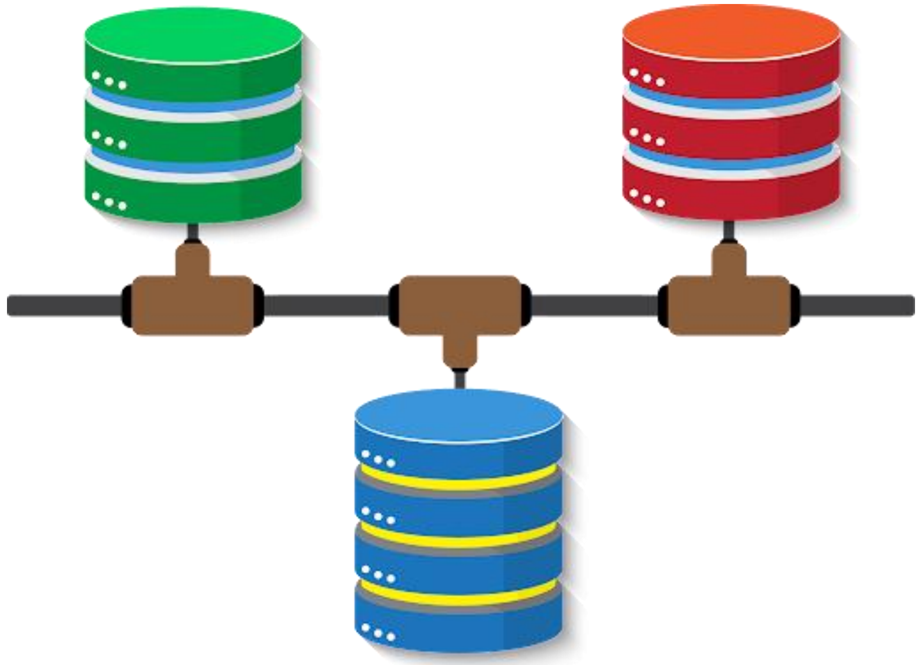
Assisted Practice: ER Diagram

Step 04: Using the information gathered in steps 1, 2, and 3, create an ER diagram



Mapping Cardinalities

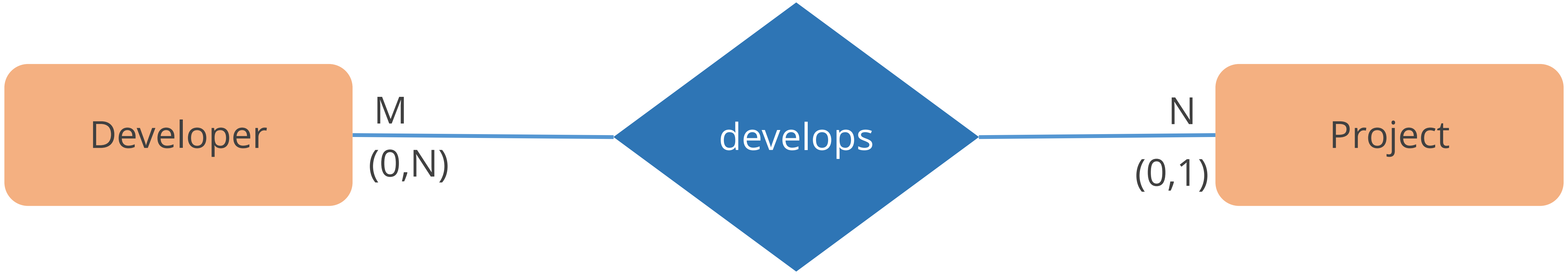
Cardinalities



- A relationship's cardinality is the number of occurrences of an entity that can be associated with another entity.
- Each relationship has a minimum and a maximum cardinality.

Cardinalities

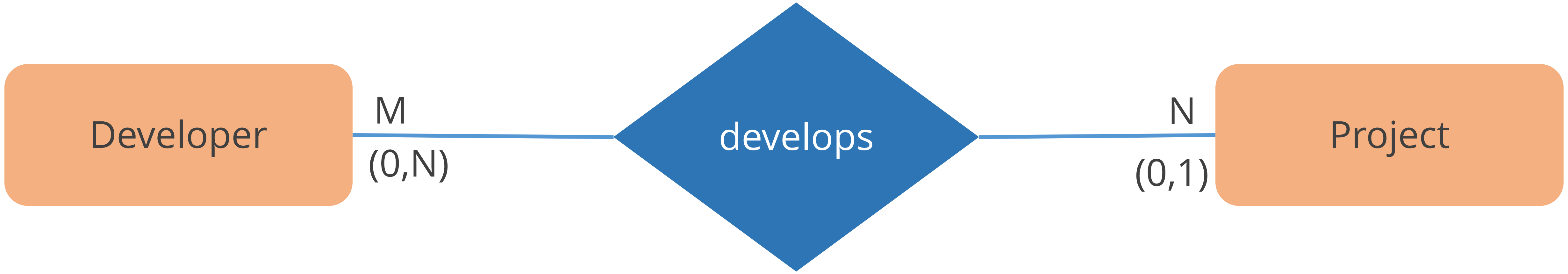
Each developer in an organization can work on an indefinite number of projects as long as his or her weekly hours do not exceed 40.



Developers may develop 0 projects if they are involved in nontechnical projects. Therefore, the cardinality limits for the developer are (O,N).

Cardinalities

According to the organization's regulations, each project is developed by a single developer, but it is possible to have projects that have not yet been assigned to a developer.

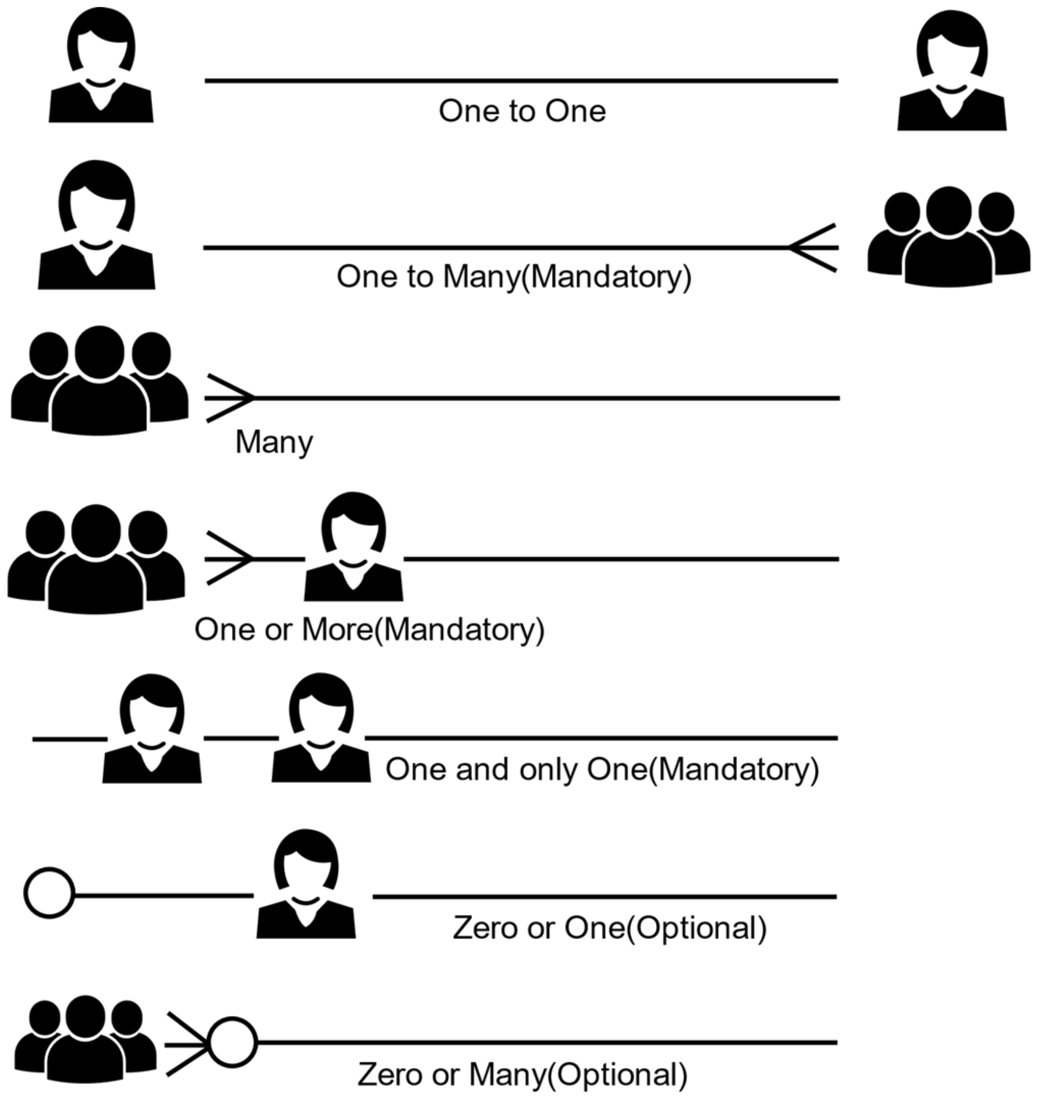


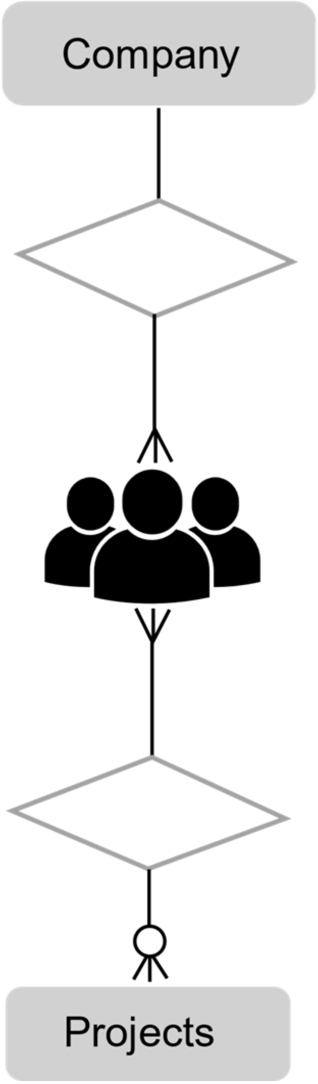
Therefore, the cardinality limits for the project are (0,1).

Cardinalities Notations of ER Diagram

Notations

Cardinality is represented by the styling of a line and its endpoint, according to the chosen notation style:





Database Normalization

Database Normalization

- The process of organizing and managing data in a database to eliminate redundancy and unnecessary anomalies is known as normalization.
- It helps in the division of large database tables into smaller ones and establishes relationships between them.

Types of Anomalies

Types of Anomalies

Data redundancy

Insert anomaly

Update anomaly

Delete anomaly

Data Redundancy

Data redundancy occurs when two or more rows or columns have the same or repeated value, causing the memory to be used inefficiently.

EmpRegistration	EmpID	EmpName	Address	Department
2305	6204	John	Los Angeles	Finance
2305	6247	John	Los Angeles	Finance
2324	6247	Bolt	New York	Admin
2330	6204	Ritchie	Egypt	IT
2330	6208	Ritchie	Egypt	HR

The records of the two employees, **John** and **Ritchie** are repetitive in the above table, which results in data redundancy.

Insert Anomaly

Insert anomaly occurs when some attributes or data items are to be inserted into the database without the existence of other attributes.

EmpRegistration	EmpID	EmpName	Address	Department
2305	6204	John	Los Angeles	Finance
2305	6247	John	Los Angeles	Finance
2324	6247	Bolt	New York	Admin
2330	6204	Ritchie	Egypt	IT
----	----	----	----	----

If we want to enter a new EmpID into the employee table, we must wait till the employee joins the organization. Hence, it is called insertion anomalies.

Update Anomaly

Update anomaly occurs when duplicate data is updated only in one location and not in other instances. As a result, the data becomes inconsistent.

EmpRegistration	EmpID	EmpName	Address	Department
2305	6204	John	Los Angeles	Finance
2305	6247	John	Los Angeles	Finance
2324	6247	Bolt	New York	Admin
2330	6204	Ritchie	Egypt	IT
2330	6208	Ritchie	Egypt	HR

In the above table, there is an employee named **John**. If we change the department in the employee database, we must also change it in the department database; otherwise, the data will be inconsistent.

Delete Anomaly

Delete anomaly occurs when certain entries are lost or deleted from a database table as a result of the deletion of other records.

EmpRegistration	EmpID	EmpName	Address	Department
2305	6204	John	Los Angeles	Finance
2305	6247	John	Los Angeles	Finance
2324	6247	Bolt	New York	Admin
2330	6204	Ritchie	Egypt	IT
2330	6208	Ritchie	Egypt	HR

If we delete Bolt from the above table, we also remove his address and other data from the table. As a result, we may argue that removing certain attributes might result in the removal of other attributes from the database table.

Types of Normalization

Types of Normalization

First normal form (1NF)

Fifth normal form (5NF)

Second normal form (2NF)

Fourth normal form (4NF)

Third normal form (3NF)

Boyce and Codd normal form (BCNF)

First Normal Form (1NF)

- The 1NF states that all the attributes in a relation must have atomic domains.
- The 1NF specifies that a table attribute cannot have multiple values.
- In 1NF, multivalued attribute, composite attribute, and their combination are not allowed.

First Normal Form (1NF)

The table on the left consists of employees who belongs to different departments. Example- John. Normalization is achieved by splitting this record into two different rows.

EmpID	EmpName	Department
6204, 6240	John	Sales, Marketing
6247	Kit	Finance
6247	Bolt	Admin
6204	Ritchie	IT
6208	Ritchie	HR

Table without first normal form

EmpID	EmpName	Department
6204	John	Sales
6240	John	Marketing
6247	Kit	Finance
6247	Bolt	Admin
6204	Ritchie	IT
6208	Ritchie	HR

Table with first normal form

Second Normal Form (2NF)

- In the second normal form, the entity should already be in 1NF.
- All attributes inside it should be based entirely on the entity's unique identifier.
- To remove the dependency, we can divide the table, remove the attribute that causes the dependency, and add it to the other table where it fits.

Second Normal Form (2NF)

The first table is a course table with the details of course name, teacher age, and teacher ID.

Teacher_ID	Course	Teacher_Age
2115	Web Development	30
2115	Python	30
4997	Machine Learning	35
8989	Artificial Engineering	38
8989	SQL	38

Course table without the second normal form

Teacher_ID	Teacher_Age
2115	30
4997	35
8989	38

Teacher detail table with the second normal form

Teacher_ID	Course
2115	Web Development
2115	Python
4997	Machine Learning
8989	Artificial Engineering
8989	SQL

Course table with the second normal form

In the given table, brand is dependent on product ID, a proper subset of the candidate key that violates the rules of 2NF.

Second Normal Form (2NF)

To convert the given table into 2NF, let’s decompose it into two tables:

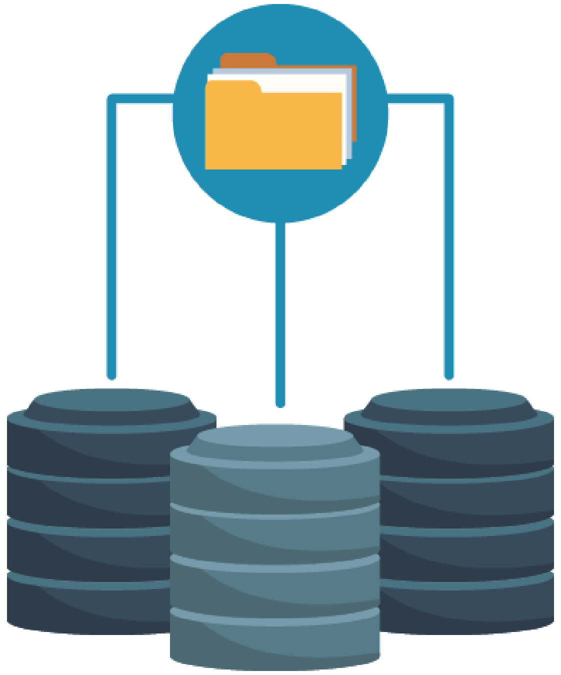
Teacher_ID	Teacher_Age
2115	30
4997	35
8989	38

Teacher details table with the second normal form

Teacher_ID	Course
2115	Web Development
2115	Python
4997	Machine Learning
8989	Artificial Engineering
8989	SQL

Course table with the second normal form

Third Normal Form (3NF)



- In the third normal form, an entity should be considered already in 2NF.
- No column entry should be dependent on any other entry (value) than the table's key.
- 3NF is used to reduce redundancy in the data and to ensure data consistency.

Third Normal Form (3NF)

The given tables have employee details along with their ratings:

EmpID	EmpName	Department
6204	John	Sales
6247	Kit	Finance
6247	Bolt	Admin
6204	Ritchie	IT
6208	Ritchie	HR

Employee table

EmpID	Ratings (out of 5)	Salary
6204	1.5	25000
6247	2	27000
6247	3	27000
6204	2.5	28000
6208	2.7	30000

Ratings table

To determine the hike percentage, HR needs to create a new column in the ratings table named as hike.

Third Normal Form (3NF)

After adding the hike column, we have achieved 3NF.

EmpID	Ratings (out of 5)	Salary	Hike
6204	1.5	25000	0%
6247	2	27000	10%
6247	3	27000	80%
6204	2.5	28000	20%
6208	2.7	30000	25%

Ratings table

Boyce and Codd Normal Form (BCNF)

- In BCNF, a table or database must be in the third normal form.
- All the tables in the database should have just one primary key.
- For every functional dependency A->B, A should be the super key of the table.

Boyce and Codd Normal Form (BCNF)

In this example you have an employee table with the following details:

E_ID	E_location	E_Dept	D_Type	D_emp_count
6204	U.S.A	Production	P101	100
6247	Canada	Sales	S102	400
6247	Australia	Design	P102	80
6204	U.K	Operations	S103	150

Employee table without BCNF

The table is not in BCNF as neither E_ID nor E_Dept alone are keys.

Boyce and Codd Normal Form (BCNF)

To make the table comply with BCNF, divide it into three different tables:

E_ID	E_location
6204	U.S.A
6247	Canada
6247	Australia
6204	U.K

Emp_Location table

E_Dept	D_Type	D_emp_count
Production	P101	100
Sales	S102	400
Design	P102	80
Operations	S103	150

Emp_Dept table

E_ID	E_Dept
6204	Production
6247	Sales
6247	Design
6204	Operations

Emp_Dept_mapping table

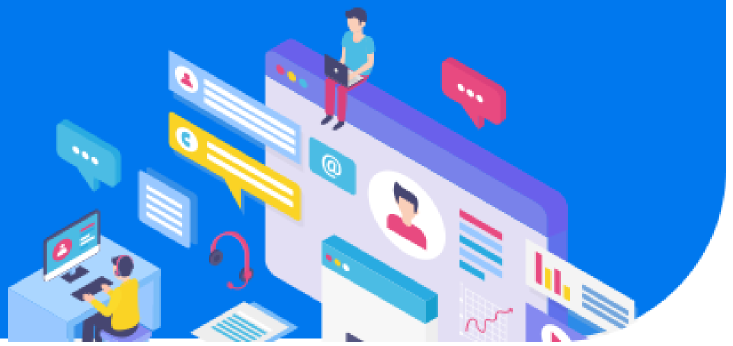
This is how you achieve BCNF as the functional dependencies in the left side part is a key.

Assisted Practice: Normal Forms

Duration: 20 mins

Problem statement: You have just joined an organization as a database administrator. You have been trained on database basics, which includes normalization, and to test your knowledge, you have been asked to solve two questions: A and B.

Assisted Practice: Normal Forms



Question A:

Duration: 20 mins

Data - This data qualifies as 1NF. Is this statement true or false? If false, transform it to 1NF.

SongID	ArtistName	SongCategory
1	Bruno Mars	Funk
2	Alan Walker	Romantic
3	Eminem	Rap, Hip Hop
4	Metallica	Metal
5	Charlie Puth	Romantic

Assisted Practice: Normal Forms

Steps to be performed:

The answer to question A is false.

Note:

The rule of 1NF states that all attributes in a relation must have atomic domains. A table attribute cannot have multiple values. Multivalued attributes, composite attributes, and their combination are not allowed.

The table structure violates the rule of atomicity, i.e., a single cell holds multiple values in the SongCategory column for SongID=3.

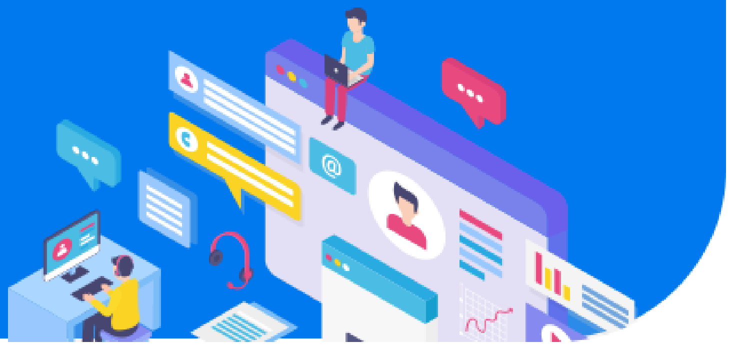
Assisted Practice: Normal Forms

Step 01: Split the single record into two records.

Output:

SongID	ArtistName	SongCategory
1	Bruno Mars	Funk
2	Alan Walker	Romantic
3	Eminem	Rap
3	Eminem	Hip Hop
4	Metallica	Metal
5	Charlie Puth	Romantic

Assisted Practice: Normal Forms



Duration: 20 mins

Question B:

Data - This data qualifies to be in 2NF. Is this statement true or false? If false, transform it to 2NF. Assume (SongID, ArtistName) is the primary key for the table.

SongID	SongName	ArtistName
1	ABC	Bruno Mars
2	PQR	Alan Walker
3	MNO	Eminem
4	XYZ	Metallica
5	DEF	Charlie Puth

ASSISTED PRACTICE

Assisted Practice: Normal Forms

Steps to be performed:

The answer to the question is false.

Note:

The 2NF rule states that the entity should already be in 1NF. All attributes inside it should be based entirely on the entity's unique identifier. To remove the dependency, divide the table, remove the attribute that causes the dependency, and add it to the artist table, where it fits.

The table structure and violates the rule of partial dependency. The SongName column can be determined by SongID which makes this relationship partially dependent.

Partial dependency is when a nonprime attribute (SongName) is functionally dependent on a part of a key (SongID and ArtistName).

Assisted Practice: Normal Forms

Step 01: Remove this dependency to transform to 2NF.

Output:

SongID	SongName
1	ABC
2	PQR
3	MNO
4	XYZ
5	DEF

ArtistID	SongID
101	2
102	5
103	1
104	3
105	4

ArtistID	ArtistName
101	Bruno Mars
102	Alan Walker
103	Eminem
104	Metallica
105	Charlie Puth

Knowledge Check

Knowledge Check

1

A __________ is one that is reliant on another entity.

A. Weak entity

B. Strong entity

C. Attribute

D. Relationship

Knowledge Check

1

A __________ is one that is reliant on another entity.

A. Weak entity

B. Strong entity

C. Attribute

D. Relationship

The correct answer is **A**

A weak entity is one that is reliant on another entity

Knowledge Check

2

The properties of entities are known as ________.

A. Strong entity

B. Attributes

C. Weak entity

D. Relationship

Knowledge Check

2

The properties of entities are known as ________.

A. Strong entity

B. Attributes

C. Weak entity

D. Relationship

The correct answer is **B**

The properties of entities are known as attributes.

Knowledge Check

3

A relationship set is a group of similar types of __________.

A. Entity

B. Attributes

C. Department

D. Relationships

Knowledge Check

3

A relationship set is a group of similar types of __________.

A. Entity

B. Attributes

C. Department

D. Relationships

The correct answer is **D**

A relationship set is a group of similar types of relationships.

Lesson-End Project: Cricket Team Model

Problem statement:
You are working as a junior DBA for a gaming company that is looking for a DB design to capture the scenario of the model cricket teams, the games they play, and the players in each team.

Objective:
Design a detailed ER diagram to capture the scenarios for the player, team, match and umpire.

Tasks to be performed:

1. Design an ER diagram that depicts teams, where each team's ID (unique identifier), name, main stadium, and city are listed

2. Design an ER diagram that depicts players (each player can only play for one team), where each has an ID (unique identifier), name, date of birth, shirt number, and year of start

Lesson-End Project: Cricket Team Model

Tasks to be performed:

3. Design an ER diagram that depicts matches, where each will have a date, final result, city (where a match was played)

4. Design an ER diagram that depicts umpires (each match has one umpire), where each has an ID (unique identifier), name, date of birth, and years of experience

5. With the concepts and notations of entities, attributes, and relationships, create an ER diagram that incorporates all aforementioned scenarios into a single model

Note: You can state any assumptions that affect your design

Key Takeaways

- An Entity–Relationship model (ER model) is a data model which describes the design of database with the help of Entity-Relationship diagram.

- The three main components of the ER diagram are entity, attribute, and relationship.

- A relationship set is a group of similar type of relationships.

- The degree of a relationship is determined by the number of entities that participate in it.