3

Exploration of Entity Attribute Relationships Notations

Objectives

After completing this lesson, you should be able to:

- Entity Relationship Diagram
- ERD Entities Attributes and Relationship Notations
- Mapping Constraints
- Keys
- Generalization Specialization and Aggregation
- Steps Involved in Creation of ER Diagram

Course Roadmap

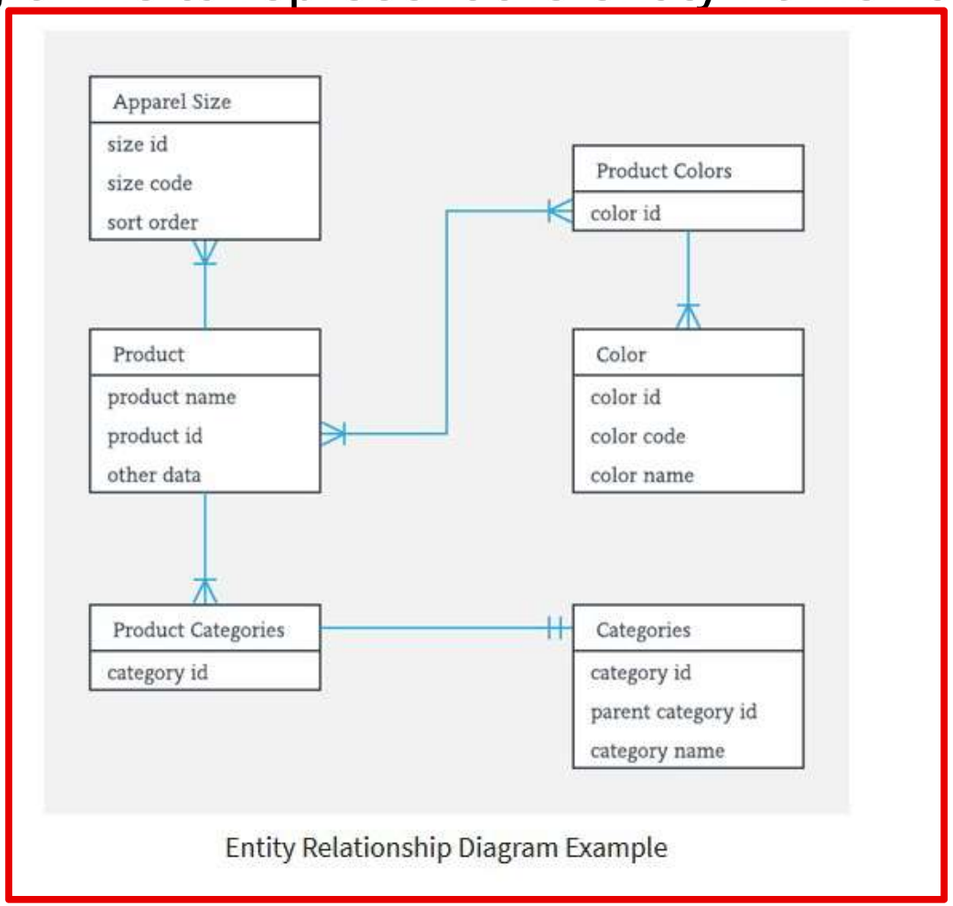
RDBMS Concepts

- Lesson 1: Introduction to Databases
- Lesson 2: Overview of Data Model
- **Lesson 3: Exploration of Entity Attribute Relationships Notations** — **You are here!**
- Lesson 4: Normalization and De-Normalization

ER (Entity Relationship) Diagram in DBMS

- ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an **Entity-Relationship** diagram.
- ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

- ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure.



Entity Relationship Diagram Example

What is ER Model?

- **ER Model** stands for Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyze data requirements to produce a well-designed database. The ER Model represents real-world entities and the relationships between them. Creating an ER Model in DBMS is considered as a best practice before implementing your database.
- ER Modeling helps you to analyze data requirements systematically to produce a well-designed database. So, it is considered a best practice to complete ER modeling before implementing your database.

History of ER models

- ER diagrams are visual tools that are helpful to represent the ER model. Peter Chen proposed ER Diagram in 1971 to create a uniform convention that can be used for relational databases and networks.
- He aimed to use an ER model as a conceptual modeling approach.

Why use ER Diagrams?

1. Helps you to define terms related to entity relationship modeling
2. Provide a preview of how all your tables should connect, what fields are going to be on each table
3. Helps to describe entities, attributes, relationships
4. ER diagrams are translatable into relational tables which allows you to build databases quickly
5. ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
6. The database designer gains a better understanding of the information to be contained in the database with the help of ER diagram
7. ERD Diagram allows you to communicate with the logical structure of the database to users

Facts about ER Diagram Model

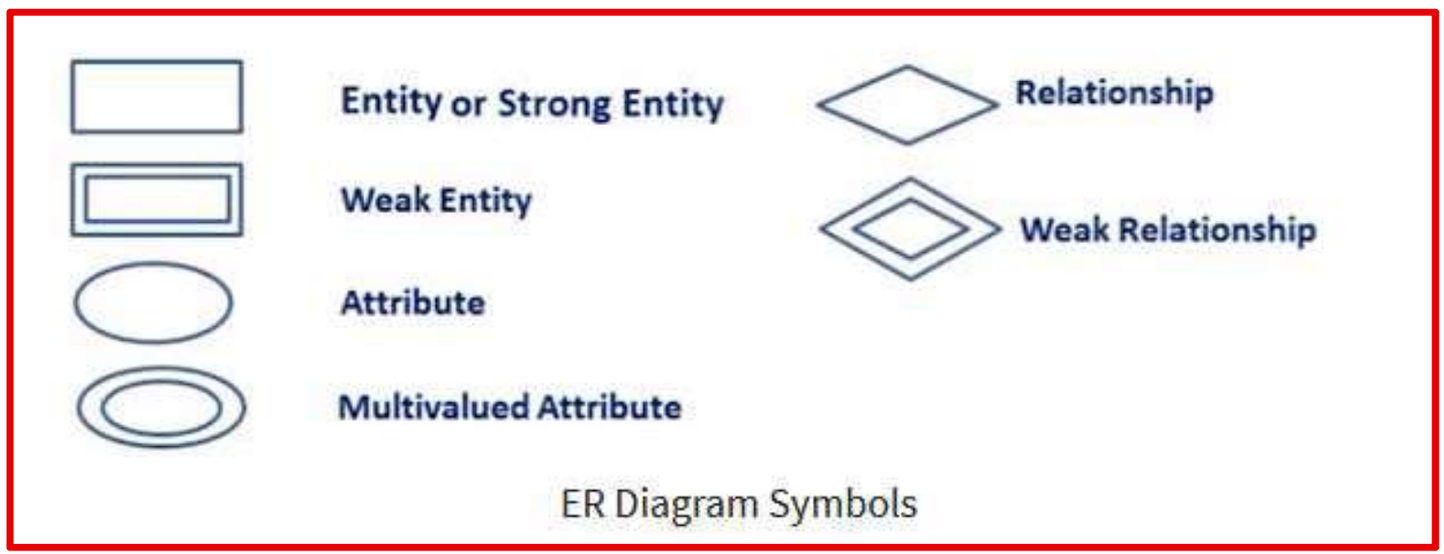
- ER model allows you to draw Database Design
- It is an easy to use graphical tool for modeling data
- Widely used in Database Design
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationships between those entities

ER Diagram Symbols And Notations

- **Entity Relationship Diagram Symbols & Notations** mainly contains three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes.
- There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Main components and its symbols in ER Diagrams:

1. **Rectangles:** This Entity Relationship Diagram symbol represents entity types
2. **Ellipses :** Symbol represent attributes
3. **Diamonds:** This symbol represents relationship types
4. **Lines:** It links attributes to entity types and entity types with other relationship types
5. **Primary key:** attributes are underlined
6. **Double Ellipses:** Represent multi-valued attributes



ER Diagram Symbols

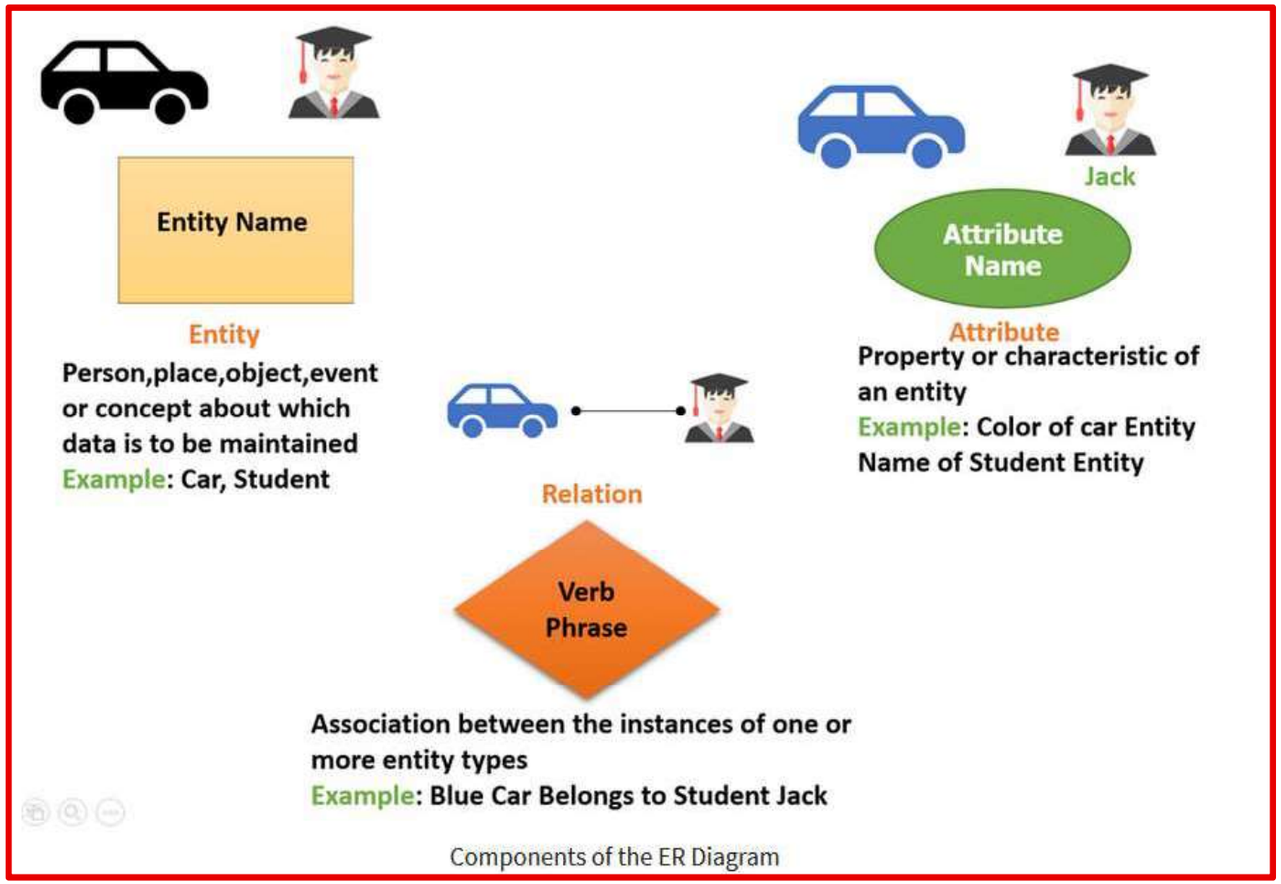
Components of the ER Diagram

This model is based on three basic concepts:

1. Entities
2. Attributes
3. Relationships

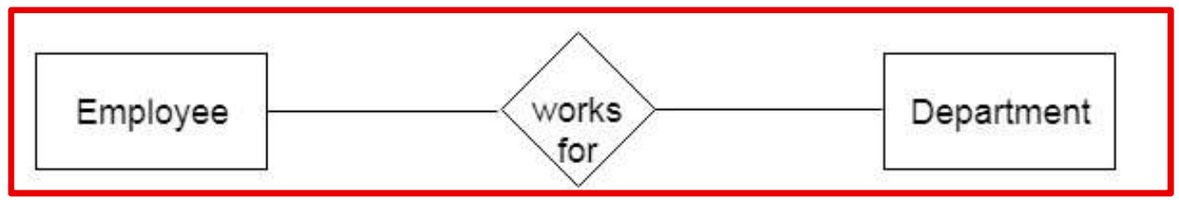
ER Diagram Examples

- For example, in a University database, we might have entities for Students, Courses, and Lecturers. Students entity can have attributes like Rollno, Name, and DeptID. They might have relationships with Courses and Lecturers.



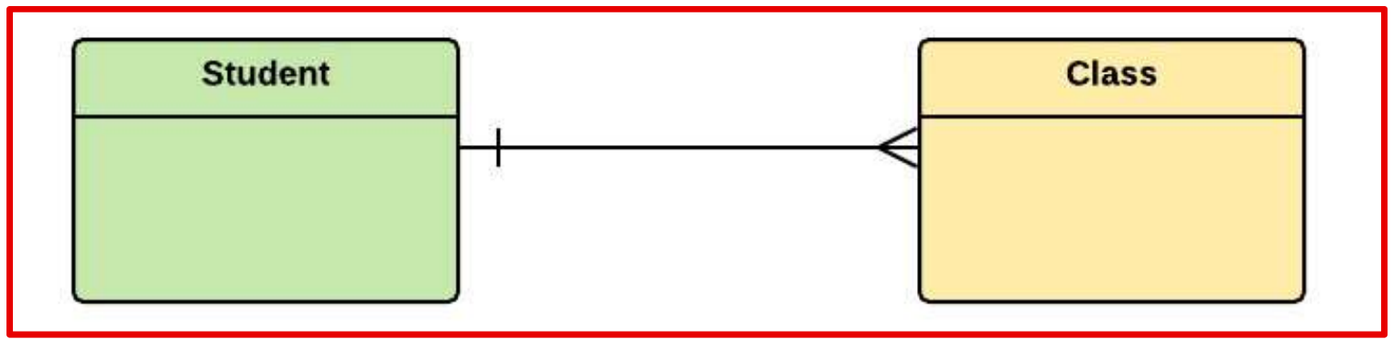
What is an Entity?

- A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.
- An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.
- **Examples of entities:**
  - **Person:** Employee, Student, Patient
  - **Place:** Store, Building
  - **Object:** Machine, product, and Car
  - **Event:** Sale, Registration, Renewal
  - **Concept:** Account, Course



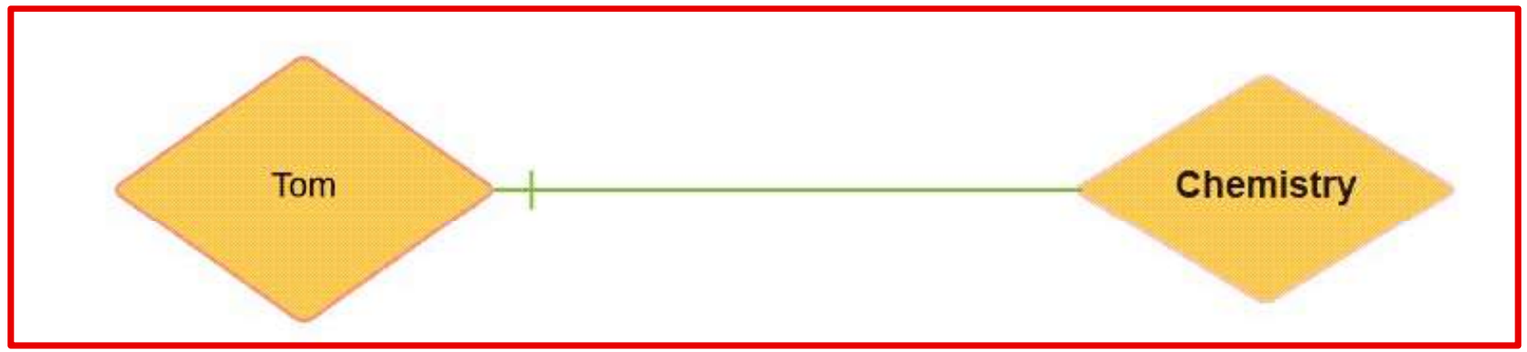
Entity set:

- An entity set is a group of similar kind of entities. It may contain entities with attribute sharing similar values.
- Entities are represented by their properties, which also called attributes. All attributes have their separate values. For example, a student entity may have a name, age, class, as attributes.



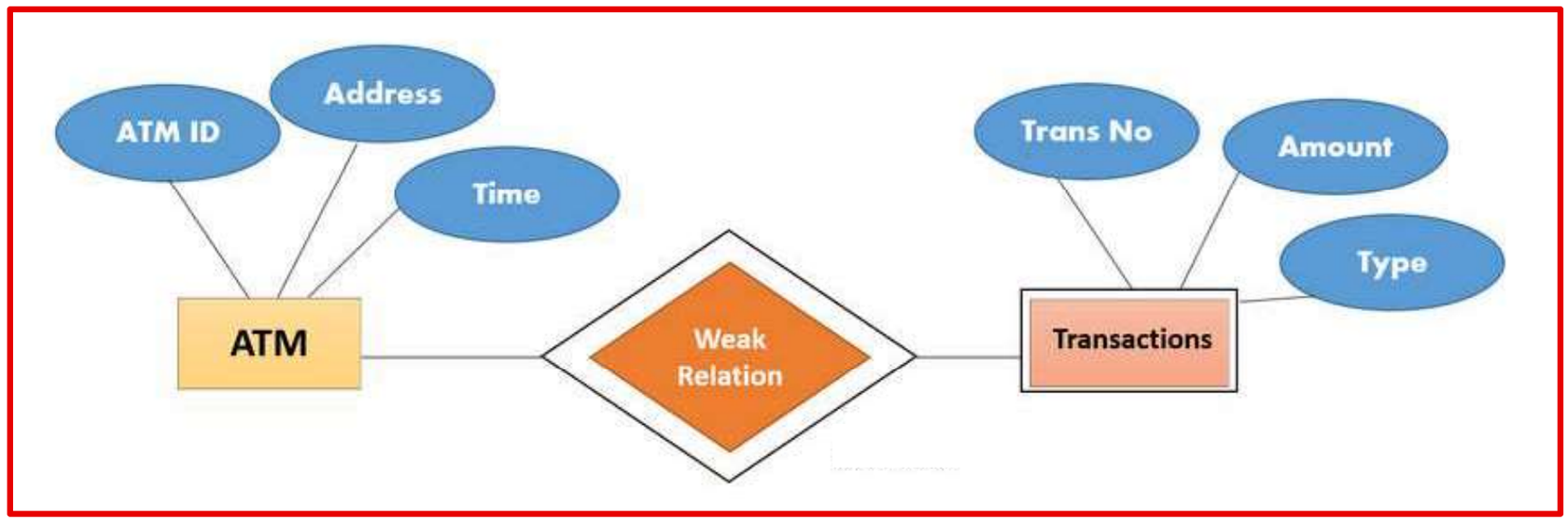
Relationship

- Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.



Weak Entities

- A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity.
- For that, weak entity sets need to have participation.

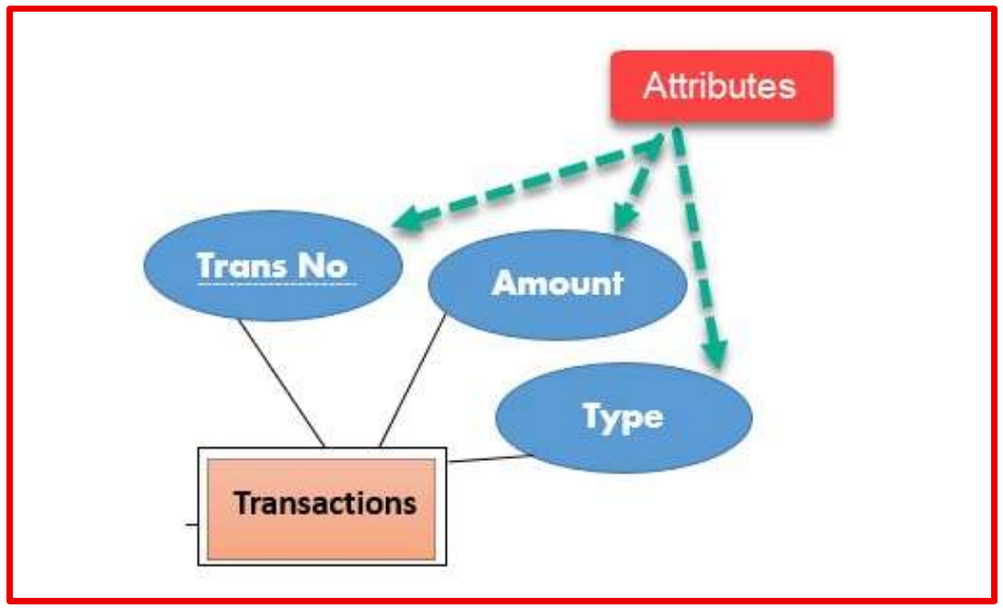


- "Trans No" is a discriminator within a group of transactions in an ATM.

Strong Entity Set	Weak Entity Set
Strong entity set always has a primary key.	It does not have enough attributes to build a primary key.
It is represented by a rectangle symbol.	It is represented by a double rectangle symbol.
It contains a Primary key represented by the underline symbol.	It contains a Partial Key which is represented by a dashed underline symbol.
The member of a strong entity set is called as dominant entity set.	The member of a weak entity set called as a subordinate entity set.
Primary Key is one of its attributes which helps to identify its member.	In a weak entity set, it is a combination of primary key and partial key of the strong entity set.
In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	The relationship between one strong and a weak entity set shown by using the double diamond symbol.
The connecting line of the strong entity set with the relationship is single.	The line connecting the weak entity set for identifying relationship is double.

Attributes

- It is a single-valued property of either an entity-type or a relationship-type.
- For example, a lecture might have attributes: time, date, duration, place, etc.
- An attribute in ER Diagram examples, is represented by an Ellipse

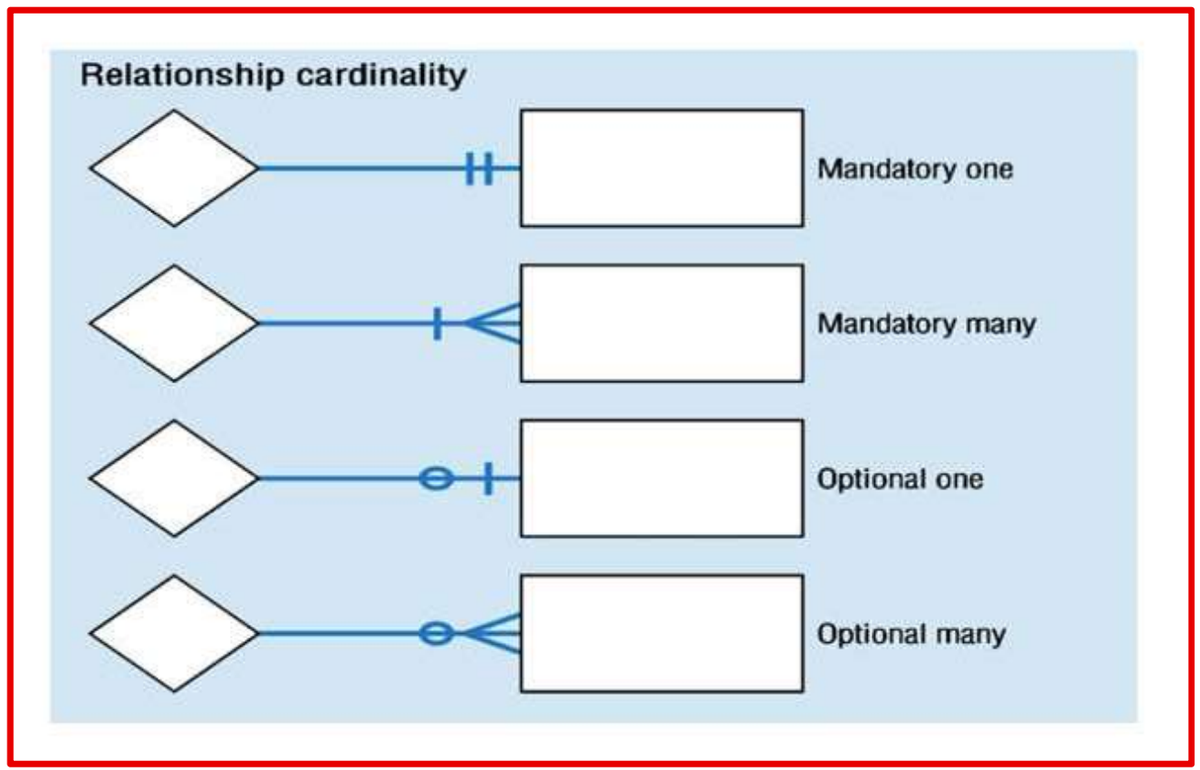


Types of Attributes

Types of Attributes	Description
Simple attribute	Simple attributes can't be divided any further. For example, a student's contact number. It is also called an atomic value.
Composite attribute	It is possible to break down composite attribute. For example, a student's full name may be further divided into first name, second name, and last name.
Derived attribute	This type of attribute does not include in the physical database. However, their values are derived from other attributes present in the database. For example, age should not be stored directly. Instead, it should be derived from the DOB of that employee.
Multivalued attribute	Multivalued attributes can have more than one values. For example, a student can have more than one mobile number, email address, etc.

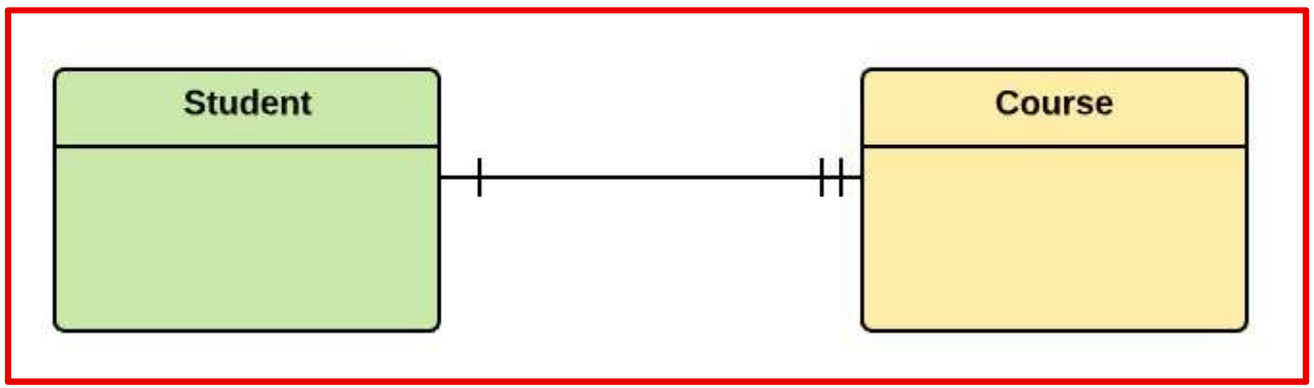
Cardinality

- Defines the numerical attributes of the relationship between two entities or entity sets.
- Different types of cardinal relationships are:
  1. One-to-One Relationships
  2. One-to-Many Relationships
  3. Many to One Relationships
  4. Many-to-Many Relationships



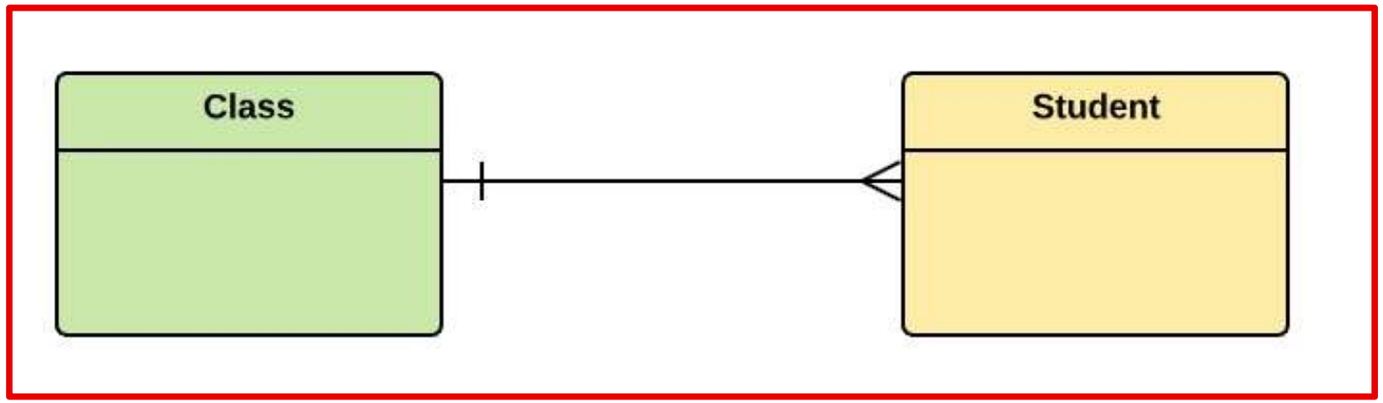
1.One-to-one:

- One entity from entity set X can be associated with at most one entity of entity set Y and vice versa.
  - Example: One student can register for numerous courses. However, all those courses have a single line back to that one student.



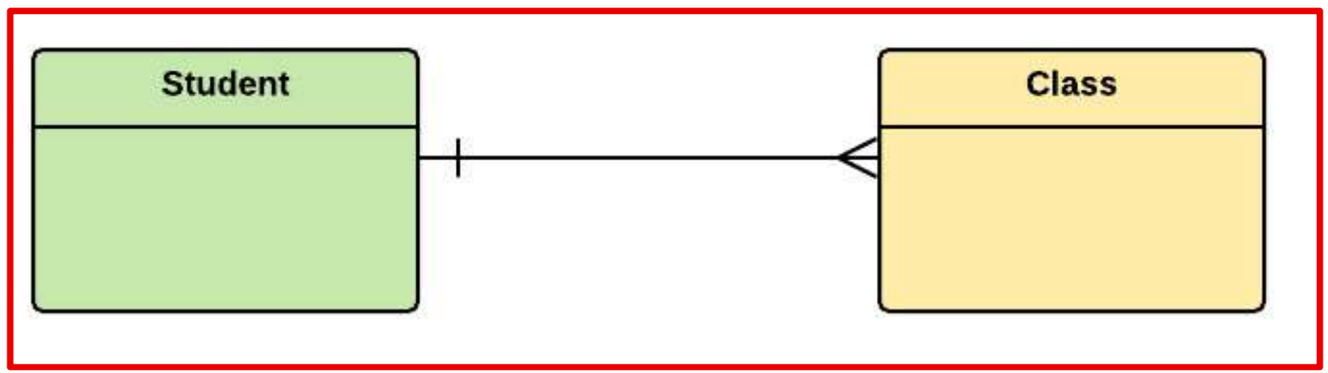
2.One-to-many:

- One entity from entity set X can be associated with multiple entities of entity set Y, but an entity from entity set Y can be associated with at least one entity.
  - For example, one class is consisting of multiple students.



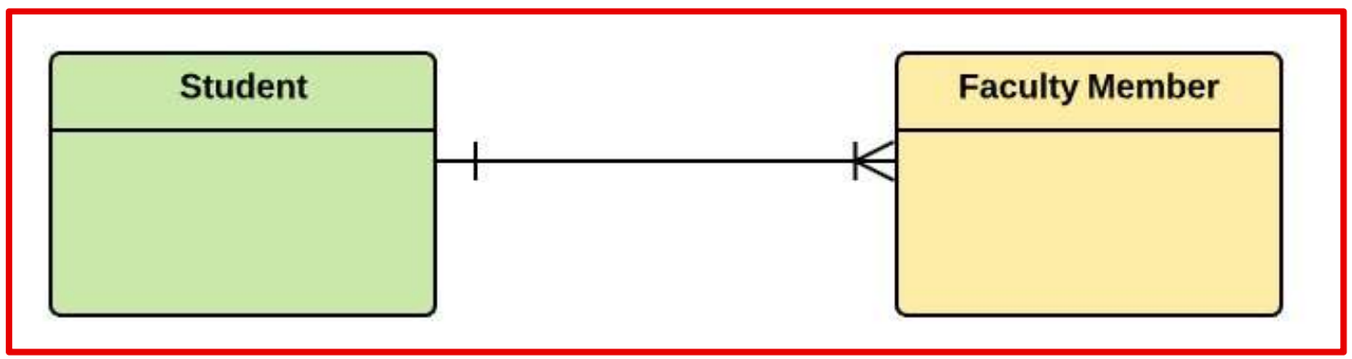
3. Many to One

- More than one entity from entity set X can be associated with at most one entity of entity set Y. However, an entity from entity set Y may or may not be associated with more than one entity from entity set X.
  - For example, many students belong to the same class.



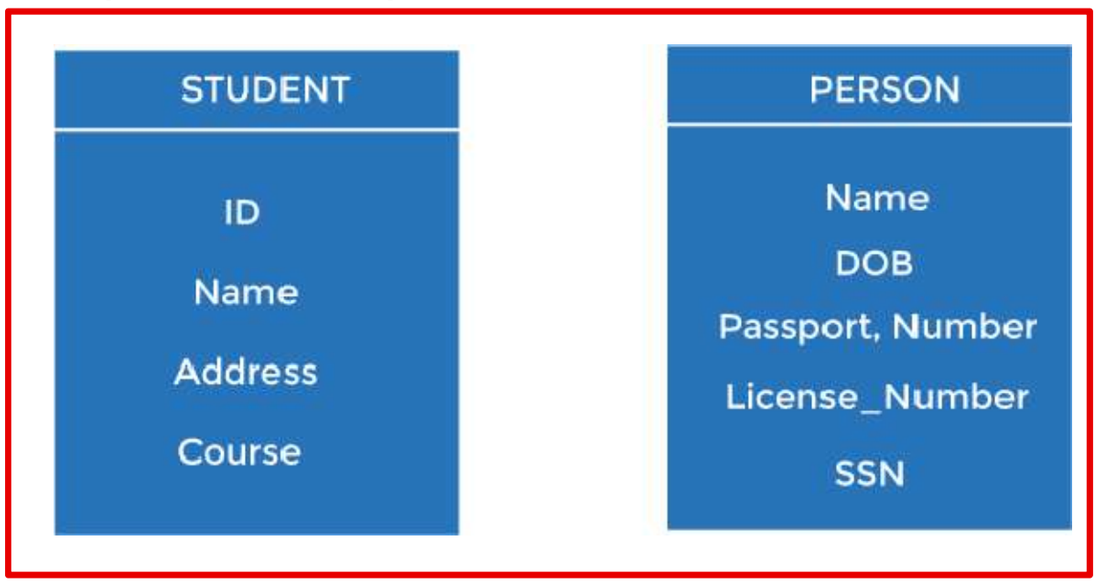
4. Many to Many:

- One entity from X can be associated with more than one entity from Y and vice versa.
  - For example, Students as a group are associated with multiple faculty members, and faculty members can be associated with multiple students.

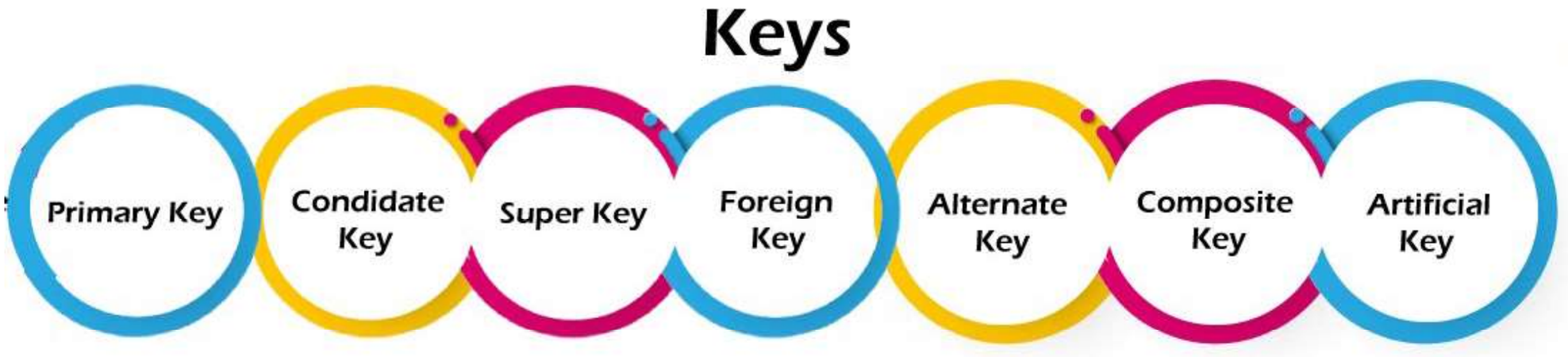


Keys

- Keys play an important role in the relational database.
- It is used to uniquely identify any record or row of data from the table. It is also used to establish and identify relationships between tables.
  - **For example,** ID is used as a key in the Student table because it is unique for each student. In the PERSON table, passport_number, license_number, SSN are keys since they are unique for each person.

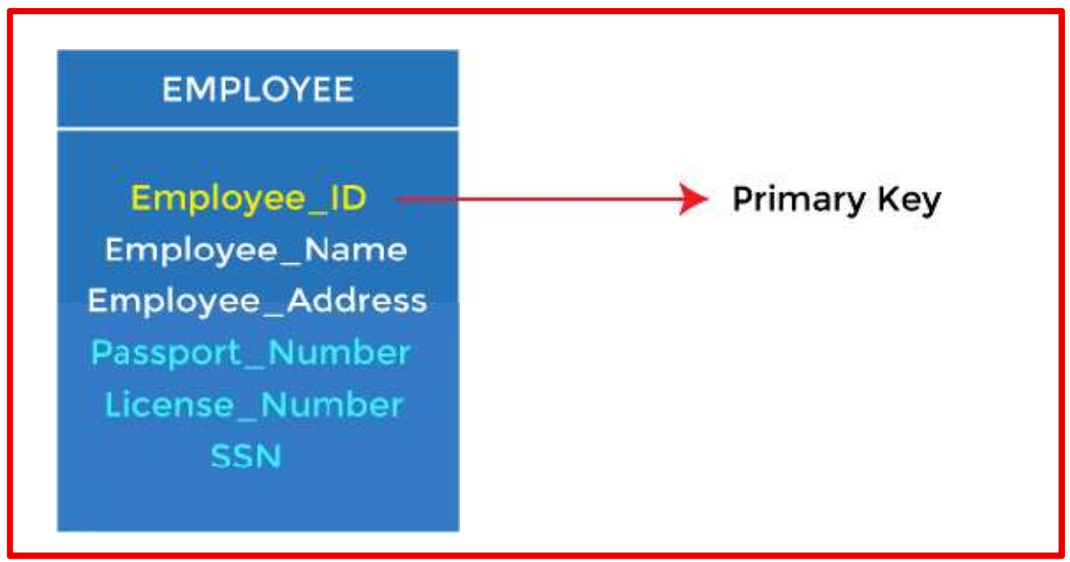


Types of keys:



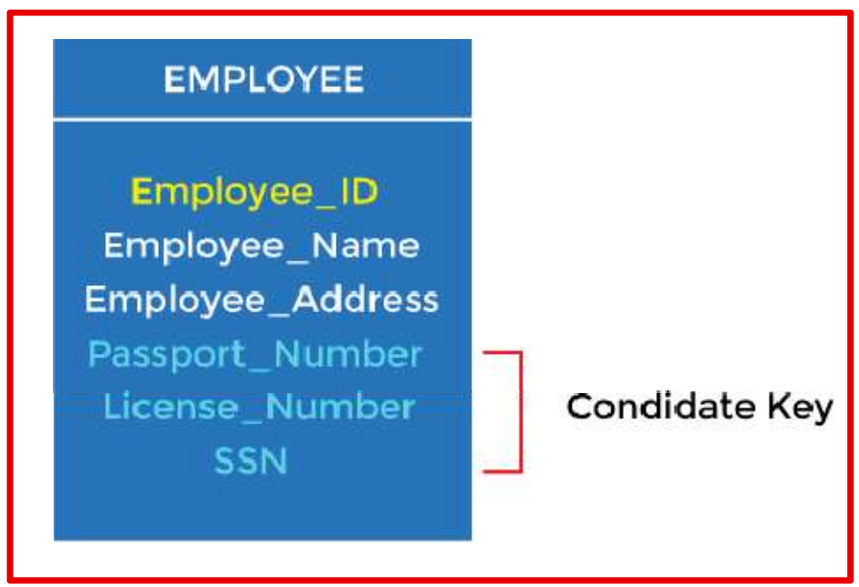
1. Primary key

- It is the first key used to identify one and only one instance of an entity uniquely. An entity can contain multiple keys, as we saw in the PERSON table. The key which is most suitable from those lists becomes a primary key.
- In the EMPLOYEE table, ID can be the primary key since it is unique for each employee. In the EMPLOYEE table, we can even select License_Number and Passport_Number as primary keys since they are also unique.
- For each entity, the primary key selection is based on requirements and developers.



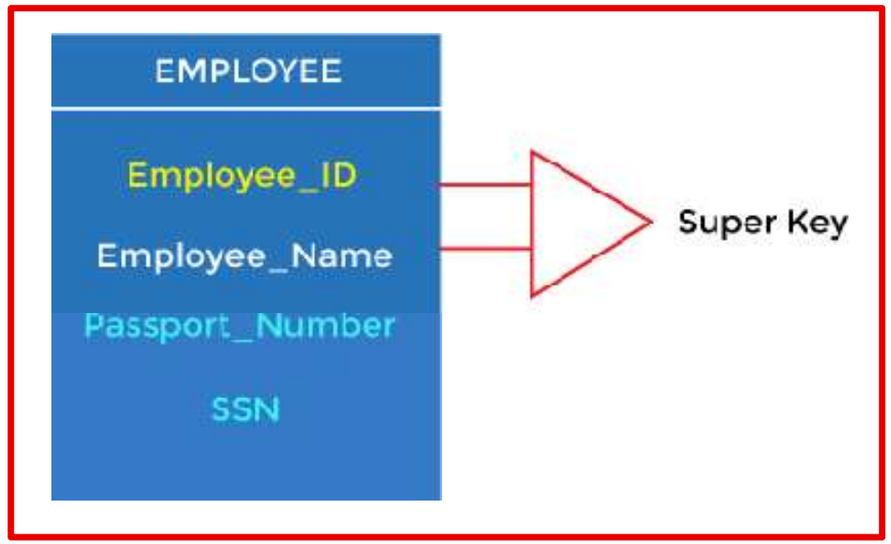
2. Candidate key

- A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Except for the primary key, the remaining attributes are considered a candidate key. The candidate keys are as strong as the primary key.
  - **For example:** In the EMPLOYEE table, id is best suited for the primary key. The rest of the attributes, like SSN, Passport_Number, License_Number, etc., are considered a candidate key.



3. Super Key

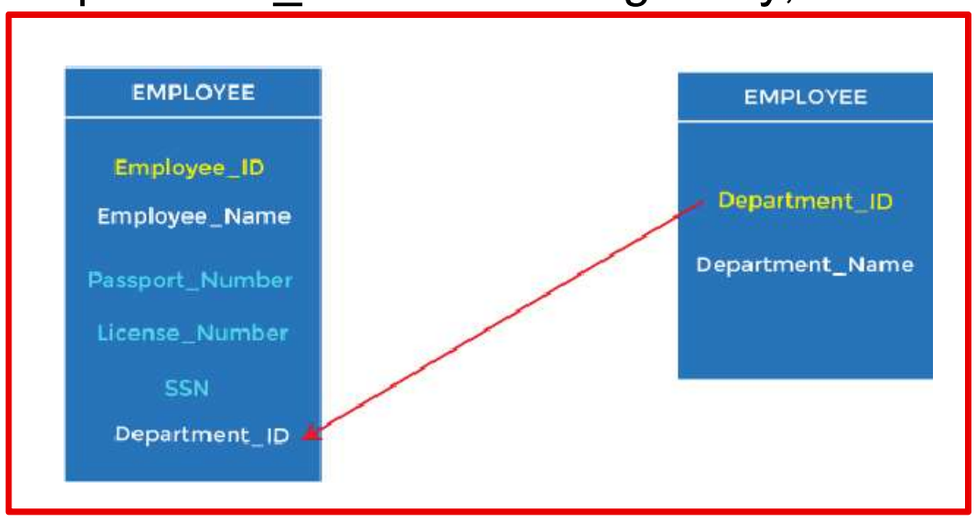
- Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.



- **For example:** In the above EMPLOYEE table, for(EMPLOEE_ID, EMPLOYEE_NAME), the name of two employees can be the same, but their EMPLYEE_ID can't be the same. Hence, this combination can also be a key.
- The super key would be EMPLOYEE-ID (EMPLOYEE_ID, EMPLOYEE-NAME), etc.

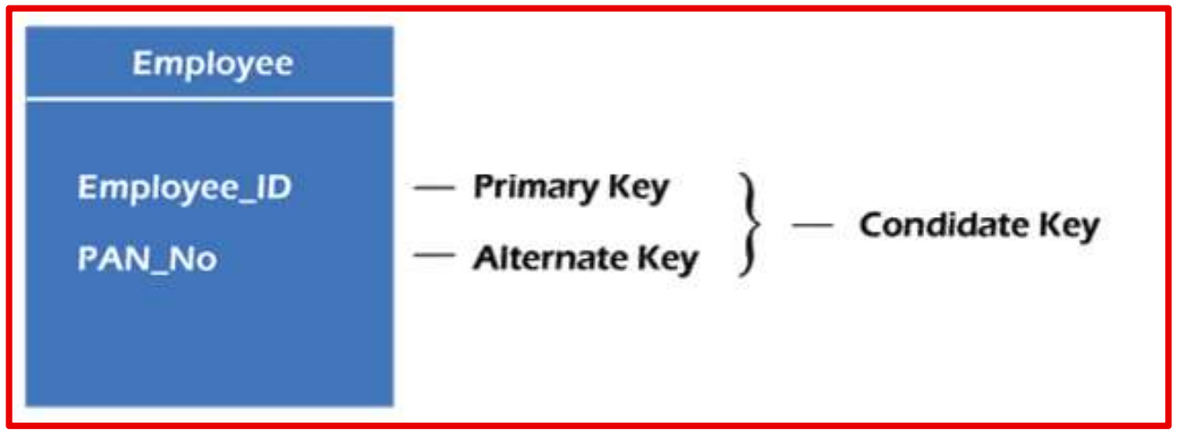
4. Foreign key

- Foreign keys are the column of the table used to point to the primary key of another table.
- Every employee works in a specific department in a company, and employee and department are two different entities. So we can't store the department's information in the employee table. That's why we link these two tables through the primary key of one table.
- We add the primary key of the DEPARTMENT table, Department_Id, as a new attribute in the EMPLOYEE table.
- In the EMPLOYEE table, Department_Id is the foreign key, and both the tables are related.



5. Alternate key

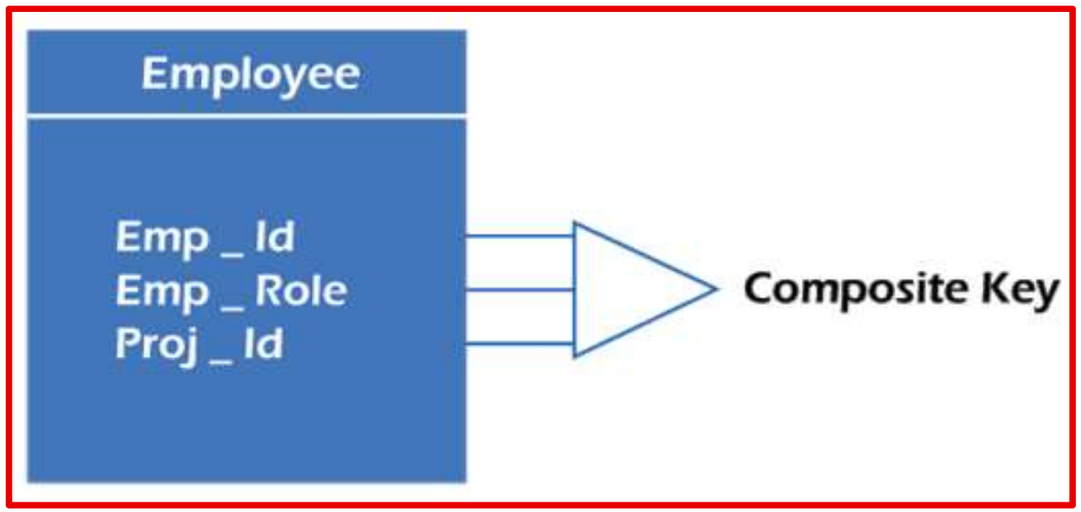
- There may be one or more attributes or a combination of attributes that uniquely identify each tuple in a relation.
- These attributes or combinations of the attributes are called the candidate keys. One key is chosen as the primary key from these candidate keys, and the remaining candidate key, if it exists, is termed the alternate key.



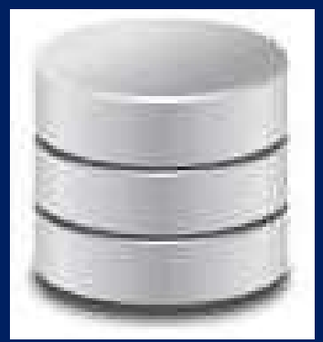
- **For example,** employee relation has two attributes, Employee_Id and PAN_No, that act as candidate keys. In this relation, Employee_Id is chosen as the primary key, so the other candidate key, PAN_No, acts as the Alternate key.

6. Composite key

- Whenever a primary key consists of more than one attribute, it is known as a composite key. This key is also known as Concatenated Key.



- **For example,** in employee relations, we assume that an employee may be assigned multiple roles, and an employee may work on multiple projects simultaneously. So the primary key will be composed of all three attributes, namely Emp_ID, Emp_role, and Proj_ID in combination. So these attributes act as a composite key since the primary key comprises more than one attribute.



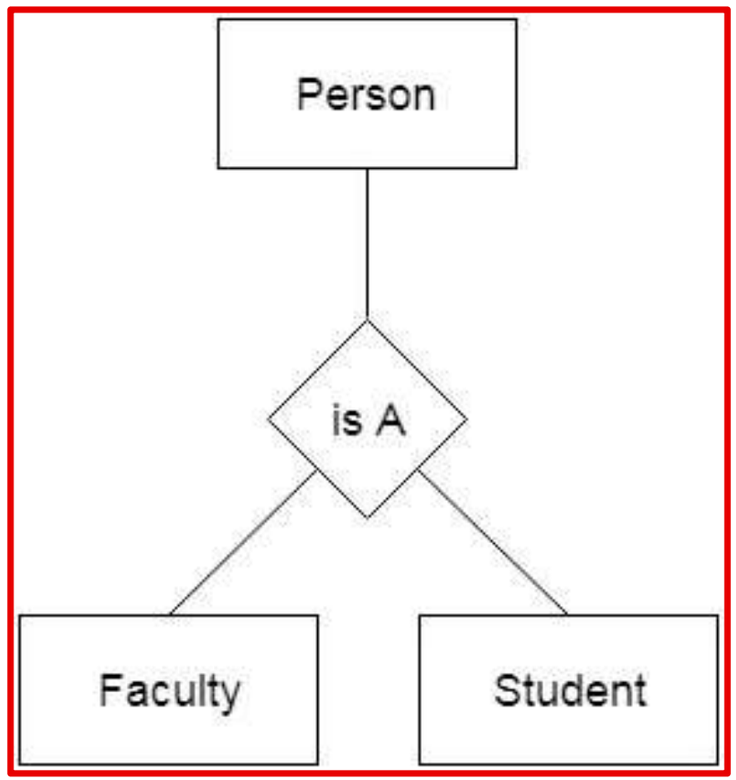
Generalization Specialization Aggregation

Generalization

- Generalization is like a bottom-up approach in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- In generalization, an entity of a higher level can also combine with the entities of the lower level to form a further higher level entity.
- Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom-up approach.
- In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass.

Eg :

- Faculty and Student entities can be generalized and create a higher level entity Person.

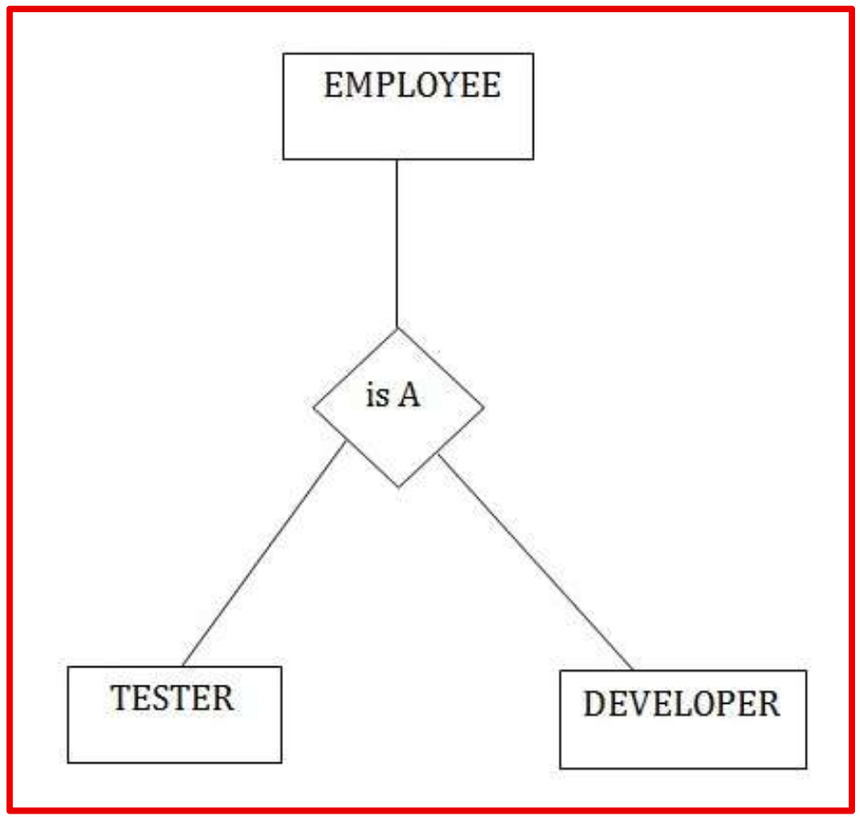


Specialization

- Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added.

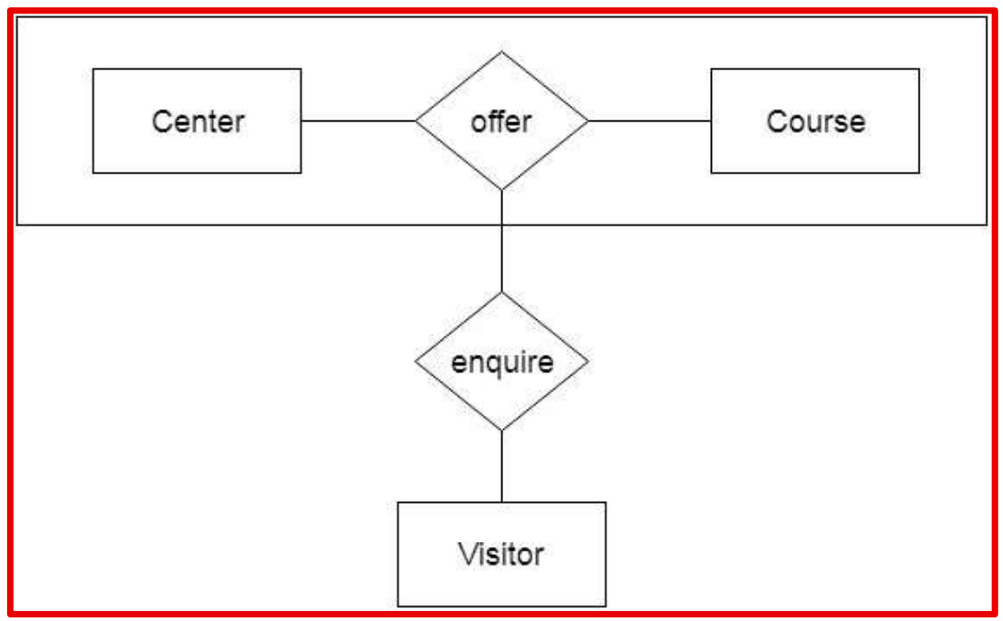
Eg:

- In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.



Aggregation

- In aggregation, the relation between two entities is treated as a single entity. In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.
- **For example:** Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.



Steps Involved in Creation of ERD

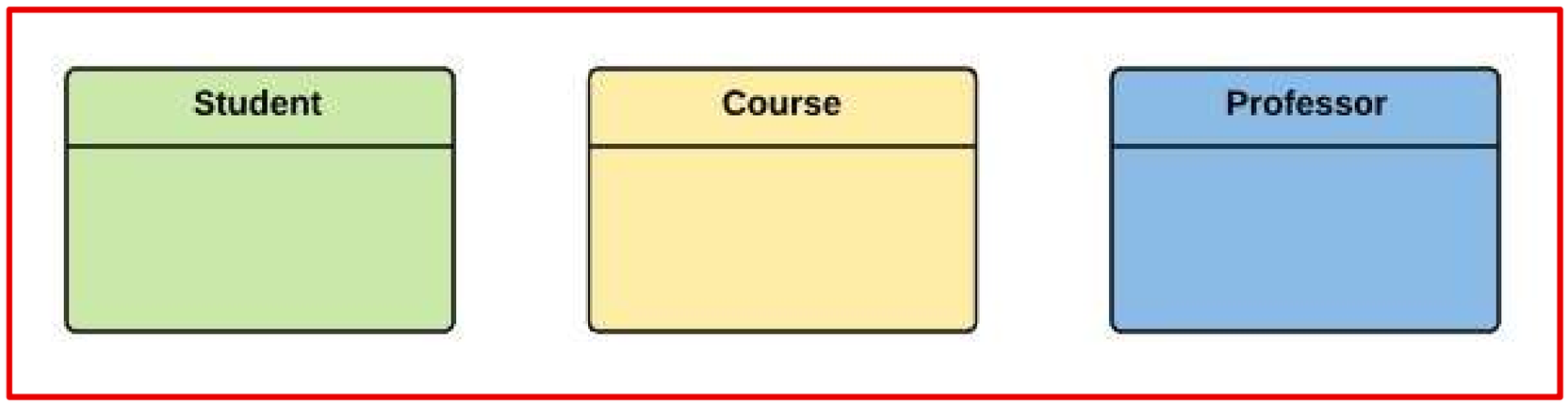
Steps Involved in Creation of Entity Relationship Diagram (ERD)

Steps to Create an ER Diagram

```
In a university, a Student enrolls in Courses. A student must be assigned to at least
one or more Courses. Each course is taught by a single Professor. To maintain
instruction quality, a Professor can deliver only one course
```

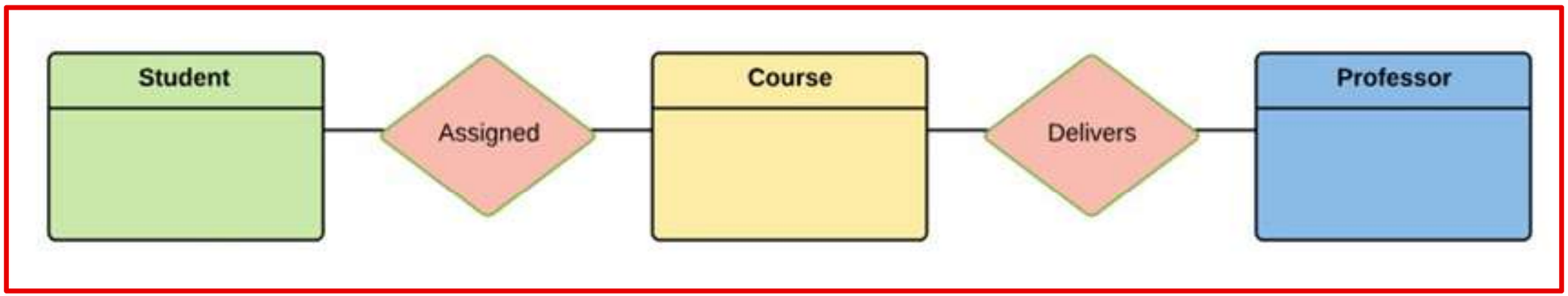
1. Entity Identification

- We have three entities
  1. Student
  2. Course
  3. Professor



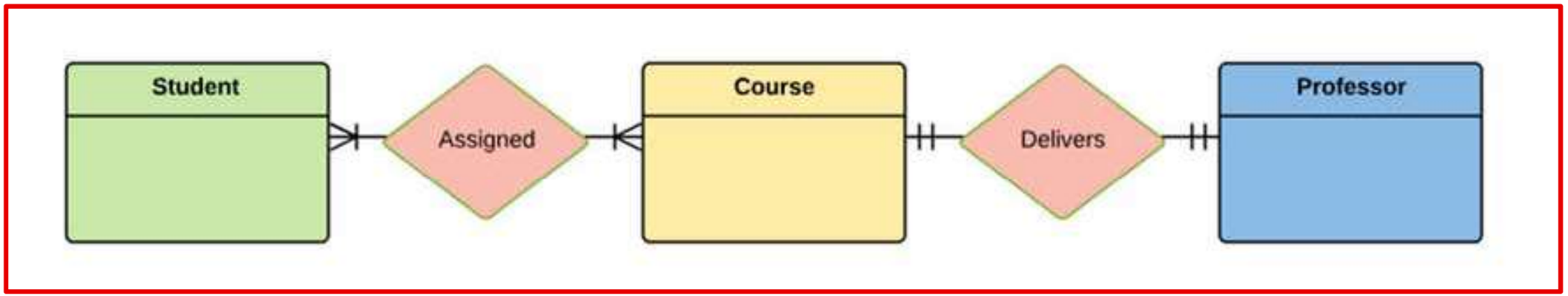
2. Relationship Identification

- We have the following two relationships
  1. The student is **assigned** a course
  2. Professor **delivers** a course



3. Cardinality Identification

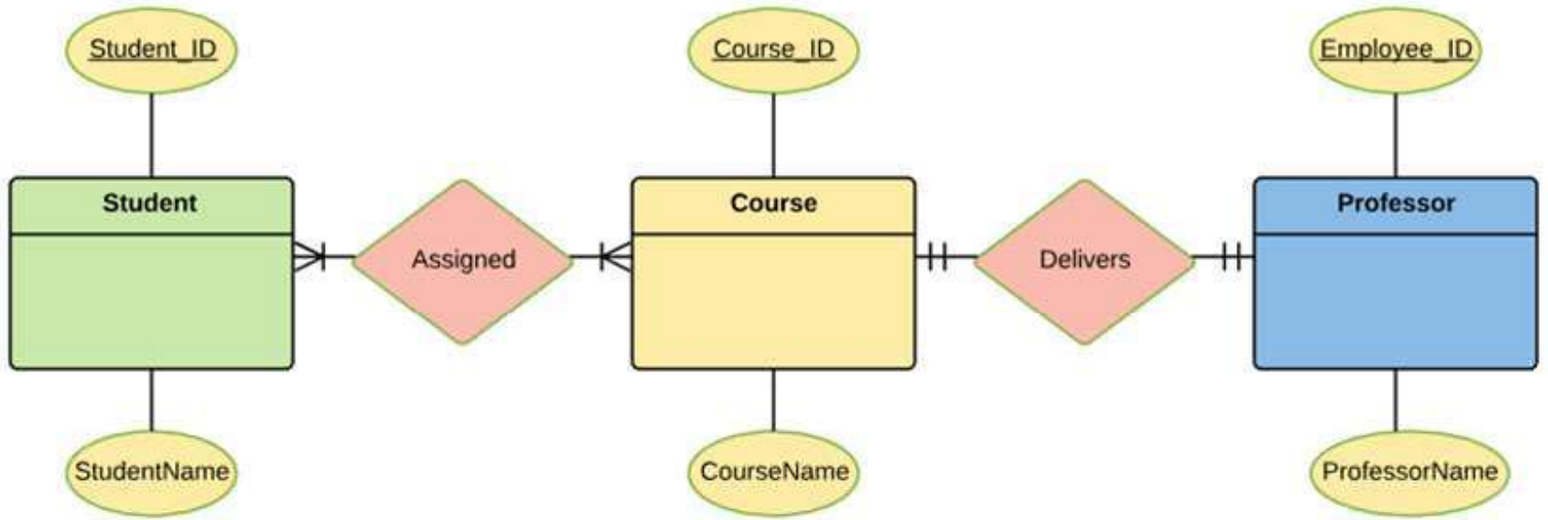
- For them problem statement we know that,
  1. A student can be assigned **multiple** courses
  2. A Professor can deliver only **one** course



4. Identify Attributes

- You need to study the files, forms, reports, data currently maintained by the organization to identify attributes. You can also conduct interviews with various stakeholders to identify entities. Initially, it's important to identify the attributes without mapping them to a particular entity.
- Once, you have a list of Attributes, you need to map them to the identified entities. Ensure an attribute is to be paired with exactly one entity. If you think an attribute should belong to more than one entity, use a modifier to make it unique.
- Once the mapping is done, identify the primary Keys. If a unique key is not readily available, create one.

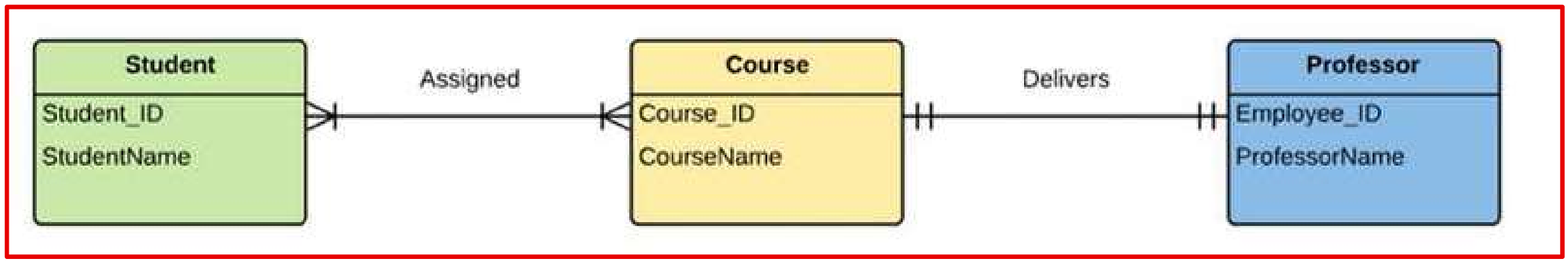
Entity	Primary Key	Attribute
Student	Student_ID	StudentName
Professor	Employee_ID	ProfessorName
Course	Course_ID	CourseName



For Course Entity, attributes could be Duration, Credits, Assignments, etc. For the sake of ease we have considered just one attribute.

5. Create the ERD Diagram

- A more modern representation of Entity Relationship Diagram Example



Best Practices for Developing Effective ER Diagrams

1. Eliminate any redundant entities or relationships
2. You need to make sure that all your entities and relationships are properly labeled
3. There may be various valid approaches to an ER diagram. You need to make sure that the ER diagram supports all the data you need to store
4. You should assure that each entity only appears a single time in the ER diagram
5. Name every relationship, entity, and attribute are represented on your diagram
6. Never connect relationships to each other
7. You should use colors to highlight important portions of the ER diagram

Definition of a Relational Database

- A relational database is a collection of relations or two-dimensional tables controlled by the DB Server.

DB server

Table name: EMPLOYEES

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL
100	Steven	King	SKING
101	Neena	Kochhar	NKOCHHAR
102	Lex	De Haan	LDEHAAN

...

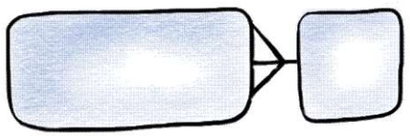
Table name: DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
10	Administration	200
20	Marketing	201
50	Shipping	124

...

Data Models

Model of system in client's mind



Entity model of client's model

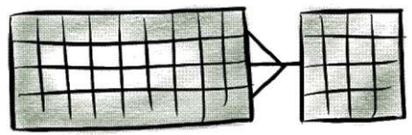
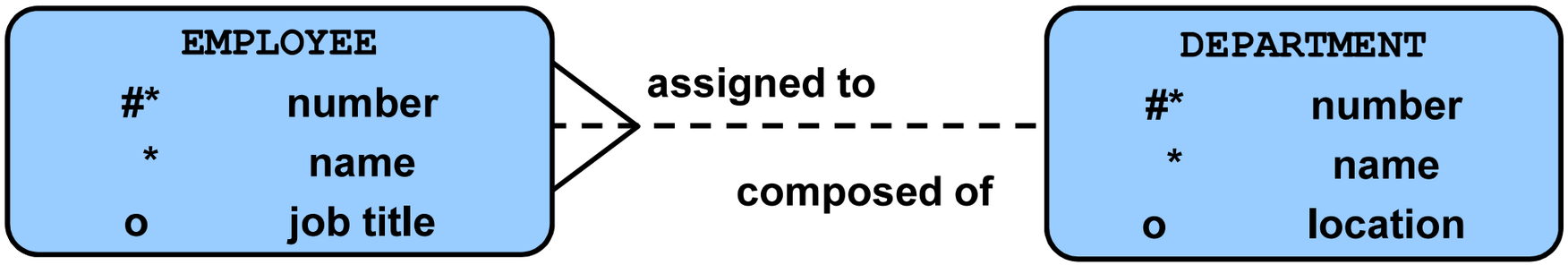


Table model of entity model

Oracle server

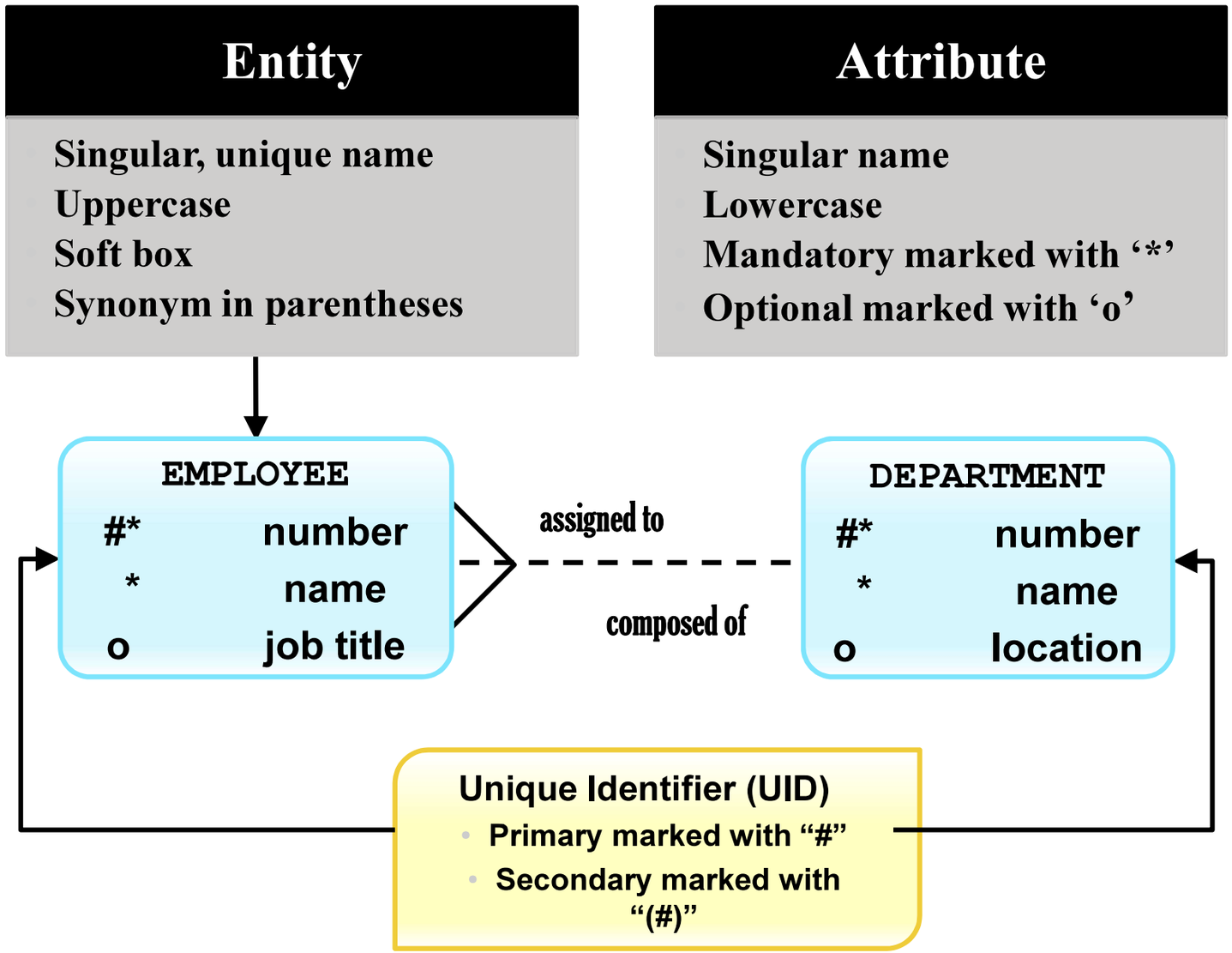
Tables on disk

- Create an entity relationship diagram from business specifications or narratives:



- Scenario:
  - ". . . Assign one or more employees to a department . . ."
  - ". . . Some departments do not yet have assigned employees . . ."

Entity Relationship Modeling Conventions



Relating Multiple Tables

- Each row of data in a table can be uniquely identified by a primary key.
- You can logically relate data from multiple tables using foreign keys.

Table name: `DEPARTMENTS`

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700

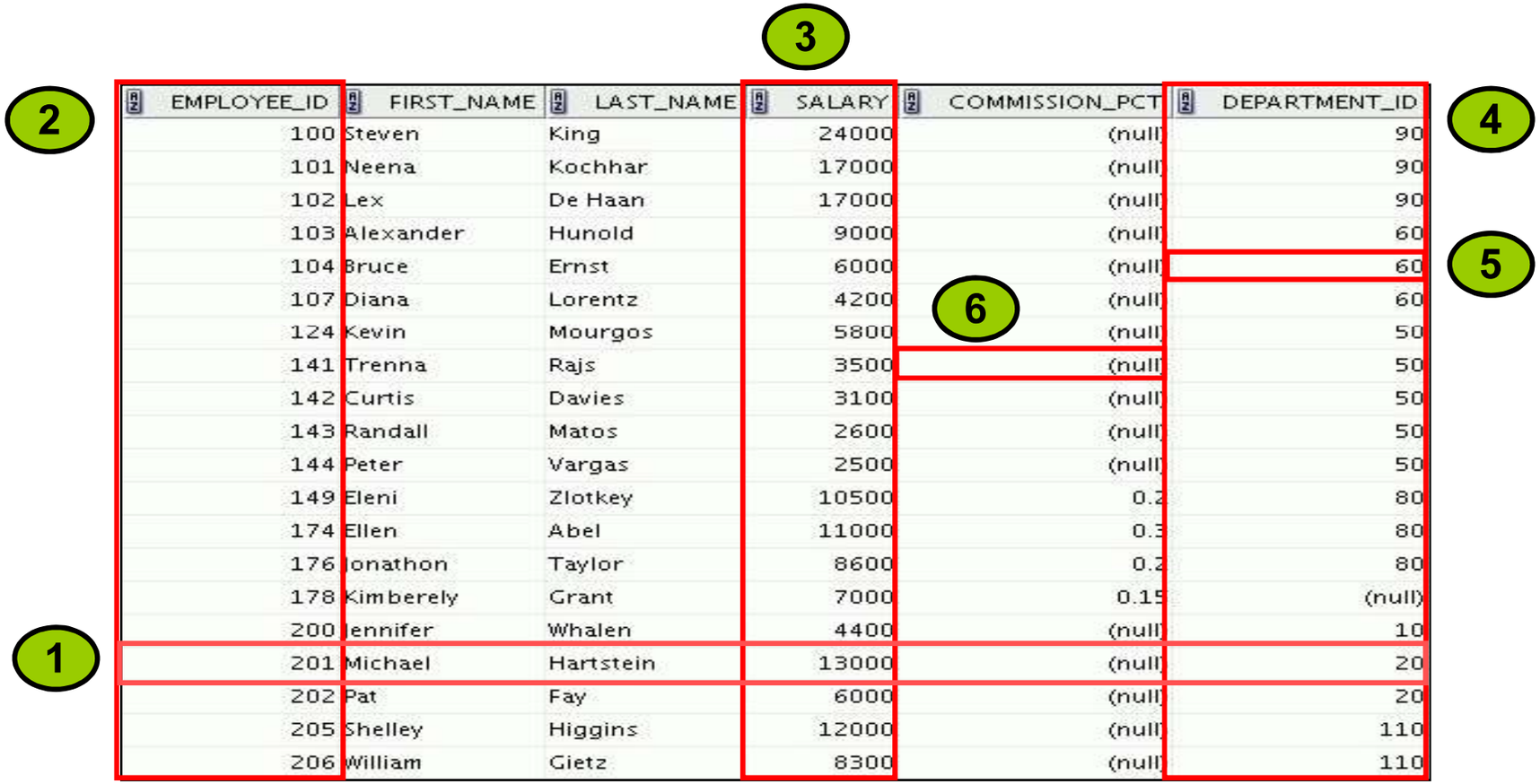
Primary key

Table name: `EMPLOYEES`

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
103	Alexander	Hunold	60
104	Bruce	Ernst	60
107	Diana	Lorentz	60
124	Kevin	Mourgos	50
141	Trenna	Rajs	50
142	Curtis	Davies	50

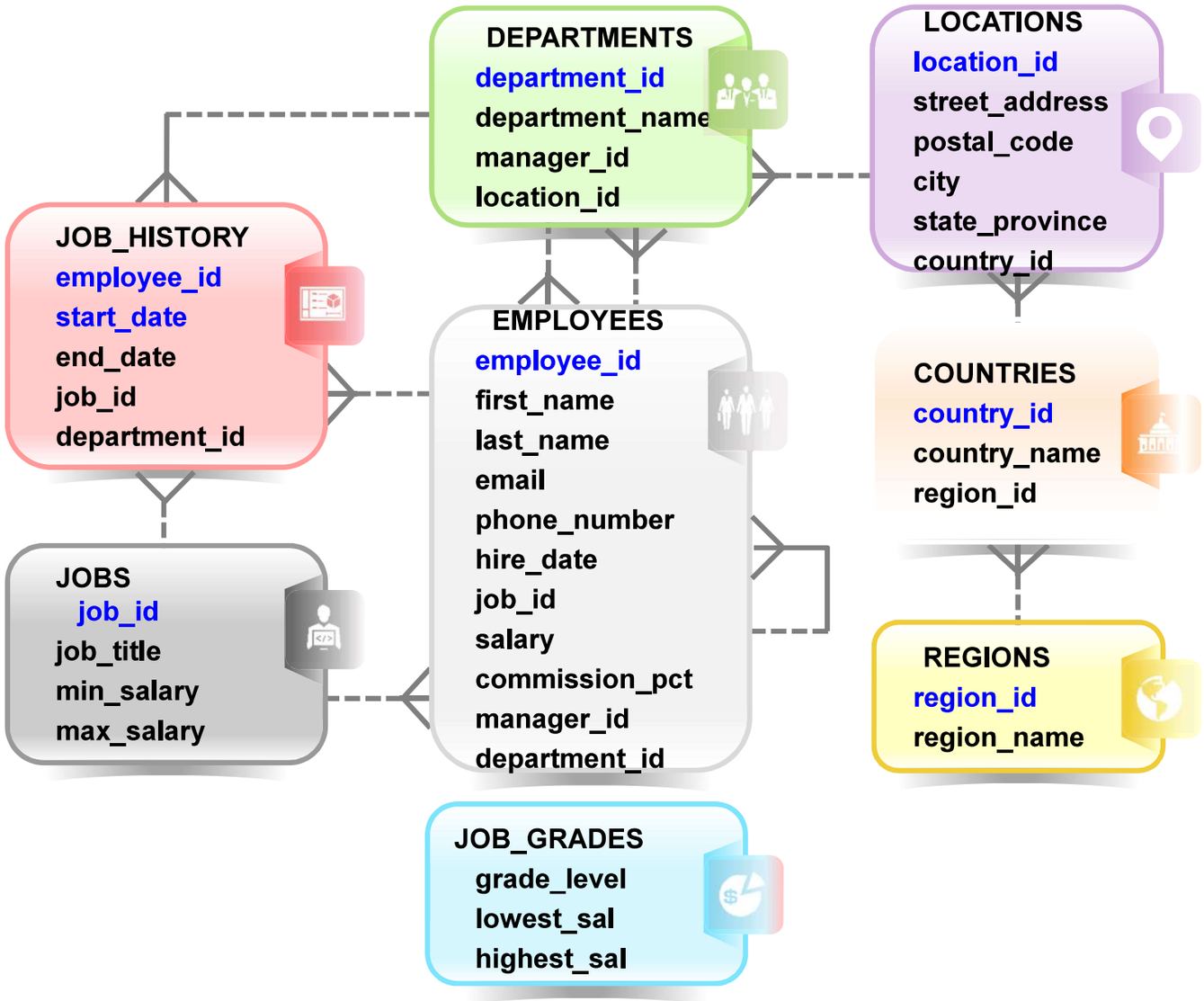
Primary key **Foreign key**

Relational Database Terminology



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
100	Steven	King	24000	(null)	90
101	Neena	Kochhar	17000	(null)	90
102	Lex	De Haan	17000	(null)	90
103	Alexander	Hunold	9000	(null)	60
104	Bruce	Ernst	6000	(null)	60
107	Diana	Lorentz	4200	(null)	60
124	Kevin	Mourgos	5800	(null)	50
141	Trenna	Rajs	3500	(null)	50
142	Curtis	Davies	3100	(null)	50
143	Randall	Matos	2600	(null)	50
144	Peter	Vargas	2500	(null)	50
149	Eleni	Zlotkey	10500	0.2	80
174	Ellen	Abel	11000	0.3	80
176	Jonathon	Taylor	8600	0.2	80
178	Kimberely	Grant	7000	0.15	(null)
200	Jennifer	Whalen	4400	(null)	10
201	Michael	Hartstein	13000	(null)	20
202	Pat	Fay	6000	(null)	20
205	Shelley	Higgins	12000	(null)	110
206	William	Gietz	8300	(null)	110

Revisiting Human Resources (HR) Schema



Summary

In this lesson, you should have learned that:

- Entity Relationship Diagram
- ERD Entities Attributes and Relationship Notations
- Mapping Constraints
- Keys
- Generalization Specialization and Aggregation
- Steps Involved in Creation of ER Diagram