
ORACLE
ACADEMY

Database Design

2-2
Entities, Instances, Attributes, and Identifiers

ORACLE ACADEMY

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.

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.

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.

Terminology

Key terms used in this lesson included:

- Attribute
- Data type
- Entity
- Instance
- Mandatory
- Intangible

Terminology

Key terms used in this lesson included:

- Null
- Optional
- Single valued
- Tangible
- Unique identifier (UID)
- Volatile

Database Design

3-1
Identifying Relationships

ORACLE ACADEMY

Relationships in Families

- A relationship is the way in which two or more people or things are connected.
- Family relationships categorize relationships between people, for example mother, father, aunt and cousin.
- The name of the relationship tells us how the family members are connected.

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

Database Design

3-3
Speaking ERDish & Drawing Relationships

ORACLE ACADEMY

ERD Language

- ERDish is the language we use to state relationships between entities in an ERD.
- You have already been speaking and writing it, when you identified relationships and specified optionality and cardinality.
- We are simply breaking down each ERDish sentence into its components.

Breaking Down ERDish

The Components of ERDish

- EACH
- Entity A
- OPTIONALITY (must be/may be)
- RELATIONSHIP NAME
- CARDINALITY (one and only one/one or more)
- Entity B

The Components of ERDish

- Since each relationship has two sides, we read the first relationship from left to right (or top to bottom, depending on the ERD layout).

1. EACH
2. Entity A
3. OPTIONALITY (must be/may be)
4. RELATIONSHIP NAME
5. CARDINALITY (one and only one/one or more)
6. Entity B

Breaking Down ERDish

1. EACH
2. **EMPLOYEE** (entity A)
3. **MUST** (optionality, solid line)
4. **WORK IN** (relationship name)
5. **ONE** (AND ONLY ONE) (cardinality, single toe)
6. **DEPARTMENT** (entity B)

The Components of ERDish

- Now we read the relationship from right to left.

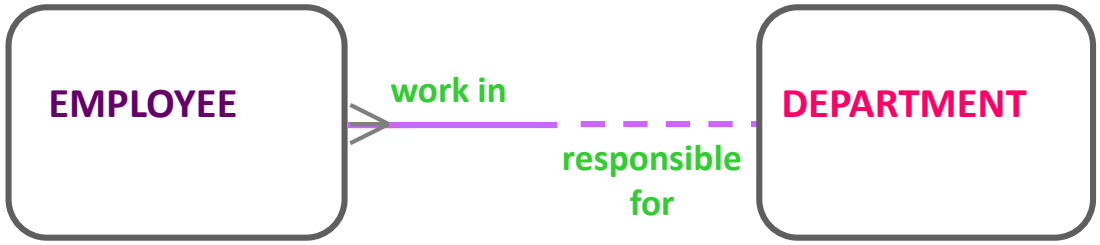
1. EACH
2. Entity B
3. OPTIONALITY (must be/may be)
4. RELATIONSHIP NAME
5. CARDINALITY (one and only one/one or more)
6. Entity A

Breaking Down ERDish

1. EACH
2. **DEPARTMENT** (entity B)
3. **MAY BE** (optionality, dotted line)
4. **RESPONSIBLE FOR** (relationship name)
5. **ONE OR MORE** (cardinality, crow's foot)
6. **EMPLOYEE** (entity A)

The Components of ERDish

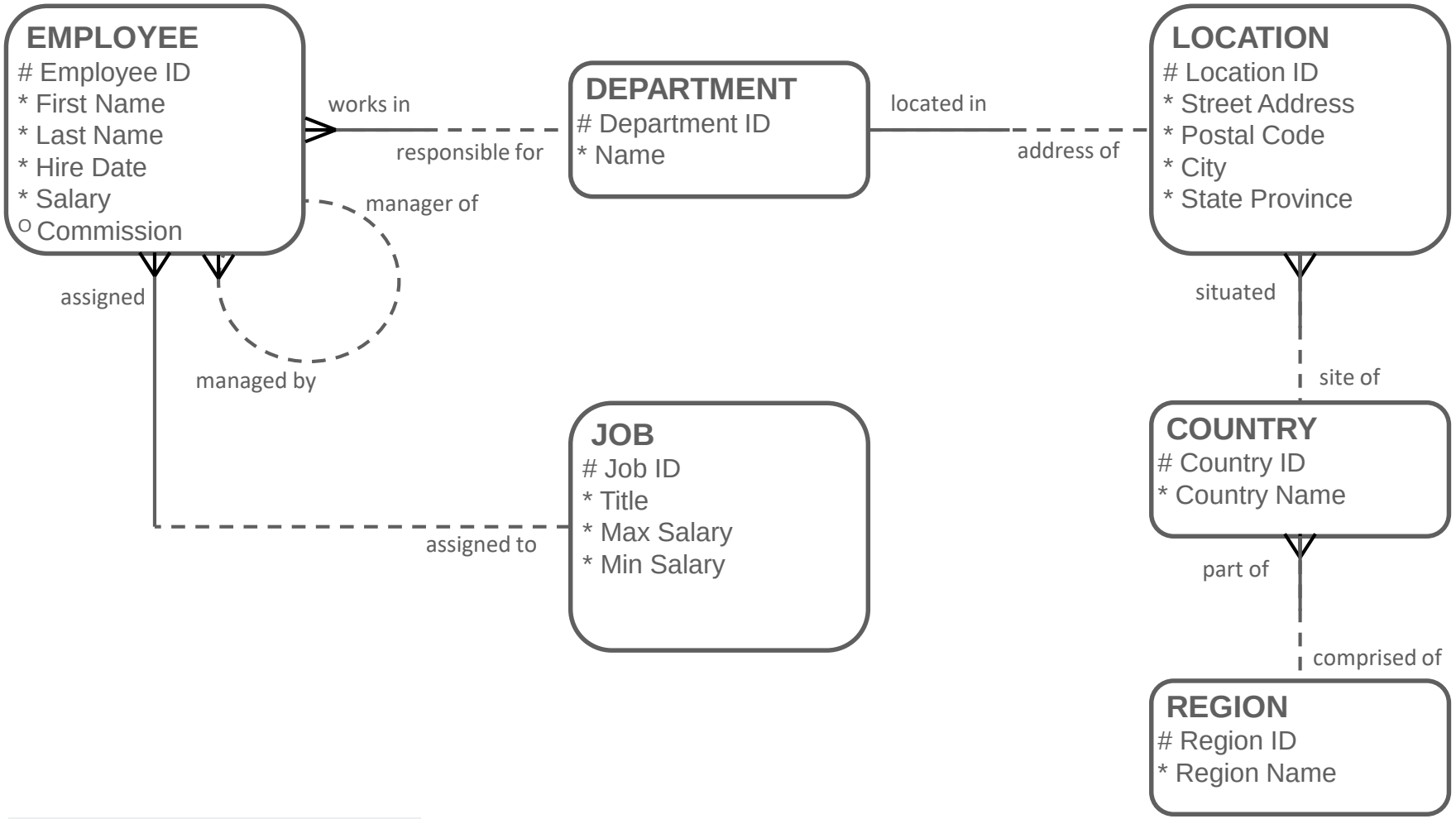
- Now bring it all together.



1. EACH
2. **EMPLOYEE** (entity A)
3. **MUST** (optionality, solid line)
4. **WORK IN** (relationship name)
5. **ONE AND ONLY ONE** (cardinality, single toe)
6. **DEPARTMENT** (entity B)

1. EACH
2. **DEPARTMENT** (entity B)
3. **MAY BE** (optionality, dotted line)
4. **RESPONSIBLE FOR** (relationship name)
5. **ONE OR MORE** (cardinality, crow's foot)
6. **EMPLOYEE** (entity B)

H.R Department ERD



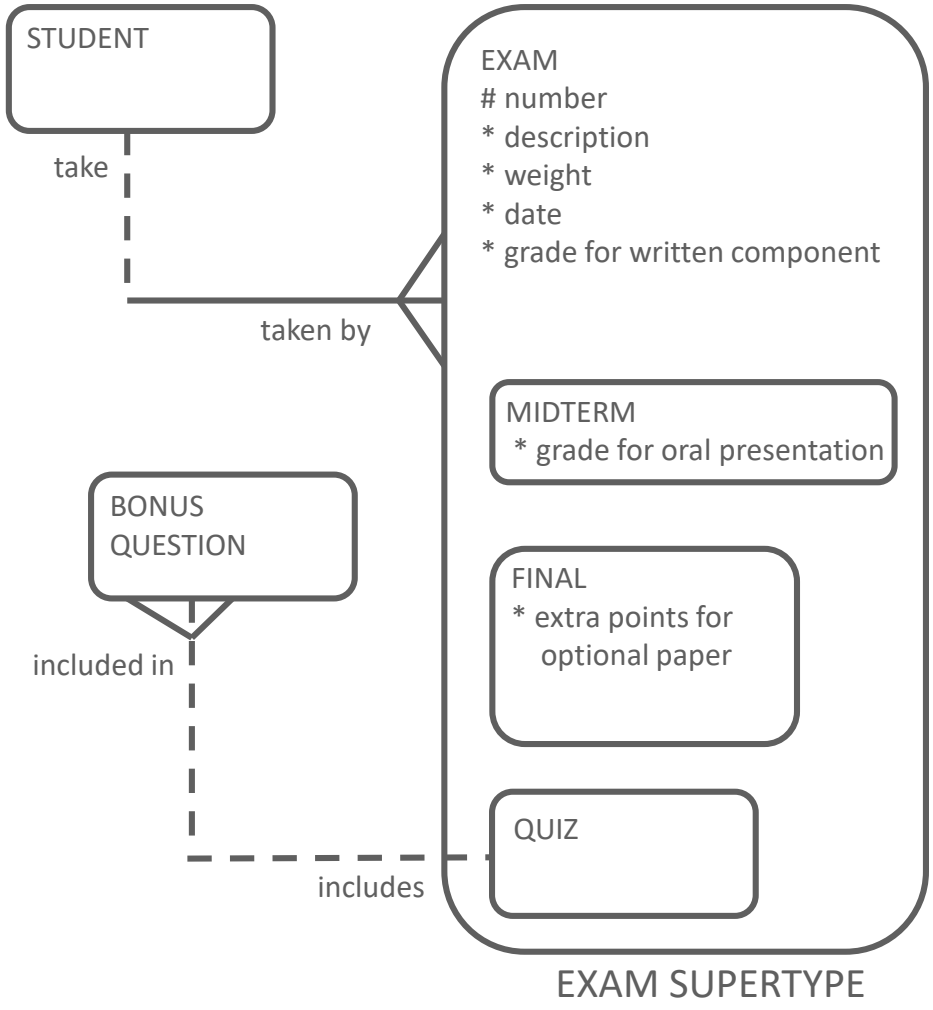
Database Design

4-1
Supertypes and Subtypes

ORACLE ACADEMY

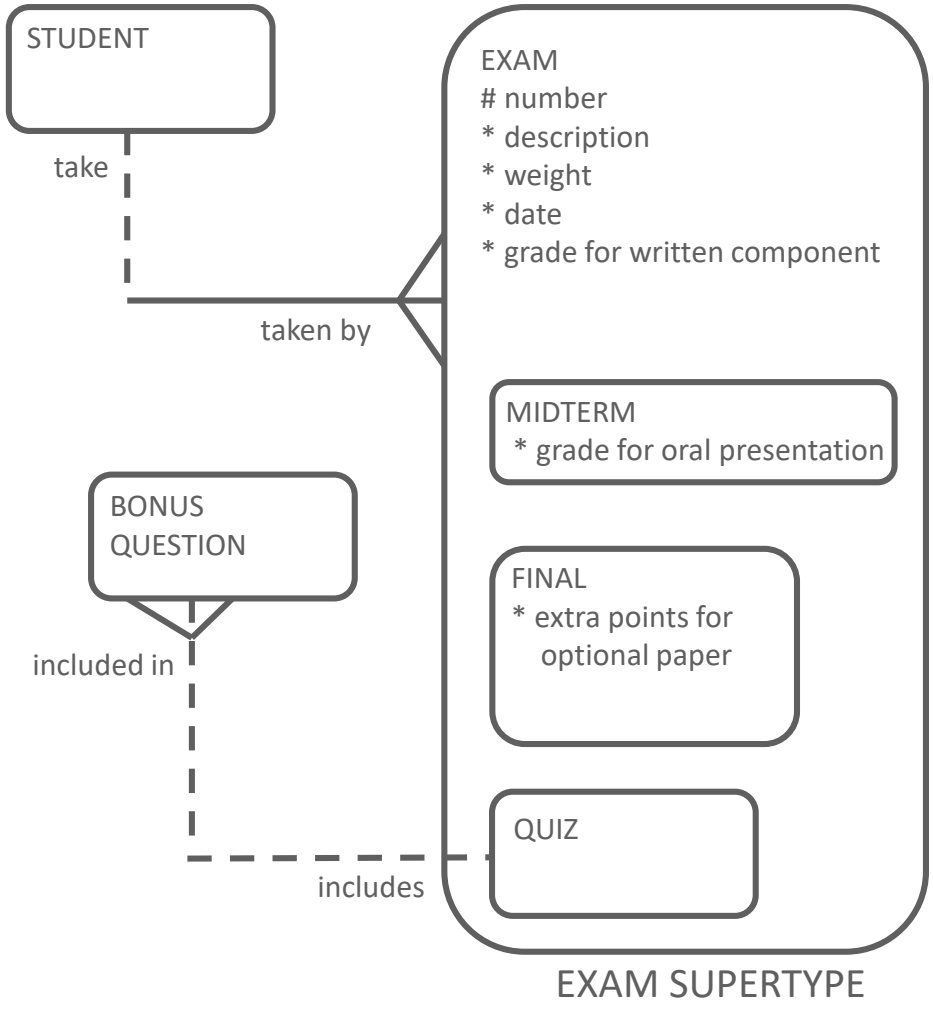
Supertype Example

- EXAM is a supertype of QUIZ, MIDTERM, and FINAL.
- The subtypes have several attributes in common.
- These common attributes are listed at the supertype level.



Supertype Example

- The same applies to relationships.
- Subtypes inherit all attributes and relationships of the supertype entity.



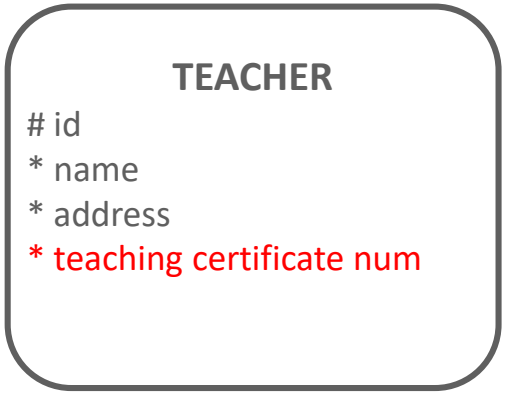
Database Design

4-2
Documenting Business Rules

ORACLE ACADEMY

Structural Rule Example

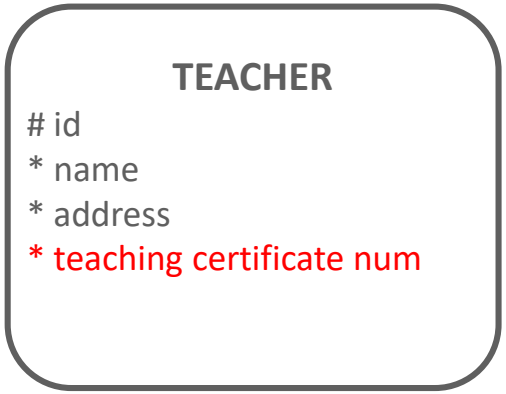
- Structural business rules indicate the types of information to be stored (attributes) and how the information elements interrelate (relationships).
- Here are a few examples:



Structural Business Rules represented in ERD models

Structural Rule Example

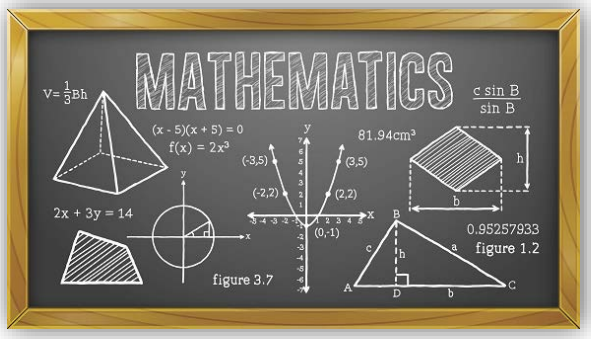
- All orders at a restaurant must be handled by a staff member (specifically, an order taker).
- There is no self-service ordering system.
- All teachers at our school must possess a valid teaching certificate.



Structural Business Rules represented in ERD models

Procedural Rule Example

- "Students must have studied algebra and geometry in order to sign up for trigonometry."
- Could you represent this in the ERD?
- How would you implement this with programming?
- If the student had taken the subjects, can you think of an additional business rule that a school may want in this scenario?



Documenting Rules

- In the process of developing a conceptual data model, not all business rules can be modeled.
- Some rules such as the two listed below must be implemented by programming the processes that interact with data:
  - Any employee whose overtime exceeds 10 hours per week must be paid 1.5 times the hourly rate.
  - Customers whose account balances are 90 days overdue will not be permitted to charge additional orders.

Terminology

Key terms used in this lesson included:

- Business rule
- Procedural business rule
- Structural business rule

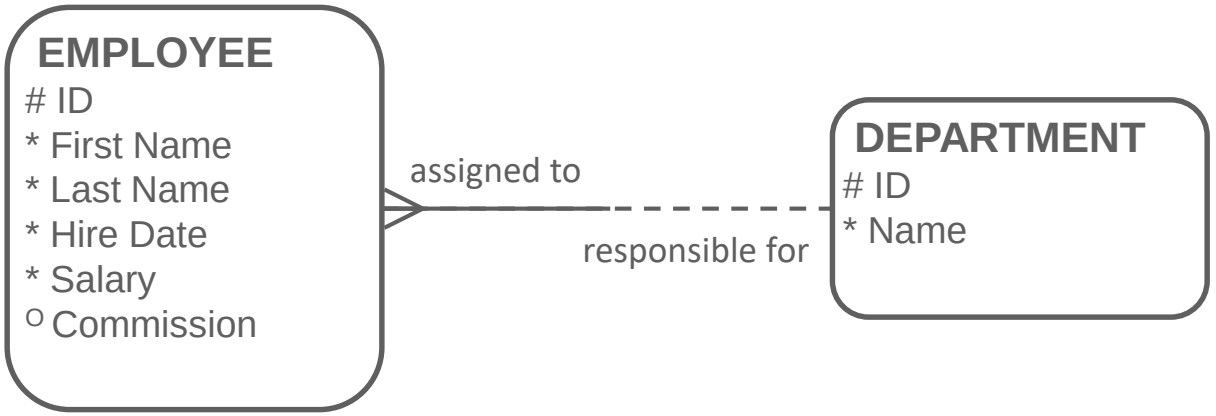
Database Design

5-1
Relationship Transferability

ORACLE ACADEMY

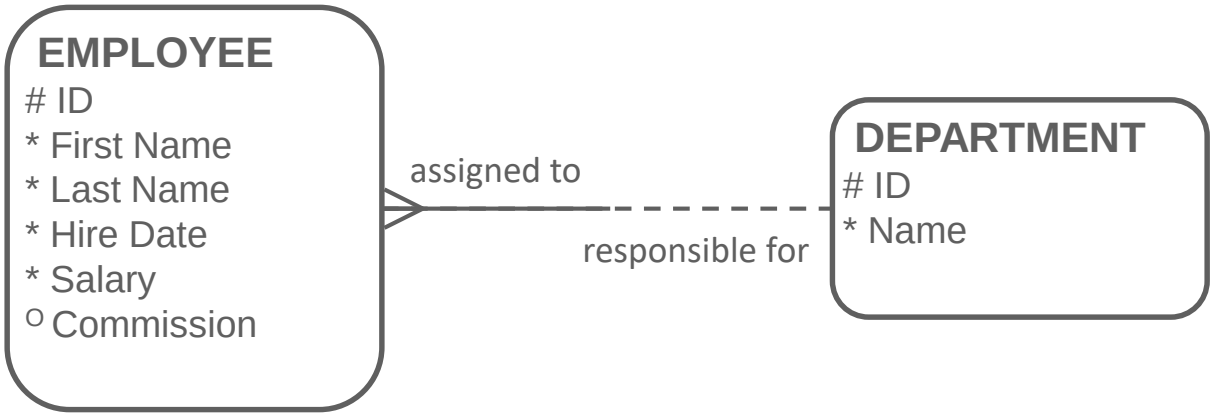
Relationship Review

- Let's review a simple relationship between EMPLOYEE and DEPARTMENT.
- Optionality:
  - Must every EMPLOYEE be assigned to a DEPARTMENT?
  - Must every DEPARTMENT be responsible for an employee?



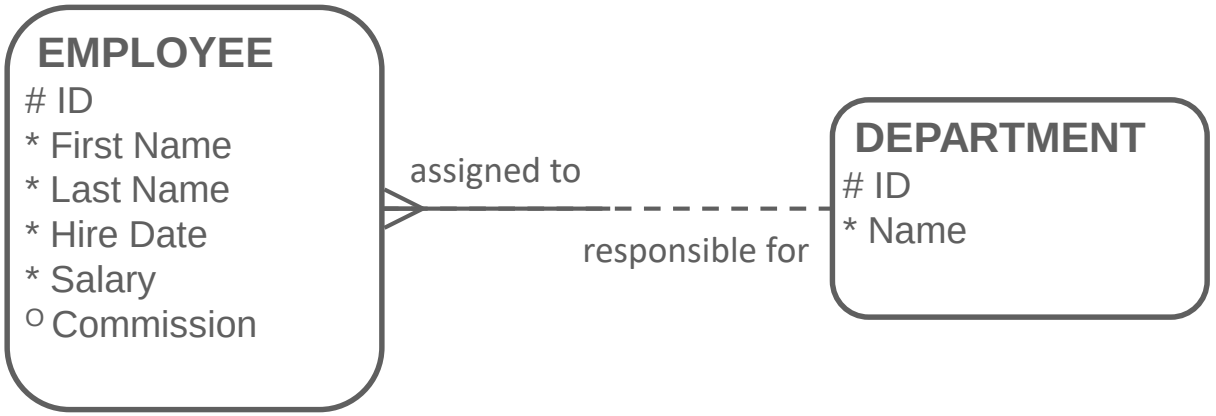
Relationship Review

- Cardinality:
  - How many EMPLOYEEs can a DEPARTMENT be responsible for?
  - How many DEPARTMENTs can an EMPLOYEE be assigned to?



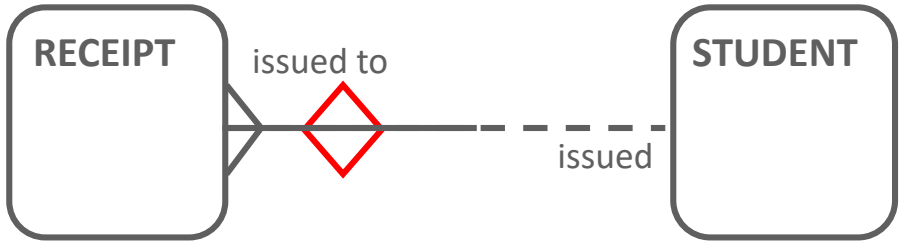
Relationship Review

- Transferability:
  - Can an EMPLOYEE be transferred from one DEPARTMENT to another DEPARTMENT?



Relationship Transferability

- Nontransferable: A STUDENT can be issued a RECEIPT for paying tuition fees, taking a certification exam, or purchasing items at the bookstore.
- Once a RECEIPT has been issued, it cannot be transferred to another STUDENT.



Nontransferable Relationship

A nontransferable relationship is represented with the diamond on the relationship.

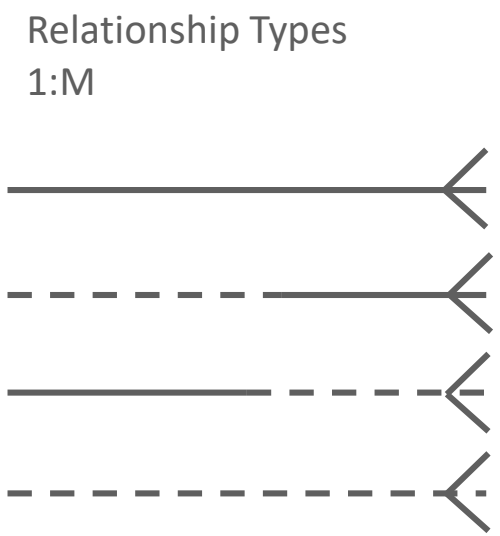
Database Design

5-2
Relationship Types

ORACLE ACADEMY

One-to-Many (1:M) Relationships

- The various types of 1:M relationships are most common in an ER Model.
- You have seen several examples already.

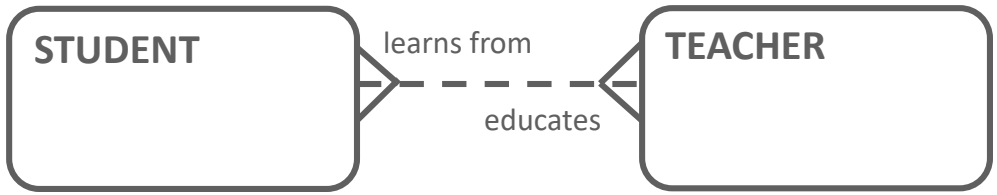


1:M Relationship

Many-to-Many (M:M) Relationships

- The various types of M:M relationships are common, particularly in a first version of an ER model.
- In later stages of the modeling process, all M:M relationships will be resolved, and disappear.

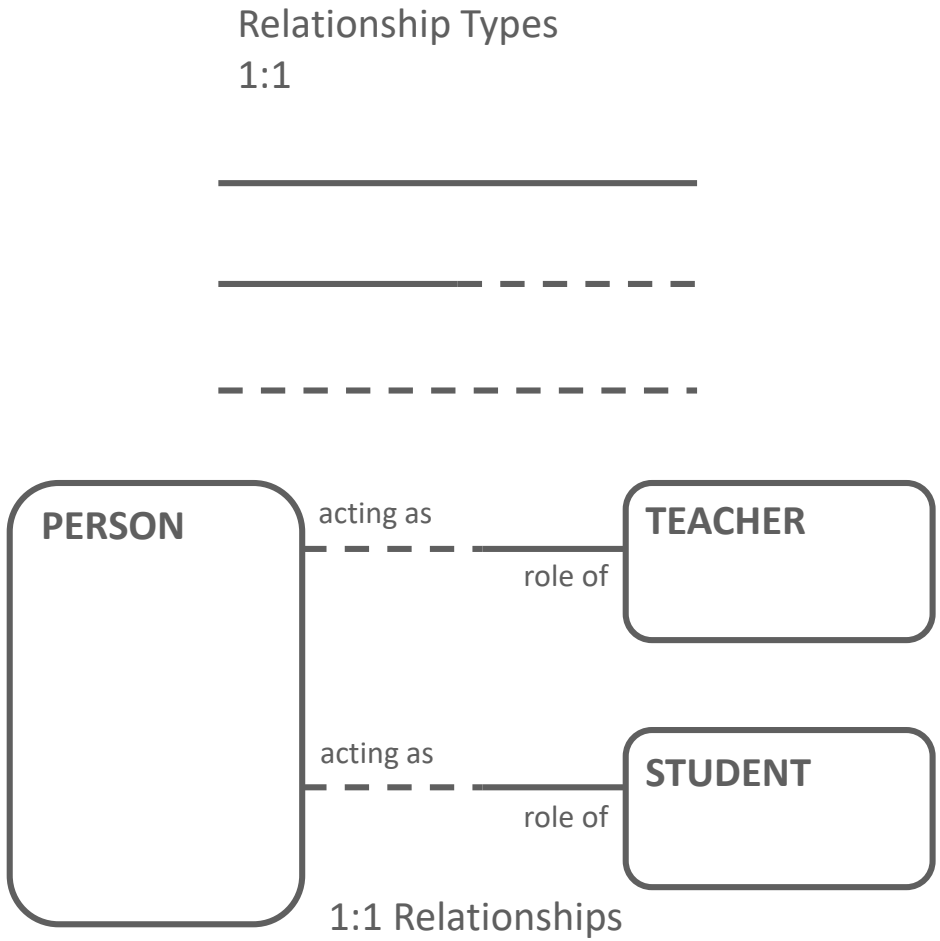
Relationship Types
M:M



M:M Relationships

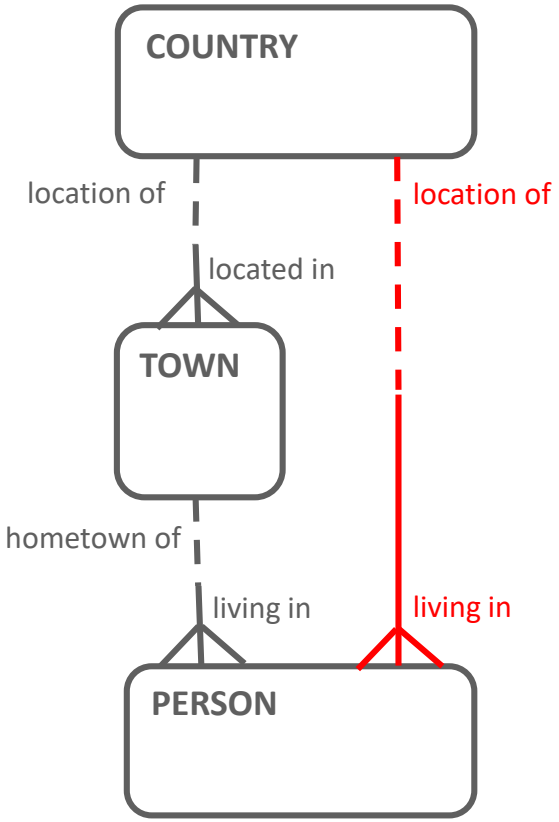
One-to-One Relationships For Roles

- Usually you will find just a few of the various types of 1:1 relationships in every ER model.
- Mandatory at one end of the 1:1 relationship commonly occurs when roles are modeled.



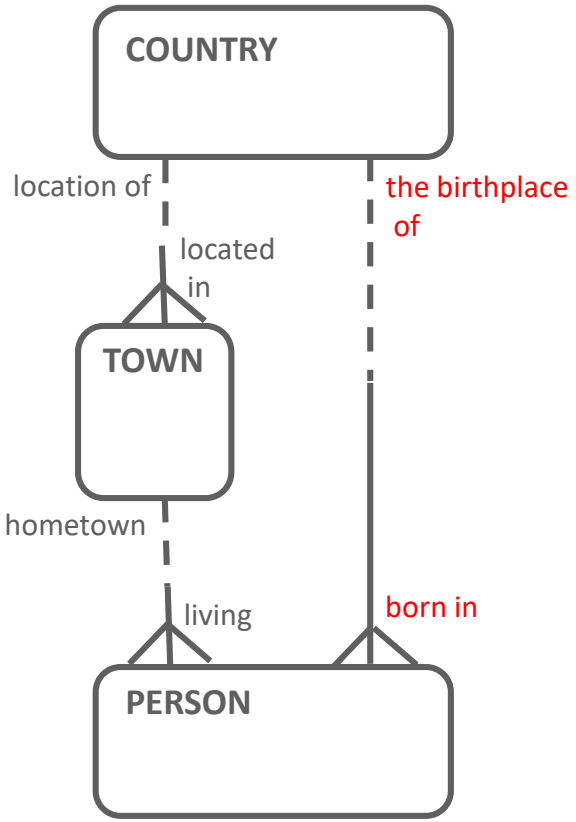
Redundant Relationships

- A redundant relationship can be derived from another relationship in the model.
- In this example, you can derive the relationship from PERSON to COUNTRY from the other two relationships (COUNTRY to TOWN, TOWN to PERSON), so you should remove the direct relationship from COUNTRY to PERSON .



Redundant Relationships

- However, be careful of concluding that a relationship is redundant based on the structure alone.
- Read the relationships to check.
- The ERD shown here does not reflect a redundant relationship.



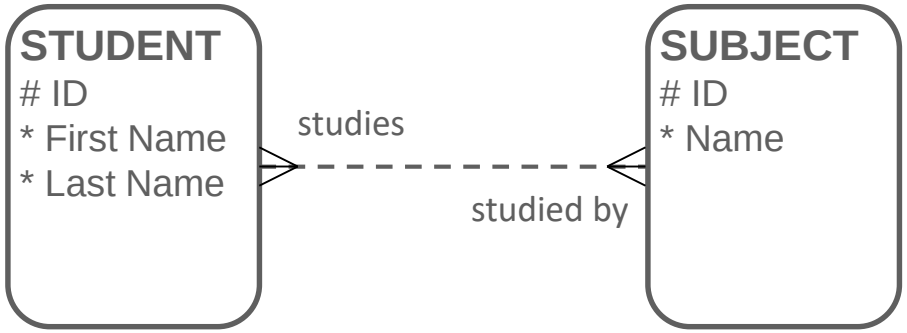
Database Design

5-3
Resolving Many-to-Many Relationships

ORACLE ACADEMY

Relationship Hiding an Attribute

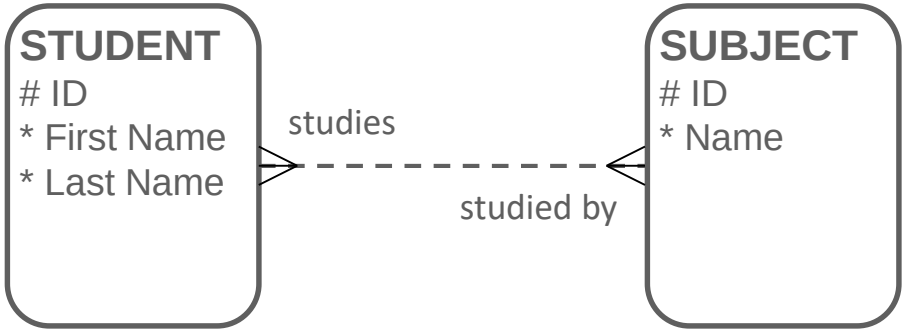
- In a school, a STUDENT may study one or more SUBJECTs.
- Each SUBJECT may be studied by one or more STUDENTs



STUDENT and SUBJECT

Relationship Hiding an Attribute

- When a student enrolls for a subject, we want to be able to record the grade they attain for that subject.
- Which entity would the attribute “Grade” belong to?
- If we put “Grade” in the STUDENT entity, how would we know which SUBJECT it is for?
- If we put “Grade” in the SUBJECT entity, how would we know which STUDENT got that grade?

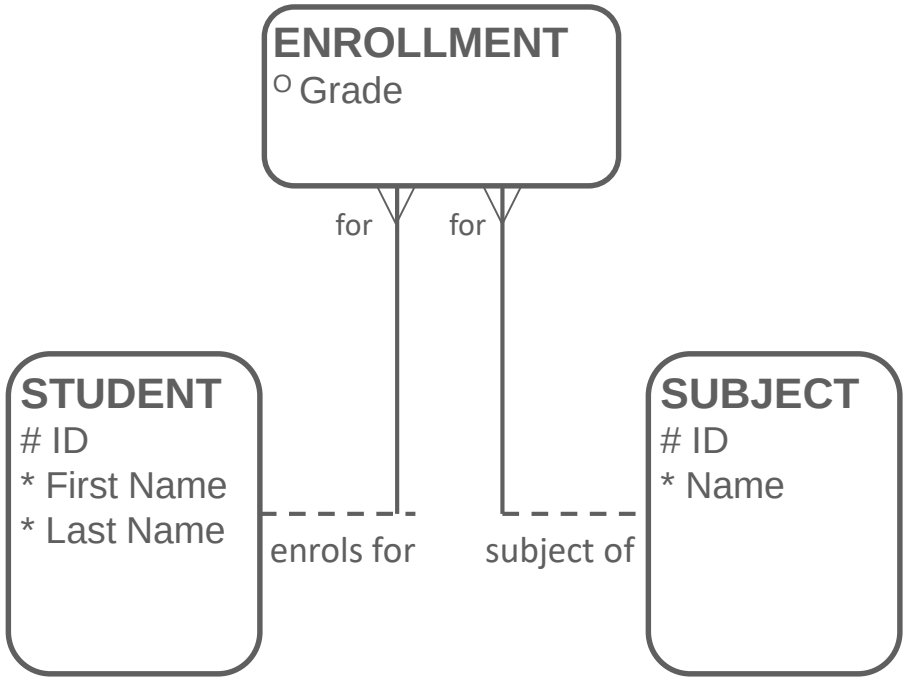


Resolution of a M:M Relationship

- A third entity is needed to resolve the M:M relationship. This is called an "intersection" entity.

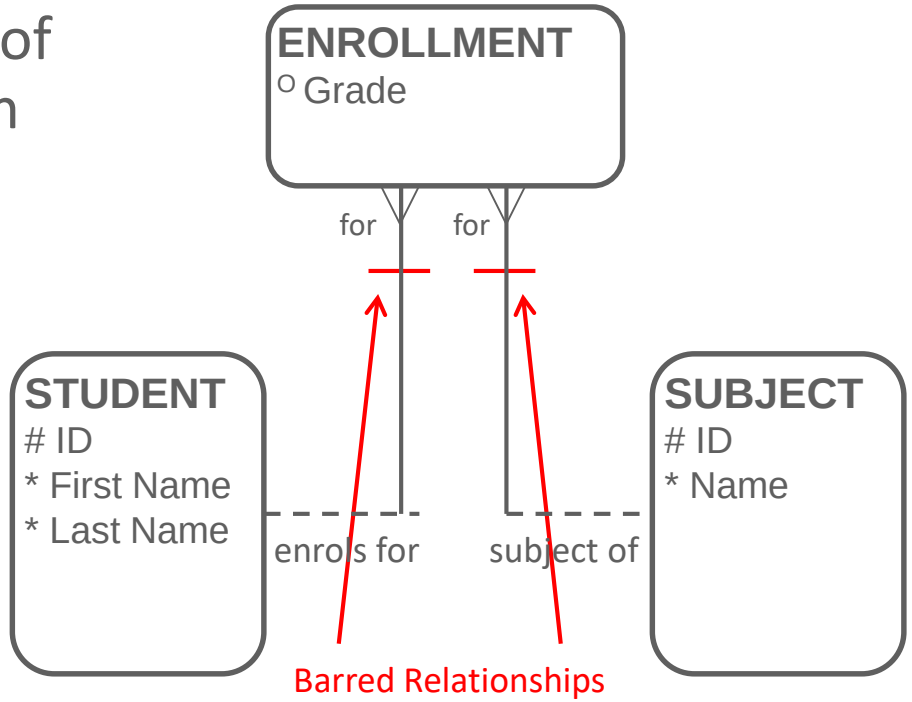
Intersection Entity

- An intersection entity – ENROLLMENT – has been added, including the "Grade" attribute.
- The original M:M relationship has become two 1:M relationships.
- What would be the UID of the intersection entity?



Barred Relationships

- The unique identifier (UID) of the intersection entity often comes from the originating relationships and is represented by the bars.
- In this case, the relationships from the originating entities to the intersection entity are called "barred" relationships.




ORACLE
ACADEMY