فوليتيكنيك بروني
POLITEKNIK BRUNEI

DATABASE DESIGN & ENTITY RELATIONSHIP MODEL

Lecturer: Jailani Abdul Rahman

Content in this lecture are taken from Lecture Notes by Dr John Colquhoun, Newcastle University, United Kingdom (2010).

Overview

- Database design process
- Entity-Relationship Diagram
- Conceptual database design (or conceptual modelling) using E-R Diagram

Database Design

- The database design process can be divided into six steps.

1. Requirement Analysis
    - What data is to be stored?
    - What applications must be built?
    - What operations are most frequent and subject to performance requirements?
2. Conceptual database design
    - High Level description of
        - Data to be stored
        - Constraints

Database Design (cont)

3. Logical Database Design
   - Choose a DBMS to implement our database design
   - Convert the conceptual database design into a database schema in the data model of the chosen DBMS
4. Schema Refinement
   - Analyse the collection of relations in our relational database
   - Identify the potential problems
   - Refine the schema

Database Design (cont)

5. Physical database design
   - Ensure that the design meets the performance requirement
   - Built index, clustering some tables etc.
   - Redesign parts of the database schema.
6. Application and security design
   - Write application programs
   - Identify data that can be accessible by certain types of users
   - Take steps to ensure that access rules are enforced

Conceptual Modeling

- It’s an important phase in designing a successful database application
- It comes after the requirement analysis
- Often carried out using Entity-Relationship model

Entity-Relationship (ER) Model

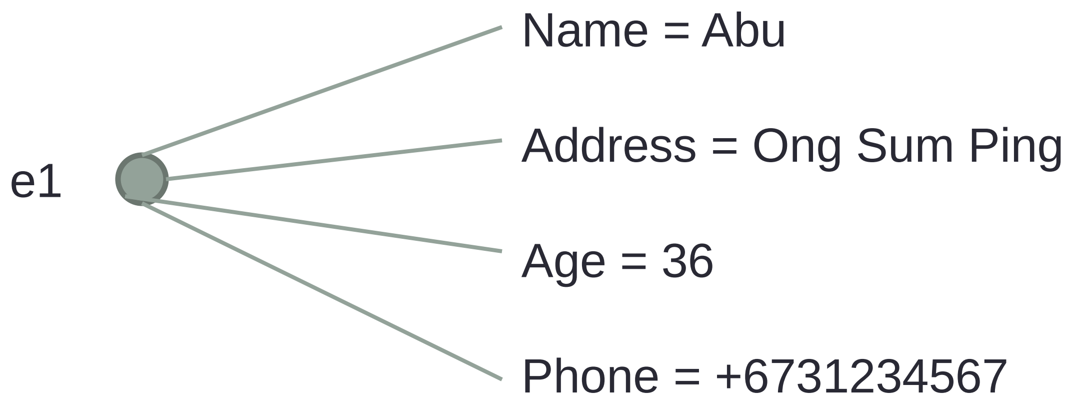
- A popular conceptual data model used to describe
  - Data to be stored, and
  - The constraints over the data

- E-R model views the real world as a collection of
  - Entities, and
  - Relationships among entities

Entities and attributes

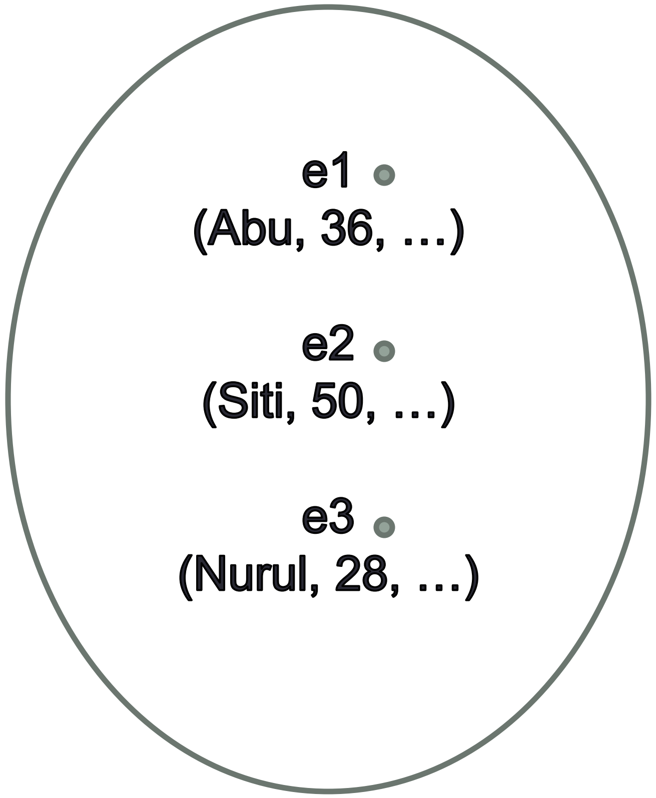
- An **entity** is a real-world object that is distinguishable from other objects.
  - E.g., a classroom, a teacher, a department

- An entity is represented by a set of **attributes**, whose values are used to distinguish one entity from another of the same type.



Entities and attributes (cont)

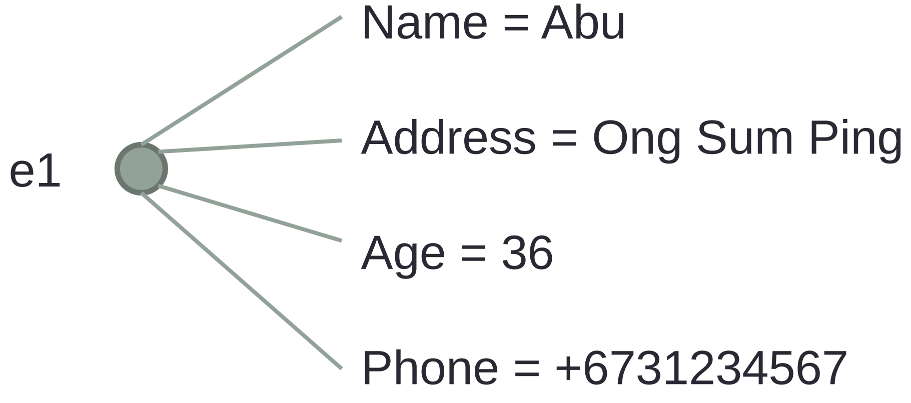
- An **entity set** is a set of entities of the same type.
  - E.g. an entity set for the entity type named EMPLOYEE



Entities and attributes (cont)

- All entities in a given entity set have the same attributes (attribute values may be different)

- Example:
  - Employee = (name, address, age, phone)



Employee 1

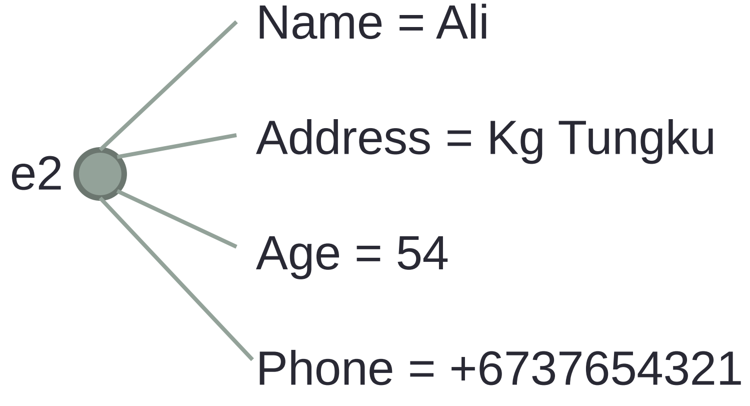
e1

Name = Abu

Address = Ong Sum Ping

Age = 36

Phone = +6731234567



Employee 2

e2

Name = Ali

Address = Kg Tungku

Age = 54

Phone = +6737654321

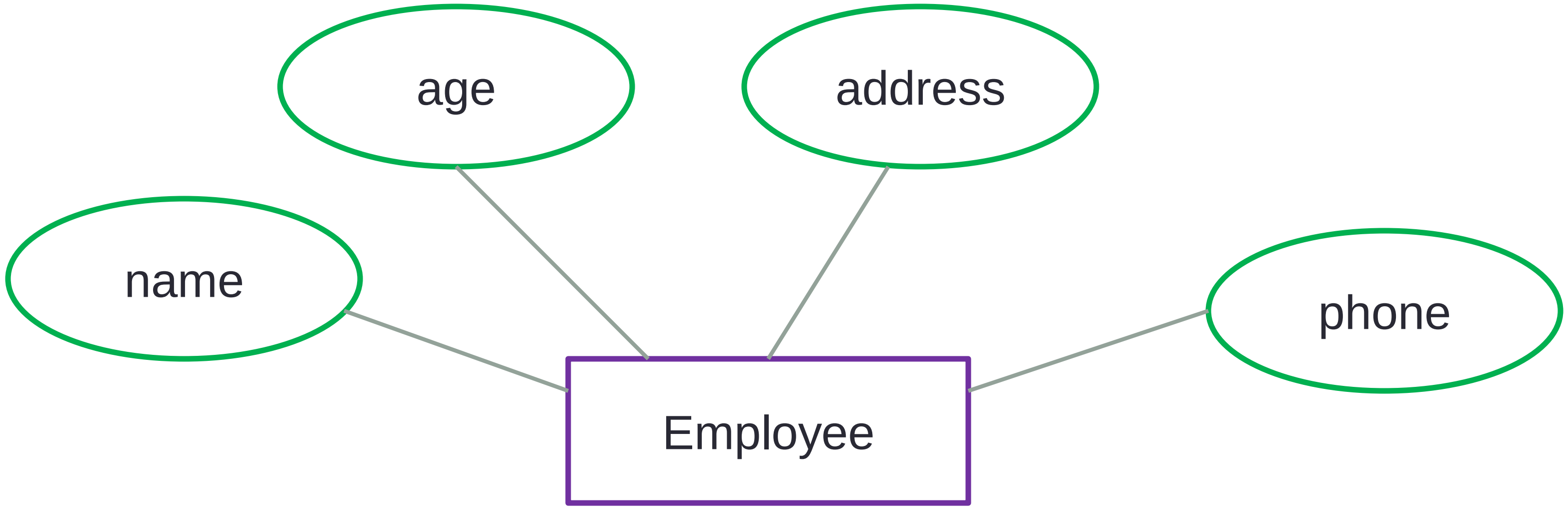
Entities and attributes (cont)

- For each attribute associated with an entity set, we must identify a domain of possible values.
  - Example
    - The domain associated with the attribute name might be the set of 20-character strings.
    - The domain associated with the attribute age can be an integer

- **Domain of an attribute:** the set of permitted values for the attribute

E-R diagram for the entity set of Employee

- **Entity: represented by a rectangle**
- **Attribute: represented by an oval**



Keys

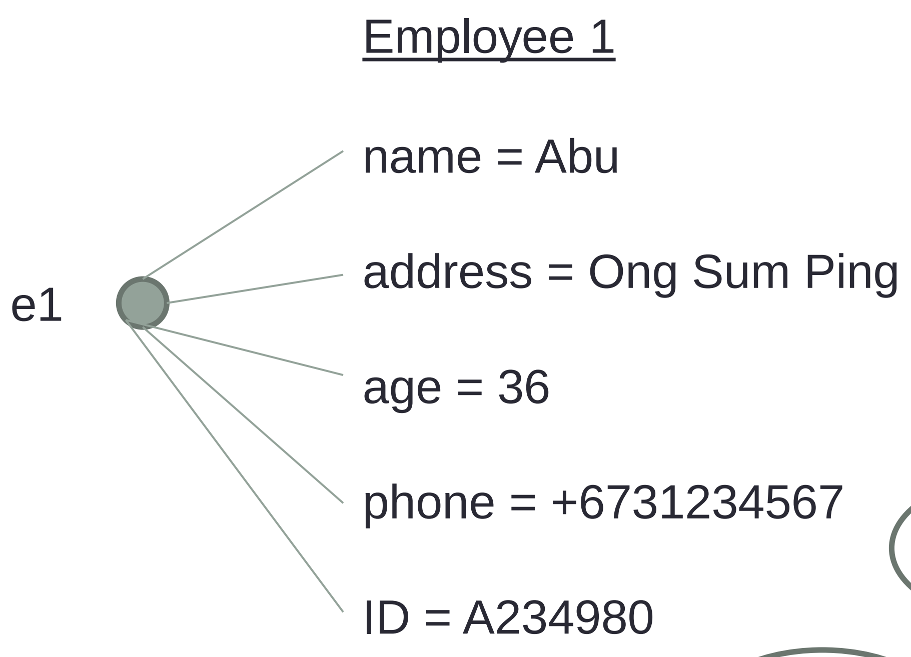
- Sometimes, the value of a **key** attribute can be used to identify each entity uniquely.

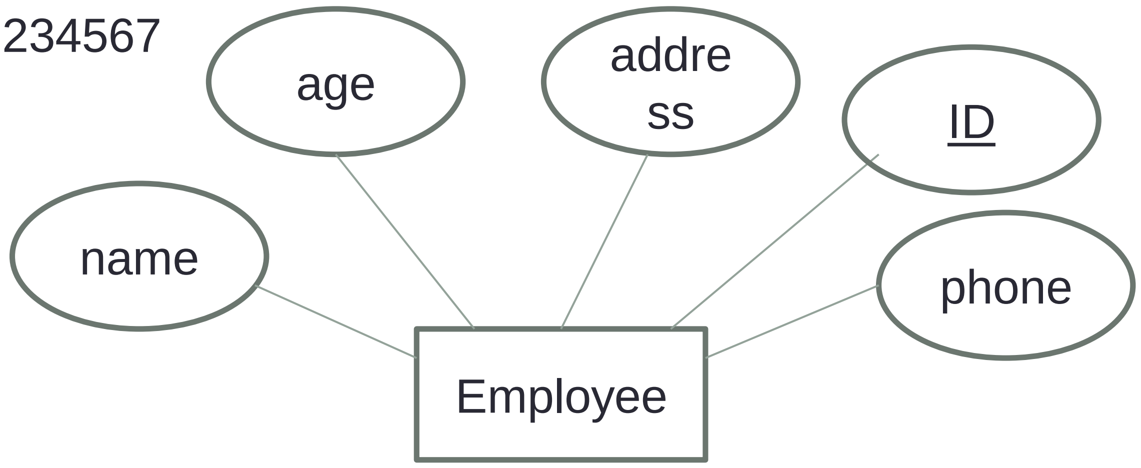
- How to decide a key?
  - Which attribute can be the key depends on real life possibilities **rather than on the current set of the data**

  - For example, in the previous set of two employees, the age can distinguish each employee.
    - However, we know that we may get a new employee with the same age as an existing employee.

Keys (cont)

- We may decide to add the attribute of **ID** as the key.



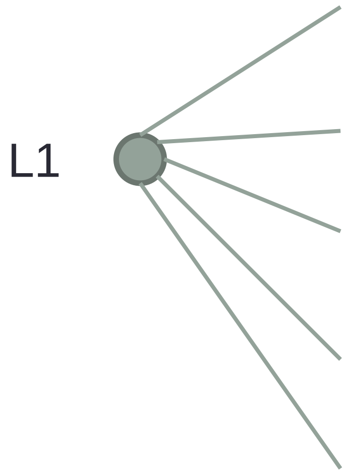


Definition

- A **key** can consist of more than one attribute

- A **key** is a **minimal** set of attributes whose values can uniquely identify an entity in the set.

A lecture L1



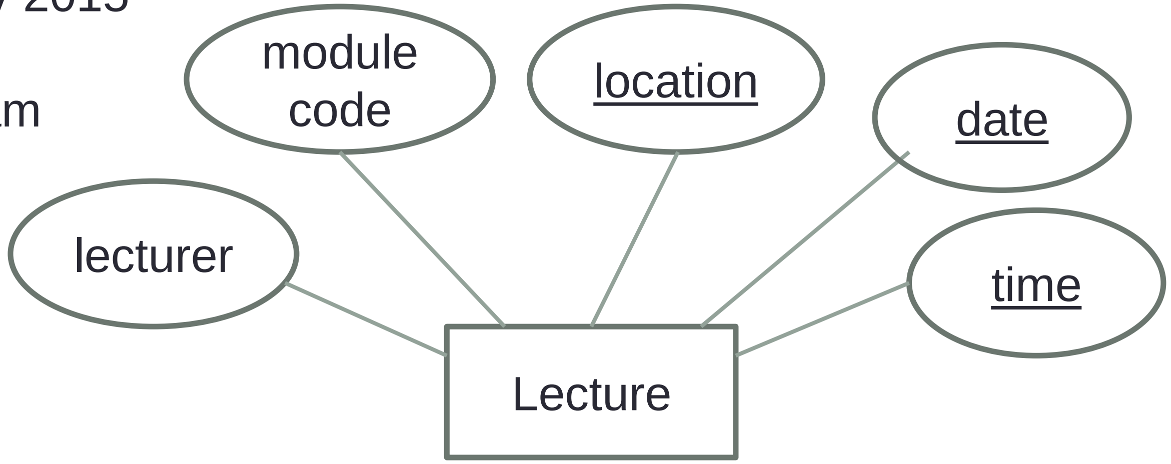
lecturer = Jailani

module code = DBMS101

location = OSP.PB.10.2L

date = 25 July 2015

time = 10.30am



- Suppose {location, date, time} is a key,

- Is {module code, location, date, time} a key?

Superkeys, Candidate Keys, Primary Keys

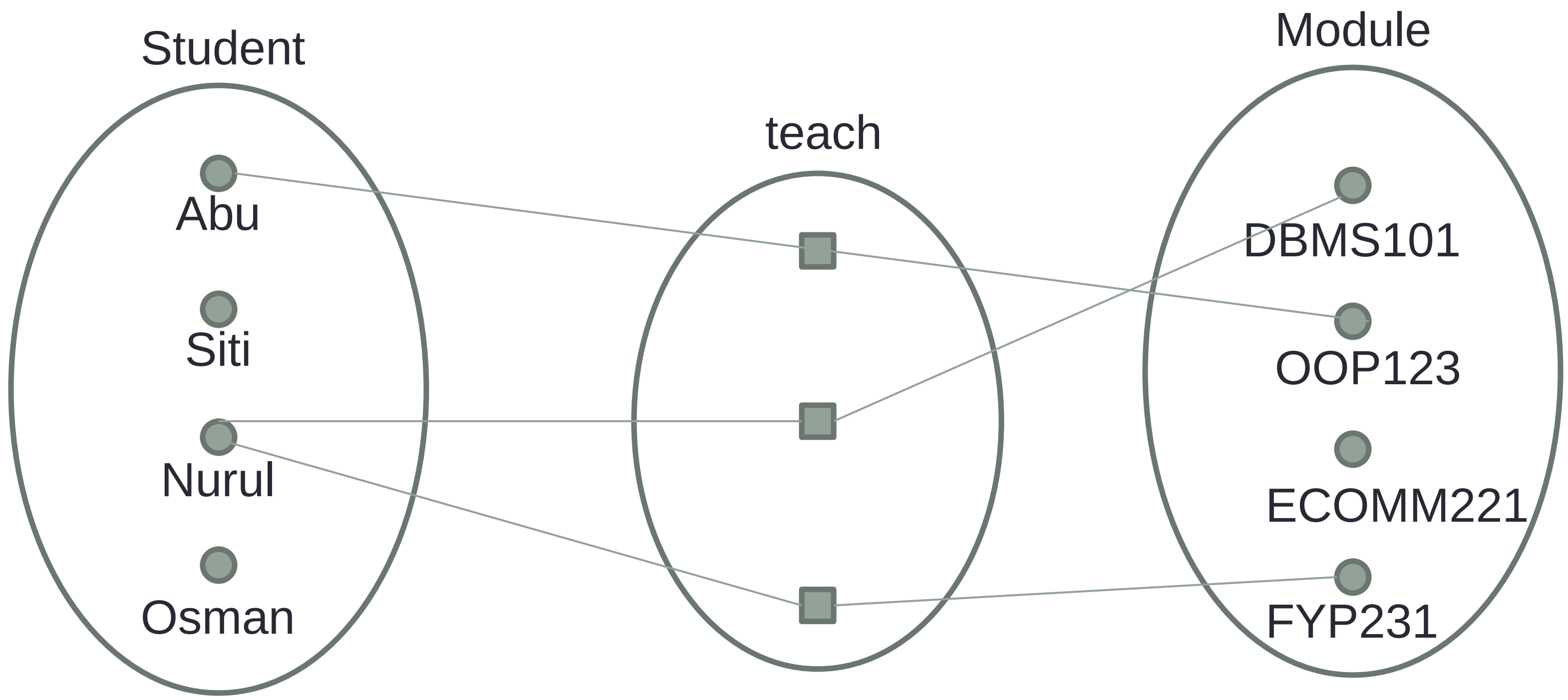
- A **superkey** is any set of attribute which can uniquely identify an entity.
  - Though {module code, location, date, time} is a **superkey**
  - {location, date, time} is a **simpler superkey**

- A **candidate key** is a set of keys that can be chosen as primary key.

- A **primary key** is a candidate key chosen to serve as the key for the entity set.

Relationship

- A **relationship** is an association among several entities.
  - Example:
    - Student = {Abu, Siti, Nurul, Osman}
    - Module = {DBMS101, OOP123, ECOMM221, FYP231}



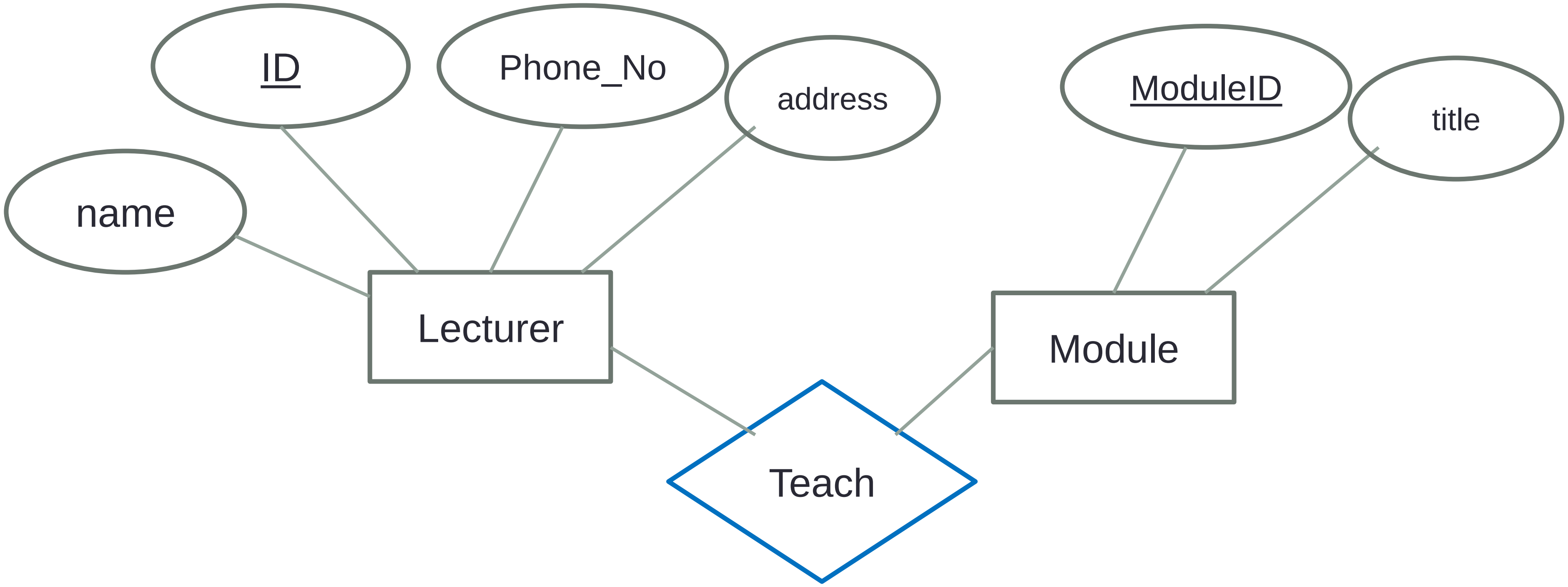
Relationship (cont)

- Relationship: teach
  - (Abu, OOP123), (Nurul, DBMS101), (Nurul, FYP231)

- As with entities, we may wish to collect a set of similar relationships into a relationship set
  - A **relationship set** is a set of relationships of the same type

E-R diagram

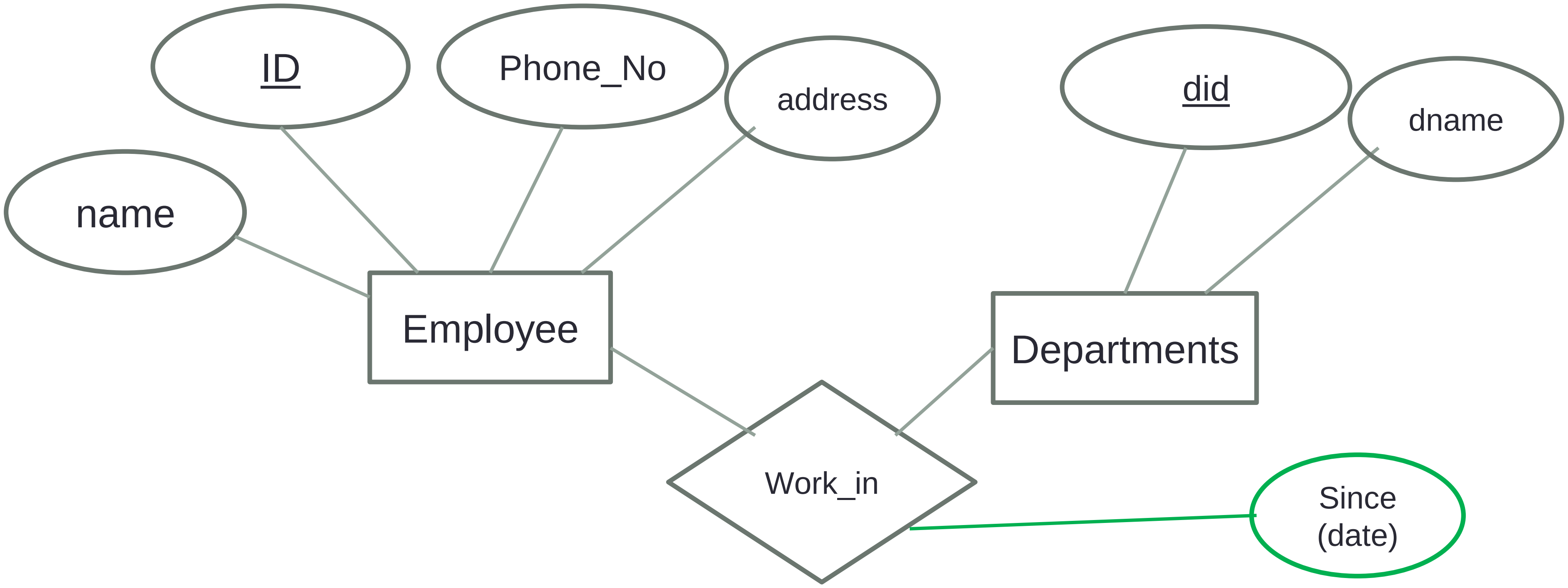
A **relationship** is represented by a **diamond**



Lecturer and **Module** are participating entities of **Teach** Relationship

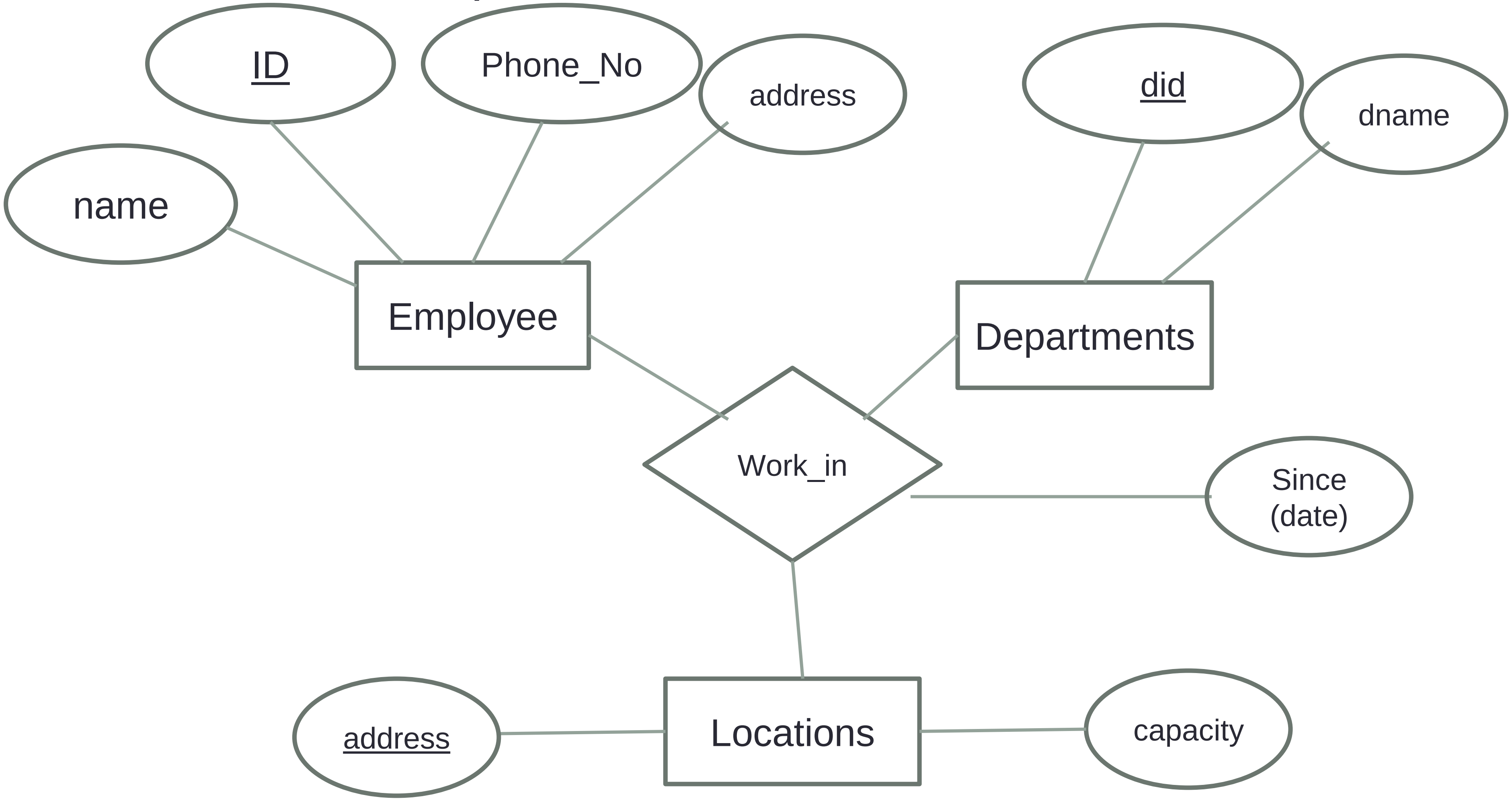
Attributes of a relationship set

- A **relationship** can also have descriptive **attributes**
- Descriptive attributes:
  - Record information about the relationship
  - Rather than about any one of the participating entities



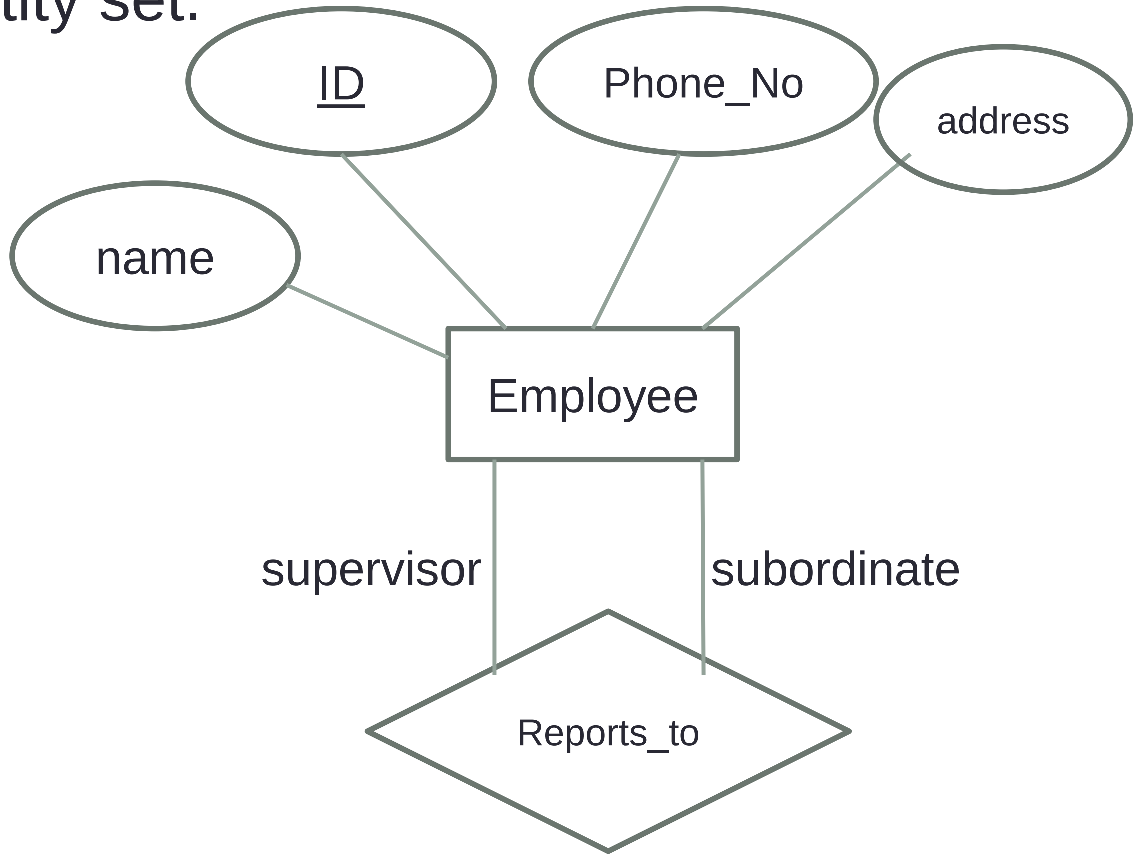
Ternary Relationship

- It is a relationship that involve **more than two entities**.



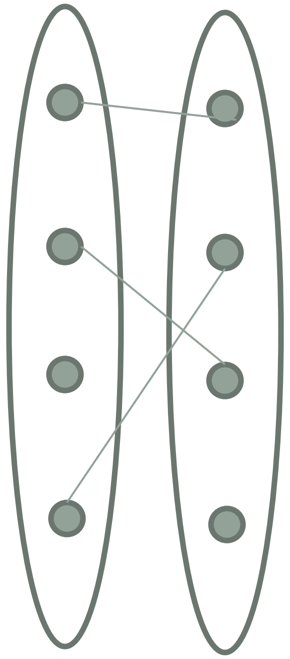
Recursive Relationship

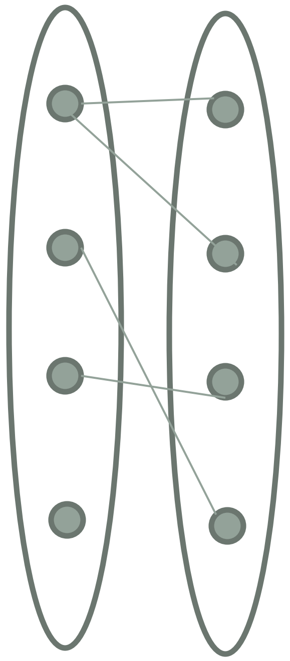
- Entity sets of a relationship need not be distinct.
- Sometimes a relationship might involve two entities in the same entity set.

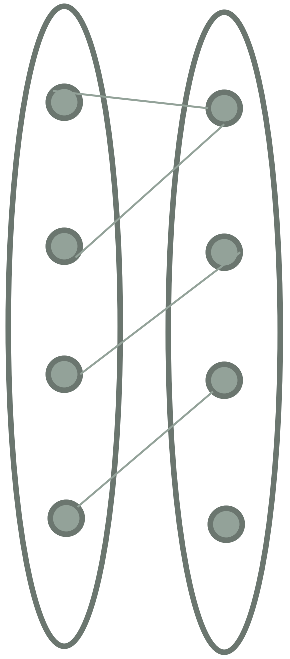


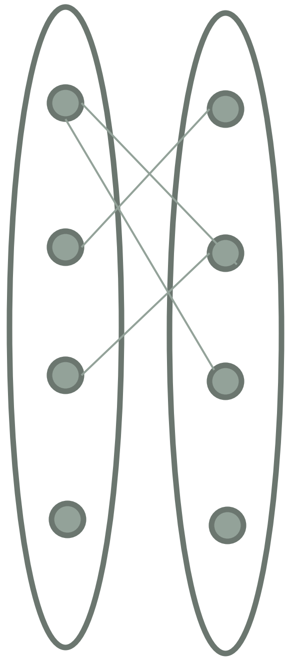
Mapping Constraints (Key Constraints)

- The mapping of the relationship can be classified into the following cases


1-to-1

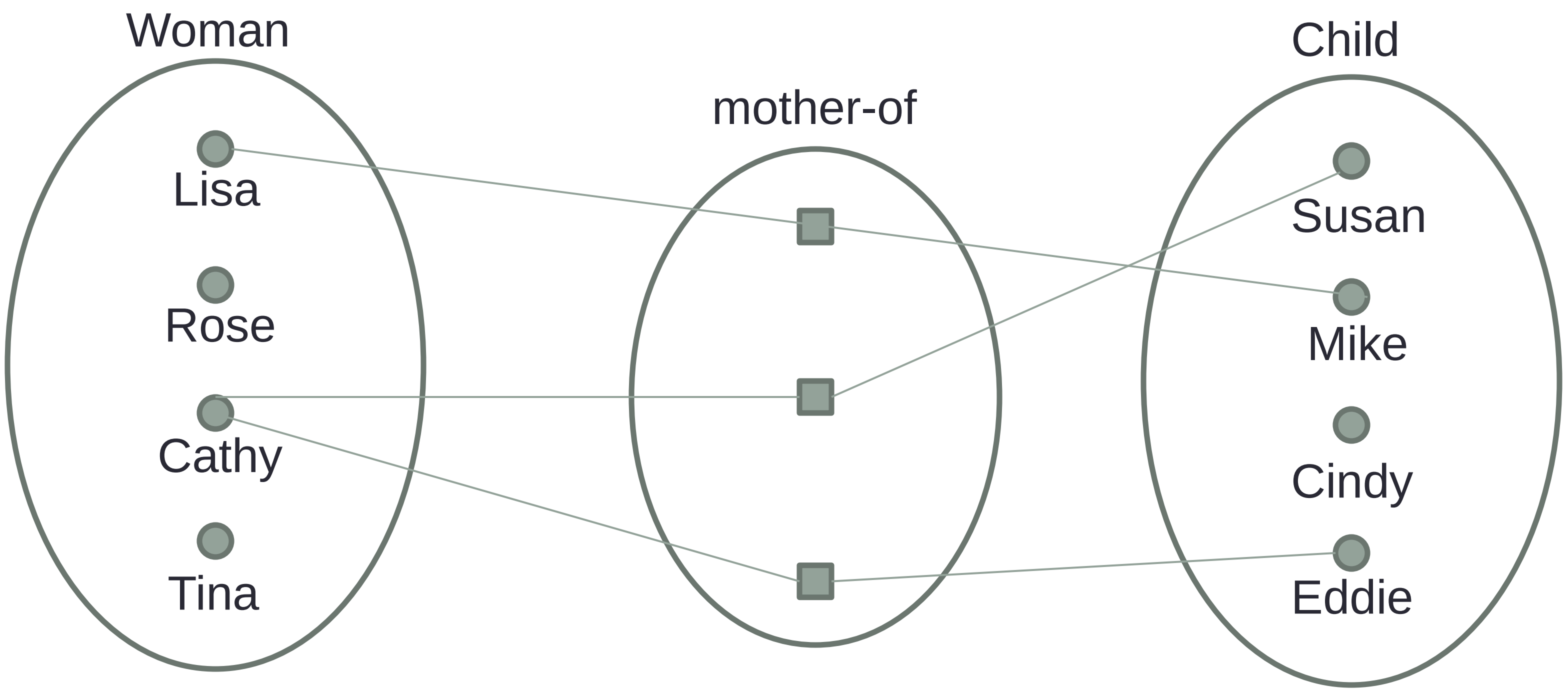

1-to-Many


Many-to-1


Many-to-Many

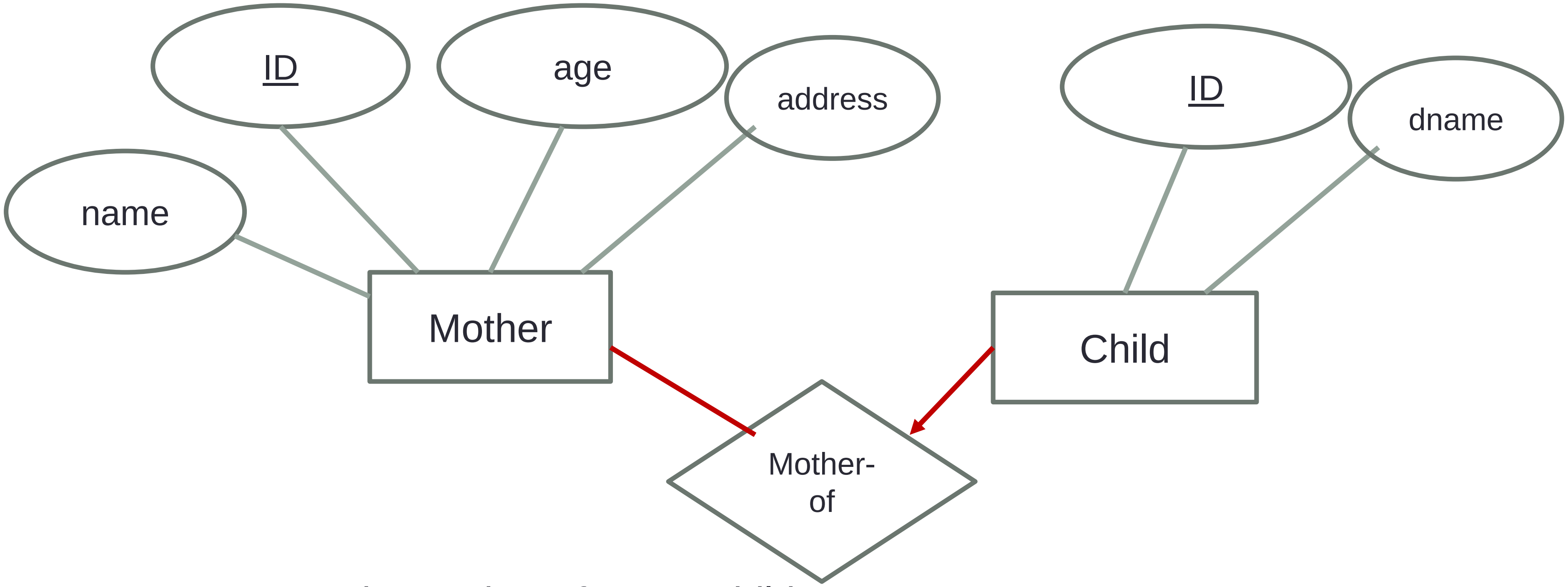
One-to-many Relationship

- **One-to-many** constraint from A to B: an entity in B can be associated with at most one entity in A
  - Example: mother-of relation
    - One woman can be mother of many child.
    - One child can only have one mother.



One-to-many Relationship (cont)

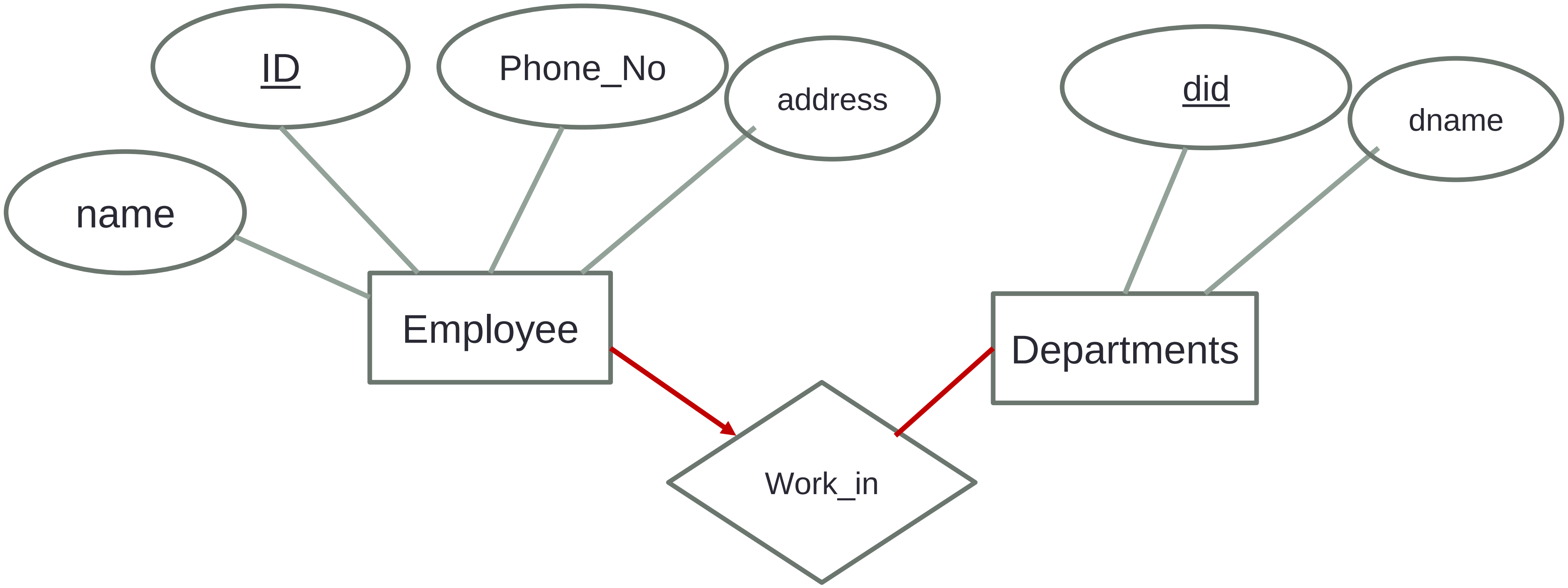
- We say that Child has a **key constraint** in the mother-of relationship set.
- This restriction can be indicated by an arrow in the E-R diagram.



- **One** woman can be mother of **many** child.
- **One** child can only have **one** mother.

Many-to-one Relationship

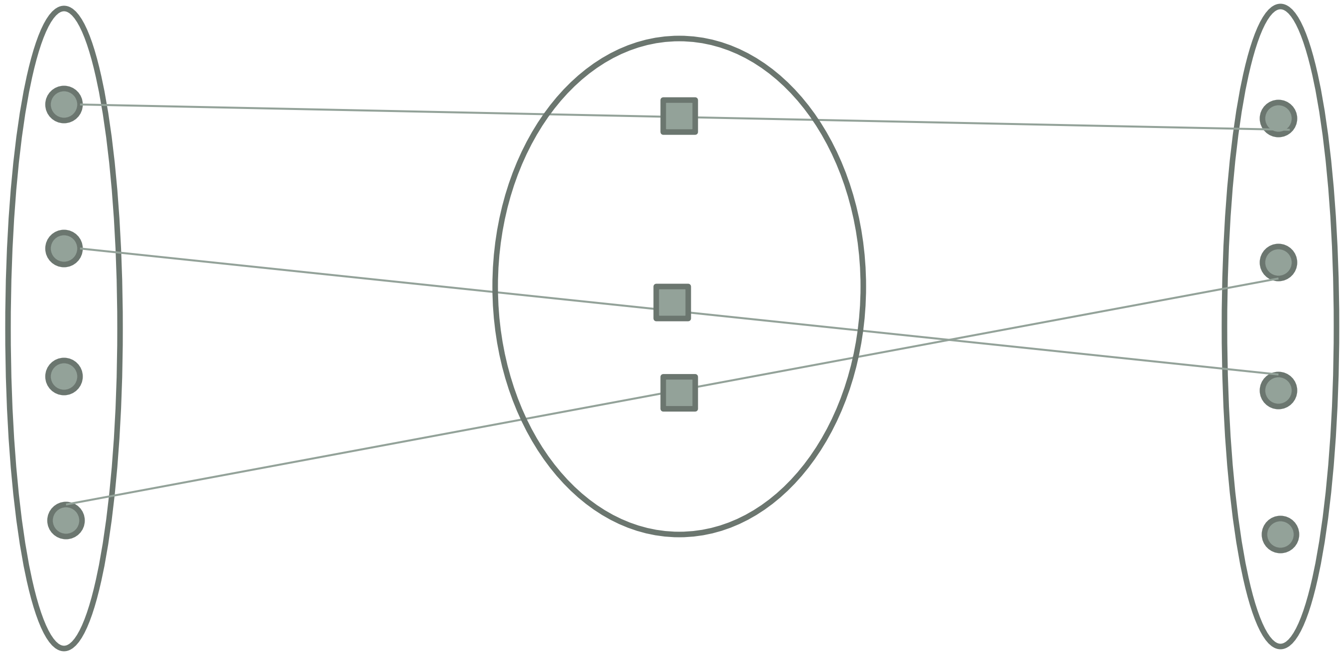
- Similar to concept as one-to-many. But the way we represent it is different.



- **One** employee can only work in **one** department
- **One** departments can have **many** employees.

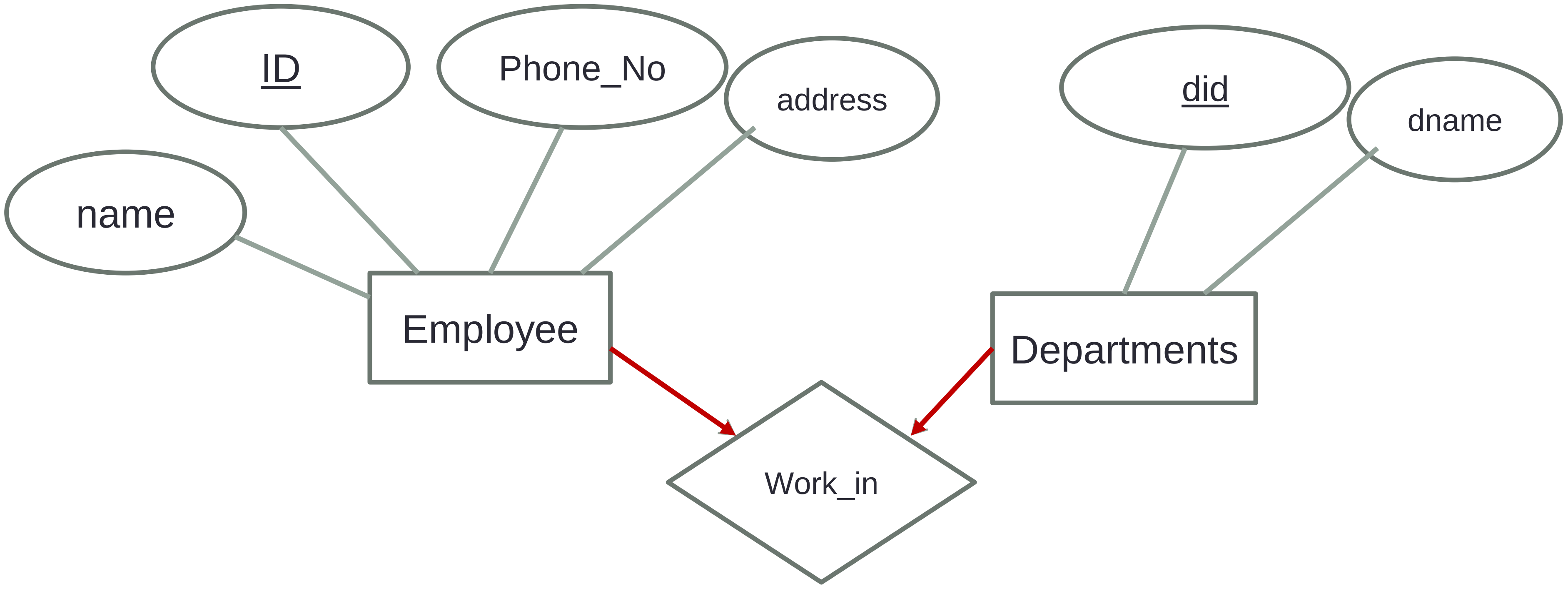
One-to-one Relationship

- If the relationship between A and B satisfies the one-to-one mapping constraint from A to B, then an entity in A is related to at most one entity in B, and an entity in B is related to at most one entity in A.



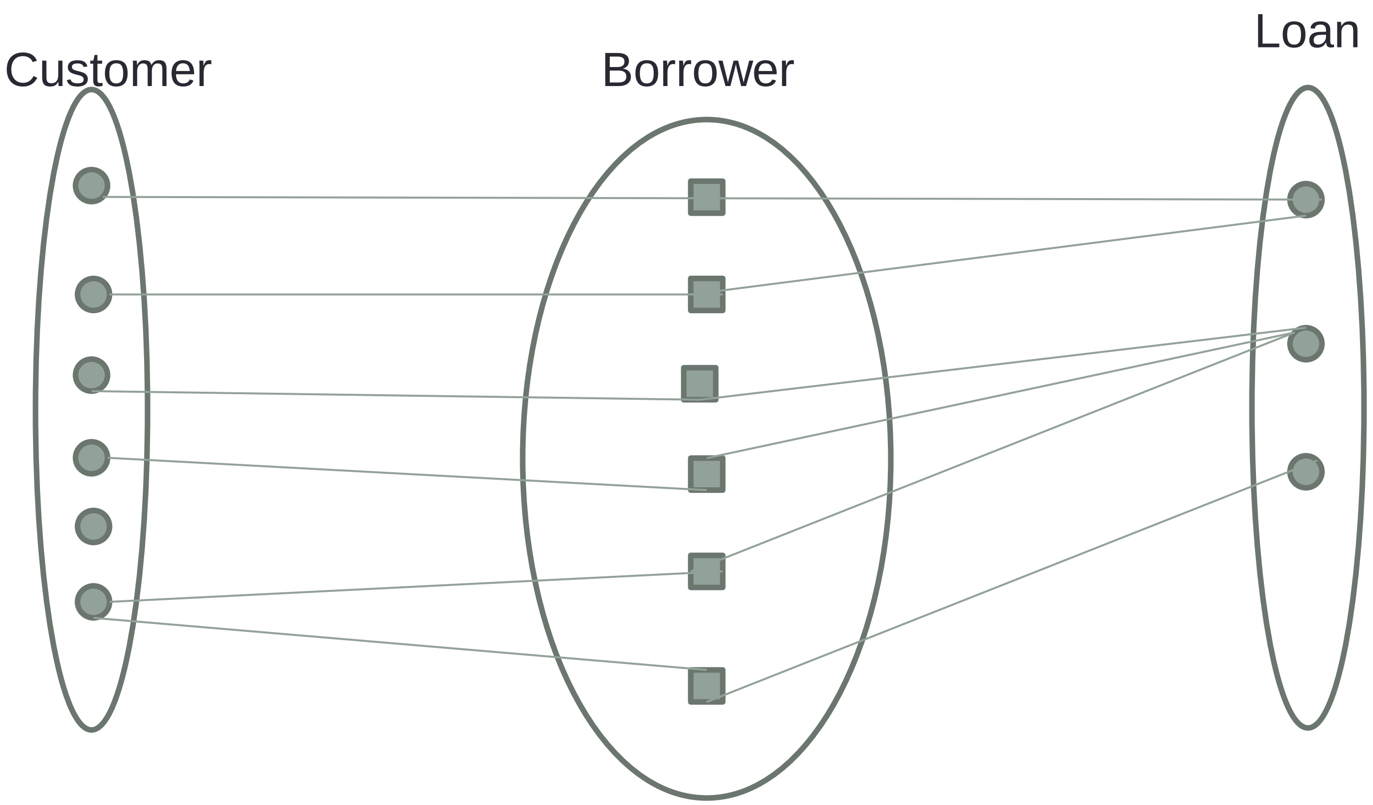
One-to-one Relationship (cont.)

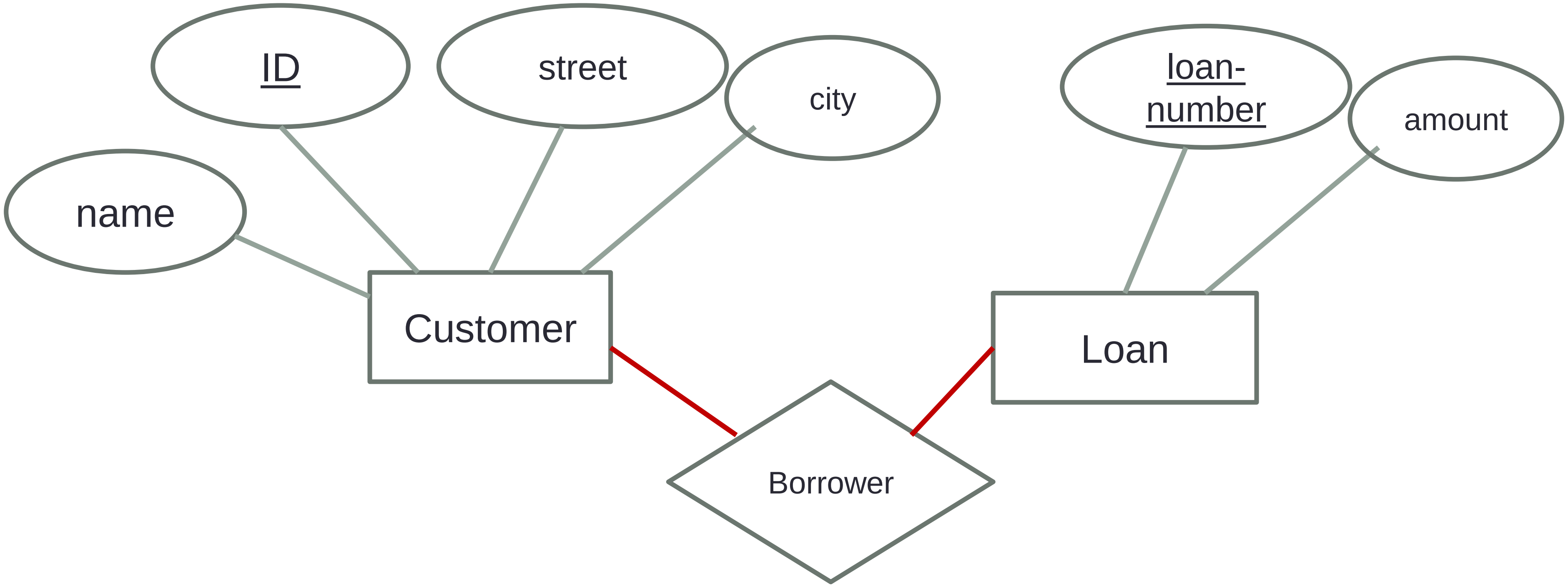
- **One** employee can only work in **one** department.
- **One** department can only have **one** employee.



Many-to-many Relationship

- An entity in A is associated with any number of entities in B, and an entity in B is associated with any number of entities in A





- **One** customer can borrowed **many** types of loan.
- **One** type of loan can be borrowed by **many** customer.

➢ Many-to-many is the most general relationship. It fact, it means that there is no restrictions

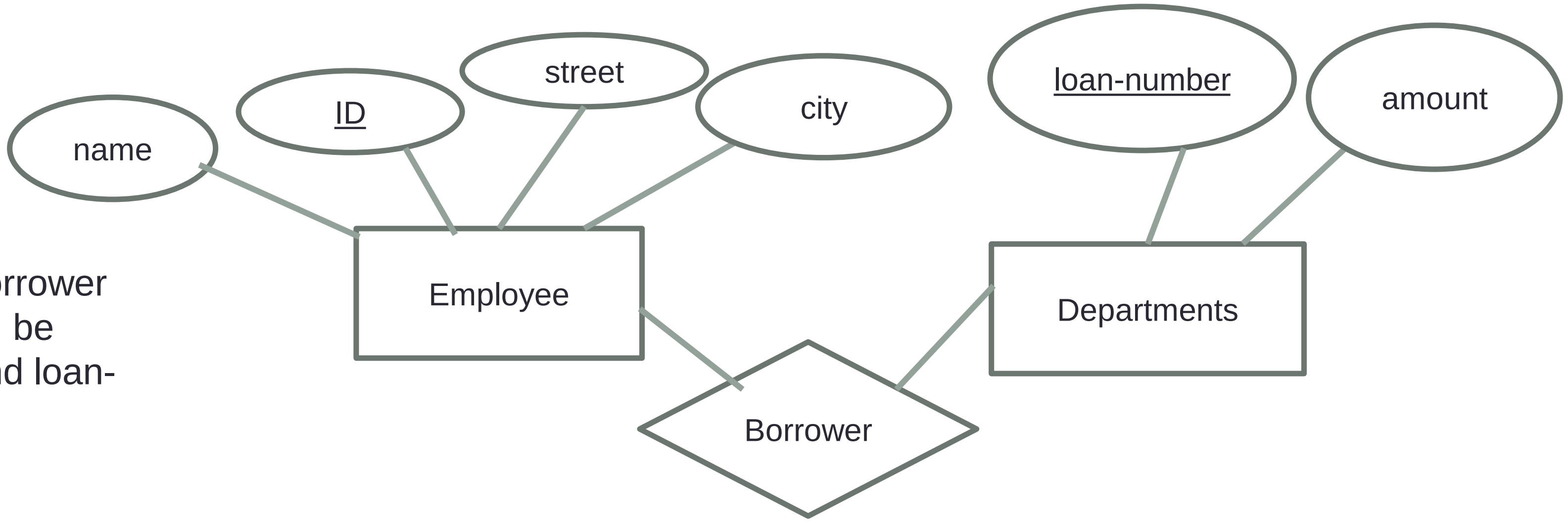
Reminder:

- Key Constraints (Mapping Constraints) are real life facts and they should determined by real-life scenarios.

- Database design has to enforce these constraints in order to avoid erroneous data.

Keys for a relationship set

- As for entities, the concept of keys is also used to identify a relationship.
- There are different scenarios:
  - For a relationship among associated entities with no mapping constraint, the primary key is normally the union of the primary keys of the associated entities.



In this case, Borrower primary key will be EmployeeID and loan-number

Keys for a relationship set (cont.)

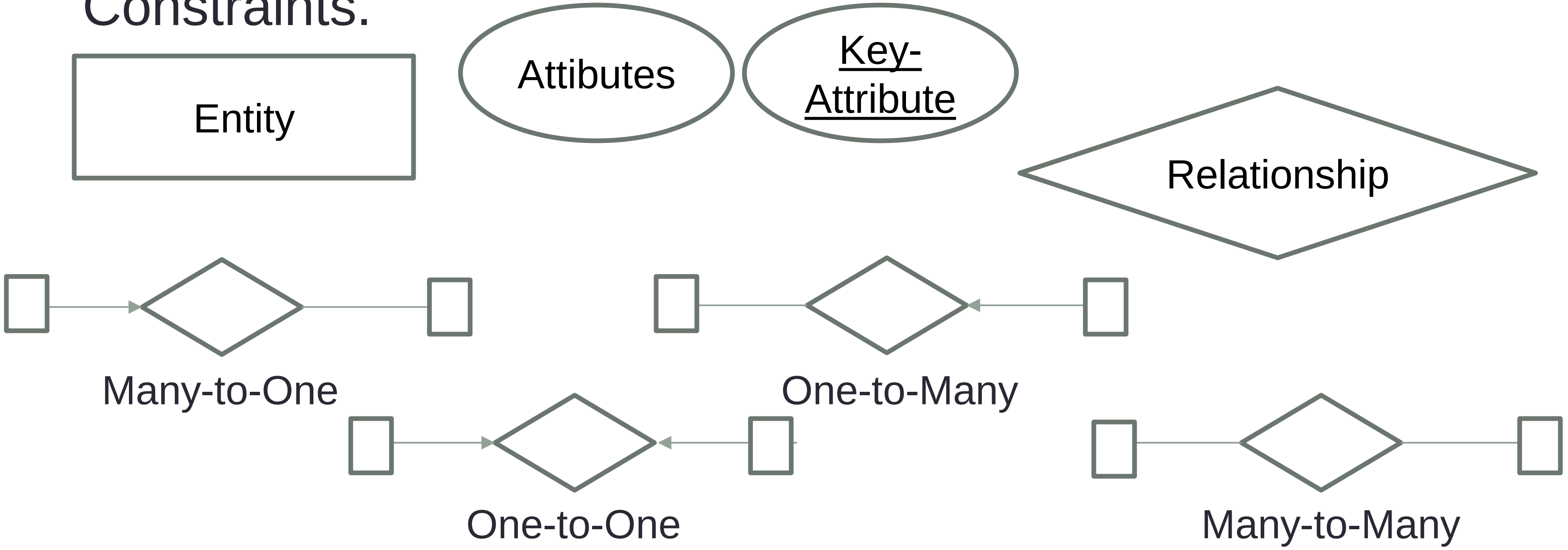
- If an entity set E has a key constraint in a relationship set R, each entity in an instance of E appears in at most one relationship in the instance of R.
  - Thus, an entity in E can uniquely identify a relationship in R.
  - ➢The key of E can be used as the key for R
- Example: child-mother, a many-to-one relationship, where entity Child has a key constraint.
  - ➢Child can be the primary key in the relationship set.

Keys for a relationship set (cont.)

- For an one-to-one relationship between two entity sets E and F, key(E) and key(F) are both keys for the relationship set.

Exercise

- Create E-R Diagram for these two scenarios.
  - University Timetabling Database
  - University Human Resource Database
- Identify the entities, attributes, relationships and Mapping Constraints.



Note:

- When designing a Database, if there are not stated in the requirement and an organisation doesn’t know what they want for that certain problem, you have to make **assumptions**.