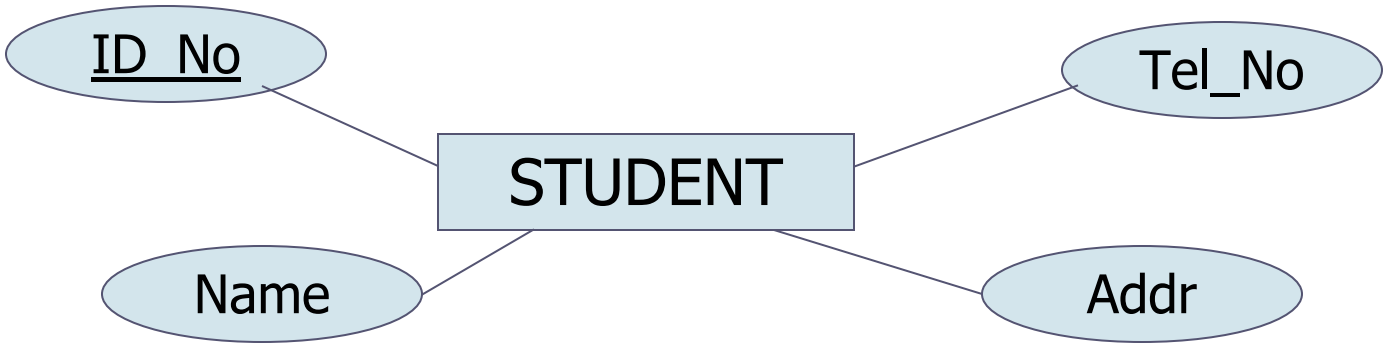
Transforming ERD into Relations

Step 1: Representing Entities

- Formally, for each entity type E in the ERD, create a relation R that includes all the attributes of E.
- This means, you need to
  - Create a 2-dimensional table whose number of columns is equal to the number of attributes in your ERD
  - Give the table the name, written in capital letters, that you gave the entity in the ERD
  - Include all the names of the attributes in the table that you created
    - If you remember one of the properties of a relation, **the order of the attributes is not important**

Step 1: Representing Entities

- Therefore, in the following ERD, STUDENT



Step 1: Representing Entities

- the table could look like this:

STUDENT

ID_No	Name	Addr	Tel_No
05-10001	dela Cruz, Juan	Dumaguete City	225-2233

- or like this:

STUDENT

Addr	Name	ID_No	Tel_No
Dumaguete City	dela Cruz, Juan	05-10001	225-2233

Step 2: Representing Relationships

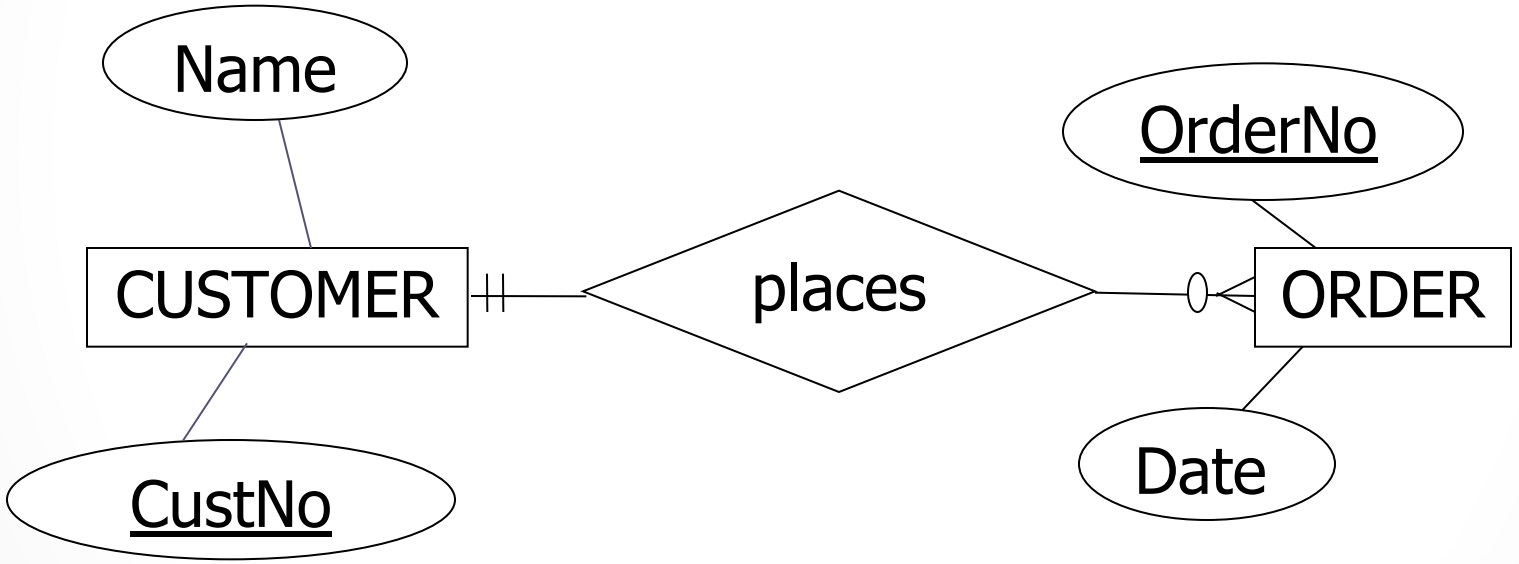
- Representing a relationship in a conceptual ERD to its equivalent construct in the relational logical model depends on both the degree of the relationship and its cardinality

Representing Binary 1:N Relationships

- This is the most common type of relationship and the easiest to represent
- Add the primary key of the entity on the one-side of the relationship as the foreign key in the relation that is on the many-side of the relationship

Representing Binary 1:N Relationships

- For example:



Representing Binary 1:N Relationships

- The relations will be as follows:

CUSTOMER

CustNo	Name
11001	Bonifacio, Andres
11002	Mabini, Apolinario
…	…

ORDER

OrderNo	Date	CustNo
1001	3/15/2009	11002
1002	3/17/2009	11002
1003	4/20/2009	11001
…	…	…

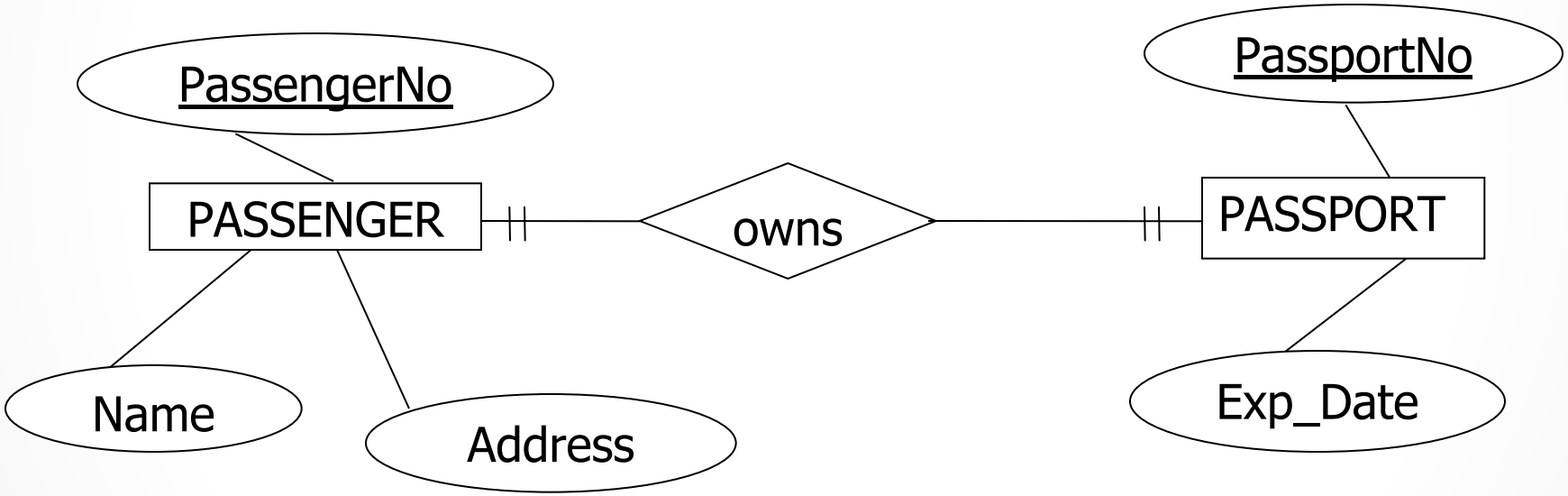
Notice that CustNo has a dashed line to indicate that it is a foreign key.

Representing Binary 1:1 Relationships

- If two entities are maintaining a one-to-one relationship, and they can share the same primary key, these may be resolved by combining the entities into one
- This means that the attribute of one entity will be moved into the other and a common primary key will be established

Representing Binary 1:1 Relationships

- For example



Representing Binary 1:1 Relationships

- Merging the two entity types and the relationship into a single relation is only appropriate when the participation of the entity types are total (an instance of one entity type ensures the existence of an instance of the other entity type) and the entity types do not participate in any other relationships.

Representing Binary 1:1 Relationships

- The relation schema of the ERD:

PASSENGER (PassengerNo, Name, Address, PassportNo, Exp_Date)

OR

PASSPORT (PassportNo, Exp_Date, PassengerNo, Name, Address)

Representing Binary 1:1 Relationships

- If the above technique is not permissible (i.e., when at least one of the entities participates in more than one relationship), the 1:1 relationship can be represented as:
  - Adding the primary key of A as a foreign key of B

PASSENGER(PassengerNo, Name, Address)
and
PASSPORT(PassportNo, Exp_Date, PassengerNo)

Representing Binary 1:1 Relationships

- Adding the primary of B as a foreign key of A

PASSENGER(PassengerNo, Name, Address, PassportNo) and
PASSPORT(PassportNo, Exp_Date)

- Both representations

PASSENGER(PassengerNo, Name, Address, PassportNo) and
PASSPORT(PassportNo, Exp_Date, PassengerNo)

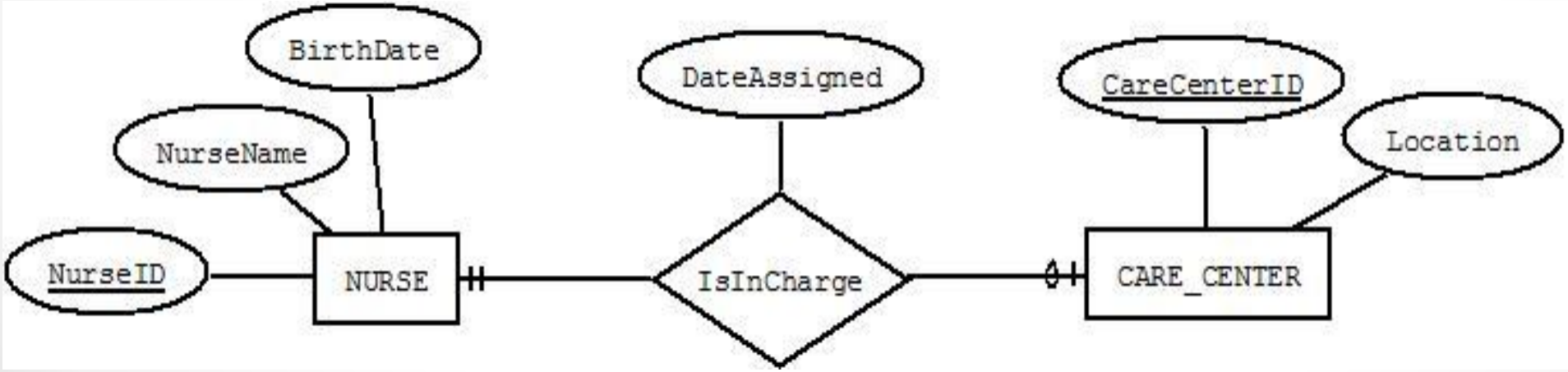
Representing Binary 1:1 Relationships

- The designer is left with the discretion as to which of the three is most appropriate.

Representing Binary 1:1 Relationships

(the relationship has an attribute)

- Any attributes in the relationship will be placed in the relation or table where the foreign key is placed (which is on the *optional* side)
- For example:



Representing Binary 1:1 Relationships

(the relationship has an attribute)

- Resulting relation schema:

NURSE (NurseID, NurseName, BirthDate)

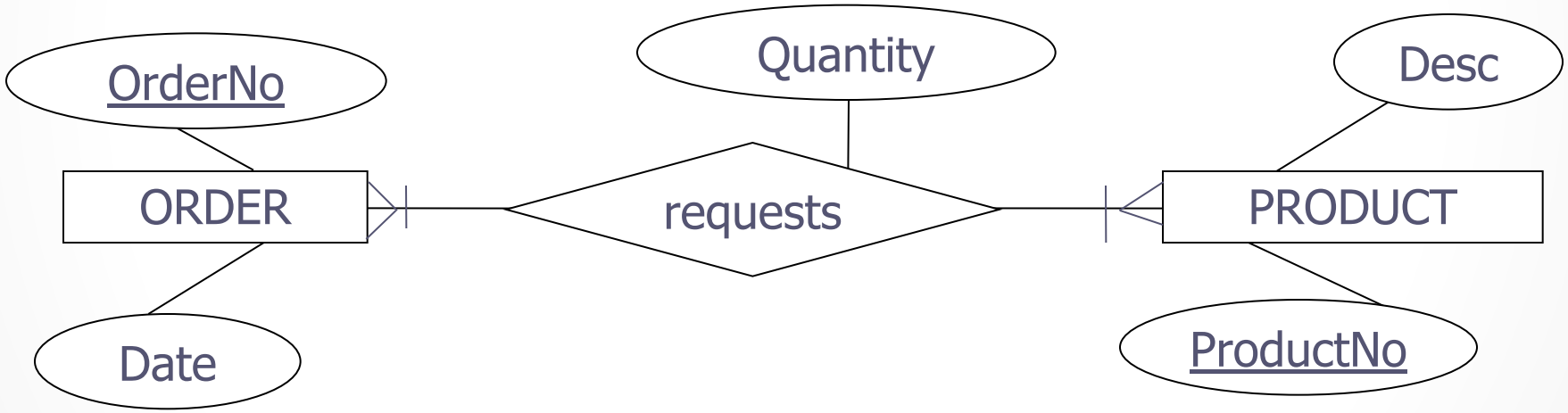
CARE_CENTER (CareCenterID, Location,
NurseInCharge, DateAssigned)

Representing Binary M:N Relationships

- Create a separate relation to represent the relationship
- Include as a foreign key, attributes in the newly created relation the primary keys of entities A and B. This combination will form the primary key of the new relation
- If the relationship has any attributes, include them in the new relation

Representing Binary M:N Relationships

- For example:



Representing Binary M:N Relationships

- Relation Schema of the ERD on the previous slide:

  ORDER (<u>OrderNo</u>, Date)

  PRODUCT (<u>ProductNo</u>, Desc)

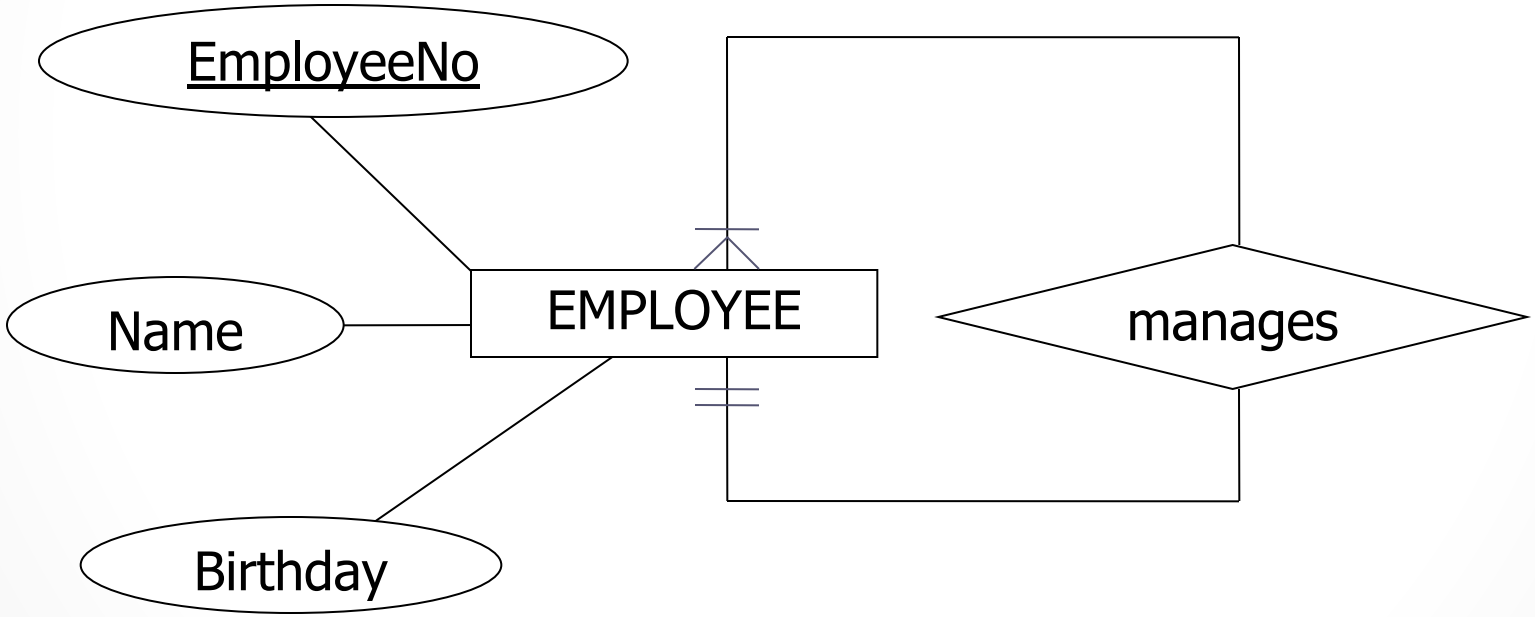
  ORDER_LIST (<u>OrderNo</u>, <u>ProductNo</u>, Quantity)

Representing Unary 1:N Relationships

- Represent a recursive relationship with cardinality ratio 1:N by simply adding a rename of the primary key with reference to the relationship

Representing Unary 1:N Relationships

- For example:



Representing Unary 1:N Relationships

- Relation Schema of the ERD:

  EMPLOYEE (EmployeeNo, Name, Birthday, ManagerID)

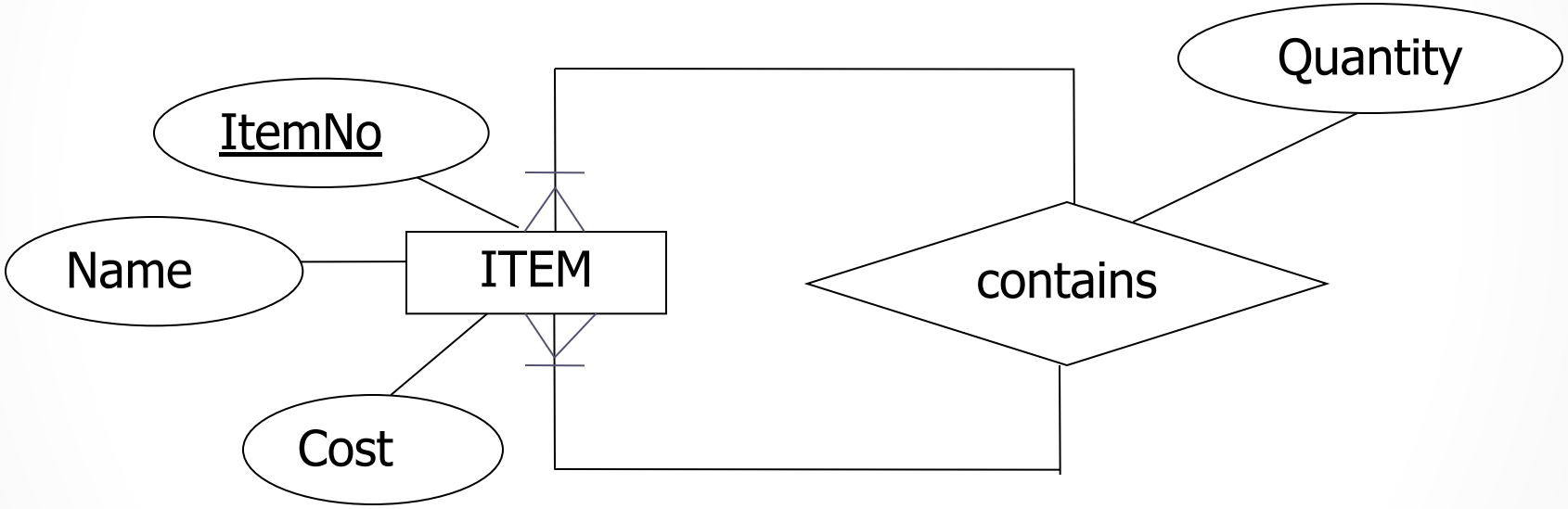
- the attribute ManagerID is actually just a *rename* of EmployeeNo
- the value of ManagerID is the Employee number of the employee's manager
- ManagerID is considered as a foreign key of the relation

Representing Unary M:N Relationships

- create a new relation to represent the relationship
- include as a foreign key attributes, the primary keys of the participating entity types
- the primary key of the new relation is composed of its set of foreign keys

Representing Unary M:N Relationships

- For example:



Representing Unary M:N Relationships

- Relation Schema:

  ITEM (ItemNo, Name, Cost)

  ITEM_BILL (ItemNo, PartNo, Quantity)

- PartNo is the rename of ItemNo

End of Presentation