MONASH University

MONASH
INFORMATION
TECHNOLOGY

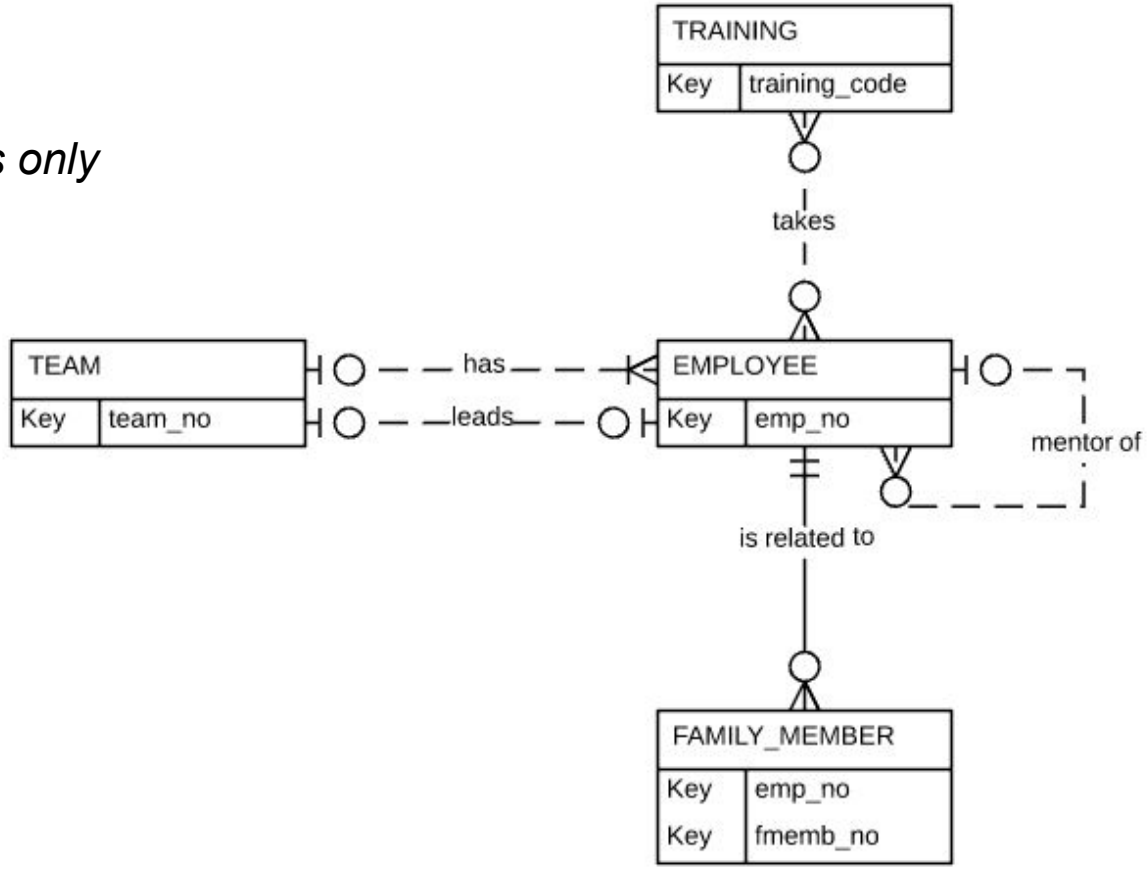
Database Design II: Logical Modelling

Lindsay Smith

GROUP OF EIGHT AUSTRALIA

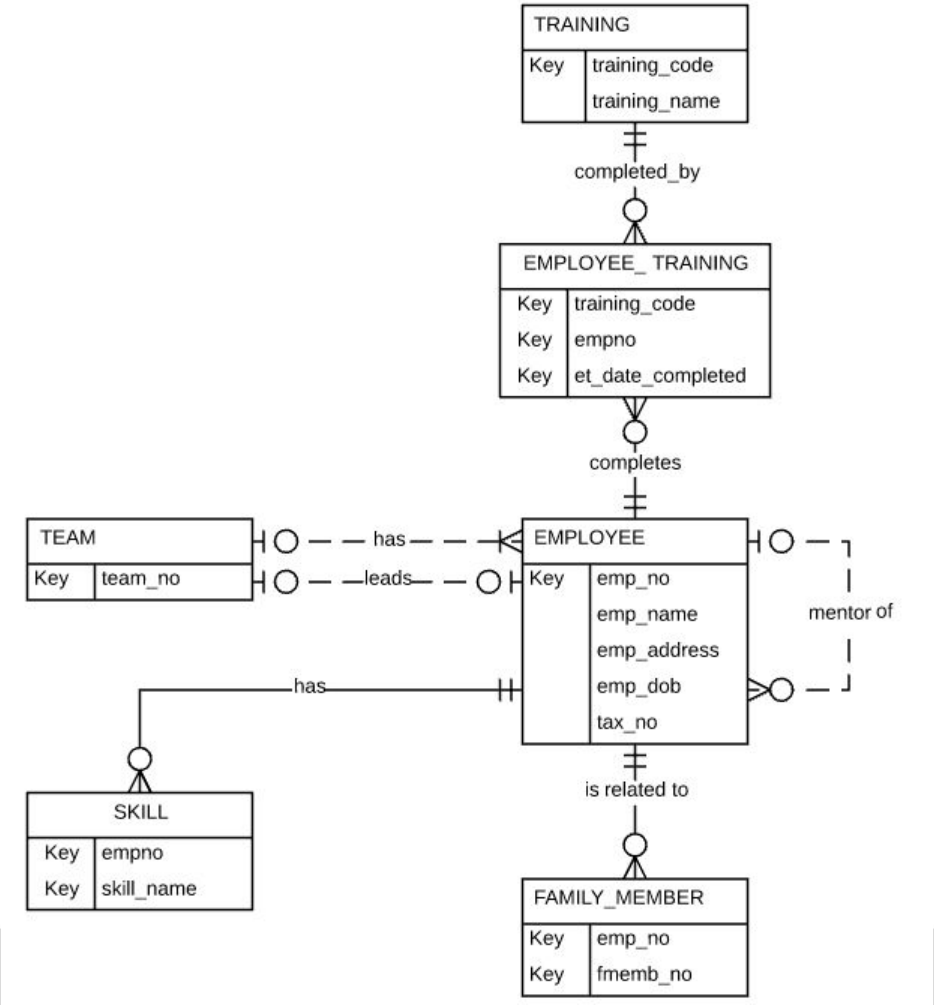
Revisit - Week 3 Conceptual Model

Keys only



Revisit - Week 3 Conceptual Model

All attributes



Summary of Terminologies at Different Levels

Conceptual	Logical	Physical
Entity	Relation	Table
Attribute	Attribute	Column
Instance	Tuple	Row
Identifier	Primary Key	Primary Key
Relationship	---	---
---	Foreign Key	Foreign Key

Properties of Relations

- Some properties to be considered:
  - Each relation has a unique name in the database.
  - Each row is unique - i.e. duplicate tuples are not allowed.
  - Each column has a (meaningful) name.
  - The order of attributes is immaterial.
  - The order of tuples is immaterial.
  - The entries are single-valued **(atomic)** - each cell contains a single entry.
    - **Multi-valued** and **composite attributes**???

Transforming ER diagrams into relations (mapping conceptual level to logical level)

- **The steps are:**
  - Map strong (regular) entities
  - Map weak entities
  - Map binary relationships
  - Map associative entities
  - Map unary relationships
  - Map ternary relationships
  - Map supertype/subtype relationships (is not part of this unit).

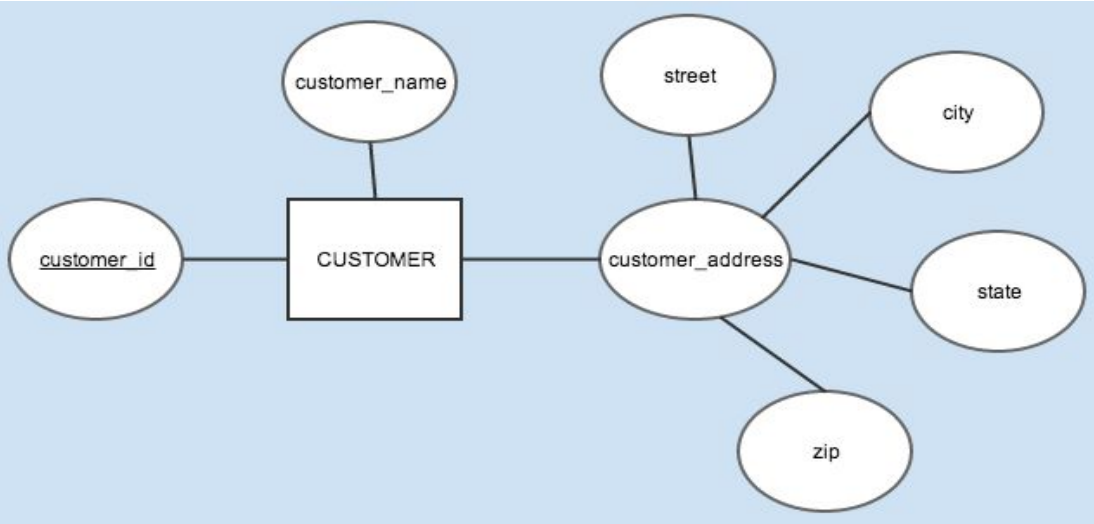
Q1. The relational model requires that each cell in a relation is single-valued (atomic). Considering this requirement, what construct in an ER diagram cannot be implemented directly (without further steps) in the relational model (logical level)?

a. Composite key.
b. Composite attribute.
c. Multi-valued attribute.
d. Dependent attribute.
e. More than one option is correct.

Map Regular Entities

- **Composite Attributes**
  - When the regular entity type contains a **composite attribute**, only the simple component attributes of the composite attribute are included in the new relation.
  - Compared to composite attributes, simple attributes not only improve data accessibility but also help in maintaining data quality

Mapping a Composite Attribute



CUSTOMER	
P *	cust_id
*	cust_name
*	cust_street
*	cust_city
*	cust_state
*	cust_zip

Monash Software Case Study

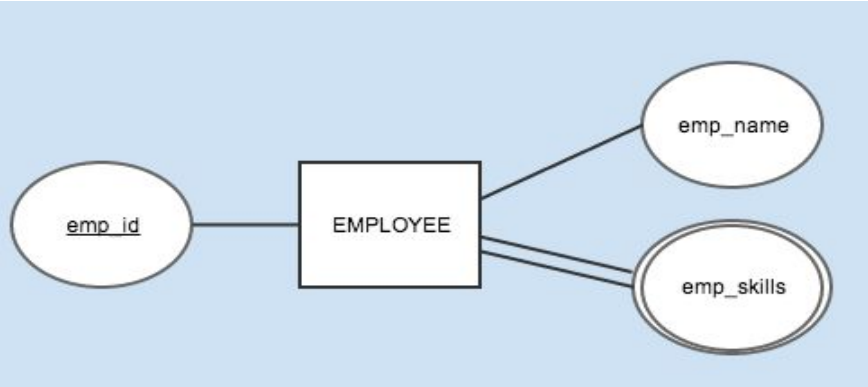
** = not null (must have value)*

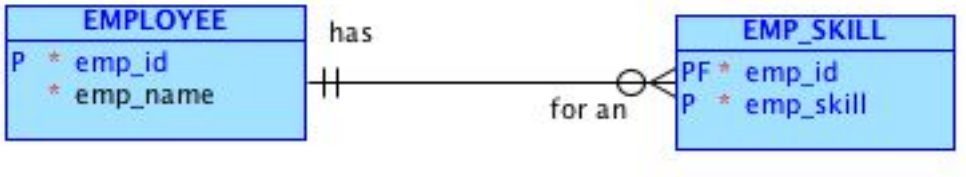
EMPLOYEE	
P *	emp_no
	emp_fname
	emp_lname
*	emp_street
*	emp_town
*	emp_pcode
*	emp_dob
	emp_taxno

Map Regular Entities

- **Multivalued Attribute**
  - When the regular entity type contains a **multivalued attribute,** two new relations are created.
  - The first relation contains all the attributes of the entity type except the multivalued attribute itself.
  - The second relation contains two attributes that form the PK. One of the attributes is the PK from the first relation, which becomes the FK in the second relation and the other is the multivalued attribute.
  - There can also be non key attributes in the second relation depending upon the data requirements.

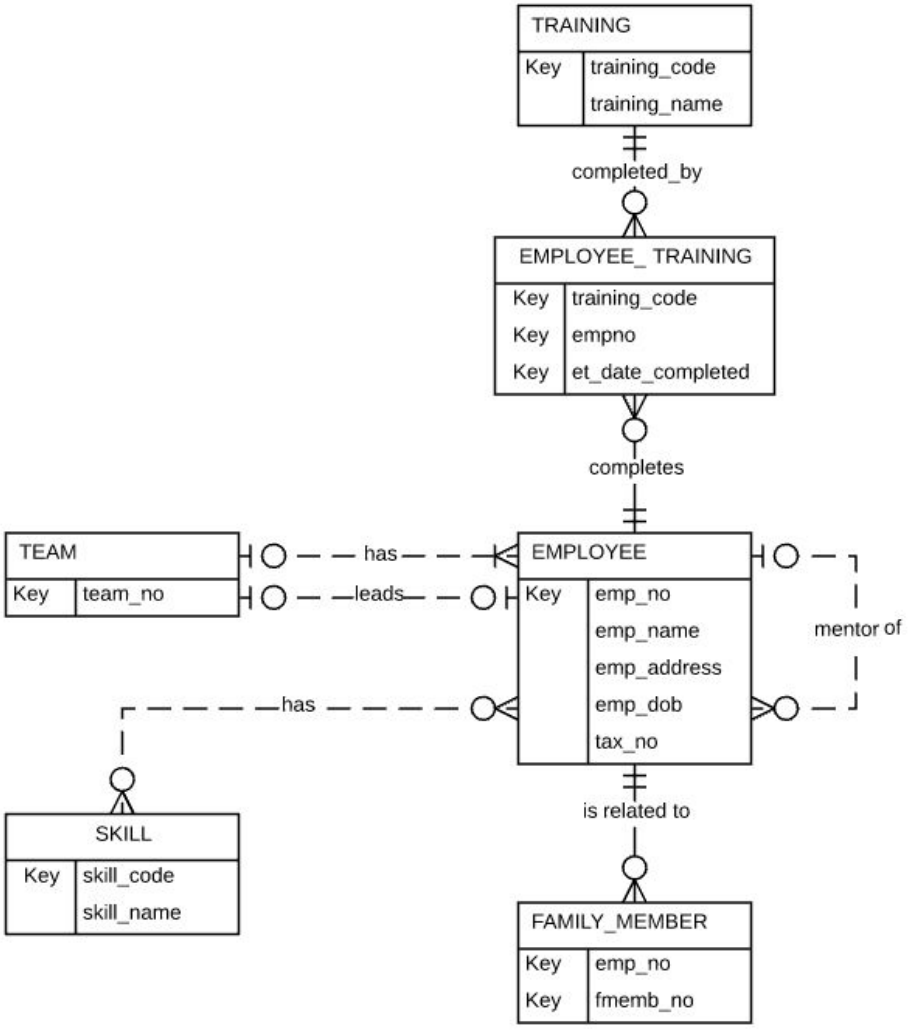
Mapping a Multi valued Attribute





Is there a better solution than the one shown above?

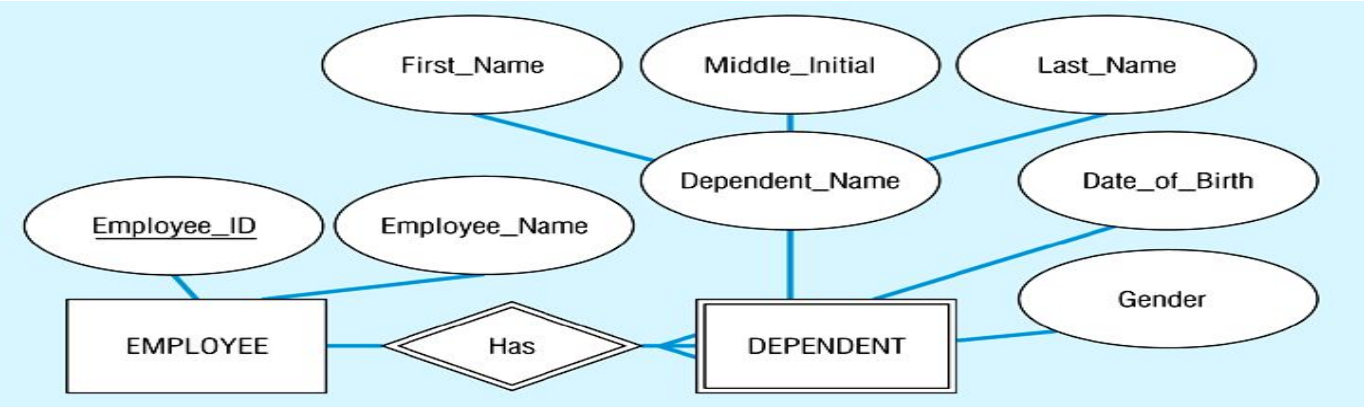
Revisit - Week 3 Conceptual Model - IMPROVED

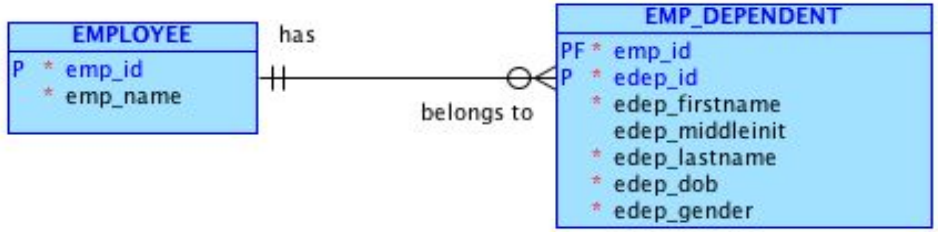


All attributes

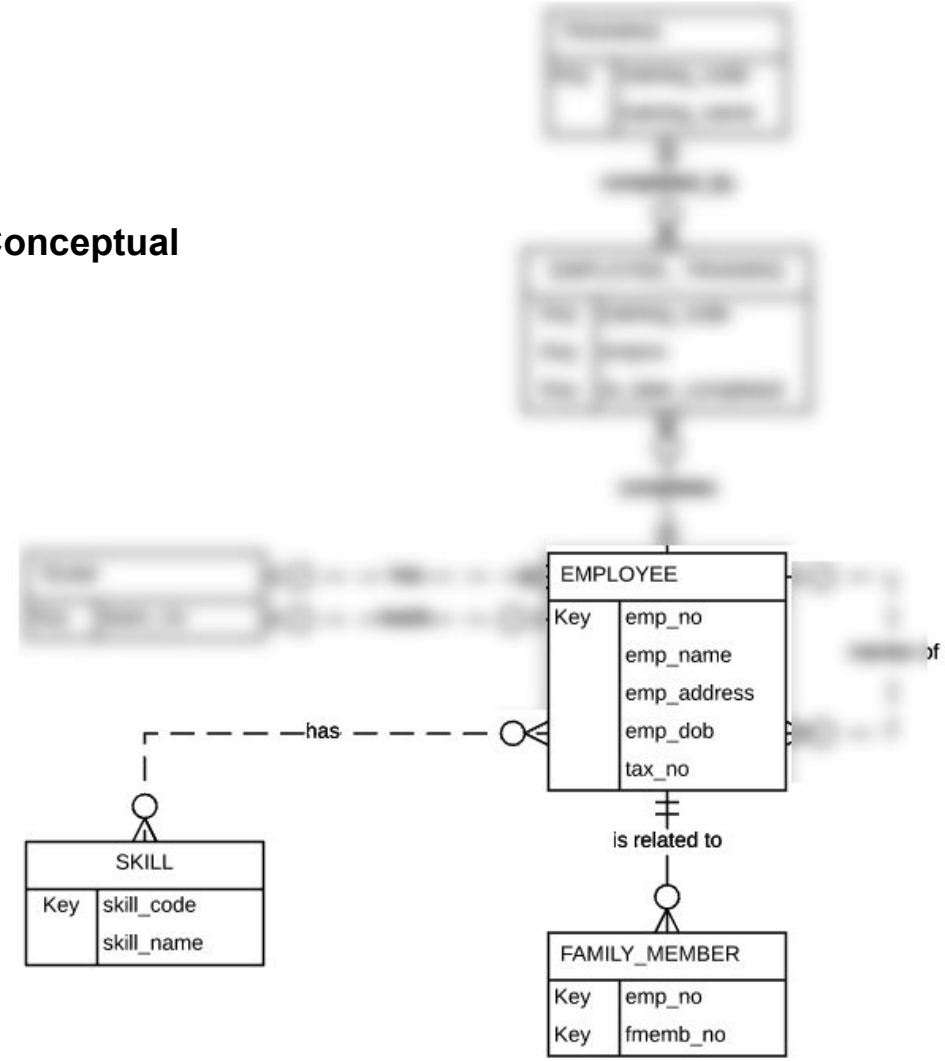
Mapping a Weak Entity

- For each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. The PK of the identifying relation is also included as the FK in this new relation.

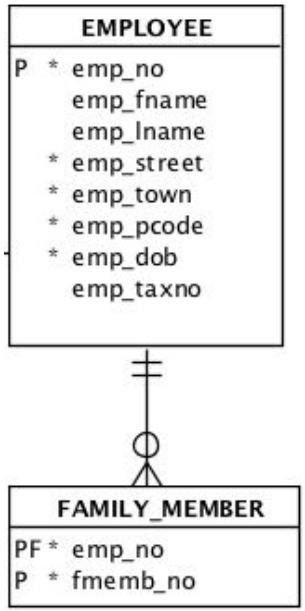




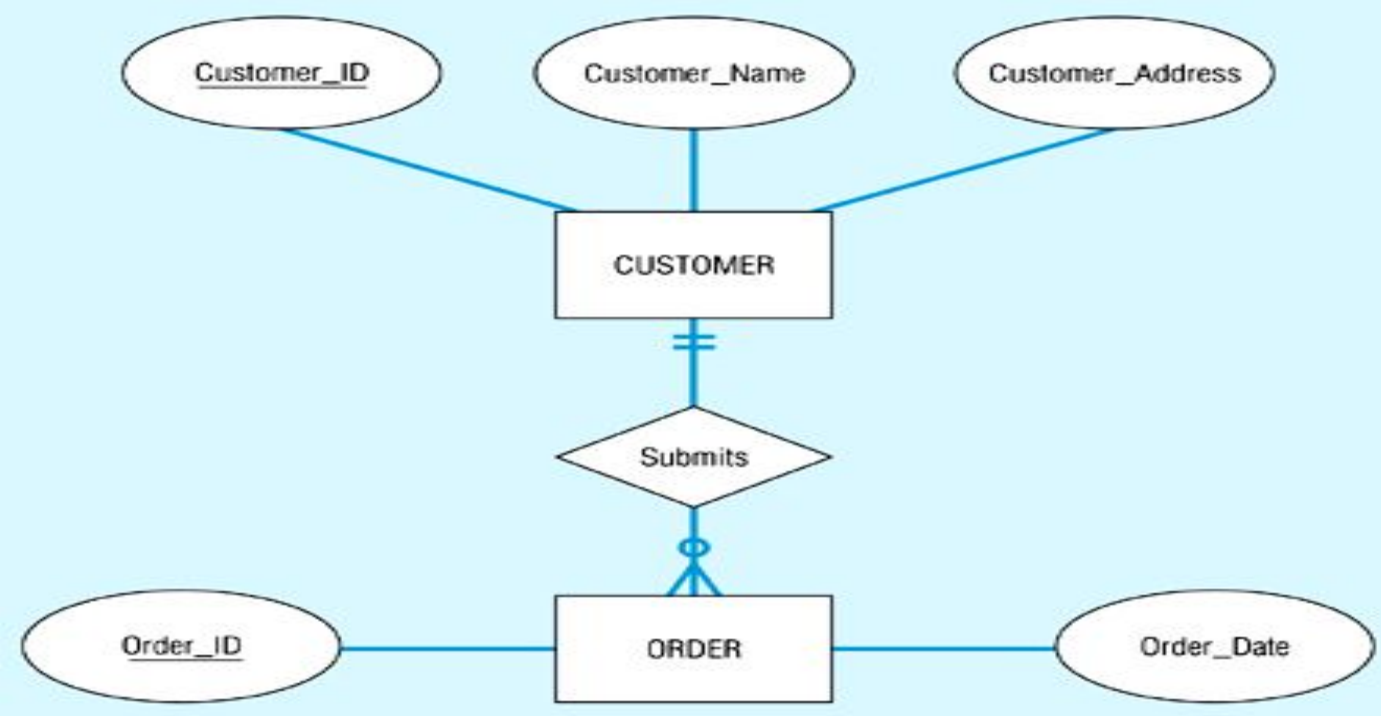
Conceptual



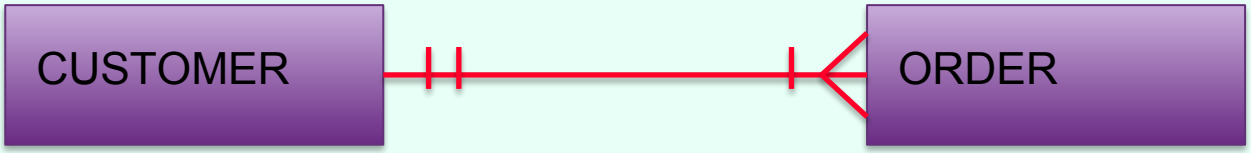
Logical



Mapping a 1:M Binary Relationship

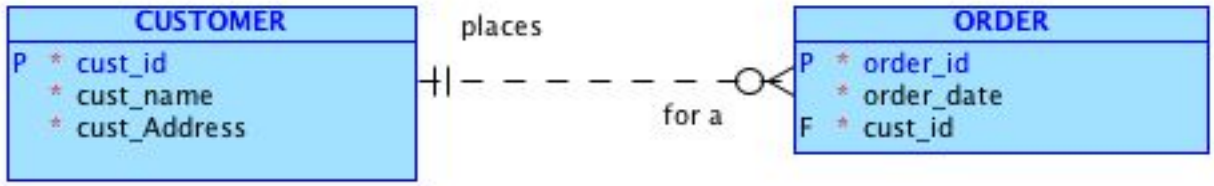


Q2. Where would you place the Foreign Key when you map this ER diagram into the relational model?



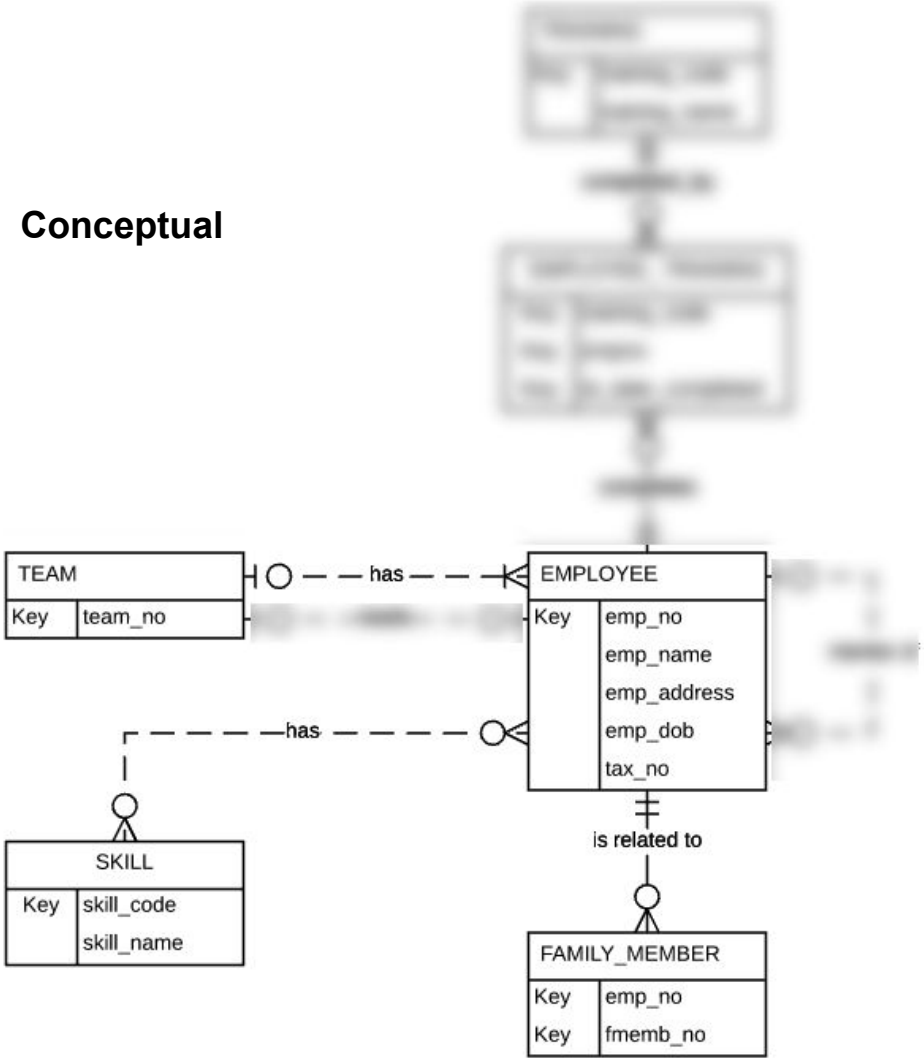
a. CUSTOMER
b. ORDER
c. Both CUSTOMER and ORDER.
d. None, no FK is needed.

Map Binary Relationships (1:M)

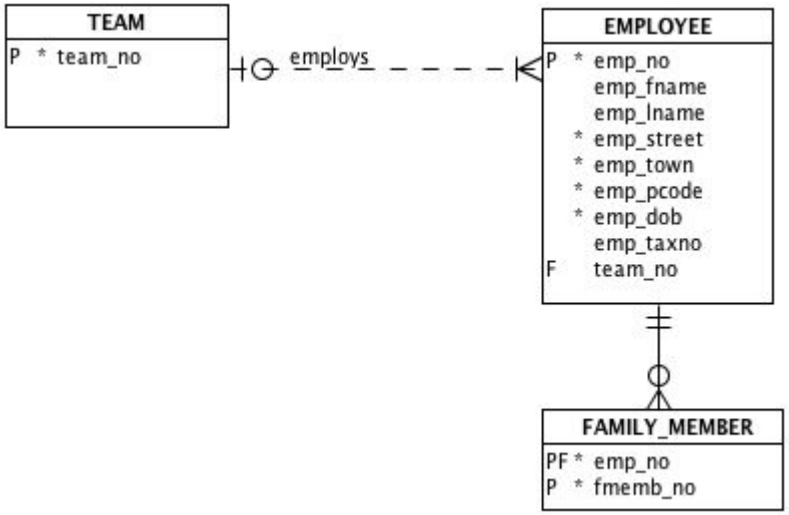


For each 1:M binary relationship, first create a relation for each of the two entity types participating in the relationship. Then include the PK attribute (or attributes) of the entity on the one-side of the relationship as the FK on the many-side of the relationship.

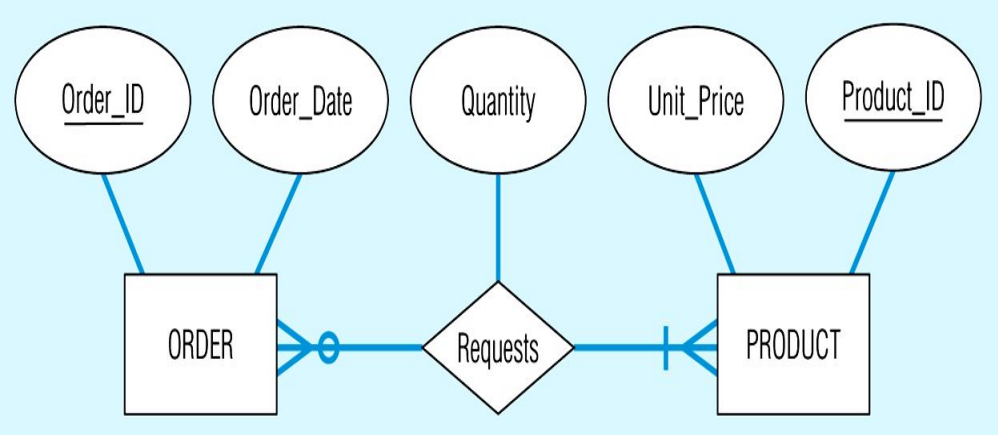
Conceptual



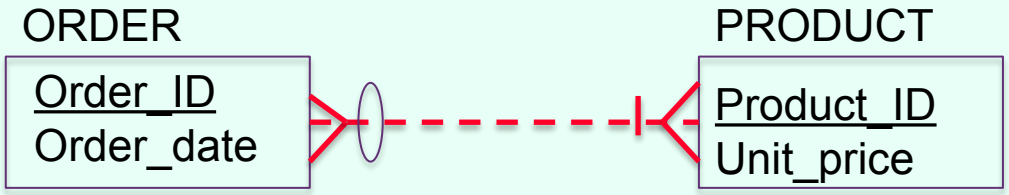
Logical



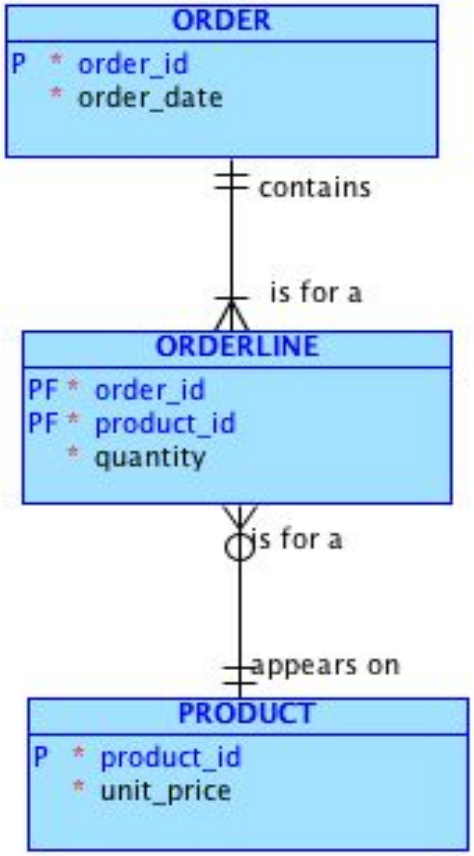
Mapping a M:N Binary Relationship



Q3. What will be the Primary Key of the new created relation resulting from mapping this ER model at the conceptual level into a relational model?



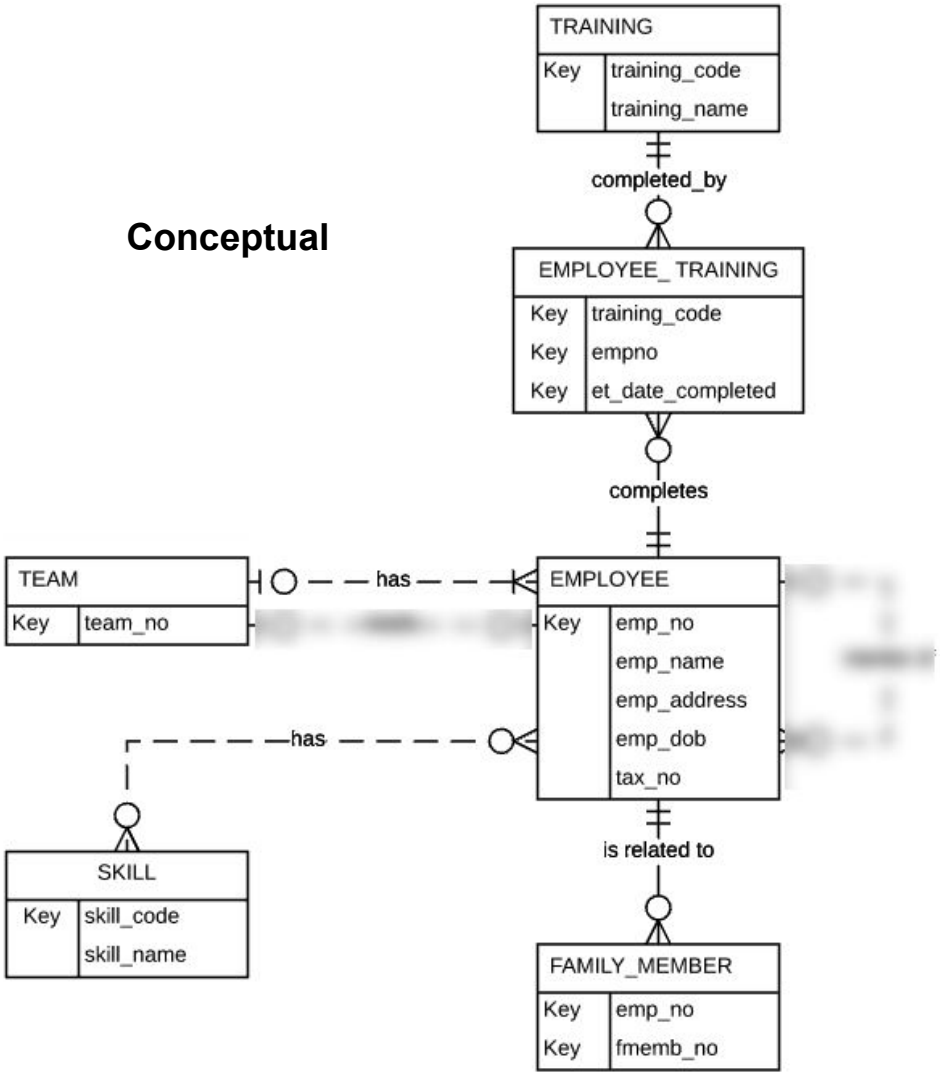
a. The primary key of the ORDER table.
b. The primary key of the PRODUCT table.
c. The combination of primary keys of ORDER and PRODUCT.


ORDER
P * order_id
* order_date
contains
is for a
ORDERLINE
PF * order_id
PF * product_id
* quantity
is for a
appears on
PRODUCT
P * product_id
* unit_price

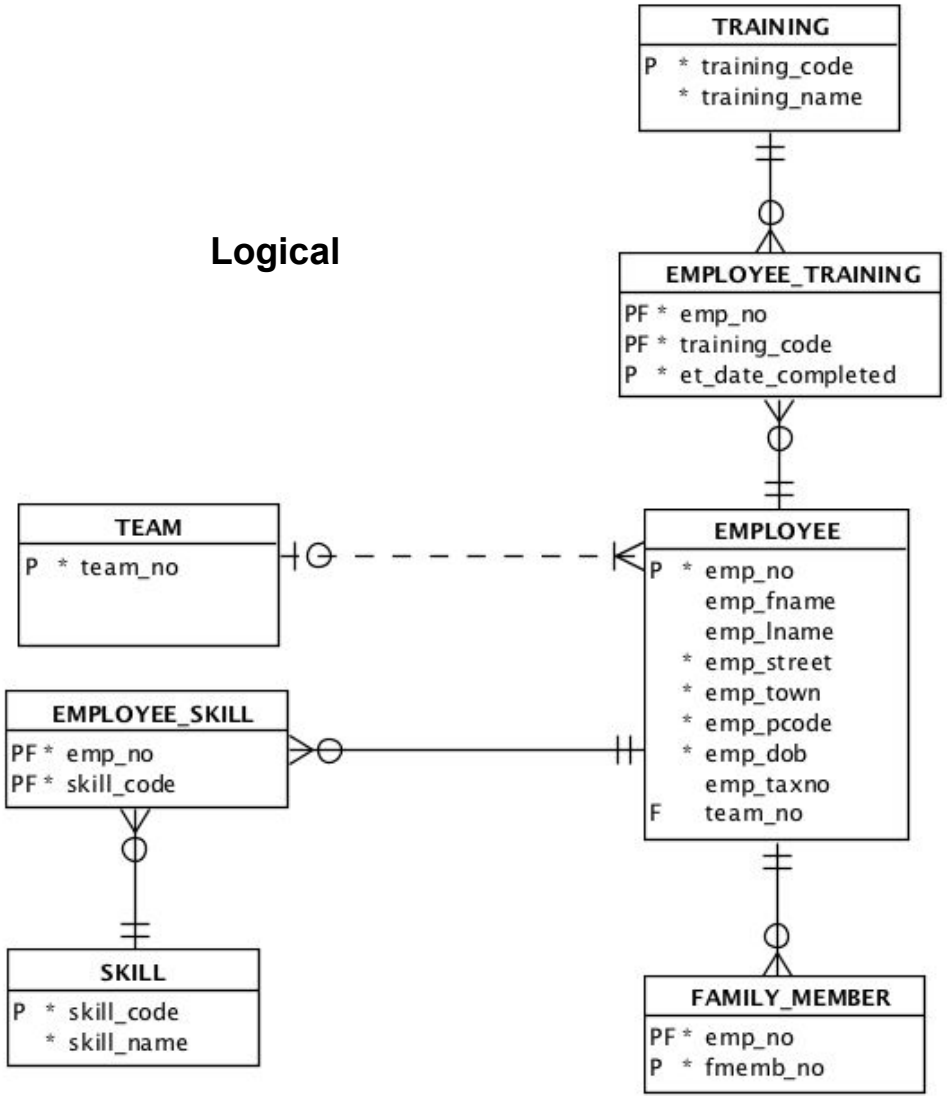
Map Binary Relationship (M:N)

- For a M:N binary relationship
  - First create a relation for each of the two entity types participating in the relationship.
  - Then create a new relation and include as foreign key attributes, the PK attribute (or attributes) for each of the two participating entity types. These attributes become the PK of the new relation.
  - If there are any nonkey attributes associated with the M:N relationship, they are also included in the new relation.

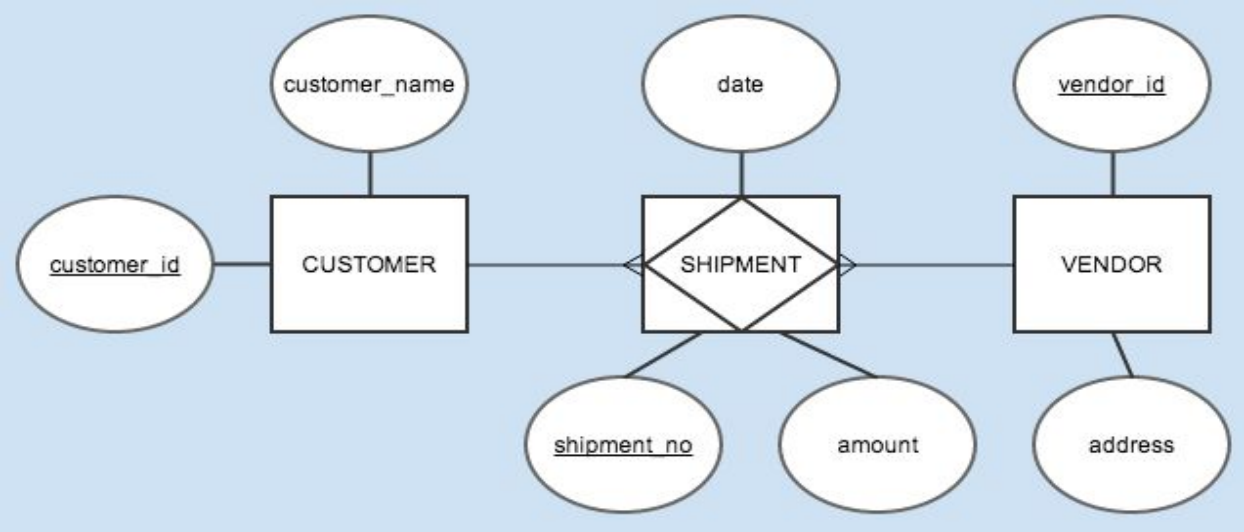
Conceptual



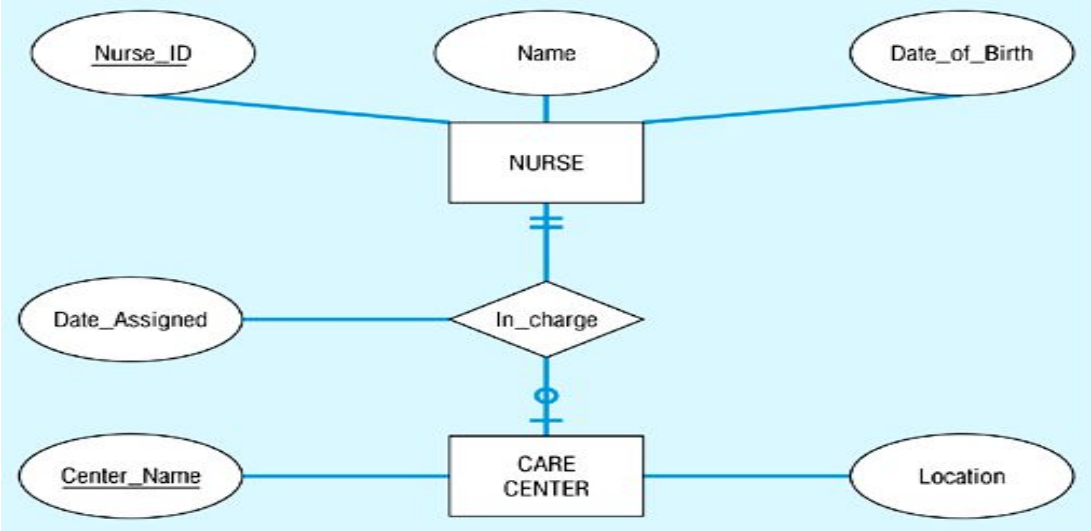
Logical



Mapping an associative entity with an Identifier



Mapping a 1:1 Binary Relationship



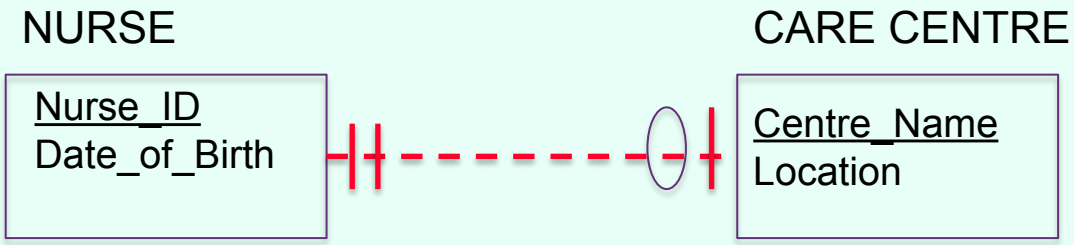
NURSE

Nurse_ID
Date_of_Birth

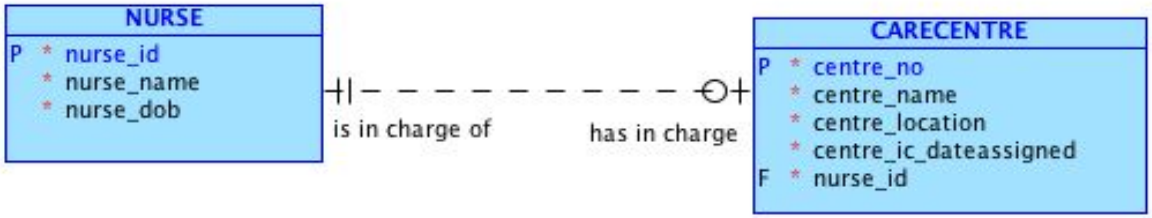
CARE CENTRE

Centre_Name
Location

Q4. Where would you place the Foreign Key when mapping this ER diagram into a relational model?



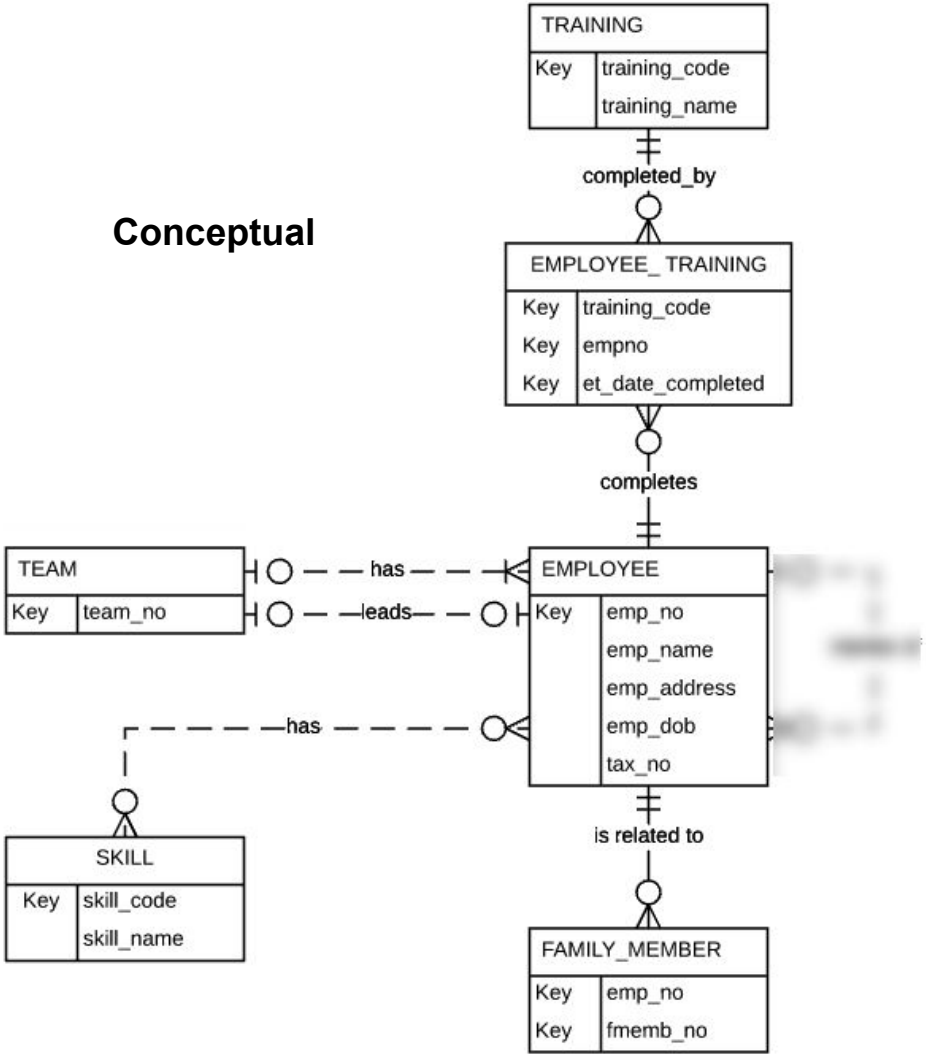
A. NURSE
B. CARE CENTRE
C. Both NURSE and CARE CENTRE
D. No FK is needed.


NURSE
P * nurse_id
* nurse_name
* nurse_dob
is in charge of
has in charge
CARECENTRE
P * centre_no
* centre_name
* centre_location
* centre_ic_dateassigned
F * nurse_id

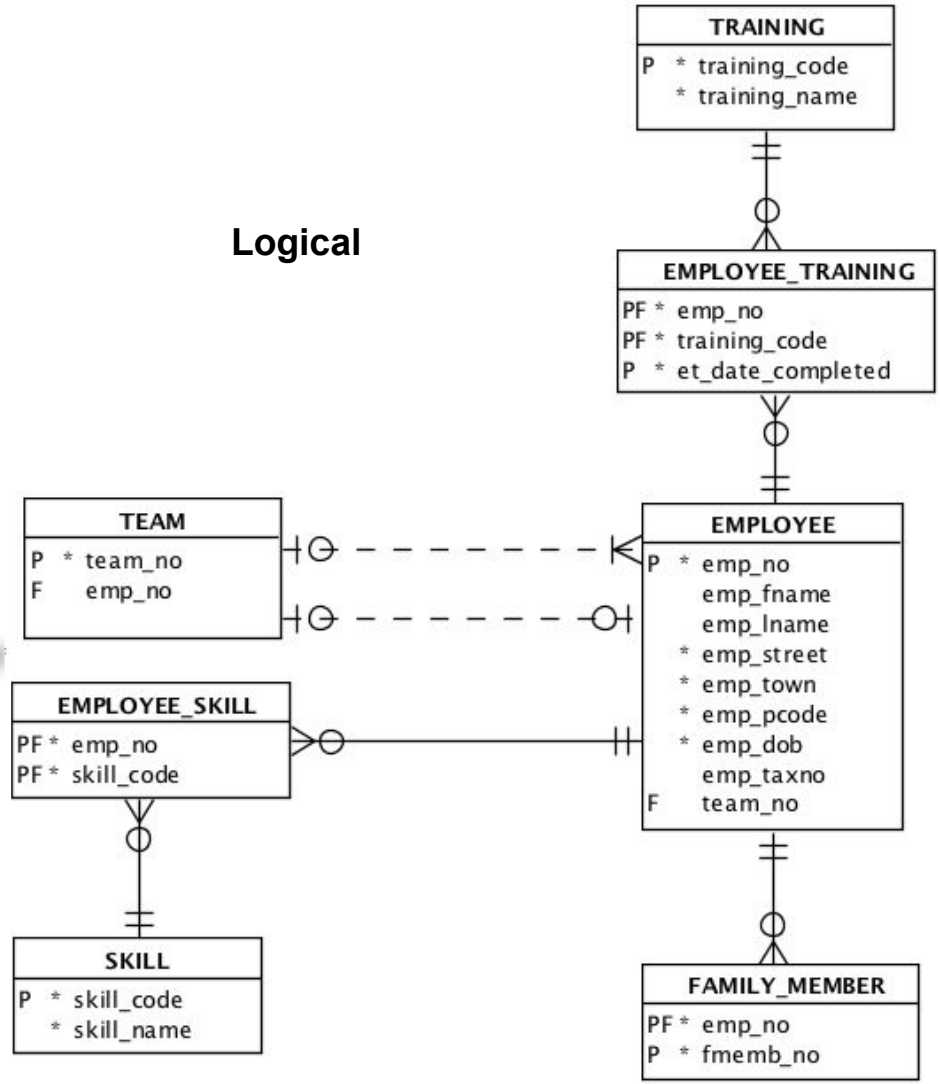
Map Binary Relationship (1:1)

- Create two relations, one for each of the participating entity types.
  - The primary key (PK) on the mandatory side of the relationship becomes the foreign key (FK) on the optional side of the relationship.
  - where both are optional place the FK on the side which causes the fewest nulls
  - Special case: *1:1 total* relationship (mandatory participation on both sides)
    - Consider consolidating the two entity types into one relation

Conceptual



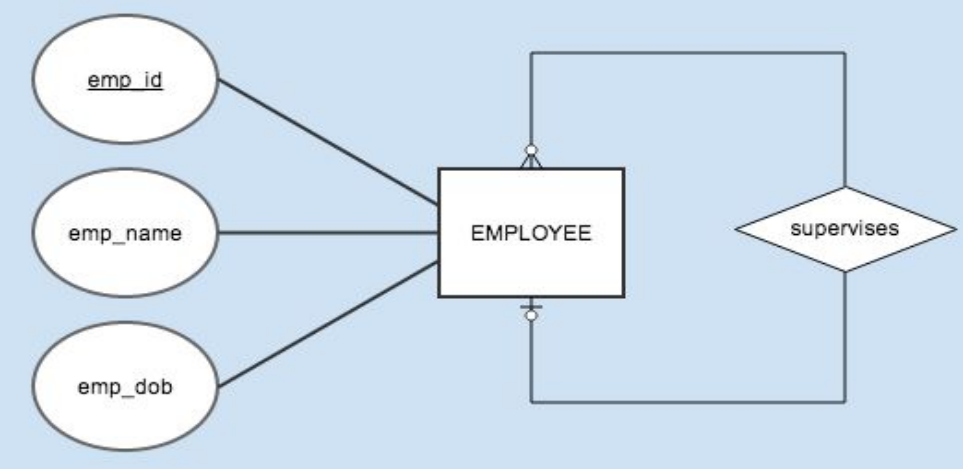
Logical



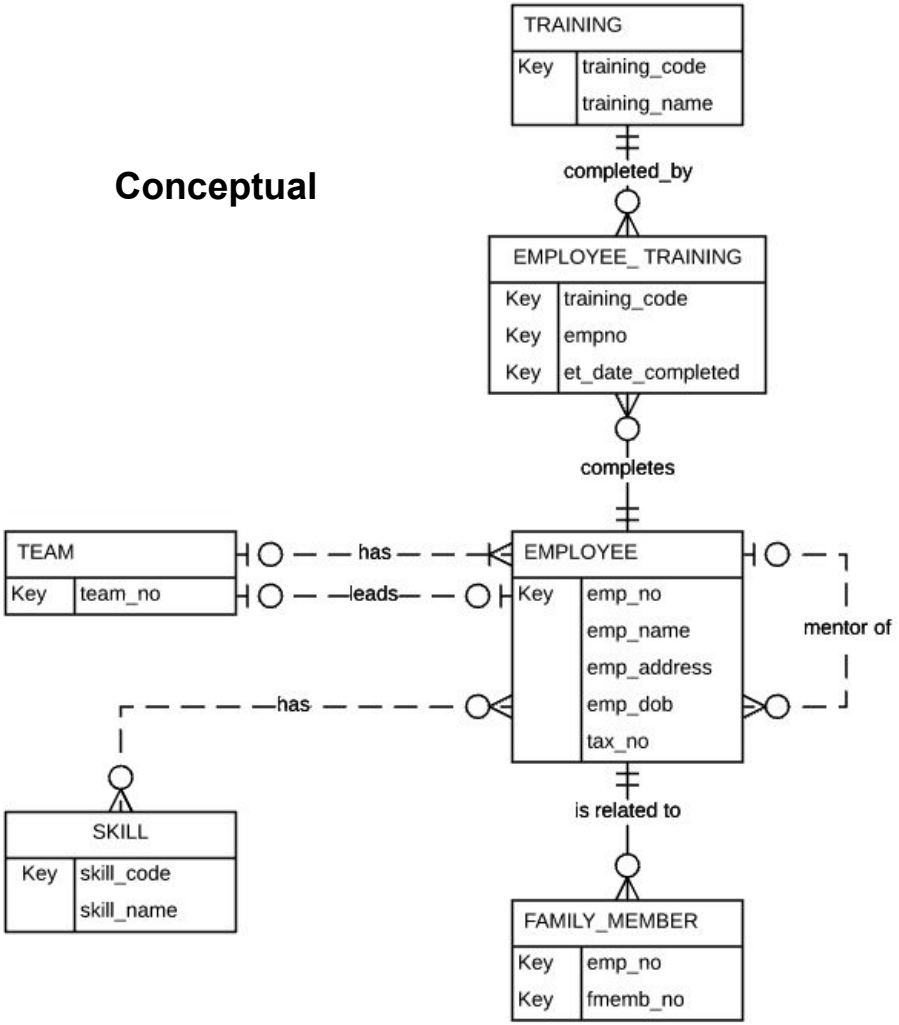
Map unary relationships

- Unary Relationship is a relationship between the instances of a single entity type.
- **Unary 1:M Relationship** – A relation is created for the entity type. Add a FK within the same relation that references the PK of the relation. A recursive foreign key is a FK in a relation that references the PK values of the same relation.
- **Unary M:N Relationship** – Two relations are created, one for the entity type in the relationship and the other as the associative relation to represent the M:N relationship itself. The PK of the associative relation consists of two attributes (with different names) taking their values from the PK of the other relation.

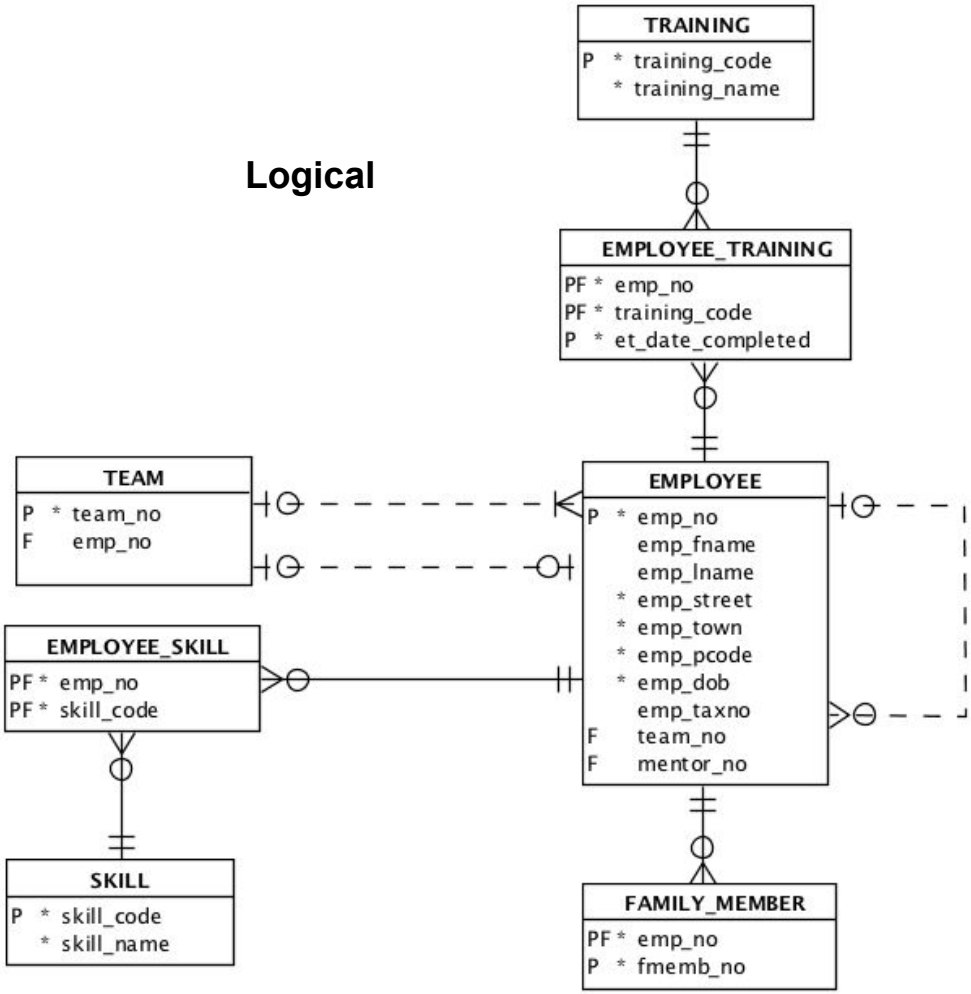
Mapping a 1:M Unary Relationship



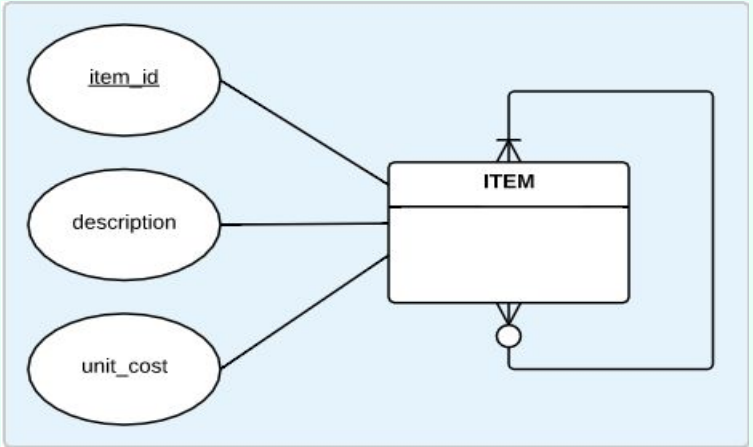
Conceptual



Logical

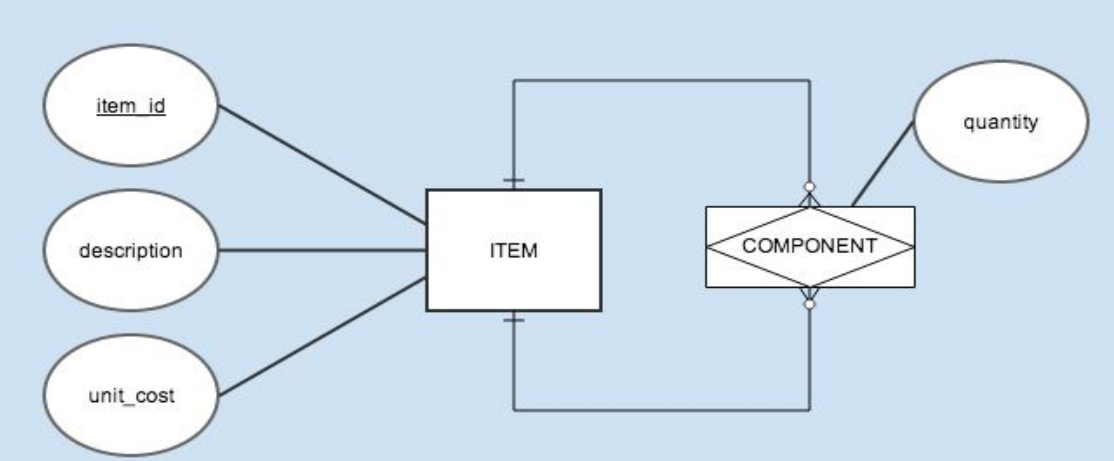


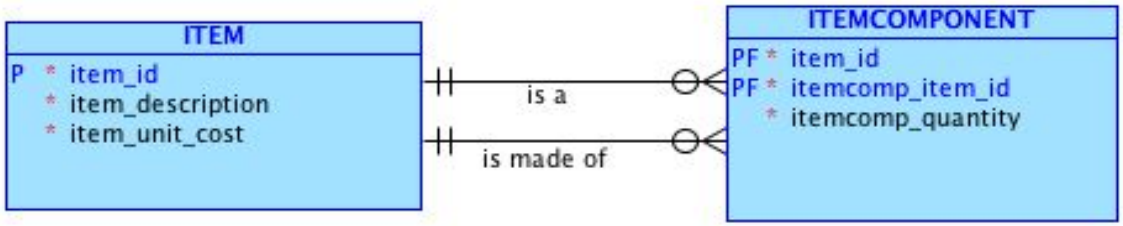
Q5. How many relations/tables and relationships do we need to implement the model below into a relational model?



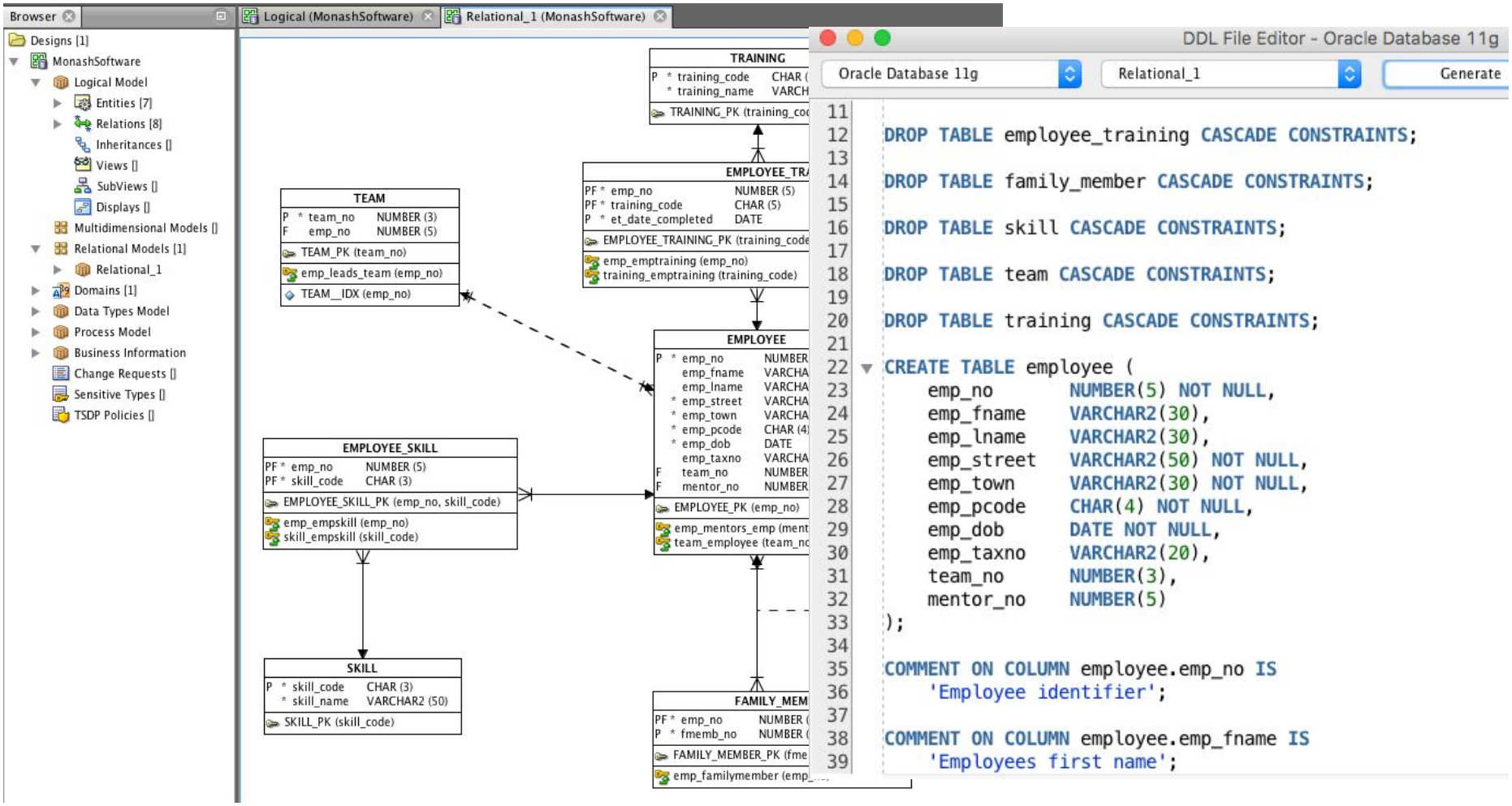
a. 2 tables, 1 relationship
b. 2 tables, 2 relationships
c. 3 tables, 2 relationships
d. 4 tables, 3 relationships

Mapping a M:N Unary Relationship

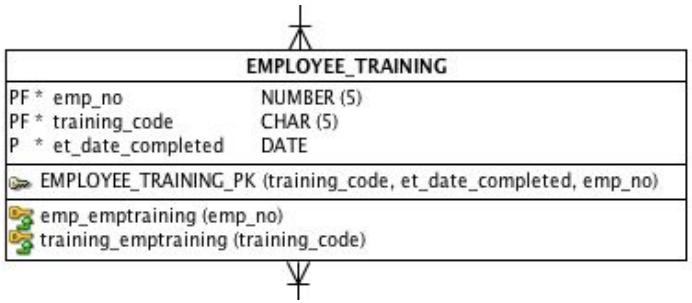




SQL Developer Data Modeler



Adding surrogate keys



Surrogate PK's may be added on the logical model provided they are justified (include in documentation / assumptions)

Potential problem:

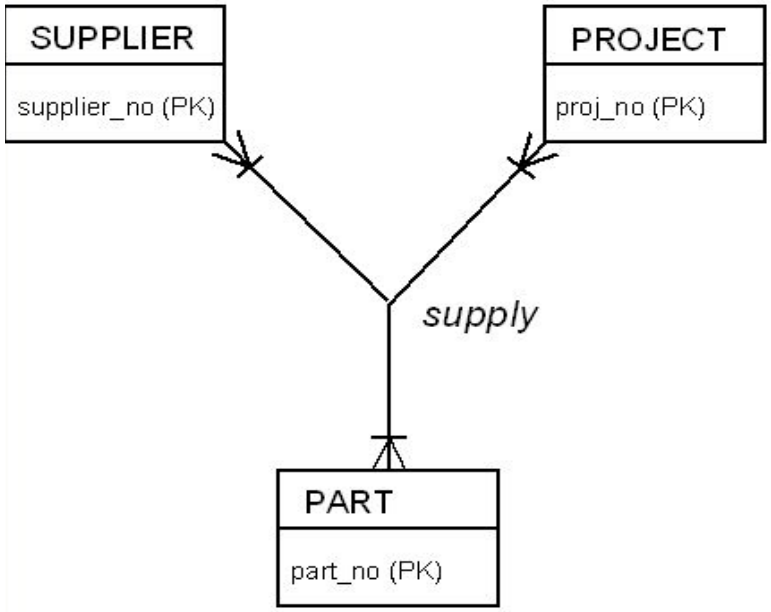
Need to ensure that the identified key from the conceptual model (emp_no, training_code, et_date_completed) will still remain unique

- define a unique index on attributes of key

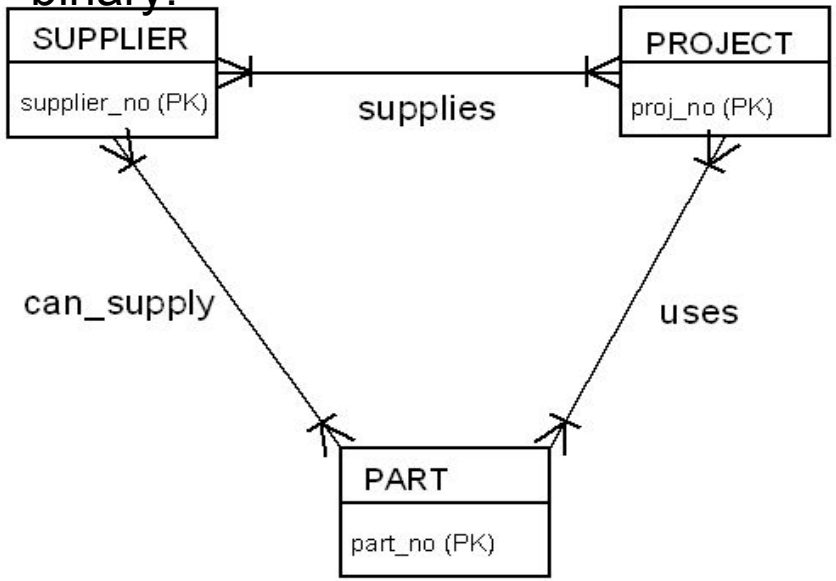
et_no	emp_no	training_code	et_date_completed
1	101	ORA01	1-Oct-2016
2	101	ORA01	1-Oct-2016
3	101	ORA01	1-Oct-2016

Ternary Relationships

Ternary



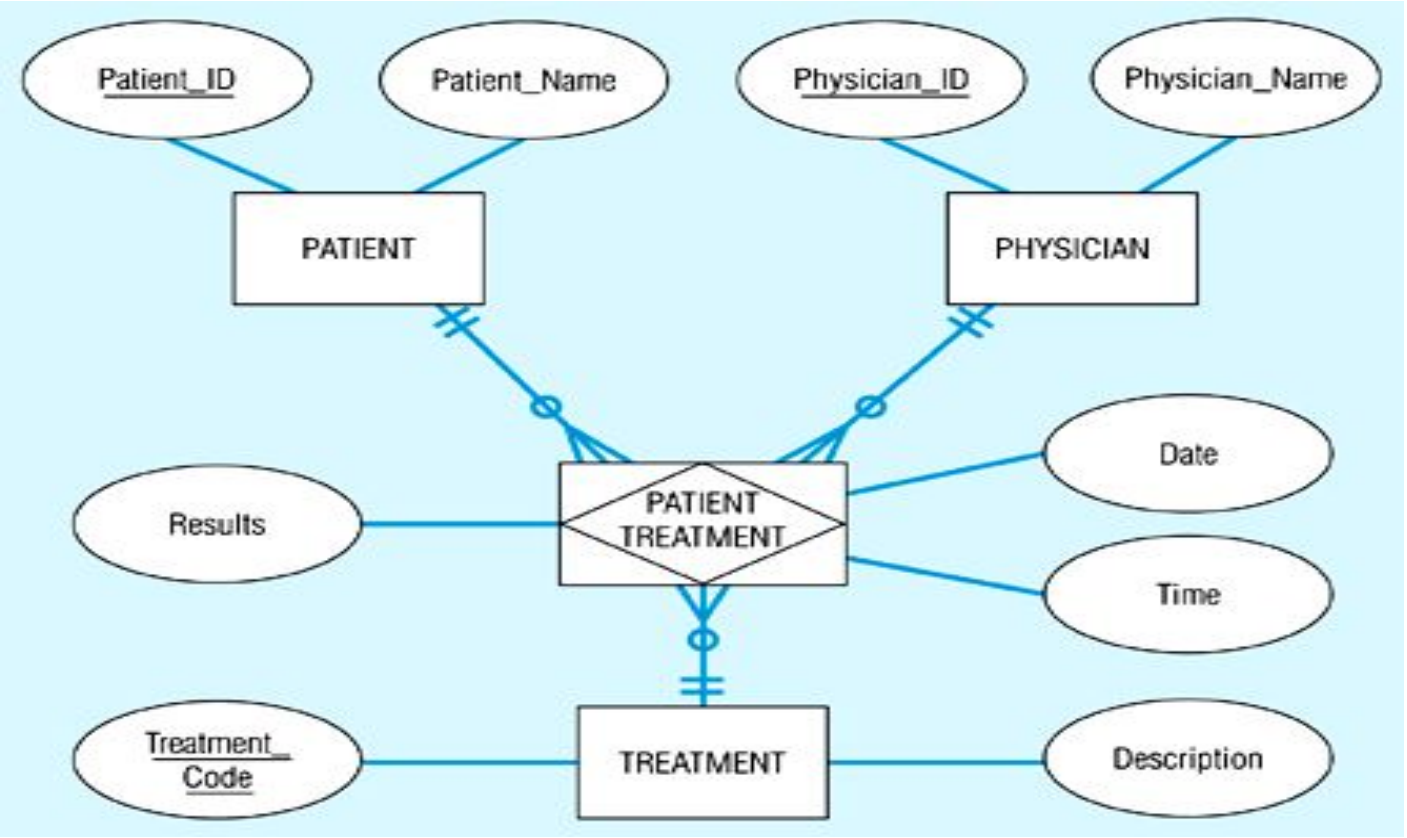
modelled as binary:



Ternary Relationships – model as binary relationships?

- Ternary represents more information than three binary relationships
- For example - Supplier 1 supplies Project 2 with Part 3 -
  - ternary
    - instance (supplier 1, project 2, part 3) exists
  - binaries
    - instances
      - (supplier1, project 2) (project 2, part 3) (supplier 1, part 3)
    - BUT does not imply (supplier 1, project 2, part 3)
- How then do we map such relationships?

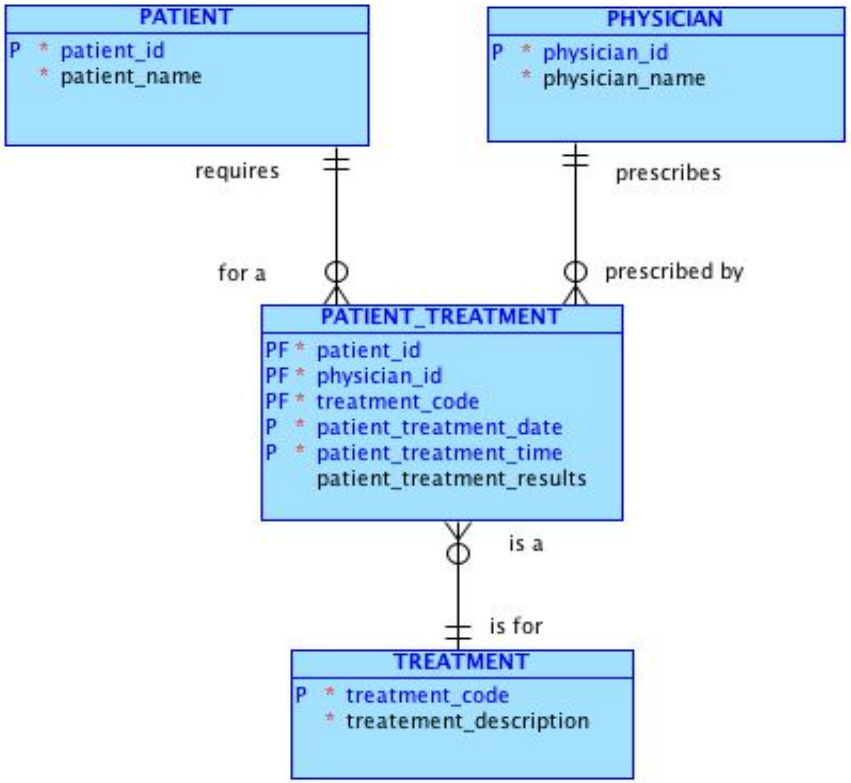
Mapping a Ternary Relationship



Map Ternary (and n-ary) Relationships

- Ternary relationship should be converted to an associative entity.
  - To map an associative entity type that links three regular entity types, an associative relation is created.
  - The default PK of this relation consists of the three PK attributes for the participating entity types.
  - Any attributes of the associative entity type become attributes of the new relation.

Mapping a Ternary Relationship



Reference

Hoffer, J. A. , Prescott, M. B. & McFadden, F. R.
“Modern Database Management”