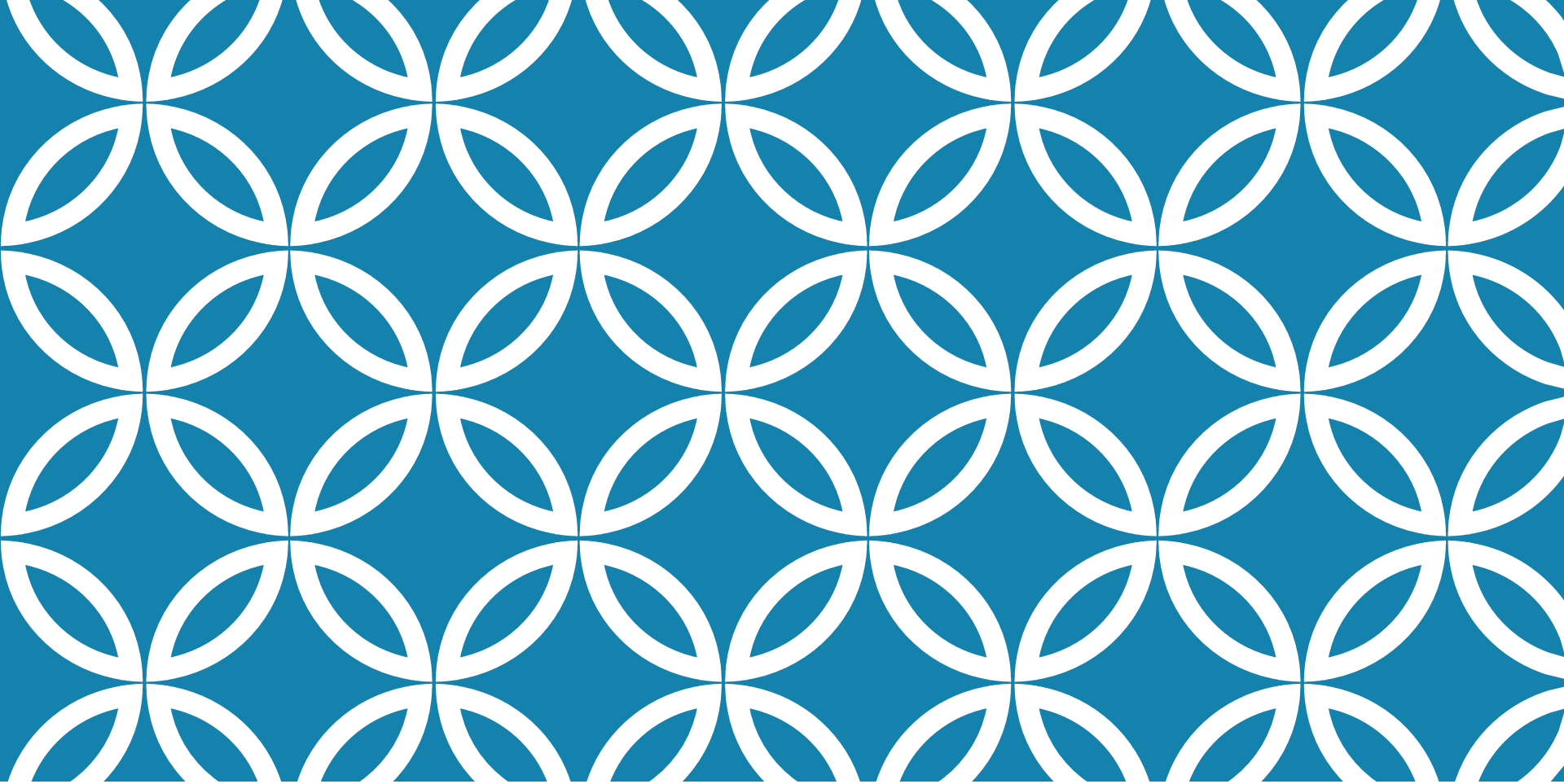


TOPIC: ENTITY RELATIONSHIP MODELS

CT230
Database
Systems 1

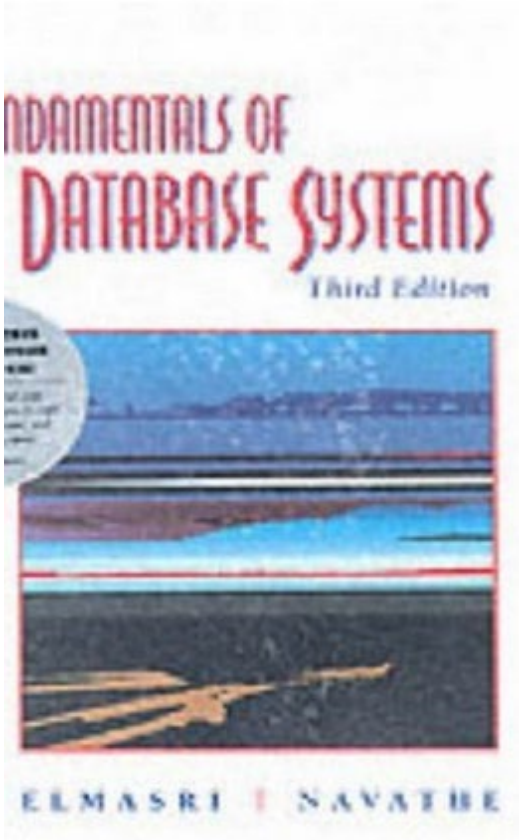
TOPIC:
Designing Tables with ER Models

See

Elmasri and Navathe book

Chapter 3 & Chapter 9

(3rd Edition)



DATA MODELS

Data models are concepts to describe the **structure** of a database. They comprise

- High level or logical models;
- Representational/Implementation data models;
- Physical Data models

Data models allow for **database abstraction**

DATA DESIGN and ENTITY RELATIONSHIP MODELS

Entity Relationship Models:

- Provide a way to **model** the data that will be stored in a system.
- The models are then used to **create tables** in the relational model.

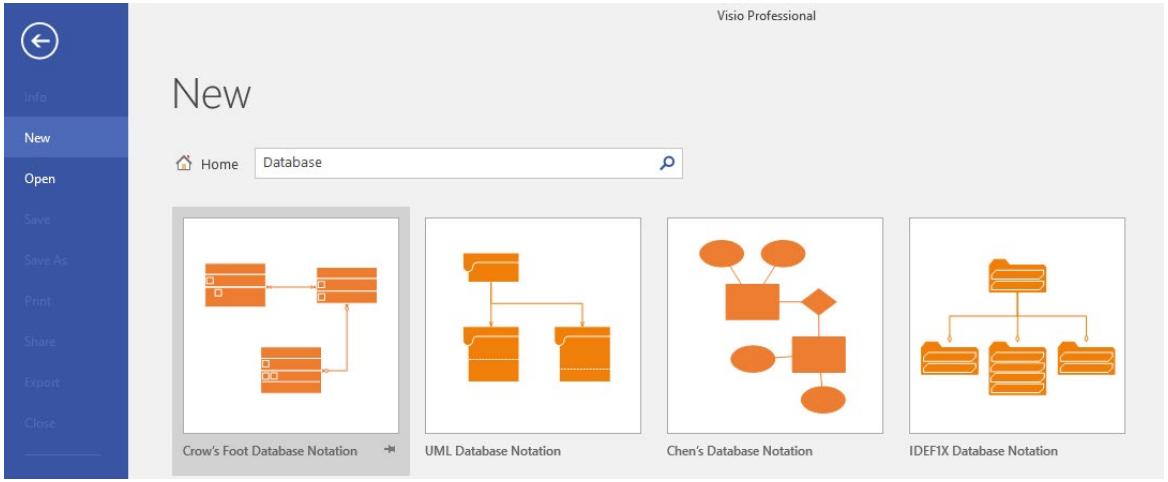
ENTITY RELATIONSHIP (ER) MODELS

ER models are a top-down approach to database design.

They are used to identify:

1. the important data to be stored in database called **entities.**
2. the **relationships** between the entities.
3. the **attributes** of entities.
4. the **constraints** of relationships and entities.

Software to Create ER Models

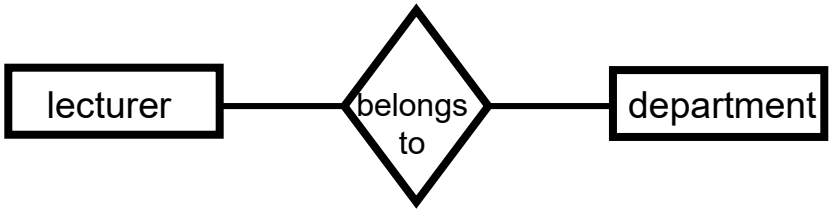


A comprehensive drawing package by Microsoft - MS Visio - supports the drawing of a large set of diagrams, including database ones. This is worth getting with your free Microsoft access.

Many other similar packages available:

- Edraw: https://www.edrawsoft.com/entity-relationship-diagrams.php
- Astah: http://astah.net/
- Lucidchart: www.lucidchart.com

NOTATION



A number of different notations can be used to represent the same model.

The original notation (Chen) uses diamonds, rectangles and ellipses.

It is easier to hand-draw so useful in an exam situation.

It is less implementation oriented than other notations.

ER MODEL NOTATIONS *CTD.*

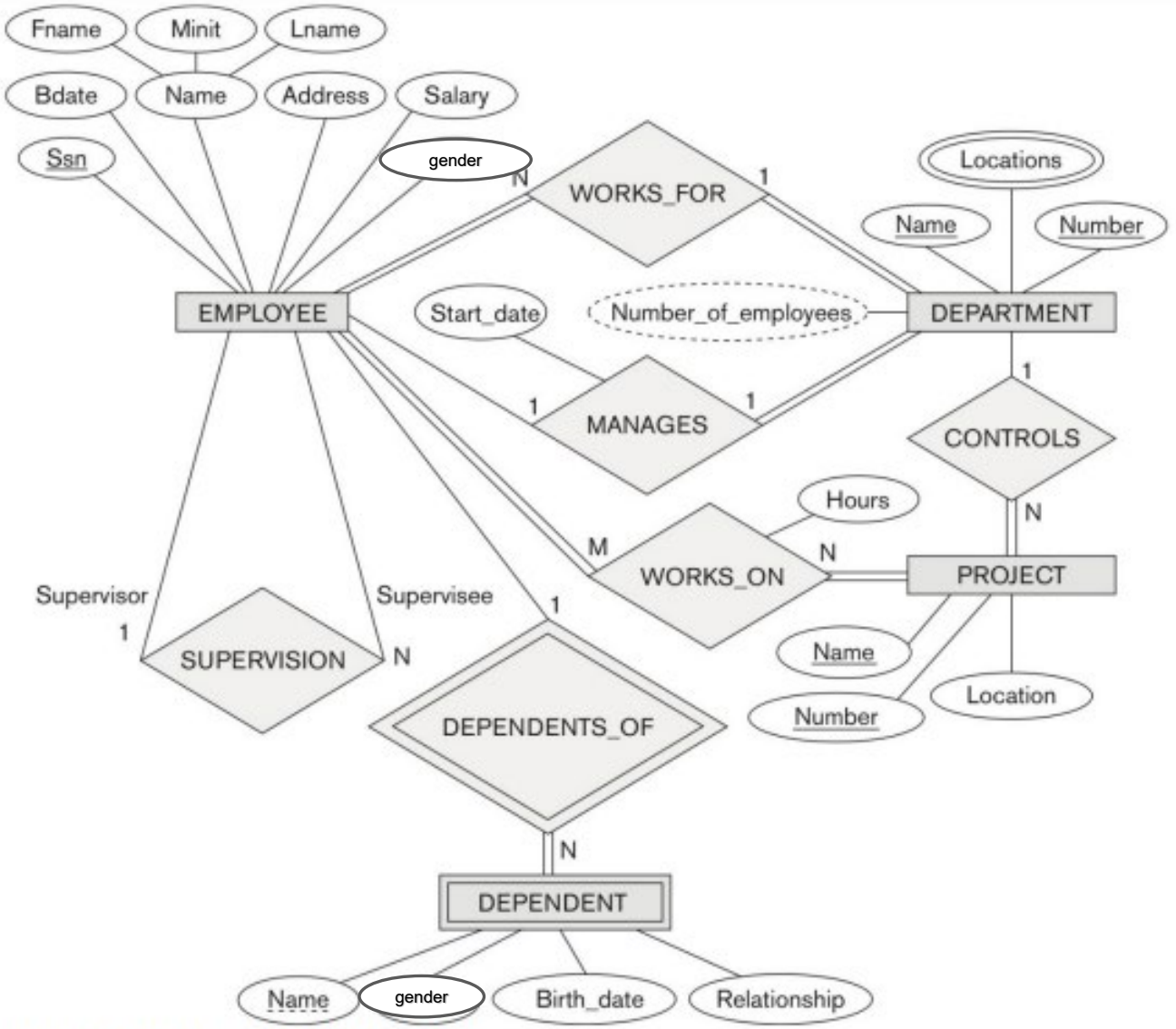
There are many notations in use, some of the more common:

- Chen Notation
- IE Crow's foot Notation
- UML
- Integrated Definition 1, Extended (IDEF1X)

Different software products often have their own minor variations of the above.

COMPANY ER MODEL EXAMPLE

Consider the ER diagram (Chen's notation) of the Company Schema ….



SOME DEFINITIONS:

Entity type: group of objects, with the same properties, which are identified as having an independent existence …

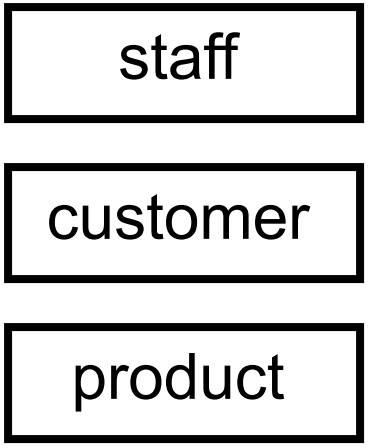
e.g.,

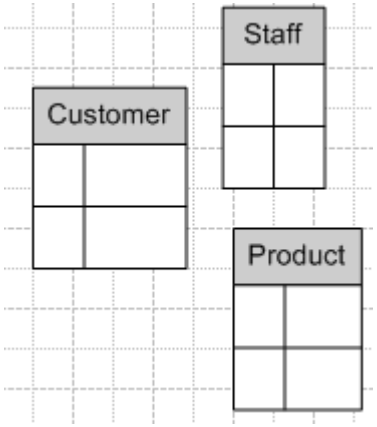
staff

customer

product

employee

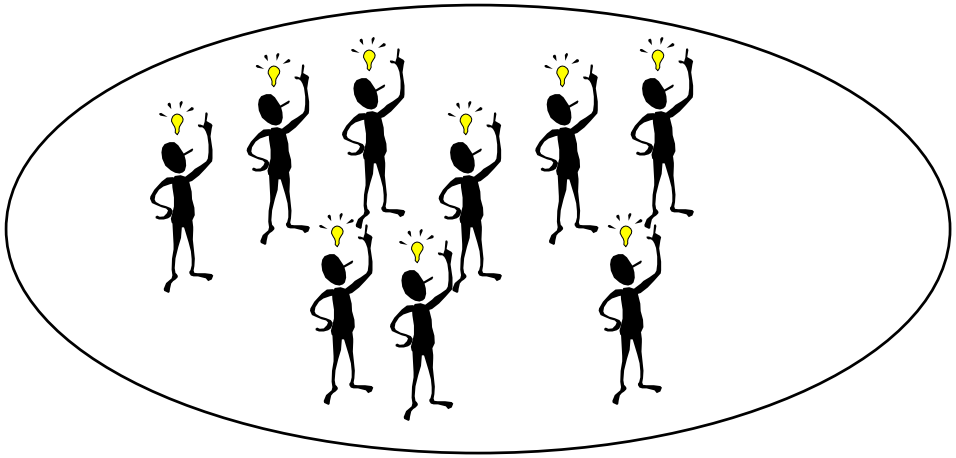




ENTITY INSTANCE AND ENTITY TYPE

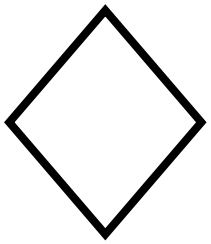
- An entity type is a collection of *entity instances* that share common properties or characteristics
- An entity instance or entity occurrence is a single uniquely identifiable occurrence of an entity type (e.g., row in a table).

Entity Instance:
One employee,
e.g. John B Smith



Entity Type: employee

RELATIONSHIP TYPE:

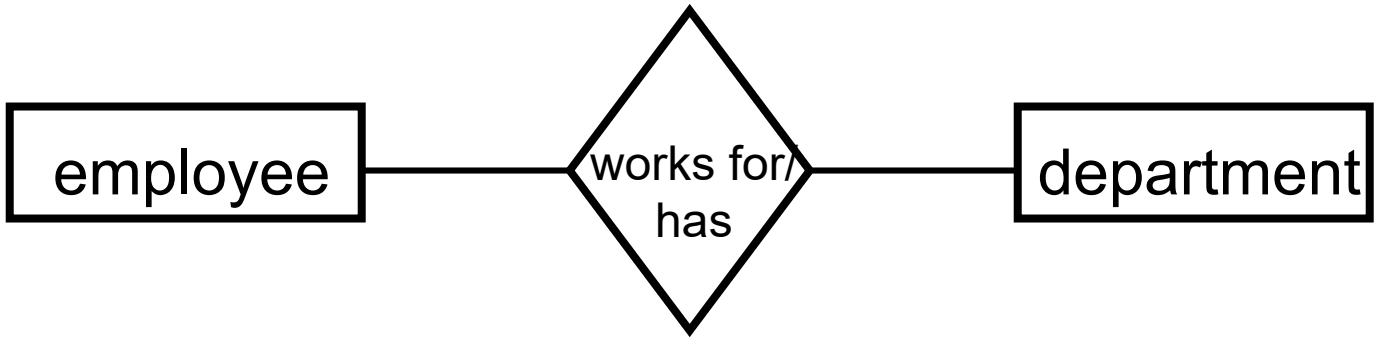


A set of meaningful relationships among entity types

e.g.,

employee “works for” department

department “has” employee

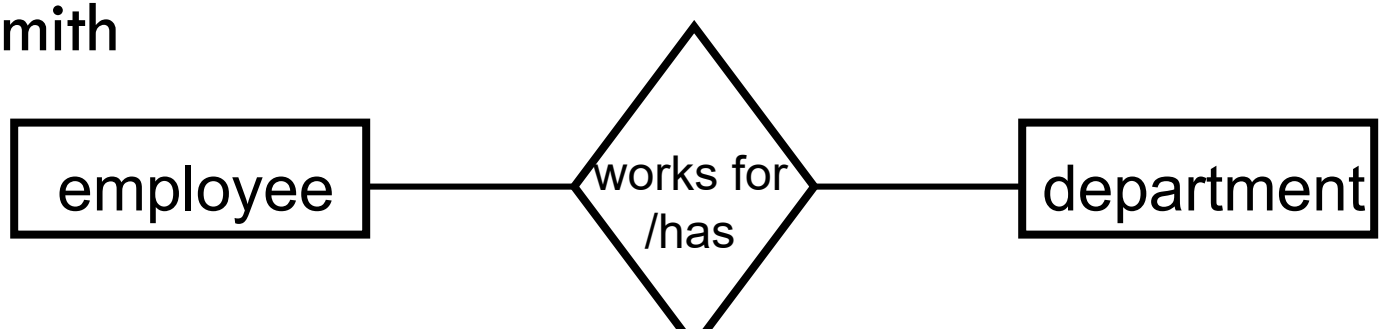


RELATIONSHIP OCCURRENCE (INSTANCE):

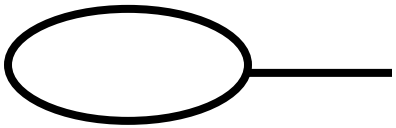
A uniquely identifiable association which includes one occurrence from each participating entity type; reading left to right and right to left.

e.g.

- Left-to-Right: John Smith "works for" Research department
- Right-to-left: Research department "has" John Smith



ATTRIBUTES



Attributes are a named property or characteristic of an entity.

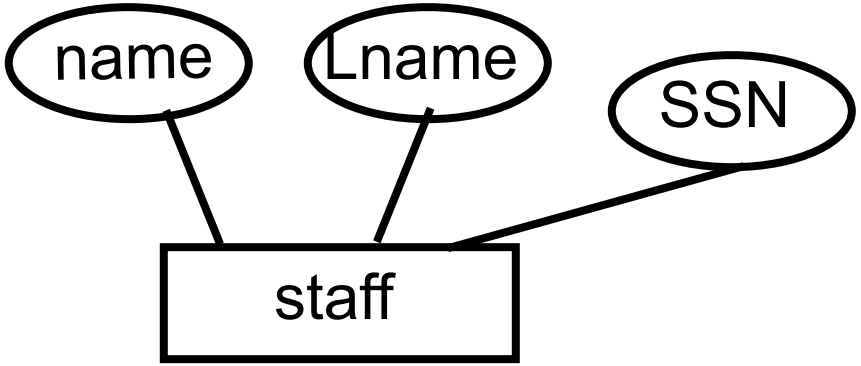
Each entity has a set of attributes associated with it.

Several types of attributes exist:

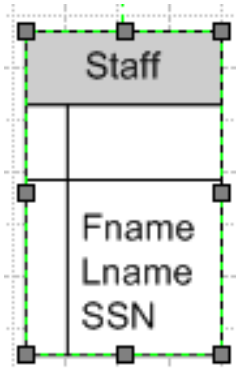
- Key
- Composite
- Derived
- Multi-valued

ATTRIBUTE NOTATION

Chen: An oval enclosing the name of the attribute



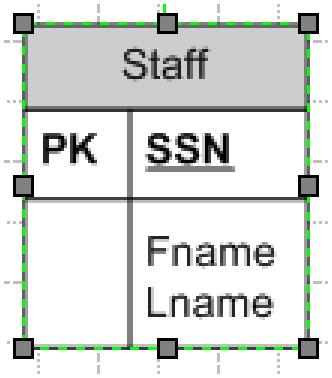
Crow: Listed in the entity box



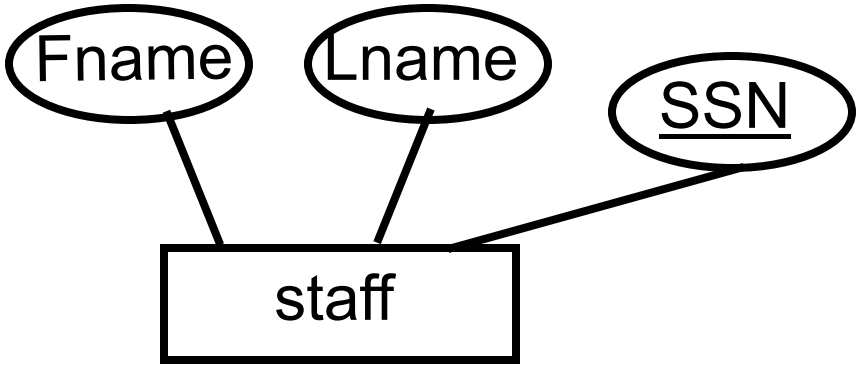
KEY ATTRIBUTES

- Each entity type must have an attribute or set of attributes that uniquely identifies each instance from other instances of the same type.
- A candidate key is an attribute (or combination of attributes) that uniquely identifies each instance of an entity type.
- A primary key (PK) is a candidate key that has been selected as the identifier for an entity type.
- **Notation:** Underline attribute name chosen as primary key

PK NOTATION: SSN PRIMARY KEY



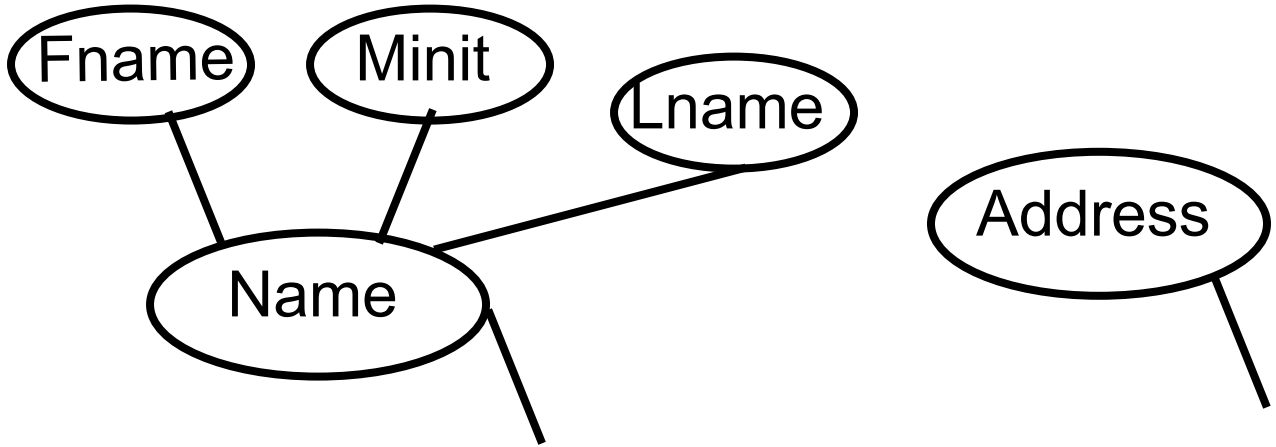
	Physical Name	Data Type	Req'd	PK	
	Fname	CHAR(10)	☐	☐	Fname is of Staff
	Lname	CHAR(10)	☐	☐	Lname is of Staff
▶	SSN	CHAR(10)	☑	☑	SSN identifies Staff
			☐	☐	

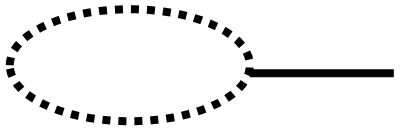


COMPOSITE AND SIMPLE (ATOMIC) ATTRIBUTES

A composite attribute is an attribute that is composed of several more basic/atomic attributes.

If the composite attribute is referenced as a whole only, then there is no need to subdivide it into component attributes, otherwise you should divide it:





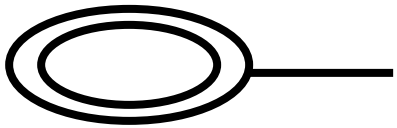
STORED AND DERIVED ATTRIBUTES

A **derived attribute** is an attribute whose value can be determined from another attribute.

For Chen's notation, the notation is a dotted oval.

For crow's foot notation, derived attributes can be represented by enclosing the attribute in [], e.g., [age].

MULTI-VALUED ATTRIBUTES

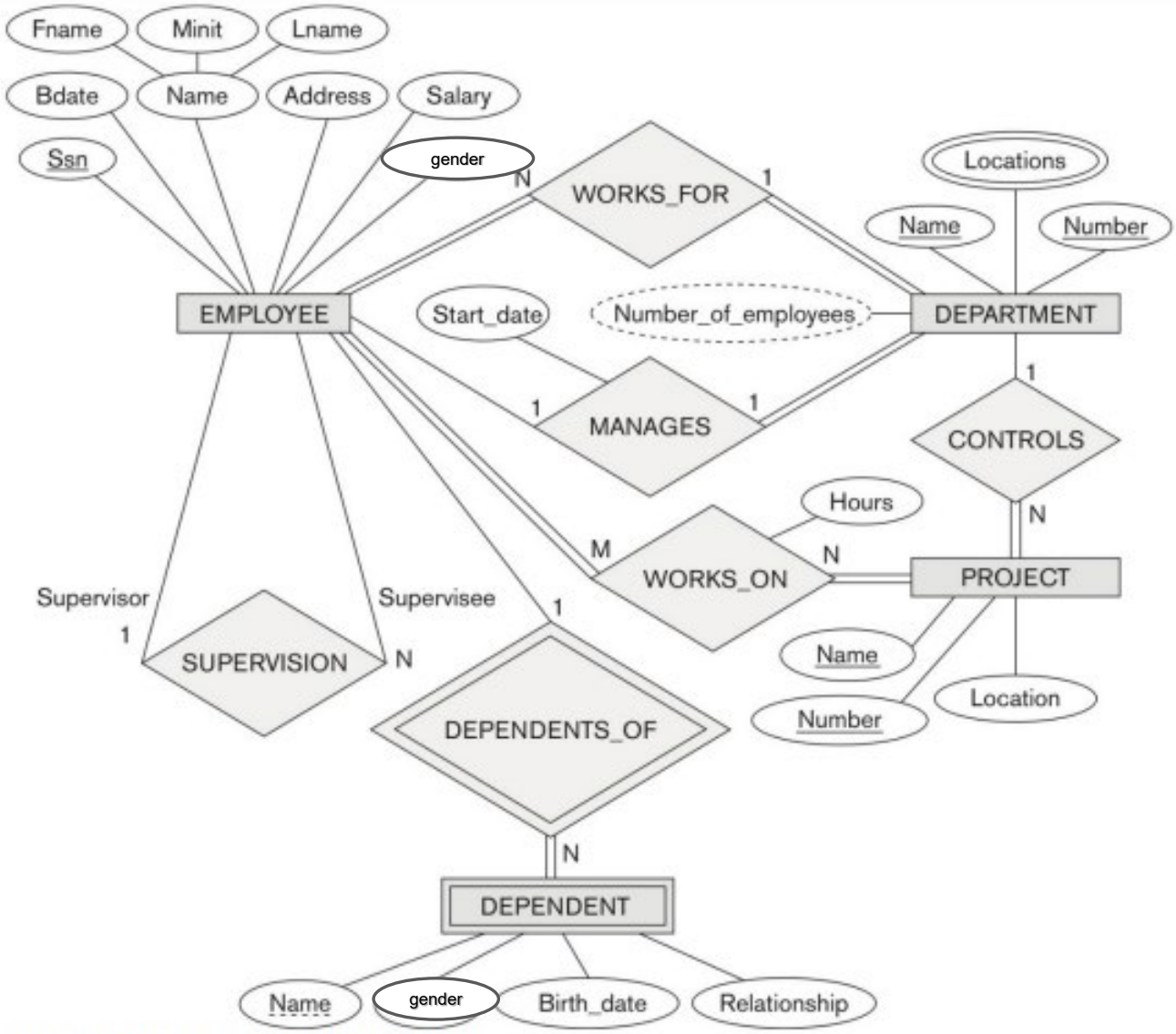


A **multi-valued attribute** is an attribute which has lower and upper bounds on the number of values for an individual entry.

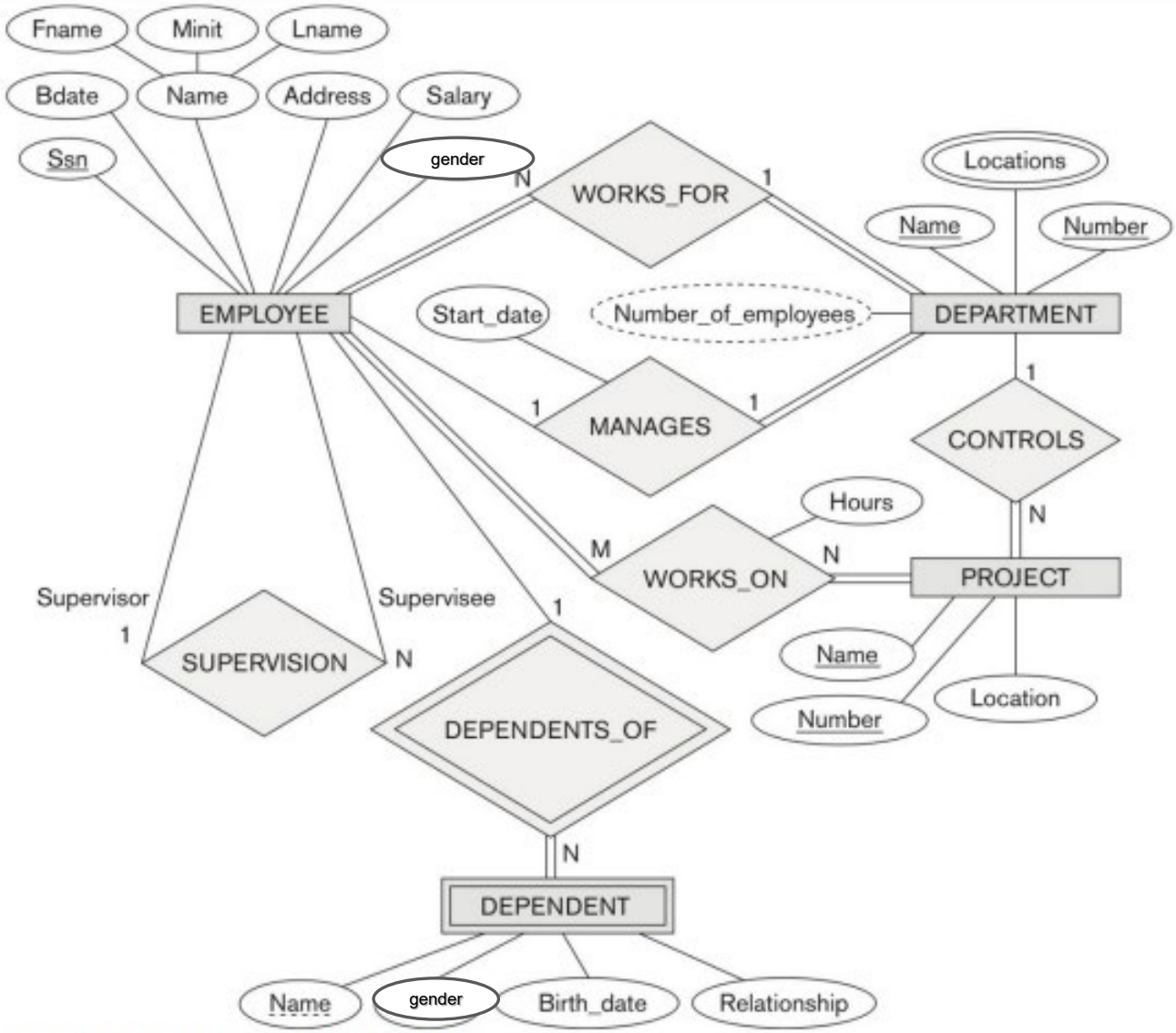
For Chen's notation, one oval inside another.

For crow's foot notation, multi-valued attributes can be represented by enclosing the attribute in {}, e.g., {skills}, {phoneNums}, etc.

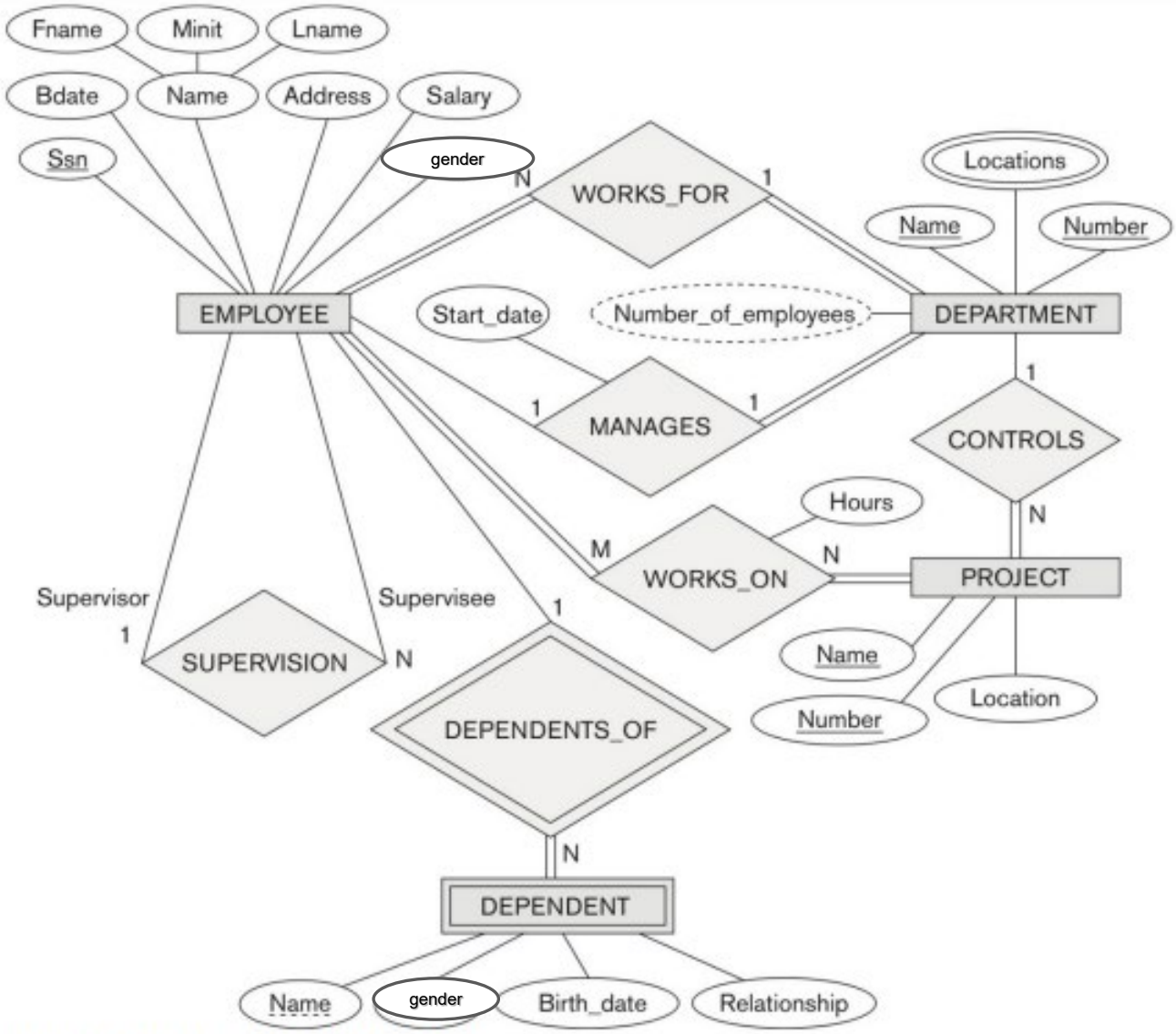
Can you identify .....



menti.com ... list all multi-valued attributes?



menti.com ... list all derived attributes?



NAMING

- The choice of names for entity types, attributes, relationship types and roles is not always straight-forward.
- Should choose names that convey as much as possible the meanings attached to the constructs.
- These names will subsequently be used as table names and attribute names in database so important to choose good names.
- Remember, should not use sql keywords (order, date, etc.)

QUESTION: What attributes might you have for these entities?

Subject/Module

Person

Exam … see menti.com

Bank account

Book … see menti.com

Film

MORE ON ENTITIES: STRONG AND WEAK ENTITIES

Strong: an entity type whose existence **is not** dependent on some other entity type.

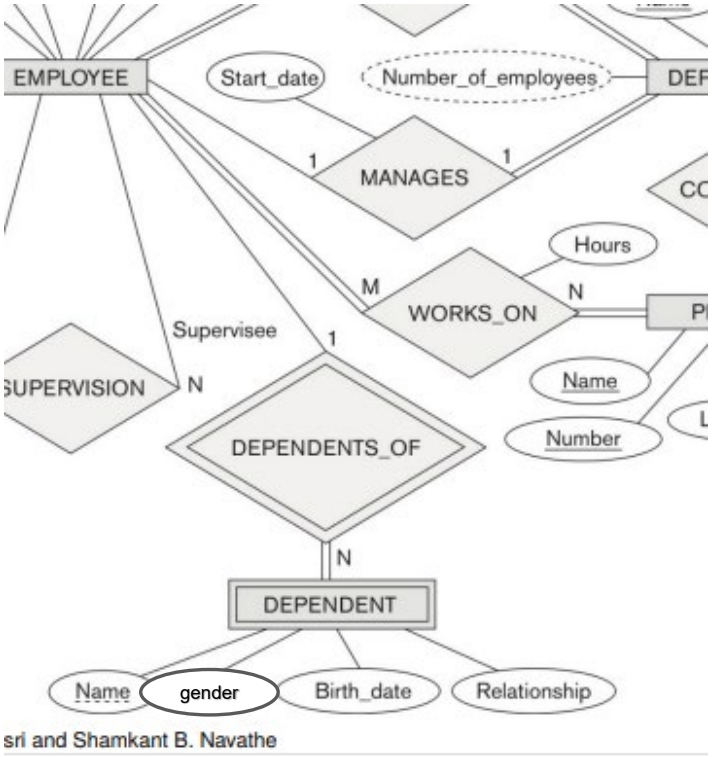
Weak: an entity type whose existence **is** dependent on some other entity type (does not have key attributes of its own)

EXAMPLE:

In the company schema the dependent relation contains data of dependents for each employee.

dependent is a weak entity because two tuples can only be distinguished based on employee SSN.

An alternative would be to have a unique ID for each dependent (e.g. their own SSN) and the dependents could be a strong entity



sri and Shamkant B. Navathe

MORE ON RELATIONSHIPS

Whenever an attribute of one entity type refers to another entity type, some relationship exists.

The degree of a relationship type is the number of participating entity types.

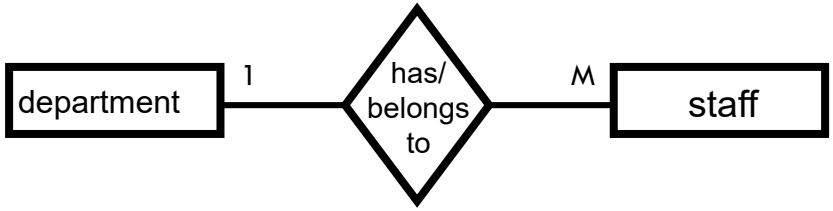
Relationship types may have certain constraints.

NOTATION

For Chen's notation: A Diamond shape is used to name the relationship. 1 and M/N are used for the "1" and "many" sides respectively.

For Crow's foot notation: The crow foot is used as the representation of "many", and one line is used for the representation of "1".

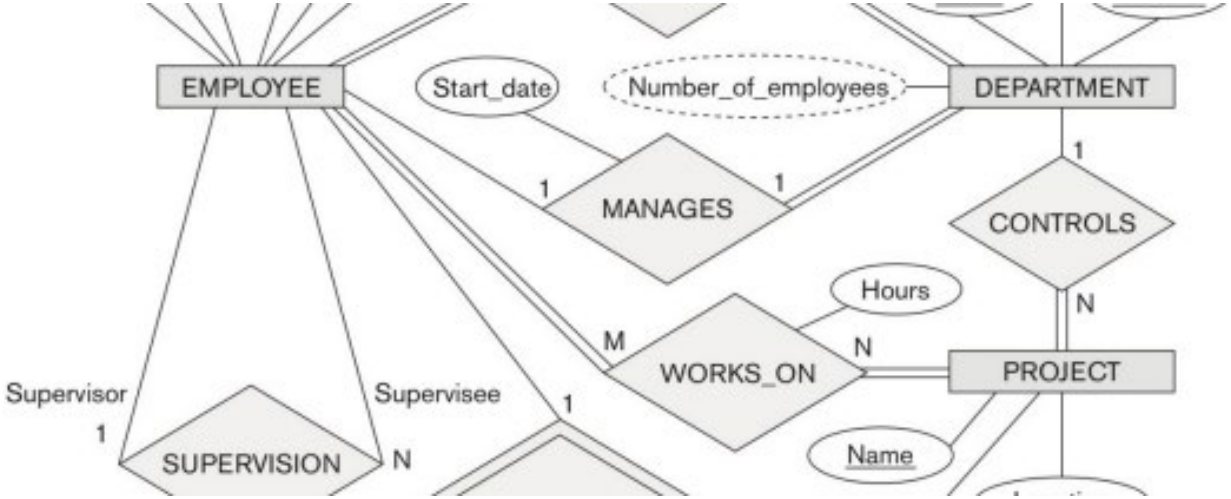
EXAMPLE: A department has many staff



MORE ON RELATIONSHIPS

With Chen's notation, relationships may have attributes

Attributes are drawn "off" the diamond shape of the relationship.



CARDINALITY RATIO

Specifies the number of relationship instances that an entity can participate in.

The possible cardinality ratios for binary relationship types are:

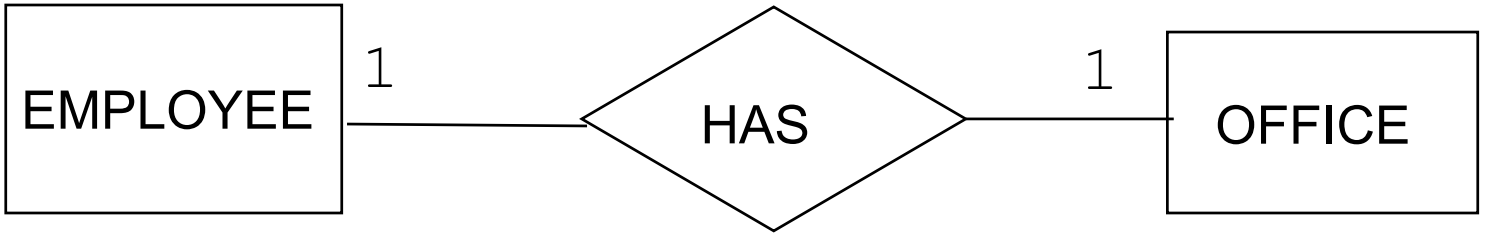
- 1:1, One to One
- 1:N, One to Many
- M:N, Many to Many

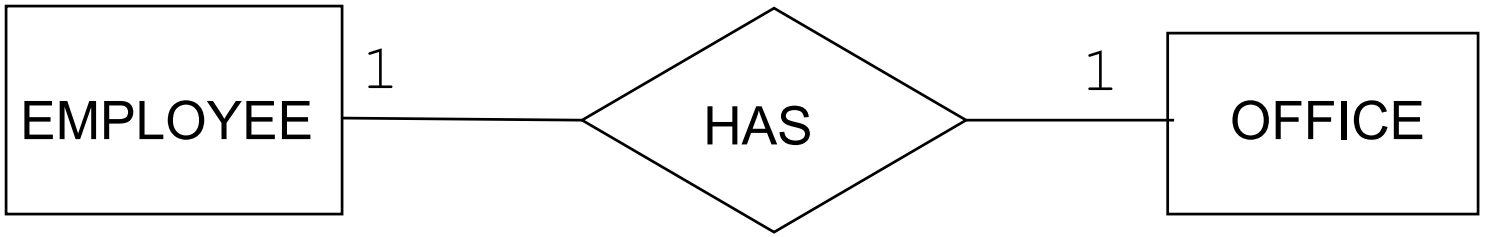
EXAMPLE: 1:1

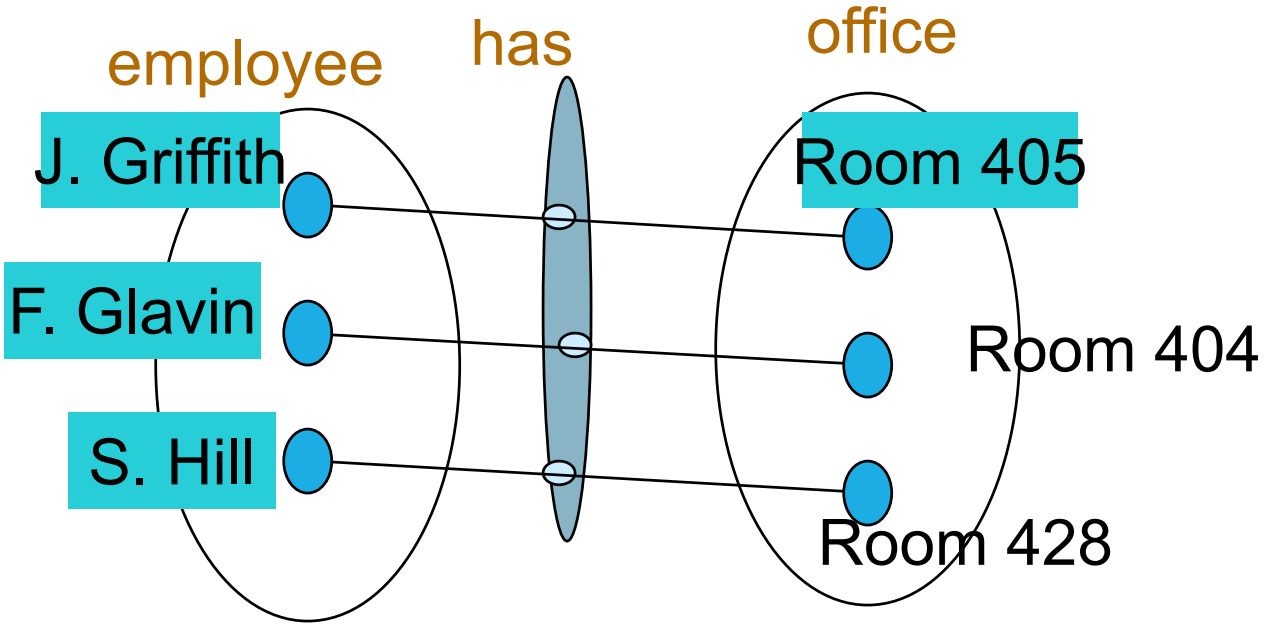
At most one instance of entity A is associated with one instance of entity B

Example: One employee has one office

Chen notation:



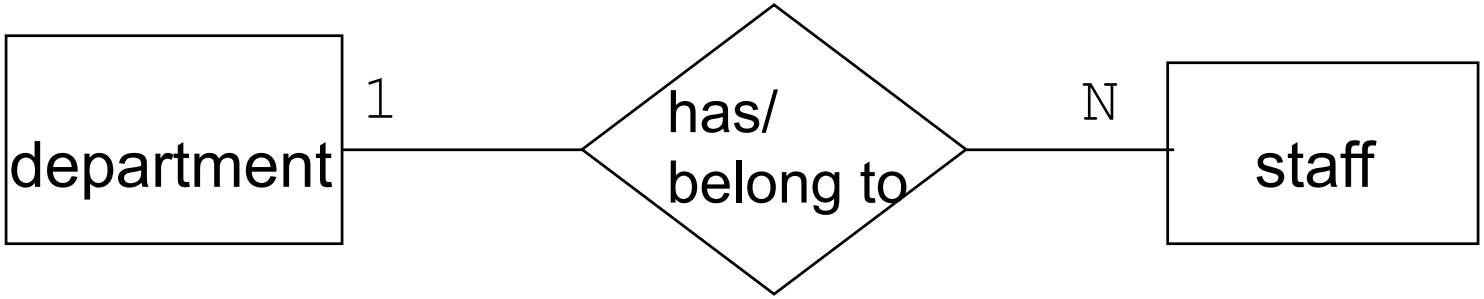

EMPLOYEE
1
HAS
1
OFFICE


employee
has
office
J. Griffith
F. Glavin
S. Hill
Room 405
Room 404
Room 428

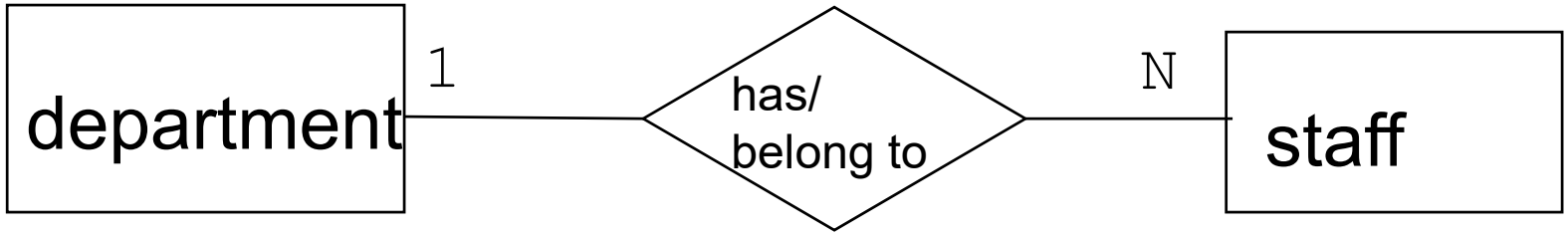
EXAMPLE: 1:N

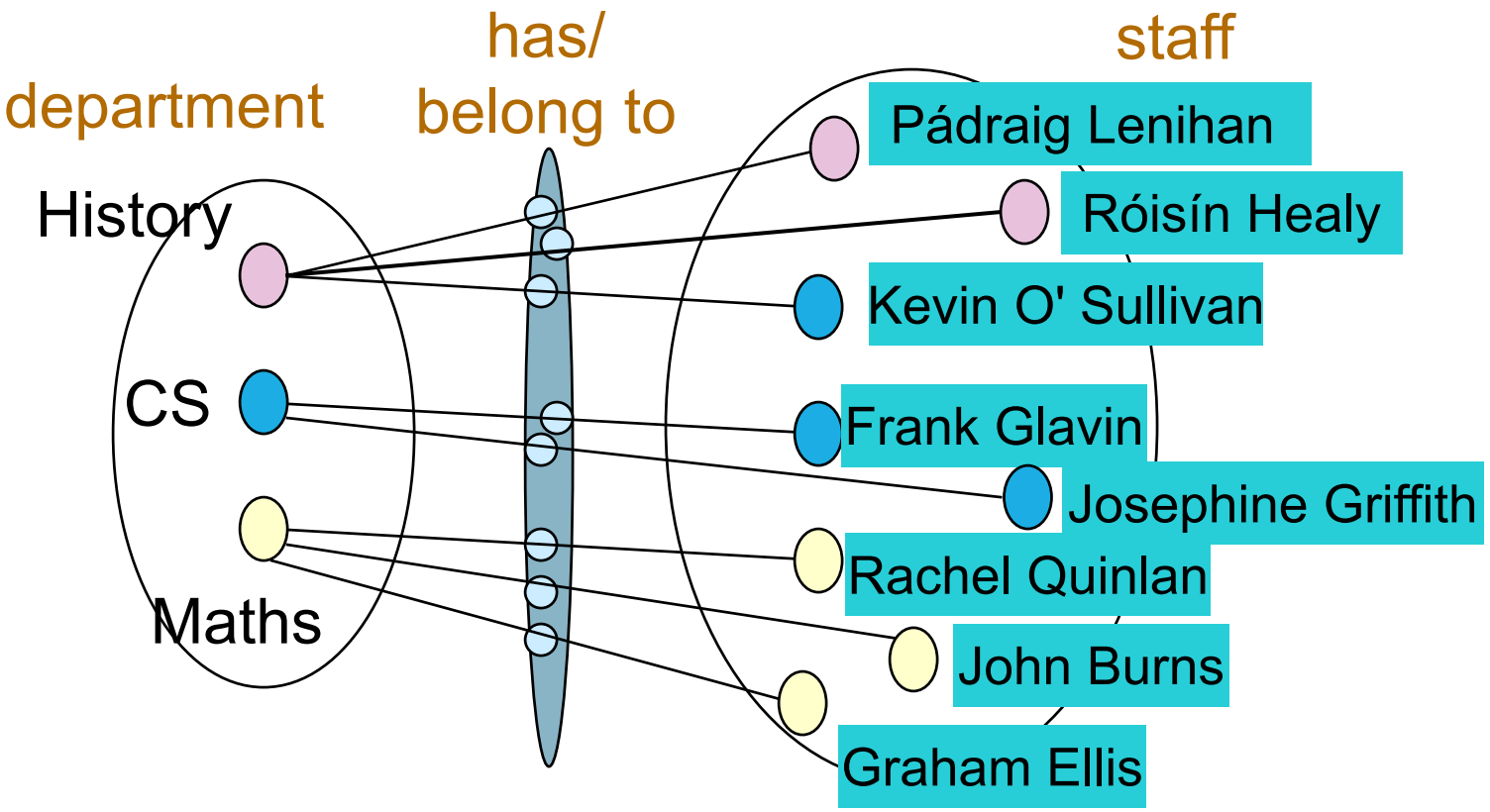
For one instance of entity A, there are 0, 1 or many instances of entity B

Chen Notation:



Crow's foot notation:

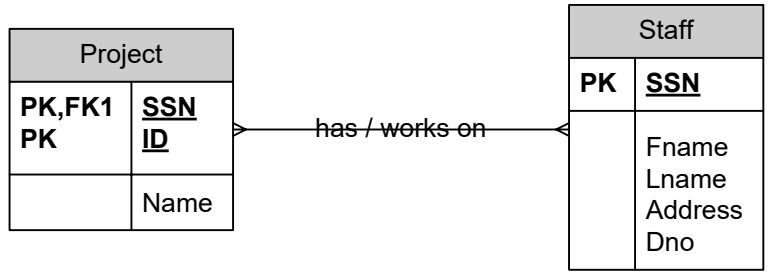

department
1
has/
belong to
N
staff

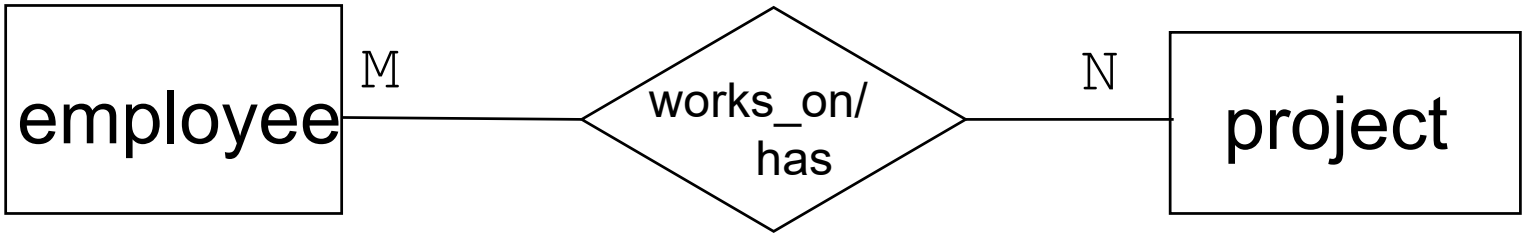

department
has/
belong to
staff
History
CS
Maths
Pádraig Lenihan
Róisín Healy
Kevin O' Sullivan
Frank Glavin
Josephine Griffith
Rachel Quinlan
John Burns
Graham Ellis

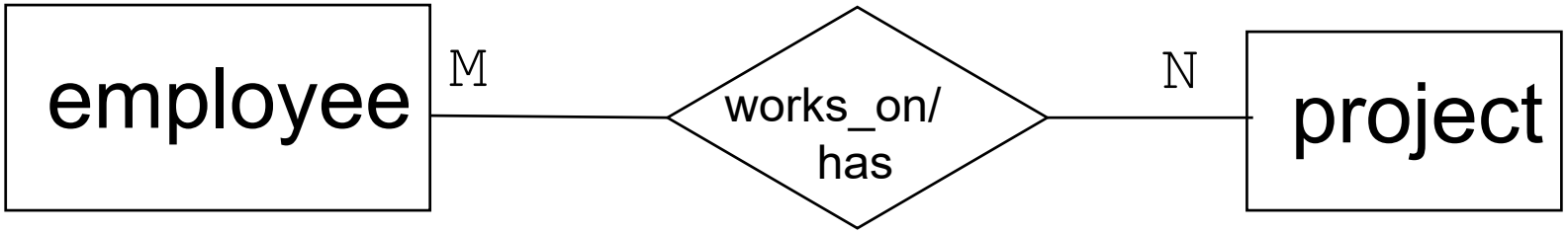
EXAMPLE: M:N

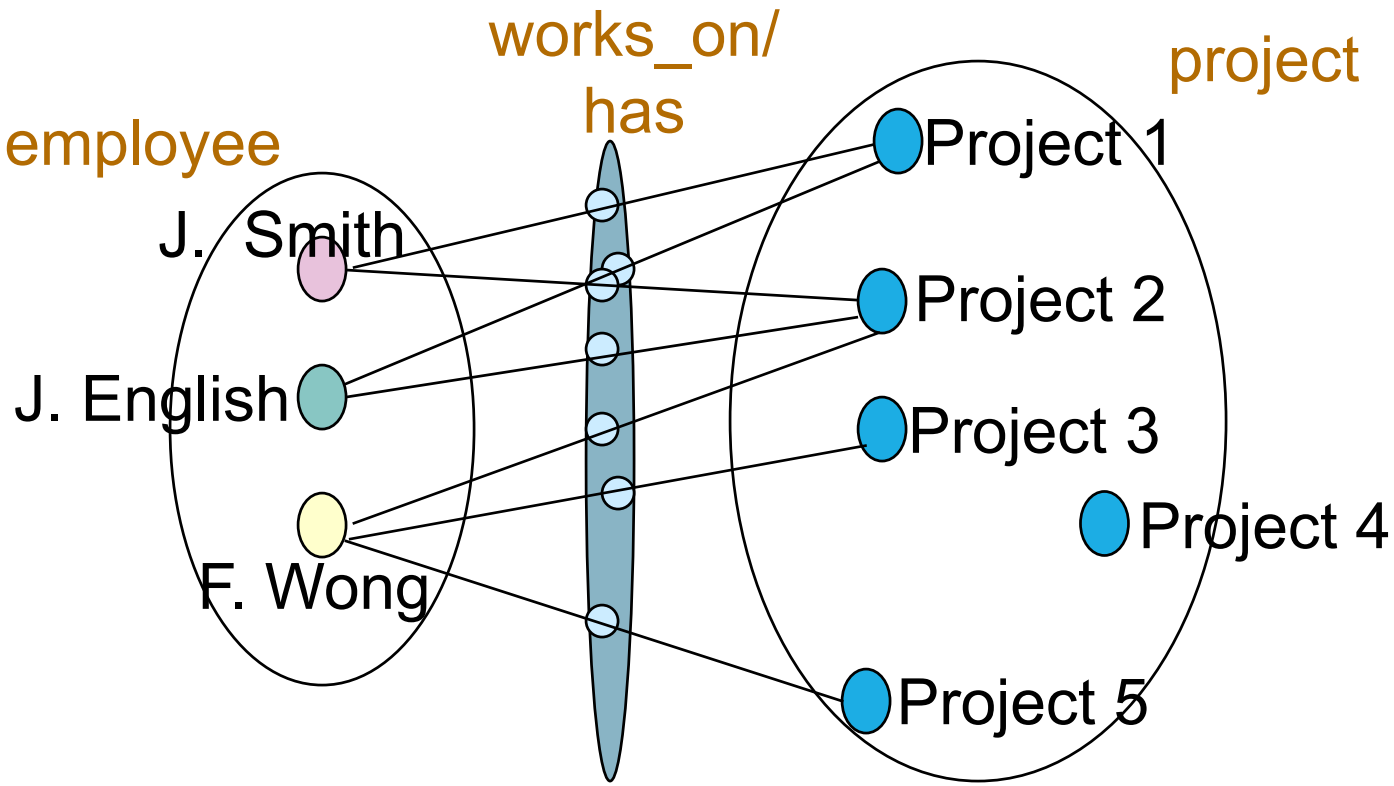
For one instance of entity A, there are 0, 1 or many instances of entity B and

For one instance of entity B, there are 0, 1 or many instances of entity A






employee
M
works_on/
has
N
project


works_on/
has
project
employee
Project 1
J. Smith
Project 2
J. English
Project 3
Project 4
F. Wong
Project 5

ASIDE: Structural constraints on relationships

Often we may know the min and max of the cardinalities

- e.g., limit to number of books which can be borrowed

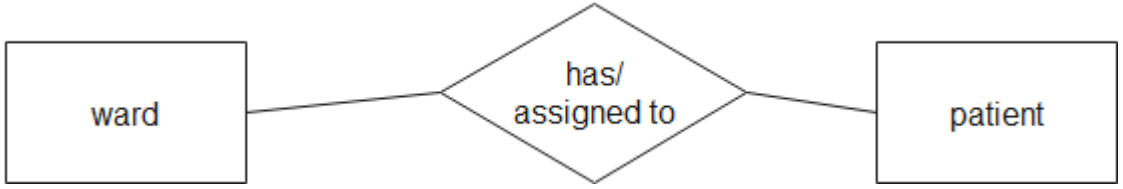
Structural constraints specify a pair of integer numbers *(min, max)* for each entity participating in a relationship

Examples: (0, 1) ,(1,1), (1, N), (1, 7)

We will not model this in our examples

CASS QUESTION – See menti.com

In a hospital, patients are assigned to wards; wards have patients. What is the cardinality of the relationship?



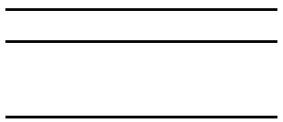
TOTAL AND PARTIAL PARTICIPATION

Total Participation: all instances of an entity must participate in the relationship, i.e., **every** entity instance in one set **must** be related to an entity instance in the second set via the relationship.

Partial Participation: some subset of instances of an entity will participate in relationship, but not all, i.e., **some** entity instances in one set are related to an entity instance in the second set via the relationship.

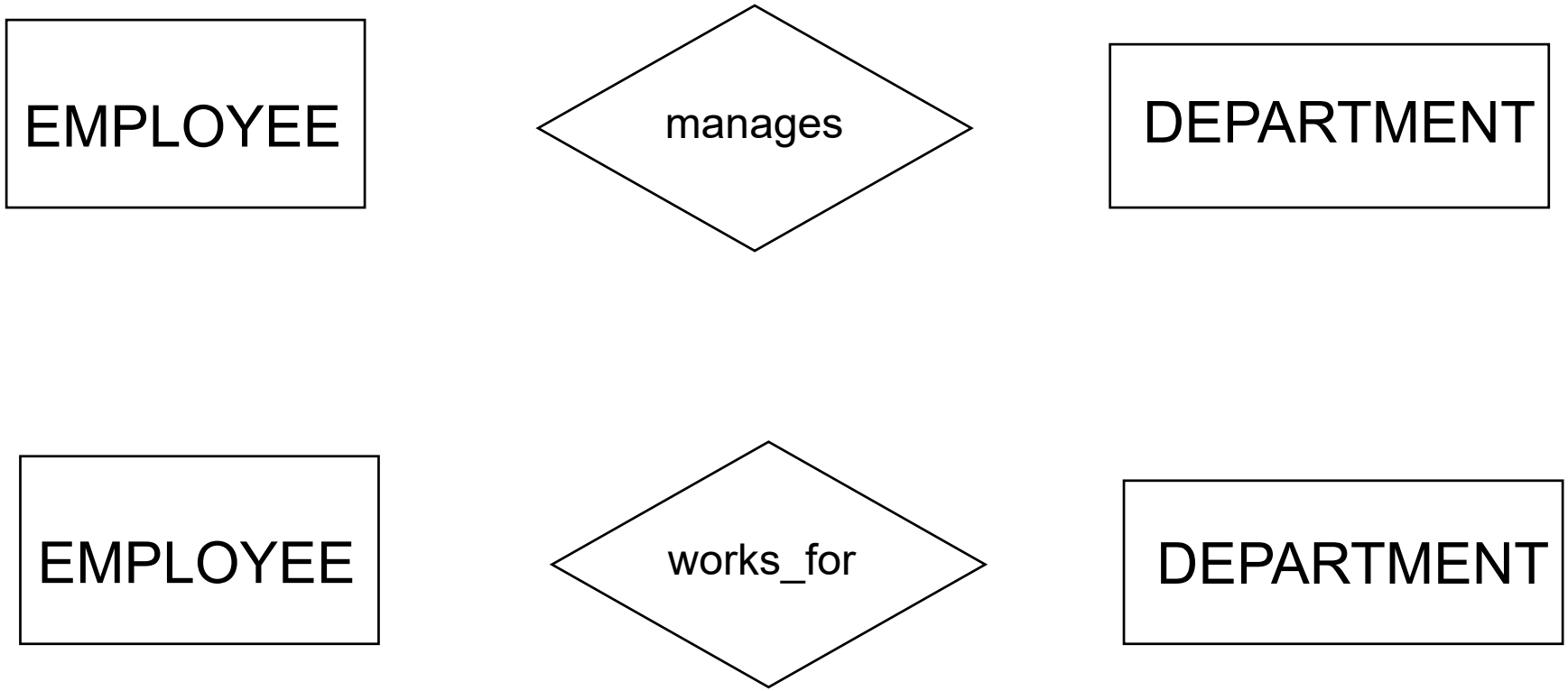
NOTATION FOR PARTICIPATION
CHEN'S NOTATION

- Double parallel lines for Total Participation
- Single line for Partial Participation

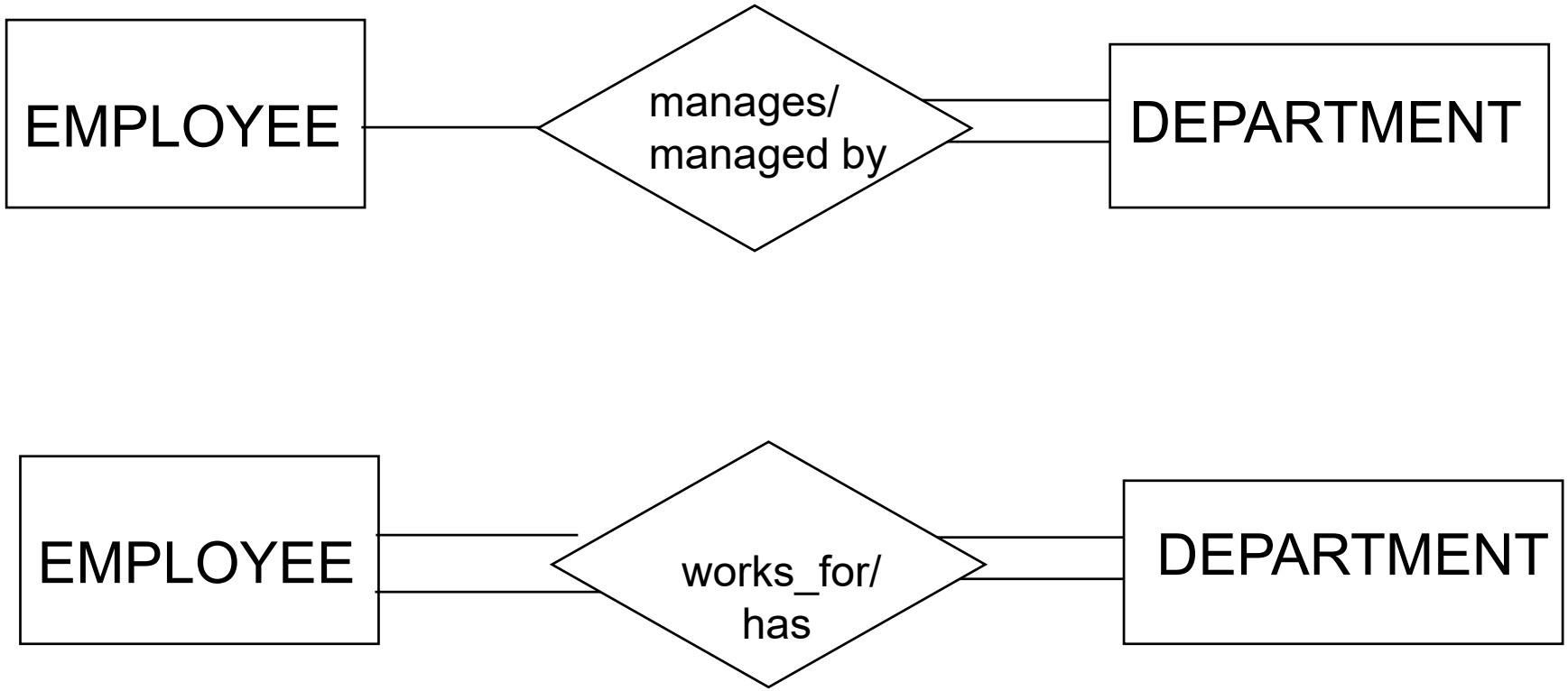


- In both cases, lines drawn from the participating entity to the relationship (the diamond) to indicate the participation of instance from that entity in the relationship

EXAMPLES


EMPLOYEE
manages
DEPARTMENT
EMPLOYEE
works_for
DEPARTMENT

EXAMPLES:
Total and partial participation



NOTATION FOR PARTICIPATION
CROW'S FOOT NOTATION

Use the idea of **Ordinality/Optionality**

- Optionality of 0: if an entity A has partial participation in a relationship to entity B then this means A is associated with 0 or more of the other entity so optionality sign goes beside B.
- Optionality of 1: if an entity A has full participation in a relationship to entity B then this means A is associated with at least 1 or more of B so optionality sign goes beside B.

(and vice versa when looking at participation of B in relationship)

CROW'S FOOT NOTATION

Bar for Optionality of 1: |

Circle (or 'o') for Optionality of 0 ○

In Crow's foot notation, there is no diamond so there is a direct relationship line between the entities. On this line:

- The optionality drawn beside entity A refers to how an instance of entity B is related to entity A.
- That is, whether B can be involved partially (0) or not (1)

Example in Following *Right to Left* Relationships:

is of 0 or more

is of 1 or more

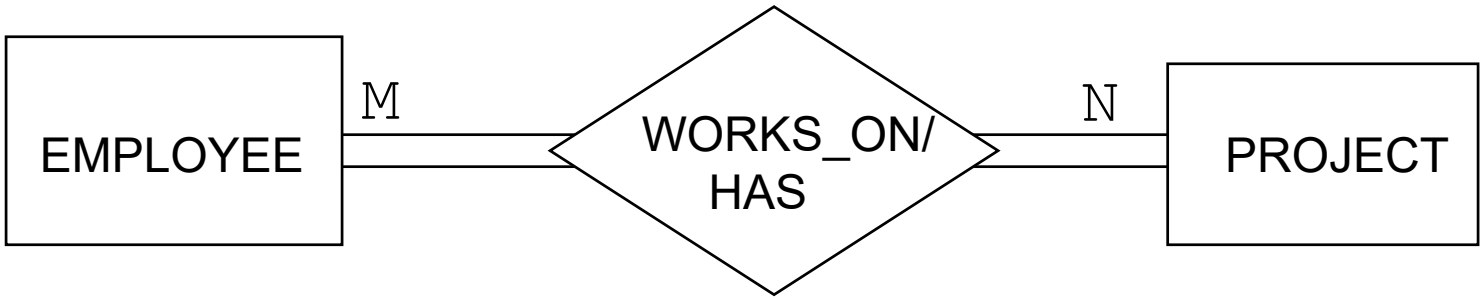
is of 1 and only 1

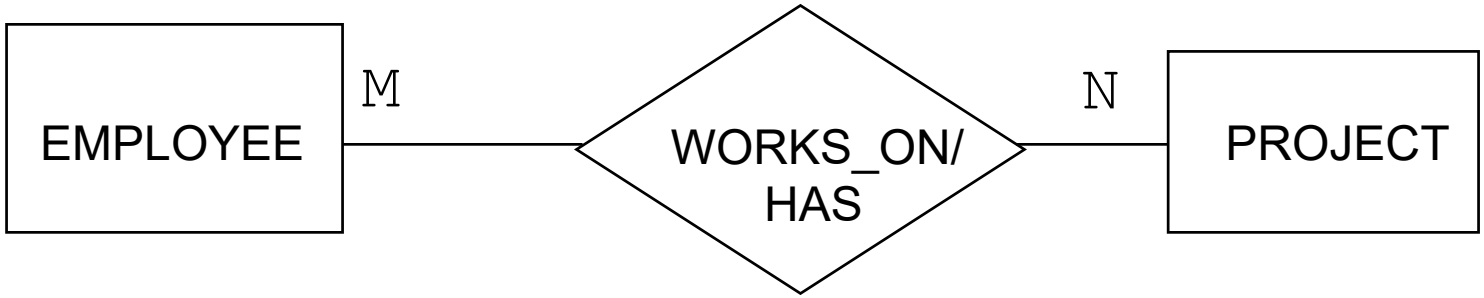
is of 0 or 1

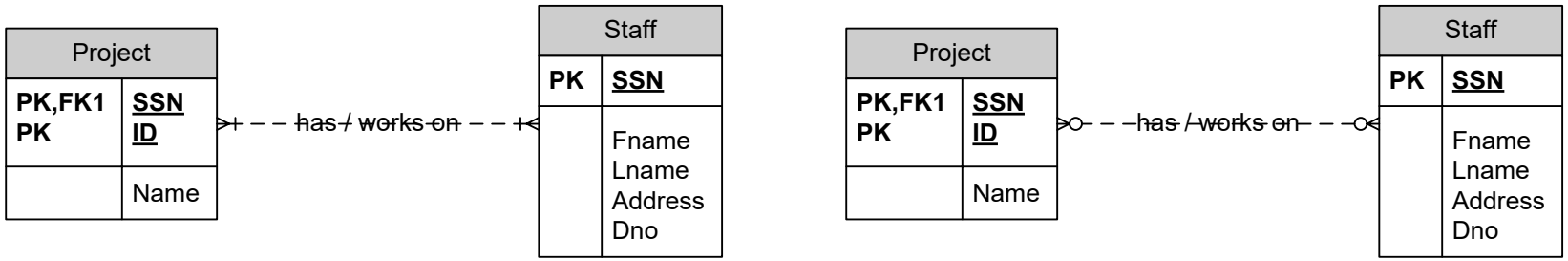
WHICH IS CORRECT FOR THIS RELATIONSHIP?

Total or partial participation?

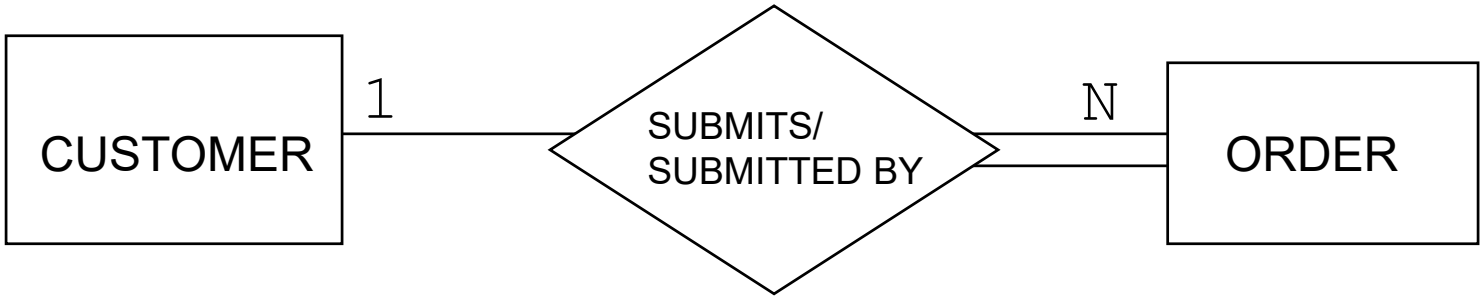
See menti.com

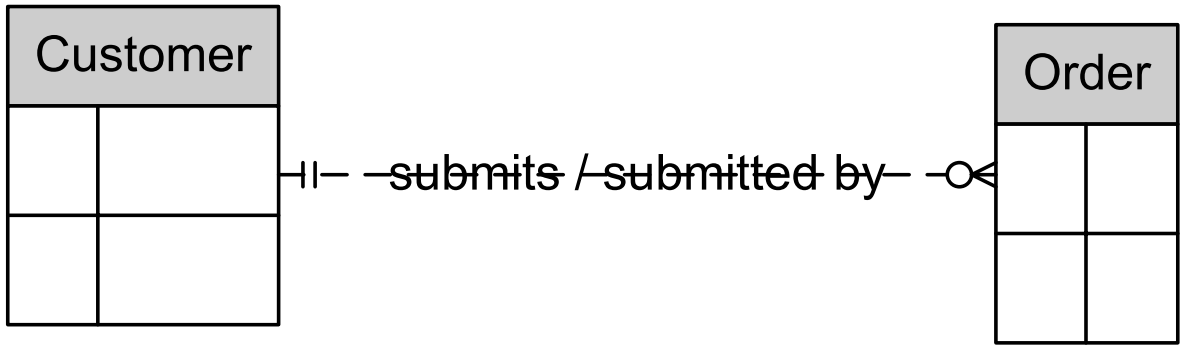






Describe the relationship in words in the following: See menti.com



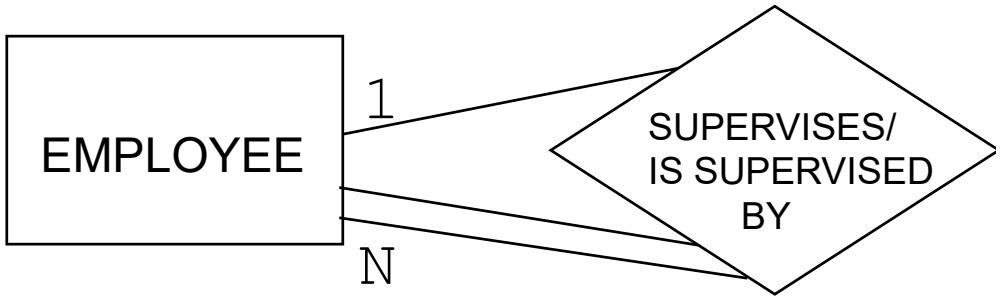


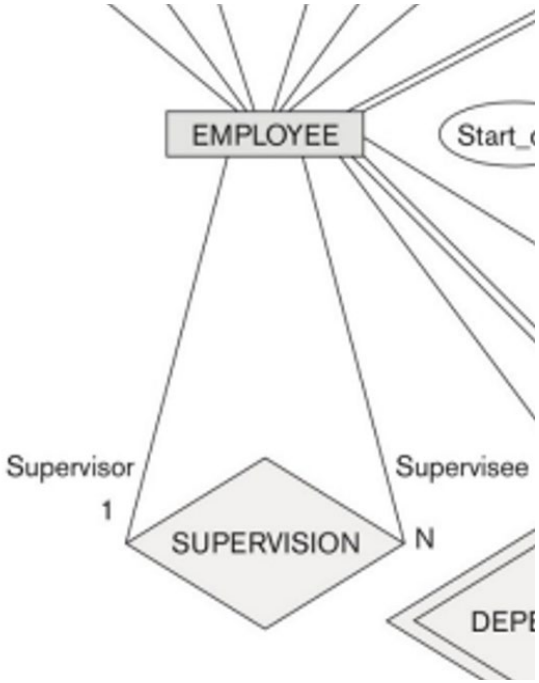
See menti.com

Describe the relationship in words in the following:

Does it look correct?

How would you fix it?





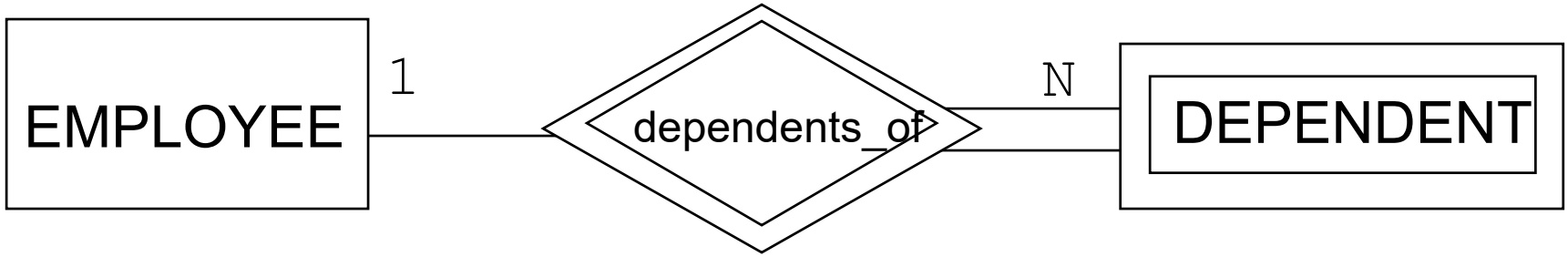
See menti.com
What is the relationship between these entities?

- Cars and people
- Students and library seats
- Students and subjects
- Exams and Locations
- Customers and Bank accounts
- Books and Authors
- Cinema and films/movies

NOTE:

A weak entity type always has a total participation constraint

Need to show the “identifying relationship”

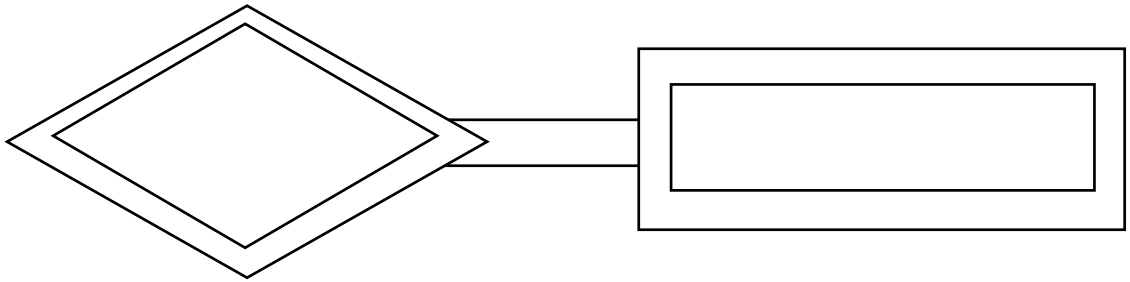


CHEN'S NOTATION FOR WEAK ENTITY

Double rectangle for Entity

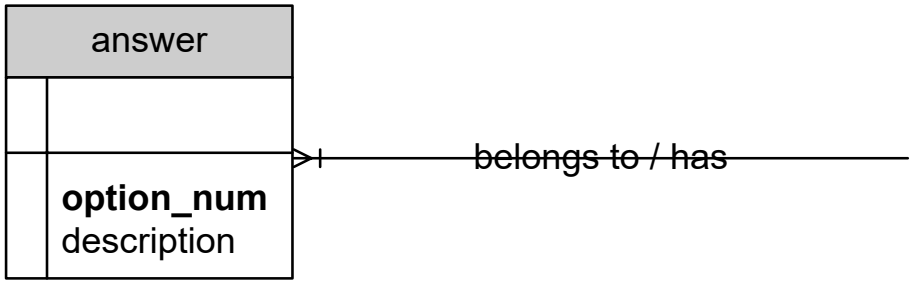
Double diamond for Relationship

Weak entity has full participation in the relationship



CROW'S FOOT NOTATION FOR WEAK ENTITY:

- Can represent the Weak Entity as a normal entity but do not choose any attributes as primary keys.
- For an attribute that partially determines the entity instances, choose the 'required' option
- Represent the relationship between entities with a **solid** line (usually)
- This indicates it is an "identifying" relationship



In general, with entities:

There may be two valid solutions, one with a weak entity and one without.

There is not a huge difficulty if you do not identify weak entities in a solution as long as all entities have **primary attributes.**

May be slightly non-optimal in terms of introducing an additional primary key that is not needed but not a huge problem for us at this level.

Entities or multi-valued attributes?

Sometimes it may not be clear whether something should be modelled as a multi-valued attribute or an Entity.

Both may be equally correct as long as you have represented all the information you were asked to.

When you map either case to tables in a database you might see very little difference between the two approaches.

CLASS EXAMPLE 1

A database is to be created to hold information on lecturers, departments, courses and modules.

Lecturers are associated with only one department. Each lecturer in addition has an associated staff id, title, name, office number and building. Each lecturer teaches a number of modules and a number of lecturers may teach one module.

Each module has an associated unique code (e.g. CT230), name, semester taught, semester examined, ECTs and zero or more prerequisites (which are modules). For example, CT103 and CT102 may be a prerequisite for CT2101.

Each module is part of one or more course instances (e.g. 2BA, 2BCT, 2BFS, 3BP). Each course has an associated name and code.

Each course is controlled by a department, and a department can control a number of courses. Each department has an associated name, and may have a number of different locations; each department has one head of department.

CLASS QUESTION:

Using Chen's notation, create an ER model to accurately model **the above information**. Show all entities, relationships, attributes, cardinalities, and total and partial participations. State any assumptions you make.

STEPS:

Identify entities.

Identify relationships between entities.

Draw entities and relationships.

Add attributes to entities (and relationships if appropriate).

Add cardinalities to relationships.

Add participation constraints (total or partial) to relationships.

Check all entities have primary keys identified.

MAPPING ER MODELS TO TABLES IN THE RELATIONAL MODEL

Once you have your ER diagram you now need to convert this into a set of tables so that you can implement this in a relational model (e.g. as MySQL tables using `CREATE TABLE` commands)

This stage is called **Mapping ER Models to Tables in the Relational Model** and it specifies a set of rules that must be followed in a certain order.

The rules specified here are based on Chen's notation.

STEPS … Mapping ER models to tables in the relational model

1. For each entity create a table R that includes all the **simple** attributes of the entity.

2. For strong entities, choose a key attribute as primary key of the table.

STEPS … Mapping ER models to tables in the relational model

3. For weak entities R, include as foreign key attributes of R the primary key attributes of the table that corresponds to the owner. The primary key of R is a combination of the primary key of owner and the partial key of the weak entity type.

The relationship of the weak and strong entity is generally taken care of by this step

STEPS *CTD*.... mapping ER models to tables in the relational model

4. For each binary 1:1 relationship, identify entities S and T that participate in relation.

- If applicable, choose the entity that has total participation in the relation. Include as foreign key in this table the primary key of other relation. Include any attributes of the relationship as attributes of chosen table.
- If both entities have total participation in the relationship, you can choose either for the foreign key and proceed as above or can map 2 entities, and their associated attributes and relationship attributes into 1 table.

STEPS *CTD*.... mapping ER models to tables in the relational model

5. For each binary 1:N relationship, identify the table S that represents the N-side and T the table that represents the 1-side.

- Include as a foreign key in S the primary key of table T such that each entity on the N-side is related to at most one entity instance on the 1-side. Include any attributes of the relationship as attributes of S.
- For recursive 1:N relationships, choose the primary key of the table and include it as a foreign key in the same table (with a different name).

STEPS *CTD*.... mapping ER models to tables in the relational model

6. For each M:N relationship, create a new table S to represent the relationship.

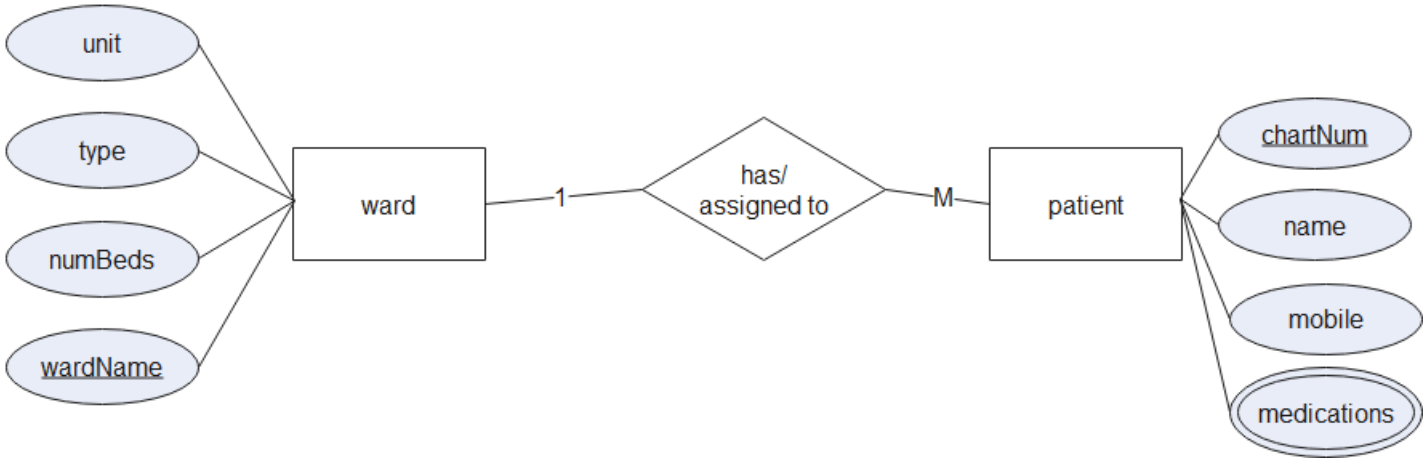
- Include as foreign key attributes in S the primary keys of the tables that represent the participating entity types – their combination will form the primary key of S. Also include in S any attributes of the relationship.
- For a recursive M:N relationship, both foreign keys come from the same table (give different name to each) and become the new primary key.

STEPS *CTD*.... mapping ER models to tables in the relational model

7. **For each multi-valued attribute A of an entity S,** create a new table R. R will include:

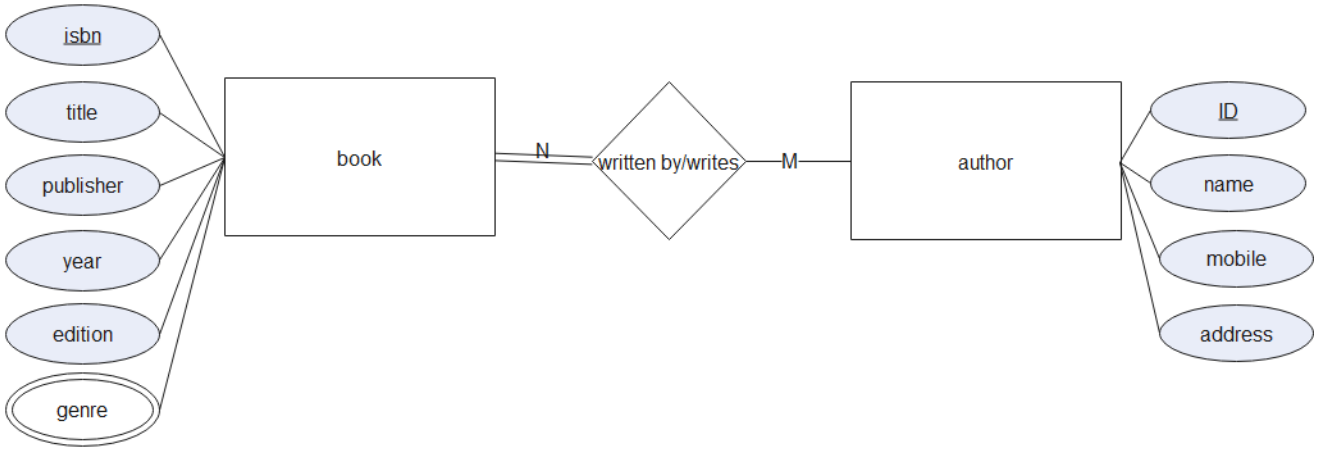
- an attribute corresponding to A,
- primary key of S which will be a foreign key in table R. Call this K.
- primary key of R is a combination of A and K

Map each of the following to tables in the relational model:
wards and patients



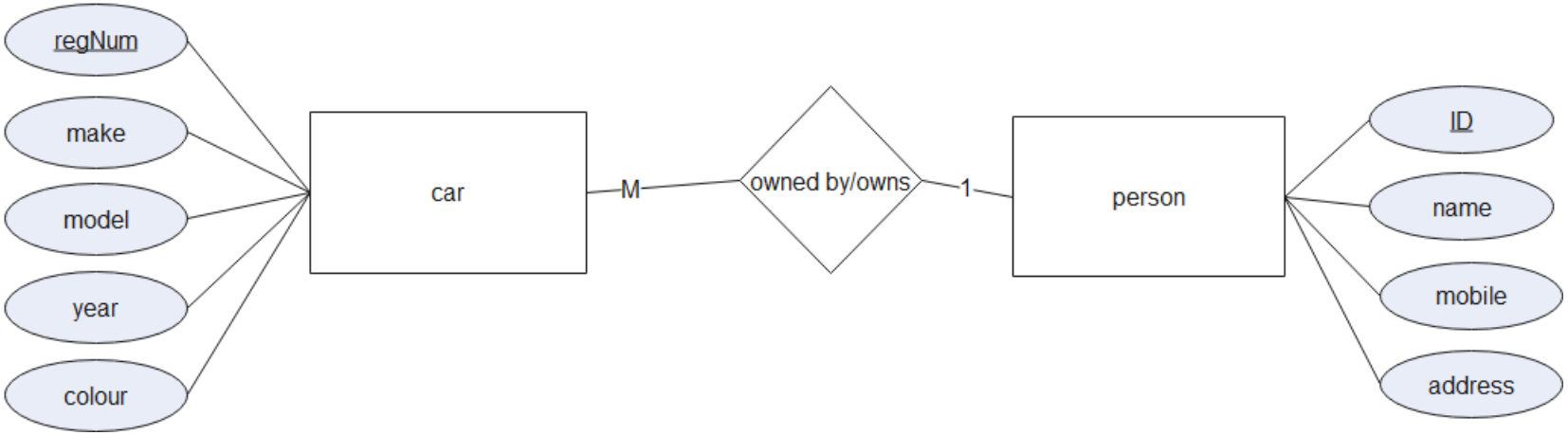
Map each of the following to tables in the relational model:

authors and books

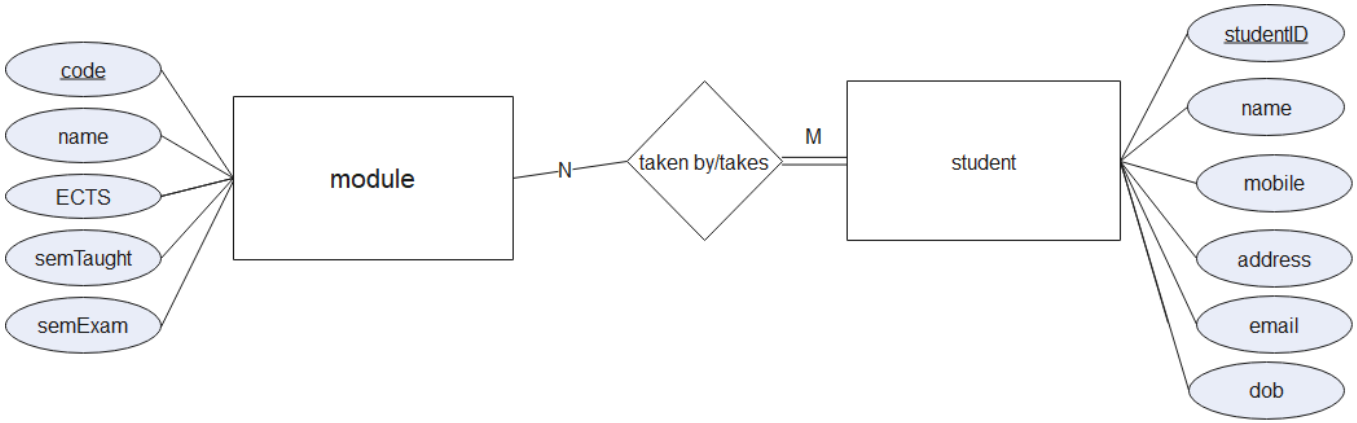


Map each of the following to tables in the relational model:

cars and people



Map each of the following to tables in the relational model:
modules and students



CLASS WORK: Map the University model created (Example 1) to tables in the relational model

PROBLEM SHEET 4

An Irish holiday home rental company wishes to create an online database system to maintain information on home owners who own holiday houses which the rental company rents on their behalf; customers who rent the holiday homes, and the rental agreements. The data which should be stored is as follows:

Details stored on holiday houses are: a unique ID for each house, the address of the house (town, county and Eircode), the number of bedrooms and bathrooms in the house and the maximum number of people the house will accommodate. Two price details should be stored: low-season price per night and high-season/weekend price per night. In addition a short description of the house amenities and surrounding amenities should be stored.

Each house is owned by one home owner. A home owner may own many houses. Details stored on the home owners are: a unique id, a username and password to login to the system, their name, address and telephone number and their email address.

Customers can book one or more houses and a house can be booked many times. Details held on customers are: unique ID, customer name, address, email address and phone number.

Details held on a booking are the dates the booking begins and ends, and the number of people wishing to stay in the house as part of the booking. Any entered bookings must be confirmed by a company employee (via phone or email). When the confirmation takes place, data should be stored to indicate that the confirmation has taken place and to indicate the amount of money paid as a deposit. This database does not currently hold any information on the check-in process and the payment of the balance due.

SUMMARY:

Important to Know:

- Basic definitions of entity, relationship, attribute (and different types), cardinality and participation for Chen and Crow's foot notation.
- Create ER Model (in Chen's notation)
- Map from ER model in Chen notation to set of tables with associated primary and foreign keys.

Common Errors:

- Missing Primary Keys for Entities.
- Missing cardinalities in Relationships.
- Only mapping entities to tables; not mapping relationships or multi-valued attributes.