Lancaster University

Welcome to WEEK 2….

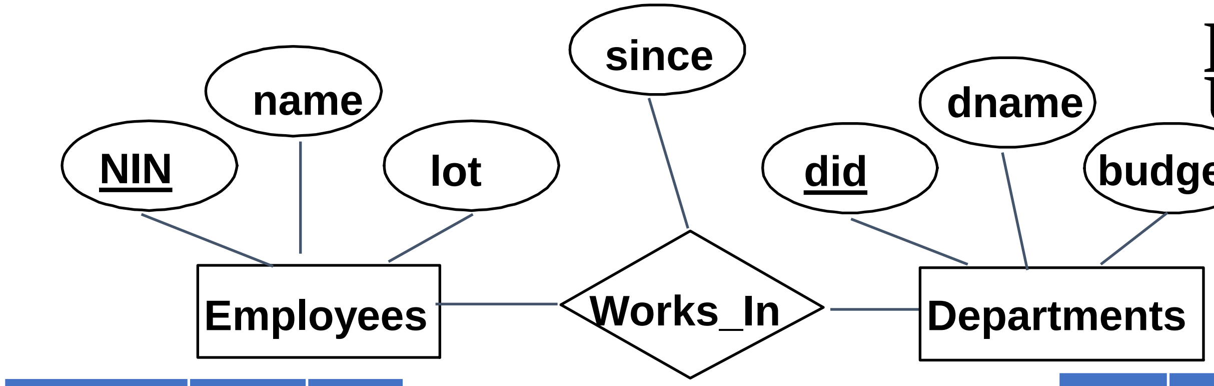
We will start at 17:00…

FROM YOU…

What did we do in last lecture?

- Entities
- Relating entities: *Relations*
- Mapping between entities
  - 1 to 1 relations.
  - 1 to many relations.
  - Many to one relations.
  - Many to many relations.

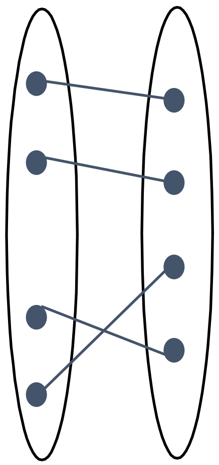
- Consider the works_in relationship
- If an employee can work in at most one department and a department can have multiple employees
- What type of relationship is that?



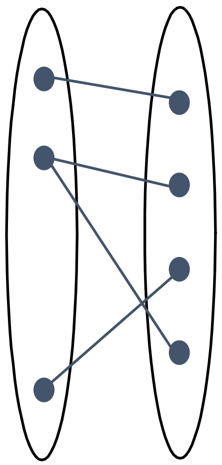
<u>NIN</u>	Name	Lot
8754702	Tom	75
6820937	Uraz	545
2313927	Nick	12

since	did	NIN
2021	3	8754702
2022	3	6820937
1992	3	2313927

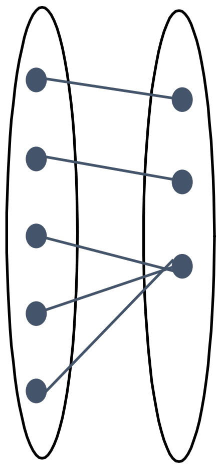
<u>did</u>	dname	budget
1	Sales	75
3	Man.	545
2	Tech.	12



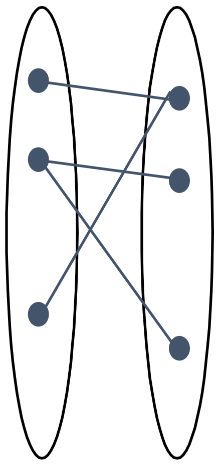
1-to-1



1-to Many

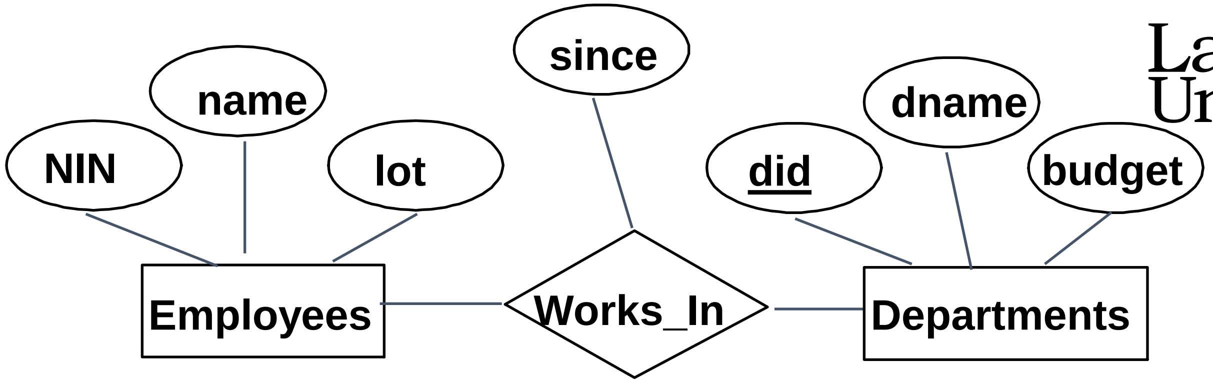


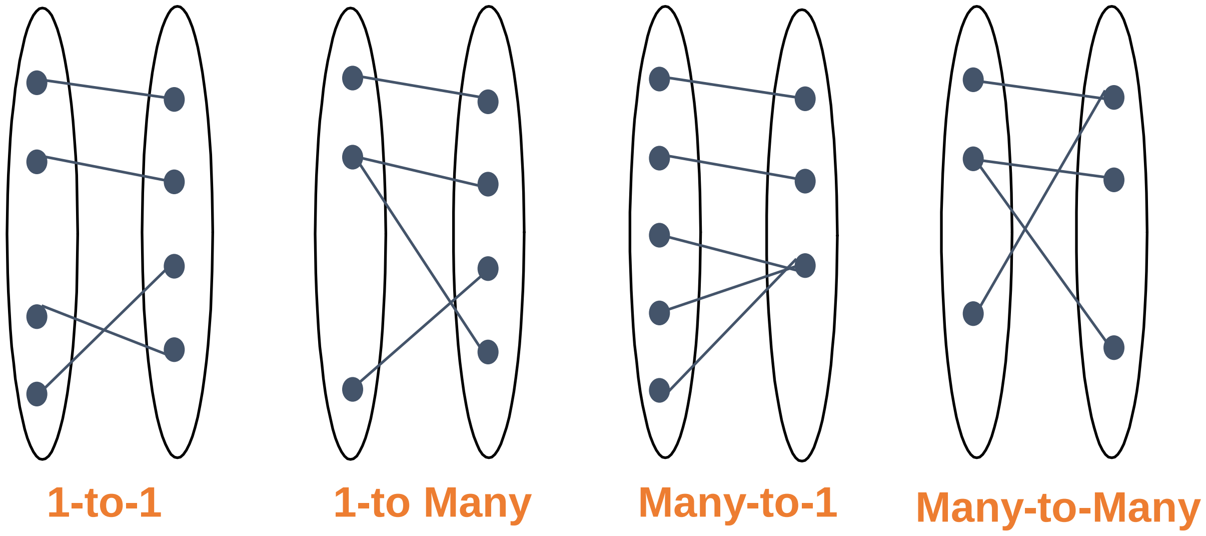
Many-to-1



Many-to-Many

- Consider the works_in relationship
- If an employee can work in several departments and a department can have multiple employees
- What type of relationship is that?

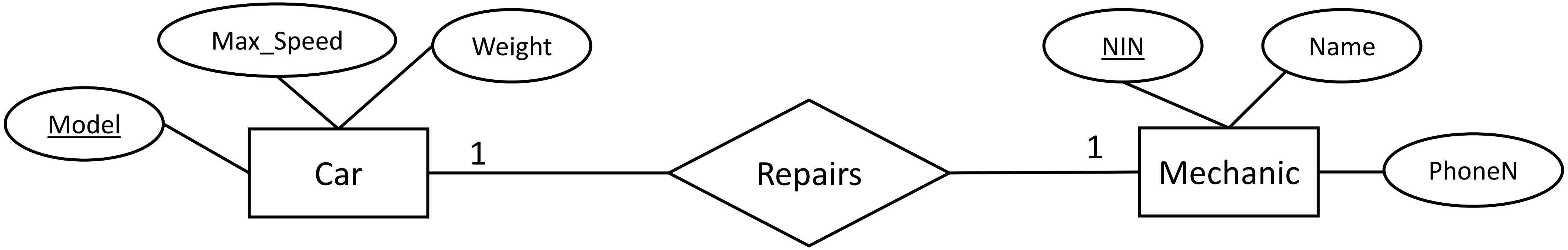




How do we encode cardinality into ER diagrams?

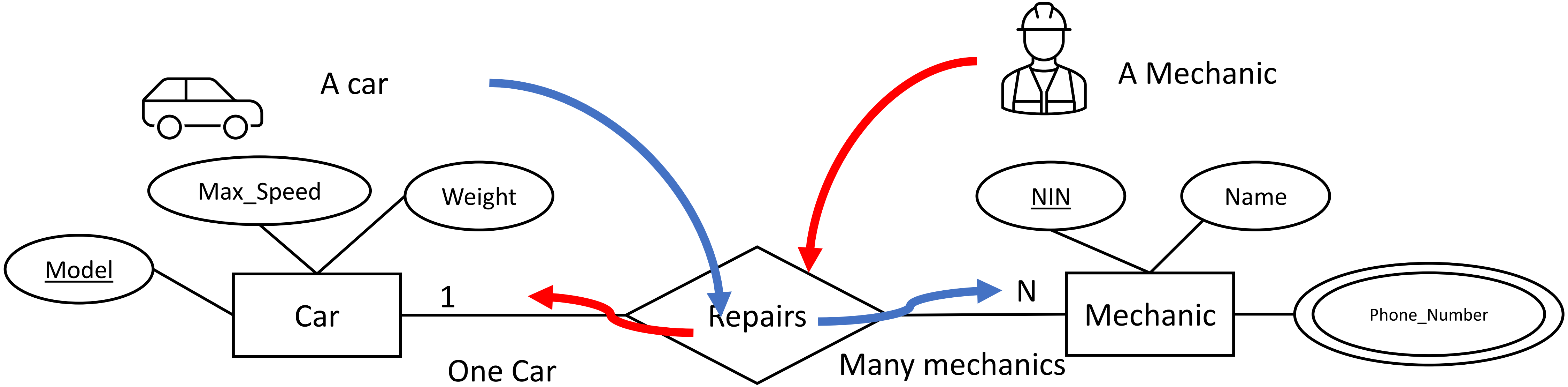
- We use **Chen’s** notation (Always look at the opposite direction)
- 1:1 is for one-to-one
- 1:N is for one-to-many
- N:1 is for many-to-one
- N:M is for many-to-many.

1:1


Max_Speed
Weight
Model
Car
1
Repairs
1
Mechanic
NIN
Name
PhoneN

Car Mechanic SIMULATOR 2021

1:N One to many



A car

A Mechanic

Max_Speed

Weight

NIN

Name

Model

Car

1

Repairs

N

Mechanic

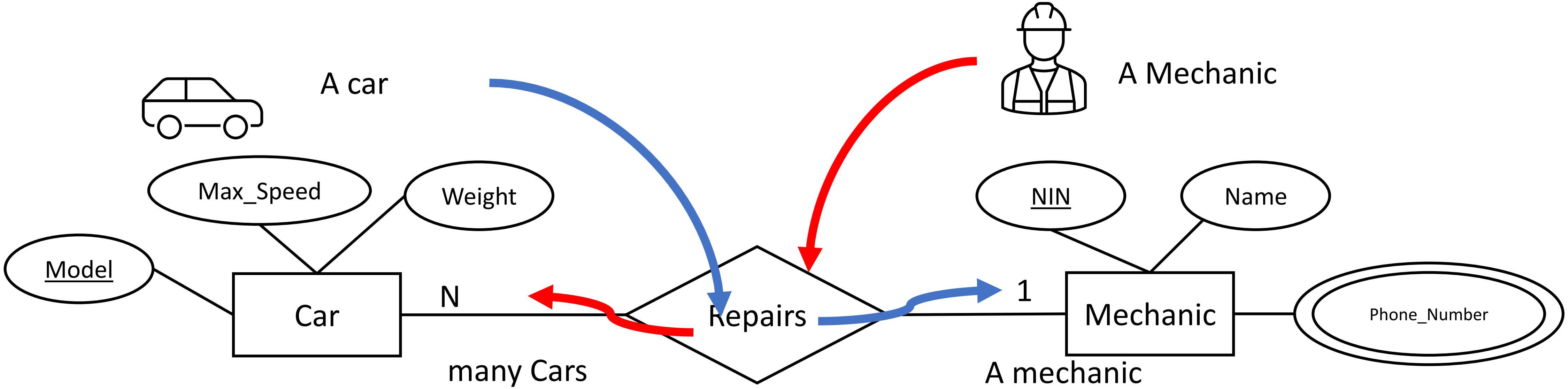
Phone_Number

One Car

Many mechanics

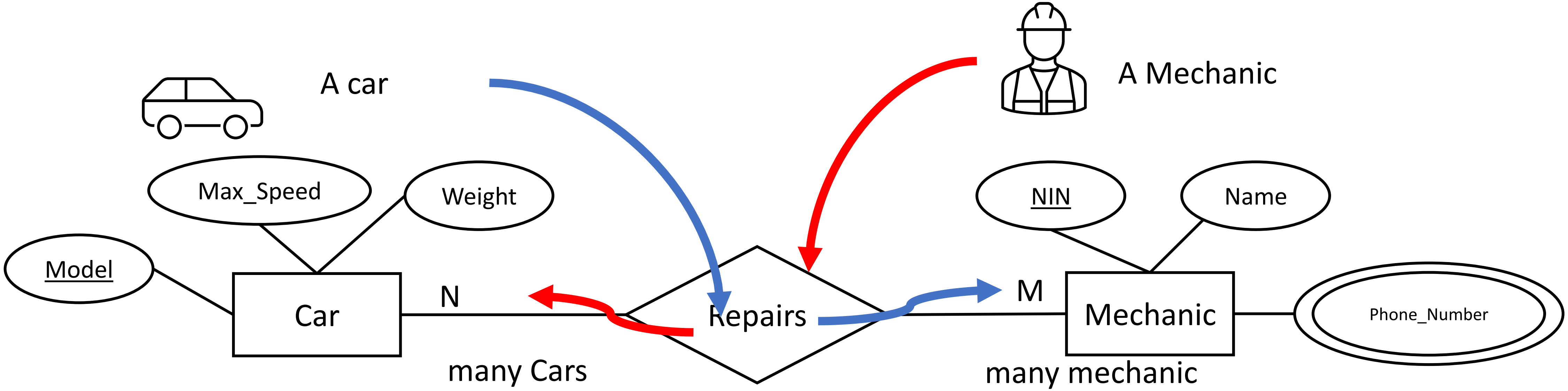
A car can be repaired by many mechanics.
A mechanic can repair one car.

N:1 Many to one



A car can be repaired by a mechanic.
A mechanic can repair many cars.

N:N many to many



A car

A Mechanic

Max_Speed

Weight

NIN

Name

Model

Car

N

Repairs

M

Mechanic

Phone_Number

many Cars

many mechanic

A car can be repaired by many mechanics.
A mechanic can repair many cars.

Key concepts of ER: Cardinalities.

IMPORTANT CONTENT!!

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

NIN	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

Given an entity (E) from one entity set, what is the relation of this entity with the entities in the other entity sets?
Can more than one mechanic repair BMW 3.21?
Can Tom repair more than one type of car?

Key concepts of ER: Participation Constraints.

IMPORTANT CONTENT!!

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

NIN	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

Given an entity (E) from one entity set, what is the relation of this entity with the entities in the other entity sets?

Can more than one mechanic repair BMW 3.21?

Can Tom repair more than one type of car?

Can there be a mechanic who does not know how to repair a car?

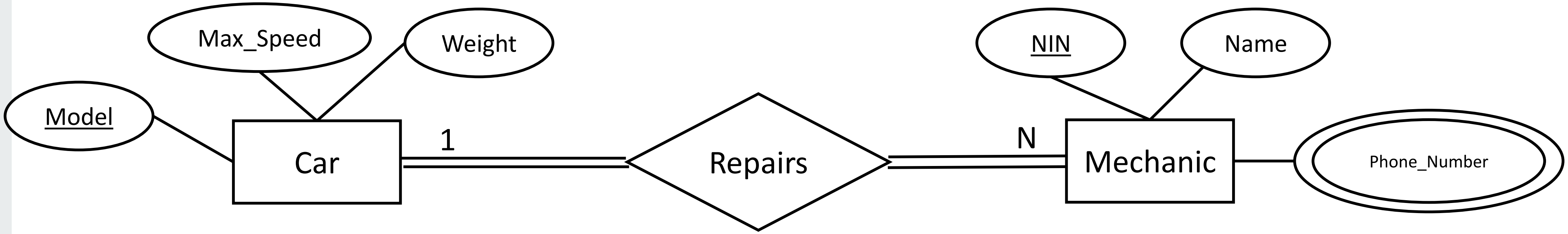
Can there be a car that cannot be repaired?

Participation constraints

- Can there be a mechanic that cannot repair a car?
- If not, we need to state that there is a **Total Participation.**
  - Total Participation implies that if an entity exists in an entity set, it must relate with at least one entity in the other entity set.
  - A **double line** identifies total participation.
- If so, then it is **Partial Participation**
  - A **single line** identifies partial participation.

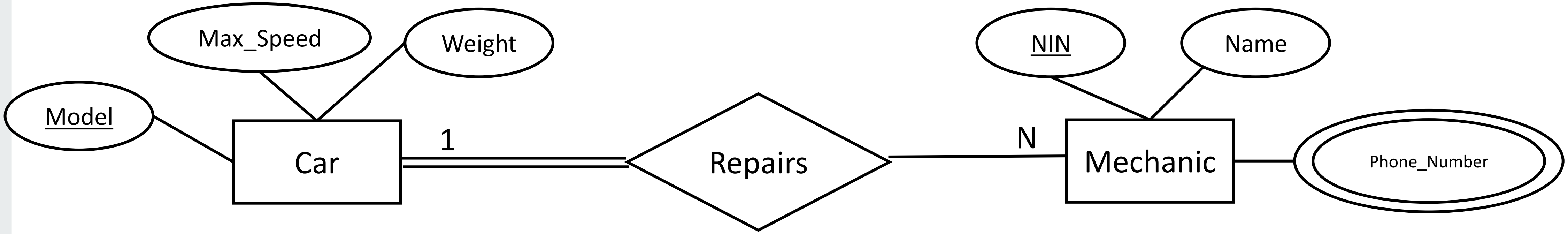
Participation constraints

- For each car, there **must be at least one** mechanic.
- Each mechanic must repair **exactly one** car.



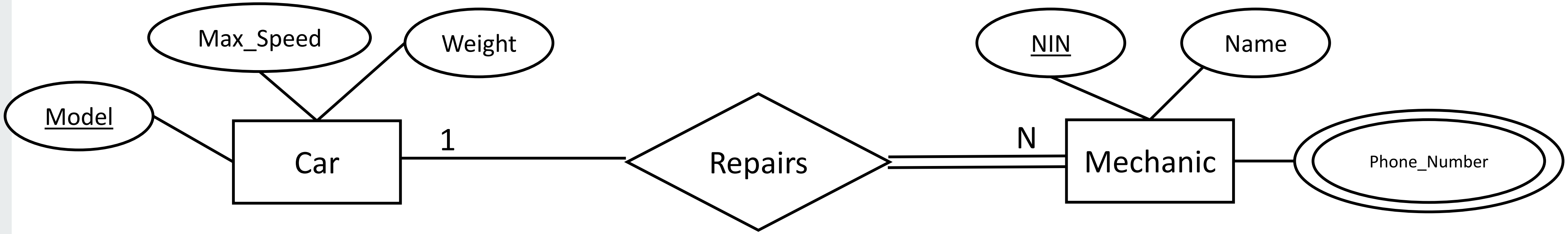
Participation constraints

- For each car, there must be at least one mechanic.
- Each mechanic repairs **at most** one car.



Participation constraints

- For each car, there **may be several mechanics**.
- Each mechanic must repair exactly one car.

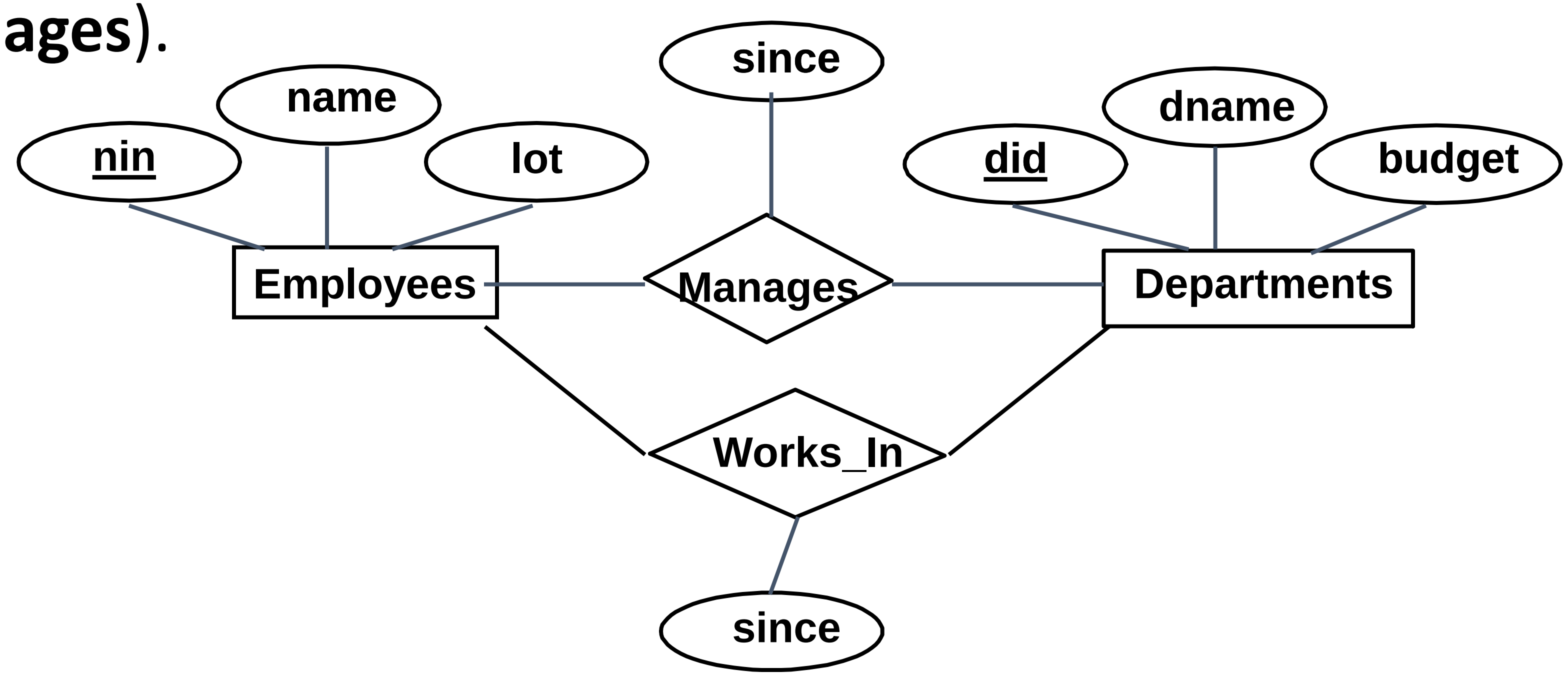


Exercise…

https://forms.office.com/e/5NcDL8BN86?origin=lprLink

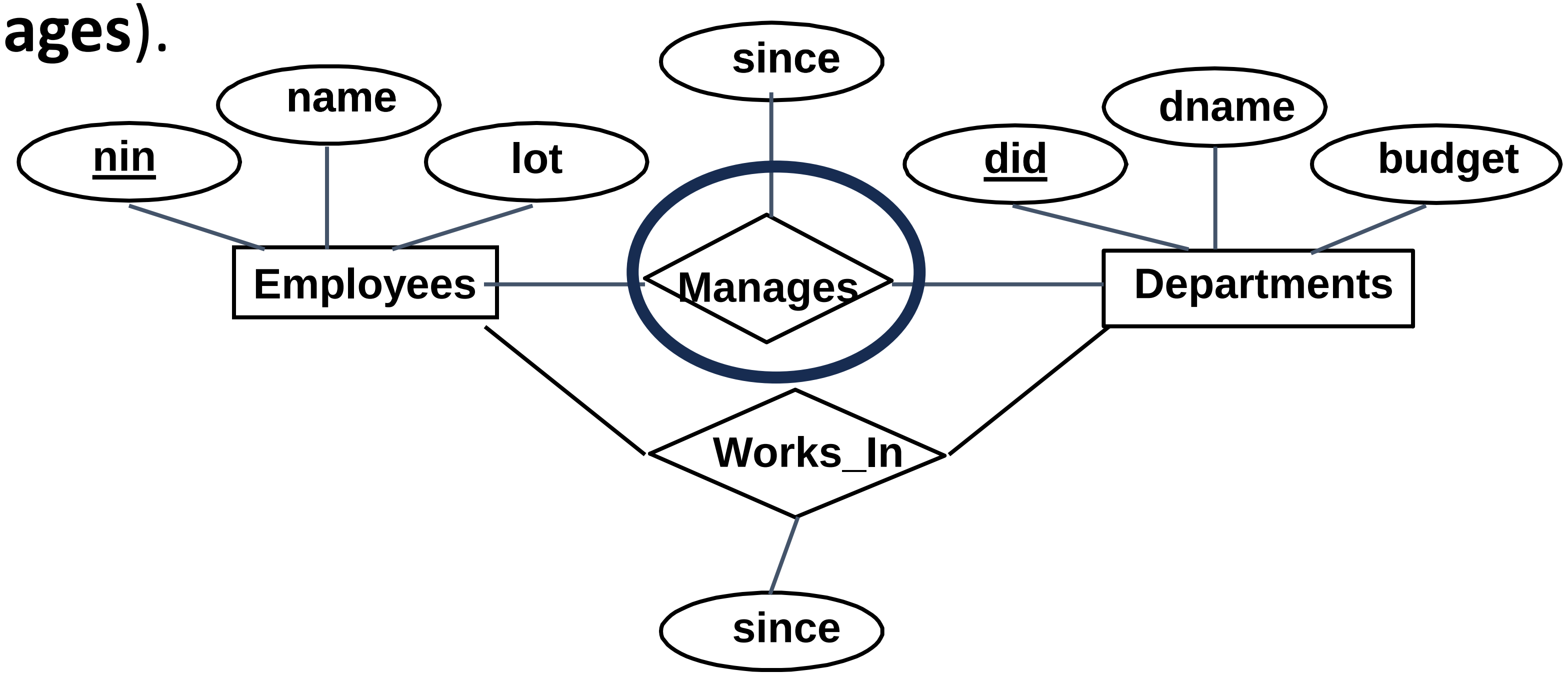
Let’s fill this.

- Consider the **cardinality** between Employees and Departments (**Manages**).



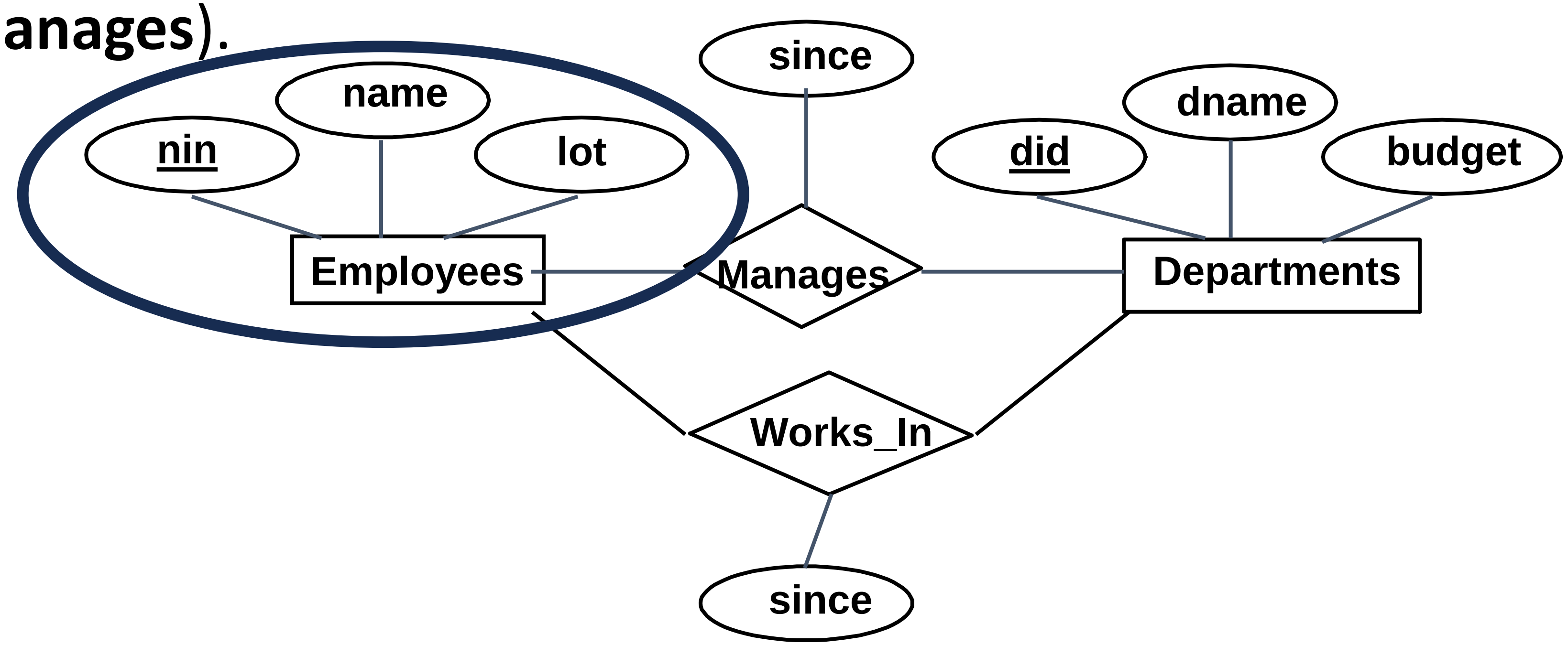
Let’s fill this.

- Consider the **cardinality** between Employees and Departments (**Manages**).



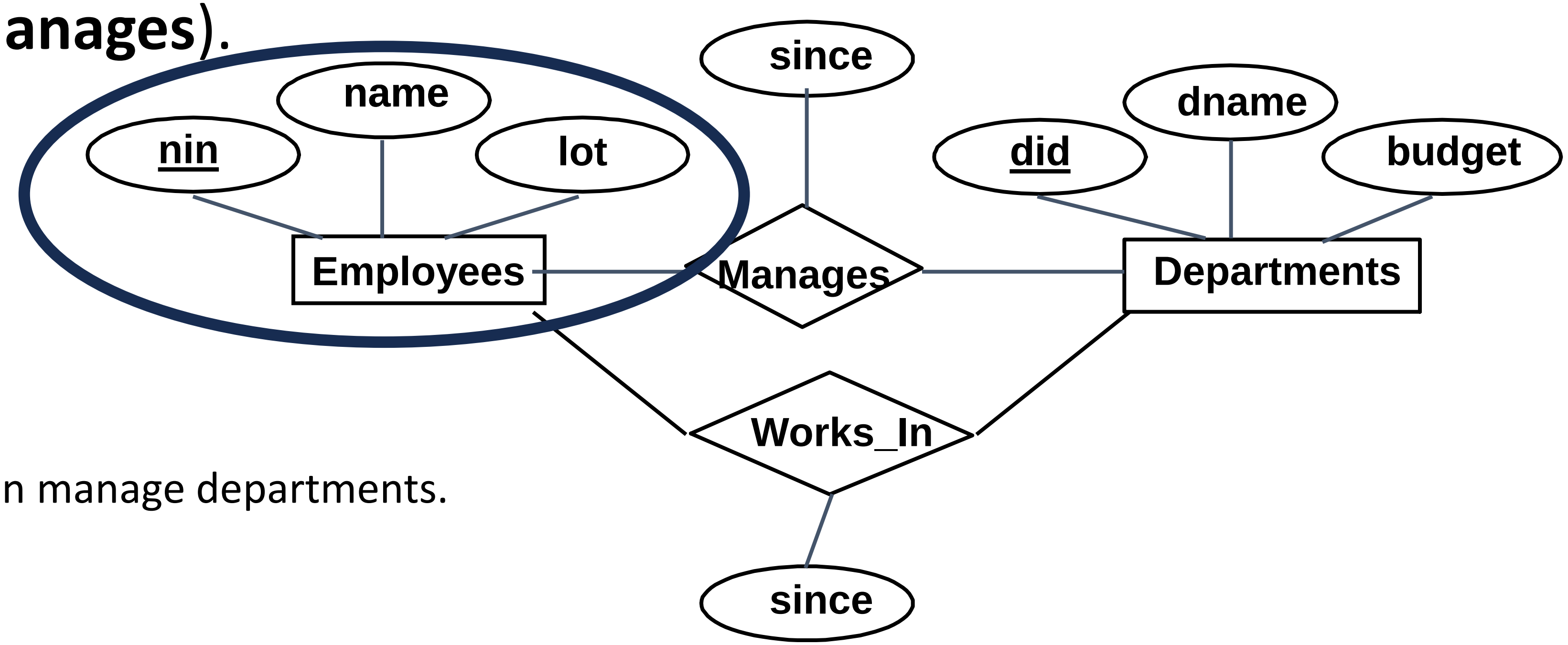
Let’s fill this.

- Consider the **cardinality** between Employees and Departments (**Manages**).



Let’s fill this.

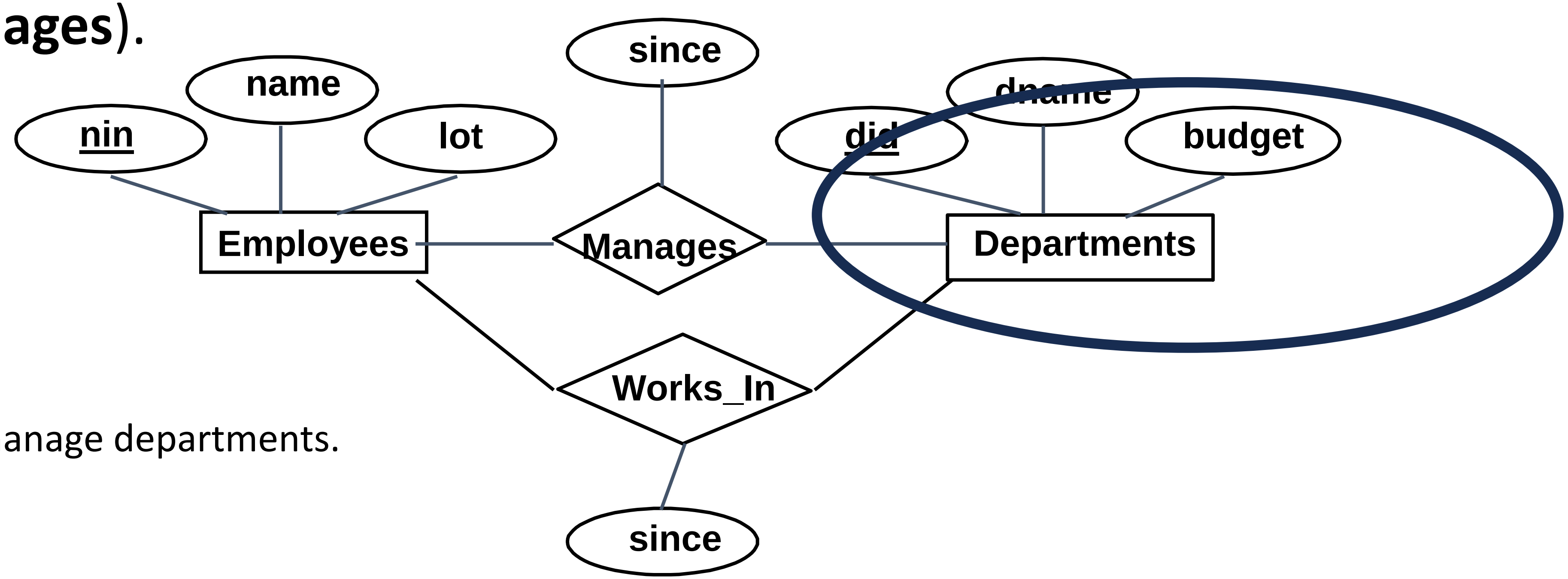
- Consider the **cardinality** between Employees and Departments (**Manages**).



An employee can manage departments.

Let’s fill this.

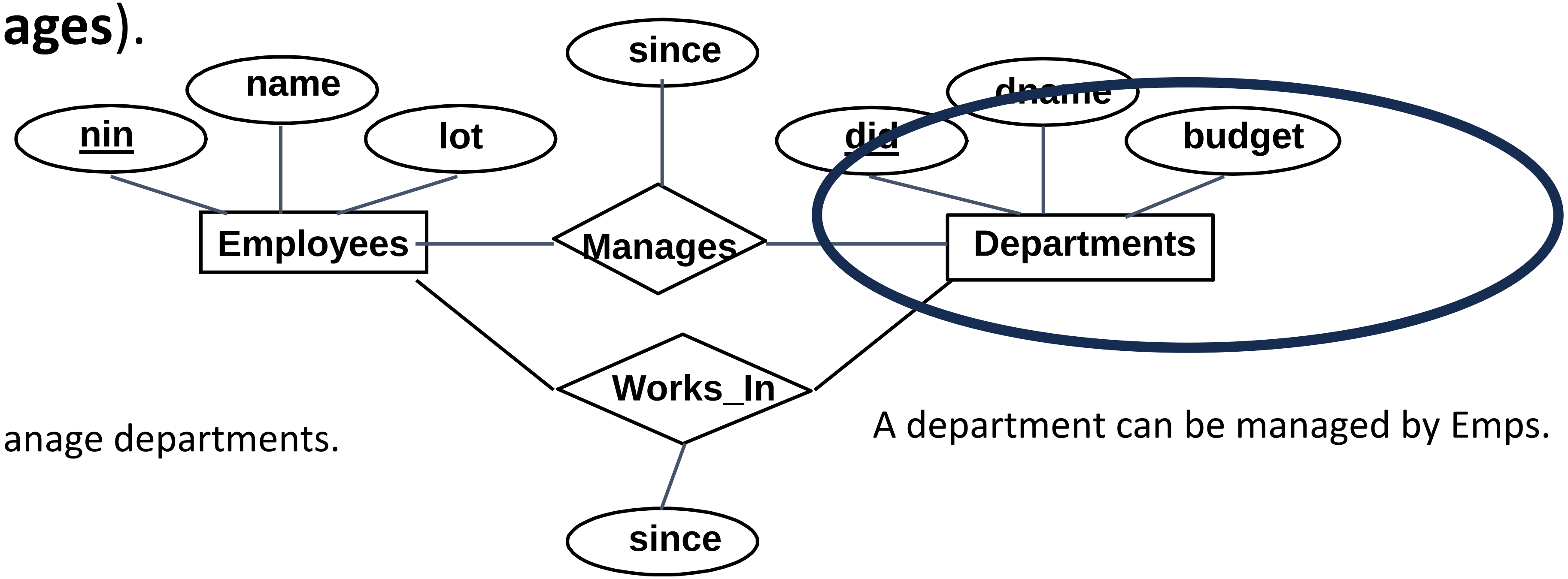
- Consider the **cardinality** between Employees and Departments (**Manages**).



An employee can manage departments.

Let’s fill this.

- Consider the **cardinality** between Employees and Departments (**Manages**).

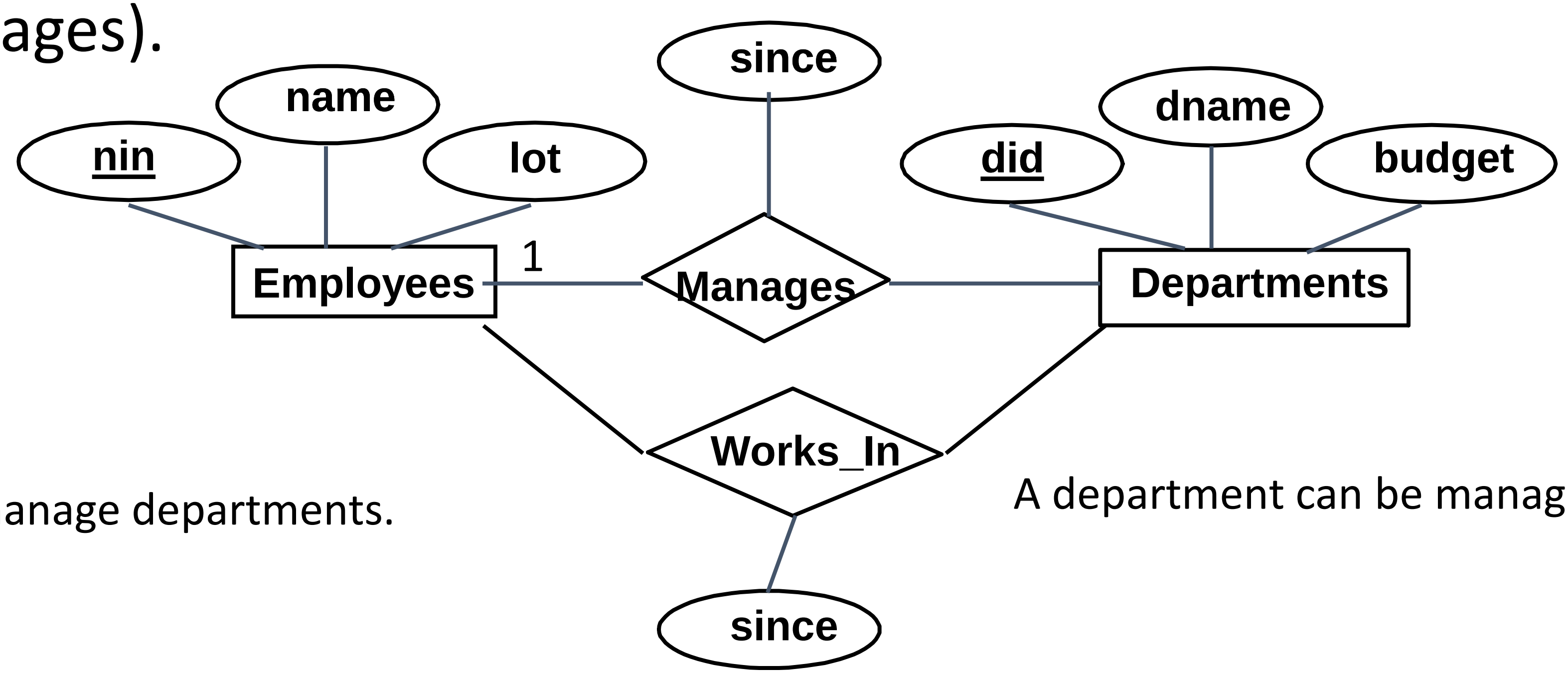


An employee can manage departments.

A department can be managed by Emps.

Let’s fill this.

- Consider the **cardinality** between Employees and Departments (Manages).

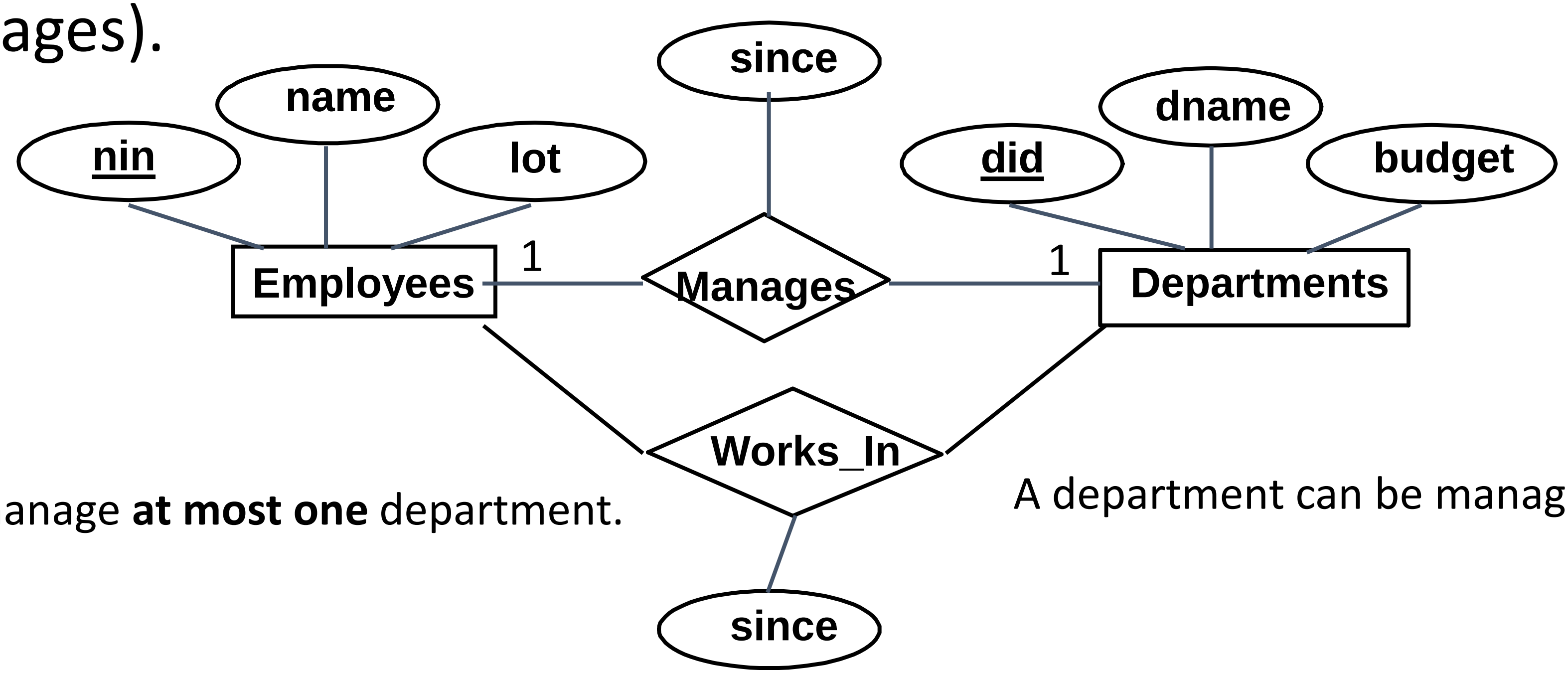


An employee can manage departments.

A department can be managed by **at most one** Emp.

Let’s fill this.

- Consider the **cardinality** between Employees and Departments (Manages).

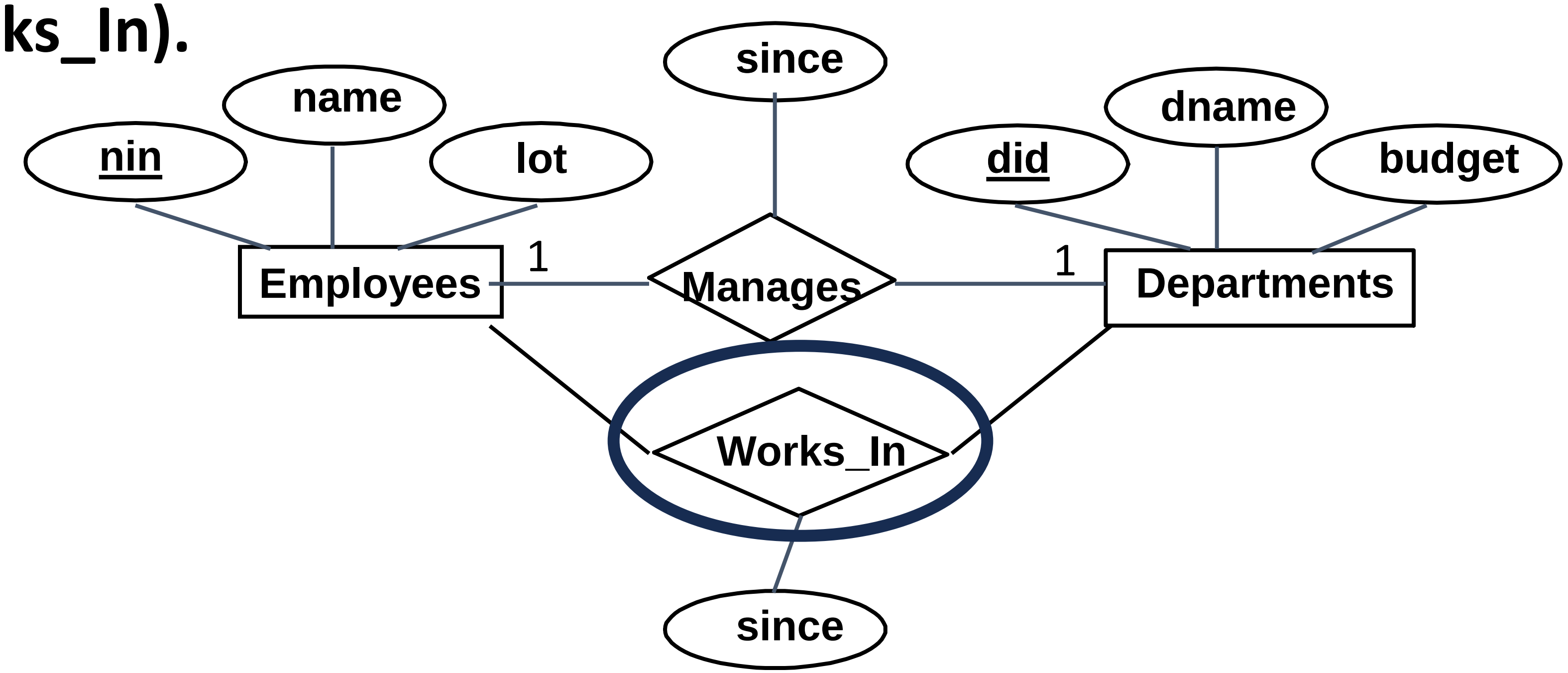


An employee can manage **at most one** department.

A department can be managed by zero or one Emps.

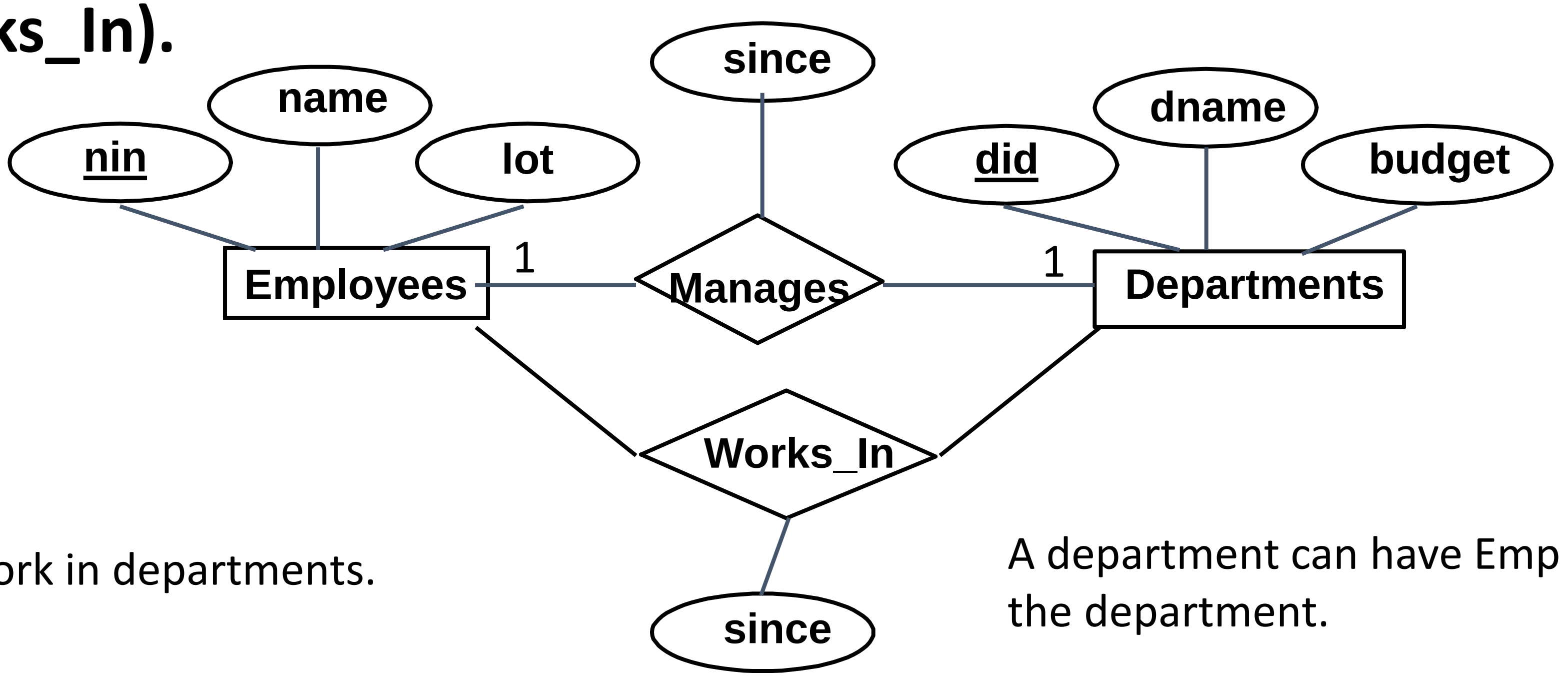
Let’s fill this.

- Consider the **cardinality** between Employees and Departments **(Works_In).**



Let’s fill this.

- Consider the **cardinality** between Employees and Departments **(Works_In).**

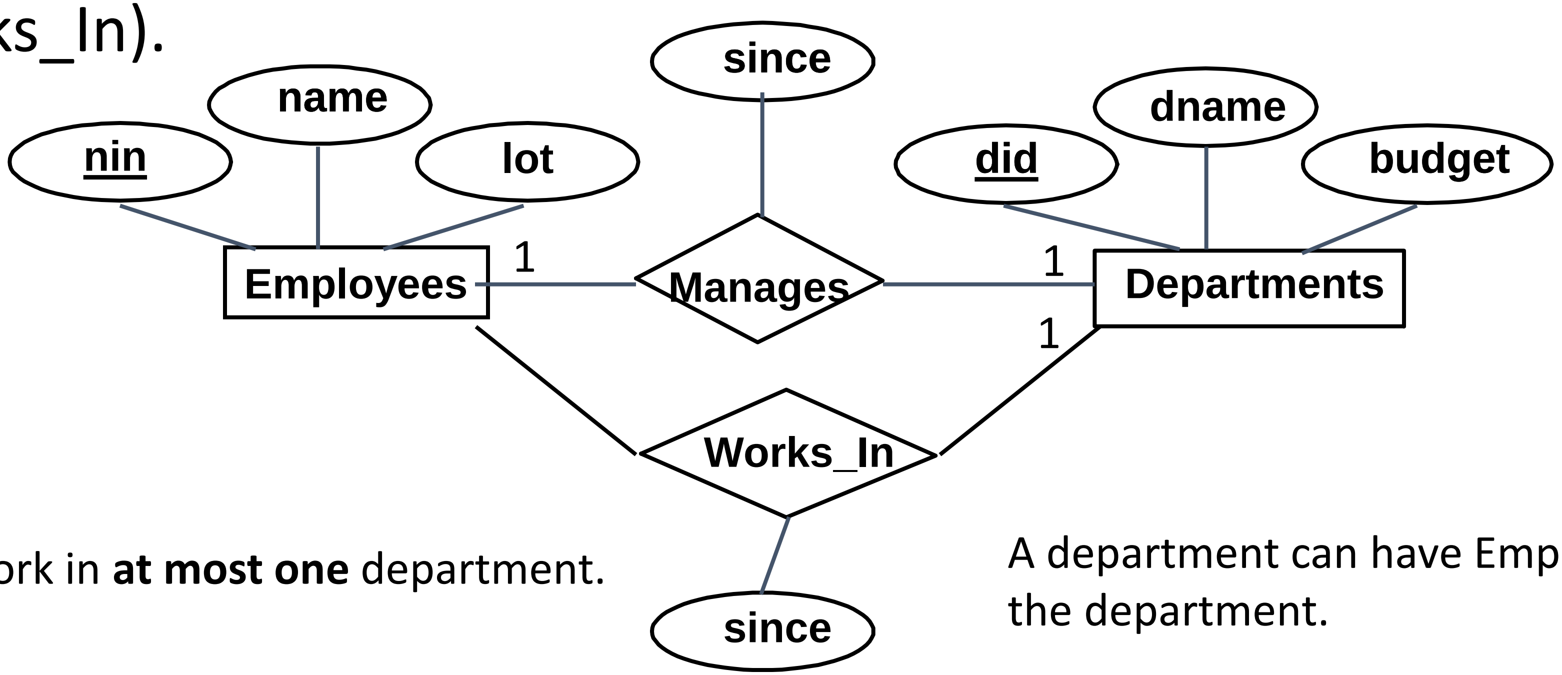


An employee can work in departments.

A department can have Emps. working in the department.

Let’s fill this.

- Consider the **cardinality** between Employees and Departments (Works_In).

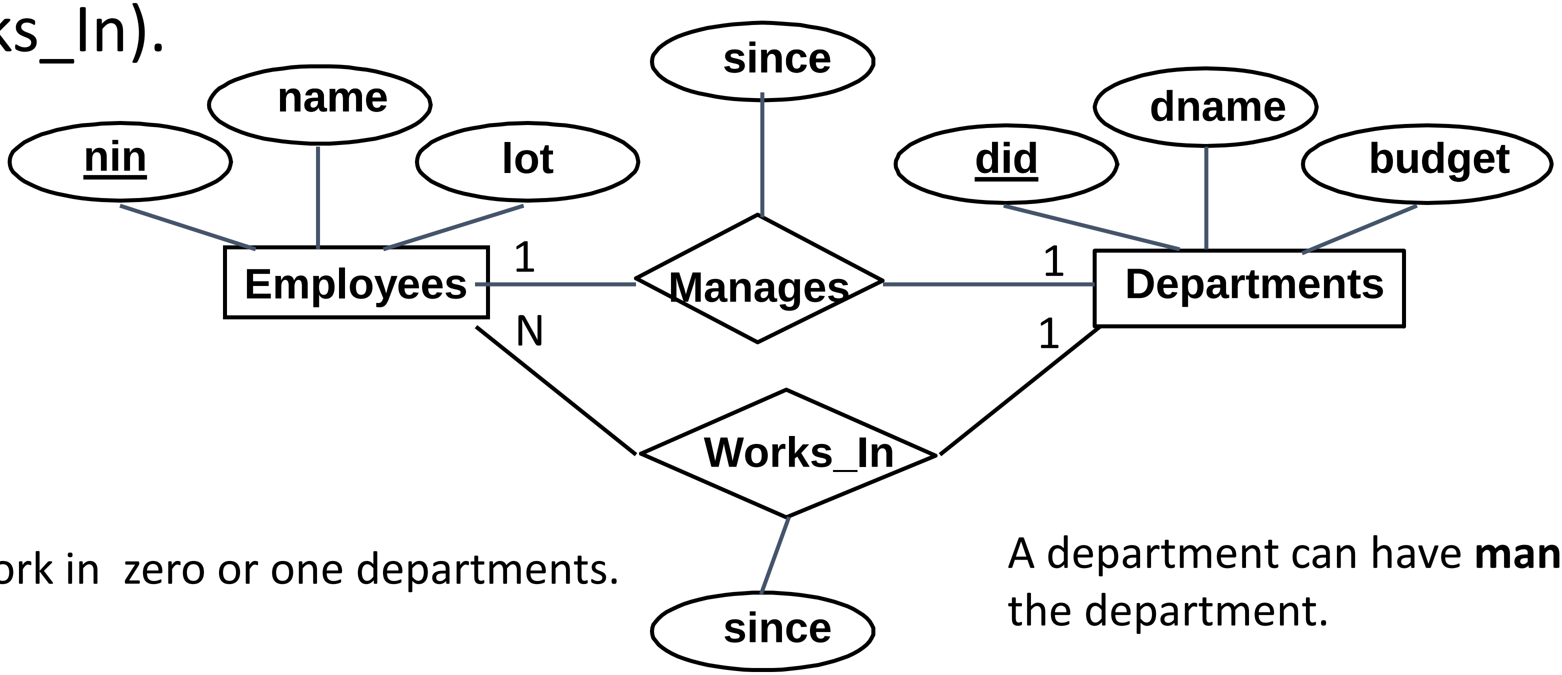


An employee can work in **at most one** department.

A department can have Emps. working in the department.

Let’s fill this.

- Consider the **cardinality** between Employees and Departments (Works_In).

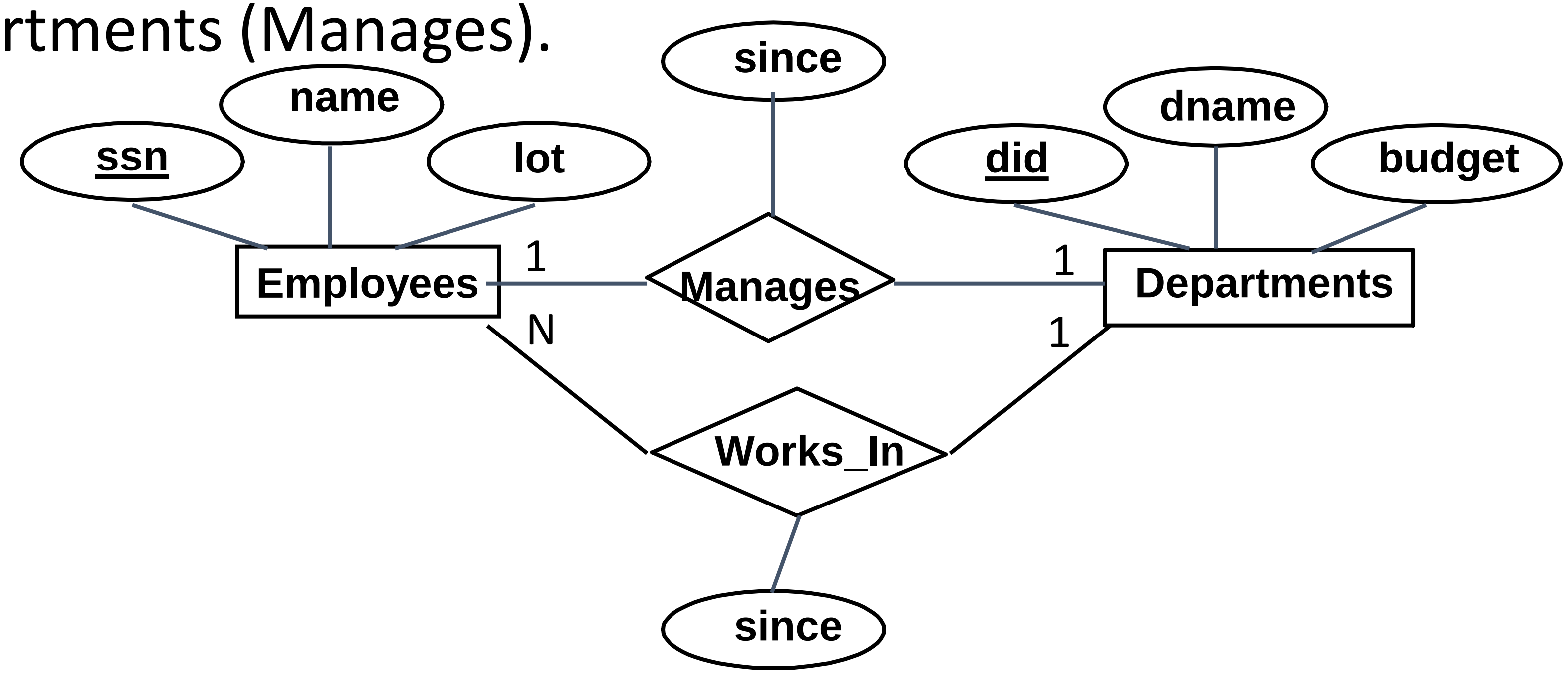


An employee can work in zero or one departments.

A department can have **many** Emps. working in the department.

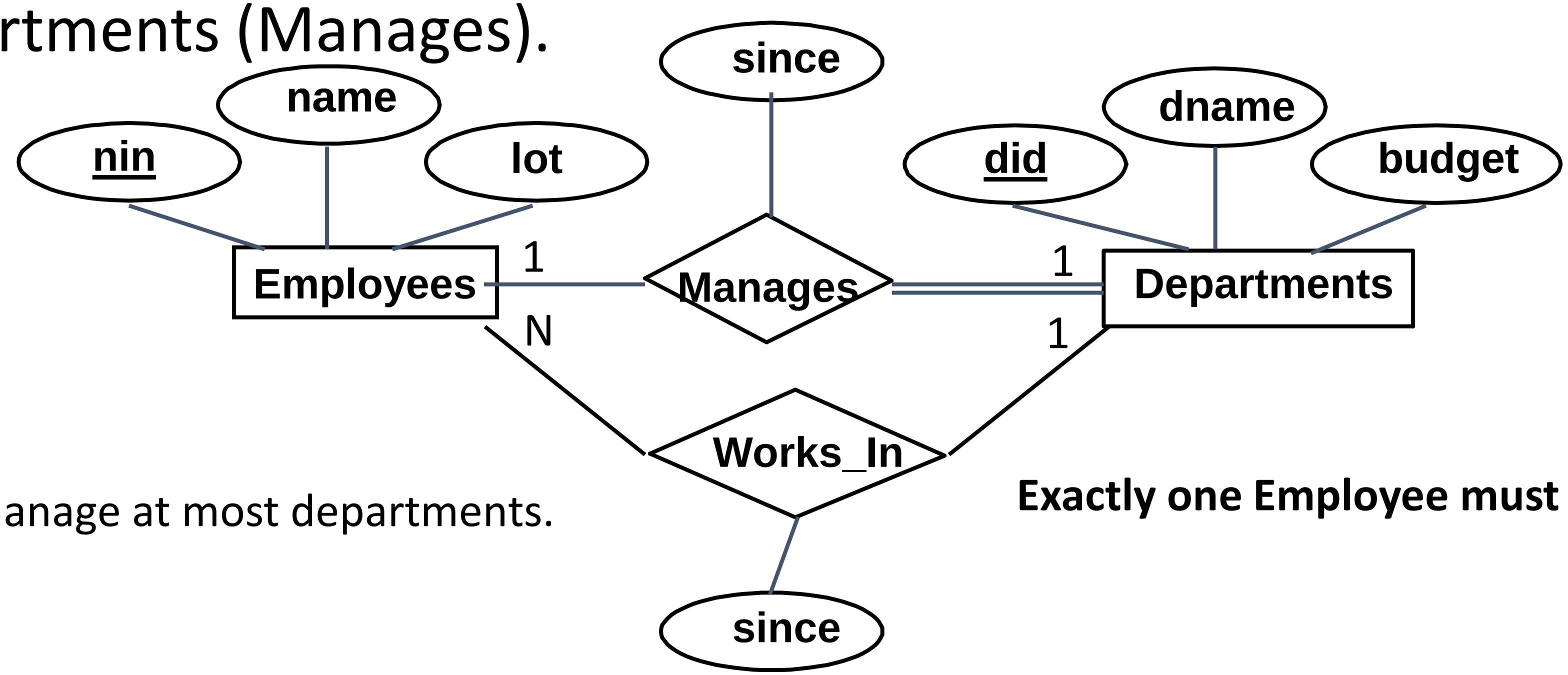
Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Manages).



Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Manages).

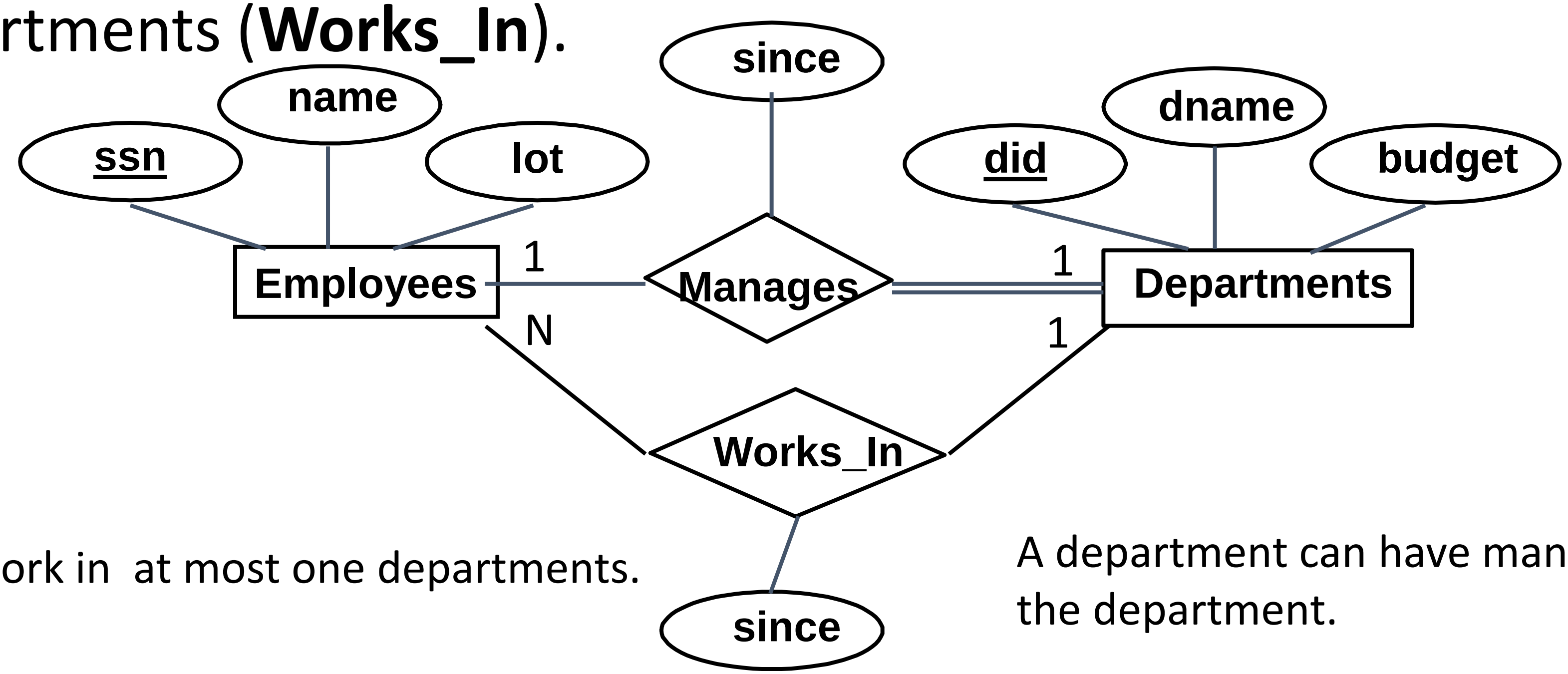


An employee can manage at most departments.

Exactly one Employee must manage a department.

Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (**Works_In**).

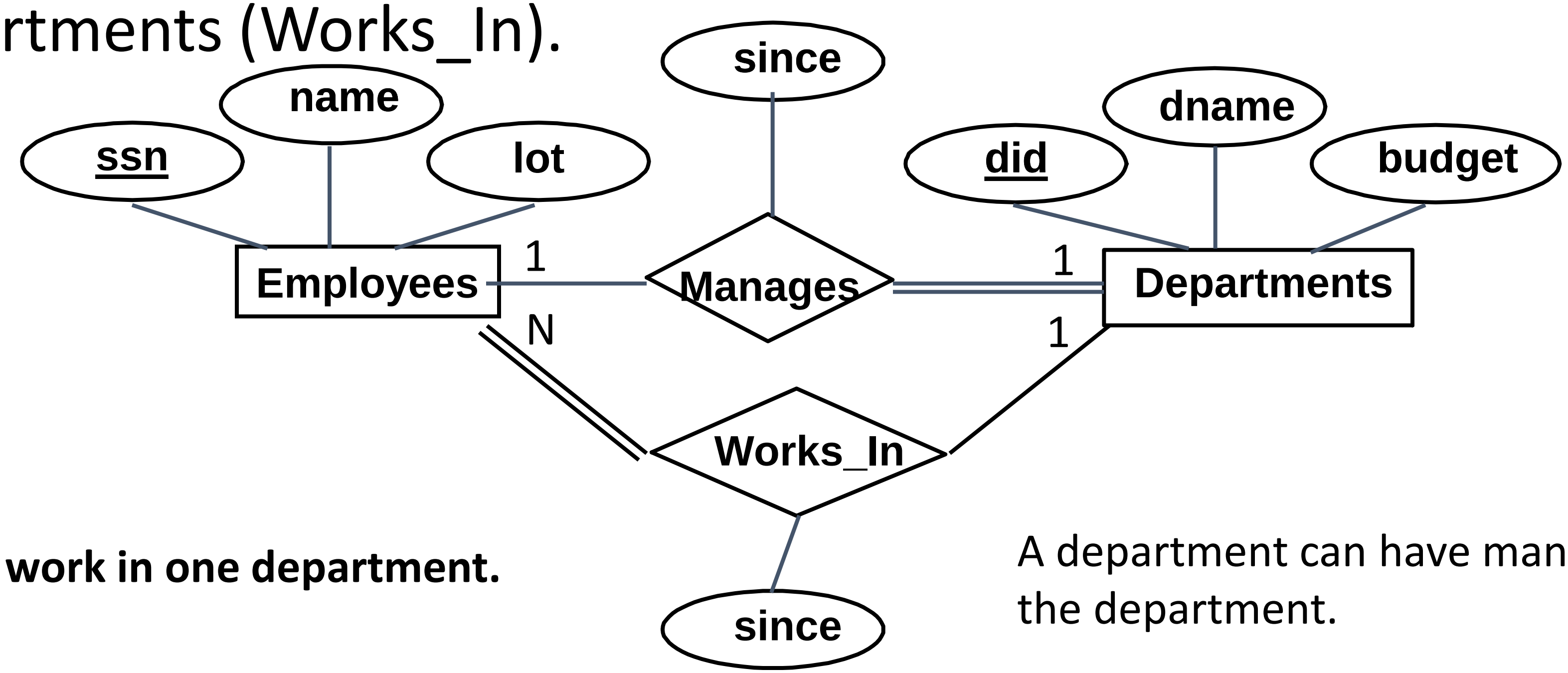


An employee can work in at most one departments.

A department can have many Emps. working in the department.

Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Works_In).

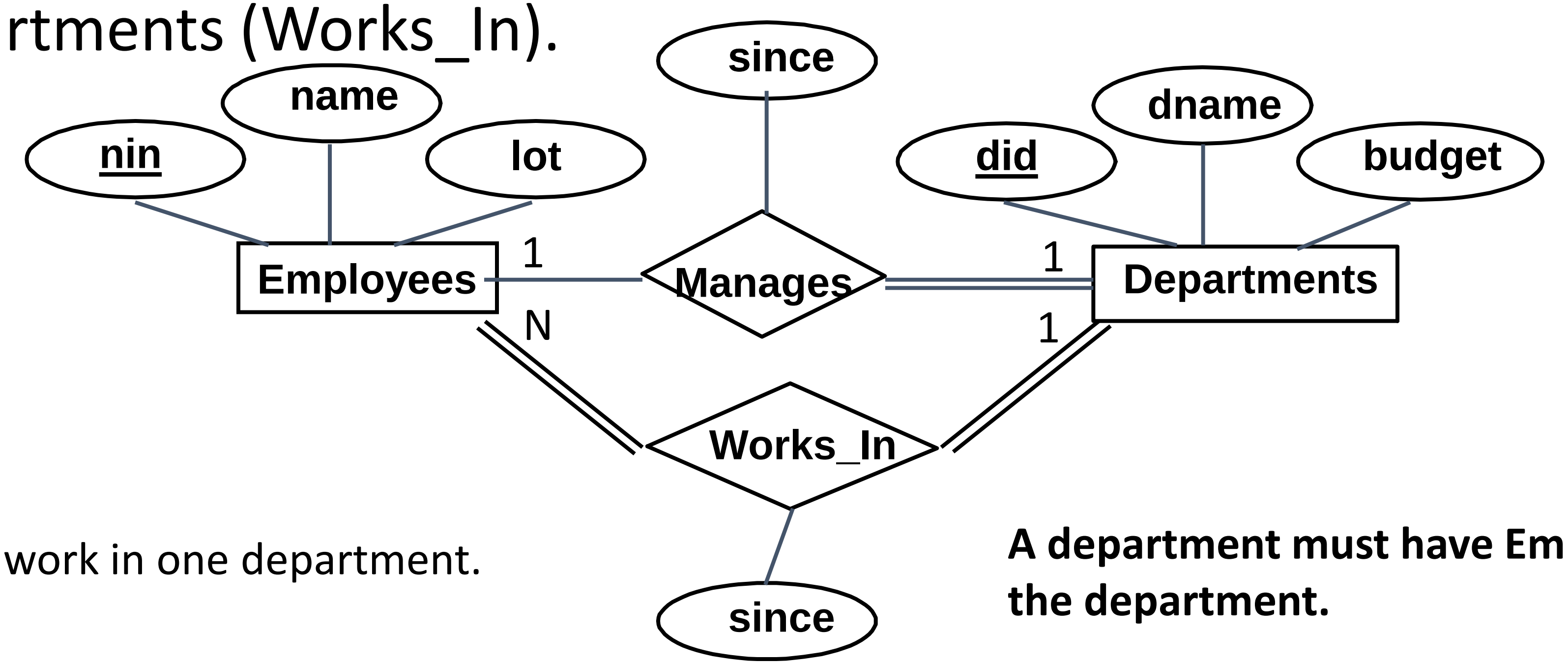


An employee must work in one department.

A department can have many Emps. working in the department.

Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Works_In).

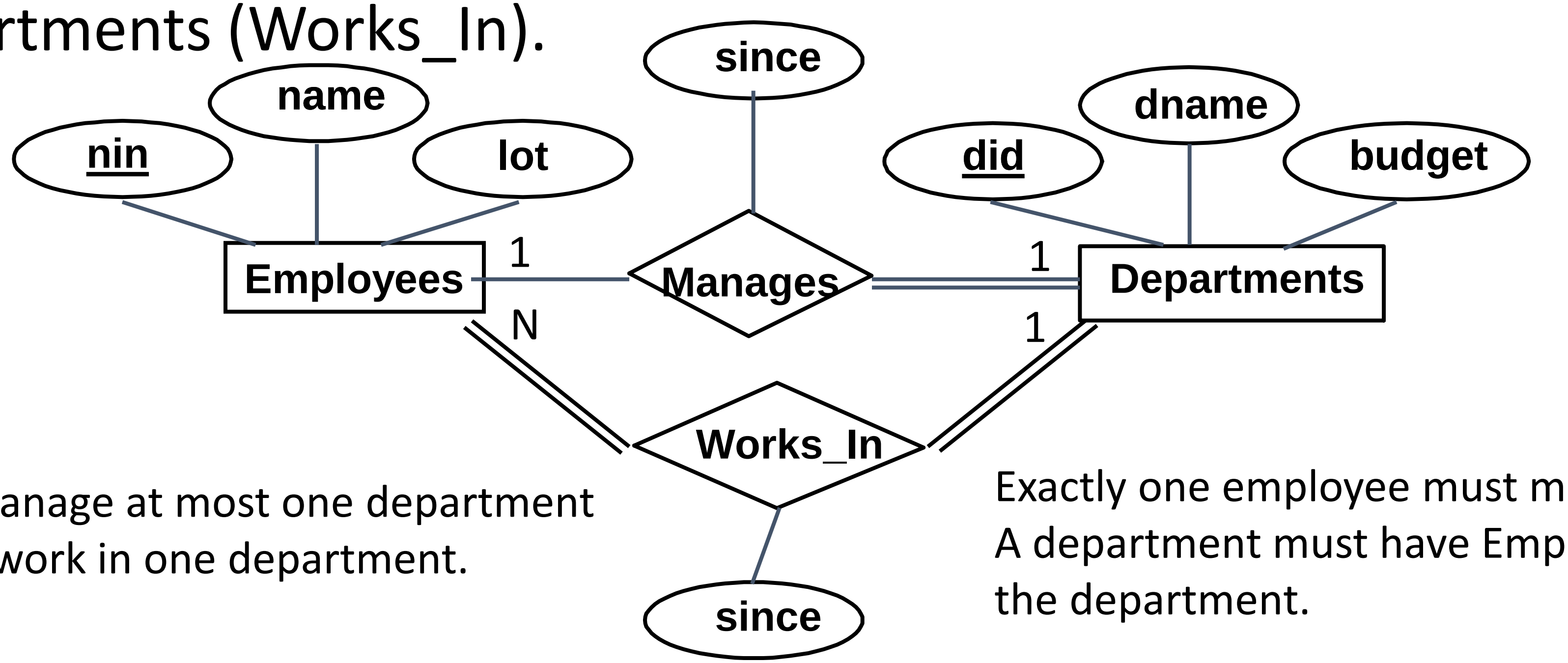


An employee must work in one department.

A department must have Emps. working in the department.

Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Works_In).

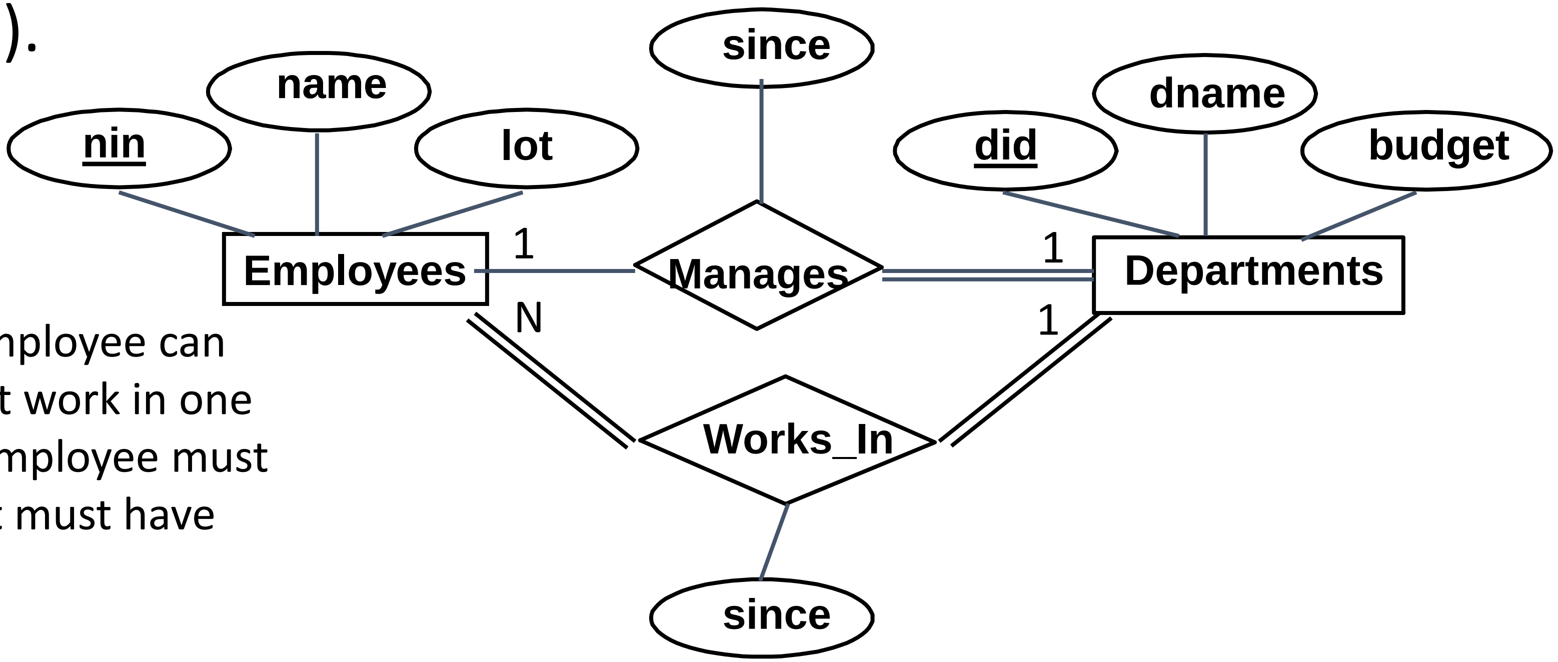


An employee can manage at most one department
An employee must work in one department.

Exactly one employee must manage a department.
A department must have Emps. working in the department.

Let’s fill this.

- Consider the **participation constraints** between Employees and Departments (Works_In).



“Please draw an ER diagram where an employee can manage at most one department or must work in one department providing that exactly one employee must manage a department and a department must have Emps. working in the department.”

Weak entity and weak relation sets.

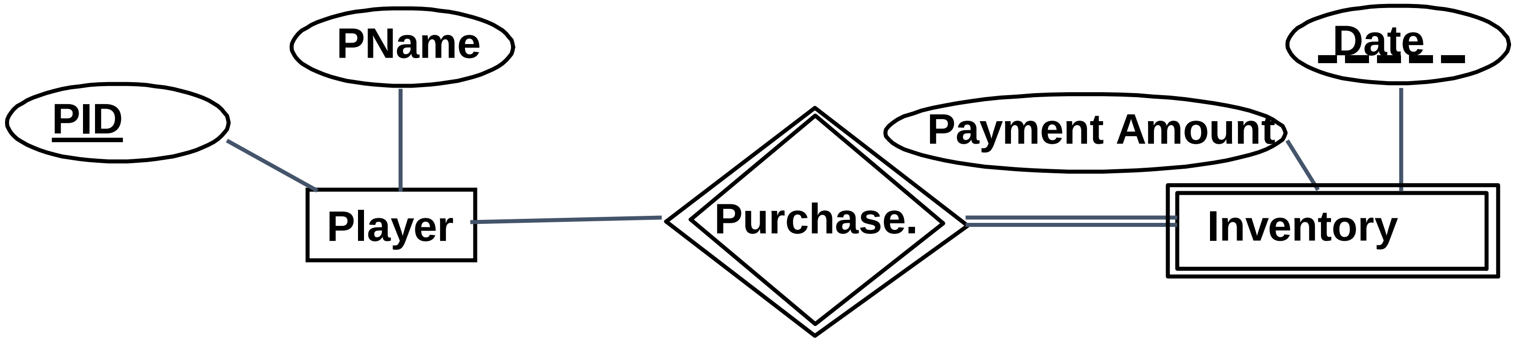
Consider the following case.

A player makes in-game equipment (armour, ammunition, etc) purchases using imaginary money (coins). The game server has to keep the information as long as the player plays the game.

Assume we do not want to keep the purchase information when the player deletes their account, and we want to automate this deletion operation.

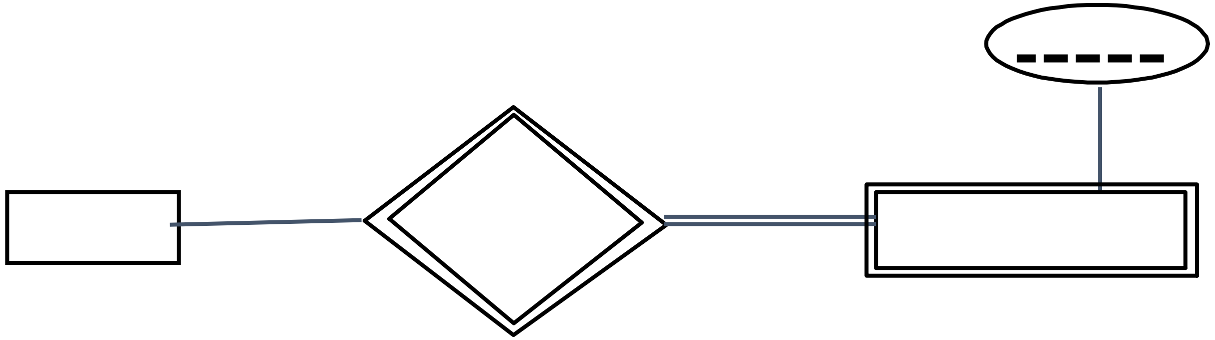
The DBMS automatically remove all the redundant purchase **data from the tables.**

We use Weak-Entity and Relation sets to represent this.



Weak Entity-Relation Sets

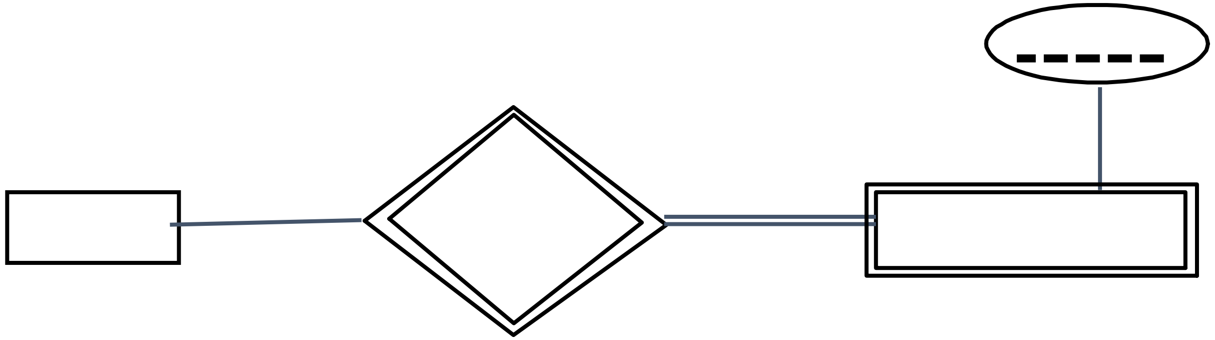
Represented by a **double-lined diamond**, a **double-lined rectangle** connected by double lines. The weak key attribute is represented by a dashed underline.



Weak Entity-Relation Sets

Represented by a **double-lined diamond**, a **double-lined rectangle** connected by double lines. The weak key attribute is represented by a dashed underline.

The double-lined rectangle is the **subject**, the single-lined rectangle is the **owner**.

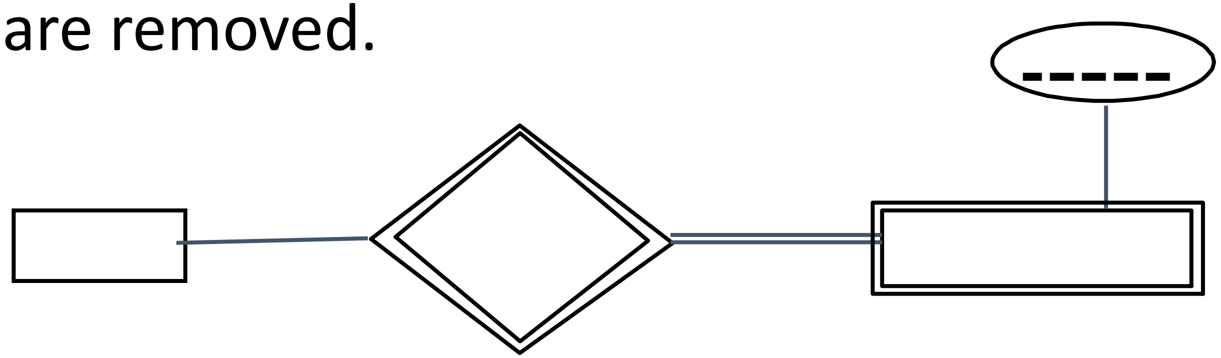


Weak Entity-Relation Sets

Represented by a **double-lined diamond**, a **double-lined rectangle** connected by double lines. The weak key attribute is represented by a dashed underline.

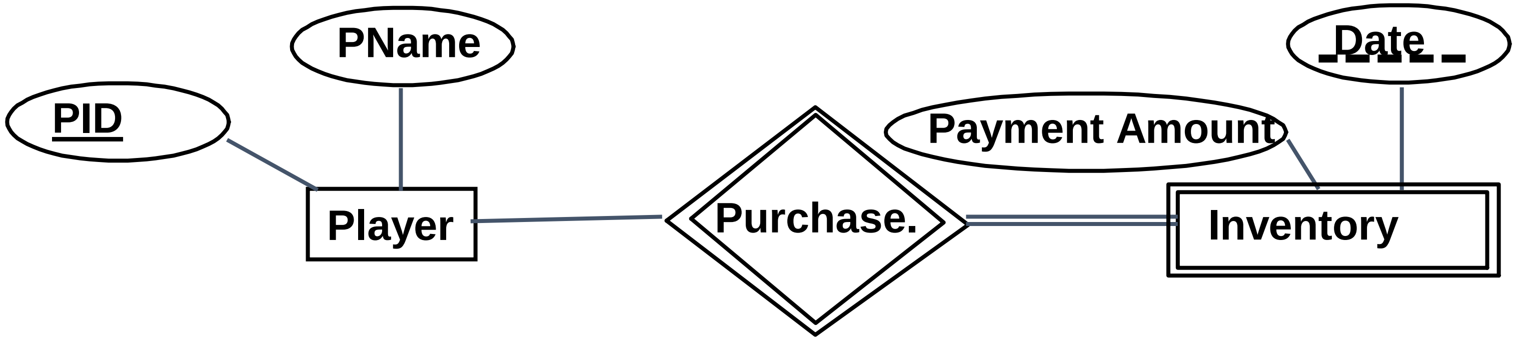
The double-lined rectangle is the **subject**, the single-lined rectangle is the **owner**.

When an entity in the owner table is removed, all the related entries in the subject table are removed.



Weak entity set and weak relation sets.

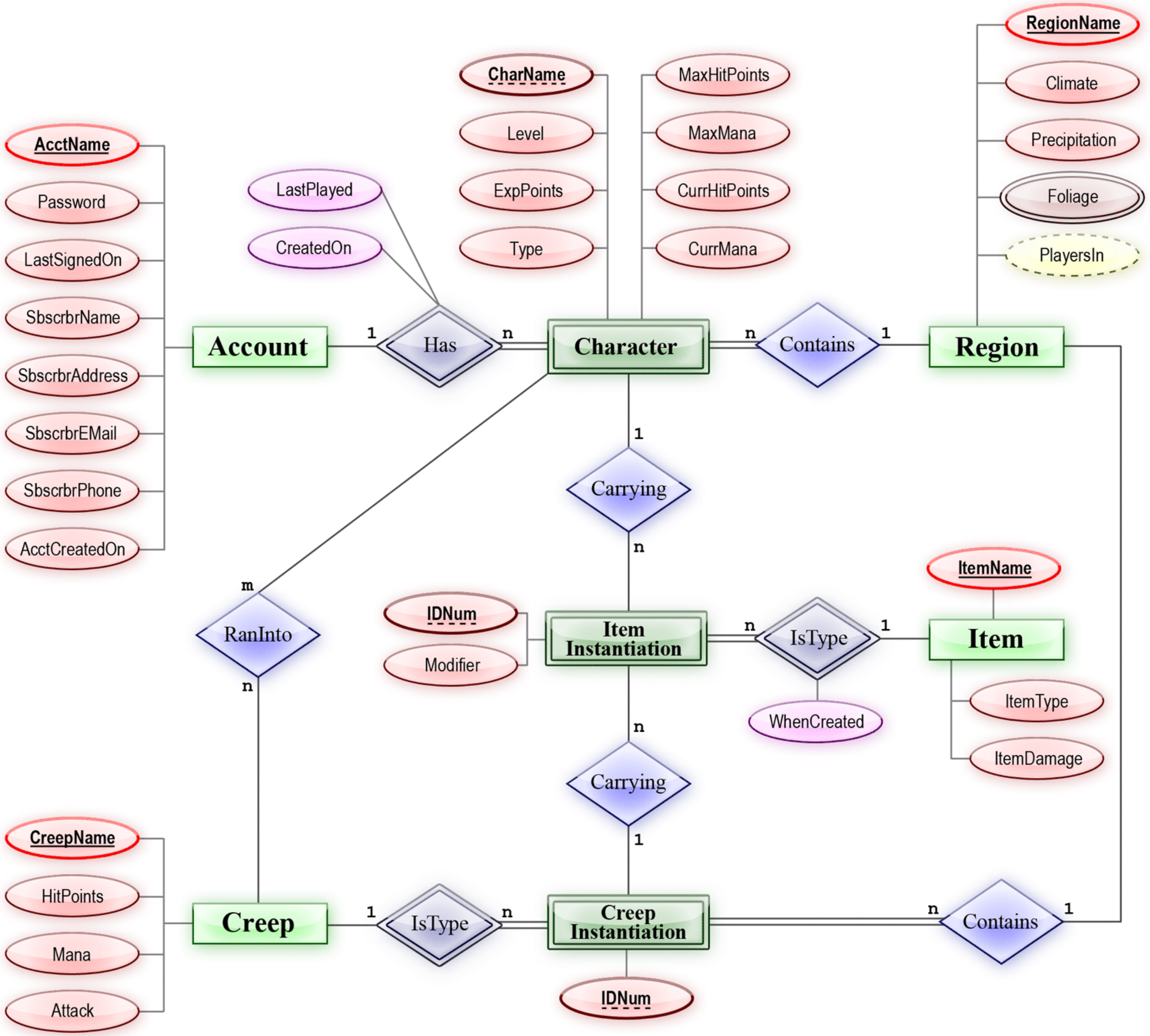
- Weak Entity sets do not possess a Primary Key; they **possess a Weak Key.**
- The primary key of the Owner table and the Weak Key of the Subject table constitutes a key for the Subject Table. (PID, Data is a key for Inventory)



PName	PID
Alex	1
Mark	2
Claudia	3
Summer	4

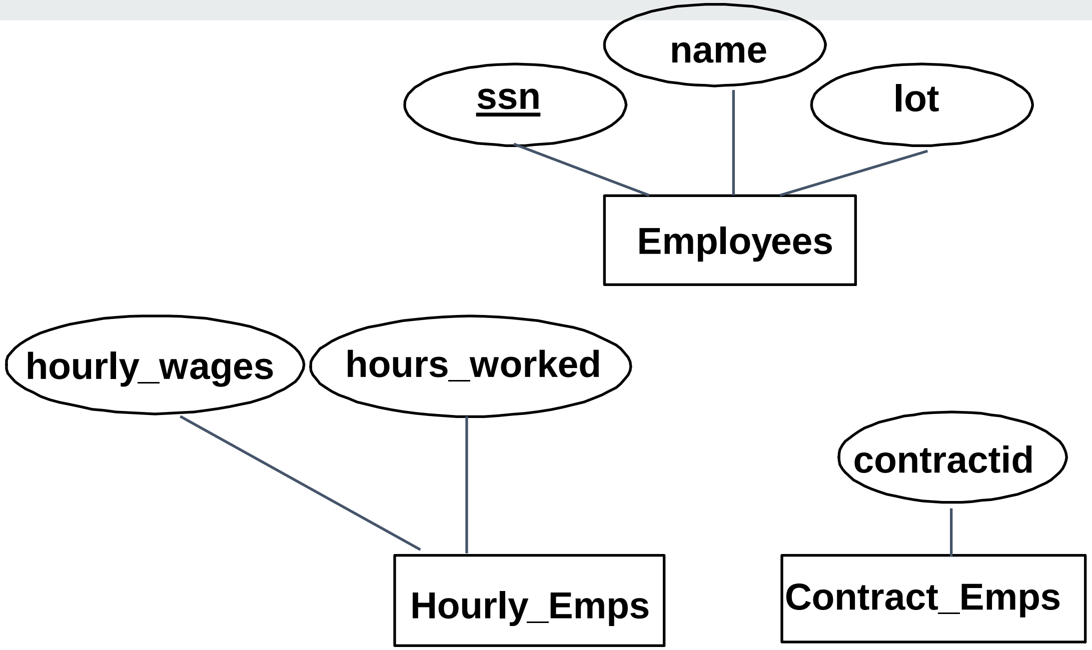
Payment Amount	Date
1121	01/04/1982
12312	01/04/1982
1121	01/04/1958
1121	10/02/1994

Massively multiplayer online role-playing game



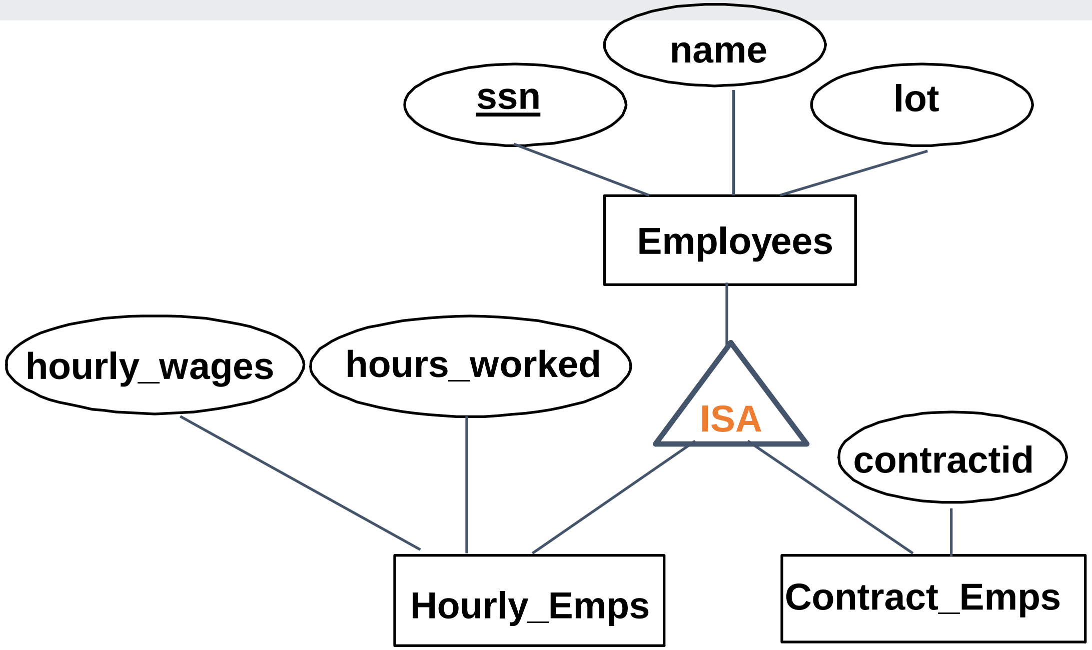
Extended ER

ISA (`is a’) Hierarchies

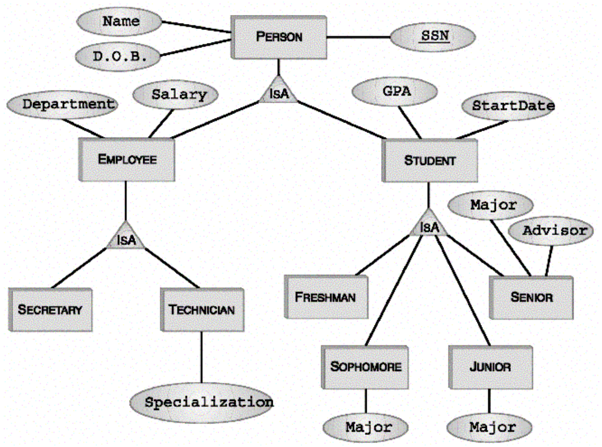


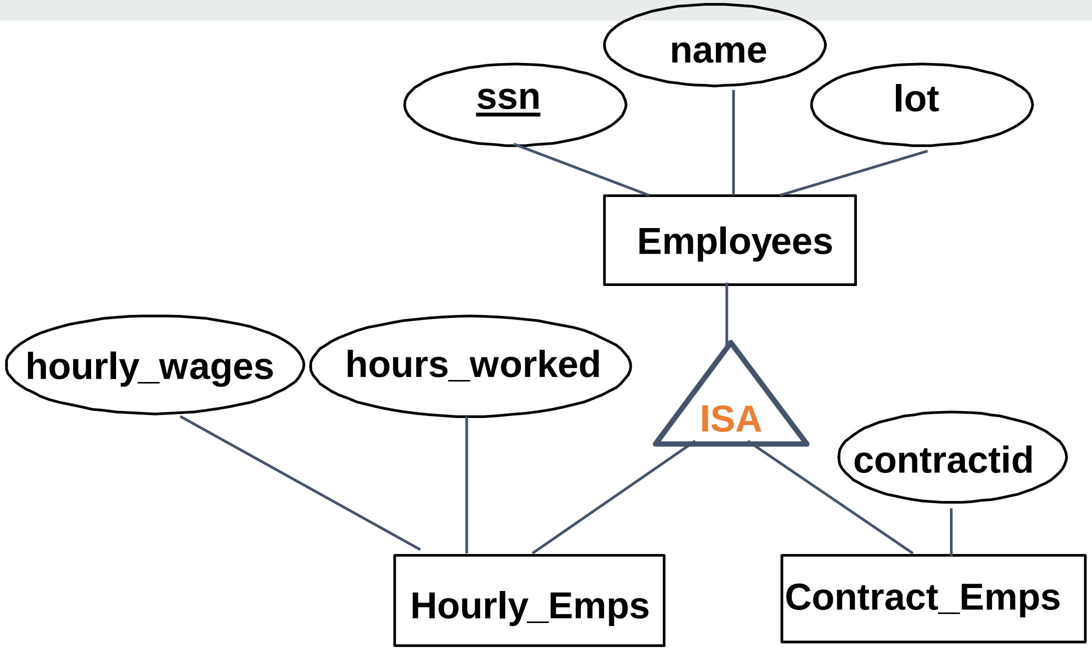
Extended ER

ISA (`is a') Hierarchies



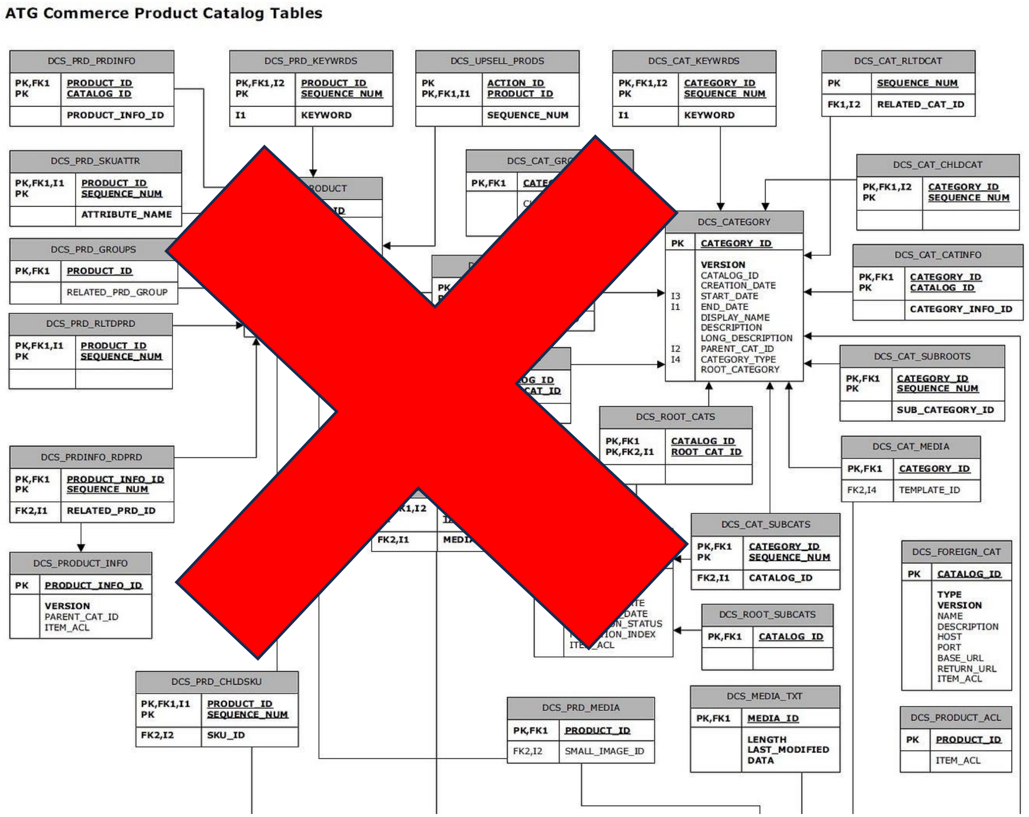
Extended ER
ISA (`is a’) Hierarchies

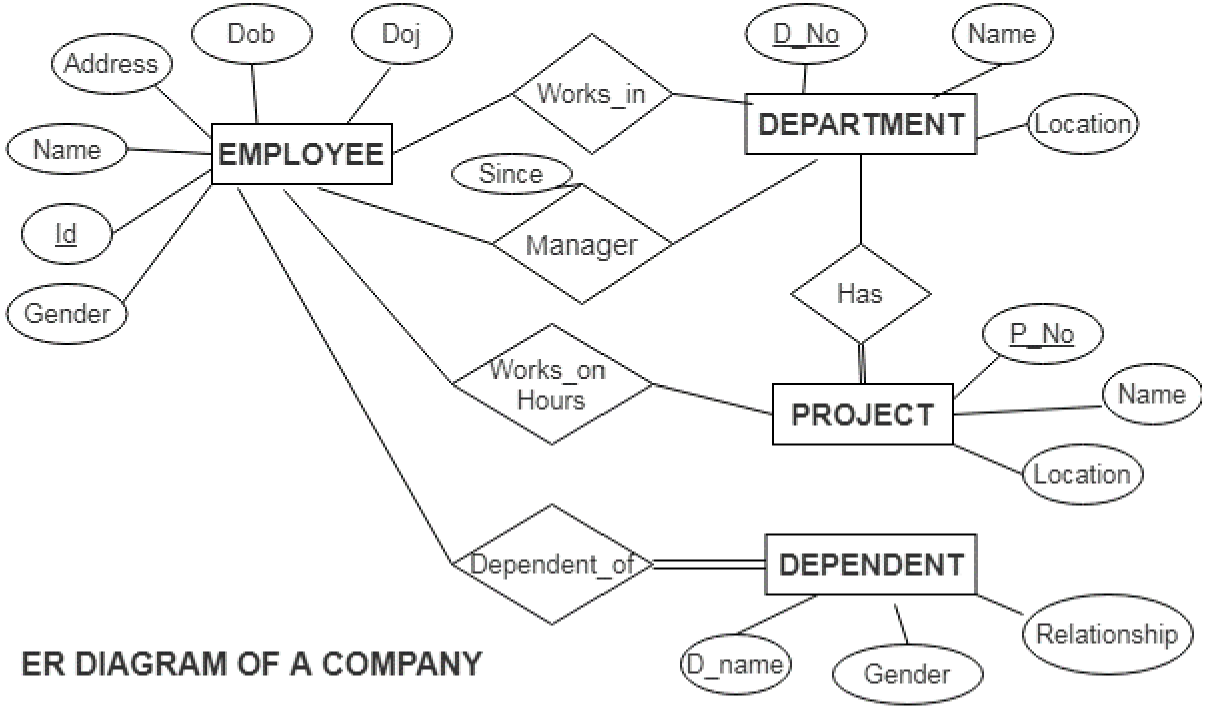




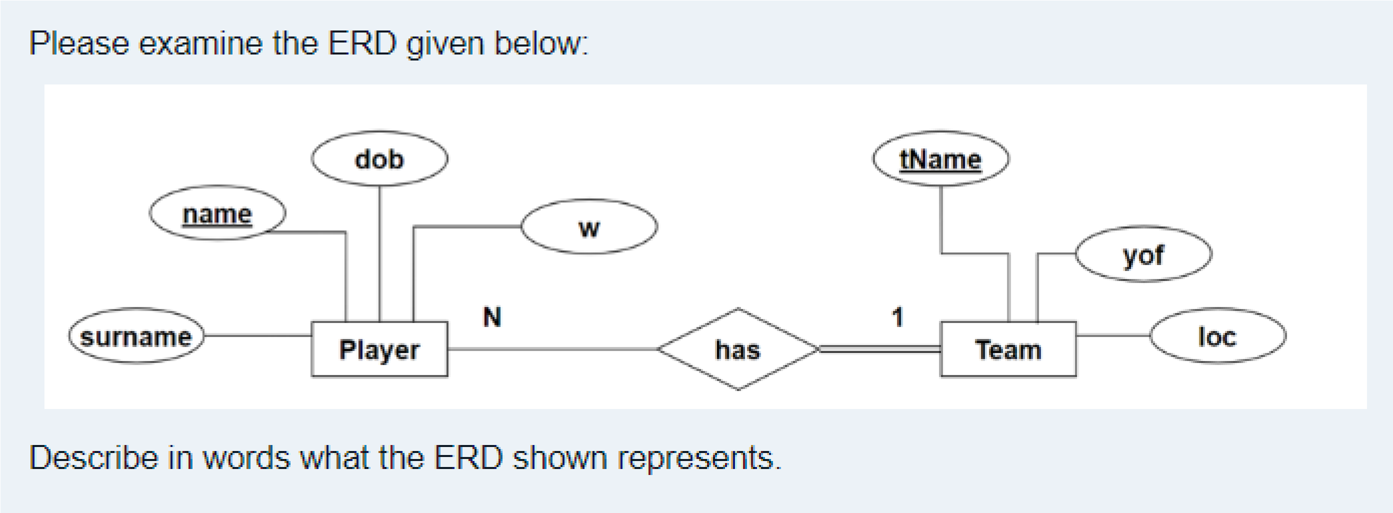
- *Overlap constraints*: Can Uraz be an Hourly Employee as well as a Contract Employee?
- *Covering constraints*: Does every Employee also have to be an Hourly Employee or a Contract Employee?
- Reasons for using ISA:
  - To add descriptive attributes specific to a subclass.
  - To identify entities that participate in a relationship.

Recall: please use the symbolism we use in the lectures.





Previous year Exam Question



Please examine the ERD given below:

Describe in words what the ERD shown represents.

A team must have at least one player.
A player may belong to at most one team.

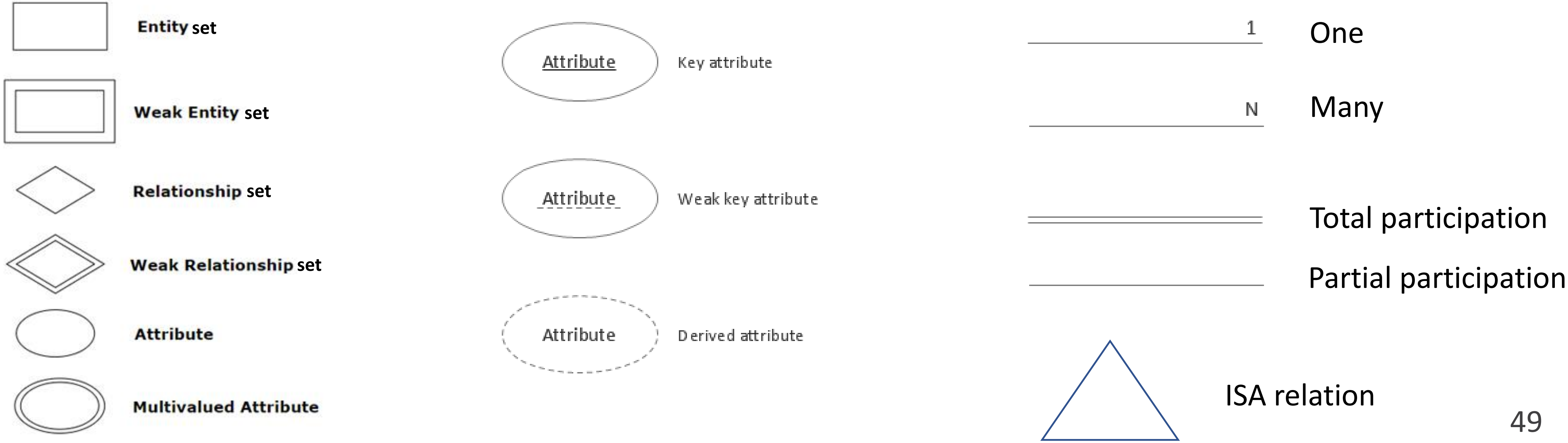
SCC.221
Data Engineering

2024 - Week 2 – Relational Database.

Uraz C Turker

What have we learnt so far?

- We learn key ER concepts which allows us to design relational databases.



Entity set

Weak Entity set

Relationship set

Weak Relationship set

Attribute

Multivalued Attribute

Attribute Key attribute

Attribute Weak key attribute

Attribute Derived attribute

1 One

N Many

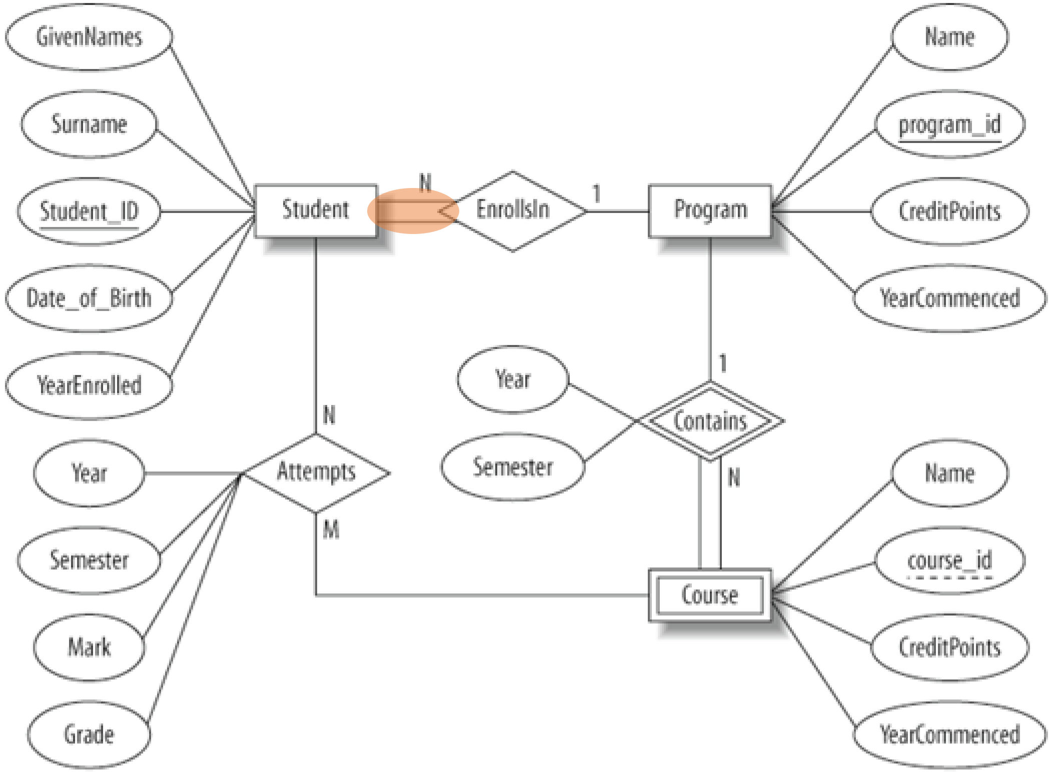
Total participation

Partial participation

ISA relation

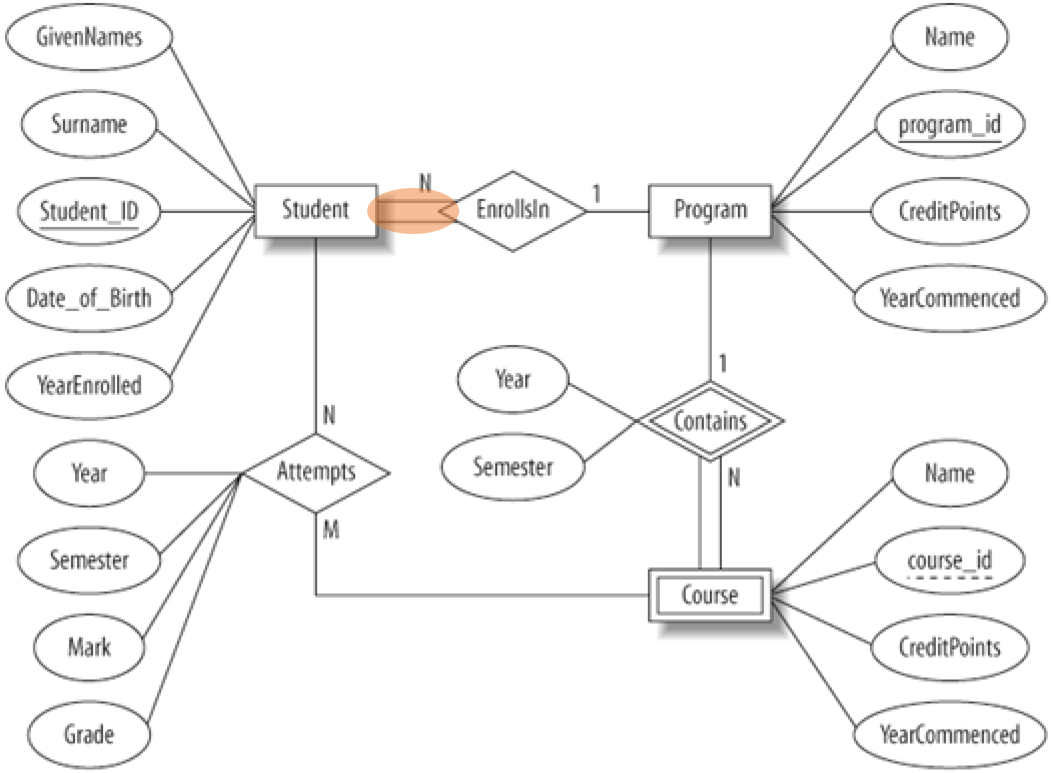
What have we learnt so far?

- A student …… enrol in …… program.



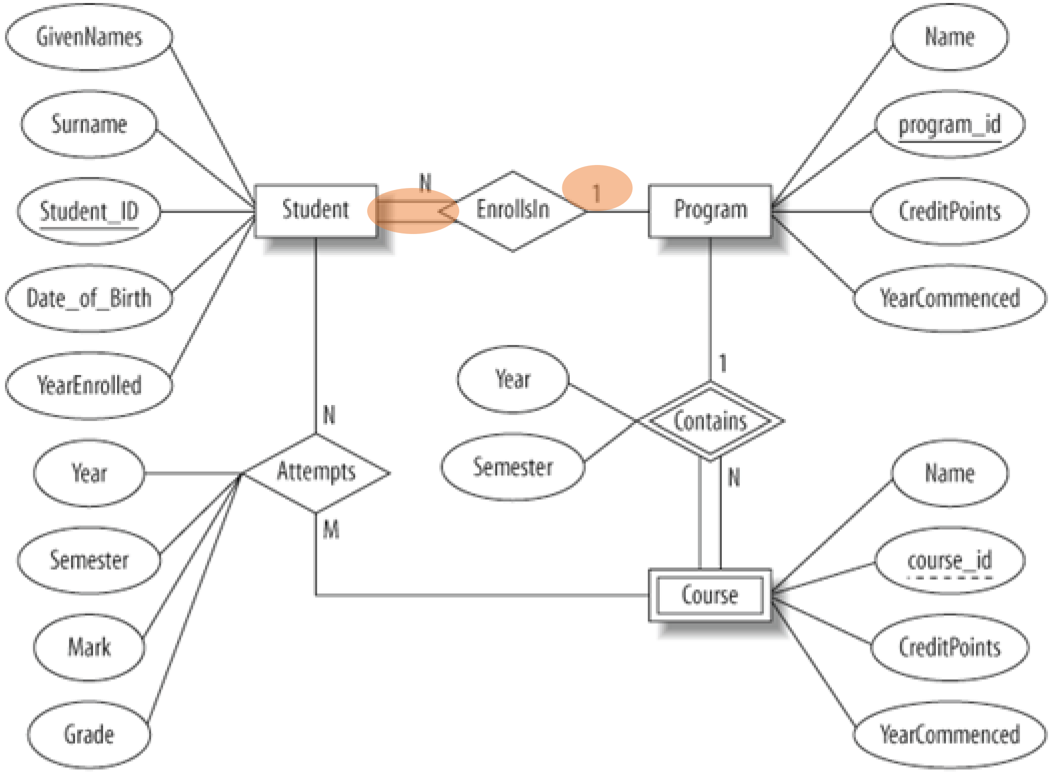
What have we learnt so far?

- A student **must** enrol in **……** program.



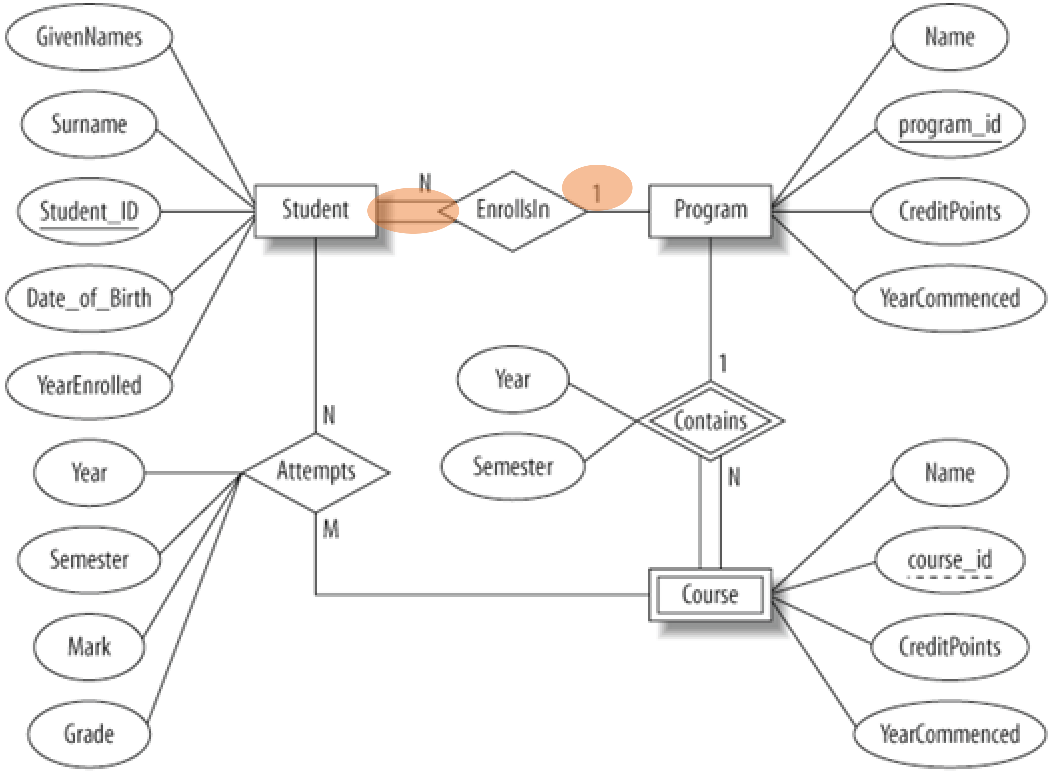
What have we learnt so far?

- A student **must** enrol in …. program.



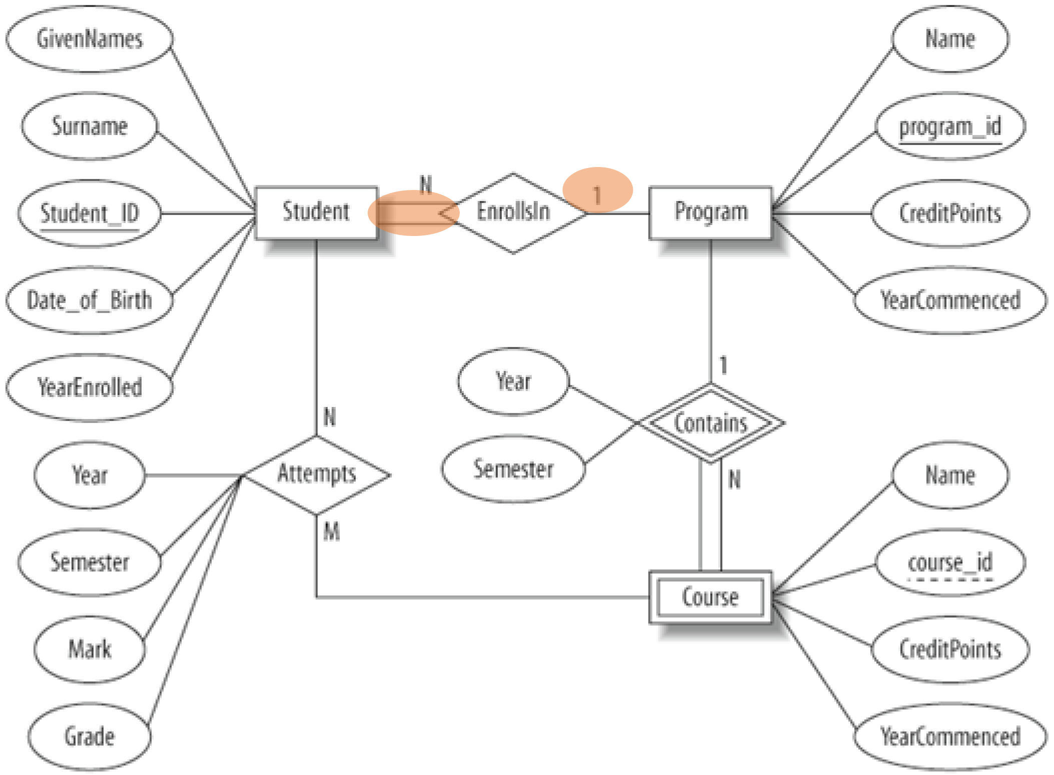
What have we learnt so far?

- A student **must** enrol in **one** program.



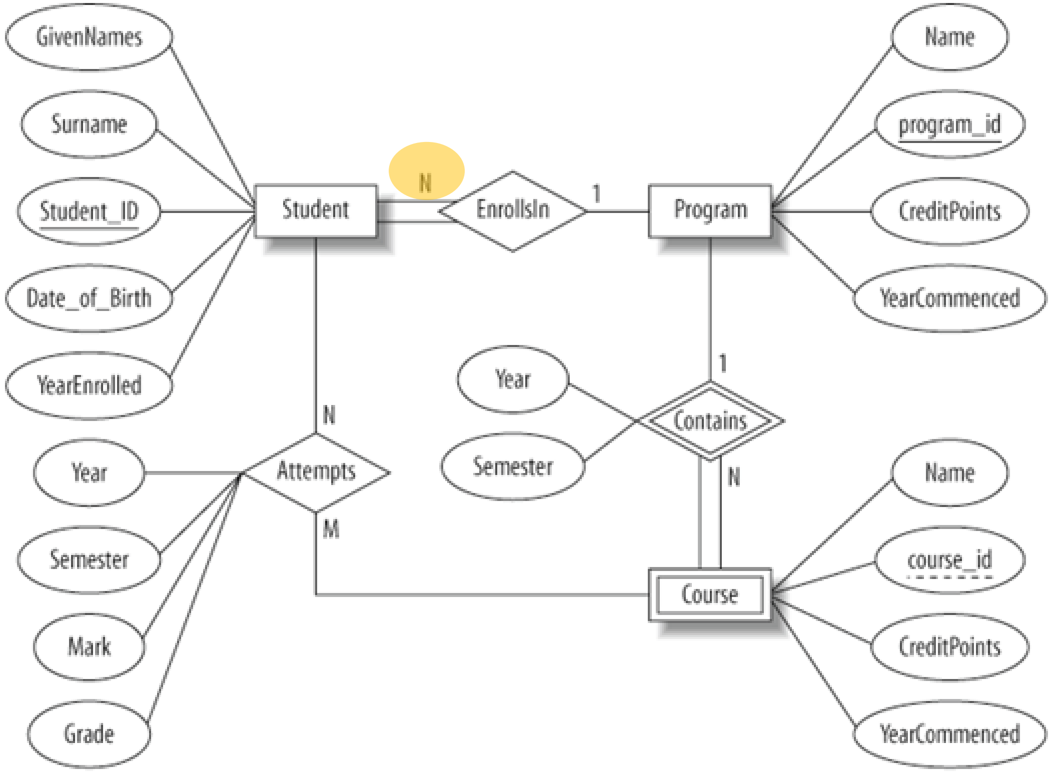
What have we learnt so far?

- A student **must** enrol in **one** program.
  - It has GivenNames, Surname, StudentID, Date_of_Birth, YearEnrolled attributes, where StudentID is a primary key.



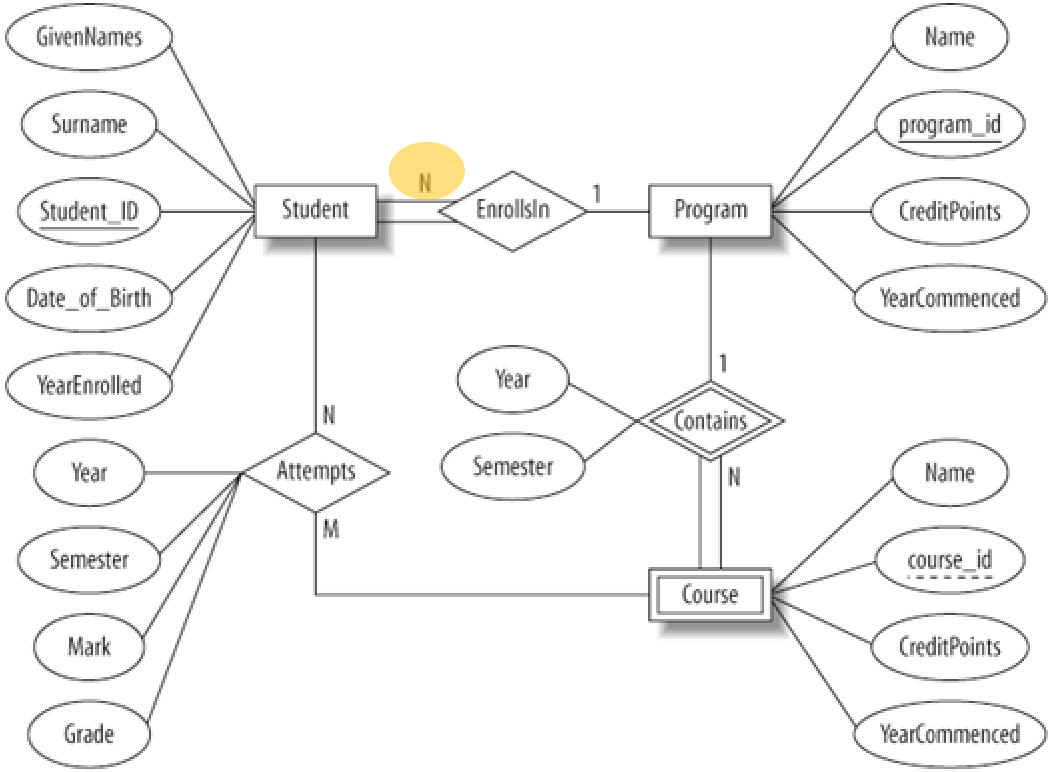
What have we learnt so far?

- Program …… have ….. students.



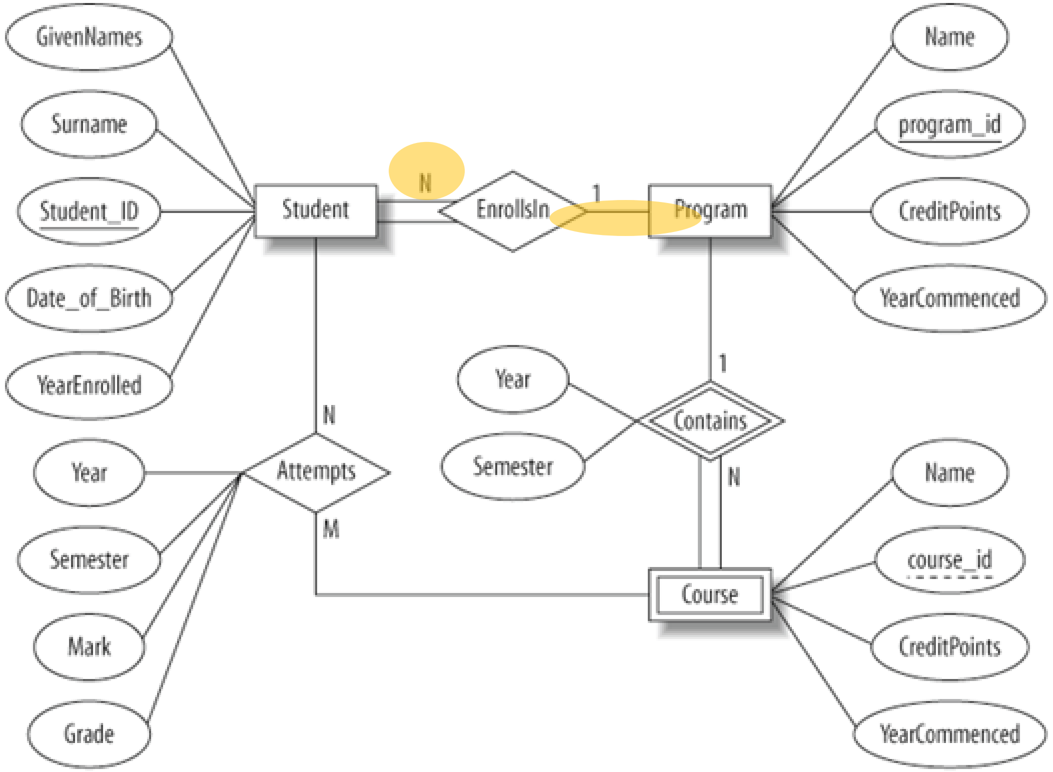
What have we learnt so far?

- Program **……** have **many** students.



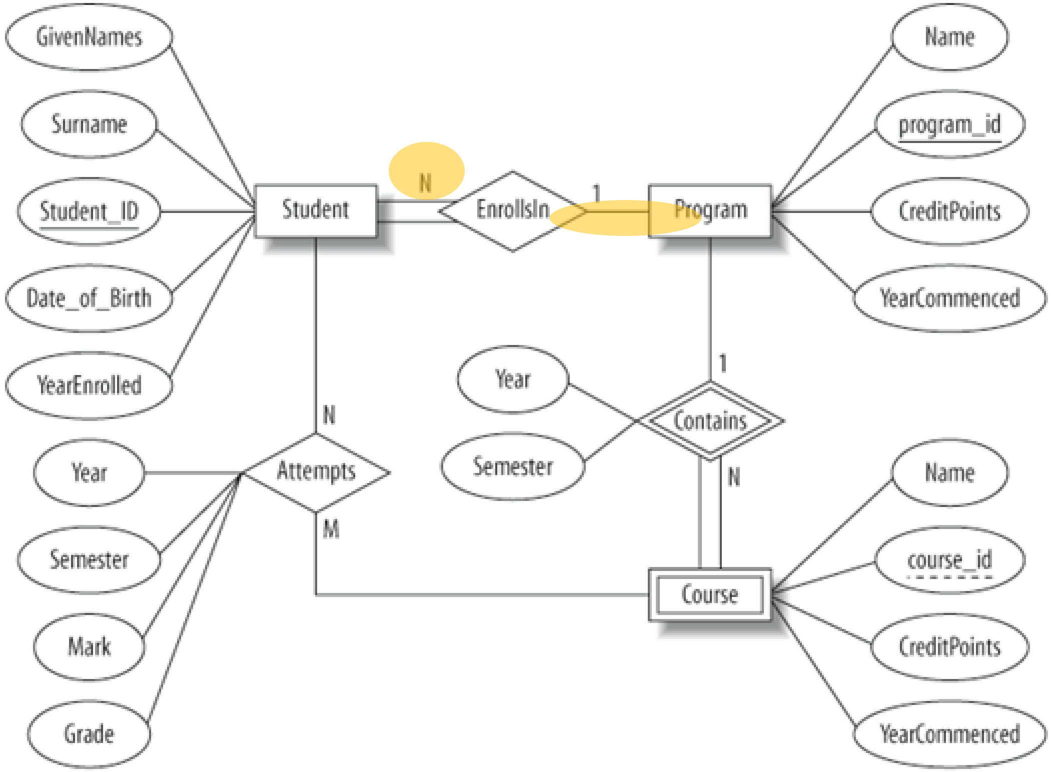
What have we learnt so far?

- Program **……** have **many** students.



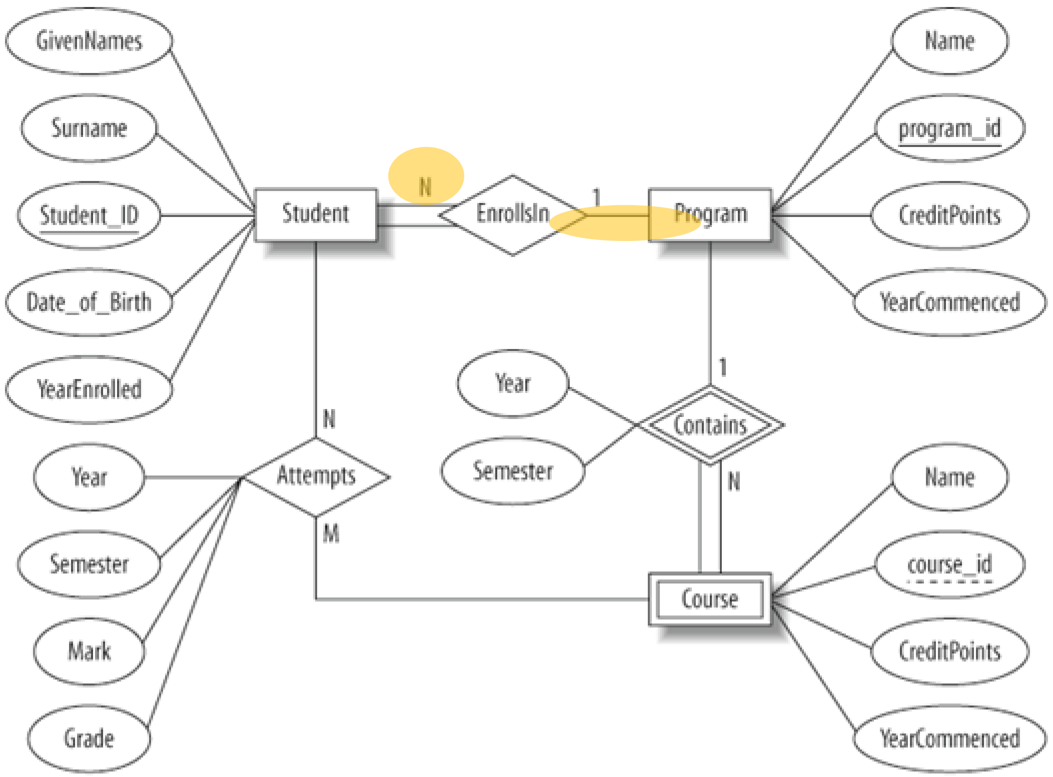
What have we learnt so far?

- Program **may** have **many** students.



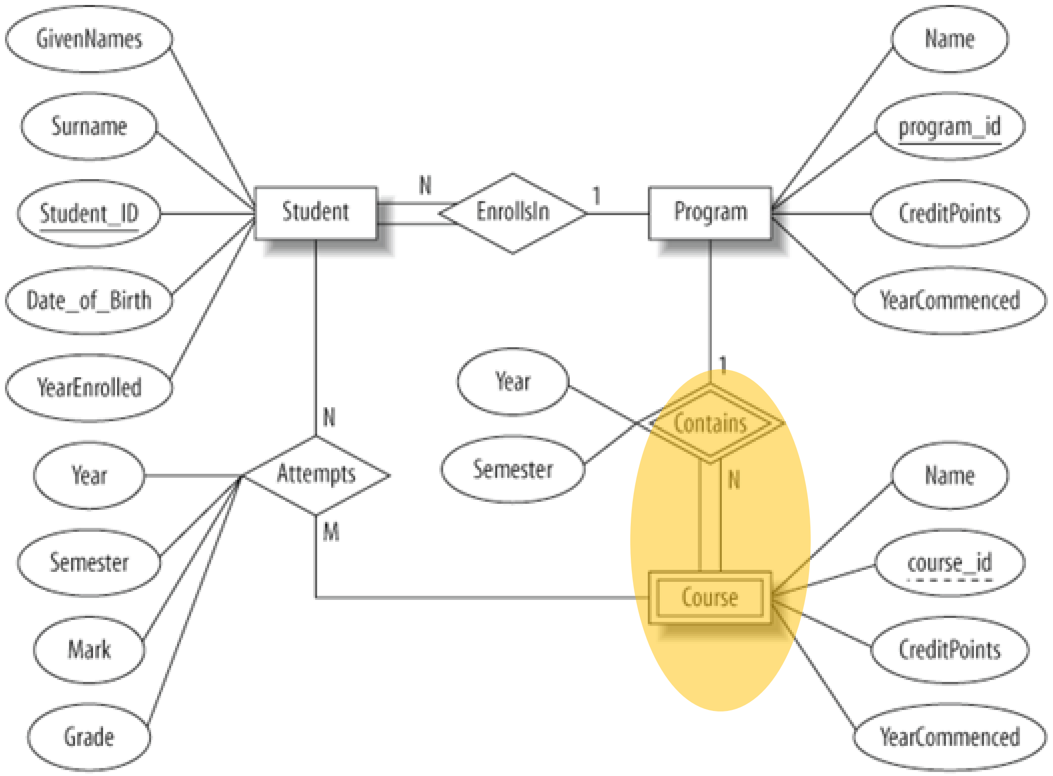
What have we learnt so far?

- Program **may** have **many** students.
  - It has name, program_iD, CreditPoints, YearCommenced where program_iD is a primary key.



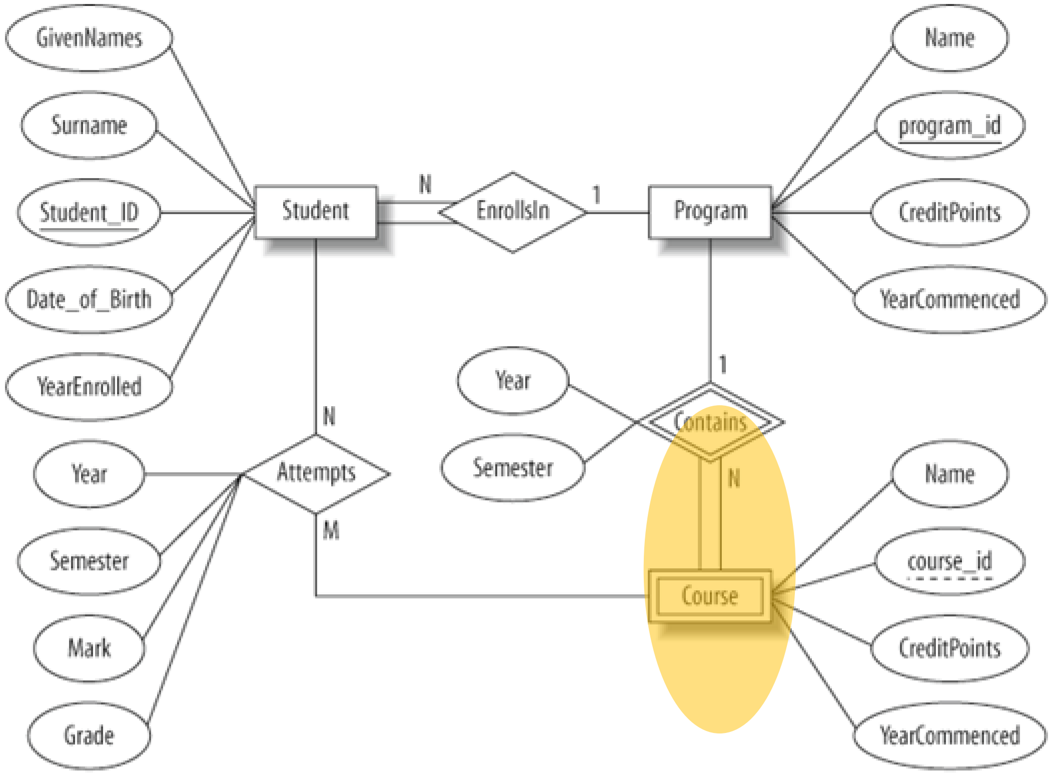
What have we learnt so far?

- When a program is deleted, the DBMS automatically deletes all related courses. (weak entity)



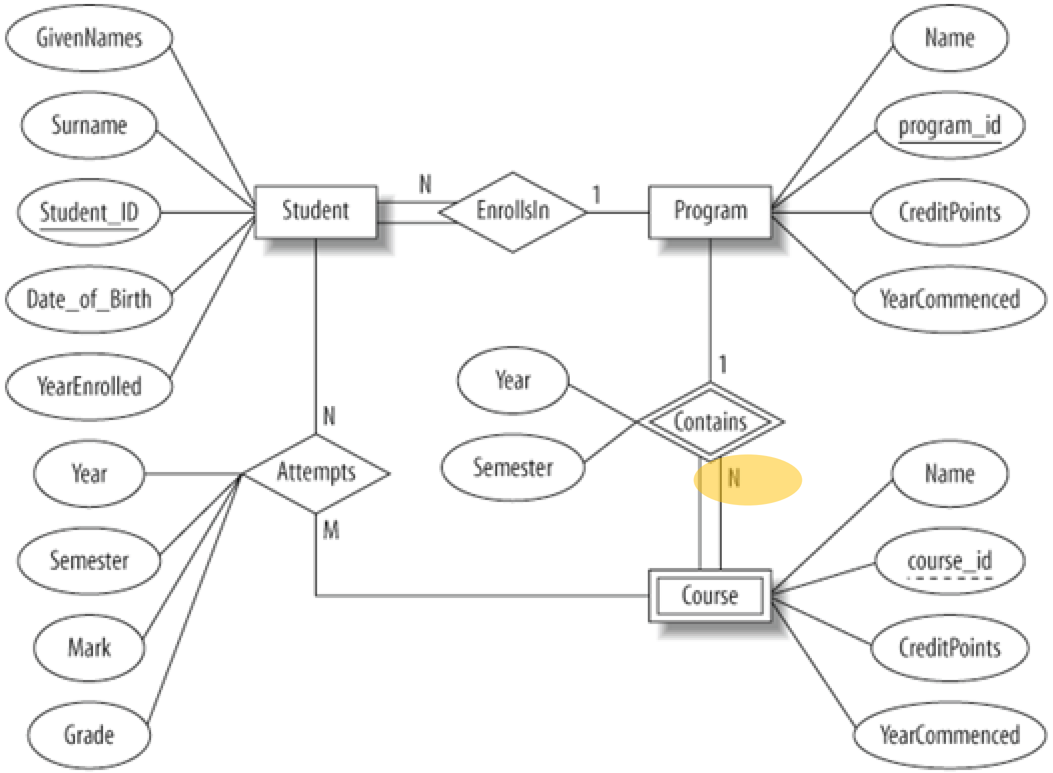
What have we learnt so far?

- When a program is deleted, the DBMS automatically deletes all related courses. (weak entity)
  - It has Name, course_id, CreditPoints, YearCommenced where course_id is the weak key.



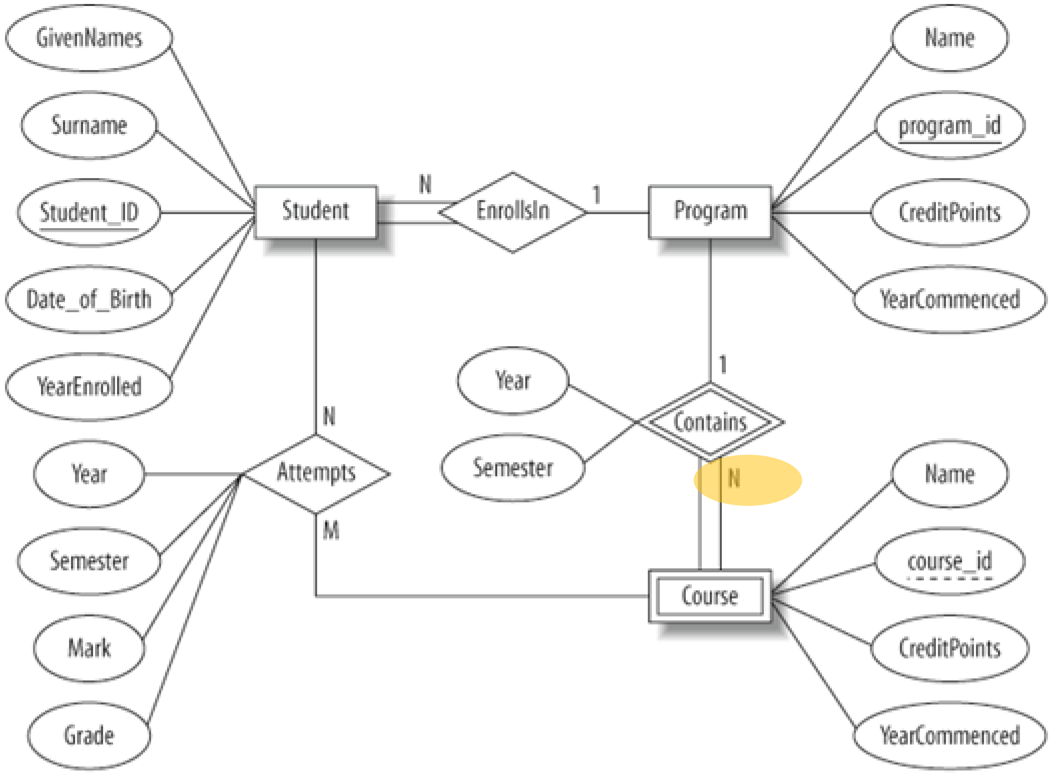
What have we learnt so far?

- Program ….. have ….. courses



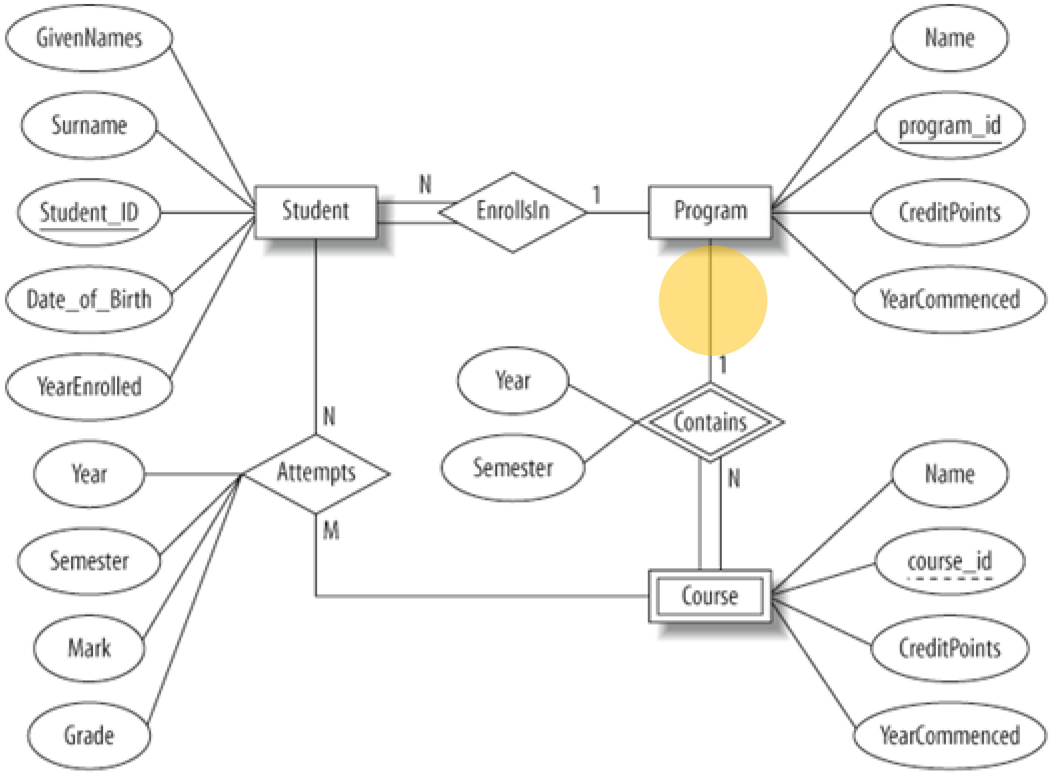
What have we learnt so far?

- Program …… have **many** courses



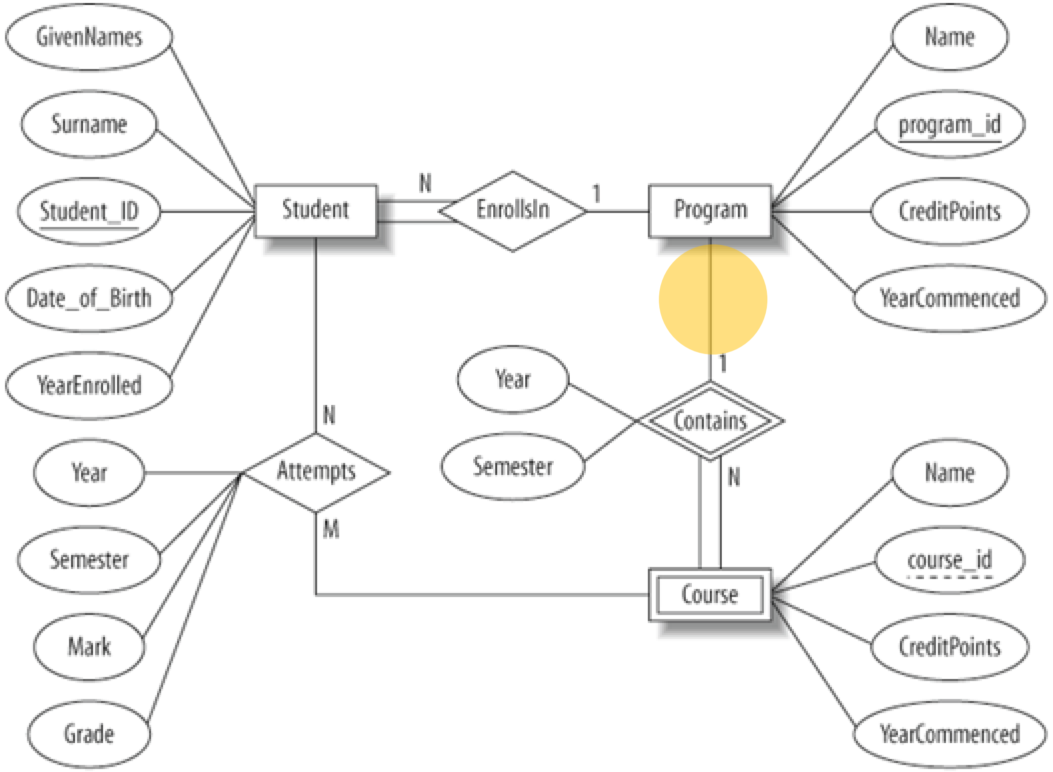
What have we learnt so far?

- Program …… have **many** courses



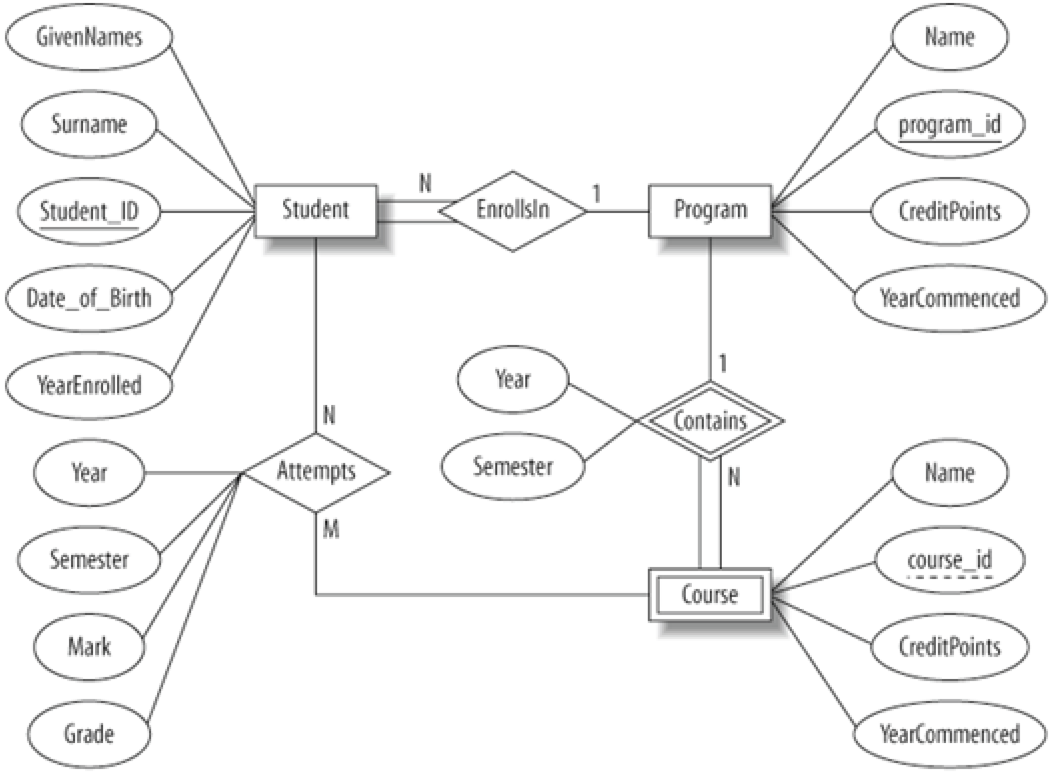
What have we learnt so far?

- Program **may** have **many** courses



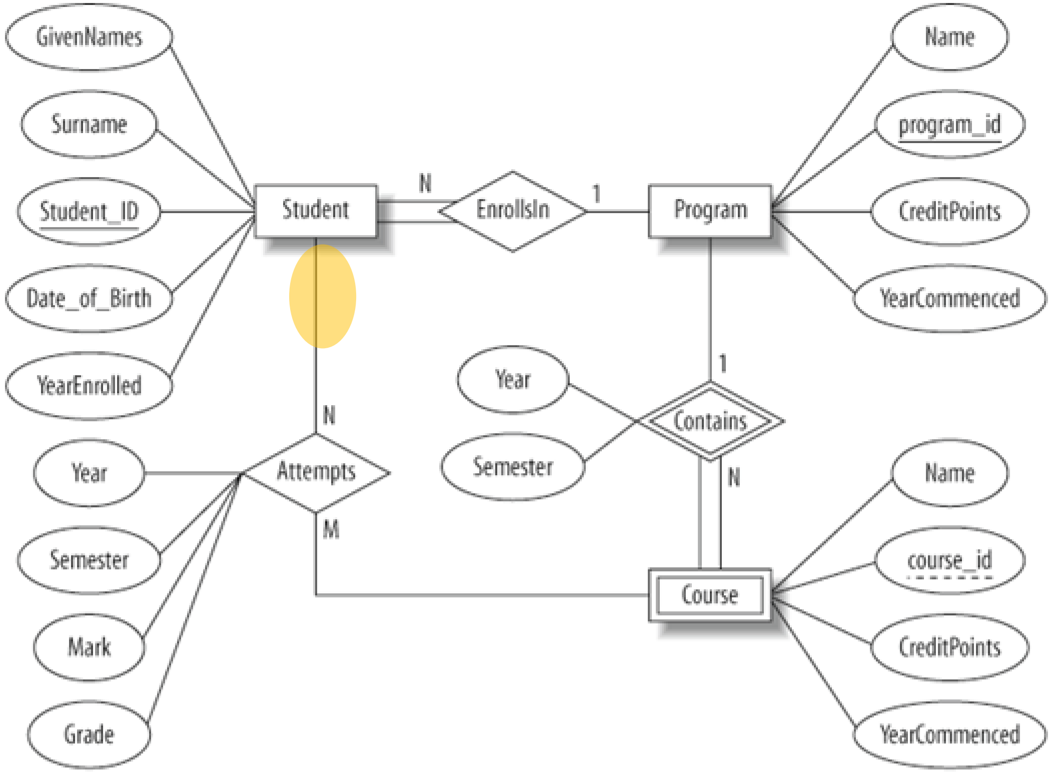
What have we learnt so far?

- A student **… attempt …** courses.
- A course **.. be attempted by ….** students.



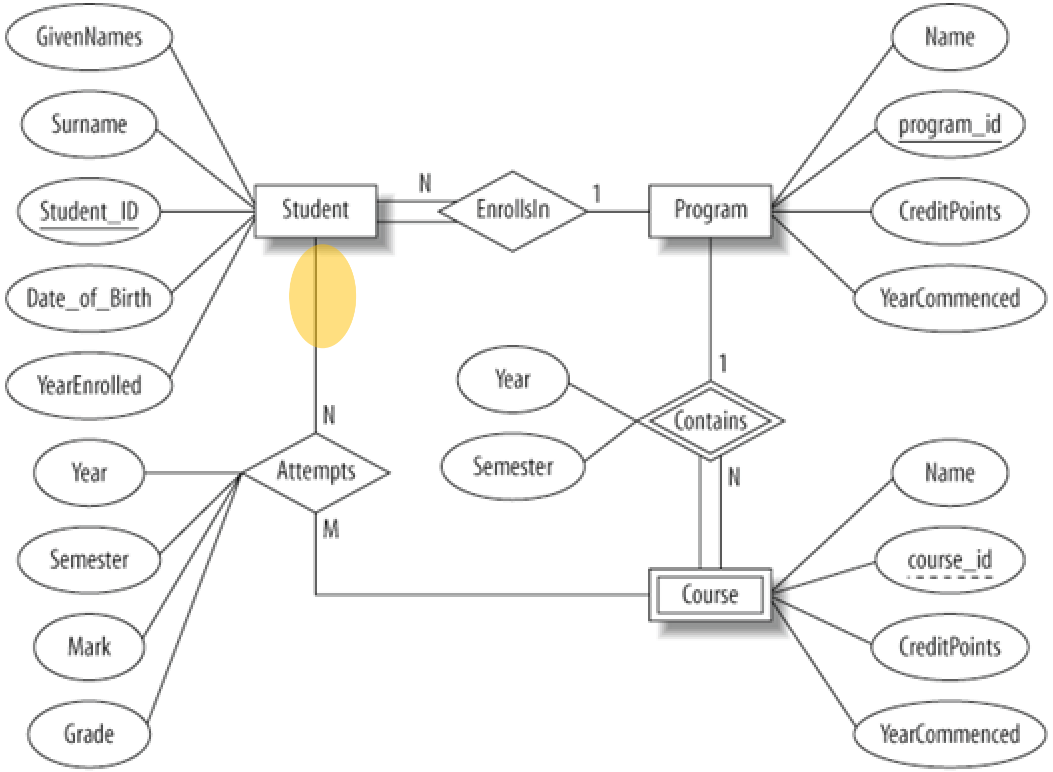
What have we learnt so far?

- A student **… attempt …** courses.
- A course **.. be attempted by ….** students.



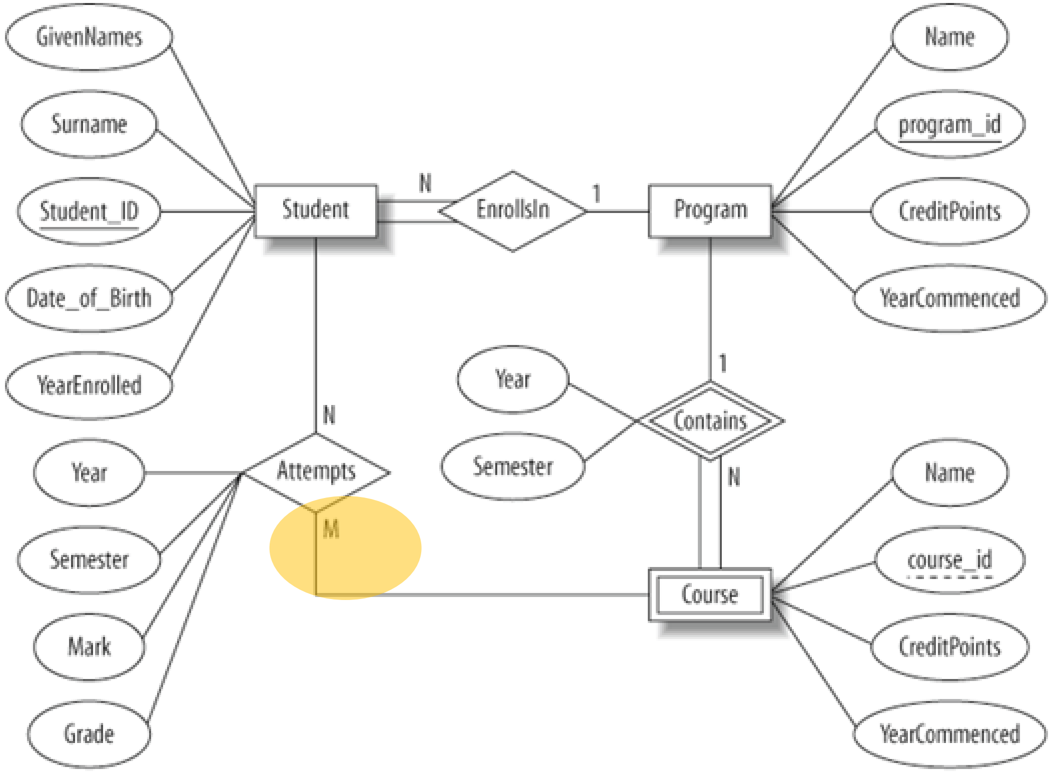
What have we learnt so far?

- A student **may attempt …** courses.
- A course **.. be attempted by ….** students.



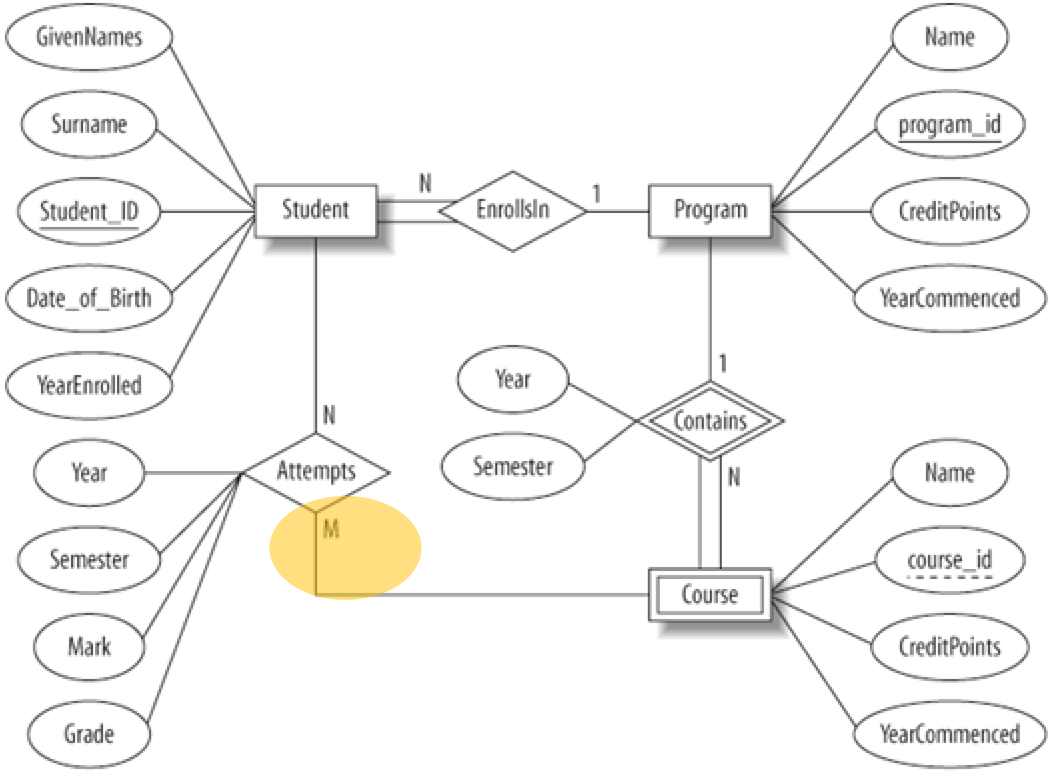
What have we learnt so far?

- A student **may attempt …** courses.
- A course **… be attempted by …** students.



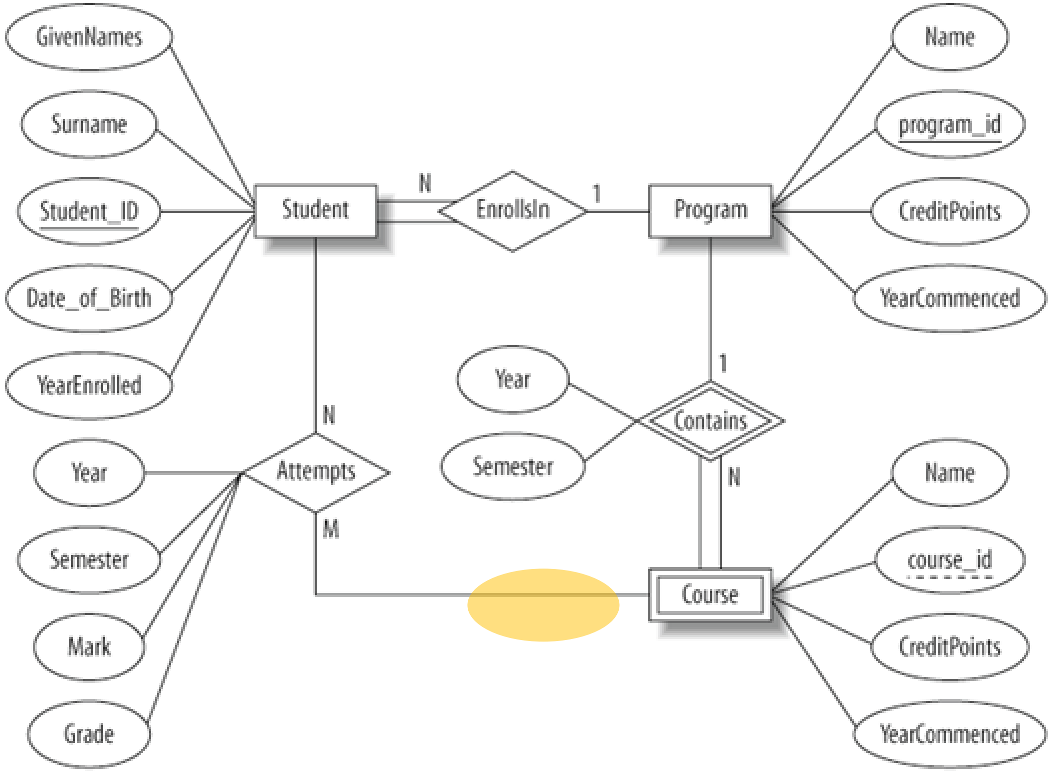
What have we learnt so far?

- A student **may attempt many** courses.
- A course **… be attempted by …** students.



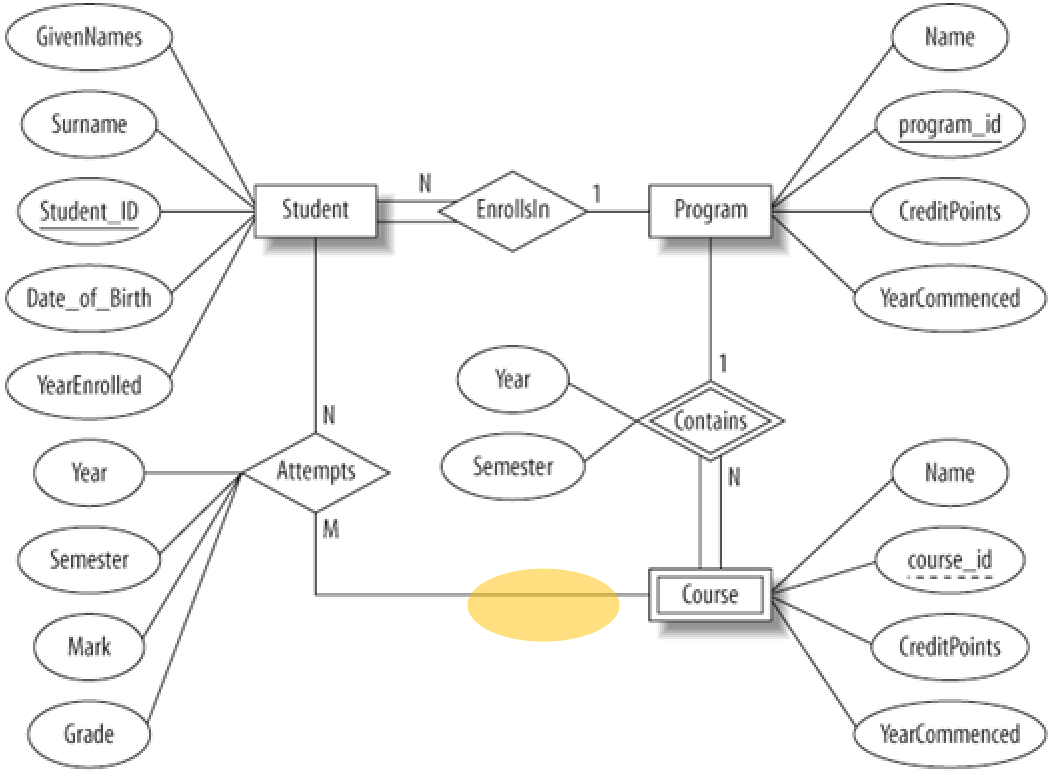
What have we learnt so far?

- A student **may attempt many** courses.
- A course **…. be attempted by ….** students.



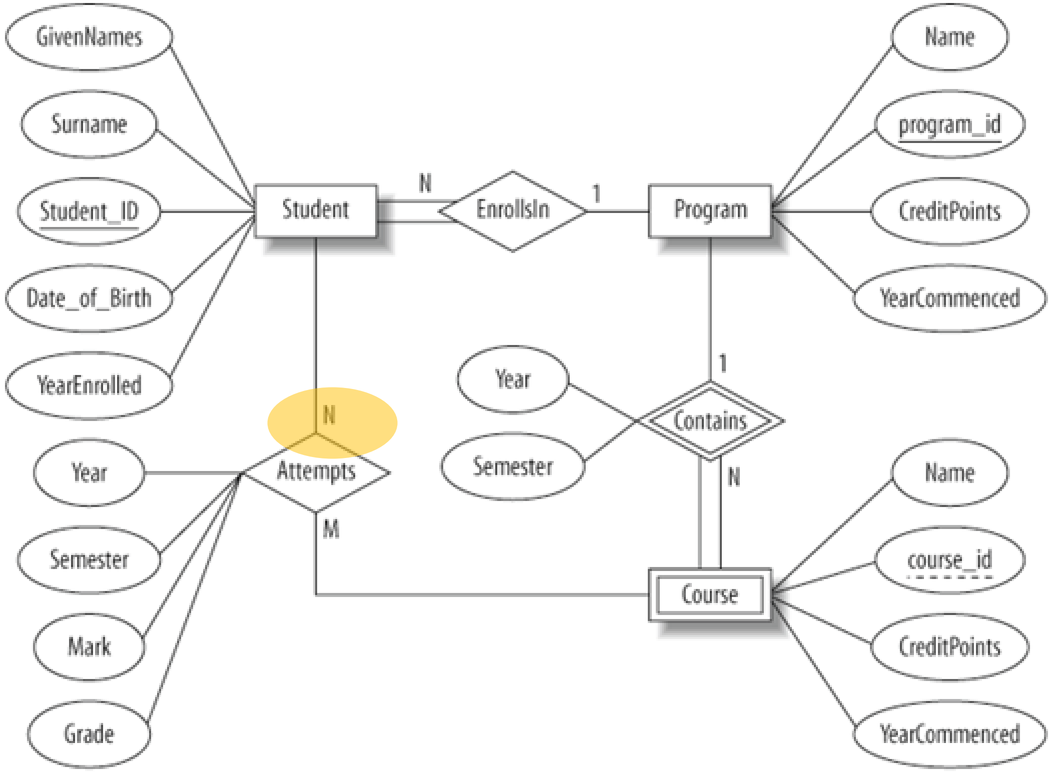
What have we learnt so far?

- A student **may attempt many** courses.
- A course **may be attempted by ….** students.



What have we learnt so far?

- A student **may attempt many** courses.
- A course **may be attempted by many** students.



Learning Outcomes

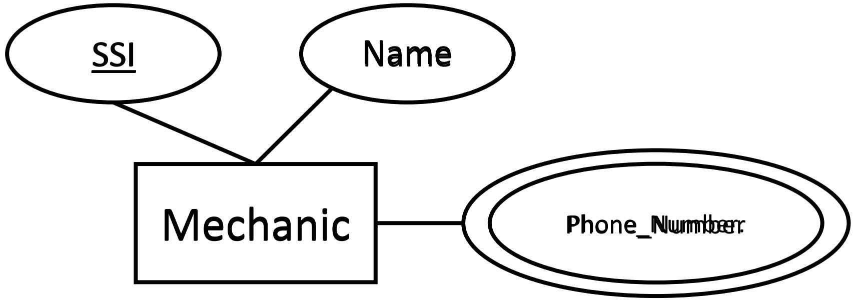
- You will learn concepts regarding the relational model, such as:
  - Integrity constraints, key constraints, schema, tuple, domain.
- After this week, you will learn how to
  - Convert a given ERD into the relational model.

This lecture

- We will start learning the **Relational Model**.
  - Schema and definition.
  - Integrity constraints.
  - Key constraints
  - Referential Integrity

That is, we will start learning correspondence between ERD and Relational database.

- ER diagram



- Relation.

SSI	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

Relational Database: Definitions

- Instead of Entity Sets and Entities
- We use **relation** as a *set* of rows or *tuples* (i.e., all rows are distinct) with description.

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2
53650	Shero	shero@math	19	3.8

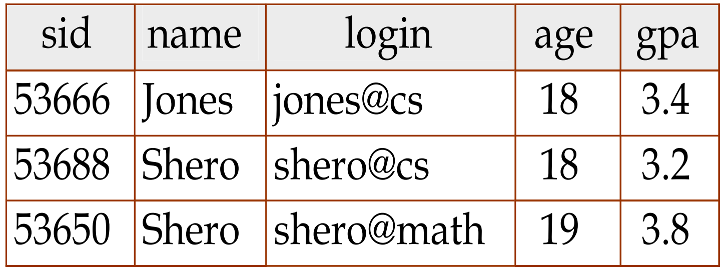
Rows/Tuples

Relational Database: Definitions

- *Relation is a mathematical statement*
- *Relation:* made up of 2 parts:

Instance: a *table* with rows and columns.

#Rows = *cardinality*, #fields/attributes = *degree / arity.*



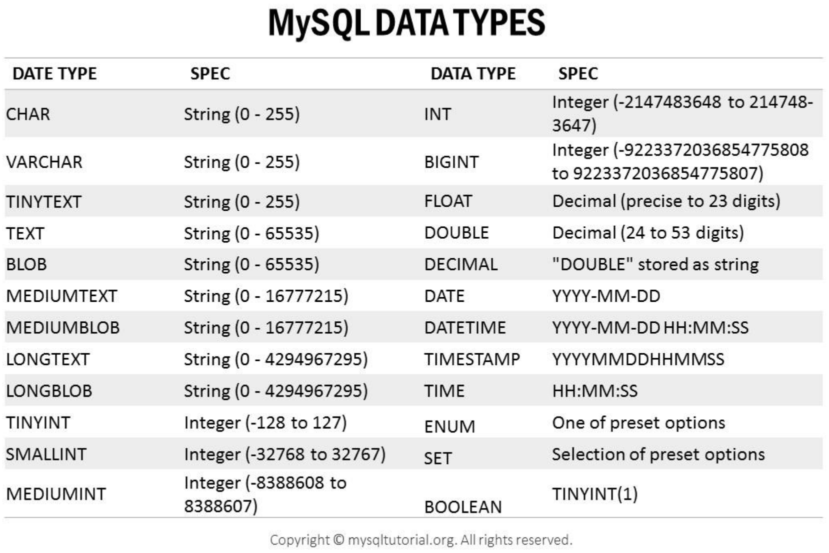
sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2
53650	Shero	shero@math	19	3.8

Relation Schema (RS): specifies the relation's name and **type (domain)** of each **column**.

Write the Relation Schema for the given Relation Instance.

Students(*sid*: INT, *name*: TEXT, *login*: TEXT, *age*: INT, *gpa*: DOUBLE).

Name of the attribute: Domain of the attribute



MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Lancaster University

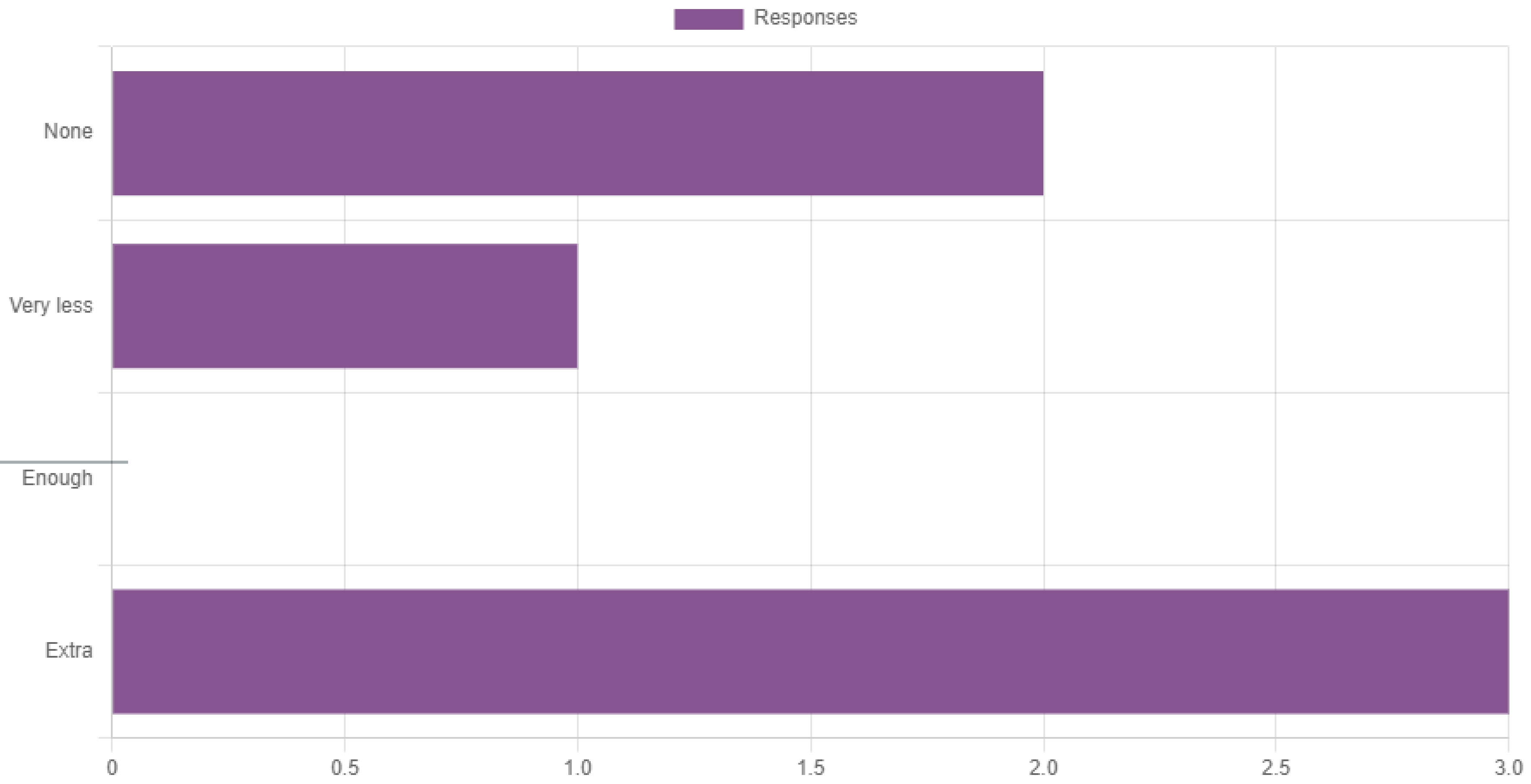
Week 2 ERD to Relational Model

We will begin at 09:00

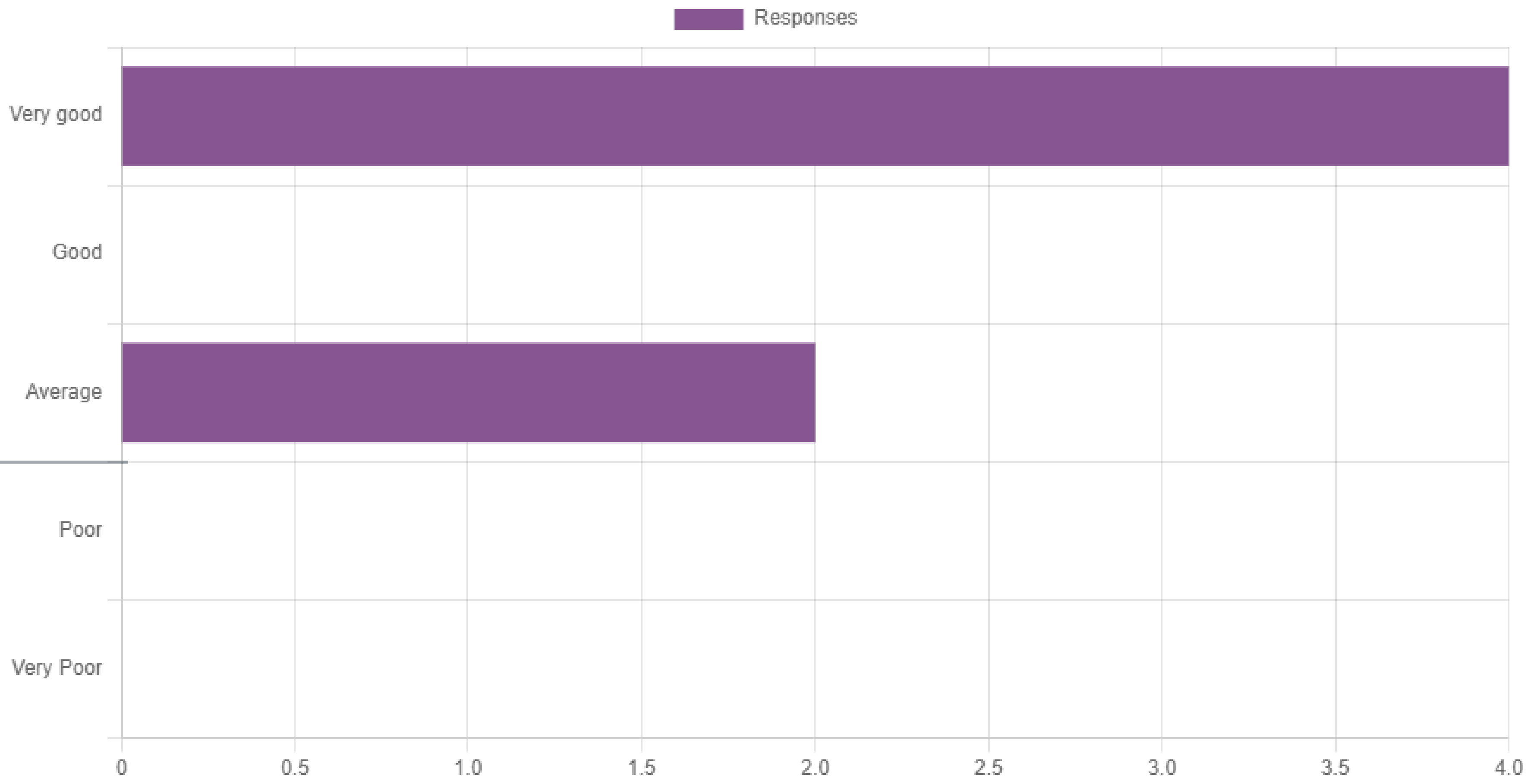
By Uraz…

From you…

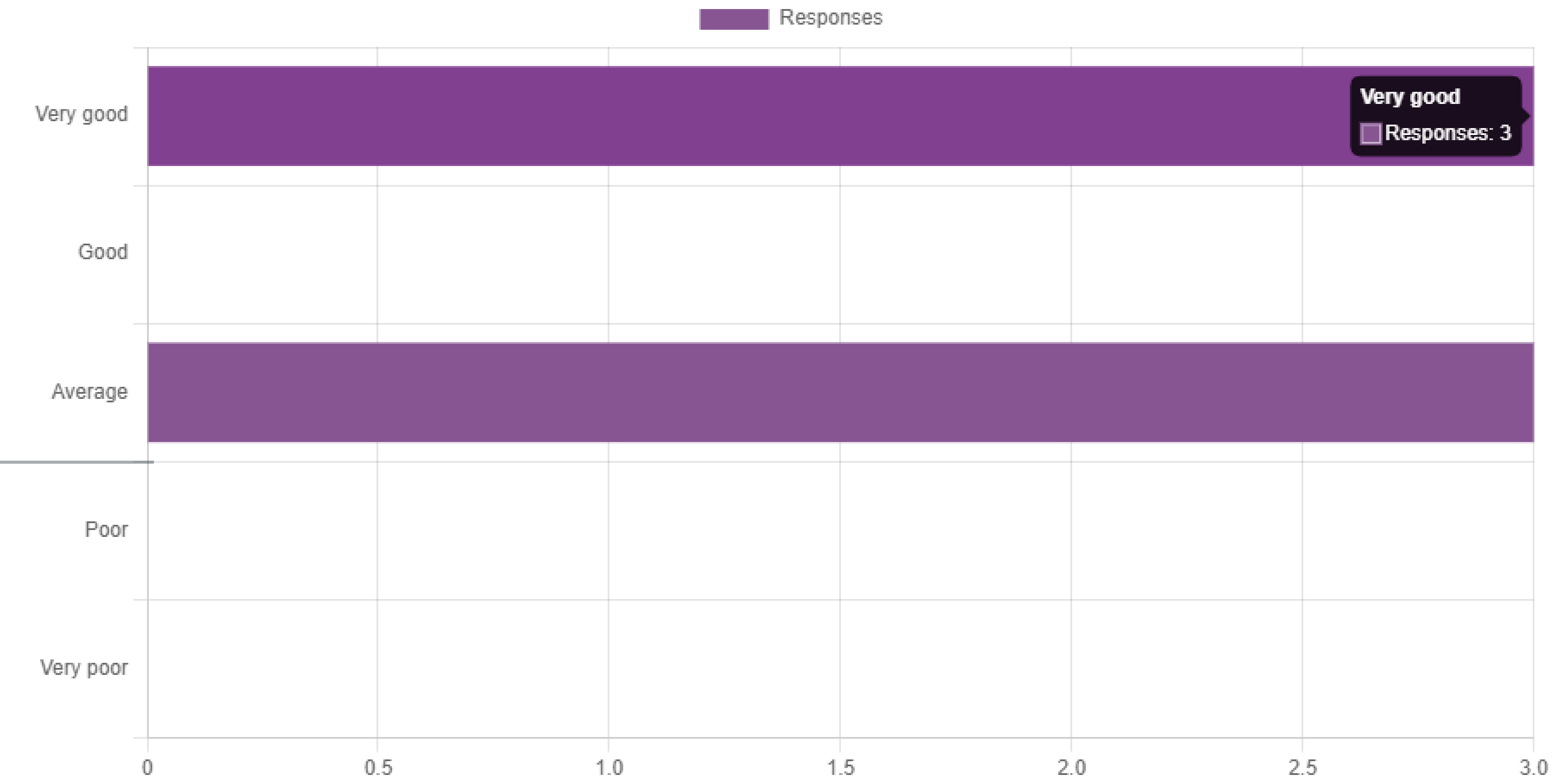
How was the support offered by module convenor?



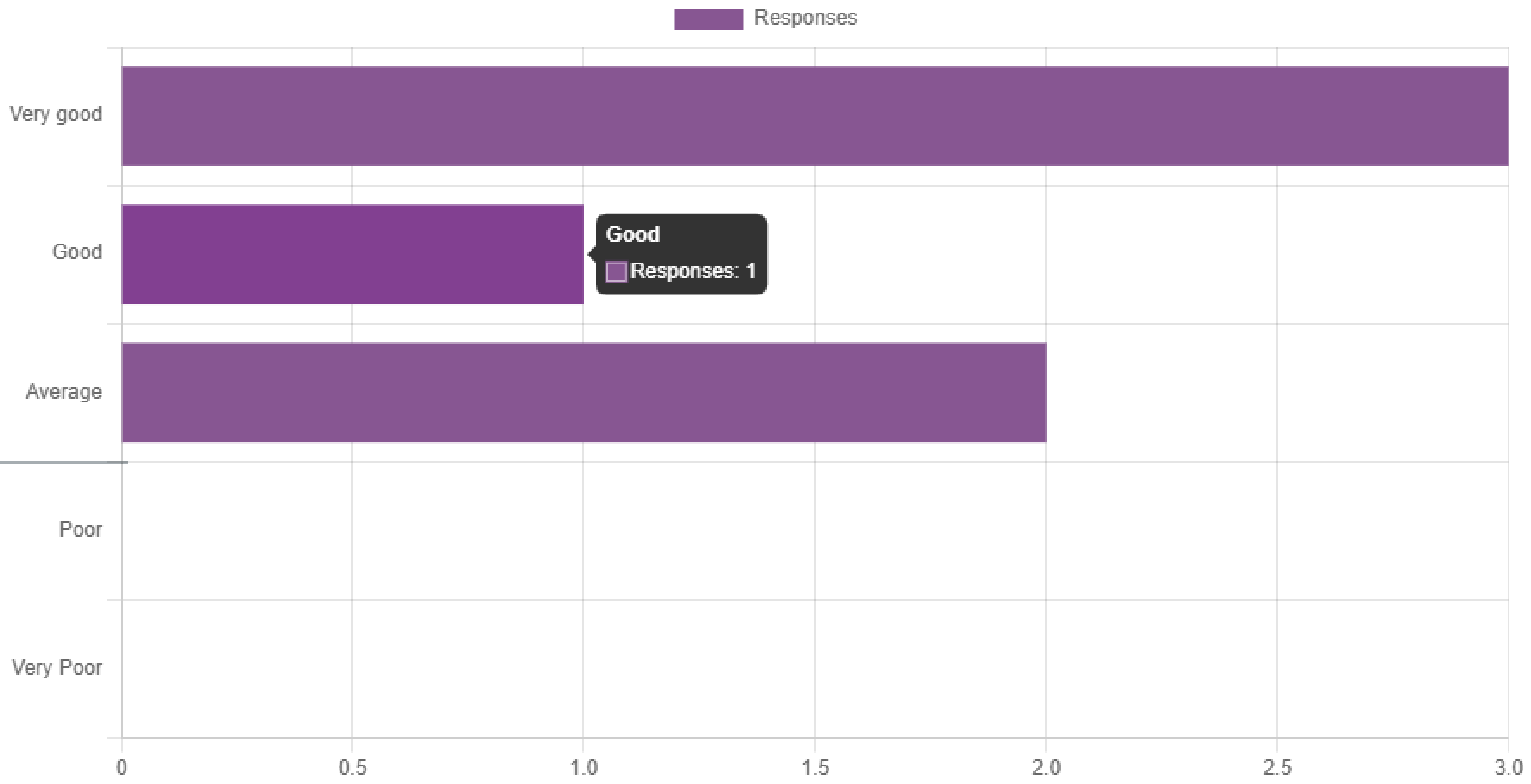
(slides) Helpfulness of the staff?

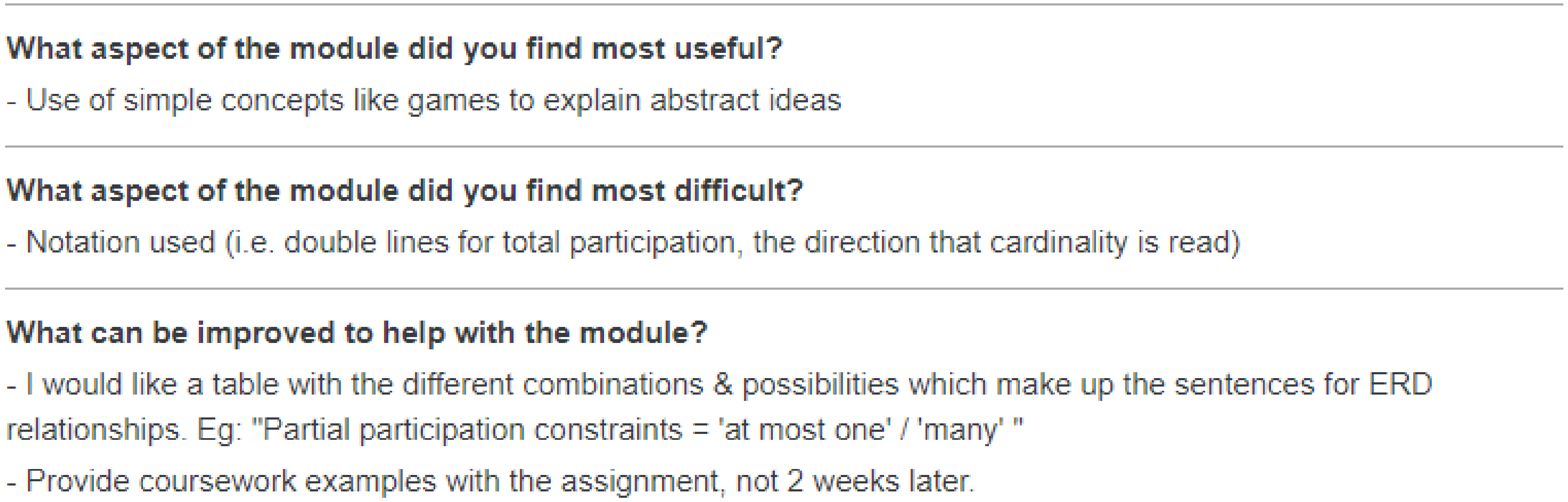


The module as a whole?



The quality of teaching?





What aspect of the module did you find most useful?

- Use of simple concepts like games to explain abstract ideas

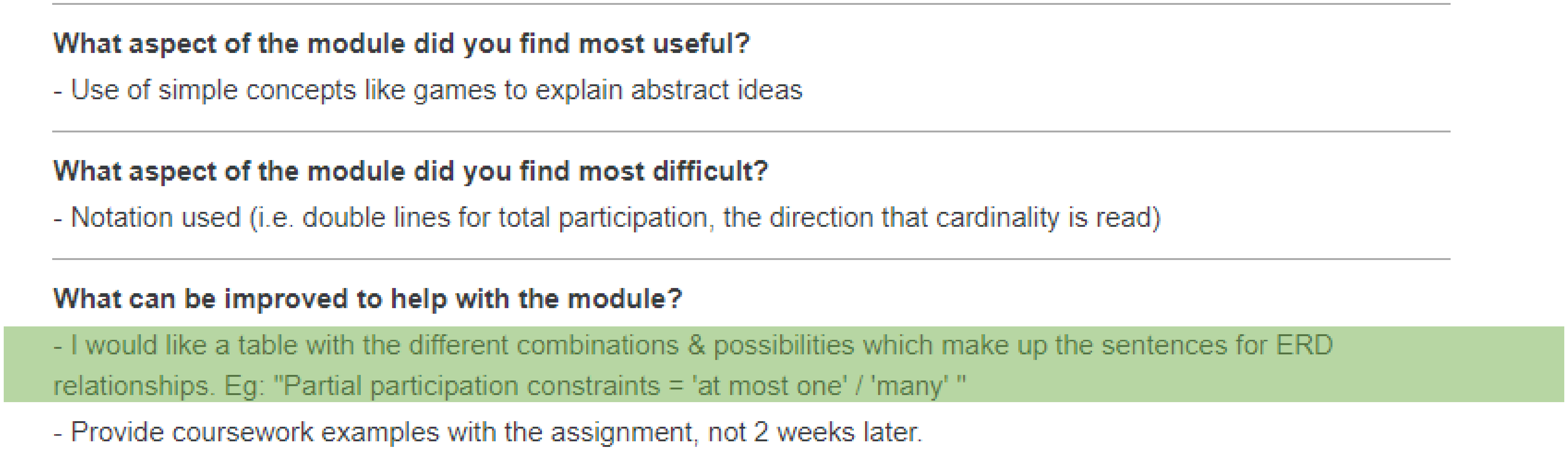
What aspect of the module did you find most difficult?

- Notation used (i.e. double lines for total participation, the direction that cardinality is read)

What can be improved to help with the module?

- I would like a table with the different combinations & possibilities which make up the sentences for ERD relationships. Eg: "Partial participation constraints = 'at most one' / 'many' "

- Provide coursework examples with the assignment, not 2 weeks later.



What aspect of the module did you find most useful?

- Use of simple concepts like games to explain abstract ideas

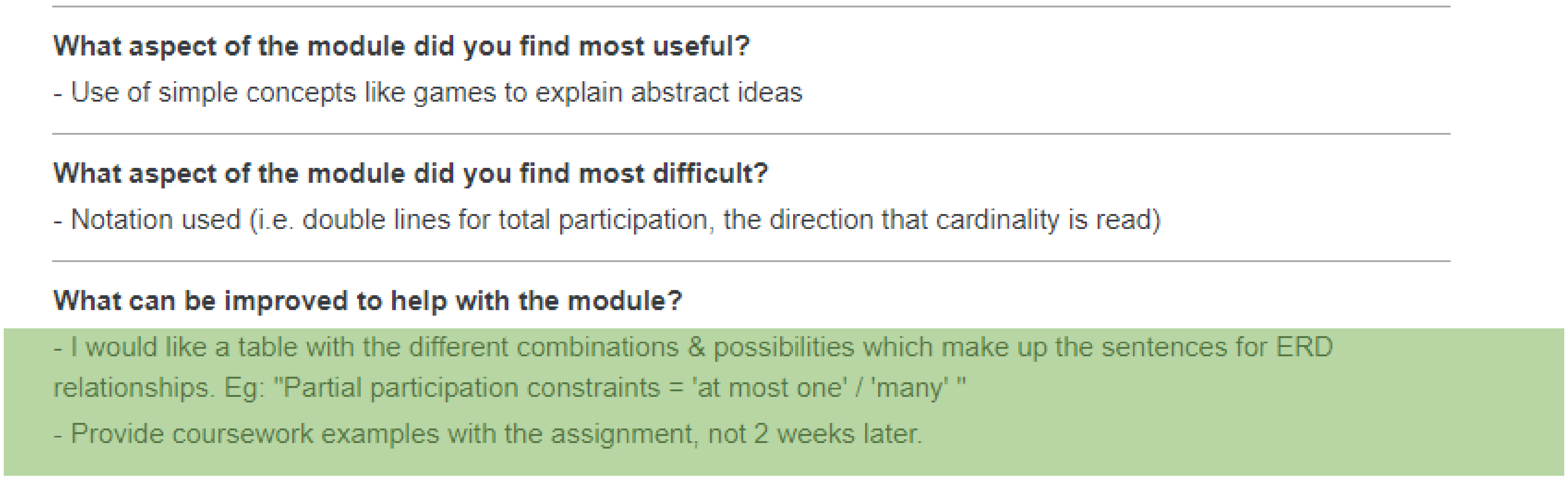
What aspect of the module did you find most difficult?

- Notation used (i.e. double lines for total participation, the direction that cardinality is read)

What can be improved to help with the module?

- I would like a table with the different combinations & possibilities which make up the sentences for ERD relationships. Eg: "Partial participation constraints = 'at most one' / 'many' "

- Provide coursework examples with the assignment, not 2 weeks later.



What aspect of the module did you find most useful?

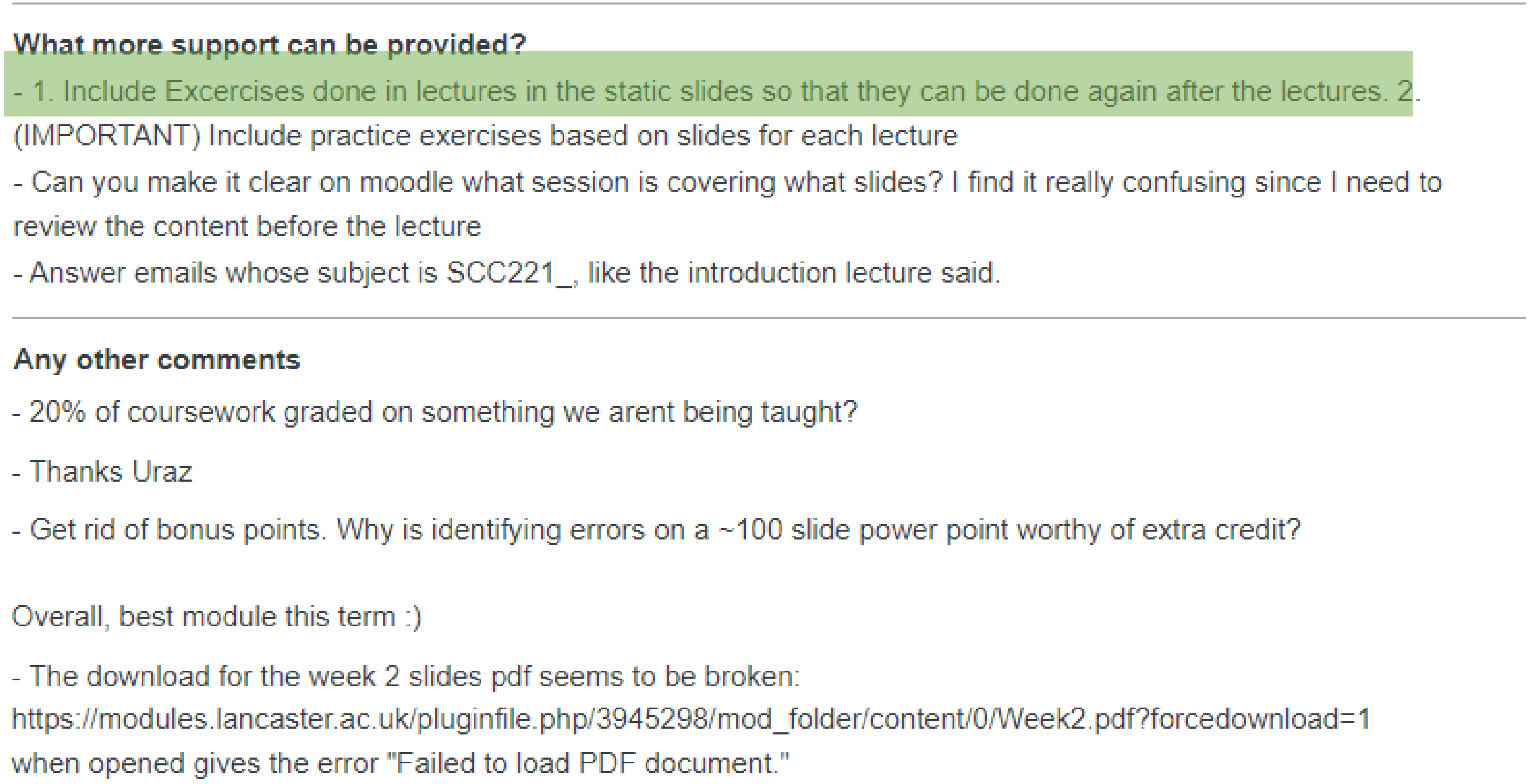
- Use of simple concepts like games to explain abstract ideas

What aspect of the module did you find most difficult?

- Notation used (i.e. double lines for total participation, the direction that cardinality is read)

What can be improved to help with the module?

- I would like a table with the different combinations & possibilities which make up the sentences for ERD relationships. Eg: "Partial participation constraints = 'at most one' / 'many' "
- Provide coursework examples with the assignment, not 2 weeks later.



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

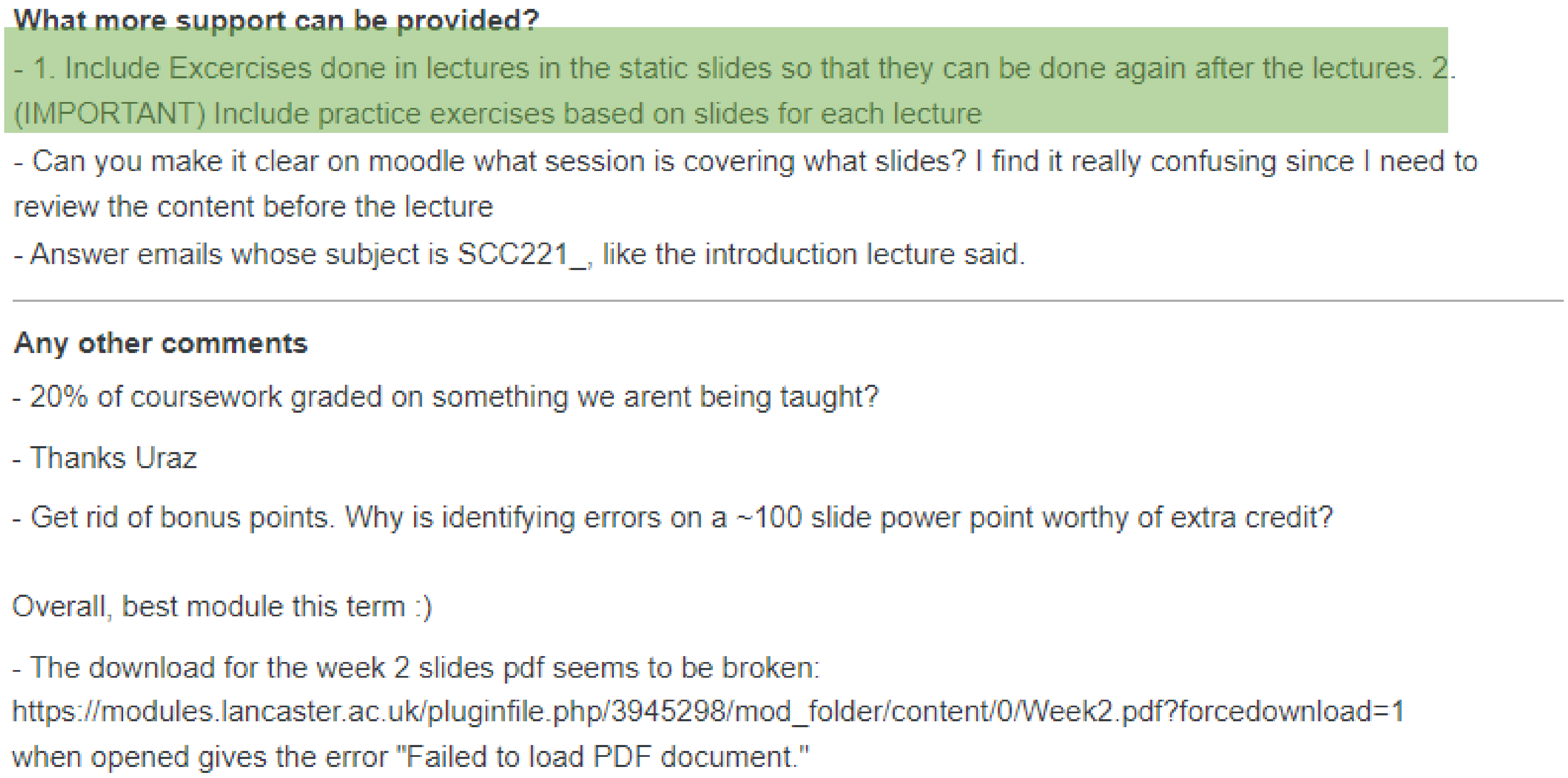
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

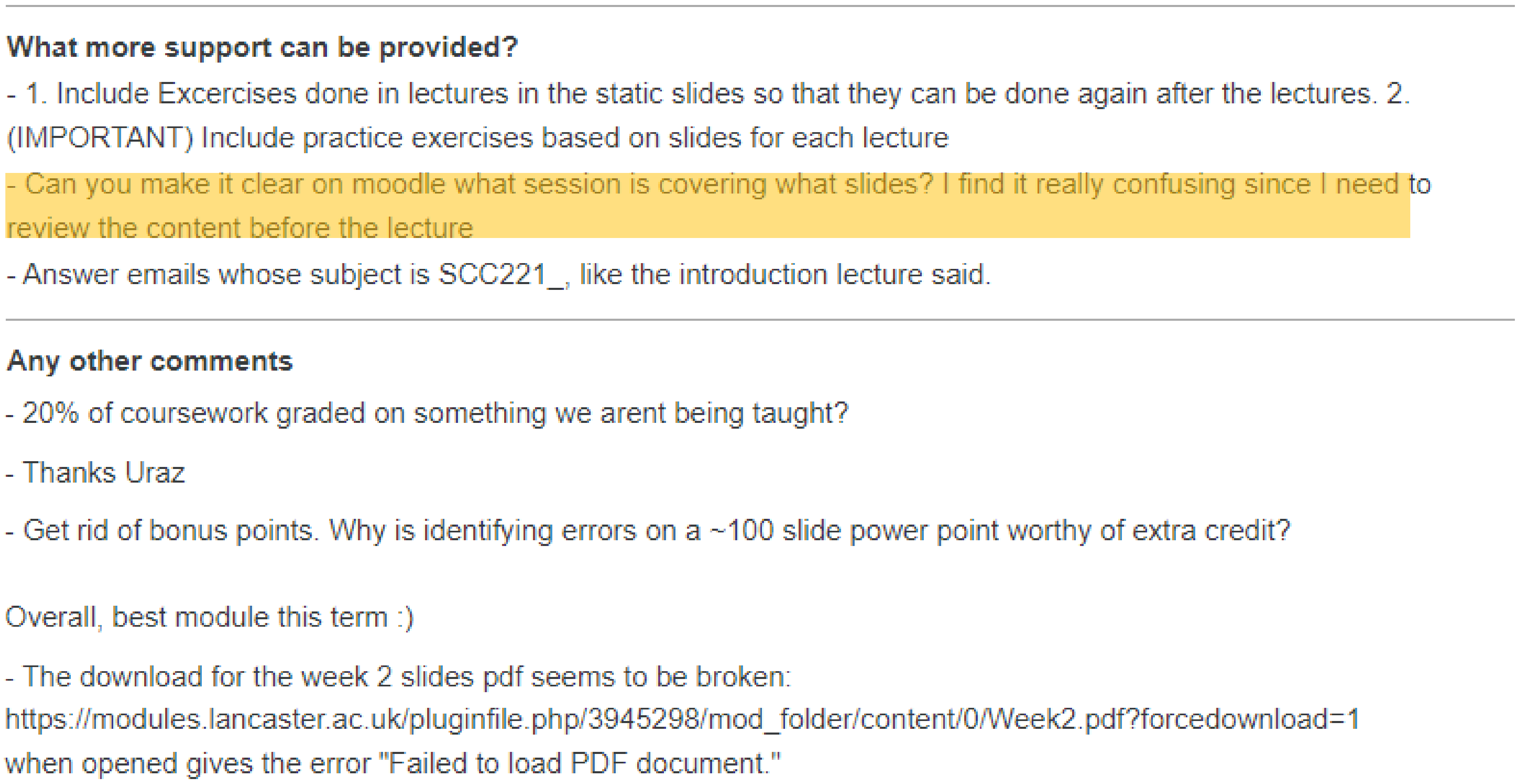
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

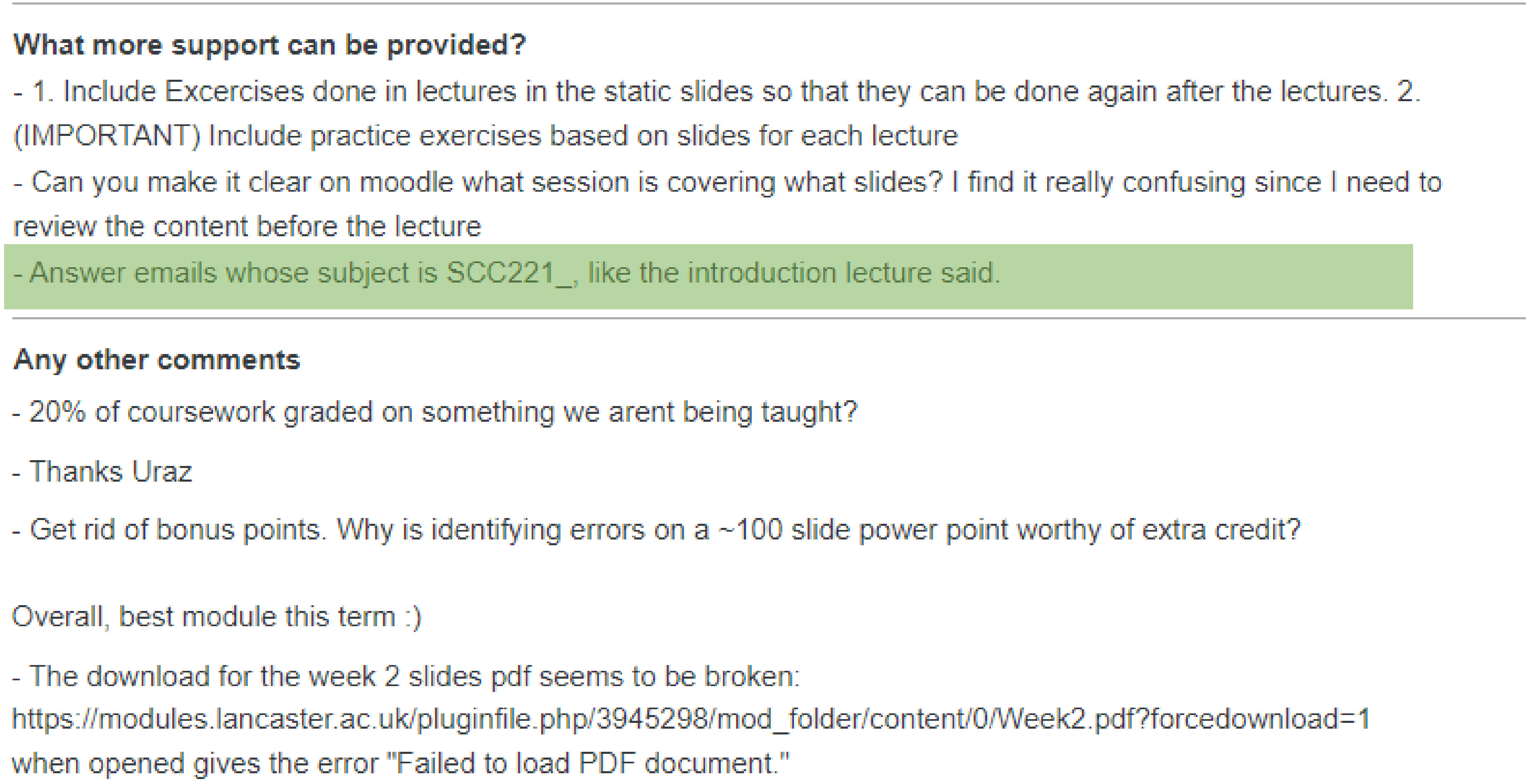
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

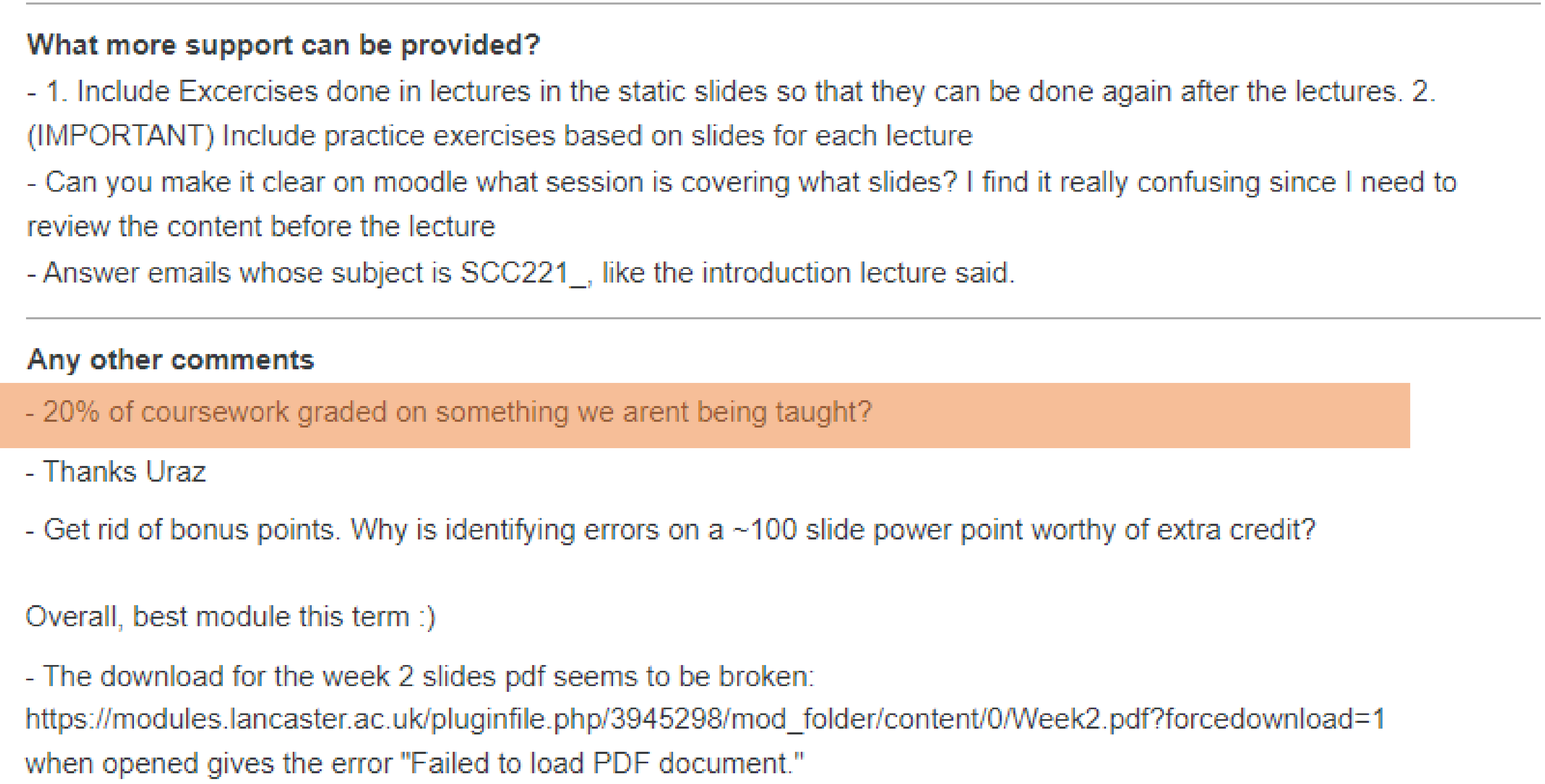
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken:
https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1
when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

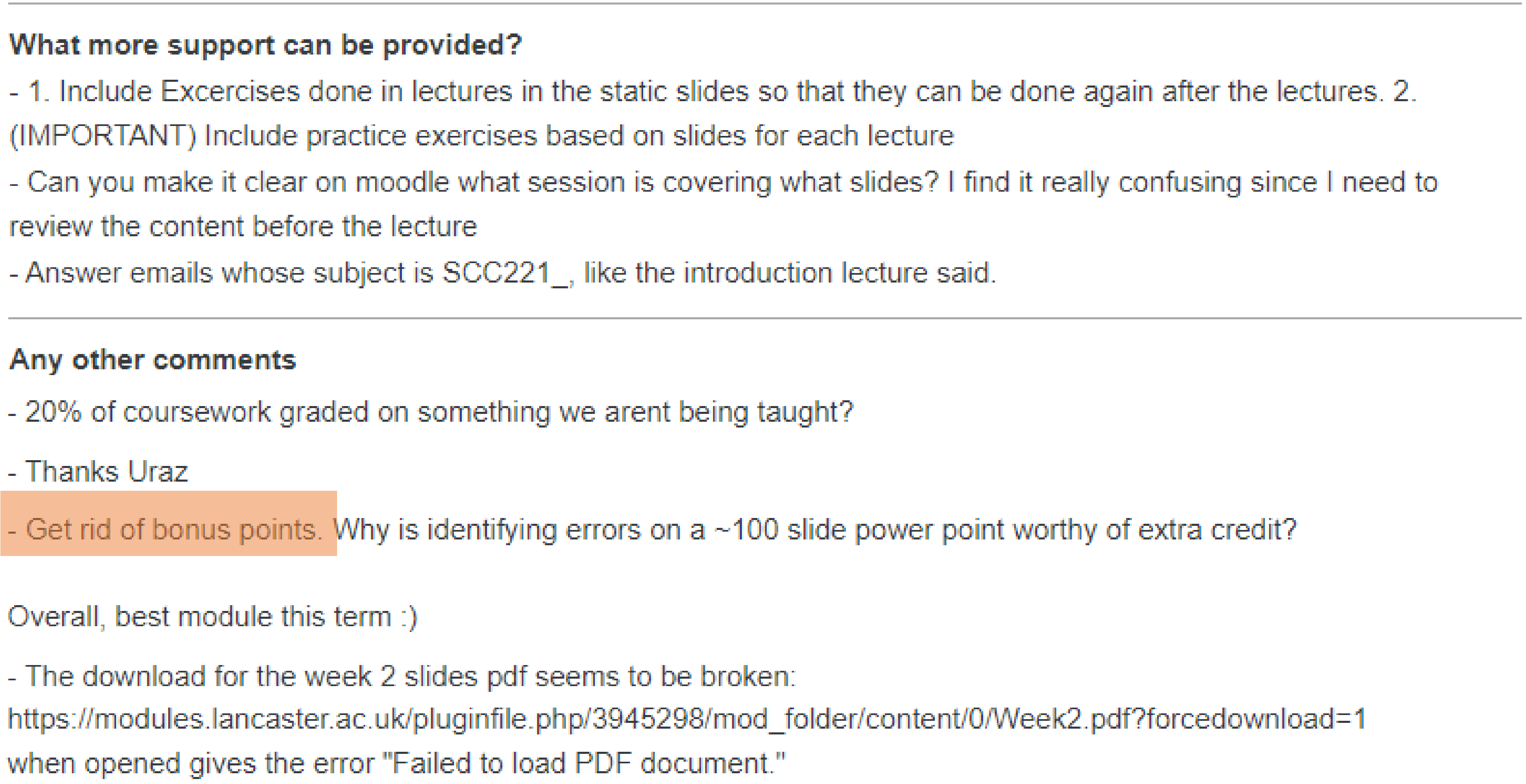
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

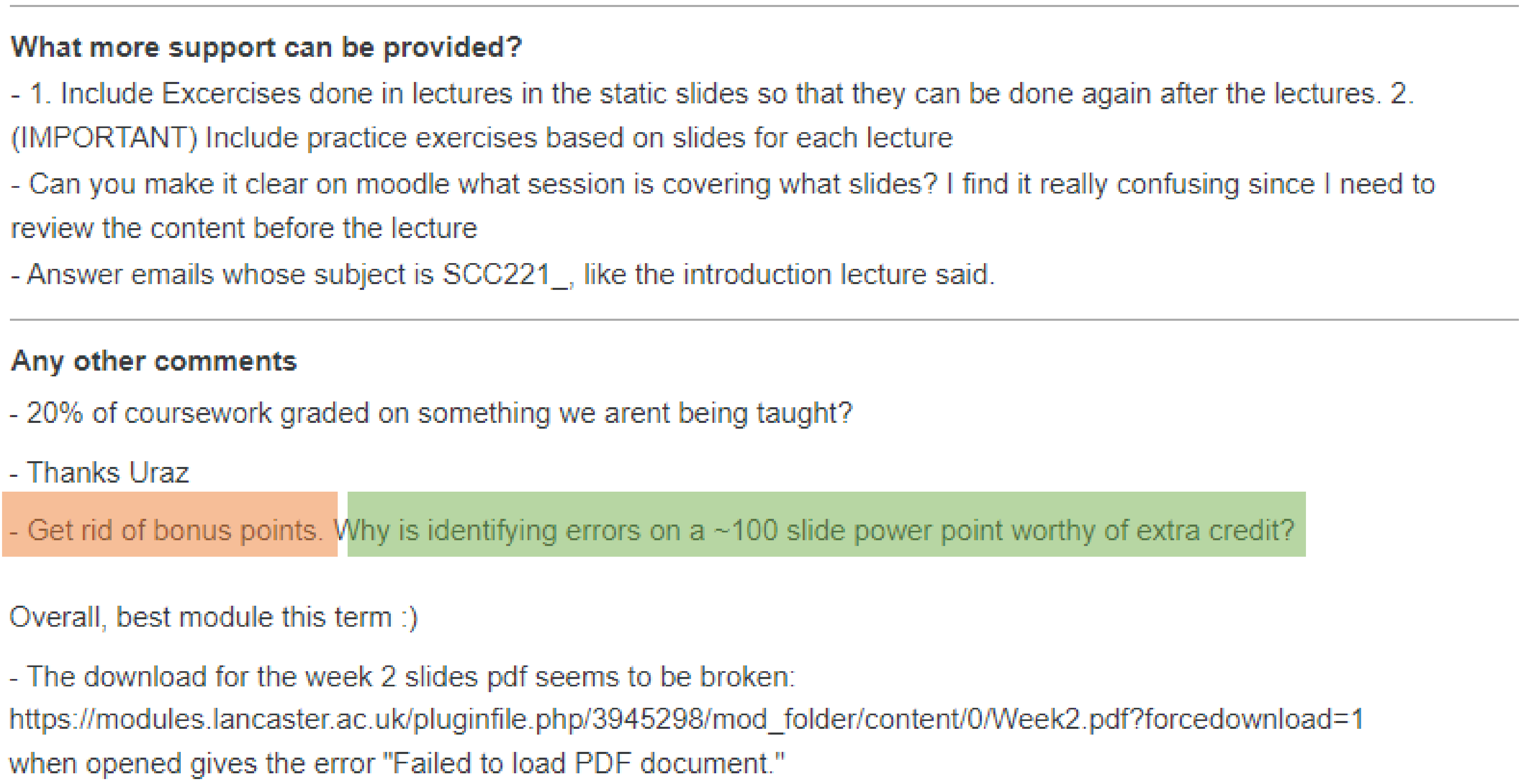
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture

- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture

- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

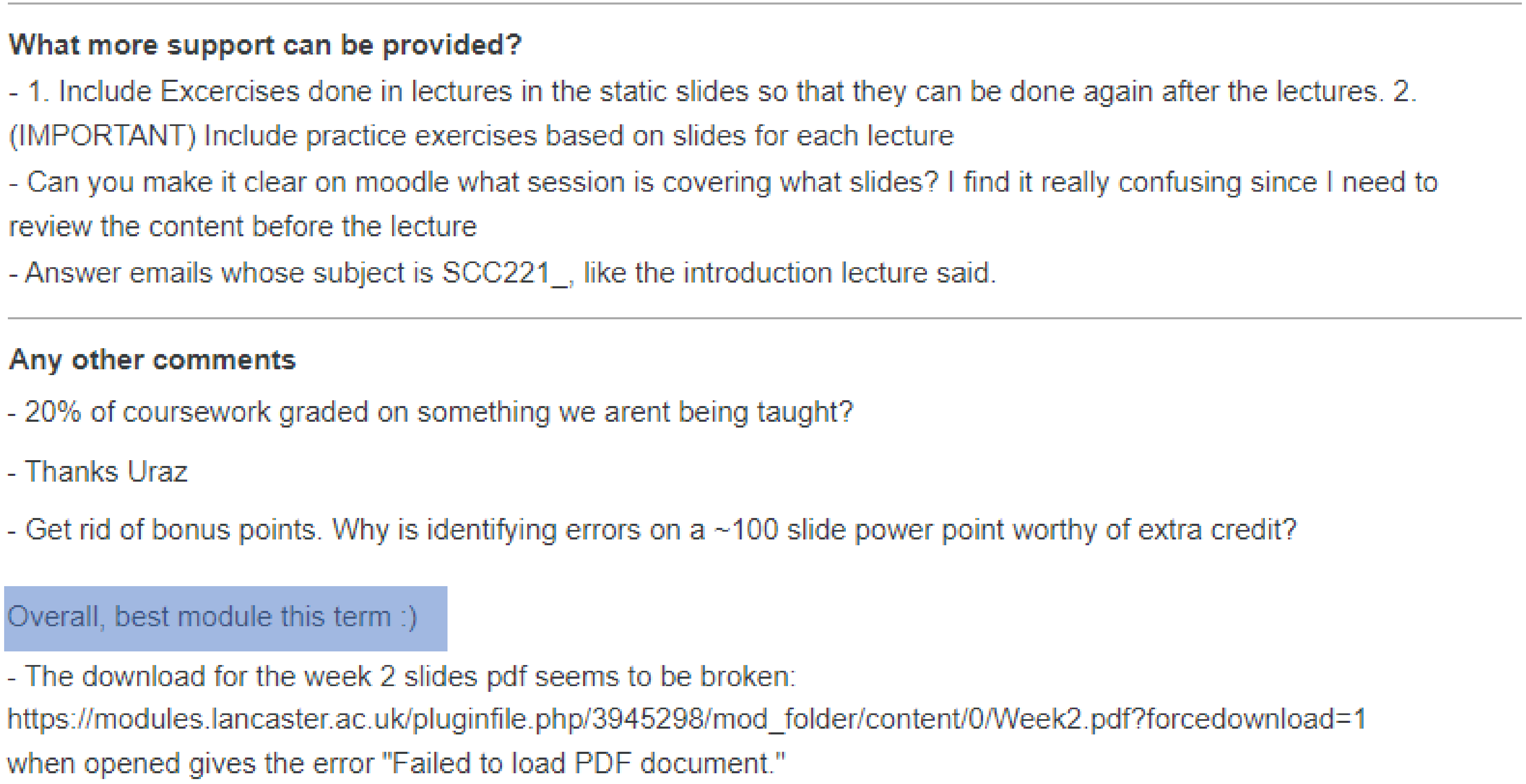
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken:
https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1
when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture
- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture
- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

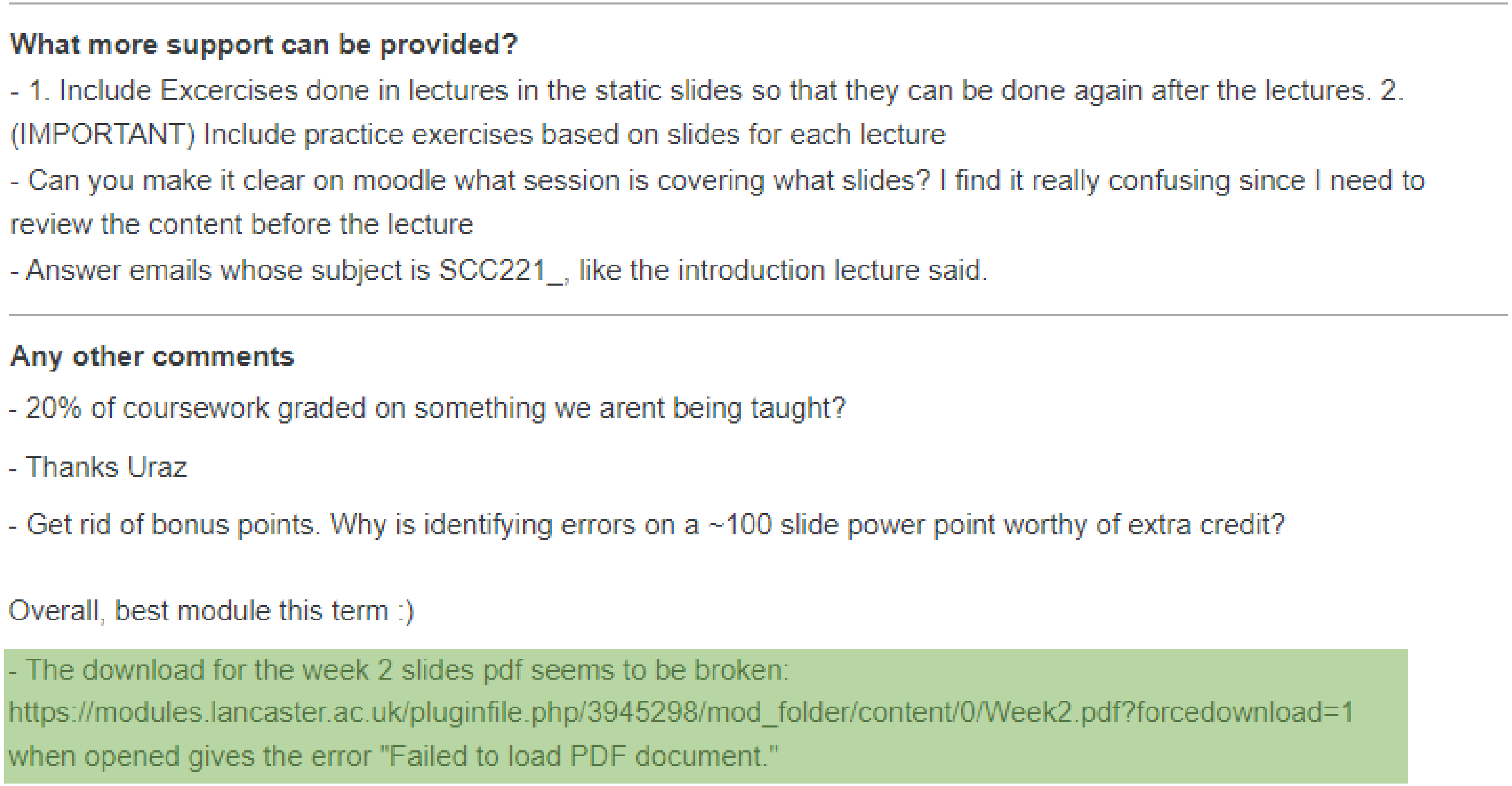
- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."



What more support can be provided?

- 1. Include Excercises done in lectures in the static slides so that they can be done again after the lectures. 2. (IMPORTANT) Include practice exercises based on slides for each lecture
- Can you make it clear on moodle what session is covering what slides? I find it really confusing since I need to review the content before the lecture
- Answer emails whose subject is SCC221_, like the introduction lecture said.

Any other comments

- 20% of coursework graded on something we arent being taught?

- Thanks Uraz

- Get rid of bonus points. Why is identifying errors on a ~100 slide power point worthy of extra credit?

Overall, best module this term :)

- The download for the week 2 slides pdf seems to be broken: https://modules.lancaster.ac.uk/pluginfile.php/3945298/mod_folder/content/0/Week2.pdf?forcedownload=1 when opened gives the error "Failed to load PDF document."

Relational Database: Definitions

https://quake.fandom.com/wiki/Weapon_(Q1)

Weapon	Damage	Ammo Type	Maximum Ammo	Fire Mode	Rate of Fire (rpm)	Muzzle Velocity (m/s)	Maximum Range (m)
Axe	20	None	∞	Single	120	-	-
Shotgun	24	Shells	100	Pump-action	120	380	78
Super Shotgun	56	Shells	100	Pump-action	85	380	78
Nailgun	9	Nails	200	Automatic	600	110	229
Super Nailgun	18	Nails	200	Automatic	600	158	229
Grenade Launcher	120	Rockets	100	Semi-auto	100	22.8	-

- *Relation:* made up of 2 parts:

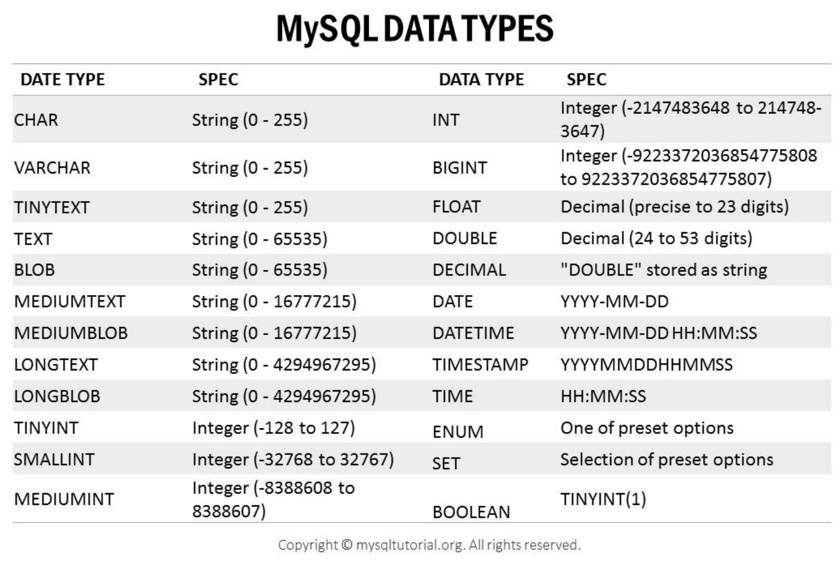
  Instance: a *table* with rows and columns.

  #Rows = *cardinality*, #fields/attributes = *degree / arity.*

  Relation Schema (RS): specifies the relation's name and **type (domain)** of each **column**.

Write the Relation Schema for the given Relation Instance.

Weapons(*Weapon*: TEXT, *Damage*: INT, *AmmoType*: TEXT,
MaximumAmmo: INT, *FireMode*: TEXT,
RateofFire: INT, *MuzzleVelocity*:DOUBLE, *MaximumRange*: INT).



MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Copyright © mysqltutorial.org. All rights reserved.

Relational Database: Definitions

- *Relation:* made up of 2 parts:

  Instance: a *table* with rows and columns.

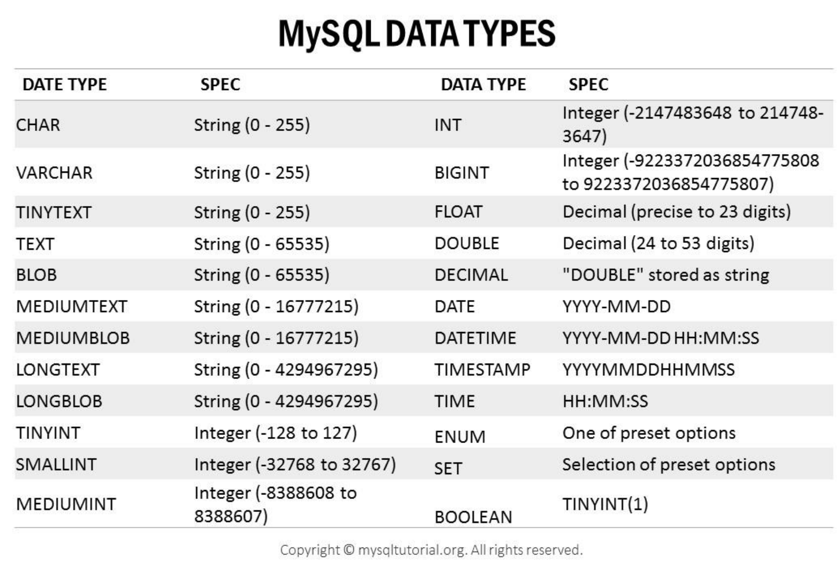
  #Rows = *cardinality*, #fields/attributes = *degree / arity.*

  Relation Schema (RS): specifies the relation's name and **type (domain)** of each **column**.

Team_ID	Wins	Loses	Name	Leader
223	1	43	Les Miserables	Uraz
14	16	0	Thanos	Thanos
1	23	3	Avengers	Iron Man

<u>Write the Relation Schema for the given Relation Instance.</u>

Teams(*Team_ID*: INT, *Wins*: INT, *Loses*: INT, *Name*: TEXT, *Leader*: TEXT).



MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Exercise…

Relational Database: Definitions

Team_ID	Wins	Loses	Name	Leader
223	1	43	Les Miserables	Uraz
14	16	0	Thanos	Thanos
1	23	3	Avengers	Iron Man

- *Relation:* made up of 2 parts:

  Instance: a *table* with rows and columns.

  #Rows = *cardinality*, #fields/attributes = *degree / arity.*

  Relation Schema (RS): specifies the relation's name and **type (domain)** of each **column**.

Write the Relation Schema for the given Relation Instance.

Teams(*Team_ID*: INT, *Wins*: INT, *Loses*: INT, *Name*: TEXT, *Leader*: TEXT).

What would happen if I made a mistake and entered the following row of data?

<Uraz, 34, 55, A poor lecturer, Uraz>

Relational model (Syntax vs Semantics)

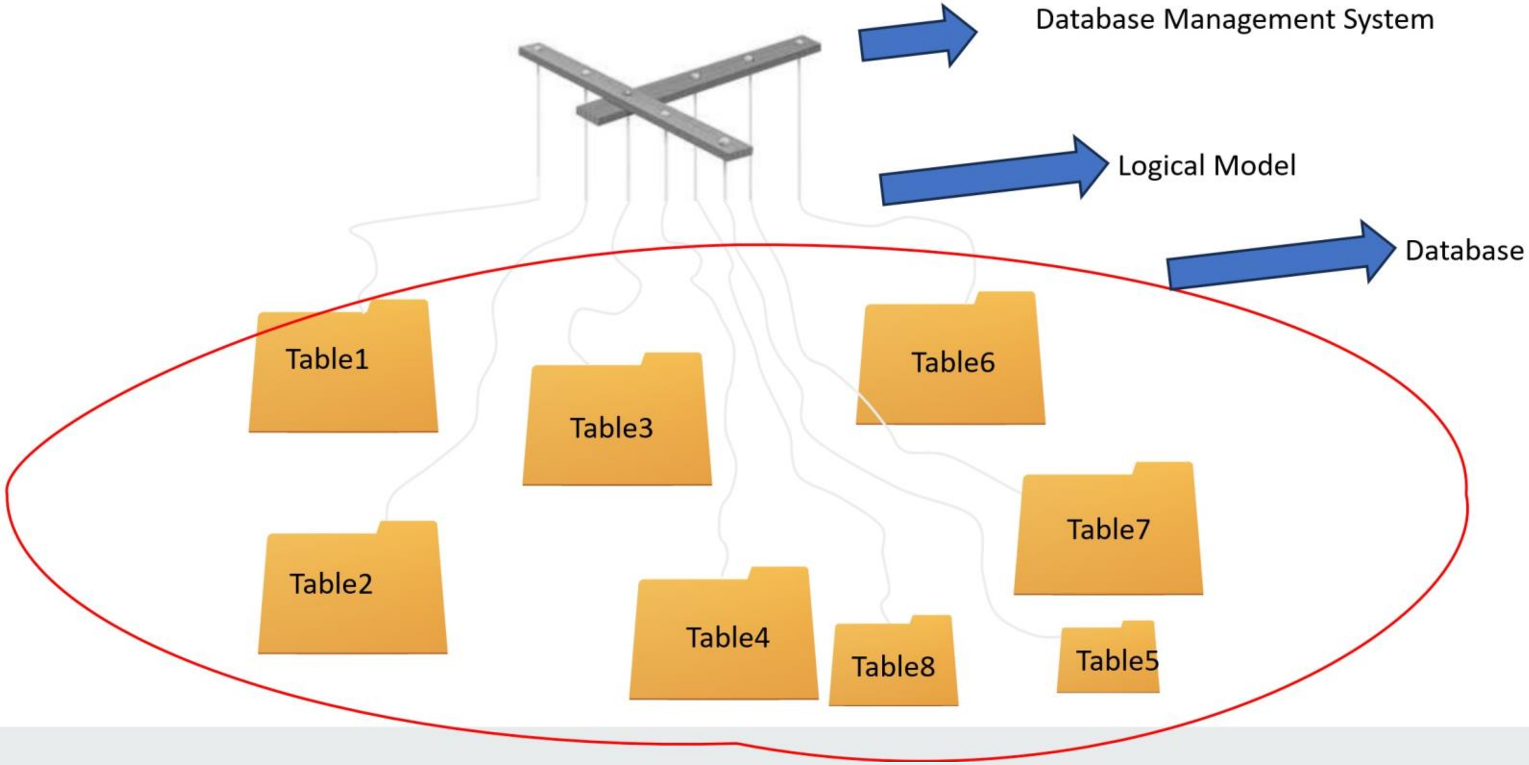
Model	Weight	ChassisN	Max_Speed
	1400	12h37	200
Toyota_Corolla	1300	84t34	200
Hyundai E.GLS	1400	43j5h2	210

What else can go wrong?

- What would happen if I erase a key value?
- Data kept by a DBMS **must obey some rule**s!
- To prevent these, relational databases are built upon logical rules (constraints) set by
  - Real-world rules (Context), designers' choices, functions, etc., using DDL.
- These rules establish what we call the **integrity** of the data
- Integrity of the data is protected by DBMS if INTEGRITY CONSTRAINTS ARE GIVEN.

Relational Database: Definitions

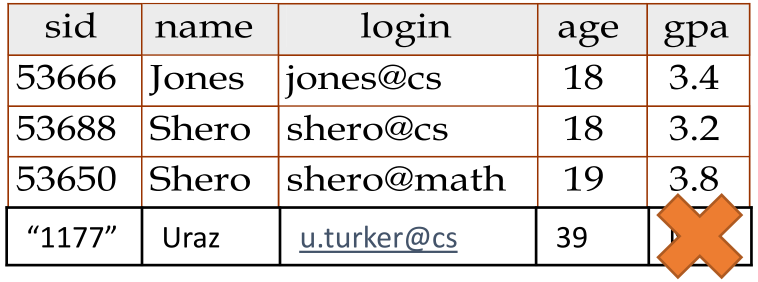
Integrity Constraints are a part of the Logical Model and are provided to the DBMS while tables are created.



Definitions : Integrity Constraints (ICs)

- Integrity constraint (IC): a condition that must be true for *any* database instance.
- A *legal* instance of a relation satisfies all specified ICs.
  - DBMS should not allow illegal instances.
- If the DBMS checks ICs, stored data is more faithful to real-world meaning.
  - Avoids data entry errors, too!
- 1) Domain constraint: In any database instance, a value of the attribute must be an element of the attribute's domain (*gpa* is a DOUBLE-valued attribute, so we can only store DOUBLE values in related cells) as set in RS.

Students(*sid*: INT, *name*: TEXT, *login*: TEXT, *age*: INT, *gpa*: DOUBLE).



sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Shero	shero@cs	18	3.2
53650	Shero	shero@math	19	3.8
“1177”	Uraz	u.turker@cs	39	✗

Domain constraint breached
(gpa is a DOUBLE value, Na is a TEXT)

Definitions : Integrity Constraints (ICs)

Map	Vehicle	Occupants	Type
	Buggy	2	Land
Miramar	PG-117	5	Water
Vikendi	C-130	100	AIR

2) Entity Integrity constraint: A key value cannot be duplicated or left empty in any instance.

- Once a DB Admin sets a key, the *DBMS must inspect every modification* on the data w.r.t the key value.
- DB Admin's task (using DDL) is to set the primary key for the data.
    - DBMS has built-in functions to protect key constraints; to activate those functions, DBA must identify them using DDL (week 4!).
- Write the Relational Schema for this table

PubG(*Map*: TEXT, *Vehicle*: Text, *Occupants*: INT, *Type*: Text, *PRIMARY KEY*: Map).

Relational Database: Definitions

- *Relation:* made up of 2 parts:

  Instance: a *table* with rows and columns.

  #Rows = *cardinality*, #fields/attributes = *degree / arity.*

Weight	cost	<u>owner</u>	explosive	Type
1kg	441	X82jxm	1	1
2kg	123	Fr29x9a	0	3
1kg	223	7ndj2qs	0	12

Relation Schema (RS): specifies the relation’s name and **type (domain)** of each **column**.

<u>Write the Relation Schema for the given Relation Instance.</u>

Ammunition(*Weight*: TEXT, *cost*: INT, *owner*: TEXT, *explosive*: BOOLEAN, *Type*: INT, *PRIMARY KEY*: owner).

Relational Database: Definitions

Team_ID	Wins	Loses	Name	Leader
223	1	43	Les Miserables	Uraz
14	16	0	Thanos	Thanos
1	23	3	Avengers	Iron Man

- *Relation:* made up of 2 parts:

  Instance: a *table* with rows and columns.

  #Rows = *cardinality*, #fields/attributes = *degree / arity.*

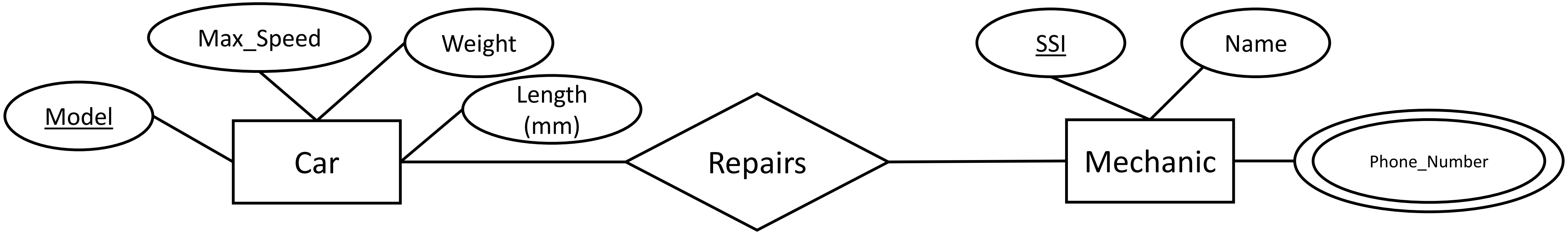
  Relation Schema (RS): specifies the relation's name and **type (domain)** of each **column**.

Write the Relation Schema for the given Relation Instance.

Teams(*Team_ID*: INT, *Wins*: INT, *Loses*: INT, *Name*: TEXT, *Leader*: TEXT, *PRIMARY KEY:* Team_ID).

Referential integrity: How about relationships?

- How do we set the integrity of relationships?



Relationship sets

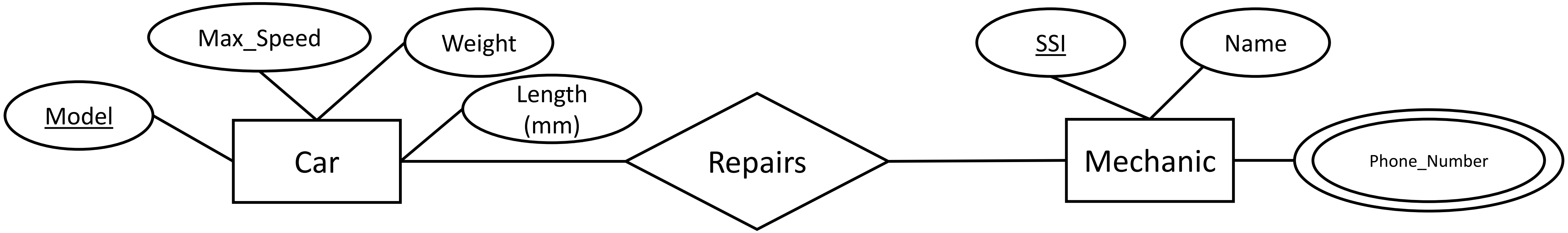
- We inform DBMS regarding the **referential integrity** for relationship sets while creating them using Foreign Keys (in DDL).

Referential integrity: Another key! Foreign keys

- **Foreign keys: a peripheral attribute that establishes referential integrity between entity sets.**

Model	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

SSI	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

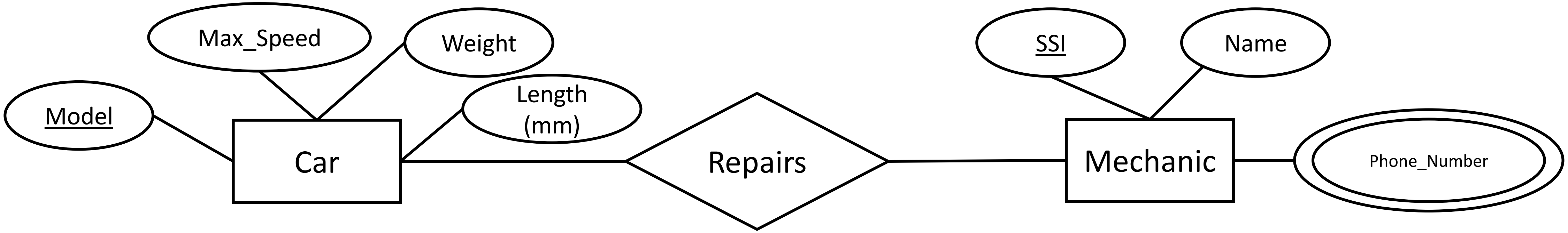


Referential integrity: Another key! Foreign keys

- **Foreign keys: a peripheral attribute that establishes referential integrity between entity sets.**
- It requires either i) **importing the Primary Key attribute of one table to the other table**

Model	**Weight**	**Length (mm)**	**Max_Speed**
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

SSI	**Name**	**Phone_Number**
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

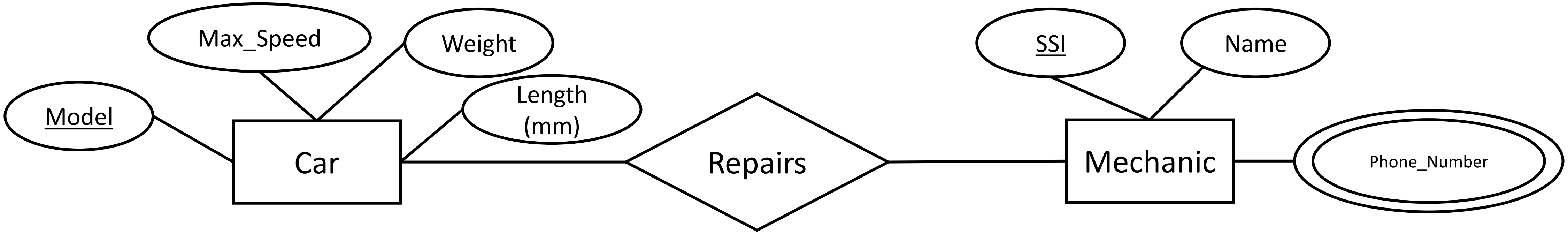


Referential integrity: Another key! Foreign keys

- **Foreign keys: a peripheral attribute that establishes referential integrity between entity sets.**
- It requires either i) **importing the Primary Key attribute of one table to the other table**

Model	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

SSI	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

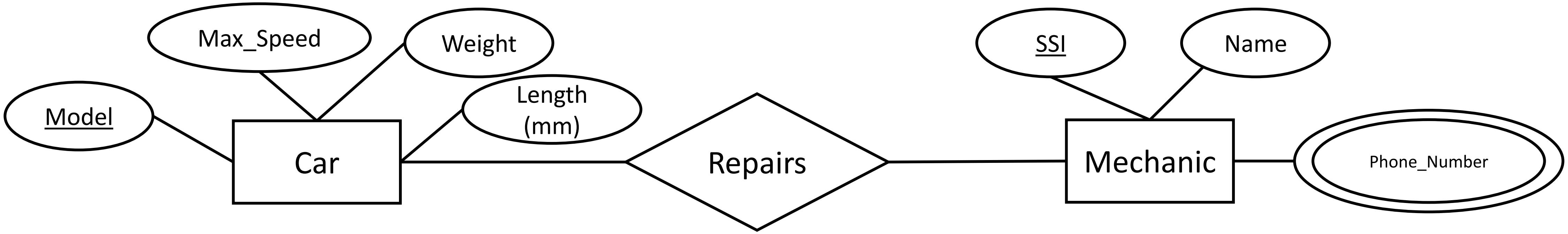


Referential integrity: Another key! Foreign keys

- **Foreign keys: a peripheral attribute that establishes referential integrity between entity sets.**
- It requires either i) **importing <u>the Primary Key</u> attribute of one table to the other table**

<u>Model</u>	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

Model	<u>SSI</u>	Name	Phone_Number
BMW 3.21	87542702	Tom	75315567, 75315264
Toyota_Corolla	68201937	Uraz	75335521, 75334567
Hyundai E.GLS	23139827	Nick	75315544, 75315237

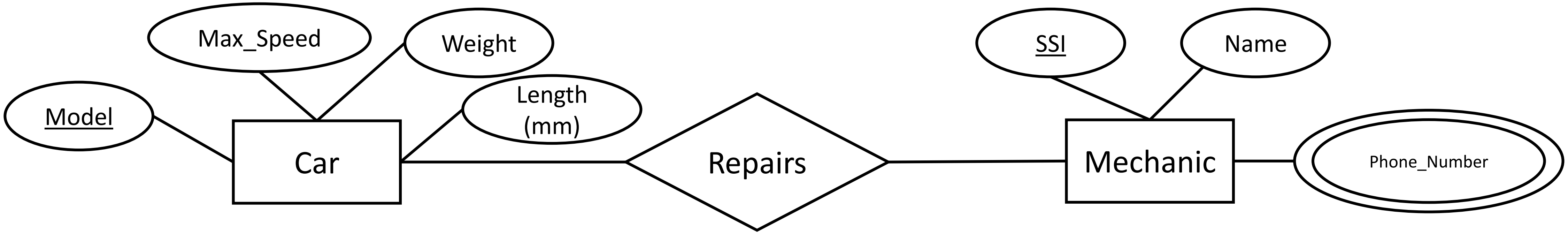


Referential integrity: Another key! Foreign keys

- **Values may be in different order!**

<u>Model</u>	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

Model	<u>SSI</u>	Name	Phone_Number
Toyota_Corolla	87542702	Tom	75315567, 75315264
Hyundai E.GLS	68201937	Uraz	75335521, 75334567
BMW 3.21	23139827	Nick	75315544, 75315237



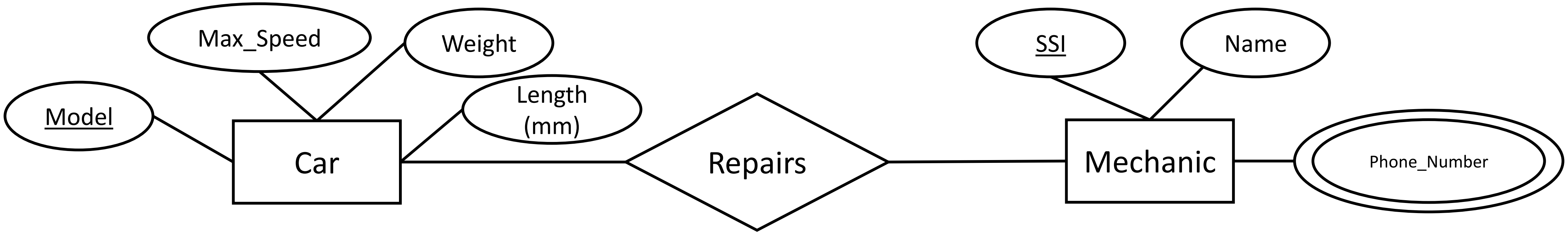
Referential integrity: Another key! Foreign keys

- **Foreign keys: a peripheral attribute that establishes referential integrity between entity sets.**
- It requires either i) **importing the Primary Key attribute of one table to the other table** or ii) **creating a new table that holds the primary keys of the tables in relation**.

Model	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

Model	SIS
Toyota_Corolla	87542..
Hyundai E.GLS	68201..
BMW 3.21	2313..

SSI	Name	Phone_Number
87542702	Tom	75315567, 75315264
68201937	Uraz	75335521, 75334567
23139827	Nick	75315544, 75315237

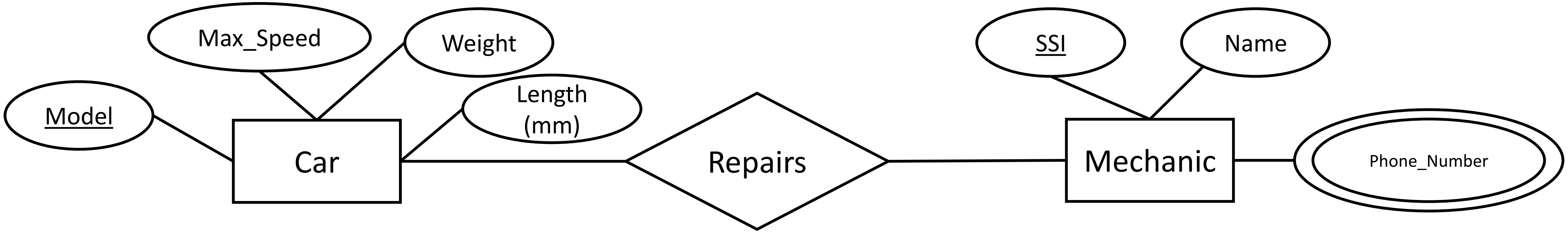


Referential integrity: Foreign keys

- ***Foreign key***: `logical **pointer**’.

Model	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

Model	SSI	Name	Phone_Number
Toyota_Corolla	87542702	Tom	75315567, 75315264
Hyundai E.GLS	68201937	Uraz	75335521, 75334567
BMW 3.21	23139827	Nick	75315544, 75315237



Properties of foreign keys

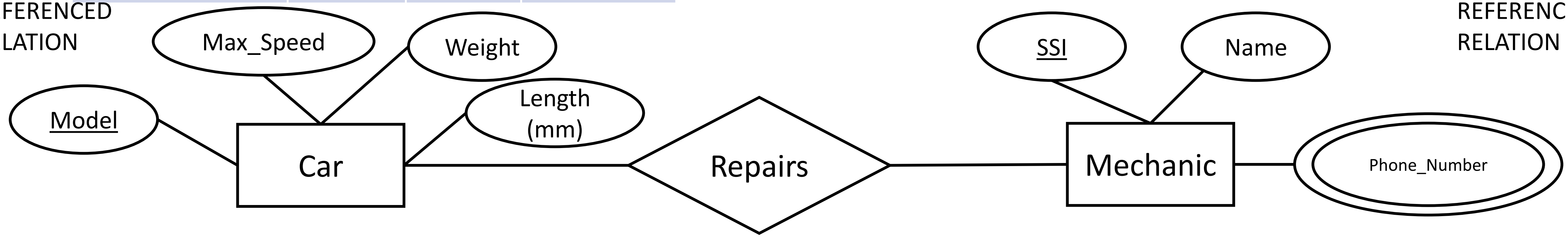
- Foreign key must:
  - Have the same name and domain/type as the referencing relation.
  - Related entities **must** have the same values.

Model	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

Model	SSI	Name	Phone_Number
Toyota_Corolla	87542702	Tom	75315567, 75315264
Hyundai E.GLS	68201937	Uraz	75335521, 75334567
BMW 3.21	23139827	Nick	75315544, 75315237

REFERENCED RELATION

REFERENCING RELATION



Properties of foreign keys

- Foreign key must:
  - Have the same name and domain/type as the referencing relation.
  - Related entities **<u>must</u>** have the same values.

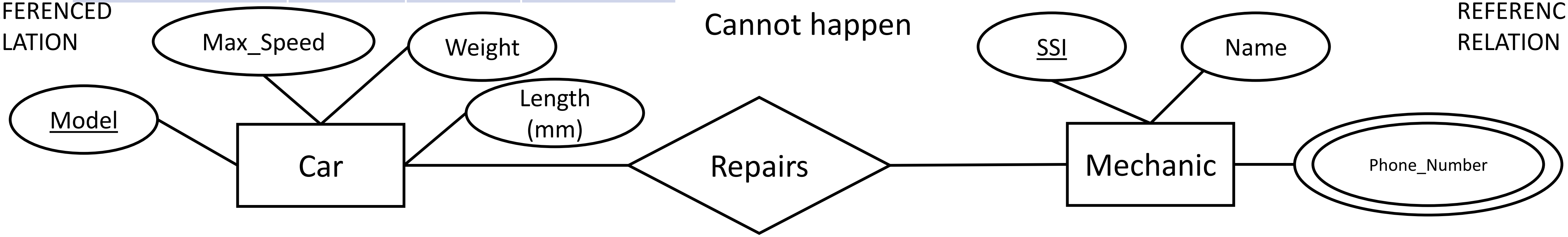
<u>Model</u>	Weight	Length (mm)	Max_Speed
BMW 3.21	1400	2501	200
Toyota_Corolla	1300	3321	200
Hyundai E.GLS	1400	3895	210

REFERENCED
RELATION

Model	SSI	Name	Phone_Number
✗ Toyota	87542702	Tom	75315567, 75315264
Hyundai E.GLS	68201937	Uraz	75335521, 75334567
BMW 3.21	23139827	Nick	75315544, 75315237

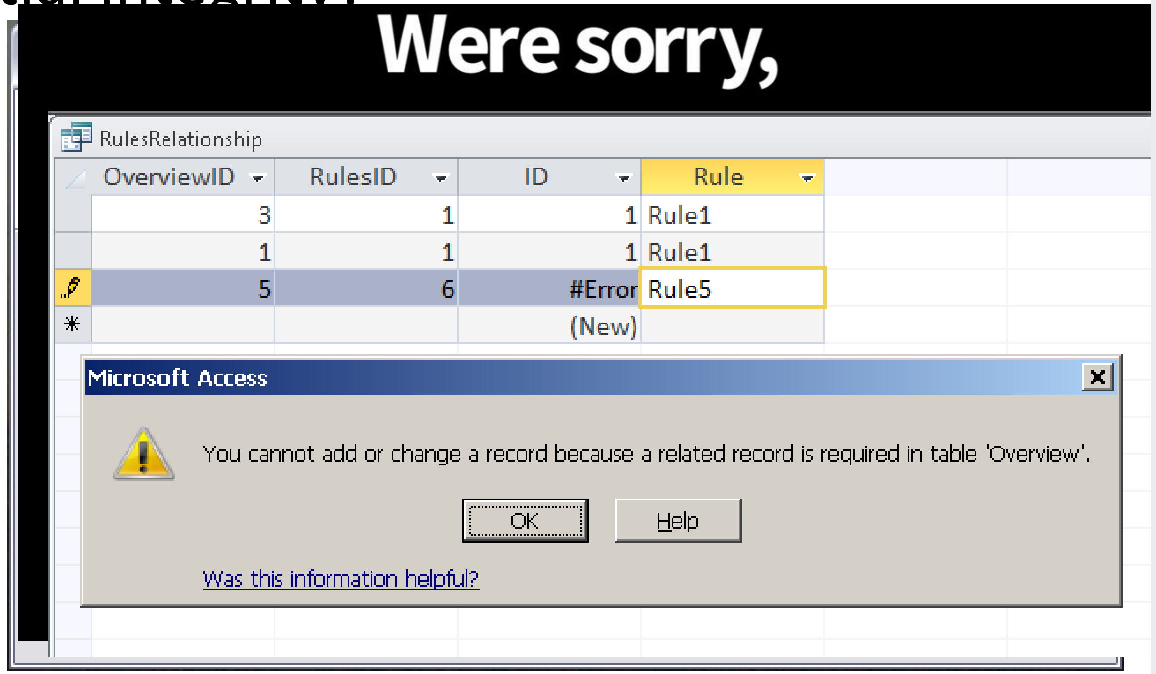
REFERENCING
RELATION

Cannot happen



Referential integrity: Foreign keys

- If all foreign key constraints are enforced, **referential integrity** is achieved, i.e., no dangling references, dissimilar values, etc.
- Can you name a data model w/o referential integrity?
- Links in HTML.
- Pointers in C++.
- Phone numbers recorded in your phone.
- In Database.



Were sorry,

RulesRelationship

	OverviewID	RulesID	ID	Rule
	3	1	1	Rule1
	1	1	1	Rule1
✎	5	6	#Error	Rule5
*			(New)	

Microsoft Access

You cannot add or change a record because a related record is required in table 'Overview'.

Foreign Key in action.

Find the total quantity and items for orders of customers living in "Pennsylvania".

Customers	Customer#	Name	Street	City	Country
	AT01	Alan Turing	Maida Vale	London	UK
	JB01	Jean Bartik	Woodland Walk	Pennsylvania	USA
	MH01	Margaret Hamilton	300 E Street	Pennsylvania	USA
	AL01	Ada Lovelace	Hucknall Road	Nottingham	UK
	EC01	Edgar F. Codd	15 Parks Road	Oxford	UK

Items	Item#	Description	Category
	0001	Hard Disk Drive	Internal Hardware
	0002	16GB RAM	Internal Hardware
	0003	Mechanical Keyboard	Peripherals
	0004	LCD 32" HD Monitor	Display
	0005	2200 RTX GPU 11GB	Internal Hardware

Orders	Order#	Item#	Customer#	Delivery_date	Quantity
	Or0022	0002	MH01	2020-02-10	2
	Or0023	0004	AL01	2020-01-30	1
	Or0024	0001	AT01	2020-02-05	1
	Or0025	0005	JB01	2020-02-06	1
	Or0026	0003	EC01	2020-02-01	3
	Or0027	0004	JB01	2020-02-03	6

Process "Customers": Find rows with "Pennsylvania" and Get the Key Values.

Process "Orders": Find Item# and Quantity using keys.

Process "Items": Find Descriptions using Item#.

->9, LCD32" Monitor, 2200 RTX GPU 11GB, 16GBram.

Can you draw the ER diagram for these relations? (May be next week!)

Strategies to enforce Referential Integrity

- Consider Students and Enrolled; *sid* in Enrolled is a foreign key that references Students.
- What should be done if an Enrolled tuple with a non-existent student id is inserted?
- *Reject it!*

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B
1177	SCC201	A

Strategies to enforce Referential Integrity

- What should be done if a Students tuple (say 53650) is deleted?

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53650	Shero	shero@eecs	18	3.2
53689	Smith	smith@math	19	3.8

- Also delete all Enrolled tuples that refer to it.
- Disallow deletion of a Students tuple that is referred to.
- Set sid in Enrolled tuples that refer to it to a *default sid*.
- (In SQL, also: Set sid in Enrolled tuples that refer to it to a special value *null*, denoting *`unknown'* or *`inapplicable'*.)

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

- What happens if the primary key of the Students tuple is updated (53650 to 00001)?

Referential Integrity in SQL/92

- SQL/92 supports all 4 options on deletes and updates.
  - Default is NO ACTION (*delete/update is rejected*)
  - CASCADE (also delete all tuples that refer to deleted tuple)
  - SET NULL / SET DEFAULT (sets foreign key value of referencing tuple)

Lets speculate about ICs. for the following tables.

Customers	Customer#	Name	Street	City	Country
	AT01	Alan Turing	Maida Vale	London	UK
	JB01	Jean Bartik	Woodland Walk	Pennsylvania	USA
	MH01	Margaret Hamilton	300 E Street	Pennsylvania	USA
	AL01	Ada Lovelace	Hucknall Road	Nottingham	UK
	EC01	Edgar F. Codd	15 Parks Road	Oxford	UK

Assume these tables.

Do tables obey key constraints?

Do tables obey domain constraints?

What are the foreign keys?

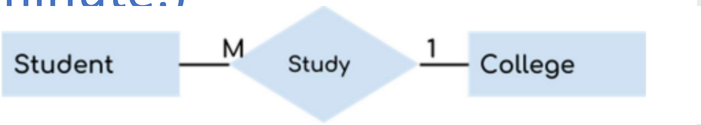
“Print the orders of Customers whose Customer number starts with A?”

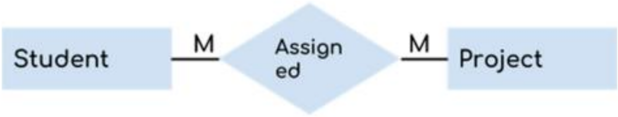
Items	Item#	Description	Category
	0001	Hard Disk Drive	Internal Hardware
	0002	16GB RAM	Internal Hardware
	0003	Mechanical Keyboard	Peripherals
	0004	LCD 32” HD Monitor	Display
	0005	2200 RTX GPU 11GB	Internal Hardware

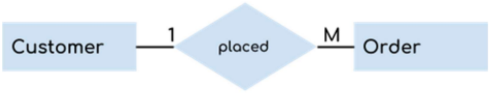
Orders	Order#	Item#	Customer#	Delivery_date	Quantity
	Or0022	0002	MH01	2020-02-10	2
	Or0023	0004	AL01	2020-01-30	1
	Or0024	0001	AT01	2020-02-05	1
	Or0025	0005	JB01	2020-02-06	1
	Or0026	0003	EC01	2020-02-01	3
	Or0027	0004	JB01	2020-02-03	6

Integrity Constraints for relational databases.

- Integrity of data: it is the state of data in which data obeys the constraints set by DBA.
- 1) Domain constraints.
  - Values in tuples should obey types of attributes. (You should not provide text to INT field)
- 2) Entity Integrity constraints.
  - Keys of a relation must be unique, non-redundant, and not Null (entity integrity constraint).
- 3) Referential integrity.
  - How are two relations related to each other? (We will see in a minute.)
    - The relation must be made by using keys,
    - The DBMS must preserve this relation.



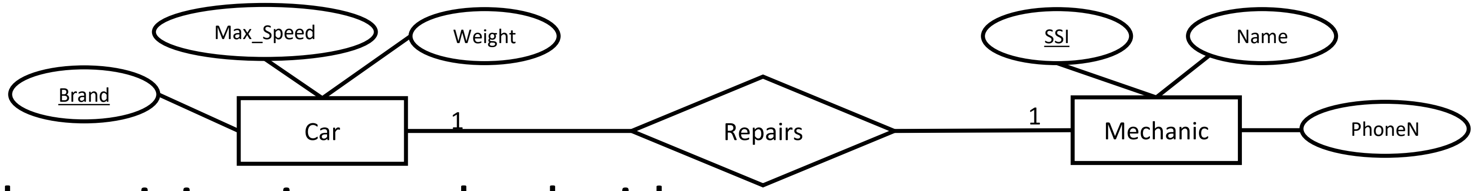




- EXTREMELY IMPORTANT CONTENT A HEAD!!!

How do we derive Foreign Keys and ICs for different relationship types?

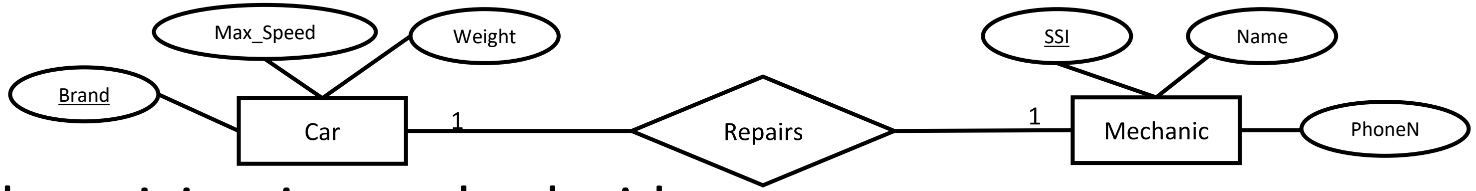
“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



- 1 to 1 relation having partial participation on both sides:

How do we derive Foreign Keys and ICs for different relationship types?

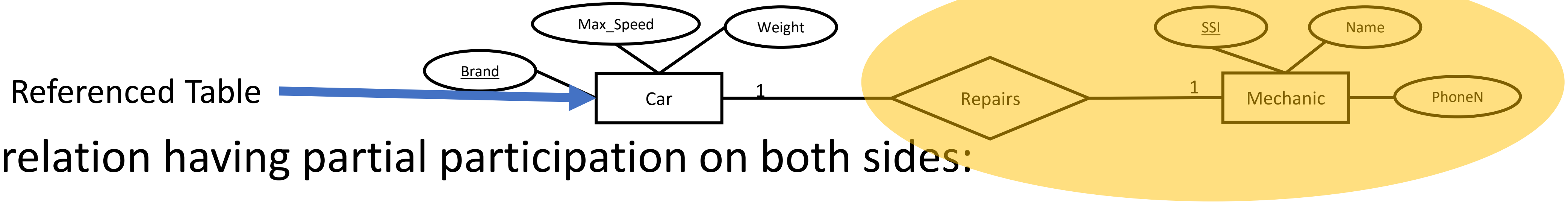
"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.

How do we derive Foreign Keys and ICs for different relationship types?

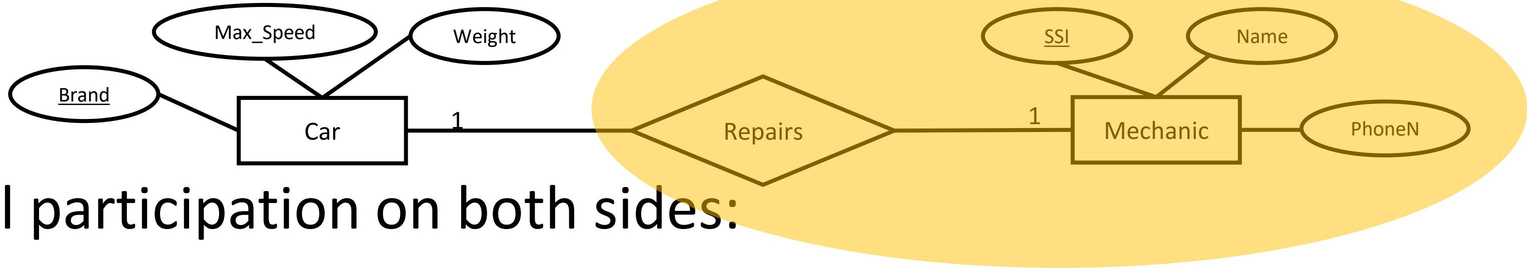
"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.

How do we derive Foreign Keys and ICs for different relationship types?

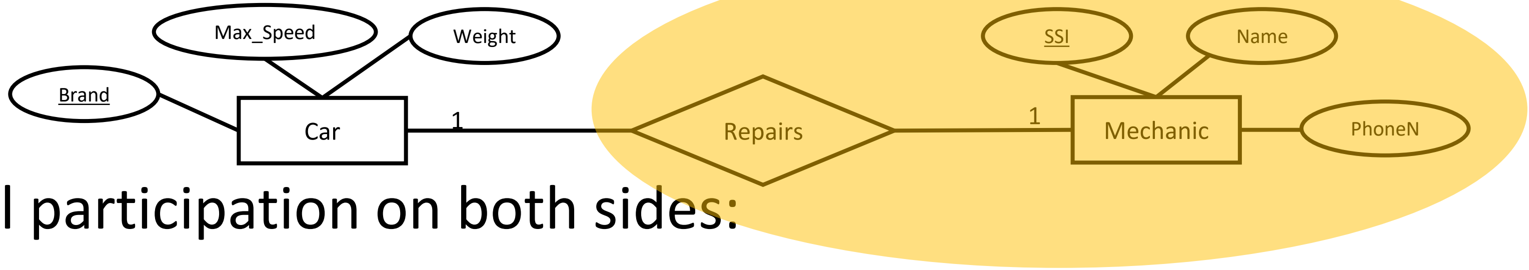
“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.
- Import the primary key of the referenced table to the referencing one.

How do we derive Foreign Keys and ICs for different relationship types?

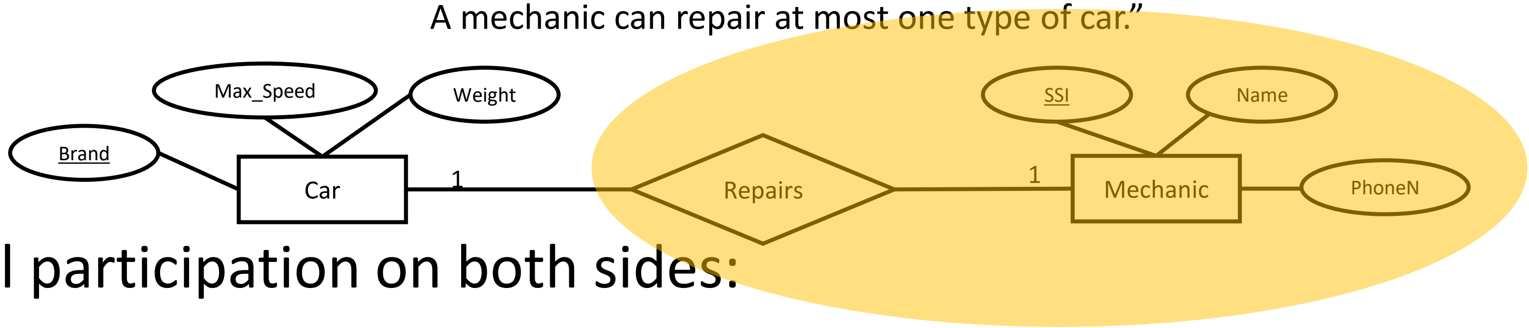
“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.
- Import the primary key of the referenced table to the referencing one.
- This key will be the foreign key and declare it its foreign key:

How do we derive Foreign Keys and ICs for different relationship types?

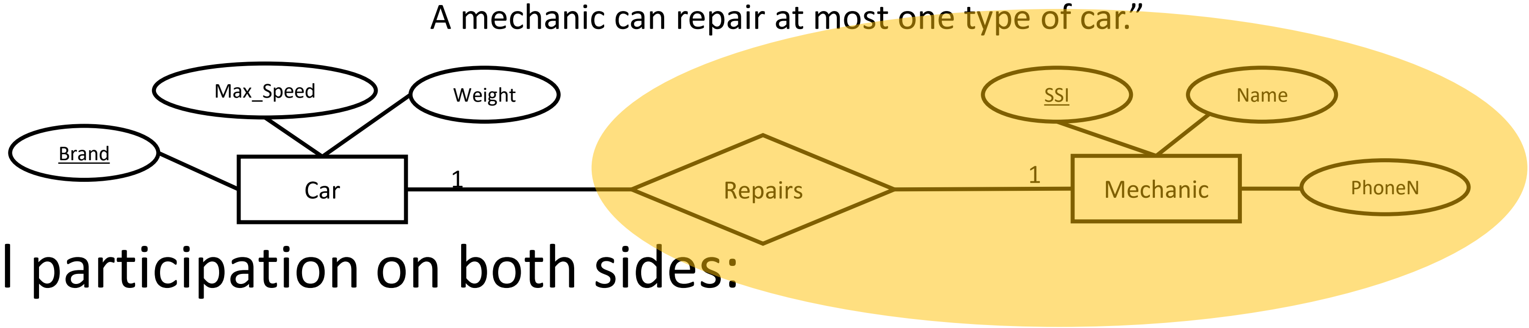
"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.
- Import the primary key of the referenced table to the referencing one.
- This key will be the foreign key and declare it its foreign key:
    - Car(*Brand*:TEXT,*Weight*:INT,*Length*:DOUBLE,*Max_Speed*:INT, *PRIMARY KEY:*BRAND)
    - Mec_Rep(SSI:*TEXT*,Name:*TEXT*,Phone:*TEXT*,Brand:*TEXT, PRIMARY KEY:*SSI, *Foreign Key: Brand REFERENCING:*CAR)

How do we derive Foreign Keys and ICs for different relationship types?

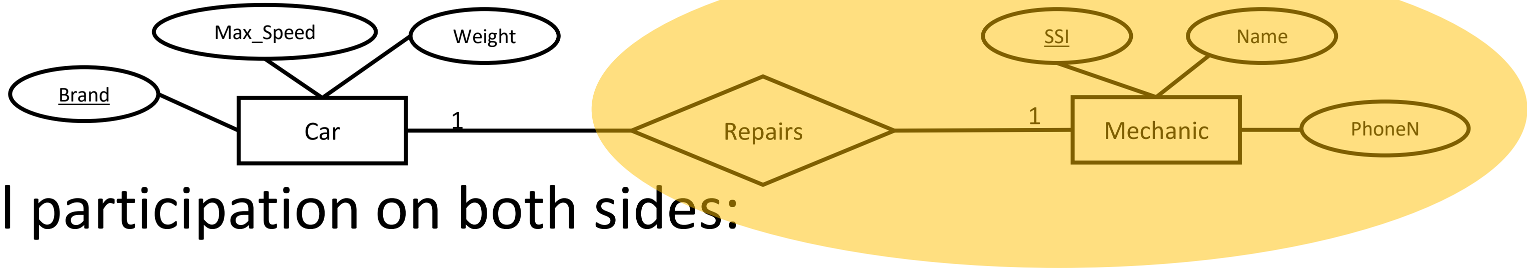
"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.
- Import the primary key of the referenced table to the referencing one.
- This key will be the foreign key and declare it its foreign key:
    - Car(*Brand*:TEXT,*Weight*:INT,*Length*:DOUBLE,*Max_Speed*:INT, *PRIMARY KEY:*BRAND)
    - Mec_Rep(SSI:*TEXT*,Name:*TEXT*,Phone:*TEXT*,Brand:*TEXT, PRIMARY KEY:*SSI, *Foreign Key: Brand REFERENCING:*CAR)

How do we derive Foreign Keys and ICs for different relationship types?

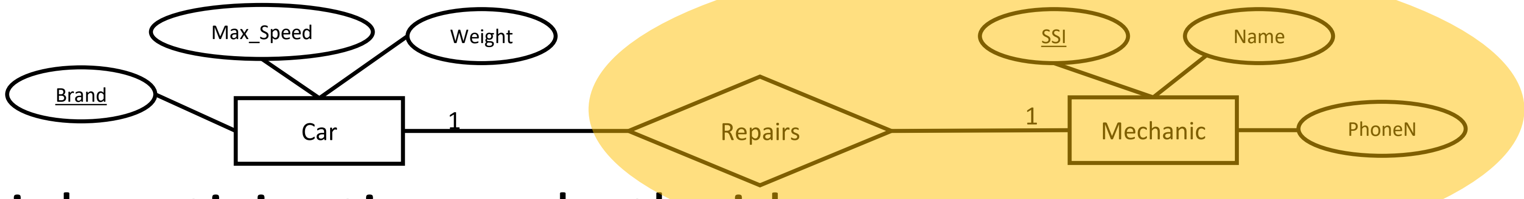
“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



- 1 to 1 relation having partial participation on both sides:
- Select one table (randomly) as the referenced table and the other as the referencing table.
- Import the primary key of the referenced table to the referencing one.
- This key will be the foreign key and declare it its foreign key:
  - Car(*Brand*:TEXT,*Weight*:INT,*Length*:DOUBLE,*Max_Speed*:INT, *PRIMARY KEY:*BRAND)
  - Mec_Rep(SSI:*TEXT*,Name:*TEXT*,Phone:*TEXT*,Brand:*TEXT*, *PRIMARY KEY:*SSI, *Foreign Key: Brand REFERENCING:*CAR)
- Do you think that this is enough?
  -

How do we derive Foreign Keys and ICs for different relationship types?

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



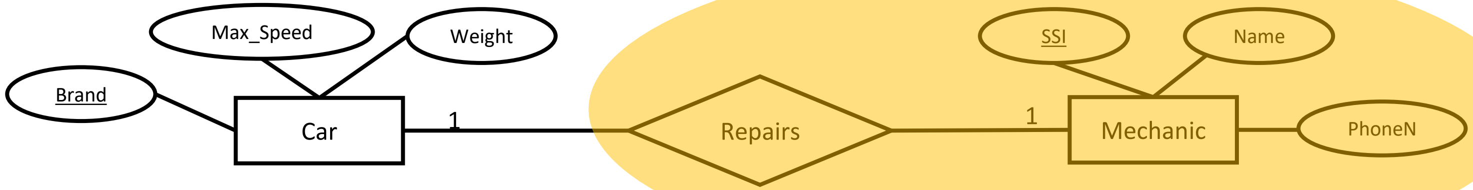
- 1 to 1 relation having partial participation on both sides:
  - Car(*Brand*:TEXT,*Weight*:INT,*Length*:DOUBLE,*Max_Speed*:INT, *PRIMARY KEY:*BRAND)
  - Mec_Rep(SSI:*TEXT*,Name:*TEXT*,Phone:*TEXT*,Brand:*TEXT*, *PRIMARY KEY:*SSI, *Foreign Key: Brand REFERENCING:*CAR)

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

How do we derive Foreign Keys and ICs for different relationship types?

“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



- 1 to 1 relation having partial participation on both sides:
  - Car(*Brand*:TEXT,*Weight*:INT,*Length*:DOUBLE,*Max_Speed*:INT, *PRIMARY KEY:*BRAND)
  - Mec_Rep(SSI:*TEXT*,Name:*TEXT*,Phone:*TEXT*,Brand:*TEXT*, *PRIMARY KEY:*SSI, *Foreign Key: Brand REFERENCING:*CAR)

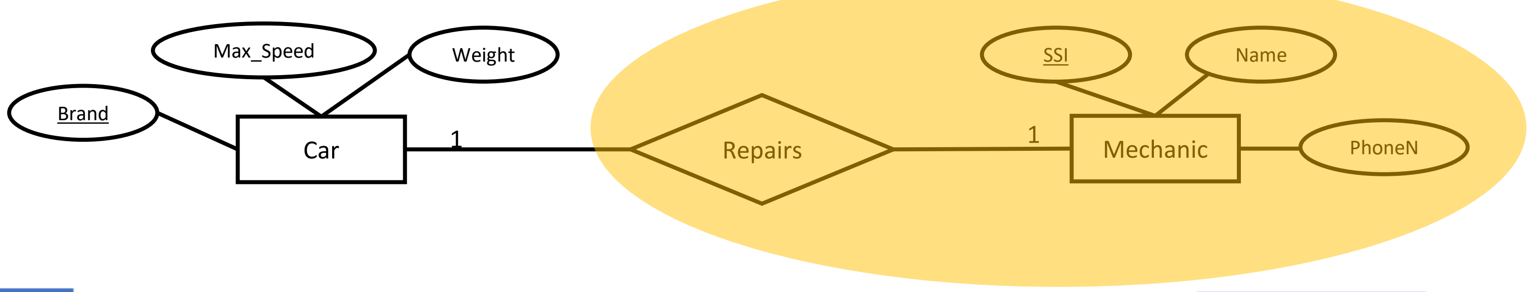
Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21
43279823	~~[illegible]~~	~~73352362~~	BMW 3.21

How do we derive Foreign Keys and ICs for different relationship types?

“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”

- 1 to 1 relations



Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

Car(Brand:TEXT,Weight:INT,Length:DOUBLE, Max_Speed:INT, PRIMARY KEY:BRAND)

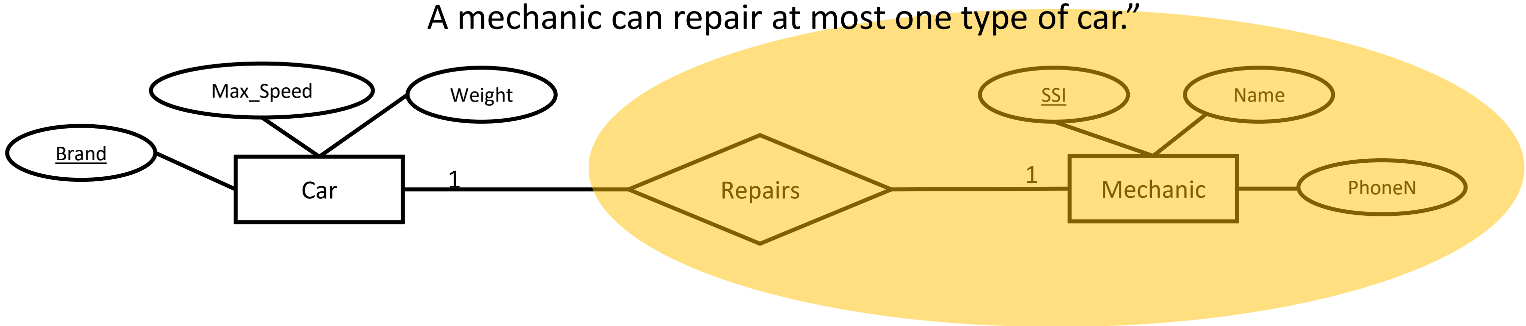
Mec_Rep(SSI:TEXT,Name:TEXT,Phone:TEXT,Brand:TEXT, PRIMARY KEY:SSI, Foreign Key: Brand REFERENCING:CAR
)

The repairs relation is one-to-one. Therefore, for every SSI, there must exist one Brand. Moreover, as Brand cannot repeat, we use the **UNIQUE** keyword.

How do we derive Foreign Keys and ICs for different relationship types?

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."

- 1 to 1 relations



Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

Car(Brand:TEXT,Weight:INT,Length:DOUBLE, Max_Speed:INT, PRIMARY KEY:BRAND)

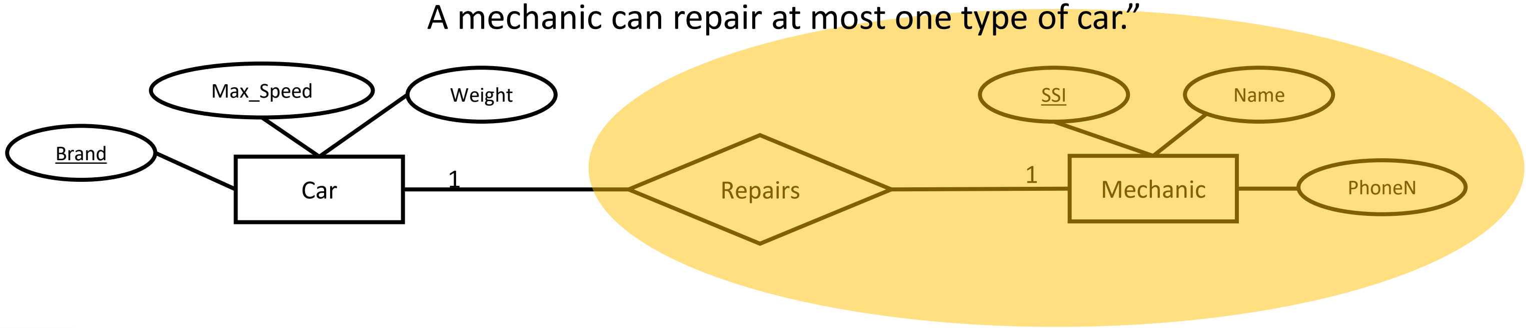
Mec_Rep(SSI:TEXT,Name:TEXT,Phone:TEXT,Brand:TEXT, PRIMARY KEY:SSI, Foreign Key: Brand REFERENCING:CAR, Brand is UNIQUE)

The repairs relation is one-to-one. Therefore, for every SSI, there must exist one Brand. Moreover, as Brand cannot repeat, we use the **UNIQUE** keyword.

How do we derive Foreign Keys and ICs for different relationship types?

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."

- 1 to 1 relations



Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

Car(Brand:TEXT,Weight:INT,Length:DOUBLE, Max_Speed:INT, PRIMARY KEY:BRAND)

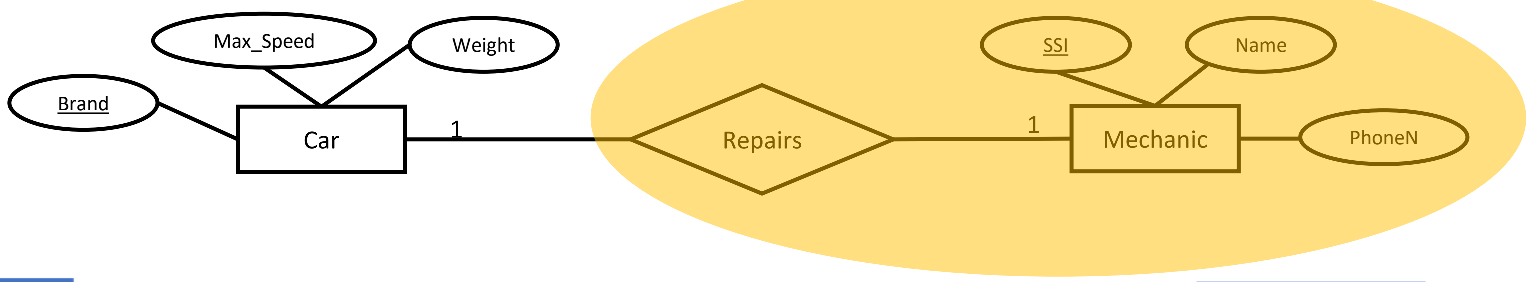
Mec_Rep(SSI:TEXT,Name:TEXT,Phone:TEXT,Brand:TEXT, PRIMARY KEY:SSI, Foreign Key: Brand REFERENCING:CAR, Brand is UNIQUE)

Assume I delete the tuple "BMW 3.21, 1400, 3.21, 200" from the CAR table. What value should DBMS set for the Mechanic that can repair BMW 3.21?

How do we derive Foreign Keys and ICs for different relationship types?

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."

- 1 to 1 relations



Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

Car(Brand:TEXT,Weight:INT,Length:DOUBLE, Max_Speed:INT, PRIMARY KEY:BRAND)

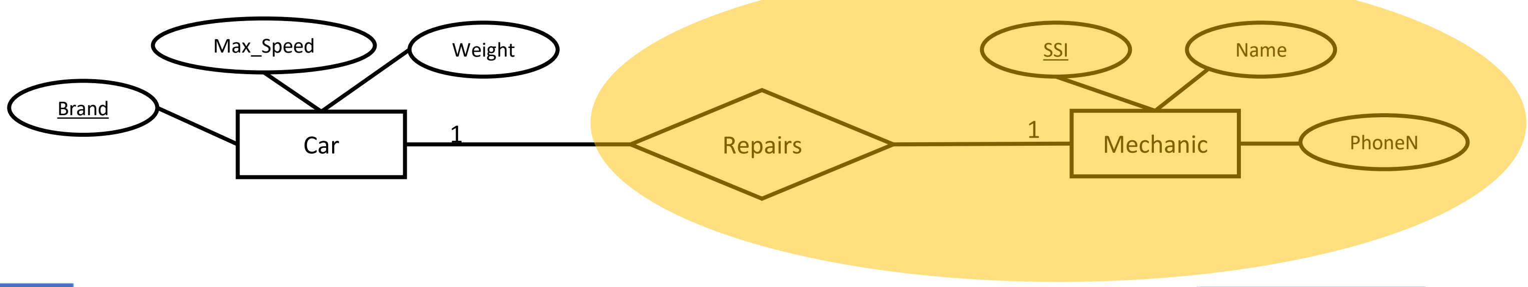
Mec_Rep(SSI:TEXT,Name:TEXT,Phone:TEXT,Brand:TEXT, PRIMARY KEY:SSI, Foreign Key: Brand REFERENCING:CAR, Brand is UNIQUE)

Assume I delete the tuple "BMW 3.21, 1400, 3.21, 200" from the CAR table. What value should DBMS set for the Mechanic that can repair BMW 3.21?
Since the Repairs Relation **partially participates** in both ends, I can select **SET NULL** or **SET DEFAULT.**

How do we derive Foreign Keys and ICs for different relationship types?

“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”

- 1 to 1 relations



Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

SSI	Name	Phone_Number	Brand
87542702	Tom	75315567	Toyota_Corolla
68201937	Uraz	75335521	Hyundai E.GLS
23139827	Nick	75315544	BMW 3.21

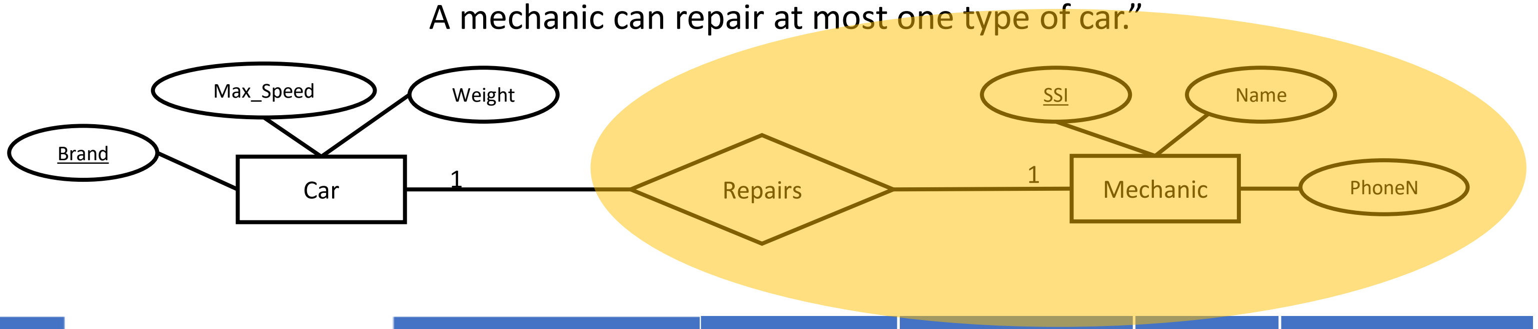
Car(Brand:TEXT,Weight:INT,Length:DOUBLE, Max_Speed:INT, PRIMARY KEY:BRAND)

Mec_Rep(SSI:TEXT,Name:TEXT,Phone:TEXT,Brand:TEXT, PRIMARY KEY:SSI, Foreign Key: Brand REFERENCING:CAR, Brand is UNIQUE, on Delete SET NULL/DEFAULT)

Assume I delete the tuple “BMW 3.21, 1400, 3.21, 200” from the CAR table. What value should DBMS set for the Mechanic that can repair BMW 3.21?
Since the Repairs Relation **partially participates** in both ends, I can select **SET NULL** or **SET DEFAULT.**

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

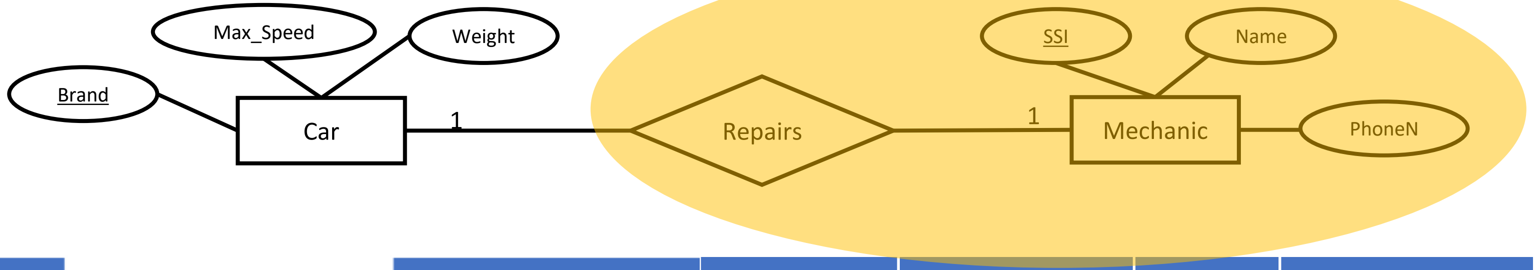
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
Toyota_Corolla	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)

Referential Integrity

“A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car.”



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

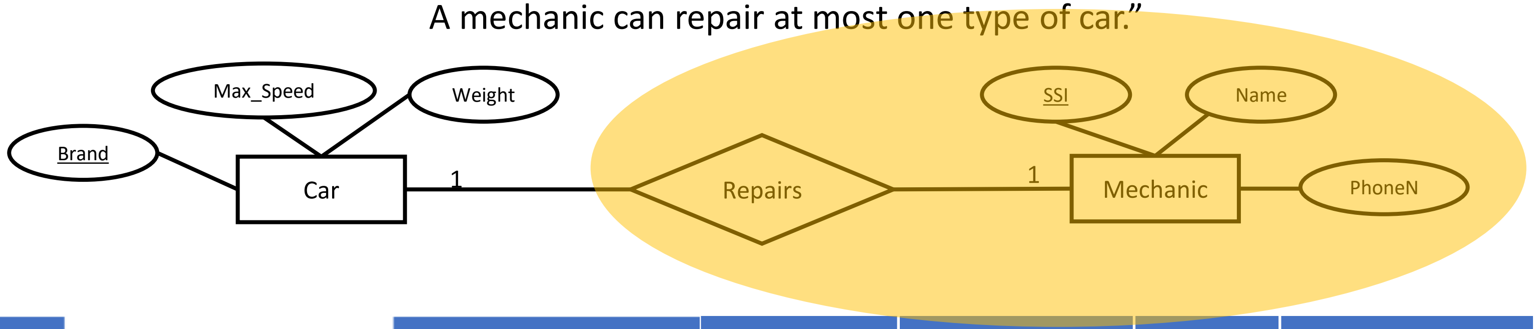
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
Toyota_Corolla	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

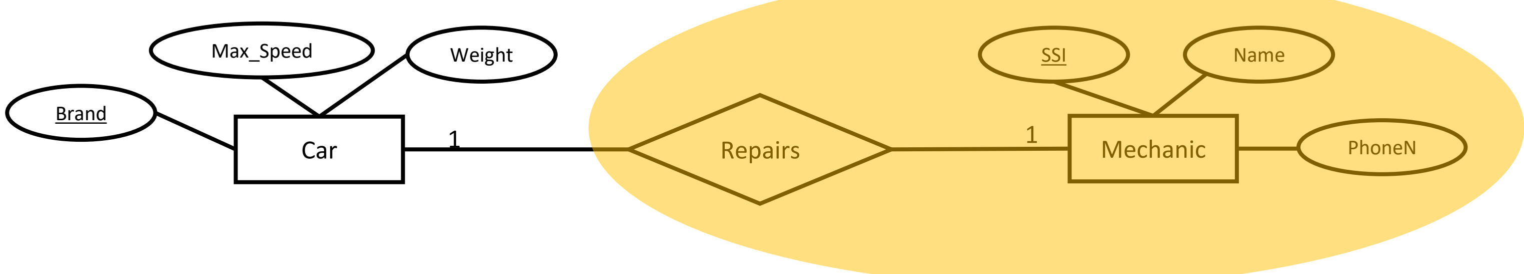
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
Toyota_Corolla	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

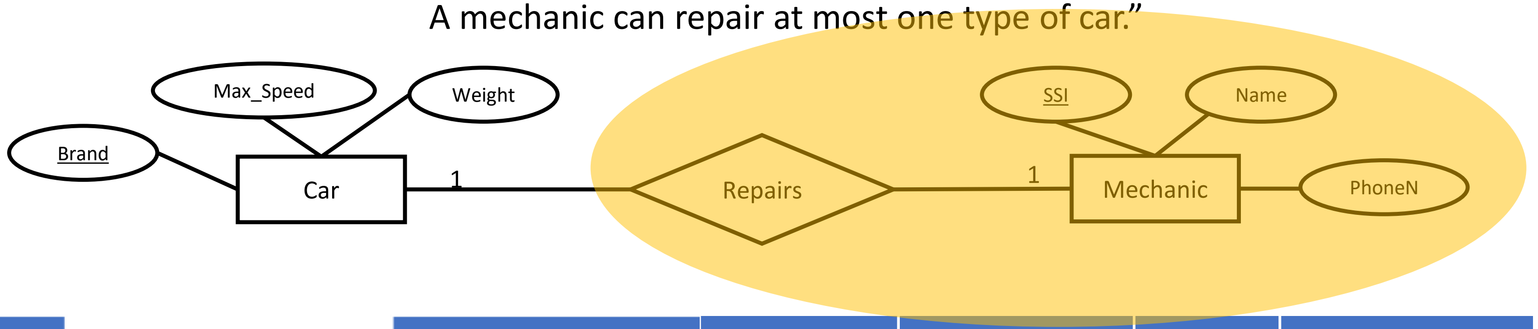
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
DEFAULT	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)
  - If SET DEFAULT -> Select all such tuples in the referencing table (MEC_REPAIR) and set the foreign key value to a default value (you have to specify this) of these tuples in the referencing table (MEC_REPAIR).

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Hyundai E.GLS	1400	3.16	210

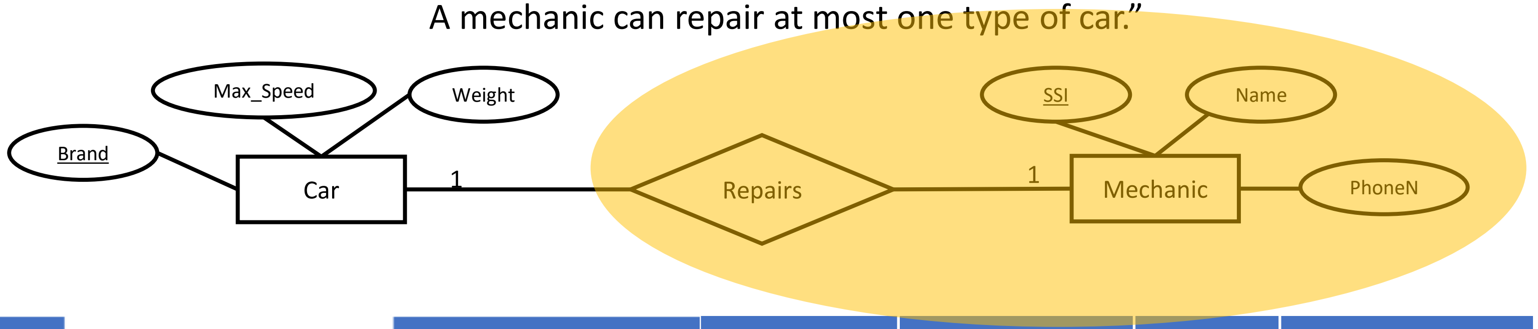
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
DEFAULT	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)
  - If SET DEFAULT -> Select all such tuples in the referencing table (MEC_REPAIR) and set the foreign key value to a default value (you have to specify this) of these tuples in the referencing table (MEC_REPAIR). And delete tuples in CAR

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

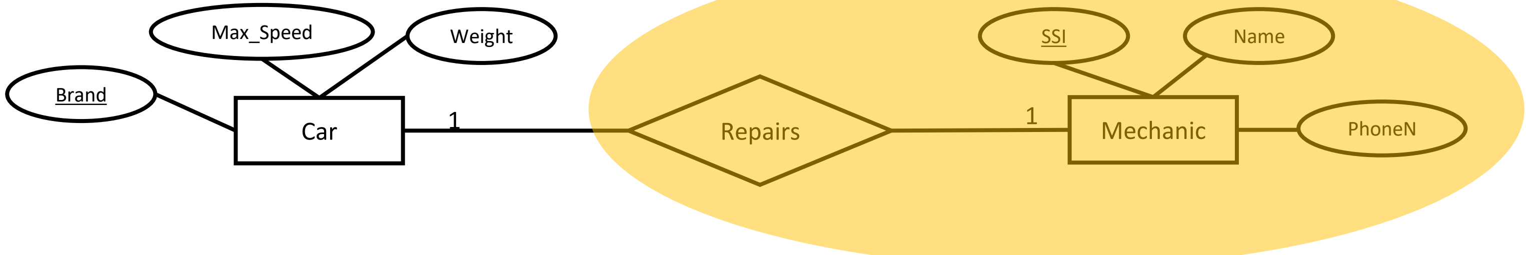
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
Toyota_Corolla	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)
  - If SET NULL -> Select all these tuples in the referencing table (MEC_REPAIR)

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Toyota_Corolla	1300	3.18	200
Hyundai E.GLS	1400	3.16	210

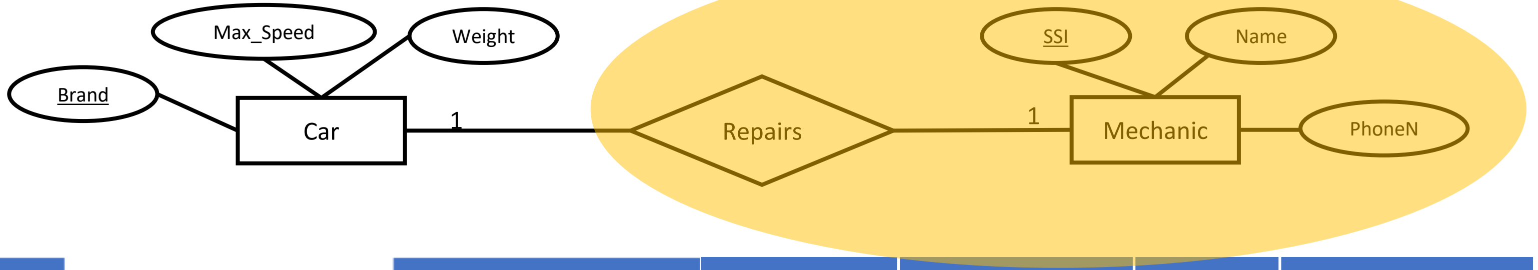
MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
NULL	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)
  - If SET NULL -> Select all these tuples in the referencing table (MEC_REPAIR) and set the foreign key value to a NULL value of these tuples in the referencing table (MEC_REPAIR).

Referential Integrity

"A car can be repaired by at most one mechanic.
A mechanic can repair at most one type of car."



CAR

Brand	Weight	Length	Max_Speed
BMW 3.21	1400	3.21	200
Hyundai E.GLS	1400	3.16	210

MEC_REPAIR

Brand	Price	SSI	Name	Phone_Number
BMW 3.21	10	87542702	Tom	75315567
NULL	23	68201937	Uraz	75335521
Hyundai E.GLS	12	23139827	Nick	75315544

- If a tuple (say 2nd tuple) is to be deleted from referenced table (CAR)
- Get the primary key value of the tuple (Toyota_Corolla).
- Find all the tuples with values (Toyota_Corolla) in the referencing table (MEC_REPAIR)
    - If SET NULL -> Select all these tuples in the referencing table (MEC_REPAIR) and set the foreign key value to a NULL value of these tuples in the referencing table (MEC_REPAIR).
    - And delete the tuples in the referenced table (CAR).