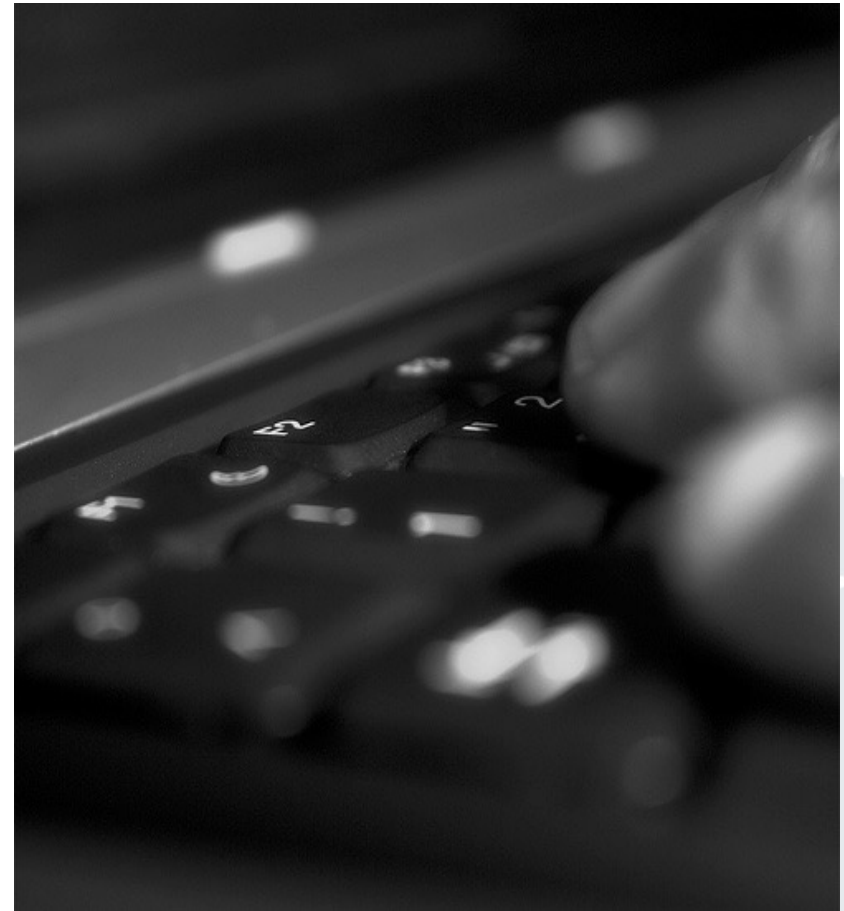


Exercise 6
Business Informatics 2 (PWIN)

Databases & Data-oriented
Modelling

SQL

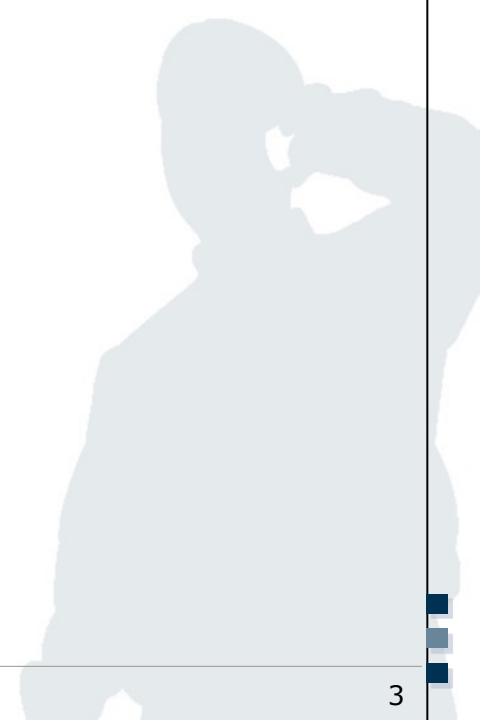
www.m-chair.de



Jenser (Flickr.com)

- Exercise 1: Entity Relationship Model
- Exercise 2: Deriving Relations from an ERM
- Exercise 3: SQL

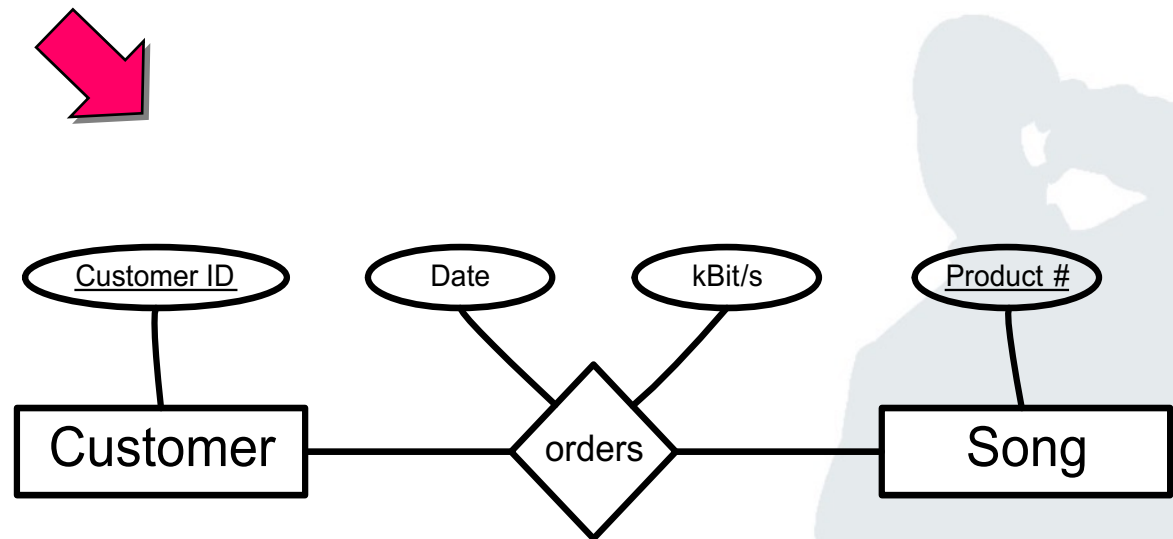
Exercise 1: Entity Relationship Model



- Modelling of the problem statement from functional perspective
- Abstraction from technical aspects and implementations
- Different modelling concepts (e.g. ERM, SERM, ...) available



Customer orders a song.

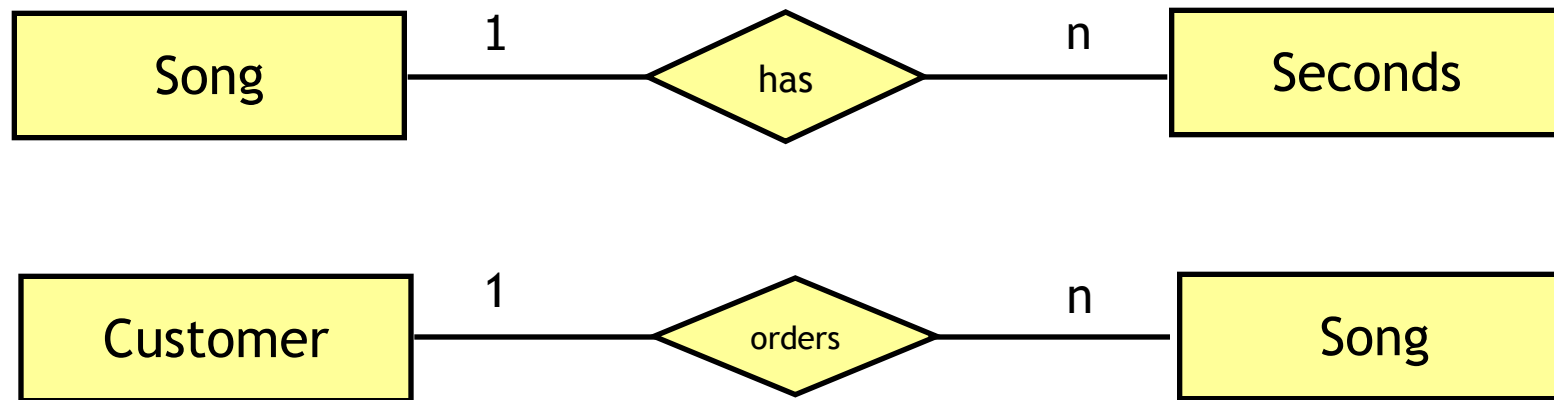


Cardinalities

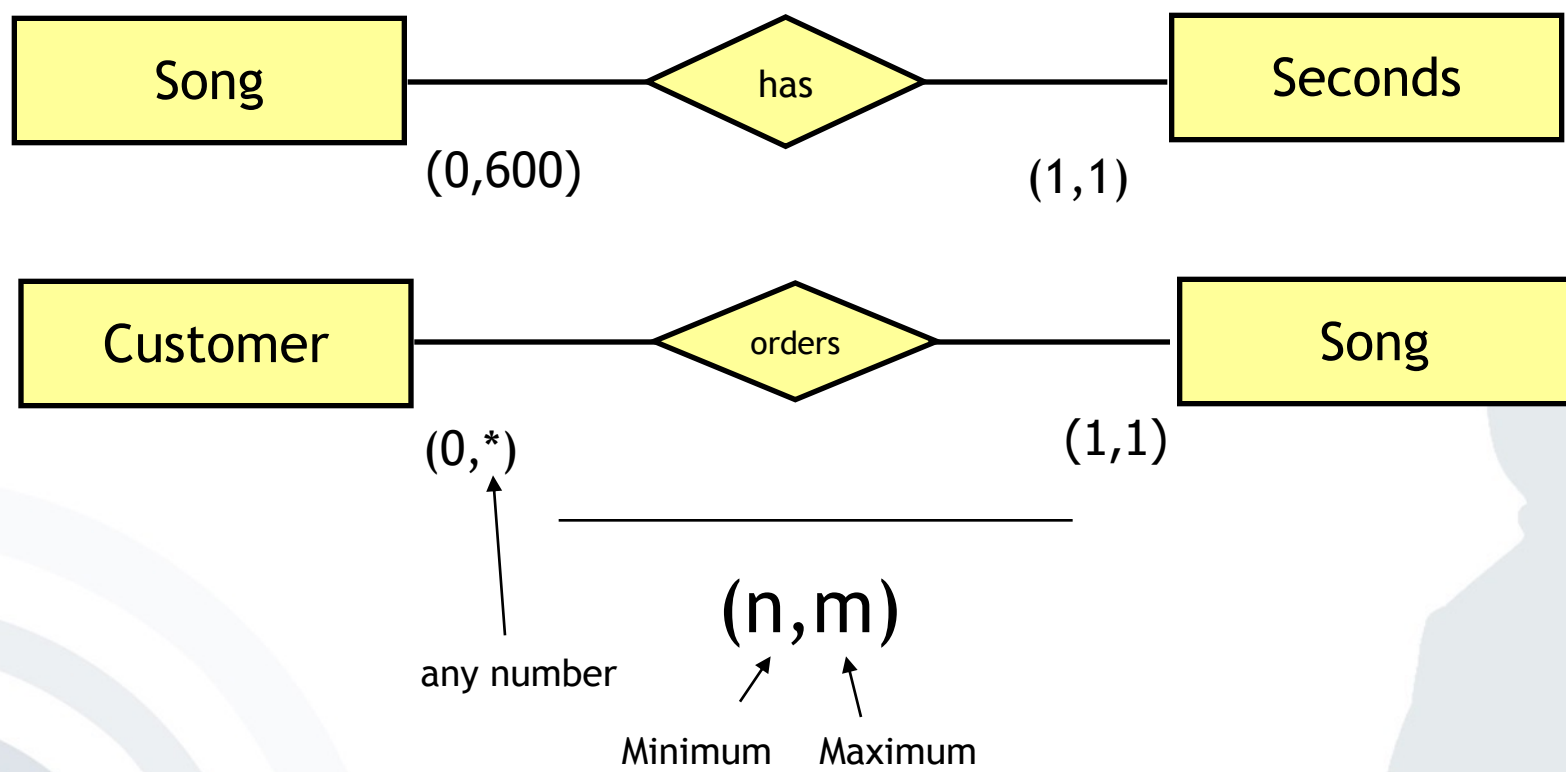
- Cardinalities describe the number of relationship instances that an entity can participate in:
 - 1:1 (one-to-one), e.g. student - student card (theoretically)
 - 1:n (one-to-many), e.g. university - student (theoretically)
 - n:m (many-to-many), e.g. professor - student (theoretically)

Intervals (min/max notation)

- Intervals allow specifying cardinalities more accurately.
- They specify that each entity participates in at least *min* and at most *max* relationship instances.



- Intervals allow specifying cardinalities more accurately.



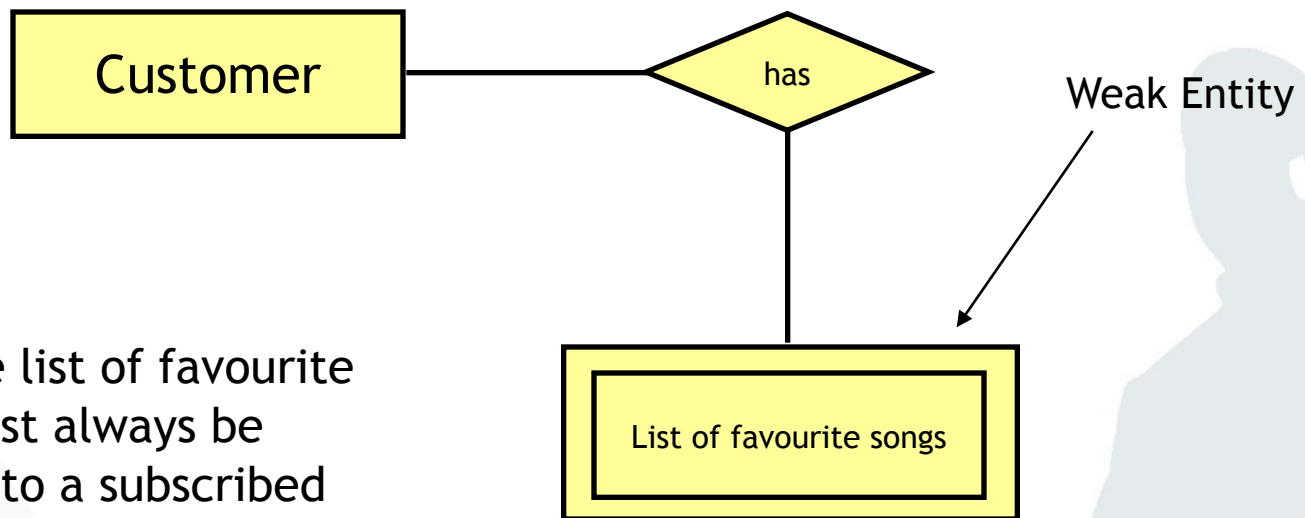
Create an ER model which represents the structure of a university:

- Identify and mark the **primary key** for each entity and avoid as far as possible artificial keys (e.g. ID).
- Define the relations, using the **interval notation**.
- Make explicitly use of **weak entities**.

The ER model should be based on the following information:

- A university consists of different **departments**. Each of them has a name and a unique number.
- Departments are structured into **chairs** with unique names. They offer at least one lecture.
- Each chair offers a number of **lectures** which are described with course number, title and description.
- **Exams** can be distinguished by its type. For each lecture two exams are offered: One normal exam and one repeat exam. The number of participants for an exam is not limited.
- A **student** can register for any number of exams. Furthermore, a student is assigned to one department and has a matriculation number and a name.

Weak entities depend on at least one entity and consequently cannot exist without them.



An active list of favourite songs must always be assigned to a subscribed customer.

Exercise 1: ER Model

1.) Define entities

Department

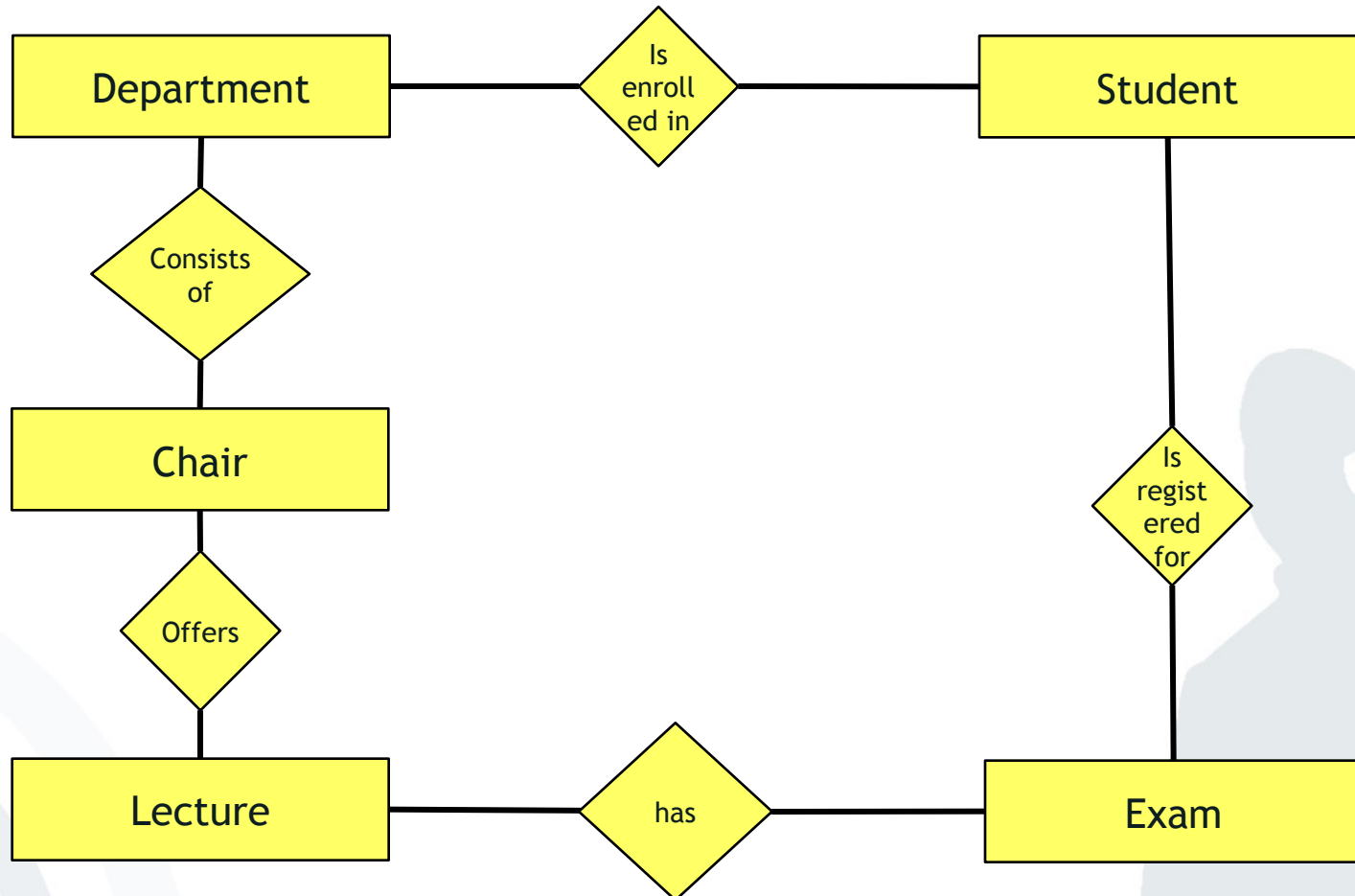
Student

Chair

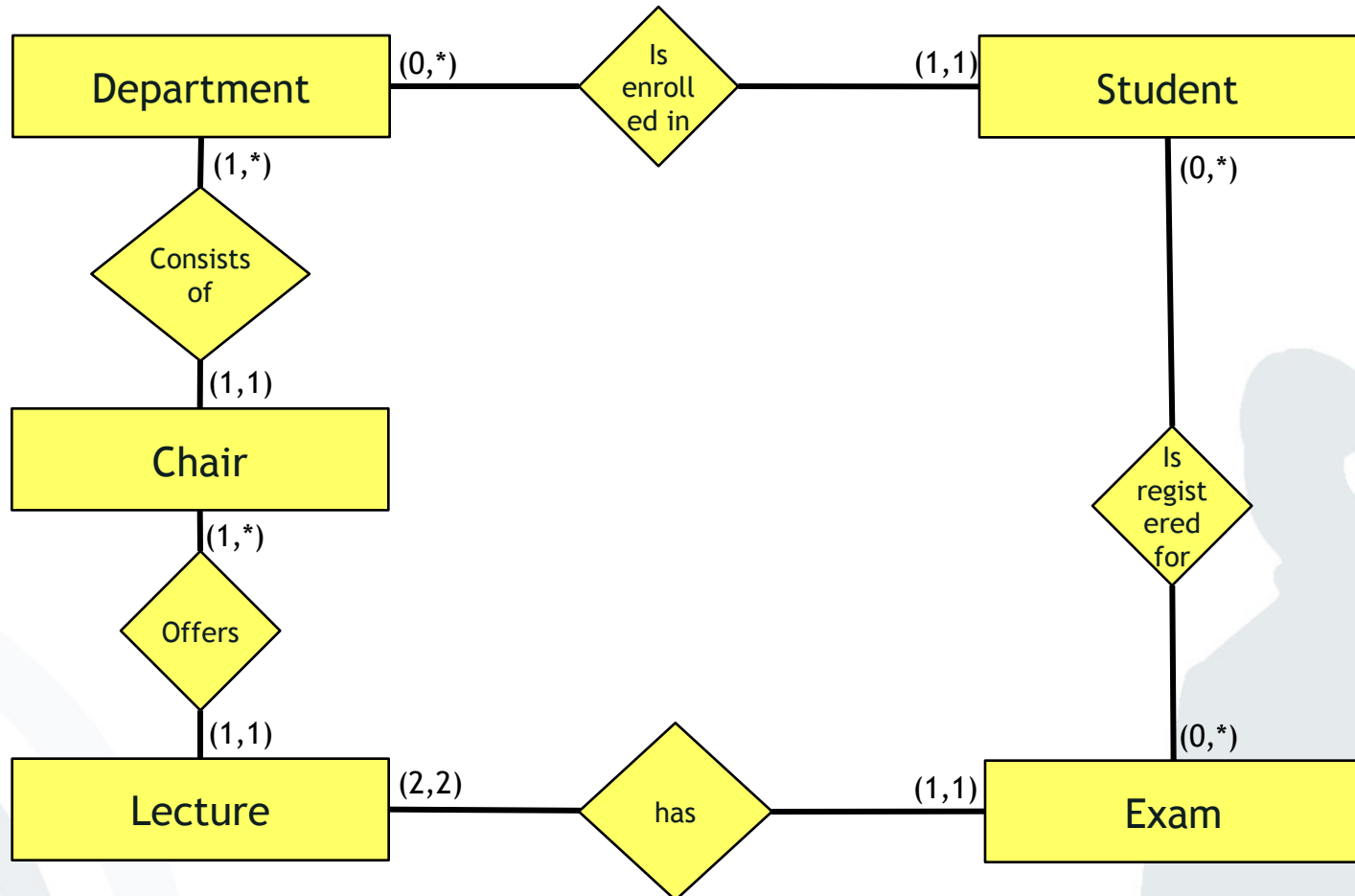
Lecture

Exam

2.) Define relationships between entities



3.) Define cardinalities (using the interval notation)



Create an ER model which represents the structure of a university:

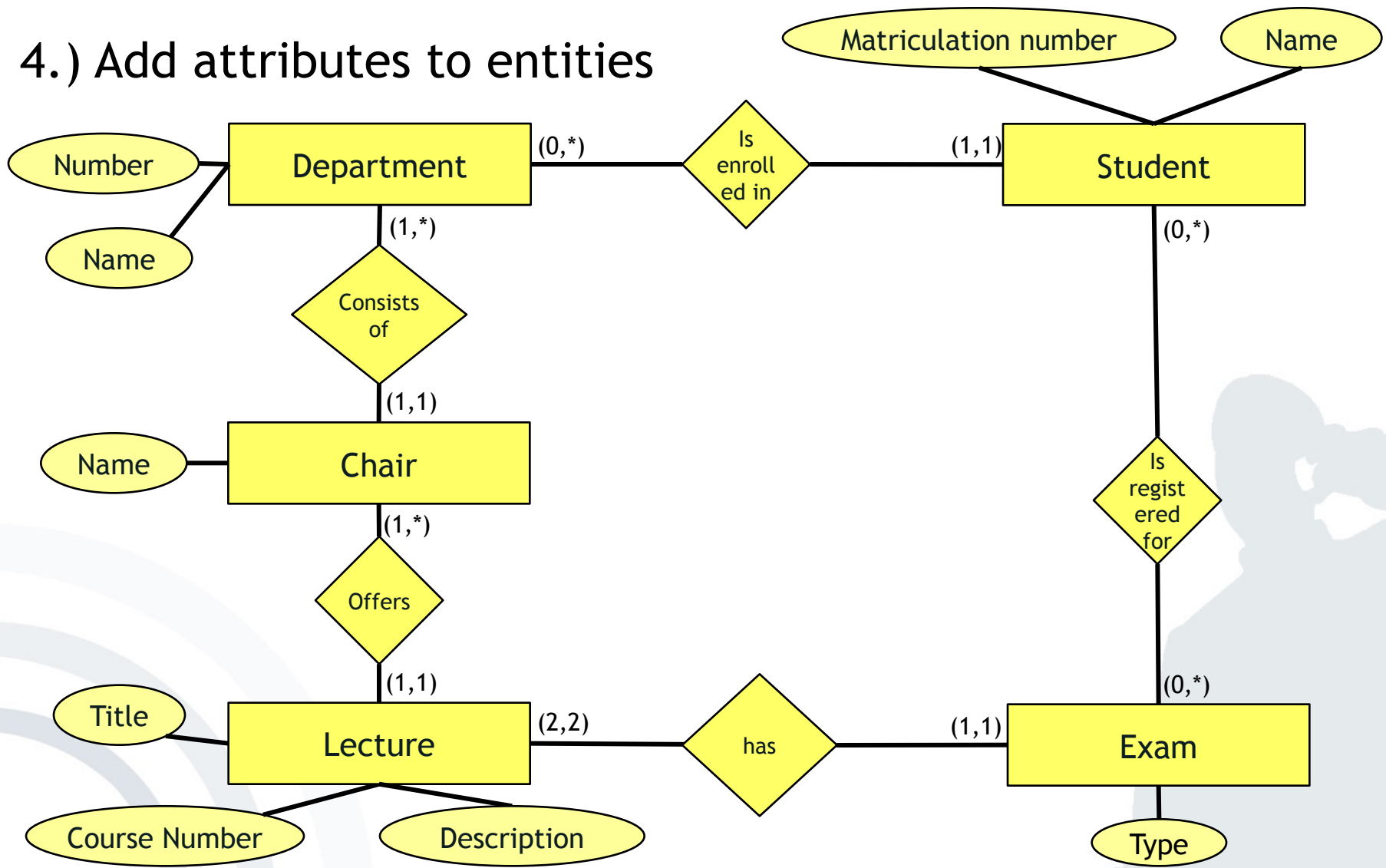
- Identify and mark the **primary key** for each entity and avoid as far as possible artificial keys (e.g. ID).
- Define the relations, using the **interval notation**.
- Make explicitly use of **weak entities**.

The ER model should be based on the following information:

- A university consists of different **departments**. Each of them has a name and a unique number.
- Departments are structured into **chairs** with unique names. They offer at least one lecture.
- Each chair offers a number of **lectures** which are described with course number, title and description.
- **Exams** can be distinguished by its type. For each lecture two exams are offered: One normal exam and one repeat exam. The number of participants for an exam is not limited.
- A **student** can register for any number of exams. Furthermore, a student is assigned to one department and has a matriculation number and a name.

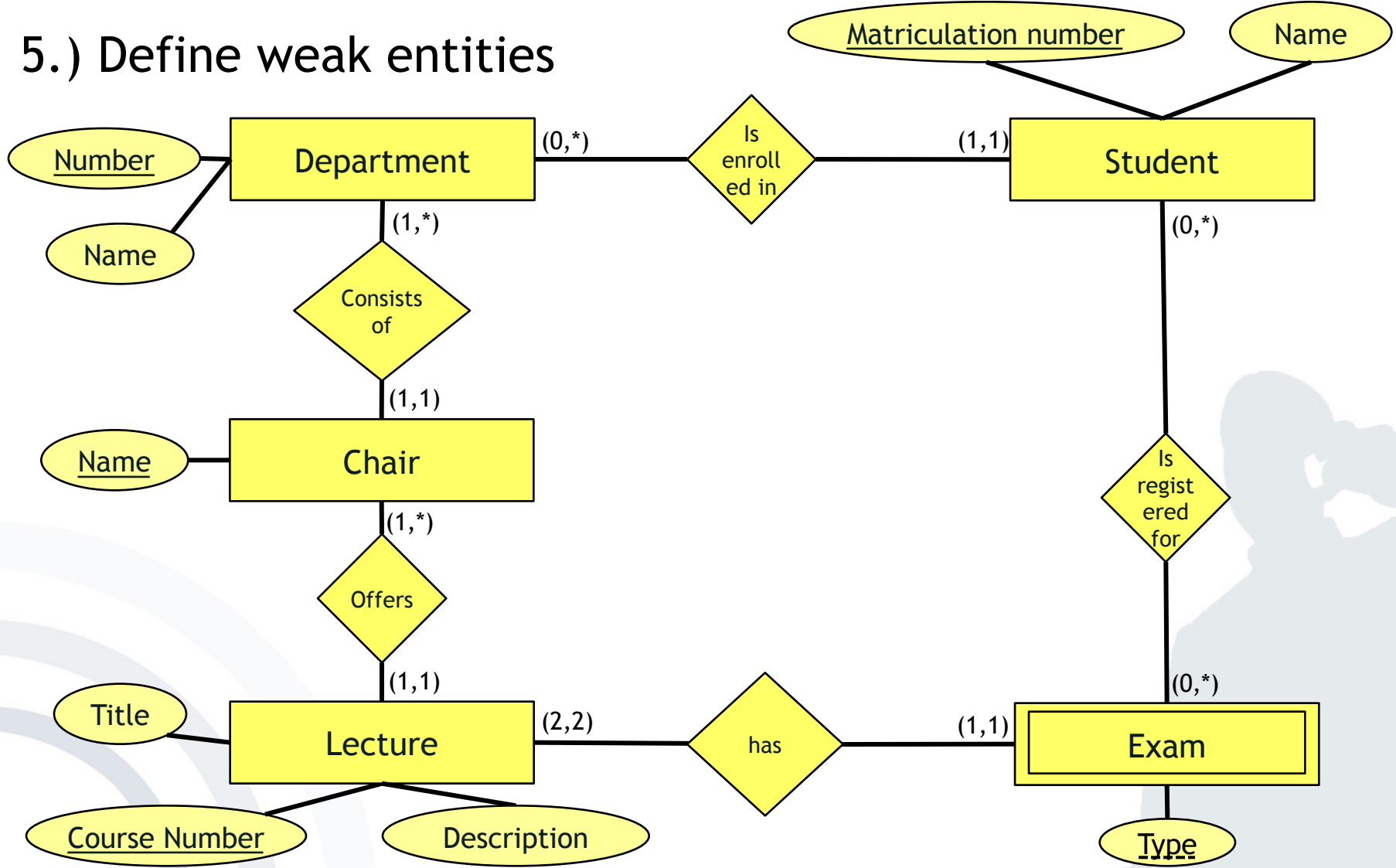
Exercise 1: ER Model

4.) Add attributes to entities

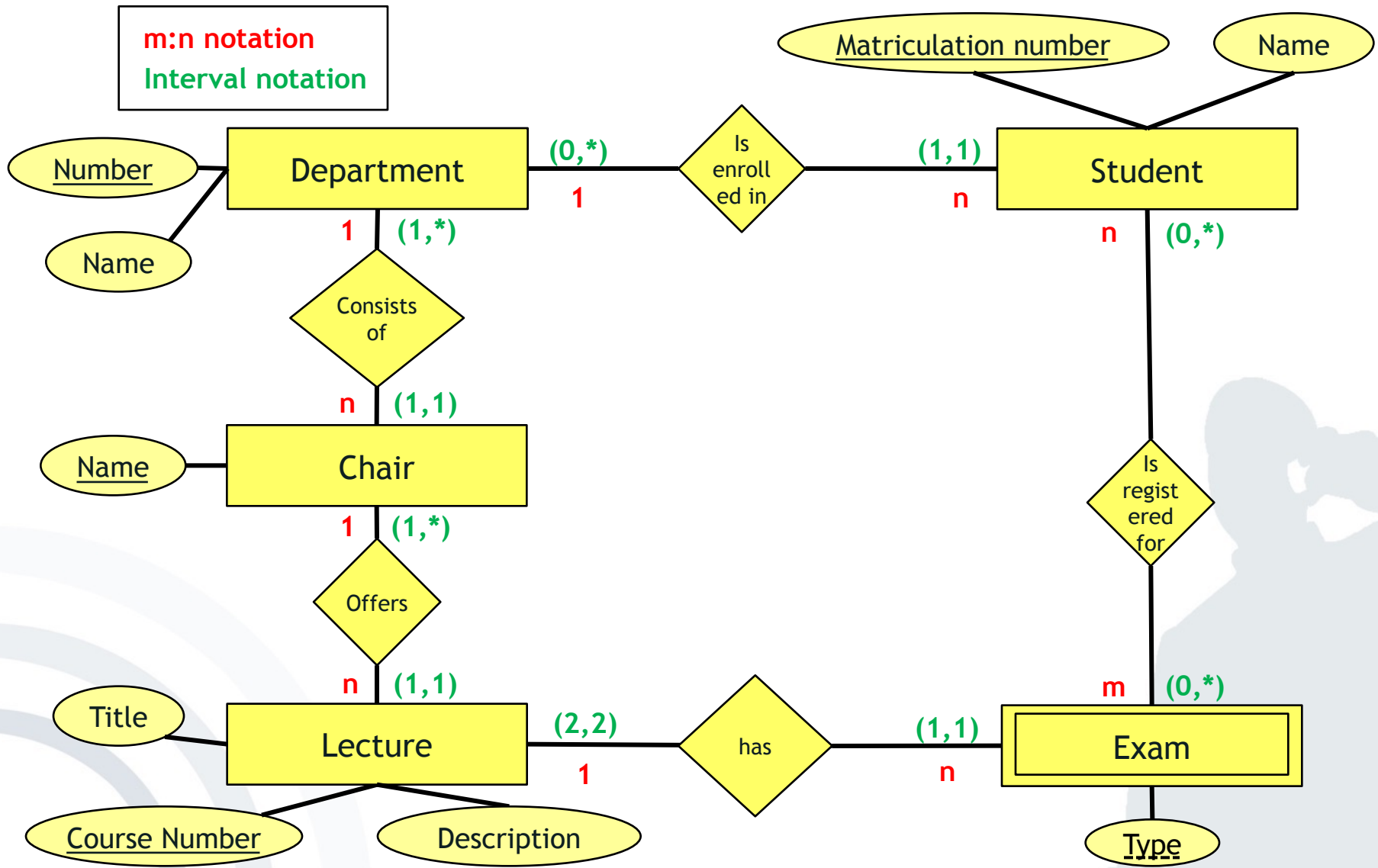


Exercise 1: ER Model

5.) Define weak entities



Exercise 1: ER Model



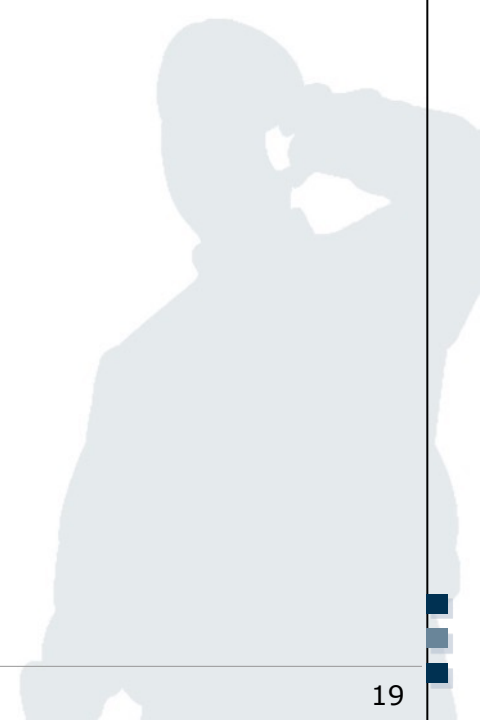
Cardinalities



Intervals (according to Ferstl/Sinz, 2001)

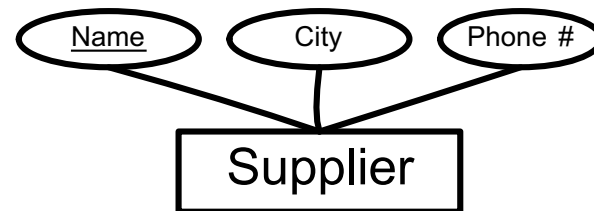
- Exercise 1: Entity Relationship Model
- Exercise 2: Deriving Relations from an ERM
- Exercise 3: SQL

Repetition: Deriving Relations from an ERM



- The relation type with its corresponding attributes is derived from the entity type.

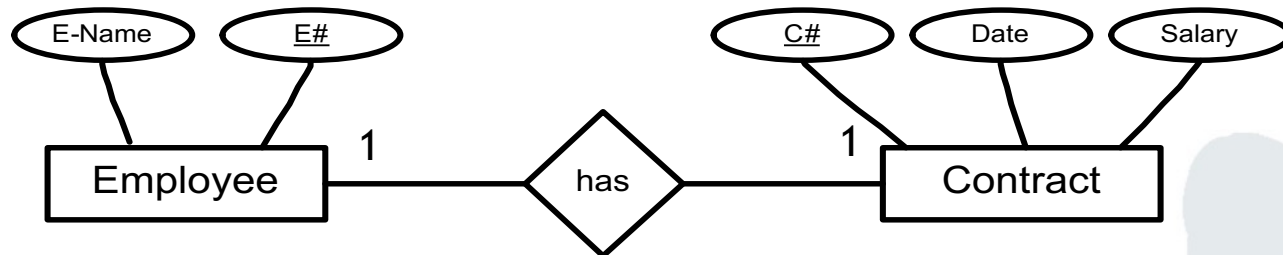
Example:



<u>Name</u>	City	Phone#

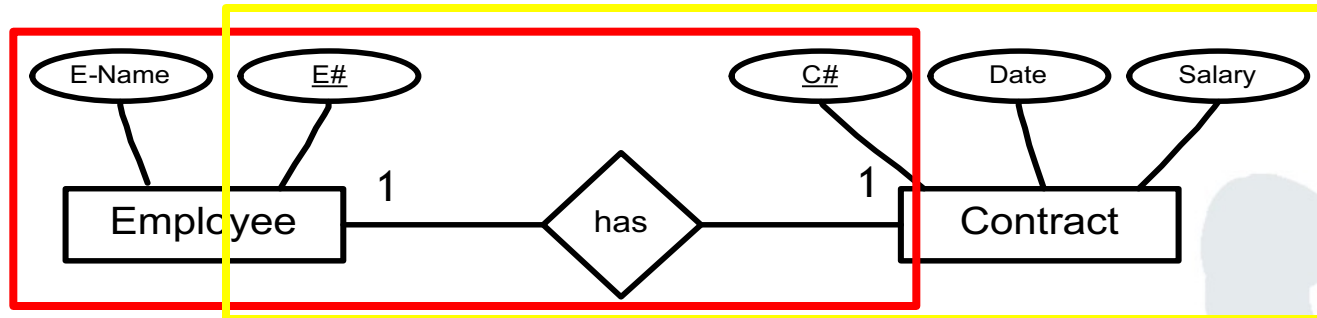
- A 1:1 relationship type does NOT become a relation on its own.
- The information is to be 'attached' to one of the involved entity types.

Example:



- A 1:1 relationship type does NOT become a relation on its own.
- The information is to be 'attached' to one of the involved entity types.

Example:



Alternative 1:

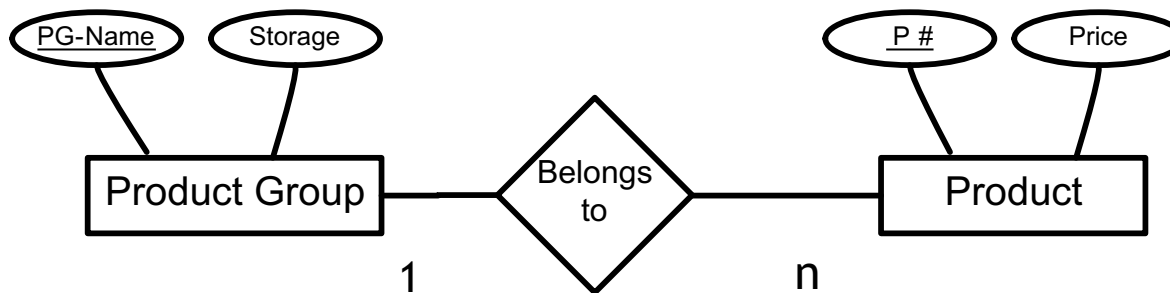
<u>E#</u>	E-Name	C#

Alternative 2:

<u>C#</u>	Date	Salary	E#

- A 1:n relationship type does NOT become a relation on its own.
- The information is to be 'attached' to that relation that corresponds to the entity type with the n-signed edge.

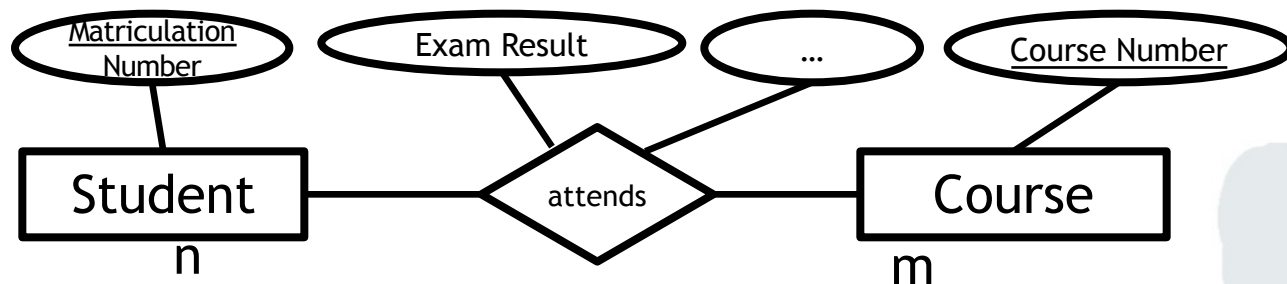
Example:



<u>P #</u>	Price	PG-Name

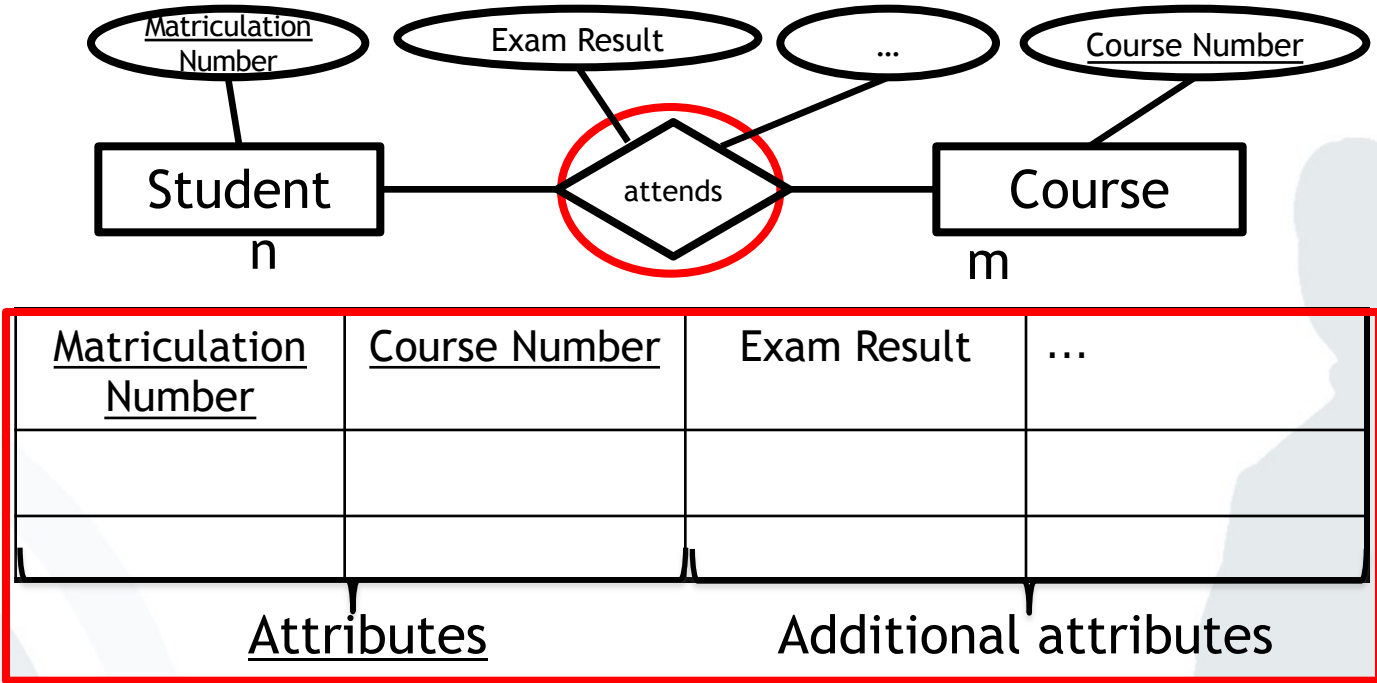
- An n:m-relationship type induces an additional relation-type.
- The relation contains
 - primary keys of involved entity types as attributes
 - and additional attributes of the relation types

Example:



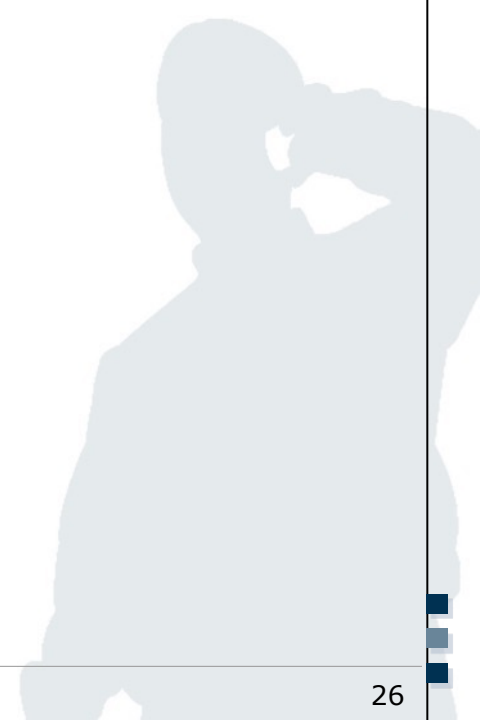
- An n:m-relationship type induces an additional relation-type.
- The relation contains
 - primary keys of involved entity types as attributes
 - and additional attributes of the relation types

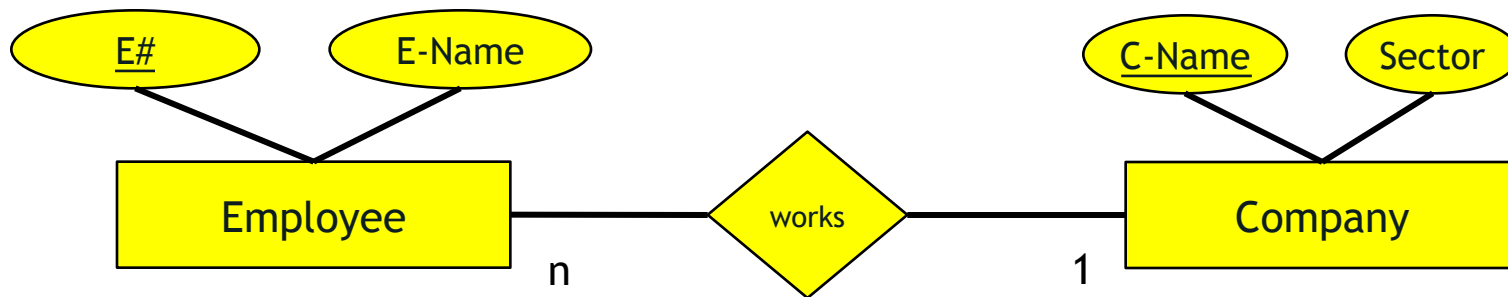
Example:

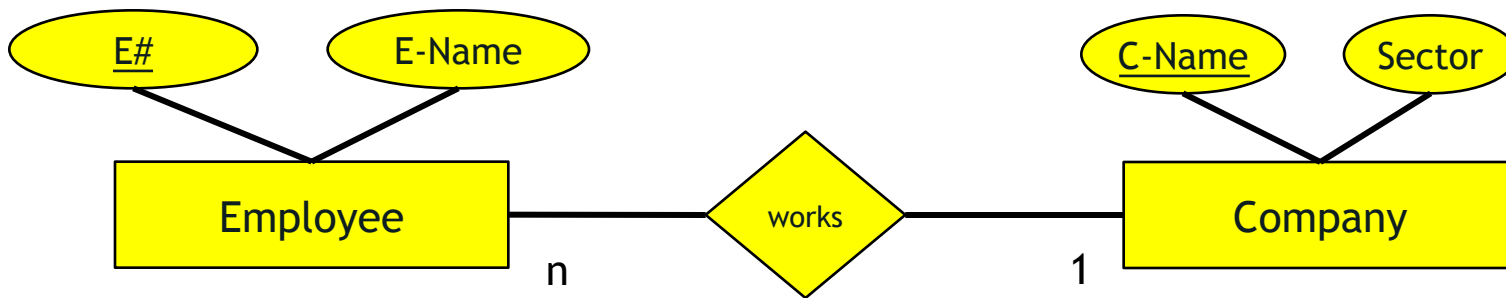


Note: In order to reflect the complete ER Model above, two more relations (Student(Matriculation Number) and Course (Course Number)) are required. The relation above connects both Student and Course entities.

Exercise 2: Deriving Relations from an ERM





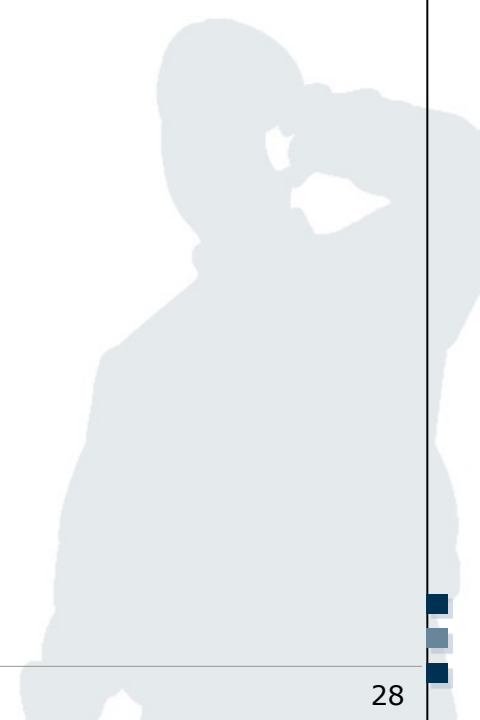


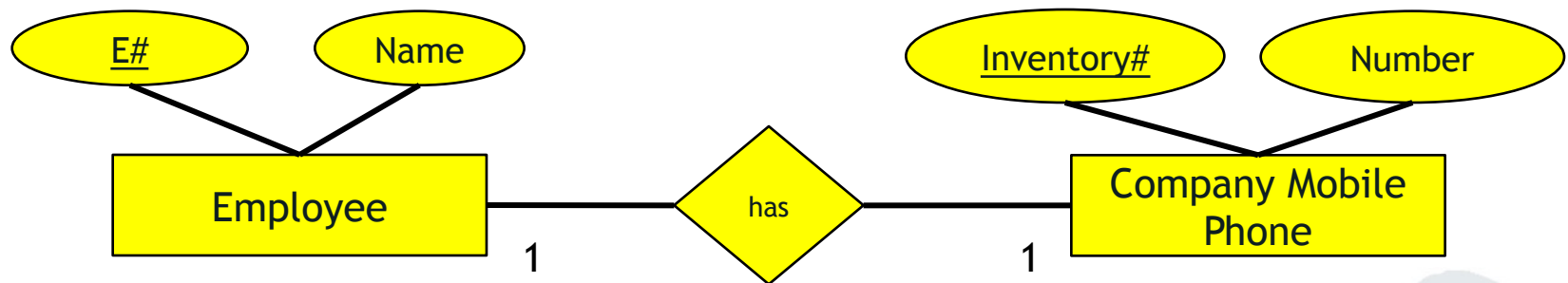
Employee:

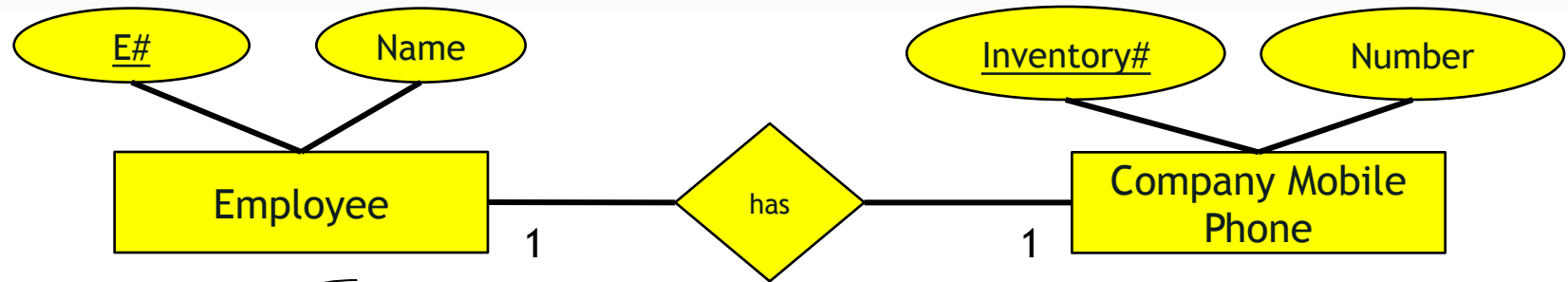
<u>E#</u>	E-Name	C-Name
...

Company:

<u>C-Name</u>	Sector
...	...







Alternative 1
↑
or
↓
Alternative 2

Company Mobile Phone:

<u>Inventory#</u>	Number	E#
...

Employee:

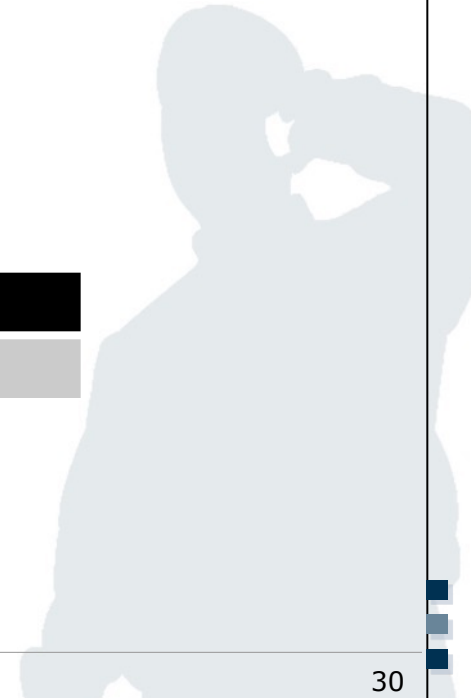
<u>E#</u>	Name
...	...

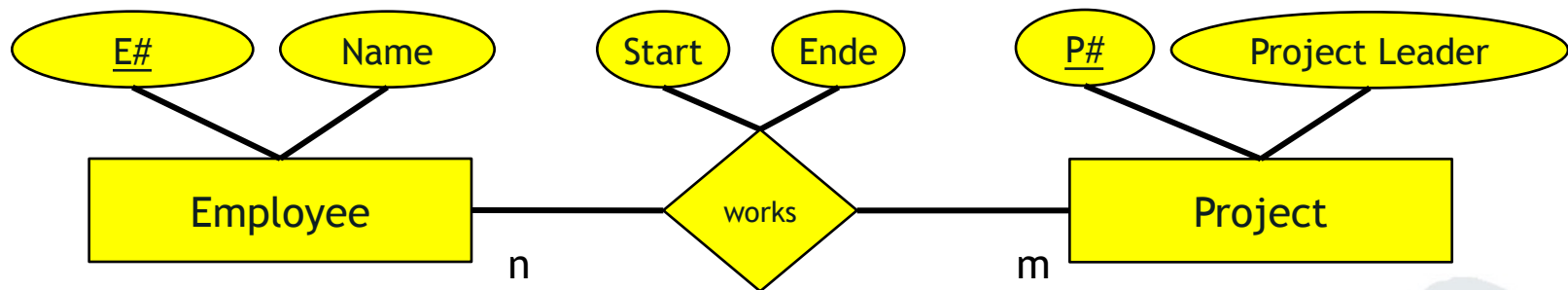
Employee:

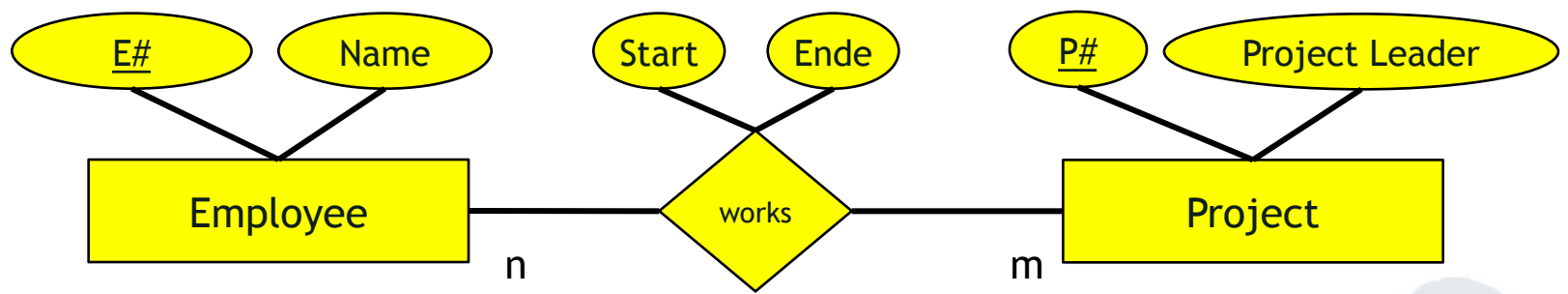
<u>E#</u>	Name	Inventory#
...

Company Mobile Phone:

<u>Inventory#</u>	Number
...	...







Employee:

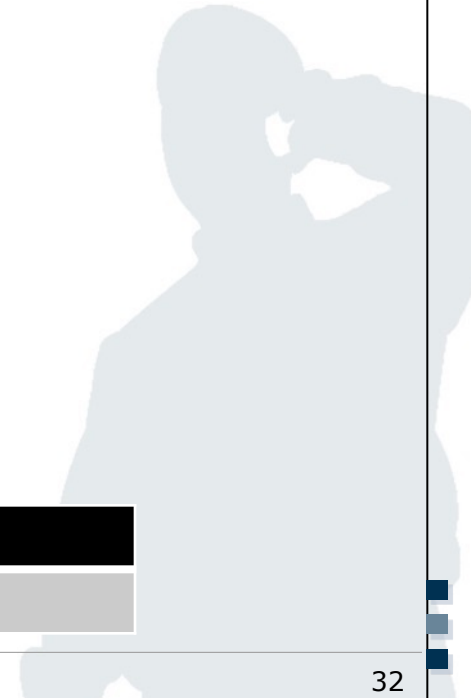
<u>E#</u>	Name
...	...

Project:

<u>P#</u>	Project Leader
...	...

works:

<u>E#</u>	<u>P#</u>	Start	End
...



- Exercise 1: Entity Relationship Model
- Exercise 2: Deriving Relations from an ERM
- Exercise 3: SQL

Write the appropriate SQL statements to answer the following questions and draw the table which will be returned as a result.

The Fortune Bank database consists of the four tables branch, customer, loan and borrower.

Database: Fortune Bank

Table: branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000.00
Downtown	Brooklyn	9000000.00
Mianus	Horseneck	400000.00
North Town	Rye	3700000.00
Perryridge	Horseneck	1700000.00
Pownal	Bennington	300000.00
Redwood	Palo Alto	2100000.00
Round Hill	Horseneck	8000000.00

Table: customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Jackson	University	Salt Lake
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	Main	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Database: Fortune Bank

Table: loan

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Table: borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Exercise 3 a): SQL

a) What is the average amount of loans over all branches?

```
SELECT AVG (amount)  
FROM loan
```

1242.857142

Table: loan

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Exercise 3 b): SQL

b) What is the total amount of loans granted by the Fortune Bank?

```
SELECT SUM(amount)  
FROM loan
```

8700.00

Table: loan

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

c) How many branches does the Fortune Bank have?

```
SELECT COUNT(branch_name)  
FROM branch
```

```
=====  
8  
=====
```

Table: branch

branch_name	branch_city	assets
Brighton	Brooklyn	7100000.00
Downtown	Brooklyn	9000000.00
Mianus	Horseneck	400000.00
North Town	Rye	3700000.00
Perryridge	Horseneck	1700000.00
Pownal	Bennington	300000.00
Redwood	Palo Alto	2100000.00
Round Hill	Horseneck	8000000.00

Exercise 3 d): SQL

d) How many loans were granted exceeding \$1000 ?

```
SELECT COUNT(loan_number)
FROM loan
WHERE amount>1000
```

4

Table: loan

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

e) How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Table: loan

Table: borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

Table: customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Jackson	University	Salt Lake
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	Main	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

e) How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Table: loan

Table: customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Jackson	University	Salt Lake
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	Main	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Table: borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

e) How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Table: loan

Table: customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Jackson	University	Salt Lake
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	Main	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

Table: borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

Table: loan

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

Table: borrower

customer_name	loan_number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

+

=

loan_number	branch_name	amount	customer_name
L-16	Perryridge	1300.00	Adams
L-93	Mianus	500.00	Curry
L-15	Perryridge	1500.00	Hayes
L-14	Downtown	1500.00	Jackson
L-17	Downtown	1000.00	Jones
L-11	Round Hill	900.00	Smith
L-23	Redwood	2000.00	Smith
L-17	Downtown	1000.00	Williams

loan INNER JOIN borrower ON loan.loan_number=borrower.loan_number



How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

loan_number	branch_name	amount	customer_name
L-16	Perryridge	1300.00	Adams
L-93	Mianus	500.00	Curry
L-15	Perryridge	1500.00	Hayes
L-14	Downtown	1500.00	Jackson
L-17	Downtown	1000.00	Jones
L-11	Round Hill	900.00	Smith
L-23	Redwood	2000.00	Smith
L-17	Downtown	1000.00	Williams

Table: customer

customer_name	customer_street	customer_city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Jackson	University	Salt Lake
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	Main	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

+

=

loan_number	branch_name	amount	customer_name	customer_street	customer_city
L-16	Perryridge	1300.00	Adams	Spring	Pittsfield
L-93	Mianus	500.00	Curry	North	Rye
L-15	Perryridge	1500.00	Hayes	Main	Harrison
L-14	Downtown	1500.00	Jackson	University	Salt Lake
L-17	Downtown	1000.00	Jones	Main	Harrison
L-11	Round Hill	900.00	Smith	Main	Rye
L-23	Redwood	2000.00	Smith	Main	Rye
L-17	Downtown	1000.00	Williams	Nassau	Princeton

(loan INNER JOIN borrower ON loan.loan_number=borrower.loan_number)
 INNER JOIN customer ON borrower.customer_name = customer.customer_name

How many borrowers are serviced by the branch 'Downtown' and live in Princeton?

```
SELECT COUNT(customer.customer_name)
FROM (loan INNER JOIN borrower ON
        loan.loan_number=borrower.loan_number) INNER JOIN customer ON
        borrower.customer_name = customer.customer_name
WHERE branch_name='Downtown' AND customer_city='Princeton'
```

1

f) Insert a new loan in the table 'loan'.

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00



loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00
L-94	Downtown	4000.00

```
INSERT INTO loan (loan_number, branch_name, amount)
VALUES ('L-94', 'Downtown', 4000)
```

g) Delete the previously inserted entry from the table 'loan'.

loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00
L-94	Downtown	4000.00



loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00

```
DELETE FROM loan
WHERE loan_number='L-94'
```

- [W3Schools.com](https://www.w3schools.com) provides databases, exercises and explanations
 - → Will be used in the mentoriums
- [Mystery.knightlab.com](https://mystery.knightlab.com) provides a murder mystery game to learn SQL
 - → Find the murder in several police databases, using SQL

Open Questions?

- When creating a table, its unique name, primary key and the list of attributes must be defined

```
CREATE TABLE Kunde (Name char (25) , Vorname char (25) , Straße char (25) , Stadt char (25) ) ;
```

Data types

Data Type	
integer(size)	Integer, “size” defines the maximum number of digits
decimal(p,s)	p specifies the maximum total number of decimal digits that can be stored, both to the left and to the right of the decimal point. s specifies the maximum number of decimal digits that can be stored to the right of the decimal point.
char(size)	Fixed-length character data (length of “size”)
varchar(size)	Variable-length character data (maximum length of “size”)
date(yyyymmdd)	Date [and time] with all four digits of the year, month, day, [hour (in 24-hour format), minute, and second], e.g. 20070115
...	

DELETE FROM Orders

Data Manipulation Language: Deletes all entries in the table Orders. New entries can be added again with INSERT INTO.

DROP TABLE Orders

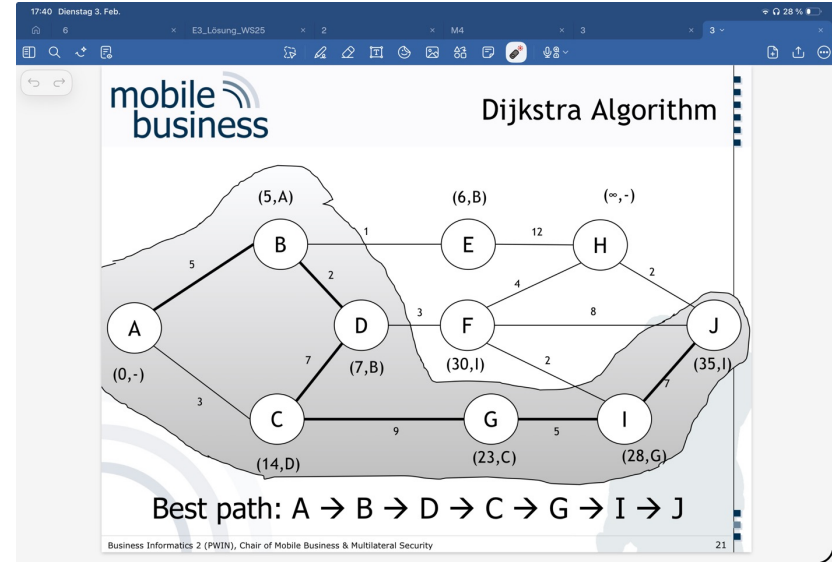
Data Definition Language: Deletes the table Orders. The table must be recreated with CREATE TABLE.

- Do we have to Capitalize operations?
 - No
- How to write “not equal to” in SQL?
 - <> OR != (see MySQL)
- When to use “GROUP BY”?
 - Attributes in a select clause that are outside of the aggregate functions must appear in a group by list

```
SELECT OrderDate, COUNT (OrderDate)  
FROM Orders  
GROUP BY OrderDate
```

Question:

In the Dijkstra Algorithm exercise (Mentorium 3), the provided solution states that the shortest path is **A-B-D-C-G-I** with a total cost of **35**. However, alternative paths such as **A-B-D-F-J** appear to have a significantly lower total cost (**18**). Can you clarify why this alternative path is not considered valid, or where the reasoning differs from the intended solution approach?



Answer:

This example is intended to illustrate that Dijkstra's algorithm **cannot reliably find the longest path** in a graph, because there may be longer paths that the algorithm is unable to detect or verify.

Question:

In the Mentorium and Exercise answers, several provided solutions seem incomplete or unclear:

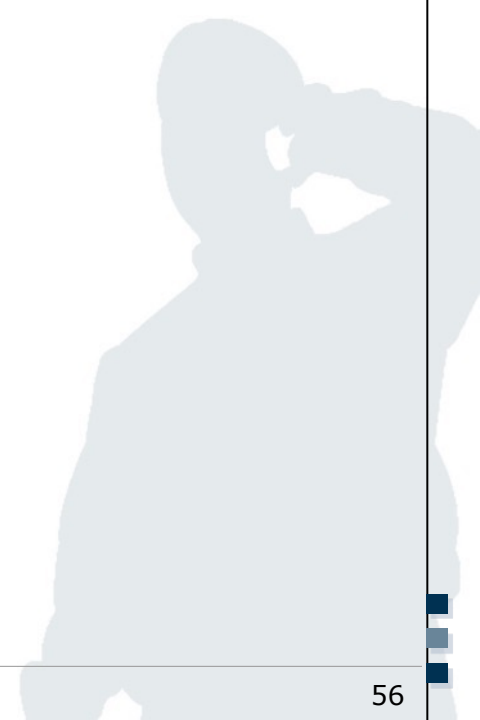
- Mentorium 3 (p.11): “*Should myPlace use TCP or UDP?*” - no reasoning given.
- Mentorium 2 (p.17, p.23): Questions on native vs. web apps and typical website-app relationships - answers lack detailed justification or context for myPlace.
- Exercise 1 (p.24): High-level ARIS model provided, but the InstaMatch modeling itself is missing.

Answer:

These slides were discussed during the Mentorium/Excercise sessions. To answer the questions, review the preceding slides in the respective Mentorium/Excercise and justify your choice of implementation or approach. It is important that you **understand the underlying concepts** and can **explain why a particular solution makes sense for your use case**.

Question:

In Exercise 2b, the solution for the ring topology mentions “*no single point of failure*” and “*slow if one way is broken.*” My understanding is that this only applies to a **dual ring**, because in a single ring, the failure of one device would halt the entire network.



Question:

Are there any example examinations available in English? The example exams currently available in the course portal are in German. If English versions exist, could you please indicate where they can be accessed or whether they can be provided by email?

Answer:

I'll will check with Prof. Rannenberg if there is a way to make English exams available this week. I cannot promise that.

