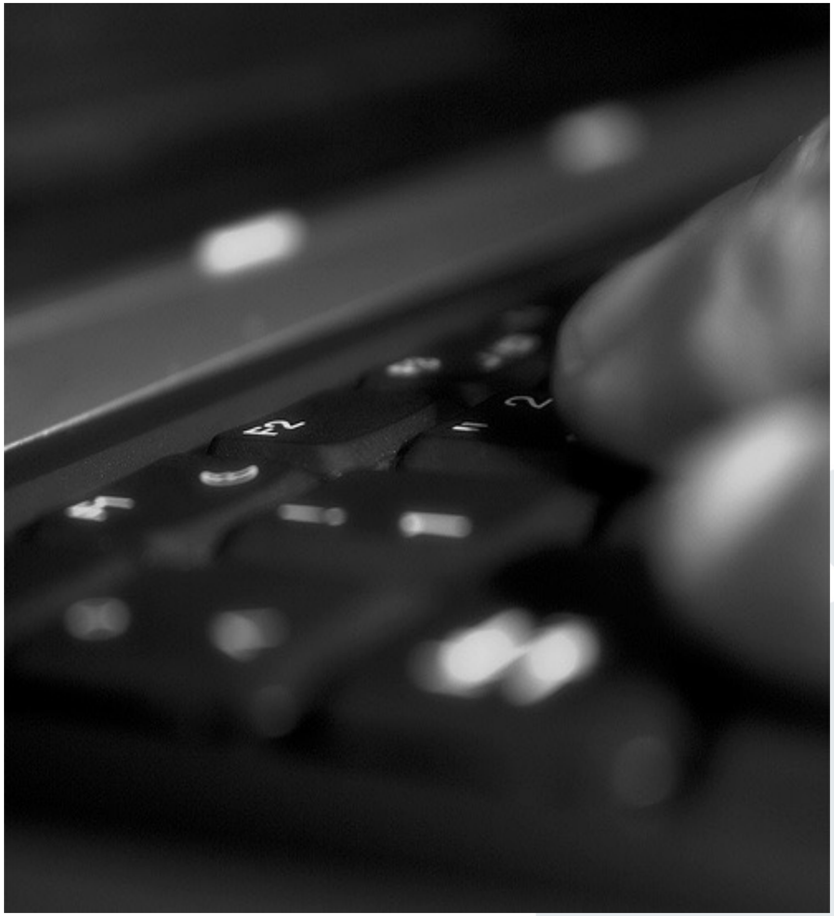
Mentorium 6

Business Informatics 2 (PWIN)

Databases & Data-oriented Modelling

SQL

Sascha Löbner M.Sc.
www.m-chair.de



Jenser (Flickr.com)

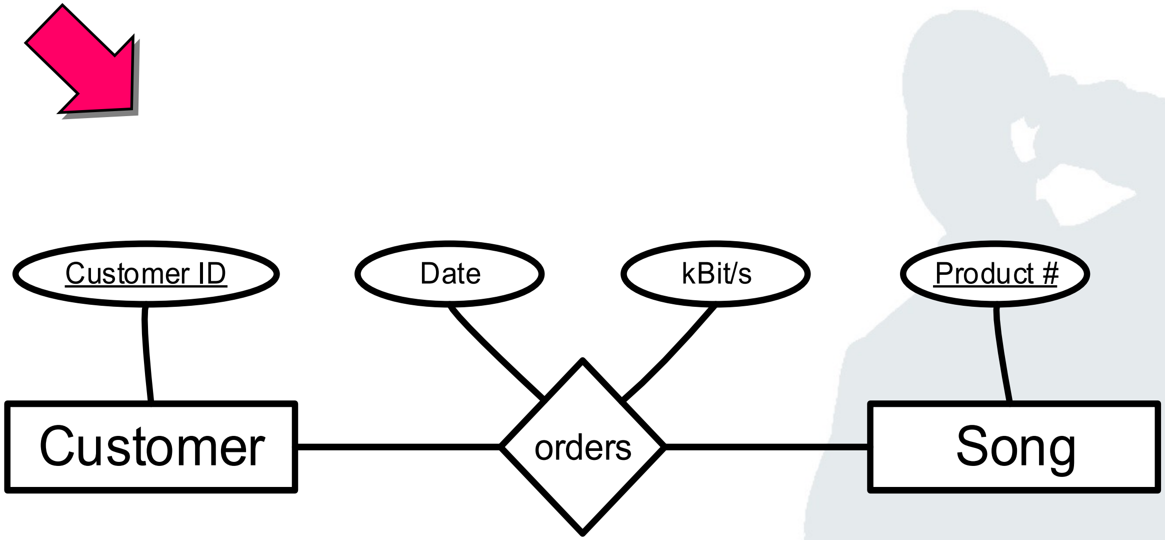
Agenda

- Entity Relationship Model
- SQL

Repetition: ERM

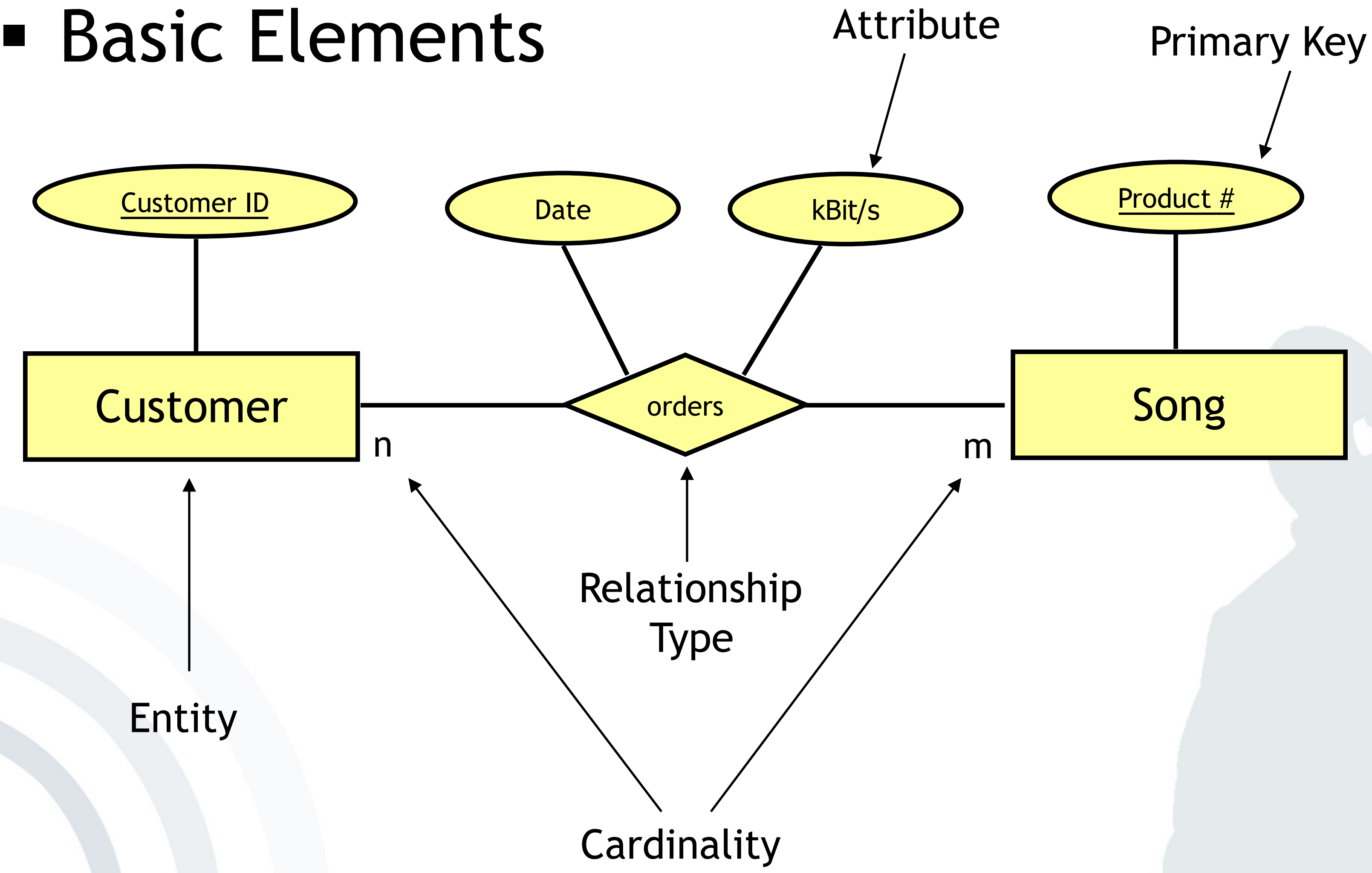
- Modelling of the problem statement from functional perspective
- Abstraction from technical aspects and implementations
- Different modelling concepts (e.g. ERM, SERM, …) available

Customer orders
a song.



Repetition: ERM

- Basic Elements



Repetition: ERM

Cardinalities

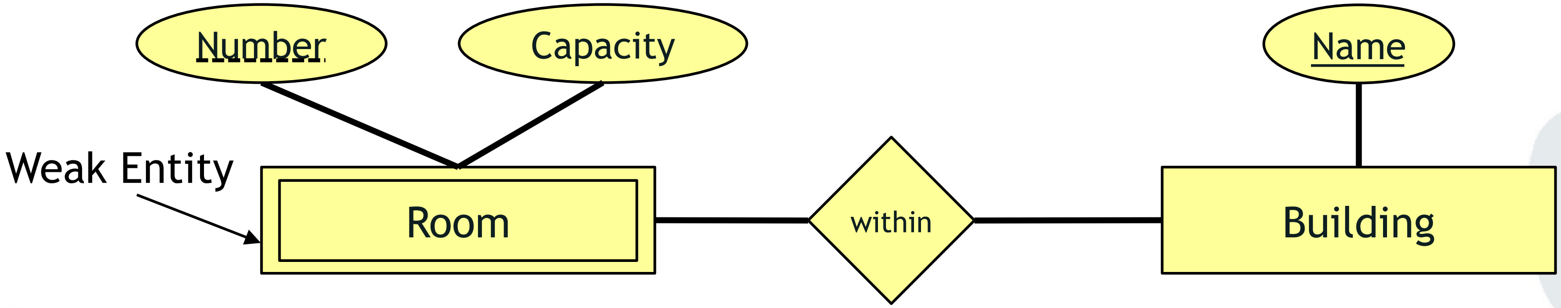
- Cardinalities describe the number of relationship instances that an entity can participate in:
  - 1:1 (one-to-one), e.g. student – student card (theoretically)
  - 1:n (one-to-many), e.g. university – student (theoretically)
  - n:m (many-to-many), e.g. professor – student (theoretically)

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.

Repetition: ERM

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



One cannot uniquely identify a room by its number, but with a combination of room number and name of a building.

Exercise 1: Entity Relationship Model

Exercise 1: ER Model (Part 1)

- Create an ER model which represents the following information. Specify the cardinalities of the relationships using both the n:m notation and the interval notation.
  1. A skill can be needed by many projects but might not be needed by any project. A project needs one or more skills.
  2. An employee can manage many projects. There are some employees who don't manage any projects. A project must be managed by an employee.
  3. An employee may have many skills but might not have any. A skill can be possessed by many employees. There are some skills that no employees possess.

Exercise 1: ER Model (Part 1)
Define Entities

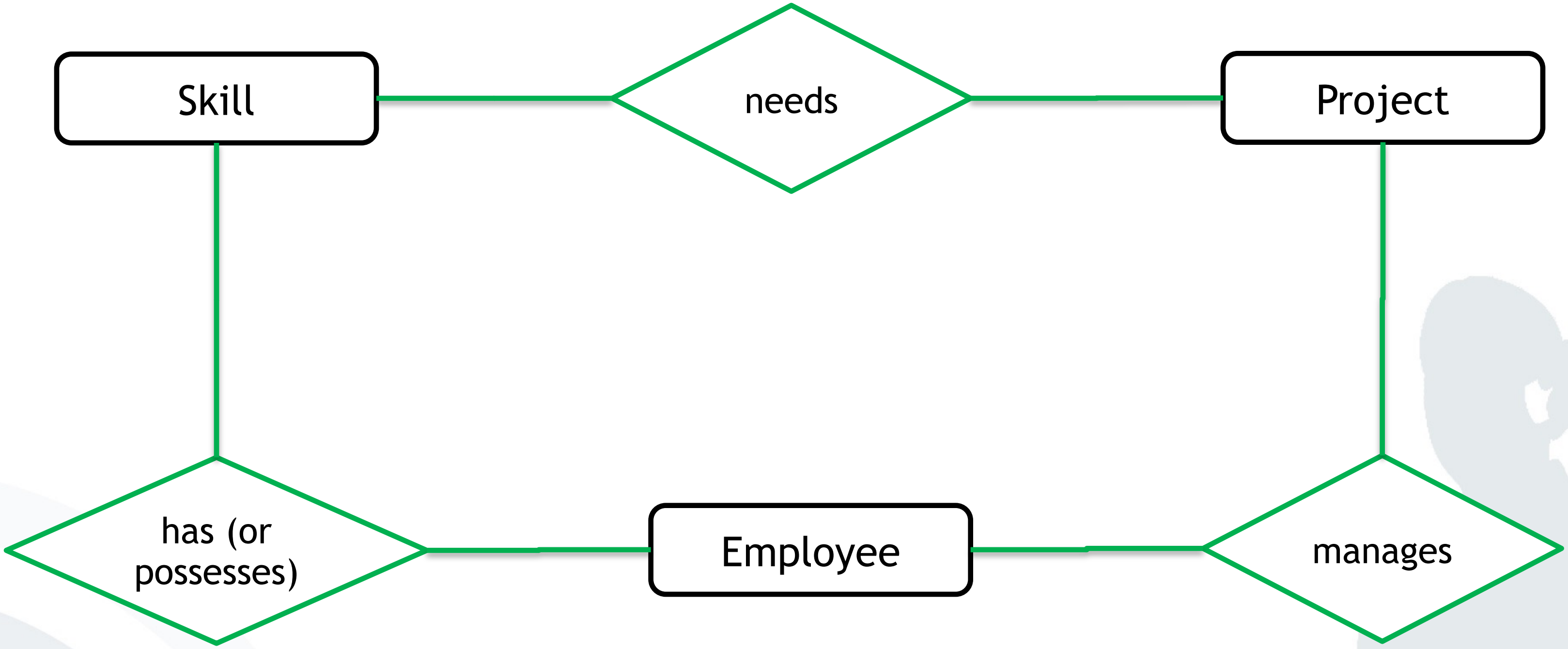
Skill

Project

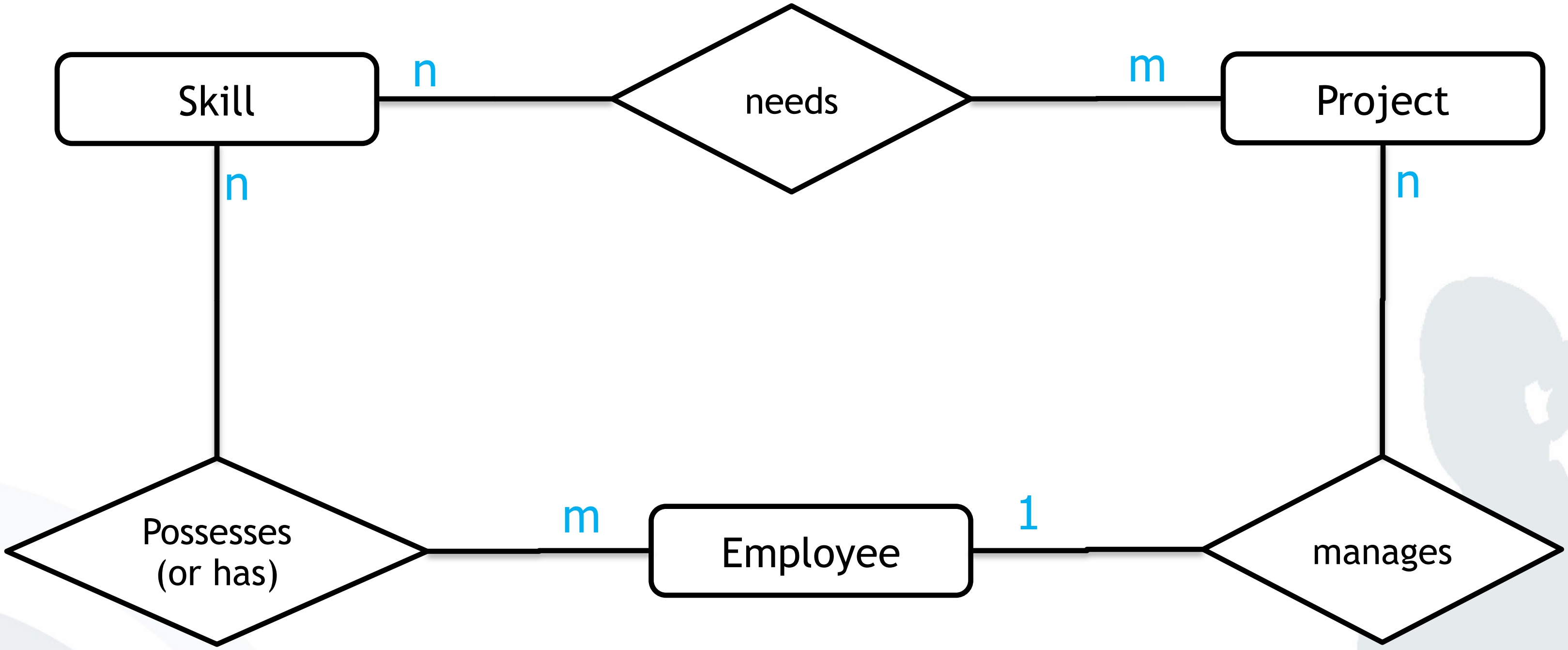
Employee

Exercise 1: ER Model (Part1)

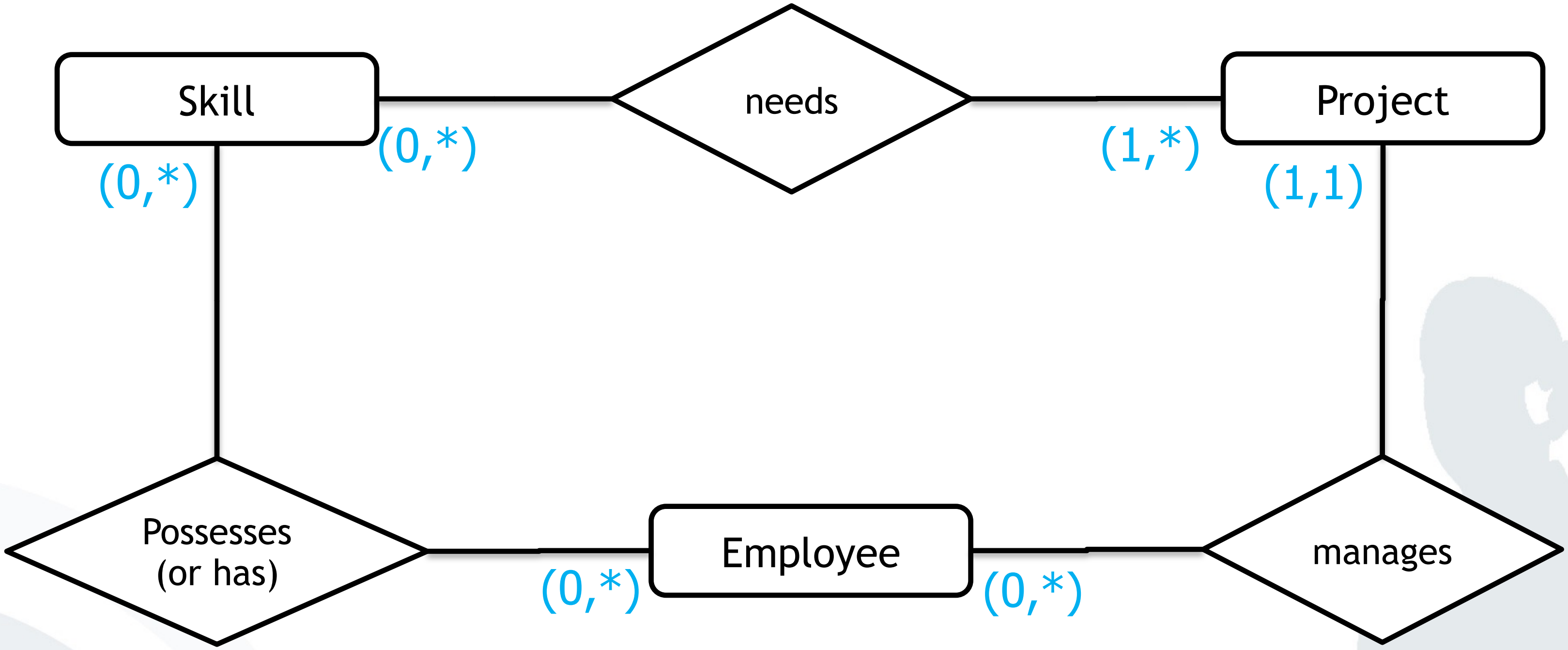
Create Relations



Exercise 1: ER Model (Part 1)
Define Cardinalities

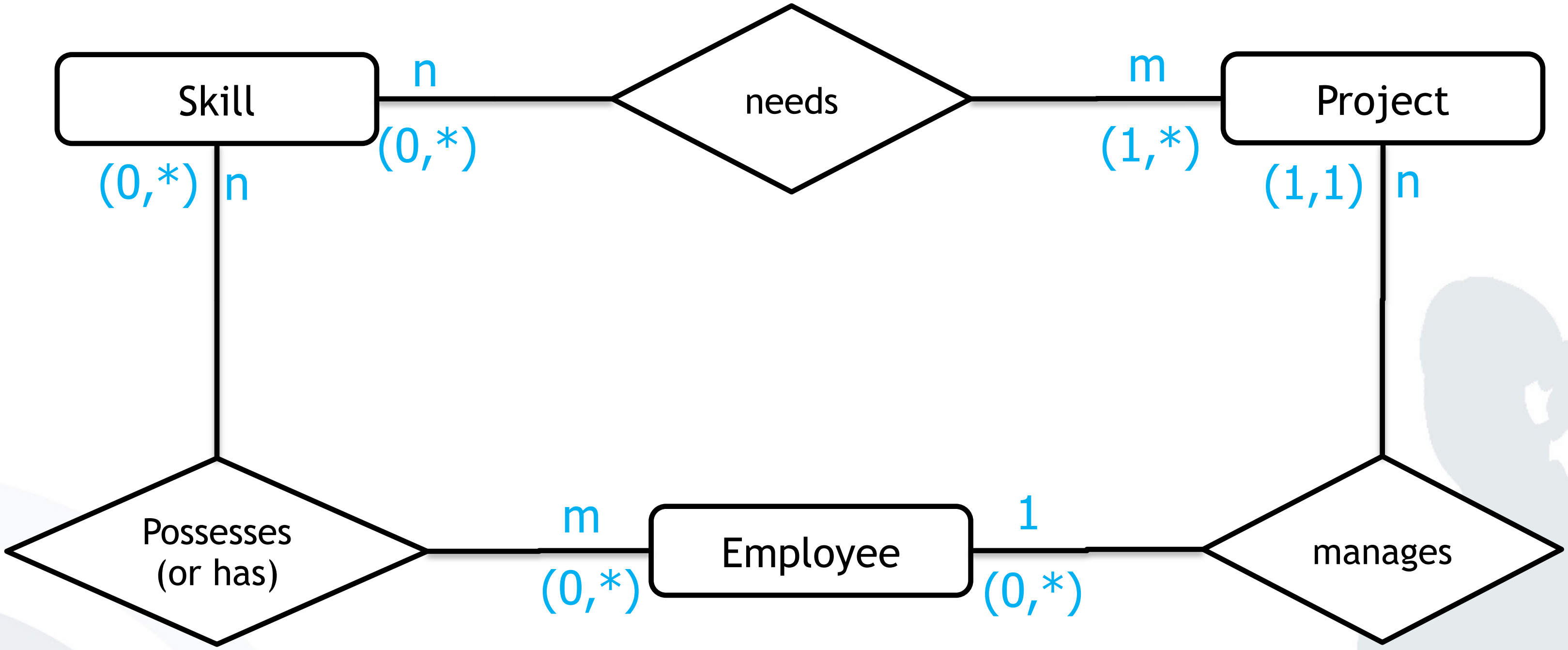


Exercise 1: ER Model (Part 1)
Define Cardinalities



Exercise 1: ER Model (Part 1)

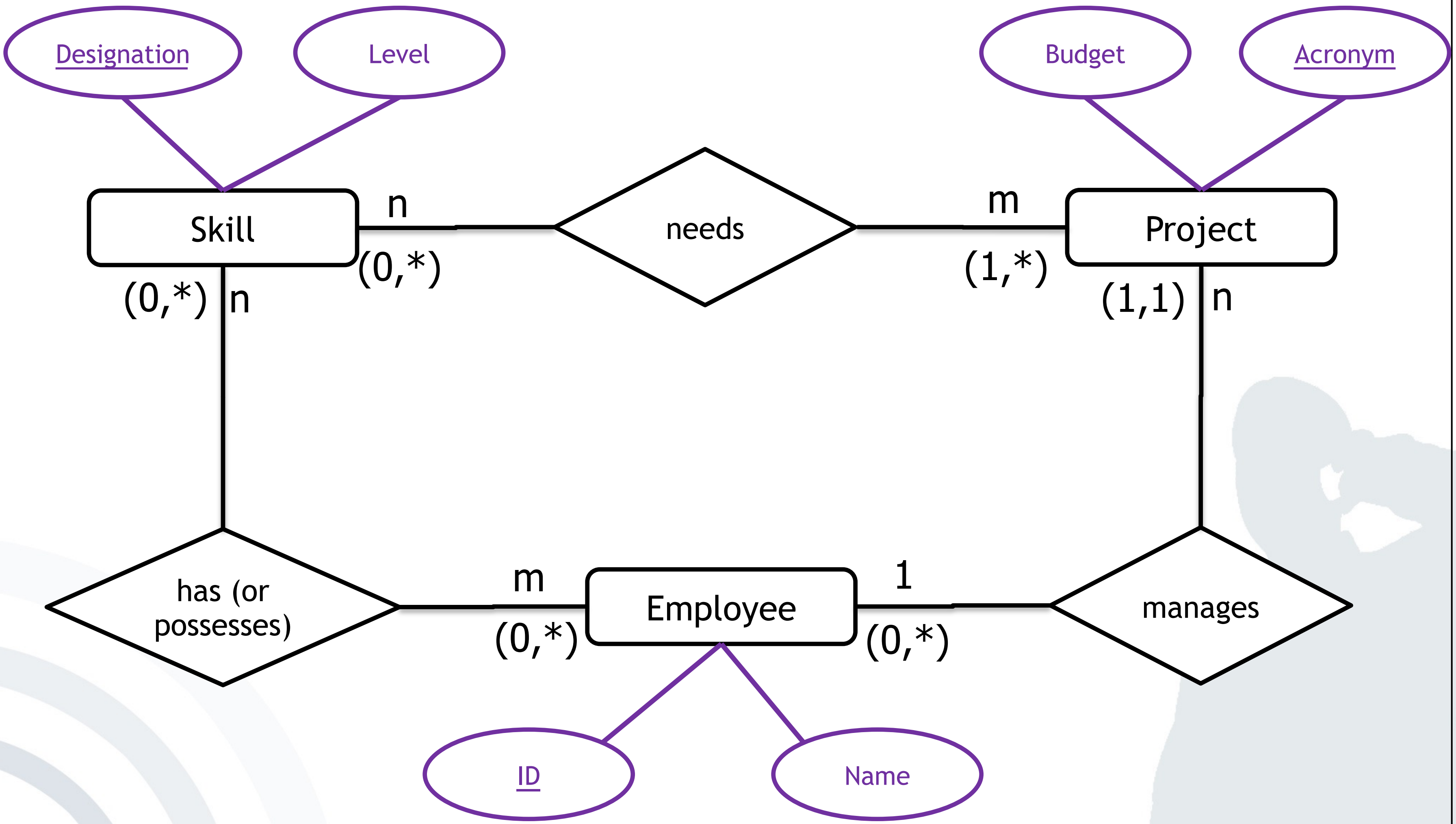
Define Intervalls



Exercise 1: ER Model (Part 2)

- Add attributes to entities with the help of the following information:
  1. A project has a unique acronym and a budget.
  2. An employee has an ID and a name.
  3. A skill is described by its designation and level.

Exercise 1: ER Model (Part 2)
Add Attributes



Exercise 2: Entity Relationship Model

Mentorium 6,

Exercise 2: ER Model

Create an ER model of the InstaMatch® system.

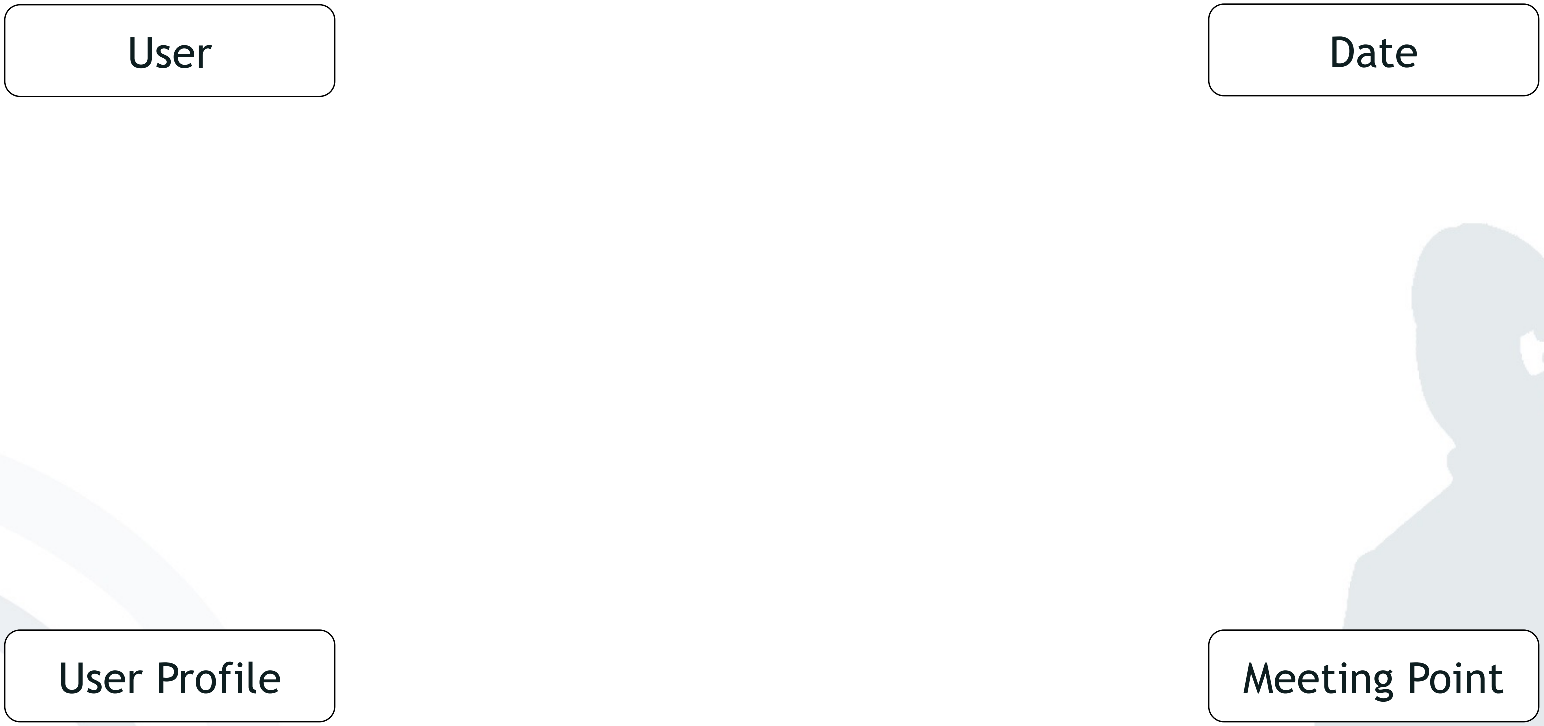
- Identify and mark the **primary key** for each entity and avoid as far as possible artificial keys (e.g. ID).
- Define the cardinalities, using the **n:m notation**.
- Make explicitly use of **weak entities**.

Use the following entities for your model:

- **Users** have a user profile. Each user chooses a unique pseudonym.
- The preference attributes of a user are stored in a **user profile**. These preference attributes are interests, age, and a unique user ID.
- Several users can have multiple **dates**. A date is only defined by its time. Multiple dates can happen at the same meeting point at the same time.
- A **meeting point** has a unique name, an address and a description. Each date has only one meeting point

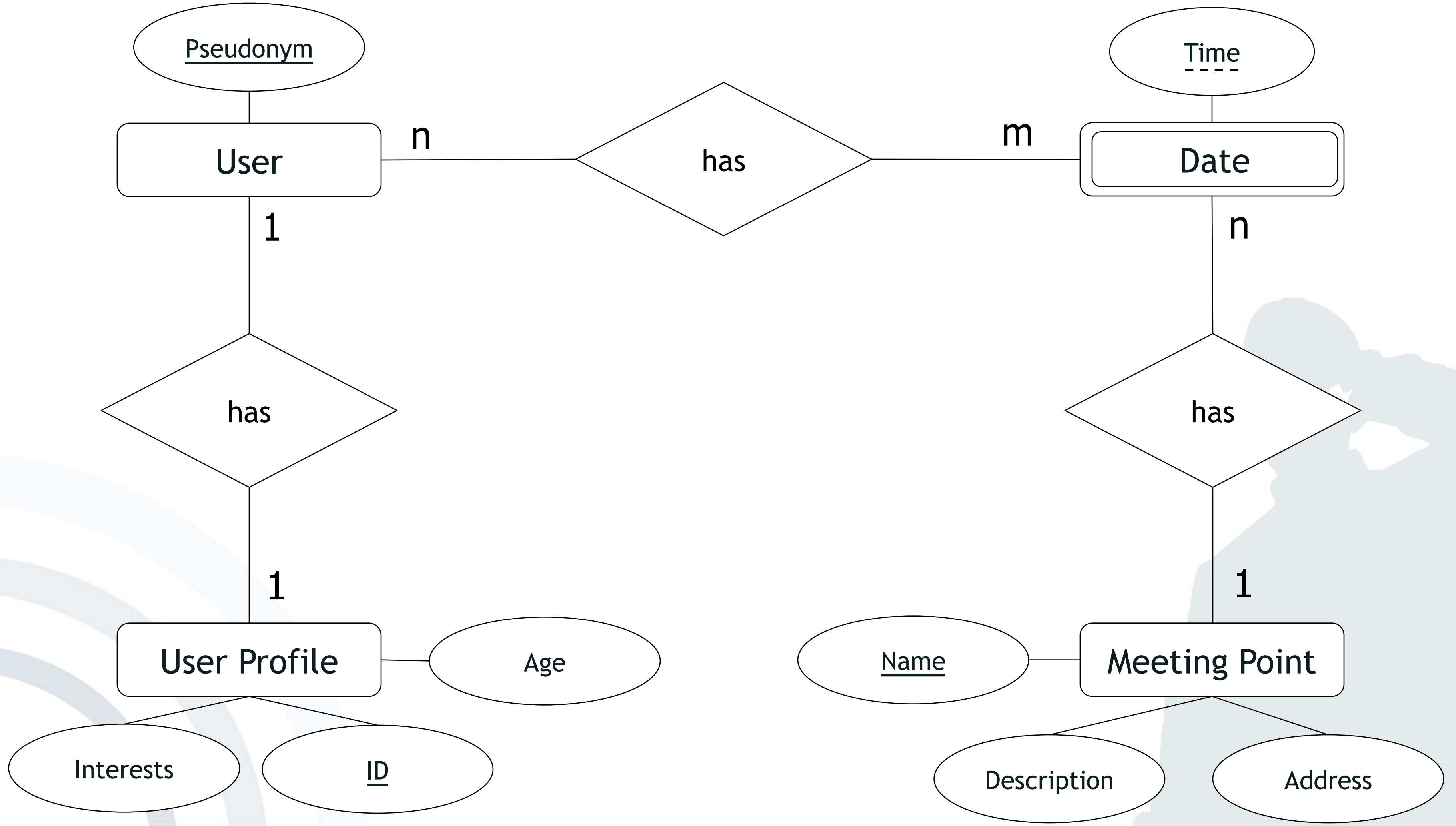
Exercise 2: ER Model

1.) Define entities



Exercise 2: ER Model

5.) Define weak entities



Agenda

- Entity Relationship Model
- SQL

SQL Repetition

Structure of Basic Elements

- Structure of the basic elements
  - SELECT attribute(s)
  - FROM relation(s)
  - [WHERE condition(s)]
  - [GROUP BY attribute(s)]
  - [ORDER BY attribute(s)]

- Date Format, Strings and Numbers
  - Date Format: ‘YYYY-MM-DD‘, e.g. ‘1974-12-31‘
  - Strings: ‘String‘, e.g. ‘I like SQL’
  - Numbers: Number, e.g. 41 or 34.12

Note: (1) Dates and Strings have to be enclosed by two apostrophes.
(2) The conditions in […] are optional.

SELECT

SELECT * ← *All columns*

FROM Products ← *table 'Products'*

ORDER BY ID ← order result by column *'ID'*

ID	Product_name	Colour	Article_no	Sale_price	Purchase_price	Stock	Items_sold	City
1	Monitor 17'	White	1297812542	399.00	249.99	50	134	Frankfurt
2	Monitor 19'	black	2457897145	499.00	379.00	12	289	Berlin
3	Monitor 17'	black	1297467815	405.00	249.99	25	124	Frankfurt
4	Monitor 19'	white	2459871327	509.00	389.99	150	12	Frankfurt
5	Monitor 20'	black	2789441512	799.00	599.00	520	1052	Berlin
6	Monitor 20'	white	2799151424	829.00	549.99	100	26	Berlin
7	Monitor 20'	anthracite	2764657527	819.00	589.99	50	127	Nürnberg
8	Monitor 21'	anthracite	2845161215	999.00	799.99	100	279	Hamburg
9	Monitor 24'	white	2945712415	1299.00	945.00	25	124	Berlin
10	Monitor 24'	black	2955745742	1350.00	956.00	450	1024	Hamburg
…								

SELECT using WHERE

```
SELECT    *
FROM      Products
WHERE     Purchase_price > 500 AND City = 'Berlin'
```

ID	Product_name	Colour	Article_no	Sale_price	Purchase_price	Stock	Sold_items	City
5	Monitor 20'	black	2789441512	799.00	599.00	520	1052	Berlin
6	Monitor 20'	white	2799151424	829.00	549.99	100	26	Berlin
9	Monitor 24'	white	2945712415	1299.00	945.00	25	124	Berlin

SELECT using SUM and GROUP BY

```
SELECT      City, SUM(Stock)
FROM        Products
GROUP BY    City
```

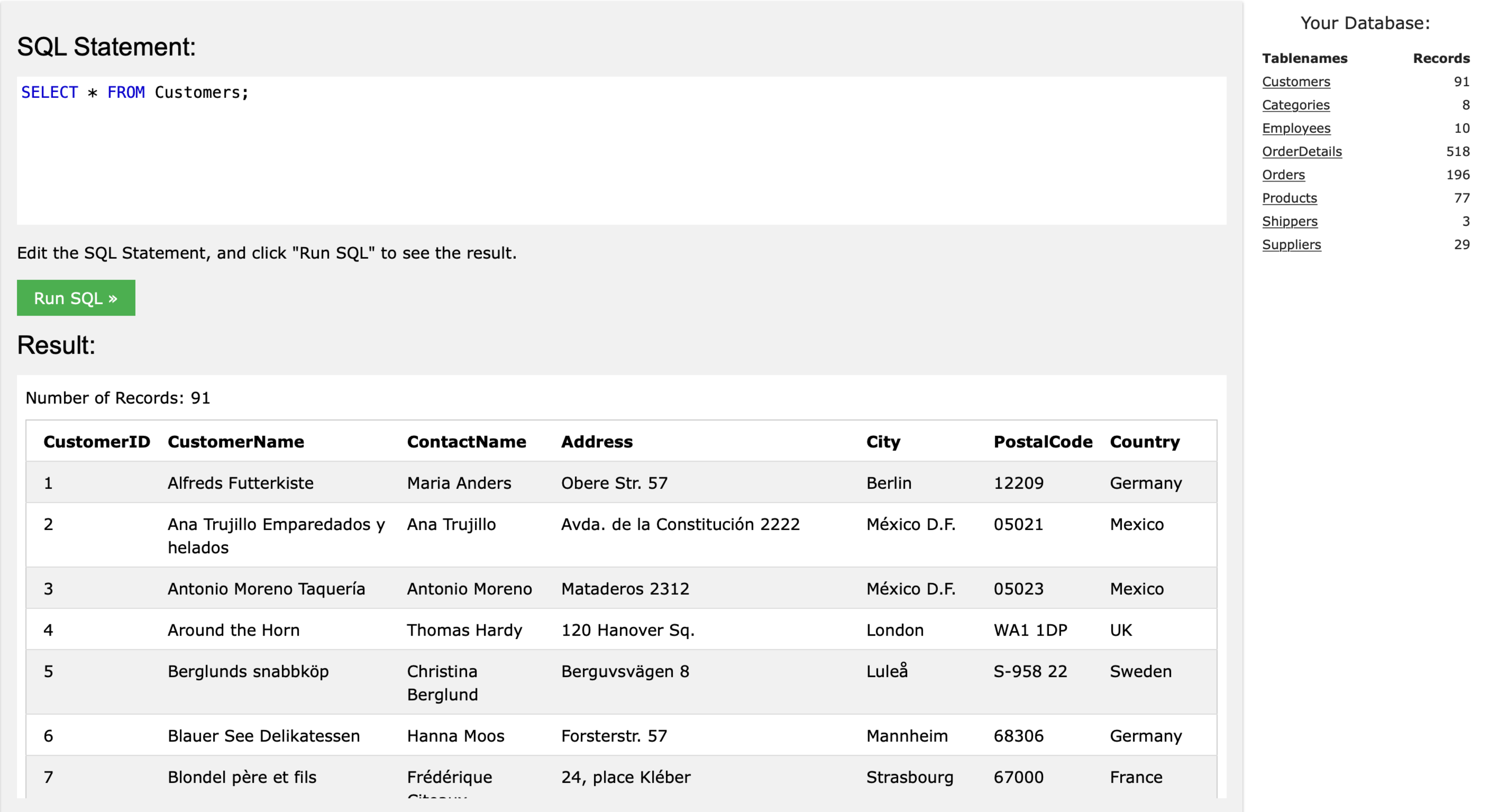
City	SUM(Stock)
Frankfurt	225
Berlin	657
Nürnberg	50
Hamburg	550
…	

Exercise 3: SQL

Exercise 3: SQL

- Please use the databases and environment provided by w3schools.com called Tryit Editor
- W3schools also provides several exercises and examples. Use them!

SQL W3Schools: Databases



SQL Statement:

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Result:

Number of Records: 91

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France

Your Database:

Tablenames	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

Link: https://www.w3schools.com/sql/trysql.asp?filename=trysql_op_in

Exercise 3a)

a) How many customers are stored in the table 'Customers'?

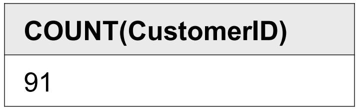
```
SELECT      COUNT(CustomerID)
FROM        Customers
```

COUNT(CustomerID)
91

Exercise 3a)

a) How many customers are stored in the table 'Customers'?

```
SELECT          COUNT(CustomerID)
FROM            Customers
```



COUNT(CustomerID)
91

Question from the exercise:
How many distinct cities are in the table „Customers“?

```
SELECT COUNT(DISTINCT City) AS individualCitys
FROM Customers
```

individualCity
69

Note: Not supported by MS Access database

Exercise 3b)

b) What is the average price over all Products?

```
SELECT      AVG(Price)
FROM        Products
```

AVG(Price)
28.866

Exercise 3c)

c) Display the Name and Price of the Products from most expensive to least expensive.

```
SELECT ProductName, Price
FROM Products
ORDER BY Price DESC
```

```
Or more complicated:
SELECT MAX(Price), ProductName
FROM Products
GROUP BY ProductName
ORDER BY MAX(Price)DESC
```

Price	ProductName
263.5	Côte de Blaye
123.79	Thüringer Rostbratwurst
97	Mishi Kobe Niku
81	Sir Rodney's Marmalade
62.5	Carnarvon Tigers
55	Raclette Courdavault
53	Manjimup Dried Apples
49.3	Tarte au sucre

Exercise 3d)

d) How many orders were made per day?

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

OrderDate	Orders
7/4/1996	1
7/5/1996	1
7/8/1996	2
7/9/1996	1
7/10/1996	1
7/11/1996	1
7/12/1996	1
7/15/1996	1
7/16/1996	....

Exercise 3e)

e) How many Customers are from Paris?

```
SELECT COUNT(City)
FROM Customers
WHERE City = "Paris"
```

Count
2

```
SELECT COUNT(City), City
FROM Customers
WHERE City = "Paris"
GROUP BY City
```

Count	City
2	Paris

Exercise 3f)

f) Display the names of all Customers from Germany

```
SELECT CustomerName, Country
FROM Customers
WHERE Country = "Germany"
```

CustomerName	Country
Alfreds Futterkiste	Germany
Blauer See Delikatessen	Germany
Drachenblut Delikatessend	Germany
Frankenversand	Germany
Königlich Essen	Germany
Lehmanns Marktstand	Germany
Morgenstern Gesundkost	Germany
Ottilies Käseladen	Germany
....	

Exercise 3g)

g) Display the Name and Adress of all customers, sorted descending

```
SELECT CustomerName, Address
FROM Customers
ORDER BY CustomerName DESC
```

CustomerName	Address
Wolski	ul. Filtrowa 68
Wilman Kala	Keskuskatu 45
White Clover Markets	305 - 14th Ave. S. Suite 3B
Wellington Importadora	Rua do Mercado, 12
Wartian Herkku	Torikatu 38
Vins et alcools Chevalier	59 rue de l'Abbaye
Victuailles en stock	2, rue du Commerce

Exercise 3h + i)

h) How many Customers are from Berlin?

```
SELECT COUNT(City)
FROM Customers
WHERE City = "Berlin"
```

Count
1

i) How many Customers are not from Berlin?

```
SELECT COUNT(City)
FROM Customers
WHERE City <> "Berlin"
```

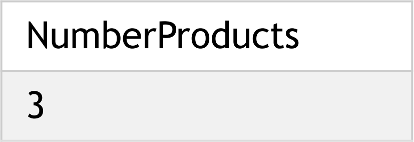
```
Or:
SELECT COUNT(City)
FROM Customers
WHERE NOT City = "Berlin"
```

Count
90

Exercise 3j)

j) How many Products cost more than 40 and have a CategoryID of less than 3?

```
SELECT COUNT(ProductID)
FROM Products
WHERE Price > 40 AND CategoryID < 3
```

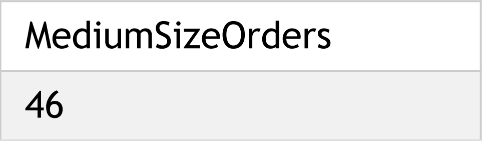


NumberProducts
3

Exercise 3k)

k) How many OrderDetailIDs had a Quantity of more than 5, but less than 10

```
SELECT COUNT(OrderDetailID) AS MediumSizeOrders
FROM OrderDetails
WHERE Quantity > 5 AND Quantity < 10
```



MediumSizeOrders
46

Exercise 3l)

l) Display ‘CategoryName', ‘ProductName' and the Price of all Products

```
SELECT Categories.CategoryName, Products.ProductName,
              Products.Price
FROM Products INNER JOIN Categories ON
              Products.CategoryID =
              Categories.CategoryID
```

CategoryName	ProductName	Price
Beverages	Chartreuse verte	18
Beverages	Chang	19
Beverages	Guaraná Fantástica	4.5
Beverages	Sasquatch Ale	14
Beverages	Steeleye Stout	18
Beverages	Chais	18
Beverages	Côte de Blaye	263.5
…..		

Exercise 3m)

m) Display all CustomerNames and OrderDates that have been made from Mexico

```
SELECT Orders.OrderDate, Customers.CustomerName,
               Customers.Country
FROM Orders INNER JOIN Customers ON Customers.CustomerID =
               Orders.CustomerID
WHERE Country = "Mexico"
```

OrderDate	CustomerName	Country
9/18/1996	Ana Trujillo Emparedados y helados	Mexico
11/27/1996	Antonio Moreno Taquería	Mexico
7/18/1996	Centro comercial Moctezuma	Mexico
11/14/1996	Pericles Comidas clásicas	Mexico
10/4/1996	Pericles Comidas clásicas	Mexico
9/12/1996	Tortuga Restaurante	Mexico
…		

Exercise 3n)

n) Insert a data record into the table “Orders” from the customer with the new CustomerID = 1 and display it.

```
INSERT INTO Orders(OrderID, CustomerID, EmployeeID, OrderDate,
ShipperID)
VALUES (12345, 1, 23243, “2020-01-01‟, 213442)
```

or

```
INSERT INTO Orders
VALUES (9999,1, 23243, “2020-01-01‟, 213442)
```

```
SELECT *
FROM Orders
WHERE CustomerID = 1
```

Important: Date enclosed by apostrophes

Exercise 3o)

o) Update the City of the customer with the CustomerID ‘1' to Frankfurt.

```
UPDATE Customers
SET City = "Frankfurt"
WHERE CustomerID = 1
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Frankfurt	12209	Germany

Exercise 3p)

p) Delete the entry inserted in Exercise n).

```
DELETE FROM     Orders
WHERE           OrderID = 10256
```

Exercise 3q)

q) Delete the whole table “Orders”.

```
DROP TABLE Orders
```

Exercise 3r)

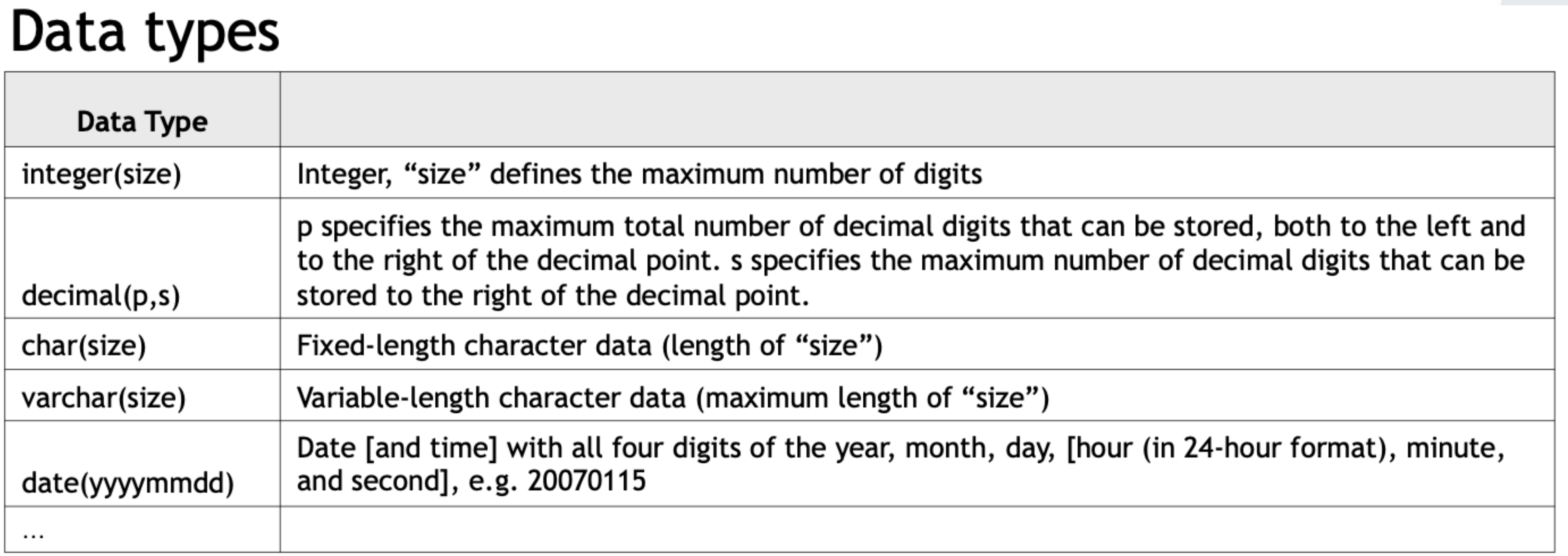
r) Delete the whole database.

```
DROP DATABASE name_of_Database
```

How to create a table?

- Beim Erstellen einer Tabelle muss deren eindeutiger Name, Primärer Schlüssel und die Liste der Attribute definiert werden

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

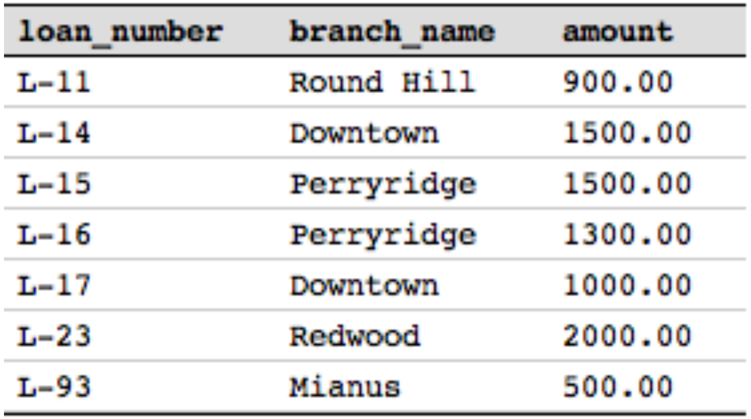


Data types

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
…	

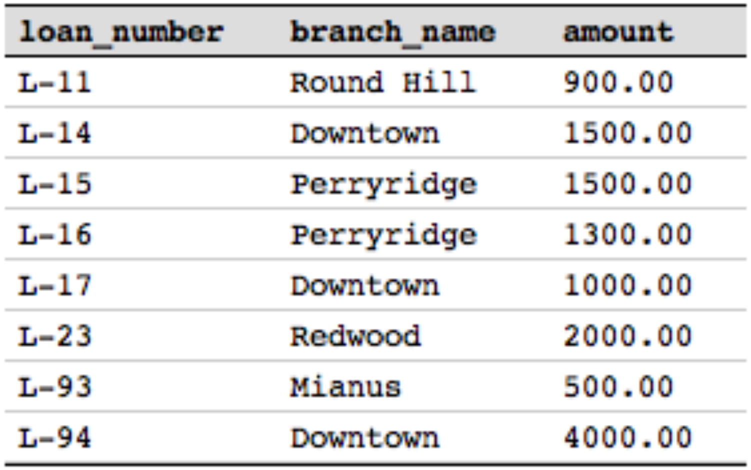
Exercise 3 f): SQL

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

→

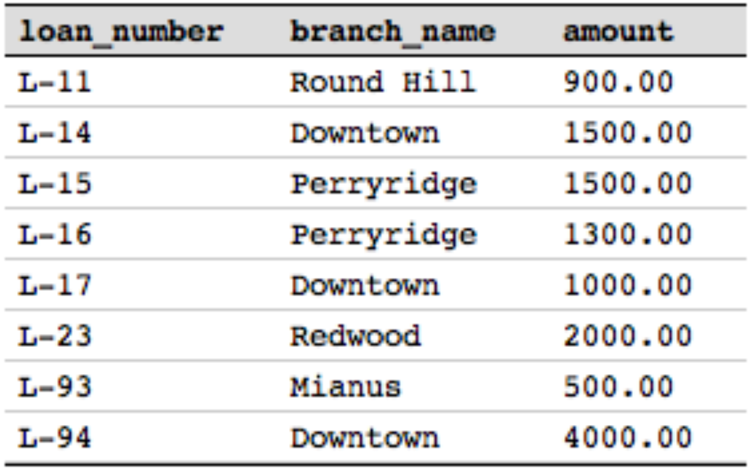


loan_number	branch_name	amount
L-11	Round Hill	900.00
L-14	Downtown	1500.00
L-15	Perryridge	1500.00
L-16	Perryridge	1300.00
L-17	Downtown	1000.00
L-23	Redwood	2000.00
L-93	Mianus	500.00
L-94	Downtown	4000.00

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

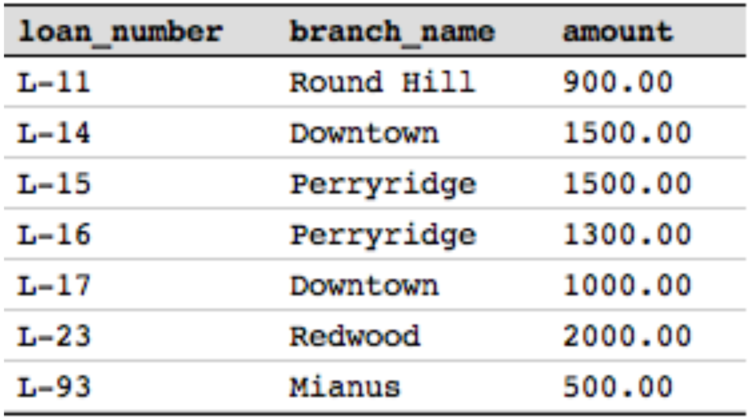
Exercise 3 g): SQL

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'
```

Delete vs Drop

```
DELETE FROM Orders
```

Data Manipulation Language: Löscht alle Einträge in der Tabelle Orders. Neue Einträge können mit INSERT INTO wieder hinzugefügt werden.

```
DROP TABLE Orders
```

Data Definition Language: Löscht die Tabelle Orders. Die Tabelle muss mit CREATE TABLE neu erstellt werden.

Open Questions?