Please fill in your name and registration number (Matrikelnr.) **immediately**.

EXAM ON	**SAMPLE SOLUTION**	09.12.2022
◯ DATENMODELLIERUNG (**184.685**)	◯ DATENBANKSYSTEME (**184.686**)	A

Matrikelnr.	Last Name	First Name

Duration: 80 minutes. Provide the solutions at the designated pages; solutions on additional sheets of paper are not graded. **Have a successful exam!**

Task	1	2	3	4	5	6	7	Σ
Max. Points	7	10	10	10	9	6	8	60
Solved								–
Points								

Please, *do not* remove the staple.

Question 1: (7)

a) For the relational schemas (R, F_1) and (R, F_2), where $R = SUVWXYZ$, find *all keys*.

(4 points)

Dependencies	Keys
$F_1 = \{UZ \to SX, UW \to SV, SV \to UZ, WZ \to XY\}$	WU, WVS ....
$F_2 = \{SV \to XZ, U \to YW, VZ \to SW\}$	UVZ, USV ....

b) Consider the relational schemas (R, F_1) and (R, F_2), where $R = ABCDEFG$. Determine their corresponding normal forms, and mark the right answers.

(3 points)

Dependencies	Keys
$F_1 = \{BDF \to CG, CF \to G, BC \to ADE, EF \to C\}$	BCF, BDF, BEF

neither 3NF nor BCNF ⊗ 3NF & not BCNF ◯
BCNF & not 3NF ◯ 3NF & BCNF ◯

Dependencies	Keys
$F_2 = \{AB \to CG, AB \to DE, F \to AB, C \to F\}$	C, F, AB

neither 3NF nor BCNF ◯ 3NF & not BCNF ◯
BCNF & not 3NF ◯ 3NF & BCNF ⊗

Attention: for each correct solution: 1.5 point, for each wrong solution: -1.5 point, unanswered questions give 0 points. In total you get at least 0 points.

MatrikelNr: | | **Name:** | |

Question 2: (10)

a) Consider the relational schema (R, F), where $R = ABCDEFG$, together with all keys are given below.

Using the (Relational) Synthesis Algorithm ("Synthesealgorithmus"), find a lossless and dependency preserving decomposition in 3NF (F is already in a canonical form/is a minimal cover).

For each schema R_i of the decomposition, state its attributes and mark exactly one key by underlining.

(6 points)

$$F = \{A \rightarrow D, BC \rightarrow E, E \rightarrow CGD, B \rightarrow A\}, \text{Keys} = \{BFC, BFE\}$$

Decomposition into 3NF (Underline one key in each relation)
R_1 <u>A</u>D .... R_2 <u>BC</u>E .... R_3 <u>E</u>CGD ...... R_4 <u>B</u>A .... R_5 <u>BFC</u> ....

b) Consider the relational schema $R = ABCDEF$ together with the functional dependencies (FDs) $F = \{ACE \rightarrow DF, BCD \rightarrow AF\}$ and keys $\{ABCE, BCDE\}$.

You are given the following subschemas R_i of R:

relational schema	<u>non-trivial</u> functional dependencies	keys
$R_1 = ABCE$	$C_1 = \emptyset$	$ABCE$
$R_2 = ACDEF$	$C_2 = \{ACE \rightarrow DF\}$	ACE
$R_3 = ABCDF$	$C_3 = \{BCD \rightarrow AF\}$	BCD

Determine for the following decompositions of (R, F) whether the decomposition is dependency preserving. If the decomposition is not dependency preserving, state at least one (non-trivial) functional dependency that was lost. In addition to this, you also need to answer whether the decomposition is lossless or not.

(4 points)

decomposition	dependency preserving		"lost" FDs	lossless	
(R_1, R_2)	◯ yes	⊗ no	$BCD \rightarrow AF$	⊗ yes	◯ no
(R_2, R_3)	⊗ yes	◯ no	....	◯ yes	⊗ no

Question 3: (10)

Assume that a bicycle seller organizes her data in the following database (primary keys are underlined):

Cycle(CycleID, Brand, Type)
Accessory(AccessoryID, Description)
Customer(CustomerID, Lastname, Givenname)
soldCycle(*CycleID: Cycle.CycleID, CustomerID Customer:CustomerID*, Date)
repaired(*CycleID: Cycle.CycleID*, Date)
soldAccessory(*AccessoryID: Accessory.AccessoryID, CustomerID Customer:CustomerID*, Date)

a) Consider the folloing query given in **relational algebra**.
Describe *briefly* (**1 short sentence!**) what values are returned by the query. (2 Points)

$$\pi_{\text{Type}}(\texttt{Cycle} \bowtie \sigma_{\text{Date}='2022'}(\texttt{soldCycle}))$$

```
Type of cycles, which have been sold in 2022.
```

b) Consider the folloing query given in **relational algebra**.
Describe *briefly* (**1 short sentence!**) what values are returned by the query. (1 Point)

$$(\texttt{Accessory} \bowtie \texttt{soldAccessory}) \div (\texttt{soldCycle})$$

```
Syntax-Error.  Division not according to definition.
Point also if student tried to explain division.
```

c) Given the following **database instance**:

Cycle		
CycleID	**Brand**	**Type**
1	Peugeot	Race
2	Giant	Race
3	Stevens	Trek
4	Allegro	Trek
5	Simplan	MTB
6	Gazelle	Omafiets
7	MEC	Trek
8	Cube	Pedelec
9	Bullit	Cargo

soldCycle		
CycleID	**CustomerID**	**Date**
3	1	2021-04-28
8	1	2021-07-28
6	2	2022-02-21
4	3	2021-03-16

Consider the following query in the **domain calculus**.
Determine the result of this query on the database instance given below. (3 Points)

$$\Big\{[\text{Brand}, \text{Type}] \;\Big|\; \exists \text{CycleID}\Big(\exists \text{Date}\Big(\exists \text{CustomerID}\ (\text{Date} \geq' 2021\text{-}07\text{-}28') \land [\text{CycleID}, \text{CustomerID}, \text{Date}] \in \texttt{soldCycle} \land [\text{CycleID}, \text{Brand}, \text{Type}] \in \texttt{Cycle}\Big)\Big)\Big\}$$

Brand	Type
Gazelle	Omafiets
Cube	Pedelec

d) Formulate a query in the **tuple calculus** that does the following:
Output name and givenname of customers, which have bought at least one bicycle and given the bike into repair. (4 Points)

$$\Big\{[c.lastname, c.givenname] \;\Big|\; \exists c \in \texttt{Customer}\Big(d \in \texttt{Cycle}\Big(\exists s \in \texttt{soldCycle}\Big(\exists r \in \texttt{reparied}\big(c.\texttt{CustomerID} = s.\texttt{CustomerID} \land s.\texttt{CycleID} = r.\texttt{CycleID} \land d.\texttt{CycleID} = r.\texttt{CycleID} \land s.\texttt{CycleID} = d.\texttt{CycleID}\big)\Big)\Big)\Big)\Big\}$$

Question 4: (10)

Given the following relational schema.

```
pet (id, name, age)
dog (id, dog_age)
cat (id, lives)
```

In addition, given the following database instance:

`dog`:

id	dog_age
0	70
1	42
2	70
3	21
4	35
5	7

`cat`:

id	lives
90	3
91	7
92	7
93	2
94	2
95	5

`pet`:

id	name	age
0	Luna	10
1	Rocky	6
2	Frida	10
3	Lucy	3
4	Rocky	5
5	Maya	1
90	Leo	8
91	Felix	4
92	Lucy	6
93	Balu	5
94	Nala	7
95	Simba	3

a) Evaluate the following SQL Query. (3 Points)

```sql
SELECT pet.id, pet.name, pet.age, dog.dog_age
FROM pet, dog
WHERE pet.id = dog.id  AND pet.age BETWEEN 3 AND 6
ORDER BY age;
```

Reminder:
The between operator is inclusive. Start und end-values are included, more precisely, ATTR BETWEEN X AND Y is the same as ATTR >= X AND attr <= Y.

Result of the query:

id	name	age	dog_age
3	Lucy	3	21
4	Rocky	5	35
1	Rocky	6	42

b) Provide an SQL query for the following task: list all names and age of cats, where lives is supposed to be between 2 and 5 and only cats whose name starts with "Ba" are of interest. (3 Points)

```
SELECT name, age, lives
FROM pet, cat
WHERE pet.id=cat.id AND  cat.lives BETWEEN 2 and 5 AND pet.name like 'Ba%';
```

c) Provide an SQL query for the following task. List all names of pets, whose name has been used for both cats as well as dogs. In addition, output how often that name has been used for cats. Sort the values by how often it has been used in descending order, then by name in ascending order. (4 Points)

Example:

```
SELECT name, count(name)
FROM pet INNER JOIN cat  on pet.id = cat.id
WHERE name IN
       (SELECT name
        FROM pet INNER JOIN dog  on pet.id = dog.id)
GROUP by name
 order by count DESC
```

Question 5: (9)

You are asked to design a database that concerns rail traffic, with a focus on passengers, private rail companies and the sale of tickets.

Create an EER-diagram based on the information described below. Use the (min,max) notation, and in case no explicit information is given, assume that there are no restrictions on the values for (min,max). The model shall work without using `NULL`-values, redundancies shall be avoided, and it is not allowed to introduce any attributes not described by the text. Finally, make sure that a key is defined for each entity type.

All passengers are uniquely identified by a combination of their name and customer number (customerNR).

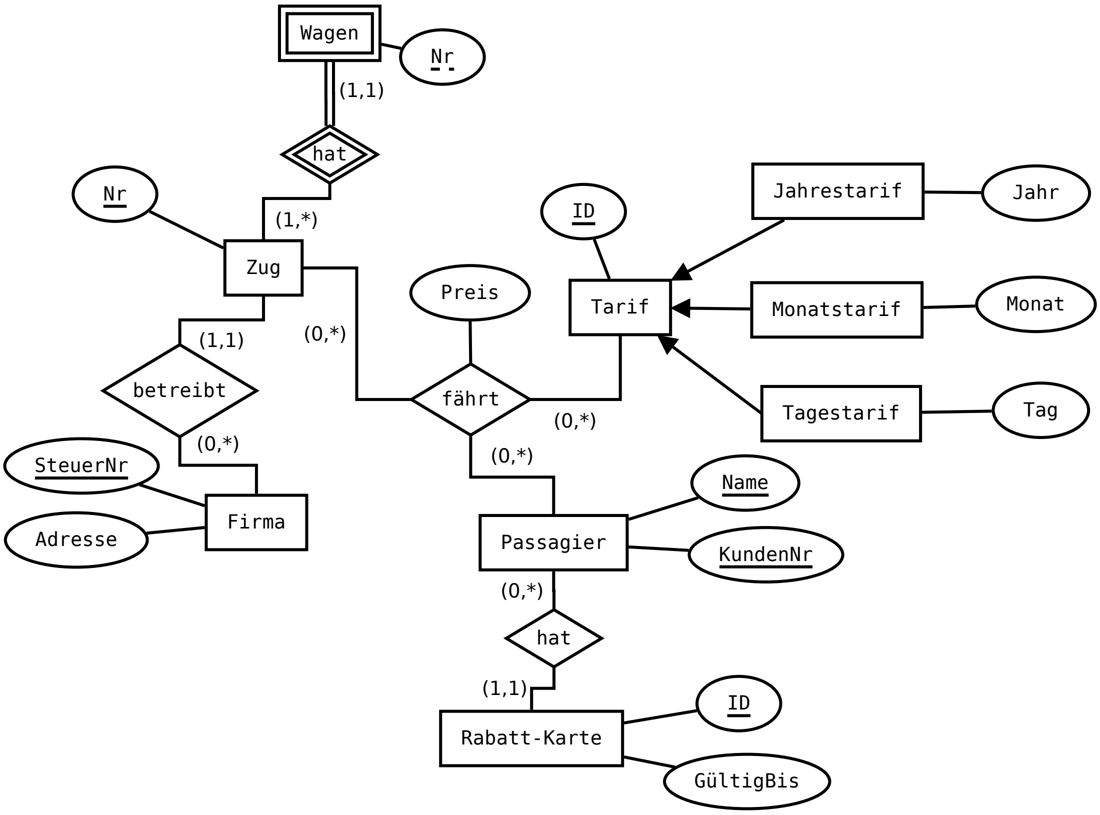
All trains have a unique number. There are also several carriages for each train. Each carriage is uniquely identified by the train number and its own carriage number. Each train must have at least one carriage.

Tariffs, or **fares**, are also modelled in this database. Each fare has a unique ID. A distinction is also made between annual fares, monthly fares and daily fares. Annual fares also have an attribute for the specific year. Monthly fares have an attribute for the month they are issued for. Finally, there is the additional 'day' attribute for daily fares.

The relation "rides" between train, passenger and fare expresses which people take which trains at which fares. Each entry of the rides relation also has an attribute "price", which holds the actual cost of the ticket.

Companies are clearly identified by their tax ID. Their address will also be recorded. Each company can operate any number of trains. On the other hand, each train must be clearly assigned to one company, which operates it.

Finally, discount cards are also to be modelled in the database. Each discount card is uniquely identified by an ID. In addition, it is also recorded as an attribute until when the discount card is valid (validThrough). Each discount card is uniquely assigned to a passenger.

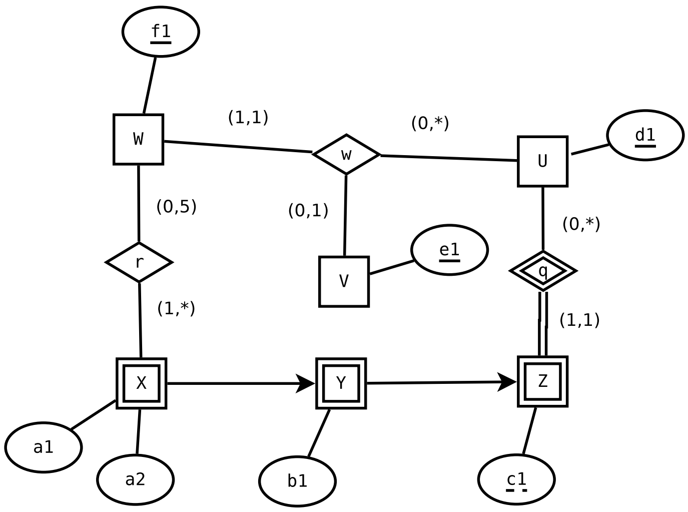


Breakdown of points:

- min/max part: 4 points
- Entities & relations: 4 points
  (and understanding keys)
- Attributes correct: 1 point

Question 6: (6)

Construct a relational schema according to the EER-diagram given below. For each relation, clearly mark the primary key by underlining the corresponding attributes. Mark foreign keys (FK) either by prefixing the name of the relation referenced by the FK (i.e., by `Relation.Attribute`) or by using the notation `NameOfAttribute:Relation.Attribute` (where `NameOfAttribute` is the name of the attribute in the current schema and `Relation.Attribute` describes the value that is referenced by the FK). You do not need to distinguish between FKs consisting of a single attribute and FKs combining several attributes. Create as few relations as possible without introducing any redundancies. Note that the database does not allow `NULL`-values.



X (d1, c1, a1, a2 ....)

Y (d1, c1, b1 ....)

Z (d1, c1 ....)

U (d1 ....)

V (e1 ....)

W (f1, *e1: V.e1, d1: U.d1* ....)

r (*f1: W.f1, d1: X.d1, c1: X.c1* ....)

(....)

MatrikelNr: **Name:**

Question 7: (8)

Consider the relational schemas $X(\underline{A}BC)$, $Y(\underline{D}E)$, and $Z(\underline{A}CE)$. Assume there exists an instance of X containing 3 tuples, an instance of Y containing 2 tuples, and an instance of Z containing 3 tuples. Thus

$$X(\underline{A}BC): 3 \qquad Y(\underline{D}E): 2 \qquad Z(\underline{A}CE): 3$$

Consider the expressions in Relational Algebra given below. For these expressions, provide the minimal and maximal possible size (= number of tuples) of their results over instances for X, Y, and Z of the given sizes. In addition, provide concrete instances over which the expressions actually realize these bounds, i.e. return results of minimal/maximal size. Make sure that the provided instances contain exactly the given number of tuples.

Attention: Points for correct instances are awarded only if the stated corresponding size is also correct!

a) **Expression:** $\rho_{E\leftarrow D}(\pi_{A,D}(X \rtimes_{X.A=Y.E} Y)) - \pi_{A,E}(Y \bowtie \rho_{A\leftarrow E} Y)$ (4 Points)

min. size of the result: 1 ....

X		
<u>A</u>	B	C
1	–	–
2	–	–
3	–	–

Y	
<u>D</u>	E
1	1
2	2

max. size of the result: 4 ....

X		
<u>A</u>	B	C
1	–	–
2	–	–
3	–	–

Y	
<u>D</u>	E
3	1
4	1

b) **Expression:** $\pi_{A,D,E}(Y \rtimes Z) \cup \pi_{A,D,E}(Y \ltimes Z)$ (4 Points)

min. size of the result: 3 ....

Y	
<u>D</u>	E
1	4
2	5

Z		
<u>A</u>	C	E
1	–	4
2	–	5
3	–	6

max. size of the result: 6 ....

Y	
<u>D</u>	E
1	4
2	4

Z		
<u>A</u>	C	E
1	–	4
2	–	4
3	–	4

`Overall: 60 points`

Have a successful exam!