Exercise sheet 1 (WS 2020) – Sample solution

3.0 VU Data Modelling / 6.0 VU Database Systems

About the exercises

General information

In this part of the exercises you are asked to create a small database using EER-diagrams, transform EER-diagrams into a relational schema and make yourself familiar with relational algebra and relational calculus.

We recommend you to solve the problems **on your own** (it is a good preparation for the exam – and also for your possible future job – to carry out the tasks autonomously). Please note that if we detect duplicate submissions or any plagiarism, both the "original" and the "copy" will be awarded 0 points.

Your submission must consist of a single, typeset PDF document (max. 5MB). **We do not accept PDF files with handwritten solutions.**

In total there are 8 tasks and at most 15 points that can be acchieved on the entire sheet.

Deadlines

until 05.11.	**12:00pm**	Upload your solutions to TUWEL
18.11.	13:00pm	Evaluation and feedback is provided in TUWEL

Further questions – TUWEL forum

If you have any further questions concerning the contents or organization, do not hestitate to ask them on TUWEL forum. **Under no circumstances should you post (partial) solutions on the forum!**

Changes due to COVID-19

Due to the ongoing situation with COVID-19 we will not offer in-person office hours for the exercise sheets. If you have technical issues, trouble understanding the tasks on this sheet, or other questions please use the TUWEL forum.

We also recommend that you get involved in the forum and actively discuss with your colleagues on the forum. From experience we believe that this helps all parties in the discussion greatly to improve their understanding of the material.

Exercise: EER-diagrams

Exercise 1 (Creating an EER-diagram) [3 points]

You are working in a startup that will revolutionize book marketing. After securing many millions in fuding, you decide to actually work on the project. You begin by designing the database on the basis of your current notes.

Draw an EER-diagram based on the available information (see next page). Use the notation presented in the lecture and the (min, max) notation. NULL values are not allowed and redundancies should be avoided. Sometimes it might be necessary to introduce additional keys.

A possible software for creating the EER-diagram is DIA (`http://wiki.gnome.org/Apps/Dia`, binaries at `http://dia-installer.de`; Attention: select ER in the diagram editor!). Of course you are also allowed to create the diagram with any other suitable software.

Description of the issues:

The crucial function of the database is to record books. For every book a title (TITLE) and a number of pages (PAGES) is stores, where the title identifies a book uniquely. Furthermore, for every book we save whether it specifically is a non-fiction, poetry, or a novel. Note that a book does not necessarily have to fall into one of these categories. For a novel we additionally store its genre (GENRE). Non-fiction books can be written about novels, but every non-fiction book can only be written about at most three different novels. It is also possible that a non-fiction book is not written about any novel. Books are written by authors. An author always has written at least one book. An author is uniquely identified by the combination of their name (NAME) and date of birth (DOB).

There can be multiple editions of each book. Different editions of a book can be distinguished by their edition number (EDNR). However, the edition number is not necessarily unique among different books. We also store a year (YEAR) value for each edition. An edition has at least one printing. The printings of an edition can be identified by their print number (PNR), which is again not necessarily unique over different editions. For every printing we record who does the printing (PRINTER). Additionally, some editions are special editions for which we also store a reason (REASON).

Editions are published by a publisher. A publisher has a name (NAME) and a budget (BUDGET) associated with them, where the name uniquely identifies a publisher. Every publisher has departments which are identified inside the publisher by a unique combination of area of responsibility (AREA) and their location (CITY). Note that the same area/location combination can occur in multiple publishers and that there are no departments without a publisher.

A publisher advertises, on a given date (DATE), to one or more target groups via up to two marketing channels. A marketing channels has a name (CHNAME) – which uniquely identifies the channel – and associated costs (COST). A marketing channel can also be a social network or a newspaper. For newspapers we additionally store the circulation (CIRC). For social networks the concrete platform (PLATFORM) is recorded. For target groups we record a description (DESC) as well as a age group (AGE), where the description is unique to the target group.

Target groups use social networks and are interested in at least one book. Every target group loves up to 5 authors. Regrettably, every target group also hates at least 1 (and arbitrarily many) authors. This motivates authors to use pseudonyms. Hence, we also record which author is an alias of which other author.

Solution: See Figure 1.

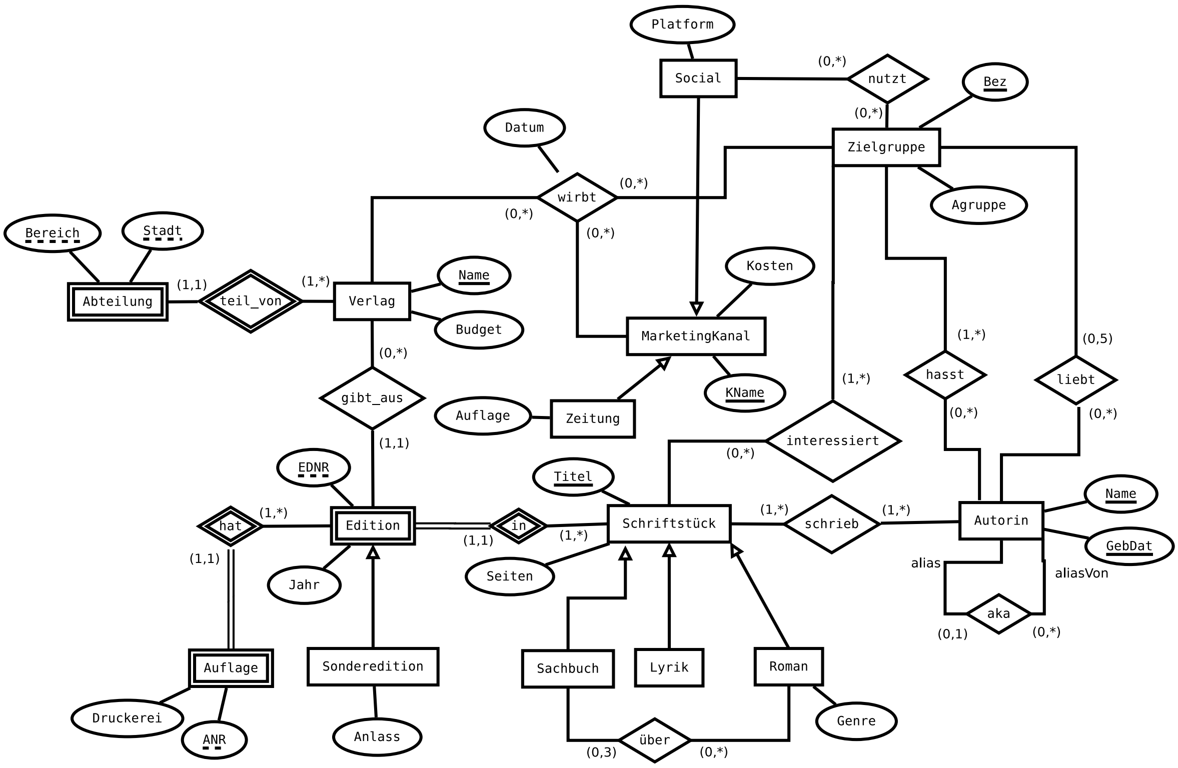

Figure 1: Solution for Exercise 1

Note: For the multi-entity relationship between publishers, target groups and marketin channels is not possible (with (min,max) Notation) to model the full details of the exercise description. As some colleagues pointed out in forum discussions, for a variety of reasons (for example the date), we can not really give any bounds for any of the entities. This kind of situation is common in all kinds of formal modelling. It is therefore important to always be aware of the limits of any formal modelling language.

In some cases, the exercise description does not clearly state whether $(0, \star)$ or $(1, \star)$ is appropriate. In these cases you are supposed to make an educated guess but both choices are acceptable when no further information is given. In a practical context it is however wise to not make assumptions but rather make an effort to obtain full information.

Exercise 2 (Semantics of EER diagrams) [1 point]

Consider the EER-diagram shown in Figure 2.

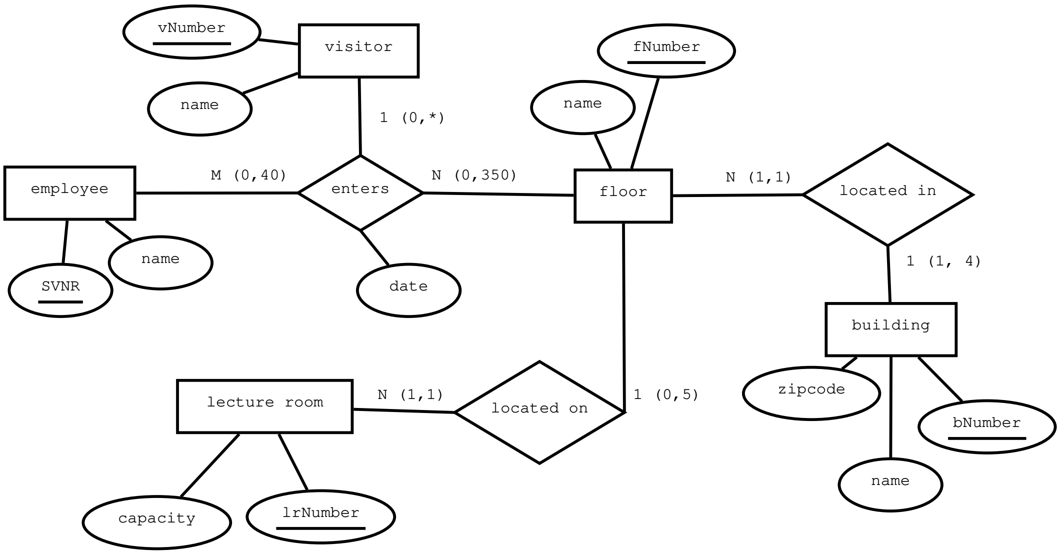

Figure 2: EER-diagram for exercise 2

In the ER diagram, both the notation by means of functionalities, as well as the (min, max) notation is used.
(*note:* this is not done in practice.)
Therefore, the diagram contains more information compared to the use of only one notation.

- Specify a specific relationship in the diagram where omitting one of the two notations causes a loss of information.
- For the chosen relationship type explain which notation, when omitted, leads to the loss of information.
- Explain briefly in your own words which information can no longer be represented.
- Provide a concrete example of the lost information, i.e. for the type of relationship you have chosen, specify an instance that violates (at least) one condition expressed by the omitted notation, but satisfies all requirements by the remaining notation.

Solution:
Yes, both notations contain information which cannot be expressed using the respective other notation. We will give one example for each notation below:

- In case we do not consider the functionalities for the relationship type `enters`, then it cannot be expressed using only the (min,max)-notation that only one visitor per employee and floor can enter the floor. A concrete example would be that the employee with SVNR 2000 enters the floor with vNumber 1 and with visitor with vNumber 2 as well as with the visitor with VNumber 3. This is not possible if we consider the cardinalities as well.

- In case we omit the *(min,max)-notation* in the relationship type `located on`, it cannot be expressed using only the functionalities that on a specific floor there cannot be more than 5 lecture rooms.

Exercise 3 (Construct a relational schema) [2 points]

Construct a relation schema according to the EER-diagram given in Figure 3. `NULL` values are not allowed (you can assume that all attributes specified for an entity type exist for all entities of this type, i.e., the definedness of all attributes is 100%). Create as few relations as possible without introducing any redundancies. For each relation clearly mark the primary keys by underlining the corresponding attributes and display foreign keys in italics.

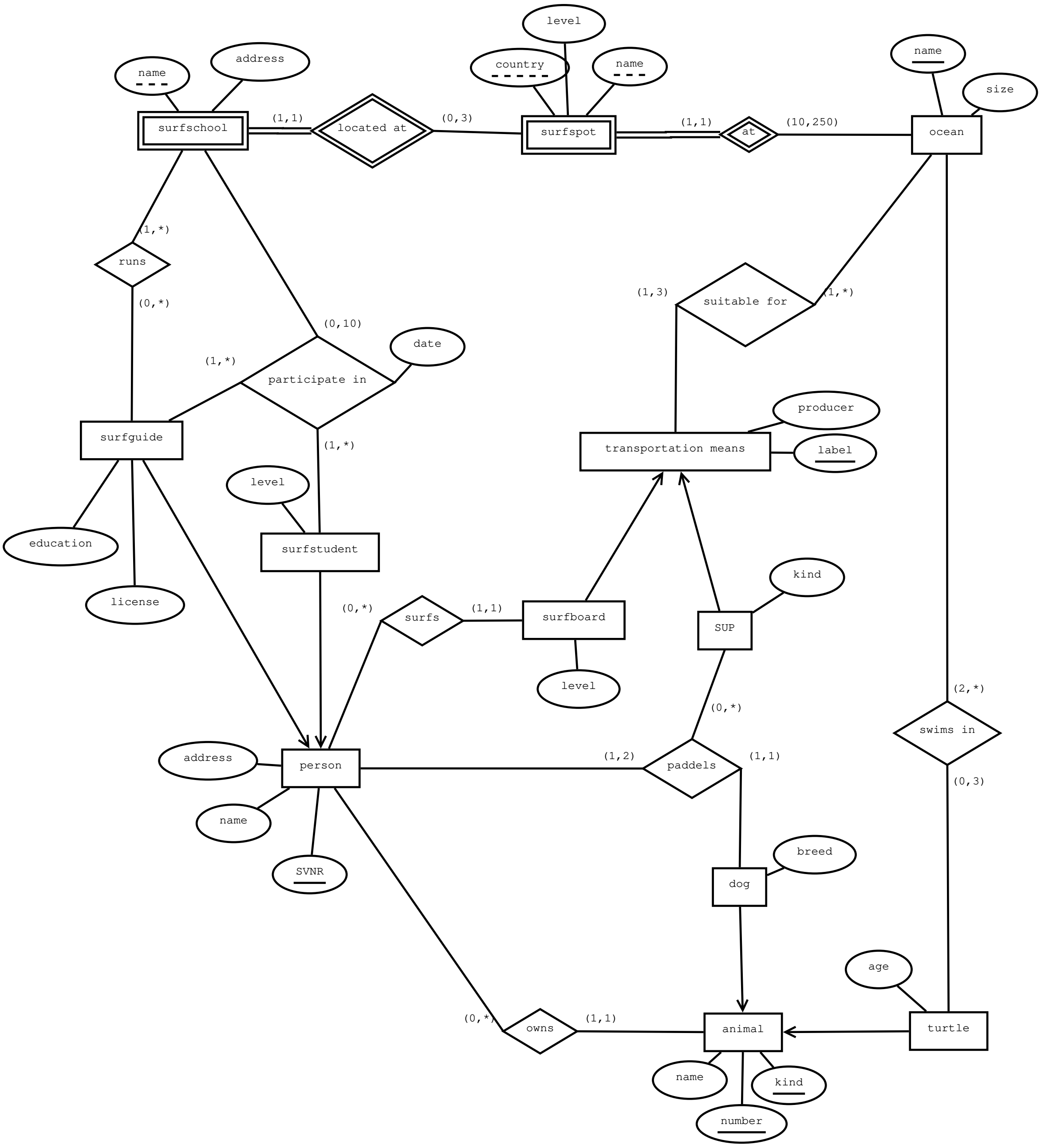

Figure 3: EER-diagram for exercise 3

Solution:

ocean	(name, size)
surfspot	(ocean_name: *ocean.name*, country, name, Level)
surfschool	(ocean_name: *surfspot.ocean_name*, spot_country: *surfspot.country*, spot_name: *surfspot.name*, school_name, address)
person	(SVNR, address, pName)
surfstudent	(SVNR:*person.SVNR*, level)
surfguide	(SVNR:*person.SVNR*, education, license)
animal	(number, kind, name, owner: *person.SVNR*)
dog	(number: *animal.number*, kind: *animal.kind*, breed, paddlesWith: *person.SVNR*, paddles: *SUP.label*)
turtle	(number: *animal.number*, kind: *animal.kind*, age)
transportation_means	(label, producer)
SUP	(label: *transportation_means.label*, kind)
surfboard	(label: *transportation_means.label*, level, surfedBy: *person.SVNR*)
suitable_for	(ocean_name: *ocean.name*, trans_label: *transportation_means.label*)
swims_in	(ocean_name: *ocean.name*, tur_number: *turtle.number*, tur_kind: *turtle.kind*)
runs	(ocean_name: *surfschool.ocean_name*, spot_country: *surfschool.spot_co* spot_name: *surfschool.spot_name*, school_name: *surfschool.school._na* SVNR:*surfguide.SVNR*)
participate_in	(ocean_name: *surfschool.ocean_name*, spot_country: *surfschool.spot_co* spot_name: *surfschule.spot_name*, school_name: *surfschool.school_nam* guide:*surfguide.SVNR*, student:*surfstudent.SVNR*, date)

Exercises: Relational Algebra - Relational Calculus

To help with typesetting the solutions to the following exercises, we compiled a list of the most important symbols for the Relational Algebra at `http://dbai.tuwien.ac.at/education/dm/resources/symbols.html`. You can copy and paste them into your Word/ LibreOffice/ OpenOffice/... document. In addition, the corresponding LaTeX commands are listed as well.

Exercise 4 (Evaluation) [0.5 points]

Consider the four relations below.

Band		
name	**genre**	**founded**
Cid Rim	TBD	2010
JSBL	Funk	2008
CSH	Indie	2010
Dorian Concept	Electronic	2006

Musician		
name	**instrument**	**studio**
Will	Guitar	False
Peter	Bass	True
Clemens	Drums	True
Dorian	Keyboard	True

partOf	
mname	**bname**
Clemens	JSBL
Clemens	Cid Rim
Will	CSH
Dorian	JSBL
Dorian	Dorian Concept

Song			
title	**by**	**length**	**genre**
Killer Whales	CSH	6:14	Indie
My Boy	CSH	2:52	Indie
Swnerve	Cid Rim	4:26	Electronic
Failures III	Dorian Concept	3:40	Jazz

Provide the results of the following queries over these relations.

(a)

$$\pi_{\text{title,genre}}\Big(\pi_{\text{title,length}}(\texttt{Song}) \times \big((\rho_{\text{mname}\leftarrow\text{name}}(\sigma_{\text{studio}=\textbf{True}}(\texttt{Musician})) \bowtie \texttt{partOf}) \ltimes \rho_{\text{bname}\leftarrow\text{name}}(\sigma_{founded\geq 2010}(\texttt{Band}))\big) \Big)$$

Solution:

q	
title	**genre**
Killer Whales	TBD
My Boy	TBD
Swnerve	TBD
Failures III	TBD

(b)

$$\begin{aligned}\{m.name, b.name \mid\ & m \in \texttt{Musician} \land b \in \texttt{Band} \land m.\text{instrument} \neq \text{'Bass'} \land \\ & \exists p \in \texttt{partOf}(\\ & b.\text{name} = p.\text{bname} \land m.\text{name} = p.\text{mname} \land \\ & (\forall s \in \texttt{Songs}(b.\text{name} = s.\text{by} \rightarrow s.\text{genre} = b.\text{genre})))\}\end{aligned}$$

Solution:

q	
m.name	**b.name**
Clemens	JSBL
Dorian	JSBL
Will	CSH

Exercise 5 (Equivalences) [2 points]

Consider the following relation schemata $R(\underline{A}BC)$, $S(B\underline{D}E)$, $T(C\underline{D}F)$ and the pairs q_i, q_j of expressions in relational algebra below. For every pair of expressions:

- Verify, whether the two expressions are equivalent (i.e. whether they produce the same result on all possible instances of the relation schemata). You may assume that no `NULL`-values occur in any instance of the schemata.
- Justify your answer with a brief **explanation**.
- In case the two expressions are *not* equivalent, additionally provide a **counterexample**. (A counterexample consists of the concrete instances of the affected relation schemata and the results of both expressions over these instances.)
  In case one of the expressions is not a valid expression in relational algebra you do not have to provide a counterexample, hence in this case an explanation suffices.

(a) $q_1\colon (\pi_C(T) - \sigma_{C>5}(R)) \cap \rho_{C\leftarrow B}(\pi_B(S))$ and
$q_2\colon \big(\sigma_{C>5}(\pi_C(T) - \pi_C(R))\big) \cap \rho_{C\leftarrow B}(\pi_B(S))$

(b) $q_3\colon S \bowtie (\pi_{CD}(T) \cap \rho_{D\leftarrow A}(\pi_{CA}(R)))$ and
$q_4\colon (S \bowtie \pi_{CD}(T)) \cap (S \bowtie \rho_{D\leftarrow A}(\pi_{CA}(R)))$

(c) $q_5\colon \pi_{AB}(\sigma_{A>B\vee A<B}(R)) \times \pi_{DF}(\sigma_{D>F\vee D<F}(T))$ and
$q_6\colon \pi_{ABDF}(\sigma_{(A>B\wedge D>F)\vee(A<B\wedge D<F)}(R \times T))$

Solution:

Ex (a) **No**, q_1 and q_2 **are not** equivalent.

q_2 is a valid expression in relational algebra, but q_1 is not. The reason is that the schemata of $\pi_C(T)$ (contains only the attribute C) and $\sigma_{C>5}(R)$ (contains all attributes from R: A, B, C) are different, which is not allowed for set difference.

Ex (b) **Yes**, q_3 and q_4 **are** equivalent.

This is based on the equivalence: $S \bowtie (T \cap R) \equiv ((S \bowtie T) \cap (S \bowtie R))$.

Ex (c) **No**, q_5 and q_6 **are not** equivalent.

In q_5 all tuples for which $A \neq B$ and $D \neq F$ are contained. In q_6 however only tuples of the form $A < B$, $D < F$ and $A > B$, $D > F$ are allowed in the result.

counterexample

R		
A	B	C
2	1	5
1	2	5

T		
C	D	F
0	5	1
7	3	8

q_5			
A	B	D	F
2	1	5	1
2	1	3	8
1	2	5	1
1	2	3	8

q_6			
A	B	D	F
2	1	5	1
1	2	3	8

Exercise 6 (Answer Sizes) [1.5 points]

Consider the relational schemas $R(\underline{A}B)$, $S(AB\underline{CD})$, and $T(AC\underline{E})$ and an instance of every schema, where there are $|R|$ tuples in the instance of R, $|S|$ tuples in the instance of S, and $|T|$ tuples in the instance of T.

- Provide the minimal and maximal size (= number of tuples) of the following expressions in Relational Algebra for the given values of $|R|$, $|S|$, $|T|$.
- Justify your answer.
- For both, the smallest and biggest possible answer size, provide concrete instances of the schemas (with R, S, and T having $|R|$, $|S|$, and $|T|$ tuples, respectively) over which the query returns an answer with the minimal/maximal number of tuples.

(a) $q_1\colon\ \rho_{B\leftarrow C}(T) ⟕ \sigma_{A=4\wedge B=2}(R)$ (with $|R| = 5$ and $|T| = 4$)

(b) $q_2\colon\ \pi_{EX}(\rho_{X\leftarrow A,Y\leftarrow C}(S) \times T) - \pi_{EX}(\rho_{X\leftarrow A}(T) \ltimes S)$ (with $|S| = 7$ and $|T| = 4$)

(c) $q_3\colon\ (\pi_A(\sigma_{A\neq 1}(R)) \cap \pi_A(\sigma_{A>3}(T))) \cup \rho_{A\leftarrow C}(\pi_C(T) \bowtie \pi_C(S))$
(with $|R| = 5$, $|S| = 3$ and $|T| = 3$)

Solution:
Aufgabe (a) [Minimum: 4 | Maximum: 4]

Maximum = Minimum: 4

R	
A	B
4	2
3	1
6	9
1	2
3	11

T		
A	C	E
4	2	1
4	2	2
4	2	3
4	2	4

Ergebnis		
A	B	E
4	2	1
4	2	2
4	2	3
4	2	4

The crucial observation to make here is that, in general, $T ⟕ R$ preserves all tuples of T. That is, the resulting relation can never have less tupes than T itself. The renaming on the side of T has no effect on this property. The selection over R can only make R smaller. Indeed, it is easy to see that all tuples of R can be removed by the selection, yielding the minimum answer size 4.

For the maximum, consider that the join is on the attributes A and B. Because of the selection on R, there can be at most one pair of values for (A, B) left: $(4, 2)$. Hence, the join can not join a tuple from T with more than one tuple of R (after selection). The only possibilities that remain are joining with 0 or 1 tuple from R. Since we consider a left outer join, both cases lead to exactly 4 tuples overall in the answer.

Aufgabe (b) [Minimum: 0 | Maximum: 28] For the maximum, observe that all 28 tuples from the cross product can remain after the projection to EX if all the values in column A of S are different. If we then have an instance (like in the example below), where the semi-join is empty, the difference can not remove any of these 28 tuples.

For the minimum, we can take the same considerations but apply them in the opposite direction. In the example, all values in column A of S are the same, which collapses the result of the cross-product after the projection to EX to only 4 tuples: $\{(1,1),(2,1),(3,1),(4,1)\}$. In the semi-join we keep all tuples of T by having all values of the C columns the same in both relations. With T as in the example, the difference then deletes all tuples from the left side of the expression.

Minimum: 0

S

A	B	C	D
1	9	2	1
1	9	2	2
1	9	2	3
1	9	2	4
1	9	2	5
1	9	2	6
1	9	2	7

T

A	C	E
1	2	1
1	2	2
1	2	3
1	2	4

Ergebnis

E	X

Maximum: 28

S

A	B	C	D
1	9	2	1
2	9	2	2
3	9	2	3
4	9	2	4
5	9	2	5
6	9	2	6
7	9	2	7

T

A	C	E
1	5	1
1	5	2
1	5	3
1	5	4

Aufgabe (c) [Minimum: 0 | Maximum: 6] For the minimum it is easy to use the selection predicates to construct an instance where both terms of the union are empty. One only needs to be careful about the key constraints. Note that the joim $\pi_C(T) \bowtie \pi_C(S)$ is semantically equivalent to an intersection, since both operands consist of only a single attribute with the same name.

For the maximum we can observe that neither term of the union can be larger than $|T|$ (because of the semantics of set intersection). Furthermore, it is easy to see that both terms are independent of each other with respect to T and one can therefore easily construct a result of size $2|T|$. Since both individual terms cannot be larger than $|T|$, this is clearly the maximum.

Minimum: 0

R	
A	B
1	1
2	2
3	3
4	4
5	5

S			
A	**B**	**C**	**D**
0	0	5	0
0	0	6	0
0	0	7	0

T		
A	**C**	**E**
1	1	1
2	2	2
3	3	2

Maximum: 6

R	
A	B
4	1
5	2
6	3
7	4
8	5

S			
A	**B**	**C**	**D**
0	0	1	0
0	0	2	0
0	0	3	0

T		
A	**C**	**E**
6	1	1
7	2	2
8	3	4

Ergebnis A: $\{1, 2, 3, 6, 7, 8\}$

Exercise 7 (Query Languages) [1 point]

Consider the relational schema $R(\underline{A}B)$, $S(AB\underline{CD})$, and $T(AC\underline{E})$.

In the following exercises you are given a query in one of the query languages from the lecture. Your task is to translate the query into the two other query languages that were discussed in the lecture.

(a) Translate the query

$$\pi_{A,C}(S \rtimes T)$$

into tuple relational and domain relational calculus.

Solution:
domain relational calculus:

$$\{a, c \mid \exists e([a, c, e] \in T \wedge \exists b, d([a, b, c, d] \in S))\}$$

tuple relational calculus:

$$\{t.A, t.C \mid t \in T \wedge \exists s(s \in S \wedge s.A = t.A \wedge t.C = s.C)\}$$

(b) Translate the query

$$\{x \mid \exists y, z([x, y] \in R \wedge [y, x, z, 3] \in S)\}$$

into tuple relational calculus and relational algebra.

Solution:
relational algebra:

$$\pi_{R.A}(R \bowtie_{R.A=S.B \wedge R.B=S.A} \sigma_{D=3}(S))$$

tuple relational algebra:

$$\{r.A \mid \ r \in R \land \exists s(s \in S \land r.A = s.B \land r.B = s.A \land s.D = 3)\}$$

(c) Translate the query

$$\{u, v \mid u \in R \land v \in T \ \land \exists w \in S(w.C > v.C)\}$$

into domain relational calculus and relational algebra.

Solution:
relational algebra:

$$R \times \pi_{A,C,E}(\sigma_{X>C}(T \times \rho_{X\leftarrow C}(\pi_C(S))))$$

domain relational calculus:

$$\{(a1, b, a2, c, e) \mid [a1, b] \in R \land [a2, c, e] \in T \ \land \exists as, bs, cs, ds([as, bs, cs, ds] \in S \land cs > c)\}$$

Exercise 8 (Formalizing Queries) [4 points]

A company manages information about the entries of its employees in a data base with the following schema (primary keys are underlined, foreign keys are written in *italics*).

```
employee     (persNr, SVNR, eName)
floor        (fName, name)
enters       (employeeNr: employee.persNr, floorName: floor.fName, date)
leaves       (employeeNr: employee.persNr, floorName: floor.fName, date)
building     (bName, zipcode)
located_in   (fName: floor.name, bName: building.bName)
lecture_room(lrNumber, name)
located_on   (lrNumber: lecture_room.lrNumber, floorName: floor.fName)
```

(In the following you may use suitable (unique) abbreviations for relations and table names.)

Express the queries described below in **relational algebra**, the **tuple relational calculus** and the **domain relational calculus**.

(a) Find the names, the personal number and the social security number (SVNR) of all employees, that are currently on the floor with name 'H1'.

Solution:
relational algebra: (attention: to be precise we have to rename all the attributes here, in order to perform a join.)

$$\pi_{\text{eName,persNr,SVNR}}(\sigma_{\text{StockName='H1'}}((\texttt{enters} - \texttt{ending}) \bowtie \texttt{employee}))$$

ending is defined as:

$$\pi_{\text{employeeNr,floorName,date}}(\sigma_{\text{date}\leq\text{endDate}}(\texttt{enters} \bowtie \rho_{\text{endDate}\leftarrow\text{date}}(\texttt{leaves})))$$

tuple relational calculus:

$$\{[\text{e.eName}, \text{e.persNr}, \text{e.SVNR}] \mid \text{e} \in \texttt{employee} \wedge \exists \text{b} \in \texttt{enters} \wedge$$
$$\text{e.persNr} = \text{b.employeeNr} \wedge \nexists \text{l} \in \texttt{leaves}($$
$$\text{l.employeeNr} = \text{b.employeeNr} \wedge \text{l.floorName} = \text{b.floorName} \wedge \text{l.date} \geq \text{b.date} \wedge \text{b.floorName} = \text{'H1'})\}$$

domain relational calculus:

$$\{[\text{name}, \text{persNr}, \text{svnr}] \mid ([\text{name}, \text{persNr}, \text{svnr}] \in \texttt{employee} \wedge$$
$$\exists \text{startDate}, \text{floorName}([\text{persNr}, \text{floorName}, \text{startDate}] \in \texttt{enters} \wedge$$
$$\nexists \text{endDate}([\text{persNr}, \text{floorName}, \text{endDate}] \in \texttt{leaves} \wedge \text{startDate} \leq \text{endDate} \wedge \text{floorName} = \text{'H1'})))\}$$

(b) Currently there are students in the lecture room with name 'Gödel'. Find all employees that are currently located on a floor, on which also the students are located.

Solution:
relational algebra:

$$\pi_{\text{persNr}}(\sigma_{\texttt{lecture\_room}.\text{name}=\text{'Gödel'}}((\texttt{enters}-\texttt{ending}) \bowtie \texttt{employee}) \bowtie \texttt{located\_on} \bowtie \texttt{lecture\_room})$$

ending is defined as:

$$\pi_{\text{employeeNr,floorName,date}}(\sigma_{\text{date}\leq\text{endDate}}(\texttt{enters} \bowtie \rho_{\text{endDate}\leftarrow\text{date}}(\texttt{leaves})))$$

tuple relational calculus:

$$\{[\text{m.persNr}] \mid \text{m} \in \texttt{employee} \wedge \exists \text{b} \in \texttt{enters} \wedge \exists \text{l} \in \texttt{located\_on} \wedge \exists \text{h} \in \texttt{lecture\_room} \wedge$$
$$\text{m.persNr} = \text{b.employeeNr} \wedge \nexists \text{v} \in \texttt{leaves}($$
$$\text{v.employeeNr} = \text{b.employeeNr} \wedge \text{v.floorName} = \text{b.floorName} \wedge \text{v.date} \geq \text{b.date}) \wedge$$
$$\text{b.floorName} = \text{l.fName} \wedge \text{l.lrNumber} = \text{h.lrNumber} \wedge \text{h.name} = \text{'Gödel'}\}$$

domain relational calculus:

$$\{[\text{persNr}] \mid \exists \text{name}, \text{svnr}([\text{name}, \text{persNr}, \text{svnr}] \in \texttt{employee} \wedge$$
$$\exists \text{startDate}, \text{floorName}([\text{persNr}, \text{floorName}, \text{startDate}] \in \texttt{enters} \wedge$$
$$\nexists \text{endDate}([\text{persNr}, \text{floorName}, \text{endDate}] \in \texttt{leaves} \wedge \text{startDate} \leq \text{endDate})) \wedge$$
$$\exists \text{lrNr}([\text{lrNr}, \text{floorName}] \in \texttt{located\_on} \wedge [\text{lrNr}, \text{lrName}] \in \texttt{lecture\_room} \wedge \text{lrName} = \text{'Gödel'}))\}$$

(c) Find all employees that were located in different buildings on the same day.

Solution:
relational algebra:

$$\pi_{\text{persNr}}(\sigma_{\text{date}=\text{compDate}\wedge\text{bName}\neq\text{compB}}((\texttt{enters} \bowtie \texttt{located\_in}) \bowtie$$
$$(\rho_{\text{compB}\leftarrow\text{bName},\text{compDate}\leftarrow\text{date}}\texttt{leaves} \bowtie \texttt{located\_in})))$$

tuple relational calculus:

$$\{[\text{b.employeeNr}] \mid \text{b} \in \texttt{enteres} \wedge \exists \text{u1} \in \texttt{located\_in} \wedge \exists \text{u2} \in \texttt{located\_in} \wedge \exists \text{v} \in \texttt{leaves} \wedge$$
$$\text{b.date} = \text{v.date} \wedge \text{b.employeeNr} = \text{v.employeeNr} \wedge$$
$$\text{b.floorName} = \text{u1.fName} \wedge \text{v.floorName} = \text{u2.fName} \wedge \text{u1.bName} \neq \text{u2.bName}\}$$

domain relational calculus:

$$\{[\text{empNr}] \mid ([\text{empNr}, \text{floorName}, \text{date}] \in \texttt{enters} \wedge$$
$$\exists \text{bName}([\text{floorName}, \text{bName}] \in \texttt{located\_in} \wedge$$
$$\exists floorV[\text{empNr}, \text{floorV}, \text{date}] \in \texttt{leaves} \wedge floorName \neq floorV \wedge$$
$$\exists \text{bNameV}([\text{floorV}, \text{bNameB}] \in \texttt{located\_in} \wedge bNameV \neq bName\}$$