Exercise sheet 1 (WS 2021) – 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 04.11.	**12:00pm**	Upload your solutions to TUWEL
17.11.	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]

As you lost the overview of your investments, making a quick Google search and finding no suitable portfolio-management tool you decided to create your own. You start by designing the database.

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.

(see next page)

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:

A Broker has a unique Name (NAME), an API key (KEY), and an endpoint URL (URL), which is used to automatically fetch data.

A Depot has a Name (NAME), which, together with the name of the broker uniquely identifies the Depot. Monthly costs are also stored (COSTS).

The database allows the user to keep track of their Securities. Each Security has a Name (NAME) and an identification number (ISIN). Three types of securities are supported: Stocks, Funds and Bonds. For Funds, the yearly costs (TER) are also stored. Securities are stored in a Depot and identified by the ISIN - which makes it possible for a security to be found in multiple depots.

Securities are bought or sold in Orders. An Order has a timestamp (TIME), a price per piece (PRICE), an amount (NR) and an order cost (FEE). It is uniquely identified by an order id (OId). An order can be a buy or a sell order. For sell-orders, the tax which is paid reclaimed is stored as well (KEST).

An order occurs at exactly one Exchange. For an Exchange, the name (NAME) and timezone (TZ) is stored. For each exchange, it is stored which brokers support the exchange.

Securities give out Dividends. Past Dividends are stored in the database. The date (DATE), the price per piece (PRICE), and the dividend yield (YIELD), are stored. A dividend is identified by the security and the date.

Funds can be made up of any number of other securities stored in the database. It is also stored by what weight these contribute towards the fund (WEIGHTING).

Solution: See Figure 1.

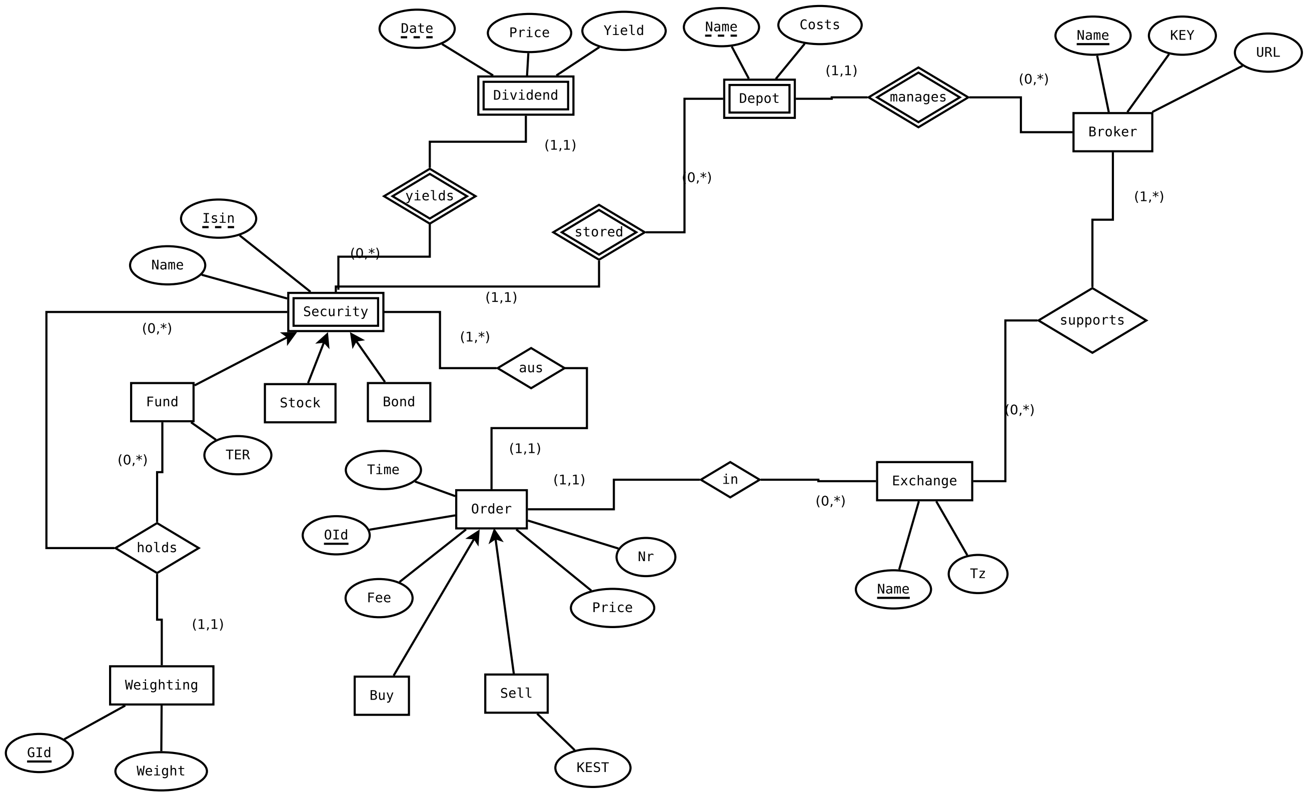

Figure 1: Solution of Exercise 1

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

Consider the EER-diagram shown in Figure 2.

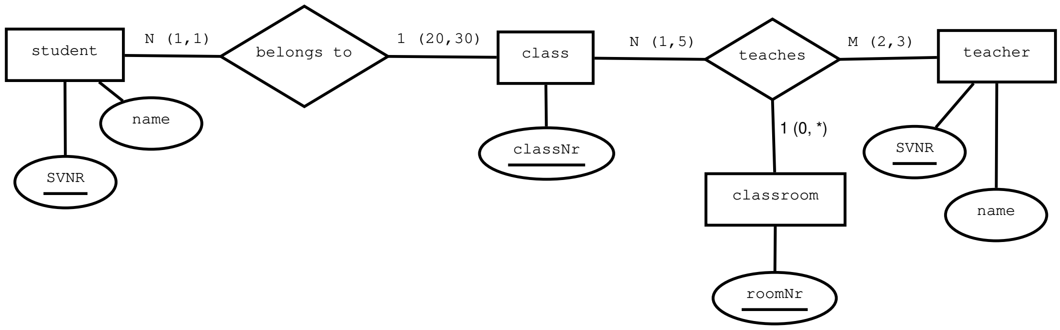

Figure 2: EER-diagram for exercise 2

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.

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 `teaches`, then it cannot be expressed using only the (min,max)-notation that one teacher can teach one class only in one classroom. A concrete example would be that the teacher with SVNR 2000 teaches the class with classNr 1 in the classroom with RoomNr 1 and in the classroom with RoomNr 2. This is not possible if we consider the cardinalities as well.
- In case we omit the *(min,max)-notation* in the relationship type `belongs to`, it cannot be expressed using only the functionalities that one specific class cannot consist of more than 30 students.

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.

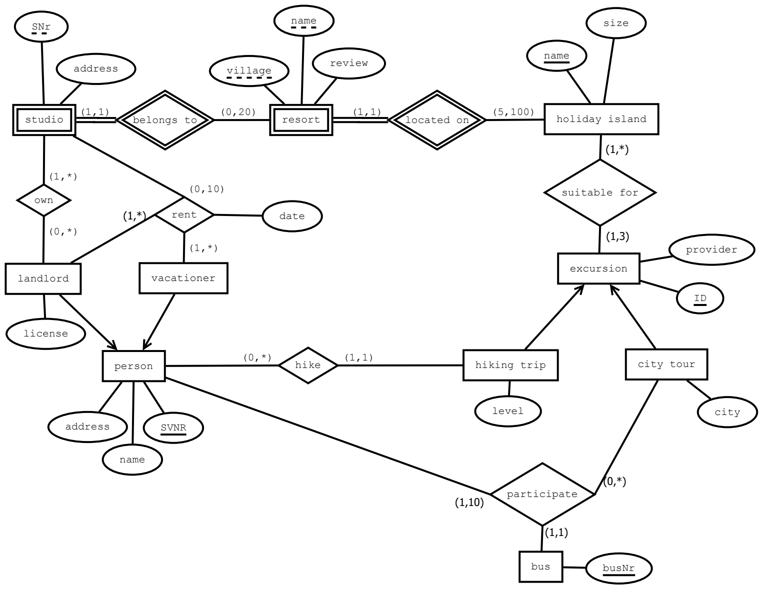

Figure 3: EER-diagram for exercise 3

holiday_island	(<u>name</u>, size)
resort	(<u>holidayisland_name</u>: *holiday_island.name*, <u>village</u>, <u>name</u>, review)
studio	(<u>holidayisland_name</u>: *resort.holidayisland_name*, <u>resort_village</u>: *resort.village*,) <u>resort_name</u>: *resort.name*, <u>SNr</u>, address)
person	(<u>SVNR</u>, address, pname)
vacationer	(<u>SVNR</u>:*person.SVNR*)
landlord	(<u>SVNR</u>:*person.SVNR*, license)
own	(<u>holidayisland_name</u>: *studio.holidayisland_name*, <u>resort_village</u>: *studio.resort_village*, <u>resort_name</u>: *studio.resort_name*, <u>studio_Nr</u>: *studio.SNr*, <u>SVNR</u>:*landlord.SVNR*)
rent	(<u>holidayisland_name</u>: *studio.holidayisland_name*, <u>resort_village</u>: *studio.resort_village*, <u>resort_name</u>: *studio.resort_name*, <u>studio_SNr</u>: *studio.SNr*, <u>landlord</u>:*landlord.SVNR*, <u>vacationer</u>:*vacationer.SVNR*, date)
excursion	(<u>ID</u>, provider)
city_tour	(<u>ID</u>: *excursion.ID*, city)
hiking_trip	(<u>ID</u>: *excursion.ID*, level, hike: *person.SVNR*)
suitable_for	(<u>holidayisland_name</u>: *holiday_island.name*, <u>excursion_ID</u>: *excursion.ID*)
bus	(<u>busNr</u>, transports: *person.SVNR*, belongs_to: *city_tour.ID*)

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.

Manufacturer		
name	**country**	**founded**
Cube	Germany	1993
Specialized	USA	1974
KTM	Austria	1992

Model		
name	**year**	**cost**
Attention	2022	900
Reaction C:62 Pro	2020	1800
Turbo Levo	2021	5600
Rockhopper Expert 29	2020	1200

producedBy	
by	**mname**
Cube	Attention
Cube	Reaction C:62 Pro
Specialized	Turbo Levo
Specialized	Rockhopper Expert 29

Sale		
mname	**date**	**payment**
Reaction C:62 Pro	2020-09-09	card
Reaction C:62 Pro	2021-01-03	cash
Reaction C:62 Pro	2021-06-08	card
Rockhopper Expert 29	2020-04-06	card
Turbo Levo	2021-06-08	card

Provide the results of the following queries over these relations.

(a)

$$\sigma_{sname=name}\Big(\rho_{sname\leftarrow name}\big(\pi_{name,year}(\texttt{Model}) \bowtie \rho_{name\leftarrow mname}(\texttt{Sale})\big) \times \Big(\pi_{name,country}\big(\sigma_{founded>1990}(\texttt{Manufacturer})\big) \ltimes \rho_{name\leftarrow by}(\texttt{producedBy})\Big)\Big)$$

The result is empty, although it still has a schema:

q					
sname	**year**	**date**	**payment**	**name**	**country**

(b)

$$\{m.country, s.date \mid m \in \texttt{Manufacturer} \land s \in \texttt{Sale} \land \exists p \in \texttt{producedBy}\big(p.by = m.name \land \exists o \in \texttt{Model}(o.name = p.mname \land o.cost < 2000)\big)\}$$

q	
country	**date**
Germany	2020-09-09
Germany	2021-01-03
Germany	2020-06-08
Germany	2020-04-06
USA	2020-09-09
USA	2020-01-03
USA	2020-06-08
USA	2020-04-06

Exercise 5 (Equivalences) [2 points]

Consider the following relation schemata $R(A\underline{BC})$, $S(B\underline{DE})$, $T(A\underline{D}F)$ 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 but valid expressions, 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) $q_1\colon \sigma_{A>3 \vee D<4}(R \bowtie S)$ and
$q_2\colon \sigma_{A>3}(R \bowtie S) \cap \sigma_{D<4}(R \bowtie S)$

(b) $q_3\colon (\sigma_{A<3}(\pi_A(T) - \pi_A(R))) \cup \rho_{A\leftarrow E}\pi_E(S)$ and
$q_4\colon ((\pi_A(T) - \sigma_{A<3}(R)) \cup \rho_{A\leftarrow E}\pi_E(S)$

(c) $q_5\colon \pi_{AB}\big(\big((\pi_D(T) - \pi_D(S)) \bowtie T\big) \bowtie S\big) \cap \pi_{AB}(R)$ and
$q_6\colon \pi_{AB}(R) - \rho_{B\leftarrow D}\big(\pi_{AD}(T \bowtie S) \cup \rho_{D\leftarrow B}(\pi_{AB}(R))\big)$

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

In q_1 only those tuples from $R \bowtie S$ are kept, which either have for A a value bigger than 3, or for D a value less than 4.

In q_2, first all tuples from $R \bowtie S$ are kept which have for A avalue bigger than 3, and then all those tuples which have a value less than 4 for D, but then the intersection of the two results is computed instead of the union.

counter example

R		
A	**B**	**C**
5	1	1

S		
B	**D**	**E**
1	6	1
1	3	8

q_1				
A	**B**	**C**	**D**	**E**
5	1	1	6	1
5	1	1	3	8

q_2				
A	**B**	**C**	**D**	**F**
5	1	1	3	8

Ex (b) **No**, q_3 and q_4 **are not** equivalent.

q_3 is a valid expression in relational algebra, but q_4 is not. The reason is that the schemata of $\pi_A(T)$ (only the attribute A) and $\sigma_{A<3}(R)$ (all attribute of R) are different, which is not permitted.

Note that in a version of this exercise sheet, there was a typo in T, where it was defined over $T(CDF)$ instead of $T(ADF)$. In this case this example cannot be solved, as the projection is defined on an attribute, which does not occur in T.

Ex (c) **Yes**, q_5 and q_6 **are** equivalent.

Both return an empty relation and their schemata are the same.

In q_5 the expression

$$\big(\big((\pi_D(T) - \pi_D(S)) \ltimes T\big) \ltimes S\big)$$

always returns an empty relation as result. Therefore, the final result is empty as well (intersection with an empty relation results in an empty relation). The sub-expression above results in an empty relation because: In a first step in

$$\pi_D(T) - \pi_D(S)$$

we pick only those D-values from T which do not occur in any tuple in S as value for D. The result of

$$\big((\pi_D(T) - \pi_D(S)) \ltimes T\big)$$

contains precisely those tuples from T for which there is no tuple in S which has the same value on attribute D. The natural join results therefore in the empty relation.

For q_6 the empty relation is obtained by: The expression

$$\rho_{B\leftarrow D}\big(\pi_{AD}(T \ltimes S) \cup \rho_{D\leftarrow B}(\pi_{AB}(R))\big)$$

contains the union of two relations, where one of these relations is

$$\pi_{AB}(R)$$

which is precisely the same relation that we have on the left side of the operator. The attribute B is renamed to D right before the relation is combined by union with

$$\pi_{AD}(T \ltimes S)$$

but in the next step D is again renamed to B. Therefore, the relation on the right side of the expression contains the relation on the left side of the expression. In the result all tuples from

$$\pi_{AB}(R)$$

are removed and we obtain an empty relation.

Exercise 6 (Answer Sizes)	[1.5 points]

Consider the relational schemas $R(A\underline{B}CD)$, $S(\underline{AB}CD\underline{E})$, and $T(A\underline{CDE})$ as well as 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) $q_1\colon\ (\rho_{A\leftarrow C,B\leftarrow D}(\pi_{C,D}(R)) \bowtie S) ⟖ R$ (with $|R| = 4$ and $|S| = 5$)

(b) $q_2\colon\ \Big(\big(\pi_{A,C}(R) - \pi_{A,C}(S)\big) - \pi_{A,C}(T)\Big) \cup \rho_{A\leftarrow D,C\leftarrow E}\big(\pi_{D,E}(T)\big)$ (with $|R| = 5$, $|S| = 4$ and $|T| = 3$)

(c) $\Big(\pi_{CDE}\big(\sigma_{A=1\vee A\neq 1}(T)\big) \cap \pi_{CDE}(S)\Big) \ltimes \sigma_{A<10}(R)$ (with $|R| = 5$, $|S| = 5$ and $|T| = 1$)

Exercise (a) [Minimum: 4 | Maximum: 8]

Minimum: 4

R

A	<u>B</u>	C	D
1	1	1	1
1	2	1	1
1	3	1	1
1	4	1	1

S

<u>A</u>	<u>B</u>	C	D	<u>E</u>
5	5	5	5	1
5	5	5	5	2
5	5	5	5	3
5	5	5	5	4
5	5	5	5	5

Ergebnis

A	B	C	D	E
1	1	1	1	-
1	2	1	1	-
1	3	1	1	-
1	4	1	1	-

Maximum: 8

R

A	<u>B</u>	C	D
1	1	1	1
1	2	1	2
1	3	1	3
1	4	1	4

S

<u>A</u>	<u>B</u>	C	D	<u>E</u>
1	1	1	1	1
1	1	1	1	2
1	1	1	1	3
1	1	1	1	4
1	1	1	1	5

Ergebnis

A	B	C	D	E
1	1	1	1	1
1	1	1	1	2
1	1	1	1	3
1	1	1	1	4
1	1	1	1	5
1	2	1	2	-
1	3	1	3	-
1	4	1	4	-

It is important to consider the renaming and the right outer join carefully.
The minimum arises from the right outer join with R.
To determine the maximum, it is first to consider that the natural join outputs all tuples from S, for which a matching tuple from R suffices. Then 3 further tuples can be added by the right outer join, since the remaining 3 tuples in R do not match with those in S.

Exercise (b) [Minimum: 1 | Maximum: 8]

Minimum: 1

R

A	B	C	D
1	1	1	1
1	2	1	1
1	3	1	1
1	4	1	1

S

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

T

A	C	D	E
1	1	1	1
1	2	1	1
1	3	1	1

Ergebnis

A	C
1	1

Maximum: 8

R

A	B	C	D
1	1	1	1
1	2	2	2
1	3	3	3
1	4	4	4
1	5	5	5

S

A	B	C	D	E
2	1	2	1	1
2	1	2	1	2
2	1	2	1	3
2	1	2	1	4

T

A	C	D	E
2	1	2	1
2	2	2	2
2	3	2	3

Ergebnis

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

In the case of the minimum, the relations can be chose such that the set differences result in the empty set. At the union with the projection at least one tuple is added.

To get the maximum, the relations can also be chosen such that the 5 tuples from R remain. Then 3 further tuples can be added in the union.

Aufgabe (c) [Minimum: 0 | Maximum: 6]

Minimum: 0

R

A	B	C	D
1	1	1	1
1	2	1	1
1	3	1	1
1	4	1	1
1	5	1	1

S

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

T

A	C	D	E
1	2	2	2

Ergebnis

C	D	E

Maximum: 1

R

A	B	C	D
1	1	1	1
1	2	1	1
1	3	1	1
1	4	1	1
1	5	1	1

S

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

T

A	C	D	E
1	1	1	1

Ergebnis

C	D	E
1	1	1

Both the minimum and the maximum are determined by the set intersection. From the size of T, the maximum is derived. The minimum of 0 can occur when the result of the intersection or the semi-join is empty.

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

$$\pi_{A,B}(R \ltimes (\pi_{A,B}(R) - \pi_{A,B}(\sigma_{C\geq 5}(S))))$$

into tuple relational and domain relational calculus. **domain relational calculus:**

$$\{[a,b] \mid [a,b] \in R \land \neg(\exists c, d([a,b,c,d] \in S \land c \geq 5)))\}$$

tuple relational calculus:

$$\{[r.A, r.B] \mid r \in R \land \neg(\exists s(s \in S \land s.A = r.A \land s.B = r.B \land s.C \geq 5))\}$$

(b) Translate the query

$$\{[a,c] \mid \ \exists b, d([a,b,c,d] \in S \ \land [c,a,5] \in T)\}$$

into tuple relational calculus and relational algebra. **relational algebra:**

$$\pi_{S.A,S.C}(S \bowtie_{S.A=T.C \land S.C=T.A} \sigma_{E=5}(T))$$

tuple relational calculus:

$$\{[s.A, s.C] \mid \ s \in S \land \exists t(t \in T \land s.A = t.C \land s.C = t.A \land t.E = 5)\}$$

(c) Translate the query

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

into domain relational calculus and relational algebra. **relational algebra:**

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

domain relational calculus:

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

Exercise 8 (Formalizing Queries) [4 points]

A company manages houses, including insurance and monthly payments (primary keys are <u>underlined</u>, foreign keys are written in *italics*).

```
House       (HNr, Adress, District)
Owner       (Name, BirthDate, IBAN)
Apartment   (HouseNr: House.HNr, Top, Floor, Size, Nutzwert,
   OwnerName: Owner.Name, OwnerBirthDate: Owner.BirthDate )
Payment     (PNr, Date, Amount,
   ApartmentHouseNr: Apartment.HouseNr, ApartmentTop: Apartment.Top )
Insurance   (Insurer, ContractNr, House: House.HNr)
Claim       (Insurer: Insuranace.Insurer,
   ContractNr: Insurance.ContractNr, ClaimNr, Descr)
involves    (Insurer: Claim.Insurer, ContractNr: Claim.ContractNr,
   ClaimNr: Claim.ClaimNr, ApartmentTop, Apartment.Top)
```

(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) Give out the name and birth date of all owners which own two apartments in a house in the district '1030', which are on the same floor.

relational algebra: (Note: technically, it would be required to rename attributes before joining)

$$\pi_{Name,BirthDate}\sigma_{\text{House.District}='1030'}($$
$$\sigma_{\text{W1.OwnerName=W2.OwnerName}\wedge\text{W1.OwnerBirthDate=W2.OwnerBirthDate}}($$
$$(\sigma_{\text{W1.HouseNr=W2.HouseNr}\wedge\text{W1.Floor=W2.Floor}}($$
$$\rho_{W1}\text{Apartment} \times \rho_{W2}\text{Apartment})) \bowtie_{W1.Hnr=House.HNr} \text{House}))$$

tuple relational calculus:

$$\{[e.Name, e.BirthDate] | e \in \text{Owner} \wedge \exists h \in \text{House} \wedge h.District =' 1030' \wedge \exists w_1, w_2 \in \text{Apartment} \wedge$$
$$w_1.\text{HNr} = h.HNr \wedge w_2.HNr = h.HNr \wedge w_1.Top \neq w_2.Top \wedge$$
$$w_1.\text{Floor} = w_2.\text{Floor} \wedge w_1.\text{OwnerName} = e.\text{Name} \wedge w_2.OwnerName = e.name \wedge$$
$$w_1.\text{OwnerBirthDate} = e.\text{BirthDate} \wedge w_2.\text{OwnerBirthDate} = e.\text{BirthDate}\}$$

domain relational calculus:

$$\{[Name, BirthDate] \mid \exists I([Name, BirthDate, I] \in \text{Owner} \wedge$$
$$\exists HNr, Ad, District([HNr, Ad, District] \in \text{House} \wedge$$
$$h.District =' 1030' \wedge$$
$$\exists Top1, Floor, Gr1, NW1([HNr, Top1, Floor, Gr1, NW1, Name, BirthDate] \in \text{Apartment} \wedge$$
$$\exists Top2, Gr2, NW2[HNr, Top2, Floor, Gr2, NW2, Name, BirthDate] \in \text{Apartment}$$
$$\wedge Top1 \neq Top2))))\}$$

(b) Give out the names of all owner of all appartments which were involved in an insurance case in house nr. 99 in the 2. floor.

relational algebra:

$$\pi_{Name}\sigma_{\text{House.HNr=99}\wedge\text{Apartment.Floor=2}}((\text{Apartment} \bowtie \text{involves} \bowtie Insurance))$$

tuple relational calculus:

$$\{[\text{w.OwnerName}] \mid w \in \text{Apartment} \wedge b \in \text{involves} \wedge v \in \text{Insurance}\wedge$$

$$b.ApartmentTop = w.Top \wedge w.Floor = 2 \wedge v.House = 99 \wedge v.VertragNr = b.VertragNr\}$$

domain relational calculus:

$$\{[EN] \mid \exists HNR, Top, St, Gr, NW, EG([HNr, Top, St, Gr, NW, EN, EG] \in \text{Apartment}\wedge$$

$$\exists Vs, VNr, SchNr([Vs, VNr, SchNr, Top] \in \text{involves}$$

$$\exists Vr, HNr([Vr, VNr, HNr] \in \text{Insurance} \wedge HNr = 99$$

$$)))\}$$

(c) Give out all addresses and districts of the houses in which there are appartements in the 5th floor or higher.

relational algebra:

$$\pi_{Adresse,District}(\text{House} \ltimes \sigma_{Floor\geq 5}(\text{Apartment}))$$

tuple relational calculus:

$$\{[h.Adresse, h.District] \mid h \in \text{House}\wedge$$

$$\exists w \in \text{Apartment} \wedge w.HNr = h.HNr \wedge w.Floor \geq 5\}$$

domain relational calculus:

$$\{[Adr, Bez] \mid \exists HNr([HNr, Adr, Bez] \in \text{House}\wedge$$

$$\exists Top, Floor, Gr, NW, EN, EG([HNr, Top, Floor, Gr, NW, EN, EG] \in \text{Apartment}\wedge$$

$$Floor \geq 5))\}$$