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

EXAM ON	**SAMPLE SOLUTION**	09.01.2023
◯ 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.

Add your student ID and lastname on every sheet, it simplifies entering points.

Question 1: (7)

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

(4 points)

Dependencies	Keys
$F_1 = \{BF \to C, BC \to AF, C \to DE, A \to DF\}$	BGF, BGC, BGA ....
$F_2 = \{F \to A, CF \to D, DG \to E, CF \to BA\}$	CFG ....

b) Consider the relational schemas (R, F_1) and (R, F_2), where $R = ABCDEFG$. Determine whether they are in the specified normal forms, and mark the right answers.

(3 points)

Dependencies	Keys
$F_1 = \{CDE \to A, DEG \to B, ACD \to AE, BF \to CDE, CD \to BFG\}$	BF, CD, DEFG

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

Dependencies	Keys
$F_2 = \{BD \to CE, BFG \to AB, AD \to DG, BEF \to C, BF \to A\}$	BDF

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: **Lastname:**

Question 2: (10)

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

$$F = \{BCD \to A, ADC \to BE, AF \to G, F \to BD, BG \to C\}, \text{Keys} = \{AF, CF, FG\}$$

Decomposition into 3NF (Underline one key in each relation)

R1 $\underline{\mathtt{BCD}}\mathtt{AE}$ ..... R2 $\underline{\mathtt{AF}}\mathtt{G}$ .... R3 $\underline{\mathtt{F}}\mathtt{BD}$ .... R4 $\underline{\mathtt{BG}}\mathtt{C}$ .... R5 ....

Attention: depending on the correct solution, it may not be necessary to use all five subschemas R_1 to R_5.

b) Consider the relational schema $R = ABCDEF$ together with the functional dependencies (FDs) $F = \{AEF \to D, CDF \to AB\}$ and keys $\{ACEF, CDEF\}$. You are given the following subschemas R_i of R:

relational schema	non-trivial functional dependencies	keys
$R_1 = ADEF$	$C_1 = \{AEF \to D\}$	AEF
$R_2 = ABCDF$	$C_2 = \{CDF \to AB\}$	CDF
$R_3 = CDEF$	$C_3 = \emptyset$	$CDEF$

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)	$\otimes$ ja ○ nein	....	○ ja $\otimes$ nein
(R_2, R_3)	○ ja $\otimes$ nein	$AEF \to D$	$\otimes$ ja ○ nein

Question 3: (10)

Assume that a train network manages data in the following database
(primary keys are underlined):

Line (<u>trainno</u>, tname, start, destination, distance)
Conductor (<u>cid</u>, name, wage)
Operator (<u>short</u>, uname, hq)
works (<u>cid</u>:*Conductor.cid*, <u>short</u>:*Operator.short*, <u>trainno</u>:*Line.trainno*)
operates (<u>trainno</u>:*Line.trainno*, <u>short</u>:*Operator.short*,region)

In addition, given the following **database instance**:

Operator:

short	uname	hq
CD	České dráhy	Prag
DB	Deutsche Bahn	Berlin
Leo	Leo Express a.s.	Prag
S	Student Agency	Brno
SZ	Slovenske železnice	Ljubljana
OEBB	Österreichische Bundesbahnen	Wien

works:

cid	short	trainno
4	CD	RJ273
4	CD	RJ274
9	SZ	EC73
9	SZ	EC74
9	CD	RJ274

Conductor:

cid	name	wage
4	Ingrid	12
9	Karin	14
2	Annika	14
1	Elina	14
5	Peter	15
6	Franz	16
3	Sarah	16

Line:

trainno	tname	start	destination	distance
RJ273	Vindobona	Prag	Graz	523
RJ274	Vindobona	Graz	Prag	523
EC73	Albertina	Wien	Ljubljana	384
EC74	Albertina	Ljubljana	Wien	384

a) Consider the folloing query given in **relational algebra**.
Describe *briefly* (**1 short sentence!**) what values are returned by the query. (1 Point)
(The expression trainno='EC*' means that the characters in trainno start with EC.)

$$\pi_{\text{short,trainno,name}}\left(\left(\sigma_{\text{trainno}='EC*'}(\texttt{Line}) \bowtie (\texttt{works} \bowtie \texttt{Conductor})\right) \bowtie \pi_{\text{short}}(\texttt{Operator})\right)$$

Output the shortname of the train operator, trainno, Conductor, where only EC trains are considered and that Conductors works on the train that is operated by a company listed.

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

$$\pi_{name}\left(\texttt{Conductor} \bowtie \pi_{cid}\left(\texttt{works} \div \pi_{trainno}\left(\sigma_{\text{trainno}='RJ*' \wedge \text{short}='CD'}(\texttt{operates})\right)\right)\right)$$

Name of Conductors, who worked on ALL Railjets operated by CD.

c) Formulate a query in the **tuple calculus** that does the following:
Output cids and names of Conductors who worked at least onces on an OEBB train that started in Wien. However, the Conductor should never have worked on a train starting in Graz. (4 Points)

$$\Big\{[s.cid, s.name] \;\Big|\; \exists s \in \texttt{Conductor}\Big(\exists a \in \texttt{works}\Big(s.cid = a.cid \wedge a.short = \texttt{'OEBB'} \wedge$$
$$\exists bl \in \texttt{Line}\big(bl.trainno = a.trainno \wedge bl.start =' \texttt{Wien}'\big)\Big) \wedge$$
$$\neg\exists a' \in \texttt{works}\Big(a'.cid = s.cid \wedge \Big(\exists bl' \in \texttt{Line}\big(bl'.trainno = a'.trainno \wedge \wedge bl.destination =' \texttt{Graz}'\big)\Big)\Big)\Big)\Big\}$$

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

$$\Big\{[\text{name}, \text{start}, \text{trainno}] \;\Big|\; \exists \text{cid}\ \ \exists \text{l}\ [\text{cid}, \text{name}, \text{l}] \in \texttt{Conductor}\ \wedge \exists \text{u}\,[\text{cid}, \text{u}, \text{trainno}] \in \texttt{works} \wedge$$
$$\exists \text{zn}\ \exists \text{d}\ \exists \text{dist}\ [\text{trainno}, \text{zn}, \text{start}, \text{d}, \text{dist}] \in \texttt{Line}\Big\}$$

Result:

name	start	trainno
Ingrid	Prag	RJ273
Ingrid	Graz	RJ274
Karin	Wien	EC73
Karin	Ljubljana	EC74
Karin	Graz	RJ274

Question 4: (10)

Given the following relational schema.

`Company (`<u>`cname`</u>`, address, turnover)`

`Keeper (`<u>`kid`</u>`, cname:`*`Company.cname`*`, wage)`

`Owner (`<u>`oid`</u>`, residence)`

`clientOf (`<u>`oid`</u>`:`*`Owner.oid`*`, `<u>`kid`</u>`:`*`Keeper.kid`*`, charge)`

In addition, given the following database instance:

`Keeper:`

kid	cname	wage
4	Dowdell Cats	12
9	Dowdell Cats	14
2	Pandas Pets	14
1	Parker Kater	14
5	Ferreira Dogs	15
6	Ferreira Dogs	16
3	Ferreira Dogs	16
8	Dowdell Cats	16
7	Parker Kater	18

`Company:`

cname	address	turnover
Dowdell Cats	6716 S. Mariposa	28
Ferreira Dogs	424 Callan Av.	17
Nix Alles	2902 Flint St.	76
Grooms Capybaras	2704 McGee Av.	53
Ludwig Capybaras	7800 River Mist Av.	64
Castleberry Cats	3228 Chettenham Dr.	23
Regalado Kater	22538 6th St.	21
Chu Capybaras	461 Alder St.	75
Parker Kater	16303 Mateo St.	14

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

```
SELECT Company.cname, sum(wage), turnover
FROM Company, Keeper WHERE Company.cname = Keeper.cname
GROUP BY Company.cname, turnover
ORDER BY turnover DESC;
```

Ergebnis der Abfrage:

cname	sum	turnover
Dowdell Cats	42	28
Ferreira Dogs	47	17
Parker Kater	32	14

b) Provide an SQL query for the following task: Output all Companies (cname, addresse, turnover) for which the turnover is higher than (>) 20. Additionally, output corresponding Keepers (pid, wage). Sort by wage of Keepers. Furthermore, restrict the output to companies whose address contains the string "S."' at any position. (3 Points)

```
SELECT Company.cname, Company.address, Company.turnover,
       Keeper.kid, Keeper.wage
FROM Company, Keeper
WHERE Company.cname = Keeper.cname AND
Company.address LIKE '%S.'%' AND Company.turnover > 20
ORDER BY Keeper.wage;
```

c) Provide an SQL query for the following task: List owners (oid), sum of charges, and corresponding company. However, we are interested only in owners whose residency is Berkeley. Sort the result by the sum of charges. (2 Points)

```
SELECT Owner.oid, sum(clientOf.charge),
        Company.name
FROM Owner, Company, kundevon, Keeper
WHERE clientOf.oid = Owner.oid AND
        clientOf.pid = Keeper.pid AND
        Keeper.cname = Company.cname AND
        Owner.residence = 'Berkeley'
GROUP BY oid, cname
ORDER BY sum(clientOf.charge)
```

d) Provide an SQL query for the following task: List owners (oid). Count for each oid how often different companies have been used. Sort by number of companies in descending order. Do not count companies twice. (2 Punkte)

```
SELECT O.kid,
(SELECT count(DISTINCT uname) as num_companies
        FROM clientOf c, Keeper k
        WHERE c.kid = k.kid AND O.kid = c.kid)
        as num_comps
FROM clientOf O
ORDER BY num_comps DESC;
```

Question 5: (9)

You are tasked with modelling a database, which should cover tenancy relationships, flats and persons living in them.

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.

Every person is uniquely identified by the combination of their name and zodiac sign.

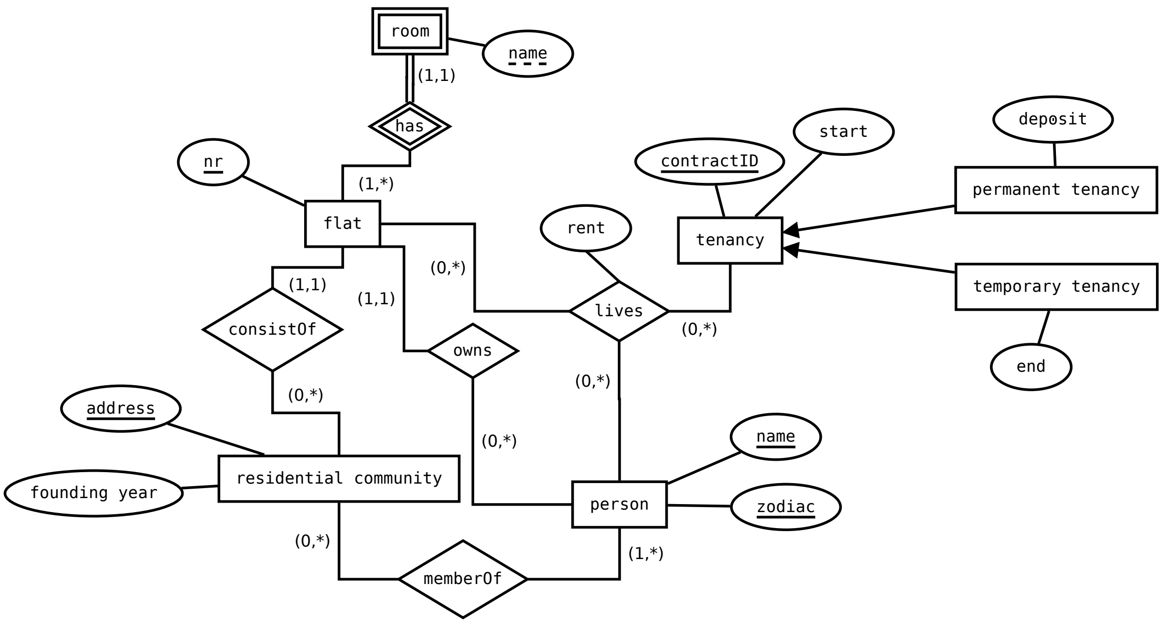
Flats are clearly identified by a number. Each flat consists of a series of rooms. Rooms are clearly marked by combining the number of the flat and their own room number. Each flat consists of at least one room. In our database, each flat is clearly assigned to a person who owns this flat.

Tenancy (relationships) are clearly identified by their contract ID. The start of the tenancy is also recorded as an attribute. A distinction is made between permanent tenancy and temporary tenancy, both of which are forms of tenancy. In the case of permanent tenancy, the deposit is also recorded as an attribute. In the case of temporary tenancy, there is also an attribute that expresses the end of the tenancy.

The relation “lives” expresses which people live in which flats and under which tenancies. The “rent” attribute is also stored as part of this relation.

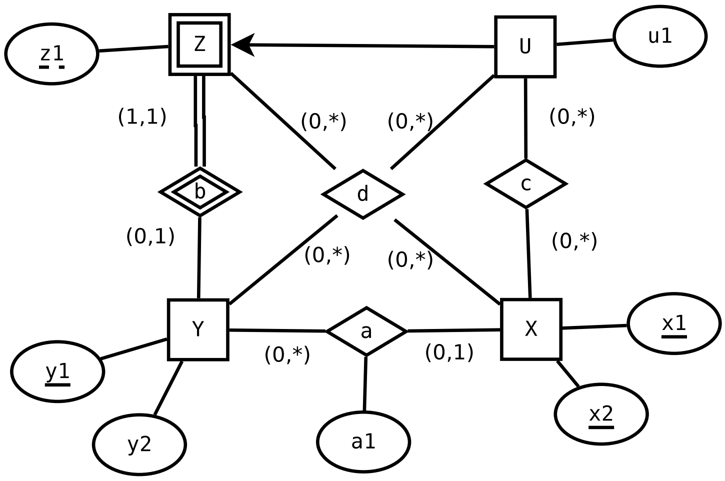
Flats are part of residential communities. Each residential community has a unique address. The founding year is also recorded. Each flat in this database must be part of an residential community. In addition, people can also belong to a residential community. Each person must belong to at least one residential community in this database, but can belong to arbitrarily many.

MatrikelNr: **Lastname:**



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 ($\underline{\text{x1}}$, $\underline{\text{x2}}$ ....)

Y ($\underline{\text{y1}}$, y2 ....)

Z ($\underline{\textit{y1: Y.y1}\text{, z1}}$ ....)

U ($\underline{\textit{y1: Y.y1}\text{, z1}}$, u1 ....)

a (*x1:X.x1*, $\underline{\textit{x2:X.x2}}$, *y1:Y.y1*, a1 ....)

c (*z1:U.z1, y1: U.y1, x1: X.x1, x2: X.x2* ....)

d (*x1: X.x1, x2: X.x2, y1: Y.y1, z1: Z.z1, y1z: Z.y1,* ....
u1: U.u1, z1u: U.z2, y1u: U.y1 ....
(....)

MatrikelNr: ______ **Lastname:** ______

Question 7: (8)

Consider the relational schemas $R(\underline{A}BC)$, $S(\underline{D}E)$, and $T(\underline{AC}E)$. Assume there exists an instance of R containing 2 tuples, an instance of S containing 4 tuples, and an instance of T containing 3 tuples. Thus

$R(\underline{A}BC)$: 2 $\quad$ $S(\underline{D}E)$: 4 $\quad$ $T(\underline{AC}E)$: 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 R, S, and T 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:** $\pi_{A,D}(R \rtimes_{R.A=S.D} S) \cup \pi_{A,D}(S \bowtie \rho_{A \leftarrow D} S)$ (4 Points)

min. size of the result: 4 ....

R

A	B	C
1	–	–
2	–	–

S

D	E
1	1
2	2
3	3
4	4

max. size of the result: 18 ....

R

A	B	C
1	–	–
2	–	–

S

D	E
3	7
4	7
5	7
6	7

b) **Expression:** $(S \bowtie \rho_{A \leftarrow E} S) - \pi_{A,D,E}(T \rtimes S)$ (4 Points)

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

S

D	E
1	5
2	6
3	7
4	8

T

A	C	E
5	–	5
6	–	6
7	–	7

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

S

D	E
1	5
2	6
3	7
4	8

T

A	C	E
–	–	9
–	–	10
–	–	11

Overall: 60 points

Have a successful exam!