Name, first name: Enrollment number: Study program:

_____ _____ _____

Exam
Database System Concepts for Non-Computer Scientists (IN4714), 2 Credits
Winter 2020/21, April 8th, 2021

Important notes for the exam:

- Processing time is 40 minutes; you can gain at most 40 points; to pass, you have to gain at least 50% of the points (20 points)
- Your answers may be in English and /or German
- Please do only use the handed-out sheets for your answers.
- Inscribe the first sheet with your name, enrollment number, study program; for the other sheets your name is sufficient.
- Please check the completeness of your exam. The exam is comprised of
    - 6 pages (including this front page) / 3 sheets
    - 4 assignments
- If you realize that your exam is not complete, please tell us immediately!

Assignments

- Please do not use pencils, and no red or green pens.
- This is a closed book exam.

- Please provide us with an ID card and a student card.
- Please sign this cover sheet.

Good luck!

Signature student: _____

Name, first name:

_____

Name, first name:

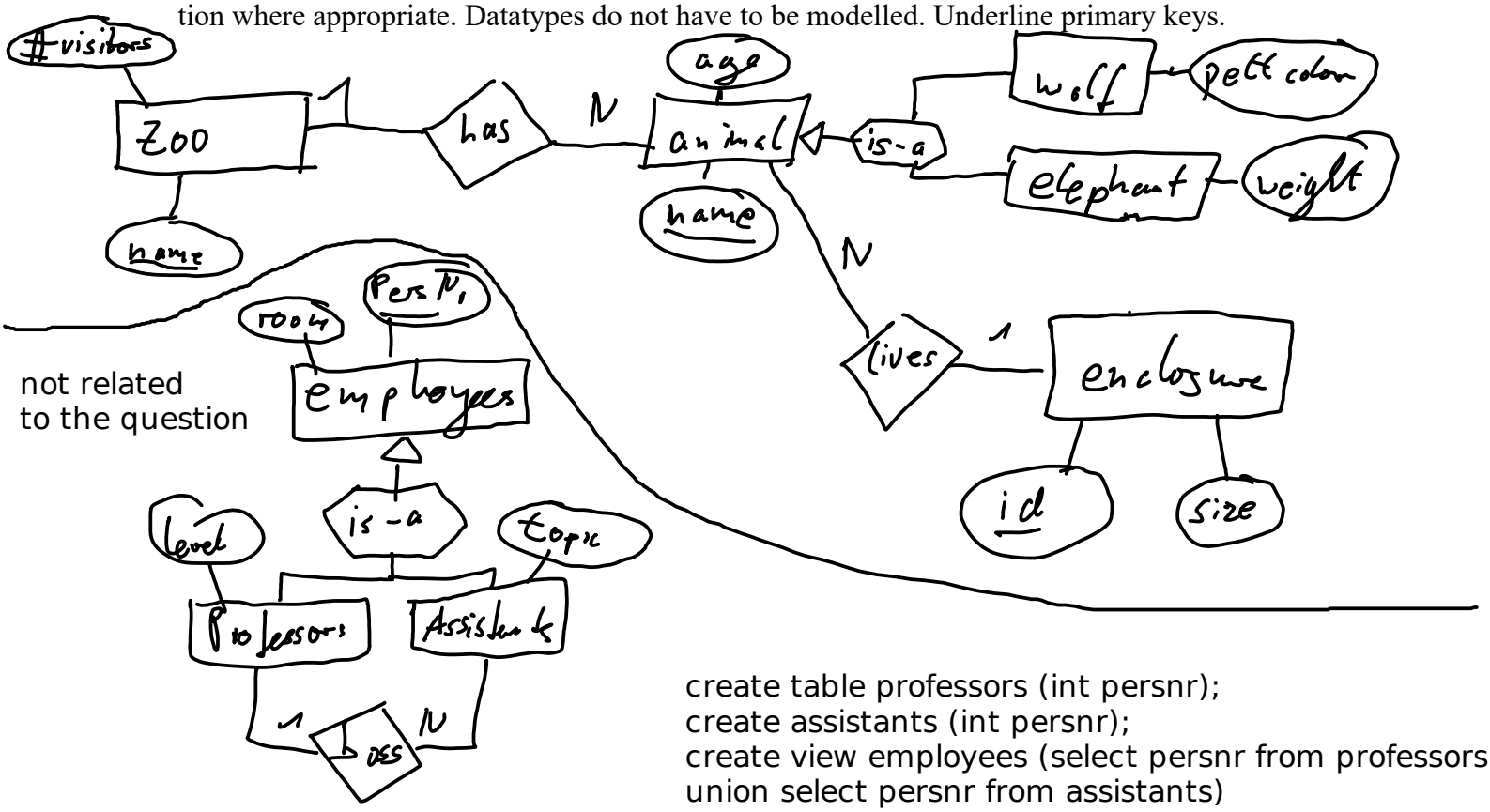
_____

Assignment 1 (ER-Diagrams) 12 Points

Consider the following requirements for a database to manage several zoological gardens (zoos).

- There should be an entity for a zoo. Every zoo has a unique name. In addition, we want to track the maximum number of visitors allowed in each zoo.
- A zoo has animals. Within a single zoo, an animal can be identified by its name. In addition, all animals have an age.
- There are also attributes specific to certain species. Therefore, we want to model each species as a distinct entity. As a start, add wolfs and elephants to the diagram. Wolfs have a pelt color and for elephants we want to track their weight.
- Lastly, each animal lives in an enclosure. Each enclosure has a unique id. In addition, we store the size of the enclosure in square meters.

Create an ER-diagram for the described zoo database. Add functionalities and use generalization where appropriate. Datatypes do not have to be modelled. Underline primary keys.



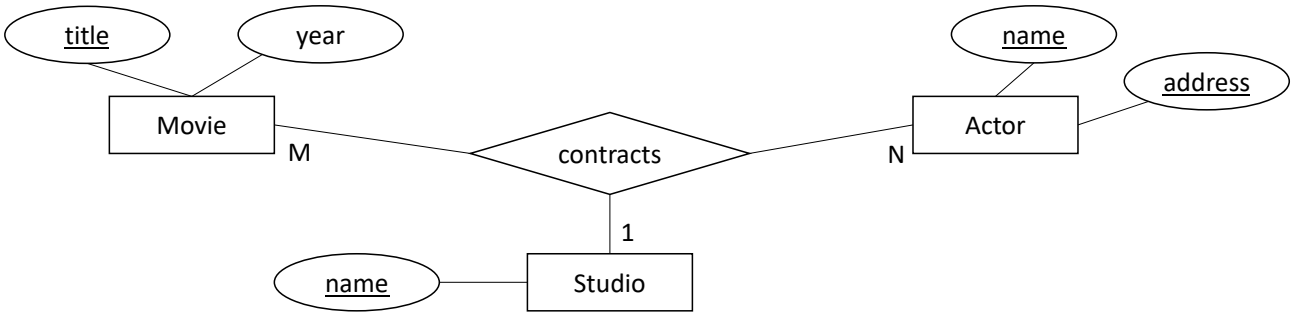

```
create table professors (int persnr);
create assistants (int persnr);
create view employees (select persnr from professors
union select persnr from assistants)
```

Name, first name:

_____

Assignment 2 (Relational Schema and SQL DDL) 8 Points

Consider the following ER-diagram:



a) Which partial function(s) is(are) holding for the *contracts* relation? State it(them) and briefly describe the implied semantics (short explanation in natural language).

a) Movie x Studio → Actor
b) Movie x Actor → Studio
c) Studio x Actor → Movie
d) none of them

b) Create database tables (SQL DLL) for the *Movie* entity and the *contracts* relationship. You can assume that the Actor and Studio tables have already been defined. Use primary keys, foreign keys and unique constraints where and if needed. Choose appropriate data types for the attributes.

```
create table movie (title text primary key, year date);

create table contract (movietitle text references movie(title),
studio text references studio,
name text, address text,
foreign key (name, address) references actor (name, address));
```

Name, first name:

_____

Assignment 3 (SQL-Queries) 15 Points

For this exercise we extend the well-known university schema. We add musical instruments (*instruments* table) and store which student is playing which instrument (*plays* table). For details see the extended schema on the supplementary sheet.
If not state otherwise, formulate SQL (DRL) queries for the extended university schema.

a) Find all students whose name start with an 'F'. Write a SQL query! Expected columns in result: student number, name.

```
select * from students where name like 'F%'
```

b) Evaluate the following query manually on the attached instantiation of the university schema (last page). Please fill in the provided table with the query results.

```
select l.title, count(*) as cnt, avg(s.semester) as sem
from Lectures l
  left outer join attend a on l.lectureNr = a.lectureNr
  left outer join students s on a.studNr = s.studNr
where l.weeklyhours = 3
group by l.lectureNr, l.title
order by l.title desc
```

	title	cnt	sem
5052	Wissenschaftstheorie	1	3
5043	Erkenntnistheorie	1	null

Name, first name:

_____

c) For each lecture, calculate the percentage of students playing at least one instrument. All lectures should be included in the result. A lecture that is not attended by any student should have a percentage of zero or null. Write a SQL query! Expected columns in result: lecture number, lecture title and percentage.

```
with lectureall as ( -- count number of students per lecture
  select l.lecturenr, count(a.studnr)
  from lectures l join attend a on l.lecturenr=a.lecturenr
  group by l.lecturenr),
lectureplays as ( -- count number of students playing in instrument
  select l.lecturenr, count(distinct a.studnr)
  from lectures l left outer join attend a on l.lecturenr=a.lecturenr
  where exists (select * from plays p where p.studnr=a.studnr)
  group by l.lecturenr)

-- divide number of students playing in instrument through of total students
select l.lecturenr, coalesce(cast(pa.count as float)/a.count,0)
from lectures l left outer join lectureplays pa on l.lecturenr=pa.lecturenr
left outer join lectureall a on pa.lecturenr=a.lecturenr
```

d) Find the student(s) who play all instruments in the database. Write a SQL query! Expected columns in result: student number and name.

```
select * from students s
where not exists (
      select * from instruments i where not exists (
            select * from plays p
                  where p.instrumentnr=i.instrumentnr and s.studnr=p.studnr))
```

Name, first name:

_____

Assignment 4 (Miscellaneous) 5 Points

a) What does SQL stand for?

structured language language

b) Give two reasons for using a database system instead of building a system yourself. Briefly explain each reason (one sentence).

acid properties: atomicity, consistency, isolation, durability

c) Name a concept in SQL that can be used to simplify a large query.

common-table-expression (with-construct)

d) What is an alternative to functionalities in the ER-diagrams?

min-max-notation