Full name: Student number: Study program:

_____ _____ _____

Test Exam
December, 18th 2019
Geodatenbanken (Database module)
im WiSe 2019/20

(Modul Geodatenbanken Master UI,
Teil des Moduls Geodatenbanken und Visualisierung Master GuG,
Teil des Moduls Angewandte Geoinformatik im Master UPIÖ)

- You have **33 minutes** to answer all questions on the exam. There are 33 regular points and 5 bonus points.
- The exam consists in 3 exercises; there are 6 sheets of paper.
- No aids are authorized.
- Please write your name, student number, study program on the first page.
- Please write your name on every page.
- Please only use the handed-out sheets.
- You may answer in English or German.
- All sheets have to be handed back after the exam.
- Do not use pencils or red/green pens.
- Please put your student id and a photo id on your desk.
- Please sign the first page.

Good Luck! 🍀

Signature: _____

Full name:

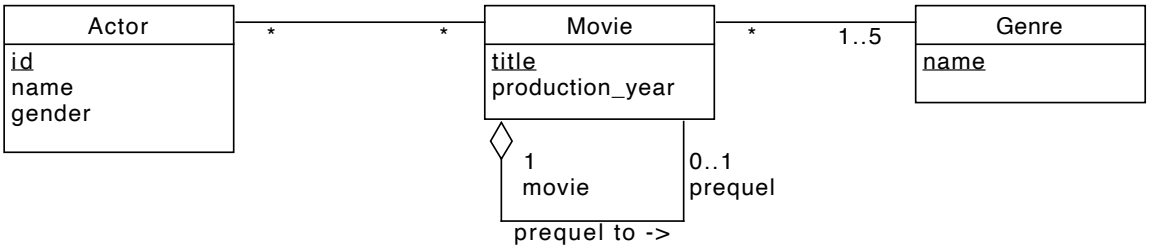
_____

Exercise 1 (Entity relationship model, relational schema, SQL DDL) 10 Points

Consider the following requirements for a database for a movie streaming web site:

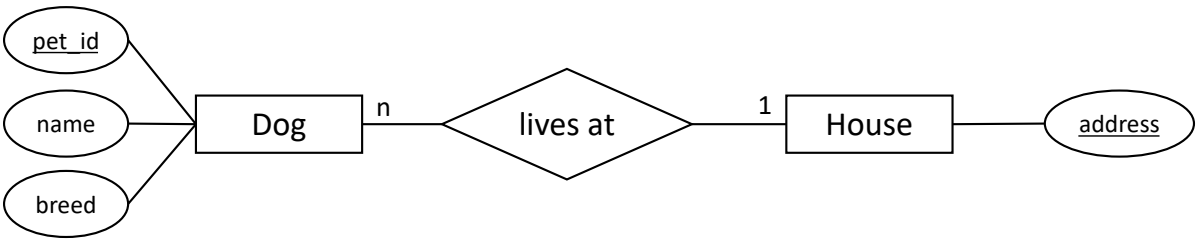
- Movies have a unique title and a production year.
- A movie may have up to one prequel (a movie containing events that precede those of another movie).
- Actors have a unique id, a name and a gender.
- An actor can star in any number of movies.
- A Genre is defined by its name.
- A movie has between 1 and 5 genres.

a) Create a UML class diagram for the described movie streaming service. Add multiplicities to the UML class diagram. Mark the primary keys by underlining them. Datatypes are **not** required.



Clarification (not part of solution): We only need to name (or add roles) to relations that are ambiguous. This often goes for recursive relations. In this case, having an un-named relation without roles between movies would be ambiguous: What does it mean? It could also be a parody or that one movie is referencing the other one.

b) Write SQL-DDL statements to create database tables for the following entity relationship diagram. Make sure to refine the model where possible and define foreign keys, primary keys, and unique constraints as and if necessary.




```
create table House (address varchar(50) primary key);
create table Dog (pet_id int primary key,
                  name varchar(30),
                  breed varchar(30),
                  address varchar(50) references House);
```

Full name:

_____

Exercise 2 (SQL Queries) 18 points [Bonus: 5 points]

Write SQL queries on the known university schema (example instantiation at the end of this exam) for the following tasks:

a) Calculate the average grade students received in the 'Ethik' lecture. Expected column in result: average grade.

```
select avg(grade)
from test t, Lectures l
where t.lectureNr = l.lectureNr
and l.title = 'Ethik'
```

b) Evaluate the following query manually on the attached instantiation of the university schema (last page). Please give attribute/column names and values in form of a table.

```
select p.name as name,
       count(l2.lectureNr) as lec_cnt,
       sum(l2.weeklyhours) as hours_sum
from Professors p
   left outer join lectures l1 on p.persNr = l1.given_by
   left outer join require r on l1.lectureNr = r.successor
   left outer join lectures l2 on r.predecessor = l2.lectureNr
where p.level = 'C4'
  and p.name <> 'Curie' and p.name <> 'Russel'
group by p.persNr, p.name
order by p.name
```

name	lec_cnt	hours_sum
Kant	0	null
Sokrates	2	8

Full name:

_____

c) Find the student (or students) with the best grade. (Expected columns in result: student name, student number, grade, the title of the lecture, and the name of the professor; one student may occur multiple times)

```
select s.name, s.studNr, t.grade, l.title, p.name
from Students s, test t, Professors p, Lectures l
where s.studNr = t.studNr
and p.persNr = t.persNr
and l.lectureNr = t.lectureNr
and t.grade = (select min(grade) from test)
```

d) Eager students: Print out a list of students that have passed a test in all lectures. (Expected columns in result: student number and student name; no duplicates). [BONUS (5pt): Write the query twice: once with counting and once without.]

```
select s.studNr, s.name
from Students s
where not exists (select *
                  from Lectures l
                  where not exists (select *
                                    from test t
                                    where l.lectureNr = t.lectureNr
                                    and t.studNr = s.studNr))
```

```
select *
from Students s
where (select count(*)
       from Lectures)
=
      (select count(distinct t.lectureNr)
       from test t
       where s.studNr = t.studNr)
```

Full name:

_____

Exercise 3 (Common database knowledge) 5 Points

a) Name one famous relational database system.

HyPer, DB2, Microsoft SQL Server, Oracle, PostgreSQL, MySQL, SQLite, Maria DB

b) Give two good reasons for using a database system and briefly explain why.

There are multiple reasons, see slide deck 1 "Introduction":
1. Data redundancy and consistency
2. Data integrity
3. Declarative query language
4. Access rights
5. Concurrency control
6. No data loss (recovery)
7. Efficiency and scalability
8. Cost

c) Assume a non-empty table created as follows:

```
create table Rel(a int primary key not null,
                 b int);
```

What is the difference (if there is any) between the three produced result columns `x, y, z` in the following query?

```
select count(a) as x, count(b) as y, count(*) as z
from Rel;
```

- x and z will always be equal and return the number of tuples (cardinality) of the tables
- y might be less, because the column could contain null values

Full name:

_____

Professors

PersNr	Name	Level	Room
2125	Sokrates	C4	226
2126	Russel	C4	232
2127	Kopernikus	C3	310
2133	Popper	C3	52
2134	Augustinus	C3	309
2136	Curie	C4	36
2137	Kant	C4	7

Students

StudNr	Name	Semester
24002	Xenokrates	18
25403	Jonas	12
26120	Fichte	10
26830	Aristoxenos	8
27550	Schopenhauer	6
28106	Carnap	3
29120	Theophrastos	2
29555	Feuerbach	2

Lectures

Lecture Nr	Title	Weekly Hours	Given_by
5001	Grundzüge	4	2137
5041	Ethik	4	2125
5043	Erkenntnistheorie	3	2126
5049	Mäeutik	2	2125
4052	Logik	4	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133
5022	Glaube und Wissen	2	2134
4630	Die 3 Kritiken	4	2137

attend

StudNr	LectureNr
26120	5001
27550	5001
27550	4052
28106	5041
28106	5052
28106	5216
28106	5259
29120	5001
29120	5041
29120	5049
25403	5022
29555	5022
29555	5001

require

Predecessor	Successor
5001	5041
5001	5043
5001	5049
5041	5216
5043	5052
5041	5052
5052	5259

test

StudNr	LectureNr	PersNr	Grade
28106	5001	2126	1
25403	5041	2125	2
27550	4630	2137	2

Assistants

PersNr	Name	Area	Boss
3002	Platon	Ideenlehre	2125
3003	Aristoteles	Syllogistik	2125
3004	Wittgenstein	Sprachtheorie	2126
3005	Rhetikus	Planetenbewegung	2127
3006	Newton	Keplersche Gesetze	2127
3007	Spinoza	Gott und Natur	2126