Full name:	Student number:	Study program:
____________________	_______________	_______________

Test Exam Solution
December, 19th 2018
Geodatenbanken (Database module)
im WiSe 2018/19

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

- You have **40 minutes** to answer all questions on the exam. There are 40 regular points and 5 bonus points. You need to get at least 20 points (in total) to pass.
- 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! 🍀

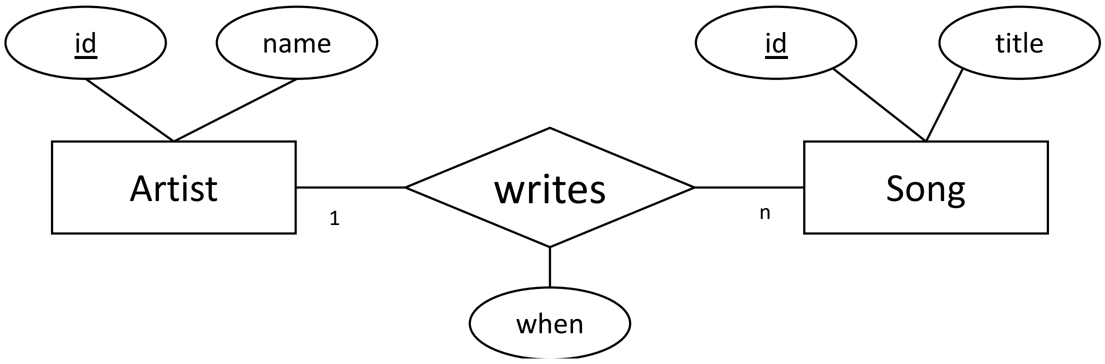
Signature: ______________________________

Full name:

____

Exercise 1 (Entity relationship model, relational schema, SQL DDL) 2+4+4+5 = 15 Points

Consider the following entity relationship model for artists and songs. We assume that composers do not collaborate on songs (i.e., a song is written by exactly one person).



a) Add functionalities to the entity relationship model (directly in the figure).
b) Translate the model into the relational schema and refine it (intermediate steps are not required). Add appropriate datatypes and mark primary keys.

```
Artist: {[id: int, name: varchar]}
Song: {[id: int, title: varchar, a_id: int, when: date]}
```

c) Create the necessary SQL DDL statement to create tables in a database system. Choose appropriate datatypes and specify primary and foreign keys as needed.

```
create table Artist (id int primary key,
                     name varchar);
create table Song (id int primary key,
                   title varchar,
                   a_id int references artist(id),
                   when date);
```

d) We want to add playlists to our system. A playlist should have a name and can contain any number of songs. Write down the SQL DDL statements to add these to the database.

```
create table playlist (id int primary key,
                       name varchar);
create table contained_in (p_id int references Playlist(id),
                           s_id int references Song(id),
                           primary key(p_id, s_id));
```

Full name:

____

Exercise 2 (SQL Queries) 4+6+5+5 = 20 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) Determine all professors that give at least 2 lectures. (Expected columns in result: person number and name of professor; no duplicates)

```
select p.persNr, p.name
from Professors p
where (select count(*)
       from lectures l
       where l.given_by = p.persNr) >= 2
```

b) Execute the following query manually on the attached instantiation of our university database (last page in the exam) and write down the result as a table, including the schema. In addition, please write a sentence explaining what this query calculates.

```
select persNr, name,
       count(lectureNr) as lecture_cnt,
       sum(weeklyhours) as sum_hours
from Professors left outer join Lectures
                on persNr = given_by
where level = 'C4'
group by persNr, name
order by lecture_cnt desc
```

persNr	name	lecture_cnt	sum_hours
2126	Russel	3	8
2125	Sokrates	3	10
2137	Kant	2	8
2136	Curie	0	null

The query calculates the number of lectures and the sum of their weekly hours for every C4 professor.

Full name:

____

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

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

d) Lazy students: Print out a list of all students that do not attend any lecture. (Expected columns in result: student number and student name; no duplicates)

```
select s.studNr, s.name
from Students s
where not exists (select *
                  from attend a
                  where a.studNr = s.studNr)
```

e) [Bonus] Busy students: Print out a list of all students that attend every lecture. (Expected columns in result: student number and student name; no duplicates). [Bonus of bonus and also a hint]: Give second solution that is or is not based on counting (depending on whether your first solution was based on counting).

The trick is to restate the query into a double negative: Find those students where there is no lecture that they do not attend.

```
select s.studNr, s.name
from Students s
where not exists (select *
                  from lectures l
                  where not exists (select *
                                    from attend a
                                    where a.studNr = s.studNr
                                      and a.lectureNr = l.lectureNr))
```

```
Another solution would be to use counting:
      select s.studNr, s.name
      from Students s
      where (select count(*) from attend a where a.studNr = s.studNr)
            = (select count(*) from lectures);
```

Full name:

____

Exercise 3 (Common database knowledge) 5 Points

a) Name one famous relational database system.

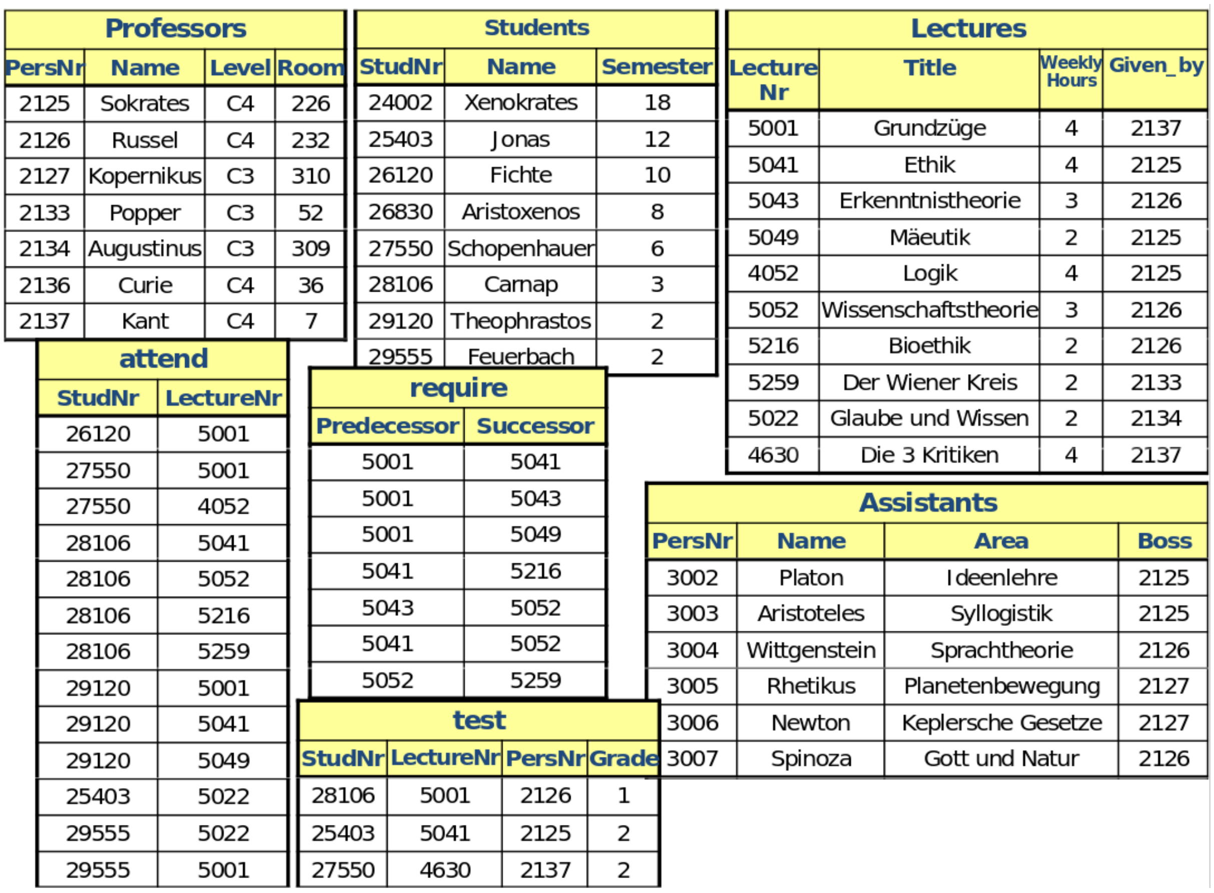
Obviously: HyPer.
However, we also accept (list not exhaustive): MySQL, SQL-Server, Oracle, db2, PostgreSQL, Hana, SQLite

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

Analytical capabilities: It is possible to query data using SQL (DRL).
Integrity constraints: Data stored in a database system has to follow the schema (data types) and adhere to additional constraints (primary key, foreign key, check ..).

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

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

test

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