Question 1 (Relational Algebra) [30 points]

Consider the following relation schema below (primary keys are underlined).

project (<u>pno</u>, pname, plocation, dnum) [dnum references dnumber in department]
work_on (<u>essn, pno</u>, hours) [essn references ssn in employee]
department (<u>dnumber</u>, dname, mgrssn, mgrstartdate)
[mgrssn refers to an ssn of a manager]
employee (<u>ssn</u>, fname, lname, bdate, addr, sex, salary, superssn, dno)
[superssn refers to an ssn of a supervisor, dno references dnumber in department]
dependent (<u>essn, dependent_name</u>, sex, bdate, relationship)
[essn references ssn in employee]

Write the following queries in Relational Algebra.

1. Find the names of all managers who have at least one dependent. [5 points]

   $\Pi_{\text{fname, lname}}\ ((\Pi_{\text{mgrssn}}\ (\text{department}) \cap \Pi_{\text{essn}}(\text{dependent})) \bowtie (\text{employee}))$

2. Find the last names and ssn of all employees who work on project number 1 and on project number 2. [5 points]

   $\Pi_{\text{ssn, lname}}\ (\sigma_{\text{pno} = 1}\ (\text{works\_on}) \bowtie \text{employee}) \cap$
   $\Pi_{\text{ssn, lname}}\ (\sigma_{\text{pno} = 2}\ (\text{works\_on}) \bowtie \text{employee})$

3. Find the names of all employees who earn more than their supervisors. [5 points]

   $\text{Supervisors} \leftarrow \Pi_{\text{superssn}}\ (\text{employee})$
   $\text{SupsSals} \leftarrow \rho_{\text{ssal=salary}}\ (\Pi_{\text{superssn, salary}}\ (\text{employee} \bowtie_{\text{ssn=superssn}} \text{Supervisors}))$
   $\Pi_{\text{fname,lname}}\ (\sigma_{\text{salary} > \text{ssal}}\ (\text{employee} \bowtie_{\text{superssn=superssn}} \text{SupsSals}))$

4. List the names of all employees whose department manager earns less than \$25,000. [5 points]

   $\text{DeptsWithPoorMng} \leftarrow \Pi_{\text{dnumber}}\ (\text{department} \bowtie_{\text{mgrssn=ssn}} (\sigma_{\text{salary} < 25{,}000}\ (\text{employee}))$

   $\Pi_{\text{fname, lname}}\ (\text{employee} \bowtie_{\text{dno=dnumber}} \text{DeptsWithPoorMng})$

5. Find the names of all employees who work on the project that is conducted by department 5. [5 points]

   $\Pi_{\text{fname, lname}}\ ((\Pi_{\text{pno, essn}}(\text{works\_on}) \div \Pi_{\text{pno}}\ (\sigma_{\text{dnum=5}}(\text{project}))) \bowtie \text{employee})$

Question 2 (SQL) [45 points]

A) Consider the following table schemas:

branch (branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
account (account_number, branch_name, balance)
loan (loan_number, branch_name, amount)
depositor (customer_name , account_number)
borrower (customer_name, loan_number)

Write SQL statements for the following queries:

1. Find the number of depositors for each branch. [4 points]

```
Select branch_name, count (distinct customer_name)
from depositor, account
where depositor.account_number = account.account_number
group by branch_name
```

2. Find all customers who have both an account and a loan at the Gainesville branch. [5 points]

```
select distinct customer_name
from borrower, loan
where borrower.loan_number = loan.loan_number
        and branch_name='Gainesville'
        and (branch_name, customer_name) in
                (select branch_name, customer_name
                 from depositor, account
                where depositor.account_number =
                        account.account_number)
```

3. Find the average balance of all customers who live in Gainesville and have more than three accounts. [5 points]

```
Select depositor.customer_name, avg(balance)
From depositor, account, customer
Where depositor.account_number = account.account_number
    and depositor.customer_name = customer.customer_name
    and customer.customer_city = 'Gainesville'
group by depositor.customer_name
having count(distinct depositor.account_number) >= 3
```

4. Find the name of the branch where the average account balance is greater than the other branches. [6 points]

```
select branch_name
from account
group by branch_name
having avg(balance) >= all
        (
          select avg(balance)
          from account
          group by branch_name
        )
```

B) Consider the following table schemas:

dept (deptno, dname, loc)
emp (empno, ename, job, hiredate, sal, deptno) (:employee's information)

Write SQL statements for the following queries:

5. Find the employees (empno, name, job, salary) whose job is the same as the employee with empno = 7777 and who receive less salary than the employee with empno = 8888. [7 points]

```
Select empno, ename, job, sal
from emp
where job =
        (
          select job from emp
          where empno = 7777
        )
and sal <
        (
          select sal from emp
          where empno = 8888
        )
```

6. Find the employees (that is, output deptno, dname, ename, job, hiredate) who work in the sales department and were hired in 2015. (in ascending order by hiredate) (4 points)

```
select e.deptno, d.dname, e.ename, e.job, e.hiredate
from emp e, dept d
where
e.deptno = d.deptno
and e.hiredate >= to_date('2015-01-01')
and e.hiredate <= to_date('2015-12-31')
and d.dname = 'sales'
order by hiredate asc
```

7. Find the departments (that is, output deptno, number of employees) for which more employees work than for other departments. [7 points]

```
select deptno, count(*)
from emp
group by deptno
having count(deptno) =
        (select max(count(*))
         from emp
         group by deptno)
```

8. Find the employees (that is, output empno, name, deptno, hiredate) who have worked longer than others in each department. [7 points]

```
select empno, ename, deptno, hiredate
from emp
where (deptno, hiredate) in (
                select deptno, min(hiredate)
                from emp
                group by deptno)
```

Question 3 (Relational Algebra and SQL) [10 points]

The division (or quotient) operator ÷ of the Relational Algebra does not have a direct equivalent in SQL. It identifies attribute values from a relation that are paired with all of the values from another relation. Without loss of generality, let $\mathbf{R} = \{A_1 : C_1, \ldots, A_n : C_n, B_1 : D_1, \ldots, B_m : D_m\}$ and $\mathbf{S} = \{B_1 : D_1, \ldots, B_m : D_m\}$ be two relation schemas. Let R be a relation with respect to $\mathbf{R}$ and S be a relation with respect to $\mathbf{S}$.

1. Provide the formal definition of the division operator by means of a Relational Algebra expression that only makes use of the basic Relational Algebra operators. [4 points]

$R \div S = \pi_{\mathbf{R}-\mathbf{S}}(R) - \pi_{\mathbf{R}-\mathbf{S}}((\pi_{\mathbf{R}-\mathbf{S}}(R) \times S) - R)$

2. A first solution to mapping the definition in 1. into SQL is somewhat difficult to understand. It makes use of a doubly nested and negated SQL statement and is based on the *not exists* predicate. We are here not interested in this solution. Instead, we follow another idea that maps the different parts of the definition in 1. *one by one* into SQL and does not use the *not exists* predicate. Write down the corresponding SQL query. [6 points]

```
SELECT DISTINCT A1, …, An
FROM R
MINUS
SELECT A1, ..., An
FROM (SELECT * FROM (SELECT A1, …, An FROM R), S)
      MINUS
      R
     );
```

It is obvious that the "…" have to be replaced by the respective attributes $A_2, \ldots, A_{n-1}$.

Question 4 (QBE) [20 points]

Consider the following database schema:

Drivers (did, dname, gender, age)
Reserve (did, cid, day, cost)
Cars (cid, cname, model, color, rid)
RentalCompany (rid, rname, revenue, rating)
IsMember(did, rid, join_time, member_type)

Primary key attributes are underlined.

Answer the following questions using QBE. Draw tables in your answer.

1. Find the oldest driver who is a member of the company 'Budget' and the company 'Avis'. [7 points]

Drivers	Did	dname	gender	age
P.	_Id			_A
¬ (negation)	_Id2			>_A

IsMember	Did	rid	join_time	member_type
	_Id	_Rid1		
	_Id	_Rid2		
	_Id2	_Rid1		
	_Id2	_Rid2		

RentalCompany	Rid	rname	revenue	rating
	_Rid1	'Avis'		
	_Rid2	'Budget'		

2. Find the name of the customer who has reserved a car named 'A6' on '01/03/2016' from the company 'Budget'. [6 points]

Drivers	Did	dname	gender	age

	_Id	P._S		

Reserve	Did	cid	day	cost
	_Id	_Cid	'01/03/2016'	

Cars	cid	cname	model	color	rid
	_Cid	'A6'			_Rid

RentalCompany	Rid	rname	revenue	rating
	_Rid	'Budget'		

3. Update the member type to 'VIP' for those drivers who were members of company 'Avis' and have spent more than 1000 in renting (reserving) cars from Avis. [7 points]

Drivers	Did	dname	gender	age
	_Did			

IsMember	Did	rid	join_time	member_type
U.	_Did	_Rid		'VIP'

RentalCompany	Rid	rname	revenue	rating
	_Rid	'AVIS'		

Reserve	Did	cid	day	cost
	G._Did			SUM.ALL._X

Condition (Reserve)
SUM.ALL._X > '1000'