LAST NAME: ______________________

FIRST NAME: ______________________

DATABASE SYSTEMS
CSCI 331, course # 66047
CSCI 711, course # 66048
Test Solution # 2
May 4, 2016
instructor: Bojana Obrenić

NOTE: **It is the policy of the Computer Science Department to issue a failing grade in the course to any student who either gives or receives help on any test.**

Your ability and readiness to follow the test protocol described below is a component of the technical proficiency evaluated by this test. If you violate the test protocol you will thereby indicate that you are not qualified to pass the test.

this is a **closed-book** test, to which it is **forbidden** to bring anything that functions as: paper, calculator, hand-held organizer, computer, telephone, camera, voice or video transmitter, recorder or player, or any device other than pencils (pens), erasers and clocks;

answers should be written only in the space marked "**Answer:** " that follows the statement of the problem (unless stated otherwise);

scratch should never be written in the answer space, but may be written in the enclosed scratch pad, the content of which *will not be graded;*

any problem to which you give **two or more (different) answers** receives the **grade of zero** automatically;

student name has to be written **clearly** on **each page** of the problem set and on the first page of the **scratch pad** the during the **first five minutes of the test**—there is a penalty of **at least 1 point** for each missing name;

when requested, **hand in** the problem set together with the scratch pad;

once you leave the classroom, you cannot come back to the test;

your **handwriting** must be legible, so as to leave no ambiguity whatsoever as to what exactly you have written.

You may work on as many (or as few) problems as you wish.
time: 75 minutes.

full credit: 100 points.

Good luck.

problem	01	02	03	04	05	06	07	08	09	10	11	12	total [%]
grade													

Problem 1 [**18 points**] Refer to page 9 for definition of the **Research Lab Database.**

(a) Describe the algorithm employed by the database engine to execute the following query, and estimate the worst-case time required for the execution of this query. Explain your answer.

```
select *
from employee
order by salary desc;
```

Answer: Algorithm: multi-way merge-sort, with running time:

$$t = 4Bt_1$$

where B is the number of blocks in the table `employee` and $t_1 = 2^{-10}$ is the block transfer time. Calculate B from record size ℓ, block size b, and record count T:

$$\ell = 32 + 40 + 40 + 16 = 128 = 2^7$$

$$T = 64 \cdot 2^{20} = 2^{26}$$

$$b = 2 \cdot 2^{10} = 2^{11}$$

$$B = \frac{\ell T}{b} = 2^{7+26-11} = 2^{22}$$

$$t = 4 \cdot 2^{22} \cdot 2^{-10} = 2^{14} \text{ [seconds]}$$

Observe that external sort is required, since the table cannot be sorted in memory, which has only

$$2^{30-11} = 2^{19}$$

blocks.

(b) Describe the algorithm employed by the database engine to execute the following query, and estimate the worst-case time required for the execution of this query. Explain your answer.

```
select *
from employee
where name = 'Lupin,Remus';
```

Answer: Algorithm: sequential search, with running time:

$$t' = Bt_1 = 2^{22-10} = 2^{12} \text{ [seconds]}$$

which is more than one hour. Observe that sequential search is unavoidable, since the table has no index on `name`, and is not ordered on name (as is evident from the code.)

LAST NAME: ________________________

FIRST NAME: ________________________

(c) Do you find the execution time of the query considered in part (b) to be satisfactory? Explain your answer.

Answer: Most certainly not satisfactory—it must not take one hour to retrieve a single record.

(d) Write the SQL code to implement a modification of the database that would improve significantly the execution time of the query considered in part (b). Explain your answer briefly. If such a modification is impossible or unnecessary, state it and explain why.

Answer:

```
create index employee_idx_name
on employee (name);
```

This code creates a secondary index on name, so that the query considered in part (b) can be executed by index trace, followed by one data access (at most) per each result record. Since the secondary index will have very low height (single digits) the query will suffer no worse than milliseconds of overhead.

LAST NAME: ______________________________

FIRST NAME: ______________________________

Problem 2 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.**

Write an expression in the relational algebra for the following query. Be careful to use renaming where necessary. Your answer may employ only the following operators: $\sigma, \pi, \times, \cup, \setminus$. (In particular, do not employ $\bowtie$.)

> Find names and target dates of those projects that are not funded by any agency located in NEW YORK.

Answer:

Losers:

$$\Psi = \pi_{\text{pid}_1} \sigma_F \,(\text{fund}_1 \times \text{agency}_2)$$

where

$$F \equiv [\ \text{aid}_1 = \text{id}_2 \ \wedge\ \text{location}_2 = \text{'NEW YORK'}\ ]$$

Winners:

$$\rho = (\pi_{\text{id}}\ \text{project}\ ) \setminus \Psi$$

Result:

$$R = \pi_{\text{name}_1,\ \text{targetDate}_1} \sigma_G \,(\text{project}_1 \times \rho_2)$$

where

$$G \equiv [\ \text{id}_1 = \text{id}_2]$$

Problem 3 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.**

Write a set of safe datalog rules for the following query.

> Find names and target dates of those projects whose funding agencies are situated in at least two different locations.

Answer:

```
Result (n, d) ⟵ project (p, n, d) and
                fund (a1, p, ___) and
                fund (a2, p, ___) and
                agency (a1, ___, c1) and
                agency (a2, ___, c2) and
                c1 ≠ c2
```

LAST NAME: ____________________________

FIRST NAME: ___________________________

Problem 4 [**7 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

Find IDs and names of those projects that do not have a target date.

Answer:

```
select id, name
from project
where targetDate is null;
```

Problem 5 [**7 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

Find IDs and names of those projects whose target date is on the day when the query runs.

Answer:

```
select id, name
from project
where trunc(targetDate)  = trunc (sysdate);
```

LAST NAME: ________________________

FIRST NAME: ________________________

Problem 6 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

> Find names and salaries of those employees which manage a project funded in the amount of at least 10 dollars by at least one agency located in QUEENS.

Answer:

```
select distinct e.name, e.salary
from employee e, manage m, fund f, agency a
where
    e.id = m.eid and
    m.pid = f.pid and
    f.aid = a.id and
    f.amount >= 10 and
    a.location = 'queens';
```

Problem 7 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

> Find IDs of those employees that manage exactly seven different projects.

Answer:

```
select eid from
    ( select eid, count (distinct pid)
      from manage
      group by eid
      having count(distinct pid) = 7
    );
```

LAST NAME: ______________________________

FIRST NAME: ______________________________

Problem 8 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code to implement the following action.

> Reduce by 10% the amount of funding in those cases where the funding agency is located in HOGSMEADE.

nsw

```
update fund
set amount = amount * 0.9
where aid in
    (  select id from agency
       where location = 'hogsmeade'
    );
```

Problem 9 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code to implement the following action.

> Agency whose identifier is '123123123123' will double its funding amount to those projects that it is funding, but on condition that the project is not managed by an employee named GILDEROY LOCKHART.

Answer:

```
update fund
set amount = amount * 2
where
     aid = '123123123123' and
     pid not in
      (  select m.pid
         from manage m, employee e
         where e.id = m.eid and e.name = 'Gilderoy Lockhart'
      );
```

LAST NAME: ______________________________

FIRST NAME: ______________________________

Problem 10 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

> For each employee that is assigned to more than one project, find the id, name, and address of the employee, and the number of projects to which the employee is assigned.

Answer:

```
select e.id, e.name, e.address, count(distinct a.pid)
from employee e, assign a
where e.id = a.eid
group by e.id, e.name, e.address
having count(distinct a.pid) > 1;
```

Problem 11 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.** Write SQL code for the following query.

> For each location where at least one agency exists that funds at least one of the projects, find location, number of such agencies (that provide funding) on that location, and the total amount of funding (to all projects by all agencies) coming from that location. Do not report locations that contribute less than 1000 dollars in total, and order the result in the decreasing order of the total amount.

Answer:

```
select location, count (distinct aid), sum (amount)
from agency a, fund f
where a.id = f.aid
group by location
having sum (amount) >= 1000
order by sum (amount) desc;
```

LAST NAME: ____________________________

FIRST NAME: ____________________________

Problem 12 [**12 points**] Refer to page 9 for definition of the **Research Lab Database.**

Your task is to upgrade the conceptual schema of the database so as to enable additional functionality, as follows.

> Progress monitoring will be introduced for those projects that receive funding. Progress is monitored by funding agencies in the following manner. When an agency wishes to obtain insight into the progress of an individual project that it funds, it agrees with the lab about a date for a review visit for that project. The lab then appoints an employee (not necessarily assigned to that project) to host the agency representatives and discuss with them the project under review. Once the reviewers complete the review, they jointly compose a brief report (say about 1000 letters.) The database must record the *most recent* date (only) of review of this project by this agency, the employee who served as the host, and the actual report.

Write SQL code to modify the database so as to enable it to perform the new functions specified above. Your design must attain the appropriate level of normalization. Your code ought to be not only correct but also readable.

Answer:

```
create table progress (
    aid         varchar (12),
    pid         varchar (12),
    reviewDate  date,
    host        varchar(32),
    report      varchar (1024),
    primary key (aid, pid),
    foreign key (aid, pid) references fund,
    foreign key (host) references employee);
```

The **Research-Lab Database** records data about employees and projects in a research lab with external grant funding.

Every employee has a lab-wide unique employee id, and the lab stores name, address, and salary of each employee. Every employee may (but need not) be assigned to some project(s) executed by the lab. Each project has a name, a target completion date, and a lab-wide unique project id. An individual project may be managed by a project manager, which is one of the lab employees. One employee may manage several projects (and need not manage any.) The lab stores information about funding agencies. Each agency has a lab-wide unique agency id, name and location. Some agencies fund one or more projects, and the lab keeps track of the total amount given by each funding agency to each project.

The database is created by the SQL code given on the right-hand side of this page.

Assume that all tables are stored as files of fixed-length records, so that every character occupies one byte, every numeric item occupies 16 bytes, every date item occupies 16 bytes, and the storage occupied by an entire record is simply the sum of sizes of individual items in that record.

The block size is equal to 2 kilobytes, and the block access time is approximately (1/1024) seconds. One gigabyte of core memory is available.

Assume that the file `employee`, when at its maximum intended size, accommodates about 64 million (potential) employees, and assume that the file is always at this maximum intended size, wherever this assumption may apply.

THIS PAGE IS READ-ONLY.
NO ANSWERS HERE.

LAST NAME: ______________________________

FIRST NAME: ______________________________

```
create table employee (
    id      varchar (32),
    name    varchar (40),
    address varchar (40),
    salary  number (*,0),
    primary key (id)
    );

create table project (
    id          varchar (12),
    name        varchar (36),
    targetDate  date,
    primary key (id)
    );

create table agency (
    id        varchar (12),
    name      varchar (36),
    location  varchar (16),
    primary key (id)
    );

create table assign (
    pid  varchar (12),
    eid  varchar (32),
    primary key (pid, eid),
    foreign key (pid) references project,
    foreign key (eid) references employee
    );

create table manage (
    pid  varchar (12),
    eid  varchar (32),
    primary key (pid),
    foreign key (pid) references project,
    foreign key (eid) references employee
    );

create table fund (
    aid     varchar (12),
    pid     varchar (12),
    amount  number (*,0) check (amount > 0),
    primary key (aid, pid),
    foreign key (aid) references agency,
    foreign key (pid) references project
    );
```