

Name: _____

CSE 444, Spring 2009, Midterm Examination
11 May 2009

Rules:

- Open books and open notes.
- No laptops or other mobile devices.
- Please write clearly.
- Relax! You are here to learn.

Question	Max	Grade
1	40	
2	30	
3	30	
Total	100	

Initials: _____

1. (40 points) **SQL**

Consider a database with the following four base relations and view. For the view, we give the SQL statement that created it.

Customer		
<u>cid</u>	name	address
1	Anna	Capitol Hill
2	Bob	University
3	Charlie	View Ridge

Owns	
<u>cid</u>	<u>aid</u>
1	1
1	2
2	1
2	2
3	3
3	4

Account		
<u>aid</u>	amount	type
1	900	checking
2	2000	savings
3	900	checking
4	100	savings

ActivityLog					
<u>opid</u>	<u>cid</u>	optype	<u>aid1</u>	<u>aid2</u>	amount
101	1	deposit	1	NULL	100
102	2	withdrawal	1	NULL	200
103	3	transfer	3	4	100

Owns.cid is a foreign key that references Customer.cid.

Owns.aid is a foreign key that references Account.aid.

ActivityLog.cid is a foreign key that references Customer.cid.

ActivityLog.aid1 and ActivityLog.aid2 are foreign keys that reference Account.aid.

```
CREATE VIEW GoldenCustomer AS
  SELECT DISTINCT C.cid, C.name
  FROM Customer C, Owns O, Account A
  WHERE C.cid = O.cid AND O.aid = A.aid AND A.amount > 1000
```

Initials: _____

- (a) (15 points) What is the output of the following query? Please draw the relation instance produced by this query in the form of a table. Include the table header to clearly show the schema of the relation. The order of the tuples in the table does not matter.

```
SELECT count(*) as total_activity
FROM Account A2, ActivityLog L2
WHERE A2.aid = L2.aid1 OR A2.aid = L2.aid2
GROUP BY A2.aid
```

Solution:

This query shows the total number of operations made on each account. Accounts without activity will not appear in the result since we are NOT doing an outer join. The output relation is as follows:

<u>total_activity</u>
1
1
2

Initials: _____

- (b) (15 points) What is the output of the following query? Notice that this query uses the previous one as a subquery.

```
SELECT C.name
FROM   Account A, ActivityLog L, Customer C, Owns O
WHERE  (A.aid = L.aid1 OR A.aid = L.aid2)
AND    A.aid = O.aid AND O.cid = C.cid
GROUP BY C.name, A.aid
HAVING count(*) >=
        ALL (SELECT count(*)
              FROM   Account A2, ActivityLog L2
              WHERE  A2.aid = L2.aid1 OR A2.aid = L2.aid2
              GROUP BY A2.aid)
```

Solution:

This query shows the names of the clients that own the account with the highest activity.

name

Anna

Bob

Initials: _____

- (c) (10 points) What is the output of the following query? For this query, you can show the attributes in any order. If two attributes have the same name, indicate the relation where they come from.

```
SELECT *
FROM GoldenCustomer G, Owns O
WHERE G.cid = O.cid
```

Solution:

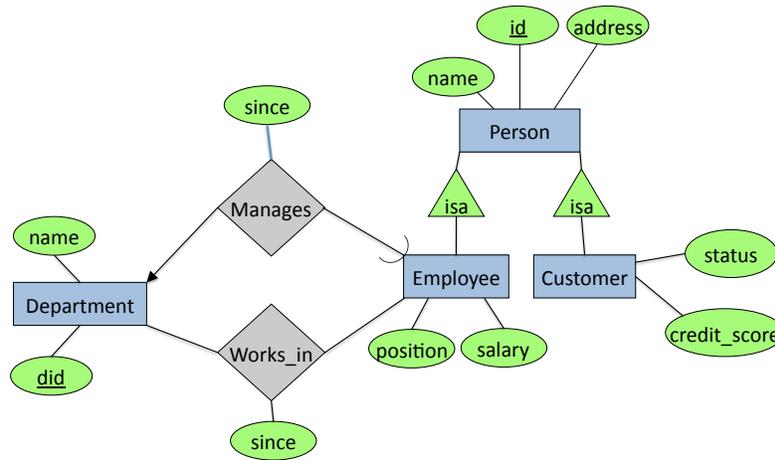
This query shows the accounts owned by those customers who have at least one account with more than \$1,000.

Customer.cid	name	Owns.cid	aid
1	Anna	1	1
1	Anna	1	2
2	Bob	2	1
2	Bob	2	2

Initials: _____

2. (30 points) **Conceptual Design**

- (a) (10 points) Consider the following E/R diagram. Show a valid conversion of this diagram into relations. For each relation, indicate the **primary key** (you can simply underline the attributes that form the primary key) and any **foreign keys**. Note: you do NOT need to show any SQL statements. You only need to show the schema of the relations. You do NOT need to specify the types of the attributes, just their names.



Solution:

Person (id, name, address)

Employee (id, position, salary)
Employee.id is a foreign key that references Person.id.

Customer (id, status, credit_score)
Customer.id is a foreign key that references Person.id.

Department(did, name, manager_id, since)
Department.manager_id is a foreign key that references Employee.id
Manager_id is also a key for Department.

Works_in (did, eid, since)
Works_in.did is a foreign key that references Department.did
Works_in.eid is a foreign key that references Employee.id

Initials: _____

- (b) (10 points) If a relation is in 1NF but not in BCNF (nor in 3NF), what type of problem (or anomaly) can occur? Please **name one** anomaly and provide a **one to two sentence description** of this anomaly.

Solution:

Consider the following relation that stores the information for patients in a clinic : **PatientOf** (patient_no, name, address, doctor_no, since). This relation is not in BCNF because the key is (patient_no, doctor_no) (since a patient can see more than one doctor), yet we have the functional dependency: patient_no \rightarrow name, address.

If a relation, such as the one in our example, is not in BCNF, the following types of anomalies can occur:

- Redundancy: Information may be repeated unnecessarily in several tuples. In the example, we repeat the patient information unnecessarily for each doctor that the patient sees.
- Update anomaly: Updating one copy of repeated data creates an inconsistency unless all copies are similarly updated. If the patient moves, we need to update multiple tuples.
- Deletion anomaly: It is not possible to delete certain information without losing some other, unrelated, information as well. In our example, we cannot have patients without associated doctors, which may be a problem when a doctor leaves and his patients need to be assigned to other doctors.
- Insertion anomaly: It is not possible to store certain information unless some other, unrelated, information is stored as well. In our example, we cannot have patients without associated doctors, which may be a problem when a new patient first joins the clinic. (Note that we did not discuss Insertion anomalies in class. They are related to the other anomalies above.)

- (c) (10 points) Consider the following relational schema and set of functional dependencies. Decompose R into BCNF. Show your work for partial credit.

R(A,B,C,D,E,F,G,H) with functional dependencies A \rightarrow BG, C \rightarrow D, and EF \rightarrow CH.

Solution:

Try A+ = BG. Decompose into R1(A,B,G) and R2(A,C,D,E,F,H).

Decompose R2(A,C,D,E,F,H). Try C+ = D. Decompose into R3(C,D) and R4(A, C, E, F, H).

Decompose R4(A, C, E, F, H). Try EF+ = CH. Decompose into R5(E,F,C,H) and R6 (A, E, F).

End result: R1(A,B,G) (key=A), R3(C,D) (key=C), R5(E,F,C,H) (key=EF), and R6 (A, E, F) (key = AEF).

Other decompositions are possible.

Initials: _____

3. (30 points) **Transactions**

After a system crash, the **redo-log** using non-quiescent checkpointing contains the following data:

```
< START T1 >
< T1, A, 10 >
< START T2 >
< T2, B, 5 >
< T1, C, 7 >
< START T3 >
< T3, D, 12 >
< COMMIT T1 >
< START CKPT ???? >
< START T4 >
< T2, E, 5 >
< COMMIT T2 >
< T3, F, 1 >
< T4, G, 15 >
< END CKPT >
< COMMIT T3 >
< START T5 >
< T5, H, 3 >
< START CKPT ???? >
< COMMIT T5 >
```

- (a) (10 points) What are the correct values of the two <START CKPT ????> records? You have to provide two correct values for the two ????s.

Solution:

First START CKPT: < START CKPT (T2, T3) >

Second START CKPT: < START CKPT (T4, T5) >

Initials: _____

- (b) (10 points) Indicate and **explain** what fragment of the log the recovery manager needs to read.

Solution:

Since the second START CKTP does not have an associated END CKPT, we cannot be sure that committed transactions prior to the start of this checkpoint had their changes written to disk. Thus, we must search for the previous checkpoint. In the previous START CKPT, T2 and T3 were the two active transactions. Both transactions committed and must thus be redone. T2 was the first one to start. The recovery manager must thus read the log record starting from < START T2 > and must read until the end of the log file.

- (c) (10 points) Assuming that the two < START CKPT ??? > records are correctly stored in the log, according to your answer above, show which elements are recovered by the redo recovery manager and compute their values after recovery.

Solution:

We must redo the changes made by all *committed* transactions there were either active during the first START CKPT or that started after that point. T2 and T3 were active during the first START CKPT and committed. T4 and T5 started after the checkpoint but only T5 committed. We must thus redo the changes by T2, T3, and T5. Elements B, D, E, F, and H are thus recovered. Their values after recovery are as follows:

- B=5
- D=12
- E=5
- F=1
- H=3