

CS 480: DATABASE SYSTEMS – Summer 2020

Homework 2: ER model, RA and SQL– 100 pts

Due on: Saturday, July 11

No late submissions accepted

Type your answers in the boxes provided in this pdf file and upload it to Gradescope. If you need more space, clearly state that in the corresponding box and attach pages at the end of the file.

Handwritten answers are NOT acceptable. Only submissions through Gradescope before the deadline will be graded.

Name:

NetID:

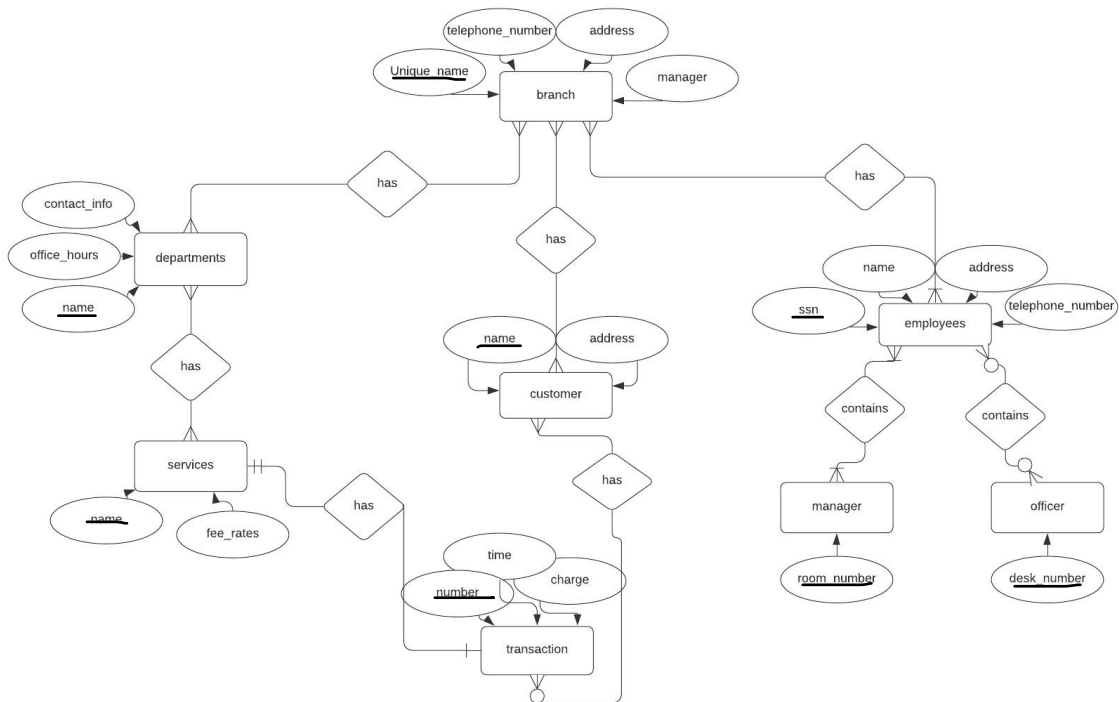
Part A (40 pts):

You are required to design a database system to assist with the administration of a Post Office System. The following is the specification for the post office system:

- a. The post office system has several branch offices. Each branch office has a unique name, one or more telephone numbers, an address, and a manager who supervises the operation of the branch office. Every manager has a designated room number.
 - b. Each branch office has several employees. (Managers are also employees.) Each employee has his/her SSN, name, address, and home telephone number recorded. Employees are either managers or officers. Every branch office has only one manager. No two branch offices have the same employees. Each officer has a designated desk number.
 - c. Branch offices have several departments, each responsible for different services. A department has its unique name, contact information, and office hours. Services provided by a department are identified by their inner service numbers within this department. The system should also record services names and fee rates.
 - d. Customers go to the branch offices for different services, which are 2 recorded as transactions. Each transaction has a unique transaction number. Transaction time and charge should also be kept. A customer's name and address are stored in the system. A transaction is finished by an officer and only one kind of service can be included in one transaction. Give your design, for the post office database system, as an ER diagram.
-

ER diagram

Ivan Madrid-Gonzalez | July 9, 2020



Part B (60 pts):

Consider the bank database

Branch (branch name, branch city, assets)

customer (customer name, customer street, customer city)

loan (loan number, branch name, amount)

borrower (customer name, loan number)

account (account number, branch name, balance)

depositor (customer name, account number)

1. (10 pts) Answer the following questions:
 - a. What are the appropriate primary keys?

branch name --> branch
customer name --> customer
loan number --> loan
account number --> account

- b. Given your choice of primary keys, identify appropriate foreign keys.

customer name, loan number --> borrower
customer name, account number --> depositor

2. (15 pts) Give an expression in the relational algebra for each of the following queries:
 - a. Find all loan numbers with a loan value greater than \$10,000.

π loannumbers(σ amount > '10000'(loan))

- b. Find the names of all depositors who have an account with a value greater than \$6,000.

π customer names (σ balance > '6000'(depositor \bowtie account))

- c. Find the names of all depositors who have an account with a value greater than \$6,000 at the "Uptown" branch.

π customer name(σ balance > '6000' /\ branch name = 'Uptown'(account \bowtie depositor))

3. (15 pts) Construct the following SQL queries for this relational database bank.

- a. Find all customers who have an account at *all* the branches located in "Brooklyn".

```
select customer name
from depositor d
join account a
on d.account number = a.account number
join branch b
on a.branch name = b.branch name
where b.branch city = 'Brooklyn';
```

- b. Find out the total sum of all loan amounts in the bank.

```
select sum(amount)
from loan;
```

- c. Find the names of all branches that have assets greater than those of at least one branch located in "Brooklyn".

```
select branch name
from branch
where assets > some(select assets from branch where branch city = 'Brooklyn');
```

4. (20 pts) Populate the bank database with the below data and execute the SQL queries you have written in 3a, 3b and 3c. Copy the screenshots of the answer to each of your queries.

```
-- branch table
create table branch
(branch_name varchar(50) primary key,
branch_city varchar(50),
assets int
);

-- insert into branch table
insert into branch(branch_name, branch_city, assets)
values('Brighton', 'Brooklyn', 7100000),
('Downtown', 'Brooklyn', 9000000),
('Mianus', 'Horseneck', 400000),
('North Town', 'Rye', 3700000),
('Perryridge', 'Horseneck', 1700000),
('Pownall', 'Bennington', 300000),
('Redwood', 'Palo Alto', 2100000),
('Round Hill', 'Horseneck', 8000000);
```

```
-- create table customer
create table customer
(customer_name varchar(50) primary key,
customer_street varchar(50),
customer_city varchar(50)
);

-- insert into customer table
insert into customer(customer_name, customer_street, customer_city)
values('Adams', 'Spring', 'Pittsfield'),
('Brooks', 'Sensor', 'Brooklyn'),
('Curry', 'North', 'Rye'),
('Glenn', 'Sand Hill', 'Woodside'),
('Green', 'Walnut', 'Stamford'),
('Hayes', 'Main', 'Harrison'),
('Johnson', 'Alma', 'Palo Alto'),
('Jones', 'Main', 'Harrison'),
('Lindsay', 'Park', 'Pittsfield'),
('Smith', 'North', 'Rye'),
('Turner', 'Putnam', 'Stamford'),
('Williams', 'Newark', 'Princeton');
```

```
-- depositor table
create table depositor
(customer_name varchar(50),
account_number varchar(50)
);

-- insert into depositor table
insert into depositor(customer_name, account_number)
values('Hayes', 'A-102'),
('Johnson', 'A-101'),
('Johnson', 'A-201'),
('Jones', 'A-217'),
('Lindsay', 'A-222'),
('Smith', 'A-215'),
('Turner', 'A-305');
```

```
-- borrower table
create table borrower
(customer_name varchar(50),
loan_number varchar(50)
);

-- insert into borrower table
insert into borrower(customer_name, loan_number)
values('Adams', 'L-16'),
('Curry', 'L-03'),
('Hayes', 'L-15'),
('Jackson', 'L-14'),
('Jones', 'L-17'),
('Smith', 'L-11'),
('Smith', 'L-23'),
('Williams', 'L-17');
```

```
-- account table
create table account
(account_number varchar(50) primary key,
branch_name varchar(50),
balance int
);

-- insert into account table
insert into account(account_number, branch_name, balance)
values('A-101', 'Downtown', 500),
('A-102', 'Perryridge', 400),
('A-201', 'Brighton', 900),
('A-215', 'Mianus', 700),
('A-217', 'Brighton', 750),
('A-222', 'Redwood', 700),
('A-305', 'Round Hill', 350);
```

```
-- loan table
create table loan
(loan_number varchar(50) primary key,
branch_name varchar(50),
amount int
);

-- insert into loan table
insert into loan(loan_number, branch_name, amount)
values('L-11', 'Round Hill', 900),
('L-14', 'Downtown', 1500),
('L-15', 'Perryridge', 1500),
('L-16', 'Perryridge', 1000),
('L-17', 'Downtown', 1000),
('L-23', 'Redwood', 2000),
('L-03', 'Mianus', 500);
```

1. branch

branch-name	branch-city	assets
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000

2. customer

customer-name	customer-street	customer-city
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

3. depositor

customer-name	account-number
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305

4. borrower

customer-name	loan-number
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

5. account

account-number	branch-name	balance
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350

6. loan

loan-number	branch-name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

a. Solution to query 3a

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```

1  Select customer_name
2  from depositor d
3  join account a
4  on d.account_number = a.account_number
5  join branch b
6  on a.branch_name = b.branch_name
7  where b.branch_city = 'Brooklyn'

```

The Results tab shows the following data:

customer_name
Johnson
Jones

The Action Output tab shows the execution log, including the creation of the loan table and the execution of the query.

b. Solution to 3b

The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL query:

```

1  Select sum(amount)
2  from loan

```

The Results tab shows the following data:

sum(amount)
1113

The Action Output tab shows the execution log, including the creation of the loan table and the execution of the query.

c. Solution to 3c

MySQL Workbench

Local instance 3306

Administration Schemas bankdatabase SQL File 3*

Limit to 1000 rows

```
1 select branch_name
2 from branch
3 where assets > some(select assets from branch where branch_city = 'Brooklyn');
```

100% 1:1

Result Grid

branch_name
Downtown
Round Hill
...

Object Info Session

No object selected

branch 25

Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 40	19:55:40	Select a.branch_name from branch a join bra...	1 row(s) returned	0.00038 sec / 0.000...
✓ 41	19:56:49	Select a.branch_name from branch a join bra...	1 row(s) returned	0.00053 sec / 0.000...
✗ 42	19:57:13	Select a.branch_name from branch a join bra...	Error Code: 1055, Expression #1 of SELECT list is not l...	0.00042 sec
✓ 43	19:57:21	Select a.branch_name from branch a join bra...	1 row(s) returned	0.00072 sec / 0.000...
✓ 44	19:57:35	Select a.branch_name from branch a join bra...	1 row(s) returned	0.00036 sec / 0.000...
✓ 45	19:59:00	Select a.branch_name from branch a inner jo...	1 row(s) returned	0.00036 sec / 0.000...
✓ 46	19:59:08	Select a.branch_name from branch a inner jo...	1 row(s) returned	0.00037 sec / 0.000...
✓ 47	20:05:38	select branch_name from branch where asse...	2 row(s) returned	0.00071 sec / 0.000...

Query Completed