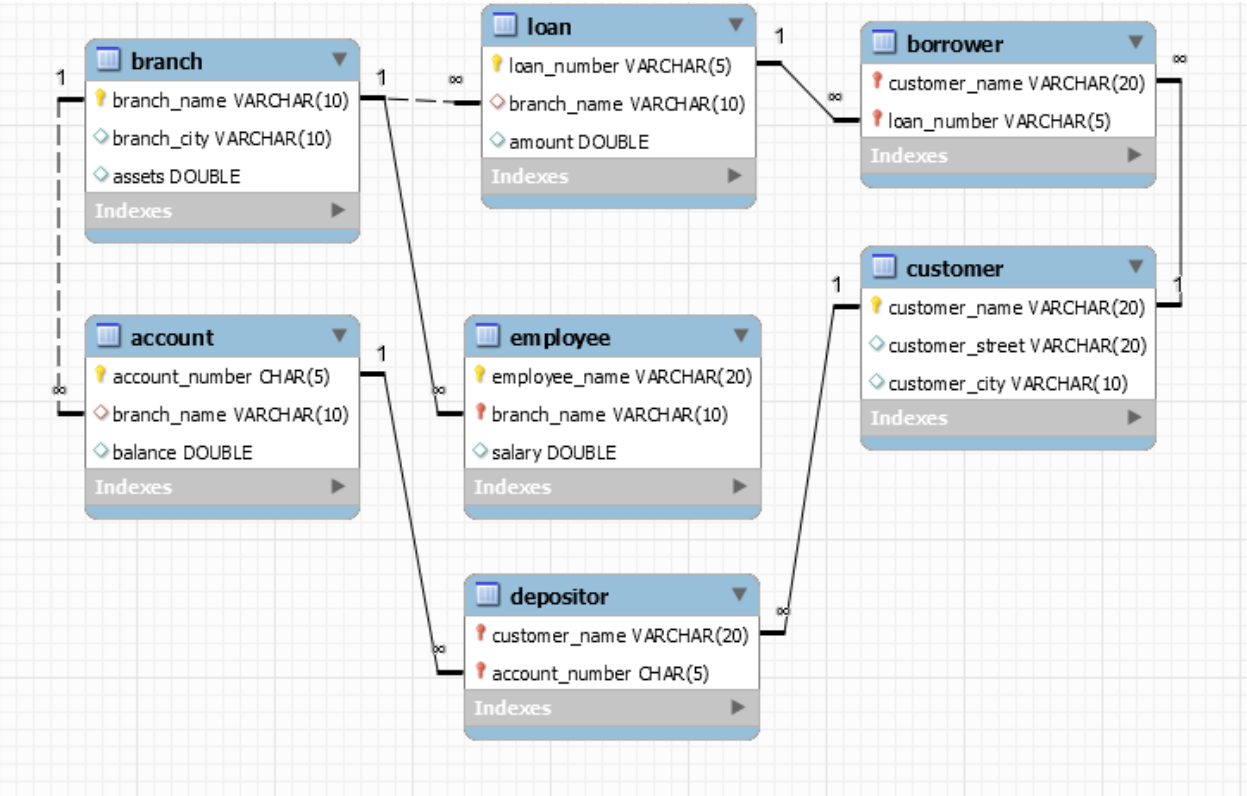
Computer Science Division
Department of Mathematics
Faculty of Science

Level 2- Semester 1
Course: COMP 207
Date: Week 9 (Nov 11), 2017

Sheet 6

يتم تقييم الطالب على انجاز هذا الـ sheet (2 درجة للتصميم ودرجة للـ insert و 7 درجات للـ Queries).
من حق الطالب الاطلاع على أي مصدر (lectures or/and sheets)

Q1) Consider the ER diagram shown in Figure for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans.



account ('A-101', 'Downtown', 500), account ('A-102', 'Perryridge' 400), account ('A-201', 'Brighton', 900), account ('A-215', 'Mianus',)), account ('A-217', 'Brighton', 750), account ('A-222', 'Redwood', 700), account ('A-305', 'Round Hill', 350).

branch ('Brighton', 'Brooklyn', 7100000), branch ('Downtown', 'Brooklyn', 9000000), branch ('Mianus', 'Horseneck', 400000), branch ('North Town', 'Rye', 3700000), branch ('Perryridge', 'Horseneck', 1700000), branch

('Pownal', 'Bennington', 300000), branch ('Redwood', 'Palo Alto', 2100000), branch ('Round Hill', 'Horseneck', 8000000);

customer ('Adams', 'Spring', 'Pittsfield'), customer ('Brooks', 'Senator', 'Brooklyn'), customer ('Curry', 'North', 'Rye'), customer ('Glenn', 'Sand Hill', 'Woodside'), customer ('Green', 'Walnut', 'Stamford'), customer ('Hayes', 'Main', 'Harrison'), customer ('Johnson', 'Alma', 'Palo Alto'); customer ('Jones', 'Main', 'Harrison'), customer ('Lindsay', 'Park', 'Pittsfield'), customer ('Smith', 'North', 'Rye'), customer ('Turner', 'Putnam', 'Stamford'), customer ('Williams', 'Nassau', 'Princeton').

depositor ('Hayes', 'A-102'), depositor ('Johnson', 'A-102'), depositor ('Johnson', 'A-201'), depositor ('Jones', 'A-217'), depositor ('Lindsay', 'A-222'), depositor ('Smith', 'A-215'), depositor ('Turner', 'A-305');

loan ('L-11', 'Round Hill', 900), loan ('L-14', 'Downtown', 1500), loan ('L-15', 'Perryridge', 1500), loan ('L-16', 'Perryridge', 1300), loan ('L-17', 'Downtown', 1000), loan ('L-23', 'Redwood', 2000), loan ('l-93', 'Mianus', 500),

borrower ('Adams', 'L-16'), borrower ('Curry', 'L-93'), borrower ('Hayes', 'L-15'), borrower ('Jackson', 'L-14'), borrower ('Jones', 'L-17'), borrower ('Smith', 'L-11'), borrower ('Smith', 'L-23'), borrower ('Williams', 'L-17');

employee ('Adams', 'Perryridge', 1500), employee ('Brown', 'Perryridge', 1300), employee ('Gopal', 'Perryridge', 5300), employee ('Johnson', 'Downtown', 1500), employee ('Loreena', 'Downtown', 1300), employee ('Peterson', 'Downtown', 2500), employee ('Rao', 'Austin', 1500), employee ('Sato', 'Austin', 1600);

Then write the SQL statement for each of the following queries:

1. Find all loans over $1200.

```
select *
FROM loan
where amount>1200
```

L-14	Downtown	1500

L-15	Perryridge	1500
L-16	Perryridge	1300
L-23	Redwood	2000

2. Find all customer names that have account number or not.
   SELECT C.customer_name, D.account_number
   FROM customer C LEFT OUTER JOIN depositor D on
   C.customer_name=D.customer_name

Adams	Null
Brooks	Null
Curry	Null
Glenn	Null
Green	Null
Hayes	A-102
Johnson	A-102
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305
Williams	Null

3. List all details of all employees with the second letter is ‘a’.
   SELECT *
   FROM EMPLOYEE

WHERE employee.employee_name like '_a%'

Rao	Austin	1500
Sato	Austin	1600

4. List all customers and the total of borrowed loans. Sort the output according to total loans.

```
SELECT C.customer_name, sum(L.amount) as Total_loans

FROM customer C,borrower B, loan L

WHERE C.customer_name=B.customer_name

and B.loan_number=L.loan_number

group by C.customer_name

order by 2
```

Curry	500
Jones	1000
Williams	1000
Adams	1300
Hayes	1500
Smith	2900

5. List all customers who borrow loans exceeds 1200.

```
SELECT C.customer_name, sum(L.amount) as Total_loans
FROM customer C,borrower B, loan L
WHERE C.customer_name=B.customer_name
and B.loan_number=L.loan_number
group by C.customer_name
having sum(L.amount)>1200
order by 2
```

Adams	1300
Hayes	1500
Smith	2900

6. List the largest loan amount in the bank.

```
SELECT max(amount)
FROM loan
```

2000

7. List all customers who borrow total loans exceeds the larges loan in the bank.

```
SELECT C.customer_name, sum(L.amount) as Total_loans
FROM customer C,borrower B, loan L
WHERE C.customer_name=B.customer_name
and B.loan_number=L.loan_number
group by C.customer_name
having sum(L.amount)>=(SELECT max(amount)FROM loan)
```

Smith	2900

List

1. Find the names of all customers who have a loan, an account, or both from the bank.

```
SELECT *
FROM customer
WHERE customer_name in (SELECT customer_name
FROM depositor) or
customer_name in (SELECT customer_name
 FROM borrower)
```

Adams	Spring	Pittsfield
Curry	North	Rye
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

2. Find the names of all customers who have a loan and an account at the bank.

```
SELECT *
FROM customer
WHERE customer_name in (SELECT customer_name
                        FROM depositor) and
customer_name in (SELECT customer_name
                  FROM borrower)
```

Hayes	Main	Harrison
Jones	Main	Harrison
Smith	North	Rye

3. Find the names of all customers who have a loan at the Perryridge branch.

```
SELECT C.customer_name
```

FROM customer C,borrower B,loan L
WHERE C.customer_name=B.customer_name and
B.loan_number=L.loan_number and
L.branch_name = 'Perryridge'

Adams
Hayes

4. Find the names of all customers who have a loan at the Perryridge branch, but no account at any branch of the bank.
5. Find the names of all customers who have an account at the Downtown and Mianus branches.

6. Find the average loan amount of each customer
   SELECT C.customer_name, avg(L.amount) as AVG_LOAN
   FROM ((Customer C natural join borrower B) natural Join Loan L)
   GROUP BY C.customer_name

Adams	1300
Curry	500
Hayes	1500
Jones	1000
Smith	1450
Williams	1000

7. Find the names of all customers who have an account at every branch located in Brooklyn
   SELECT C.customer_name,B.branch_city

FROM (((Customer C natural join depositor D) natural Join account A) natural JOIN branch B)
WHERE B.branch_city like '%Brooklyn%'.

Johnson	Brooklyn
Jones	Brooklyn

8. Update the salary of adams by increasing 30% to his salary

   update employee set salary=1.3*salary where employee_name like 'adams'

9. Delete the employees who has salary less than 1200.

   DELETE employees WHERE salary <1300

10. List the count of employees that work on the downtown branch.

```
SELECT count(*)
FROM employee
WHERE branch_name like 'Downtown'
```

3

[Hint: after finishing answer the sheet, create a view for each query by writing the following query in the sql editor]

- **create view view1 as (SELECT * FROM player)**

- or go to the views on MySQL workbench, right click, create view, then write your query.

Very important note, you can write a query based on a stored view such as
SELECT name FROM view1.