CSC634_DMS hw2
by Sihyuan Han

- Retrieval Queries

1. find all loan number for loans made at the Perryridge branch with loan amounts greater than $1100

```
SELECT
   loan_number
FROM
   loan
WHERE
   branch_name = 'Perryridge'
      AND amount > 1100;
```

```
# loan_number
'L-16'
'L-15'
```

2. find the loan number of those loans with loan amounts between $1,000 and $1,500

```
SELECT
   loan_number
FROM
   loan
WHERE
   amount BETWEEN 1000 AND 1500;
```

```
# loan_number
'L-14'
'L-15'
'L-16'
'L-17'
```

3. find the names of all branches that have greater assets than some branch located in Brooklyn

```
SELECT
   b1.branch_name
FROM
   branch b1
      JOIN
   branch b2
WHERE
   b1.assets > b2.assets
```

```
        AND b2.branch_city = 'Brooklyn';
```

```
# branch_name
'Downtown'
'Round Hill'
```

4. find the customer names and their loan numbers for all customers having a loan at some branch

```
SELECT
    bo.customer_name, bo.loan_number
FROM
    borrower bo
        INNER JOIN
    loan l ON (bo.loan_number = l.loan_number);
```

```
# 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. find all customers who have a loan, an account, or both

```
SELECT
    customer_name
FROM
    borrower
UNION SELECT
    customer_name
FROM
    depositor;
```

```
# customer_name
'Adams'
'Curry'
'Hayes'
'Jackson'
'Jones'
'Smith'
```

```
'Williams'
'Johnson'
'Lindsay'
'Turner'
```

6. find all customers who have an account but no loan (no minus operator provided in mysql)

```
SELECT DISTINCT
   customer_name
FROM
   depositor
WHERE
   customer_name NOT IN (SELECT
         customer_name
      FROM
         borrower);
```

```
# customer_name
'Johnson'
'Lindsay'
'Turner'
```

7. find the number of depositors for each branch

```
SELECT
   branch_name, COUNT(customer_name) AS depositor_total
FROM
   branch,
   customer
WHERE
   branch.branch_city = customer.customer_city
GROUP BY branch_name;
```

```
# branch_name, depositor_total
'Brighton', '1'
'Downtown', '1'
'North Town', '2'
'Redwood', '1'
```

8. find the names of all branches where the average account balance is more than $500

```
SELECT
   branch.branch_name, AVG(balance)
```

```
FROM
    branch
        JOIN
    account ON (branch.branch_name = account.branch_name)
GROUP BY branch.branch_name
HAVING AVG(balance) > 500;
```

```
# branch_name, AVG(balance)
'Brighton', '825'
'Mianus', '700'
'Redwood', '700'
```

9. find all customers who have both an account and a loan at the bank

```
SELECT DISTINCT
    c.customer_name
FROM
    customer c
        INNER JOIN
    depositor d ON (c.customer_name = d.customer_name)
        INNER JOIN
    borrower bo ON (c.customer_name = bo.customer_name);
```

```
# customer_name
'Hayes'
'Jones'
'Smith'
```

10. find all customers who have a loan at the bank but do not have an account at the bank

```
SELECT DISTINCT
    customer_name
FROM
    borrower
WHERE
    customer_name NOT IN (SELECT
            customer_name
        FROM
            depositor);
```

```
# customer_name
'Adams'
'Curry'
```

```
'Jackson'
'Williams'
```

11. find the names of all branches that have greater assets than all branches located in Horseneck (using both non-nested and nested select statement)
(1) non-nested

```
SELECT DISTINCT
   br1.branch_name
FROM
   branch br1,  (SELECT
         MAX(assets) as max_assets
      FROM
         branch
      WHERE
         branch_city = 'HorseNeck') as br2
WHERE
   br1.assets > br2.max_assets;
```

```
# branch_name
'Downtown'
```

(2) nested

```
SELECT
   branch_name
FROM
   branch
WHERE
   assets > ALL (SELECT
         assets
      FROM
         branch
      WHERE
         branch_city = 'HorseNeck');
```

```
# branch_name
'Downtown'
```

12. 1 query of your choice involving aggregate functions: highest employee's salary

```
SELECT
   MAX(salary) AS highest_salary
FROM
```

```
    employee;

# highest_salary
'5300'
```

13. 1 query of your choice involving group by feature: average salary by branch_city

```
SELECT
    branch_city, ROUND(AVG(assets), 2) AS avg_assets
FROM
    branch
GROUP BY branch_city;

# branch_city, avg_assets
'Brooklyn', '8050000'
'Horseneck', '3366666.67'
'Rye', '3700000'
'Bennington', '300000'
'Palo Alto', '2100000'
```

- Insert Queries
1. create a HighLoan table with loan amount >=1500

```
CREATE TABLE HighLoan AS SELECT * FROM
    loan
WHERE
    amount >= 1500;

SELECT
    *
FROM
    HighLoan;

# loan_number, branch_name, amount
'L-14', 'Downtown', '1500'
'L-15', 'Perryridge', '1500'
'L-23', 'Redwood', '2000'
```

2. create a HighSalaryEmployee table with employee having salary more than 2000

```
CREATE TABLE HighSalaryEmployee AS SELECT * FROM
    employee
WHERE
    salary > 2000;
```

```
SELECT
    *
FROM
    HighSalaryEmployee;
```

```
# employee_name, branch_name, salary
'Gopal', 'Perryridge', '5300'
'Peterson', 'Downtown', '2500'
```

3. 1 more query (meaningful) of your choice on any table: create table with customers in Stamford

```
CREATE TABLE StamfordCustomer AS SELECT * FROM
    customer
WHERE
    customer_city = 'Stamford';
```

```
SELECT
    *
FROM
    StamfordCustomer;
```

```
# customer_name, customer_street, customer_city
'Green', 'Walnut', 'Stamford'
'Turner', 'Putnam', 'Stamford'
```

- Update Queries
1. increase all accounts with balances over $800 by 7%, all other accounts receive 8%

```
UPDATE account
SET
    balance = CASE
        WHEN balance > 800 THEN balance * 1.07
        WHEN balance <= 800 THEN balance * 1.08
    END;
```

before

```
# 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'
```

after

```
# account_number, branch_name, balance
'A-101', 'Downtown', '540'
'A-102', 'Perryridge', '432'
'A-201', 'Brighton', '963'
'A-215', 'Mianus', '756'
'A-217', 'Brighton', '810'
'A-222', 'Redwood', '756'
'A-305', 'Round Hill', '378'
```

2. do 2 update queries, each involving 2 tables
(1) increase 50% of employee salary where branches are located in Brooklyn

```
UPDATE branch br
        JOIN
    employee e ON (br.branch_name = e.branch_name)
SET
    e.salary = e.salary * 1.5
WHERE
    br.branch_city = 'Brooklyn';
```

before

```
# branch_city, employee_name, salary
'Horseneck', 'Adams', '1500'
'Horseneck', 'Brown', '1300'
'Horseneck', 'Gopal', '5300'
'Brooklyn', 'Johnson', '1500'
'Brooklyn', 'Loreena', '1300'
'Brooklyn', 'Peterson', '2500'
```

after

```
# branch_city, employee_name, salary
'Horseneck', 'Adams', '1500'
'Horseneck', 'Brown', '1300'
'Horseneck', 'Gopal', '5300'
'Brooklyn', 'Johnson', '2250'
'Brooklyn', 'Loreena', '1950'
'Brooklyn', 'Peterson', '3750'
```

(2) decrease 10% of amount in Horseneck city

```
UPDATE branch br
        JOIN
    loan l ON (br.branch_name = l.branch_name)
SET
    l.amount = l.amount * 0.9
WHERE
    br.branch_city = 'Horseneck';
```

before

```
# branch_city, loan_number, amount
'Horseneck', 'L-11', '900'
'Brooklyn', 'L-14', '1500'
'Horseneck', 'L-15', '1500'
'Horseneck', 'L-16', '1300'
'Brooklyn', 'L-17', '1000'
'Palo Alto', 'L-23', '2000'
'Horseneck', 'l-93', '500'
```

after

```
# branch_city, loan_number, amount
'Horseneck', 'L-11', '810'
'Brooklyn', 'L-14', '1500'
'Horseneck', 'L-15', '1350'
'Horseneck', 'L-16', '1170'
'Brooklyn', 'L-17', '1000'
'Palo Alto', 'L-23', '2000'
'Horseneck', 'l-93', '450'
```

3. 1 more update query of your choice on any table: revise city Brooklyn to Queens

```
UPDATE branch
SET
    branch_city = 'Queens'
WHERE
    branch_city = 'Brooklyn';
```

before

```
# 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'
```

after

```
# branch_name, branch_city, assets
'Brighton', 'Queens', '7100000'
'Downtown', 'Queens', '9000000'
'Mianus', 'Horseneck', '400000'
'North Town', 'Rye', '3700000'
'Perryridge', 'Horseneck', '1700000'
'Pownal', 'Bennington', '300000'
'Redwood', 'Palo Alto', '2100000'
'Round Hill', 'Horseneck', '8000000'
```

- Delete Queries

1. delete the record of all accounts with balances below the average at the bank

```
DELETE a1 . * FROM account a1
        JOIN
    (SELECT
        AVG(balance) AS avg_balance
    FROM
        account) a2
WHERE
    a1.balance < a2.avg_balance;
```

before

```
# 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'
```

after

```
# account_number, branch_name, balance
'A-201', 'Brighton', '900'
'A-215', 'Mianus', '700'
'A-217', 'Brighton', '750'
'A-222', 'Redwood', '700'
```

2. do 2 delete queries, each involving 2 tables
(1) delete records from account table which branch is located in Brooklyn with balance < 600

```
DELETE a FROM account a
        JOIN
    branch br ON a.branch_name = br.branch_name
WHERE
    branch_city = 'Brooklyn'
    AND balance < 600;
```

before

```
# 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'
```

after

```
# account_number, branch_name, balance
'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'
```

(2) delete records which branch is located in HorseNeck with salary < 5000

```
DELETE br, e FROM branch br
        JOIN
    employee e ON br.branch_name = e.branch_name
WHERE
    branch_city = 'HorseNeck'
    AND salary < 5000;
```

before

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'
```

employee

```
# employee_name, branch_name, salary
'Adams', 'Perryridge', '1500'
'Brown', 'Perryridge', '1300'
'Gopal', 'Perryridge', '5300'
'Johnson', 'Downtown', '1500'
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
'Rao', 'Austin', '1500'
'Sato', 'Austin', '1600'
```

after

branch

```
# branch_name, branch_city, assets
'Brighton', 'Brooklyn', '7100000'
'Downtown', 'Brooklyn', '9000000'
'Mianus', 'Horseneck', '400000'
'North Town', 'Rye', '3700000'
'Pownal', 'Bennington', '300000'
'Redwood', 'Palo Alto', '2100000'
'Round Hill', 'Horseneck', '8000000'
```

employee

```
# employee_name, branch_name, salary
'Gopal', 'Perryridge', '5300'
'Johnson', 'Downtown', '1500'
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
'Rao', 'Austin', '1500'
'Sato', 'Austin', '1600'
```

3. 1 more delete query of your choice from any table: delete branch which assets is lower than 500000

```
DELETE FROM branch
WHERE
    assets < 500000;
```

<u>before</u>

```
# 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'
```

<u>after</u>

```
# branch_name, branch_city, assets
'Brighton', 'Brooklyn', '7100000'
'Downtown', 'Brooklyn', '9000000'
'North Town', 'Rye', '3700000'
'Perryridge', 'Horseneck', '1700000'
'Redwood', 'Palo Alto', '2100000'
'Round Hill', 'Horseneck', '8000000'
```

- Views Queries

1. a view consisting of branches and their customers

```
CREATE VIEW BrandandCustomer AS
    SELECT
        br.branch_name, c.customer_name
    FROM
        branch br
            JOIN
        customer c ON br.branch_city = c.customer_city;
```

```
# branch_name, customer_name
'Brighton', 'Brooks'
'Downtown', 'Brooks'
'North Town', 'Curry'
'Redwood', 'Johnson'
'North Town', 'Smith'
```

2. create a view of HQEmployee who work in downtown branch

```
CREATE VIEW HQEmployee AS
    SELECT
        *
    FROM
        employee
    WHERE
        branch_name = 'Downtown';
```

```
# employee_name, branch_name, salary
'Johnson', 'Downtown', '1500'
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
```

3. do one insert, delete, update, and select queries on HQEmployee view
(1) insert: insert Flora with salary 1500

```
insert into HQEmployee (employee_name, branch_name, salary) values ('Flora', 'Downtown', 3000);
```

before

```
# employee_name, branch_name, salary
'Johnson', 'Downtown', '1500'
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
```

after

```
# employee_name, branch_name, salary,
'Flora', 'Downtown', '3000'
'Johnson', 'Downtown', '1500'
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
```

(2) delete: delete employee whose salary is lower than 1500

```
DELETE FROM HQEmployee
WHERE
    salary < 1500;
```

before

```
# employee_name, branch_name, salary,
'Flora', 'Downtown', '3000'
'Johnson', 'Downtown', '1500'
```

```
'Loreena', 'Downtown', '1300'
'Peterson', 'Downtown', '2500'
```

after

```
# employee_name, branch_name, salary
'Flora', 'Downtown', '3000'
'Johnson', 'Downtown', '1500'
'Peterson', 'Downtown', '2500'
```

(3) update: add 50 to each employee’s salary

```
UPDATE HQEmployee
SET
    salary = salary + 50;
```

before

```
# employee_name, branch_name, salary
'Flora', 'Downtown', '3000'
'Johnson', 'Downtown', '1500'
'Peterson', 'Downtown', '2500'
```

after

```
# employee_name, branch_name, salary
'Flora', 'Downtown', '3050'
'Johnson', 'Downtown', '1550'
'Peterson', 'Downtown', '2550'
```

(4) select: show average salary of HQEmployee

```
SELECT
    round(avg(salary), 2) AS avg_salary
FROM
    HQEmployee;
```

```
# avg_salary
'2383.33'
```

- Complex Queries

1. 1 select query involving 3 tables: branch city's average assets where customer account balance is more than 500

```
SELECT
    br.branch_city, AVG(assets) AS avg_assets
FROM
```

```
    customer c
        JOIN
    branch br ON c.customer_city = br.branch_city
        JOIN
    account a ON br.branch_name = a.branch_name
WHERE
    a.balance > 500
GROUP BY br.branch_city;
```

```
# branch_city, avg_assets
'Brooklyn', '7100000'
'Palo Alto', '2100000'
```

2. 1 Delete query involving 3 tables: delete customer from Rye and amount < 1000

```
DELETE c FROM customer c
        JOIN
    borrower bo ON c.customer_name = bo.customer_name
        JOIN
    loan l ON bo.loan_number = l.loan_number
WHERE
    c.customer_city = 'Rye'
    AND amount < 1000;
```

before

```
# 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'
```

after

```
# customer_name, customer_street, customer_city
'Adams', 'Spring', 'Pittsfield'
'Brooks', 'Senator', 'Brooklyn'
'Glenn', 'Sand Hill', 'Woodside'
```

```
'Green', 'Walnut', 'Stamford'
'Hayes', 'Main', 'Harrison'
'Johnson', 'Alma', 'Palo Alto'
'Jones', 'Main', 'Harrison'
'Lindsay', 'Park', 'Pittsfield'
'Turner', 'Putnam', 'Stamford'
'Williams', 'Nassau', 'Princeton'
```

3. Update query involving 3 tables: revise Smith's street information to South and add 50 to each of his or her loan

```
UPDATE customer c
        JOIN
    borrower bo ON c.customer_name = bo.customer_name
        JOIN
    loan l ON bo.loan_number = l.loan_number
SET
    c.customer_street = 'South',
    l.amount = l.amount + 50
WHERE
    c.customer_name = 'Smith';
```

before

```
# customer_name, customer_street, loan_number, amount
'Smith', 'North', 'L-11', '900'
'Smith', 'North', 'L-23', '2000'
```

after

```
# customer_name, customer_street, loan_number, amount
'Smith', 'South', 'L-11', '950'
'Smith', 'South', 'L-23', '2050'
```