EE488G Database and Big Data Systems, Spring 2019
HW1

Name: 김준범
Student ID: 20180155
Discussion Group (People with whom you discussed ideas used in your answers): None
On-line or hardcopy documents used as part of your answers: None

Answer to Problem 1

(1) First name, last name, income of customers whose income is within [30,000, 80,000], order by income (desc), lastName, firstName.

```
SELECT firstname, lastname, income
FROM Customer
WHERE income >= 30000 AND income <= 80000
ORDER BY income DESC LIMIT 10;
```

(2) SIN, branch name, salary and manager's salary - salary (that is, the salary of the employee's manager minus salary of the employee) of all employees in London or New York, order by ascending (manager' salary - salary).

```
SELECT E.sin, B.branchName, E.salary, M.salary - E.salary AS salarygap
FROM Employee M, Employee E, Branch B
WHERE (B.branchName = 'London' OR B.branchName = 'New York')
AND B.managerSIN = M.sin
AND E.branchNumber = B.branchNumber
ORDER BY salarygap ASC LIMIT 10;
```

(3) First name, last name, and income of customers whose income is at least twice the income of every customer whose lastName is Butler, order by last name then first name.

```
SELECT firstName, lastName, income
FROM Customer
WHERE income > ALL (
    SELECT 2*income
    FROM Customer
```

```
    WHERE lastName = 'Butler'
) ORDER BY lastName, firstName ASC LIMIT 10 ;
```

(4) Customer ID, income, account numbers and branch numbers of customers with income greater than 70,000 who own an account at both London and Latveria branches, order by customer ID then account number. The result should contain all the account numbers of customers who meet the criteria, even if the account itself is not held at London or Latveria.

```
SELECT C.customerID, C.income, A.accNumber, B.branchNumber
FROM Customer C, Account A, Owns O, Branch B
WHERE C.customerID IN (
    SELECT C.customerID
    FROM Customer C, Account A1, Account A2, Owns O1,
            Owns O2, Branch B1, Branch B2
    WHERE C.customerID = O1.customerID
    AND A1.accNumber = O1.accNumber
    AND A1.branchNumber = B1.branchNumber
    AND B1.branchName = 'London'
    AND C.customerID = O2.customerID
    AND A2.accNumber = O2.accNumber
    AND A2.branchNumber = B2.branchNumber
    AND B2.branchName = 'Latveria')
AND C.customerID = O.customerID
AND A.accNumber = O.accNumber
AND A.branchNumber = B.branchNumber
AND C.income > 70000
ORDER BY C.customerID, A.accNumber ASC LIMIT 10 ;
```

(5) Customer ID, types, account numbers and balances of business (type BUS) and savings (type SAV) accounts owned by customers who own at least one business account or at least one savings account, order by customer ID, then type, then account number.

```
SELECT C.customerId, A.type, A.accNumber, A.balance
FROM Customer C, Account A, Owns O
WHERE C.customerId IN (
    SELECT DISTINCT C.customerId
    FROM Customer C, Account A, Owns O
    WHERE A.accNumber IN (
```

```
            SELECT accNumber
            FROM Account
            WHERE type = 'BUS' OR type = 'SAV' )
        AND A.accNumber = O.accNumber
        AND C.customerID = O.customerId )
    AND C.customerID = O.customerId
    AND A.accNumber = O.accNumber
    AND ( A.type = 'BUS' OR A.type = 'SAV' )
    ORDER BY C.customerID, A.type, A.accNumber ASC LIMIT 10;
```

(6) Branch name, account number and balance of accounts with balances greater than $80,000 held at the branch managed by Phillip Edwards, order by account number.

```
    SELECT B.branchName, A.accNumber, A.balance
    FROM Account A, Branch B
    WHERE A.accNumber IN (
        SELECT A.accNumber
        FROM Account A, Branch B, Employee E
        WHERE A.branchNumber = B.branchNumber
        AND B.managerSin = E.sin
        AND E.firstName = 'Phillip'
        AND E.lastName = 'Edwards' )
    AND A.balance > 80000
    AND B.branchNumber = A.branchNumber
    ORDER BY A.accNumber ASC LIMIT 10;
```

(7) Customer ID of customers who have an account at the New York branch, who do not own an account at the London branch and who do not co-own an account with another customer who owns an account at the London branch, order by customer ID. The result should not contain duplicate customer IDs (write the query satisfying all three conditions).

```
SELECT a.customerId
FROM (SELECT DISTINCT C.customerId
    FROM Customer C, Owns O, Account A, Branch B
    WHERE C.customerId = O.customerId
    AND A.accNumber = O.accNumber
    AND A.branchNumber = B.branchNumber
    AND B.branchName = 'New York') a
```

```
 LEFT JOIN
    (SELECT C.customerId
    FROM Customer C, Owns O, Account A
    WHERE A.accNumber in
        (SELECT A.accNumber
        FROM Customer C, Owns O, Account A
        WHERE C.customerId IN
          (SELECT DISTINCT C.customerId
          FROM Customer C, Owns O, Account A, Branch B
          WHERE C.customerId = O.customerId
          AND A.accNumber = O.accNumber
          AND A.branchNumber = B.branchNumber
          AND B.branchName = 'London')
        AND C.customerId = O.customerID
        AND A.accNumber = O.accNumber)
    AND C.customerId = O.customerID
    AND A.accNumber = O.accNumber) b
ON a.customerId = b.customerId
WHERE b.customerId IS NULL
ORDER BY customerId ASC LIMIT 10;
```

(8) SIN, first name, last name, and salary of employees who earn more than $60,000, if they are managers show the branch name of their branch in a fifth column (which should be NULL/NONE for most employees), order by branch name (desc) and first Name (asc). You must use an outer join in your solution (which is the easiest way to do it).

```
    SELECT * FROM
        ( SELECT sin, firstName, lastName, salary, branchName
        FROM Employee , Branch
        WHERE managerSIN = sin
        AND salary > 60000 ) a
        NATURAL RIGHT OUTER JOIN
        (SELECT sin, firstName, lastName, salary
        FROM Employee
        WHERE salary > 60000) b
    ORDER BY branchName DESC, firstName ASC LIMIT 10;

;
```

(9) Exactly as question (8), except that your query cannot include any join operation.

```
SELECT sin, firstName, lastName, salary, (
    SELECT branchName
    FROM Branch
    WHERE managerSIN = sin
) AS branchName
FROM Employee
WHERE salary > 60000
ORDER BY branchName DESC, firstName ASC LIMIT 10;
```

(10) Customer ID, first name, last name and income of customers who have income greater than 6000 and own accounts in ALL of the branches that Helen Morgan owns accounts in, order by income in decreasing order (The output includes Helen Morgan).

```
SELECT C.customerId, C.firstName, C.lastName, C.income
FROM Customer C, Owns O, Account A,
        ( SELECT B.branchNumber
        FROM Branch B, Customer C, Owns O, Account A
        WHERE C.firstName ='Helen'
        AND C.lastName ='Morgan'
        AND C.customerId = O.customerId
        AND O.accNumber = A.accNumber
    AND B.branchNumber = A.branchNumber) AS B
WHERE C.customerId = O.customerId
AND O.accNumber = A.accNumber
AND A.branchNumber = B.branchNumber
AND C.income > 6000
GROUP BY C.customerId
HAVING COUNT(DISTINCT A.branchNumber) >= 3
ORDER BY C.income DESC LIMIT 10;
```

(11) SIN, first name, last name and salary of the lowest paid employee (or employees) of the Berlin branch, order by sin.

```
SELECT E.sin, E.firstName, E.lastName, E.salary
FROM Employee E, Branch B
WHERE E.salary = ( SELECT MIN(E.salary)
    FROM Branch B, Employee E
    WHERE B.branchName = 'Berlin'
    AND B.branchNumber = E.branchNumber)
AND B.branchName = 'Berlin'
AND B.branchNumber = E.branchNumber
ORDER BY E.sin ASC LIMIT 10;
```

(12) Branch name, and the difference of maximum and minimum (salary gap) and average salary of the employees at each branch, order by salary gap.

```
SELECT B.branchName,
    ( SELECT Max(E.salary) - Min(E.salary)
    FROM Employee E
    WHERE E.branchNumber = B.branchNumber ) AS salaryGap,
    ( SELECT Avg(E.salary)
    FROM Employee E
    WHERE E.branchNumber = B.branchNumber ) AS averageSalary
FROM Branch B
ORDER BY salaryGap ASC LIMIT 10;
```

(13) Count of the number of employees working at the New York branch and Count of the number of different last names of employees working at the New York branch (two numbers in a single row).

```
SELECT
    ( SELECT COUNT(*)
    FROM Employee E
    WHERE E.branchNumber = B.branchNumber
    ) AS numEmployee,
    ( SELECT COUNT(DISTINCT E.lastName)
    FROM Employee E
    WHERE E.branchNumber = B.branchNumber
    ) AS numDiffLastName
 FROM Branch B
```

```
WHERE B.branchName = 'New York' LIMIT 10;
```

(14) Sum of the employee salaries (a single number) at the Berlin branch.

```
SELECT SUM(E.salary)
FROM Employee E, Branch B
WHERE B.branchName = 'Berlin'
AND E.branchNumber = B.branchNumber LIMIT 10;
```

(15) Customer ID, first name and last name of customers who own accounts from three different branches (only three different types of branches), order by Last Name and first Name.

```
SELECT C.customerId, C.firstName, C.lastName
FROM Customer C
WHERE 3 =
    (SELECT COUNT(DISTINCT B.branchNumber)
    FROM Account A, Owns O, Branch B
    WHERE O.customerId = C.customerId
    AND A.accNumber = O.accNumber
    AND B.branchNumber= A.branchNumber)
ORDER BY C.lastName, C.firstName ASC LIMIT 10;
```

(16) Average income of customers older than 58 and average income of customers younger than 28, the result must have two named columns, with one row, in one result set

```
SELECT O.oldIncome, Y.youngIncome
FROM ( SELECT AVG(C.income) AS oldIncome
    FROM Customer C
    WHERE TIMESTAMPDIFF(YEAR, C.birthdata, CURDATE()) >58 ) O ,
    ( SELECT AVG(C.income) AS youngIncome
    FROM Customer C
    WHERE TIMESTAMPDIFF(YEAR, C.birthdata, CURDATE())<28 ) Y ;
```

(17) Customer ID, first name, income, and average account balance of customers who have at least three accounts, and whose last names begin with S and contain an e (e.g. Steve), order by customer ID.

```
SELECT C.customerId, C.firstName, C.income, (
```

```
        SELECT AVG(A.balance)
        FROM Account A, Owns O
        WHERE C.customerId = O.customerId
        AND A.accNumber = O.accNumber
    ) AS averageBalance
    FROM Customer C
    WHERE 3 <=
        (SELECT COUNT(A.accNumber)
        FROM Account A, Owns O
        WHERE O.customerId = C.customerId
        AND A.accNumber = O.accNumber)
    AND C.lastName LIKE 's%'
    AND C.lastName LIKE '%e%'
    ORDER BY C.customerID ASC LIMIT 10;
```

(18) Account number, balance, sum of transaction amounts, and balance – transaction sum (sum of transaction amount) for accounts in the London branch that have at least 14 transactions, order by transaction sum.

```
    SELECT A.accNumber, A.balance, (
        SELECT SUM(T.amount)
        FROM Transactions T
        WHERE T.accNumber = A.accNumber
    ) AS sumTransAmount, A.balance - (
        SELECT SUM(T.amount)
        FROM Transactions T
        WHERE T.accNumber = A.accNumber
    ) AS balMinusTranSum
    FROM Account A, Branch B
    WHERE B.branchName = 'London'
    AND A.branchNumber = B.branchNumber
    AND 14 <= ( SELECT COUNT(*)
       FROM Transactions T
       WHERE T.accNumber = A.accNumber)
    ORDER BY sumTransAmount ASC LIMIT 10;
```

(19) Branch name, account type, and average transaction amount of each account type for each branch for branches that have at least 40 accounts of any type, order by branch name, then

account type.

```
SELECT T.branchName, T.type, AVG(T.avgTransaction)
FROM
   ( SELECT B.branchName, A.type, (
        SELECT AVG(T.amount)
        FROM Transactions T
        WHERE T.accNumber = A.accNumber
   ) AS avgTransaction
   FROM Branch B, Account A
   WHERE 40 <= ( SELECT COUNT(*)
        FROM Account A
        WHERE A.branchNumber = B.branchNumber )
   AND A.branchNumber = B.branchNumber ) T
GROUP BY T.branchName, T.type
ORDER BY T.branchName, T.type ASC LIMIT 10;
```

(20) Account type, account number, transaction number and amount of transactions of accounts where the average transaction amount is greater than two-and-half times the (overall) average transaction amount of accounts of that type. For example, if the average transaction amount of all business accounts is $2,000 then return transactions from business accounts where the average transaction amount for that account is greater than $5,000. Order by account type, then account number and finally transaction number. Note that all transactions of qualifying accounts should be returned even if they are less than the average amount of the account type.

```
SELECT A.type, A.accNumber, T.transNumber, T.amount
FROM Account A, Transactions T
WHERE
     (SELECT AVG(T.amount)
     FROM Transactions T
     WHERE A.accNumber = T.accNumber)
     >
     (SELECT 2.5*AVG(T.amount)
     FROM Transactions T, Account A2
     WHERE A2.type = A.type
     AND A2.accNumber = T.accNumber)
 AND A.accNumber = T.accNumber
ORDER BY A.type, A.accNumber, T.transNumber ASC LIMIT 10;
```