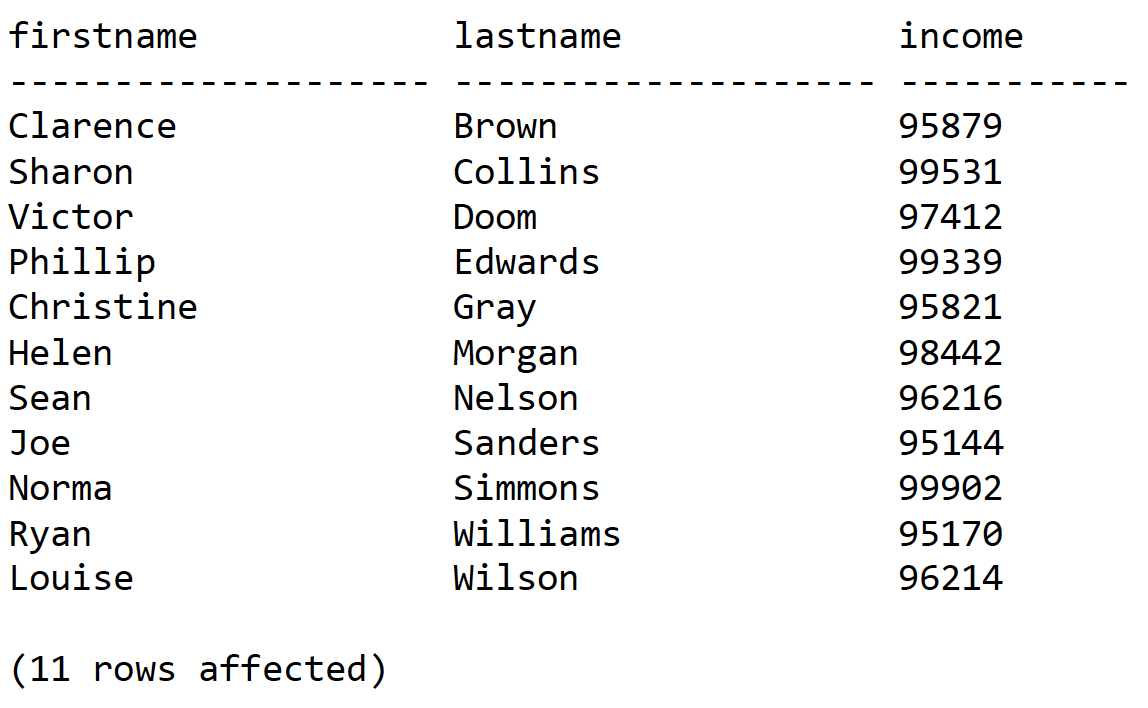
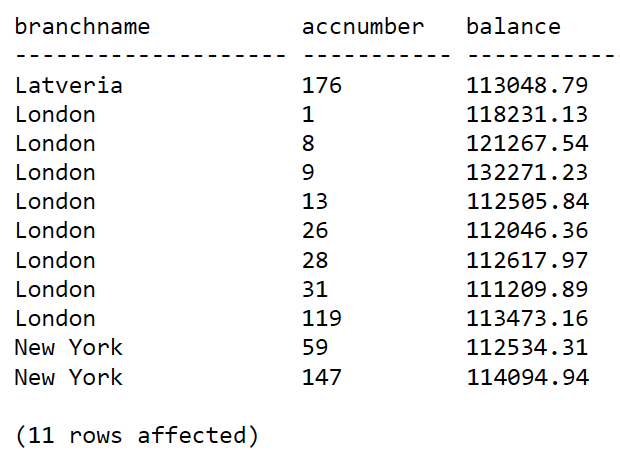
CMPT 354 Assignment 3

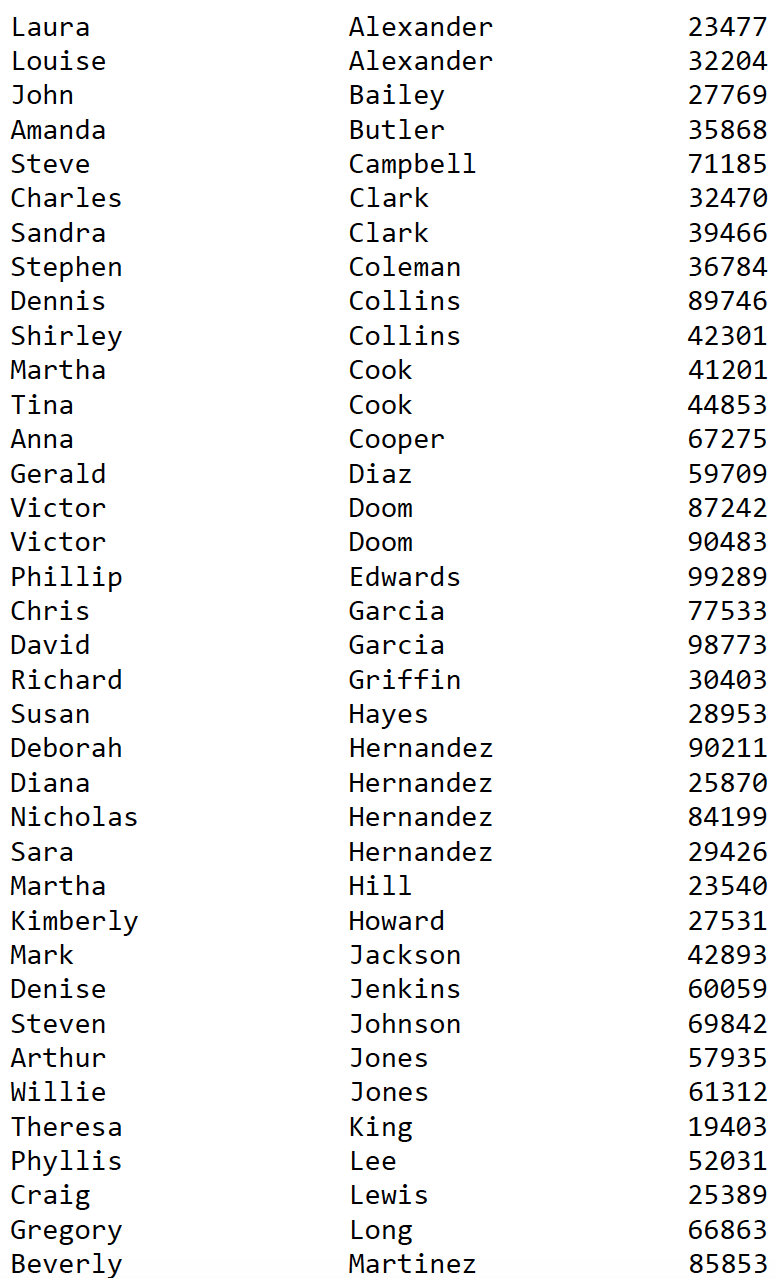
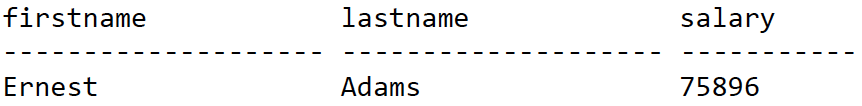
Junchen Li

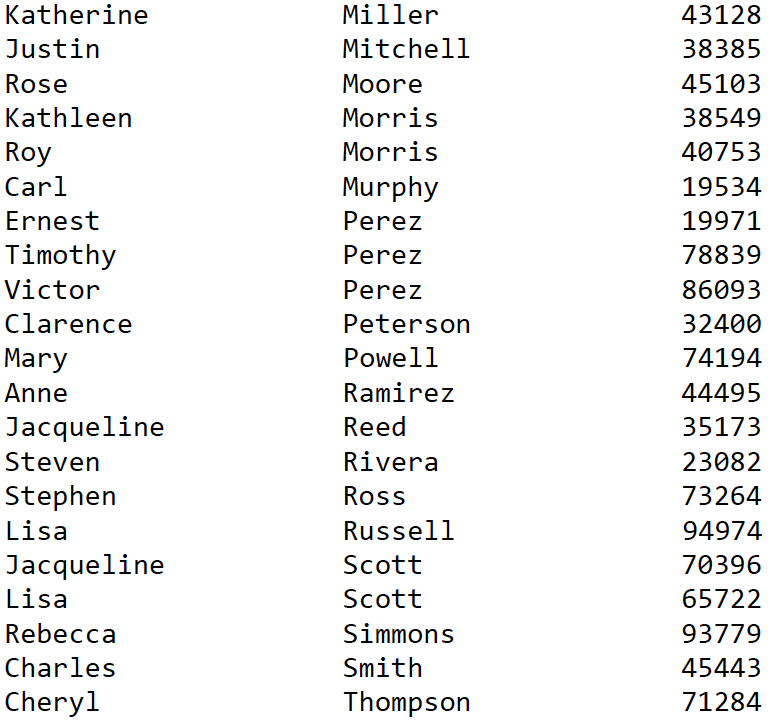
301385486

2021/7/5

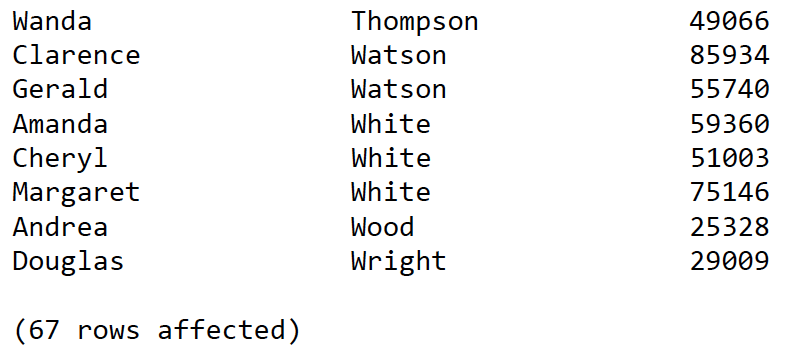
1. *First name*, *last name and income*of customers whose income is over $95,000, order by last name, then first name.  
     
   SELECT firstName, lastName, income  
   FROM Customer  
   WHERE income > 95000  
   ORDER BY lastName, firstName

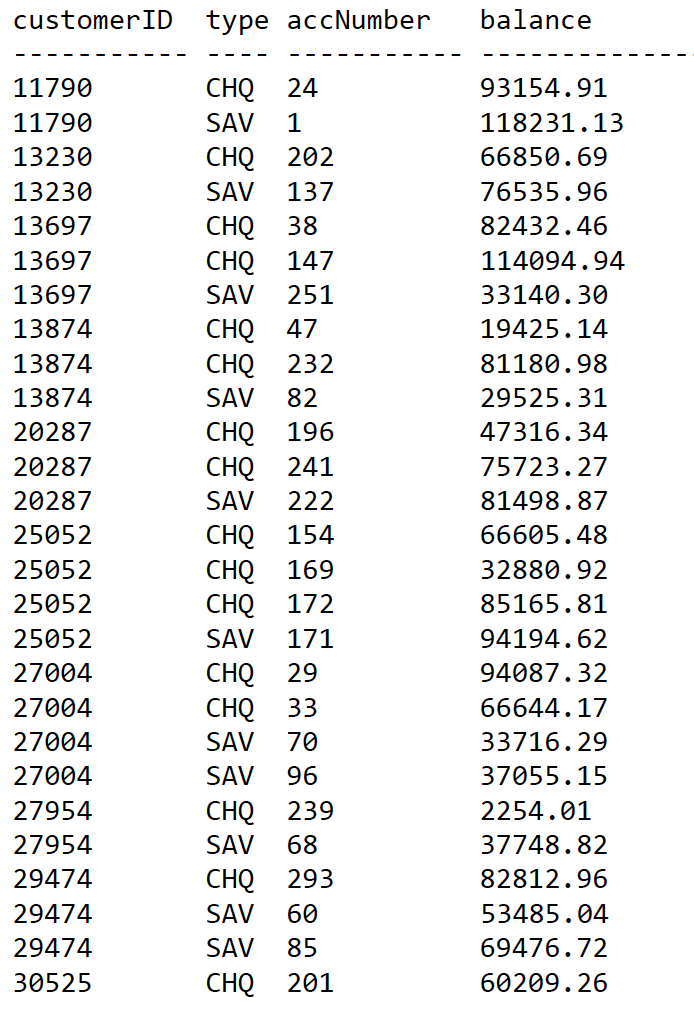
  
*2. Branch name, account number* and *balance*of accounts with balances over $110,000 held at branches with budgets greater than $2,000,000, order by branch name, then account number.  
   
 SELECT B.branchName, A.accNumber, A.balance  
 FROM Branch B, Account A  
 WHERE A.branchNumber = B.branchNumber AND A.balance > 110000 AND   
 B. budget > 2000000  
 ORDER BY B.branchName, A.accNumber

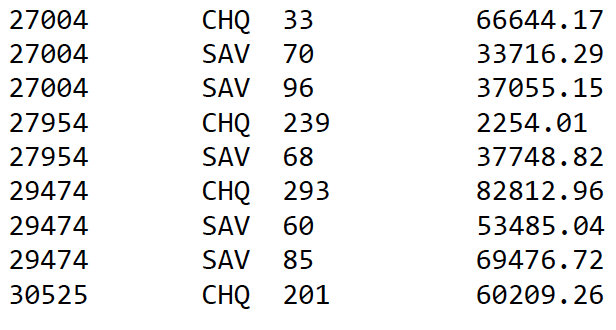
1. *First name*, *last name*, and *salary* of employees whose salary is at least twice the salary of any employee named *Victor Doom*, order by last name then first name.  
     
   SELECT F.firstName, F.lastName, F.salary  
   FROM Employee F, Employee E  
   WHERE E.firstName = ‘Victor’ AND E.lastName = ‘Doom’ AND F.salary > E.salary \* 2  
   ORDER BY lastName, firstName

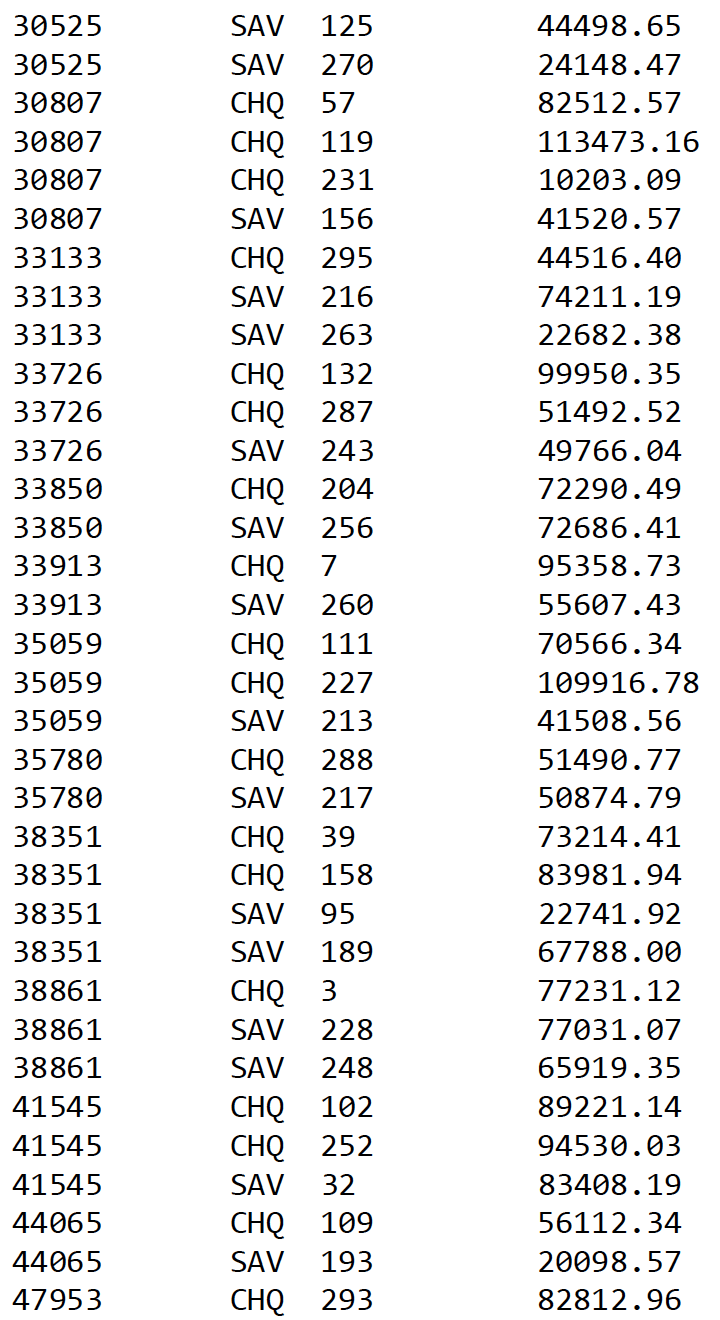
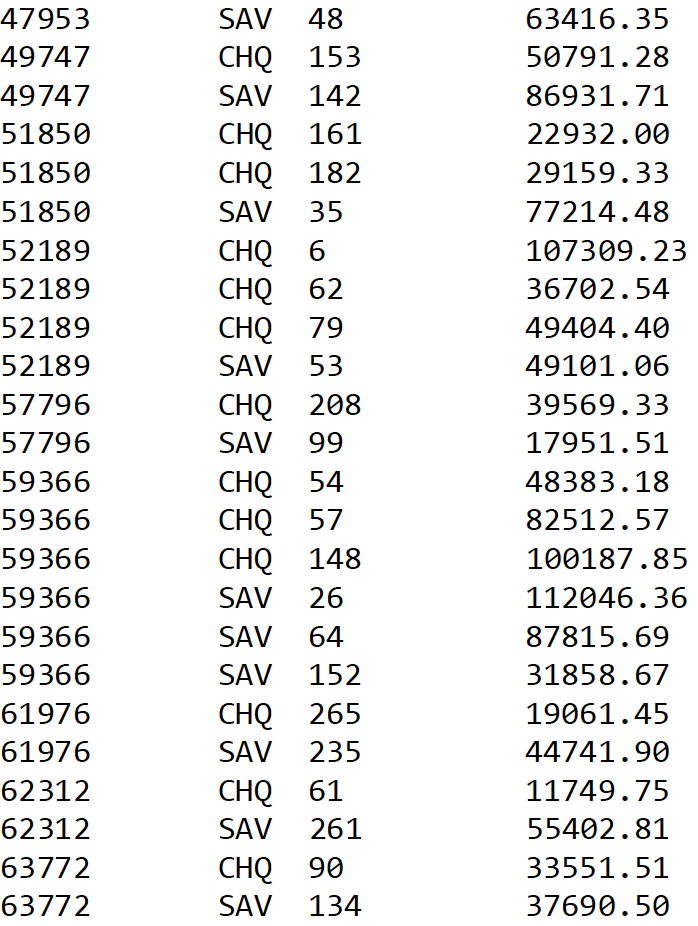


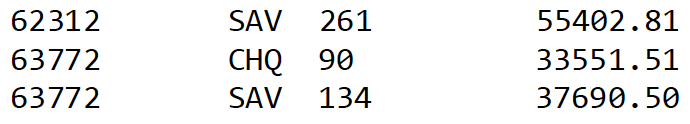
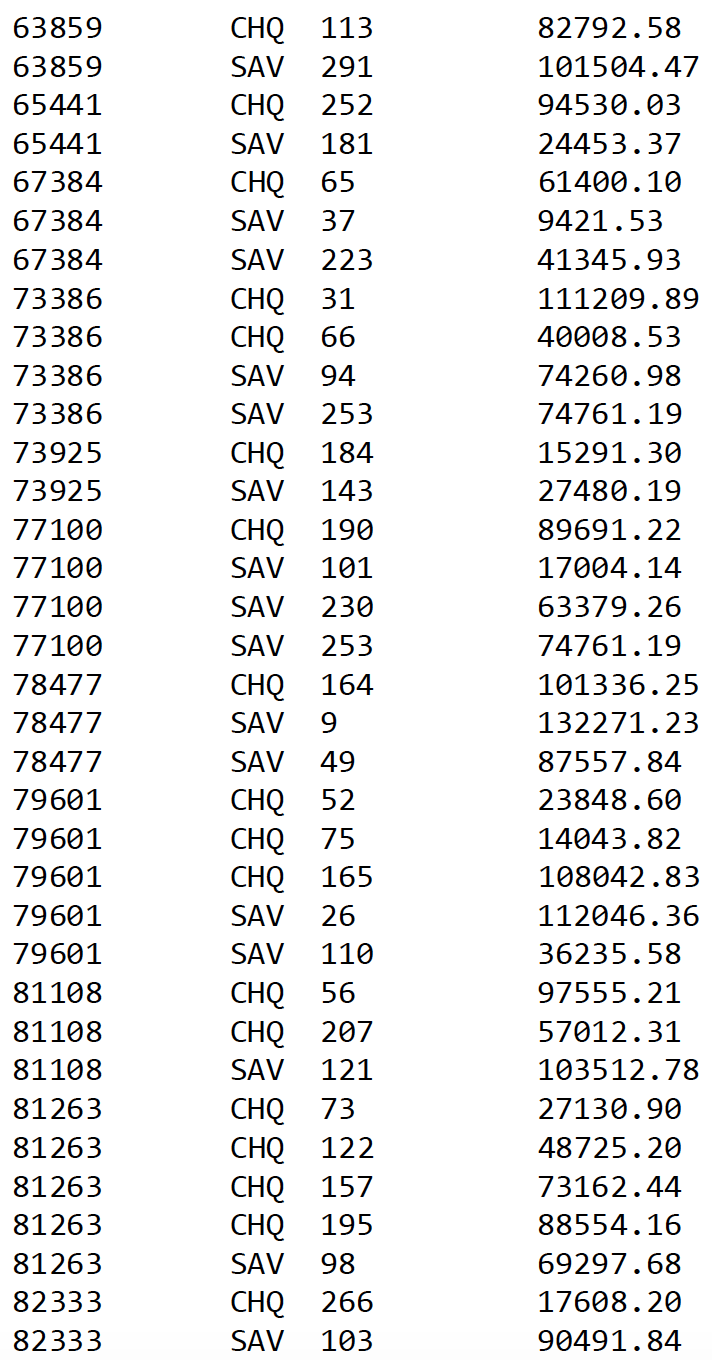
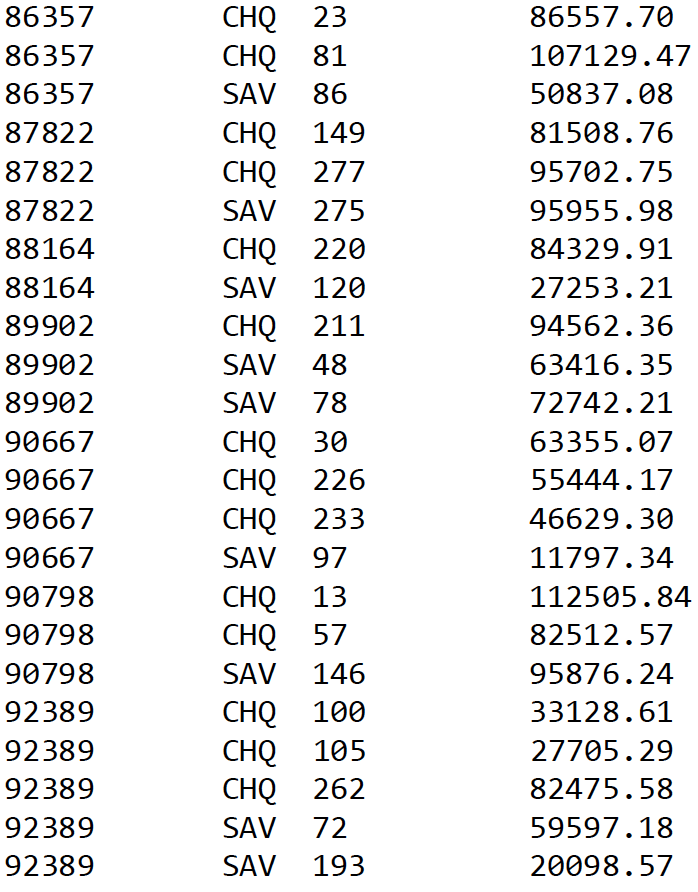
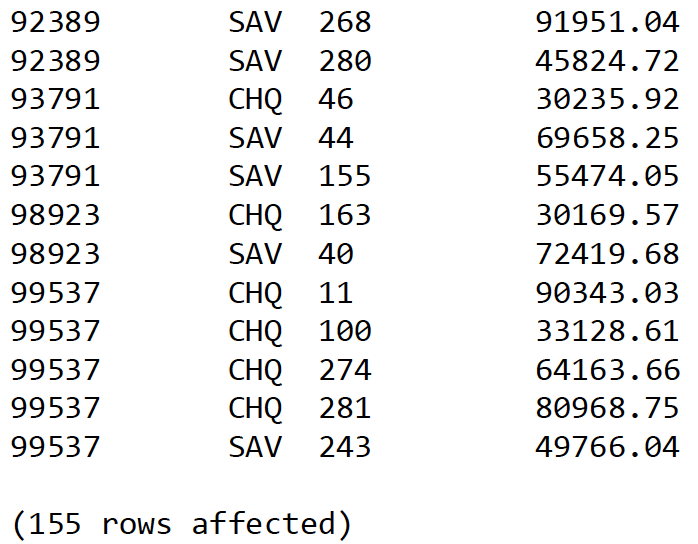


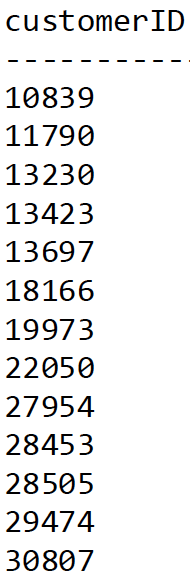
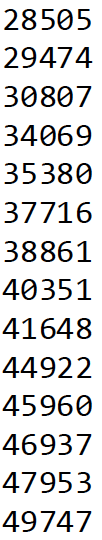
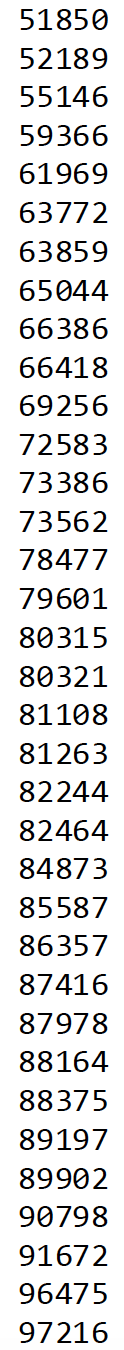
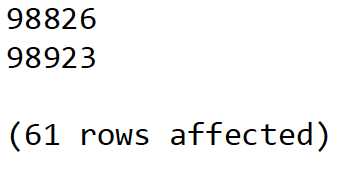


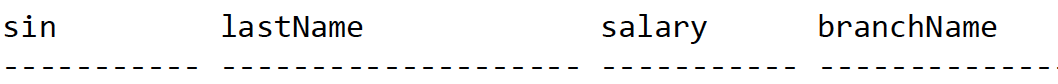
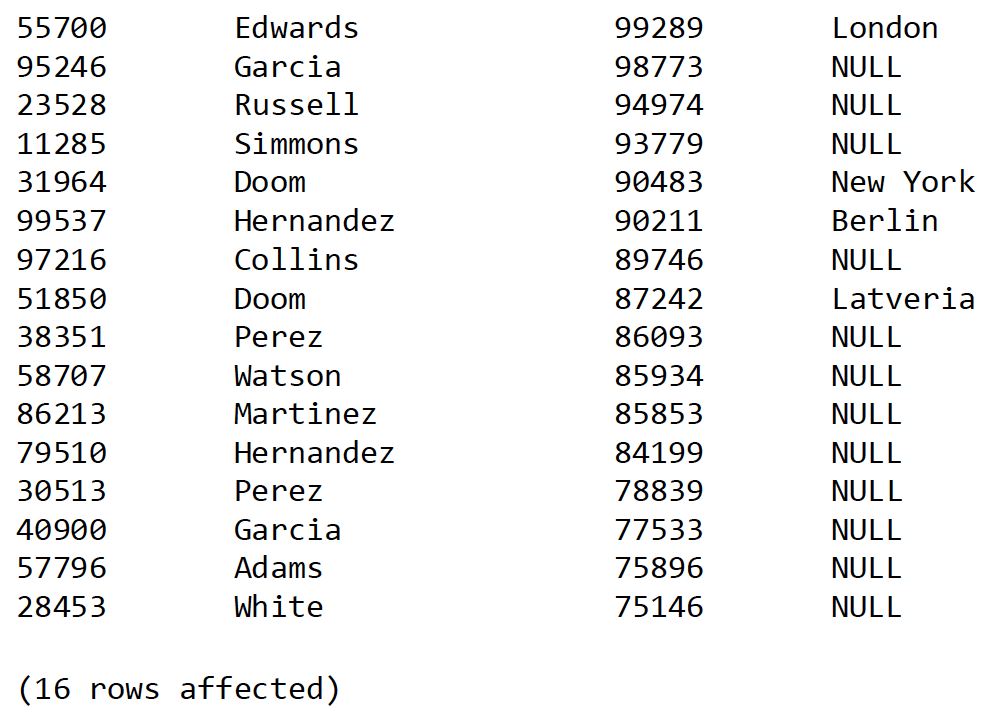
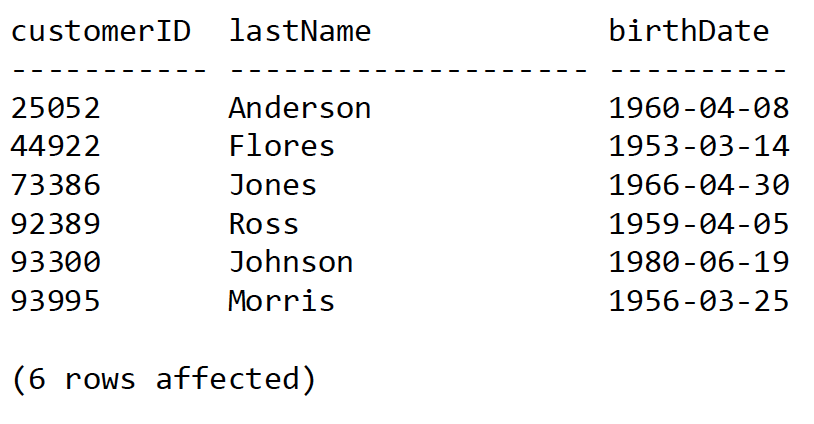
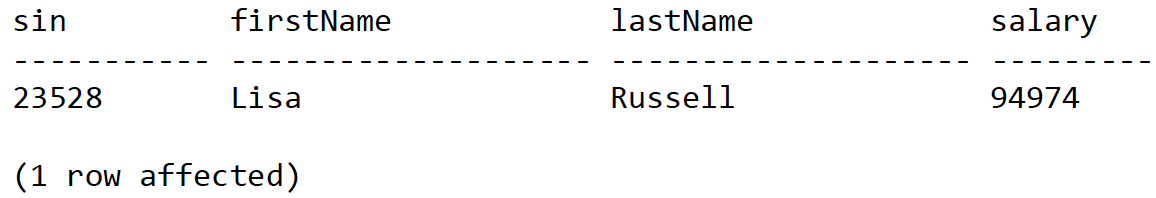
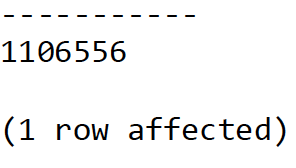
1. *Customer ID*, *types*, *account number*s and *balances*of chequing (type *chq*) and savings (type *sav*) accounts owned by customers who own at least one chequing account and at least one savings account, order by customer ID, then type, then account number.  
     
   SELECT O.customerID, type O.accNumber, balance  
   FROM (Owns O JOIN (SELECT DISTINCT customerID   
   FROM Owns O1 JOIN Account A1 ON O1.accNumber = A1.accNumber WHERE type = ‘chq’ AND customerID in (SELECT customerID   
   FROM Owns O2 JOIN Account A2 ON O2.accNumber = A2.accNumber   
   WHERE type = ‘sav’)) AS temp\_table ON O.customerID = temp\_table.customerID) JOIN Account A3 ON O.accNumber = A3.accNumber WHERE type = ‘chq’ OR type = ‘sav’   
   ORDER BY O.CustomerID, type, O.accNumber





1. *Customer ID*of customers who have an account at the *London*branch, who do ***not***own an account at the *Moscow*branch and who do ***not****own an account* with another customer who owns an account at the *Moscow* branch, order by customer ID. The result *should****not contain duplicate****customer ID*s.  
     
   SELECT DISTINCT O.customerID  
   FROM Owns O, Account A, Branch B  
   WHERE O.accNumber = A.accNumber   
   AND A.branchNumber = B.branchNumber AND B.branchName = ‘London’ AND O.customerID not in   
   (SELECT temp1.customerID   
   FROM Owns temp1, Owns temp2  
   WHERE temp1.accNumber = temp2.accNumber AND temp2.customerID in  
   (SELECT MO.customerID   
   FROM Owns MO, Account MOA, Branch MOB  
   WHERE MOB.branchName = ‘Moscow’ AND MO.accNumber = MOA.accNumber AND MOA.branchNumber = MOB.branchNumber) )  
   ORDER BY customerID  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   *6. SIN*, *last name*, and *salary*of employees who earn more than $75,000, if they are managers show the *branch name* of their branch in a fourth column (which should be NULL for most employees), order by salary in ***decreasing*** order. You must use an outer join in your solution (which is the easiest way to do it).

SELECT E.sin, E.lastName, E.salary, B.branchName  
FROM Employee E LEFT OUTER JOIN Branch B on E.sin = B.managerSIN  
WHERE E.salary > 75000  
ORDER BY E.salary DECS  
  
  
  
  
  
  
  
  
  
  
  
  
  
7. *Customer ID, last name and birth dates of customers* who own accounts in all the branches that Jack Anderson owns accounts in, order by customer ID.  
  
SELECT C.customerID, C.lastName, C.birthDate  
FROM Customer C   
WHERE NOT EXISTS(   
(SELECT DISTINCT A.branchNumber   
FROM Customer C, Owns O, Account A   
WHERE C.firstName = ‘Jack’ AND lastName = ‘Anderson’   
AND C.customerID = O.customerID AND O.accNumber = A.accNumber)  
EXCEPT(SELECT DISTINCT AC.branchNumber  
FROM Owns OW, Account AC   
WHERE OW.Account = AC.accNumber AND OW.customerID = C.customerID) )  
ORDER BY C.customerID  
  
  
  
  
  
  
  
  
  
8. *SIN, first name, last name and salary* of the highest paid employee (or employees) of the New York branch, order by sin.  
  
SELECT E.sin, E.firstName. E.lastName, E.salary  
FROM Employee E, Branch B  
WHERE B.branchName = ‘NEW YORK’ AND E.branchNumber = B.branchNumber AND E.salary = (SELECT max(salary)   
FROM Employee E1, Branch B1  
WHERE B1.branchName = ‘New York’ AND E1.branchNumber = B1.branchNumber)  
ORDER BY E.sin  
  
  
  
  
9. *Sum* of the employee salaries (a single number) at the London branch  
  
SELECT sum(E.salary)  
FROM Branch B, Employee E  
WHERE B.branchNumber = E.branchNumber   
AND B.branchName = ‘London’  
  
  
  
  
  
10. *Count* of the number of different first names of employees working at the Latveria branch and a count of the number of employees working at the Latveria branch (two numbers in a single row).  
  
SELECT count(DISTINCT E.firstName) AS count\_firstName,   
count(E.sin) AS count\_employees  
FROM Branch B, Employee E  
WHERE B.branchName = ‘Latveria’ AND E.branchNumber = B.branchNumber  
  
count\_firstName count\_employees

--------------- ---------------

12 13

(1 row affected)  
  
11. *Branch name, and minimum, maximum and average salary* of the employees at each branch, order by branch name.  
  
SELECT B.branchName, min(E.salary) AS min\_salary, max(E.salary) AS max\_salary, avg(E.salary) AS average\_salary  
FROM Branch B, Employee E  
WHERE B.branchNumber = E.branchNumber  
GROUP BY B.branchName  
ORDER BY B.branchName  
  
branchName min\_salary max\_salary average\_salary

---------- ----------- ----------- --------------

Berlin 3349 90211 34714

Latveria 9491 98773 56143

London 13950 99289 50298

Moscow 12525 71284 49065

New York 10953 94974 48649

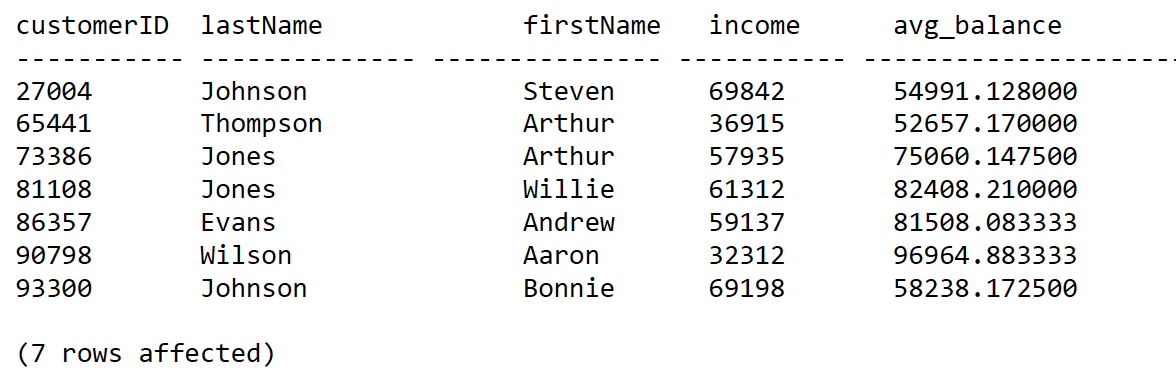
(5 rows affected)  
  
12. *Average income* of customers older than 60 and average income of customers younger than 60, the result must have two named columns, with one row, in one result set (hint: look up T-SQL time and date functions).  
  
SELECT Average\_Older60.inc AS Average\_Older60, Average\_younger60.inc AS Average\_younger60   
FROM (SELECT avg(income) AS inc FROM Customer  
WHERE year(getdate( ) ) – YEAR( birthDate) > 60) AS Average\_Older60,  
(SELECT avg(income) AS inc FROM Customer  
WHERE year(getDate( ) ) – YEAR(birthDate) < 60) AS Average\_younger60  
  
Average\_Older60 Average\_younger60

--------------- -----------------

55256 53090

(1 row affected)

*13. Customer ID, last name, first name, income, and average account balance of customers* who have at least three accounts, and whose last names begin with Jo and contain an s (e.g. Johnson) or whose first names begin with A and have a vowel as the letter just before the last letter (e.g. Aaron), order by customer ID. Note that this will be much easier if you look up LIKE wildcards in the MSDN T-SQL documentation. Also note - to appear in the result customers must have at least three accounts and satisfy one (or both) of the name conditions.

SELECT C.custonerID, C.lastName C.firstName, C.income, avg(A.balance) AS avg\_balance   
FROM Customer C, Own O, Account A  
WHERE O.accNumber = A.accNumber AND C.customerID = O.customerID AND (C.lastName LIKE ‘Jo%s%’ OR C.firstName LIKE ‘A%[aeiou]\_’)  
GROUP BY C.customerID, C.lastName, C.firstName, C.income   
HAVING count(O.accNumber) >= 3  
ORDER BY C.customerID

25052 Anderson Jack 35755 73910.330000

27004 Johnson Steven 69842 54991.128000

73386 Jones Arthur 57935 75060.147500

81108 Jones Willie 61312 82408.210000

89197 Anderson Lawrence 28761 77278.070000

93300 Johnson Bonnie 69198 58238.172500

14. Account number, balance, sum of transaction amounts, and balance - transaction sum for accounts in the London branch that have at least ten transactions, order by account number.

SELECT A.accNumber, A.balance, sum(T.amount) AS sum\_trans, A.balance – sum(T.amount) AS balance\_transaction  
FROM Account A, Branch B, Transactions T  
WHERE A.accNumber = T.accNumber AND B.branchNumber = A.branchNumber AND B.branchName = ‘London’  
GROUP BY A.accNumber, A.balance  
Having count(\*) >= 10  
ORDER BY A.accNumber  
  
  
  
accNumber balance sum\_trans balance\_transaction

----------- -------------- ----------------- ----------------------

1 118231.13 118231.13 0.00

2 100808.03 100808.03 0.00

5 105696.04 105696.04 0.00

8 121267.54 121267.54 0.00

9 132271.23 132271.23 0.00

17 103356.07 103356.07 0.00

19 83432.52 83432.52 0.00

31 111209.89 111209.89 0.00

32 83408.19 83408.19 0.00

33 66644.17 66644.17 0.00

35 77214.48 77214.48 0.00

36 65482.68 65482.68 0.00

39 73214.41 73214.41 0.00

89 97457.14 97457.14 0.00

108 66088.83 66088.83 0.00

110 36235.58 36235.58 0.00

112 31854.76 31854.76 0.00

113 82792.58 82792.58 0.00

114 67973.27 67973.27 0.00

125 44498.65 44498.65 0.00

127 54938.10 54938.10 0.00

130 102776.09 102776.09 0.00

131 65314.36 65314.36 0.00

132 99950.35 99950.35 0.00

135 105420.87 105420.87 0.00

136 32694.57 32694.57 0.00

137 76535.96 76535.96 0.00

139 101394.11 101394.11 0.00

141 93073.14 93073.14 0.00

142 86931.71 86931.71 0.00

144 31521.61 31521.61 0.00

(31 rows affected)