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
```

```
firstname            lastname             income
-------------------- -------------------- -----------
Clarence             Brown                95879
Sharon               Collins              99531
Victor               Doom                 97412
Phillip              Edwards              99339
Christine            Gray                 95821
Helen                Morgan               98442
Sean                 Nelson               96216
Joe                  Sanders              95144
Norma                Simmons              99902
Ryan                 Williams             95170
Louise               Wilson               96214

(11 rows affected)
```

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
```

```
branchname           accnumber   balance
-------------------- ----------- -----------
Latveria             176         113048.79
London               1           118231.13
London               8           121267.54
London               9           132271.23
London               13          112505.84
London               26          112046.36
London               28          112617.97
London               31          111209.89
London               119         113473.16
New York             59          112534.31
New York             147         114094.94

(11 rows affected)
```

3. *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
```

firstname	lastname	salary
Ernest	Adams	75896
Laura	Alexander	23477
Louise	Alexander	32204
John	Bailey	27769
Amanda	Butler	35868
Steve	Campbell	71185
Charles	Clark	32470
Sandra	Clark	39466
Stephen	Coleman	36784
Dennis	Collins	89746
Shirley	Collins	42301
Martha	Cook	41201
Tina	Cook	44853
Anna	Cooper	67275
Gerald	Diaz	59709
Victor	Doom	87242
Victor	Doom	90483
Phillip	Edwards	99289
Chris	Garcia	77533
David	Garcia	98773
Richard	Griffin	30403
Susan	Hayes	28953
Deborah	Hernandez	90211
Diana	Hernandez	25870
Nicholas	Hernandez	84199
Sara	Hernandez	29426
Martha	Hill	23540
Kimberly	Howard	27531
Mark	Jackson	42893
Denise	Jenkins	60059
Steven	Johnson	69842
Arthur	Jones	57935
Katherine	Miller	43128
Justin	Mitchell	38385
Rose	Moore	45103
Kathleen	Morris	38549
Roy	Morris	40753
Carl	Murphy	19534
Ernest	Perez	19971
Timothy	Perez	78839
Victor	Perez	86093
Clarence	Peterson	32400
Mary	Powell	74194
Anne	Ramirez	44495
Jacqueline	Reed	35173
Steven	Rivera	23082
Stephen	Ross	73264

```
Lisa                 Russell              94974
Jacqueline           Scott                70396
Lisa                 Scott                65722
Rebecca              Simmons              93779
Charles              Smith                45443
Cheryl               Thompson             71284
Wanda                Thompson             49066
Clarence             Watson               85934
Gerald               Watson               55740
Amanda               White                59360
Cheryl               White                51003
Margaret             White                75146
Andrea               Wood                 25328
Douglas              Wright               29009

(67 rows affected)
```

4. *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
```

```
customerID  type accNumber   balance
----------- ---- ----------- -------------
11790       CHQ  24          93154.91
11790       SAV  1           118231.13
13230       CHQ  202         66850.69
13230       SAV  137         76535.96
13697       CHQ  38          82432.46
13697       CHQ  147         114094.94
13697       SAV  251         33140.30
13874       CHQ  47          19425.14
13874       CHQ  232         81180.98
13874       SAV  82          29525.31
20287       CHQ  196         47316.34
20287       CHQ  241         75723.27
20287       SAV  222         81498.87
25052       CHQ  154         66605.48
25052       CHQ  169         32880.92
25052       CHQ  172         85165.81
25052       SAV  171         94194.62
27004       CHQ  29          94087.32
```

27004	CHQ	33	66644.17
27004	SAV	70	33716.29
27004	SAV	96	37055.15
27954	CHQ	239	2254.01
27954	SAV	68	37748.82
29474	CHQ	293	82812.96
29474	SAV	60	53485.04
29474	SAV	85	69476.72
30525	CHO	201	60209.26
30525	SAV	125	44498.65
30525	SAV	270	24148.47
30807	CHQ	57	82512.57
30807	CHQ	119	113473.16
30807	CHQ	231	10203.09
30807	SAV	156	41520.57
33133	CHQ	295	44516.40
33133	SAV	216	74211.19
33133	SAV	263	22682.38
33726	CHQ	132	99950.35
33726	CHQ	287	51492.52
33726	SAV	243	49766.04
33850	CHQ	204	72290.49
33850	SAV	256	72686.41
33913	CHQ	7	95358.73
33913	SAV	260	55607.43
35059	CHQ	111	70566.34
35059	CHQ	227	109916.78
35059	SAV	213	41508.56
35780	CHQ	288	51490.77
35780	SAV	217	50874.79
38351	CHQ	39	73214.41
38351	CHQ	158	83981.94
38351	SAV	95	22741.92
38351	SAV	189	67788.00
38861	CHQ	3	77231.12
38861	SAV	228	77031.07
38861	SAV	248	65919.35
41545	CHQ	102	89221.14
41545	CHQ	252	94530.03
41545	SAV	32	83408.19
44065	CHQ	109	56112.34
44065	SAV	193	20098.57
47953	CHQ	293	82812.96
47953	SAV	48	63416.35
49747	CHQ	153	50791.28
49747	SAV	142	86931.71
51850	CHQ	161	22932.00
51850	CHQ	182	29159.33
51850	SAV	35	77214.48
52189	CHQ	6	107309.23
52189	CHQ	62	36702.54
52189	CHQ	79	49404.40
52189	SAV	53	49101.06
57796	CHQ	208	39569.33
57796	SAV	99	17951.51
59366	CHQ	54	48383.18
59366	CHQ	57	82512.57
59366	CHQ	148	100187.85
59366	SAV	26	112046.36
59366	SAV	64	87815.69
59366	SAV	152	31858.67
61976	CHQ	265	19061.45
61976	SAV	235	44741.90
62312	CHQ	61	11749.75

62312	SAV	261	55402.81
63772	CHQ	90	33551.51
63772	SAV	134	37690.50
63859	CHQ	113	82792.58
63859	SAV	291	101504.47
65441	CHQ	252	94530.03
65441	SAV	181	24453.37
67384	CHQ	65	61400.10
67384	SAV	37	9421.53
67384	SAV	223	41345.93
73386	CHQ	31	111209.89
73386	CHQ	66	40008.53
73386	SAV	94	74260.98
73386	SAV	253	74761.19
73925	CHQ	184	15291.30
73925	SAV	143	27480.19
77100	CHQ	190	89691.22
77100	SAV	101	17004.14
77100	SAV	230	63379.26
77100	SAV	253	74761.19
78477	CHQ	164	101336.25
78477	SAV	9	132271.23
78477	SAV	49	87557.84
79601	CHQ	52	23848.60
79601	CHQ	75	14043.82
79601	CHQ	165	108042.83
79601	SAV	26	112046.36
79601	SAV	110	36235.58
81108	CHQ	56	97555.21
81108	CHQ	207	57012.31
81108	SAV	121	103512.78
81263	CHQ	73	27130.90
81263	CHQ	122	48725.20
81263	CHQ	157	73162.44
81263	CHQ	195	88554.16
81263	SAV	98	69297.68
82333	CHQ	266	17608.20
82333	SAV	103	90491.84
86357	CHQ	23	86557.70
86357	CHQ	81	107129.47
86357	SAV	86	50837.08
87822	CHQ	149	81508.76
87822	CHQ	277	95702.75
87822	SAV	275	95955.98
88164	CHQ	220	84329.91
88164	SAV	120	27253.21
89902	CHQ	211	94562.36
89902	SAV	48	63416.35
89902	SAV	78	72742.21
90667	CHQ	30	63355.07
90667	CHQ	226	55444.17
90667	CHQ	233	46629.30
90667	SAV	97	11797.34
90798	CHQ	13	112505.84
90798	CHQ	57	82512.57
90798	SAV	146	95876.24
92389	CHQ	100	33128.61
92389	CHQ	105	27705.29
92389	CHQ	262	82475.58
92389	SAV	72	59597.18
92389	SAV	193	20098.57

```
92389       SAV  268         91951.04
92389       SAV  280         45824.72
93791       CHQ  46          30235.92
93791       SAV  44          69658.25
93791       SAV  155         55474.05
98923       CHQ  163         30169.57
98923       SAV  40          72419.68
99537       CHQ  11          90343.03
99537       CHQ  100         33128.61
99537       CHQ  274         64163.66
99537       CHQ  281         80968.75
99537       SAV  243         49766.04

(155 rows affected)
```

5. *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
```

```
customerID
----------
10839
11790
13230
13423
13697
18166
19973
22050
27954
28453
```

```
28505
29474
30807
34069
35380
37716
38861
40351
41648
44922
45960
46937
47953
49747
51850
52189
55146
59366
61969
63772
63859
65044
66386
66418
69256
72583
73386
73562
78477
79601
80315
80321
81108
81263
82244
82464
84873
85587
86357
87416
87978
88164
88375
89197
89902
90798
91672
96475
97216
98826
98923

(61 rows affected)
```

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
```

```
sin         lastName             salary      branchName
----------- -------------------- ----------- -------------
55700       Edwards              99289       London
95246       Garcia               98773       NULL
23528       Russell              94974       NULL
11285       Simmons              93779       NULL
31964       Doom                 90483       New York
99537       Hernandez            90211       Berlin
97216       Collins              89746       NULL
51850       Doom                 87242       Latveria
38351       Perez                86093       NULL
58707       Watson               85934       NULL
86213       Martinez             85853       NULL
79510       Hernandez            84199       NULL
30513       Perez                78839       NULL
40900       Garcia               77533       NULL
57796       Adams                75896       NULL
28453       White                75146       NULL

(16 rows affected)
```

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
```

```
customerID  lastName            birthDate
----------- ------------------- ----------
25052       Anderson            1960-04-08
44922       Flores              1953-03-14
73386       Jones               1966-04-30
92389       Ross                1959-04-05
93300       Johnson             1980-06-19
93995       Morris              1956-03-25

(6 rows affected)
```

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
```

```
sin         firstName           lastName            salary
----------- ------------------- ------------------- ---------
23528       Lisa                Russell             94974

(1 row affected)
```

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'
```

```
-----------
1106556

(1 row affected)
```

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
```

customerID	lastName	firstName	income	avg_balance
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)
```