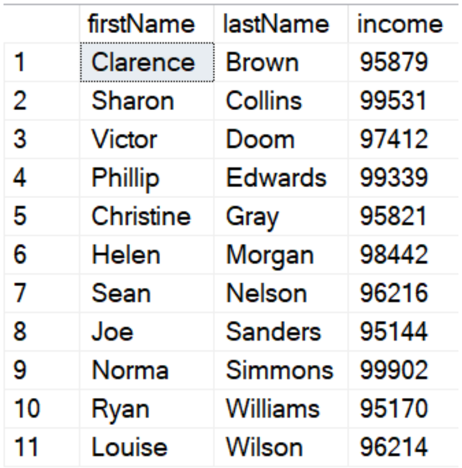
Question 1

First name, *last name and income* of customers whose income is over $95,000, order by last name, then first name.

```
/* Num 1
*/
    SELECT firstName, lastName, income
    FROM Customer
    WHERE income > 95000
    ORDER BY lastName, firstName;
```

OUTPUT:



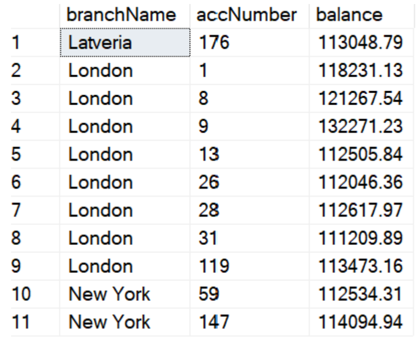
	firstName	lastName	income
1	Clarence	Brown	95879
2	Sharon	Collins	99531
3	Victor	Doom	97412
4	Phillip	Edwards	99339
5	Christine	Gray	95821
6	Helen	Morgan	98442
7	Sean	Nelson	96216
8	Joe	Sanders	95144
9	Norma	Simmons	99902
10	Ryan	Williams	95170
11	Louise	Wilson	96214

Question 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.

```
/* Num 2
*/
SELECT branchName, accNumber,balance
FROM Branch B INNER JOIN Account A ON B.branchNumber=A.branchNumber
WHERE balance > 110000 AND budget>2000000
ORDER BY branchName, accNumber;
```

OUTPUT:



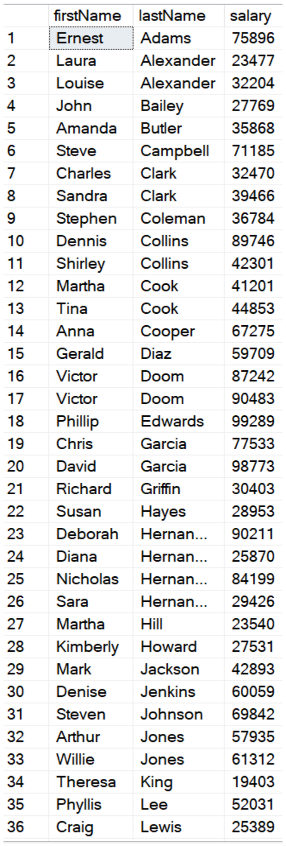
	branchName	accNumber	balance
1	Latveria	176	113048.79
2	London	1	118231.13
3	London	8	121267.54
4	London	9	132271.23
5	London	13	112505.84
6	London	26	112046.36
7	London	28	112617.97
8	London	31	111209.89
9	London	119	113473.16
10	New York	59	112534.31
11	New York	147	114094.94

Question 3

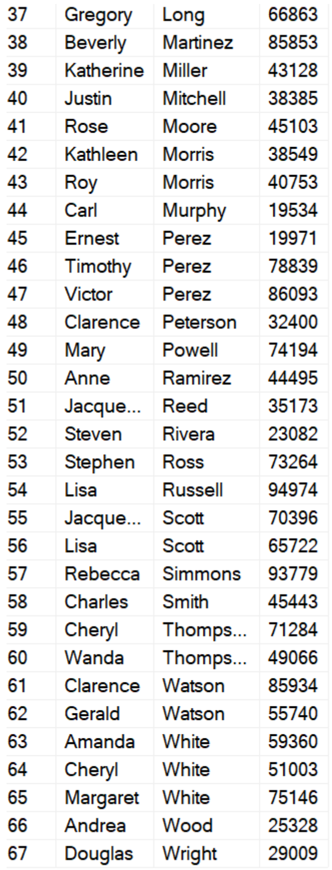
First name, *last name*, and *salary* of employees whose salary is at least twice the salary of any employee named *Victor Doom*, order by last name then first name.

```
/* Num 3
*/
SELECT E1.firstName ,E1.lastName,E1.salary
FROM Employee E1,Employee E2
WHERE E2.firstName='Victor' AND E2.lastName='Doom' AND E1.salary > E2.salary *2
ORDER BY E1.lastName, E1.firstName;
```

OUTPUT:



	firstName	lastName	salary
1	Ernest	Adams	75896
2	Laura	Alexander	23477
3	Louise	Alexander	32204
4	John	Bailey	27769
5	Amanda	Butler	35868
6	Steve	Campbell	71185
7	Charles	Clark	32470
8	Sandra	Clark	39466
9	Stephen	Coleman	36784
10	Dennis	Collins	89746
11	Shirley	Collins	42301
12	Martha	Cook	41201
13	Tina	Cook	44853
14	Anna	Cooper	67275
15	Gerald	Diaz	59709
16	Victor	Doom	87242
17	Victor	Doom	90483
18	Phillip	Edwards	99289
19	Chris	Garcia	77533
20	David	Garcia	98773
21	Richard	Griffin	30403
22	Susan	Hayes	28953
23	Deborah	Hernan...	90211
24	Diana	Hernan...	25870
25	Nicholas	Hernan...	84199
26	Sara	Hernan...	29426
27	Martha	Hill	23540
28	Kimberly	Howard	27531
29	Mark	Jackson	42893
30	Denise	Jenkins	60059
31	Steven	Johnson	69842
32	Arthur	Jones	57935
33	Willie	Jones	61312
34	Theresa	King	19403
35	Phyllis	Lee	52031
36	Craig	Lewis	25389



37	Gregory	Long	66863
38	Beverly	Martinez	85853
39	Katherine	Miller	43128
40	Justin	Mitchell	38385
41	Rose	Moore	45103
42	Kathleen	Morris	38549
43	Roy	Morris	40753
44	Carl	Murphy	19534
45	Ernest	Perez	19971
46	Timothy	Perez	78839
47	Victor	Perez	86093
48	Clarence	Peterson	32400
49	Mary	Powell	74194
50	Anne	Ramirez	44495
51	Jacque...	Reed	35173
52	Steven	Rivera	23082
53	Stephen	Ross	73264
54	Lisa	Russell	94974
55	Jacque...	Scott	70396
56	Lisa	Scott	65722
57	Rebecca	Simmons	93779
58	Charles	Smith	45443
59	Cheryl	Thomps...	71284
60	Wanda	Thomps...	49066
61	Clarence	Watson	85934
62	Gerald	Watson	55740
63	Amanda	White	59360
64	Cheryl	White	51003
65	Margaret	White	75146
66	Andrea	Wood	25328
67	Douglas	Wright	29009

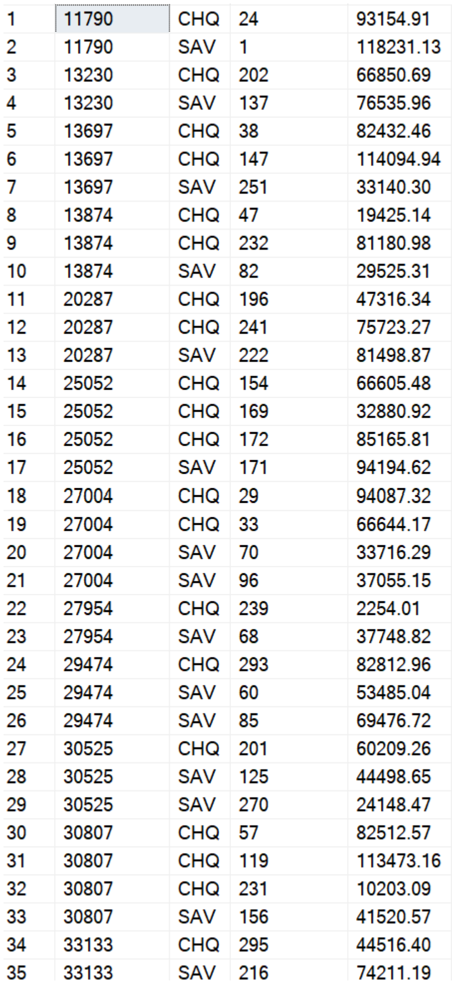
Question 4:

> *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.

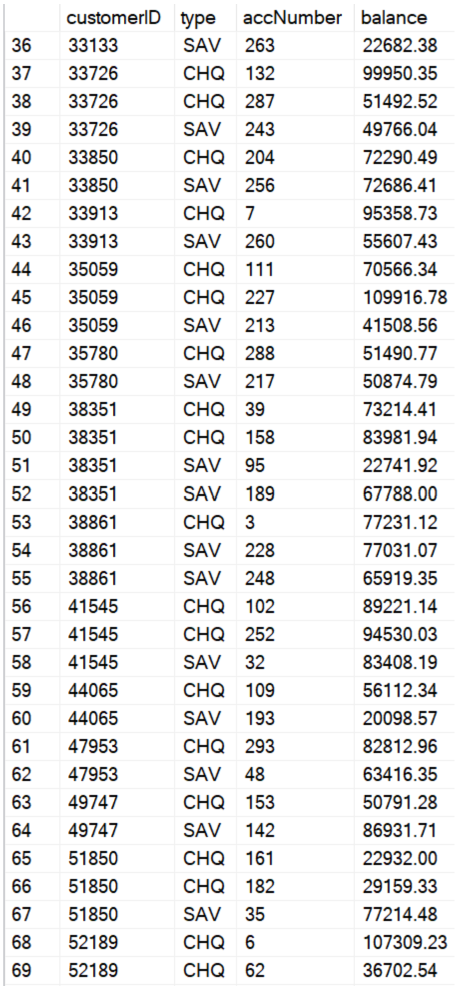
```
 SELECT O1.customerID, A1.type,A1.accNumber, A1.balance
 FROM Owns O1, Account A1
 WHERE O1.accNumber=A1.accNumber AND (A1.type='SAV' OR A1.type='CHQ') AND
O1.customerID IN (
SELECT C1.customerID
FROM Account A1,Customer C1,Owns O1
WHERE A1.accNumber = O1.accNumber AND O1.customerID= C1.customerID AND
A1.type='CHQ'
INTERSECT
SELECT C2.customerID
FROM Account A2,Customer C2,Owns O2
WHERE A2.accNumber = O2.accNumber AND O2.customerID= C2.customerID AND
A2.type='SAV'
)
```

ORDER BY customerID, type, accNumber;

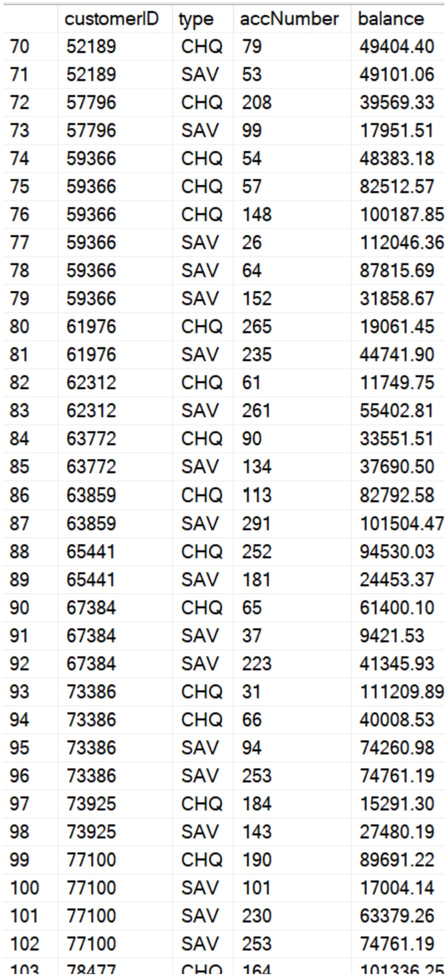
OUTPUT:



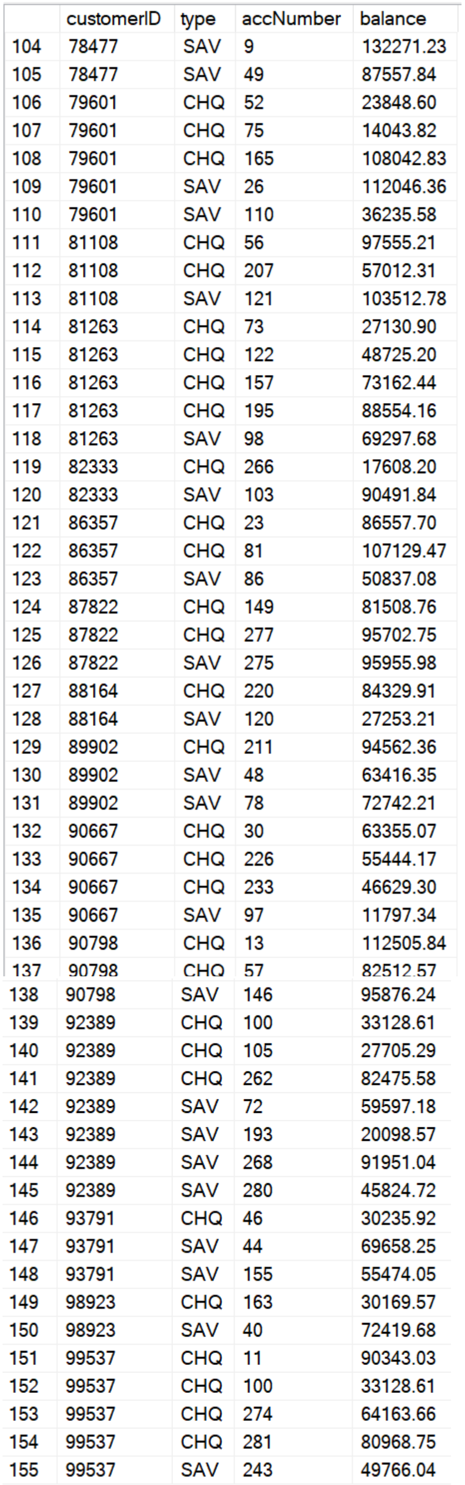
```
1   11790  CHQ  24    93154.91
2   11790  SAV  1     118231.13
3   13230  CHQ  202   66850.69
4   13230  SAV  137   76535.96
5   13697  CHQ  38    82432.46
6   13697  CHQ  147   114094.94
7   13697  SAV  251   33140.30
8   13874  CHQ  47    19425.14
9   13874  CHQ  232   81180.98
10  13874  SAV  82    29525.31
11  20287  CHQ  196   47316.34
12  20287  CHQ  241   75723.27
13  20287  SAV  222   81498.87
14  25052  CHQ  154   66605.48
15  25052  CHQ  169   32880.92
16  25052  CHQ  172   85165.81
17  25052  SAV  171   94194.62
18  27004  CHQ  29    94087.32
19  27004  CHQ  33    66644.17
20  27004  SAV  70    33716.29
21  27004  SAV  96    37055.15
22  27954  CHQ  239   2254.01
23  27954  SAV  68    37748.82
24  29474  CHQ  293   82812.96
25  29474  SAV  60    53485.04
26  29474  SAV  85    69476.72
27  30525  CHQ  201   60209.26
28  30525  SAV  125   44498.65
29  30525  SAV  270   24148.47
30  30807  CHQ  57    82512.57
31  30807  CHQ  119   113473.16
32  30807  CHQ  231   10203.09
33  30807  SAV  156   41520.57
34  33133  CHQ  295   44516.40
35  33133  SAV  216   74211.19
```



	customerID	type	accNumber	balance
36	33133	SAV	263	22682.38
37	33726	CHQ	132	99950.35
38	33726	CHQ	287	51492.52
39	33726	SAV	243	49766.04
40	33850	CHQ	204	72290.49
41	33850	SAV	256	72686.41
42	33913	CHQ	7	95358.73
43	33913	SAV	260	55607.43
44	35059	CHQ	111	70566.34
45	35059	CHQ	227	109916.78
46	35059	SAV	213	41508.56
47	35780	CHQ	288	51490.77
48	35780	SAV	217	50874.79
49	38351	CHQ	39	73214.41
50	38351	CHQ	158	83981.94
51	38351	SAV	95	22741.92
52	38351	SAV	189	67788.00
53	38861	CHQ	3	77231.12
54	38861	SAV	228	77031.07
55	38861	SAV	248	65919.35
56	41545	CHQ	102	89221.14
57	41545	CHQ	252	94530.03
58	41545	SAV	32	83408.19
59	44065	CHQ	109	56112.34
60	44065	SAV	193	20098.57
61	47953	CHQ	293	82812.96
62	47953	SAV	48	63416.35
63	49747	CHQ	153	50791.28
64	49747	SAV	142	86931.71
65	51850	CHQ	161	22932.00
66	51850	CHQ	182	29159.33
67	51850	SAV	35	77214.48
68	52189	CHQ	6	107309.23
69	52189	CHQ	62	36702.54



	customerID	type	accNumber	balance
70	52189	CHQ	79	49404.40
71	52189	SAV	53	49101.06
72	57796	CHQ	208	39569.33
73	57796	SAV	99	17951.51
74	59366	CHQ	54	48383.18
75	59366	CHQ	57	82512.57
76	59366	CHQ	148	100187.85
77	59366	SAV	26	112046.36
78	59366	SAV	64	87815.69
79	59366	SAV	152	31858.67
80	61976	CHQ	265	19061.45
81	61976	SAV	235	44741.90
82	62312	CHQ	61	11749.75
83	62312	SAV	261	55402.81
84	63772	CHQ	90	33551.51
85	63772	SAV	134	37690.50
86	63859	CHQ	113	82792.58
87	63859	SAV	291	101504.47
88	65441	CHQ	252	94530.03
89	65441	SAV	181	24453.37
90	67384	CHQ	65	61400.10
91	67384	SAV	37	9421.53
92	67384	SAV	223	41345.93
93	73386	CHQ	31	111209.89
94	73386	CHQ	66	40008.53
95	73386	SAV	94	74260.98
96	73386	SAV	253	74761.19
97	73925	CHQ	184	15291.30
98	73925	SAV	143	27480.19
99	77100	CHQ	190	89691.22
100	77100	SAV	101	17004.14
101	77100	SAV	230	63379.26
102	77100	SAV	253	74761.19
103	78477	CHQ	164	101336.25

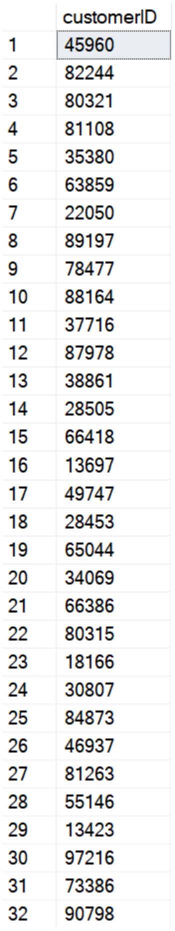


	customerID	type	accNumber	balance
104	78477	SAV	9	132271.23
105	78477	SAV	49	87557.84
106	79601	CHQ	52	23848.60
107	79601	CHQ	75	14043.82
108	79601	CHQ	165	108042.83
109	79601	SAV	26	112046.36
110	79601	SAV	110	36235.58
111	81108	CHQ	56	97555.21
112	81108	CHQ	207	57012.31
113	81108	SAV	121	103512.78
114	81263	CHQ	73	27130.90
115	81263	CHQ	122	48725.20
116	81263	CHQ	157	73162.44
117	81263	CHQ	195	88554.16
118	81263	SAV	98	69297.68
119	82333	CHQ	266	17608.20
120	82333	SAV	103	90491.84
121	86357	CHQ	23	86557.70
122	86357	CHQ	81	107129.47
123	86357	SAV	86	50837.08
124	87822	CHQ	149	81508.76
125	87822	CHQ	277	95702.75
126	87822	SAV	275	95955.98
127	88164	CHQ	220	84329.91
128	88164	SAV	120	27253.21
129	89902	CHQ	211	94562.36
130	89902	SAV	48	63416.35
131	89902	SAV	78	72742.21
132	90667	CHQ	30	63355.07
133	90667	CHQ	226	55444.17
134	90667	CHQ	233	46629.30
135	90667	SAV	97	11797.34
136	90798	CHQ	13	112505.84
137	90798	CHQ	57	82512.57
138	90798	SAV	146	95876.24
139	92389	CHQ	100	33128.61
140	92389	CHQ	105	27705.29
141	92389	CHQ	262	82475.58
142	92389	SAV	72	59597.18
143	92389	SAV	193	20098.57
144	92389	SAV	268	91951.04
145	92389	SAV	280	45824.72
146	93791	CHQ	46	30235.92
147	93791	SAV	44	69658.25
148	93791	SAV	155	55474.05
149	98923	CHQ	163	30169.57
150	98923	SAV	40	72419.68
151	99537	CHQ	11	90343.03
152	99537	CHQ	100	33128.61
153	99537	CHQ	274	64163.66
154	99537	CHQ	281	80968.75
155	99537	SAV	243	49766.04

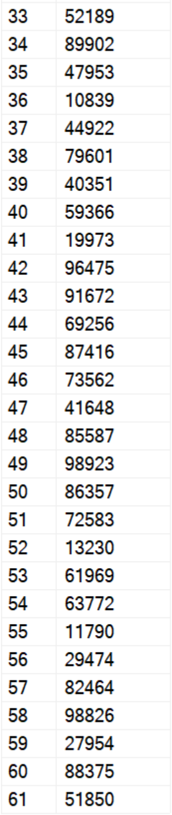
Question 5

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.

```
/* Num 5
*/
SELECT DISTINCT O1.customerID
FROM  Branch B1 , Owns O1,Account A1
WHERE B1.branchName='London'AND  O1.accNumber=A1.accNumber AND
A1.branchNumber=B1.branchNumber
EXCEPT
SELECT O3.customerID
FROM Owns O3, Owns O4
WHERE O3.accNumber=O4.accNumber AND O4.customerID IN (SELECT DISTINCT
O2.customerID
FROM  Branch B2 , Owns O2,Account A2
WHERE B2.branchName='Moscow' AND O2.accNumber=A2.accNumber AND
A2.branchNumber=B2.branchNumber)
```



	customerID
1	45960
2	82244
3	80321
4	81108
5	35380
6	63859
7	22050
8	89197
9	78477
10	88164
11	37716
12	87978
13	38861
14	28505
15	66418
16	13697
17	49747
18	28453
19	65044
20	34069
21	66386
22	80315
23	18166
24	30807
25	84873
26	46937
27	81263
28	55146
29	13423
30	97216
31	73386
32	90798



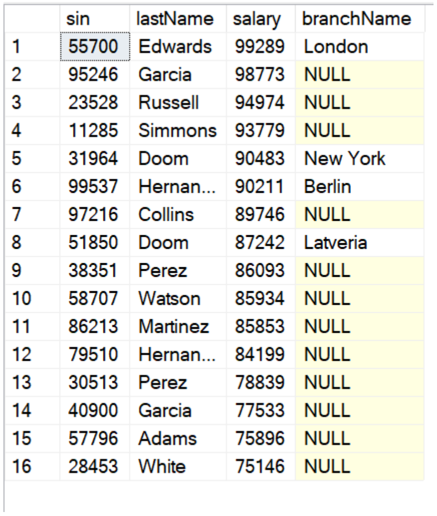
33	52189
34	89902
35	47953
36	10839
37	44922
38	79601
39	40351
40	59366
41	19973
42	96475
43	91672
44	69256
45	87416
46	73562
47	41648
48	85587
49	98923
50	86357
51	72583
52	13230
53	61969
54	63772
55	11790
56	29474
57	82464
58	98826
59	27954
60	88375
61	51850

Question 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).

```
/* Num 6
*/
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 salary DESC;
```

OUTPUT:



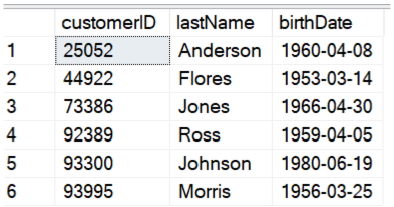
	sin	lastName	salary	branchName
1	55700	Edwards	99289	London
2	95246	Garcia	98773	NULL
3	23528	Russell	94974	NULL
4	11285	Simmons	93779	NULL
5	31964	Doom	90483	New York
6	99537	Hernan...	90211	Berlin
7	97216	Collins	89746	NULL
8	51850	Doom	87242	Latveria
9	38351	Perez	86093	NULL
10	58707	Watson	85934	NULL
11	86213	Martinez	85853	NULL
12	79510	Hernan...	84199	NULL
13	30513	Perez	78839	NULL
14	40900	Garcia	77533	NULL
15	57796	Adams	75896	NULL
16	28453	White	75146	NULL

Question 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.

```
 /* Num 7
 */
 SELECT C.customerID, C.lastName, C.birthDate
FROM Customer C
WHERE NOT EXISTS(
(SELECT DISTINCT A.branchNumber
FROM Customer C1, Owns O, Account A
WHERE C1.firstName = 'Jack' AND C1.lastName = 'Anderson'
AND C1.customerID = O.customerID AND O.accNumber = A.accNumber)
EXCEPT
(SELECT A.branchNumber
FROM Account A, Owns O
WHERE O.accNumber = A.accNumber AND O.customerID = C.customerID ))
ORDER BY customerID
```

OUTPUT:



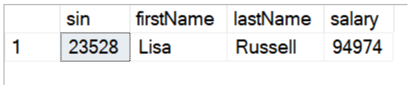
	customerID	lastName	birthDate
1	25052	Anderson	1960-04-08
2	44922	Flores	1953-03-14
3	73386	Jones	1966-04-30
4	92389	Ross	1959-04-05
5	93300	Johnson	1980-06-19
6	93995	Morris	1956-03-25

Question 8:

SIN, *first name*, *last name* and *salary* of the highest paid employee (or employees) of the *New York* branch, order by sin.

```
/* Num 8
*/
SELECT E2.sin,E2.firstName,E2.lastName,E2.salary
FROM Employee E2
WHERE E2.salary IN
( SELECT MAX(E.salary) AS highSalary
FROM Employee E, Branch B
WHERE E.branchNumber=B.branchNumber AND B.branchName='New York'
)
ORDER BY sin;
```

OUTPUT:



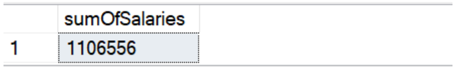
	sin	firstName	lastName	salary
1	23528	Lisa	Russell	94974

Question 9:

Sum of the employee salaries (a single number) at the *London* branch

```
/*Num 9
*/
SELECT SUM(E.salary) AS sumOfSalaries
FROM Employee E , Branch B
WHERE E.branchNumber=B.branchNumber AND B.branchName='London'
```

OUTPUT:



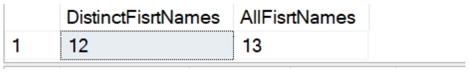
	sumOfSalaries
1	1106556

Question 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).

```
/*Num 10
*/
SELECT COUNT(DISTINCT E.firstName) AS DistinctFisrtNames, COUNT(E.firstName) AS
AllFisrtNames
FROM Employee E, Branch B
WHERE E.branchNumber=B.branchNumber AND B.branchName='Latveria'
```

OUTPUT:



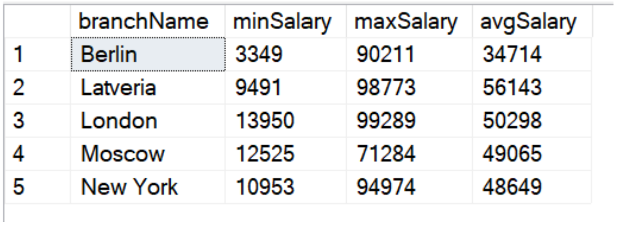
	DistinctFisrtNames	AllFisrtNames
1	12	13

Question 11

Branch name, and *minimum*, *maximum* and *average salary* of the employees at each branch, order by branch name.

```
/*Num 11
*/
SELECT B.branchName,MIN(E.salary) AS minSalary,MAX(E.salary) AS maxSalary,AVG(E.salary)
AS avgSalary
FROM Employee E , Branch B
WHERE E.branchNumber=B.branchNumber
GROUP BY branchName
ORDER BY branchName;
```

OUTPUT:



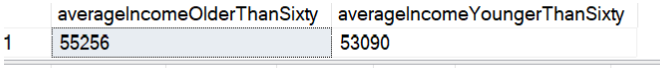
	branchName	minSalary	maxSalary	avgSalary
1	Berlin	3349	90211	34714
2	Latveria	9491	98773	56143
3	London	13950	99289	50298
4	Moscow	12525	71284	49065
5	New York	10953	94974	48649

Question 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).

```
 /*Num 12
 */
 SELECT AVG(C1.income) AS averageIncomeOlderThanSixty ,AVG(C2.income) AS
averageIncomeYoungerThanSixty
 FROM Customer C1, Customer C2
 WHERE DATEDIFF(year,C1.birthDate,GETDATE()) > 60 AND
DATEDIFF(year,C2.birthDate,GETDATE()) < 60
```

OUTPUT:



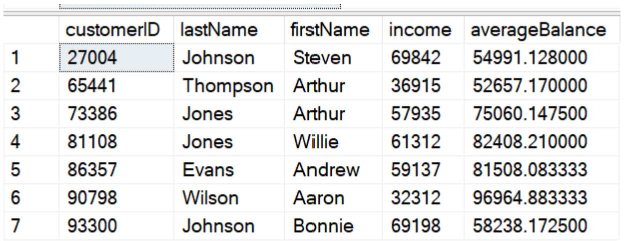
	averageIncomeOlderThanSixty	averageIncomeYoungerThanSixty
1	55256	53090

Question 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. **Jo**hn**s**on) **or** whose *first names* begin with *A* and have a vowel as the letter just before the last letter (e.g. **A**ar**o**n), 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.

```
 /*Num 13
 */
 SELECT C.customerID,C.lastName,C.firstName,C.income, AVG (A.balance) AS averageBalance
 FROM Customer C, Account A, Owns O
 WHERE C.customerID=O.customerID AND O.accNumber=A.accNumber AND (C.lastName LIKE
'JO%s%' OR (C.firstName LIKE 'A%[a,e,i,o,u]_'))
 GROUP BY C.customerID, C.firstName,C.lastName,C.income
 HAVING COUNT(C.customerID) > 2
 ORDER BY customerID;
```

OUTPUT:



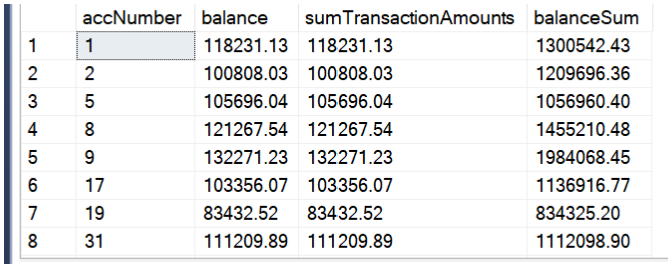
	customerID	lastName	firstName	income	averageBalance
1	27004	Johnson	Steven	69842	54991.128000
2	65441	Thompson	Arthur	36915	52657.170000
3	73386	Jones	Arthur	57935	75060.147500
4	81108	Jones	Willie	61312	82408.210000
5	86357	Evans	Andrew	59137	81508.083333
6	90798	Wilson	Aaron	32312	96964.883333
7	93300	Johnson	Bonnie	69198	58238.172500

Question 14

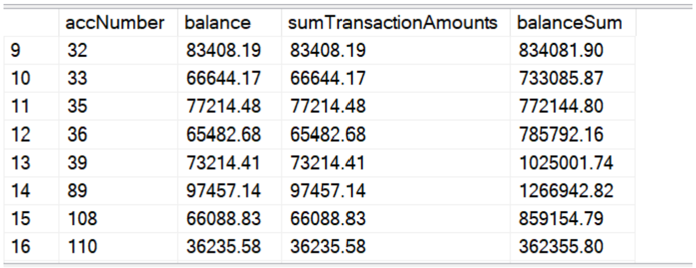
Account number, *balance*, *sum of transaction amounts*, and *balance - transaction sum* for accounts in the *London*branch that have at least ten transactions, order by account number.

```
/*Num 14
 */
 SELECT T.accNumber,A.balance, SUM(T.amount) AS sumTransactionAmounts, Sum(A.balance)
AS balanceSum
 FROM Transactions T, Branch B,Account A
 WHERE T.accNumber=A.accNumber AND A.branchNumber=B.branchNumber AND
B.branchName='London'
 GROUP BY T.accNumber,balance
 HAVING COUNT(T.transNumber)>9
```

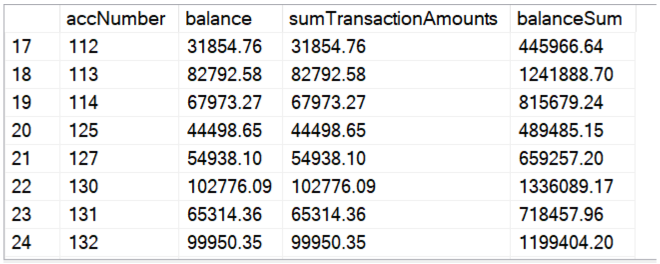
OUTPUT:



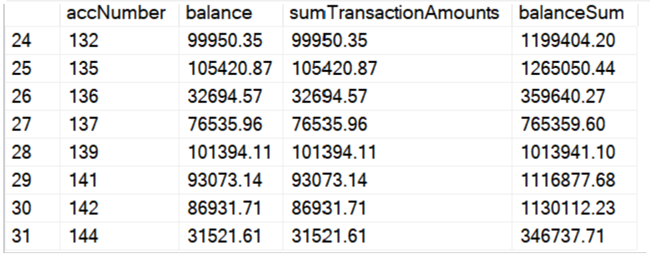
	accNumber	balance	sumTransactionAmounts	balanceSum
1	1	118231.13	118231.13	1300542.43
2	2	100808.03	100808.03	1209696.36
3	5	105696.04	105696.04	1056960.40
4	8	121267.54	121267.54	1455210.48
5	9	132271.23	132271.23	1984068.45
6	17	103356.07	103356.07	1136916.77
7	19	83432.52	83432.52	834325.20
8	31	111209.89	111209.89	1112098.90



	accNumber	balance	sumTransactionAmounts	balanceSum
9	32	83408.19	83408.19	834081.90
10	33	66644.17	66644.17	733085.87
11	35	77214.48	77214.48	772144.80
12	36	65482.68	65482.68	785792.16
13	39	73214.41	73214.41	1025001.74
14	89	97457.14	97457.14	1266942.82
15	108	66088.83	66088.83	859154.79
16	110	36235.58	36235.58	362355.80



	accNumber	balance	sumTransactionAmounts	balanceSum
17	112	31854.76	31854.76	445966.64
18	113	82792.58	82792.58	1241888.70
19	114	67973.27	67973.27	815679.24
20	125	44498.65	44498.65	489485.15
21	127	54938.10	54938.10	659257.20
22	130	102776.09	102776.09	1336089.17
23	131	65314.36	65314.36	718457.96
24	132	99950.35	99950.35	1199404.20



	accNumber	balance	sumTransactionAmounts	balanceSum
24	132	99950.35	99950.35	1199404.20
25	135	105420.87	105420.87	1265050.44
26	136	32694.57	32694.57	359640.27
27	137	76535.96	76535.96	765359.60
28	139	101394.11	101394.11	1013941.10
29	141	93073.14	93073.14	1116877.68
30	142	86931.71	86931.71	1130112.23
31	144	31521.61	31521.61	346737.71