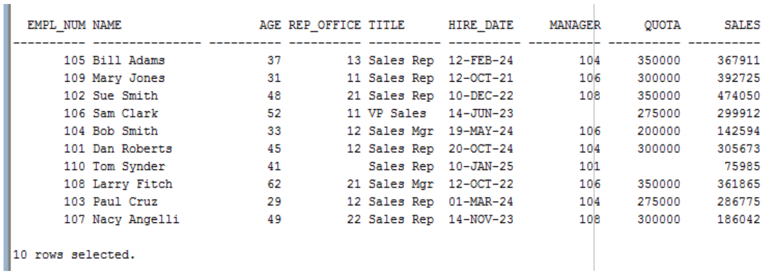
Assignment 6

No Duplicate Rows are allowed in the following questions

For each of the following questions, you need to show three things: the question itself, the query, and the snapshot of the answer.

For example, if the question is:

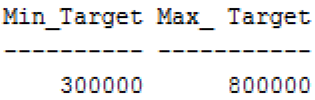
Question: **print the rows in the salesreps table**
Query: **select * from salesreps**
Output: **the snapshot of the result is:**



```
  EMPL_NUM NAME                  AGE REP_OFFICE TITLE      HIRE_DATE    MANAGER      QUOTA      SALES
---------- -------------- ---------- ---------- ---------- --------- ---------- ---------- ----------
       105 Bill Adams             37         13 Sales Rep  12-FEB-24        104     350000     367911
       109 Mary Jones             31         11 Sales Rep  12-OCT-21        106     300000     392725
       102 Sue Smith              48         21 Sales Rep  10-DEC-22        108     350000     474050
       106 Sam Clark              52         11 VP Sales   14-JUN-23                275000     299912
       104 Bob Smith              33         12 Sales Mgr  19-MAY-24        106     200000     142594
       101 Dan Roberts            45         12 Sales Rep  20-OCT-24        104     300000     305673
       110 Tom Synder             41            Sales Rep  10-JAN-25        101                 75985
       108 Larry Fitch            62         21 Sales Mgr  12-OCT-22        106     350000     361865
       103 Paul Cruz              29         12 Sales Rep  01-MAR-24        104     275000     286775
       107 Nacy Angelli           49         22 Sales Rep  14-NOV-23        108     300000     186042

10 rows selected.
```

1) Return the Minimum and Maximum Target for all offices.

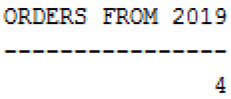
```
SELECT MIN(offices.target) AS "Min_Target", MAX(offices.target) AS "Max_ Target"
FROM offices;
```



```
Min_Target Max_ Target
---------- -----------
    300000      800000
```

2) Determine how many orders were made in 2019. Return the number of rows that meet this condition.

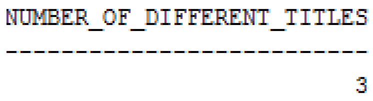
```
SELECT COUNT(*) AS "ORDERS FROM 2019"
FROM ORDERS
WHERE ORDERS.ORDER_DATE LIKE '%22';
```



```
ORDERS FROM 2019
----------------
               4
```

3) How many different titles in the sales reps table.

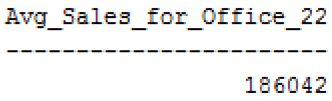
```
SELECT COUNT(DISTINCT title) AS NUMBER_OF_DIFFERENT_TITLES
FROM salesreps;
```



```
NUMBER_OF_DIFFERENT_TITLES
--------------------------
                         3
```

4) What is the average sales for salesreps in office 22.

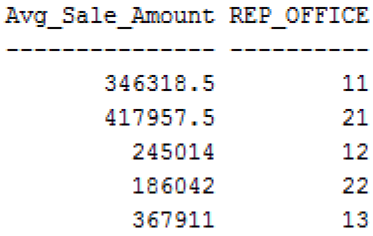
SELECT AVG(sales) AS "Avg_Sales_for_Office_22"
FROM salesreps
WHERE rep_office IN ('22');



```
Avg_Sales_for_Office_22
-----------------------
                 186042
```

5) What is the average sale amount for each sale reps in each office. Null should be ignored

SELECT AVG(salesreps.sales) AS "Avg_Sale_Amount", salesreps.rep_office
FROM salesreps, offices
WHERE salesreps.rep_office = offices.office and salesreps.sales is not null
GROUP BY salesreps.rep_office;

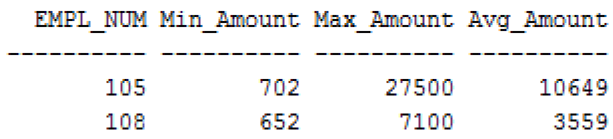


```
Avg_Sale_Amount REP_OFFICE
--------------- ----------
       346318.5         11
       417957.5         21
         245014         12
         186042         22
         367911         13
```

6) For each salesrep that has made an order, list the minimum, maximum and average order amount for all their orders. Include only those orders made anytime from 2020-2021. Omit from the list any salesrep that has only made 1 order in this time frame. Sort the results by Empl_Num.

SELECT salesreps.empl_num, MIN(ORDERS.AMOUNT) AS "Min_Amount",
MAX(ORDERS.AMOUNT) AS "Max_Amount", AVG(ORDERS.AMOUNT) AS "Avg_Amount"
FROM salesreps, ORDERS
WHERE salesreps.empl_num = ORDERS.REP AND EXTRACT(year FROM
ORDERS.ORDER_DATE) BETWEEN 2020 AND 2021 --include orders made anytime from
2020-2021
GROUP BY salesreps.empl_num
HAVING COUNT(ORDERS.ORDER_NUM) > 1 -- salesreps that made 1 order

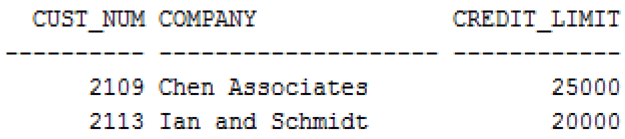
```
ORDER BY salesreps.empl_num;
```



```
  EMPL_NUM Min_Amount Max_Amount Avg_Amount
---------- ---------- ---------- ----------
       105        702      27500      10649
       108        652       7100       3559
```

7) Use a sub-query to list the Customer number; Name and Credit Limit of any customers who have exceeded their credit limit (amount > credit limit) on any order.

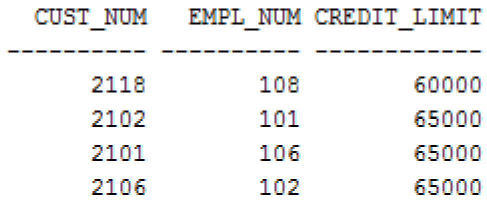
```
SELECT customers.CUST_NUM, customers.COMPANY, customers.CREDIT_LIMIT
FROM customers
WHERE customers.CREDIT_LIMIT < ANY
      (SELECT ORDERS.AMOUNT
       FROM ORDERS
       WHERE ORDERS.CUST = customers.CUST_NUM);
```



```
  CUST_NUM COMPANY              CREDIT_LIMIT
---------- -------------------- ------------
      2109 Chen Associates             25000
      2113 Ian and Schmidt             20000
```

8) Use a subquery and using the “all” keyword to find the customer number, Salesrep id, and CreditLimit of every customer whose CreditLimit is larger than the CreditLimit of all of the customers of sales rep number 109.

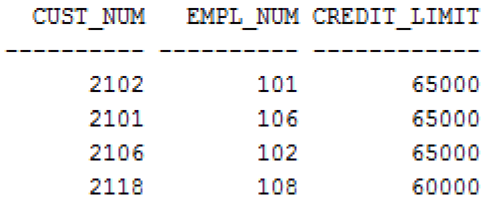
```
SELECT customers.CUST_NUM, salesreps.empl_num, customers.CREDIT_LIMIT
FROM customers, salesreps
WHERE customers.CUST_REP = salesreps.empl_num AND customers.CREDIT_LIMIT > ALL
      (SELECT customers.CREDIT_LIMIT
       FROM customers
       WHERE customers.CUST_REP = 109);
```



```
  CUST_NUM   EMPL_NUM CREDIT_LIMIT
---------- ---------- ------------
      2118        108        60000
      2102        101        65000
      2101        106        65000
      2106        102        65000
```

9) Do question 8, still using the subquery but do not use the “all” keyword.

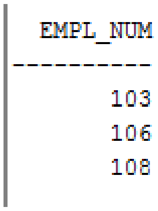
```
SELECT customers.CUST_NUM, salesreps.empl_num, customers.CREDIT_LIMIT
FROM customers, salesreps
WHERE customers.CUST_REP = salesreps.empl_num AND customers.CREDIT_LIMIT >
      (SELECT MAX(customers.CREDIT_LIMIT)
       FROM customers
       WHERE customers.CUST_REP = 109);
```



```
  CUST_NUM   EMPL_NUM CREDIT_LIMIT
---------- ---------- ------------
      2102        101        65000
      2101        106        65000
      2106        102        65000
      2118        108        60000
```

10) Use sub query and “in” keyword to print the salesreps (ids) who have taken order for the companies starts with letter ‘Z’ or with letter ‘J’. Duplicate rows are not allowed

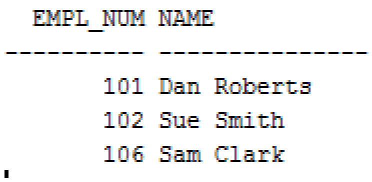
```
SELECT DISTINCT salesreps.empl_num
FROM salesreps
WHERE salesreps.empl_num IN
      (SELECT customers.cust_rep
       FROM customers
       WHERE customers.company LIKE 'Z%' OR customers.company LIKE 'J%');
```



```
  EMPL_NUM
----------
       103
       106
       108
```

11) Use sub query to find the id and the name of every sales rep that represents at least one customer with a credit limit of greater than $60,000.

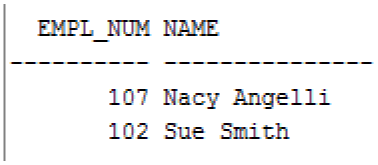
```
SELECT salesreps.empl_num, salesreps.name
FROM salesreps
WHERE salesreps.empl_num IN
      (SELECT customers.CUST_REP
       FROM customers
       WHERE customers.CREDIT_LIMIT > 60000.00);
```



```
  EMPL_NUM NAME
---------- ---------------
       101 Dan Roberts
       102 Sue Smith
       106 Sam Clark
```

12) Use sub query and keyword “exists” to list the id and the name of the salesreps in which some customers have orders some products in their hiredate.

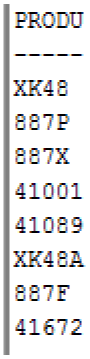
```
SELECT salesreps.empl_num, salesreps.name
FROM salesreps
WHERE EXISTS
      (SELECT ORDERS.REP
       FROM ORDERS
       WHERE salesreps.hire_date = ORDERS.ORDER_DATE);
```



```
  EMPL_NUM NAME
---------- ---------------
       107 Nacy Angelli
       102 Sue Smith
```

13) List all the products (only Product_ID) that have never been sold.

```
SELECT PRODUCT_ID
FROM PRODUCTS
WHERE NOT EXISTS
(SELECT ORDERS.QTY
 FROM ORDERS
 WHERE ORDERS.PRODUCT = PRODUCTS.PRODUCT_ID);
```

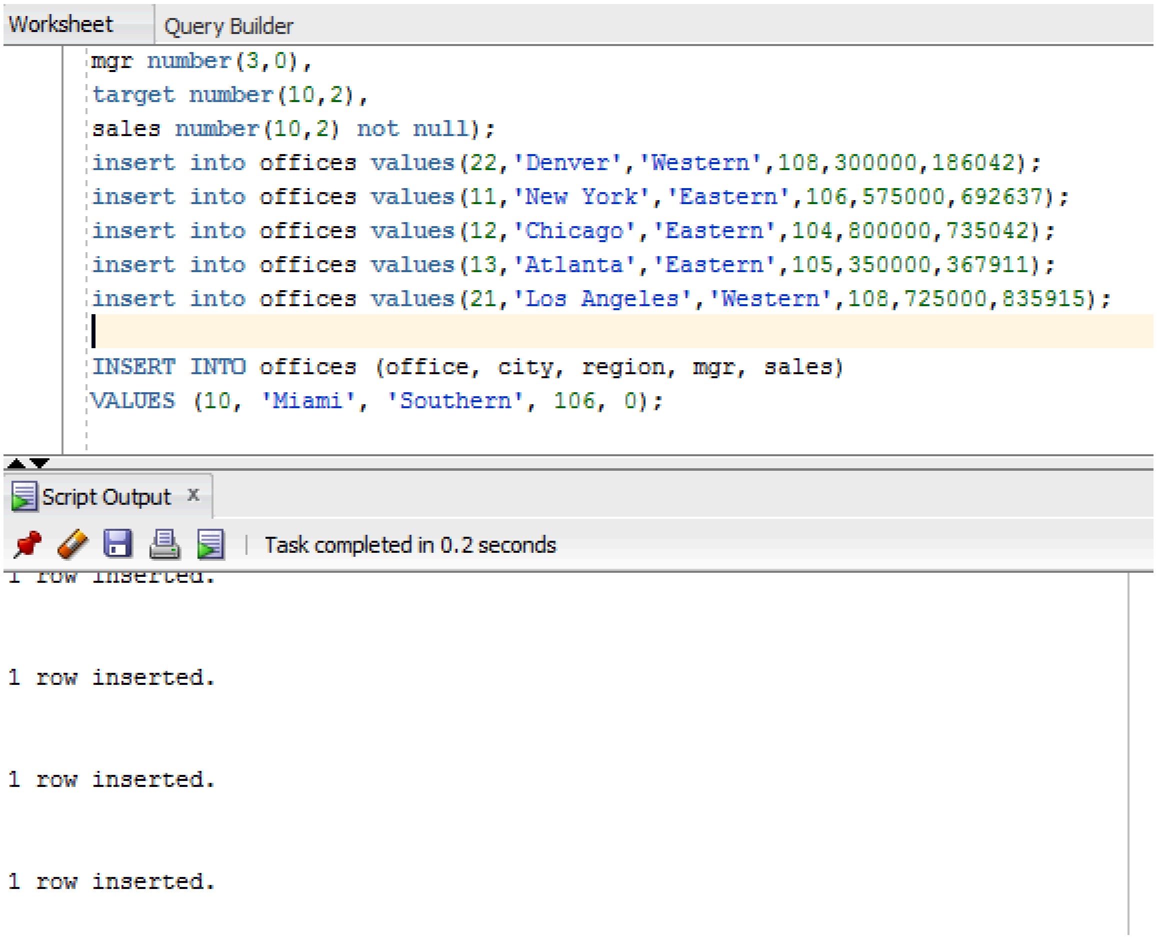


```
PRODU
-----
XK48
887P
887X
41001
41089
XK48A
887F
41672
```

14) Insert the following information into the OFFICES table: Office: 10 City: Miami Region: Southern Manager: 106 Sales: 0 Target should be Null. Do not use explicit Null for the

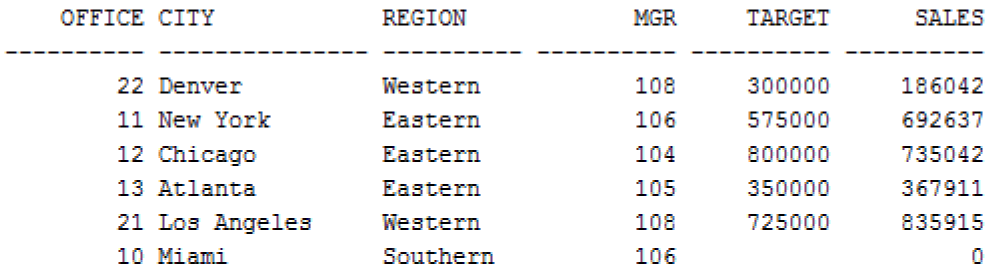
target in your insert statement. Show that office 10 is inserted by writing (select * from offices where office = 10) to revise the table to its original values Do (delete from offices where office = 10)

INSERT INTO offices (office, city, region, mgr, sales)
VALUES (10, 'Miami', 'Southern', 106, 0);



```
mgr number(3,0),
target number(10,2),
sales number(10,2) not null);
insert into offices values(22,'Denver','Western',108,300000,186042);
insert into offices values(11,'New York','Eastern',106,575000,692637);
insert into offices values(12,'Chicago','Eastern',104,800000,735042);
insert into offices values(13,'Atlanta','Eastern',105,350000,367911);
insert into offices values(21,'Los Angeles','Western',108,725000,835915);

INSERT INTO offices (office, city, region, mgr, sales)
VALUES (10, 'Miami', 'Southern', 106, 0);
```

```
1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

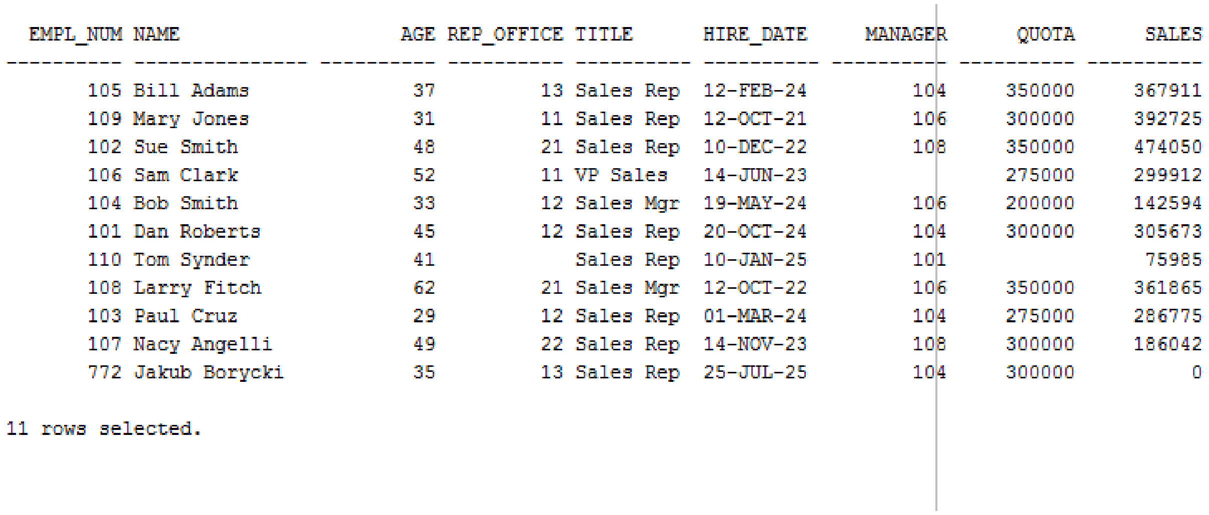


```
    OFFICE CITY            REGION            MGR     TARGET      SALES
---------- --------------- ---------- ---------- ---------- ----------
        22 Denver          Western           108     300000     186042
        11 New York        Eastern           106     575000     692637
        12 Chicago         Eastern           104     800000     735042
        13 Atlanta         Eastern           105     350000     367911
        21 Los Angeles     Western           108     725000     835915
        10 Miami           Southern          106                     0
```

15) Write an insert statement to add Your Name as Empl_Num 772. Use the date the insert is done for the hire date (sysdate). Sales is zero. - Other columns should remain NULL.

Use explicit null to make the other fields to be null; - Now delete this row to make the salesreps table goes back to its original state

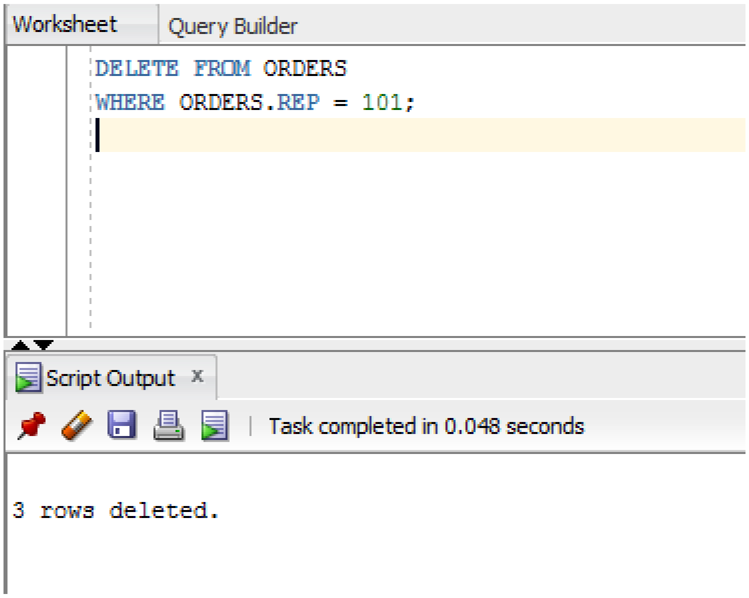
INSERT INTO salesreps
(empl_num, name, age, rep_office, title, hire_date, manager, quota, sales)
VALUES
(772, 'Jakub Borycki', 35, 13, 'Sales Rep', SYSDATE, 104, 300000, 0);



```
  EMPL_NUM NAME                   AGE REP_OFFICE TITLE      HIRE_DATE     MANAGER      QUOTA      SALES
---------- --------------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
       105 Bill Adams              37         13 Sales Rep  12-FEB-24         104     350000     367911
       109 Mary Jones              31         11 Sales Rep  12-OCT-21         106     300000     392725
       102 Sue Smith               48         21 Sales Rep  10-DEC-22         108     350000     474050
       106 Sam Clark               52         11 VP Sales   14-JUN-23                 275000     299912
       104 Bob Smith               33         12 Sales Mgr  19-MAY-24         106     200000     142594
       101 Dan Roberts             45         12 Sales Rep  20-OCT-24         104     300000     305673
       110 Tom Synder              41            Sales Rep  10-JAN-25         101                 75985
       108 Larry Fitch             62         21 Sales Mgr  12-OCT-22         106     350000     361865
       103 Paul Cruz               29         12 Sales Rep  01-MAR-24         104     275000     286775
       107 Nacy Angelli            49         22 Sales Rep  14-NOV-23         108     300000     186042
       772 Jakub Borycki           35         13 Sales Rep  25-JUL-25         104     300000          0

11 rows selected.
```

16) Use subquery to Delete all orders for employees 'Dan Roberts'. To make the orders table back to its original state, drop the order table and recreate it with its original records Recreate the orders table after doing the delete

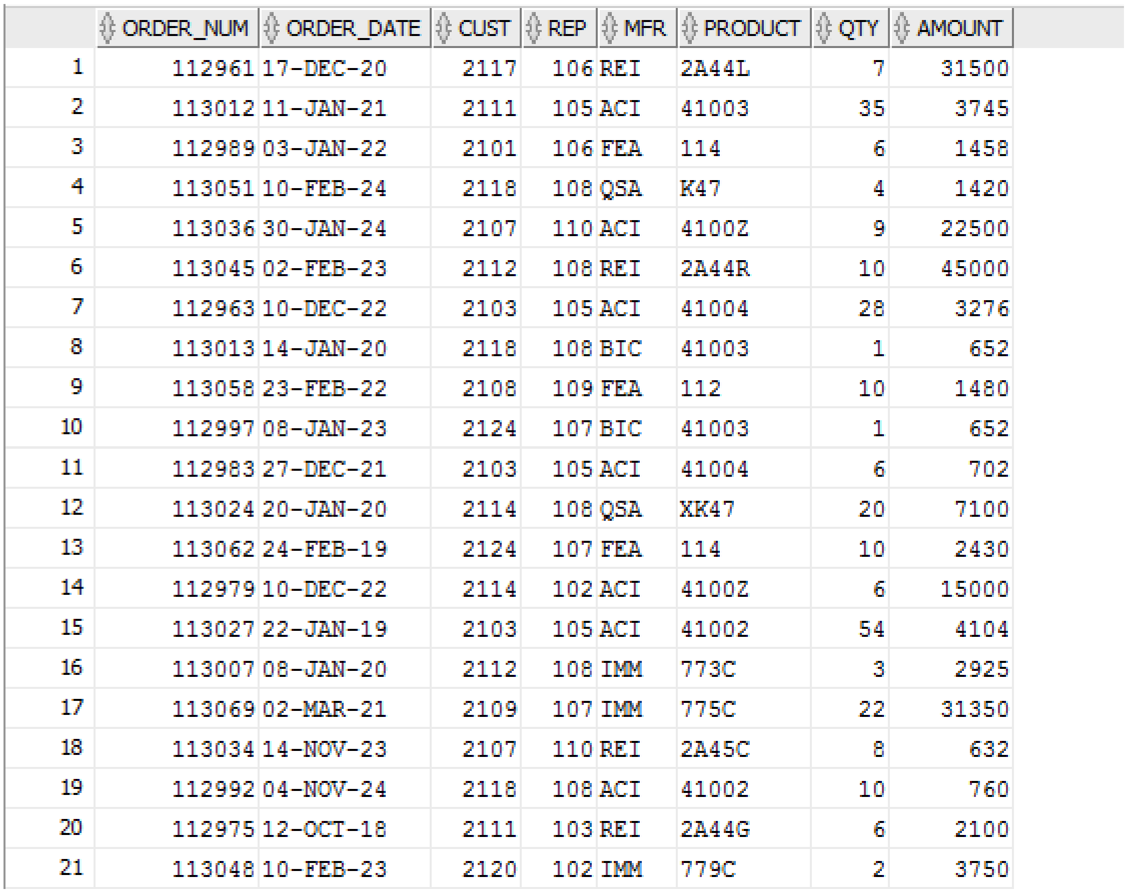
DELETE FROM ORDERS
WHERE ORDERS.REP = 101;



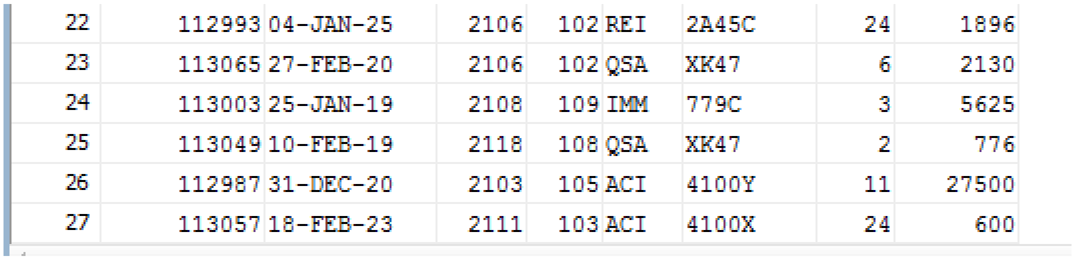
```
DELETE FROM ORDERS
WHERE ORDERS.REP = 101;
```

Task completed in 0.048 seconds

```
3 rows deleted.
```



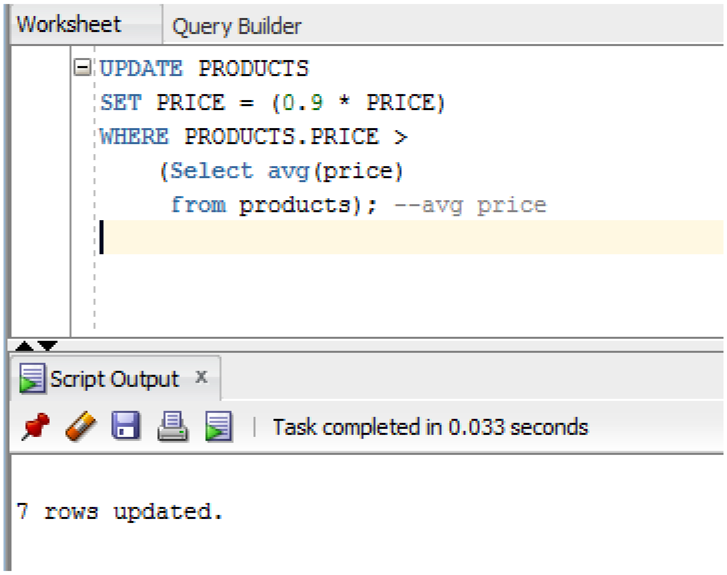
	ORDER_NUM	ORDER_DATE	CUST	REP	MFR	PRODUCT	QTY	AMOUNT
1	112961	17-DEC-20	2117	106	REI	2A44L	7	31500
2	113012	11-JAN-21	2111	105	ACI	41003	35	3745
3	112989	03-JAN-22	2101	106	FEA	114	6	1458
4	113051	10-FEB-24	2118	108	QSA	K47	4	1420
5	113036	30-JAN-24	2107	110	ACI	4100Z	9	22500
6	113045	02-FEB-23	2112	108	REI	2A44R	10	45000
7	112963	10-DEC-22	2103	105	ACI	41004	28	3276
8	113013	14-JAN-20	2118	108	BIC	41003	1	652
9	113058	23-FEB-22	2108	109	FEA	112	10	1480
10	112997	08-JAN-23	2124	107	BIC	41003	1	652
11	112983	27-DEC-21	2103	105	ACI	41004	6	702
12	113024	20-JAN-20	2114	108	QSA	XK47	20	7100
13	113062	24-FEB-19	2124	107	FEA	114	10	2430
14	112979	10-DEC-22	2114	102	ACI	4100Z	6	15000
15	113027	22-JAN-19	2103	105	ACI	41002	54	4104
16	113007	08-JAN-20	2112	108	IMM	773C	3	2925
17	113069	02-MAR-21	2109	107	IMM	775C	22	31350
18	113034	14-NOV-23	2107	110	REI	2A45C	8	632
19	112992	04-NOV-24	2118	108	ACI	41002	10	760
20	112975	12-OCT-18	2111	103	REI	2A44G	6	2100
21	113048	10-FEB-23	2120	102	IMM	779C	2	3750



22	112993	04-JAN-25	2106	102	REI	2A45C	24	1896
23	113065	27-FEB-20	2106	102	QSA	XK47	6	2130
24	113003	25-JAN-19	2108	109	IMM	779C	3	5625
25	113049	10-FEB-19	2118	108	QSA	XK47	2	776
26	112987	31-DEC-20	2103	105	ACI	4100Y	11	27500
27	113057	18-FEB-23	2111	103	ACI	4100X	24	600

17) Lower the price of the products by 10% if they are higher the average price Recreate the products table after doing the update

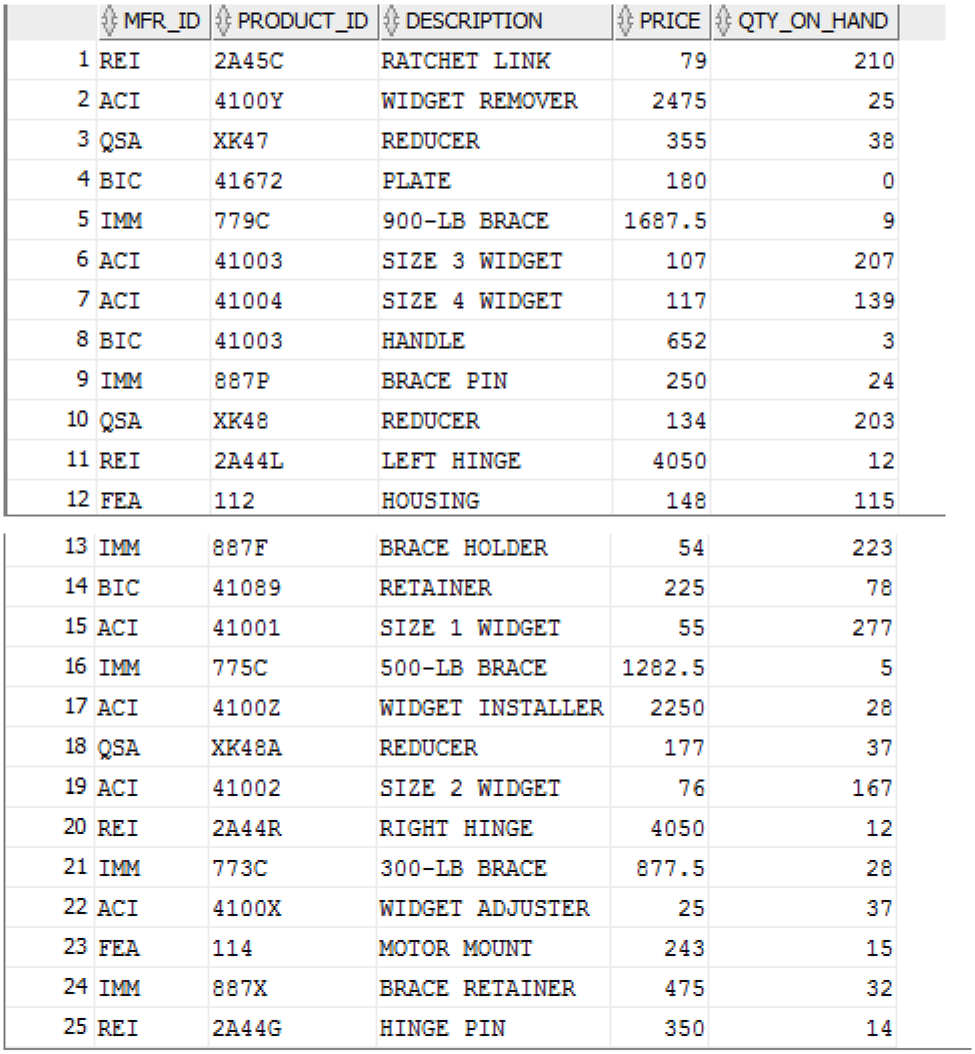
UPDATE PRODUCTS
SET PRICE = (0.9 * PRICE)
WHERE PRODUCTS.PRICE >
    (Select avg(price)
     from products); --avg price



```
UPDATE PRODUCTS
SET PRICE = (0.9 * PRICE)
WHERE PRODUCTS.PRICE >
    (Select avg(price)
     from products); --avg price
```

Task completed in 0.033 seconds

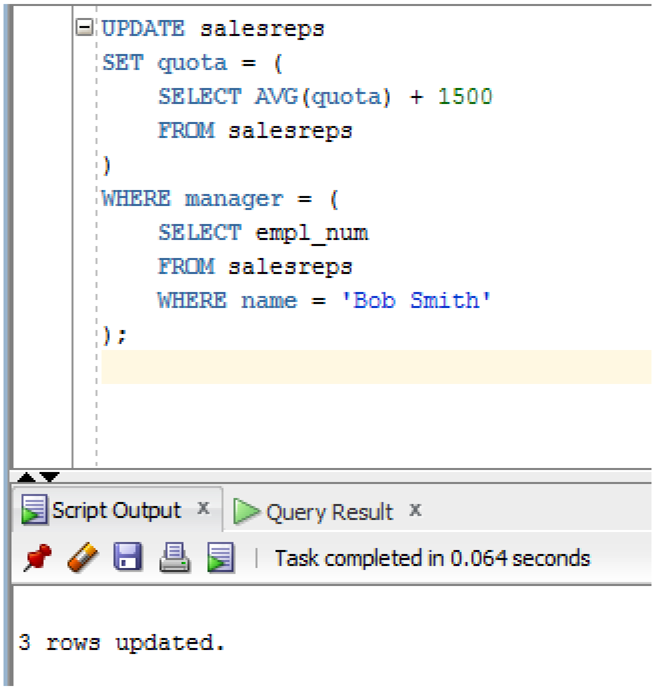
```
7 rows updated.
```



	MFR_ID	PRODUCT_ID	DESCRIPTION	PRICE	QTY_ON_HAND
1	REI	2A45C	RATCHET LINK	79	210
2	ACI	4100Y	WIDGET REMOVER	2475	25
3	QSA	XK47	REDUCER	355	38
4	BIC	41672	PLATE	180	0
5	IMM	779C	900-LB BRACE	1687.5	9
6	ACI	41003	SIZE 3 WIDGET	107	207
7	ACI	41004	SIZE 4 WIDGET	117	139
8	BIC	41003	HANDLE	652	3
9	IMM	887P	BRACE PIN	250	24
10	QSA	XK48	REDUCER	134	203
11	REI	2A44L	LEFT HINGE	4050	12
12	FEA	112	HOUSING	148	115
13	IMM	887F	BRACE HOLDER	54	223
14	BIC	41089	RETAINER	225	78
15	ACI	41001	SIZE 1 WIDGET	55	277
16	IMM	775C	500-LB BRACE	1282.5	5
17	ACI	4100Z	WIDGET INSTALLER	2250	28
18	QSA	XK48A	REDUCER	177	37
19	ACI	41002	SIZE 2 WIDGET	76	167
20	REI	2A44R	RIGHT HINGE	4050	12
21	IMM	773C	300-LB BRACE	877.5	28
22	ACI	4100X	WIDGET ADJUSTER	25	37
23	FEA	114	MOTOR MOUNT	243	15
24	IMM	887X	BRACE RETAINER	475	32
25	REI	2A44G	HINGE PIN	350	14

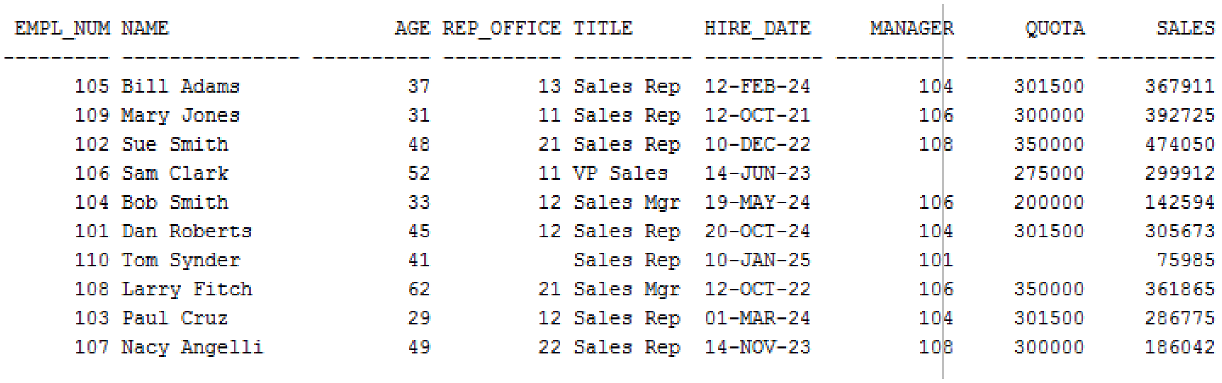
18) Set the quota of the salesreps to (average of the quota) + 1500 if his manager is “Bob Smith”.

```
UPDATE salesreps
SET quota = (
   SELECT AVG(quota) + 1500
   FROM salesreps
)
WHERE manager = (
   SELECT empl_num
   FROM salesreps
   WHERE name = 'Bob Smith'
);
```



```
UPDATE salesreps
SET quota = (
    SELECT AVG(quota) + 1500
    FROM salesreps
)
WHERE manager = (
    SELECT empl_num
    FROM salesreps
    WHERE name = 'Bob Smith'
);
```

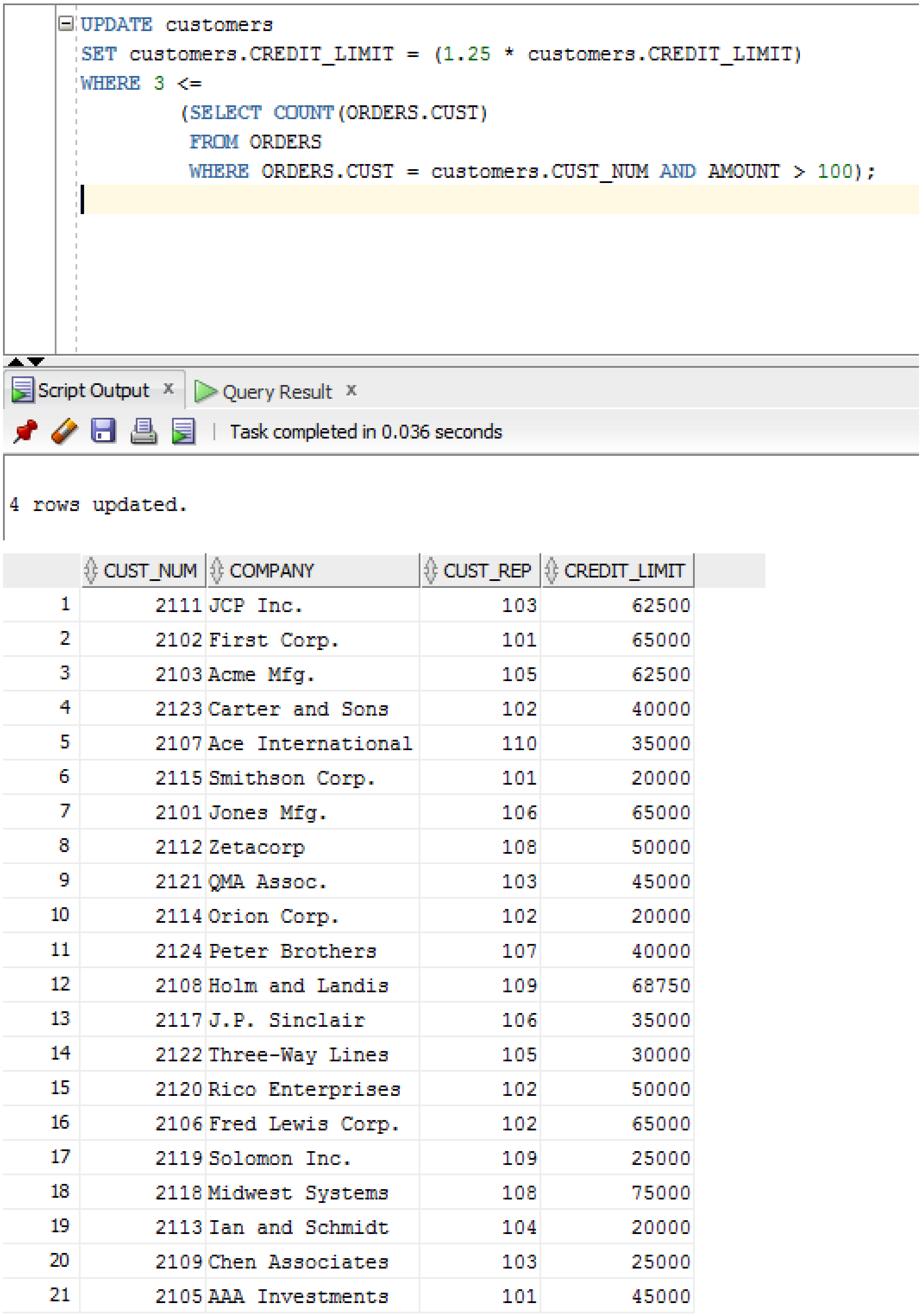
```
3 rows updated.
```



```
 EMPL_NUM NAME                   AGE REP_OFFICE TITLE      HIRE_DATE    MANAGER      QUOTA      SALES
--------- --------------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
      105 Bill Adams              37         13 Sales Rep  12-FEB-24         104     301500     367911
      109 Mary Jones              31         11 Sales Rep  12-OCT-21         106     300000     392725
      102 Sue Smith               48         21 Sales Rep  10-DEC-22         108     350000     474050
      106 Sam Clark               52         11 VP Sales   14-JUN-23                 275000     299912
      104 Bob Smith               33         12 Sales Mgr  19-MAY-24         106     200000     142594
      101 Dan Roberts             45         12 Sales Rep  20-OCT-24         104     301500     305673
      110 Tom Synder              41            Sales Rep  10-JAN-25         101                 75985
      108 Larry Fitch             62         21 Sales Mgr  12-OCT-22         106     350000     361865
      103 Paul Cruz               29         12 Sales Rep  01-MAR-24         104     301500     286775
      107 Nacy Angelli            49         22 Sales Rep  14-NOV-23         108     300000     186042
```

19) Increase customers credit limit by 25% for all customers that have 3 or more orders in which each order is more than $100. Recreate the customers table after doing this update

```
UPDATE customers
SET customers.CREDIT_LIMIT = (1.25 * customers.CREDIT_LIMIT)
WHERE 3 <=
                (SELECT COUNT(ORDERS.CUST)
                 FROM ORDERS
                 WHERE ORDERS.CUST = customers.CUST_NUM AND AMOUNT > 100);
```



```
UPDATE customers
SET customers.CREDIT_LIMIT = (1.25 * customers.CREDIT_LIMIT)
WHERE 3 <=
        (SELECT COUNT(ORDERS.CUST)
         FROM ORDERS
         WHERE ORDERS.CUST = customers.CUST_NUM AND AMOUNT > 100);
```

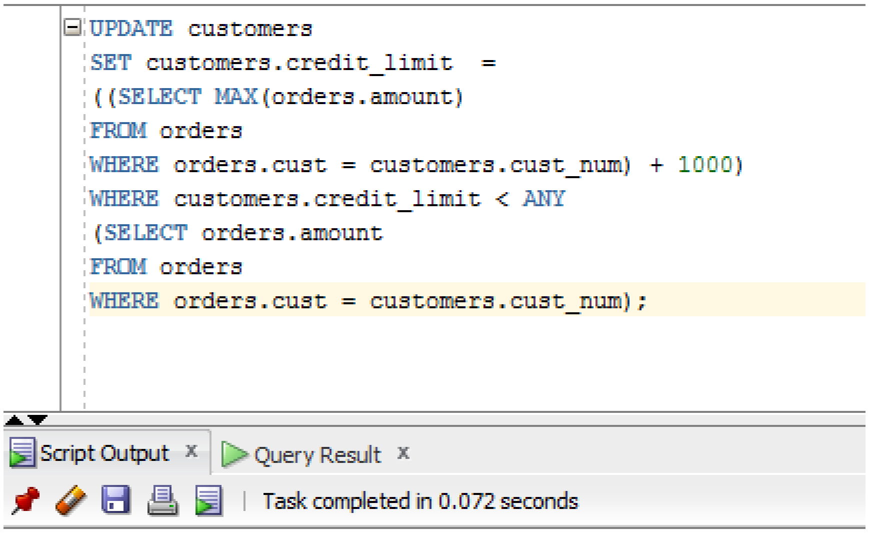
Task completed in 0.036 seconds

```
4 rows updated.
```

	CUST_NUM	COMPANY	CUST_REP	CREDIT_LIMIT
1	2111	JCP Inc.	103	62500
2	2102	First Corp.	101	65000
3	2103	Acme Mfg.	105	62500
4	2123	Carter and Sons	102	40000
5	2107	Ace International	110	35000
6	2115	Smithson Corp.	101	20000
7	2101	Jones Mfg.	106	65000
8	2112	Zetacorp	108	50000
9	2121	QMA Assoc.	103	45000
10	2114	Orion Corp.	102	20000
11	2124	Peter Brothers	107	40000
12	2108	Holm and Landis	109	68750
13	2117	J.P. Sinclair	106	35000
14	2122	Three-Way Lines	105	30000
15	2120	Rico Enterprises	102	50000
16	2106	Fred Lewis Corp.	102	65000
17	2119	Solomon Inc.	109	25000
18	2118	Midwest Systems	108	75000
19	2113	Ian and Schmidt	104	20000
20	2109	Chen Associates	103	25000
21	2105	AAA Investments	101	45000

20) Increase the credit limit of any customer who has any order that exceeds their credit limit. The new credit limit should be set to their maximum order amount plus $1,000. This must be done in 1 SQL statement. Recreate the customers table after doing this update

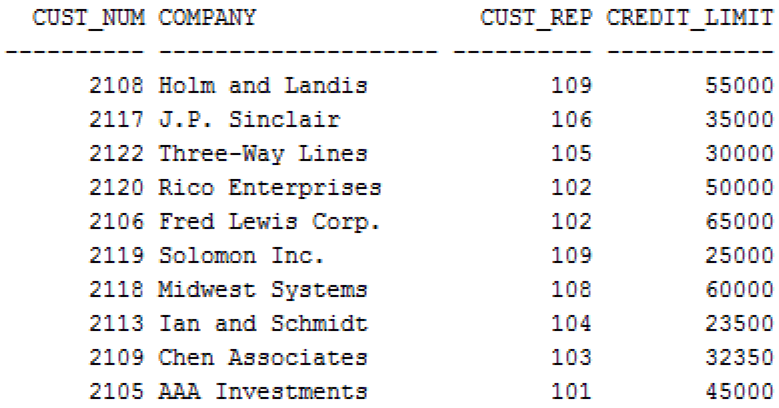
```
UPDATE customers
SET customers.credit_limit  =
((SELECT MAX(orders.amount)
FROM orders
WHERE orders.cust = customers.cust_num) + 1000)
WHERE customers.credit_limit < ANY
(SELECT orders.amount
FROM orders
WHERE orders.cust = customers.cust_num);
```



```
UPDATE customers
SET customers.credit_limit  =
((SELECT MAX(orders.amount)
FROM orders
WHERE orders.cust = customers.cust_num) + 1000)
WHERE customers.credit_limit < ANY
(SELECT orders.amount
FROM orders
WHERE orders.cust = customers.cust_num);
```

Task completed in 0.072 seconds

```
2 rows updated.



  CUST_NUM COMPANY                CUST_REP CREDIT_LIMIT
---------- -------------------- ---------- ------------
      2111 JCP Inc.                    103        50000
      2102 First Corp.                 101        65000
      2103 Acme Mfg.                   105        50000
      2123 Carter and Sons             102        40000
      2107 Ace International           110        35000
      2115 Smithson Corp.              101        20000
      2101 Jones Mfg.                  106        65000
      2112 Zetacorp                    108        50000
      2121 QMA Assoc.                  103        45000
      2114 Orion Corp.                 102        20000
      2124 Peter Brothers              107        40000
```



```
  CUST_NUM COMPANY              CUST_REP CREDIT_LIMIT
---------- -------------------- ---------- ------------
      2108 Holm and Landis             109        55000
      2117 J.P. Sinclair               106        35000
      2122 Three-Way Lines             105        30000
      2120 Rico Enterprises            102        50000
      2106 Fred Lewis Corp.            102        65000
      2119 Solomon Inc.                109        25000
      2118 Midwest Systems             108        60000
      2113 Ian and Schmidt             104        23500
      2109 Chen Associates             103        32350
      2105 AAA Investments             101        45000
```