

CS443 -- A3 - Solution

Write the queries necessary to obtain the required information. Make sure all columns you return have descriptive column headings. You must show the result of the query. For example, if the query is:

⇒ Show the office id, the city, and the region

⇒ Your query should be:

```
select office, city, region
from offices;
```

⇒ and then you need to show the following on the screen: (snapshot of your result)

OFFICE	CITY	REGION
22	Denver	Western
11	New York	Eastern
12	Chicago	Eastern
13	Atlanta	Eastern
21	Los Angeles	Western

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

```
Select min(target), max(target)
from offices;
```

MIN (TARGET)	MAX (TARGET)
30000	725000

2) Determine how many orders were made in 2022. Return the number of rows that meet this condition.

```
select count(*) from orders
where order_date like '%22';
```

COUNT (*)
2

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

```
select count( Distinct TITLE) from SALESREPS;
```

```
COUNT (DISTINCTTITLE)
-----
                        3
```

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

```
select avg(sales) from salesreps
where rep_office = 22
```

```
AVG (SALES)
-----
      186042
```

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

```
select rep_office, avg(sales)
from salesreps
where rep_office is not NULL
group by rep_office;
```

```
REP_OFFICE  AVG (SALES)
-----
11      346318.5
21      417957.5
12       245014
22       186042
13       367911
```

- 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 rep, avg(amount)"Average Order", max(Amount) "Maximum", min (amount)
"Minimum"
from orders
where order_date like '%20' or order_date like '%21'
group by rep
having count(Amount) > 1;
```

REP	Average Order	Maximum	Minimum
108	354071.5	722260	29250
107	209144	312350	22430
105	579805.75	921702	307745
110	171566	221500	121632

- 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 cust_num, company, Credit_limit
from customers
where credit_limit < Any (select amount from orders where cust = cust_num);
```

CUST_NUM	COMPANY	CREDIT_LIMIT
2111	First Corp.	650000
2124	Ace International	350000
2125	Smithson Corp.	200000
2155	Three-Way Lines	300000
2171	Jose Inc.	250000
2175	Iane and Schmidt	200000
2189	Chena Associates	250000

7 rows selected.

- 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 cust_num, cust_rep, Credit_Limit
from customers
where credit_limit > all (select credit_limit from customers where cust_rep = 109);
```

CUST_NUM	CUST_REP	CREDIT_LIMIT
2174	108	600000
2126	106	650000
2111	101	650000
2170	102	650000

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

```
select cust_num, cust_rep, Credit_Limit
from customers
where credit_limit > (select max(credit_limit) from customers where cust_rep = 109);
```

CUST_NUM	CUST_REP	CREDIT_LIMIT
2174	108	600000
2126	106	650000
2111	101	650000
2170	102	650000

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

```
select Empl_num, name
from salesreps
where Empl_num in (select cust_rep from customers
where (company like 'Z%' or company like 'J%'));
```

EMPL_NUM	NAME
109	Mary Jones
106	Sam Clark
108	Larry Fitch
103	Paul Cruz

- 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 \$600,000.

```
select Empl_num, name
from salesreps
where Empl_num in (select cust_rep from customers
where (credit_limit > 600000));
```

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 Empl_num, name
from salesreps
where exists (select * from orders where hire_date = order_date);
```

EMPL_NUM	NAME
109	Mary Jones
108	Larry Fitch

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

```
select product_id
from products
where product_id not in (select product from orders);
```

```
PRODU
-----
XK48
887P
887X
41001
775C
41089
XK48A
887F
41672

9 rows selected.
```

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.
- Delete office 10 to revise the table to its original values

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*

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

```
delete from orders where rep in
(select empl_num from salesreps where name in ('Dan Roberts'));
```

3 rows were deleted

Recreate the orders table after doing the delete

- 17) Lower the price of the products by 10% if they are higher the average price

```
update products
set price = (0.9 * price)
where price > (select avg(price) from products);
```

7 rows updated

Recreate the products table after doing the update

- 18) Set the quota of the salesreps to (average of the quota) + 1500 if they are hired in 2021.

```
update salesreps
set Quota = (select Avg(quota) from salesreps) + 1500
where hire_date like '%21';
```

2 rows updated

Recreate the salesreps table after doing the update

- 19) Increase customers credit limit by 25% for all customers that have 3 or more orders in which each order is more than 25,000.

```
UPDATE Customers
SET Credit_Limit = CREDIT_LIMIT+CREDIT_LIMIT*.25
WHERE 3<=(SELECT COUNT(*)
          FROM ORDERS
          WHERE AMOUNT>25000 and cust=cust_num);
```

4 rows updated

Recreate the customers table after doing this update

- 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.

```
Update customers
set Credit_Limit = (select max(amount) from orders where cust=cust_num) + 1000
where Credit_Limit < any (select amount from orders where cust = cust_num);
```

7 rows updated

Recreate the customers table after doing this update