Victor Nguyen

Professor Hadaegh

CS 443

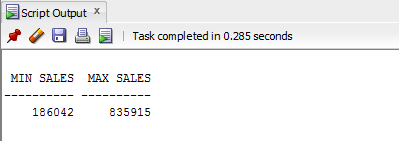
4 April 2020

Assignment 3

1) Return the Minimum and Maximum sales for all salesreps.

**SELECT MIN(SALES) AS "MIN SALES", MAX(SALES) AS "MAX SALES"**

**FROM OFFICES;**

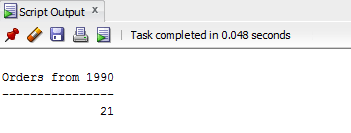


2) Determine how many orders were made in 1990. Return the number of rows that meet this condition.

**SELECT COUNT(\*) AS "Orders from 1990"**

**FROM ORDERS**

**WHERE ORDER\_DATE LIKE '%90';**

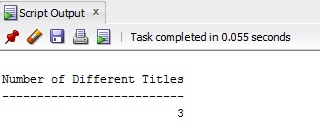


3) How many different titles in the sales reps table?

**SELECT COUNT(DISTINCT TITLE) AS "Number of Different Titles"**

**FROM SALESREPS**

**WHERE TITLE IS NOT NULL;**

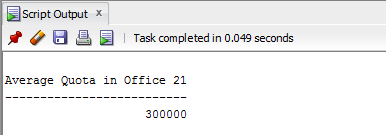


4) What is the average sale for salesreps in office 12?

**SELECT AVG(SALESREPS.QUOTAS) AS "Average Quota in Office 21"**

**FROM SALESREPS, OFFICES**

**WHERE OFFICE = 12;**



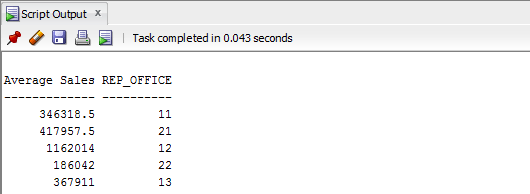
5) What is the average sale amount for each sale reps in each office. Null should be ignored

**SELECT AVG(SALESREPS.SALES) AS "Average Sales", SALESREPS.REP\_OFFICE**

**FROM SALESREPS, OFFICES**

**WHERE SALESREPS.REP\_OFFICE = OFFICES.OFFICE**

**GROUP BY SALESREPS.REP\_OFFICE;**



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 that are made in 1990. Include only the salesreps that has only made at least 2 orders 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"**

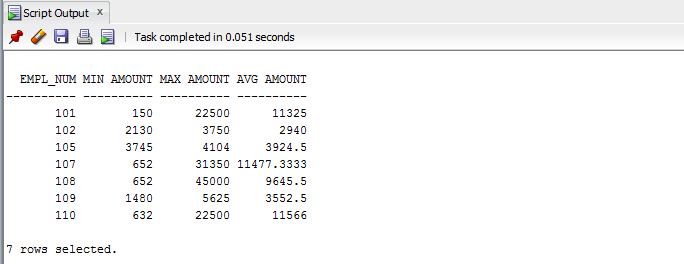
**FROM ORDERS, SALESREPS**

**WHERE SALESREPS.EMPL\_NUM = ORDERS.REPS AND ORDERS.ORDER\_DATE LIKE '%90'**

**GROUP BY SALESREPS.EMPL\_NUM**

**HAVING COUNT(ORDERS.ORDER\_NUM) > 1**

**ORDER BY SALESREPS.EMPL\_NUM;**



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

**SELECT CUSTOMERS.CUST\_NUM, CUSTOMERS.COMPANY, CUSTOMERS.CREDIT\_LIMIT**

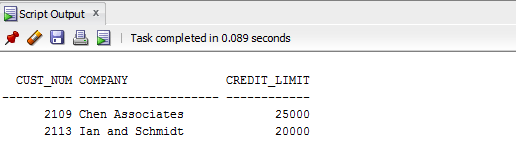
**FROM CUSTOMERS**

**WHERE CUSTOMERS.CREDIT\_LIMIT < ANY**

**(SELECT ORDERS.AMOUNT**

**FROM ORDERS**

**WHERE ORDERS.CUST = CUSTOMERS.CUST\_NUM);**



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 108.

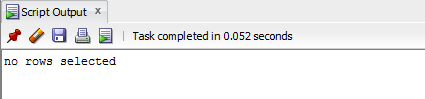
**SELECT CUST\_NUM AS "Customer #", CUST\_REP AS "REP ID", CREDIT\_LIMIT AS "CREDIT LIMIT"**

**FROM CUSTOMERS**

**WHERE CREDIT\_LIMIT >ALL (SELECT CREDIT\_LIMIT**

**FROM CUSTOMERS**

**WHERE CUST\_REP = 108);**

****

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**

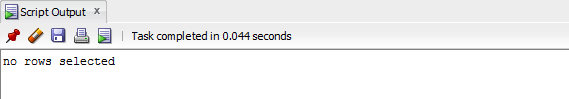
**FROM CUSTOMERS, SALESREPS**

**WHERE CUSTOMERS.CUST\_REP = SALESREPS.EMPL\_NUM AND CUSTOMERS.CREDIT\_LIMIT >**

**(SELECT MAX(CUSTOMERS.CREDIT\_LIMIT)**

**FROM CUSTOMERS**

**WHERE CUSTOMERS.CUST\_REP = 108);**

****

10) Use sub query and “in” keyword to print the salesreps (ids) who have taken order for the companies ‘Midwest Systems’.

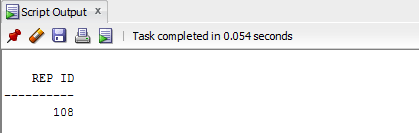
**SELECT DISTINCT REPS AS "REP ID"**

**FROM ORDERS**

**WHERE CUST IN (SELECT CUST\_NUM**

**FROM CUSTOMERS**

**WHERE COMPANY IN ('Midwest System'));**



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.

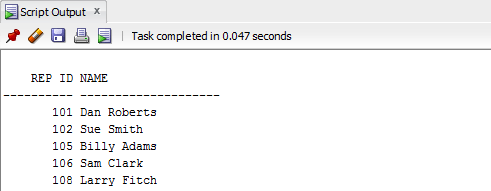
**SELECT EMPL\_NUM AS "REP ID", NAMES AS "NAME"**

**FROM SALESREPS**

**WHERE EMPL\_NUM IN (SELECT DISTINCT CUST\_REP**

**FROM CUSTOMERS**

**WHERE CREDIT\_LIMIT > 60000);**

****

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.

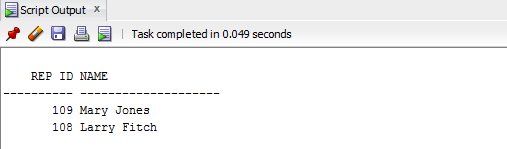
**SELECT SALESREPS.EMPL\_NUM AS "REP ID", SALESREPS.NAMES AS "NAME"**

**FROM SALESREPS**

**WHERE EXISTS (SELECT ORDERS.REPS**

**FROM ORDERS**

**WHERE SALESREPS.HIRE\_DATE = ORDERS.ORDER\_DATE);**

****

13) Use sub query to list all the products (only the Product\_ID) that has never been sold.

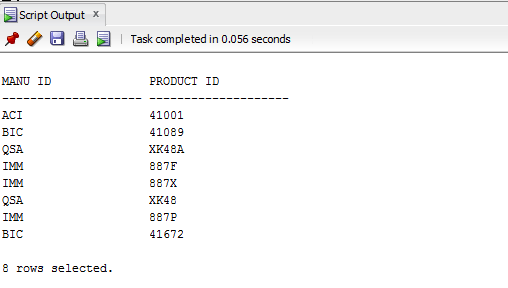
**SELECT PRODUCTS.MFR\_ID AS "MANU ID", PRODUCTS.PRODUCT\_ID AS "PRODUCT ID"**

**FROM PRODUCTS**

**WHERE NOT EXISTS (SELECT ORDERS.QTY**

**FROM ORDERS**

**WHERE (ORDERS.PRODUCT = PRODUCTS.PRODUCT\_ID AND ORDERS.MFR = PRODUCTS.MFR\_ID));**

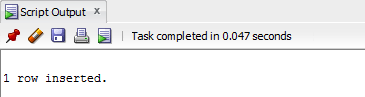


14) Insert the following information into the OFFICES table:

Office: 10 City: Miami Region: Southern Manager: 106 Sales: 0

**INSERT INTO OFFICES (OFFICE, CITY, REGION, MGR, SALES)**

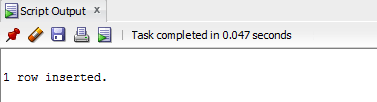
**VALUES (10, 'Miami', 'Southern', 106, 0);**



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.

**INSERT INTO SALESREPS (NAMES, EMPL\_NUM, HIRE\_DATE, SALES)**

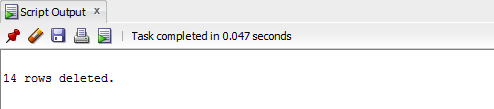
**VALUES ('Victor Nguyen', 772, sysdate, 0);**



16) Delete all orders for employees 'Dan Roberts', 'Sue Smith', and 'Larry Fitch'.

**DELETE FROM ORDERS**

**WHERE ORDERS.REPS IN (108, 101, 102);**



17) Increase customers credit limit by 25% for all customers that have 2 or more orders in which each order is more than 2500.

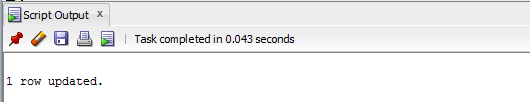
**UPDATE CUSTOMERS**

**SET CUSTOMERS.CREDIT\_LIMIT = (1.25 \* CUSTOMERS.CREDIT\_LIMIT)**

**WHERE 2 <= (SELECT COUNT(ORDERS.CUST)**

**FROM ORDERS**

**WHERE ORDERS.CUST = CUSTOMERS.CUST\_NUM AND ORDERS.AMOUNT > 2500);**



18) 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 CUSTOMERS.CREDIT\_LIMIT = 1000 + (SELECT MAX(ORDERS.AMOUNT)**

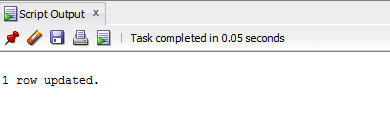
**FROM ORDERS**

**WHERE ORDERS.CUST = CUSTOMERS.CUST\_NUM)**

**WHERE CUSTOMERS.CREDIT\_LIMIT < ANY (SELECT ORDERS.AMOUNT**

**FROM ORDERS**

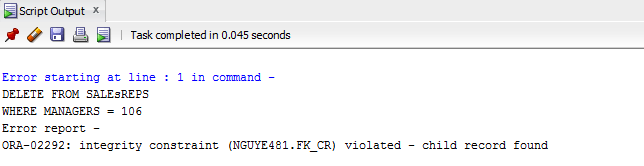
**WHERE ORDERS.CUST = CUSTOMERS.CUST\_NUM);**

****

19) Delete the salesreps whose managers is ‘Sam Clark’

**DELETE FROM SALESREPS**

**WHERE MANAGERS = 106;**



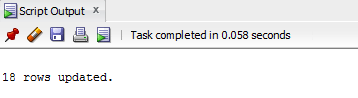
20) Double the price of the products that are less than the average of all prices

**UPDATE PRODUCTS**

**SET PRODUCTS.PRICE = 2 \* PRODUCTS.PRICE**

**WHERE PRODUCTS.PRICE < ANY (SELECT AVG(PRODUCTS.PRICE)**

**FROM PRODUCTS);**

****