5/9/2018

Parth Kapur

Assignment 3

CS443 – Professor. Hadaegh

CS443 -- Assignment 3

Write the queries necessary to obtain the required information. Make sure all columns you return have descriptive column headings.

1. Return the Minimum and Maximum sales for all offices.

**QUERY:**

SELECT MIN(SALES), MAX(SALES)

FROM OFFICES;

**RESULT:**

MIN(SALES) MAX(SALES)

---------- ----------

186042 835915

1. Determine how many orders were made in 1989. Return the number of rows that meet this condition.

**QUERY:**

SELECT COUNT(ORDER\_DATE)

FROM ORDERS

WHERE ORDER\_DATE LIKE '%89';

**RESULT:**

COUNT(ORDER\_DATE)

---------------------------------------

9

1. List the number of different titles in the sales reps table. Only list each title once and unknown titles should be ignored.

**QUERY:**

SELECT COUNT (DISTINCT TITLE)

FROM SALESREPS

WHERE TITLE IS NOT NULL;

**RESULT:**

COUNT(DISTINCTTITLE)

---------------------------------------

3

1. List the average quota for salesreps in office 21.

**QUERY:**

SELECT AVG(QUOTA)

FROM SALESREPS

WHERE REP\_OFFICE = 21;

**RESULT:**

AVG(QUOTA)

----------

350000

1. List the average sale amount for all sale reps in each office.

**QUERY:**

SELECT OFFICE, AVG(SALESREPS.SALES)

FROM SALESREPS, OFFICES

WHERE SALESREPS.REP\_OFFICE = OFFICES.OFFICE

GROUP BY OFFICE;

**RESULT:**

OFFICE AVG(SALESREPS.SALES)

---------- ---------------------------------------

22 186042

13 367911

11 346318.5

21 417957.5

12 245014

1. 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 1990-1999. Omit from the list any salesrep that has only made 1 order in this time frame. Sort the results by Empl\_Num.

**QUERY:**

SELECT SALESREPS.EMPL\_NUM, MIN(AMOUNT), MAX(AMOUNT), AVG(AMOUNT)

FROM SALESREPS, ORDERS

WHERE SALESREPS.EMPL\_NUM = ORDERS.REP

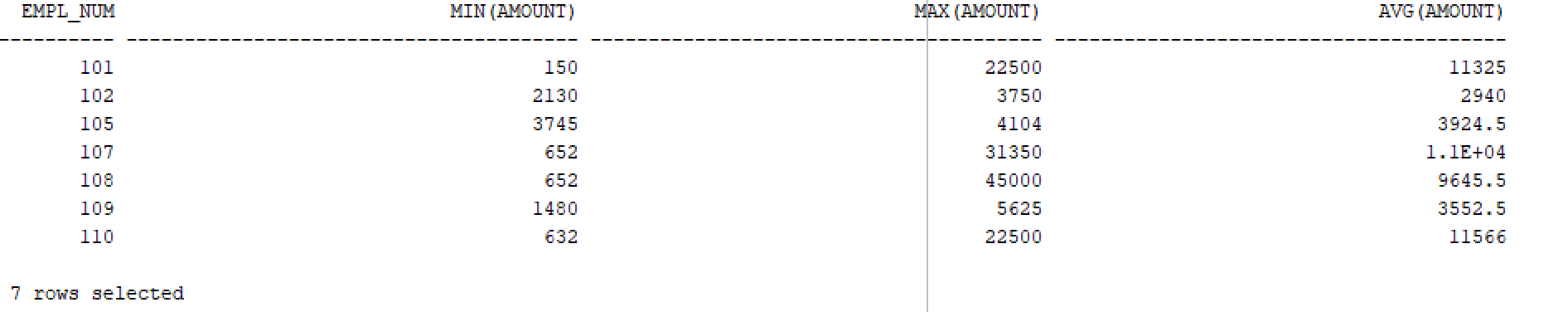
AND EXTRACT (YEAR FROM ORDERS.ORDER\_DATE) BETWEEN 1990 AND 1999

GROUP BY SALESREPS.EMPL\_NUM

HAVING COUNT(ORDERS.ORDER\_NUM) > 1

ORDER BY SALESREPS.EMPL\_NUM;

**RESULT:**



1. 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.

**QUERY:**

SELECT CUST\_NUM, COMPANY, CREDIT\_LIMIT

FROM CUSTOMERS

WHERE CREDIT\_LIMIT < ANY

(SELECT ORDERS.AMOUNT

FROM ORDERS

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

**RESULT:**

CUST\_NUM COMPANY CREDIT\_LIMIT

---------- -------------------- ------------

2109 Chen Associates 25000

2113 Ian and Schmidt 20000

1. 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.

**QUERY:**

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);

**RESULT:**

CUST\_NUM EMPL\_NUM CREDIT\_LIMIT

---------- ---------- ------------

2118 108 60000

2102 101 65000

2101 106 65000

2106 102 65000

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

**QUERY:**

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);

**RESULT:**

CUST\_NUM EMPL\_NUM CREDIT\_LIMIT

---------- ---------- ------------

2102 101 65000

2101 106 65000

2106 102 65000

2118 108 60000

1. Use sub query and “in” keyword to print the salesreps (ids) who have taken order for the companies ‘Zetacorp’ or ‘JCP Inc.’ . Duplicate rows is not allowed

**QUERY:**

SELECT DISTINCT REP

FROM ORDERS

WHERE CUST IN (SELECT CUST\_NUM

FROM CUSTOMERS

WHERE COMPANY = 'Zetacorp' OR COMPANY = 'JCP Inc.');

**RESULT:**

REP

----------

108

105

103

1. 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 $5000.

**QUERY:**

SELECT salesreps.empl\_num, salesreps.name

FROM salesreps

WHERE salesreps.empl\_num IN

(SELECT customers.cust\_rep

FROM customers

WHERE (customers.credit\_limit > 5000.00));

**RESULT:**

EMPL\_NUM NAME

---------- ---------------

101 Dan Roberts

102 Sue Smith

103 Paul Cruz

104 Bob Smith

105 Bill Adams

106 Sam Clark

107 Nacy Angelli

108 Larry Fitch

109 Mary Jones

110 Tom Synder

10 rows selected

1. 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.

**QUERY:**

SELECT salesreps.empl\_num, salesreps.name

FROM salesreps

WHERE EXISTS

(SELECT orders.rep

FROM orders

WHERE salesreps.hire\_date = orders.order\_date);

**RESULT:**

EMPL\_NUM NAME

---------- ---------------

109 Mary Jones

108 Larry Fitch

1. List all the products (Mfr\_ID and Product\_ID) that have never been sold. Use the ‘Exists’ clause.

**QUERY:**

SELECT products.mfr\_id, products.product\_id

FROM products

WHERE NOT EXISTS (

SELECT orders.QTY

FROM orders

WHERE (orders.product = products.product\_id AND orders.mfr = products.mfr\_id));

**RESULT:**

MFR PRODU

--- -----

ACI 41001

BIC 41089

IMM 887F

IMM 887X

QSA XK48A

QSA XK48

IMM 887P

BIC 41672

8 rows selected

## Updates Questions:

1. Insert the following information into the OFFICES table:

# Office: 39 City: Miami Region: Southern Manager: 106 Target: 1000000 Sales: 0

**QUERY:**

INSERT INTO offices (office, city, region, mgr, target, sales)

VALUES (39, 'Miami', 'Southern', 106, 1000000, 0);

**RESULT:**

1 row inserted.

1. Write an insert statement to add Your Name as Empl\_Num 772. Use the date the insert is run as the Hire date (sysdate). Sales are zero. Other column remain NULL;

**QUERY:**

INSERT INTO salesreps (name, empl\_num, hire\_date, sales)

VALUES ('Parth Kapur', 772, sysdate, 0);

**RESULT:**

1 row inserted.

1. Write an insert statement to add 'Tom Sawyer' Empl\_Num 814. Use the date the insert is run as the Hire date (sysdate). Sales are zero. Use implicit null values for columns that are not mentioned.

**QUERY:**

INSERT INTO salesreps (name, empl\_num, hire\_date, age, rep\_office, title, manager, quota, sales)

VALUES ('Tom Sawyer', 814, sysdate, NULL, NULL, NULL, NULL, NULL, 0);

**RESULT:**

1 row inserted.

1. Delete all orders for employees 108, 101, 102.

**QUERY:**

DELETE FROM orders

WHERE orders.rep IN (108, 101, 102);

**RESULT:**

14 rows deleted.

1. Delete all sales reps that have no orders and were hired before Jan 1 1987.

**QUERY:**

DELETE FROM salesreps

WHERE salesreps.hire\_date < '01-JAN-87' AND NOT EXISTS

(SELECT orders.rep

FROM orders

WHERE orders.rep = salesreps.empl\_num);

**RESULT:**

0 rows deleted.

1. Update your employee record with the following:

**Age: 37 Rep\_Office:39 Title: Senior VP Manager: NULL Quota: 100000**

**QUERY:**

UPDATE salesreps

SET age = 37, rep\_office = 39, title = 'Senior VP', manager = NULL, quota = 100000

WHERE empl\_num = 772;

**RESULT:**

1 row updated.

1. Increase customers credit limit by 25% for all customers that have 2 or more orders in which each order is more than 25,00.

**QUERY:**

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 amount > 2500);

**RESULT:**

1 row updated.

1. 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.

**QUERY:**

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);

**RESULT:**

1 row updated.

## Views and Security Questions

1. Create a view to show the Sales rep Name, and city that the Sales rep works in.

**QUERY:**

CREATE VIEW SALESREPCITYVIEW AS

SELECT salesreps.name, offices.city

FROM salesreps, offices

WHERE salesreps.rep\_office = offices.office;

**RESULT:**

View SALESREPCITYVIEW created.

1. Grant select access of the view created in question 22 to your Database instructors: Ahmad R. Hadaegh (with user id ahadaegh).

**QUERY:**

GRANT SELECT

ON SALESREPCITYVIEW

TO ahadaegh;

**RESULT:**

Error starting at line : 1 in command -

GRANT SELECT

ON SALESREPCITYVIEW

TO ahadaegh

Error report -

SQL Error: ORA-01917: user or role 'AHADAEGH' does not exist

01917. 00000 - "user or role '%s' does not exist"

\*Cause: There is not a user or role by that name.

\*Action: Re-specify the name.

**NOTE, THIS ERROR OCCURS BECAUSE I AM WORKING FROM MY OWN DATABASE WHICH I HAVE RUN THESE SCRIPTS ON. IN THIS DATABASE, THERE DOES NOT EXIST A USER OR ROLE CALLED AHADAEGH WITHIN THE SCHEMA. OTHERWISE “GRANT SUCCEEDED” WILL RESULT.**

**OTHERWISE “GRANT SUCCEEDED” WILL RESULT.**

“Grant Succeeded”

1. Create a view to show the customer name, product, description, quantity ordered and value of parts ordered. The column heading for the customers name should be ‘CustName’ and the column heading for value of parts ordered should be ‘Value’.

**QUERY:**

CREATE VIEW CUSTINFOVIEW AS

SELECT company AS "CustName", product, description, qty, price AS "Value"

FROM customers, products, orders

WHERE orders.cust = customers.cust\_num AND orders.product = products.product\_id AND orders.mfr = products.mfr\_id;

**RESULT:**

View CUSTINFOVIEW created.

1. Grant select access of the view created in question 24 to public

**QUERY:**

GRANT SELECT

ON CUSTINFOVIEW

TO public;

**RESULT:**

Grant succeeded.

1. Revoke access on view created in question 24 from Public.

**QUERY:**

REVOKE SELECT

ON CUSTINFOVIEW

FROM public;

**RESULT:**

Revoke succeeded.

1. Using the view created in question 24 above, list all information for product ‘l14’.

**QUERY:**

SELECT \*

FROM CUSTINFOVIEW

WHERE product = 'l14';

**RESULT:**

no rows selected

1. Create a view called TheManagers to list the name of all sales reps that manage some office. Along with the managers name, list the office number and city for each office.

**QUERY:**

CREATE VIEW TheManagers AS

SELECT salesreps.name, offices.office, offices.city

FROM salesreps, offices

WHERE salesreps.empl\_num = offices.mgr;

**RESULT:**

View THEMANAGERS created.

1. Grant all privileges on the view created in question 28 to your instructor.

**QUERY:**

GRANT ALL

ON TheManagers

TO ahadaegh;

**RESULT:**

Error starting at line : 1 in command -

GRANT ALL

ON TheManagers

TO ahadaegh

Error report -

SQL Error: ORA-01917: user or role 'AHADAEGH' does not exist

01917. 00000 - "user or role '%s' does not exist"

\*Cause: There is not a user or role by that name.

\*Action: Re-specify the name.

**NOTE, THIS ERROR OCCURS BECAUSE I AM WORKING FROM MY OWN DATABASE WHICH I HAVE RUN THESE SCRIPTS ON. IN THIS DATABASE, THERE DOES NOT EXIST A USER OR ROLE CALLED AHADAEGH WITHIN THE SCHEMA. OTHERWISE “GRANT SUCCEEDED” WILL RESULT.**

“Grant Succeeded”

1. Grant Select, Insert and Update on the Offices table to userids ‘jschmidt’ and ‘kmart’.

**QUERY:**

GRANT SELECT, INSERT, UPDATE

ON offices

TO jschmidt, kmart;

**RESULT:**

Error starting at line : 1 in command -

GRANT SELECT, INSERT, UPDATE

ON offices

TO jschmidt, kmart

Error report -

SQL Error: ORA-01917: user or role 'JSCHMIDT' does not exist

01917. 00000 - "user or role '%s' does not exist"

\*Cause: There is not a user or role by that name.

\*Action: Re-specify the name.

**NOTE, THIS ERROR OCCURS BECAUSE I AM WORKING FROM MY OWN DATABASE WHICH I HAVE RUN THESE SCRIPTS ON. IN THIS DATABASE, THERE DOES NOT EXIST A USER OR ROLE CALLED AHADAEGH WITHIN THE SCHEMA.**