Chris Banci

CS443

April 19, 2017

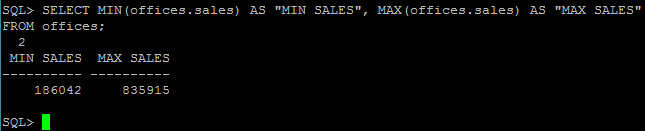
**CS443 -- Assignment 3**

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.**

SELECT MIN(offices.sales) AS "MIN SALES", MAX(offices.sales) AS "MAX SALES"

FROM offices;

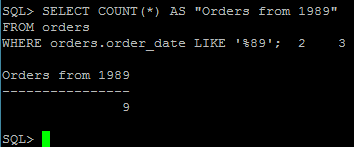


1. **Determine how many orders were made in 1989. Return the number of rows that meet this condition.**

SELECT COUNT(\*) AS "Orders from 1989"

FROM orders

WHERE orders.order\_date LIKE '%89';

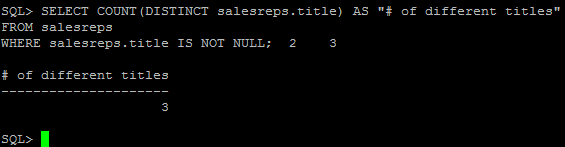


1. **List the number of different titles in the sales reps table. Only list each title once and unknown titles should be ignored.**

SELECT COUNT(DISTINCT salesreps.title) AS "# of different titles"

FROM salesreps

WHERE salesreps.title IS NOT NULL;

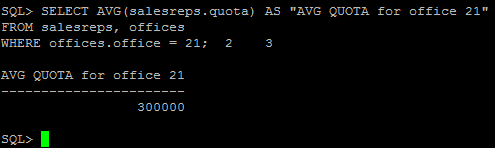


1. **List the average quota for salesreps in office 21.**

SELECT AVG(salesreps.quota) AS "AVG QUOTA for office 21"

FROM salesreps, offices

WHERE offices.office = 21;



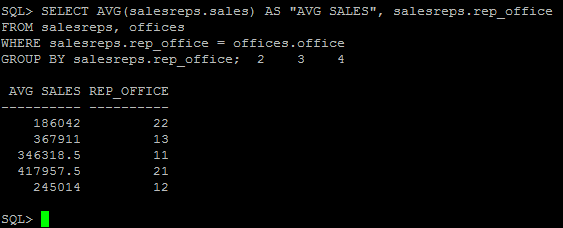
1. **List the average sale amount for all sale reps in each office.**

SELECT AVG(salesreps.sales) AS "AVG SALES", salesreps.rep\_office

FROM salesreps, offices

WHERE salesreps.rep\_office = offices.office

GROUP BY salesreps.rep\_office;



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.**

SELECT salesreps.empl\_num, MIN(orders.amount) AS "MIN AMOUNT", MAX(orders.amount) AS "MIN AMOUNT", AVG(orders.amount) AS "AVG AMOUNT"

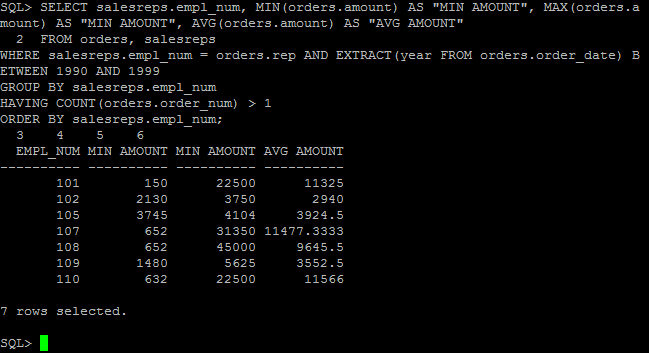
FROM orders, salesreps

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;



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.**

SELECT customers.cust\_num, customers.company, customers.credit\_limit

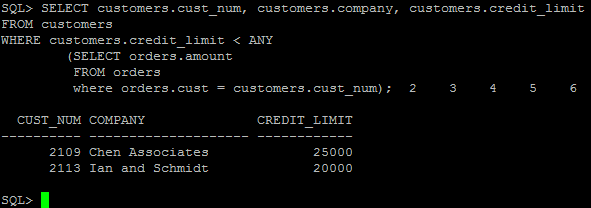
FROM customers

WHERE customers.credit\_limit < ANY

(SELECT orders.amount

FROM orders

where orders.cust = customers.cust\_num);



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.**

SELECT customers.cust\_num, salesreps.empl\_num, customers.credit\_limit

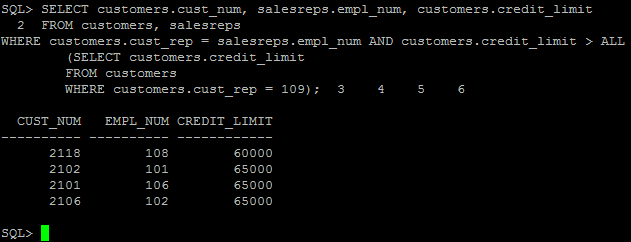
FROM customers, salesreps

WHERE customers.cust\_rep = salesreps.empl\_num AND customers.credit\_limit > ALL

(SELECT customers.credit\_limit

FROM customers

WHERE customers.cust\_rep = 109);



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

SELECT customers.cust\_num, salesreps.empl\_num, customers.credit\_limit

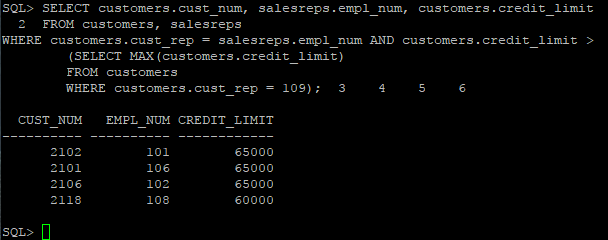
FROM customers, salesreps

WHERE customers.cust\_rep = salesreps.empl\_num AND customers.credit\_limit >

(SELECT MAX(customers.credit\_limit)

FROM customers

WHERE customers.cust\_rep = 109);



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**

SELECT DISTINCT salesreps.empl\_num

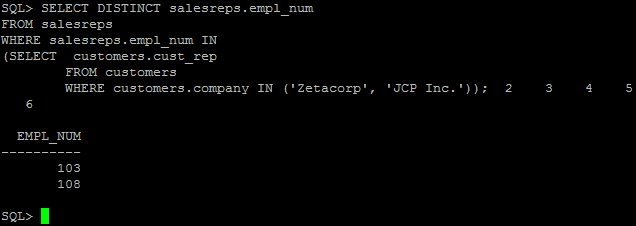
FROM salesreps

WHERE salesreps.empl\_num IN

(SELECT customers.cust\_rep

FROM customers

WHERE customers.company IN ('Zetacorp', 'JCP Inc.'));



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.**

SELECT salesreps.empl\_num, salesreps.name

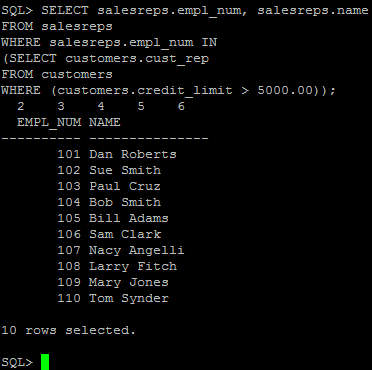
FROM salesreps

WHERE salesreps.empl\_num IN

(SELECT customers.cust\_rep

FROM customers

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



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.**

SELECT salesreps.empl\_num, salesreps.name

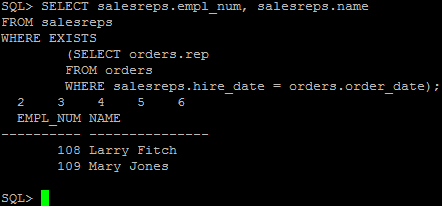
FROM salesreps

WHERE EXISTS

(SELECT orders.rep

FROM orders

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



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

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

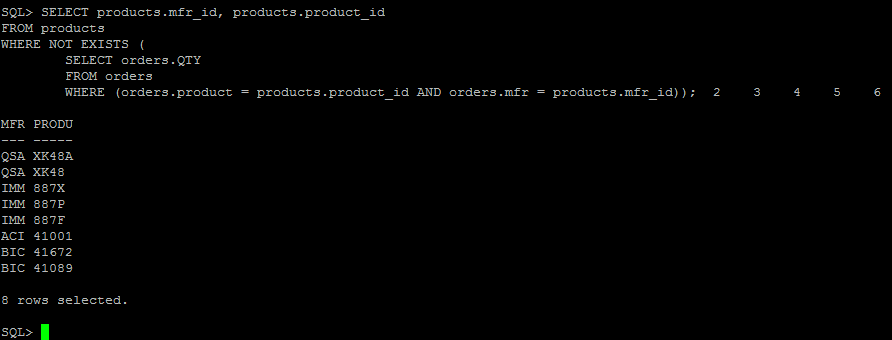
FROM products

WHERE NOT EXISTS (

SELECT orders.QTY

FROM orders

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



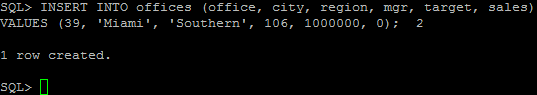
## Updates Questions:

1. **Insert the following information into the OFFICES table:**

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

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

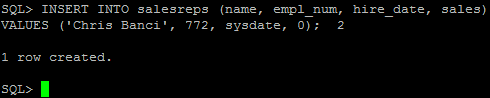
VALUES (39, 'Miami', 'Southern', 106, 1000000, 0);



**15) 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;**

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

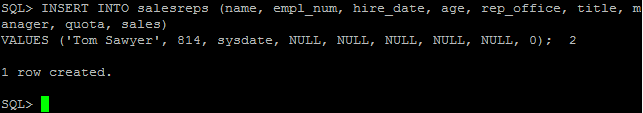
VALUES ('Chris Banci', 772, sysdate, 0);



**16) 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.**

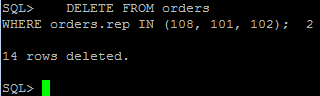
INSERT INTO salesreps (name, empl\_num, hire\_date, age, rep\_office, title, manager, quota, sales)

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



**17) Delete all orders for employees 108, 101, 102.**  
 DELETE FROM orders

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



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

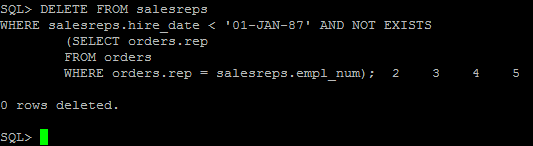
DELETE FROM salesreps

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

(SELECT orders.rep

FROM orders

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



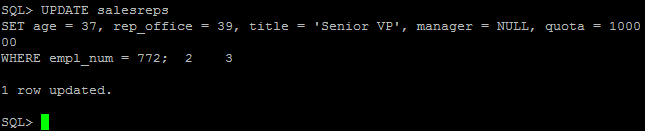
**19) Update your employee record with the following:**

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

UPDATE salesreps

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

WHERE empl\_num = 772;



**20) Increase customers credit limit by 20% for all customers that have 2 or more orders in which each order is more than 25,000.**

UPDATE customers

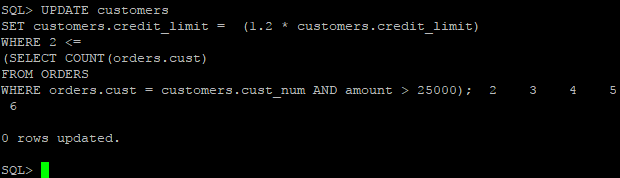
SET customers.credit\_limit = (1.2 \* customers.credit\_limit)

WHERE 2 <=

(SELECT COUNT(orders.cust)

FROM ORDERS

WHERE orders.cust = customers.cust\_num AND amount > 25000);



**21) 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 =

((SELECT MAX(orders.amount)

FROM orders

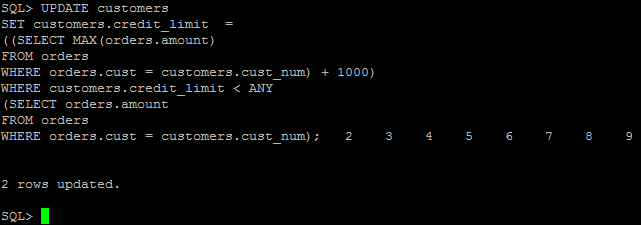
WHERE orders.cust = customers.cust\_num) + 1000)

WHERE customers.credit\_limit < ANY

(SELECT orders.amount

FROM orders

WHERE orders.cust = customers.cust\_num);



## Views and Security Questions

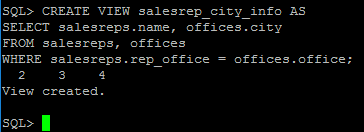
**22) Create a view to show the Sales rep Name, and city that the Sales rep works in.**

CREATE VIEW salesrep\_city\_info AS

SELECT salesreps.name, offices.city

FROM salesreps, offices

WHERE salesreps.rep\_office = offices.office;

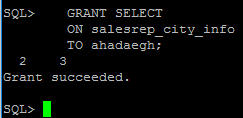


**23) Grant select access of the view created in question 22 to your Database instructors: Ahmad R. Hadaegh (with user id ahadaegh).**

GRANT SELECT

ON salesrep\_city\_info

TO ahadaegh;



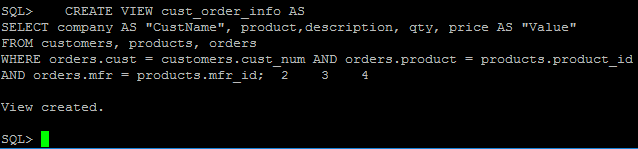
**24) Create a view to show the customer name, product, description, quantity ordered and value of parts ordered. The column heading for the customers name should be ‘CustName’ and the column heading for value of parts ordered should be ‘Value’.**

CREATE VIEW cust\_order\_info 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;

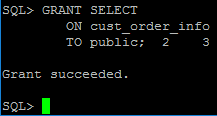


**25) Grant select access of the view created in question 24 to public**

GRANT SELECT

ON cust\_order\_info

TO public;

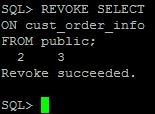


**26) Revoke access on view created in question 24 from public.**

REVOKE SELECT

ON cust\_order\_info

FROM public;

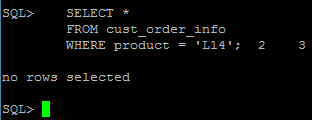


**27) Using the view created in question 24 above, list all information for product ‘L14’.**

SELECT \*

FROM cust\_order\_info

WHERE product = 'L14';



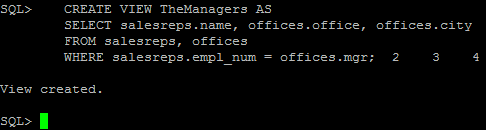
**28) Create a view called TheManagers to list the name of all sales reps that manage some office. Along with the managers name, list the office number and city for each office.**

CREATE VIEW TheManagers AS

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

FROM salesreps, offices

WHERE salesreps.empl\_num = offices.mgr;

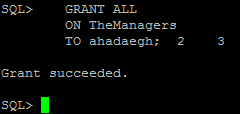


**29) Grant all privileges on the view created in question 28 to your instructor.**

GRANT ALL

ON TheManagers

TO ahadaegh;



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

GRANT SELECT, INSERT, UPDATE

ON offices

TO jschmidt, kmart;

