CS 443

ASSIGNMENT 3

Malialosa Taupule

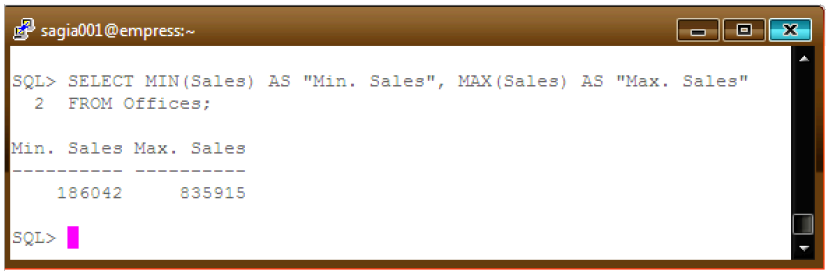
April 15, 2017

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

Query 1:

```
SELECT MIN(Sales) AS “Min. Sales”, MAX(Sales) AS “Max. Sales”
FROM Offices;
```

Result 1:

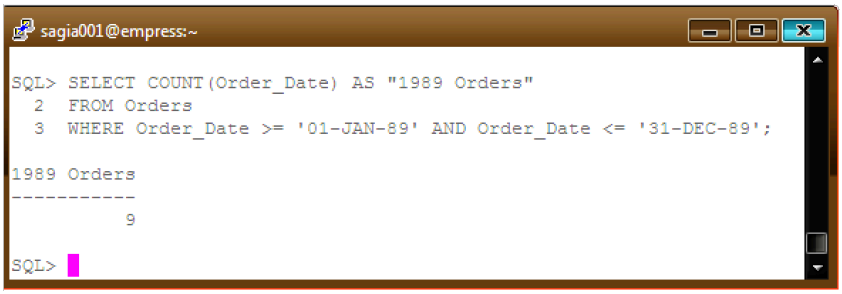


Question 2: Determine how many orders were made in 1989. Return the number of rows that meet this condition.

Query 2:

```
SELECT COUNT(*) AS “1989 Orders”
FROM Orders
WHERE Order_Date >= ’01-JAN-89’ AND Order_Date <= ’31-DEC-89’;
```

Result 2:

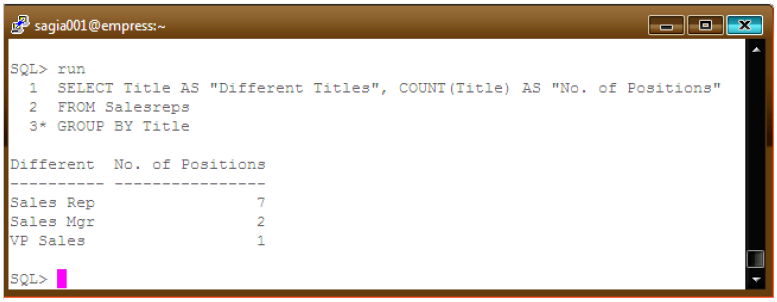


Question 3: List the number of different titles in the sales reps table. Only list each title once and unknown titles should be ignored.

Query 3:

```
SELECT Title AS “Different Titles”, COUNT(Title) AS “No. of Positions”
FROM Salesreps
GROUP BY Title;
```

Result 3:

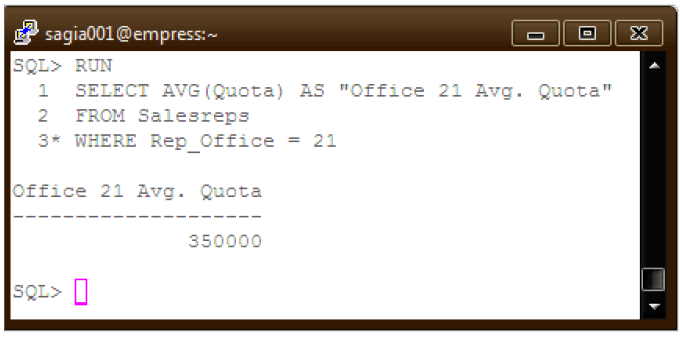


Question 4: List the average quota for salesreps in office 21.

Query 4:

```
SELECT AVG(Quota) AS “Avg. Quota”
FROM Salesreps
WHERE Rep_Office = 21;
```

Result 4:

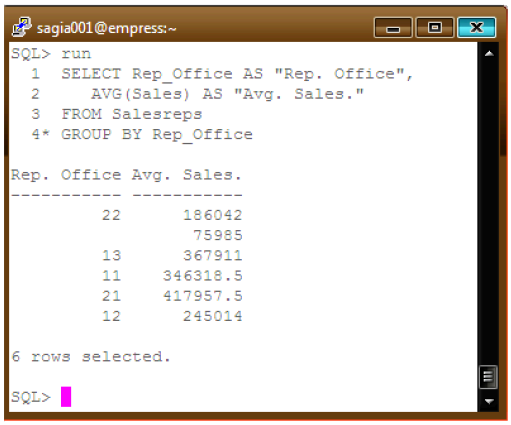


Question 5: List the average sale amount for all sale reps in each office.

Query 5:

```
SELECT Rep_Office AS “Rep. Office”, AVG(Sales) AS “Avg. Sales”
FROM Salesreps
GROUP BY Rep_Office;
```

Result 5:

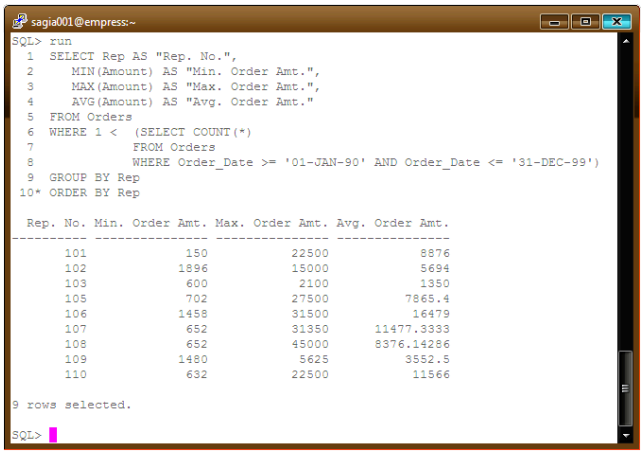


Question 6: For each salesrep that has made an order, list the minimum, maximum and average order amount for all their orders. Include only those orders made anytime from 1990-1999. Omit from the list any salesrep that has only made 1 order in this time frame. Sort the results by Empl_Num.

Query 6:

```
SELECT Rep AS "Rep. No.",
       MIN(Amount) AS "Min. Order Amt.",
     MAX(Amount) AS "Max. Order Amt.",
     AVG(Amount) AS "Avg. Order Amt."
FROM Orders
WHERE 1 <       (SELECT COUNT(*)
                FROM Orders
                WHERE Order_Date >= '01-JAN-90' AND Order_Date <= '31-DEC-99')
GROUP BY Rep
ORDER BY Rep;
```

Result 6:

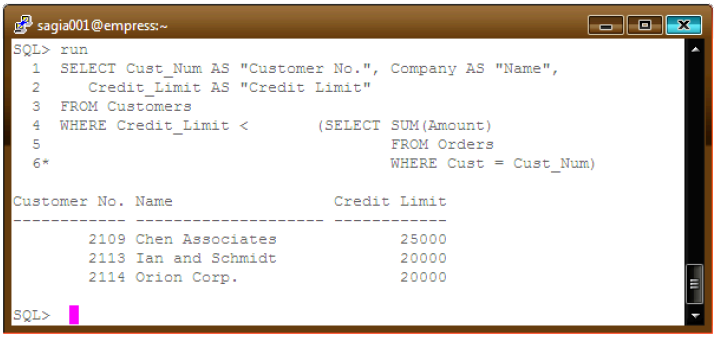


Question 7: Use a sub-query to list the Customer number; Name and Credit Limit of any customers who have exceeded their credit limit (amount > credit limit) on any order.

Query 7:

```
SELECT Cust_Num AS “Customer No.”, Company AS “Name”,
       Credit_Limit AS “Credit Limit”
FROM Customers
WHERE Credit_Limit <       (SELECT SUM(Amount)
                                  FROM Orders
                                  WHERE Cust = Cust_Num);
```

Result 7:

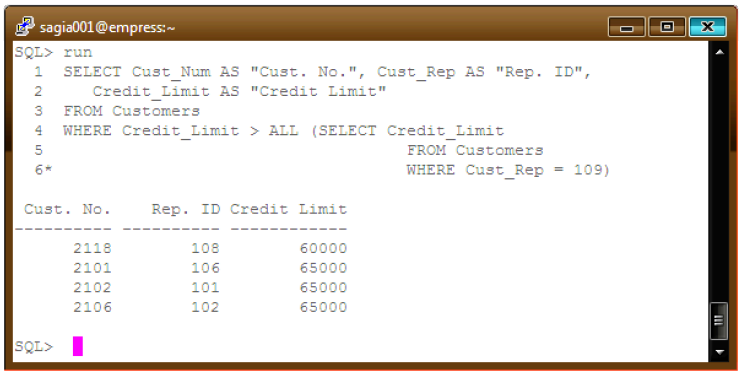


Question 8: Use a subquery and using the “all” keyword to find the customer number, Salesrep id, and CreditLimit of every customer whose CreditLimit is larger than the CreditLimit of all of the customers of sales rep number 109.

Query 8:

```
SELECT Cust_Num AS “Cust. No.”, Cust_Rep AS “Rep. ID”,
       Credit_Limit AS “Credit Limit”
FROM Customers
WHERE Credit_Limit > ALL (SELECT Credit_Limit
                                  FROM Customers
                                  WHERE Cust_Rep = 109);
```

Result 8:

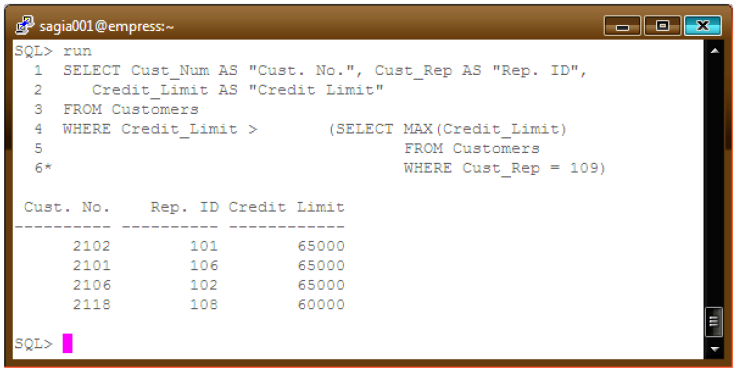


Question 9: Do question 8, still using the subquery but do not use the "all" keyword.

Query 9:

```
SELECT Cust_Num AS "Cust. No.", Cust_Rep AS "Rep. ID",
       Credit_Limit AS "Credit Limit"
FROM Customers
WHERE Credit_Limit >       (SELECT MAX(Credit_Limit)
                                  FROM Customers
                                  WHERE Cust_Rep = 109)
```

Result 9:

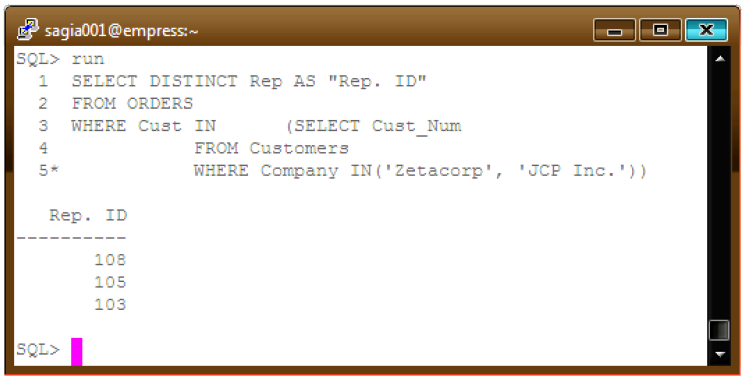


Question 10: Use sub query and "in" keyword to print the salesreps (ids) who have taken order for the companies 'Zetacorp' or 'JCP Inc.' . Duplicate rows is not allowed.

Query 10:

```
SELECT DISTINCT Rep AS "Rep. ID"
FROM ORDERS
WHERE Cust IN   (SELECT Cust_Num
                 FROM Customers
                 WHERE Company IN('Zetacorp', 'JCP Inc.'));
```

Result 10:

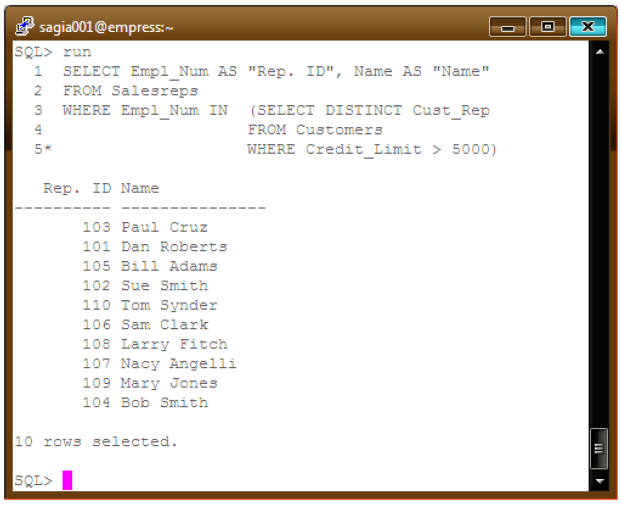


Question 11: Use sub query to find the id and the name of every sales rep that represents at least one customer with a credit limit of greater than $5000.

Query 11:

```
SELECT Empl_Num AS “Rep. ID”, Name AS “Name”
FROM Salesreps
WHERE Empl_Num IN   (SELECT DISTINCT Cust_Rep
                            FROM Customers
                            WHERE Credit_Limit > 5000);
```

Result 11:

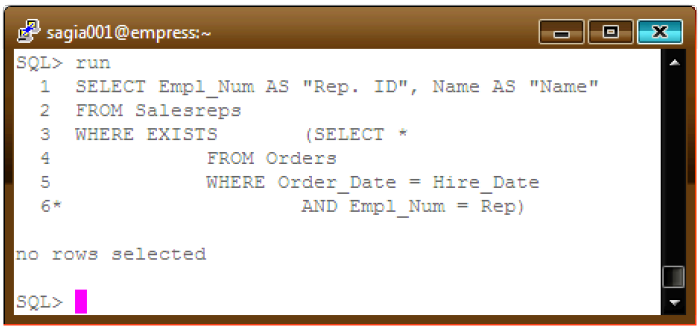


Question 12: Use sub query and keyword “exists” to list the id and the name of the salesreps in which some customers have orders of products in their hiredate.

Query 12:

```
SELECT Empl_Num AS “Rep. ID”, Name AS “Name”
FROM Salesreps
WHERE EXISTS (SELECT *
                    FROM Orders
                    WHERE Order_Date = Hire_Date
                            AND Empl_Num = Rep);
```

Result 12:

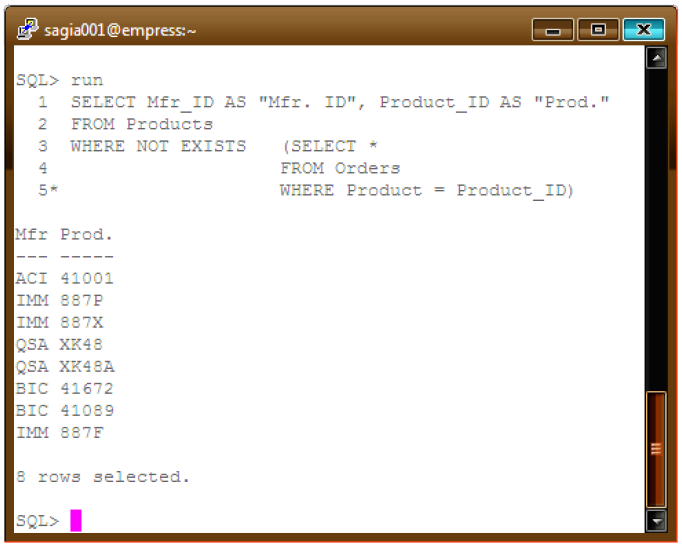


Question 13: List all the products (Mfr_ID and Product_ID) that have never been sold. Use the 'Exists' clause.

Query 13:

```
SELECT Mfr_ID AS "Mfr. ID", Product_ID AS "Prod."
FROM Products
WHERE NOT EXISTS        (SELECT *
                         FROM Orders
                         WHERE Product = Product_ID);
```

Result 13:



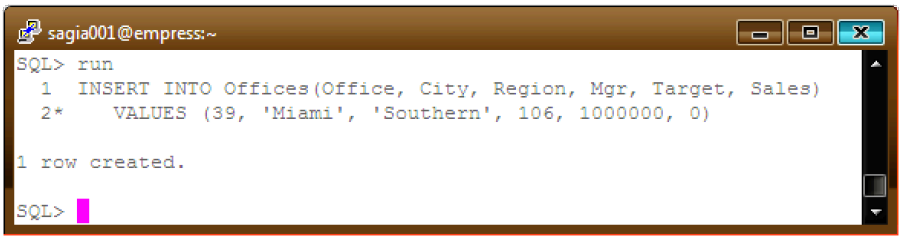
Question 14: Insert the following information into the OFFICES table:

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

Query 14:

```
INSERT INTO Offices(Office, City, Region, Mgr, Target, Sales)
       VALUES (39, "Miami, 'Southern', 106, 1000000, 0);
```

Result 14:

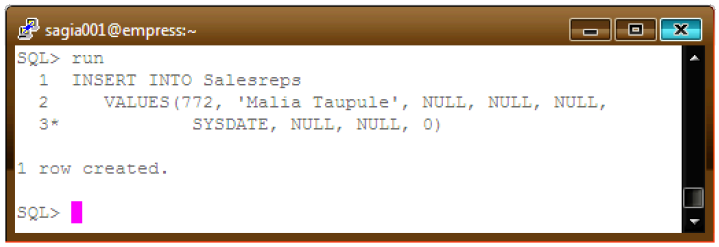


Question 15: Write an insert statement to add Your Name as Empl_Num 772. Use the date the insert is run as the Hire date (sysdate). Sales are zero. Other column remain NULL.

Query 15:

```
INSERT INTO Salesreps
      VALUES(772, 'Malialosa Taupule', NULL, NULL,
            NULL, SYSDATE, NULL, NULL, 0);
```

Result 15:

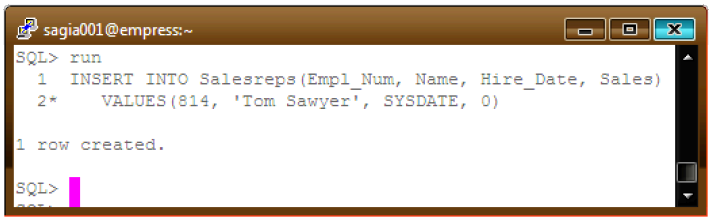


Question 16: Write an insert statement to add 'Tom Sawyer' Empl_Num 814. Use the date the insert is run as the Hire date (sysdate). Sales are zero. Use implicit null values for columns that are not mentioned.

Query 16:

```
INSERT INTO Salesreps(Empl_Num, Name, Hire_Date, Sales)
      VALUES(814, 'Tom Sawyer', SYSDATE, 0);
```

Result 16:

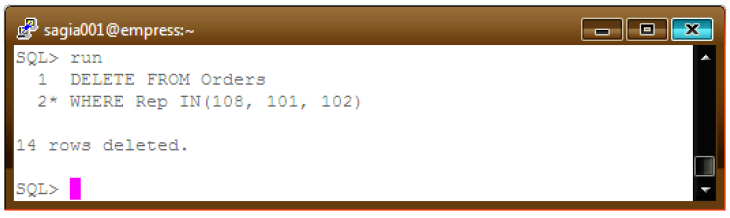


Question 17: Delete all orders for employees 108, 101, 102.

Query 17:

```
DELETE FROM Orders
WHERE Rep IN(108, 101, 102);
```

Result 17:

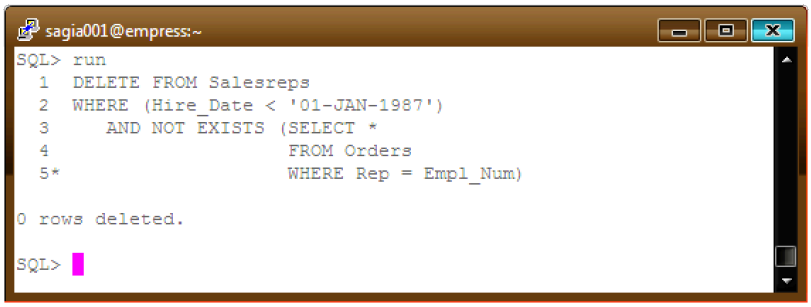


Question 18: Delete all sales reps that have no orders and were hired before Jan 1 1987.

Query 18:

```
DELETE FROM Salesreps
WHERE (Hire_Date < '01-JAN-1987')
       AND NOT EXISTS (SELECT *
                       FROM Orders
                       WHERE Rep = Empl_Num);
```

Result 18:



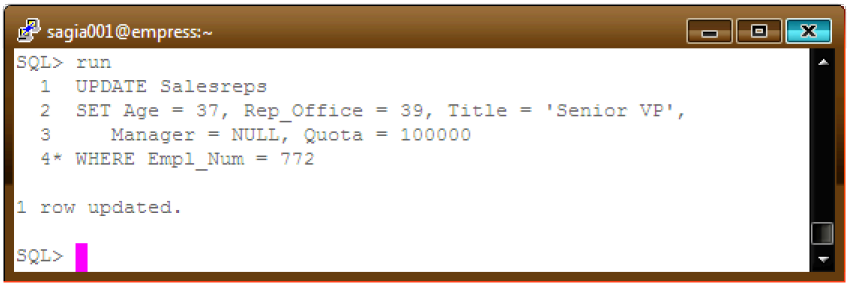
Question 19: Update your employee record with the following:

Age: 37	Rep_Office:39	Title: Senior VP
Manager: NULL	Quota: 100000	

Query 19:

```
UPDATE Salesreps
SET Age = 37, Rep_Office = 39, Title = 'Senior VP',
      Manager = NULL, Quota = 100000
WHERE Empl_Num = 772;
```

Result 19:

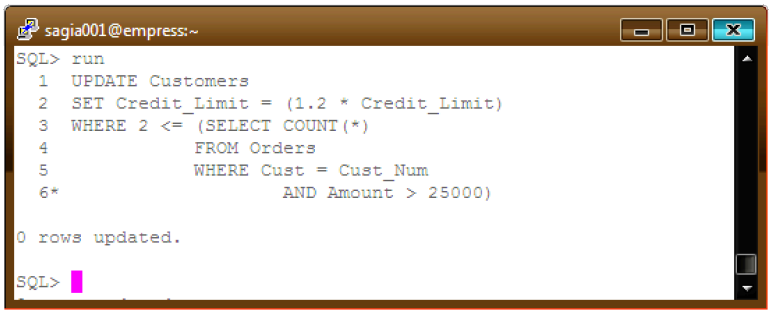


Question20: Increase customers credit limit by 20% for all customers that have 2 or more orders in which each order is more than 25,000.

Query 20:

```
UPDATE Customers
SET Credit_Limit = (1.2 * Credit_Limit)
WHERE 2 <=    (SELECT COUNT(*)
                    FROM Orders
                    WHERE Cust = Cust_Num AND Amount > 25000);
```

Result 20:

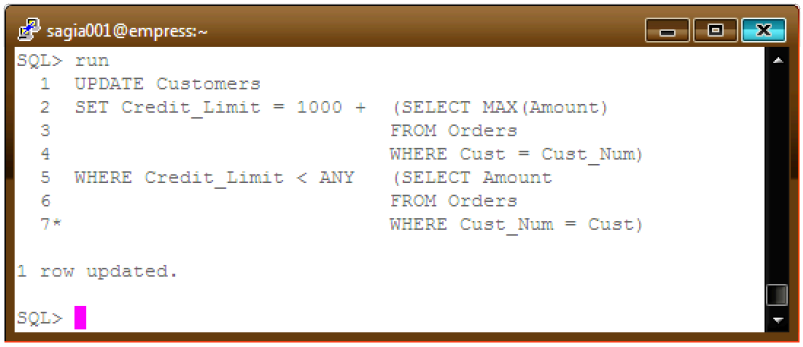


Question 21: Increase the credit limit of any customer who has any order that exceeds their credit limit. The new credit limit should be set to their maximum order amount plus $1,000. This must be done in 1 SQL statement.

Query 21:

```
UPDATE Customers
SET Credit_Limit = 1000 +       (SELECT MAX(Amount)
                                 FROM Orders
                                 WHERE Cust = Cust_Num)
WHERE Credit_Limit < ANY        (SELECT Amount
                                 FROM Orders
                                 WHERE Cust_Num = Cust)
```

Result 21:

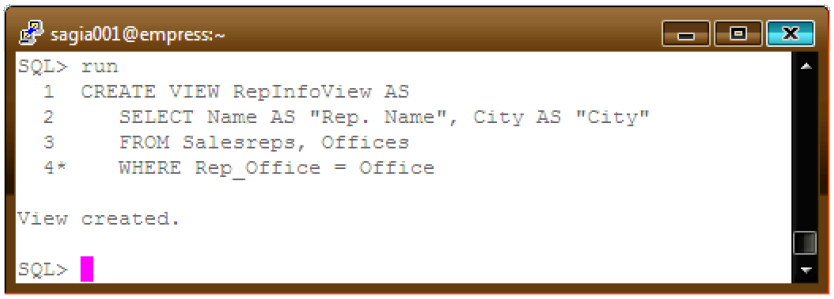


Question 22: Create a view to show the Sales rep Name, and city that the Sales rep works in.

Query 22:

```
CREATE VIEW RepInfoView AS
       SELECT Name AS “Rep. Name”, City AS “City”
       FROM Salesreps, Offices
       WHERE Rep_Office = Office;
```

Result 22:

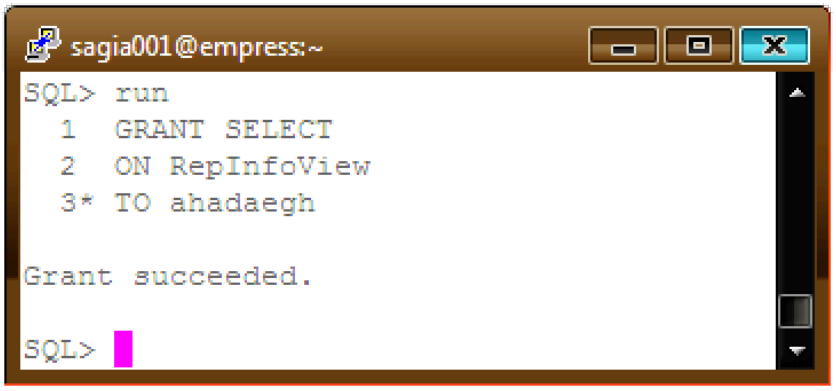


Question 23: Grant select access of the view created in question 22 to your Database instructors: Ahmad R. Hadaegh (with user id ahadaegh).

Query 23:

```
GRANT SELECT
ON RepInfoView
TO ahadaegh;
```

Result 23:

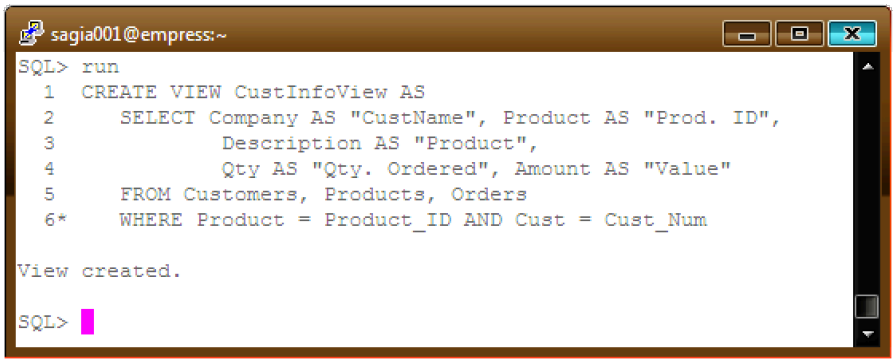


Question 24: Create a view to show the customer name, product, description, quantity ordered and value of parts ordered. The column heading for the customers name should be 'CustName' and the column heading for value of parts ordered should be 'Value'.

Query 24:

```
CREATE VIEW CustInfoView AS
        SELECT Company AS "CustName", Product AS "Prod. ID",
                Description AS "Product",
                Qty AS "Qty. Ordered", Amount AS "Value"
        FROM Customers, Products, Orders
        WHERE Product = Product_ID AND Cust = Cust_Num;
```

Result 24:

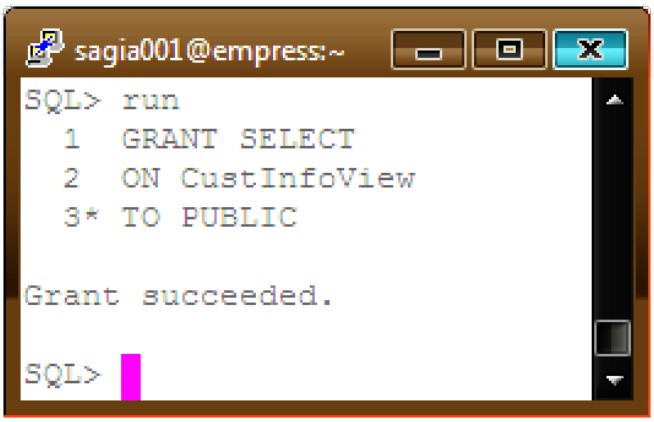


Question 25: Grant select access of the view created in question 24 to public.

Query 25:

```
GRANT SELECT
ON CustInfoView
TO PUBLIC;
```

Result 25:

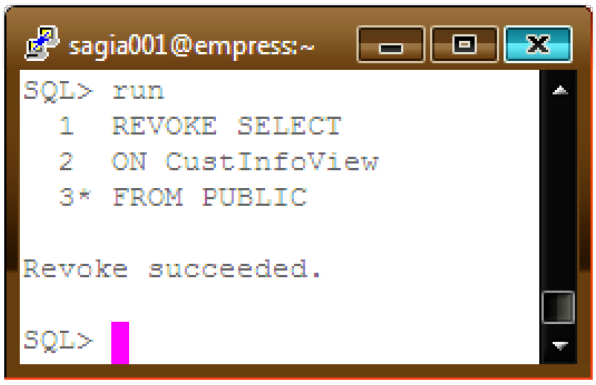


Question 26: Revoke access on view created in question 24 from Public.

Query 26:

```
REVOKE SELECT
ON CustInfoView
FROM PUBLIC;
```

Result 26:

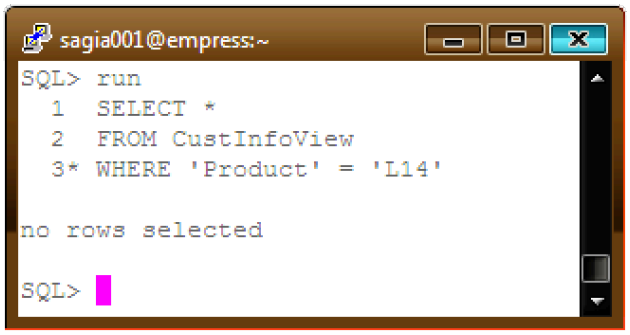


Question 27: Using the view created in question 24 above, list all information for product 'L14'.

Query 27:

```
SELECT *
FROM CustInfoView
WHERE 'Product' = 'L14';
```

Result 27:

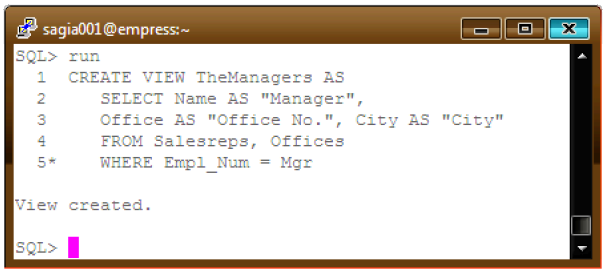


Question 28: Create a view called TheManagers to list the name of all sales reps that manage some office. Along with the managers name, list the office number and city for each office.

Query 28:

```
CREATE VIEW TheManagers AS
       SELECT Name AS “Manager”, Office AS “Office No.”, City AS “City”
       FROM Salesreps, Offices
       WHERE Empl_Num = Mgr;
```

Result 28:

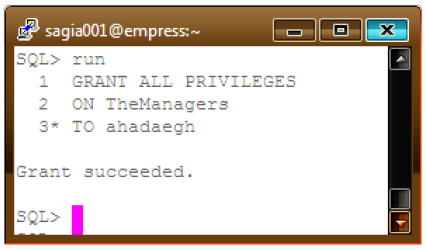


Question 29: Grant all privileges on the view created in question 28 to your instructor.

Query 29:

```
GRANT ALL PRIVILEGES
ON TheManagers
TO ahadaegh;
```

Result 29:



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

Query 30:

```
GRANT SELECT, INSERT, UPDATE
ON Offices
TO jschmidt, kmart;
```

Result 30: