Name: Omid Azodi

Professor: Dr. Hadaegh

Class: CS443

Due Date: 04/05/2016

**Assignment: Assignment #2**

**Question 1:**

1. Draw your ERD based on the above assumptions and data



1. Draw tables from the initial ERD and normalize them. Place all tables in 3rd normal form (if necessary)

**Initial tables from initial ERD:**

Hotel(hNum, hAdd, hPhone, hNoOfRooms, mID)

Room(rNum, hNum, rLoc, rRate, rOccupied, rSize)

Manager(mID, mAdd, mEmail, mPhone, mSal, mName)

CareTaker(ctID, ctEmail, ctSal, ctPhone, ctName, ctAdd, hNum)

Customer(cNum, cAdd, cName, cEndDate, cAmtOwing, cSSN, cCreditCard, cStartDate, hNum, rNum)

**Tables in 3rd normal form:**

* I removed cAmtOwing from the Customers table because, it is a derived dependency that can be calculated/derived from subtracting cEndDate – cStartDate both from the derived dependency and multiply that by the rRate.

Hotel(hNum, hAdd, hPhone, hNoOfRooms, mID)

Room(rNum, hNum, rLoc, rRate, rOccupied, rSize)

Manager(mID, mAdd, mEmail, mPhone, mSal, mName)

CareTaker(ctID, ctEmail, ctSal, ctPhone, ctName, ctAdd, hNum)

Customer(cNum, cAdd, cName, cEndDate, cSSN,cCreditCard, cStartDate, hNum, rNum)

1. Revise your ERD (if necessary)



1. Create your tables based on the given types and restrictions using the following rules:

CREATE TABLE Hotel

(

hNum INTEGER,

hAdd VARCHAR2(50),

mID INTEGER,

hPhone CHAR(7)

CONSTRAINT uniq\_hPhone UNIQUE,

hNoOfRooms INTEGER,

CONSTRAINT pk\_hNum PRIMARY KEY (hNum),

CONSTRAINT check\_hNoOfRooms CHECK (hNoOfRooms>0 AND hNoOfRooms <=200),

CONSTRAINT check\_hPhone CHECK (hPhone>='2222222' AND hPhone <='9999999')

);

CREATE TABLE Room

(

rNum INTEGER,

rLoc INTEGER,

rRate NUMBER(5,2),

rOccupied CHAR(5),

hNum INTEGER,

rSize INTEGER,

CONSTRAINT pk\_rNum PRIMARY KEY (rNum, hNum),

CONSTRAINT check\_rNum CHECK (rNum>0 AND rNum<=200),

CONSTRAINT check\_rRate CHECK (rRate>=50),

CONSTRAINT check\_rSize CHECK (rSize>=2 AND rSize<=4),

CONSTRAINT check\_rOccupied CHECK (rOccupied = 'false' OR rOccupied = 'true'),

CONSTRAINT check\_rLoc CHECK (rLoc>0 AND rLoc<=100)

);

CREATE TABLE Manager

(

mID INTEGER,

mAdd VARCHAR2(50),

mEmail VARCHAR2(200),

mPhone CHAR(7),

mSal NUMBER(7,2),

mName VARCHAR2(50),

CONSTRAINT pk\_mID PRIMARY KEY (mID),

CONSTRAINT check\_mID CHECK(mID>=111111 AND mID<=999999),

CONSTRAINT check\_mPhone CHECK(mPhone>= '2222222' AND mPhone<= '9999999')

);

CREATE TABLE CareTaker

(

ctID INTEGER,

ctEmail VARCHAR2(200),

ctSal NUMBER(7,2),

hNum INTEGER,

ctPhone CHAR(7),

ctName VARCHAR2(50),

ctAdd VARCHAR2(50),

CONSTRAINT pk\_ctID PRIMARY KEY (ctID),

CONSTRAINT check\_ctID CHECK (ctID>=111111 AND ctID<=999999),

CONSTRAINT check\_ctPhone CHECK (ctPhone>= '2222222' AND ctPhone <= '9999999'),

CONSTRAINT check\_ctSal CHECK (ctSal>20000 AND ctSal<40000)

);

CREATE TABLE Customer

(

cNum INTEGER,

cAdd VARCHAR2(50),

hNum INTEGER,

rNum INTEGER,

cName VARCHAR2(50),

cEndDate DATE,

cSSN CHAR(9)

CONSTRAINT uniq\_cSSN UNIQUE,

cCreditCard VARCHAR2(25),

cStartDate DATE,

CONSTRAINT pk\_cNum PRIMARY KEY (cNum),

CONSTRAINT check\_cSSN CHECK (cSSN > '600000000' AND cSSN <= '699999999')

);

ALTER TABLE CONSTRAINTS:

ALTER TABLE Hotel ADD CONSTRAINT fk\_mIDForHotel FOREIGN KEY(mID) REFERENCES Manager(mID);

ALTER TABLE Room ADD CONSTRAINT fk\_hNumForRoom FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

ALTER TABLE CareTaker ADD CONSTRAINT fk\_hNumForCareTaker FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

ALTER TABLE Customer ADD CONSTRAINT fk\_hNumrNumForCustomer FOREIGN KEY(rNum,hNum) REFERENCES Room(rNum,hNum);

ALTER TABLE Customer ADD CONSTRAINT fk\_hNumForCustomer FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

**Question 2:**

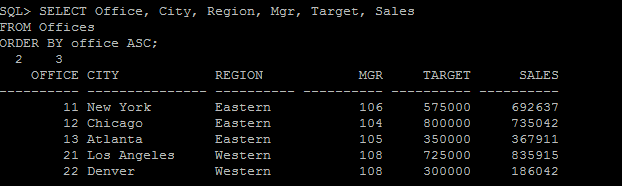
Do the following SQL questions. The resulting columns must all have descriptive names

1. Write a select statement to list all the columns from the Offices table. 'Select \*' is not allowed. Return the list in Office order.

SELECT Office, City, Region, Mgr, Target, Sales

FROM Offices

ORDER BY office ASC;

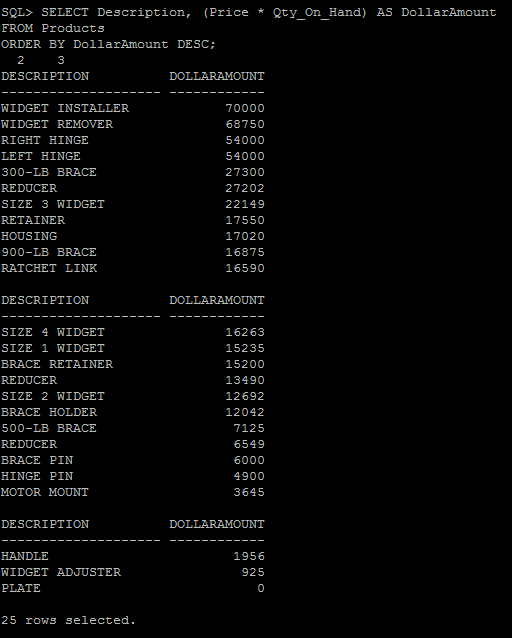


1. List the Product Name (its description), and dollar value of quantity on hand (price \* quantity) . Return the list in descending order by value.

SELECT Description, (Price \* Qty\_On\_Hand) AS DollarAmount

FROM Products

ORDER BY DollarAmount DESC;

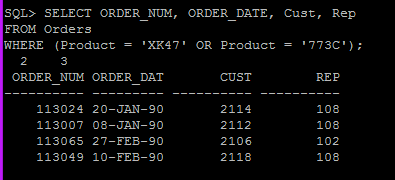


1. List the Order Number, Order Date, Customer Number and Sales Rep Number for orders for part 'XK47' or '773C'. (Use a compound search condition - OR.)

SELECT ORDER\_NUM, ORDER\_DATE, Cust, Rep

FROM Orders

WHERE (Product = 'XK47' OR Product = '773C');

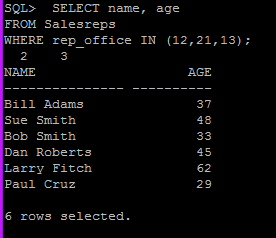


1. List the Name and Age for all Sales Reps in the following offices: 12; 21; 13. (Use the set membership test - IN.)

SELECT name, age

FROM Salesreps

WHERE rep\_office IN (12,21,13);

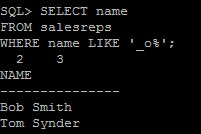


1. List the names of all Sales Reps who have the letter 'o' (this is lower case o) as the second character of their name.

SELECT name

FROM salesreps

WHERE name LIKE '\_o%';

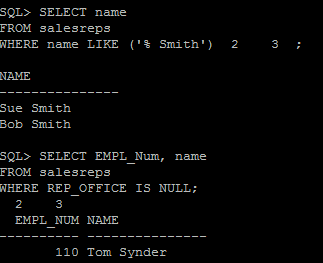


1. Return the Sales Rep ID and Name of any Sales Rep not assigned to an office yet.

SELECT EMPL\_Num, name

FROM salesreps

WHERE REP\_OFFICE IS NULL;

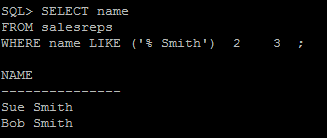


1. Show all the sales rep names with last name “Smith”.

SELECT name

FROM salesreps

WHERE name LIKE ('% Smith');

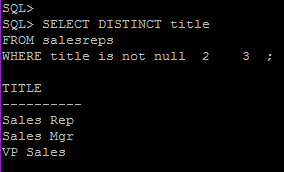


1. List different titles in the sales reps table. Only list each title once and unknown titles should be ignored.

SELECT DISTINCT title

FROM salesreps

WHERE title is not null;

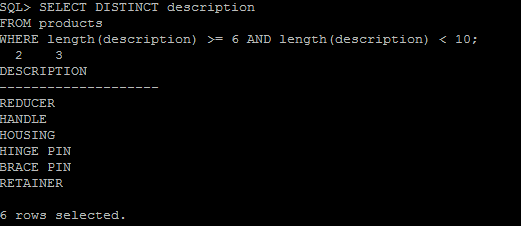


1. List the description of the products which are at least 6 character and less than 10 character long. No duplicate row is allowed. You can use the build in function length() to do this. For example, length(name) return the number of characters for attribute called “name”.

SELECT DISTINCT description

FROM products

WHERE length(description) >= 6 AND length(description) < 10;

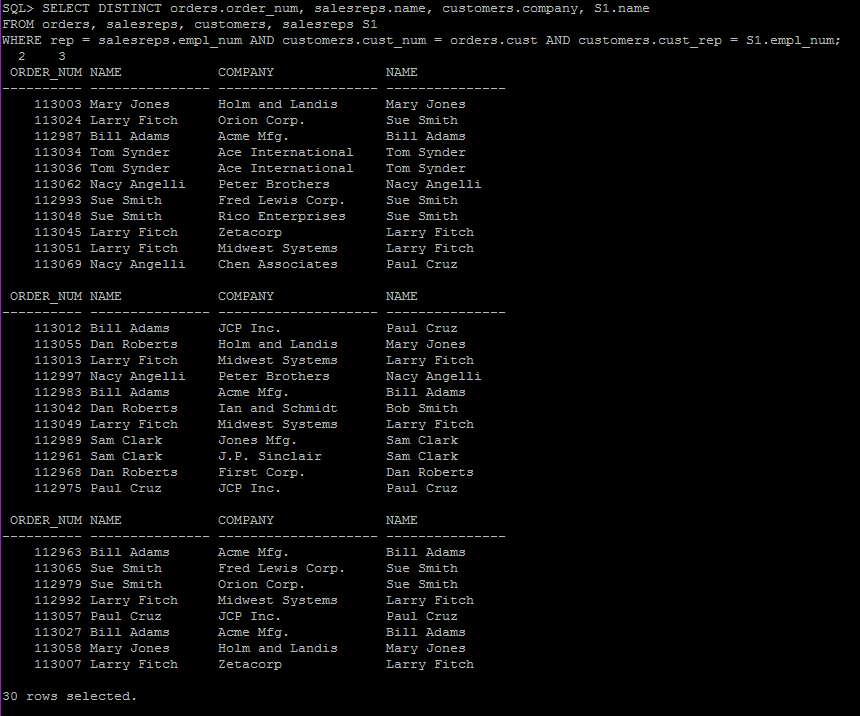


1. List the the order nums with the name of the rep who placed the order and the name of the customer who made the order and the name of the rep for that customer

SELECT DISTINCT orders.order\_num, salesreps.name, customers.company, S1.name

FROM orders, salesreps, customers, salesreps S1

WHERE rep = salesreps.empl\_num AND customers.cust\_num = orders.cust AND customers.cust\_rep = S1.empl\_num;



**Question 3:**

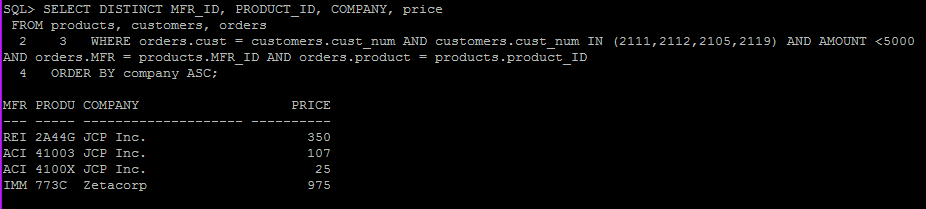
1. List the Mfr Id, the Product Id, Company and PRICE of all products brought by customers where customer number is one of (2111,2112,2105,2119) and where the amount of the order < $5000.00. Order the results by ascending Company.

SELECT DISTINCT MFR\_ID, PRODUCT\_ID, COMPANY, price

FROM products, customers, orders

WHERE orders.cust = customers.cust\_num AND customers.cust\_num IN (2111,2112,2105,2119) AND AMOUNT <5000 AND orders.MFR = products.MFR\_ID AND orders.product = products.product\_ID

ORDER BY company ASC;



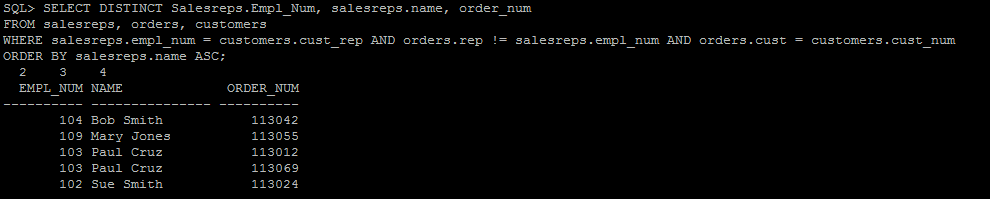
1. List all salesreps (id and names) and all orders (orderNumber) in which the salesrep is the company’s (i.e. the customer) rep (Cust Rep), but didn’t take the order. Order the result based on name in ascending order.

SELECT DISTINCT Salesreps.Empl\_Num, salesreps.name, order\_num

FROM salesreps, orders, customers

WHERE salesreps.empl\_num = customers.cust\_rep AND orders.rep != salesreps.empl\_num AND orders.cust = customers.cust\_num

ORDER BY salesreps.name ASC;



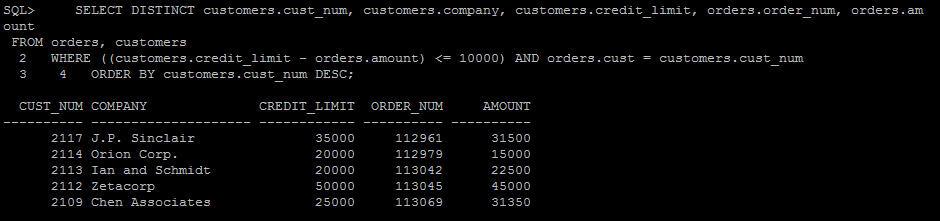
1. List all customers (Customer number, Company, and Credit Limit) and orders (Order Number, Amount) where the order is within $10000.00 of the Credit Limit (less than or equal to $10000). Sort the result by Customer number in descending order.

SELECT DISTINCT customers.cust\_num, customers.company, customers.credit\_limit, orders.order\_num, orders.amount

FROM orders, customers

WHERE ((customers.credit\_limit - orders.amount) <= 10000) AND orders.cust = customers.cust\_num

ORDER BY customers.cust\_num DESC;

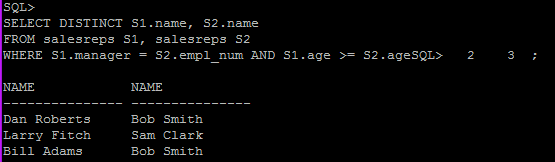


1. List all salesreps names and their managers’ names where the salesrep is at least as old as the manager.

SELECT DISTINCT S1.name, S2.name

FROM salesreps S1, salesreps S2

WHERE S1.manager = S2.empl\_num AND S1.age >= S2.age;



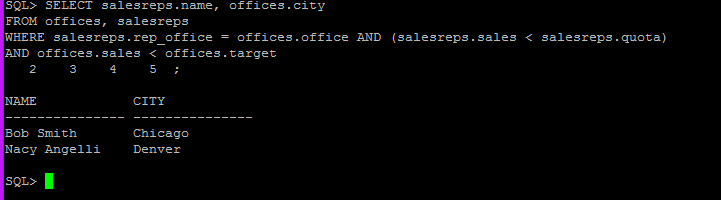
1. List all salesreps (Name) and the City they work in where the sales of the salesrep < Quota and the sales for the office is < Target.

SELECT salesreps.name, offices.city

FROM offices, salesreps

WHERE salesreps.rep\_office = offices.office AND (salesreps.sales < salesreps.quota)

AND offices.sales < offices.target;



1. List the name, id, and hire date of the salesreps where at least one of the two conditions hold:

* The salesrep sales should be greater than the quota
* The salesrep has taken an order from Customer number 2117, 2111, or 2101.

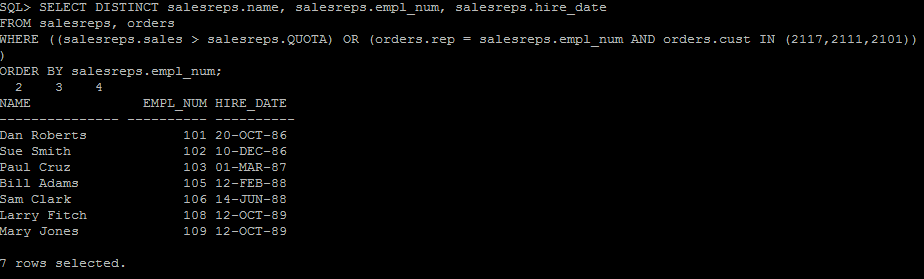
Sort the result by the salesrep’s id.

SELECT DISTINCT salesreps.name, salesreps.empl\_num, salesreps.hire\_date

FROM salesreps, orders

WHERE ((salesreps.sales > salesreps.QUOTA) OR (orders.rep = salesreps.empl\_num AND orders.cust IN (2117,2111,2101)))

ORDER BY salesreps.empl\_num;

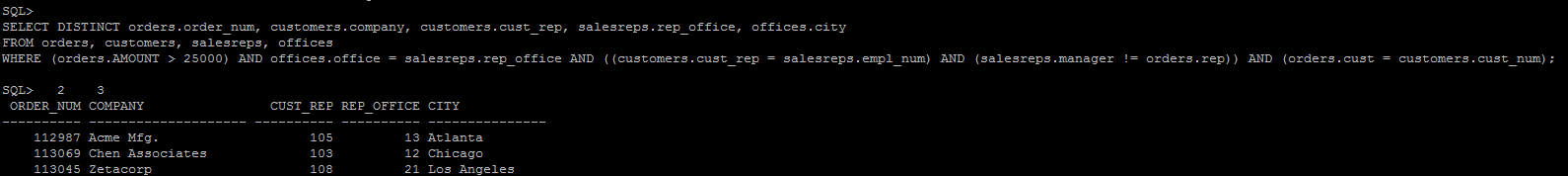


1. List all orders (Order Number) over $25000 showing the company placing the order, the Customer Rep assign to the company, the Office id and the city where the Customer Rep works in, such that the Customer Rep’s manager is not the person who actually took the order.

SELECT DISTINCT orders.order\_num, customers.company, customers.cust\_rep, salesreps.rep\_office, offices.city

FROM orders, customers, salesreps, offices

WHERE (orders.AMOUNT > 25000) AND offices.office = salesreps.rep\_office AND ((customers.cust\_rep = salesreps.empl\_num) AND (salesreps.manager != orders.rep)) AND (orders.cust = customers.cust\_num);



1. List all customer reps (their name and their id) and their managers name in which the manager has taken an order for the customer Rep’s company or the manager is based in New York or Denver. Use appropriate column header

SELECT DISTINCT salesreps.empl\_num, salesreps.name, S1.name

FROM salesreps, salesreps S1, orders, offices, customers

WHERE salesreps.empl\_num = customers.cust\_rep

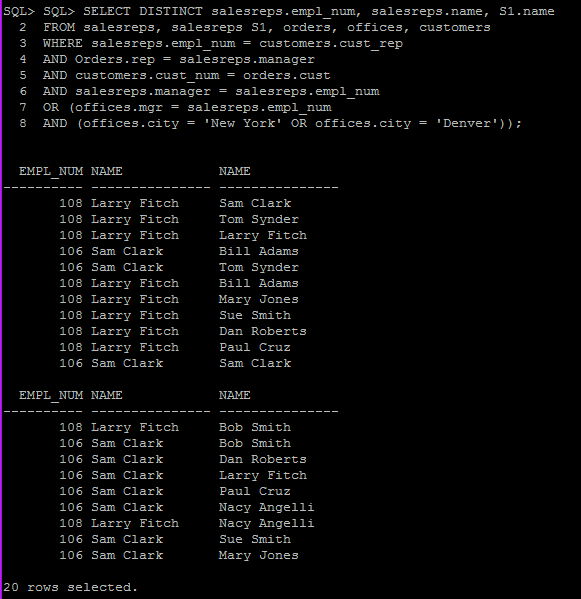
AND Orders.rep = salesreps.manager

AND customers.cust\_num = orders.cust

AND salesreps.manager = salesreps.empl\_num

OR (offices.mgr = salesreps.empl\_num

AND (offices.city = 'New York' OR offices.city = 'Denver'));



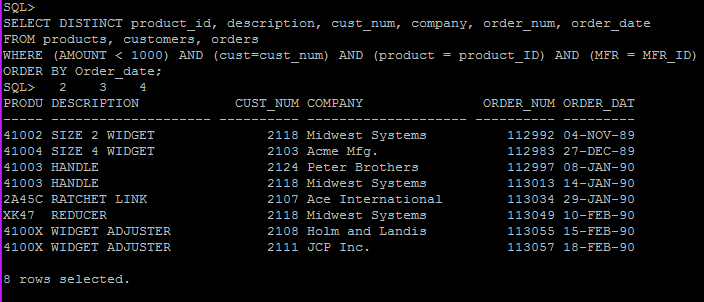
1. List all products (ProductId, and Description), customers (CustNum, Company) who have bought that product, and orders (Order Number, and Order Date) where the order < $1000. Sort the rows by the OrderDate.

SELECT DISTINCT product\_id, description, cust\_num, company, order\_num, order\_date

FROM products, customers, orders

WHERE (AMOUNT < 1000) AND (cust=cust\_num) AND (product = product\_ID) AND (MFR = MFR\_ID)

ORDER BY Order\_date;



1. List the name of the salesreps and the name of their managers only if the manager has taken care of some orders.

SELECT DISTINCT salesreps.name, S1.name

FROM salesreps, salesreps S1, orders

WHERE (salesreps.manager = S1.empl\_num) AND (orders.rep = S1.empl\_num);

