Chris Banci

Dr. Hadaegh

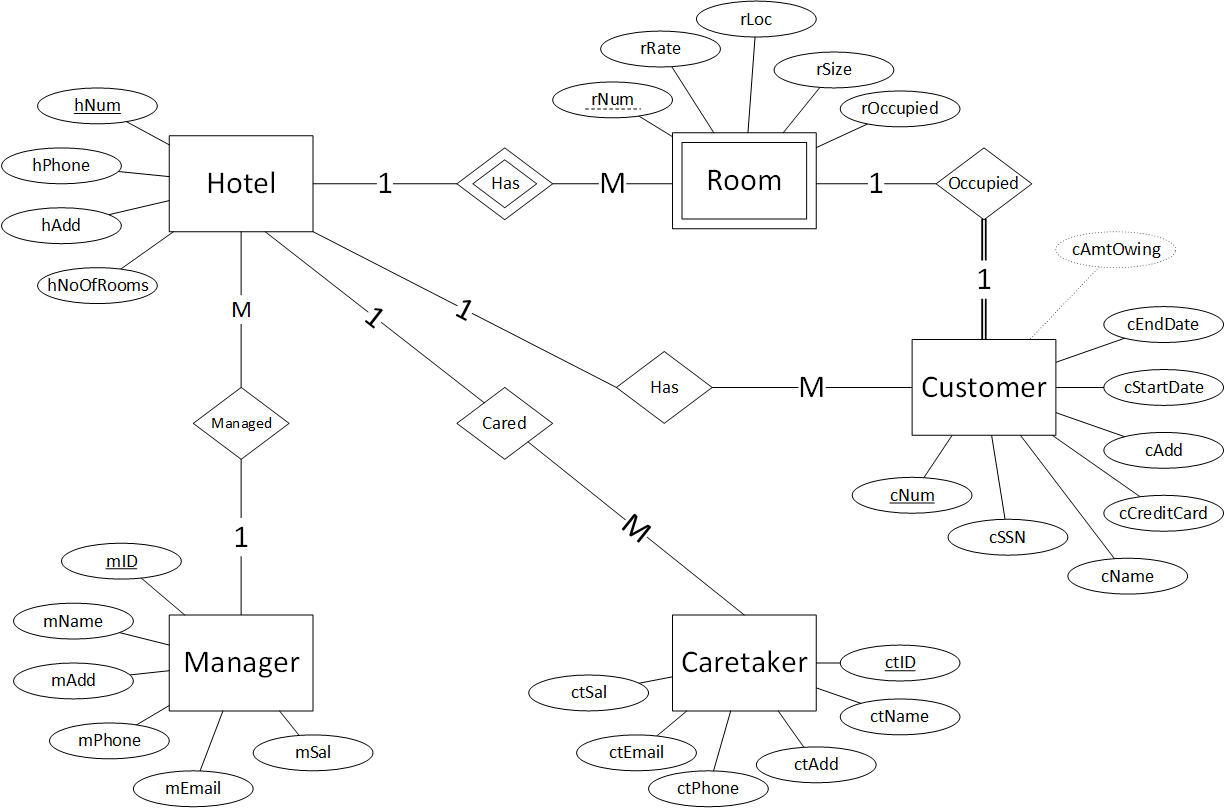
CS443

3/28/17

Assignment #2

**Question 1:**

**A) Draw your ERD based on the above assumptions and data**



\*note: cAmtOwing is derived attribute in initial ERD.

**B) Draw tables from the initial ERD and normalize them**

Initial tables:

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

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

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

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

Caretaker (ctID, ctName, ctAdd, ctPhone, ctEmail, ctSal, hNum\*)

Normalized tables:

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

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

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

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

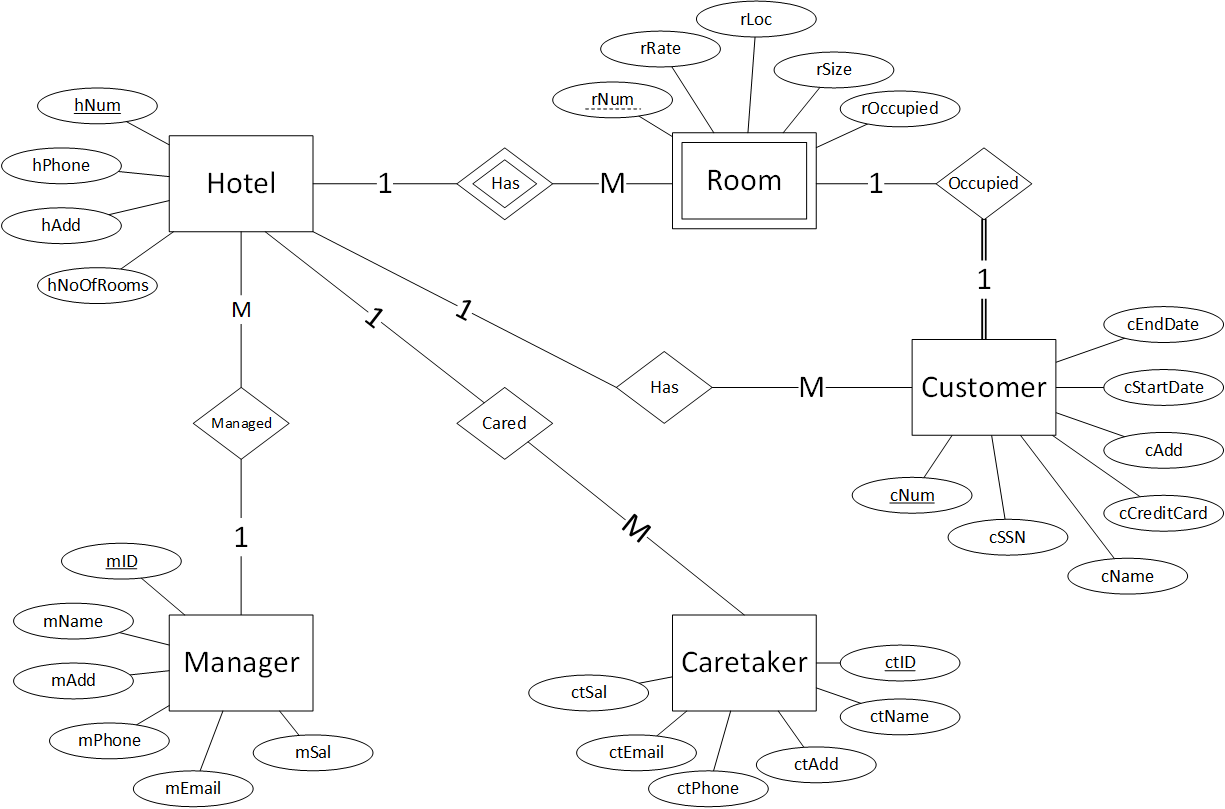
Caretaker (ctID, ctName, ctAdd, ctPhone, ctEmail, ctSal, hNum\*)

cAmtOwing is removed from the customer table because it can be calculated or derived from the attributes of rRate, cStartDate, and cEndDate, by using the formula:

* *cAmtOwing = (cEndDate - cStartDate) \* rRate*

The tables are now in third normal form because no transitive or derived dependencies exist.

**C) Revise your ERD (if necessary)**



**D) Create your tables based on the givetypes and restrictions using the following rules**

DROP TABLE Hotel CASCADE CONSTRAINTS;

CREATE TABLE Hotel (

hNum NUMBER,

hNoOfRooms NUMBER,

hPhone CHAR(7) CONSTRAINT hPhone\_UQ UNIQUE,

hAdd VARCHAR2(50),

mID NUMBER,

CONSTRAINT Hotel\_PK PRIMARY KEY (hNum),

CONSTRAINT room\_CK CHECK(hNoOfRooms > 0 AND hNoOfRooms <= 200),

CONSTRAINT hPhone\_CK CHECK(hPhone >= '2202222' and hPhone <= '9909999')

);

DROP TABLE Room CASCADE CONSTRAINTS;

CREATE TABLE Room (

rNum NUMBER,

rRate NUMBER(5, 2),

rSize NUMBER,

rLoc NUMBER,

rOccupied CHAR(5),

hNum NUMBER,

CONSTRAINT Room\_PK PRIMARY KEY(rNum, hNum),

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

CONSTRAINT rRate\_CK CHECK(rRate >= 50),

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

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

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

);

DROP TABLE Customer CASCADE CONSTRAINTS;

CREATE TABLE Customer (

cNum NUMBER,

cSSN CHAR(9) CONSTRAINT cSSN\_UQ UNIQUE,

cName VARCHAR2(50),

cCreditCard VARCHAR2(25),

cAdd VARCHAR2(50),

cStartDate DATE,

cEndDate DATE,

rNum NUMBER,

hNum NUMBER,

CONSTRAINT Customer\_PK PRIMARY KEY(cNum),

CONSTRAINT cSSN\_CK CHECK(cSSN > '660000000' AND cSSN <= '609999999')

);

DROP TABLE Manager CASCADE CONSTRAINTS;

CREATE TABLE Manager (

mID NUMBER,

mName VARCHAR2(50),

mAdd VARCHAR2(50),

mPhone CHAR(7),

mEmail VARCHAR2(200),

mSal NUMBER(7, 2),

CONSTRAINT Manager\_PK PRIMARY KEY(mID),

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

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

);

DROP TABLE Caretaker CASCADE CONSTRAINTS;

CREATE TABLE Caretaker (

ctID NUMBER,

ctName VARCHAR2(50),

ctAdd VARCHAR2(50),

ctPhone CHAR(7),

ctEmail VARCHAR2(200),

ctSal NUMBER(7, 2),

hNum NUMBER,

CONSTRAINT Caretaker\_PK PRIMARY KEY(ctID),

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

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

CONSTRAINT ctSal\_CK CHECK(ctSal > 20000 AND ctSal < 40000)

);

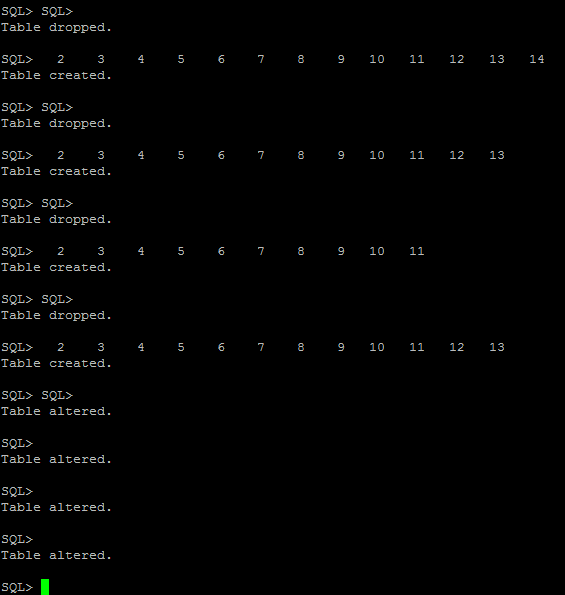
ALTER TABLE Hotel ADD CONSTRAINT Hotel\_FK1 FOREIGN KEY(mID) REFERENCES Manager(mID);

ALTER TABLE Room ADD CONSTRAINT Room\_FK1 FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

ALTER TABLE Customer ADD CONSTRAINT Customer\_FK1 FOREIGN KEY(rNum, hNum) REFERENCES Room(rNum, hNum);

ALTER TABLE Customer ADD CONSTRAINT Customer\_FK2 FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

ALTER TABLE Caretaker ADD CONSTRAINT Caretaker\_FK1 FOREIGN KEY(hNum) REFERENCES Hotel(hNum);



**Question 2:**

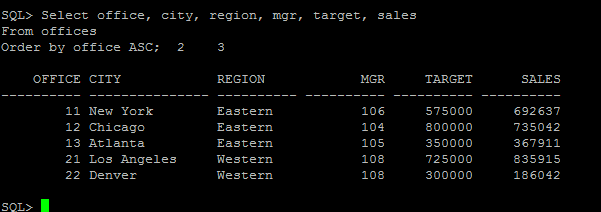
Do the following SQL questions. The resulting columns must all have descriptive names

**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;

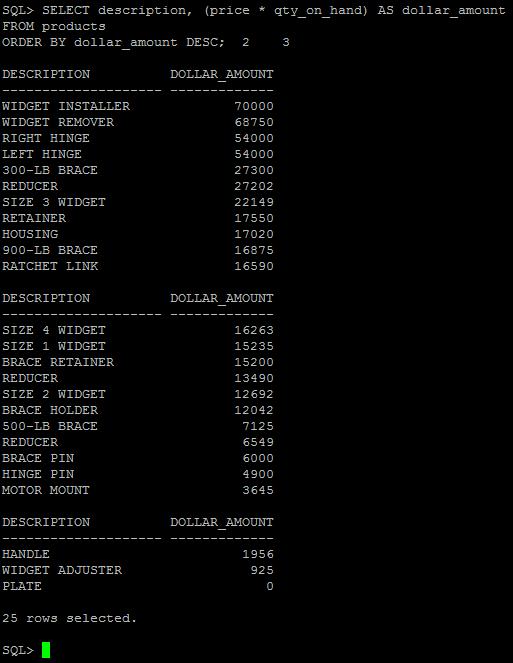


**2) List the Product Name (its description), and dollar value of quantity on hand (price \* quantity) . Return the list in descending order by value.**

SELECT description, (price \* qty\_on\_hand) AS dollar\_amount

FROM products

ORDER BY dollar\_amount DESC;

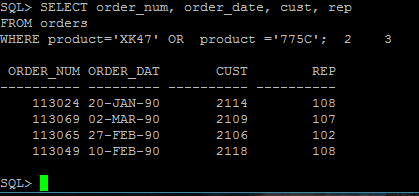


**3) New: List the Order Number, Order Date, Customer Number and Sales Rep Number for orders for part 'XK47' or '775C'. (Use a compound search condition - OR.)**

SELECT order\_num, order\_date, cust, rep

FROM orders

WHERE product='XK47' OR product ='775C';

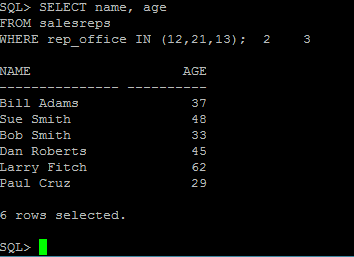


**4) List the Name and Age for all Sales Reps in the following offices: 12; 21; 13. (Use the set membership test - IN.)**

SELECT name, age

FROM salesreps

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

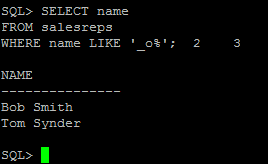


**5) List the names of all Sales Reps who have the letter 'o' (this is lower case o) as the second character of their name.**

SELECT name

FROM salesreps

WHERE name LIKE '\_o%';

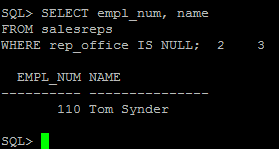


**6) Return the Sales Rep ID and Name of any Sales Rep not assigned to an office yet.**

SELECT empl\_num, name

FROM salesreps

WHERE rep\_office IS NULL;

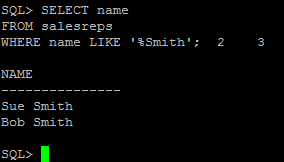


**7) Show all the sales rep names with last name “Smith”.**

SELECT name

FROM salesreps

WHERE name LIKE '%Smith';

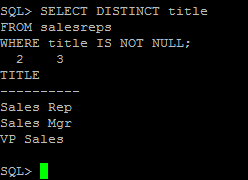


**8) List different titles in the sales reps table. Only list each title once and unknown titles should be ignored.**

SELECT DISTINCT title

FROM salesreps

WHERE title IS NOT NULL;

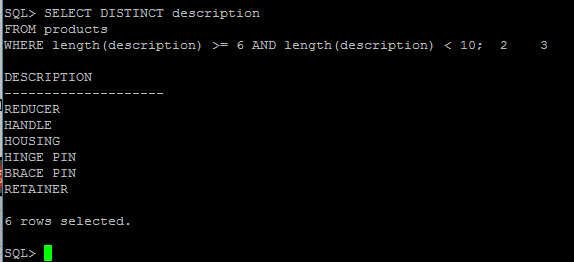


**9) List the description of the products which are at least 6 character and less than 10 character long. No duplicate row is allowed. You can use the build in function length() to do this. For example, length(name) return the number of characters for attribute called “name”.**

SELECT DISTINCT description

FROM products

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

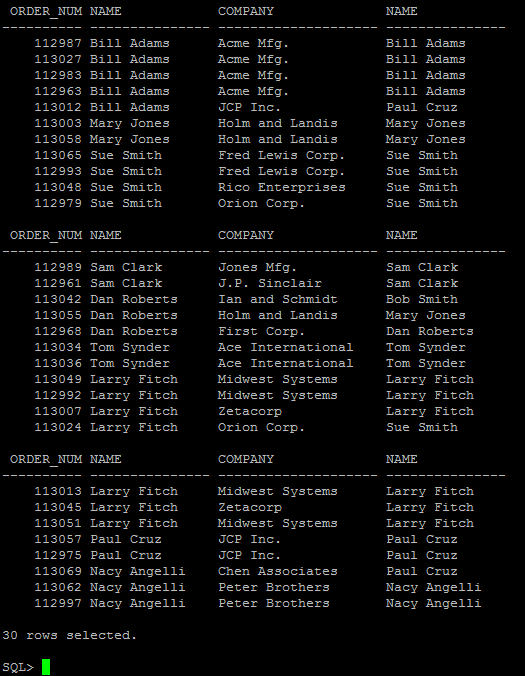


**10) List the the order nums with the name of the rep who placed the order and the name of the customer who made the order and the name of the rep for that customer**

SELECT order\_num, salesreps.name, company, cr.name

FROM orders, salesreps, customers, salesreps cr

WHERE rep = salesreps.empl\_num AND cust\_num = cust AND cust\_rep = cr.empl\_num;



**Question 3:**

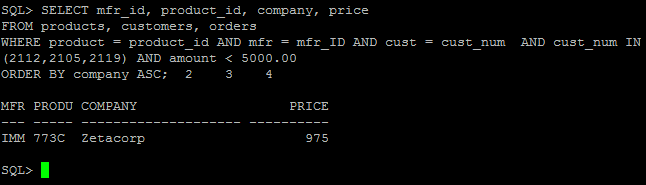
**1) List the Mfr Id, the Product Id, Company and PRICE of all products brought by customers where customer number is one of (2112,2105,2119) and where the amount of the order < $5000.00. Order the results by ascending Company.**

SELECT mfr\_id, product\_id, company, price

FROM products, customers, orders

WHERE product = product\_id AND mfr = mfr\_ID AND cust = cust\_num AND cust\_num IN (2112,2105,2119) AND amount < 5000.00

ORDER BY company ASC;



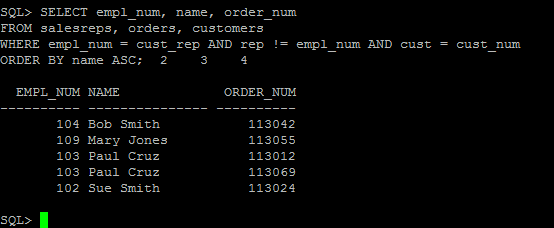
**2) List all salesreps (id and names) and all orders (orderNumber) in which the salesrep is the company’s (i.e. the customer) rep (Cust Rep), but didn’t take the order. Order the result based on name in ascending order.**

SELECT empl\_num, name, order\_num

FROM salesreps, orders, customers

WHERE empl\_num = cust\_rep AND rep != empl\_num AND cust = cust\_num

ORDER BY name ASC;



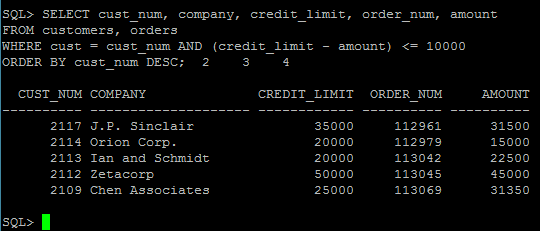
**3) List all customers (Customer number, Company, and Credit Limit) and orders (Order Number, Amount) where the order is within $10000.00 of the Credit Limit (less than or equal to $10000). Sort the result by Customer number in descending order.**

SELECT cust\_num, company, credit\_limit, order\_num, amount

FROM customers, orders

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

ORDER BY cust\_num DESC;

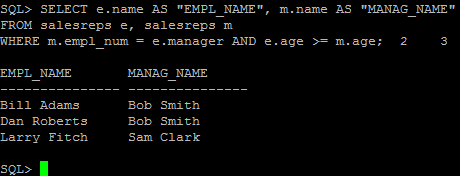


**4) List all salesreps names and their managers’ names where the salesrep is at least as old as the manager.**

SELECT e.name AS "EMPL\_NAME", m.name AS "MANAG\_NAME"

FROM salesreps e, salesreps m

WHERE m.empl\_num = e.manager AND e.age >= m.age;

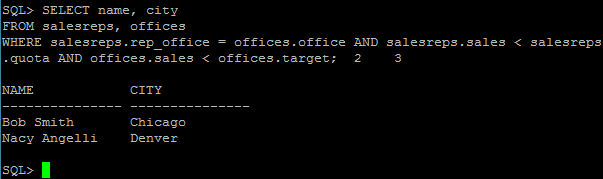


**5) List all salesreps (Name) and the City they work in where the sales of the salesrep < Quota and the sales for the office is < Target.**

SELECT name, city

FROM salesreps, offices

WHERE salesreps.rep\_office = offices.office AND salesreps.sales < salesreps.quota AND offices.sales < offices.target;



**6) 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.**

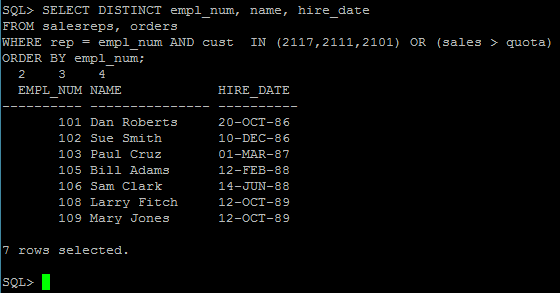
**Sort the result by the salesrep’s id.**

SELECT DISTINCT empl\_num, name, hire\_date

FROM salesreps, orders

WHERE rep = empl\_num AND cust IN (2117,2111,2101) OR (sales > quota)

ORDER BY empl\_num;

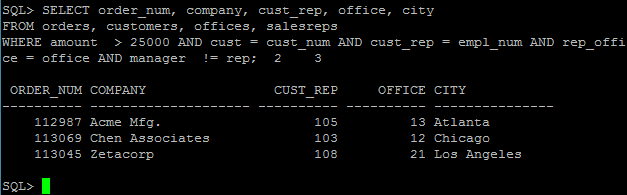


**7) List all orders (Order Number) over $25000 showing the company placing the order, the Customer Rep assign to the company, the Office id and the city where the Customer Rep works in, such that the Customer Rep’s manager is not the person who actually took the order.**

SELECT order\_num, company, cust\_rep, office, city

FROM orders, customers, offices, salesreps

WHERE amount > 25000 AND cust = cust\_num AND cust\_rep = empl\_num AND rep\_office = office AND manager != rep;

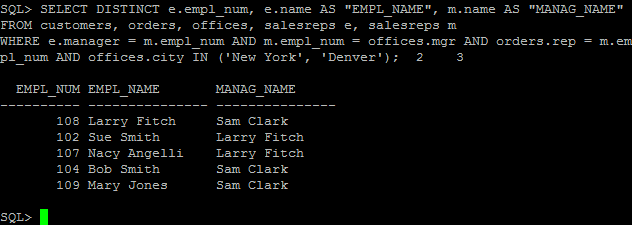


**8) List all customer reps (their name and their id) and their managers name in which the manager has taken an order for the customer Rep’s company or the manager is based in New York or Denver. Use appropriate column header**

SELECT DISTINCT e.empl\_num, e.name AS "EMPL\_NAME", m.name AS "MANAG\_NAME"

FROM customers, orders, offices, salesreps e, salesreps m

WHERE e.manager = m.empl\_num AND m.empl\_num = offices.mgr AND orders.rep = m.empl\_num AND offices.city IN ('New York', 'Denver');



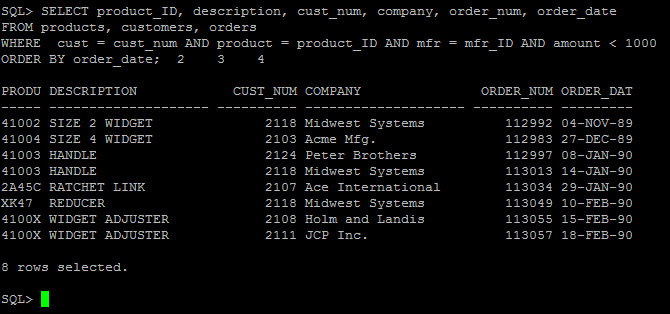
**9) List all products (ProductId, and Description), customers (CustNum, Company) who have bought that product, and orders (Order Number, and Order Date) where the order < $1000. Sort the rows by the OrderDate.**

SELECT product\_ID, description, cust\_num, company, order\_num, order\_date

FROM products, customers, orders

WHERE cust = cust\_num AND product = product\_ID AND mfr = mfr\_ID AND amount < 1000

ORDER BY order\_date;



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

SELECT DISTINCT e.name, m.name

FROM salesreps e, salesreps m, orders

WHERE e.manager = m.empl\_num AND rep = m.empl\_num;

