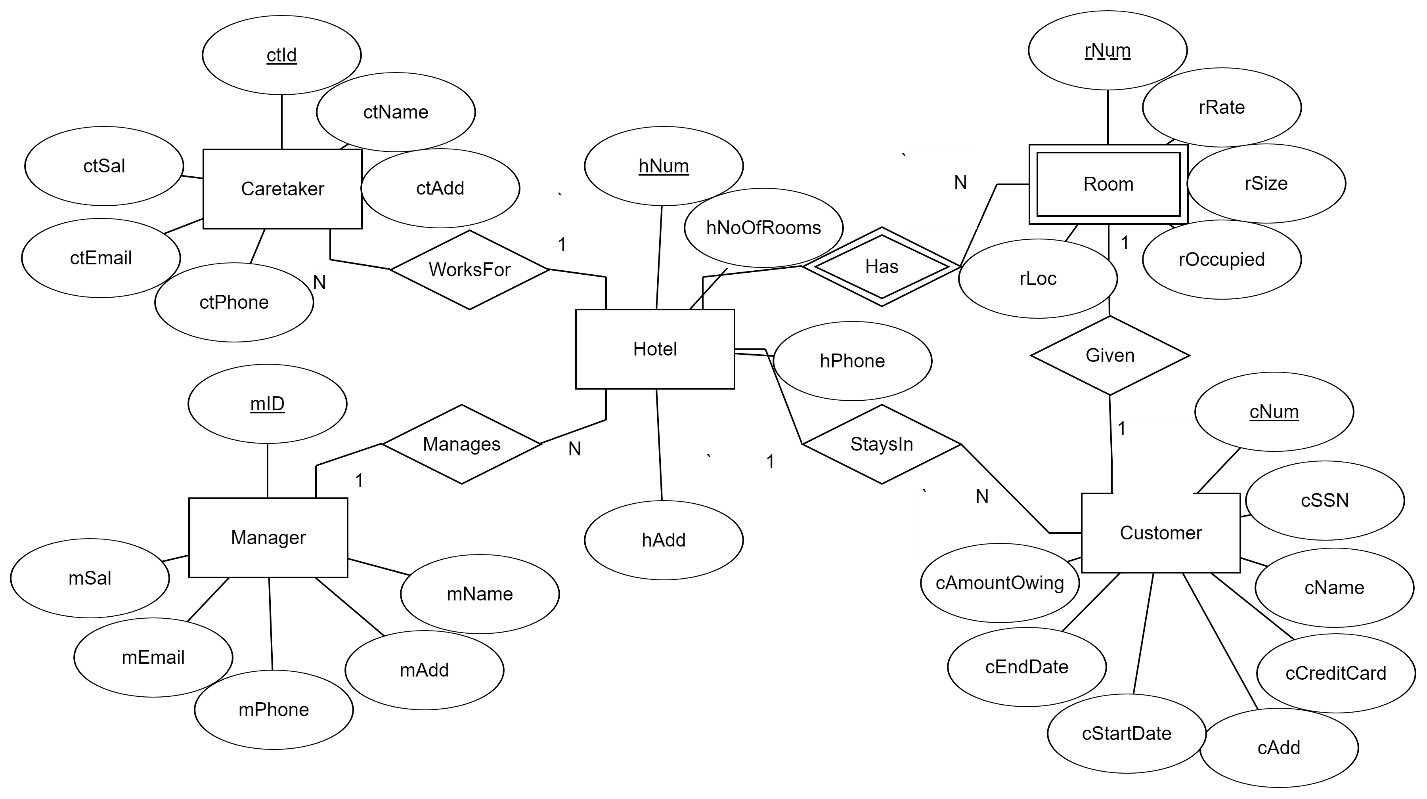
1)

a)

b) Manager(mID, mName, mAddm mPhone, mEmail, mSal)

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

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

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

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

3rd normal form: cAmountOwing was removed since it can be calculated by

Finding out how many days the customer stayed from cStartDate and cEndDate and multiplying that by rRate.

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

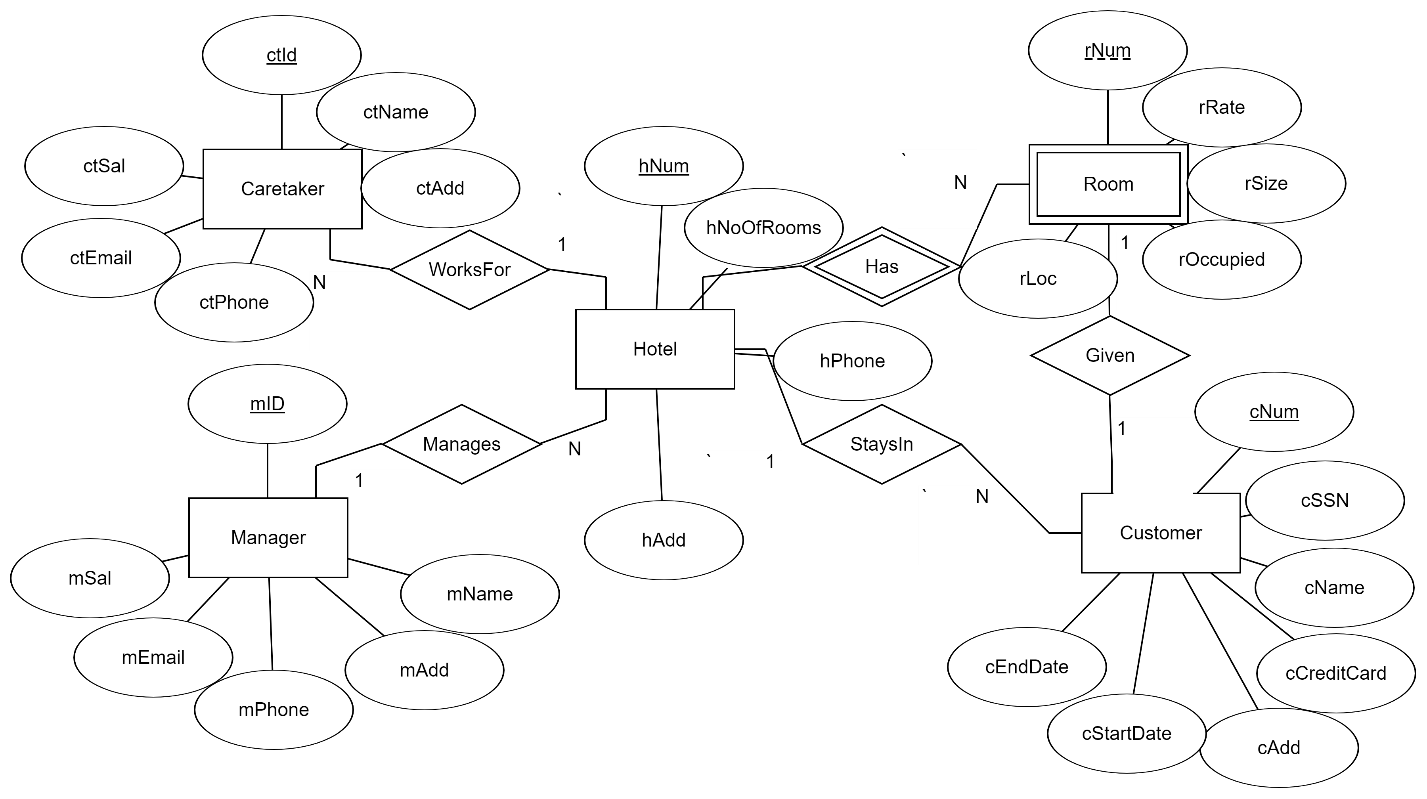
Hotel(hNum, hNoOfRooms, hPhone, hAdd, hMID\*)

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

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

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

C)



D)

DROP TABLE Hotel;

DROP TABLE Manager;

DROP TABLE Room;

DROP TABLE Caretaker;

DROP TABLE Customer;

CREATE TABLE Hotel(

hNum INT CONSTRAINT PK\_hNum PRIMARY KEY,

hNoOfRooms INT CONSTRAINT CH\_NoRooms CHECK(hNoOfRooms > 0 AND hNoOfRooms <= 200),

hPhone CHAR(7) CONSTRAINT UN\_hPhone UNIQUE,

hAdd VARCHAR(50),

hMID INT,

CONSTRAINT CH\_hPhone CHECK(hPhone LIKE '[2-9][2-9][0-9][2-9][2-9][2-9][2-9]')

);

CREATE TABLE Manager(

mID INT CONSTRAINT PK\_mID PRIMARY KEY,

mName VARCHAR(50),

mAdd VARCHAR(50),

mPhone CHAR(7) CONSTRAINT CH\_mPhone CHECK(mPhone LIKE '[2-9][2-9][2-9][2-9][2-9][2-9][2-9]'),

mEmail VARCHAR(100),

mSal DECIMAL(7,2),

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

);

CREATE TABLE Room(

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

hNum INT,

rRate DECIMAL(5,2) CONSTRAINT CH\_rRate CHECK(rRate>=50),

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

rOccupied CHAR(5) CONSTRAINT CH\_rOccupied CHECK(rOccupied In ('true', 'false')),

rLoc INT CONSTRAINT CH\_rLoc CHECK (rLoc > 0 AND rLoc <= 100),

CONSTRAINT PK\_hrNum PRIMARY KEY (hNum, rNum)

);

CREATE TABLE Caretaker(

ctID INT CONSTRAINT PK\_ctID PRIMARY KEY,

ctName VARCHAR(50),

ctAdd VARCHAR(50),

ctPhone CHAR(7) CONSTRAINT CH\_ctPhone CHECK(ctPhone LIKE '[2-9][2-9][2-9][2-9][2-9][2-9][2-9]'),

ctEmail VARCHAR(200),

ctSal INT CONSTRAINT CH\_ctSal CHECK(ctSal > 20000 and ctSal < 40000),

hNum INT

);

CREATE TABLE Customer(

cNum int CONSTRAINT PK\_cNum PRIMARY KEY,

cSSN CHAR(9) CONSTRAINT CH\_cSSN CHECK(cSSN LIKE'[6][06][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

cName VARCHAR(50),

cCreditCard VARCHAR(50),

cAdd VARCHAR(50),

cStartDate DATE,

cEndDate DATE,

hNum INT,

rNum INT,

CONSTRAINT UN\_cSSN UNIQUE(cSSN)

);

ALTER TABLE Hotel

ADD CONSTRAINT FK\_mID

FOREIGN KEY (hMID) REFERENCES Manager(mID);

ALTER TABLE Room

ADD CONSTRAINT FK\_roomHNum

FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

ALTER TABLE Customer

ADD CONSTRAINT FK\_hrNUM

FOREIGN KEY(hNum,rNum) REFERENCES Room(hNum, rNum);

ALTER TABLE Caretaker

ADD CONSTRAINT FK\_hNum

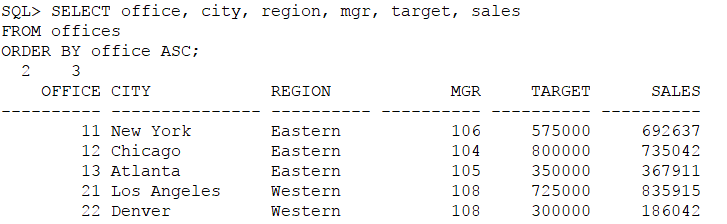
FOREIGN KEY(hNum) REFERENCES Hotel(hNum);

2)

1)

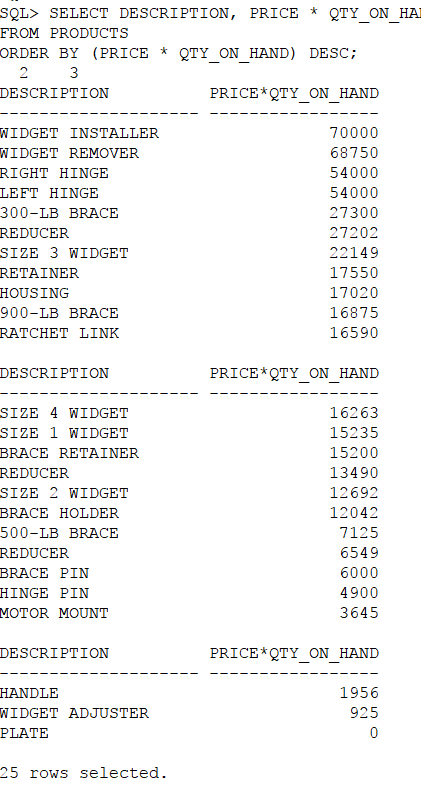
Write a select statement to list all the columns from the Offices table. 'Select \*' is not allowed. Return the list in Office order:

SELECT office, city, region, mgr, target, sales FROM offices ORDER BY office ASC;



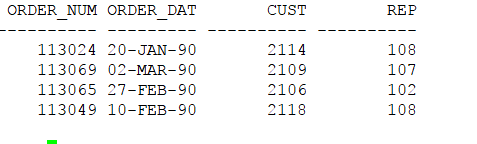
2) List the Product Name (its description), and dollar value of quantity on hand (price \* quantity) . Return the list in descending order by value.

SELECT DESCRIPTION, PRICE \* QTY\_ON\_HAND FROM PRODUCTS ORDER BY (PRICE \* QTY\_ON\_HAND) DESC;



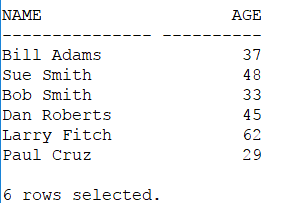
3) New: List the Order Number, Order Date, Customer Number and Sales Rep Number for orders for part 'XK47' or '775C'. (Use a compound search condition - OR.)

SELECT ORDER\_NUM, ORDER\_DATE, CUST, REP FROM ORDERS WHERE PRODUCT = 'XK47' OR PRODUCT = '775C';



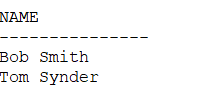
4) List the Name and Age for all Sales Reps in the following offices: 12; 21; 13. (Use the set membership test - IN.)

SELECT name, age FROM salesreps WHERE rep\_office IN('12', '13', '21');



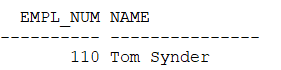
1. List the names of all Sales Reps who have the letter 'o' (this is lower case o) as the second character of their name.

SELECT name FROM salesreps WHERE name LIKE '\_o%';



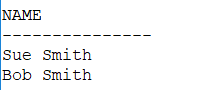
1. Return the Sales Rep ID and Name of any Sales Rep not assigned to an office yet.

SELECT empl\_num, name FROM salesreps WHERE rep\_office IS NULL;

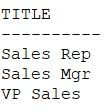


1. Show all the sales rep names with last name “Smith”.

SELECT name FROM Salesreps WHERE name LIKE '\_%Smith';

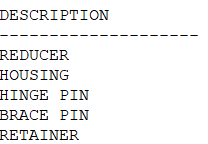


1. List different titles in the sales reps table. Only list each title once and unknown titles should be ignored. SELECT DISTINCT title FROM salesreps;



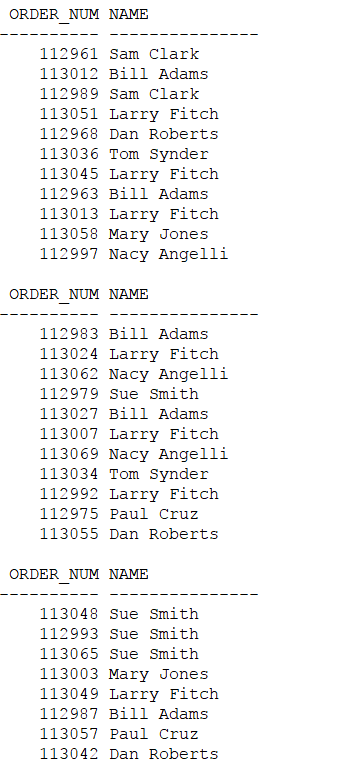
9)List the description of the products which are at least 6 character and less than 10 character long. No duplicate row is allowed. You can use the build in function length() to do this. For example, length(name) return the number of characters for attribute called “name”.

SELECT DISTINCT DESCRIPTION FROM PRODUCTS WHERE length(DESCRIPTION) > 6 AND length(DESCRIPTION) < 10;



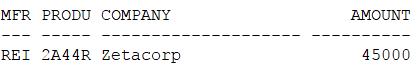
1. List the the order nums with the name of the rep who placed the order and the name of the customer who made the order and the name of the rep for that customer

SELECT ORDER\_NUM, name FROM ORDERS, salesreps WHERE REP = empl\_num;



3)

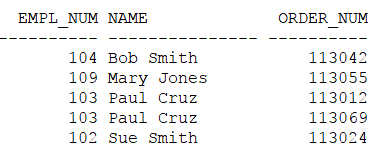
1. List the Mfr Id, the Product Id, Company and PRICE of all products brought by customers where customer number is one of (2112,2105,2119) and where the amount of the order < $5000.00. Order the results by ascending Company.



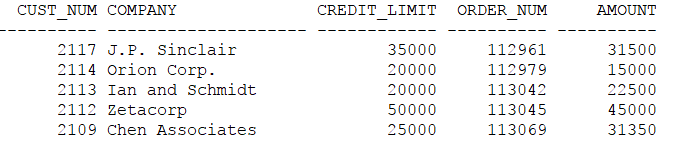
SELECT MFR, PRODUCT, COMPANY, AMOUNT FROM ORDERS, CUSTOMERS WHERE CUST = CUST\_NUM AND CUST IN('2112', '2105', '2119') AND AMOUNT > 5000 ORDER BY COMPANY ASC;

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 ORDERS, salesreps, Customers WHERE empl\_num = CUST\_REP AND REP!= empl\_num AND CUST = CUST\_NUM ORDER BY name ASC;

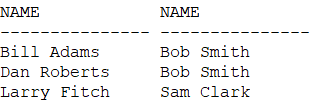


3) List all customers (Customer number, Company, and Credit Limit) and orders (Order Number, Amount) where the order is within $10000.00 of the Credit Limit (less than or equal to $10000). Sort the result by Customer number in descending order. SELECT CUST\_NUM, COMPANY, CREDIT\_LIMIT, ORDER\_NUM, AMOUNT 2 FROM ORDERS, CUSTOMERS 3 WHERE CUST = CUST\_NUM AND (CREDIT\_LIMIT - AMOUNT <= 10000) 4 ORDER BY CUST\_NUM DESC;

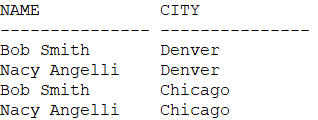


4) List all salesreps names and their managers’ names where the salesrep is at least as old as the manager.

SELECT Employees.name, Managers.name FROM salesreps Employees, salesreps Managers WHERE Employees.manager = Managers.empl\_num AND Employees.age >= Managers.age;



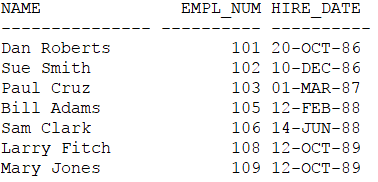
1. List all salesreps (Name) and the City they work in where the sales of the salesrep < Quota and the sales for the office is < Target. SELECT name, city FROM salesreps, offices WHERE salesreps.sales < quota AND offices.sales < target;



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

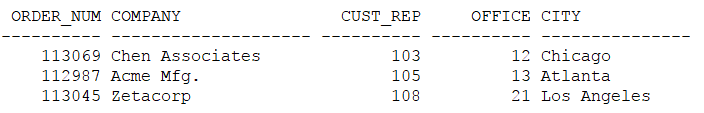
* The salesrep sales should be greater than the quota
* The salesrep has taken an order from Customer number 2117, 2111, or 2101.

Sort the result by the salesrep’s id.

SELECT DISTINCT name, empl\_num, hire\_date FROM salesreps, ORDERS WHERE sales > quota OR REP = empl\_num AND CUST IN('2117', '2111', '2101') ORDER BY empl\_num ASC;  


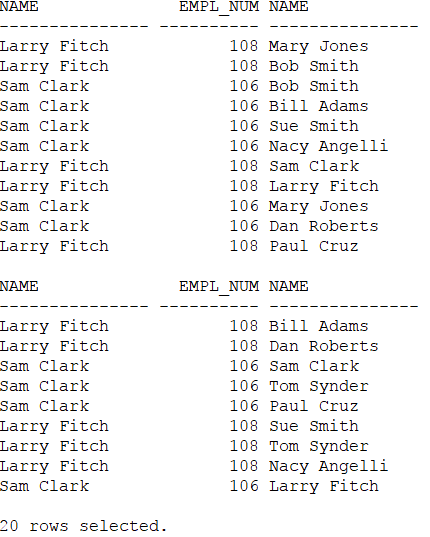
7) List all orders (Order Number) over $25000 showing the company placing the order, the Customer Rep assign to the company, the Office id and the city where the Customer Rep works in, such that the Customer Rep’s manager is not the person who actually took the order.

SELECT DISTINCT ORDER\_NUM, COMPANY, CUST\_REP, office, city FROM ORDERS, CUSTOMERS, salesreps, offices WHERE AMOUNT > 25000 AND office = rep\_office AND CUST\_REP = empl\_num and manager != REP AND CUST = CUST\_NUM;



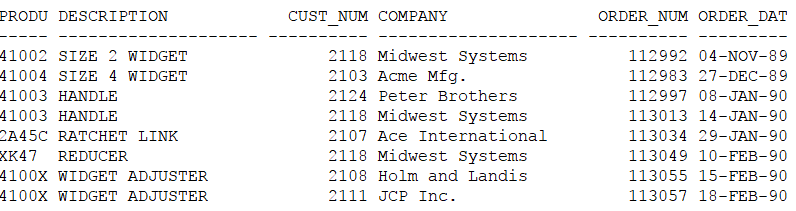
8) List all customer reps (their name and their id) and their managers name in which the manager has taken an order for the customer Rep’s company or the manager is based in New York or Denver. Use appropriate column header

SELECT DISTINCT Employees.name, Employees.empl\_num, Managers.name FROM salesreps Employees, salesreps Managers, ORDERS, offices, CUSTOMERS WHERE Employees.empl\_num = CUST\_REP AND REP = Employees.manager AND CUST\_NUM = CUST AND Employees.manager = Employees.empl\_num OR (mgr = Employees.empl\_num AND ( city = 'New York' OR city ='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 DISTINCT PRODUCT\_ID, DESCRIPTION, CUST\_NUM, COMPANY, ORDER\_NUM, ORDER\_DATE FROM PRODUCTS, ORDERS, CUSTOMERS WHERE CUST = CUST\_NUM AND PRODUCT = PRODUCT\_ID AND MFR = MFR\_ID AND AMOUNT < 1000 ORDER BY ORDER\_DATE ASC;



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 PRODUCT\_ID, DESCRIPTION, CUST\_NUM, COMPANY, ORDER\_NUM, ORDER\_DATE FROM PRODUCTS, ORDERS, CUSTOMERS WHERE CUST = CUST\_NUM AND PRODUCT = PRODUCT\_ID AND MFR = MFR\_ID AND AMOUNT < 1000 ORDER BY ORDER\_DATE ASC;

