

PROGRAM 2: FLIGHT DATABASE

2. The following relations keep track of airline flight information:

FLIGHTS(no: integer, *from1*: string, *to1*: string, *distance*: integer, *Departs*: time, *arrives*: time, *price*: real)

AIRCRAFT (aid: integer, *aname*: string, *cruisingrange*: integer)

CERTIFIED(*eid*: integer, aid: integer)

EMPLOYEES (eid: integer, *ename*: string, *salary*: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Design an ER model and schema diagram for the relations.

Write each of the following queries in SQL.

- Create the tables for the schemas provided with primary keys and foreign keys.
- Insert five tuples of values to each table.
- Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
- For each pilot who is certified for more than three aircrafts, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.
- Find the names of pilots whose *salary* is less than the price of the cheapest route from Bengaluru to Frankfurt.

SCHEMA DIAGRAM

FLIGHT

<u>no</u>	from1	to1	distance	departs	arrives	price
-----------	-------	-----	----------	---------	---------	-------

AIRCRAFT

<u>aid</u>	aname	cruisingrange
------------	-------	---------------

CERTIFIED

<u>eid</u>	<u>aid</u>
------------	------------

EMPLOYEES

<u>eid</u>	ename	salary
------------	-------	--------

a) Create the above tables for the schemas provided with primary keys and foreign keys.

```
CREATE TABLE FLIGHT
(NO INTEGER PRIMARY KEY,
FROM1 VARCHAR(20),
TO1 VARCHAR(20),
DISTANCE INTEGER,
DEPARTS DATE,
ARRIVES DATE,
PRICE REAL);
```

Table created.

```
CREATE TABLE AIRCRAFT
(AID INT PRIMARY KEY,
ANAME VARCHAR2(15),
CRUISINGRANGE INTEGER);
```

Table created.

```
CREATE TABLE EMPLOYEES
(EID INT PRIMARY KEY,
ENAME VARCHAR2(15),
SALARY REAL);
```

Table created.

```
CREATE TABLE CERTIFIED
(EID INT,
AID INT,
PRIMARY KEY(EID,AID),
FOREIGN KEY (EID) REFERENCES EMPLOYEES(EID),
FOREIGN KEY (AID) REFERENCES AIRCRAFT(AID));
```

Table created.

b) Insert five tuples of values to each table.

```
INSERT INTO FLIGHT VALUES (001,'BANGALORE','Frankfurt',2870,TO_DATE('20-12-2021','DD-MM-YYYY'),TO_DATE('20-12-2021','DD-MM-YYYY'),10000);
INSERT INTO FLIGHT VALUES (002,'CHENNAI','NOVA',4000,TO_DATE('27-07-2021','DD-MM-YYYY'),TO_DATE('27-07-2021','DD-MM-YYYY'),20000);
INSERT INTO FLIGHT VALUES (003,'BANGALORE','MIRAMER',5000,TO_DATE('09-07-2021','DD-MM-YYYY'),TO_DATE('09-07-2021','DD-MM-YYYY'),30000);
INSERT INTO FLIGHT VALUES (004,'BANGALORE','SHANHOK',60000,TO_DATE('19-08-2021','DD-MM-YYYY'),TO_DATE('20-08-2021','DD-MM-YYYY'),50000);
INSERT INTO FLIGHT VALUES (005,'BANGALORE','ERANGEL',2870,TO_DATE('20-12-2021','DD-MM-YYYY'),TO_DATE('20-12-2021','DD-MM-YYYY'),10000);
```

```
INSERT INTO AIRCRAFT VALUES (1,'INDIAN',10000);
INSERT INTO AIRCRAFT VALUES (2,'IndiGo',10000);
INSERT INTO AIRCRAFT VALUES (3,'GoAir',10000);
INSERT INTO AIRCRAFT VALUES (4,'SpiceJet',10000);
INSERT INTO AIRCRAFT VALUES (5,'TruJet',10000);
UPDATE AIRCRAFT SET CRUISINGRANGE ='90000' WHERE AID=5;
UPDATE AIRCRAFT SET CRUISINGRANGE ='80000' WHERE AID=2;
UPDATE AIRCRAFT SET CRUISINGRANGE ='70000' WHERE AID=3;
UPDATE AIRCRAFT SET CRUISINGRANGE ='60000' WHERE AID=4;
```

```
INSERT INTO EMPLOYEES VALUES (11,'KIRAN',1000000);
INSERT INTO EMPLOYEES VALUES (12,'REVANNA',2000000);
INSERT INTO EMPLOYEES VALUES (13,'RLK',3000000);
INSERT INTO EMPLOYEES VALUES (14,'AKASH',4000000);
INSERT INTO EMPLOYEES VALUES (15,'STARNET',5000000);
```

```
UPDATE EMPLOYEES SET SALARY='9000' WHERE SALARY='4000000';
INSERT INTO CERTIFIED VALUES (11, 1);
INSERT INTO CERTIFIED VALUES (11, 2);
INSERT INTO CERTIFIED VALUES (11, 3);
INSERT INTO CERTIFIED VALUES (11, 4);
INSERT INTO CERTIFIED VALUES (15, 5);
INSERT INTO CERTIFIED VALUES (14, 5);
INSERT INTO CERTIFIED VALUES (15, 4);
INSERT INTO CERTIFIED VALUES (15, 3);
INSERT INTO CERTIFIED VALUES (15, 2);
```

- c) Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.**

```
SELECT DISTINCT A.ANAME
FROM AIRCRAFT A
WHERE A.AID NOT IN(SELECT C.AID
FROM CERTIFIED C,EMPLOYEES E
WHERE C.EID=E.EID AND E.SALARY<80000);
```

- d) For each pilot who is certified for more than three aircrafts, find the *eid* and the maximum *cruisingrange* of the aircraft for which she or he is certified.**

```
SELECT C.EID,MAX(A.CRUISINGRANGE)
FROM CERTIFIED C, AIRCRAFT A
WHERE C.AID =A.AID
GROUP BY C.EID
HAVING COUNT(C.AID)>3;
```

- e) Find the names of pilots whose *salary* is less than the price of the cheapest route from Bengaluru to Frankfurt.**

```
SELECT ENAME  
FROM EMPLOYEES  
WHERE SALARY < (SELECT MIN(PRICE)  
FROM FLIGHT  
WHERE FROM1='BANGALORE' AND TO1='FRANKFURT');
```

PROGRAM 3: BOOK DEALER DATABASE

3. The following tables are maintained by a book dealer:

AUTHOR(author-id: int, name: String, city: String, country: String)

PUBLISHER(publisher-id: int, name: String, city: String, country: String)

CATALOG(book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

CATEGORY(category-id: int, description: String)

ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

Design an ER model and schema diagram for the relations.

Write each of the following queries in SQL.

- Create the tables for the schemas provided with primary keys and foreign keys.
- Insert five tuples of values to each table.
- Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.
- Find the author of the book which has maximum sales.
- Demonstrate how you increase the price of books published by a specific publisher by 10%.

SCHEMA DIAGRAM

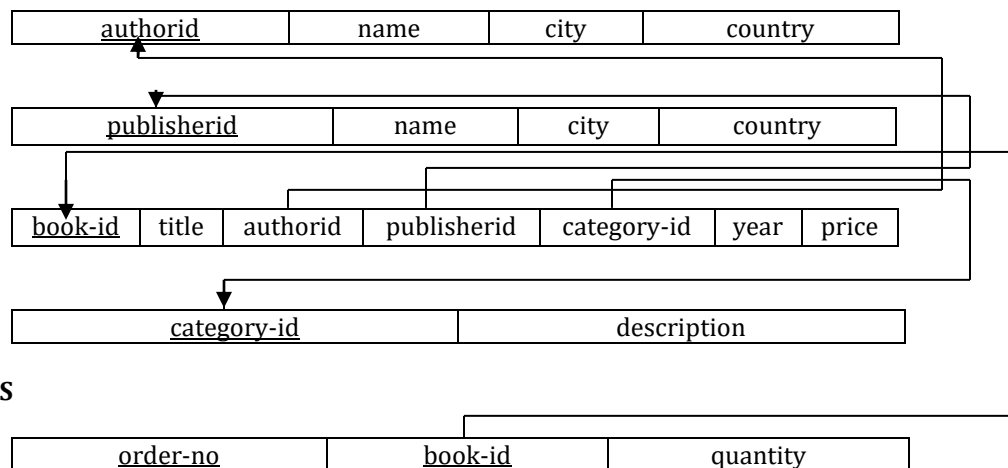
AUTHOR

PUBLISHER

CATALOG

CATEGORY

ORDER-DETAILS



a) Create the above tables for the schemas provided with primary keys and foreign keys.

```
CREATE TABLE AUTHOR
(AUTHORID INT,
NAME VARCHAR(20),
CITY VARCHAR(20),
COUNTRY VARCHAR(15), PRIMARY KEY(AUTHORID));
```

Table created.

```
CREATE TABLE PUBLISHER
(PUBLISHERID INT,
NAME VARCHAR(20),
CITY VARCHAR(20),
COUNTRY VARCHAR(15),
PRIMARY KEY(PUBLISHERID));
```

Table created.

```
CREATE TABLE CATEGORY
(CATEGORYID INT,
DESCRIPTION VARCHAR(20),
PRIMARY KEY(CATEGORYID));
```

Table created.

```
CREATE TABLE CATALOG
(BOOKID INT,
TITLE VARCHAR(20),
AUTHORID INT,
PUBLISHERID INT,
CATEGORYID INT,
YEAR INT,
PRICE INT,
PRIMARY KEY(BOOKID),
FOREIGN KEY(AUTHORID) REFERENCES AUTHOR(AUTHORID),
FOREIGN KEY(PUBLISHERID) REFERENCES PUBLISHER(PUBLISHERID),
FOREIGN KEY(CATEGORYID) REFERENCES CATEGORY(CATEGORYID));
```

Table created.

```
CREATE TABLE ORDER_DETAILS
(ORDERNO INT,
BOOKID INT,
QUANTITY INT,
PRIMARY KEY(ORDERNO,BOOKID),
FOREIGN KEY(BOOKID) REFERENCES CATALOG(BOOKID));
```

Table created.

b) Insert five tuples of values to each table.

```
INSERT INTO AUTHOR VALUES(1,'KIRAN','BNAGALORE','INDIA');
INSERT INTO AUTHOR VALUES(2,'AKASH','BNAGALORE','INDIA');
INSERT INTO AUTHOR VALUES(3,'REVANNA','BNAGALORE','INDIA');
INSERT INTO AUTHOR VALUES(4,'RLK','BNAGALORE','INDIA');
INSERT INTO AUTHOR VALUES(5,'SHAKTHI','BNAGALORE','INDIA');

INSERT INTO PUBLISHER VALUES(01,'Once Upon a Time','BANGALORE','INDIA');
INSERT INTO PUBLISHER VALUES(02,'RAINY WAY','BANGALORE','INDIA');
INSERT INTO PUBLISHER VALUES(03,'ATTITUDE','BANGALORE','INDIA');
INSERT INTO PUBLISHER VALUES(04,'THINK LIKE A MONK','BANGALORE','INDIA');
INSERT INTO PUBLISHER VALUES(05,'FAULT IN STARS','BANGALORE','INDIA');

INSERT INTO CATEGORY VALUES(001,'BREAVE WAORIOR');
INSERT INTO CATEGORY VALUES(002,'STORY');
INSERT INTO CATEGORY VALUES(031,'HORROR');
INSERT INTO CATEGORY VALUES(003,'GOD');
INSERT INTO CATEGORY VALUES(005,'GAMING');

INSERT INTO CATALOG VALUES(11,'BOOK',1,01,001,2000,700);
INSERT INTO CATALOG VALUES(12,'BOOK',1,02,002,2000,800);
INSERT INTO CATALOG VALUES(13,'BOOK',1,03,003,2000,900);
INSERT INTO CATALOG VALUES(14,'BOOK',3,04,031,2000,1000);
INSERT INTO CATALOG VALUES(15,'BOOK',3,05,005,2000,600);
INSERT INTO CATALOG VALUES(16,'SCIENCE',4,05,005,1998,600);

INSERT INTO ORDER_DETAILS VALUES(20,11,100);
INSERT INTO ORDER_DETAILS VALUES(21,15,150);
INSERT INTO ORDER_DETAILS VALUES(22,14,200);
INSERT INTO ORDER_DETAILS VALUES(23,13,90);
INSERT INTO ORDER_DETAILS VALUES(24,12,80);
```

c) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog should be greater than average price of the books and the year of publication is after 2000.

```
SELECT *
FROM AUTHOR
WHERE AUTHORID IN (SELECT AUTHORID
FROM CATALOG
GROUP BY AUTHORID
HAVING COUNT(*) >2 AND AUTHORID IN(SELECT AUTHORID
FROM CATALOG
WHERE (PRICE >
(SELECT AVG(PRICE) FROM CATALOG)
AND YEAR>=2000)));
```

d) Find the author of the book which has maximum sales.

```
SELECT name FROM author a,catalog c
WHERE a.authorid=c.authorid AND
bookid IN (SELECT bookid FROM order_details WHERE quantity=
(SELECT MAX(quantity) FROM order_details));
```

e) Demonstrate how you increase the price of books published by a specific publisher by 10%.

```
SELECT BOOKID, TITLE, PRICE, PRICE*1.1
FROM CATALOG;
```


PROGRAM 4 : BANKING ENTERPRISE DATABASE

4. Consider the following database for a banking enterprise.

BRANCH (branch-name: String, branch-city: String, assets: real)

ACCOUNTS (accno: int, branch-name: String, balance: real)

DEPOSITOR (customer-name: String, accno: int)

CUSTOMER(customer-name:string, customer-street:string, customer-city:string)

LOAN (loan-number: int, branch-name: String, amount: real)

BORROWER (customer-name: String, loan-number: int)

- Create the tables for the schemas provided with primary keys and foreign keys.
- Insert five tuples of values to each table.
- Find all the customers who have at least two accounts at the *Main* branch.
- Find all the customers who have an account at *all* the branches located in a specific city.
- Demonstrate how you delete all account tuples at every branch located in a specific city.

SCHEMA DIAGRAM

BRANCH

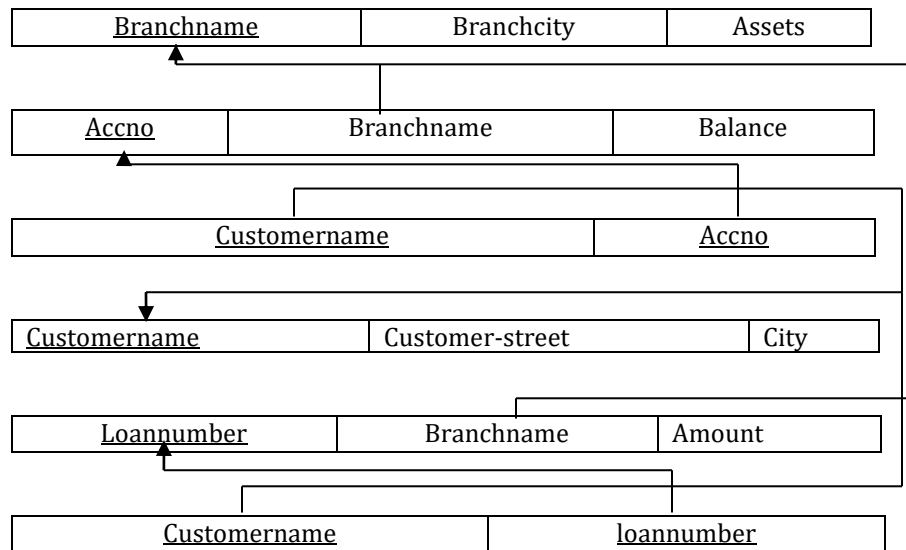
ACCOUNT

DEPOSITOR

CUSTOMER

LOAN

BORROWER



a) Create the above tables for the schemas provided with primary keys and foreign keys.

```
CREATE TABLE BRANCH
(BRANCH_NAME VARCHAR(20) PRIMARY KEY,
BRANCH_CITY VARCHAR(10),
ASSETS REAL);
```

Table created.

```
CREATE TABLE ACCOUNT
(ACCNO INT PRIMARY KEY,
BRANCH_NAME VARCHAR(20),
BALANCE REAL,
FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

Table created.

```
CREATE TABLE CUSTOMER
(CUSTOMER_NAME VARCHAR(20) PRIMARY KEY,
CUSTOMER_STREET VARCHAR(20),
CUST_CITY VARCHAR(20));
```

Table created.

```
CREATE TABLE DEPOSITOR
(CUSTOMER_NAME VARCHAR(20),
ACCNO INT,
PRIMARY KEY(CUSTOMER_NAME,ACCNO),
FOREIGN KEY(CUSTOMER_NAME) REFERENCES CUSTOMER(CUSTOMER_NAME),
FOREIGN KEY(ACCNO) REFERENCES ACCOUNT(ACCNO) ON DELETE CASCADE);
```

Table created.

```
CREATE TABLE LOAN
(LOAN_NO INT PRIMARY KEY,
BRANCH_NAME VARCHAR(20),
AMOUNT REAL,
FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

Table created.

```
CREATE TABLE BORROWER
(CUSTOMER_NAME VARCHAR(20),
LOAN_NO INT,
FOREIGN KEY(CUSTOMER_NAME) REFERENCES CUSTOMER(CUSTOMER_NAME),
FOREIGN KEY(LOAN_NO) REFERENCES LOAN(LOAN_NO));
```

Table created.

b) Insert five tuples of values to each table.

```
INSERT INTO BRANCH VALUES('NOVA','MANGALORE',10000);
INSERT INTO BRANCH VALUES('GEROGEPOL','ERANGEL',20000);
INSERT INTO BRANCH VALUES('FARM','BRIDGE',30000);
INSERT INTO BRANCH VALUES('LUMBERYARD','LIVIK',30000);
INSERT INTO BRANCH VALUES('CAVE','SHANHOK',20000);
INSERT INTO BRANCH VALUES('MAIN SBI VIJAY NAGAR','BANGALORE',20000);
INSERT INTO BRANCH VALUES('VIVEK NAGAR','BANGALORE',20000);
INSERT INTO BRANCH VALUES('KITHAGNUR','BANGALORE',20000);
INSERT INTO BRANCH VALUES('SUBHASH NAGAR','BANGALORE',20000);
```

```
INSERT INTO CUSTOMER VALUES ('KIRAN', 'REVANNA STREET', 'BANGALORE');
INSERT INTO ACCOUNT VALUES(019,'SUBHASH NAGAR',150000);
INSERT INTO ACCOUNT VALUES(018,'KITHAGNUR',150000);
INSERT INTO ACCOUNT VALUES(017,'VIVEK NAGAR',150000);
INSERT INTO ACCOUNT VALUES(016,'MAIN SBI VIJAY NAGAR',150000);
INSERT INTO ACCOUNT VALUES(015,'MAIN SBI VIJAY NAGAR',500000);
INSERT INTO ACCOUNT VALUES(014,'MAIN SBI VIJAY NAGAR',300000);
INSERT INTO ACCOUNT VALUES(013,'FARM',300000);
INSERT INTO ACCOUNT VALUES(012,'GEROGEPOL',200000);
INSERT INTO ACCOUNT VALUES(011,'NOVA',100000);
```

```
INSERT INTO CUSTOMER VALUES ('REVANNA', 'A STREET', 'BANGALORE');
INSERT INTO CUSTOMER VALUES ('KIRAN', 'REVANNA STREET', 'BANGALORE');
INSERT INTO CUSTOMER VALUES ('SHAKTHI', 'D STREET', 'BANGALORE');
INSERT INTO CUSTOMER VALUES ('AKASH', 'C STREET', 'BANGALORE');
INSERT INTO CUSTOMER VALUES ('R L K', 'B STREET', 'BANGALORE');
```

```
INSERT INTO DEPOSITOR VALUES('SHAKTHI',011);
INSERT INTO DEPOSITOR VALUES('AKASH',011);
INSERT INTO DEPOSITOR VALUES('R L K',011);
INSERT INTO DEPOSITOR VALUES('REVANNA',011);
INSERT INTO DEPOSITOR VALUES('KIRAN',011);
INSERT INTO DEPOSITOR VALUES('KIRAN',019);
INSERT INTO DEPOSITOR VALUES('AKASH',019);
INSERT INTO DEPOSITOR VALUES('AKASH',015);
INSERT INTO DEPOSITOR VALUES('AKASH',016);
INSERT INTO DEPOSITOR VALUES('AKASH',017);
INSERT INTO DEPOSITOR VALUES('AKASH',018);
```

```
INSERT INTO LOAN VALUES(7,'SUBHASH NAGAR', 10000);
INSERT INTO LOAN VALUES(6,'KITHAGNUR', 10000);
INSERT INTO LOAN VALUES(5,'VIVEK NAGAR', 10000);
INSERT INTO LOAN VALUES(4,'VIVEK NAGAR', 10000);
INSERT INTO LOAN VALUES(3,'MAIN SBI VIJAY NAGAR', 10000);
```

```
INSERT INTO LOAN VALUES(2,'MAIN SBI VIJAY NAGAR', 10000);
INSERT INTO LOAN VALUES(1,'MAIN SBI VIJAY NAGAR', 10000);
```

```
INSERT INTO BORROWER VALUES('KIRAN', 1);
INSERT INTO BORROWER VALUES('REVANNA', 2);
INSERT INTO BORROWER VALUES('R L K', 3);
INSERT INTO BORROWER VALUES('AKASH', 4);
INSERT INTO BORROWER VALUES('REVANNA', 5);
```

c) Find all the customers who have at least two accounts at the *Main* branch.

```
SELECT CUSTOMER_NAME
FROM DEPOSITOR
WHERE ACCNO IN(SELECT ACCNO
FROM DEPOSITOR
WHERE ACCNO IN( SELECT ACCNO
FROM ACCOUNT
WHERE BRANCH_NAME IN(SELECT BRANCH_NAME
FROM ACCOUNT
WHERE BRANCH_NAME='MAIN SBI VIJAY NAGAR'
GROUP BY BRANCH_NAME
HAVING COUNT(*) > 1) ))
GROUP BY CUSTOMER_NAME HAVING COUNT(*) > 1;
```

d) Find all the customers who have an account at *all* the branches located in a specific city.

```
SELECT CUSTOMER_NAME
FROM BRANCH B,ACCOUNT A,DEPOSITOR D
WHERE B.BRANCH_NAME=A.BRANCH_NAME AND A.ACCNO=D.ACCNO AND
B.BRANCH_CITY='BANGALORE' GROUP BY CUSTOMER_NAME
HAVING COUNT(DISTINCT B.BRANCH_NAME)=(SELECT COUNT(BRANCH_NAME)
FROM BRANCH
WHERE BRANCH_CITY='BANGALORE');
```

e) Demonstrate how you delete all account tuples at every branch located in a specific city.

```
DELETE FROM ACCOUNT
WHERE BRANCH_NAME IN (SELECT BRANCH_NAME
FROM BRANCH
WHERE BRANCH_CITY='BANGALORE');
```

PROGRAM 5 : ORDER PROCESSING DATABASE

5. Consider the following relations for an Order Processing database application in a company.

CUSTOMER (CUST #: int, cname: String, city: String)

ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)

ITEM (item #: int, unit-price: int)

ORDER-ITEM (order #: int, item #: int, qty: int)

WAREHOUSE (warehouse #: int, city: String)

SHIPMENT (order #: int, warehouse #: int, ship-date: date)

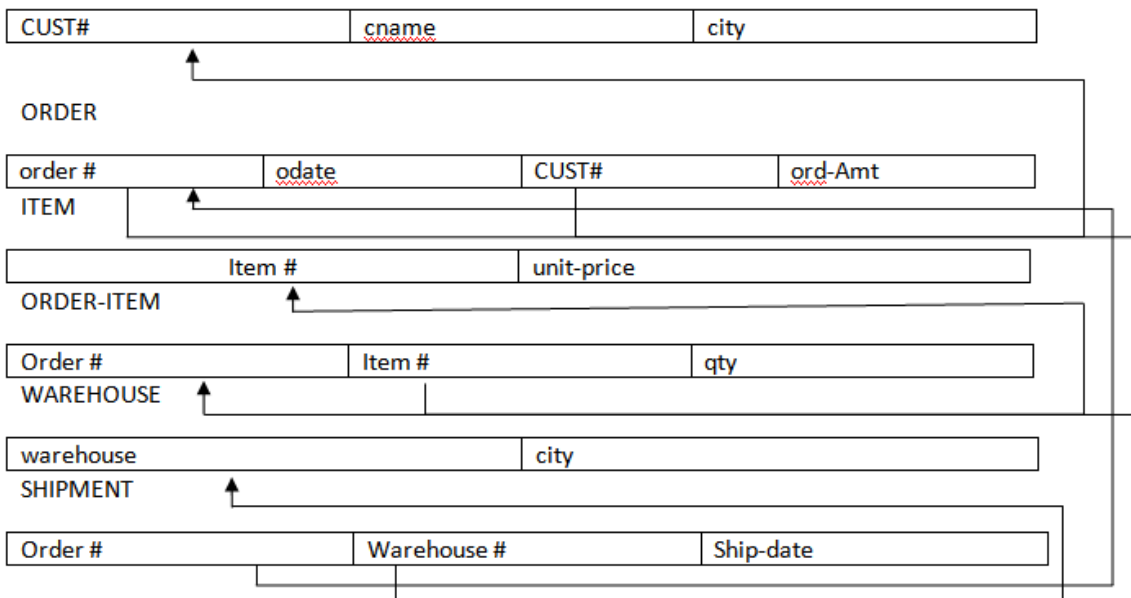
Design an ER model and Schema diagram for the relations

Write each of the following queries in SQL.

- Create the tables for the schemas provided with primary keys and foreign keys
- Insert five tuples of values to each table.
- Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.
- List the order# for orders that were shipped from all warehouses that the company has in a specific city.
- Demonstrate how you delete item# 10 from the ITEM table and make that field *null* in the ORDER_ITEM table.

SCHEMA DIAGRAM

CUSTOMER



a) Create the above tables for the schemas provided with primary keys and foreign keys.

```
CREATE TABLE CUST
(CID INT,
 CNAME VARCHAR2(10),
 CITY VARCHAR2(10),
 PRIMARY KEY (CID));
```

Table created.

```
CREATE TABLE ORDERS
(ONO INT,
 ODATE DATE,
 CID INT,
 ORD_AMT INT,
 PRIMARY KEY (ONO),
 FOREIGN KEY(CID) REFERENCES CUST(CID));
```

Table created.

```
CREATE TABLE ITEM(INO INT,
 UNIT_PRICE INT,
 PRIMARY KEY (INO));
```

Table created.

```
CREATE TABLE ORDER_ITEM (ONO INT,
 INO INT,
 QTY INT,
 PRIMARY KEY (ONO,INO),
 FOREIGN KEY (ONO) REFERENCES ORDERS(ONO),
 FOREIGN KEY (INO) REFERENCES ITEM (INO) ON DELETE CASCADE);
```

Table created.

```
CREATE TABLE WAREHOUSE
(WARENO INT,
 CITY VARCHAR2(10),
 PRIMARY KEY (WARENO));
```

Table created.

```
CREATE TABLE SHIPMENT
(ONO INT,
 WARENO INT,
 SHIPDATE DATE,
 PRIMARY KEY (ONO,WARENO),
```

```
FOREIGN KEY (ONO) REFERENCES ORDERS(ONO),  
FOREIGN KEY (WARENO) REFERENCES WAREHOUSE (WARENO));
```

Table created.

b) Insert five tuples of values to each table.

```
INSERT INTO CUST VALUES(1,'KIRAN','BANGALORE');  
INSERT INTO CUST VALUES(2,'REVANNA','BANGALORE');  
INSERT INTO CUST VALUES(3,'R L K','BANGALORE');  
INSERT INTO CUST VALUES(4,'AKASH','BANGALORE');  
INSERT INTO CUST VALUES(5,'SHAKTHI','BANGALORE');  
INSERT INTO CUST VALUES(6,'Ram','Mysore');
```

```
INSERT INTO ORDERS VALUES (01,TO_DATE('20-12-2021','DD-MM-YYYY'),1,1000);  
INSERT INTO ORDERS VALUES (02,TO_DATE('21-12-2021','DD-MM-YYYY'),2,2000);  
INSERT INTO ORDERS VALUES (03,TO_DATE('22-12-2021','DD-MM-YYYY'),3,3000);  
INSERT INTO ORDERS VALUES (04,TO_DATE('23-12-2021','DD-MM-YYYY'),4,4000);  
INSERT INTO ORDERS VALUES (05,TO_DATE('24-12-2021','DD-MM-YYYY'),5,5000);  
INSERT INTO ORDERS VALUES (06,TO_DATE('25-12-2021','DD-MM-YYYY'),6,6000);
```

```
INSERT INTO ITEM VALUES(5,500);  
INSERT INTO ITEM VALUES(6,1000);  
INSERT INTO ITEM VALUES(7,700);  
INSERT INTO ITEM VALUES(8,800);  
INSERT INTO ITEM VALUES(9,900);  
INSERT INTO ITEM VALUES(10,1000);
```

```
INSERT INTO ORDER_ITEM VALUES(01,5,1);  
INSERT INTO ORDER_ITEM VALUES(02,6,2);  
INSERT INTO ORDER_ITEM VALUES(03,7,3);  
INSERT INTO ORDER_ITEM VALUES(04,8,4);  
INSERT INTO ORDER_ITEM VALUES(05,9,5);
```

```
INSERT INTO WAREHOUSE VALUES(11,'BANGALORE');  
INSERT INTO WAREHOUSE VALUES(12,'BANGALORE');  
INSERT INTO WAREHOUSE VALUES(13,'BANGALORE');  
INSERT INTO WAREHOUSE VALUES(14,'BANGALORE');  
INSERT INTO WAREHOUSE VALUES(15,'BANGALORE');  
INSERT INTO WAREHOUSE VALUES(16,'Mysore');
```

```
INSERT INTO SHIPMENT VALUES(01,11,TO_DATE('20-1-2021','DD-MM-YYYY'));  
INSERT INTO SHIPMENT VALUES(02,12,TO_DATE('21-1-2021','DD-MM-YYYY'));  
INSERT INTO SHIPMENT VALUES(03,13,TO_DATE('22-1-2021','DD-MM-YYYY'));  
INSERT INTO SHIPMENT VALUES(04,14,TO_DATE('23-1-2021','DD-MM-YYYY'));
```

```
INSERT INTO SHIPMENT VALUES(05,15,TO_DATE('24-1-2021','DD-MM-YYYY'));
INSERT INTO SHIPMENT VALUES(06,16,TO_DATE('25-1-2021','DD-MM-YYYY'));
```

- c) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.**

```
SELECT C.CID, C.CNAME, COUNT(*) AS NO_OF_ORDERS, AVG(OI.QTY*I.UNIT_PRICE)
FROM CUST C, ORDERS OT, ITEM I, ORDER_ITEM OI
WHERE C.CID=OT.CID AND OT.ONO=OI.ONO AND OI.INO=I.INO
GROUP BY (C.CID, C.CNAME);
```

- d) List the order# for orders that were shipped from all warehouses that the company has in a specific city.**

```
SELECT S.ONO
FROM SHIPMENT S, WAREHOUSE W
WHERE S.WARENO=W.WARENO AND W.CITY='BANGALORE';
```

- e) Demonstrate how you delete item# 10 from the ITEM table and make that field *null* in the ORDER_ITEM table.**

```
DELETE FROM ITEM
WHERE INO=10;

UPDATE ORDER_ITEM
SET INO=NULL
WHERE INO=10;
```