EXERCSE-1   INSURANCE DATABASE

 Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

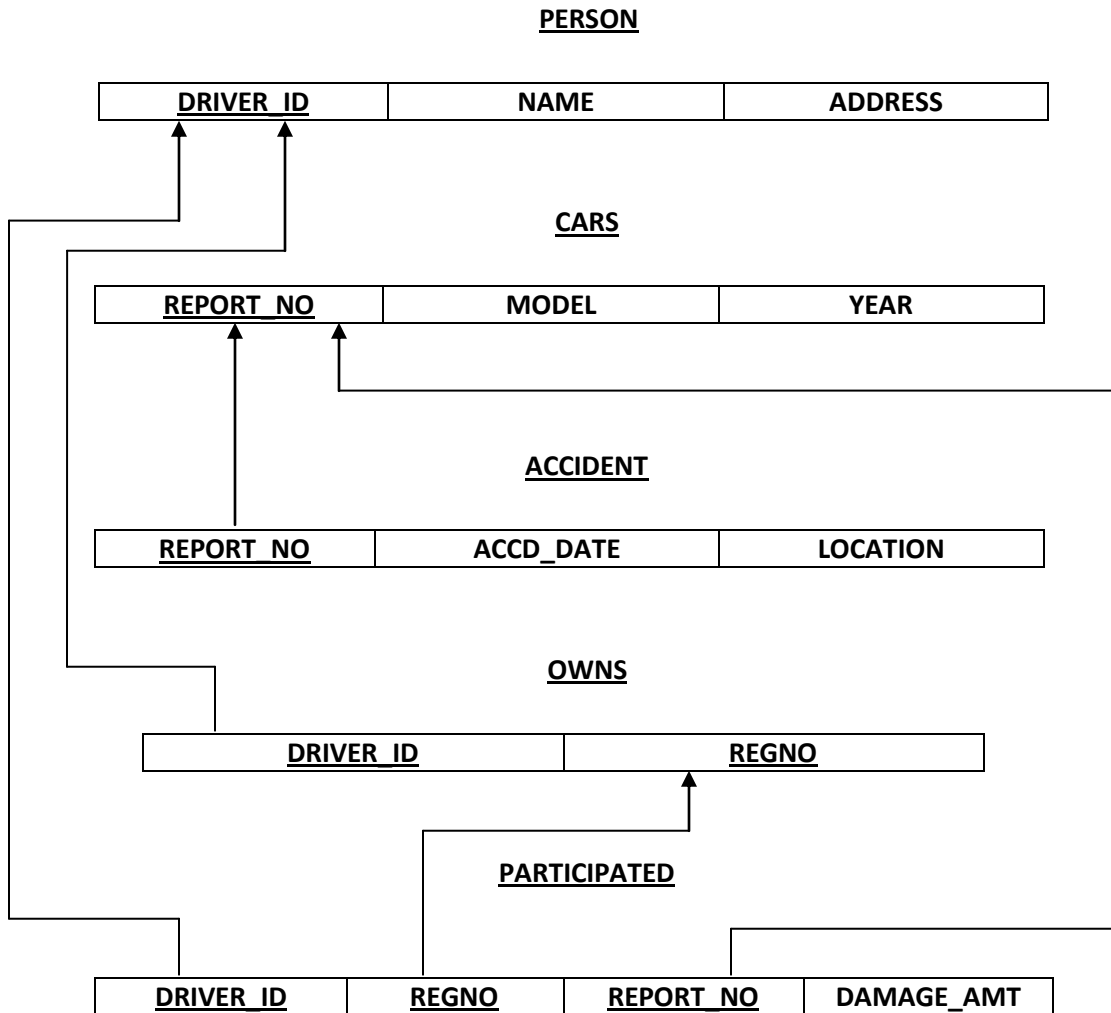PERSON (driver – id #: String, name: string, address: strong)

CAR (Regno: string, model: string, year: int)

ACCIDENT (report-number: int, accd-date: date, location: string)

OWNS (driver-id #:string, Regno:string)

PARTICIPATED (driver-id: string, Regno:string, report-number:int, damage amount:int)

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.
(ii)   Enter at least five tuples for each relation.
(iii)   Demonstrate how you
    a.   Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.
    b.   Add a new accident to the database.
(iv) Find the total number of people who owned cars that were involved in accidents in 2008.
(v)   Find the number of accidents in which cars belonging to a specific model were involved.
(vi) Generate suitable reports.

SCHEMA DIAGRAM:-

PERSON

DRIVER_ID	NAME	ADDRESS

CARS

REPORT_NO	MODEL	YEAR

ACCIDENT

REPORT_NO	ACCD_DATE	LOCATION

OWNS

DRIVER_ID	REGNO

PARTICIPATED

DRIVER_ID	REGNO	REPORT_NO	DAMAGE_AMT

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.

SQL> create table person(driver_id varchar2(10) primary key,name char(15),address varchar2(25));
Table created.

SQL> create table car(regno varchar2(10) primary key,model varchar2(15),year number(4));
Table created.

SQL> create table accident(report_no number(8) primary key,accd_date date,location char(15));
Table created.

SQL> create table owns(driver_id varchar2(10) references person(driver_id),regno varchar2(10)
references car(regno));
Table created.

SQL>create table participated(driver_id varchar2(10) references person(driver_id),regno varchar2(10)
references car(regno),report_no number(8)references accident(report_no),damage_amt number(10));
Table created.

(ii)   Enter at least five tuples for each relation.

SQL> insert into person values('a101','anil','bangalore');
1 row created.

SQL> insert into person values('a102','amit','mangalore');
1 row created.

SQL> insert into person values('a103','rohit','orissa');
1 row created.

SQL> insert into person values('a104','rakesh','lucknow');
1 row created.

SQL> insert into person values('a015','maddy','punjab');
1 row created.

SQL> SELECT * FROM PERSON;

DRIVER_ID	NAME	ADDRESS
a101	anil	bangalore
a102	amit	mangalore
a103	rohit	orissa
a104	rakesh	lucknow
a015	maddy	Punjab

SQL> insert into car values('ka03hh7014','sxi-dlx',2009);
1 row created.

SQL> insert into car values('ka02eh5590','i20',2010);
1 row created.

SQL> insert into car values('ka03k1234','fiat',1990);
1 row created.

SQL> insert into car values('mh12p6969','ferrari',2008);
1 row created.

SQL> insert into car values('rj02m0755','gypsy',2009);
1 row created.

SQL> SELECT * FROM CAR;

REGNO	MODEL	YEAR
ka03hh7014	sxi-dlx	2009
ka02eh5590	i20	2010
ka03k1234	fiat	1990
mh12p6969	ferrari	2008
rj02m0755	gypsy	2009

SQL> insert into accident values(501,'05-feb-2008','ring road');
1 row created.

SQL> insert into accident values(502,'21-mar-2009','peenya');
1 row created.

SQL> insert into accident values(503,'10-aug-2008','tumkur');
1 row created.

SQL> insert into accident values(12,'29-sep-2009','mysore');
1 row created.

SQL> insert into accident values(20,'01-jun-2010','hebbal');
1 row created.

SQL> select * from accident;

REPORT_NO	ACCD_DATE	LOCATION
501	05-FEB-08	ring road
502	21-MAR-09	peenya
503	10-AUG-08	tumkur
12	29-SEP-09	mysore
20	01-JUN-10	hebbal

SQL> insert into owns values('a101','ka03hh7014');
1 row created.

SQL> insert into owns values('a102','mh12p6969');
1 row created.

SQL> insert into owns values('a103','rj02m0755');
1 row created.

SQL> insert into owns values('a101','ka03k1234');
1 row created.

SQL> insert into owns values('a102','ka02eh5590');
1 row created.

SQL> select * from owns;

DRIVER_ID	REGNO
a101	ka03hh7014
a102	mh12p6969
a103	rj02m0755
a101	ka03k1234
a102	ka02eh5590

SQL> insert into participated values('a101','ka03k1234',12,7000);
1 row created.

SQL> insert into participated values('a102','mh12p6969',501,12000);
1 row created.

SQL> insert into participated values('a103','rj02m0755',20,8500);
1 row created.

SQL> insert into participated values('a104','ka02eh5590',503,4000);
1 row created.

SQL> insert into participated values('a101','ka03hh7014',501,20000);
1 row created.

SQL> select * from participated;

DRIVER_ID	REGNO	REPORT_NO	DAMAGE_AMT
a101	ka03k1234	12	7000
a102	mh12p6969	501	12000
a103	rj02m0755	20	8500
a104	ka02eh5590	503	4000
a101	ka03hh7014	501	20000

(iii)   Demonstrate how you
   a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.

SQL> update participated set damage_amt=25000 where (regno='ka03k1234' and report_no=12);
1 row updated.

SQL> select * from participated;

DRIVER_ID	REGNO	REPORT_NO	DAMAGE_AMT
a101	ka03k1234	12	25000
a102	mh12p6969	501	12000
a103	rj02m0755	20	8500
a104	ka02eh5590	503	4000
a101	ka03hh7014	501	20000

(iii)   Demonstrate how you
b.   Add a new accident to the database.

SQL> insert into accident values(504,'18-oct-2008','krpuram');
1 row created.
SQL> select * from accident;

REPORT_NO	ACCD_DATE	LOCATION
501	05-FEB-08	ring road
502	21-MAR-09	peenya
503	10-AUG-08	tumkur
12	29-SEP-09	mysore
20	01-JUN-10	hebbal
504	18-OCT-08	krpuram

6 rows selected.

(iv) Find the total number of people who owned cars that were involved in accidents in 2008.

SQL> select count(pr.driver_id) from car c,person pr,accident a,participated p where(c.regno=p.regno
 and pr.driver_id=p.driver_id and a.report_no=p.report_no and year like 2008);

COUNT(PR.DRIVER_ID)

           1

(v)   Find the number of accidents in which cars belonging to a specific model were involved.

SQL> select count(a.report_no) from car c,accident a,participated p where(c.regno=p.regno and
a.report_no=p.report_no and model like 'gypsy');

COUNT(A.REPORT_NO)

           1

(vi) Generate suitable reports.

SQL> select pr.driver_id,a.report_no,a.location,p.damage_amt,p.regno from person pr,accident a,
Parti cipated p where (p.driver_id=pr.driver_id and a.report_no=p.report_no);

DRIVER_ID	REPORT_NO	LOCATION	DAMAGE_AMT	REGNO
a101	12	mysore	25000	ka03k1234
a102	501	ring road	12000	mh12p6969
a103	20	hebbal	8500	rj02m0755
a104	503	tumkur	4000	ka02eh5590
a101	501	ring road	20000	ka03hh7014

EXERCISE-2 ORDER PROCESSING DATABASE

Consider the following relations for an order processing database application in a company.

CUSTOMER (cust #: int , cname: string, city: string)

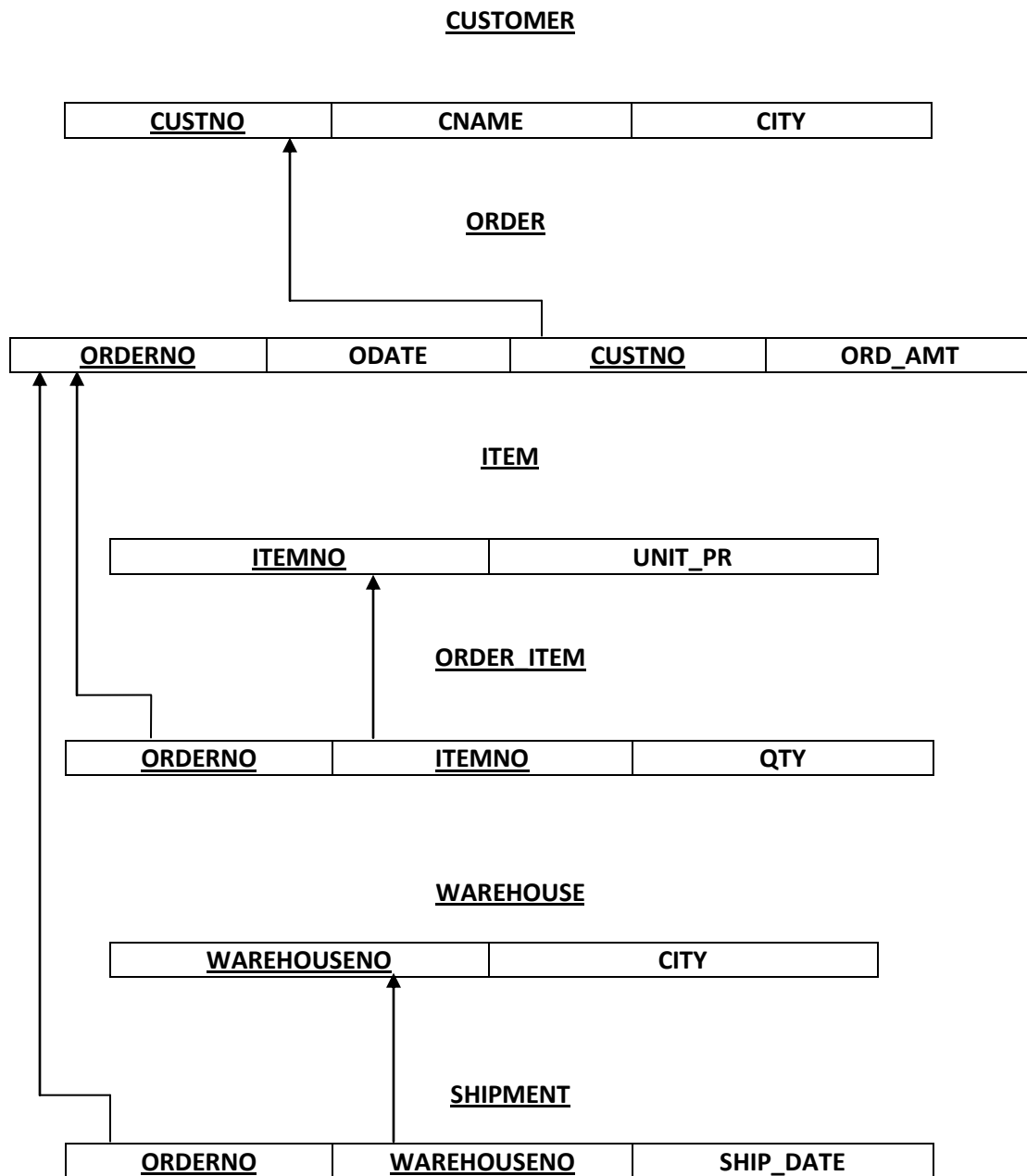**ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)**

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)

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.
(ii)   Enter at least five tuples for each relation.
(iii)   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.
(iv)   List the order# for orders that were shipped from *all* the warehouses that the company has in a specific city.
(v)   Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contain this particular item.
(vi)   Generate suitable reports.

SCHEMA DIAGRAM:-

CUSTOMER

CUSTNO	CNAME	CITY

ORDER

ORDERNO	ODATE	CUSTNO	ORD_AMT

ITEM

ITEMNO	UNIT_PR

ORDER_ITEM

ORDERNO	ITEMNO	QTY

WAREHOUSE

WAREHOUSENO	CITY

SHIPMENT

ORDERNO	WAREHOUSENO	SHIP_DATE

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.

SQL> create table customer(cust_no number(5) primary key,cname varchar2(10),city varchar2(10));
Table created.

SQL> create table ordr(order_no number(5) primary key,odate date,cust_no number(5) references
customer(cust_no),ord_amt number(6));
Table created.

SQL> create table item(item_no number(4) primary key,unit_pr number(5));
Table created.

SQL> create table order_item(order_no number(5) references ordr(order_no),item_no number(3)
references item(item_no),qty number(4));
Table created.

SQL> create table warehouse(warehouse_no number(5) primary key,city varchar2(10));
Table created.

SQL> create table shipment(order_no number(5) references ordr(order_no),warehouse_no number(5)
references warehouse(warehouse_no),ship_date date);
Table created.

(ii)   Enter at least five tuples for each relation.

SQL> insert into customer values(1001,'anil','orissa');
1 row created.

SQL> insert into customer values(1002,'ankit','bangalore');
1 row created.

SQL> insert into customer values(1003,'abhinav','muzaffar');
1 row created.

SQL> insert into customer values(1004,'amit','kolkata');
1 row created.

SQL> insert into customer values(1005,'manish','lucknow');
1 row created.

SQL> select * from customer;

CUST_NO	CNAME	CITY
1001	anil	orissa
1002	ankit	bangalore
1003	abhinav	muzaffar
1004	amit	kolkata
1005	manish	lucknow

SQL> insert into ordr values(501,'11-feb-2010',1002,5500);
1 row created.

SQL> insert into ordr values(502,'20-oct-2009',1001,9700);
1 row created.

SQL> insert into ordr values(505,'13-aug-2008',1005,5600);
1 row created.

SQL> insert into ordr values(568,'29-jun-2010',1004,4766);
1 row created.

SQL> insert into ordr values(609,'31-aug-2010',1003,9831);
1 row created.

SQL> select * from ordr;

ORDER_NO	ODATE	CUST_NO	ORD_AMT
501	11-FEB-10	1002	5500
502	20-OCT-09	1001	9700
505	13-AUG-08	1005	5600
568	29-JUN-10	1004	4766
609	31-AUG-10	1003	9831

SQL> insert into item values(111,255);
1 row created.

SQL> insert into item values(121,199);
1 row created.

SQL> insert into item values(131,177);
1 row created.

SQL> insert into item values(141,211);
1 row created.

SQL> insert into item values(151,311);
1 row created.

SQL> select * from item;

ITEM_NO	UNIT_PR
111	255
121	199
131	177
141	211
151	311

SQL> insert into order_item values(501,111,200);
1 row created.

SQL> insert into order_item values(502,121,100);
1 row created.

SQL> insert into order_item values(505,131,150);
1 row created.

SQL> insert into order_item values(568,121,300);
1 row created.

SQL> insert into order_item values(568,111,500);
1 row created.

SQL> select * from order_item;

ORDER_NO	ITEM_NO	QTY
501	111	200
502	121	100
505	131	150
568	121	300
568	111	500

SQL> insert into warehouse values(1500,'bangalore');
1 row created.

SQL> insert into warehouse values(1550,'orissa');
1 row created.

SQL> insert into warehouse values(1600,'paradeep');
1 row created.

SQL> insert into warehouse values(1650,'vishak');
1 row created.

SQL> insert into warehouse values(1700,'bangalore');
1 row created.

SQL> select * from warehouse;

WAREHOUSE_NO	CITY
1500	bangalore
1550	orissa
1600	paradeep
1650	vishak
1700	bangalore

SQL> insert into shipment values(501,1500,'23-sep-2010');
1 row created.

SQL> insert into shipment values(502,1600,'30-oct-2009');
1 row created.

SQL> insert into shipment values(568,1500,'28-feb-2010');
1 row created.

SQL> insert into shipment values(609,1550,'19-aug-2010');
1 row created.

SQL> insert into shipment values(502,1500,'23-jul-2010');
1 row created.

SQL> select * from shipment;

ORDER_NO	WAREHOUSE_NO	SHIP_DATE
501	1500	23-SEP-10
502	1600	30-OCT-09
568	1500	28-FEB-10
609	1550	19-AUG-10
502	1500	23-JUL-10

(iii)   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.

SQL>select c.cname, count(order_no), avg(ord_amt) from customer c, ordr o where c.cust_no=o.cust_no group by cname;

CNAME	COUNT(ORDER_NO)	AVG(ORD_AMT)
abhinav	1	9831
amit	1	4766
anil	1	9700
ankit	1	5500
manish	1	5600

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

SQL> select s.order_no,w.city from shipment s,warehouse w where w.warehouse_no=s.warehouse_no and city like 'bangalore';

ORDER_NO	CITY
501	bangalore
568	bangalore
502	bangalore

(v) Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER_ITEM table that contain this particular item.

SQL> update order_item set item_no=NULL where item_no=111;
2 rows updated.

SQL> delete from item where item_no=111;
1 row deleted.

(vi)   Generate suitable reports.

SQL> select cname,c.cust_no,o.order_no from customer c,ordr o where c.cust_no=o.cust_no;

CNAME	CUST_NO	ORDER_NO
ankit	1002	501
anil	1001	502
manish	1005	505
amit	1004	568
abhinav	1003	609

EXERCISE 3 – STUDENT ENROLLMENT IN COURSES AND BOOK ADOPTION DATABASE

Consider the following database of student enrollment in courses & books adopted for each course.

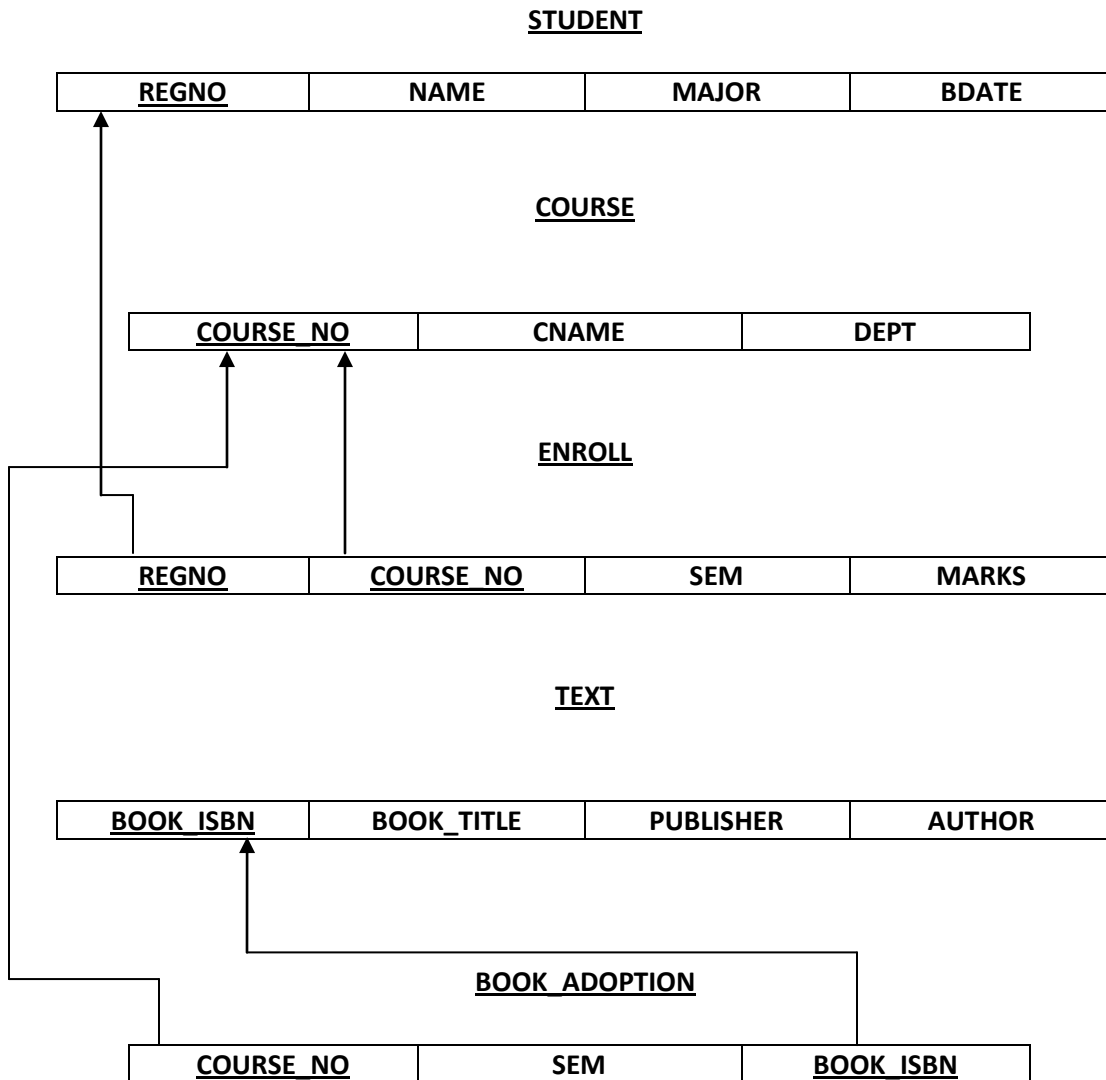**STUDENT (regno: string, name: string, major: string, bdate:date)**

COURSE (course #:int, cname:string, dept:string)

ENROLL (regno:string, course#:int, sem:int, marks:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)

 (i)   Create the above tables by properly specifying the primary keys and the foreign keys.
 (ii)   Enter at least five tuples for each relation.
 (iii)   Demonstrate how you add a new text book to the database and make this book be adopted by some department.
 (iv)   Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
 (v)   List any department that has *all* its adopted books published by a specific publisher.
 (vi)   Generate suitable reports.

SCHEMA DIAGRAM:

STUDENT

REGNO	NAME	MAJOR	BDATE

COURSE

COURSE_NO	CNAME	DEPT

ENROLL

REGNO	COURSE_NO	SEM	MARKS

TEXT

BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR

BOOK_ADOPTION

COURSE_NO	SEM	BOOK_ISBN

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.

SQL> create table student(regno varchar2(5) primary key,name varchar2(10),major varchar2(5),bdate date);
Table created.

SQL> create table course(course_no number(5) primary key,cname varchar2(10),dept varchar2(10));
Table created.

SQL> create table enroll(regno varchar2(5) references student(regno),course_no number(5) references course(course_no),sem number(3),marks number(4));
Table created.

SQL> create table text(book_isbn number(4) primary key,book_title varchar2(10),publisher varchar2(10),author varchar2(10));
Table created.

SQL> create table book_adoption(course_no number(5) references course(course_no),sem number(3),book_isbn number(4) references text(book_isbn));
Table created.

(ii)   Enter at least five tuples for each relation.

SQL> insert into student values('a101','anil','bcom','09-nov-1987');
1 row created.

SQL> insert into student values('a102','vineet','bsc','16-aug-1988');
1 row created.

SQL> insert into student values('a103','sourav','mba','20-may-1988');
1 row created.

SQL> insert into student values('a104','mandeep','msc','28-aug-1989');
1 row created.

SQL> insert into student values('a015','vicky','btech','17-aug-1989');
1 row created.

SQL> select * from student;

REGNO	NAME	MAJOR	BDATE
a101	anil	bcom	09-NOV-87
a102	vineet	bsc	16-AUG-88
a103	sourav	mba	20-MAY-88
a104	mandeep	msc	28-AUG-89
a015	vicky	btech	17-AUG-89

SQL> insert into course values(1001,'mca','dca');
1 row created.

SQL> insert into course values(1002,'bsc','cs');
1 row created.

SQL> insert into course values(1003,'bbm','dom');
1 row created.

SQL> insert into course values(1004,'msc','cs');
1 row created.

SQL> insert into course values(1005,'bca','dca');
1 row created.

SQL> select * from course;

COURSE_NO	CNAME	DEPT
1001	mca	dca
1002	bsc	cs
1003	bbm	dom
1004	msc	cs
1005	bca	dca

SQL> insert into enroll values('a101',1001,2,678);
1 row created.

SQL> insert into enroll values('a102',1005,3,775);
1 row created.

SQL> insert into enroll values('a103',1003,4,883);
1 row created.

SQL> insert into enroll values('a104',1002,1,798);
1 row created.

SQL> insert into enroll values('a101',1003,5,990);
1 row created.

SQL> select * from enroll;

REGNO	COURSE_NO	SEM	MARKS
a101	1001	2	678
a102	1005	3	775
a103	1003	4	883
a104	1002	1	798
a101	1003	5	990

SQL> insert into text values(2100,'networks','pearson','stallings');
1 row created.

SQL> insert into text values(3123,'dbms','mcgraw','navathe');
1 row created.

SQL> insert into text values(4567,'oops','tata','venugopal');
1 row created.

SQL> insert into text values(6412,'sys soft','reilly','dhamdere');
1 row created.

SQL> insert into text values(8455,'tech com','peterson','mathew');
1 row created.

SQL> select * from text;

BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
2100	networks	pearson	stallings
3123	dbms	mcgraw	navathe
4567	oops	tata	venugopal
6412	sys soft	reilly	dhamdere
8455	tech com	peterson	mathew

SQL> insert into book_adoption values(1002,2,2100);
1 row created.

SQL> insert into book_adoption values(1004,3,3123);
1 row created.

SQL> insert into book_adoption values(1004,2,2100);
1 row created.

SQL> insert into book_adoption values(1005,4,8455);
1 row created.

SQL> insert into book_adoption values(1003,3,4567);
1 row created.
SQL> select * from book_adoption;

COURSE_NO	SEM	BOOK_ISBN
1002	2	2100
1004	3	3123
1004	2	2100
1005	4	8455
1003	3	4567

(iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.

SQL> insert into text values(5398,'java','sonic','ranchor');
1 row created.

SQL> insert into book_adoption values(1004,4,5398);
1 row created.

(iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

SQL> select c.course_no,cname,t.book_isbn,book_title from course c,text t,book_adoption b where c.course_no=b.course_no and b.book_isbn=t.book_isbn and dept='cs' and c.course_no in(select course_no from book_adoption group by course_no having count(*) >= 1) order by c.cname;

COURSE_NO	CNAME	BOOK_ISBN	BOOK_TITLE
1002	bsc	2100	networks
1004	msc	3123	dbms
1004	msc	2100	networks
1004	msc	5398	java

(v) List any department that has *all* its adopted books published by a specific publisher.

SQL> select c.dept from course c where course_no in(select course_no from book_adoption where book_isbn=(select book_isbn from text where publisher='mcgraw'));

DEPT
----
dca

(vi) Generate suitable reports.

SQL> select c.course_no,c.cname,c.dept,t.book_isbn,t.book_title from course c,book_adoption b,text t where c.course_no=b.course_no and t.book_isbn=t.book_isbn;

COURSE_NO	CNAME	DEPT	BOOK_ISBN	BOOK_TITLE
1004	msc	cs	5398	java
1003	bbm	dom	5398	java
1002	bsc	cs	5398	java

EXERCISE-4 : BOOK DEALER DATABASE

The following tables are maintained by a book dealer.

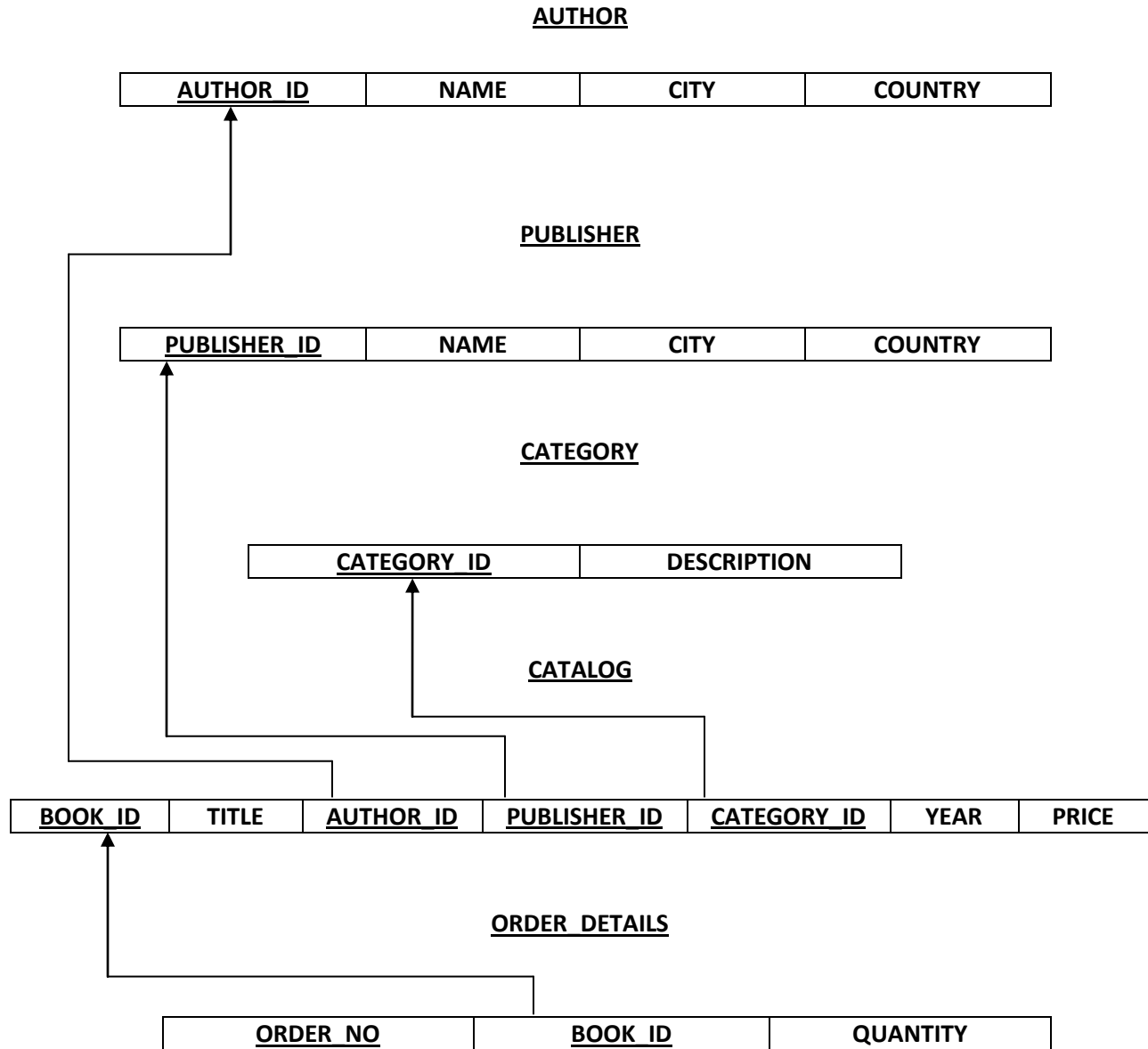   **AUTHOR (author-id:int, name:string, city:string, country:string)**

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

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

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

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

(i)   **Create the above tables by properly specifying the primary keys and the foreign keys.**
(ii)   **Enter at least five tuples for each relation.**
(iii)   **Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.**
(iv)   **Find the author of the book which has maximum sales.**
(v)   **Demonstrate how you increase the price of books published by a specific publisher by 10%.**
(vi)   **Generate suitable reports.**

SCHEMA DIAGRAM

AUTHOR

AUTHOR_ID	NAME	CITY	COUNTRY

PUBLISHER

PUBLISHER_ID	NAME	CITY	COUNTRY

CATEGORY

CATEGORY_ID	DESCRIPTION

CATALOG

BOOK_ID	TITLE	AUTHOR_ID	PUBLISHER_ID	CATEGORY_ID	YEAR	PRICE

ORDER_DETAILS

ORDER_NO	BOOK_ID	QUANTITY

(i)   Create the above tables by properly specifying the primary keys and the foreign keys.

SQL> create table author(author_id number(3) primary key,name varchar2(15),city varchar2(10),country varchar2(10));
Table created.

SQL> create table publisher(publisher_id number(3) primary key,name varchar2(10),city varchar2(10),
country varchar2(10));
Table created.

SQL> create table category(category_id number(4) primary key,descrip varchar2(10));
Table created.

SQL> create table catalog(book_id number(4) primary key,title varchar2(10),author_id number(3) references author(author_id),publisher_id number(3) references publisher(publisher_id),category_id number(4) references category(category_id),year number(4),price number(5));
Table created.

SQL> create table order_det(order_no number(4) primary key,book_id number(4) references catalog(book_id),qty number(4));
Table created**.**

(ii)   Enter at least five tuples for each relation.

SQL> insert into author values(101,'anil','pilani','india');
1 row created.

SQL> insert into author values(102,'amit','kolkata','india');
1 row created.

SQL> insert into author values(103,'william','shinsui','hongkong');
1 row created.

SQL> insert into author values(104,'akhil','haridwar','utarakand');
1 row created.

SQL> insert into author values(105,'surya','vishak','india');
1 row created.

SQL> select * from author;

AUTHOR_ID	NAME	CITY	COUNTRY
101	anil	pilani	india
102	amit	kolkata	india
103	william	shinsui	hongkong
104	akhil	haridwar	utarakand
105	surya	vishak	india

SQL> insert into publisher values(550,'pearson','arab','afgan');
1 row created.

SQL> insert into publisher values(660,'peterson','patiala','india');
1 row created.

SQL> insert into publisher values(770,'ranchor','laskar','india');
1 row created.

SQL> insert into publisher values(880,'chanchad','dehradun','india');
1 row created.

SQL> insert into publisher values(990,'munna','mumbai','india');
1 row created.

SQL> select * from publisher;

PUBLISHER_ID	NAME	CITY	COUNTRY
550	pearson	arab	afgan
660	peterson	patiala	india
770	ranchor	laskar	india
880	chanchad	dehradun	india
990	munna	mumbai	india

SQL> insert into category values(011,'oops');
1 row created.

SQL> insert into category values(012,'networks');
1 row created.

SQL> insert into category values(013,'dbms');
1 row created.

SQL> insert into category values(014,'sys soft');
1 row created.

SQL> insert into category values(015,'java');
1 row created.

SQL> select * from category;

CATEGORY_ID	DESCRIP
11	oops
12	networks
13	dbms
14	sys soft
15	java

SQL> insert into catalog values(900,'com nw',101,660,13,2009,667);
1 row created.

SQL> insert into catalog values(901,'oracle',102,770,14,2008,555);
1 row created.

SQL> insert into catalog values(905,'security',101,660,11,2009,189);
1 row created.

SQL> insert into catalog values(910,'costing',103,550,14,2010,335);
1 row created.

SQL> insert into catalog values(917,'accounting',105,770,12,2006,412);
1 row created.

SQL> select * from catalog;

BOOK_ID	TITLE	AUTHOR_ID	PUBLISHER_ID	CATEGORY_ID	YEAR	PRICE
900	com nw	101	660	13	2009	667
901	oracle	102	770	14	2008	555
905	security	101	660	11	2009	189
910	costing	103	550	14	2010	335
917	accounting	105	770	12	2006	412

SQL> insert into order_det values(001,900,200);
1 row created.

SQL> insert into order_det values(002,905,150);
1 row created.

SQL> insert into order_det values(003,917,170);
1 row created.

SQL> insert into order_det values(004,910,132);
1 row created.

SQL> insert into order_det values(005,905,120);
1 row created.

SQL> select * from order_det;

ORDER_NO	BOOK_ID	QTY
1	900	200
2	905	150
3	917	170
4	910	132
5	905	120

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

SQL> select * from author where author_id in (select author_id from catalog where (year>2000) and (catalog.price>(select avg(price)from catalog))group by author_id having count(*)>2);

AUTHOR_ID	NAME	CITY	COUNTRY
101	anil	pilani	india
102	amit	kolkata	india

(iv)   Find the author of the book which has maximum sales.

SQL> select a.name from author a,catalog c,order_det o where (a.author_id=c.author_id) and (c.book_id=o.book_id) and (o.qty=(select max(qty) from order_det));

NAME

----

anil

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

SQL> update catalog set price=price*1.1 where publisher_id=770;

2 rows updated.

(vi)   Generate suitable reports.

SQL> select a.author_id,a.name,p.publisher_id,p.name,c.book_id,title from author a,publisher p, catalog c,order_det o where a.author_id=c.author_id and p.publisher_id=c.publisher_id and c.book_id=o.book_id;

AUTHOR_ID	NAME	PUBLISHER_ID	NAME	BOOK_ID	TITLE
101	anil	660	peterson	900	com nw
101	anil	660	peterson	905	security
105	surya	770	ranchor	917	accounting
102	amit	550	pearson	910	costing
101	anil	660	peterson	905	security

EXERCISE-5: BANKING DATABASE

Consider the following database for a banking enterprise

BRANCH(<u>branch-name</u>:string, branch-city:string, assets:real)

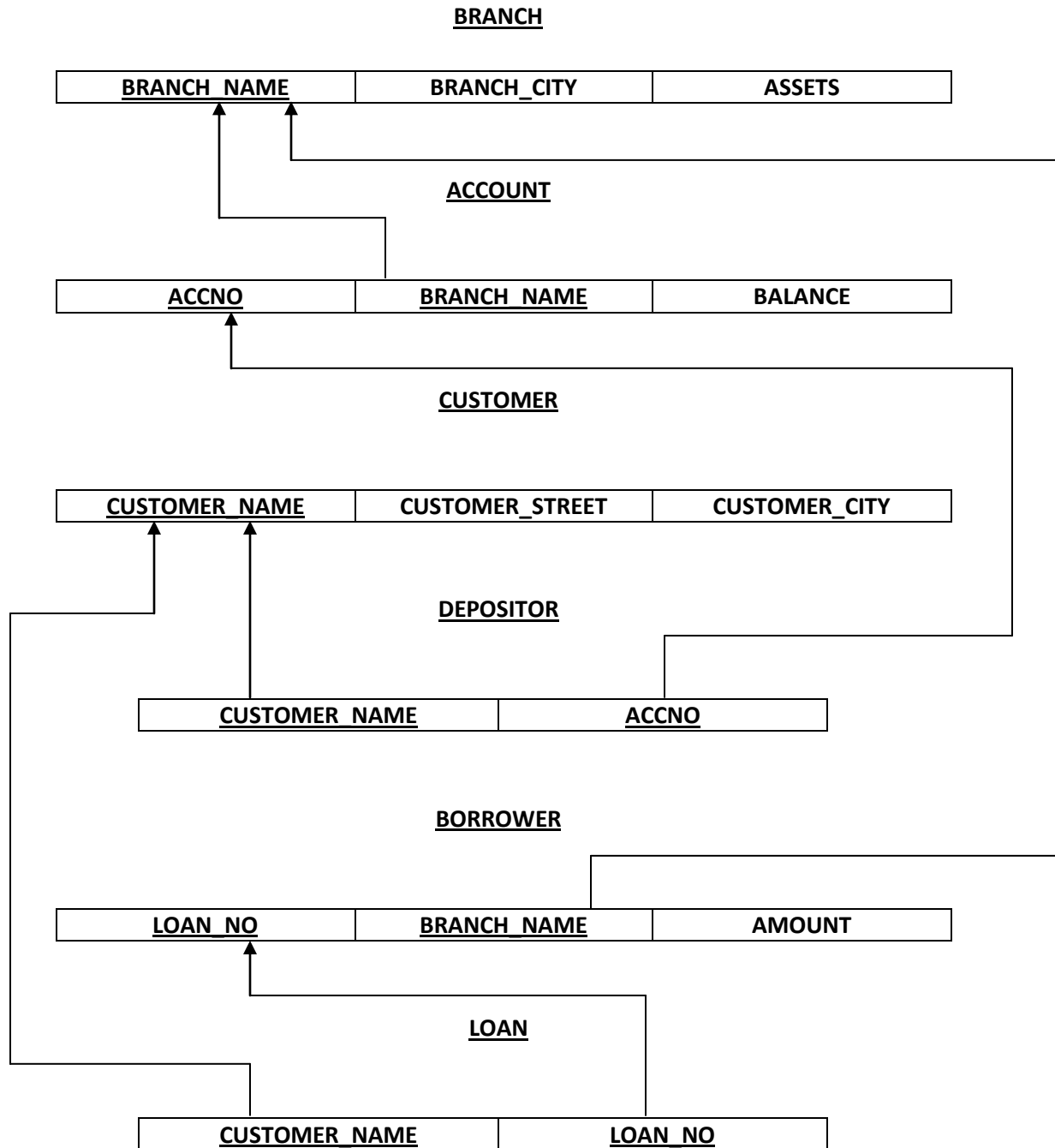ACCOUNT(<u>accno</u>:int, branch-name:string, balance:real)

CUSTOMER(<u>customer-name</u>:string, customer-street:string, customer-city:string)

DEPOSITOR(<u>customer-name</u>:string, <u>accno</u>:int)

LOAN(<u>loan-number</u>:int, branch-name:string, amount:real)

BORROWER(<u>customer-name</u>:string, <u>loan-number</u>:int)

(i)   Create the above tables by properly specifying the primary keys and the foreign keys
(ii) Enter at least five tuples for each relation
(iii) Find all the customers who have at least two accounts at the Main branch.
(iv) Find all the customers who have an account at all the branches located in a specific city.
(v)   Demonstrate how you delete all account tuples at every branch located in a specific city.
(vi) Generate suitable reports.

SCHEMA DIAGRAM

BRANCH

BRANCH_NAME	BRANCH_CITY	ASSETS

ACCOUNT

ACCNO	BRANCH_NAME	BALANCE

CUSTOMER

CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY

DEPOSITOR

CUSTOMER_NAME	ACCNO

BORROWER

LOAN_NO	BRANCH_NAME	AMOUNT

LOAN

CUSTOMER_NAME	LOAN_NO

(i)   Create the above tables by properly specifying the primary keys and the foreign keys

SQL> create table branch(br_name varchar2(10) primary key,br_city varchar2(10),assets number(6,2));
Table created.

SQL> create table account(accno number(5) primary key,br_name varchar2(10) references branch(br_name),balance number(8,2));
Table created.

SQL> create table cust_det(cust_name varchar2(10) primary key,cust_strt varchar2(10),cust_city varchar2(10));
Table created.

SQL> create table depositor(cust_name varchar2(10) references cust_det(cust_name),accno number(5) references account(accno));
Table created.

SQL> create table loan(loan_no number(6) primary key,br_name varchar2(10) references branch(br_name),amount number(6,2));
Table created.

SQL> create table borrower(cust_name varchar2(10) references cust_det(cust_name), loan_no number(6) references loan(loan_no));
Table created.

(ii)   Enter at least five tuples for each relation

SQL> insert into branch values('peenya','bangalore',8765.20);
1 row created.

SQL> insert into branch values('sahid nagr','bhubneswar',9876.50);
1 row created.

SQL> insert into branch values('col road','keonjhar',5643.60);
1 row created.

SQL> insert into branch values('old town','keonjhar',6890.99);
1 row created.

SQL> insert into branch values('nhce','bangalore',1234.56);
1 row created.

SQL> select * from branch;

BR_NAME	BR_CITY	ASSETS
peenya	bangalore	8765.2
sahid nagr	bhubneswar	9876.5
col road	keonjhar	5643.6
old town	keonjhar	6890.99
nhce	bangalore	1234.56

SQL> insert into account values(19901,'peenya',7500.55);
1 row created.

SQL> insert into account values(19956,'nhce',63999);
1 row created.

SQL> insert into account values(23856,'old town',8567.32);
1 row created.

SQL> insert into account values(78231,'col road',2000);
1 row created.

SQL> insert into account values(34518,'peenya',6500);
1 row created.

SQL> select * from account;

ACCNO	BR_NAME	BALANCE
19901	peenya	7500.55
19956	nhce	63999
23856	old town	8567.32
78231	col road	2000
34518	peenya	6500

SQL> insert into cust_det values('anil','park strt','bangalore');
1 row created.

SQL> insert into cust_det values('amit','palika','delhi');
1 row created.

SQL> insert into cust_det values('ankit','kundanhali','bangalore');
1 row created.

SQL> insert into cust_det values('manish','vignan','bangalore');
1 row created.

SQL> insert into cust_det values('abhinav','monikola','bangalore');
1 row created.

SQL> select * from cust_det;

CUST_NAME	CUST_STRT	CUST_CITY
anil	park strt	bangalore
amit	palika	delhi
ankit	kundanhali	bangalore
manish	vignan	bangalore
abhinav	monikola	Bangalore

SQL> insert into depositor values('anil',19901);
1 row created.

SQL> insert into depositor values('amit',78231);
1 row created.

SQL> insert into depositor values('manish',19956);
1 row created.

SQL> insert into depositor values('abhinav',23856);
1 row created.

SQL> insert into depositor values('anil',19956);
1 row created.

SQL> select * from depositor;

CUST_NAME	ACCNO
anil	19901
amit	78231
manish	19956
abhinav	23856
anil	19956

SQL> insert into loan values(36969,'peenya',2000.00);
1 row created.

SQL> insert into loan values(21001,'nhce',5986.67);
1 row created.

SQL> insert into loan values(51234,'old town',3477);
1 row created.

SQL> insert into loan values(78911,'col road',9950);
1 row created.

SQL> insert into loan values(56091,'col road',7600);
1 row created.

SQL> select * from loan;

LOAN_NO	BR_NAME	AMOUNT
36969	peenya	2000
21001	nhce	5986.67
51234	old town	3477
78911	col road	9950
56091	col road	7600

SQL> insert into borrower values('anil',36969);
1 row created.

SQL> insert into borrower values('amit',56091);
1 row created.

SQL> insert into borrower values('anil',21001);
1 row created.

SQL> insert into borrower values('ankit',78911);
1 row created.

SQL> insert into borrower values('amit',51234);
1 row created.

SQL> select * from borrower;

CUST_NAME	LOAN_NO
anil	36969
amit	56091
anil	21001
ankit	78911
amit	51234

(iii) Find all the customers who have at least two accounts at the Main branch.

SQL>   select   c.cust_name   from   cust_det   c,depositor   d,account   a,branch   b
where(d.cust_name=c.cust_name) and (a.accno=d.accno) and (b.br_name=a.br_name) and (b.br_name
like 'peenya') group by c.cust_name having count(d.accno)>=2;

CUST_NAME
----
anil

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

SQL> select distinct d.cust_name from depositor d where accno in (select a.accno from account a where br_name in(select br_name from branch where br_city='bangalore'));

CUST_NAME

----

anil
manish

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

SQL> delete from depositor where accno in(select accno from account a,branch b where a.br_name=b.br_name and br_city='bangalore');

4 rows deleted.

SQL> delete from account where br_name in(select br_name from branch where br_city='bangalore');

3 rows deleted.

(vi) Generate suitable reports.

SQL>   select   c.cust_name,a.accno,c.cust_strt   from   cust_det   c,account   a,depositor   d where(a.accno=d.accno);

CUST_NAME	ACCNO	CUST_STRT
anil	23856	park strt
amit	23856	palika
ankit	23856	kundanhali
manish	23856	vignan
abhinav	23856	monikola