TERMWORK 2

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

# 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.**

CREATE TABLE PERSON(

ID VARCHAR(10) DEFAULT '0',

NAME VARCHAR(10) NOT NULL,

ADDRESS VARCHAR(32) NOT NULL,

PRIMARY KEY(ID)

);

CREATE TABLE CAR(

REGNO VARCHAR(10) DEFAULT '0',

MODEL VARCHAR(10) NOT NULL,

YEAROFMAKING INT NOT NULL,

PRIMARY KEY(REGNO)

);

CREATE TABLE ACCIDENT(

REPORT\_NUMBER INT DEFAULT 0,

ACCD\_DATE DATE NOT NULL,

LOCATION VARCHAR(32) NOT NULL,

PRIMARY KEY(REPORT\_NUMBER)

);

CREATE TABLE OWNS(

OWNS\_ID VARCHAR(10) DEFAULT '0',

CAR\_REGNO VARCHAR(10) DEFAULT '0',

FOREIGN KEY(OWNS\_ID) REFERENCES PERSON(ID),

FOREIGN KEY(CAR\_REGNO) REFERENCES CAR(REGNO)

);

CREATE TABLE PARTICIPATED(

PARTICIPATED\_ID VARCHAR(10) DEFAULT '0',

NAME VARCHAR(20) NOT NULL,

PARTICIPATED\_REGNO VARCHAR(10) DEFAULT '0',

PARTICIPATEDREPORT\_NUMBER INT DEFAULT 0,

DAMAGE INT NOT NULL,

FOREIGN KEY(PARTICIPATED\_ID) REFERENCES PERSON(ID),

FOREIGN KEY(PARTICIPATED\_REGNO) REFERENCES CAR(REGNO),

FOREIGN KEY(PARTICIPATEDREPORT\_NUMBER) REFERENCES ACCIDENT(REPORT\_NUMBER)

);

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

INSERT INTO PERSON VALUES('1','AMIT','AGRA');

INSERT INTO PERSON VALUES('2','ARJUN','HUBLI');

INSERT INTO PERSON VALUES('3','SAHIL','DHARWAD');

INSERT INTO PERSON VALUES('4','GOURISH','DELHI');

INSERT INTO PERSON VALUES('5','ADITIYA','RANCHI');

INSERT INTO CAR VALUES('12345','Nissan',2018);

INSERT INTO CAR VALUES('12346','Mahindra',2017);

INSERT INTO CAR VALUES('12347','Maruti',2015);

INSERT INTO CAR VALUES('12348','Hyundai',2014);

INSERT INTO CAR VALUES('12349','Toyota',2011);

INSERT INTO CAR VALUES('12350','BMW',2017);

INSERT INTO ACCIDENT VALUES(1,'12-FEB-2015','HUBLI');

INSERT INTO ACCIDENT VALUES(2,'14-DEC-2017','DHARWAD');

INSERT INTO ACCIDENT VALUES(3,'23-MAR-2018','AGRA');

INSERT INTO ACCIDENT VALUES(4,'21-OCT-2017','DELHI');

INSERT INTO ACCIDENT VALUES(5,'24-JUN-2015','RANCHI');

INSERT INTO ACCIDENT VALUES(12,'24-DEC-2017','DELHI');

INSERT INTO OWNS VALUES('1','12345');

INSERT INTO OWNS VALUES('2','12349');

INSERT INTO OWNS VALUES('3','12346');

INSERT INTO OWNS VALUES('4','12347');

INSERT INTO OWNS VALUES('5','12348');

INSERT INTO PARTICIPATED VALUES('1','12345',3,1230);

INSERT INTO PARTICIPATED VALUES('2','12349',1,1100);

INSERT INTO PARTICIPATED VALUES('3','12346',2,1290);

INSERT INTO PARTICIPATED VALUES('4','12347',4,7000);

INSERT INTO PARTICIPATED VALUES('5','12348',5,23000);

INSERT INTO PARTICIPATED VALUES('1','12348',4,12345);

INSERT INTO PARTICIPATED VALUES('3','12348',12,23090);

**(iii) Demonstrate how you**

**a. Update the damage amount to 25000 for the car with a**

**specific Regno in the ACCIDENT**

**table with report number 12.**

UPDATE PARTICIPATED SET DAMAGE=25000 WHERE PARTICIPATEDREPORT\_NUMBER=12;

**b. Add a new accident to the database.**

INSERT INTO ACCIDENT VALUES(13,'24-DEC-2017','HUBLI');

**(iv) Find the total number of people who owned cars that**

**were involved in accidents in 2018.(DONE)**

SELECT COUNT(\*) FROM ACCIDENT WHERE ACCD\_DATE LIKE '%18';

**(v) Find the number of accidents in which cars belonging to**

**a specific model were involved.(DONE)**

SELECT COUNT(\*) FROM PARTICIPATED WHERE PARTICIPATED\_REGNO IN

(SELECT REGNO FROM CAR WHERE MODEL='Hyundai');

**vi) Find number of cars involved in each accident.(DONE)**

SELECT COUNT(\*),PARTICIPATEDREPORT\_NUMBER FROM PARTICIPATED GROUP BY(PARTICIPATEDREPORT\_NUMBER);

**vii) Find total damage amount of each accident.(DONE)**

SELECT SUM(DAMAGE),PARTICIPATEDREPORT\_NUMBER FROM PARTICIPATED GROUP BY(PARTICIPATEDREPORT\_NUMBER);

**viii) List the report no ,which has max damage amount.(DONE)**

SELECT PARTICIPATEDREPORT\_NUMBER FROM PARTICIPATED WHERE

DAMAGE >= ALL(SELECT DAMAGE FROM PARTICIPATED);

**ix) list the car regno, which has maximum damage amount in a specific accident.(done)**

SELECT PARTICIPATED\_REGNO FROM PARTICIPATED WHERE PARTICIPATEDREPORT\_NUMBER='4' AND DAMAGE >= ALL(SELECT DAMAGE FROM PARTICIPATED WHERE PARTICIPATEDREPORT\_NUMBER='4');

**x) List the number of times each time every driver**

**has met with an accident.(DONE)**

SELECT COUNT(PARTICIPATED\_ID),PARTICIPATED\_ID FROM PARTICIPATED GROUP BY(PARTICIPATED\_ID);

**xi) List the cars that have not met with accident.(DONE)**

SELECT \* FROM CAR WHERE REGNO <> ALL (SELECT PARTICIPATED\_REGNO FROM PARTICIPATED);

**xii) List the number of times the same combination driver\_id and**

**car reg\_no has met with accidents.(DONE)**

SELECT COUNT(\*),OWNS\_ID,CAR\_REGNO FROM OWNS GROUP BY OWNS\_ID,CAR\_REGNO;

**xiii) List day-wise number of accidents in the year 2015.(DONE)**

SELECT COUNT(\*),ACCD\_DATE FROM ACCIDENT WHERE ACCD\_DATE LIKE '%15' GROUP BY(ACCD\_DATE);

**xiv) List the number of times the car has met with**

**accidents along with total damage amount and average damage amount.(DONE)**

SELECT COUNT(\*),SUM(DAMAGE),AVG(DAMAGE),PARTICIPATED\_REGNO FROM PARTICIPATED GROUP BY PARTICIPATED\_REGNO;

**xv) List the driver name and address who have met with**

**accident who are not the owners of the**

**car.(DONE)**

SELECT NAME,ADDRESS FROM PERSON WHERE ID IN (SELECT PARTICIPATED\_ID FROM PARTICIPATED,OWNS WHERE PARTICIPATED\_ID=OWNS\_ID AND PARTICIPATED\_REGNO <>CAR\_REGNO);

TERMWORK3

# 3. 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)

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

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

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

# WAREHOUSE (warehouse #: int, city: string)

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

CREATE TABLE CUSTOMER(

CUST# INT,

CNAME VARCHAR(10) NOT NULL,

CITY VARCHAR(10),

PRIMARY KEY(CUST#)

);

CREATE TABLE ORDERS(

ORDERS# INT,

ODATE DATE NOT NULL,

CUST# INT,

ORD\_AMOUNT INT,

PRIMARY KEY(ORDERS#),

FOREIGN KEY(CUST#) REFERENCES CUSTOMER(CUST#) ON DELETE SET NULL

);

CREATE TABLE ITEM(

ITEM# INT ,

UNIT\_PRICE INT DEFAULT 0,

PRIMARY KEY(ITEM#)

);

CREATE TABLE ORDER\_ITEM(

ORDERS# INT,

ITEM# INT ,

QTY INT DEFAULT 0,

FOREIGN KEY(ORDERS#) REFERENCES ORDERS(ORDERS#) ON DELETE SET NULL,

FOREIGN KEY(ITEM#) REFERENCES ITEM(ITEM#) ON DELETE SET NULL

);

CREATE TABLE WAREHOUSE(

WAREHOUSE# INT ,

CITY VARCHAR(20) NOT NULL,

PRIMARY KEY(WAREHOUSE#)

);

CREATE TABLE SHIPMENT(

ORDERS# INT,

WAREHOUSE# INT,

SHIP\_DATE DATE,

FOREIGN KEY(ORDERS#) REFERENCES ORDERS(ORDERS#) ON DELETE SET NULL,

FOREIGN KEY(WAREHOUSE#) REFERENCES WAREHOUSE(WAREHOUSE#) ON DELETE SET NULL

);

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

INSERT INTO CUSTOMER VALUES(1,'AMIT','AGRA');

INSERT INTO CUSTOMER VALUES(2,'ARJUN','INDIA’);

INSERT INTO CUSTOMER VALUES(3,'SAHIL','DHARWAD');

INSERT INTO CUSTOMER VALUES(4,'SAHITYA','HUBLI');

INSERT INTO CUSTOMER VALUES(5,'SWATHI','MARS');

INSERT INTO ORDERS VALUES(1,'24-DEC-2018',1,1200);

INSERT INTO ORDERS VALUES(2,'24-MAR-2018',2,1600);

INSERT INTO ORDERS VALUES(3,'24-MAY-2018',3,1400);

INSERT INTO ORDERS VALUES(4,'24-JUN-2018',4,1200);

INSERT INTO ORDERS VALUES(5,'24-JUL-2018',5,12300);

INSERT INTO ORDERS VALUES(6,'24-NOV-2018',4,1678);

INSERT INTO ORDERS VALUES(7,'24-NOV-2018',3,2345);

INSERT INTO ITEM VALUES(1,100);

INSERT INTO ITEM VALUES(2,200);

INSERT INTO ITEM VALUES(3,150);

INSERT INTO ITEM VALUES(4,687);

INSERT INTO ITEM VALUES(5,1000);

INSERT INTO ITEM VALUES(6,1000);

INSERT INTO ITEM VALUES(7,1200);

INSERT INTO ITEM VALUES(8,1100);

INSERT INTO ITEM VALUES(9,500);

INSERT INTO ITEM VALUES(10,800);

INSERT INTO ORDER\_ITEM VALUES(1,1,3);

INSERT INTO ORDER\_ITEM VALUES(1,2,3);

INSERT INTO ORDER\_ITEM VALUES(1,3,2);

INSERT INTO ORDER\_ITEM VALUES(2,1,4);

INSERT INTO ORDER\_ITEM VALUES(2,5,1);

INSERT INTO ORDER\_ITEM VALUES(2,2,1);

INSERT INTO ORDER\_ITEM VALUES(3,2,2);

INSERT INTO ORDER\_ITEM VALUES(3,9,2);

INSERT INTO ORDER\_ITEM VALUES(4,10,1);

INSERT INTO ORDER\_ITEM VALUES(4,2,2);

INSERT INTO ORDER\_ITEM VALUES(5,1,3);

INSERT INTO ORDER\_ITEM VALUES(5,5,5);

INSERT INTO ORDER\_ITEM VALUES(5,6,5);

INSERT INTO ORDER\_ITEM VALUES(5,2,10);

INSERT INTO ORDER\_ITEM VALUES(6,9,1);

INSERT INTO ORDER\_ITEM VALUES(6,4,1);

INSERT INTO ORDER\_ITEM VALUES(6,2,2);

INSERT INTO ORDER\_ITEM VALUES(6,1,1);

INSERT INTO ORDER\_ITEM VALUES(6,8,1);

INSERT INTO WAREHOUSE VALUES(1,'AGRA');

INSERT INTO WAREHOUSE VALUES(2,'DELHI');

INSERT INTO WAREHOUSE VALUES(3,'HUBLI');

INSERT INTO WAREHOUSE VALUES(4,'DHARWAD');

INSERT INTO WAREHOUSE VALUES(5,'INDIA');

INSERT INTO WAREHOUSE VALUES(6,'GOA');

INSERT INTO WAREHOUSE VALUES(7,'CHINA');

INSERT INTO WAREHOUSE VALUES(8,'USA');

INSERT INTO WAREHOUSE VALUES(9,'AGRA');

INSERT INTO SHIPMENT VALUES(1,1,'26-DEC-2018');

INSERT INTO SHIPMENT VALUES(2,2,'26-MAR-2018');

INSERT INTO SHIPMENT VALUES(3,3,'26-MAY-2018');

INSERT INTO SHIPMENT VALUES(4,4,'26-JUN-2018');

INSERT INTO SHIPMENT VALUES(5,5,'26-JUL-2018');

INSERT INTO SHIPMENT VALUES(6,6,'26-NOV-2018');

INSERT INTO SHIPMENT VALUES(1,6,'21-NOV-2018');

**(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. (DONE)**

SELECT CNAME,COUNT(\*) AS NO\_OF\_ORDERS ,AVG(ORD\_AMOUNT) AS AVG\_ORDER\_AMT FROM (CUSTOMER C JOIN ORDERS O ON O.CUST#=C.CUST#) GROUP BY(CNAME);

**(iv) List the order# for orders that were shipped from**

**all the warehouses that the company has in a specific city.(DONE)**

SELECT ORDERS# FROM ORDERS WHERE ORDERS# IN (SELECT ORDERS# FROM SHIPMENT WHERE WAREHOUSE# IN (SELECT WAREHOUSE# FROM WAREHOUSE WHERE CITY='AGRA'));

**(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.**

DELETE \* FROM ITEM WHERE ITEM#=7;

**(vi) List the order no, which have max item present.**

SELECT ORDERS# FROM ORDER\_ITEM HAVING COUNT(ORDERS#)=(SELECT MAX(COUNT(\*)) FROM ORDER\_ITEM GROUP BY ORDERS#)GROUP BY ORDERS#;

**(vii) List the warehouse#, which involve/s maximum orders.**

SELECT WAREHOUSE# FROM SHIPMENT HAVING COUNT(ORDERS#)=(SELECT MAX(COUNT(\*)) FROM SHIPMENT GROUP BY WAREHOUSE#) GROUP BY WAREHOUSE#;

**(viii) List the day-wise total order amount in the year 2012.**

SELECT SUM(ORD\_AMOUNT),ODATE FROM ORDERS WHERE ODATE LIKE '%18' GROUP BY ODATE;

**(ix) List the customer name that has placed orders amounting maximum sum.**

SELECT CNAME FROM CUSTOMER WHERE CUST#=(SELECT CUST# FROM ORDERS HAVING SUM(ORD\_AMOUNT)=(SELECT MAX(SUM(ORD\_AMOUNT)) FROM ORDERS GROUP BY CUST#) GROUP BY CUST#);

**(x) List the item# which was never bought in the year 2011.**

SELECT ITEM# FROM ITEM WHERE ITEM# <> ALL (SELECT DISTINCT ITEM# FROM ORDER\_ITEM);

**(xi) List the city that has minimum orders for a particular item.**

NO ANSWER YET FOUND(IF ANYONE GETS IT PLZZ TELL )

**(xii) List the customer\_id, CName and city name whose city is same as that of the city in which**

**warehouses are situated.**

SELECT CUST#,CNAME,CITY FROM CUSTOMER WHERE CITY IN (SELECT CITY FROM WAREHOUSE);

**(xiii) List the customer\_id, Customer name and address of those customers that are non-**

**residents of the cities of any of the warehouses of the company.**

SELECT CUST#,CNAME,CITY FROM CUSTOMER WHERE CITY NOT IN (SELECT CITY FROM WAREHOUSE);

**(xiv) List the customer\_id, CName and city name that have placed atleast one order, whose city**

**is same as that of the city in which warehouse/s are situated.**

SELECT DISTINCT C.CUST#,CNAME,CITY FROM CUSTOMER C,ORDERS O WHERE C.CUST#= O.CUST# AND CITY IN(SELECT DISTINCT CITY FROM WAREHOUSE);

**(xv) List the customer\_id, Customer name and address of customers that have placed atleast one**

**order that are non-residents of the cities of any of the warehouses of the company.**

SELECT DISTINCT C.CUST#,CNAME,CITY FROM CUSTOMER C,ORDERS O WHERE C.CUST#= O.CUST# AND CITY NOT IN(SELECT DISTINCT CITY FROM WAREHOUSE);

TERMWORK 4

# 4. 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)

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

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

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

CREATE TABLE STUDENT(

REGNO VARCHAR(10),

NAME VARCHAR(32) NOT NULL,

MAJOR VARCHAR(15) DEFAULT 'NOT DECIDED',

BDATE DATE,

PRIMARY KEY(REGNO)

);

CREATE TABLE COURSE(

COURSE# INT,

CNAME VARCHAR(10),

DEPT VARCHAR(10),

PRIMARY KEY(COURSE#)

);

CREATE TABLE ENROLL(

REGNO VARCHAR(10),

COURSE# INT,

SEM INT,

MARKS INT DEFAULT 0,

FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO) ON DELETE SET NULL,

FOREIGN KEY(COURSE#) REFERENCES COURSE(COURSE#) ON DELETE SET NULL

);

CREATE TABLE BOOK\_ADOPTION(

COURSE# INT,

SEM INT,

BOOK\_ISBN INT,

FOREIGN KEY(COURSE#) REFERENCES COURSE(COURSE#)ON DELETE SET NULL,

FOREIGN KEY(BOOK\_ISBN) REFERENCES TEXT(BOOK\_ISBN) ON DELETE SET NULL

);

CREATE TABLE TEXT(

BOOK\_ISBN INT,

BOOK\_TITLE VARCHAR(30),

PUBLISHER VARCHAR(20),

AUTHOR VARCHAR(20),

PRIMARY KEY(BOOK\_ISBN)

);

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

INSERT INTO STUDENT VALUES('2SD16CS012','AMIT SHARMA','COMPUTER SCIENCE','24-DEC-1998');

INSERT INTO STUDENT VALUES('2SD16CS022','ARJUN NAYAK','SCIENCE','28-JUL-1998');

INSERT INTO STUDENT VALUES('2SD16CS070','RAKSHITA NAYAK','CS','01-AUG-1998');

INSERT INTO STUDENT VALUES('2SD16CS074','SAHIL MAHALE','SCIENCE','20-NOV-1998');

INSERT INTO STUDENT VALUES('2SD16CS075','SAHITYA S','CS','26-FEB-1998');

INSERT INTO COURSE VALUES(1,'CO','CS');

INSERT INTO COURSE VALUES(2,'BIO','SCIENCE');

INSERT INTO COURSE VALUES(3,'SE','CS');

INSERT INTO COURSE VALUES(4,'C++','CS');

INSERT INTO COURSE VALUES(5,'PHYSICS','SCIENCE');

INSERT INTO ENROLL VALUES('2SD16CS012',4,3,44);

INSERT INTO ENROLL VALUES('2SD16CS022',5,1,98);

INSERT INTO ENROLL VALUES('2SD16CS070',3,5,90);

INSERT INTO ENROLL VALUES('2SD16CS074',2,9,101);

INSERT INTO ENROLL VALUES('2SD16CS075',1,1,95);

INSERT INTO TEXT VALUES(1,'C++','ACP','SWATHI HI');

INSERT INTO TEXT VALUES(2,'PHYSICS','ACD','NEWTON');

INSERT INTO TEXT VALUES(3,'BIO','SIN','DAS');

INSERT INTO TEXT VALUES(4,'SE','ACP','IRAN');

INSERT INTO TEXT VALUES(5,'CO','YT','RAM');

INSERT INTO TEXT VALUES(7,'C#','ACP','VARUN');

INSERT INTO BOOK\_ADOPTION VALUES(4,3,7);

INSERT INTO BOOK\_ADOPTION VALUES(1,1,5);

INSERT INTO BOOK\_ADOPTION VALUES(5,1,2);

INSERT INTO BOOK\_ADOPTION VALUES(3,5,4);

INSERT INTO BOOK\_ADOPTION VALUES(4,3,1);

INSERT INTO BOOK\_ADOPTION VALUES(2,9,3);

**(iii) Demonstrate how you add a new text book to the database and make this book be adopted**

**by some department.**

INSERT INTO TEXT VALUES(6,'C','DCP','SWATHI'); INSERT INTO BOOK\_ADOPTION VALUES(4,3,6);

**(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.**

SELECT C.COURSE#,T.BOOK\_TITLE,T.BOOK\_ISBN FROM COURSE C,BOOK\_ADOPTION B,TEXT T WHERE C.COURSE#=B.COURSE# AND T.BOOK\_ISBN=B.BOOK\_ISBN AND B.COURSE# IN (SELECT B.COURSE# FROM BOOK\_ADOPTION B WHERE C.DEPT='CS' GROUP BY B.COURSE# HAVING COUNT(\*)>2) ORDER BY T.BOOK\_TITLE;

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

SELECT C.DEPT,C.CNAME,T.BOOK\_TITLE,T.PUBLISHER FROM TEXT T,BOOK\_ADOPTION B,COURSE C WHERE C.COURSE#=B.COURSE# AND B.BOOK\_ISBN=T.BOOK\_ISBN AND C.DEPT='CS' AND PUBLISHER='ACP';

**(vi) List the regno and name of the students who have enrolled for maximum number of**

**courses.**

INSERT INTO ENROLL VALUES('2SD16CS012',3,5,42);

SELECT REGNO,NAME FROM STUDENT WHERE REGNO IN (SELECT REGNO FROM ENROLL HAVING COUNT(REGNO) IN (SELECT MAX(COUNT(REGNO)) FROM ENROLL GROUP BY REGNO) GROUP BY REGNO);

**(vii) List the text books which have not been adopted by any of the course.**

INSERT INTO TEXT VALUES(8,'JAVA','DCP','SHREENIDHI');

SELECT \* FROM TEXT WHERE BOOK\_ISBN NOT IN (SELECT BOOK\_ISBN FROM BOOK\_ADOPTION);

**(viii) List the book-ISBNs which have been adopted by both C++ and CO.**

INSERT INTO BOOK\_ADOPTION VALUES(4,1,5);

SELECT BOOK\_ISBN FROM BOOK\_ADOPTION WHERE COURSE# IN (SELECT COURSE# FROM COURSE WHERE CNAME='C++') INTERSECT

SELECT BOOK\_ISBN FROM BOOK\_ADOPTION WHERE COURSE# IN (SELECT COURSE# FROM COURSE WHERE CNAME='CO');

**(ix) List the Book-ISBN and title of book which have been adopted by both C++ and**

**CO**

SELECT BOOK\_ISBN,BOOK\_TITLE FROM TEXT WHERE BOOK\_ISBN IN( (

SELECT BOOK\_ISBN FROM BOOK\_ADOPTION WHERE COURSE# IN (SELECT COURSE# FROM COURSE WHERE CNAME='C++')) INTERSECT

(SELECT BOOK\_ISBN FROM BOOK\_ADOPTION WHERE COURSE# IN (SELECT COURSE# FROM COURSE WHERE CNAME='CO')));

**(x) List the departments that have all its adopted books published by a specific publisher or**

NOT YET DONE (PLZZ TELL ME IF U GET IT)

**(xi) list the publisher in ascending alphabetical order along with descending alphabetical order**

**of Book-ISBN,who have published minimum TWO books.**

SELECT PUBLISHER,BOOK\_ISBN FROM TEXT WHERE PUBLISHER IN (SELECT PUBLISHER FROM TEXT

HAVING COUNT(\*)>=2 GROUP BY PUBLISHER) ORDER BY PUBLISHER ASC,BOOK\_ISBN DESC;

**(xii) List depts which have minimum 2 enrollments for any course that have adopted at least**

**TWO books.**

INSERT INTO BOOK\_ADOPTION VALUES(3,5,5);

SELECT DEPT FROM COURSE WHERE COURSE# IN (SELECT COURSE# FROM ENROLL HAVING COUNT(\*)>=2 GROUP BY COURSE#) AND

COURSE# IN( SELECT COURSE# FROM BOOK\_ADOPTION HAVING COUNT(\*)>=2 GROUP BY COURSE#);

**(xiii) List the courses that have minimum enrollments with maximum books being adopted.**

SELECT COURSE# FROM BOOK\_ADOPTION HAVING COUNT(\*) IN(

SELECT MAX(COUNT(\*)) FROM BOOK\_ADOPTION WHERE COURSE# IN (

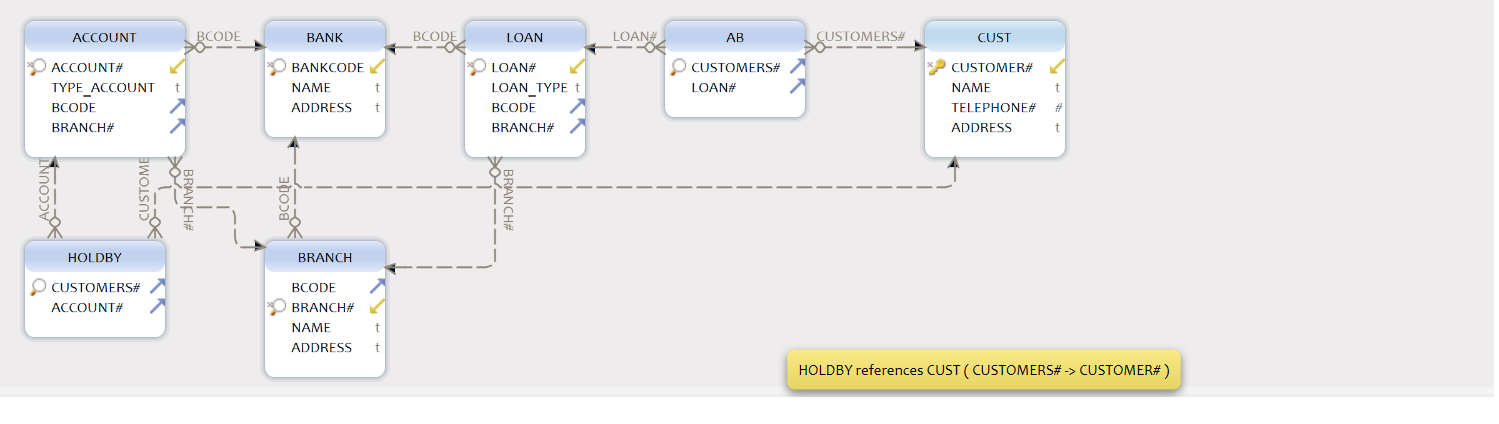
SELECT COURSE# FROM ENROLL HAVING COUNT(\*) IN (SELECT MIN(COUNT(\*)) FROM ENROLL GROUP BY COURSE#) GROUP BY COURSE#) GROUP BY COURSE#)GROUP BY COURSE#;

**(xiv) List the Course No, Title of the book adopted and Author name if author has written more**

**than ONE book.**

SELECT B.COURSE#,T.BOOK\_TITLE,T.AUTHOR FROM BOOK\_ADOPTION B,TEXT T WHERE B.BOOK\_ISBN=T.BOOK\_ISBN AND T.AUTHOR IN(SELECT AUTHOR FROM TEXT HAVING COUNT(\*)>1 GROUP BY AUTHOR);

TERMWORK 5



**CREATION-:**

CREATE TABLE BANK(

BANKCODE INT,

NAME VARCHAR(20),

ADDRESS VARCHAR(30),

PRIMARY KEY(BANKCODE)

);

CREATE TABLE BRANCH(

BCODE INT,

BRANCH# INT,

NAME VARCHAR(20),

ADDRESS VARCHAR(30),

PRIMARY KEY(BRANCH#),

FOREIGN KEY(BCODE) REFERENCES BANK(BANKCODE) ON DELETE SET NULL

);

CREATE TABLE ACCOUNT(

ACCOUNT# INT,

TYPE\_ACCOUNT VARCHAR(12),

BCODE INT,

BRANCH# INT,

PRIMARY KEY(ACCOUNT#),

FOREIGN KEY(BCODE) REFERENCES BANK(BANKCODE) ON DELETE SET NULL,

FOREIGN KEY(BRANCH#) REFERENCES BRANCH(BRANCH#) ON DELETE SET NULL

);

CREATE TABLE LOAN(

LOAN# INT,

LOAN\_TYPE VARCHAR(10),

BCODE INT,

BRANCH# INT,

PRIMARY KEY(LOAN#),

FOREIGN KEY(BCODE) REFERENCES BANK(BANKCODE) ON DELETE SET NULL,

FOREIGN KEY(BRANCH#) REFERENCES BRANCH(BRANCH#) ON DELETE SET NULL

);

CREATE TABLE CUST(

CUSTOMER# INT,

NAME VARCHAR(20),

TELEPHONE# INT,

ADDRESS VARCHAR(30),

PRIMARY KEY(CUSTOMER#)

);

CREATE TABLE AB(

CUSTOMERS# INT,

LOAN# INT,

FOREIGN KEY(CUSTOMER#) REFERENCES CUST(CUSTOMER#) ON DELETE SET NULL,

FOREIGN KEY(LOAN#) REFERENCES LOAN(LOAN#) ON DELETE SET NULL

);

CREATE TABLE HOLDBY(

CUSTOMERS# INT,

ACCOUNT# INT,

FOREIGN KEY(CUSTOMER#) REFERENCES CUST(CUSTOMER#) ON DELETE SET NULL,

FOREIGN KEY(ACCOUNT#) REFERENCES ACCOUNT(ACCOUNT#) ON DELETE SET NULL

);

**INSERTION-:**

INSERT INTO BANK VALUES(301,'SBI','HUBLI');

INSERT INTO BANK VALUES(302,'SBM','DHARWAD');

INSERT INTO BANK VALUES(303,'CANARA','BELGAVI');

INSERT INTO BANK VALUES(304,'KOTAK','PUNE');

INSERT INTO BANK VALUES(305,'HDFC','MUMBAI');

INSERT INTO BRANCH VALUES(301,005,'RAM','HUBLI');

INSERT INTO BRANCH VALUES(301,010,'LAKSH','HUBLI');

INSERT INTO BRANCH VALUES(302,006,'LAKSHMAN','DHARWAD');

INSERT INTO BRANCH VALUES(303,007,'LAWA','BELGAVI');

INSERT INTO BRANCH VALUES(304,008,'KUSH','PUNE');

INSERT INTO BRANCH VALUES(305,009,'BHARAT','MUMBAI');

INSERT INTO LOAN VALUES(900,'EDUCATION',301,005);

INSERT INTO LOAN VALUES(901,'HOME',302,006);

INSERT INTO LOAN VALUES(902,'AGRI',303,007);

INSERT INTO LOAN VALUES(903,'BUSINESS',304,008);

INSERT INTO LOAN VALUES(904,'GOLD',305,009);

INSERT INTO ACCOUNT VALUES(61970,'SAVING',301,005);

INSERT INTO ACCOUNT VALUES(61975,'FIX\_DEP',301,010);

INSERT INTO ACCOUNT VALUES(61971,'FIX\_DEP',302,006);

INSERT INTO ACCOUNT VALUES(61972,'CURRENT',303,007);

INSERT INTO ACCOUNT VALUES(61973,'RECUR\_DEP',304,008);

INSERT INTO ACCOUNT VALUES(61974,'FOREIGN\_EX',305,009);

INSERT INTO CUST VALUES(101,'ARPANA',3427671829,'HUBLI');

INSERT INTO CUST VALUES(102,'APARNA',4328671831,'DHARWAD');

INSERT INTO CUST VALUES(103,'ANJALI',5428571829,'BELGAVI');

INSERT INTO CUST VALUES(104,'SUMANJALI',3427572939,'PUNE');

INSERT INTO CUST VALUES(105,'KUSMANJALI',3427427940,'MUMBAI');

INSERT INTO AB VALUES(101,900);

INSERT INTO AB VALUES(102,901);

INSERT INTO AB VALUES(103,902);

INSERT INTO AB VALUES(104,903);

INSERT INTO AB VALUES(105,904);

INSERT INTO HOLDBY VALUES(101,61970);

INSERT INTO HOLDBY VALUES(101,61975);

INSERT INTO HOLDBY VALUES(102,61971);

INSERT INTO HOLDBY VALUES(103,61972);

INSERT INTO HOLDBY VALUES(104,61973);

INSERT INTO HOLDBY VALUES(105,61974);

**3)FIND ALL THE CUSTOMERS WHO HAVE AT LEAST TWO ACCOUNT AT THE BHARAT.**

SELECT \* FROM CUST WHERE CUSTOMER# IN (SELECT CUSTOMER# FROM HOLDBY WHERE ACCOUNT# IN (SELECT ACCOUNT#

FROM ACCOUNT WHERE BRANCH# IN (SELECT BRANCH# FROM BRANCH WHERE NAME='BHARAT')) HAVING COUNT(\*)>=2 GROUP BY CUSTOMER#);

**4)FIND ALL THE CUSTOMERS WHO HAVE ACCOUNT AT ALL BRANCH LOCATED IN SPECIFIC CITY**

SELECT H.CUSTOMER#,A.ACCOUNT# FROM ACCOUNT A,HOLDBY H WHERE A.ACCOUNT#=H.ACCOUNT# AND BRANCH# IN (SELECT BRANCH# FROM BRANCH WHERE ADDRESS='HUBLI') GROUP BY H.CUSTOMER#,A.ACCOUNT#;

**5)DEMONSTRATE HOW YOU DELETE ALL ACCOUNT TUPLES AT EVERY BRANCH LOCATED IN A SPECIFIC CITY**

DELETE FROM ACCOUNT WHERE BRANCH# IN (SELECT BRANCH# FROM BRANCH WHERE ADDRESS='PUNE');