

Record of Practical Work Subject: DataBase Management System (MySQL)

Name	Kiran M K
USN	1BM19CS073
Semester	4
Section	В
Dept.	CSE

CONTENTS

Expt. No.	Date	Experiment	Page No.
1	28-04-2021	Insurance Database	
2	28-04-2021	Bookdealer Database	
3	05-05-2021	OrderProcessing Database	
4	12-05-2021	Banking Database	
5	26-05-2021	Student Database	

Experiment 1:

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

data types are specified.

PERSON (driver-id #: String, name: String, address: String)

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

ACCIDENT (report-number: int, adate: 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 involved in accidents in 2008
- v. Find the number of accidents in which cars belonging to a specific model were involved.

Code:

create database insurance;

use insurance;

CREATE TABLE PERSON (DRIVER_ID VARCHAR(10),NAME

VARCHAR(20), ADDRESS

VARCHAR(15), PRIMARY KEY(DRIVER_ID));

create table car(regno varchar(10), Model varchar(20), Year date, Primary key(Regno)); create table Accident(report_no int, ADATE DATE, Location varchar(15), Primary key(report_no));

create table owns(driver_id varchar(10),regno varchar(10),primary key(driver_id,regno),

foreign key(driver_id) references person(driver_id) on delete cascade, foreign key(regno) references car(regno) on delete cascade);

CREATE TABLE PARTICIPATED(driver_id varchar(10),regno

varchar(10),report_no int,

damage_amt float, foreign key (driver_id,regno) references OWNS(driver_id,regno) ON DELETE CASCADE,

foreign key (REPORT_NO) references ACCIDENT(REPORT_NO) ON DELETE CASCADE);

INSERT INTO PERSON VALUES("1111","RAMU","K.S.LAYOUT"); INSERT INTO PERSON VALUES("2222","JOHN","INDIRANAGAR"), ("3333","PRIYA","JAYNAGAR"), ("4444","GOPAL","WHITEFIELD"), ("5555","LATHA","VIJAYNAGAR");

INSERT INTO CAR VALUES("KA04Q2301","MARUTHI-DX","2000-01-01"), ("KA05P1000","FORDICON","2000-01-01"), ("KA03L1234","ZEN-VXI","1999-01-01"), ("KA03L9999","MARUTHI-DX","2002-01-01"), ("KA01P4020","INDICA-VX","2002-01-01");

INSERT INTO ACCIDENT VALUES(12,"2002-06-01","M G ROAD"), (200,"2002-12-10","DOUBLE ROAD"),(300,"1999-07-23","M G ROAD"), (25000,"2000-06-11","RESIDENCY ROAD"),(26500,"2001-10-16","RICHMOND ROAD");

INSERT INTO OWNS VALUES("1111","KA04Q2301"),("1111","KA05P1000"), ("2222","KA03L1234"),("3333","KA03L9999"),("4444","KA01P4020");

INSERT INTO PARTICIPATED VALUES("1111","KA04Q2301",12,20000), ("2222","KA03L1234",200,500), ("3333","KA03L9999",300,10000), ("4444","KA01P4020",25000,2375), ("1111","KA05P1000",26500,70000);

UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT NO=12 AND REGNO="KA04Q2301";

SELECT * FROM PARTICIPATED; SELECT COUNT(*) FROM ACCIDENT WHERE ADATE LIKE "2002- - ";

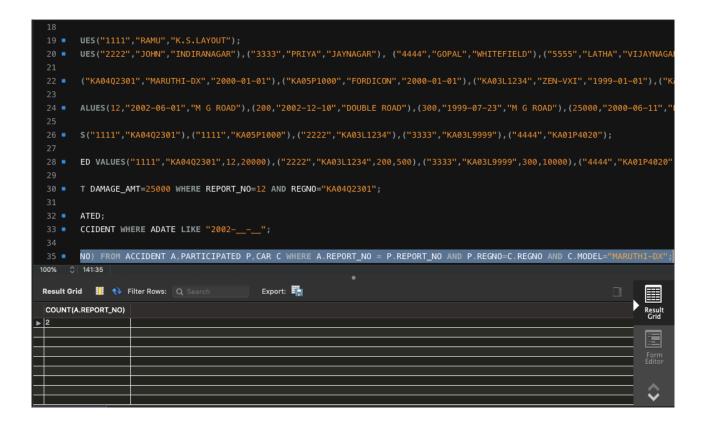
SELECT COUNT(A.REPORT_NO) FROM ACCIDENT A, PARTICIPATED P, CAR C WHERE A.REPORT_NO = P.REPORT_NO AND P.REGNO=C.REGNO AND C.MODEL="MARUTHI-DX";

Output:



Result Grid

COUNT(*)



Experiment 2:

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)

- 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 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%.

Code:

CREATE DATABASE BOOKDEALER;

USE BOOKDEALER;

CREATE TABLE AUTHOR(AUTHOR_ID INT,NAME VARCHAR(30),CITY VARCHAR(30),COUNTRY VARCHAR(30));

CREATE TABLE PUBLISHER(PUBLISHER_ID INT,NAME VARCHAR(30),CITY VARCHAR(30),COUNTRY VARCHAR(30));

CREATE TABLE AUTHOR(AUTHOR_ID INT,NAME VARCHAR(30),CITY

VARCHAR(30),COUNTRY VARCHAR(30),PRIMARY KEY(AUTHOR_ID));

CREATE TABLE PUBLISHER(PUBLISHER_ID INT,NAME VARCHAR(30),CITY VARCHAR(30),COUNTRY VARCHAR(30),PRIMARY KEY(PUBLISHER_ID));

CREATE TABLE CATEGORY(CATEGORY ID INT.DESCRIPTION

VARCHAR(200), PRIMARY KEY(CATEGORY_ID));

CREATE TABLE CATALOG(BOOK_ID INT,TITLE VARCHAR(50), AUTHOR_ID INT, PUBLISHER ID INT,CATEGORY ID INT,AYEAR INT,PRICE

INT,FOREIGN KEY(AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID) ON DELETE CASCADE, FOREIGN KEY(PUBLISHER ID) REFERENCES

PUBLISHER (PUBLISHER ID) ON DELETE CASCADE, FOREIGN

KEY(CATEGORY_ID) REFERENCES CATEGORY(CATEGORY_ID) ON DELETE CASCADE);

CREATE TABLE CATALOG(BOOK_ID INT,TITLE VARCHAR(50), AUTHOR_ID INT, PUBLISHER_ID INT,CATEGORY_ID INT,AYEAR INT,PRICE INT,FOREIGN KEY(AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID) ON DELETE CASCADE, FOREIGN KEY(PUBLISHER ID) REFERENCES

PUBLISHER(PUBLISHER_ID) ON DELETE CASCADE, FOREIGN

```
KEY(CATEGORY ID) REFERENCES CATEGORY(CATEGORY ID) ON
DELETE CASCADE, PRIMARY KEY(BOOK ID));
CREATE TABLE ORDERDETAILS(ORDER NO INT, BOOK ID INT,
QUANTITY INT, FOREIGN KEY(BOOK ID) REFERENCES
CATALOG(BOOK ID));
insert into author values(1001, "Teras Chan", "CA", "USA");
insert into author values(1002, "Stevens", "Zombi", "Uganda");
insert into author values(1003, "M Mano", "Cair", "Canada");
insert into author values(1004,"Karthik B P","New York","USA");
insert into author values(1005, "Willian Stalling", "Las Vegas", "USA");
insert into publisher values(1,"Pearson","New York","USA");
insert into publisher values(2,"EEE","New South Vales","USA");
insert into publisher values(3,"PHI","Delhi","India");
insert into publisher values(4,"Willey","Berlin","Germany");
insert into publisher values(5,"MGH","New York","USA");
insert into category values(1001,"Computer Science");
insert into category values(1002,"Algorithm Design");
insert into category values(1003, "Electronics");
insert into category values(1004, "Programming");
insert into category values(1005,"Operating Systems");
insert into catalog values(11,"Unix System",1001,1,1001,2000,251);
insert into catalog values(12,"Digital Signals",1002,2,1003,2001,425);
insert into catalog values(13,"Logic Design",1003,3,1002,1999,225);
insert into catalog values(14, "Server Prg", 1004, 4, 1004, 2001, 333);
insert into catalog values(15, "Linux OS", 1005, 5, 1005, 2003, 326);
insert into catalog values(16,"C++ Bible",1005,5,1001,2000,526);
insert into catalog values(17, "Cobol Handbook", 1005, 4, 1001, 2000, 658);
insert into orderdetails values(1,11,5);
insert into orderdetails values(2,12,8);
insert into orderdetails values(3.13.15):
insert into orderdetails values(4,14,22);
insert into orderdetails values(5,15,3);
insert into orderdetails values(6,17,10);
/*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.*/
```

SELECT * FROM AUTHOR WHERE AUTHOR_ID IN (SELECT AUTHOR_ID FROM CATALOG WHERE AYEAR>2000 AND PRICE> (SELECT AVG(PRICE) FROM CATALOG) GROUP BY AUTHOR_ID HAVING COUNT(*)>1);

/*Find the author of the book which has maximum sales.*/

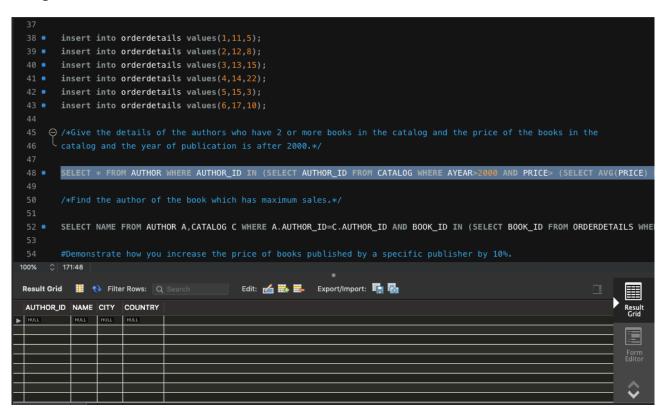
SELECT NAME FROM AUTHOR A, CATALOG C WHERE A.AUTHOR_ID=C.AUTHOR_ID AND BOOK_ID IN (SELECT BOOK_ID FROM ORDERDETAILS WHERE QUANTITY= (SELECT MAX(QUANTITY) FROM ORDERDETAILS));

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

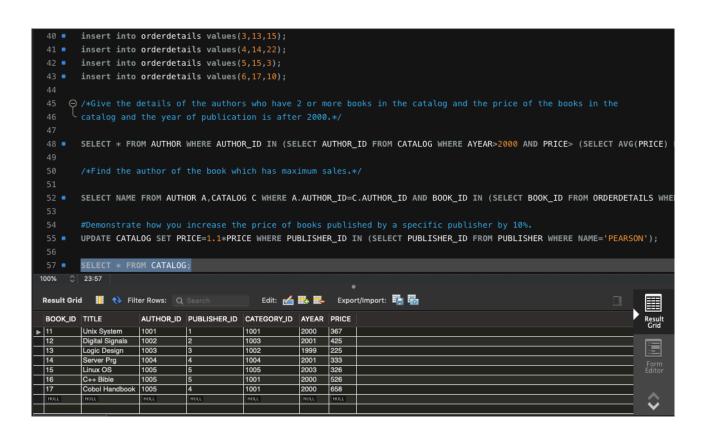
UPDATE CATALOG SET PRICE=1.1*PRICE WHERE PUBLISHER_ID IN (SELECT PUBLISHER ID FROM PUBLISHER WHERE NAME='PEARSON');

SELECT * FROM AUTHOR;

Output:



```
insert into orderdetails values(3,13,15);
        insert into orderdetails values(4,14,22);
 41 •
        insert into orderdetails values(5,15,3);
 43 •
        insert into orderdetails values(6,17,10);
        SELECT * FROM AUTHOR WHERE AUTHOR_ID IN (SELECT AUTHOR_ID FROM CATALOG WHERE AYEAR>2000 AND PRICE> (SELECT AVG(PRICE)
        SELECT NAME FROM AUTHOR A, CATALOG C WHERE A.AUTHOR_ID=C.AUTHOR_ID AND BOOK_ID IN (SELECT BOOK_ID FROM ORDERDETAILS WHE
 52 •
        UPDATE CATALOG SET PRICE=1.1*PRICE WHERE PUBLISHER_ID IN (SELECT PUBLISHER_ID FROM PUBLISHER WHERE NAME='PEARSON');
 57 • SELECT * FROM AUTHOR;
100% 🗘 174:52
                                            Export:
Result Grid ## Tilter Rows: Q Search
 NAME
▶ Karthik B P
```



Experiment 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)

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 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 warehouses that the company has in a

specific city.

v) Demonstrate how you delete item# 10 from the ITEM table and make that field null in the

ORDER_ITEM

table.

Code:

CREATE DATABASE ORDERPROCESSING; USE ORDERPROCESSING;

CREATE TABLE CUSTOMER(CUST_NO INT,CNAME VARCHAR(30),CITY VARCHAR(30), PRIMARY KEY(CUST_NO));

CREATE TABLE ORDERDET(ORDER_NO INT,ODATE DATE, CUST_NO INT,ORDER_AMT INT,PRIMARY KEY(ORDER_NO), FOREIGN KEY(CUST_NO) REFERENCES CUSTOMER(CUST_NO) ON DELETE CASCADE);

CREATE TABLE ITEM(ITEM_NO INT, UNIT_PRICE INT, PRIMARY KEY(ITEM NO));

CREATE TABLE ORDERITEM(ORDER_NO INT, ITEM_NO INT, QTY INT, FOREIGN KEY(ORDER_NO) REFERENCES ORDERDET(ORDER_NO) ON DELETE CASCADE, FOREIGN KEY(ITEM_NO) REFERENCES ITEM(ITEM_NO) ON DELETE CASCADE);

```
CREATE TABLE WAREHOUSE (WAREHOUSE NO INT, CITY VARCHAR(30),
PRIMARY KEY(WAREHOUSE NO));
CREATE TABLE SHIPMENT (ORDER NO INT, WAREHOUSE NO INT,
SHIP DATE DATE, FOREIGN KEY (ORDER NO) REFERENCES
ORDERDET(ORDER NO) ON DELETE CASCADE, FOREIGN
KEY(WAREHOUSE NO) REFERENCES WAREHOUSE(WAREHOUSE NO) ON
DELETE CASCADE);
INSERT INTO CUSTOMER VALUES(771,"PUSHPA K","BANGALORE"),
(772, "SUMAN", "MUMBAI"), (773, "SOURAV", "CALICUT"),
(774, "LAILA", "HYDERABAD"), (775, "FAIZAL", "BANGALORE");
INSERT INTO ORDERDET VALUES(111," 2002-01-22", 771, 18000),
(112,"2002-07-30", 774, 6000),
(113, "2003-04-03", 775, 9000),
(114, "2003-11-03", 775, 29000),
(115, "2003-12-10", 773, 29000),
(116, "2004-08-19", 772, 56000),
(117, "2004-09-10", 771, 20000),
(118, "2004-11-20", 775, 29000),
(119, "2005-02-13", 774, 29000),
(120, "2005-10-13", 775, 29000);
INSERT INTO ITEM VALUES(5001,503),
(5002, 750),
(5003, 150),
(5004, 600),
(5005, 890);
INSERT INTO ORDERITEM VALUES(111, 5001, 50),
(112, 5003, 20),
(113, 5002, 50),
(114, 5005, 60),
(115, 5004, 90),
(116, 5001, 10),
(117, 5003, 80),
(118, 5005, 50),
(119, 5002, 10),
(120, 5004, 45);
INSERT INTO WAREHOUSE VALUES(1, "DELHI"),
(2, "BOMBAY"),
(3, "CHENNAI"),
(4,"BANGALORE"),
(5,"BANGALORE"),
```

```
(6, "DELHI"),
(7,"BOMBAY"),
(8, "CHENNAI"),
(9, "DELHI"),
(10, "BANGALORE");
INSERT INTO SHIPMENT VALUES(111, 1, "2002-02-10"),
(112, 5, "2002-09-10"),
(113, 8, "2003-02-10"),
(114, 3, "2003-12-10"),
(115, 9, "2004-01-19"),
(116, 1, "2004-09-20"),
(117, 5, "2004-09-10"),
(118, 7, "2004-11-30"),
(119, 7, "2005-04-30"),
(120, 6, "2005-12-21");
/*List the order# for orders that were shipped from all warehouses that the company
has in a
specific city.*/
SELECT O.ORDER NO FROM ORDERDET O, WAREHOUSE W, SHIPMENT S
WHERE W.WAREHOUSE NO = S.WAREHOUSE NO AND W.CITY =
"BANGALORE":
/*Demonstrate how you delete item# 10 from the ITEM table and make that field null
in the
ORDER ITEM*/
DELETE FROM ITEM WHERE ITEM NO = 5005;
SELECT * FROM ORDERITEM;
/*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.*/
CREATE VIEW DIFFTABLE(CUSTNAME, NO OF ORDERS,
AVG ORDER AMT) AS SELECT C.CNAME, COUNT(O.ORDER NO),
AVG(O.ORDER AMT) FROM CUSTOMER C, ORDERDET O WHERE
C.CUST NO = O.CUST NO GROUP BY O.CUST NO;
```

SELECT * FROM DIFFTABLE;

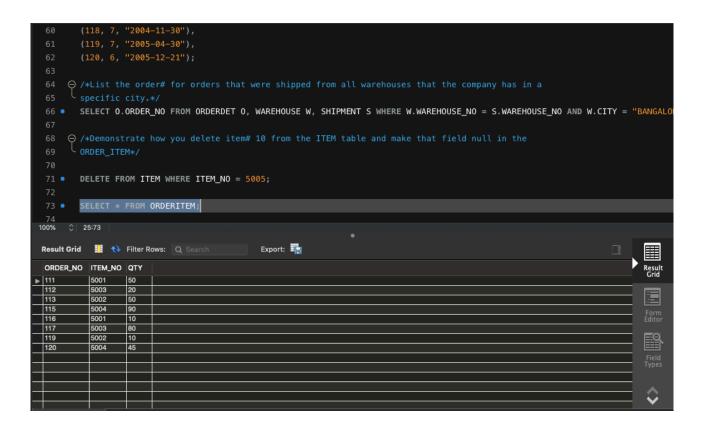
Output:

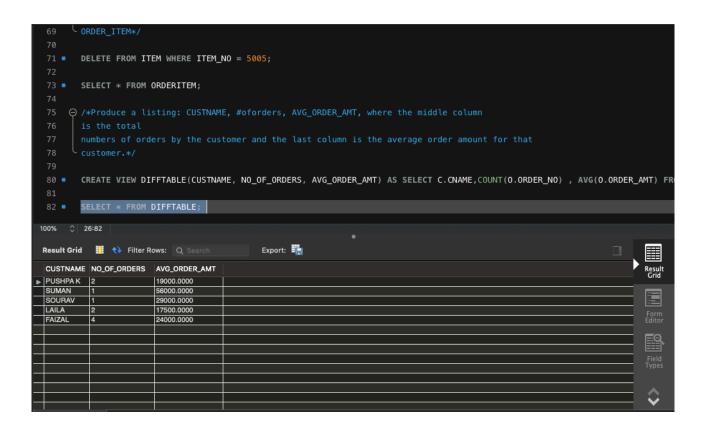
```
(117, 5, "2004-09-10"),
          (117, 3, 2004-09-10),

(118, 7, "2004-11-30"),

(119, 7, "2005-04-30"),

(120, 6, "2005-12-21");
        specific city.*/
  66 SELECT 0.ORDER_NO FROM ORDERDET 0, WAREHOUSE W, SHIPMENT S WHERE W.WAREHOUSE_NO = S.WAREHOUSE_NO AND W.CITY = "BANGAL
        ORDER_ITEM*/
         DELETE FROM ITEM WHERE ITEM_NO = 5005;
  71 •
73 • SELECT * FROM ORDERITEM;
100% $\( | \) 123:66 \( | \)
Result Grid ## Filter Rows: Q Search
                                                     Export:
 ORDER NO
111
117
 116
116
115
112
 119
 113
```





Experiment 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, customer-street: String,

customer-city: String)

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

BORROWER (customer-name: String, loan-number: 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.

Code:

CREATE DATABASE BANKING;

USE BANKING;

CREATE TABLE BRANCH(BRANCH_NAME VARCHAR(30), BRANCH_CITY VARCHAR(30), ASSETS FLOAT, PRIMARY KEY(BRANCH_NAME));

CREATE TABLE ACCOUNTS(ACC_NO INT, BRANCH_NAME VARCHAR(30), BALANCE FLOAT, PRIMARY KEY(ACC_NO),FOREIGN

KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME) ON DELETE CASCADE);

CREATE TABLE CUSTOMER(ACC_NO INT,CUST_NAME VARCHAR(30), CUST_STREET VARCHAR(100), CUST_CITY VARCHAR(30), PRIMARY KEY(CUST_NAME), FOREIGN KEY ACCOUNTS(ACC_NO) REFERENCES ACCOUNTS(ACC_NO) ON DELETE CASCADE ON UPDATE CASCADE); CREATE TABLE LOAN(LOAN_NO INT, BRANCH_NAME VARCHAR(30), AMOUNT FLOAT, PRIMARY KEY(LOAN_NO), FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME) ON

KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME) ON DELETE CASCADE);

CREATE TABLE BORROWER(CUST_NAME VARCHAR(30), LOAN_NO INT, FOREIGN KEY(CUST_NAME) REFERENCES CUSTOMER(CUST_NAME) ON DELETE CASCADE, FOREIGN KEY(LOAN_NO) REFERENCES LOAN(LOAN_NO) ON DELETE CASCADE);

```
INSERT INTO BRANCH VALUES("SBI PD NAGAR", "BANGALORE", 200000),
("SBI RAJAJI NAGAR", "BANGALORE", 500000),
("SBI JAYANAGAR", "BANGALORE", 660000),
("SBI VIJAY NAGAR", "BANGALORE", 870000),
("SBI HOSAKEREHALLI", "BANGALORE", 550000);
INSERT INTO BRANCH VALUES ("CANARA J C ROAD", "BANGALORE",
3000000).
("CANARA-SYNDICATE MANIPAL"."MANIPAL".2000000).
("CANARA K M MARG", "UDUPI", 100000),
("CANARA PAHARGANJ", "DELHI", 200000),
("CANARA LAJPATHNAGAR", "DELHI", 300000);
INSERT INTO ACCOUNTS VALUES(1234602, "SBI HOSAKEREHALLI", 5000),
(1234603, "SBI VIJAY NAGAR", 5000),
(1234604, "SBI JAYANAGAR", 5000),
(1234605, "SBI RAJAJI NAGAR", 10000),
(1234503, "SBI VIJAY NAGAR", 40000),
(1234504, "SBI PD NAGAR", 4000);
INSERT INTO ACCOUNTS VALUES(282577, "CANARA J C ROAD", 15000),
(235887, "CANARA J C ROAD", 25000),
(367733, "CANARA J C ROAD", 56300),
(462822, "CANARA J C ROAD", 23500),
(127298, "CANARA-SYNDICATE MANIPAL", 13000),
(877373,"CANARA-SYNDICATE MANIPAL",19000),
(122933, "CANARA-SYNDICATE MANIPAL", 16770),
(544556,"CANARA K M MARG",12000),
(896565,"CANARA K M MARG",12300),
(453433,"CANARA PAHARGANJ",67000),
(453462,"CANARA PAHARGANJ", 34000),
(232377, "CANARA LAJPATHNAGAR", 12000),
(655665, "CANARA LAJPATHNAGAR", 23000);
INSERT INTO CUSTOMER VALUES(1234602,"KEZAR", "M G ROAD",
"BANGALORE"),
(1234603,"LAL KRISHNA","ST MKS ROAD","BANGALORE"),
(1234604, "RAHUL", "AUGSTEN ROAD", "BANGALORE"),
(1234605, "LALLU", "V S ROAD", "BANGALORE"),
(1234503,"FAIZAL","RESEDENCY ROAD","BANGALORE"),
(1234504, "RAJEEV", "DICKNSN ROAD", "BANGALORE");
INSERT INTO CUSTOMER
```

VALUES(282577,"KRISHNA","VIJAYNAGAR","BANGALORE"),

(235887,"ANIRUDH","BANASHANKARI 5TH STAGE","BANGALORE"),

```
(367733,"CHIRANTH","HOSAHALLI 5TH MAIN","BANGALORE"),
(462822,"MAITHILI","RAJAJINAGAR 4TH BLOCK","BANGALORE"),
(127298, "SAHANA", "GANDHI ROAD", "MANIPAL"),
(877373, "SARIKA", "KASTURBA ROAD", "MANIPAL"),
(122933, "SANJAY", "PAI MARG", "MANIPAL"),
(544556,"SHREYA","NEW THARAGUPET","UDUPI"),
(896565,"UDISHA","COURT ROAD","UDUPI"),
(453433, "SHAMBHAVI", "KAROLBAGH", "DELHI"),
(453462, "PRANAV", "PAHARGANJ", "DELHI"),
(655665,"VISHVESH","LAJPATHNAGAR","DELHI"),
(232377, "VAISHAK", "LAJPATHNAGAR", "DELHI");
INSERT INTO LOAN VALUES(10011, "SBI JAYANAGAR", 10000),
(10012, "SBI VIJAY NAGAR", 5000),
(10013, "SBI HOSAKEREHALLI", 20000),
(10014, "SBI PD NAGAR", 15000),
(10015, "SBI RAJAJI NAGAR", 25000);
INSERT INTO BORROWER VALUES ("KEZAR", 10011),
("LAL KRISHNA", 10012),
("RAHUL", 10013),
("LALLU", 10014),
("LAL KRISHNA", 10015);
CREATE TABLE DEPOSITORS(ACC NO INT, CUST NAME VARCHAR(30),
FOREIGN KEY(ACC NO) REFERENCES ACCOUNTS(ACC NO) ON DELETE
CASCADE, FOREIGN KEY(CUST NAME) REFERENCES
CUSTOMER(CUST NAME) ON DELETE CASCADE);
INSERT INTO DEPOSITORS VALUES(1234602, "KEZAR"),
(1234603, "LAL KRISHNA"),
(1234604,"RAHUL"),
(1234605,"LALLU"),
(1234503, "FAIZAL"),
(1234504,"RAJEEV");
INSERT INTO DEPOSITORS VALUES(282577, "KRISHNA"),
(235887, "ANIRUDH"),
(367733,"CHIRANTH"),
(462822,"MAITHILI"),
(127298, "SAHANA"),
(877373, "SARIKA"),
(122933, "SANJAY"),
(544556, "SHREYA"),
```

(896565,"UDISHA"), (453433,"SHAMBHAVI"),

(453462, "PRANAV"),

(655665,"VISHVESH"),

(232377,"VAISHAK");

/*Find all the customers who have at least two accounts at

the Main branch.*/

SELECT CUST_NAME FROM DEPOSITORS D, ACCOUNTS A WHERE D.ACC_NO = A.ACC_NO AND A.BRANCH_NAME = "SBI VIJAY NAGAR" GROUP BY D.CUST_NAME HAVING COUNT(A.ACC_NO)>=2;

/*Find all the customers who have an account at all the

branches located in a specific city.*/

SELECT * FROM BRANCH;

SELECT * FROM DEPOSITORS;

SELECT * FROM ACCOUNTS;

SELECT D.CUST NAME FROM ACCOUNTS A, BRANCH B, DEPOSITORS D

WHERE B.BRANCH NAME=A.BRANCH NAME AND

A.ACC NO=D.ACC NO AND

B.BRANCH CITY='DELHI'

GROUP BY D.CUST NAME

HAVING COUNT(DISTINCT B.BRANCH NAME)=(SELECT

COUNT(BRANCH NAME)

FROM BRANCH

WHERE BRANCH_CITY='DELHI');

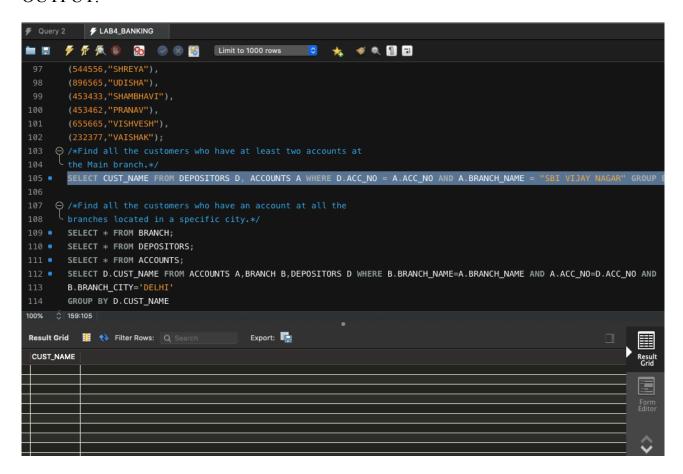
/*Demonstrate how you delete all account tuples at every

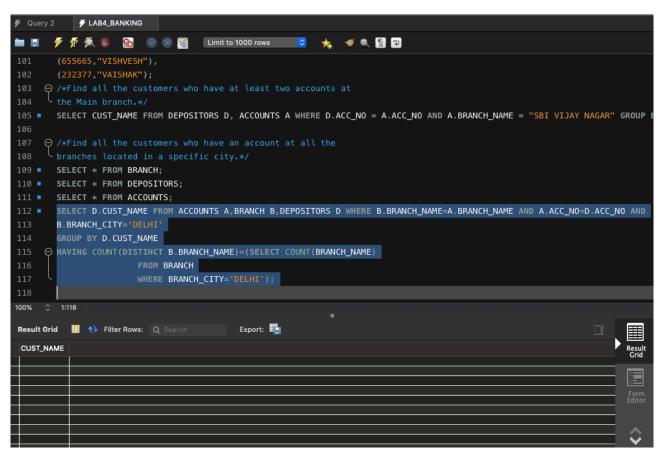
branch located in a specific city.*/

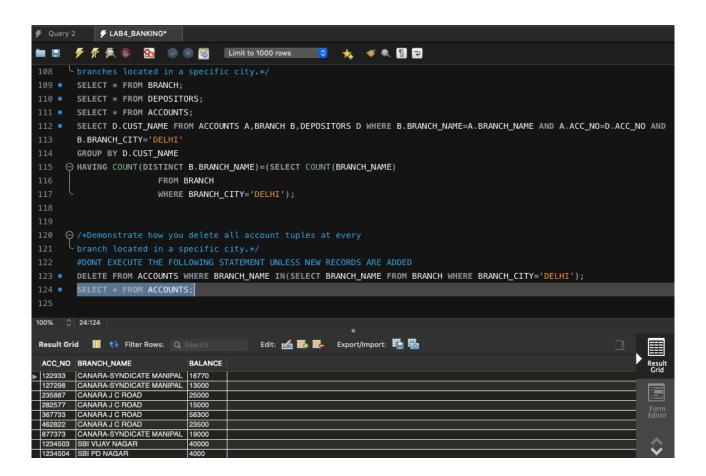
#DONT EXECUTE THE FOLLOWING STATEMENT UNLESS NEW RECORDS ARE ADDED

DELETE FROM ACCOUNTS WHERE BRANCH_NAME IN(SELECT BRANCH_NAME FROM BRANCH WHERE BRANCH_CITY='UDUPI'); SELECT * FROM ACCOUNTS;

OUTPUT:







Experiment 5:

Consider the following database of student enrollment in courses and books adopted for

each course.

STUDENT (regno: String, name: String, major: String, bdate: date)

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

ENROLL (regno: String, cname: String, 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.
- 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.

Code:

CREATE DATABASE STUDENT_ENROLLMENT; USE STUDENT ENROLLMENT;

CREATE TABLE STUDENT(REGNO VARCHAR(10), NAME VARCHAR(30), MAJOR VARCHAR(10), BDATE DATE, PRIMARY KEY(REGNO));

CREATE TABLE COURSE(COURSE_NO INT, CNAME VARCHAR(30), DEPT VARCHAR(4), PRIMARY KEY(COURSE_NO));

CREATE TABLE ENROLL(REGNO VARCHAR(10), COURSE_NO INT, SEM INT, MARKS INT, FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(COURSE NO) REFERENCES

COURSE(COURSE NO) ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE TEXTBOOK(ISBN INT, TITLE VARCHAR(30), PUBLISHER VARCHAR(30), AUTHOR VARCHAR(30), PRIMARY KEY(ISBN));

CREATE TABLE ADOPTION(COURSE_NO INT, SEM INT, ISBN INT, FOREIGN KEY(COURSE_NO) REFERENCES COURSE(COURSE_NO) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY(ISBN) REFERENCES TEXTBOOK(ISBN) ON DELETE CASCADE ON UPDATE CASCADE);

```
INSERT INTO STUDENT VALUES("CS01", "RAM", "DS", "1986-03-12"), ("IS02", "SMITH", "USP", "1987-12-23"), ("EC03", "AHMED", "SNS", "1985-04-17"), ("CS03", "SNEHA", "DBMS", "1987-01-01"), ("TC05", "AKHILA", "EC", "1986-10-06");
```

INSERT INTO COURSE VALUES(11 ,"DS" ,"CS"), (22 ,"USP" ,"IS"),

```
(44,"DBMS","CS"),
(55,"EC","TC");
INSERT INTO ENROLL VALUES("CS01", 11, 4,85),
("IS02",22,6,80),
("EC03", 33, 2, 80),
("CS03", 44, 6, 75),
("TC05", 55, 2, 8);
INSERT INTO TEXTBOOK VALUES(1, "DS and C", "Princeton", "Padma Reddy"),
(2, "Fundamentals of DS", "Princeton", "Godse"),
(3, "Fundamentals of DBMS", "Princeton", "Navathe"),
(4, "SQL", "Princeton", "Foley"),
(5,"Electronic circuits", "TMH", "Elmasri"),
(6,"Adv unix prog","TMH","Stevens");
INSERT INTO ADOPTION VALUES(11, 4, 1),
(11,4,2),
(44,6,3),
(44,6,4),
(55, 2, 5),
(22, 6, 6);
#Demonstrate how you add a new text book to the database and make this book be adopted by some
```

department.

INSERT INTO TEXTBOOK VALUES(7, "Operating System Concepts", "Wiley", "Silberschatz-Galvin-Gagne");

INSERT INTO ADOPTION VALUES(55, 2, 7);

SELECT * FROM TEXTBOOK;

(33, "SNS", "EC"),

/*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_NO ,T.ISBN,T.TITLE FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE C.DEPT = "CS" AND C.COURSE NO = A.COURSE NO AND A.ISBN = T.ISBN;

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

SELECT DISTINCT C.DEPT FROM COURSE C

WHERE C.DEPT IN (SELECT C.DEPT FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE T.PUBLISHER = "PRINCETON" AND C.COURSE NO = A.COURSE NO AND A.ISBN = T.ISBN

AND C.DEPT NOT IN (SELECT C.DEPT FROM COURSE C, TEXTBOOK T, ADOPTION A WHERE T.PUBLISHER != "PRINCETON" AND C.COURSE NO = A.COURSE NO AND A.ISBN = T.ISBN);

OUTPUT:

