



fsmk.org/labmanual

**V
SEMESTER
FOR CS Branch
Database Application Lab
10CSL54**

LAB MANUAL

VTU SYLLABUS 2010

**FREE SOFTWARE MOVEMENT
KARNATAKA**



Content

1. Foreword
2. Contributors
3. About
4. Introduction to MySQL
5. Syllabus
6. Problem 1: Student Class
7. Problem 2: Airline Flight
8. Problem 3: Student Course Books
9. Problem 4: Book dealer
10. Problem 5: Banking Enterprise

Foreword

"Free Software is the future, future is ours" is the motto with which Free Software Movement Karnataka has been spreading Free Software to all parts of society, mainly amongst engineering college faculty and students. However our efforts were limited due to number of volunteers who could visit different colleges physically and explain what is Free Software and why colleges and students should use Free Software. Hence we were looking for ways to reach out to colleges and students in a much larger way. In the year 2013, we conducted two major Free Software camps which were attended by close to 160 students from 25 different colleges. But with more than 150 engineering colleges in Karnataka, we still wanted to find more avenues to reach out to students and take the idea of free software in a much larger way to them.

Lab Manual running on Free Software idea was initially suggested by Dr. Ganesh Aithal, Head of Department, CSE, P.A. Engineering College and Dr. Swarnajyothi L., Principal, Jnana Vikas Institute of Technology during our various interactions with them individually. FSMK took their suggestions and decided to create a lab manual which will help colleges to migrate their labs to Free Software, help faculty members to get access to good documentation on how to conduct various labs in Free Software and also help students by providing good and clear explanations of various lab programs specified by the university. We were very clear on the idea that this lab manual should be produced also from the students and faculty members of the colleges as they knew the right way to explain the problems to a large audience with varying level knowledge in the subject. FSMK promotes freedom of knowledge in all respects and hence we were also very clear that the development and release of this lab manual should be under Creative Commons License so that colleges can adopt the manual and share, print, distribute it to their students and thereby helping us in spreading free software.

Based on this ideology, we decided to conduct a documentation workshop for college faculty members where they all could come together and help us produce this lab manual. As this was a first attempt for even FSMK, we decided to conduct a mock documentation workshop for one day at Indian Institute of Science, Bangalore on 12 Jan, 2014. Close to 40 participants attended it, mainly our students from various colleges and we tried documenting various labs specified by VTU. Based on this experience, we conducted a 3 day residential documentation workshop jointly organized with Jnana Vikas Institute of Technology, Bidadi at their campus from 23 January, 2014. It was attended by 16 faculty members of different colleges and 40 volunteers from FSMK. The documentation workshop was sponsored by Spoken Tutorial Project, an initiative by Government of India to promote IT literacy through Open Source software. Spoken Tutorials are very good learning material to learn about various Free Software tools and hence the videos are excellent companion to this Lab Manual. The videos themselves are released under Creative Commons license, so students can easily download them and share it with others. We would highly recommend our students to go through the Spoken Tutorials while using this Lab Manual and web links to the respective spoken tutorials are shared within the lab manual also.

Finally, we are glad that efforts and support by close to 60 people for around 3 months has led to creation of this Lab Manual. However like any Free Software project, the lab manual will go through constant improvement and we would like the faculty members and students to send us regular feedback on how we can improve the quality of the lab manual. We are also interested to extend the lab manual project to cover MCA departments and ECE departments and are looking for volunteers who can put the effort in this direction. Please contact us if you are interested to support us.

Contributors

Following is the list of all the volunteers who contributed to the making of the Lab Manual.

- Ananda Kumar H N, Faculty, A.T.M.E College of Engineering, Mysore
- Arun Chavan L, Faculty, The Oxford College of Engineering, Bangalore
- Bhavya D, Faculty, P.E.S. College of Engineering, Mandya
- Bindu Madavi K P, Faculty, The Oxford College of Engineering, Bangalore
- Byregowda B K, Faculty, Sir M. Visvesvaraya Institute of Technology, Bangalore
- Deepika, Faculty, P.E.S. College of Engineering, Mandya
- Kiran B, Faculty, A.T.M.E College of Engineering, Mysore
- Manjunatha H C, Faculty, Sir M. Visvesvaraya Institute of Technology, Bangalore
- Neeta Ann Jacob, Faculty, The Oxford College of Engineering, Bangalore
- Rajesh N, Faculty, Sir M. Visvesvaraya Institute of Technology, Bangalore
- Shashidhar S, Faculty, A.T.M.E College of Engineering, Mysore
- Shwetha M K, Faculty, P.E.S. College of Engineering, Mandya
- Abdul Nooran, Student, Vivekananda College of Engineering and Technology, Puttur
- Abhiram R., Student, P.E.S Institute of Technology, Bangalore South Campus
- Ajith K.S., Student, The Oxford College of Engineering, Bangalore
- Akshay Gudi, Student, P.E.S. College of Engineering, Mandya
- Arjun M.Y., Student, P.E.S. College of Engineering, Mandya
- Chaitra Kulkarani, Student, Government Engineering College, Hassan
- John Paul S., Student, Jnana Vikas Institute of Technology, Bidadi
- Karan Jain, Student, P.E.S Institute of Technology, Bangalore South Campus
- Kuna Sharathchandra, Student, P.E.S. College of Engineering, Mandya
- Manas J.K., Student, Dr. Ambedkar Institute of Technology, Bangalore
- Manu H., Student, P.E.S. College of Engineering, Mandya
- Meghana S., Student, Government Engineering College, Hassan
- Nagaraj, Student, Vivekananda College of Engineering and Technology, Puttur
- Nandan Hegde, Student, Vivekananda College of Engineering and Technology, Puttur
- Narmada B., Student, P.E.S Institute of Technology, Bangalore South Campus
- Nawaf Abdul, Student, P.A. College of Engineering, Mangalore
- Nitesh A. Jain, Student, B.M.S. Institute of Technology, Bangalore
- Nitin R., Student, The Oxford College of Engineering, Bangalore
- Padmavathi K., Student, B.M.S. Institute of Technology, Bangalore
- Poojitha Konet, Student, B.M.S. Institute of Technology, Bangalore
- Rahul Kondi, Student, S.J.B. Institute of Technology, Bangalore
- Rohit G.S., Student, Dr. Ambedkar Institute of Technology, Bangalore
- Samruda, Student, Government Engineering College, Hassan
- Samyama H.M., Student, Government Engineering College, Hassan
- Santosh Kumar, Student, Dr. Ambedkar Institute of Technology, Bangalore
- Shashank M C., Student, S.J.B. Institute of Technology, Bangalore
- Soheb Mohammed, Student, P.E.S Institute of Technology, Bangalore South Campus
- Indra Priyadarshini, Student, P.E.S Institute of Technology, Bangalore South Campus
- Suhas, Student, P.A. College of Engineering, Mangalore
- Vamsikrishna G., Student, P.E.S Institute of Technology, Bangalore South Campus
- Vikram S., Student, P.E.S Institute of Technology, Bangalore South Campus
- Vivek Basavraj, Student, Jnana Vikas Institute of Technology, Bidadi
- Aruna S, Core member of FSMK
- HariPrasad, Core member of FSMK
- Isham, Core member of FSMK
- Jayakumar , Core member of FSMK
- Jeeva J, Core member of FSMK
- Jickson, Core member of FSMK
- Karthik Bhat, Core member of FSMK
- Prabodh C P, Core member of FSMK
- RaghuRam N, Core member of FSMK
- Rameez Thonnakkal, Core member of FSMK
- Sarath M S, Core member of FSMK
- Shijil TV, Core member of FSMK
- Vignesh Prabhu, Core member of FSMK
- Vijay Kulkarni, Core member of FSMK
- Yajnesh, Core member of FSMK

About Free Software Movement Karnataka (FSMK)

Free Software Movement Karnataka (FSMK) is a nonprofit organization formed in March 2009 to spread the ideals of free software. We try to increase the understanding of the philosophy behind free software and encourage its use in educational institutions, our very own community center and other organisations.

The phrase "free and open source software" can be used to collectively describe a set of operating systems and standalone applications that are free from the clutches of large corporate organisations which produce software only for commercial purposes. Free software is often freely available and is created by a thriving community of programmers, designers and writers. The source code (i.e. the computer program) that underlies an application or an operating system is also open to the perusal of the common individual, which allows every curious tinkerer, student or otherwise, to play around with the source code. Most free software organisations welcome the general public to be part of their community and to contribute in any of the three spheres mentioned above. In the event of you not being a programmer, designer or writer, you can also become a member of the community by actively using free software applications, thereby reporting any issues that you notice, and also by spreading awareness about free software among your friends and family. At FSMK, we believe that it is unfortunate that schools, colleges and other small organisations that can use Linux and related free software for no cost, instead invest in using proprietary and commercial software, thereby spending large amounts annually on licensing and other fees, which can instead be used for the betterment of education or related services, and thereby, the betterment of society.

I want to be involved Website: <http://fsmk.org/>

Mailing list: <http://www.fsmk.org/?q=mailinglistssubscribe>

About Spoken Tutorial Project:

The Spoken Tutorial project is an initiative of National Mission on Education through ICT, Government of India, to promote IT literacy through Free and Open Source Software. The project is being developed and coordinated by IIT-Bombay and led by Dr. Kannan M. Moudgalya. The project aims at building a repository of self learning courses through video tutorials of various open source softwares. These courses are then used to organize 2 hour workshops in government organizations, NGOs, SMEs and School and Colleges in India completely free of cost for the participants. These tutorials are not only available in English but also in various regional languages for the learner to be able to learn in the language he/she is comfortable in. Currently, Spoken Tutorials are available for free software tools like Blender, GIMP, Latex, Scilab, LibreOffice Suite, Ubuntu Linux, Mozilla Firefox, Thunderbird, MySQL and also programming languages and scripts like C, C++, Python, Ruby, Perl, PHP, Java. All the spoken tutorials which are released under Creative Commons License are available for download free of cost at their website <http://spoken-tutorial.org>.

About Jnana Vikas Institute of Technology, Bidadi:

Jnana Vikas Institute of Technology was established in the year 2001 by JNANA VIKAS VIDYA SANGHA with a mission to not just provide a solid educational foundation to students but to build their careers, to make them eminent personalities in the society and to make the industry doors open to them. It is approved by AICTE, New Delhi and affiliated to VTU, Belgaum. It has a residential campus with nearly 73 faculties, 28 technical and non-technical supporting staff, 27 administrative and supporting staff and 590 students and is a self-contained campus located in a beautiful green land of about 25 acres. The institute has four academic departments in various disciplines of engineering and three departments in general science with nearly 19 laboratories all together, organized in a unique pattern. There is a separate department for management discipline. The campus is located at Bidadi, in southern part of city of Bengaluru. More information about the college is provided at their website, <http://www.jvitedu.in/>

MySQL User guide

1. How to install Mysql?

o With internet connection:

1. Using terminal:

To install MySQL, run the following command from a terminal prompt:

sudo apt-get install mysql-server

During the installation process you will be prompted to enter a password for the MySQL root user. Once the installation is complete, the MySQL server should be started automatically. You can run the following command from a terminal prompt to check whether the MySQL server is running:

sudo netstat -tap | grep mysql

When you run this command, you should see the following line or something similar:

```
tcp      0      0 localhost:mysql      *:~      LISTEN 2556/mysql
```

If the server is not running correctly, you can type the following command to start it:

sudo /etc/init.d/mysql restart

1. Using Ubuntu software Center:

Open ubuntu software centre, type Mysql Server in search, Click Install.

■ Offline installation:

1. Using terminal:

a. Download or acquire the deb packages of mysql.

<http://dev.mysql.com/downloads/mysql/5.6.html> <http://packages.ubuntu.com/quantal/multiarch-support>

b. Copy it in a empty folder.

c. Go to to the empty folder path through terminal and type

Sudo dpkg -i .deb*

2. Using Ubuntu software Center:

a. Right click the deb package and click open through ubuntu software centre.

1. To get Mysql Prompt in Terminal

Type the following command to get MySQL prompt :

```
mysql -u root -p
```

Enter the Mysql password

```
root123
```

NOTE: mysql refers to command

-u root -p refers to user root password

```
>>login to MySQL as root user(u) with mysql password(p).
```

1. Creation of Database and query execution.

1. You will get the MySQL prompt as shown below:

```

manu@manu-HP-Pavilion-g6-Notebook-PC:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.5.28-0ubuntu0.12.04.3 (Ubuntu)

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 

```

2. Database Object Creation:

- To create Database

```
Create Database Student;
```

- To Use Database

```
use Student;
```

- To view the databases

```
show databases;
```

```

mysql> create Database Student;
Query OK, 1 row affected (0.00 sec)

mysql> use Student;
Database changed
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| STUDENT |
| Student |
| bloodbank |
| contact |
| dbms_workshop |
| first |
| glug |
| glug12 |
| mysql |
| performance_schema |
| phpmyadmin |
| student_prog |
| weblab |
+-----+
14 rows in set (0.00 sec)

mysql> 

```

3. Table Creation:

```

create table student(snum int(4) primary key,sname varchar(10),major varchar(10),lev varchar(2),age int(2));
create table faculty(fid int(4) primary key,fname varchar(10), deptid int(2));
create table class(cname varchar(10) primary key,meetat varchar(10),room varchar(10));
create table enrolled(snum int(4) references student(snum),cname varchar(10));

```

```

mysql> create table student(snum int(4) primary key,sname varchar(10),major varchar(10),lev varchar(2),age int(2));
Query OK, 0 rows affected (0.42 sec)

mysql> create table faculty(fid int(4) primary key,fname varchar(10), deptid int(2));
Query OK, 0 rows affected (0.11 sec)

mysql>

```

4. Inserting values into Table:

```

insert into student values(121,'Agrawal','CSE','SR',21);
insert into student values(119, 'Tony ', 'CSE', 'SR',21);
insert into student values(122, ' Krishna ', 'CSE', 'JR',20);

```

![Alt text](DBScreenShots/4.png)

5. to execute Query;

To find the author1 of the book which has maximum sales:

```

select author1_name from author1 a,catalogue1 c where a.author1_id=c.author1_id and c.sales>=all(c.sales);

```

![Alt text](DBScreenShots/5.png)

6. To get the tables information use DESC command.

![Alt text](DBScreenShots/6.png)

7. Contents of the table can be viewed by using the command SELECT as follows:

![Alt text](DBScreenShots/7.png)

8. In between if user wants to clear the screen of the terminal then press:

ctrl+l.

9. To get any information about the commands used in the mysql use "help" command.

Eg: help insert;

help create;

10. Type exit to exit from the MySQL prompt.

Note 1 : MYSQL is a a case sensitive.

Ex: "desc MEMBERS" is differ from "desc members".

![Alt text](DBScreenShots/8.png)

Note 2:

Handy MySQL Commands	
Description	Command
To login (from unix shell) use -h only if needed.	[mysql dir]/bin/mysql -h hostname -u root -p
Create a database on the sql server.	create database [databasename];
List all databases on	

the sql server.	show databases;
Switch to a database.	use [db name];use [db name];use [db name];
To see all the tables in the db.	show tables;
To see database's field formats.	describe [table name];
To delete a db.	drop database [database name];
To delete a table.	drop table [table name];
Show all data in a table.	SELECT FROM [table name];
Returns the columns and column information pertaining to the designated table.	show columns from [table name];
Show certain selected rows with the value "whatever".	SELECT FROM [table name] WHERE [field name] = "whatever";
Show all records containing the name "Bob" AND the phone number '3444444'.	SELECT FROM [table name] WHERE name = "Bob" AND phonenumber = '3444444';
Show all records not containing the name "Bob" AND the phone number '3444444' order by the phonenumber field.	SELECT FROM [table name] WHERE name != "Bob" AND phonenumber = '3444444' order by phonenumber;
Show all records starting with the letters 'bob' AND the phone number '3444444'.	SELECT FROM [table name] WHERE name like "Bob%" AND phonenumber = '3444444';
Use a regular expression to find records. Use "REGEXP BINARY" to force case-sensitivity. This finds any record beginning with a.	SELECT FROM [table name] WHERE rec RLIKE "^a\$";
Show unique records.	SELECT DISTINCT [column name] FROM [table name];
Show selected records sorted in an ascending (asc) or descending (desc).	SELECT [col1],[col2] FROM [table name] ORDER BY [col2] DESC;
Count rows.	SELECT COUNT(*) FROM [table name];
Join tables on common columns.	select lookup.illustrationid, lookup.personid,person.birthday from lookup left join person on lookup.personid=person.personid=statement to join birthday in person table with primary illustration id;
Switch to the mysql db. Create a new user.	INSERT INTO [table name] (Host,User>Password) VALUES ('%','user',PASSWORD('password'));
Change a users password.(from unix shell).	[mysql dir]/bin/mysqladmin -u root -h hostname.blah.org -p password 'new-password'
Change a users password.(from MySQL prompt).	SET PASSWORD FOR 'user'@'hostname' = PASSWORD('passwordhere');
Switch to mysql db.Give user privileges for a db.	INSERT INTO [table name] (Host,Db,User,Selectpriv,Insertpriv,Updatepriv,Deletepriv,Createpriv,Droppriv) VALUES ('%','db','user','Y','Y','Y','Y','Y','N');
To update info already in a table.	UPDATE [table name] SET Selectpriv = 'Y',Insertpriv = 'Y',Updatepriv = 'Y' where [field name] = 'user';
Delete a row(s) from a table.	DELETE from [table name] where [field name] = 'whatever';

Update database permissions/privileges.	FLUSH PRIVILEGES;
Delete a column.	alter table [table name] drop column [column name];
Add a new column to db.	alter table [table name] add column [new column name] varchar (20);
Change column name.	alter table [table name] change [old column name] [new column name] varchar (50);
Make a unique column so you get no dupes.	alter table [table name] add unique ([column name]);
Make a column bigger.	alter table [table name] modify [column name] VARCHAR(3);
Delete unique from table.	alter table [table name] drop index [column name];
Load a CSV file into a table.	LOAD DATA INFILE '/tmp/filename.csv' replace INTO TABLE [table name] FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' (field1,field2,field3);
Dump all databases for backup. Backup file is sql commands to recreate all db's.	[mysql dir]/bin/mysqldump -u root -ppassword --opt >/tmp/alldatabases.sql
Dump one database for backup.	[mysql dir]/bin/mysqldump -u username -ppassword --databases databasename >/tmp/databasename.sql
Dump a table from a database.	[mysql dir]/bin/mysqldump -c -u username -ppassword databasename tablename > /tmp/databasename.tablename.sql
Restore database (or database table) from backup.	[mysql dir]/bin/mysql -u username -ppassword databasename < /tmp/databasename.sql
Create Table Example 1.	CREATE TABLE [table name] (firstname VARCHAR(20), middleinitial VARCHAR(3), lastname VARCHAR(35),suffix VARCHAR(3), officeid VARCHAR(10),userid VARCHAR(15),username VARCHAR(8),email VARCHAR(35),phone VARCHAR(25), groups VARCHAR(15),datestamp DATE,timestamp time,pgpemail VARCHAR(255));
Create Table Example 2.	create table [table name] (personid int(50) not null auto_increment primary key,firstname varchar(35),middlename varchar(50),lastname varchar(50) default 'bato');

Advantages of MySQL:

1. MySQL is the widely used open source database. MySQL is the backend database of most of the websites.
2. As a Free Software(Free as in freedom), MySQL can be downloaded and used by the developer for free.
3. MySQL is robust and it provides excellent performance due to usage of MyISAM.
4. MySQL occupies very less disk space.
5. MySQL can be easily installed in all major operating systems like Microsoft Windows, Linux, UNIX.
6. MySQL can be easily learnt using the tutorials that are available on internet. We would recommend users to go through the Spoken Tutorial videos given below to get more information on MySQL.
7. MySQL is best suited for small and medium applications.

Resources

- Please go through the video tutorials on MySQL developed and released by **Spoken Tutorial Project**, an initiative of National Mission on Education through ICT, Government of India, to promote IT literacy through Open Source Software. Students can go through these video tutorials to get better understanding of the subject. The tutorials for MySQL can be downloaded from [here](#). More info about the project can be found [here](#).
- The MySQL Handbook is also available in the mysql-doc-5.0 package. To install the package enter the following in a terminal: **sudo apt-get install mysql-doc-5.0**
- The documentation is in HTML format, to view them enter file:///usr/share/doc/mysql-doc-5.0/refman-5.0-en.html-chapter/index.html in your browser's address bar.

Lab Programs list for Database Applications Lab as specified by VTU for 5th Semester students:

1. Consider the following relations:

Student (snum: integer, sname: string, major: string, level: string, age: integer)

Class (name: string, meets at: string, room: string, d: integer)

Enrolled (snum: integer, cname: string)

Faculty (fid: integer, fname: string, deptid: integer)

The meaning of these relations is straight forward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc) Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
2. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
3. Find the names of all students who are enrolled in two classes that meet at the same time.
4. Find the names of faculty members who teach in every room in which some class is taught.
5. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

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

Flights (no: integer, from: string, to: 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.

Write each of the following queries in SQL.

1. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
2. 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.
3. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
4. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the average salary of all pilots certified for this aircraft.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

3. 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)

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Enter at least five tuples for each relation.
3. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
4. 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.
5. List any department that has all its adopted books published by a specific publisher.
6. Generate suitable reports.
7. Create suitable front end for querying and displaying the results.

4. 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)

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Enter at least five tuples for each relation.
3. 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.
4. Find the author of the book which has maximum sales.
5. Demonstrate how you increase the price of books published by a specific publisher by 10%.
6. Generate suitable reports.
7. Create suitable front end for querying and displaying the results.

5. Consider the following database for a banking enterprise

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

ACCOUNT(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)

1. Create the above tables by properly specifying the primary keys and the foreign keys
2. Enter at least five tuples for each relation
3. Find all the customers who have at least two accounts at the Main branch.
4. Find all the customers who have an account at all the branches located in a specific city.
5. Demonstrate how you delete all account tuples at every branch located in a specific city.
6. Generate suitable reports.
7. Create suitable front end for querying and displaying the results.

PROGRAM 1

DESCRIPTION:

The following relations keep track of students, their enrollment for classes along with faculty information.

- Student (snum: integer, sname: string, major: string, level: string, age: integer)
- Class (name: string, meets at: string, room: string, d: integer)
- Enrolled (snum: integer, cname: string)
- Faculty (fid: integer, fname: string, deptid: integer)

NOTE: The meaning of these relations is straight forward. For example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc)

Queries:

Write the following queries in SQL. No duplicates should be printed in any of the answers.

1. Find the names of all juniors (level=Jr) who are enrolled for class taught by professor Harshith.
2. Find the names of all classes that either meet in room128 or have 5 or more students enrolled.
3. Find the names of all students who are enrolled in two classes that meet at same time.
4. Find the names of faculty members who teach in every room in which some class is taught.
5. Find the names of the faculty members for whom the combined enrollment of the classes that they teach is less than five.

Create:

```
mysql> create database student;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> use flights;
Database changed
```

```
mysql> create table student(
    snum int, sname varchar(10),
    major varchar(2),
    level varchar(2),
    age int, primary key(snum));
Query OK, 0 rows affected (0.10 sec)
```

```
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
snum	int(11)	NO	PRI	0	
sname	varchar(10)	YES		NULL	
major	varchar(2)	YES		NULL	
level	varchar(2)	YES		NULL	
age	int(11)	YES		NULL	

5 rows in set (0.01 sec)

```
mysql> create table faculty(
    fid int, fname varchar(20),
    deptid int,
    primary key(fid));
Query OK, 0 rows affected (0.08 sec)
mysql> desc faculty;
```

Field	Type	Null	Key	Default	Extra
fid	int(11)	NO	PRI	0	
fname	varchar(20)	YES		NULL	
deptid	int(11)	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> create table class(
  cname varchar(20),
  metts_at varchar(10),
  room varchar(10),
  fid int,
  primary key(cname),
  foreign key(fid) references faculty(fid));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> desc class;
```

Field	Type	Null	Key	Default	Extra
cname	varchar(20)	NO	PRI		
meets_at	varchar(10)	YES		NULL	
room	varchar(10)	YES		NULL	
fid	int(11)	YES	MUL	NULL	

4 rows in set (0.01 sec)

```
mysql> create table enrolled(
  snum int,
  cname varchar(20),
  primary key(snum,cname),
  foreign key(snum) references student(snum),
  foreign key(cname) references class(cname));
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> desc enrolled;
```

Field	Type	Null	Key	Default	Extra
snum	int(11)	NO	PRI	0	
cname	varchar(20)	NO	PRI		

2 rows in set (0.00 sec)

INSERTIONS:

```
mysql> insert into student values
-> (1, 'jhon', 'CS', 'Sr', 19),
-> (2, 'smith', 'CS', 'Jr', 20),
-> (3, 'jacob', 'CV', 'Sr', 20),
-> (4, 'tom', 'CS', 'Jr', 20),
-> (5, 'sid', 'CS', 'Jr', 20),
-> (6, 'harry', 'CS', 'Sr', 21)
Query OK, 2 rows affected (0.04 sec)
```

```
mysql> select *from student;
```

snum	sname	major	level	age
1	jhon	CS	Sr	19
2	smith	CS	Jr	20
3	jacob	CV	Sr	20
4	tom	CS	Jr	20
5	sid	CS	Jr	20
6	Harry	CS	Sr	21

6 rows in set (0.00 sec)

```
mysql> insert into faculty values
->(11, 'Harshith', 1000),
->(12, 'Mohan', 1000),
->(13, 'Kumar', 1001),
->(14, 'Shobha', 1002),
->(15, 'Shan', 1000);
```

Query OK, 1 row affected (0.03 sec)

mysql> select *from faculty;

fid	fname	deptid
11	Harshith	1000
12	Mohan	1000
13	Kumar	1001
14	Shobha	1002
15	Shan	1000

5 rows in set (0.00 sec)

mysql> insert into class values

-> ('class1','noon','room1',14),
-> ('class10','morning','room128',14),
-> ('class2','morning','room2',12),
-> ('class3','morning','room3',11),
-> ('class4','evening','room4',14),
-> ('class5','night','room3',15),
-> ('class6','morning','room2',14),
-> ('class7','morning','room3',14);

Query OK, 8 rows affected (0.05 sec)

Records: 8 Duplicates: 0 Warnings: 0

cname	meets_at	room	fid
class1	noon	room1	14
class10	morning	room128	14
class2	morning	room2	12
class3	morning	room3	11
class4	evening	room4	14
class5	night	room3	15
class6	morning	room2	14
class7	morning	room3	14

8 rows in set (0.00 sec)

mysql> insert into enrolled values

-> (1,'class1'),
-> (2,'class1'),
-> (3,'class3'),
-> (4,'class3'),
-> (3,'class3'),
-> (5,'class4');
-> (1,'class5');
-> (2,'class5');
-> (3,'class5');
-> (4,'class5');
-> (5,'class5');
-> (6,'class5');

Query OK, 12 rows affected (0.03 sec)

mysql> select *from enrolled;

snum	cname
1	class1
2	class1
3	class2
3	class3
4	class3
5	class4
1	class5

```
|      2 | class5 |
|      3 | class5 |
|      4 | class5 |
|      5 | class5 |
|      6 | class5 |
+-----+-----+
12 rows in set (0.00 sec)
```

QUERIES:

Query 1: Find the names of all juniors (level=Jr) who are enrolled for class taught by professor Harshith.

```
mysql> select distinct s.sname
      from student s,class c,faculty f,enrolled e
      where  s.snum=e.snum      and
             e.cname=c.cname    and
             s.level='jr'       and
             f.fname='Harshith' and
             f.fid=c.fid;
+-----+
| sname |
+-----+
| tom   |
+-----+
1 row in set (0.00 sec)
```

Description : Query checks whether the students are enrolled for the class and the level

Query 2: Find the names of all classes that either meet in room128 or have 5 or more students enrolled.

```
mysql> select distinct cname
      from class
      where room='room128'
      or
      cname in (select e.cname from enrolled e group by e.cname having count(*)>=5);
+-----+
| cname |
+-----+
| class10 |
| class5 |
+-----+
2 rows in set (0.00 sec)
```

Description : Query results displays the class names that either have room number as room

Query 3: Find the names of all students who are enrolled in two classes that meet at same time.

```
mysql> select distinct s.sname
      from student s
      where s.snum in (select e1.snum
                       from enrolled e1,enrolled e2,class c1,class c2
                       where e1.snum=e2.snum      and
                             e1.cname<>e2.cname    and
                             e1.cname=c1.cname      and
                             e2.cname=c2.cname      and
                             c1.meets_at=c2.meets_at );
+-----+
| sname |
+-----+
| jacob |
+-----+
1 row in set (0.00 sec)
```

Description : Outer part of the query extraxts the name of the students from table student

Query 4: Find the names of faculty members who teach in every room in which some class is taught.


```
mysql> select f.fname,f.fid from faculty f
       where f.fid in ( select fid from class
                       group by fid having count(*)=(select count(distinct room) from class) );
```

```
+-----+-----+
| fname | fid |
+-----+-----+
| Shobha | 14 |
+-----+-----+
1 row in set (0.00 sec)
```

Description : The outer part of the query fetches the name and id of the faculty from table faculty.

Query 5: Find the names of the faculty members for whom the combined enrollment of the classes that they teach is less than five.

```
mysql> select distinct f.fname
       from faculty f
       where f.fid in ( select c.fid
                       from class c, enrolled e
                       where c.cname = e.cname group by c.cname having count(c.cname)< 5 );
```

```
+-----+
| fname |
+-----+
| Harshith |
| Mohan |
| Shobha |
+-----+
3 rows in set (0.01 sec)
```

Description : The outer query fetches the name of the faculty for fid obtained from the inner query.

PROGRAM 2

DESCRIPTION:

The following relations keep track of airline flight information:

- FLIGHTS (no:integer,from:string,to: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.

Queries:

Write each of the following queries in SQL.

1. Find the names of aircraft such that all pilots certified to operate them have salaries less than 10000.
2. For each pilot who is certified for more than three aircrafts, find the eid and the max salary of the pilots.
3. Find the names of all pilots whose salary is less than the price of the cheapest route from Bangalore to Delhi.
4. For all aircrafts with cruisingrange over 1000 kms, find the name of the aircraft and the max salary of the pilots.
5. Find the names of pilots certified for some Boeing aircraft.
6. Find the aid's of all aircraft that can be used on routes from Bangalore to Delhi.

Create:

```
mysql> create database flights;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> use flights;
Database changed
mysql> create table flight(
    -> no int,
    -> frm varchar(20),
    -> too varchar(20),
    -> distance int,
    -> departs varchar(20),
    -> arrives varchar(20),
    -> price real,
    -> primary key (no) );
Query OK, 0 rows affected (0.17 sec)
```

```
mysql> desc flight;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| no    | int(11)       | NO   | PRI | 0        |       |
| frm   | varchar(20)   | YES  |     | NULL     |       |
| too   | varchar(20)   | YES  |     | NULL     |       |
| distance | int(11)      | YES  |     | NULL     |       |
| departs | varchar(20)  | YES  |     | NULL     |       |
| arrives | varchar(20)  | YES  |     | NULL     |       |
| price | double        | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> create table aircraft(
    -> aid int,
    -> aname varchar(20),
    -> cruisingrange int,
    -> primary key (aid) );
Query OK, 0 rows affected (0.19 sec)
```

```
mysql> desc aircraft;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| aid        | int(11)   | NO   | PRI | 0        |       |
| aname      | varchar(20) | YES  |     | NULL     |       |
| cruisingrange | int(11)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

```
mysql> create table employees(
-> eid int,
-> ename varchar(20),
-> salary int,
-> primary key (eid) );
Query OK, 0 rows affected (0.29 sec)
```

```
mysql> desc employees;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eid        | int(11)   | NO   | PRI | 0        |       |
| ename      | varchar(20) | YES  |     | NULL     |       |
| salary     | int(11)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> create table certified(
-> eid int,
-> aid int,
-> primary key (eid,aid),
-> foreign key (eid) references employees (eid),
-> foreign key (aid) references aircraft (aid) );
Query OK, 0 rows affected (0.43 sec)
```

```
mysql> desc certified;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| eid        | int(11)   | NO   | PRI | 0        |       |
| aid        | int(11)   | NO   | PRI | 0        |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Insertion:

```
mysql> insert into flight values (1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000);
Query OK, 7 rows affected (0.06 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
mysql> select * from flight;
+-----+-----+-----+-----+-----+-----+-----+
| no | frm      | too      | distance | departs | arrives | price |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bangalore | Mangalore | 360      | 10:45:00 | 12:00:00 | 10000 |
| 2 | Bangalore | Delhi    | 5000     | 12:15:00 | 04:30:00 | 25000 |
| 3 | Bangalore | Mumbai   | 3500     | 02:15:00 | 05:25:00 | 30000 |
| 4 | Delhi     | Mumbai   | 4500     | 10:15:00 | 12:05:00 | 35000 |
| 5 | Delhi     | Frankfurt | 18000    | 07:15:00 | 05:30:00 | 90000 |
| 6 | Bangalore | Frankfurt | 19500    | 10:00:00 | 07:45:00 | 95000 |
| 7 | Bangalore | Frankfurt | 17000    | 12:00:00 | 06:30:00 | 99000 |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> insert into aircraft values (123,'Airbus',1000),(302,'Boeing',5000),(306,'Jet01',5000);
Query OK, 7 rows affected (0.07 sec)
Records: 7 Duplicates: 0 Warnings: 0

mysql> select * from aircraft;
+-----+-----+-----+
| aid | aname      | cruisingrange |
+-----+-----+-----+
| 123 | Airbus     | 1000           |
| 302 | Boeing     | 5000           |
| 306 | Jet01      | 5000           |
| 378 | Airbus380  | 8000           |
| 456 | Aircraft   | 500            |
| 789 | Aircraft02 | 800            |
| 951 | Aircraft03 | 1000           |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> insert into employees values(1,'Ajay',30000),(2,'Ajith',85000),(3,'Arnab',50000),(4,'Harry',45000),(5,'Ron',90000),(6,'Josh',75000),(7,'Ram',100000);
Query OK, 7 rows affected (0.29 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

```
mysql> select * from employees;
+-----+-----+-----+
| eid | ename | salary |
+-----+-----+-----+
| 1   | Ajay  | 30000  |
| 2   | Ajith | 85000  |
| 3   | Arnab | 50000  |
| 4   | Harry | 45000  |
| 5   | Ron   | 90000  |
| 6   | Josh  | 75000  |
| 7   | Ram   | 100000 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> insert into certified values (1,123),(2,123),(1,302),(5,302),(7,302),(1,306),(2,306),(1,378),(2,378),(4,378),(3,456),(6,456),(1,789),(5,789),(6,789),(1,951),(3,951);
Query OK, 17 rows affected (0.30 sec)
Records: 17 Duplicates: 0 Warnings: 0
```

```
mysql> select * from certified;
+-----+-----+
| eid | aid |
+-----+-----+
| 1   | 123 |
| 2   | 123 |
| 1   | 302 |
| 5   | 302 |
| 7   | 302 |
| 1   | 306 |
| 2   | 306 |
| 1   | 378 |
| 2   | 378 |
| 4   | 378 |
| 3   | 456 |
| 6   | 456 |
| 1   | 789 |
| 5   | 789 |
| 6   | 789 |
| 1   | 951 |
| 3   | 951 |
+-----+-----+
17 rows in set (0.00 sec)
```

Queries:

1.Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs

80,000.

```
mysql> select distinct a.aname
-> from aircraft a,certified c,employees e
-> where a.aid=c.aid
-> and c.eid=e.eid
-> and not exists
-> (select *
-> from employees e1
-> where e1.eid=e.eid
-> and e1.salary<80000);
+-----+
| aname |
+-----+
| Airbus |
| Boeing |
| Jet01 |
| Airbus380 |
| Aircraft02 |
+-----+
5 rows in set (0.00 sec)
```

2.For each pilot who is certified for more than three aircrafts,find the eid and the maximum cruisingrange of the aircraft for which he/she is certified.

```
mysql> select c.eid,max(cruisingrange)
-> from certified c,aircraft a
-> where c.aid=a.aid
-> group by c.eid
-> having count(*)>3;
+-----+-----+
| eid | max(cruisingrange) |
+-----+-----+
| 1 | 8000 |
+-----+-----+
1 row in set (0.00 sec)
```

3.Find the names of all pilots whose salary is less than the price of the cheapest route from Bangalore to Frankfurt.

```
mysql> select distinct e.ename
-> from employees e
-> where e.salary<
-> (select min(f.price)
-> from flight f
-> where f.frm='Bangalore'
-> and f.too='Frankfurt');
+-----+
| ename |
+-----+
| Ajay |
| Ajith |
| Arnab |
| Harry |
| Ron |
| Josh |
+-----+
6 rows in set (0.00 sec)
```

4.For all aircrafts with cruisingrange over 1000 kms,find the name of the aircraft and the average salary of all pilots certified for this aircraft.

```
mysql> select a.aid,a.aname,avg(e.salary)
-> from aircraft a,certified c,employees e
-> where a.aid=c.aid
-> and c.eid=e.eid
-> and a.cruisingrange>1000
-> group by a.aid,a.aname;
+-----+-----+-----+
| aid | aname      | avg(e.salary) |
+-----+-----+-----+
| 302 | Boeing     | 73333.3333    |
| 306 | Jet01      | 57500.0000    |
| 378 | Airbus380  | 53333.3333    |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

5.Find the names of pilots certified for some Boeing aircraft.

```
mysql> select distinct e.ename
-> from employees e,aircraft a,certified c
-> where e.eid=c.eid
-> and c.aid=a.aid
-> and a.aname='Boeing';
+-----+
| ename |
+-----+
| Ajay  |
| Ron   |
| Ram   |
+-----+
3 rows in set (0.00 sec)
```

6.Find the aid's of all aircraft that can be used on routes from Bangalore to Delhi.

```
mysql> select a.aid
-> from aircraft a
-> where a.cruisingrange>
-> (select min(f.distance)
-> from flight f
-> where f.frm='Bangalore'
-> and f.too='Delhi');
+-----+
| aid |
+-----+
| 378 |
+-----+
1 row in set (0.00 sec)
```

PROGRAM 3

DESCRIPTION:

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)

Queries:

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Enter at least five tuples for each relation.
3. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
4. 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.
5. List any department that has all its adopted books published by a specific publisher.
6. Generate suitable reports.
7. Create suitable front end for querying and displaying the results.

Create:

```
mysql> create database books;  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use books;  
Database changed  
mysql> create table student(  
    regno varchar(15),  
    name varchar(20),  
    major varchar(20),  
    bdate date,  
    primary key (regno) );  
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> desc student;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| regno | varchar(15)   | NO   | PRI |          |       |  
| name  | varchar(20)   | YES  |     | NULL     |       |  
| major | varchar(20)   | YES  |     | NULL     |       |  
| bdate | date          | YES  |     | NULL     |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.01 sec)
```

```
mysql> create table course(  
    courseno int,  
    cname varchar(20),  
    dept varchar(20),  
    primary key (courseno) );  
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> desc course;  
+-----+-----+-----+-----+-----+-----+  
| Field  | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| courseno | int(11)       | NO   | PRI | 0        |       |  
| cname   | varchar(20)   | YES  |     | NULL     |       |  
| dept    | varchar(20)   | YES  |     | NULL     |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

```
mysql> create table enroll(
    regno varchar(15),
    courseno int,
    sem int(3),
    marks int(4),
    primary key (regno,courseno),
    foreign key (regno) references student (regno),
    foreign key (courseno) references course (courseno) );
Query OK, 0 rows affected (0.19 sec)
```

```
mysql> desc enroll;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| regno      | varchar(15)   | NO   | PRI |          |       |
| courseno   | int(11)       | NO   | PRI | 0        |       |
| sem        | int(3)        | YES  |     | NULL     |       |
| marks      | int(4)        | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> create table text(
    book_isbn int(5),
    book_title varchar(20),
    publisher varchar(20),
    author varchar(20),
    primary key (book_isbn) );
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> desc text;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| book_isbn  | int(5)        | NO   | PRI | 0        |       |
| book_title | varchar(20)   | YES  |     | NULL     |       |
| publisher  | varchar(20)   | YES  |     | NULL     |       |
| author     | varchar(20)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> create table book_adoption(
    courseno int,
    sem int(3),
    book_isbn int(5),
    primary key (courseno,book_isbn),
    foreign key (courseno) references course (courseno),
    foreign key (book_isbn) references text(book_isbn) );
Query OK, 0 rows affected (0.17 sec)
```

```
mysql> desc book_adoption;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| courseno   | int(11)       | NO   | PRI | 0        |       |
| sem        | int(3)        | YES  |     | NULL     |       |
| book_isbn  | int(5)        | NO   | PRI | 0        |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Insertion:

```
mysql> insert into student values ('1pe11cs001','a','sr',19931230);
Query OK, 1 row affected (0.05 sec)
```



```
mysql> insert into student values ('1pe11cs002','b','sr','19930924'),('1pe11cs003','c','s
Query OK, 4 rows affected (0.07 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from student;
+-----+-----+-----+-----+
| regno      | name | major | bdate      |
+-----+-----+-----+-----+
| 1pe11cs001 | a    | sr    | 1993-12-30 |
| 1pe11cs002 | b    | sr    | 1993-09-24 |
| 1pe11cs003 | c    | sr    | 1993-11-27 |
| 1pe11cs004 | d    | sr    | 1993-04-13 |
| 1pe11cs005 | e    | jr    | 1994-08-24 |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> insert into course values (111,'OS','CSE'),(112,'EC','CSE'),(113,'SS','ISE'),(114,
Query OK, 5 rows affected (0.06 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from course;
+-----+-----+-----+
| courseno | cname | dept |
+-----+-----+-----+
| 111      | OS    | CSE  |
| 112      | EC    | CSE  |
| 113      | SS    | ISE  |
| 114      | DBMS  | CSE  |
| 115      | SIGNALS | ECE  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into text values (10,'DATABASE SYSTEMS','PEARSON','SCHIELD'),(900,'OPERATI
Query OK, 8 rows affected (0.06 sec)
Records: 8 Duplicates: 0 Warnings: 0
```

```
mysql> select * from text;
+-----+-----+-----+-----+
| book_isbn | book_title      | publisher | author |
+-----+-----+-----+-----+
| 10        | DATABASE SYSTEMS | PEARSON   | SCHIELD |
| 900       | OPERATING SYS    | PEARSON   | LELAND  |
| 901       | CIRCUITS         | HALL INDIA | BOB     |
| 902       | SYSTEM SOFTWARE  | PETERSON  | JACOB   |
| 903       | SCHEDULING       | PEARSON   | PATIL   |
| 904       | DATABASE SYSTEMS | PEARSON   | JACOB   |
| 905       | DATABASE MANAGER | PEARSON   | BOB     |
| 906       | SIGNALS          | HALL INDIA | SUMIT   |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> insert into enroll values ('1pe11cs001',115,3,100),('1pe11cs002',114,5,100),('1pe
Query OK, 5 rows affected (0.08 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from enroll;
+-----+-----+-----+-----+
| regno      | courseno | sem  | marks |
+-----+-----+-----+-----+
| 1pe11cs001 |      115 |    3 |    100 |
| 1pe11cs002 |      114 |    5 |    100 |
| 1pe11cs003 |      113 |    5 |    100 |
| 1pe11cs004 |      111 |    5 |    100 |
| 1pe11cs005 |      112 |    3 |    100 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into book_adoption values
(111,5,900),
(111,5,903),
(111,5,904),
(112,3,901),
(113,3,10),
(114,5,905),
(113,5,902),
(115,3,906);
Query OK, 8 rows affected (0.06 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

```
mysql> select * from book_adoption;
+-----+-----+-----+
| courseno | sem  | book_isbn |
+-----+-----+-----+
|      111 |    5 |      900 |
|      111 |    5 |      903 |
|      111 |    5 |      904 |
|      112 |    3 |      901 |
|      113 |    3 |        10 |
|      113 |    5 |      902 |
|      114 |    5 |      905 |
|      115 |    3 |      906 |
+-----+-----+-----+
8 rows in set (0.00 sec).
```

Queries:

4. 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.

```
mysql> select c.courseno,t.book_isbn,t.book_title
from course c,book_adoption ba,text t
where c.courseno=ba.courseno
and ba.book_isbn=t.book_isbn
and c.dept='CSE'
and 2<(
select count(book_isbn)
from book_adoption b
where c.courseno=b.courseno)
order by t.book_title;
+-----+-----+-----+
| courseno | book_isbn | book_title      |
+-----+-----+-----+
|      111 |      904 | DATABASE SYSTEMS |
|      111 |      900 | OPERATING SYS   |
|      111 |      903 | SCHEDULING      |
+-----+-----+-----+
3 rows in set (0.01 sec)
```

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

```
mysql> select distinct c.dept
      from course c
      where c.dept in
      ( select c.dept
        from course c,book_adoption b,text t
        where c.courseno=b.courseno
        and t.book_isbn=b.book_isbn
        and t.publisher='PEARSON')
      and c.dept not in
      (select c.dept
        from course c,book_adoption b,text t
        where c.courseno=b.courseno
        and t.book_isbn=b.book_isbn
        and t.publisher!='PEARSON');
+-----+
| dept |
+-----+
| CSE  |
+-----+
1 row in set (0.00 sec).
```


PROGRAM 4

DESCRIPTION:

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)

Queries:

Write each of the following queries in SQL.

1. Create the above tables by properly specifying the primary keys and the foreign keys.
2. Enter at least five tuples for each relation.

Create:

```
mysql> create database book_dealer;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> use book_dealer;
Database changed
mysql> create table author1 (
    author1_id int,
    author1_name varchar(20),
    author1_city varchar(20),
    author1_country varchar(20),
    primary key(author1_id));
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> desc author1;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| author1_id     | int(11)       | NO   | PRI | 0        |       |
| author1_name   | varchar(20)   | YES  |     | NULL     |       |
| author1_city   | varchar(20)   | YES  |     | NULL     |       |
| author1_country | varchar(20)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> create table publisher1 (
    publisher1_id int,
    publisher1_name varchar(20),
    publisher1_city varchar(20),
    publisher1_country varchar(20),
    primary key(publisher1_id));
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> desc publisher1;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| publisher1_id  | int(11)       | NO   | PRI | 0        |       |
| publisher1_name | varchar(20)   | YES  |     | NULL     |       |
| publisher1_city | varchar(20)   | YES  |     | NULL     |       |
| publisher1_country | varchar(20) | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> create table category1 (
    category_id int,
    description varchar(30),
    primary key(category_id) );
Query OK, 0 rows affected (0.14 sec)
```

```
mysql> desc category1;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| category_id | int(11)    | NO   | PRI | 0        |       |
| description | varchar(30) | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> create table catalogue1(
    book_id int,
    book_title varchar(30),
    author1_id int,
    publisher1_id int,
    category_id int,
    year int,
    price int,
    primary key(book_id),
    foreign key(author1_id) references author1(author1_id),
    foreign key(publisher1_id) references publisher1(publisher1_id),
    foreign key(category_id) references category1(category_id) );
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> desc catalogue1;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| book_id     | int(11)    | NO   | PRI | 0        |       |
| book_title  | varchar(30) | YES  |     | NULL     |       |
| author1_id  | int(11)    | YES  | MUL | NULL     |       |
| publisher1_id | int(11)    | YES  | MUL | NULL     |       |
| category_id | int(11)    | YES  | MUL | NULL     |       |
| year        | int(11)    | YES  |     | NULL     |       |
| price       | int(11)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> create table orderdetails1(
    order_id int,
    book_id int,
    quantity int,
    primary key(order_id),
    foreign key(book_id) references catalogue1(book_id));
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> desc orderdetails1;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| order_id   | int(11)    | NO   | PRI | 0        |       |
| book_id    | int(11)    | YES  | MUL | NULL     |       |
| quantity   | int(11)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

INSERTIONS:

```
mysql> insert into author1 values
    (1001, 'JK Rowling', 'London', 'England'),
    (1002, 'Chetan Bhagat', 'Mumbai', 'India'),
    (1003, 'John McCarthy', 'Chicago', 'USA'),
    (1004, 'Dan Brown', 'California', 'USA') ;
```

```
Query OK, 4 rows affected (0.08 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from author1;
+-----+-----+-----+-----+
| author1_id | author1_name | author1_city | author1_country |
+-----+-----+-----+-----+
|      1001 | JK Rowling   | London       | England         |
|      1002 | Chetan Bhagat | Mumbai       | India           |
|      1003 | John McCarthy | Chicago      | USA             |
|      1004 | Dan Brown    | California   | USA             |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> insert into publisher1 values
      (2001, 'Bloomsbury', 'London', 'England'),
      (2002, 'Scholastic', 'Washington', 'USA'),
      (2003, 'Pearson', 'London', 'England'),
      (2004, 'Rupa', 'Delhi', 'India') ;
```

Query OK, 4 rows affected (0.06 sec)
Records: 4 Duplicates: 0 Warnings: 0

```
mysql> select * from publisher1;
+-----+-----+-----+-----+
| publisher1_id | publisher1_name | publisher1_city | publisher1_country |
+-----+-----+-----+-----+
|      2001 | Bloomsbury      | London          | England            |
|      2002 | Scholastic      | Washington      | USA                |
|      2003 | Pearson         | London          | England            |
|      2004 | Rupa            | Delhi           | India              |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> insert into category1 values
      (3001, 'Fiction'),
      (3002, 'Non-Fiction'),
      (3003, 'thriller'),
      (3004, 'action'),
      (3005, 'fiction') ;
```

Query OK, 5 rows affected (0.04 sec)
Records: 5 Duplicates: 0 Warnings: 0

```
mysql> select * from category1;
+-----+-----+
| category_id | description |
+-----+-----+
|      3001 | Fiction     |
|      3002 | Non-Fiction |
|      3003 | thriller    |
|      3004 | action      |
|      3005 | fiction      |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into catalogue1 values
      (4001, 'HP and Goblet Of Fire', 1001, 2001, 3001, 2002, 600),
      (4002, 'HP and Order Of Phoenix', 1001, 2002, 3001, 2005, 650),
      (4003, 'Two States', 1002, 2004, 3001, 2009, 65),
      (4004, '3 Mistakes of my life', 1002, 2004, 3001, 2007, 55),
      (4005, 'Da Vinci Code', 1004, 2003, 3001, 2004, 450),
      (4006, 'Angels and Demons', 1004, 2003, 3001, 2003, 350),
      (4007, 'Artificial Intelligence', 1003, 2002, 3002, 1970, 500) ;
Query OK, 7 rows affected (0.36 sec)
Records: 7  Duplicates: 0  Warnings: 0
```

```
mysql> select * from catalogue1;
```

book_id	book_title	author1_id	publisher1_id	category_id	year
4001	HP and Goblet Of Fire	1001	2001	3001	2002
4002	HP and Order Of Phoenix	1001	2002	3001	2005
4003	Two States	1002	2004	3001	2009
4004	3 Mistakes of my life	1002	2004	3001	2007
4005	Da Vinci Code	1004	2003	3001	2004
4006	Angels and Demons	1004	2003	3001	2003
4007	Artificial Intelligence	1003	2002	3002	1970

7 rows in set (0.00 sec)

```
mysql> insert into orderdetails1 values
      (5001, 4001, 5),
      (5002, 4002, 7),
      (5003, 4003, 15),
      (5004, 4004, 11),
      (5005, 4005, 9),
      (5006, 4006, 8),
      (5007, 4007, 2),
      (5008, 4004, 3) ;
Query OK, 8 rows affected (0.47 sec)
Records: 8  Duplicates: 0  Warnings: 0
```

```
mysql> select * from orderdetails1;
```

order_id	book_id	quantity
5001	4001	5
5002	4002	7
5003	4003	15
5004	4004	11
5005	4005	9
5006	4006	8
5007	4007	2
5008	4004	3

8 rows in set (0.00 sec)

QUERIES:

3: 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


```
mysql> select * from author1
      where author1_id in
      (select author1_id from catalogue1 where
      year>2000 and price>
      (select avg(price) from catalogue1)
      group by author1_id having count(*)>1);
```

OR

```
mysql> select * from author1
      where author1_id in
      (select author1_id from catalogue1 where
      year>2000 and price>
      (select avg(price) from catalogue1)
      group by author1_id having count(*)>1);
```

Description:

(select avg(price) from catalogue1):-it select the average price of the books from catalogue1

```
+-----+-----+-----+-----+
| author1_id | author1_name | author1_city | author1_country |
+-----+-----+-----+-----+
|      1001 | JK Rowling   | London       | England         |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

4: Find the author1 of the book which has maximum sales.

```
mysql> select c.eid,max(cruisingrange)
      from certified c,aircraft a
      where c.aid=a.aid
      group by c.eid
      having count(*)>3;
+-----+-----+
| eid | max(cruisingrange) |
+-----+-----+
|    1 |                8000 |
+-----+-----+
mysql> select author1_name
      from author1 a,catalogue1 c
      where a.author1_id=c.author1_id
      and book_id in
      (select book_id from orderdetails1
      where quantity= (select max(quantity)
      from orderdetails1));
```

OR

```
mysql> SELECT a.author1_name FROM author1 a,catalogue1 c
      WHERE a.author1_id=c.author1_id AND
      c.book_id IN (SELECT book_id
      FROM orderdetails1
      GROUP BY book_id HAVING
      SUM(quantity)>=ALL(SELECT SUM(quantity)
      FROM orderdetails1 GROUP BY book_id));
```

Description:

(select max(quantity) from orderdetails1):-it selects the maximum quantity of books from orderdetails1

```
+-----+
| author1_name |
+-----+
| Chetan Bhagat |
+-----+
1 row in set (0.00 sec)-----+
1 row in set (0.00 sec)
```

5: Demonstrate how you increase the price of books published by a specific publisher1 by 10%.

```
mysql> update catalogue1 set price=1.1*price
      where publisher1_id in
      (select publisher1_id from publisher1 where
       publisher1_name='pearson');
Query OK, 2 rows affected (0.41 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

Description:

This query is used to update the price of the books by 10% which are published by a spec

```
mysql> select * from catalogue1;
```

book_id	book_title	author1_id	publisher1_id	category_id	year
4001	HP and Goblet Of Fire	1001	2001	3001	2002
4002	HP and Order Of Phoenix	1001	2002	3001	2005
4003	Two States	1002	2004	3001	2009
4004	3 Mistakes of my life	1002	2004	3001	2007
4005	Da Vinci Code	1004	2003	3001	2004
4006	Angels and Demons	1004	2003	3001	2003
4007	Artificial Intelligence	1003	2002	3002	1970

```
7 rows in set (0.00 sec)
```

PROGRAM 5

DESCRIPTION:

The following relations keep track of a banking enterprise.

- BRANCH(branch-name:string, branch-city:string, assets:real)
- ACCOUNT(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)

Queries:

Write each of the following queries in SQL.

1. Create the above tables by properly specifying the primary keys and the foreign keys
2. Enter at least five tuples for each relation.

Create:

```
CREATE TABLE BRANCH
( branch_name VARCHAR(15),
  branch_city VARCHAR(15),
  assets NUMBER(10,2),
  PRIMARY KEY(branch_name)
);

CREATE TABLE ACCOUNT
( accno INTEGER(8),
  branch_name VARCHAR(15),
  balance NUMBER(10,2),
  PRIMARY KEY(accno),
  FOREIGN KEY(branch_name) REFERENCES BRANCH(branch_name) ON DELETE CASCADE
);

CREATE TABLE CUSTOMER
( customer_name VARCHAR(15),
  customer_street VARCHAR(15),
  customer_city VARCHAR(15),
  PRIMARY KEY(customer_name)
);

CREATE TABLE LOAN
( loan_number INTEGER(8),
  branc_hname VARCHAR(15),
  amount NUMBER(10,2),
  PRIMARY KEY(loan_number),
  FOREIGN KEY(branch_name) REFERENCES BRANCH(branch_name)
);

CREATE TABLE DEPOSITOR
( customer_name VARCHAR(15),
  accno INTEGER,
  PRIMARY KEY(customer_name, accno),
  FOREIGN KEY(customer_name) REFERENCES CUSTOMER(customer_name),
  FOREIGN KEY(accno) REFERENCES ACCOUNT(accno)
);

CREATE TABLE BORROWER
( customer_name VARCHAR(15),
  loan_number INTEGER(8),
  PRIMARY KEY(customer_name, loan_number),
  FOREIGN KEY(customer_name) REFERENCES CUSTOMER(customer_name),
  FOREIGN KEY(loan_number) REFERENCES LOAN(loan_number)
);
```

INSERTIONS:

```
mysql> insert into branch values
->      ("b1","c1",10000),
->      ("b2","c2",20000),
->      ("b3","c3",30000),
->      ("b4","c4",40000),
->      ("b5","c5",50000);
Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings:0
```

```
mysql> select * from branch;
```

```
+-----+-----+-----+
| branch_name | branch_city | assets |
+-----+-----+-----+
| b1          | c1          | 10000  |
| b2          | c2          | 20000  |
| b3          | c3          | 30000  |
| b4          | c4          | 40000  |
| b5          | c5          | 50000  |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into account values
->      (12,"b1",3000),
->      (22,"b2",4000),
->      (32,"b3",5000),
->      (42,"b4",6000),
->      (52,"b5",7000);

Query OK, 5 rows affected (0.06 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> select * from account;
```

```
+-----+-----+-----+
| accno | branch_name | balance |
+-----+-----+-----+
| 12    | b1          | 3000    |
| 22    | b2          | 4000    |
| 32    | b3          | 5000    |
| 42    | b4          | 6000    |
| 52    | b5          | 7000    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into customer values
->      ("cust1","cstreet1","ccity1"),
->      ("cust2","cstreet2","ccity2"),
->      ("cust3","cstreet3","ccity3"),
->      ("cust4","cstreet4","ccity4"),
->      ("cust5","cstreet5","ccity5");
```

```
Query OK, 5 rows affected (0.07 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> select * from customer;
```

```
+-----+-----+-----+
| customer_name | customer_street | customer_city |
+-----+-----+-----+
| cust1         | cstreet1        | ccity1        |
| cust2         | cstreet2        | ccity2        |
| cust3         | cstreet3        | ccity3        |
| cust4         | cstreet4        | ccity4        |
| cust5         | cstreet5        | ccity5        |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> insert into depositor values
```

```
-> ("cust1",12),  
-> ("cust2",22),  
-> ("cust3",32),  
-> ("cust4",42),  
-> ("cust5",52);
```

```
Query OK, 5 rows affected (0.06 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from depositor;
```

```
+-----+-----+  
| customer_name | accno |  
+-----+-----+  
| cust1        | 12    |  
| cust2        | 22    |  
| cust3        | 32    |  
| cust4        | 42    |  
| cust5        | 52    |  
+-----+-----+  
5 rows in set (0.00 sec)
```

```
mysql> insert into values loan
```

```
-> (10,"b1",10000),  
-> (20,"b2",20000),  
-> (30,"b3",30000),  
-> (40,"b4",40000),  
-> (50,"b5",50000);
```

```
Query OK, 5 rows affected (0.06 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from loan;
```

```
+-----+-----+-----+  
| loan_number | branch_name | amount |  
+-----+-----+-----+  
| 10          | b1          | 10000  |  
| 20          | b2          | 20000  |  
| 30          | b3          | 30000  |  
| 40          | b4          | 40000  |  
| 50          | b5          | 50000  |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

```
mysql> insert into borrower values
```

```
-> ("cust1",10),  
-> ("cust2",20),  
-> ("cust3",30),  
-> ("cust4",40),  
-> ("cust5",50);
```

```
Query OK, 5 rows affected (0.05 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from borrower;
```

customer_name	loan_number
cust1	10
cust2	20
cust3	30
cust4	40
cust5	50

```
5 rows in set (0.00 sec)
```

QUERIES:

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

```
mysql> SELECT customer_name FROM depositor d,account a WHERE  
        d.accno=a.accno AND a.branch_name='Main'  
        GROUP BY d.customer_name HAVING COUNT(d.customer_name)>=2;
```

```
Empty set (0.00 sec)
```

Note: Here we are getting empty set because in our 'account' table there is no branch_name with value 'Main' and also there are no customer who has two accounts at the Main branch. So we have to either update the table or else add the proper tuples so that we can get the proper outputs.

updating can be done with the following commands.

```
mysql> update account set branch_name='Main' where branch_name="b1";  
mysql> update account set branch_name='Main' where branch_name="b2";  
mysql> update account set customer_name='cust1' where customer_name="cust2";
```

customer_name
cust1

```
1 row in set (0.00 sec)
```

Description: The query is selecting the customer's name such that the account number associated with name is in both the account table and depositor table and also the name of the branch in the account table is 'Main' and then the tuples are being grouped by customer name in the depositor table and also the customer name having count atleast equal to 2 are being selected.

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

```
mysql> SELECT d.customer_name FROM account a,branch b,depositor d WHERE  
        b.branch_name=a.branch_name AND a.accno=d.accno AND b.branch_city='c3'  
        GROUP BY d.customer_name HAVING COUNT(distinct b.branch_name)=(select count(branch
```

customer_name
cust3

```
1 row in set (0.00 sec)
```

Description: The query selects the customers from the the depositor table such that branch name is in both the branch table and also account table and the account number in the selected tuples is in both account table and in depositor table and also the name of the branch city is 'c3'. The selected tuples are grouped by the customer name of the depositor table whose count should be equal to the count of the branch name in the branch table with branch city 'c3'.

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

```
mysql> DELETE FROM account where branch_name IN(SELECT branch_name FROM branch WHERE bran
Query OK, 1 row affected (0.04 sec)

mysql>SELECT * FROM account;
```

accno	branch_name	balance
12	b1	3000
22	b2	4000
32	b3	5000
42	b4	6000

```
4 rows in set (0.00 sec)
```

Description: The inner query "SELECT branch_name FROM branch WHERE branch_city='c5' " selects the branch names from the branch table where the branch city is c5. The selected tuples are given as input to the outer query which deletes the tuples with the selected branch names.

Table of Contents

Introduction	1
Content	1
Foreword	1
Contributors	1
About	1
Introduction to MySQL	6
Syllabus	6
Problem 1: Student Class	13
Problem 2: Airline Flight	18
Problem 3: Student Course Books	23
Problem 4: Book dealer	28
Problem 5: Banking Enterprise	35