B.M.S. COLLEGE OF ENGINEERING, BANGALORE-19

(Autonomous College under VTU)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATABASE MANAGEMENT SYSTEM LABORATORY RECORD

NAME: Prithvi.J

USN: 1BM19CS122

PROGRAM: BACHELOR OF ENGINEERING

SEMESTER: IV

SESSION: APR-JUL 2021

COURSE CODE: 19CS4PCDBM

COURSE TITLE: DATABASE MANAGEMENT SYSTEM

CREDITS: 4

DBMS Lab List

Experiment #	Name of Experiment
1	Insurance Database
2	Banking Enterprise Database
3	Supplier Database
4	Student Faculty Database
5	Airline Flight Database
6	Order Processing Database
7	Book dealer Database
8	Student Enrolment Database
9	Movie Database
10	College Database

PROGRAM 1:

INSURANCE DATABASE

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

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

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

ACCIDENT (report-number: int, date: date, location: String)

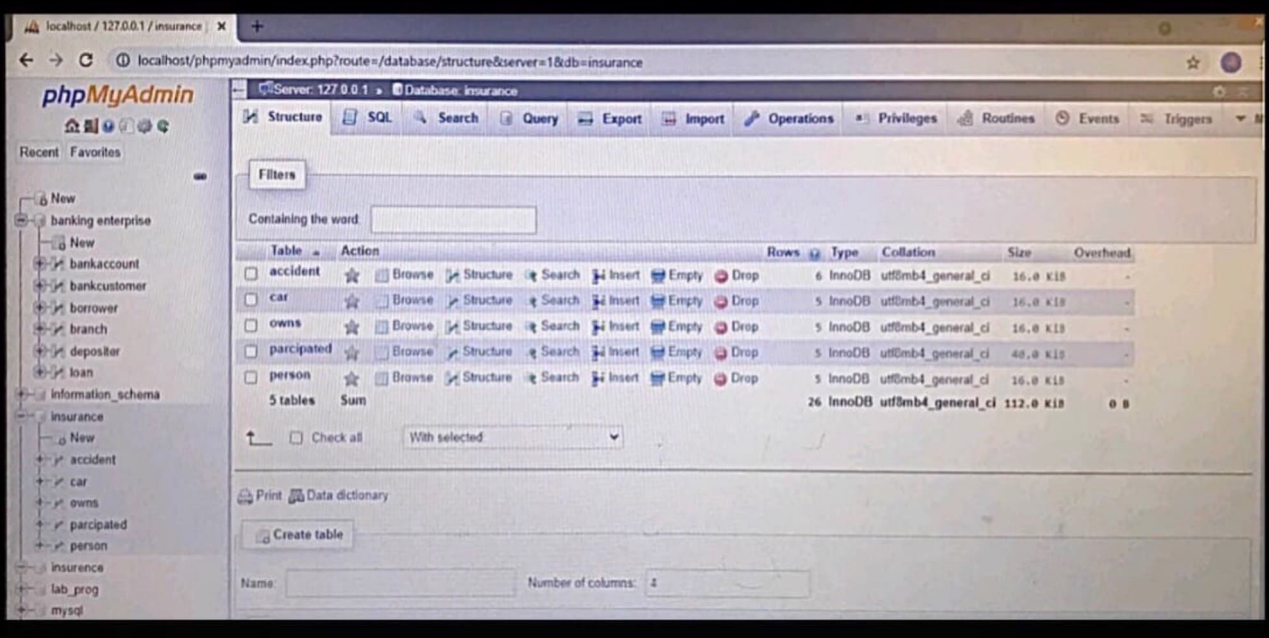
OWNS (driver-id #: String, Regno: String)

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

Queries Executed:

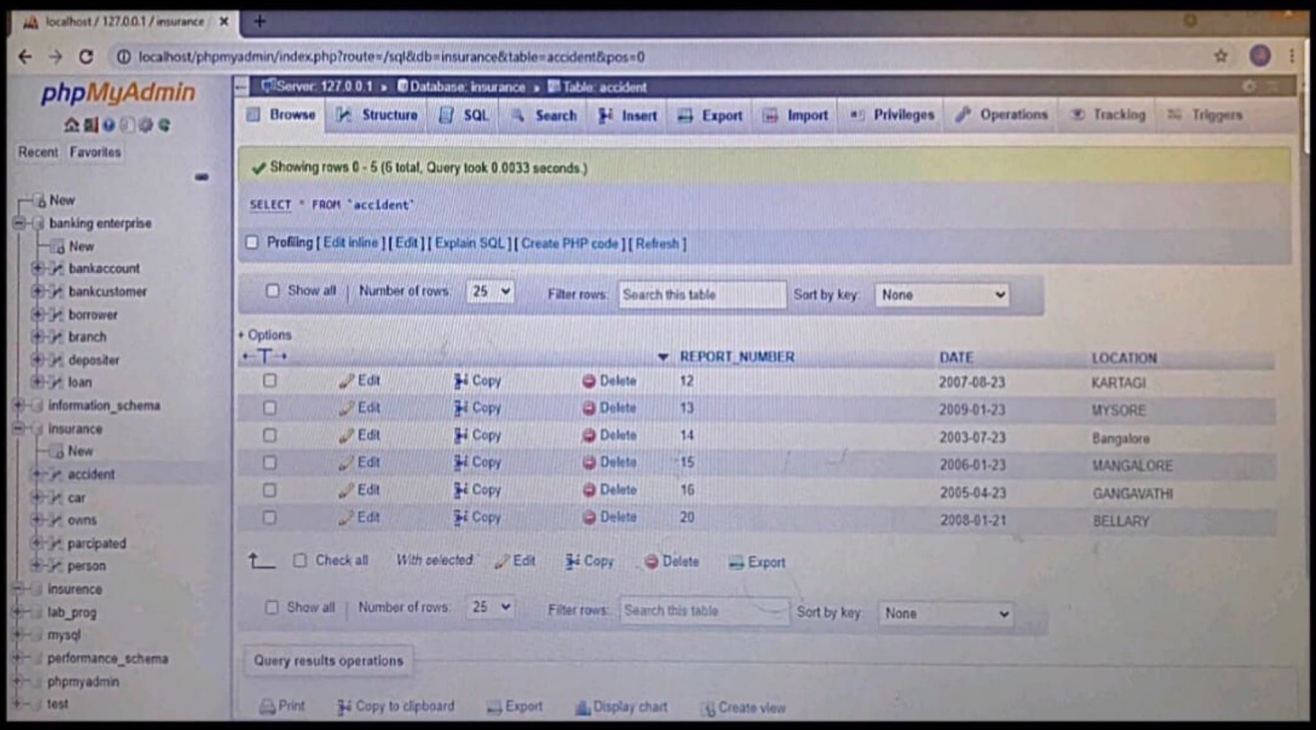
i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.

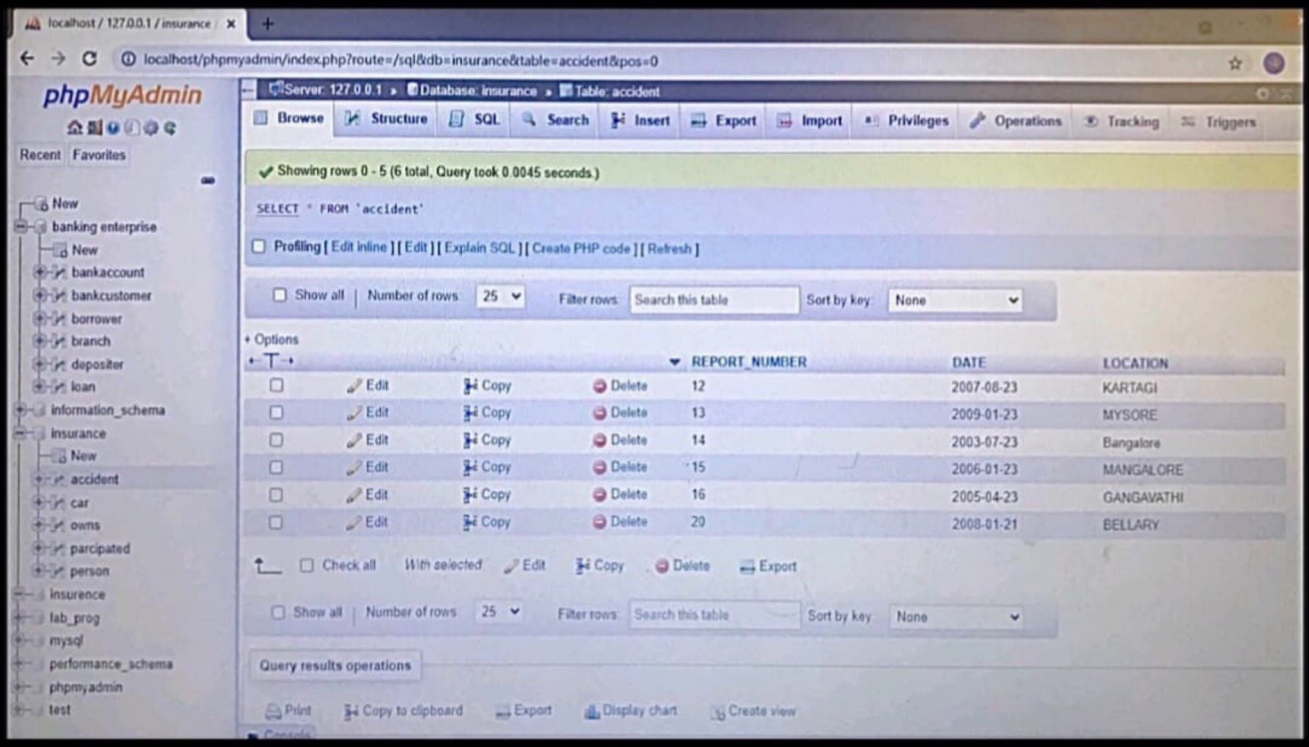


iii.Demonstrate how you

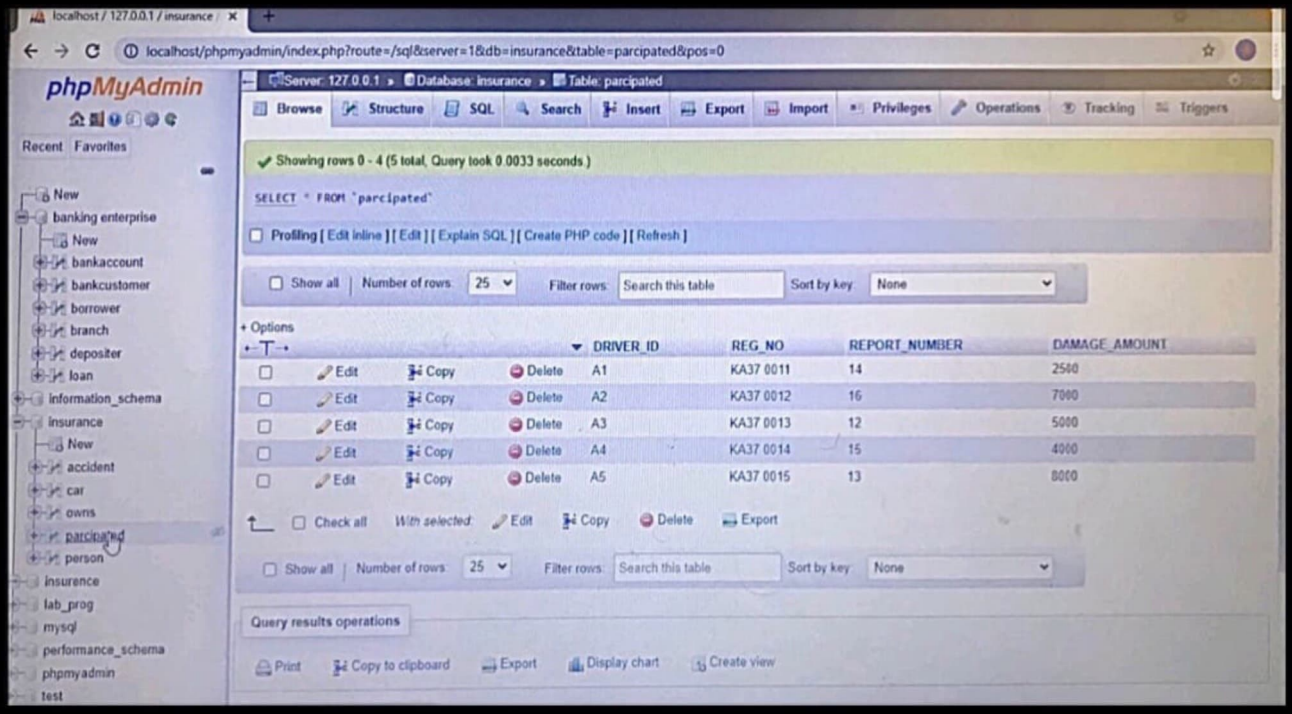
a.Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000



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.



PROGRAM 2. BANKING ENTERPRISE DATABASE

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)

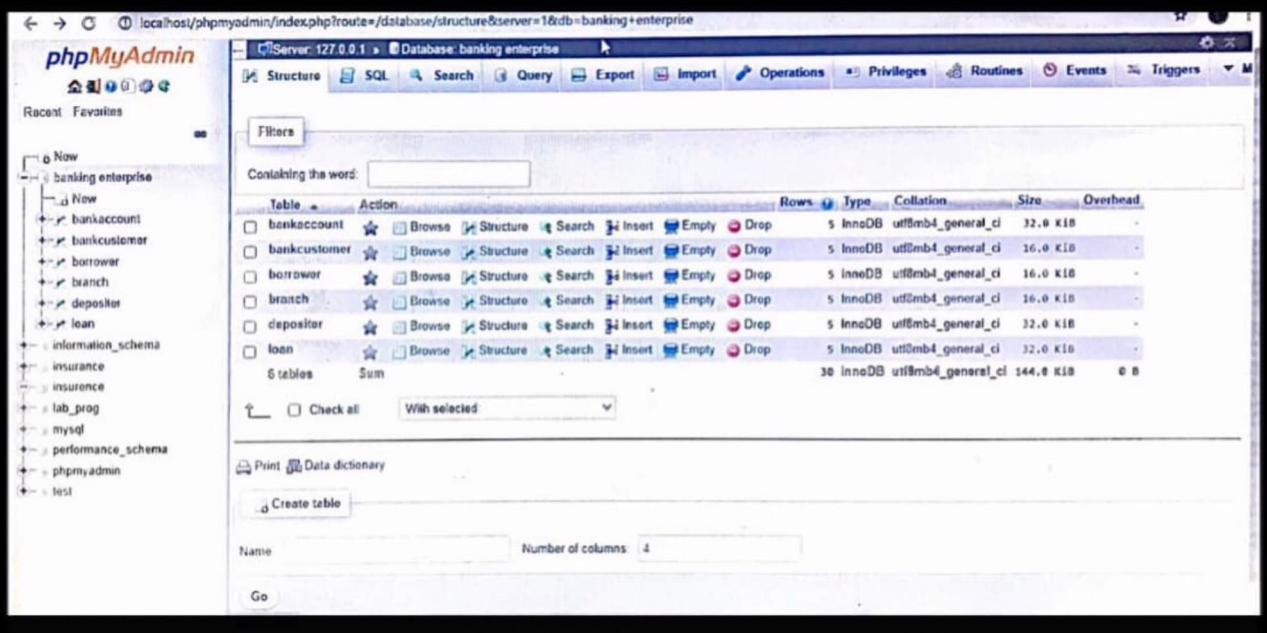
LOAN (loan-number: int, branch-name: String, amount: real)

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

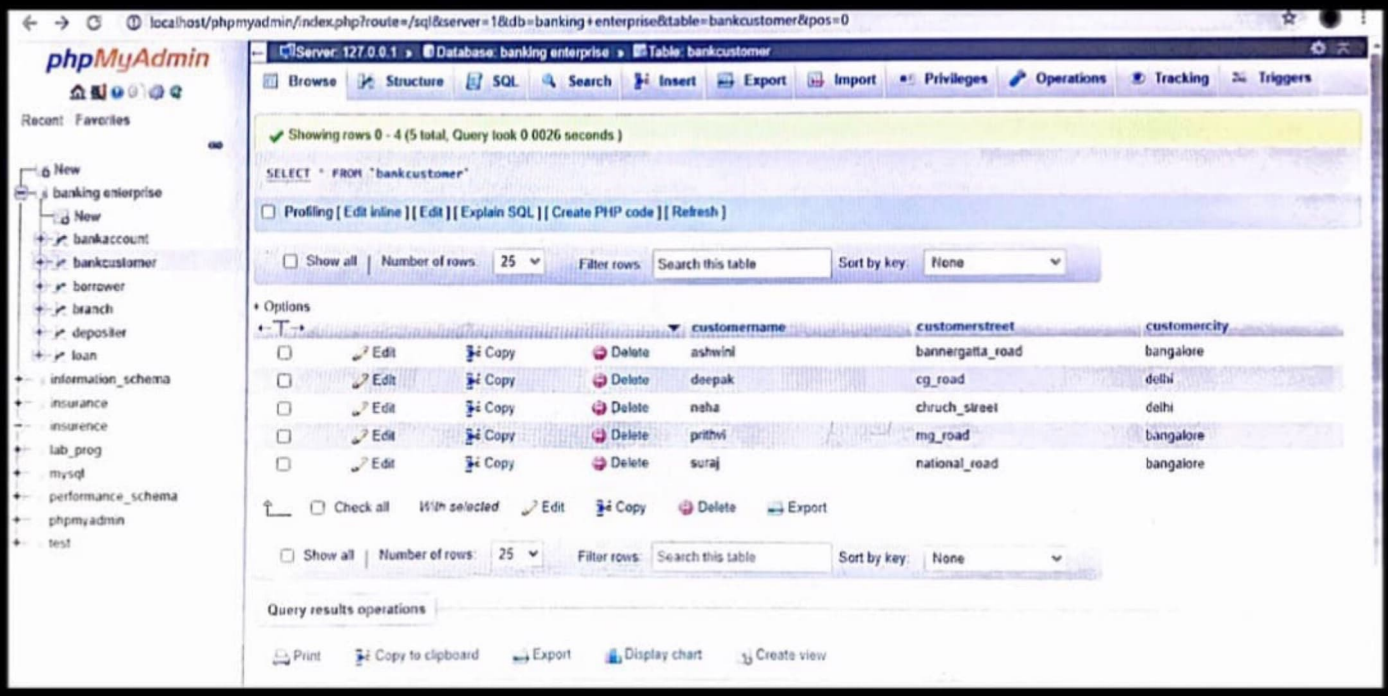
Queries:

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

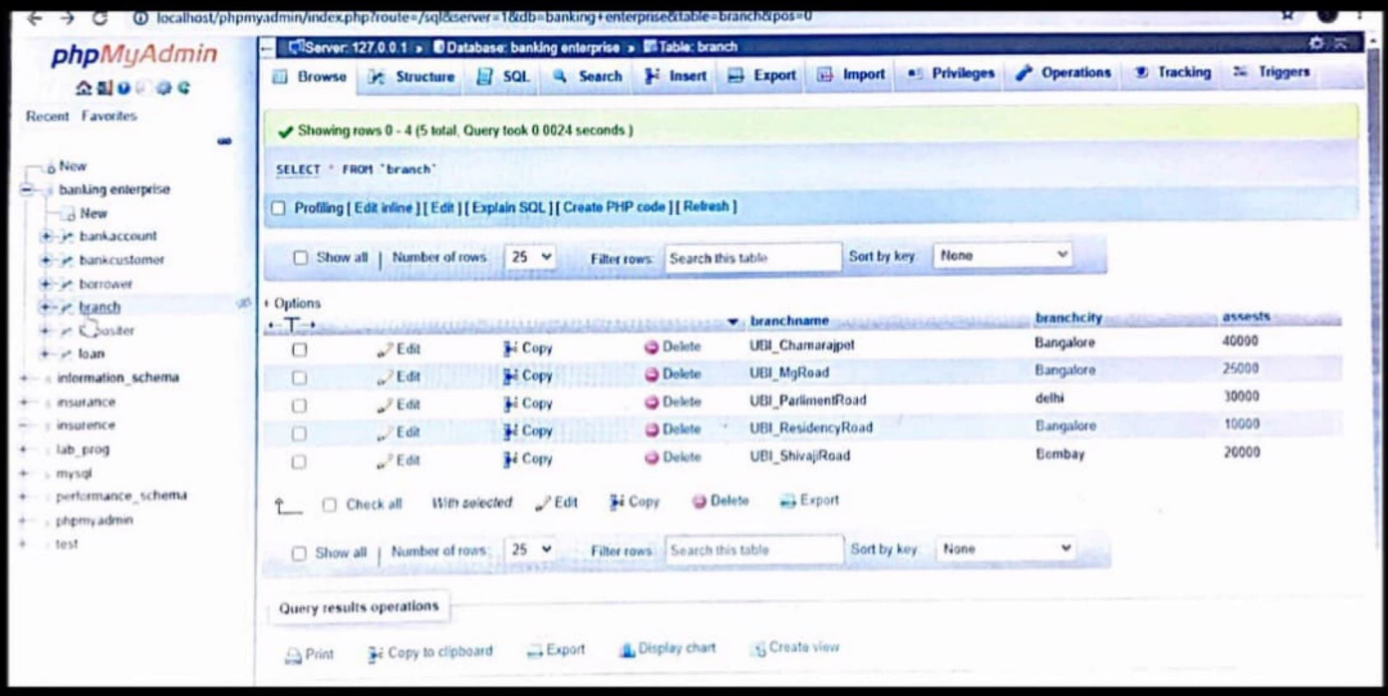
ii. Enter at least five tuples for each relation.



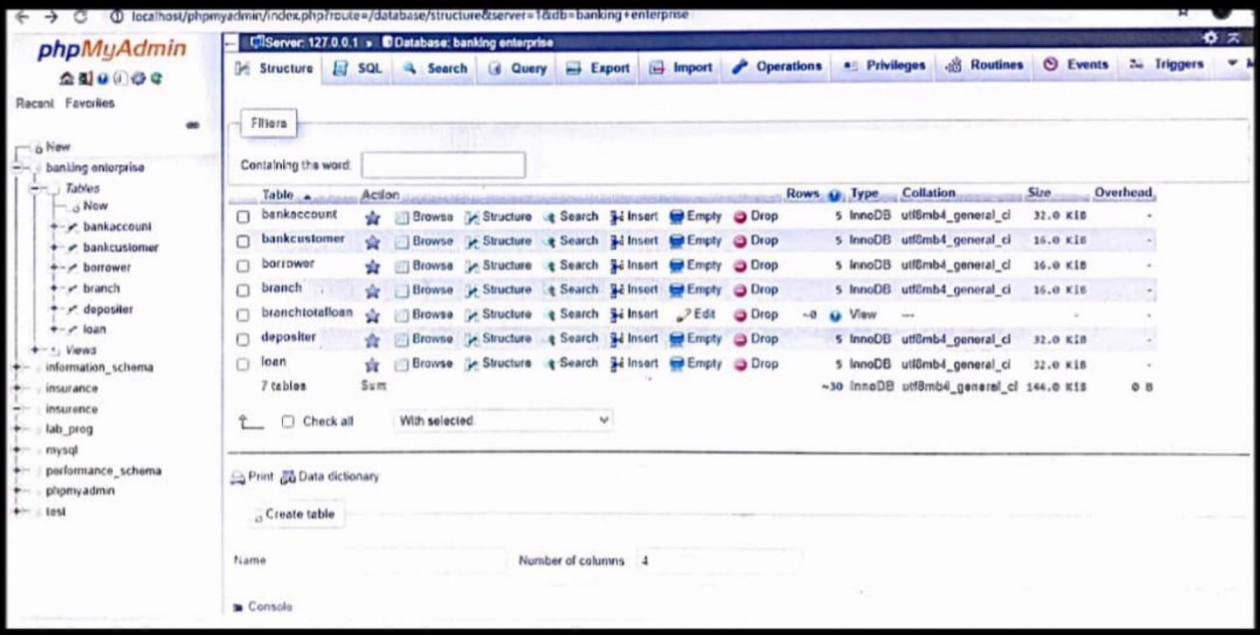
iii. Find all the customers who have at least two accounts at the Main branch.



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.



PROGRAM 3. SUPPLIER DATABASE

Consider the following schema:

SUPPLIERS (sid: integer, sname: string, address: string)

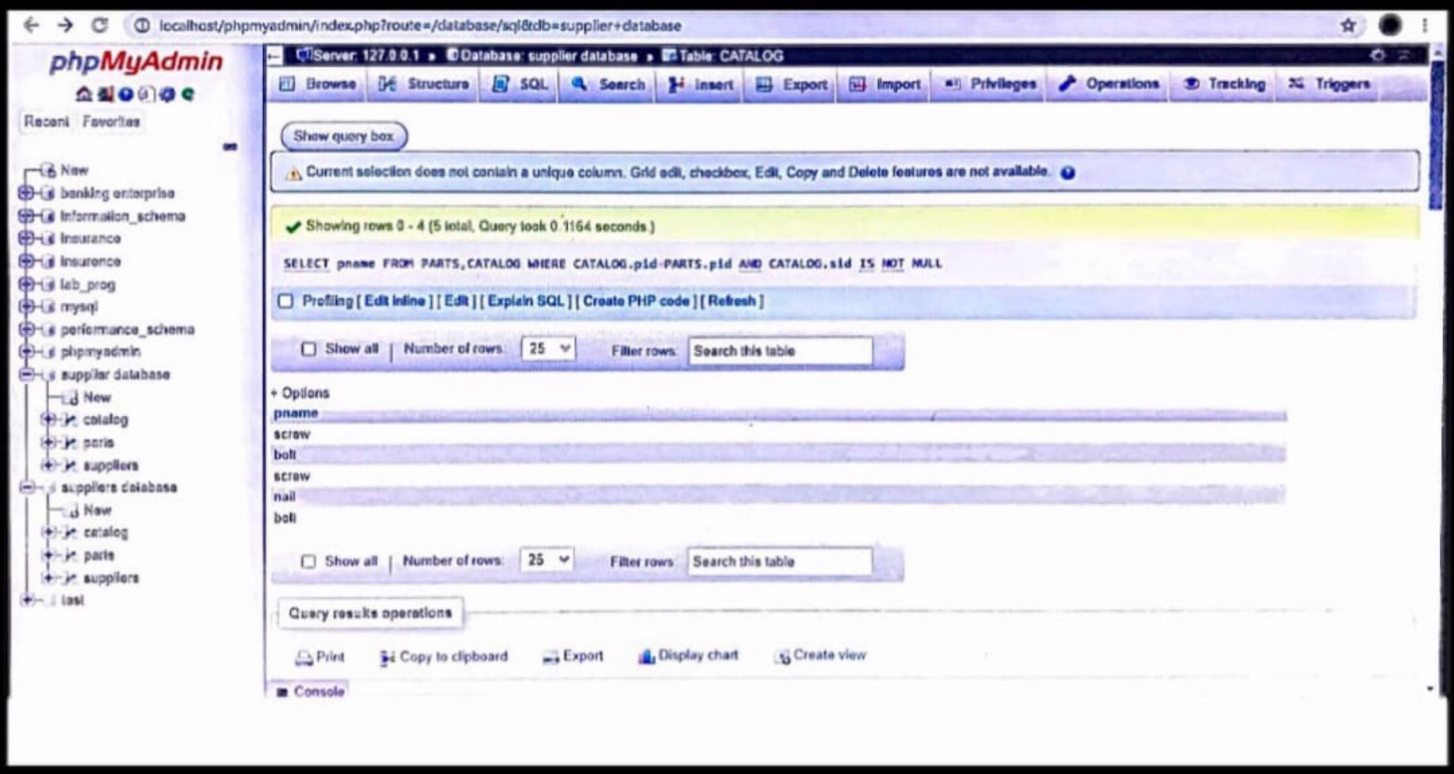
PARTS (pid: integer, pname: string, color: string)

CATALOG (sid: integer, pid: integer, cost: real)

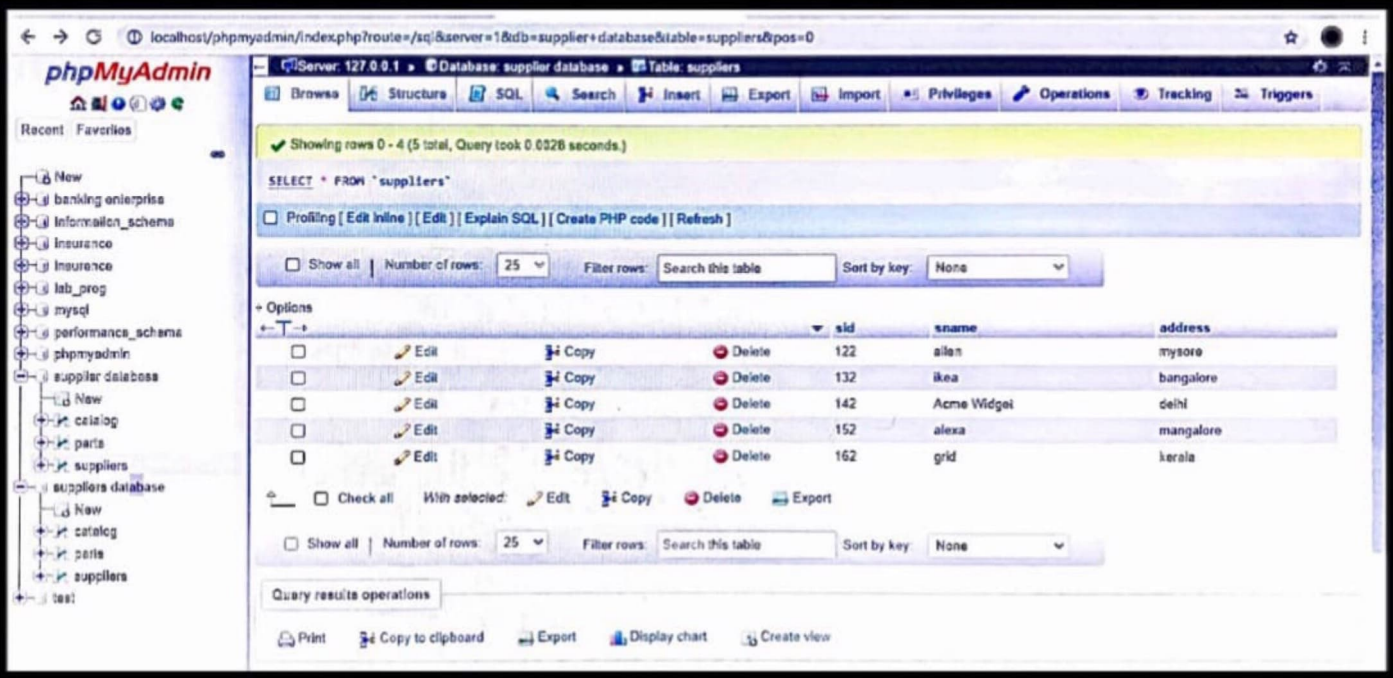
The Catalog relation lists the prices charged for parts by Suppliers.

Write the following queries in SQL:

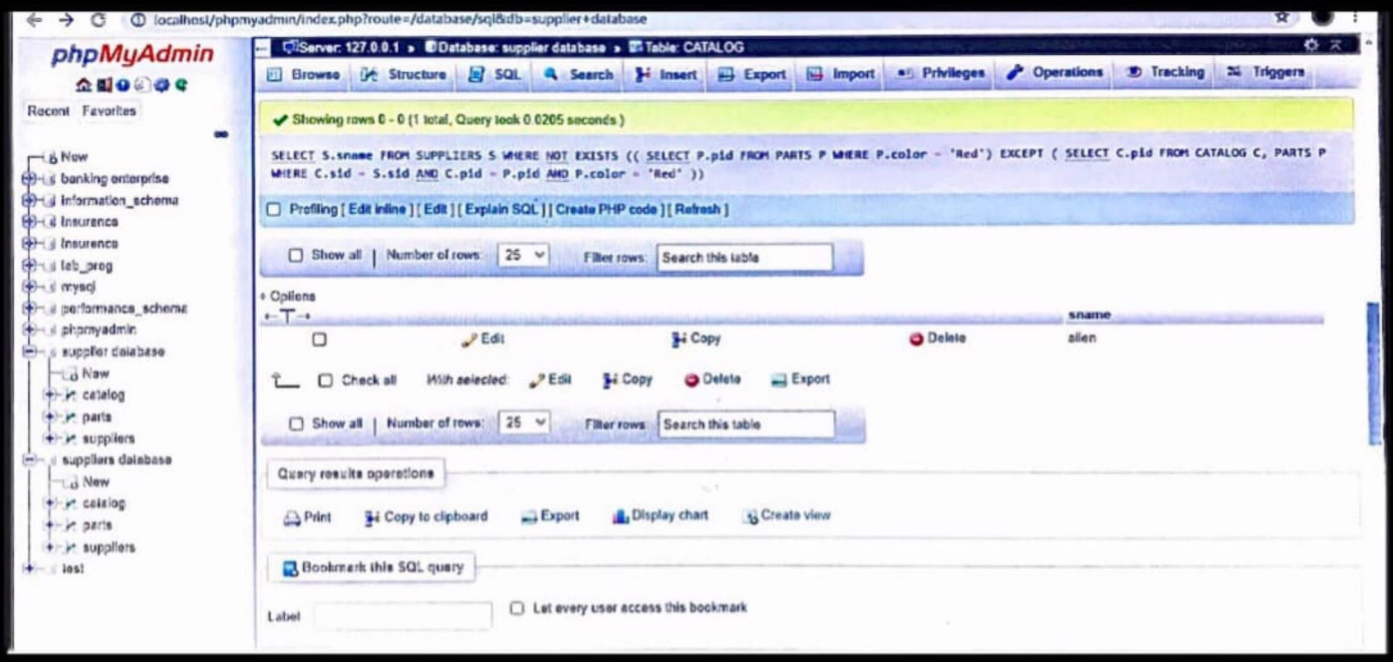
i. Find the pnames of parts for which there is some supplier.



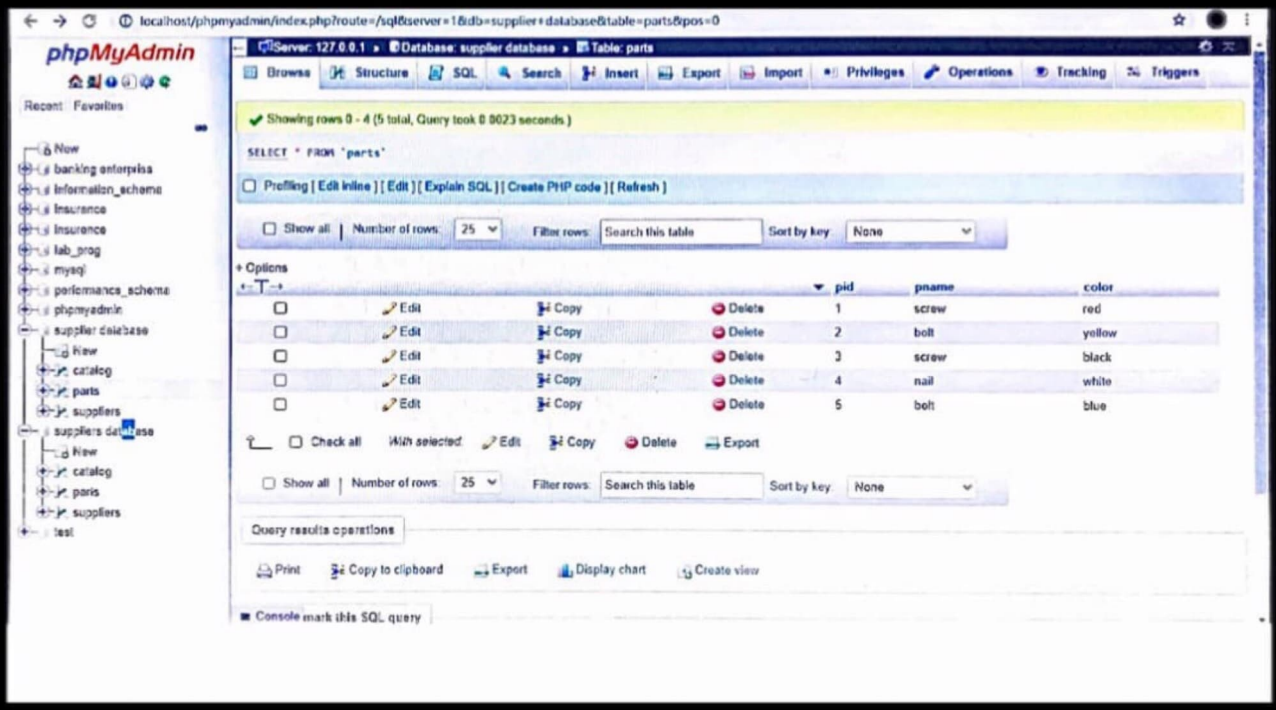
ii. Find the snames of suppliers who supply every part.



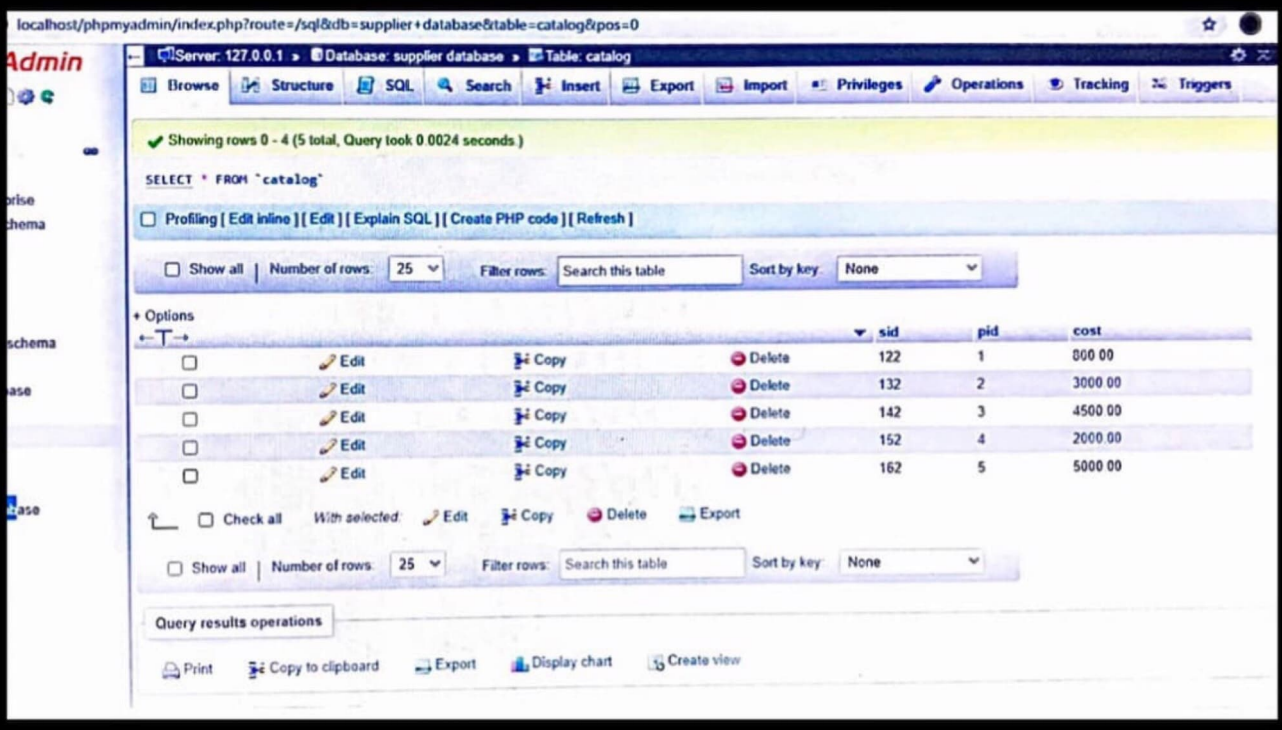
iii. Find the snames of suppliers who supply every red part.



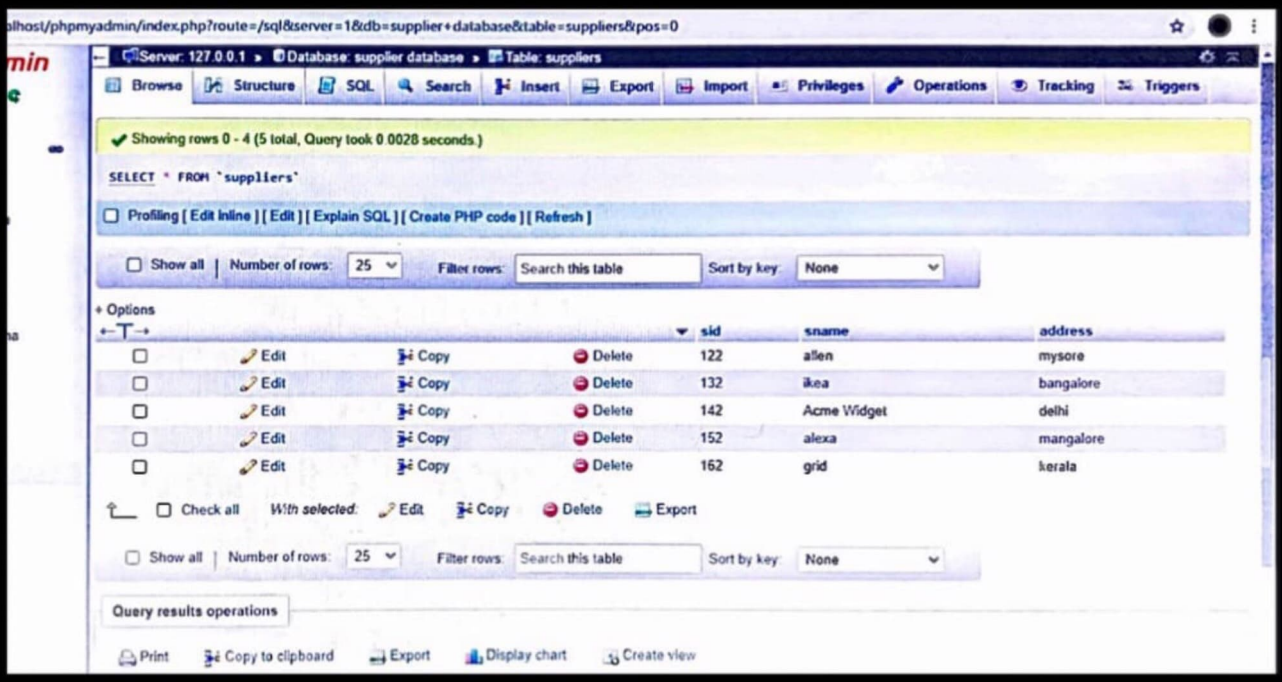
iv. Find the pnames of parts supplied by Acme Widget Suppliers and by no one else.



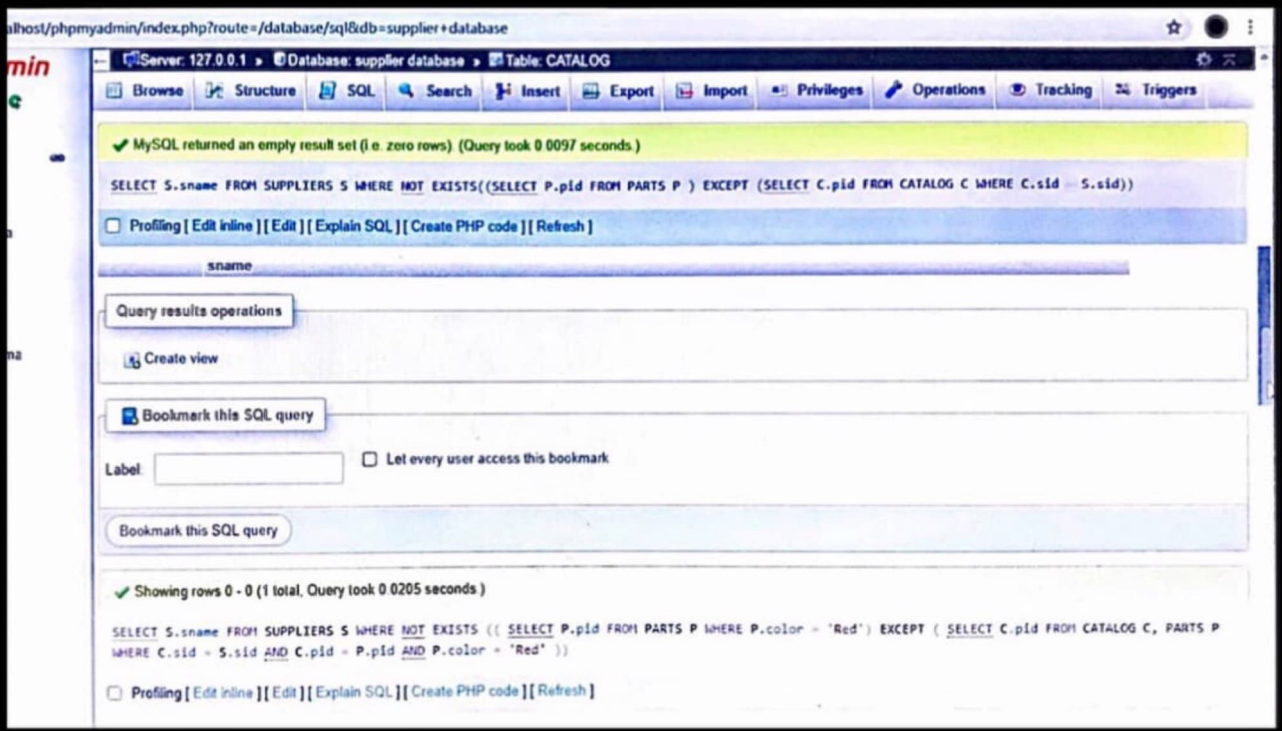
v. Find the sids of suppliers who charge more for some part than the average cost of that part (averaged over all the suppliers who supply that part).



vi. For each part, find the sname of the supplier who charges the most for that part.

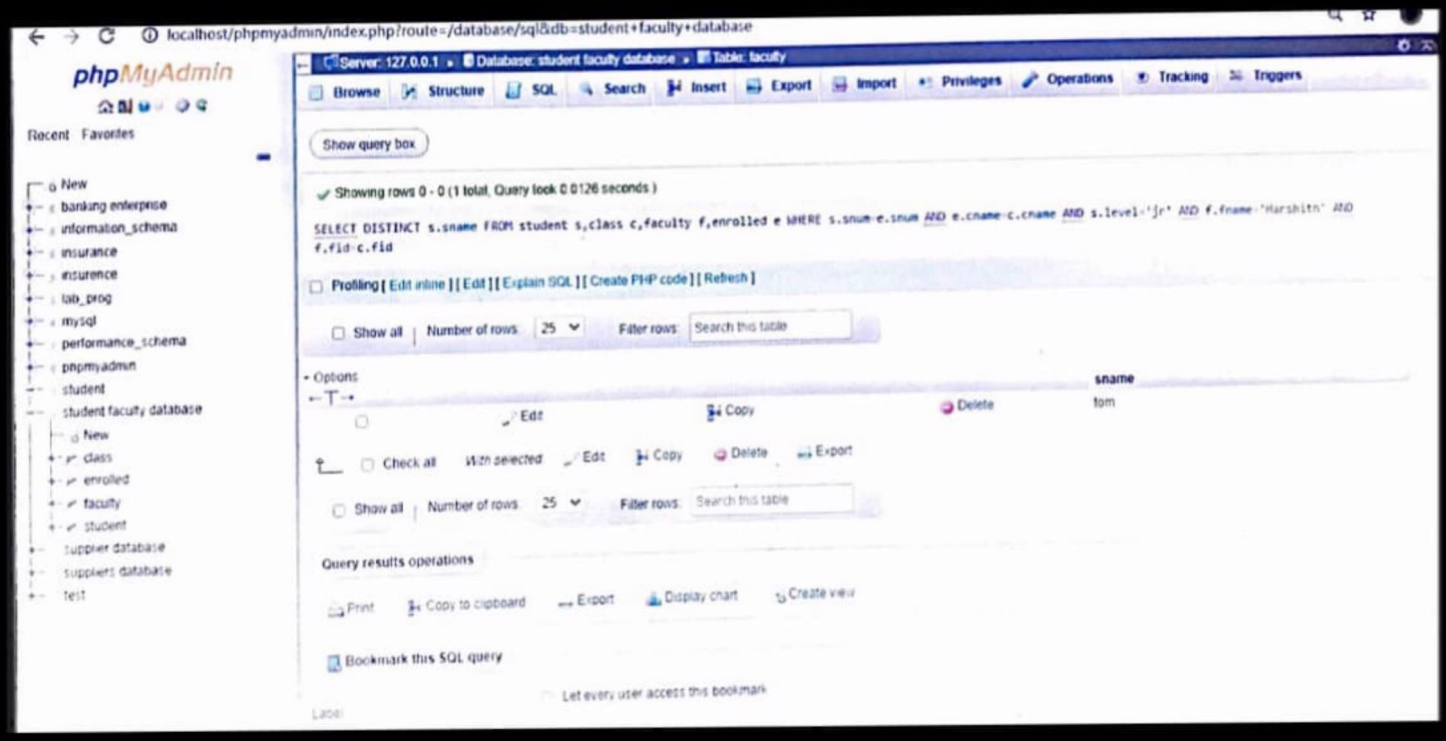


vii. Find the sids of suppliers who supply only red parts.

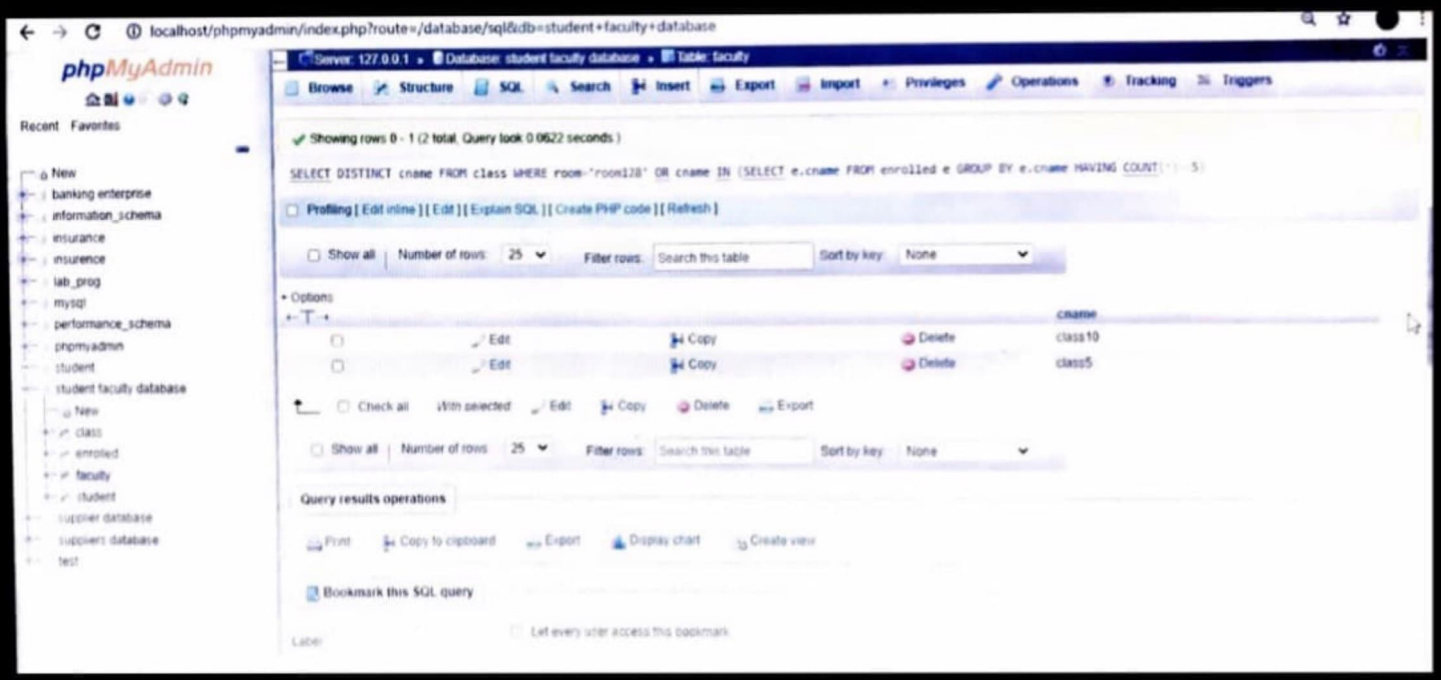


PROGRAM 4. STUDENT FACULTY DATABASE
Consider the following database for student enrolment for course:
STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)
CLASS (name: string, meets at: time, room: string, fid: integer)
ENROLLED (snum: integer, cname: string)
FACULTY (fid: integer, fname: string, deptid: integer)
The meaning of these relations is straightforward; 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.
i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by

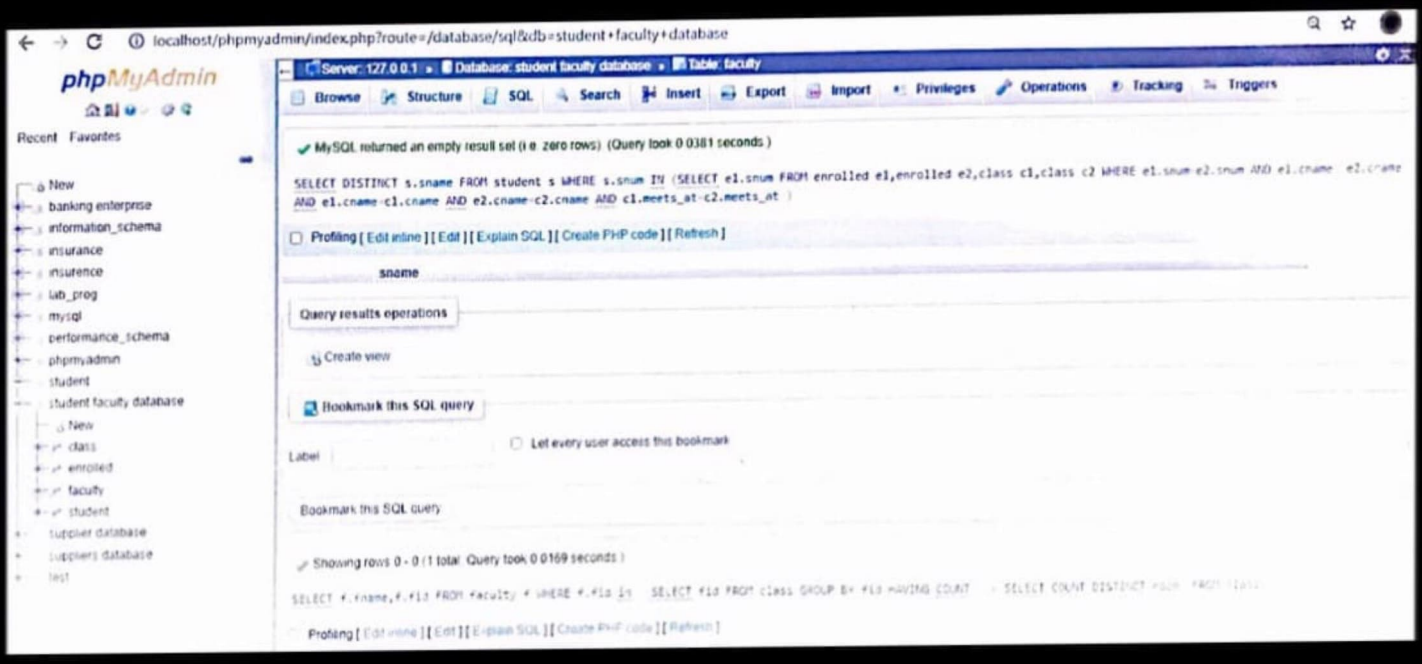
i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by



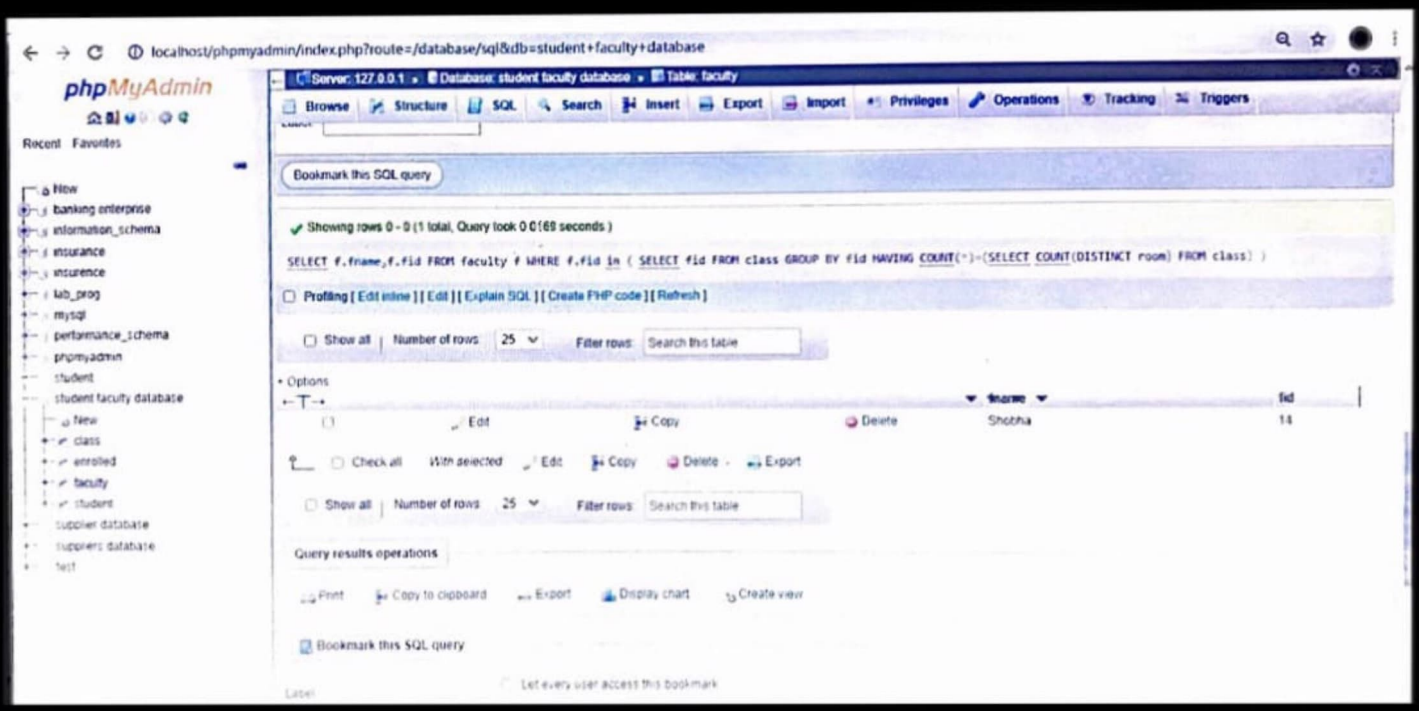
ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.



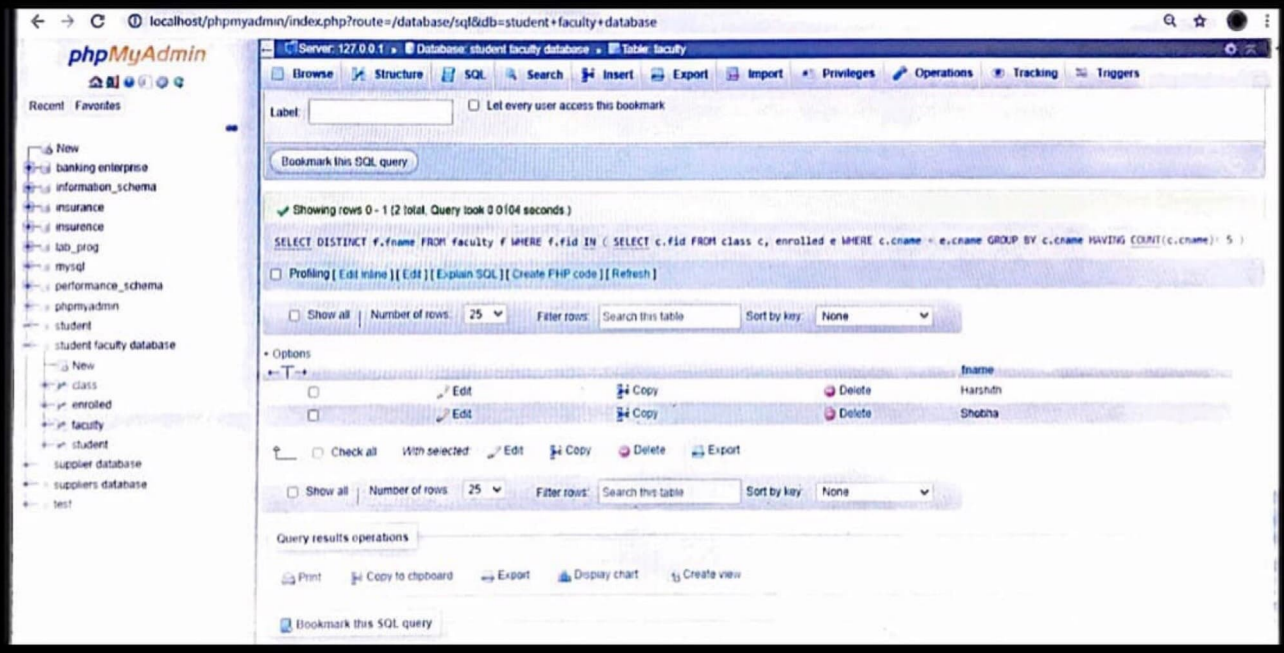
iii. Find the names of all students who are enrolled in two classes that meet at the same time.



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



v. Find the names of faculty members for whom the combined enrolment of the courses that they teach is less
than five.



PROGRAM 5. AIRLINE FLIGHT DATABASE

Consider the following database that keeps track of airline flight information:

FLIGHTS (flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time, price: integer) AIRCRAFT (aid: integer, aname: string, cruisingrange: integer)

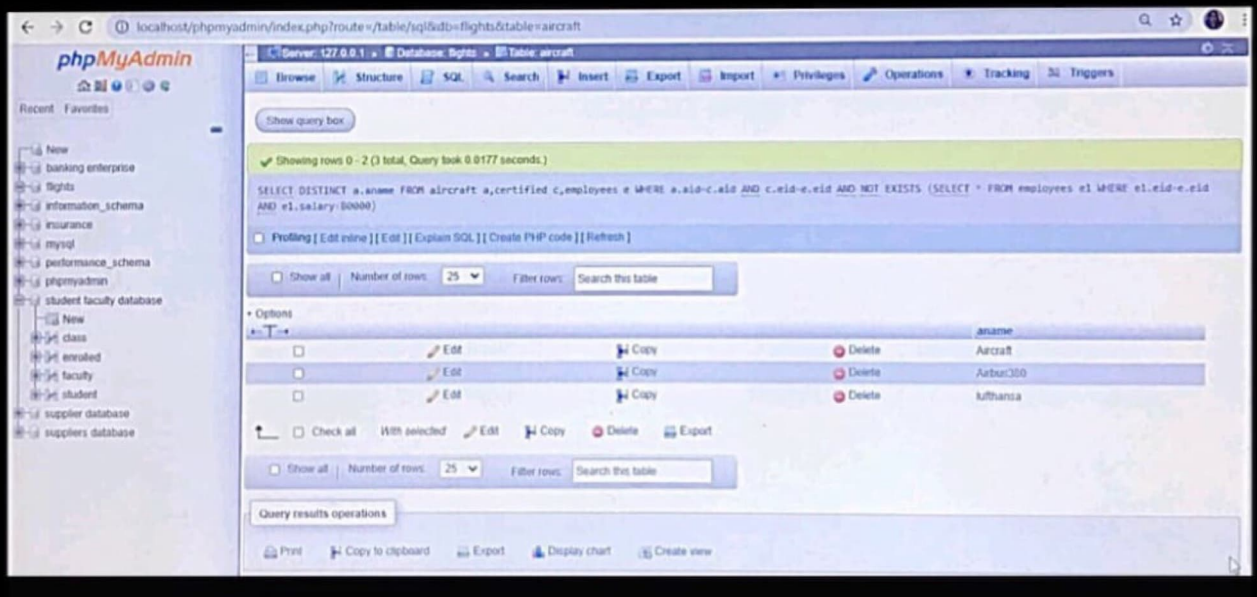
CERTIFIED (eid: integer, aid: integer)

EMPLOYEE (eid: integer, ename: string, salary: integer)

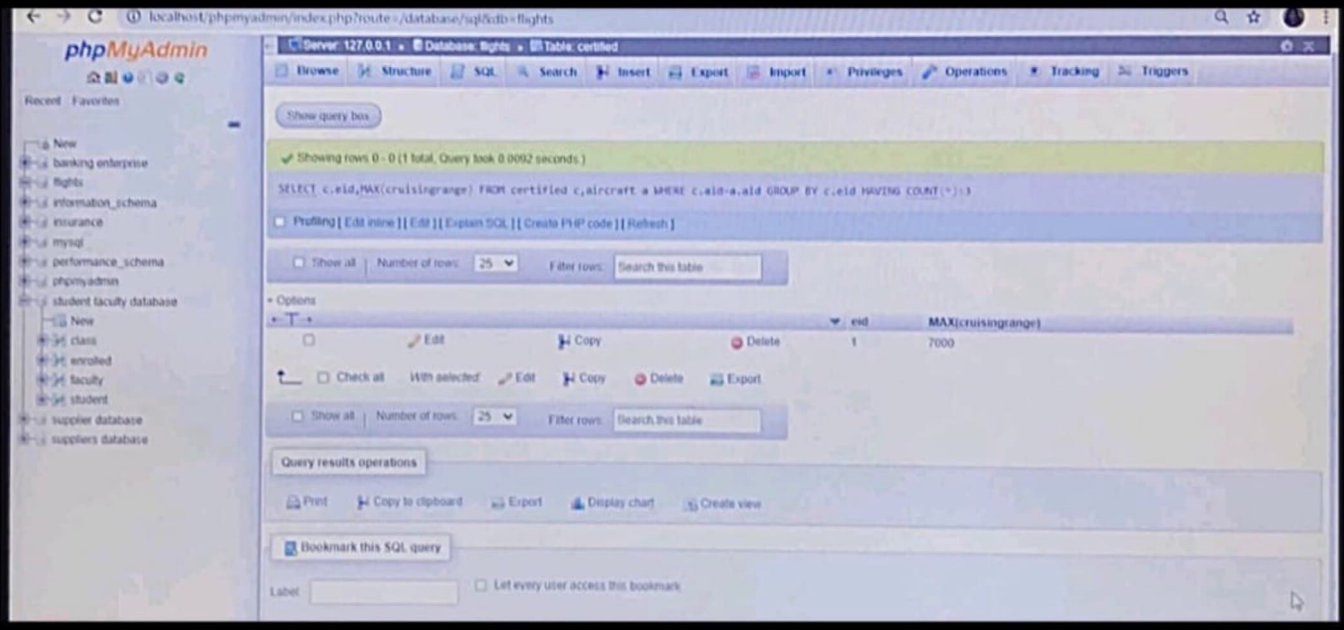
Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly.

Write each of the following queries in SQL.

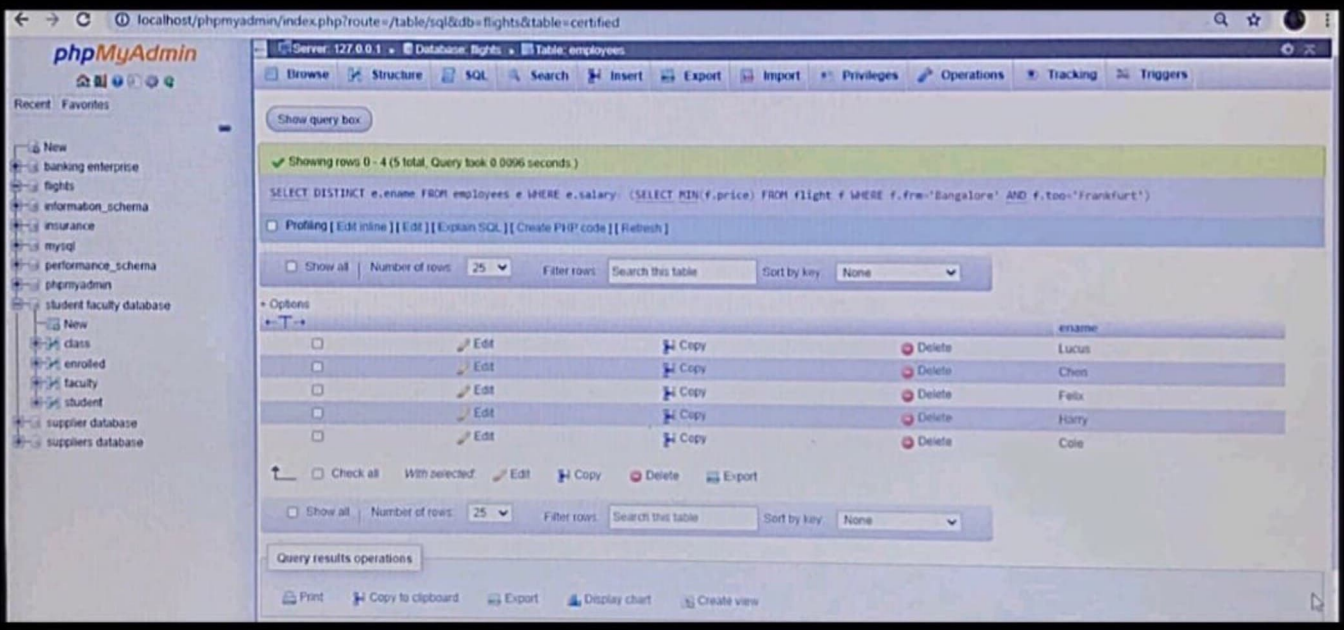
i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000.



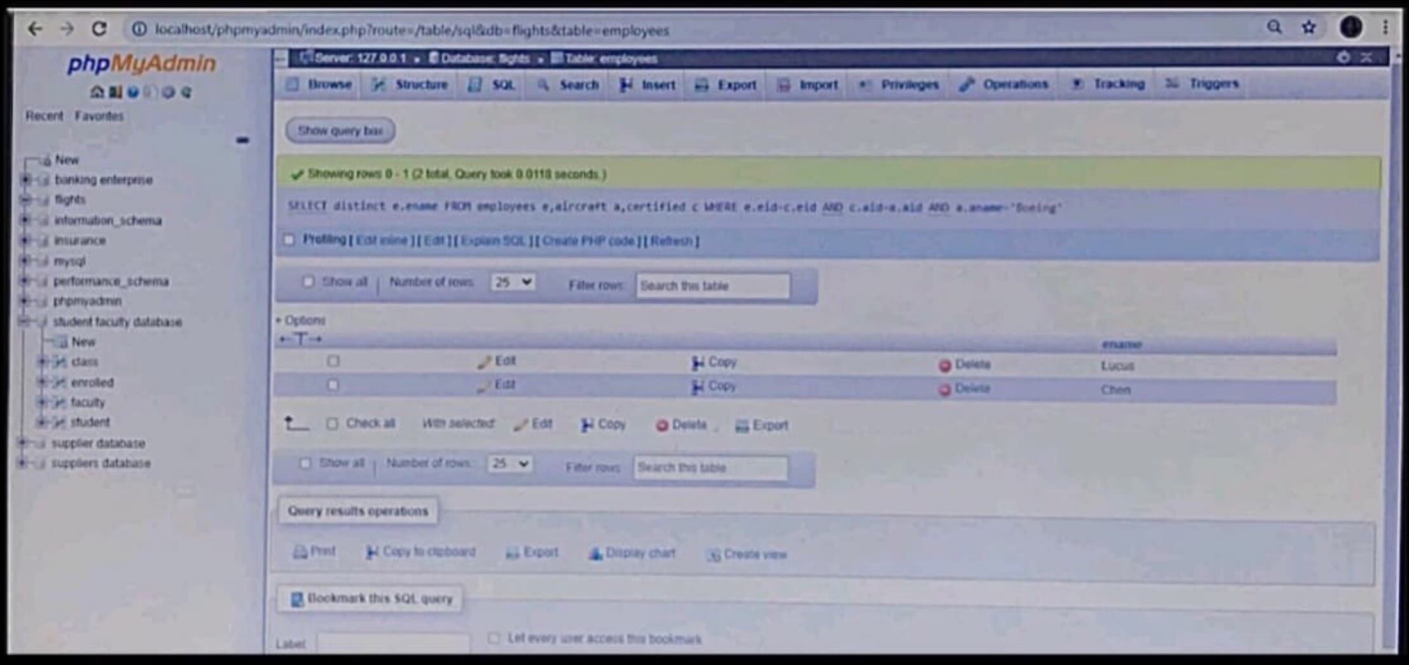
ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruising range of
the aircraft for which she or he is certified.



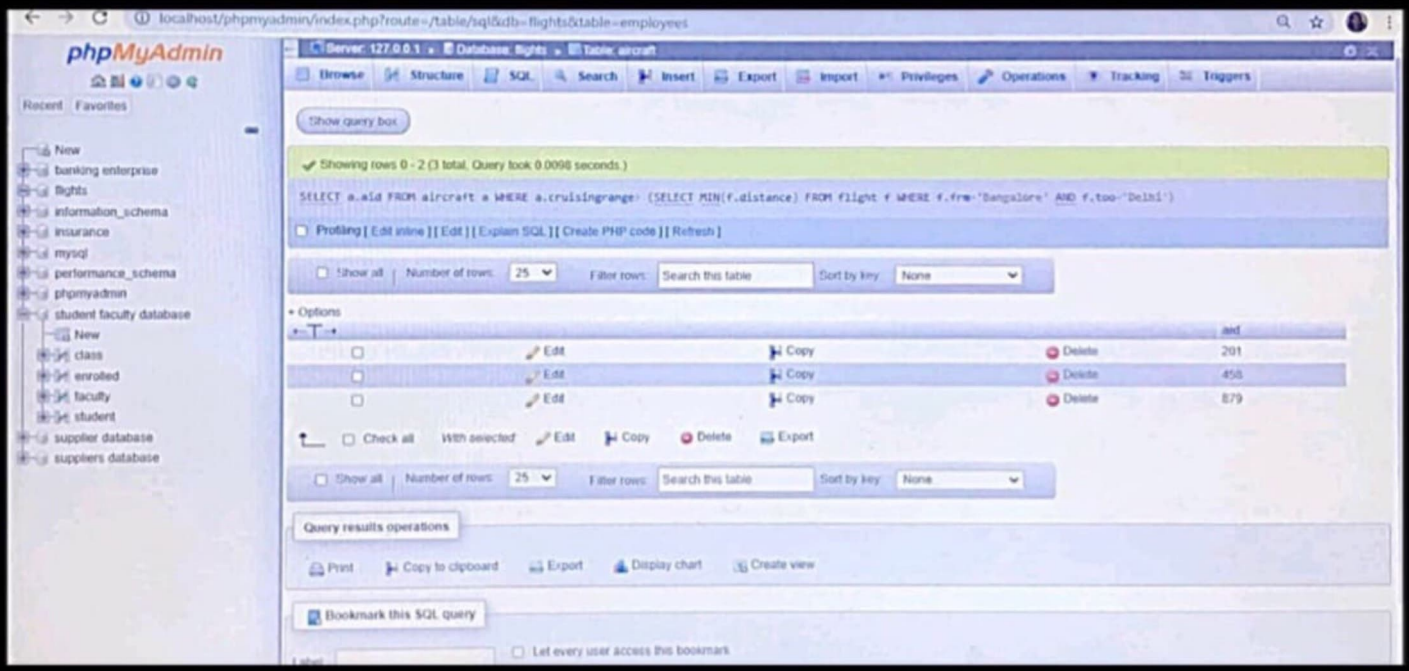
iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.



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

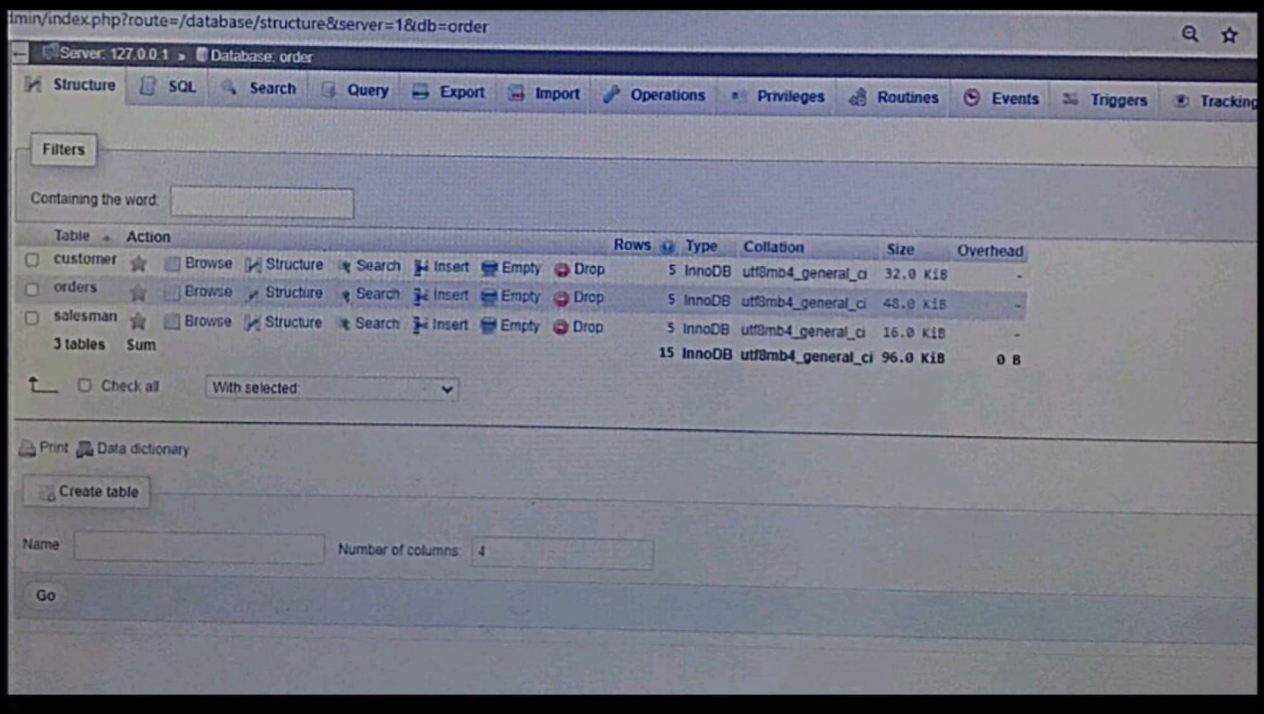


v. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

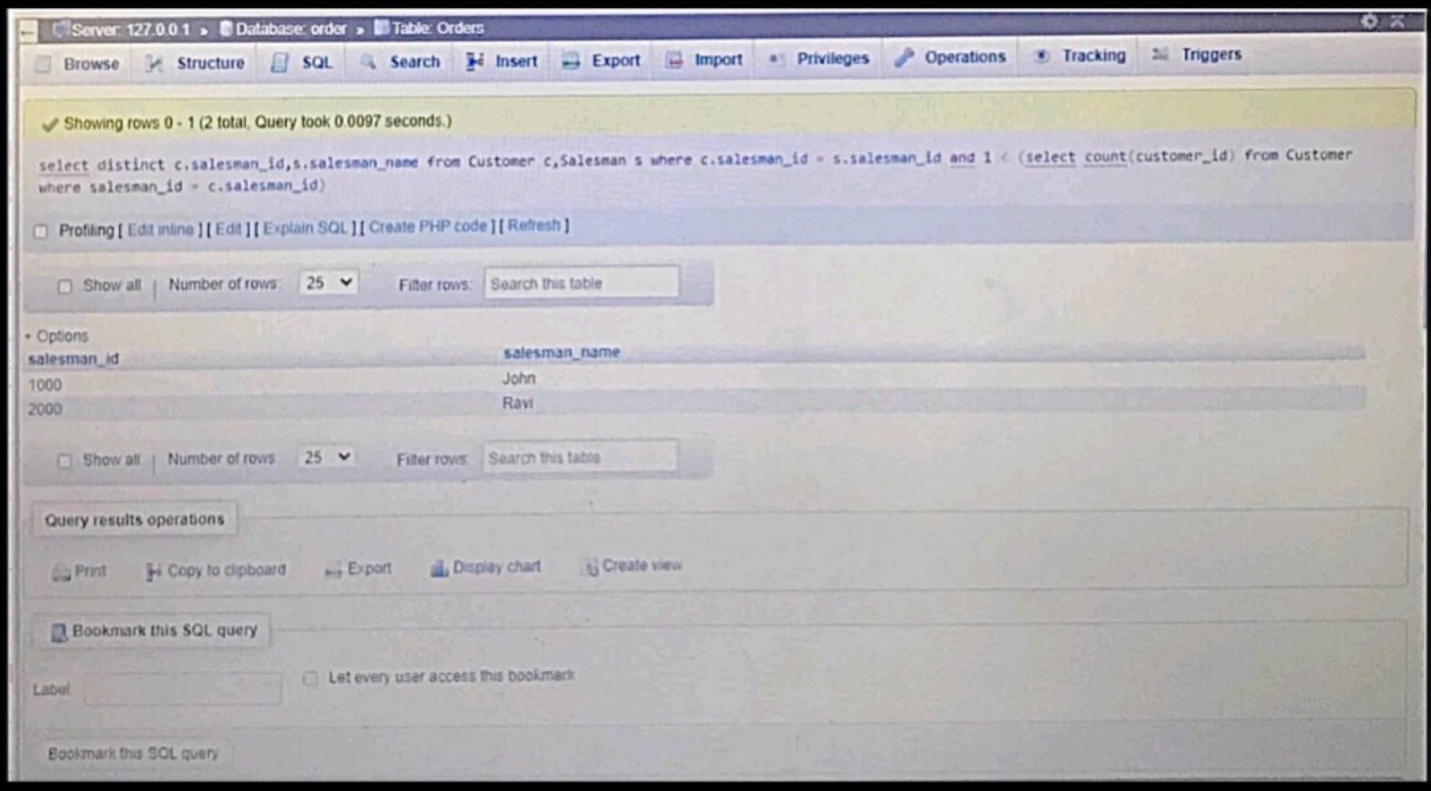


PROGRAM 6. ORDER PROCESSING DATABASE
Consider the following relations for an Order Processing database application in a company. CUSTOMER (CUST #: int, cname: String, city: String)
ORDER (order #: int, odate: date, cust #: int, ord-Amt: int)
ITEM (item #: int, unit-price: int)
ORDER-ITEM (order #: int, item #: int, qty: int)
WAREHOUSE (warehouse #: int, city: String)
SHIPMENT (order #: int, warehouse #: int, ship-date: date)

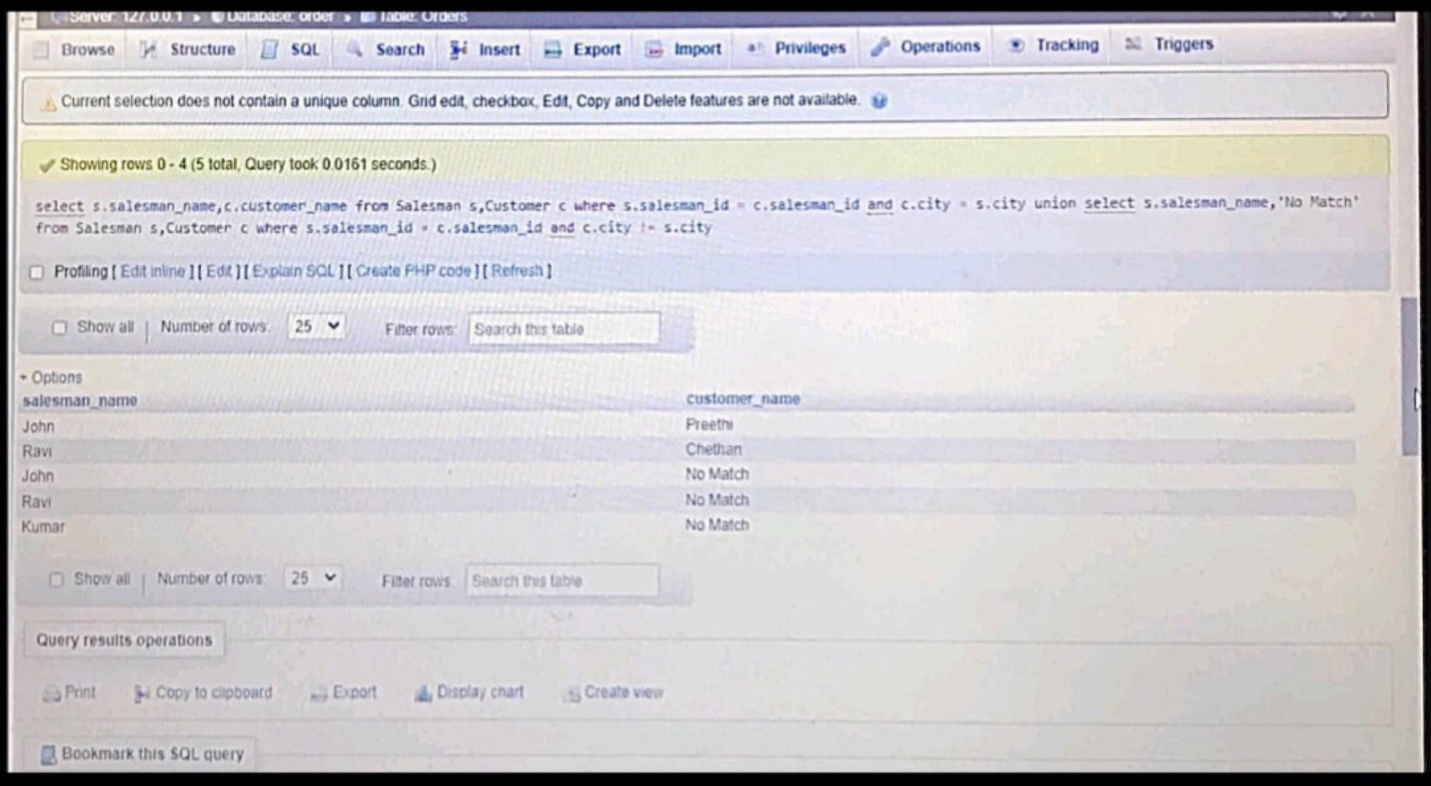
i. Create the above tables by properly specifying the primary keys and the foreign keys and the foreign keys.
ii. Enter at least five tuples for each relation.



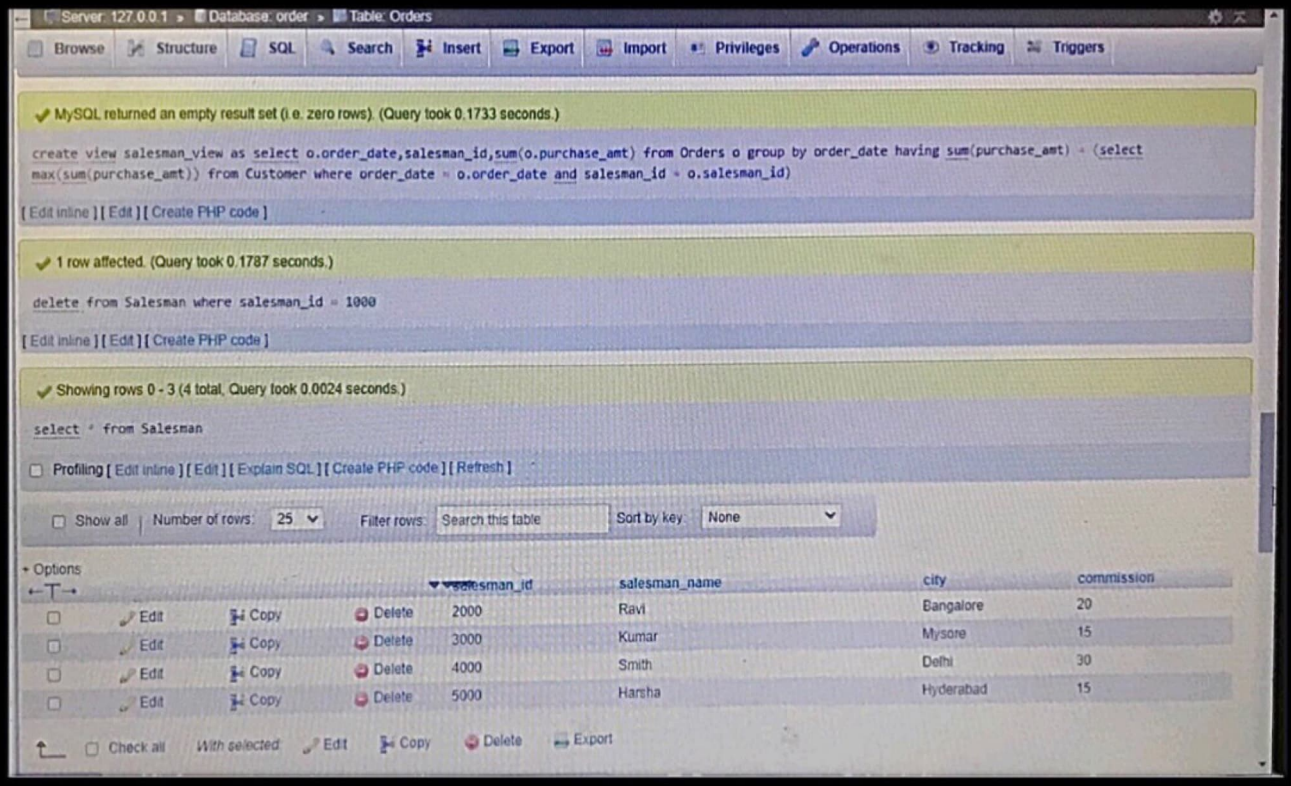
ii. Find the name and numbers of all salesmen who had more than one customer.



iii.List all salesmen and indicate those who have and don't have customers in their cities



iv. Create a view that finds the salesman who has the customer with the highest order of a day.



PROGRAM 7. BOOK DEALER DATABASE

The following tables are maintained by a book dealer:

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

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

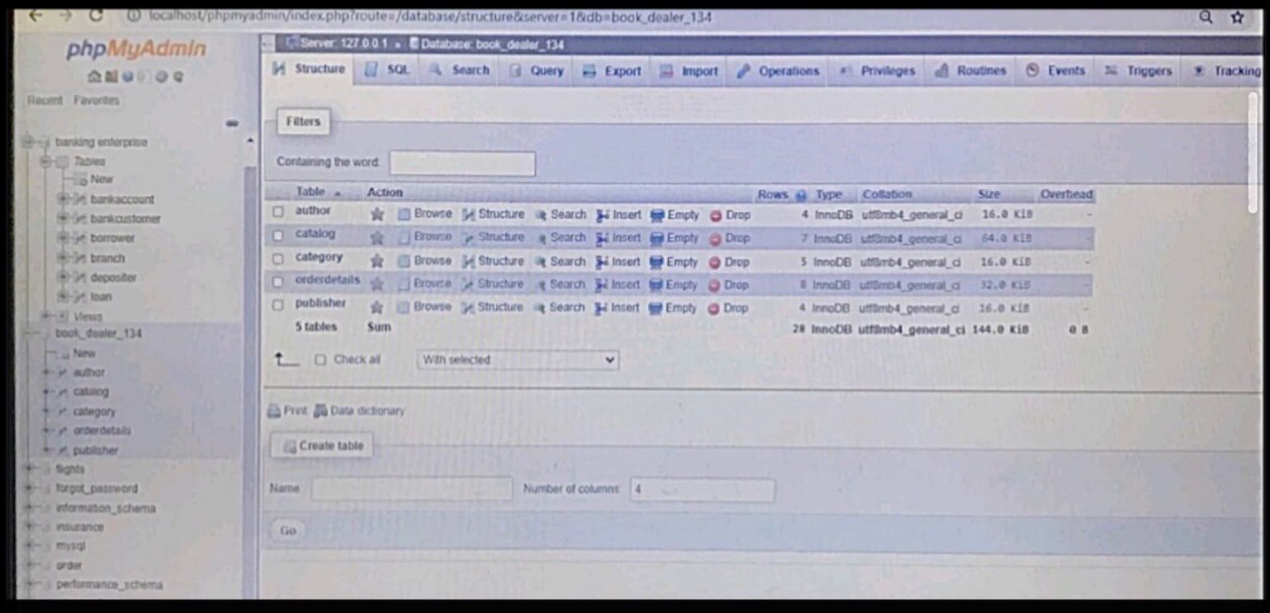
CATALOG (book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

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

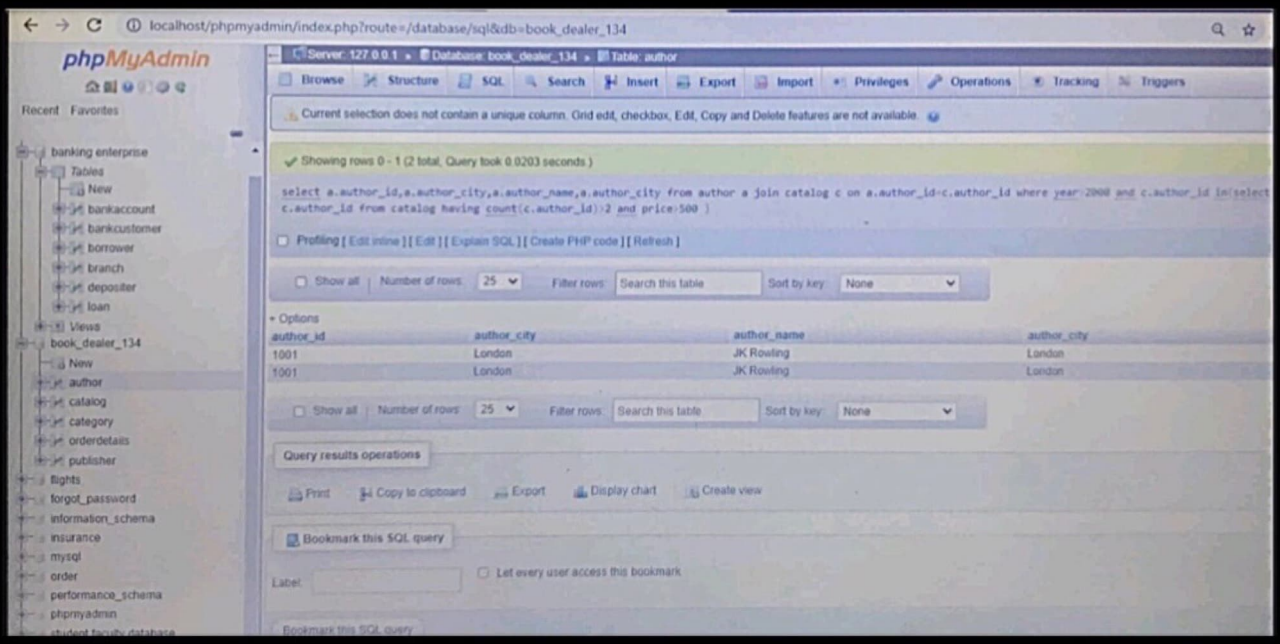
ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

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

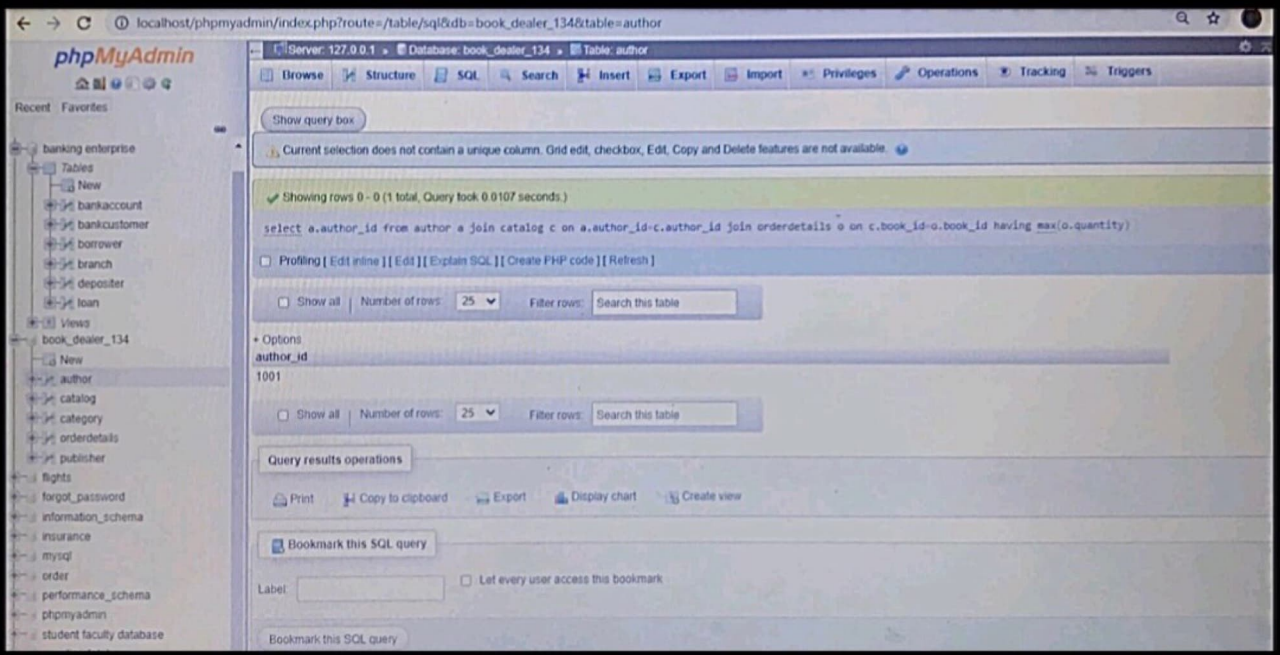
ii. Enter at least five tuples for each relation.



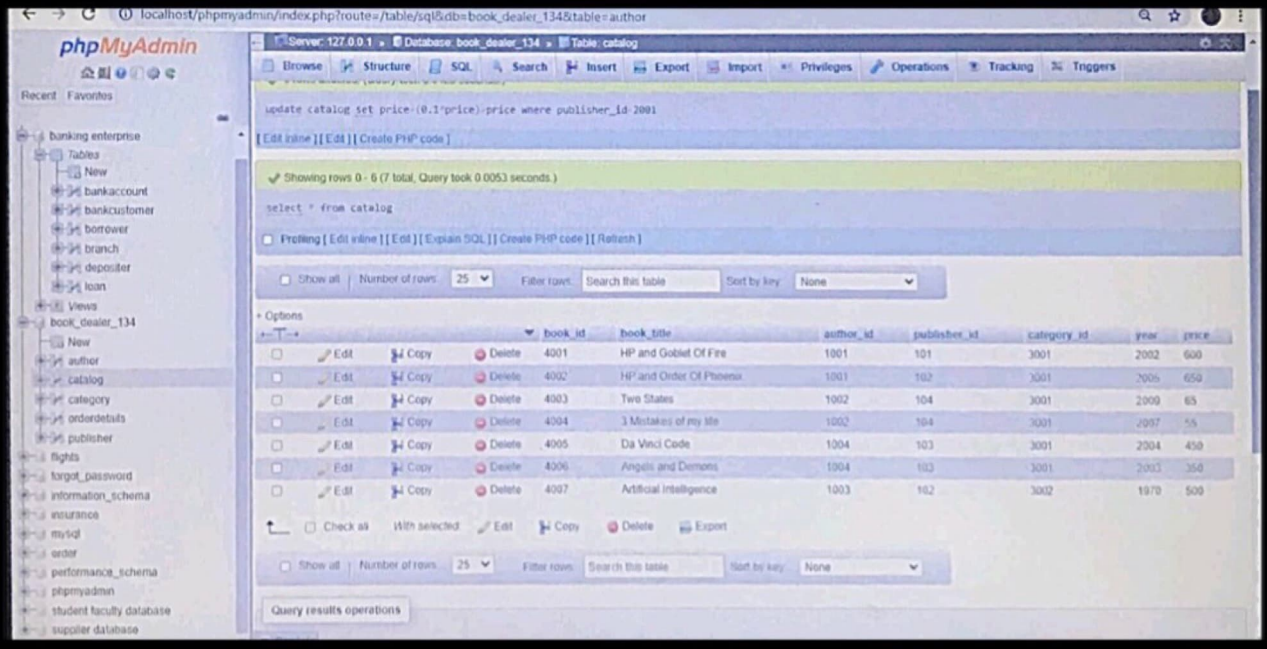
iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books in the
catalog and the year of publication is after 2000.



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%



PROGRAM 8. STUDENT ENROLLMENT DATABASE

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)

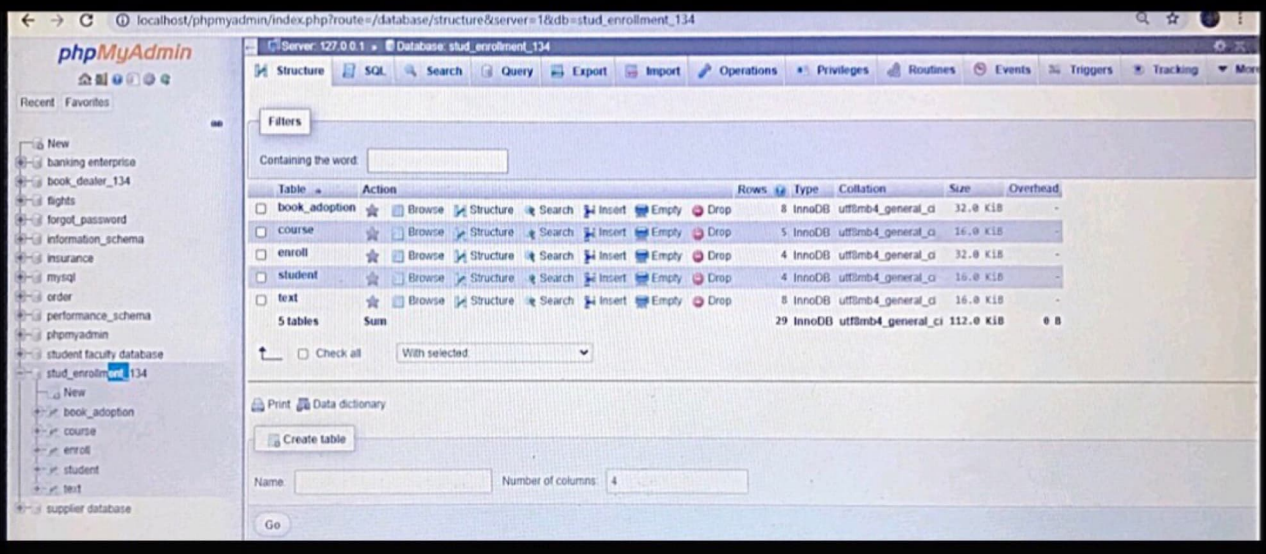
ENROLL (regno: String, cname: String, sem: int, marks: int)

BOOK_ADOPTION (course #: int, sem: int, book-ISBN: int)

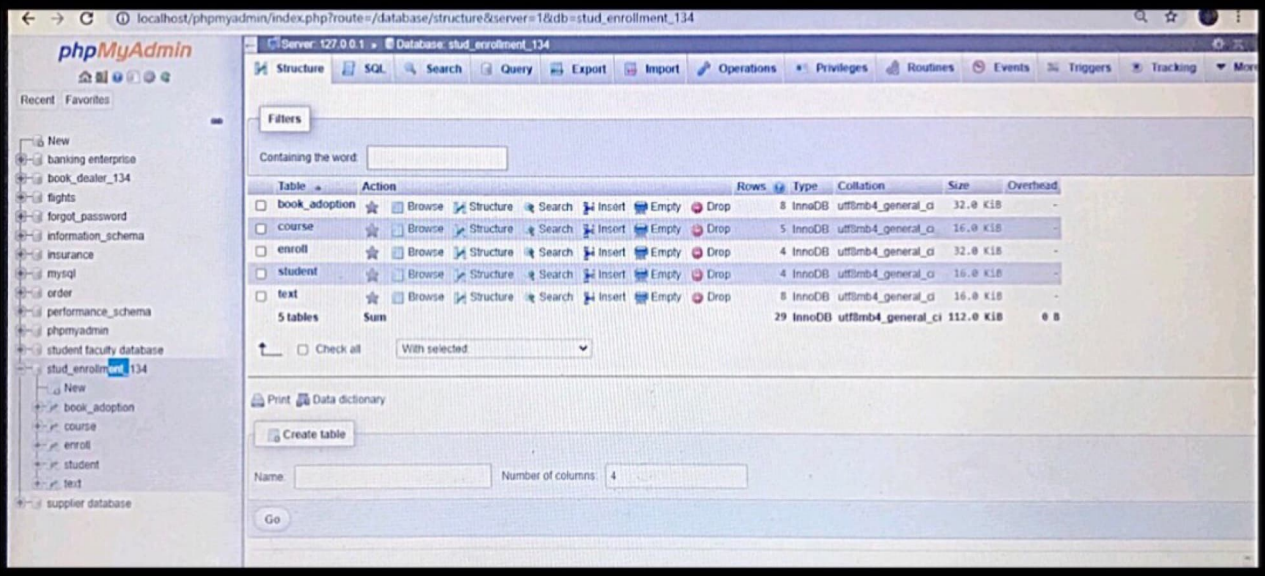
TEXT(book-ISBN:int, book-title:String, publisher:String, author:String)

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

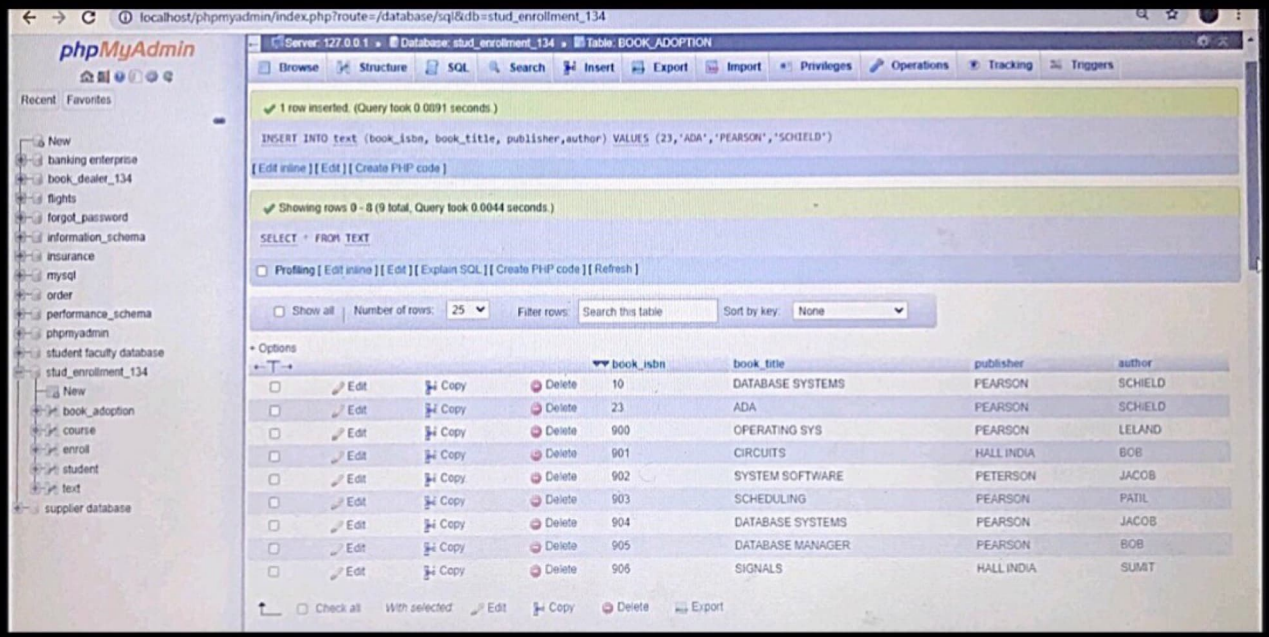
ii. Enter at least five tuples for each relation.



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

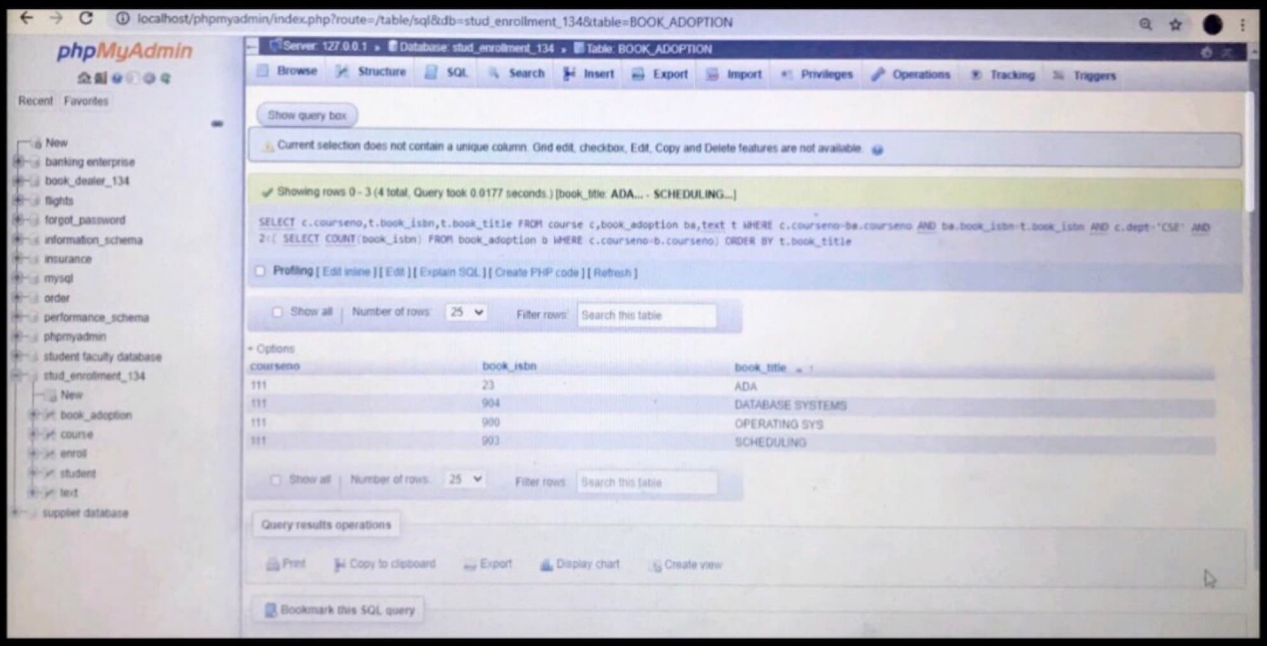


iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses
offered by the ‘CS’ department that use more than two books.



v. List any department that has all its adopted

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



PROGRAM 9: MOVIE DATABASE

Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

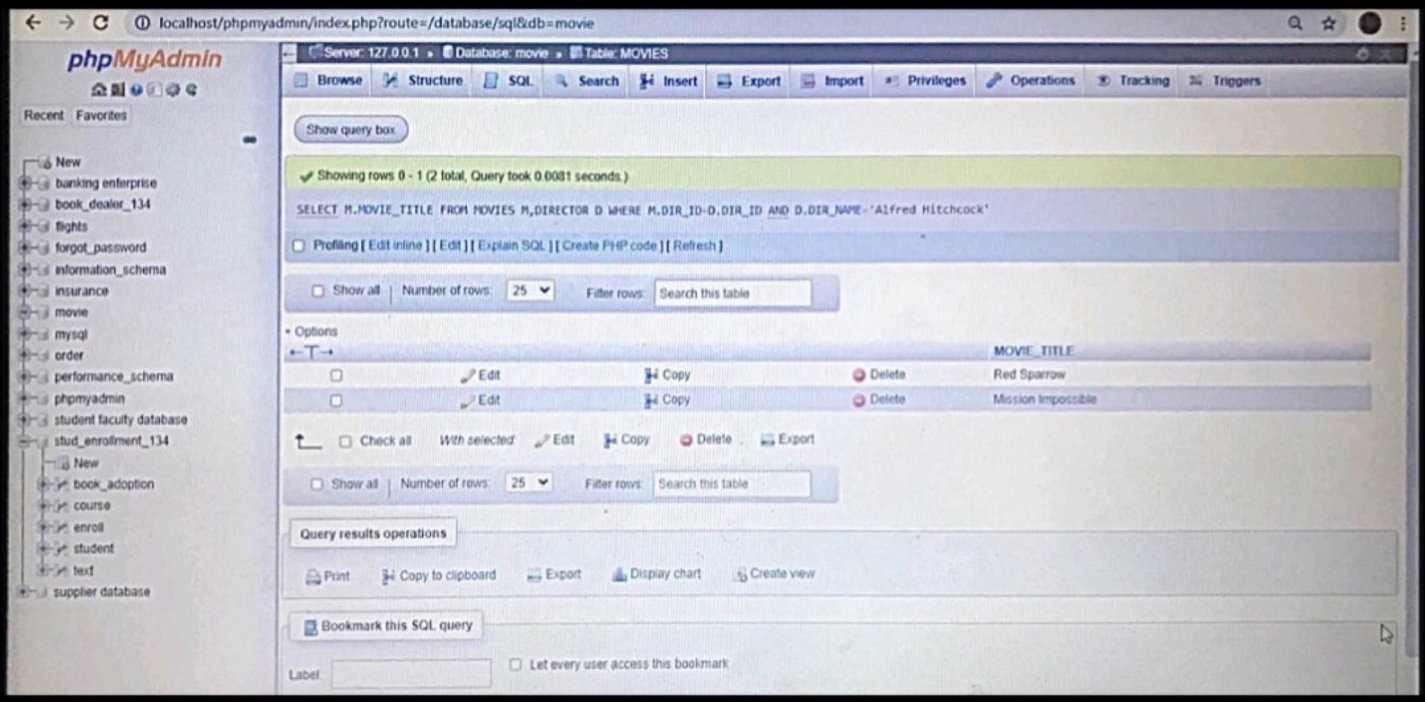
MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

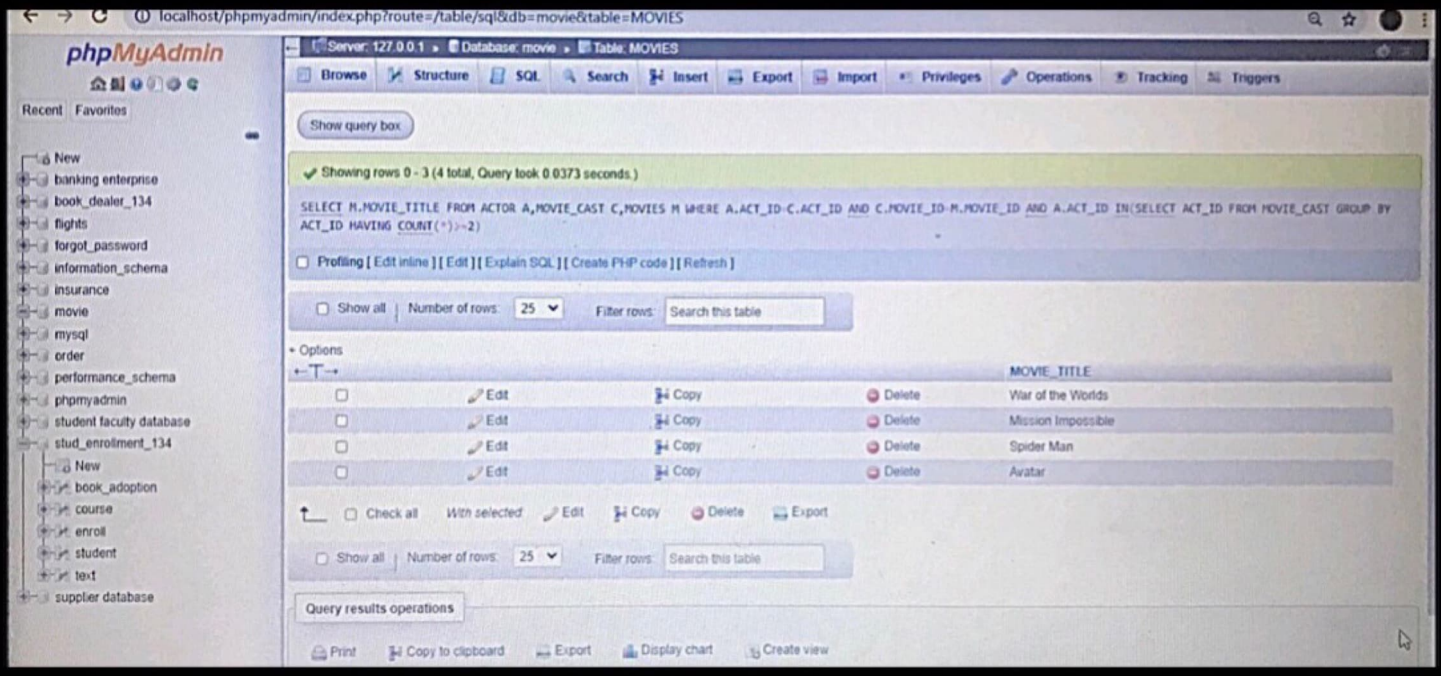
RATING(Mov_id, Rev_Stars)

Write SQL queries to

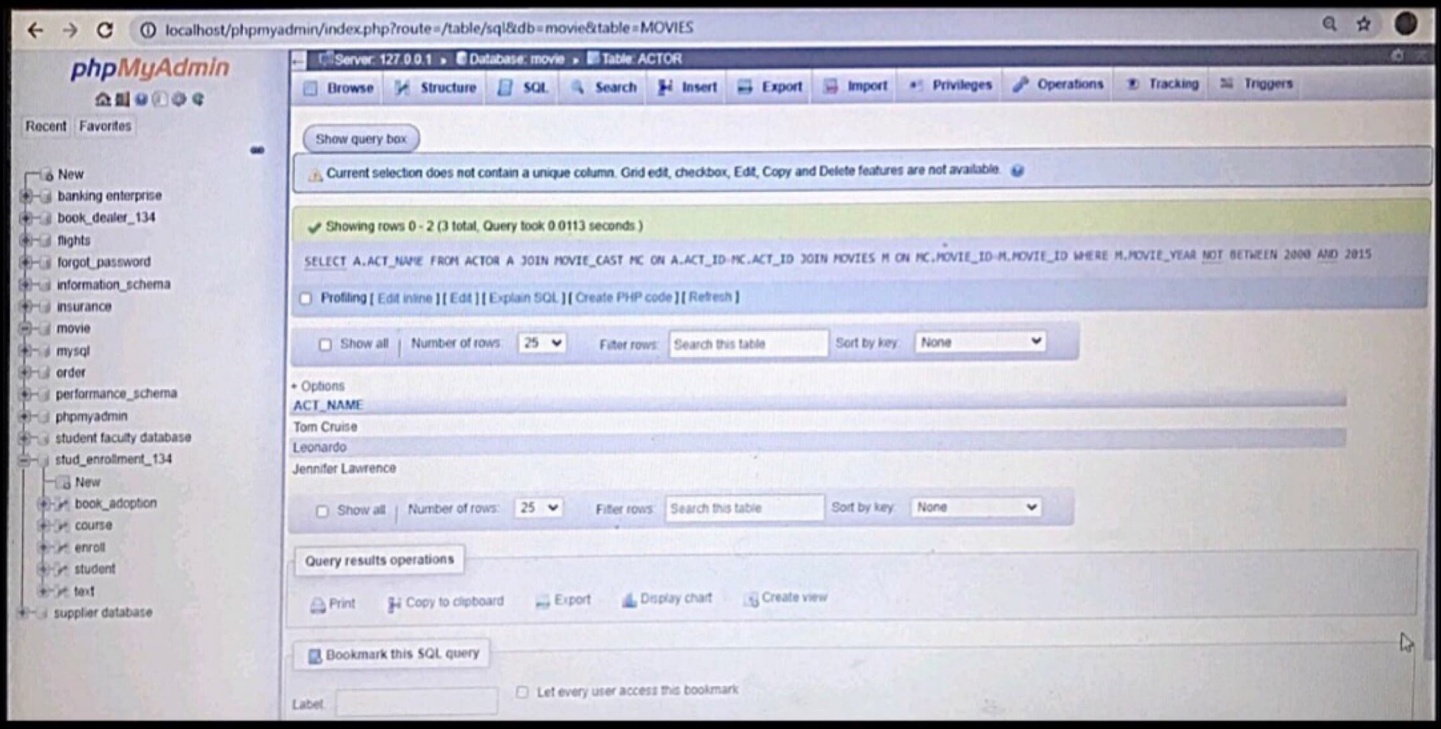
i. List the titles of all movies directed by 'Hitchcock'.



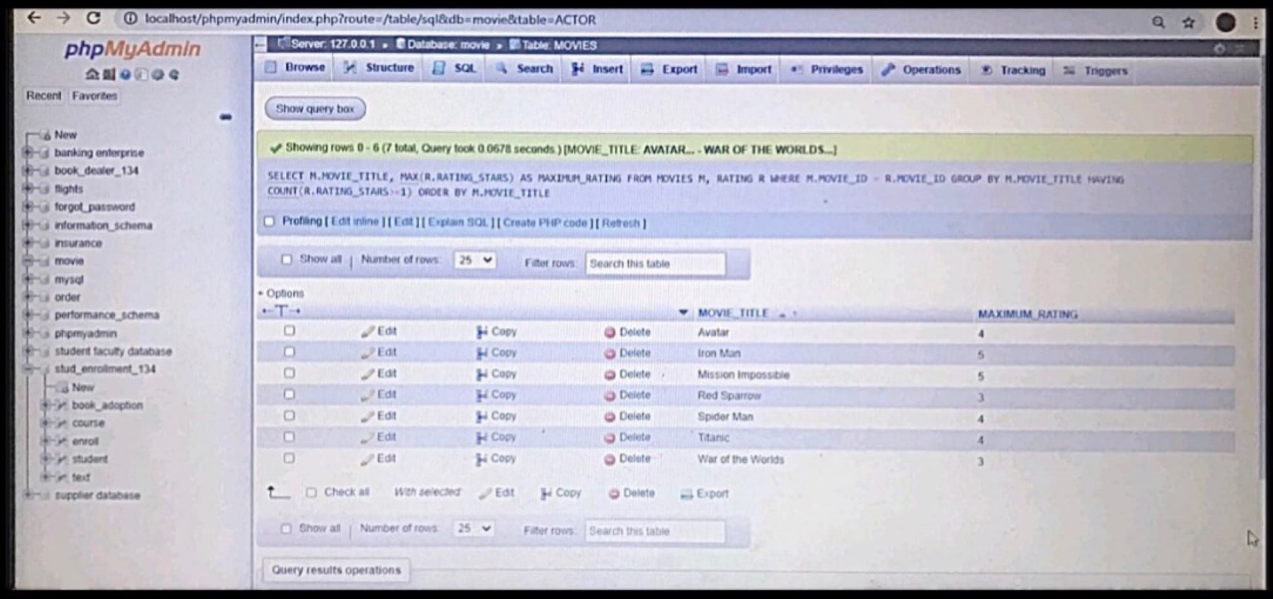
ii. Find the movie names where one or more actors acted in two or more movies.



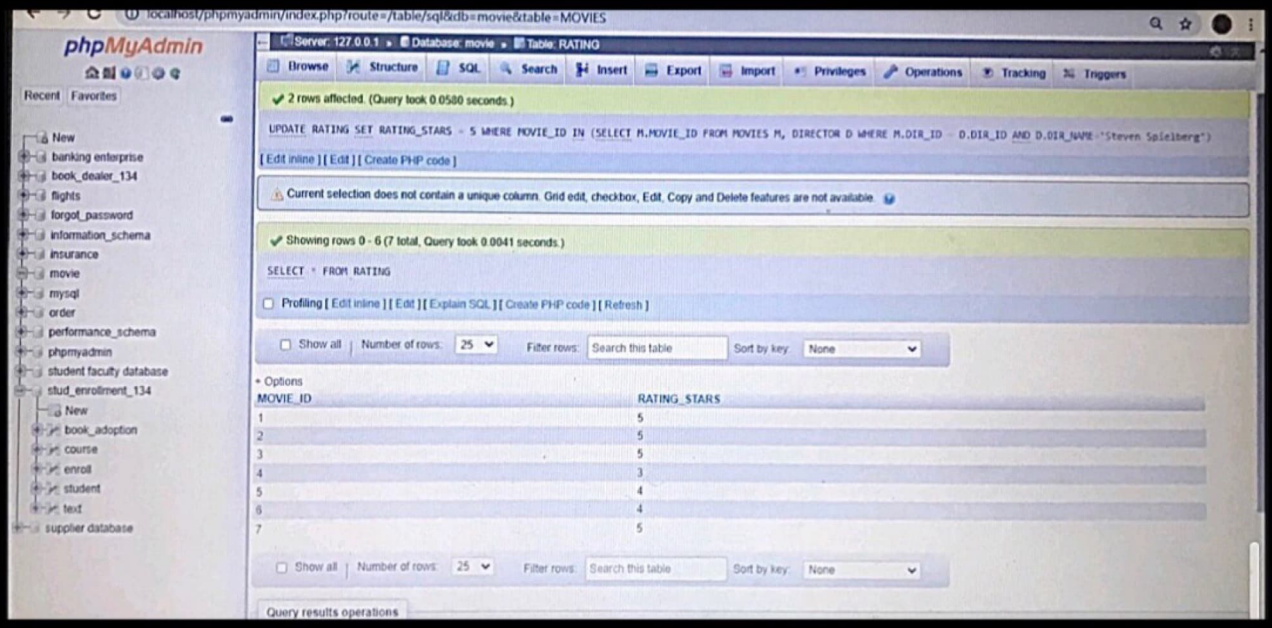
iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).



iv. Find the title of movies and number of stars for each movie that has at least one rating and find the highest
number of stars that movie received. Sort the result by movie title.



v. Update rating of all movies directed by ‘Steven Spielberg’ to 5.



PROGRAM 10:COLLEGE DATABASE

Consider the schema for College Database:

STUDENT(USN, SName, Address, Phone, Gender)

SEMSEC(SSID, Sem, Sec)

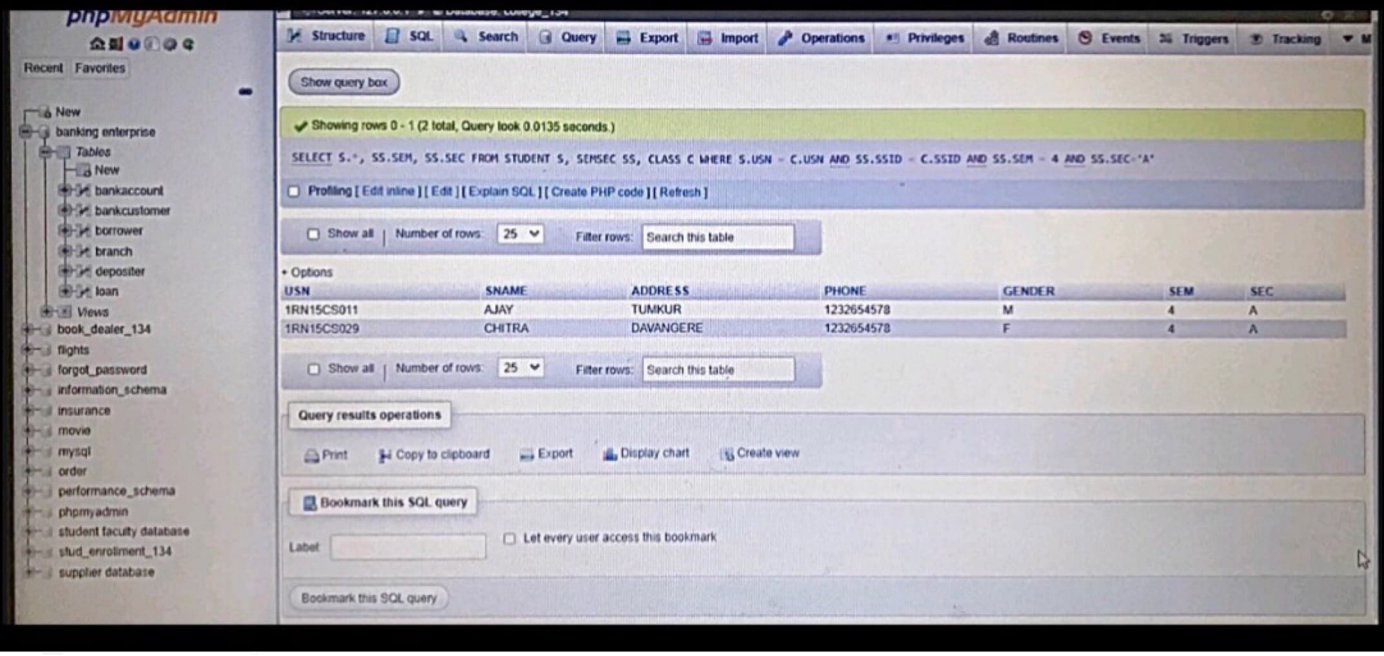
CLASS(USN, SSID)

SUBJECT(Subcode, Title, Sem, Credits)

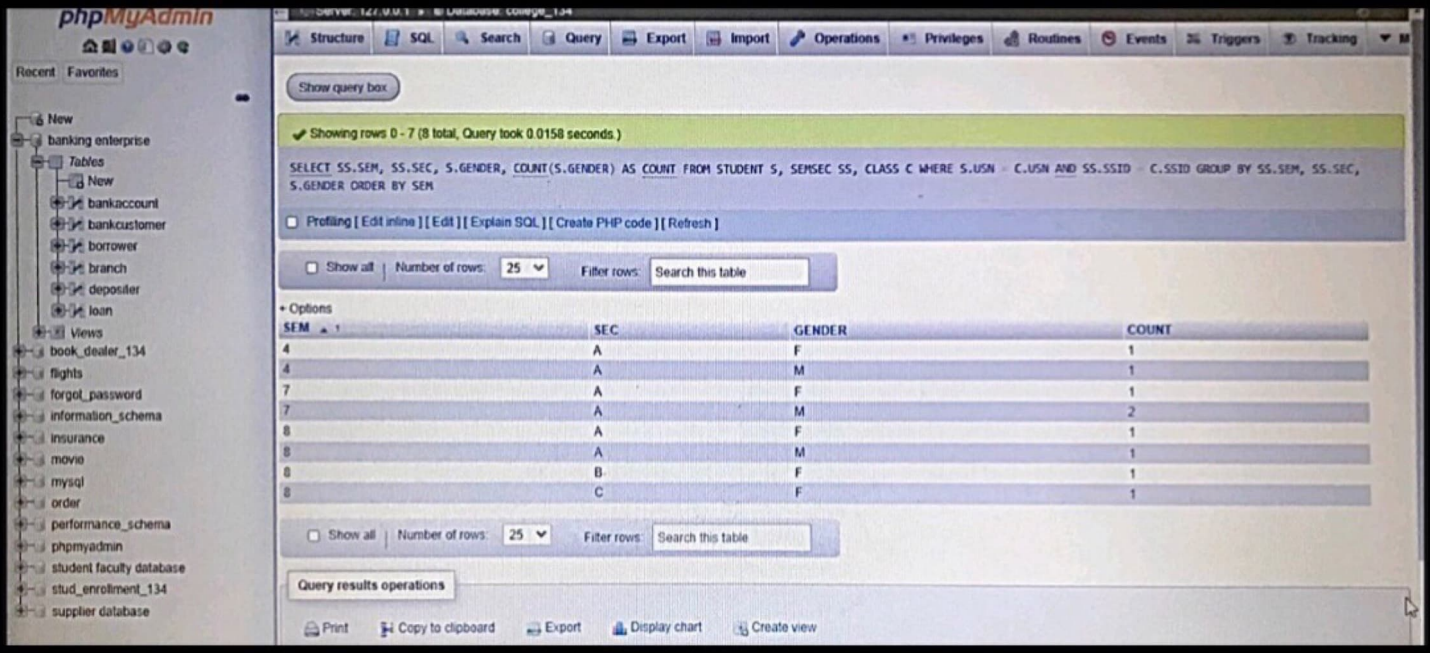
IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)

Write SQL queries to

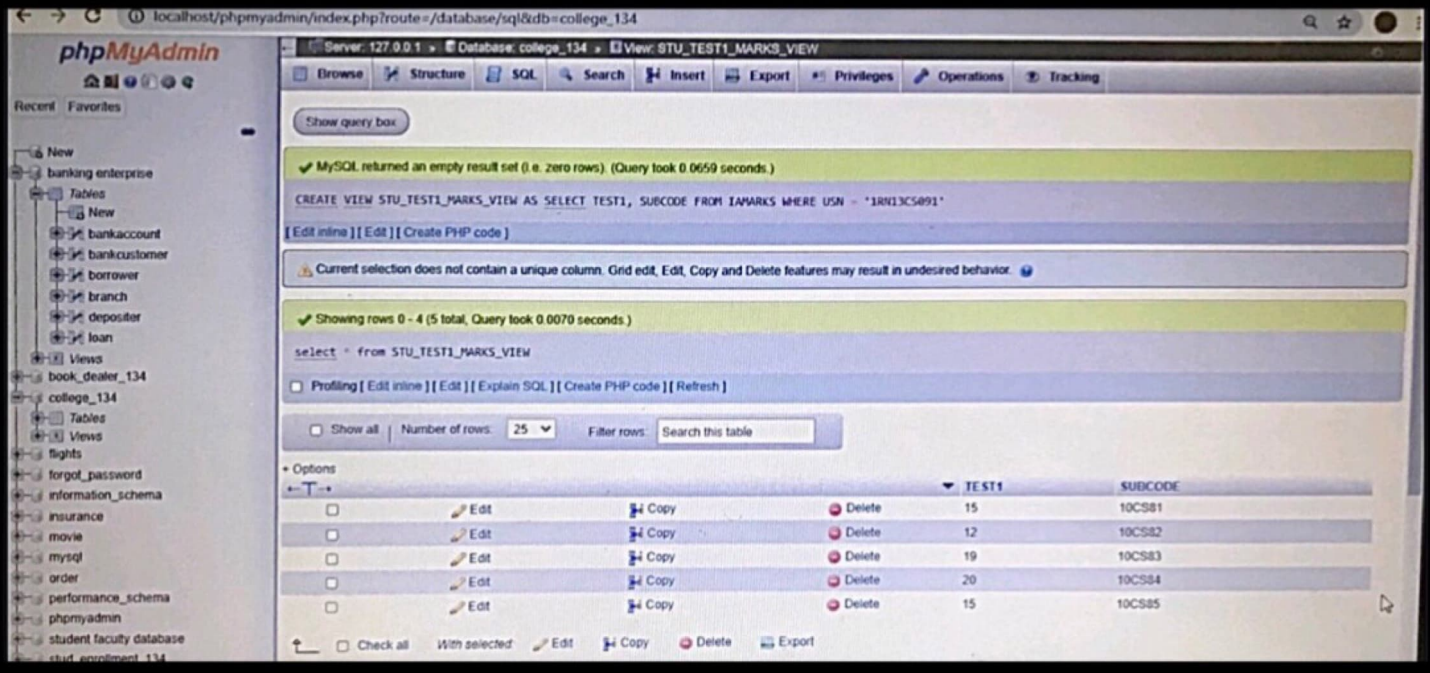
i. List all the student details studying in fourth semester ‘C’ section.



ii. Compute the total number of male and female students in each semester and in each section.



iii. Create a view of Test1 marks of student USN '1BI15CS101' in all subjects.



iv. Calculate the FinallA (average of best two test marks) and update the corresponding table for all students.