

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)
```

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

```
CREATE DATABASE INSURANCE;
```

```
CREATE TABLE
PERSON( DRIVER_ID
VARCHAR(20), NAME
VARCHAR(20), ADDRESS
VARCHAR(20), PRIMARY
KEY(DRIVER_ID));
```

```
CREATE TABLE
CAR( REG_NO
VARCHAR(20), MODEL
VARCHAR(20), YEAR
INT,
PRIMARY KEY(REG_NO));
```

```
CREATE TABLE
ACCIDENT( REPORT_NUM
BER INT, DATE DATE,
LOCATION VARCHAR(20),
PRIMARY KEY(REPORT_NUMBER));
```

```
CREATE TABLE
OWNS( DRIVER_ID
VARCHAR(20), REG_NO
VARCHAR(20), PRIMARY
KEY(DRIVER_ID),
FOREIGN KEY(DRIVER_ID) REFERENCES PERSON(DRIVER_ID));
```

```

CREATE TABLE
PARCIPATED( DRIVER_ID
VARCHAR(20), REG_NO
VARCHAR(20),
REPORT_NUMBER INT,
DAMAGE_AMOUNT INT,
PRIMARY KEY(DRIVER_ID,REG_NO,REPORT_NUMBER),
FOREIGN KEY(DRIVER_ID) REFERENCES PERSON(DRIVER_ID),
FOREIGN KEY(REG_NO) REFERENCES CAR(REG_NO),
FOREIGN KEY(REPORT_NUMBER) REFERENCES ACCIDENT(REPORT_NUMBER));

```

ii. Enter at least five tuples for each relation.

```

INSERT INTO PERSON
VALUES('A1','TYAX','BASAVANGUDI,BANGLORE');
INSERT INTO PERSON
VALUES('A2','SAHANA','GANGAVATHI');
INSERT INTO PERSON
VALUES('A3','CHINMAYEE','MG ROAD,BANGLORE');
INSERT INTO PERSON
VALUES('A4','SINDHU','BAGALKOT');
INSERT INTO PERSON
VALUES('A5','JAYSHREE','HUBLI');

```

```

INSERT INTO CAR(REG_NO,MODEL,YEAR)
VALUES('KA37 0019','MARUTHI SWIFT','2015');
INSERT INTO CAR(REG_NO,MODEL,YEAR)
VALUES('KA37 0018','HYUNDAI VENUE','2016');
INSERT INTO CAR(REG_NO,MODEL,YEAR)
VALUES('KA37 0016','VITARA BREZZA','2017');
INSERT INTO CAR(REG_NO,MODEL,YEAR)
VALUES('KA37 0015','ERTIGA','2018');
INSERT INTO CAR(REG_NO,MODEL,YEAR)
VALUES('KA37 0014','RENAULT TRIBER','2019');

```

```

INSERT INTO ACCIDENT
VALUES('0013','2008-01-21','BANGLORE');
INSERT INTO ACCIDENT
VALUES('0014','2004-07-21','MANGLORE');
INSERT INTO ACCIDENT
VALUES('0015','2008-01-21','BELLARY');
INSERT INTO ACCIDENT

```

```

VALUES('0016','2009-04-21','GANGAVATHI');
INSERT INTO ACCIDENT
VALUES('0012','2007-08-21','KARTAGI');

```

```

INSERT INTO OWNS
VALUES('A1','KA37 0019');
INSERT INTO OWNS
VALUES('A2','KA37 0018');
INSERT INTO OWNS
VALUES('A3','KA37 0016');
INSERT INTO OWNS
VALUES('A4','KA37 0015');
INSERT INTO OWNS
VALUES('A5','KA37 0017');

```

```

INSERT INTO PARTICIPATED ('DRIVER_ID', 'REG_NO', 'REPORT_NUMBER',
'DAMAGE_AMOUNT') VALUES ('A1', 'KA37 0019', '0013', '2000');
INSERT INTO PARTICIPATED ('DRIVER_ID', 'REG_NO', 'REPORT_NUMBER',
'DAMAGE_AMOUNT') VALUES ('A2', 'KA37 0018', '0014', '3000');
INSERT INTO PARTICIPATED ('DRIVER_ID', 'REG_NO', 'REPORT_NUMBER',
'DAMAGE_AMOUNT') VALUES ('A3', 'KA37 0016', '0015', '5000');
INSERT INTO PARTICIPATED ('DRIVER_ID', 'REG_NO', 'REPORT_NUMBER',
'DAMAGE_AMOUNT') VALUES ('A4', 'KA37 0015', '0016', '8000');
INSERT INTO PARTICIPATED ('DRIVER_ID', 'REG_NO', 'REPORT_NUMBER',
'DAMAGE_AMOUNT') VALUES ('A5', 'KA37 0017', '0012', '9000');

```

Recent Favorites

- New
- bankenterprise
- information_schema
- insurance
 - New
 - accident
 - car
 - owns
 - participated
 - person
- mysql
- performance_schema
- phpmyadmin
- studentfaculty
- supplier
- test

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

SELECT * FROM `accident`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	REPORT_NUMBER	DATE	LOCATION
<input type="checkbox"/> Edit	12	2007-08-21	KARTAGI
<input type="checkbox"/> Edit	13	2008-01-21	BANGLORE
<input type="checkbox"/> Edit	14	2004-07-21	MANGLORE
<input type="checkbox"/> Edit	15	2008-01-21	BELLARY
<input type="checkbox"/> Edit	16	2009-04-21	GANGAVATHI
<input type="checkbox"/> Edit	20	2008-01-21	BELLARY

Recent Favorites

- New
- bankenterprise
- information_schema
- insurance
 - New
 - accident
 - car
 - owns
 - participated
 - person
- mysql
- performance_schema
- phpmyadmin
- studentfaculty
- supplier
- test

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

SELECT * FROM `car`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	REG_NO	MODEL	YEAR
<input type="checkbox"/> Edit	KA37 0014	RENAULT TRIBER	2002
<input type="checkbox"/> Edit	KA37 0015	ERTIGA	2008
<input type="checkbox"/> Edit	KA37 0016	VITARA BREZZA	2006
<input type="checkbox"/> Edit	KA37 0018	HYUNDAI VENUE	2005
<input type="checkbox"/> Edit	KA37 0019	MARUTHI SWIFT	2004
<input type="checkbox"/> Edit	KA37 009	RENAULT TRIBER	2004

Showing rows 0 - 4 (5 total, Query took 0.0006 seconds.)

SELECT * FROM `owns`

Number of rows: 25

DRIVER_ID	REG_NO
A1	KA37 0019
A2	KA37 0018
A3	KA37 0016
A4	KA37 0015
A5	KA37 0017

Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

SELECT * FROM `person`

Number of rows: 25

DRIVER_ID	NAME	ADDRESS
A1	TYAX	BASAVANGUDI,BANGLORE
A2	SAHANA	GANGAVATHI
A3	MIKE	PENYA
A4	PREMA	SINGANAL
A5	PREM	KARTAGI

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

SELECT * FROM `participated`

Number of rows: 25

DRIVER_ID	REG_NO	REPORT_NUMBER	DAMAGE_AMOUNT
A1	KA37 0019	13	2000
A2	KA37 0015	15	3000
A3	KA37 0016	16	50000
A4	KA37 0018	14	7000
A5	KA37 0014	12	25000
A5	KA37 009	12	2500

iii. Demonstrate how you

- a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.
- b. Add a new accident to the database.

a.

UPDATE PARTICIPATED

SET DAMAGE_AMOUNT=25000

WHERE REG_NO='KA37 0014' AND REPORT_NUMBER='0012'

1 row affected. (Query took 0.0061 seconds.)

UPDATE PARTICIPATED SET DAMAGE_AMOUNT=25000 WHERE REG_NO='KA37 009' AND REPORT_NUMBER='0012'

b.

INSERT INTO ACCIDENT

VALUES('0020','2008-01-21','BELLARY');



phpMyAdmin

Server: 127.0.0.1 » Database: insurance » Table: accident

Browse Structure SQL Search Insert Export Import Privileges Operations More

Show query box

1 row inserted. (Query took 0.0108 seconds.)

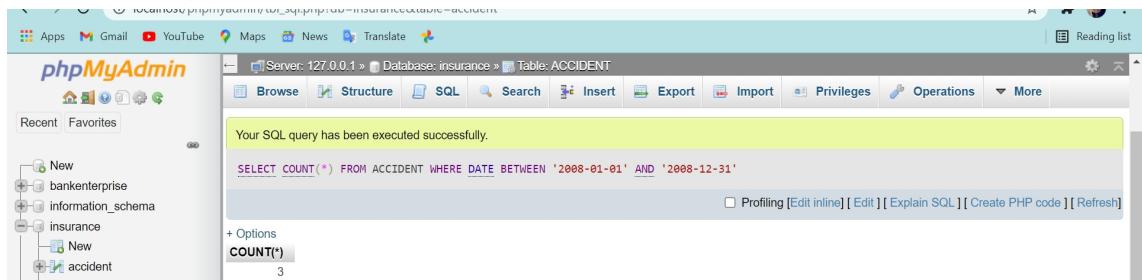
```
INSERT INTO ACCIDENT VALUES('0020','2008-01-21','BELLARY')
```

[Edit inline] [Edit] [Create PHP code]

The screenshot shows the phpMyAdmin interface for a database named 'insurance'. The 'accident' table is selected. In the 'Operations' tab, an 'Insert' query is run: `INSERT INTO ACCIDENT VALUES('0020','2008-01-21','BELLARY')`. A success message indicates 1 row was inserted in 0.0108 seconds.

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

`SELECT COUNT(*) FROM ACCIDENT WHERE DATE BETWEEN '2008-01-01' AND '2008-12-31'`



phpMyAdmin

Server: 127.0.0.1 » Database: insurance » Table: ACCIDENT

Browse Structure SQL Search Insert Export Import Privileges Operations More

Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM ACCIDENT WHERE DATE BETWEEN '2008-01-01' AND '2008-12-31'
```

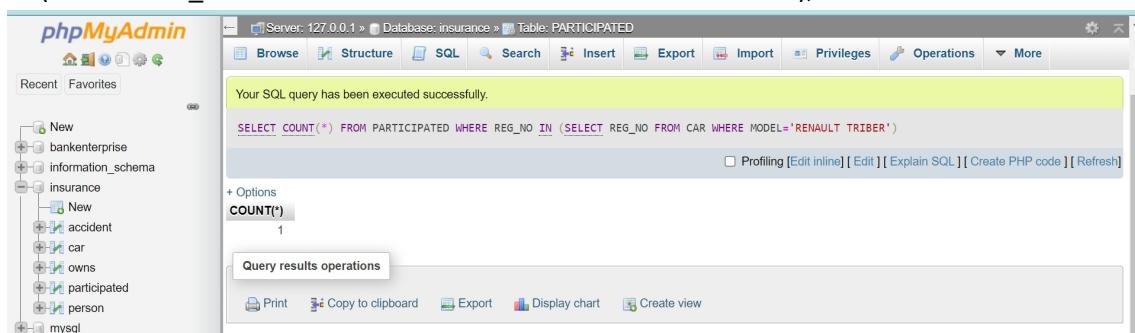
[Profiling] [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

+ Options COUNT(*) 3

The screenshot shows the phpMyAdmin interface for a database named 'insurance'. The 'ACCIDENT' table is selected. In the 'Operations' tab, a query is run: `SELECT COUNT(*) FROM ACCIDENT WHERE DATE BETWEEN '2008-01-01' AND '2008-12-31'`. The result is displayed as a table with one row: COUNT(*) 3.

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

```
SELECT COUNT(*) FROM PARTICIPATED WHERE REG_NO
IN (SELECT REG_NO FROM CAR WHERE MODEL='RENAULT TRIBER');
```



The screenshot shows the phpMyAdmin interface. The left sidebar displays the database structure with the 'insurance' database selected, containing tables: New, accident, car, owns, participated, and person. The main area shows the results of a SQL query:

```
Server: 127.0.0.1 » Database: insurance » Table: PARTICIPATED
Your SQL query has been executed successfully.
SELECT COUNT(*) FROM PARTICIPATED WHERE REG_NO IN (SELECT REG_NO FROM CAR WHERE MODEL='RENAULT TRIBER')
+ Options
COUNT(*)
1
Query results operations
Print Copy to clipboard Export Display chart Create view
```

The query executed is: `SELECT COUNT(*) FROM PARTICIPATED WHERE REG_NO IN (SELECT REG_NO FROM CAR WHERE MODEL='RENAULT TRIBER')`. The result is a single row with `COUNT(*)` set to 1. The 'Query results operations' section includes links for Print, Copy to clipboard, Export, Display chart, and Create view.

LAB 2

BANKING ENTERPRISE

Consider the following database for a banking enterprise.

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

ACCOUNTS (accno: int, branch-name: String, balance: real)

DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)

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

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

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

```
CREATE TABLE BRANCH(  
  BRANCH_NAME VARCHAR(20),  
  BRANCH_CITY VARCHAR(20),  
  ASSETS REAL,  
  PRIMARY KEY(BRANCH_NAME));
```

```
CREATE TABLE BANK_ACCOUNT(  
  ACCNO INT,  
  BRANCH_NAME VARCHAR(20),  
  BALANCE REAL,  
  PRIMARY KEY(ACCNO),  
  FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

```
CREATE TABLE BANK_CUSTOMER(  
  CUSTOMERNAME VARCHAR(20),  
  CUSTOMERSTREET VARCHAR(30),  
  CUSTOMERCITY VARCHAR(30),  
  PRIMARY KEY(CUSTOMERNAME));
```

```
CREATE TABLE DEPOSITOR(  
  CUSTOMERNAME VARCHAR(20),  
  ACCNO INTEGER,  
  PRIMARY KEY(CUSTOMERNAME,ACCNO),  
  FOREIGN KEY(CUSTOMERNAME) REFERENCES BANK_CUSTOMER(CUSTOMERNAME),
```

```
FOREIGN KEY(ACCNO) REFERENCES BANK_ACCOUNT(ACCNO));
```

```
CREATE TABLE LOAN(  
    LOAN_NUMBER INT,  
    BRANCH_NAME VARCHAR(20),  
    AMOUNT REAL,  
    PRIMARY KEY(LOAN_NUMBER),  
    FOREIGN KEY(BRANCH_NAME) REFERENCES BRANCH(BRANCH_NAME));
```

ii. Enter at least five tuples for each relation.

```
INSERT INTO BRANCH  
VALUES('IOB-HANUMANTHNAGAR','BANGLORE','50000'),('IOB-  
GANGAVATHI','GANGAVATHI','20000'),('IOB-JAYANAGAR','BANGLORE','70000'),('IOB-  
GANDHINAGAR','GANGAVATHI','30000'),('IOB-KRISHNANAGAR','GULBURGA','90000')
```

```
INSERT INTO BANK_ACCOUNT  
VALUES('101','IOB-HANUMANTHNAGAR','2000'),('102','IOB-  
GANGAVATHI','2000'),('1021','IOB-JAYANAGAR','3000'),('103','IOB-  
KRISHNANAGAR','4000'),('104','IOB-GANDHINAGAR','5000'),('105','IOB-  
HANUMANTHNAGAR','8000'),('106','IOB-GANDHINAGAR','1000'),('107','IOB-  
HANUMANTHNAGAR','500')
```

```
INSERT INTO bank_customer  
VALUES  
('DARSHAN','THYAGRAJNAGAR','BANGLORE'),('SUDEEP','JAYANAGAR','BANGLORE'),('TYAX','G  
ANDHINAGAR','GANGAVATHI'),('SAHANA','KRISHNANAGAR','GULBURGA'),('SINDHU','PADMA  
NAGAR','BANGLORE'),('JAYSHREE','RAJEEVNAGAR','GANGAVATHI'),('AMOGH','JAYANAGAR','  
BANGLORE')
```

```
INSERT INTO DEPOSITOR  
VALUES('SAHANA','101'),('TYAX','102'),('SUDEEP','1021'),('DARSHAN','103'),('SINDHU','104'),(  
'JAYSHREE','106'),('AMOGH','107')  
INSERT INTO LOAN  
VALUES('1','IOB-HANUMANTHNAGAR','200000'),('2','IOB-JAYANAGAR','100000'),('3','IOB-  
GANDHINAGAR','50000'),('4','IOB-KRISHNANAGAR','300000')
```

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

```
SELECT CUSTOMERNAME FROM DEPOSITER GROUP BY CUSTOMERNAME HAVING  
COUNT(ACCNO)>=2;
```



Recent Favorites

Show query box

SELECT CUSTOMERNAME FROM DEPOSITER GROUP BY CUSTOMERNAME HAVING COUNT(ACCNO)>=2;

AMOGH

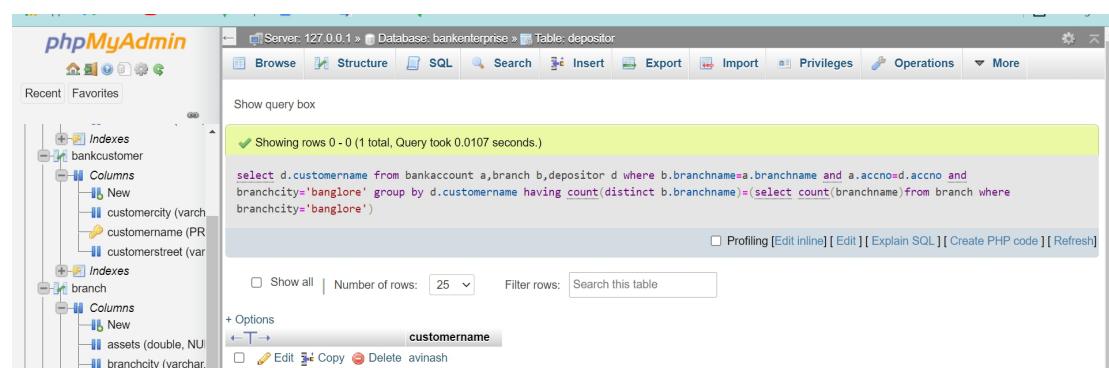
JAYSHREE

Customername

Check all With selected: Edit Copy Delete

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

```
select d.customername from bankaccount a,branch b,depositor d where b.branchname=a.branchname  
and a.accno=d.accno and branchcity='banglore' group by d.customername having count(distinct b.branchname)=  
(select count(branchname)from branch where branchcity='banglore')
```



Show query box

select d.customername from bankaccount a,branch b,depositor d where b.branchname=a.branchname and a.accno=d.accno and branchcity='banglore' group by d.customername having count(distinct b.branchname)=(select count(branchname)from branch where branchcity='banglore')

avinash

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

```
delete from bankaccount where branchname in(select branchname from branch where branchcity='banglore')
```



LAB-3

SUPPLIERS 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:

```
create database supplier ;
create table suppliers(
sid int,
sname varchar(20) not null,
address varchar(20) not null,
primary key(sid));
```

```
create table parts(
pid int,
pname varchar(20) not null,
color varchar(20) not null,
primary key(pid));
```

```
create table catalog(
sid int,
pid int,
cost real,
primary key(sid,pid),
foreign key(sid) references suppliers(sid) on delete cascade on update cascade,
foreign key(pid) references parts(pid) on delete cascade on update cascade);
```

```
insert into suppliers
values
('1','axe','banglore'),
('2','ben','gangavathi'),
('3','chin','banglore south'),
('4','den','gulburga'),
('5','eleven','bellary');
```

```
insert into parts
values('11','A','red'),
```

```

('12','B','green'),
('13','C','red'),
('14','D','yellow'),
('15','E','orange');

```

```

insert into catalog
values('1','11','1000'),
('2','12','2000'),
('3','13','3000'),
('4','14','4000'),
('5','15','5000');

```

The image shows three separate phpMyAdmin database tables:

- parts** table (pid, pname, color):

pid	pname	color
11	A	red
12	B	green
13	C	red
14	D	yellow
15	E	orange

- suppliers** table (sid, sname, address):

sid	sname	address
1	axe	banglore
2	ben	gangavathi
3	chin	banglore south
4	den	gulburga
5	eleven	bellary

- catalog** table (c.pid, c.sid):

c.pid	c.sid
11	1
12	2
13	3
14	4
15	5

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

```
Select distinct p.name from parts p,catalog c where p.pid=c.pid;
```

The screenshot shows the phpMyAdmin interface with the following details:

- Server: 127.0.0.1 > Database: supplier > Table: parts**
- Query:** select distinct p.pname from parts p,catalog c where p.pid=c.pid
- Result:** Showing rows 0 - 4 (5 total, Query took 0.0034 seconds.)
- Table Data:** A table showing the distinct part names (A, B, C, D, E) from the joined parts and catalog tables.

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

```
Select s.sname from suppliers s where not exists ((select p.pid from parts p ) except (select c.pid from catalog c where c.sid=s.sid));
```

Server: 127.0.0.1 > Database: supplier > Table: suppliers

Show query box

```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0099 seconds.)
```

```
select s.sname from suppliers s where not exists ( ( select p.pid from parts p ) except ( select c.pid from catalog c where c.sid = s.sid ) )
```

lname

Query results operations

Create view

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

```
select s.sname from suppliers s where s.sid in select c.sid from catalog c where c.pid
in(select p.pid from parts p where p.color='red'));
```

Server: 127.0.0.1 > Database: supplier > Table: suppliers

Show query box

```
Showing rows 0 - 1 (2 total, Query took 0.0067 seconds.)
```

```
select s.sname from suppliers s where s.sid in(select c.sid from catalog c where c.pid in(select p.pid from parts p where p.color = 'red'))
```

lname

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

lname

axe chin

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

```
Select p.pname from parts p where p.pid in (select c.pid from catalog c where c.sid in(select
s.sid from suppliers s where s.sname='axe'));
```

Server: 127.0.0.1 > Database: supplier > Table: parts

Show query box

```
Showing rows 0 - 0 (1 total, Query took 0.0061 seconds.)
```

```
select p.pname from parts p where p.pid in(select c.pid from catalog c where c.sid in(select s.sid from suppliers s where
s.sname='axe'))
```

lname

A

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

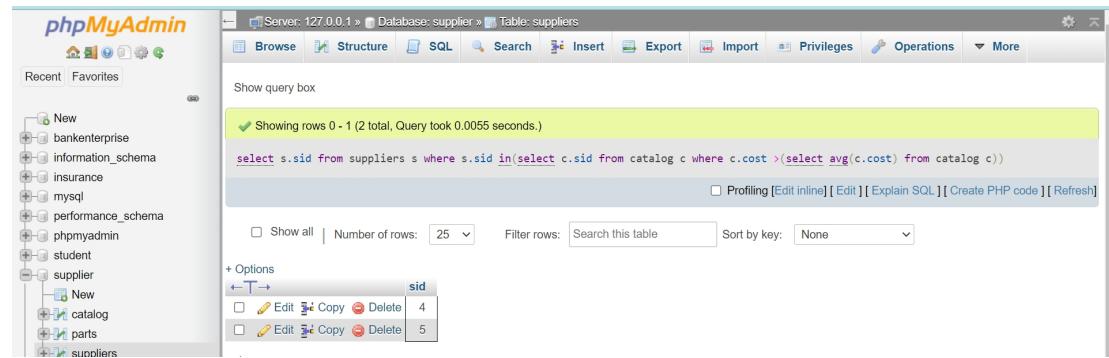
lname

A

Check all With selected: Edit Copy Delete Export

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).

```
Select s.sid from suppliers s where s.sid in(select c.sid from catalog c where c.cost>(select avg(c.cost) from catalog c);
```



The screenshot shows the phpMyAdmin interface for the 'supplier' database. The 'Table: suppliers' page is displayed. The query results are shown in a table with columns 'sid' and 'Number of rows' (25). The results are:

sid
4
5

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

```
Select p.pname ,s.sname from parts p ,suppliers s ,catalog c where c.pid =p.pid and c.sid=s.sid and c.cost = (select max(c.cost) from catalog c where c.pid=p.pid);
```



The screenshot shows the phpMyAdmin interface for the 'supplier' database. The 'Table: suppliers' page is displayed. The query results are shown in a table with columns 'pname' and 'sname'. The results are:

pname	sname
A	axe
B	ben
C	chin
D	den
E	eleven

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

```
SELECT DISTINCT c.sid FROM catalog c WHERE NOT EXISTS ( SELECT *FROM parts p WHERE p.pid = c.pid AND p.color = 'red' )
```

phpMyAdmin

Server: 127.0.0.1 > Database: supplier > Table: catalog

Recent Favorites

New

- bankenterprise
- information_schema
- insurance
- mysql
- performance_schema
- phpmyadmin
- student
- supplier
- New
- catalog
- parts
- suppliers

Browse Structure SQL Search Insert Export Import Privileges Operations More

Show query box

Showing rows 0 - 2 (3 total, Query took 0.00069 seconds.)

```
SELECT DISTINCT c.sid FROM catalog c WHERE NOT EXISTS ( SELECT * FROM parts p WHERE p.pid = c.pid AND p.color = 'red' )
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

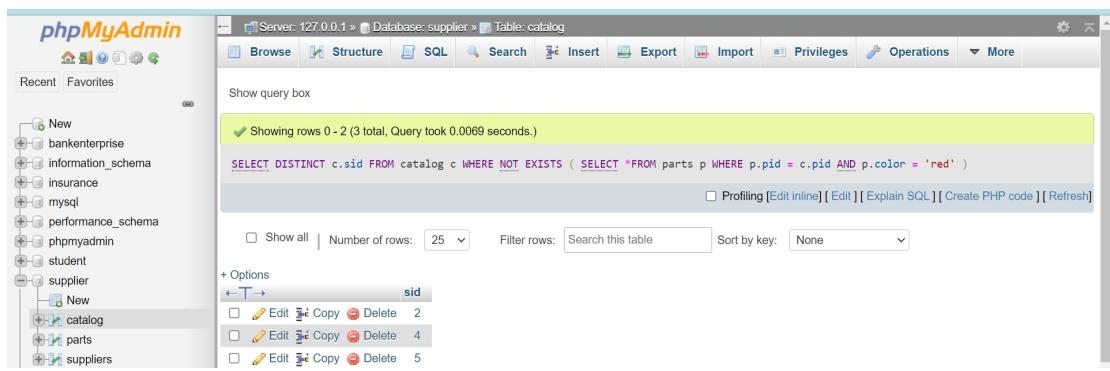
+ Options

sid
2
4
5

Edit Copy Delete 2

Edit Copy Delete 4

Edit Copy Delete 5



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)

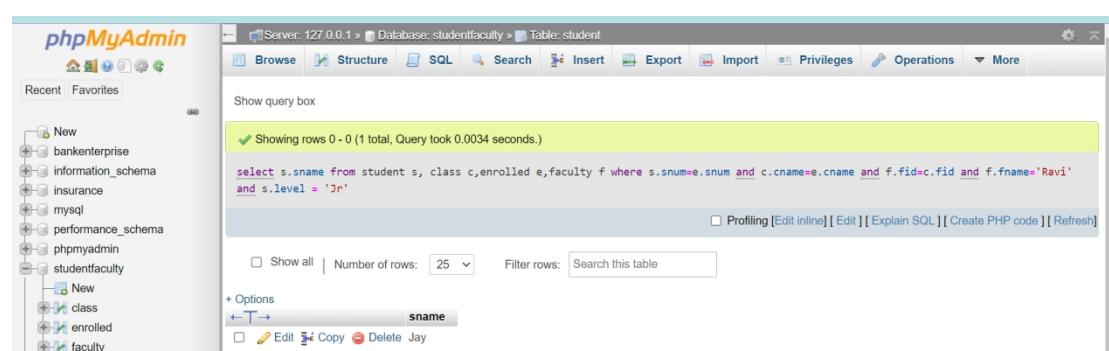
FACULTY (fid: integer, fname: string, deptid: integer)

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

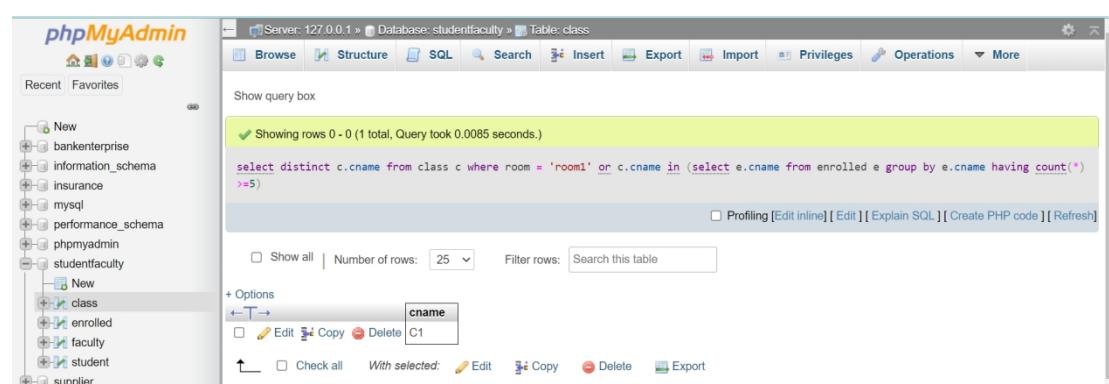
```
select s.sname from student s, class c, enrolled e, faculty f where s.snum=e.snum and  
c cname=e cname and f fid=c fid and f fname='Ravi' and s level = 'Jr'
```



sname
Jay

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

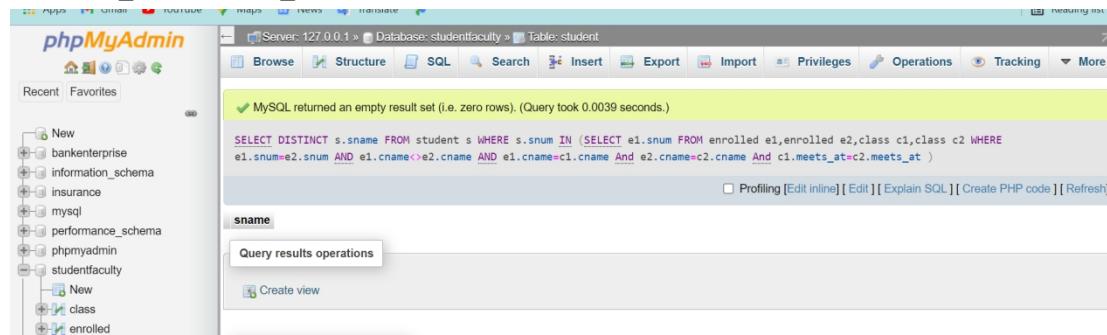
```
select distinct c cname from class c where room = 'room1' or c cname in (select e cname  
from enrolled e group by e cname having count(*) >=5)
```



cname
C1

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

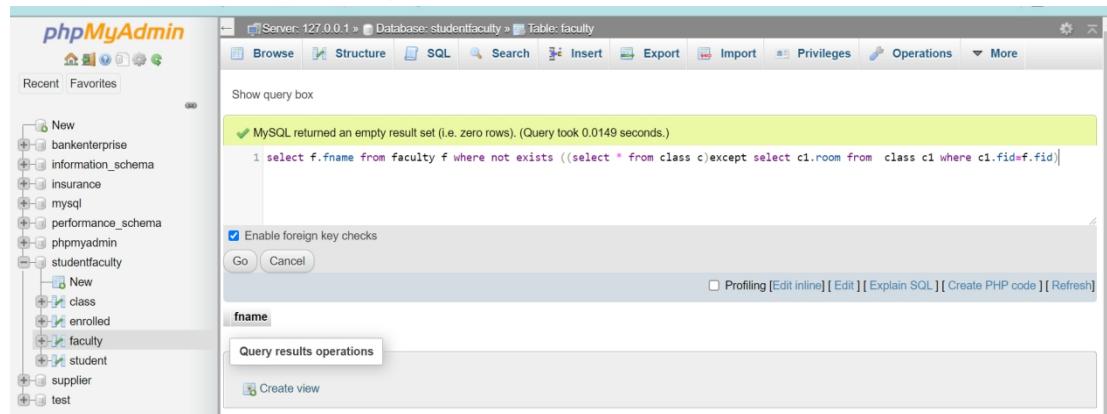
```
SELECT DISTINCT s.sname FROM student s WHERE s.snum IN (SELECT e1.snum FROM enrolled e1,enrolled e2,class c1,class c2 WHERE e1.snum=e2.snum AND e1 cname<>e2 cname AND e1 cname=c1 cname And e2 cname=c2 cname And c1 meets_at=c2 meets_at )
```



The screenshot shows the phpMyAdmin interface for a database named 'studentfaculty'. The 'student' table is selected. The SQL query entered is the one provided in the text above. The results pane shows a message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0039 seconds.)'. The query itself is displayed in the SQL tab.

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

```
select f.fname from faculty f where not exists ((select * from class c)except select c1.room from class c1 where c1.fid=f.fid)
```



The screenshot shows the phpMyAdmin interface for a database named 'studentfaculty'. The 'faculty' table is selected. The SQL query entered is the one provided in the text above. The results pane shows a message: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0149 seconds.)'. The query itself is displayed in the SQL tab. The 'Show query box' button is visible at the top left.

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

```
select distinct f.fname from faculty f where 5 >(select count(e.snum) from class c ,enrolled e where c cname = e cname and c.fid= f.fid )
```

Showing rows 0 - 4 (5 total, Query took 0.0125 seconds.)

```
select distinct f.fname from faculty f where 5 >(select count(e.snum) from class c ,enrolled e where c.cname = e cname and c.fid= f.fid)
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	fname
<input type="checkbox"/>	Ravi
<input type="checkbox"/>	Chirag
<input type="checkbox"/>	Kumar
<input type="checkbox"/>	Savitha
<input type="checkbox"/>	Sam

vi. Find the names of students who are not enrolled in any class.

select distinct s.sname from student s where s.snum not in (select e.snum from enrolled e)

Showing MySQL returned an empty result set (i.e. zero rows). (Query took 0.0067 seconds.)

```
select distinct s.sname from student s where s.snum not in (select e.snum from enrolled e )
```

Query results operations

Create view

Bookmark this SQL query

vii. For each age value that appears in Students, find the level value that appears most often. For example, if there are more FR level students aged 18 than SR, JR, or SO students aged 18, you should print the pair (18, FR).

SELECT S.age, S.level FROM Student S GROUP BY S.age, S.level HAVING S.level IN (SELECT S1.level FROM Student S1 WHERE S1.age = S.age GROUP BY S1.level, S1.age HAVING COUNT(*) >= ALL (SELECT COUNT(*) FROM Student S2 WHERE S1.age = S2.age GROUP BY S2.level, S2.age))

	age	level
<input type="checkbox"/>	19	jr
<input type="checkbox"/>	20	jr
<input type="checkbox"/>	21	sr

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, fname: 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.

```
create database airline_flight
```

```
create table flights( flno int, fromplace varchar(15), toplace varchar(15), distance int, departs datetime, arrives datetime, price int, primary key (flno))
```

```
create table aircraft( aid int, fname varchar(15), cruisingrange int, primary key (aid) )
```

```
create table employees ( eid int, ename varchar(15), salary int, primary key (eid) )
```

```
create table certified ( eid int, aid int, foreign key (eid) references employees(eid), foreign key (aid) references aircraft(aid) )
```

```
insert into flights values(101, 'Bangalore', 'Delhi', 2500, '2005-05-13 07:15:31', '2005-05-13 18:15:31', 5000);
insert into flights values(102, 'Bangalore', 'Lucknow', 3000, '2013-05-05 07:15:31', '2013-05-05 11:15:31', 6000);
insert into flights values(103, 'Lucknow', 'Delhi', 500, '2013-05-05 12:15:31', '2013-05-05 17:15:31', 3000);
insert into flights values(107, 'Bangalore', 'Frankfurt', 8000, '2013-05-05 07:15:31', '2013-05-05 22:15:31', 60000);
insert into flights values(104, 'Bangalore', 'Frankfurt', 8500, '2013-05-05 07:15:31', '2013-05-05 23:15:31', 75000);
insert into flights values(105, 'Kolkata', 'Delhi', 3400, '2013-05-05 07:15:31', '2013-05-05 09:15:31', 7000);
insert into flights values(106, 'Bangalore', 'Kolkata', 1000, '2013-05-05 01:15:30', '2013-05-05 09:20:30', 10000);
insert into flights values(108, 'Lucknow', 'Kolkata', 1000, '2013-05-05 11:30:30', '2013-05-05 15:20:30', 10000);
```

```
insert into aircraft values(101, '747', 3000);
insert into aircraft values(102, 'Boeing', 900);
insert into aircraft values(103, '647', 800);
insert into aircraft values(104, 'Dreamliner', 10000);
insert into aircraft values(105, 'Boeing', 3500);
insert into aircraft values(106, '707', 1500);
insert into aircraft values(107, 'Dream', 120000);
insert into aircraft values(108, '707', 760);
```

```
insert into aircraft values(109, '747', 1000);
```

```
insert into employees values(701, 'A', 50000);
insert into employees values(702, 'B', 100000);
insert into employees values(703, 'C', 150000);
insert into employees values(704, 'D', 90000);
insert into employees values(705, 'E', 40000);
insert into employees values(706, 'F', 60000);
insert into employees values(707, 'G', 90000);
```

```
insert into certified values(701, 101);
insert into certified values(701, 102);
insert into certified values(701, 106);
insert into certified values(701, 105);
insert into certified values(702, 104);
insert into certified values(703, 104);
insert into certified values(704, 104);
insert into certified values(702, 107);
insert into certified values(703, 107);
insert into certified values(704, 107);
insert into certified values(702, 101);
insert into certified values(702, 108);
insert into certified values(701, 109);
```

The screenshot shows the phpMyAdmin interface with the following details:

Database Structure: The sidebar shows the database structure with the following tables under the `airline_flight` schema:

- `aircraft`
- `certified`
- `employees`
- `flights`

flights Table: The main table view shows the following data:

	fino	fromplace	toplace	distance	departs	arrives	price
101	101	Bangalore	Delhi	2500	2005-05-13 07:15:31	2005-05-13 18:15:31	5000
102	102	Bangalore	Lucknow	3000	2013-05-05 07:15:31	2013-05-05 11:15:31	6000
103	103	Lucknow	Delhi	500	2013-05-05 12:15:31	2013-05-05 17:15:31	3000
104	104	Bangalore	Frankfurt	8500	2013-05-05 07:15:31	2013-05-05 23:15:31	75000
105	105	Kolkata	Delhi	3400	2013-05-05 07:15:31	2013-05-05 09:15:31	7000
106	106	Bangalore	Kolkata	1000	2013-05-05 01:15:30	2013-05-05 09:20:30	10000
107	107	Bangalore	Frankfurt	8000	2013-05-05 07:15:31	2013-05-05 22:15:31	60000
108	108	Lucknow	Kolkata	1000	2013-05-05 11:30:30	2013-05-05 15:20:30	10000

aircraft Table: The sidebar shows the following data for the `aircraft` table:

eid	aid
701	101
701	102
701	106
701	105
702	104
703	104
704	104
702	107
703	107
704	107
702	101
702	108
701	109

Employees Table Data:

	eid	ename	salary
1	701	A	50000
2	702	B	100000
3	703	C	150000
4	704	D	90000
5	705	E	40000
6	706	F	60000
7	707	G	90000

Aircraft Table Data:

	aid	aname	cruisingrange
1	101	747	3000
2	102	Boeing	900
3	103	647	800
4	104	Dreamliner	10000
5	105	Boeing	3500
6	106	707	1500
7	107	Dream	120000
8	108	707	760
9	109	747	1000

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

```
select distinct a.aname from aircraft a where a.aid in (
    select c.aid from certified c, employees e where
    c.eid = e.eid and not exists(
        select * from employees e1 where e1.eid=e.eid and e1.salary<80000
    )
);
```

Aircraft Table Data:

	aname
1	747
2	Dreamliner
3	Dream
4	707

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.

```
select max(a.cruisingrange), c.eid from certified c, aircraft a where c.aid = a.aid group
by c.eid having count(c.eid)>3;
```

Showing rows 0 - 1 (2 total, Query took 0.0065 seconds.)

```
select max(a.cruisingrange), c.eid from certified c, aircraft a where c.aid = a.aid group by c.eid having count(c.eid)>3
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

max(a.cruisingrange)	eid
3500	701
120000	702

iii. find the names of pilots whose salary is less than the price of the cheapest route from bengaluru to frankfurt.

select ename from employees where salary < (select min(price) from flights where fromplace='Bangalore' and toplace='Frankfurt');

Showing rows 0 - 1 (2 total, Query took 0.0065 seconds.)

```
select ename from employees where salary < ( select min(price) from flights where fromplace='Bangalore' and toplace='Frankfurt')
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

ename
A
E

Check all | With selected: Edit Copy Delete | Edit Copy Delete Export

iv. 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.

select avg(e.salary), c.aid from certified c, employees e where c.aid in (select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by c.aid;

Showing rows 0 - 4 (5 total, Query took 0.0107 seconds.)

```
select avg(e.salary), c.aid from certified c, employees e where c.aid in ( select aid from aircraft where cruisingrange>1000) and e.eid = c.eid group by c.aid
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

avg(e.salary)	aid
75000.0000	101
11333.3333	104
50000.0000	105
50000.0000	106
11333.3333	107

v. find the names of pilots certified for some boeing aircraft.

select ename from employees where eid in (select eid from certified where aid in (select aid from aircraft where aname = 'Boeing'));

vi. find the aids of all aircraft that can be used on routes from bengaluru to new delhi.

select aname from aircraft where cruisingrange > any (select distance from flights where fromplace='Bangalore' and toplace='Delhi');

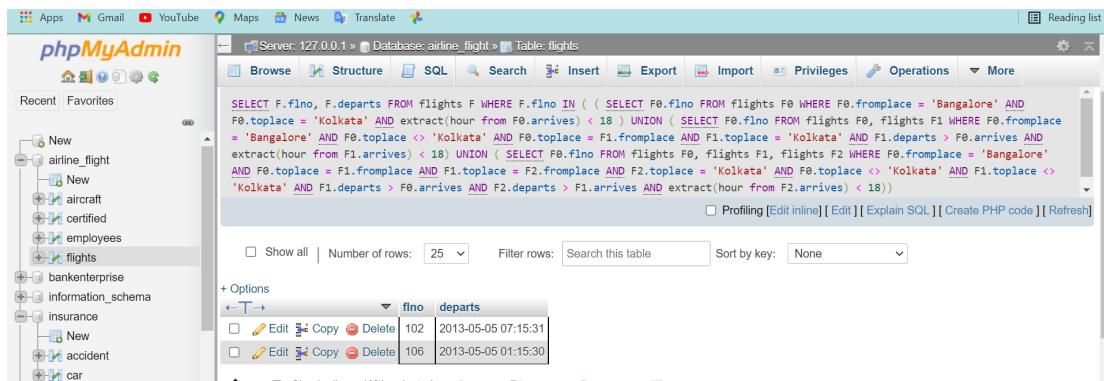
vii. a customer wants to travel from madison to new york with no more than two changes of flight. list the choice of departure times from madison if the customer wants to arrive in new york by 6 p.m.

```

SELECT F.flno, F.deperts
FROM flights F
WHERE F.flno IN ( ( SELECT F0.flno
FROM flights F0
WHERE F0.fromplace = 'Bangalore' AND F0.toplace = 'Kolkata'
AND extract(hour from F0.arrives) < 18 )
UNION
( SELECT F0.flno
FROM flights F0, flights F1
WHERE F0.fromplace = 'Bangalore' AND F0.toplace <> 'Kolkata'
AND F0.toplace = F1.fromplace AND F1.toplace = 'Kolkata'
AND F1.deperts > F0.arrives
AND extract(hour from F1.arrives) < 18 )
UNION
( SELECT F0.flno
FROM flights F0, flights F1, flights F2
WHERE F0.fromplace = 'Bangalore'
AND F0.toplace = F1.fromplace
AND F1.toplace = F2.fromplace
AND F2.toplace = 'Kolkata'
AND F0.toplace <> 'Kolkata'
AND F1.toplace <> 'Kolkata'

```

AND F1.deploys > F0.arrives
AND F2.deploys > F1.arrives
AND extract(hour from F2.arrives) < 18);



The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel:** Shows the database structure with a tree view. The 'flights' table is selected.
- Top Bar:** Shows the URL as 'Server: 127.0.0.1 > Database: airline_flight > Table: flights' and a 'Reading list' icon.
- Toolbar:** Includes 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'More'.
- SQL Tab:** Displays a complex multi-table query involving flights F, F0, and F1. The query filters flights from Bangalore to Kolkata with a 18-hour arrival window, and then finds flights from Kolkata to Bangalore with a 18-hour arrival window. It also includes a condition where the arrival hour of the second flight is less than 18.
- Browse Tab:** Shows the results of the query in a table format. The table has columns 'fno' and 'deploys'.
- Table Data:**

fno	deploys
102	2013-05-05 07:15:31
106	2013-05-05 01:15:30