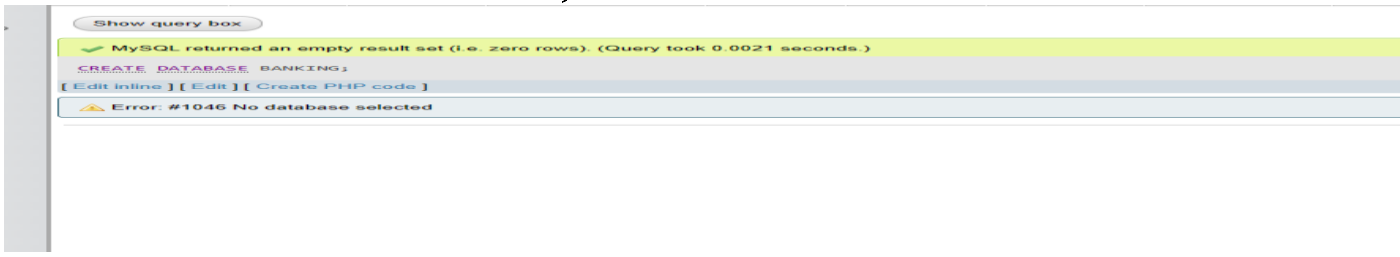
PRACTICALS

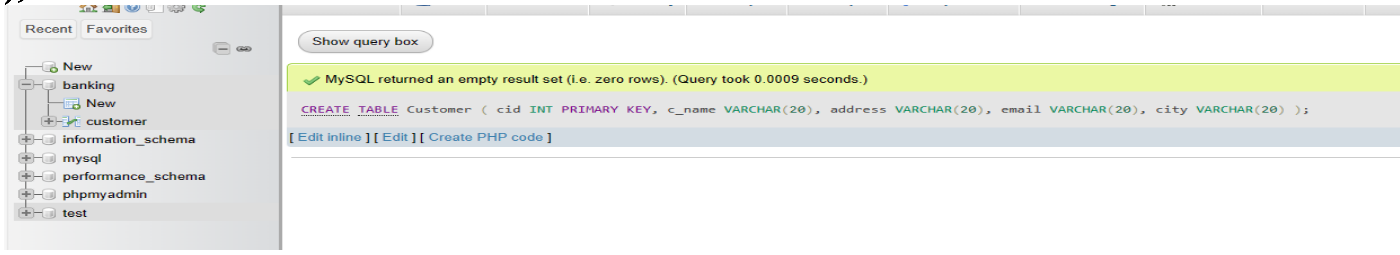
1. Create a database BANKING with the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model.

CREATE DATABASE BANKING;



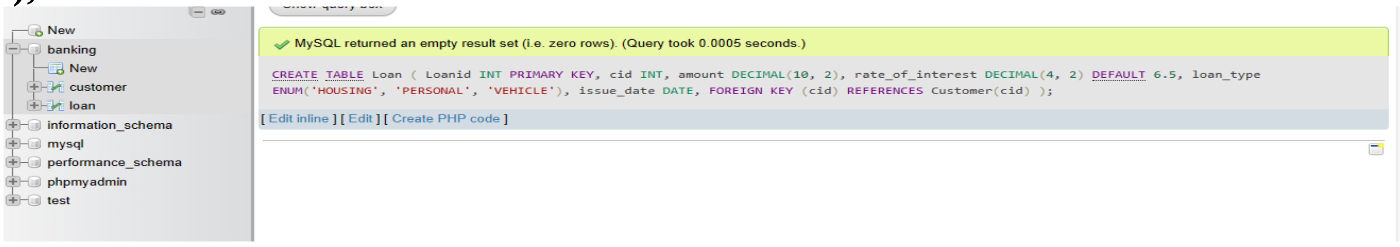
```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)
CREATE DATABASE BANKING;
[ Edit inline ] [ Edit ] [ Create PHP code ]
Error: #1046 No database selected
```

```
CREATE TABLE Customer (
   cid INT PRIMARY KEY,
   c_name VARCHAR(20),
   address VARCHAR(20),
   email VARCHAR(20),
   city VARCHAR(20)
);
```



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)
CREATE TABLE Customer ( cid INT PRIMARY KEY, c_name VARCHAR(20), address VARCHAR(20), email VARCHAR(20), city VARCHAR(20) );
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

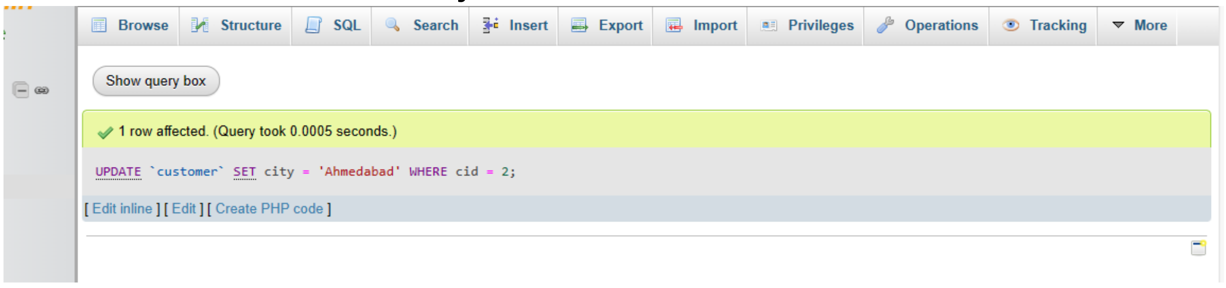
```
CREATE TABLE Loan (
   Loanid INT PRIMARY KEY,
   cid INT,
   amount DECIMAL(10, 2),
   rate_of_interest DECIMAL(4, 2) DEFAULT 6.5,
   loan_type ENUM('HOUSING', 'PERSONAL','VEHICLE'),
   issue_date DATE,
   FOREIGN KEY (cid) REFERENCES Customer(cid)
 );
```



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)
CREATE TABLE Loan ( Loanid INT PRIMARY KEY, cid INT, amount DECIMAL(10, 2), rate_of_interest DECIMAL(4, 2) DEFAULT 6.5, loan_type
ENUM('HOUSING', 'PERSONAL', 'VEHICLE'), issue_date DATE, FOREIGN KEY (cid) REFERENCES Customer(cid) );
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Perform the following queries

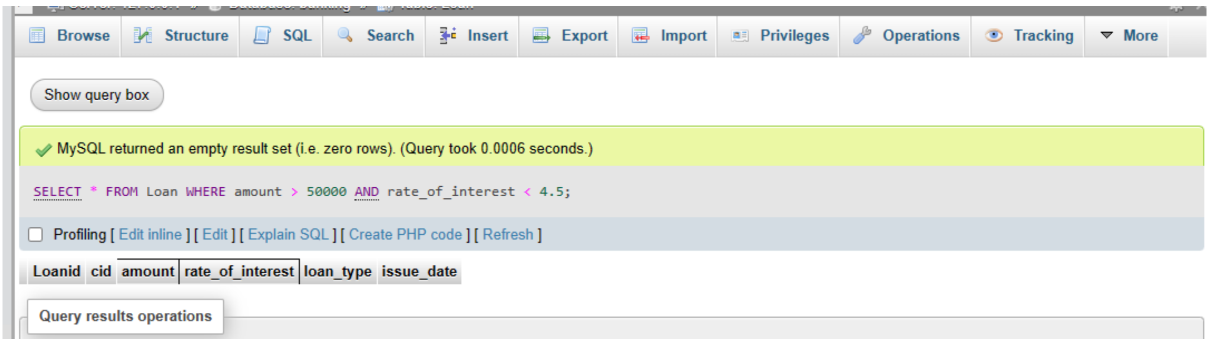
a) Update the city of cid=2 to "Ahmedabad"
UPDATE customers SET city = 'Ahmedabad' WHERE cid = 2;



```
1 row affected. (Query took 0.0005 seconds.)

UPDATE `customer` SET city = 'Ahmedabad' WHERE cid = 2;

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

b) Display the loan details of customers where loan amount is > 50,000 and rate_of_interest is less than 4.5%,

SELECT * FROM loan WHERE loan_amount > 50000 AND rate_of_interest < 4.5;

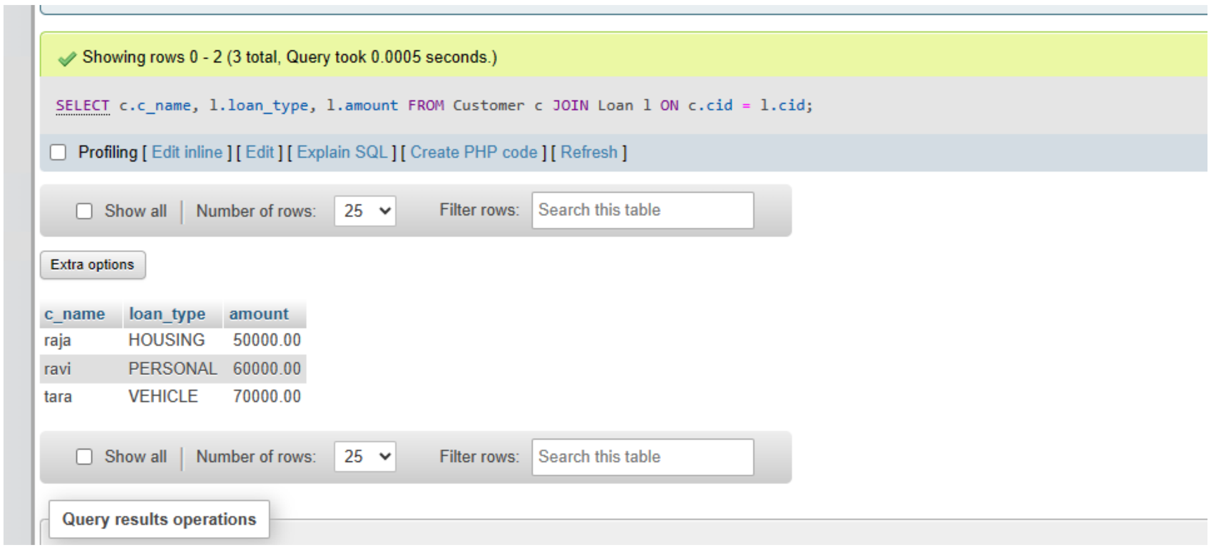


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

SELECT * FROM Loan WHERE amount > 50000 AND rate_of_interest < 4.5;

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

Loanid	cid	amount	rate_of_interest	loan_type	issue_date

C) Display customer name, loan type and loan amount.

SELECT c.c_name, l.loan_type, l.amount
FROM Customer c
JOIN Loan l ON c.cid = l.cid;

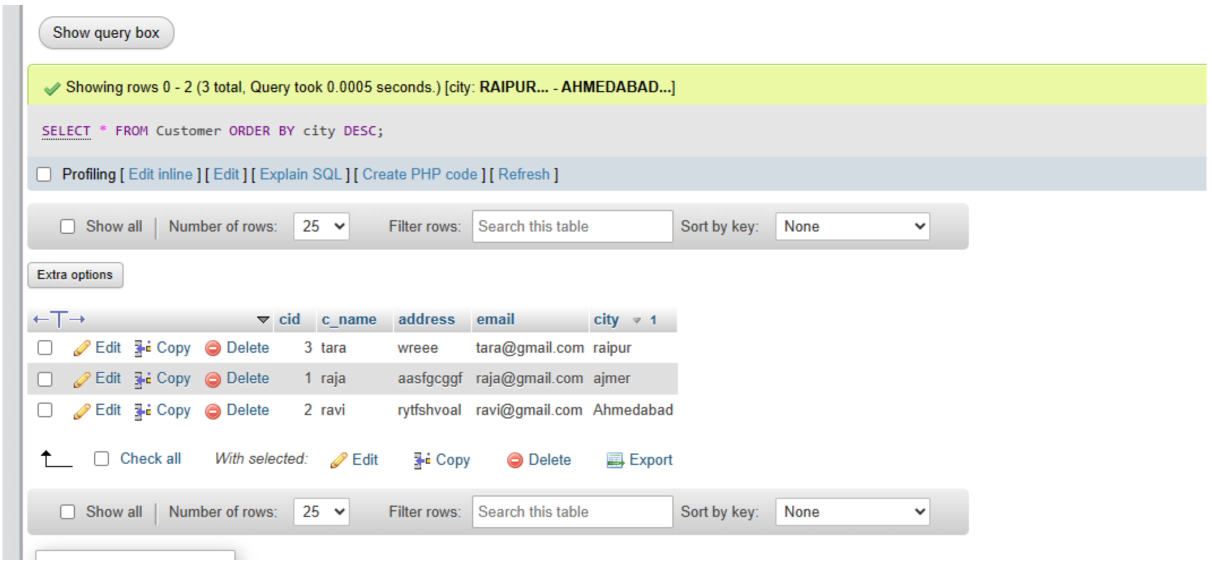

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT c.c_name, l.loan_type, l.amount FROM Customer c JOIN Loan l ON c.cid = l.cid;
```

c_name	loan_type	amount
raja	HOUSING	50000.00
ravi	PERSONAL	60000.00
tara	VEHICLE	70000.00

e) Arrange the records of customer in descending order of city.

SELECT * FROM Customer ORDER BY city DESC;

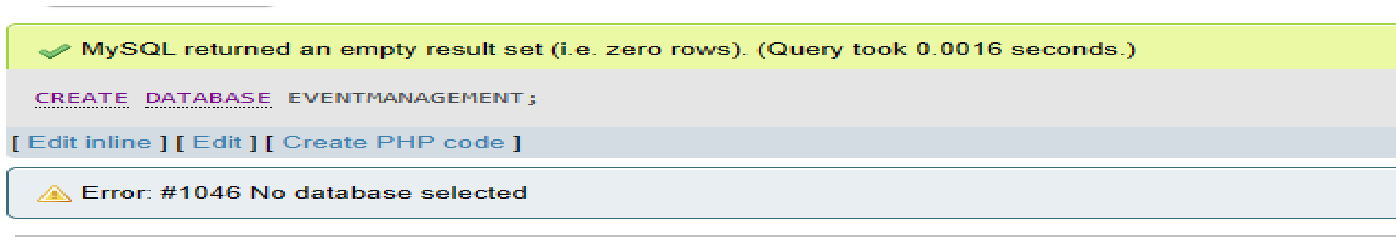

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.) [city: RAIPUR... - AHMEDABAD...]

```
SELECT * FROM Customer ORDER BY city DESC;
```

cid	c_name	address	email	city
3	tara	wreee	tara@gmail.com	raipur
1	raja	aasfgcggf	raja@gmail.com	ajmer
2	ravi	rytfshvoal	ravi@gmail.com	Ahmedabad

2. Create a database EVENT_MANAGEMENT with the following tables
Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model.

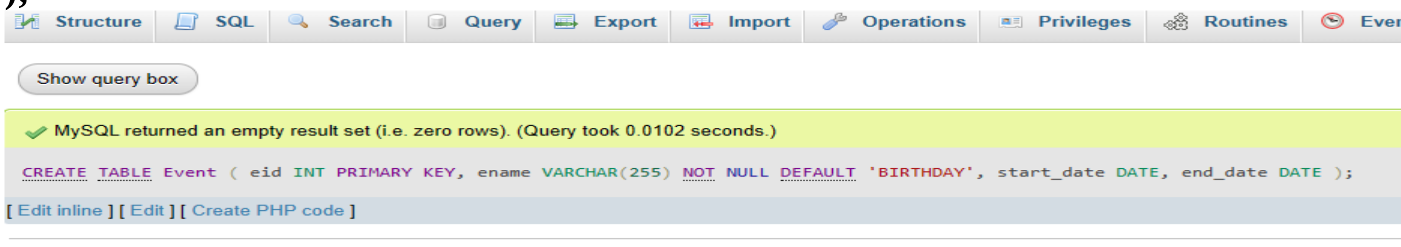
CREATE DATABASE EVENTMANAGEMENT;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0016 seconds.)
CREATE DATABASE EVENTMANAGEMENT;
[ Edit inline ] [ Edit ] [ Create PHP code ]
Error: #1046 No database selected
```

Event(eid,ename, start_date,end_date)

```
CREATE TABLE Event (
   eid INT PRIMARY KEY,
   ename VARCHAR(255) NOT NULL DEFAULT 'BIRTHDAY',
   start_date DATE,
   end_date DATE
);
```

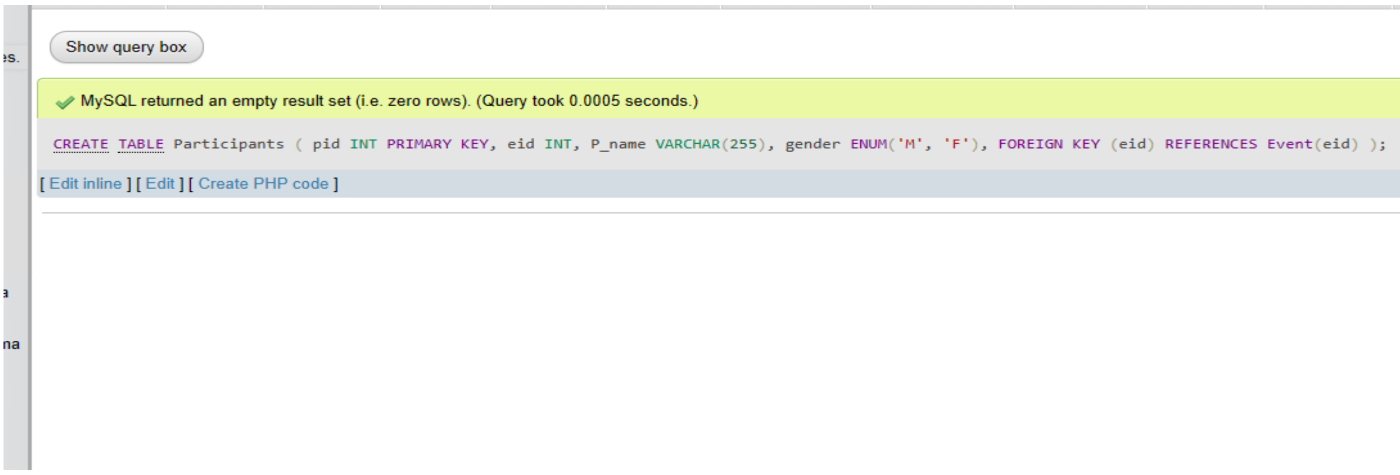


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0102 seconds.)
CREATE TABLE Event ( eid INT PRIMARY KEY, ename VARCHAR(255) NOT NULL DEFAULT 'BIRTHDAY', start_date DATE, end_date DATE );
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Participants(pid, eid,P_name,gender)

Gender should be 'M' or 'F' only. And Value for ename should not be null and default value should be "BIRTHDAY".

```
CREATE TABLE Participants (
   pid INT PRIMARY KEY,
   eid INT,
   P_name VARCHAR(25),
   gender ENUM('M', 'F'),
   FOREIGN KEY (eid) REFERENCES Event(eid)
);
```



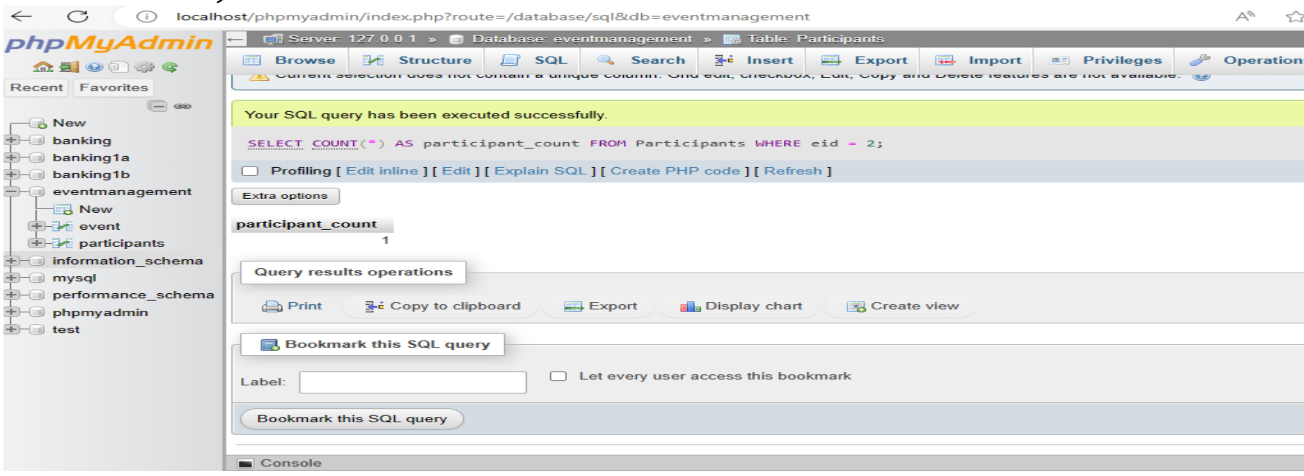
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
CREATE TABLE Participants ( pid INT PRIMARY KEY, eid INT, P_name VARCHAR(255), gender ENUM('M', 'F'), FOREIGN KEY (eid) REFERENCES Event(eid) );
```

Perform the following queries

a) Count the number of participants where eid=2.

SELECT COUNT(*) AS participant_count FROM Participants WHERE eid = 2;



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

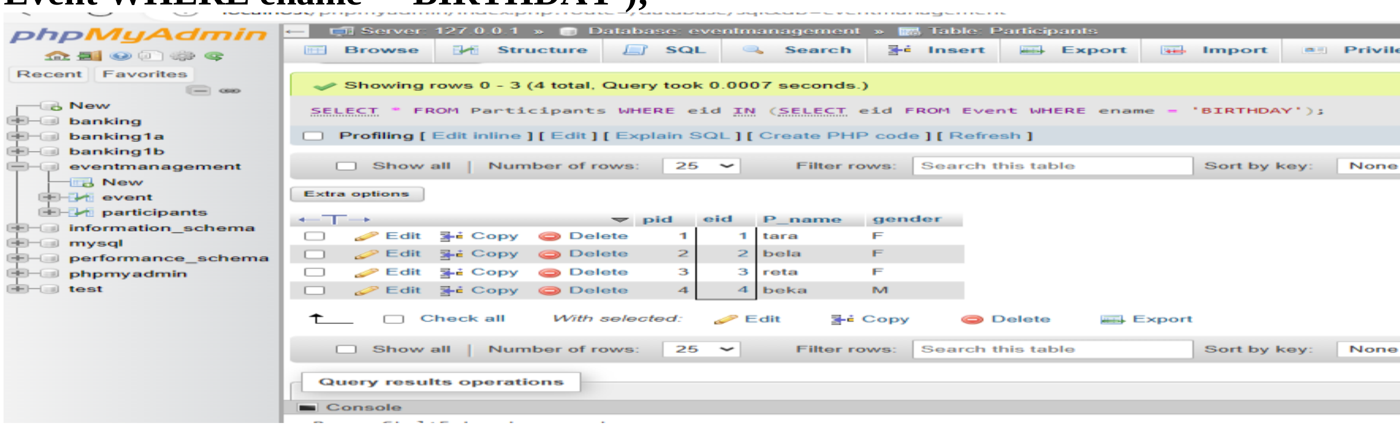
Your SQL query has been executed successfully.

```
SELECT COUNT(*) AS participant_count FROM Participants WHERE eid = 2;
```

```
participant_count
                1
```

b) Display the records of participants who have participated in "BIRTHDAY".

SELECT * FROM Participants WHERE eid IN (SELECT eid FROM Event WHERE ename = 'BIRTHDAY');



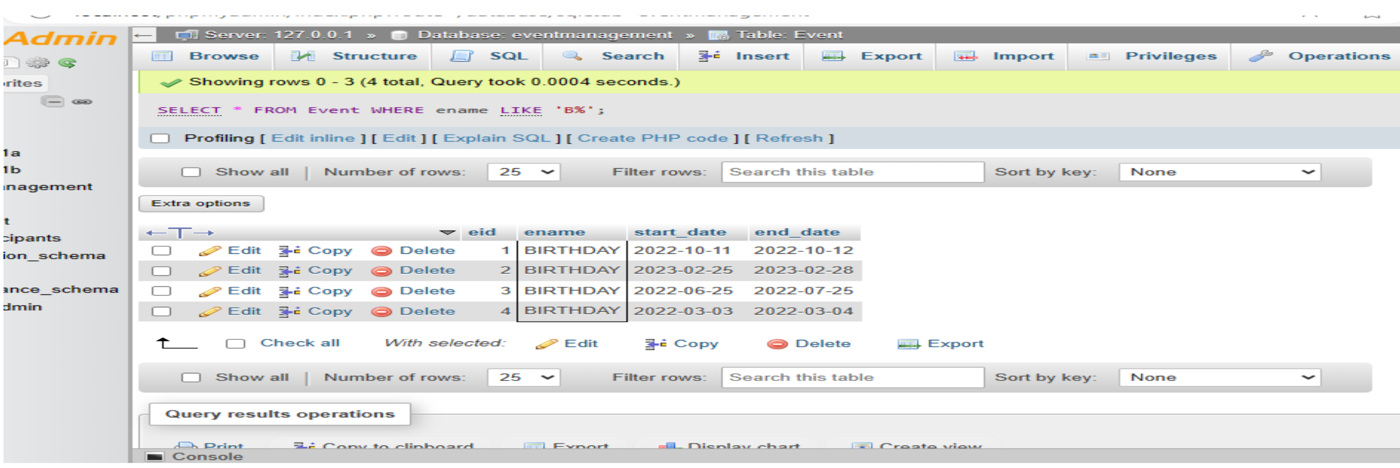
Showing rows 0 - 3 (4 total, Query took 0.0007 seconds.)

```
SELECT * FROM Participants WHERE eid IN (SELECT eid FROM Event WHERE ename = 'BIRTHDAY');
```

	pid	eid	P_name	gender
Edit Copy Delete	1	1	tara	F
Edit Copy Delete	2	2	bela	F
Edit Copy Delete	3	3	reta	F
Edit Copy Delete	4	4	beka	M

c) Display the records of events where ename begins with 'B'.

SELECT * FROM Event WHERE ename LIKE 'B%';



Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

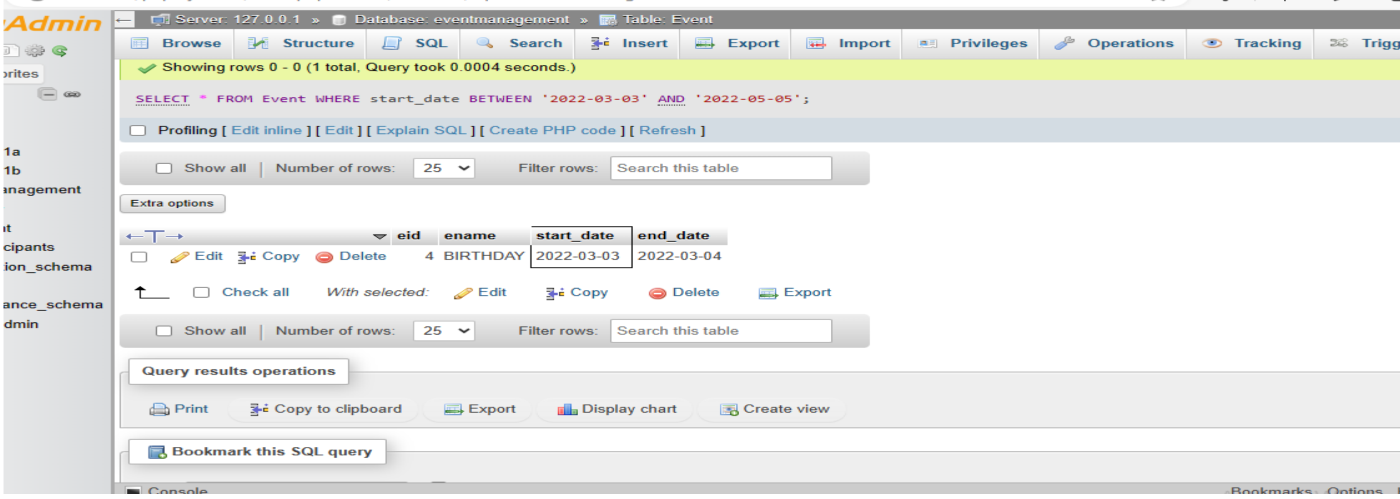
```
SELECT * FROM Event WHERE ename LIKE 'B%';
```

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

eid	ename	start_date	end_date
1	BIRTHDAY	2022-10-11	2022-10-12
2	BIRTHDAY	2023-02-25	2023-02-28
3	BIRTHDAY	2022-06-25	2022-07-25
4	BIRTHDAY	2022-03-03	2022-03-04

d) Display all event starting between "2022/03/03" and "2022/05/05"

SELECT * FROM Event WHERE start_date BETWEEN '2022-03-03' AND '2022-05-05';



Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

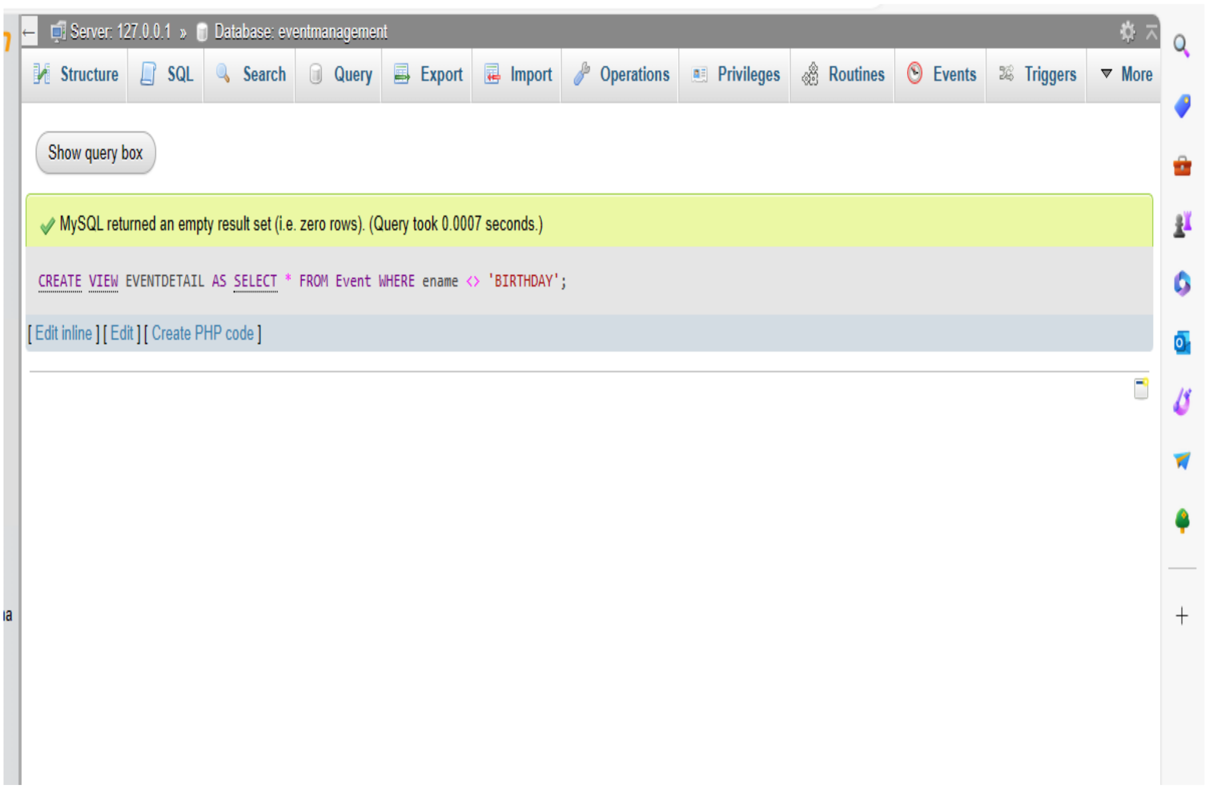
```
SELECT * FROM Event WHERE start_date BETWEEN '2022-03-03' AND '2022-05-05';
```

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

eid	ename	start_date	end_date
4	BIRTHDAY	2022-03-03	2022-03-04

e) Create a view EVENTDETAIL, that displays events other than "BIRTHDAY".

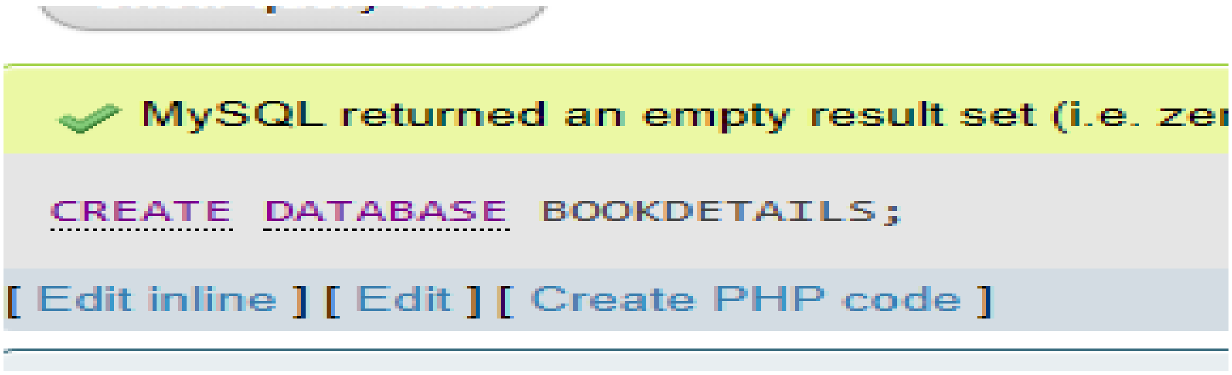
CREATE VIEW EVENTDETAIL AS SELECT *FROM Event WHERE ename <> 'BIRTHDAY';



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
CREATE VIEW EVENTDETAIL AS SELECT * FROM Event WHERE ename <> 'BIRTHDAY';
```

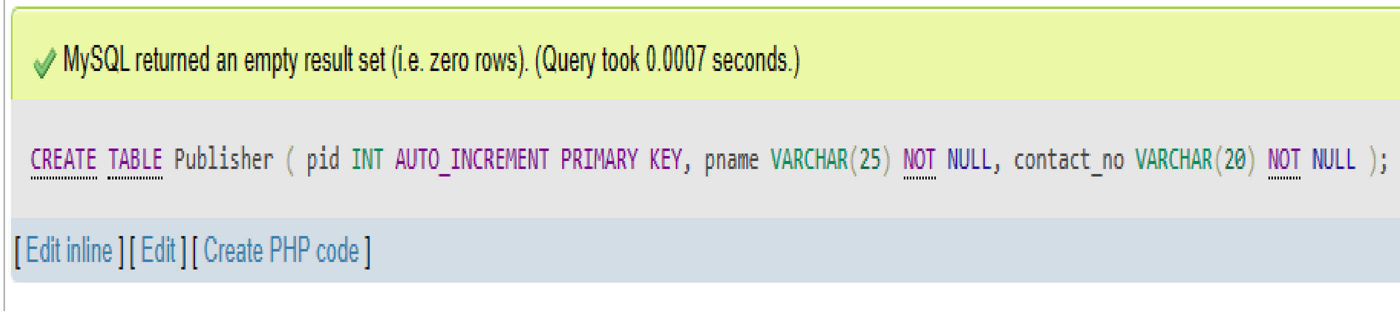
3. Create a database BOOKDETAILS with the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model.



MySQL returned an empty result set (i.e. zer

```
CREATE DATABASE BOOKDETAILS;
```

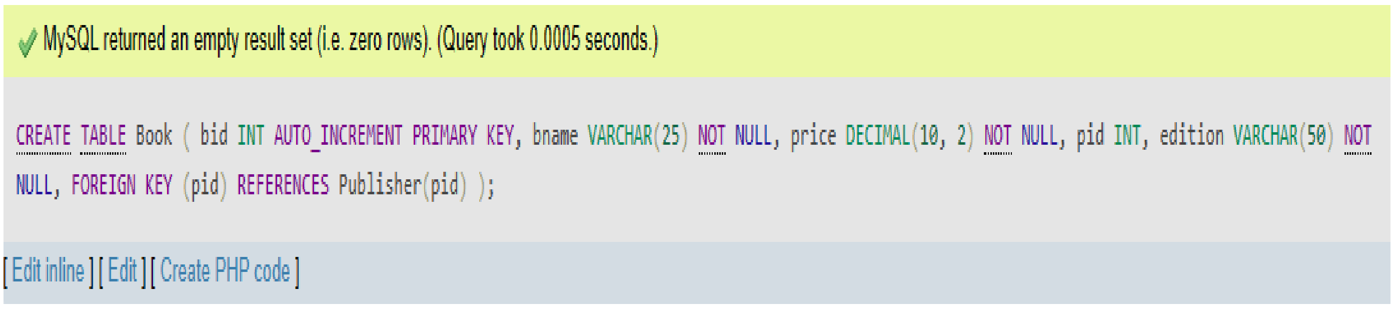
Publisher(pid,pname,contact_no)



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
CREATE TABLE Publisher ( pid INT AUTO_INCREMENT PRIMARY KEY, pname VARCHAR(25) NOT NULL, contact_no VARCHAR(20) NOT NULL );
```

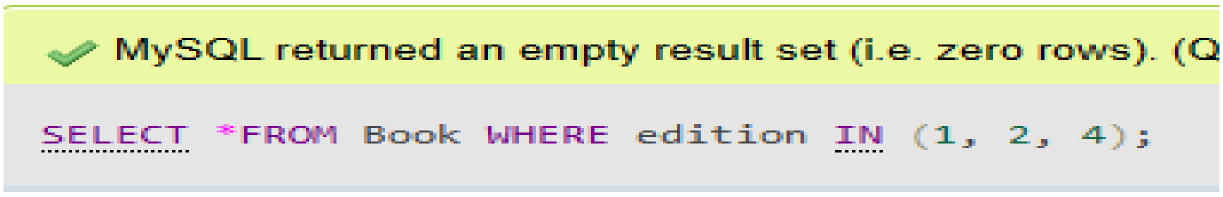
Book(bid,bname,price,pid,edition)



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE Book ( bid INT AUTO_INCREMENT PRIMARY KEY, bname VARCHAR(25) NOT NULL, price DECIMAL(10, 2) NOT NULL, pid INT, edition VARCHAR(50) NOT
NULL, FOREIGN KEY (pid) REFERENCES Publisher(pid) );

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

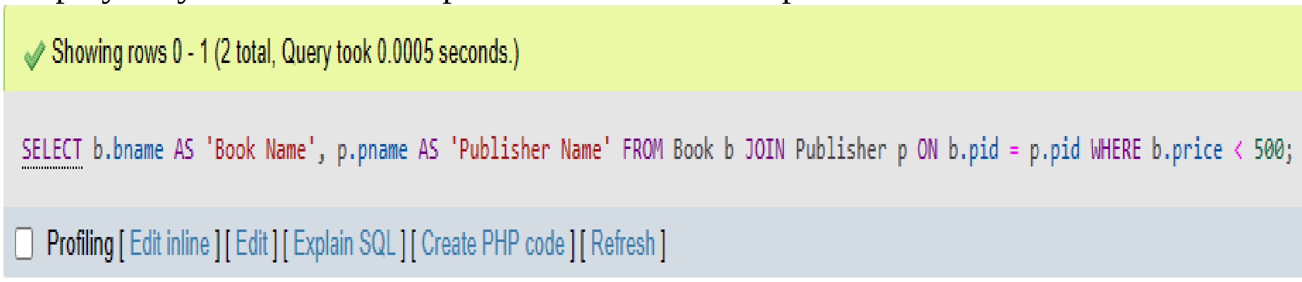
Perform the following queries

a) Display all the records from Book table where edition is either (1,2 or 4) using IN operator.



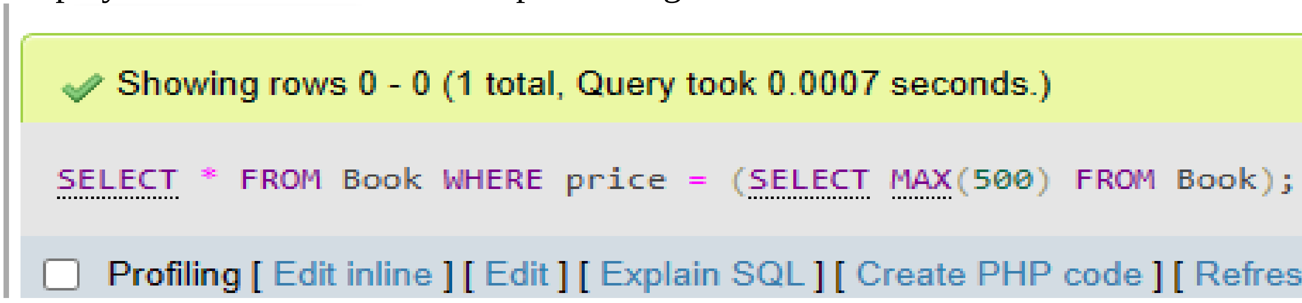
```
✓ MySQL returned an empty result set (i.e. zero rows). (Q

SELECT *FROM Book WHERE edition IN (1, 2, 4);
```

b) Display only book name and publisher name where price is less than 500.



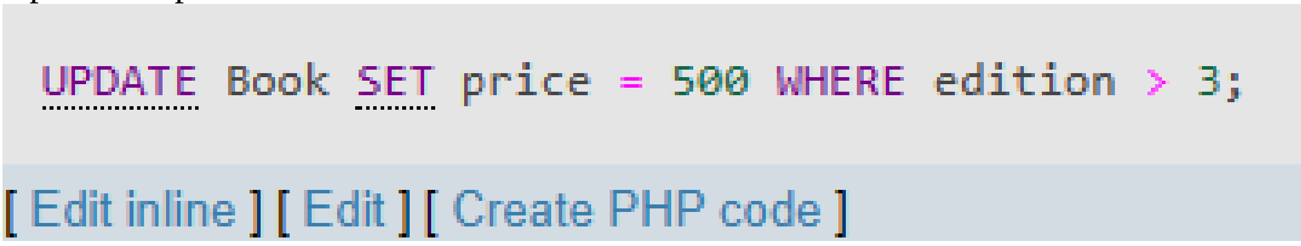
```
✓ Showing rows 0 - 1 (2 total, Query took 0.0005 seconds.)

SELECT b.bname AS 'Book Name', p.pname AS 'Publisher Name' FROM Book b JOIN Publisher p ON b.pid = p.pid WHERE b.price < 500;

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

c) Display the details of book where price is highest.



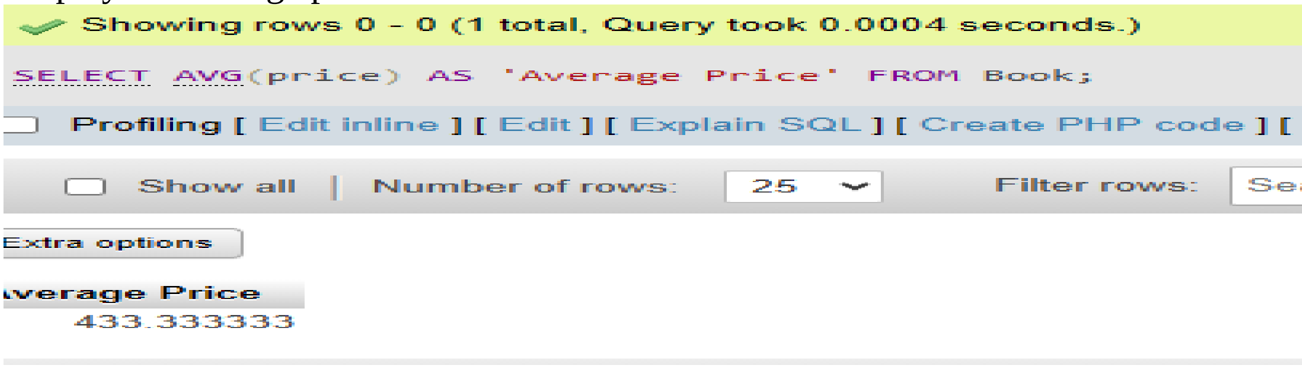
```
✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

SELECT * FROM Book WHERE price = (SELECT MAX(500) FROM Book);

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refres
```

d) Update the price of book to 500 where edition is > 3



```
UPDATE Book SET price = 500 WHERE edition > 3;

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

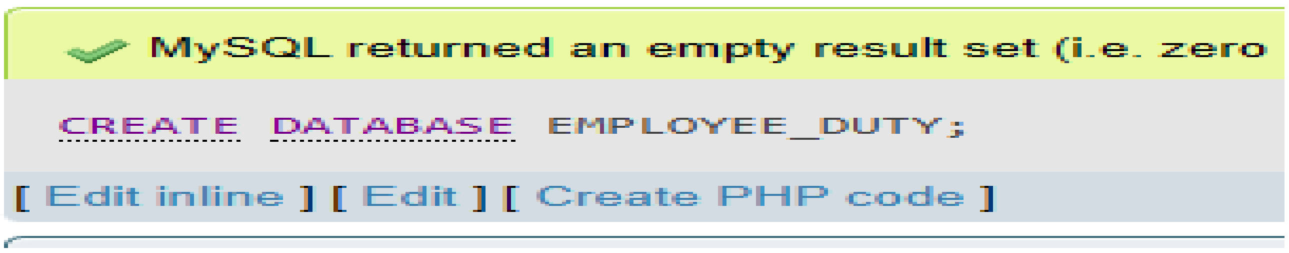
e) Display the average price of the books



```
Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)
SELECT AVG(price) AS 'Average Price' FROM Book;
Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [
Show all | Number of rows: 25   Filter rows: Se
Extra options
verage Price
    433.333333
```

4. Create a database EMPLOYEE_DUTY with the following tables Set appropriate datatypes and set primary keys and foreign keys, Create relationship betweentables and create E-R Model

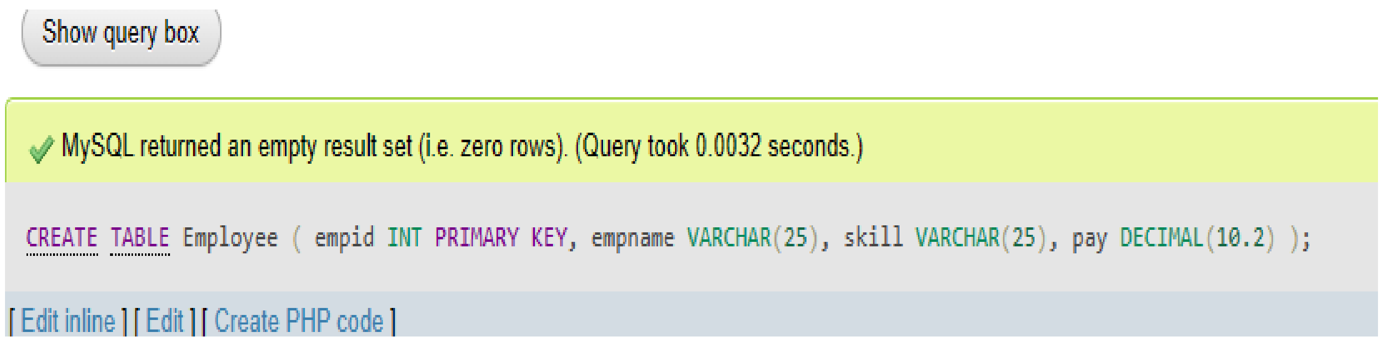
CREATE DATABASE EMPLOYEE_DUTY;



```
MySQL returned an empty result set (i.e. zero
CREATE DATABASE EMPLOYEE_DUTY;
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Employee(empid,empname,skill,pay)

```
CREATE TABLE Employee (
   empid INT PRIMARY KEY,
   empname VARCHAR(25),
   skill VARCHAR(25),
   pay DECIMAL(10.2)
);
```

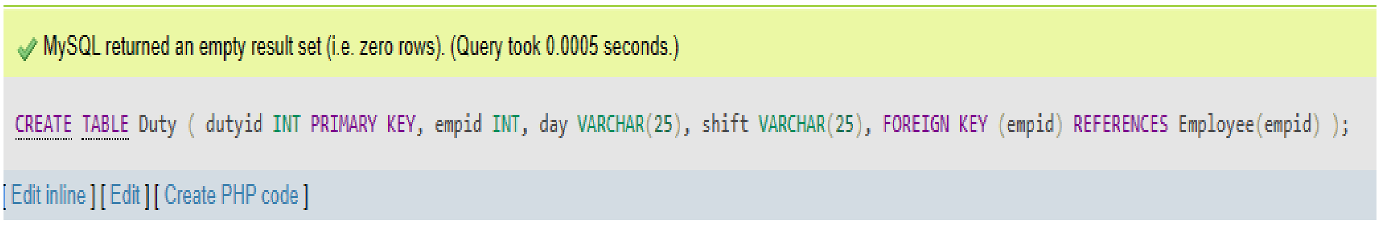


```
Show query box
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0032 seconds.)
CREATE TABLE Employee ( empid INT PRIMARY KEY, empname VARCHAR(25), skill VARCHAR(25), pay DECIMAL(10.2) );
[ Edit inline ] [ Edit ] [ Create PHP code ]
```

Duty(dutyid,empid, day, shift)

```
CREATE TABLE Duty (
   dutyid INT PRIMARY KEY,
   empid INT,
   day VARCHAR(25),
```

shift VARCHAR(25),
FOREIGN KEY (empid) REFERENCES Employee(empid)
);

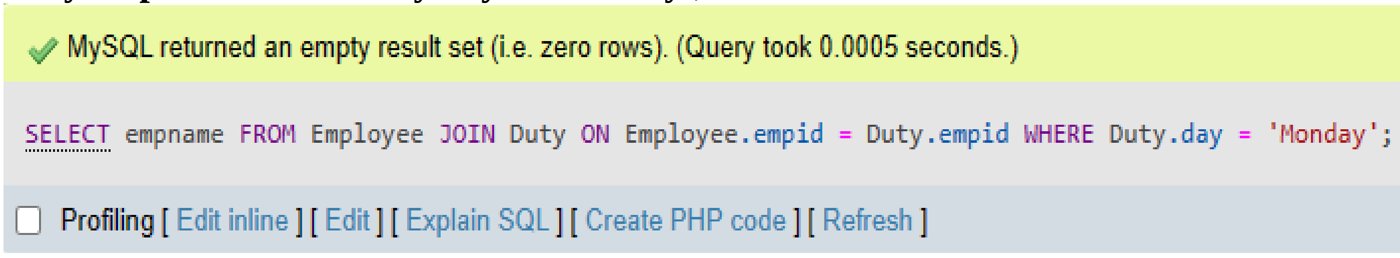


```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE Duty ( dutyid INT PRIMARY KEY, empid INT, day VARCHAR(25), shift VARCHAR(25), FOREIGN KEY (empid) REFERENCES Employee(empid) );

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

Perform the following Queries:

a) Display names of all employees who came on Monday.

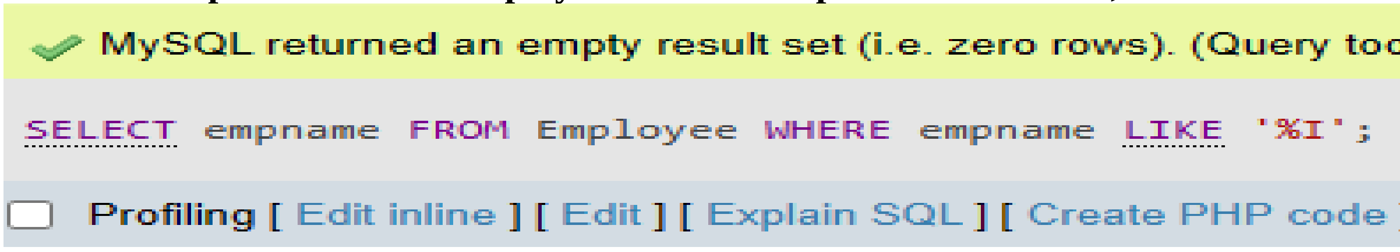
SELECT empname FROM Employee JOIN Duty ON Employee.empid = Duty.empid WHERE Duty.day = 'Monday';



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

SELECT empname FROM Employee JOIN Duty ON Employee.empid = Duty.empid WHERE Duty.day = 'Monday';

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

b) Find the employee names which ends with 'I'.

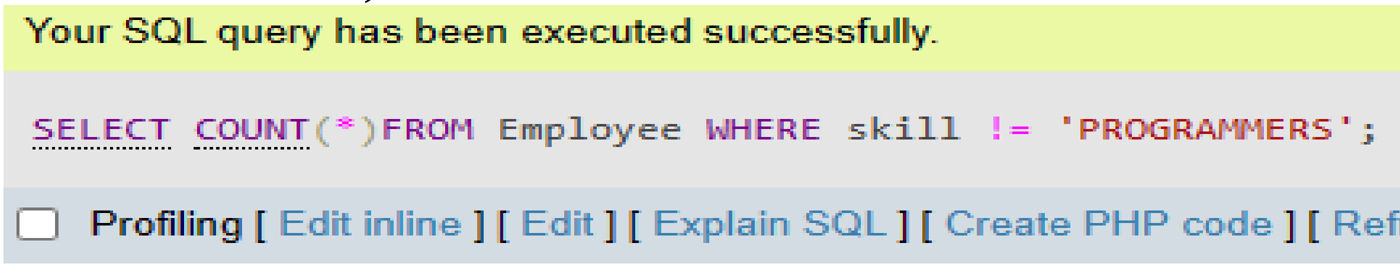
SELECT empname FROM Employee WHERE empname LIKE '%I';



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query too

SELECT empname FROM Employee WHERE empname LIKE '%I';

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

c) Display total no. of employees where skill is other than "PROGRAMMERS".

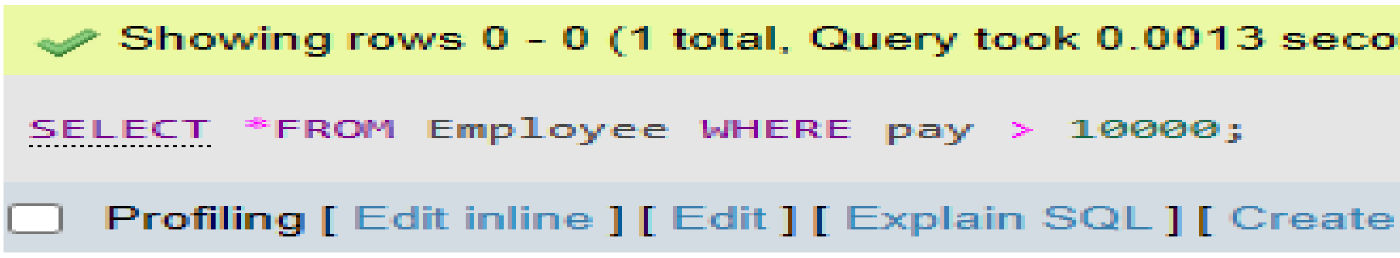
SELECT COUNT(*)FROM Employee WHERE skill !='PROGRAMMERS';



```
Your SQL query has been executed successfully.

SELECT COUNT(*)FROM Employee WHERE skill != 'PROGRAMMERS';

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Ref
```

d) Display employee details where pay is more than 10000.

SELECT *FROM Employee WHERE pay > 10000;

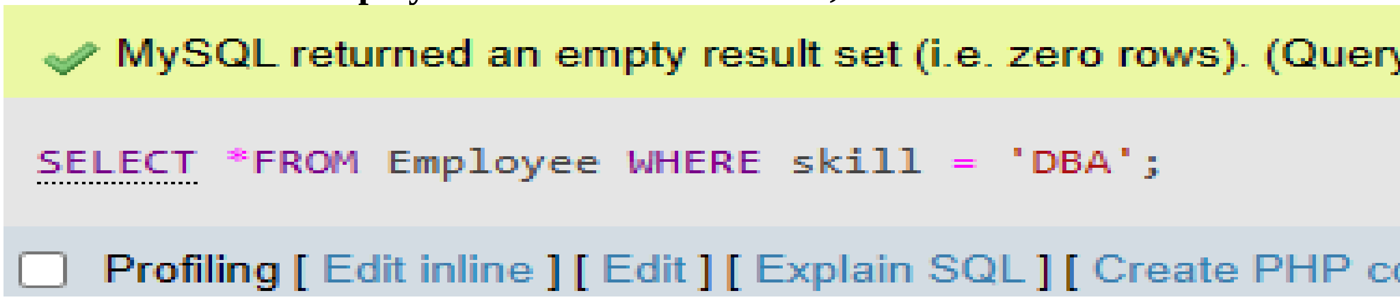

Showing rows 0 - 0 (1 total, Query took 0.0013 seco

```
SELECT *FROM Employee WHERE pay > 10000;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create

e) Display employee details of where skill is "DBA"

SELECT *FROM Employee WHERE skill = 'DBA';

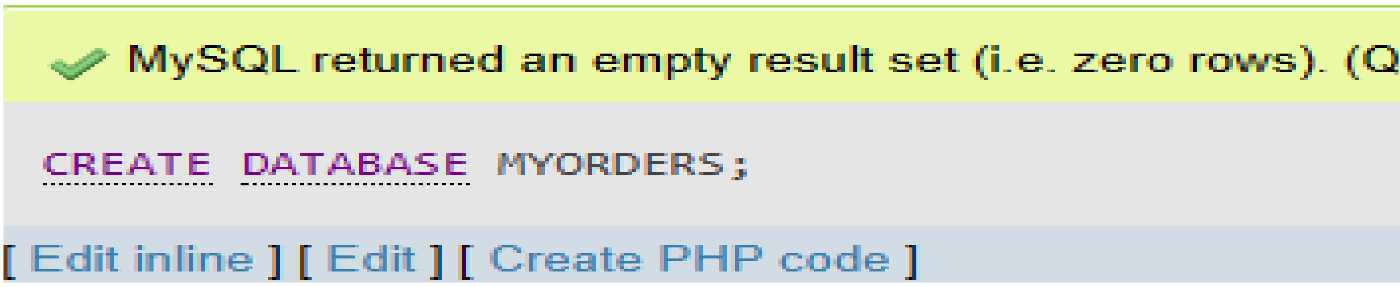

MySQL returned an empty result set (i.e. zero rows). (Query

```
SELECT *FROM Employee WHERE skill = 'DBA';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP co

5. Create a database MYORDERS with the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model.

CREATE DATABASE myorders;


MySQL returned an empty result set (i.e. zero rows). (Q

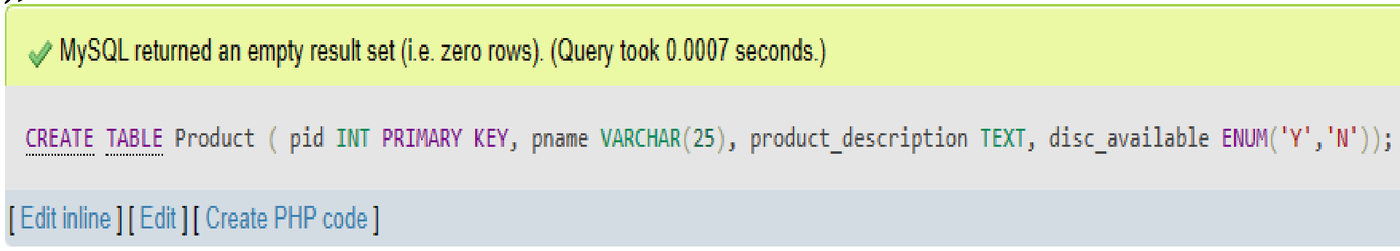
```
CREATE DATABASE MYORDERS;
```

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

Product(pid,pname,product_description, disc_available)

Values for discount available should be 'Y' or 'N'

```
CREATE TABLE Product (
   pid INT PRIMARY KEY,
   pname VARCHAR(25),
   product_description TEXT,
   disc_available  ENUM('Y','N')
);
```


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
CREATE TABLE Product ( pid INT PRIMARY KEY, pname VARCHAR(25), product_description TEXT, disc_available ENUM('Y','N'));
```

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

Order(pid,oid,odate,qty,price)

CREATE TABLE ORDER(
oid INT PRIMARY KEY,
pid INT,
odate DATE,
qty INT,
price DECIMAL(10.2),
FOREIGN KEY (pid) REFERENCES Product(pid)
);

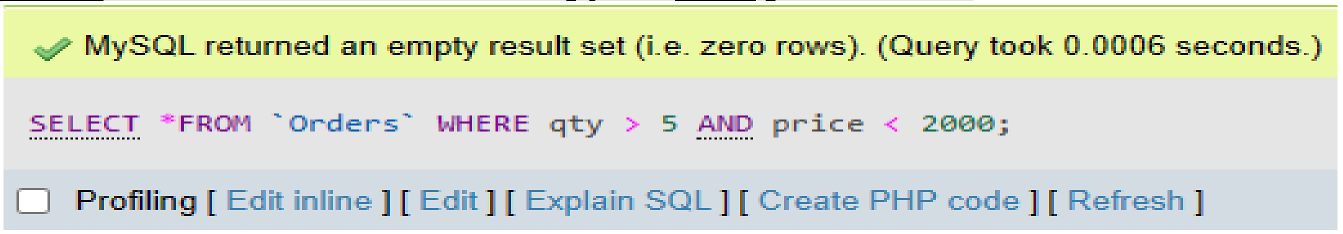
Perform the following queries

a) Display product details where disc available is 'Y'.

SELECT * FROM Product WHERE disc_available = 'Y';

b) Display order details where quantity>5 and price< 2000.

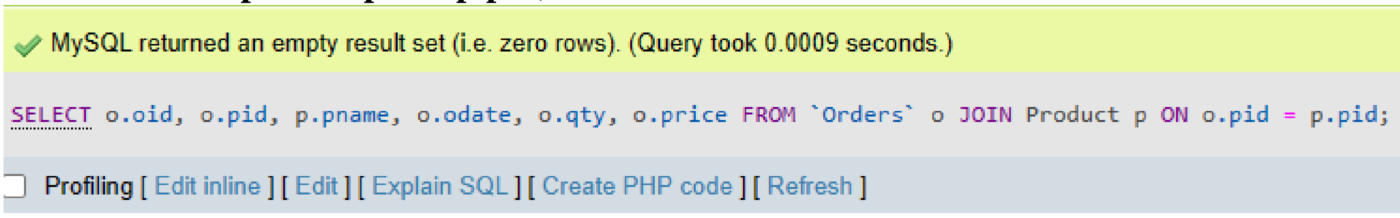
SELECT *FROM `Orders` WHERE qty > 5 AND price < 2000;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

SELECT *FROM `Orders` WHERE qty > 5 AND price < 2000;

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

c) Display order details along with product name.

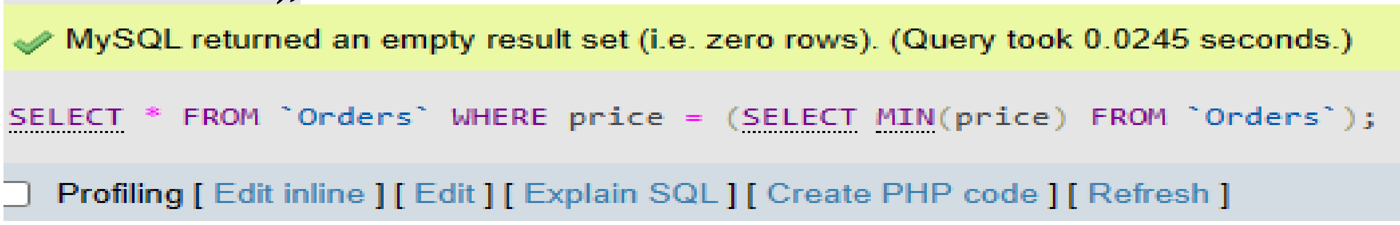
SELECT o.oid, o.pid, p.pname, o.odate, o.qty, o.price
FROM `Order` o
JOIN Product p ON o.pid = p.pid;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

SELECT o.oid, o.pid, p.pname, o.odate, o.qty, o.price FROM `Orders` o JOIN Product p ON o.pid = p.pid;

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

d) Display minimum price order.

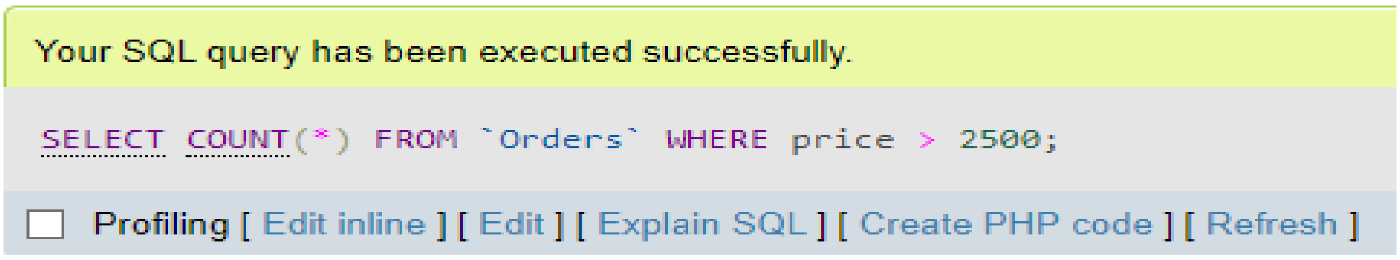
SELECT * FROM `Orders` WHERE price = (SELECT MIN(price)
FROM `Orders`);



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0245 seconds.)

SELECT * FROM `Orders` WHERE price = (SELECT MIN(price) FROM `Orders`);

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

e) Count the number of orders where price is > 2500.

SELECT COUNT(*) FROM `Orders` WHERE price > 2500;

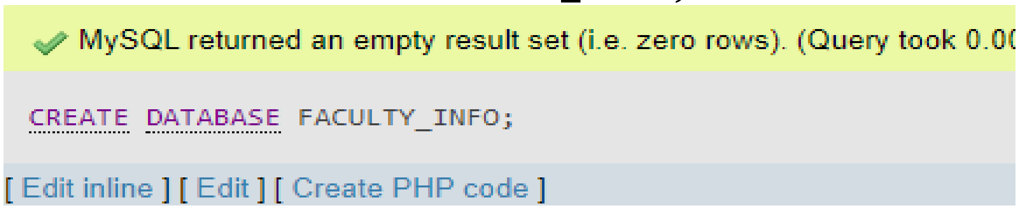

Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM `Orders` WHERE price > 2500;
```

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

6. Create a database FACULTY_INFO with the following tables and create E-R Model.Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables

CREATE DATABASE FACULTY_INFO;

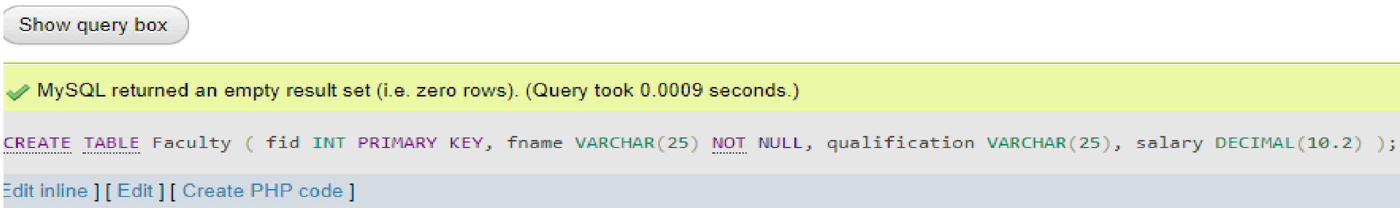

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0

```
CREATE DATABASE FACULTY_INFO;
```

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

Faculty (fid,fname,qualification, salary)

```
CREATE TABLE Faculty (
   fid INT PRIMARY KEY,
   fname VARCHAR(25) NOT NULL,
   qualification VARCHAR(25),
   salary DECIMAL(10.2)
);
```


Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CREATE TABLE Faculty ( fid INT PRIMARY KEY, fname VARCHAR(25) NOT NULL, qualification VARCHAR(25), salary DECIMAL(10.2) );
```

dit inline] [Edit] [Create PHP code]

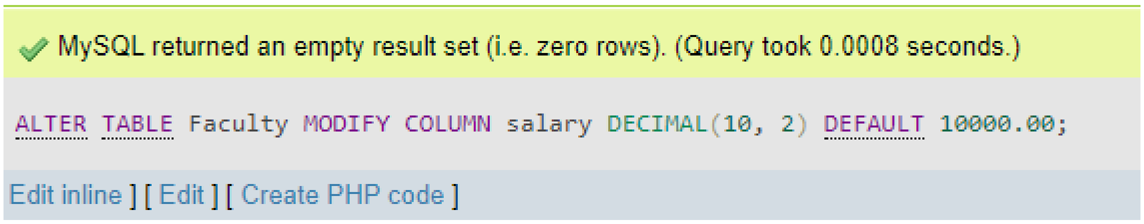
Course(cid,fid,course_name, fees)

```
CREATE TABLE  Course (
   cid INT PRIMARY KEY,
   fid INT,
   course_name VARCHAR(25) NOT NULL,
   fees DECIMAL(10.2),
   FOREIGN KEY (fid) REFERENCES Faculty(fid)
);
```

Perform the following queries

a) Default value for salary should be 10,000.

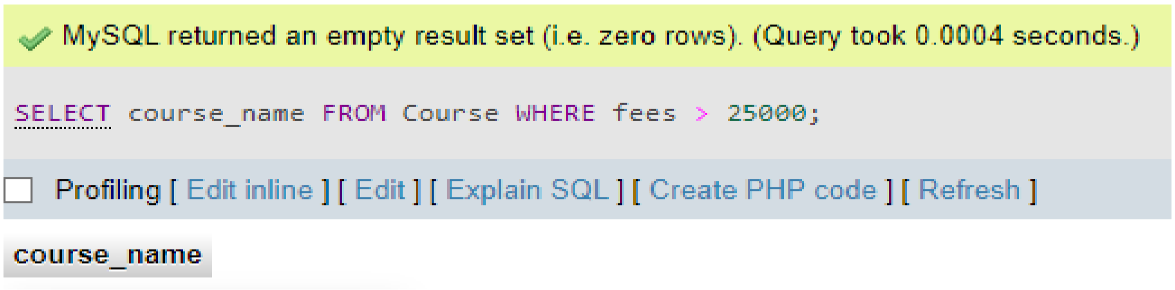
ALTER TABLE Faculty MODIFY COLUMN salary DECIMAL(10, 2) DEFAULT 10000.00;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0008 seconds.)

ALTER TABLE Faculty MODIFY COLUMN salary DECIMAL(10, 2) DEFAULT 10000.00;

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

b) Display the name of courses where fees are greater than 25000.

SELECT course_name FROM Course WHERE fees > 25000;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

SELECT course_name FROM Course WHERE fees > 25000;

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

course_name
```

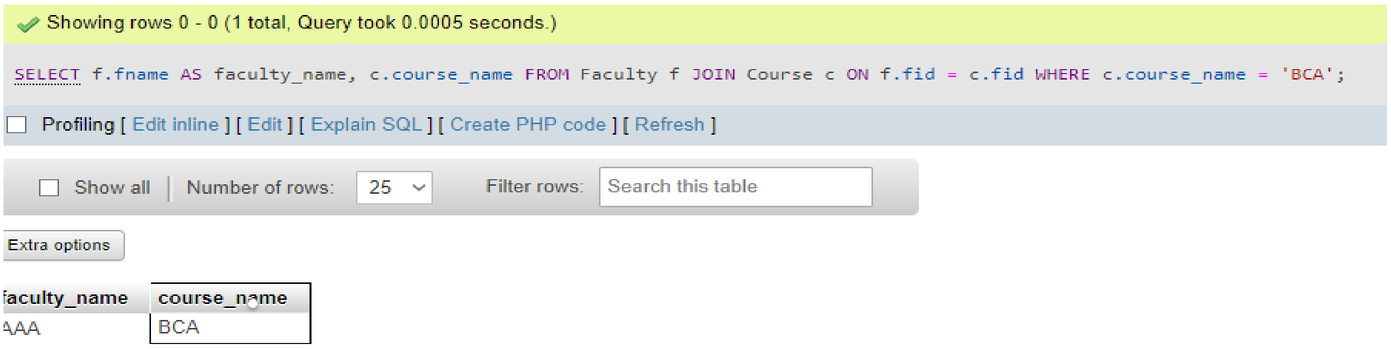
c) Display the faculty names and course name for BCA course

SELECT f.fname AS faculty_name, c.course_name

FROM Faculty f

JOIN Course c ON f.fid = c.fid

WHERE c.course_name = 'BCA';

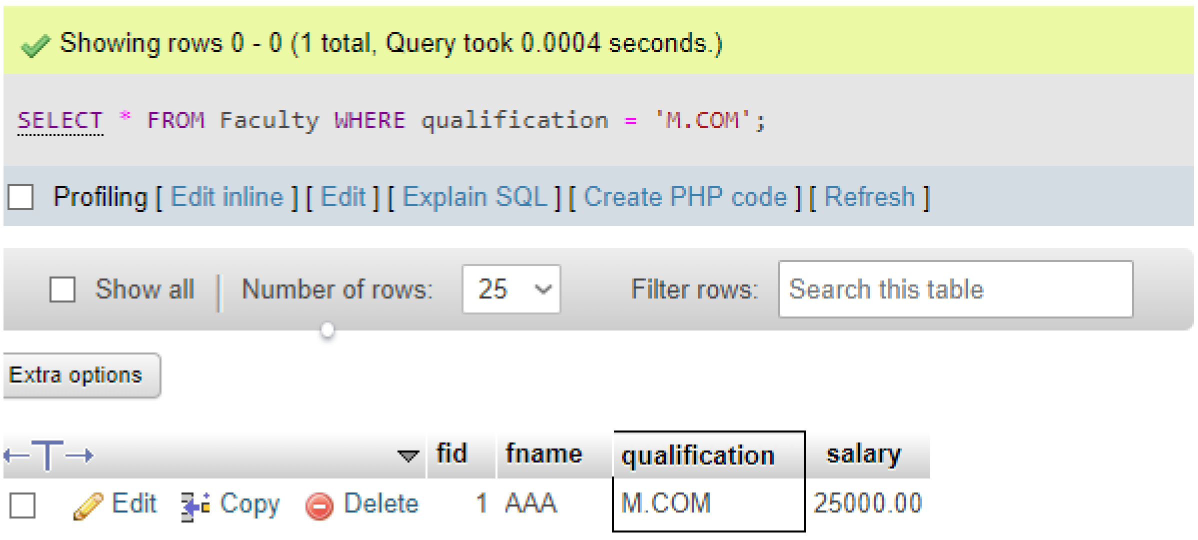


```
Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

SELECT f.fname AS faculty_name, c.course_name FROM Faculty f JOIN Course c ON f.fid = c.fid WHERE c.course_name = 'BCA';

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

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

Extra options
```

aculty_name	course_name
AA	BCA

d) Display the records of faculties where qualification is "M.COM".

SELECT * FROM Faculty WHERE qualification = 'M.COM';

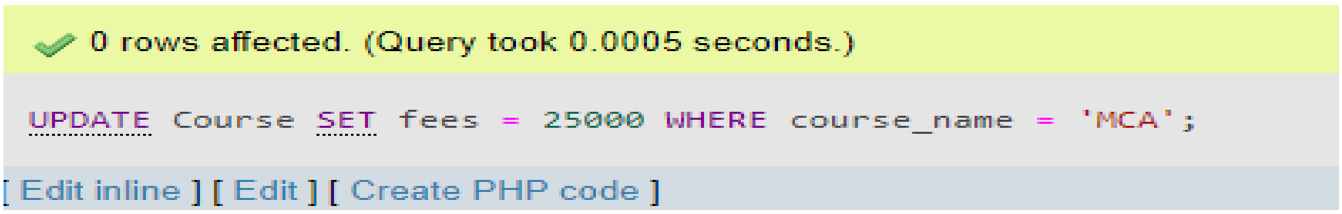


```
Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

SELECT * FROM Faculty WHERE qualification = 'M.COM';

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

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

Extra options
```

	fid	fname	qualification	salary
Edit Copy Delete	1	AAA	M.COM	25000.00

e) Update the fees of course to 25000 where where course_name is "MCA"

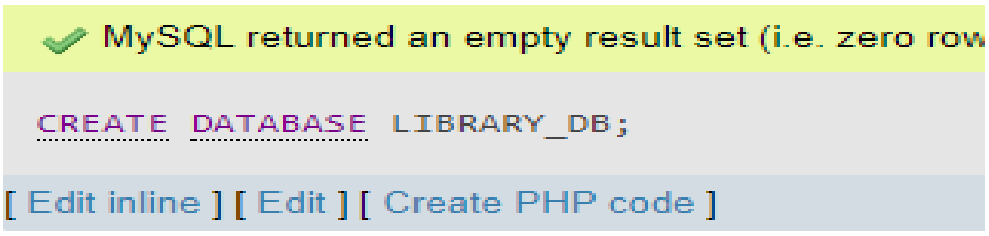
UPDATE Course SET fees = 25000 WHERE course_name = 'MCA';



```
0 rows affected. (Query took 0.0005 seconds.)

UPDATE Course SET fees = 25000 WHERE course_name = 'MCA';

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

7. Create a database LIBRARY DB with the following tables Set appropriate datatypes and set primary keys and foreign keys. Create relationship between tables and create E-R Model.

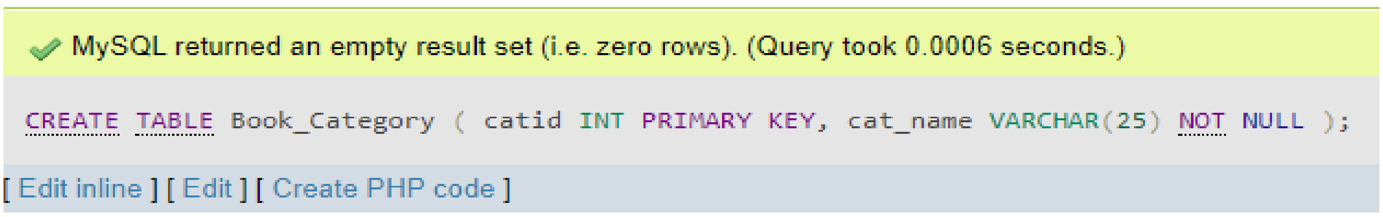
CREATE DATABASE LIBRARY_DB;



```
MySQL returned an empty result set (i.e. zero row

CREATE DATABASE LIBRARY_DB;

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

Book Category(catid,cat_name)

```
CREATE TABLE  Book_Category (
   catid INT PRIMARY KEY,
   cat_name VARCHAR(25) NOT NULL
);
```

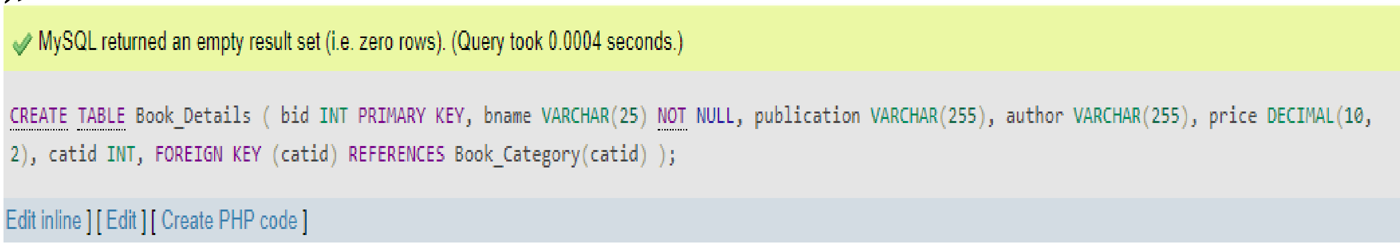

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
CREATE TABLE Book_Category ( catid INT PRIMARY KEY, cat_name VARCHAR(25) NOT NULL );
```

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

Book_details(bid,bname, publication, author, price,catid)

```
CREATE TABLE  Book_Details (
   bid INT PRIMARY KEY,
   bname VARCHAR(25) NOT NULL,
   publication VARCHAR(255),
   author VARCHAR(255),
   price DECIMAL(10, 2),
   catid INT,
   FOREIGN KEY (catid) REFERENCES Book_Category(catid)
);
```


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

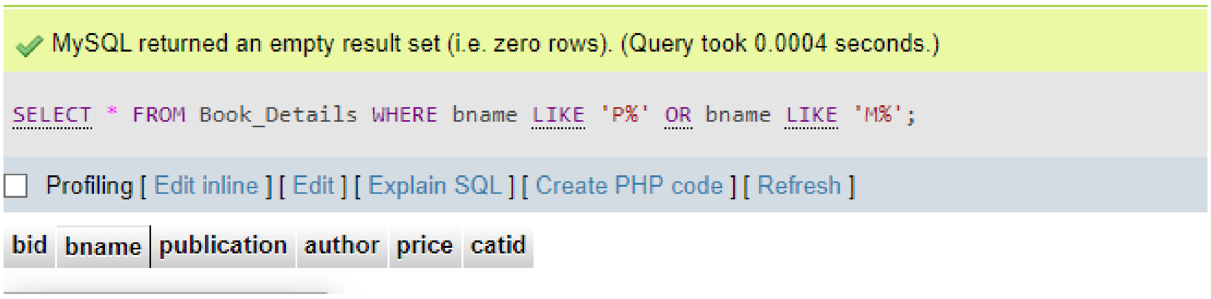
```
CREATE TABLE Book_Details ( bid INT PRIMARY KEY, bname VARCHAR(25) NOT NULL, publication VARCHAR(255), author VARCHAR(255), price DECIMAL(10,
2), catid INT, FOREIGN KEY (catid) REFERENCES Book_Category(catid) );
```

Edit inline] [Edit] [Create PHP code]

Perform the following queries

a) Display books where book name begins with letter "P" or "M".

SELECT * FROM Book_Details WHERE bname LIKE 'P%' OR bname LIKE 'M%';


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

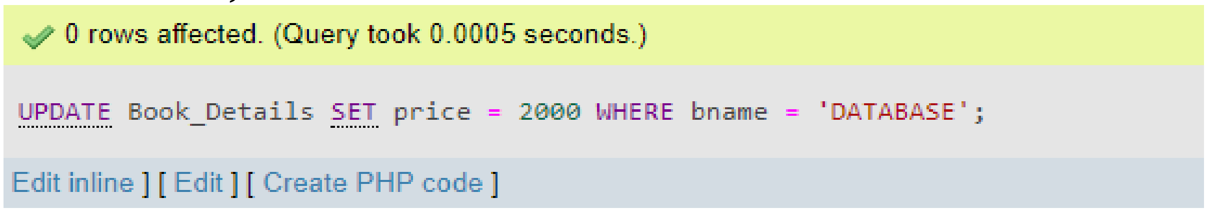
```
SELECT * FROM Book_Details WHERE bname LIKE 'P%' OR bname LIKE 'M%';
```

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

bid	bname	publication	author	price	catid

b) Update price of book "DATABASE" to 2000.

UPDATE Book_Details SET price = 2000 WHERE bname = 'DATABASE';

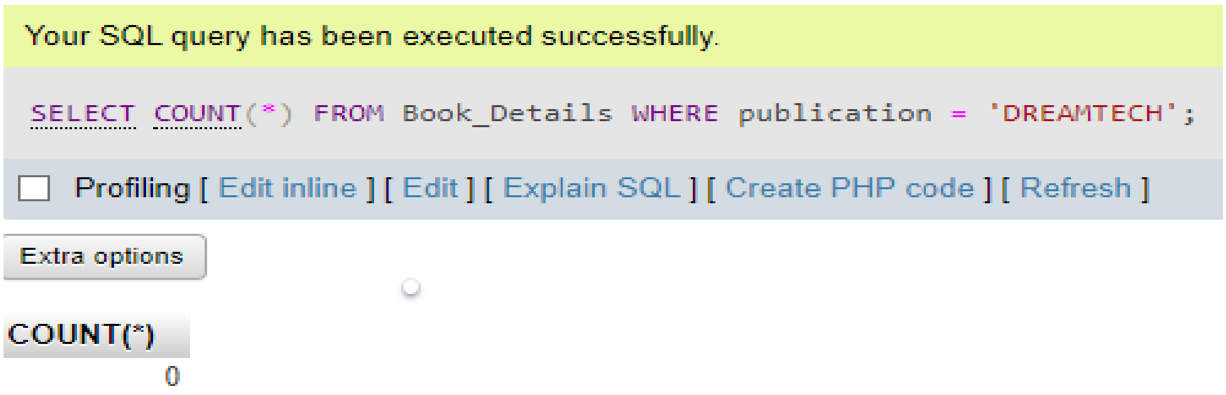

0 rows affected. (Query took 0.0005 seconds.)

```
UPDATE Book_Details SET price = 2000 WHERE bname = 'DATABASE';
```

Edit inline] [Edit] [Create PHP code]

c) Count number of books where publication is "DREAMTECH"

SELECT COUNT(*) FROM Book_Details WHERE publication = 'DREAMTECH';


Your SQL query has been executed successfully.

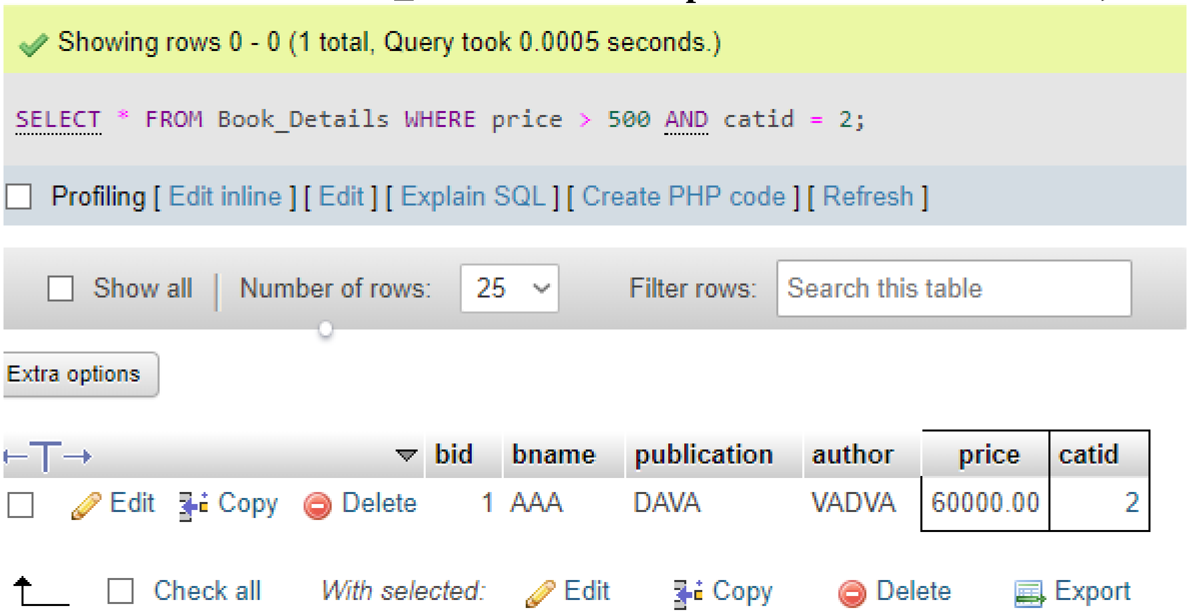
```
SELECT COUNT(*) FROM Book_Details WHERE publication = 'DREAMTECH';
```

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

Extra options

```
COUNT(*)
       0
```

d) Display books where price is >500 and catid is 2
SELECT * FROM Book_Details WHERE price > 500 AND catid = 2;


Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM Book_Details WHERE price > 500 AND catid = 2;
```

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

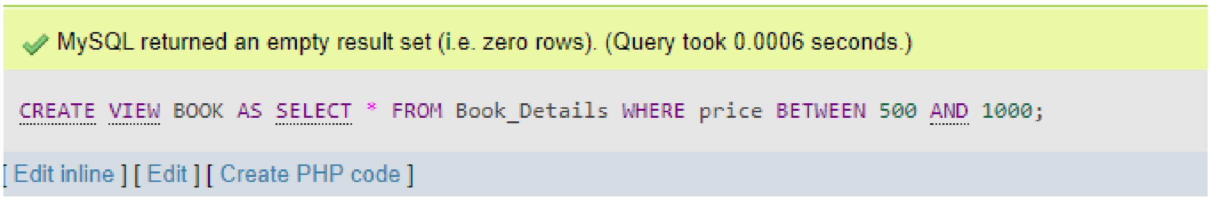
Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	bid	bname	publication	author	price	catid
Edit Copy Delete	1	AAA	DAVA	VADVA	60000.00	2

Check all With selected: Edit Copy Delete Export

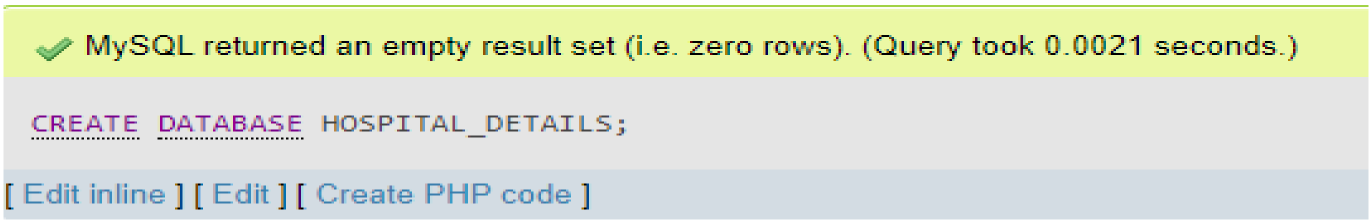
e) Create a view BOOK which will display all the book details where price is between 500 and 1000. CREATE VIEW BOOK AS
SELECT * FROM Book_Details WHERE price BETWEEN 500 AND 1000;


MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
CREATE VIEW BOOK AS SELECT * FROM Book_Details WHERE price BETWEEN 500 AND 1000;
```

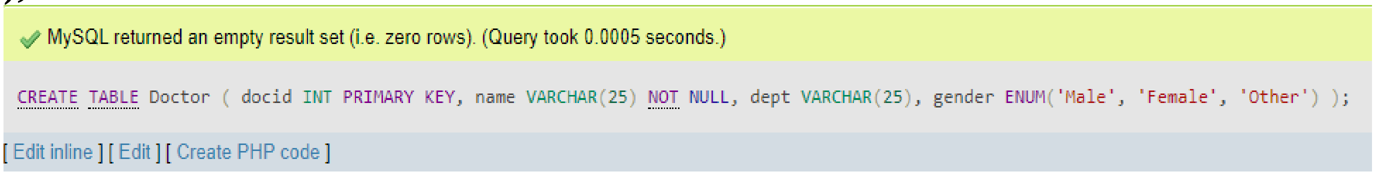
[Edit inline] [Edit] [Create PHP code]

8. Create a database HOSPITAL DETAILS with the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model.
CREATE DATABASE HOSPITAL_DETAILS;


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

CREATE DATABASE HOSPITAL_DETAILS;
```

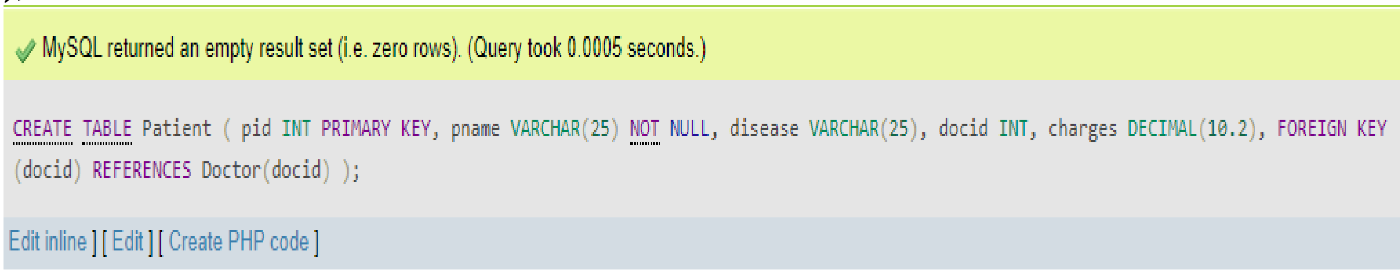
Doctor(docid,name,dept,gender)

```
CREATE TABLE Doctor (
   docid INT PRIMARY KEY,
   name VARCHAR(25) NOT NULL,
   dept VARCHAR(25),
   gender ENUM('Male', 'Female', 'Other')
);
```


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE Doctor ( docid INT PRIMARY KEY, name VARCHAR(25) NOT NULL, dept VARCHAR(25), gender ENUM('Male', 'Female', 'Other') );
```

Patient(pid,pname,disease,docid,charges)

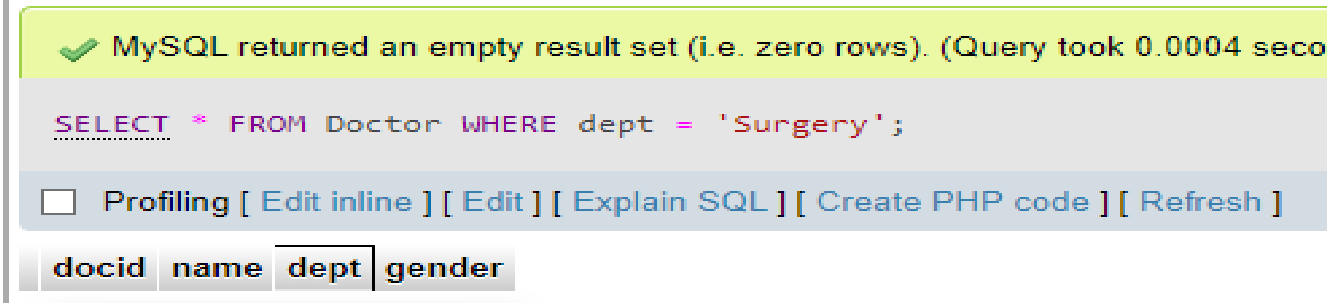
```
CREATE TABLE Patient (
   pid INT PRIMARY KEY,
   pname VARCHAR(25) NOT NULL,
   disease VARCHAR(25),
   docid INT,
   charges DECIMAL(10.2),
   FOREIGN KEY (docid) REFERENCES Doctor(docid)
);
```


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE Patient ( pid INT PRIMARY KEY, pname VARCHAR(25) NOT NULL, disease VARCHAR(25), docid INT, charges DECIMAL(10.2), FOREIGN KEY
(docid) REFERENCES Doctor(docid) );
```

Perform the following queries

a) Display the records for doctors where dept = "Surgery".

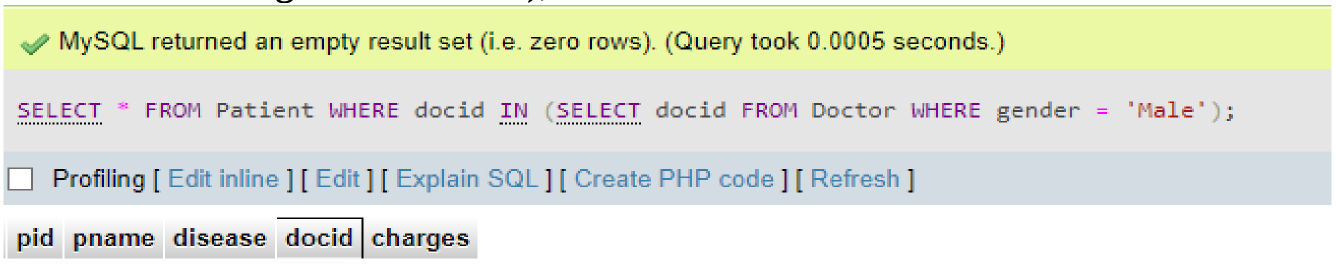
SELECT * FROM Doctor WHERE dept = 'Surgery';


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seco

SELECT * FROM Doctor WHERE dept = 'Surgery';
```
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

docid	name	dept	gender

b) Display the records of patients treated by MALE doctors.

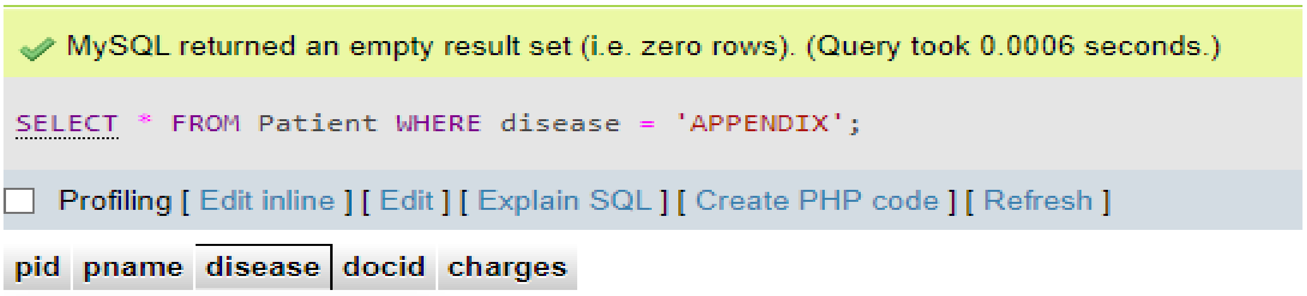
SELECT * FROM Patient WHERE docid IN (SELECT docid FROM Doctor WHERE gender = 'Male');



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

SELECT * FROM Patient WHERE docid IN (SELECT docid FROM Doctor WHERE gender = 'Male');

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

pid | pname | disease | docid | charges
```

c) Display the patients records where disease is "APPENDIX".

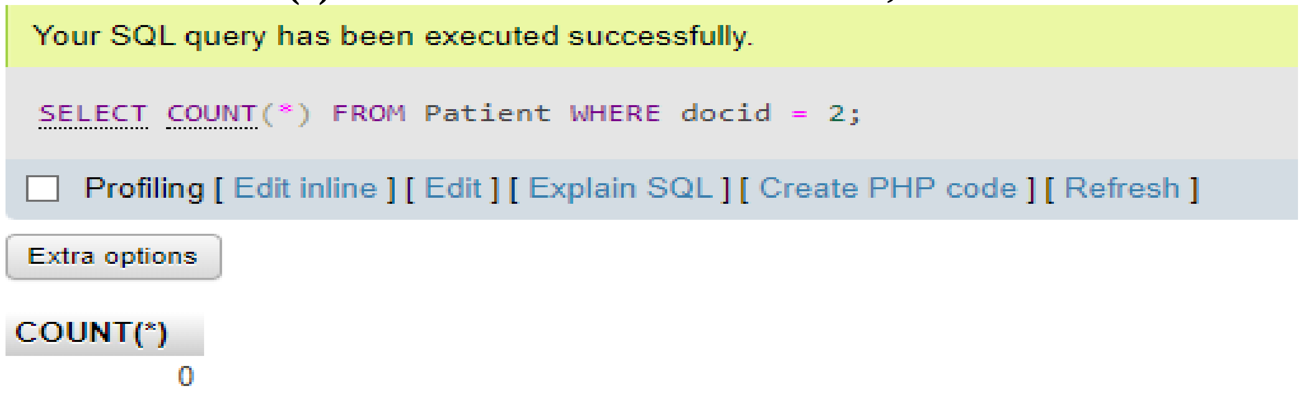
SELECT * FROM Patient WHERE disease = 'APPENDIX';



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

SELECT * FROM Patient WHERE disease = 'APPENDIX';

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

pid | pname | disease | docid | charges
```

d) Count the number of patients where docid is 2

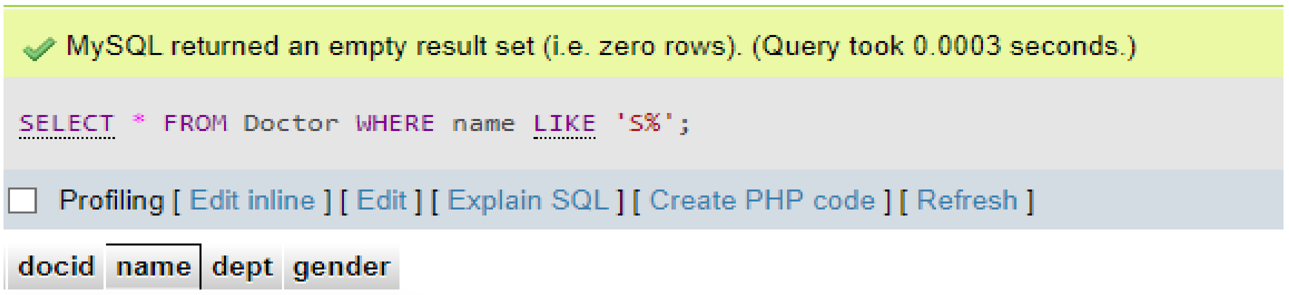
SELECT COUNT(*) FROM Patient WHERE docid = 2;



```
Your SQL query has been executed successfully.

SELECT COUNT(*) FROM Patient WHERE docid = 2;

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

Extra options

COUNT(*)
       0
```

e) Display the records of doctors where doctors name begins with letter S.

SELECT * FROM Doctor WHERE name LIKE 'S%';

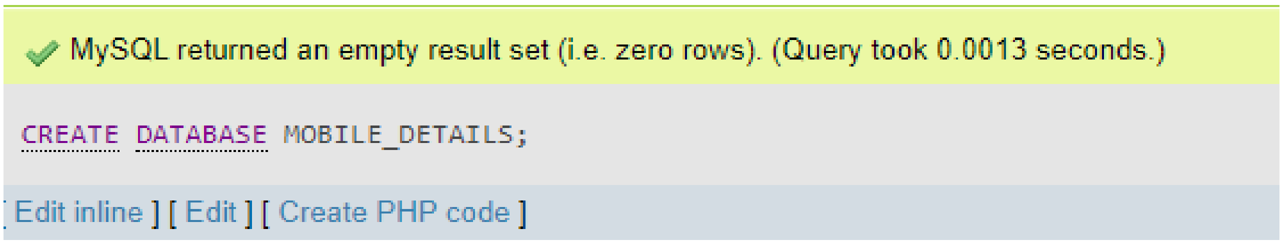


```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

SELECT * FROM Doctor WHERE name LIKE 'S%';

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

docid | name | dept | gender
```

9. Create a database MOBILE_DETAILS with the following tables Set appropriate datatypes and set primary keys and

foreign keys, create relationship between tables and create E-R Model.

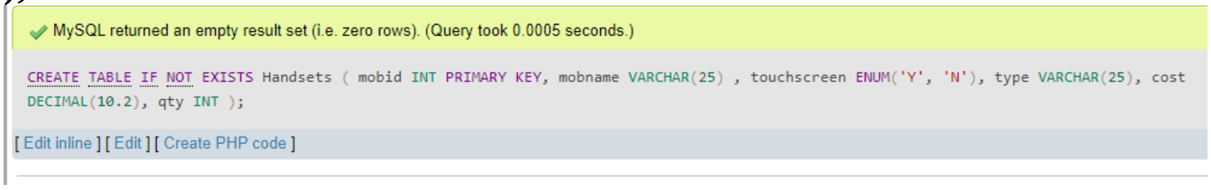
CREATE DATABASE MOBILE_DETAILS;



```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0013 seconds.)

CREATE DATABASE MOBILE_DETAILS;

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

Handsets(mobid, mobname, touchscreen, type, cost,qty)

CREATE TABLE Handsets (
   mobid INT PRIMARY KEY,
   mobname VARCHAR(25) ,
   touchscreen ENUM('Y', 'N'),
   type VARCHAR(25),
   cost DECIMAL(10.2),
   qty INT
);

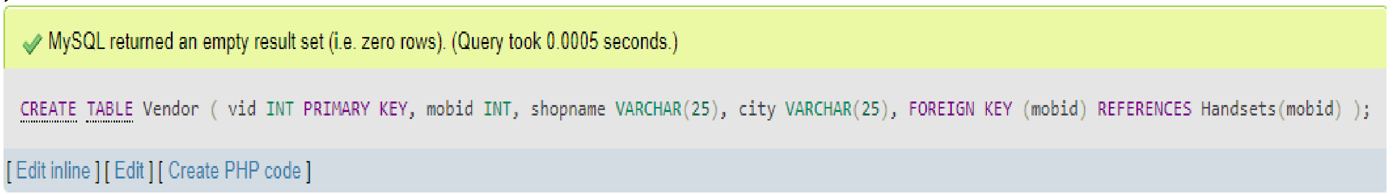


```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE IF NOT EXISTS Handsets ( mobid INT PRIMARY KEY, mobname VARCHAR(25) , touchscreen ENUM('Y', 'N'), type VARCHAR(25), cost
DECIMAL(10.2), qty INT );

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

vendor (vid, mobid, shopname,city)

Value for touchscreen should be either "Y" or "N"

CREATE TABLE Vendor (
   vid INT PRIMARY KEY,
   mobid INT,
   shopname VARCHAR(25),
   city VARCHAR(25),
   FOREIGN KEY (mobid) REFERENCES Handsets(mobid)
);

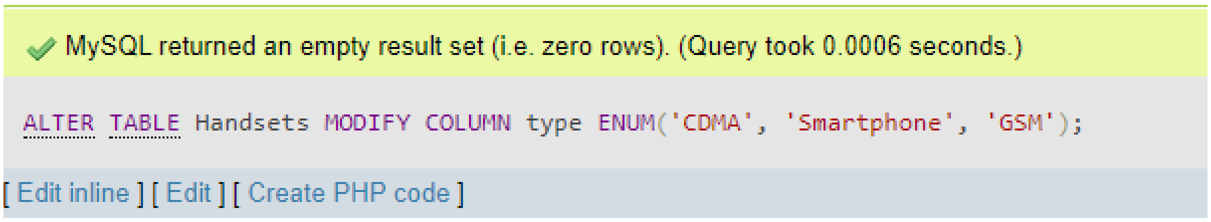


```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

CREATE TABLE Vendor ( vid INT PRIMARY KEY, mobid INT, shopname VARCHAR(25), city VARCHAR(25), FOREIGN KEY (mobid) REFERENCES Handsets(mobid) );

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

Perform the following queries

a) Handset type should be "CDMA", "Smartphone" and "GSM" only.

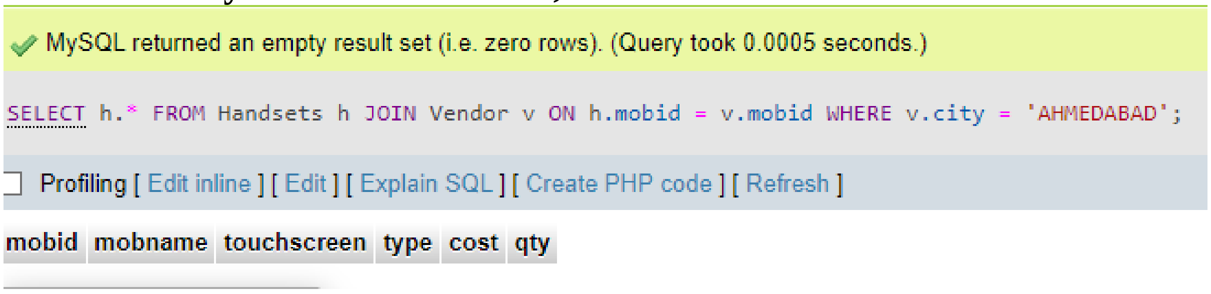
ALTER TABLE Handsets MODIFY COLUMN type ENUM('CDMA', 'Smartphone', 'GSM');



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

ALTER TABLE Handsets MODIFY COLUMN type ENUM('CDMA', 'Smartphone', 'GSM');

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

b) Display handset details for "AHMEDABAD" city

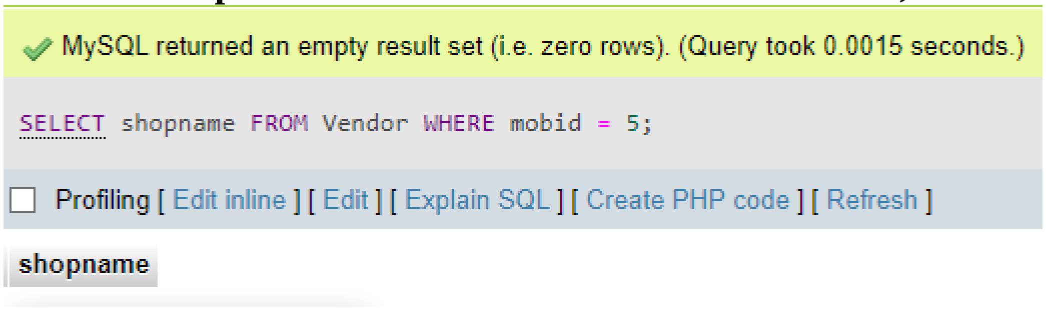
SELECT h.* FROM Handsets h
JOIN Vendor v ON h.mobid = v.mobid
WHERE v.city = 'AHMEDABAD';



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

SELECT h.* FROM Handsets h JOIN Vendor v ON h.mobid = v.mobid WHERE v.city = 'AHMEDABAD';

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

mobid  mobname  touchscreen  type  cost  qty
```

c) Display the shop name where mobid = 5.

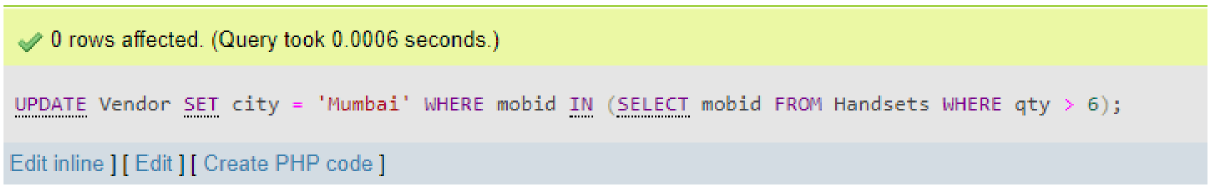
SELECT shopname FROM Vendor WHERE mobid = 5;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0015 seconds.)

SELECT shopname FROM Vendor WHERE mobid = 5;

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

shopname
```

d) Update city to Mumbai where qty > 6.

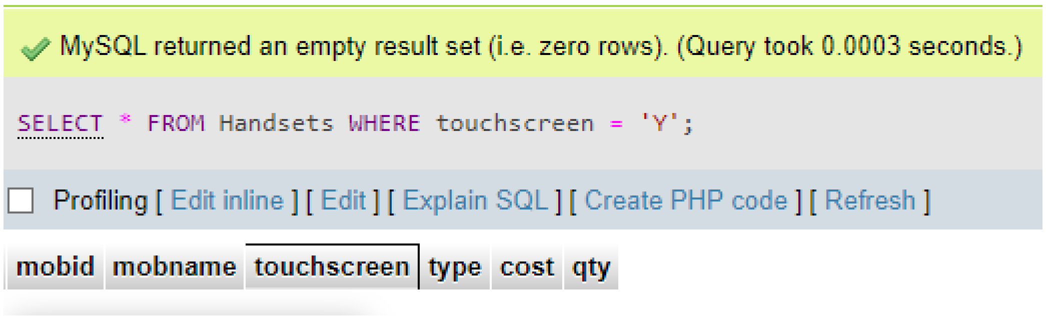
UPDATE Vendor SET city = 'Mumbai' WHERE mobid IN (SELECT mobid FROM Handsets WHERE qty > 6);



```
0 rows affected. (Query took 0.0006 seconds.)

UPDATE Vendor SET city = 'Mumbai' WHERE mobid IN (SELECT mobid FROM Handsets WHERE qty > 6);

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

e) Display handset details where touchscreen = 'Y'.

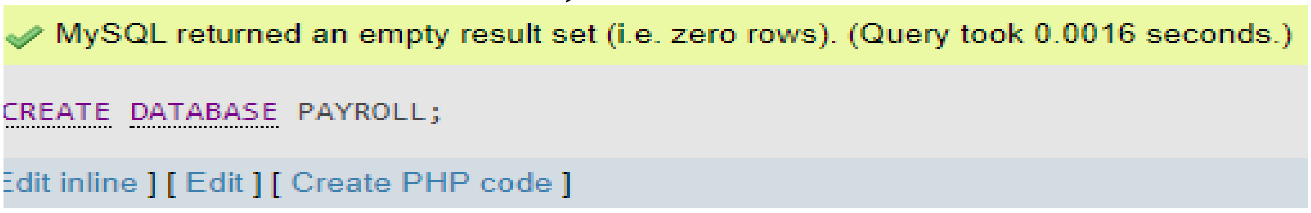
SELECT * FROM Handsets WHERE touchscreen = 'Y';



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0003 seconds.)

SELECT * FROM Handsets WHERE touchscreen = 'Y';

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

mobid  mobname  touchscreen  type  cost  qty
```

10. Create a database PAYROLL with the following tablesSet appropriate datatypes and set primary keys and foreign keys,create relationship between tables and create E-R Model .

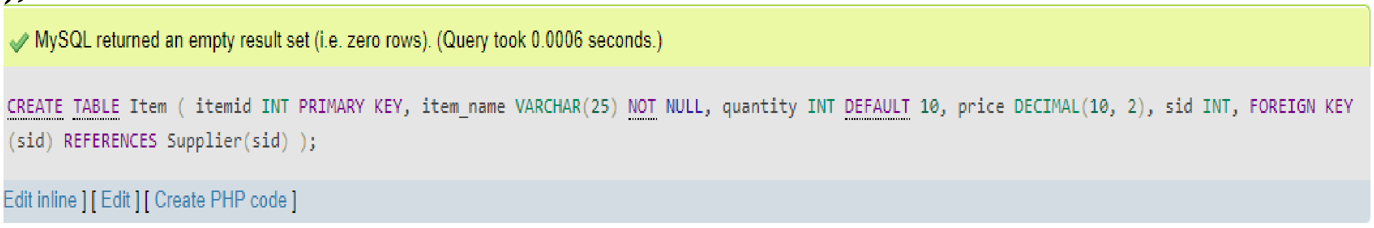
CREATE DATABASE PAYROLL;



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0016 seconds.)

CREATE DATABASE PAYROLL;

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

Item(itemid,item_name,quantity,price,sid)

```
CREATE TABLE  Item (
   itemid INT  PRIMARY KEY,
   item_name VARCHAR(25) NOT NULL,
   quantity INT DEFAULT 10,
   price DECIMAL(10, 2),
   sid INT,
   FOREIGN KEY (sid) REFERENCES Supplier(sid)
);
```



```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

CREATE TABLE Item ( itemid INT PRIMARY KEY, item_name VARCHAR(25) NOT NULL, quantity INT DEFAULT 10, price DECIMAL(10, 2), sid INT, FOREIGN KEY
(sid) REFERENCES Supplier(sid) );

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

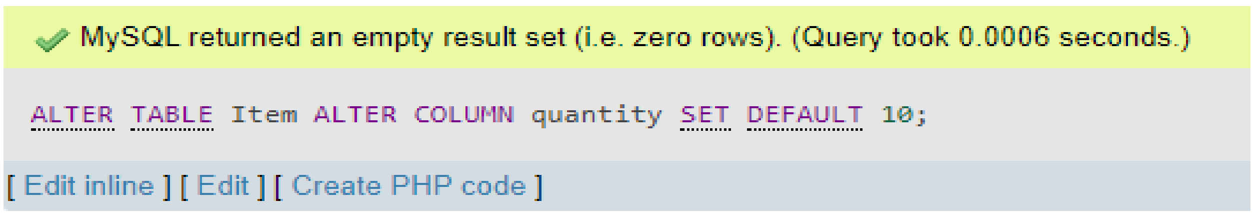
Supplier(sid,sname,contact_no,address)

```
CREATE TABLE Supplier (
   sid INT PRIMARY KEY,
   sname VARCHAR(25) NOT NULL,
   contact_no VARCHAR(20),
   address VARCHAR(25)
);
```

Perform the following queries

a) Default value for quantity should be 10.

ALTER TABLE Item ALTER COLUMN quantity SET DEFAULT 10;



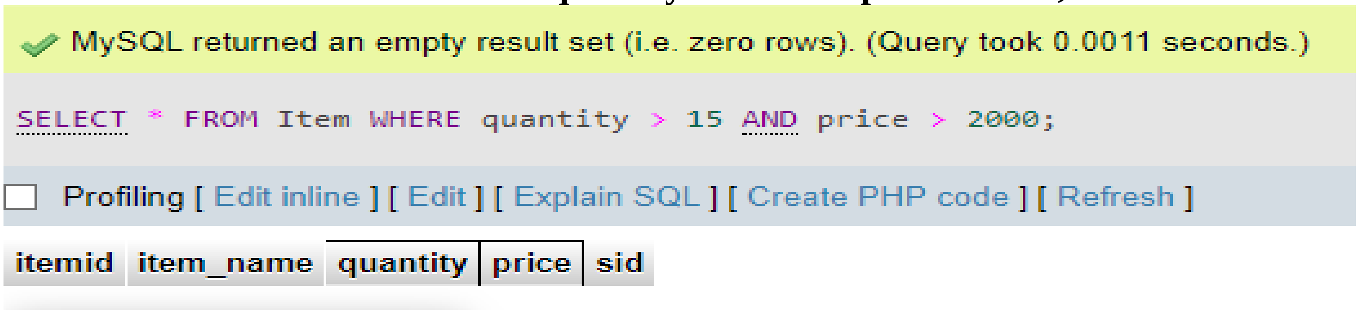
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```
ALTER TABLE Item ALTER COLUMN quantity SET DEFAULT 10;
```

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

b) Update quantity = 15 where price < 2000.

SELECT * FROM Item WHERE quantity > 15 AND price > 2000;



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0011 seconds.)

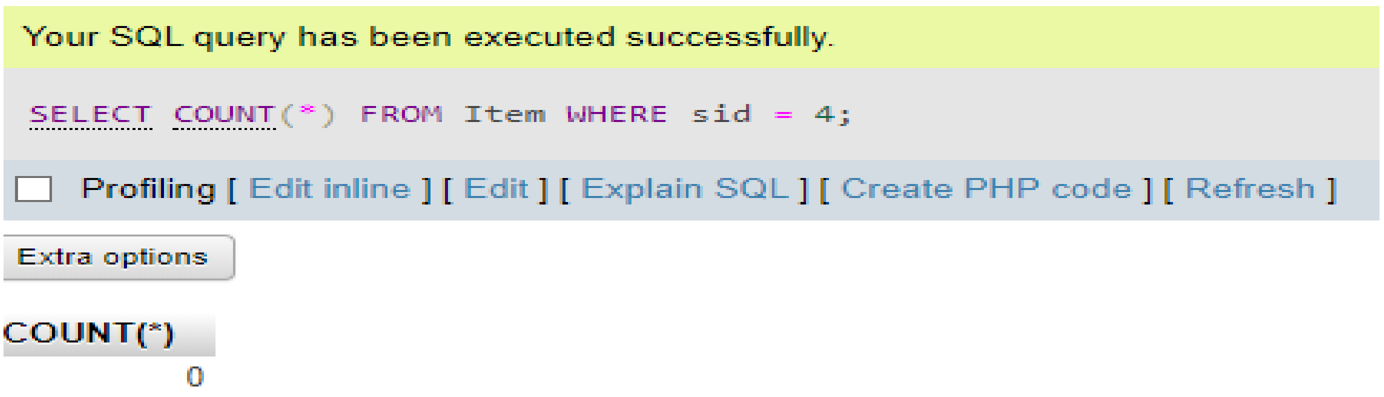
```
SELECT * FROM Item WHERE quantity > 15 AND price > 2000;
```

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

itemid	item_name	quantity	price	sid

c) Count the number of items where sid=4.

SELECT COUNT(*) FROM Item WHERE sid = 4;



Your SQL query has been executed successfully.

```
SELECT COUNT(*) FROM Item WHERE sid = 4;
```

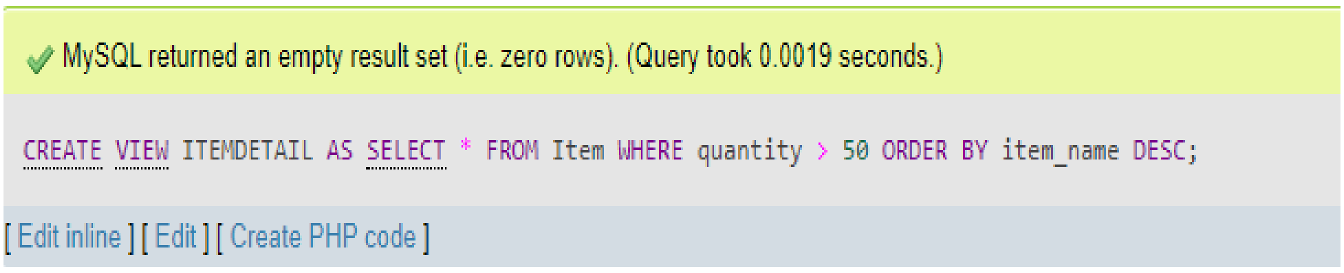
☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

COUNT(*)
0

d) Create a view ITEMDETAIL which displays all the items where quantity is >50 in descending order of item name.

CREATE VIEW ITEMDETAIL AS
SELECT * FROM Item WHERE quantity > 50 ORDER BY item_name DESC;



✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
CREATE VIEW ITEMDETAIL AS SELECT * FROM Item WHERE quantity > 50 ORDER BY item_name DESC;
```

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

11. Create a database School with the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model .
Passenger(pid,pname,age)
Bus(bid,pid,color)
Perform the following queries

a) Change size of color field to 20 in bus table
b) Update color for the busid -11 with blue color.
c) Find passenger whose age is between 20 to 30.
d) Display all bus whose color is blue.

12.Create a database OFFICE_DETTAILwith the following tables Set appropriate datatypes and set primary keys and foreign keys, create relationship between tables and create E-R Model .
EMP(EID,ENAME)
Company(cid,cname,city)
Works(eid,cid,salary)
Salary must be >=20000 and <=80000

a) Add field age in emp table and must be between 18 and 60 5
b) List all employees whose names contains maximum upto 3 character
c) List all employees whose names starts with either 'a' or 'r'.
d) List all companies in ahmedabad.
e) Increase salary of all employees by 500.
f) Delete all companies which are in Vadodara.

❖ <u>THEORY QUESTIONS:</u>

1. Write a short-note on Logical Operators used in SQL.

1. <u>AND Operator:</u>

- The AND operator returns TRUE only if both Boolean expressions are TRUE.
- Example: To find employees in Allahabad, India, we can use

SELECT * FROM employee WHERE emp_city = 'Allahabad' AND emp_country = 'India';

2. IN Operator:

- The IN operator simplifies multiple OR conditions in SELECT, INSERT, UPDATE, or DELETE statements.
- It checks if the operand matches any value in a list.
- Example: To find employees in Allahabad or Patna:
  SELECT * FROM employee WHERE emp_city IN ('Allahabad', 'Patna');

3. NOT Operator:

- The NOT operator reverses the value of any other Boolean operator.
- Example: To find employees whose city does not start with 'A':
  SELECT * FROM employee WHERE emp_city NOT LIKE 'A%';

4. OR Operator:

- The OR operator returns TRUE if either Boolean expression is TRUE.
- Example: To find employees in Varanasi or India: SELECT * FROM employee WHERE emp_city = 'Varanasi' OR emp_country = 'India';

5. LIKE Operator:

- The LIKE operator searches for a specified pattern in a column.
- Use % as a wildcard for zero or more characters.
- Example: To find employees with names starting with 'Utkarsh':

SELECT * FROM employee WHERE emp_name LIKE 'Utkarsh%';

2. Explain ALTER command with different modifiers.

1. Adding Columns:

- To add a new column to an existing table, use the following syntax:
- ALTER TABLE table_name ADD column_name data_type;
  For example, if we want to add an "Age" column to the "Students" table:
  ALTER TABLE Students ADD Age INT;

2. Dropping Columns:

- To remove unwanted columns from a table, use the DROP COLUMN modifier:
- ALTER TABLE table_name DROP COLUMN column_name;
- For instance, to drop the "Email" column from the "Students" table:
- ALTER TABLE Students DROP COLUMN Email;

3. Modifying Columns:

- The MODIFY modifier allows you to alter the data type or other properties of an existing column:
- In SQL Server
  ALTER TABLE table_name ALTER COLUMN column_name new_data_type;
- In MySQL or Oracle

```
ALTER TABLE table_name MODIFY COLUMN column_name new_data_type;
```

For example, to modify the data type of the "Course" column in the "Student" table:

```
ALTER TABLE Student MODIFY COLUMN Course VARCHAR(20);
```

3. Write a short-note on Aggregate Functions.

1. SUM():
- Calculates the sum of numeric values in a column.
- Example: To find the total salary of all employees:
  ```
  SELECT SUM(salary) FROM employees;
  ```
2. AVG():
- Computes the average (mean) of numeric values in a column.
- Useful for finding average scores, ratings, or other continuous data.
- Example: To determine the average age of customers
  ```
  SELECT AVG(age) FROM customers;
  ```
3. COUNT():
- Counts the number of rows that match a condition.
- Can be used with DISTINCT to count unique values.
- Example: To count the number of orders:
  ```
  SELECT COUNT(order_id) FROM orders;
  ```
4. MIN() and MAX():
- MIN() returns the smallest value in a column.
- MAX() returns the largest value in a column.
- Example: To find the oldest and youngest employees:
  ```
  SELECT MIN(age), MAX(age) FROM employees;
  ```
5. GROUP_CONCAT() :
- Concatenates values from multiple rows into a single string.
- Useful for creating comma-separated lists.
- Example: To list all products purchased by a customer:
  ```
  SELECT customer_id, GROUP_CONCAT(product_name) AS purchased_products
  FROM orders
  GROUP BY customer_id;
  ```

4. Write a short-note on SQL Constraints

1. NOT NULL Constraint:
- Ensures that a column cannot contain NULL values.
- When defining a column, you can specify it as NOT NULL:

- In the example above, the emp_name column must always have a non-null value.

```
CREATE TABLE employees (
emp_id INT PRIMARY KEY,
emp_name VARCHAR(50) NOT NULL,
emp_salary DECIMAL(10, 2)
);
```

2. <u>UNIQUE Constraint:</u>

- Ensures that all values in a column are unique (no duplicates).
- Useful for columns like email addresses or usernames.
- Example:

```
CREATE TABLE customers (
customer_id INT PRIMARY KEY,
email VARCHAR(100) UNIQUE,
phone VARCHAR(15)
);
```

- The email column must have distinct values

3. <u>PRIMARY KEY Constraint:</u>

- Combines the properties of NOT NULL and UNIQUE.
- Uniquely identifies each row in a table.
- Typically applied to a single column or a combination of columns.
- Example:

```
 CREATE TABLE orders (
 order_id INT PRIMARY KEY,
customer_id INT,
order_date DATE
 );
```

- The order_id ensures uniqueness for each order

4. <u>FOREIGN KEY Constraint:</u>

- Establishes a relationship between tables.
- Ensures referential integrity by linking a column to a primary key in another table.
- Example:

```
CREATE TABLE order_items (
item_id INT PRIMARY KEY,
order_id INT,
product_id INT,
FOREIGN KEY (order_id) REFERENCES orders(order_id)
);
```

5. <u>CHECK Constraint:</u>

- Validates that values in a column satisfy a specific condition.
- Example:

```
CREATE TABLE students (
 student_id INT PRIMARY KEY,
 age INT CHECK (age >= 18),
 grade CHAR(1) CHECK (grade IN ('A', 'B', 'C'))
 );
```

- The age must be 18 or older, and the grade must be ‘A’, ‘B’, or ‘C’.

6. DEFAULT Constraint:

- Sets a default value for a column if no value is specified during insertion.
- Example:

```
CREATE TABLE products (
product_id INT PRIMARY KEY,
product_name VARCHAR(50),
price DECIMAL(10, 2) DEFAULT 0.00
);
```

- If no price is provided, it defaults to $0.00.

7. <u>CREATE INDEX:</u>

- Used to create an index on one or more columns.
- Improves query performance by allowing faster data retrieval.
- Example:
  CREATE INDEX idx_product_name ON products (product_name);
  Creates an index on the product_name column.

5. Difference between DDL and DML, give example in support of your answer.

<u>DDL (Data Definition Language):</u>

- Purpose: DDL is used to define the structure of a database. It focuses on creating and modifying database objects such as tables, indexes, views, and constraints.
- Commands: Common DDL commands include CREATE, ALTER, DROP, RENAME, and TRUNCATE.
- Irreversible: DDL statements modify the database’s schema, but they have no direct effect on the data within the database.
- Examples:

  Creating a new table:

```
CREATE TABLE Students (
   student_id INT PRIMARY KEY,
   student_name VARCHAR(50),
   age INT
```

);

- Adding a new column to an existing table:

```
ALTER TABLE Students ADD email VARCHAR(100);
```

DML (Data Manipulation Language):

- Purpose: DML deals with manipulating the data stored in the database. It allows you to insert, update, retrieve, and delete data.
- Commands: Common DML commands include SELECT, INSERT, UPDATE, DELETE, and MERGE.
- Direct Impact: DML statements directly affect the data within the database.
- Reversibility: In case of errors, data can be recovered due to the reversibility of DML statements.
- Examples:
  - Inserting a new student record:
    INSERT INTO Students (student_id, student_name, age, email) VALUES (101, 'Alice', 20, 'alice@example.com');
  - Updating an existing record:
    UPDATE Students SET age = 21 WHERE student_id = 101;
  - Deleting a record:
    DELETE FROM Students WHERE student_id = 101;

6. Explain SQL datatype.

1. String Data Types:

- CHAR(size): A fixed-length string that can contain letters, numbers, and special characters. The size parameter specifies the column length in characters (ranging from 0 to 255).
- VARCHAR(size): A variable-length string that can also contain letters, numbers, and special characters. The size parameter specifies the maximum string length in characters (ranging from 0 to 65535).
- BINARY(size): Similar to CHAR(), but stores binary byte strings. The size parameter specifies the column length in bytes.
- VARBINARY(size): Similar to VARCHAR(), but stores binary byte strings. The size parameter specifies the maximum column length in bytes.

2. Numeric Data Types:

- BIT(size): A bit-value type with a specified number of bits per value (ranging from 1 to 64).
- TINYINT(size): A very small integer. Signed range is from -128 to 127, and unsigned range is from 0 to 255.

- Other numeric data types
  include SMALLINT, INT, BIGINT, FLOAT, DOUBLE, and DECIMAL.

3. Date and Time Data Types:

- DATE: Stores date values (e.g., ‘2024-03-15’).
- TIME: Stores time values (e.g., ‘14:30:00’).
- DATETIME or TIMESTAMP: Stores both date and time values.
- YEAR: Stores year values (e.g., ‘2024’).

4. Other Common Data Types:

- ENUM(val1, val2, val3, …): A string object with a predefined list of possible values. You can list up to 65535 values.
- SET(val1, val2, val3, …): A string object that can have zero or more values from a list of possible values (up to 64).