MODULE 5: DATABASE

1)What do you understand By Database.

Database is a collection of inter-related data, events and transactions. It is organised in a particular manner and provide access to the various users simultaneously. In this system, data is gathered and stored for a particular operation.

2) What is Normalization?

Normalization is the process of elemeneting theredundancy,duplication,inconsitancy of the table form elemeting columns.

Normalization is the process of organizing a database to reduce redundancy and improve data integrity.

Also referred to as database normalization or data normalization, normalization is an important part of relational database design, as it helps with the speed, accuracy, and efficiency of the database.

By normalizing a database, you arrange the data into tables and columns. You ensure that each table contains only related data. If data is not directly related, you create a new table for that data.

3)What is Difference between DBMS and RDBMS?

DBMS:- 1) in dbms realtionship between two table or file are maintained programatically.

2)DBMS does not support client-server.

3)DBMS does not support distributed.

4)No security.

RDBMS:-1)in rdbms relationsip between two table or file can be specified at the time of table creation.

2)RDBMS support client-server architecture.

3)RDBMS support distributed.

4)multiple levels of secqurity.a)log in at os level b)commend level c)object level.

4)What is MF Cod Rule of RDBMS Systems?

Rule 0: The Foundation Rule

The database must be in relational form. So that the system can handle the database through its relational capabilities.

Rule 1: Information Rule

A database contains various information, and this information must be stored in each cell of a table in the form of rows and columns.

Rule 2: Guaranteed Access Rule

Every single or precise data (atomic value) may be accessed logically from a relational database using the combination of primary key value, table name, and column name.

Rule 3: Systematic Treatment of Null Values

This rule defines the systematic treatment of Null values in database records. The null value has various meanings in the database, like missing the data, no value in a cell, inappropriate information, unknown data and the primary key should not be null.

Rule 4: Active/Dynamic Online Catalog based on the relational model

It represents the entire logical structure of the descriptive database that must be stored online and is known as a database dictionary. It authorizes users to access the database and implement a similar query language to access the database.

Rule 5: Comprehensive Data SubLanguage Rule

The relational database supports various languages, and if we want to access the database, the language must be the explicit, linear or well-defined syntax, character strings and supports the comprehensive: data definition, view definition, data manipulation, integrity constraints, and limit transaction management operations. If the database allows access to the data without any language, it is considered a violation of the database.

Rule 6: View Updating Rule

All views table can be theoretically updated and must be practically updated by the database systems.

Rule 7: Relational Level Operation (High-Level Insert, Update and delete) Rule

A database system should follow high-level relational operations such as insert, update, and delete in each level or a single row. It also supports union, intersection and minus operation in

the database system.

Rule 8: Physical Data Independence Rule

All stored data in a database or an application must be physically independent to access the database. Each data should not depend on other data or an application. If data is updated or the physical structure of the database is changed, it will not show any effect on external applications that are accessing the data from the database.

Rule 9: Logical Data Independence Rule

It is similar to physical data independence. It means, if any changes occurred to the logical level (table structures), it should not affect the user's view (application). For example, suppose a table either split into two tables, or two table joins to create a single table, these changes should not be impacted on the user view application.

Rule 10: Integrity Independence Rule

A database must maintain integrity independence when inserting data into table's cells using the SQL query language. All entered values should not be changed or rely on any external factor or application to maintain integrity. It is also helpful in making the database-independent for each front-end application.

Rule 11: Distribution Independence Rule

The distribution independence rule represents a database that must work properly, even if it is stored in different locations and used by different end-users. Suppose a user accesses the database through an application; in that case, they should not be aware that another user uses particular data, and the data they always get is only located on one site. The end users can access the database, and these access data should be independent for every user to perform the SQL queries.

Rule 12: Non Subversion Rule

The non-submersion rule defines RDBMS as a SQL language to store and manipulate the data in the database. If a system has a low-level or separate language other than SQL to access the database system, it should not subvert or bypass integrity to transform data.

5)What do you understand By Data Redundancy?

It is defined as the redundancy means duplicate data and it is also stated that the same parts of data exist in multiple locations into the database. This condition is known as Data Redundancy.

Data redundancy can occur within an organization intentionally or accidentally. If done intentionally, the same data is kept in different locations with the organization making a conscious effort to protect it and ensure its consistency. This data is often used for backups or disaster recovery.

6) What is DDL Interpreter?

DDL stands for Data Definition Language. As the name suggests, the DDL commands help to define the structure of the databases or schema. When we execute DDL statements, it takes effect immediately. The changes made in the database using this command are saved permanently because its commands are auto-committed. The following commands come under DDL language:

CREATE: It is used to create a new database and its objects such as table, views, function, stored procedure, triggers, etc.

DROP: It is used to delete the database and its objects, including structures, from the server permanently.

ALTER: It's used to update the database structure by modifying the characteristics of an existing attribute or adding new attributes.

TRUNCATE: It is used to completely remove all data from a table, including their structure and space allocates on the server.

RENAME: This command renames the content in the database.

7) What is DML Compiler in SQL?

DML stands for Data Manipulation Language. DML compiler translates the DML statements which are there in a query language into the low-level instructions which the query evaluation engine understands easily.

8) What is SQL Key Constraints writing an Example of SQL Key Constraints ?

Constraints ensure that data entered by the user into columns must be within the criteria specified by the condition

For example, if you want to maintain only unique IDs in the employee table or if you want to enter only age under 18 in the student table etc

We have 5 types of key constraints in DBMS

NOT NULL: ensures that the specified column doesn’t contain a NULL value.

UNIQUE : provides a unique/distinct values to specified columns.

DEFAULT: provides a default value to a column if none is specified.

CHECK :checks for the predefined conditions before inserting the data inside the table.

PRIMARY KEY: it uniquely identifies a row in a table.

FOREIGN KEY: ensures referential integrity of the relationship

```
CREATE TABLE users(
id int PRIMARY KEY AUTO_INCREMENT,
name varchar(50),
email varchar(50) UNIQUE,
salary int NOT NULL,
gender varchar(50) DEFAULT 'MALE',
age int CHECK(age<18)
);
```

9) What is save Point? How to create a save Point write a Query?

Savepoint is a command in SQL that is used with the rollback command. It is a command in Transaction Control Language that is used to mark the transaction in a table. Consider you are making a very long table, and you want to roll back only to a certain position in a table then; this can be achieved using the savepoint.

```
START TRANSACTION;
SELECT * FROM EMP;
UPDATE EMP SET AGE = AGE + 1;
SAVEPOINT samplesavepoint;
INSERT INTO EMP ('Mac', 'Mohan', 26, 'M', 2000);
ROLLBACK TO SAVEPOINT samplesavepoint;
```

COMMIT;

10) What is trigger and how to create a Trigger in SQL?

- Trigger is the same as the procedure

- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.

Type Trigger

1) After Insert

2) After Update

3) After Delete

4) Before Insert

5) Before Update

6) Before Delete

Create a trigger in sql

create trigger [trigger_name]

[before | after]

{insert | update | delete}

on [table_name]

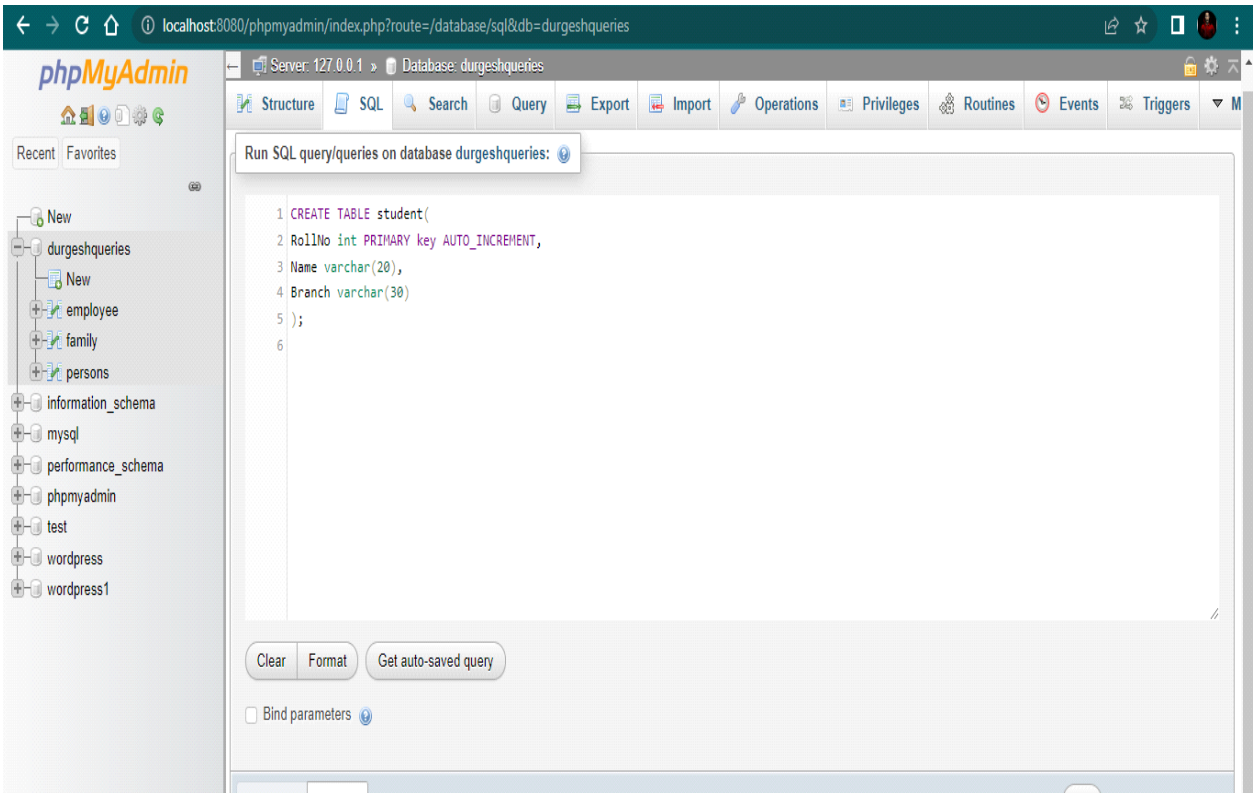
[for each row]

[trigger_body]

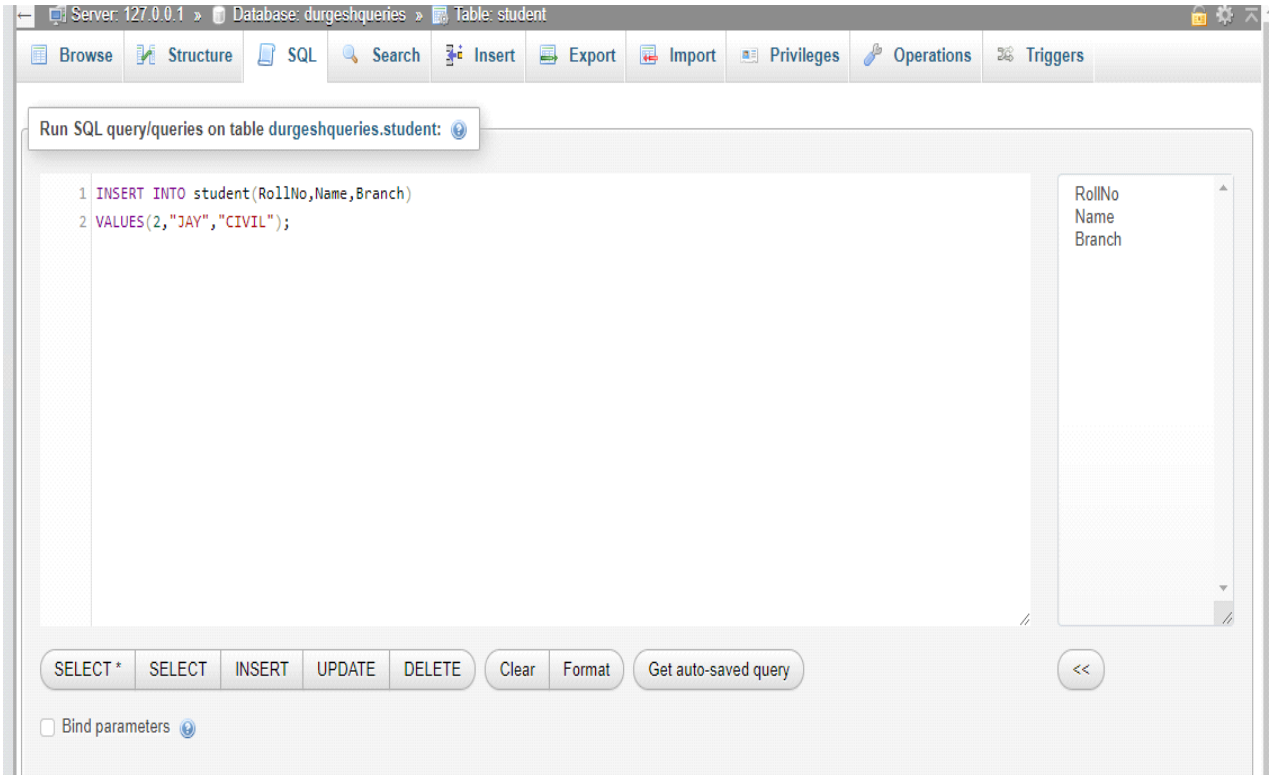
Task:

1)Create Table Name : Student and Exam

CREATE TABLE STUDENT:



INSERT INTO TABLE:**ANS:**



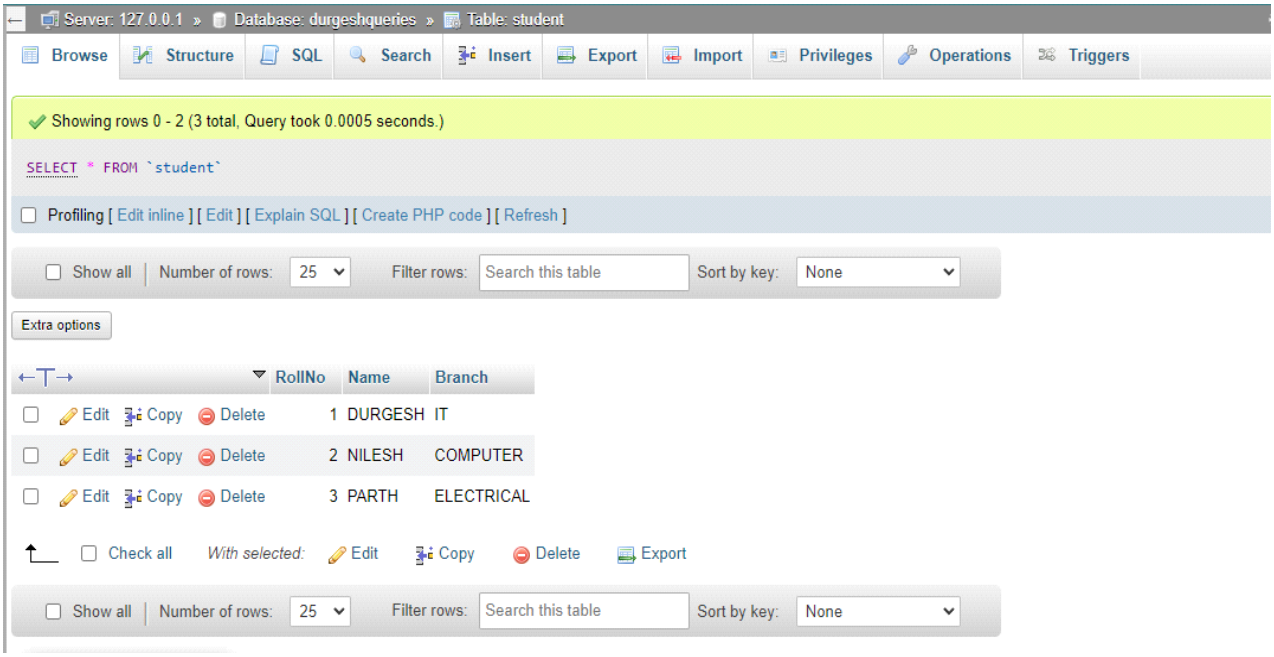


TABLE EXAM USING FORIGIN KEY:

create table examination:

Server: 127.0.0.1 » Database: durgeshqueries

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events | Triggers

Run SQL query/queries on database durgeshqueries:

```
CREATE table examination(
rollnum int,
s_code varchar(40),
marks int,
p_code varchar(50),
FOREIGN KEY(rollnum) REFERENCES student(Rollno))
```

Clear | Format | Get auto-saved query

table examination:

Server: 127.0.0.1 » Database: durgeshqueries » Table: examination

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

```
SELECT * FROM `examination`
```

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

Extra options

rollnum	s_code	marks	p_code
1	com11	70	com
1	com12	60	com
2	com11	66	com
2	com12	70	com
3	civ101	45	ce
3	civ102	50	ce

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

3)Create table given below: Employee and Incentive

create table employee

Server: 127.0.0.1 » Database: durgeshqueries » Table: employees

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Run SQL query/queries on table durgeshqueries.employees:

```
CREATE TABLE Employees(
emp_id int,
first_name varchar(30),
last_name varchar(40),
salary INT NOT NULL,
joining_date date,
department varchar(40))
```

emp_id
first_nam
last_nam
salary
joining_d
departme

SELECT * | SELECT | INSERT | UPDATE | DELETE | Clear | Format | Get auto-saved query | <<

Bind parameters

1)table employee:

Server: 127.0.0.1 » Database: durgeshqueries » Table: employees

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

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

Showing rows 0 - 7 (8 total, Query took 0.0004 seconds.)

```
SELECT * FROM `employees`
```

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

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

Extra options

emp_id	first_name	last_name	salary	joining_date	department
1	john	abraham	1000000	2013-01-01	banking
2	michael	clarke	800000	2013-01-01	insurance
3	roy	thomas	700000	2013-02-01	banking
4	torm	josh	600000	2013-02-01	insurance
5	jerrry	pinto	650000	2013-02-01	insurance
6	philip	mathew	750000	2013-01-01	services
7	testname1	123	650000	2013-01-01	services
8	testname2	iname%	600000	2013-02-01	insurance

Table creat incentive

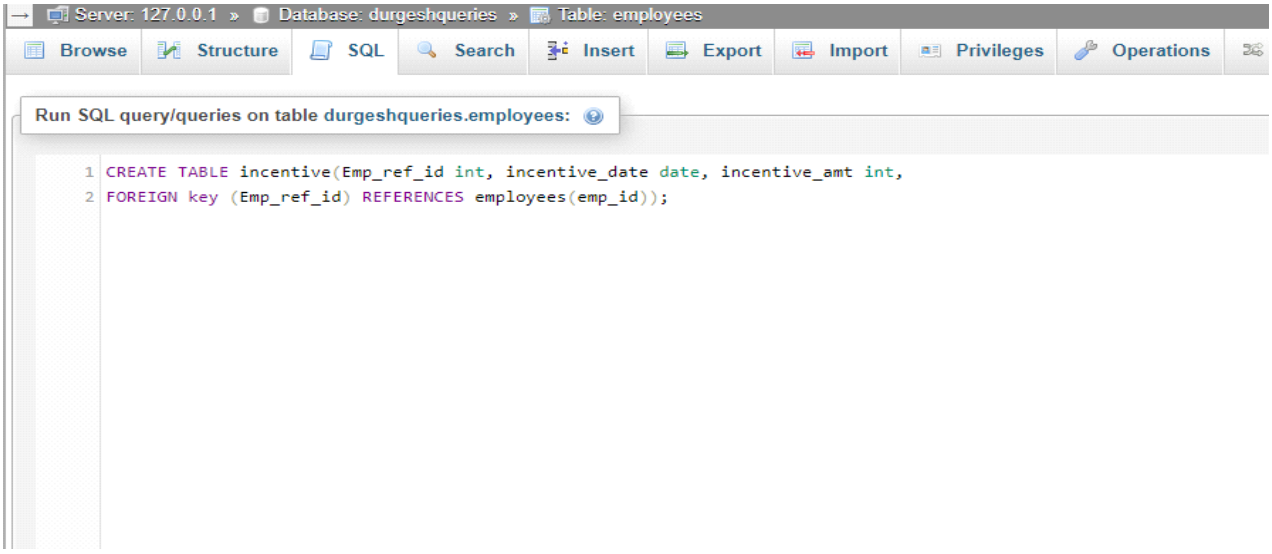


Table incentive:

Emp_ref_id	incentive_date	incentive_amt
1	2013-02-01	5000
2	2013-02-01	3000
3	2013-02-01	4000
1	2013-01-01	4500
2	2013-01-01	3500

a) Get First_Name from employee table using Tom name “Employee Name”.

Server: 127.0.0.1 » Database: durgeshqueries » Table: employees

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Show query box

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

Showing rows 0 - 7 (8 total, Query took 0.0006 seconds.)

```
SELECT first_name AS employee_name FROM `employees`;
```

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

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

Extra options

employee_name
john
michael
roy
torm
jerrry
philip
testname1
testname2

b) Get FIRST_NAME, Joining Date, and Salary from employee table.

Server: 127.0.0.1 » Database: durgeshqueries » Table: employees

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Show query box

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

Showing rows 0 - 7 (8 total, Query took 0.0008 seconds.)

```
SELECT first_name,joining_date,salary FROM `employees`;
```

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

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

Extra options

first_name	joining_date	salary
john	2013-01-01	1000000
michael	2013-01-01	800000
roy	2013-02-01	700000
torm	2013-02-01	600000
jerrry	2013-02-01	650000
philip	2013-01-01	750000
testname1	2013-01-01	650000
testname2	2013-02-01	600000

c) Get all employee details from the employee table order by First_Name Ascending and Salary descending?

1)Ascending

Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.) [first_name: **JERRRY... - TORM...**]

```
SELECT * FROM `employees` ORDER BY first_name ASC;
```

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

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

Extra options

emp_id	first_name ▲ 1	last_name	salary	joining_date	department
5	jerrry	pinto	650000	2013-02-01	insurance
1	john	abraham	1000000	2013-01-01	banking
2	michael	clarke	800000	2013-01-01	insurance
6	philip	mathew	750000	2013-01-01	services
3	roy	thomas	700000	2013-02-01	banking
7	testname1	123	650000	2013-01-01	services
8	testname2	iname%	600000	2013-02-01	insurance
4	torm	josh	600000	2013-02-01	insurance

2)descending

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Show query box

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

Showing rows 0 - 7 (8 total, Query took 0.0005 seconds.) [salary: **1000000... - 600000...**]

```
SELECT * FROM `employees` ORDER BY salary DESC;
```

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

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

Extra options

emp_id	first_name	last_name	salary ▼ 1	joining_date	department
1	john	abraham	1000000	2013-01-01	banking
2	michael	clarke	800000	2013-01-01	insurance
6	philip	mathew	750000	2013-01-01	services
3	roy	thomas	700000	2013-02-01	banking
5	jerrry	pinto	650000	2013-02-01	insurance
7	testname1	123	650000	2013-01-01	services
4	torm	josh	600000	2013-02-01	insurance
8	testname2	iname%	600000	2013-02-01	insurance

d) Get employee details from employee table whose first name contains ‘J’

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Show query box

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

Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT * FROM `employees` WHERE first_name LIKE "j%";
```

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

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

Extra options

emp_id	first_name	last_name	salary	joining_date	department
1	john	abraham	1000000	2013-01-01	banking
5	jerrry	pinto	650000	2013-02-01	insurance

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

Query results operations

e)Get department wise maximum salary from employee table order by salary ascending?

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Triggers

Show query box

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

Showing rows 0 - 2 (3 total, Query took 0.0406 seconds.)

```
SELECT * FROM `employees` GROUP BY department ORDER BY MAX(salary);
```

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

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

Extra options

emp_id	first_name	last_name	salary	joining_date	department
6	philip	mathew	750000	2013-01-01	services
2	michael	clarke	800000	2013-01-01	insurance
1	john	abraham	1000000	2013-01-01	banking

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

f) Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

```
SELECT FIRST_NAME,INCENTIVE_AMT FROM EMPLOYEE INNER JOIN INCENTIVE ON EMP_ID=EMP_REF_ID AND INCENTIVE_AMT >3000;
```

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

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

Extra options

first_name	incentive_amt
john	5000
roy	4000
john	4500
michael	3500

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

Query results operations

Print Copy to clipboard Export Display chart Create view

g)Create After Insert trigger on Employee table which insert records in view table.

Server: 127.0.0.1 » Database: durgeshqueries » Table: employes

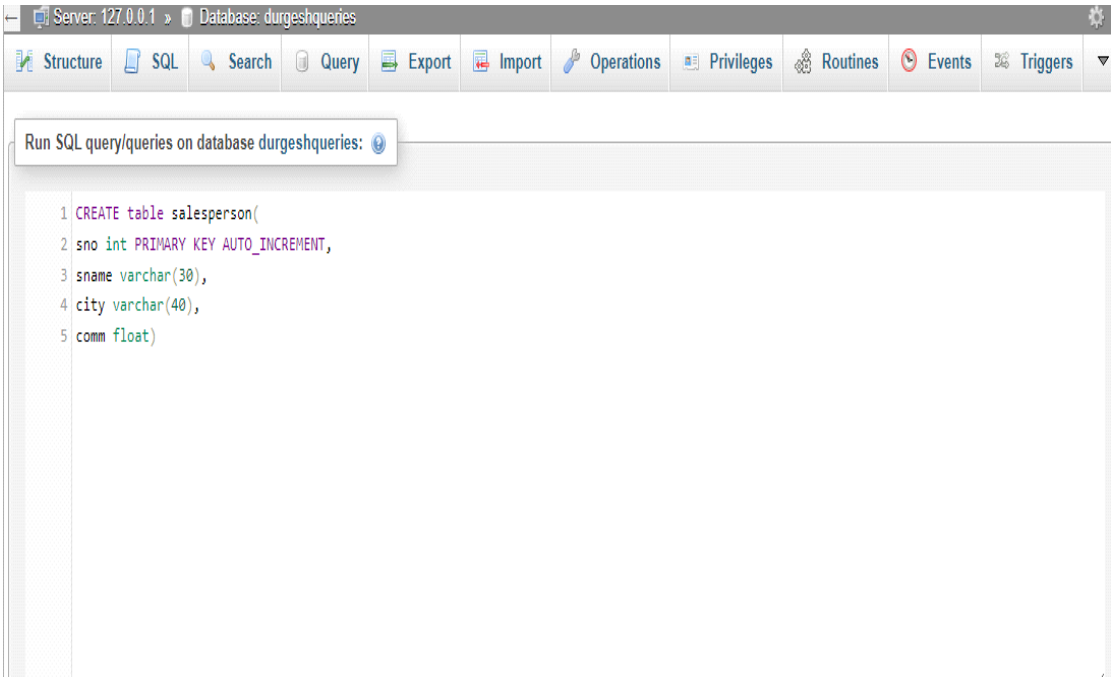
Browse Structure SQL Search Insert Export Import Privileges Operations

Run SQL query/queries on table durgeshqueries.employes:

```
DELIMITER//
CREATE TRIGGER after_insert
AFTER INSERT
ON employes
FOR EACH ROW
BEGIN
INSERT INTO emploee_log(emp_id,insert_time)VALUES(new.emp_id,CURRENT_TIME())
END//
```

4)Create table given below: Salesperson and Customer

salseperdon create table:



insert salseperson:

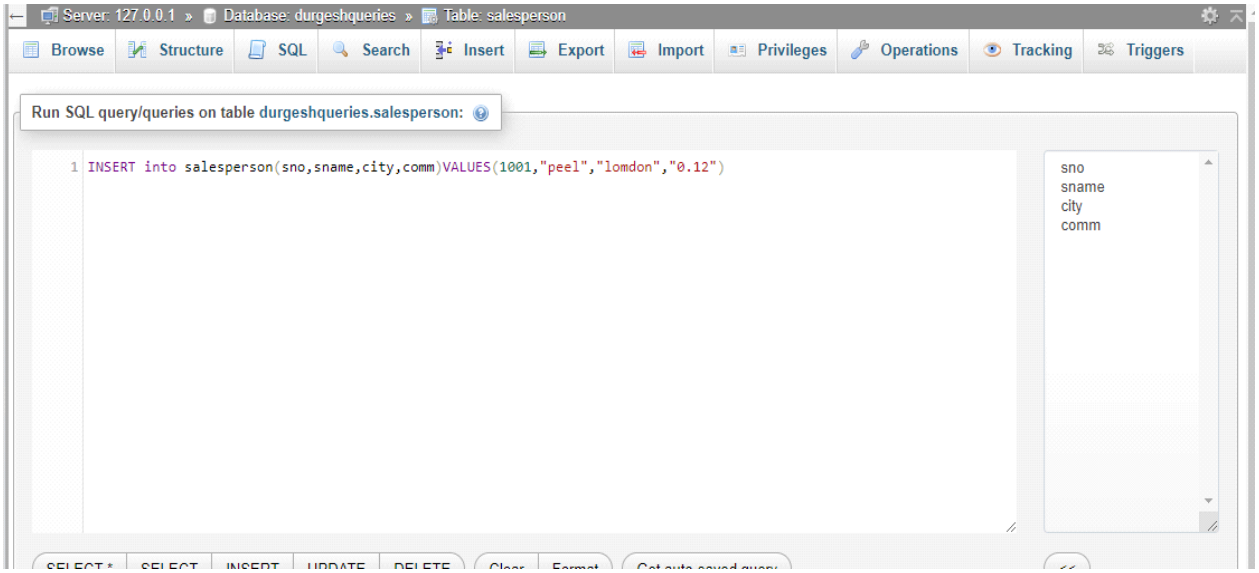


table selsperson:

Showing rows 0 - 4 (5 total, Query took 0.0005 seconds.)

```
SELECT * FROM `salesperson`
```

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

Extra options

	sno	sname	city	comm
Edit Copy Delete	1001	peel	london	0.12
Edit Copy Delete	1002	serres	san jose	0.13
Edit Copy Delete	1003	axelrod	new york	0.1
Edit Copy Delete	1004	motika	london	0.11
Edit Copy Delete	1007	rafkin	barcelona	0.15

Check all With selected: Edit Copy Delete Export

2)crearing table customer

Server: 127.0.0.1 » Database: durgeshqueries » Table: customer

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking | Triggers

Run SQL query/queries on table durgeshqueries.customer:

```
CREATE TABLE customers(
cnm int PRIMARY key AUTO_INCREMENT,
cname varchar(50),
city varchar(50),
rating int,
sno int,
FOREIGN KEY(sno)REFERENCES salesperson(sno))
```

cnm
cname
city
rating
sno

SELECT * | SELECT | INSERT | UPDATE | DELETE | Clear | Format | Get auto-saved query | <<

Bind parameters

insert into customer table:

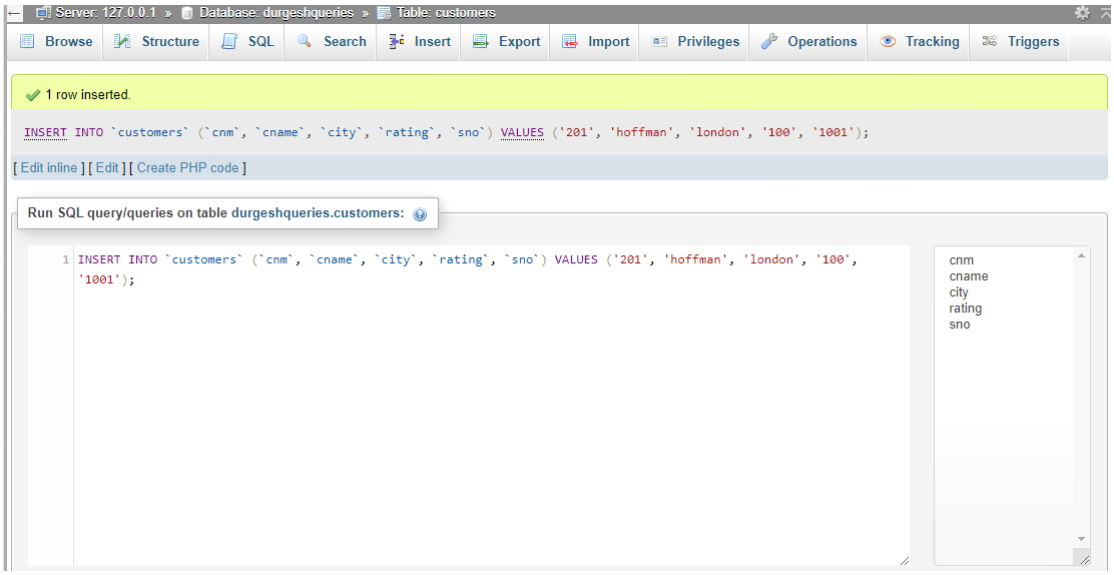
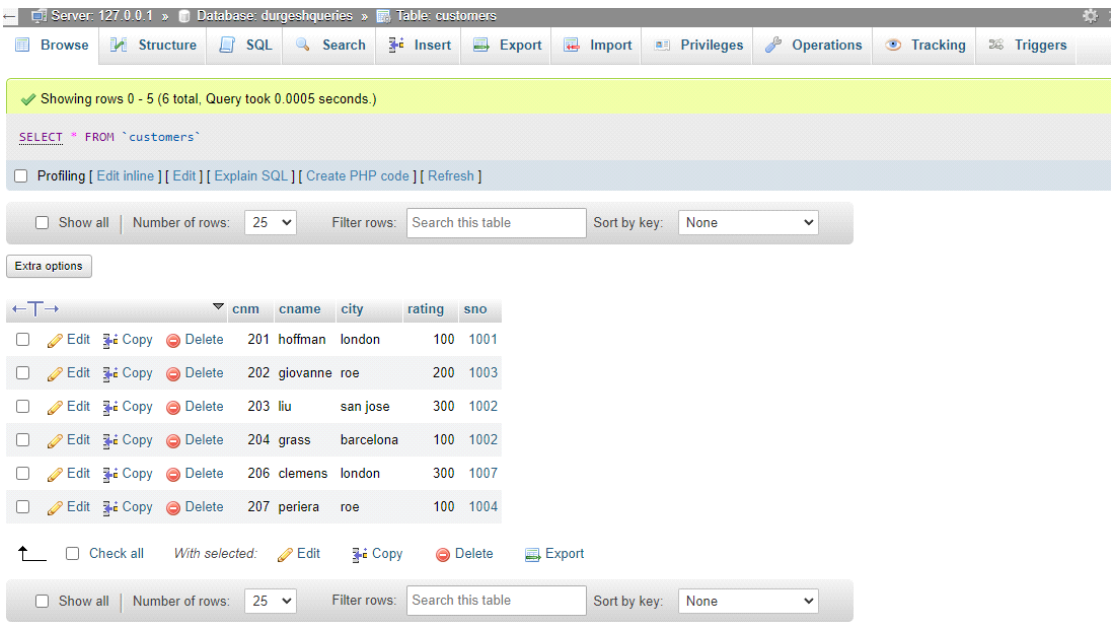


table customer:



a) All orders for more than $1000.

```
select * FROM orders where Amt>1000;
```

				onm	Amt	ode	cnm	sno
☐	Edit	Copy	Delete	3002	1900.1	1994-10-03	207	1004
☐	Edit	Copy	Delete	3005	3005	1994-10-03	203	1002
☐	Edit	Copy	Delete	3007	3007	1994-10-05	204	1002
☐	Edit	Copy	Delete	3008	3008	1994-10-05	206	1001
☐	Edit	Copy	Delete	3009	3009	1994-10-04	202	1003
☐	Edit	Copy	Delete	3010	3010	1994-10-06	204	1002
☐	Edit	Copy	Delete	3011	3011	1994-10-06	206	1001

b) Names and cities of all salespeople in London with commission above 0.12

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations

Show query box

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

```
SELECT * FROM `salesperson` WHERE city = "%london" and comm>0.12;
```

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

sno	sname	city	comm

Query results operations

Create view

Bookmark this SQL query

Label: ☐ Let every user access this bookmark

Bookmark this SQL query

c)All salespeople either in Barcelona or in London

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `salesperson` WHERE city="london" OR city="barcelona";
```

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

Extra options

	sno	sname	city	comm
Edit Copy Delete	1001	peel	london	0.12
Edit Copy Delete	1004	motika	london	0.11
Edit Copy Delete	1007	rafkin	barcelona	0.15

Check all *With selected:* Edit Copy Delete Export

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

d) All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `salesperson` WHERE comm >0.1 AND comm<0.12;
```

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

Extra options

	sno	sname	city	comm
Edit Copy Delete	1001	peel	london	0.12
Edit Copy Delete	1003	axelrod	new york	0.1
Edit Copy Delete	1004	motika	london	0.11

Check all *With selected:* Edit Copy Delete Export

e) All customers excluding those with rating <= 100 unless they are located in Rome

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

```
SELECT * FROM `customers` WHERE rating <=100 and city ="roe";
```

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

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

Extra options

	cnm	cname	city	rating	sno
Edit Copy Delete	207	periera	roe	100	1004

Check all *With selected:* Edit Copy Delete Export

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

Query results operations