CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO SQL

MySQL is a leading open source database management system. It is a multi-user, multithreaded database management system. MySQL is especially popular on the web. It is one of the parts of the very popular LAMP platform. Linux, Apache, MySQL, and PHP. MySQL database is available on most important OS platforms. It runs on BSD Unix, Linux, Windows or Mac. Wikipedia, YouTube, Facebook use MySQL. These sites manage millions of queries each day. MySQL comes in two versions: MySQL server system and MySQL embedded system. The MySQL server software and the client libraries are dual-licensed: GPL version 2 and proprietary license.

The development of MySQL began in 1994 by a Swedish company MySQL AB. Sun Microsystems acquired MySQL AB in 2008. Sun was bought by Oracle in 2010.

MySQL, PostgreSQL, Firebird, SQLite, Derby, and HSQLDB are the most well known open source database systems.

MySQL is developed in C/C++. Except for C/C++, APIs exist for PHP, Python, Java, C#, Eiffel, Ruby, Tcl or Perl

1.2 INTRODUCTION TO FRONT END SOFTWARE

HyperText Markup Language (HTML) :

HTML is a simple markup system used to create hypertext documents that are portable from one platform to another. HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. HTML markup can represent hypertext news, mail, documentation, and hypermedia; menus of options; database query results; simple structured documents with in-lined graphics; and hypertext views of existing bodies of information.

CSS:

CSS stands for Cascading Style Sheets.
CSS describes how HTML elements are to be displayed on the screen, paper, or in other media.
CSS saves a lot of work. It can control the layout of multiple web pages all at once External stylesheets are stored in CSS files.
CSS is used to define styles for your web pages, including the design, layout, and variations in the display for different devices and screen sizes.

PHP(recursive acronym for PHP: Hypertext Pre-processor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP's development is focused on server-side scripting. Instead of a lot of commands to output HTML, PHP pages contain HTML with embedded code that does "something". The PHP code is enclosed in special start and end processing instructions that allow you to jump into and out of "PHP mode". The best thing in using PHP is that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer.

1.3 PROJECT REPORT OUTLINE

The report is arranged in the following way:

Chapter 1:	INTRODUCTION
Chapter 2:	REQUIREMENT SPECIFICATION
Chapter 3:	OBJECTIVE OF THE PROJECT
Chapter 4:	PROJECT IMPLEMENTATION
Chapter 5:	FRONT END DESIGN
Chapter 6:	TESTING
Chapter 7:	RESULTS

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

Operating System :	Windows 7 and above
Database :	MySQL
Tools :	Notepad++, WAMP Server.

2.2 HARDWARE REQUIREMENTS

Processor :	Any Processor above 500 MHz
RAM :	Any Ram above 2GB
Hard Disk :	250MB and above
Compact Disk :	Not Required
Input device :	Keyboard, Mouse
Output device :	Monitor
System type:	32-bit Operating System

CHAPTER 3

OBJECTIVE OF THE PROJECT

This project is aimed at developing an application for the Training and Placement Department of the College. The system is an application that can be accessed throughout the organization. This system can be used as an application for the Training and Placement Officers of the Company to manage the student information with regard to placement. Students should be able to upload their information in the form as a CV. The key feature of this project is that it is a hassle-free delivery of respected students information and also provides a requested list of candidates to recruit the students based on a given query. The administrator can also check in on the information uploaded by the students. This project will aid Company’s Placement Cell to practice full IT deployment. This will also help in fast access procedures in placement related activities.

ABOUT PRESENT SYSTEM

In Various companies, training and placement officers have to manage the students profile and documents of students for their training and placement manually.

Placement Officer have to collect the information from various colleges who want to place students and notify students about them.

Placement Officer have to arrange profiles of students according to various streams and notify them according to company requirements.

If any modifications or updates are required in the profile of any student, it has to searched and to be done it manually.

CHAPTER 4

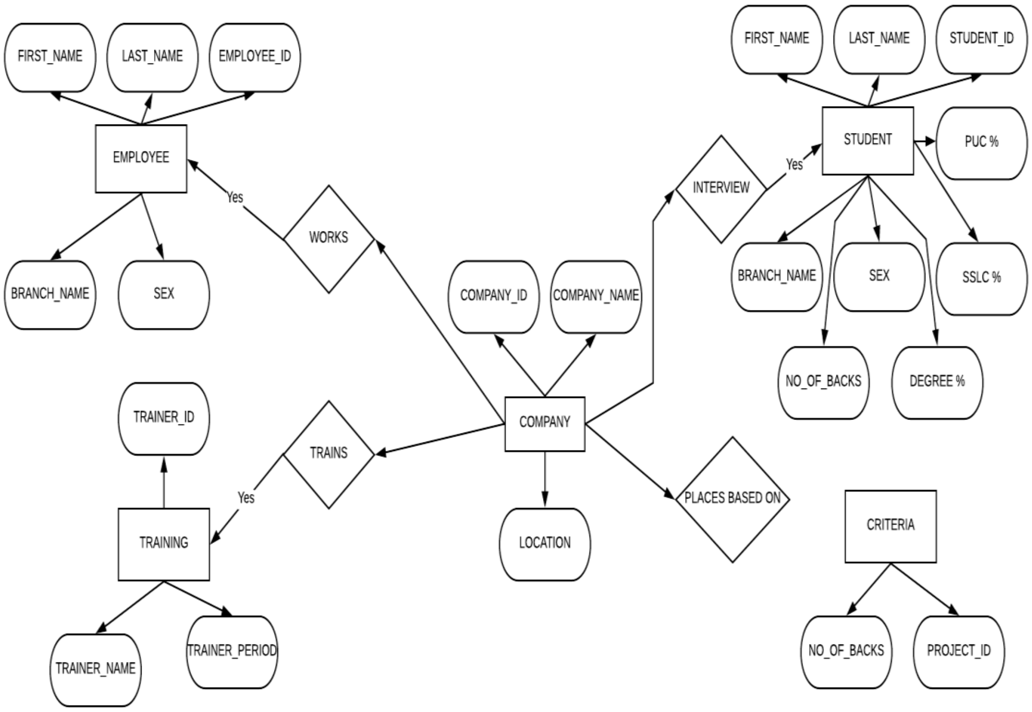
IMPLEMENTATION

4.1 ER DIAGRAM

From this ER diagram the tables obtained are as follows:

Company, Employee, Student, Training, Criteria.

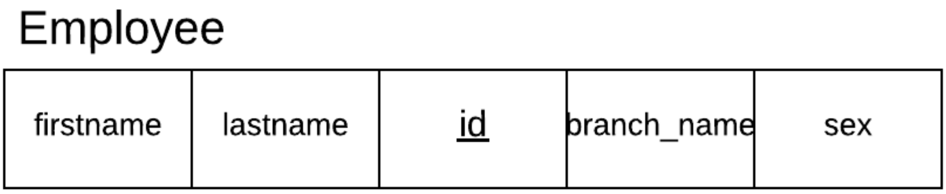
The relation between the entities present in this ER diagram are M:N, 1:N.



4.2 MAPPING OF ER DIAGRAM TO SCHEMA DIAGRAM

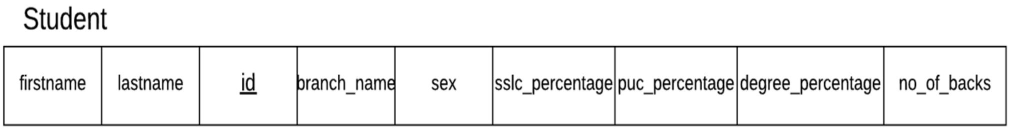
1. Mapping of regular entity types.

- For each regular entity type E in ER schema, create a relation R that includes all the simple attributes of ER.
- Include only the simplest component attributes of a composite attribute.



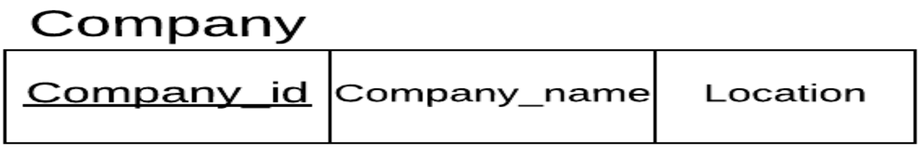
Employee

firstname	lastname	id	branch_name	sex



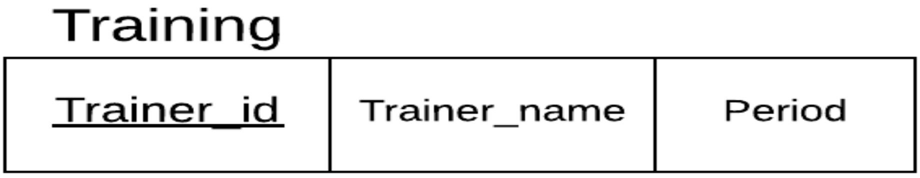
Student

firstname	lastname	id	branch_name	sex	sslc_percentage	puc_percentage	degree_percentage	no_of_backs



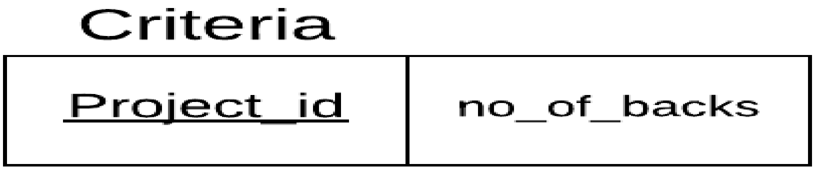
Company

Company_id	Company_name	Location



Training

Trainer_id	Trainer_name	Period

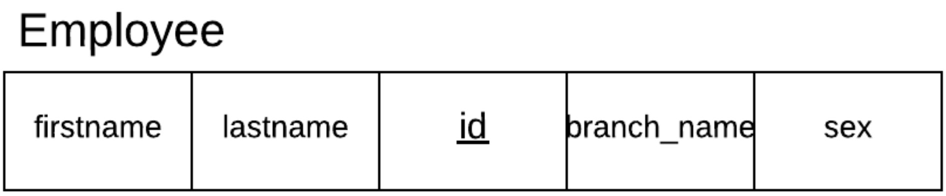


Criteria

Project_id	no_of_backs

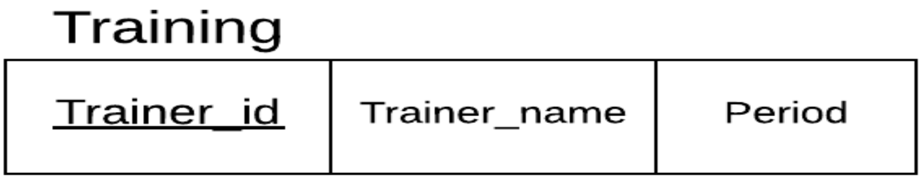
2. Mapping of 1:N relationship type.

- For each regular 1:N relationship type R, identify the relation S that represents the participating entity type at N side of relationship type.
- Include a foreign key in S, the primary key of relation T that represents other entity in R.
- Include any simple attributes of 1:N relationship type as an attribute of S.



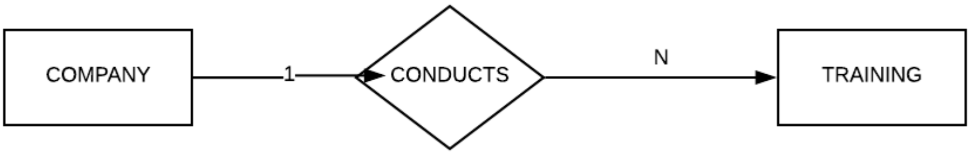
Employee

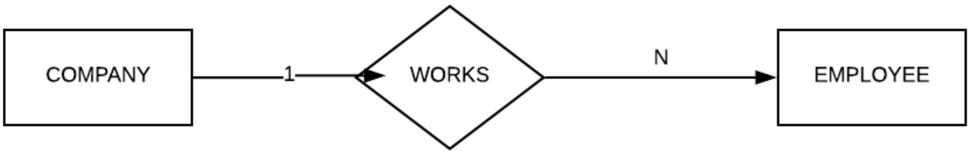
firstname	lastname	id	branch_name	sex



Training

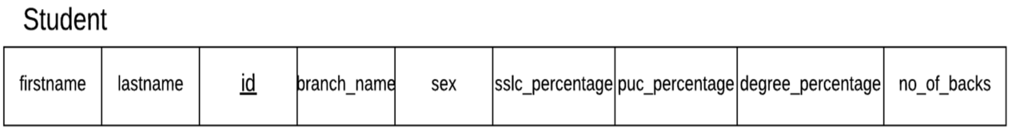
Trainer_id	Trainer_name	Period





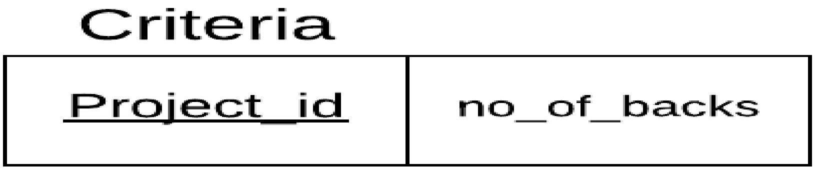
3. Mapping of M:N relationship type.

- For each M:N relation R, create a new relation S to represent R.
- Include a foreign key attribute the primary key of relations that represent the participating entity types.
- Their combination will form primary key of S.
- Also include any simple attributes of M:N relationship as attributes of S.



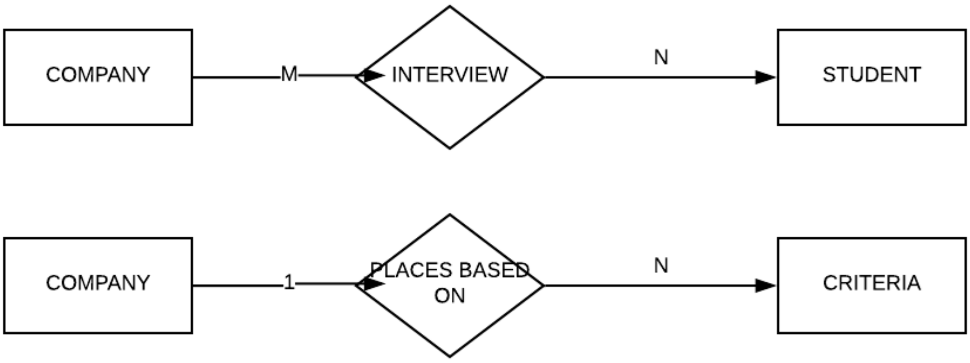
Student

firstname	lastname	id	branch_name	sex	sslc_percentage	puc_percentage	degree_percentage	no_of_backs

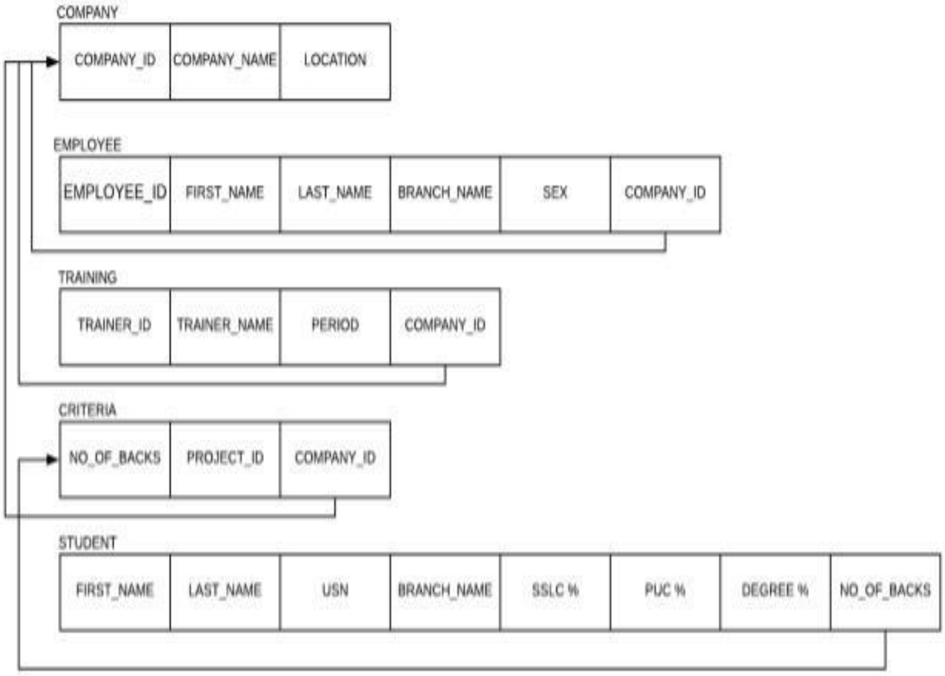


Criteria

Project_id	no_of_backs


COMPANY
M
INTERVIEW
N
STUDENT
COMPANY
1
PLACES BASED ON
N
CRITERIA

4.3 MAPPING OF THE ER SCHEMA TO RELATIONS



4.4 NORMALIZE THE RELATIONS

Normalization is a process of organizing the data in the database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

-- Table is in 1NF (First normal form)

-- No non-prime attribute is dependent on the proper subset of any candidate key of the table.

An attribute that is not part of any candidate key is known as a non-prime attribute.

Third Normal Form (3NF)

- A table is in 3NF if it is in 2NF and if it has a transitive dependency.
- X->Y,Y->Z,X->Z
- According to CODD’s definition, a relational schema R is in 3NF.
- If it satisfies 2NF and no non-primary attributes R is in transitively dependent on the primary key.

All the tables in the schema are satisfied up to 3NF.

4.5 CREATION OF TABLES

create table employee (firstname varchar(32), lastname varchar(32), id int primary key, branch_name varchar(32), sex varchar(7));

create table student (firstname varchar(32), lastname varchar(32), id int primary key, branch_name varchar(32), sex varchar(7),sslc_percentage int, puc_percentage int, degree_percentage int, no_of_backs int);

create table company (company_name varchar(32), company_id integer primary key, location varchar(7));

create table training (trainer_name varchar(32), trainer_id integer primary key, period varchar(7));

create table criteria (no_of_backs int, company_id int foreign key, project_id int primary key);

4.6 INSERTION OF TUPLES

Inserting into the Employee table:

insert into employee values (“Peter”, “Parker”, 123, “Sales”, “Male”);

insert into employee values (“Cindy”, “Pink”, 456, “Sales”, “Female”);

insert into employee values (“Ryan”, “Bateman”, 345, “RnD”, “Male”);

insert into employee values (“Anjali”, “Kapoor”, 234, “RnD”, “Female”);

Inserting into the Student Table:

insert into student values (“Raj”, “Prakash”, 212, “CSE”, “Male”, 60, 62, 64, 1);

insert into student values (“Amber”, “Anderson”, 312, “CSE”, “Female”, 70, 72, 74, 0);

insert into student values (“Mya”, “Brooks”, 321, “CSE”, “Female”, 80, 62, 54, 4);

Inserting into Company Values:

```
insert into company values (“Intel”, 09876, “Bengaluru”);
insert into company values (“GE”, 09678, “Bengaluru”);
insert into company values (“Caterpillar”, 78654, “Chennai”);
```

Inserting into Training Values:

```
insert into training values (“Ruby”, 4657635, “Morning”);
insert into training values (“Vijaya”, 77376987, “Afternoon”);
insert into training values (“Sundar”, 6647763, “Evening”);
```

Inserting into Criteria Values:

```
insert into criteria values (4, 09876, 33);
insert into criteria values (0, 09678, 24);
insert into criteria values (1, 78654, 12);
```

4.7 CREATION OF TRIGGERS

A trigger is a special kind of stored procedure that automatically execute when an event occurs in the database server. DML trigger execute when a user tries to modify data through a Data Manipulation Language (DML) event. DML events are insert, update or delete statement on a table or view.

DDL triggers execute in response to a variety of Data Definition Language(DDL) events. These events primarily correspond to SQL create, alter and drop statement.

```
delimiter $$
create trigger abc
BEFORE INSERT
ON student
FOR EACH ROW
BEGIN
SET NEW.id=CURRENT_TIMESTAMP;
```

4.8 CREATION OF STORED PROCEDURES

A stored procedure is a set of SQL statements with an assigned name ,which are stored in RDBMS as a group, so it can be reused and shared by multiple programs. Users of stored procedures include data validation or access control mechanisms.

Create PROCEDURE ‘xer’() NOT DETERMINISTIC NO SQL SQL SECURITY DEFINER SELECT * FROM student;

CHAPTER 5 FRONT END DESIGN

5.1 CONNECTIVITY TO DATABASE

The MySQL Connector/ODBC is the name of the family of MySQL ODBC driver that provides across to a MySQL database using the industry standard Open Database Connectivity (ODBC) API.

Connector/ODBC provides both driver-manager based and native interfaces to the MySQL database, with full support for MySQL functionality, including stored procedure, transaction.

Typically, we need to install Connector/ODBC on Windows machine. For Unix and OS X, use native MySQL network or named pipes to communicate with your MYSQL database. The application that requires ODBC to communicate with MySQL includes ColdFusion, Microsoft Office, and Filemaker Pro.

```
<?php
$servername="localhost";
$username="root";
$password="";
$dbname="placement";
$conn=new mysqli($servername,$username,$password,$dbname); if($conn-
>connect_error)
{
die("Connection failed: ".$conn->connect_error);
}
?>
```

5.2 FRONT END CODE

Creating Front End Page For The Entity Employee:

```
<html>
<head>
<title>Page Title</title>
```

```
</head>
<style>
body {
    background-image: url("./dbmsproject.jpg");
    background-repeat: no-repeat;
          background-size: 99%;
          background-position: right-bottom;
}
</style>
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
.topnav a.active {
  background-color: #4CAF50;
  color: white;
}
</style>
<body>
<div class="topnav">
  <a class="active" href="main_page.html">Home</a>
  <a href="employee.html">Employee</a>
  <a href="student.html">Student</a>
  <a href="about_comapany.html">About Comapany</a>
</div>
<h1><a href="about_company.html">About Company</a></h1>
</body>
</html>
```

Insertion From Front End HTML Code:

```
<html>
<head>
<title>Employee Details Form</title>
</head>
<style>
body {
    background-image: url("./car.jpg");
    background-repeat: no-repeat;
          background-size: 100%;
```

```
        background-position: right-bottom;
        color: white;
}
</style>
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
.topnav a.active {
  background-color: #4CAF50;
  color: white;
```

```
}
</style>
<body>
<div class="topnav">
  <a class="active" href="main_page.html">Home</a>
  <a href="employee.html">Employee</a>
  <a href="student.html">Student</a>
  <a href="about_comapany.html">About Comapany</a>
</div>
<h2>Employee Details Form</h2>
<form action="insert_employee.php" method="post">
  First name:<br>
  <input type="text" name="firstname">
  <br><br>
  Last name:<br>
  <input type="text" name="lastname">
  <br><br>
  Employee ID:<br>
  <input type="text" name="id">
  <br><br>
  Branch Name:<br>
  <input type="text" name="branch_name">
  <br><br>
  Sex:<br>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other<br><br>
  <input type="submit" value="Submit"><br><br>
```

```
</form>
</body>
</html>
```

Connectivity To Back End To Insert Using PHP:

```
<html>
<body>
<?php
$host="localhost";
$user="root";
$password="";
$con= new mysqli($host,$user,$password,'project');      // here project is
the database name
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
echo "Inserting";
$firstname=$_POST['firstname'];
$lastname=$_POST['lastname'];
$id=$_POST['id'];
$branch_name=$_POST['branch_name'];
$sex=$_POST['gender'];
if($firstname!="" && $id!="")
{
$sql3="insert into employee
values('$firstname','$lastname','$id','$branch_name','$sex')";// student is
the table name
$result = mysqli_query($con,$sql3);
echo "New record created successfully";
}
else{
echo "One of field is Empty!";
}
$con->close();
 }
?>
</body>
</html>
```

<u>Deleting From Front End HTML Code:</u>

```
<!DOCTYPE html>
<html>
<head>
<title>Employee Retrival Details Form</title>
</head>
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
.topnav a.active {
  background-color: #4CAF50;
  color: white;
}
</style>
<body>
<form action="employee_delete.php" method="POST"</form>
<p>Enter the Employee id to delete</p>
<input type="text" name="id">
<input type="submit" value="Submit">
</body>
</html>
```

Connectivity To Back End To Delete Using PHP:

```
<?php
$servername="localhost";
$username="root";
$password="";
$dbname="project";
$conn=new mysqli($servername,$username,$password,$dbname);
```

```
if($conn->connect_error)
{
die("Connection failed: ".$conn->connect_error);
}
else
{
echo "Connected</br>";
}
$id=$_POST['id'];
$sql="delete from employee where id='$id'";
mysqli_query($conn,$sql);
echo "Row deleted";
?>
```

Searching From Front End HTML Code:

```
<!DOCTYPE html>
<html>
<head>
<title>Employee Retrival Details Form</title>
</head>
<style>
body {
  margin: 0;
  font-family: Arial, Helvetica, sans-serif;
}
.topnav {
  overflow: hidden;
  background-color: #333;
}
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
.topnav a.active {
  background-color: #4CAF50;
  color: white;
}
</style>
<body>
<div class="topnav">
  <a class="active" href="main_page.html">Home</a>
  <a href="employee.html">Employee</a>
  <a href="student.html">Student</a>
  <a href="about_company.html">About Comapany</a>
</div>
<h2>Employee Retrival Details Form</h2>
<form action="employee_retrival.php" method="POST">
Employee ID:<br>
```

```
  <input type="text" name="id">
  <br><br>
  <input type="submit" value="Submit"><br><br>
</form>
</body>
</html>
```

Connectivity To Back End To Retrive Using PHP:

```
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>
<?php
$host="localhost";
$user="root";
$password="";
$con= new mysqli($host,$user,$password,"project");
if ($con->connect_error) {
                        die("Connection failed: " . $con->connect_error);
                    }
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
          $n=$_POST['firstname'];
          echo "Entered name is $n <br /><br />";
```

```
        $sql="select * from employee where firstname='$n'";
                                $result = $con->query($sql);
                        if ($result->num_rows > 0) {
                   echo
"<table><tr><th>firstname</th><th>lastname</th><th>id</th><th>branch</th><th>sex</th></tr>";
                                    while($row = $result->fetch_assoc()) {
                      echo "<tr><td>" . $row["firstname"]. "</td><td>" .
$row["lastname"]. "</td><td>" . $row["id"]. "</td><td>" .
$row["branch_name"]. "</td><td>" . $row["sex"]. "</td></tr>";
                   }
                   echo "</table>";
                } else {
                   echo "0 results";
                }
                }

                $con->close();
?>
</body>
</html>
```

CHAPTER 6 TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

TESTING OBJECTIVES

The main objectives of testing process are as follows.

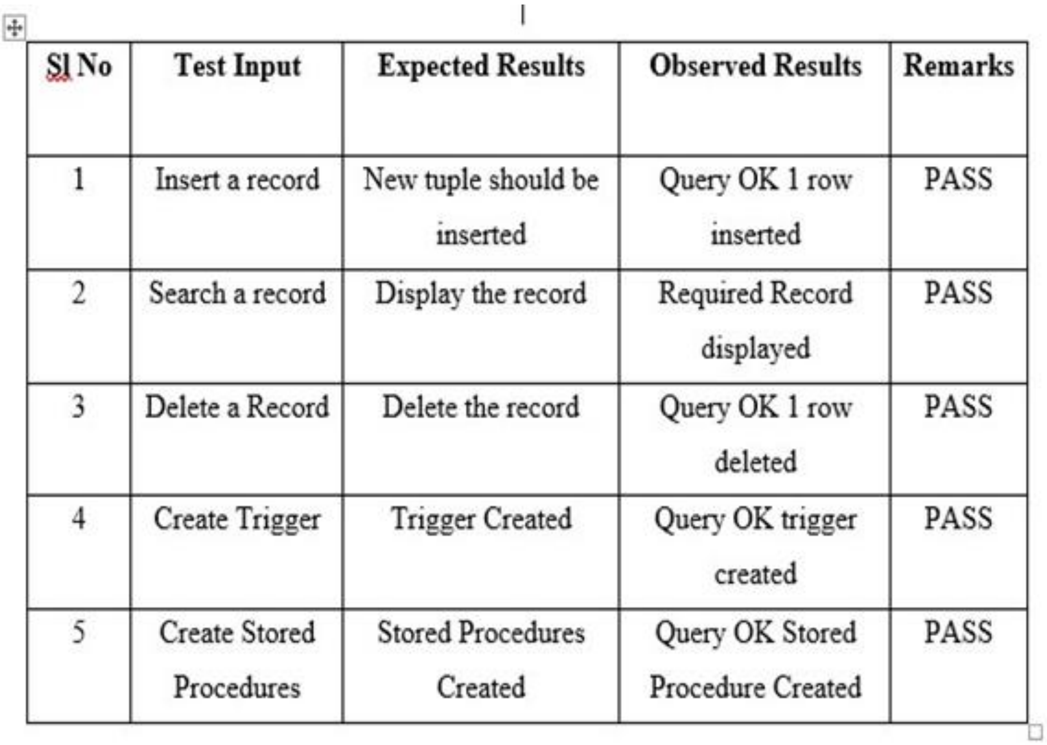
- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

6.3 TEST CASES

The test cases provided here test the most important features of the project.

6.3.1 Test cases for the project

Table 6.1 ------- Test Case



Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Insert a record	New tuple should be inserted	Query OK 1 row inserted	PASS
2	Search a record	Display the record	Required Record displayed	PASS
3	Delete a Record	Delete the record	Query OK 1 row deleted	PASS
4	Create Trigger	Trigger Created	Query OK trigger created	PASS
5	Create Stored Procedures	Stored Procedures Created	Query OK Stored Procedure Created	PASS

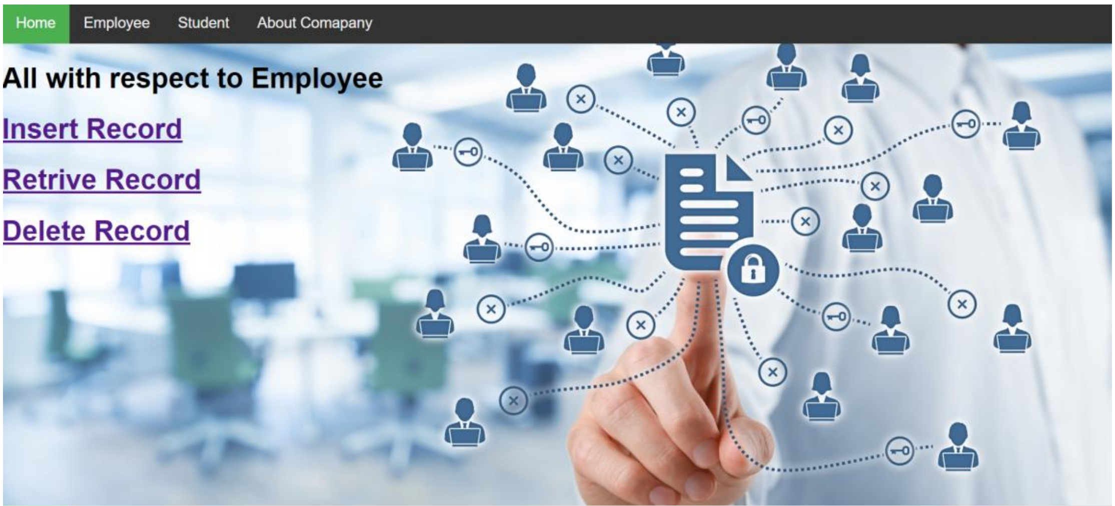
Table 6.1 Test Case

CHAPTER 7

RESULTS

This section describes the screens of the “Placement Management Database”. The snapshots are shown below for each module

7.1 SNAPSHOTS

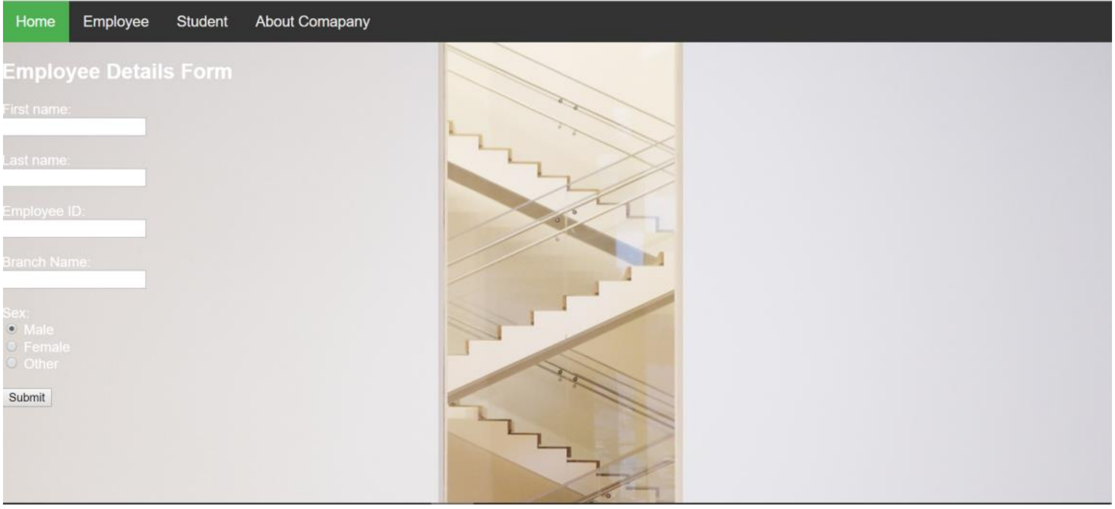


All with respect to Employee

Insert Record

Retrive Record

Delete Record



Employee Details Form

First name:

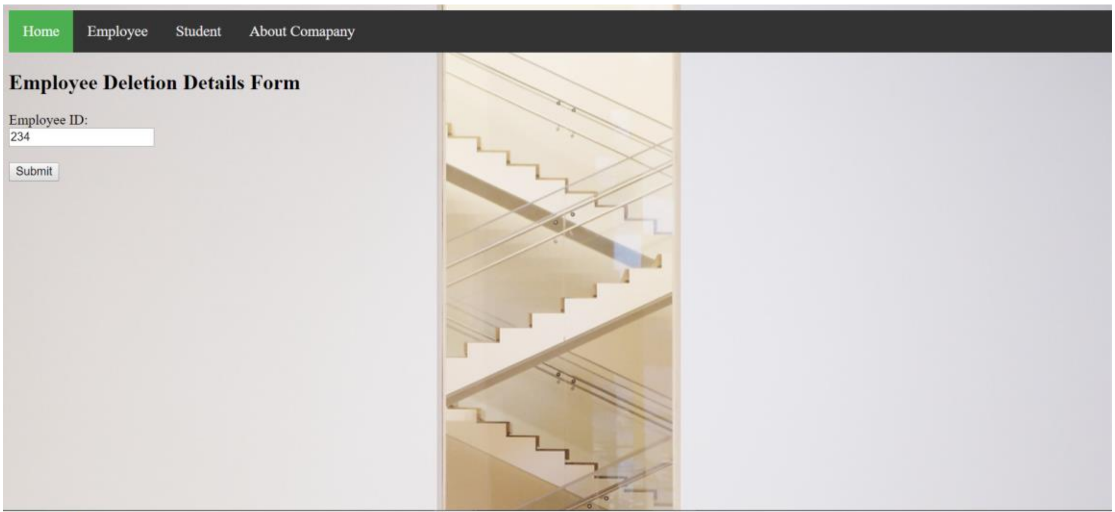
Last name:

Employee ID:

Branch Name:

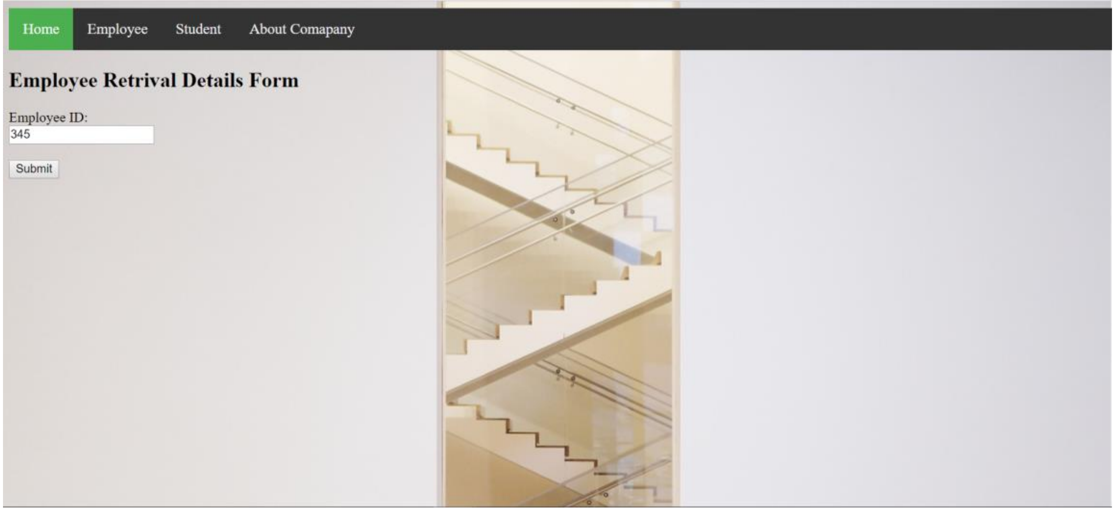
Sex:
● Male
○ Female
○ Other

Submit



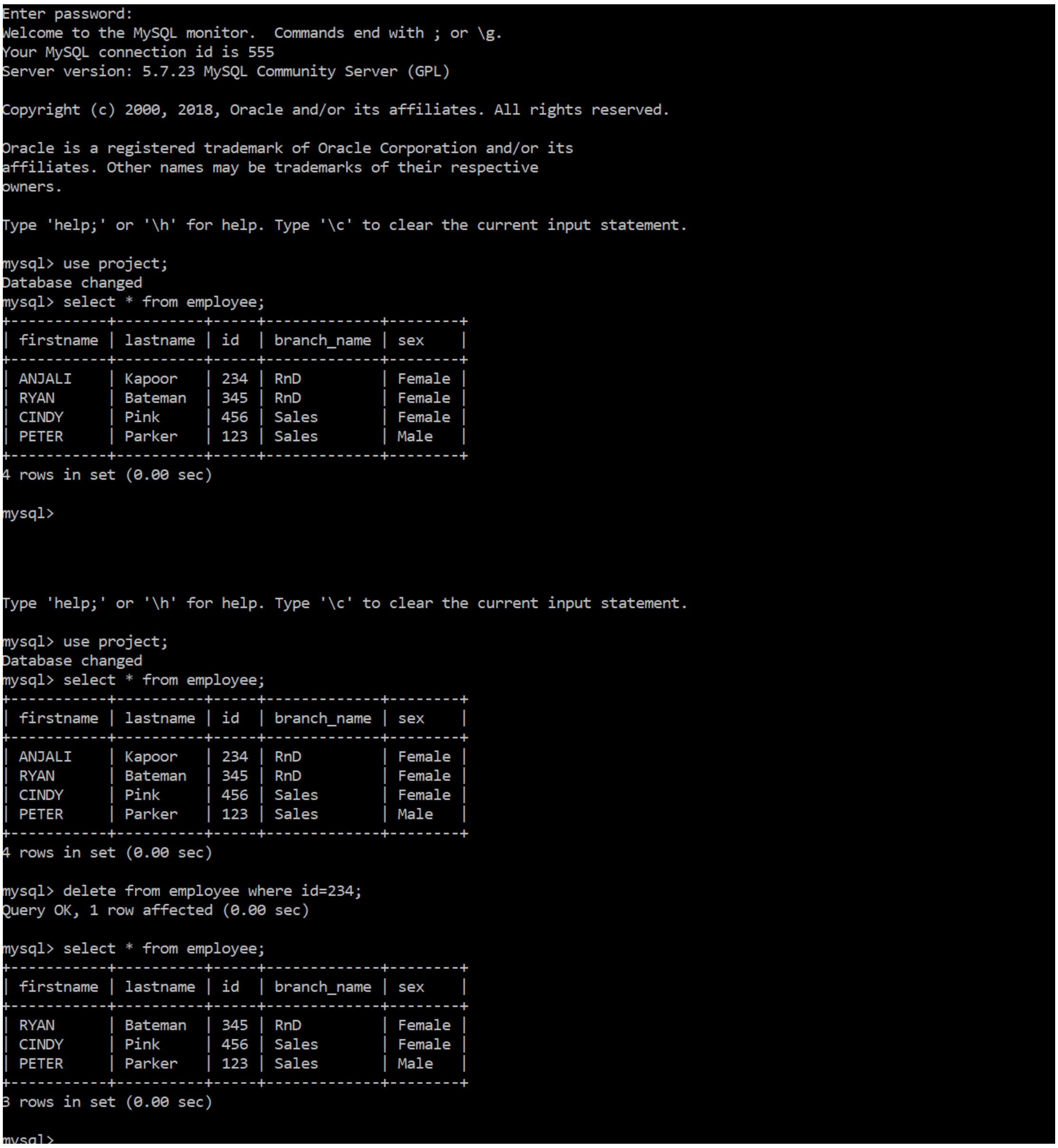
Employee Deletion Details Form

Employee ID: 234

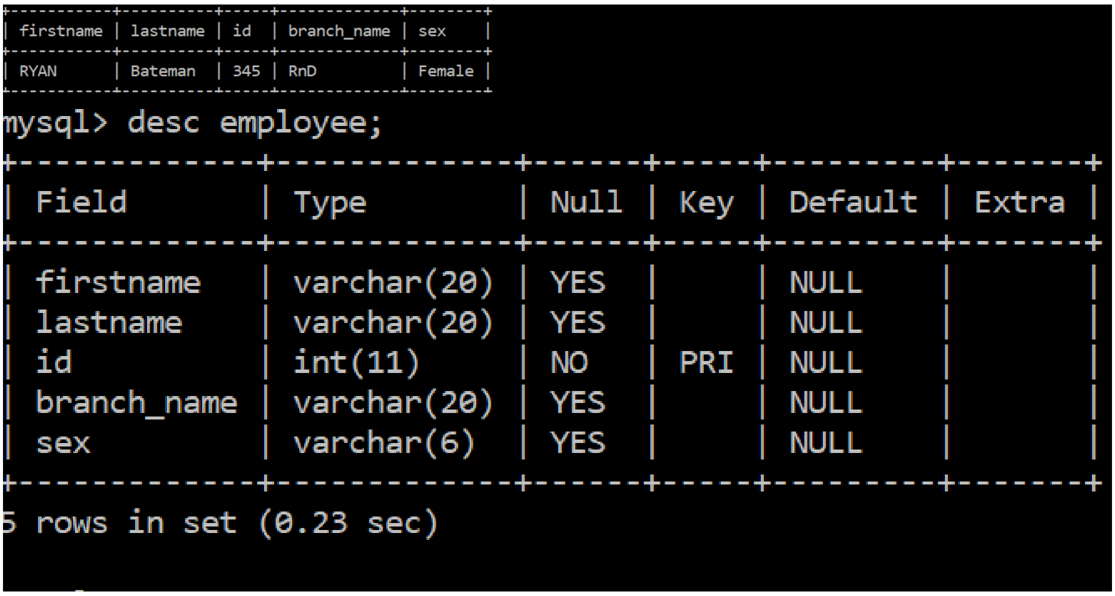


Employee Retrival Details Form

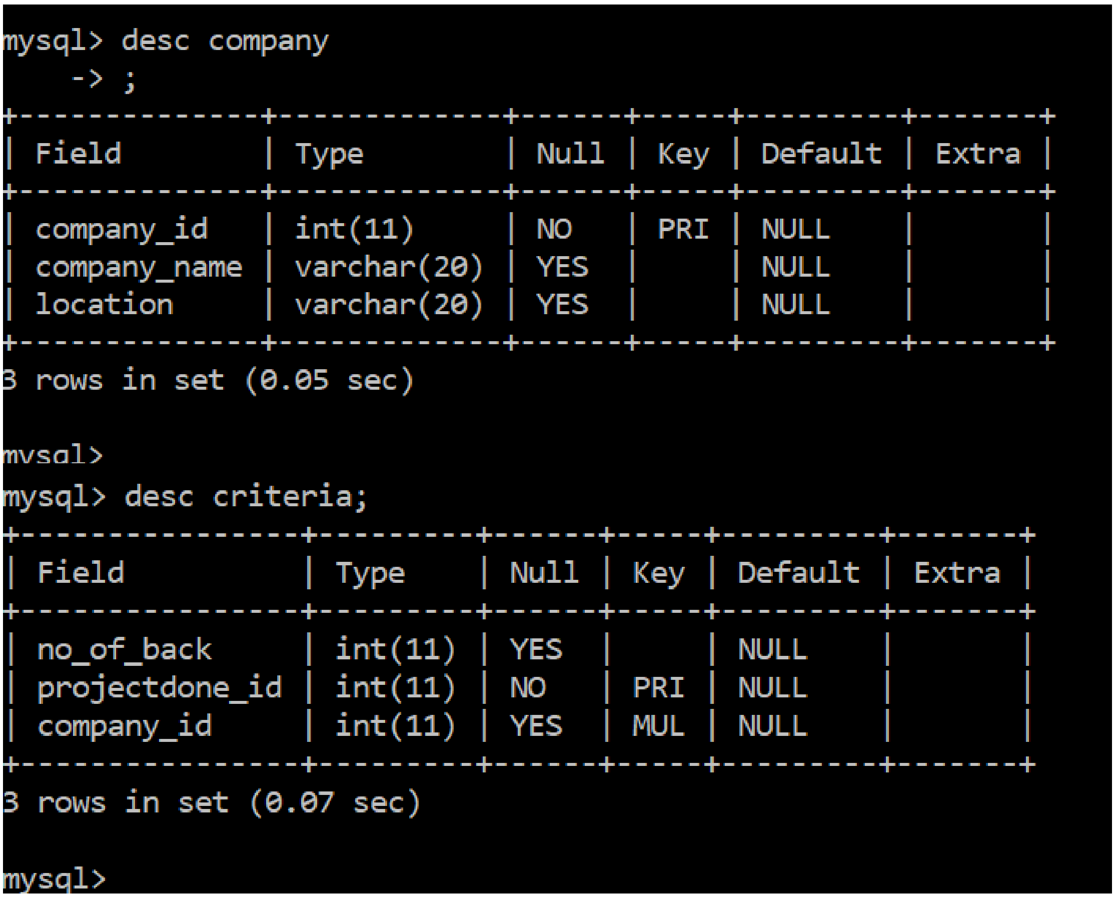
Employee ID: 345



```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 555
Server version: 5.7.23 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> use project;
Database changed
mysql> select * from employee;
+-----------+----------+-----+-------------+--------+
| firstname | lastname | id  | branch_name | sex    |
+-----------+----------+-----+-------------+--------+
| ANJALI    | Kapoor   | 234 | RnD         | Female |
| RYAN      | Bateman  | 345 | RnD         | Female |
| CINDY     | Pink     | 456 | Sales       | Female |
| PETER     | Parker   | 123 | Sales       | Male   |
+-----------+----------+-----+-------------+--------+
4 rows in set (0.00 sec)

mysql>



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

mysql> use project;
Database changed
mysql> select * from employee;
+-----------+----------+-----+-------------+--------+
| firstname | lastname | id  | branch_name | sex    |
+-----------+----------+-----+-------------+--------+
| ANJALI    | Kapoor   | 234 | RnD         | Female |
| RYAN      | Bateman  | 345 | RnD         | Female |
| CINDY     | Pink     | 456 | Sales       | Female |
| PETER     | Parker   | 123 | Sales       | Male   |
+-----------+----------+-----+-------------+--------+
4 rows in set (0.00 sec)

mysql> delete from employee where id=234;
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+-----------+----------+-----+-------------+--------+
| firstname | lastname | id  | branch_name | sex    |
+-----------+----------+-----+-------------+--------+
| RYAN      | Bateman  | 345 | RnD         | Female |
| CINDY     | Pink     | 456 | Sales       | Female |
| PETER     | Parker   | 123 | Sales       | Male   |
+-----------+----------+-----+-------------+--------+
3 rows in set (0.00 sec)

mysql>
```



```
+-----------+----------+-----+-------------+--------+
| firstname | lastname | id  | branch_name | sex    |
+-----------+----------+-----+-------------+--------+
| RYAN      | Bateman  | 345 | RnD         | Female |
+-----------+----------+-----+-------------+--------+
mysql> desc employee;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| firstname   | varchar(20) | YES  |     | NULL    |       |
| lastname    | varchar(20) | YES  |     | NULL    |       |
| id          | int(11)     | NO   | PRI | NULL    |       |
| branch_name | varchar(20) | YES  |     | NULL    |       |
| sex         | varchar(6)  | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.23 sec)
```



```
mysql> desc company
    -> ;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| company_id   | int(11)     | NO   | PRI | NULL    |       |
| company_name | varchar(20) | YES  |     | NULL    |       |
| location     | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.05 sec)

mysql>
mysql> desc criteria;
+----------------+---------+------+-----+---------+-------+
| Field          | Type    | Null | Key | Default | Extra |
+----------------+---------+------+-----+---------+-------+
| no_of_back     | int(11) | YES  |     | NULL    |       |
| projectdone_id | int(11) | NO   | PRI | NULL    |       |
| company_id     | int(11) | YES  | MUL | NULL    |       |
+----------------+---------+------+-----+---------+-------+
3 rows in set (0.07 sec)

mysql>
```

CONCLUSION

The project was successfully designed and is tested for accuracy and quality. Project has accomplished all the objectives. The project can be maintained for life-time purpose as the data can be edited, deleted and added if required. It basically aims at most durable, secure and reliable user experience.

I conclude that this project has helped me gain knowledge about the topic that I was indulged into.

The developed application is tested with simple inputs and outputs obtained in according to the requirements.

REFERENCES

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkanth B. Navathe. 7th edition. 2017.Pearson
2. https://w3schools.com/html.htm
3. http://w3schools.com/css.htm
4. https://w3schools.com/php.htm