**Database Development using MySQL**

**LAB: 05**



**Spring-22**

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr.Naina Said**

June 1, 2022

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawa

**Objectives:**

* To learn about MySQL
* Using MySQL command line in windows
* MySQL command categories
* Use of MySQL commands

**Task 6.2**

**What is difference between SQL and MySQL? Why is MySQL used? What are its features?**

SQL - Structured Query Language. It is declarative computer language aimed at querying relational databases.

MySQL is a relational database - a piece of software optimized for data storage and retrieval. There are many such databases - Oracle, Microsoft SQL Server, SQLite and many others are examples of such.

MySQL is used due to security, storage engine, high availability, cloud development, and access tools. It is also used due to its high level of data updating, data retrieving, and data manipulation.

**MySQL Features:**

Some of MySQL features are ease of management, robust transactional support, comprehensive application development, high performance, the low total cost of ownership, open-source, secure data protection, and high availability.

**Task 6.2**

**What is a database engine? What purpose does it serve? How many types of engines are supported by MySQL? Which database engine is most commonly used and why?**

**Answer:**

the database engine is the software that does the real work of sorting the information, finding specific data that you request, and so on. The term is used to refer to a separate piece of software that ran on a central computer (in this case, it is more or less synonymous with the term “back-end”). Widely used database engines include Oracle, DB2, and Sybase. Separate front-end software running on your own computer lets you tell the database engine what to do (how to sort the data, what data to find), and displays the results of your commands.

There are **two types** of storage engines in MySQL: transactional and non-transactional.

MySQL is one of the most popular and widely used SQL databases. It is also one of the most used databases in Web Applications.

**Task 6.3**

**Consider the Relational Schema given in Figure 6.2 and its tables given in Figure 6.3. Write SQL commands to create all the tables. Take the appropriate attribute type and length from the data provided. (Note: Use the following hierarchy for table creation: 1) Type, Tournament and Team, 2) Member, and 3) Entry).**

**Answer:**

|  |
| --- |
| CREATE TABLE Type(      Type VARCHAR(10) NOT NULL,      Fee INT NOT NULL  ); |

|  |
| --- |
| CREATE TABLE Tournament(      TourID INT PRIMARY KEY,      TourName VARCHAR(12) NOT NULL,      TourType VARCHAR(10) NOT NULL  ); |

|  |
| --- |
| CREATE TABLE Team(  TeamName VARCHAR(10) NOT NULL,  PracticeNight VARCHAR(10) NOT NULL,  Manager INT NOT NULL  ); |

|  |
| --- |
| CREATE TABLE Member(      MemberID INT PRIMARY KEY,      LastName VARCHAR(10) NOT NULL,      FirstName VARCHAR(10) NOT NULL,      Handicap INT,      Gender VARCHAR(1) NOT NULL,      Team VARCHAR(5),      MType VARCHAR(8) NOT NULL,      Coach INT,      Phone INT NOT NULL,      JoinDate DATE NOT NULL  ); |

|  |
| --- |
| CREATE TABLE Entry (      Member INT NOT NULL,      TourID INT NOT NULL,      Year INT NOT NULL  ); |

**Task 6.5**

**Using the insert command, populate all the records in member, type, entry, team, and tournament tables according to Figure 6.3a and Figure 6.3b.**

|  |
| --- |
| INSERT INTO Member(MemberID,LastName,FirstName,Handicap,Gender,Team,MType,Coach,Phone,JoinDate)  VALUES (118,'McKerne','Melixxs',30,'F',NULL,'Junoir',153,963270,'2009-5-10'),      (138,'Stome','Michal',30,'M',NULL,'Senior',NULL,983223,'2013-5-13'),      (153,'Nolan','Brenda',11,'F','TeamB','Senior',NULL,442649,'2010-7-25'),      (176,'Branch','Helan',NULL,'F',NULL,'Social',NULL,589419,'2015-10-18'),      (178,'Beck','Sarah',NULL,'F',NULL,'Social',NULL,226596,'2014-1-6'),      (228,'Burton','Sandra',26,'F',NULL,'Junior',153,244493,'2015-6-21'),      (235,'Cooper','Wiliam',14,'M','TeamB','Senior',153,722954,'2012-2-12'),      (239,'Spence','Thomas',10,'M',NULL,'Senior',NULL,697720,'2010-6-4'),      (258,'Olson','Barbar',16,'F',NULL,'Senior',NULL,370186,'2015-7-12'),      (286,'Pollard','Robert',19,'M','TeamB','Junior',235,617681,'2015-7-26'),      (290,'Boxton','Thomas',26,'M',NULL,'Senior',235,268936,'2012-7-10'),      (323,'Wilcox','Daniel',3,'M','TeamA','Senior',NULL,665993,'2013-4-30'),      (331,'Schmidt','Thomas',25,'M',NULL,'Senior',153,867492,'2013-3-20'),      (332,'Brideges','Deborah',12,'F',NULL,'Senior',235,279087,'2011-3-05'),      (339,'Young','Betty',21,'F','TeamB','Senior',NULL,507813,'2013-3-30'),      (414,'Glimore','Jane',5,'F','TeamA','Junoit',153,458558,'2011-5-12'),      (415,'Taylor','Wiliam',7,'M','TeamA','Senior',235,137353,'2011-10-09'),      (461,'Reed','Robert',3,'M','TeamA','Senior',235,994664,'2009-7-18'),      (469,'Willis','Carolyn',29,'F',NULL,'Junior',NULL,688378,'2014-12-27'),      (487,'Kent','Susan',NULL,'F',NULL,'Social',NULL,707217,'2019-9-19')  ; |

|  |
| --- |
| INSERT INTO Type(Type,Fee) VALUES      ('Associate',60),      ('Junior',150),      ('Senior',300),      ('Social',50)  ; |

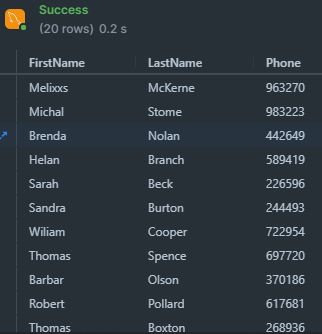
|  |
| --- |
| INSERT INTO Entry(Member,TourID,Year) VALUES   (118,24,2013),  (228,24,2014),  (228,25,2014),  (228,36,2014),  (235,38,2012),  (235,38,2014),  (235,40,2013),  (235,40,2014),  (239,25,2014),  (239,40,2012),  (258,24,2013),  (258,38,2013),  (286,24,2012),  (286,24,2013),  (286,24,2014),  (415,25,2014),  (415,36,2013),  (415,36,2014),  (415,38,2012),  (415,36,2014),  (415,40,2012)  ; |

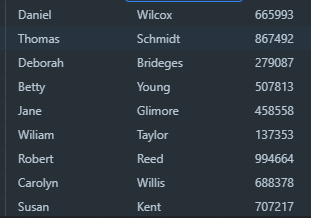
|  |
| --- |
| INSERT INTO Team(TeamName,PracticeNight,Manager) VALUES      ('TeamA','Tuesday',239),      ('TeamB','Monday',153)      ; |

|  |
| --- |
| INSERT INTO Tournament(TourID,TourName,TourType) VALUES      (24,'Leeston','Social'),      (25,'Kaiapoi','Social'),      (36,'WestCoast','Social'),      (38,'Centerburry','Open'),      (40,'Otago','Open')      ; |

1. **List the first name, last name, and phone numbers of all the members.**

SELECT FirstName, LastName,Phone FROM Member;

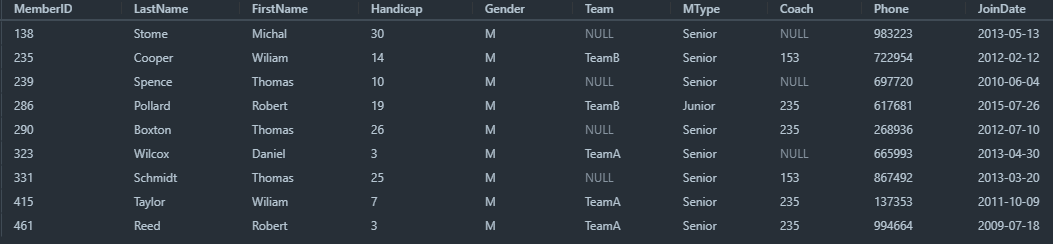
****

****

1. **List complete information of all the male members.**

SELECT \* FROM Member

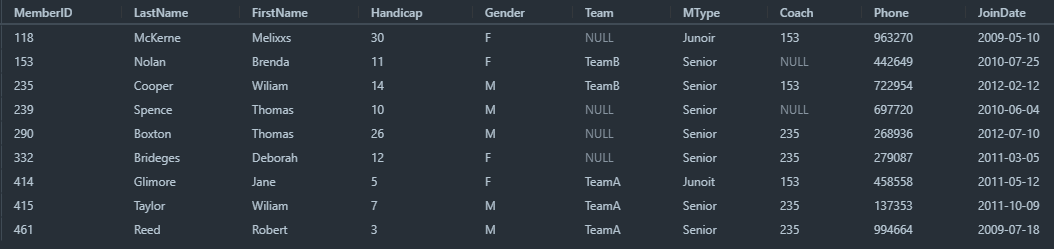
WHERE Gender = 'M';

****

1. **List complete information of all the members who joined after 01‐01‐2013.**

SELECT \* FROM Member

WHERE JoinDate < '2013-01-01';

****

1. **List name of all the members who belonged to Team A.**

SELECT FirstName,LastName FROM Member

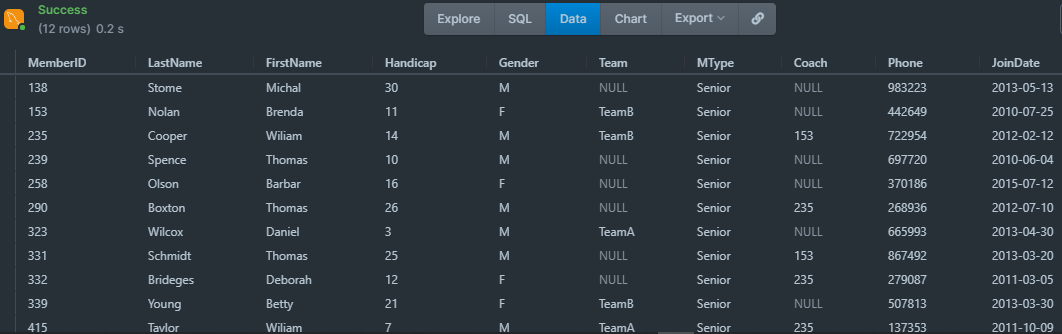
WHERE Team = 'TeamA';

****

1. **List complete information of all the senior members.**

SELECT \* FROM Member

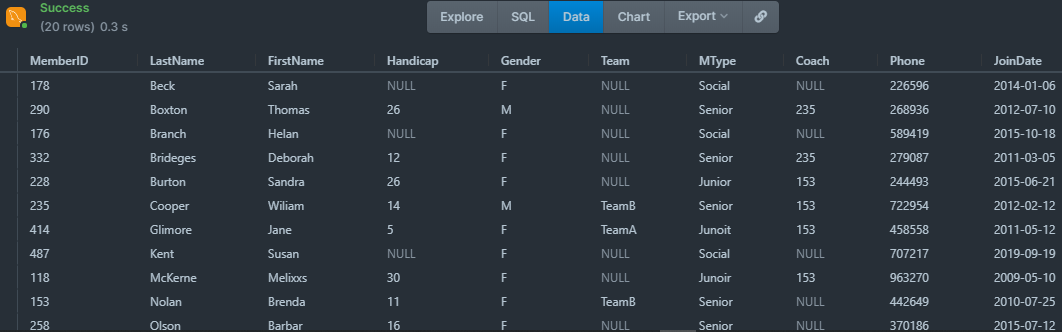
WHERE MType = 'Senior';

****

1. **List complete information of all the members in order of LastName.**

SELECT \* FROM Member

ORDER BY LastName;

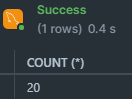
****

****

1. **Retrieve the number of records in Member table.**

SELECT COUNT (\*)

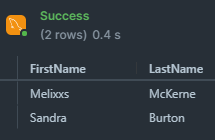
FROM Member;

****

1. **Provide the first name and last name of the two coaches.**

SELECT FirstName,LastName FROM Member

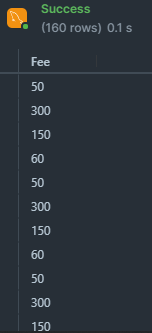
WHERE Coach LIMIT 2;

****

1. **Find the amount of fee provided by each member by mentioning member first name, last name, and fee. (Hint: use the member and type tables.)**

SELECT Fee FROM Member,Type

WHERE FirstName AND LastName OR Fee;

****

1. **Delete the record from Entry table where Member=415 and TourID=40.**

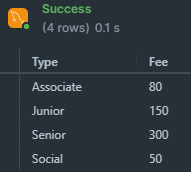
DELETE FROM Entry

WHERE Member = 415 AND TourID = 20;

1. **Update the Fee of Associate in Type table from 60 to 80.**

UPDATE Type SET Fee = 80

    WHERE Type = 'Associate';

****

**Task 6.7**

**MySQL supports various built‐in functions belonging to various categories such as numeric functions, string functions, and date & time functions. Write MySQL commands for following numeric functions: ceiling, cos, degrees, log10, mod, radians, round, sqrt, and truncate. Next write MySQL commands for following string functions: concat, upper, lower, repeat, reverse, regexp, replace, length, ltrim, and rtrim. Finally write MySQL commands for following date & time functions: curdate, week, date\_from, quarter, now, sysdate, and date\_format.**

**Answers:**

**Numeric commands:**

* SELECT CEILING(245.47);
* SELECT COS(5);
* SELECT DEGREES(1.5);
* SELECT LOG10(2);
* SELECT MOD(18, 4);
* SELECT RADIANS(180);
* SELECT ROUND(135.375, 2);
* SELECT SQRT(64);
* SELECT TRUNCATE(135.375, 2);

**String Commands:**

* SELECT CONCAT(Address, " ", PostalCode, " ", City) AS Address  
  FROM Customers;
* SELECT UPPER("SQL Tutorial is FUN!");
* SELECT LOWER(CustomerName) AS LowercaseCustomerName  
  FROM Customers;
* SELECT REPEAT("SQL Tutorial", 3);
* SELECT REVERSE("SQL Tutorial");
* SELECT\*FROM author

WHEREaut\_nameREGEXP'^w';

* SELECT REPLACE("SQL Tutorial", "SQL", "HTML");
* SELECT LENGTH("SQL Tutorial") AS LengthOfString;
* SELECT LTRIM("     SQL Tutorial") AS LeftTrimmedString;
* SELECT RTRIM("SQL Tutorial     ") AS RightTrimmedString;

**Date and Time Commands:**

* SELECT CURDATE();
* SELECT WEEK("2017-06-15");
* DATE\_FROM Date,
* SELECT QUARTER("2017-06-15");
* SELECT NOW();
* SELECT SYSDATE();
* SELECT DATE\_FORMAT("2017-06-15", "%Y");

**Task 6.8**

**MySQL uses various operators such as Comparison (<, >, <=, >=, ==, and !=), Boolean (AND, OR, and NOT), and Special Operators (Between, Like, IN, Is Null, and Distinct). Explore these.**

**Answer;**

**Comparison Operators:**

* SELECT \* FROM Products

WHERE Price > 30;

* SELECT \* FROM Products

WHERE Price < 30;

* SELECT \* FROM Products

WHERE Price >= 30;

* SELECT \* FROM Products

WHERE Price <= 30;

* SELECT \* FROM Products

WHERE Price == 30;

* SELECT \* FROM Products

WHERE Price != 30;

**Boolean Operators:**

* SELECT \* FROM Customers

WHERE City = "London" AND Country = "UK";

* SELECT \* FROM Customers

WHERE City = "London" OR Country = "UK";

* SELECT \* FROM Customers

WHERE City NOT LIKE 's%';

**Special Operators:**

**BETWEEN operator**

The BETWEEN operator returns the information within a given range of values, where the minimum and maximum of the range is specified. For example −

Select\*fromEmployee

WhereEmp\_Salary BETWEEN 20000 AND 60000;

## LIKE operator

The LIKE operator is used to select the values that match the patterns specified in the query. Two wildcard operators are used for this. For example −

Select\*fromEmployee

WhereEmp\_Name LIKE “A%”

## IN operator

The IN operator is true if the query results in values that are contained in the list of constant values for the IN operator. For example −

Select\*fromEmployee

WhereEmp\_ID IN (1,2,5);

### IS NULL

The IS NULL operator is used to filter results with a value of NULL.

SELECT\*

FROM users

WHERE age ISNULL;

## SELECT DISTINCT

The SELECT DISTINCT command returns only distinct (different) values in the result set.

SELECT DISTINCT Country FROM Customers;