## Education and Research Department



# Assignments for Introduction to MySQL

*Sep, 2009*

| Document No. | Authorized By | Ver. Revision | Signature / Date |
|---|---|---|---|
| *ER/CORP/CRS/Introduction to MySQL/004* | Srikantan Moorthy | Ver.1.0 | Sep, 2009 |

# COPYRIGHT NOTICE



Education and Research Department
Infosys Technologies Limited
Electronic City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infosys.com
mailto:E&R@infosys.com

# Document Revision History

| Ver. Revision | Date | Author(s) | Reviewer(s) | Description |
|---|---|---|---|---|
| 1.0 | Sep 2009 | Deepak Bagchi | | Initial Draft |

# Contents

# Assignments for Introduction to MySQL

## Assignment 1: Creating Database and Tables on MySQL

**Objective:** To create database and tables on MySQL.

**Problem Description**: Create a database on MySQL which stores data about booksellers who sold books of various topics in various cities. The tables in the database are:

**Table name: BOOKSELLER**

| Column name | Type | Constraint |
|---|---|---|
| SHOP_NAME | VARCHAR(60) | Primary Key |
| SPECIALISATIONS | VARCHAR(40) | NOT NULL |
| SHOP_PLACE | VARCHAR (25) | NOT NULL |
| NO_OF_SHOPS_IN_CITY | INT | |

**Table name: BOOKSHOP**

| Column name | Type | Constraint |
|---|---|---|
| BOOKSHOP_ADDRESS | INT | NOT NULL |
| NAME_OWNER | VARCHAR(60) | NOT NULL |
| BOOKSHOP_PLACE | VARCHAR | (10) |

**Table name: BOOK**

| Column name | Type | Constraint |
|---|---|---|
| B_ID | INT | Primary, AUTO_INCREMENT |
| TITLE | VARCHAR(60) | NOT NULL |
| AUTHOR | VARCHAR(40) | Foreign key(Author table) |
| SOLD_AT | VARCHAR(10) | Foreign key(Bookshop table) |
| PUBLISHER | VARCHAR | (10) |
| PUBLISHER_PLACE | VARCHAR(10) | |
| YEAR_PUBLICATION | VARCHAR | (10) |

**Table name: AUTHOR**

| Column name | Type | Constraint |
|---|---|---|
| NAME | VARCHAR (40) | Primary key |
| YEAR_OF_BIRTH | INT (4) | NOT NULL |
| PLACE_OF_BIRTH | VARCHAR (20) | |
| AGE | INT(3) | |

## Assignment 2: Creating Tables and Writing Queries

**Objective:** To create database and tables on MySQL.

**Problem Description**: Create a employee database on MySQL which stores data about employees and customers of an organization.

**Table name: Employee_Details**

| Column Name | Type | Constraint |
|---|---|---|
| Emp_ID | Integer(4) | Primary Key |
| First_Name | Varchar(20) | |
| Last_Name | Varchar(20) | Not Null |
| User_ID | Varchar(30) | Not Null |
| Salary | Float(9,2) | |

**Data for the table Employee:**

| Emp_ID | First_Name | Last_Name | User_ID | Salary |
|---|---|---|---|---|
| 1 | Deepak | Bagchi | Dbagchi | 1000 |
| 2 | Sanjeev | Dutt | Sdutt | 3000 |
| 3 | Hema | Singh | Hsingh | 3400 |
| 4 | Chetan | Danivas | Cdanivas | 4600 |
| 5 | Sundar | Rajan | Srajan | 2300 |

Write the following SQL statements:

1. Add first record mentioning the column list in the insert clause.
2. Add next two records without mentioning the column list in the insert clause.
3. Add 4th record and enter only EMP_ID and First_Name
4. Add 5th record and enter EMP_ID, User_ID and Last_Name only.
5. For record with EMP_ID = 4 update record with Last_Name, User_ID and Salary.
6. For record with EMP_ID = 5 update records with First_Name and Salary.
7. Commit the changes.
8. Modify the Salary and increase it by 1000, for all who get salary less than 3000.
9. Delete the employee record having Last_Name as Singh.
10. Commit the changes.
11. Remove the entire contents of the table.
12. Add a PRIMARY KEY constraint to the table Employee using the Emp_ID column.
13. Add a NOT NULL constraint to the table Emploee on column First_Name.
14. Disable NOT NULL Constraint on the column First_Name from the table Employee.
15. Create a view Employee_V with columns EMP_ID, First_Name and Dept_ID from table Employee.

16. Create a view Employee_V2 which should include column Emp_ID, EName and DeptNo from the table Employee and change the column headings as EmpNumber, Employee, Department.
17. Select VIEW_NAME and TEXT from the data dictionary.

## Assignment 3: Creating Tables and Writing Queries

**Objective:** To design and create database. Also write queries on it.

**Problem Description**: Create a database that contains information about the Badminton Association. Following are some assumptions:

- There could be 3 types of game Singles, Doubles and Mixed Doubles.
- Each player has a id number called as PlayerID, Eacg tournament has a id number called tour_id and each match also has match_id. A match_id uniquely represents it across all matches across all tournaments.
- To participate in a tournament, a player has to first register. A new registration number will be given to all players that would contain the tour_id, so that through the registration number we can know which tournament it belongs to. A player_id is only used in database for registration in tournaments and at all other places we use the registration number.
- Two players participating in doubles share the same registration number. But same player in different forms gets different registration number. Depending on the registration number we can make a team of 1 or 2.
- A column called seed is given to top players. Some players may not have any seed.
- We need to store number of rounds in a tournament as an integer. The match also has a round number as integer.
- Also need to store details of each match having score line, winner and loser name.
- Store the information of the matches that have been forfeited i.e. one of the player has not taken part due to injury or any other reasons.
- Many of the queries will require some date manipulation. Specifically, you may have to extract the year a particular tournament was held. Only concern yourself with the year of the start date.

Write SQL statements for the following:

1. List the names of all the players in the database.

2. List the names of all players who have ever been assigned a seed for any tournament (doubles or singles).
3. List the pairs of players who played doubles at Thomas Cup in 2008. Do not include duplicate entries.
4. List the names of all players who have lost to a player (you can choose the name from the table) in singles match in the finals of any tournament along with the name of the tournament.
5. For all final round single matches, list the winner and loser of matches that were between two seeded players, as well as their seeds.
6. List the names of all players who have participated in at least two tournaments in a given year.
7. List all tournaments having more than 5 rounds.
8. List all doubles matches that were won because one of the players retired.
9. For all tournaments in the database in ascending order by the number of rounds.
10. List the names, tournament types, and lengths (in days) of all tournaments that were longer than one week.

## Assignment 4: Understanding Subqueries

**Assignment 4a:**

**Objective:** To understand subqueries.

**Problem Description**: There are 3 tables as – ServicePerson, Customer and Assignment. The assignment table assigns service persons having unique id with each customer request. The table looks like:

| Assignment_Id | Service_Person_Id | Customer_Id | Date | Time |
|---|---|---|---|---|
| 1 | SP1 | C12 | 2009-01-03 | 11:12:23 |
| 2 | SP4 | C13 | 2009-02-03 | 12:12:23 |
| 3 | SP2 | C15 | 2009-10-03 | 10:12:23 |
| 4 | SP1 | C16 | 2009-10-05 | 09:12:23 |

Write an SQL query to list which customer is currently assigned to each service person.

**Assignment 4b:**

**Objective:** To understand subqueries.

**Problem Description**: There are following tables:

**Table name: Sales_Information**

| City Name | Sales | Date |
|---|---|---|
| Calcutta | 10000 | 05-Dec-2008 |
| Bangalore | 20000 | 02-Feb-2009 |
| Mumbai | 15000 | 06-Mar-2009 |
| Delhi | 5000 | 10-Jul-2009 |
| Bhubaneswar | 7000 | 07-Jul-2009 |

**Table name: Region_Details**

| Region_name | City_name |
|---|---|
| East | Calcutta |
| East | Bhubaneswar |
| North | Delhi |
| South | Bangalore |
| South | Chennai |
| West | Mumbai |
| West | Ahmedabad |

Write an SQL statement to find the sales of all stores in the East region during 2008.

## Assignment 5: Understanding Triggers

**Assignment 5a:**

**Objective:** To write a trigger.

**Problem Description**: Refer to assignment 3. Write a trigger for ensuring the integrity constraints upon insertions to the players table:

- A player may participate in at most one tournament at any given point of time.
- A player cannot have more than 3 participations with different registration id's including single, double and mixed doubles.

**Assignment 5b:**

**Objective:** To write a trigger.

**Problem Description**: Refer to assignment 2. Create the following table:

**Table name: Employee_log**

| Name | Type |
|---|---|

| | |
|---|---|
| Employee_id | Varchar(10) |
| First_name | Varchar(20) |
| mod_user | Varchar(10) |
| mod_timestamp | Timestamp |

Write a trigger EMPLOYEE table that will be executed after every insert or update and does the following:

- For insert, put the new EMPLOYEE.Emp_ID value, the user in mod_user, and the system date in mod_timestamp to record the creation of the data into EMPLOYEE_LOG.

- For updates, put the previous EMPLOYEE.Emp_ID value into the corresponding EMPLOYEE_LOG column as well as recording the mod_user and mod_timestamp.

## Assignment 6: Understanding Functions and Procedures

**Objective:** To write a functions and procedures.

**Problem Description**: Create a Telephone_users table as follows:

**Table name**: Telephone_users

| Column Name | Data Type | Constraint |
|---|---|---|
| Phone_no | Varchar(10) | Primary key |
| first_name | Varchar(30) | Unique |
| last_name | Varchar(30) | |
| Extension | Integer(4) | |

**Assignment 6a:**

Create a function name Extension_calc which takes a string as input parameter. The value will be the person's first name. The output will be the value for extension column that is a string calculated as:

- Use only first four characters of first_name column.
- For each character use the following table:

| Character | Integer |
|---|---|
| ABC | 1 |
| DEF | 2 |
| GHI | 3 |
| JKL | 4 |
| MNO | 5 |
| PQR | 6 |

| | |
|---|---|
| STU | 7 |
| VWX | 8 |
| YZ | 9 |

- For names shorter than 4 characters, use zeros in those character positions.

**Assignment 6b:**

Write a procedure that will iterate n times where n is the parameter you pass to your procedure. For each iteration of the loop, copy the corresponding rows of data from employee table into your PHONE_USERS table.

As you insert this data you should perform the following transformations:

- Each Phone_no should be unique. Make it is unique any way you want as long as it looks like a phone number.

- first_name should be first_name with 4 digits appended to the end. These digits will be 0001 through 000n where n is the iteration.

- Calculate and store the extension using the function created above.

- All other columns will be copied exactly as they appear in the DROBERTS.BBT_USERS_TEMP table.

- Before inserting data, truncate this table.