

# DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY, JAMSHORO

# Database Management Systems (4<sup>th</sup> Semester) 18CS Lab Experiment 3

Roll No:	Date	Date of Conduct:				
<b>Submission Date:</b>	Grade Obtained:					
Problem Recognition (0.3)	Completeness & accuracy (0.4)	Timeliness (0.3)	Score (1.0)			
Objective: To be familiar wi	ith Table creation and p	oopulation of table.				

## **Creating Databases and Tables**

In order to be able to add and manipulate data, you first have to create a database. There's not much to this. You're creating just a container in which you will add tables. Creating a table is more involved and offers many choices.

There are a few basic things to decide when creating a structure for your data:

- The number of tables to include in your database, as well as the table names
- For each table, the number of columns it should contain, as well as the column names
- For each column, what kind of data is to be stored.

## **Creating a Database:**

Creating a database is simple, mostly because there's nothing much to it. Use the SQL statement CREATE DATABASE. You will have to provide a name for the database with this SQL statement. You could call it something bland like db1.

CREATE DATABASE rookery;
--------------------------

# **Creating Tables:**

The next step for structuring a database is to create tables. Although this can be complicated, we'll keep it simple to start. We'll initially create one main table and two smaller tables for reference information. The main table will have a bunch of columns, but the reference tables will have only a few columns.

```
CREATE TABLE birds (
Bird_id INT AUTO_INCREMENT PRIMARY KEY,
Scientific_name VARCHAR (255) UNIQUE,
Common_name VARCHAR (50),
Family_id INT,
Description TEXT);
```

This SQL statement creates the table birds with five fields, or columns, with commas separating the information about each column.

The names of the columns can be anything other than words that are reserved for SQL statements, clauses, and functions. Actually, you can use a reserve word, but it must always be given within quotes to distinguish it.

## Adding data to your table

Let's inject some data into that table. We'll add information for a fictional member of the editorial staff. The data to be added is:

• Name: Olivia

• ID: 01

• Email: olivia@company.com

The command to add this would be:

INSERT INTO editorial (id, name, email) VALUES (01,"Olivia","olivia@company.com"); You can view the information added to the table with the command:

#### Lab Task

1. Create the DEPT table with field names DEPTNO, DNAME and LOCATION. Take datatypes accordingly.

#### **Solution:** dept I SELECT \* FROM sqltask.dept; Result Grid 43 Filter Rows: deptno deptname loc CS MUET 1 2 SW MUET 3 ES MUET 4 TI. MUET EL MUET 5 NULL Column Name Default/Expression Datatype NN UQ В UN ZF AI $\checkmark$ deptno V ename VARCHAR(45) V job VARCHAR(45) $\Box$ o mgr VARCHAR(45) NULL V hiredate DATE **V** sal o comm VARCHAR(45) NULL

2. Create EMP table with columns EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO. Make EMPNO as primary key and DEPTNO as foreign key.

#### **Solution:** Column Name Datatype NN UN AI Default/Expression V deptno $\vee$ VARCHAR(45) ename job VARCHAR(45) o mgr VARCHAR(45) NULL $\vee$ hiredate DATE ~ sal INT o comm VARCHAR(45) NULL | Edit: 🕍 🖶 | Export/Import: 📳 0 Result Grid Filter Rows: Wrap Cell Content: IA empno deptno ename job ▲ mgr hiredate sal comm NULL HULL Clerk 2020-02-02 10000 1 Raza 1 NULL NULL 2 Ali 7000 Asst:Clerk 2020-05-03 1 NULL NULL 3 Basit Lab:Asst 2020-09-13 13000 NULL NULL

3. Create the STUDENT table based on the following table instance chart. Choose the appropriate data types and be sure to add integrity constraints.

Column_ Name	STUDENT_	NAME	FIRST_NAM E	ADDRESS	DEPTNO	PHONE	DATE
Key Type	PK				FK		
Null/ Unique	NN,U	NN			NN		NN
FK Ref Table					DEPT		
FK Ref Column					DEPTNO		
Default Value							System Date
Data Type	NUMBER	VARCHAR2	VARCHAR2	VARCHAR2	NUMBER	VARCHAR2	DATE
Length	10	15	15	25	2	15	

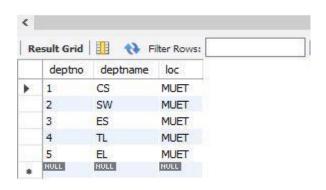
#### **Solution:**

```
1 9 1 Q 0 1 80 1 0
                                       Limit to 1000 rows
                                                      - | 🏂 | 🥩 Q 🗐 🔁
       Create Table STUDENT (
        stduent_id int(10) PRIMARY KEY NOT NULL UNIQUE,
2
        last_name varchar (15) NOT Null,
3
4
        first_name varchar (15),
5
        address varchar(25),
6
        deptno int (2) NOT NULL,
        phone varchar (15),
7
        adm_date datetime Not Null default NOW(),
8
        foreign key (deptno) references dept(deptno)
9
10
11
       );
```

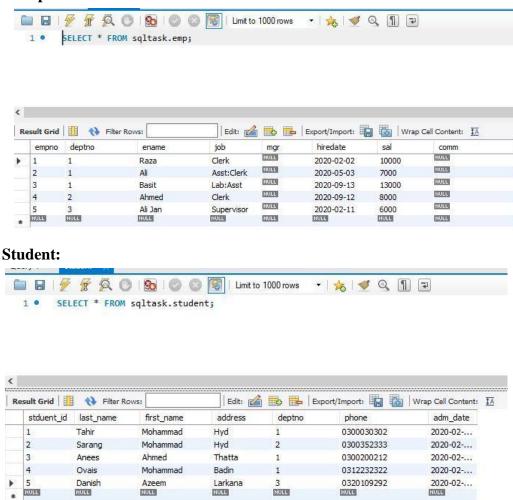
4. Add at least 5 rows in each table by using constant values of your own choice.

## **Solution**:





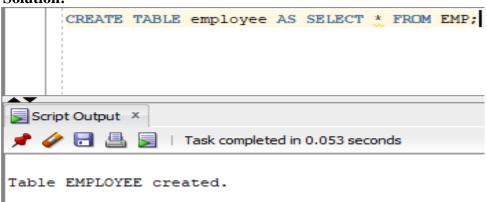
## **Emp Table:**



5. Create the employee table based on the structure of EMP table.

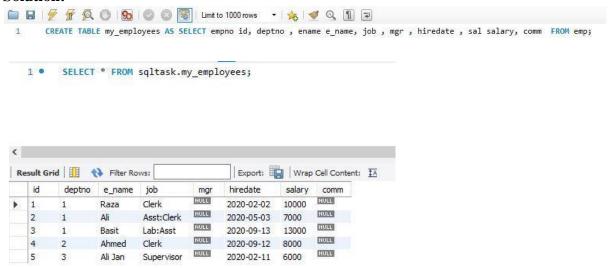
#### **Solution:**

student 1 ×



6. Create a table MY\_EMPLOYEE based on the structure of EMP table. Include the columns Empno, Ename, and Sal. Name them in new table as Id, Name, and Salary.

#### **Solution:**



7. Rename employee1 table to Employee.

#### **Solution:**



8. Drop employee table.

## **Solution:**



9. Add a comment to the student table describing the table.

#### **Solution:**

10. Add a comment on ADM DATE column as 'Admission Date of student'.

#### **Solution:**

1 • ALTER TABLE `student` CHANGE `ADM\_DATE` `ADM\_DATE` DATETIME NOT NULL DEFAULT NOW() COMMENT 'Admission date of student';
2 • SHOW FULL COLUMNS FROM student;

