



DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
MEHRAN UNIVERSITY OF ENGINEERING & TECHNOLOGY,
JAMSHORO
Database Management Systems (4th Semester) 18CS
Lab Experiment 3

Roll No:

Date of Conduct:

Submission Date:

Grade Obtained:

Problem Recognition (0.3)	Completeness & accuracy (0.4)	Timeliness (0.3)	Score (1.0)

Objective: To be familiar with Table creation and population of table.

Tools: MySql / Oracle.

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:



Result Grid   Filter Rows:

	deptno	deptname	loc
▶	1	CS	MUET
	2	SW	MUET
	3	ES	MUET
	4	TL	MUET
	5	EL	MUET
★	NULL	NULL	NULL

[illegible]

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:

[illegible][illegible]

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_ID	LAST_NAME	FIRST_NAME	ADDRESS	DEPTNO	PHONE	ADM_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  Create Table STUDENT (
2      stduent_id int(10) PRIMARY KEY NOT NULL UNIQUE,
3      last_name varchar (15) NOT Null,
4      first_name varchar (15),
5      address varchar(25),
6      deptno int (2) NOT NULL,
7      phone varchar (15),
8      adm_date datetime Not Null default NOW(),
9      foreign key (deptno) references dept(deptno)
10
11 );

```

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

Solution:

Department

```

1  SELECT * FROM sqltask.dept;

```

Result Grid			
Filter Rows:			
	deptno	deptname	loc
▶	1	CS	MUET
	2	SW	MUET
	3	ES	MUET
	4	TL	MUET
	5	EL	MUET
*	NULL	NULL	NULL

Emp Table:

```
1 • SELECT * FROM sqltask.emp;
```

empno	deptno	ename	job	mgr	hiredate	sal	comm
1	1	Raza	Clerk	NULL	2020-02-02	10000	NULL
2	1	Ali	Asst:Clerk	NULL	2020-05-03	7000	NULL
3	1	Basit	Lab:Asst	NULL	2020-09-13	13000	NULL
4	2	Ahmed	Clerk	NULL	2020-09-12	8000	NULL
5	3	Ali Jan	Supervisor	NULL	2020-02-11	6000	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Student:

```
1 • SELECT * FROM sqltask.student;
```

stduent_id	last_name	first_name	address	deptno	phone	adm_date
1	Tahir	Mohammad	Hyd	1	0300030302	2020-02-...
2	Sarang	Mohammad	Hyd	2	0300352333	2020-02-...
3	Anees	Ahmed	Thatta	1	0300200212	2020-02-...
4	Ovais	Mohammad	Badin	1	0312232322	2020-02-...
5	Danish	Azeem	Larkana	3	0320109292	2020-02-...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

student 1 x

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

Solution:

```
CREATE TABLE employee AS SELECT * FROM EMP;
```

Script Output x

Task completed in 0.053 seconds

Table EMPLOYEE created.

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:

```
1 CREATE TABLE my_employees AS SELECT empno id, deptno , ename e_name, job , mgr , hiredate , sal salary, comm FROM emp;
```

```
1 • SELECT * FROM sqltask.my_employees;
```

	id	deptno	e_name	job	mgr	hiredate	salary	comm
▶	1	1	Raza	Clerk	NULL	2020-02-02	10000	NULL
	2	1	Ali	Asst:Clerk	NULL	2020-05-03	7000	NULL
	3	1	Basit	Lab:Asst	NULL	2020-09-13	13000	NULL
	4	2	Ahmed	Clerk	NULL	2020-09-12	8000	NULL
	5	3	Ali Jan	Supervisor	NULL	2020-02-11	6000	NULL

7. Rename employee1 table to Employee.

Solution:

```
1 Alter Table employee1 rename to employee ;
```

8. Drop employee table.

Solution:

```
emp employee2 x
```

```
1 Drop Table employee ;
```

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

Solution:

```
1 • ALTER TABLE student COMMENT 'Details of students';
2 • SELECT table_comment
3 FROM INFORMATION_SCHEMA.TABLES
4 WHERE TABLE_NAME = 'student';
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
TABLE_COMMENT			
Details of students			

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;
```

Field	Type	Collation	Null	Key	Default	Extra	Privileges	Comment
STUDENT_ID	int	utf8mb4_0900_ai_ci	NO	PRI	NULL		select,insert,update,references	
LAST_NAME	varchar(15)	utf8mb4_0900_ai_ci	NO		NULL		select,insert,update,references	
FIRST_NAME	varchar(15)	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
ADDRESS	varchar(25)	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
DEPTNO	int	utf8mb4_0900_ai_ci	NO	MUL	NULL		select,insert,update,references	
PHONE	varchar(15)	utf8mb4_0900_ai_ci	YES		NULL		select,insert,update,references	
ADM_DATE	datetime	utf8mb4_0900_ai_ci	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED	select,insert,update,references	Admission date of student