

# ***SQL MODULE***

## ***LAB – 2***

***BY***

***S MOHAMMED***

***SHARFUDDIN***

***AF0366347***

# Questions

Use the Database and table from Day 1 lab. Insert 5 records in each table

and retrieve data from all tables and display.

Table1: Student

```
Database changed
mysql> show tables;
+-----+
| Tables_in_studentmanagementsystem |
+-----+
| course
| enrollment
| feedback
| instructor
| score
| student
+-----+
6 rows in set (0.00 sec)

mysql> desc student;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| StudentID  | int       | NO   | PRI | NULL    |       |
| FirstName  | varchar(50) | NO   |     | NULL    |       |
| LastName   | varchar(50) | NO   |     | NULL    |       |
| DateOfBirth | date      | NO   |     | NULL    |       |
| Gender     | varchar(6) | NO   |     | NULL    |       |
| Email      | varchar(50) | NO   | UNI | NULL    |       |
| phone      | bigint    | NO   | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> insert into student values
-> (1,'Mohammed Sharfuddin','Shaik','2002-02-02','sharfoddin28@gmail.com',6303729550),
-> (2,'Gautham','Lakhude','2003-03-26','Male','lakhundegautham@gmail.com',9381631295),
-> (3,'Susmitha','Pothireddy','2002-10-26','Female','susmil123@gmail.com','9472549645'),
-> (4,'Sirisha','Yarrampally','2001-05-12','Female','Siril123@gmail.com',9276539234),
-> (5,'Kusuma','dulam','2002-07-18','Female','kusuma123@gmail.com',9652900626);
```

```
mysql> select * from student;
```

StudentID	FirstName	LastName	DateOfBirth	Gender	Email	phone
1	Mohammed Sharfuddin	Shaik	2002-02-02	Male	sharfoddin28@gmail.com	6303729550
2	Gautham	Lakhude	2003-03-26	Male	lakhundegautham@gmail.com	9381631295
3	Susmitha	Pothireddy	2002-10-26	Female	susmil123@gmail.com	9472549645
4	Sirisha	Yarrampally	2001-05-12	Female	Siril123@gmail.com	9276539234
5	Kusuma	dulam	2002-07-18	Female	kusuma123@gmail.com	9652900626

```
5 rows in set (0.00 sec)
```

Table2: Course

```
mysql> show tables;
```

Tables_in_studentmanagementsystem
course
enrollment
feedback
instructor
score
student

```
6 rows in set (0.00 sec)
```

```
mysql> desc course;
```

Field	Type	Null	Key	Default	Extra
CourseID	int	NO	PRI	NULL	
CourseTitle	varchar(50)	YES	UNI	NULL	
Credits	int	NO		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> insert into course
-> values(1,'Electrical',10),
-> (2,'Electronics',8),
-> (3,'Computer',7),
-> (4,'Civil',4),
-> (5,'Mechanical',6);
```

```
Query OK, 5 rows affected (0.01 sec)
```

```
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from course;
```

CourseID	CourseTitle	Credits
1	Electrical	10
2	Electronics	8
3	Computer	7
4	Civil	4
5	Mechanical	6

```
5 rows in set (0.00 sec)
```

Table3: Instructor

```
mysql> desc instructor;
```

Field	Type	Null	Key	Default	Extra
InstructorID	int	NO	PRI	NULL	
FirstName	varchar(15)	NO		NULL	
LastName	varchar(15)	NO		NULL	
Email	varchar(50)	YES		NULL	

```
4 rows in set (0.00 sec)

mysql> insert into instructor
-> values(1,'Afshan','Shaik','afshan.anudip.org'),
-> (2,'Nagarjuna','Akkineni','nagarjuna@anudip.org'),
-> (3,'Balakrishan','Golla','balakrishan@srit.ac.in'),
-> (4,'Meerimatha','G','meerimatha@srit.ac.in'),
-> (5,'Sathish','Kumar','sathish@srit.ac.in');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from instructor;
```

InstructorID	FirstName	LastName	Email
1	Afshan	Shaik	afshan@anudip.org
2	Nagarjuna	Akkineni	nagarjuna@anudip.org
3	Balakrishan	Golla	balakrishan@srit.ac.in
4	Meerimatha	G	meerimatha@srit.ac.in
5	Sathish	Kumar	sathish@srit.ac.in

```
5 rows in set (0.00 sec)
```

## Table4: Enrollment

```
mysql> desc enrollment;
```

Field	Type	Null	Key	Default	Extra
EnrollmentID	int	NO	PRI	NULL	
EnrollmentDate	date	NO		NULL	
StudentID	int	NO	MUL	NULL	
CourseID	int	NO	MUL	NULL	
InstructorID	int	NO	MUL	NULL	

5 rows in set (0.00 sec)

```
mysql> insert into enrollment
-> values(1,'2024-05-21',1,1,1),
-> (2,'2024-05-22',2,2,2),
-> (3,'2024-05-23',3,3,3),
-> (4,'2024-05-24',4,4,4),
-> (5,'2024-05-25',5,5,5);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> select * from enrollment;
```

EnrollmentID	EnrollmentDate	StudentID	CourseID	InstructorID
1	2024-05-21	1	1	1
2	2024-05-22	2	2	2
3	2024-05-23	3	3	3
4	2024-05-24	4	4	4
5	2024-05-25	5	5	5

5 rows in set (0.00 sec)

## Table5: Score

```
mysql> desc score;
```

Field	Type	Null	Key	Default	Extra
ScoreID	int	NO	PRI	NULL	
CourseID	int	NO	MUL	NULL	
StudentID	int	NO	MUL	NULL	
DateOFEexam	date	NO		NULL	
CreditObtained	int	NO		NULL	

```
5 rows in set (0.00 sec)

mysql> insert into score
-> values(1,1,1,'2024-05-21',10),
-> (2,2,2,'2024-05-21',8),
-> (3,3,3,'2024-05-21',6),
-> (4,4,4,'2024-05-21',9),
-> (5,5,5,'2024-05-21',4);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> select * from score;
```

ScoreID	CourseID	StudentID	DateOFEexam	CreditObtained
1	1	1	2024-05-21	10
2	2	2	2024-05-21	8
3	3	3	2024-05-21	6
4	4	4	2024-05-21	9
5	5	5	2024-05-21	4

```
5 rows in set (0.00 sec)
```

Table6: feedback

```
mysql> desc feedback;
```

Field	Type	Null	Key	Default	Extra
FeedbackID	int	NO	PRI	NULL	
StudentID	int	NO	MUL	NULL	
Date	date	NO		NULL	
InstructorName	varchar(25)	NO		NULL	
Feedback	varchar(255)	NO		NULL	

5 rows in set (0.01 sec)

```
mysql> insert into feedback
-> values(1,1,'2024-05-22','Nagaraju J N','Excellent'),
-> (2,2,'2024-05-22','Nagaraju J N','Best'),
-> (3,3,'2024-05-22','Nagaraju J N','Best'),
-> (4,4,'2024-05-22','Nagaraju J N','Best'),
-> (5,5,'2024-05-22','Nagaraju J N','Improve');
```

Query OK, 5 rows affected (0.01 sec)  
Records: 5 Duplicates: 0 Warnings: 0

```
mysql> select * from feedback;
```

FeedbackID	StudentID	Date	InstructorName	Feedback
1	1	2024-05-22	Nagaraju J N	Excellent
2	2	2024-05-22	Nagaraju J N	Best
3	3	2024-05-22	Nagaraju J N	Best
4	4	2024-05-22	Nagaraju J N	Best
5	5	2024-05-22	Nagaraju J N	Improve

5 rows in set (0.00 sec)

Chatgpt Question:

Generating a normalized table using ChatGPT as a continuation of our earlier

ChatGPT scenario.

Put the below statement in the message box and see what ChatGPT says:

I have created a database of emergency contact information. Kindly help me by

providing normalized tables with understanding.

I got following information from chatgpt

Creating a normalized database for emergency contact information involves organizing the data into multiple related tables to reduce redundancy and improve data integrity. Here's an outline of what the normalized tables could look like along with explanations for each.

### Assumptions:

1. Each individual may have multiple emergency contacts.
2. Each contact might be connected to multiple individuals (e.g., in the case of family members).
3. Contacts may have various types of relationships with the individuals.
4. Address information might be necessary for contacts.

### First Normal Form (1NF)

1. **Individuals:** To store the personal details of the individuals.
2. **Contacts:** To store the details of the emergency contacts.
3. **IndividualContacts:** To store the relationships between individuals and their contacts.

### Tables and Structures

#### 1. Individuals

- **Table Name:** Individuals
- **Columns:**
  - **IndividualID** (Primary Key, unique identifier for each individual)
  - **FirstName**



- **LastName**
- **DateOfBirth**
- **PhoneNumber**
- **Email**
- **Address**

## 2. Contacts

- **Table Name: Contacts**
- **Columns:**

- **ContactID** (Primary Key, unique identifier for each contact)
- **FirstName**
- **LastName**
- **PhoneNumber**
- **Email**
- **RelationshipType** (e.g., parent, sibling, friend)
- **Address**

## 3. IndividualContacts

- **Table Name: IndividualContacts**
- **Columns:**

- **IndividualContactID** (Primary Key, unique identifier for each relationship entry)
- **IndividualID** (Foreign Key, references **Individuals(IndividualID)**)
- **ContactID** (Foreign Key, references **Contacts(ContactID)**)

- **RelationshipType** (e.g., primary contact, secondary contact)

- **Example Data Entries**
- **Individuals Table**

IndividualID	FirstName	LastName	DateOfBirth	PhoneNumber	
1	John	Doe	1980-01-01	123-456-7890	john

### Contacts Table

ContactID	FirstName	LastName	PhoneNumber	Email
1	Jane	Doe	234-567-8901	jane.doe@example.c
2	Mike	Smith	345-678-9012	mike.smith@exampl

### IndividualContacts Table

IndividualContactID	IndividualID	ContactID	RelationshipType
1	1	1	Primary
2	1	2	Secondary

### Explanation

- **Individuals Table:** Contains information about the individuals for whom the emergency contacts are being recorded.
- **Contacts Table:** Stores contact information of the emergency contacts.

- **IndividualContacts Table:** Manages the many-to-many relationship between individuals and their contacts, allowing each individual to have multiple contacts and each contact to be associated with multiple individuals if needed. It also specifies the nature of each relationship.

By following this normalization process, you reduce redundancy (e.g., not repeating contact details for each individual) and maintain data integrity, ensuring that updates to contact information need only be made in one place.

