COMPUTER ORGANIZATION AND ARCHITECTURE

Lecture 9

Advanced Data Definition Language

Dr Victoria Mukami

INTRODUCTION

During the last lecture, we were able to start coding in SQL. We were able to create a database and create tables within the database. This lecture we will work on updating some fields within tables as well as altering various field constraints and data types. Finally, we will work with deleting fields or tables from the database.

Learning objectives

By the end of this topic, you should be able to:

1. Alter the table structure
2. Drop tables and various fields

OVERVIEW

The command we will concentrate on within this lecture is the ALTER and the DROP command. Sometimes, it is necessary to make changes to the structure of the database or even of the tables. This might be necessitated by some oversight during the design phase or if something new came up that needs to be captured. For instance, perhaps the Bustani college has undergone some changes and now needs to capture additional employee data. It would be necessary for the administrator to update the tables either to add fields or to update existing ones.

During the last lecture, we created two tables using the CREATE command and the ERD is as shown in figure 1.

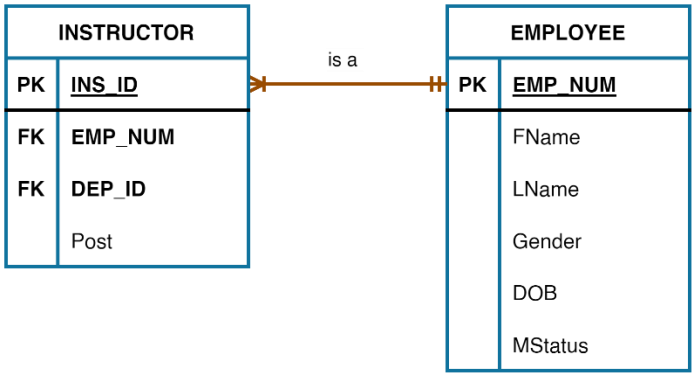

Figure 1: Simplified ERD

We now want to make some changes to this initial ERD by adding a table and at the same time updating some fields within the instructor field.

ALTER TABLE

So far, we have created two tables i.e., Employee and Instructor. Remember for the Instructor table, we could not add Dep_ID as a foreign key. To do so, we will create a department table that has the following fields: Dep_ID and Dept_Name. Table 1 has the records for the Department table.

Table 1: Department Data Table

Dep_ID	Dep_Name
CIT	Computing and Information Technology
MTH	Statistics and Mathematics

The updated ERD diagram will be shown in Figure 3.

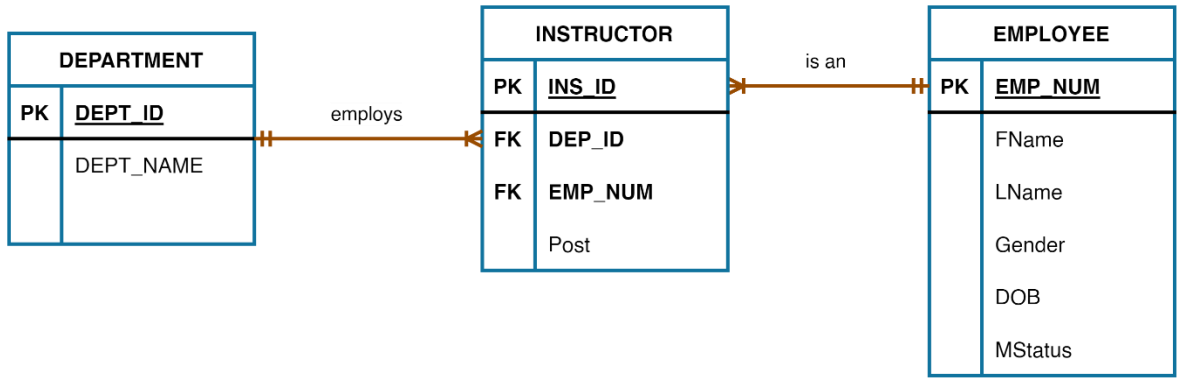

Figure 2: Simplified ERD Diagram

The statements to create the table are shown below.

```
CREATE TABLE Department (
        Dep_ID          CHAR(3)         NOT NULL    UNIQUE,
        Dep_Name        VARCHAR (45)    NOT NULL,
        PRIMARY KEY     (Dep_ID) );
```

The output once all three tables have been created.

```
mysql> SHOW TABLES;
+-------------------+
| Tables_in_bustani |
+-------------------+
| department        |
| employee          |
| instructor        |
+-------------------+
3 rows in set (0.01 sec)
```

The syntax used to make changes to a table is as shown:

```
ALTER TABLE tablename(
        ADD fieldname datatype constraint,
        MODIFY (fieldname datatype),
        ADD PRIMARY KEY (fieldname),
        ADD FOREIGN KEY (fieldname) REFERENCES tablename(fieldname),
        DROP fieldname );
```

Add is used to input new field names, primary keys or foreign keys. Modify is used to update existing columns by changing the datatype while DROP is used to delete a column. Now we would like to update the Dep_ID field in the instructor table to be a foreign key from the department table. Additionally, Dep_ID was not set to not null as shown below.

```
mysql> SHOW COLUMNS FROM Instructor;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Ins_ID  | int         | NO   | PRI | NULL    |       |
| Emp_Num | char(6)     | YES  | MUL | NULL    |       |
| Dep_ID  | char(3)     | YES  |     | NULL    |       |
| Post    | varchar(15) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

We will combine those in the following statement.

```
ALTER TABLE Instructor
        MODIFY Dep_ID CHAR(3) NOT NULL,
        ADD FOREIGN KEY (Dep_ID) REFERENCES Department(Dep_ID);
```

The output is as shown below.

```
mysql> SHOW COLUMNS FROM Instructor;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Ins_ID  | int         | NO   | PRI | NULL    |       |
| Emp_Num | char(6)     | YES  | MUL | NULL    |       |
| Dep_ID  | char(3)     | NO   | MUL | NULL    |       |
| Post    | varchar(15) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

ALTER COLUMN

We can alter a column's data type. Let us refer to the data types from the three tables that we have created so far in Table 2.

Table 2: Table Fields and associated datatypes

Table	Field Name	Datatype
Instructor	Ins_ID	INTEGER
	Emp_Num	CHAR (6),
	Dep_ID	CHAR (3),
	Post	VARCHAR (15)
Employee	Emp_Num	CHAR (6)
	FName	VARCHAR (25)
	LName	VARCHAR (25)
	Gender	CHAR (1)
	DOB	DATE
	MStatus	VARCHAR(15)
Department	Dep_ID	CHAR(3)
	Dep_Name	VARCHAR (45)

Assuming we wanted to change the employee Gender field from CHAR to a VARCHAR then we would have the following code.

```
ALTER TABLE Employee
        MODIFY Gender VARCHAR(10);
```

The output is as shown below

```
mysql> ALTER TABLE Employee
    -> MODIFY Gender VARCHAR(10);
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Employee;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Emp_Num | char(6)     | NO   | PRI | NULL    |       |
| FName   | varchar(25) | NO   |     | NULL    |       |
| LName   | varchar(25) | NO   |     | NULL    |       |
| Gender  | varchar(10) | YES  |     | NULL    |       |
| DOB     | date        | NO   |     | NULL    |       |
| MStatus | varchar(15) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

As you can see from the results above, the datatype for Gender has been changed from a CHAR to VARCHAR.

Assuming we wanted to add a column to our Instructor table called School. To do so, we would use the following statement.

```
ALTER TABLE Instructor
        ADD  School        VARCHAR(35)       NOT NULL;
```

The output is as shown

```
mysql> DESCRIBE Instructor;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Ins_ID  | int         | NO   | PRI | NULL    |       |
| Emp_Num | char(6)     | YES  | MUL | NULL    |       |
| Dep_ID  | char(3)     | NO   | MUL | NULL    |       |
| Post    | varchar(15) | NO   |     | NULL    |       |
| School  | varchar(35) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Assuming we made a mistake and the school table was meant to be added to the department, we would need to delete the School field and then add it to the Department table.

First, we delete the school field from the Instructor table. This is achieved with a combination of the ALTER and DROP commands. The syntax is shown below.

```
ALTER TABLE tablename
        DROP columnname;
```

To delete the school column from the Instructor table we use the following code.

```
ALTER TABLE Instructor
        DROP School;
```

The output is shown below without a School column.

```
mysql> ALTER TABLE Instructor
    -> DROP School;
Query OK, 0 rows affected (0.03 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Instructor;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Ins_ID  | int         | NO   | PRI | NULL    |       |
| Emp_Num | char(6)     | YES  | MUL | NULL    |       |
| Dep_ID  | char(3)     | NO   | MUL | NULL    |       |
| Post    | varchar(15) | NO   |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Second, we add the field to the correct table – Department.

```
ALTER TABLE Department
        ADD  School        VARCHAR(35)       NOT NULL;
```

The output is shown below

```
mysql> ALTER TABLE Department
    -> ADD School VARCHAR(35) NOT NULL;
Query OK, 0 rows affected (0.04 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESCRIBE Department;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| Dep_ID   | char(3)     | NO   | PRI | NULL    |       |
| Dep_Name | varchar(45) | NO   |     | NULL    |       |
| School   | varchar(35) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

DROP TABLE

Finally, the last SQL command we review is called DROP table. The DROP table command is used to delete a table and is used in the following format.

```
DROP TABLE tablename;
```

A table can only be deleted if it does not have a relationship with any other table [1]. As seen previously the DROP statement can also be used to delete a column.

SUMMARY

This lecture has focused on two DDL commands: ALTER and DROP. We used the CREATE command to add a table then we used the ALTER command to add a foreign key. We also used the ALTER command to add a field, drop a field and change the data type of a particular table. Finally, we used the DROP command to delete a table.

DISCUSSION TOPIC

This discussion topic is a continuation of lecture 8 during which we created tables for the Lands Office. Try to think of other tables than can be added to your database and the kind of relationships they will have with your already existing tables. Create the tables within the database then use the ALTER key to add the new foreign keys to the existing tables. Look for guidance from your instructor.

REFERENCES

[1] Database systems: design, implementation, and management, Coronel, C., & Morris, S, Cengage Learning, 2019.

[2] Database Systems: A Practical Approach to Design, Implementation, and Management, Connolly, T., & Begg, C., Pearson, 2015.

[3] Fundamentals of database systems, Elmasri, R., & Navathe, S. B., Pearson Education Limited, 2016.