MYSQL - ALTER, DROP, RENAME, MODIFY

WHAT IS THE ALTER COMMAND?

As the saying goes **Change is the only constant**

With time business requirements change as well. As business requirements change, Database designs need changing as well.

MySQL provides the **ALTER** function that helps us **incorporate the changes to the already existing database design**.

The alter command is used to modify an existing database, table, view or other database objects that might need to change during the life cycle of a database.

Let's suppose that we have completed our database design and it has been implemented. Our database users are using it and then they realize some of the vital information was left out in the design phase. They don't want to lose the existing data but just want to incorporate the new information. The alter command comes in handy in such situations. We can use the alter command to change the data type of a field from say string to numeric, change the field name to a new name or even add a new column in a table.

Alter- syntax

The basic syntax used to add a column to an already existing table is shown below

```
ALTER TABLE `table_name` ADD COLUMN `column_name`
`data_type`;
```

HERE

- **"ALTER TABLE `table_name`"** is the command that tells MySQL server to modify the table named `table_name`.
- **"ADD COLUMN `column_name` `data_type`"** is the command that tells MySQL server to add a new column named `column_name` with data type `data_type'.

Let's suppose that Myflix has introduced online billing and payments. Towards that end, we have been asked to add a field for the credit card number in our members table. We can use the ALTER command to do that. Let's first look at the structure of the members table before we make any amendments. The script shown below helps us to do that.

```
SHOW COLUMNS FROM `members`;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
full_names	varchar(350)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
physical_address	varchar(255)	YES		NULL	
postal_address	varchar(255)	YES		NULL	
contact_number	varchar(75)	YES		NULL	

email	varchar(255)	YES		NULL	

We can use the script shown below to add a new field to the members table.

```
ALTER TABLE `members` ADD COLUMN `credit_card_number`
VARCHAR(25);
```

Executing the above script in MySQL against the Myflixdb adds a new column named credit card number to the members table with VARCHAR as the data type. Executing the show columns script gives us the following results.

```
SHOW COLUMNS FROM `members`;
```

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
full_names	varchar(350)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
physical_address	varchar(255)	YES		NULL	
postal_address	varchar(255)	YES		NULL	

contact_number	varchar(75)	YES	NULL
email	varchar(255)	YES	NULL
credit_card_number	varchar(25)	YES	

As you can see from the results returned, credit card number has been added to the members table. The data contained in the members' data is not affected by the addition of the new column.

WHAT IS THE DROP COMMAND?

The DROP command is used to

1. Delete a database from MySQL server
2. Delete an object (like Table , Column)from a database.

Let's now look at practical examples that make use of the DROP command.

In our previous example on the Alter Command, we added a column named credit card number to the members table.

Suppose the online billing functionality will take some time and we want to DROP the credit card column

We can use the following script

```
ALTER TABLE `members` DROP COLUMN `credit_card_number`;
```

Executing the above script drops the column credit_card_number from the members table

Let's now look at the columns in the members table to confirm if our column has been dropped.

```
SHOW COLUMNS FROM `members`;
```

Executing the above script in MySQL workbench against the myflixdb gives us the following results.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
`membership_number`	`int(11)`	`NO`	`PRI`	`NULL`	`auto_increment`
`full_names`	`varchar(350)`	`NO`		`NULL`	
`gender`	`varchar(6)`	`YES`		`NULL`	
`date_of_birth`	`date`	`YES`		`NULL`	
`physical_address`	`varchar(255)`	`YES`		`NULL`	
`postal_address`	`varchar(255)`	`YES`		`NULL`	
`contact_number`	`varchar(75)`	`YES`		`NULL`	
`email`	`varchar(255)`	`YES`		`NULL`	

Notice that the credit card number has been dropped from the fields list.

DROP TABLE

The syntax to DROP a table from Database is as follow -

```
DROP TABLE `sample_table`;
```

Let's Look at an example

```
DROP TABLE `categories_archive`;
```

Executing the above script deletes the table named ` categories_archive ` from our database.

WHAT IS THE RENAME COMMAND?

The rename command is used to **change the name of an existing database object(like Table,Column) to a new name**.

Renaming a table does not make it lose any data contained within it.

Syntax:-

The rename command has the following basic syntax.

```
RENAME TABLE `current_table_name` TO `new_table_name`;
```

Let's suppose that we want to rename the movierentals table to movie_rentals, we can use the script shown below to achieve that.

```
RENAME TABLE `movierentals` TO `movie_rentals`;
```

Executing the above script renames the table `movierentals` to `movie_rentals`.

We will now rename the movie_rentals table back to its original name.

```
RENAME TABLE `movie_rentals` TO `movierentals`;
```

CHANGE KEYWORD

Change Keywords allows you to

1. Change Name of Column
2. Change Column Data Type
3. Change Column Constraints

Let's look at an example. The full names field in the members table is of varchar data type and has a width of 150.

```
SHOW COLUMNS FROM `members`;
```

Executing the above script in MySQL workbench against the myflixdb gives us the following results.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
full_names	varchar(150)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
physical_address	varchar(255)	YES		NULL	
postal_address	varchar(255)	YES		NULL	
contact_number	varchar(75)	YES		NULL	
email	varchar(255)	YES		NULL	

Suppose we want to

1. Change the field name from "full_names" to "fullname
2. Change it to char data type with a width of 250
3. Add a NOT NULL constraint

We can accomplish this using the change command as follows -

```
ALTER TABLE `members` CHANGE COLUMN `full_names`
`fullname` char(250) NOT NULL;
```

Executing the above script in MySQL workbench against myflixdb and then executing the show columns script given above gives the following results.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
fullnames	char(250)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
physical_address	varchar(255)	YES		NULL	
postal_address	varchar(255)	YES		NULL	
contact_number	varchar(75)	YES		NULL	
email	varchar(255)	YES		NULL	

MODIFY KEYWORD

The MODIFY Keyword allows you to

1. Modify Column Data Type
2. Modify Column Constraints

In the CHANGE example above, we had to change the field name as well other details. **Omitting the field name from the CHANGE statement will generate an error.** Suppose we are only interested in changing the data type and constraints on the field without affecting the field name, we can use the MODIFY keyword to accomplish that.

The script below changes the width of "fullname" field from 250 to 50.

```
ALTER TABLE `members`MODIFY `fullname` char(50) NOT NULL;
```

Executing the above script in MySQL workbench against myflixdb and then executing the show columns script given above gives the following results shown below.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
fullnames	char(50)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
physical_address	varchar(255)	YES		NULL	
postal_address	varchar(255)	YES		NULL	

contact_number	varchar(75)	YES		NULL	
email	varchar(255)	YES		NULL	

AFTER KEYWORD

Suppose that we want to add a new column at a specific position in the table.

We can use the alter command together with the AFTER keyword.

The script below adds "date_of_registration" just after the date of birth in the members table.

```
ALTER TABLE  `members` ADD  `date_of_registration` date
NULL AFTER  `date_of_birth`;
```

Executing the above script in MySQL workbench against myflixdb and then executing the show columns script given above gives the following results shown below.

Field	**Type**	**Null**	**Key**	**Default**	**Extra**
membership_number	int(11)	NO	PRI	NULL	auto_increment
fullnames	char(50)	NO		NULL	
gender	varchar(6)	YES		NULL	
date_of_birth	date	YES		NULL	
date_of_registration	date	YES		NULL	

physical_address	varchar(255)	YES	NULL
postal_address	varchar(255)	YES	NULL
contact_number	varchar(75)	YES	NULL
email	varchar(255)	YES	NULL

Note: The Highlighted row is **added after date_of_birth** cloumn

Summary

- The alter command is used when we want to modify a database or any object contained in the database.
- The drop command is used to delete databases from MySQL server or objects within a database.
- The rename command is used to change the name of a table to a new table name.
- The Change keyword allows you to change a column name , data type and constraints
- The Modify Keyword allows you to modify a column data type and constraints
- The After keyword is used to specify position of a column in a table