

Table of Contents

1. Changing the Table Design.....	1
2. MySQL : Change column and Modify column.....	3
3. Temporary Tables.....	4

1. Changing the Table Design

The alter table statement allows you to make some changes to a table that already contains data.

This is a simplified discussion of the alter table statement including the simpler alterations allowed. You can add and drop columns, constraints, indexes and modify columns data types. If the table contains data, some of these changes are restricted. Some of the allowed alter table statements produce warning since the alter table statement can be run but the engine does not accept that change. When you are using the alter table statement always follow it with the show warnings command.

I am also assuming that strict mode is enabled; if not then some of the changes you can make with Alter Table will change the data in the table.

Demo 01: An alter query to add another column to an existing table

```
Create table ddl_alter (id integer primary key);
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	

```
alter table ddl_alter
add d_office varchar(10);
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
d_office	varchar(10)	YES		NULL	

```
alter table ddl_alter
add constraint office_un unique (d_office);
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
d_office	varchar(10)	YES	UNI	NULL	

Demo 02: Adding more than one column. Delimit the column collection with parentheses.

```
alter table ddl_alter
add ( e_ssn char(11)
, e_namefirst varchar(20)
, e_salary numeric(6) );
```

```
desc ddl_alter;
```

Field	Type	Null	Key	Default	Extra
e_ssn	char(11)				
e_namefirst	varchar(20)				
e_salary	numeric(6)				

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
d_office	varchar(10)	YES	UNI	NULL	
e_ssn	char(11)	YES		NULL	
e_namefirst	varchar(20)	YES		NULL	
e_salary	decimal(6,0)	YES		NULL	

Demo 03: You can drop a column.

```
alter table ddl_alter
drop column e_ssn;
```

Named constraints can be dropped by using an Alter table statement. You can use the system defined constraint names. You can also drop some constraints by naming the constraint type- for example:

Demo 04: Dropping the primary key; this does not drop the attribute - it drops the pk constraint

```
alter table ddl_alter
drop primary key;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		NULL	
d_office	varchar(10)	YES	UNI	NULL	
e_namefirst	varchar(20)	YES		NULL	
e_salary	decimal(6,0)	YES		NULL	

Demo 05: Increasing the width of a column

```
alter table ddl_alter
modify e_namefirst varchar(25);
```

Demo 06: Suppose you want to decrease the width of a column and you already have data in the table(this is run under strict mode)

```
insert into ddl_alter values (101, 'sales', 'anastasia-marie', 500);
```

```
alter table ddl_alter
modify e_namefirst varchar(10);
```

```
ERROR 1265 (01000): Data truncated for column 'e_namefirst' at row 1
```

```
select * from ddl_alter;
```

id	d_office	e_namefirst	e_salary
101	sales	anastasia-marie	500

```
desc ddl_alter;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		NULL	
d_office	varchar(10)	YES	UNI	NULL	
e_namefirst	varchar(25)	YES		NULL	

```
| e_salary      | decimal(6,0) | YES |      | NULL      |      |
+-----+-----+-----+-----+-----+-----+
```

Demo 07: Update that one row to a short name and then try changing the column width. That works because the current data would not be truncated

```
update ddl_alter set e_namefirst = 'sue';
```

```
alter table ddl_alter
modify e_namefirst varchar(10);
```

```
desc ddl_alter;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   |     | NULL    |      |
| d_office   | varchar(10)   | YES  | UNI | NULL    |      |
| e_namefirst | varchar(10)   | YES  |     | NULL    |      |
| e_salary   | decimal(6,0)  | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

2. MySQL : Change column and Modify column

Demo 08: You can use Modify if you want to change the data type of a column without renaming the column.

```
Alter table ddl_alter
Modify e_salary float not null;
```

```
desc ddl_alter;
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   |     | NULL    |      |
| d_office   | varchar(10)   | YES  | UNI | NULL    |      |
| e_namefirst | varchar(10)   | YES  |     | NULL    |      |
| e_salary   | float         | NO   |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

Demo 09: You need to include any features of the current column that you want to keep for the new column definition.

```
Alter table ddl_alter
Modify e_salary real;
```

```
desc ddl_alter; Note that we lost the not null feature in the last alter
statement.
```

```
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   |     | NULL    |      |
| d_office   | varchar(10)   | YES  | UNI | NULL    |      |
| e_namefirst | varchar(10)   | YES  |     | NULL    |      |
| e_salary   | double        | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
```

Demo 10: Change a column-

This version does not work

```
Alter table ddl_alter
change e_namefirst varchar(30) not null;
```

You need to provide a full definition on the new column including the name. If you don't want to change the name you can simply repeat it in the new column definition.

```
Alter table ddl_alter
change e_namefirst e_namefirst varchar(30) not null;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		NULL	
d_office	varchar(10)	YES	UNI	NULL	
e_namefirst	varchar(30)	NO		NULL	
e_salary	double	YES		NULL	

3. Temporary Tables

You can create temporary tables by adding the word Temporary before table.

Create Temporary Table *tblName* ...

Temporary table are session scoped. If you exit your session or lose your connection the temporary tables you created are removed. You can drop them explicitly.

It is possible to create a temporary table with the same name as a permanent table. If you want to drop a temporary table it is a good idea to include the word temporary in the drop statement.

```
drop temporary table if exists z_1;
```

If you do not include the word temporary in the drop statement, then doing the drop twice could delete a permanent table with that name.

You can create a temporary table using the regular Create Table statement

```
Create temporary table tblX ( col_1 int , col_2 int);
```

You can also make a temporary table with the same columns as another table with the Like syntax.

```
Create temporary table z_1 like zoo_animals;
```