

# Manipulating data.: How Do I Remove Duplicates From My Databases?

**Removing duplicate rows** from database tables can be a problem. The problem is that with the common SQL syntax

```
DELETE FROM mytable WHERE {where condition};
```

.. you cannot specify a where-condition that will be satisfied with all duplicates minus one. This SQL will remove ALL rows that satisfy the where-condition. And what you wanted was to remove all but one! SQLyog will warn you, but cannot do anything else! [Refer to http://webyog.com/faq/28\\_70\\_en.html](http://webyog.com/faq/28_70_en.html)

If you know that there are for instance 4 duplicate rows you can of course

```
DELETE FROM mytable WHERE {where condition} LIMIT 3;
```

.. but if you don't know the numbers of duplicates (and you even might not know for which rows of a table duplicates exist) you will have to execute a

```
SELECT COUNT(*) from mytable WHERE {where condition};
```

... for every row. When duplicates exist it most likely is because of a buggy application and hundreds or thousands of such duplicate rows may exist.

**A more efficient solution** to your problem is to create a copy of the table and use the SQL syntax "INSERT IGNORE INTO..." or "REPLACE INTO..." instead of just "INSERT INTO".

If your old/source table is like

```
CREATE TABLE `oldtest` (  
  `ID` int(10) unsigned NOT NULL auto_increment,  
  `n` int(11) default NULL,  
  `t` varchar(50) default NULL,  
  PRIMARY KEY (`ID`)  
)
```

or just

```
CREATE TABLE `test` (  
  `n` int(11) NOT NULL,  
  `t` varchar(50) NOT NULL,  
)
```

then create a new/target table like (note: you define a PK on ALL or at least A LOT of columns of the table)

```
CREATE TABLE `test` (  
  `n` int(11) NOT NULL,
```

# Manipulating data.: How Do I Remove Duplicates From My Databases?

```
`t` varchar(50) NOT NULL,  
PRIMARY KEY (`n`,`t`)  
)
```

Now you need to read values from source (`oldtest`) and for every row in source execute

```
INSERT IGNORE INTO newtest (n,t)  
values (n_value_for_the_source_row,t_value_for_the_source_row);  
(or REPLACE INTO...)
```

INSERT IGNORE INTO will skip duplicate rows in target, REPLACE INTO will overwrite, but the result will be the same: only 1 row with the same data!

Now you can ALTER TABLE, drop the 'intermediate multi-column PK', create a new ID column and define it as the PK.

However there is no way to do this in 'pure' SQL. You have more options:

- 1) using an external script/application reading the source on a per row base and INSERT IGNORE/REPLACE INTO the target.
- 2) use a Stored Procedure (with a cursor that 'runs through' the source row-by-row and does the same)
- 3) You may use SQLyog Import External Data Tool. It is very easy actually!

## With SQLyog Import External Data Tool and the above example do this:

- a) Create the target table (with the 'intermediate multi-column PK') in advance in another database than source
- b) Create a DNS with the MyODBC driver 3.51 pointing to the database of the source
- c) Migrate from Source to Target with the Import External Data Tool:

-- in the 'map' dialogue uncheck the current PK column (if there is any)  
-- use 'advanced' setting like attached screenshot (the Import External Data Tool will REPLACE INTO)

- 4) ALTER TABLE target: drop the 'intermediate multi-column PK', create the new ID-column and define it as a PK!

(Note however that re-organizing or renumbering PK's may cause problems with existing applications, as the PK of a table may be referenced by Foreign Key or a application pointer. That we cannot help! Always backup you data before replacing the original tables with new tables created like described here!)

# Manipulating data.: How Do I Remove Duplicates From My Databases?

Unique solution ID: #1133

Author: Peter Laursen

Last update: 2007-07-01 10:10