

Chapter 5

SQL Queries to Add, Modify, and Delete Table Data

Skills you will learn: How to add, alter and delete data in tables using SQL queries.

As discussed in Chapter 5 of *The Data Journalist*, there are three main types of queries that alter data in tables. The first is the update query, the second the insert query, and the last the delete query. You'll use these queries when you want to change data in your tables, add new data, or remove data you no longer need. You need to be extremely careful when using any of these queries because they make irreversible changes to the data in your tables. It is best to work with a copy of whichever table(s) you are modifying, rather than the originals.

Make sure you have all your data backed up before you start running such queries.

UPDATE queries

An UPDATE query is used to change data in existing records and fields in a table. It can be used to overwrite existing values or populate a new field. We'd recommend rather than overwriting values that you add a new field to a table and make the changes there, only deleting the original field once you are satisfied you have not made any mistakes, if then.

The basic syntax for an UPDATE query is:

```
UPDATE tablename
```

```
SET columnname = [some string or numeric value or function that creates one]
```

```
WHERE [you can use standard WHERE syntax here]
```

An update query can be used with a join as well, updating the values in one table based on the results of a join.

```
UPDATE tablename1, tablename2
```

```
SET tablename1.fieldname = [some value, or a field from tablename2]
```

WHERE tablename1.joinfield = tablename2.joinfield

You can use a one-to-one join or a one-to-many join. An example of the latter use is in data cleaning where you might create a lookup table of clean values, then join on the old, dirty values to populate a new clean value field in the main table. You can read more about data cleaning in the online Appendix A.

INSERT queries

Insert queries are used to add new records to an existing table. You will most likely use this kind of query if you have additional data, such as a new month or year of data, to add to an existing table, and you want to clean it up before adding it, or for adding data to a table based on a join of two or more other tables.

You can also import external data directly into an existing table. See the tutorials **Making Tables and Importing Data into Access** and **Making Tables and Importing Data into MySQL** for instructions on importing data.

The syntax for an insert query in MySQL and Microsoft Access is:

```
INSERT INTO tablename
```

```
SELECT [data with the same number of columns, and the same data types]
```

```
FROM one or more tables
```

MySQL also has a variant of INSERT that uses the REPLACE keyword in place of INSERT.

If REPLACE is used, if a row in the new table has an identical value for a primary key or unique index as exists in the old table, the row in the old table will be overwritten. This is a useful behaviour if you have obtained data for which some rows have been updated and wish to bring the data in a table up to date. This happens often with live datasets in which details will change as certain events occur, such as people paying parking tickets or new information being discovered in an ongoing investigation, to give two examples. Be sure to keep a backup copy of your data in case something goes wrong.

Delete queries

Delete queries are used to get rid of one or more rows in a table. By get rid of, we mean permanently, so be sure you have another copy of the table, or a backup, before you use a delete query.

The query uses the same syntactical pattern as a SELECT query.

```
DELETE
```

```
FROM tablename
```

```
WHERE [standard where conditions]
```

Be sure to include a WHERE clause unless you intend to delete all of the rows in a table. The effect of the WHERE clause will be to limit the deleted rows to those rows that meet the test set in the WHERE line. Rows not meeting the test will be untouched.