The UPDATE Statement

The UPDATE statement is used to modify or update existing data. The basic syntax for the UPDATE statement is shown here:

```
UPDATE  table_name  SET  column_name  =  expression
WHERE  search_condition
```

Updating Rows

Building on our sample table, *items*, we'll first update the *junk food* row that we inserted earlier without a price. To identify the row, specify *fried pork skins* in the search condition. To set (update) the price to $2, use the following statement:

```
UPDATE  items  SET  price  =  2.00
WHERE  item_desc  =  'fried  pork  skins'
```

Now select the *junk food* row by using this query:

```
SELECT  *  FROM  items
WHERE  item_desc  =  'fried  pork  skins'
```

The output for the *junk food* row appears as follows, with the original *NULL* value for *price* replaced by *2.00*:

```
item_category      item_id  price       item_desc
---------------    -------  ----------  ----------------
junk  food           2       2.00        fried  pork  skins
```

To increase the price of this item by 10 percent, you would run the following statement:

```
UPDATE  items  SET  price  =  price  *  1.10
WHERE  item_desc  =  'fried  pork  skins'
```

Now if you select the *junk food* row, you will notice the price has been changed to $2.20 ($2 multiplied by 1.10). The prices of the other items have not changed.

You can update more than one row by using an UPDATE statement. For example, to update all the rows in the *items* table by increasing their *price* values by 10 percent, run the following statement:

```
UPDATE  items  SET  price  =  price  *  1.10
```

Now if you examine the *items* table, it will look like this:

```
tem_category       item_id  price       item_desc
---------------    -------  ----------  ----------------
health  food         1       4.40        tofu  6  oz.
```

```
junk  food          2          2.42          fried  pork  skins
toys                3          NULL           No  desc
```

Rows with a value of *NULL* for *price* will not be affected because *NULL* * 1.10 = *NULL*. This is not a problem; you will not get an error.

The DELETE Statement

The DELETE statement is used to remove (delete) a row or rows from a table. DELETE does not affect the table definition; it simply deletes rows of data from the table. The basic syntax for the DELETE statement is shown here:

```
DELETE  [FROM]  table_name
WHERE  search_condition
```

The FROM keyword is optional.

Deleting Individual Rows

By using the WHERE clause with DELETE, you can specify certain rows to delete from a table. For example, to delete all rows from the *items* table that have an *item_category* value of *toys*, run the following statement:

```
DELETE  FROM  items
WHERE  item_category  =  'toys'
```

This statement deletes one row from our *items* table.

Deleting All Rows

To delete all rows from a table, use DELETE without a WHERE clause. The following DELETE statement will delete all rows in the *two_newest_items* table.

```
DELETE  FROM  two_newest_items
```

The *two_newest_items* table is now an empty table—it contains no data. If you want to delete the table definition as well, use the DROP TABLE command.