Useful Statements in SQL

The INSERT Statement

The INSERT statement is used to insert new rows into a table.

Syntax

```
INSERT INTO table_name (column1, column2, …) VALUES
(value1, value2, …)
```

For example, to INSERT a new row into the *sales_contacts* table with the following information: Dominic Marsat, 01483304030, 07904579320

```
INSERT INTO sales_contacts (first_name, last_name,
direct_tel, mobile) VALUES ('Dominic', 'Marsat', '01484
304030', '07904579320')
```

Notes

- The *contactid* is generated automatically by the database and is not specified in the INSERT statement. This is the same for all identity columns in other tables/databases
- Columns/values not specified in the INSERT statement will be inserted with NULL values unless the database is setup otherwise (CRM database inserts empty strings or zeros instead of NULLS)
- In order to link a contact to a company the *companyid* column should be used
- The *date_created* column is inserted automatically by the database
- Telephone numbers in the CRM database are stored as strings not numbers. This is to accommodate the '+' symbol or other non-number characters

The UPDATE Statement

The UPDATE statement is used to modify the data in a table.

Syntax

```
UPDATE table_name SET column_name = new_value WHERE
column_name = some_value
```

For example, in order to UPDATE one column in a row from the *sales_contacts* table: Change the first name from 'Dominic' to 'Pascal' for contact Dominic Marsat. If the *contactid* is known then…

```
UPDATE sales_contacts SET first_name='Pascal' WHERE
contactid=100
```


Useful Statements in SQL

If the *contactid* is unknown then it's best to specify a few columns in the WHERE clause to make sure the correct contact is modified by the UPDATE statement.

```
UPDATE sales_contacts SET first_name='Pascal' WHERE
first_name='Dominic' AND last_name='Marsat' AND
mobile='07904579320'
```

In order to UPDATE several columns in a row from the *sales_contacts* table: Change the first name from 'Dominic' to 'Pascal' and the last name from 'Marsat' to 'Lawton for the contact Dominic Marsat…

```
UPDATE sales_contacts SET first_name='Pascal',
last_name='Lawton' WHERE contactid=100
```

The DELETE Statement

The DELETE statement is used to delete rows in a table.

Syntax

```
DELETE FROM table_name WHERE column_name=some_value
```

To DELETE the contact Dominic Marsat from the *sales_contacts* table:

```
DELETE FROM sales_contacts WHERE contactid=100
```

It is possible to delete all rows in a table without deleting the table. This means that the table structure, attributes and indexes will be intact:

```
DELETE * FROM table_name
```

ORDER BY Keyword

The ORDER BY keyword is used to sort the result.

To display the companies from the *sales_company* table in alphabetical order:

```
SELECT company_name FROM sales_company ORDER BY
company_name
```

To display the companies in reverse alphabetical order:

```
SELECT company_name FROM sales_company ORDER BY
company_name DESC
```