

There are 5 basic Open SQL statements which will be used regularly in programs from here forward. These are SELECT, INSERT, UPDATE, MODIFY and DELETE.

The SELECT statement allows one to select records from database tables which will then be used in a program.

INSERT allows new records to be inserted into a database table.

UPDATE allows records which already exist in the table to be modified.

MODIFY performs a similar task to update, with slight differences which we will discuss shortly.

DELETE, of course, allows records to be deleted from a table.

Whenever any of these statements are used in an ABAP program, it is important to check whether the action executed has been successful.

#### Insert Statement

```
DATA wa_employees LIKE zemployees.  
  
wa_employees-employee = '10000006'.  
wa_employees-surname = 'WESTMORE'.  
wa_employees-forename = 'BRUCE'.  
wa_employees-title = 'MR'.  
wa_employees-dob = '19921213'.  
  
INSERT zemployees FROM wa_employees.
```

The check statement can then be formulated as follows, meaning that if the record is inserted correctly, the system will state this, if not then the SY-SUBRC code which will not equal zero is will be displayed

```
IF sy-subrc = 0.  
  WRITE 'Record Inserted Correctly'.  
ELSE.  
  WRITE: 'We have a return code of ', sy-subrc.  
ENDIF.
```

### Update Statement

The UPDATE statement allows one or more existing records in a table to be modified at the same time. In this example it will just be applied to one, but for more the same principles generally apply.

```
wa_employees-employee = '10000006'.  
wa_employees-surname = 'EASTMORE'.  
wa_employees-forename = 'ANDY'.  
wa_employees-title = 'MR'.  
wa_employees-dob = '19921213'.  
  
UPDATE zemployees FROM wa_employees.
```

## Modify Statement

The MODIFY statement could be said to be like a combination of the INSERT and UPDATE statements. It can be used to either insert a new record or modify an existing one. Generally, though the INSERT and UPDATE statements are more widely used for these purposes, since these offer greater clarity.

In the first section of code in the image below, since employee number is the key field, and '10000006' already exists, the record for that employee number will be updated with the new name in the code. A validation check is performed next. The CLEAR statement is then used so a new entry can be put into the work area, and then employee 10000007 is added. Since this is a new, unique key field value, a new record will be inserted, and another validation check executed:

```
wa_employees-employee = '10000006'.
wa_employees-surname = 'NORTHMORE'.
wa_employees-forename = 'PETER'.
wa_employees-title = 'MR'.
wa_employees-dob = '19921213'.

MODIFY zemployees FROM wa_employees.

IF sy-subrc = 0.
  WRITE: / 'Record Modified Correctly'.
ELSE.
  WRITE: / 'We have a return code of ', sy-subrc.
ENDIF.
```

```
CLEAR wa_employees.

wa_employees-employee = '10000007'.
wa_employees-surname = 'SOUTHMORE'.
wa_employees-forename = 'SUSAN'.
wa_employees-title = 'MRS'.
wa_employees-dob = '19921113'.

MODIFY zemployees FROM wa_employees.

IF sy-subrc = 0.
  WRITE: / 'Record Modified Correctly'.
ELSE.
  WRITE: / 'We have a return code of ', sy-subrc.
ENDIF.
```

When this is executed, and the data then viewed in the Data Browser, employee number 10000006 will have been updated with the new name, Peter Northmore, and a new record will have been created for number 10000007, Susan Southmore:

000	10000006	NORTHMORE	PETER
000	10000007	SOUTHMORE	SUSAN

## Delete Statement

Unlike the previous SQL statements, the DELETE statement does not take into account most fields, only the primary key field. When you want to delete a record from a table, the system only needs to be told what the primary key field value for that record is.

In this example, the last record created, for the employee Susan Southmore will be deleted. For the zemployees table, there are two key fields, the client field and the employee number. The client field is dealt with automatically by the system, and this never has to be included in programs, so the important field here is the employee number field. The syntax to delete the last record created in the previous section would be this:

```
CLEAR wa_employees.  
  
wa_employees-employee = '10000007'.  
  
DELETE zemployees FROM wa_employees.
```

```
CLEAR wa_employees.  
  
DELETE FROM zemployees WHERE surname = 'BROWN'.  
  
IF sy-subrc = 0.  
    WRITE: / '2 Records Deleted Correctly'.  
ELSE.  
    WRITE: / 'We have a return code of ', sy-subrc.  
ENDIF.
```

In the above case all records with Surname = 'BROWN' will be deleted.

Note that, if one uses the following piece of code, without specifying the logic addition, all of the records will in fact be deleted:

```
DELETE FROM zemployees.
```