

# SQL Cursor

- A cursor is a pointer to the private memory area allocated by the Oracle Server. It is used to handle the result set of a `SELECT` statement.
- There are two types of cursors: implicit and explicit.
  - **Implicit:** Created and managed internally by the Oracle Server to process SQL statements
  - **Explicit:** Declared explicitly by the programmer



Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

## SQL Cursor

You have already learned that you can include SQL statements that return a single row in a PL/SQL block. The data retrieved by the SQL statement should be held in variables using the `INTO` clause.

Where Does the Oracle Server Process SQL Statements?

The Oracle Server allocates a private memory area called the context area for processing SQL statements. The SQL statement is parsed and processed in this area. The information required for processing and the information retrieved after processing are all stored in this area. You have no control over this area because it is internally managed by the Oracle Server.

A cursor is a pointer to the context area. However, this cursor is an implicit cursor and is automatically managed by the Oracle Server. When the executable block issues a SQL statement, PL/SQL creates an implicit cursor.

Types of Cursors

There are two types of cursors:

**Implicit:** An implicit cursor is created and managed by the Oracle Server. You do not have access to it. The Oracle Server creates such a cursor when it has to execute a SQL statement.

SQL Cursor (continued) Types of Cursors (continued)

**Explicit:** As a programmer, you may want to retrieve multiple rows from a database table, have a pointer to each row that is retrieved, and work on the rows

one at a time. In such cases, you can declare cursors explicitly depending on your business requirements. A cursor that is declared by programmers is called an explicit cursor. You declare such a cursor in the declarative section of a PL/SQL block.

## SQL Cursor Attributes for Implicit Cursors

Using SQL cursor attributes, you can test the outcome of your SQL statements.

<b>SQL%FOUND</b>	Boolean attribute that evaluates to <b>TRUE</b> if the most recent SQL statement affected at least one row
<b>SQL%NOTFOUND</b>	Boolean attribute that evaluates to <b>TRUE</b> if the most recent SQL statement did not affect even one row
<b>SQL%ROWCOUNT</b>	An integer value that represents the number of rows affected by the most recent SQL statement

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

### SQL Cursor Attributes for Implicit Cursors

SQL cursor attributes enable you to evaluate what happened when an implicit cursor was last used. Use these attributes in PL/SQL statements but not in SQL statements.

You can test the `SQL%ROWCOUNT`, `SQL%FOUND`, and `SQL%NOTFOUND` attributes in the executable section of a block to gather information after the appropriate DML command executes. PL/SQL does not return an error if a DML statement does not affect rows in the underlying table. However, if a `SELECT` statement does not retrieve any rows, PL/SQL returns an exception. Observe that the attributes are prefixed with `SQL`. These cursor attributes are used with implicit cursors that are automatically created by PL/SQL and for which you do not know the names. Therefore, you use `SQL` instead of the cursor name.

The `SQL%NOTFOUND` attribute is the opposite of `SQL%FOUND`. This attribute may be used as the exit condition in a loop. It is useful in `UPDATE` and `DELETE` statements when no rows are changed because exceptions are not returned in these cases.

You learn about explicit cursor attributes in the lesson titled “Using Explicit Cursors.”

## SQL Cursor Attributes for Implicit Cursors

Delete rows that have the specified employee ID from the `employees` table. Print the number of rows deleted.

Example:

```
DECLARE
  v_rows_deleted VARCHAR2(30);
  v_ordstat orders. order_status%TYPE := 0;

BEGIN
  DELETE FROM orders
  WHERE order_status = v_ordstat;
  v_rows_deleted := ( SQL%ROWCOUNT ||
                    ' row deleted.' );
  DBMS_OUTPUT.PUT_LINE ( v_rows_deleted );
END;
```

ORACLE

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

### SQL Cursor Attributes for Implicit Cursors (continued)

The example in the slide deletes a row with `order_status 0` from the `orders` table. Using the `SQL%ROWCOUNT` attribute, you can print the number of rows deleted.