**What is Cursor?**

* The oracle engine uses a work area for its internal processing in order to execute an SQL

Statement. This work area is private to SQL's operations and is called cursor.

* The data that is stored in the cursor is called the Active Data set.
* Conceptually, the size of the cursor in memory is the size required to hold the number of

Rows in the Active Data set.

* The actual size is however is determined by the oracle engine's built in memory management

Capabilities and the amount of RAM available.

* Oracle has a pre-defined area in main memory set aside, within which cursors are opened.
* Hence the cursor's size will be limited by the size of this predefined area.
* The values retrieved from a table are held in a cursor opened in memory by the oracle engine.
* This data is the transferred to the client machine via network.
* In order to hold this data, a cursor is opened at the client's end.
* If the number of rows returned by the oracle engine is more than the area available in the

Cursor opened on the client.

* The cursor data and the retrieved data are swapped between the OS's swap area and RAM.
* When a cursor is loaded with multiple rows via a query the oracle engine opens and maintains a row pointer.
* Depending on user’s requests to view data the row pointer will be relocated within the cursor’s active data set.
* Additionally oracle also maintains multiple cursor variables. The values held in these variables indicate the status of the processing being done by the cursor.

**Types of cursors**

* Cursors are classified depending on the circumstances under which they are opened.
* If the oracle engine opened a cursor for its internal processing it is known as an Implicit cursors.
* A cursor can also be opened for processing data through a PL/SQL block on demand. Such a user defined cursor is known as an Explicit cursor.

**General Cursor Attributes**

* When the oracle engine creates an Implicit or Explicit cursor, cursor control variables are also created to control the execution of the cursor.
* These are a set of four system variables, which keep track of the **current status** of a cursor.
* These cursor variables can be accessed and used in a PL/SQL code block.
* Both Implicit and Explicit cursors have four attributes.

|  |  |
| --- | --- |
| Attribute Name | Description |
| %ISOPEN | Returns true if cursor is open. False otherwise. |
| %FOUND | Returns true if record was fetched successfully, FALSE otherwise. |
| %NOTFOUND | Returns true if record was not fetched successfully, FALSE otherwise. |
| %ROWCOUNT | Returns number of records processed from the cursor. |

**Implicit Cursor**

* The oracle engine implicitly opens a cursor on the server to process each SQL statement, since the implicit cursor is opened and managed by the oracle engine internally, the function of reserving an area in memory, populating this area with appropriate data, processing the data in the memory area, releasing the memory area when the processing is complete is taken care of by the oracle engine.
* The resultant data is then passed to the client machine via the network.
* A cursor is then opened in memory on the client machine to hold the rows returned by the oracle engine.
* The number of rows held in the cursor on the client is managed by the client’s OS and its swap area.
* Implicit cursor attributes can be used to access information about the status of the last insert, update, delete or single row select statements.
* This can be done by preceding the implicit cursor attribute with the cursor name (i.e. SQL).
* The values of the cursor attributes always refer to the most recently executed SQL statement, wherever the statement appears.
* If an attribute value is to be saved for later use, it must be assigned to a Boolean memory variable.

**Implicit and Explicit Cursor Attributes**

|  |  |
| --- | --- |
| Attribute Name | Description |
| %ISOPEN | The oracle engine automatically opens and closes the SQL cursor after executing its associated select, insert, update or delete SQL statement has been processed in case of implicit cursors. Thus the SQL%ISOPEN attribute of an implicit cursor cannot be referenced outside of its SQL statements. As a result SQL%ISOPEN always evaluates to FALSE. |
| %FOUND | Evaluates to true, if an insert, update or delete affected one or more rows, or a single row select returned one or more rows. Otherwise, it evaluates to FALSE. The syntax for accessing this attribute is SQL%FOUND. |
| %NOTFOUND | IS the logical opposite of %FOUND. It evaluates to TRUE, if an insert, update or delete affected no rows, or a single row select returns no rows. Otherwise it evaluates to FALSE.  The syntax for accessing this attribute is SQL%NOTFOUND. |
| %ROWCOUNT | Returns the number of rows affected by an insert, update or delete, or select into statement. The syntax for accessing this attribute is SQL%ROWCOUNT. |

**Explicit Cursor**

* When individual records in a table have to be processed inside a PL/SQL code block a cursor is used. This cursor will be declared and mapped to an SQL query in the Declare Section of PL/SQL block and used within its Executable section.
* A cursor thus created and used is known as an Explicit Cursor.

**Explicit Cursor Management**

The steps involved in using an explicit cursor and manipulating data in its active set are:

* Declare a cursor mapped to SQL select statement that retrieves data for processing.
* Open the cursor.
* Fetch data from the cursor one row at a time into memory variables.
* Process the data held in the memory variables as required using a loop.
* Exit from the loop after processing is complete.

**Cursor Declaration**

* A cursor is defined in the declarative part of PL/SQL block. This is done by naming the cursor and mapping it to a query.
* When a cursor is declared, the oracle engine is informed that a cursor of the said name needs to be opened. The declaration is only intimation.
* There is no memory allocation at this point in time.
* The three commands used to control the cursor subsequently are open, fetch and close.

**The functionality of open, fetch and close commands**

Initialization of a cursor takes place via the open statement.

* Define a private SQL area named after the cursor name.
* Executes a query associated with the cursor.
* Retrieves table data and populates the named private SQL area in memory. i.e. create active data set.
* Sets the cursor row pointer in the Active Data Set to the first record.

Fetch statement is placed inside a loop…end loop construct, which causes the data to be fetched into the memory variables and processed until all the rows in the active data set are processed. The fetch loop then exits. The exiting of the fetch loop is user controlled.

After the fetch loop exits, the cursor must be closed with the close statement. This will release the memory occupied by the cursor and its active data set. A PL/SQL block is necessary to declare a cursor and create active data set. The cursor name is used to reference the active data set held within the cursor.