CURSORS

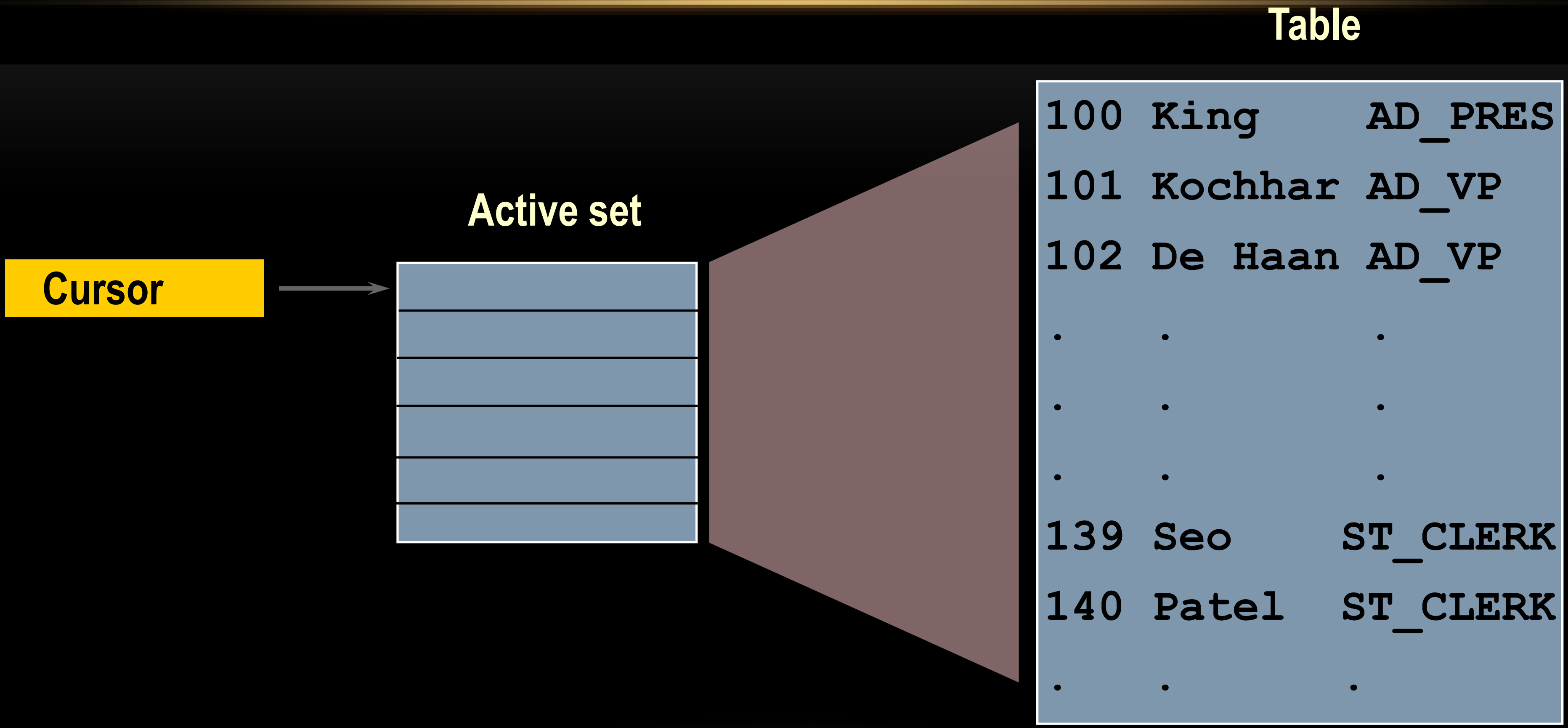
Ambili P K

ABOUT CURSORS

Every SQL statement executed by the Oracle Server has an individual cursor associated with it:

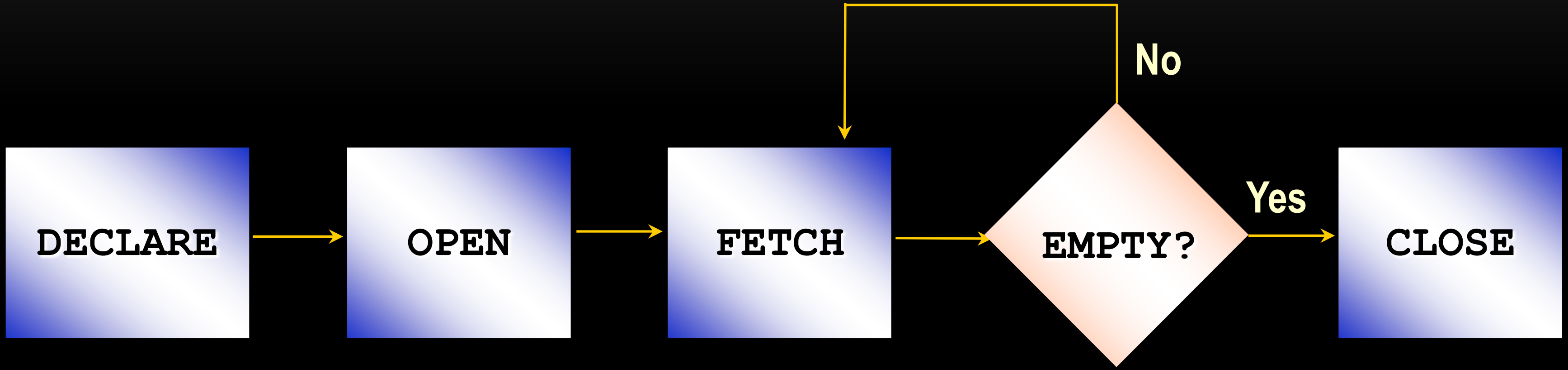
- Implicit cursors: Declared for all DML and PL/SQL `SELECT` statements
- Explicit cursors: Declared and named by the programmer

EXPLICIT CURSOR FUNCTIONS



100	King	AD_PRES
101	Kochhar	AD_VP
102	De Haan	AD_VP
.	.	.
.	.	.
.	.	.
139	Seo	ST_CLERK
140	Patel	ST_CLERK
.	.	.

CONTROLLING EXPLICIT CURSORS



- Create a named SQL area

- Identify the active set

- Load the current row into variables

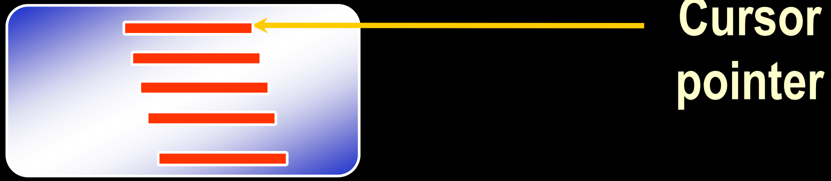
- Test for existing rows
- Return to **FETCH** if rows are found

- Release the active set

CONTROLLING EXPLICIT CURSORS

1. Open the cursor
2. Fetch a row
3. Close the Cursor

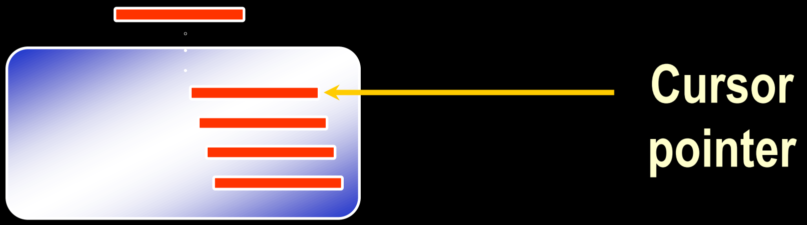
1. Open the cursor.



CONTROLLING EXPLICIT CURSORS

1. Open the cursor
2. Fetch a row
3. Close the Cursor

2. Fetch a row using the cursor.

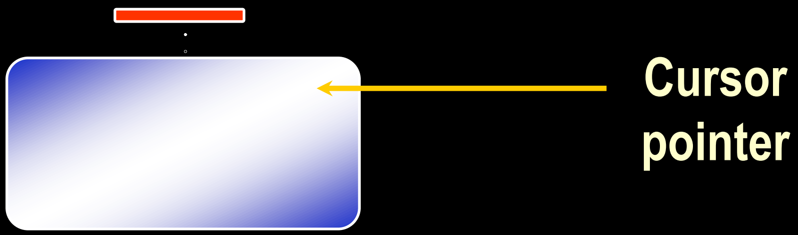


Continue until empty.

CONTROLLING EXPLICIT CURSORS

1. Open the cursor
2. Fetch a row
3. Close the Cursor

3. Close the cursor.



DECLARING THE CURSOR

Syntax:

```
CURSOR cursor_name IS
     select_statement;
```

- Do not include the `INTO` clause in the cursor declaration.
- If processing rows in a specific sequence is required, use the `ORDER BY` clause in the query.

DECLARING THE CURSOR

Example:

```
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, last_name
    FROM   employees;

  CURSOR dept_cursor IS
    SELECT *
    FROM   departments
    WHERE  location_id = 170;
BEGIN
  ...
```

OPENING THE CURSOR

Syntax:

```
OPEN    cursor_name;
```

- Open the cursor to execute the query and identify the active set.
- If the query returns no rows, no exception is raised.
- Use cursor attributes to test the outcome after a fetch.

FETCHING DATA FROM THE CURSOR

Syntax:

```
FETCH cursor_name INTO  [variable1,
variable2, ...]
                    | record_name];
```

- Retrieve the current row values into variables.
- Include the same number of variables.
- Match each variable to correspond to the columns positionally.
- Test to see whether the cursor contains rows.

FETCHING DATA FROM THE CURSOR

Example:

```
LOOP
  FETCH emp_cursor INTO
v_empno,v_ename;
  EXIT WHEN ...;
  ...
    -- Process the retrieved data
  …
END LOOP;
```

CLOSING THE CURSOR

Syntax:

```
CLOSE     cursor_name;
```

- Close the cursor after completing the processing of the rows.
- Reopen the cursor, if required.
- Do not attempt to fetch data from a cursor after it has been closed.

EXPLICIT CURSOR ATTRIBUTES

Obtain status information about a cursor.

Attribute	Type	Description
`%ISOPEN`	Boolean	Evaluates to `TRUE` if the cursor is open
`%NOTFOUND`	Boolean	Evaluates to `TRUE` if the most recent fetch does not return a row
`%FOUND`	Boolean	Evaluates to `TRUE` if the most recent fetch returns a row; complement of `%NOTFOUND`
`%ROWCOUNT`	Number	Evaluates to the total number of rows returned so far

THE %ISOPEN ATTRIBUTE

- Fetch rows only when the cursor is open.
- Use the %ISOPEN cursor attribute before performing a fetch to test whether the cursor is open.

Example:

```
IF NOT emp_cursor%ISOPEN THEN
   OPEN emp_cursor;
END IF;
LOOP
  FETCH emp_cursor...
```

CONTROLLING MULTIPLE FETCHES

- Process several rows from an explicit cursor using a loop.
- Fetch a row with each iteration.
- Use explicit cursor attributes to test the success of each fetch.

%NOTFOUND AND %ROWCOUNT ATTRIBUTES

- Use the %ROWCOUNT cursor attribute to retrieve an exact number of rows.
- Use the %NOTFOUND cursor attribute to determine when to exit the loop.

EXAMPLE

```
DECLARE
      v_empnoemployees.employee_id%TYPE;
      v_enameemployees.last_name%TYPE;
      CURSOR emp_cursor IS
        SELECT employee_id, last_name
        FROM   employees;
    BEGIN
      OPEN emp_cursor;
      LOOP
        FETCH emp_cursor INTO v_empno, v_ename;
        EXIT WHEN emp_cursor%ROWCOUNT > 10 OR
                                  emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE (TO_CHAR(v_empno)
                                      ||' '|| v_ename);
      END LOOP;
      CLOSE emp_cursor;
END ;
```

SUMMARY

In this lesson you should have learned to:

- Distinguish cursor types:
  - Implicit cursors: used for all `DML` statements and single-row queries
  - Explicit cursors: used for queries of zero, one, or more rows
- Manipulate explicit cursors
- Evaluate the cursor status by using cursor attributes