# PL/SQL Cursors

- Introduction to PL/SQL Cursors

- Types of cursors

- How to work with Cursors

- Example of using cursors in Stored Procedures

- A PL/SQL cursor is a pointer that points to the result set of an SQL query against database tables.

- You can name a cursor so that it could be referred in a program to fetch and process the rows returned by the SQL statement, one at a time.

Types

- ✓ Implicit Cursors
- ✓ Explicit Cursors

# Implicit Cursors

- Implicit cursors are automatically created by Oracle whenever an SQL statement is executed

- Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement.

| Attribute | Description |
|---|---|
| %FOUND | Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE. |
| %NOTFOUND | The logical opposite of %FOUND. |
| %ISOPEN | Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement. |
| %ROWCOUNT | Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement. |

# Implicit Cursors – Example

## Product

| Product_Id | Price | Pdt_Type |
|---|---|---|
| 300 | 4000 | Electronics |
| 301 | 2000 | Books |

## Example

```
DECLARE
total_rows number(2);
BEGIN
UPDATE Product SET Price = Price + 500;
IF sql%notfound THEN
dbms_output.put_line('no products selected');
ELSIF sql%found THEN
total_rows := sql%rowcount;
dbms_output.put_line( total_rows || ' products selected ');
END IF;
END;
/
```

2 products selected

PL/SQL procedure successfully completed.

- Explicit cursors are programmer-defined cursors for gaining more control over the context area.

## How to work with explicit cursor

1) Declare a cursor

   CURSOR cursor_name IS SELECT_statement;

2) Open the cursor by using the OPEN statement

   OPEN cursor_name;

3) Use the FETCH statement to retrieve the next row

   FETCH cursor_name INTO variables list;

4) Call the CLOSE statement

   CLOSE cursor_name;

# Explicit Cursors – Example

## Product

| Product _Id | Price | Pdt_Type |
|---|---|---|
| 300 | 4000 | Electronics |
| 301 | 2000 | Books |

300  4000  Electronics
301  2000  Books

PL/SQL procedure successfully completed.

## Example

```
SET SERVEROUTPUT ON
DECLARE
p_id Product.Product_Id%type;
price Product.Price%type;
ptype Product.Pdt_Type%type;
CURSOR p_product is
SELECT Product_Id, Price, Pdt_Type FROM Product;
BEGIN
OPEN p_product;
LOOP
FETCH p_product into p_id, price, ptype;
EXIT WHEN p_product%notfound;
dbms_output.put_line(p_id || ' ' || price || ' ' || ptype);
END LOOP;
CLOSE p_product;
END; /
```

# PL/SQL Parameterized Cursor

## Product

| Product_Id | Price | Pdt_Type |
|---|---|---|
| 300 | 4000 | Electronics |
| 301 | 2000 | Books |
| 302 | 8000 | Accessories |
| 303 | 3000 | Kitchen |

## Example

```
SET SERVEROUTPUT ON
DECLARE
cursor c(no number) is
    select * from Product where Product_Id = no;
Tmp Product%rowtype;
BEGIN
OPEN c(301);
FOR tmp IN c(301) LOOP
FETCH c INTO tmp;
dbms_output.put_line('Product Id: '||tmp.Product_Id);
dbms_output.put_line('Price: '||tmp.Price);
dbms_output.put_line('Type: '||tmp.Pdt_Type);
END LOOP;
CLOSE c;
END;
/
```

# THANKS

eBOX

BY AMPHISOFT