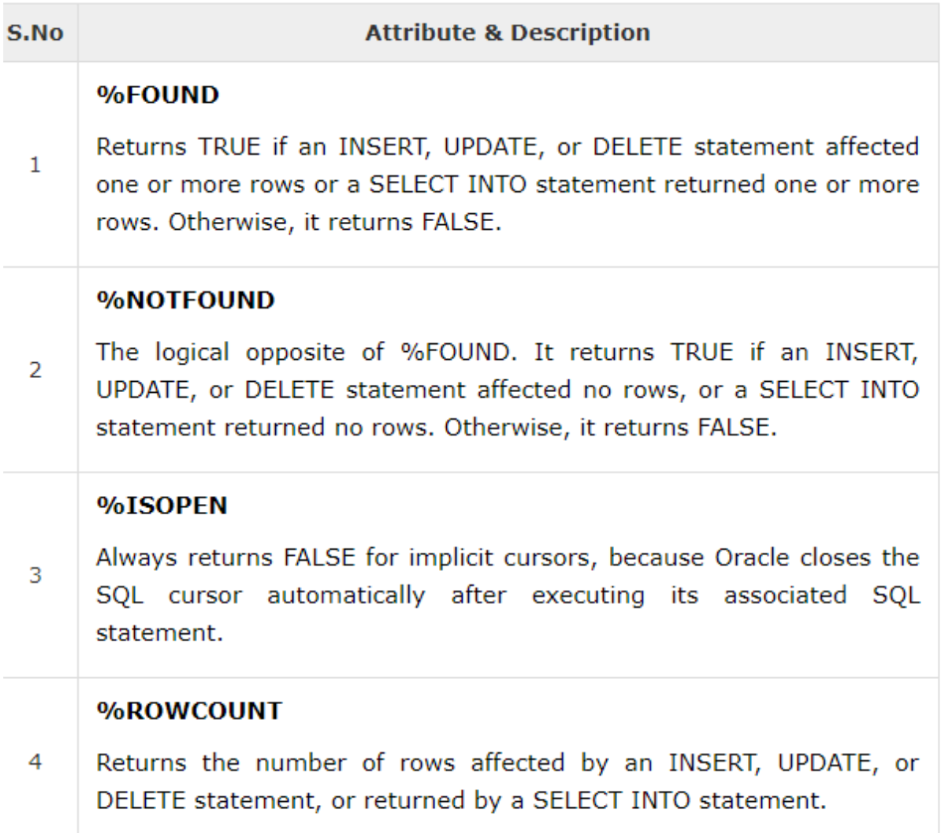
A **cursor** is a control structure in SQL and PL/SQL that allows you to process individual rows returned by a query. Imagine a pointer or a bookmark that points to the current row within a result set (the set of rows returned by a SELECT statement).

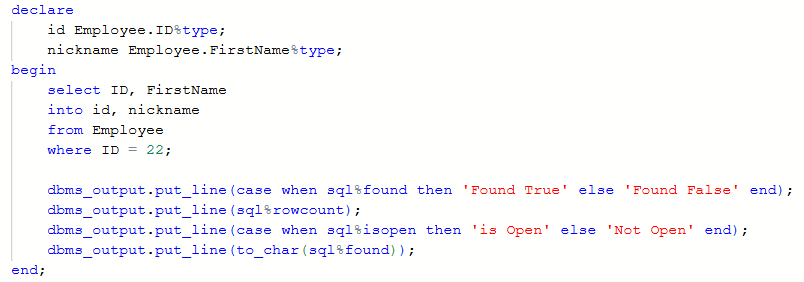
**Why Do We Need Cursors?**

Normally, when you execute a SELECT statement in SQL (outside of PL/SQL), the database immediately returns all matching rows to your client application. However, within a PL/SQL block, you often need to perform operations row by row, especially if you want to:

* **Process each row individually:** Apply specific logic to each record.
* **Update or delete specific rows** based on some calculations or conditions.
* **Fetch data incrementally:** Retrieve data one row at a time, which can be memory-efficient for large result sets.

This row-by-row processing is exactly what a cursor enables.

******

****

**A black text on a white background

AI-generated content may be incorrect.**

***How Cursors Work:***

**🡪 Declare the Cursor:** You define the cursor by giving it a name and associating it with a SELECT statement. This doesn't execute the query yet; it just sets up the blueprint.

A close-up of a code

AI-generated content may be incorrect.

**🡪** **Open the Cursor:** When you open the cursor, the database executes the SELECT statement and identifies the set of rows that match the query criteria. These rows are then available for fetching.



**🡪Fetch Data:** You fetch rows one by one from the opened cursor into PL/SQL variables. Each fetch advances the cursor to the next row in the result set.



**🡪 Close the Cursor:** Once you're done processing the rows, you close the cursor to release the resources associated with it.



***Types of Cursors:*1. Implicit Cursors:**

* These are automatically managed by Oracle for all DML statements (INSERT, UPDATE, DELETE) and single-row SELECT INTO statements.
* You don't explicitly declare, open, fetch, or close them. Oracle handles it behind the scenes.
* You can check their status using cursor attributes like **SQL%ROWCOUNT** (number of rows affected), **SQL%FOUND** (true if at least one row affected), **SQL%NOTFOUND** (true if no rows affected), **SQL%ISOPEN** (always false for implicit cursors after execution because they are always automatically closed by Oracle).

**2. Explicit Cursors:**

* These are cursors that you explicitly declare, open, fetch, and close in your PL/SQL code.
* They are used when your SELECT statement is expected to return multiple rows and you need to process them individually.
* They provide fine-grained control over row processing.

**Cursor For Loop:**

A Cursor FOR Loop is a special construct in PL/SQL that provides a highly simplified and efficient way to process every row returned by a SELECT statement. It's a syntactic sugar (a shorthand) for manually declared explicit cursors.

Instead of needing to explicitly:

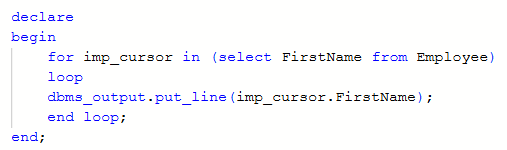
1. **Declare** a cursor.
2. **Open** the cursor.
3. **Fetch** rows one by one into local variables.
4. **Loop** through the fetched rows, checking for NOTFOUND.
5. **Close** the cursor.

...the Cursor FOR Loop handles all these steps **automatically** for you, making your code cleaner, safer, and less prone to common cursor-related errors (like forgetting to close a cursor).

***Syntax:***

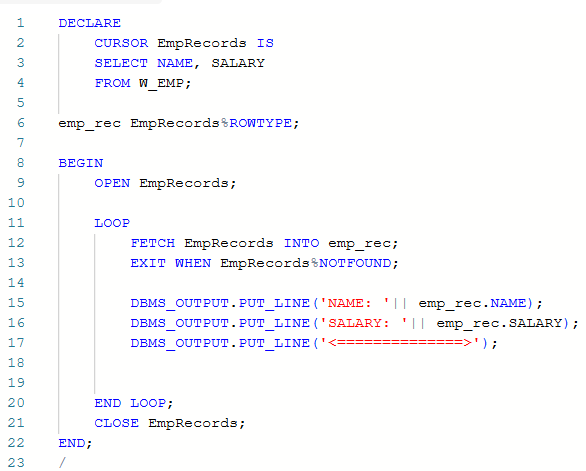
A screenshot of a computer program

AI-generated content may be incorrect.

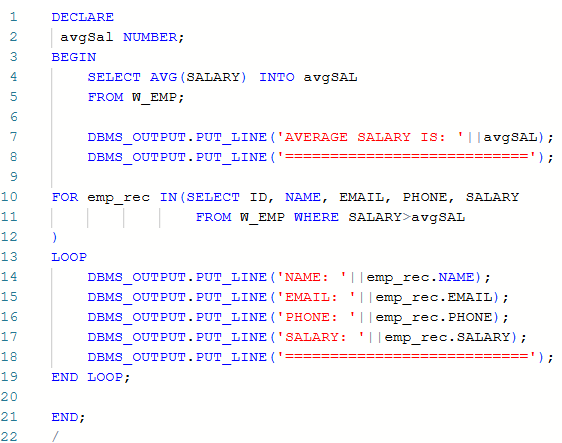
****

**A group of names on a white background

AI-generated content may be incorrect.**

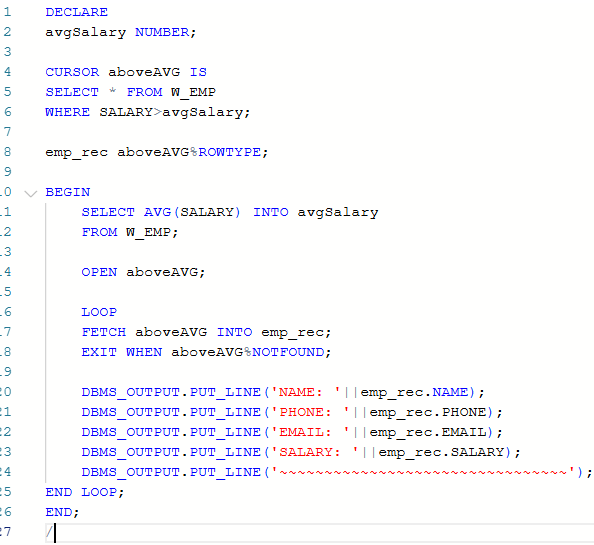
**EXPLICIT CURSOR EXAMPLES:**A screenshot of a computer code

AI-generated content may be incorrect.

***CURSOR FOR LOOP EXAMPLE:*** A white text with black text

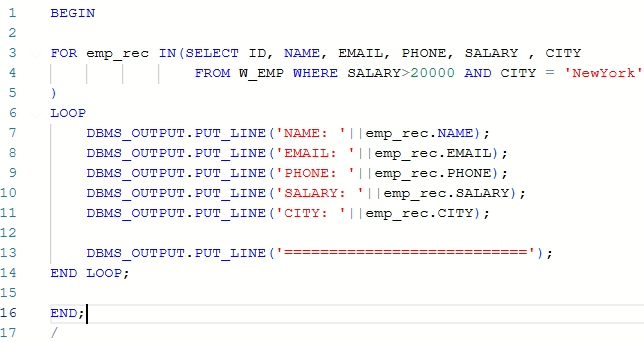
AI-generated content may be incorrect.

***SAME CURSOR USING EXPLICIT CURSOR:***

 A screenshot of a computer

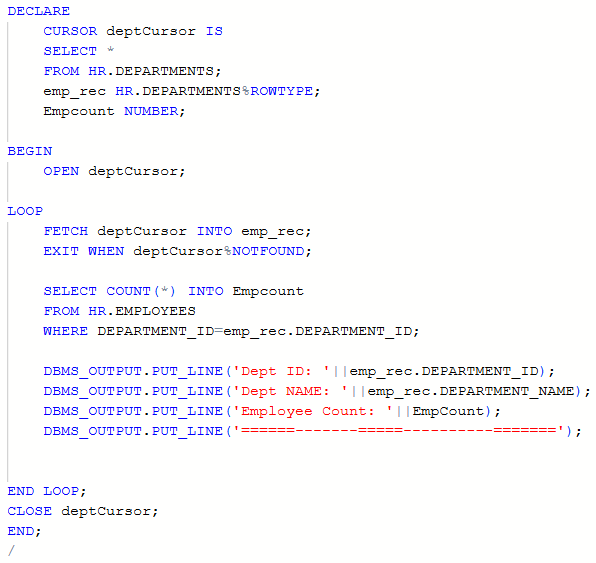
AI-generated content may be incorrect.

***Find the Employee with salary greater than 20000 and live in Newyork (Using Cursor For Loop):***

 A screenshot of a computer

AI-generated content may be incorrect.

***Write a cursor that selects all the departments from the department table and displays their department\_id, department\_name, and the total number of employees in each department.***

****** ***A screenshot of a computer

AI-generated content may be incorrect.***

***Passing Parameters in Cursor:***

