

## Implicit Cursors:

### 1. Query 1: Implicit Cursor to Retrieve All Rows from a Table

- Question:** How can we use an implicit cursor to fetch all rows from the "employees" table?

```
BEGIN
```

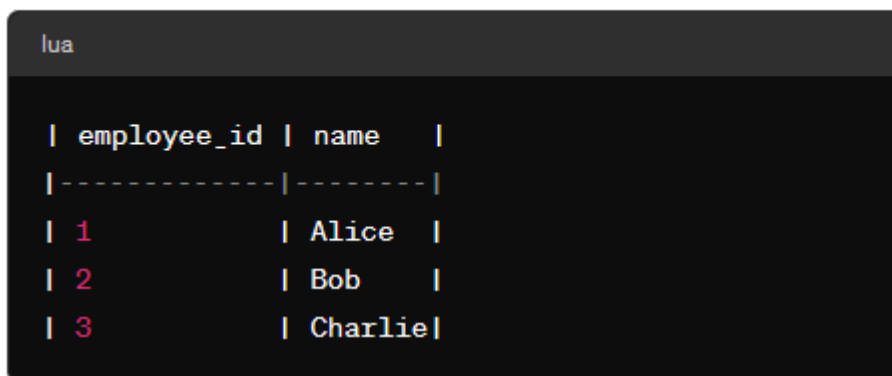
```
FOR emp_record IN (SELECT * FROM employees) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id || ', Name: ' ||  
emp_record.name);
```

```
END LOOP;
```

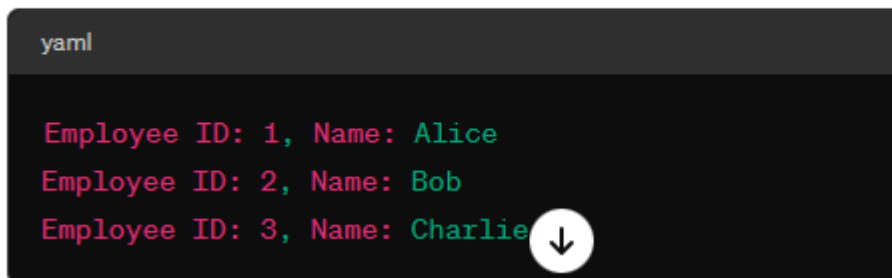
```
END;
```

**Table: `employees`**

A screenshot of a database table named 'employees'. The table has two columns: 'employee\_id' and 'name'. It contains three rows of data: (1, Alice), (2, Bob), and (3, Charlie). The table is displayed in a dark-themed interface with a light gray header.

employee_id	name
1	Alice
2	Bob
3	Charlie

**Expected Output:**

A screenshot showing the output of the SQL query. It displays three lines of text: 'Employee ID: 1, Name: Alice', 'Employee ID: 2, Name: Bob', and 'Employee ID: 3, Name: Charlie'. The text is in a dark-themed interface with a light gray header. A white circular button with a downward arrow is located at the bottom right of the output area.

```
Employee ID: 1, Name: Alice  
Employee ID: 2, Name: Bob  
Employee ID: 3, Name: Charlie
```

### Query 2: Implicit Cursor with Order By Clause

- Question:** How do we use an implicit cursor to fetch employees sorted by their salaries in descending order?

```
BEGIN
```

```
FOR emp_record IN (SELECT * FROM employees ORDER BY salary DESC) LOOP
```

```
    DBMS_OUTPUT.PUT_LINE('Name: ' || emp_record.name || ', Salary: ' ||  
emp_record.salary);
```

```
END LOOP;
```

END;

Table: `employees`

lua	
name	salary
Alice	50000
Bob	60000
Charlie	45000

Expected Output:

```
yaml
Name: Bob, Salary: 60000
Name: Alice, Salary: 50000
Name: Charlie, Salary: 45000
```

### Query 3: Implicit Cursor with WHERE Clause

- **Question:** How can we use an implicit cursor to fetch employees whose salary is greater than 50000?
- **Answer**
- BEGIN
- FOR emp\_record IN (SELECT \* FROM employees WHERE salary > 50000)
- LOOP
- DBMS\_OUTPUT.PUT\_LINE('Name: ' || emp\_record.name || ', Salary: ' || emp\_record.salary);
- END LOOP;
- END;

Table: `employees`

lua	
name	salary
Bob	60000

Expected Output:

```
yaml
Name: Bob, Salary: 60000
```

#### Query 4: Implicit Cursor with Counting Rows

- **Question:** How can we use an implicit cursor to count the number of employees?

DECLARE

total\_employees NUMBER := 0;

BEGIN

FOR emp\_record IN (SELECT \* FROM employees) LOOP

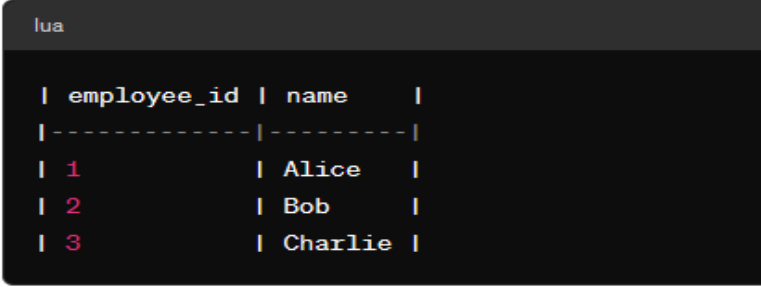
total\_employees := total\_employees + 1;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('Total Employees: ' || total\_employees);

END;

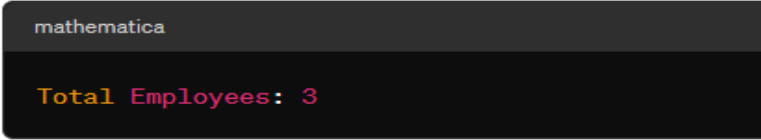
Table: `employees`



A screenshot of a terminal window with a dark background. The title bar says 'lua'. Inside the terminal, a table is displayed with two columns: 'employee\_id' and 'name'. The table contains three rows of data: (1, Alice), (2, Bob), and (3, Charlie). The text is color-coded, with IDs in red and names in white.

employee_id	name
1	Alice
2	Bob
3	Charlie

Expected Output:



A screenshot of a terminal window with a dark background. The title bar says 'mathematica'. Inside the terminal, the text 'Total Employees: 3' is displayed in a yellow and red color scheme.

Total Employees: 3

#### Query 6: Explicit Cursor Declaration and Fetch

- **Question:** How do we declare an explicit cursor to fetch employee records from the "employees" table?

DECLARE

CURSOR emp\_cursor IS

SELECT \* FROM employees;

emp\_rec employees%ROWTYPE;

BEGIN

OPEN emp\_cursor;

FETCH emp\_cursor INTO emp\_rec;

```
DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.employee_id || ', Name: ' ||  
emp_rec.name);
```

```
CLOSE emp_cursor;
```

```
END;
```

### Table: `employees`

lua

employee_id	name
1	Alice

### Expected Output:

yaml

Employee ID: 1, Name: Alice