CURSOR

Cursor declarations must appear before handler declarations. Variable and condition declarations must appear before cursor or handler declarations.

This statement declares a cursor and associates it with a SELECT statement that retrieves the rows to be traversed by the cursor.

```
DECLARE cursor_name CURSOR FOR select_statement
```

This statement opens a previously declared cursor.

```
OPEN cursor_name
```

This statement fetches the next row for the SELECT statement associated with the specified cursor (which must be open), and advances the cursor pointer. If a row exists, the fetched columns are stored in the named variables. The number of columns retrieved by the SELECT statement must match the number of output variables specified in the FETCH statement

```
FETCH [[NEXT] FROM] cursor_name INTO var_name [, var_name] ...
```

If no more rows are available, a No Data condition.

This statement closes a previously opened cursor.

```
CLOSE cursor_name
```

What is a Cursor in PostgreSQL?

- A **cursor in PostgreSQL** is a database object that enables traversal over the result set of a query. It acts as a **pointer** that allows us to **fetch** rows sequentially.
- **PostgreSQL cursors** are particularly useful when working with large tables containing millions of records where traditional **SELECT** operations may lead to performance issues or even out-of-memory errors.

Syntax:

```
DO $$
DECLARE
   rec RECORD;
   cur CURSOR FOR SELECT id, name FROM employees;
BEGIN
   OPEN cur;
   LOOP
      FETCH cur INTO rec;
      EXIT WHEN NOT FOUND;
      RAISE NOTICE 'ID: %, Name: %', rec.id, rec.name;
   END LOOP;
   CLOSE cur;
END $$;
```