

Actividad portafolio n°3: 'Manipulación de datos y transaccionalidad'

Nicolás Jofré Andrade - Desarrollo Python Full stack

Considerando la instrucciones del portafolio solicitado:

1. Preparación de entorno:

- a. Crear base de datos llamar 'EjercicioDB' con una tabla llamada empleados y que contenga los campos: ID_Empleado, Nombre, Apellido y Salario.

2. DML:

- a. Actualizar la información de un empleado (ID_Empleado = 1) para cambiar el salario.
- b. Borrar la información de un empleado específico (ID_Empleado = 2)
- c. Insertar nueva información de un empleado
- d. Utilización de secuencias para asignar identificadores (suponiendo que hay una secuencia llamada 'seq_empleados')
- e. Insertar datos manteniendo la integridad referencial (suponiendo que hay una tabla 'Departamentos' relacionada)

3. Restricciones en una tabla:

- a. Deben haber restricciones en la tabla 'Empleados' de llaves primarias y llaves foráneas como 'ID_Departamento'.

Desarrollo:

1.a

```
Query  Query History
1  CREATE TABLE Empleados (
2      Id_Empleado INT PRIMARY KEY,
3      Nombre VARCHAR(50),
4      Apellido VARCHAR(50),
5      Salario INT
6  );
7
8  -- 2. Inserción en la tabla 'Empleados'
9  INSERT INTO Empleados (Id_Empleado, Nombre, Apellido, Salario) VALUES
10 (1, 'Matias', 'Fernandez', 1000000),
11 (2, 'Carla', 'Sepulveda', 5000000),
12 (4, 'Nicolás', 'Jofré', 450000);

Data Output  Messages  Notifications
INSERT 0 3

Query returned successfully in 62 msec.
```

2.a

```
Query  Query History
13
14 -- 2.a Actualizar la información de un empleado para cambiar el salario.
15 UPDATE Empleados SET salario = 480000 WHERE Id_Empleado = 1;
16
17
18
19
20
21
22
23
24
25
26
27
28

Data Output  Messages  Notifications
UPDATE 1

Query returned successfully in 110 msec.
```

2.b

```
Query  Query History
14  -- 2.b Borrar la información de un usuario en específico
15  DELETE FROM Empleados WHERE ID_Empleado = 4;
16
17
18
19
20
21
22
23
24
25
26
27
28
29
```

Data Output Messages Notifications

```
DELETE 1

Query returned successfully in 49 msec.
```

2.c

```
Query  Query History
17  -- 2.c Insertar nueva información de un empleado
18  INSERT INTO Empleados (ID_Empleado, Nombre, Apellido, Salario) VALUES
19  (5, 'Bastian', 'Landksron', 1000000),
20  (6, 'Katherine', 'Ani', 2000000)
21
22
23
24
25
26
27
28  |
29
30
31
32
```

```
Data Output  Messages  Notifications
INSERT 0 2

Query returned successfully in 51 msec.
```

2.d, e

```
Query  Query History
22  -- 2.d Utilizar secuencias para tabla empleados
23  CREATE SEQUENCE seq_empleados_e START WITH 10 INCREMENT BY 1;
24
25  -- 2.e Inserción con secuencia
26  INSERT INTO Empleados (Id_Empleado, Nombre, Apellido, Salario) VALUES
27  (NEXTVAL('seq_empleados_e'), 'Tomás', 'Medina', 200000),
28  (NEXTVAL('seq_empleados_e'), 'Angelina', 'Troncoso', 929000),
29  (NEXTVAL('seq_empleados_e'), 'Fernanda', 'Clavero', 1200000),
30  (NEXTVAL('seq_empleados_e'), 'Camilo', 'Clavijo', 8500000);
31
32
33
34
35
36
37
Data Output  Messages  Notifications
INSERT 0 4
Query returned successfully in 50 msec.
```

3.

```
Query  Query History
32  -- 3. Restricciones de tablas con las llaves de cada tabla
33  -- La restricción se hace al referenciar las llaves primarias entre Empleados y Departamento
34  CREATE TABLE Departamento_d (
35      Id_Departamento INT PRIMARY KEY,
36      Nombre VARCHAR(50),
37      Id_Empleado INT,
38      FOREIGN KEY (ID_empleado) REFERENCES Empleados(ID_empleado)
39  );
40
41
42
43
44
45
46
47
Data Output  Messages  Notifications
CREATE TABLE
Query returned successfully in 55 msec.
```