**Función: Crea una función en PL/SQL que calcule y devuelva la edad promedio de los usuarios en la tabla "Usuarios"**

CREATE OR REPLACE FUNCTION calcular\_edad\_promedio RETURN NUMBER IS

edad\_promedio NUMBER;

BEGIN

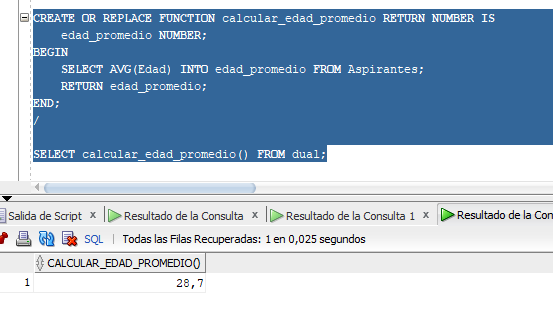
SELECT AVG(Edad) INTO edad\_promedio FROM Aspirantes;

RETURN edad\_promedio;

END;

/

SELECT calcular\_edad\_promedio() FROM dual;



En esta función, se utiliza la cláusula **SELECT AVG(Edad) INTO edad\_promedio FROM Aspirantes** para calcular la edad promedio de los aspirantes en la tabla "Aspirantes" y almacenar el resultado en la variable **edad\_promedio**. Luego, se utiliza la instrucción **RETURN edad\_promedio** para devolver el valor de la edad promedio.

**2. Procedimiento almacenado: Escribe un procedimiento almacenado en PL/SQL que reciba el ID de un usuario y muestre su nombre y correo electrónico.**

CREATE OR REPLACE PROCEDURE obtener\_nombre\_y\_correo (p\_id\_aspirante IN NUMBER) AS

v\_nombre VARCHAR2(50);

v\_correo VARCHAR2(100);

BEGIN

SELECT Nombre, Email INTO v\_nombre, v\_correo FROM Aspirantes WHERE IDAspirante = p\_id\_aspirante;

DBMS\_OUTPUT.PUT\_LINE('Nombre: ' || v\_nombre);

DBMS\_OUTPUT.PUT\_LINE('Correo: ' || v\_correo);

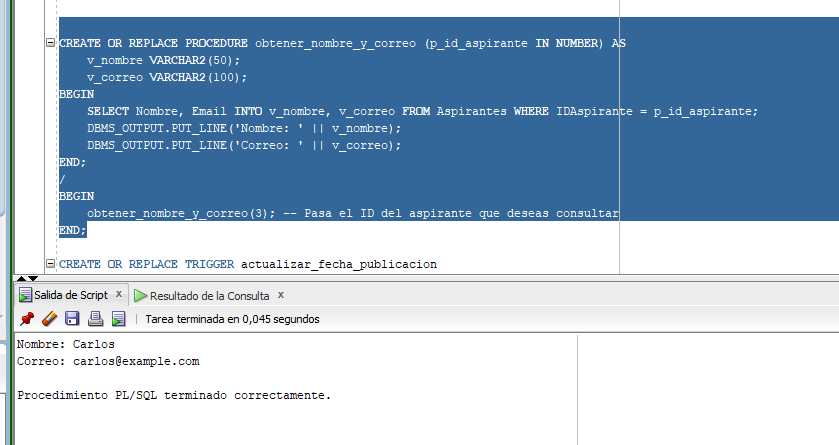
END;

/

BEGIN

obtener\_nombre\_y\_correo(3); -- Pasa el ID del aspirante que deseas consultar

END;



En este procedimiento almacenado, se utiliza la cláusula **SELECT Nombre, Email INTO v\_nombre, v\_correo FROM Aspirantes WHERE IDAspirante = p\_id\_aspirante** para obtener el nombre y correo electrónico del aspirante cuyo ID corresponde al parámetro **p\_id\_aspirante**. Luego, se utiliza **DBMS\_OUTPUT.PUT\_LINE** para mostrar el nombre y el correo electrónico en la salida.

**3. Disparador: Crea un disparador en PL/SQL que se active cada vez que se inserte un nuevo registro en la tabla "Empleos" y actualice automáticamente la fecha de publicación a la fecha actual.**

CREATE OR REPLACE TRIGGER actualizar\_fecha\_publicacion

BEFORE INSERT ON OfertasDeEmpleo

FOR EACH ROW

BEGIN

:NEW.FechaPublicacion := SYSDATE;

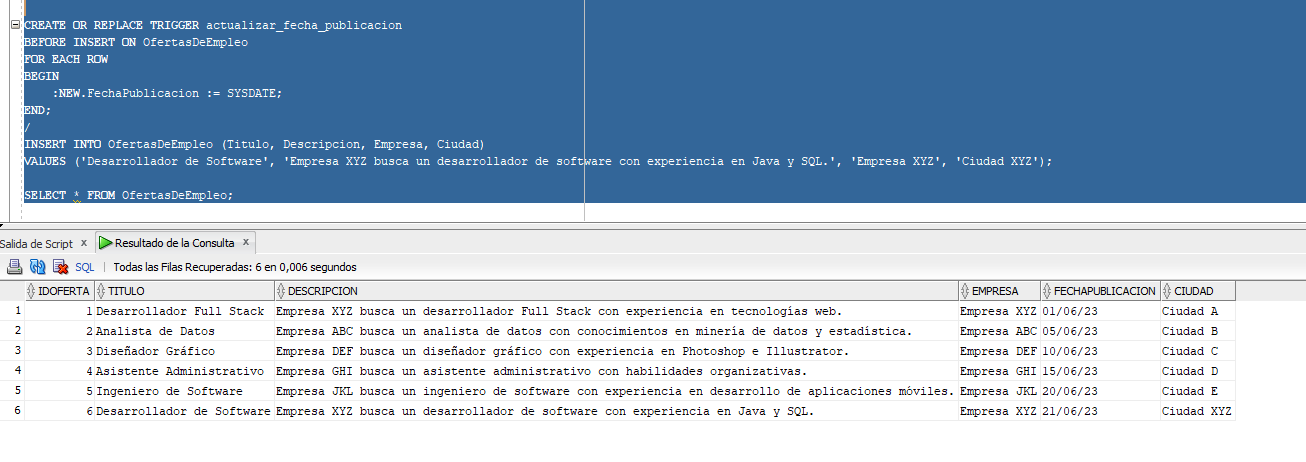
END;

/

INSERT INTO OfertasDeEmpleo (Titulo, Descripcion, Empresa, Ciudad)

VALUES ('Desarrollador de Software', 'Empresa XYZ busca un desarrollador de software con experiencia en Java y SQL.', 'Empresa XYZ', 'Ciudad XYZ');

SELECT \* FROM OfertasDeEmpleo;



En este disparador, utilizamos la cláusula **BEFORE INSERT ON OfertasDeEmpleo** para indicar que se activará antes de que se inserte un nuevo registro en la tabla "OfertasDeEmpleo". La cláusula **FOR EACH ROW** especifica que el disparador se ejecutará una vez por cada fila afectada por la inserción.

En el bloque del disparador, asignamos la fecha actual a la columna **FechaPublicacion** del nuevo registro que se está insertando, utilizando **:NEW.FechaPublicacion := SYSDATE;**. **:NEW** hace referencia a la fila que se está insertando.

De esta manera, cada vez que se inserte un nuevo registro en la tabla "OfertasDeEmpleo", la columna **FechaPublicacion** se actualizará automáticamente con la fecha actual.

**4. Procedimiento con cursor: Escribe un procedimiento almacenado en PL/SQL que utilice un cursor para mostrar todos los usuarios que tienen una edad mayor a 30 años.**

CREATE OR REPLACE PROCEDURE mostrar\_aspirantes\_mayor\_30 AS

CURSOR c\_aspirantes IS

SELECT Nombre, Apellido, Edad

FROM Aspirantes

WHERE Edad > 30;

v\_nombre Aspirantes.Nombre%TYPE;

v\_apellido Aspirantes.Apellido%TYPE;

v\_edad Aspirantes.Edad%TYPE;

BEGIN

OPEN c\_aspirantes;

LOOP

FETCH c\_aspirantes INTO v\_nombre, v\_apellido, v\_edad;

EXIT WHEN c\_aspirantes%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Nombre: ' || v\_nombre || ', Apellido: ' || v\_apellido || ', Edad: ' || v\_edad);

END LOOP;

CLOSE c\_aspirantes;

END;

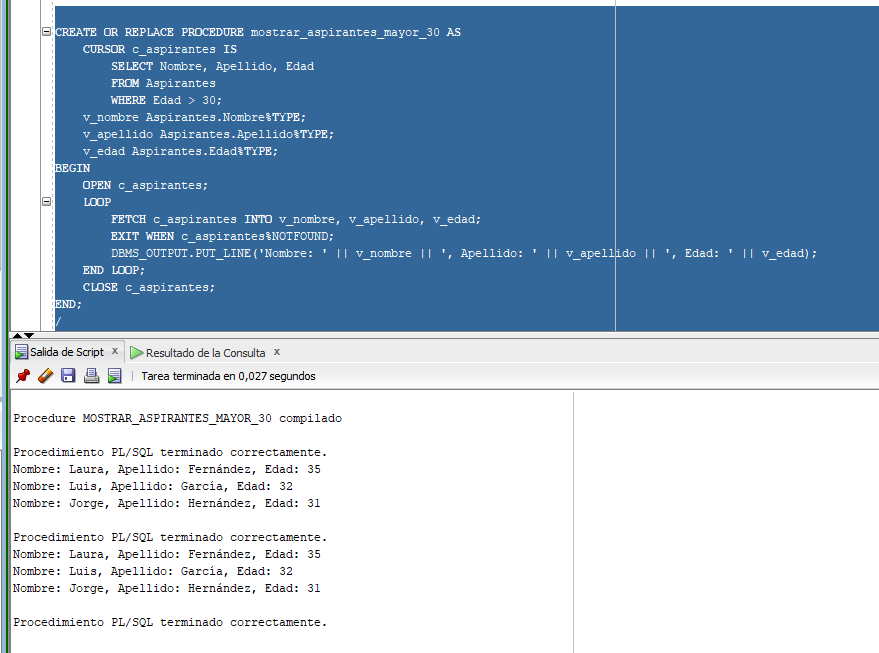
/

BEGIN

mostrar\_aspirantes\_mayor\_30;

END;

/



En este procedimiento, definimos un cursor llamado **c\_aspirantes** que selecciona los nombres, apellidos y edades de los aspirantes que tienen una edad mayor a 30 años.

Dentro del bloque del procedimiento, abrimos el cursor con la sentencia **OPEN c\_aspirantes** y luego utilizamos un bucle **LOOP** para recorrer los resultados del cursor. En cada iteración del bucle, almacenamos los valores de nombre, apellido y edad en variables **v\_nombre**, **v\_apellido** y **v\_edad**, respectivamente, utilizando la sentencia **FETCH c\_aspirantes INTO v\_nombre, v\_apellido, v\_edad**.

Luego, utilizamos **DBMS\_OUTPUT.PUT\_LINE** para mostrar en la salida estándar los datos de cada aspirante que cumple con la condición de edad mayor a 30 años.

Finalmente, cerramos el cursor con la sentencia **CLOSE c\_aspirantes**.

**5. Procedimiento con bucle: Define un procedimiento almacenado en PL/SQL que utilice un bucle para mostrar todos los empleos de la tabla "Empleos" junto con la cantidad de usuarios inscritos en cada uno.**

CREATE OR REPLACE PROCEDURE mostrar\_ofertas\_con\_inscritos AS

v\_idoferta OfertasDeEmpleo.IDOferta%TYPE;

v\_titulo OfertasDeEmpleo.Titulo%TYPE;

v\_cantidad NUMBER;

BEGIN

FOR oferta IN (SELECT IDOferta, Titulo FROM OfertasDeEmpleo) LOOP

v\_idoferta := oferta.IDOferta;

v\_titulo := oferta.Titulo;

SELECT COUNT(\*) INTO v\_cantidad

FROM Candidaturas

WHERE IDOferta = v\_idoferta;

DBMS\_OUTPUT.PUT\_LINE('Oferta: ' || v\_titulo || ', Inscritos: ' || v\_cantidad);

END LOOP;

END;

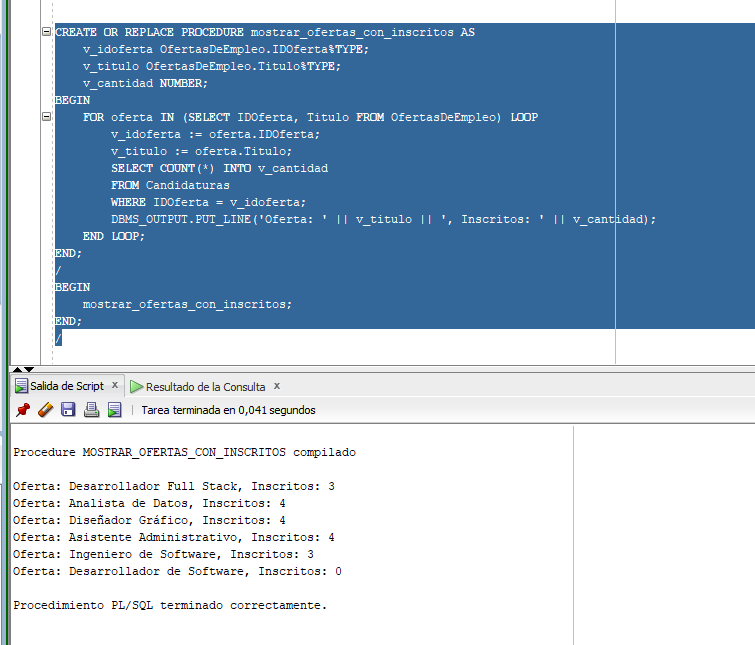
/

BEGIN

mostrar\_ofertas\_con\_inscritos;

END;

/



En este procedimiento, utilizamos un bucle **FOR** para iterar a través de las filas de la tabla "OfertasDeEmpleo". En cada iteración del bucle, almacenamos el ID de la oferta y el título en las variables **v\_idoferta** y **v\_titulo**, respectivamente.

Luego, ejecutamos una consulta **SELECT COUNT(\*)** para obtener la cantidad de usuarios inscritos en cada oferta. Utilizamos la cláusula **WHERE** para filtrar las candidaturas por el ID de la oferta actual en la iteración del bucle.

Guardamos el resultado de la consulta en la variable **v\_cantidad** utilizando la sentencia **INTO**.

Finalmente, utilizamos **DBMS\_OUTPUT.PUT\_LINE** para mostrar en la salida estándar el título de la oferta y la cantidad de usuarios inscritos.

Se llamará al procedimiento **mostrar\_ofertas\_con\_inscritos** y mostrará en la salida estándar el título de cada oferta de empleo y la cantidad de usuarios inscritos en cada una.