

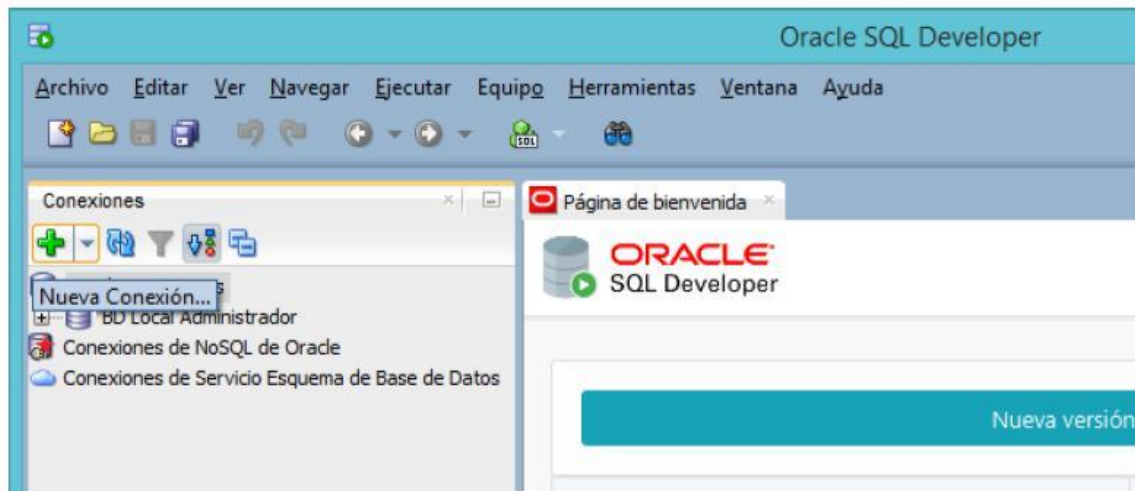
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

Bases de Datos
4ta. práctica Dirigida
(Segundo Semestre 2020)

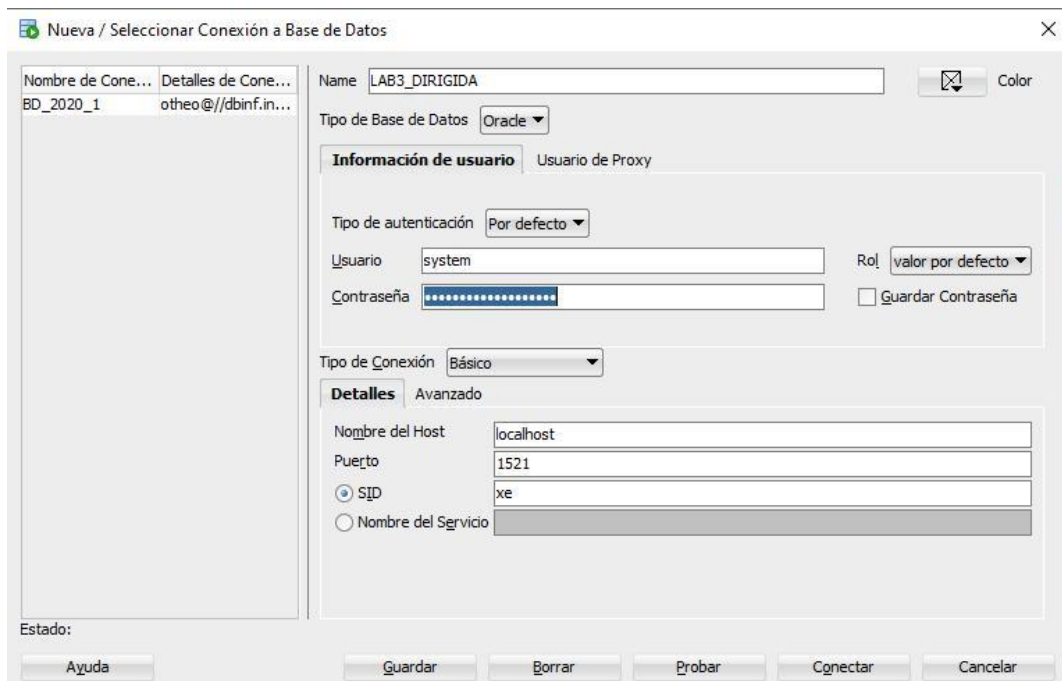
GUIA – FUNCIONES - PROCEDIMIENTOS ALMACENADOS

Conectarse a una base de datos

Primero, ejecute **Oracle SQL Developer**, cierre la pestaña **Página de bienvenida**, y en el panel de **Conexiones**, haga clic en el ícono + para crear una nueva conexión.



Se abrirá la siguiente ventana:

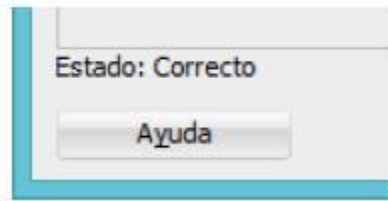


Ingresa los siguientes datos para la conexión:

- Nombre de conexión (Name): **LAB4_DIRIGIDA**
- Usuario: **system**
- Contraseña: **Debe escribir la contraseña que ingresó al instalar Oracle Database Express Edition.**

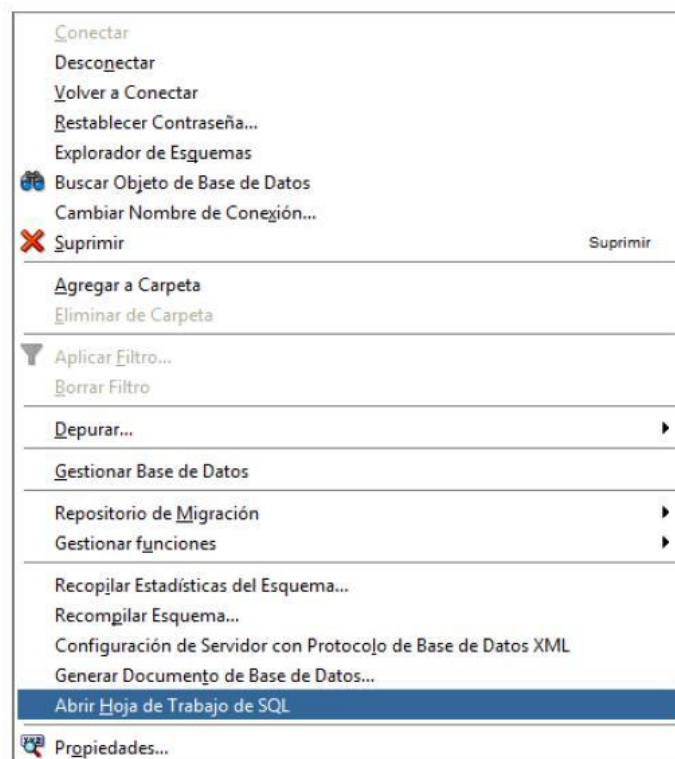
Puede dejar el resto de los parámetros tal y como aparecen por defecto.

Haga clic en el botón **Probar**. Si todo está bien configurado, aparecerá el mensaje **Estado: Correcto** en la parte inferior izquierda.

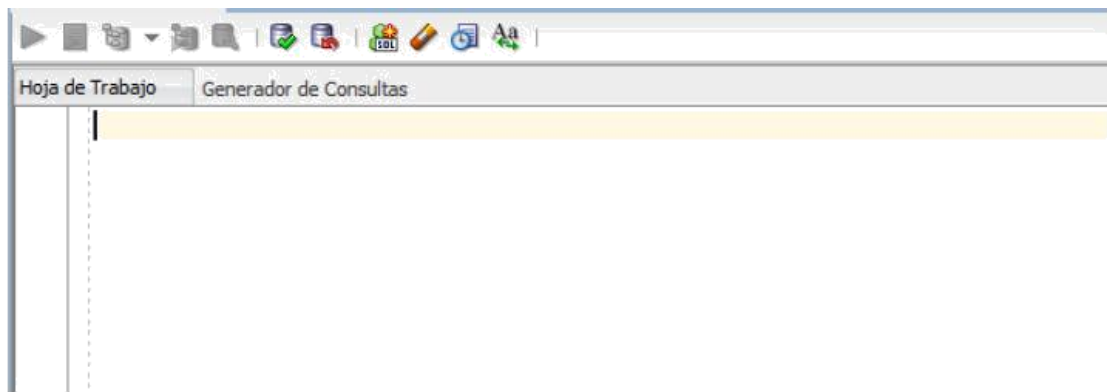


Luego, haga clic en el botón **Guardar**, y luego clic en el botón **Conectar**. El nombre de la conexión creada (**LAB4_DIRIGIDA**) aparecerá en el panel de Conexiones.

Haga clic derecho sobre el nombre de la conexión **LAB4_DIRIGIDA** y aparecerá el siguiente menú contextual:



Hacemos clic en **Abrir Hoja de Trabajo de SQL**. Nos saldrá una pantalla similar a la siguiente:



Esta es la **hoja de trabajo**, donde se podrán escribir y ejecutar los scripts SQL.

Para iniciar, ejecute los scripts que se adjuntan en la parte dirigida en el siguiente orden:

- INF246_2020-2_LAB4_01_dirigida_drops.sql
- INF246_2020-2_LAB4_02_dirigida_modelo.sql
- INF246_2020-2_LAB4_03_dirigida_inserts.sql

Para esta parte dirigida se usarán las siguientes tablas:

TABLA CLIENTE

Campo	Tipo	Tamaño	¿Clave primaria ?	¿Clave foránea ?
ID_CLIENTE	NUMBER	-	Sí	-
APE_PATERNO	VARCHAR2	30	-	-
APE_MATERNO	VARCHAR2	30	-	-
NOMBRE	VARCHAR2	30	-	-
SEXO	VARCHAR2	1	-	-
FECHA_NACIMIENTO	DATE	-	-	-

TABLA PRODUCTO

Campo	Tipo	Tamaño	¿Clave primaria?	¿Clave foránea ?
ID_PRODUCTO	NUMBER	-	Sí	-
NOMBRE	VARCHAR2	30	-	-
PRECIO	NUMBER		-	-
STOCK	NUMBER		-	-

TABLA VENTA

Campo	Tipo	Tamaño	¿Clave primaria?	¿Clave foránea ?
ID_VENTA	NUMBER	-	Sí	-
ID_CLIENTE	NUMBER	-	-	Sí
FECHA_VENTA	DATE		-	-

TABLA DETALLE_VENTA

Campo	Tipo	Tamaño	¿Clave primaria?	¿Clave foránea ?
ID_DETALLE_VENTA	NUMBER	-	Sí	-
ID_VENTA	NUMBER	-	-	Sí
ID_PRODUCTO	NUMBER	-	-	Sí
CANTIDAD	NUMBER	-	-	-
TOTAL	NUMBER	-	-	-

- 1) Mostrar mediante el uso de un bloque anónimo el producto con menor y mayor stock.

SET SERVEROUTPUT ON

DECLARE

vNombreMenor producto.nombre%TYPE;

vNombreMayor producto.nombre%TYPE;

vStockMenor producto.stock%TYPE;

vStockMayor producto.stock%TYPE;

BEGIN

SELECT NOMBRE, STOCK

INTO vNombreMenor, vStockMenor

FROM PRODUCTO

WHERE stock= (SELECT MIN(stock) FROM PRODUCTO);

dbms_output.put_line('Producto con menor stock: '||vNombreMenor||' - '||'Stock: '||vStockMenor);

SELECT NOMBRE, STOCK

INTO vNombreMayor, vStockMayor

FROM PRODUCTO

WHERE stock= (SELECT MAX(stock) FROM PRODUCTO);

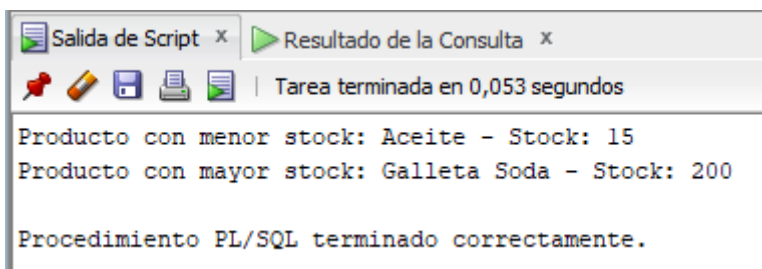
dbms_output.put_line('Producto con mayor stock: '||vNombreMayor||' - '||'Stock: '||vStockMayor);

EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line('Ocurrió un ERROR');

END;

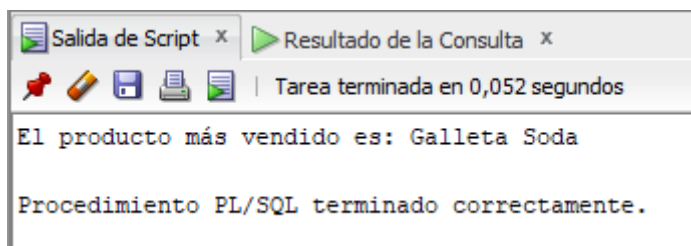


- 2) Mostrar mediante el uso de un bloque anónimo el producto más vendido

```
SET SERVEROUTPUT ON

DECLARE
vNombre producto.nombre%TYPE;

BEGIN
SELECT NOMBRE
INTO vNombre
FROM PRODUCTO
WHERE ID_PRODUCTO=(SELECT ID_PRODUCTO
FROM DETALLE_VENTA
WHERE CANTIDAD=(SELECT MAX(CANTIDAD)FROM DETALLE_VENTA));
dbms_output.put_line('El producto más vendido es: '||vNombre);
EXCEPTION
WHEN OTHERS THEN
dbms_output.put_line('Ocurrió un ERROR');
END;
```



- 3) Mostrar mediante el uso de un bloque anónimo la cantidad de ventas realizadas el día de hoy.

```
SET SERVEROUTPUT ON

DECLARE
vCantidad detalle_venta.cantidad%TYPE;

BEGIN
SELECT SUM(CANTIDAD)
INTO vCantidad
FROM DETALLE_VENTA D, VENTA V
WHERE (D.id_venta=V.id_venta) AND V.fecha_venta= SYSDATE;
IF vCantidad IS NULL THEN
dbms_output.put_line('No existen ventas el día de hoy');
ELSE
```

```

dbms_output.put_line('La cantidad de ventas realizadas el día de hoy
son:'||vCantidad);

END IF;

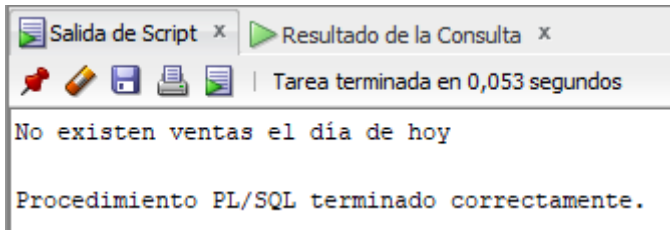
EXCEPTION

WHEN OTHERS THEN

dbms_output.put_line('Ocurrió un ERROR');

END;

```



- 4) Mostrar mediante una función el nombre y apellido del cliente que realizo compras en una determinada fecha.

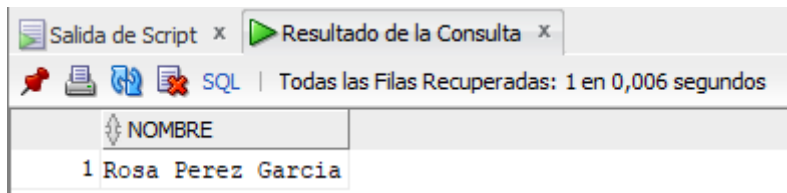
```

CREATE OR REPLACE FUNCTION f_Obtener_Nombre_Cliente(fecha date)
return varchar2
AS
vNombre cliente.nombre%TYPE;
vApePaterno cliente.ape_paterno%TYPE;
vApeMaterno cliente.ape_materno%TYPE;
BEGIN
SELECT C.NOMBRE, C.APE_PATERNO, C.APE_MATERNO
INTO vNombre, vApePaterno, vApeMaterno
FROM CLIENTE C, VENTA V
WHERE C.id_cliente=V.id_cliente AND V.fecha_venta= fecha;
RETURN (vNombre||' '||vapepaterno||' '||vapematerno);
EXCEPTION
WHEN NO_DATA_FOUND THEN
return('ERROR: El registro no existe');
WHEN TOO_MANY_ROWS THEN
return('ERROR: Devuelve muchas filas');
WHEN OTHERS THEN
return 'Ocurrió un error';
END;

```

** Probando la ejecución:

```
SELECT f_Obtener_Nombre_Cliente('15/10/20') AS NOMBRE FROM DUAL;
```

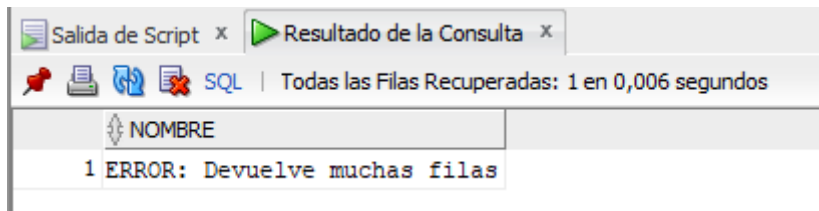


Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,006 segundos

NOMBRE
1 Rosa Perez Garcia

```
SELECT f_Obtener_Nombre_Cliente('17/10/20') AS NOMBRE FROM DUAL;
```

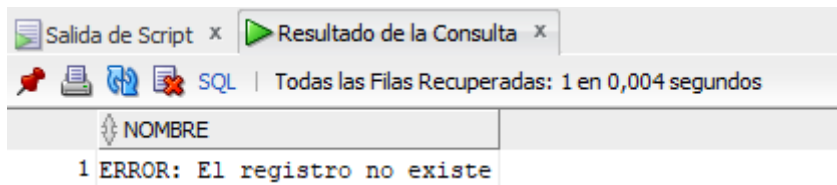


Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,006 segundos

NOMBRE
1 ERROR: Devuelve muchas filas

```
SELECT f_Obtener_Nombre_Cliente('20/10/20') AS NOMBRE FROM DUAL;
```



Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,004 segundos

NOMBRE
1 ERROR: El registro no existe

5) Mostrar la fecha de nacimiento del cliente obtenido en la pregunta 4.

```
SET SERVEROUTPUT ON
```

```
CREATE OR REPLACE PROCEDURE p_Obtener_Fecha_Nacimiento
```

```
AS
```

```
vNombre varchar2(60);
```

```
vFechaNacimiento date;
```

```
BEGIN
```

```
vNombre:= f_Obtener_Nombre_Cliente('15/10/20');
```

```
SELECT FECHA_NACIMIENTO
```

```
INTO vFechaNacimiento
```

```
FROM CLIENTE
```

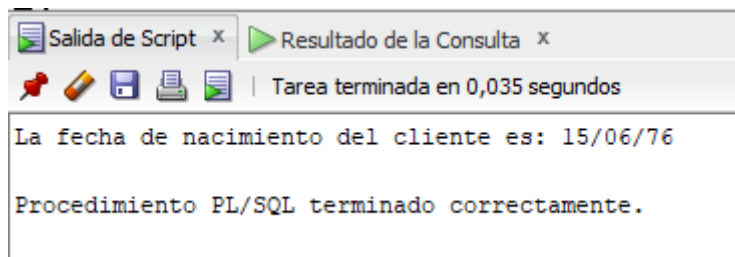
```
WHERE cliente.nombre||' '||cliente.ape_paterno||' '||cliente.ape_materno =  
vNombre;
```

```
dbms_output.put_line('La fecha de nacimiento del cliente es:  
'||vFechaNacimiento);
```

```
END;
```

** Probando la ejecución:

```
EXEC p_Obtener_Fecha_Nacimiento;
```



- 6) Crear un procedimiento que mediante el nombre completo de un cliente muestre la cantidad de productos que ha comprado y el monto promedio que ha pagado por los mismos. **(Hacer uso de parámetros IN y OUT)**

```
CREATE OR REPLACE PROCEDURE p_Obtener_Cantidad_Promedio(vNombre  
IN VARCHAR2,vCantidad OUT NUMBER, vPago_Promedio OUT NUMBER)
```

```
AS
```

```
vCantidad1 NUMBER;
```

```
vPago_Promedio1 NUMBER;
```

```
BEGIN
```

```
SELECT COUNT(*), ROUND( AVG(TOTAL), 2)
```

```
INTO vCantidad1, vPago_Promedio1
```

```
FROM CLIENTE C, VENTA V, DETALLE_VENTA D
```

```
WHERE C.ID_CLIENTE = V.ID_CLIENTE AND V.ID_VENTA = D.ID_VENTA
```

```
AND C.nombre||' '||C.ape_paterno||' '||C.ape_materno= vNombre;
```

```
vCantidad:=vCantidad1;
```

```
vPago_Promedio:=vPago_Promedio1;
```

```
IF vCantidad= 0 THEN
```

```
    dbms_output.put_line('ERROR: El cliente ingresado no ha realizado  
    compras');
```

```
ELSE
```

```
    dbms_output.put_line('La cantidad de productos comprados por '||vNombre||'  
    es: '||vCantidad||' y el promedio pagado por estos productos es:  
    '||vPago_Promedio);
```

```
END IF;
```

```
END;
```

**** Probando la ejecución en un bloque anónimo**

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

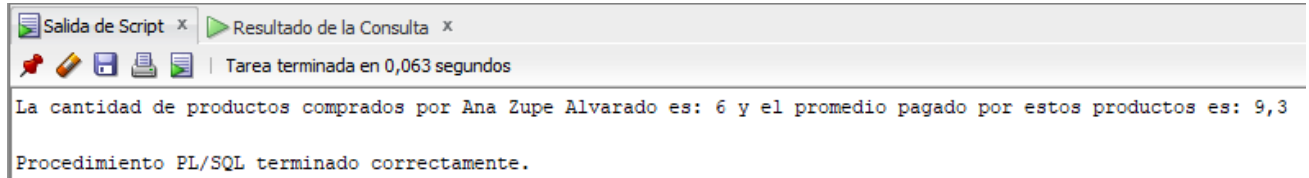
```
vCantidad number;
```

```
vPago_Promedio number;
```


BEGIN

**p_Obtener_Cantidad_Promedio('Ana Zupe Alvarado',vCantidad,
vPago_Promedio);**

END;



SET SERVEROUTPUT ON

DECLARE

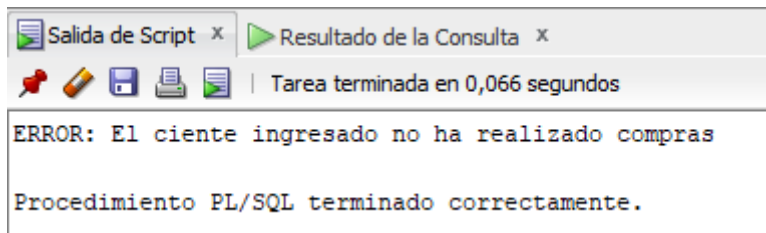
vCantidad number;

vPago_Promedio number;

BEGIN

**p_Obtener_Cantidad_Promedio('Mercedes Luna Valenzuela',vCantidad,
vPago_Promedio);**

END;



09 de Noviembre del 2020

MLV