

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

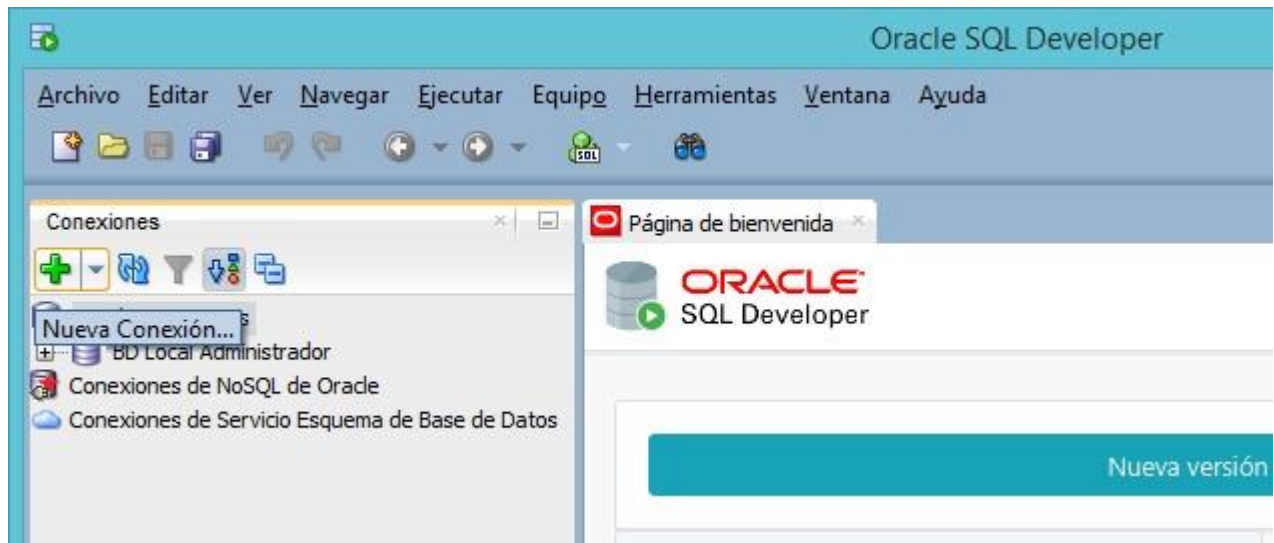
Bases de Datos

2da. Práctica Dirigida
(Segundo Semestre 2021)

GUIA – SQL DDL – SQL DML

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.



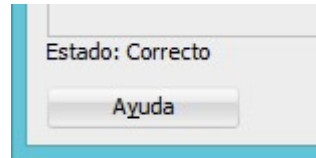
Se abrirá la siguiente ventana:

Ingrese los siguientes datos para la conexión:

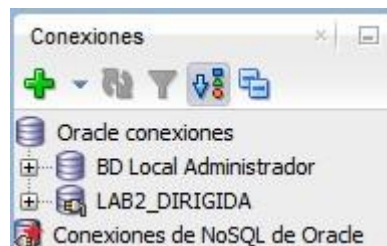
- Nombre de conexión (Name): **LAB2_DIRIGIDA**
- Usuario: **system**
- Contraseña: **Debe escribir la contraseña que ingresó al instalar Oracle Database Express Edition.**

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

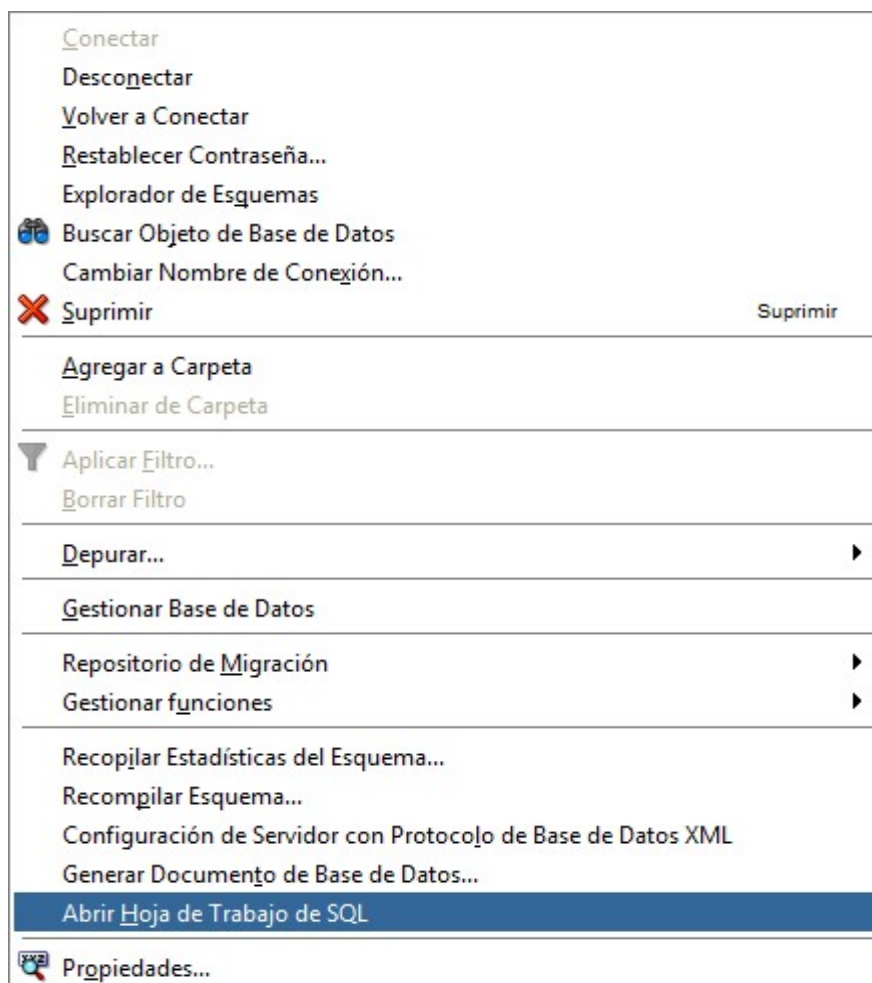
Haga clic en el botón **Probar**. Si todo está bien configurado, aparecerá el mensaje **Estado: Correcto** en la parte inferior izquierda.



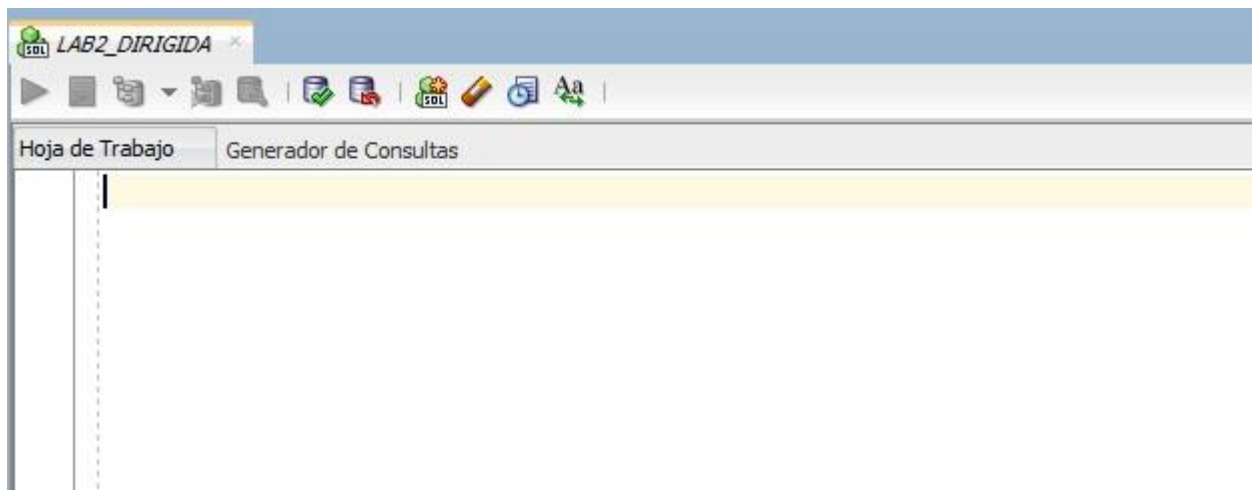
Luego, haga clic en el botón **Guardar**, y luego clic en el botón **Conectar**. El nombre de la conexión creada (**LAB2_DIRIGIDA**) aparecerá en el panel de Conexiones.



Haga clic derecho sobre el nombre de la conexión **LAB2_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.

Creación de tablas de base de datos (CREATE TABLE)

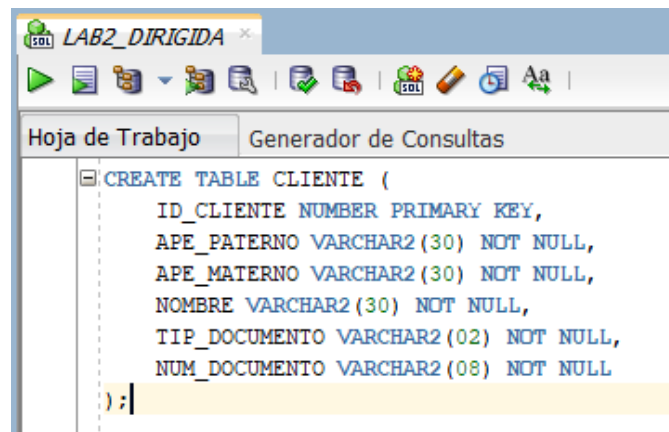
1. Cree la tabla **CLIENTE** con los siguientes campos:

Campo	Tipo	Tamaño	¿Clave primaria?	¿Admite nulos?
ID_CLIENTE	NUMBER	-	Sí	No
APE_PATERNO	VARCHAR2	30	No	No
APE_MATERNO	VARCHAR2	30	No	No
NOMBRE	VARCHAR2	30	No	No
TIP_DOCUMENTO	VARCHAR2	02	No	No
NUM_DOCUMENTO	VARCHAR2	08	No	No

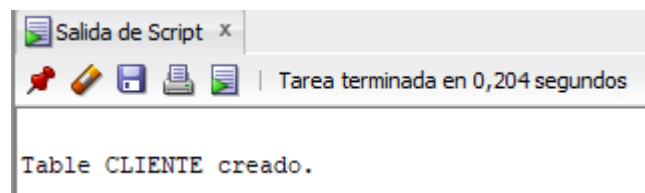
Escriba el siguiente script SQL en la hoja de trabajo:

```
CREATE TABLE CLIENTE (  
    ID_CLIENTE NUMBER PRIMARY KEY,  
    APE_PATERNO VARCHAR2(30) NOT NULL,  
    APE_MATERNO VARCHAR2(30) NOT NULL,  
    NOMBRE VARCHAR2(30) NOT NULL,  
    TIP_DOCUMENTO VARCHAR2(02) NOT NULL,  
    NUM_DOCUMENTO VARCHAR2(08) NOT NULL  
);
```

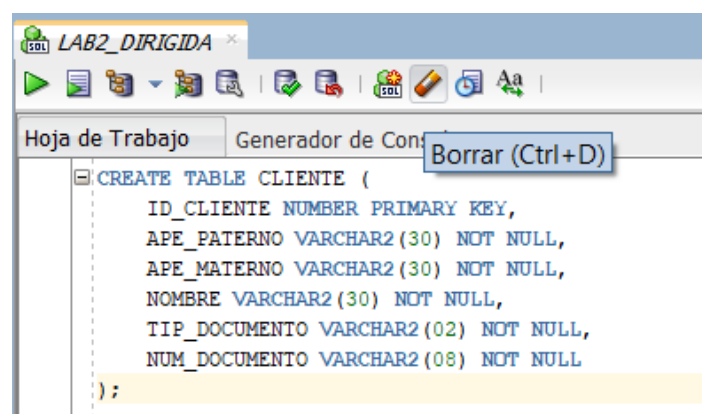
Para ejecutar el script SQL en la hoja de trabajo, haga clic en el botón **Ejecutar Script**, o presione la tecla **F5**.



En el panel Salida de Script, deberá salir el siguiente mensaje:



Puede hacer clic en el botón **Borrar**, o presionar las teclas **Ctrl+D**, para limpiar la hoja de trabajo.



2. Cree la tabla **PRODUCTO** con los siguientes campos:

Campo	Tipo	Tamaño	¿Clave primaria?	¿Admite nulos?
ID_PRODUCTO	NUMBER	-	Sí	No
NOMBRE	VARCHAR2	50	No	No
PRECIO	NUMBER		No	No

Escriba el siguiente script SQL en la hoja de trabajo y ejecútelo:

```
CREATE TABLE PRODUCTO (
  ID_PRODUCTO NUMBER NOT NULL,
  NOMBRE VARCHAR2(30) NOT NULL,
  PRECIO NUMBER NOT NULL,
  PRIMARY KEY (ID_PRODUCTO)
);
```

3. Cree la tabla **DETALLE_VENTA** con los siguientes campos:

Campo	Tipo	Tamaño	¿Clave primaria?	¿Admite nulos?
ID_DETALLE_VENTA	NUMBER	-	Sí	No
ID_CLIENTE	NUMBER	-	No	No
ID_PRODUCTO	NUMBER	-	No	No
CANTIDAD	NUMBER	-	No	No
TOTAL	NUMBER	-	No	No

```
CREATE TABLE DETALLE_VENTA (  
  ID_DETALLE_VENTA NUMBER NOT NULL,  
  ID_PRODUCTO NUMBER NOT NULL,  
  CANTIDAD NUMBER NOT NULL,  
  TOTAL NUMBER NOT NULL,  
  PRIMARY KEY (ID_DETALLE_VENTA)  
);
```

4. Cree la tabla **VENTA** con los siguientes campos:

Campo	Tipo	Tamaño	¿Clave primaria?	¿Admite nulos?
ID_VENTA	NUMBER	-	Sí	No
ID_CLIENTE	NUMBER	-	No	No
FECHA_VENTA	DATE		No	No

```
CREATE TABLE VENTA (  
  ID_VENTA NUMBER NOT NULL,  
  ID_CLIENTE NUMBER NOT NULL,  
  FECHA_VENTA DATE NOT NULL,  
  PRIMARY KEY (ID_VENTA)  
);
```

5. Cree la tabla **CUPON** con los siguientes campos:

Campo	Tipo	Tamaño	¿Clave primaria?	¿Admite nulos?
ID_CUPON	NUMBER	-	Sí	No
NOMBRE	VARCHAR2(50)	50	No	No
DESCRIPCION	VARCHAR2(100)	100	No	No
FECHA_VIGENCIA_INI	DATE	-	No	No
FECHA_VIGENCIA_FIN	DATE	-	No	No

```
CREATE TABLE CUPON (  
  ID_CUPON NUMBER NOT NULL,  
  NOMBRE VARCHAR2(50) NOT NULL,  
  DESCRIPCION VARCHAR2(100) NOT NULL,  
  FECHA_VIGENCIA_INI DATE NOT NULL,  
  FECHA_VIGENCIA_FIN DATE NOT NULL,  
  PRIMARY KEY (ID_CUPON)  
);
```

Modificación de tablas de base de datos (ALTER TABLE)

1. Modifique el tamaño del campo **NUM_DOCUMENTO** de la tabla **CLIENTE** de tal manera que su **tamaño** sea **12**.

```
ALTER TABLE CLIENTE MODIFY NUM_DOCUMENTO VARCHAR2(12);
```

2. Agregue la columna **FECHA_NACIMIENTO** tipo **DATE** a la tabla **CLIENTE**.

```
ALTER TABLE CLIENTE ADD FECHA_NACIMIENTO DATE;
```

3. Agregar en la tabla **DETALLE_VENTA** el campo **ID_VENTA** de tipo **NUMBER**.

```
ALTER TABLE DETALLE_VENTA ADD ID_VENTA NUMBER;
```

4. Agregar en la tabla **PRODUCTO** el campo **STOCK** de tipo **NUMBER**.

```
ALTER TABLE PRODUCTO ADD STOCK NUMBER;
```

5. Hacer que el campo **ID_VENTA** recién añadido a la tabla **DETALLE_VENTA** sea una **clave foránea** que haga referencia a la tabla **VENTA**.

```
ALTER TABLE DETALLE_VENTA ADD FOREIGN KEY (ID_VENTA) REFERENCES VENTA (ID_VENTA);
```

6. Hacer que el campo **ID_CLIENTE** en la tabla **VENTA** sea una **clave foránea** que haga referencia a la tabla **CLIENTE**.

```
ALTER TABLE VENTA ADD FOREIGN KEY (ID_CLIENTE) REFERENCES CLIENTE (ID_CLIENTE);
```

7. Agregar en la tabla **CUPON** el campo **ID_PRODUCTO** de tipo **NUMBER**.

```
ALTER TABLE CUPON ADD ID_PRODUCTO NUMBER;
```

8. Hacer que el campo **ID_PRODUCTO** recién añadido a la tabla **CUPON** sea una **clave foránea** que haga referencia a la tabla **PRODUCTO**.

```
ALTER TABLE CUPON ADD FOREIGN KEY (ID_PRODUCTO) REFERENCES PRODUCTO (ID_PRODUCTO);
```

Sentencias SQL para manipulación de datos (SELECT, INSERT, UPDATE, DELETE)

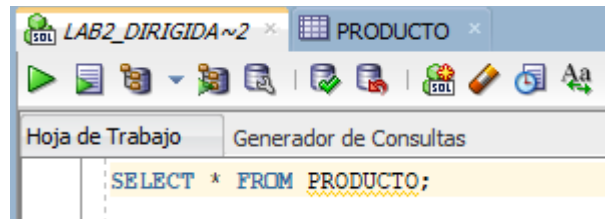
1. Insertar los siguientes datos en la tabla **PRODUCTO**.

```
INSERT INTO PRODUCTO (ID_PRODUCTO, NOMBRE, PRECIO, STOCK) VALUES (1, 'Arroz 1kg. Costeño', 3, 10);  
INSERT INTO PRODUCTO (ID_PRODUCTO, NOMBRE, PRECIO, STOCK) VALUES (2, 'Azúcar Rubia 1kg', 2, 8);  
INSERT INTO PRODUCTO (ID_PRODUCTO, NOMBRE, PRECIO, STOCK) VALUES (3, 'Galleta soda', 1, 20);  
INSERT INTO PRODUCTO (ID_PRODUCTO, NOMBRE, PRECIO, STOCK) VALUES (4, 'Chocolate Sublime', 1, 3);  
INSERT INTO PRODUCTO (ID_PRODUCTO, NOMBRE, PRECIO, STOCK) VALUES (5, 'Shampoo', 16, 2);
```

2. Mostrar el contenido de la tabla **PRODUCTO**.

```
SELECT * FROM PRODUCTO;
```

Para visualizar consultas (SELECT), puede seleccionar todo el script, y hacer clic en el botón **Sentencia de Ejecución**, o presionar las teclas **Ctrl+Enter**.



La consulta debería mostrarse de la siguiente manera:

SQL Todas las Filas Recuperadas: 9 en 0.016 segundos				
	ID_PROD...	NOMBRE	PRECIO	STOCK
1		1 Arroz 1kg. Costeño	3	10
2		2 Azúcar Rubia 1kg	2	8
3		3 Galleta soda	1	20
4		4 Chocolate Sublime	1	3
5		5 Shampoo	16	2
6		6 Pañales Babysec	29	4
7		7 Pañales Tena	34	4
8		8 Desodorante	2	5
9		9 Vitamina C	3	5

3. Copie el contenido del archivo **INF246_2021-1_LAB2_DIRIGIDA_inserts.sql** en la hoja de trabajo y ejecute el script.
4. Muestre el contenido de la tabla **CLIENTE**.

```
SELECT * FROM CLIENTE;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 3 en 0.008 segundos

ID_CLIENTE	APE_PATERNO	APE_MATERNO	NOMBRE	TIP_DOCUMENTO	NUM_DOCUMENTO	FECHA_NACIMIENTO
1	1 Perez	Garcia	Rosa	01	40897410	15/06/76
2	2 Prado	Garcia	Maria	01	07925529	10/01/56
3	3 Zupe	Alvarado	Anna	04	123456789012	09/03/80

5. Muestre los **apellidos y nombres** de las personas, ordenadas por **apellido paterno**.

```
SELECT APE_PATERNO, APE_MATERNO, NOMBRE  
FROM CLIENTE  
ORDER BY APE_PATERNO;
```

	APE_PATERNO	APE_MATERNO	NOMBRE
1	Perez	Garcia	Rosa
2	Prado	Garcia	Maria
3	Zupe	Alvarado	Anna

6. Muestre los productos con stock mayor a 1 y precio mayor a 2 ordenados por el precio y stock de forma descendiente.

```
SELECT NOMBRE, PRECIO, STOCK
FROM PRODUCTO
WHERE PRECIO > 2 AND STOCK > 1
ORDER BY PRECIO ASC, STOCK ASC;
```

	NOMBRE	PRECIO	STOCK
1	Arroz 1kg. Costeño	3	10
2	Shampoo	16	2

7. Muestre los apellidos y nombres concatenados con espacios y comas de los clientes que realizaron compras entre Agosto y Setiembre del 2019 y que sean ciudadanos extranjeros (TIP_DOCUMENTO = '04')

```
SELECT C.APE_PATERNO || ' ' || C.APE_MATERNO || ', ' || C.NOMBRE AS NOMBRE_COMPLETO,
V.FECHA_VENTA
FROM CLIENTE C, VENTA V
WHERE V.ID_CLIENTE = C.ID_CLIENTE
AND FECHA_VENTA BETWEEN to_date('01/08/2019', 'dd/mm/yyyy') AND
to_date('30/09/2019', 'dd/mm/yyyy')
AND C.TIP_DOCUMENTO = '04'
ORDER BY C.APE_PATERNO ASC, C.APE_MATERNO ASC;
```

	NOMBRE_COMPLETO	FECHA_VENTA
1	Zupe Alvarado, Anna	15/08/19
2	Zupe Alvarado, Anna	01/08/19

8. Muestre la cantidad de ítems realizados en cada venta

```
SELECT V.ID_VENTA, COUNT(ID_PRODUCTO) AS ITEMS
FROM DETALLE_VENTA DV, VENTA V
WHERE DV.ID_VENTA = V.ID_VENTA
GROUP BY V.ID_VENTA
ORDER BY ID_VENTA ASC;
```


	ID_VENTA	ITEMS
1	1	2
2	2	1
3	3	1
4	4	1
5	5	3

9. Muestre el monto total de cada una de las ventas ordenados de manera descendente por el monto

```
SELECT V.ID_VENTA, COUNT(ID_PRODUCTO) AS ITEMS, SUM(TOTAL) AS TOTAL
FROM DETALLE_VENTA DV, VENTA V
WHERE DV.ID_VENTA = V.ID_VENTA
GROUP BY V.ID_VENTA
ORDER BY 3 DESC;
```

	ID_VENTA	ITEMS	TOTAL
1	4	1	16
2	5	3	14
3	1	2	7
4	2	1	6
5	3	1	4

10. Muestre los productos únicos vendidos entre los meses de agosto y septiembre del 2019.

```
SELECT DISTINCT P.ID_PRODUCTO, P.NOMBRE
FROM DETALLE_VENTA DV, VENTA V, PRODUCTO P
WHERE DV.ID_VENTA = V.ID_VENTA
AND DV.ID_PRODUCTO = P.ID_PRODUCTO
AND V.FECHA_VENTA BETWEEN to_date('01/08/2019', 'dd/mm/yyyy') AND
to_date('30/09/2019', 'dd/mm/yyyy');
```

	ID_PRODUCTO	NOMBRE
1	1	Arroz 1kg. Costeño
2	5	Shampoo
3	2	Azúcar Rubia 1kg

11. Mostrar los clientes que han comprado el producto Shampoo para ofrecerles un descuento en su siguiente compra.

```
SELECT DISTINCT ( C.APE_PATERNO || ' ' || C.APE_MATERNO || ', ' || C.NOMBRE) AS
NOMBRE_COMPLETO
FROM DETALLE_VENTA DV, VENTA V, PRODUCTO P, CLIENTE C
WHERE DV.ID_VENTA = V.ID_VENTA
AND DV.ID_PRODUCTO = P.ID_PRODUCTO
AND C.ID_CLIENTE = V.ID_CLIENTE
AND P.NOMBRE = 'Shampoo';
```

NOMBRE_COMPLETO
1 Zupe Alvarado, Anna

12. Mostrar los cupones de descuento que están vigentes entre los meses de mayo y septiembre del 2019 y el stock de los productos asociados.

```
SELECT C.NOMBRE, C.DESCRIPCION, C.FECHA_VIGENCIA_INI, C.FECHA_VIGENCIA_FIN,
P.NOMBRE, P.STOCK, P.PRECIO
FROM CUPON C, PRODUCTO P
WHERE FECHA_VIGENCIA_INI >= to_date('01/05/2019', 'dd/mm/yyyy')
AND FECHA_VIGENCIA_FIN <= to_date('30/09/2019', 'dd/mm/yyyy')
AND C.ID_PRODUCTO = P.ID_PRODUCTO
```

NOMBRE	DESCRIPCION	FECHA_VIGENCIA_INI	FECHA_VIGENCIA_FIN	NOMBRE_1	STOCK	PRECIO
1 Cupon Todo Pañales Bebe	15% Dcto. en pañales	01/08/19	01/09/19	Pañales Babysec	4	29
2 Cupon Todo Pañales Adulto	10% Dcto. en pañales	01/08/19	01/09/19	Pañales Tena	4	34
3 Cupon Mes de mamá	20% Dcto. en Limpieza personal	01/05/19	31/05/19	Desodorante	5	2

13. Actualizar el total en el detalle de la venta del producto Shampoo, debido a que hubo un problema al calcular el monto total de este.

Nota: Para visualizar los nuevos valores puede hacer una consulta a la tabla usados en la sentencia.

```
SELECT * FROM PRODUCTO
WHERE NOMBRE = 'Shampoo';
--ID PRODUCTO = 5
--PRECIO = 16

UPDATE DETALLE_VENTA DV
SET DV.TOTAL = DV.CANTIDAD * 16
WHERE DV.ID_PRODUCTO = 5;
```

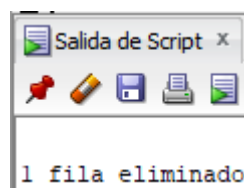
14. Mostrar la cantidad de clientes que nacieron antes del año 1980.

```
SELECT COUNT(*)
FROM CLIENTE
WHERE FECHA_NACIMIENTO <= to_date('01/01/1980', 'dd/mm/yyyy');
```

COUNT(*)
1
2

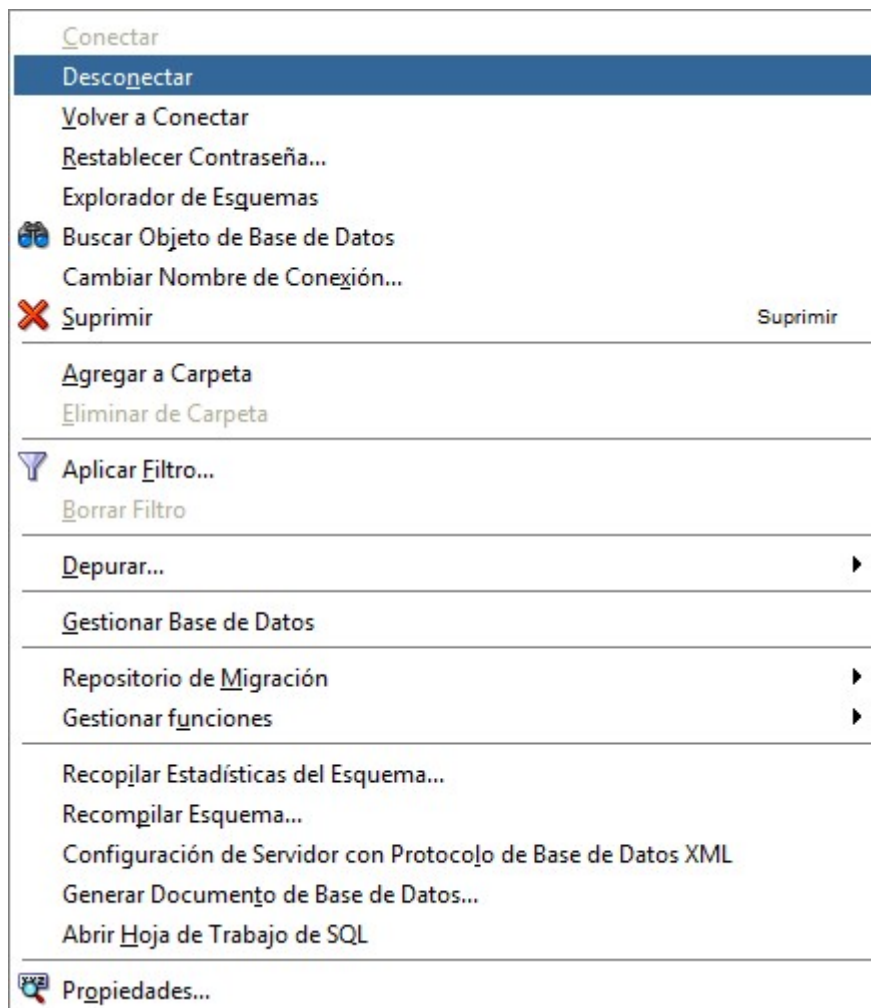
15. Eliminar los datos de los clientes que nacieron en el año 1976.

```
DELETE FROM CLIENTE
WHERE FECHA_NACIMIENTO BETWEEN to_date('01/01/1976', 'dd/mm/yyyy') AND
to_date('31/12/1976', 'dd/mm/yyyy');
```

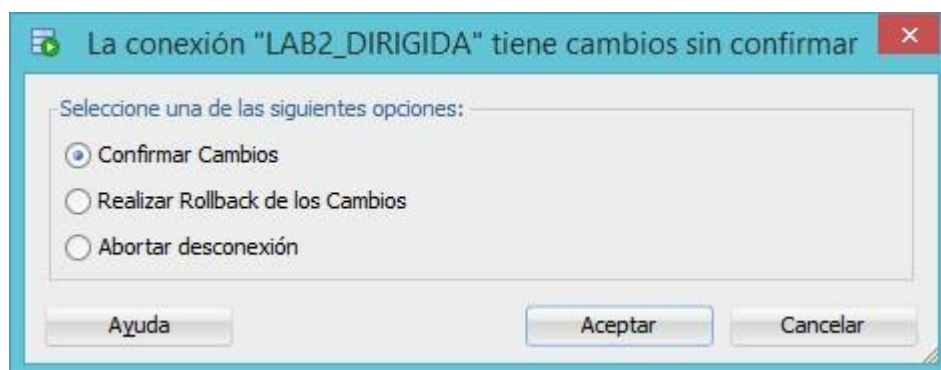


Desconectarse de una base de datos

Para desconectarse de la base de datos, haga clic derecho sobre el nombre de la conexión **LAB2_DIRIGIDA** y aparecerá el siguiente menú contextual:



Haga clic en la opción **Desconectar**. Si le sale un cuadro de diálogo como el que se muestra a continuación, seleccione la opción **Confirmar cambios** y haga clic en **Aceptar**.



San Miguel, 03 de mayo de 2021