

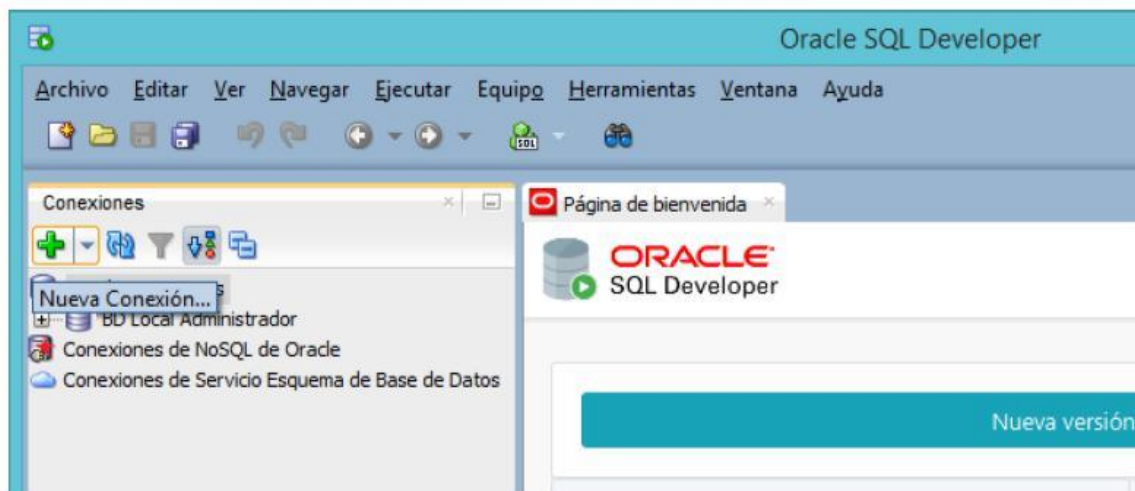
**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

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

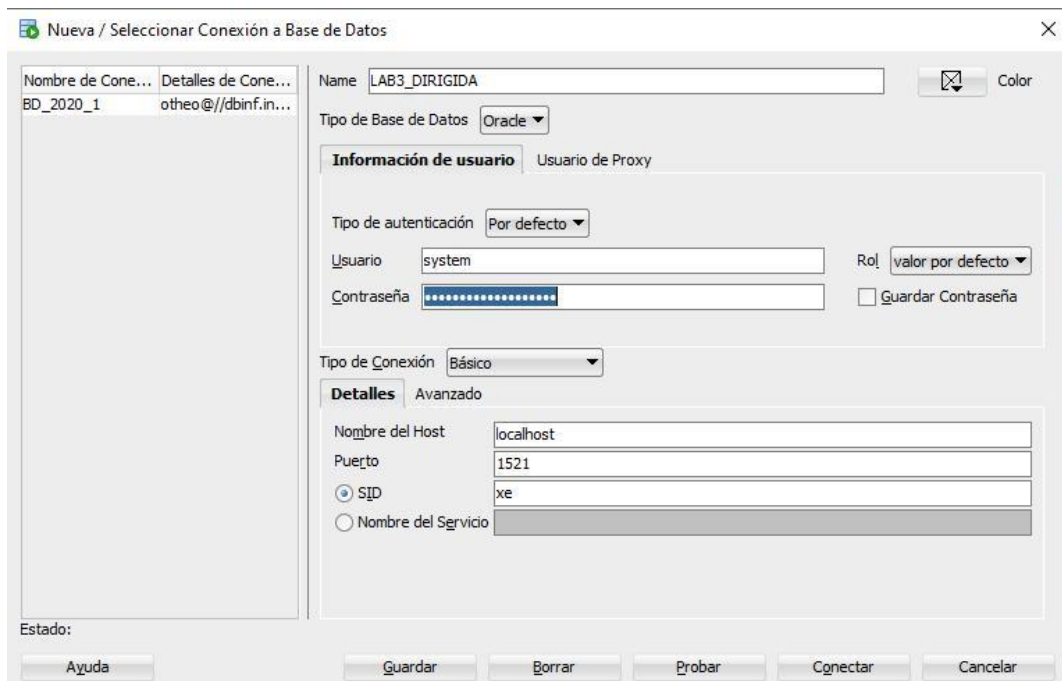
**GUIA – PROCEDIMIENTOS ALMACENADOS - CURSORES**

**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:

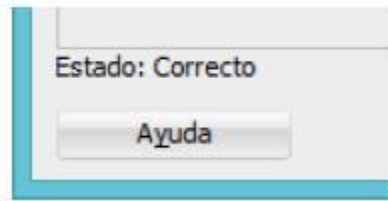


Ingresa los siguientes datos para la conexión:

- Nombre de conexión (Name): **LAB4\_DIRIGIDA**
- Usuario: **system**
- Contraseña: **Debe escribir la contraseña que ingresó al instalar Oracle Database Express Edition.**

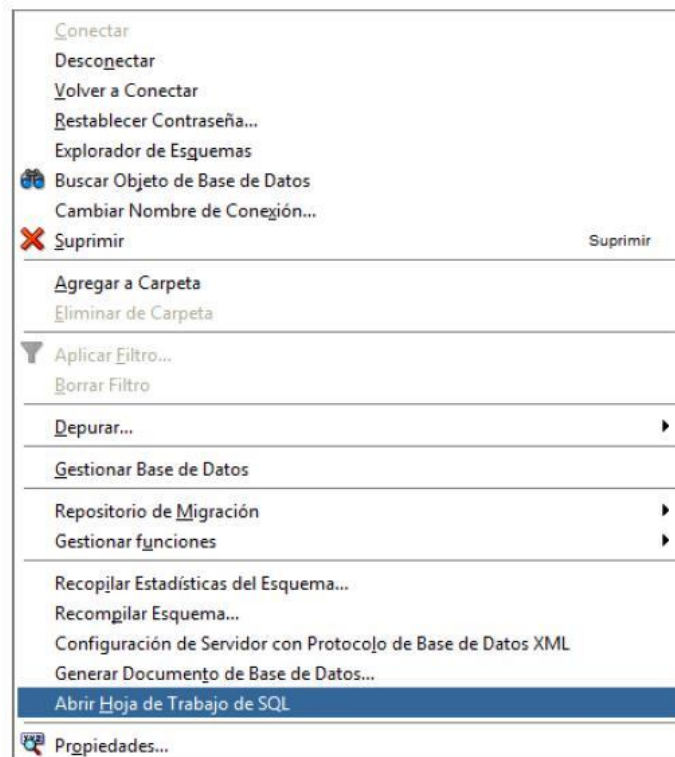
Puede dejar el resto de los parámetros tal y como aparecen por defecto.

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

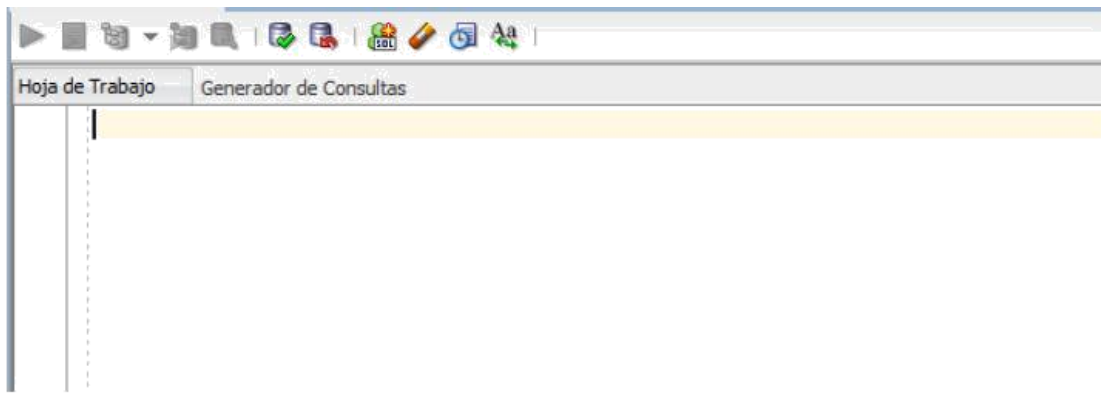


Luego, haga clic en el botón **Guardar**, y luego clic en el botón **Conectar**. El nombre de la conexión creada (**LAB4\_DIRIGIDA**) aparecerá en el panel de Conexiones.

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:

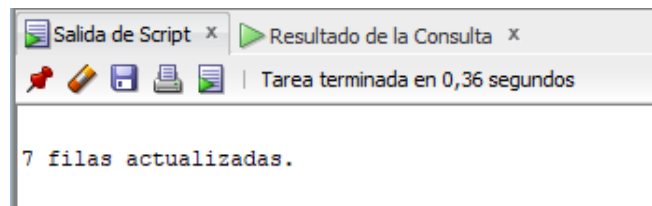
- INF246\_2020-1\_LAB4\_01\_dirigida\_drops.sql
  - INF246\_2020-1\_LAB4\_02\_dirigida\_modelo.sql
  - INF246\_2020-1\_LAB4\_03\_dirigida\_inserts.sql
1. Obtener los dos montos mayores de apertura de cuenta, es decir obtener los montos antes de realizar las transferencias. Tener en consideración el tipo de moneda de la transferencia y de la cuenta. El tipo de cambio es de 3.5 de dólares a soles.
- (\*) Considerar solo a los clientes que han hecho transferencias.

```
SELECT cl.ID_CLIENTE, cl.APE_PATERNO || cl.APE_MATERNO || cl.NOMBRE AS cliente,
c.ID_CUENTA, c.SALDO,
(select SUM(CASE WHEN t.MONEDA = 'D' THEN t.MONTO ELSE (round(t.Monto/3.5)) end)
FROM
SB_TRANSFERENCIA t
WHERE
t.ID_CUENTA_ORIGEN = c.id_cuenta GROUP BY t.ID_CUENTA_ORIGEN) as monto
,
(c.SALDO +
(SELECT SUM(CASE WHEN t.MONEDA = 'D' THEN t.MONTO ELSE (round(t.Monto / 3.5)) END)
FROM
SB_TRANSFERENCIA t
WHERE
t.ID_CUENTA_ORIGEN = c.id_cuenta
GROUP BY t.ID_CUENTA_ORIGEN)
) as monto_apertura_cuenta
FROM
SB_CUENTA c, SB_CLIENTE_CUENTA clc, SB_CLIENTE cl
WHERE
clc.ID_CLIENTE = cl.ID_CLIENTE
AND clc.ID_CUENTA = c.ID_CUENTA
AND (c.SALDO + (select SUM(t.MONTO) from SB_TRANSFERENCIA t where t.ID_CUENTA_ORIGEN =
c.id_cuenta GROUP BY t.ID_CUENTA_ORIGEN)) > 0
ORDER BY 6 DESC;
```

ID_CLIENTE	CLIENTE	ID_CUENTA	SALDO	MONTO	MONTO_APERTURA_CUENTA
1	2 BENAVIDESQUISPERAUL CARLOS	1	16980	371	17351
2	1 TORRESSALAZARROMINA ISABEL	3	3964	143	4107

2. Debido a un error en el sistema, se desea actualizar las cuotas que no fueron procesadas correctamente pero que ya han sido pagadas.

```
UPDATE SB_CUOTA C
SET
C.FECHA_PAGO = C.FECHA_VENCIMIENTO,
C.ESTADO = 1,
C.MONTO_PAGO = (SELECT P.cuota FROM SB_PRESTAMO P WHERE C.ID_PRESTAMO =
P.ID_PRESTAMO)
WHERE
C.estado = 0 AND C.FECHA_PAGO IS null AND
C.FECHA_VENCIMIENTO <= TO_DATE(sysdate, 'dd/mm/yy');
```



3. Crear un procedimiento que permita generar transferencia en soles en la Agencia Aviación a un determinado número de cuenta destino que pertenece a un banco en específico para los dos clientes de la pregunta 1.

---- 1) Creación del procedimiento

```
CREATE or REPLACE PROCEDURE SP_TRANSFERENCIA_OTRA_CUENTA(CUENTA_ORIGEN
NUMBER, NUMERO_CUENTA VARCHAR, ID_BANCO INTEGER, MONTO NUMBER, ID_AGENCIA
NUMBER)
IS
id_trans number;
BEGIN

SELECT MAX(ID_TRANSFERENCIA) INTO id_trans FROM SB_TRANSFERENCIA ;

INSERT INTO SB_TRANSFERENCIA(ID_TRANSFERENCIA,ID_CUENTA_ORIGEN,
NUM_CUENTA_DESTINO, ID_BANCO, MONTO, MONEDA, FECHA_HORA, ID_AGENCIA)
VALUES(id_trans + 1,CUENTA_ORIGEN,numero_cuenta,ID_BANCO,MONTO,'S',TO_DATE(sysdate,
'dd/mm/yy'),2 );

UPDATE SB_CUENTA
SET SALDO = SALDO - MONTO
WHERE id_cuenta = CUENTA_ORIGEN;

END;
```

--2) Probando la ejecución

```
EXECUTE SP_TRANSFERENCIA_OTRA_CUENTA(1,'6732184363905', 3,200,2);
```

4. Obtener el monto de la última cuota pagada del préstamo

-- 1) Creación de la función

```
CREATE OR REPLACE FUNCTION SP_OBTENER_ULTIMA_CUOTA(IDCLIENTE NUMBER)
RETURN NUMBER is
```

```

monto SB_CUOTA.MONTO_PAGO%TYPE;
id_cuota SB_CUOTA.ID_CUOTA%TYPE;
BEGIN

SELECT MAX(C.ID_CUOTA), MAX(C.MONTO_PAGO) INTO id_cuota, monto
FROM SB_CUOTA C, SB_PRESTAMO P
WHERE C.ID_PRESTAMO = P.ID_PRESTAMO
AND P.ID_CLIENTE = idcliente AND C.ESTADO = 1;

RETURN monto;

END;

```

-- 2) Probando la ejecución

```
SELECT sp_obtener_ultima_cuota(3) FROM dual;
```

5. El Banco desea listar los clientes que han solicitado un crédito desde el inicio de la cuarentena para poder brindarles una tasa preferencial.

```

SELECT ROWNUM, C.*
FROM (
    SELECT
        cli.id_cliente, cli.dni, cli.ape_paterno||' '|| cli.ape_materno || ' ' ||cli.nombre as cliente
    FROM
        SB_CLIENTE cli
    WHERE
        cli.id_cliente IN (
            SELECT
                pr.id_cliente
            FROM
                SB_PRESTAMO PR
            WHERE
                pr.fecha >= to_date('15/03/2020','dd/mm/yyyy')
                AND pr.fecha <=LAST_DAY('19/06/2020')
        )
    ORDER BY CLI.APE_PATERNO DESC)
C;

```

	ROWNUM	ID_CLIENTE	DNI	CLIENTE
1	1	1	40894715	TORRESSALAZARROMINA ISABEL
2	2	4	05930128	NESTORUBILLUZFELIPE

6. Crear un procedimiento que permita actualizar el interés a 5% a los clientes de la pregunta 5.

---- 1) Creación del procedimiento

```

CREATE OR REPLACE PROCEDURE SP_ACTUALIZAR_TASA_INTERES AS

CURSOR ccliente IS

SELECT C.ID_CLIENTE
FROM (
    SELECT
        cli.id_cliente,cli.dni, cli.ape_paterno || cli.ape_materno || cli.nombre as cliente

```

```

FROM
SB_CLIENTE cli
WHERE
cli.id_cliente IN (
    SELECT
    pr.id_cliente
    FROM
    SB_PRESTAMO PR
    WHERE
    pr.fecha >= to_date('15/03/2020','dd/mm/yyyy')
    AND pr.fecha <=LAST_DAY('19/06/2020')
)
ORDER BY CLIAPE_PATERNO DESC)
C;
id_cliente SB_PRESTAMO.id_cliente%TYPE;

BEGIN

    OPEN ccliente;
LOOP
    FETCH ccliente INTO id_cliente;
    EXIT WHEN ccliente%NOTFOUND;
    UPDATE SB_PRESTAMO
    SET INTERES = 5
    WHERE ID_CLIENTE = id_cliente and FECHA >= to_date('15/03/2020','dd/mm/yyyy')
    AND FECHA <=LAST_DAY('19/06/2020') ;
END LOOP;
    CLOSE ccliente;
END;

```

--2) Probando la ejecución

```
EXECUTE SP_ACTUALIZAR_TASA_INTERES;
```

7. Crear un procedimiento que permita actualizar la cuota y el plazo de los clientes de la pregunta 5, pues el banco ha dado facilidades de reprogramar sus deudas.

**Tener en cuenta que debe usar parámetros IN y OUT.**

---- 1) Creación del procedimiento que obtiene la cuota y plazo de un determinado cliente

```

CREATE OR REPLACE PROCEDURE SP_OBTENER_PLAZO_CUOTAS(idcliente IN number,v_plazo OUT
number, v_cuota OUT number )
is
BEGIN

SELECT P.PLAZO, P.CUOTA INTO v_plazo, v_cuota FROM SB_PRESTAMO P
WHERE P.ID_CLIENTE = idcliente ;

END;

```

---- 2) Creación del procedimiento que actualiza el plazo y cuotas de todos los clientes

```

CREATE OR REPLACE PROCEDURE SP_ACTUALIZAR_PLAZO_CUOTA AS
CURSOR ccliente IS

```

```

SELECT C.ID_CLIENTE
FROM (
  SELECT
    cli.id_cliente,cli.dni, cli.ape_paterno || cli.ape_materno || cli.nombre as cliente
  FROM
    SB_CLIENTE cli
  WHERE
    cli.id_cliente IN (
      SELECT
        pr.id_cliente
      FROM
        SB_PRESTAMO PR
      WHERE
        pr.fecha >= to_date('15/03/2020','dd/mm/yyyy')
        AND pr.fecha <=LAST_DAY('19/06/2020')
    )
  ORDER BY CLI.APE_PATERNO DESC)
C;

```

```

id_cliente number;
plazo number;
cuota number;

```

```

BEGIN

```

```

  OPEN ccliente;
  LOOP
    FETCH ccliente INTO id_cliente;
    EXIT WHEN ccliente%NOTFOUND;
    SP_OBTENER_PLAZO_CUOTAS(id_cliente,plazo, cuota);
    IF plazo > 20 THEN
      --dbms_output.put_line(plazo);
      UPDATE SB_PRESTAMO
      SET PLAZO = 60, CUOTA = ROUND(MONTO/60)
      WHERE ID_CLIENTE = id_cliente and FECHA >= to_date('15/03/2020','dd/mm/yyyy')
      AND FECHA <=LAST_DAY('19/06/2020') ;
    END IF;
  END LOOP;
  CLOSE ccliente;
END;

```

---- 3) Probando la ejecución

```

EXECUTE SP_ACTUALIZAR_PLAZO_CUOTA;

```

19 de junio del 2020

MPQ - DB