

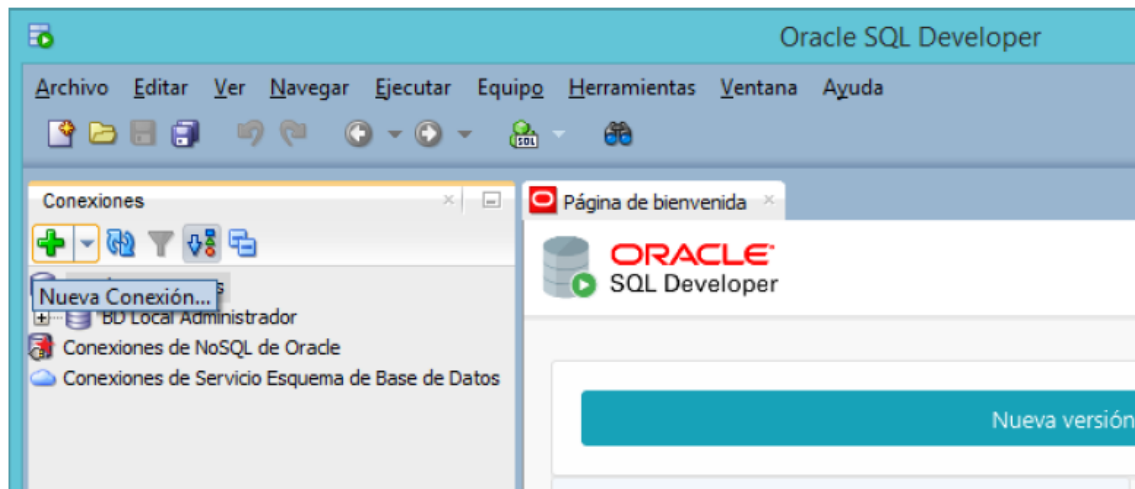
PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

Bases de Datos
3ra. práctica Dirigida
(Primer Semestre 2020)

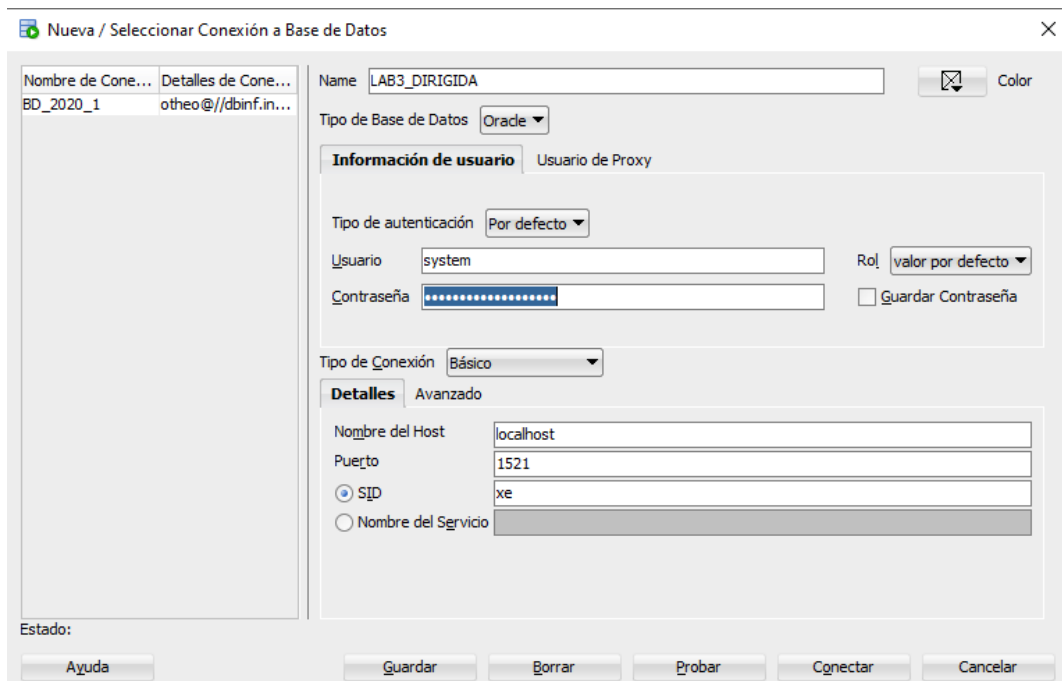
GUIA – SQL SUBCONSULTA – SQL AGRUPACIÓN

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:

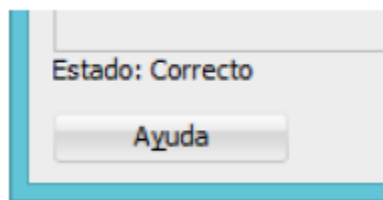


Ingrese los siguientes datos para la conexión:

- Nombre de conexión (Name): **LAB3_DIRIGIDA**
- Usuario: **system**
- Contraseña: **Debe escribir la contraseña que ingresó al instalar Oracle Database Express Edition.**

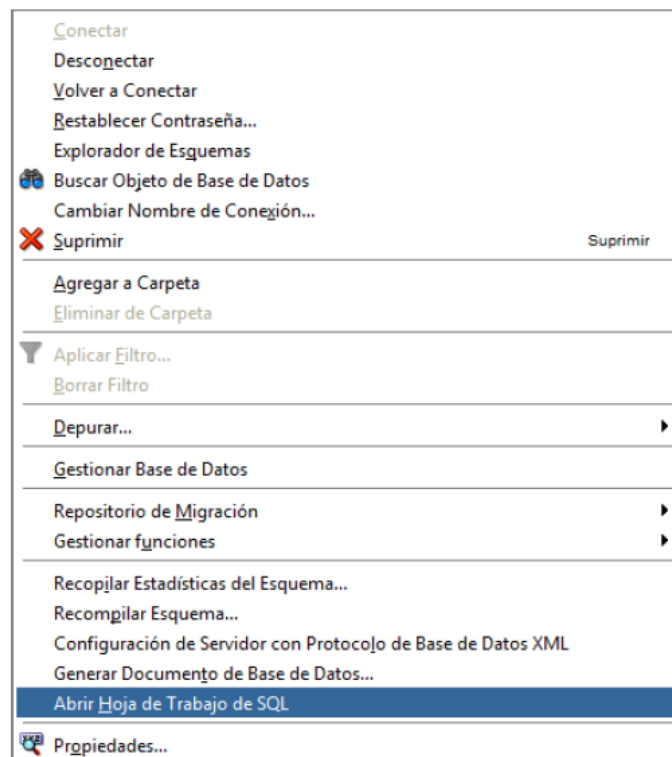
Puede dejar el resto de los parámetros tal y como aparecen por defecto.

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

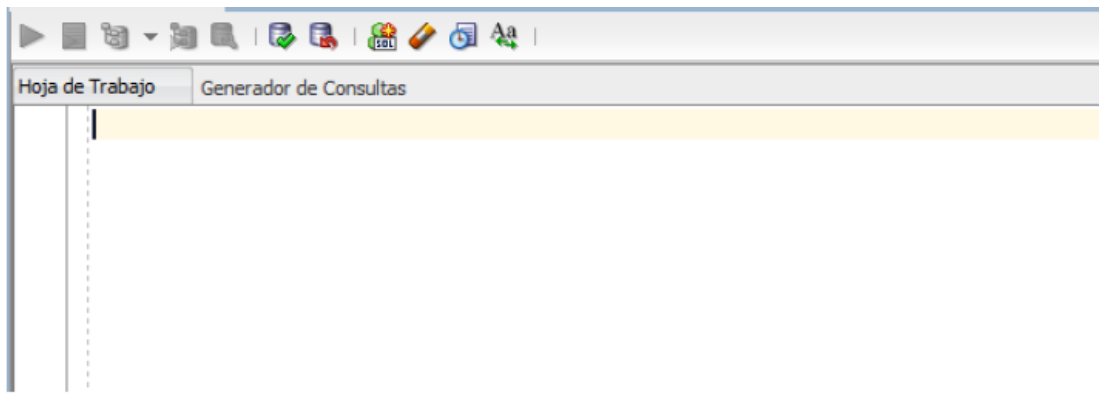


Luego, haga clic en el botón **Guardar**, y luego clic en el botón **Conectar**. El nombre de la conexión creada (**LAB3_DIRIGIDA**) aparecerá en el panel de Conexiones.

Haga clic derecho sobre el nombre de la conexión **LAB3_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_LAB3_01_dirigida_drops.sql
- INF246_2020-1_LAB3_02_dirigida_modelo.sql
- INF246_2020-1_LAB3_03_dirigida_inserts.sql

En la tabla SB_PRESTAMO vamos a manejar nuevo concepto llamado Tipo de Préstamo con los valores 'P' (Personal) y 'C' (Compra de Deuda) para ello debe realizar lo siguiente:

1. Agregar la columna Tipo de tipo VARCHAR2(1) a la tabla SB_PRESTAMO y asignar el valor por defecto que es el valor 'P'.

```
ALTER TABLE SB_PRESTAMO ADD TIPO VARCHAR2(1) default 'P';
```

Con esto se obtiene la siguiente salida:

```
Table SB_PRESTAMO alterado.
```

2. Agregar la columna Tipo_Agencia de tipo VARCHAR2(1) a la tabla SB_PRESTAMO y asignar el valor por defecto que es el valor 'N' (Agencia Normal). Hay que considerar que los valores válidos para este campo son 'N' (Agencia Normal) y 'C' (Agencia de Cobranzas).

```
ALTER TABLE SB_AGENCIA ADD TIPO VARCHAR2(1) default 'N';
```

Con esto se obtiene la siguiente salida:

```
Table SB_AGENCIA alterado.
```

3. Actualizaremos la Agencia Las Musas para que sea del tipo 'C' (Agencia de Cobranzas).

```
UPDATE SB_AGENCIA SET TIPO='C' WHERE ID_AGENCIA=6;
```

Con esto se obtiene la siguiente salida:

```
1 fila actualizadas.
```

4. Actualizaremos los prestamos asociados a la Agencia Las Musas para que sean del tipo 'C' (Compra de Deuda).

```
UPDATE SB_PRESTAMO SET TIPO='C' WHERE ID_AGENCIA=6;
```

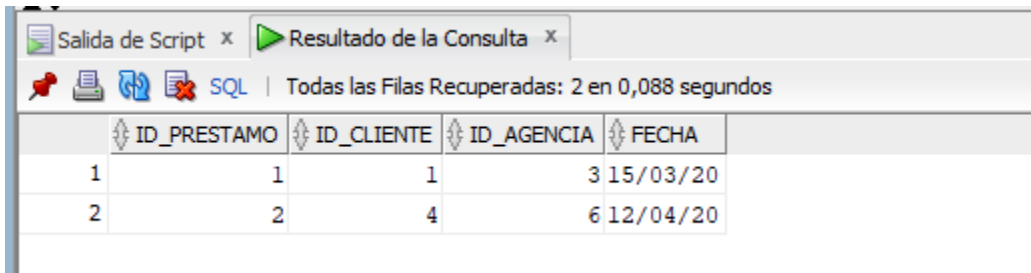
Con esto se obtiene la siguiente salida:

```
1 fila actualizadas.
```

5. Mostrar identificador del préstamo, identificador del cliente, identificador de la agencia y la fecha del préstamo de todos los préstamos que se han realizado entre enero y mayo del año actual (2020). Usar la función de fecha LAST_DAY.

```
SELECT
    pr.id_prestamo, pr.id_cliente, pr.id_agencia, pr.fecha
FROM
    SB_PRESTAMO PR
WHERE
    pr.fecha >= to_date('01/01/2020','dd/mm/yyyy')
    and
    pr.fecha <=LAST_DAY('01/05/2020');
```

La consulta debería mostrarse de la siguiente manera:

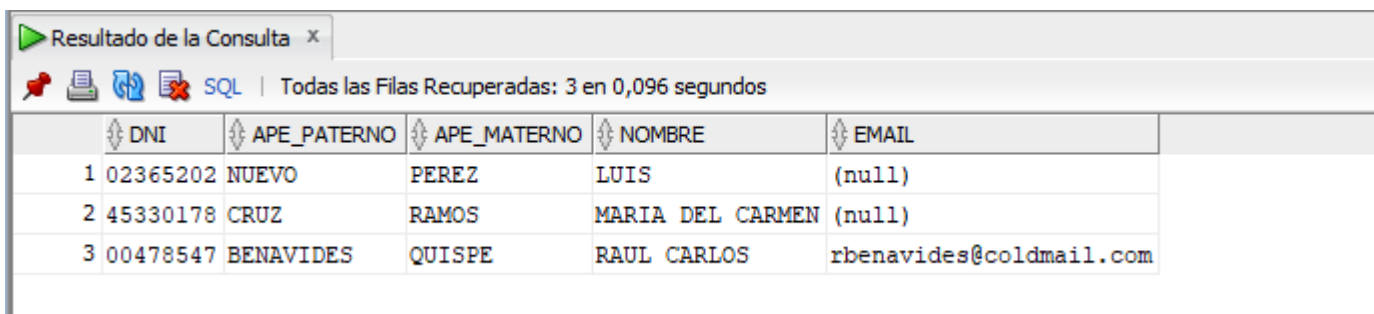


The screenshot shows a database application window with two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, displaying a table with 4 columns: ID_PRESTAMO, ID_CLIENTE, ID_AGENCIA, and FECHA. The table contains two rows of data. The status bar indicates 'Todas las Filas Recuperadas: 2 en 0,088 segundos'.

ID_PRESTAMO	ID_CLIENTE	ID_AGENCIA	FECHA
1	1	1	3 15/03/20
2	2	4	6 12/04/20

6. Mostrar DNI, apellido paterno, apellido materno, nombre y email de todos los clientes que no han realizado préstamos entre enero y mayo del año actual (2020) ordenado alfabéticamente por apellido paterno.

```
SELECT
    cli.dni, cli.ape_paterno, cli.ape_materno, cli.nombre, cli.email
FROM
    SB_CLIENTE cli
WHERE
    cli.id_cliente NOT IN (
        SELECT
            pr.id_cliente
        FROM
            SB_PRESTAMO PR
        WHERE
            pr.fecha >= to_date('01/01/2020','dd/mm/yyyy')
            and
            pr.fecha <= LAST_DAY('01/05/2020')
    )
ORDER BY CLI.APE_PATERNO DESC
```



Resultado de la Consulta x

Todas las Filas Recuperadas: 3 en 0,096 segundos

	DNI	APE_PATERNO	APE_MATERNO	NOMBRE	EMAIL
1	02365202	NUEVO	PEREZ	LUIS	(null)
2	45330178	CRUZ	RAMOS	MARIA DEL CARMEN	(null)
3	00478547	BENAVIDES	QUISPE	RAUL CARLOS	rbenavides@coldmail.com

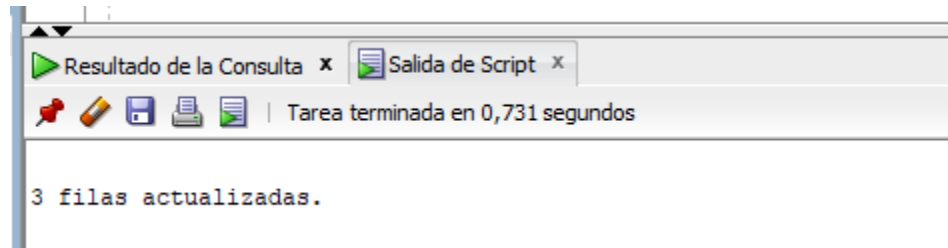
7. Usando la consulta del punto 6 actualizar la columna EMAIL con el valor de null de la tabla SB_CLIENTE.

```
UPDATE
    SB_CLIENTE cli
SET
    cli.EMAIL = NULL
WHERE
    cli.id_cliente NOT IN (
        SELECT
```

```

pr.id_cliente
FROM
  SB_PRESTAMO PR
WHERE
  pr.fecha >= to_date('01/01/2020','dd/mm/yyyy')
  and
  pr.fecha <= LAST_DAY('01/05/2020')
)

```



8. Agregar la columna TIPO de tipo VARCHAR2(1) a la tabla SB_TARJETA y asignar el valor por defecto 'C'. Hay que considerar que los valores que puede tener este nuevo campo son 'C' (Clásica) y 'O' (Oro)

```
ALTER TABLE SB_TARJETA ADD TIPO VARCHAR2(1) default 'C';
```

Table SB_TARJETA alterado.

9. Mostrar el identificador de la tarjeta, el número de la tarjeta y el identificador del cliente de todas las tarjetas que pertenecen a clientes del género Femenino.

```

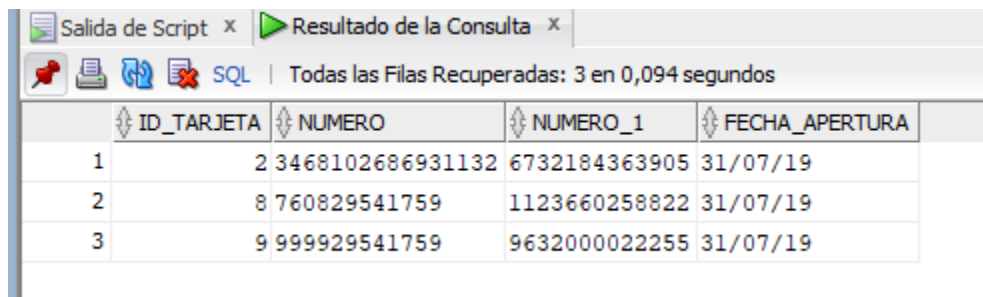
SELECT TAR.ID_TARJETA, TAR.NUMERO, CLI.ID_CLIENTE
FROM SB_CLIENTE CLI, SB_TARJETA TAR
WHERE CLI.ID_CLIENTE = TAR.ID_CLIENTE AND
  CLI.SEXO = 'F'

```

ID_TARJETA	NUMERO	ID_CLIENTE
1	1 455799991478	1
2	4 3468387283452460	1
3	5 1700786329	1
4	3 8530786586	3

10. Mostrar el identificador de la tarjeta, el número de la tarjeta, el número de cuenta y la fecha de apertura de la cuenta para todas las cuentas en dólares que fueron abiertas entre enero y diciembre del año 2019.

```
SELECT TAR.ID_TARJETA, TAR.NUMERO, CUE.NUMERO, CUE.FECHA_APERTURA
FROM SB_CUENTA CUE, SB_TARJETA_CUENTA TC, SB_TARJETA TAR
WHERE CUE.ID_CUENTA = TC.ID_CUENTA AND
      TC.ID_TARJETA = TAR.ID_TARJETA AND
      CUE.MONEDA = 'D' AND
      CUE.FECHA_APERTURA >= '01/01/2019' AND
      CUE.FECHA_APERTURA <= LAST_DAY('01/12/2019');
```



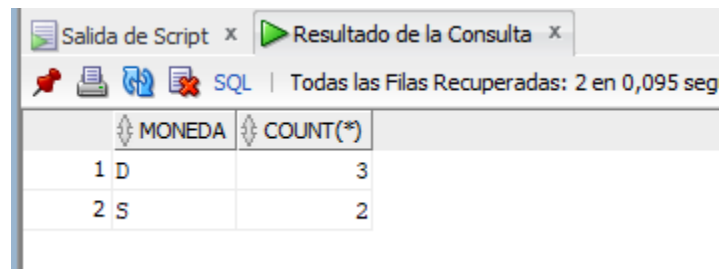
Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 3 en 0,094 segundos

	ID_TARJETA	NUMERO	NUMERO_1	FECHA_APERTURA
1	2	3468102686931132	6732184363905	31/07/19
2	8	760829541759	1123660258822	31/07/19
3	9	999929541759	9632000022255	31/07/19

11. Mostrar el tipo de moneda de la cuenta y la cantidad de tarjetas que poseen las cuentas que fueron abiertas entre enero y diciembre del año 2019.

```
SELECT CUE.MONEDA, COUNT(*)
FROM SB_CUENTA CUE, SB_TARJETA_CUENTA TC, SB_TARJETA TAR
WHERE CUE.ID_CUENTA = TC.ID_CUENTA AND
      TC.ID_TARJETA = TAR.ID_TARJETA AND
      CUE.FECHA_APERTURA >= '01/01/2019' AND
      CUE.FECHA_APERTURA <= LAST_DAY('01/12/2019')
GROUP BY CUE.MONEDA;
```



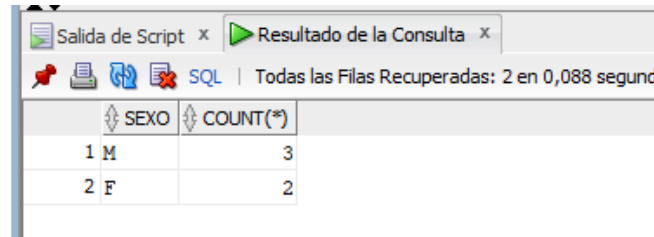
Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,095 seg

	MONEDA	COUNT(*)
1	D	3
2	S	2

12. Mostrar la cantidad de clientes que existen por cada género (Femenino y Masculino).

```
SELECT CLI.SEXO, COUNT(*)  
FROM SB_CLIENTE CLI  
GROUP BY CLI.SEXO
```



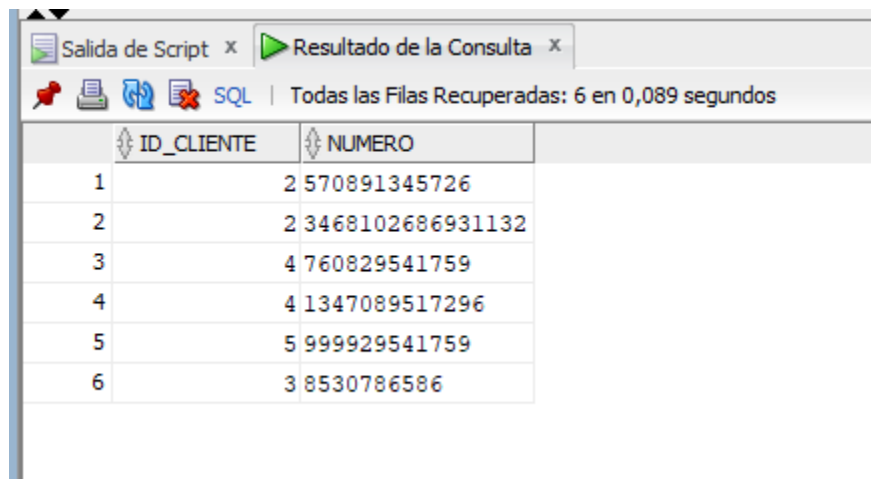
Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0,088 segundos

SEXO	COUNT(*)
1 M	3
2 F	2

13. Mostrar el identificador del cliente y el número de tarjeta asociado a los clientes que tienen más de una tarjeta.

```
SELECT TAR.ID_CLIENTE, TAR.NUMERO  
FROM SB_TARJETA TAR  
WHERE TAR.ID_CLIENTE IN (  
    SELECT TARJETA.ID_CLIENTE  
    FROM SB_TARJETA TARJETA  
    GROUP BY TARJETA.ID_CLIENTE  
    HAVING TARJETA.ID_CLIENTE >1);
```



Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 6 en 0,089 segundos

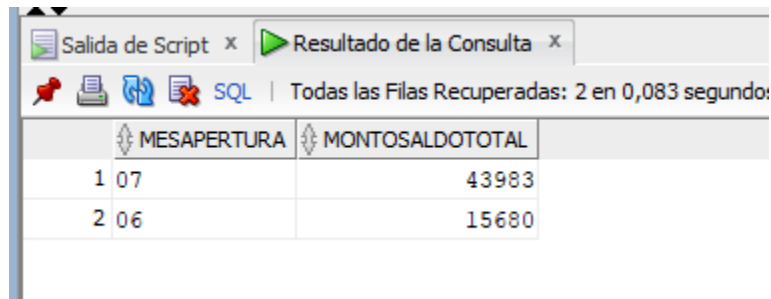
ID_CLIENTE	NUMERO
1	2 570891345726
2	2 3468102686931132
3	4 760829541759
4	4 1347089517296
5	5 999929541759
6	3 8530786586

14. Mostrar el monto total de los saldos de las cuentas por cada mes de acuerdo con el mes de su fecha de apertura del año 2019 ordenado por saldo de mayor a menor. Usar la función de texto TO_CHAR.


```

SELECT
    TO_CHAR(CUE.FECHA_APERTURA, 'MM') mesApertura, sum(CUE.SALDO) montoSaldoTotal
FROM
    SB_CUENTA CUE
WHERE
    CUE.FECHA_APERTURA >= '01/01/2019' AND
    CUE.FECHA_APERTURA <= LAST_DAY('01/12/2019')
GROUP BY TO_CHAR(CUE.FECHA_APERTURA, 'MM')
ORDER BY SUM(CUE.SALDO) DESC;

```



MESAPERTURA	MONTOSALDOTOTAL
1 07	43983
2 06	15680


15. Mostrar la cantidad total de agencias (recuerde que en una agencia se pudo abrir más de una cuenta) y la cantidad total de cuentas aperturadas en cada mes entre enero a diciembre del año 2019 ordenado por la cantidad total de cuentas de mayor a menor. Usar la función de agregación COUNT con DISTINCT

```

SELECT
    TO_CHAR(CUE.FECHA_APERTURA, 'MM') mesApertura, COUNT(DISTINCT CUE.ID_AGENCIA)
cantidadTotalAgencias,
    COUNT(CUE.ID_CUENTA) cantidadTotalCuentas
FROM
    SB_CUENTA CUE
WHERE
    CUE.FECHA_APERTURA >= '01/01/2019' AND
    CUE.FECHA_APERTURA <= LAST_DAY('01/12/2019')
GROUP BY TO_CHAR(CUE.FECHA_APERTURA, 'MM')
ORDER BY COUNT(CUE.ID_CUENTA) DESC;

```

Salida de Script x Resultado de la Co... x

 | Todas las Filas Recuperadas: 2 en 0,127 segundos

	MESAPERTURA	CANTIDADTOTALAGENCIAS	CANTIDADTOTALCUENTAS
1	07	1	4
2	06	1	1

16. Mostrar la cantidad de cuentas y la suma de saldos de las cuentas por cada cliente y mes de fecha de apertura de cuenta realizados durante el año 2019 ordenados por la suma total del saldo de forma descendente.

```

SELECT TO_CHAR(CUE.FECHA_APERTURA, 'MM'),
       CLI.ID_CLIENTE,
       SUM(CUE.SALDO),
       COUNT(CUE.ID_CUENTA)
FROM SB_CLIENTE CLI,
     SB_CLIENTE_CUENTA CLICUE,
     SB_CUENTA CUE
WHERE
  CLI.ID_CLIENTE = CLICUE.ID_CLIENTE AND
  CLICUE.ID_CUENTA = CUE.ID_CUENTA AND
  CUE.FECHA_APERTURA >= '01/01/2019' AND
  CUE.FECHA_APERTURA <= LAST_DAY('01/12/2019')
GROUP BY
  TO_CHAR(CUE.FECHA_APERTURA, 'MM'),
  CLI.ID_CLIENTE
ORDER BY
  SUM(CUE.SALDO) DESC;

```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 5 en 0,09 segundos

	TO_CHAR(CUE.FECHA_APERTURA,'MM')	ID_CLIENTE	SUM(CUE.SALDO)	COUNT(CUE.ID_CUENTA)
1	07	1	28827	1
2	07	4	28827	1
3	06	2	15680	1
4	07	5	13500	2
5	07	3	1656	1

21 de mayo del 2020

DA - LCH