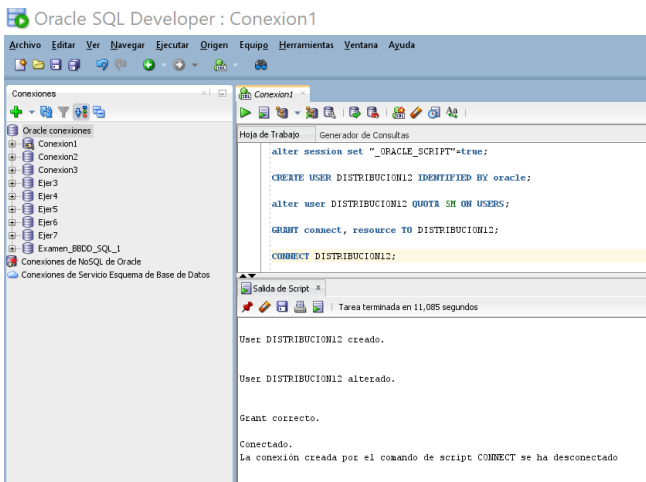


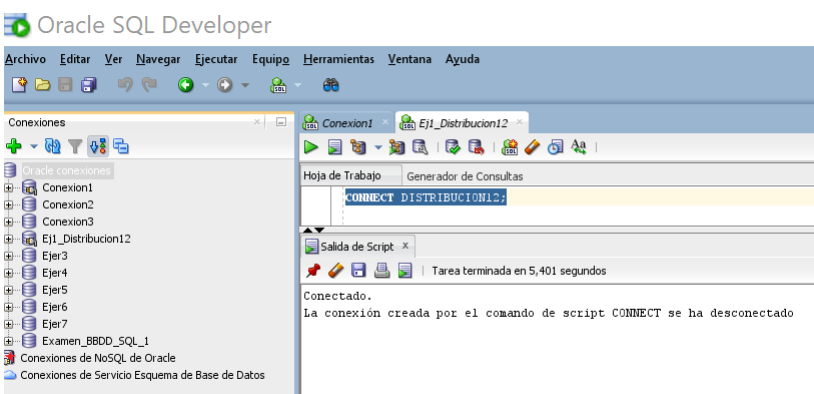
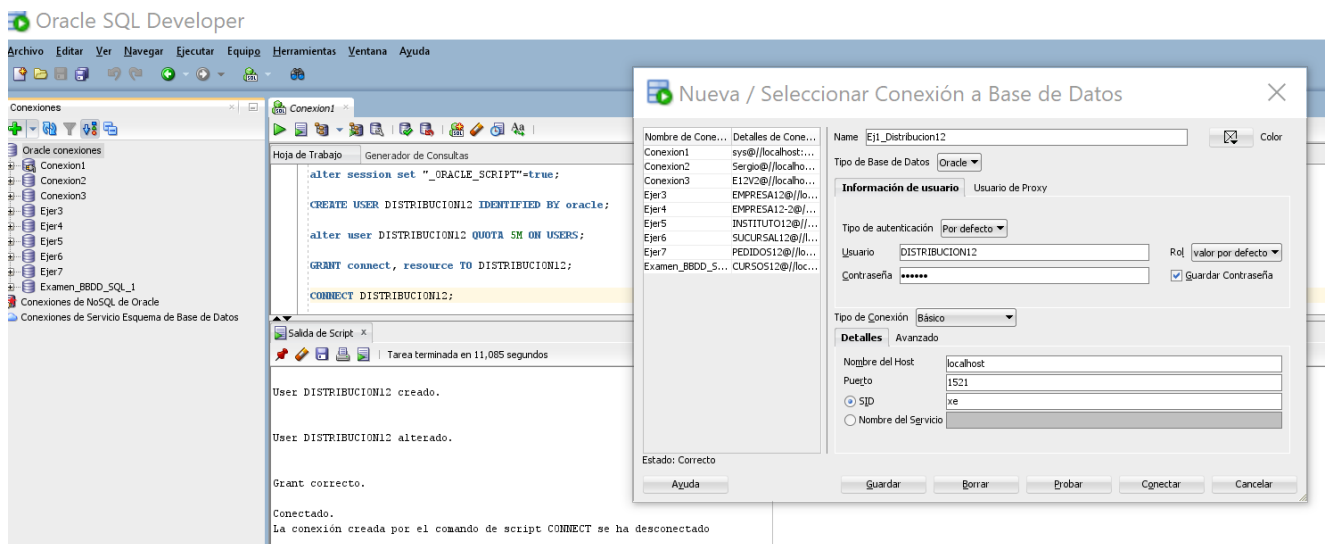
# Base de datos de una distribuidora

Uno de nuestros clientes se dedica a la distribución de distintos productos y nos ha pedido la creación de una base de datos, la carga de datos. Para ello, sigue las siguientes instrucciones:

1. Crea el usuario “distribucion” con contraseña “distribución” y una cuota de 5 Megabytes en el tablespace “USERS”.
2. Concédale permiso de acceso y utilización de los recursos.
3. Conéctate al usuario “distribucion” desde SQLPLUS.



Aquí me equivoco y conecto a DISTRIBUCION12 en SYS porque lo hago desde su ventana de Conexion1 que es la conexión por defecto de SYS



Para arreglar esto, creamos una nueva conexión como siempre con la interfaz de Developer, y desde esa nueva ventana de conexión, conectamos con DISTRIBUCION12

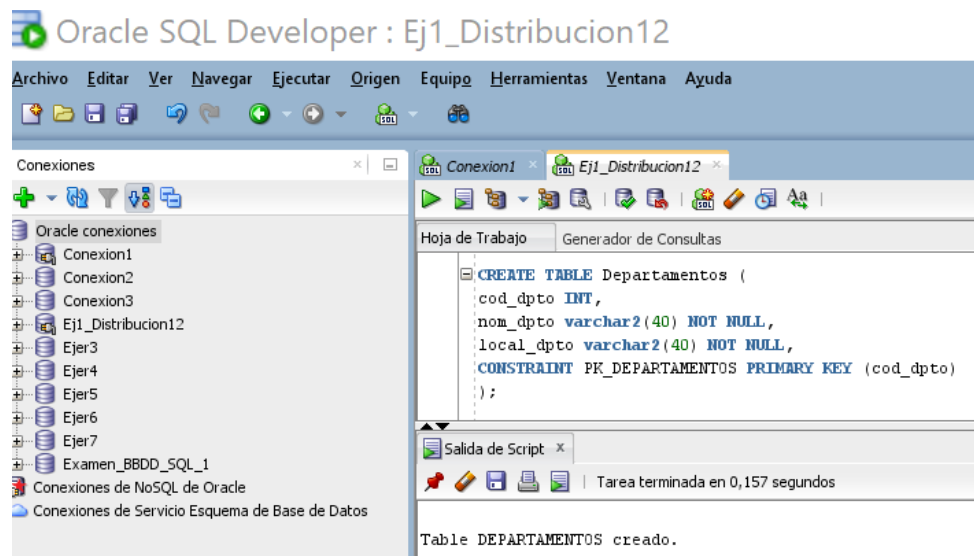
## 4. Crea la tabla “departamentos” con la siguiente especificación:

Campo	Tipo de dato	Restricciones
COD_DEPARTAMENTO	INT	Clave primaria
NOMBRE	VARCHAR2, longitud 40	No puede ser nulo
LOCALIDAD	VARCHAR2, longitud 40	No puede ser nulo

```

CREATE TABLE Departamentos (
cod_dpto INT,
nom_dpto varchar2(40) NOT NULL,
local_dpto varchar2(40) NOT NULL,
CONSTRAINT PK_DEPARTAMENTOS PRIMARY KEY (cod_dpto)
);

```



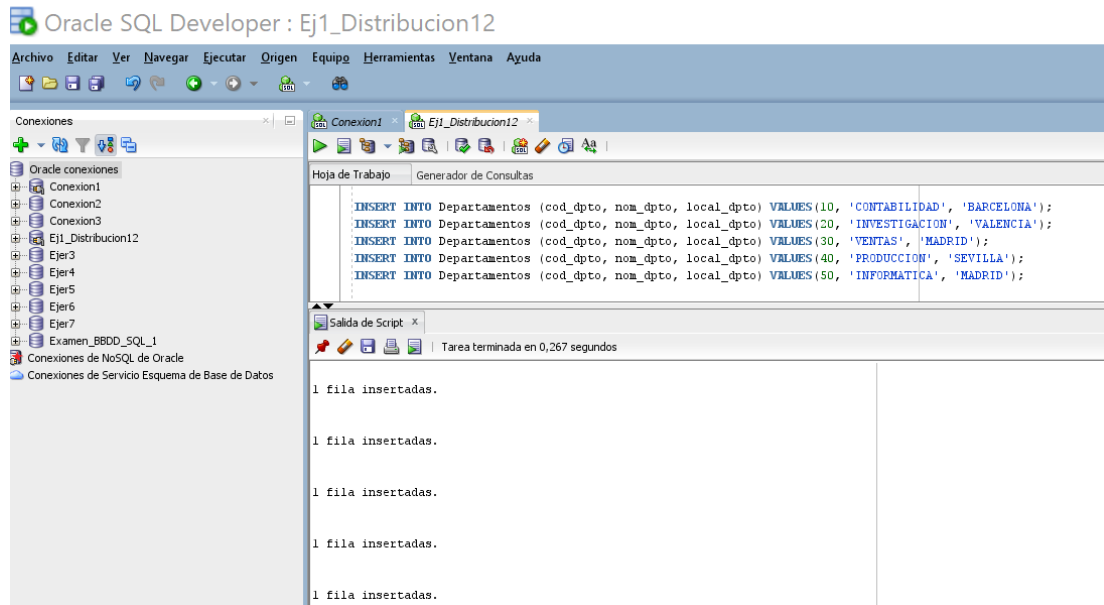
## 5. Inserta los siguientes datos en la tabla departamentos:

COD_DEPARTAMENTO	NOMBRE	LOCALIDAD
10	CONTABILIDAD	BARCELONA
20	INVESTIGACION	VALENCIA
30	VENTAS	MADRID
40	PRODUCCION	SEVILLA
50	INFORMATICA	MADRID

```

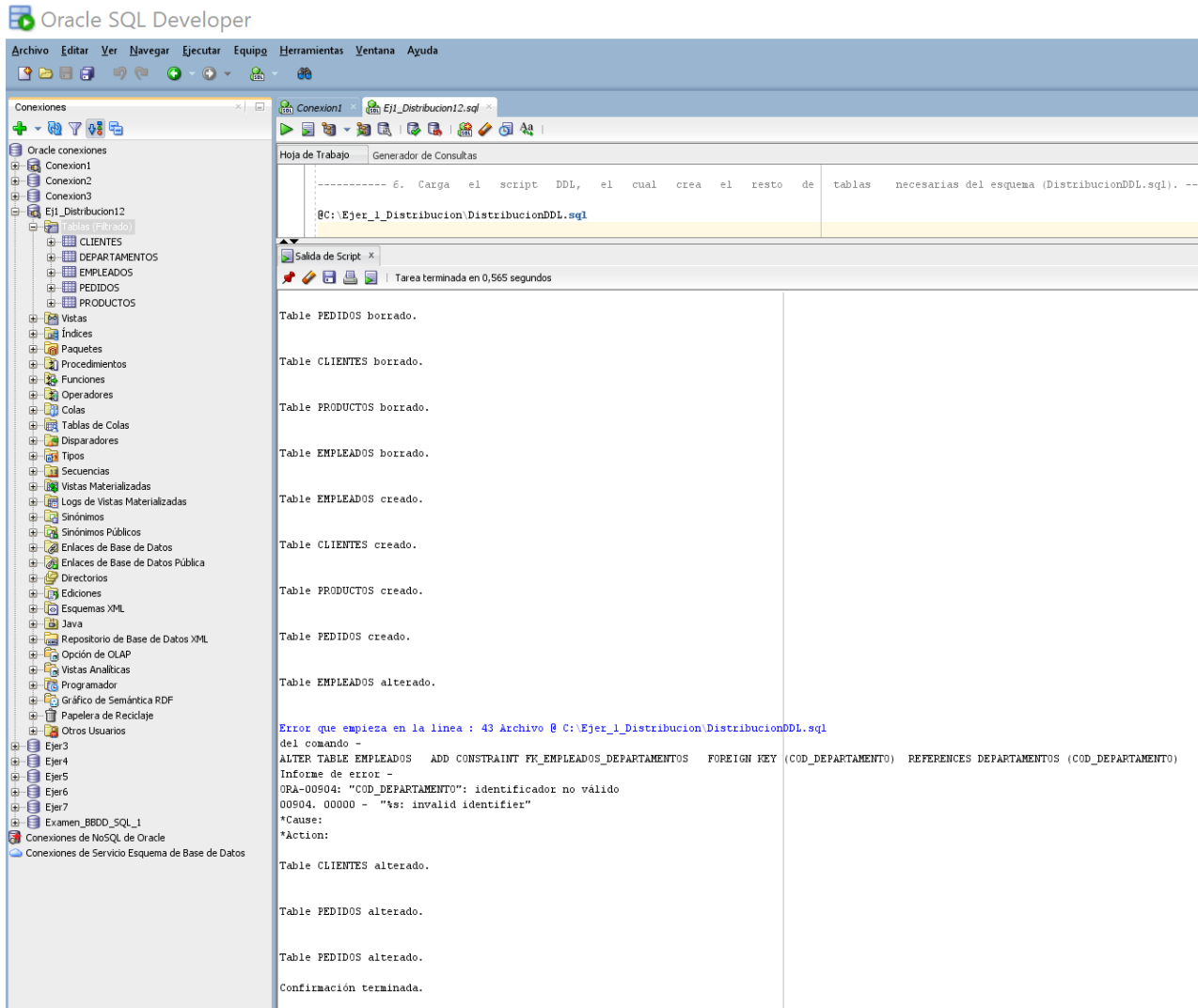
INSERT INTO Departamentos (cod_dpto, nom_dpto, local_dpto) VALUES(10,
'CONTABILIDAD', 'BARCELONA');
INSERT INTO Departamentos (cod_dpto, nom_dpto, local_dpto) VALUES(20,
'INVESTIGACION', 'VALENCIA');
INSERT INTO Departamentos (cod_dpto, nom_dpto, local_dpto) VALUES(30, 'VENTAS',
'MADRID');
INSERT INTO Departamentos (cod_dpto, nom_dpto, local_dpto) VALUES(40,
'PRODUCCION', 'SEVILLA');
INSERT INTO Departamentos (cod_dpto, nom_dpto, local_dpto) VALUES(50,
'INFORMATICA', 'MADRID');

```

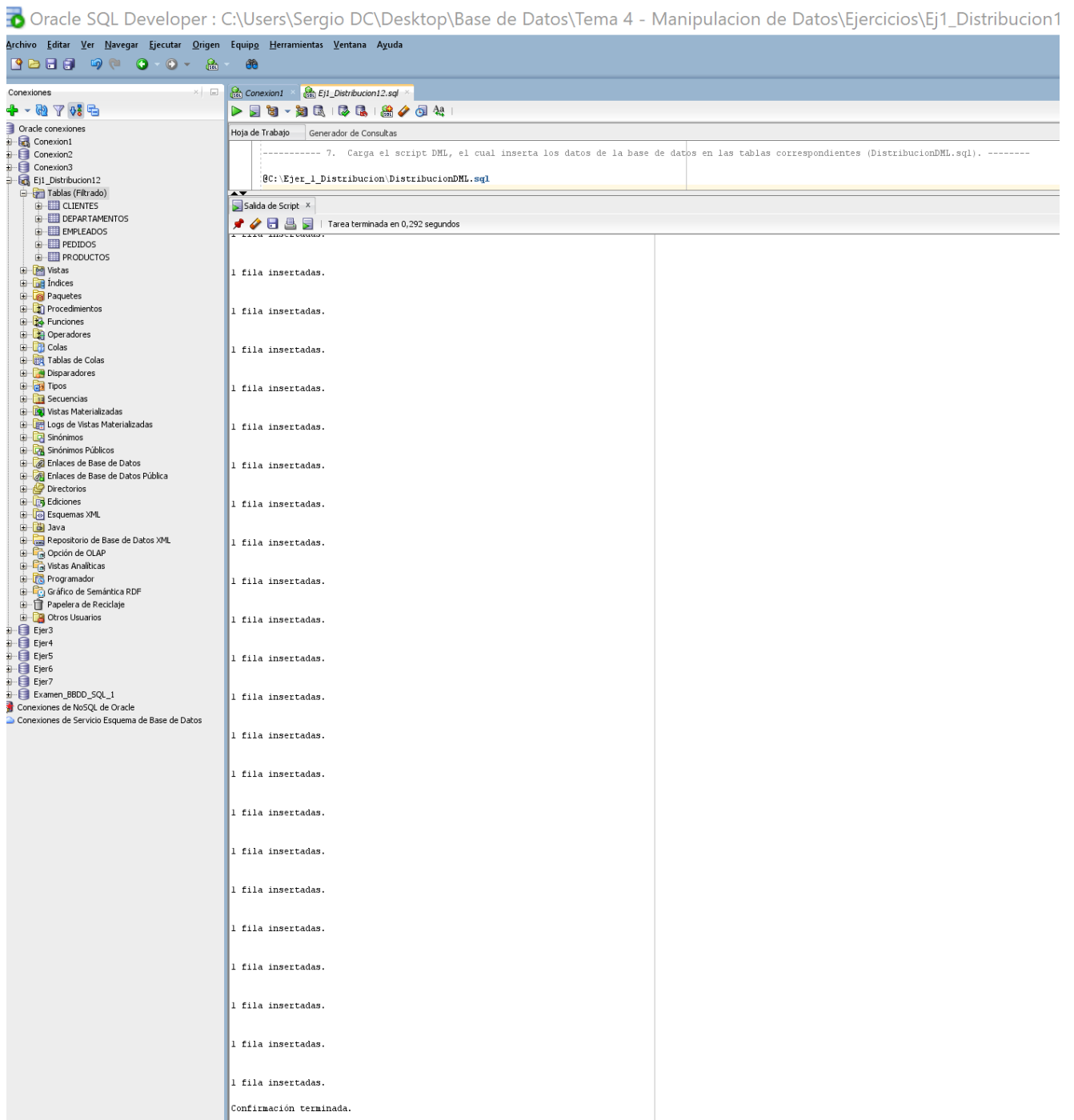


6. Carga el script DDL, el cual crea el resto de tablas necesarias del esquema (DistribucionDDL.sql).

@C:\Ejer\_1\_Distribucion\DistribucionDDL.sql



7. Carga el script DML, el cual inserta los datos de la base de datos en las tablas correspondientes (DistribucionDML.sql).



A continuación, la empresa cliente necesita algunos informes sobre su base de datos, los cuales vamos a proceder a realizar mediante consultas (para ello usaremos SQL DEVELOPER). Concretamente necesitamos:

1. Un listado que contenga el código de cliente y nombre de todos los clientes que no se encuentren en 'BARCELONA', ordenados por nombre, alfabéticamente.

**SELECT COD\_CLIENTE, NOMBRE FROM CLIENTES WHERE LOCALIDAD != 'BARCELONA' ORDER BY NOMBRE;**  
(lo del != es equivalente al NOT LIKE)



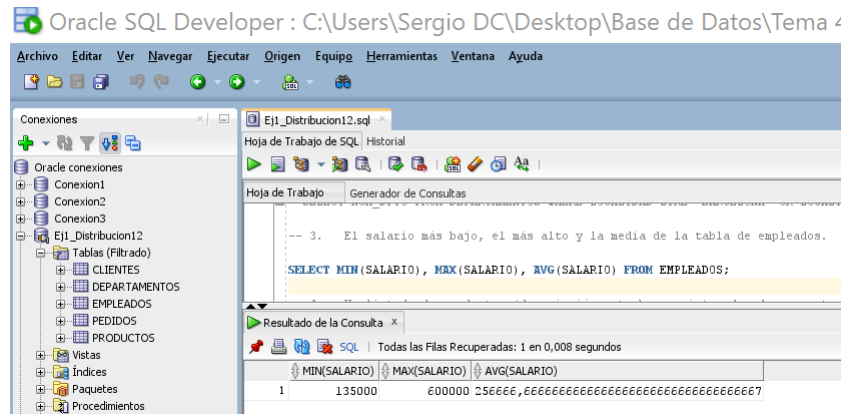
2. Un listado que contenga el nombre de los departamentos que se encuentran en la siguiente lista (se deben utilizar listas) de ciudades: Madrid, Barcelona

**SELECT NOM\_DPTO, LOCAL\_DPTO FROM DEPARTAMENTOS WHERE LOCAL\_DPTO IN ('MADRID', 'BARCELONA')**



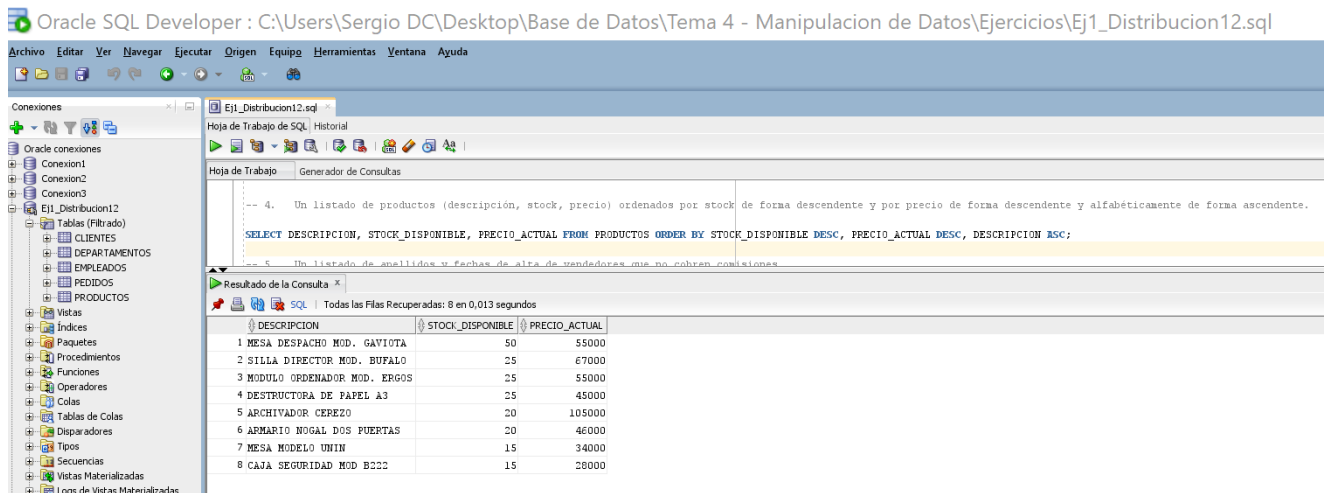
3. El salario más bajo, el más alto y la media de la tabla de empleados.

```
SELECT MIN(SALARIO), MAX(SALARIO), AVG(SALARIO) FROM EMPLEADOS;
```



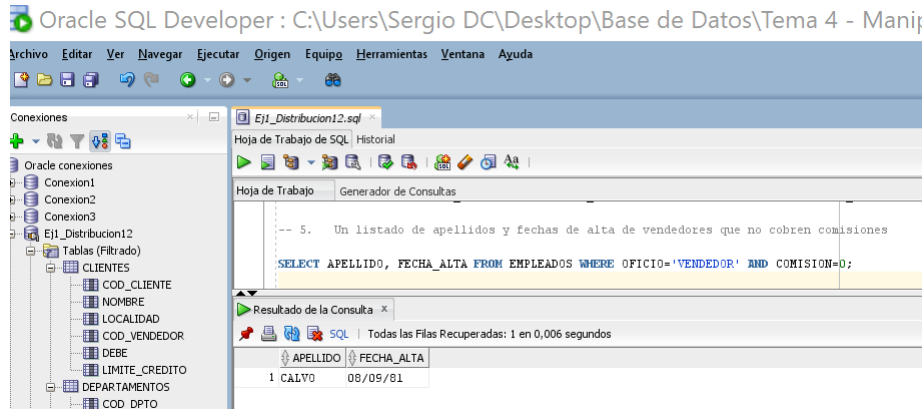
4. Un listado de productos (descripción, stock, precio) ordenados por stock de forma descendente y por precio de forma descendente y alfabéticamente de forma ascendente.

```
SELECT DESCRIPCION, STOCK_DISPONIBLE, PRECIO_ACTUAL FROM  
PRODUCTOS ORDER BY STOCK_DISPONIBLE DESC, PRECIO_ACTUAL DESC,  
DESCRIPCION ASC;
```



### 5. Un listado de apellidos y fechas de alta de vendedores que no cobren comisiones

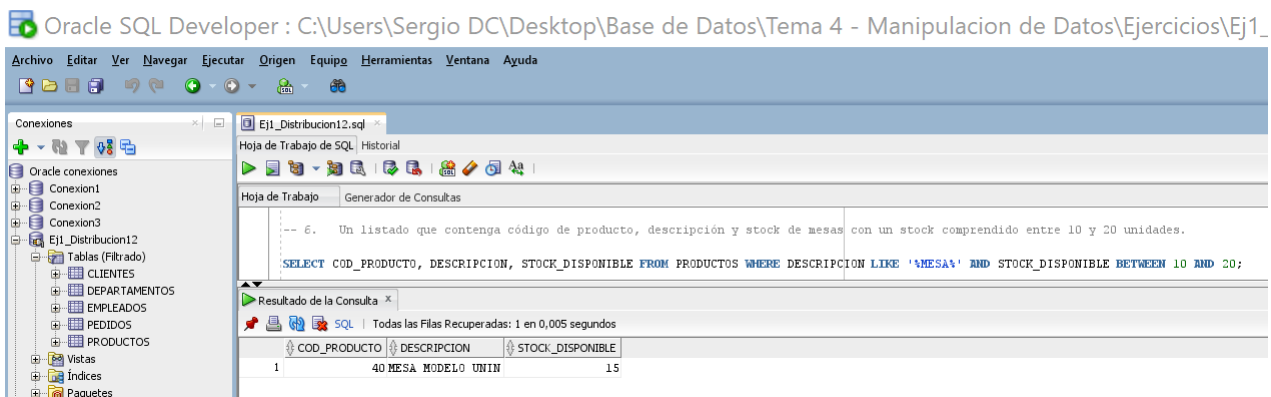
**SELECT APELLIDO, FECHA\_ALTA FROM EMPLEADOS WHERE OFICIO='VENDEDOR' AND COMISION=0;**



### 6. Un listado que contenga código de producto, descripción y stock de mesas con un stock comprendido entre 10 y 20 unidades.

**SELECT COD\_PRODUCTO, DESCRIPCION, STOCK\_DISPONIBLE FROM PRODUCTOS WHERE DESCRIPCION LIKE '%MESA%' AND STOCK\_DISPONIBLE BETWEEN 10 AND 20;**

**(SELECT COD\_PRODUCTO, DESCRIPCION, STOCK\_DISPONIBLE FROM PRODUCTOS WHERE DESCRIPCION LIKE '%MESA%' AND STOCK\_DISPONIBLE > 10 AND STOCK\_DISPONIBLE < 20;)**

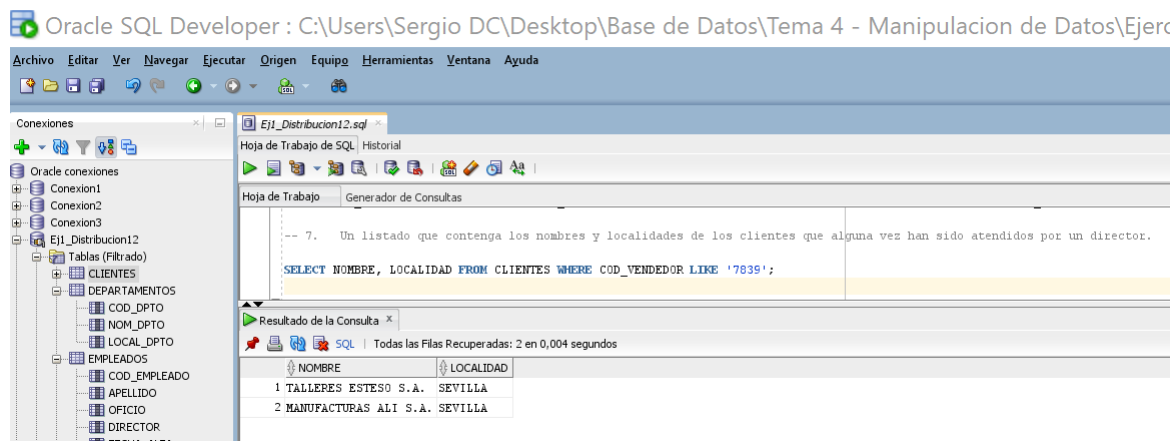




7. Un listado que contenga los nombres y localidades de los clientes que alguna vez han sido atendidos por un director.

*Nota: Aquí el truco está en abrir la tabla de EMPLEADOS y copiar el código del campo de quien ponga que es “director”, y después irse a la tabla de CLIENTES y pegar el dato en la casilla que queramos del cliente que queramos en su campo de “cod\_vendedor” (en mi caso, a los dos de Sevilla)*

**SELECT NOMBRE, LOCALIDAD FROM CLIENTES WHERE COD\_VENDEDOR LIKE '7839';**



8. Listado con el nombre y localidad de los clientes que tienen pedidos con menos de 3 unidades en el mes de noviembre de 1999.

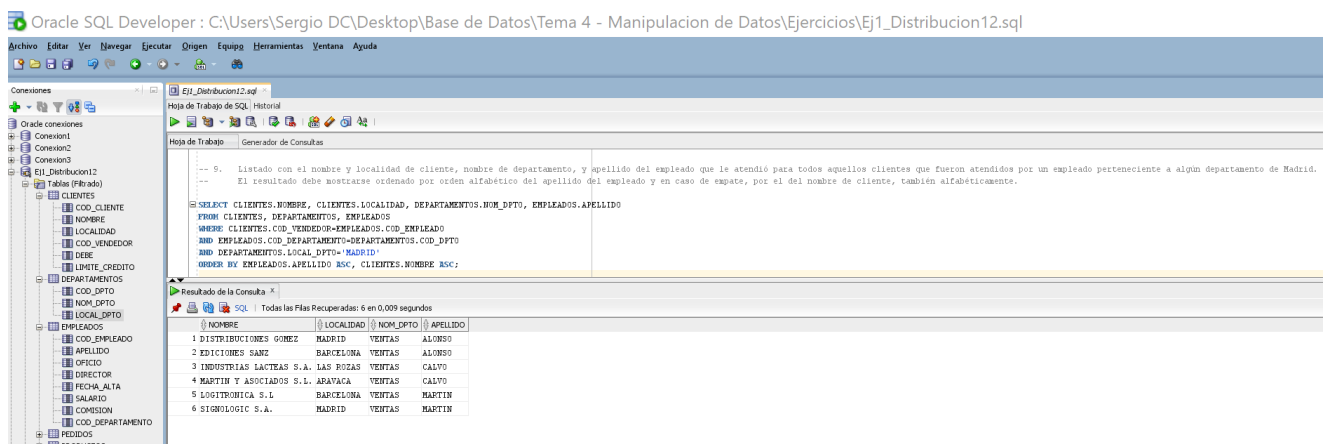
**ALTER SESSION SET nls\_date\_format = 'YYYY-MM-DD';**

**SELECT CLIENTES.NOMBRE, CLIENTES.LOCALIDAD  
FROM CLIENTES, PEDIDOS  
WHERE CLIENTES.COD\_CLIENTE=PEDIDOS.COD\_CLIENTE  
AND PEDIDOS.UNIDADES < 3  
AND PEDIDOS.FECHA\_PEDIDO BETWEEN '1999-11-01' AND '1999-11-30';**



9. Listado con el nombre y localidad de cliente, nombre de departamento, y apellido del empleado que le atendió para todos aquellos clientes que fueron atendidos por un empleado perteneciente a algún departamento de Madrid. El resultado debe mostrarse ordenado por orden alfabético del apellido del empleado y en caso de empate, por el del nombre de cliente, también alfabéticamente.

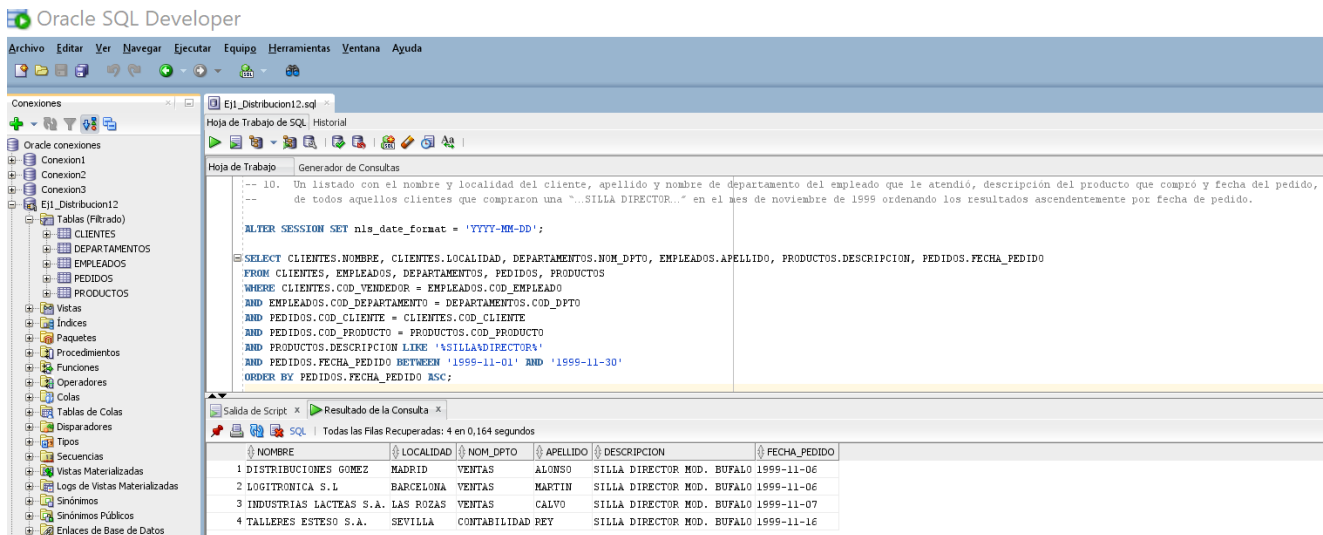
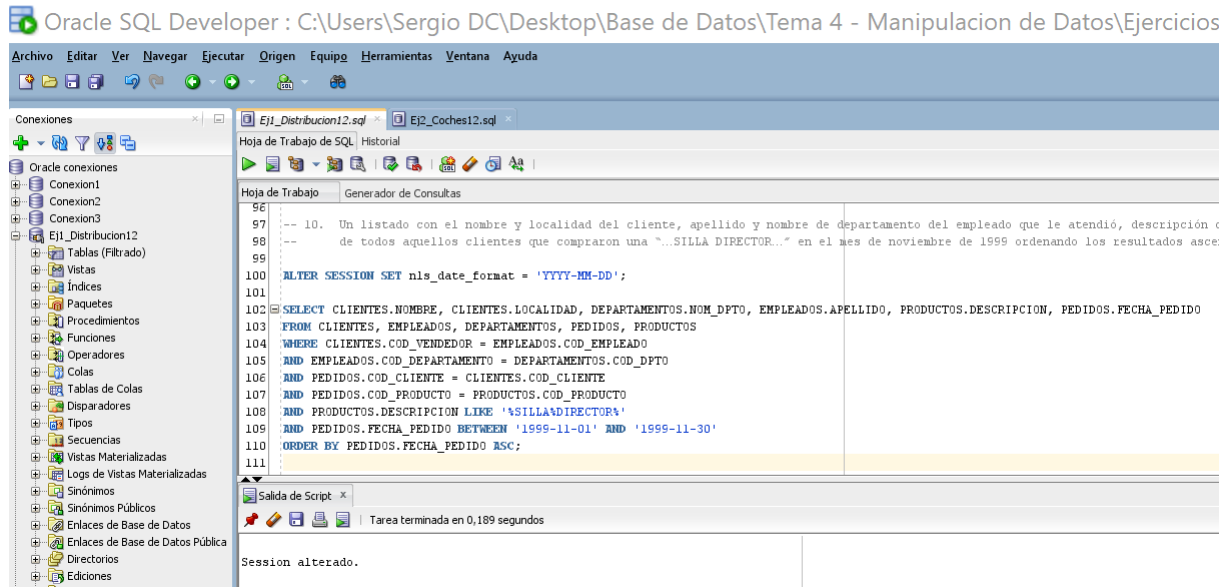
```
SELECT CLIENTES.NOMBRE, CLIENTES.LOCALIDAD, DEPARTAMENTOS.NOM_DPTO,
EMPLEADOS.APELLIDO
FROM CLIENTES, DEPARTAMENTOS, EMPLEADOS
WHERE CLIENTES.COD_VENDEDOR=EMPLEADOS.COD_EMPLEADO
AND EMPLEADOS.COD_DEPARTAMENTO=DEPARTAMENTOS.COD_DPTO
AND DEPARTAMENTOS.LOCAL_DPTO='MADRID'
ORDER BY EMPLEADOS.APELLIDO ASC, CLIENTES.NOMBRE ASC;
```



10. Un listado con el nombre y localidad del cliente, apellido y nombre de departamento del empleado que le atendió, descripción del producto que compró y fecha del pedido, de todos aquellos clientes que compraron una "...SILLA DIRECTOR..." en el mes de noviembre de 1999 ordenando los resultados ascendentemente por fecha de pedido.

```
ALTER SESSION SET nls_date_format = 'YYYY-MM-DD';
```

```
SELECT CLIENTES.NOMBRE, CLIENTES.LOCALIDAD,
DEPARTAMENTOS.NOM_DPTO, EMPLEADOS.APELLIDO,
PRODUCTOS.DESCRIPCION, PEDIDOS.FECHA_PEDIDO
FROM CLIENTES, EMPLEADOS, DEPARTAMENTOS, PEDIDOS, PRODUCTOS
WHERE CLIENTES.COD_VENDEDOR = EMPLEADOS.COD_EMPLEADO
AND EMPLEADOS.COD_DEPARTAMENTO = DEPARTAMENTOS.COD_DPTO
AND PEDIDOS.COD_CLIENTE = CLIENTES.COD_CLIENTE
AND PEDIDOS.COD_PRODUCTO = PRODUCTOS.COD_PRODUCTO
AND PRODUCTOS.DESCRIPCION LIKE '%SILLA%DIRECTOR%'
AND PEDIDOS.FECHA_PEDIDO BETWEEN '1999-11-01' AND '1999-11-30'
ORDER BY PEDIDOS.FECHA_PEDIDO ASC;
```



Por último, el cliente nos informa de que va a necesitar algunos informes estadísticos para los que vamos a necesitar algunas sentencias más complejas. En concreto necesitamos:

11. Un listado con el nombre de la localidad y la suma del crédito que tienen sus clientes, agrupado por localidad.

**SELECT CLIENTES.LOCALIDAD, SUM(CLIENTES.LIMITE\_CREDITO) FROM CLIENTES GROUP BY CLIENTES.LOCALIDAD;**

Oracle SQL Developer : C:\Users\Sergio DC\Desktop\Base de Datos\Tema 4 - Manipulacion de Datos\E

Conexiones

Oracle conexiones

- Conexion1
- Conexion2
- Conexion3
- Ej1\_Distribucion12

Tablas (Filtrado)

- Clientes
- DEPARTAMENTOS
- EMPLEADOS
- PEDIDOS
- PRODUCTOS
- STOCK\_DISPONIBLE

Hoja de Trabajo de SQL

Hoja de Trabajo

Generador de Consultas

-- 11. Un listado con el nombre de la localidad y la suma del crédito que tienen sus clientes, agrupado por localidad.

**SELECT CLIENTES.LOCALIDAD, SUM(CLIENTES.LIMITE\_CREDITO) FROM CLIENTES GROUP BY CLIENTES.LOCALIDAD;**

Resultado de la Consulta

Todas las Filas Recuperadas: 5 en 0,207 segundos

LOCALIDAD	SUM(CLIENTES.LIMITE_CREDITO)
1 MADRID	1000000
2 SEVILLA	1000000
3 LAS ROZAS	1000000
4 BARCELONA	1000000
5 ARAVACA	1000000

12. Un listado que muestre nombre del departamento, oficio y media salarial, agrupado por departamento y oficio, que tengan asignado un salario superior a 200000 €, todo ello ordenado descendientemente por la media salarial.

**SELECT DEPARTAMENTOS.NOM\_DPTO, EMPLEADOS.OFICIO,  
AVG(EMPLEADOS.SALARIO)  
FROM DEPARTAMENTOS, EMPLEADOS  
WHERE EMPLEADOS.COD\_DEPARTAMENTO = DEPARTAMENTOS.COD\_DPTO  
GROUP BY DEPARTAMENTOS.NOM\_DPTO, EMPLEADOS.OFICIO  
HAVING AVG(EMPLEADOS.SALARIO) > 200000  
ORDER BY AVG(EMPLEADOS.SALARIO) DESC;**

Oracle SQL Developer : C:\Users\Sergio DC\Desktop\Base de Da

Conexiones

Oracle conexiones

- Conexion1
- Conexion2
- Conexion3
- Ej1\_Distribucion12

Tablas (Filtrado)

- Clientes
- DEPARTAMENTOS
- EMPLEADOS
- PEDIDOS
- PRODUCTOS
- STOCK\_DISPONIBLE

Hoja de Trabajo de SQL

Hoja de Trabajo

Generador de Consultas

-- 12. Un listado que muestre nombre del departamento, oficio

**SELECT DEPARTAMENTOS.NOM\_DPTO, EMPLEADOS.OFICIO,  
AVG(EMPLEADOS.SALARIO)  
FROM DEPARTAMENTOS, EMPLEADOS  
WHERE EMPLEADOS.COD\_DEPARTAMENTO = DEPARTAMENTOS.COD\_DPTO  
GROUP BY DEPARTAMENTOS.NOM\_DPTO, EMPLEADOS.OFICIO  
HAVING AVG(EMPLEADOS.SALARIO) > 200000  
ORDER BY AVG(EMPLEADOS.SALARIO) DESC;**

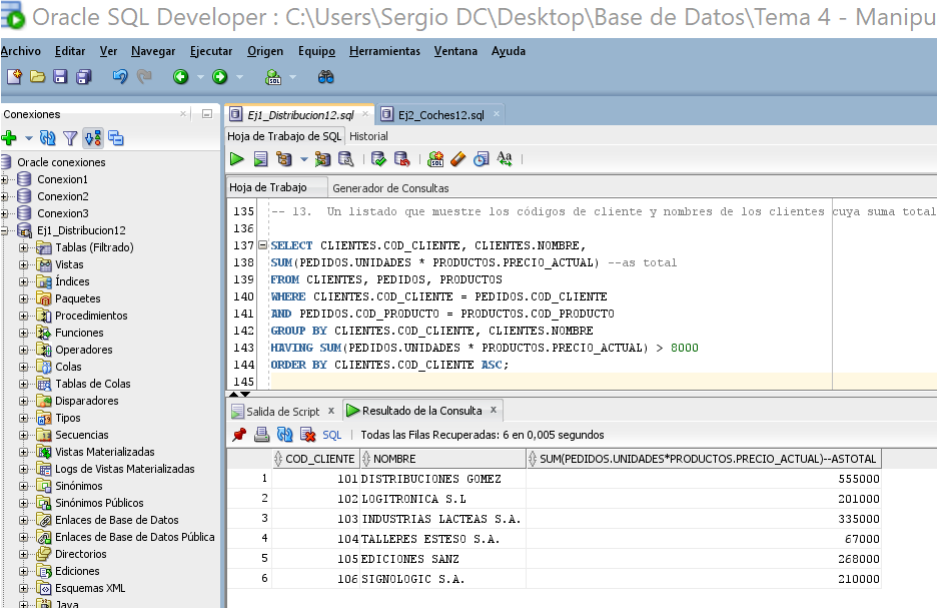
Resultado de la Consulta

Todas las Filas Recuperadas: 4 en 0,007 segundos

NOM_DPTO	OFICIO	AVG(EMPLEADOS.SALARIO)
1 CONTABILIDAD	PRESIDENTE	600000
2 VENTAS	DIRECTOR	385000
3 INVESTIGACION	ANALISTA	335000
4 CONTABILIDAD	DIRECTOR	245000

13. Un listado que muestre los códigos de cliente y nombres de los clientes cuya suma total de los pedidos (también hay que mostrarla) que han realizado sea mayor que 8.000.000€, ordenado ascendentemente por número de código de cliente.

```
SELECT CLIENTES.COD_CLIENTE, CLIENTES.NOMBRE,
SUM(PEDIDOS.UNIDADES * PRODUCTOS.PRECIO_ACTUAL) --as total
FROM CLIENTES, PEDIDOS, PRODUCTOS
WHERE CLIENTES.COD_CLIENTE = PEDIDOS.COD_CLIENTE
AND PEDIDOS.COD_PRODUCTO = PRODUCTOS.COD_PRODUCTO
GROUP BY CLIENTES.COD_CLIENTE, CLIENTES.NOMBRE
HAVING SUM(PEDIDOS.UNIDADES * PRODUCTOS.PRECIO_ACTUAL) > 8000
ORDER BY CLIENTES.COD_CLIENTE ASC;
```



Oracle SQL Developer : C:\Users\Sergio DC\Desktop\Base de Datos\Tema 4 - Manipu

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

- Conexion1
- Conexion2
- Conexion3
- Ej1\_Distribucion12
- Ej2\_Coches12

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```
-- 13. Un listado que muestre los códigos de cliente y nombres de los clientes cuya suma total
SELECT CLIENTES.COD_CLIENTE, CLIENTES.NOMBRE,
SUM(PEDIDOS.UNIDADES * PRODUCTOS.PRECIO_ACTUAL) --as total
FROM CLIENTES, PEDIDOS, PRODUCTOS
WHERE CLIENTES.COD_CLIENTE = PEDIDOS.COD_CLIENTE
AND PEDIDOS.COD_PRODUCTO = PRODUCTOS.COD_PRODUCTO
GROUP BY CLIENTES.COD_CLIENTE, CLIENTES.NOMBRE
HAVING SUM(PEDIDOS.UNIDADES * PRODUCTOS.PRECIO_ACTUAL) > 8000
ORDER BY CLIENTES.COD_CLIENTE ASC;
```

Salida de Script x Resultado de la Consulta x

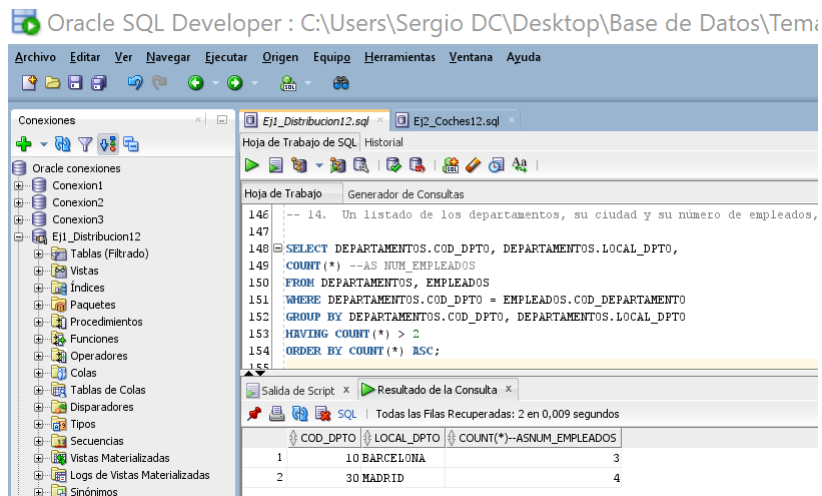
Todas las Filas Recuperadas: 6 en 0,005 segundos

COD_CLIENTE	NOMBRE	SUM(PEDIDOS.UNIDADES*PRODUCTOS.PRECIO_ACTUAL)--ASTOTAL
1	101 DISTRIBUCIONES GOMEZ	555000
2	102 LOGITRONICA S.L	201000
3	103 INDUSTRIAS LACTEAS S.A.	335000
4	104 TALLEPES ESTESO S.A.	67000
5	105 EDICIONES SANZ	268000
6	106 SIGNOLOGIC S.A.	210000

14. Un listado de los departamentos, su ciudad y su número de empleados, cuyo departamento tenga más de 2 empleados, ordenado por número de empleados ascendentemente.

```
SELECT DEPARTAMENTOS.COD_DPTO, DEPARTAMENTOS.LOCAL_DPTO,
COUNT(*) --AS NUM_EMPLEADOS
FROM DEPARTAMENTOS, EMPLEADOS
WHERE DEPARTAMENTOS.COD_DPTO = EMPLEADOS.COD_DEPARTAMENTO
GROUP BY DEPARTAMENTOS.COD_DPTO, DEPARTAMENTOS.LOCAL_DPTO
HAVING COUNT(*) > 2
ORDER BY COUNT(*) ASC;
```

*Nota: lo del asterisco (\*) es para contar todas las líneas de datos*



15. Un listado de los nombres de departamento y el mínimo salario de los departamentos en los que todos sus empleados cobren más de 135.000€, ordenado por salario mínimo descendente y por nombre de departamento ascendentemente.

```
SELECT DEPARTAMENTOS.NOM_DPTO,
MIN(EMPLEADOS.SALARIO) --as salario
FROM DEPARTAMENTOS, EMPLEADOS
WHERE EMPLEADOS.COD_DEPARTAMENTO = DEPARTAMENTOS.COD_DPTO
GROUP BY DEPARTAMENTOS.NOM_DPTO
HAVING MIN(EMPLEADOS.SALARIO) >= 135000
ORDER BY MIN(EMPLEADOS.SALARIO) DESC, DEPARTAMENTOS.NOM_DPTO ASC;
```

