

TAREA UT04 – BASES DE DATOS

ANTONIO SANZ PANS
IES MAESTRE DE CALATRAVA

Tabla de contenido

Apartado 0..... 2

Apartado 1..... 3

Apartado 2..... 4

 VERSIÓN 1 4

 VERSIÓN 2 5

Apartado 3..... 6

 VERSIÓN 1 6

 VERSIÓN 2 6

 VERSIÓN 3 7

Apartado 4..... 9

Apartado 5..... 10

Apartado 6..... 11

APARTADO 1

Listar por las ventas de medicamentos con receta realizadas por los médicos de cada centro de salud. Las columnas que visualizaremos son: población del médico, centro de salud del médico, nombre medicamento, nombre laboratorio, fecha venta, unidades, precio unitario, total venta (unidades*precio unitario). Sólo listaremos aquellos medicamentos cuyas ventas se hayan realizado entre las fechas: 01/10/2021 y 31/12/2021, y que el nombre de medicamento contenga las palabras: medicamento antibiótico en cualquier lugar. Se ordenará por población, dentro de ésta por centro salud, por nombre medicamento y por fecha venta.

CONSULTA SQL

```
SELECT
    m_medicos.poblacion ,
    m_medicos.centro_salud,
    m_medicamentos.nombre_med,
    m_laboratorios.nombre_lab,
    m_ventas_med.fecha_venta,
    m_ventas_med.unidades,
    m_medicamentos.precio_unit,
    (m_ventas_med.unidades * m_medicamentos.precio_unit) as
TOTAL_VENTA
FROM m_medicos
    INNER JOIN m_ventas_recetas ON m_ventas_recetas.dnim =
m_medicos.dnim
    INNER JOIN m_ventas_med ON m_ventas_recetas.id_venta =
m_ventas_med.id_venta
    INNER JOIN m_medicamentos ON m_ventas_med.id_med =
m_medicamentos.id_med
    INNER JOIN m_laboratorios ON m_medicamentos.id_lab =
m_laboratorios.id_lab
WHERE m_ventas_med.fecha_venta >= CAST('01/10/2021' AS date)
    AND m_ventas_med.fecha_venta <= CAST('31/12/2021' AS date)
    AND UPPER(m_medicamentos.nombre_med) LIKE UPPER('MEDICAMENTO
%ANTIBIOTICO%');
```

CAPTURA

The screenshot shows the Oracle SQL Developer interface. The top pane displays the SQL query, and the bottom pane shows the results of the query execution. The results are displayed in a table with the following columns: POBLACION, CENTRO_SALUD, NOMBRE_MED, NOMBRE LAB, FECHA_VENTA, UNIDADES, PRECIO_UNIT, and TOTAL_VENTA. The table contains three rows of data.

POBLACION	CENTRO_SALUD	NOMBRE_MED	NOMBRE LAB	FECHA_VENTA	UNIDADES	PRECIO_UNIT	TOTAL_VENTA	
1	Ciudad Real Centro Salud	Medicamento Antibiótico 3-COMPRIMIDOS	LABORATORIO HERBARIOS ROS	21/12/2021	2	9,5	19,0	
2	PUERTOLLANO CENTRO SALUD	Medicamento Antibiótico 1-COMPRIMIDOS	LABORATORIO REGIONAL BARCHENO	12/10/2021	3	12	36	
3	DANIEL	CENTRO SALUD 3	Medicamento Antibiótico 1-COMPRIMIDOS	LABORATORIO REGIONAL BARCHENO	12/12/2021	1	12	12

APARTADO 2

VERSIÓN 1

Listado del nombre de los medicamento y las cantidades en stock de aquellos medicamentos cuya presentación sea jarabe y el nombre de su laboratorio contenga la palabra regional.

CONSULTA SQL

```
SELECT
    m_medicamentos.nombre_med,
    SUM(m_medicamentos.stock) "STOCK",
    m_presentaciones.nombre_pres
FROM m_medicamentos
    INNER JOIN m_presentaciones ON m_medicamentos.id_pres =
m_presentaciones.id_pres
    INNER JOIN m_laboratorios ON m_medicamentos.id_lab =
m_medicamentos.id_lab
WHERE m_presentaciones.nombre_pres LIKE 'JAR_BE'
AND UPPER(m_laboratorios.nombre_lab) LIKE UPPER('%REGIONAL%')
GROUP BY m_medicamentos.nombre_med,
m_presentaciones.nombre_pres;
```

CAPTURA

The screenshot shows the Oracle SQL Developer interface. The 'Conexiones' pane on the left lists various database connections. The 'Hoja de Trabajo' pane in the center displays the SQL query from the previous section. The 'Resultado de la Consulta' pane at the bottom shows the results of the query execution. The results are displayed in a table with four columns: NOMBRE_MED, STOCK, NOMBRE_PRE, and a fourth column that is not explicitly named but contains the values 'JARABE'.

NOMBRE_MED	STOCK	NOMBRE_PRE	
1 MEDICAMENTO ANTIINFLAMATORIO 1	450	JARABE	JARABE
2 MEDICAMENTO ANTIBIOTICO 1	350	JARABE	JARABE
3 MEDICAMENTO FIEBRE 1	240	JARABE	JARABE
4 MEDICAMENTO FIEBRE 2	600	JARABE	JARABE

VERSIÓN 2

Sólo saldrán los medicamentos de los que hayan realizado más de una venta

CONSULTA SQL

```
SELECT
    m_medicamentos.nombre_med,
    SUM(m_medicamentos.stock) "STOCK",
    m_presentaciones.nombre_pres
FROM m_medicamentos
    INNER JOIN m_presentaciones ON m_medicamentos.id_pres =
m_presentaciones.id_pres
    INNER JOIN m_laboratorios ON m_medicamentos.id_lab =
m_medicamentos.id_lab
    INNER JOIN m_ventas_med ON m_ventas_med.id_med =
m_medicamentos.id_med
WHERE m_presentaciones.nombre_pres LIKE 'JAR_BE'
AND UPPER(m_laboratorios.nombre_lab) LIKE UPPER('%REGIONAL%')
AND m_ventas_med.unidades > 1
    GROUP BY m_medicamentos.nombre_med,
m_presentaciones.nombre_pres ;
```

CAPTURA

The screenshot displays the Oracle SQL Developer interface. The main window shows the following SQL query:

```
--VERSIÓN 2
--Sólo saldrán los medicamentos de los que hayan realizado más de una venta
SELECT
    m_medicamentos.nombre_med,
    SUM(m_medicamentos.stock) "STOCK",
    m_presentaciones.nombre_pres
FROM m_medicamentos
    INNER JOIN m_presentaciones ON m_medicamentos.id_pres = m_presentaciones.id_pres
    INNER JOIN m_laboratorios ON m_medicamentos.id_lab = m_medicamentos.id_lab
    INNER JOIN m_ventas_med ON m_ventas_med.id_med = m_medicamentos.id_med
WHERE m_presentaciones.nombre_pres LIKE 'JAR_BE'
AND UPPER(m_laboratorios.nombre_lab) LIKE UPPER('%REGIONAL%')
AND m_ventas_med.unidades > 1
    GROUP BY m_medicamentos.nombre_med, m_presentaciones.nombre_pres ;
```

The bottom panel shows the results of the query:

NOMBRE_MED	STOCK	NOMBRE_PRES
1 MEDICAMENTO ANTIFLAGATOZO 1-DARABE	400	DARABE
2 MEDICAMENTO FIEBRE 1-DARABE	400	DARABE

APARTADO 3

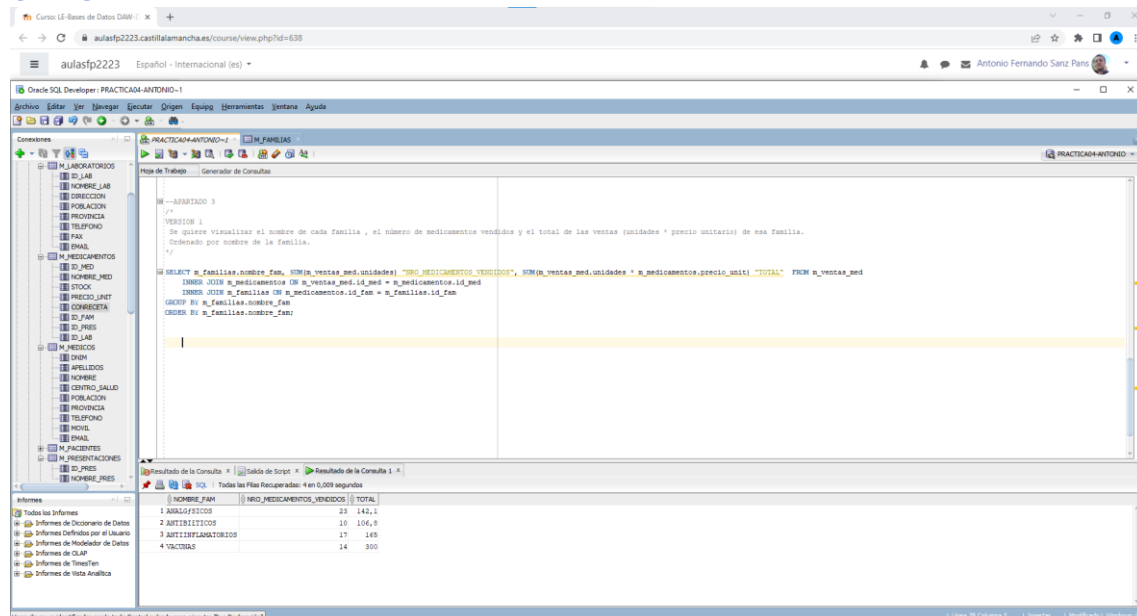
VERSIÓN 1

Se quiere visualizar el nombre de cada familia, el número de medicamentos vendidos y el total de las ventas (unidades * precio unitario) de esa familia. Ordenado por nombre de la familia.

SENTENCIA SQL

```
SELECT
    m_familias.nombre_fam,
    SUM(m_ventas_med.unidades) "NRO_MEDICAMENTOS_VENDIDOS",
    SUM(m_ventas_med.unidades * m_medicamentos.precio_unit) "TOTAL"
FROM m_ventas_med
    INNER JOIN m_medicamentos ON m_ventas_med.id_med =
m_medicamentos.id_med
    INNER JOIN m_familias ON m_medicamentos.id_fam =
m_familias.id_fam
GROUP BY m_familias.nombre_fam
ORDER BY m_familias.nombre_fam;
```

CAPTURA



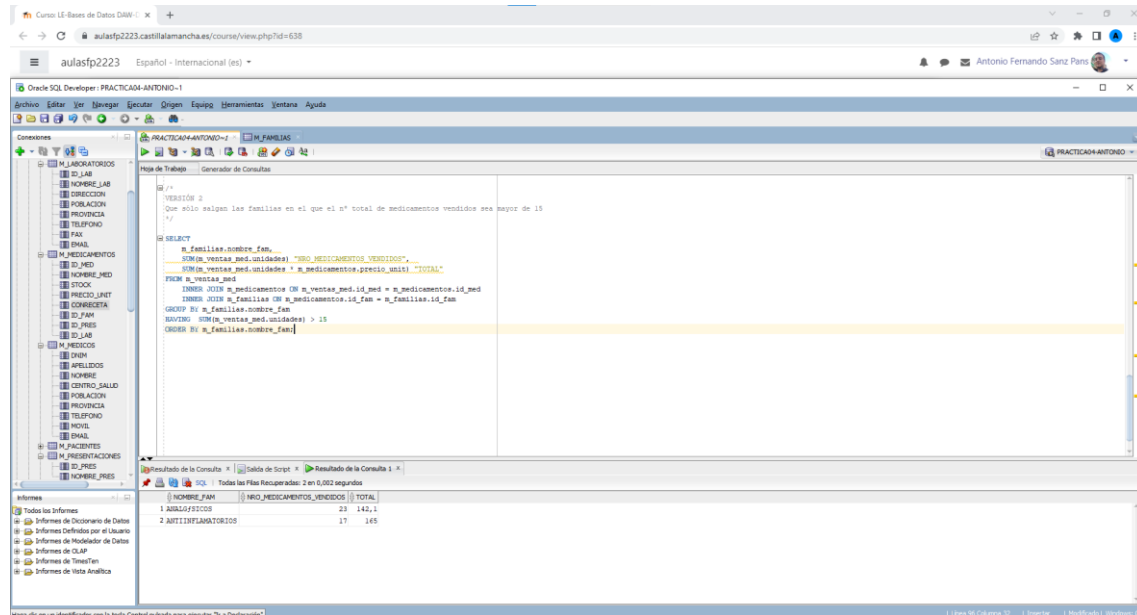
VERSIÓN 2

Que sólo salgan las familias en el que el nº total de medicamentos vendidos sea mayor de 15.

SENTENCIA SQL

```
SELECT
    m_familias.nombre_fam,
    SUM(m_ventas_med.unidades) "NRO_MEDICAMENTOS_VENDIDOS",
    SUM(m_ventas_med.unidades * m_medicamentos.precio_unit) "TOTAL"
FROM m_ventas_med
    INNER JOIN m_medicamentos ON m_ventas_med.id_med =
m_medicamentos.id_med
    INNER JOIN m_familias ON m_medicamentos.id_fam =
m_familias.id_fam
GROUP BY m_familias.nombre_fam
HAVING SUM(m_ventas_med.unidades) > 15
ORDER BY m_familias.nombre_fam;
```

CAPTURA



VERSIÓN 3

No he podido mostrarlo como se indicaba en el documento ya que los datos me dan de otra forma. He puesto el select que calcula la media de las ventas de medicamentos para que pueda comprobar por qué en mi ejercicio, se ve distinto al pdf del enunciado.

CODIGO SQL

```
--Calcular la media de las ventas
SELECT
    AVG(m_ventas_med.unidades * m_medicamentos.precio_unit)
MEDIA_VENTAS
FROM m_ventas_med
    INNER JOIN m_medicamentos ON m_ventas_med.id_med =
m_medicamentos.id_med;

--Mostrar los valores que esten por encima de esa media
SELECT
    m_familias.nombre_fam,
    SUM(m_ventas_med.unidades) "NRO_MEDICAMENTOS_VENDIDOS",
    SUM(m_ventas_med.unidades * m_medicamentos.precio_unit)
"TOTAL_VENTAS"
FROM m_ventas_med
    INNER JOIN m_medicamentos ON m_ventas_med.id_med =
m_medicamentos.id_med
    INNER JOIN m_familias ON m_medicamentos.id_fam =
m_familias.id_fam
GROUP BY m_familias.nombre_fam
HAVING SUM(m_ventas_med.unidades * m_medicamentos.precio_unit) >
AVG(m_ventas_med.unidades * m_medicamentos.precio_unit)
ORDER BY m_familias.nombre_fam;--
```


CAPTURA

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the 'Script' tab. The query is designed to find families where the average sales of a specific medicine are higher than the overall average sales of all medicines. The query uses a subquery to calculate the average sales per family and medicine, and then compares it to the overall average sales.

```
--Que sólo saigan las familias en las que la media de sus ventas sea mayor que la media de todas las ventas de todos los medicamentos.
--Calcular la media de las ventas
SELECT AVG(m_venta_med.unidades * m_medicamentos.precio_unit) MEDIA_VENTAS FROM m_venta_med
INNER JOIN m_medicamentos ON m_venta_med.id_med = m_medicamentos.id_med;

--Mostrar los valores que estén por encima de esa media
SELECT
  m_familias.nombre_fam,
  SUM(m_venta_med.unidades) "MEDICAMENTOS VENDIDOS",
  SUM(m_venta_med.unidades * m_medicamentos.precio_unit) "TOTAL VENTAS"
FROM m_venta_med
INNER JOIN m_medicamentos ON m_venta_med.id_med = m_medicamentos.id_med
INNER JOIN m_familias ON m_medicamentos.id_fam = m_familias.id_fam
GROUP BY m_familias.nombre_fam
HAVING SUM(m_venta_med.unidades * m_medicamentos.precio_unit) > AVG(m_venta_med.unidades * m_medicamentos.precio_unit)
ORDER BY m_familias.nombre_fam;
```

The results of the query are displayed in the 'Results' tab, showing a table with three columns: FAMILIAS, MEDICAMENTOS VENDIDOS, and TOTAL VENTAS. The data is as follows:

FAMILIAS	MEDICAMENTOS VENDIDOS	TOTAL VENTAS
1 ADALGESICOS	23	142,1
2 ANTIBIOTICOS	10	104,8
3 AUTOTRANSFUSION	17	145
4 VACCINAS	14	200

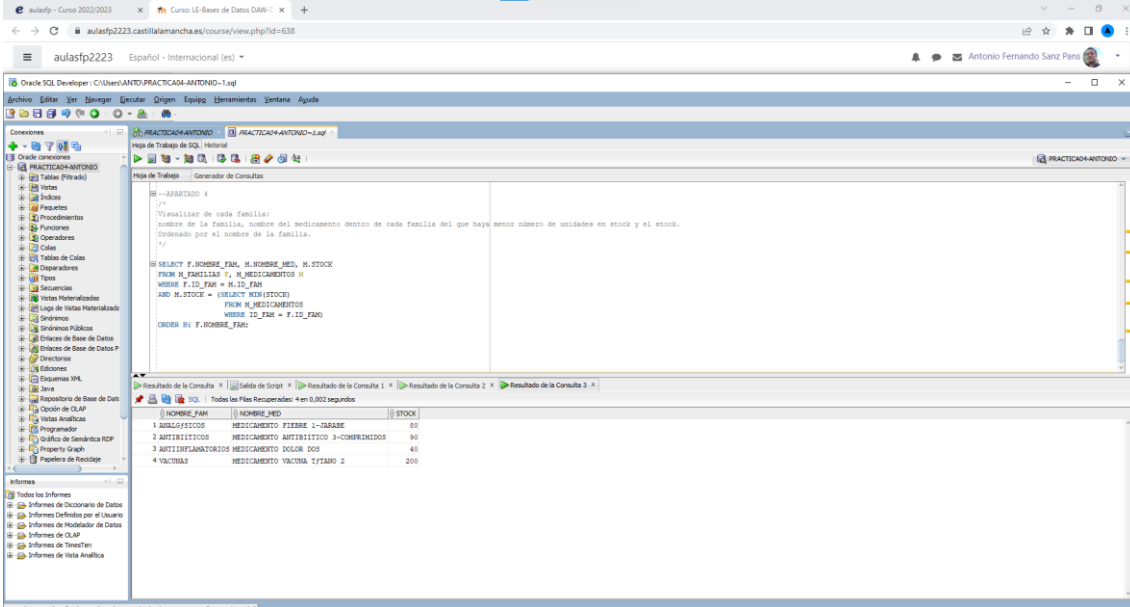
APARTADO 4

Visualizar de cada familia: nombre de la familia, nombre del medicamento dentro de cada familia del que haya menor número de unidades en stock y el stock. Ordenado por el nombre de la familia.

CODIGO SQL

```
SELECT F.NOMBRE_FAM, M.NOMBRE_MED, M.STOCK
FROM M_FAMILIAS F, M_MEDICAMENTOS M
WHERE F.ID_FAM = M.ID_FAM
AND M.STOCK = (SELECT MIN(STOCK)
                FROM M_MEDICAMENTOS
                WHERE ID_FAM = F.ID_FAM)
ORDER BY F.NOMBRE_FAM;
```

CAPTURA



Oracle SQL Developer - C:\Users\ANTONIO\Documents\PRACTICA4-ANTONIO-1.sql

Conexiones: PRACTICA4-ANTONIO

Hoja de Trabajo: PRACTICA4-ANTONIO-1.sql

Hoja de Trabajo: Generador de Consultas

```
--APARTADO 4
/*
Visualizar de cada familia:
Nombre de la familia, nombre del medicamento dentro de cada familia del que haya menor número de unidades en stock y el stock.
Ordenado por el nombre de la familia.
*/
SELECT F.NOMBRE_FAM, M.NOMBRE_MED, M.STOCK
FROM M_FAMILIAS F, M_MEDICAMENTOS M
WHERE F.ID_FAM = M.ID_FAM
AND M.STOCK = (SELECT MIN(STOCK)
                FROM M_MEDICAMENTOS
                WHERE ID_FAM = F.ID_FAM)
ORDER BY F.NOMBRE_FAM;
```

Resultado de la Consulta 1: Todas las Filas Recuperadas: 4 en 0,002 segundos

NOMBRE_FAM	NOMBRE_MED	STOCK
1 ANALGÉSICOS	MEDICAMENTO FIEBRE 1-2ARABE	50
2 ANTIINFLAMATORIOS	MEDICAMENTO ANTIINFLAMATORIO 3-COMPRIMIDOS	50
3 ANTIINFLAMATORIOS	MEDICAMENTO DOLOR DOS	50
4 VACUNAS	MEDICAMENTO VACUNA TYPHO 2	200

Informes: Todos los Informes, Informes de Diccionario de Datos, Informes Definidos por el Usuario, Informes de Modelador de Datos, Informes de OLAP, Informes de Semántica RPD, Informes de Trazado, Informes de Vista Analítica

APARTADO 5

Se listará el apellido y nombre del paciente, población, la suma de las unidades vendidas de medicamentos, total de sus ventas (unidades * precio venta) y fecha de la última venta . Sólo de tendrán en cuenta aquellos medicamentos que se hayan vendido más de 2 veces.

CODIGO SQL

```
SELECT p.apellidos, p.nombre, p.poblacion, SUM(vm.unidades) AS "SUMA
UNIDADES MED. VENDIDOS", SUM(vm.unidades*m.precio_unit) AS
"TOTAL COMPRADO", MAX(vm.fecha_venta) "FECHA DE SU ULTIMA COMPRA",
COUNT(vm.unidades)
FROM m_ventas_recetas VR
INNER JOIN m_pacientes P ON vr.dnip = p.dnip
INNER JOIN m_ventas_med VM ON vm.id_venta = vr.id_venta
INNER JOIN m_medicamentos M ON VM.id_med = M.id_med
GROUP BY p.apellidos, p.nombre, p.poblacion
HAVING COUNT(vm.unidades) >= 2;
```

CAPTURA

The screenshot shows the Oracle SQL Developer interface. The main window displays the SQL query from the previous section. Below the query editor, the 'Results' tab is active, showing the output of the query. The results are displayed in a table with 6 columns: APELLIDOS, NOMBRE, POBLACION, SUMA UNIDADES MED. VENDIDOS, TOTAL COMPRADO, and FECHA DE SU ULTIMA COMPRA. There are 3 rows of data.

APELLIDOS	NOMBRE	POBLACION	SUMA UNIDADES MED. VENDIDOS	TOTAL COMPRADO	FECHA DE SU ULTIMA COMPRA
PAPELLIDO11	PAPELLIDO13	PROVINCIA MADRID	8	81,24/10/21	
PAPELLIDO13	PAPELLIDO13	PROVINCIA TOLEDO	12	76,42/11/21	
PAPELLIDO15	PAPELLIDO15	PROVINCIA PUERTO RICO	8	141,28/11/21	

APARTADO 6

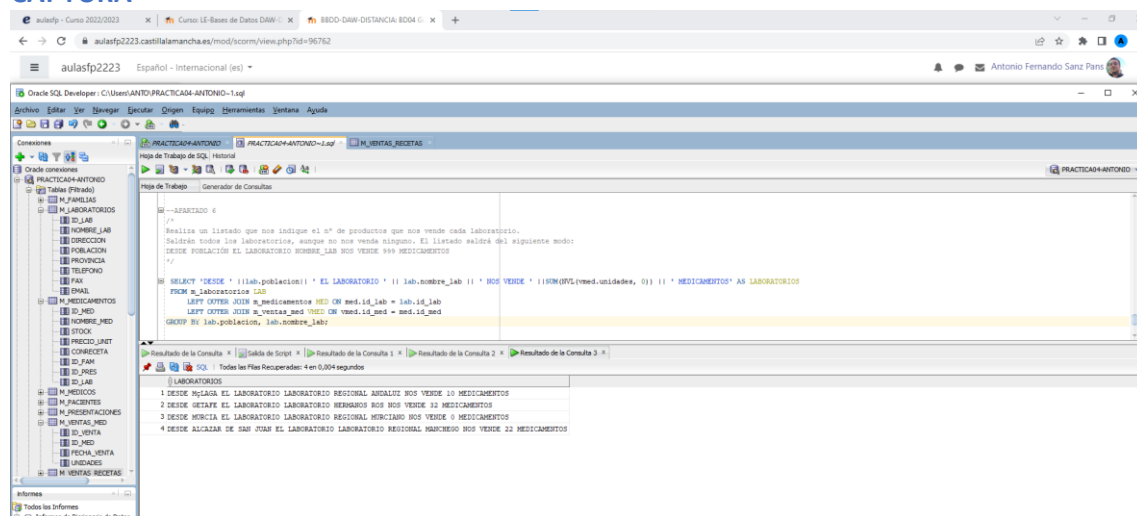
Realiza un listado que nos indique el nº de productos que nos vende cada laboratorio. Saldrán todos los laboratorios, aunque no nos venda ninguno. El listado saldrá del siguiente modo:

DESDE **POBLACIÓN** EL LABORATORIO **NOMBRE_LAB** NOS VENDE **999** MEDICAMENTOS

CODIGO SQL

```
SELECT 'DESDE ' || lab.poblacion || ' EL LABORATORIO ' ||  
lab.nombre_lab || ' NOS VENDE ' || SUM(NVL(vmed.unidades, 0)) || '  
MEDICAMENTOS' AS LABORATORIOS  
FROM m_laboratorios LAB  
LEFT OUTER JOIN m_medicamentos MED ON med.id_lab = lab.id_lab  
LEFT OUTER JOIN m_ventas_med VMED ON vmed.id_med = med.id_med  
GROUP BY lab.poblacion, lab.nombre_lab;
```

CAPTURA



The screenshot shows the Oracle SQL Developer interface. The query from the previous block is pasted into the SQL Editor and executed. The results are shown in the 'Resultado de la Consulta' pane, displaying a table with 4 rows and 1 column labeled 'LABORATORIOS'.

LABORATORIOS
1 DESDE MELILLA EL LABORATORIO LABORATORIO REGIONAL AUGALUT NOS VENDE 10 MEDICAMENTOS
2 DESDE GETAFE EL LABORATORIO LABORATORIO BERNABOS NOS VENDE 12 MEDICAMENTOS
3 DESDE MÉRIDA EL LABORATORIO LABORATORIO REGIONAL MÉRIDIANO NOS VENDE 0 MEDICAMENTOS
4 DESDE ALCAZAR DE SAN JUAN EL LABORATORIO LABORATORIO REGIONAL MADRIBENOS NOS VENDE 22 MEDICAMENTOS