

EJERCICIO 1

ENUNCIADO:

Versión 1: Para cada cliente: Nombre del cliente, población y nº de ventas realizadas (cada registro de la tabla A_VENTAS es una venta). Ordenado por nombre del cliente.

Versión 2: en la que solo se visualicen los clientes de la provincia de CIUDAD REAL que hayan realizado mas de 3 ventas.

SENTENCIA SQL:

Versión 1:

```
SELECT
    C.NOMBRECLI "Nombre cliente",
    C.POBLACION "Poblacion",
    COUNT(V.IDVENTA) "Numero ventas"
FROM A_CLIENTES C, A_VENTAS V
WHERE V.NIF=C.NIF
GROUP BY C.NOMBRECLI, C.POBLACION
ORDER BY C.NOMBRECLI
;
```

Versión 2:

```
SELECT
    C.NOMBRECLI "Nombre cliente",
    C.POBLACION "Poblacion",
    COUNT(V.IDVENTA) "Numero ventas"
FROM A_CLIENTES C, A_VENTAS V
WHERE V.NIF=C.NIF
AND UPPER(C.PROVINCIA)='CIUDAD REAL'
GROUP BY C.NOMBRECLI, C.POBLACION
HAVING COUNT(V.IDVENTA)>3
ORDER BY C.NOMBRECLI
;
```

CAPTURA:

Versión 1

The screenshot displays the Oracle SQL Developer environment. The central query editor shows two SQL queries. The first query, labeled 'Versión 1', is a SELECT statement that retrieves client names, populations, and the count of sales for each client. The second query, labeled 'Versión 2', is a similar SELECT statement but includes a filter for clients in the province of Ciudad Real who have made more than 3 sales. The results window at the bottom shows the output of the first query, displaying a table with columns for client name, population, and number of sales. The table contains four rows of data.

Nombre cliente	Poblacion	Numero ventas
CLIENTE 1	MIGUELITURRA	5
cliente 2	toledo	2
CLIENTE 4	ciudad real	3
CLIENTE 6	MADRID	5

The screenshot shows a desktop environment with two main windows. On the left, a web browser displays a course page for 'LE-Bases de Datos' with a sidebar menu containing links like 'Home / My courses /', 'Situación actual: Antes de iniciar la pr...', 'Si te surge cualquier...', 'Tutor del módulo: I...', 'Contacto: A través de las dist...', 'Enviando un co...', 'Abriendo un hi...', 'Mensaje en de...', and '(Preferiblemen...'. The main content area of the browser shows a list of activities: 'Activities', 'Assignments', 'Forums', 'Glossaries', 'Quizzes', 'Resources', 'SCORM packages', 'COURSES', and a list of course titles including 'LE-Leng. marcas y sist. gest. inf DAW E', 'LE-Entornos de desarrollo DAW E', 'LE-Programación DAW E', 'LE-Bases de Datos DAW E', 'LE-Form. y Orient. Lab. DAW E', 'LE-Sistemas informáticos DAW E', 'LE-Inglés DAW E', and 'Curso Común: CFGS Desarrollo de Aplicaciones Web (Maestre de Calatrava)'. On the right, the Oracle SQL Developer window is open, showing a query for 'EJERCICIO 2' and its results. The query is:

```

SELECT
  C.NOMBRECLI "Nombre cliente",
  C.POBACION "Población",
  COUNT(V.IDVENTA) "Numero ventas"
FROM A_CLIENTES C, A_VENTAS V
WHERE V.NIF=C.NIF
GROUP BY C.NOMBRECLI, C.POBACION
ORDER BY C.NOMBRECLI

```

The results show one client: 'CLIENTE 1' with 'POBACION' 'MIGUELTURRA' and 'Numero ventas' '5'. The SQL Developer window also shows a list of tables and views on the left, and a 'Hoja de Trabajo' (Worksheet) on the right.

EJERCICIO 2

ENUNCIADO:

De cada cliente: los siguientes datos de las ventas realizadas:

NOMBRE DEL CLIENTE, NIF, FECHA DE LA VENTA, NOMBRE DEL ARTÍCULO, PRECIO VENTA(A_ARTICULOS), PRECIO VENTA (V_VENTA), UNIDADES VENDIDAS, TOTAL VENTA (PRECIO VENTA*UNIDADES VENDIDAS)

Solo visualizaremos las ventas de los clientes de la provincia de Ciudad Real y Toledo.

El nombre de los artículos debe llevar un 1 en cualquier lugar.

La fecha de la venta esté entre 01/03/2019 y el 31/12/2020

La fecha de la venta tendrá el siguiente formato: 13/JUNIO/2020

Se ordenará por el nombre del cliente.

Realiza 2 versiones: una cruzando las tablas y otra con el INNER JOIN.

SENTENCIA SQL:

Versión 1:

SELECT

```

  C.NOMBRECLI "Nombre cliente",
  C.NIF,
  TO_CHAR(V.FECHA_VENTA, 'DD/MONTH/YYYY') "Fecha venta",
  A.NOMBREART "Nombre articulo",
  A.PRECIO_VENTA "Precio venta",
  V.PRECIO_VENTA "Precio vendido",
  V.UNIDADES_VENDIDAS "Unidades vendidas",
  V.PRECIO_VENTA*V.UNIDADES_VENDIDAS "Total"

```

```

FROM A_CLIENTES C, A_VENTAS V, A_ARTICULOS A
WHERE C.NIF=V.NIF AND V.IDARTICULO=A.IDARTICULO
AND UPPER(C.PROVINCIA) IN ('CIUDAD REAL', 'TOLEDO')
AND V.FECHA_VENTA BETWEEN '01/03/2019' AND '31/12/2020'
AND A.NOMBREART LIKE '%1%'
ORDER BY C.NOMBRECLI
;

```

Versión 2:

SELECT

```
C.NOMBRECLI "Nombre cliente",  
C.NIF,  
TO_CHAR(V.FECHA_VENTA, 'DD/MONTH/YYYY') "Fecha venta",  
A.NOMBREART "Nombre articulo",  
A.PRECIO_VENTA "Precio venta",  
V.PRECIO_VENTA "Precio vendido",  
V.UNIDADES_VENDIDAS "Unidades vendidas",  
V.PRECIO_VENTA*V.UNIDADES_VENDIDAS "Total"
```

```
FROM A_CLIENTES C INNER JOIN A_VENTAS V ON C.NIF=V.NIF INNER JOIN A_ARTICULOS A ON V.IDARTICULO=A.IDARTICULO  
WHERE UPPER(C.PROVINCIA) IN ('CIUDAD REAL', 'TOLEDO')  
AND V.FECHA_VENTA BETWEEN '01/03/2019' AND '31/12/2020'  
AND A.NOMBREART LIKE '%1%'  
ORDER BY C.NOMBRECLI
```

;

CAPTURA:

Versión 1:

The screenshot shows the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) window displays a SQL query. The 'Resultado de la Consulta' (Query Result) window shows the results of the query, which are 4 rows of data. The results are as follows:

	Nombre cliente	NIF	Fecha venta	Nombre articulo	Precio venta	Precio vendido	Unidades vendidas	Total
1	CLIENTE 1	1B	01/ENERO /2020	ARTICULO 1	100	100	10	1000
2	CLIENTE 1	1B	15/JULIO /2019	ARTICULO 13	1300	1300	100	130000
3	CLIENTE 1	1B	14/NOVIEMBRE /2020	ARTICULO 12	1200	1200	203	243600
4	cliente 2	2B	04/MAYO /2019	ARTICULO 1	100	85	30	2550

Versión 2:

The screenshot shows the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) window displays a SQL query. The 'Resultado de la Consulta' (Query Result) window shows the results of the query, which are 4 rows of data. The results are as follows:

	Nombre cliente	NIF	Fecha venta	Nombre articulo	Precio venta	Precio vendido	Unidades vendidas	Total
1	CLIENTE 1	1B	01/ENERO /2020	ARTICULO 1	100	100	10	1000
2	CLIENTE 1	1B	15/JULIO /2019	ARTICULO 13	1300	1300	100	130000
3	CLIENTE 1	1B	14/NOVIEMBRE /2020	ARTICULO 12	1200	1200	203	243600
4	cliente 2	2B	04/MAYO /2019	ARTICULO 1	100	85	30	2550

EJERCICIO 3

ENUNCIADO:

Realiza una nueva versión del ejercicio anterior en la que sólo se tengan en cuenta los artículos que hayan comprado en total mas de 80 unidades.

SENTENCIA SQL:

SELECT

```
C.NOMBRECLI "Nombre cliente",  
C.NIF,  
TO_CHAR(V.FECHA_VENTA, 'DD/MONTH/YYYY') "Fecha venta",  
A.NOMBREART "Nombre articulo",  
A.PRECIO_VENTA "Precio venta",  
V.PRECIO_VENTA "Precio vendido",  
V.UNIDADES_VENDIDAS "Unidades vendidas",  
V.PRECIO_VENTA*V.UNIDADES_VENDIDAS "Total"
```

FROM A_CLIENTES C, A_VENTAS V, A_ARTICULOS A

WHERE C.NIF=V.NIF AND V.IDARTICULO=A.IDARTICULO

AND UPPER(C.PROVINCIA) IN ('CIUDAD REAL', 'TOLEDO')

AND V.FECHA_VENTA BETWEEN '01/03/2019' AND '31/12/2020'

AND A.NOMBREART LIKE '%1%'

AND V.UNIDADES_VENDIDAS > 80

ORDER BY C.NOMBRECLI

;

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The main window shows the execution of the SQL query for 'EJERCICIO 3'. The query is as follows:

```
-- EJERCICIO 3:  
-- Realiza una nueva versión del ejercicio anterior en la que sólo se tengan en cuenta los artículos que hayan  
SELECT  
C.NOMBRECLI "Nombre cliente",  
C.NIF,  
TO_CHAR(V.FECHA_VENTA, 'DD/MONTH/YYYY') "Fecha venta",  
A.NOMBREART "Nombre articulo",  
A.PRECIO_VENTA "Precio venta",  
V.PRECIO_VENTA "Precio vendido",  
V.UNIDADES_VENDIDAS "Unidades vendidas",  
V.PRECIO_VENTA*V.UNIDADES_VENDIDAS "Total"  
FROM A_CLIENTES C, A_VENTAS V, A_ARTICULOS A  
WHERE C.NIF=V.NIF AND V.IDARTICULO=A.IDARTICULO  
AND UPPER(C.PROVINCIA) IN ('CIUDAD REAL', 'TOLEDO')  
AND V.FECHA_VENTA BETWEEN '01/03/2019' AND '31/12/2020'  
AND A.NOMBREART LIKE '%1%'  
AND V.UNIDADES_VENDIDAS > 80  
ORDER BY C.NOMBRECLI
```

The 'Resultado de la Consulta' window shows the following data:

	Nombre cliente	NIF	Fecha venta	Nombre articulo	Precio venta	Precio vendido	Unidades vendidas	Total
1	CLIENTE 1	1B	15/JULIO /2019	ARTICULO 13	1300	1300	100	130000
2	CLIENTE 1	1B	14/NOVIEMBRE /2020	ARTICULO 12	1200	1200	203	243600

The background shows a course page for 'LE-Bases de Datos' with various navigation links and a sidebar with course details.

EJERCICIO 4

ENUNCIADO:

De cada fabricante: las ventas realizadas en el año 2019 de artículos que pertenezcan a la categoría CATEGORÍA 1, los datos a visualizar serán:

Nombre del fabricante, Nombre de los artículos que ha vendido y por cada artículo la suma de las unidades vendidas y el total de las ventas de ese artículo (multiplicando las unidades vendidas por precio de venta de la tabla ventas).

Ordenado por nombre de fabricante y dentro de éste por nombre artículo.

SENTENCIA SQL:

```
SELECT
    F.NOMBREFAB "Nombre fabricante",
    A.NOMBREART "Nombre articulo",
    SUM(V.UNIDADES_VENDIDAS) "Unidades vendidas",
    SUM(V.UNIDADES_VENDIDAS*V.PRECIO_VENTA) "Total ventas"
FROM A_FABRICANTES F, A_VENTAS V, A_ARTICULOS A
WHERE V.FECHA_VENTA BETWEEN '01/01/2019' AND '31/12/2019'
AND A.IDCATEGORIA = '1'
AND F.IDFABRICANTE=A.IDFABRICANTE AND A.IDARTICULO=V.IDARTICULO
GROUP BY F.NOMBREFAB, A.NOMBREART
ORDER BY F.NOMBREFAB, A.NOMBREART
;
```

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) window shows the following SQL query:

```
-- EJERCICIO 4:
-- De cada fabricante: las ventas realizadas en el año 2019 de artículos que pertenezcan a la categoría CATEGORÍA 1, los datos a visualizar serán:
-- Nombre del fabricante, Nombre de los artículos que ha vendido y por cada artículo la suma de las unidades vendidas y el total de las ventas de ese artículo
-- Ordenado por nombre de fabricante y dentro de éste por nombre artículo

SELECT
    F.NOMBREFAB "Nombre fabricante",
    A.NOMBREART "Nombre articulo",
    SUM(V.UNIDADES_VENDIDAS) "Unidades vendidas",
    SUM(V.UNIDADES_VENDIDAS*V.PRECIO_VENTA) "Total ventas"
FROM A_FABRICANTES F, A_VENTAS V, A_ARTICULOS A
WHERE V.FECHA_VENTA BETWEEN '01/01/2019' AND '31/12/2019'
AND A.IDCATEGORIA = '1'
AND F.IDFABRICANTE=A.IDFABRICANTE AND A.IDARTICULO=V.IDARTICULO
GROUP BY F.NOMBREFAB, A.NOMBREART
ORDER BY F.NOMBREFAB, A.NOMBREART
;
```

The 'Resultado de la Consulta' (Query Result) window shows the following data:

	Nombre fabricante	Nombre articulo	Unidades vendidas	Total ventas
1	FABRICANTE 1	ARTICULO 1	50	4350
2	FABRICANTE 5	ARTICULO 13	100	130000
3	FABRICANTE 5	ARTICULO 3	6	1800
4	FABRICANTE 5	ARTICULO 5	67	31490

The background shows a web browser with a course page titled 'LE-Bases de Datos' and a sidebar with navigation links like 'Activities', 'Assignments', 'Forums', 'Glossaries', 'Quizzes', 'Resources', 'SCORM packages', 'COURSES', and 'LE-Leng. marcas y sist. gest. inf DAW E'.

EJERCICIO 5

ENUNCIADO:

Será parecido al anterior, se visualizará de cada fabricante, las ventas realizadas en el año 2019 de artículos que pertenezcan las categorías que tengan mas de 3 artículos. Datos a visualizar:

Nombre del fabricante, Nombre de los artículos que ha vendido, Nombre de la categoría y por cada artículo la suma de las unidades vendidas y el total de las ventas de ese artículo (multiplicando las unidades vendidas por precio de venta de la tabla ventas).

SENTENCIA SQL:

SELECT

```
F.NOMBREFAB "Nombre fabricante",  
A.NOMBREART "Nombre articulo",  
C.NOMBRECATEG "Nombre categoria",  
SUM(V.UNIDADES_VENDIDAS) "Unidades vendidas",  
SUM(V.UNIDADES_VENDIDAS*V.PRECIO_VENTA) "Total ventas"
```

FROM A_FABRICANTES F, A_VENTAS V, A_ARTICULOS A, A_CATEGORIAS C

WHERE V.FECHA_VENTA BETWEEN '01/01/2019' AND '31/12/2019'

AND F.IDFABRICANTE=A.IDFABRICANTE AND A.IDARTICULO=V.IDARTICULO AND C.IDCATEGORIA=A.IDCATEGORIA

GROUP BY F.NOMBREFAB, A.NOMBREART, C.NOMBRECATEG

HAVING SUM(A.IDCATEGORIA)<=3

ORDER BY F.NOMBREFAB, A.NOMBREART

;

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The main window shows the SQL script for Exercise 5, which filters for the year 2019 and groups by manufacturer, article, and category. The result window shows the following data:

Nombre fabricante	Nombre articulo	Nombre categoria	Unidades vendidas	Total ventas
1 FABRICANTE 1	ARTICULO 1	Categoria 1	50	4350
2 fabricante 4	ARTICULO 4	CATEGORIA 3	10	4000
3 FABRICANTE 5	ARTICULO 13	Categoria 1	100	130000
4 FABRICANTE 5	ARTICULO 3	Categoria 1	6	1800
5 FABRICANTE 5	ARTICULO 5	Categoria 1	67	31490

EJERCICIO 6

ENUNCIADO:

Realiza una versión del ejercicio anterior en la que solo se visualicen aquellos artículos cuya media de las unidades vendidas por ese fabricante sea mayor que la media de las unidades vendidas de todos los artículos.

SENTENCIA SQL:

```
SELECT
    F.NOMBREFAB "Nombre fabricante",
    A.NOMBREART "Nombre artículo",
    C.NOMBRECATEG "Nombre categoría",
    SUM(V.UNIDADES_VENDIDAS) "Unidades vendidas",
    SUM(V.UNIDADES_VENDIDAS*V.PRECIO_VENTA) "Total ventas"
FROM A_FABRICANTES F, A_VENTAS V, A_ARTICULOS A, A_CATEGORIAS C
WHERE V.FECHA_VENTA BETWEEN '01/01/2019' AND '31/12/2019'
AND F.IDFABRICANTE=A.IDFABRICANTE AND A.IDARTICULO=V.IDARTICULO AND C.IDCATEGORIA=A.IDCATEGORIA
GROUP BY F.NOMBREFAB, A.NOMBREART, C.NOMBRECATEG
HAVING SUM(A.IDCATEGORIA)<=3
AND AVG(V.UNIDADES_VENDIDAS) > (SELECT AVG(unidades_vendidas) FROM a_ventas)
ORDER BY F.NOMBREFAB, A.NOMBREART
;
```

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The main window shows the execution of the SQL query for Exercise 6. The query is as follows:

```
SELECT
    F.NOMBREFAB "Nombre fabricante",
    A.NOMBREART "Nombre artículo",
    C.NOMBRECATEG "Nombre categoría",
    SUM(V.UNIDADES_VENDIDAS) "Unidades vendidas",
    SUM(V.UNIDADES_VENDIDAS*V.PRECIO_VENTA) "Total ventas"
FROM A_FABRICANTES F, A_VENTAS V, A_ARTICULOS A, A_CATEGORIAS C
WHERE V.FECHA_VENTA BETWEEN '01/01/2019' AND '31/12/2019'
AND F.IDFABRICANTE=A.IDFABRICANTE AND A.IDARTICULO=V.IDARTICULO AND C.IDCATEGORIA=A.IDCATEGORIA
GROUP BY F.NOMBREFAB, A.NOMBREART, C.NOMBRECATEG
HAVING SUM(A.IDCATEGORIA)<=3
AND AVG(V.UNIDADES_VENDIDAS) > (SELECT AVG(unidades_vendidas) FROM a_ventas)
ORDER BY F.NOMBREFAB, A.NOMBREART
;
```

The result of the query is displayed in the 'Resultado de la Consulta' window, showing one record:

Nombre fabricante	Nombre artículo	Nombre categoría	Unidades vendidas	Total ventas
FABRICANTE 5	ARTICULO 13	Categoría 1	100	130000

The background shows a web browser with the course page 'LE-Bases de Datos DAW E' and a sidebar with navigation links.

EJERCICIO 7

ENUNCIADO:

Se quiere visualizar de cada tienda, los siguientes datos de todos los pedidos realizados:

Nombre de la tienda, email, nombre artículo, nombre del fabricante, fecha pedido, unidades pedidas, precio unitario, total del pedido (unidades pedidas* precio unitario)

Se visualizarán los pedidos realizados en el año 2020, de los fabricantes FABRICANTE 1, FABRICANTE 2 Y FABRICANTE 5.

Ordenado por nombre de la tienda y nombre del artículo.

SENTENCIA SQL:

SELECT

```
T.NOMBRETIEN "Nombre tienda",
T.EMAIL "Correo electr.",
A.NOMBREART "Nombre articulo",
F.NOMBREFAB "Nombre fabricante",
P.FECHAPEDIDO "Fecha pedido",
P.UNIDADES_PEDIDAS "Unidades pedidas",
P.PRECIO_UNITARIO "Precio unitario",
SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO) "Total"
```

FROM A_TIENDAS T, A ARTICULOS A, A_FABRICANTES F, A_PEDIDOS P

WHERE P.FECHAPEDIDO BETWEEN '01/01/2020' AND '21/12/2020'

AND F.NOMBREFAB IN ('FABRICANTE 1', 'FABRICANTE 2', 'FABRICANTE 5')

AND T.CIF=P.CIF AND P.IDARTICULO=A.IDARTICULO AND A.IDFABRICANTE=F.IDFABRICANTE

GROUP BY T.NOMBRETIEN, T.EMAIL, A.NOMBREART, F.NOMBREFAB, P.FECHAPEDIDO, P.UNIDADES_PEDIDAS,

P.PRECIO_UNITARIO

ORDER BY T.NOMBRETIEN, A.NOMBREART

;

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The main window shows the SQL query for 'EJERCICIO 7' being executed. The query filters for the year 2020 and specific manufacturers (FABRICANTE 1, FABRICANTE 2, FABRICANTE 5). The result window shows 6 rows of data for different stores, articles, and manufacturers.

NOMBRE tienda	Correo electr.	Nombre articulo	Nombre fabricante	Fecha pedido	Unidades pedidas	Precio unitario	Total
1 TIENDA 1	tienda1@gmail.com	ARTICULO 2	FABRICANTE 5	01/03/20	10	20	200
2 TIENDA 2	tienda2@gmail.com	ARTICULO 1	FABRICANTE 1	01/01/20	50	30	1500
3 TIENDA 4	tienda4@gmail.com	ARTICULO 5	FABRICANTE 5	01/05/20	11	43	473
4 TIENDA 5	tienda5@gmail.com	ARTICULO 2	FABRICANTE 5	01/12/20	130	17	2210
5 TIENDA 6	tienda6@gmail.com	ARTICULO 3	FABRICANTE 5	01/05/20	20	35	700
6 TIENDA 6	tienda6@gmail.com	ARTICULO 5	FABRICANTE 5	01/10/20	12	50	600

EJERCICIO 8

ENUNCIADO:

Realiza una nueva versión del ejercicio anterior, en la que cambiaremos la condición de los fabricantes y sólo se tengan en cuenta los fabricantes que hayan realizado mas de 5 pedidos.

SENTENCIA SQL:

SELECT

```
T.NOMBRETIEN "Nombre tienda",
T.EMAIL "Correo electr.",
A.NOMBREART "Nombre articulo",
F.NOMBREFAB "Nombre fabricante",
P.FECHAPEDIDO "Fecha pedido",
P.UNIDADES_PEDIDAS "Unidades pedidas",
P.PRECIO_UNITARIO "Precio unitario",
SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO) "Total"
```

FROM A_TIENDAS T, A_ARTICULOS A, A_FABRICANTES F, A_PEDIDOS P

WHERE P.FECHAPEDIDO BETWEEN '01/01/2020' AND '21/12/2020'

AND A.IDFABRICANTE IN (SELECT A.IDFABRICANTE

FROM A_ARTICULOS A

GROUP BY A.IDFABRICANTE

HAVING COUNT(A.IDFABRICANTE)>=5)

AND T.CIF=P.CIF AND P.IDARTICULO=A.IDARTICULO AND A.IDFABRICANTE=F.IDFABRICANTE

GROUP BY T.NOMBRETIEN, T.EMAIL, A.NOMBREART, F.NOMBREFAB, P.FECHAPEDIDO, P.UNIDADES_PEDIDAS,

P.PRECIO_UNITARIO

ORDER BY T.NOMBRETIEN, A.NOMBREART

;

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) window shows the SQL query for Exercise 8, which is a modified version of a previous exercise. The query selects various fields from the A_TIENDAS, A_ARTICULOS, A_FABRICANTES, and A_PEDIDOS tables, including a calculated field for the total value. The query filters for orders between 01/01/2020 and 21/12/2020 and only includes manufacturers who have made 5 or more orders. The results are grouped by tienda, email, articulo, fabricante, fecha pedido, unidades pedidas, and precio unitario, and ordered by tienda and articulo.

The 'Resultado de la Consulta' (Query Result) window shows the output of the query, which is a table with 7 columns: Nombre tienda, Correo electr., Nombre articulo, Nombre fabricante, Fecha pedido, Unidades pedidas, and Precio unitario. The table contains 5 rows of data, representing 5 different stores.

Nombre tienda	Correo electr.	Nombre articulo	Nombre fabricante	Fecha pedido	Unidades pedidas	Precio unitario
1 TIENDA 1	tienda1@gmail.com	ARTICULO 2	FABRICANTE 5	01/03/20	10	200
2 TIENDA 4	tienda4@gmail.com	ARTICULO 5	FABRICANTE 5	01/05/20	11	43
3 tienda 5	tienda5@gmail.com	ARTICULO 2	FABRICANTE 5	01/12/20	130	17
4 TIENDA 6	tienda6@gmail.com	ARTICULO 3	FABRICANTE 5	01/05/20	20	35
5 TIENDA 6	tienda6@gmail.com	ARTICULO 5	FABRICANTE 5	01/10/20	12	50

EJERCICIO 9

ENUNCIADO:

Visualizar el CIF, el nombre de la tienda y el total facturado de la tienda que haya realizado la mayor venta . Para saber cuál es la mayor venta, por cada tienda hay que sumar todos los totales de sus pedidos (unidades pedidas * precio unitario).

SENTENCIA SQL:

```
SELECT
    P.CIF,
    T.NOMBRETIEN "Nombre tienda",
    SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO) "Total"
FROM A_TIENDAS T, A_PEDIDOS P
WHERE P.CIF=T.CIF
GROUP BY P.CIF, T.NOMBRETIEN
HAVING SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO)=(SELECT MAX(SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO))
FROM A_PEDIDOS P
GROUP BY P.CIF)
```

;

CAPTURA:

The screenshot displays the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) window shows a SQL query designed to find the store with the highest total sales. The query uses a subquery to determine the maximum total sales across all stores and then filters the main query to show only the store(s) that match this maximum. The results window shows the output of the query, displaying the CIF (store code), the store name, and the total sales for the store with the highest sales.

SQL Query:

```
SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO) "Total"
FROM A_TIENDAS T, A_PEDIDOS P
WHERE P.CIF=T.CIF
GROUP BY P.CIF, T.NOMBRETIEN
HAVING SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO)=(SELECT MAX(SUM(P.UNIDADES_PEDIDAS*P.PRECIO_UNITARIO))
FROM A_PEDIDOS P
GROUP BY P.CIF)
```

Resultado de la Consulta:

CIF	Nombre tienda	Total
1 AS	tienda 5	6000