



UNIVERSITAT_{DE}
BARCELONA

Módulo I:

Tarea – Consultas a Bases de datos relacionales con SQL

Estudiante:

Heiner Romero Leiva

Máster en Big Data & Data Science

Noviembre, 2021

Descripción de la tarea:

Has sido contratado por una multinacional como Data Engineer. Se trata de una empresa global que vende hardware informático, incluidos almacenamiento, placas base, RAM, tarjetas de video y CPUs. Te dan acceso a su base de datos relacional, la cual está montada en un servidor Oracle. El modelo de datos (tablas, campos y relaciones) es el siguiente:

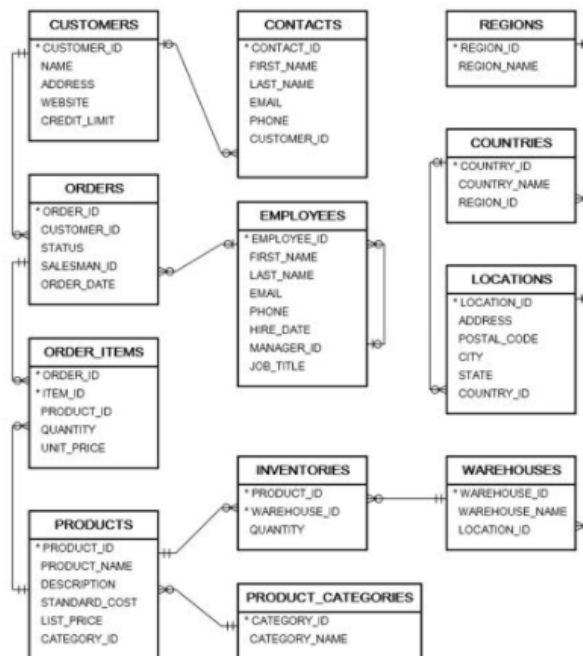


Tabla	Descripción	Registros
CONTACTS	Tabla con información de la persona de contacto de los clientes	319 registros
COUNTRIES	Tabla con información de países	25 registros
CUSTOMERS	Tabla con información de clientes	319 registros
EMPLOYEES	Tabla con información de empleados	107 registros
INVENTORIES	Tabla con información del inventario de productos	1112 registros
LOCATIONS	Tabla con ubicación de los almacenes	23 registros
ORDERS	Tabla de pedidos	105 registros
ORDER_ITEMS	Tabla con las líneas de los pedidos	665 registros
PRODUCT_CATEGORIES	Tabla con las categorías de productos	5 registros
PRODUCTS	Tabla con la información de los productos	288 registros
REGIONS	Tabla con las regiones donde la compañía opera	4 registros
WAREHOUSES	Tabla con información de los almacenes	9 registros

Debes crear las sentencias SQL necesarias para poder responder a las siguientes tareas:

1. ¿Cuáles son los datos de los almacenes que tiene la compañía? Se necesita:

- Identificador del almacén.
- Nombre del almacén.
- Nombre de la ciudad, país y región donde está ubicado.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main window contains a SQL query in the 'Hoja de Trabajo' (Worksheet) tab. The query is a SELECT statement that joins the WAREHOUSES, LOCATIONS, COUNTRIES, and REGIONS tables. The results are displayed in the 'Resultado de la Consulta' (Query Result) tab, showing 9 rows of data.

```
SELECT
  w.warehouse_id,
  w.warehouse_name,
  l.city,
  c.country_name,
  r.region_name
FROM warehouses w
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
ORDER BY 1;
```

WAREHOUSE_ID	WAREHOUSE_NAME	CITY	COUNTRY_NAME	REGION_NAME
1	Southlake, Texas	Southlake	United States of America	Americas
2	San Francisco	South San Francisco	United States of America	Americas
3	New Jersey	South Brunswick	United States of America	Americas
4	Seattle, Washington	Seattle	United States of America	Americas
5	Toronto	Toronto	Canada	Americas
6	Sydney	Sydney	Australia	Asia
7	Mexico City	Mexico City	Mexico	Americas
8	Beijing	Beijing	China	Asia
9	Bombay	Bombay	India	Asia

Query:

```
SELECT
  w.warehouse_id,
  w.warehouse_name,
  l.city,
  c.country_name,
  r.region_name
FROM warehouses w
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
ORDER BY 1;
```

2. ¿Cuál es el nombre del producto que tiene más stock en Europa?

Para este caso en particular me percaté de que en Europa no hay almacén solo existen en América y en Asia, entonces lo que hice fue crear el query con “Europe” pero como vi que no arrojaba resultados, hice otro query para ver si efectivamente no habían almacenes en Europa.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like CONTACTS, COUNTRIES, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main pane shows a SQL query in the 'Hoja de Trabajo' (Worksheet) tab:

```
SELECT
  p.product_name,
  i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Europe'
ORDER BY quantity DESC;
```

The bottom pane shows the 'Resultado de la Consulta' (Query Result) tab, which is currently empty, indicating no results were returned for the specified region.

Con este query me di cuenta de que en efecto en Europa no hay almacenes, solo en América y Asia.

The screenshot shows the Oracle SQL Developer interface with a different query in the 'Hoja de Trabajo' (Worksheet) tab:

```
SELECT
  DISTINCT(w.warehouse_id),
  c.country_name,
  r.region_name
FROM warehouses w
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
ORDER BY 1;
```

The bottom pane shows the 'Resultado de la Consulta' (Query Result) tab, which displays a table with 9 rows of data:

WAREHOUSE_ID	COUNTRY_NAME	REGION_NAME
1	United States of America	Americas
2	United States of America	Americas
3	United States of America	Americas
4	United States of America	Americas
5	Canada	Americas
6	Australia	Asia
7	Mexico	Americas
8	China	Asia
9	India	Asia

Luego lo que hice fue comprobar si el query estaba bien hecho y arrojaba resultados, así que lo probe para “Americas” y “Asia”.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main window shows a SQL query in the 'Hoja de Trabajo' (Worksheet) tab, which is filtered for the 'Americas' region. The 'Resultado de la Consulta' (Query Result) pane shows 50 rows of data.

```

SELECT
  p.product_name,
  i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Americas'
ORDER BY quantity DESC;

```

PRODUCT_NAME	QUANTITY
1 Kingston SA400S37/120G	353
2 Gigabyte GV-NI070WF20C-8GD	304
3 Zotac ZT-P108100-10P	304
4 Kingston SA400S37/120G	294
5 G.Skill Trident Z	282
6 G.Skill Ripjaws V Series	276
7 Corsair Vengeance LPX	275
8 G.Skill Trident Z	275
9 MSI GeForce GTX 1080 TI ARMOR 11G OC	273
10 Zotac ZT-P10810G-10P	273
11 Zotac ZT-P10810C-10P	273
12 G.Skill Ripjaws 4 Series	272
13 MSI GeForce GTX 1080 Ti GAMING X 11G	272
14 MSI X99A GODLIKE GAMING CARBON	271

The screenshot shows the Oracle SQL Developer interface with the same schema and query structure as the previous image, but filtered for the 'Asia' region. The 'Resultado de la Consulta' pane shows 50 rows of data.

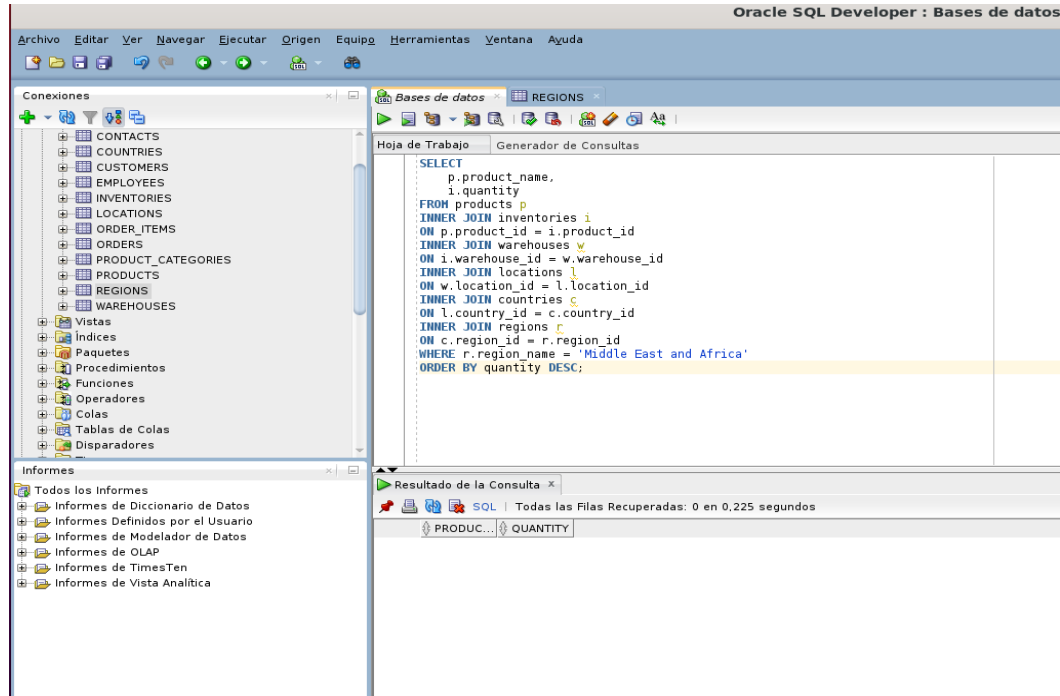
```

SELECT
  p.product_name,
  i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Asia'
ORDER BY quantity DESC;

```

PRODUCT_NAME	QUANTITY
1 Kingston SA400S37/120G	320
2 Kingston SA400S37/120G	268
3 G.Skill Trident Z	232
4 G.Skill Ripjaws V Series	226
5 G.Skill Trident Z	225
6 Corsair Vengeance LPX	225
7 G.Skill Ripjaws 4 Series	222
8 MSI X99A GODLIKE GAMING CARBON	221
9 Gigabyte GV-NI070WF20C-8GD	214
10 Zotac ZT-P108100-10P	214
11 Asus ROG STRIX X99 GAMING	213
12 MSI X99A XPOWER GAMING TITANIUM	212
13 Samsung MZ-V6E500	203
14 Samsung MZ-75ELT0B/AM	202
15 G.Skill Trident Z	194
16 Asus ROG-POSEIDON-GTX1080TI-P11G-GAMING	191

Finalmente también comprobé con “Middle East and Africa” para corroborar que no arroja ningún dato.



Queries:

/* Query para Europa*/

```
SELECT
  p.product_name,
  i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Europe'
ORDER BY quantity DESC;
```

/* Query para comprobar ubicación de los almacenes*/

```
SELECT
    DISTINCT(w.warehouse_id)
    c.country_name
    r.region_name
FROM warehouses w
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
```

/* Query para América*/

```
SELECT
    p.product_name,
    i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Americas'
ORDER BY quantity DESC;
```

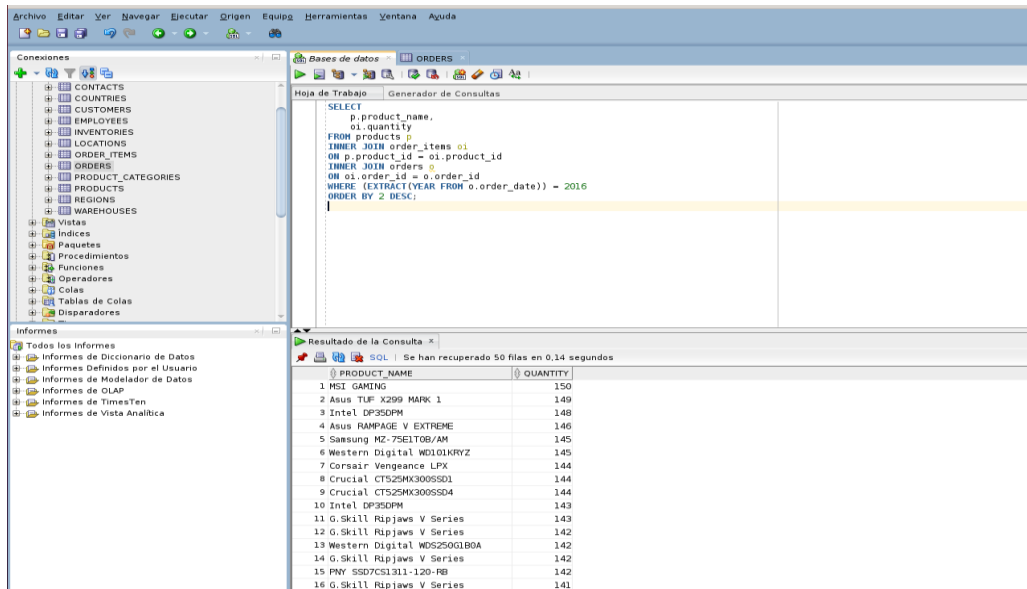
/* Query para Asia*/

```
SELECT
    p.product_name,
    i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Asia'
ORDER BY quantity DESC;
```

/* Query para Medio Este y África */

```
SELECT
    p.product_name,
    i.quantity
FROM products p
INNER JOIN inventories i
ON p.product_id = i.product_id
INNER JOIN warehouses w
ON i.warehouse_id = w.warehouse_id
INNER JOIN locations l
ON w.location_id = l.location_id
INNER JOIN countries c
ON l.country_id = c.country_id
INNER JOIN regions r
ON c.region_id = r.region_id
WHERE r.region_name = 'Middle East and Africa'
ORDER BY quantity DESC;
```


3. ¿Cuál es el producto que ha vendido más unidades durante 2016?



The screenshot shows a database management tool interface. On the left, a tree view displays the database schema with tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main window displays a SQL query in the 'Hoja de Trabajo' (Worksheet) tab. The query is as follows:

```
SELECT
    p.product_name,
    oi.quantity
FROM products p
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2016
ORDER BY 2 DESC;
```

Below the query, the 'Resultado de la Consulta' (Query Result) tab shows the results of the query. The results are displayed in a table with two columns: PRODUCT_NAME and QUANTITY. The top result is MSI GAMING with a quantity of 150.

PRODUCT_NAME	QUANTITY
1 MSI GAMING	150
2 Asus TUF X299 MARK 1	149
3 Intel DP350PM	148
4 Asus RAMPAGE V EXTREME	146
5 Samsung MZ-75ELT08/AH	145
6 Western Digital WD101HRYZ	145
7 Corsair Vengeance LPX	144
8 Crucial CT525MK300SSD1	144
9 Crucial CT525MK300SSD4	144
10 Intel DP350PM	143
11 G.Skill Ripjaws V Series	143
12 G.Skill Ripjaws V Series	142
13 Western Digital WD5250GLB0A	142
14 G.Skill Ripjaws V Series	142
15 PNY SSD7CS1311-120-RB	142
16 G.Skill Ripjaws V Series	141

El producto con más venta de unidades para 2016 es MSI GAMING con 150 unidades.

Query:

```
SELECT
    p.product_name,
    oi.quantity,
FROM products p
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2016
ORDER BY 2 DESC;
```

También lo comprobé haciendo otro tipo de sentencia con la fecha y fue el mismo:

The screenshot shows the Oracle SQL Developer interface. The 'Bases de datos' (Databases) pane on the left lists various database objects. The 'Hoja de Trabajo' (Worksheet) pane in the center contains the following SQL query:

```
SELECT
  p.product_name,
  oi.quantity
FROM products p
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE o.order_date BETWEEN '01/01/2016' AND '31/12/2016'
ORDER BY 2 DESC;
```

The 'Resultado de la Consulta' (Query Result) pane at the bottom displays the results of the query, showing 50 rows. The first 16 rows are as follows:

	PRODUCT_NAME	QUANTITY
1	MSI GAMING	150
2	Asus TUF X299 MARK 1	149
3	Intel DP35DPH	148
4	Asus RAMPAGE V EXTREME	146
5	Western Digital WD101KRYZ	145
6	Samsung MZ-75E1T0B/AM	145
7	Crucial CT525MX300SSD4	144
8	Crucial CT525MX300SSD1	144
9	Corsair Vengeance LPX	144
10	Intel DP35DPH	143
11	G.Skill Ripjaws V Series	143
12	G.Skill Ripjaws V Series	142
13	Western Digital WD5250G1B0A	142
14	G.Skill Ripjaws V Series	142
15	PNY SS07CS1311-120-R8	142
16	G.Skill Ripjaws V Series	141

Query:

```
SELECT
  p.product_name,
  oi.quantity,
FROM products p
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE o.order_date BETWEEN '01/01/2016' AND '31/12/2016'
ORDER BY 2 DESC;
```

4. ¿Cuál es la categoría de productos que ha vendido más unidades durante 2017?

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane lists various database objects. The main window displays a SQL query in the 'Hoja de Trabajo' pane. The query is as follows:

```
SELECT
  pc.category_name,
  oi.quantity
FROM product_categories pc
INNER JOIN products p
ON pc.category_id = p.category_id
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE o.order_date BETWEEN '01/01/2017' AND '31/12/2017'
ORDER BY 2 DESC;
```

The 'Resultado de la Consulta' pane shows the results of the query, which are 50 rows. The first two columns are 'CATEGORY_NAME' and 'QUANTITY'. The results are sorted by quantity in descending order.

CATEGORY_NAME	QUANTITY
Storage	150
CPU	149
Mother Board	148
Video Card	148
CPU	148
Storage	148
Mother Board	147
CPU	147
Mother Board	147
Mother Board	146
Storage	146
Video Card	146
Video Card	146
CPU	146
Video Card	145
Mother Board	144

La categoría es Storage con 150 unidades para 2017.

Query:

```
SELECT
  pc.category_name,
  oi.quantity
FROM product_categories pc
INNER JOIN products p
ON pc.category_id = p.category_id
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE o.order_date BETWEEN '01/01/2017' AND '31/12/2017'
ORDER BY 2 DESC;
```

De igual forma lo comprobé usando otro tipo de sentencia para extraer la variable de año y dio el mismo resultado.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The central pane contains the following SQL query:

```
SELECT
  pc.category_name,
  oi.quantity
FROM product_categories pc
INNER JOIN products p
ON pc.category_id = p.category_id
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2017
ORDER BY 2 DESC;
```

The bottom pane shows the query results, which are 50 rows of data. The first few rows are:

1	CATEGORY_NAME	QUANTITY
1	Storage	150
2	CPU	149
3	Mother Board	148
4	Video Card	148
5	CPU	148
6	Storage	148
7	Mother Board	147
8	CPU	147
9	Mother Board	147
10	Mother Board	146
11	Storage	146
12	Video Card	146
13	Video Card	146
14	CPU	146
15	Video Card	145
16	Storage	144

Query:

```
SELECT
  pc.category_name,
  oi.quantity
FROM product_categories pc
INNER JOIN products p
ON pc.category_id = p.category_id
INNER JOIN order_items oi
ON p.product_id = oi.product_id
INNER JOIN orders o
ON oi.order_id = o.order_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2017
ORDER BY 2 DESC;
```

5. ¿Cuál es el nombre del cliente cuyo gasto ha sido más alto en 2015?

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane lists various database objects like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main window displays a SQL query in the 'Hoja de Trabajo' pane. The query is as follows:

```
SELECT
  SUM(oi.quantity * oi.unit_price) AS sales,
  c.name
FROM order_items oi
INNER JOIN orders o
ON oi.order_id = o.order_id
INNER JOIN customers c
ON o.customer_id = c.customer_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2015
GROUP BY c.name
ORDER BY 1 DESC;
```

The 'Resultado de la Consulta' pane shows the results of the query, displaying a table with two columns: SALES and NAME. The results are sorted by SALES in descending order.

SALES	NAME
1468200,87	International Paper
1198331,59	Plains GP Holdings
1050939,97	Jabil Circuit
1043144,72	PPG Industries
893175,62	General Mills
630829,45	NGL Energy Partners
590302,91	AbbVie
576401,99	Nucor
499526,41	Alcoa
450561,12	US Foods Holding
422361,97	Community Health Systems

Para el año 2015 el cliente: International Paper es el que registra el gasto más alto con \$1.4M. Cabe aclarar algo, y es que aunque la base de datos contiene 319 registros de clientes, hay unos que aunque registrados, no hacen compras.

Query:

```
SELECT
  SUM(oi.quantity * oi.unit_price) AS sales,
  c.name
FROM order_items oi
INNER JOIN orders o
ON oi.order_id = o.order_id
INNER JOIN customers c
ON o.customer_id = c.customer_id
WHERE (EXTRACT(YEAR FROM o.order_date)) = 2015
GROUP BY c.name
ORDER BY 1 DESC;
```

6. ¿Cuánto ha facturado la compañía en cada uno de los años de los que tiene datos?

The screenshot shows a database management interface with a menu bar (Archivo, Editar, Ver, Navegar, Ejecutar, Origen, Equipo, Herramientas, Ventana, Ayuda) and several toolbars. On the left, a 'Conexiones' pane lists database objects like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main 'Hoja de Trabajo' pane contains a SQL query in the 'Generador de Consultas' tab. The 'Informes' pane on the bottom left lists various report types. The bottom right pane shows the 'Resultado de la Consulta' with a table of results.

```
SELECT
    SUM(oi.quantity * oi.unit_price) AS sales,
    EXTRACT(YEAR FROM o.order_date) AS year
FROM order_items oi
INNER JOIN orders o
ON oi.order_id = o.order_id
GROUP BY EXTRACT(YEAR FROM o.order_date)
ORDER BY 2 DESC;
```

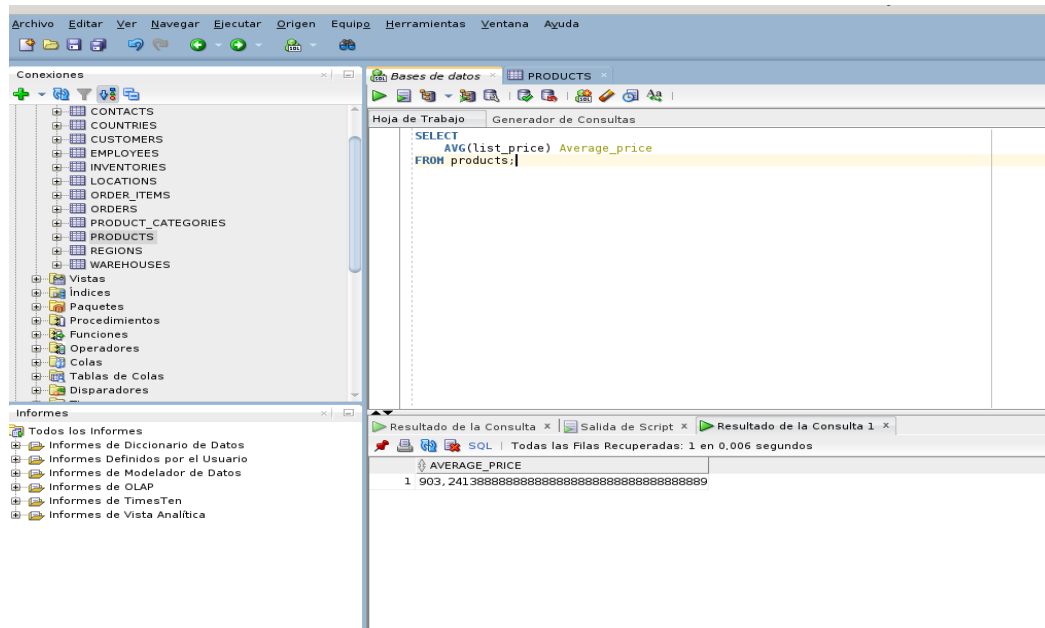
	SALES	YEAR
1	21914442,01	2017
2	21489606,1	2016
3	8823776,62	2015
4	514267,91	2013

Query:

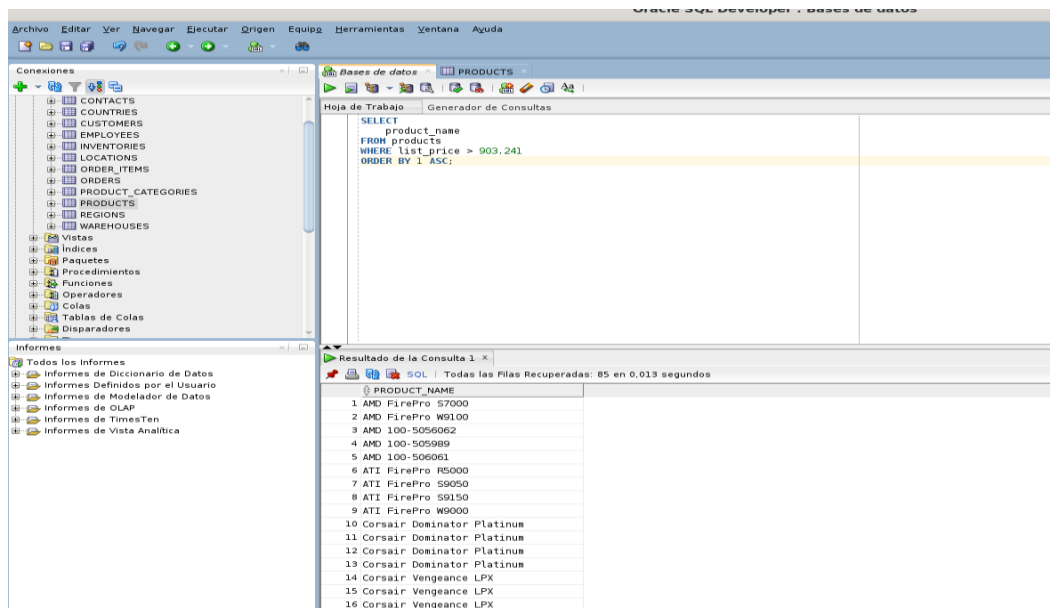
```
SELECT
    SUM(oi.quantity * oi.unit_price) AS sales,
    EXTRACT(YEAR FROM o.order_date) AS year
FROM order_items oi
INNER JOIN orders o
ON oi.order_id = o.order_id
GROUP BY EXTRACT(YEAR FROM o.order_date)
ORDER BY 2 DESC;
```


7. ¿Cuáles son los nombres de los productos cuyo precio es superior la media?

Aquí primero calculé la media de la lista de productos.



Luego con esa media, me di a la tarea de hacer un filtro con el `list_price`, con todos aquellos productos que tuviesen un precio superior a \$903.24 y el resultado me arrojó que hay 85 productos cuyo precio es mayor a la media y esos productos los ordené en forma ascendente.



Oracle SQL Developer : Bases de datos

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Conexiones

- CONTACTS
- COUNTRIES
- CUSTOMERS
- EMPLOYEES
- INVENTORIES
- LOCATIONS
- ORDER_ITEMS
- ORDERS
- PRODUCT_CATEGORIES
- PRODUCTS
- REGIONS
- WAREHOUSES
- Vistas
- Indices
- Paquetes
- Procedimientos
- Funciones
- Operadores
- Colas
- Tablas de Colas
- Disparadores

Informes

- Todos los Informes
- Informes de Diccionario de Datos
- Informes Definidos por el Usuario
- Informes de Modelador de Datos
- Informes de OLAP
- Informes de TimesTen
- Informes de Vista Analítica

Bases de datos

PRODUCTS

Hoja de Trabajo Generador de Consultas

```
SELECT
  product_name
FROM products
WHERE list_price > 903.241
ORDER BY 1 ASC;
```

Resultado de la Consulta 1 x

Todas las Filas Recuperadas: 85 en 0,013 segundos

PRODUCT_NAME
17 Corsair Vengeance LPX
18 Crucial
19 EVGA 120-P4-1999-KR
20 EVGA 120-P4-3992-KR
21 G.Skill Ripjaws V Series
22 G.Skill Ripjaws 4 Series
23 G.Skill Ripjaws 4 Series
24 G.Skill Trident Z
25 G.Skill Trident Z RGB
26 G.Skill TridentZ RGB
27 HP C2J95AT
28 Intel Core i7-4960X Extreme Edition
29 Intel Core i7-5960X
30 Intel Core i7-5960X (OEM/Tray)
31 Intel Core i7-6950X
32 Intel Core i7-6950X (OEM/Tray)

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Conexiones

- CONTACTS
- COUNTRIES
- CUSTOMERS
- EMPLOYEES
- INVENTORIES
- LOCATIONS
- ORDER_ITEMS
- ORDERS
- PRODUCT_CATEGORIES
- PRODUCTS
- REGIONS
- WAREHOUSES
- Vistas
- Indices
- Paquetes
- Procedimientos
- Funciones
- Operadores
- Colas
- Tablas de Colas
- Disparadores

Informes

- Todos los Informes
- Informes de Diccionario de Datos
- Informes Definidos por el Usuario
- Informes de Modelador de Datos
- Informes de OLAP
- Informes de TimesTen
- Informes de Vista Analítica

Bases de datos

PRODUCTS

Hoja de Trabajo Generador de Consultas

```
SELECT
  product_name
FROM products
WHERE list_price > 903.241
ORDER BY 1 ASC;
```

Resultado de la Consulta 1 x

Todas las Filas Recuperadas: 85 en 0,013 segundos

PRODUCT_NAME
32 Intel Core i7-6950X (OEM/Tray)
33 Intel Core i7-990X Extreme Edition
34 Intel Core i9-7900X
35 Intel Core 2 Extreme QX6800
36 Intel SSDPECKME040T401
37 Intel Xeon E5-1660 V3 (OEM/Tray)
38 Intel Xeon E5-1680 V3 (OEM/Tray)
39 Intel Xeon E5-2470V2
40 Intel Xeon E5-2637 V2 (OEM/Tray)
41 Intel Xeon E5-2643 V2 (OEM/Tray)
42 Intel Xeon E5-2643 V3 (OEM/Tray)
43 Intel Xeon E5-2643 V4 (OEM/Tray)
44 Intel Xeon E5-2650
45 Intel Xeon E5-2650 V2
46 Intel Xeon E5-2650 V3
47 Intel Xeon E5-2650 V3 (OEM/Tray)

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Bases de datos PRODUCTS

Hoja de Trabajo Generador de Consultas

```
SELECT
    product_name
FROM products
WHERE list_price > 903.241
ORDER BY 1 ASC;
```

Resultado de la Consulta 1

SQL | Todas las Filas Recuperadas: 85 en 0.013 segundos

PRODUCT_NAME
47 Intel Xeon ES-2650 V3 (OEM/Tray)
48 Intel Xeon ES-2650 V4
49 Intel Xeon ES-2650L V3 (OEM/Tray)
50 Intel Xeon ES-2660 V3
51 Intel Xeon ES-2660 V3 (OEM/Tray)
52 Intel Xeon ES-2660 V4
53 Intel Xeon ES-2667 V3 (OEM/Tray)
54 Intel Xeon ES-2670 V3
55 Intel Xeon ES-2680
56 Intel Xeon ES-2690 V3 (OEM/Tray)
57 Intel Xeon ES-2690 V4
58 Intel Xeon ES-2683 V4
59 Intel Xeon ES-2683 V4 (OEM/Tray)
60 Intel Xeon ES-2685 V3 (OEM/Tray)
61 Intel Xeon ES-2687M V3
62 Intel Xeon ES-2687M V4

Oracle SQL Developer : bases de datos

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Bases de datos PRODUCTS

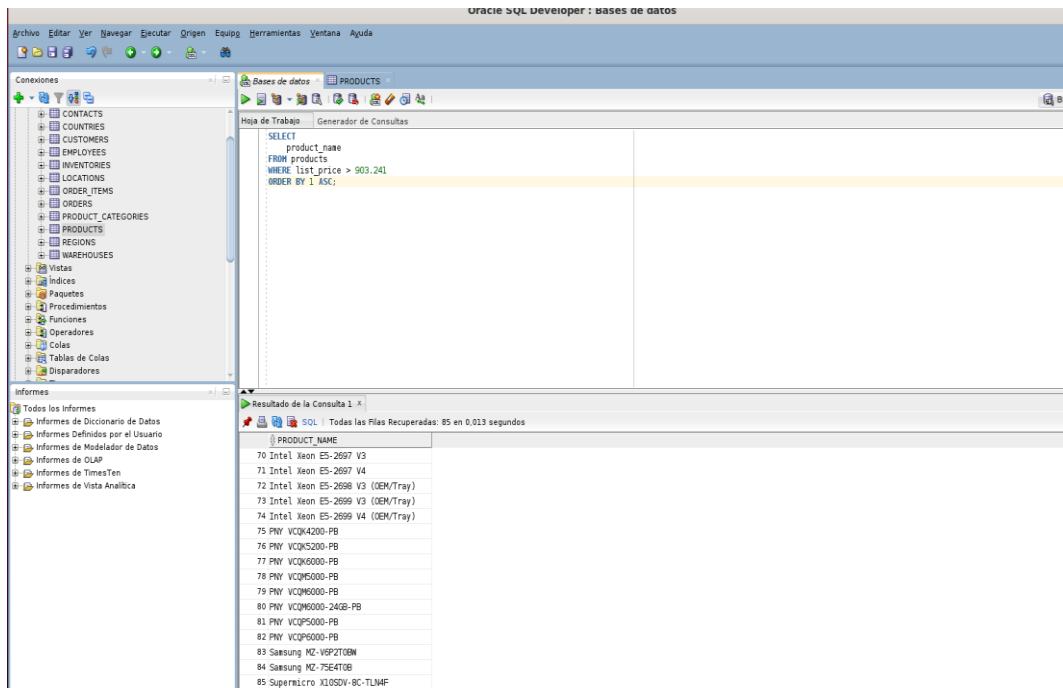
Hoja de Trabajo Generador de Consultas

```
SELECT
    product_name
FROM products
WHERE list_price > 903.241
ORDER BY 1 ASC;
```

Resultado de la Consulta 1

SQL | Todas las Filas Recuperadas: 85 en 0.013 segundos

PRODUCT_NAME
62 Intel Xeon ES-2687M V4
63 Intel Xeon ES-2690 (OEM/Tray)
64 Intel Xeon ES-2690 V3
65 Intel Xeon ES-2690 V4
66 Intel Xeon ES-2695 V2
67 Intel Xeon ES-2695 V3 (OEM/Tray)
68 Intel Xeon ES-2695 V4
69 Intel Xeon ES-2697 V2
70 Intel Xeon ES-2697 V3
71 Intel Xeon ES-2697 V4
72 Intel Xeon ES-2698 V3 (OEM/Tray)
73 Intel Xeon ES-2699 V3 (OEM/Tray)
74 Intel Xeon ES-2699 V4 (OEM/Tray)
75 PNY VCQK4200-PB
76 PNY VCQK5200-PB
77 PNY VCQK6000-PB



Queries:

```
SELECT
    AVG(list_price) Average_price
FROM products;
```

```
SELECT
    product_name
FROM products
WHERE list_price > 903,241
ORDER BY 1 ASC;
```

8. ¿Cuáles son los empleados (nombre y apellido) que han vendido más de 50K durante 2017?

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane shows the 'Bases de datos' tree with various tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main window displays a SQL query in the 'Hoja de Trabajo' tab. The query is as follows:

```
SELECT
  e.first_name,
  e.last_name,
  SUM(oi.quantity * oi.unit_price) AS total_sales
FROM employees e
INNER JOIN orders o
  ON e.employee_id = o.salesman_id
INNER JOIN order_items oi
  ON o.order_id = oi.order_id
WHERE EXTRACT(YEAR FROM o.order_date)= 2017
GROUP BY e.first_name, e.last_name
HAVING SUM(oi.quantity * oi.unit_price) > 50000
ORDER BY 3 DESC;
```

Below the query, the 'Resultado de la Consulta' pane shows the results of the query. It indicates that 9 rows were recovered in 0.088 seconds. The results are as follows:

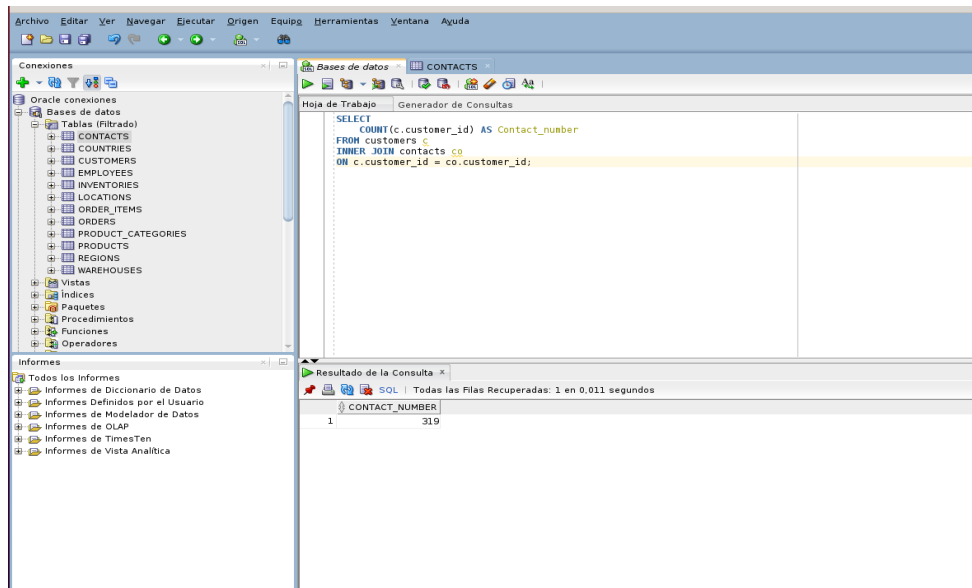
	FIRST_NAME	LAST_NAME	TOTAL_SALES
1	Freya	Gomez	3768957.03
2	Isabelle	Marshall	2692044.24
3	Grace	Ellis	1990776.95
4	Daisy	Ortiz	1586804.44
5	Evie	Harrison	1335817.98
6	Scarlett	Gibson	1310556.91
7	Lily	Fisher	1160350.79
8	Florence	Freeman	545057.39
9	Chloe	Cruz	277585.23

Query:

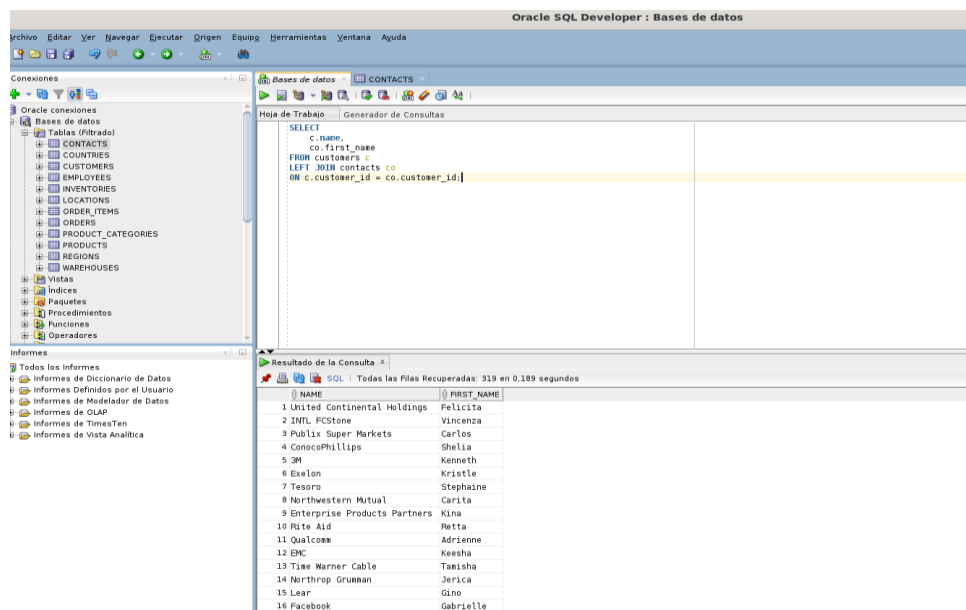
```
SELECT
  e.first_name,
  e.last_name,
  SUM(oi.quantity * oi.unit_price) AS total_sales
FROM employees e
INNER JOIN orders o
  ON e.employee_id = o.salesman_id
INNER JOIN order_items oi
  ON o.order_id = oi.order_id
WHERE EXTRACT(YEAR FROM o.order_date)= 2017
GROUP BY e.first_name, e.last_name
HAVING SUM(oi.quantity * oi.unit_price) > 50000
ORDER BY 3 DESC;
```

9. ¿Cuánto clientes no tienen persona de contacto?

Aquí me di cuenta que necesitaba hacer un join para ver si alguno de esos clientes no tenía persona de contacto, entonces el primero que hice fue un INNER JOIN e hice un conteo para ver si tanto en la tabla de “customers” como en la de “contacts” había correspondencia y sí había, 319 registros arrojó (misma cantidad que se dio en la table inicial) es decir, todos los clientes tienen persona de contacto.



Sin embargo quise comprobar con un left join a ver si había correspondencia de la table “customers” con la tabla “contacts”.



Oracle SQL Developer : bases de datos

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

Bases de datos

Tablas (filtrado)

CONTACTS

COUNTRIES

CUSTOMERS

EMPLOYEES

INVENTORIES

LOCATIONS

ORDER_ITEMS

ORDERS

PRODUCT_CATEGORIES

PRODUCTS

REGIONS

WAREHOUSES

Vistas

Indice

Paquetes

Procedimientos

Funciones

Operadores

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analitica

Bases de datos

CONTACTS

Hoja de Trabajo

Generador de Consultas

```

SELECT
  c.name,
  co.first_name
FROM customers c
LEFT JOIN contacts co
ON c.customer_id = co.customer_id;

```

Resultado de la Consulta

Todas las Filas Recuperadas: 319 en 0.189 segundos

NAME	FIRST_NAME
17 Supervalu	Isreal
18 NextEra Energy	Gertrude
19 PG&E Corp.	Lachon
20 Goodyear Tire & Rubber	Melany
21 Micron Technology	Tennie
22 ConAgra Foods	Birgit
23 Bank of New York Mellon Corp.	Jessika
24 Genuine Parts	Nada
25 Omnicom Group	Calandra
26 Monsanto	Josh
27 National Oilwell Varco	Fae
28 Marriott International	Lasonya
29 Kinder Morgan	Kenton
30 Molina Healthcare	Kandi
31 Lincoln National	Vida
32 C.H. Robinson Worldwide	Shaeika

Oracle SQL Developer : bases de datos

Archivo Editar Ver Navegar Ejecutar Origen Equip Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

Bases de datos

Tablas (filtrado)

CONTACTS

COUNTRIES

CUSTOMERS

EMPLOYEES

INVENTORIES

LOCATIONS

ORDER_ITEMS

ORDERS

PRODUCT_CATEGORIES

PRODUCTS

REGIONS

WAREHOUSES

Vistas

Indice

Paquetes

Procedimientos

Funciones

Operadores

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analitica

Bases de datos

CONTACTS

Hoja de Trabajo

Generador de Consultas

```

SELECT
  c.name,
  co.first_name
FROM customers c
LEFT JOIN contacts co
ON c.customer_id = co.customer_id;

```

Resultado de la Consulta

Todas las Filas Recuperadas: 319 en 0.189 segundos

NAME	FIRST_NAME
304 Caseron International	Amber
305 Xcel Energy	Maryrose
306 State Farm Insurance Cos.	Janeen
307 Johnson & Johnson	Debra
308 Marathon Petroleum	Leandro
309 Aetna	Philomena
310 AIG	Lauren
311 Intel	Ha
312 Pfizer	Elvera
313 FedEx	Mae
314 New York Life Insurance	Heike
315 Ingram Micro	Glinda
316 American Airlines Group	Jamee
317 Johnson Controls	Solange
318 Goldman Sachs Group	Christal
319 Oracle	Yolanda

Al final comprobé que efectivamente en los 319 registros de la tabla “customers” todos tenían su correspondencia con la tabla de “contacts” y no había ningún registro nulo o huérfano.

Query:

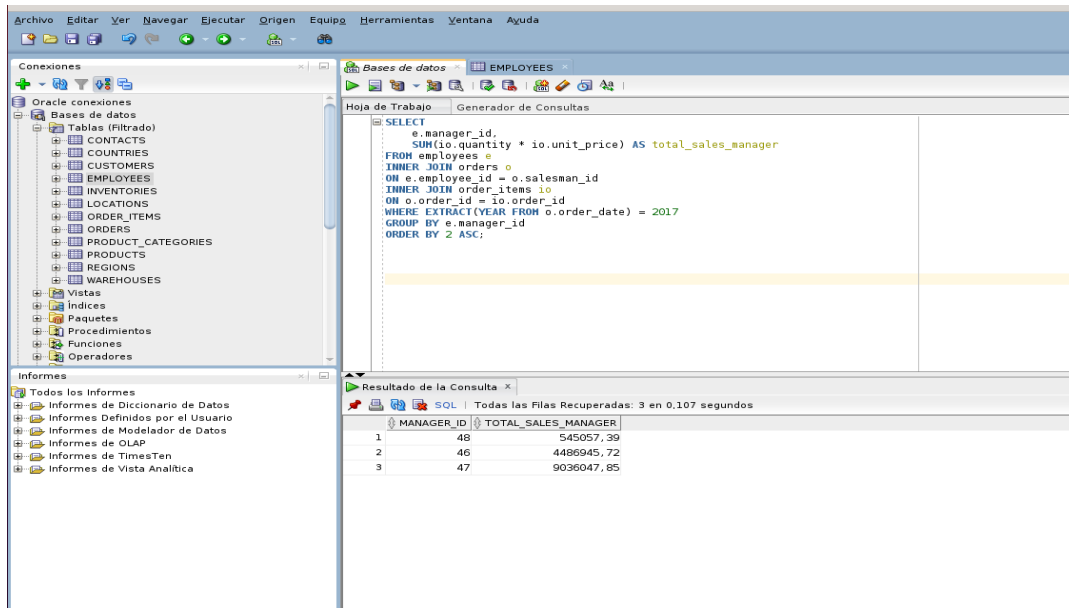
```
SELECT
    COUNT(c.CUSTOMER_ID) AS Contact_number
FROM customers c
INNER JOIN contacts co
ON c.customer_id = co.customer_id
```

```
SELECT
    c.name,
    co.first_name
FROM customers c
LEFT JOIN contacts co
ON c.customer_id = co.customer_id
```

10. ¿Cuál es el Manager (nombre y apellido identificado por el campo manager_id) que menos ha vendido durante 2017?

El manager que menos vendió durante el 2017 fue el manager identificado con el número o registro “48” (en la tabla de employees, no tenía ni el nombre ni el apellido, este correspondía al nombre y apellido del vendedor).

Sin embargo, luego de encontrar el resultado me generó curiosidad porqué para el 2017 solo tengo 3 managers, 46, 47 y 48, y en la tabla habían más registros de managers, entonces hice otras consultas.



The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Conexiones' (Connections) tree with 'Bases de datos' (Databases) expanded, showing various tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main pane shows a SQL query in the 'Hoja de Trabajo' (Worksheet) tab:

```
SELECT
  e.manager_id,
  SUM(io.quantity * io.unit_price) AS total_sales_manager
FROM employees e
INNER JOIN orders o
ON e.employee_id = o.salesman_id
INNER JOIN order_items io
ON o.order_id = io.order_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2017
GROUP BY e.manager_id
ORDER BY 2 ASC;
```

The bottom pane shows the 'Resultado de la Consulta' (Query Result) with the following data:

MANAGER_ID	TOTAL_SALES_MANAGER
48	545057,39
46	4486945,72
47	9036047,85

Con esta consulta me quise cerciorar si todos los empleados vendieron en 2017, ya que vi muchos vendedores en la tabla con diferentes managers que no eran efectivamente los 46, 47 y 48 y me di cuenta de que en realidad solo unos cuantos empleados vendieron en 2017, no todos los que aparecen en la tabla.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Oracle conexiones' tree with 'Bases de datos' expanded, showing various tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main pane shows a SQL query in the 'Hoja de Trabajo' tab:

```
SELECT
  e.employee_id,
  COUNT(o.order_id) AS Number_of_sales
FROM employees e
LEFT JOIN orders o
ON e.employee_id = o.salesman_id
INNER JOIN order_items io
ON o.order_id = io.order_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2017
GROUP BY e.employee_id
ORDER BY 1;
```

The bottom pane shows the 'Resultado de la Consulta' with the following data:

EMPLOYEE_ID	NUMBER_OF_SALES
1	54
2	55
3	56
4	57
5	59
6	60
7	61
8	62
9	64

Luego lo que hice fue comprobar si esos empleados le reportaban a alguno de los tres managers (46, 47, 48) y efectivamente ellos le reportaban a alguno de los mencionados.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Oracle conexiones' tree with 'Bases de datos' expanded, showing various tables like CONTACTS, COUNTRIES, CUSTOMERS, EMPLOYEES, INVENTORIES, LOCATIONS, ORDER_ITEMS, ORDERS, PRODUCT_CATEGORIES, PRODUCTS, REGIONS, and WAREHOUSES. The main pane shows a SQL query in the 'Hoja de Trabajo' tab:

```
SELECT
  employee_id,
  manager_id
FROM employees
WHERE employee_id IN (54,55,56,57,59,60,61,62,64);
```

The bottom pane shows the 'Resultado de la Consulta' with the following data:

EMPLOYEE_ID	MANAGER_ID
1	54
2	55
3	56
4	57
5	59
6	60
7	61
8	62
9	64

Queries:

```
SELECT
    e.manager_id,
    SUM(io.quantity * io.unit_price) AS total_sales_manager
FROM employees e
INNER JOIN orders o
ON e.employee_id = o.salesman_id
INNER JOIN order_items io
ON o.order_id = io.order_id
WHERE EXTRACT(YEAR FROM o.order_date)= 2017
GROUP BY e.manager_id
ORDER BY 2 ASC;
```

```
SELECT
    e.employee_id,
    COUNT(o.order_id) AS Number_of_sales
FROM employees e
LEFT JOIN orders o
ON e.employee_id = o.salesman_id
INNER JOIN order_items io
ON o.order_id = io.order_id
WHERE EXTRACT(YEAR FROM o.order_date) = 2017
GROUP BY e.employee_id
ORDER BY 1;
```

```
SELECT
    employee_id,
    manager_id
FROM employees
WHERE employee_id IN (54,55,56,57,59,60,61,62,64) ;
```

----- FIN -----