



Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
Bases de Datos Avanzadas PF-3861

Practica aplicada #3
Bases de datos distribuidas

Estudiante:
Gabriel Umaña Frías C09913

Profesor:
Msc. Richard Delgado

II Ciclo 2020

Resultados:

Primera Parte - Configuración del Ambiente Base para parte 2

1. Se utiliza la cuenta de Oracle Cloud Academy para crear una máquina virtual denominada oracle_xe con la imagen de Oracle Linux y se procede a instalar la base de datos Oracle XE 18c:

Create instance									
Name	State	Public IP	Private IP	Shape	OCPU count	Memory (GB)	Availability domain	Fault domain	Created
oracle_xe	Running	140.238.178.191	10.0.2.173	VM.Standard.E3.Flex	2	8	AD-1	FD-2	Wed, Nov 10, 2021, 02:45:51 UTC
oci-oraclelinuxdb1	Stopped	140.238.238.217	10.0.2.150	VM.Optimized3.Flex	2	4	AD-1	FD-3	Tue, Nov 9, 2021, 01:05:36 UTC

Showing 2 Items < 1 of 1 >

2. Confirmamos que Oracle XE está corriendo:

```
gabriel@Gabriels-MacBook-Air [PF-3861] > ssh -i ~/.ssh/id_rsa opc@140.238.178.191
[opc@oracle-xe ~]$ sqlplus sys/Practica3 as sysdba

SQL*Plus: Release 18.0.0.0.0 - Production on Wed Nov 10 03:53:45 2021 Version 18.4.0.0.0

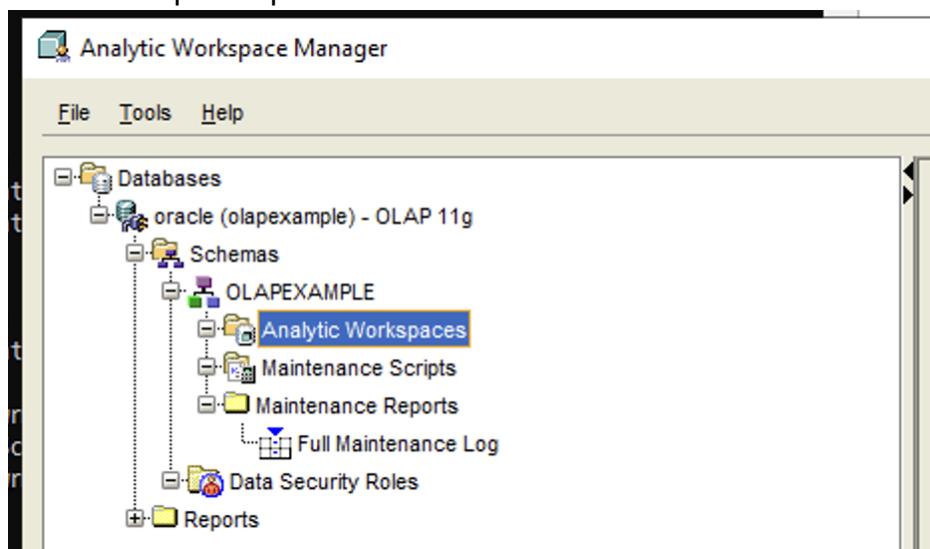
Copyright (c) 1982, 2018, Oracle. All rights reserved.

Connected to:
Oracle Database 18c Express Edition Release 18.0.0.0.0 - Production Version 18.4.0.0.0
```

3. Abrimos el puerto 1521 en el firewall:

```
[opc@oracle-xe ~]$ sudo firewall-cmd --zone=public --add-port=1521/tcp
success
```

4. Por último utilizando el programa Oracle Analytic Workspace, corriendo en una máquina virtual Windows, para conectarnos a la base de datos con el usuario "olapexample":




Primera Parte - Configuración del Ambiente Base para parte 3

1. Se utiliza la cuenta de Oracle Cloud Academy para crear un autonomous database denominada DBOLAP:

Create Autonomous Database							
Display Name	State	Dedicated	OCPUs	Storage	Workload Type	Autonomous Data Guard	Created
DBOLAP <small>Always Free</small>	Available	No	1	20 GB	Data Warehouse	—	Thu, Nov 11, 2021, 06:58:24 UTC

2. Una vez que la base de datos se encuentra provisionada, se crea un nuevo usuario llamado STARUSER:

Current User



ADMIN

REST Enabled

Graph Enabled

ORDS Alias: admin

Last Login: 11/11/2021, 7:01:28 AM

Password Expires in 359 days


https://g3236a0b7623359-dbolap.adb.sa-saopaulo-1.oraclecloudapps.com/ords/admin/_sdw/

All Users

STARUSER

Filter by

Sort by



STARUSER

REST Enabled

ORDS Alias: staruser

Password Expires in 359 days

https://g3236a0b7623359-dbolap.adb.sa-saopaulo-1.oraclecloudapps.com/ords/...

3. El siguiente paso es conectarse a la base de datos con el usuario STARUSER, mediante SQL Developer, usando el cloud wallet. Probamos que este funcione correctamente mediante el botón "Test":

New / Select Database Connection

Connection Name

Connection Det...

Name

DBOLAP

Color

Database Type

Oracle

User Info

Proxy User

Authentication Type

Default

Username

STARUSER

Role

default

Password

.....

Save Password

Connection Type

Cloud Wallet

Details

Advanced

Proxy

Configuration File

/Users/gabriel/Wallet_DBOLAP.zip

Browse...

Service

dbolap_high

Configure OQS Classic

Status : Success

Help

Save

Clear

Test

Connect

Cancel

4. Por último, se cargan todos los datos a la base de datos DBOLAP:

Worksheet | Query Builder

```
select count(*) as total_rows from channels;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.275 seconds

TOTAL_ROWS
8

Run Statement (^+↵) | idler

```
select count(*) as total_rows from customers;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.268 seconds

TOTAL_ROWS
64072

Worksheet | Query Builder

```
select count(*) as total_rows from times;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.279 seconds

TOTAL_ROWS
1096

Welcome Page | DBOLAP x

Welcome Page

Worksheet | Query Builder

```
select count(*) as total_rows from products;
```

Query Result x

SQL | All Rows Fetched: 1 in 0.287 seconds

TOTAL_ROWS
2174

```
select count(*) as total_rows from sales_fact;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 1 in 0.305 seconds

TOTAL_ROWS
277069

Segunda Parte - Resultados de configuración de cubo OLAP

1. Mediante Oracle Analytic Workspace Manager, se crea un cubo OLAP y se cargan los datos necesarios para hacer que este funcione.
2. Una vez que se han creado los objetos y cargado los datos al cubo, se usa el OLAP Worksheet para producir un reporte que cumpla los siguientes objetivos:
 - a. Obtenga los 3 productos a nivel de ITEM con mayores ventas considerando AMOUNT, y reporte a nivel de region y a nivel de año las unidades vendidas en año anterior y su porcentaje de crecimiento:

Oracle OLAP Worksheet - 12.2.0.1.0 - Connected to //152.67.54.2:1521/XEPDB1 104.211.53.54

File Edit Worksheet Options Help

```
limit products to products_levelrel 'ITEM'  
->limit periods to periods_levelrel 'YEAR'  
->limit geography to geography_levelrel 'REGION'  
->limit products keep top 3 basedon sales_cube_amount  
->report down periods heading 'Units' sales_cube_units heading 'Last Year' lag(sales_cube_units 1 periods) heading 'Growth' lagpct(sales_cube_units 1 periods)
```

PRODUCTS: DB

PERIODS	America			Asia			Europe			Pacific		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2017	2,926.00	NA	NA	1,377.00	NA	NA	2,124.00	NA	NA	927.00	NA	NA
2018	12,640.00	2,926.00	3.32	4,444.00	1,377.00	2.23	7,394.00	2,124.00	2.48	5,391.00	927.00	4.82
2019	11,506.00	12,640.00	-0.09	4,627.00	4,444.00	0.04	5,595.00	7,394.00	-0.24	5,127.00	5,391.00	-0.05
2020	12,636.00	11,506.00	0.10	4,027.00	4,627.00	-0.13	8,575.00	5,595.00	0.53	3,482.00	5,127.00	-0.32
2021	8,051.00	12,636.00	-0.36	4,303.00	4,027.00	0.07	5,749.00	8,575.00	-0.33	4,006.00	3,482.00	0.15

PRODUCTS: DBSVC

PERIODS	America			Asia			Europe			Pacific		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2017	2,777.00	NA	NA	1,083.00	NA	NA	1,801.00	NA	NA	1,111.00	NA	NA
2018	12,277.00	2,777.00	3.42	5,017.00	1,083.00	3.63	7,639.00	1,801.00	3.24	4,577.00	1,111.00	3.12
2019	13,268.00	12,277.00	0.08	5,572.00	5,017.00	0.11	6,807.00	7,639.00	-0.11	5,215.00	4,577.00	0.14
2020	11,599.00	13,268.00	-0.13	4,749.00	5,572.00	-0.15	7,156.00	6,807.00	0.05	3,849.00	5,215.00	-0.26
2021	9,198.00	11,599.00	-0.21	4,518.00	4,749.00	-0.05	6,009.00	7,156.00	-0.16	3,364.00	3,849.00	-0.13

PRODUCTS: SRVR

PERIODS	America			Asia			Europe			Pacific		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2017	2,935.00	NA	NA	990.00	NA	NA	1,949.00	NA	NA	983.00	NA	NA
2018	12,486.00	2,935.00	3.25	4,687.00	990.00	3.73	6,712.00	1,949.00	2.44	4,157.00	983.00	3.23
2019	12,013.00	12,486.00	-0.04	4,322.00	4,687.00	-0.08	7,049.00	6,712.00	0.05	5,085.00	4,157.00	0.22
2020	12,574.00	12,013.00	0.05	4,546.00	4,322.00	0.05	8,310.00	7,049.00	0.18	4,690.00	5,085.00	-0.08
2021	8,377.00	12,574.00	-0.33	3,129.00	4,546.00	-0.31	5,565.00	8,310.00	-0.33	4,438.00	4,690.00	-0.05

->

```
limit products to products_levelrel 'ITEM';  
limit periods to periods_levelrel 'YEAR';  
limit geography to geography_levelrel 'REGION';  
limit products keep top 3 basedon sales_cube_amount;  
report down periods heading 'Units' sales_cube_units heading 'Last Year'  
lag(sales_cube_units 1 periods) heading 'Growth' lagpct(sales_cube_units 1 periods);
```

- b. Modifique el reporte anterior para que solamente se muestren los 3 últimos años:

104.211.53.54

File Edit Worksheet Options Help

```

limit products to products_levelrel 'ITEM'
->limit periods to periods_levelrel 'YEAR'
->limit geography to geography_levelrel 'REGION'
->limit periods keep last 3
->limit products keep top 3 basedon sales_cube_amount

->report down periods heading 'Units' sales_cube_units heading 'Last Year' lag(sales_cube_units 1 periods) heading 'Growth' lagpct(sales_cube_units 1 periods)

```

PRODUCTS: DBSVC

PERIODS	--America--			--Asia--			--Europe--			--Pacific--		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2019	13,268.00	12,277.00	0.08	5,572.00	5,017.00	0.11	6,807.00	7,639.00	-0.11	5,215.00	4,577.00	0.14
2020	11,599.00	13,268.00	-0.13	4,749.00	5,572.00	-0.15	7,156.00	6,807.00	0.05	3,849.00	5,215.00	-0.26
2021	9,198.00	11,599.00	-0.21	4,518.00	4,749.00	-0.05	6,009.00	7,156.00	-0.16	3,364.00	3,849.00	-0.13

PRODUCTS: MWARE

PERIODS	--America--			--Asia--			--Europe--			--Pacific--		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2019	11,761.00	11,063.00	0.06	5,003.00	3,797.00	0.32	6,353.00	7,638.00	-0.17	4,478.00	4,994.00	-0.10
2020	11,229.00	11,761.00	-0.05	5,076.00	5,003.00	0.01	7,116.00	6,353.00	0.12	4,874.00	4,478.00	0.09
2021	8,930.00	11,229.00	-0.20	2,986.00	5,076.00	-0.41	5,042.00	7,116.00	-0.29	3,373.00	4,874.00	-0.31

PRODUCTS: VM

PERIODS	--America--			--Asia--			--Europe--			--Pacific--		
	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth	Units	Last Year	Growth
2019	13,786.00	11,421.00	0.21	5,916.00	4,420.00	0.34	7,810.00	8,363.00	-0.07	3,486.00	5,259.00	-0.34
2020	12,325.00	13,786.00	-0.11	3,985.00	5,916.00	-0.33	7,198.00	7,810.00	-0.08	5,480.00	3,486.00	0.57
2021	7,094.00	12,325.00	-0.42	3,947.00	3,985.00	-0.01	5,066.00	7,198.00	-0.30	3,797.00	5,480.00	-0.31

->

```

limit products to products_levelrel 'ITEM';
limit periods to periods_levelrel 'YEAR';
limit geography to geography_levelrel 'REGION';
limit periods keep last 3;
limit products keep top 3 basedon sales_cube_amount;
report down periods heading 'Units' sales_cube_units heading 'Last Year'
lag(sales_cube_units 1 periods) heading 'Growth' lagpct(sales_cube_units 1
periods)

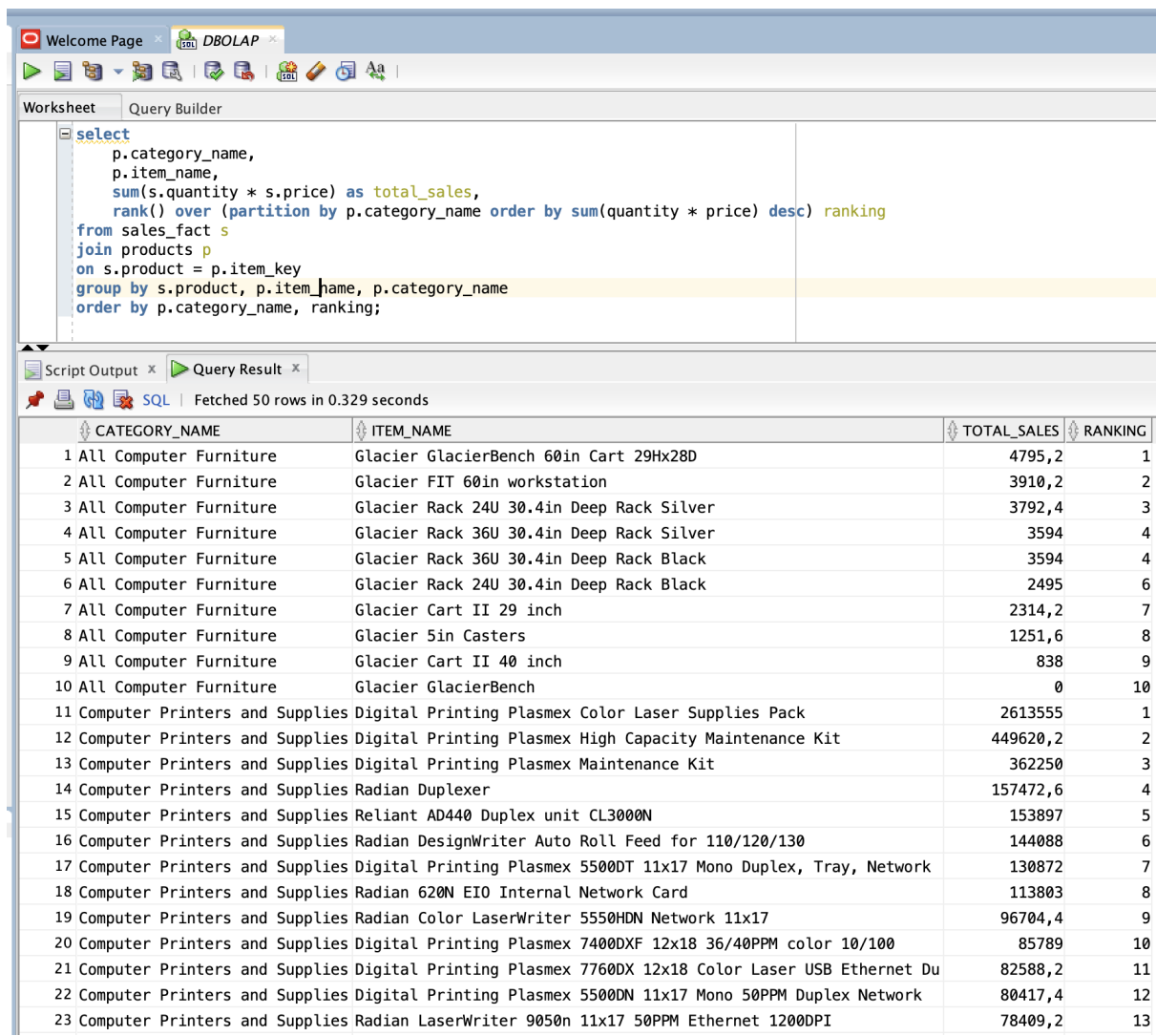
```

Tercera Parte - Resultados de ejecución de consultas analíticas en star schema

Utilizando la base de datos DBOLAP y el usuario STARUSER, utilice SQL Analytics para responder a las siguientes preguntas.

1. Obtenga el ranking de cada producto dentro de su categoría de acuerdo a las ventas totales (de mayor a menor), considerando todos los años. Las ventas = $QUANTITY * PRICE$. Ordene el resultado por categoría y ranking. Su reporte debería verse más o menos así:

```
select
  p.category_name,
  p.item_name,
  sum(s.quantity * s.price) as total_sales,
  rank() over (partition by p.category_name order by
sum(quantity * price) desc) ranking
from sales_fact s
join products p
on s.product = p.item_key
group by s.product, p.item_name, p.category_name
order by p.category_name, ranking;
```

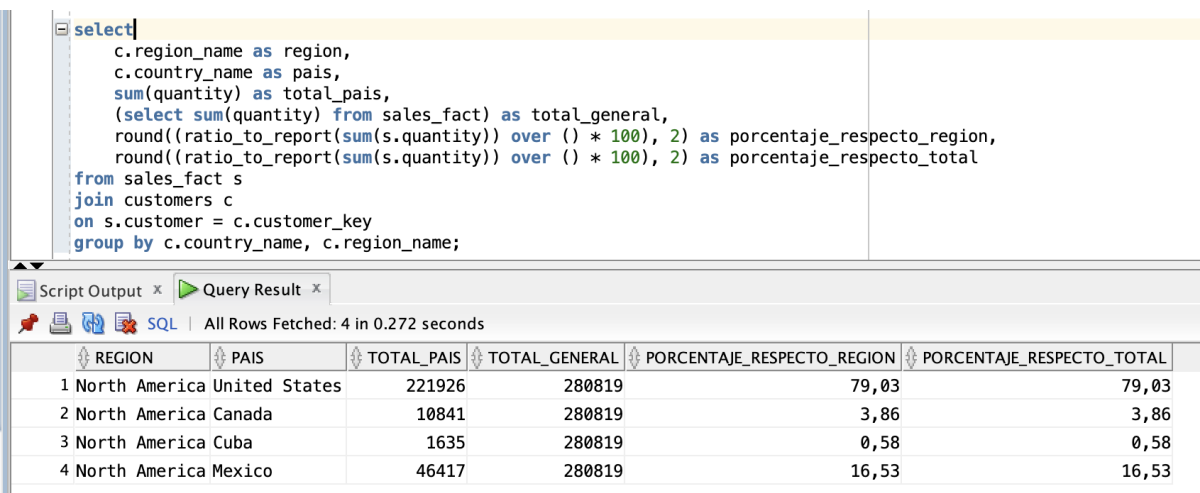


The screenshot shows the DBOLAP SQL Analytics interface. The top bar includes a 'Welcome Page' and 'DBOLAP' tabs. Below the toolbar, the 'Query Builder' tab is active, displaying the SQL query. The 'Script Output' and 'Query Result' tabs are also visible. The 'Query Result' tab shows the execution results, indicating that 50 rows were fetched in 0.329 seconds. The results are displayed in a table with four columns: CATEGORY_NAME, ITEM_NAME, TOTAL_SALES, and RANKING. The table lists 23 rows of data, showing the ranking of products within their categories based on total sales.

CATEGORY_NAME	ITEM_NAME	TOTAL_SALES	RANKING
1 All Computer Furniture	Glacier GlacierBench 60in Cart 29Hx28D	4795,2	1
2 All Computer Furniture	Glacier FIT 60in workstation	3910,2	2
3 All Computer Furniture	Glacier Rack 24U 30.4in Deep Rack Silver	3792,4	3
4 All Computer Furniture	Glacier Rack 36U 30.4in Deep Rack Silver	3594	4
5 All Computer Furniture	Glacier Rack 36U 30.4in Deep Rack Black	3594	4
6 All Computer Furniture	Glacier Rack 24U 30.4in Deep Rack Black	2495	6
7 All Computer Furniture	Glacier Cart II 29 inch	2314,2	7
8 All Computer Furniture	Glacier 5in Casters	1251,6	8
9 All Computer Furniture	Glacier Cart II 40 inch	838	9
10 All Computer Furniture	Glacier GlacierBench	0	10
11 Computer Printers and Supplies	Digital Printing Plasmex Color Laser Supplies Pack	2613555	1
12 Computer Printers and Supplies	Digital Printing Plasmex High Capacity Maintenance Kit	449620,2	2
13 Computer Printers and Supplies	Digital Printing Plasmex Maintenance Kit	362250	3
14 Computer Printers and Supplies	Radian Duplexer	157472,6	4
15 Computer Printers and Supplies	Reliant AD440 Duplex unit CL3000N	153897	5
16 Computer Printers and Supplies	Radian DesignWriter Auto Roll Feed for 110/120/130	144088	6
17 Computer Printers and Supplies	Digital Printing Plasmex 5500DT 11x17 Mono Duplex, Tray, Network	130872	7
18 Computer Printers and Supplies	Radian 620N EIO Internal Network Card	113803	8
19 Computer Printers and Supplies	Radian Color LaserWriter 5550HDN Network 11x17	96704,4	9
20 Computer Printers and Supplies	Digital Printing Plasmex 7400DXF 12x18 36/40PPM color 10/100	85789	10
21 Computer Printers and Supplies	Digital Printing Plasmex 7760DX 12x18 Color Laser USB Ethernet Du	82588,2	11
22 Computer Printers and Supplies	Digital Printing Plasmex 5500DN 11x17 Mono 50PPM Duplex Network	80417,4	12
23 Computer Printers and Supplies	Radian LaserWriter 9050n 11x17 50PPM Ethernet 1200DPI	78409,2	13

2. Obtenga el porcentaje del total que representan las unidades vendidas de cada país con respecto al total de la región, y con respecto al total general. Incluya el nombre de la región, del país y el total de las unidades vendidas en su reporte, además de los porcentajes por supuesto.

```
select
  c.region_name as region,
  c.country_name as pais,
  sum(quantity) as total_pais,
  (select sum(quantity) from sales_fact) as total_general,
  round((ratio_to_report(sum(s.quantity)) over () * 100), 2) as
porcentaje_respecto_region,
  round((ratio_to_report(sum(s.quantity)) over () * 100), 2) as
porcentaje_respecto_total
from sales_fact s
join customers c
on s.customer = c.customer_key
group by c.country_name, c.region_name;
```

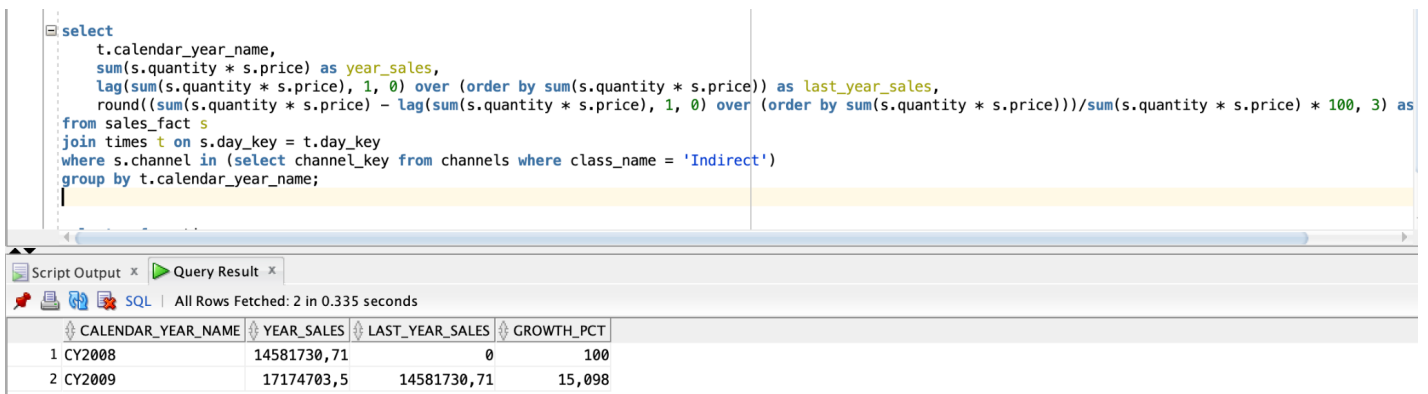


The screenshot shows a SQL query editor with the query text in the top pane and a query result table in the bottom pane. The query is identical to the one in the previous block. The result table has 6 columns: REGION, PAIS, TOTAL_PAIS, TOTAL_GENERAL, PORCENTAJE_RESPECTO_REGION, and PORCENTAJE_RESPECTO_TOTAL. It contains 4 rows of data.

REGION	PAIS	TOTAL_PAIS	TOTAL_GENERAL	PORCENTAJE_RESPECTO_REGION	PORCENTAJE_RESPECTO_TOTAL
1 North America	United States	221926	280819	79,03	79,03
2 North America	Canada	10841	280819	3,86	3,86
3 North America	Cuba	1635	280819	0,58	0,58
4 North America	Mexico	46417	280819	16,53	16,53

- Obtenga el porcentaje de crecimiento de las ventas (unidades * precio) de cada año comparado con el año anterior para todos los años, considerando solamente los canales de ventas indirectos. Además del porcentaje de crecimiento, incluya para cada año el monto de las ventas del año anterior.

```
select
    t.calendar_year_name,
    sum(s.quantity * s.price) as year_sales,
    lag(sum(s.quantity * s.price), 1, 0) over (order by sum(s.quantity *
s.price)) as last_year_sales,
    round((sum(s.quantity * s.price) - lag(sum(s.quantity * s.price), 1,
0) over (order by sum(s.quantity * s.price))))/sum(s.quantity * s.price)
* 100, 3) as growth_pct
from sales_fact s
join times t on s.day_key = t.day_key
where s.channel in (select channel_key from channels where class_name =
'Indirect')
group by t.calendar_year_name;
```

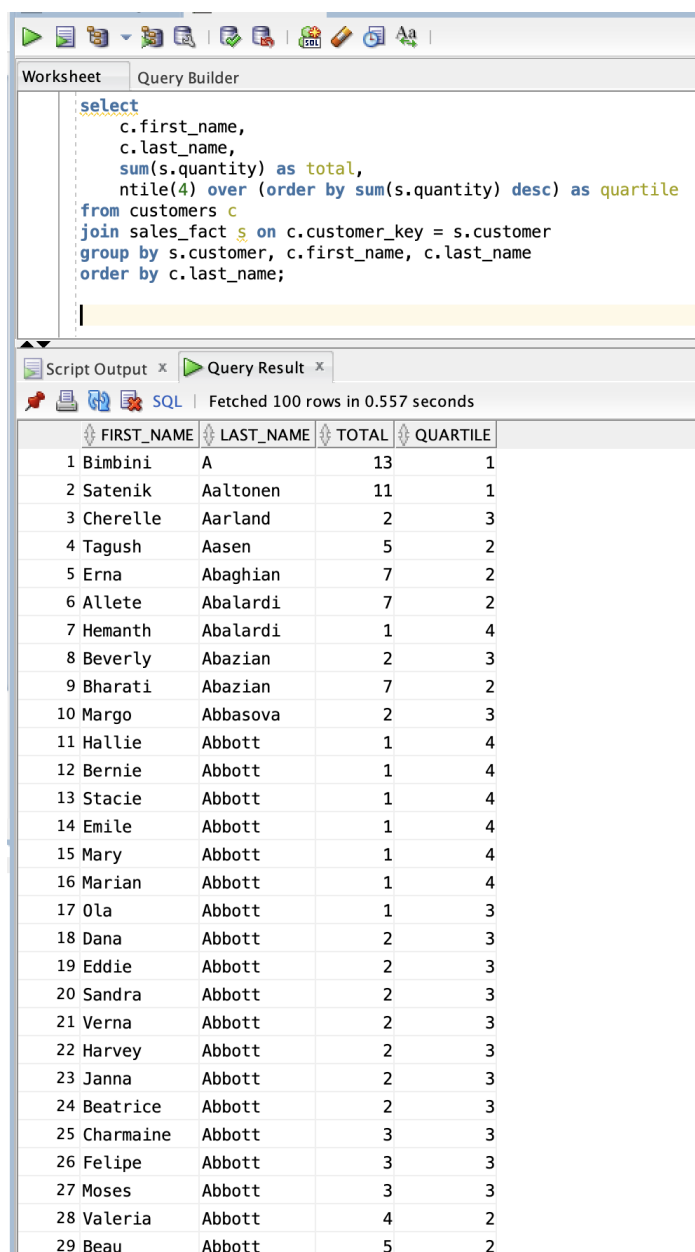


The screenshot shows a SQL IDE interface. The top pane displays the SQL query, which is identical to the one in the previous block. The bottom pane shows the query results in a table format. The table has four columns: CALENDAR_YEAR_NAME, YEAR_SALES, LAST_YEAR_SALES, and GROWTH_PCT. There are two rows of data, corresponding to the years CY2008 and CY2009. The GROWTH_PCT for CY2008 is 100, and for CY2009 it is 15,098.

	CALENDAR_YEAR_NAME	YEAR_SALES	LAST_YEAR_SALES	GROWTH_PCT
1	CY2008	14581730,71	0	100
2	CY2009	17174703,5	14581730,71	15,098

4. Agrupe los clientes en cuatro grupos (cuartiles) de acuerdo a las compras basadas en unidades que han hecho históricamente, considerando las unidades de mayor a menor. Su reporte debe verse más o menos así:

```
select
  c.first_name,
  c.last_name,
  sum(s.quantity) as total,
  ntile(4) over (order by sum(s.quantity) desc) as quartile
from customers c
join sales_fact s on c.customer_key = s.customer
group by s.customer, c.first_name, c.last_name
order by c.last_name;
```



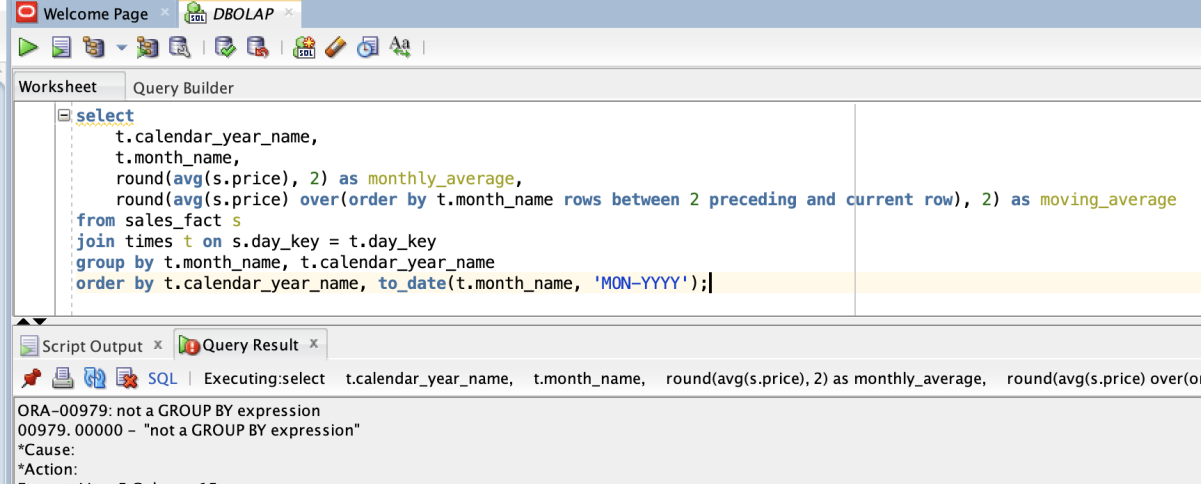
Script Output x Query Result x

SQL | Fetched 100 rows in 0.557 seconds

	FIRST_NAME	LAST_NAME	TOTAL	QUARTILE
1	Bimbini	A	13	1
2	Satenik	Aaltonen	11	1
3	Cherelle	Aarland	2	3
4	Tagush	Aasen	5	2
5	Erna	Abaghian	7	2
6	Allete	Abalardi	7	2
7	Hemanth	Abalardi	1	4
8	Beverly	Abazian	2	3
9	Bharati	Abazian	7	2
10	Margo	Abbasova	2	3
11	Hallie	Abbott	1	4
12	Bernie	Abbott	1	4
13	Stacie	Abbott	1	4
14	Emile	Abbott	1	4
15	Mary	Abbott	1	4
16	Marian	Abbott	1	4
17	Ola	Abbott	1	3
18	Dana	Abbott	2	3
19	Eddie	Abbott	2	3
20	Sandra	Abbott	2	3
21	Verna	Abbott	2	3
22	Harvey	Abbott	2	3
23	Janna	Abbott	2	3
24	Beatrice	Abbott	2	3
25	Charmaine	Abbott	3	3
26	Felipe	Abbott	3	3
27	Moses	Abbott	3	3
28	Valeria	Abbott	4	2
29	Beau	Abbott	5	2

5. Considerando el promedio de precio de cada mes, obtenga el promedio de este valor considerando para cada mes los 3 meses anteriores (el “promedio móvil” de los últimos 3 meses). Su reporte debe verse más o menos así:

-- En esta no logré completar el query... sé que va algo como el de abajo, pero por más que lo corro de diferentes maneras me dio error:



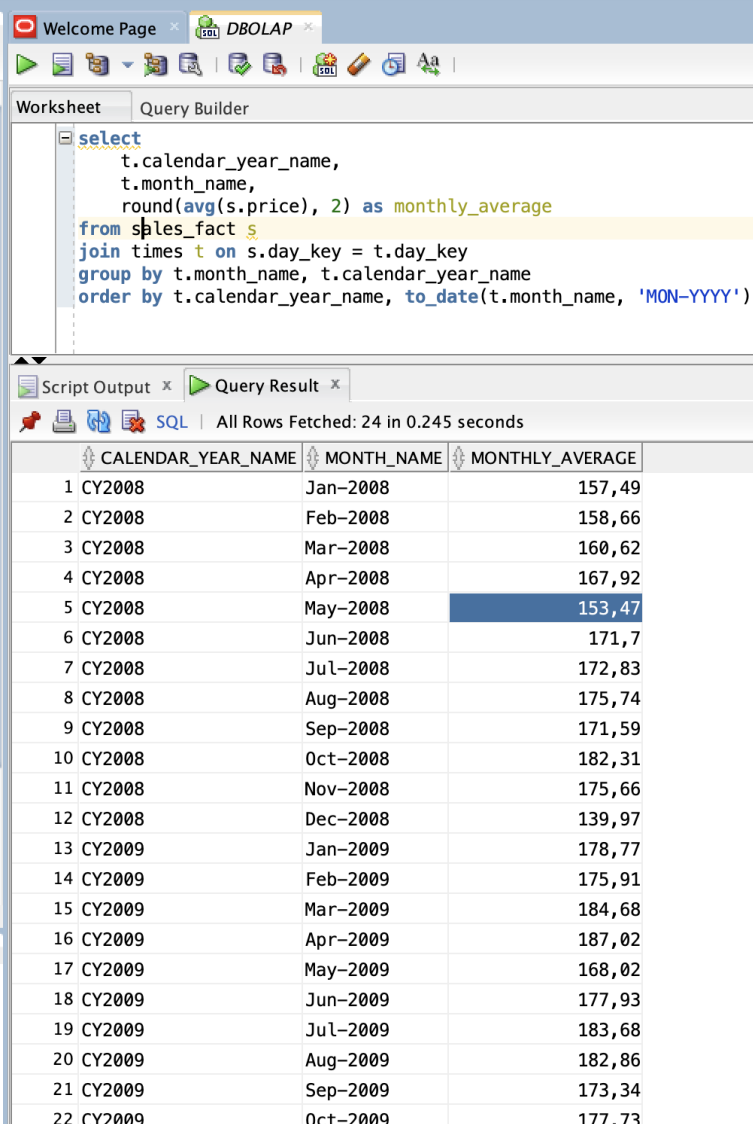
The screenshot shows the SQL Developer interface with a query in the Query Builder. The query is:

```
select
  t.calendar_year_name,
  t.month_name,
  round(avg(s.price), 2) as monthly_average,
  round(avg(s.price) over(order by t.month_name rows between 2 preceding and current row), 2) as moving_average
from sales_fact s
join times t on s.day_key = t.day_key
group by t.month_name, t.calendar_year_name
order by t.calendar_year_name, to_date(t.month_name, 'MON-YYYY');
```

The error message in the Script Output pane is:

```
ORA-00979: not a GROUP BY expression
00979. 00000 - "not a GROUP BY expression"
*Cause:
*Action:
Error at Line: 5 Column: 15
```

-- Removiendo lo que da error:



The screenshot shows the SQL Developer interface with the corrected query in the Query Builder. The query is:

```
select
  t.calendar_year_name,
  t.month_name,
  round(avg(s.price), 2) as monthly_average
from sales_fact s
join times t on s.day_key = t.day_key
group by t.month_name, t.calendar_year_name
order by t.calendar_year_name, to_date(t.month_name, 'MON-YYYY');
```

The Script Output pane shows the query executed successfully, fetching 24 rows in 0.245 seconds. The results are displayed in a table with the following columns: CALENDAR_YEAR_NAME, MONTH_NAME, and MONTHLY_AVERAGE.

	CALENDAR_YEAR_NAME	MONTH_NAME	MONTHLY_AVERAGE
1	CY2008	Jan-2008	157,49
2	CY2008	Feb-2008	158,66
3	CY2008	Mar-2008	160,62
4	CY2008	Apr-2008	167,92
5	CY2008	May-2008	153,47
6	CY2008	Jun-2008	171,7
7	CY2008	Jul-2008	172,83
8	CY2008	Aug-2008	175,74
9	CY2008	Sep-2008	171,59
10	CY2008	Oct-2008	182,31
11	CY2008	Nov-2008	175,66
12	CY2008	Dec-2008	139,97
13	CY2009	Jan-2009	178,77
14	CY2009	Feb-2009	175,91
15	CY2009	Mar-2009	184,68
16	CY2009	Apr-2009	187,02
17	CY2009	May-2009	168,02
18	CY2009	Jun-2009	177,93
19	CY2009	Jul-2009	183,68
20	CY2009	Aug-2009	182,86
21	CY2009	Sep-2009	173,34
22	CY2009	Oct-2009	177,73