# Sistemas MultiDimensionales

Práctica 1. PostgreSQL

**Ejercicio: Informes SQL** 



8 de marzo de 2025 **Prof. José Samos Jiménez** 

Autor:
Ana Aragón Jerónimo

# UNIVERSIDAD DE GRANADA

E.T.S. de Ingenierías Informática y de Telecomunicación

# Apartados:

1.	Informe inicial y operaciones:		2
	*	Informe inicial	2
	*	Roll-Up	2
	*	Drill-Down	2
	*	Slice&Dice	2
2.	Base de datos OLTP:		3
	*	Informe inicial	3
	*	Roll-Up	5
	*	Drill-Down	6
	*	Slice&Dice	8
3.	Base de datos OLAP:		10
	*	Informe inicial	10
	*	Roll-Up	12
	*	Drill-Down	13
	*	Slice&Dice	15

# 1. Informe inicial y operaciones.

En cada base de datos se van a realizar las mismas operaciones, que serán las descritas a continuación.

## ❖ Informe inicial focalizado en las ventas:

"Ventas por año y marca".

## Nivel de detalle:

CUÁNDO: Año. DÓNDE: Todo. QUÉ: Marca.

A partir del informe inicial, se obtienen 3 informes:

# Roll-Up (se obtiene menos nivel de detalle):

"Ventas por año".

## Nivel de detalle:

CUÁNDO: Año. DÓNDE: Todo. QUÉ: Todo.

# ❖ <u>Drill-down</u> (se obtiene más nivel de detalle):

"Ventas por año y por tamaño del producto dentro de cada marca".

## Nivel de detalle:

CUÁNDO: Año. DÓNDE: Todo.

QUÉ: Tamaño del producto según la marca.

## Slice&Dice (no cambia el nivel de detalle):

"Ventas por año y primeras 6 marcas en orden alfabético".

#### Nivel de detalle:

CUÁNDO: Año. DÓNDE: Todo. QUÉ: Marca.

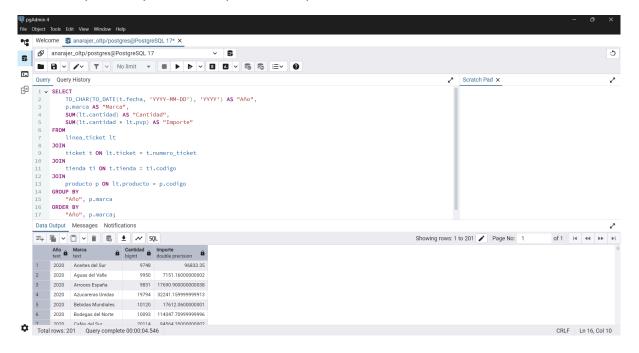
A continuación se realizarán dichos informes en cada base de datos (OLTP y OLAP), haciéndose en primer lugar el informe inicial para seguidamente pasar a las operaciones de Roll-Up, Drill-Down y Slice&Dice.

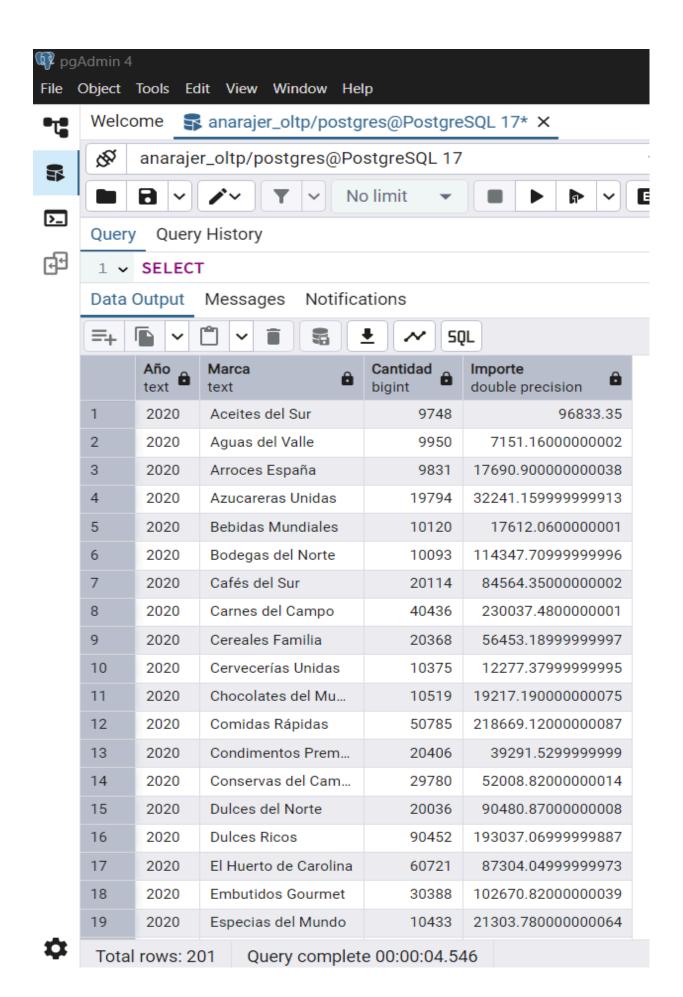
# 2. Base de datos OLTP

❖ Informe de partida: "Ventas por año y marca".

#### Sentencias SQL:

```
SELECT
  TO CHAR(TO DATE(t.fecha, 'YYYY-MM-DD'), 'YYYY') AS "Año",
  p.marca AS "Marca",
  SUM(It.cantidad) AS "Cantidad",
  SUM(It.cantidad * It.pvp) AS "Importe"
FROM
  linea ticket It
JOIN
  ticket t ON lt.ticket = t.numero_ticket
JOIN
  tienda ti ON t.tienda = ti.codigo
JOIN
  producto p ON It.producto = p.codigo
GROUP BY
  "Año", p.marca
ORDER BY
  "Año", p.marca;
```





❖ Roll-Up: "Ventas por año".

## Sentencias SQL:

```
SELECT

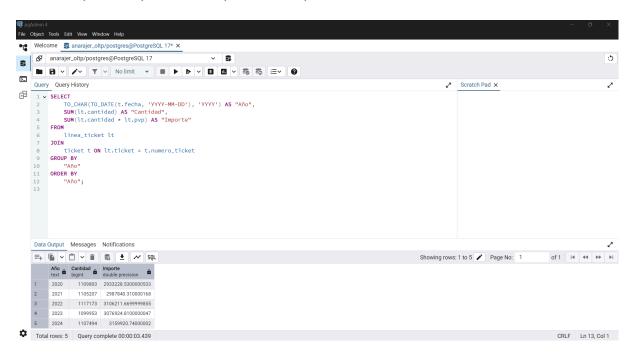
TO_CHAR(TO_DATE(t.fecha, 'YYYY-MM-DD'), 'YYYY') AS "Año",
SUM(lt.cantidad) AS "Cantidad",
SUM(lt.cantidad * lt.pvp) AS "Importe"

FROM
linea_ticket lt

JOIN
ticket t ON lt.ticket = t.numero_ticket

GROUP BY
"Año"

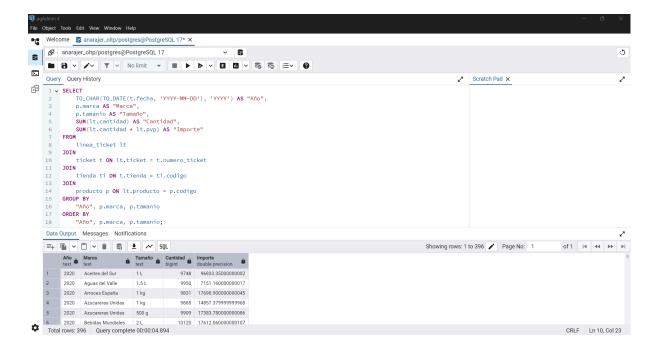
ORDER BY
"Año";
```

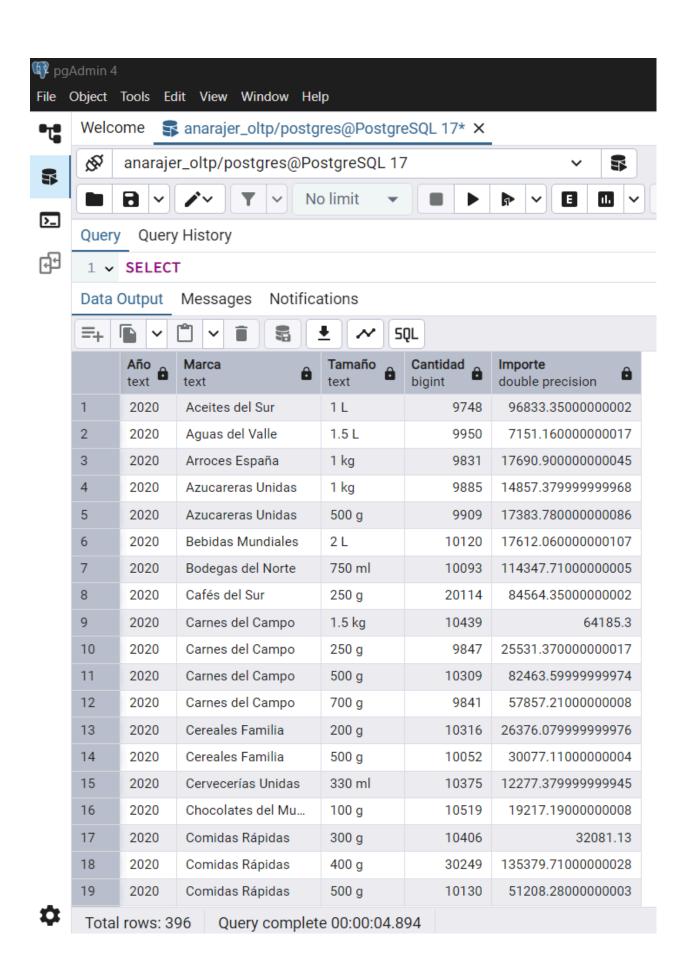


❖ <u>Drill-Down:</u> "Ventas por año y por tamaño del producto dentro de cada marca".

#### Sentencias SQL:

```
SELECT
  TO_CHAR(TO_DATE(t.fecha, 'YYYY-MM-DD'), 'YYYY') AS "Año",
  p.marca AS "Marca",
  p.tamanio AS "Tamaño",
  SUM(It.cantidad) AS "Cantidad",
  SUM(It.cantidad * It.pvp) AS "Importe"
FROM
  linea_ticket It
JOIN
  ticket t ON lt.ticket = t.numero ticket
JOIN
  tienda ti ON t.tienda = ti.codigo
  producto p ON It.producto = p.codigo
GROUP BY
  "Año", p.marca, p.tamanio
ORDER BY
  "Año", p.marca, p.tamanio;
```

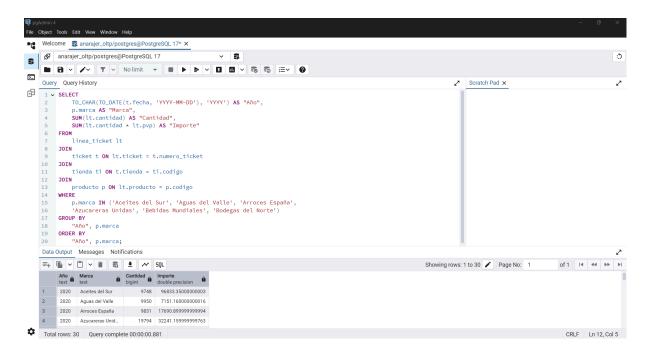


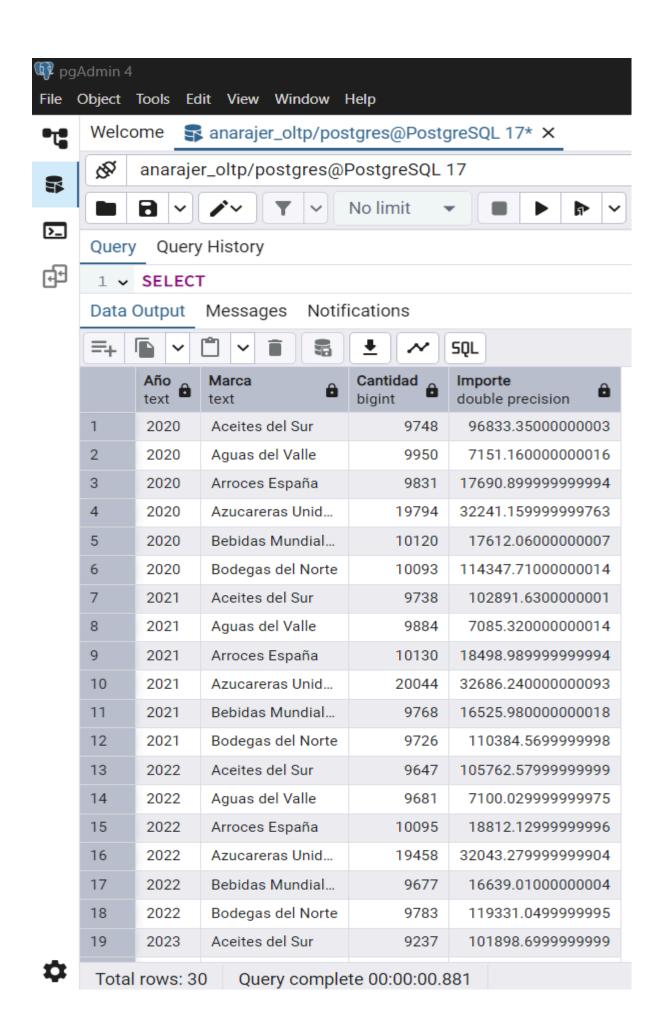


Slice&Dice: "Ventas por año y primeras 6 marcas en orden alfabético".

#### Sentencias SQL:

```
SELECT
  TO_CHAR(TO_DATE(t.fecha, 'YYYY-MM-DD'), 'YYYY') AS "Año",
  p.marca AS "Marca",
  SUM(It.cantidad) AS "Cantidad",
  SUM(It.cantidad * It.pvp) AS "Importe"
FROM
  linea ticket It
JOIN
  ticket t ON lt.ticket = t.numero_ticket
  tienda ti ON t.tienda = ti.codigo
JOIN
  producto p ON lt.producto = p.codigo
WHERE
  p.marca IN ('Aceites del Sur', 'Aguas del Valle', 'Arroces España',
  'Azucareras Unidas', 'Bebidas Mundiales', 'Bodegas del Norte')
GROUP BY
  "Año", p.marca
ORDER BY
  "Año", p.marca;
```





# 3. Base de datos OLAP

❖ Informe de partida: "Ventas por año y marca".

#### Sentencias SQL:

```
SELECT

dc.anio AS "Año",

dq.marca AS "Marca",

SUM(fv.cantidad) AS "Cantidad",

SUM(fv.pvp * fv.cantidad) AS "Importe"

FROM

fact_ventas_dia fv

JOIN

dim_cuando dc ON fv.dim_cuando_key = dc.dim_cuando_key

JOIN

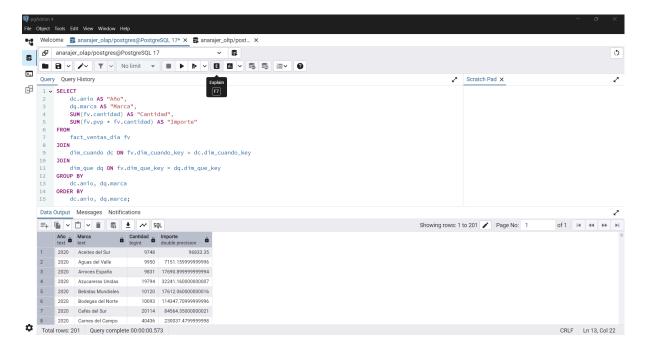
dim_que dq ON fv.dim_que_key = dq.dim_que_key

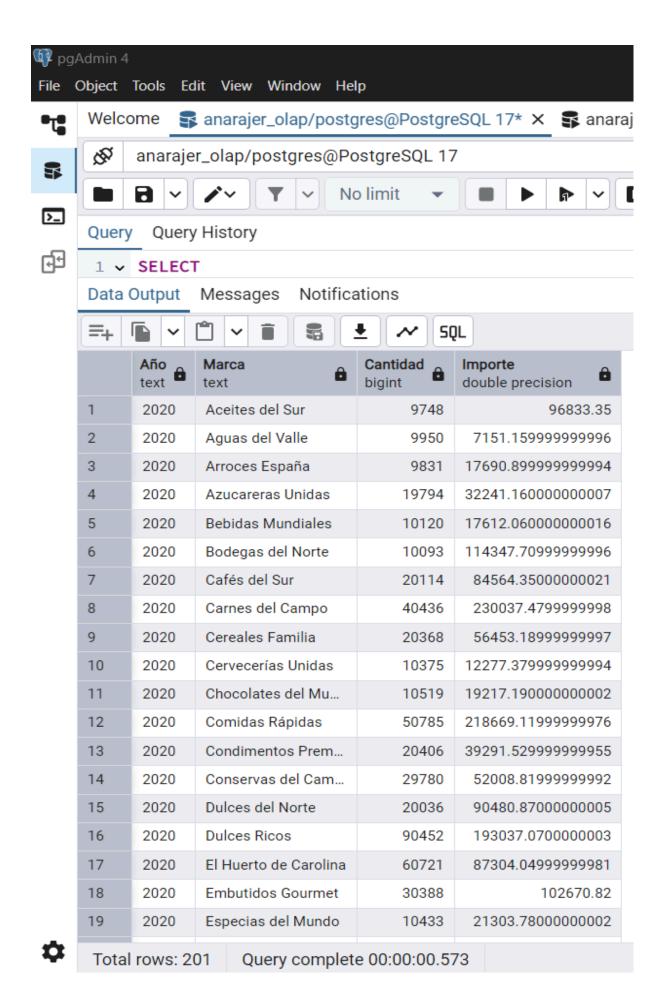
GROUP BY

dc.anio, dq.marca

ORDER BY

dc.anio, dq.marca;
```





❖ Roll-Up: "Ventas por año".

## Sentencias SQL:

```
SELECT

dc.anio AS "Año",

SUM(fv.cantidad) AS "Cantidad",

SUM(fv.pvp * fv.cantidad) AS "Importe"

FROM

fact_ventas_dia fv

JOIN

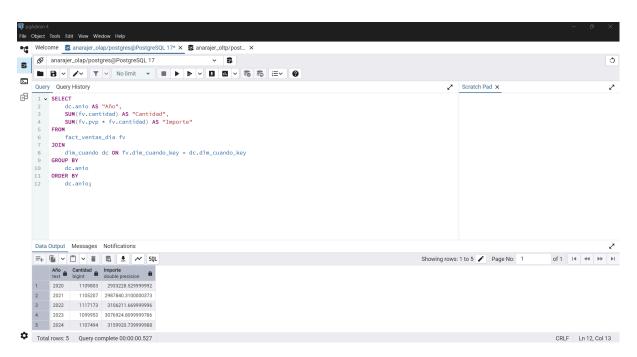
dim_cuando dc ON fv.dim_cuando_key = dc.dim_cuando_key

GROUP BY

dc.anio

ORDER BY

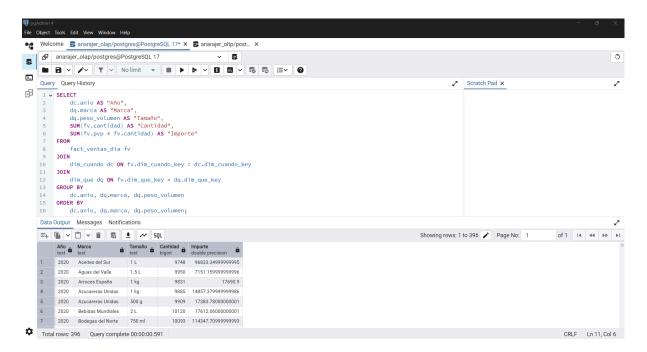
dc.anio;
```

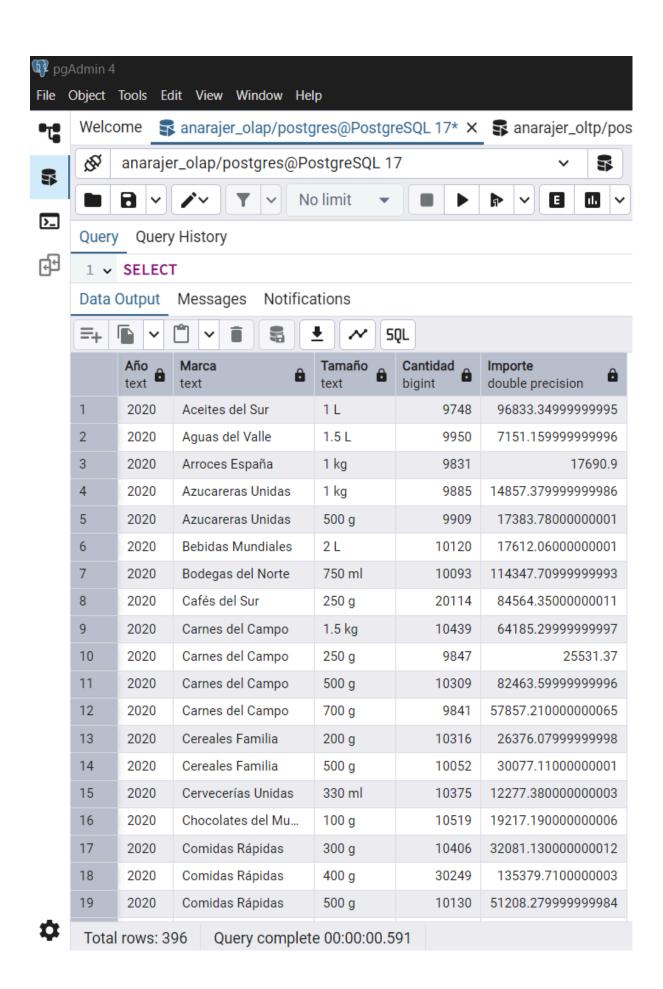


❖ <u>Drill-Down:</u> "Ventas por año y por tamaño del producto dentro de cada marca".

# Sentencias SQL:

```
SELECT
  dc.anio AS "Año",
  dq.marca AS "Marca",
  dq.peso_volumen AS "Tamaño",
  SUM(fv.cantidad) AS "Cantidad",
  SUM(fv.pvp * fv.cantidad) AS "Importe"
FROM
  fact_ventas_dia fv
JOIN
  dim cuando dc ON fv.dim cuando key = dc.dim cuando key
JOIN
  dim_que dq ON fv.dim_que_key = dq.dim_que_key
GROUP BY
  dc.anio, dq.marca, dq.peso_volumen
ORDER BY
  dc.anio, dq.marca, dq.peso_volumen;
```





❖ Slice&Dice: "Ventas por año y primeras 6 marcas en orden alfabético".

# Sentencias SQL:

```
SELECT
  dc.anio AS "Año",
  dq.marca AS "Marca",
  SUM(fv.cantidad) AS "Cantidad",
  SUM(fv.pvp * fv.cantidad) AS "Importe"
FROM
  fact ventas dia fv
JOIN
  dim_cuando dc ON fv.dim_cuando_key = dc.dim_cuando key
JOIN
  dim_que dq ON fv.dim_que_key = dq.dim_que_key
WHERE
  dq.marca IN ('Aceites del Sur', 'Aguas del Valle', 'Arroces España',
  'Azucareras Unidas', 'Bebidas Mundiales', 'Bodegas del Norte')
GROUP BY
  dc.anio, dq.marca
ORDER BY
  dc.anio, dq.marca;
```

