**-- Día 5**

**--Ejemplo**

WITH my\_subquery as(

SELECT sector, SUM(revenue) as total\_revenue

FROM fortune

GROUP BY 1)

SELECT AVG(total\_revenue) as avg\_sector\_revenue

from my\_subquery;

**--Ex.1: Los directores de películas de James Bond han trabajado en promedio en dos películas cada uno. ¿Cómo obtendrías esta información?**

with tab as(

SELECT count(DISTINCT director) as directores, count(film) as pelis

from jamesbond\_new)

SELECT pelis/directores as media\_pelis\_x\_director

from tab

--profe

with tab as(

SELECT director, count(film) as pelis

from jamesbond\_new

group by 1)

SELECT avg(pelis) as media\_pelis\_x\_director

from tab

**--Ex.2: La tabla gobierno\_paro reporta data mensual de paro para municipios de España en 2018. Calcular la media de paro de cada comunidad autónoma en 2018.**

with sub as(

SELECT comunidad\_autonoma, sum(total\_paro\_registrado) as suma\_paro

FROM gobierno\_paro\_dos

GROUP by comunidad\_autonoma)

SELECT comunidad\_autonoma, avg(suma\_paro) as avg\_paro -- realmente esta parte no aporta nada, pq habia 2 registros y se mantiene, no hay AVG como tal (1/1)

from sub

GROUP by comunidad\_autonoma

**--Ej.3: Extraer consolas no discontinuadas y su fecha de lanzamiento (first retail availability). Agregar ventas globales de videojuegos publicados desde el año 2000. Ordenar según ventas totales de videojuegos (descendiente).**

SELECT c.console\_name, c.first\_retail\_availability, round(sum(g.na\_sales + g.eu\_sales + g.jp\_sales + g.other\_sales), 0) AS total\_games\_sales

FROM consoles c

LEFT JOIN games g ON c.platform\_code = g.platform\_code --tiene que ser la columna que coincida en ambas tablas (en contenido, aunque tengan diferente nombre, ej 'id' en una y 'idcliente' en otra)

WHERE c.discontinued = '' AND g.year >= 2000

GROUP BY c.console\_name, c.first\_retail\_availability --o 1,2 -> es para que salga más de un solo valor (si hay sum, count etc. pasaría eso)

ORDER BY total\_games\_sales DESC;

--poniendo la tabla entera

SELECT consoles.console\_name,

consoles.first\_retail\_availability,

round(sum(games.na\_sales + games.eu\_sales + games.jp\_sales + games.other\_sales), 0) AS total\_games\_sales

FROM consoles

LEFT JOIN games ON consoles.platform\_code = games.platform\_code

WHERE consoles.discontinued = '' AND games.year >= 2000

GROUP BY consoles.console\_name, consoles.first\_retail\_availability

ORDER BY total\_games\_sales DESC;

**--Punto a) Una empresa de venta de equipos informáticos necesita conocer a través de un informe, el total de las ventas diarias por cliente que está generando, ordenadas por fecha. Del cliente debemos mostrar (nombres, DNI, correo).**

SELECT ventas.fecha, clientes.nombre, clientes.dni, clientes.correo, SUM(ventas.total) AS total\_ventas\_diarias

FROM ventas

INNER JOIN clientes ON ventas.clienteid = clientes.id

GROUP BY clientes.nombre

ORDER BY total\_ventas\_diarias desc;

**--Punto b) En otro reporte la empresa necesita saber el detalle de cada una de las ventas (cuantos y que productos fueron registrados en cada venta)**

SELECT VENTAS\_DETALLE.id, clientes.nombre, clientes.correo, productos.nombre, VENTAS\_DETALLE.cantidad --fecha y DNI no aportan nada

FROM ventas

INNER JOIN clientes ON ventas.clienteid = clientes.id

INNER JOIN VENTAS\_DETALLE ON ventas.id = VENTAS\_DETALLE.venta\_id

INNER JOIN productos ON VENTAS\_DETALLE.productoid = productos.id

ORDER BY VENTAS\_DETALLE.id;

--profe:

SELECT VEN.ID AS FACTURA, PR.CODIGO AS CODIGO\_PRODUCTO, PR.NOMBRE AS NOMBRE\_PRODUCTO, DT.CANTIDAD AS CANTIDAD\_PRODUCTO

FROM VENTAS VEN

INNER JOIN VENTAS\_DETALLE DT ON VEN.ID=DT.VENTA\_ID

INNER JOIN PRODUCTOS PR ON DT.ID=PR.ID