Variables sintéticas

Columnas que nunca estuvieron en el archivo Nombres _____

Superset, base de datos `Dades classe`, esquema `public`. Partís de la tabla `games` y el script `games_clean.sql` (pedídselo al profesor). Parejas: uno clica, otro lee; cambio en la Tarea 3. Resultados aquí.

Referencia · el tique

En el tique	Qué es	En SQL / Superset
Total de línea (unidades × precio)	Un valor nuevo **para cada fila**, hecho solo a partir de esa fila.	**Columna calculada**: `owners_min * price`
TOTAL de abajo	Un valor **para todo el grupo**; las filas desaparecen.	**Métrica**: `SUM(owners_min * price)`
«Esta línea es el 40% del tique»	Un valor de grupo escrito **junto a cada fila**; las filas se quedan.	**Función de ventana**: `SUM(...) OVER ()`

Una **variable sintética** es una columna que nadie recogió: se calcula a partir de las que sí. Leed `OVER ()` como «sobre todas las filas, sin colapsarlas»; `PARTITION BY` es el `GROUP BY` de una ventana.

Parte A · Montad `games_clean` (8 min)

Paso	Haced esto	Deberíais ver / escribid
A1	**SQL → SQL Lab**, esquema `public`: pegad todo el `games_clean.sql` y ejecutadlo. Después ejecutad `SELECT COUNT(*) FROM games_clean;`	Filas: _____ `games` tiene 26 688. ¿Qué línea del script ha eliminado el resto? _____
A2	Leed la última sentencia del script (`ALTER TABLE ...`). Añade una columna que nunca estuvo en el archivo.	Su nombre: _____ Calculada a partir de: _____
A3	**Datasets → + Dataset** → Database `Dades classe`, Schema `public`, Table `games_clean` → **Create dataset** (la flecha junto a **Create and explore dataset**).	`games_clean` está en vuestra lista de Datasets. No empecéis la Parte B sin eso.

Parte B · Tarea 1 · Una columna calculada (6 min)

Paso	Haced esto	Deberíais ver / escribid
1.1	**Datasets** → fila `games_clean`, lápiz de **Actions** (**Edit**) → pestaña **Calculated columns** → **Add item**. Nombre: `est_revenue`. **SQL expression**: en lugar del texto de ejemplo, `owners_min * price`. **Data type**: `NUMERIC`. **Save** → **OK**.	¿Qué línea del tique es? _____

1.2 Comprobad la misma expresión en SQL Lab:

```
SELECT game, owners_min, price, owners_min * price AS est_revenue
FROM games_clean ORDER BY est_revenue DESC LIMIT 5;
```

Primer juego: _____ `est_revenue`: _____ ¿Millones o miles de millones? _____

¿Atascados?

- **`This database does not allow for DDL/DML...`**: avisad al profesor. Es una casilla en los ajustes de la conexión.
- **El script se ha ejecutado a medias**: volved a ejecutarlo entero. Siempre reconstruye la tabla desde cero.
- **`This table already has a dataset`**: ya está registrada. Mirad en la lista de Datasets.
- ¿Seguís atascados tras dos intentos? Preguntad a los vecinos y después al profesor.

Tarea 2 · Dos métricas (7 min)

Paso	Haced esto	Deberíais ver / escribid

2.1	Mismo editor, pestaña **Metrics** → **Add item**, dos veces.
	`total_est_revenue` = `SUM(owners_min * price)`
	`pct_free` = `100.0 * SUM(CASE WHEN is_free THEN 1 ELSE 0 END) / COUNT(*)`
	Save → **OK**.
		Haced una predicción antes del paso 2.2: ¿qué `owners_tier` (Hit, Mid, Niche) tiene la proporción más alta de juegos gratuitos? _____

2.2	En SQL Lab, calculad las dos expresiones por nivel: escribid la consulta vosotros mismos, con `GROUP BY owners_tier`.	Hit: _____ · _____ % Mid: _____ · _____ % Niche: _____ · _____ %
2.3	Ejecutad `SELECT SUM(est_revenue) FROM games_clean;` Falla **a propósito**.	El error dice: _____ Creasteis `est_revenue` en la Tarea 1. ¿Dónde vive? _____

Tarea 3 · Funciones de ventana (11 min)

3.1 Ejecutadla tal como está escrita: la proporción de juegos de cada franja de precio.

```sql
SELECT price_band, COUNT(*) AS games,
       ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (), 1) AS pct_of_games
FROM games_clean GROUP BY price_band ORDER BY games DESC;
```

Franja más grande: _____ con _____ % En el tique, ¿qué es `SUM(COUNT(*)) OVER ()`? _____

3.2 Cada juego frente al precio medio de su año.

```sql
SELECT game, release_year, price,
       ROUND(AVG(price) OVER (PARTITION BY release_year), 2)         AS year_avg,
       ROUND(price - AVG(price) OVER (PARTITION BY release_year), 2) AS vs_year_avg
FROM games_clean ORDER BY owners_min DESC, number LIMIT 5;
```

¿Cuál de los cinco cuesta más que la media de su año? _____ ¿Cuánto más? _____

3.3 Adaptad la consulta de 3.1: en lugar de la proporción de **juegos** por `price_band`, la proporción de **ingresos estimados** por `owners_tier`.

El nivel Hit es el _____ % de los juegos y el _____ % de los ingresos estimados.

3.4 Mirad la fila Niche de 3.3. ¿De verdad ganaron eso esos juegos? Dad un motivo por el que `est_revenue` **no** es lo que un juego ganó realmente.

_____

_____

Habéis acabado cuando Todos los espacios en blanco están rellenados · `games_clean` está en vuestra lista de Datasets. **¿Pareja rápida?** Haced clic en el nombre del dataset `games_clean` para abrirlo en Explore. Tipo de gráfico **Table**; **Dimensions**: `owners_tier`; **Metrics**: elegid `total_est_revenue` en la pestaña **Saved**; **Create chart**. No lo guardéis. Después probad las mismas tres variables en vuestro conjunto de datos abierto.

En casa Acabad esta hoja. Una ventana más: con `RANK() OVER (PARTITION BY release_year ORDER BY owners_min DESC)`, encontrad los tres juegos con más propietarios de 2017.