Guardar la pregunta o la respuesta

Datasets virtuales, `CREATE TABLE ... AS` y una API

Nombres _____

Superset, base de datos `Dades classe`, esquema `public`, tablas `pokemon` y `generations`. La analogía de hoy es una cocina: las tablas son los ingredientes. Por parejas: uno teclea, el otro escribe aquí; cambiad en A2.

Referencia

	Receta	Táper
Guardáis	La pregunta (el SQL).	La respuesta (las filas).
En Superset	Un **dataset virtual**: una consulta guardada.	Una tabla nueva: `CREATE TABLE ... AS`.
Vive en	Superset: solo el texto de la consulta.	PostgreSQL: una tabla real, junto a `pokemon`.
Al usarlo	Se cocina de nuevo, cada vez.	Se sirve tal como se guardó.

Un **dataset** es aquello a partir de lo que Superset puede construir gráficos. Un **dataset físico** apunta a una tabla real: una etiqueta en un tarro, no una copia.

Parte A · Guardar la pregunta, guardar la respuesta (20 min)

A1 · La receta. SQL → SQL Lab. Escribid y **Run** (es el JOIN del miércoles):

```sql
SELECT p.id, p.pokemon, p.type_1, p.type_2, p.hp, p.attack, p.defense, p.speed,
       g.generation_id, g.region, g.release_year
FROM pokemon p
JOIN generations g ON p.generation_id = g.generation_id
```

Filas _____ `pokemon` tiene 949. ¿Por qué no 949? _____

Flecha junto a **Save** → **Save dataset** → **Save as new**, nombre `pokemon_with_region` → **Save & Explore**. El creador de gráficos se abre en una pestaña nueva del navegador: cerradla y abrid **Datasets**.

Tipo de `pokemon_with_region` _____ Tipo de `pokemon` y `generations` _____

A2 · El táper. Pestaña nueva de SQL Lab. Primero ejecutad solo el `SELECT` (desde la línea 2). Luego añadid la línea 1 y ejecutadlo todo.

```sql
CREATE TABLE pokemon_by_generation AS
SELECT g.generation_id, g.region, g.release_year,
       COUNT(p.id) AS n_pokemon,
       ROUND(AVG(p.attack)) AS avg_attack,
       MAX(p.speed) AS max_speed
FROM generations g
LEFT JOIN pokemon p ON p.generation_id = g.generation_id
GROUP BY g.generation_id, g.region, g.release_year;
```

Solo el `SELECT`: filas _____ Con la línea 1: ¿qué muestra SQL Lab? _____

¿Adónde han ido las filas? _____

Ahora ejecutad `SELECT * FROM pokemon_by_generation ORDER BY generation_id;` Filas _____

Más Pokémon: _____ con _____ `n_pokemon = 0`: _____ Su `avg_attack` dice _____

¿Por qué esa celda no es 0? _____

A3 · Etiquetad el tarro. Superset todavía no tiene ningún dataset para la tabla nueva. **Datasets → + Dataset**: Database `Dades classe`, Schema `public`, Table `pokemon_by_generation`. Flecha junto a **Create and explore dataset** → **Create dataset**.

Tipo de `pokemon_by_generation` en **Datasets** _____

¿Cuál se queda desfasado? (con el profesor, 5 min)

El profesor añade una novena región (Paldea, 2022) a `generations` en el proyector. No lo ejecutéis vosotros: vuestras tablas deben quedarse como están. Escribid vuestra predicción **antes** de que se ejecute.

¿Cuántas regiones mostrará?	Predicción	Ha mostrado
La **tabla** `pokemon_by_generation` (el táper)		
El `SELECT` de A2 ejecutado de nuevo como **consulta** (la receta)		

¿Por qué? _____

Parte B · De una API a un dataset (16 min)

Una **API** es una dirección a la que preguntáis y que responde con datos, no con una página web. Responde en **JSON**: texto de nombres y valores, donde un valor puede contener más valores. Superset no puede preguntar a una API: **extraéis** y luego **cargáis**.

B1 Abrid `https://pokeapi.co/api/v2/pokemon?limit=20` en el navegador. Sin clave, sin inicio de sesión.

`count` dice _____ Pokémon mostrados _____ ¿Qué campo contiene la dirección del resto? _____

B2 Abrid `https://pokeapi.co/api/v2/pokemon/charizard`. En SQL Lab ejecutad `SELECT * FROM pokemon WHERE pokemon = 'charizard'`.

Altura: API _____ tabla _____ Peso: API _____ tabla _____ ¿Por qué? _____

¿Cómo guarda cada una los dos tipos? _____

B3 **Extraer, aplanar, cargar. Aplanar** es convertir un JSON anidado en filas y columnas. Descargad `api_to_csv.py` de la página Recursos del sitio web del curso. En una terminal, en esa carpeta: `python3 api_to_csv.py`. Abrid el `pokeapi_sample.csv` que escribe. Luego, en Superset: **+ → Data → Upload CSV to database**, base de datos `Dades classe`, esquema `public`, nombre de tabla `pokeapi_sample` → **Upload**. En SQL Lab: `SELECT type_1, COUNT(*) AS n FROM pokeapi_sample GROUP BY type_1 ORDER BY n DESC`

CSV: filas _____ columnas _____ Los dos primeros tipos: _____ Tipo en **Datasets** _____

Mañana sale un Pokémon nuevo. ¿Cómo lo hacéis llegar a esta tabla? _____

B4 **La lista de comprobación de una API.** Seis preguntas antes de construir nada sobre ella. Anotad lo visto.

Pregunta	Qué averiguar	PokéAPI
Dirección	La URL y sus **parámetros** después de la `?`.	
Clave	¿Hace falta una cuenta o una **clave de API**? Una clave es una contraseña: no la peguéis nunca en ningún sitio.	
Límites	¿Cuántas peticiones podéis hacer?	
Paginación	¿Cabe todo en una respuesta? ¿Qué campo es el siguiente?	
Licencia	¿Podéis reutilizar y publicar los datos? ¿A quién citar?	
Instantánea	La tabla es una copia de un momento. Anotad la fecha.	

Habéis acabado cuando Todos los huecos están rellenos · **Datasets** lista un **Virtual** y al menos tres **Physical**.
¿Pareja rápida? Repetid A1 y A2 con vuestro conjunto de datos abiertos.

¿Atascados?

- **`relation "pokemon_by_generation" already exists`**: la habéis ejecutado dos veces. Buena noticia: la tabla está ahí.
- **`does not allow for DDL/DML`**: no es culpa vuestra, avisad al profesor. **`Table already exists` al subir el CSV**: poned **If table already exists** en **Replace**. **`pokeapi.co` no carga**: pedid al profesor el `pokeapi_sample.csv` ya hecho y subidlo.