The rows that vanish

Two tables, one answer: `pokemon` + `generations` Names _____

Superset → **SQL → SQL Lab**, database `Dades classe`, schema `public`. Two shoeboxes today: `pokemon` (949 cards) and a second, tiny one you upload yourself, `generations` (region cards). Pairs, one keyboard; swap who types when Part A is done. Write results here.

Reference

You write	With the two boxes
`FROM pokemon AS p`	`p` is an **alias**: a short name for the box. `p.pokemon` is the `pokemon` line of a card from box `p`.
`ON p.x = g.x`	The rule for clipping: which line must be equal on both cards. That shared column is the **key**.
`INNER JOIN`	Keep only the **clipped pairs**. A card with no partner is left on the table.
`LEFT JOIN`	Keep **every card of the left box** (the table written first, after `FROM`). No partner? It gets a blank region card: `NULL` in every `g.` column.

Part A · A second box, clipped (14 min)

A1 Download `generations.csv` from the Resources page of the course site and upload it: **+ (top right) → Data → Upload CSV to database**. **CSV file** `generations.csv` · **Database** `Dades classe` · **Schema** `public` (never empty) · **Table name** `generations` · **Delimiter** `,` → **Upload**. Then run `SELECT * FROM generations;`

Rows _____ Columns _____ Which column is printed on the cards of **both** boxes? _____

A2. Clip the cards. Run this and read the first row.

```
SELECT p.pokemon, p.type_1, g.region, g.release_year
FROM pokemon AS p
INNER JOIN generations AS g ON p.generation_id = g.generation_id
ORDER BY p.id;
```

First row: _____

`pokemon` has 949 rows. **Before you run anything else**, predict: how many rows does this join have? _____
Now count them: replace the first line with `SELECT COUNT(*)` and delete the `ORDER BY` line.

Rows _____ Was there an error or a warning? _____

A3 In your count, change one word: `INNER` → `LEFT`.

Rows _____ Difference with A2 _____ Where did you see that number last week? _____

A4. This one fails **on purpose**: the error is the lesson.

```
SELECT pokemon, region FROM pokemon
INNER JOIN generations ON generation_id = generation_id;
```

The error says: _____

Why can the database not answer? _____

The `ON` line that works: _____

Stuck?

- **`generations` has 16 rows**: the CSV was uploaded twice. Upload again with **If table already exists → Replace**.
- **The join gives 7592 rows**: your `ON` is wrong or missing: every Pokémon got clipped to every region (949 × 8).
- **`is ambiguous`** outside A4: a column that is in both boxes needs its prefix, `p.` or `g.`.
- Run one query at a time. Still stuck after two tries? Ask your neighbours, then the teacher.

Part B · The rows that vanish (20 min)

One query per question, always with the aliases `p` and `g` and `ON p.generation_id = g.generation_id`. For each join, write down how many rows come out.

B1 Region and first game of `pikachu`, `lucario` and `greninja`. One query.

`pikachu` _____ `lucario` _____ `greninja` _____

B2 How many Pokémon per region? `INNER JOIN` + `GROUP BY g.region`.

Rows _____ Most: _____ with _____ Fewest: _____ with _____

B3 The same with `LEFT JOIN`.

Rows _____ What is the new row? _____

B4 Search `p.pokemon LIKE 'charizard%'` with each join.

`INNER` _____ rows `LEFT` _____ rows Who is missing? _____

B5 List the Pokémon that the `INNER JOIN` drops: `LEFT JOIN` plus `WHERE g.region IS NULL`. Read twenty names.

How many? _____ What kind of Pokémon are they? _____

Why do they vanish? _____

B6 Turn the join around: `FROM generations AS g LEFT JOIN pokemon AS p`, group by region, use `COUNT(p.id)`.

Rows _____ Region that shows up for the first time: _____ with _____ Pokémon

Why did it never appear before? _____

B7 The five Pokémon with the highest attack, with their region: once with `INNER`, once with `LEFT`.

Top with `INNER` _____ Top with `LEFT` _____

Which top 5 would you publish, and why? _____

_____

B8 Average attack per region, strongest region first (rounded, `INNER JOIN`).

Top three: _____

Three counts, one table

Fill it in at the end of the class, with the whole group.

Join	Rows	Who is missing
`pokemon INNER JOIN generations`		
`pokemon LEFT JOIN generations`		
`generations LEFT JOIN pokemon`		

Done when Every blank is filled · B5, B6 and B7 have their sentence. **Fast pair?** On your own open dataset: which column is a code that needs a second table to be understood (municipality code, island, category number)? Name the key and the three columns that lookup table should have.

At home A shop has `customers(customer_id, name)` and `orders(order_id, customer_id)`. Write the query that lists the customers who have never ordered. Which join, which table on the left, which filter?