A real date

Text in disguise: `release_date` goes through the furnace

Names _____

Superset → **SQL → SQL Lab**, database `Dades classe`, schema `public`, table `games`. The furnace turns ore (text) into ingots (typed values). Pairs, one types and one writes; swap at Task 4. Run **one statement at a time**: select it, then **Run**. Ask the teacher for the long SQL of Tasks 3 and 4 as text and paste it: do not retype it.

Reference

Tool	What it does
`CAST(x AS type)` `x::type`	The furnace: turn `x` into that type, or stop with an error.
`TRIM(x)`	Cuts the spaces at both ends.
`REPLACE(x, a, b)`	Swaps every `a` for `b`.
`TO_DATE(x, pattern)`	Reads a date written the way the pattern says: `Mon` three-letter month, `DD` day, `MM` month number, `YYYY` year.

Pattern piece	Means
`x ~ 'pattern'`	Does the text `x` match?
`^` … `$`	From the very start to the very end.
`[A-Z][a-z]{2}`	One capital + two small letters: `Nov`.
`[0-9]{1,2}`	One or two digits (the day).
`[0-9]{4}`	Exactly four digits (the year).

Task 1 · Warm up the furnace

Write your prediction first. Then run the line.

Line	You predict	It returns
`SELECT ' Valve ' = 'Valve', TRIM(' Valve ') = 'Valve';`		
`SELECT REPLACE('1,234.50', ',', '')::numeric;`		
`SELECT REPLACE(REPLACE('1.234,50', '.', ''), ',', '.')::numeric;`		

`1,234.50` (English) and `1.234,50` (Spanish) are the same number. Why does the Spanish one need two `REPLACE`? _____

Task 2 · Try the lazy way

Both lines fail **on purpose**: the error is the lesson.

```
SELECT release_date::date FROM games;
SELECT TO_DATE(release_date, 'Mon DD, YYYY') FROM games;
```

First error: _____

Second error: _____

What kind of value breaks both lines? _____

Task 3 · Count the shapes

`CASE` is a fork in the road: each row takes the first road whose condition is true.

```
SELECT CASE WHEN release_date ~ '^[A-Z][a-z]{2} [0-9]{1,2}, [0-9]{4}$' THEN 'full'
            WHEN release_date ~ '^[A-Z][a-z]{2} [0-9]{4}$'             THEN 'month only'
            ELSE 'other' END AS shape, COUNT(*) AS n
FROM games GROUP BY shape ORDER BY n DESC;
```

`full` _____ `month only` _____ `other` _____ Look at the rows in `other` (a `WHERE` with the same patterns, or last week's `NOT LIKE`). Their value: _____ What is odd about it? _____

Task 4 · Build the converter

The same fork, but each road now smelts its own shape. No `ELSE`: whatever fits neither road becomes `NULL`. `WITH step1 AS (...)` gives a query a **name**, so the `SELECT` below can use its result like a table. `numeric(8,2)` is an exact decimal with two decimals: the right type for money.

```
WITH step1 AS (
  SELECT number, TRIM(game) AS game,
    CASE WHEN release_date ~ '^[A-Z][a-z]{2} [0-9]{1,2}, [0-9]{4}$'
           THEN TO_DATE(release_date, 'Mon DD, YYYY')
         WHEN release_date ~ '^[A-Z][a-z]{2} [0-9]{4}$'
           THEN TO_DATE(release_date, 'Mon YYYY')
    END AS release_date,
    price::numeric(8,2) AS price,
    owners, developer, publisher, average_playtime, median_playtime, metascore
  FROM games
  WHERE game IS NOT NULL
)
SELECT COUNT(*), MIN(release_date), MAX(release_date), COUNT(*) - COUNT(release_date) AS no_date
FROM step1;
```

Rows _____ Oldest _____ Newest _____ `no_date` _____

Which value ended up without a date? Accident or decision? _____

Run `SELECT TO_DATE('Jun 2009', 'Mon YYYY');` It returns _____. Who chose that day? _____

Task 5 · Use it and save it

Keep the `WITH` block and change only the final `SELECT`: games released per year, impossible last week.

```
SELECT EXTRACT(YEAR FROM release_date)::int AS year, COUNT(*) AS n
FROM step1 GROUP BY year ORDER BY year;
```

Busiest year _____ with _____ games. Then **Save** (next to **Run**) → name `games_step1`. It appears under **SQL → Saved Queries**.

Task 6 · The silent error

Today is 4 November, written the Spanish way. Predict what each line returns. Then run them.

```
SELECT '04/11/2026'::date;
SELECT TO_DATE('04/11/2026', 'DD/MM/YYYY');
```

First line _____ Second line _____ Any error message? _____

Which is worse, the error of Task 2 or this result? Why? _____

Done when Every blank is filled · `games_step1` is in your saved queries (Monday starts from it). **Fast pair?** Convert one text column of your own open dataset into a number or a date.

Stuck?

- **`Error parsing near ')'`** after the `WITH` block: you left a `;` before the final `SELECT`. **`Error parsing near 'SELECT'`**: the closing `)` is missing.
- **Task 2 gives no error**: your row limit is under 1 000. Raise it. **Save is greyed out**: keep the SQL in a text file.
- **`no_date` is in the tens or in the thousands**: a pattern is mistyped (a missing space or $). Paste it again from the teacher's text.

At home A column `preu` holds text like `1.250,75`. Write the expression that makes it a number, and explain each step.