The dirt inventory

Look before you clean: what is wrong with `games`?

Names _____

Superset → **SQL → SQL Lab**, database `Dades classe`, schema `public`, table `games`: the loot chest you uploaded on Monday, one row per game. Pairs, one types and one writes; swap at Task 3. Run **one statement at a time**: select it, then **Run**. Today you fix nothing: you open the chest, count what is wrong and write it down.

Reference · four kinds of dirt

Kind	In the chest	In a table	How to find it
Missing	An empty slot.	A `NULL`: no value at all.	`COUNT(*) - COUNT(col)`
Duplicate	The same item twice.	The same real thing in two rows.	`GROUP BY` + `HAVING COUNT(*) > 1`
Wrong type or format	A sword stored as a picture of a sword.	A date or a number stored as text.	Look at the type; sort it; `LIKE`
Outlier	An item with 9,999 damage.	A value far from all the others. An error, or real?	`MIN`, `MAX`, `ORDER BY ... DESC`

`COUNT(*)` counts rows · `COUNT(col)` counts the rows where `col` is **not** empty · in a `LIKE` pattern, `%` is any text and _ is exactly one character.

Task 1 · Size and types

```
SELECT COUNT(*) FROM games;
SELECT column_name, data_type FROM information_schema.columns
WHERE table_name = 'games' ORDER BY ordinal_position;
```

Rows _____ Columns _____ Two columns are `text` and should not be: _____ and _____

Task 2 · Missing values

One `COUNT(*) - COUNT(col)` per column, all ten in a single `SELECT`. It starts like this:

```
SELECT COUNT(*) - COUNT(price) AS no_price, COUNT(*) - COUNT(game) AS no_game   -- ...and eight more
FROM games;
```

`number` _____ `game` _____ `release_date` _____ `price` _____ `owners` _____

`developer` _____ `publisher` _____ `average_playtime` _____ `median_playtime` _____ `metascore` _____

The worst column is _____ : empty in _____ % of the rows.

Now the missing values in disguise:

```
SELECT COUNT(*) FROM games WHERE average_playtime = 0;
SELECT COUNT(*) FROM games WHERE price = 0;
SELECT game, price FROM games WHERE game = 'Dota 2';
```

`average_playtime = 0` in _____ rows. Did nobody play them, or did nobody measure? _____

`price = 0` in _____ rows. Dota 2 is free to play. So what does an empty price mean? _____

Task 3 · Wrong formats

This query should list the three oldest games. Run it.

```
SELECT game, release_date FROM games ORDER BY release_date LIMIT 3;
```

First date _____ Second date _____ Third date _____

Is that oldest first? Why did the database sort them like this? _____

Most dates look like `Nov 16, 2004`. Find the ones that do not, then look at `owners`:

```
SELECT release_date, COUNT(*) AS n FROM games
WHERE release_date NOT LIKE '___ %, ____'       -- 3 letters, space, day, comma, 4 digits
GROUP BY release_date ORDER BY n DESC;
SELECT owners, COUNT(*) AS n FROM games GROUP BY owners ORDER BY n DESC;
SELECT game, owners FROM games ORDER BY owners DESC LIMIT 3;
```

What do most of the odd dates lack? _____ One is odd in another way: _____

Rows with an odd date (`SELECT COUNT(*)` with the same `WHERE`) _____ Different values of `owners` _____

The third query should give the most owned games. It says _____

Why is that wrong? _____

Task 4 · Duplicates: is any game in the table twice?

```
SELECT game, COUNT(*) AS n FROM games WHERE game IS NOT NULL
GROUP BY game HAVING COUNT(*) > 1 ORDER BY n DESC, game;
SELECT number, game, release_date, developer FROM games WHERE game = 'Dark Matter';
SELECT COUNT(*) AS all_rows, COUNT(DISTINCT number) AS numbers FROM games;
```

Names that appear more than once _____ Rows called `Dark Matter` _____

Are they the same game? What tells you? _____

`all_rows` _____ `numbers` _____ Is `number` an identifier? _____

Task 5 · Outliers (prices are in US dollars)

```
SELECT MIN(price), MAX(price), ROUND(AVG(price)::numeric, 2) AS mean FROM games;
SELECT game, price, developer FROM games ORDER BY price DESC NULLS LAST LIMIT 8;
```

Min _____ Max _____ Mean _____ The most expensive row: _____

Is it a typing error? Your reason: _____

Done when Every blank is filled, and the inventory, on a separate sheet, has at least eight rows: column, kind of dirt, evidence (the query and its number). Leave **Decision** empty: we fill it together. **Fast pair?** Run Task 2 on your own open dataset and add its three worst columns.

Stuck?

- **`relation "games" does not exist`**: check the schema, it must be `public`.
- **Only one result for three queries**: SQL Lab shows the last one. Run them one at a time. **Only 1,000 rows come back**: that is the display limit, not the table. Count with `COUNT(*)`.
- "There are NULLs" is not evidence. No number, no evidence. Still stuck after two tries? Ask your neighbours, then the teacher.

At home Finish the inventory with its **Decision** column. Then three inventory lines for your own open dataset: does any of its `NULL` mean something?