

PostgreSQL Data Cleaning & Pattern Matching Summary

1. String & Data Transformation

Function	Usage	Example
INITCAP()	Capitalize first letter of each word.	INITCAP('hello world') -> 'Hello World'
REPLACE()	Swap text values.	REPLACE('A-1', '-', '_') -> 'A_1'
LPAD() / RPAD()	Padding strings to a fixed length.	LPAD('7', 3, '0') -> '007'
SPLIT_PART()	Split string by delimiter and pick part.	SPLIT_PART('A/B/C', '/', 2) -> 'B'
COALESCE()	Replace NULL with a default value.	COALESCE(bonus, 0)

2. Date & Time Handling (The Crucial Part)

 **PostgreSQL Default Format:** دايماً يبخزن التاريخ بتنسيق YYYY-MM-DD .

Function	Purpose	Key Patterns
TO_DATE()	String -> Date	TO_DATE('2026/02/20', 'YYYY/MM/DD')
TO_CHAR()	Date/Timestamp -> String	TO_CHAR(NOW(), 'Day, DDth')
EXTRACT()	Get specific unit (Year, Month, etc.)	EXTRACT('month' FROM date_col)
TO_TIMESTAMP()	String -> Timestamp	TO_TIMESTAMP('23:15', 'HH24:MI')

3. Pattern Matching: The "Filtering" Battle

A. LIKE & ILIKE (Standard SQL)

- تستخدم الـ Wildcards البسيطة فقط:

- % : أي عدد من الحروف.
- _ : حرف واحد فقط.
- .(Small Capital مش مهم Case-Insensitive) هي النسخة الـ ILIKE.

B. SIMILAR TO (SQL Standard Hybrid)

- خليط بين الـ LIKE والـ Regex.
- بيفهم الـ | (أو) والـ [] (مجموعة حروف).
- لا يفهم الـ {n} للتكرار في أغلب الأنظمة. !

C. POSIX Regular Expressions (The Pro Way)

- تستخدم الـ Operators دي في Postgres :
 - ~ : Regex (حساس للحروف).
 - ~* : Regex (تجاهل حالة الحروف).
 - مثال لـ 4 حروف: '\$column ~ '^[a-zA-Z]{4}'
-

4. Basic Regex Meta-characters

- a \d : أي رقم (9-0).
- b ? : العنصر السابق موجود 0 أو 1 مرة.
- c + : العنصر السابق موجود 1 أو أكثر.
- d * : أي حرف موجود 0 أو أكثر.
- e [] : أي حرف جوه الأقواس.