EXPRESIONES REGULARES

PostgreSQL admite una variedad de expresiones regulares para buscar y manipular datos.

Listado de caracteres especiales

^: Indica el inicio de la cadena.
$: Indica el final de la cadena.
.: Coincide con cualquier carácter excepto con el salto de línea.
[]: Encierra un conjunto de caracteres permitidos. Por ejemplo, [abc] coincidirá con cualquier cadena que contenga los caracteres a, b o c.
[^]: Encierra un conjunto de caracteres no permitidos. Por ejemplo, [^abc] coincidirá con cualquier cadena que no contenga los caracteres a, b o c.
-: Indica un rango de caracteres permitidos. Por ejemplo, [a-z] coincidirá con cualquier letra minúscula del alfabeto inglés.
|: Indica una alternativa. Por ejemplo, a|b coincidirá con cualquier cadena que contenga a o b.
*: Indica que el patrón anterior se puede repetir cero o más veces.
+: Indica que el patrón anterior se puede repetir una o más veces.
?: Indica que el patrón anterior es opcional, es decir, que puede aparecer cero o una vez.
{n}: Indica que el patrón anterior se debe repetir exactamente n veces.
{n,}: Indica que el patrón anterior se debe repetir al menos n veces.
{n,m}: Indica que el patrón anterior se debe repetir al menos n veces y como máximo m veces.
\: Indica que el siguiente carácter es un carácter especial. Por ejemplo, \. coincidirá con el carácter . literal.

Ejemplos de expresiones regulares en PostgreSQL:

•~ : Esta expresión se utiliza para buscar una cadena que cumpla con la expresión regular. Por ejemplo, la expresión nombre ~ '^[A-Za-z]+$' buscará en la columna nombre todos los valores que contengan solo letras y espacios.
•~* : Esta expresión es similar a ~, pero es insensible a mayúsculas y minúsculas. Por ejemplo, la expresión nombre ~* '^[a-z]+$' buscará en la columna nombre todos los valores que contengan solo letras minúsculas.
•LIKE : La cláusula LIKE se utiliza para buscar una cadena que contenga un patrón especificado. Por ejemplo, la expresión nombre LIKE 'S%' buscará en la columna nombre todos los valores que comiencen con la letra 'S'.

Ejemplos:

- ^[A-Za-z]+$: esta expresión regular permite que una columna solo contenga letras y espacios.

ALTER TABLE mi_tabla ADD CONSTRAINT columna_check CHECK (columna ~ '^[A-Za-z]+$');

- ^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$: esta expresión regular permite que una columna solo contenga direcciones de correo electrónico válidas.

ALTER TABLE mi_tabla ADD CONSTRAINT columna_check CHECK (columna ~ '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$');

Realiza los siguientes ejercicios de expresiones regulares.

1-Crear una tabla llamada usuarios con los siguientes campos: id (entero), nombre (cadena de texto) y email (cadena de texto). Agregar una restricción de CHECK para asegurarse de que el correo electrónico sea válido.

2-Añade el campo código_postal a la tabla usuarios. Agrega una restricción de CHECK para validar los códigos postales .El patrón busca una cadena que contenga exactamente cinco dígitos.

3- Añade el campo teléfono a la tabla usuarios. Agrega una restricción de CHECK para validar si un número de teléfono es válido en un formato específico. El patrón busca una cadena que contenga exactamente nueve dígitos.