

# 93\_Find Valid Emails Easy - Solution

Source - <https://leetcode.com/problems/find-valid-emails/description/>

Running Notes:

- find all the **valid email addresses**.
- Valid email address:
  - It contains exactly one @ symbol.
  - It ends with .com .
  - The part before the @ symbol contains only **alphanumeric** characters and **underscores**.
  - The part after the @ symbol and before .com contains a domain name **that contains only letters**.

```
-- Write your PostgreSQL query statement below
SELECT *
FROM Users
WHERE email ~* '^[a-zA-Z\d]+@[a-zA-Z]+\com$'
```

This question was easy.

Whenever I want to validate email ids it has a set of rules, and all rules can be incorporated together using a regular expression. With postgresQL we use a POSIX expression and all the above conditions were incorporated in this POSIX expression.

Part	Meaning
^	Asserts the <b>start</b> of the string (ensures the match starts from the beginning).
[a-zA-Z\d]+	Matches <b>one or more</b> characters that are: - Uppercase letters ( A-Z ) - Lowercase letters ( a-z ) - Digits ( 0-9 ) This is the <b>username</b> or <b>local part</b>

	of the email.
@	Matches the @ symbol — separates the local part from the domain.
[a-zA-Z]+	Matches <b>one or more letters</b> — the domain name (e.g., <code>gmail</code> , <code>yahoo</code> , etc.). No numbers or hyphens allowed here.
\.	Matches a <b>literal dot</b> ( <code>.</code> ). The backslash escapes the dot, preventing it from acting as a wildcard.
com	Matches the string <code>"com"</code> exactly — enforces <code>.com</code> emails only.
\$	Asserts the <b>end</b> of the string — ensures the email ends in <code>.com</code> (not <code>.com.au</code> , <code>.company</code> , etc.).

Written By,  
Harshee Pitroda