Exercício 8

Mostre os anfitriões que têm pelo menos 3 imóveis, trazendo seu nome, sua cidade e a quantidade de imóveis dos quais ele é dono

```
%%sql
SELECT U.user_id, U.nome,'Rio de Janeiro' AS cidade, count(*) AS qtt_propriedades
FROM Usuario U JOIN Propriedade P USING(user_id)
GROUP BY U.user_id, U.nome, U.sobrenome
HAVING count(*) >= 3
LIMIT 20
```

[131] ✓ 0.0s Python

```
postgresql://ziel:***@localhost:5432/template1
 * postgresql://ziel:***@localhost:5432/trabalho2
20 rows affected.
```

user_id	nome	cidade	qtt_propriedades
11739	Eric	Rio de Janeiro	9
19065	Morena	Rio de Janeiro	3
48024	Gustavo	Rio de Janeiro	7
70933	Goitaca	Rio de Janeiro	10
132230	Pierre	Rio de Janeiro	3
149407	Ricardo	Rio de Janeiro	24
153721	Shushma	Rio de Janeiro	32
224192	David	Rio de Janeiro	18
235496	Paola	Rio de Janeiro	4
240289	Maria	Rio de Janeiro	6
241334	Andrea	Rio de Janeiro	4
261954	Elisa	Rio de Janeiro	3
319672	Valeria	Rio de Janeiro	5
336908	Alexandre	Rio de Janeiro	5
357307	Dircélia	Rio de Janeiro	5
403781	Dieine	Rio de Janeiro	3
406989	Casa 48 Guesthouse	Rio de Janeiro	5
413577	Bernardo	Rio de Janeiro	6
449677	Renato / Louri	Rio de Janeiro	6
459544	Temporada	Rio de Janeiro	31

Realimos o `JOIN` das tabelas *Usuario* e *Propriedade* a fim de se obter as propriedades por usuário e filtramos, pela cláusula `HAVING` por aqueles que possuem pelo menos 3 imóveis.

Fixamos a cidade como Rio de Janeiro, tendo em vista que todas as tabelas envolvidas são referentes a dados do AirBnB da cidade do Rio de Janeiro.

Crie um índice que agilize a execução dessa consulta

```
%%sql
-- Índice na coluna user_id da tabela Propriedade
CREATE INDEX idx_propriedade_user_id ON Propriedade(user_id);
-- Índice composto nas colunas user_id, nome e sobrenome da tabela Usuario
CREATE INDEX idx_usuario_user_id_nome_sobrenome ON Usuario(user_id, nome, sobrenome);
```

[132] ✓ 0.6s Python

```
postgresql://ziel:***@localhost:5432/template1
 * postgresql://ziel:***@localhost:5432/trabalho2
Done.
Done.
[]
```

1. **Índice na tabela `Propriedade`**:
    - `CREATE INDEX idx_propriedade_user_id ON Propriedade(user_id);`
    - Esse índice acelera a junção entre `Usuario` e `Propriedade`, facilitando a busca dos registros na tabela `Propriedade` usando a coluna `user_id`.

2. **Índice composto na tabela `Usuario`**:
    - `CREATE INDEX idx_usuario_user_id_nome_sobrenome ON Usuario(user_id, nome, sobrenome);`
    - Esse índice é benéfico para operações de agrupamento (`GROUP BY`) e filtragem (`HAVING`). Como a consulta agrupa por `user_id`, `nome`, e `sobrenome`, o índice composto ajuda a reduzir o tempo de busca e processamento desses registros.