

*--- Creé las tablas en una BD de prueba*

```
CREATE TABLE PUBLIC.prueba(  
    Fecha DATE, Nombre VARCHAR(100),  
    Teléfono INT8,  
    Correo_electrónico VARCHAR(100),  
    Precio_desde REAL,  
    Ciudad VARCHAR(100),  
    Tipo_Vivienda VARCHAR(100),  
    Region VARCHAR(100)  
);  
CREATE TABLE PUBLIC.region(  
    Ciudad VARCHAR(100),  
    Region VARCHAR(100)  
);
```

*--- Cargué las tablas para realizar los Queries solicitados*

```
COPY PUBLIC.prueba FROM 'D:\2023\Prueba Estrenar Vivienda\Prueba\Pruebalimpio.csv'  
DELIMITER ',' CSV HEADER;  
COPY PUBLIC.region FROM 'D:\2023\Prueba Estrenar Vivienda\Prueba\Regiones.csv' DELIMITER ','  
CSV HEADER;
```

*--- Agregué la columna "id" en la tabla "prueba" para garantizar el  
contéo de todos los datos y evitar pérdida de información*

```
ALTER TABLE prueba ADD COLUMN id SERIAL PRIMARY KEY;
```

*--- Asigné el "id" a cada registro*

```
UPDATE prueba SET id = subquery.new_id  
FROM (  
    SELECT id, row_number() OVER (ORDER BY (SELECT NULL)) AS new_id  
    FROM prueba  
    ) AS subquery  
WHERE prueba.id = subquery.id;
```

### --- Queries Solicitadas:

---

/\*

Hacer un Query que muestre la cantidad de usuarios que se registraron agrupado por regiones, teniendo en cuenta no contar las filas con datos vacíos en el campo 'Nombre', 'Telefono' y 'Correo electrónico':

\*/

```
SELECT r.Region, COUNT(pr.id) AS Cantidad_Usuarios_Region
```

```
FROM region r
```

```
RIGHT JOIN prueba pr ON r.Ciudad = pr.Ciudad
```

/\* Puesto que en cada la tabla original, hay dos registros que no tienen "Ciudad",

preferí utilizar "RIGHT JOIN" para garantizar el contéo total de los datos \*/

```
WHERE pr.Nombre IS NOT NULL AND pr.Teléfono IS NOT NULL AND pr.Correo_electrónico IS NOT NULL
```

```
GROUP BY r.Region
```

```
ORDER BY r.Region ASC;
```

	region character varying (100) 🔒	cantidad_usuarios_region bigint 🔒
1	BOGOTÁ	25
2	CARIBE	13
3	CENTRAL	21
4	ORIENTAL	4
5	SUR OCCIDENTE	20
6	[null]	2

---

/\*

Hacer un Query que cree una nueva columna llamada 'Tipo\_Vivienda' teniendo en cuenta el punto g del ejercicio anterior y Mostrar la cantidad de usuarios que se registraron agrupado por 'Tipo\_Vivienda', teniendo en cuenta no contar las filas con datos vacíos en el campo 'Nombre', 'Telefono' y 'Correo electrónico'

\*/

```
SELECT pr.Tipo_Vivienda, COUNT(pr.id) AS Cantidad_Usuarios_Tipo_Vivienda
```

```
FROM prueba pr
```

```
WHERE pr.Nombre IS NOT NULL AND pr.Teléfono IS NOT NULL AND pr.Correo_electrónico IS NOT NULL
```

```
GROUP BY pr.Tipo_Vivienda
```

```
ORDER BY pr.Tipo_Vivienda ASC;
```

	tipo_vivienda character varying (100) 🔒	cantidad_usuarios_tipo_vivienda bigint 🔒
1	ALTA	7
2	MEDIA	28
3	VIS	48
4	[null]	2