BITÁCORA

1 LIMPIEZA DE DATOS

Comenzamos con la limpieza de datos, para saber con que datos podíamos trabajar para sacar los kpis.
Nos encontramos con columnas con muchos datos NULL, también varios datos que correspondían a otros paises y cuidades diferentes a Madrid- España.

```
#Obtenemos las columnas cuya suma de valores nulos es mayor a 1000
nulldata = df_airbnb.isnull().sum()
print(nulldata[nulldata > 1000].sort_values(ascending=False))
```

```
#A pesar de que el dataset es de Madrid podemos observar como hay otros paises y ciudades
print(pd.unique(df_airbnb["Country"]))

print(pd.unique(df_airbnb["City"]))
```

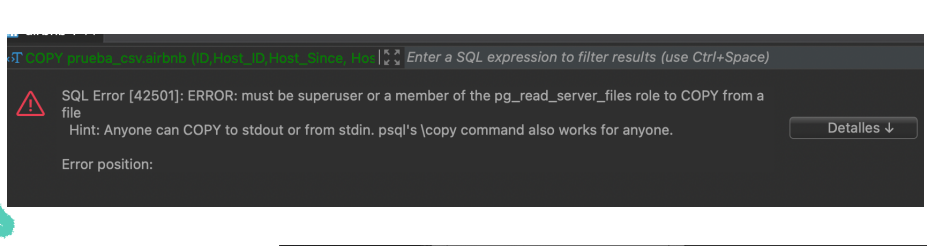
2 DECIDIR CON CUALES DATOS TRABAJAREMOS

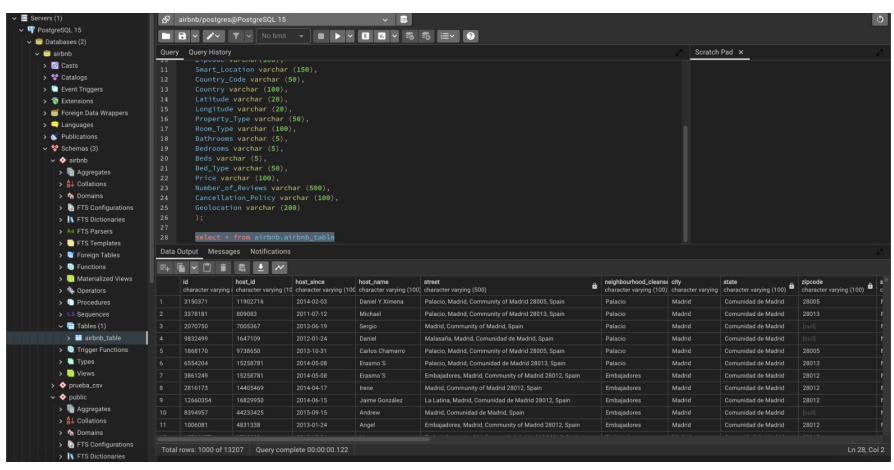
Tomamos la decisión de cuales datos son los mas completos y los cuales nos ayudarían a definir nuestro Storytelling, Kpis, las supociones iniciales.

```
#Creamos un nuevo df donde seleccionamos las columnas cuyos valores sean compatibles con lo que queremos estudiar
df = df_airbnb[['ID','Host ID', 'Host Since', 'Host Name', 'Street', 'Neighbourhood Cleansed', 'City', 'State', 'Zipcode',
                'Smart Location', 'Country Code', 'Country', 'Latitude', 'Longitude', 'Property Type', 'Room Type', 'Bathrooms', 'Bedrooms', 'Beds', 'Bed Type',
                'Price','Number of Reviews', 'Cancellation Policy', 'Geolocation']]
```

3 DEFINIR E IMPLEMENTAR EL DATAWAREHOUSE

Utilizarmo Dbeaver con PostreSQL para importa el csv; al crear la consulta sql para importar el csv limpio nos dió un error que no pudimos solucionar con Dbeaver, decimos utilizar PgAdmin y se puedo importar el csv con la consulta que habíamos realizado.





4 REGRESION LINEAL

Al realizar la regresión lineal al principio no incluimos el dato de la cantidad de habitaciones que tenía cada alojamiento, dándonos cuenta posteriormente que había que incluirlo para que diera valores reales.

```
rsquared = multi_regression.score(x_train, y_train)
print(f'Training data r-squared: {rsquared:.2}')

# Sin beds da r = 0.25 se decide incluir beds
```

[239] ✓ 0.0s

```
Training data r-squared: 0.29
```

```
rsquared = multi_regression.score(x_train, y_train)
print(f'Training data r-squared: {rsquared:.2}')

# Sin beds da r = 0.29 se decide añadir el numero de camas
```

Python

```
Training data r-squared: 0.31
```

```
example = pd.DataFrame.from_dict({'Bedrooms':[4],'Bathrooms':[1],'Beds':[4]})
prediction = multi_regression.predict(example)[0]
print(f"According to the model, a home with 3 bedrooms, 3 beds and 1 bathroom would cost {prediction:.2f}/night")
```

Python

```
According to the model, a home with 3 bedrooms, 3 beds and 1 bathroom would cost 137.40/night
```