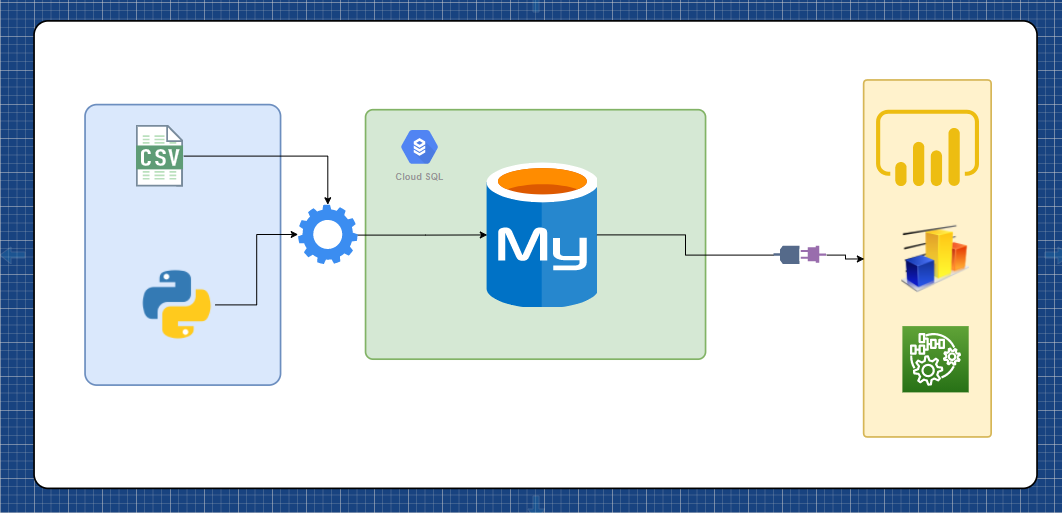
**Conexión a MySql desde Python**

En la presente práctica se realizará conexión a base de datos mysql en nube de Google desde scipt Python para realizar operaciones de consulta y actualización de datos. Adicionalmente se realizará proceso de cargue de archivo CSV a base de datos y realización de informes desde herramienta Google DataStudio.



# Cargue de base de datos

Se ha creado una base de datos para cada estudiantes. El nombre de la base de datos corresponde a cuenta de correo antes de símbolo “@” y si el símbolo “.” . El usuario y contraseña son el mismo nombre de la base de datos. Así para el estudiante [leandro.pajarof@konradlorenz.edu.co](mailto:leandro.pajarof@konradlorenz.edu.co) los datos de conexión son:

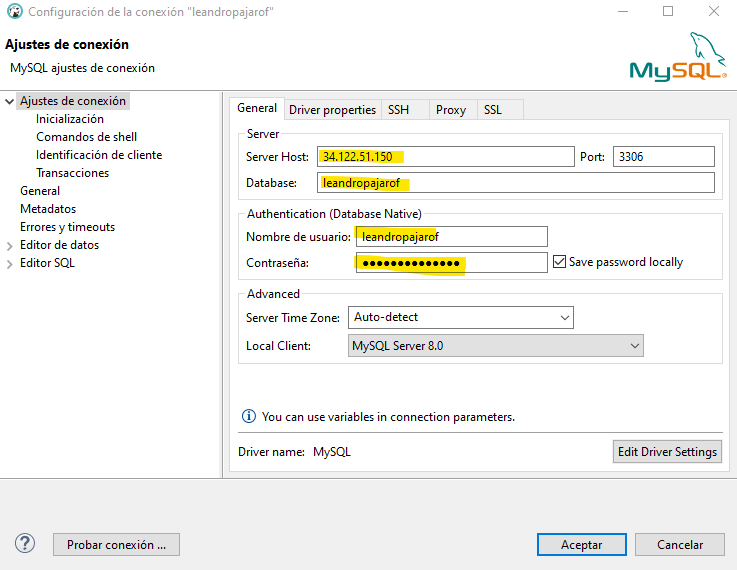
Host: 34.122.51.150

Base de datos: usuario del correo sin el punto

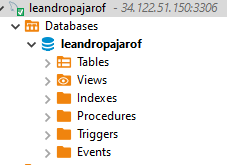
Usuario: usuario del correo sin el punto

Contraseña: usuario del correo sin el punto

## Conexión a base de datos

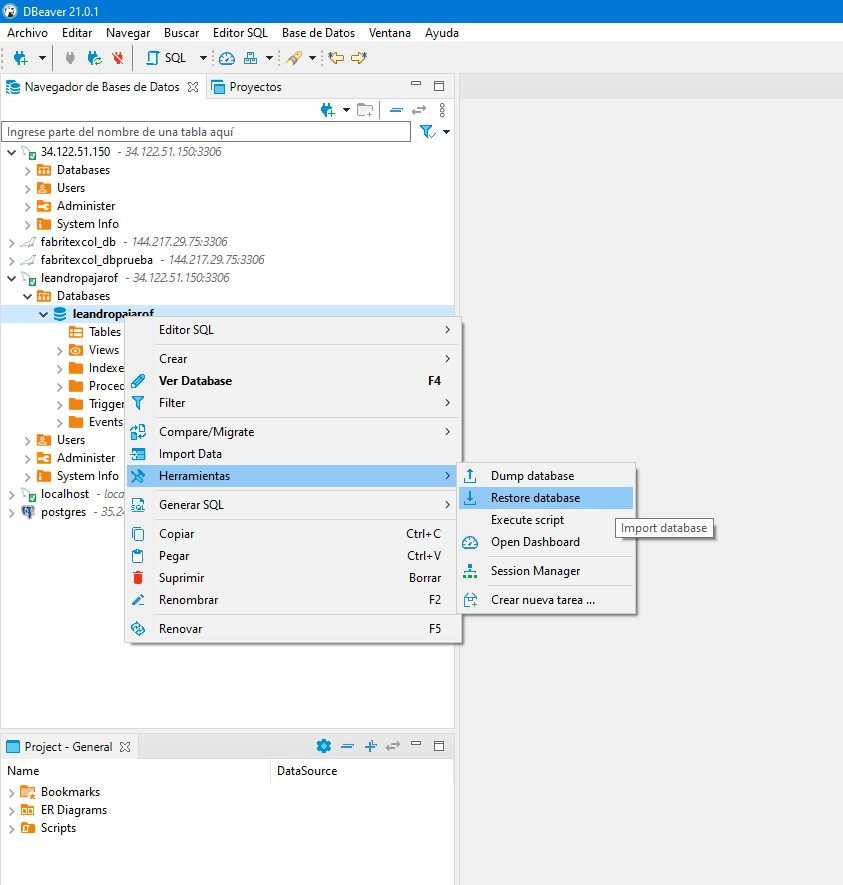


Consulta de bases de datos

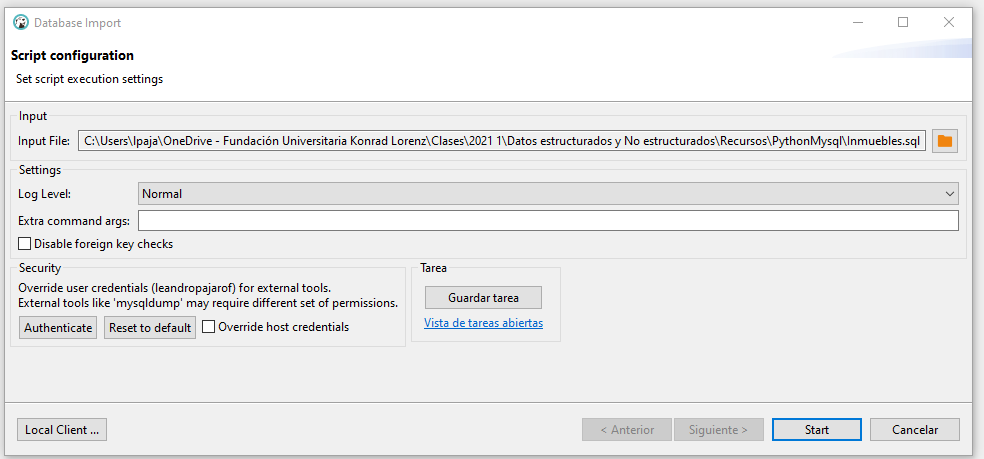


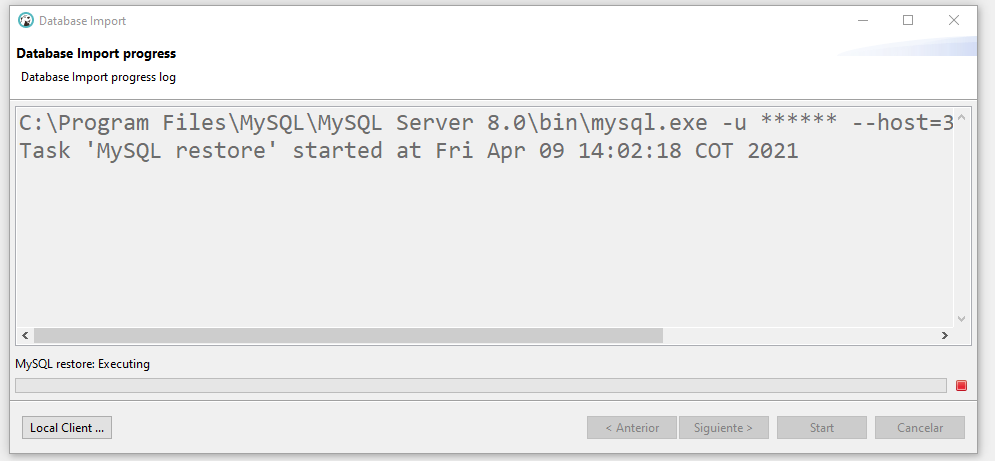
## Importación de base de datos

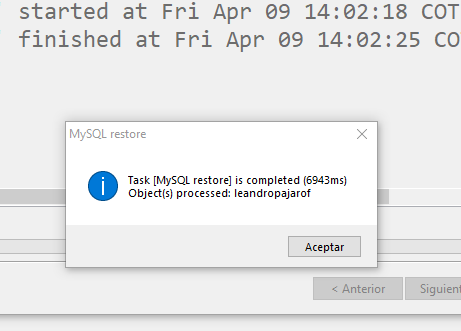
Realizar la importación de la base de datos inmuebles. En la opción “Herramientas” de la base de datos seleccionamos “Restore database”



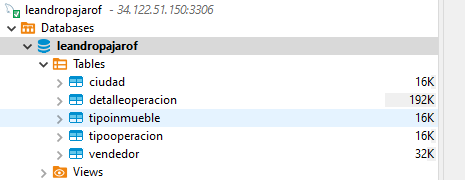
Seleccionamos el archivo “inmuebles.sql”

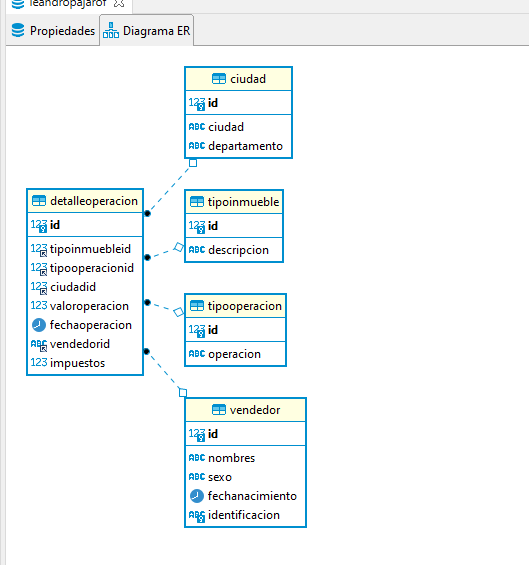






Una vez finalice el proceso cerramos ventana y deberán aparecer las siguientes tablas

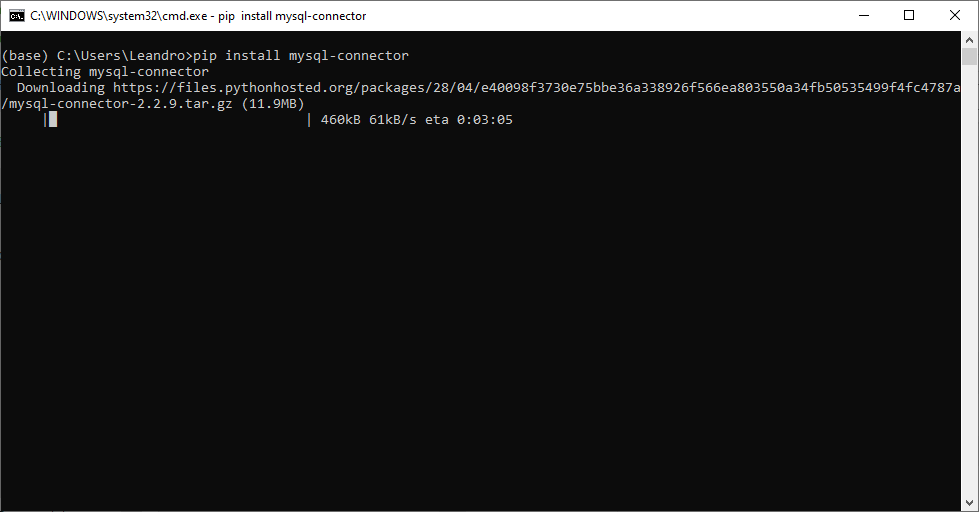




# Conexión Python Mysql

Para establecer conexión desde Python a base de datos Mysql se debe tener instalado “mysql-connector”

En caso de no tenerlo ejecutar pip install mysql-connector en consola anaconda



El siguiente script establece conexión con base de datos

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión: ",e)**

**return conexion**

**conexion=crear\_conexion('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

# Selección de datos

El siguiente script visualiza los registros de la tabla vendedor

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión: ",e)**

**return conexion**

**conexion=crear\_conexion('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**if(conexion != None):**

**cursor = conexion.cursor(dictionary=True)**

**result = None**

**try:**

**cursor.execute("SELECT \* FROM vendedor")**

**result = cursor.fetchall()**

**except Error as e:**

**print("Se ha presentado error: ",e)**

**for row in result:**

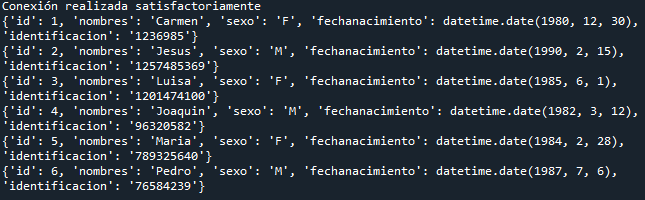
**print(row)**

**else:**

**print('No es posible ejecutar la consulta')**

**#Modificar script para visualizar los datos de la tabla tipoinmueble y de la tabla ciudad**

El resultado de la ejecución sería el siguiente:



# Inserción de registros

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión: ",e)**

**return conexion**

**conexion=crear\_conexion('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**sql = "INSERT INTO vendedor(id,nombres,sexo,fechanacimiento,identificacion) values(7,'Lola','F','1980-09-04','22222')"**

**cursor = conexion.cursor()**

**cursor.execute(sql)**

**conexion.commit()**

**print("Registro ingresado satisfactoriamente")**

**conexion.close()**

**#Modificar el script para crear función que inserte los datos del vendedor pero estos sean ingresados por teclado**

# Modificación de registros

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión: ",e)**

**return conexion**

**conexion=crear\_conexion('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**sql = "UPDATE vendedor SET nombres='Will i am' where id=6"**

**cursor = conexion.cursor()**

**cursor.execute(sql)**

**conexion.commit()**

**print("Registro modificado satisfactoriamente")**

**conexion.close()**

**#Modificar el script para crear función que modifique los datos de nombres, sexo, fechanacimiento digitados por el usuario**

# Eliminación de registros

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión: ",e)**

**return conexion**

**conexion=crear\_conexion('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**def elimina\_vendedor():**

**id = input('Digite id del vendedor a eliminar : ')**

**sql = "DELETE FROM vendedor where id="+id**

**cursor = conexion.cursor()**

**cursor.execute(sql)**

**conexion.commit()**

**print("Registro eliminado satisfactoriamente")**

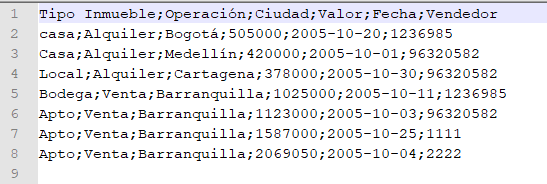
**elimina\_vendedor()**

**conexion.close()**

# Lectura de archivo y carga en Mysql

## Preparación archivo

Se cuenta con archivo “octubre.csv” el cual contiene datos de operaciones del mes de octubre y sobre el cual se debe realizar una preparación de datos antes de poder cargarlo en base de datos. A continuación, muestra del archivo:



Se utilizará script para realizar algunos reemplazos en archivo

El siguiente script realiza algunos reemplazos y genera un nuevo archivo “octubre\_.csv” con los cambios realizados

**import pandas as pd**

**datosventas = pd.read\_csv("octubre.csv",sep=";")**

**print(datosventas)**

**tipoInmueble = datosventas["Tipo Inmueble"]**

**tipoInmueble = tipoInmueble.str.replace('casa','1')**

**tipoInmueble = tipoInmueble.str.replace('Casa','1')**

**tipoInmueble = tipoInmueble.str.replace('Local','3')**

**tipoInmueble = tipoInmueble.str.replace('Bodega','2')**

**tipoInmueble = tipoInmueble.str.replace('Apto','6')**

**datosventas["Tipo Inmueble"] = tipoInmueble**

**operacion = datosventas["Operación"]**

**operacion = operacion.str.replace('Alquiler','1')**

**operacion = operacion.str.replace('Venta','2')**

**datosventas["Operación"] = operacion**

**ciudad = datosventas["Ciudad"]**

**ciudad = ciudad.str.replace('Bogotá','1')**

**ciudad = ciudad.str.replace('Medellín','2')**

**ciudad = ciudad.str.replace('Cartagena','3')**

**ciudad = ciudad.str.replace('Barranquilla','4')**

**datosventas["Ciudad"] = ciudad**

**print(datosventas)**

**datosventas.to\_csv('octubre\_.csv',index=False, sep=";")**

## Cargue de archivo

El siguiente script realiza carga de archivo procesado en punto anterior y genera archivo con errores encontrados

**import mysql.connector**

**from mysql.connector import Error**

**import pandas as pd**

**from datetime import date**

**def crear\_conexion\_db(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión",e)**

**return conexion**

**def insert\_detalle\_operacion(conexion, tipoinmueble, operacion,ciudad,valor,fecha,vendedor):**

**cursor = conexion.cursor()**

**try:**

**impuestos = valor\*0.16**

**sql ='INSERT INTO detalleoperacion(tipoinmuebleid,tipooperacionid,ciudadid,valoroperacion,fechaoperacion,vendedorid, impuestos) '**

**sql = sql + 'values( '+ str(tipoinmueble) + ',' + str(operacion) + ',' + str(ciudad) + ','+ str(valor) +',\''+ str(fecha) +'\','+ str(vendedor) +','+ str(impuestos)+')'**

**cursor.execute(sql)**

**conexion.commit()**

**print("Registro ingresado satisfactoriamente")**

**return True**

**except Error as e:**

**print("Se ha presentado error: ",e)**

**return False**

**conexion = crear\_conexion\_db('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**datosventas = pd.read\_csv("octubre\_.csv",sep=";")**

**errores = pd.DataFrame(columns=('Tipo Inmueble', 'Operación', 'Ciudad', 'Valor','Fecha','Vendedor'))**

**for i in datosventas.index:**

**res = insert\_detalle\_operacion(conexion,datosventas["Tipo Inmueble"][i],datosventas["Operación"][i],datosventas["Ciudad"][i],datosventas["Valor"][i],datosventas["Fecha"][i],datosventas["Vendedor"][i])**

**if(res== False):**

**errores.loc[len(errores)] = [datosventas["Tipo Inmueble"][i],datosventas["Operación"][i],datosventas["Ciudad"][i],datosventas["Valor"][i],datosventas["Fecha"][i],datosventas["Vendedor"][i]]**

**print(errores)**

**errores.to\_csv('Errores'+str(date.today())+'.csv', sep=";",index=False)**

**conexion.close()**

## Creación tabla reporte Google DataStudio

La siguiente función creará tabla con los datos para ser construido reporte en Google Data Studio

**import mysql.connector**

**from mysql.connector import Error**

**def crear\_conexion\_db(nombre\_host, usuario, contrasena, basedatos):**

**conexion = None**

**try:**

**conexion = mysql.connector.connect(**

**host = nombre\_host,**

**user = usuario,**

**passwd = contrasena,**

**database = basedatos**

**)**

**print('Conexión realizada satisfactoriamente')**

**except Error as e:**

**print("Error al establecer conexión",e)**

**return conexion**

**conexion=crear\_conexion\_db('34.122.51.150','** usuario del correo sin el punto **','** usuario del correo sin el punto **','** usuario del correo sin el punto **')**

**def crea\_tabla\_reporte():**

**cursor = conexion.cursor()**

**try:**

**sql ='DROP TABLE IF EXISTS reporte'**

**cursor.execute(sql)**

**sql = """ CREATE TABLE reporte AS**

**SELECT do.id,**

**ti.descripcion Inmueble,**

**top.operacion Operacion,**

**ci.ciudad,**

**ve.nombres,**

**do.fechaoperacion,**

**do.valoroperacion,**

**do.impuestos**

**FROM detalleoperacion do**

**INNER JOIN tipoinmueble ti**

**ON do.tipoinmuebleid = ti.id**

**INNER JOIN tipooperacion top**

**ON do.tipooperacionid = top.id**

**INNER JOIN ciudad ci**

**ON do.ciudadid = ci.id**

**INNER JOIN vendedor ve**

**ON do.vendedorid = ve.identificacion"""**

**cursor.execute(sql)**

**conexion.commit()**

**print("Tabla reporte creada satisfactoriamente")**

**return True**

**except Error as e:**

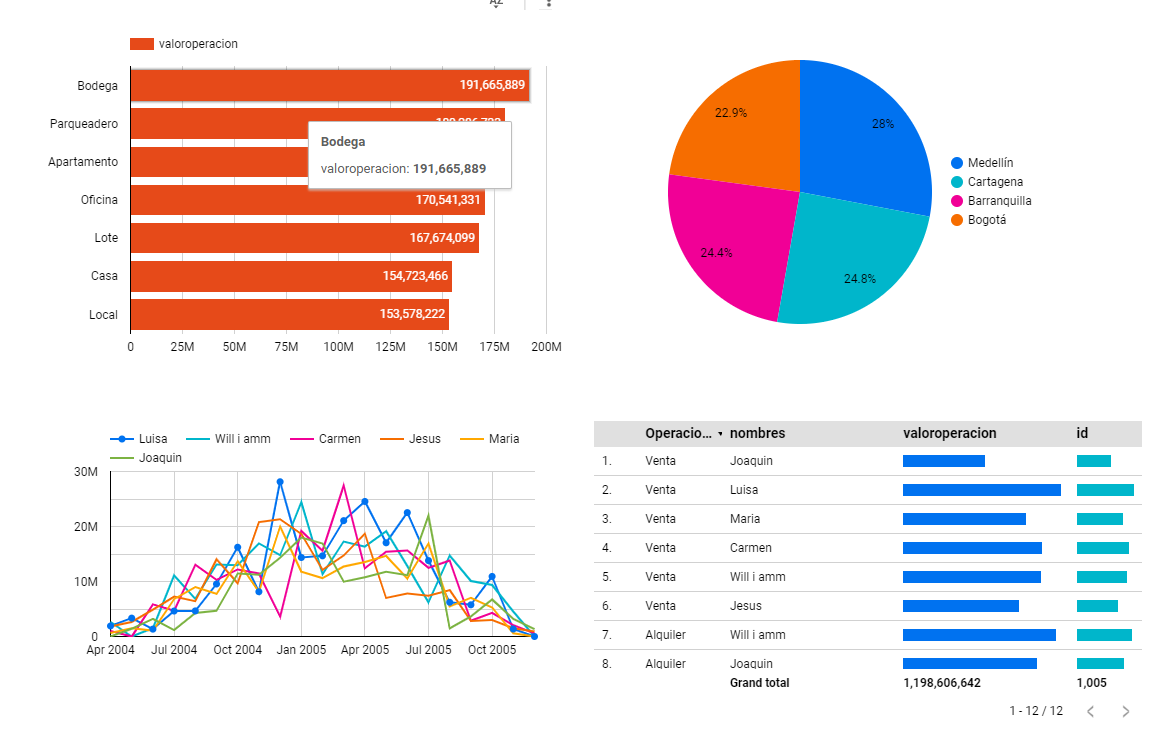
**print("Se ha presentado error: ",e)**

**return False**

**crea\_tabla\_reporte()**

**conexion.close()**

## Informe en Google DataStudio



# Práctica

* Realizar script python para cargar archivo de “Noviembre.csv”
* Realizar script python que permita cambiar el porcentaje de impuestos de cada operación
* Modificar script de creación de tabla reporte para incluir campo de impuestos y generar reporte similar al punto 3.4, pero teniendo en cuenta los impuestos de la operación y no el valor de la operación.
* Generar enlace de consulta de reporte