

Modelo Entidad Relación: Agencia de Viajes

**Asignatura:** Sistemas de gestión de bases de datos y paralelismo de datos

**Profesor:** Sergio García

**Integrantes:**

Leonardo Rivas Duarte

Marta del Carmen Mayoral Santos

MertxeGalán Espiga

Valesca Bravo

Contenido

[Introducción 2](#_Toc125286780)

[1.- Modelo Entidad Relación 3](#_Toc125286781)

[2.- Modelo Relacional 5](#_Toc125286782)

[3.- Secuencias de SQL en un SGBD- Reglas de integridad referencial y/o reglas de negocio 7](#_Toc125286783)

[4.- Carga de datos 10](#_Toc125286784)

[5.- Consultas 15](#_Toc125286785)

[6.- Hemos realizado un modelo relacional. Sin embargo, ¿podrían implementar lo anterior en otro tipo de base de datos? 19](#_Toc125286786)

[7.- Analicen las ventajas y las desventajas de utilizar una base de datos no relacional. 20](#_Toc125286787)

[Conclusión 22](#_Toc125286788)

# Introducción

En el presente trabajo veremos cómo desarrollar una Base de Datos, desde el modelado hasta la implementación de esta en MySQL a través de SQL.

Hemos elegido la industria turística, poniéndonos en el supuesto de una Agencia de viajes. Empezamos nuestra labor, creando un modelo de entidad relación en el programa Día, donde creamos las reglas que nuestro modelo de negocio tendría.

Para pasar el modelo de Entidad Relación a un Modelo Relacional exploramos algunas alternativas de modelado online, como es el caso de [vertabelo](https://vertabelo.com), que brinda oportunidades de exportación para diferentes motores de BBDD y en el caso de ser estudiantes tiene costo cero.

Una vez diseñados ambos modelos, nos encontramos con una página web ([mockaroo](http://www.mockaroo.com)) que nos ayudó a generar una base de datos con los atributos antes creados, además de emplear esta herramienta, también generamos datos aleatorios empleando algunas reglas en Excel.

Finalmente, utilizamos SQL para crear las tablas e insertar la información en MySQL.

# 1.- Modelo Entidad Relación

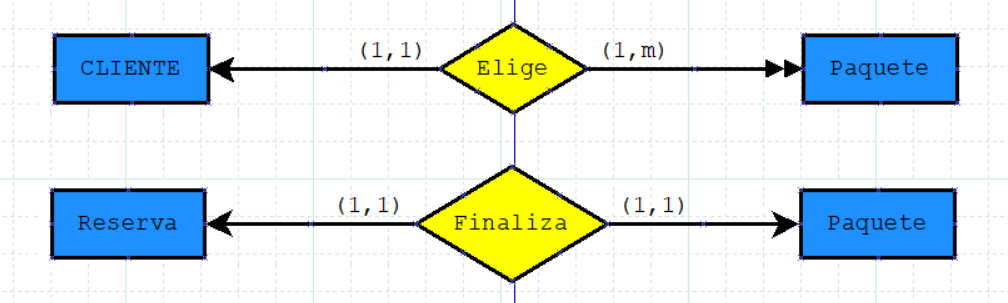
Trabajamos con el programa “Dia”para diagramar el modelo de Entidad Relación usando Notación Bachman.

Para crear nuestra Agencia de Viajes, empezamos definiendo las Entidades y sus atributos:

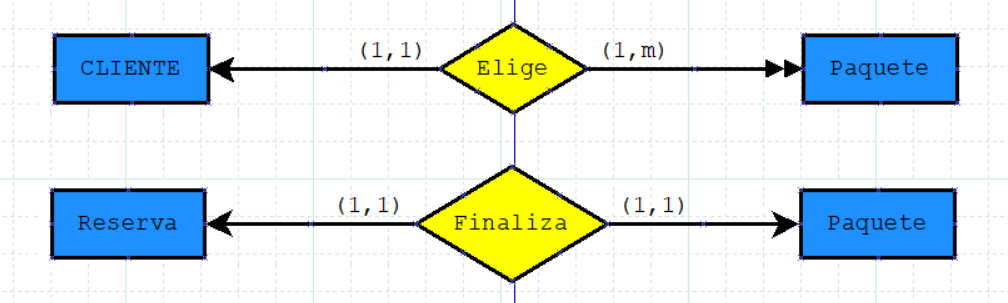
* **Cliente**. Definimos cuáles serían sus atributos; Nombre, Apellido, Género, E-mail, Móvil, año de nacimiento y finalmente ID del cliente, que actuaria para nosotros como Clave Primaria.
* **Paquete.** Llevará su propio código identificatorio que será la Clave Primaria y tendrá otros atributos como Destino, Transporte y Alojamiento.
* **Reserva**. Al igual que Entidad Paquete, tendrá un atributo Clave Primaria identificatoria, además de Precio, Fecha y Número de Personas.

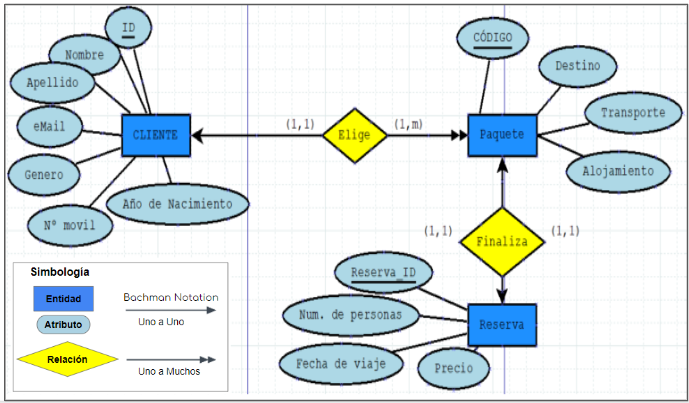
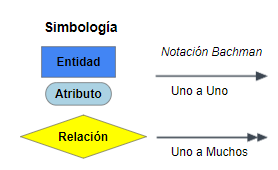
**Reglas y Condiciones:**

* Definimos que la Entidad Cliente se relacionaría con la Entidad Paquete, siendo que un cliente pudiese comprar muchos paquetes, pero paquete solamente tuviese un Cliente.



* La Entidad Paquete y la Entidad Reserva se relacionarían entre sí con una cardinalidad uno a uno.





#### 

# 2.- Modelo Relacional

Para realizar el Modelo Relacional decidimos utilizar ‘Vertabelo’, un modelador de data base online, que usa InformationEngineeringNotation (Cartmell, 2019) como la representación de una relación, permitiendo mostrar cada atributo con el tipo de valor que puede almacenar.

**Reglas y Condiciones:**

* **Cliente:**

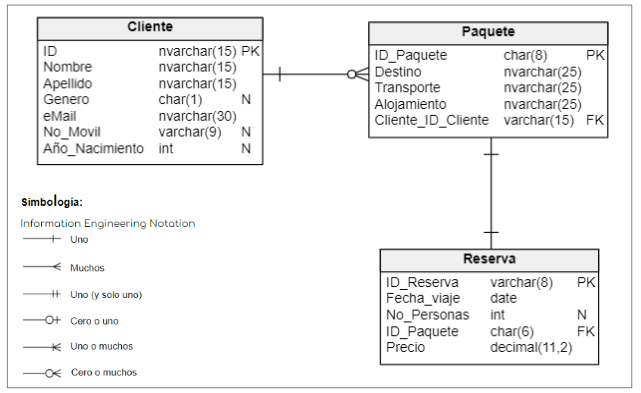
1. ID, usaremos el pasaporte o ID de cada persona, si bien en general los pasaporte se componen de 9 números, decidimos generarlo de 15 espacios para darle libertad a las personas que quieran usar su ID de identificación nacional.
2. Nombre, Apellido y e-Mail, usamos “nvarchar”, porque creemos que es importante tomar ventaja de su propiedad de Unicode caracteres, ya que es una agencia de viajes y tenemos una proyección internacional, aunque este tipo de dato tome el doble de espacio de almacenamiento(2 bytes) que un “varchar” (https://learn.microsoft, 2022).
3. Género, Movil y Año de nacimiento son variables Nullables, pueden o no ser rellenadas por el Cliente. Género es un “char” que acepta solo un carácter, Movil es un “varchar” con nueve espacios y año de nacimiento que decidimos, es un “SmallInt” acá se mestra como “Int” porque la plataforma ‘Vertabelo’ no muestra la opción “SmallInt”.

* **Paquete:**

1. ID\_Paquete no requiere de más de un “char”con 8 espacios.
2. Destino, transporte y Alojamiento se crearon como “nvarchar” con capacidad de 25 espacios.

* **Reserva:**

1. ID\_Reserva, “Varchar” con 8 espacios
2. Fecha de Viaje, se designó como tipo “Date”
3. No\_personas, “Int” con 6 espacios, que será nullable
4. Precio, trabajamos con “Decimal” y no #Float”, ya que el primero es más exacto y estamos trabajando con dinero. (Skov, 2009)

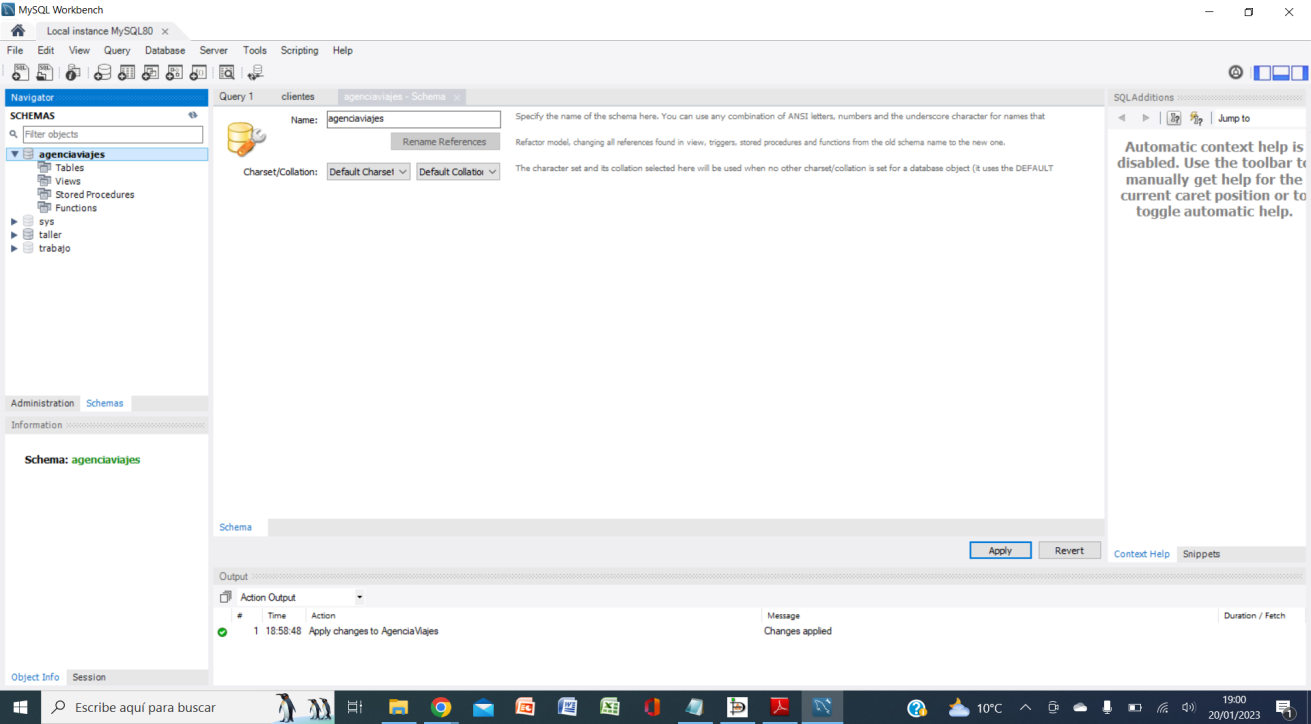


#### 

# 3.- Secuencias de SQL en un SGBD- Reglas de integridad referencial y/o reglas de negocio

En este apartado desarrollamos la implementación de nuestras tablas mediante el sistema de gestión de bases de datos MySQL, y la descripción de nuestras diferentes reglas de integridad.

Primero crearemos nuestra base de datos a la que denominamos agenciaviajes:



Añadimos tres tablas en las que cargaremos nuestros datos, que describimos de la siguiente manera:

CLIENTE

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **NULLS allowed** |
| ID | Documento de identidad | CHAR(15) | N |
| Nombre | Nombre | NVARCHAR (15) | N |
| Apellido | Primer apellido | NVARCHAR (15) | N |
| Email | Dirección email | NVARCHAR (30) | N |
| Genero | Género (M masculino, F femenino) | NVARCHAR (1) | Y |
| Móvil | Número teléfono móvil | NVARCHAR (9) | Y |
| Año de nacimiento | Año de nacimiento | SMALLINT | Y |

PAQUETE

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **NULLS allowed** |
| Código | Código de paquete | CHAR (8) | N |
| Destino | Ciudad de destino | NVARCHAR (25) | N |
| Transporte | Medio de transporte | NVARCHAR (25) | Y |
| Alojamiento | Tipo de alojamiento | NVARCHAR (25) | Y |
| ID | Documento de identidad | NVARCHAR (15) | N |

RESERVA

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Meaning** | **Data Type** | **NULLS allowed** |
| Reservaid | Código de reserva | CHAR (8) | N |
| Num. De Personas | Ciudad de destino | SMALLINT | Y |
| Fecha de viaje | Medio de transporte | DATE | N |
| Precio | Tipo de alojamiento | FLOAT (11,2) | N |
| Código | Documento de identidad | NVARCHAR (15) | N |

* Regla de integridad de la clave primaria

La regla de integridad de clave primaria es una restricción en una base de datos que asegura que cada fila en una tabla tenga un valor único y no nulo en la columna o conjunto de columnas que forman la clave primaria. Esto significa que ninguna fila puede tener un valor vacío o duplicado en la columna de clave primaria.

La clave primaria es un identificador único para cada fila en una tabla y se utiliza para relacionar las tablas entre sí.

Por ejemplo en nuestro BBDD tenemos 3 tablas con una columna que serán primarykey, en la tabla clientes tenemos “ID”, en la tabla paquete tenemos “Código” y en la tabla Reserva tenemos “Reservaid” las cuales son reglas de integridad de clave primaria se aplica automáticamente al crear una tabla con una clave de primaria ya especificada y se asegura que los datos de la tabla sean precisos.

* Regla de integridad de la clave foránea.

La clave foránea se utiliza para establecer una relación entre dos tablas. por ejemplo en nuestras tablas de Cliente tenemos una clave primaria llamada “ID” y nuestra tabla Paquete tenemos tenemos la clave foránea “ID” así se establece una relación entre cada paquete y un cliente en concreto.

* Reglas de negocio

Va a definir aquellas directrices que determinan las actividades comerciales de una organización, pero dentro de nuestro diagrama entidad-relación, estas reglas van a establecer nuestros atributos, relaciones y restricciones determinadas sobre nuestros datos. Como ejemplo, una de nuestras reglas de negocio se basa en que nuestros clientes, pueden tener más de un paquete asociado a ellos, siempre y cuando cada paquete lleve asociado una reserva. Otro ejemplo sería, que todos nuestros clientes deben tener como origen Madrid, independientemente cual sea su destino.

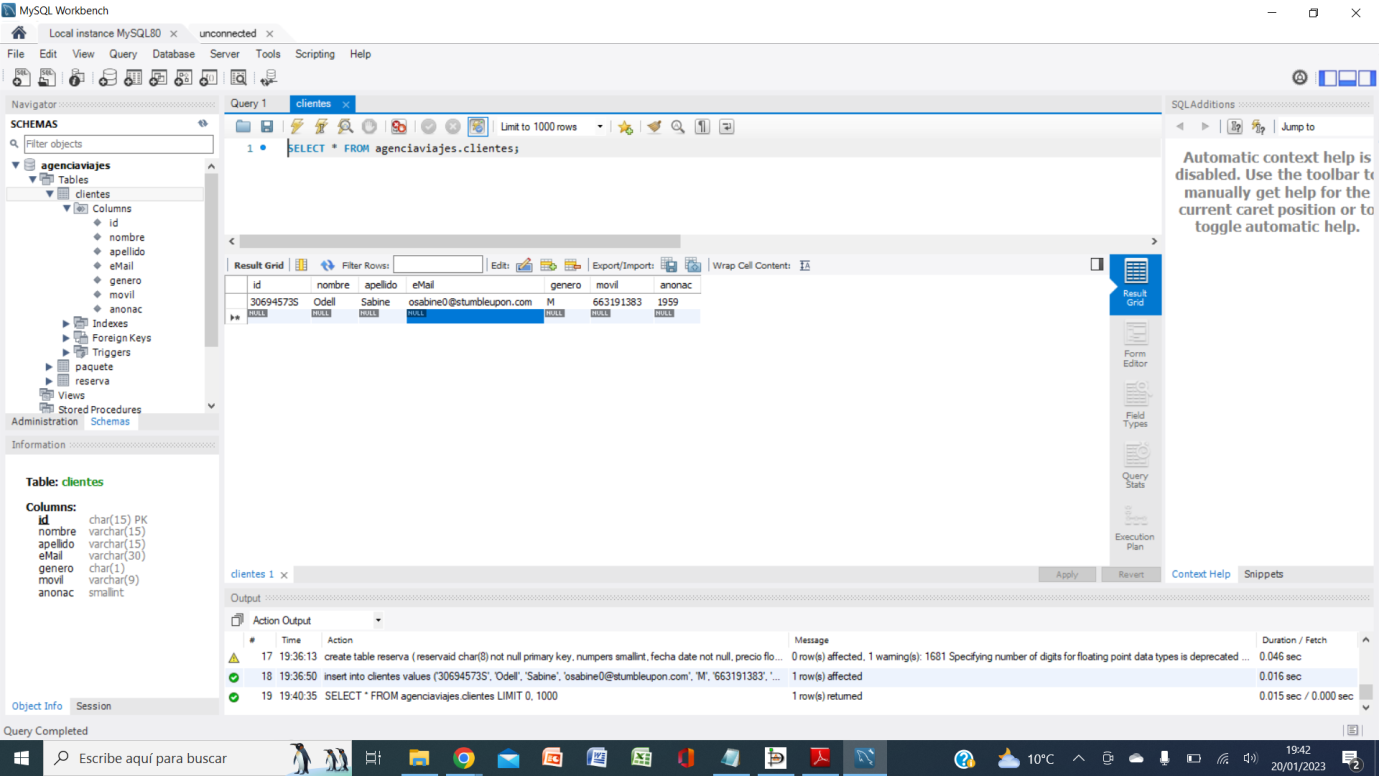
#### 

# 4.- Carga de datos

Vamos a realizar una carga manual de un registro en la tabla de clientes, con la siguiente sentencia SQL:

insertinto clientes values ('30694573S', 'Odell', 'Sabine', 'osabine0@stumbleupon.com', 'M', '663191383', '1959')

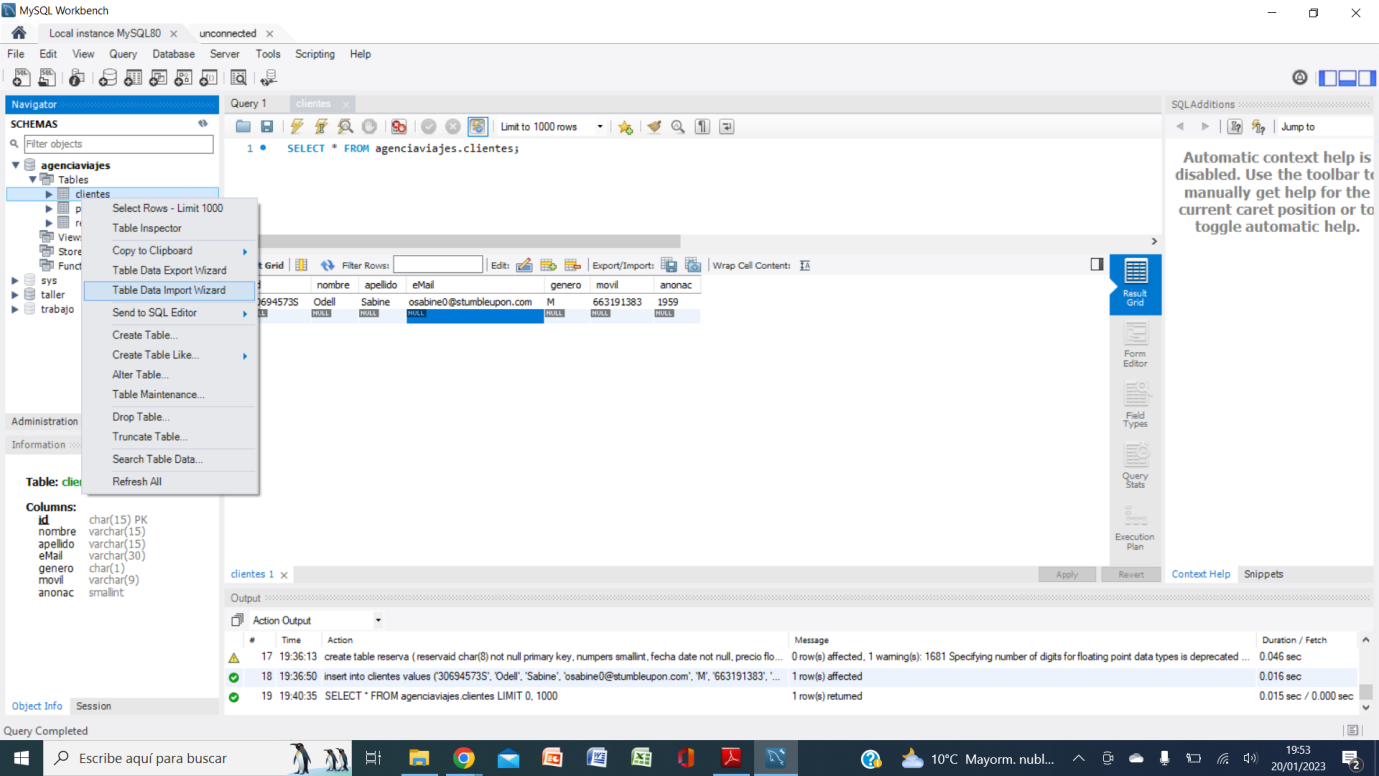
Una vez ejecutada esta sentencia, hacemos la select de la tabla de clientes y obtenemos este registro:

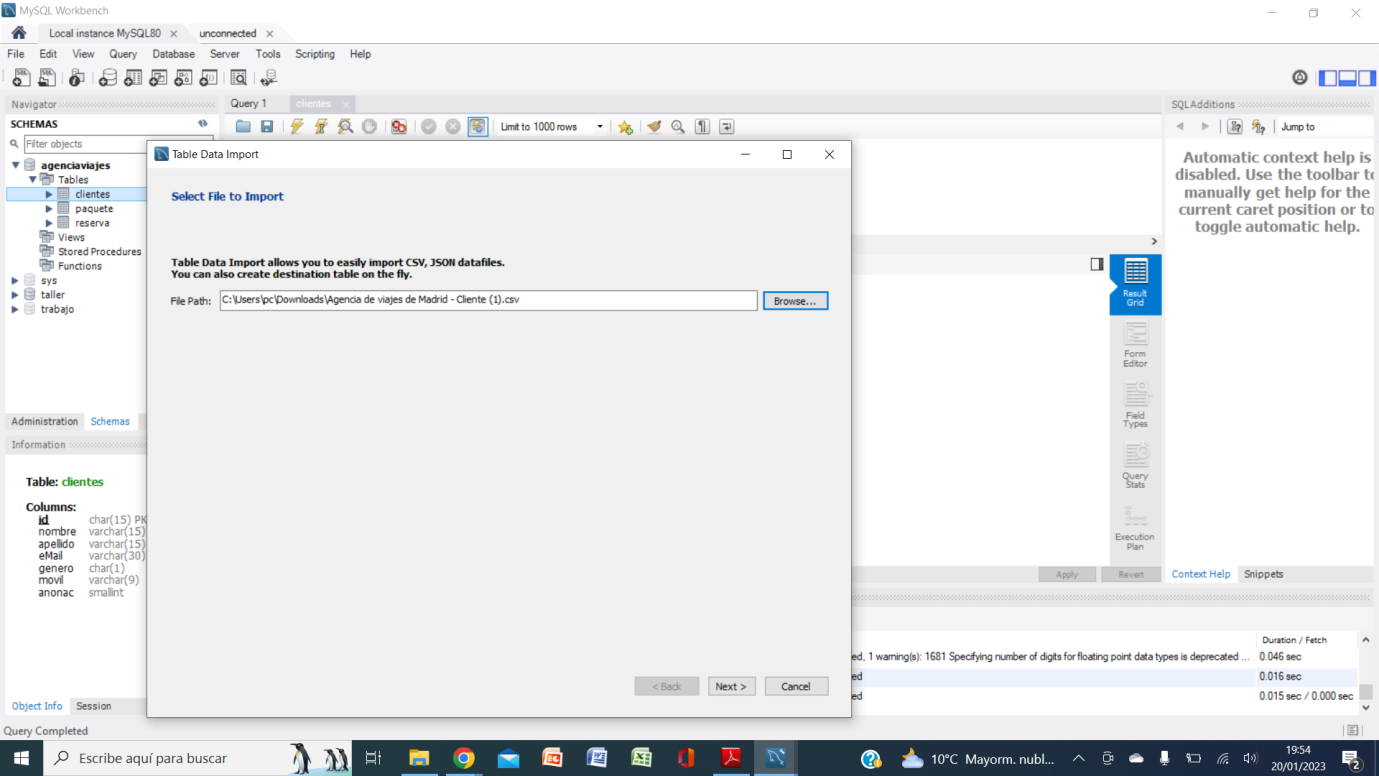


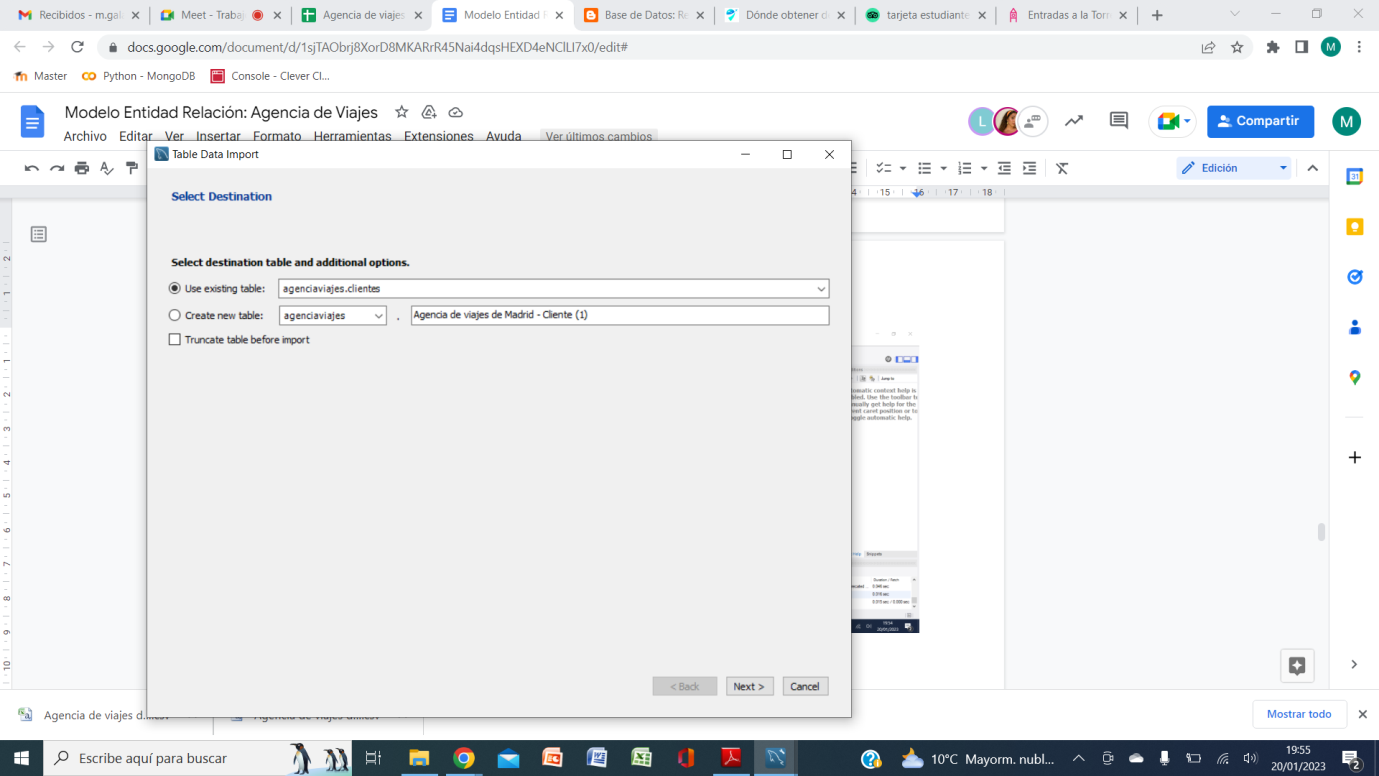
Para optimizar la carga de datos, y conociendo ya como se hace de forma manual, para el resto de datos utilizaremos cargas masivas, mediante la importación desde archivos csv.

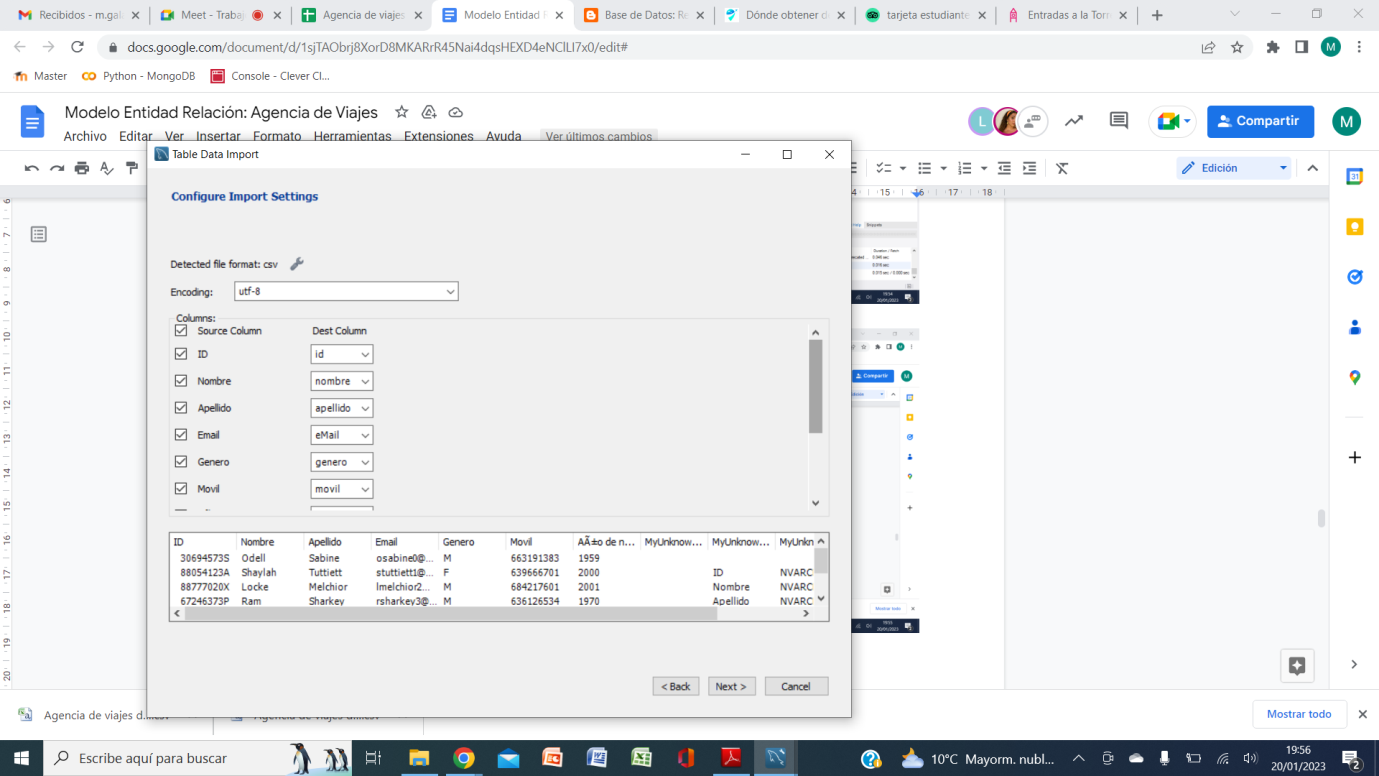
Partimos de un documento excel en drive con 3 hojas (una de clientes, una de paquetes y una de reservas), donde los integrantes del grupo introdujimos datos, y exportamos cada hoja a un archivo con formato csv.

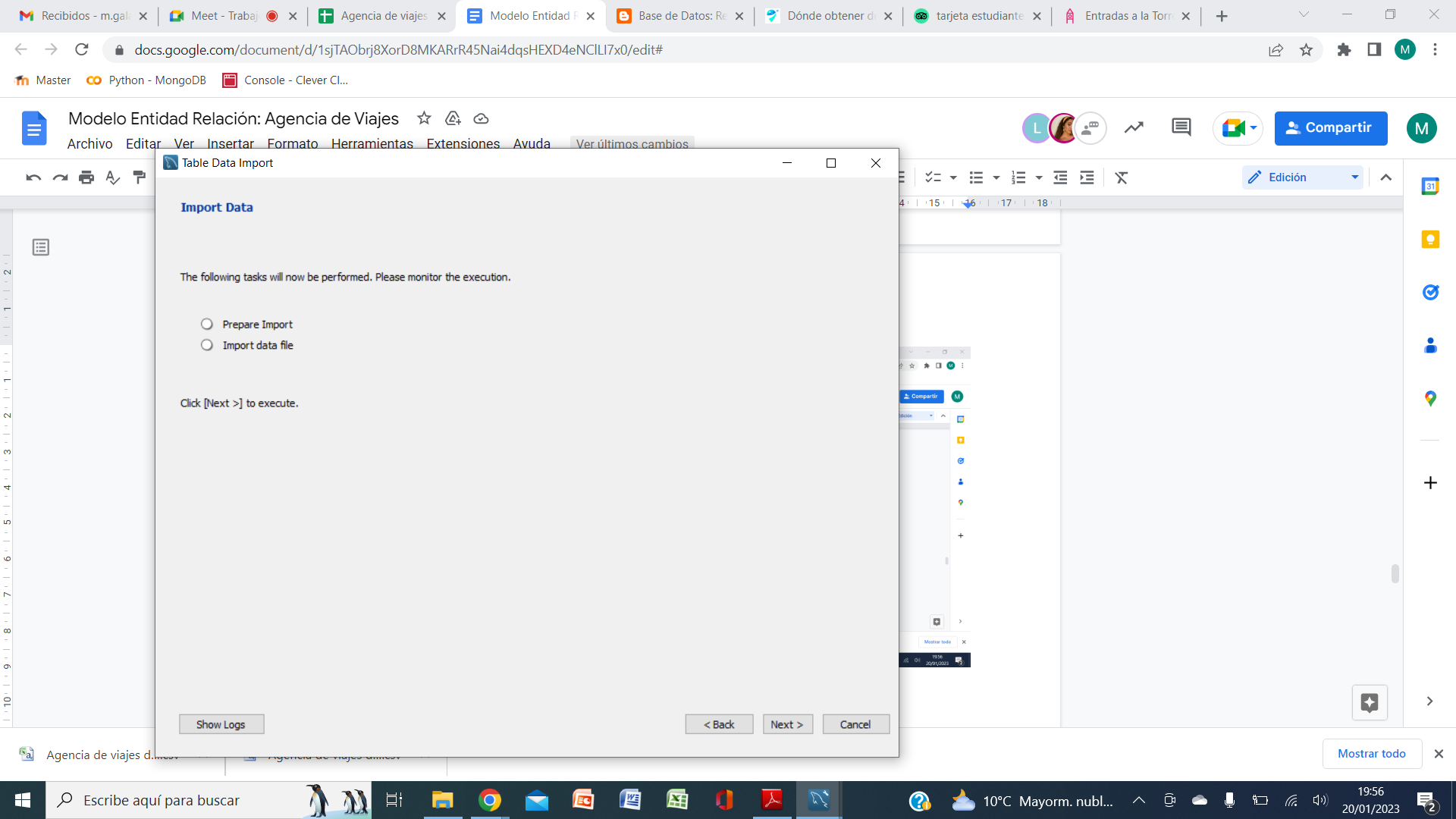
Importación masiva de datos de la tabla de clientes, nos colocamos sobre la tabla de clientes, y con botón derecho, seleccionamos la opción ‘Table Data ImportWizard’. A continuación de muestran los pasos recorridos:

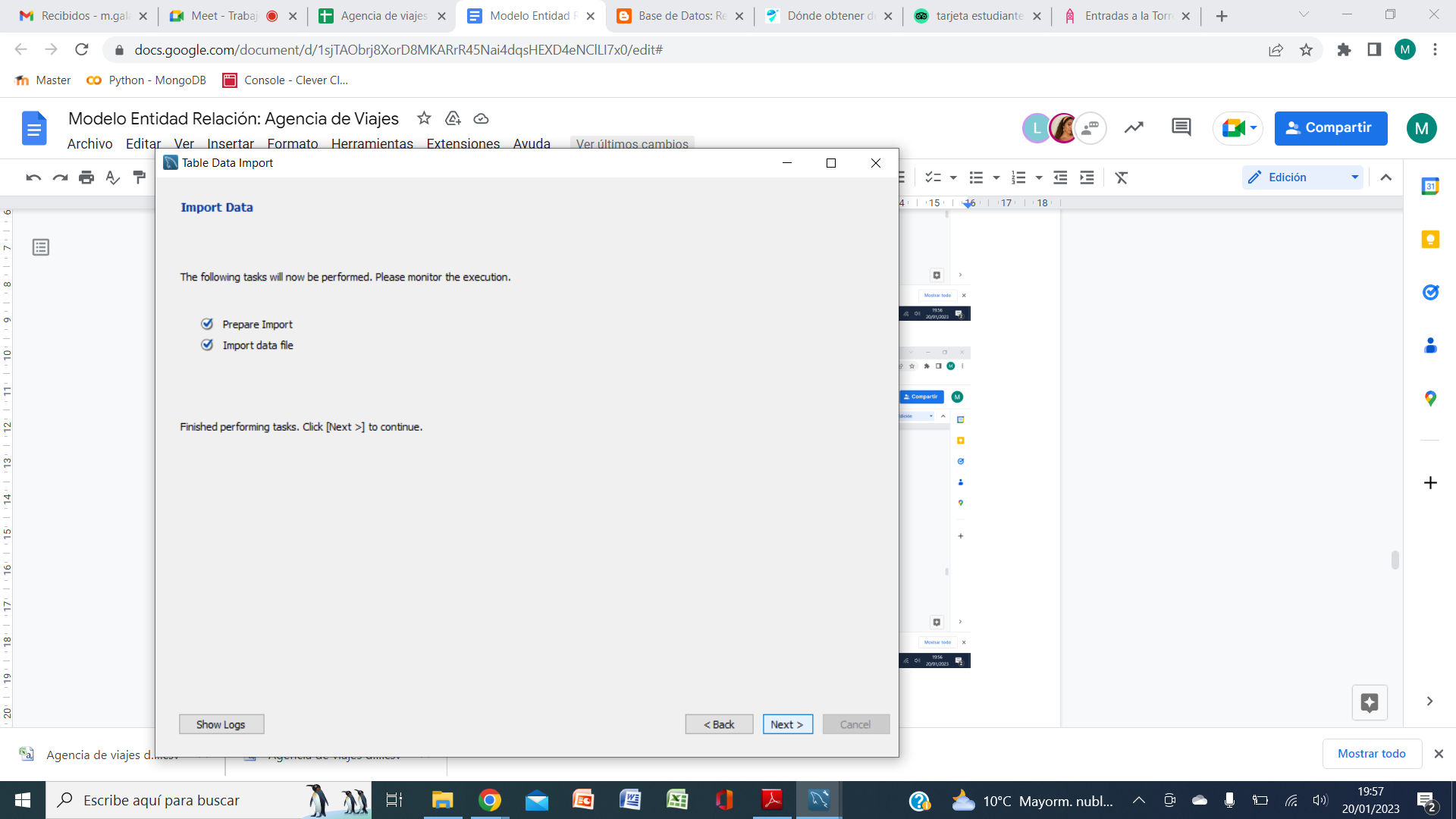




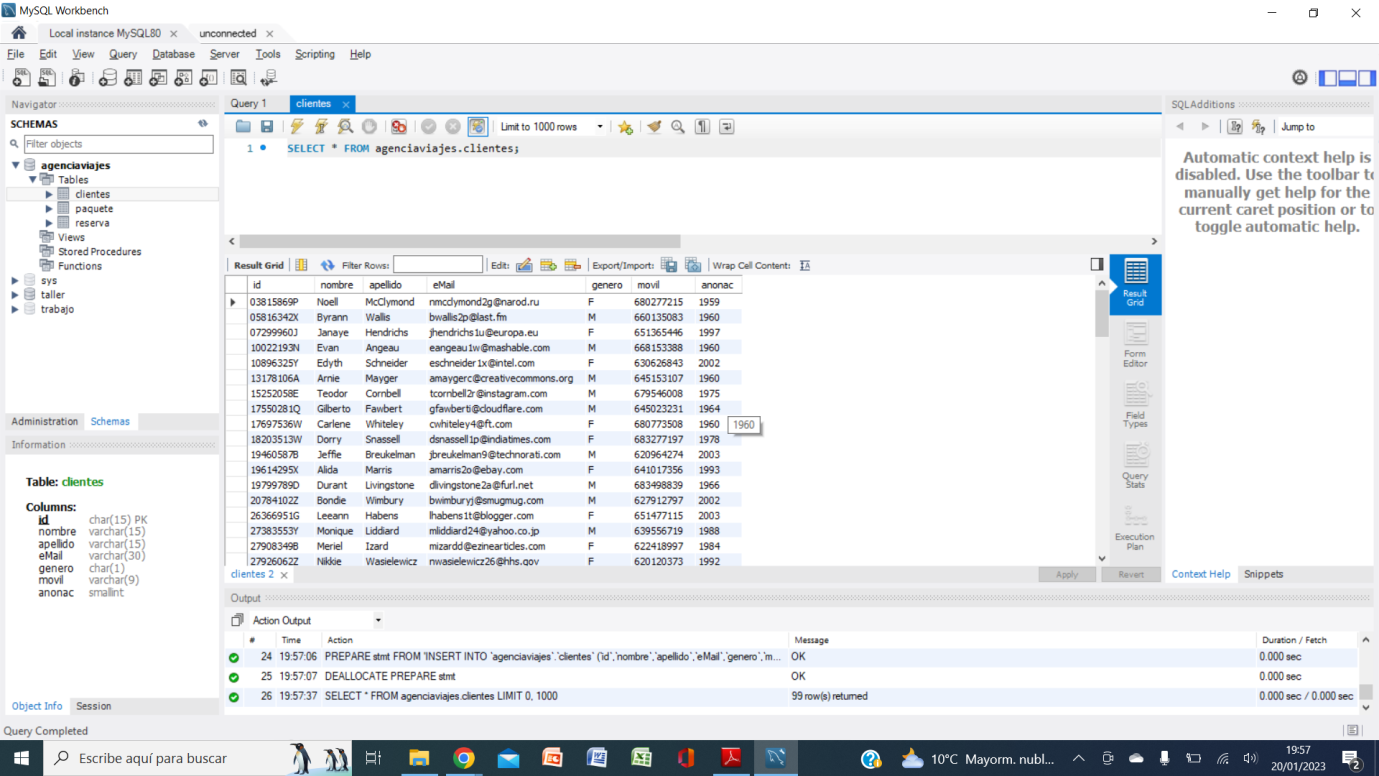








Comprobamos que se han cargado correctamente los datos:



Seguimos estos mismos pasos para cargar los datos de las tablas paquete y reserva.

#### 

# 5.- Consultas

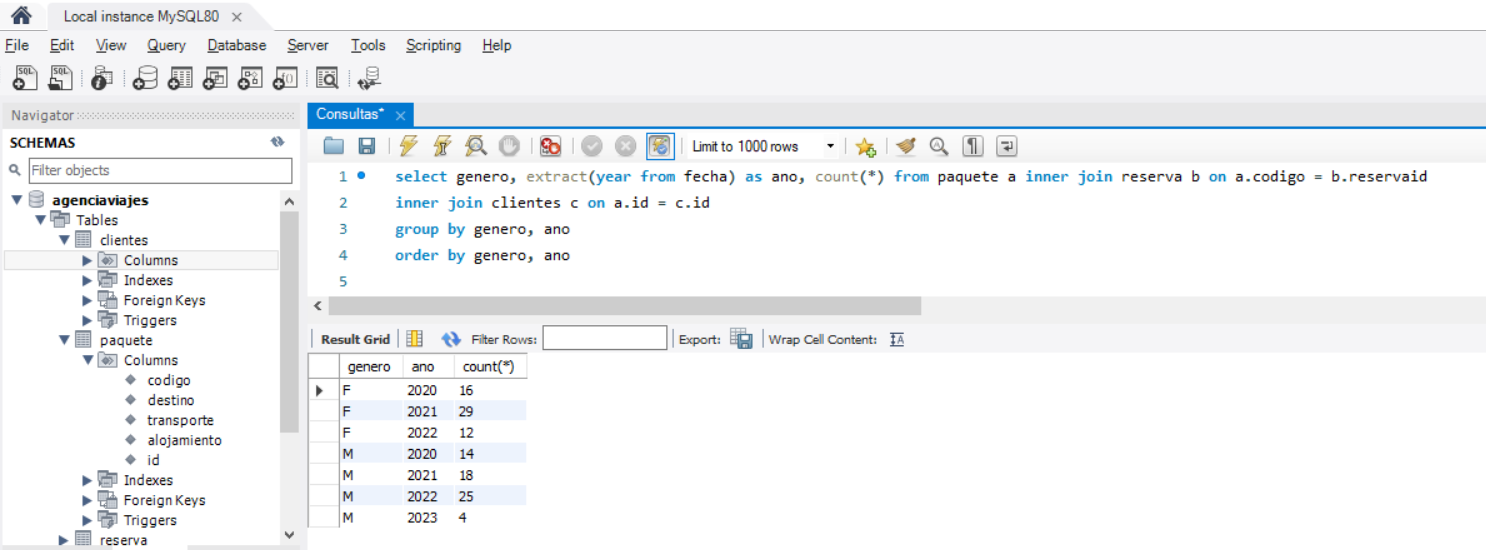
Planteamos las siguientes consultas, para trabajar con nuestra base de datos:

1. ¿Cuántos paquetes se han comprado por sexo y año de reserva?
2. Enumerar los diferentes destinos elegidos cada año
3. ¿Cuántos clientes femeninos entre 18 y 25 años, tomaron como medio de transporte el avión?
4. ¿Cuál fue el medio de transporte más utilizado por los clientes nacidos en la década de los 60?
5. ¿Cuántos viajes se realizaron en el 2021 con un precio inferior a 1000 y que el código de reserva esté entre el MAD00010 y el MAD00050?
6. Consultar las personas que viaja en grupo de 3 antes del 2022 que pagaron más de 2000 y ordenar el precio de formas descendente

A continuación, pasamos a desarrollar cada consulta y su resultado:

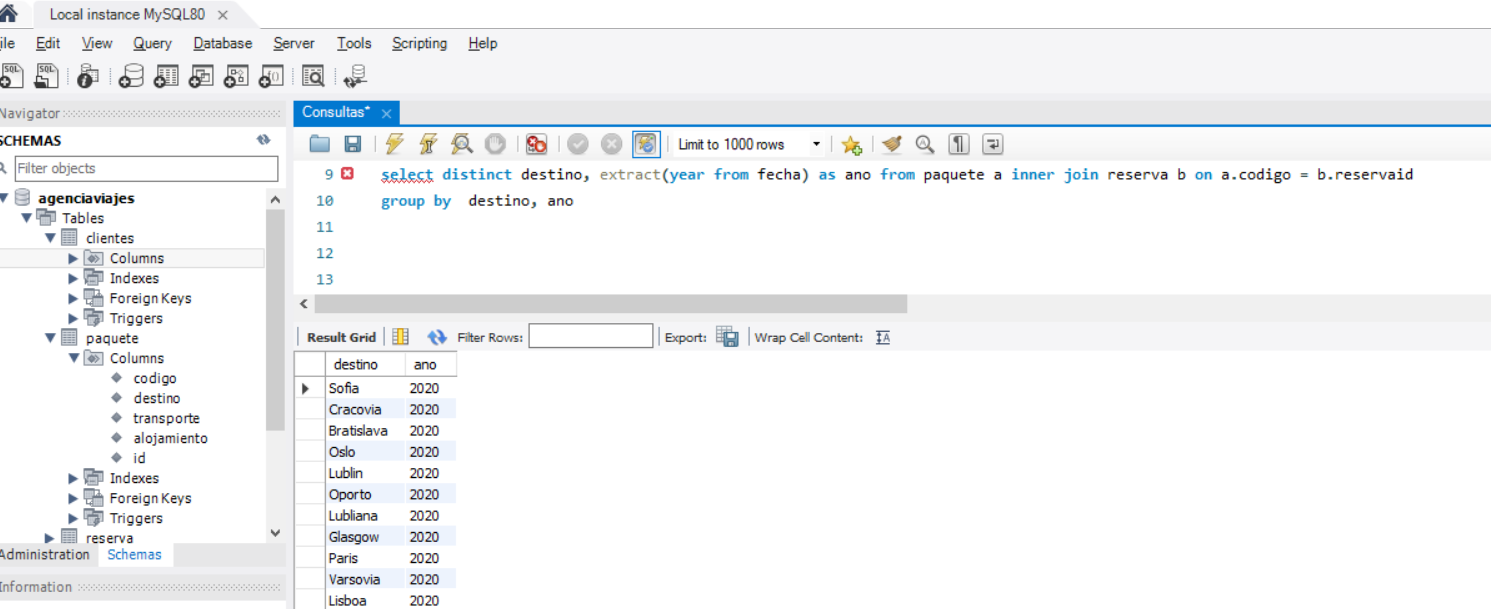
**¿Cuantos paquetes se han comprado por sexo y año de reserva?**

selectgenero, extract(yearfrom fecha) as ano, count(\*) from paquete a innerjoin reserva b ona.codigo = b.reservaidinnerjoin clientes c ona.id = c.idgroupbygenero, ano orderbygenero, ano



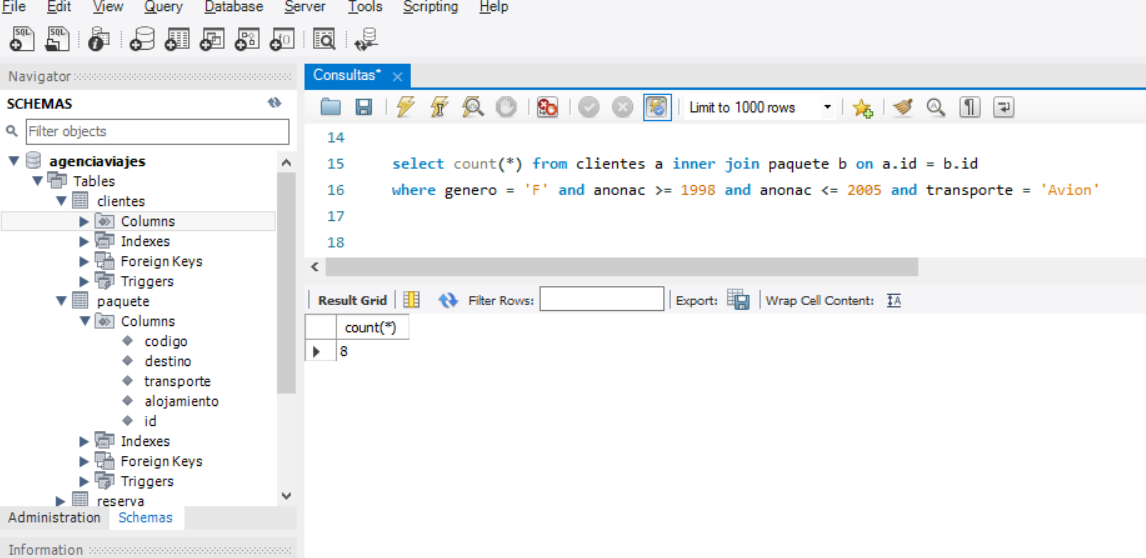
**Enumerar los diferentes destinos elegidos cada año**

selectdistinct destino, extract(yearfrom fecha) as ano from paquete a innerjoin reserva b ona.codigo = b.reservaidgroupby destino, ano



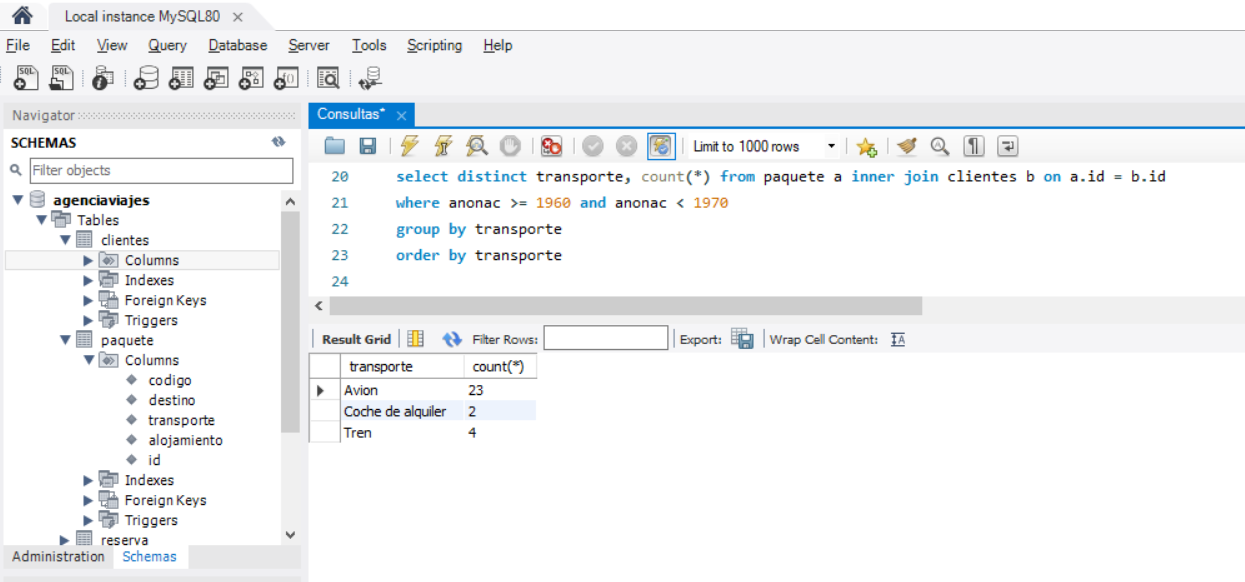
**¿Cuántos clientes femeninos entre 18 y 25 años, tomaron como medio de transporte el avión?**

selectcount(\*) from clientes a innerjoin paquete b ona.id = b.idwhere genero = 'F' and anonac>= 1998 and anonac<= 2005 and transporte = 'Avion'

****

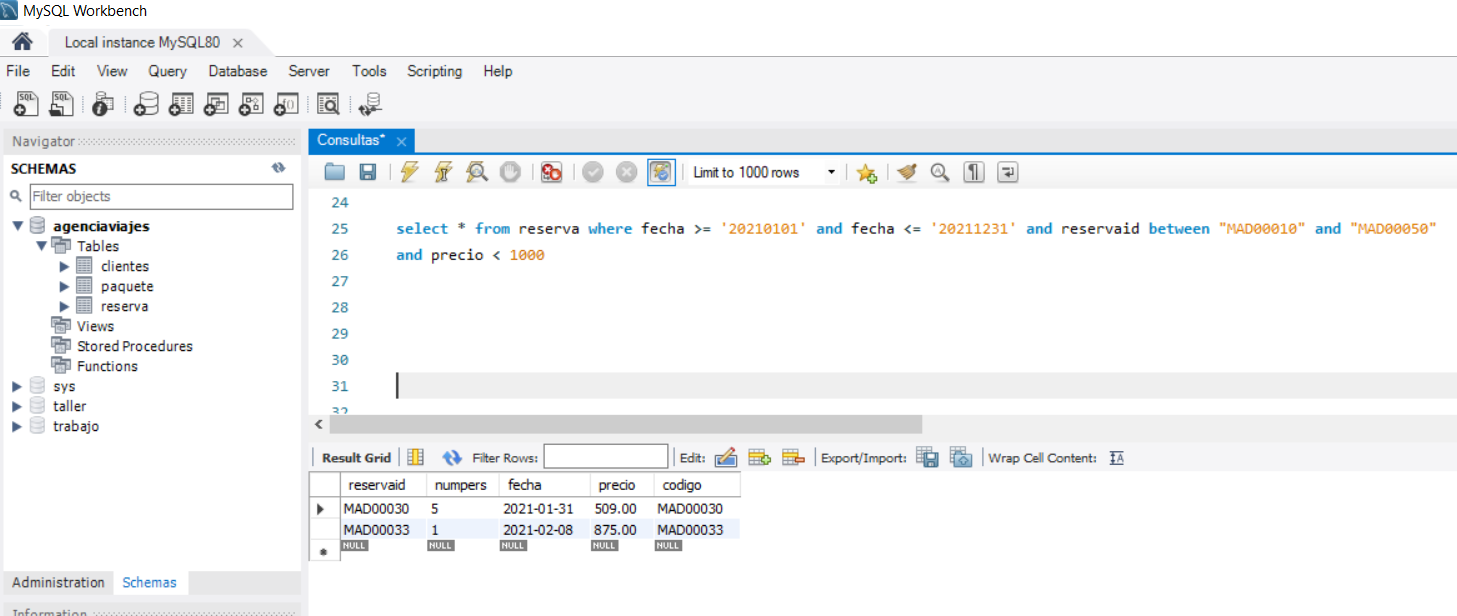
**¿Cuál fue el medio de transporte más utilizado por los clientes nacidos en la década de los 60?**

selectdistinct transporte, count(\*) from paquete a innerjoin clientes b ona.id = b.idwhereanonac>= 1960 and anonac< 1970 groupby transporte orderby transporte

****

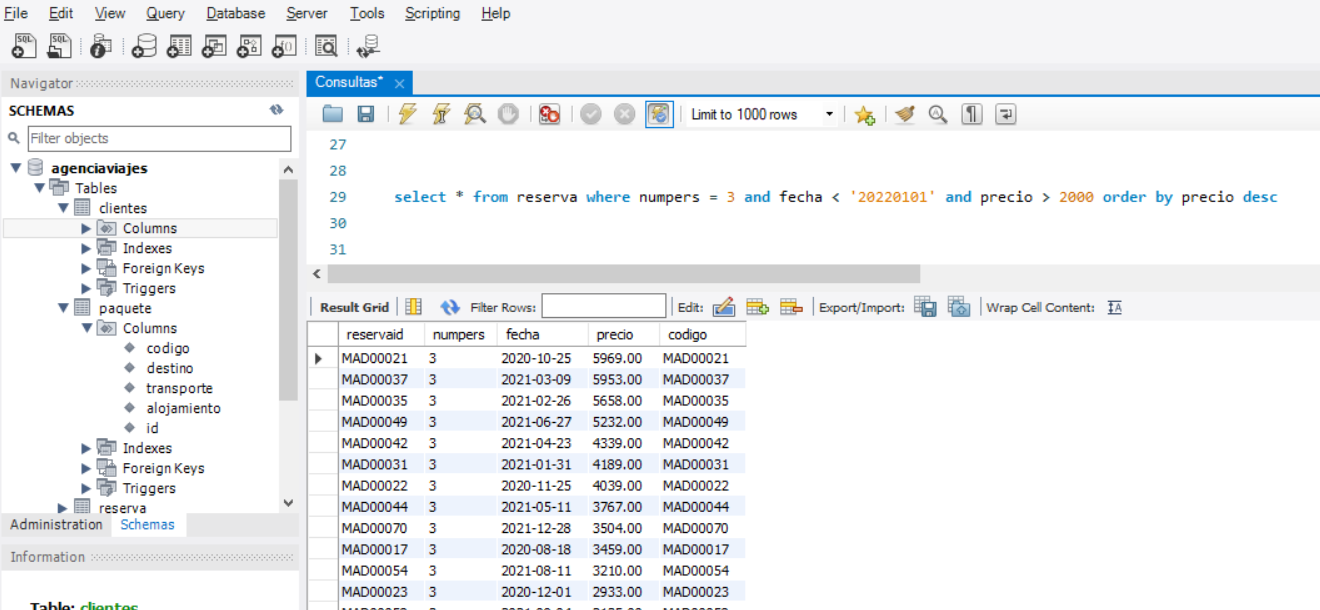
**¿Cuántos viajes se realizaron en el 2021 con un precio inferior a 1000 y que el código de reserva esté entre el MAD00010 y el MAD00050?**

select \* from reserva where fecha >= '20210101' and fecha <= '20211231' and reservaidbetween "MAD00010" and "MAD00050" and precio < 1000

****

**Consultar las personas que viaja en grupo de 3 antes del 2022 que pagaron más de 2000 y ordenar el precio de formas descendente**

select \* from reserva wherenumpers = 3 and fecha < '20220101' and precio > 2000 orderby precio desc



# 6.- Hemos realizado un modelo relacional. Sin embargo, ¿podrían implementar lo anterior en otro tipo de base de datos?

Sí, es posible implementar un modelo relacional en otro tipo de base de datos. Por ejemplo, se podría implementar en una base de datos NoSQL, como MongoDB o Cassandra, utilizando un esquema de documentos o columnas, respectivamente. También se podría implementar en una base de datos de grafos, como Neo4j, utilizando nodos y relaciones para representar las entidades y relaciones del modelo relacional, el modelo relacional se adapta mejor a las bases de datos relacionales y puede requerir una mayor cantidad de esfuerzo para adaptarlo a una base de datos NoSQL. Sin embargo, es importante tener en cuenta que cada tipo de base de datos tiene sus propias ventajas y desventajas, por lo que es importante evaluar cuál es la mejor opción para una aplicación específica antes de tomar una decisión.

#### 

# 7.- Analicen las ventajas y las desventajas de utilizar una base de datos no relacional.

Partiendo del material proporcionado en esta asignatura, se presentan a continuación de manera clara las diferentes ventajas y desventajas de emplear una base de datos no relacional.

* Ventajas

-Volumen de datos. Hace referencia a que son capaces de manejar datos desmesurados gracias a la utilización de una estructura distribuida. Además, añadir, que permite guardar datos de cualquier tipo y nos permitirá la distribución de datos dentro de un clúster.

-Su versatilidad a la hora de realizar crecimientos o cambios en el método de almacenaje de la información, ya que nos permite realizar estos cambios sin tener que realizar configuraciones extra. Se realizan las consultas utilizando JSON.

-Escalabilidad horizontal. Nos referimos aquí, a que soportan estructuras distribuidas, haciendo que sea posible el crecimiento horizontal, es decir, que si es necesaria la instalación de nuevos nodos operativos, así se hará, permitiendo que se balancee la carga de trabajo. Esto permitirá una expansión mucho más sencilla. Se basan en una arquitectura de tipo ‘shared-nothing’.

-Disponibilidad de recursos. No necesitan de muchas cantidades de recursos, lo que permite que estos vayan creciendo a medida que son necesarios.

-Optimización. NoSQL emplea un algoritmo interno para reescribir consultas que hayan sido escritas por los usuarios o aplicaciones, para así conseguir tener un nivel óptimo en todas las operaciones que tenga que realizar.

-También mencionar las principales ventajas que nos ofrece trabajar sin esquema, donde distinguimos la flexibilidad y el tratamiento de los datos semiestructurados:

Flexibilidad. Muchas veces nos va a resultar más útil guardar nuestros datos en bruto, que, teniendo un esquema, ya que esto nos determinará la necesidad de conocer a priori que datos almacenaremos. Si tenemos un esquema flexible, menos tendremos que trabajar. Si trabajamos en entornos ágiles, nuestro esquema necesitará cambiar constantemente en función de nuestras demandas.

Tratamiento de datos semiestructurados. Será mucho más sencillo tratar datos semiestructurados o desestructurados.

* Desventajas

-Atomicidad. Si no cuenta con esta característica, no vamos a encontrar una información consistente.

-Documentación del Software. Al tratarse de una BBDD bastante nueva, encontramos que hay operaciones que se encuentran limitadas, por lo que a veces se requerirán de conocimientos muy avanzados respecto de las herramientas y de las personas que realizan el desarrollo de estos softwares.

-Estándares en el lenguaje. No encontramos uno predeterminado para todas las soluciones posibles de NoSQL.

-Herramientas GUI (GraphicalUser Interface). No vamos a tener disponible una interfaz gráfica, sino que cuentan con acceso por consola, por lo que se necesitarán amplios conocimientos de las instrucciones para hacer posible su mantenimiento.

-De igual manera, que antes mencionamos las ventajas que nos ofrecía no utilizar un esquema, también encontramos alguna desventaja debido a la no utilización:

Ausencia de metadatos, y en muchos casos se requiere de su consulta para herramientas de explotación.

Esquema implícito. Se deberá crear un esquema a partir del programa desde el que accedemos, el cual quedará únicamente determinado dentro de esta aplicación; sin embargo, con esto surgen otros problemas: que se acceda a la misma base de datos desde diferentes aplicaciones (generando diferentes esquemas) y que nuestra información sufrirá cambios que se verá repercutido en los programas.

# Conclusión

Podemos concluir que los SGBD son una herramienta útil para cualquier negocio que podamos tener, en el cuál podemos llevar a cabo todo tipo de consulta de forma prácticamente automatizada cuándo tenemos que realizarlas constantemente, podemos indicar nuestras propias reglas para que de una manera más óptima logremos enlazar nuestras distintas tablas y gestionarlas también a través de programas como DBeaver que nos ayuda a la mejora de las importaciones, exportaciones y sintaxis.

En este caso que hemos utilizado MySQL podemos decir que es una herramienta potente y fácil de usar que nos permite almacenar, recuperar y manipular los datos.

**References**

Cartmell, J. (2019, August 30). *GUÍA MODELOS ENTIDAD RELACIÓN Departamento Nacional de Planeación Bogotá, 2019*. ​​​​​​​​​​​​​​Subdirección de Gestión y Desarrollo del Talento Humano. RetrievedJanuary 18, 2023, from https://colaboracion.dnp.gov.co/CDTI/Oficina%20Informatica/Sistemas%20de%20informaci%C3%B3n/Gu%C3%ADas%20Formatos%20Plantillas/Lineamientos%20Modelos%20Entidad%20Relaci%C3%B3n.pdf?

https://learn.microsoft. (2022, December 16). *nchar and nvarchar (Transact-SQL) - SQL Server*. Microsoft Learn. RetrievedJanuary 18, 2023, from https://learn.microsoft.com/en-us/sql/t-sql/data-types/nchar-and-nvarchar-transact-sql?view=sql-server-ver16

Manuel, J. (2018, April 18). *Diferencias entre el modelo Entidad-Relación y Relacional*. PC Solución. RetrievedJanuary 18, 2023, from https://pc-solucion.es/tecnologia/diferencias-entre-el-modelo-entidad-relacion-y-relacional/

Manuel, J. (2018, April 18). *Modelo Relacional*. PC Solución. RetrievedJanuary 18, 2023, from https://pc-solucion.es/terminos/modelo-relacional/

N.D. (n.d.). *Differencebetween VARCHAR and CHAR data type in SQL Server? [Explained]*. Java67. RetrievedJanuary 18, 2023, from https://www.java67.com/2019/06/difference-between-varchar-and-char-data-type-in-sql-server.html

Skov, E. (2009, June 25). *Float vs. Decimal Data Types in SQL Server*. CatapultSystems. RetrievedJanuary 21, 2023, from https://www.catapultsystems.com/blogs/float-vs-decimal-data-types-in-sql-server/