next EDUCACIÓN

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

Mertxe Galán Espiga

Valesca Bravo

Table of Contents

Introduction

In this paper, we will see how to develop a database, from modeling to its implementation in MySQL using SQL.

We have chosen the tourism industry, specifically a travel agency. We began by creating an entity-relationship model in the Dia program, where we defined the rules for our business model.

To convert the entity-relationship model to a relational model, we explored some online modeling alternatives, such as Vertabelo, which offers export options for different database engines and is free for students.

Once both models were designed, we found a website (mockaroo) that helped us generate a database with the previously created attributes. In addition to using this tool, we also generated random data using rules in Excel.

Finally, we used SQL to create the tables and insert the information into MySQL. We tested the queries using MySQL and RStudio (attached as appendices).

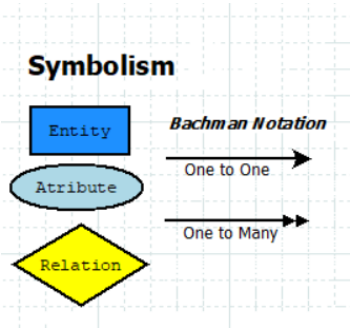
1.- Entity-Relationship Model

We used the "Dia" program to diagram the Entity-Relationship model using Bachman notation.

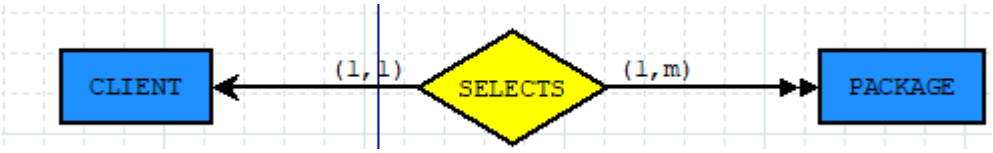
To create our Travel Agency, we started by defining the Entities and their attributes:

- **Client:** We defined its attributes as Name, Surname, Gender, Email, Mobile, Year of Birth, and finally Client ID, which would serve as the Primary Key.
- **Package:** It would have its own identifying code as the Primary Key, along with other attributes like Destination, Transportation, and Accommodation.
- **Reservation:** Like the Package Entity, it would have a Primary Key attribute to identify it, along with Price, Date, and Number of People.

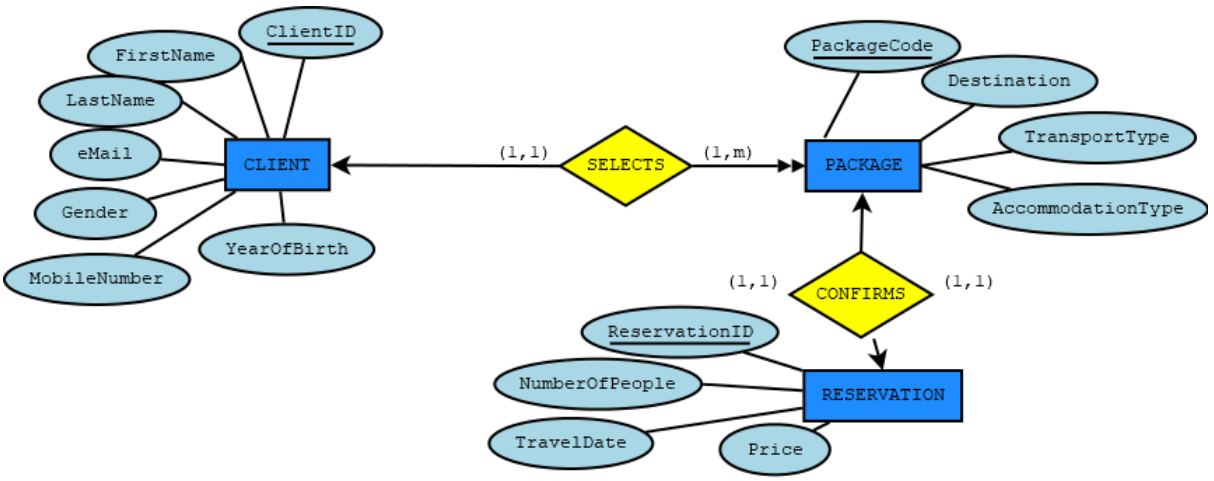
Rules and Conditions:



We defined that the Client Entity would be related to the Package Entity, where a client could purchase many packages, but each package would be associated with only one client.



The Package Entity and the Reservation Entity would be related to each other with a one-to-one cardinality.



2.- Relational Model

To create the Relational Model, we decided to use 'Vertabelo', an online database modeler that uses Information Engineering Notation (Cartmell, 2019) to represent a relationship, allowing each attribute to be displayed with its data type.

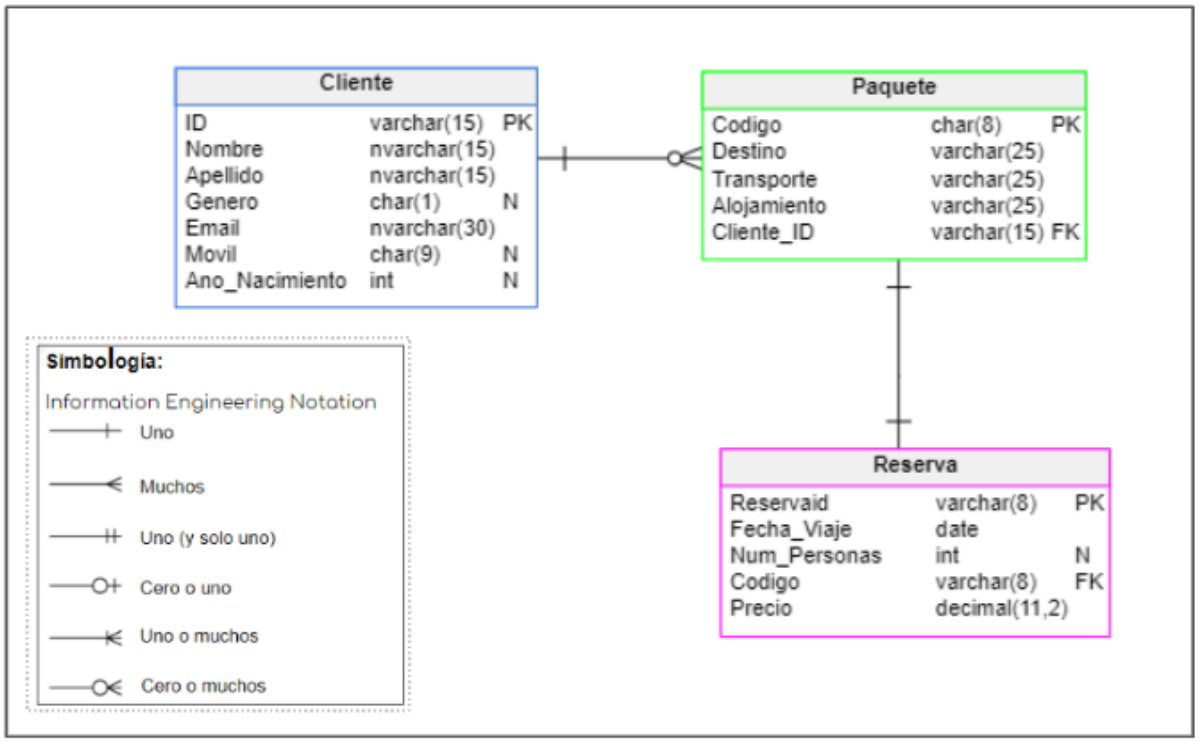
Rules and Conditions:

- **Client:**
    1. ID(PK), ID: We decided to use the passport or ID of each person. While passports generally consist of 9 digits, we set the field to have 15 spaces to accommodate people who might want to use their national ID
    2. Name, Surname, and Email: We used "nvarchar" because we believe it's important to take advantage of its Unicode character property, as this is a travel agency with an international outlook, even though this data type takes twice as much storage space (2 bytes) as "varchar" (https://learn.microsoft, 2022).
    3. Gender, Mobile, and Year of Birth, are nullable variables,—they can be filled

in or left blank by the client. Gender is a "char" that accepts only one character, Mobile is a "varchar" with nine spaces, and Year of Birth, which we decided on, is a "SmallInt." However, in Vertabelo, it's shown as "Int" because the platform doesn't offer a "SmallInt" option.

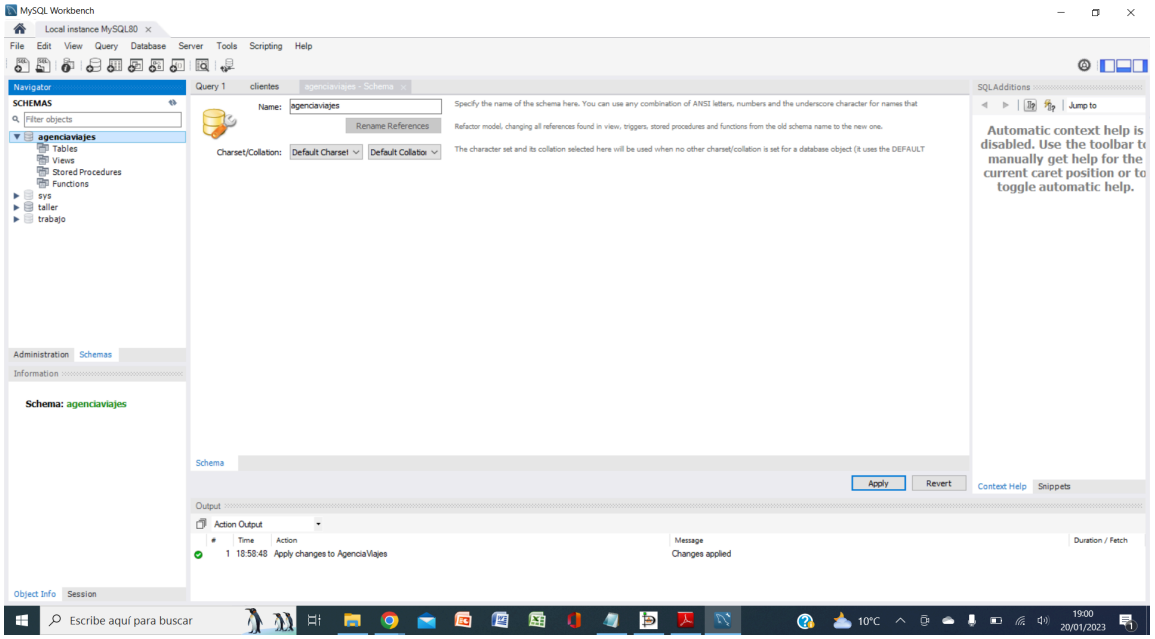
- **Package:**
    1. ID_Package (PK) doesn't require more than an "char" with 8 spaces.
    2. Destination, TransportationType, and AccommodationType were created as "nvarchar" with a capacity of 25 spaces.

- **Reserva:**
    1. Reservaid(PK), "Varchar" with 8 spaces
    2. TravelDate, Assigned as a "Date" type.
    3. NumberofPeople, "Int" with 6 spaces, which will be nullable.
    4. Price: We used "Decimal" instead of "Float" because the former is more precise, and we're dealing with money. (Skov, 2009)


Cliente
ID varchar(15) PK
Nombre nvarchar(15)
Apellido nvarchar(15)
Genero char(1) N
Email nvarchar(30)
Movil char(9) N
Ano_Nacimiento int N
Paquete
Codigo char(8) PK
Destino varchar(25)
Transporte varchar(25)
Alojamiento varchar(25)
Cliente_ID varchar(15) FK
Reserva
Reservaid varchar(8) PK
Fecha_Viaje date
Num_Personas int N
Codigo varchar(8) FK
Precio decimal(11,2)
Simbología:
Information Engineering Notation
Uno
Muchos
Uno (y solo uno)
Cero o uno
Uno o muchos
Cero o muchos

3.- SQL Sequences in a DBMS - Referential Integrity Rules and/or Business Rules

In this section, we discuss the implementation of our tables using the MySQL database management system, as well as the description of our various integrity rules. First, we'll create our database, which we'll call agenciaviajes:



We add three tables in which we will load our data, which we describe as follows:

CLIENTE

Column Name	Meaning	Data Type	NULLS allowed
ClientID	Identity document	CHAR(15)	N
FirstName	First Name	NVARCHAR (15)	N
LastName	Last Name	NVARCHAR (15)	N
Email	email adress	NVARCHAR (30)	N
Gender	Gender (M male, F female)	CHAR (1)	Y
MobileNumber	Mobile Number	VARCHAR (9)	Y
YearOfBirth	Year Of Birth	SMALLINT	Y

PAQUETE

Column Name	Meaning	Data Type	NULLS allowed

PackageCode	Package Code	CHAR (8)	N
Destination	Destination	VARCHAR (25)	N
TransportType	Transport Type	VARCHAR (25)	Y
AccommodationType	Accommodation Type	VARCHAR (25)	Y
ClientID	Client ID	VARCHAR (15)	N

RESERVA

Column Name	**Meaning**	**Data Type**	**NULLS allowed**
Reservaid	Reserva Id	CHAR (8)	N
NumberOfPeople	Number Of People	SMALLINT	Y
TravelDate	Travel Date	DATE	N
Price	price	DECIMAL (11,2)	N
ClientID	Identity document	NVARCHAR (15)	N

- **Primary Key Integrity Rule**

The primary key integrity rule is a constraint in a database that ensures each row in a table has a unique and non-null value in the column or set of columns forming the primary key. This means that no row can have an empty or duplicate value in the primary key column.
The primary key is a unique identifier for each row in a table and is used to relate tables to each other.
For example, in our database, we have 3 tables with a column that will be the primary key: in the Clients table, we have "ID", in the Package table, we have "Code", and in the Reservation table, we have "ReservationID". The primary key integrity rules apply automatically when creating a table with a specified primary key, ensuring that the data in the table is accurate.

- **Foreign Key Integrity Rule**

The foreign key is used to establish a relationship between two tables. For example, in our Clients table, we have a primary key called "ID", and in our Package table, we have a foreign key "ID", establishing a relationship between each package and a specific client.

- Business Rules

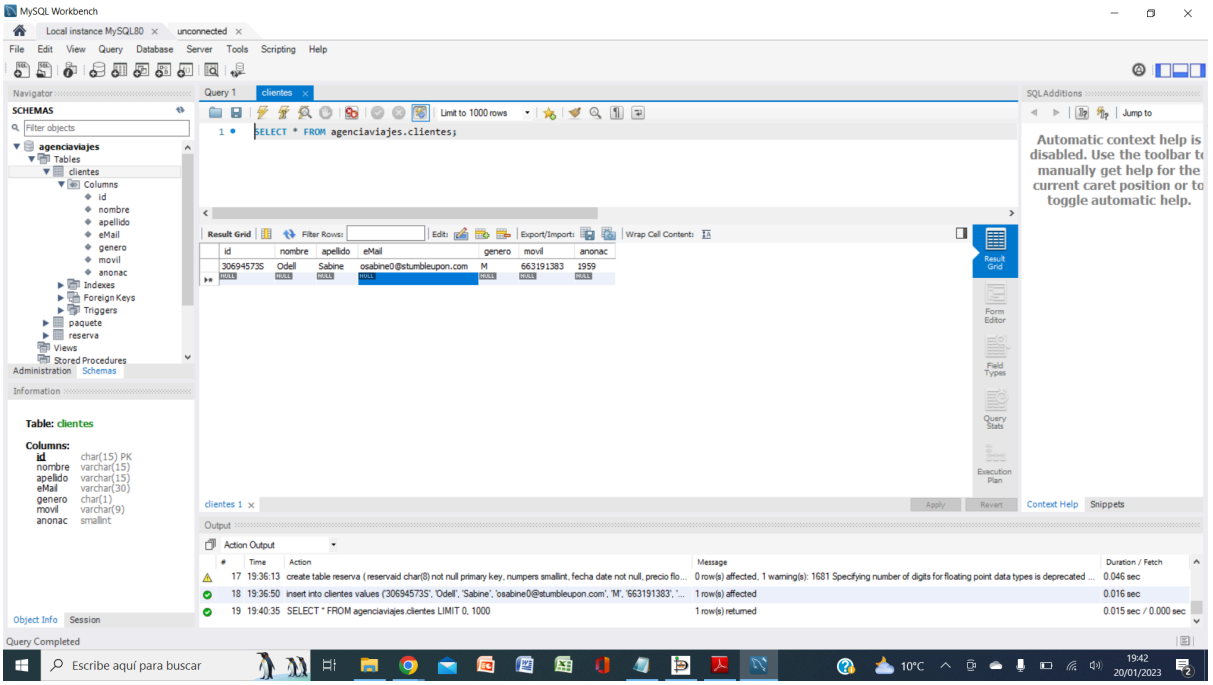
Business rules define the guidelines that determine an organization's business activities, but within our entity-relationship diagram, these rules will set our attributes, relationships, and constraints on our data. For example, one of our business rules is that our clients can have more than one package associated with them, as long as each package has an associated reservation. Another example is that all our clients must have Madrid as their origin, regardless of their destination.

4.- Data Loading

We will manually insert a record into the Clients table with the following SQL statement:

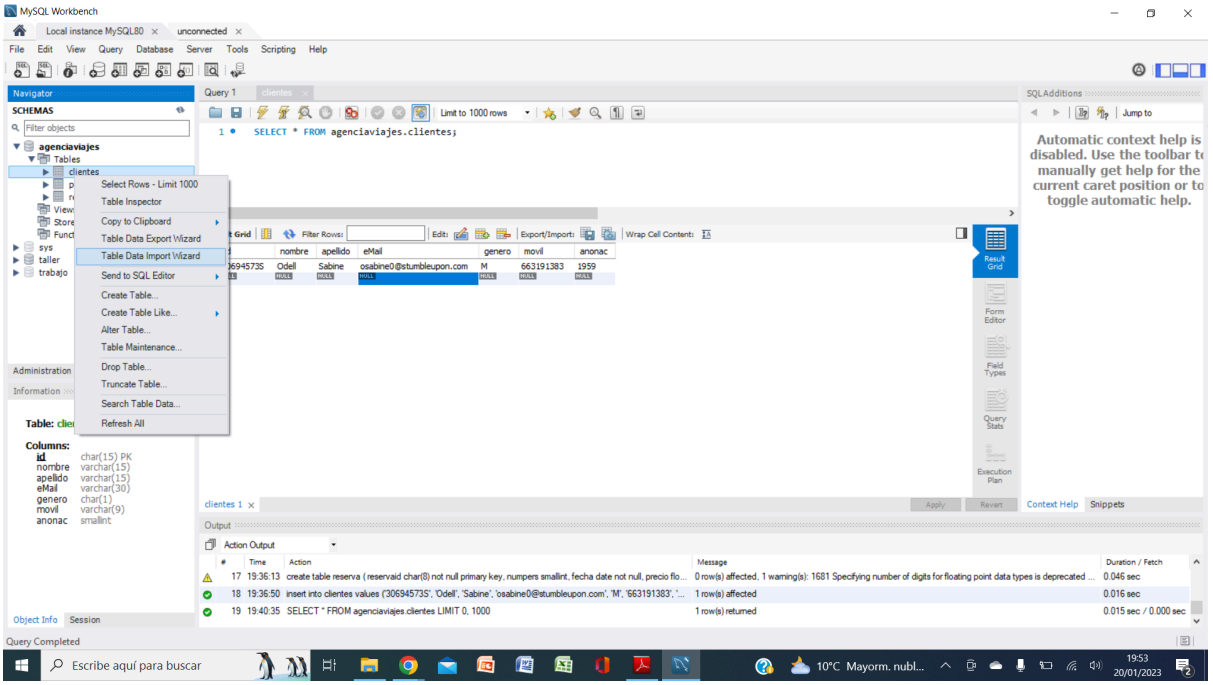
insertinto clientes values ('30694573S', 'Odell', 'Sabine', 'osabine0@stumbleupon.com', 'M', '663191383', '1959')

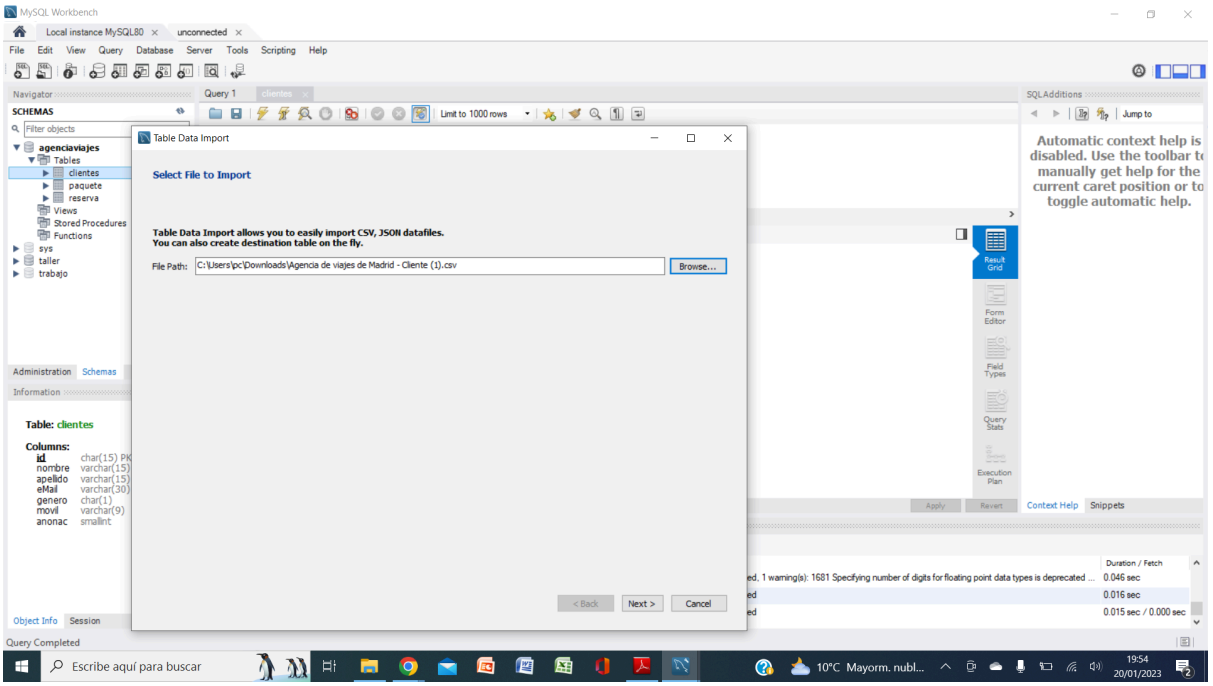
After executing this statement, we query the Clients table and get this record::

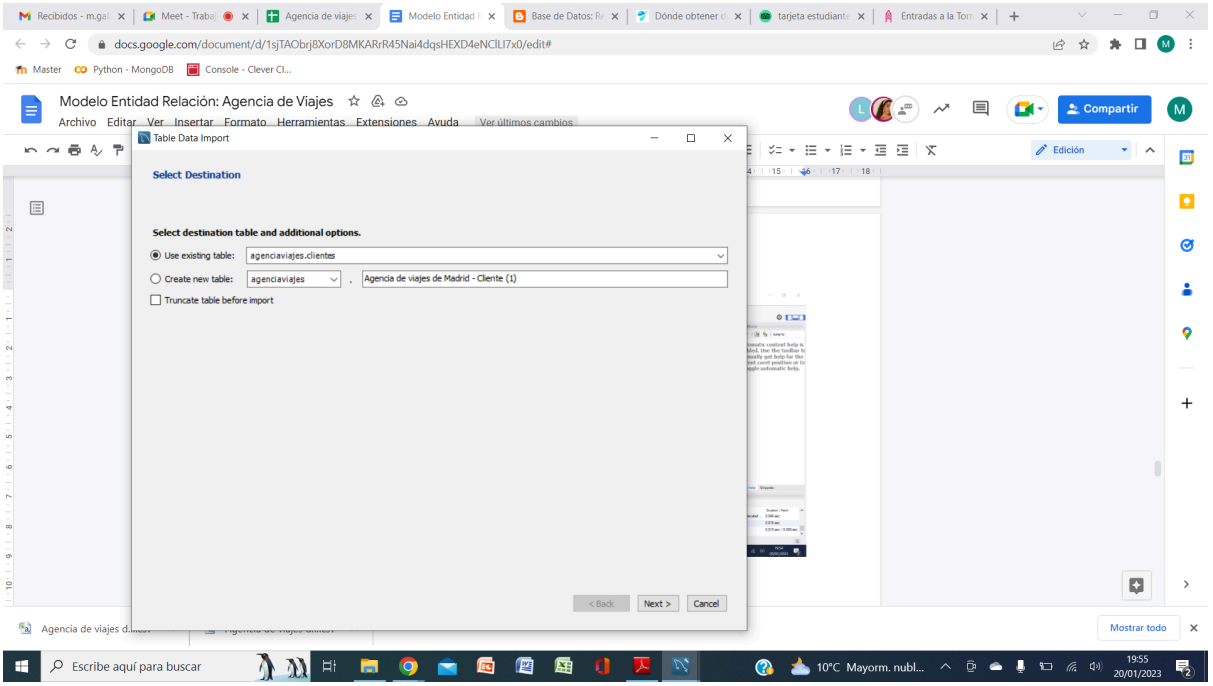


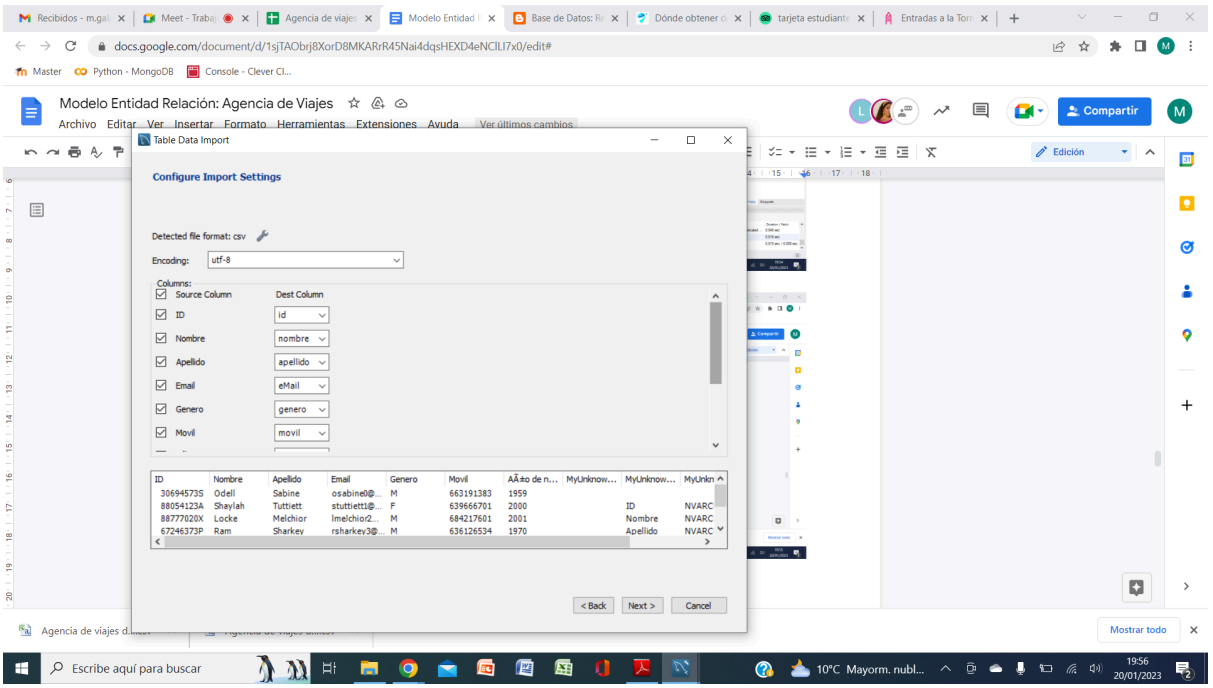
To optimize data loading, and knowing how to do it manually, for the rest of the data, we will use bulk imports through CSV file imports. We start with an Excel document in Google Drive with 3 sheets (one for Clients, one for Packages, and one for Reservations), where group members entered data, and we exported each sheet to a CSV file.

Bulk Import of Data into the Clients Table To import data into the Clients table, we select the table, right-click, and choose the 'Table Data Import Wizard' option. The following outlines the steps we took:


MySQL Workbench
Local instance MySQL80
unconnected
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
agenciaviajes
Tables
clientes
Select Rows - Limit 1000
Table Inspector
Copy to Clipboard
Table Data Export Wizard
Table Data Import Wizard
Send to SQL Editor
Create Table...
Create Table Like...
Alter Table...
Table Maintenance...
Drop Table...
Truncate Table...
Search Table Data...
Refresh All
Query 1
SELECT * FROM agenciaviajes.clientes;
Limit to 1000 rows
nombre apellido eMail genero movil anonac
Odell Sabine osabine0@stumbleupon.com M 663191383 1959
Result Grid
Form Editor
Field Types
Query Stats
Execution Plan
SQLAdditions
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
17 19:36:13 create table reserva (reservaid char(8) not null primary key, numpers smallint, fecha date not null, precio flo... 0 row(s) affected, 1 warning(s): 1681 Specifying number of digits for floating point data types is deprecated ... 0.046 sec
18 19:36:50 insert into clientes values ('30694573S', 'Odell', 'Sabine', 'osabine0@stumbleupon.com', 'M', '663191383', '... 1 row(s) affected 0.016 sec
19 19:40:35 SELECT * FROM agenciaviajes.clientes LIMIT 0, 1000 1 row(s) returned 0.015 sec / 0.000 sec
Table: clientes
Columns:
id char(15) PK
nombre varchar(15)
apellido varchar(15)
eMail varchar(30)
genero char(1)
movil varchar(9)
anonac smallint
Object Info Session
Query Completed
Escribe aquí para buscar
10°C Mayorm. nubl...
19:53
20/01/2023


MySQL Workbench
Local instance MySQL80
unconnected
Table Data Import
Select File to Import
Table Data Import allows you to easily import CSV, JSON datafiles.
You can also create destination table on the fly.
File Path: C:\Users\pc\Downloads\Agencia de viajes de Madrid - Cliente (1).csv
Browse...
< Back
Next >
Cancel
Table: clientes
Columns:
id char(15) PK
nombre varchar(15)
apellido varchar(15)
eMail varchar(30)
genero char(1)
movil varchar(9)
anonac smallint
Escribe aquí para buscar
10°C Mayorm. nubl...
19:54
20/01/2023


Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda
Compartir
Table Data Import
Select Destination
Select destination table and additional options.
Use existing table: agenciaviajes.clientes
Create new table: agenciaviajes . Agencia de viajes de Madrid - Cliente (1)
Truncate table before import
< Back
Next >
Cancel
Escribe aquí para buscar
10°C Mayorm. nubl...
19:55
20/01/2023


Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda
Compartir
Table Data Import
Configure Import Settings
Detected file format: csv
Encoding: utf-8
Columns:
Source Column
Dest Column
ID
id
Nombre
nombre
Apellido
apellido
Email
eMail
Genero
genero
Movil
movil
< Back
Next >
Cancel
Escribe aquí para buscar
10°C Mayorm. nubl...
19:56
20/01/2023

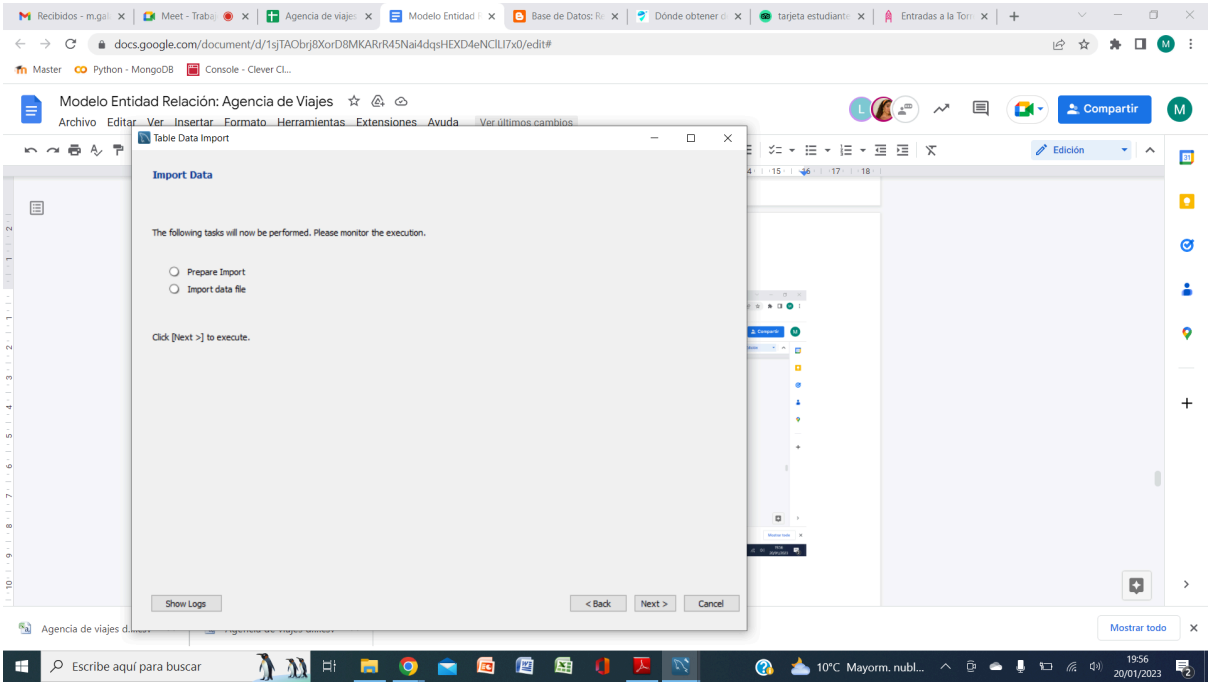


Table Data Import
Import Data
The following tasks will now be performed. Please monitor the execution.
Prepare Import
Import data file
Click [Next >] to execute.
Show Logs
< Back
Next >
Cancel
Modelo Entidad Relación: Agencia de Viajes
Compartir
Escribe aquí para buscar

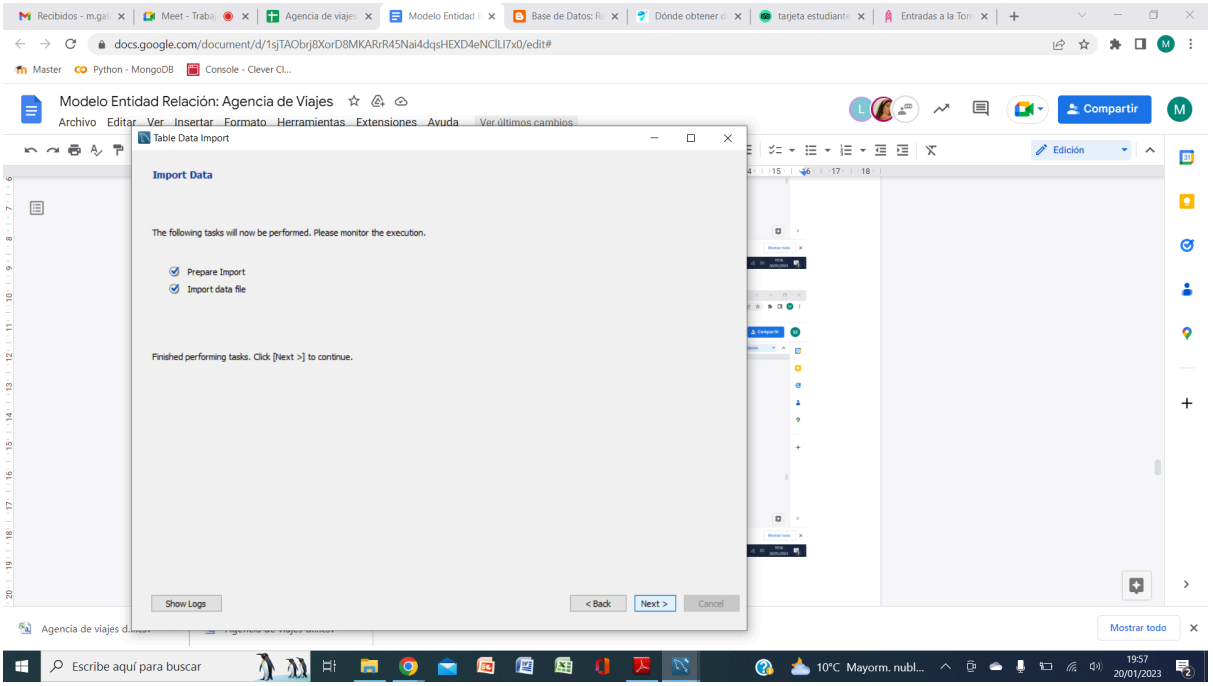
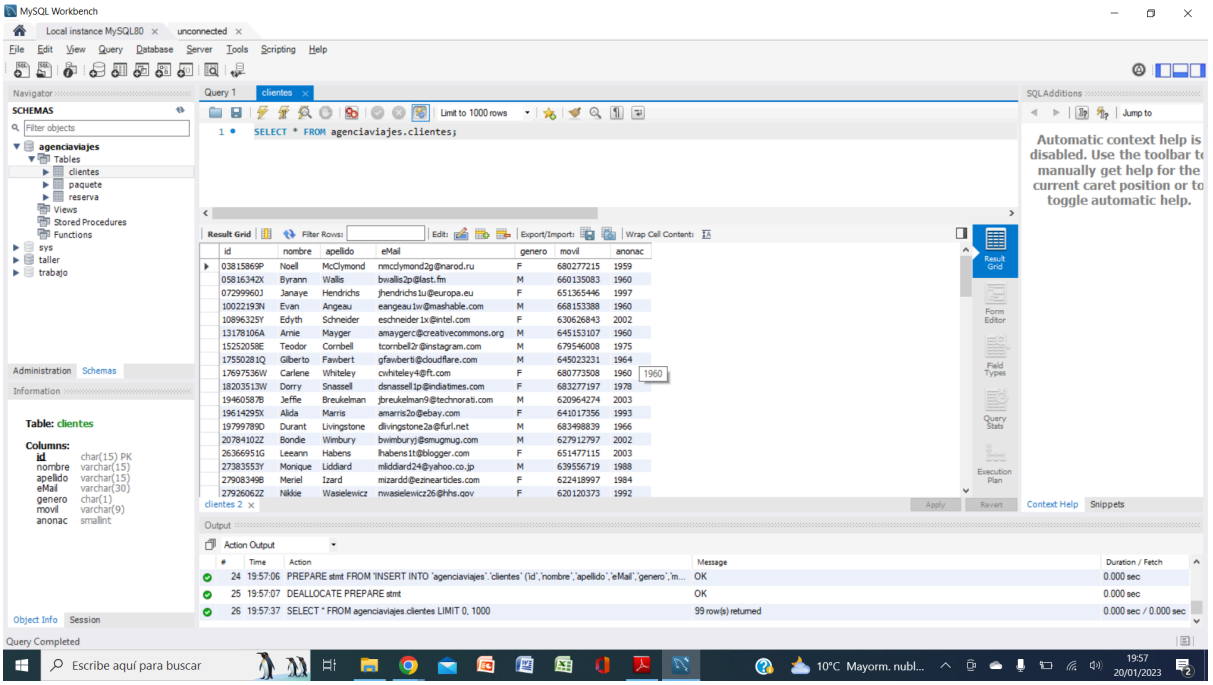


Table Data Import
Import Data
The following tasks will now be performed. Please monitor the execution.
Prepare Import
Import data file
Finished performing tasks. Click [Next >] to continue.
Show Logs
< Back
Next >
Cancel
Modelo Entidad Relación: Agencia de Viajes
Compartir
Escribe aquí para buscar

We verify that the data has been loaded correctly:



We followed these same steps to load the data for the Package and Reservation tables.

5.- Queries

We propose the following queries to work with our database:

1. How many packages have been purchased by gender and reservation year?

2. List the different destinations chosen each year.

3. How many female clients aged 18 to 25 used the airplane as a means of transport?

4. What was the most commonly used means of transport for clients born in the 1960s?

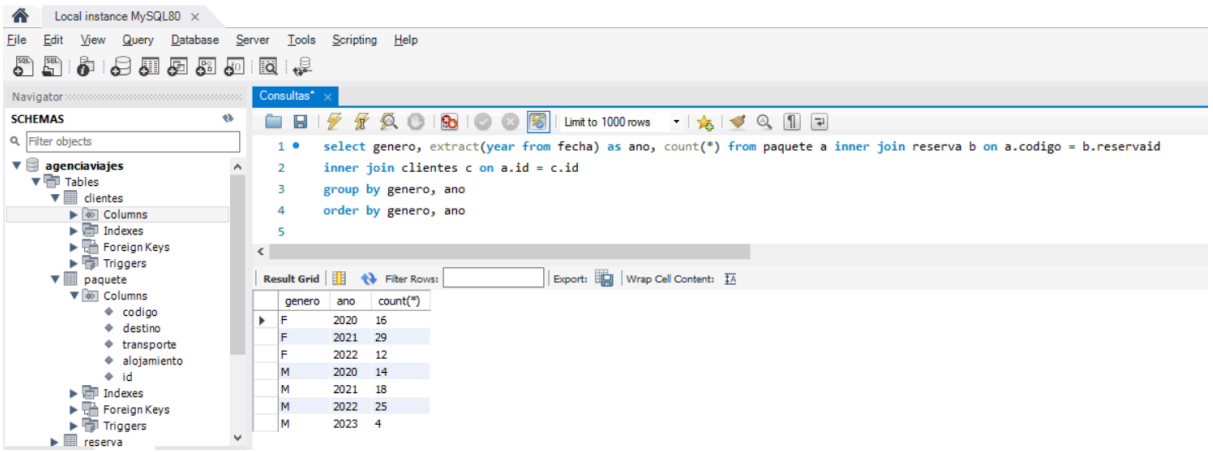
5. How many trips were made in 2021 with a price under 1000 and where the reservation code is between MAD00010 and MAD00050?

6. Find the people who traveled in groups of 3 before 2022, paid more than 2000, and sort the price in descending order

Next, we proceed to develop each query and its result:

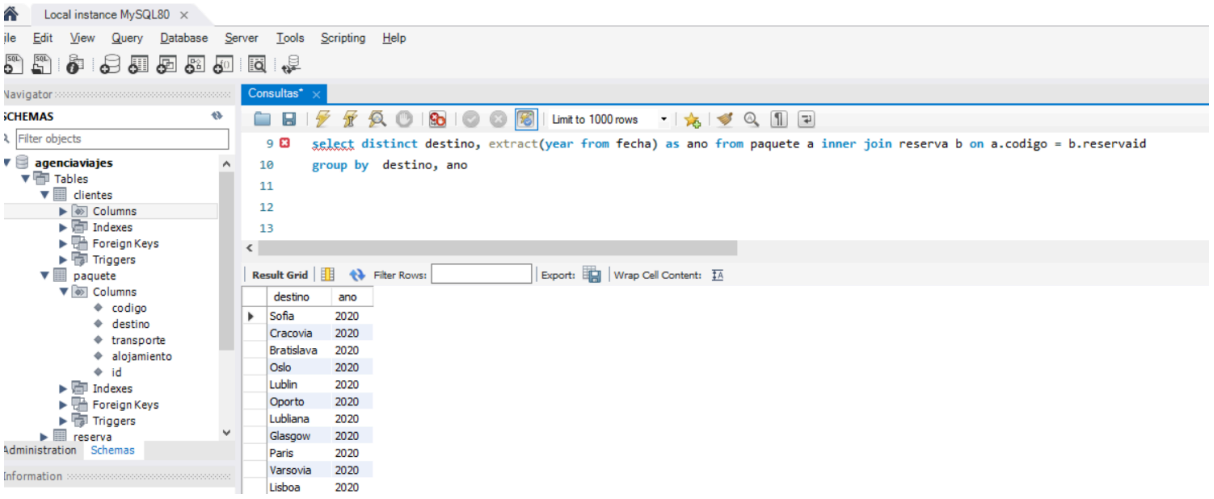
5.I.1.- How many packages have been purchased by gender and reservation year?

select genero, extract(year from fecha) as ano, count(*) from paquete a inner join reserva b on a.codigo = b.reservaid inner join clientes c on a.id = c.id group by genero, ano order by genero, ano



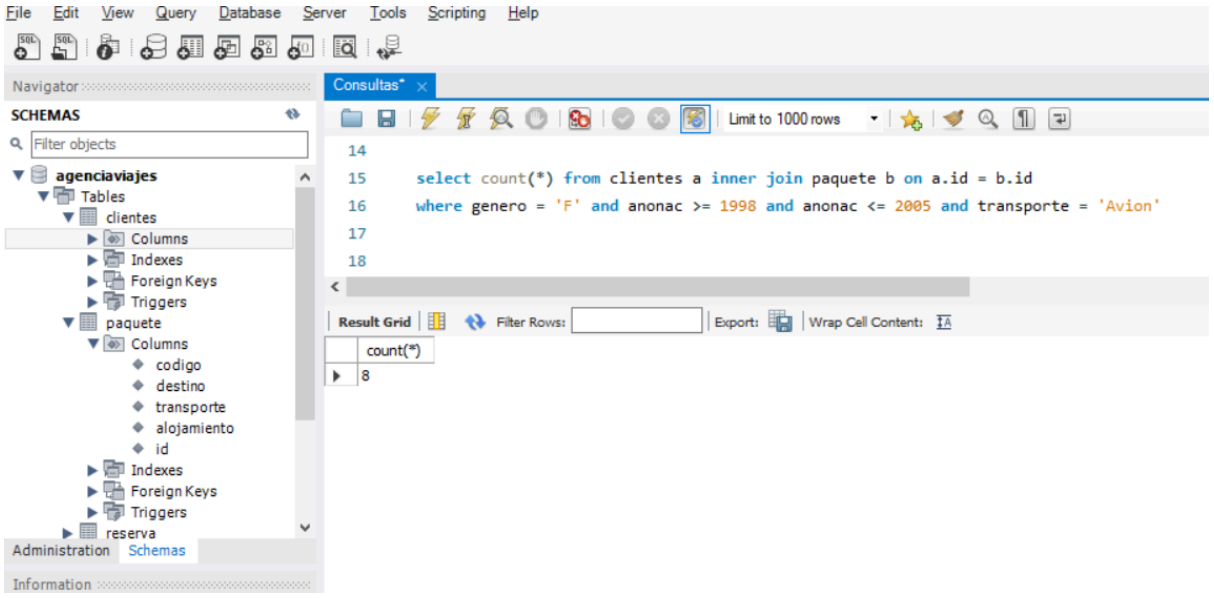
5.I.2.-List the different destinations chosen each year.

select distinct destino, extract(year from fecha) as ano from paquete a inner join reserva b on a.codigo = b.reservaid group by destino, ano



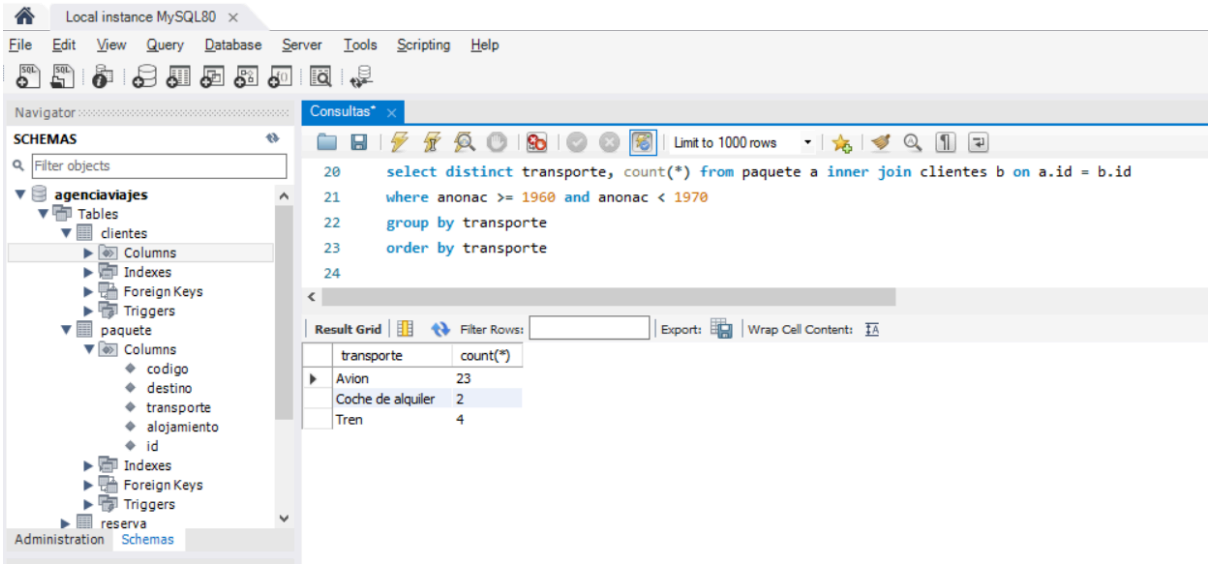
5.I.3.-How many female clients aged 18 to 25 used the airplane as a means of transport?

select count(*) from clientes a inner join paquete b on a.id = b.id where genero = 'F' and anonac >= 1998 and anonac <= 2005 and transporte = 'Avion'



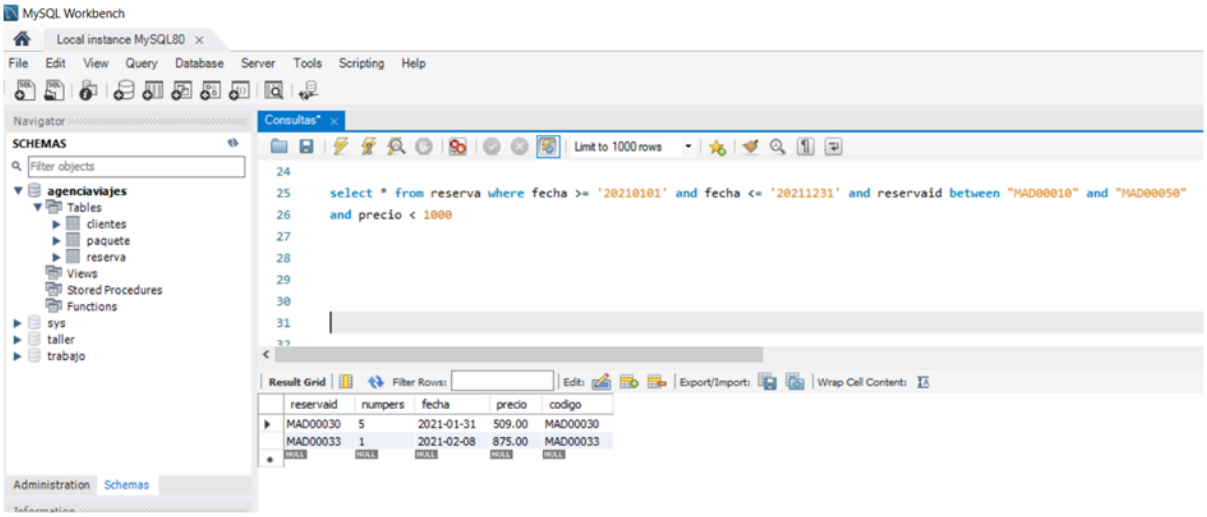
5.I.4.-What was the most commonly used means of transport for clients born in the 1960s?

select distinct transporte, count(*) from paquete a inner join clientes b on a.id = b.id where anonac >= 1960 and anonac < 1970 group by transporte order by transporte



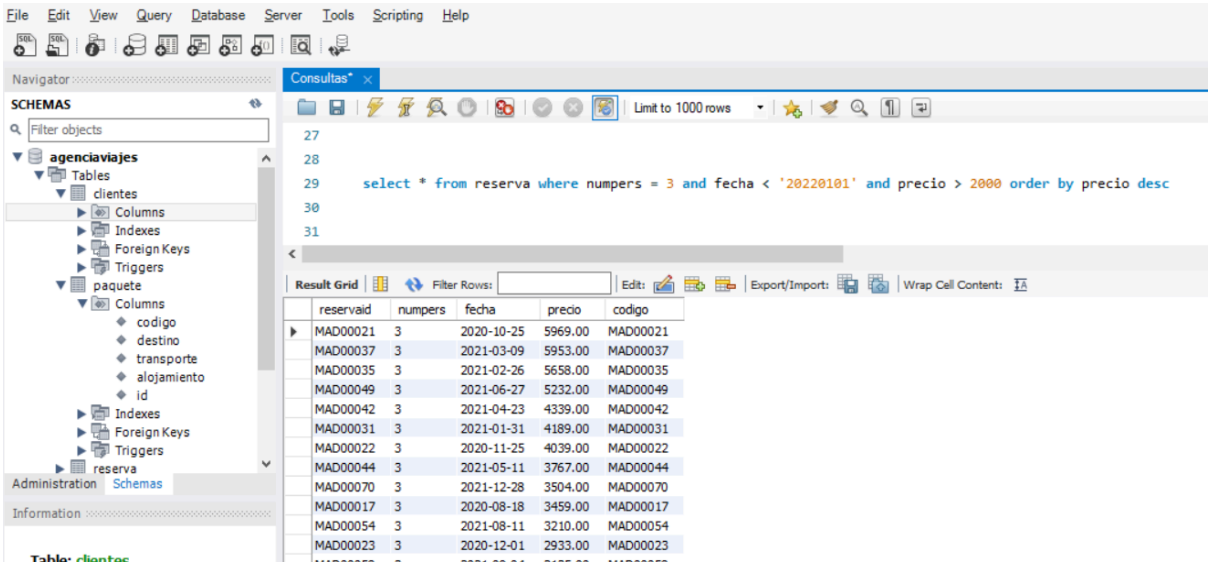
5.I.5.-How many trips were made in 2021 with a price under 1000 and where the reservation code is between MAD00010 and MAD00050?

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



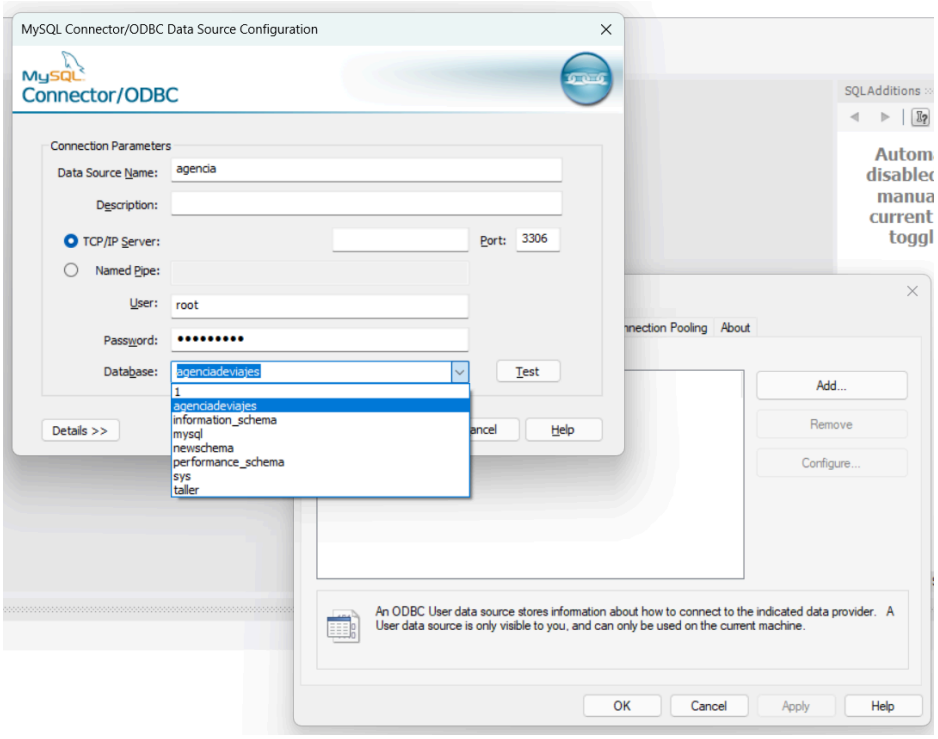
5.I.6.-Find the people who traveled in groups of 3 before 2022, paid more than 2000, and sort the price in descending order.

select * from reserva where numpers = 3 and fecha < '20220101' and precio > 2000 order by precio desc



5.II.-Consultas vía RStudio.

Lo primero es descargar los paquetes SQLDF y RODBC en RStudio, y por medio de MySQL First, download the SQLDF and RODBC packages in RStudio, and then download the "ODBC Data Source" using the MySQL Installer. Next, establish the connection by filling in the information requested by the connector. For example, the password is the same as the one used for the MySQL engine, and you should locate the database you created. Then, execute the SQL queries in the same way as before.



6.- We have created a relational model. However, could you implement the above in another type of database?

Yes, it's possible to implement a relational model in another type of database. For example, you could implement it in a NoSQL database, like MongoDB or Cassandra, using a document-based or column-based schema, respectively. It could also be implemented in a graph database, like Neo4j, using nodes and relationships to represent the entities and relationships of the relational model. However, the relational model is better suited to relational databases and might require more effort to adapt it to a NoSQL database. It's important to remember that each type of database has its own advantages and disadvantages, so it's crucial to evaluate which is the best option for a specific application before making a decision.

7.- Analyze the Advantages and Disadvantages of Using a Non-Relational Database .

Based on the material provided in this course, the following are the various advantages and disadvantages of using a non-relational database.

Advantages

Data Volume: Non-relational databases can handle large amounts of data due to their distributed structure. They can store data of any type and allow data distribution across a cluster.

● Versatility: Non-relational databases offer flexibility in terms of scaling or changing storage methods without needing extra configurations. Queries are executed using JSON.

● Horizontal Scalability: They support distributed structures, enabling horizontal scaling. This means that if additional operational nodes are needed, they can be added, allowing for easier workload balancing. This is based on a "shared-nothing" architecture.

● Resource Availability: Non-relational databases do not require large amounts of resources, allowing for gradual growth as needed.

● Optimization: NoSQL databases use internal algorithms to rewrite queries submitted by users or applications, optimizing performance for all operations.

● Working without Schema: Working without a predefined schema offers flexibility and is suitable for handling semi-structured data:

● Flexibility: It's often more useful to store raw data without a strict schema. With a flexible schema, there's less work to be done upfront. Agile environments might require constant schema changes based on evolving requirements.

● Handling Semi-Structured Data: Non-relational databases are better suited for processing semi-structured or unstructured data.

Disadvantages

- Atomicity. Without this feature, we won't find consistent information.
- Software Documentation. Since this is a relatively new database, some operations are limited, sometimes requiring advanced knowledge of the tools and the people who develop this software.
- Language Standards. We didn't find a predetermined language for all possible NoSQL solutions.
- GUI (Graphical User Interface) Tools. A graphical interface isn't available; access is via the console, requiring extensive knowledge of the commands for maintenance.
- Similar to the advantages of not using a schema, we also found some disadvantages:
    - Absence of metadata, which is often required for analysis tools.
    - Implicit Schema. A schema must be created by the program from which we access the database, and this schema will only be defined within that application. However, this gives rise to other problems: the same database being accessed from different applications (generating different schemas) and our information undergoing changes that will be reflected in the programs.

Conclusion

We can conclude that DBMSs are a useful tool for any business, allowing various automated queries when needed. They enable the establishment of rules to optimally link and manage different tables and can be used with software like DBeaver, which helps improve import/export processes and syntax management. In this case, using MySQL, we found it to be a powerful and easy-to-use tool that allows data storage, retrieval, and manipulation. 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. Retrieved January 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. Retrieved January 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. Retrieved January 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. Retrieved January 18, 2023, from https://pc-solucion.es/terminos/modelo-relacional/

N.D. (n.d.). *Difference between VARCHAR and CHAR data type in SQL Server? [Explained]*. Java67. Retrieved January 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*. Catapult Systems. Retrieved January 21, 2023, from https://www.catapultsystems.com/blogs/float-vs-decimal-data-types-in-sql-server/