next EDUCACIÓN

Entity-Relationship Model: Travel Agency

Subject: Database Management Systems and Data Parallelism

Profesor: Sergio García

Integrantes:

Valesca Bravo

Leonardo Rivas Duarte

Marta del Carmen Mayoral Santos

Mertxe Galán Espiga

Table of Contents

Introduction

In this work, we will examine how to develop a database, from modeling to its implementation in MySQL through SQL.

We have chosen the tourism industry, considering a hypothetical travel agency. We begin our work by creating an entity-relationship model in the program Dia, where we establish the rules that our business model would follow.

To convert the Entity-Relationship model into a Relational Model, we explored some online modeling alternatives, such as Vertabelo, which offers export opportunities for various database engines and, for students, is free of charge.

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 by applying some rules in Excel.

Finally, we used SQL to create the tables and insert the information into MySQL.

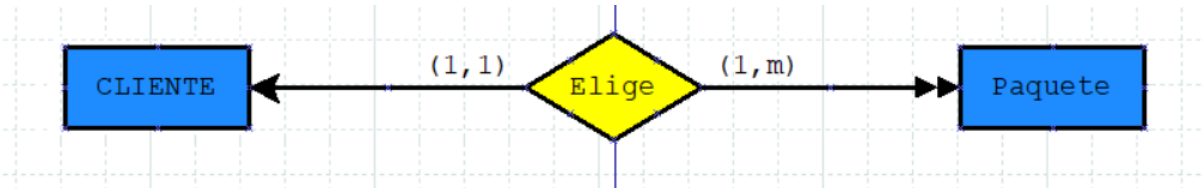
1.- Entity-Relationship Model

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:

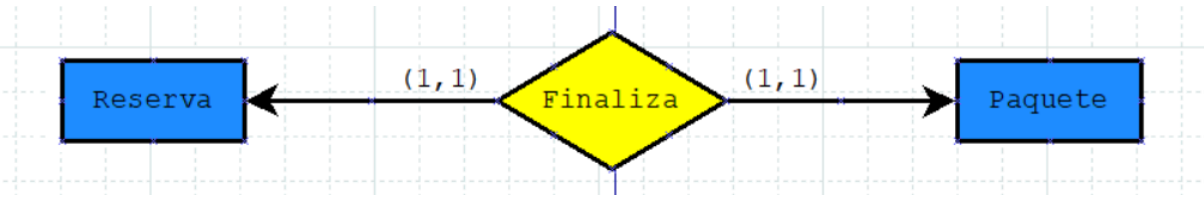
Client: We defined its attributes as Name, Surname, Gender, Email, Mobile, Year of Birth, and finally Client ID, which would serve as the Primary Key.
Package: It would have its own identifying code as the Primary Key, along with other attributes like Destination, Transportation, and Accommodation.
Reservation: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

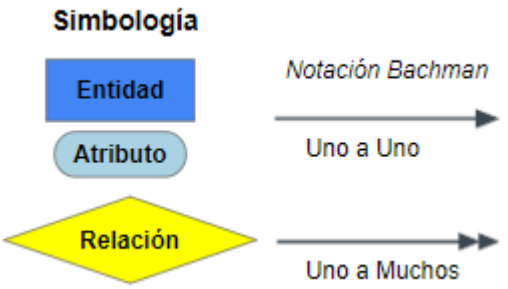
Rules and Conditions:

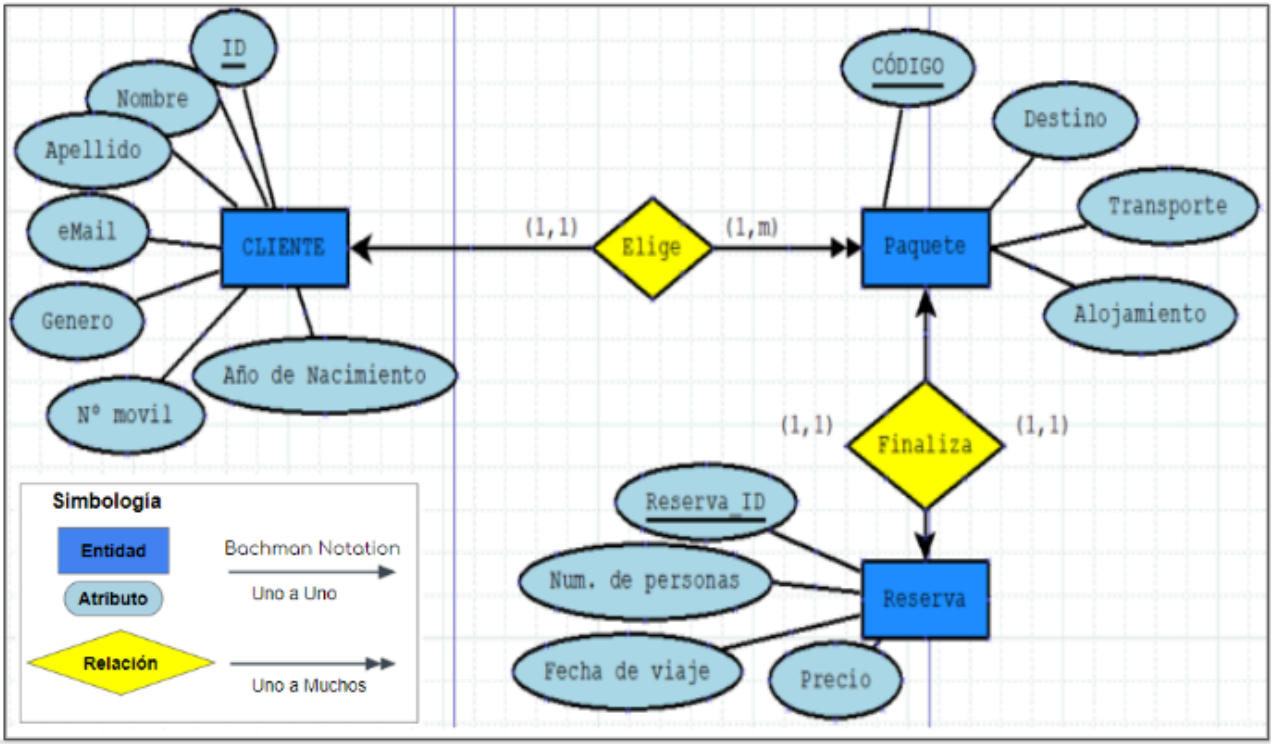
- We defined that the Client Entity would be related to the Package Entity, where a client could purchase many packages, but each package would be associated with only one client.



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




Simbología
Entidad
Atributo
Relación
Notación Bachman
Uno a Uno
Uno a Muchos


ID
Nombre
Apellido
eMail
Genero
Año de Nacimiento
Nº movil
CLIENTE
(1,1)
Elige
(1,m)
CÓDIGO
Destino
Transporte
Paquete
Alojamiento
(1,1)
Finaliza
(1,1)
Reserva_ID
Num. de personas
Reserva
Fecha de viaje
Precio
Simbología
Entidad
Atributo
Relación
Bachman Notation
Uno a Uno
Uno a Muchos

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:
    - 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.
    - 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).
    - 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.

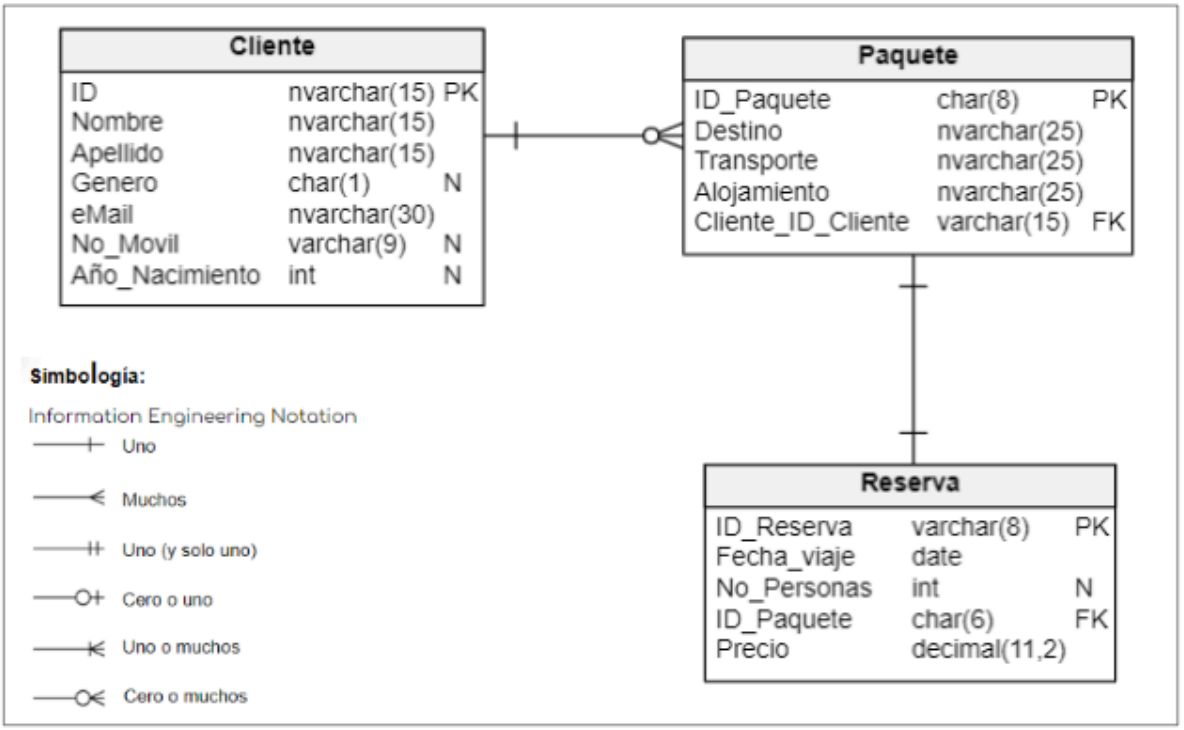
Package:

- ID_Paquete doesn't require more than an "char" with 8 spaces.
- Destination, Transportation, and Accommodation were created as "nvarchar" with a capacity of 25 spaces.

Reservation:

- ID_Reserva: "Varchar" with 8 spaces.
- Travel Date: Assigned as a "Date" type.
- Number of People: "Int" with 6 spaces, which will be nullable.

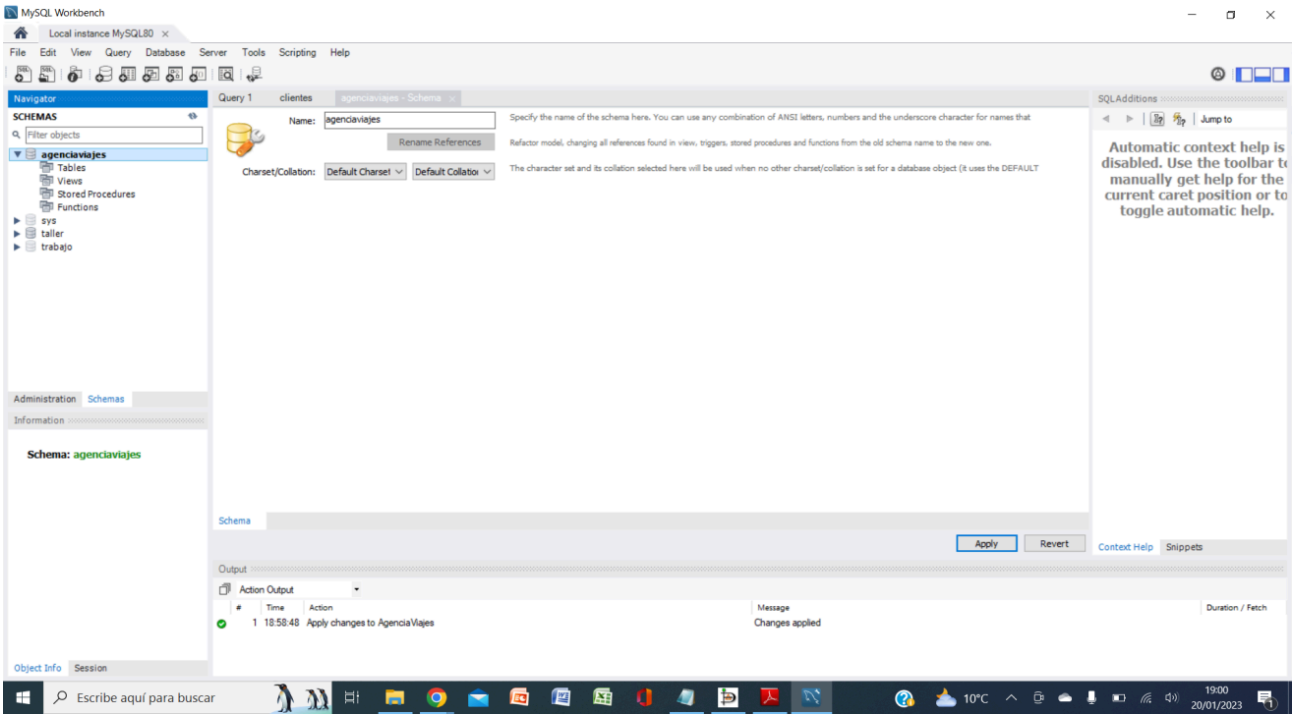
    Price: We used "Decimal" instead of "Float" because the former is more precise, and we're dealing with money. (Skov, 2009).

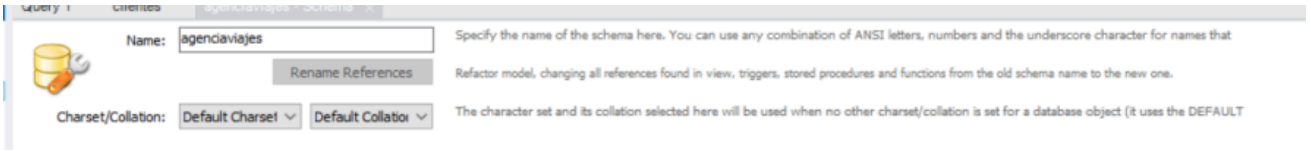

Cliente
ID nvarchar(15) PK
Nombre nvarchar(15)
Apellido nvarchar(15)
Genero char(1) N
eMail nvarchar(30)
No_Movil varchar(9) N
Año_Nacimiento int N
Paquete
ID_Paquete char(8) PK
Destino nvarchar(25)
Transporte nvarchar(25)
Alojamiento nvarchar(25)
Cliente_ID_Cliente varchar(15) FK
Reserva
ID_Reserva varchar(8) PK
Fecha_viaje date
No_Personas int N
ID_Paquete char(6) 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
ID	Identity document.	CHAR(15)	N

Nombre	First Name	NVARCHAR (15)	N
Apellido	Last Name	NVARCHAR (15)	N
E-mail	Dirección email	NVARCHAR (30)	N
Genero	Gender (M male, F female).	NVARCHAR (1)	Y
Móvil	Mobile phone number.	NVARCHAR (9)	Y
Año de nacimiento	Year of birth.	SMALLINT	Y

PAQUETE

Column Name	**Meaning**	**Data Type**	**NULLS allowed**
Código	Package code	CHAR (8)	N
Destino	Destination city	NVARCHAR (25)	N
Transporte	Mode of transport	NVARCHAR (25)	Y
Alojamiento	Type of accommodation	NVARCHAR (25)	Y
ID	Identity document.	NVARCHAR (15)	N

RESERVA

Column Name	**Meaning**	**Data Type**	**NULLS allowed**
Reservaid	Reservation code	CHAR (8)	N
Num. De Personas	Destination city	SMALLINT	Y
Fecha de viaje	Mode of transport	DATE	N
Precio	Type of accommodation	FLOAT (11,2)	N
Código	Destination city	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

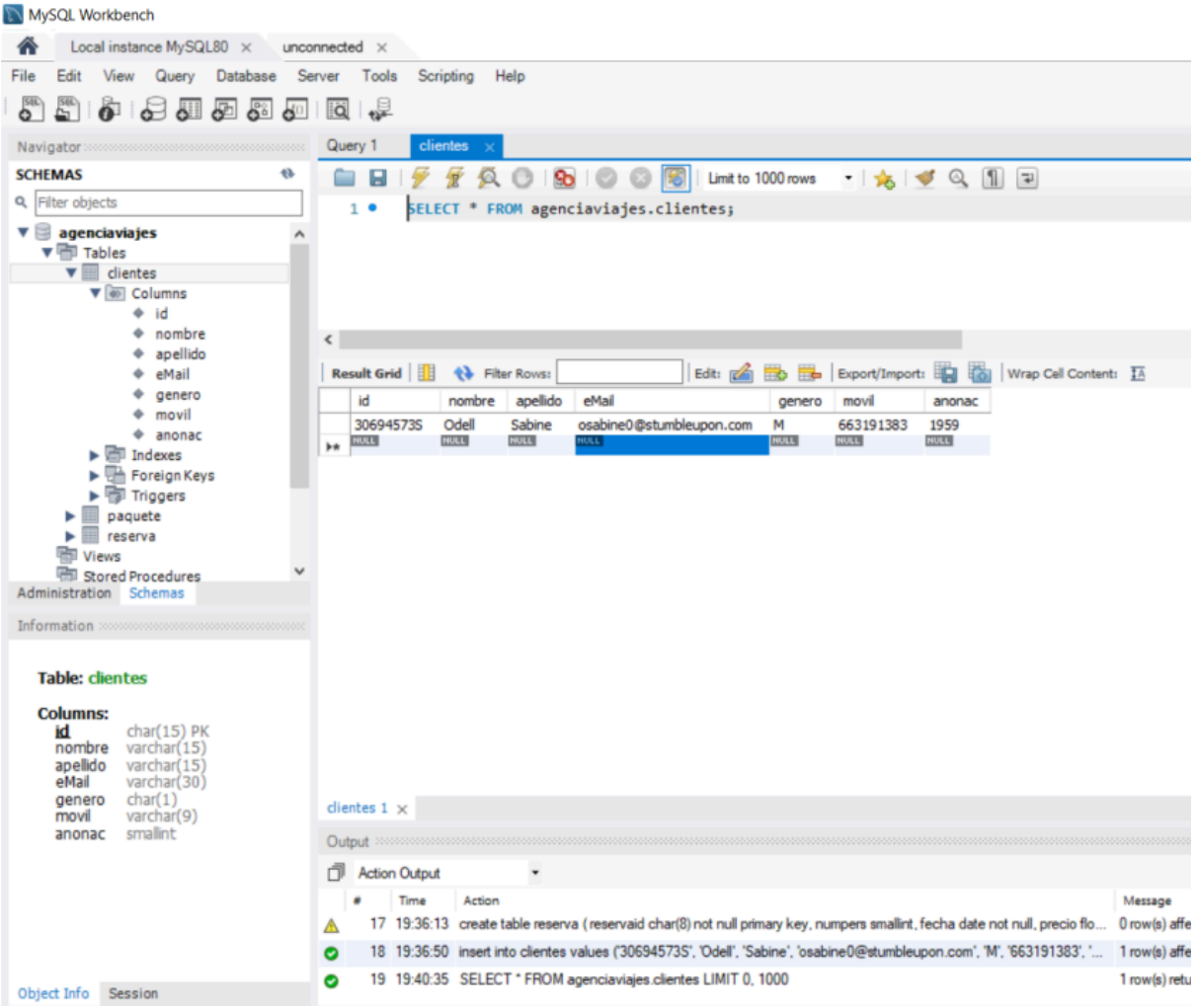
Business rules define the guidelines that determine an organization's business activities, but within our entity-relationship diagram, these rules will set our attributes, relationships, and constraints on our data. For example, one of our business rules is that our clients can have more than one package 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')
```

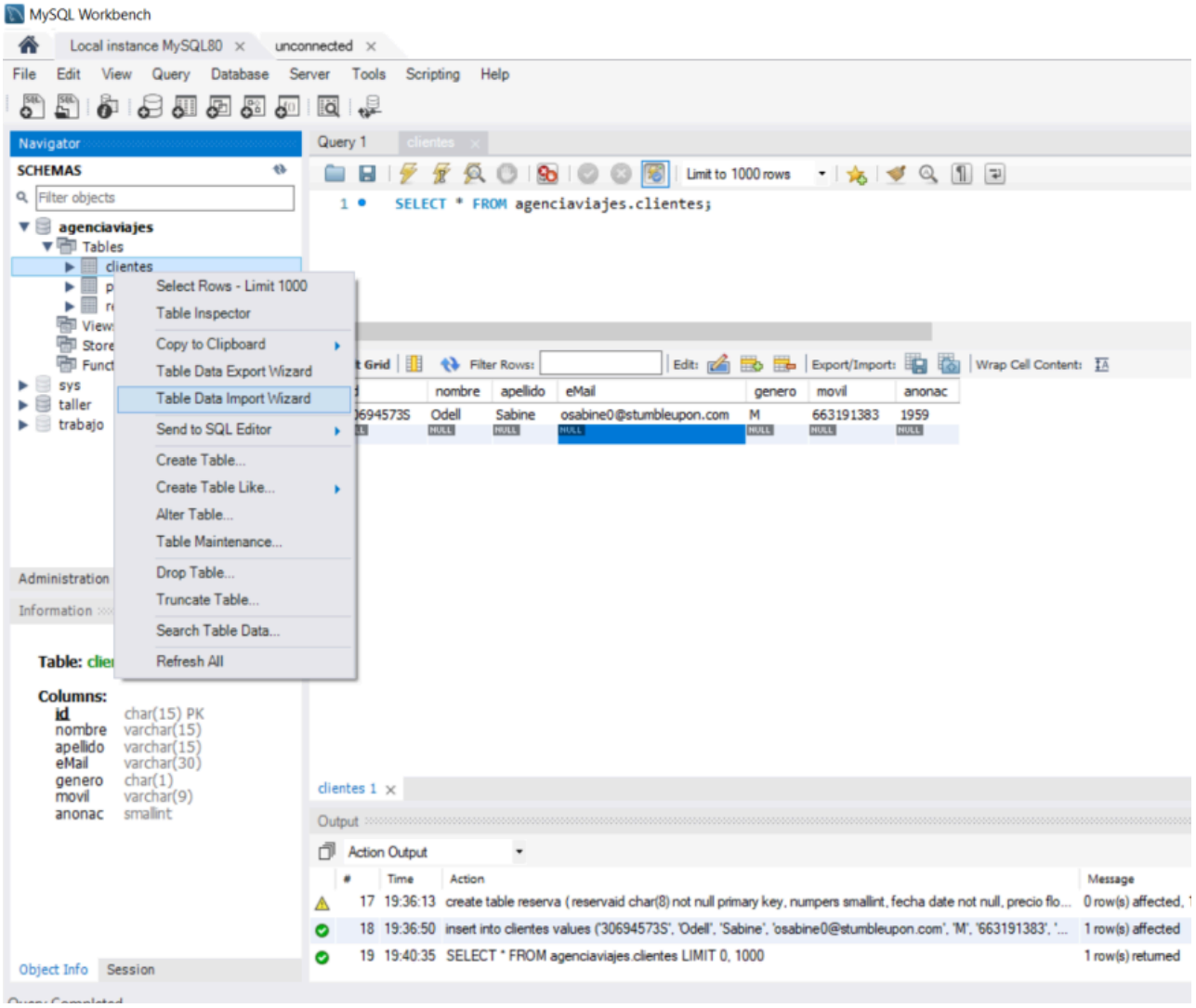
After executing this statement, we query the Clients table and get this record:

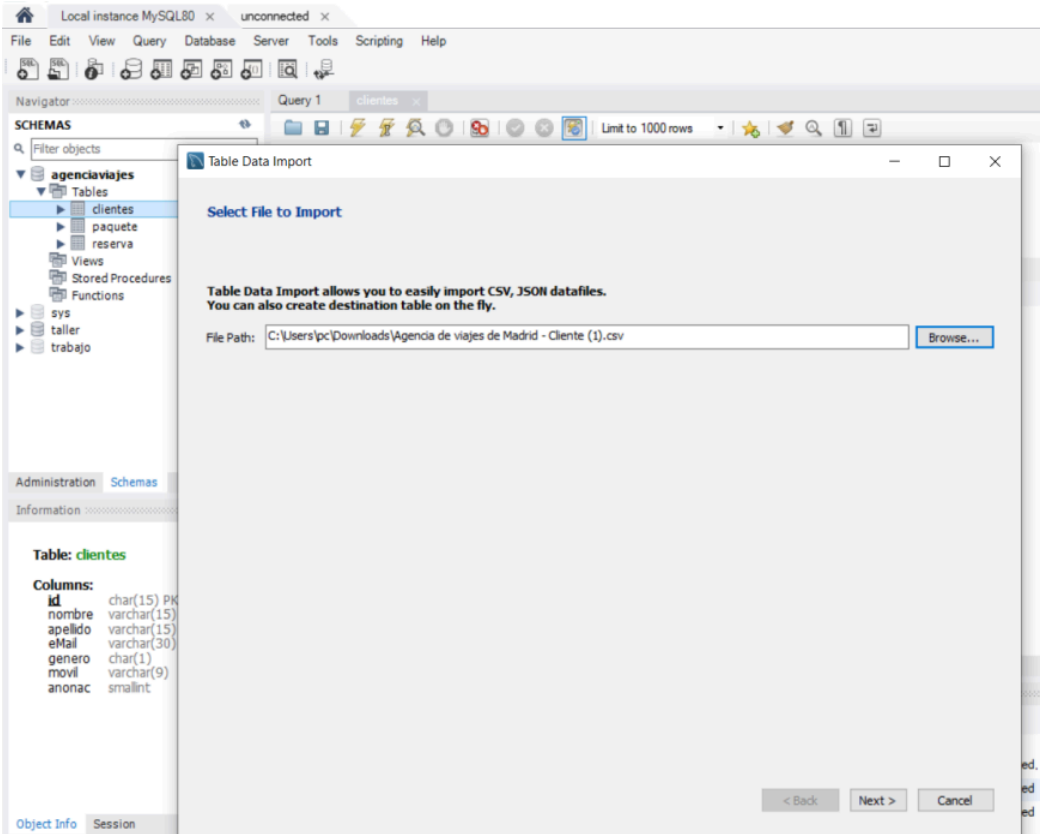


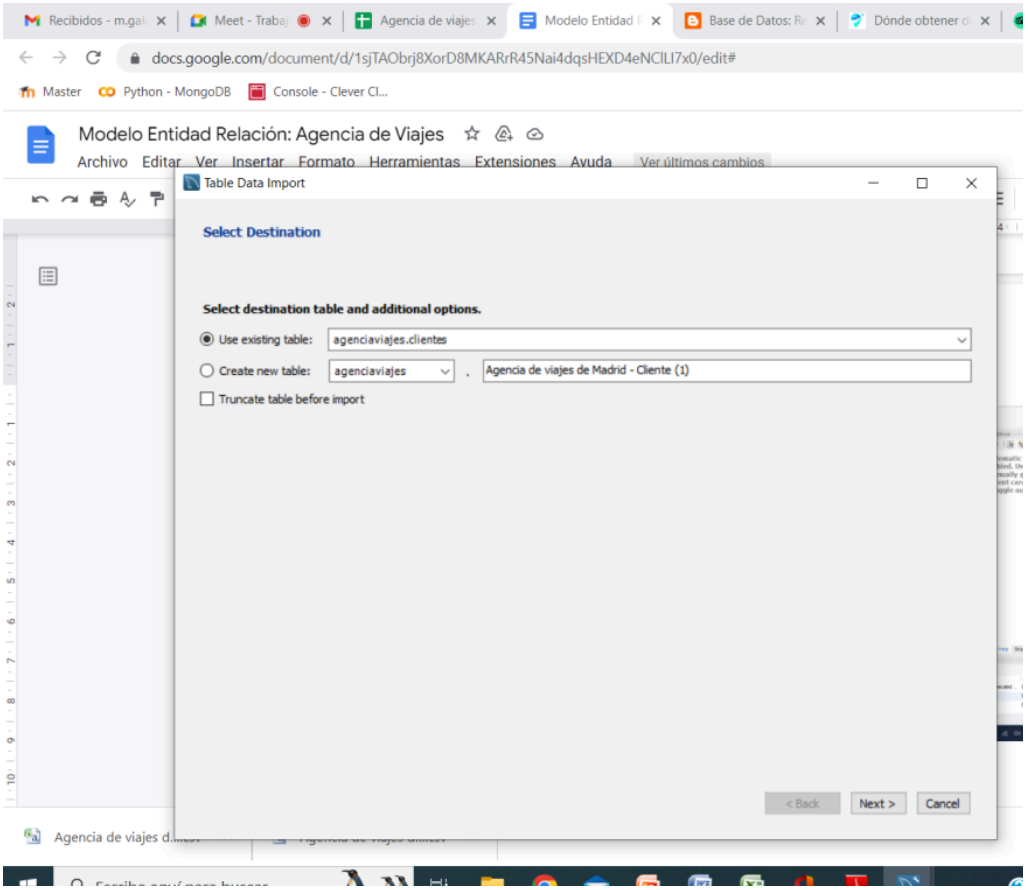
To optimize data loading, and knowing how to do it manually, for the rest of the data, we will use bulk imports through CSV file imports. We start with an Excel document in Google Drive with 3 sheets (one for Clients, one for Packages, and one for Reservations), where group members entered data, and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

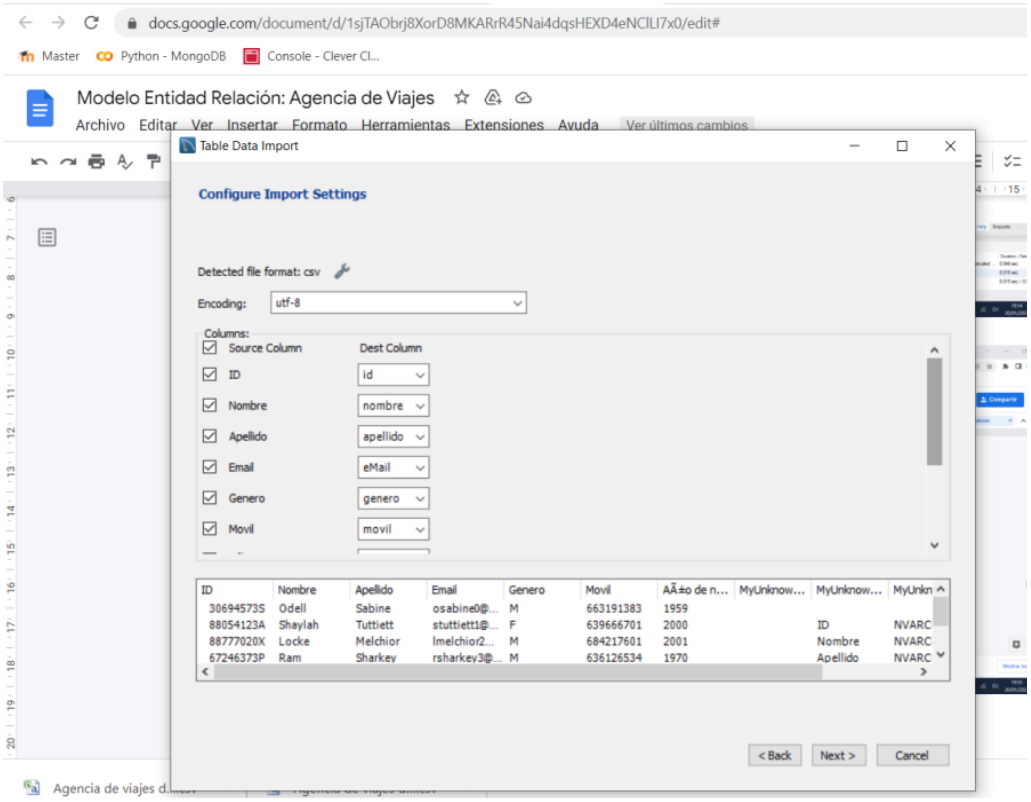
Bulk Import of Data into the Clients Table

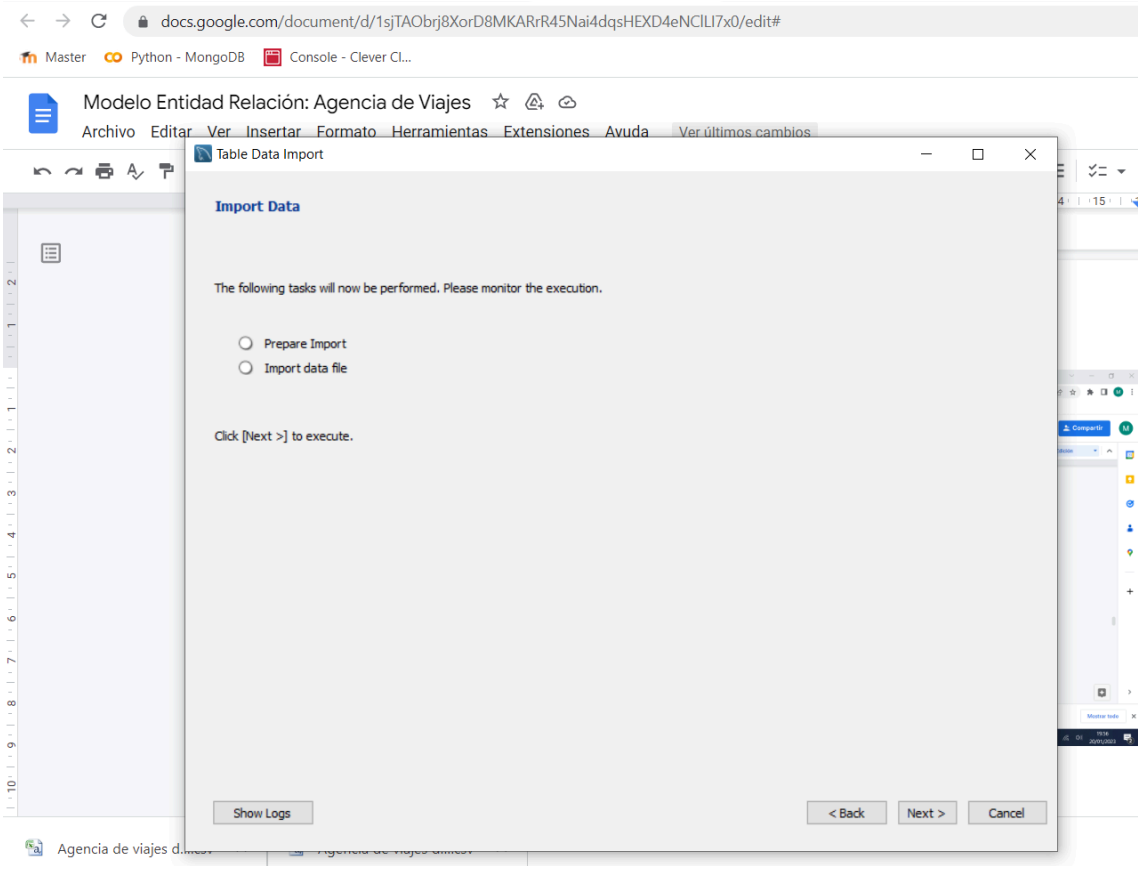
To import data into the Clients table, we select the table, right-click, and choose the 'Table Data Import Wizard' option. The following outlines the steps we took:

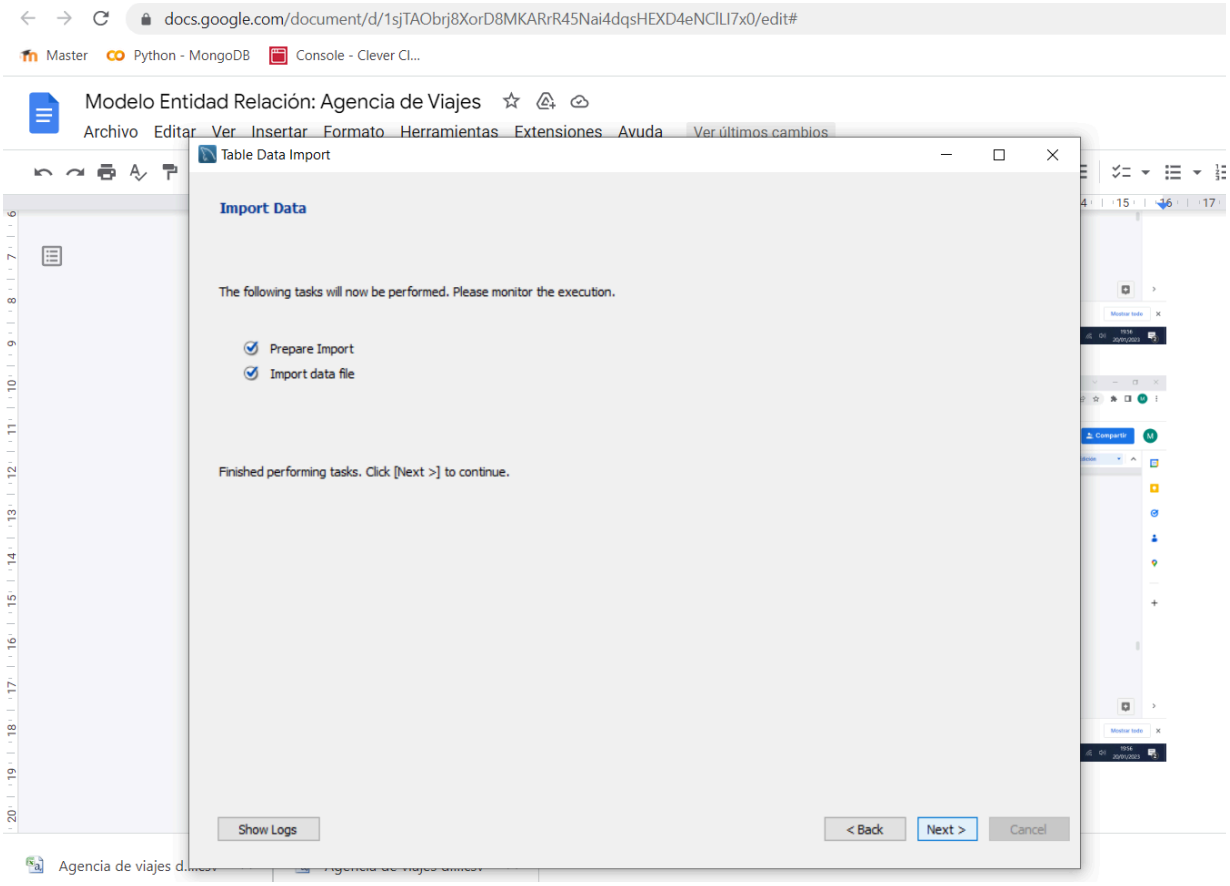



Local instance MySQL80
unconnected
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
agenciaviajes
Tables
clientes
paquete
reserva
Views
Stored Procedures
Functions
sys
taller
trabajo
Administration Schemas
Information
Table: clientes
Columns:
id char(15) PK
nombre varchar(15)
apellido varchar(15)
eMail varchar(30)
genero char(1)
movil varchar(9)
anonac smallint
Object Info Session
Query 1
clientes
Limit to 1000 rows
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

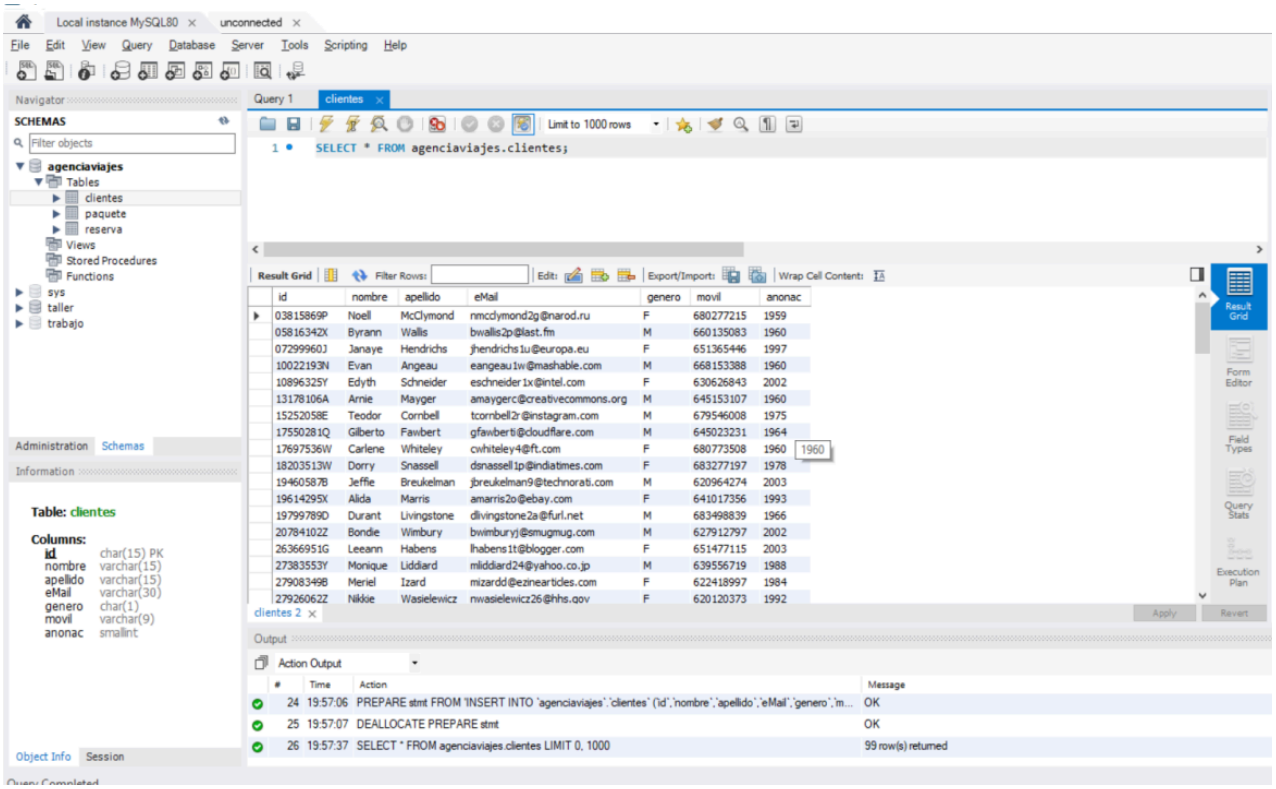

docs.google.com/document/d/1sjTAObrj8XorD8MKARrR45Nai4dqsHEXD4eNClLI7x0/edit#
Master
Python - MongoDB
Console - Clever Cl...
Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda
Table Data Import
Select Destination
Select destination table and 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


docs.google.com/document/d/1sjTAObrj8XorD8MKARrR45Nai4dqsHEXD4eNClLI7x0/edit#
Master
Python - MongoDB
Console - Clever Cl...
Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda
Ver últimos cambios
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
ID Nombre Apellido Email Genero Movil AÃ±o de n... MyUnknow... MyUnknow... MyUnkn
30694573S Odell Sabine osabine0@... M 663191383 1959
88054123A Shaylah Tuttiett stuttiett1@... F 639666701 2000 ID NVARC
88777020X Locke Melchior lmelchior2... M 684217601 2001 Nombre NVARC
67246373P Ram Sharkey rsharkey3@... M 636126534 1970 Apellido NVARC
< Back
Next >
Cancel
Agencia de viajes d...


docs.google.com/document/d/1sjTAObrj8XorD8MKARrR45Nai4dqsHEXD4eNClLI7x0/edit#
Master
Python - MongoDB
Console - Clever Cl...
Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda
Ver últimos cambios
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
Agencia de viajes d...


docs.google.com/document/d/1sjTAObrj8XorD8MKARrR45Nai4dqsHEXD4eNClLI7x0/edit#
Master
Python - MongoDB
Console - Clever Cl...
Modelo Entidad Relación: Agencia de Viajes
Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda Ver últimos cambios
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
Agencia de viajes d...

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

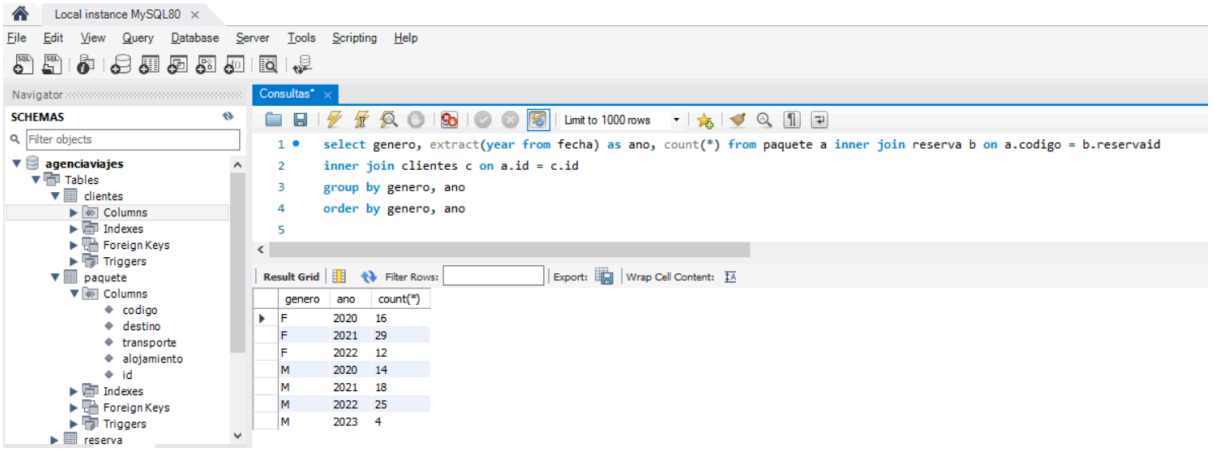
We propose the following queries to work with our database:

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:

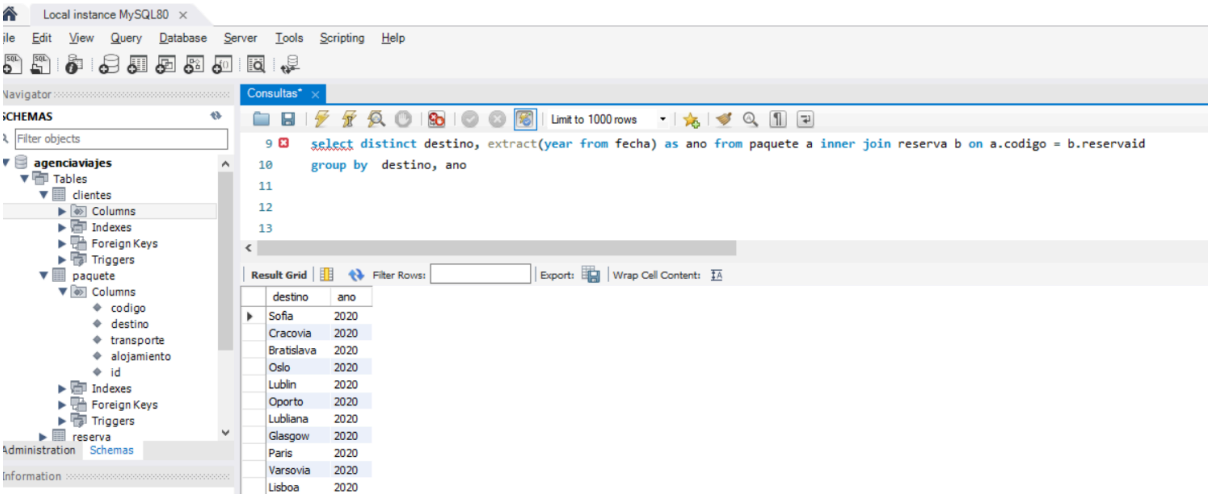
1. **How many packages have been purchased by gender and reservation year?**

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



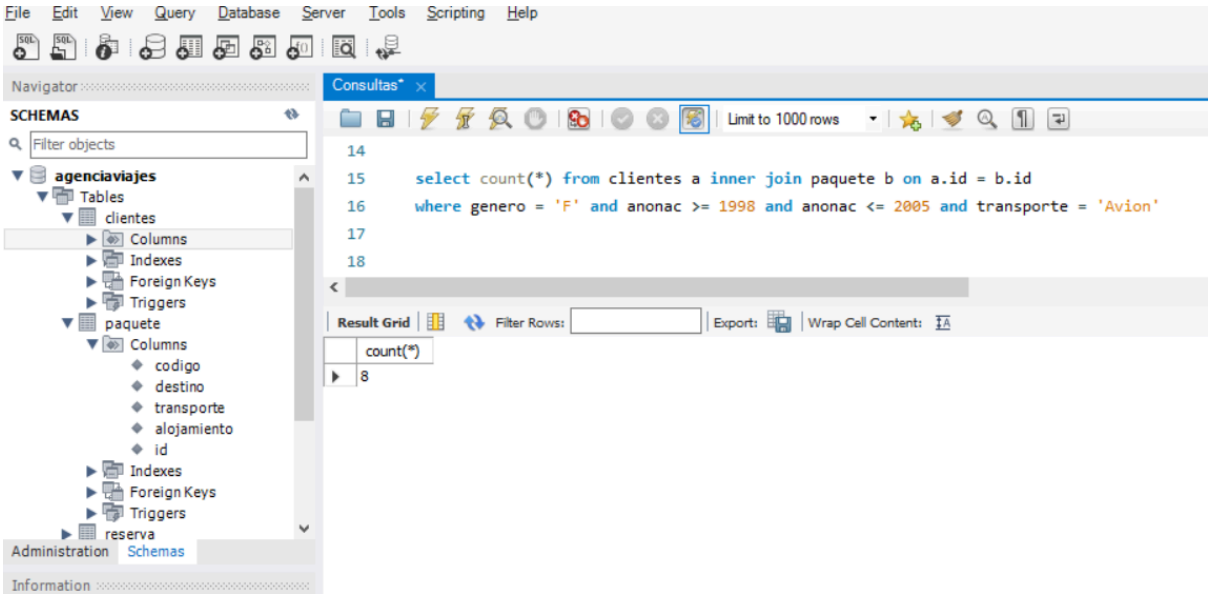
2. **List the different destinations chosen each year.**

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



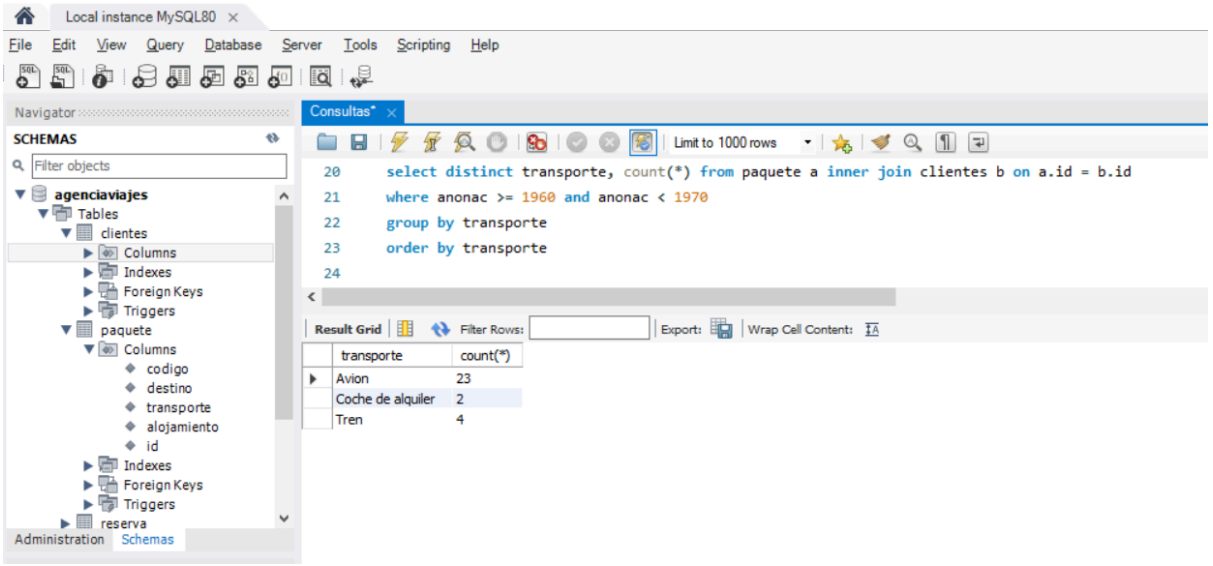
3. **How many female clients aged 18 to 25 used the airplane as a means of transport?**

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



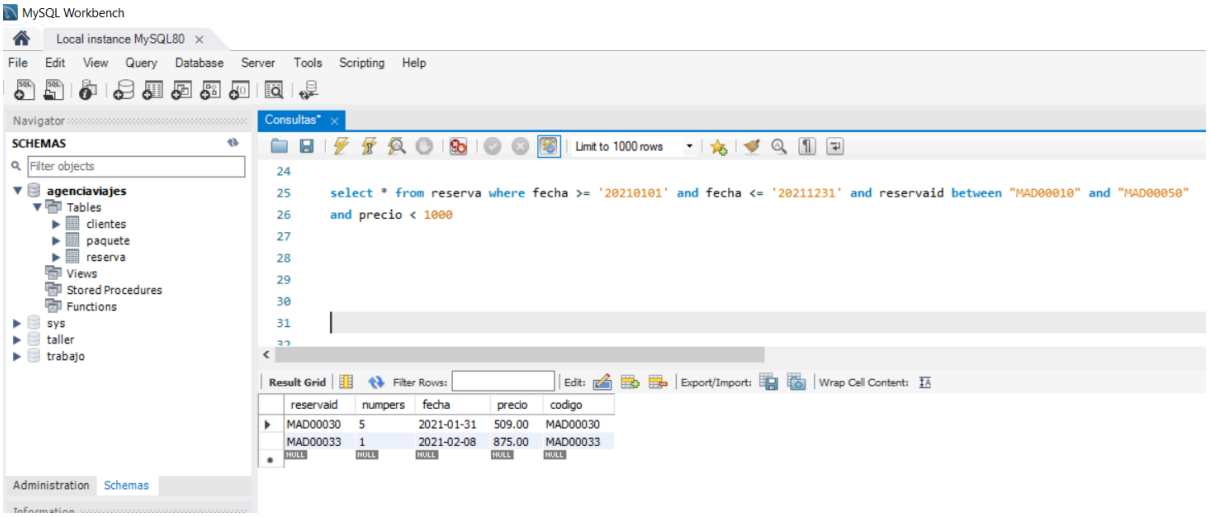
4. **What was the most commonly used means of transport for clients born in the 1960s?**

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



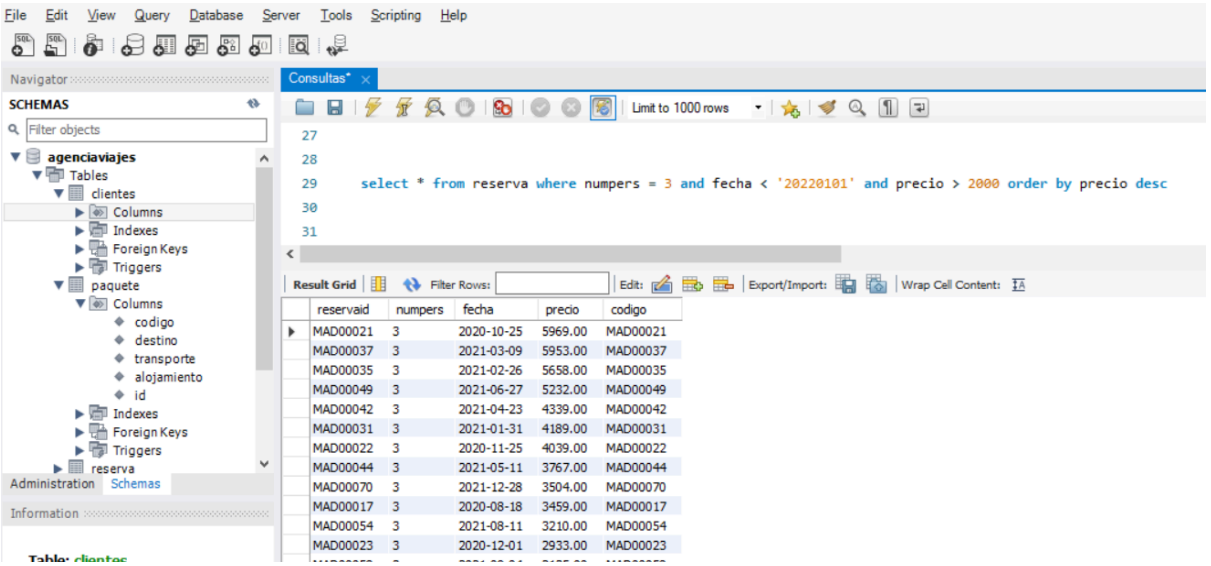
5. **How many trips were made in 2021 with a price under 1000 and where the reservation code is between MAD00010 and MAD00050?**

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



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 wherenumpers = 3 and fecha < '20220101' and precio > 2000 orderby precio desc


File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
agenciaviajes
Tables
clientes
Columns
Indexes
Foreign Keys
Triggers
paquete
Columns
codigo
destino
transporte
alojamiento
id
Indexes
Foreign Keys
Triggers
reserva
Administration Schemas
Information
Consultas*
Limit to 1000 rows
27
28
29 select * from reserva where numpers = 3 and fecha < '20220101' and precio > 2000 order by precio desc
30
31
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
reservaid numpers fecha precio codigo
MAD00021 3 2020-10-25 5969.00 MAD00021
MAD00037 3 2021-03-09 5953.00 MAD00037
MAD00035 3 2021-02-26 5658.00 MAD00035
MAD00049 3 2021-06-27 5232.00 MAD00049
MAD00042 3 2021-04-23 4339.00 MAD00042
MAD00031 3 2021-01-31 4189.00 MAD00031
MAD00022 3 2020-11-25 4039.00 MAD00022
MAD00044 3 2021-05-11 3767.00 MAD00044
MAD00070 3 2021-12-28 3504.00 MAD00070
MAD00017 3 2020-08-18 3459.00 MAD00017
MAD00054 3 2021-08-11 3210.00 MAD00054
MAD00023 3 2020-12-01 2933.00 MAD00023

6. We have created a relational model. However, could the above be implemented 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: Lack of atomicity may result in inconsistent information.
- Software Documentation: Non-relational databases are relatively new, so some operations are limited, requiring advanced knowledge from developers and tool users.
- Language Standards: No standardized language for all NoSQL solutions.
- GUI Tools: Often, there's no graphical user interface; access is via console commands, requiring advanced knowledge of command-line instructions.
- Working without Schema: While schema-less design offers flexibility, it can also lead to disadvantages:
    - Lack of Metadata: Often, metadata is required for specific tools and operations.
    - Implicit Schema: An implicit schema must be defined within the application accessing the data, which can lead to problems when multiple applications access the same database, creating varying schemas, or when information changes and impacts dependent applications.

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.

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/