

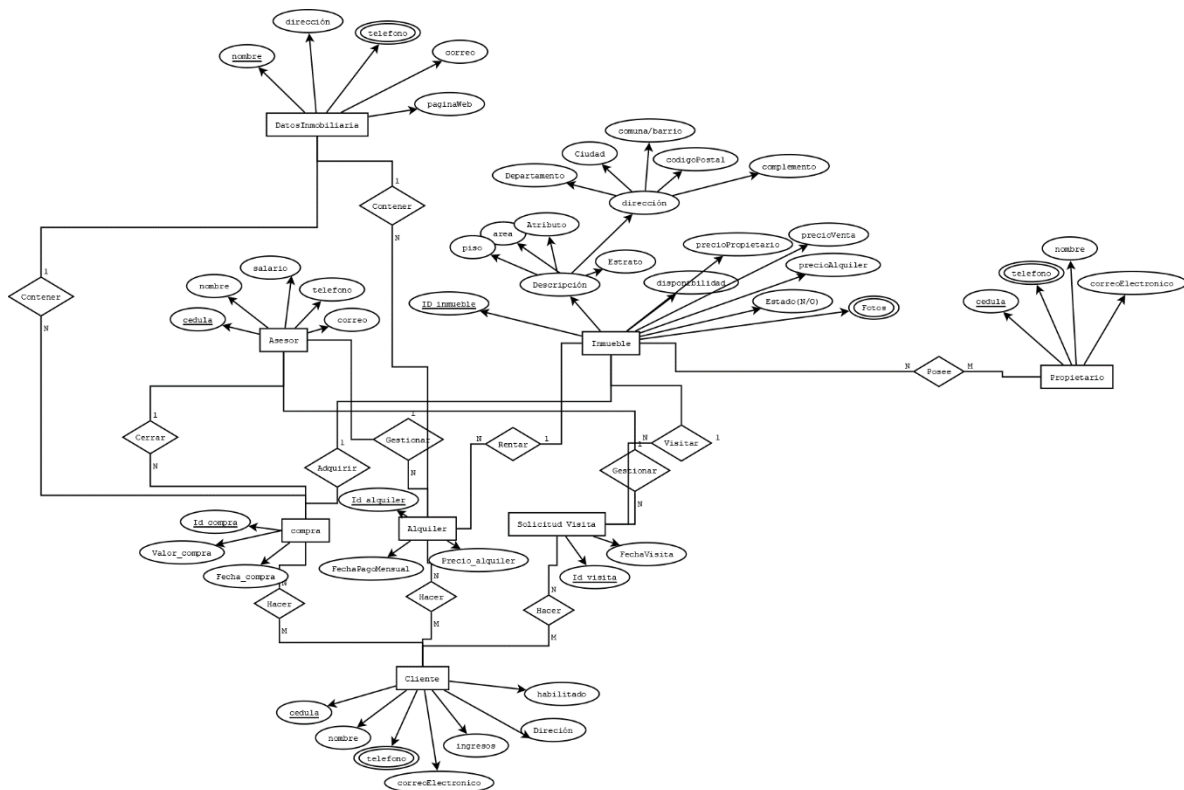
## DOCUMENTACION TALLER BASES DE DATOS

1. Empecé a hacer fueron las cardinalidades en el modelo Entidad-relación.

Entonces, las cardinalidades que encontré fueron las siguientes:

- ✓ Cliente-compra: N:M
- ✓ Cliente-alquiler: N:M
- ✓ Cliente-Solicitud Visita: N:M
- ✓ Compra-datos inmobiliaria: N:1
- ✓ Compra-asesor: N:1
- ✓ Compra-inmueble: N:1
- ✓ Alquiler-asesor: N:1
- ✓ Alquiler-datos inmobiliaria: N:1
- ✓ Alquiler-inmueble: N:1
- ✓ Solicitud Visita-asesor: N:1
- ✓ solicitud Visita-inmueble: N:1
- ✓ inmueble-propietario: N:M

Anexo fotos del modelo entidad relación sobre una inmobiliaria

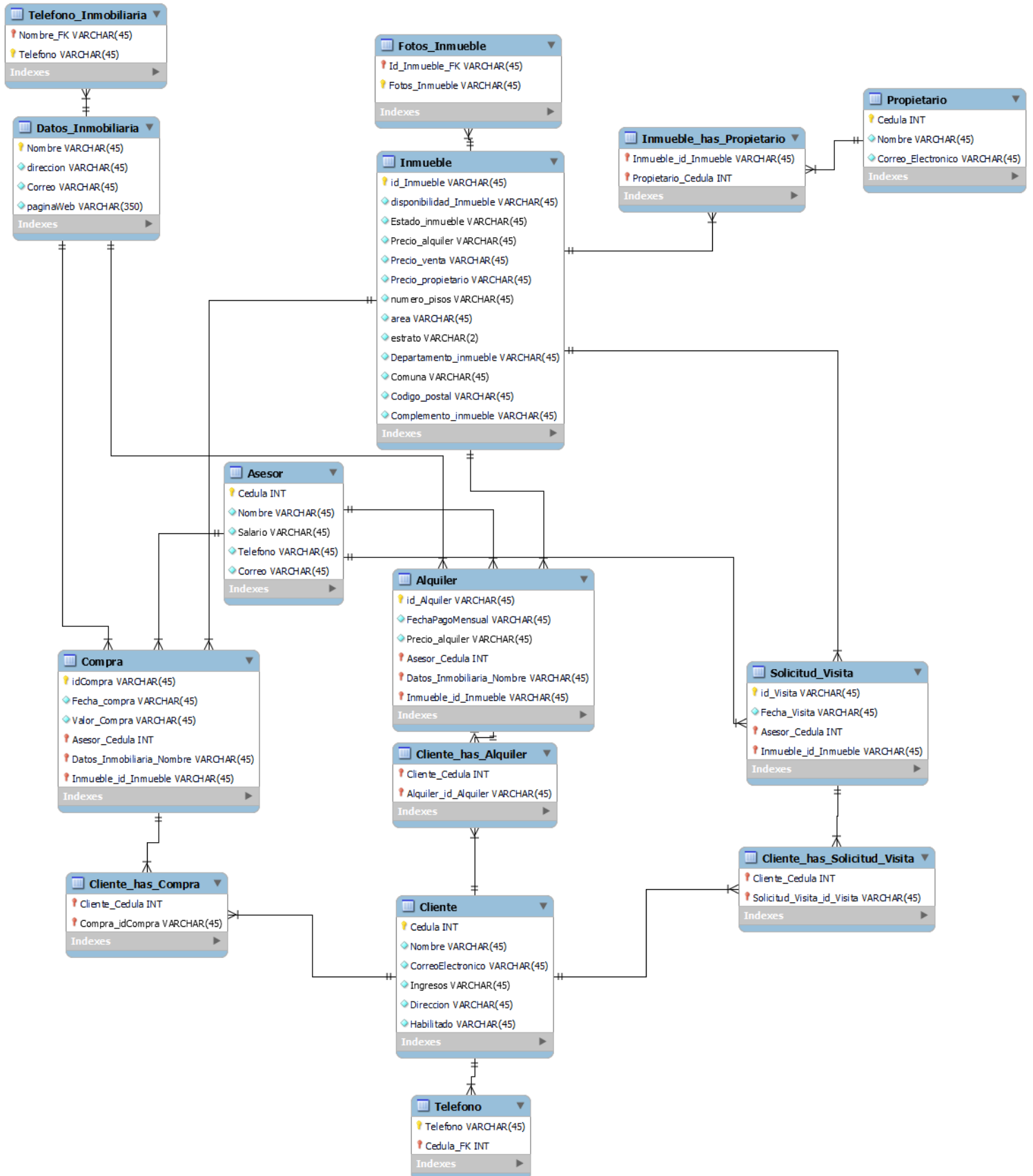


2. Después empecé por transformar el modelo entidad-relación a un modelo relación, Para esto utilice el programa WorkBench para hacer el diagrama de modelo relación teniendo en cuenta las entidades y cuáles eran sus atributos y de que tipo eran.

Una vez con las entidades definidas en el programa con todos sus atributos y con sus respectivas llaves primarias empecé a sacar los atributos que eran compuestos para hacerlos en una tabla aparte y hacer el respectivo procedimiento con sus clases madres.

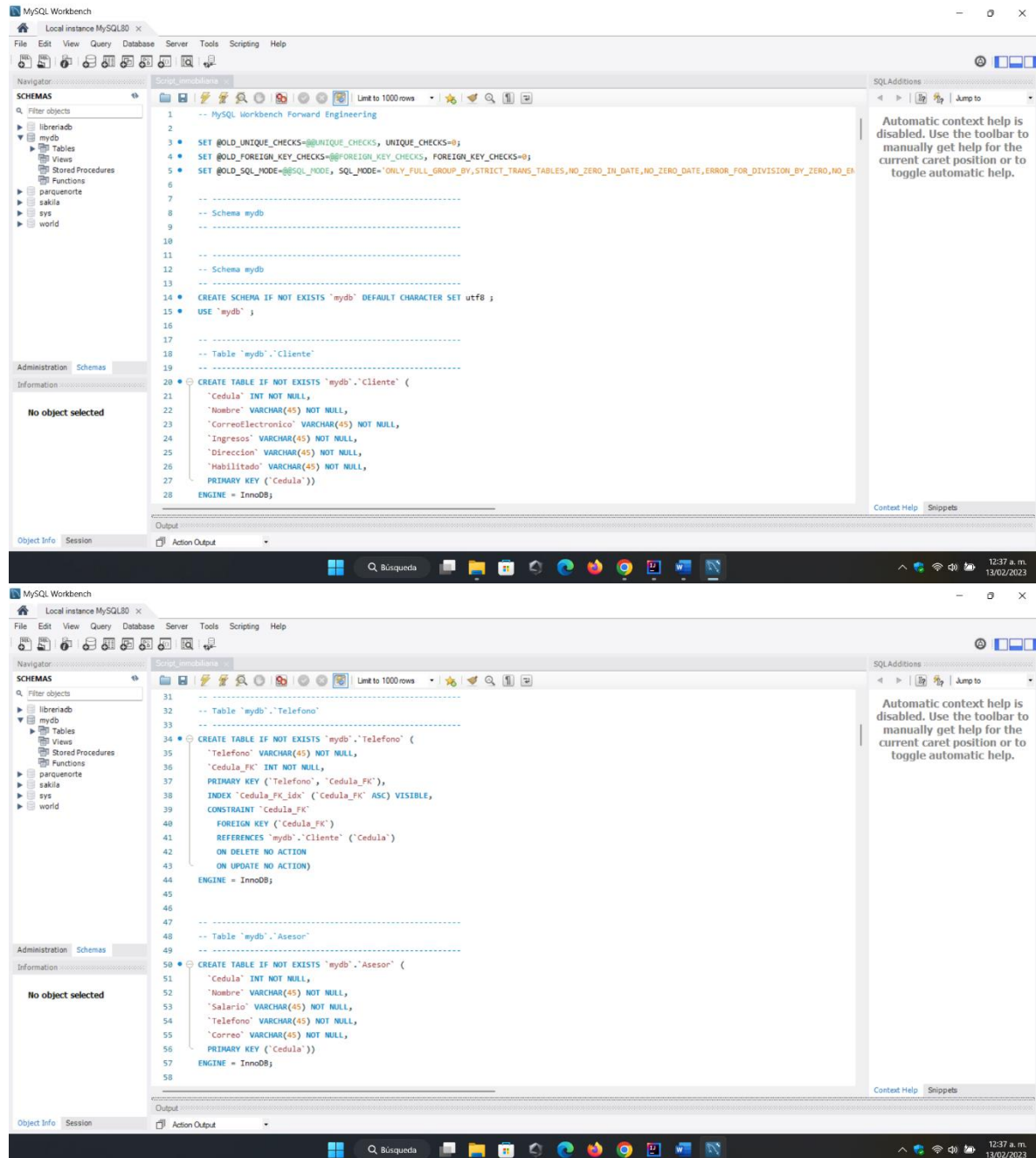
3. Una vez tuve todo lo anterior listo empecé a relacionar mi Modelo relacional, debo aclarar que esta parte fue relativamente fácil ya que el programa detecta el tipo de cardinalidad y crea las tablas correspondientes por ejemplo las de N:M

El resultado fue el siguiente:



Una vez hecho el modelo relacional procedí a aplicar la ingeniería hacia adelante, esto genero un script correspondiente al diagrama relacional antes creado.

Este fue el resultado:



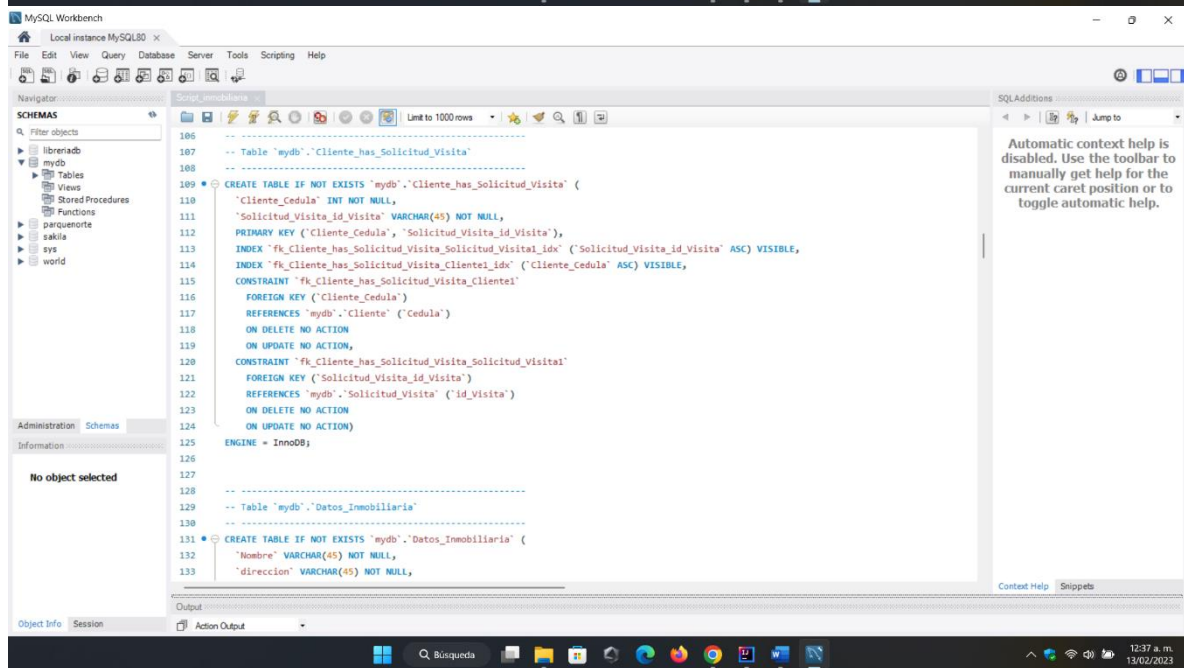
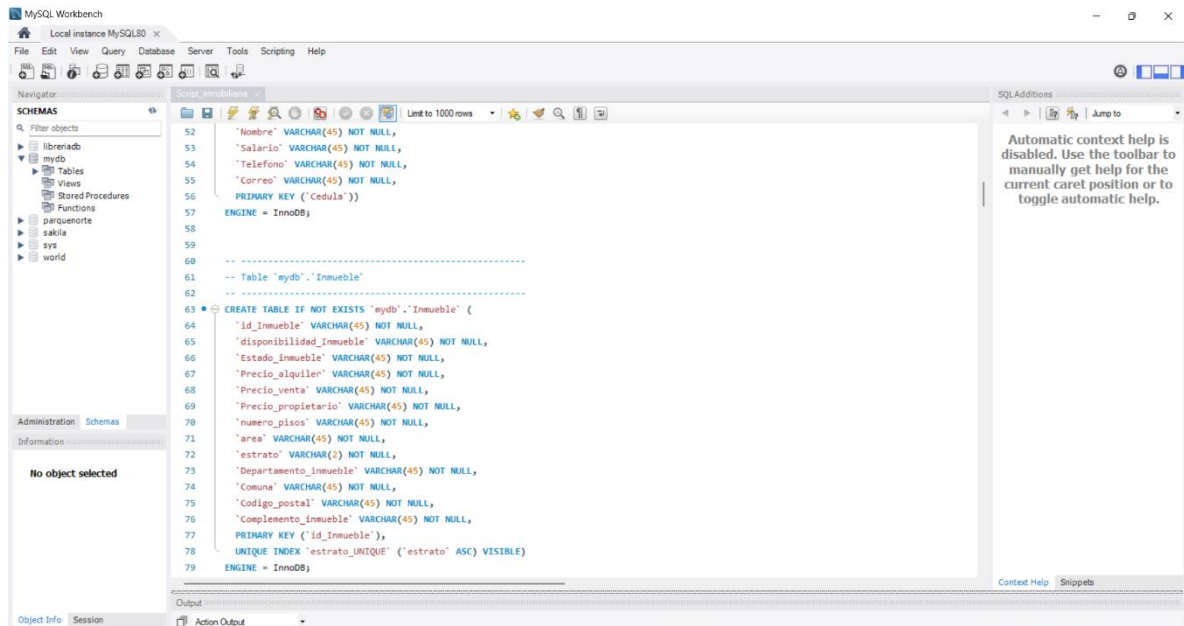
The image displays two screenshots of the MySQL Workbench interface, showing the execution of SQL scripts to create a database schema.

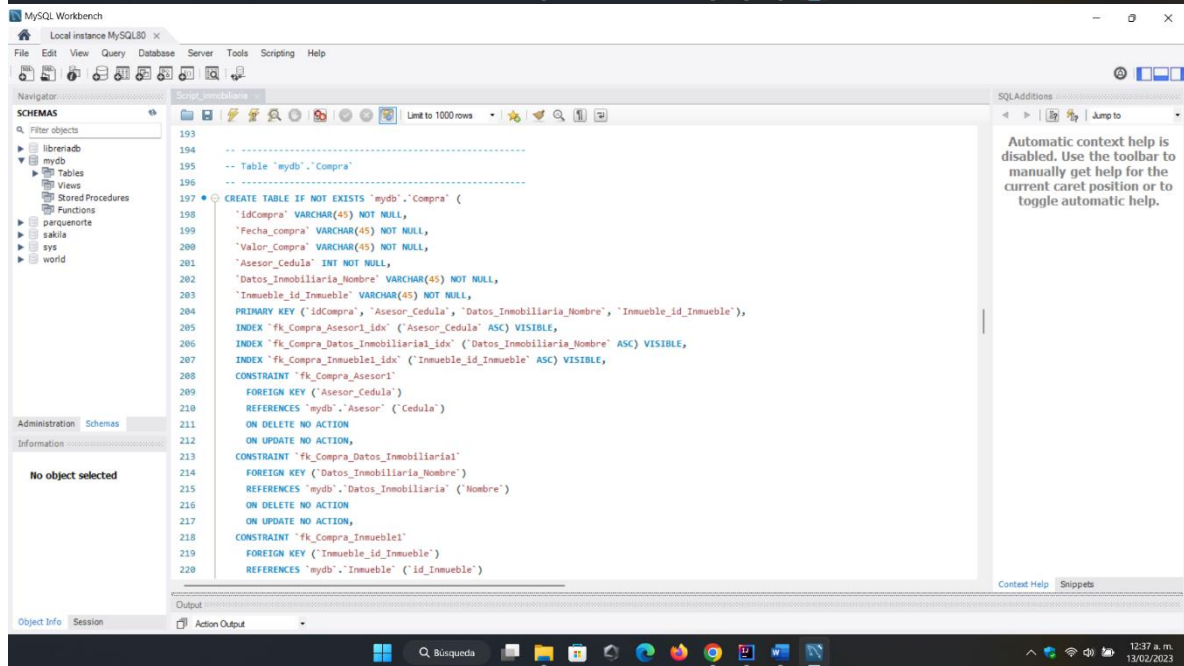
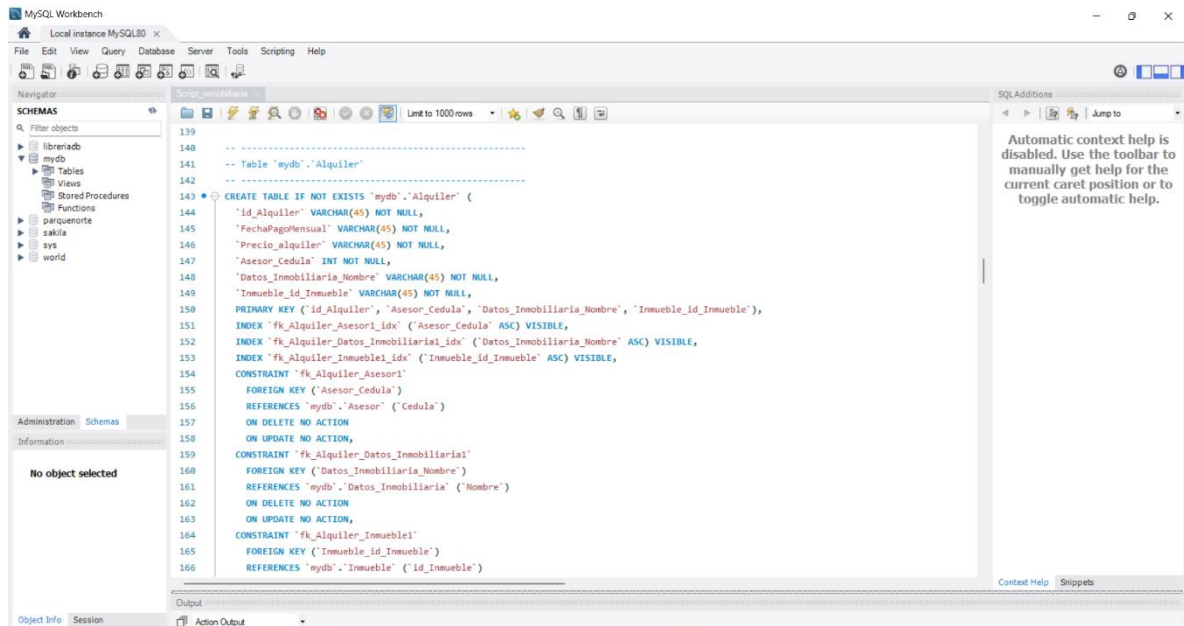
**Top Screenshot:** The script defines the initial setup and the 'Cliente' table.

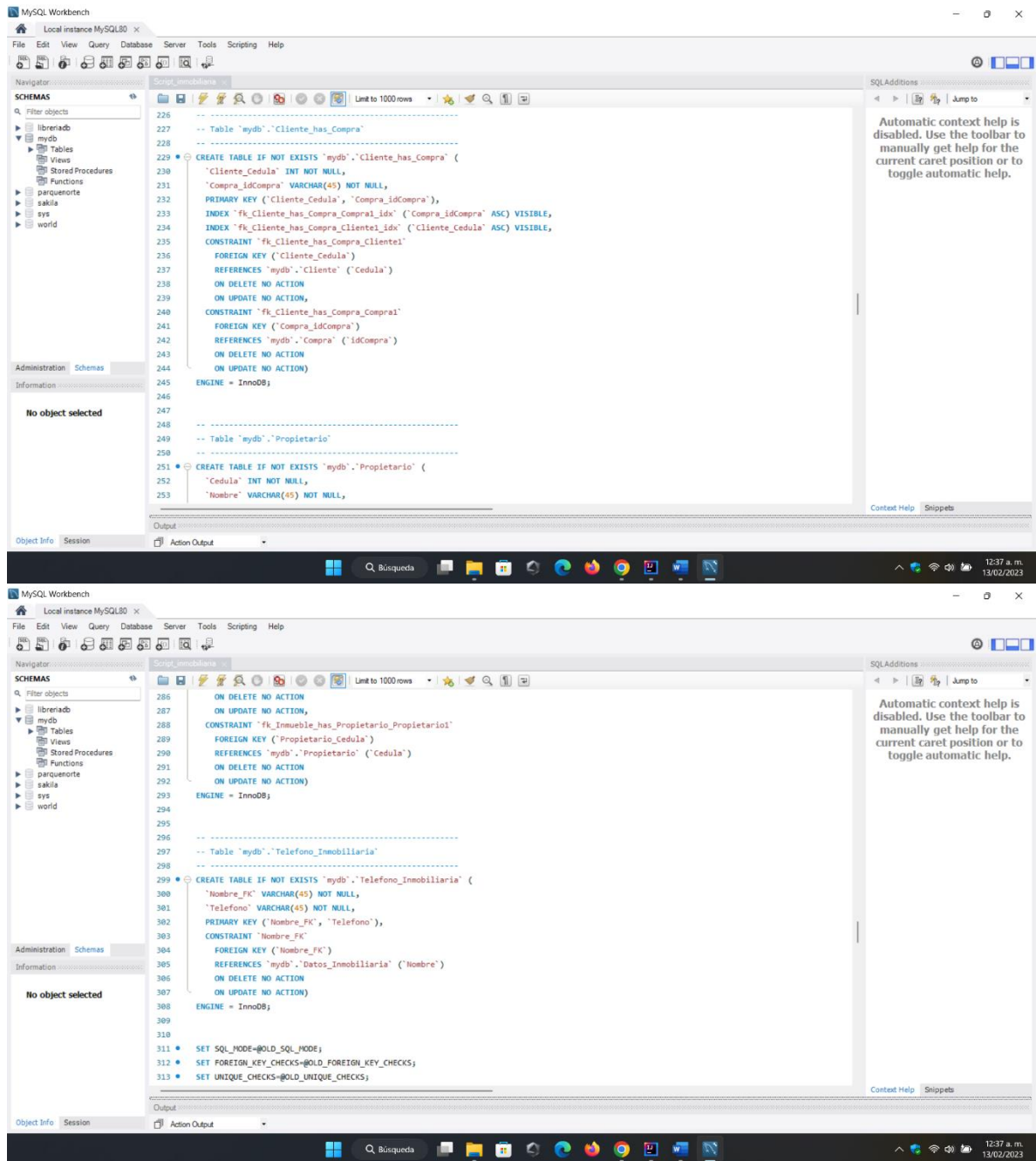
```
1 -- MySQL Workbench Forward Engineering
2
3 SET @OLD_UNIQUE_CHECKS=@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4 SET @OLD_FOREIGN_KEY_CHECKS=@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7 --
8 -- Schema mydb
9 --
10
11 -- Schema mydb
12 --
13
14 CREATE SCHEMA IF NOT EXISTS 'mydb' DEFAULT CHARACTER SET utf8 ;
15 USE 'mydb' ;
16
17 --
18 -- Table 'mydb`.`Cliente`
19 --
20 CREATE TABLE IF NOT EXISTS 'mydb`.`Cliente' (
21   'Cedula' INT NOT NULL,
22   'Nombre' VARCHAR(45) NOT NULL,
23   'CorreoElectronico' VARCHAR(45) NOT NULL,
24   'Ingresos' VARCHAR(45) NOT NULL,
25   'Direccion' VARCHAR(45) NOT NULL,
26   'Habilitado' VARCHAR(45) NOT NULL,
27   PRIMARY KEY ('Cedula'))
28 ENGINE = InnoDB;
```

**Bottom Screenshot:** The script continues with the creation of the 'Telefono' and 'Asesor' tables.

```
31
32 -- Table 'mydb`.`Telefono`
33 --
34 CREATE TABLE IF NOT EXISTS 'mydb`.`Telefono' (
35   'Telefono' VARCHAR(45) NOT NULL,
36   'Cedula_FK' INT NOT NULL,
37   PRIMARY KEY ('Telefono'),
38   INDEX 'Cedula_FK_idx' ('Cedula_FK' ASC) VISIBLE,
39   CONSTRAINT 'Cedula_FK'
40     FOREIGN KEY ('Cedula_FK')
41       REFERENCES 'mydb`.`Cliente' ('Cedula')
42       ON DELETE NO ACTION
43       ON UPDATE NO ACTION)
44 ENGINE = InnoDB;
45
46 --
47 -- Table 'mydb`.`Asesor`
48 --
49
50 CREATE TABLE IF NOT EXISTS 'mydb`.`Asesor' (
51   'Cedula' INT NOT NULL,
52   'Nombre' VARCHAR(45) NOT NULL,
53   'Salario' VARCHAR(45) NOT NULL,
54   'Telefono' VARCHAR(45) NOT NULL,
55   'Correo' VARCHAR(45) NOT NULL,
56   PRIMARY KEY ('Cedula'))
57 ENGINE = InnoDB;
58
```







Luego de esto Procedí a corré el script y empecé a poblar la base de datos:



Aquí muestro los resultados:

Navigator

Filter objects

- libreriaadb
- mydb**
  - Tables
  - Views
  - Stored Procedures
  - Functions
- parquenorte
- sakila
- sys
- world

Script\_inmobiliaria

Limit to 1000 rows

VALUES (111222333, 'Carlos Gonzalez', '55000', '555-555-5557', 'carlosgonzalez@email.com');

INSERT INTO Asesor (Cedula, Nombre, Salario, Telefono, Correo)

VALUES (444221111, 'Ana Martinez', '58000', '555-555-5558', 'anamartinez@email.com');

INSERT INTO Asesor (Cedula, Nombre, Salario, Telefono, Correo)

VALUES (555332211, 'Luis Lopez', '52000', '555-555-5559', 'luislopez@email.com');

INSERT INTO Asesor (Cedula, Nombre, Salario, Telefono, Correo)

VALUES (666111000, 'Sofia Garcia', '56000', '555-555-5560', 'sofiagarcia@email.com');

SELECT \*

FROM Asesor;

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

Cedula	Nombre	Salario	Telefono	Correo
111222333	Carlos Gonzalez	55000	555-555-5557	carlosgonzalez@email.com
123456789	Juan Perez	50000	555-555-5555	juanperez@email.com
444221111	Ana Martinez	58000	555-555-5558	anamartinez@email.com
555332211	Luis Lopez	52000	555-555-5559	luislopez@email.com
666111000	Sofia Garcia	56000	555-555-5560	sofiagarcia@email.com
987654321	Maria Rodriguez	60000	555-555-5556	maria Rodriguez@email.com

No object selected

SCHEMAS

Filter objects

libreriaadb

mydb

Tables

Views

Stored Procedures

Functions

parquenorte

sakila

sys

world

Limit to 1000 rows

Limit to 1000 rows

Limit to 1000 rows

Limit to 1000 rows

Limit to 1000 rows

```

314
315
316 • INSERT INTO Cliente (Cedula, Nombre, CorreoElectronico, Ingresos, Direccion, Habilitado)
317 VALUES
318 (1, 'Juan Pérez', 'juanperez@email.com', '$1000', 'Calle 1 #2-3', 'Sí'),
319 (2, 'María Rodríguez', 'mariaRodriguez@email.com', '$1500', 'Calle 4 #5-6', 'Sí'),
320 (3, 'Pedro García', 'pedrogarcia@email.com', '$2000', 'Calle 7 #8-9', 'Sí'),
321 (4, 'Ana López', 'analopez@email.com', '$2500', 'Calle 10 #11-12', 'Sí'),
322 (5, 'José Martínez', 'josemartinez@email.com', '$3000', 'Calle 13 #14-15', 'Sí'),
323 (6, 'Luz Díaz', 'luzdiaz@email.com', '$3500', 'Calle 16 #17-18', 'Sí');
324 • SELECT *
325 FROM Cliente;
326

```

Result Grid

Filter Rows:

Edit

Export/Imports

Wrap Cell Contents

	Cedula	Nombre	CorreoElectronico	Ingresos	Direccion	Habilitado
▶	1	Juan Pérez	juanperez@email.com	\$1000	Calle 1 #2-3	Sí
	2	María Rodríguez	mariaRodriguez@email.com	\$1500	Calle 4 #5-6	Sí
	3	Pedro García	pedrogarcia@email.com	\$2000	Calle 7 #8-9	Sí
	4	Ana López	analopez@email.com	\$2500	Calle 10 #11-12	Sí
	5	José Martínez	josemartinez@email.com	\$3000	Calle 13 #14-15	Sí
	6	Luz Díaz	luzdiaz@email.com	\$3500	Calle 16 #17-18	Sí
•	NULL	NULL	NULL	NULL	NULL	NULL

Client 1 x

Administration

Schemas

Information

No object selected



Navigator: SCHEMAS

Filter objects

- libreria
- mydb
  - Tables
  - Views
  - Stored Procedures
  - Functions
- parqueno
- sakila
- sys
- world

Administration Schemas

Information

No object selected

Script\_inmobiliaria

```

390 FROM Cliente_has_Solicitud_Visita;
391
392 • INSERT INTO Datos_Inmobiliaria (Nombre, direccion, Correo, paginaWeb)
393 VALUES
394 ('Inmobiliaria 1', 'Calle 1', 'inmobiliaria1@example.com', 'www.inmobiliaria1.com'),
395 ('Inmobiliaria 2', 'Calle 2', 'inmobiliaria2@example.com', 'www.inmobiliaria2.com'),
396 ('Inmobiliaria 3', 'Calle 3', 'inmobiliaria3@example.com', 'www.inmobiliaria3.com'),
397 ('Inmobiliaria 4', 'Calle 4', 'inmobiliaria4@example.com', 'www.inmobiliaria4.com'),
398 ('Inmobiliaria 5', 'Calle 5', 'inmobiliaria5@example.com', 'www.inmobiliaria5.com'),
399 ('Inmobiliaria 6', 'Calle 6', 'inmobiliaria6@example.com', 'www.inmobiliaria6.com');
400 • SELECT *
401 FROM Datos_Inmobiliaria;
402

```

Result Grid

Nombre	direccion	Correo	paginaWeb
Inmobiliaria 1	Calle 1	inmobiliaria1@example.com	www.inmobiliaria1.com
Inmobiliaria 2	Calle 2	inmobiliaria2@example.com	www.inmobiliaria2.com
Inmobiliaria 3	Calle 3	inmobiliaria3@example.com	www.inmobiliaria3.com
Inmobiliaria 4	Calle 4	inmobiliaria4@example.com	www.inmobiliaria4.com
Inmobiliaria 5	Calle 5	inmobiliaria5@example.com	www.inmobiliaria5.com
Inmobiliaria 6	Calle 6	inmobiliaria6@example.com	www.inmobiliaria6.com

Datos\_Inmobiliaria4

Navigator: SCHEMAS

Filter objects

- libreria
- mydb
  - Tables
  - Views
  - Stored Procedures
  - Functions
- parqueno
- sakila
- sys
- world

Administration Schemas

Information

No object selected

Script\_inmobiliaria

```

366 VALUES ('INM003', 'Disponible', 'Semi-Nuevo', '1100', '1900000', '2300000', '4', '120', '2', 'Valle del Cauca', 'Cali', '22222', 'Apto 103')
367
368 • INSERT INTO Inmueble (id_Inmueble, disponibilidad_Inmueble, Estado_inmueble, Precio_alquiler, Precio_venta, Precio_propietario, numero_pisos
369 VALUES ('INM004', 'No Disponible', 'Antiguo', '800', '1600000', '2000000', '1', '80', '4', 'Santander', 'Bucaramanga', '33333', 'Apto 104');
370
371 • INSERT INTO Inmueble (id_Inmueble, disponibilidad_Inmueble, Estado_inmueble, Precio_alquiler, Precio_venta, Precio_propietario, numero_pisos
372 VALUES ('INM005', 'Disponible', 'Nuevo', '1200', '20023', '20090', '5', '80', '1', 'antioquia', '13', '051040', 'Apto 105');
373 • SELECT *
374 FROM Inmueble;
375
376 • INSERT INTO Solicitud_Visita (id_Visita, Fecha_Visita, Asesor_Cedula, Inmueble_id_Inmueble)
377 VALUES
378 ('V001', '2023-02-13', '123456', 'INM0001');

```

Result Grid

id_Inmueble	disponibilidad_Inmueble	Estado_inmueble	Precio_alquiler	Precio_venta	Precio_propietario	numero_pisos	area	estrato	Departamento_inmueble	Comuna
INM002	No Disponible	Usado	900	1800000	2200000	2	90	5	Bogota	Chapinero
INM003	Disponible	Semi-Nuevo	1100	1900000	2300000	4	120	2	Valle del Cauca	Cali
INM004	No Disponible	Antiguo	800	1600000	2000000	1	80	4	Santander	Bucaramanga
INM005	Disponible	Nuevo	1200	20023	20090	5	80	1	antioquia	13
INM001	Disponible	Nuevo	1000	2000000	2500000	3	100	01	Antioquia	Medellin

Inmueble 3

## Preguntas:

### ❖ ¿Qué debería cambiar o agregar para incluir la renovación de contratos de alquiler?

Se podría crear una nueva entidad llamada contrato y en aquella entidad uno de los campos podría ser la duración del contrato y si aquel contrato estaba renovado o no, como un booleano. Por ejemplo, si actualmente tuviéramos una entidad llamada "Contratos de Alquiler" que registrara información sobre los contratos de alquiler y con atributos como: fecha de inicio, fecha de finalización, monto del alquiler entre otras. También se podría agregar un atributo que contenga información sobre las fechas de renovación del contrato así mismo como funcionan las suscripciones a servicios como Netflix o amazon prime. Donde al finalizar la fecha del servicio se hace una renovación automática si el cliente así lo desea.

### ❖ ¿Cómo podría controlar que la misma persona que compra un inmueble con identificador X y se lo entrega a la agencia Inmobiliaria SofkaU, NO pueda alquilar el inmueble con identificador X?

1. Uso de un sistema de registro centralizado: Se puede utilizar un sistema centralizado que registre todas las transacciones inmobiliarias y alquileres en una base de datos. De esta manera, la agencia Inmobiliaria SofkaU puede verificar que un inmueble específico no ha sido alquilado antes de proceder con la transacción.
2. Verificación manual: La agencia Inmobiliaria SofkaU puede llevar a cabo una verificación manual antes de proceder con el alquiler. Esto puede incluir la revisión de documentos relevantes, como contratos de compraventa y alquiler, y la verificación de los registros de transacciones inmobiliarias.