

# Mer Inmobiliaria SofkaU

## Requerimientos:

- Complete **unicamente** las cardinalidades del modelo entidad relacional adjunto (diagrama1.dia).

Se completaron únicamente las cardinalidades.

A continuación, adjunto el diagrama propuesto durante el ejercicio, y el resultado de completar el diagrama.

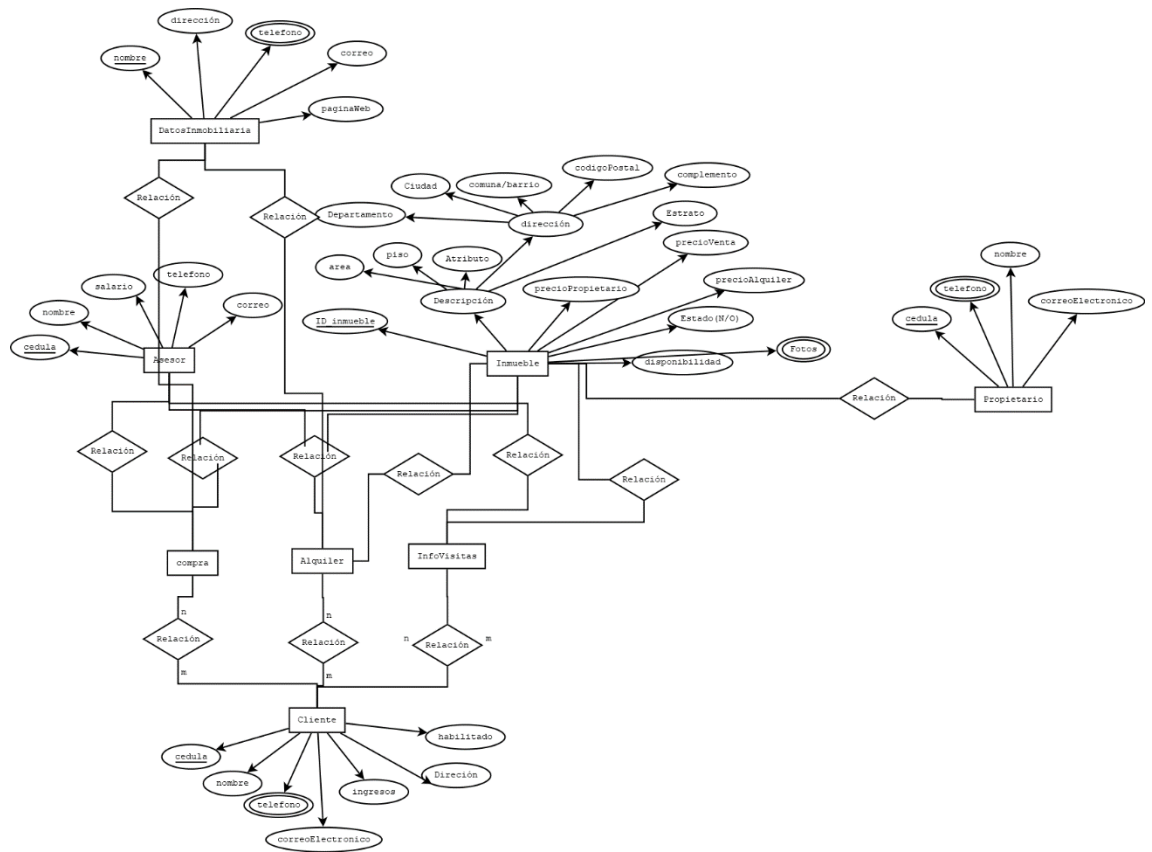
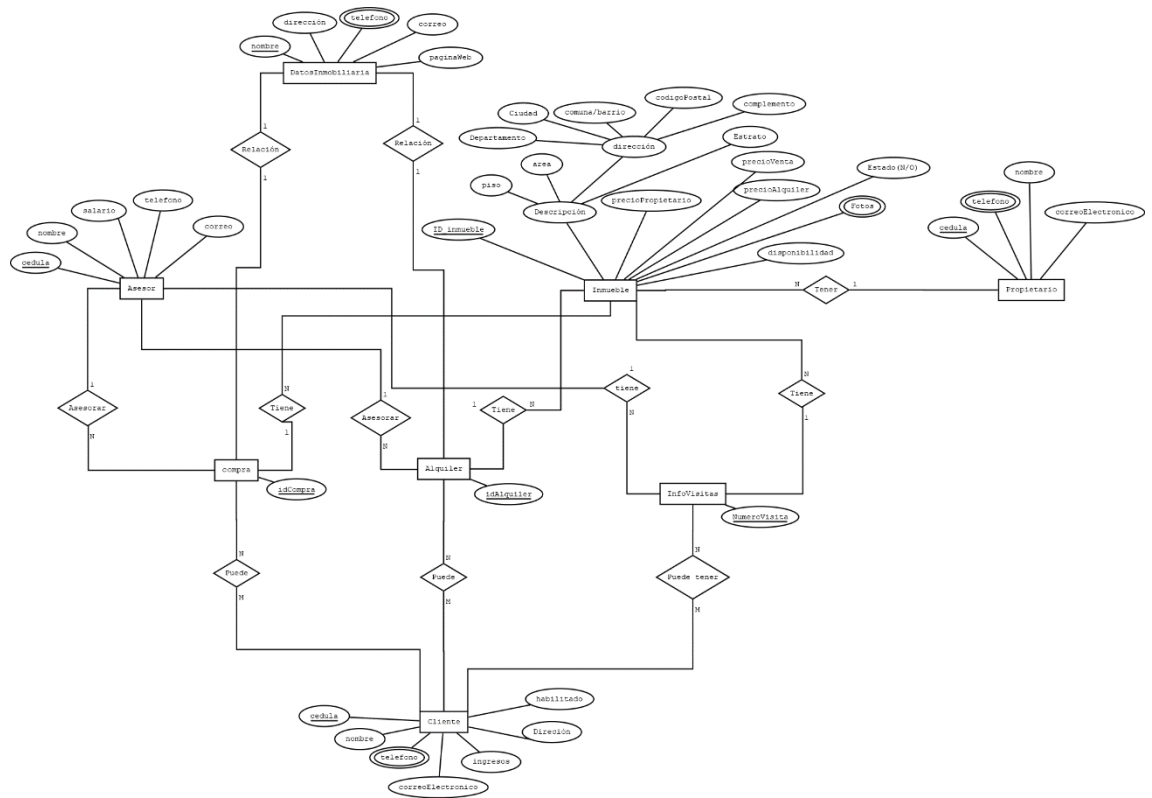


Diagrama ER propuesto inicialmente durante la clase

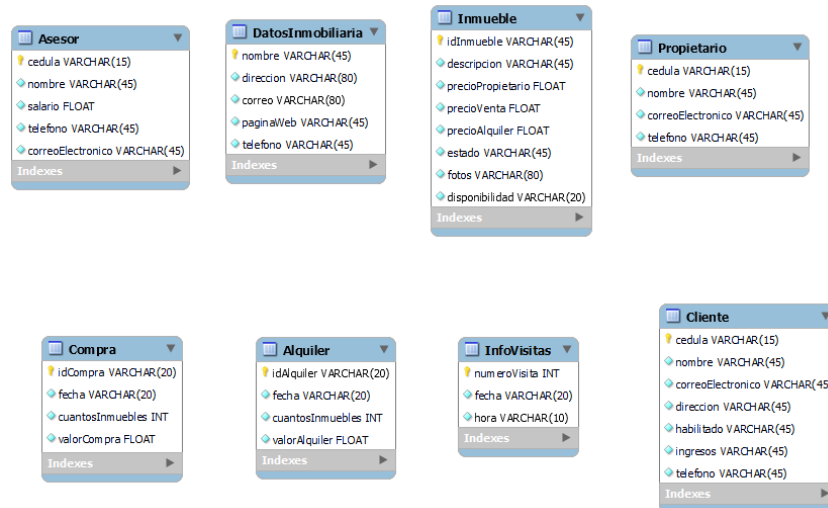


*Resultado del diagrama ER con sus respectivas cardinalidades*

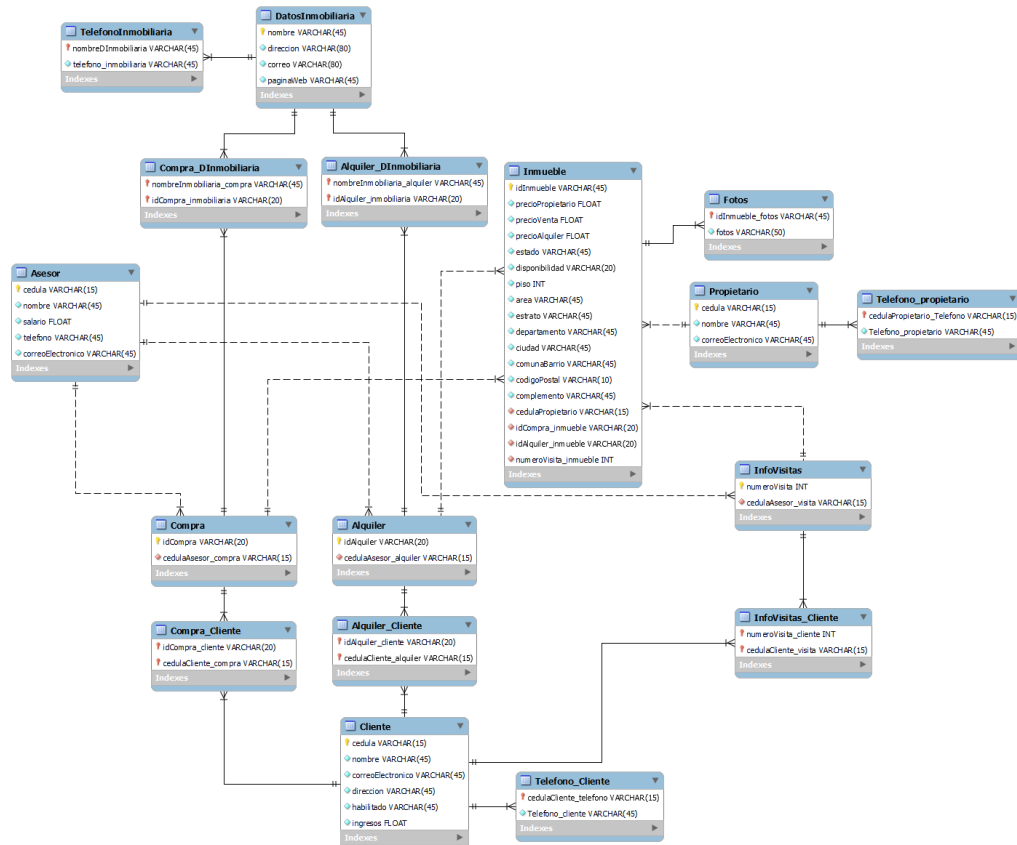
- Un Cliente puede hacer muchas compras, una compra puede ser hecha por muchos clientes.
- Un cliente puede hacer muchos alquileres, un alquiler puede ser hecho por muchos clientes.
- Un cliente puede tener muchas informaciones sobre visitas, la información de visitas puede ser obtenida por muchos clientes.
- Una compra puede ser asesorada por un asesor, un asesor puede asesorar muchas compras.
- Un alquiler puede ser asesorado por un asesor, un asesor puede asesorar muchos alquileres.
- Una información sobre visitas puede ser hecha por un asesor, un asesor puede hacer muchas informaciones sobre visitas.
- Una compra puede tener muchos inmuebles, un inmueble puede tener 1 sola compra a la vez.
- Un alquiler puede tener muchos inmuebles, un inmueble puede tener 1 alquiler a la vez.
- Un inmueble puede tener una información de visita a la vez, una información de visitas puede tener muchos inmuebles.
- Un inmueble puede tener un propietario, un propietario puede tener muchos inmuebles.
- Una compra puede tener 1 datos de inmobiliaria, datos de inmobiliaria puede ser incluida en una sola compra.

- Un alquiler puede tener 1 datos de inmobiliaria, datos de inmobiliaria puede ser incluida en un solo alquiler a la vez.
- **Pase el diagrama a workbench donde incluya todas las cardinalidades que especifíco en el punto anterior.**

Las entidades se convierten en tablas.



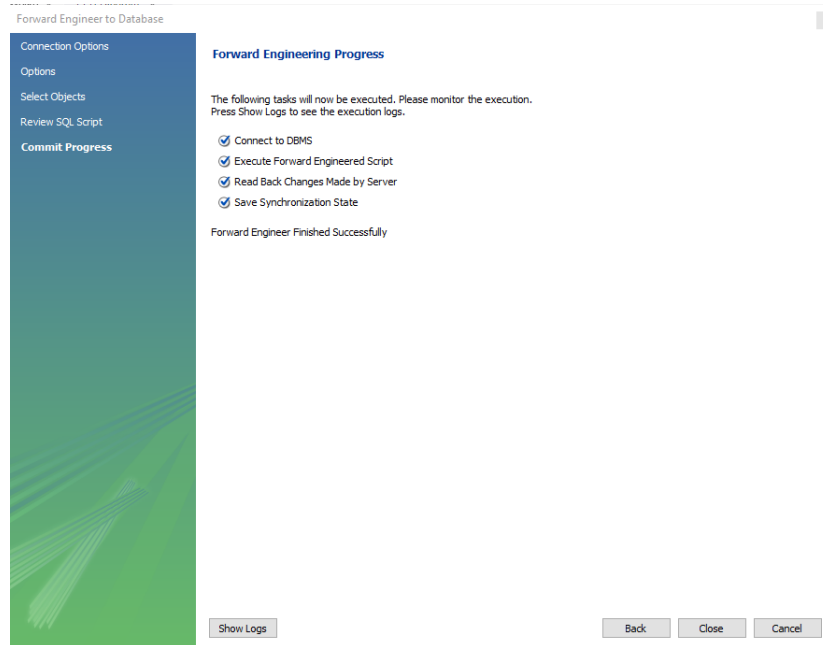
Se tienen en cuenta las cardinalidades y los atributos compuestos



En la imagen anterior se aprecia todas las tablas correspondientes a las entidades, a las agregadas según las cardinalidades, las claves foráneas necesarias para relacionar las tablas.

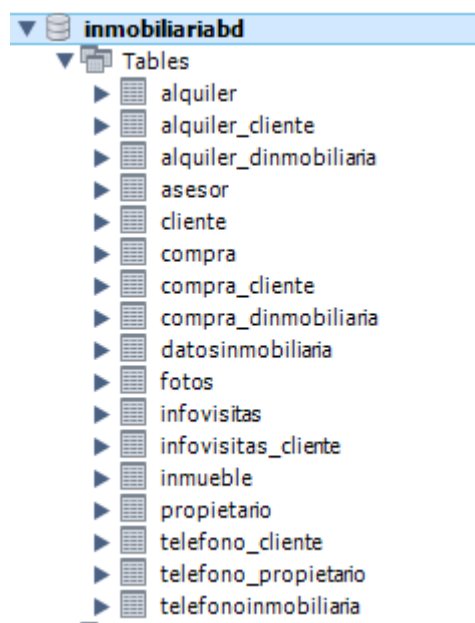
- ***Gerene el Script aplicando ingeniería hacia adelante.***

Se genera el script aplicando la ingeniería hacia adelante.



- ***Corra el Script y genere la base de datos.***

Se corre el Script y se genera la base de datos. Adjunto imágenes del Script y la base de datos creada.



## Jessica Andrea López Obando

```
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 InmobiliariaBD
9  -----
10
11 -----
12 -- Schema InmobiliariaBD
13 -----
14 ● CREATE SCHEMA IF NOT EXISTS `InmobiliariaBD` DEFAULT CHARACTER SET utf8 ;
15 ● USE `InmobiliariaBD` ;
16
17 -----
18 -- Table `InmobiliariaBD`.`Propietario`
19 -----
20 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Propietario` (
21   `cedula` VARCHAR(15) NOT NULL,
22   `nombre` VARCHAR(45) NOT NULL,
23   `correoElectronico` VARCHAR(45) NOT NULL,
24   PRIMARY KEY (`cedula`))
25   ENGINE = InnoDB;
26
27 -----
28 -- Table `InmobiliariaBD`.`Asesor`
29 -----
30
31 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Asesor` (
32   `cedula` VARCHAR(15) NOT NULL,
33   `nombre` VARCHAR(45) NOT NULL,
34   `salario` FLOAT NOT NULL,
35   `telefono` VARCHAR(45) NOT NULL,
36   `correoElectronico` VARCHAR(45) NOT NULL,
37   PRIMARY KEY (`cedula`))
38   ENGINE = InnoDB;
39
40 -----
41
42 -- Table `InmobiliariaBD`.`Compra`
43 -----
44 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Compra` (
45   `idCompra` VARCHAR(20) NOT NULL,
46   `cedulaAsesor_compra` VARCHAR(15) NOT NULL,
47   PRIMARY KEY (`idCompra`),
48   INDEX `cedulaAsesor_compra_idx` (`cedulaAsesor_compra` ASC) VISIBLE,
49   CONSTRAINT `cedulaAsesor_compra`
50     FOREIGN KEY (`cedulaAsesor_compra`)
51     REFERENCES `InmobiliariaBD`.`Asesor` (`cedula`)
52     ON DELETE NO ACTION
53     ON UPDATE NO ACTION)
54   ENGINE = InnoDB;
55
56 -----
57
58 -- Table `InmobiliariaBD`.`Alquiler`
59 -----
60 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Alquiler` (
61   `idAlquiler` VARCHAR(20) NOT NULL,
62   `cedulaAsesor_alquiler` VARCHAR(15) NOT NULL,
63   PRIMARY KEY (`idAlquiler`),
64   INDEX `cedulaAsesor_alquiler_idx` (`cedulaAsesor_alquiler` ASC) VISIBLE,
65   CONSTRAINT `cedulaAsesor_alquiler`
66     FOREIGN KEY (`cedulaAsesor_alquiler`)
67     REFERENCES `InmobiliariaBD`.`Asesor` (`cedula`)
68     ON DELETE NO ACTION
69     ON UPDATE NO ACTION)
70   ENGINE = InnoDB;
71
72 -----
73
74 -- Table `InmobiliariaBD`.`Infovisitas`
75 -----
76 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Infovisitas` (
77   `numeroVisita` INT NOT NULL,
78   `cedulaAsesor_visita` VARCHAR(15) NOT NULL,
79   PRIMARY KEY (`numeroVisita`),
80   INDEX `cedulaAsesor_visita_idx` (`cedulaAsesor_visita` ASC) VISIBLE,
81   CONSTRAINT `cedulaAsesor_visita`
82     FOREIGN KEY (`cedulaAsesor_visita`)
83     REFERENCES `InmobiliariaBD`.`Asesor` (`cedula`)
84     ON DELETE NO ACTION
85     ON UPDATE NO ACTION)
86   ENGINE = InnoDB;
87
88 -----
```

```

89  -----
90  -- Table `InmobiliariaBD`.`Inmueble`
91  -----
92  ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Inmueble` (
93      `idInmueble` VARCHAR(45) NOT NULL,
94      `precioPropietario` FLOAT NOT NULL,
95      `precioVenta` FLOAT NOT NULL,
96      `precioAlquiler` FLOAT NOT NULL,
97      `estado` VARCHAR(45) NOT NULL,
98      `disponibilidad` VARCHAR(20) NOT NULL,
99      `piso` INT NOT NULL,
100     `area` VARCHAR(45) NOT NULL,
101     `estrato` VARCHAR(45) NOT NULL,
102     `departamento` VARCHAR(45) NOT NULL,
103     `ciudad` VARCHAR(45) NOT NULL,
104     `comunaBarrio` VARCHAR(45) NOT NULL,
105     `codigoPostal` VARCHAR(10) NOT NULL,
106     `complemento` VARCHAR(45) NOT NULL,
107     `cedulaPropietario` VARCHAR(15) NOT NULL,
108     `idCompra_inmueble` VARCHAR(20) NOT NULL,
109     `idAlquiler_inmueble` VARCHAR(20) NOT NULL,
110     `numeroVisita_inmueble` INT NOT NULL,
111     PRIMARY KEY (`idInmueble`),
112     INDEX `cedulaPropietario_idx` (`cedulaPropietario` ASC) VISIBLE,
113     INDEX `idCompra_inmueble_idx` (`idCompra_inmueble` ASC) VISIBLE,
114     INDEX `idAlquiler_inmueble_idx` (`idAlquiler_inmueble` ASC) VISIBLE,
115     INDEX `numeroVisita_inmueble_idx` (`numeroVisita_inmueble` ASC) VISIBLE,
116     CONSTRAINT `cedulaPropietario`
117         FOREIGN KEY (`cedulaPropietario`)
118         REFERENCES `InmobiliariaBD`.`Propietario` (`cedula`)
119         ON DELETE NO ACTION
120         ON UPDATE NO ACTION,
121     CONSTRAINT `idCompra_inmueble`
122         FOREIGN KEY (`idCompra_inmueble`)
123         REFERENCES `InmobiliariaBD`.`Compra` (`idCompra`)
124         ON DELETE NO ACTION
125         ON UPDATE NO ACTION,
126     CONSTRAINT `idAlquiler_inmueble`
127         FOREIGN KEY (`idAlquiler_inmueble`)
128         REFERENCES `InmobiliariaBD`.`Alquiler` (`idAlquiler`)
129         ON DELETE NO ACTION
130         ON UPDATE NO ACTION,
131     CONSTRAINT `numeroVisita_inmueble`
132         FOREIGN KEY (`numeroVisita_inmueble`)
133         REFERENCES `InmobiliariaBD`.`InfoVisitas` (`numeroVisita`)
134         ON DELETE NO ACTION
135         ON UPDATE NO ACTION)
136  ENGINE = InnoDB;

137
138
139  -----
140  -- Table `InmobiliariaBD`.`DatosInmobiliaria`
141  -----
142  ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`DatosInmobiliaria` (
143      `nombre` VARCHAR(45) NOT NULL,
144      `direccion` VARCHAR(80) NOT NULL,
145      `correo` VARCHAR(80) NOT NULL,
146      `paginaWeb` VARCHAR(45) NOT NULL,
147      PRIMARY KEY (`nombre`))
148  ENGINE = InnoDB;

149
150
151  -----
152  -- Table `InmobiliariaBD`.`Cliente`
153  -----
154  ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Cliente` (
155      `cedula` VARCHAR(15) NOT NULL,
156      `nombre` VARCHAR(45) NOT NULL,
157      `correoElectronico` VARCHAR(45) NOT NULL,
158      `direccion` VARCHAR(45) NOT NULL,
159      `habilitado` VARCHAR(45) NOT NULL,
160      `ingresos` FLOAT NOT NULL,
161      PRIMARY KEY (`cedula`))
162  ENGINE = InnoDB;

163
164
165  -----
166  -- Table `InmobiliariaBD`.`Fotos`
167  -----
168  ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Fotos` (
169      `idInmueble_fotos` VARCHAR(45) NOT NULL,
170      `fotos` VARCHAR(50) NOT NULL,
171      PRIMARY KEY (`idInmueble_fotos`),
172      CONSTRAINT `idInmuebleFotos`
173          FOREIGN KEY (`idInmueble_fotos`)
174          REFERENCES `InmobiliariaBD`.`Inmueble` (`idInmueble`)
175          ON DELETE NO ACTION
176          ON UPDATE NO ACTION)
177  ENGINE = InnoDB;
178
179

```

```

179
180 -----
181 -- Table `InmobiliariaBD`.`Telefono_propietario`
182 -----
183 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Telefono_propietario` (
184     `cedulaPropietario_Telefono` VARCHAR(15) NOT NULL,
185     `Telefono_propietario` VARCHAR(45) NOT NULL,
186     PRIMARY KEY (`cedulaPropietario_Telefono`),
187     CONSTRAINT `cedulaPropietario_telefono`
188     FOREIGN KEY (`cedulaPropietario_Telefono`)
189     REFERENCES `InmobiliariaBD`.`Propietario` (`cedula`)
190     ON DELETE NO ACTION
191     ON UPDATE NO ACTION)
192 ENGINE = InnoDB;
193
194 -----
195 -- Table `InmobiliariaBD`.`Compra_Cliente`
196 -----
197
198 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Compra_Cliente` (
199     `idCompra_cliente` VARCHAR(20) NOT NULL,
200     `cedulaCliente_compra` VARCHAR(15) NOT NULL,
201     PRIMARY KEY (`idCompra_cliente`, `cedulaCliente_compra`),
202     INDEX `cedulaCliente_compra_idx` (`cedulaCliente_compra` ASC) VISIBLE,
203     CONSTRAINT `idCompra_cliente`
204     FOREIGN KEY (`idCompra_cliente`)
205     REFERENCES `InmobiliariaBD`.`Compra` (`idCompra`)
206     ON DELETE NO ACTION
207     ON UPDATE NO ACTION,
208     CONSTRAINT `cedulaCliente_compra`
209     FOREIGN KEY (`cedulaCliente_compra`)
210     REFERENCES `InmobiliariaBD`.`Cliente` (`cedula`)
211     ON DELETE NO ACTION
212     ON UPDATE NO ACTION)
213 ENGINE = InnoDB;
214
215 -----
216 -- Table `InmobiliariaBD`.`Alquiler_Cliente`
217 -----
218
219 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Alquiler_Cliente` (
220     `idAlquiler_cliente` VARCHAR(20) NOT NULL,
221     `cedulaCliente_alquiler` VARCHAR(15) NOT NULL,
222     PRIMARY KEY (`idAlquiler_cliente`, `cedulaCliente_alquiler`),
223     INDEX `cedulaCliente_alquiler_idx` (`cedulaCliente_alquiler` ASC) VISIBLE,
224     CONSTRAINT `idAlquiler_cliente`
225     FOREIGN KEY (`idAlquiler_cliente`)
226     REFERENCES `InmobiliariaBD`.`Alquiler` (`idAlquiler`)
227     ON DELETE NO ACTION
228     ON UPDATE NO ACTION,
229     CONSTRAINT `cedulaCliente_alquiler`
230     FOREIGN KEY (`cedulaCliente_alquiler`)
231     REFERENCES `InmobiliariaBD`.`Cliente` (`cedula`)
232     ON DELETE NO ACTION
233     ON UPDATE NO ACTION)
234 ENGINE = InnoDB;
235
236 -----
237 -- Table `InmobiliariaBD`.`InfoVisitas_Cliente`
238 -----
239
240 ● CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`InfoVisitas_cliente` (
241     `numeroVisita_cliente` INT NOT NULL,
242     `cedulaCliente_visita` VARCHAR(15) NOT NULL,
243     PRIMARY KEY (`numeroVisita_cliente`, `cedulaCliente_visita`),
244     INDEX `cedulaCliente_visita_idx` (`cedulaCliente_visita` ASC) VISIBLE,
245     CONSTRAINT `numeroVisita_cliente`
246     FOREIGN KEY (`numeroVisita_cliente`)
247     REFERENCES `InmobiliariaBD`.`InfoVisitas` (`numeroVisita`)
248     ON DELETE NO ACTION
249     ON UPDATE NO ACTION,
250     CONSTRAINT `cedulaCliente_visita`
251     FOREIGN KEY (`cedulaCliente_visita`)
252     REFERENCES `InmobiliariaBD`.`Cliente` (`cedula`)
253     ON DELETE NO ACTION
254     ON UPDATE NO ACTION)
255 ENGINE = InnoDB;
256
257

```

```

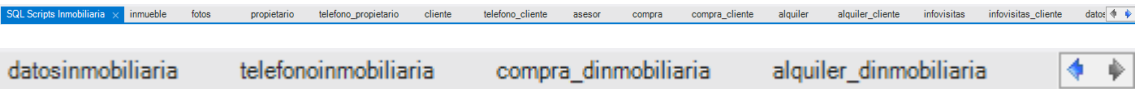
258 -----
259 -- Table `InmobiliariaBD`.`Telefono_Cliente`
260 -----
261 ● ○ CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Telefono_Cliente` (
262     `cedulaCliente_telefono` VARCHAR(15) NOT NULL,
263     `telefono_cliente` VARCHAR(45) NOT NULL,
264     PRIMARY KEY (`cedulaCliente_telefono`),
265     CONSTRAINT `cedulaClienteTelefono`
266         FOREIGN KEY (`cedulaCliente_telefono`)
267         REFERENCES `InmobiliariaBD`.`Cliente` (`cedula`)
268         ON DELETE NO ACTION
269         ON UPDATE NO ACTION)
270 ENGINE = InnoDB;
271
272
273 -----
274 -- Table `InmobiliariaBD`.`Compra_DInmobiliaria`
275 -----
276 ● ○ CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Compra_DInmobiliaria` (
277     `nombreInmobiliaria_compra` VARCHAR(45) NOT NULL,
278     `idCompra_inmobiliaria` VARCHAR(20) NOT NULL,
279     PRIMARY KEY (`nombreInmobiliaria_compra`, `idCompra_inmobiliaria`),
280     INDEX `idCompra_inmobiliaria_idx` (`idCompra_inmobiliaria` ASC) VISIBLE,
281     CONSTRAINT `nombreInmobiliaria_compra`
282         FOREIGN KEY (`nombreInmobiliaria_compra`)
283         REFERENCES `InmobiliariaBD`.`DatosInmobiliaria` (`nombre`)
284         ON DELETE NO ACTION
285         ON UPDATE NO ACTION,
286     CONSTRAINT `idCompra_inmobiliaria`
287         FOREIGN KEY (`idCompra_inmobiliaria`)
288         REFERENCES `InmobiliariaBD`.`Compra` (`idCompra`)
289         ON DELETE NO ACTION
290         ON UPDATE NO ACTION)
291 ENGINE = InnoDB;
292
293
294 -----
295 -- Table `InmobiliariaBD`.`Alquiler_DInmobiliaria`
296 -----
297 ● ○ CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`Alquiler_DInmobiliaria` (
298     `nombreInmobiliaria_alquiler` VARCHAR(45) NOT NULL,
299     `idAlquiler_inmobiliaria` VARCHAR(20) NOT NULL,
300     PRIMARY KEY (`nombreInmobiliaria_alquiler`, `idAlquiler_inmobiliaria`),
301     INDEX `idAlquiler_inmobiliaria_idx` (`idAlquiler_inmobiliaria` ASC) VISIBLE,
302     CONSTRAINT `nombreInmobiliaria_alquiler`
303         FOREIGN KEY (`nombreInmobiliaria_alquiler`)
304         REFERENCES `InmobiliariaBD`.`DatosInmobiliaria` (`nombre`)
305         ON DELETE NO ACTION
306         ON UPDATE NO ACTION,
307     CONSTRAINT `idAlquiler_inmobiliaria`
308         FOREIGN KEY (`idAlquiler_inmobiliaria`)
309         REFERENCES `InmobiliariaBD`.`Alquiler` (`idAlquiler`)
310         ON DELETE NO ACTION
311         ON UPDATE NO ACTION)
312 ENGINE = InnoDB;
313
314
315 -----
316 -- Table `InmobiliariaBD`.`TelefonoInmobiliaria`
317 -----
318 ● ○ CREATE TABLE IF NOT EXISTS `InmobiliariaBD`.`TelefonoInmobiliaria` (
319     `nombreDInmobiliaria` VARCHAR(45) NOT NULL,
320     `telefono_inmobiliaria` VARCHAR(45) NOT NULL,
321     PRIMARY KEY (`nombreDInmobiliaria`),
322     CONSTRAINT `nombreDInmobiliaria`
323         FOREIGN KEY (`nombreDInmobiliaria`)
324         REFERENCES `InmobiliariaBD`.`DatosInmobiliaria` (`nombre`)
325         ON DELETE NO ACTION
326         ON UPDATE NO ACTION)
327 ENGINE = InnoDB;
328
329
330 ● SET SQL_MODE=@OLD_SQL_MODE;
331 ● SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
332 ● SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
333

```

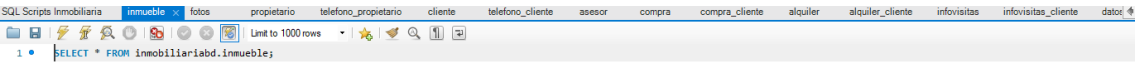


- **Poblar todas las tabla de la base de datos al menos con 5 registros.**

Se pueblan las tablas de la base de datos con 5 registros



**Inmueble**

A screenshot of the 'Result Grid' showing the data returned by the SQL query. It displays 5 rows of property records with various attributes like id, price, status, and location.

idInmueble	precioPropietario	precioVenta	precioAlquiler	estado	disponibilidad	piso	area	estrato	departam	ciudad	comunaBarrio	codigoPostal	complemento	cedulaPropietario	idCompra_inmueble	idAlquiler_inmueble	numeroVisita
0000001	120000000	125000000	600000	usado	inmediata	1	60m2	3	Quindio	Armenia	Casa Blanca	630002	calle 30 - 12	739548	00001	00001	1
0000002	350000000	355000000	850000	nuevo	inmediata	2	80m2	5	Quindio	Armenia	Iberica	630002	carrera 19 -25	781112	00002	00002	2
0000003	980000000	103000000	560000	usado	inmediata	5	40m2	2	Quindio	Armenia	Isabela	630002	calle 30 - 30	1092635487	00003	00003	3
0000004	550000000	560000000	1200000	nuevo	inmediata	6	102m2	6	Quindio	Armenia	Los Profesionales	630002	carrera 19 -19	3189245	00004	00004	4
0000005	1000000000	1100000000	4500000	usado	inmediata	1	1000m2	6	Quindio	Pueblo Ta...	Prodesa	633008	via. Las pal...	1896450	00005	00005	5

**Fotos**

A screenshot of the 'Result Grid' showing 5 rows of photo records. Each row contains an id and a description of the photo content.

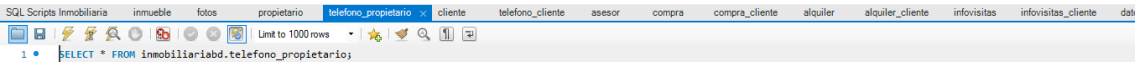
idInmueble_fotos	fotos
0000001	sala, comedor, cocina, baño(s), habitaciones
0000002	sala, comedor, cocina, baño(s), habitaciones
0000003	sala, comedor, cocina, baño(s), habitaciones
0000004	sala, comedor, cocina, baño(s), habitaciones
0000005	sala, comedor, cocina, baño(s), habitaciones

**Propietario**

A screenshot of the 'Result Grid' showing 5 rows of owner records with fields for cedula, name, and email.

cedula	nombre	correoElectronico
1092635487	Laureano Espino	laureanoespino@outlook.com
1896450	Leonardo Fernández	leufer@hotmail.com
3189245	Carla Diez	carladedez@gmail.com
739548	Veronica Alamo	veronicaalamo@gmail.com
781112	Tatiana Sousa	tatianasousa@gmail.com

**Teléfono propietario**

A screenshot of the 'Result Grid' showing 5 rows of phone numbers linked to specific cedulas.

cedulaPropietario_Telefono	Telefono_propietario
1092635487	3103810521
1896450	3102400596
3189245	3114444569
739548	3124896547
781112	3141002153

**Cliente**

## Jessica Andrea López Obando

SQL Scripts Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1 • <code>SELECT * FROM inmobiliariabd.cliente;</code>													

cedula	nombre	correoElectronico	direccion	habilitado	ingresos
1080567410	Bernardino Viera	viera123@outlook.com	Maria Cristina etapa 1 bl 11 apto ...	Si	2500000
1094928323	Juan Andres ...	juan1landrea@gmail.com	Las Colinas casa F # 14	Si	1160000
1256317488	Gustavo Molina	molinagustavo@gmail.com	Alfonso Lopez casa h # 22	Si	1800000
712485	Edgar Tomas	edgartomas@gmail.com	San Juan de Loma bl 12 apto 701	Si	4500000
763895	Delfina Pino	delfinapino1960@hotmail.c...	La Castellana casa e # 2	Si	5700000

## Telefono cliente

SQL Scripts Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1 • <code>SELECT * FROM inmobiliariabd.telefono_cliente;</code>													

cedulaCliente_telefono	Telefono_cliente
1080567410	3107965848
1094928323	3205142386
1256317488	3115221011
712485	3001459647
763895	3002200335

## Asesor

SQL Scripts Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1 • <code>SELECT * FROM inmobiliariabd.asesor;</code>													

cedula	nombre	salario	telefono	correoElectronico
1001523547	Saida Raya	1160000	3124123548	asesor3@inmueblesraulmeja.com
1077452113	Felsa Porras	1160000	3217770512	asesor2@inmueblesraulmeja.com
1122653120	Aya Barros	1160000	3137962485	asesor1@inmueblesraulmeja.com
1896425	Adrian Reina	1160000	3207568996	asesor4@inmueblesraulmeja.com
3041117	Paz Godoy	1160000	3124111102	asesor5@inmueblesraulmeja.com

## Compra

SQL Scripts Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1 • <code>SELECT * FROM inmobiliariabd.compra;</code>													

idCompra	cedulaAsesor_compra
00001	1001523547
00002	1077452113
00003	1122653120
00004	1896425
00005	3041117

## Compra cliente

SQL Scripts Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1 • <code>SELECT * FROM inmobiliariabd.compra_cliente;</code>													

idCompra_cliente	cedulaCliente_compra
00001	1080567410
00002	1094928323
00003	1256317488
00004	712485
00005	763895

Alquiler

SQL Scripts	Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1	SELECT * FROM inmobiliariaabd.alquiler;													

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	idAlquiler	cedulaAsesor_alquiler			
▶	00001	1001523547			
	00002	1077452113			
	00003	1122653120			
	00004	1896425			
	00005	3041117			

Alquiler cliente

SQL Scripts	Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1	SELECT * FROM inmobiliariaabd.alquiler_cliente;													

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	idAlquiler_cliente	cedulaCliente_alquiler			
	00001	1080567410			
	00002	1094928323			
	00003	1256317488			
	00004	712485			
▶	00005	763895			

Infovisitas

SQL Scripts	Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1	SELECT * FROM inmobiliariaabd.infovisitas;													

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	numeroVisita	cedulaAsesor_visita			
	1	1001523547			
	2	1077452113			
	3	1122653120			
	4	1896425			
▶	5	3041117			

Infovisitas cliente

SQL Scripts	Inmobiliaria	inmueble	fotos	propietario	telefono_propietario	cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente
1	SELECT * FROM inmobiliariaabd.infovisitas_cliente;													

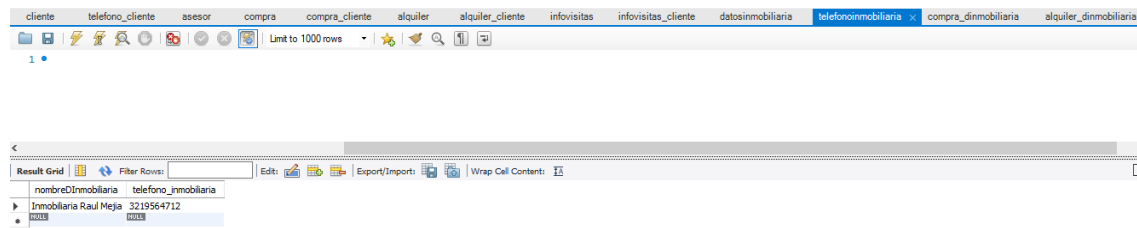
Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	numeroVisita_cliente	cedulaCliente_visita			
	1	1080567410			
	2	1094928323			
	3	1256317488			
	4	712485			
▶	5	763895			

Datosinmobiliaria

cliente	telefono_cliente	asesor	compra	compra_cliente	alquiler	alquiler_cliente	infovisitas	infovisitas_cliente	datosinmobiliaria	telefonoinmobiliaria	compra_dinmobiliaria	alquiler_dinmobiliaria
1	SELECT * FROM inmobiliariaabd.datosinmobiliaria;											

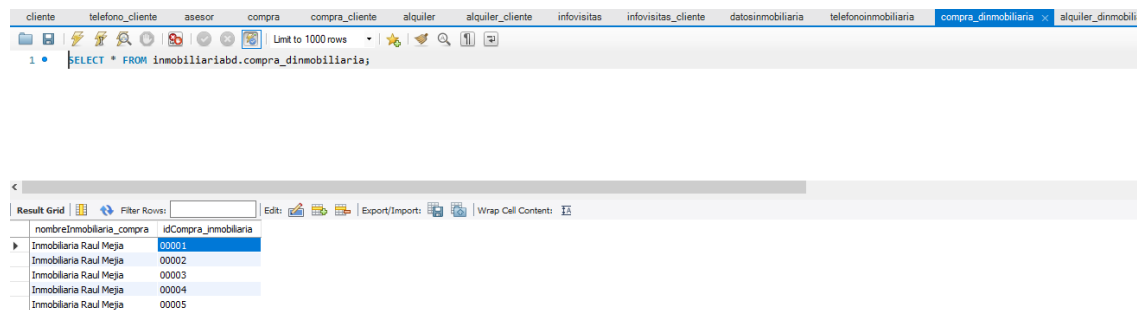
Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	nombre	direccion	correo	paginaWeb	
▶	Inmobiliaria Raul Mejia	Cra 14 # 22-23	inmobiliariaraulmejia@raulmejia.com	www.inmobiliariaraulmejia.com	

### Telefonoinmobiliaria



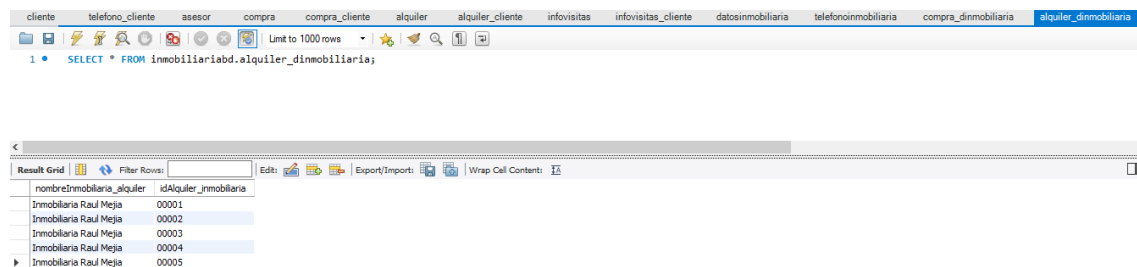
nombreInmobiliaria	telefono_inmobiliaria
Inmobiliaria Raul Mejia	3219564712

### Compra dinmobiliaria



nombreInmobiliaria_compra	idCompra_inmobiliaria
Inmobiliaria Raul Mejia	00001
Inmobiliaria Raul Mejia	00002
Inmobiliaria Raul Mejia	00003
Inmobiliaria Raul Mejia	00004
Inmobiliaria Raul Mejia	00005

### Alquilerdinmobiliaria



nombreInmobiliaria_alquiler	idAlquiler_inmobiliaria
Inmobiliaria Raul Mejia	00001
Inmobiliaria Raul Mejia	00002
Inmobiliaria Raul Mejia	00003
Inmobiliaria Raul Mejia	00004
Inmobiliaria Raul Mejia	00005

- **Después de haber realizado todo lo anterior mencionado responda:**

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

- Crearía una entidad que se llame contrato, esta entidad tendría como atributos Id del contrato, fecha de inicio NOT NULL, fecha de vencimiento NOT NULL, disponibilidad para renovar como booleano, siendo true que se renovará, siendo false que no se renovará.
- La relación sería entre la entidad alquiler y contrato, la cardinalidad sería:
- Un alquiler solo puede tener un contrato, un contrato puede tener muchos alquileres, N:1
- Siendo alquiler quien recibiría como FK el id del contrato

**¿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?***

Teniendo en cuenta la pregunta anterior lo que se comprende es que el problema consiste en controlar que una persona que haya entregado un inmueble a una agencia inmobiliaria (en este caso la agencia inmobiliaria "SofkaU") no pueda alquilar el mismo inmueble en otro lugar. Esto se desea controlar en el sistema para asegurarse de que un inmueble sólo sea alquilado una vez, y para evitar cualquier situación en la que una persona pueda obtener ganancias ilícitas de un mismo inmueble.

Para solucionar considero que en la tabla inmueble podría agregar un atributo disponible en la inmobiliaria not null de tipo booleano, para aclarar si el inmueble sigue estando habilitado o no para ser alquilado o comprado, cualquiera que ingrese a esa base de datos sabrá si el inmueble sigue en registro en la inmobiliaria y de esta forma se llevaría el control

**Genere un archivo PDF con la documentación clara que indique el paso a paso de como realizo la solución del taller y las respuestas de las dos preguntas anteriores.**

- Emplee el uso adecuado de los commits.