

Solución Taller BD 4

Después de organizar nuevamente la estructura del MER propuesta, se renombran algunos atributos para que sean atómicos directamente. Además, se añaden las cardinalidades faltantes siguiendo una lógica subjetiva bajo el contexto de la Inmobiliaria SofkaU, obteniendo así el MER de la Figura 1.

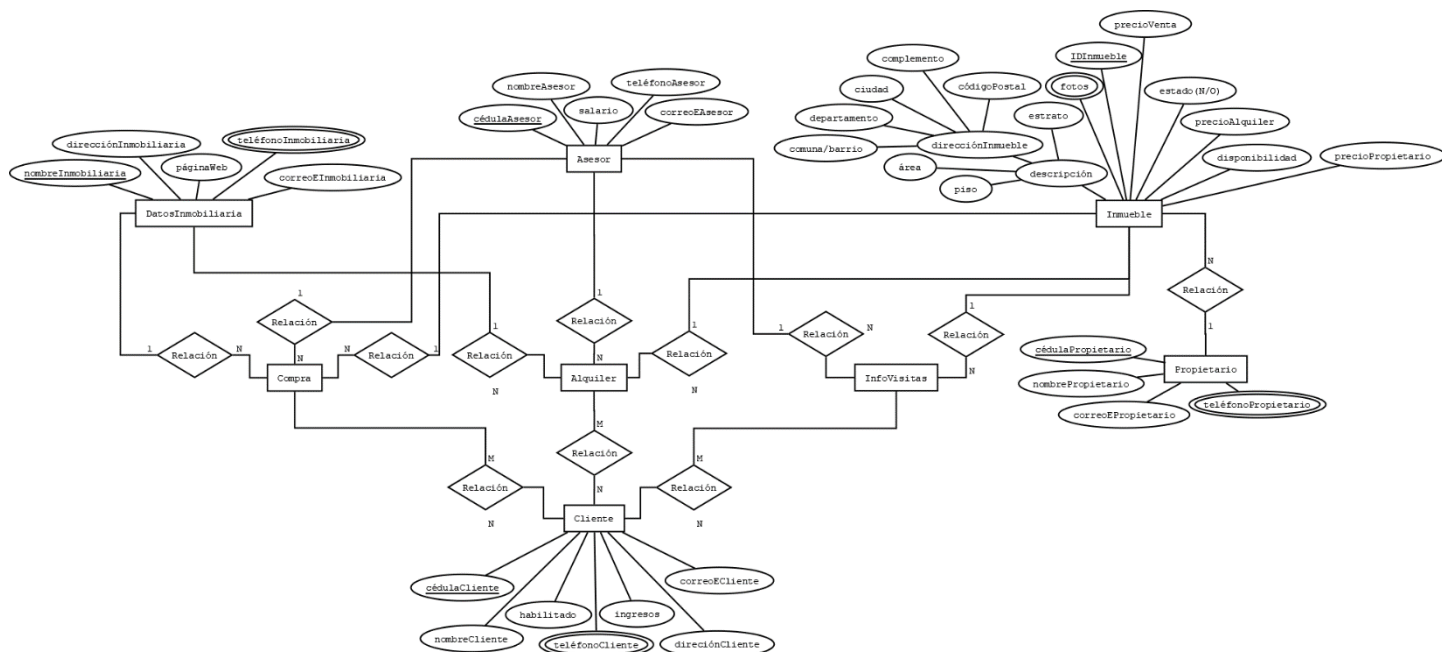


Figura 1 MER con cardinalidades completas.

Las cardinalidades especificadas corresponden a las relaciones entre las entidades de Compra, Alquiler e InfoVisitas, con las entidades DatosInmobiliaria, Asesor e Inmueble; además de la relación entre Inmueble y Propietario. Según la lógica subjetiva que se aplicó, todas estas relaciones son de Uno a Muchos (1:N) y se tomará como ejemplo general para todas las relaciones la siguiente explicación de una de estas. Una Compra puede ser realizada a un solo Inmueble, y un Inmueble se le pueden realizar muchas Compras.

Luego, se hace una transformación del MER a un MR directamente en el software Workbench, donde después de aplicar las reglas para transformar las cardinalidades, se obtiene el MR que se aprecia en la Figura 2.

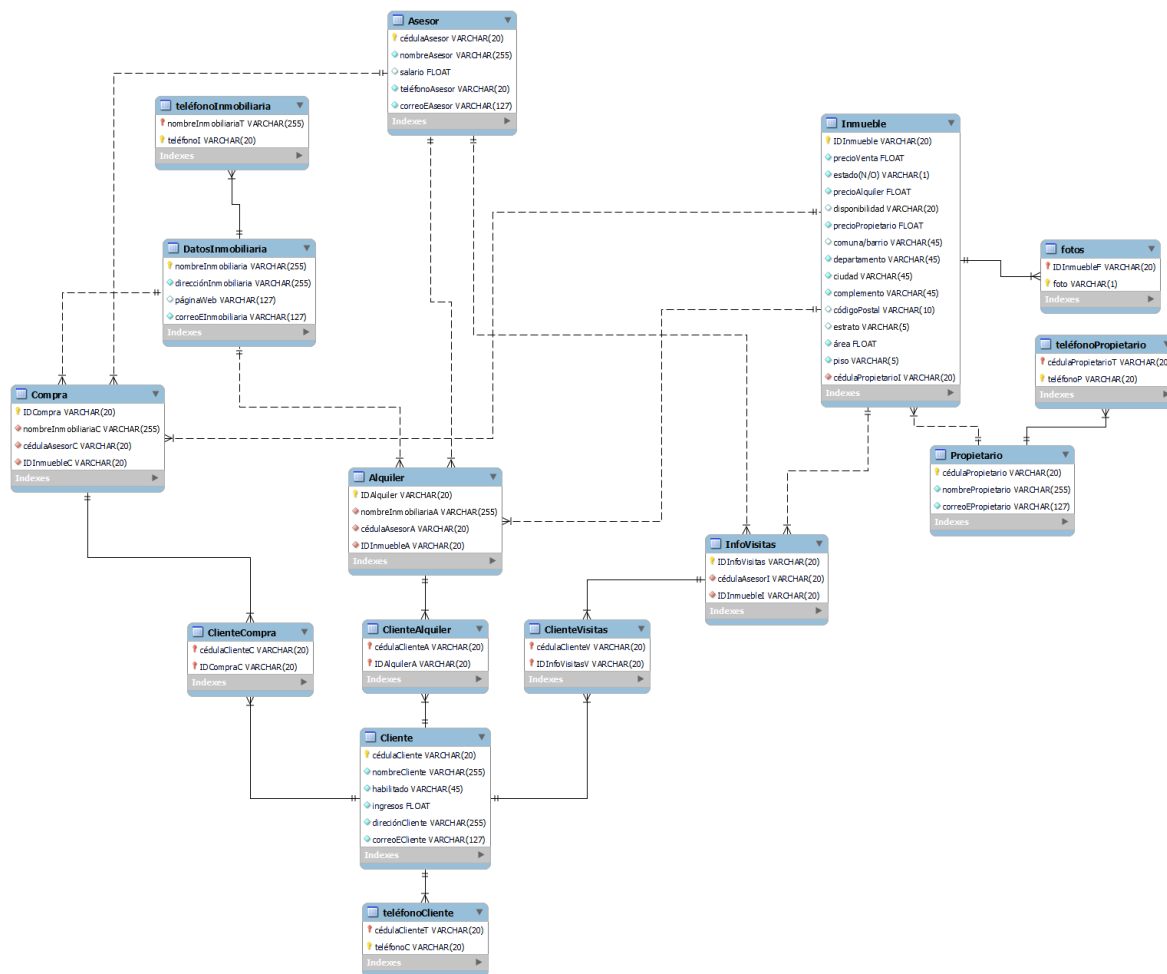



















Figura 2 MR de la Inmobiliaria SofkaU.









Se debe tener en cuenta que en el MER de la Figura 1 las entidades de Compra, Alquiler e InfoVisitas no poseen atributos, por lo que en la transformación se añade un atributo identificador para cada una, con el fin de poder realizar la transformación adecuada de las cardinalidades propuestas.









Así, se realiza el llenado de las tablas algunas resultantes, ya que es un proceso repetitivo y al ser una cantidad alta de tablas y columnas, se toma de ejemplo una sola línea de relaciones, donde cada tabla resulta como se muestra a continuación.

Cliente - Table						
<div>  Table Name: <input type="text" value="Cliente"/> </div> <div>  Filter Rows: <input type="text"/> </div> <div> Edit:    Export/Import:   Wrap Cell Content: <input type="checkbox"/> </div>						
	cédulaCliente	nombreCliente	habilitado	ingresos	direcciónCliente	correoECliente
▶	1111111111	Juan	si	123456	Calle 1 # 1 - 1	juan@gmail.com
	2222222222	José	si	456789	Calle 1 # 1 - 2	jose@gmail.com
	3333333333	Luis	no	789456	Calle 1 # 1 - 3	luis@gmail.com
	4444444444	Felipe	si	741852	Calle 1 # 1 - 4	felipe@gmail.com
	5555555555	David	si	159487	Calle 1 # 1 - 5	david@gmail.com
*	NULL	NULL	NULL	NULL	NULL	NULL

teléfonoCliente - Table		
<div>  Table Name: <input type="text" value="tel"/> </div> <div>  Filter Rows: <input type="text"/> </div>		
	cédulaClienteT	teléfonoC
▶	1111111111	4564564564
	1111111111	1231231231
	2222222222	7537537533
	3333333333	9519519511
	4444444444	7533577533
	5555555555	4264264264
*	NULL	NULL

DatosInmobiliaria - Table				
<div>  Table Name: <input type="text" value="DatosInmobiliaria"/> </div>				
<div>   Filter Rows: <input type="text"/> </div>				
<div> Edit:    Export/Import:   </div>				
	nombreInmobiliaria	direcciónInmobiliaria	páginaWeb	correoEInmobiliaria
▶	Hogar	Calle 2 # 1 - 1	hogar.com	hogar@inmo.com
	Casas	Calle 2 # 1 - 2	casas.com	casas@inmo.com
	Buendia	Calle 2 # 1 - 3	buendia.com	buendia@inmo.com
	SofkaU	Calle 2 # 1 - 4	sofkau.com	sofkau@inmo.com
	Castillo	Calle 2 # 1 - 5	castillo.com	castillo@inmo.com
•	NULL	NULL	NULL	NULL

teléfonoInmobiliaria - Table		
<div>  Table Name: <input type="text" value="teléfonoInmobiliaria"/> </div>		
<div>   Filter Rows: <input type="text"/> </div>		
<div> Edit:    Export/Import:   </div>		
	nombreInmobiliariaT	teléfonoI
▶	Castillo	7897897899
	Castillo	1231231233
	SofkaU	2362362366
	Hogar	5211255211
	Casas	4122144122
	Buendia	8744784788
•	NULL	NULL

Asesor - Table					
<div>  Table Name: <input type="text" value="Asesor"/> </div>					
<div>   Filter Rows: <input type="text"/> </div>					
<div> Edit:    Export/Import:   Wrap Cell: <input type="checkbox"/> </div>					
	cédulaAsesor	nombreAsesor	salario	teléfonoAsesor	correoEAsesor
▶	7777777777	Carlos	951951	5688566588	carlos@gmail.com
•	NULL	NULL	NULL	NULL	NULL

fotos - Table		
Table Name: fotos		
Filter Rows:		
	IDInmuebleF	foto
▶	11	A
•	NULL	NULL

Compra - Table				
Table Name: Compra				
Filter Rows:				
Edit: Export/Import:				
	IDCompra	nombreInmobiliariaC	cédulaAsesorC	IDInmuebleC
▶	1	SofkaU	777777777	11
•	NULL	NULL	NULL	NULL

ClienteCompra - Table		
Table Name: Cliente		
Filter Rows:		
	cédulaClienteC	IDCompraC
▶	1111111111	1
•	NULL	NULL

Finalmente, luego de crear el MR y rellenar algunos campos de las tablas, se hace ingeniería hacia adelante para generar el script SQL que me crearía la BD. Este se presenta a continuación.

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
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';
```

-- Schema mydb

-- Schema mydb

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
USE `mydb` ;
```

-- Table `mydb`.`DatosInmobiliaria`

```
CREATE TABLE IF NOT EXISTS `mydb`.`DatosInmobiliaria` (
  `nombreInmobiliaria` VARCHAR(255) NOT NULL,
  `direcciónInmobiliaria` VARCHAR(255) NOT NULL,
  `páginaWeb` VARCHAR(127) NULL,
  `correoEInmobiliaria` VARCHAR(127) NOT NULL,
  PRIMARY KEY (`nombreInmobiliaria`))
ENGINE = InnoDB;
```

-- Table `mydb`.`Asesor`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Asesor` (
  `cédulaAsesor` VARCHAR(20) NOT NULL,
  `nombreAsesor` VARCHAR(255) NOT NULL,
  `salario` FLOAT NULL,
  `teléfonoAsesor` VARCHAR(20) NOT NULL,
  `correoEAsesor` VARCHAR(127) NOT NULL,
  PRIMARY KEY (`cédulaAsesor`))
ENGINE = InnoDB;
```

-- Table `mydb`.`Propietario`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Propietario` (  
  `cédulaPropietario` VARCHAR(20) NOT NULL,  
  `nombrePropietario` VARCHAR(255) NOT NULL,  
  `correoEPropietario` VARCHAR(127) NOT NULL,  
  PRIMARY KEY (`cédulaPropietario`))  
ENGINE = InnoDB;
```

-- Table `mydb`.`Inmueble`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Inmueble` (  
  `IDInmueble` VARCHAR(20) NOT NULL,  
  `precioVenta` FLOAT NOT NULL,  
  `estado(N/O)` VARCHAR(1) NOT NULL,  
  `precioAlquiler` FLOAT NOT NULL,  
  `disponibilidad` VARCHAR(20) NULL,  
  `precioPropietario` FLOAT NOT NULL,  
  `comuna/barrio` VARCHAR(45) NULL,  
  `departamento` VARCHAR(45) NOT NULL,  
  `ciudad` VARCHAR(45) NOT NULL,  
  `complemento` VARCHAR(45) NOT NULL,  
  `códigoPostal` VARCHAR(10) NULL,  
  `estrato` VARCHAR(5) NULL,  
  `área` FLOAT NOT NULL,  
  `piso` VARCHAR(5) NOT NULL,  
  `cédulaPropietarioI` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`IDInmueble`),  
  INDEX `cédulaPropietarioI_idx` (`cédulaPropietarioI` ASC) VISIBLE,  
  CONSTRAINT `cédulaPropietarioI`  
    FOREIGN KEY (`cédulaPropietarioI`)  
    REFERENCES `mydb`.`Propietario` (`cédulaPropietario`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`Compra`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Compra` (  
  `IDCompra` VARCHAR(20) NOT NULL,  
  `nombreInmobiliariaC` VARCHAR(255) NOT NULL,  
  `cédulaAsesorC` VARCHAR(20) NOT NULL,  
  `IDInmuebleC` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`IDCompra`),  
  INDEX `nombreInmobiliariaC_idx` (`nombreInmobiliariaC` ASC) VISIBLE,  
  INDEX `cédulaAsesorC_idx` (`cédulaAsesorC` ASC) VISIBLE,  
  INDEX `IDInmuebleC_idx` (`IDInmuebleC` ASC) VISIBLE,  
  CONSTRAINT `nombreInmobiliariaC`  
    FOREIGN KEY (`nombreInmobiliariaC`)  
      REFERENCES `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `cédulaAsesorC`  
    FOREIGN KEY (`cédulaAsesorC`)  
      REFERENCES `mydb`.`Asesor` (`cédulaAsesor`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `IDInmuebleC`  
    FOREIGN KEY (`IDInmuebleC`)  
      REFERENCES `mydb`.`Inmueble` (`IDInmueble`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`Alquiler`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Alquiler` (  
  `IDAlquiler` VARCHAR(20) NOT NULL,  
  `nombreInmobiliariaA` VARCHAR(255) NOT NULL,  
  `cédulaAsesorA` VARCHAR(20) NOT NULL,  
  `IDInmuebleA` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`IDAlquiler`),  
  INDEX `nombreInmobiliariaA_idx` (`nombreInmobiliariaA` ASC) VISIBLE,
```

```

INDEX `cédulaAsesorA_idx` (`cédulaAsesorA` ASC) VISIBLE,
INDEX `IDInmuebleA_idx` (`IDInmuebleA` ASC) VISIBLE,
CONSTRAINT `nombreInmobiliariaA`
  FOREIGN KEY (`nombreInmobiliariaA`)
  REFERENCES `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `cédulaAsesorA`
  FOREIGN KEY (`cédulaAsesorA`)
  REFERENCES `mydb`.`Asesor` (`cédulaAsesor`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `IDInmuebleA`
  FOREIGN KEY (`IDInmuebleA`)
  REFERENCES `mydb`.`Inmueble` (`IDInmueble`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`InfoVisitas`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`InfoVisitas` (
  `IDInfoVisitas` VARCHAR(20) NOT NULL,
  `cédulaAsesorI` VARCHAR(20) NOT NULL,
  `IDInmuebleI` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`IDInfoVisitas`),
  INDEX `cédulaAsesorI_idx` (`cédulaAsesorI` ASC) VISIBLE,
  INDEX `IDInmueble_idx` (`IDInmuebleI` ASC) VISIBLE,
  CONSTRAINT `cédulaAsesorI`
    FOREIGN KEY (`cédulaAsesorI`)
    REFERENCES `mydb`.`Asesor` (`cédulaAsesor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `IDInmueble`
    FOREIGN KEY (`IDInmuebleI`)
    REFERENCES `mydb`.`Inmueble` (`IDInmueble`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-- Table `mydb`.`Cliente`

```
CREATE TABLE IF NOT EXISTS `mydb`.`Cliente` (  
  `cédulaCliente` VARCHAR(20) NOT NULL,  
  `nombreCliente` VARCHAR(255) NOT NULL,  
  `habilitado` VARCHAR(45) NOT NULL,  
  `ingresos` FLOAT NOT NULL,  
  `direcciónCliente` VARCHAR(255) NOT NULL,  
  `correoECliente` VARCHAR(127) NOT NULL,  
  PRIMARY KEY (`cédulaCliente`))  
ENGINE = InnoDB;
```

-- Table `mydb`.`teléfonoInmobiliaria`

```
CREATE TABLE IF NOT EXISTS `mydb`.`teléfonoInmobiliaria` (  
  `nombreInmobiliariaT` VARCHAR(255) NOT NULL,  
  `teléfonoI` VARCHAR(20) NOT NULL,  
  INDEX `nombreInmobiliariaT_idx` (`nombreInmobiliariaT` ASC) VISIBLE,  
  PRIMARY KEY (`nombreInmobiliariaT`, `teléfonoI`),  
  CONSTRAINT `nombreInmobiliariaT`  
    FOREIGN KEY (`nombreInmobiliariaT`)  
      REFERENCES `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `mydb`.`fotos`

```
CREATE TABLE IF NOT EXISTS `mydb`.`fotos` (  
  `IDInmuebleF` VARCHAR(20) NOT NULL,  
  `foto` VARCHAR(1) NOT NULL,  
  PRIMARY KEY (`IDInmuebleF`, `foto`),  
  CONSTRAINT `IDInmuebleF`  
    FOREIGN KEY (`IDInmuebleF`)
```

```
REFERENCES `mydb`.`Inmueble` (`IDInmueble`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`telefonoPropietario`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`telefonoPropietario` (
  `cédulaPropietarioT` VARCHAR(20) NOT NULL,
  `telefonoP` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`cédulaPropietarioT`, `telefonoP`),
  CONSTRAINT `cédulaPropietarioT`
    FOREIGN KEY (`cédulaPropietarioT`)
    REFERENCES `mydb`.`Propietario` (`cédulaPropietario`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`telefonoCliente`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`telefonoCliente` (
  `cédulaClienteT` VARCHAR(20) NOT NULL,
  `telefonoC` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`cédulaClienteT`, `telefonoC`),
  CONSTRAINT `telefonoC`
    FOREIGN KEY (`cédulaClienteT`)
    REFERENCES `mydb`.`Cliente` (`cédulaCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `mydb`.`ClienteCompra`
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`ClienteCompra` (
  `cédulaClienteC` VARCHAR(20) NOT NULL,
```

```

`IDCompraC` VARCHAR(20) NOT NULL,
PRIMARY KEY (`cédulaClienteC`, `IDCompraC`),
INDEX `IDCompraC_idx` (`IDCompraC` ASC) VISIBLE,
CONSTRAINT `IDCompraC`
  FOREIGN KEY (`IDCompraC`)
    REFERENCES `mydb`.`Compra` (`IDCompra`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `cédulaClienteC`
  FOREIGN KEY (`cédulaClienteC`)
    REFERENCES `mydb`.`Cliente` (`cédulaCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`ClienteAlquiler`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`ClienteAlquiler` (
  `cédulaClienteA` VARCHAR(20) NOT NULL,
  `IDAlquilerA` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`cédulaClienteA`, `IDAlquilerA`),
  CONSTRAINT `IDAlquilerA`
    FOREIGN KEY (`IDAlquilerA`)
      REFERENCES `mydb`.`Alquiler` (`IDAlquiler`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `cédulaClienteA`
    FOREIGN KEY (`cédulaClienteA`)
      REFERENCES `mydb`.`Cliente` (`cédulaCliente`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `mydb`.`ClienteVisitas`
-----

```

```

CREATE TABLE IF NOT EXISTS `mydb`.`ClienteVisitas` (
  `cédulaClienteV` VARCHAR(20) NOT NULL,

```

```

`IDInfoVisitasV` VARCHAR(20) NOT NULL,
PRIMARY KEY (`cédulaClienteV`, `IDInfoVisitasV`),
INDEX `IDInfoVisitasV_idx` (`IDInfoVisitasV` ASC) VISIBLE,
CONSTRAINT `IDInfoVisitasV`
  FOREIGN KEY (`IDInfoVisitasV`)
    REFERENCES `mydb`.`InfoVisitas` (`IDInfoVisitas`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `cédulaClienteV`
  FOREIGN KEY (`cédulaClienteV`)
    REFERENCES `mydb`.`Cliente` (`cédulaCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

```

-----
-- Data for table `mydb`.`DatosInmobiliaria`
-----

```

```

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`, `direcciónInmobiliaria`,
`páginaWeb`, `correoEInmobiliaria`) VALUES ('Hogar', 'Calle 2 # 1 - 1', 'hogar.com',
'hogar@inmo.com');
INSERT INTO `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`, `direcciónInmobiliaria`,
`páginaWeb`, `correoEInmobiliaria`) VALUES ('Casas', 'Calle 2 # 1 - 2', 'casas.com',
'casas@inmo.com');
INSERT INTO `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`, `direcciónInmobiliaria`,
`páginaWeb`, `correoEInmobiliaria`) VALUES ('Buendia', 'Calle 2 # 1 - 3', 'buendia.com',
'buendia@inmo.com');
INSERT INTO `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`, `direcciónInmobiliaria`,
`páginaWeb`, `correoEInmobiliaria`) VALUES ('SofkaU', 'Calle 2 # 1 - 4', 'sofkau.com',
'sofkau@inmo.com');
INSERT INTO `mydb`.`DatosInmobiliaria` (`nombreInmobiliaria`, `direcciónInmobiliaria`,
`páginaWeb`, `correoEInmobiliaria`) VALUES ('Castillo', 'Calle 2 # 1 - 5', 'castillo.com',
'castillo@inmo.com');

```

COMMIT;

-- Data for table `mydb`.`Asesor`

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Asesor` (`cédulaAsesor`, `nombreAsesor`, `salario`, `teléfonoAsesor`,
`correoEAsesor`) VALUES ('7777777777', 'Carlos', 951951, '5688566588',
'carlos@gmail.com');

COMMIT;

-- Data for table `mydb`.`Propietario`

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Propietario` (`cédulaPropietario`, `nombrePropietario`,
`correoEPropietario`) VALUES ('9999999999', 'Fernando', 'fernando@gmail.com');

COMMIT;

-- Data for table `mydb`.`Inmueble`

START TRANSACTION;

USE `mydb`;

INSERT INTO `mydb`.`Inmueble` (`IDInmueble`, `precioVenta`, `estado(N/O)`,
`precioAlquiler`, `disponibilidad`, `precioPropietario`, `comuna/barrio`, `departamento`,
`ciudad`, `complemento`, `códigoPostal`, `estrato`, `área`, `piso`, `cédulaPropietarioI`)
VALUES ('11', 5000000, 'N', 300000, 'si', 3000000, 'Belén', 'Antioquia', 'Medellín', 'Calle 3 #
1 - 1', '05000505', '3', 40, '1', '9999999999');

COMMIT;

-- Data for table `mydb`.`Compra`

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`Compra` (`IDCompra`, `nombreInmobiliariaC`, `cédulaAsesorC`,
`IDInmuebleC`) VALUES ('1', 'SofkaU', '7777777777', '11');

COMMIT;

-- Data for table `mydb`.`Cliente`

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`Cliente` (`cédulaCliente`, `nombreCliente`, `habilitado`, `ingresos`,
`direcciónCliente`, `correoECliente`) VALUES ('1111111111', 'Juan', 'si', 123456, 'Calle 1 # 1 -
1', 'juan@gmail.com');
INSERT INTO `mydb`.`Cliente` (`cédulaCliente`, `nombreCliente`, `habilitado`, `ingresos`,
`direcciónCliente`, `correoECliente`) VALUES ('2222222222', 'José', 'si', 456789, 'Calle 1 # 1 -
2', 'jose@gmail.com');
INSERT INTO `mydb`.`Cliente` (`cédulaCliente`, `nombreCliente`, `habilitado`, `ingresos`,
`direcciónCliente`, `correoECliente`) VALUES ('3333333333', 'Luis', 'no', 789456, 'Calle 1 # 1 -
3', 'luis@gmail.com');
INSERT INTO `mydb`.`Cliente` (`cédulaCliente`, `nombreCliente`, `habilitado`, `ingresos`,
`direcciónCliente`, `correoECliente`) VALUES ('4444444444', 'Felipe', 'si', 741852, 'Calle 1 # 1
- 4', 'felipe@gmail.com');
INSERT INTO `mydb`.`Cliente` (`cédulaCliente`, `nombreCliente`, `habilitado`, `ingresos`,
`direcciónCliente`, `correoECliente`) VALUES ('5555555555', 'David', 'si', 159487, 'Calle 1 # 1
- 5', 'david@gmail.com');

COMMIT;

-- Data for table `mydb`.`teléfonoInmobiliaria`

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`teléfonoInmobiliaria` (`nombreInmobiliariaT`, `teléfonoI`) VALUES
('Castillo', '7897897899');


```
INSERT INTO `mydb`.`telefonoInmobiliaria` (`nombreInmobiliariaT`, `telefonoI`) VALUES
('Castillo', '1231231233');
INSERT INTO `mydb`.`telefonoInmobiliaria` (`nombreInmobiliariaT`, `telefonoI`) VALUES
('SofkaU', '2362362366');
INSERT INTO `mydb`.`telefonoInmobiliaria` (`nombreInmobiliariaT`, `telefonoI`) VALUES
('Hogar', '5211255211');
INSERT INTO `mydb`.`telefonoInmobiliaria` (`nombreInmobiliariaT`, `telefonoI`) VALUES
('Casas', '4122144122');
INSERT INTO `mydb`.`telefonoInmobiliaria` (`nombreInmobiliariaT`, `telefonoI`) VALUES
('Buendia', '8744784788');
```

```
COMMIT;
```

```
-- -----
-- Data for table `mydb`.`fotos`
-- -----
```

```
START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`fotos` (`IDInmuebleF`, `foto`) VALUES ('11', 'A');
```

```
COMMIT;
```

```
-- -----
-- Data for table `mydb`.`telefonoPropietario`
-- -----
```

```
START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`telefonoPropietario` (`cédulaPropietarioT`, `telefonoP`) VALUES
('9999999999', '5411255144');
```

```
COMMIT;
```

```
-- -----
-- Data for table `mydb`.`telefonoCliente`
-- -----
```

```
START TRANSACTION;
USE `mydb`;
```

```

INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('1111111111', '4564564564');
INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('1111111111', '1231231231');
INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('2222222222', '7537537533');
INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('3333333333', '9519519511');
INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('4444444444', '7533577533');
INSERT INTO `mydb`.`telefonoCliente` (`cedulaClienteT`, `telefonoC`) VALUES
('5555555555', '4264264264');

```

```

COMMIT;

```

```

-----
-- Data for table `mydb`.`ClienteCompra`
-----

```

```

START TRANSACTION;
USE `mydb`;
INSERT INTO `mydb`.`ClienteCompra` (`cedulaClienteC`, `IDCompraC`) VALUES
('1111111111', '1');

```

```

COMMIT;

```

Preguntas

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

Agregarle 3 atributos a la entidad Alquiler, FechaInicio, FechaActual y FechaFin, donde cada que pase un mes en la FechaActual se cobre de nuevo el dinero hasta que se alcance la FechaFin.

- ¿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?

Creando una relación entre Cliente y Propietario, donde en la relación haya un atributo multivaluado que identifique las propiedades que tiene ese cliente y las excluya de las disponibles para ese cliente en específico.