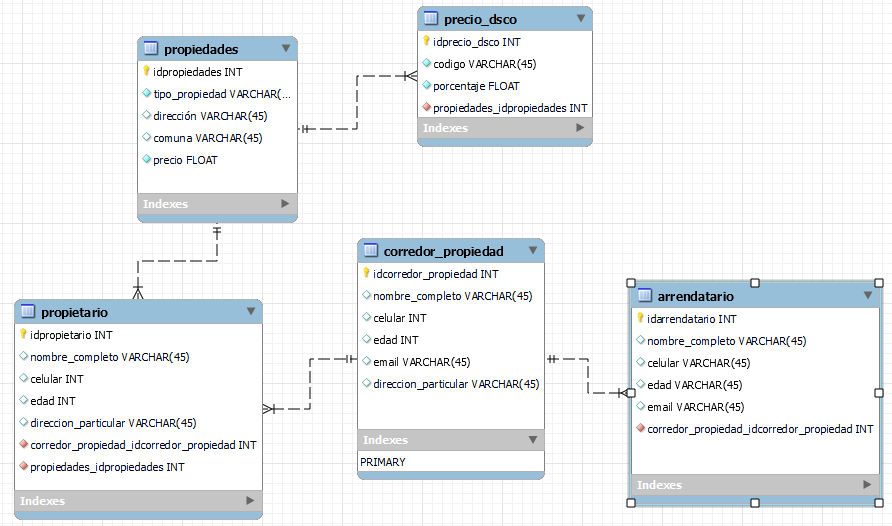
 La problemática que elegí fue una relación entre propietarios de casa, arrendatarios y propiedades.

Básicamente se refiere a que un propietario puede tener varias propiedades a ciertos precios, y esos precios pueden tener varios descuentos, pero solo se le pueden aplicar un solo descuento.

También que un corredor de propiedades puede comunicarse con muchos propietarios y arrendatarios, pero estos solo con un corredor.



Script

-- MySQL Script generated by MySQL Workbench

-- Thu Jun 17 11:03:50 2021

-- Model: Sakila Full Version: 2.0

-- 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='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema new\_schema1

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema new\_schema1

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `new\_schema1` ;

USE `new\_schema1` ;

-- -----------------------------------------------------

-- Table `new\_schema1`.`propiedades`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `new\_schema1`.`propiedades` (

`idpropiedades` INT NOT NULL AUTO\_INCREMENT,

`tipo\_propiedad` VARCHAR(45) NOT NULL,

`dirección` VARCHAR(45) NULL,

`comuna` VARCHAR(45) NULL,

`precio` FLOAT NOT NULL,

PRIMARY KEY (`idpropiedades`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `new\_schema1`.`precio\_dsco`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `new\_schema1`.`precio\_dsco` (

`idprecio\_dsco` INT NOT NULL AUTO\_INCREMENT,

`codigo` VARCHAR(45) NOT NULL,

`porcentaje` FLOAT NOT NULL,

`propiedades\_idpropiedades` INT NOT NULL,

PRIMARY KEY (`idprecio\_dsco`),

INDEX `fk\_precio\_dsco\_propiedades\_idx` (`propiedades\_idpropiedades` ASC),

CONSTRAINT `fk\_precio\_dsco\_propiedades`

FOREIGN KEY (`propiedades\_idpropiedades`)

REFERENCES `new\_schema1`.`propiedades` (`idpropiedades`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `new\_schema1`.`corredor\_propiedad`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `new\_schema1`.`corredor\_propiedad` (

`idcorredor\_propiedad` INT NOT NULL AUTO\_INCREMENT,

`nombre\_completo` VARCHAR(45) NULL,

`celular` INT NULL,

`edad` INT NULL,

`email` VARCHAR(45) NULL,

`direccion\_particular` VARCHAR(45) NULL,

PRIMARY KEY (`idcorredor\_propiedad`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `new\_schema1`.`propietario`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `new\_schema1`.`propietario` (

`idpropietario` INT NOT NULL AUTO\_INCREMENT,

`nombre\_completo` VARCHAR(45) NULL,

`celular` INT NULL,

`edad` INT NULL,

`direccion\_particular` VARCHAR(45) NULL,

`corredor\_propiedad\_idcorredor\_propiedad` INT NOT NULL,

`propiedades\_idpropiedades` INT NOT NULL,

PRIMARY KEY (`idpropietario`),

INDEX `fk\_propietario\_corredor\_propiedad1\_idx` (`corredor\_propiedad\_idcorredor\_propiedad` ASC),

INDEX `fk\_propietario\_propiedades1\_idx` (`propiedades\_idpropiedades` ASC),

CONSTRAINT `fk\_propietario\_corredor\_propiedad1`

FOREIGN KEY (`corredor\_propiedad\_idcorredor\_propiedad`)

REFERENCES `new\_schema1`.`corredor\_propiedad` (`idcorredor\_propiedad`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_propietario\_propiedades1`

FOREIGN KEY (`propiedades\_idpropiedades`)

REFERENCES `new\_schema1`.`propiedades` (`idpropiedades`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `new\_schema1`.`arrendatario`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `new\_schema1`.`arrendatario` (

`idarrendatario` INT NOT NULL AUTO\_INCREMENT,

`nombre\_completo` VARCHAR(45) NULL,

`celular` VARCHAR(45) NULL,

`edad` VARCHAR(45) NULL,

`email` VARCHAR(45) NULL,

`corredor\_propiedad\_idcorredor\_propiedad` INT NOT NULL,

PRIMARY KEY (`idarrendatario`),

INDEX `fk\_arrendatario\_corredor\_propiedad1\_idx` (`corredor\_propiedad\_idcorredor\_propiedad` ASC),

CONSTRAINT `fk\_arrendatario\_corredor\_propiedad1`

FOREIGN KEY (`corredor\_propiedad\_idcorredor\_propiedad`)

REFERENCES `new\_schema1`.`corredor\_propiedad` (`idcorredor\_propiedad`)

ON DELETE NO ACTION

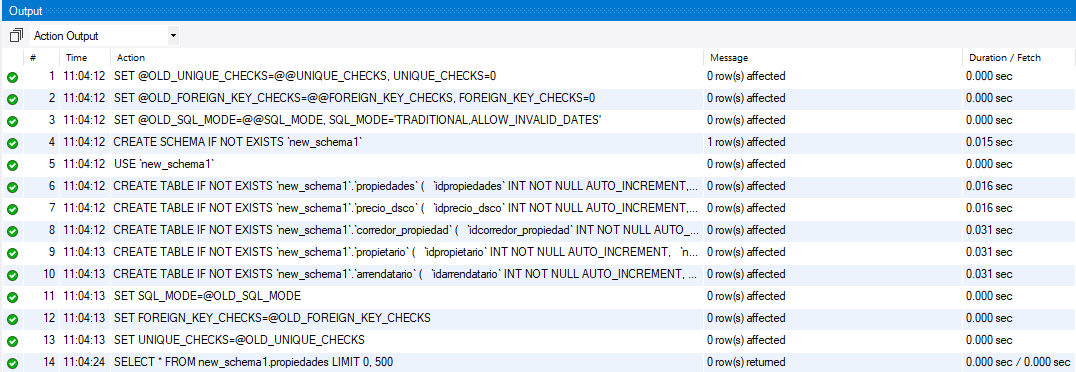
ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

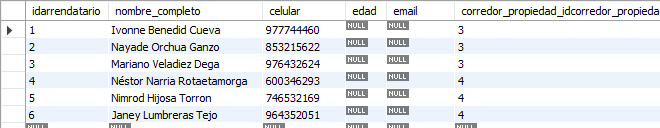
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;



**Columnas**

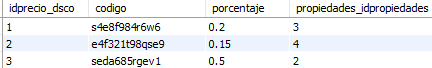
**Arrendatarios**



**Corredores de propiedades**



**Descuentos**



**Propiedades**



**Propietarios**



Preguntas:

1. Primero ver quiénes son los corredores de propiedades.

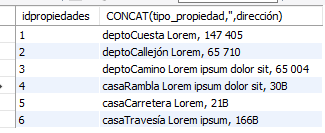
SELECT\*FROM corredor\_propiedad;



1. Quiero ver las propiedades con el tipo de propiedades que es.

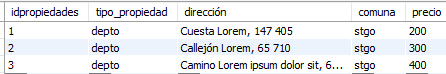
SELECT\*FROM propiedades;

SELECT idpropiedades,CONCAT(tipo\_propiedad,'',dirección) FROM propiedades;

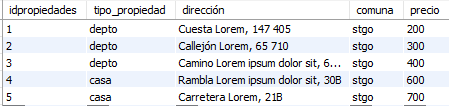


1. Para ver propiedades entre precios.

select\*from propiedades where precio>=200 and precio<=400;

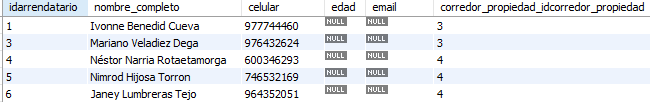


select\*from propiedades where precio between 200 and 700;



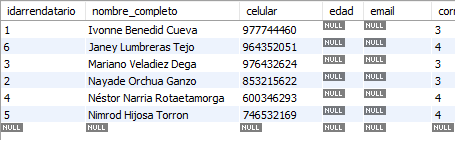
1. Para buscar a un arrendatario que tenga un 4 en el número de teléfono.

SELECT\*FROM arrendatario where celular like '%4%';

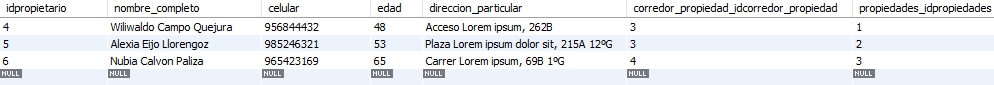


1. Ordenar a los arrendatarios por nombre ascendente y propietarios por edad.

SELECT\*FROM arrendatario order by nombre\_completo asc;



SELECT\*FROM propietario group by edad;



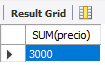
1. Para sacar el promedio de la edad de los propietarios de las propiedades.

SELECT avg(edad) from propietario;



1. Sacar suma de los precios.

SELECT SUM(precio) from propiedades;



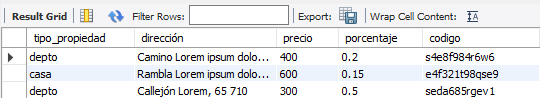
1. Para sacar el porcentaje determinado de descuento en cada una de las propiedades con dichos descuentos, se necesita crear un JOIN para mezclar y observar las tablas.

SELECT propiedades.tipo\_propiedad, propiedades.dirección, propiedades.precio, precio\_dsco.porcentaje, precio\_dsco.codigo

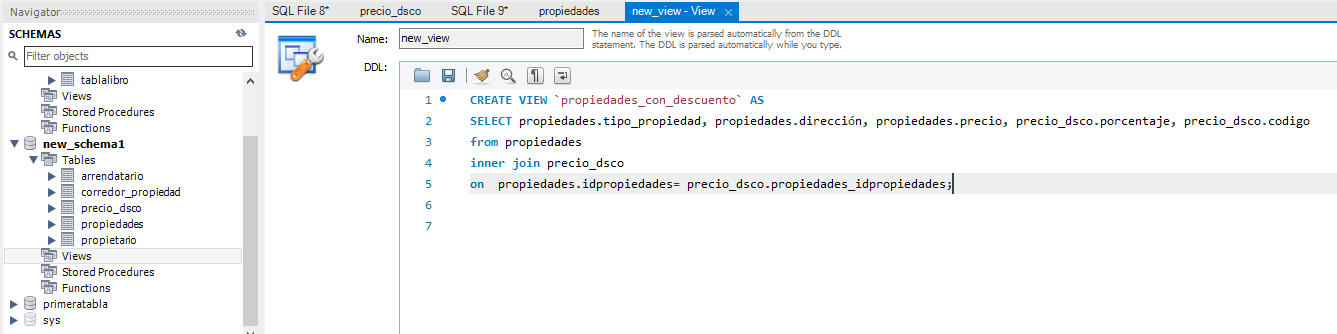
from propiedades

inner join precio\_dsco

on propiedades.idpropiedades= precio\_dsco.propiedades\_idpropiedades;



**VISTA**



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `new\_schema1`.`propiedades\_con\_descuento` AS

SELECT

`new\_schema1`.`propiedades`.`tipo\_propiedad` AS `tipo\_propiedad`,

`new\_schema1`.`propiedades`.dirección AS dirección,

new\_schema1.propiedades.precio AS precio,

new\_schema1.precio\_dsco.porcentaje AS porcentaje,

new\_schema1.precio\_dsco.codigo AS codigo

FROM

(new\_schema1.propiedades

JOIN new\_schema1.precio\_dsco ON ((new\_schema1.propiedades.idpropiedades = new\_schema1.precio\_dsco.propiedades\_idpropiedades)))

Para la creación de la vista utilice el inner join para obtener de manera mas especifica de las propiedades con descuentos.

