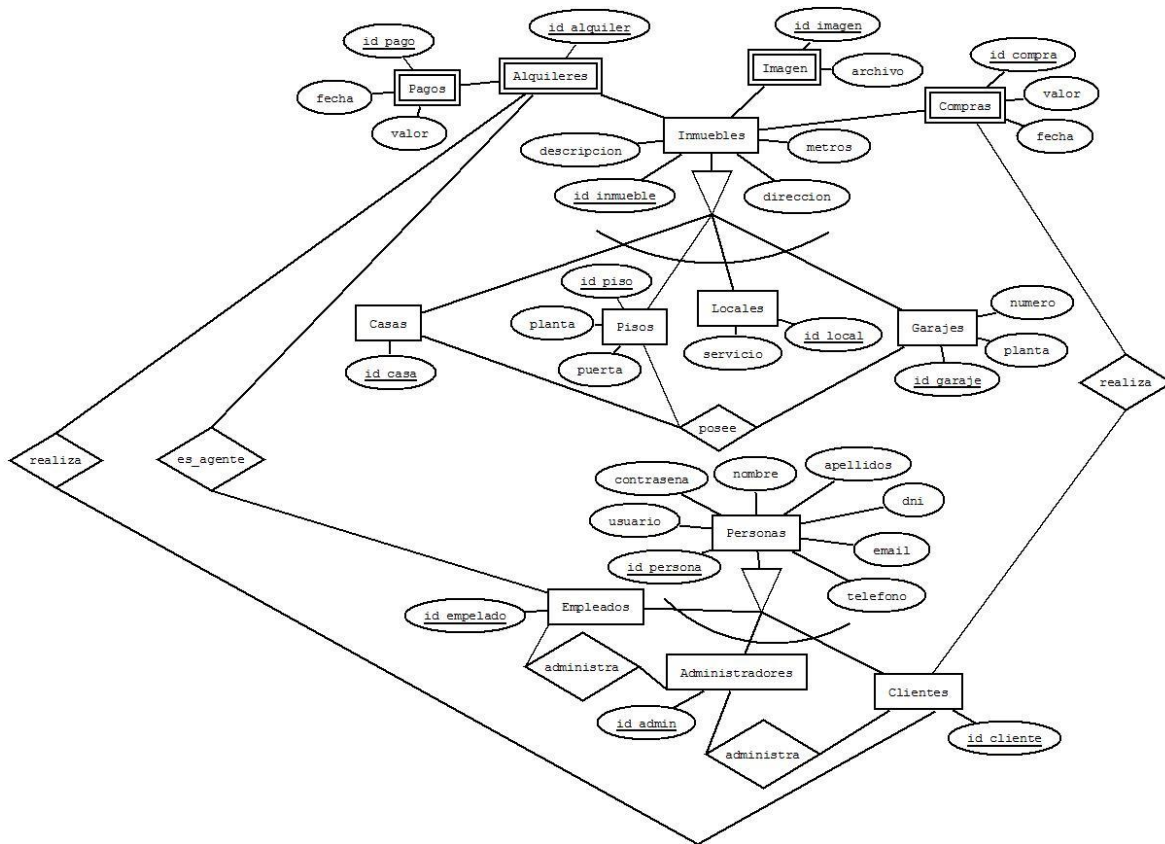


Proyecto ASGBD

BASE DE DATOS: INMOBILIARIA AZTECA. REALIZADO POR RAÚL ORTEGA

Diagrama realizado.



La base de datos consta de una inmobiliaria, que permita guardar los administradores que administran el sitio de la inmobiliaria, los empleados puedan poner en venta o en alquiler los inmuebles y los clientes puedan acceder a esa venta de inmuebles.

Los inmuebles tienen distintos tipos: Casas, Pisos, Locales y Garajes. Los garajes pueden estar o no unidos a las casas o pisos. Los garajes pueden estar solos independientemente sin las casas y los pisos para ponerse en venta o alquiler. Los locales no pueden tener garajes puestos (la inmobiliaria vende o alquila los locales con garaje propio). Los pisos se diferencian por dos atributos, planta y puerta de ese piso.

Una inmobiliaria puede tener más de dos imágenes puestas en el mismo inmueble.

Los empleados serán los agentes de los alquileres de esos inmuebles y ellos contendrán cuántos pagos recibe ese mismo alquiler.

Las compras guardarán el valor y la fecha cuando se cree ese inmueble en venta.

Modelo relacional

personas(id_persona(pk), usuario, contrasena, nombre, apellidos, dni, email, telefono)

empleados(id_empleado(pk))

clientes(id_cliente(pk))

administradores(id_admin(pk))

inmuebles(id_inmueble(pk), descripcion, direccion, metros)

imagen(id_imagen(pk), archivo, inmueble(fk))

garajes(id_garaje(pk), planta, numero, id_cliente(fk))

casas(id_casa(pk), id_garaje(fk), id_cliente(fk))

pisos(id_piso(pk), planta, puerta, id_garaje(fk), id_cliente(fk))

locales(id_local(pk), servicio, id_cliente(fk))

compras(id_compra(pk), valor, fecha, id_inmueble(fk), id_cliente(fk))

alquileres(id_alquiler(pk), id_inmueble(fk), id_empleado(fk), id_cliente(fk))

pagos(id_pago(pk), fecha, valor, id_alquiler(fk))

Código SQL

- **Creacion de la base de datos y tablas.**

```
CREATE DATABASE azteca;
```

```
USE azteca;
```

- Tabla empleados

```
CREATE TABLE empleados (
```

```
id_empleado INT(2) NOT NULL PRIMARY KEY AUTO_INCREMENT,
```

```
usuario VARCHAR(12) NOT NULL,
```

```
contrasena VARCHAR(50) NOT NULL,
```

```
nombre VARCHAR(20) NOT NULL,
```

```
apellidos VARCHAR(50) NOT NULL,
```

```
dni VARCHAR(9) NOT NULL,
```

```
email VARCHAR(125) NOT NULL,
```

```
telefono INT(9) NOT NULL,
```

```
UNIQUE (usuario)
```

```
)ENGINE=InnoDB;
```

- Tabla de los administradores

```
CREATE TABLE administradores (
```

```
id_admin INT(2) NOT NULL PRIMARY KEY AUTO_INCREMENT,
```

```
usuario VARCHAR(12) NOT NULL,
```

```
contrasena VARCHAR(50) NOT NULL,
```

```
nombre VARCHAR(20) NOT NULL,
```

```
apellidos VARCHAR(50) NOT NULL,
```

```
dni VARCHAR(9) NOT NULL,
```

```
email VARCHAR(125) NOT NULL,
```

```
telefono INT(9) NOT NULL,
```

```
UNIQUE (usuario)
```

```
)ENGINE=InnoDB;
```

- Tabla de los clientes

```
CREATE TABLE clientes (  
  id_cliente INT(4) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  usuario VARCHAR(12) NOT NULL,  
  contrasena VARCHAR(50) NOT NULL,  
  nombre VARCHAR(20) NOT NULL,  
  apellidos VARCHAR(50) NOT NULL,  
  dni VARCHAR(9) NOT NULL,  
  email VARCHAR(125) NOT NULL,  
  telefono INT(9) NOT NULL,  
  UNIQUE (usuario)  
)ENGINE=InnoDB;
```

- Tabla de los garajes

```
CREATE TABLE garajes (  
  id_garaje INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  descripcion VARCHAR(240) NOT NULL,  
  metros INT(4) NOT NULL,  
  direccion VARCHAR(50) NOT NULL,  
  planta INT(1) NOT NULL,  
  numero INT(1) NOT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Tabla de las casas

```
CREATE TABLE casas (  
  id_casa INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  descripcion TEXT,  
  metros INT(4) NOT NULL,  
  direccion VARCHAR(50) NOT NULL,  
  garaje INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (garaje) REFERENCES garajes(id_garaje) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Tabla de los pisos

```
CREATE TABLE pisos (  
  id_piso INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  descripcion TEXT,  
  metros INT(4) NOT NULL,  
  direccion VARCHAR(50) NOT NULL,  
  planta INT(1) NOT NULL,  
  puerta VARCHAR(1) NOT NULL,  
  garaje INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (garaje) REFERENCES garajes(id_garaje) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Tabla de los locales

```
CREATE TABLE locales (  
  id_local INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  descripcion TEXT,  
  metros INT(4) NOT NULL,  
  direccion VARCHAR(50) NOT NULL,  
  servicio VARCHAR(20) NOT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Imagen casas

```
CREATE TABLE imagen_casas (  
  id_imagen_casa INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  archivo VARCHAR(20) NOT NULL,  
  casa INT(6) DEFAULT NULL,  
  FOREIGN KEY (casa) REFERENCES casas(id_casa) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Imagen pisos

```
CREATE TABLE imagen_pisos (  
  id_imagen_piso INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  archivo VARCHAR(20) NOT NULL,  
  piso INT(6) DEFAULT NULL,  
  FOREIGN KEY (piso) REFERENCES pisos(id_piso) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Imagen locales

```
CREATE TABLE imagen_locales (  
  id_imagen_local INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  archivo VARCHAR(20) NOT NULL,  
  local INT(6) DEFAULT NULL,  
  FOREIGN KEY (local) REFERENCES locales(id_local) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Imagen garajes

```
CREATE TABLE imagen_garajes (  
  id_imagen_garaje INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  archivo VARCHAR(20) NOT NULL,  
  garaje INT(6) DEFAULT NULL,  
  FOREIGN KEY (garaje) REFERENCES garajes(id_garaje) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Compras y alquileres de casas

```
CREATE TABLE compras_casas (  
  id_compra_casa INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  valor INT(8) NOT NULL,  
  fecha DATE NOT NULL,  
  edificio INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES casas(id_casa) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE alquiler_casas (  
  id_alquiler_casa INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  edificio INT(6) DEFAULT NULL,  
  agente INT(2) DEFAULT NULL,  
  cliente INT(2) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES casas(id_casa) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (agente) REFERENCES empleados(id_empleado) ON UPDATE CASCADE ON DELETE  
  CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE pagos_casas (  
  id_pago_casa INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  fecha DATE NOT NULL,  
  valor INT(4) NOT NULL,  
  alquiler INT(6) NOT NULL,  
  FOREIGN KEY (alquiler) REFERENCES alquiler_casas(id_alquiler_casa) ON UPDATE CASCADE ON DELETE  
  CASCADE  
)ENGINE=InnoDB;
```


- Compras y alquileres de pisos

```
CREATE TABLE compras_pisos (  
  id_compra_piso INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  valor INT(8) NOT NULL,  
  fecha DATE NOT NULL,  
  edificio INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES pisos(id_piso) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE alquiler_pisos (  
  id_alquiler_piso INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  edificio INT(6) DEFAULT NULL,  
  agente INT(2) DEFAULT NULL,  
  cliente INT(2) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES pisos(id_piso) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (agente) REFERENCES empleados(id_empleado) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE pagos_pisos (  
  id_pago_piso INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  fecha DATE NOT NULL,  
  valor INT(4) NOT NULL,  
  alquiler INT(6) NOT NULL,  
  FOREIGN KEY (alquiler) REFERENCES alquiler_pisos(id_alquiler_piso) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Compras y alquileres de locales

```
CREATE TABLE compras_locales (  
  id_compra_local INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  valor INT(8) NOT NULL,  
  fecha DATE NOT NULL,  
  edificio INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES locales(id_local) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE alquiler_locales (  
  id_alquiler_local INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  edificio INT(6) DEFAULT NULL,  
  agente INT(2) DEFAULT NULL,  
  cliente INT(2) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES locales(id_local) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (agente) REFERENCES empleados(id_empleado) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE pagos_locales (  
  id_pago_local INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  fecha DATE NOT NULL,  
  valor INT(4) NOT NULL,  
  alquiler INT(6) NOT NULL,  
  FOREIGN KEY (alquiler) REFERENCES alquiler_locales(id_alquiler_local) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

- Compras y alquileres de garajes

```
CREATE TABLE compras_garajes (  
  id_compra_garaje INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  valor INT(8) NOT NULL,  
  fecha DATE NOT NULL,  
  edificio INT(6) DEFAULT NULL,  
  cliente INT(4) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES garajes(id_garaje) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE alquiler_garajes (  
  id_alquiler_garaje INT(6) NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  edificio INT(6) DEFAULT NULL,  
  agente INT(2) DEFAULT NULL,  
  cliente INT(2) DEFAULT NULL,  
  FOREIGN KEY (edificio) REFERENCES garajes(id_garaje) ON UPDATE CASCADE ON DELETE CASCADE,  
  FOREIGN KEY (agente) REFERENCES empleados(id_empleado) ON UPDATE CASCADE ON DELETE  
  CASCADE,  
  FOREIGN KEY (cliente) REFERENCES clientes(id_cliente) ON UPDATE CASCADE ON DELETE CASCADE  
)ENGINE=InnoDB;
```

```
CREATE TABLE pagos_garajes (  
  id_pago_garaje INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  fecha DATE NOT NULL,  
  valor INT(4) NOT NULL,  
  alquiler INT(6) NOT NULL,  
  FOREIGN KEY (alquiler) REFERENCES alquiler_garajes(id_alquiler_garaje) ON UPDATE CASCADE ON  
  DELETE CASCADE  
)ENGINE=InnoDB;
```

- **Creación de usuarios y permisos**
- Administrador

```
GRANT ALL PRIVILEGES ON azteca.* TO admin@localhost IDENTIFIED BY "inves";
```

- Jefe de los empleados

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.empleados TO jefe_empleado@192.168.70.20  
IDENTIFIED BY "inves";
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.casas TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.pisos TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.locales TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.garajes TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.alquiler_casas TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.pagos_casas TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.alquiler_pisos TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.pagos_pisos TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.alquiler_locales TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.pagos_locales TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.alquiler_garajes TO jefe_empleado@192.168.70.20;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON azteca.pagos_garajes TO jefe_empleado@192.168.70.20;
```

- Empleados

```
GRANT SELECT, INSERT ON azteca.casas TO empleado1 IDENTIFIED BY "inves";
```

```
GRANT SELECT, INSERT ON azteca.pisos TO empleado1;
```

```
GRANT SELECT, INSERT ON azteca.locales TO empleado1;
```

```
GRANT SELECT, INSERT ON azteca.garajes TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_casas TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_casas TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_pisos TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_pisos TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_locales TO empleado1;
```

```
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_locales TO empleado1;  
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_garajes TO empleado1;  
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_garajes TO empleado1;
```

```
GRANT SELECT, INSERT ON azteca.casas TO empleado2 IDENTIFIED BY "inves";  
GRANT SELECT, INSERT ON azteca.pisos TO empleado2;  
GRANT SELECT, INSERT ON azteca.locales TO empleado2;  
GRANT SELECT, INSERT ON azteca.garajes TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_casas TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_casas TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_pisos TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_pisos TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_locales TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_locales TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.alquiler_garajes TO empleado2;  
GRANT SELECT, INSERT, UPDATE ON azteca.pagos_garajes TO empleado2;
```

- **Procedimientos**
- Procedimiento que se introduce la funcion sumatodopagos como parametro de entrada y muestra el total de pagos con el impuesto incluido

Resultado

```
MariaDB [azteca]> DELIMITER //
```

```
MariaDB [azteca]> CREATE PROCEDURE impuesto_pago(IN valor INT)
```

```
-> BEGIN
```

```
-> DECLARE operacion INT;
```

```
-> DECLARE resultado INT;
```

```
-> SET operacion=valor*0.21;
```

```
-> SET resultado=valor - operacion;
```

```
-> SELECT resultado;
```

```
-> END//
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [azteca]> DELIMITER ;
```

```
MariaDB [azteca]> CALL impuesto_pago(sumatodopagos());
```

resultado
2856

```
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.03 sec)
```

- Procedimiento que se introduce la funcion sumatodocompras como parametro de entrada y muestra el total de compras con el impuesto incluido

Resultado

```
MariaDB [azteca]> DROP PROCEDURE IF EXISTS impuesto_compra;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [azteca]> DELIMITER //
```

```
MariaDB [azteca]> CREATE PROCEDURE impuesto_compra(IN valor INT)
```

```
-> BEGIN
```

```
-> DECLARE operacion INT;
```

```
-> DECLARE resultado INT;
```

```
-> SET operacion=valor*0.21;
```

```
-> SET resultado=valor - operacion;
```

```
-> SELECT resultado;
```

```
-> END//
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [azteca]> DELIMITER ;
```

```
MariaDB [azteca]> CALL impuesto_compra(sumatodocompras());
```

resultado
635950

```
1 row in set (0.00 sec)
```

```
Query OK, 0 rows affected (0.03 sec)
```

- **Funciones**
- Funcion de sumar el total de dinero recaudado con los alquileres

Resultado

```
MariaDB [azteca]> DROP FUNCTION IF EXISTS sumatodopagos;
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE FUNCTION sumatodopagos() RETURNS INTEGER
-> BEGIN
-> DECLARE casas INTEGER DEFAULT 0;
-> DECLARE pisos INTEGER DEFAULT 0;
-> DECLARE locales INTEGER DEFAULT 0;
-> DECLARE garajes INTEGER DEFAULT 0;
-> DECLARE resultado INTEGER DEFAULT 0;
-> SELECT SUM(valor) FROM pagos_casas INTO casas;
-> SELECT SUM(valor) FROM pagos_pisos INTO pisos;
-> SELECT SUM(valor) FROM pagos_locales INTO locales;
-> SELECT SUM(valor) FROM pagos_garajes INTO garajes;
-> SELECT SUM(casas + pisos + locales + garajes) INTO resultado;
-> RETURN resultado;
-> END//
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> SELECT sumatodopagos();
+-----+
| sumatodopagos() |
+-----+
|          3615 |
+-----+
1 row in set (0.00 sec)
```

- Funcion de sumar todas las compras de las casas, piso, locales y garajes

Resultado

```
MariaDB [azteca]> DROP FUNCTION IF EXISTS sumatodocompras;
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE FUNCTION sumatodocompras() RETURNS INTEGER
-> BEGIN
-> DECLARE casas INTEGER DEFAULT 0;
-> DECLARE pisos INTEGER DEFAULT 0;
-> DECLARE locales INTEGER DEFAULT 0;
-> DECLARE garajes INTEGER DEFAULT 0;
-> DECLARE resultado INTEGER DEFAULT 0;
-> SELECT SUM(valor) FROM compras_casas INTO casas;
-> SELECT SUM(valor) FROM compras_pisos INTO pisos;
-> SELECT SUM(valor) FROM compras_locales INTO locales;
-> SELECT SUM(valor) FROM compras_garajes INTO garajes;
-> SELECT SUM(casas + pisos + locales + garajes) INTO resultado;
-> RETURN resultado;
-> END//
Query OK, 0 rows affected (0.01 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> SELECT sumatodocompras();
+-----+
| sumatodocompras() |
+-----+
|          805000 |
+-----+
1 row in set (0.00 sec)
```

- Funcion sumar todo lo recaudado

Resultado

```
MariaDB [azteca]> DROP FUNCTION IF EXISTS sumatodo;
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE FUNCTION sumatodo() RETURNS INTEGER
    -> BEGIN
    -> DECLARE resultado INTEGER DEFAULT 0;
    -> SELECT SUM(sumatodopagos() + sumatodocompras()) INTO resultado;
    -> RETURN resultado;
    -> END//
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> SELECT sumatodo();
+-----+
| sumatodo() |
+-----+
|      808615 |
+-----+
1 row in set (0.01 sec)
```


- **Cursores**
- Sacar el valor total de pagos del alquiler mas caro de la casa

Resultado

```
MariaDB [azteca]> DROP PROCEDURE IF EXISTS sumapagocasa;
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE PROCEDURE sumapagocasa()
-> BEGIN
-> DECLARE seguir BOOLEAN DEFAULT TRUE;
-> DECLARE miid INT DEFAULT 0;
-> DECLARE midireccion TEXT DEFAULT 0;
-> DECLARE mimetros INT DEFAULT 0;
-> DECLARE mivalor INT DEFAULT 0;
-> DECLARE maxid INT DEFAULT 0;
-> DECLARE maxdireccion TEXT DEFAULT 0;
-> DECLARE maxmetros INT DEFAULT 0;
-> DECLARE maxvalor INT DEFAULT 0;
-> DECLARE cursormaxpagos CURSOR FOR SELECT id_casa, direccion, metros, (SELECT sum(valor) FROM pagos_casas WH
ERE pagos_casas.alquiler=casas.id_casa) AS "Valor total" FROM casas GROUP BY id_casa;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET seguir=FALSE;
-> OPEN cursormaxpagos;
-> WHILE seguir DO
->   FETCH cursormaxpagos INTO miid, midireccion, mimetros, mivalor;
->   IF seguir THEN
->     IF mivalor>maxvalor THEN
->       SET maxid=miid;
->       SET maxdireccion=midireccion;
->       SET maxmetros=mimetros;
->       SET maxvalor=mivalor;
->     END IF;
->   END IF;
-> END WHILE;
-> CLOSE cursormaxpagos;
-> SELECT CONCAT("La casa con ID ", maxid, " en la direccion ", maxdireccion, " con ", maxmetros, " metros cua
drados, tiene en total ", maxvalor, " en total de pagos") AS "Resultado";
-> END//
Query OK, 0 rows affected (0.02 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> CALL sumapagocasa();

+-----+
| Resultado |
+-----+
| La casa con ID 2 en la direccion Calle Prueba 2 con 130 metros cuadrados, tiene en total 600 en total de pagos |
+-----+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.10 sec)

MariaDB [azteca]>
```

- Lo mismo pero para pisos

Resultado

```
MariaDB [azteca]> DROP PROCEDURE IF EXISTS sumapagopiso;
Query OK, 0 rows affected (0.06 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE PROCEDURE sumapagopiso()
-> BEGIN
-> DECLARE seguir BOOLEAN DEFAULT TRUE;
-> DECLARE miid INT DEFAULT 0;
-> DECLARE midireccion TEXT DEFAULT 0;
-> DECLARE mimetros INT DEFAULT 0;
-> DECLARE miplanta INT(1) DEFAULT 0;
-> DECLARE mipuerta VARCHAR(1) DEFAULT 0;
-> DECLARE mivalor INT DEFAULT 0;
-> DECLARE maxid INT DEFAULT 0;
-> DECLARE maxdireccion TEXT DEFAULT 0;
-> DECLARE maxmetros INT DEFAULT 0;
-> DECLARE maxplanta INT(1) DEFAULT 0;
-> DECLARE maxpuerta VARCHAR(1) DEFAULT 0;
-> DECLARE maxvalor INT DEFAULT 0;
-> DECLARE cursormaxpagos CURSOR FOR SELECT id_piso, direccion, metros, planta, puerta, (SELECT sum(valor) FROM
M pagos_pisos WHERE pagos_pisos.alquiler=pisos.id_piso) AS "Valor total" FROM pisos GROUP BY id_piso;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET seguir=FALSE;
-> OPEN cursormaxpagos;
-> WHILE seguir DO
->   FETCH cursormaxpagos INTO miid, midireccion, mimetros, miplanta, mipuerta, mivalor;
->   IF seguir THEN
->     IF mivalor>maxvalor THEN
->       SET maxid=miid;
->       SET maxdireccion=midireccion;
->       SET maxmetros=mimetros;
->       SET maxplanta=miplanta;
->       SET maxpuerta=mipuerta;
->       SET maxvalor=mivalor;
->     END IF;
->   END IF;
-> END WHILE;
-> CLOSE cursormaxpagos;
-> SELECT CONCAT("El piso con ID ", maxid, " en la direccion ", maxdireccion, " con ", maxmetros, " metros cuad
rados, con numero ", maxplanta, " y letra ", maxpuerta, " tiene en total ", maxvalor, " en total de pagos") AS "Res
ultado";
-> END//
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> CALL sumapagopiso();
```

```
+-----+
| Resultado |
+-----+
| El piso con ID 2 en la direccion Calle Prueba 5 con 80 metros cuadrados, con numero 3 y letra F tiene en total 54
0 en total de pagos |
+-----+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.11 sec)
```

- Lo mismo pero para locales

Resultado

```
MariaDB [azteca]> DROP PROCEDURE IF EXISTS sumapagolocal;
Query OK, 0 rows affected (0.05 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE PROCEDURE sumapagolocal()
-> BEGIN
-> DECLARE seguir BOOLEAN DEFAULT TRUE;
-> DECLARE miid INT DEFAULT 0;
-> DECLARE midireccion TEXT DEFAULT 0;
-> DECLARE mimetros INT DEFAULT 0;
-> DECLARE miservicio VARCHAR(20) DEFAULT 0;
-> DECLARE mivalor INT DEFAULT 0;
-> DECLARE maxid INT DEFAULT 0;
-> DECLARE maxdireccion TEXT DEFAULT 0;
-> DECLARE maxmetros INT DEFAULT 0;
-> DECLARE maxservicio VARCHAR(20) DEFAULT 0;
-> DECLARE maxvalor INT DEFAULT 0;
-> DECLARE cursormaxpagos CURSOR FOR SELECT id_local, direccion, metros, servicio,(SELECT sum(valor) FROM pagos_locales WHERE pagos_locales.alquiler=locales.id_local) AS "Valor total" FROM locales GROUP BY id_local;
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET seguir=FALSE;
-> OPEN cursormaxpagos;
-> WHILE seguir DO
->   FETCH cursormaxpagos INTO miid, midireccion, mimetros, miservicio, mivalor;
->   IF seguir THEN
->     IF mivalor>maxvalor THEN
->       SET maxid=miid;
->       SET maxdireccion=midireccion;
->       SET maxmetros=mimetros;
->       SET maxservicio=miservicio;
->       SET maxvalor=mivalor;
->     END IF;
->   END IF;
-> END WHILE;
-> CLOSE cursormaxpagos;
-> SELECT CONCAT("El local con ID ",maxid, " en la direccion ", maxdireccion, " con ", maxmetros, " metros cuadrados, con servicio de ", maxservicio, " tiene en total ", maxvalor, " en total de pagos") AS "Resultado";
-> END//
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> CALL sumapagolocal();

+-----+
| Resultado |
+-----+
| El local con ID 1 en la direccion Calle San Juan 20 con 200 metros cuadrados, con servicio de Discoteca tiene en total 1500 en total de pagos |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.08 sec)
```

- Lo mismo pero para garajes

Resultado

```
MariaDB [azteca]> DROP PROCEDURE IF EXISTS sumapagogaraje;
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE PROCEDURE sumapagogaraje()
-> BEGIN
-> DECLARE seguir BOOLEAN DEFAULT TRUE;
-> DECLARE miid INT DEFAULT 0;
-> DECLARE midireccion TEXT DEFAULT 0;
-> DECLARE mimetros INT DEFAULT 0;
-> DECLARE miplanta INT(1) DEFAULT 0;
-> DECLARE minumero INT(1) DEFAULT 0;
-> DECLARE mivalor INT DEFAULT 0;
-> DECLARE maxid INT DEFAULT 0;
-> DECLARE maxdireccion TEXT DEFAULT 0;
-> DECLARE maxmetros INT DEFAULT 0;
-> DECLARE maxplanta INT(1) DEFAULT 0;
-> DECLARE maxnumero INT(1) DEFAULT 0;
-> DECLARE maxvalor INT DEFAULT 0;
-> DECLARE cursormaxpagos CURSOR FOR SELECT id_garaje, direccion, metros, planta, numero, (SELECT sum(valor) FROM pagos_garajes WHERE pagos_garajes.alquiler=garajes.id_garaje) AS "Valor total" FROM garajes GROUP BY id_garaje;

-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET seguir=FALSE;
-> OPEN cursormaxpagos;
-> WHILE seguir DO
->   FETCH cursormaxpagos INTO miid, midireccion, mimetros, miplanta, minumero, mivalor;
->   IF seguir THEN
->     IF mivalor>maxvalor THEN
->       SET maxid=miid;
->       SET maxdireccion=midireccion;
->       SET maxmetros=mimetros;
->       SET maxplanta=miplanta;
->       SET maxnumero=minumero;
->       SET maxvalor=mivalor;
->     END IF;
->   END IF;
-> END WHILE;
-> CLOSE cursormaxpagos;
-> SELECT CONCAT("El garaje con ID ", maxid, " en la direccion ", maxdireccion, " con ", maxmetros, " metros cuadrados, con planta ", maxplanta, " y numero ", maxnumero, " tiene en total ", maxvalor, " en total de pagos") AS "Resultado";
-> END//
Query OK, 0 rows affected (0.00 sec)

MariaDB [azteca]> DELIMITER ;
MariaDB [azteca]> CALL sumapagogaraje();
```

```
+-----+
| Resultado |
+-----+
| El garaje con ID 1 en la direccion Calle Prueba 1 con 20 metros cuadrados, con planta 1 y numero 1 tiene en total 75 en total de pagos |
+-----+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.08 sec)
```

- **Disparadores**
- Comprobar la fecha que se introduce, si se introduce una fecha anterior a la actual, se ponga la actual en la tabla pagos_casas

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchapagocasa;
Query OK, 0 rows affected (0.03 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchapagocasa BEFORE INSERT ON pagos_casas FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.03 sec)
```

Comprobar la fecha que se introduce, si se introduce una fecha anterior a la actual, se ponga la actual en la tabla compras_casas

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchacompracasa;
Query OK, 0 rows affected, 1 warning (0.00 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchacompracasa BEFORE INSERT ON compras_casas FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.03 sec)
```

- Lo mismo para pisos

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchapagopiso;
Query OK, 0 rows affected (0.02 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchapagopiso BEFORE INSERT ON pagos_pisos FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchacomprapiso;
Query OK, 0 rows affected (0.07 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchacomprapiso BEFORE INSERT ON compras_pisos FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.02 sec)
```

- Lo mismo para locales

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchapagolocal;
Query OK, 0 rows affected (0.05 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchapagolocal BEFORE INSERT ON pagos_locales FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.03 sec)
```

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchacompralocal;
Query OK, 0 rows affected (0.02 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchacompralocal BEFORE INSERT ON compras_locales FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.01 sec)
```

- Lo mismo para garajes

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchapagogaraje;
Query OK, 0 rows affected (0.02 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchapagogaraje BEFORE INSERT ON pagos_garajes FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.02 sec)
```

```
MariaDB [azteca]> DROP TRIGGER IF EXISTS comfchacompragaraje;
Query OK, 0 rows affected (0.02 sec)

MariaDB [azteca]> DELIMITER //
MariaDB [azteca]> CREATE TRIGGER comfchacompragaraje BEFORE INSERT ON compras_garajes FOR EACH ROW
-> BEGIN
-> IF NEW.fecha<curdate() THEN
-> SET NEW.fecha=curdate();
-> END IF;
-> END//
Query OK, 0 rows affected (0.09 sec)
```

- **Eventos**
- Evento que guarda el resultado de la funcion sumatodopagos cada mes en la tabla suma_pagos_mes

```
CREATE TABLE suma_pagos_mes (  
id_pago INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
fecha DATE NOT NULL,  
valor INT NOT NULL  
)ENGINE=InnoDB;
```

```
MariaDB [azteca]> CREATE EVENT evento_sumapagos_mes  
-> ON SCHEDULE EVERY 1 MONTH  
-> starts "2017-01-01" ENABLE  
-> DO INSERT INTO azteca.suma_pagos_mes(fecha, valor) VALUES (CURDATE(), sumatodopagos());  
Query OK, 0 rows affected (0.00 sec)
```

Resultado

```
MariaDB [azteca]> select * from suma_pagos_mes;  
+-----+-----+-----+  
| id_pago | fecha      | valor |  
+-----+-----+-----+  
|      1 | 2017-03-13 | 3615 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

- Evento que guarda el resultado de la funcion sumatodocompras cada mes en la tabla suma_compras_mes

```
CREATE TABLE suma_compras_mes (  
id_pago INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
fecha DATE NOT NULL,  
valor INT NOT NULL  
)ENGINE=InnoDB;
```

```
MariaDB [azteca]> CREATE EVENT evento_sumacompras_mes  
-> ON SCHEDULE EVERY 1 MONTH  
-> starts "2017-01-01" ENABLE  
-> DO INSERT INTO azteca.suma_compras_mes(fecha, valor) VALUES (CURDATE(), sumatodocompras());  
Query OK, 0 rows affected (0.00 sec)
```

Resultado

```
MariaDB [azteca]> select * from suma_compras_mes;  
+-----+-----+-----+  
| id_pago | fecha      | valor |  
+-----+-----+-----+  
|      1 | 2017-03-13 | 805000 |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```