



PROYECTO ASGB

Eduardo de Lamo Téllez

ÍNDICE

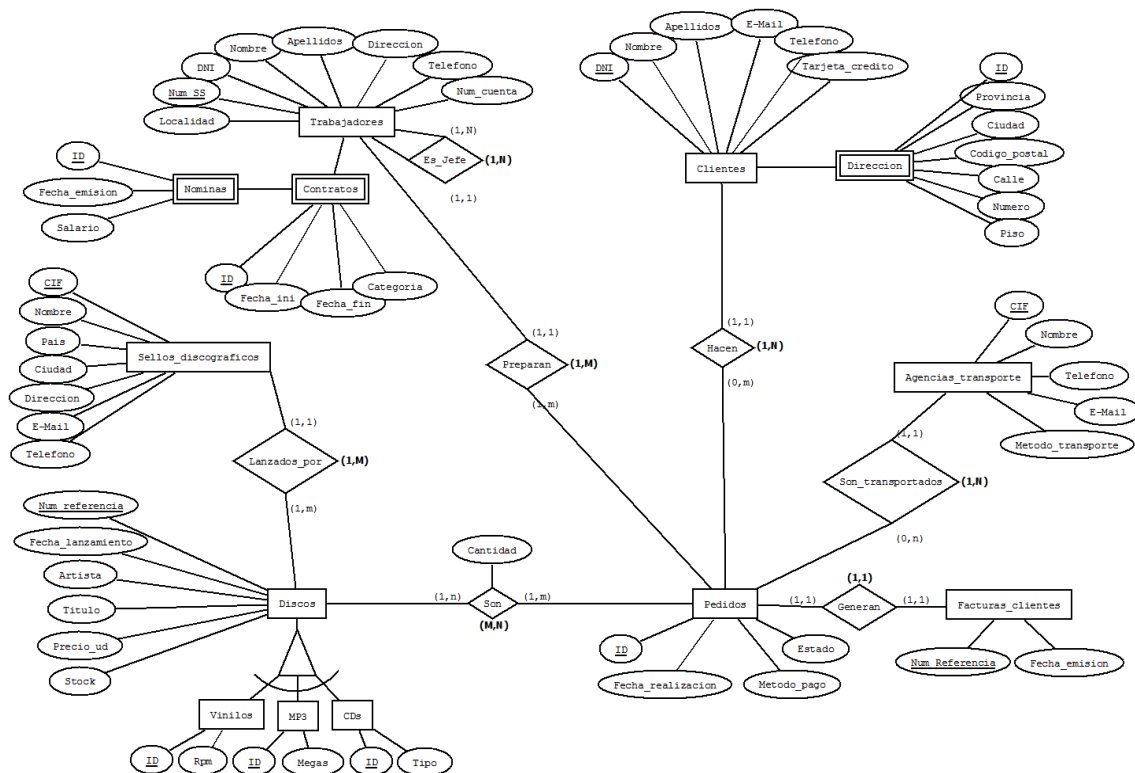
Planteamiento.....	1
Diagrama E/R	2
Paso a tablas	2
Creación de la base de datos y las tablas.....	3
Inserción de datos en las tablas	7
Consultas	11
Exportación de la base de datos	15
Logs de MySQL Windows	15
Logs de MySQL Linux.....	18
Conexiones remotas.....	21
Comando mysqlcheck	22
Usuarios y permisos	23
Gestión de recursos.....	27
Borrar usuarios.....	27
Funciones	28
Procedimientos	30
Vistas	35
Triggers.....	36
Eventos	38

Planteamiento

Crearemos un diseño E/R que nos permita guardar toda la información de una tienda de discos para DJ's que vende sus productos a través de la red en España.

- Guardaremos los datos de los clientes como son; DNI, nombre, apellidos, correo electrónico, teléfono, tarjeta de crédito y dirección (provincia, ciudad, CP, calle, número, piso, etc.)
- De cada disco debemos almacenar; su número de referencia, sello discográfico, artista, título, fecha de lanzamiento, formato (Vinilo, CD o MP3), cuantos hay en stock y el precio de cada unidad.
- Trabajamos con distintos sellos discográficos a los cuales compramos los discos, anotamos; su CIF, nombre, país, ciudad, dirección, teléfono y e-mail.
- De los pedidos realizados por los clientes guardamos; la fecha en la que se realizan, los artículos pedidos, el método de pago, método de transporte y el estado (en preparación, enviado, entregado, etc.)
- Almacenamos las facturas de los clientes en las que anotaremos su número de referencia y la fecha de emisión.
- Trabajamos con agencias de transporte, de las cuales debemos saber; su CIF, nombre, teléfono, e-mail y modo de transporte que ofrece.
- En la empresa hay 3 tipos de trabajadores; los programadores web, los administrativos y los preparadores de los pedidos (logística). Almacenaremos su DNI, número de la SS, nombre, apellidos, localidad, dirección, teléfono, número de cuenta bancaria y contrato. Además almacenamos las nóminas de todos ellos en las cuales apuntaremos su fecha de emisión y el sueldo.

Diagrama E/R



Paso a tablas

Trabajadores (Numero_SS, DNI, Nombre, Apellidos, Localidad, Direccion, Telefono, Num_cuenta, Num_SS_es_Jefe)

Contratos (ID, Fecha_ini, Fecha_fin, Categoria, Num_SS_Trabajadores)

Nominas (ID, Fecha_emision, Salario, ID_Contrato)

Cientes (DNI, Nombre, Apellidos, E-Mail, Telefono, Num_tarjeta_credito)

Direccion (ID, Provincia, Ciudad, Codigo_postal, Calle, Numero, Piso, DNI_Cliente)

Pedidos (ID, Fecha_realizacion, Metodo_pago, Estado, DNI_Cliente,

CIF Agencia Transporte, Num SS trabajador prepara)

Facturas_clientes (Num_Referencia, Fecha_emision, ID_Pedido)

Agencias transporte (CIF, Nombre, Telefono, E-Mail, Metodo transporte)

Discos (Num_referencia, Fecha_lanz, Artista, Titulo, Precio_und, Stock,

CIF Sellos discograficos)

Vinilos (ID, Rpm, Num_Referencia_Discos)

MP3 (ID, Megas, Num, Referencia, Discos)

CD (ID, Tipo, Num_Referencia_Discos)

Sellos discograficos (CIF, Nombre, Pais, Ciudad, Direccion, E-Mail, Telefono)

Discos son Pedidos (ID, ID Pedidos, Num Referencia Discos, Cantidad)

Creación de la base de datos y las tablas

```
CREATE DATABASE tienda_discos;  
USE tienda_discos;  
SET FOREIGN_KEY_CHECKS=0; /*Desactivo claves foráneas*/
```

```
/*Tabla trabajadores*/
```

```
CREATE TABLE trabajadores (  
numero_ss DECIMAL(12) NOT NULL PRIMARY KEY,  
dni VARCHAR(9) NOT NULL,  
nombre VARCHAR(20),  
apellidos VARCHAR(40),  
localidad VARCHAR(30),  
direccion VARCHAR(50),  
telefono DECIMAL(9),  
num_cuenta VARCHAR(20),  
num_ss_jefe DECIMAL(12),  
FOREIGN KEY (num_ss_jefe) REFERENCES trabajadores(numero_ss) ON DELETE  
RESTRICT ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

```
/*Tabla contratos*/
```

```
CREATE TABLE contratos (  
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
fecha_inicio DATE NOT NULL,  
fecha_fin DATE,  
categoria ENUM ("Administrativo", "Logistica", "Programador", "Jefe"),  
num_ss_trabajador DECIMAL(12),  
FOREIGN KEY (num_ss_trabajador) REFERENCES trabajadores(numero_ss) ON DELETE  
RESTRICT ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

```
/*Tabla nóminas*/
```

```
CREATE TABLE nominas (  
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
fecha_emision DATE NOT NULL,  
salario DECIMAL(7,2),  
id_contrato INT UNSIGNED,  
FOREIGN KEY (id_contrato) REFERENCES contratos(id) ON DELETE RESTRICT ON UPDATE  
CASCADE  
)ENGINE=InnoDB;
```

*/*Tabla clientes*/*

```
CREATE TABLE clientes (  
dni VARCHAR(9) NOT NULL PRIMARY KEY,  
nombre VARCHAR(20),  
apellidos VARCHAR(40),  
e_mail VARCHAR(40),  
telefono DECIMAL(9),  
num_tarjeta_credito DECIMAL(16) NOT NULL  
)ENGINE=InnoDB;
```

*/*Tabla dirección clientes*/*

```
CREATE TABLE direccion_clientes (  
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
provincia VARCHAR(30),  
ciudad VARCHAR(30),  
codigo_postal DECIMAL(5),  
calle VARCHAR(30),  
numero VARCHAR(5),  
piso VARCHAR(3),  
dni_cliente VARCHAR(9),  
FOREIGN KEY (dni_cliente) REFERENCES clientes(dni) ON DELETE RESTRICT ON UPDATE  
CASCADE  
)ENGINE=InnoDB;
```

*/*Tabla pedidos*/*

```
CREATE TABLE pedidos (  
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
fecha_realizacion DATE NOT NULL,  
metodo_pago ENUM ("Tarjeta", "Reembolso", "PayPal"),  
estado VARCHAR(20),  
dni_cliente VARCHAR(9),  
FOREIGN KEY (dni_cliente) REFERENCES clientes(dni) ON DELETE RESTRICT ON UPDATE  
CASCADE,  
cif_agencia_transporte VARCHAR(9),  
FOREIGN KEY (cif_agencia_transporte) REFERENCES agencias_transporte(cif) ON  
DELETE RESTRICT ON UPDATE CASCADE,  
num_ss_trabajador_prepara DECIMAL(12),  
FOREIGN KEY (num_ss_trabajador_prepara) REFERENCES trabajadores(numero_ss) ON  
DELETE RESTRICT ON UPDATE CASCADE  
)ENGINE=InnoDB;
```

*/*Tabla facturas clientes*/*

```
CREATE TABLE facturas_clientes (  
  num_referencia INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  fecha_emision DATE NOT NULL,  
  id_pedido INT UNSIGNED,  
  FOREIGN KEY (id_pedido) REFERENCES pedidos(id) ON DELETE RESTRICT ON UPDATE  
  CASCADE  
  )ENGINE=InnoDB;
```

*/*Tabla agencias transporte*/*

```
CREATE TABLE agencias_transporte (  
  cif VARCHAR(9) NOT NULL PRIMARY KEY,  
  nombre VARCHAR(20), telefono DECIMAL(9),  
  e_mail VARCHAR(40),  
  metodo_transporte VARCHAR(30)  
  )ENGINE=InnoDB;
```

*/*Tabla sellos discográficos*/*

```
CREATE TABLE sellos_discograficos (  
  cif VARCHAR(9) NOT NULL PRIMARY KEY,  
  nombre VARCHAR(30), pais VARCHAR(30),  
  ciudad VARCHAR(30), direccion VARCHAR(50),  
  e_mail VARCHAR(40),  
  telefono DECIMAL(9) NOT NULL  
  )ENGINE=InnoDB;
```

*/*Tabla discos*/*

```
CREATE TABLE discos (  
  num_referencia INT UNSIGNED NOT NULL PRIMARY KEY,  
  fecha_lanzamiento DATE,  
  artista VARCHAR(30),  
  titulo VARCHAR (50),  
  precio_unidad DECIMAL(5,2),  
  stock INT UNSIGNED,  
  cif_sello_discografico VARCHAR(9),  
  FOREIGN KEY (cif_sello_discografico) REFERENCES sellos_discograficos(cif) ON DELETE  
  RESTRICT ON UPDATE CASCADE  
  )ENGINE=InnoDB;
```

*/*Tabla vinilos*/*

```
CREATE TABLE vinilos (  
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
  rpm ENUM ("33", "45"), /*Puenden ser de 33 o 45 rpm*/  
  num_ref_disco INT UNSIGNED,  
  FOREIGN KEY (num_ref_disco) REFERENCES discos(num_referencia) ON DELETE  
  RESTRICT ON UPDATE CASCADE  
  )ENGINE=InnoDB;
```

```

/*Tabla mp3*/
CREATE TABLE mp3 (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
megas DECIMAL(3,1),
num_ref_disco INT UNSIGNED,
FOREIGN KEY (num_ref_disco) REFERENCES discos(num_referencia) ON DELETE
RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB;

```

```

/*Tabla CD's*/
CREATETABLE cds(
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
tipo ENUM ("CD", "DVD", "BLUE-RAY"), /*Pueden estar en formato CD, DVD,
BLUE-RAY*/
num_ref_disco INT UNSIGNED,
FOREIGN KEY (num_ref_disco) REFERENCES discos(num_referencia) ON DELETE
RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB;

```

```

/*Tabla discos son pedidos*/
CREATE TABLE discos_pedidos (
id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
id_pedido INT UNSIGNED,
FOREIGN KEY (id_pedido) REFERENCES pedidos(id) ON DELETE RESTRICT ON UPDATE
CASCADE,
num_ref_disco INT UNSIGNED,
FOREIGN KEY (num_ref_disco) REFERENCES discos(num_referencia) ON DELETE
RESTRICT ON UPDATE CASCADE,
cantidad INT UNSIGNED
)ENGINE=InnoDB;

```

```

SET FOREIGN_KEY_CHECKS=1; /*Activo claves foráneas*/

```


Inserción de datos en las tablas

```
USE tienda_discos;  
SET FOREIGN_KEY_CHECKS=0;
```

```
/*Tabla trabajadores*/
```

```
INSERT INTO trabajadores (numero_ss, dni, nombre, apellidos, localidad, direccion,  
telefono, num_cuenta, num_ss_jefe) VALUES  
(865325698541, "78542651C", "Antonio", "Lopez Menchen", "Valdepeñas", "C/  
CabañasNº16", 656324587, 58965412325478965412, 757575757575),  
(754625986532, "78411458J", "Guillermo", "Gimenez Sanchez", "Membrilla", "C/ Real  
Nº 76", 654789874, 56589874123232145689, 757575757575),  
(898745145632, "74445882R", "Laura", "Gomez Marquez", "Santa Cruz", "C/ Ramirez  
Lasala Nº 67", 784547893, 12454632569874589658, 757575757575),  
(748489563254, "68955412Z", "Jaime", "Señoret Señoret", "Villamanrique", "C/  
Bataneros Nº 40", 658989712, 75698456325654123699, 757575757575),  
(125458976326, "58479321K", "Romualdo", "Conejo Pellejo", "Torrenueva", "C/ Postas  
Nº 104", 789456123, 12365258975632449582, 757575757575),  
(896573652145, "76589425T", "Anabel", "Tellez Gormaz", "Santa Cruz", "C/ Esperanza  
Nº 33", 658547899, 84512365985521457893, 757575757575),  
(877963325411, "41477896L", "Jesus", "Malpica Fernandez", "Santa Cruz", "C/ Castillo  
Nº 90", 696696696, 87963114523256478965, 757575757575),  
(956832564785, "58463215G", "Esperanza", "Dominguez Bravo", "Valdepeñas", "C/  
Datiles Nº 57 1º B", 753951852, 26589636974114789652, 757575757575),  
(596832148956, "75369894V", "Fernando", "Barrajon Huertas", "Moral", "C/ Carneros  
Nº 23", 685225897, 78998745552145632594, 757575757575),  
(757575757575, "75889364F", "Eduardo", "de Lamo Tellez", "Santa Cruz", "C/ Cervantes  
Nº 75 2º C", 753357753, 96589658965893214785, 757575757575);
```

```
/*Tabla contratos*/
```

```
INSERT INTO contratos (fecha_inicio, fecha_fin, categoria, num_ss_trabajador) VALUES  
("2015-5-1", "2018-5-1", "Programador", 865325698541), ("2017-4-5", "2019-4-5",  
"Administrativo", 754625986532),  
("2014-9-8", "2020-9-8", "Programador", 898745145632), ("2015-1-15", "2019-1-15",  
"Administrativo", 748489563254),  
("2016-8-24", "2020-8-24", "Programador", 125458976326), ("2018-1-1", "2020-1-1",  
"Administrativo", 896573652145),  
("2015-12-5", "2019-12-5", "Logistica", 877963325411), ("2017-5-21", "2020-5-21",  
"Logistica", 956832564785),  
("2016-2-5", "2019-2-5", "Logistica", 596832148956), ("2014-4-1", "2024-4-1", "Jefe",  
757575757575);
```

/*Tabla nóminas*/

```
INSERT INTO nominas (fecha_emision, salario, id_contrato) VALUES
("2015-5-1", 950.80, 1), ("2016-4-1", 970.50, 2), ("2015-6-1", 965.25, 3), ("2017-9-1",
974.25, 4), ("2018-1-1", 956.70, 5),
("2016-7-1", 985.65, 6), ("2017-8-1", 974.50, 7), ("2017-11-1", 967.45, 8), ("2018-2-1",
992.45, 9), ("2016-3-1", 948.80, 10);
```

/*Tabla clientes*/

```
INSERT INTO clientes (dni, nombre, apellidos, e_mail, telefono, num_tarjeta_credito)
VALUES
("78563256W", "Eladio", "Cortes Perez", "eladiocortes@gmail.com", 926585413,
8965321458974563),
("65894125P", "Amanda", "Trujillo Incertis", "amanda56@hotmail.com", 755213694,
5698541230259787),
("71698331S", "Carmen", "Gracia Laguna", "carmengracia30@gmail.com", 654159735,
8954732001459036),
("58793214F", "Rodrigo", "Garcia Carrion", "rodricarrion35@hotmail.com", 693214588,
2310056987453014),
("78990012H", "Francisco", "Osuna Galvez", "pacoosuna75@gmail.com", 663201484,
4552001326985471),
("48552589K", "Noelia", "Fernandez Ochoa", "noefbk@hotmail.com", 759824563,
6987452001145698),
("25898741Z", "Fernando", "Fajardo Cepa", "fernandocepas@gmail.com", 655887412,
8788963321258475),
("73550486B", "Diana", "Romero Clemente", "diana57@hotmail.com", 687522456,
8874566320025897);
```

/*Tabla dirección clientes*/

```
INSERT INTO direccion_clientes (provincia, ciudad, codigo_postal, calle, numero, piso,
dni_cliente) VALUES
("Albacete", "Villarobledo", 14520, "Capitan Casado", 54, "0", "78563256W"),
("Madrid", "Madrid", 50000, "Alcala", 452, "1ºC", "65894125P"),
("Barcelona", "Cornella", 85690, "Buendia", 104, "4ºE", "71698331S"),
("Toledo", "Torrijos", 40150, "San Sebastian", 58, "0", "58793214F"),
("Burgos", "Aranda de Duero", 80560, "Marques de Verona", 145, "3ºF", "78990012H"),
("Valencia", "Sedaví", 70155, "Ramirez Lasala", 71, "2ºE", "48552589K"), ("Granada",
"Durcal", 14580, "San Marcos", 32, "0", "25898741Z"),
("Badajoz", "Badajoz", 10257, "Carneros", 168, "3ºC", "73550486B");
```

/*Tabla pedidos*/

```
INSERT INTO pedidos (fecha_realizacion, metodo_pago, estado, dni_cliente,
cif_agencia_transporte, num_ss_trabajador_prepara) VALUES
("2015-6-22", "Tarjeta", "Entregado", "78563256W", 158966400, 877963325411),
("2015-9-17", "Reembolso", "Entregado", "65894125P", 325488706, 956832564785),
("2017-12-15", "Tarjeta", "Enviado", "71698331S", 102204583, 596832148956),
("2018-1-15", "Reembolso", "En transito", "58793214F", 325488706, 877963325411),
("2018-2-3", "Tarjeta", "Enpreparacion", "78990012H", 102204583, 596832148956),
```

```
("2015-8-4", "PayPal", "Entregado", "78563256W", 158966400, 877963325411),
("2015-12-9", "Reembolso", "Entregado", "65894125P", 325488706, 956832564785),
("2016-1-29", "Tarjeta", "Enviado", "71698331S", 102204583, 596832148956),
("2018-3-21", "Reembolso", "En transito", "58793214F", 325488706, 877963325411),
("2018-3-22", "PayPal", "En preparacion", "78990012H", 102204583, 596832148956);
```

*/*Tabla facturas clientes*/*

```
INSERT INTO facturas_clientes (num_referencia, fecha_emision, id_pedido) VALUES
(00001, "2015-6-23", 1), (00002, "2015-9-18", 2), (00003, "2017-12-16", 3), (00004,
"2018-1-16", 4), (00005, "2018-2-4", 5),
(00006, "2015-8-5", 6), (00007, "2015-12-10", 7), (00008, "2016-1-30", 8), (00009,
"2018-3-22", 9), (00010, "2018-3-23", 10);
```

*/*Tabla agencias transporte*/*

```
INSERT INTO agencias_transporte (cif, nombre, telefono, e_mail, metodo_transporte)
VALUES
(158966400, "Transportalia", 665521307, "transportalia@hotmail.com", "24 horas"),
(325488706, "Rapidtrans", 771125680, "rapidtrans@hotmail.com", "Urgente"),
(102204583, "Inmotrans", 633011589, "inmotrans@gmail.com", "Peninsula y
Balears");
```

*/*Tabla sellos discográficos*/*

```
INSERT INTO sellos_discograficos (cif, nombre, pais, ciudad, direccion, e_mail, telefono)
VALUES
(233147852, "Drumcode", "Belgica", "Amberes", "C/ Heeinek Nº 75",
"drumcode@drumcode.com", 699014425),
(847710025, "Kompact", "Alemania", "Berlin", "C/ Werebeen Nº 234",
"kompact@gmail.com", 862014589),
(554012397, "Footwork", "Reino Unido", "Londres", "C/ Stringer Nº 28",
"footwork@hotmail.com", 454201269),
(772201596, "Suara", "España", "Valencia", "C/ Cid Campeador 356 3º H",
"suara@discografica.es", 695224701),
(654630114, "Mindshake", "España", "Madrid", "C/ Preciados 350 7º A",
"mindshake@records.es", 652004778);
```

*/*Tabla discos*/*

```
INSERT INTO discos (num_referencia, fecha_lanzamiento, artista, titulo, precio_unidad,
stock, cif_sello_discografico) VALUES
(56789, "2017-9-5", "Paco Osuna", "Mindblowing", 8.99, 100, 654630114), (45609,
"2018-1-30", "Nicole Moudaber", "Spring", 7.60, 50, 654630114),
(57412, "2017-5-16", "Sam Michaels", "Venore", 7.99, 124, 554012397), (37890,
"2018-2-1", "Oscar Mulero", "Frio y Calor", 6.70, 54, 772201596),
(89651, "2018-1-15", "Amelie Lens", "Awakenings", 8.65, 142, 233147852), (89564,
"2017-11-5", "Stephan Bozin", "Strand", 7.40, 80, 847710025),
(22869, "2018-2-15", "Jay Lumen", "Inside", 8.80, 94, 554012397), (47258, "2018-1-
23", "Carl Cox", "See You", 6.75, 41, 772201596),
(74982, "2017-5-22", "Mario Ochoa", "Aldebaran", 5.35, 50, 654630114), (83256,
```

```
"2016-8-19", "Weska", "Voyage", 5.90, 32, 233147852),
(45896,"2015-3-30","EliBrown","Sumatra",6.40, 66, 847710025), (14589,"2016-10-
29", "Nick Curly", "Dont Takers", 8.50, 71, 554012397);
```

```
/*Tabla Vinilos*/
```

```
INSERT INTO vinilos (rpm, num_ref_disco) VALUES
("45", 56789), ("33", 45609), ("33", 57412), ("45", 37890), ("45", 89651), ("33", 89564),
("45", 22869), ("45", 47258),
("33", 74982), ("33", 83256), ("45", 45896), ("45", 14589);
```

```
/*Tabla MP3*/
```

```
INSERT INTO mp3 (megas, num_ref_disco) VALUES
(12.5, 56789), (13.3, 45609), (8.5, 57412), (9.2, 37890), (14.4, 89651), (8.9, 89564),
(10.2, 22869), (12.1, 47258),
(11.5, 74982), (8.9, 83256), (9.8, 45896), (10.3, 14589);
```

```
/*Tabla CD's*/
```

```
INSERT INTO cds (tipo, num_ref_disco) VALUES
("CD", 56789), ("CD", 45609), ("DVD", 57412), ("Blue-Ray", 37890), ("Blue-Ray",
89651), ("CD", 89564), ("DVD", 22869), ("DVD", 47258),
("CD", 74982), ("Blue-Ray", 83256), ("CD", 45896), ("CD", 14589);
```

```
/*Tabla discos son pedidos*/
```

```
INSERT INTO discos_pedidos (id_pedido, num_ref_disco, cantidad) VALUES (1, 56789,
1), (1, 57412, 1), (1, 89564, 1), (1, 22869, 1), (1, 83256, 1), (1, 45896, 1), (1,
14589, 1),
(2, 56789, 1), (2, 45609, 1), (2, 57412, 1), (2, 37890, 1), (2, 89651, 1), (2, 89564, 1), (2,
22869, 1),
(3, 22869, 1), (3, 47258, 1), (3, 74982, 1), (3, 83256, 1), (3, 45896, 1), (3, 47258, 1), (3,
14589, 1),
(4, 56789, 1), (4, 45609, 1), (4, 57412, 1), (4, 37890, 1), (4, 89651, 1), (4, 89564, 1), (4,
22869, 1),
(5, 22869, 1), (5, 47258, 1), (5, 74982, 1), (5, 83256, 1), (5, 45896, 1), (5, 47258, 1), (5,
14589, 1),
(6, 56789, 1), (6, 57412, 1), (6, 89651, 1), (6, 22869, 1), (6, 74982, 1), (6, 45896, 1), (6,
45609, 1),
(7, 45609, 2), (7, 37890, 2), (7, 89564, 2), (7, 47258, 2), (7, 83256, 2), (7, 14589, 2), (7,
74982, 2),
(8, 56789, 3), (8, 57412, 3), (8, 89651, 3), (8, 22869, 3), (8, 74982, 3), (8, 45896, 3), (8,
45609, 3),
(9, 45609, 2), (9, 37890, 2), (9, 89564, 2), (9, 47258, 2), (9, 83256, 2), (9, 14589, 2), (9,
74982, 2),
(10, 56789, 1), (10, 57412, 1), (10, 89651, 1), (10, 22869, 1), (10, 74982, 1), (10, 45896,
1), (10, 45609, 1);
```

```
SET FOREIGN_KEY_CHECKS=1;
```

Consultas

Número de discos distintos que se han solicitado en cada pedido

```
select id_pedido as "Num Pedido", count(*) as "Discos distintos" from discos_pedidos  
group by id_pedido;
```

Trabajadores (nombre y apellidos) que son de la localidad Santa Cruz ordenados alfabéticamente por nombre

```
select nombre, apellidos from trabajadores where localidad="Santa Cruz" order by  
nombre asc;
```

Ver disco más caro (artista, título y precio unidad)

```
select artista, titulo, precio_unidad as "Precio" from discos order by precio_unidad desc  
limit 1;
```

Ver disco más barato (artista, título y precio unidad)

```
select artista, titulo, precio_unidad as "Precio" from discos order by precio_unidad asc  
limit 1;
```

Ver todas las nóminas emitidas en el año 2015

```
select * from nominas where fecha_emision between "2015-01-01" and "2015-12-31";
```

Ver pedidos realizados en 2018

```
select * from pedidos where fecha_realizacion>="2018-01-01" and  
fecha_realizacion<="2018-12-31";
```

Ver contratos que finalizan en 2020

```
select * from contratos where fecha_fin between "2020-01-01" and "2020-12-31";
```

Ver tabla clientes ordenada por apellidos

```
select * from clientes order by apellidos;
```

Ver en cuantos pedidos se ha solicitado cada disco y la cantidad total de unidades pedidas

```
select num_ref_disco, count(*) as "Veces pedido", sum(cantidad) as "Total unidades"  
from discos_pedidos group by num_ref_disco;
```

Ver clientes cuyo apellido empieza por g

```
select * from clientes where apellidos like "g%";
```

Ver clientes cuyo nombre empieza por "E o C"

```
select * from clientes where nombre regexp "^[E|C]";
```

Ver títulos de disco que tengan 7 letras exactas

```
select titulo from discos where titulo regexp "^[A-z]{7}$";
```

Ver discos con stock inferior a 10 (número de referencia, artista y título)

```
select num_referencia, artista, titulo from discos where stock<=10;
```

Ver el nombre y apellidos del cliente con DNI 65894125P además de la fecha de realización y el método de pago de sus pedidos (UNION).

```
select nombre, apellidos from clientes where dni="65894125P" union  
select fecha_realizacion, metodo_pago from pedidos where dni_cliente="65894125P";
```

Ver trabajadores con contrato categoría Programador (nombre y apellidos)

Join

```
select nombre, apellidos from trabajadores join contratos on  
numero_ss=num_ss_trabajador where categoria="Programador";
```

Subconsulta

```
select nombre, apellidos from trabajadores where numero_ss=(select  
num_ss_trabajador from contratos where categoria="Programador");
```

Ver discos lanzados por la discográfica Footwork (artista y título)

Join

```
select artista, titulo from discos join sellos_discograficos on cif_sello_discografico=cif  
where nombre="Footwork";
```

Subconsulta

```
select artista, titulo from discos where cif_sello_discografico=(select cif from  
sellos_discograficos where nombre="Footwork");
```

Ver que trabajador ha preparado cada pedido (nombre y apellidos)

Join

```
select id as "Num Pedido", nombre as "Nombre Preparador", apellidos from pedidos join  
trabajadores on num_ss_trabajador_prepara=numero_ss order by id;
```

Subconsulta

```
select id as "Num Pedido", (select nombre from trabajadores where  
numero_ss=num_ss_trabajador_prepara) as "Nombre Preparador", (select apellidos  
from trabajadores where numero_ss=num_ss_trabajador_prepara) as "Apellidos" from  
pedidos order by id;
```

Nombre y apellidos de la persona que ha realizado el pedido número 1 y su fecha de realización

Join

```
select nombre, apellidos, fecha_realizacion from pedidos join clientes on  
dni=dni_cliente where id=1;
```

Subconsulta

```
select (select nombre from clientes where dni=dni_cliente) as "Nombre", (select  
apellidos from clientes where dni=dni_cliente) as "Apellidos", fecha_realizacion from  
pedidos where id=1;
```

Cientes que han realizado pedidos (nombre, apellidos y dni)

Join

```
select distinct nombre, apellidos, dni from clientes join pedidos on dni=dni_cliente;
```

Subconsulta

```
select nombre, apellidos, dni from clientes where dni in (select dni_cliente from pedidos where dni_cliente=dni);
```

Cientes que no han realizado ningún pedido (nombre, apellidos y dni)

Join

```
select nombre, apellidos, dni from clientes left join pedidos on dni=dni_cliente where id is null;
```

Misma consulta que la anterior pero con right join

```
select nombre, apellidos, dni from pedidos right join clientes on dni_cliente=dni where id is null;
```

Subconsulta

```
select nombre, apellidos, dni from clientes where dni not in (select dni_cliente from pedidos where dni_cliente=dni);
```

Ver las direcciones del cliente Francisco

Join

```
select provincia, ciudad, codigo_postal, calle, numero, piso from direccion_clientes join clientes on dni=dni_cliente where nombre="Francisco";
```

Subconsulta

```
select provincia, ciudad, codigo_postal, calle, numero, piso from direccion_clientes where dni_cliente=(select dni from clientes where nombre="Fernando");
```

Ver los discos que tenemos en formato BLUE-RAY (artista y título)

Join

```
select artista, titulo from discos join cds on num_referencia=num_ref_disco where tipo="BLUE-RAY";
```

Subconsulta

```
select artista, titulo from discos where num_referencia in (select num_ref_disco from cds where tipo="BLUE-RAY");
```

Ver la cantidad que hemos vendido de cada disco (artista, título y cantidad)

Join

```
select artista, titulo, sum(cantidad) as "Vendidos" from discos join discos_pedidos on num_referencia=num_ref_disco group by num_referencia;
```

Subconsulta

```
select (select artista from discos where num_referencia=num_ref_disco) as "Artista", (select titulo from discos where num_referencia=num_ref_disco) as "Titulo", sum(cantidad) as "Vendidos" from discos_pedidos group by num_ref_disco;
```

Ver los discos que se han solicitado en el pedido numero 5 (artista, título y número de referencia)

Join

```
select distinct artista, titulo, num_referencia from discos join discos_pedidos on num_referencia=num_ref_disco where id_pedido=5;
```

Subconsulta

```
select artista, titulo, num_referencia from discos where num_referencia in (select num_ref_disco from discos_pedidos where num_ref_disco=num_referencia and id_pedido=5);
```

Ver el precio total de cada pedido (sin IVA)

Join

```
select id_pedido as "Num Pedido", sum(precio_unidad * cantidad) as "Total" from discos_pedidos join discos on num_ref_disco=num_referencia group by id_pedido;
```

Subconsulta

```
select id_pedido as "Num Pedido", sum(cantidad*(select precio_unidad from discos where num_referencia=num_ref_disco)) as "Importe" from discos_pedidos group by id_pedido;
```

Ver el precio total de cada pedido (con IVA 10%)

Join

```
select id_pedido as "Num Pedido", round(sum(precio_unidad * cantidad)*1.10,2) as "Total" from discos_pedidos join discos on num_ref_disco=num_referencia group by id_pedido;
```

Subconsulta

```
select id_pedido as "Num Pedido", round(sum(cantidad*(select precio_unidad * 1.21 from discos where num_referencia=num_ref_disco)),2) as "Importe" from discos_pedidos group by id_pedido;
```

Ver precio medio de los discos de cada pedido

Join

```
select id_pedido as "Num Pedido", round(sum(precio_unidad)*cantidad / sum(cantidad),2) as "Precio medio disco" from discos_pedidos join discos on num_ref_disco=num_referencia group by id_pedido;
```

Subconsulta

```
select id_pedido as "Num Pedido", round(sum((select precio_unidad from discos where num_referencia=num_ref_disco) * cantidad) / sum(cantidad),2) as "Importe" from discos_pedidos group by id_pedido;
```



Exportación de la base de datos

Comando para la exportación de la base de datos a un archivo .sql desde el CMD.

```
C:\> Símbolo del sistema

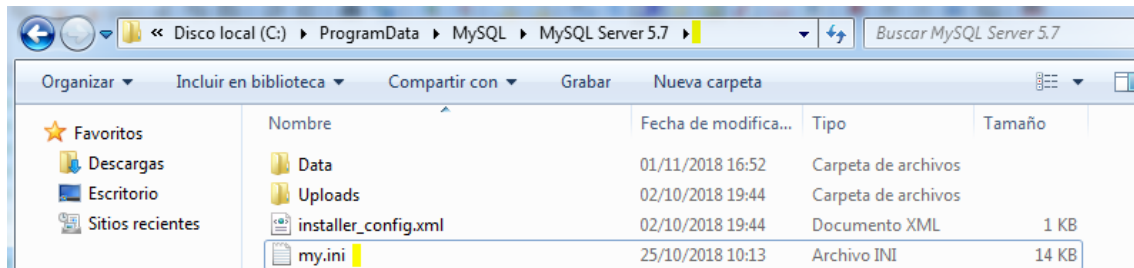
C:\Users\Edu\Downloads>mysqldump -u root -p tienda_discos > tienda_discos_backup.sql
Enter password: *****

C:\Users\Edu\Downloads>
```

Nombre	Fecha de modificación	Tipo	Tamaño
 tienda_discos_backup.sql	13/06/2018 12:24	Archivo SQL	26 KB

Logs de MySQL Windows

Para activar los logs de MySQL en Windows hay que editar el archivo de configuración my.ini que se encuentra en la carpeta oculta C:\ProgramData\MySQL\MySQL Server 5.7.



Desde aquí activamos el log general, el log de consultas lentas, el log binario y el log de errores.

```
# General and Slow logging.
log-output=FILE

general-log=1

general_log_file="EDU-PC.log"

slow-query-log=1

slow_query_log_file="EDU-PC-slow.log"

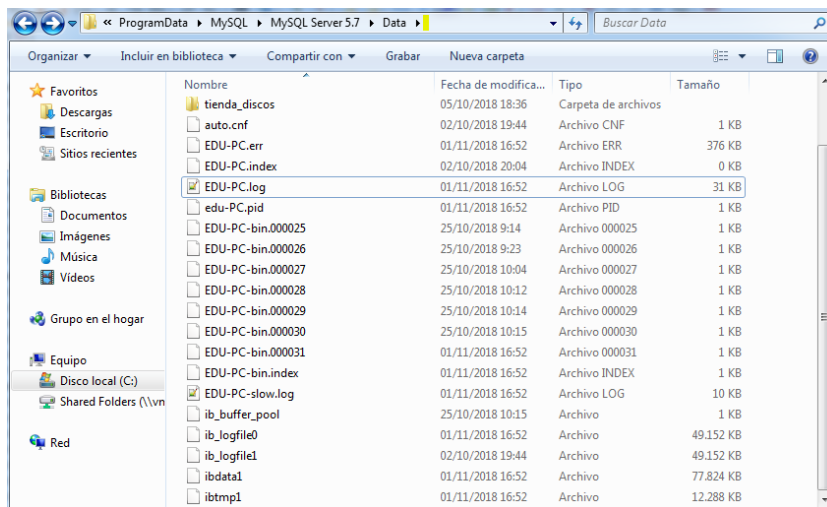
long_query_time=0.1

# Binary Logging.
log-bin="EDU-PC-bin"
binlog-format=STATEMENT
expire_logs_days=10
max_binlog_size=100M

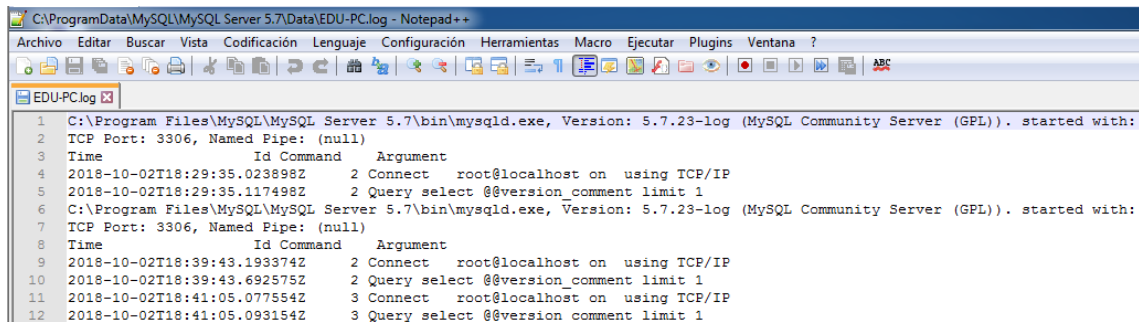
# Error Logging.
log-error="EDU-PC.err"

# Server Id.
server-id=1
```

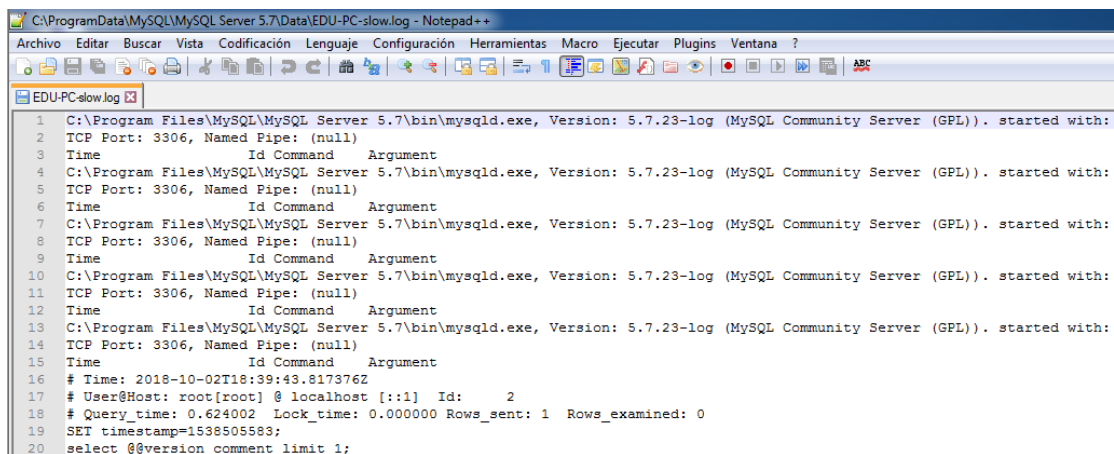
Encontraremos todos los archivos de log en la carpeta Data.



Log general.

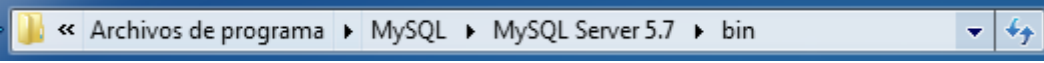
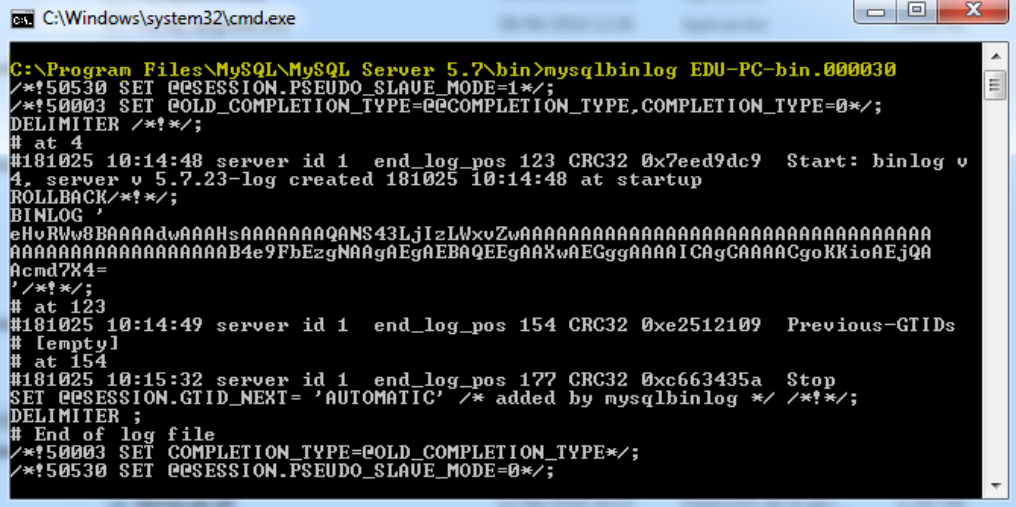


Log de consultas lentas.



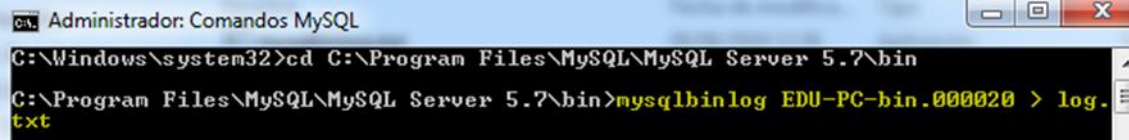
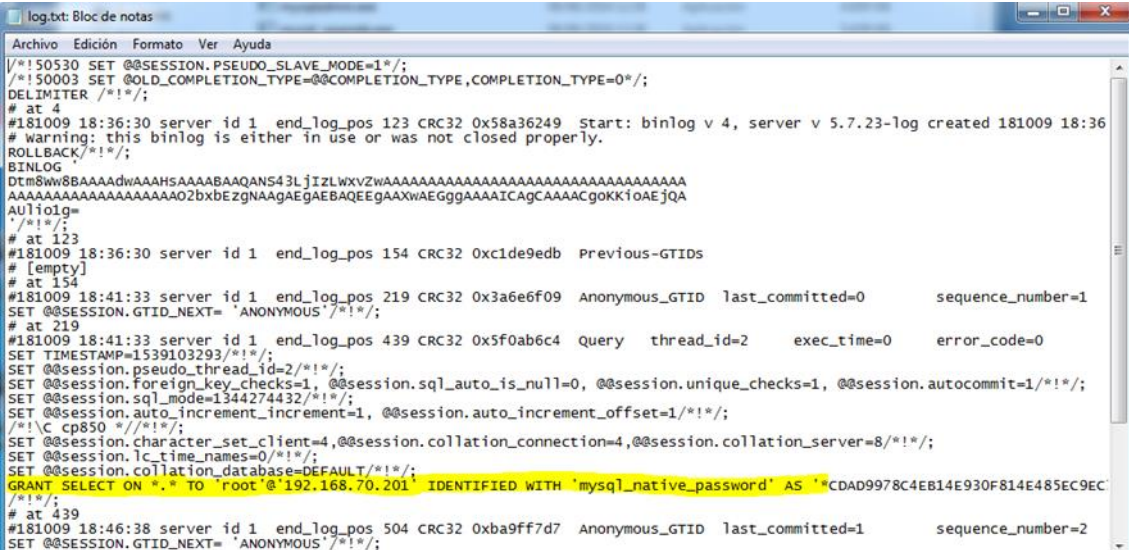
Logs binarios.

Para verlos debo llevarme el archivo de log binario que deseo consultar al directorio bin de MySQL y poner en la consola mysqlbinlog archivo_log_binario.

```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqlbinlog EDU-PC-bin.000030
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @@COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#181025 10:14:48 server id 1 end_log_pos 123 CRC32 0x7eed9dc9 Start: binlog v
4, server v 5.7.23-log created 181025 10:14:48 at startup
ROLLBACK/*!*/;
BINLOG
eHvRWw8BAAAAAwAAHsAAAAAAQANS43LjIzLWxvZWAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAB4e9FbEzgNAAGAEgAEBAQEgEgAA
XwAEgGgAAAAICAgCAAAACGoKKioAEjQA
Acmd7X4=
/*!*/;
# at 123
#181025 10:14:49 server id 1 end_log_pos 154 CRC32 0xe2512109 Previous-GTIDs
# [empty]
# at 154
#181025 10:15:32 server id 1 end_log_pos 177 CRC32 0xc663435a Stop
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=COLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

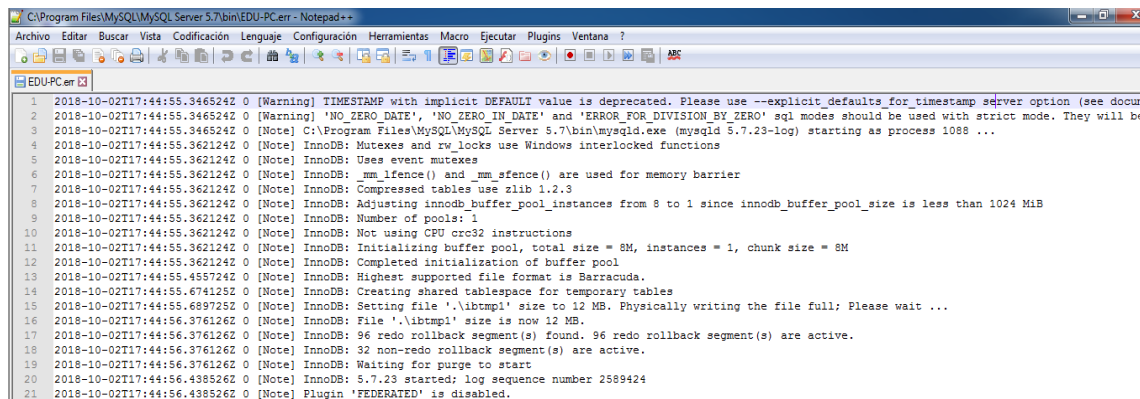
Podemos desviar la salida hacia un fichero TXT para verlo mejor (abrir CMD como administrador).

```
Administrador: Comandos MySQL
C:\Windows\system32>cd C:\Program Files\MySQL\MySQL Server 5.7\bin
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqlbinlog EDU-PC-bin.000020 > log.
txt

log.txt: Bloc de notas
Archivo Edici3n Formato Ver Ayuda
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @@COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#181009 18:36:30 server id 1 end_log_pos 123 CRC32 0x58a36249 Start: binlog v 4, server v 5.7.23-log created 181009 18:36
# warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG
DTm8Ww8BAAAAAwAAHsAAAAABAAQANS43LjIzLWxvZWAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAO2xbEzgNAAGAEgAEBAQEgEgAA
XwAEgGgAAAAICAgCAAAACGoKKioAEjQA
AULioIg=
/*!*/;
# at 123
#181009 18:36:30 server id 1 end_log_pos 154 CRC32 0xc1de9edb Previous-GTIDs
# [empty]
# at 154
#181009 18:41:33 server id 1 end_log_pos 219 CRC32 0x3a6e6f09 Anonymous_GTID last_committed=0 sequence_number=1
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 219
#181009 18:41:33 server id 1 end_log_pos 439 CRC32 0x5f0ab6c4 Query thread_id=2 exec_time=0 error_code=0
SET TIMESTAMP=1539103293/*!*/;
SET @@session.pseudo_thread_id=2/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1, @@session.autocommit=1/*!*/;
SET @@session.sql_mode=1344274432/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*!*/;
/*!*/;
SET @@session.character_set_client=4, @@session.collation_connection=4, @@session.collation_server=8/*!*/;
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
GRANT SELECT ON *.* TO 'root'@'192.168.70.201' IDENTIFIED WITH 'mysql_native_password' AS 'CDAD9978C4EB14E930F814E485EC9EC'
/*!*/;
# at 439
#181009 18:46:38 server id 1 end_log_pos 504 CRC32 0xba9ff7d7 Anonymous_GTID last_committed=1 sequence_number=2
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
```

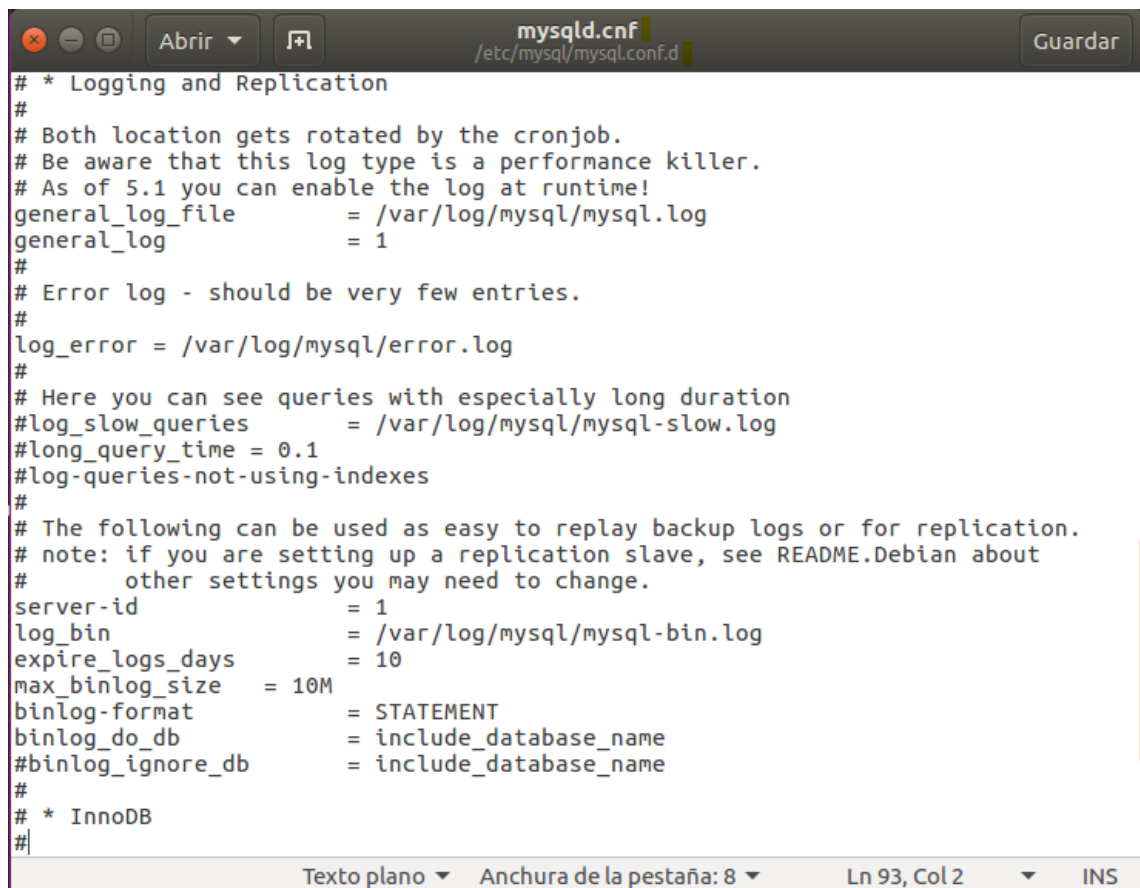
Log de errores.



```
C:\Program Files\MySQL\MySQL Server 5.7\bin\EDU-PC-err - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Herramientas  Macro  Ejecutar  Plugins  Ventana  ?
EDU-PC-err
1 2018-10-02T17:44:55.346524Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see docu
2 2018-10-02T17:44:55.346524Z 0 [Warning] 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and 'ERROR_FOR_DIVISION_BY_ZERO' sql modes should be used with strict mode. They will b
3 2018-10-02T17:44:55.346524Z 0 [Note] C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqld.exe (mysqld 5.7.23-log) starting as process 1088 ...
4 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Mutexes and rw_locks use Windows interlocked functions
5 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Uses event mutexes
6 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: mm_lfence() and mm_sfence() are used for memory barrier
7 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Compressed tables use zlib 1.2.3
8 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Adjusting innodb_buffer_pool_instances from 8 to 1 since innodb_buffer_pool_size is less than 1024 MiB
9 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Number of pools: 1
10 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Not using CPU crc32 instructions
11 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Initializing buffer pool, total size = 8M, instances = 1, chunk size = 8M
12 2018-10-02T17:44:55.362124Z 0 [Note] InnoDB: Completed initialization of buffer pool
13 2018-10-02T17:44:55.455724Z 0 [Note] InnoDB: Highest supported file format is Barracuda.
14 2018-10-02T17:44:55.674125Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
15 2018-10-02T17:44:55.689725Z 0 [Note] InnoDB: Setting file 'ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
16 2018-10-02T17:44:56.376126Z 0 [Note] InnoDB: File 'ibtmp1' size is now 12 MB.
17 2018-10-02T17:44:56.376126Z 0 [Note] InnoDB: 96 redo rollback segment(s) found, 96 redo rollback segment(s) are active.
18 2018-10-02T17:44:56.376126Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
19 2018-10-02T17:44:56.376126Z 0 [Note] InnoDB: Waiting for purge to start
20 2018-10-02T17:44:56.438526Z 0 [Note] InnoDB: 5.7.23 started: log sequence number 2589424
21 2018-10-02T17:44:56.438526Z 0 [Note] Plugin 'FEDERATED' is disabled.
```

Logs de MySQL Linux

En Linux la configuración de los logs se realiza en el fichero `/etc/mysql/mysql.conf.d/mysqld.cnf`



```
mysql.cnf
/etc/mysql/mysql.conf.d
Guardar

# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
general_log_file      = /var/log/mysql/mysql.log
general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries     = /var/log/mysql/mysql-slow.log
#long_query_time = 0.1
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
# other settings you may need to change.
server-id             = 1
log_bin               = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size       = 10M
binlog-format         = STATEMENT
binlog_do_db          = include_database_name
#binlog_ignore_db     = include_database_name
#
# * InnoDB
#
```

Como se puede observar, todos se almacenan en `/var/log/mysql`.

Log general.

```
Abrir  mysql.log  Guardar
/var/log/mysql

/usr/sbin/mysqld, Version: 5.7.24-0ubuntu0.16.04.1-log ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
2018-11-01T16:52:51.206478Z      4 Quit
/usr/sbin/mysqld, Version: 5.7.24-0ubuntu0.16.04.1-log ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
2018-11-01T17:10:22.755370Z      2 Connect  root@localhost on  using Socket
2018-11-01T17:10:22.755459Z      2 Connect  Access denied for user
'root'@'localhost' (using password: NO)
/usr/sbin/mysqld, Version: 5.7.24-0ubuntu0.16.04.1-log ((Ubuntu)). started with:
Tcp port: 3306  Unix socket: /var/run/mysqld/mysqld.sock
Time          Id Command  Argument
2018-11-01T17:10:52.167209Z      2 Connect  root@localhost on  using Socket
2018-11-01T17:10:52.167286Z      2 Connect  Access denied for user
'root'@'localhost' (using password: NO)
```

Log de errores.

```
Abrir  error.log  Guardar
/var/log/mysql

MB. Physically writing the file full; Please wait ...
2018-11-01T17:13:53.606753Z 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
2018-11-01T17:13:53.609111Z 0 [Note] InnoDB: 96 redo rollback segment(s) found.
96 redo rollback segment(s) are active.
2018-11-01T17:13:53.609162Z 0 [Note] InnoDB: 32 non-redo rollback segment(s) are
active.
2018-11-01T17:13:53.609920Z 0 [Note] InnoDB: Waiting for purge to start
2018-11-01T17:13:53.661044Z 0 [Note] InnoDB: 5.7.24 started; log sequence number
6438247
2018-11-01T17:13:53.663780Z 0 [Note] Plugin 'FEDERATED' is disabled.
2018-11-01T17:13:53.667443Z 0 [Note] InnoDB: Loading buffer pool(s) from /var/
lib/mysql/ib_buffer_pool
2018-11-01T17:13:53.681609Z 0 [Note] InnoDB: Buffer pool(s) load completed at
181101 18:13:53
2018-11-01T17:13:53.695769Z 0 [Warning] Failed to set up SSL because of the
following SSL library error: SSL context is not usable without certificate and
private key
```


Logs binarios.

Para verlos hay que navegar con el terminal hacia /var/log/mysql y poner el comando mysqlbinlog --base64-output=DECODE-ROWS como root y a continuación el fichero de log que se desea ver.

```
edu@edu-asgb:~$ cd /var/log/mysql
edu@edu-asgb:/var/log/mysql$ sudo mysqlbinlog --base64-output=DECODE-ROWS mysql-
bin.000044
[sudo] password for edu:
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @@OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#181129 9:27:27 server id 1 end_log_pos 123 CRC32 0x6d8e5122 Start: binlog v
4, server v 5.7.24-0ubuntu0.16.04.1-log created 181129 9:27:27 at startup
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
# at 123
#181129 9:27:27 server id 1 end_log_pos 154 CRC32 0xe7cd494a Previous-GTIDs
# [empty]
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

Podemos desviar la salida y verlo en un fichero de texto más comodamente.

```
edu@edu-asgb: /var/log/mysql
edu@edu-asgb:/var/log/mysql$ sudo mysqlbinlog --base64-output=DECODE-ROWS mysql-
bin.000044 > /home/edu/binlog44
edu@edu-asgb:/var/log/mysql$
```

```
Abrir binlog44 Guardar
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @@OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#181129 9:27:27 server id 1 end_log_pos 123 CRC32 0x6d8e5122 Start: binlog v
4, server v 5.7.24-0ubuntu0.16.04.1-log created 181129 9:27:27 at startup
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
# at 123
#181129 9:27:27 server id 1 end_log_pos 154 CRC32 0xe7cd494a Previous-GTIDs
# [empty]
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

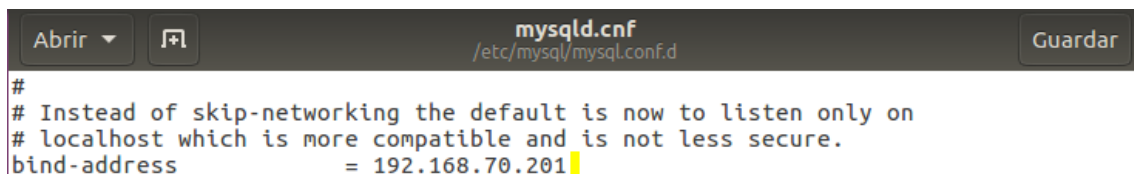
Conexiones remotas

Puedo permitir conexiones a mi servidor MySQL con el comando GRANT, de esta forma asigno permisos a usuarios para que puedan acceder remotamente a mis bases de datos.

En este ejemplo estoy permitiendo que se conecten desde el host 192.168.70.23 (host de Álvaro) como usuario root a mi servidor, la clave para este usuario es "inves", tiene permisos de select en todas las bases de datos y sus tablas, es decir puede hacer consultas.

```
mysql> grant select on *.* to root@192.168.70.23 identified by "inves";
Query OK, 0 rows affected, 1 warning (0.05 sec)
```

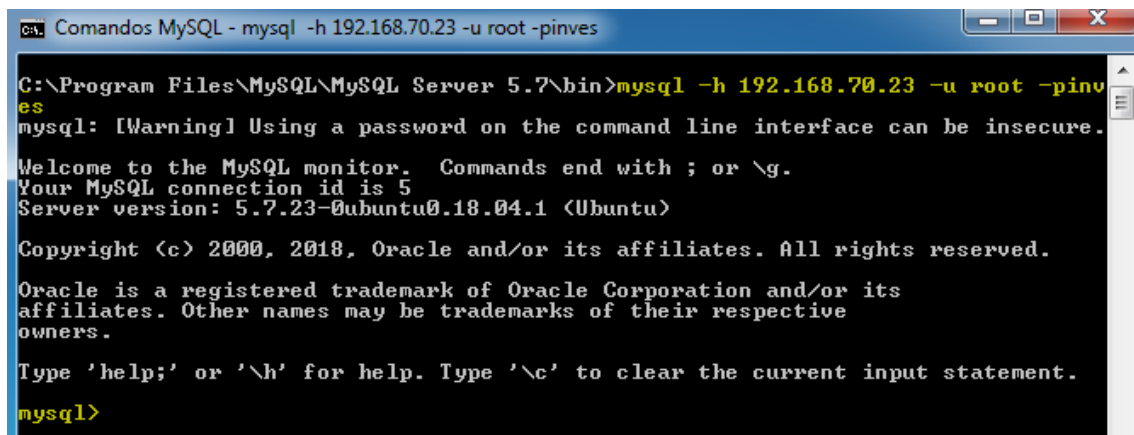
Para hacer esto en Linux es necesario poner la IP del servidor al que permitimos conexiones en el fichero de configuración mysql.cnf, en Windows no es necesario.



Para conectarse a un servidor remoto se puede hacer con la siguiente orden.

```
mysql -h IP_host_remoto -u usuario -pcontraseña
```

En este caso me conecto al host de mi compañero Álvaro.



Comando mysqlcheck

El comando mysqlcheck comprueba y repara tablas, también puede optimizar y analizar las tablas de nuestras bases de datos.

Sintaxis → `mysqlcheck -u usuario -p base_de_datos <opciones>`

Las opciones más utilizadas son: -a (analizar), -r (reparar) y -o (optimizar). Podemos utilizar `--all-databases` para usar el comando sobre todas la bases de datos del servidor.

Ejemplos:

```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqlcheck -u root -p tienda_discos
-a
Enter password: *****
tienda_discos.agencias_transporte      OK
tienda_discos.cds                      OK
tienda_discos.clientes                 OK
tienda_discos.contratos                 OK
tienda_discos.direccion_clientes       OK
tienda_discos.discos                   OK
tienda_discos.discos_pedidos            OK
tienda_discos.facturas_clientes        OK
tienda_discos.mp3                      OK
tienda_discos.nominas                  OK
tienda_discos.pedidos                   OK
tienda_discos.sellos_discograficos     OK
tienda_discos.trabajadores              OK
tienda_discos.vinilos                  OK
```

```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqlcheck -u root -p --all-database
s -o
Enter password: *****
mysql.columns_priv                     Table is already up to date
mysql.db                               OK
mysql.engine_cost
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
mysql.event                            Table is already up to date
mysql.func                             Table is already up to date
mysql.general_log
note      : The storage engine for the table doesn't support optimize
mysql.gtid_executed
Warning   : Please do not modify the gtid_executed table. This is a mysql interna
l system table to store GTIDs for committed transactions. Modifying it can lead
to an inconsistent GTID state.
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
mysql.help_category
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
mysql.help_keyword
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
mysql.help_relation
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
mysql.help_topic
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
```

```
C:\Program Files\MySQL\MySQL Server 5.7\bin>mysqlcheck -u root -p --all-database
s -r
Enter password: *****
mysql.columns_priv                     OK
mysql.db                               OK
mysql.engine_cost
note      : The storage engine for the table doesn't support repair
mysql.event                            OK
mysql.func                             OK
mysql.general_log                      OK
mysql.gtid_executed
```


Usuarios y permisos

- Creación de un usuario directamente en la tabla user de la base de datos mysql.

```
insert into mysql.user (Host, User, Password) values ("localhost", "edu", password("inves"));
```

flush privileges; ← Recargar la tabla para que tenga efecto.

- Crear usuarios con CREATE USER.

```
CREATE USER [IF NOT EXISTS] user [auth_option] [, user [auth_option]] DEFAULT ROLE  
role [, role] [REQUIRE {NONE | tls_option [(AND) tls_option] ...}] [WITH resource_option  
[resource_option] ...] [password_option | lock_option]
```

```
mysql> create user eduardo@localhost identified by "inves";  
Query OK, 0 rows affected (0,01 sec)
```

```
edu@edu-asgb:~$ mysql -u eduardo -pinves  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 5.7.24-0ubuntu0.16.04.1-log (Ubuntu)
```

Cambiar contraseña a usuario que está logueado, es decir al que tiene la sesión activa.

```
mysql> set password=password("inves123");  
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

Cambiar contraseña a otros usuarios con el usuario root.

```
mysql> SET PASSWORD FOR 'eduardo'@'localhost' = PASSWORD('miclave');  
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

```
edu@edu-asgb:~$ mysql -u eduardo -pmiclave  
mysql: [Warning] Using a password on the command line interface can be insecure.  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 20  
Server version: 5.7.24-0ubuntu0.16.04.1-log (Ubuntu)
```

Comando GRANT

Se utiliza para dar permisos a usuarios y si el usuario no existe lo crea.

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ON \ TO user [IDENTIFIED  
BY [PASSWORD] 'password'] [, user [IDENTIFIED BY [PASSWORD] 'password']]  
[REQUIRE NONE | [\] [CIPHER 'cipher' [AND]] [ISSUER 'issuer' [AND]] [SUBJECT 'subject']]  
[WITH [GRANT OPTION | MAX_QUERIES_PER_HOUR count |  
MAX_UPDATES_PER_HOUR count | MAX_CONNECTIONS_PER_HOUR count |  
MAX_USER_CONNECTIONS count]]
```

Creamos al usuario samuel@localhost pero sin privilegios.

```
mysql> GRANT USAGE ON *.* TO samuel@localhost IDENTIFIED BY 'inves';
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

localhost	samuel	N	N	N
-----------	--------	---	---	---

```
edu@edu-asgb:~$ mysql -u samuel -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 25
Server version: 5.7.24-0ubuntu0.16.04.1-log (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use tienda_discos;
ERROR 1044 (42000): Access denied for user 'samuel'@'localhost' to database 'tie
nda_discos'
```

- Permisos a nivel global, cuidado al asignarlos porque prevalecen sobre todos los demás.

Todos los permisos sobre todas las bases de datos al usuario genaro@localhost con contraseña inves.

```
mysql> GRANT ALL ON *.* TO genaro@localhost identified by "inves";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Permisos de select sobre todas las bases de datos al usuario maria@localhost con contraseña inves.

```
mysql> GRANT select ON *.* TO maria@localhost identified by "inves";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Permisos de alter sobre todas las bases de datos al usuario alfredo@localhost con contraseña inves.

```
mysql> GRANT alter ON *.* TO alfredo@localhost identified by "inves";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

- Permisos a nivel de base de datos, prevalecen sobre los de tabla.

Permisos de insert y update al usuario oscar@localhost con contraseña clave en la base de datos tienda_discos.

```
mysql> GRANT insert, update ON tienda_discos.* TO oscar@localhost identified by
"clave";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Permisos de select, drop y update al usuario ricardo@172.16.45.10 con contraseña clave en la base de datos ebanca.

```
mysql> GRANT select, drop, update on ebanca.* to ricardo@172.16.45.10 identified by "clave";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Permisos de insert y alter a usuario sergio@localhost con contraseña clave1 en la base de datos mysql.

```
mysql> GRANT insert, alter ON mysql.* TO sergio@localhost identified by "clave1";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

- Permisos a nivel de tabla, prevalecen sobre los de columna.

Permiso de delete en la tabla clientes de la base de datos tienda_discos.

```
mysql> GRANT delete ON tienda_discos.clientes TO ricardo@172.16.45.10;
Query OK, 0 rows affected (0,00 sec)
```

Permiso insert en la tabla mp3 de la base de datos tienda_discos.

```
mysql> GRANT insert ON tienda_discos.mp3 TO oscar@localhost;
Query OK, 0 rows affected (0,00 sec)
```

Permiso de update en la tabla usuarios de la base de datos ebanca.

```
mysql> GRANT update ON ebanca.usuarios TO alvaro@localhost identified by "5";
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Otra forma de dar permisos sobre tablas.

```
mysql> use tienda_discos;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> grant update on table nominas to azahara@172.16.45.89 identified by "inves";
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

- Permisos a nivel de columna, son los más específicos.

Permiso de update en la columna cantidad de la tabla imes de la base de datos ebanca.

```
mysql> GRANT update(cantidad) ON ebanca.imes TO alvaro@localhost;
Query OK, 0 rows affected (0,01 sec)
```

Permiso de select en la columna categoría y update en la columna id de la tabla contratos de la base de datos tienda_discos.

```
mysql> GRANT select(categoria), update(id) ON tienda_discos.contratos TO alvaro@localhost;
Query OK, 0 rows affected (0,01 sec)
```

Permiso de update en la columna artista de la tabla discos de la base de datos tienda_discos al usuario alvaro@172.16.45.21 con clave inves.

```
mysql> GRANT update(artista) ON tienda_discos.discos TO alvaro@172.16.45.21 identified by "inves";
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

Permisos de select, insert y update sobre la columna fecha_fin de la tabla contratos de la base de datos tienda_discos al usuario marisa@172.16.45.28 con clave 34.

```
mysql> GRANT select(fecha_fin), insert(fecha_fin), update(fecha_fin) ON tienda_discos.contratos TO marisa@172.16.45.28 identified by "34";
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

- grant option

Asigna permisos a ese usuario para que pueda dar permisos a otros usuarios.

```
mysql> grant all on ebanca.* to aroa@172.16.45.84 identified by "123" with grant option;
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

- usage

No da más permisos, solo los que ya posee ese usuario, pero a su vez podemos dar el permiso grant option con él.

```
mysql> grant usage on *.* to jose@localhost identified by "321" with grant option;
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

- Asignar permisos a varios usuarios.

Se puede asignar permisos a varios usuarios separándolos por comas, si lo que queremos es crearlos a la vez, hay que poner la contraseña a cada uno.

```
mysql> grant select on mysql.user to alberto@localhost identified by "inves", alberto2@localhost identified by "inves";
Query OK, 0 rows affected, 2 warnings (0,00 sec)
```

Comando REVOKE

Se utiliza para quitar permisos a los usuarios, hay que estar atento porque si por ejemplo quitamos un permiso sobre una tabla pero ese usuario tiene permisos globales sobre la misma no estaríamos haciendo nada porque prevalecen los permisos globales.

```
REVOKE priv_type [(column_list)] [, priv_type [(column_list)]] ON [object_type] {tbl_name | * | *.* | db_name.*} FROM user [, user]
```

Quito permiso global de select al usuario genaro@localhost.

```
mysql> REVOKE select ON *.* FROM genaro@localhost;
Query OK, 0 rows affected (0,00 sec)
```

Quito permisos select, drop, update en la base de datos ebanca a ricardo@172.16.45.10.

```
mysql> REVOKE select, drop, update ON ebanca.* FROM ricardo@172.16.45.10;
Query OK, 0 rows affected (0,01 sec)
```

Quito permiso delete en la tabla clientes de la base de datos tienda_discos a ricardo@172.16.45.10.

```
mysql> REVOKE delete ON tienda_discos.clientes FROM ricardo@172.16.45.10;
Query OK, 0 rows affected (0,00 sec)
```

Quito permisos de select en la columna categoría, update en la columna id de la tabla contratos de la base de datos tienda_discos al usuario alvaro@localhost.

```
mysql> REVOKE select(categoria), update(id) ON tienda_discos.contratos FROM alvaro@localhost;
Query OK, 0 rows affected (0,00 sec)
```

Quito el permiso grant option en la base de datos ebanca a aroa@172.16.45.84.

```
mysql> REVOKE grant option ON ebanca.* FROM aroa@172.16.45.84;
Query OK, 0 rows affected (0,00 sec)
```

Gestión de recursos

Podemos gestionar los recursos de nuestro servidor de base de datos con la opción with del comando GRANT usage (no tocamos los permisos).

GRANT USAGE ON *base_de_datos.tabla* TO *usuario@host* with *opciones*

- MAX_QUERIES_PER_HOUR número (Consultas por hora).
- MAX_UPDATES_PER_HOUR número (Updates por hora).
- MAX_CONNECTIONS_PER_HOUR número (Conexiones a la base de datos por hora).
- MAX_USER_CONNECTIONS número (Conexiones simultaneas a la base de datos).

1 conexión simultánea, 5 consultas por hora y 5 conexiones a la hora al usuario ricardo@172.16.45.10.

```
mysql> grant usage on *.* to ricardo@172.16.45.10 with MAX_USER_CONNECTIONS 1 MAX_QUERIES_PER_HOUR 5 MAX_CONNECTIONS_PER_HOUR 5;
Query OK, 0 rows affected, 1 warning (0,01 sec)
```

Poniéndolas a 0 eliminamos la restricción, es decir equivale a infinito.

```
mysql> grant usage on *.* to ricardo@172.16.45.10 with MAX_USER_CONNECTIONS 0 MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0;
Query OK, 0 rows affected, 1 warning (0,00 sec)
```

Borrar usuarios

Borramos usuarios con el comando DROP USER.

DROP USER [IF EXISTS] user [, user]

```
mysql> drop user samuel@localhost;
Query OK, 0 rows affected (0,01 sec)

mysql> drop user ricardo@172.16.45.10;
Query OK, 0 rows affected (0,03 sec)

mysql> drop user marisa@172.16.45.28;
Query OK, 0 rows affected (0,01 sec)
```

Funciones

Ver la cantidad de discos en total que tengo para vender en la tienda (función vacía, sin parámetros de entrada) válida para un inventario de la tienda

```
drop function if exists stock_total;
delimiter //
create function stock_total() returns int
begin
declare resultado int;
select sum(stock) from discos into resultado;
return resultado;
end //
delimiter ;

select stock_total() as "Total de discos";
```

```
MariaDB [tienda_discos]> select stock_total() as "Total de discos";
+-----+
| Total de discos |
+-----+
|           903 |
+-----+
1 row in set (0.00 sec)
```

Ver cuánto dinero cobra un empleado al año en total (pasando como parámetro el DNI del empleado)

```
drop function if exists salario_anual;
delimiter //
create function salario_anual(dni_empleado varchar(9)) returns
decimal(9,2)
begin
declare resultado decimal(9,2);
select salario * 14 from nominas n join contratos c on
n.id_contrato=c.id join trabajadores t on
c.num_ss_trabajador=t.numero_ss where dni=dni_empleado into resultado;
return resultado;
end //
delimiter ;

select salario_anual('7841145J');
```

```
MariaDB [tienda_discos]> select salario_anual('74445882R');
+-----+
| salario_anual('74445882R') |
+-----+
|           13513.50 |
+-----+
1 row in set (0.00 sec)
```

Función que me devuelva 0 si el stock de los discos es superior a 10 o 1 si es inferior, es decir, comprobar si hay que realizar un pedido de ese disco a la discográfica porque nos quedan pocos para vender

```
drop function if exists stock_bajo;
delimiter //
create function stock_bajo(ref_disco int(10)) returns boolean
begin
declare stock_disco int;
declare resultado boolean default false;
select stock from discos where num_referencia=ref_disco into
stock_disco;
if stock_disco <= 10 then
set resultado=true;
end if;
return resultado;
end //
delimiter ;

select artista, titulo, stock_bajo(num_referencia) as "stock bajo"
from discos;
```

```
MariaDB [tienda_discos]> select artista, titulo, stock_bajo(num_referencia) as "stock bajo" from discos;
```

artista	titulo	stock bajo
Nick Curly	Dont Takers	0
Jay Lumen	Inside	1
Oscar Mulero	Frio y Calor	0
Nicole Moudaber	Spring	0
Eli Brown	Sumatra	0
Carl Cox	See You	1
Paco Osuna	Mindblowing	0
Sam Michaels	Venore	0
Mario Ochoa	Aldebaran	0
Weska	Voyage	1
Stephan Bozin	Strand	0
Amelie Lens	Awakenings	0

```
12 rows in set (0.00 sec)
```

Función que me diga la cantidad de pedidos que hemos tenido en un mes concreto de un año pasados como parámetro

```
drop function if exists pedidos_mes;
delimiter //
create function pedidos_mes(mes int(2), anio int(4)) returns int
begin
declare resultado int;
select count(*) from pedidos where year(fecha_realizacion)=anio and
month(fecha_realizacion)=mes into resultado;
return resultado;
end //
delimiter ;

select pedidos_mes(3, 2018) as "Pedidos";
```

```
MariaDB [tienda_discos]> select pedidos_mes(3, 2018) as "Pedidos";
```

Pedidos
2

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


Procedimientos

Ver cuánto tiempo de contrato le queda a un empleado poniendo como parámetro su DNI

```
drop procedure if exists fin_contrato;
delimiter //
create procedure fin_contrato(dni_empleado varchar(9))
begin
declare anios, meses, dias, resto int;
select datediff(fecha_fin, now()) from contratos join trabajadores on
num_ss_trabajador=numero_ss where dni=dni_empleado into dias;
set anios=dias div 365;
set resto=dias % 365;
set meses=resto div 30;
set dias=resto % 30;
select concat(anios, " años ", meses, " meses y ", dias, " días de
contrato") as "Le quedan";
end //
delimiter ;

call fin_contrato("74445882R");
```

```
MariaDB [tienda_discos]> call fin_contrato("74445882R");
+-----+
| Le quedan |
+-----+
| 5 años 6 meses y 7 días de contrato |
+-----+
1 row in set (0.00 sec)
```

Introduciendo un número de pedido, ver el nombre, apellidos y dirección del cliente que lo ha realizado

```
drop procedure if exists direccion;
delimiter //
create procedure direccion(num_pedido int)
begin
if num_pedido <=0 then
select "Número de pedido incorrecto";
else
select nombre, apellidos, provincia, ciudad, codigo_postal, calle,
numero, piso from direccion_clientes d join clientes c on
d.dni_cliente=c.dni join pedidos p on c.dni=p.dni_cliente where
p.id=num_pedido;
end if;
end //
delimiter ;

call direccion(1);
```

```
MariaDB [tienda_discos]> call direccion(1);
+-----+-----+-----+-----+-----+-----+-----+-----+
| nombre | apellidos | provincia | ciudad | codigo_postal | calle | numero | piso |
+-----+-----+-----+-----+-----+-----+-----+-----+
| Eladio | Cortes Perez | Albacete | Villarobledo | 14520 | Capitan Casado | 54 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.08 sec)

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


Ver el DNI del cliente y el estado de los pedidos de un rango que le pasemos como parámetro, es decir si le pasamos 1, 3 vemos los pedidos con ID del 1 al 3

```
drop procedure if exists estado_pedidos;
delimiter //
create procedure estado_pedidos(primerio int, ultimo int)
begin
if primerio >0 and ultimo>=primerio then
repeat
select id as "Pedido", dni_cliente, estado from pedidos where
id=primerio;
set primerio=primerio+1;
until primerio>ultimo
end repeat;
else
select "Los parámetros no son correctos";
end if;
end //
delimiter ;

call estado_pedidos(1, 3);
```

```
MariaDB [tienda_discos]> call estado_pedidos(1, 3);
+-----+-----+-----+
| Pedido | dni_cliente | estado |
+-----+-----+-----+
|      1 | 78563256W  | Entregado |
+-----+-----+-----+
1 row in set (0.00 sec)

+-----+-----+-----+
| Pedido | dni_cliente | estado |
+-----+-----+-----+
|      2 | 65894125P  | Entregado |
+-----+-----+-----+
1 row in set (0.02 sec)

+-----+-----+-----+
| Pedido | dni_cliente | estado |
+-----+-----+-----+
|      3 | 71698331S  | Enviado |
+-----+-----+-----+
1 row in set (0.04 sec)

Query OK, 0 rows affected (0.07 sec)
```

Pasando como argumento el DNI de un trabajador, que nos diga cuánto dinero ha ganado en total desde que se le contrató

```
drop procedure if exists ganado_total;
delimiter //
create procedure ganado_total(t_dni varchar(9))
begin
declare n_ss decimal(12,0);
declare t_id, meses_trabajados int;
declare total_ganado double;
select numero_ss from trabajadores where dni=t_dni into n_ss;
select id, TIMESTAMPDIFF(MONTH, fecha_inicio, now()) from contratos
where num_ss_trabajador=n_ss into t_id, meses_trabajados;
select round(salario * meses_trabajados,2) from nominas where id=t_id
into total_ganado;
select total_ganado as "Total Euros ganados";
end //
delimiter ;

call ganado_total("75889364F");
```

```
MariaDB [tienda_discos]> call ganado_total("68955412Z");
+-----+
| Total Euros ganados |
+-----+
|          47738.25 |
+-----+
1 row in set (0.00 sec)
```

Ampliar un año los contratos de los trabajadores que vencen este mismo año (Cursor)

```
drop procedure ampliar_contratos;
DELIMITER //
CREATE PROCEDURE ampliar_contratos()
BEGIN
DECLARE fin boolean default false;
DECLARE id_contrato int;
DECLARE micursor CURSOR FOR select id from contratos where
year(fecha_fin)=year(now());
DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin=true;
OPEN micursor;
WHILE not fin DO
FETCH micursor into id_contrato;
if not fin then
update contratos set fecha_fin = date_add(fecha_fin, interval 1 year)
where id=id_contrato;
end if;
END WHILE;
CLOSE micursor;
END; //
delimiter ;
```

Antes de aplicar el procedimiento:

```
select * from contratos;
```

```
MariaDB [tienda_discos]> select * from contratos;
```

id	fecha_inicio	fecha_fin	categoria	num_ss_trabajador
1	2015-05-01	2022-05-01	Programador	865325698541
2	2017-04-05	2019-04-05	Administrativo	754625986532
3	2014-09-08	2024-09-08	Programador	898745145632
4	2015-01-15	2019-01-15	Administrativo	748489563254
5	2016-08-24	2024-08-24	Programador	125458976326
6	2018-01-01	2019-01-01	Administrativo	896573652145
7	2015-12-05	2023-12-05	Logistica	877963325411
8	2017-05-21	2019-05-21	Logistica	956832564785
9	2016-02-05	2023-02-05	Logistica	596832148956
10	2014-04-01	2028-04-01	Jefe	757575757575

```
10 rows in set (0.00 sec)
```

Después de aplicarlo:

```
call ampliar_contratos();
```

```
select * from contratos;
```

```
MariaDB [tienda_discos]> call ampliar_contratos();
Query OK, 0 rows affected (0.22 sec)
```

```
MariaDB [tienda_discos]> select * from contratos;
```

id	fecha_inicio	fecha_fin	categoria	num_ss_trabajador
1	2015-05-01	2022-05-01	Programador	865325698541
2	2017-04-05	2020-04-05	Administrativo	754625986532
3	2014-09-08	2024-09-08	Programador	898745145632
4	2015-01-15	2020-01-15	Administrativo	748489563254
5	2016-08-24	2024-08-24	Programador	125458976326
6	2018-01-01	2020-01-01	Administrativo	896573652145
7	2015-12-05	2023-12-05	Logistica	877963325411
8	2017-05-21	2020-05-21	Logistica	956832564785
9	2016-02-05	2023-02-05	Logistica	596832148956
10	2014-04-01	2028-04-01	Jefe	757575757575

```
10 rows in set (0.00 sec)
```

Calcular el total de facturación de un año que pasamos como argumento (Cursor)

```
drop procedure if exists facturacion_anio;
delimiter //
create procedure facturacion_anio(anio int)
begin
DECLARE fin boolean default false;
DECLARE id_ped int;
DECLARE resultado double default 0;
DECLARE prov_resultado double;
DECLARE micursor CURSOR FOR select id_pedido from facturas_clientes
where year(fecha_emision)=anio;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin=true;
OPEN micursor;
WHILE not fin DO
FETCH micursor into id_ped;
if not fin then
select round(sum(precio_unidad*1.10),2) from discos join
discos_pedidos on num_referencia=num_ref_disco where id_pedido=id_ped
into prov_resultado;
set resultado=resultado+prov_resultado;
end if;
end while;
close micursor;
select resultado as "Facturación";
end //
delimiter ;

call facturacion_anio(2018);
```

```
MariaDB [tienda_discos]> call facturacion_anio(2018);
+-----+
| Facturación |
+-----+
|      227.33 |
+-----+
1 row in set (0.00 sec)

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

Vistas

Preparadores de los pedidos (vista no actualizable)

```
create ALGORITHM=TEMPTABLE view preparadores as select id as "Num  
Pedido", nombre as "Nombre Preparador", apellidos from pedidos join  
trabajadores on num_ss_trabajador_prepara=numero_ss order by id;  
  
select * from preparadores;
```

```
MariaDB [tienda_discos]> select * from preparadores;
```

Num Pedido	Nombre Preparador	apellidos
1	Jesus	Malpica Fernandez
2	Esperanza	Domínguez Bravo
3	Fernando	Barrajon Huertas
4	Jesus	Malpica Fernandez
5	Fernando	Barrajon Huertas
6	Jesus	Malpica Fernandez
7	Esperanza	Domínguez Bravo
8	Fernando	Barrajon Huertas
9	Jesus	Malpica Fernandez
10	Fernando	Barrajon Huertas

```
10 rows in set (0.00 sec)
```

Precio total de todos los pedidos con IVA (10%) con permisos para el que la ejecuta la vista

```
create SQL SECURITY INVOKER view pedidos_iva as select id_pedido as  
"Num Pedido", round(sum(cantidad*(select precio_unidad * 1.10 from  
discos where num_referencia=num_ref_disco)),2) as "Importe" from  
discos_pedidos group by id_pedido;  
  
select * from pedidos_iva
```

```
MariaDB [tienda_discos]> select * from pedidos_iva;
```

Num Pedido	Importe
1	56.46
2	59.14
3	50.68
4	59.14
5	50.68
6	56.67
7	100.85
8	170.02
9	100.85
10	56.67

```
10 rows in set (0.00 sec)
```

Triggers

Cuando un disco se quede sin stock guardaré su número de referencia, artista y título en una tabla llamada sin_stock

```
drop trigger if exists no_stock;
delimiter //
CREATE TRIGGER no_stock after UPDATE ON discos
FOR EACH ROW begin
if new.stock=0 then
insert into sin_stock values (new.num_referencia, new.artista,
new.titulo);
end if;
end;//
delimiter ;
```

```
MariaDB [tienda_discos]> update discos set stock=0 where num_referencia=22869;
Query OK, 1 row affected (0.16 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [tienda_discos]> select * from sin_stock;
+-----+-----+-----+
| num_referencia | artista  | titulo |
+-----+-----+-----+
|          22869 | Jay Lumen | Inside |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Cuando volvamos a añadir stock a un disco que estaba a 0 lo borramos de la tabla sin_stock

```
drop trigger if exists si_stock;
delimiter //
CREATE TRIGGER si_stock before UPDATE ON discos
FOR EACH ROW begin
if old.stock=0 and new.stock > 0 then
delete from sin_stock where num_referencia=old.num_referencia;
end if;
end;//
delimiter ;
```

```
MariaDB [tienda_discos]> update discos set stock=57 where num_referencia=22869;
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [tienda_discos]> select * from sin_stock;
Empty set (0.00 sec)
```

Cuando añadamos a la base de datos un disco del sello Drumcode que cueste más de 8 € lo venderemos con un 10% de descuento

```
drop trigger if exists descuento_drumcode;
delimiter //
CREATE TRIGGER descuento_drumcode before INSERT ON discos
FOR EACH ROW begin
declare cif_sello varchar(9);
select cif from sellos_discograficos where nombre="Drumcode" into
cif_sello;
if new.cif_sello_discografico=cif_sello and new.precio_unidad > 8 then
set new.precio_unidad=new.precio_unidad-(new.precio_unidad*0.10);
end if;
end;//
delimiter ;
```

```
MariaDB [tienda_discos]> insert into discos values (55479, '2019-03-04', 'Charlotte the Witte', 'Stars', 8.70, 75, 233147852, null);
Query OK, 1 row affected (0.09 sec)

MariaDB [tienda_discos]> select * from discos where num_referencia=55479;
+-----+-----+-----+-----+-----+-----+-----+-----+
| num_referencia | fecha_lanzamiento | artista           | titulo | precio_unidad | stock | cif_sello_discografico | imagen |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 55479 | 2019-03-04 | Charlotte the Witte | Stars | 7.83 | 75 | 233147852 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Bonificar con un 1% más de salario a los trabajadores con antigüedad superior o igual a 3 años

```
drop trigger if exists bonificacion_salario;
delimiter //
CREATE TRIGGER bonificacion_salario before INSERT ON nominas
FOR EACH ROW begin
declare antiguedad int;
select TIMESTAMPDIFF(YEAR, fecha_inicio, now()) from contratos where
id=new.id_contrato into antiguedad;
if antiguedad >= 3 then
set new.salario=new.salario+(new.salario*0.01);
end if;
end;//
delimiter ;
```

```
MariaDB [tienda_discos]> select * from contratos where id=1;
+-----+-----+-----+-----+-----+-----+
| id | fecha_inicio | fecha_fin | categoria | num_ss_trabajador |
+-----+-----+-----+-----+-----+-----+
| 1 | 2015-05-01 | 2022-05-01 | Programador | 865325698541 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [tienda_discos]> insert into nominas values(null, now(), 975.80, 1);
Query OK, 1 row affected, 1 warning (0.08 sec)

MariaDB [tienda_discos]> select * from nominas where fecha_emision='2019-03-03';
+-----+-----+-----+-----+
| id | fecha_emision | salario | id_contrato |
+-----+-----+-----+-----+
| 12 | 2019-03-03 | 985.56 | 1 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Eventos

Primero activo el programador de eventos para poder utilizarlos

```
MariaDB [tienda_discos]> set GLOBAL event_scheduler=ON;  
Query OK, 0 rows affected (0.00 sec)
```

Los discos con fecha de lanzamiento hace 1 año o más bajarán su precio un 5% cada 3 meses

```
drop event if exists rebaja;  
delimiter //  
CREATE EVENT rebaja  
ON SCHEDULE EVERY 1 QUARTER  
STARTS CURRENT_TIMESTAMP ENABLE  
DO  
BEGIN  
update discos set precio_unidad=precio_unidad-(precio_unidad*0.05)  
where TIMESTAMPDIFF(YEAR, fecha_lanzamiento, now()) >= 1;  
end; //  
delimiter ;
```

El disco con número de referencia 56789 (Paco Osuna – Mindblowing) aumentará su precio 5 € cada año que pase ya que es una reliquia

```
drop event if exists aumento_56789;  
delimiter //  
CREATE EVENT aumento_56789  
ON SCHEDULE EVERY 1 YEAR  
STARTS "2019-04-01 00:00:00" ENABLE  
DO  
BEGIN  
update discos set precio_unidad=precio_unidad+5 where  
num_referencia=56789;  
end; //  
delimiter ;
```