Nuestra Chuleta de SQL

| Tipo | Qué quiero hacer | Instrucción |
| --- | --- | --- |
| **DDL** | Crear una base de datos | CREATE DATABASE ferreteria; |
|  | Usar una BD | USE ferreteria; |
|  | Crear una tabla de productos | CREATE TABLE productos (  codigo varchar(3),  nombre varchar(30),  precio decimal(6,2),  fechaalta date,  PRIMARY KEY (codigo) ); |
|  | Crea una tabla proveedores | CREATE TABLE proveedor ( codigo VARCHAR(5) NOT NULL , direccion VARCHAR(50) NOT NULL , ciudad VARCHAR(50) NOT NULL , provincia VARCHAR(50) NOT NULL ) ENGINE = InnoDB; |
|  | Agregar un campo clave a una tabla | [ALTER TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) proveedor ADD PRIMARY KEY(codigo); |
|  | Crear un índice en una tabla | [ALTER TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/alter-table.html) categoria ADD INDEX(nombre); |
|  | Eliminar un índice en una tabla | ALTER TABLE categoria DROP INDEX nombre; |
|  | Ejemplo de relación por sql | ALTER TABLE publica ADD FOREIGN KEY(id\_sucursal) REFERENCES sucursal(id\_sucursal) |
|  | Crear una Tabla con la clave foránea directamente en SQL.  ***Nota: Ten en cuenta que el campo de la tabla a la que se hace referencia debe estar creado previamente*** | CREATE TABLE pieza (  codigo INT AUTO\_INCREMENT PRIMARY KEY,  nombre VARCHAR(100) NOT NULL,  color VARCHAR(25) NOT NULL,  precio DECIMAL(7,2) NOT NULL,  codigo\_categoria INT NOT NULL,  FOREIGN KEY (codigo\_categoria) REFERENCES categoria(codigo)  ); |
|  | Contar el número de usuarios por compañía telefónica | SELECT COUNT(usuario) FROM tblusuarios GROUP BY compañia; |
|  | Mostrar el email de los usuarios que usan gmail | SELECT usuario, email FROM tblusuarios WHERE email like '%gmail%'; |
|  | RENAME TABLE `pinturas\_montana`.`hoja2` TO `pinturas\_montana`.`clientes`; |  |
|  |  |  |
|  |  |  |
| DML | Mostrar toda la tabla | SELECT \* FROM `tabla` |
|  | Todos los registros de la tabla | SELECT \* FROM `tabla` |
|  | El Nombre empieza por A | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` WHERE `Nombre` [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "A%" |
|  | El Nombre contiene una A | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` WHERE `Nombre` [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "%A%" |
|  | El Nombre termina en una A | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` WHERE `Nombre` [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "%A" |
|  | Concatenar nombre y apellidos | SELECT CONCAT(actor.last\_name, ', ',actor.first\_name) AS A FROM actor |
|  | Ordenar alfabéticamente de la A-Z | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` ORDER BY `Nombre` ASC |
|  | Ordenar alfabéticamente de la Z-A | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` ORDER BY `Nombre` DESC |
|  | Ordenar alfabéticamente los Nombre que contienen una A | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM ` tabla ` WHERE `Nombre` [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "%A%" ORDER BY `Nombre` ASC |
|  | Cuenta los registros por compañía | SELECT `compañia`, COUNT(compañia) cantidad FROM `tblusuarios` GROUP BY `compañia` order by cantidad |
|  | CREAR BBDD CON CHARSET ELEGIDO | CREATE DATABASE IF NOT EXISTS agencia CHARACTER SET utf8mb4 COLLATE utf8mb4\_spanish\_ci |
|  | AÑADIR COLUMNA Y, SIMULTÁNEAMENTE, PONERLE CLAVE FORÁNEA CON INTEGRIDAD REFERENCIAL (UPDATE EN ESTE CASO) | ALTER TABLE toma ADD COLUMN id\_sucursal int(11),  ADD CONSTRAINT fk\_sucursal  FOREIGN KEY(id\_sucursal) REFERENCES sucursal(id)  ON UPDATE CASCADE  ; |
|  |  | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) nombre,telefono, marca FROM tblusuarios WHERE marca not [in](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/comparison-operators.html#function_in)('nokia', 'blackberry', 'sony'); |
|  | Contar el número de usuarios por compañía telefónica | SELECT COUNT(usuario) FROM tblusuarios GROUP BY compañia; |
|  | Listar los números de teléfono con saldo menor o igual a 300 | SELECT telefono FROM tblusuarios WHERE saldo <= 300; |
|  | Contar, agrupar y ordenar po un alias | [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) nivel, [COUNT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_count)(usuario) as numero FROM `tblusuarios` GROUP BY nivel ORDER BY numero DESC; |
| **Crear Triggers** | | |
|  | Un trigger es un objeto de la base de datos que está asociado con una tabla y que se activa cuando ocurre un evento sobre la tabla.  Los eventos que pueden ocurrir sobre la tabla son:   1. INSERT: El trigger se activa cuando se inserta una nueva fila sobre la tabla asociada. 2. UPDATE: El trigger se activa cuando se actualiza una fila sobre la tabla asociada. 3. DELETE: El trigger se activa cuando se elimina una fila sobre la tabla asociada | DELIMITER //  CREATE TRIGGER nombreTrigger\_AI AFTER INSERT ON tabla\_original  FOR EACH ROW  BEGIN  INSERT INTO tabla\_copia (campo1, campo2, campo3, insertado)  VALUES (NEW.campo1, NEW.campo2, NEW.campo3, NOW() );  END;//  DELIMITER ;  EJEMPLO BD BIBLIOTECA. CASO 1 INSERTAR:  -- Paso 1: crear la tabla  CREATE TABLE prestamo\_copia(IdPrestamo int, `FechaPrestamo` date, `FechaEntrega` date,`idSocio` int, `idLibro` int, insertado datetime, usuario varchar(30));  -- Paso 2: crear el trigger  CREATE TRIGGER nuevo\_prestamo\_AI AFTER INSERT ON Prestamo FOR EACH ROW INSERT INTO  prestamo\_copia(IdPrestamo,FechaPrestamo,FechaEntrega,idSocio,idLibro,Insertado,usuario)VALUES(new.IdPrestamo,new.FechaPrestamo,new.FechaEntrega,new.idSocio,new.idLibro, now(), CURRENT\_USER);  -- Paso 3: insertar un registro en la tabla prestamo sobre la que actúa el disparador  EJEMPLO BD BIBLIOTECA. CASO 2 ACTUALIZAR:  -- Paso 1: crear la tabla  CREATE TABLE `actualizaciones\_libro` (  `ISBNAnterior` varchar(20) NOT NULL,  `TituloAnterior` varchar(65) NOT NULL,  `NumeroEjemplaresAnterior` tinyint(4) NOT NULL,  `ISBNNuevo` varchar(20) NOT NULL,  `TituloNuevo` varchar(65) NOT NULL,  `NumeroEjemplaresNuevo` tinyint(4) NOT NULL,  `Usuario` varchar(15) NOT NULL,  `FechaModificacion` datetime NOT NULL  ) ENGINE=MyISAM DEFAULT CHARSET=utf8;  -- Paso 2: crear el trigger  CREATE TRIGGER `Actualizar\_Libro\_AU` AFTER UPDATE ON `libro`  FOR EACH ROW INSERT INTO Actualizaciones\_Libro  (  ISBNAnterior,  TituloAnterior,  NumeroEjemplaresAnterior,  ISBNNuevo,  TituloNuevo,  NumeroEjemplaresNuevo,  Usuario,  FechaModificacion  )  VALUES  (  OLD.ISBN,  OLD.Titulo,  OLD.NumeroEjemplares,  NEW.ISBN,  NEW.Titulo,  NEW.NumeroEjemplares,  CURRENT\_USER(),  NOW()  )  -- Paso 3: Modifica un registro el la tabla sobre la que actúa el disparador (libro)  EJEMPLO BD BIBLIOTECA. CASO 3 ELIMINAR:  CREATE TABLE socio\_borrado (idSocio INT(11) NOT NULL primary key auto\_increment, NombreCompleto VARCHAR(60),Direccion VARCHAR(100), Correo VARCHAR(25), Telefono VARCHAR(15), Foto VARCHAR(20), borrado DATE);  CREATE TRIGGER socio\_borrado\_AD  AFTER DELETE ON Socio  FOR EACH ROW  INSERT INTO socio\_borrado(idSocio, NombreCompleto,Direccion, Correo, Telefono, Foto, borrado) VALUES (old.idSocio,old.NombreCompleto,old.Direccion,old.Correo,old.Telefono,  old.Foto,now()); |

TRIGGERS

| INSERT: El trigger se activa cuando se inserta una nueva fila sobre la tabla asociada | EJEMPLO SINTAXIS. CASO 1 INSERTAR:  -- Paso 1: crear la tabla donde se hará la copia  CREATE TABLE tabla\_copia(campo1 int, campo2 ….);  -- Paso 2: crear el trigger  CREATE TRIGGER nombre\_trigger\_AI AFTER INSERT ON tabla\_original  FOR EACH ROW INSERT INTO  tabla\_copia(campo1\_tabla\_copia, campo2\_tabla\_copia,...., campo\_fecha\_inseratdo, campo\_usuarioqueinserta)  VALUES(new.campo1Tabla\_original,new.campo2Tabla\_original, now(), CURRENT\_USER);  -- Paso 3: insertar un registro en la tabla original sobre la que actúa el disparador |
| --- | --- |
| UPDATE: El trigger se activa cuando se actualiza una fila sobre la tabla asociada | EJEMPLO SINTAXIS. CASO 2 ACTUALIZAR:  -- Paso 1: crear la tabla donde se hará la copia  CREATE TABLE tabla\_copia (campo1\_copia, campo2\_copia, ….,usuario, FechaModificacion)  -- Paso 2: crear el trigger  CREATE TRIGGER `Nombre\_Trigger\_AU`  AFTER UPDATE ON tabla\_original  FOR EACH ROW INSERT INTO tabla\_copia  ( campo1\_tabla\_copia,  campo2\_tabla\_copia,  Usuario,  FechaModificacion  )  VALUES  (  OLD.campo1\_tabla\_original,  OLD.campo2\_tabla\_original,...    NEW.campo1\_tabla\_original,  NEW.campo1\_tabla\_original, …  CURRENT\_USER(),  NOW()  )  -- Paso 3: Modifica un registro el la tabla sobre la que actúa el disparador (libro) |
| DELETE: El trigger se activa cuando se elimina una fila sobre la tabla asociada | EJEMPLO CASO 3 ELIMINAR:  CREATE TABLE tabla\_copia (campo1, campo2 …, borrado DATETIME);  CREATE TRIGGER nombre\_trigger\_AD  AFTER DELETE ON tabla\_original  FOR EACH ROW  INSERT INTO tabla\_copia(campo1\_tabla\_copia, campo2\_tabla\_copia, borrado, DATE) VALUES (old.campo1\_tabla\_original,old.campo2\_tabla\_original,...,  now()); |

PASOS PARA TRABAJAR LOS TRIGGERS

CREATE PROCEDURE nombre\_procedimiento()

SELECT \* FROM nombre\_tabla;

DELIMITER //

CREATE PROCEDURE nombre\_procedimiento()

BEGIN

SELECT \* FROM nombre\_tabla;

SELECT \* FROM nombre\_tabla;

END//

DELIMITER ;

CALL nombre\_procedimiento( );

DELIMITER //

CREATE PROCEDURE nombre\_procedimiento(IN X varchar(20))

BEGIN

SELECT \* FROM nombre\_tabla WHERE `Nombre` like X;

END//

DELIMITER ;

CREATE TABLE tabla\_nueva LIKE tabla\_original;

CREATE VIEW nombre as select \* from tabla