

BASE DE DATOS PARA SISTEMA DE GESTION DE CURSOS PARA PROCESOS

```
CREATE USER 'anonimo'@'localhost' IDENTIFIED BY '12345';  
GRANT INSERT, DELETE, SELECT, UPDATE, EXECUTE ON *.* TO  
'anonimo'@'localhost';  
FLUSH PRIVILEGES;
```

```
CREATE TABLE Usuario(nombre_usuario VARCHAR(95), contraseña  
VARCHAR(95), numero_personal VARCHAR(35), PRIMARY KEY(nombre_usuario));
```

```
CREATE TABLE Personal(numero_personal VARCHAR(35), nombre VARCHAR(95),  
rfc VARCHAR(15), correo VARCHAR(95), fecha_nacimiento DATE, PRIMARY  
KEY(numero_personal));
```

```
CREATE TABLE Directivo(numero_personal VARCHAR(35), sindicato  
VARCHAR(95), PRIMARY KEY(numero_personal), FOREIGN KEY(numero_personal)  
REFERENCES Personal(numero_personal));
```

```
CREATE TABLE Encargado(numero_personal VARCHAR(35), area VARCHAR(95),  
PRIMARY KEY(numero_personal), FOREIGN KEY(numero_personal) REFERENCES  
Personal(numero_personal));
```

```
CREATE TABLE Docente(numero_personal VARCHAR(35), anos_experiencia INT,  
perfil_profesional enum('Profesionales', 'Medio superior', 'Superior'),  
numero_encargado VARCHAR(35), PRIMARY KEY(numero_personal), FOREIGN  
KEY(numero_personal) REFERENCES Personal(numero_personal), FOREIGN  
KEY(numero_encargado) REFERENCES Encargado(numero_personal));
```

```
CREATE TABLE Curso(clave_curso VARCHAR(35), nombre VARCHAR(95),  
descripcion TEXT, seccion VARCHAR(95), numero_personal VARCHAR(35),  
numero_encargado VARCHAR(35), horario VARCHAR(200), duracion  
VARCHAR(95), PRIMARY KEY(clave_curso), FOREIGN KEY(numero_personal)  
REFERENCES Docente(numero_personal), FOREIGN KEY(numero_encargado)  
REFERENCES Encargado(numero_personal) );
```

```
CREATE TABLE Plan(id_plan INT AUTO_INCREMENT, fecha_elaboracion DATE,  
estado enum('Activo', 'Inactivo'), avance FLOAT, fecha_actualizacion DATE,  
numero_personal VARCHAR(35), clave_curso VARCHAR(35), PRIMARY  
KEY(id_plan), FOREIGN KEY(numero_personal) REFERENCES  
Docente(numero_personal), FOREIGN KEY(clave_curso) REFERENCES
```

```
Curso(clave_curso) );
```

```
CREATE TABLE Tema(id_tema INT AUTO_INCREMENT, nombre VARCHAR(95),  
estado enum('Visto', 'Pendiente'), clave_curso VARCHAR(35), fecha DATE,  
PRIMARY KEY(id_tema), FOREIGN KEY(clave_curso) REFERENCES  
Curso(clave_curso));
```

```
CREATE TABLE Actividad(actividad VARCHAR(95), id_tema INT, FOREIGN  
KEY(id_tema) REFERENCES Tema(id_tema));
```

```
CREATE TABLE Host(mac_address varchar(75), attempts INT);
```

```
DELIMITER $$
```

```
CREATE PROCEDURE agregarCurso(IN_clave_curso VARCHAR(35), IN_nombre  
VARCHAR(95), IN_descripcion TEXT, IN_seccion VARCHAR(95),  
IN_numero_personal VARCHAR(35), IN_numero_encargado VARCHAR(35),  
IN_horario VARCHAR(200), IN_duracion VARCHAR(95))
```

```
BEGIN
```

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
```

```
BEGIN
```

```
SHOW ERRORS LIMIT 1;
```

```
RESIGNAL;
```

```
ROLLBACK;
```

```
END;
```

```
DECLARE EXIT HANDLER FOR SQLWARNING
```

```
BEGIN
```

```
SHOW WARNINGS LIMIT 1;
```

```
RESIGNAL;
```

```
ROLLBACK;
```

```
END;
```

```
START TRANSACTION;
```

```
IF 'SIN DOCENTE' = IN_numero_personal THEN
```

```
SET @id_personal = NULL;
```

```
ELSE
```

```
SET @id_personal = IN_numero_personal;
```

```
END IF;
```

```
INSERT INTO Curso VALUES(IN_clave_curso, IN_nombre, IN_descripcion,  
IN_seccion, @id_personal , IN_numero_encargado, IN_horario, IN_duracion);
```

```
COMMIT;
```

```
END $$
```

```
DELIMITER ;
```

```

DELIMITER $$
CREATE PROCEDURE sendAddress(dir varchar(64))
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
SHOW ERRORS LIMIT 1;
RESIGNAL;
ROLLBACK;
END;
DECLARE EXIT HANDLER FOR SQLWARNING
BEGIN
SHOW WARNINGS LIMIT 1;
RESIGNAL;
ROLLBACK;
END;
DECLARE CONTINUE HANDLER FOR 1329
BEGIN
set @idMac = NULL;
END;
START TRANSACTION;
SELECT SUM(attempts) FROM Host WHERE mac_address = dir INTO
@numAttempts;
IF @numAttempts > 4 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'Attempts limit reached!';
ELSE
SELECT mac_address FROM Host WHERE mac_address = dir LIMIT 1 INTO
@idMac;
IF @idMac IS NOT NULL THEN
UPDATE Host set attempts = attempts + 1 WHERE mac_address = dir;
ELSE
INSERT INTO Host(mac_address, attempts) VALUES(dir, 1);
END IF;
END IF;
COMMIT;
END $$
DELIMITER ;

```

```

DELIMITER $$

```

```

CREATE PROCEDURE agregarDocente(IN_numero_personal VARCHAR(35),
IN_nombre VARCHAR(95), IN_rfc VARCHAR(15), IN_correo VARCHAR(95),
IN_fecha_nacimiento Date, IN_anos_experiencia INT, IN_perfil_profesional
enum('Profesionales', 'Medio_Superior', 'Superior'),IN_numero_encargado
VARCHAR(35))
BEGIN
DECLARE EXIT HANDLER FOR SQLEXCEPTION
BEGIN
SHOW ERRORS LIMIT 1;
RESIGNAL;
ROLLBACK;
END;
DECLARE EXIT HANDLER FOR SQLWARNING
BEGIN
SHOW WARNINGS LIMIT 1;
RESIGNAL;
ROLLBACK;
END;
START TRANSACTION;
INSERT INTO Personal(numero_personal, nombre, rfc, correo, fecha_nacimiento)
VALUES(IN_numero_personal, IN_nombre, IN_rfc, IN_correo, IN_fecha_nacimiento);
INSERT INTO Usuario(nombre_usuario, contraseña, numero_personal)
VALUES(IN_numero_personal, SHA2('1', 256), IN_numero_personal);
INSERT INTO Docente(numero_personal, anos_experiencia, perfil_profesional,
numero_encargado) VALUES(IN_numero_personal, IN_anos_experiencia,
IN_perfil_profesional, IN_numero_encargado);
COMMIT;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE verificar_existencia_docente(IN_numero_personal
VARCHAR(35))
BEGIN
SELECT numero_personal FROM Personal WHERE numero_personal =
IN_numero_personal;
END $$
DELIMITER ;

```

```

DELIMITER $$

```

```

CREATE PROCEDURE obtener_cuenta_acceso(IN_nombre_usuario VARCHAR(95),
IN_contraseña VARCHAR(95) )
BEGIN
SELECT US.nombre_usuario, US.contraseña, US.numero_personal,
PERS.nombre, PERS.rfc, PERS.correo, PERS.fecha_nacimiento,
DOC.anos_experiencia, DOC.perfil_profesional, DOC.numero_encargado,
DIC.sindicato,
ENC.area
FROM Usuario AS US
INNER JOIN Personal AS PERS ON PERS.numero_personal =
US.numero_personal AND US.nombre_usuario = IN_nombre_usuario AND
US.contraseña = IN_contraseña
LEFT JOIN Docente AS DOC ON DOC.numero_personal = US.numero_personal
LEFT JOIN Directivo AS DIC ON DIC.numero_personal = US.numero_personal
LEFT JOIN Encargado AS ENC ON ENC.numero_personal =
US.numero_personal;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE obtener_docentes()
BEGIN
SELECT US.nombre_usuario, US.contraseña, US.numero_personal,
PERS.nombre, PERS.rfc, PERS.correo, PERS.fecha_nacimiento,
DOC.anos_experiencia, DOC.perfil_profesional, DOC.numero_encargado
FROM Usuario AS US
INNER JOIN Personal AS PERS ON PERS.numero_personal =
US.numero_personal
INNER JOIN Docente AS DOC ON DOC.numero_personal =
US.numero_personal;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE obtener_cursos()
BEGIN
SELECT CUR.clave_curso, CUR.nombre, CUR.descripcion, CUR.seccion,
CUR.horario, CUR.duracion,
DOC.numero_personal, DOC.anos_experiencia, DOC.numero_encargado,
DOC.perfil_profesional,

```

```
PERS.nombre, PERS.rfc, PERS.correo, PERS.fecha_nacimiento,  
PL.id_plan, PL.fecha_elaboracion, PL.estado, PL.avance,  
PL.fecha_actualizacion  
FROM Curso AS CUR  
LEFT JOIN Docente AS DOC ON DOC.numero_personal =  
CUR.numero_personal  
LEFT JOIN Personal AS PERS ON DOC.numero_personal =  
PERS.numero_personal  
LEFT JOIN Plan AS PL ON CUR.clave_curso = PL.clave_curso;  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE PROCEDURE obtener_cursos_agregado( IN_clave_curso VARCHAR(35) )  
BEGIN  
SELECT clave_curso FROM Curso WHERE clave_curso = IN_clave_curso LIMIT  
1;  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE PROCEDURE obtener_docente_registrado( IN_numero_personal  
VARCHAR(35) )  
BEGIN  
SELECT numero_personal FROM Docente WHERE numero_personal =  
IN_numero_personal LIMIT 1;  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE PROCEDURE obtener_cursos_sin_plan(IN_numero_personal  
VARCHAR(35))  
BEGIN  
SELECT CUR.clave_curso, CUR.nombre, CUR.descripcion, CUR.seccion,  
CUR.horario, CUR.duracion,  
DOC.numero_personal, DOC.anos_experiencia, DOC.numero_encargado,  
DOC.perfil_profesional,  
PERS.nombre, PERS.rfc, PERS.correo, PERS.fecha_nacimiento  
FROM Curso AS CUR  
INNER JOIN Docente AS DOC ON DOC.numero_personal =  
CUR.numero_personal AND DOC.numero_personal = IN_numero_personal  
INNER JOIN Personal AS PERS ON DOC.numero_personal =  
PERS.numero_personal WHERE CUR.clave_curso NOT IN (SELECT clave_curso
```

```
FROM Plan);  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE PROCEDURE agregarPlan( IN_fecha_elaboracion DATE, IN_estado  
ENUM('Activo', 'Inactivo'), IN_avance FLOAT, IN_fecha_actualizacion DATE,  
IN_numero_personal VARCHAR(35), IN_clave_curso VARCHAR(35))  
BEGIN  
INSERT INTO Plan(fecha_elaboracion, estado, avance, fecha_actualizacion,  
numero_personal, clave_curso) VALUES(IN_fecha_elaboracion, IN_estado,  
IN_avance, IN_fecha_actualizacion, IN_numero_personal, IN_clave_curso);  
END $$  
DELIMITER ;
```

```
DELIMITER $$  
CREATE PROCEDURE obtener_cursos_por_profesor(IN_numero_personal  
VARCHAR(35))  
BEGIN  
SELECT CUR.clave_curso, CUR.nombre, CUR.descripcion, CUR.seccion,  
CUR.horario, CUR.duracion,  
DOC.numero_personal, DOC.anos_experiencia, DOC.numero_encargado,  
DOC.perfil_profesional,  
PERS.nombre, PERS.rfc, PERS.correo, PERS.fecha_nacimiento,  
PL.id_plan, PL.fecha_elaboracion, PL.estado, PL.avance,  
PL.fecha_actualizacion  
FROM Curso AS CUR  
LEFT JOIN Docente AS DOC ON DOC.numero_personal =  
CUR.numero_personal AND DOC.numero_personal = IN_numero_personal  
LEFT JOIN Personal AS PERS ON DOC.numero_personal =  
PERS.numero_personal  
LEFT JOIN Plan AS PL ON CUR.clave_curso = PL.clave_curso;  
END $$  
DELIMITER ;
```

```
IN_nombre_tema VARCHAR(35), IN_estado_tema ENUM('Visto', 'Pendiente'),  
IN_fecha_tema DATE
```

```
INSERT INTO Tema(nombre, estado, clave_curso, fecha)  
VALUES(IN_nombre_tema, IN_estado_tema, IN_clave_curso,IN_fecha_tema);
```

```
INSERT INTO Personal(numero_personal, nombre, rfc, correo, fecha_nacimiento)  
VALUES('A1', 'Alberto Reyes', 'ADFKOAS234023', 'asd@gmail.com', '2020/05/05');  
INSERT INTO Usuario(nombre_usuario, contraseña, numero_personal) VALUES  
('usuarioA', SHA2('1', 256), 'A1');
```