



**BASE DE DATOS DE
REGISTRO PARA
INSTITUCIÓN EDUCATIVA
DE NIVEL PRIMARIO.**

ÍNDICE

1. Diseñador, fecha de inicio y fin del proyecto	Página 2
2. Descripción General del Proyecto	Página 3
3. Entidades creadas	Página 4
4. Diagrama Relacional	Página 5
5. Diagrama Relacional desde WORKBENCH	Página 6
6. Creación de la BD y tablas	página 7-8
7. Creación y descripción de vistas.....	Página 9-11
8. Creación de Funciones	Página 12-13
9. Creación de Stored Procedure	Página 14-15
10. Creación de Triggers	Página 16-18
11. Final proyecto y Link a script sql	Página 19

FECHA DE INICIO	16/05/2023
-----------------	------------

FECHA DE FINAL	9/07/2023
----------------	-----------

<p>DISEÑADOR DE LA BASE DE DATOS Sanchez Agustin Leonel</p>



Descripción del proyecto

Objetivo Principal:

- Almacenado de datos.
- Consistencia de la información.
- Registro y creación de alumnos y maestros Base de datos.

Modelo de Negocio:

Este Proyecto se centra en una base de datos relacional para un aplicación web de gestión interna de un sistema escolar primario, este proyecto está en una etapa inicial y tiene como principal característica gestionar de manera eficiente la información relacionada con sus alumnos, directivos y maestros.

La implementación de esta base de datos facilitará el registro y almacenamiento de los datos de los alumnos, maestros y también almacenará sus nombres, edades, direcciones y cualquier otra información relevante. Esto permitirá a las instituciones tener un acceso rápido y sencillo a la información de cada estudiante, lo que resultará en una gestión más efectiva y organizada.

Además, la base de datos permitirá almacenar las calificaciones cuatrimestrales de los alumnos en cada asignatura. Esto proporcionará una herramienta centralizada para registrar las calificaciones de los estudiantes de manera oportuna y precisa. Los maestros podrán ingresar las calificaciones de manera sistemática.

ENTIDADES DE LA BASE DE DATOS

Tablas en workbench

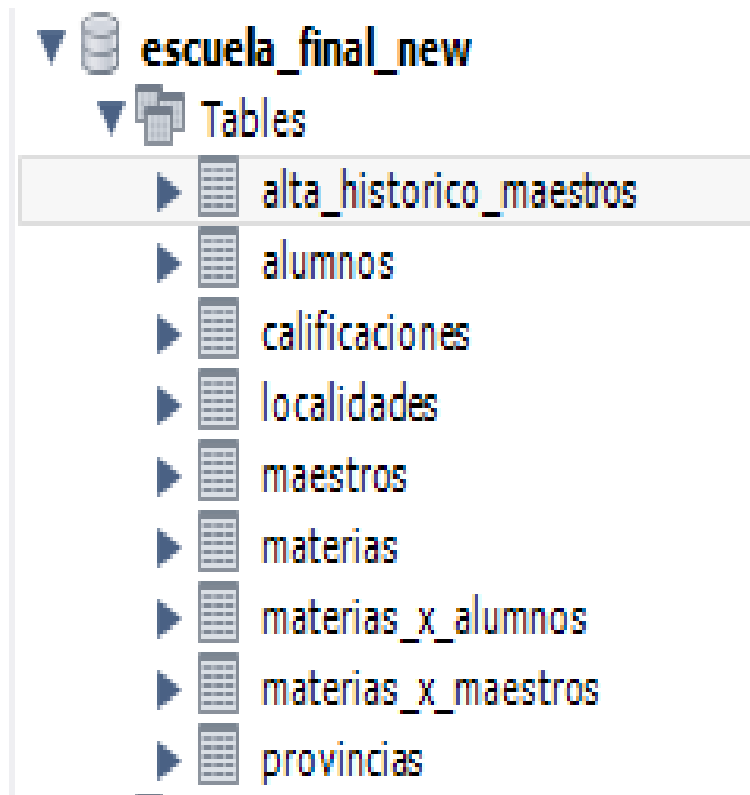
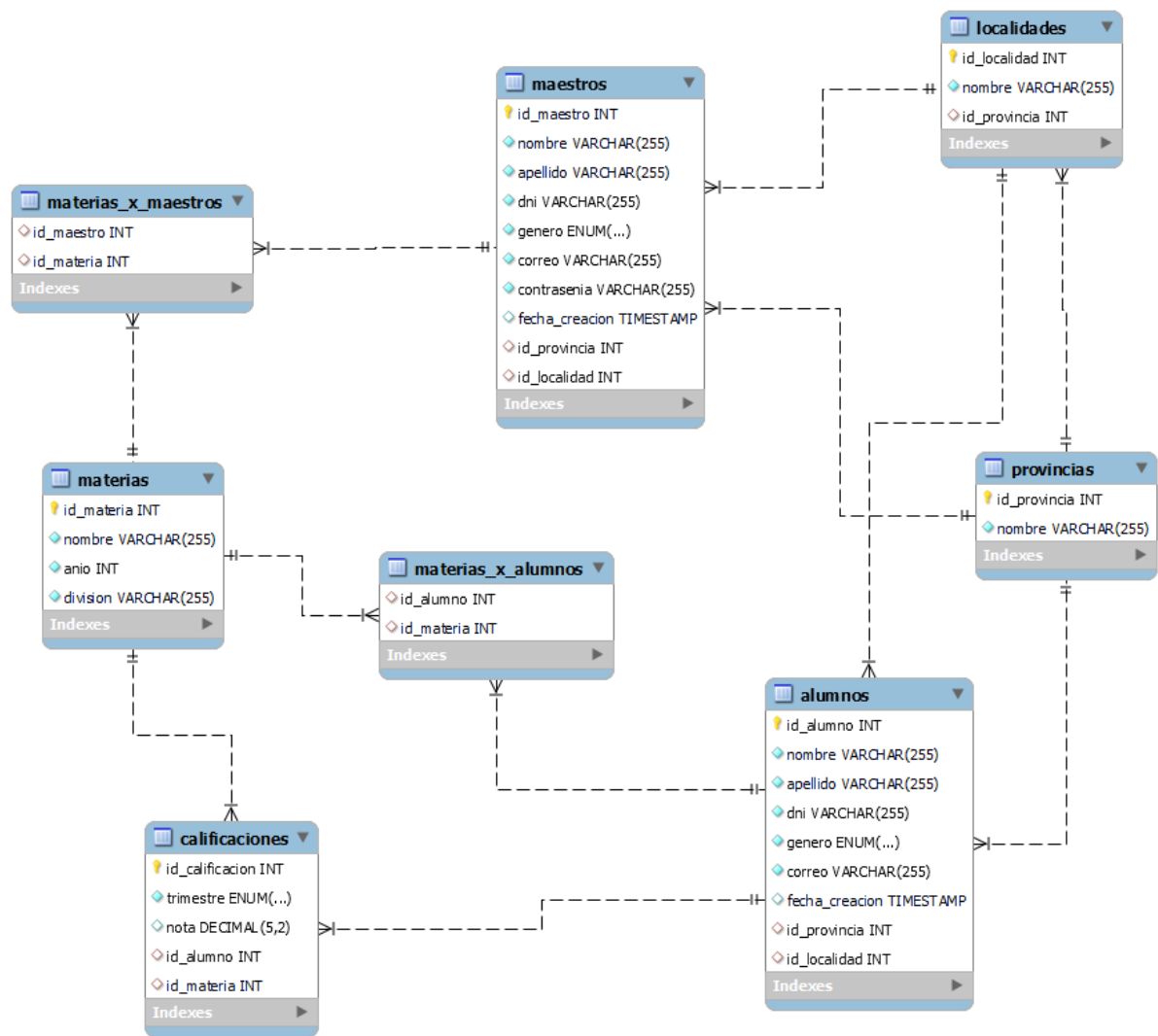


DIAGRAMA RELACIONAL DESDE WORKBENCH

Este diagrama está generado con el motor de base de datos de WORKBENCH utilizando Reverse Engineer



CREACIÓN DE LA BASE DE DATOS Y TABLAS.

```
1 • CREATE DATABASE IF NOT EXISTS escuela_final_new;
2
3 • use escuela_final_new;
4
5 -- TABLA PROVINCIAS
6 • CREATE TABLE IF NOT EXISTS provincias (
7     id_provincia INT PRIMARY KEY auto_increment,
8     nombre VARCHAR(255) NOT NULL
9 );
10
11 -- TABLA LOCALIDADES
12 • CREATE TABLE IF NOT EXISTS Localidades (
13     id_localidad INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
14     nombre varchar(255) NOT NULL,
15     id_provincia INT ,
16     FOREIGN KEY (id_provincia) REFERENCES provincias(id_provincia)
17 );
18
19 -- TABLA MATERIAS
20 • CREATE TABLE IF NOT EXISTS materias(
21     id_materia INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
22     nombre VARCHAR(255) NOT NULL,
23     anio int not null,
24     division VARCHAR (255) NOT NULL
25 );
```

```

27      -- TABLA MAESTROS
28  * CREATE TABLE IF NOT EXISTS maestros(
29      id_maestro INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
30      nombre VARCHAR(255) NOT NULL,
31      apellido VARCHAR(255) NOT NULL,
32      dni VARCHAR(255) NOT NULL,
33      genero ENUM ('Masculino','Femenino','No binario') NOT NULL,
34      correo VARCHAR(255) NOT NULL,
35      contraseña VARCHAR(255) NOT NULL,
36      fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
37      id_provincia INT ,
38      FOREIGN KEY (id_provincia) REFERENCES provincias(id_provincia),
39      id_localidad INT ,
40      FOREIGN KEY (id_localidad) REFERENCES Localidades(id_localidad)
41  );
42
43      -- TABLA ALUMNOS
44  * CREATE TABLE IF NOT EXISTS alumnos(
45      id_alumno INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
46      nombre VARCHAR(255) NOT NULL,
47      apellido VARCHAR(255) NOT NULL,
48      dni VARCHAR(255) NOT NULL,
49      genero ENUM ('Masculino','Femenino','No binario') NOT NULL,
50      correo VARCHAR(255) NOT NULL,
51      fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
52      id_provincia INT ,
53      FOREIGN KEY (id_provincia) REFERENCES provincias(id_provincia),
54      id_localidad INT ,
55      FOREIGN KEY (id_localidad) REFERENCES Localidades(id_localidad)
56  );
57

```

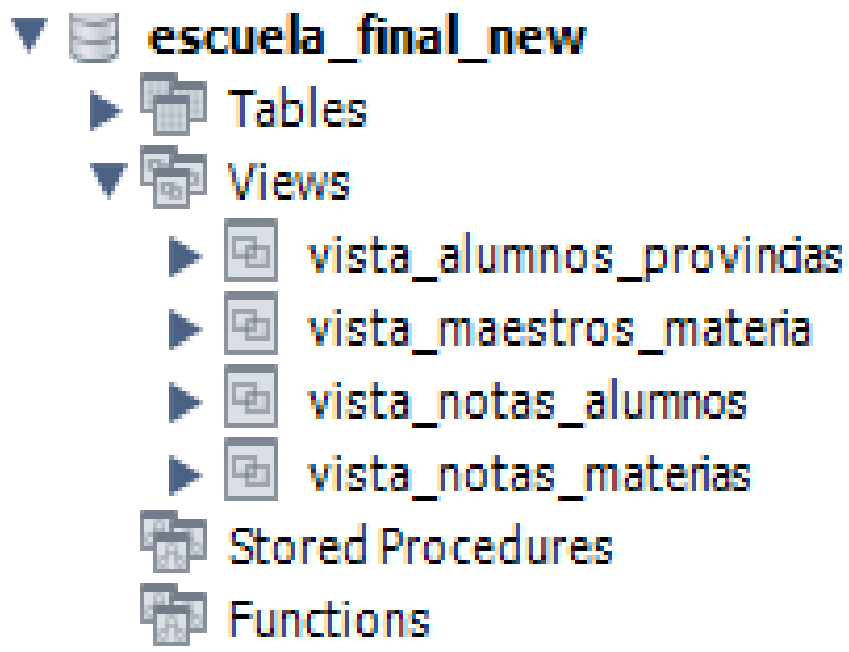


```

59  -- TABLA CALIFICACIONES
60  ● CREATE TABLE IF NOT EXISTS calificaciones (
61      id_calificacion INT PRIMARY KEY AUTO_INCREMENT,
62      trimestre enum('Primer trimestre','Segundo trimestre','Tercer trimestre') NOT NULL,
63      nota DECIMAL(5,2),
64      id_alumno INT,
65      FOREIGN KEY (id_alumno) REFERENCES alumnos(id_alumno),
66      id_materia INT,
67      FOREIGN KEY (id_materia) REFERENCES materias(id_materia)
68  );
69
70  -- TABLA MATERIAS_X_MAESTROS
71  ● CREATE TABLE IF NOT EXISTS materias_x_maestros(
72      id_maestro INT,
73      id_materia INT,
74      FOREIGN KEY (id_maestro) REFERENCES maestros(id_maestro),
75      FOREIGN KEY (id_materia) REFERENCES materias(id_materia)
76  );
77
78
79  -- TABLA MATERIAS_X_ALUMNOS
80  ● CREATE TABLE IF NOT EXISTS materias_x_alumnos(
81      id_alumno INT,
82      id_materia INT,
83      FOREIGN KEY (id_alumno) REFERENCES alumnos(id_alumno),
84      FOREIGN KEY (id_materia) REFERENCES materias(id_materia)
85  );
86

```

Creación y descripción de vistas



Esta base de datos tendrá las siguientes cuatro vistas:

Estas vistas te permiten consultar los datos de forma más sencilla y organizada, ya que encapsulan las consultas complejas en objetos reutilizables.

1) Vista_alumnos_provincias

Esta vista me devuelve todos los alumnos con nombre y apellido y su provincia correspondiente.

```
CREATE VIEW vista_alumnos_provincias AS
SELECT a.nombre AS "Nombre del alumno", a.apellido AS "Apellido del alumno", p.nombre AS "Provincia del alumno"
FROM alumnos AS a
INNER JOIN provincias AS p ON a.id_provincia = p.id_provincia;
```

	Nombre del alumno	Apellido del alumno	Provincia del alumno
►	Juan	Pérez	Buenos Aires
	María	González	Ciudad Autónoma de Buenos Aires
	Carlos	López	Buenos Aires
	Laura	Rodríguez	Buenos Aires
	Andrés	Fernández	Buenos Aires
	Ana	López	Buenos Aires
	Pedro	Gómez	Ciudad Autónoma de Buenos Aires
	Marta	Martínez	Ciudad Autónoma de Buenos Aires
	Luis	García	Buenos Aires
	Sofía	Sánchez	Buenos Aires

2) Vista_maestros_materias

Esta vista me devuelve todos los maestros con nombre , apellido y materia que imparte por cada división. Los maestros pueden estar en varias materias a lo largo del año

```
CREATE VIEW vista_maestros_materia AS
SELECT a.id_alumno, a.nombre AS "Nombre del maestro", a.apellido AS "Apellido del maestro",
m.nombre AS "Nombre de la materia que imparte", m.anio, m.division
FROM alumnos AS a
INNER JOIN materias_x_maestros AS mm ON a.id_alumno = mm.id_maestro
INNER JOIN materias AS m ON mm.id_materia = m.id_materia;
```

	id_alumno	Nombre del maestro	Apellido del maestro	Nombre de la materia que imparte	anio	division
►	1	Juan	Pérez	Lengua	2023	1° "A"
	1	Juan	Pérez	Matematica	2023	1° "A"
	2	María	González	Geografia	2023	1° "A"
	3	Carlos	López	Musica	2023	1° "A"
	4	Laura	Rodríguez	Economia Moderna	2023	1° "A"

3) Vista_notas_alumnos

Esta vista me devuelve todos los alumnos con su nombre, apellido, trimestre, nota, nombre de la materia, anio, división y también todos aquellos que no tienen una nota ni materia asignada

```
CREATE VIEW vista_notas_alumnos AS
SELECT a.id_alumno, a.nombre, a.apellido, c.trimestre, c.nota, m.nombre AS nombre_materia, m.anio, m.division
FROM alumnos AS a
LEFT JOIN calificaciones AS c ON a.id_alumno = c.id_alumno
LEFT JOIN materias AS m ON c.id_materia = m.id_materia;
```

	id_alumno	nombre	apellido	trimestre	nota	nombre_materia	anio	division
►	1	Juan	Pérez	Tercer trimestre	10.00	Lengua	2023	1° "A"
	1	Juan	Pérez	Segundo trimestre	5.00	Lengua	2023	1° "A"
	1	Juan	Pérez	Primer trimestre	8.00	Lengua	2023	1° "A"
	2	María	González	Segundo trimestre	10.00	Matematica	2023	1° "A"
	2	María	González	Primer trimestre	4.50	Matematica	2023	1° "A"
	3	Carlos	López	NULL	NULL	NULL	NULL	NULL
	4	Laura	Rodríguez	NULL	NULL	NULL	NULL	NULL
	5	Andrés	Fernández	NULL	NULL	NULL	NULL	NULL
	6	Ana	López	NULL	NULL	NULL	NULL	NULL
	7	Pedro	Gómez	NULL	NULL	NULL	NULL	NULL

4) Vista_notas_materias

Esta vista me devuelve todos los alumnos con sus notas,nombre y los trimestres con notas asignadas con la condición que sea en la materia lengua y división 1° "A".

```
CREATE VIEW vista_notas_materias AS
SELECT a.id_alumno, a.nombre, a.apellido, c.trimestre, c.nota, m.nombre AS nombre_materia, m.anio, m.division
FROM alumnos AS a
LEFT JOIN calificaciones AS c ON a.id_alumno = c.id_alumno
LEFT JOIN materias AS m ON c.id_materia = m.id_materia
WHERE m.nombre='Lengua' and m.division = '1° "A"';
```

	id_alumno	nombre	apellido	trimestre	nota	nombre_materia	anio	division
►	1	Juan	Pérez	Primer trimestre	8.00	Lengua	2023	1° "A"
	1	Juan	Pérez	Segundo trimestre	5.00	Lengua	2023	1° "A"
	1	Juan	Pérez	Tercer trimestre	10.00	Lengua	2023	1° "A"

Creación y descripción de Funciones



Functions

f() get_Provincia

f() get_Provincia_localidad

Esta base de datos tendrá las siguientes dos Funciones:

FUNC NÚMERO 1: F() GET_PROVINCIA

-Esta función permitirá ver en la tabla alumnos la provincias de cada uno dependiendo el Identificador asignado a la provincia.

```
DELIMITER $$
CREATE FUNCTION get_Provincia (p_id_provincia INT)
RETURNS VARCHAR(255)
READS SQL DATA
BEGIN
    DECLARE resultado VARCHAR(255);
    SET resultado =(SELECT nombre FROM provincias where id_provincia = p_id_provincia);
    return resultado;
END $$

SELECT nombre, apellido, dni, genero, get_Provincia(id_provincia) AS 'Provincia', id_localidad FROM alumnos;
```

nombre	apellido	dni	genero	Provincia	id_localidad
Juan	Pérez	12345678	Masculino	Buenos Aires	1
María	González	87654321	Femenino	Ciudad Autónoma de Buenos Aires	14
Carlos	López	98765432	Masculino	Buenos Aires	2
Laura	Rodríguez	54321678	Femenino	Buenos Aires	3
Andrés	Fernández	87654321	Masculino	Buenos Aires	5
Ana	López	65432187	Femenino	Buenos Aires	6
Pedro	Gómez	98765432	Masculino	Ciudad Autónoma de Buenos Aires	16
Marta	Martínez	12347865	Femenino	Ciudad Autónoma de Buenos Aires	15
Luis	García	87651234	Masculino	Buenos Aires	1
Sofía	Sánchez	43215678	Femenino	Buenos Aires	7

FUNC NÚMERO 2: F) GET_PROVINCIA_LOCALIDAD

-Esta función es parecida a la anterior solo que tiene una peculiaridad permitirá ver en la tabla alumnos la provincias y localidades de cada alumno de forma concatenada.

```
DELIMITER $$
CREATE FUNCTION get_Provincia_localidad(p_id_provincia INT, p_id_localidad INT)
RETURNS VARCHAR(255)
READS SQL DATA
BEGIN
    DECLARE resultado VARCHAR(255);
    DECLARE nombre_provincia VARCHAR(255);
    DECLARE nombre_localidad VARCHAR(255);

    SELECT nombre INTO nombre_provincia
    FROM provincias
    WHERE id_provincia = p_id_provincia;

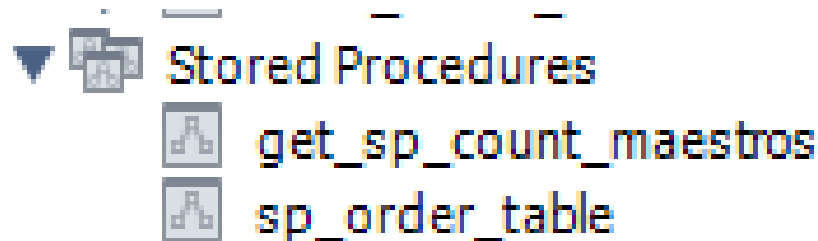
    SELECT nombre INTO nombre_localidad
    FROM localidades
    WHERE id_localidad = p_id_localidad;

    SET resultado = CONCAT('PROVINCIA: ', nombre_provincia, ', ' Y LOCALIDAD: ', nombre_localidad, '.');

    RETURN resultado;
END $$
SELECT nombre, apellido, dni, genero, get_Provincia_localidad(id_provincia,id_localidad)AS 'Provincia y localidad' from alumnos;
```

nombre	apellido	dni	genero	Provincia y localidad
Juan	Pérez	12345678	Masculino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Tres De Febrero.
María	González	87654321	Femenino	PROVINCIA: Ciudad Autónoma de Buenos Aires. Y LOCALIDAD: Bermejo.
Carlos	López	98765432	Masculino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Zarate.
Laura	Rodríguez	54321678	Femenino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Adolfo Alsina.
Andrés	Fernández	87654321	Masculino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Alberti.
Ana	López	65432187	Femenino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Almirante Brown.
Pedro	Gómez	98765432	Masculino	PROVINCIA: Ciudad Autónoma de Buenos Aires. Y LOCALIDAD: Caballito.
Marta	Martínez	12347865	Femenino	PROVINCIA: Ciudad Autónoma de Buenos Aires. Y LOCALIDAD: Boedo.
Luis	García	87651234	Masculino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Tres De Febrero.
Sofía	Sánchez	43215678	Femenino	PROVINCIA: Buenos Aires. Y LOCALIDAD: Arrecifes.

Creación y descripción de stored Procedure



Esta base de datos tendrá las siguientes dos SP:

SP NÚMERO 1: GET_SP_COUNT_MAESTROS

-Este stored procedure permitirá hacer el recuento total de registros de maestros en la tabla "maestros" de nuestra base de datos.

```
DELIMITER $$  
CREATE PROCEDURE get_sp_count_maestros(OUT recuento int)  
BEGIN  
    SELECT count(*) INTO recuento from maestros;  
END $$  
  
SET @recuento=0;  
call get_sp_count_maestros(@recuento);  
SELECT @recuento as 'Total maestros registrados';
```

	Total maestros registrados
▶	5

SP NÚMERO 1: SP_ORDER_TABLE

-Este stored procedure permitirá ordenar cualquier tabla de manera Ascendente o descendente

```
DELIMITER $$
CREATE PROCEDURE sp_order_table(IN campo varchar(255), IN tipo_ordenamiento ENUM('ASC','DESC',''), IN tabla varchar(255))
BEGIN
    IF campo <> '' THEN
        SET @ordenar = CONCAT(' ORDER BY ',campo);
    ELSE
        SET @ordenar = '';
    END IF ;

    IF tipo_ordenamiento <> '' THEN
        SET @tipo = CONCAT(' ',tipo_ordenamiento);
    ELSE
        SET @tipo = '';
    END IF ;

    SET @clausula = CONCAT('SELECT * FROM ', tabla ,@ordenar, @tipo);
    PREPARE runSQL FROM @clausula;
    EXECUTE runSQL;
    DEALLOCATE PREPARE runSQL;
END $$
CALL sp_order_table('nombre','DESC','alumnos')
```

id_alumno	nombre	apellido	dni	genero	correo	fecha_creacion	id_provincia	id_localidad
10	Sofía	Sánchez	43215678	Femenino	sofia@example.com	2023-06-14 21:15:16	1	7
7	Pedro	Gómez	98765432	Masculino	pedro@example.com	2023-06-14 21:15:16	2	16
8	Marta	Martínez	12347865	Femenino	marta@example.com	2023-06-14 21:15:16	2	15
2	María	González	87654321	Femenino	maria@example.com	2023-06-14 21:15:16	2	14
9	Luis	García	87651234	Masculino	luis@example.com	2023-06-14 21:15:16	1	1
4	Laura	Rodríguez	54321678	Femenino	laura@example.com	2023-06-14 21:15:16	1	3
1	Juan	Pérez	12345678	Masculino	juan@example.com	2023-06-14 21:15:15	1	1
3	Carlos	López	98765432	Masculino	carlos@example.com	2023-06-14 21:15:16	1	2
5	Andrés	Fernández	87654321	Masculino	andres@example.com	2023-06-14 21:15:16	1	5
6	Ana	López	65432187	Femenino	ana@example.com	2023-06-14 21:15:16	1	6

Creación y descripción de TRIGGERS

TRIGGER NÚMERO 1: COMPROBAR_CORREO

-Este trigger antes (BEFORE) que el usuario maestro sea insertado a la tabla maestros dispara una validación que comprueba si el correo electrónico ingresado sea válido en nuestra base de datos si no es válido arroja un error por consola que dice el siguiente mensaje **'El correo electrónico no tiene un formato válido.'**

```
363 DELIMITER $$
364 CREATE TRIGGER comprobar_correo
365 BEFORE INSERT ON maestros
366 FOR EACH ROW
367 BEGIN
368 IF NEW.correo IS NOT NULL AND NEW.correo NOT REGEXP '^([A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,})$' THEN
369     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'El correo electrónico no tiene un formato válido.';
370 END IF;
371 END $$
372 INSERT INTO maestros(id_maestro,nombre,apellido,dni,genero,correo,contrasenia,id_provincia,id_localidad) value(8,'Samuel','Etoo','15857867','Masculino','samuel','Pesadilla1',2,2);
373
374
375
```

Output

Action Output

#	Time	Action
1	12:51:23	INSERT INTO maestros(id_maestro,nombre,apellido,dni,genero,correo,contrasenia,id_provincia,id_localidad) value(8,'Samuel','Etoo','15857867','Ma...

Message

Error Code: 1644. El correo electrónico no tiene un formato válido.

TRIGGER NÚMERO 2: HISTORICO_MAESTRO

-Este trigger después (AFTER) que un usuario maestro se inserta en la tabla maestros dispara una acción que almacena un registro histórico, la importancia de este trigger es almacena los datos históricos de la fecha de ALTA, fecha de carga, nombre, apellido, etc del usuario maestro y que tipo de USUARIO hace esa inserción a nuestra base de datos.

CREACIÓN TABLA ALTA_HISTORICO_MAESTRO

```
CREATE TABLE IF NOT EXISTS alta_historico_maestros(  
  id INT,  
  nombre_maestro VARCHAR(255),  
  apellido_maestro VARCHAR(255),  
  dni_maestro VARCHAR(255),  
  fecha_hora DATETIME,  
  usuario VARCHAR(255),  
  operacion varchar(255)  
);
```

CREACIÓN TRIGGER HISTORICO_MAESTRO

```
DELIMITER $$  
CREATE TRIGGER historico_maestro  
AFTER INSERT ON maestros  
FOR EACH ROW  
BEGIN  
  INSERT INTO alta_historico_maestros (id,nombre_maestro,apellido_maestro,dni_maestro,fecha_hora,usuario,operacion)value(  
    NEW.id_maestro, NEW.nombre, NEW.apellido, NEW.dni, now(), user(), "ALTA MAESTRO"  
  );  
END $$
```

Luego de la inserción a la tabla maestros nuestra tabla de registro histórico quedaría de la siguiente manera:

Result Grid						
		Filter Rows:		Export:	Wrap Cell Content:	
	id	nombre_maestro	apellido_maestro	dni_maestro	fecha_hora	operacion
▶	6	Miguel	Rios	17857888	2023-07-09 11:51:40	root@localhost ALTA MAESTRO
	7	Luis	Pirilli	19857888	2023-07-09 12:02:35	root@localhost ALTA MAESTRO

FIN DEL PROYECTO.

LINKS AL SCRIPT DE LA BASE DE DATOS:

https://drive.google.com/drive/folders/1HS7YKmc9yG3_Kba6lkq4EXPtj7YTz0_s?usp=drive_link