

Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

USE master;

GO

CREATE OR ALTER PROCEDURE sp\_crearobaseytablas AS

BEGIN SET NOCOUNT ON;

IF DB\_ID ('Plataforma Cursos') IS NOT NULL

DROP DATABASE Plataforma Cursos;

CREATE DATABASE Plataforma Cursos;

EXEC C'

CREATE TABLE Usuarios C

Usuario ID INT PRIMARY KEY IDENTITY,

Nombre NVARCHAR (100) NOT NULL,

Email NVARCHAR (100) NOT NULL UNIQUE,

FechaRegistro DATE NOT NULL DEFAULT GETDATE ()

);

CREATE TABLE Cursos C

Curso ID PRIMARY KEY IDENTITY,

Titulo NVARCHAR (100) NOT NULL,

Descripcion NVARCHAR (255),

FechaExamen DATE NOT NULL DEFAULT GETDATE ()

);

CREATE TABLE Categorías C

Categorías ID INT PRIMARY KEY IDENTITY,

Nombre NVARCHAR (50) NOT NULL UNIQUE

);



CREATE TABLE Instructores (

InstructorID NVARCHAR(100) NOT NULL,

Especialidad NVARCHAR(100) NOT NULL,

nombre NVARCHAR(100) NOT NULL,

CONSTRAINT Ck\_Especialidad CHECK (LEN(Especialidad) > 3)

);

CREATE TABLE certificados (

CertificadoID INT PRIMARY KEY IDENTITY,

UsuarioID NOT NULL,

CursoID NOT NULL,

FechaEntrega DATE NOT NULL DEFAULT GETDATE(),

FOREIGN KEY (UsuarioID) REFERENCES Usuarios (UsuarioID),

FOREIGN KEY (CursoID) REFERENCES Cursos (CursoID)

);

CREATE TABLE CursoCategorial

CursoID INT NOT NULL,

CategoriaID INT NOT NULL,

PRIMARY KEY (CursoID, CategoriaID),

FOREIGN KEY (CursoID) REFERENCES Cursos (CursoID),

FOREIGN KEY (CategoriaID) REFERENCES Categorias (CategoriaID)

);

CREATE TABLE CursoUsuario (

CursoID INT NOT NULL,

UsuarioID INT NOT NULL,

FechaInscripcion DATE NOT NULL DEFAULT GETDATE(),

PRIMARY KEY (CursoID, UsuarioID),

FOREIGN KEY REFERENCES Cursos (CursoID),

FOREIGN KEY REFERENCES Usuarios (UsuarioID)

);



EXEC sp\_crear base de datos y tablas;

GO

USE plataforma cursos;

GO

INSERT INTO usuarios (Nombre, email) VALUES

('Ana Torres', 'anao@gmail.com'), ('Luis Perez', 'luis@gmail.com'), ('Marta Lopez', 'marta@gmail.com'), ('Juan Ruiz', 'juanv@gmail.com'), ('Carmen Diaz', 'carmen@gmail.com'), ('Pedro Nunez', 'pedro@gmail.com'), ('Lucia Fernandez', 'lucia@gmail.com'), ('Hugo Ortega', 'hugo@gmail.com'), ('Sofia Romero', 'sofia@gmail.com'), ('Diego Vargas', 'diego@gmail.com');

INSERT INTO cursos (Titulo, Descripcion) VALUES

('Curso HTML', 'Instruccion al HTML'), ('Curso CSS', 'Estilos con CSS'), ('Curso JS', 'Javascript Basico'), ('Curso SQL', 'Bases de datos en SQL'), ('Curso Python', 'Python desde cero'), ('Curso Excel', 'Excel avanzado'), ('Curso Marketing', 'Marketing digital'), ('Curso UX', 'Diseño UX/UI'), ('Curso Power BI', 'Visualizacion de datos'), ('Curso WordPress', 'Sitios con WordPress');

INSERT INTO categorias (Nombre) VALUES

('Programacion'), ('Diseño'), ('Marketing'), ('Ofimatica'), ('Datos'), ('Base de datos'), ('web'), ('negocios'), ('Educacion'), ('tecnologia');

INSERT INTO instructores (Nombre, Especialidad) VALUES

('Carlos Ruiz', 'Programacion'), ('Lourdes Medina', 'Diseño UX/UI'), ('Jorge Silva', 'Marketing'), ('Ana Beltran', 'Educacion digital'), ('Esteban Cruz', 'Analisis de datos'), ('Paula Ortiz', 'Base de datos'), ('Nicolas Bravo', 'Negocios online'), ('Sandra Ramos', 'Visualizacion de datos'), ('Diego Peña', 'Tecnologia web'), ('Marta Gomez', 'Productividad y ofimatica');



```
INSERT INTO Certificados (UsuarioID, CursoID) VALUES  
(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9), (10,10);
```

```
INSERT INTO CursoCategoría (CursoID, CategoríaID) VALUES  
(1,1), (2,1), (3,1), (4,6), (5,1), (6,4), (7,3), (8,2), (9,5), (10,7);
```

```
INSERT INTO CursoUsuario (CursoID, UsuarioID) VALUES  
(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (8,8), (9,9), (10,10);
```

```
CREATE OR ALTER VIEW Vista_UsuariosConCursos AS
```

```
SELECT
```

```
Usuarios.nombre AS NombreUsuario, Cursos.titulo AS NombreCurso,
```

```
CursoUsuario.FechaInscripcion
```

```
FROM CursoUsuario
```

```
JOIN Usuarios ON CursoUsuario.UsuarioID = Usuarios.UsuarioID,
```

```
JOIN Cursos ON CursoUsuario.CursoID = Cursos.CursoID;
```

```
CREATE OR ALTER VIEW Vista_CursosCategorías AS
```

```
SELECT
```

```
Cursos.titulo AS NombreCurso,
```

```
Categorías.nombre AS NombreCategoría
```

```
FROM CursoCategoría
```

```
JOIN Cursos ON CursoCategoría.CursoID = Cursos.CursoID
```

```
JOIN Categorías ON CursoCategoría.CategoríaID = Categorías.CategoríaID;
```

```
CREATE OR ALTER VIEW Vista-certificados AS
```

```
SELECT
```

```
Usuarios.nombre AS NombreUsuario, Cursos.titulo AS NombreCurso, Certificados,
```

```
FechaEntrega
```

```
FROM Certificados
```

```
JOIN Usuarios ON Certificados.UsuarioID = Usuarios.UsuarioID
```

```
JOIN Cursos ON Certificados.CursoID = Cursos.CursoID;
```



Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
CREATE OR ALTER VIEW v_ido_usuariospor curso AS
SELECT
  cursos.titulo AS nombreCurso,
  COUNT(*) AS TotalInscritos
FROM cursos_usuario
JOIN cursos ON cursos_usuario.cursoID = cursos.cursoID
GROUP BY cursos.titulo;

CREATE OR ALTER VIEW v_ido_instructores AS
SELECT instructorID, Nombre AS NombreInstructor, Especialidad
FROM Instructores;

SELECT Nombre AS NombreUsuario
FROM Usuarios

WHERE usuarioID NOT IN (SELECT usuarioID FROM certificados);

SELECT cursos.titulo AS NombreCurso, COUNT(*) AS ContadorInscritos
FROM cursos_usuario
JOIN cursos ON cursos_usuario.cursoID = cursos.cursoID
GROUP BY cursos.titulo
HAVING (COUNT(*) > 1);

SELECT titulo AS NombreCurso
FROM cursos

WHERE cursoID NOT IN (SELECT cursoID FROM cursos_usuario);
```



```
SELECT categories, Nombre AS NombreCategories, COUNT(*) AS TotalUsers  
FROM UsersCategories  
JOIN (categories ON (UsersCategories.categoriesID) = categories.categoriesID)  
GROUP BY categories, Nombre  
ORDER BY Total.Users DESC  
  
SELECT UsersCategories, Nombre AS NombreUsers, COUNT(*) AS UsersIndexID  
FROM UsersCategories  
JOIN Users ON (UsersCategories.UsersIndexID) = Users.UsersIndexID  
GROUP BY UsersCategories, Nombre  
HAVING COUNT(*) > 1;
```