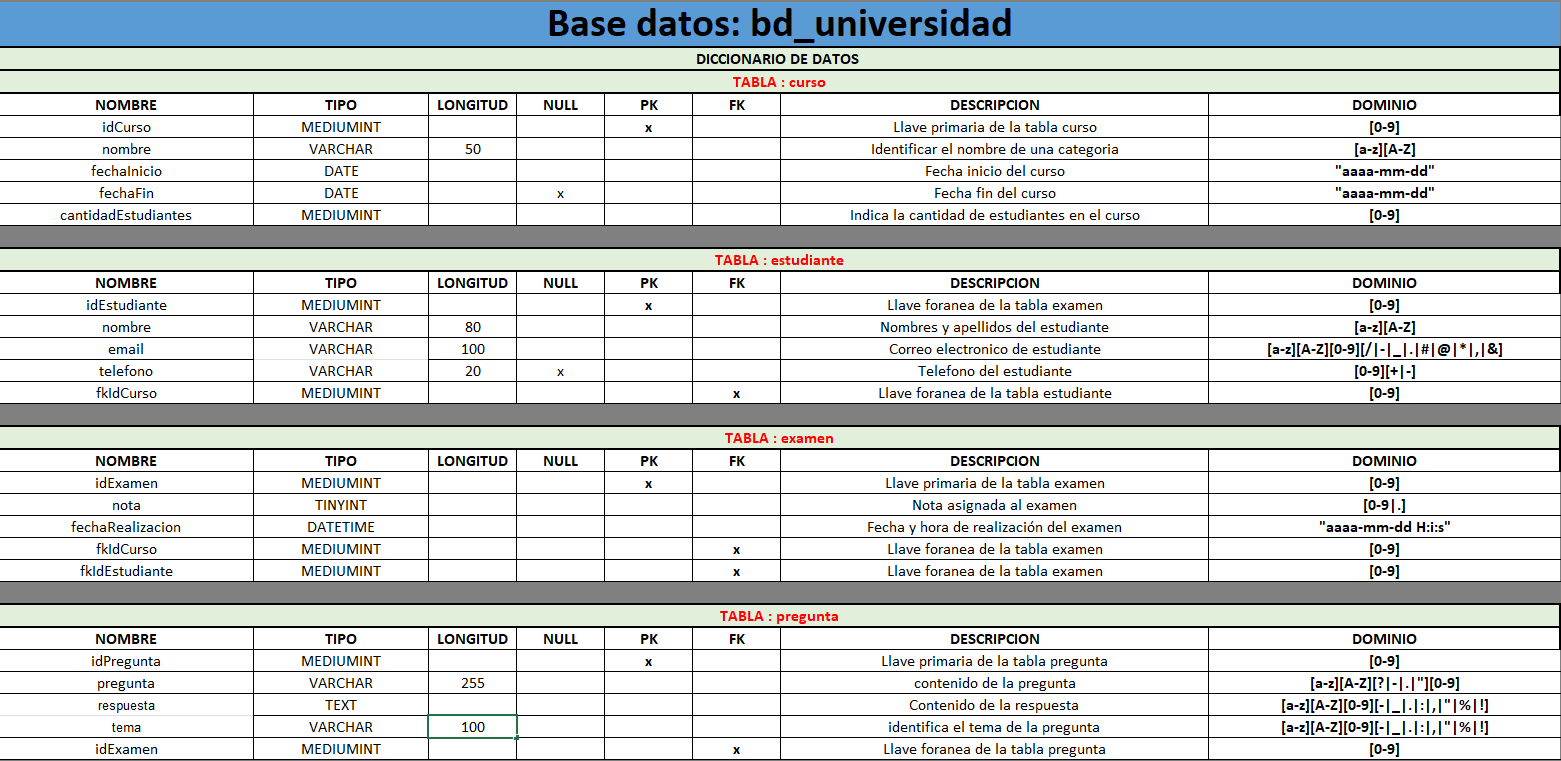
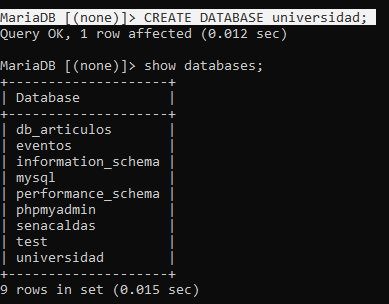
TALLER 1 DE BASE DE DATOS

1. 
2. CREATE DATABASE universidad;



1. CREATE TABLE curso(

idCurso MEDIUMINT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

fechaInicio DATE NOT NUll,

fechaFin DATE,

cantidadEstudiantes MEDIUMINT NOT NULL

);

Texto

Descripción generada automáticamente

CREATE TABLE estudiante(

idEstudiante MEDIUMINT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(80) NOT NULL,

email VARCHAR(100) NOT NULL,

telefono VARCHAR(20),

fkIdCurso MEDIUMINT NOT NULL,

FOREIGN KEY (fkIdCurso) REFERENCES curso(idCurso)

);

Texto

Descripción generada automáticamente

CREATE TABLE examen(

idExamen MEDIUMINT PRIMARY KEY AUTO\_INCREMENT,

nota TINYINT NOT NULL,

fechaRealizacion DATETIME NOT NULL,

fkIdCurso MEDIUMINT NOT NULL,

fkIdEstudiante MEDIUMINT NOT NULL,

FOREIGN KEY (fkIdEstudiante) REFERENCES estudiante(idEstudiante),

FOREIGN KEY (fkIdCurso) REFERENCES curso(idCurso)

);

Texto

Descripción generada automáticamente

CREATE TABLE pregunta(

idPregunta MEDIUMINT PRIMARY KEY AUTO\_INCREMENT,

pregunta VARCHAR(255) NOT NULL,

respuesta TEXT NOT NULL,

tema VARCHAR(100) NOT NULL,

fkIdExamen MEDIUMINT NOT NULL,

FOREIGN KEY (fkIdExamen) REFERENCES examen(idExamen)

);

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

1. # Insertar Datos en la tabla curso

INSERT INTO curso (nombre,fechaInicio,fechaFin,cantidadEstudiantes)

VALUES ('Curso 1','2024-02-12','2025-07-24','10');

INSERT INTO curso (nombre,fechaInicio,fechaFin,cantidadEstudiantes)

VALUES ('Curso 2','2024-03-05','2025-09-30','10');

INSERT INTO curso (nombre,fechaInicio,fechaFin,cantidadEstudiantes)

VALUES ('Curso 3','2024-04-15','2025-11-20','10');

INSERT INTO curso (nombre,fechaInicio,fechaFin,cantidadEstudiantes)

VALUES ('Curso 4','2024-05-30','2026-01-15','10');

Calendario

Descripción generada automáticamente

# Insertar Datos en la tabla estudiante

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Daniel','daniel@gmail.com','31123212',1);

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Isabela','isa@gmail.com','312315434',2);

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Viviana','vivi@gmail.com','356736539',3);

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Sofia','sofi@gmail.com','312374323',5);

Texto

Descripción generada automáticamente

# Insertar Datos en la tabla examen

INSERT INTO examen (fechaRealizacion,fkIdCurso,fkIdEstudiante)

VALUES ('2024-03-15 12:00:00',1,1);

INSERT INTO examen (fechaRealizacion,fkIdCurso,fkIdEstudiante)

VALUES ('2024-04-10 10:00:00',2,2);

INSERT INTO examen (fechaRealizacion,fkIdCurso,fkIdEstudiante)

VALUES ('2024-06-20 4:00:00',5,3);

Imagen que contiene Tabla

Descripción generada automáticamente

# Insertar Datos en la tabla pregunta

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('Cuanto es 1+1','Matematicas',1);

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('Cual es la segunda ley de Newton','Fisica',1);

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('Cual es la capital de Colombia','Sociales',2);

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('Para que se utiliza la tilde','Español',2);

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('Para que funciona if en dart','Programación',4);

INSERT INTO pregunta (pregunta,tema,fkIdExamen)

VALUES ('¿Qué es RAM?','MODELADO de software',4);

Texto

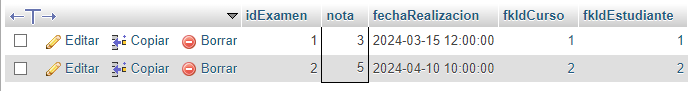
Descripción generada automáticamente

1. Ejecutar las siguientes consultas
   1. Seleccionar los exámenes con nota superior a 3

SELECT \*

FROM examen

WHERE nota >=3;

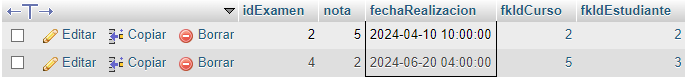


* 1. Seleccionar los exámenes con fecha de realización mayor a 5 de Julio de 2024

SELECT \*

FROM examen

WHERE fechaRealizacion > "2024-04-01";

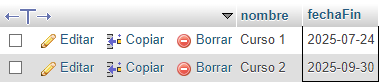


* 1. Seleccionar solo los nombres de cursos, con fecha de finalización menor a 1 de Octubre de 2024

SELECT nombre, fechaFin

FROM curso

WHERE fechaFin <= "2025-10-1";



* 1. Seleccionar los estudiantes cuya nota sea mayor a 3 (Utilizando INNER JOIN)

Seleccionar los estudiantes cuya nota sea mayor a 3 (Utilizando INNER JOIN)

SELECT nombre, email, nota, fechaRealizacion

FROM estudiante INNER JOIN examen

ON estudiante.idEstudiante = examen.fkIdEstudiante;

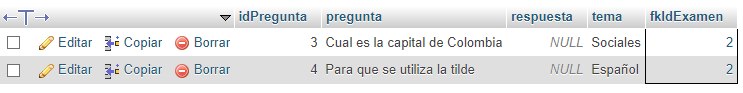


* 1. Seleccionar las preguntas que tiene un examen específico

SELECT \*

FROM pregunta

WHERE fkIdExamen = 2;

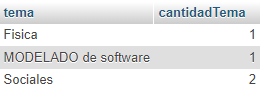


* 1. Seleccionar la cantidad de preguntas que tiene cada tema. Para este punto se debe mostrar una lista con solamente: el tema, la cantidad de preguntas y la fecha de realización del examen (Utilizando GROUP BY)

SELECT tema, COUNT(\*) AS cantidadTema

FROM pregunta

GROUP BY tema;



1. #Modificar el nombre y la fecha de inicio de dos cursos

UPDATE curso

SET nombre = 'Curso 5'

WHERE nombre= 'Curso 4';

UPDATE curso

SET nombre = 'Curso 3-1'

WHERE nombre = 'Curso 3';

#Modificar la fecha de inicio de dos cursos

UPDATE curso

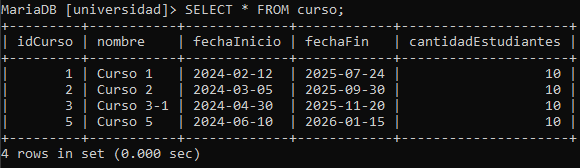
SET fechaInicio = '2024-06-10'

WHERE fechaInicio = '2024-05-30';

UPDATE curso

SET fechaInicio = '2024-04-30'

WHERE fechaInicio = '2024-04-15';



#Modificar email de dos estudiantes

UPDATE estudiante

SET email = 'daniii@gmail.com'

WHERE email = 'daniel@gmail.com';

UPDATE estudiante

SET email = 'viviana@gmail.com'

WHERE email = 'vivi@gmail.com';

#Modificar telefono de dos estudiantes

UPDATE estudiante

SET telefono = '313456545'

WHERE telefono = '3134565456';

UPDATE estudiante

SET telefono = '3124654157'

WHERE telefono = '331274323';

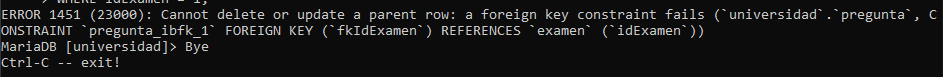
Texto

Descripción generada automáticamente

1. #Eliminar dos exámenes

DELETE FROM examen

WHERE idExamen = 1;



No es posible eliminar los exámenes ya que tienen llaves foráneas asociadas, y su llave primaria esta como llave foránea en la tabla pregunta

#Eliminar dos examenes

DELETE FROM examen

WHERE idExamen = 1;

#Elimintar dos preguntas

DELETE FROM pregunta

WHERE idPregunta = 1;

DELETE FROM pregunta

WHERE idPregunta = 5;

Texto

Descripción generada automáticamente

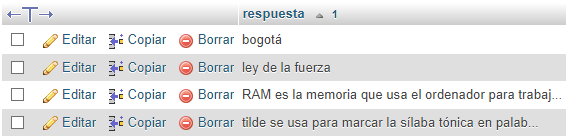
# Puntos adicionales

1. Ordenar las preguntas por el campo “respuesta” de forma descontente y ascendente (utilizando ORDER BY)

SELECT respuesta

FROM pregunta

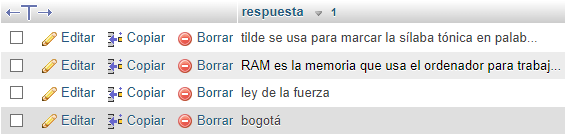
ORDER BY respuesta ASC;



SELECT respuesta

FROM pregunta

ORDER BY respuesta DESC;



1. Ingresar estudiantes para que haya un total de 7 registros en la tabla estudiante. Traer solamente los 4 estudiantes con mejor nota (utilizando LIMIT y ORDER BY)

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Manuel','gonzales@gmail.com','3342452344',1);

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('jose','joseee@gmail.com','3546544677',1);

INSERT INTO estudiante (nombre,email,telefono,fkIdCurso)

VALUES ('Sara','ssssara@gmail.com','3226067640',1);



SELECT \*

FROM examen

ORDER BY nota DESC

LIMIT 4;



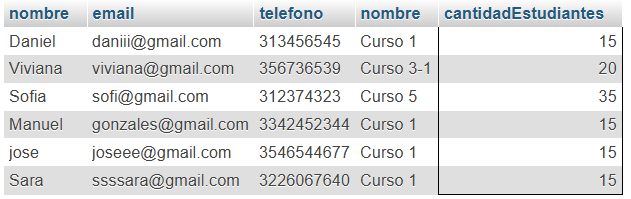
1. Consultar los estudiantes (solamente los campos: nombre, email y teléfono) que pertenezcan a cursos donde la cantidad de estudiantes sea mayor a 10. Traer también el nombre del curso

SELECT estudiante.nombre, email, telefono, curso.nombre, curso.cantidadEstudiantes

FROM estudiante

INNER JOIN curso ON estudiante.fkIdCurso = curso.idCurso

WHERE curso.cantidadEstudiantes > 10;



1. Consultar estudiantes que (utilizando LIKE): - Nombre empiece por la letra "S" - Nombre termine con la letra "l" - Nombre tenga una letra "n" en la mitad

SELECT nombre

FROM estudiante

WHERE nombre LIKE 'S\_%';



SELECT nombre

FROM estudiante

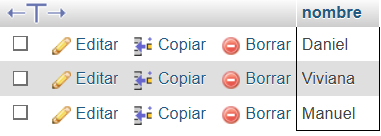
WHERE nombre LIKE '%\_l';



SELECT nombre

FROM estudiante

WHERE nombre LIKE '%\_n\_%';



1. Aplicar cada una de las siguientes funciones de MySql al taller realizado

AVG()

SELECT AVG(nota) AS promedio\_notas

FROM examen;



SUM()

SELECT SUM(nota) AS suma\_notas

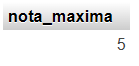
FROM examen;



MAX()

SELECT MAX(nota) AS nota\_maxima

FROM examen;



MIN()

SELECT MIN(nota) AS nota\_minima

FROM examen;

