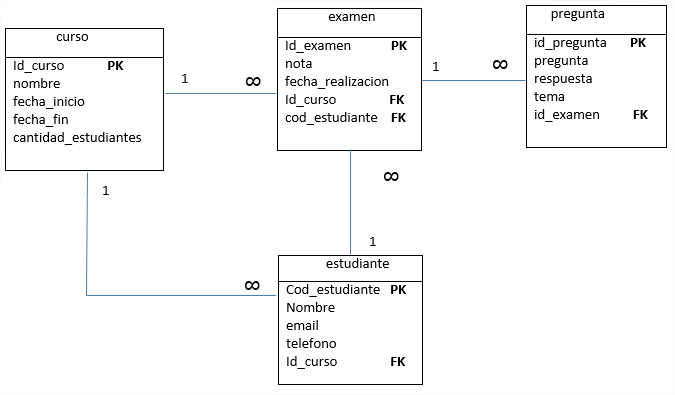
**Taller SQL 01 - ADSO**

1. Realizar el siguiente taller en un documento de Word y presentarlo de manera individual.

* Para los puntos **2** en adelante, se debe poner en este documento el texto de las sentencias SQL realizadas en cada punto
* Para los puntos **5** en adelante, se debe poner pantallazo de evidencia en el documento, al igual que el texto de la sentencia SQL realizada



1. Realizar el diccionario de datos con base al diagrama de tablas anterior
2. Con comandos SQL, crear una base de datos en MySQL llamada: **universidad**.

**CREATE DATABASE** universidad;

1. Con base al diccionario de datos y utilizando comandos SQL, crear las tablas de la base de datos.

CREATE TABLE curso (

id\_curso INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

fecha\_inicio DATE,

fecha\_fin DATE,

cantidad\_estudiantes INT

);

CREATE TABLE estudiante (

cod\_estudiante INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

email VARCHAR(100),

telefono VARCHAR(15),

id\_curso INT,

FOREIGN KEY (id\_curso) REFERENCES curso(id\_curso)

);

CREATE TABLE examen (

id\_examen INT AUTO\_INCREMENT PRIMARY KEY,

nota DECIMAL(3, 2),

fecha\_realizacion DATE,

id\_curso INT,

cod\_estudiante INT,

FOREIGN KEY (id\_curso) REFERENCES curso(id\_curso),

FOREIGN KEY (cod\_estudiante) REFERENCES estudiante(cod\_estudiante)

);

CREATE TABLE pregunta (

id\_pregunta INT AUTO\_INCREMENT PRIMARY KEY,

pregunta TEXT,

respuesta TEXT,

tema VARCHAR(100),

id\_examen INT,

FOREIGN KEY (id\_examen) REFERENCES examen(id\_examen)

);

1. Insertar los siguientes registros:
   1. 4 estudiantes
   2. 3 exámenes (con 2 preguntas cada uno)
   3. 4 cursos

INSERT INTO curso (nombre, fecha\_inicio, fecha\_fin, cantidad\_estudiantes) VALUES

('Curso Algebra', '2024-06-01', '2024-09-30', 30),

('Curso Historia', '2024-06-15', '2024-10-15', 25),

('Curso Biología', '2024-06-20', '2024-09-25', 20);

INSERT INTO estudiante (nombre, email, telefono, id\_curso) VALUES

('Juan Perez', 'juan@example.com', '555-1234', 1),

('Ana Gomez', 'ana@example.com', '555-5678', 2),

('Carlos Ramirez', 'carlos@example.com', '555-8765', 3);

INSERT INTO examen (nota, fecha\_realizacion, id\_curso, cod\_estudiante) VALUES

(4.5, '2024-07-10', 1, 1),

(3.8, '2024-07-20', 2, 2),

(4.0, '2024-07-25', 3, 3);

INSERT INTO pregunta (pregunta, respuesta, tema, id\_examen) VALUES

('¿Cuál es la raíz cuadrada de 16?', '4', 'Matemáticas', 1),

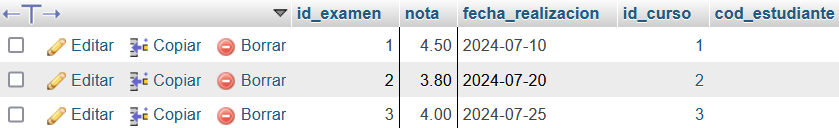
('¿Qué es la mitosis?', 'Proceso de división celular', 'Biología', 3);

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

SELECT \*

FROM examen

WHERE nota > 3;

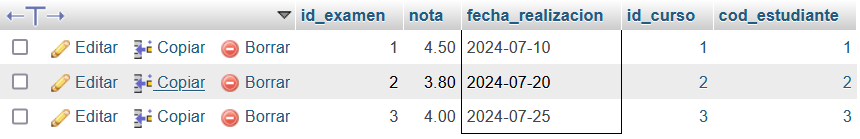


* 1. Seleccionar los exámenes con fecha de realización mayor a 5 de Julio de 2024

SELECT \*

FROM examen

WHERE fecha\_realizacion > '2024-07-05';



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

SELECT nombre

FROM curso

WHERE fecha\_fin < '2024-10-01';



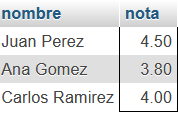
* 1. Seleccionar los estudiantes cuya nota sea mayor a 3 (Utilizando INNER JOIN).

SELECT estudiante.nombre, examen.nota

FROM estudiante

INNER JOIN examen ON estudiante.cod\_estudiante = examen.cod\_estudiante

WHERE examen.nota > 3;



* 1. Seleccionar las preguntas que tiene un examen específico

SELECT \*

FROM pregunta

WHERE id\_examen = ;



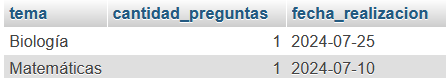
* 1. Seleccionar la cantidad de preguntas que tiene cada tema. Para este punto se debe mostrar una lista con solamente: el tema, la cantidad de preguntas y la fecha de realización del examen (Utilizando GROUP BY)

SELECT tema, COUNT(\*) AS cantidad\_preguntas, examen.fecha\_realizacion

FROM pregunta

INNER JOIN examen ON pregunta.id\_examen = examen.id\_examen

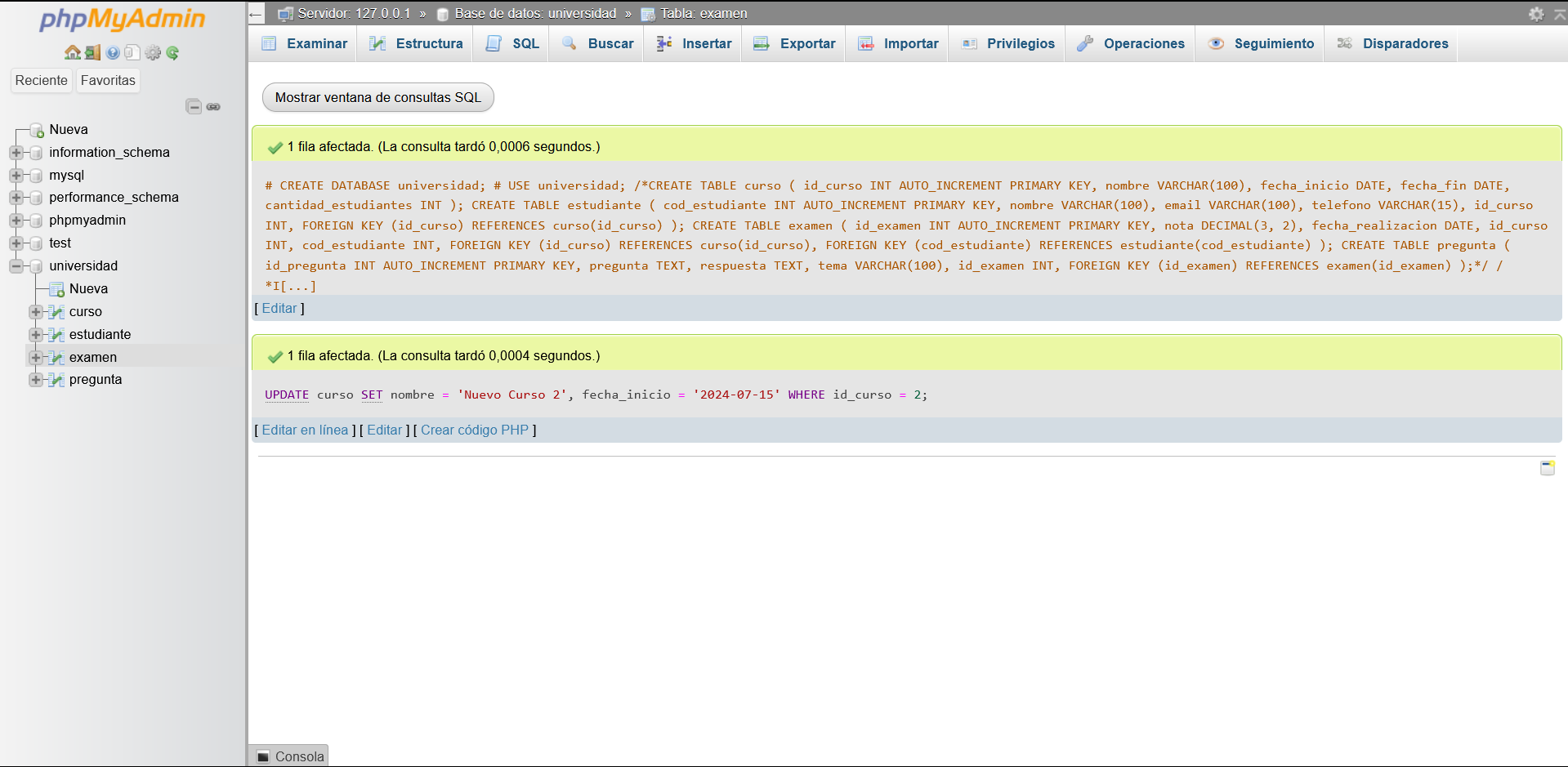
GROUP BY tema, examen.fecha\_realizacion;



1. Actualizar los siguientes registros:
   1. De la tabla cursos, actualizar el nombre y la fecha de inicio de 2 cursos

UPDATE curso SET nombre = 'Nuevo Curso 1', fecha\_inicio = '2024-07-01' WHERE id\_curso = 1;

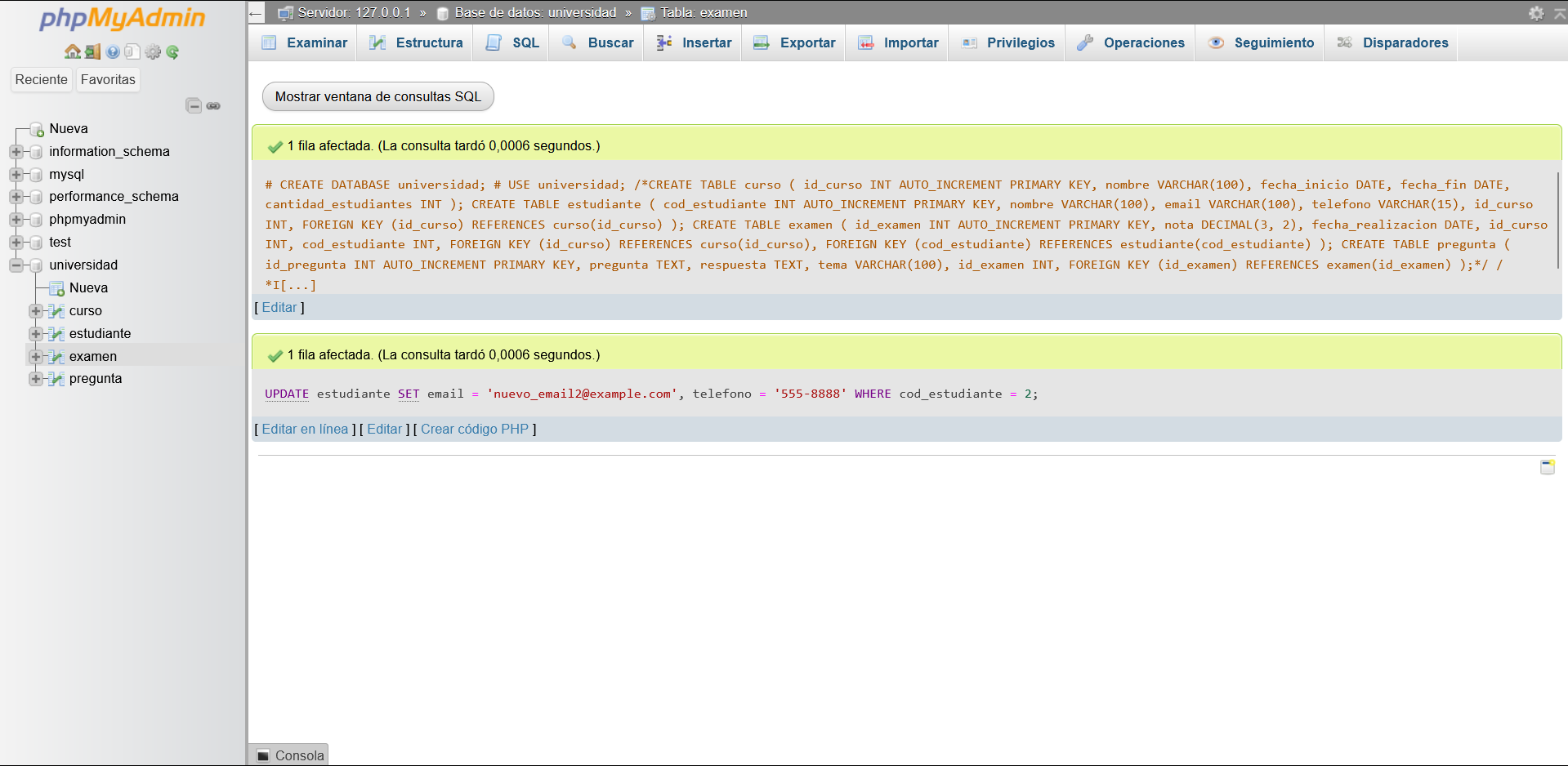
UPDATE curso SET nombre = 'Nuevo Curso 2', fecha\_inicio = '2024-07-15' WHERE id\_curso = 2;



* 1. De la tabla estudiantes, actualizar el email y teléfono de dos estudiantes

UPDATE estudiante SET email = 'nuevo\_email1@example.com', telefono = '555-9999' WHERE cod\_estudiante = 1;

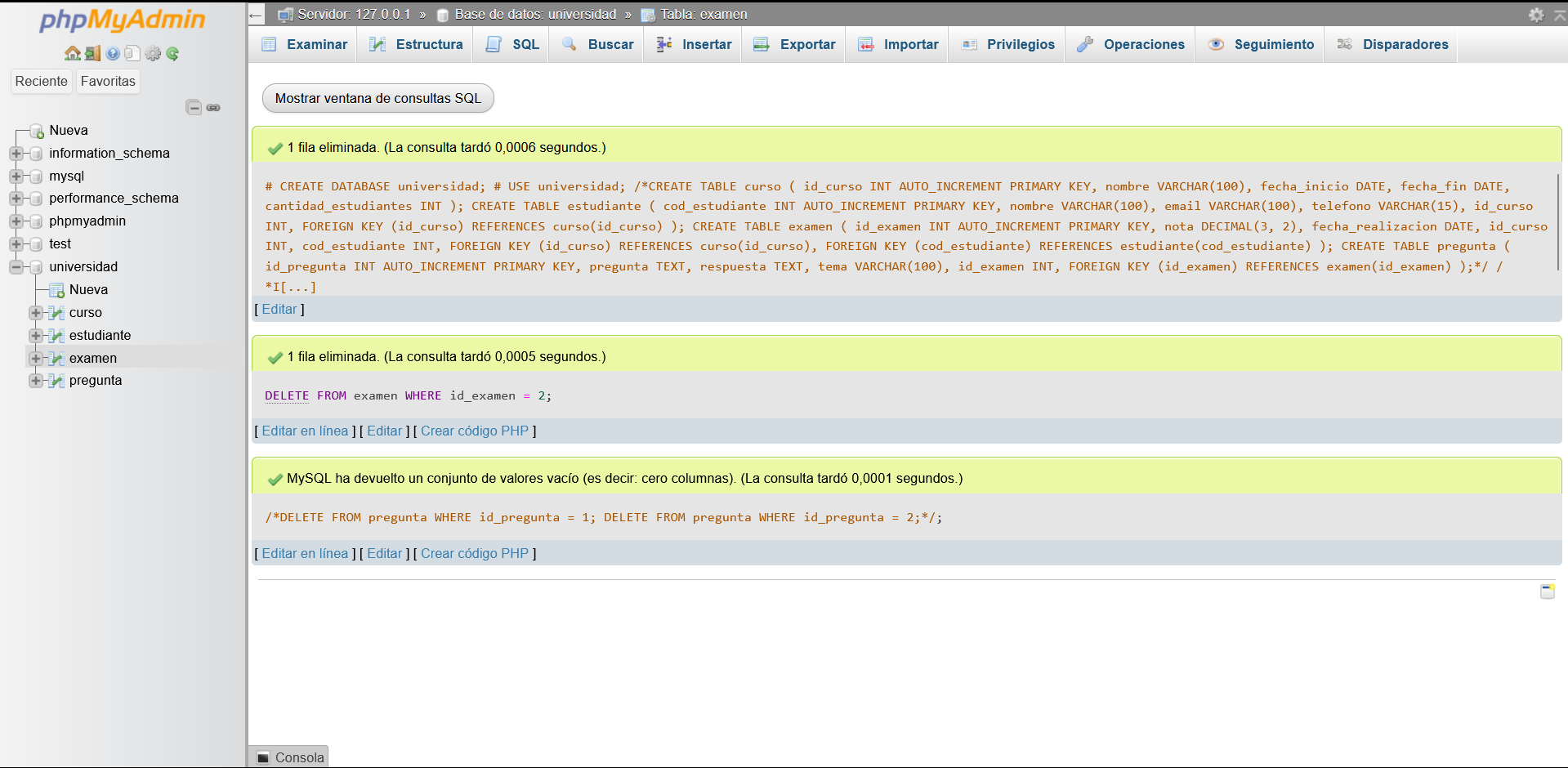
UPDATE estudiante SET email = 'nuevo\_email2@example.com', telefono = '555-8888' WHERE cod\_estudiante = 2;



1. Eliminar:
   1. Eliminar dos exámenes. Poner pantallazo del resultado

DELETE FROM examen WHERE id\_examen = 1;

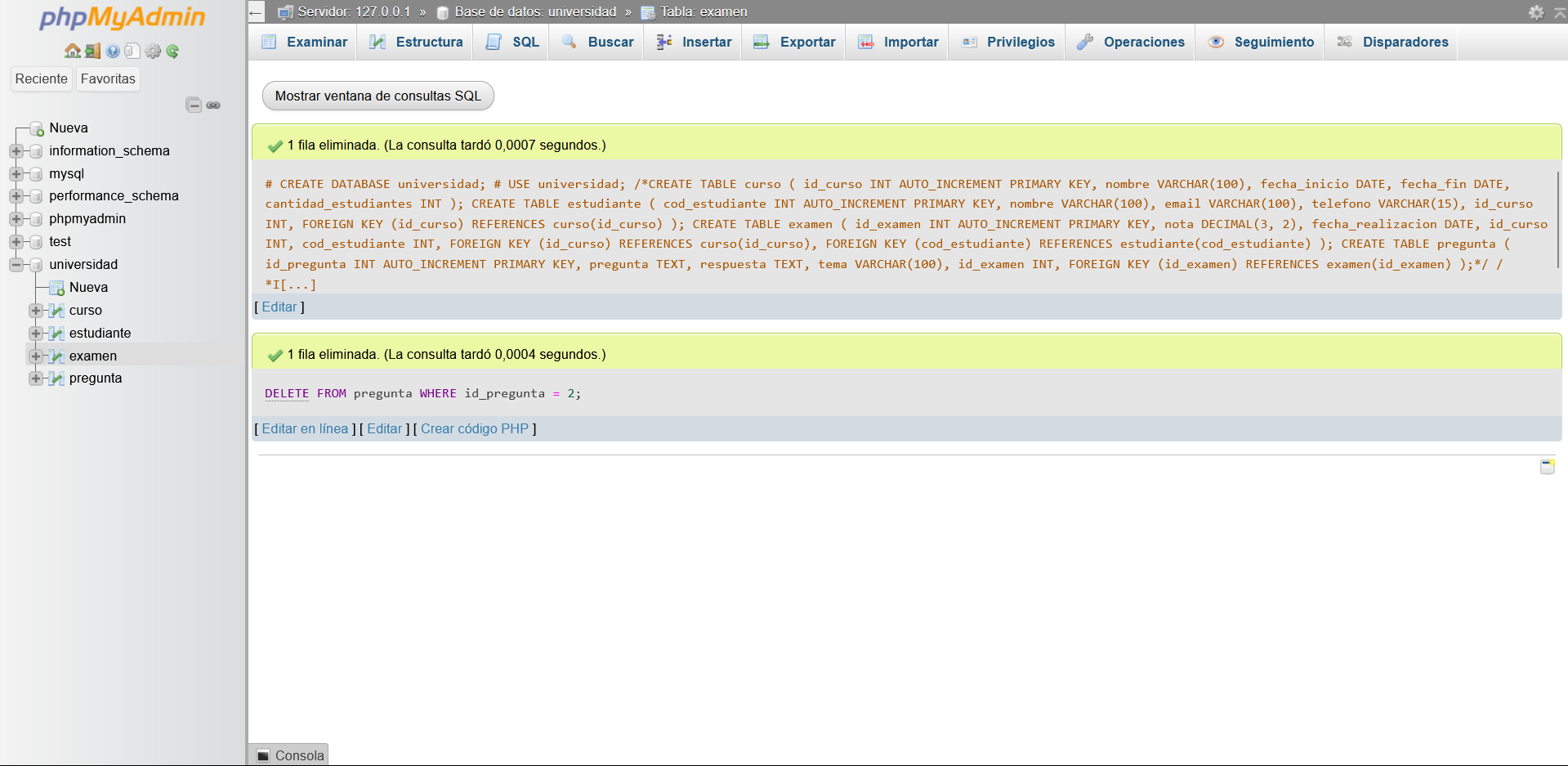
DELETE FROM examen WHERE id\_examen = 2;



* 1. Eliminar dos preguntas. Poner pantallazo del resultado

DELETE FROM pregunta WHERE id\_pregunta = 1;

DELETE FROM pregunta WHERE id\_pregunta = 2;



**Puntos adicionales taller SQL 01**

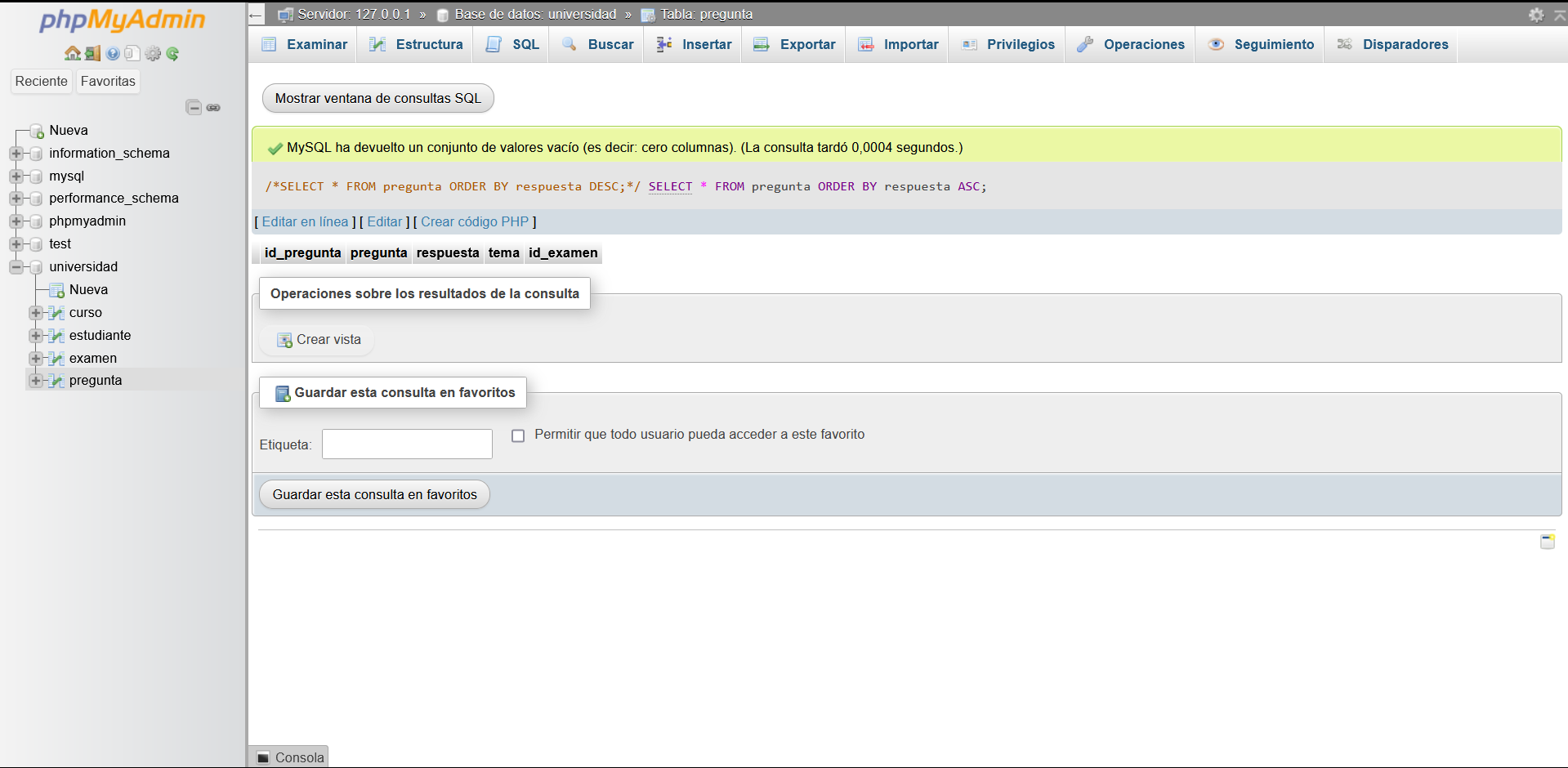
1. Realizar los siguientes puntos sobre el taller realizado de SQL de curso\_estudiante.
2. Ordenar las preguntas por el campo “respuesta” de forma descentente y ascendente (utilizando ORDER BY)

Ascendente:

SELECT \*

FROM preguntas

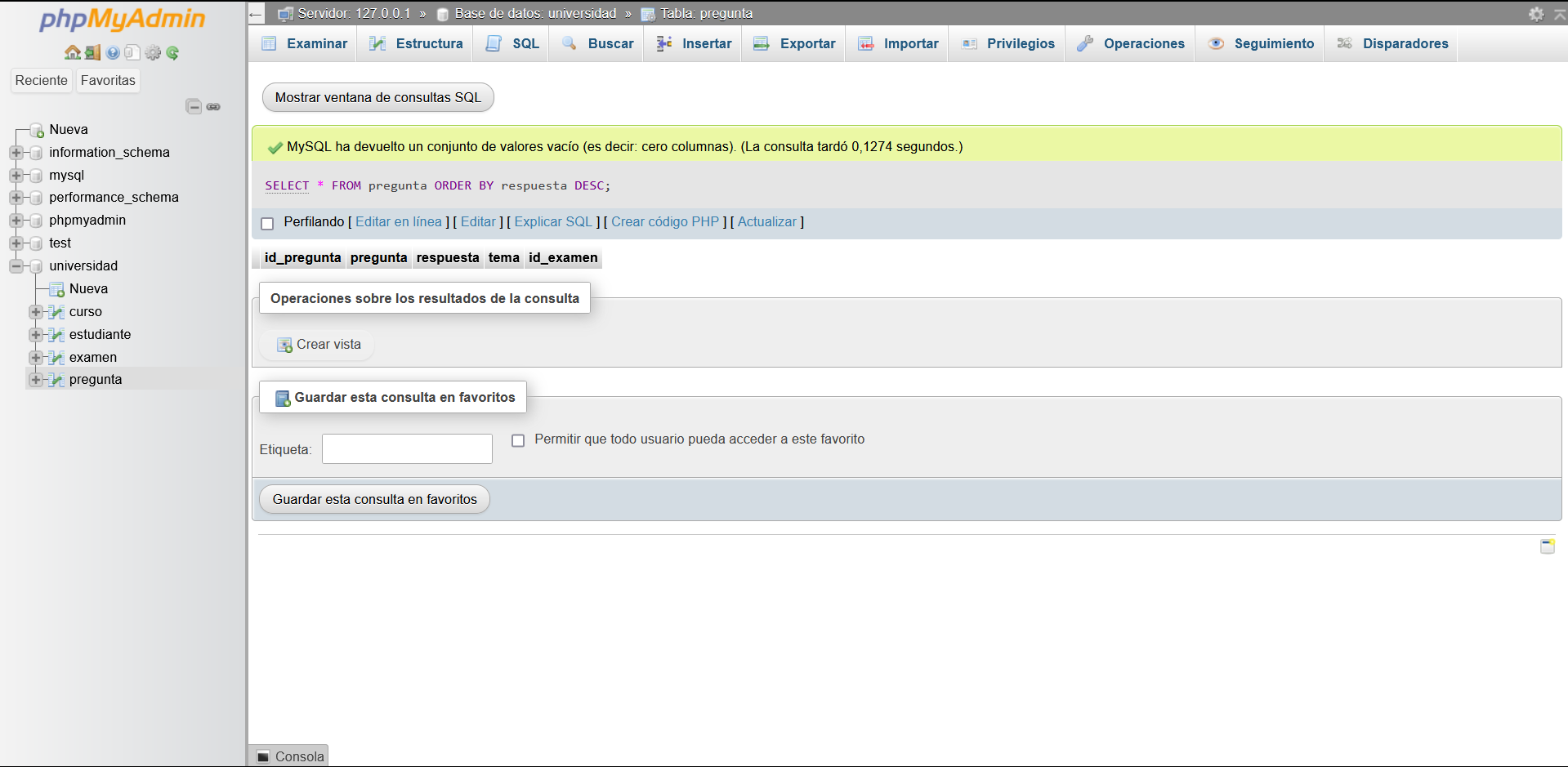
ORDER BY respuesta ASC;



Descendente:  
 SELECT \*

FROM preguntas

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, nota) VALUES ('Juan', 'juan@example.com', '1234567890', 95);

INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Ana', 'ana@example.com', '1234567891', 89);

INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Luis', 'luis@example.com', '1234567892', 92);

INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Marta', 'marta@example.com', '1234567893', 88);

INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Pedro', 'pedro@example.com', '1234567894', 85);

INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Sara', 'sara@example.com', '1234567895', 90);

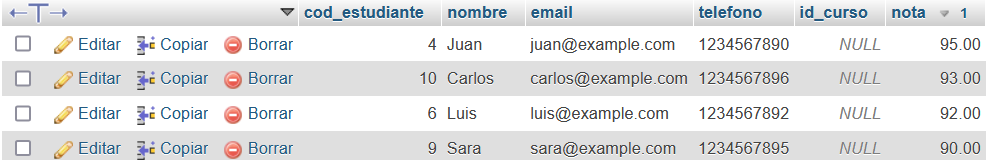
INSERT INTO estudiante (nombre, email, telefono, nota) VALUES ('Carlos', 'carlos@example.com', '1234567896', 93);

SELECT \*

FROM estudiante

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

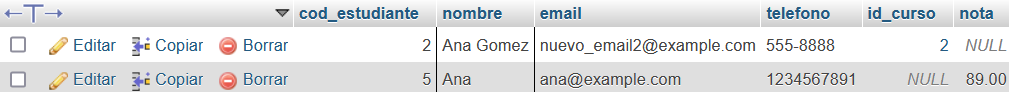
Consultar estudiantes que (utilizando LIKE):

* + Nombre empiece por la letra "A"

SELECT \*

FROM estudiante

WHERE nombre LIKE 'A%';

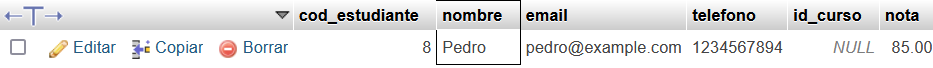


* + Nombre termine con la letra "O"

SELECT \*

FROM estudiante

WHERE nombre LIKE '%O';

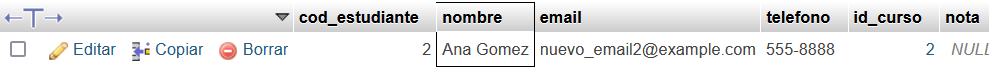


* + Nombre tenga una letra "G" en la mitad

SELECT \*

FROM estudiante

WHERE nombre LIKE '%G%';



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

* AVG()

SELECT AVG(nota) AS promedio\_nota

FROM estudiante;



* MAX()

SELECT MAX(nota) AS nota\_maxima

FROM estudiante;



* MIN()

SELECT MIN(nota) AS nota\_minima

FROM estudiante;



• SUM()

SELECT SUM(nota) AS suma\_notas

FROM estudiante;

