**Segundo parcial de base de datos**

**Parte A.** (1.5)

**Crear una base de datos para la universidad con 4 tablas, docentes, estudiantes, programas y materias, incluir los diferentes atributos (incluidos las llaves) y llenar dos instancias para cada una de ellas**

DROP DATABASE IF EXISTS universidad;

CREATE DATABASE universidad CHARACTER SET utf8mb4;

USE universidad;

CREATE TABLE programa (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(50) NOT NULL

);

CREATE TABLE estudiante (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY, nif VARCHAR(9) UNIQUE,

nombre VARCHAR(25) NOT NULL,

apellido1 VARCHAR(50) NOT NULL,

apellido2 VARCHAR(50),

ciudad VARCHAR(25) NOT NULL,

direccion VARCHAR(50) NOT NULL,

telefono VARCHAR(9),

fecha\_nacimiento DATE NOT NULL,

sexo ENUM('H', 'M') NOT NULL

);

CREATE TABLE docente (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY, nif VARCHAR(9) UNIQUE,

nombre VARCHAR(25) NOT NULL,

apellidoprim VARCHAR(50) NOT NULL,

apellidosegun VARCHAR(50),

ciudad VARCHAR(25) NOT NULL,

direccion VARCHAR(50) NOT NULL,

telefono VARCHAR(9),

fecha\_nacimiento DATE NOT NULL,

sexo ENUM('H', 'M') NOT NULL,

id\_programa INT UNSIGNED NOT NULL,

FOREIGN KEY (id\_programa) REFERENCES programa(id)

);

CREATE TABLE materia (

id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR(100) NOT NULL, creditos FLOAT UNSIGNED NOT NULL,

tipo ENUM('básica', 'obligatoria', 'optativa') NOT NULL, curso TINYINT UNSIGNED NOT NULL, cuatrimestre TINYINT UNSIGNED NOT NULL, id\_profesor INT UNSIGNED,

FOREIGN KEY(id\_profesor) REFERENCES docente(id)

);

CREATE TABLE alumno\_se\_matricula\_materia ( id\_estudiante INT UNSIGNED NOT NULL, id\_materia INT UNSIGNED NOT NULL, PRIMARY KEY (id\_estudiante, id\_materia),

FOREIGN KEY (id\_estudiante) REFERENCES estudiante(id), FOREIGN KEY (id\_materia) REFERENCES materia(id)

);

**//datos**

INSERT INTO programa VALUES (11, 'ingenieria de eletronica');

INSERT INTO programa VALUES (22, 'ingenieria sistema');

INSERT INTO estudiante VALUES (1, '79542419S', 'carlos', 'corpas', 'castilla', 'Almería', 'C/ Mercurio', '618253876', '1982/08/08', 'H');

INSERT INTO estudiante VALUES (2, '86902806M', 'jaime', 'perez', 'casadiego', 'Almería', 'C/ Real del barrio', '950254837', '1981/06/29', 'H');

INSERT INTO docente VALUES (3, '11105554G', 'david', 'Ramirez', 'Gea', 'Almería', 'C/ Marte', '618223876', '1989/08/19', 'M', 11);

INSERT INTO docente VALUES (5, '38223286T', 'David', 'luis', 'Fisher', 'agustin', 'C/ Venus', '678516294', '1998/01/7', 'H', 25);

INSERT INTO materia VALUES (1, 'Álgebra diferencial y ecuaciones', 6, 'básica', 1, 1, NULL);

INSERT INTO materia VALUES (2, 'Cálculo', 6, 'básica', 1, 1, NULL);

**Parte B**(3.5)

**Utilizando la base de datos de Jardinería Realizar las**

**siguientes consultas** [**SQL**](https://aulaweb.unicesar.edu.co/mod/resource/view.php?id=361308)

1. Muestre la suma total de todos los pagos que se realizaron para cada uno de los años que aparecen en la tabla

[select](http://127.0.0.1/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [sum](http://127.0.0.1/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/group-by-functions.html#function_sum)(total) from pago;

1. Devuelve un listado que muestre los empleados que no tienen una oficina asociada y los que no tienen un cliente asociad.

SELECT empleado.codigo\_empleado, empleado.nombre from empleado

LEFT JOIN oficina ON oficina.codigo\_oficina=empleado.codigo\_oficina

RIGHT JOIN cliente ON

cliente.codigo\_empleado\_rep\_ventas=empleado.codigo\_empleado

WHERE empleado.codigo\_oficina IS NULL AND

cliente.codigo\_empleado\_rep\_ventas IS NULL;

1. Devuelve un listado con los clientes que han realizado algún pedido, pero no han realizado ningún pago

SELECT DISTINCT cliente.codigo\_cliente, nombre\_cliente FROM cliente LEFT JOIN pago ON pago.codigo\_cliente=cliente.codigo\_cliente RIGHT JOIN pedido ON pedido.codigo\_cliente=cliente.codigo\_cliente WHERE pago.codigo\_cliente is NULL AND pedido.codigo\_cliente is NOT NULL;

1. Devuelve un listado de los productos que nunca han aparecido en un pedido.

SELECT DISTINCT detalle\_pedido.codigo\_pedido, producto.nombre FROM producto

RIGHT JOIN detalle\_pedido ON producto.codigo\_producto=detalle\_pedido.codigo\_pedido

WHERE detalle\_pedido.codigo\_pedido is NULL;

1. **Devuelve el nombre de los clientes que no hayan hecho pagos y el nombre de sus representantes junto con la ciudad de la oficina a la que pertenece el representante**.

SELECT DISTINCT cliente.nombre\_cliente, empleado.nombre As Rventas, oficina.ciudad As Ociudad FROM pago

RIGHT JOIN cliente ON cliente.codigo\_cliente=pago.codigo\_cliente

JOIN empleado ON cliente.codigo\_empleado\_rep\_ventas=empleado.codigo\_empleado JOIN oficina ON empleado.codigo\_oficina=oficina.codigo\_oficina WHERE pago.forma\_pago IS NULL;

1. **Devuelve el nombre de los clientes a los que no se les ha entregado a tiempo un pedido.**

SELECT DISTINCT cliente.codigo\_cliente, nombre\_cliente, pedido.fecha\_esperada, pedido.fecha\_entrega FROM cliente

JOIN pedido ON pedido.codigo\_cliente=cliente.codigo\_cliente

WHERE pedido.fecha\_esperada<pedido.fecha\_entrega;

1. Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de entrega de los pedidos que no han sido entregados a tiempo

SELECT codigo\_pedido, cliente.codigo\_cliente, fecha\_esperada, fecha\_entrega FROM pedido

JOIN cliente ON cliente.codigo\_cliente=pedido.codigo\_cliente WHERE pedido.fecha\_esperada<pedido.fecha\_entrega;