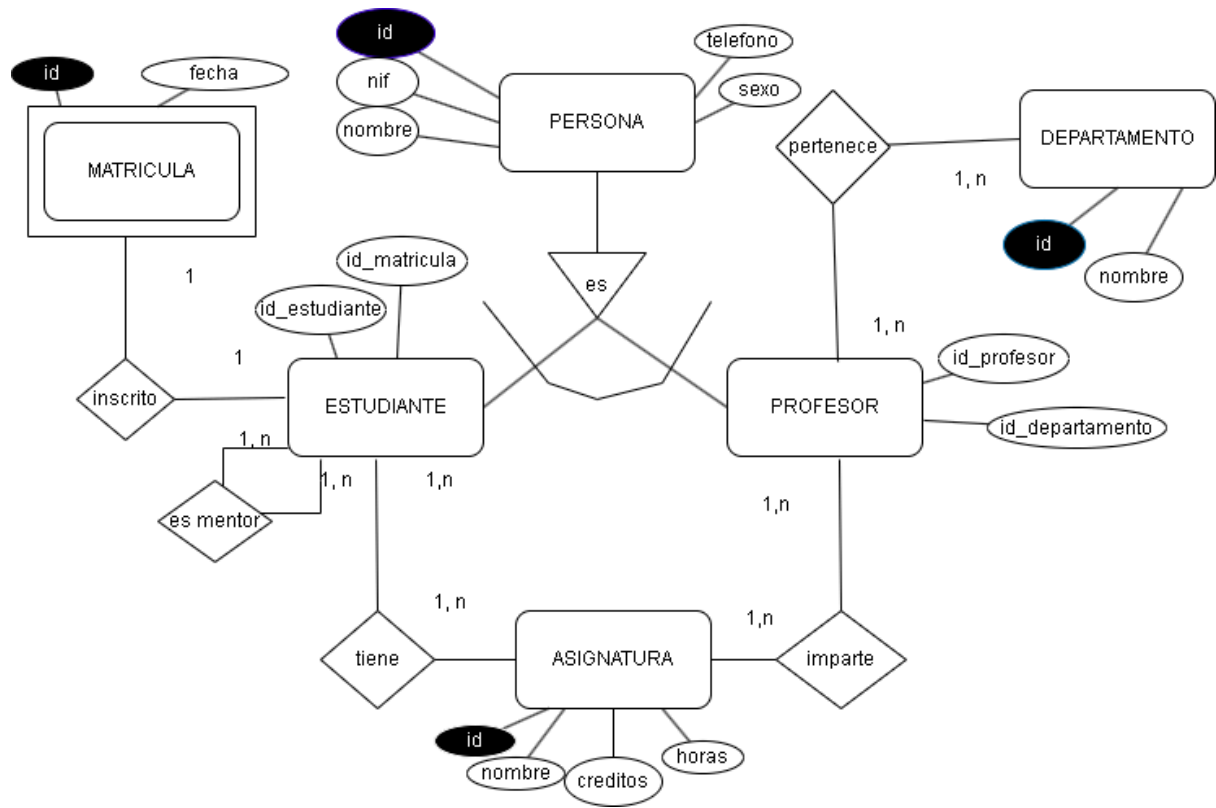
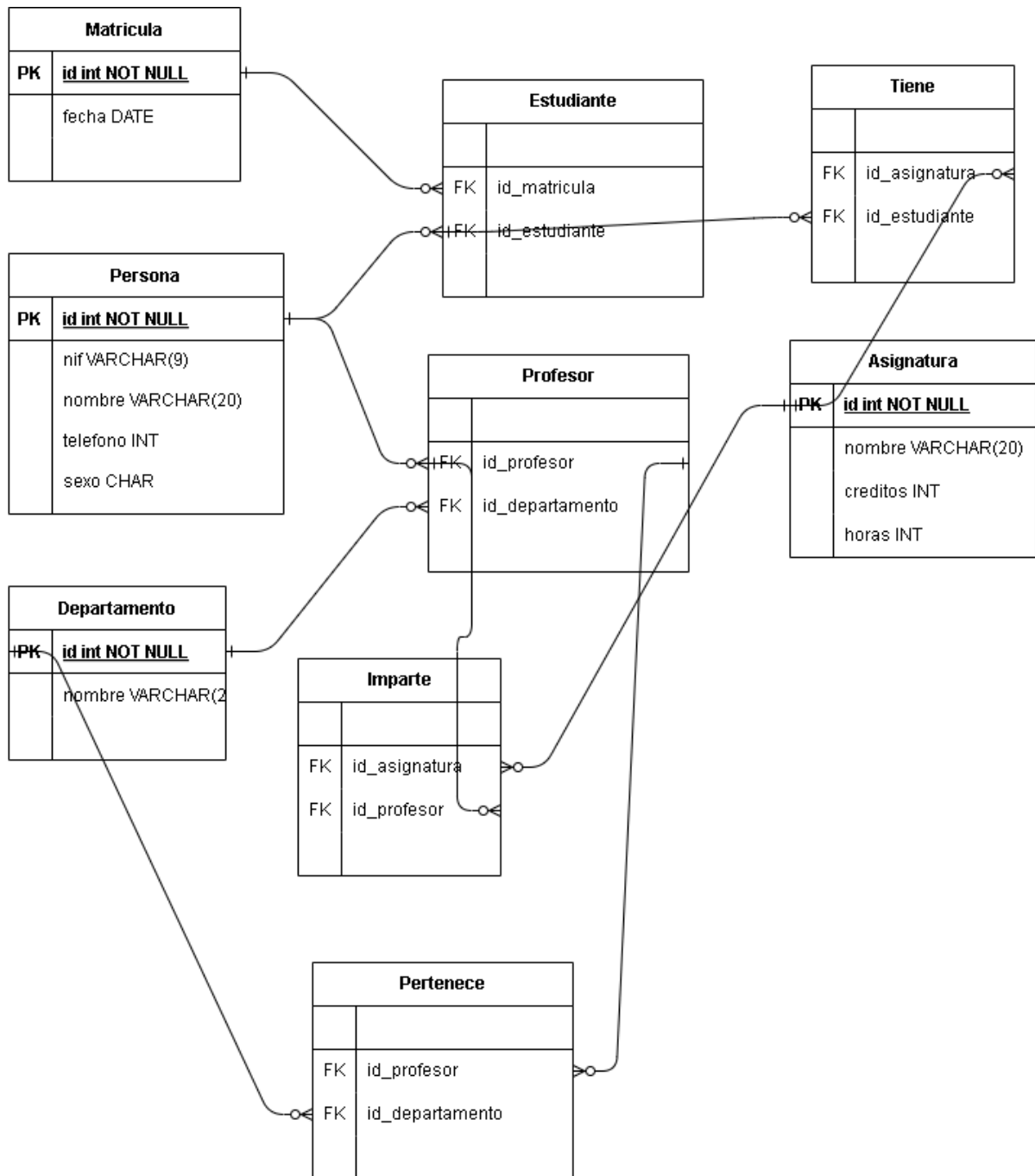


## MODELO E-R



## GRAFO RELACIONAL



## CREACIÓN DE LAS TABLAS

```
CREATE DATABASE UNIVERSIDAD;
```

```
USE UNIVERSIDAD;
```

```
CREATE TABLE PERSONA (
id INT PRIMARY KEY NOT NULL,
```

```
nif VARCHAR(9),  
nombre VARCHAR(20),  
telefono INT,  
sexo CHAR CHECK (sexo IN ('H', 'F'))  
);
```

```
CREATE TABLE MATRICULA (  
id INT PRIMARY KEY NOT NULL,  
fecha DATE  
);
```

```
CREATE TABLE DEPARTAMENTO (  
id INT PRIMARY KEY NOT NULL,  
nombre VARCHAR(20)  
);
```

```
CREATE TABLE ASIGNATURA (  
id INT PRIMARY KEY NOT NULL,  
nombre VARCHAR(20),  
creditos INT CHECK (creditos > 6),  
horas INT  
);
```

```
CREATE TABLE ESTUDIANTE (  
id_estudiante INT,  
id_matricula INT,  
FOREIGN KEY(id_estudiante) REFERENCES PERSONA(id) ON DELETE  
CASCADE,  
FOREIGN KEY(id_matricula) REFERENCES MATRICULA(id) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE PROFESOR (  
id_profesor INT,  
id_departamento INT,  
FOREIGN KEY(id_profesor) REFERENCES PERSONA(id) ON DELETE  
CASCADE,  
FOREIGN KEY(id_departamento) REFERENCES DEPARTAMENTO(id) ON  
DELETE CASCADE  
);
```

```
CREATE TABLE TIENE (  
id_estudiante INT,  
id_asignatura INT,  
FOREIGN KEY(id_estudiante) REFERENCES ESTUDIANTE(id_estudiante)  
ON DELETE CASCADE,  
FOREIGN KEY(id_asignatura) REFERENCES ASIGNATURA(id) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE IMPARTE (  
id_profesor INT,  
id_asignatura INT,  
FOREIGN KEY(id_profesor) REFERENCES PROFESOR(id_profesor) ON  
DELETE CASCADE,  
FOREIGN KEY(id_asignatura) REFERENCES ASIGNATURA(id) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE PERTENECE (  
id_departamento INT,  
id_profesor INT,  
FOREIGN KEY(id_profesor) REFERENCES PROFESOR(id_profesor) ON  
DELETE CASCADE,  
FOREIGN KEY(id_departamento) REFERENCES DEPARTAMENTO(id) ON  
DELETE CASCADE  
);
```

## INSERCIÓN DE DATOS

```
USE UNIVERSIDAD;
```

```
-- Insertar registros en la tabla PERSONA
```

```
INSERT INTO PERSONA (id, nif, nombre, telefono, sexo) VALUES
```

```
(1, '12345678A', 'Juan', 123456789, 'H'),
```

```
(2, '87654321B', 'María', 987654321, 'F'),
```

```
(3, '98765432C', 'Pedro', 654321987, 'H'),
```

```
(4, '23456789D', 'Ana', 456789123, 'F'),
```

```
(5, '54321678E', 'Carlos', 789123456, 'H');
```

```
-- Insertar registros en la tabla MATRICULA
INSERT INTO MATRICULA (id, fecha) VALUES
(1, '2024-01-15'),
(2, '2024-02-20'),
(3, '2024-03-25'),
(4, '2024-04-10'),
(5, '2024-05-05');
```

```
-- Insertar registros en la tabla DEPARTAMENTO
INSERT INTO DEPARTAMENTO (id, nombre) VALUES
(1, 'Informática'),
(2, 'Matemáticas'),
(3, 'Física'),
(4, 'Biología'),
(5, 'Química');
```

```
-- Insertar registros en la tabla ASIGNATURA
INSERT INTO ASIGNATURA (id, nombre, creditos, horas) VALUES
(1, 'Programación', 8, 120),
(2, 'Cálculo I', 10, 90),
(3, 'Física General', 7, 100),
(4, 'Biología Celular', 9, 150),
(5, 'Química Orgánica', 7, 110);
```

```
-- Insertar registros en la tabla ESTUDIANTE
INSERT INTO ESTUDIANTE (id_estudiante, id_matricula) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);
```

```
-- Insertar registros en la tabla PROFESOR
INSERT INTO PROFESOR (id_profesor, id_departamento) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);
```

-- Insertar registros en la tabla TIENE

INSERT INTO TIENE (id\_estudiante, id\_asignatura) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5);

-- Insertar registros en la tabla IMPARTE

INSERT INTO IMPARTE (id\_profesor, id\_asignatura) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5);

-- Insertar registros en la tabla PERTENECE

INSERT INTO PERTENECE (id\_departamento, id\_profesor) VALUES

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5);

## QUERYS

- Subconsulta 1: Obtener el número de estudiantes en cada departamento que hayan matriculado asignaturas de más de 8 créditos:

```
SELECT D.nombre AS departamento, COUNT(E.id_estudiante) AS num_estudiantes
FROM DEPARTAMENTO D
JOIN PROFESOR P ON D.id = P.id_departamento
JOIN IMPARTE I ON P.id_profesor = I.id_profesor
JOIN ASIGNATURA A ON I.id_asignatura = A.id
JOIN TIENE T ON A.id = T.id_asignatura
JOIN ESTUDIANTE E ON T.id_estudiante = E.id_estudiante
JOIN MATRICULA M ON E.id_matricula = M.id
WHERE A.creditos > 8
GROUP BY D.nombre;
```

- Subconsulta 2: Obtener el nombre y la cantidad de asignaturas que imparte cada profesor, excluyendo aquellos cuyo nombre comience con 'A':

```
SELECT P.id_profesor, PER.nombre AS nombre_profesor, COUNT(I.id_asignatura) AS
num_asignaturas
FROM PROFESOR P
JOIN PERSONA PER ON P.id_profesor = PER.id
JOIN IMPARTE I ON P.id_profesor = I.id_profesor
JOIN ASIGNATURA A ON I.id_asignatura = A.id
WHERE PER.nombre NOT LIKE 'A%'
GROUP BY P.id_profesor, PER.nombre;
```

- Subconsulta 3: Obtener el nombre de las asignaturas que tienen un número de horas por encima del promedio de todas las asignaturas:

```
SELECT nombre
FROM ASIGNATURA
WHERE horas > (
    SELECT AVG(horas) AS promedio_horas
    FROM ASIGNATURA
);
```

## PL/SQL

- Procedimiento para actualizar los créditos de una asignatura:

```
CREATE PROCEDURE ActualizarCreditos(IN asignatura_id INT, IN nuevos_creditos INT)
BEGIN
    UPDATE ASIGNATURA
    SET credits = nuevos_creditos
    WHERE id = asignatura_id;
END;
```

- Función para calcular el promedio de horas de las asignaturas:

```
CREATE FUNCTION CalcularPromedioHoras() RETURNS DECIMAL DETERMINISTIC
BEGIN
    DECLARE promedio DECIMAL;

    SELECT AVG(horas) INTO promedio
    FROM ASIGNATURA;

    RETURN promedio;
END;
```

- Trigger para actualizar el nombre antes de insertar un dato en la tabla persona:

```
CREATE TRIGGER ActualizarNombreDespuesDeInsercion
BEFORE INSERT ON PERSONA
FOR EACH ROW
BEGIN
    SET NEW.nombre = CONCAT(NEW.nombre, ' [Nuevo]');
END;
```