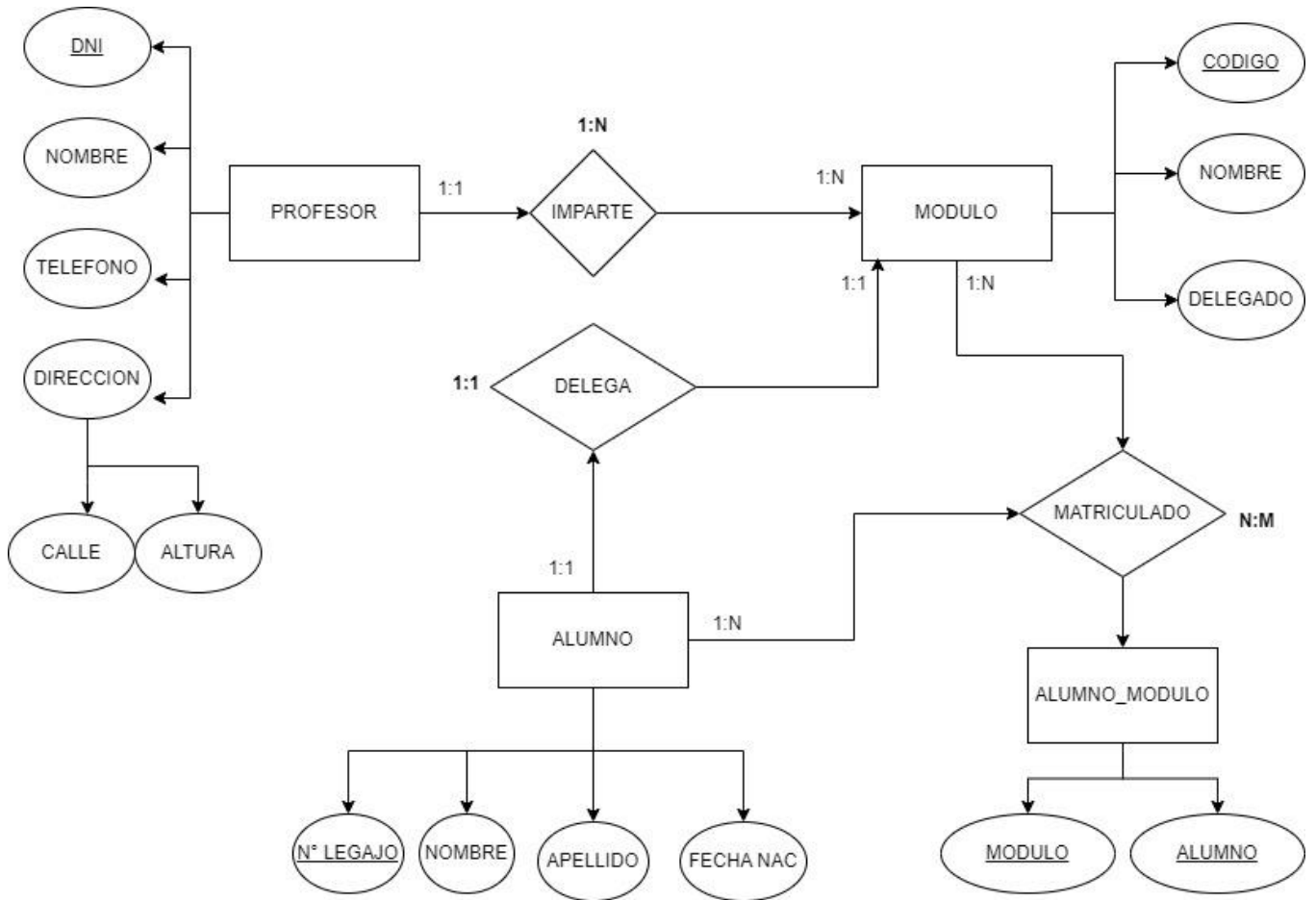


1) Diseñar el modelo entidad-relación de un Instituto.

En la base de datos se desea guardar los datos de los profesores del Instituto (DNI, nombre, dirección y teléfono). Los profesores imparten módulos, y cada módulo tiene un código y un nombre. Cada alumno está matriculado en uno o varios módulos. De cada alumno se desea guardar el N.º de legajo, nombre, apellido y fecha de nacimiento. Los profesores pueden impartir varios módulos, pero un módulo sólo puede ser impartido por un profesor. Cada curso tiene un grupo de alumnos, uno de los cuales es el delegado del grupo.



2) Aplicar las reglas de normalización

Ordenes								
Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00
2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

1FN) Nueva tabla: ORDEN_ARTICULO, con la información sobre el “artículo” que está generando redundancia (Num_art, nom_art, cant, Precio)

2FN) Nuevas tablas con columnas que no dependen de la PK: ARTICULO (nom_art, precio) y CLIENTE (Nom_cliente, Estado)

3FN) Nueva tabla para los Estados

ORDEN:

Id_orden	Fecha	Id_cliente
2301	23/02/11	101
2302	25/02/11	107
2303	27/02/11	110

ORDEN_ARTICULO:

Id_orden	Num_art	cant
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

ARTICULO:

Num_art	Nom_art	precio
3786	Red	35.00
4011	Raqueta	65.00
9132	Paq-3	4.75
5794	Paq-6	5.00
3141	Funda	10.00

ESTADO:

Id_estado	nombre_estado
1	Caracas
2	Coro
3	Maracay

CLIENTE:

Id_cliente	Nom_cliente	id_estado
101	Martin	1
107	Herman	2
110	Pedro	3

3) Tenemos el siguiente problema

- **CLIENTE** (NCliente, Nombre, Dirección, Teléfono, CP)
- **PRODUCTO** (CodProducto, Descripción, Precio)
- **VENTA** (IdVenta, CodProducto, NCliente, Cantidad)

3.1) Resolver utilizando álgebra relacional y explicar de manera coloquial el razonamiento.

3.1 a) Indicar el código y descripción de los productos cuyo precio sea inferior a los \$ 2000.

Π CodProducto, Descripción (σ Precio < 2000 (PRODUCTO))

Hago una selección (σ) de todos los productos con precio menor a 2000 de la tabla PRODUCTO, y de ese resultado usando proyección (Π) obtengo solo los atributos CodProducto y Descripción.

3.1 b) Mostrar la descripción del producto de aquellos productos que vendieron menos de 50 unidades.

Π Descripción ((σ Cantidad < 50 VENTA) \bowtie VENTA. CodProducto = PRODUCTO. CodProducto PRODUCTO)

Hago una selección (σ) de todas las ventas menores a 50 unidades, luego uno con un JOIN (\bowtie) la tablas VENTA y PRODUCTO a través del atributo "CodProducto" y finalmente hago una proyección (Π) del atributo "Descripción" de la tabla PRODUCTO.

NOTA: No aclaro de que tabla son los atributos Descripción y Cantidad porque son únicos de cada tabla.

3.1 c) Mostrar el identificador de las ventas cuya cantidad supera a la cantidad vendida en la venta número 15.

$C2 = \sigma \text{ idVenta} = 15 \text{ (Venta)}$

$\text{Venta2} = \rho \text{ idVenta2} \leftarrow \text{idVenta}, \text{CodProducto 2} \leftarrow \text{CodProducto}, \text{NCliente 2} \leftarrow \text{NCliente}, \text{Cantidad2} \leftarrow \text{Cantidad (C2)}$

$\pi \text{ idVenta } (\sigma \text{ Venta.Cantidad} > \text{Venta.Cantidad2 (Venta} \times \text{Venta2)})$

Primero hago una selección de la venta número 15 y la guardo como “C2”, luego cambio el nombre de cada una de sus columnas para poder diferenciarlas de las columnas de la otra relación. Finalmente hago otra selección a través de un producto cartesiano que me una cada una de las tuplas de la primera relación con la única (en este caso) de la segunda relación, hago la comparación de cantidades y una proyección del atributo idVenta.

3.2) Resolver utilizando SQL

3.2 a) Mostrar la descripción de los productos cuyo precio sea mayor de \$ 2000.

```
SELECT Descripción FROM PRODUCTO WHERE Precio > 2000
```

3.2 b) Mostrar los nombres de los clientes junto a su teléfono.

```
SELECT Nombre, Teléfono FROM Cliente;
```

```
SELECT CONCAT (Nombre, ' ', Teléfono) AS nombre_telefono FROM CLIENTE;
```

3.2 c) Mostrar la descripción y precio de los productos cuyo identificador de venta es igual a 65.

```
SELECT Descripción, Precio FROM PRODUCTO  
JOIN VENTA  
ON PRODUCTO. CodProducto = VENTA. CodProducto  
WHERE idVenta = 65
```