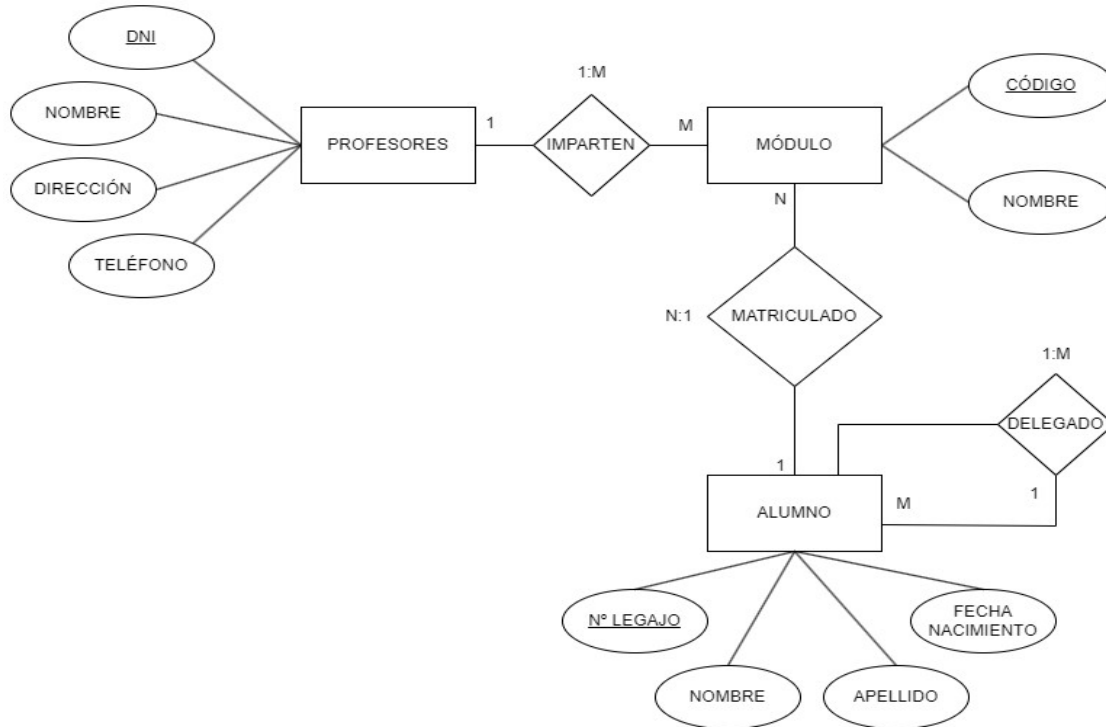


Primera Práctica Formativa Obligatoria

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 Nº 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.

Diagrama Entidad Relación Instituto



2. Aplicar las Reglas de Normalización

Primera Forma Normal

ordenes (id_orden, fecha, id_cliente, nom_cliente, estado, num_art, nom_art, cant, precio)

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	Hernan	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

La **Tabla Ordenes** **"no cumple"** con el requisito de la Primera Forma Normal (1FN), ya que tiene un grupo repetido.

El grupo repetido es: Num_art, nom_art, cant, Precio. Estos datos deben estar en una Tabla diferente, ya que la (1FN) no permite campos repetidos. Por otra parte **cumple con el requisito de tener una clave primaria y tener los campos atómicos.**

ordenes (id_orden, fecha, id_cliente, nom_cliente, estado)

Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado
2301	23/02/11	101	Martin	Caracas
2301	23/02/11	101	Martin	Caracas
2301	23/02/11	101	Martin	Caracas
2302	25/02/11	107	Hernan	Coro
2303	27/02/11	110	Pedro	Maracay
2303	27/02/11	110	Pedro	Maracay

Separamos la Tabla Ordenes.

Articulos_ordenes (id_orden, num_art, nom_art, cant, precio)

Articulos_ordenes

Id_orden	Num_art	nom_art	cant	Precio
2301	3786	Red	3	35,00
2301	4011	Raqueta	6	65,00
2301	9132	Paq-3	8	4,75
2302	5794	Paq-6	4	5,00
2303	4011	Raqueta	2	65,00
2303	3141	Funda	2	10,00

Creamos la Tabla Articulos_ordenes

Segunda Forma Normal

Primeramente tenemos que eliminar cualquier columna no llave que no dependa de la llave primaria de la Tabla.

Seguimos estos pasos:

- * Determinamos las columnas que no son llave y no dependen de la llave primaria de la tabla.
- * Eliminamos esas columnas de la tabla base.
- * Creamos una segunda Tabla con esas columnas y la columna de la PK de la cual depende.

Articulos_ordenes (id_orden, num_art, cant)

Articulos_ordenes

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

La Tabla Ordenes está en (2FN) porque todos los atributos no clave (Num_art y cant) dependen de la clave primaria (Id_orden).

Articulos (num_art, nom_art, precio)

Articulos

Num_art	nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
4011	Raqueta	65,00
3141	Funda	10,00

La Tabla Articulos está en (2FN) porque todos los atributos no clave (nom_art y Precio) dependen de la clave primaria (Num_art).

Tercera Forma Normal

La Tercera Forma Normal (3FN) nos dice que tenemos que eliminar cualquier columna no llave que sea dependiente de otra columna no llave.

Seguimos estos pasos:

- * Determinamos las columnas que son dependientes de otra columna no llave.
- * Eliminamos esas columnas de la tabla base.
- * Creamos una segunda Tabla con esas columnas y con la columna no llave de la cual son dependientes.

Articulos_ordenes (id_orden, num_art, cant)

Articulos_ordenes

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

La Tabla Ordenes está en (3FN) porque ya se encuentra en (2FN) y cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave.

Articulos (num_art, nom_art, precio)

Articulos

Num_art	nom_art	Precio
3786	Red	35,00
4011	Raqueta	65,00
9132	Paq-3	4,75
5794	Paq-6	5,00
4011	Raqueta	65,00
3141	Funda	10,00

La Tabla Articulos está en (3FN) porque ya se encuentra en (2FN) y cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave.

La Tabla Ordenes no se encuentra en (3FN), ya que **nom_cliente** y **Estado** son dependientes de **Id_cliente** y esta columna no es la llave primaria.

Para normalizar la esta Tabla, moveremos las columnas no llave y la columna llave de la cual dependen dentro de una nueva tabla clientes.

ordenes (id_orden, fecha, id_cliente)

Ordenes

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

La Tabla Articulos está en (3FN) porque ya se encuentra en **(2FN)** y cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave.

clientes (id_cliente, nom_cliente, estado)

Clientes

Id_cliente	Nom_cliente	Estado
101	Martin	Caracas
107	Hernan	Coro
110	Pedro	Maracay

La Tabla Articulos está en (3FN) porque ya se encuentra en **(2FN)** y cada atributo que no forma parte de la clave primaria no depende transitivamente de la clave.

En resumen, la Base de Datos queda de la siguiente manera:

Ordenes (id_orden, fecha, id_cliente)

Clientes (id_cliente, nom_cliente, estado)

Articulos_ordenes (id_orden, num_art, cant)

Articulos (num_art, nom_art, precio)

3. Tenemos el siguiente Problema

Tenemos el siguiente esquema relacional de base de datos:

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

La tabla de cliente almacena información sobre cada posible cliente de nuestra empresa.

En la tabla de producto almacenamos información sobre cada producto de la empresa.

La tabla de ventas relaciona a las dos anteriores utilizando el atributo CodProducto para indicar el producto que se venda, y el atributo NCliente para indicar el cliente al que vendimos el producto.

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

Primero renombramos las Tablas para facilitar su manejo:

Clientes → C
Productos → P
Ventas → V

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

$\Pi \text{ CodProducto, descripcion } (\sigma \text{ precio} < 2000 (P))$

En este caso hacemos un Select del Código del Producto y la descripción. Indicamos que esas columnas se encuentran en la Tabla Productos y le colocamos la condición que el precio sea menor a \$2000.

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

$\Pi \text{ descripcion } (\sigma \text{ cantidad} < 50 (P \bowtie V))$

En este caso hacemos un Select de la descripción del Producto. Le indicamos que esa columna la tiene que obtener del producto cartesiano natural de las tablas Productos y Ventas. Agregamos la condición que la cantidad tiene que ser menor a 50 unidades.

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

$R = \Pi \text{ cantidad } (\sigma \text{ IdVenta} = 15 (V))$

$\Pi \text{ IdVenta } (\sigma \text{ V.cantidad} > R.\text{Cantidad} (V \times R))$

Primeramente para poder comparar tenemos que crear una nueva tabla R que va a tener la columna cantidad y un solo registro, el IdVenta 15.

Ahora procedemos a hacer un Select del IdVenta. Le indicamos que esa columna la va a obtener del producto cartesiano de las Tablas Ventas y la Tabla R que acabamos de crear. Le indicamos que la cantidad de la Tabla Ventas tiene que ser mayor a la cantidad de la Tabla R.

3.2) Resolver utilizando SQL

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

```
SELECT descripcion
FROM productos
WHERE precio > 2000
```

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

```
SELECT nombre, telefono
FROM clientes
```

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

```
SELECT DISTINCT descripcion, precio
FROM productos
WHERE codproducto = (SELECT codproducto FROM ventas WHERE idventa = 65)
```

```
SELECT DISTINCT descripcion, precio
FROM productos
INNER JOIN ventas
ON producto.codProducto = venta.codProducto
WHERE idventa = 65
```