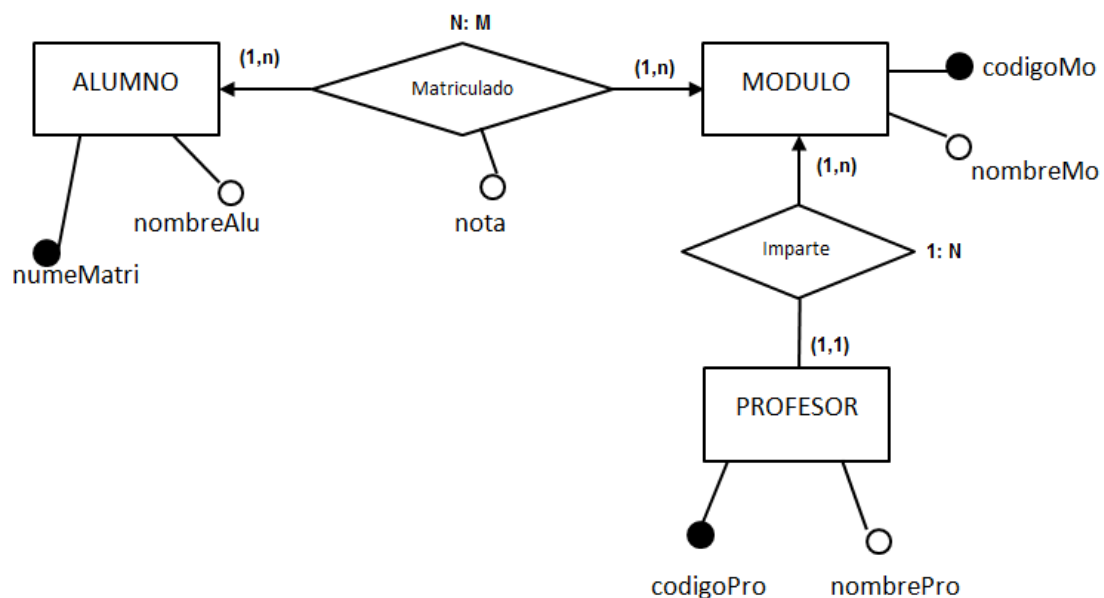


Práctica 3-3 - Solución

Consideramos un Instituto, crear una base de datos para gestionar su funcionamiento.

- Un alumno puede estar matriculado en más de un módulo y un módulo tiene matriculado varios alumnos.
- Un profesor puede dar clase en más de un módulo pero cada módulo es impartido por un único profesor.

1.- Diseñar el diagrama E-R (esquema conceptual) donde se muestren las entidades y la relación entre ellas. Incluir en cada entidad los atributos que se consideren oportunos



2.- Realizar el paso del diagrama E-R a un modelo relacional que será el que se llevará a la BD de Oracle. Indicar las tablas así como las restricciones que se consideren oportunas (claves primarias, ajenas, verificación, etc.)

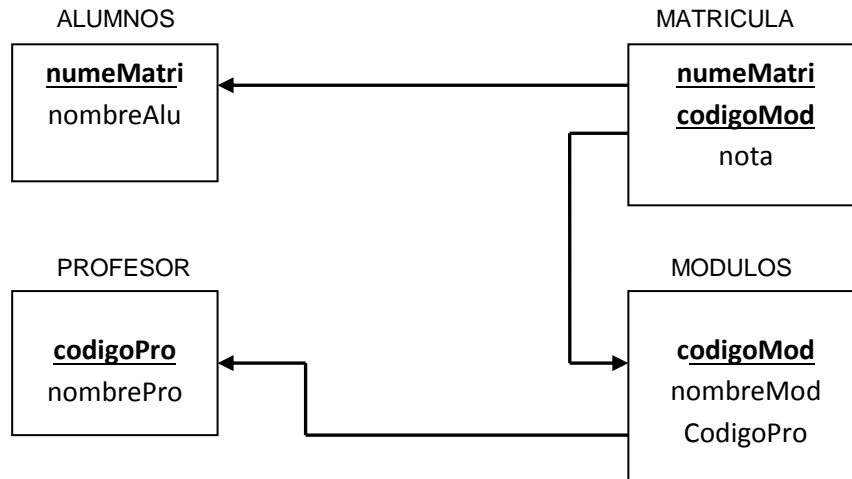
ALUMNO (numeMatri, nombreAlu)

MODULO (codigoMod, nombreMod, codigoPro)

PROFESOR (codigoPro, nombrePro)

MATRICULA (numeMatri, codigoMod, nota)

- 3.- Dibujar cada una de las tablas con sus atributos (los que se consideren necesarios), subrayando la clave primaria de cada una de ellas y enlazando las claves ajenas de cada una de las tablas (si existen) con su correspondiente clave primaria de referencia en otra tabla.



- 4.- En Oracle crear un nuevo usuario llamado **instituto** y con este usuario crear la tablas obtenidas, con claves y restricciones

Desde SYSTEM

Creamos el usuario **instituto**

Usuario: SYSTEM

Inicio

Administración de Objetos SQL Utilidades

Usuarios de Base de Datos

Crear Usuario

Usuario: SYSTEM

Inicio > Administración > Gestionar Usuarios de Base de Datos > Crear Usuario de Base de Datos

Crear Usuario de Base de Datos

Nombre de Usuario:

Contraseña:

Confirmar Contraseña:

Forzar Vencimiento de Contraseña: ☐

Estado de la Cuenta:

Tablespace por Defecto: **USERS**

Tablespace Temporal: **TEMP**

Privilegios de Usuario

Roles:

☒ CONNECT ☒ RESOURCE ☒ **DBA**

Privilegios del Sistema de Otorgamiento Directo:

☐ CREATE DATABASE LINK ☐ CREATE MATERIALIZED VIEW ☐ CREATE PROCEDURE

☐ CREATE PUBLIC SYNONYM ☐ CREATE ROLE ☐ CREATE SEQUENCE

☐ CREATE SYNONYM ☐ CREATE TABLE ☐ CREATE TRIGGER

☐ CREATE TYPE ☐ CREATE VIEW

Le asignamos privilegios DBA

Una vez creado nos desconectamos de SYSTEM y nos conectamos con el usuario INSTITUTO. Introducimos los comandos SQL para la creación de las tablas.



- Creamos las tablas sin ninguna restricción, ya las añadiremos luego.

- Creamos la Tabla **ALUMNO**

```
CREATE TABLE ALUMNO
(
    numeMatri NUMBER (4),
    nombreAlu VARCHAR2(20)
);
```

- Creamos la Tabla **PROFESOR**

```
CREATE TABLE PROFESOR
(
    codigoPro NUMBER(4),
    nombrePro VARCHAR2(20)
);
```

- Creamos la Tabla **MODULO**

```
CREATE TABLE MODULO
(
    codigoMod VARCHAR2(3),
    nombreMod VARCHAR2(20) ,
    codigoPro NUMBER(4)
);
```

- Creamos la Tabla **MATRICULA**

```
CREATE TABLE MATRICULA
(
    numeMatri NUMBER(4),
    codigoMod VARCHAR2(3),
    nota NUMBER(4, 2)
);
```

- Una vez creadas las tablas vamos a añadir las restricciones

- En la Tabla **ALUMNO**

```
ALTER TABLE ALUMNO  
ADD CONSTRAINT ALU_PK PRIMARY KEY (numeMatri);
```

- En la tabla **PROFESOR**

```
ALTER TABLE PROFESOR  
ADD CONSTRAINT PRO_PK PRIMARY KEY (codigoPro);
```

- En la Tabla **MODULO**

```
ALTER TABLE MODULO  
ADD CONSTRAINT MOD_PK PRIMARY KEY (codigoMod);
```

```
ALTER TABLE MODULO  
ADD CONSTRAINT MOD_PRO_FK1 FOREIGN KEY (codigoPro)  
REFERENCES PROFESOR (codigoPro);
```

- En la tabla Tabla MATRICULA

```
ALTER TABLE MATRICULA  
ADD CONSTRAINT MAT_PK PRIMARY KEY (numeMatri, codigoMod);
```

```
ALTER TABLE MATRICULA  
ADD CONSTRAINT MAT_ALU_FK1 FOREIGN KEY (numeMatri)  
REFERENCES ALUMNO (numeMatri);
```

```
ALTER TABLE MATRICULA  
ADD CONSTRAINT MAT_MOD_FK1 FOREIGN KEY (codigoMod)  
REFERENCES MODULO (codigoMod);
```

- 5.- Teniendo en cuenta el diseño E-R y el modelo relacional obtenido, añadir un campo llamado **repetidor** en la tabla correspondiente, teniendo en cuenta que un alumno puede repetir algún módulo y no todos, en cuyo caso solo sería repetidor en dicho módulo. Además este nuevo campo solo puede tener un valor S o N.

Desde el usuario **Instituto** añadimos el campo **repetidor** en la tabla **Matricula**.

```
ALTER TABLE MATRICULA
```

```
ADD (repetidor CHAR(1) );
```

```
ALTER TABLE MATRICULA
```

```
ADD CONSTRAINT MAT_CHK1 CHECK (repetidor IN ('S', 'N'));
```

6.- Qué pasa si borro un registro de la tabla PROFESOR que tiene asignado un módulo?

Se produce una violación de restricción de la clave ajena

7.- Probar si es posible borrar dicho profesor y el mensaje que da

No es posible ya que existe una clave ajena con la opción RESTRICTED, con lo que no puedo eliminar un registro de una tabla que tiene asociado valores en otra tabla.

8.- ¿Si quiero poder ELIMINAR/BORRAR datos de un profesor de la tabla PROFESORES, independientemente de que tenga registros asociados en la tabla MODULOS, y sin que se elimine la información del módulo que imparte dicho profesor, ¿ qué podría hacer.....?

Tendría que modificar dicha restricción de clave ajena poniéndola en vez de RESTRICTED hacerla ON DELETE SET NULL

9.- Indicar como tendría que modificar la estructura de la tabla y concretamente dicha restricción.

```
ALTER TABLE MODULOS DROP CONSTRAINT MOD_PRO_FK1;
```

```
ALTER TABLE MODULOS
```

```
ADD CONSTRAINT MOD_PRO_FK2 FOREIGN KEY (codigoPro)
```

```
REFERENCES PROFESOR (codigoPro) ON DELETE SET NULL;
```