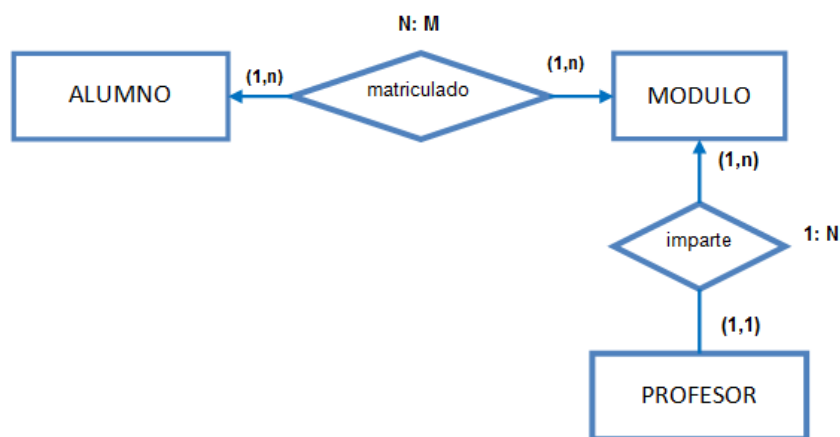


Practica 6-3-Solución

Tenemos una base de datos de un instituto.

Consideramos que un alumno puede estar matriculado en más de un módulo y un módulo tiene matriculados varios alumnos. Así mismo 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.



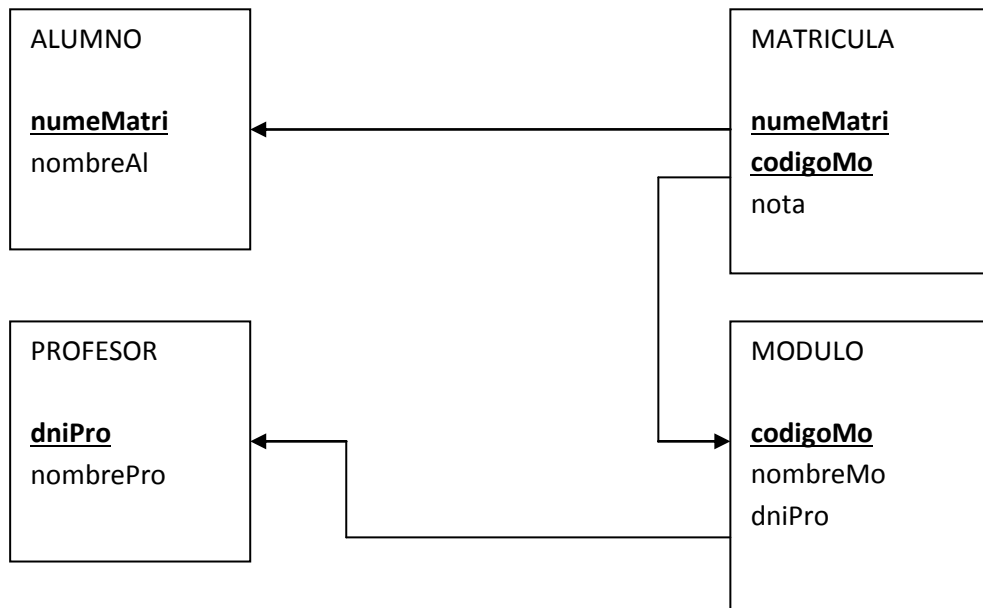
- 2.- Realizar el paso del diagrama E-R a un modelo relacional que será el que se llevará a la BD de Oracle. Indicar las tablas así como las restricciones que se consideren oportunas (claves primarias, ajenas, etc.)

ALUMNO (numeMatri, nombreAl)

MODULO (codigoMo, nombreMo, dniPro)

PROFESOR (dniPro, nombrePro)

MATRICULA (numeMatri, codigoMo, nota)



3.- Conéctate a Oracle con el usuario de **SYSTEM**

4.- Para realizar la práctica trabajar en comandos SQL

5.- Crear un usuario llamado **Usuario1** con contraseña “usuario1” sin asignar ningún rol ni privilegio. Indicar el usuario desde el que se realiza la acción, así como la instrucción para crear el usuario

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **CREATE USER usuario1 IDENTIFIED BY usuario1;**

6.- Intentar acceder con dicho usuario a la BD ¿Qué sucede? ¿Por qué no puede acceder?

Si solo se crea el usuario y no se le asigna ningún rol como el **CONNECT** o el **RESOURCE**, no podrá acceder a la BD, ya que para que un usuario pueda acceder a la BD debe tener asignado el privilegio de **CREATE SESSION**, y este debe ser asignado por el administrador de la BD al usuario bien directamente o bien a través del rol **CONNECT**

7.- Asignar al usuario **Usuario1** el rol para que se pueda conectar a la BD.

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: **GRANT CONNECT TO Usuario1;**

- 8.- Acceder a la BD con dicho usuario **Usuario1**, y comprobar si podemos crear las tablas resultantes del ejercicio planteado1. ¿Qué sucede?

Si le asignamos el rol **CONNECT** al usuario, si podremos acceder a la BD pero si intentamos crear una tabla, no nos permitirá crearla, ya que el rol **CONNECT** solo permite asignar el privilegio de conexión a la BD pero ningún más.

- 9.- ¿Cómo podemos solucionar el problema anterior? Indicar las acciones para solucionar el problema anterior y el usuario desde el que se realizan dichas acciones?

Usuario desde el que se realiza la acción: **SYSTEM**

Instrucción SQL: GRANT RESOURCE TO Usuario1;

- 10.- Crear las tablas resultantes del ejercicio planteado en el esquema del usuario **Usuario1**. Indicar las instrucciones para crear las tablas y el usuario desde el que se realizan dichas acciones

Usuario desde el que se realiza la acción: **Usuario1**

Instrucción SQL:

CREATE TABLE alumno

```
(
  numematri    NUMBER(4),
  nombreal     VARCHAR2(20)
);
```

CREATE TABLE modulo

```
(
  codigomo     VARCHAR2(3),
  nombremo     VARCHAR2(20),
  dnipro       NUMBER(4)
);
```

CREATE TABLE profesor

```
(
  dnipro       NUMBER(4),
  nombrepro    VARCHAR2(20)
);
```

CREATE TABLE matricula

```
(
  numematri    NUMBER(4),
  codigomo     VARCHAR2(3),
  nota         NUMBER(4,2)
);
```

11.- Indicar las instrucciones para crear las restricciones que no se han añadido a las tablas en el paso anterior.

Usuario desde el que se realiza la acción: **usuario1**

Instrucción SQL:

Recordar que para poder crear claves ajenas, es necesario tener creadas previamente la correspondiente tabla con la clave primaria a la que va a referenciar la clave ajena

Para la tabla ALUMNO

ALTER TABLE alumno

ADD CONSTRAINT alumno_pk **PRIMARY KEY** (numematri);

Para la tabla PROFESOR

ALTER TABLE profesor

ADD CONSTRAINT profesor_pk **PRIMARY KEY** (dnipro);

Para la tabla MODULO

ALTER TABLE modulo

ADD CONSTRAINT modulo_pk **PRIMARY KEY** (codigomo);

ALTER TABLE modulo

ADD CONSTRAINT moduloprofe_fk1 **FOREIGN KEY** (dnipro)

REFERENCES profesor (dnipro);

Para tabla MATRICULA

ALTER TABLE matricula

ADD CONSTRAINT matricula_pk **PRIMARY KEY** (numematri, codigomo);

ALTER TABLE matricula

ADD CONSTRAINT matricula_al_fk1 **FOREIGN KEY** (numematri)

REFERENCES alumno (numematri);

ALTER TABLE matricula

ADD CONSTRAINT matricula_mo_fk2 **FOREIGN KEY** (codigomo)

REFERENCES modulo (codigomo);

- 12.- Teniendo en cuenta el diseño E-R y el modelo relacional obtenido, añadir un campo llamado REPETIDOR a 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.

Usuario desde el que se realiza la acción: **usuario1**

Instrucción SQL:

ALTER TABLE matricula

ADD (repetidor CHAR(1));

ALTER TABLE matricula

ADD CONSTRAINT matricula_chk1 **CHECK** (repetidor **IN** ('S', 'N'));