

IMPLEMENTANDO NUESTRAS BD

Unidad 2 – Practica 1

EJERCICIO 12

DIAGRAMA E/R2

MODELO RELACIONAL4

CREACIÓN DE TABLAS5

EJERCICIO 27

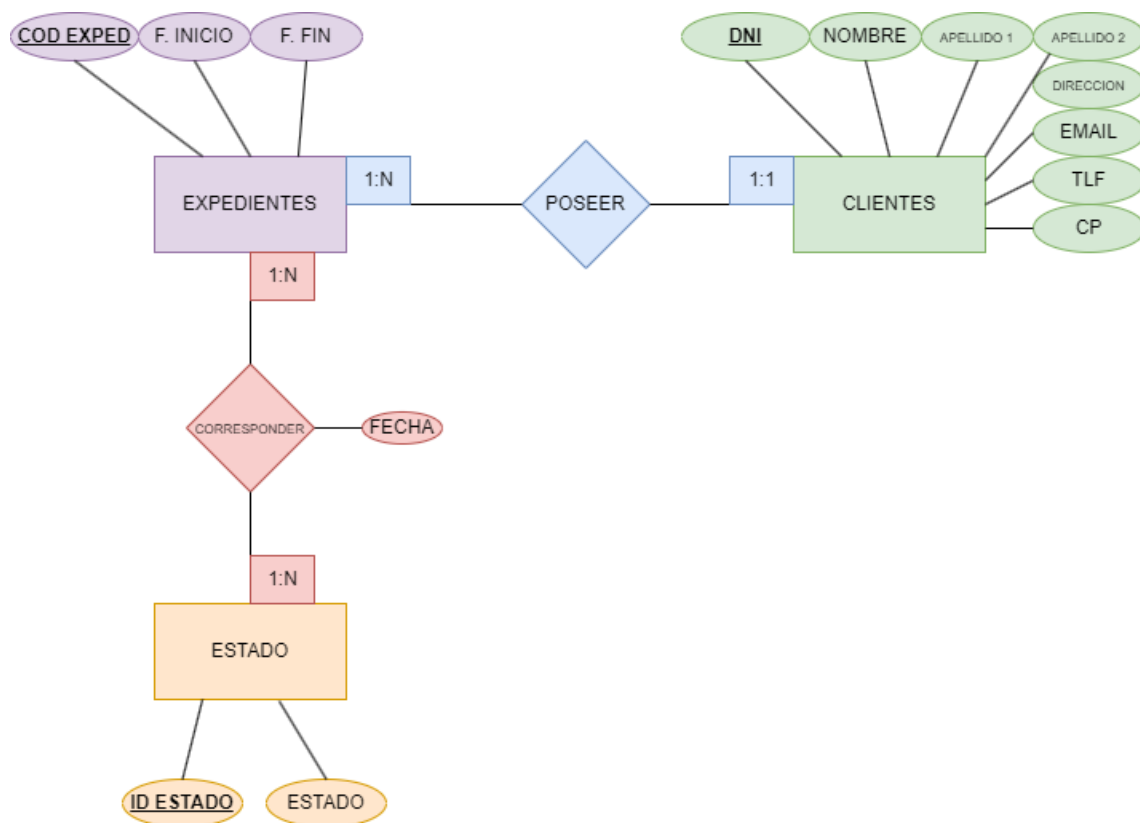
DIAGRAMA E/R7

MODELO RELACIONAL9

CONSTRUCCIÓN DE TABLAS..... 10

EJERCICIO 1

DIAGRAMA E/R



Este sería el diagrama que para mí representa el resultado del enunciado.

Por un lado tenemos la relación entre las entidades CLIENTE/EXPEDIENTE mediante el verbo POSEER, con una cardinalidad de 1:1 (un expediente es poseído por un solo cliente) y de 1:N del lado del CLIENTE (un cliente puede poseer 1 o varios EXPEDIENTES).

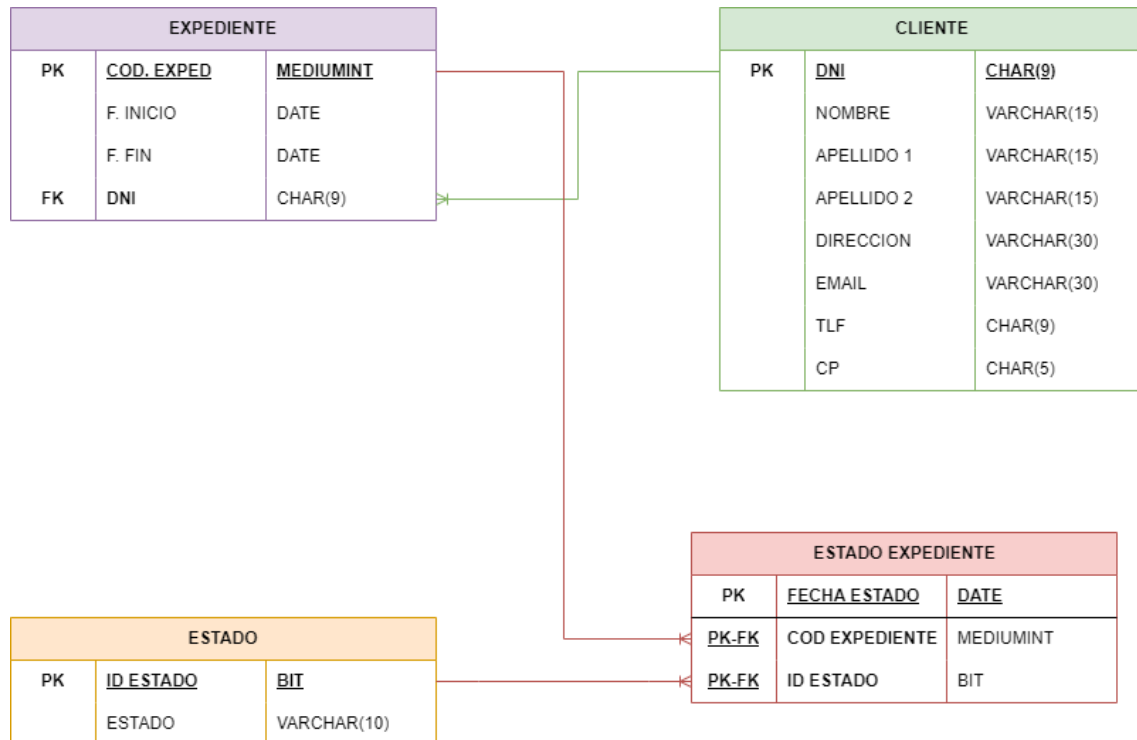
A estas dos entidades les he añadido los atributos correspondientes, indicando su PK en negrita y subrayada.

Por otro lado, tenemos la relación que se produce entre las entidades EXPEDIENTE/ESTADO mediante el verbo corresponder. A cada EXPEDIENTE le corresponden 1 o varios ESTADOS, y cada ESTADO puede corresponder a 1 o varios EXPEDIENTES.

Como atributo de la relación, siguiendo el enunciado del ejercicio, he creído que debía incluir la fecha en la que un EXPEDIENTE adquiere un ESTADO.

La cardinalidad de la relación es 1:N en ambos sentidos.

MODELO RELACIONAL



Para pasar el diagrama E/R de este ejemplo, he decidido resolverlo de esta manera.

La PK de un CLIENTE pasa como FK a la tabla de EXPEDIENTE para así relacionar cada expediente con su cliente correspondiente.

Para resolver la relación entre expediente y estado, ya que es una relación 1:N en ambos sentidos, debemos de crear una tabla intermedia.

A esta tabla intermedia le asigno una PK de la siguiente manera:

La PK de la tabla expediente pasa como FK a esta tabla.

La PK de la tabla estado pasa como FK a esta tabla.

El atributo de esta relación, junto con las dos FKs, forman una PK compuesta.

He elegido que la PK la formen los 3 campos ya que he considerado que quizá por alguna razón un expediente se pudiese volver a reabrir y quizá puede recibir varias veces el mismo estado, pudiendo obtener el mismo varias veces, por lo tanto, para que la PK sea siempre única, los tres atributos deben de formarla.

Los tipos de datos son bastante estándar. La mayoría varchar o char con una longitud específica. En algunos casos he usado el tipo bit en datos que creo que son de tipo numérico pero no van a alcanzar grandes valores. Por último, dos datos de tipo date ya que necesito la fecha pero sin la hora.

CREACIÓN DE TABLAS

Lo primero sería crear una DATABASE

CREATE OR REPLACE DATABASE bufete

Después vamos a ir creando las tablas correspondientes a cada una de las entidades identificadas en el enunciado

La primera que tengo que crear es CLIENTE ya que no recibe ninguna FK

```
CREATE TABLE cliente{  
    dni CHAR(9) NOT NULL,  
    nombre VARCHAR(15) NOT NULL,  
    apellido1 VARCHAR(15) NOT NULL,  
    apellido2 VARCHAR(15) NOT NULL,  
    direccion VARCHAR(30) NOT NULL,  
    email VARCHAR(15),  
    teléfono CHAR(9) NOT NULL,  
    cp CHAR(5) NOT NULL,  
    CONSTRAINT dni PRIMARY KEY(dni)  
  
};
```

Una vez creada la tabla CLIENTE puedo crear la tabla EXPEDIENTE que recibe un FK.

```
CREATE TABLE expediente{  
    cod_exped MEDIUMINT [UNSIGNED] AUTO_INCREMENT NOT NULL,  
    F. inicio DATE(YYYY-MM-DD),  
    F. fin DATE(YYYY-MM-DD),  
    dni CHAR(9) NOT NULL,  
    CONSTRAINT cod_exped PRIMARY KEY(cod_exped)  
    CONSTRAINT dni FOREIGN KEY (dni) REFERENCES cliente (dni)  
};
```

Después de estas dos puedo crear la tabla ESTADO que es más sencilla

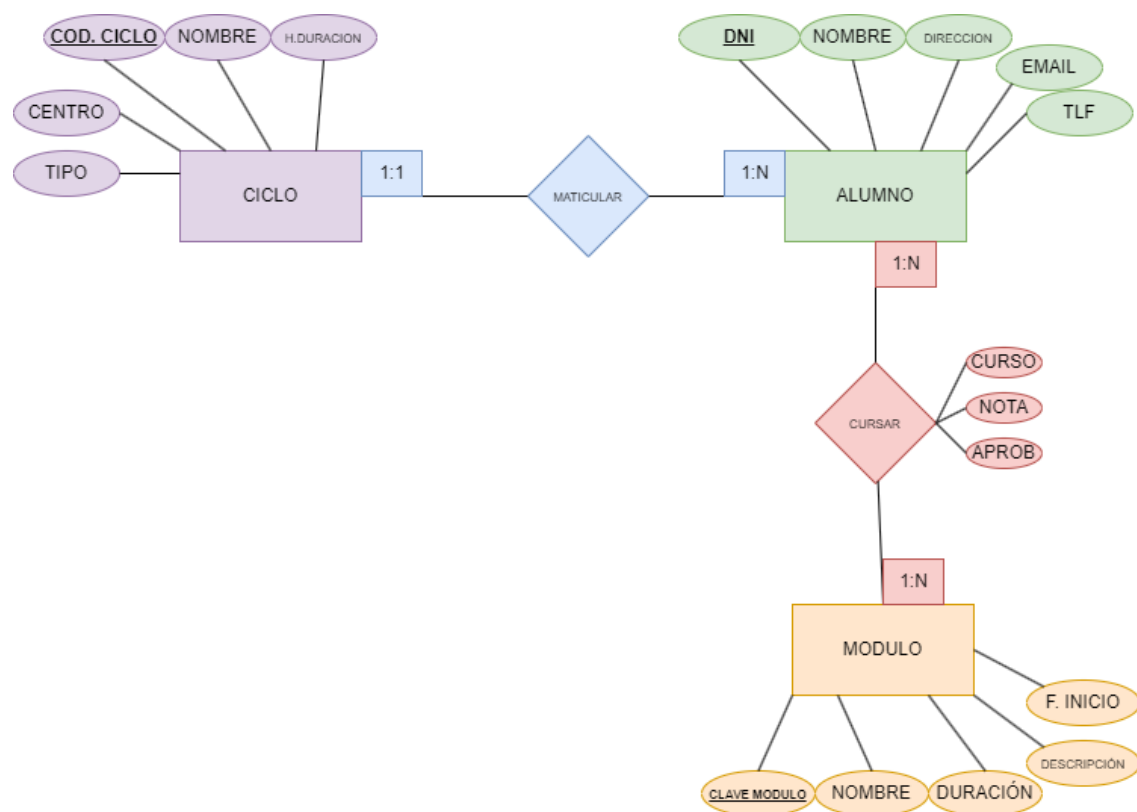
```
CREATE TABLE estado{  
    Id_estado BIT() AUTO_INCREMENT NOT NULL,  
    estado VARCHAR(10),  
    CONSTRAINT id_estado PRIMARY KEY(id_estado)  
};
```

Una vez las tengo creadas todas, puedo pasar a crear la tabla intermedia.

```
CREATE TABLA estado_expediente{  
    fecha_estado DATE(YYYY-MM-DD) NOT NULL,  
    cod_expediente MEDIUMINT [UNSIGNED],  
    Id_estado BIT(),  
    CONSTRAINT est_exp_pk PRIMARY KEY (fecha_estado, cod_expediente,  
id_estado),  
    CONSTRAINT cod_expediente FOREIGN KEY (cod_expediente) REFERENCES  
expediente (cod_expediente),  
    CONSTRAINT id_estado FOREIGN KEY (id_estado) REFERENCES estado  
(id_estado)  
};
```

EJERCICIO 2

DIAGRAMA E/R



En este caso, como creación propia, he decidido basarme en los estudios de la siguiente manera.

Un estudiante puede matricularse en 1 curso, y solo 1 a la vez. En un curso pueden matricularse varios alumnos.

Un alumno puede cursar varios módulos (no tendría porque estar matriculado en todos o tener que matricularse en todos, véase mi caso) y un módulo puede ser cursado por varios alumnos.

De un ciclo necesitamos saber: el código de ciclo, el nombre del mismo, las horas de duración, el centro en el que se estudia y el tipo (ya sea FP1, FP2, o algún otro tipo).

De un alumno queremos registrar: el dni, el nombre completo, la dirección, el email y el teléfono.

De un módulo: la clave del módulo, el nombre, la duración del módulo, la descripción y por último la fecha de inicio.

Como resultado de que un alumno curse un módulo se generará una nota, si ha aprobado o no y el curso en el que obtuvo esos resultados.

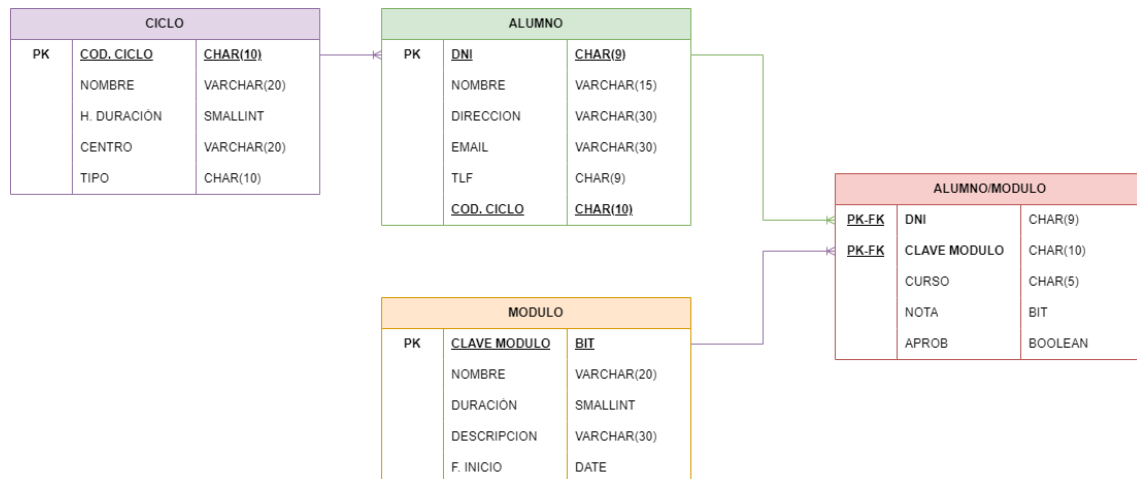
Explicando mi diagrama diré, como resumen, que:

CICLO Y ALUMNO se relacionan mediante el verbo MATRICULAR y que, como ya he explicado en enunciado, un alumno sólo se puede matricular en 1 ciclo al mismo tiempo, pero que en un ciclo se pueden matricular varios alumnos. De ahí las cardinalidades que muestro en el diagrama.

ALUMNO Y MODULO se relacionan con cardinalidades 1:N en ambos sentidos mediante el verbo CURSAR. 1 ALUMNO puede cursar varios MÓDULOS y viceversa, un MÓDULO puede ser cursado por varios ALUMNOS.

De esta última relación surgen los atributos derivados de ella.

MODELO RELACIONAL



La resolución del anterior diagrama E/R da como resultado este modelo relacional.

Para explicarlo diré que:

Como CICLO y ALUMNO tienen de cardinalidad 1:1 para alumno y 1:N para ciclo, la PK de ciclo debe de pasar como FK a la tabla alumno.

Después, para resolver la relación entre alumno y módulo debo de crear una tabla intermedia, dadas las cardinalidades de 1:N de su relación.

Para ello, las PKs de las respectivas tablas pasarán como FKs a la tabla intermedia, y formarán una PK compuesta junto con curso, así obtendremos una PK única sea cual sea el caso.

Por ejemplo, si decimos que el alumno con DNI 13456789H que estudia el módulo FG145A en el curso 24/25, lo normal es que esa clave no se repita nunca, incluso en el caso de que el alumno repita el curso y tenga que volver a cursar el módulo, ya que el valor de curso siempre será distinto.

Tipos de datos elegidos:

En la gran mayoría de datos, los que creí más largos opté por VARCHAR(n), con diferentes longitudes.



Aquellos de los que siempre se conoce su longitud fija, CHAR(n), ya que, en principio, su longitud no varía.

En cuanto a los campos Duración y Horas Duración los puse de tipo SMALLINT ya que considero que no van a ser números demasiado grandes y que quizá se pueda realizar alguna operación con ellos. De igual forma, el campo nota, es de tipo BIT, un número pequeño pero que quizá se pueda calcular algo, como una media de las notas.

Un dato de tipo BOOLEAN para indicar si ha aprobado o no un módulo y también del tipo DATE para el único campo Fecha.

CONSTRUCCIÓN DE TABLAS

Lo primero sería crear una DATABASE

CREATE OR REPLACE DATABASE curso_escolar;

Después vamos a ir creando las tablas correspondientes a cada una de las entidades identificadas en el enunciado

La primera que tengo que crear es CICLO ya que no recibe ninguna FK

```
CREATE TABLE ciclo{  
    cod_ciclo CHAR(10) NOT NULL,  
    nombre VARCHAR(20) NOT NULL,  
    h_duracion SMALLINT() UNSIGNED,  
    centro VARCHAR(20) NOT NULL,  
    tipo CHAR(10) NOT NULL,  
    CONSTRAINT cod_ciclo PRIMARY KEY(cod_ciclo)  
};
```

Una vez creada la tabla CICLO puedo crear la tabla ALUMNO que recibe un FK.

```
CREATE TABLE alumno{  
    dni CHAR(9) NOT NULL,  
    nombre VARCHAR(15) NOT NULL,  
    direccion VARCHAR(30) NOT NULL,  
    email VARCHAR(30),  
    teléfono CHAR(9) NOT NULL,  
    cod_ciclo CHAR(10) NOT NULL,  
    CONSTRAINT dni PRIMARY KEY(dni),  
    CONSTRAINT cod_ciclo FOREIGN KEY (cod_ciclo) REFERENCES ciclo  
(cod_ciclo)  
};
```

Después puedo crear la tabla MODULO que es la última que voy a crear antes de crear la tabla intermedia y necesito esta.

```
CREATE TABLE modulo{  
    clave_modulo BIT() NOT NULL,  
    nombre VARCHAR(20) NOT NULL,  
    duracion SMALLINT UNSIGNED NOT NULL,  
    descripcion VARCHAR(30) NOT NULL,  
    f_inicio DATE(YYYY-MM-DD) NOT NULL,  
    CONSTRAINT clave módulo PRIMARY KEY (clave modulo)  
};
```

Una vez las tengo creadas todas, puedo pasar a crear la tabla intermedia.

```
CREATE TABLA alumno_modulo{  
    dni CHAR(9) NOT NULL,  
    clave_modulo BIT() NOT NULL,  
  
    curso CHAR(5) NOT NULL,  
    nota BIT() NOT NULL,  
    aprobado BOOLEAN() NOT NULL,  
    CONSTRAINT módulos_alumno_pk PRIMARY KEY (dni, clave_modulo,  
curso),  
  
    CONSTRAINT dni FOREIGN KEY (dni) REFERENCES alumno (dni),  
    CONSTRAINT clave_modulo FOREIGN KEY (clave_modulo) REFERENCES  
modulo (clave_modulo)  
};
```