

Solución-Práctica 3-1:

Vamos a crear las tablas para una Academia donde se imparten distintos cursos de informática. Empezaremos creando con SQL las siguientes tablas:

- Tabla **ALUMNOS**:

ALUMNOS (nombreAlu, apellido1Alu, apellido2Alu, nifAlu, direccionAlu, sexoAlu, fechaNacAlu, cursoAlu)

- Tabla **CURSOS**:

CURSOS (nombreCur, codigoCur, nifProfe, maxAlu, fechalni, fechaFin, numHoTo).

- Tabla **PROFESORES**:

PROFESORES (nombrePro, apellido1Pro, apellido2Pro, nifPro, direPro, tituloPro, salario.)

- Restricciones:

- ✓ El alumno/a debe matricularse en un curso antes de que se le pueda dar de alta.

Significa que **cursoAlu** de la tabla **ALUMNOS** será **clave ajena** en esta tabla y es el mismo campo que **codigoCur** de la tabla **CURSOS**.

- ✓ En un curso, el número de horas es un dato que no puede faltar, es obligatorio que contenga información.

Significa que hay que poner una restricción NOT NULL al campo **numHoTo** de la tabla **CURSO**

- ✓ En la tabla **PROFESORES**, el atributo Salario no puede estar vacío.

Significa que hay que poner una restricción NOT NULL en el campo **salario** de la tabla **PROFESORES**

- ✓ Dos cursos no pueden llamarse de la misma forma.

Significa que el campo **nombreCur** nombre del curso será **UNIQUE**.

- ✓ Dos profesores no pueden llamarse igual.

Significa que el campo **nombrePro** será **UNIQUE**.

- ✓ Podremos diferenciar las tuplas de la tabla CURSOS por el Código del Curso.

Luego el campo **codigoCur** de la tabla CURSOS será clave primaria.

- ✓ Podremos diferenciar las tuplas de la tabla PROFESORES y ALUMNOS por el NIF.

Por tanto, el campo **nifPro** de la tabla PROFESORES y el campo **nifAlu** de la tabla ALUMNOS será también clave primaria en sus respectivas tablas.

- ✓ La fecha de comienzo del curso nunca puede ser mayor que la fecha de finalización.

Significa que tenemos que poner una restricción de validación en este campo.

- ✓ El dominio del atributo sexo es M (mujer) y H (hombre).

Pondremos una restricción para que en campo **sexoAlu** únicamente pueda tomar estos dos valores.

- ✓ Se debe cumplir la regla de integridad referencial de borrado en cascada.

Para que se cumpla la integridad referencial podemos poner en las restricciones de claves ajenas **ON DELETE CASCADE** después de **REFERENCES**.

■ La **creación de las tablas** nos quedaría:

```
CREATE TABLE PROFESORES1
(
    nombrePro      VARCHAR2(10) UNIQUE,
    apellido1Pro   VARCHAR2(15),
    apellido2Pro   VARCHAR2(15),
    nifPro         VARCHAR2(9),
    direPro        VARCHAR2(30),
    tituloPro      VARCHAR2(30),
    salario        NUMBER(6,2) NOT NULL,
    CONSTRAINT pro_nif_PK PRIMARY KEY(nifPro)
);
```

```
CREATE TABLE CURSOS1
```

```
(  
    nombreCur      VARCHAR2(10) UNIQUE,  
    codigoCur     NUMBER(5) PRIMARY KEY,  
    nifProfe       VARCHAR2(9),  
    maxAlu        NUMBER(3),  
    fechalni      DATE,  
    fechaFin      DATE,  
    numHoTo       NUMBER(3) NOT NULL,  
    CONSTRAINT cur_nif_FK FOREIGN KEY (nifProfe)  
        REFERENCES PROFESORES (nifPro) ON DELETE CASCADE,  
    CONSTRAINT cur_fec_CK CHECK (fechaFin > fechalni)  
);
```

```
CREATE TABLE ALUMNOS 1
```

```
(  
    nombreAlu      VARCHAR2(10),  
    apellido1Alu   VARCHAR2(15),  
    apellido2Alu   VARCHAR2(15),  
    nifAlu         VARCHAR2(9) PRIMARY KEY,  
    direccionAlu   VARCHAR2(30),  
    sexoAlu        VARCHAR2(1),  
    fechaNacAlu   DATE,  
    cursoAlu       NUMBER(5),  
    CONSTRAINT alu_cur_FK FOREIGN KEY (cursoAlu)  
        REFERENCES CURSOS (codigoCur),  
    CONSTRAINT alu_sex_CK CHECK (sexoAlu IN ('H','M'))  
);
```

