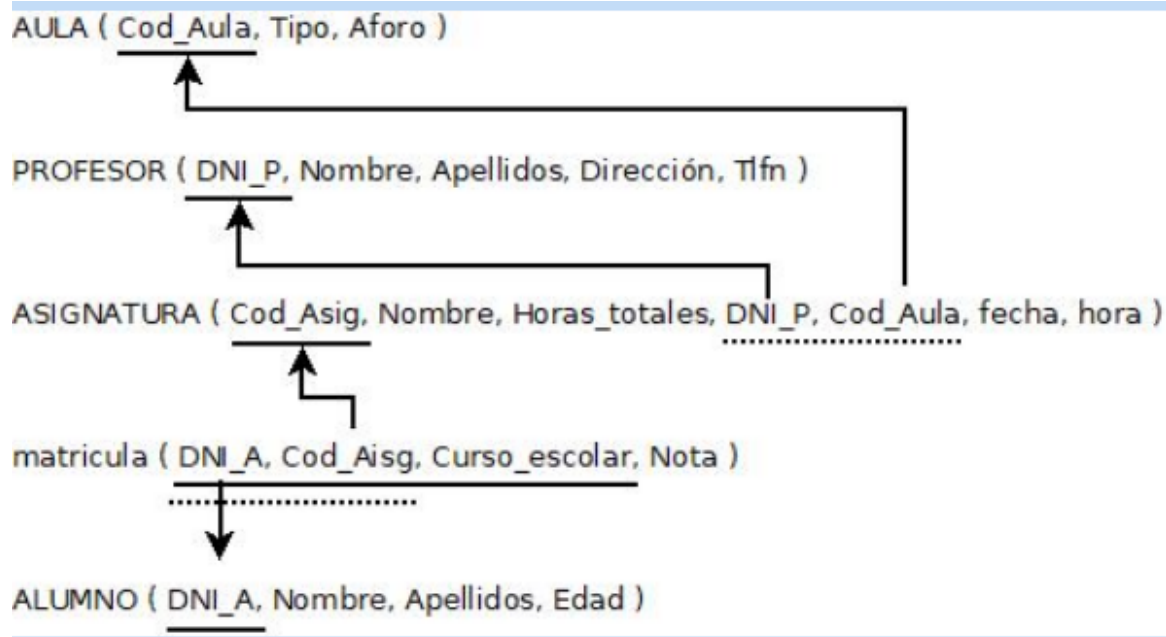


SOLUCIÓN TAREA 3

APARTADO A



Partiendo de dichas tablas, debes realizar los siguientes apartados:

A) Escribe las sentencias de creación de tablas **en el orden apropiado** utilizando el lenguaje SQL **para que funcionen correctamente en MySQL**. Debes tener en cuenta lo siguiente:

- Elegir el **tipo de dato MySQL** y el **tamaño** más adecuado para cada campo teniendo en cuenta los valores que éstos pueden almacenar.
- Recuerda nombrar adecuadamente los **atributos**, **tablas** y **restricciones** teniendo en cuenta las reglas que se explican en los contenidos de la unidad.
- Se debe **cumplir la regla de integridad referencial** teniendo en cuenta que **para cualquier clave ajena cuando un valor de la tabla principal se elimine, se pondrán a NULL los valores a los que referencia**.
- Incluir en las sentencias de creación de tablas las siguientes restricciones utilizando el lenguaje SQL:
 1. Los campos **Cod_Asig**, **Cod_Aula**, **DNI_A** y **DNI_P** identifican de forma única cada registro de sus tablas correspondientes y están formados por una cadena de caracteres alfanuméricos de longitud fija
 2. Los campos **DNI_A** y **Cod_Asig** de la tabla Se_matricula provienen de las tablas de ALUMNO y ASIGNATURA mediante las relaciones correspondientes.
 3. El valor del campo **Aforo** en la tabla AULA por defecto debe ser 30.
 4. El campo **nombre** de la tabla ASIGNATURA no puede tener valores duplicados.
 5. El campo **horas_totales** en la tabla ASIGNATURA no puede tomar valores nulos.

SOLUCIÓN:

```
CREATE DATABASE IF NOT EXISTS
TAREA3_BD1920; USE TAREA3_BD1920;
```

```
CREATE TABLE IF NOT EXISTS AULA (
Cod_Aula CHAR(5) PRIMARY KEY,
```

```
tipo VARCHAR(10),  
aforo TINYINT DEFAULT ' 30 '  
);
```

```
CREATE TABLE IF NOT EXISTS  
PROFESOR( DNI_P CHAR(9) PRIMARY  
KEY,  
nombre VARCHAR(30),  
apellido VARCHAR(30),  
dirección VARCHAR (50),  
teléfono CHAR(9)  
);
```

```
CREATE TABLE IF NOT EXISTS  
ASIGNATURA(  
Cod_Asig CHAR(5) PRIMARY KEY,  
nombre VARCHAR(30) UNIQUE,  
horas_totales TINYINT NOT NULL,  
DNI_P CHAR(9),  
Cod_aula CHAR(5),  
CONSTRAINT prof_part_FK FOREIGN KEY (DNI_P) REFERENCES  
PROFESOR(DNI_P) ON DELETE CASCADE,  
CONSTRAINT aula_part_FK FOREIGN KEY (Cod_aula) REFERENCES  
AULA(Cod_Aula) ON DELETE CASCADE,  
fecha DATE,  
hora TIME  
  
);
```

```
CREATE TABLE IF NOT EXISTS ALUMNO(  
DNI_A CHAR(9) PRIMARY KEY,  
nombre VARCHAR(30),  
apellido VARCHAR(30),  
edad TINYINT  
);
```

```
CREATE TABLE IF NOT EXISTS Se_matricula(  
DNI_A CHAR(9),  
Cod_Asig CHAR(5),  
curso_escolar YEAR,  
nota TINYINT,  
CONSTRAINT se_matricula_PK PRIMARY KEY (DNI_A,Cod_Asig,curso_escolar),  
CONSTRAINT semat_asig_FK FOREIGN KEY (Cod_Asig) REFERENCES  
ASIGNATURA(Cod_Asig) ON DELETE CASCADE,  
CONSTRAINT semat_alum_FK FOREIGN KEY (DNI_A) REFERENCES
```

ALUMNO(DNI_A) ON DELETE CASCADE
);

APARTADO B

B) Después de crear las tablas, vamos a realizar algunas modificaciones sobre ellas. Escribe las sentencias SQL apropiadas para cada uno de los subapartados siguientes:

1. Eliminar el campo **edad** de la tabla ALUMNO.
2. Crear un índice en el campo **nota** de la tabla Se_matricula.
3. Añadir el campo **estudios** en la tabla PROFESOR y que éste sólo pueda tomar los valores "Diplomado", "Licenciado" o "Catedrático".
4. Crear el usuario "**director**" con la contraseña "**admin**" para que se pueda conectar desde cualquier equipo.
5. Conceder los privilegios de actualizar y borrar datos al usuario "**secretaria**" en la tabla ASIGNATURA.
6. Quitar el privilegio para eliminar datos en la tabla ASIGNATURA al usuario "**secretaria**".

SOLUCIÓN:

7. Eliminar el campo **edad** de la tabla ALUMNO.

```
ALTER TABLE ALUMNO DROP COLUMN edad;
```

8. Crear un índice en el campo **nota** de la tabla Se_matricula.

```
ALTER TABLE Se_matricula ADD INDEX sm_notas (nota);
```

9. Añadir el campo **estudios** en la tabla PROFESOR y que éste sólo pueda tomar los valores "Diplomado", "Licenciado" o "Catedrático".

```
ALTER TABLE PROFESOR ADD estudios ENUM ('DIPLOMADO',  
'LICENCIADO', 'CATEDRÁTICO');
```

10. Crear el usuario "**director**" con la contraseña "**admin**" para que se pueda conectar desde cualquier equipo.

```
CREATE USER director@'%' IDENTIFIED BY 'admin';
```

11. Conceder los privilegios de actualizar y borrar datos al usuario "**secretaria**" en la tabla ASIGNATURA.

```
GRANT UPDATE, DELETE ON ASIGNATURA TO secretaria;
```

12. Quitar el privilegio para eliminar datos en la tabla ASIGNATURA al usuario "**secretaria**".

```
REVOKE DELETE ON ASIGNATURA FROM secretaria;
```