*Objetivo del ejercicio:*

1. *Repasar el tipo de correspondencia muchos a muchos (m a n).*
2. *Aprender el concepto de atributo compuesto.*
3. *Repasar el concepto de atributo derivado y opcional.*
4. *Repasar las cardinalidades mínimas*
5. *Aprender a modelar directamente en 3FN.*

Diseñar una base de datos para almacenar y gestionar información de los alumnos y las asignaturas de la ESO que refuerzan en una academia.

De los alumnos queremos conocer la siguiente información: Nombre, Apellidos (almacenar los apellidos por separado, ten en cuenta que algunos alumnos no tendrán segundo apellido), DNI, edad y dirección completa, junto con la localidad y ciudad.

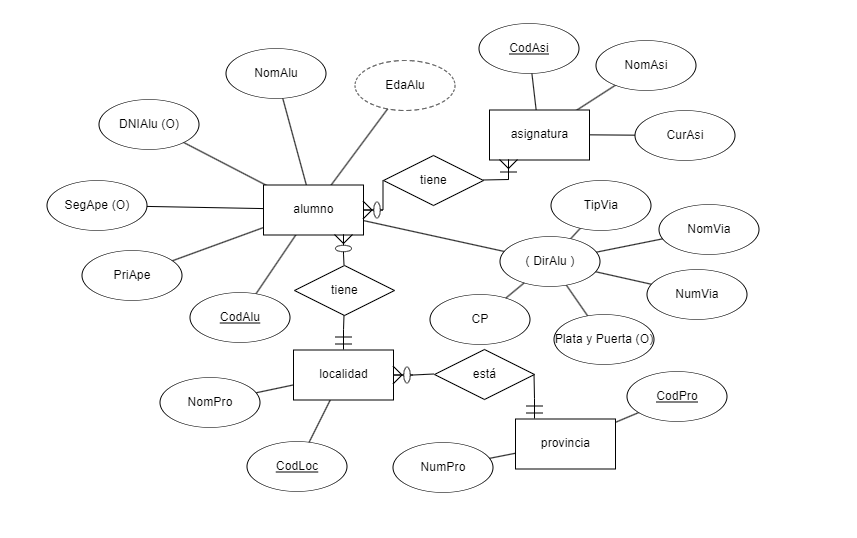
De las asignaturas almacenamos el código, nombre y curso para el que se imparte. Por ejemplo, 1ºESO.

Un alumno puede reforzar muchas asignaturas en la academia. Cuando se matricula al alumno (se da de alta), se debe indicar para qué asignatura/s lo hace.

Las asignaturas que se dan en la academia pueden estar almacenadas y que aún no se hayan matriculado ningún alumno.

Se pide:

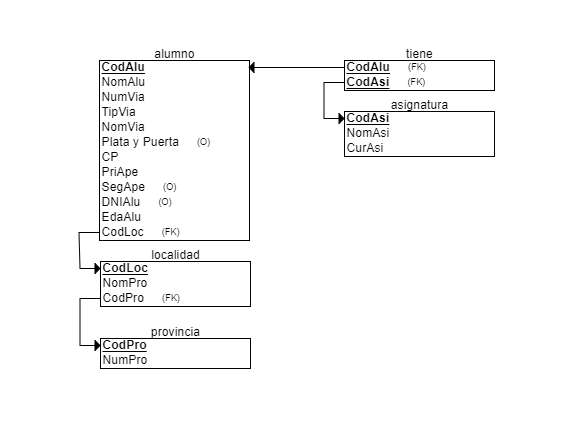
1. Modelar la base de datos. Para ello haremos:
   1. Diseño Conceptual de Datos utilizando un Diagrama o Modelo Entidad-Relación. Lo hacemos en papel y lo pasamos a la Herramienta CASE ERD Plus.



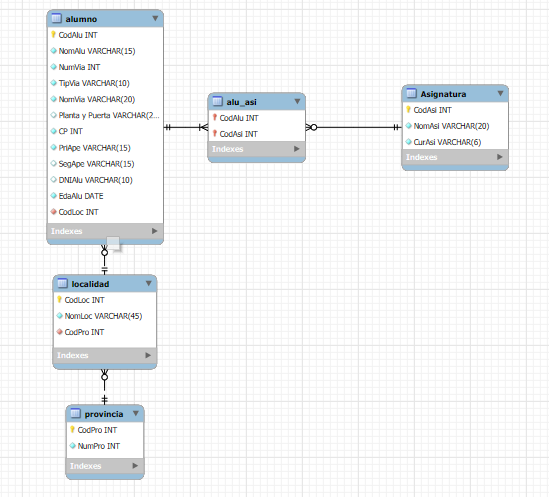
* 1. Diseño Lógico de Datos utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench. En este apartado también vamos a poner el Diagrama Referencial que genera ERD Plus a partir del Modelo Entidad-Relación.

Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

**Diagrama Referencial:**



**DED:**



* 1. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL.

CREATE DATABASE ejercicio4;

CREATE TABLE provincia

(

CodPro INT NOT NULL,

NumPro INT NOT NULL,

PRIMARY KEY (CodPro)

);

CREATE TABLE localidad

(

CodLoc INT NOT NULL,

NomLoc VARCHAR(45) NOT NULL,

CodPro INT NOT NULL,

PRIMARY KEY (CodLoc),

FOREIGN KEY (CodPro) REFERENCES provincia(CodPro)

);

CREATE TABLE asignatura

(

CodAsi INT NOT NULL,

NomAsi VARCHAR(20) NOT NULL,

CurAsi VARCHAR(6) NOT NULL,

PRIMARY KEY (CodAsi)

);

CREATE TABLE alumno

(

CodAlu INT NOT NULL,

NomAlu VARCHAR(15) NOT NULL,

NumVia INT NOT NULL,

TipVia VARCHAR(10) NOT NULL,

NomVia VARCHAR(20) NOT NULL,

Planta VARCHAR(5) NULL,

Puerta VARCHAR(5) NULL,

CP CHAR(5) NOT NULL,

PriApe VARCHAR(15) NOT NULL,

SegApe VARCHAR(15) NULL,

DNIAlu CHAR(9) NULL,

CodLoc INT NOT NULL,

Edad DATE NOT NULL,

PRIMARY KEY (CodAlu),

FOREIGN KEY (CodLoc) REFERENCES localidad(CodLoc)

);

CREATE TABLE alu\_asi

(

CodAlu INT NOT NULL,

CodAsi INT NOT NULL,

PRIMARY KEY (CodAsi, CodAlu),

FOREIGN KEY (CodAlu) REFERENCES alumno(CodAlu),

FOREIGN KEY (CodAsi) REFERENCES asignatura(CodAsi)

);

1. Insertar datos desde phpmyadmin.

INSERT INTO provincia

(CodPro,NumPro)

VALUES (1, 14),

(2, 7),

(3, 76),

(4, 23);

INSERT INTO localidad

(CodLoc,NomLoc,CodPro)

VALUES (1, 'Perú', 3),

(2, 'Málaga', 2),

(3,'Ceuta', 1),

(4,'Gibraltar', 4);

INSERT INTO asignatura

(CodAsi,NomAsi,CurAsi)

VALUES (1, 'Física y geología', '3º'),

(4, 'Educación química', '2º'),

(2, 'Biología', '9º'),

(3, 'Comentario de texto', '6º');

INSERT INTO alumno

(CodAlu,NomAlu,NumVia,TipVia,NomVia,Planta,Puerta,CP,PriApe,SegApe,DNIAlu,CodLoc,Edad)

VALUES (06, 'Santiago', 5, 'Avenida', 'Castilla', 5, 4, 29720, 'Abascal', 'Roldán', '842784910O', 2, '12/10/1492'),

(17, 'Laura', 12, 'Calle', 'Mendoza', 3, 2, 29721, 'Gómez', NULL, NULL, 4, '01/01/2000'),

(22, 'Ana', 8, 'Paseo', 'Central', 2, 3, 29723, 'Martínez', 'Jiménez', NULL, 3, '22/07/1992'),

(31, 'Carlos', 7, 'Plaza', 'Mayor', 1, 1, 29722, 'López', NULL, '87654321B', 1, '15/05/1995');

INSERT INTO alu\_asi

(CodAlu,CodAsi)

VALUES (06, 4),

(31, 1),

(17, 2),

(22, 3);

1. Realizar las siguientes consultas en SQL:

* Muestra todas las filas y todos los campos de las tablas. Ordenar el resultado de la consulta.

SELECT \*

FROM alumno

JOIN localidad ON alumno.CodLoc = localidad.CodLoc

JOIN provincia ON localidad.CodPro = provincia.CodPro

JOIN alu\_asi ON alumno.CodAlu = alu\_asi.CodAlu

JOIN asignatura ON alu\_asi.CodAsi = asignatura.CodAsi;

* Muestra algunos campos de las tablas.

SELECT CodAlu, NomAlu, PriApe, SegApe, DNIAlu FROM alumno;

* Mostrar para cada asignatura los alumnos que están matriculados. Muestra primero toda la información y posteriormente muestra solo el nombre de la asignatura y nombre con apellidos del alumno. ¿Qué ocurre con las asignaturas que aún no tiene alumnos matriculados?
* Mostrar para cada alumnos cuáles son las asignaturas que refuerza. Muestra primero toda la información y posteriormente muestra solo el nombre y apellidos del alumno junto con el nombre de la asignatura. Ordenar ascendentemente por asignatura y dentro de cada asignatura, también ascendentemente por apellidos y nombre del alumno.
* Para la asignatura de Matemáticas de 1º de la ESO, mostrar el nombre y apellidos de los alumnos que están matriculados.
* Para el alumno con DNI 12345678A, mostrar las asignaturas que refuerza.