

UD2 – Examen (Modelo B)

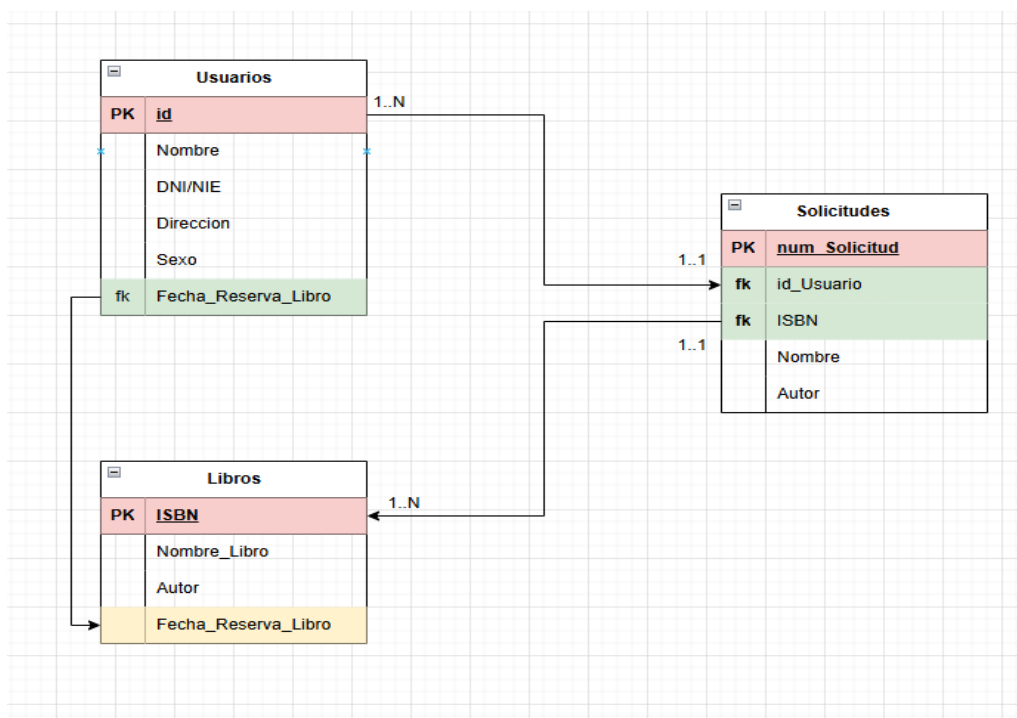
LEE LAS INSTRUCCIONES ATENTAMENTE. DEBERÁS ENTREGAR ESTE MISMO DOCUMENTO EN FORMATO PDF (CUALQUIER OTRO FORMATO DISTINTO DE PDF NO SERÁ CORREGIDO). SE ESPERA QUE PUEDAS RESOLVER ESTAS ACTIVIDADES EMPLEANDO EXCLUSIVAMENTE TUS CONOCIMIENTOS Y LOS MATERIALES QUE HAYAS DESCARGADO PREVIAMENTE. EN CASO DE CUALQUIER EVIDENCIA DE COPIA, EL EXAMEN SE EVALUARÁ COMO SUSPENSO (0).

Pregunta 1. Considerando las siguientes especificaciones:

- Los usuarios se identifican mediante su DNI/NIE, nombre, dirección y sexo.
- Los usuarios toman prestados libros. Es importante conocer la fecha en la que un usuario toma prestado un libro.
- Cada libro se identifica por su ISBN, nombre y autor
- Como se trata de una biblioteca nueva, no hay muchos libros. Los usuarios pueden completar las solicitudes para que se puedan comprar nuevos libros. Cada solicitud está vinculada a un usuario y debe incluir, como mínimo, el nombre del libro, el autor y el ISBN

Se pide:

1. Un diagrama relacional (no hace falta decir que no olvides incluir todas las tablas, columnas, relaciones y tipos de datos apropiados). (2 puntos)



2. Tu implementación SQL (que debe ajustarse a tu diagrama relacional). (2 puntos)

```
DROP DATABASE actividad2;
CREATE DATABASE actividad2;
USE actividad2;

CREATE TABLE Libros (
  ISBN INT PRIMARY KEY,
  Nombre_Libro VARCHAR (80) NOT NULL,
  Autor VARCHAR (40) NOT NULL,
  Fecha_Reserva_Libro DATETIME NOT NULL
);

CREATE TABLE Usuarios (
  id_Usuario INT PRIMARY KEY,
  Nombre VARCHAR (40) NOT NULL,
  Direccion VARCHAR (80) NOT NULL,
  Sexo Varchar (10),
  Fecha_Reserva_Libro DATETIME NOT NULL,
  foreign key (Fecha_Reserva_Libro) references Libros (Fecha_Reserva_Libro)
);

CREATE TABLE Solicitudes (
  num_Solicitud INT PRIMARY KEY,
  Nombre VARCHAR (40) NOT NULL,
  Autor VARCHAR (80) NOT NULL,
  id_Usuario INT NOT NULL,
  ISBN INT NOT NULL,
  foreign key (id_Usuario) references Usuarios (id_Usuario),
  foreign key (ISBN) references Libros (ISBN)
);
```

3. Tu diagrama de ingeniería inversa (no olvides que debe coincidir con tu diagrama relacional). (1 punto)

Pregunta 2. Piensa en una relación N:N (ejemplo: módulos y estudiantes) y una relación 1:1 (ejemplo: mensajero y furgoneta). Se pide:

1. Una breve implementación de SQL para la relación N:N (no más de 4 columnas para cada tabla) y una captura de pantalla de la ingeniería inversa. (2 puntos)
2. Una breve implementación de SQL para la relación 1:1 (no más de 4 columnas para cada tabla) y una captura de pantalla de la ingeniería inversa. (1 punto)

1 persona en el comedor del colegio, que se identifica por Nombre, apellidos, dni y edad, puede comer varios platos en la hora de la comida.

1 plato del menú, que se identifica por la comida, color de plato (dependiendo de la edad de la persona es un color diferente) y su número de plato en el menú, ese plato lo pueden pedir varias personas.

Además los utensilios solo podrán ser utilizados por 1 persona, 1 persona solo puede utilizar un par de utensilios (cuchillo y tenedor/cuchara).

```

DROP DATABASE actividad2;
CREATE DATABASE actividad2;
USE actividad2;

CREATE TABLE Persona (
  DNI INT primary key,
  Nombre varchar(40) NOT NULL,
  Apellidos VARCHAR (40) NOT NULL,
  edad int
);
CREATE TABLE Utensilios (
  id_utensilio INT PRIMARY KEY,
  tipo_utensilio varchar (10)
);
CREATE TABLE Plato (
  id_plato INT PRIMARY KEY,
  color_plato varchar (20),
  comida varchar (50),
  edad int,
  foreign key (edad) references Persona (edad)
);

```

NO PUEDES UTILIZAR LOS EJEMPLOS. EL EJERCICIO NO SERÁ EVALUADO SI LO HACES

Pregunta 3. Lee atentamente el siguiente código SQL. Hay 4 errores diferentes, resáltalos y explica cómo puedes solucionarlos. (2 puntos)

```


DROP DATABASE IF EXISTS examB;
CREATE DATABASE examB;
USE examB;

```

```

CREATE TABLE Orders (

```



```

OrderID INT,
CustomerID INT,
OrderDate INT,
TotalAmount DECIMAL(10, 2),
CONSTRAINT OrderID PRIMARY KEY (OrderID),
CONSTRAINT CustFK FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE Customers (
    CustomerID INT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100),
    PhoneNumber VARCHAR(15),
    CONSTRAINT CustomerID PRIMARY KEY (CustomerID)
);
CREATE TABLE OrderDetails (
    OrderDetailID INT,
    OrderID INT,
    Quantity INT,
    Price DECIMAL(10, 2),
    CONSTRAINT OrderDetailID PRIMARY KEY (OrderDetailID),
    CONSTRAINT OrderID FOREIGN KEY (OrderID) REFERENCES Customers(OrderID),
);

```

En este código, la table Customers debería de ser la primera ya que no tiene ninguna fk y en Orders no va a ejecutarse la fk de Customers.

En la ultima línea de la tabla OrderDetails sobra la coma del final ya que con esa coma nos daría error y no se ejecutaría la base de datos.

En la primera tabla el nombre de la foreign key está mal ya que debería de ser el mismo nombre que tiene la primary key de la tabla customers.

En las tablas order y order details hay dos atributos con el mismo nombre “OrderID” con lo cual debería ser una foreign key en una de las dos tablas.