



Tecnicatura Universitaria  
en Programación

## LABORATORIO DE COMPUTACIÓN I

Unidad Temática N°1:

Introducción al SQL. Sentencias de  
Definición de Datos

Guía de Estudio  
1° Año – 1° Cuatrimestre



## Índice

CREACIÓN DE TABLAS.....	2
Problema 1 .....	2
Problema 2: .....	5
Problema 3 .....	5
Problema 4: .....	6
Problema 5: .....	6
PROBLEMA INTEGRADOR – PARTE I.....	7
MODIFICACIÓN Y ELIMINACIÓN DE TABLAS.....	10
Problema 6: .....	10
Problema 7: .....	11
Problema Integrador – Parte II .....	12
BIBLIOGRAFÍA .....	13

## CREACIÓN DE TABLAS

### Problema 1

Se tiene el diagrama de una base de datos de una biblioteca barrial que necesita registrar los datos de los libros y los datos de sus escritores donde cada libro tiene un solo escritor y el escritor puede tener publicado varios libros, se necesita es la construcción de esta base de dato en el SGBDR.

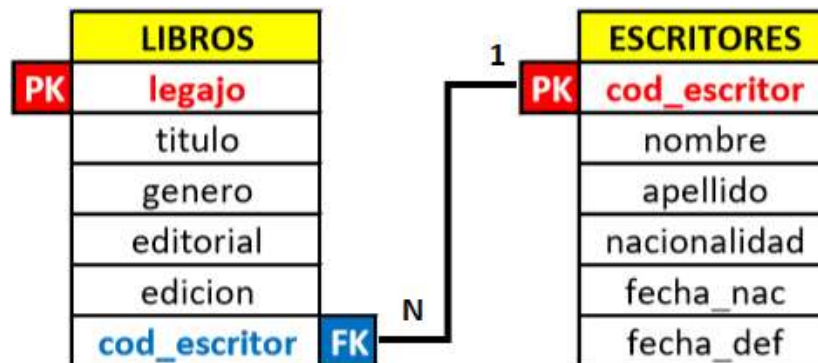
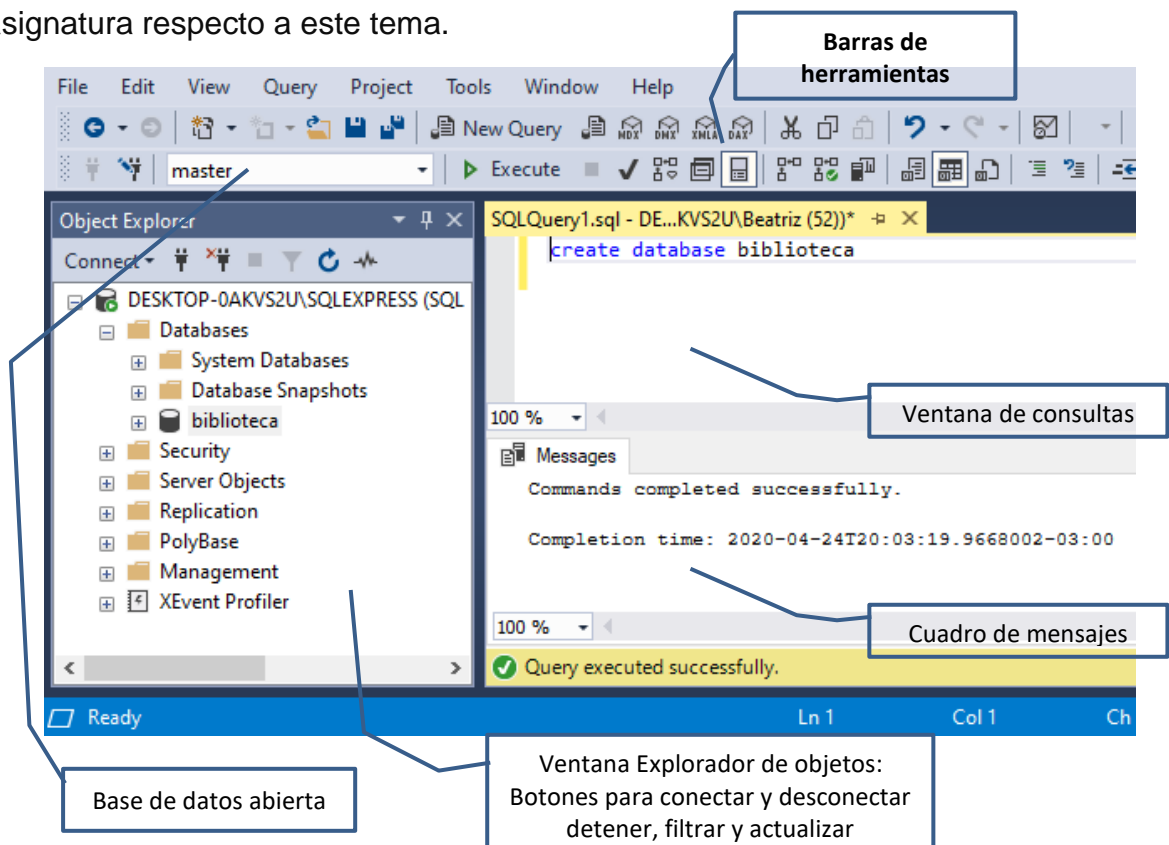


Ilustración 1: Elaboración propia

Para dar solución a este problema se utilizará SQL del Transact de Microsoft SQL Server. Utilizando la Sentencia de Definición de Datos “CREATE TABLE” se van crear las tablas de la base de datos BIBLIOTECA tal cual se presenta en el diagrama precedente. Para ello se debe abrir el SQL Server Management Studio y conectarse a un servidor de base de datos. Si no se sabe cómo hacerlo volver a ver el video de la asignatura respecto a este tema.



En una nueva consulta crear una base de datos llamada “BIBLIOTECA” con la siguiente sentencia:

**create database biblioteca**

Si no se comprende el uso de la Ventana del Management Studio ver el video de la asignatura que corresponde a este tema

Una vez escrito, seleccionar la línea y hacer clic en el botón “Ejecutar” o bien presionar la tecla F5. De esta forma acaba de crear una nueva base de datos en el servidor de SQL. Si se observa el área izquierda de la ventana se podrá observar el árbol de bases de datos, si no aparece BIBLIOTECA, hacer clic con el botón derecho del mouse sobre la carpeta “Databases” y clic en Actualizar.

Antes de la creación de los objetos dentro de la base de datos hay que abrirla (o ponerla en uso); observar en la barra de herramientas que la base de datos abierta es “master”; para abrir BIBLIOTECA escribir y ejecutar la siguiente sentencia:

**use biblioteca**

O bien desplegar la lista de bases de datos y elegir la que se quiere abrir.

Ahora sí se está en condiciones de crear las tablas anteriores y para ello se deberá utilizar la sentencia DDL CREATE del Transact-SQL (se puede repasar la misma en el material de estudio)

**create table escritores**

```
(cod_escritor int identity(1,1), --entero, autonumérico, comienza
                                --en 1 incremento = 1
apellido varchar(50) not null, -- no acepta nulos
nombre varchar(50),
nacionalidad varchar(30),
fecha_nac datetime,
fecha_nac datetime,
constraint pk_escritor primary key (cod_escritor) --crea clave
                                --primaria
)
```

Observar el texto en verde, el que está luego de los dos guiones medios, son comentarios, estos no se ejecutarán, son solo aclaraciones.

Prestar atención a cada parte de la sentencia, y verificar su utilidad en lo explicado en el material de estudio.

Una vez que se escriba absolutamente todas las líneas anteriores, seleccionárlas (desde la palabra “create” hasta el paréntesis que cierra) y ejecutarlas.

Verificar que así se haya concretado leyendo los mensajes que el sistema emite en la parte de debajo de la ventana y además que la tabla ya esté creada en la estructura en árbol de la parte izquierda abriendo su estructura hasta encontrar los campos (se debe actualizar la información que se muestra ante cada cambio).

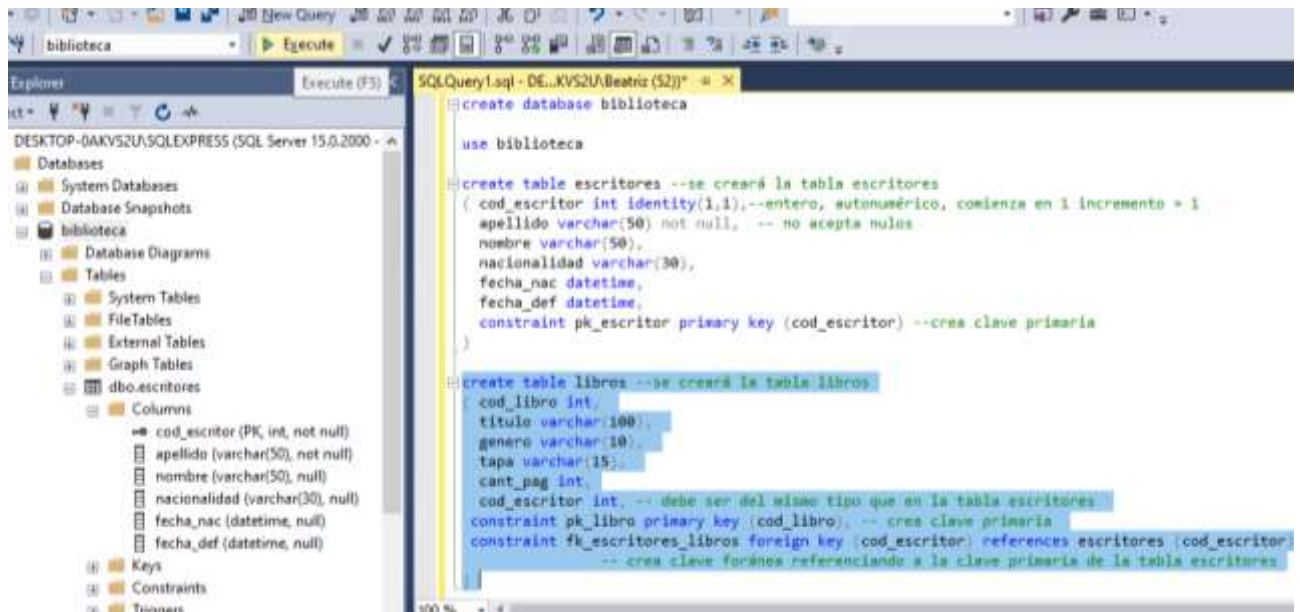


Imagen 2: Impr de Pantalla. Elaboración propia.

Ahora la segunda tabla:

```
create table libros
(cod_libro int,
titulo varchar(100),
genero varchar(10),
tapa varchar(15),
cant_pag int,
cod_escritor int, --debe ser del mismo tipo que en la tabla escritores
constraint pk_libro primary key (cod_libro), -- crea clave primaria
constraint fk_escritores_libros foreign key (cod_escritor)
references escritores(cod_escritor)
--crea clave foránea referenciando a
--la clave primaria de la tabla escritores
)
```



Ejecutar esta sentencia en forma completa y verificar que se haya creado correctamente.

**Nota:** Es importante que, al realizar este Problema, se comprenda lo siguiente:

- ¿Por qué es necesario crear primero la tabla "Escritores" y luego la tabla "Libros"?
- ¿Qué implica que el campo id\_escritor sea identity? ¿Es obligatorio crearlo como identity?
- ¿Por qué se utiliza la palabra "constraint" para la creación de la clave primaria y foránea? ¿Qué significa ese término?
- En la definición de cada campo, ¿por qué se eligió ese tipo de datos en particular y no otro?

Si no se es capaz de responder estas preguntas, rever el material teórico, ver el video sobre el tema o consulte con su tutor.

### Problema 2:

Cree en SQL una base de datos "FARMACIA" y en ella cree las tablas del diagrama que sigue que representan a la necesidad de la farmacia Cabildo de registrar los datos de los medicamentos y los datos del Laboratorio que los provee. En este caso,

**¿Qué tablas se debería crear primero?**

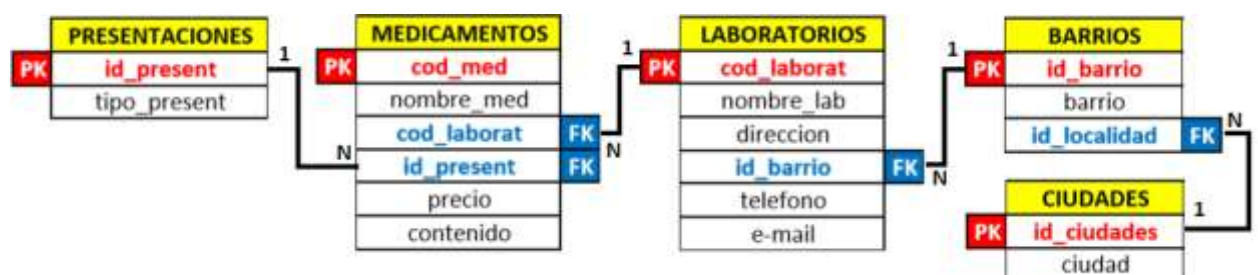


Imagen 2: Elaboración propia

### Problema 3

Cree en SQL una base de datos "SUPERMERCADO" y en ella cree las tablas diseñadas las que diagraman la siguiente necesidad de almacenamiento de datos: El dueño de un supermercado Los Álamos desea registrar los productos ingresados y su proveedor. Se debe seguir el siguiente diagrama. Antes de comenzar a crear

estas tablas se debe prestar atención a las claves primarias, foráneas y a la relación entre ellas.

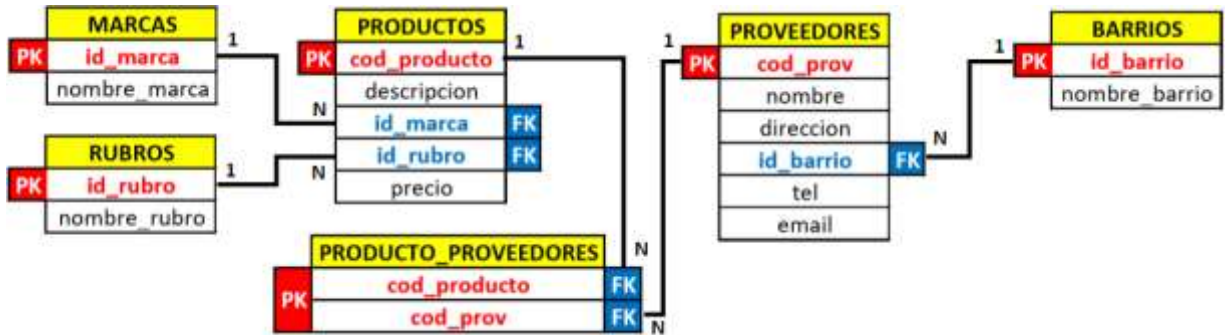


Imagen 3: Elaboración propia

Observar, también, que la tabla Productos-Proveedores tiene una clave principal compuesta, en ese caso la restricción de llave primaria sería:

**constraint** pk\_prod\_prov **primary key** (cod\_producto,cod\_prov)

#### Problema 4:

Se solicita que se construya en SQL las tablas según el siguiente diagrama que corresponden a la base de datos PagosCuotas.

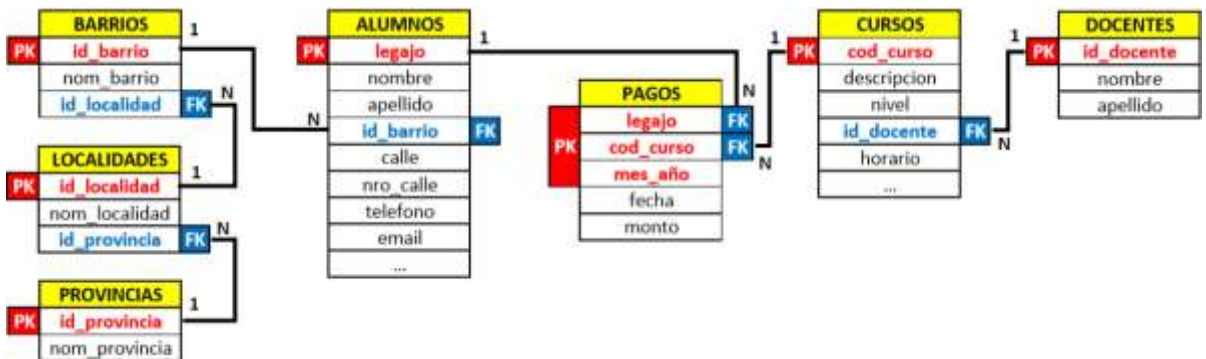


Imagen 4: Elaboración propia

#### Problema 5:

Se necesita crear las tablas de la base de datos de la secretaría del Club Los Carolinos donde se registran los pagos de los socios mediante un Recibo, en el cual se deja constancia de: la fecha del mismo, socio que pagó, la actividad que realiza, mes que paga y el monto. Con un mismo recibo el socio puede pagar varios meses y actividades.

El diagrama resultante es el siguiente:

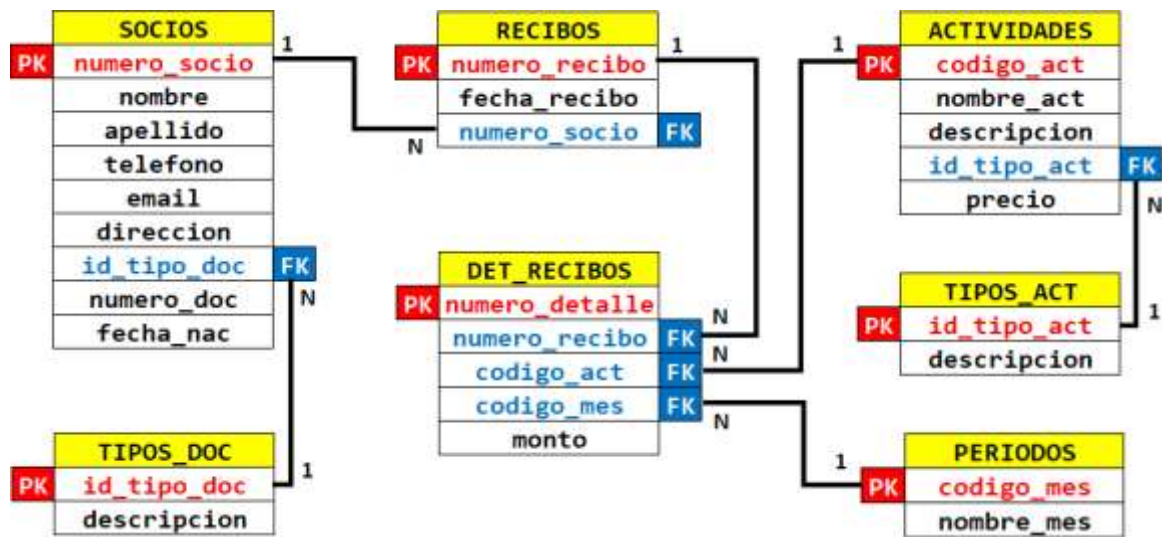


Imagen 5: Elaboración propia.

## PROBLEMA INTEGRADOR – PARTE I

### Objetivos:

Que el alumno sea capaz de:

- Integrar los conocimientos adquiridos en la Nivelación y las asignaturas Laboratorio de Computación I y Programación I.
- Aplicar dichos conocimientos a un caso práctico.
- Trabajar en grupo en forma ordenada y eficiente.

### Evaluación y Calificación:

- Se evaluará la calidad, complejidad y presentación del trabajo como así también la oportunidad de entrega en fecha
- Este práctico tiene carácter de obligatorio y deberá aprobarse con un 60% (al igual que los parciales) para regularizar la materia.



### Especificaciones y requerimientos del trabajo a presentar

- Realizar un sistema codificado en el lenguaje de programación utilizado en la materia Programación I, que sea capaz de mostrar, reportes diferentes que pueden ser propuestos por el docente del curso o por los integrantes de los diferentes grupos.
- El docente asignará los estudiantes a diferentes equipos de trabajo. No se aceptarán alumnos resolviendo el práctico en forma individual.
- Los temas podrán ser bien designados por el docente del curso o bien elegidos libremente por cada grupo (esta decisión dependerá de cada docente).
- La entrega se efectuará en cuatro partes:

1. **Entrega de la base de datos**, normalizada; es decir que se deberá presentar el diagrama de la base de datos diseñada utilizando el software que el alumno crea conveniente. No así las sentencias de definición ni de actualización de datos empleados para la creación y edición de las tablas de la base de datos

Esta presentación se realizará en formato digital (preferentemente en PDF). Este documento debe contar con carátula donde deberá constar el legajo, apellido y nombre de cada integrante del grupo, curso, materia y fecha de entrega. Y si fuera necesario una hoja con la explicación o aclaración de parte o toda la base de datos en caso de ser necesario.

El archivo a entregar debe tener un nombre acorde, por ejemplo: 1WX\_GrupoX\_Parte1, donde la X corresponde al curso (1W1, 1W3) y GrupoX al número de grupo correspondiente. En caso de presentarlo una segunda vez para corrección pueden agregarle la palabra "Correccion1" para diferenciarlo del original.

2. **Entrega de las consultas**: enunciado de lo que se quiere mostrar en la consulta y código de la consulta. Deberá entregarse junto con la primer parte en un único documento con un nombre apropiado, por ejemplo: 1WX\_GrupoX\_Parte2. Para entregar esta segunda parte, la primera debe estar ya aprobada.

El nivel de complejidad de estas consultas debe estar de acuerdo con el grado de complicación de los ejercicios trabajados en clase, utilizando todos los conocimientos adquiridos sobre Consultas SQL en esta materia, donde existan composición de varias tablas y utilizando distintos tipos de test (distintas condiciones de búsquedas), consultas agrupadas y subconsultas.

Los resultados que se pretenden obtener de estas consultas pueden ser propuestos por el docente del curso o por los integrantes del curso. En éste último caso, lo que se pretende mostrar, debe ser de utilidad para un supuesto usuario del sistema (empleado, gerente, encargado o decisor de la entidad para la cual se diseña la propuesta).

3. **Entrega del ejecutable**, el que será presentado por todos los integrantes del grupo, en una computadora, al docente del curso; para ello las tablas de la base de datos deberá contener previamente datos cargados para tal fin, es decir todas las tablas de resultado deben mostrar filas. La muestra será realizada mediante Zoom y todos los integrantes deben poseer cámara o dispositivo similar.

Los test de comparación, correspondencia, pertenencia, rango etc., deben incluir el ingreso de datos por parte del usuario del sistema desde la respectiva pantalla de la aplicación.

La tabla resultante de la consulta debe ser visualizada en pantalla, el formato de presentación de dicha información queda a criterio del grupo que la presenta teniendo en cuenta que dentro de los criterios de evaluación cuenta la calidad de la presentación.

El docente podrá realizar preguntas, referidas al práctico, a cualquier integrante del grupo, de tal forma que todos puedan demostrar que participaron activamente en la realización de todo el práctico.

La presentación de esta tercera parte supone que las dos partes anteriores ya están aprobadas.

A modo de sugerencia, con respecto a la calidad de la presentación, el grupo puede suponer que debe vender la aplicación al docente el cual puede tomar el papel de usuario final del sistema.

No se requiere la programación de todo el sistema de base de datos; solamente la muestra de las consultas.

### Enunciados de los casos propuestos para el Práctico Integrador

La elección de uno de estos enunciados, dependerá de la decisión del docente o bien de la puesta en común de todos los integrantes del grupo. Luego de realizado esto, deberán buscar información o asesoramiento del caso elegido con el objeto de realizar un correcto análisis del funcionamiento y de las necesidades de información típicas de ese tipo de sistema.

## MODIFICACIÓN Y ELIMINACIÓN DE TABLAS

### Problema 6:

Por errores en el diseño de la base de BIBLIOTECA creada como solución del problema 1, se deberá modificar para que esté correctamente normalizada como lo indica el nuevo diagrama.

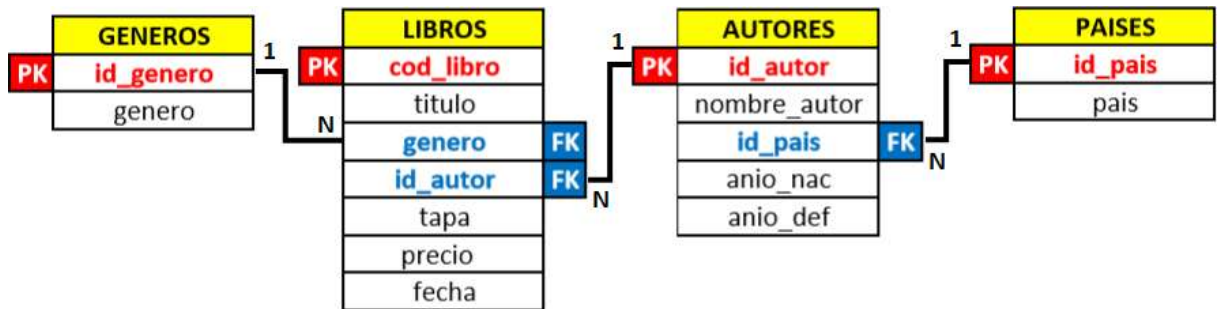


Imagen 6: Elaboración propia.

1- Para ello, abrir la base de datos y seguir los puntos siguientes:

**Nota:** Es importante que se comprenda **qué es lo que se hace en cada paso**. Si se tienen dudas se puede rever el material teórico, visualizar los videos del tema, investigar sobre el tema en Internet o consultar a su docente/tutor.

a) Agregar a la tabla libros, la columna fecha de tipo datetime.

```
alter table libros
add fecha datetime
```

b) Agregar la columna id\_país de origen a la tabla autores de tipo int y que no acepte nulos

c) Agregue la columna año de nacimiento y otra para año de defunción (que acepte nulos) ambas de tipo int.

d) Cambie el tipo de la columna genero por int

```
alter table libros
alter column genero int
```

e) Cambie el tipo de dato de la columna nombre\_autor por varchar(80)

f) Agregue la columna precio a la tabla libros

- g) Modifique el tipo de la columna precio por decimal (10,2)
  - h) Elimine la columna cant\_pag  

```
alter table libros  
drop column cant_pag
```
  - i) Elimine la columnaseudónimo
2. Cree la tabla generos con estos dos campos: id\_genero int y genero varchar(50) (sin clave primaria)
- a) Cambie el campo id\_genero para que no acepte nulos  

```
alter table generos  
alter column id_genero int not null
```
  - b) Agregar la clave primaria  

```
alter table generos  
add constraint pk_generos primary key (id_genero)
```
3. Cree la tabla Países con estos dos campos: id\_pais int y pais varchar(50) Sin clave primaria
- a) Cambie el campo id\_pais de la tabla autores para que no acepte nulos
  - b) Agregar la clave primaria en la tabla países sobre el campo id\_pais
4. Agregar la clave foránea sobre género en la tabla libros y sobre id\_país de la tabla autores
- ```
alter table libros  
add constraint fk_libros_generos foreign key (genero)  
references generos (id_genero)
```

### Problema 7:

Se tiene una lista de especificaciones a realizar para la creación de una nueva base de datos llamada "PERSONAL\_DOCENTE"

1. Crear la base de datos "PERSONAL\_DOCENTE" y abrirla para su uso

2. Crear la tabla docentes sin claves (legajo\_docente(int), nombre, apellido, e\_mail)
3. Cambie el campo legajo\_docente para que no acepte nulos
4. Agregar la clave primaria a la tabla docentes
5. Crear la tabla titulos con los siguientes campos: id\_titulo, descripcion. Cree la tabla sin PK
6. Cambie el campo id\_titulo para que no acepte nulos
7. Agregar la clave primaria sobre el campo id\_titulo
8. Agregar el campo id\_titulo a la tabla docentes
9. Modifique el campo id\_titulo de la tabla docente para que sea int
10. Agregar la clave foránea sobre id\_titulo en la tabla docentes. ¿Se podría haber creado la clave foránea antes de crear la tabla títulos? ¿Por qué?
11. Cree en SQL Server el diagrama de la base de datos para observar su estructura
12. Elimine la clave foránea sobre id\_titulo de la tabla docentes.

**alter table docentes**

**drop constraint fk\_docentes\_titulo**

13. Elimine la clave primaria de la tabla título. ¿Podría haber eliminado esta clave primaria antes del realizar el punto 14? ¿Por qué?
14. Eliminar una tabla creada por Ud. ¿Puede eliminar la tabla autores antes de libros? ¿Por qué?

## Problema Integrador – Parte II

Revisar el diseño de la base de datos designada al grupo de trabajo y realizarle los cambios que sean necesarios para optimizar la misma.



## BIBLIOGRAFÍA

Gorman K., Hirt A., Noderer D., Rowland-Jones J., Sirpal A., Ryan D. & Woody B (2019) Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud. Packt Publishing Ltd. Birmingham UK

Microsoft. SQL Server 2016. Disponible en: <https://www.microsoft.com/es-es/sql-server/sql-server-2016>

Opel, A. & Sheldon, R. (2010). Fundamentos de SQL. Madrid. Editorial Mc Graw Hill

Varga S., Cherry D., D'Antoni J. (2016). Introducing Microsoft SQL Server 2016 Mission-Critical Applications, Deeper Insights, Hyperscale Cloud. Washington. Microsoft Press



### Atribución-No Comercial-Sin Derivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterado su contenido, ni se comercialice. Referenciarlo de la siguiente manera: Universidad Tecnológica Nacional Facultad Regional Córdoba (S/D). Material para la Tecnicatura Universitaria en Programación, modalidad virtual, Córdoba, Argentina.