



UTN FRSN

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN - 2022

DISEÑO Y ADMINISTRACIÓN DE BASE DE DATOS

PROF. MESAS TABARES, LUCAS

BIBLIOTECA ESCOLAR

EES N° 9 “MARÍA ELENA WALSH”

AUTORA:

NOVELLI, MARIANELA

marianelanovelli@gmail.com



ÍNDICE

| | |
|--|----|
| INTRODUCCIÓN | 3 |
| CONTEXTO | 4 |
| ANÁLISIS DE REQUISITOS | 5 |
| DOCUMENTO DE DETALLES | 7 |
| MODELO E-R | 12 |
| MODELO RELACIONAL Y DEPENDENCIAS FUNCIONALES | 13 |
| REQUERIMIENTOS DE DATOS Y ÁLGEBRA RELACIONAL | 17 |
| DEFINICIÓN DE TABLAS SQL | 18 |
| CARGA DE DATOS SQL | 23 |
| CONSULTAS SQL | 25 |
| USUARIOS Y PERMISOS..... | 27 |

INTRODUCCIÓN

El objetivo del presente trabajo es aplicar los conocimientos adquiridos en materia de Metodologías y de Base de Datos para poder diseñar las herramientas necesarias que permitan llevar a cabo el análisis, la gestión y la administración de datos en el marco del desarrollo de un sistema para la Biblioteca Escolar de la EES N° 9 “María Elena Walsh” de la ciudad de Chivilcoy.

Se trata de un caso de trabajo cuyo origen es un problema real. La biblioteca de la mencionada institución educativa no cuenta con ningún sistema o herramienta tecnológica que le permita registrar/gestionar las operaciones diarias que realiza.

Cabe destacar que, por cuestiones de tiempo, los datos utilizados en este proyecto serán una pequeña muestra del total existente y, en consecuencia, las conclusiones emitidas pueden resultar erróneas o no reflejar la realidad con exactitud. Se hará énfasis en la verificación del buen funcionamiento, diseño y análisis de la base de datos implementada junto con todas las tablas que la conformen.

CONTEXTO

La biblioteca escolar de la EES N° 9 “María Elena Walsh” ofrece su servicio a todos los miembros de dicho establecimiento educativo. Dentro de sus múltiples funciones: facilita recursos necesarios para el proceso de enseñanza-aprendizaje de los estudiantes; promueve la lectura y sus hábitos; ofrece herramientas para el manejo e interpretación de la información; potencia la imaginación, el espíritu crítico y la toma de conciencia. Teniendo en cuenta el contexto actual en el que vivimos, dentro de una sociedad basada en la información y el conocimiento, el rol de una biblioteca escolar se ve potenciado ya que tiene total incidencia sobre la formación integral de los alumnos para su desenvolvimiento en la vida como ciudadanos responsables.

En relación a los recursos disponibles, esta biblioteca escolar cuenta con mucho material bibliográfico (alrededor de 3500 ejemplares) de múltiples géneros dispuestos en estanterías separadas por áreas o categorías. Cada libro posee un código que lo identifica y, además, es clasificado según su género, tipo o materia a la que pertenece. Además de libros, la biblioteca ofrece material periodístico, cartográfico (como mapas y globos terráqueos) y audiovisual (cortos, películas, música, fotografías...).

En cuanto al espacio físico, la biblioteca es independiente del resto de la escuela. El salón que se destinó para este espacio posee todos los servicios, incluyendo internet, pero resulta bastante chico para el mobiliario y material disponible. Más allá de los problemas de infraestructura, las bibliotecarias a cargo manifiestan necesitar una PC en buen estado (que ya fue solicitada a las autoridades) y un software práctico para el registro de las operaciones comunes que se ejecutan a diario en la biblioteca: alta y baja de libros; préstamos a estudiantes, docentes o directivos y devoluciones.

Para dar respuesta a esta necesidad, y poder desarrollar un sistema a medida, es imprescindible, en materia de base de datos, contar con información precisa que hace al funcionamiento de la biblioteca (usuarios y libros) para luego diseñar herramientas que permitan ordenar dicha información y disponerla de manera adecuada a los requerimientos del software a desarrollar.

ANÁLISIS DE REQUISITOS

Con el propósito de recopilar los requerimientos del software para la biblioteca escolar, a nivel de base de datos, se implementó una serie de herramientas de recolección de información: observación directa, entrevista con una de las bibliotecarias y acceso a documentación/manual de uso de software similar ("Aguapey", Software de Gestión para Bibliotecas Escolares y Especializadas).

A partir del análisis de lo obtenido, se concluye que los requerimientos funcionales del sistema son los siguientes:

- ❖ Registrar usuarios administradores (bibliotecarias) y usuarios cliente (alumnos, profesores y directivos)
- ❖ Dar de alta un libro a la biblioteca
- ❖ Modificar información de cada libro cargado
- ❖ Dar de baja un libro por extravío, robo, rotura, obsolescencia u otro motivo
- ❖ Registrar el préstamo de un libro
- ❖ Modificar la fecha de devolución de un libro para renovar un préstamo
- ❖ Registrar la devolución de un libro
- ❖ Modificar datos personales de todos los usuarios registrados
- ❖ Eliminar usuario cliente
- ❖ Realizar búsquedas de libros por algún campo
- ❖ Realizar búsquedas de usuarios por algún campo
- ❖ Notificar a los administradores de las devoluciones fuera de término
- ❖ Brindar información para determinar estadísticas

Todo el volumen de datos que manejará la biblioteca estará compuesto por los datos de cada integrante de la comunidad educativa de la EES N° 9 y por la información de cada uno de los libros que forma parte del mobiliario de la misma. De cada libro se registrará su código por sistema de Clasificación Decimal Universal (CDU) o Sistema de Clasificación Decimal Dewey (CDD), título, autor/es, editorial, fecha de edición, género, materia, descripción, ubicación física en la biblioteca y total de ejemplares. Mientras que, de cada usuario, se pedirá su DNI, apellido, nombre, tipo de usuario (docente/alumno/directivo), curso, división, turno, dirección y teléfono.

Cada préstamo o devolución será registrado con fecha y hora para transparentar el servicio y ejercer un mejor control. Los alumnos podrán retirar hasta tres ejemplares por un tiempo máximo de 7 días. Mientras que, directivos y docentes no tendrán un máximo de libros estipulado, pero sí deberán respetar los 7 días del préstamo.

En caso de que un usuario no cumpla con la fecha de devolución del material que tiene en su poder, será notificado por las bibliotecarias y se le pedirá que devuelva dicho

material o que renueve su préstamo vencido. De lo contrario, se les pedirá a las autoridades de la escuela que tomen medidas disciplinarias.

Todas las tablas de la base de datos de la biblioteca deberán estar bien pensadas, diseñadas y organizadas de manera tal que puedan ser relacionadas entre sí para, por ejemplo, ejecutar búsquedas por algún campo o mostrar historial de alguna operación particular.

En la actualidad, las bibliotecarias hacen el registro de todo el funcionamiento de la biblioteca en forma manual y por escrito. No utilizan ninguna herramienta, ni siquiera Hojas de Cálculo, para llevar adelante dicha tarea. Por tanto, urge pensar en un software para mejorar la calidad del servicio que brindan y su quehacer diario.

DOCUMENTO DE DETALLES

| | | |
|------------------|---------------------|-----------------|
| libro | total ejemplares | estadísticas |
| ejemplar | administradores | DNI |
| código | clientes | nombre |
| título | usuario | apellido |
| autor/es | alta | tipo de usuario |
| editorial | baja | curso |
| fecha de edición | préstamo | división |
| género | fecha y hora | turno |
| materia/área | fecha de devolución | dirección |
| descripción | devolución | teléfono |
| ubicación física | búsquedas | |

❖ *Libro = alta = ejemplar*

Tabla: “Libro”

| libro |
|--|
| id_libro serial not null codigo varchar(20) not null titulo varchar(200) not null año_edicion integer null genero varchar(50) null descripcion varchar(150) null ubicacion varchar(50) not null total_ejemplares integer not null estado smallint not null fecha_alta date not null |

Datos de identificación de cada libro:

- 1) **ID libro:** clave primaria con valor autoincremental. No puede quedar vacío este campo [serial not null].
- 2) **Código:** cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (20) not null]
- 3) **Título:** cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (200) not null]
- 4) **Año de edición:** dato numérico entero. Puede quedar vacío este campo. [integer null]
- 5) **Género:** cadena de caracteres de longitud variable limitada. Puede quedar vacío este campo. [varchar (50) null]
- 6) **Descripción:** cadena de caracteres de longitud variable limitada. Puede quedar vacío este campo. [varchar (150) null]

7) Ubicación: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (50) not null]

8) Total de ejemplares: dato numérico entero. No puede quedar vacío este campo. [integer not null]

9) Estado: dato numérico entero pequeño. No puede quedar vacío este campo. [smallint not null]

10) Fecha: fecha del día en formato aaaa/mm/dd. No puede quedar vacío este campo. [date not null]

❖ **Administradores = clientes = usuario**

Tabla: "Usuario"

| usuario |
|--|
| dni integer not null apellido char(45) not null nombre char(45) not null tipo char(25) not null curso varchar(10) null division varchar(10) null turno varchar(15) not null calle varchar(30) not null numero integer not null piso_depto varchar(20) null telefono integer not null |

Datos de identificación de cada usuario:

1) DNI: clave primaria numérica entera. No puede quedar vacío este campo. [integer not null]

2) Apellido: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [char (45) not null]

3) Nombre: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [char (45) not null]

4) Tipo: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [char (25) not null]

5) Curso: cadena de caracteres de longitud variable limitada. Puede quedar vacío este campo. [varchar (10) null]

6) División: cadena de caracteres de longitud variable limitada. Puede quedar vacío este campo. [varchar (10) null]

7) Turno: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (15) not null]

8) Dirección:

- ✓ **Calle:** cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (30) not null]
- ✓ **Número:** dato numérico entero. No puede quedar vacío este campo. [integer not null]
- ✓ **Piso y departamento:** cadena de caracteres de longitud variable limitada. Puede quedar vacío este campo. [varchar (20) null]

9) Teléfono: dato numérico entero. No puede quedar vacío este campo. [integer not null]

❖ *Préstamo = renovación*

Tabla: “Préstamo”

| prestamo |
|---|
| id_prestamo integer not null fecha_prestamo timestamp with time zone not null fecha_dev_programada timestamp with time zone not null estado_prestamo smallint not null |

Datos del registro de cada préstamo realizado:

1) ID préstamo: clave primaria entera. No puede quedar vacío este campo [integer not null].

2) Fecha de préstamo: fecha y hora del día en formato aaaa/mm/dd 00:00:00.00. No puede quedar vacío este campo. [timestamp with time zone not null]

3) Fecha de devolución programada: fecha y hora del día en formato aaaa/mm/dd 00:00:00.00. No puede quedar vacío este campo. [timestamp with time zone not null]

4) Estado: dato numérico entero pequeño. No puede quedar vacío este campo. [smallint not null]

❖ *Baja*

Tabla: “Baja”

| baja |
|---|
| id_baja serial not null motivo text not null ubicacion varchar(50) not null fecha_baja date not null |

Datos de identificación de cada libro descartado de la biblioteca:

1) ID baja: clave primaria con valor autoincremental. No puede quedar vacío este campo [serial not null].

2) Motivo: cadena de caracteres de tipo texto de longitud variable ilimitada. No puede quedar vacío este campo. [text not null]

3) Ubicación: cadena de caracteres de longitud variable limitada. No puede quedar vacío este campo. [varchar (50) not null]

4) Fecha: fecha del día en formato aaaa/mm/dd. No puede quedar vacío este campo. [date not null]

❖ **Devolución**

Tabla: “Devolucion”

| devolucion |
|---|
| id_devolucion serial not null fecha_dev_real timestamp with time zone not null comentario text null |

Datos del registro de cada devolución de libro prestado:

1. ID devolución: clave primaria con valor autoincremental. No puede quedar vacío este campo [serial not null].

2. Fecha de devolución real: fecha y hora del día en formato aaaa/mm/dd 00:00:00.00. No puede quedar vacío este campo. [timestamp with time zone not null]

3. Comentario: cadena de caracteres de tipo texto de longitud variable ilimitada. Puede quedar vacío este campo. [text null]

❖ **Tablas adicionales:**

Tabla “Autor”: para registrar ID, Apellido y Nombre de cada autor.

Tabla “Materia”: para registrar ID y Nombre de la Materia/Área.

Tabla “Editorial”: para registrar ID, Nombre/Razón Social, correo electrónico y teléfono de cada editorial.

| autor |
|--|
| id_autor serial not null apellido char(45) not null nombre char(45) null |

| materia |
|--|
| id_materia serial not null nombre char(50) not null |

| editorial |
|--|
| id_editorial serial not null nombre char(50) not null correo varchar(50) null telefono integer null |

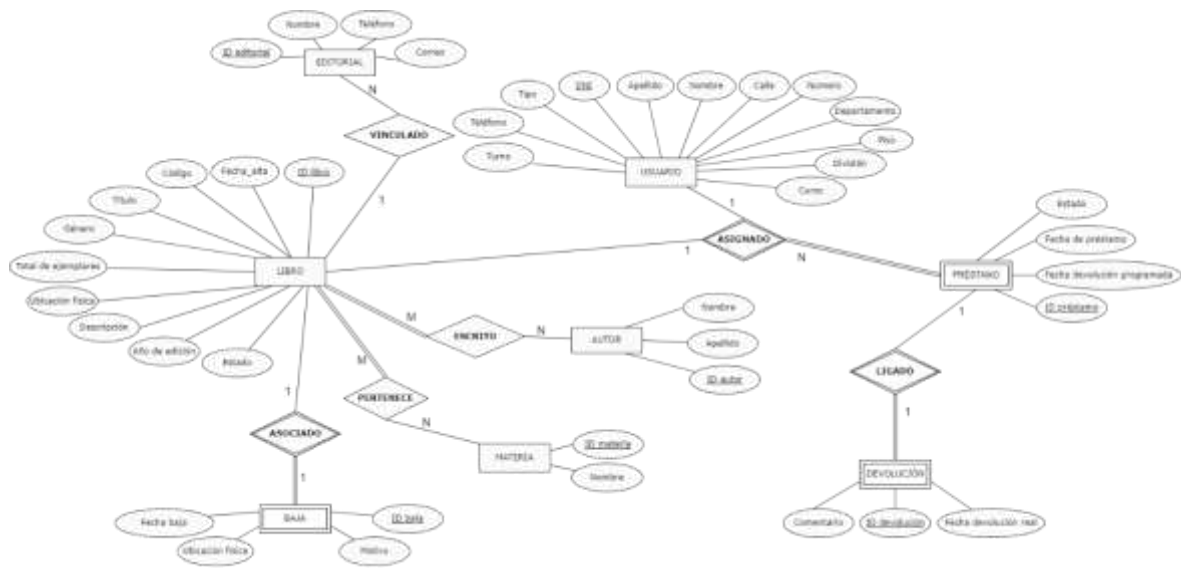
❖ **Búsquedas**

Consultas para agilizar transacciones con la base de datos.

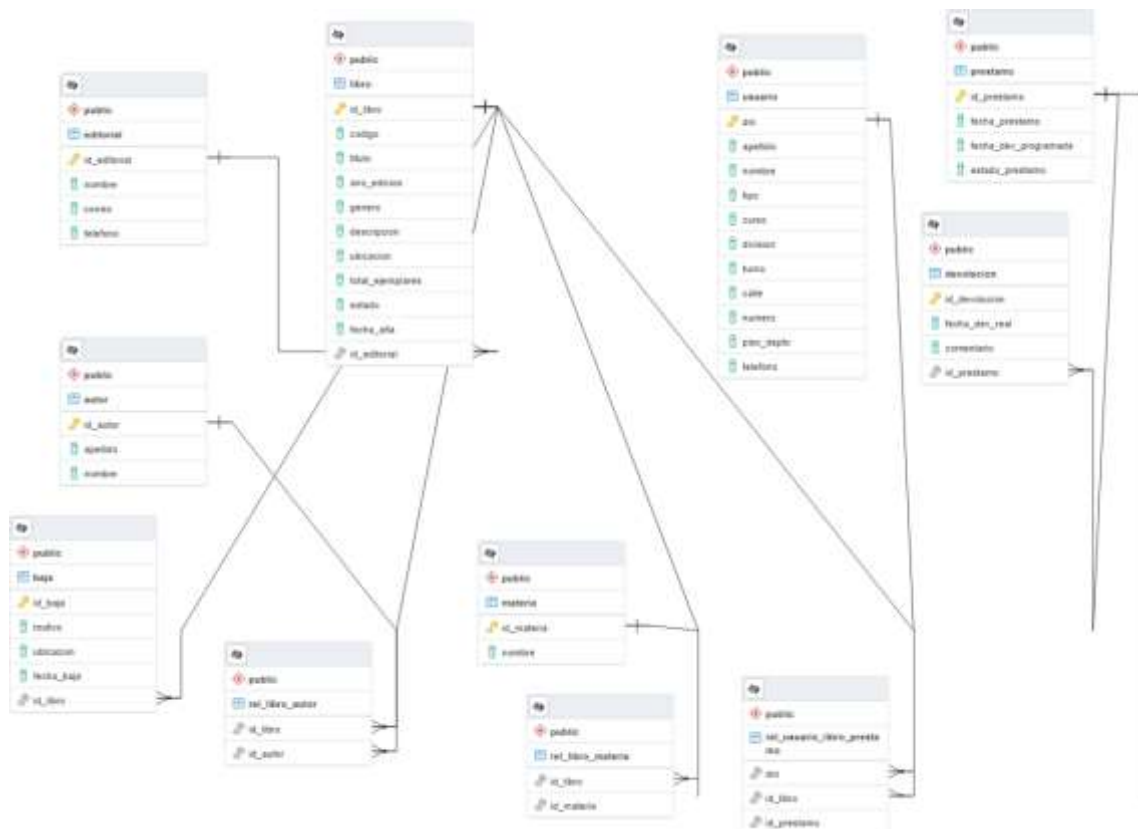
❖ **Estadísticas**

Consultas que permiten elaborar conclusiones o que brindan información clara e idónea para la toma de decisiones. Por ejemplo, mostrar cantidad de préstamos, devoluciones, altas o bajas en el último año; identificar libros más pedidos, etcétera.

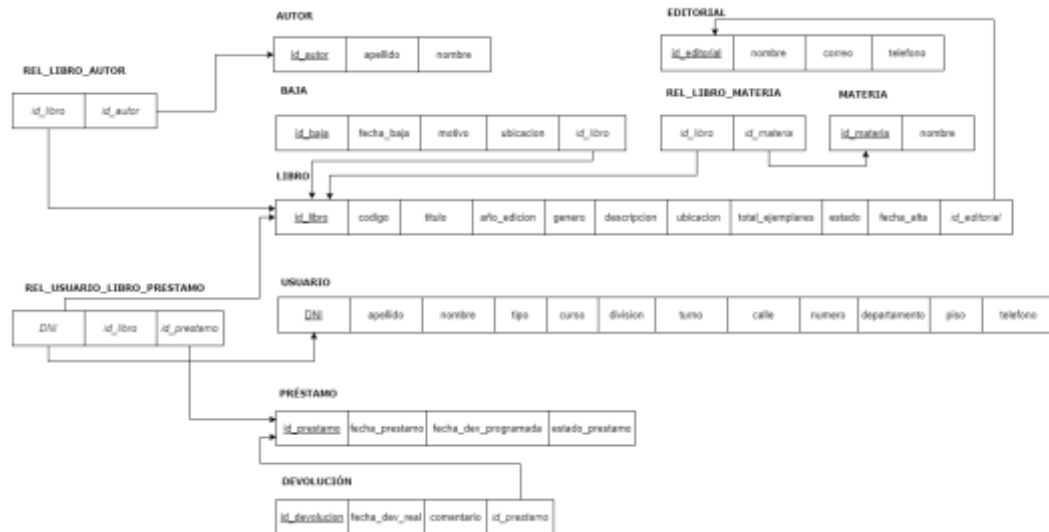
MODELO E-R



Modelo de Entidad-Relación (ERD) generado por PostgreSQL



MODELO RELACIONAL Y DEPENDENCIAS FUNCIONALES



| <u>id_libro</u> | codigo | titulo | año_edicion | genero | descripcion | ubicacion | total_ej | estado | fecha_alta |
|-----------------|--------|--------|-------------|--------|-------------|-----------|----------|--------|------------|
| | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |

$id_libro \rightarrow codigo, titulo, año_edicion, genero, descripcion, ubicacion, total_ej, estado, fecha_alta$

$id_libro \rightarrow codigo$

$id_libro \rightarrow titulo$

$id_libro \rightarrow año_edicion$

$id_libro \rightarrow genero$

$id_libro \rightarrow descripcion$

$id_libro \rightarrow ubicación$

$id_libro \rightarrow total_ej$

$id_libro \rightarrow estado$

$id_libro \rightarrow fecha_alta$

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.

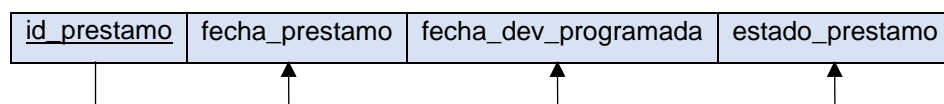
| <u>DNI</u> | apellido | nombre | tipo | curso | division | turno | calle | numero | piso_depto | telefono |
|------------|----------|--------|------|-------|----------|-------|-------|--------|------------|----------|
| | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ | ↑ |

DNI → apellido, nombre, tipo, curso, division, turno, calle, numero, piso_depto, teléfono
 DNI → apellido, nombre
 DNI → calle, numero, piso_depto
 DNI → turno, curso, division
 DNI → apellido
 DNI → nombre
 DNI → calle
 DNI → numero
 DNI → piso_depto
 DNI → telefono
 DNI → turno
 DNI → curso
 DNI → division
 DNI → tipo

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.



id_prestamo → fecha_prestamo, fecha_dev_programada, estado prestamo

id_prestamo → fecha_prestamo

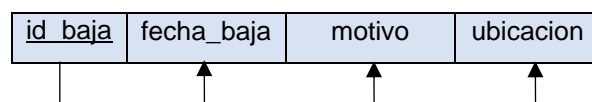
id_prestamo → fecha_dev_programada

id_prestamo → estado prestamo

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.



id_baja → fecha_baja, motivo, ubicacion

id_baja → fecha_baja

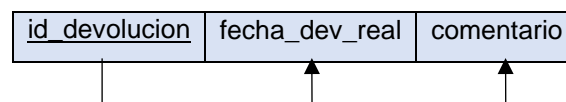
id_baja → motivo

id_baja → ubicación

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.



id_devolucion → fecha_dev_real, comentario

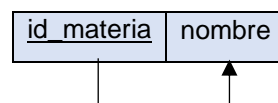
id_devolucion → fecha_dev_real

id_devolucion → comentario

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.

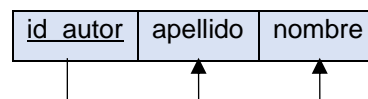


id_materia → nombre

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.



id_autor → apellido, nombre

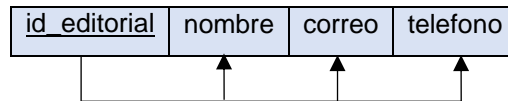
id_autor → apellido

id_autor → nombre

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.



id_editorial → nombre, correo, telefono

id_editorial → nombre

id_editorial → correo

id_editorial → telefono

Está en 1FN porque no tiene atributos multivalor ni compuestos.

Está en 2FN porque no presenta claves primarias compuestas.

Está en 3FN porque no presenta dependencias transitivas.

REQUERIMIENTOS DE DATOS Y ÁLGEBRA RELACIONAL

1. Listar, de todos los préstamos en curso, aquellos que no cumplieron con la fecha de devolución pautada.

$$\sigma((fechaActual > fechaDevProgramada) \text{ AND } estadoPrestamo = 0)(PRESTAMO)$$

2. Mostrar el título y el género del libro más pedido por los usuarios.

$$R \leftarrow (idLibro \bowtie (COUNT idPrestamo (RELUSUARIOLIBROPRESTAMO)) \text{ as } cantidadPrestamos) \\ \bowtie (\pi_{idLibro, titulo, genero}((RELUSUARIOLIBROPRESTAMO) \bowtie_{R.idLibro = libro.idLibro} (LIBRO))(LIBRO))$$

La segunda función \bowtie engloba las operaciones de orden (order by) y de límite de cantidad de registros devueltos por la consulta (limit).

3. Calcular la cantidad de altas en el año 2022.

$$\bowtie_{(COUNT ((fechaAlta \geq '2022/01/01') \text{ AND } (fechaAlta \leq '2022/12/31')) \text{ as } cantidadAltas)}(LIBRO)$$

4. Calcular la cantidad de bajas en el año 2022.

$$\bowtie_{(COUNT ((fechaBaja \geq '2022/01/01') \text{ AND } (fechaBaja \leq '2022/12/31')) \text{ as } cantidadBajas)}(BAJA)$$

5. Listar todos los libros pedidos por la alumna Marianela Novelli.

$$R \leftarrow \sigma_{apellido='Novelli' \text{ AND } nombre='Marianela'}(USUARIO) \\ S \leftarrow (USUARIO) \bowtie_{R.DNI = RELUSUARIOLIBROPRESTAMO.DNI} (RELUSUARIOLIBROPRESTAMO) \\ \pi_{apellido, nombre, idPrestamo, titulo, codigo}((RELUSUARIOLIBROPRESTAMO) \bowtie_{S.idLibro = libro.idLibro} (LIBRO))$$

6. Listar préstamos que se encuentran en curso actualmente, detallando datos del usuario y del libro que componen cada solicitud.

$$\pi_{apellido, nombre, curso, titulo, codigo}(((RELUSUARIOLIBROPRESTAMO) \\ \bowtie_{idPrestamo = prestamo.idPrestamo} (\sigma_{estadoPrestamo=0} (PRESTAMO)) \\ \bowtie_{DNI = usuario.DNI} (USUARIO) \\ \bowtie_{idLibro = libro.idLibro} (LIBRO)))$$

DEFINICIÓN DE TABLAS SQL

❖ *Tablas:*

LIBRO

| | | | | | | | | | |
|-----------------|--------|--------|-------------|--------|-------------|-----------|------------|--------|------------|
| <u>id_libro</u> | codigo | titulo | año_edicion | genero | descripcion | ubicacion | total_ejem | estado | fecha_alta |
|-----------------|--------|--------|-------------|--------|-------------|-----------|------------|--------|------------|

USUARIO

| | | | | | | | | | | |
|------------|----------|--------|------|-------|----------|-------|-------|--------|------------|----------|
| <u>DNI</u> | apellido | nombre | tipo | curso | division | turno | calle | numero | piso_depto | telefono |
|------------|----------|--------|------|-------|----------|-------|-------|--------|------------|----------|

PRÉSTAMO

| | | | |
|--------------------|----------------|----------------------|-----------------|
| <u>id_prestamo</u> | fecha_prestamo | fecha_dev_programada | estado_prestamo |
|--------------------|----------------|----------------------|-----------------|

BAJA

| | | | |
|----------------|------------|--------|-----------|
| <u>id_baja</u> | fecha_baja | motivo | ubicacion |
|----------------|------------|--------|-----------|

DEVOLUCIÓN

| | | |
|----------------------|----------------|------------|
| <u>id_devolucion</u> | fecha_dev_real | comentario |
|----------------------|----------------|------------|

MATERIA

| | |
|-------------------|--------|
| <u>id_materia</u> | nombre |
|-------------------|--------|

AUTOR

| | | |
|-----------------|----------|--------|
| <u>id_autor</u> | apellido | nombre |
|-----------------|----------|--------|

EDITORIAL

| | | | |
|---------------------|--------|--------|----------|
| <u>id_editorial</u> | nombre | correo | telefono |
|---------------------|--------|--------|----------|

❖ *Relaciones:*

1. Usuario - libro – préstamo (ternaria)

Genera tabla “rel_usuario_libro_prestamo” con las tres PK de las tablas integrantes de la relación. Estas tres claves se agregan como FK en la tercera tabla.

REL_USUARIO_LIBRO_PRESTAMO

| | | |
|-----|----------|-------------|
| DNI | id_libro | id_prestamo |
|-----|----------|-------------|

2. Préstamo – Devolución (binaria 1:1)

Se agrega la PK de “Prestamo” como FK en la tabla “Devolucion”.

DEVOLUCIÓN

| | | | |
|----------------------|----------------|------------|-------------|
| <u>id_devolucion</u> | fecha_dev_real | comentario | id_prestamo |
|----------------------|----------------|------------|-------------|

3. Libro – Baja (binaria 1:1)

Se agrega la PK de “Libro” como FK en la tabla “Baja”.

BAJA

| | | | | |
|----------------|------------|--------|-----------|----------|
| <u>id_baja</u> | fecha_baja | motivo | ubicacion | id_libro |
|----------------|------------|--------|-----------|----------|

4. Libro – Autor (binaria M:N)

Genera una nueva tabla “rel_libro_autor” que incluye las PK de las tablas “Libro” y “Autor” como FK.

REL_LIBRO_AUTOR

| | |
|----------|----------|
| id_libro | id_autor |
|----------|----------|

5. Libro – Materia (binaria M:N)

Genera una nueva tabla “rel_libro_materia” que incluye las PK de las tablas “Libro” y “Materia” como FK.

REL_LIBRO_MATERIA

| | |
|----------|------------|
| id_libro | id_materia |
|----------|------------|

6. Libro – Editorial (binaria N:1)

Se agrega la PK de “Editorial” como FK en la tabla “Libro”.

LIBRO

| | | | | | | | | | |
|-----------------|--------|--------|-------------|--------|-------------|-----------|----------|--------|------------|
| <u>id_libro</u> | codigo | titulo | año_edicion | genero | descripcion | ubicacion | total_ej | estado | fecha_alta |
|-----------------|--------|--------|-------------|--------|-------------|-----------|----------|--------|------------|

| | | |
|----------|------------|--------------|
| id_autor | id_materia | id_editorial |
|----------|------------|--------------|

❖ *Tablas en PostgreSQL:*

```
CREATE TABLE libro (  
    id_libro serial NOT NULL PRIMARY KEY,  
    codigo varchar(20) NOT NULL,  
    titulo varchar(200) NOT NULL,  
    ano_edicion integer NULL,  
    genero varchar(50) NULL,  
    descripcion varchar(150) NULL,  
    ubicacion varchar(50) NOT NULL,  
    total_ejemplares integer NOT NULL,  
    estado smallint NOT NULL,  
    fecha_alta date NOT NULL,  
    id_editorial integer NOT NULL  
);
```

```
ALTER TABLE libro  
    ADD CONSTRAINT fk_id_editorial  
    FOREIGN KEY (id_editorial)  
    REFERENCES editorial(id_editorial);
```

```
CREATE TABLE usuario (  
    dni integer NOT NULL PRIMARY KEY,  
    apellido char(45) NOT NULL,  
    nombre char(45) NOT NULL,  
    tipo char(25) NOT NULL,  
    curso varchar(10) NULL,  
    division varchar(10) NULL,  
    turno varchar(15) NOT NULL,  
    calle varchar(30) NOT NULL,  
    numero integer NOT NULL,  
    piso_depto varchar(20) NULL,  
    telefono integer NOT NULL  
);
```

```
CREATE TABLE prestamo (  
    id_prestamo integer NOT NULL PRIMARY KEY,  
    fecha_prestamo timestamp with time zone NOT NULL,  
    fecha_dev_programada timestamp with time zone NOT NULL,  
    estado_prestamo smallint NOT NULL  
);
```

```
CREATE TABLE baja (  
    id_baja serial NOT NULL PRIMARY KEY,  
    motivo text NOT NULL,  
    ubicacion varchar(50) NOT NULL,  
    fecha_baja date NOT NULL,  
    id_libro integer NOT NULL  
);
```

```
ALTER TABLE baja  
    ADD CONSTRAINT fk_id_libro  
    FOREIGN KEY (id_libro)  
    REFERENCES libro(id_libro);
```

```
CREATE TABLE devolucion (  
    id_devolucion serial NOT NULL PRIMARY KEY,  
    fecha_dev_real timestamp with time zone NOT NULL,  
    comentario text NULL,  
    id_prestamo integer NOT NULL  
);
```

```
ALTER TABLE devolucion  
    ADD CONSTRAINT fk_id_prestamo  
    FOREIGN KEY (id_prestamo)  
    REFERENCES prestamo(id_prestamo);
```

```
CREATE TABLE materia (  
    id_materia serial NOT NULL PRIMARY KEY,  
    nombre char(50) NOT NULL  
);
```

```
CREATE TABLE autor (  
    id_autor serial NOT NULL PRIMARY KEY,  
    apellido char(45) NOT NULL,  
    nombre char(45) NULL  
);
```

```
CREATE TABLE editorial (  
    id_editorial serial NOT NULL PRIMARY KEY,  
    nombre char(50) NOT NULL,  
    correo varchar(50) NULL,  
    telefono integer NULL  
);
```

```
CREATE TABLE rel_usuario_libro_prestamo (  
    dni integer NOT NULL,  
    id_libro integer NOT NULL,  
    id_prestamo integer NOT NULL  
);
```

```
ALTER TABLE rel_usuario_libro_prestamo  
    ADD CONSTRAINT fk_dni_usuario  
    FOREIGN KEY (dni)  
    REFERENCES usuario(dni),  
    ADD CONSTRAINT fk_id_libro  
    FOREIGN KEY (id_libro)  
    REFERENCES libro(id_libro),  
    ADD CONSTRAINT fk_id_prestamo  
    FOREIGN KEY (id_prestamo)  
    REFERENCES prestamo(id_prestamo);
```

```
CREATE TABLE rel_libro_autor (  
    id_libro integer NOT NULL,  
    id_autor integer NOT NULL  
);
```

```
ALTER TABLE rel_libro_autor  
    ADD CONSTRAINT fk_id_libro  
    FOREIGN KEY (id_libro)  
    REFERENCES libro(id_libro),  
    ADD CONSTRAINT fk_id_autor  
    FOREIGN KEY (id_autor)  
    REFERENCES autor(id_autor);
```

```
CREATE TABLE rel_libro_materia (  
    id_libro integer NOT NULL,  
    id_materia integer NOT NULL  
);
```

```
ALTER TABLE rel_libro_materia  
    ADD CONSTRAINT fk_id_libro  
    FOREIGN KEY (id_libro)  
    REFERENCES libro(id_libro),  
    ADD CONSTRAINT fk_id_materia  
    FOREIGN KEY (id_materia)  
    REFERENCES materia(id_materia);
```

CARGA DE DATOS SQL

```
INSERT INTO usuario (dni, apellido, nombre, tipo, curso, division, turno, calle, numero, piso_depto, telefono)
VALUES
(48712284, 'Lucero', 'Bruno', 1, '6', 'U', 'Mañana', 'Ceballos', 488, NULL, 506166),
(44991232, 'Reynoso', 'Rodrigo', 1, '5', 'U', 'Mañana', 'Av. Mitre', 1388, NULL, 588888),
(45882489, 'Bustos', 'Ariana', 1, '3', 'U', 'Mañana', 'Av. Lisandro de la Torre', 68, NULL, 485861),
(37783962, 'Novelli', 'Marianela', 2, '2', 'U', 'Mañana', 12, 46, NULL, 506572),
(17649128, 'Suárez', 'Claudia', 3, NULL, NULL, 'Mañana', 'Ceballos', 18, '6° B', 581483),
(46843883, 'Bruno', 'Uriel', 1, '3', 'U', 'Mañana', 508, 55, NULL, 433749),
(44721888, 'Pérez', 'Tatiana', 1, '1', 'U', 'Mañana', 'Alem', 498, '1° A', 486578),
(45638988, 'Campos', 'Laura', 1, '2', 'U', 'Mañana', 19, 1288, NULL, 488828),
(36996843, 'Massafra', 'Carolina', 2, NULL, NULL, 'Mañana', 'Brown', 58, NULL, 481421),
(44549322, 'Mesplet', 'Mateo', 1, '2', 'U', 'Mañana', 'Alvear', 1888, NULL, 459798);
```

```
INSERT INTO libro (codigo, titulo, ano_edicion, genero, descripcion, ubicacion, total_ejemplares, estado, fecha_alta, id_editorial)
VALUES
(4881, 'Don Quijote de la Mancha', 1991, 'caballeresco', 'clásico', 'estantería 3', 5, 8, '2085/83/15', 1),
(4882, 'El Principito', 1996, 'aventura', 'clásico', 'estantería 3', 4, 1, '2888/81/18', 2),
(5881, 'Matemática I', 2081, 'libro de texto', 'para secundaria superior', 'estantería 2', 38, 1, '2885/11/28', 4),
(9881, 'Introducción a la Geopolítica', 2822, 'libro de texto', 'nuevo', 'estantería 1', 15, 1, '2822/88/81', 6),
(5882, 'Ciencias Naturales I', 2885, 'libro de texto', 'para secundaria básica', 'estantería 4', 32, 1, '2828/82/81', 4),
(4883, 'Rayuela', 1963, 'novela', 'obra latinoamericana', 'estantería 4', 2, 8, '2888/88/81', 3),
(4818, 'Encuentros', 2812, 'novela', 'psicología', 'archivo 1', 1, 8, '2828/89/28', 7),
(4885, 'Fortunata y Jacinta', 1984, 'novela', NULL, 'estantería 3', 1, 2, '2888/83/18', 5),
(8881, 'Series fantásticas', 2828, 'ficción', 'infantil', 'estantería 1', 5, 8, '2821/85/88', 1),
(5888, 'Matemática II', 2821, 'libro de texto', 'para 1° o 2° año', 'estantería 2', 29, 8, '2822/83/18', 4);
```

```
INSERT INTO editorial (nombre, correo, telefono)
VALUES
('Anaya', NULL, 81144465231),
('Andina', 'andina@info.com', 81122278787),
('Sudamericana', NULL, 81188578621),
('Santillana', 'infoar@santillana.com', 81174412345),
('Píaza & Janés', NULL, 81178845236),
('SINTESIS', NULL, NULL),
('Booket', 'contactobooket@info.com', 81178965288);
```

```
INSERT INTO autor (apellido, nombre)
VALUES
('Cerbantes', 'Miguel'),
('Saint-Exupery', 'Antony'),
('Cortázar', 'Julio'),
('Kaczor', 'Pablo'),
('Pérez Galdós', NULL),
('Loyer', 'Bárbara'),
('Alcoles', 'Ana'),
('Rolón', 'Gabriel');
```

```
INSERT INTO materia (nombre)
VALUES
('Prácticas del Lenguaje'),
('Matemática'),
('Geografía'),
('Infantiles'),
('Ciencias Naturales'),
('Psicología');
```

```
INSERT INTO baja (motivo, ubicacion, fecha_baja, id_libro)
VALUES
('desuso', 'estantería 3', '2822/83/38', 18),
('rotura', 'estantería 2', '2822/18/82', 28);
```

```
INSERT INTO devolucion (fecha_dev_real, comentario, id_prestamo)
VALUES
('2822/18/25', NULL, 3),
('2822/11/11', 'Fuera de término', 1);
```

```

INSERT INTO prestamo (id_prestamo, fecha_prestamo, fecha_dev_programada, estado_prestamo)
VALUES
('1', '2022/10/27', '2022/11/03', 1),
('2', '2022/10/26', '2022/11/02', 0),
('3', '2022/10/20', '2022/10/27', 1),
('4', '2022/10/28', '2022/11/04', 0),
('5', '2022/09/23', '2022/09/30', 0),
('6', '2022/11/08', '2022/11/15', 0),
('7', '2022/11/07', '2022/11/14', 0),
('8', '2022/11/14', '2022/11/21', 0),
('9', '2022/11/14', '2022/11/21', 0);

```

```

INSERT INTO rel_libro_autor (id_libro, id_autor)
VALUES
(11, 1),
(12, 2),
(13, 4),
(14, 6),
(15, 4),
(16, 3),
(17, 8),
(18, 5),
(19, 7),
(20, 4);

```

```

INSERT INTO rel_libro_materia (id_libro, id_materia)
VALUES
(11, 1),
(12, 1),
(13, 2),
(14, 3),
(15, 5),
(16, 1),
(17, 6),
(18, 1),
(19, 4),
(20, 2);

```

```

INSERT INTO rel_usuario_libro_prestamo (dni, id_libro, id_prestamo)
VALUES
(40712284, 12, 1),
(37783962, 13, 2),
(36996843, 14, 3),
(46843003, 15, 4),
(45630908, 13, 5),
(44549322, 13, 6),
(40712284, 14, 7),
(37783962, 19, 8),
(37783962, 17, 9);

```


CONSULTAS SQL

1. Listar, de todos los préstamos en curso, aquellos que no cumplieron con la fecha de devolución pautada.

```
SELECT * FROM prestamo
WHERE (CURRENT_DATE > fecha_dev_programada) AND estado_prestamo = 0
```

Resultado de la consulta ejecutada el 12/11/2022:

| | id_prestamo [PK] integer | fecha_prestamo timestamp with time zone | fecha_dev_programada timestamp with time zone | estado_prestamo smallint |
|---|-----------------------------|--|--|-----------------------------|
| 1 | 2 | 2022-10-26 00:00:00-03 | 2022-11-02 00:00:00-03 | 0 |
| 2 | 4 | 2022-10-28 00:00:00-03 | 2022-11-04 00:00:00-03 | 0 |
| 3 | 5 | 2022-09-23 00:00:00-03 | 2022-09-30 00:00:00-03 | 0 |

2. Mostrar el título y el género del libro más pedido por los usuarios.

```
SELECT libro.id_libro, count (rel_usuario_libro_prestamo.id_libro) AS cantidad_prestamos, titulo, genero
FROM rel_usuario_libro_prestamo
INNER JOIN libro
ON rel_usuario_libro_prestamo.id_libro = libro.id_libro
GROUP BY libro.id_libro
ORDER BY cantidad_prestamos DESC
LIMIT 1
```

Resultado de la consulta:

| | id_libro [PK] integer | cantidad_prestamos bigint | titulo character varying (200) | genero character varying (50) |
|---|--------------------------|------------------------------|-----------------------------------|----------------------------------|
| 1 | 13 | 3 | Matemática I | libro de texto |

3. Calcular la cantidad de altas en el año 2022.

```
SELECT count ((fecha_alta >= '2022/01/01') AND (fecha_alta <= '2022/12/31'))
AS cantidad_altas
FROM libro
```

Resultado de la consulta:

| | cantidad_altas bigint |
|---|--------------------------|
| 1 | 10 |

4. Calcular la cantidad de bajas en el año 2022.

```
SELECT count ((fecha_baja >= '2022/01/01') AND (fecha_baja <= '2022/12/31'))  
AS cantidad_bajas  
FROM baja
```

Resultado de la consulta:

| | cantidad_bajas bigint |
|---|--------------------------|
| 1 | 2 |

5. Listar todos los libros pedidos por la alumna Marianela Novelli.

```
SELECT apellido, nombre, id_prestamo, titulo, codigo FROM usuario  
INNER JOIN rel_usuario_libro_prestamo  
ON usuario.dni = rel_usuario_libro_prestamo.dni  
INNER JOIN libro  
ON rel_usuario_libro_prestamo.id_libro = libro.id_libro  
WHERE apellido = 'Novelli' AND nombre = 'Marianela'
```

Resultado de la consulta:

| | apellido character (45) | nombre character (45) | id_prestamo integer | titulo character varying (200) | codigo character varying (20) |
|---|----------------------------|--------------------------|------------------------|-----------------------------------|----------------------------------|
| 1 | Novelli | Marianela | 2 | Matemática I | 5001 |
| 2 | Novelli | Marianela | 8 | Seres fantásticos | 8001 |
| 3 | Novelli | Marianela | 9 | Encuentros | 4010 |

6. Listar préstamos que se encuentran en curso actualmente, detallando datos del usuario y del libro que componen cada solicitud.

```
SELECT apellido, nombre, curso, titulo, codigo FROM rel_usuario_libro_prestamo  
INNER JOIN prestamo  
ON prestamo.id_prestamo = rel_usuario_libro_prestamo.id_prestamo  
INNER JOIN usuario  
ON usuario.dni = rel_usuario_libro_prestamo.dni  
INNER JOIN libro  
ON libro.id_libro = rel_usuario_libro_prestamo.id_libro  
WHERE estado_prestamo = 0
```

Resultado de la consulta:

| | apellido character (45) | nombre character (45) | curso character varying (10) | titulo character varying (200) | codigo character varying (20) |
|---|----------------------------|--------------------------|---------------------------------|-----------------------------------|----------------------------------|
| 1 | Novelli | Marianela | 2° | Matemática I | 5001 |
| 2 | Bruno | Uriel | 3° | Ciencias Naturales I | 5002 |
| 3 | Campos | Laura | 2° | Matemática I | 5001 |
| 4 | Mesplet | Mateo | 2° | Matemática I | 5001 |
| 5 | Lucero | Bruno | 6° | Introducción a la Geop. | 9001 |
| 6 | Novelli | Marianela | 2° | Seres fantásticos | 8001 |
| 7 | Novelli | Marianela | 2° | Encuentros | 4010 |

USUARIOS Y PERMISOS

Los usuarios administradores (las bibliotecarias) contarán con todos los permisos disponibles sobre la base de datos. Por tanto, podrán acceder a toda su estructura y a todos los datos ingresados.

Por otra parte, docentes, directivos y alumnos (usuarios cliente) tendrán permiso de lectura para consultar y ver los libros disponibles en la biblioteca.

Los permisos de acceso en PostgreSQL se administran definiendo *roles*. Cada uno de estos roles tiene asignados los permisos necesarios para poder realizar ciertas acciones.

```
/* CREACIÓN USUARIO ADMINISTRADOR */
CREATE USER administrador WITH
SUPERUSER LOGIN
PASSWORD 'biblioteca';
/* PERMISOS */
GRANT ALL PRIVILEGES ON DATABASE biblioteca TO administrador;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO administrador;

/*Creación de usuarios*/
CREATE USER alumno WITH NOLOGIN;
CREATE USER docente WITH NOLOGIN;
CREATE USER directivo WITH NOLOGIN;

/* ROL SÓLO LECTURA */
CREATE ROLE readonly;
--Conecto el rol a la DB
GRANT CONNECT ON DATABASE biblioteca TO readonly;
--Otorgo permiso de uso del rol al esquema
GRANT USAGE ON SCHEMA public TO readonly;
--Otorgo permisos a readonly
GRANT SELECT ON TABLE libro TO readonly;

/* ROL LECTURA Y ESCRITURA */
CREATE ROLE readwrite;
--Conecto el rol a la DB
GRANT CONNECT ON DATABASE biblioteca TO readwrite;
--Otorgo permiso de uso del rol al esquema
GRANT USAGE ON SCHEMA public TO readwrite;
--Otorgo permisos a readonly
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public TO readwrite;

/* ASIGNACIÓN DE PERMISO LECTURA Y ESCRITURA */
GRANT readwrite TO administrador;

/* ASIGNACIÓN DE PERMISO SÓLO LECTURA */
GRANT readonly TO alumno;
GRANT readonly TO docente;
GRANT readonly TO directivo;
```