

Ejercicios

BASE DE DATOS

Esteban Oller

3 de abril de 2024

Enlace de ejercicios en GitHub:

<https://github.com/zEstebanz/DB2>

Caso 1:

En la biblioteca de la Facultad, existen dos tipos de usuario: sin carnet, que sólo pueden realizar consultas en sala de lectura y con carnet, que pueden llevarse libros en préstamo hasta por una semana. Es de interés registrar todos los préstamos. El material bibliográfico está compuesto por libros y revistas, y estas últimas sólo pueden consultarse en la sala de lectura. Un libro o revista puede tener varios ejemplares.

1.Creación DB:

```
mysql> CREATE DATABASE biblioteca;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| biblioteca |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

2. Creación Tabla Usuarios:

```
mysql> CREATE TABLE usuarios( id_usuario INT AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(100), tipo_usuario ENUM('con carnet', 'sin carnet'));
Query OK, 0 rows affected (0.10 sec)

mysql> SHOW COLUMNS FROM usuarios;
+----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+----+-----+-----+-----+-----+-----+
| id_usuario | int                  | NO   | PRI | NULL    | auto_increment |
| nombre     | varchar(100)         | YES  |     | NULL    |              |
| tipo_usuario | enum('con carnet','sin carnet') | YES  |     | NULL    |              |
+----+-----+-----+-----+-----+-----+
```

3-Creación de tabla Material Bibliográfico:

```
mysql> CREATE TABLE MaterialBibliografico (
-> id_material INT AUTO_INCREMENT KEY,
-> titulo VARCHAR(100),
-> tipo ENUM('libro', 'revista')
-> );
Query OK, 0 rows affected (0.05 sec)

mysql> _
```

4- Creación tabla de ejemplares:

```
mysql> CREATE TABLE Ejemplares (
-> id_ejemplar INT AUTO_INCREMENT PRIMARY KEY,
-> id_material INT,
-> disponible BOOLEAN,
-> FOREIGN KEY (id_material) REFERENCES MaterialBibliografico(id_material)
-> );
Query OK, 0 rows affected (0.06 sec)
```

5- Creación tabla de Préstamos:

```
mysql> CREATE TABLE Prestamos ( id_prestamo INT AUTO_INCREMENT PRIMARY KEY, id_usuario  
INT, id_ejemplar INT, fecha_prestamo DATE, fecha_devolucion DATE, FOREIGN KEY (id_usu  
ario) REFERENCES usuarios(id_usuario), FOREIGN KEY (id_ejemplar) REFERENCES Ejemplares  
(id_ejemplar) );  
Query OK, 0 rows affected (0.19 sec)  
  
mysql>
```

Relaciones entre tablas:

- La Tabla **usuario** tiene una relación con **Prestamos** está asociado con un usuario, de la tabla **usuarios**, y un usuario puede tener varios préstamos.
- La tabla **Material Bibliográfico** esta tiene una relación con **Ejemplares**, cada ejemplar está asociado con un material bibliográfico y un material bibliográfico puede tener varios ejemplares.
- La tabla **Ejemplares** tiene una relación con **Préstamos**, cada prestamos está asociado con un ejemplar y un ejemplar puede ser prestado varias veces.

Caso 2:

Una empresa constructora se encuentra organizada en distintos sectores, cada uno de los cuales tiene un encargado de quién dependen diversos empleados. Cada sector se corresponde con un grupo de actividades específico en una obra y puede o no estar afectado a una obra en un cierto momento. Los empleados pueden ser efectivos y eventuales. Estos últimos se contratan durante la obra de acuerdo con las necesidades de la misma, y están asignados a una obra en particular, mientras que los empleados efectivos pueden atender más de una obra simultáneamente

1-Creación y utilización de Base de Datos 'constructora'

```
mysql> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| biblioteca  
| constructora  
| information_schema  
| mysql  
| performance_schema  
| sys  
+-----+  
6 rows in set (0.00 sec)
```

```
mysql> USE constructora;  
Database changed
```

2-Creación de tabla Empleados:

```
mysql> CREATE TABLE Empleados (  
->     id_empleado INT AUTO_INCREMENT PRIMARY KEY,  
->     nombre VARCHAR(100),  
->     tipo_empleado ENUM('efectivo', 'eventual', 'encargado'),  
->     id_sector INT,  
->     id_obra INT  
-> );  
Query OK, 0 rows affected (0.04 sec)  
mysql>
```

3-Creacion de tabla Sectores:

```
mysql> CREATE TABLE Sectores (  
->     id_sector INT AUTO_INCREMENT PRIMARY KEY,  
->     nombre VARCHAR(100),  
->     id_encargado INT  
-> );  
Query OK, 0 rows affected (0.05 sec)
```

4-Creacion de tabla Obras:

```
mysql> CREATE TABLE Obras (  
->     id_obra INT AUTO_INCREMENT PRIMARY KEY,  
->     nombre VARCHAR(100),  
->     id_sector INT  
-> );  
Query OK, 0 rows affected (0.04 sec)
```

5-Editamos la tabla Sectores para agregar las referencias y relaciones:

```
mysql> ALTER TABLE Sectores  
->     ADD CONSTRAINT fk_encargado  
->     FOREIGN KEY (id_encargado) REFERENCES Empleados(id_empleado);  
Query OK, 0 rows affected (0.10 sec)  
Records: 0  Duplicates: 0  Warnings: 0
```

6-Editamos la tabla Sectores para agregar las referencias y relaciones:

Field	Type	Null	Key	Default	Extra
id_obra	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	YES		NULL	
id_sector	int	YES		NULL	

7-Editamos la tabla Empleados para agregar las referencias y relaciones:

```
mysql> DESCRIBE Empleados;
```

Field	Type	Null	Key	Default	Extra
id_empleado	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	YES		NULL	
tipo_empleado	enum('efectivo','eventual','encargado')	YES		NULL	
id_sector	int	YES	MUL	NULL	
id_obra	int	YES	MUL	NULL	

Relaciones entre tablas:

- La tabla **Empleados** está relacionada con la tabla **Sectores** a través de la clave id_sector.
- La tabla **Empleados** también está relacionada con la tabla **Obras** a través de la clave id_obra.
- La tabla **Sectores** está relacionada con la tabla **Empleados** a través de la clave id_encargado, que indica quién es el encargado del sector.
- La tabla **Obras** está relacionada con la tabla **Sectores** a través de la clave id_sector, indicando a qué sector pertenece la obra.