

Documentación Primer Examen Parcial Base de Datos

Acceso a la base de datos

- En windows presione Windows + R
- Escriba cmd presione enter
- Ingrese a SQL escribiendo: mysql -u root -p
- Ingrese su contraseña
- Ingrese a la base de datos escribiendo: use biblioteca

Como ejecutar sus Scripts

Ya dentro de la base ejecute lo siguiente para cada consulta

1.

```
SELECT ejemplar.id_ejemplar, ejemplar.ubicacion_fisica, estado_libro.descripcion AS estado
FROM ejemplar
JOIN estado_libro ON ejemplar.id_estado = estado_libro.id_estado
WHERE ejemplar.id_libro = 2 AND estado_libro.descripcion = 'Disponible';
```

2.

```
SELECT libros.id_libro, libros.titulo, COUNT(ejemplar.id_ejemplar) AS cantidad_disponibles
FROM libros
LEFT JOIN ejemplar ON libros.id_libro = ejemplar.id_libro
LEFT JOIN estado_libro ON ejemplar.id_estado = estado_libro.id_estado
WHERE estado_libro.descripcion = 'Disponible'
GROUP BY libros.id_libro, libros.titulo;
```

3.

```
SELECT * FROM prestamo
WHERE prestamo.id_socio = 1;
```

4.

```
SELECT * FROM prestamo;
```

5.

```
SELECT ejemplar.id_ejemplar, libros.id_libro, libros.titulo, categorias.nombre_categoria
FROM libros
LEFT JOIN ejemplar ON libros.id_libro = ejemplar.id_libro
LEFT JOIN categorias ON categorias.id_categoria= libros.id_categoria
WHERE libros.id_categoria = 1;
```

Primero creamos la base de datos, ingresamos a la base de datos, creamos y eliminamos las tablas si ya existen insertar información lo más parecido a la información de los compañeros y autores. Creamos tabla estado de los libros donde tendremos: disponible, reservado, prestado y después relacionarla con la tabla libros creando la tabla que relaciona al autor con los libros, relacionando a cada autor con su libro.

1. Se creo una tabla para categorías

```
DROP TABLE IF EXISTS categorias;  
CREATE TABLE categorias(  
    id_categoria INT AUTO_INCREMENT,  
    nombre_categoria VARCHAR(30),  
    PRIMARY KEY(id_categoria)  
);
```

Se insertaron categorías para los libros

```
INSERT INTO categorias(nombre_categoria) VALUES  
    ('Ficción'),  
    ('No Ficción'),  
    ('Ciencia Ficción'),  
    ('Literatura Infantil'),  
    ('Biografía');
```

2. Se creo la tabla para la relación entre libros y categorías

```
DROP TABLE IF EXISTS categoria_libros;  
  
CREATE TABLE categoria_libros(  
    id_categoria INT NOT NULL,  
    id_libro INT NOT NULL,  
    PRIMARY KEY(id_categoria, id_libro),  
    FOREIGN KEY(id_categoria) REFERENCES categorias(id_categoria) ON DELETE  
    CASCADE,  
    FOREIGN KEY(id_libro) REFERENCES libros(id_libro) ON DELETE CASCADE  
);
```

Se crearon relaciones

```
INSERT INTO categoria_libros (id_categoria, id_libro) VALUES  
    (1, 1),  
    (2, 3),  
    (3, 4),  
    (4, 5);
```

3. Se agrego tabla para el estado de un libro

```
DROP TABLE IF EXISTS estado_libro;  
CREATE TABLE estado_libro(  
    id_estado INT AUTO_INCREMENT,  
    descripcion VARCHAR(20),  
    PRIMARY KEY(id_estado)  
);
```

Se agregaron los posibles estados en los que pueden estar un libro

```
INSERT INTO estado_libro(descripcion)  
VALUES
```

```
    ('Disponible'),  
    ('Prestado'),  
    ('Reservado');
```

4. Se modifiko la tabla anterior ejemplares, se agregaron columnas ubicacion_fisica, fecha_adquisicion, condicion.

```
ALTER TABLE ejemplar  
    ADD ubicacion_fisica VARCHAR(40) NOT NULL,  
    ADD fecha_adquisicion DATETIME NOT NULL,  
    ADD condicion VARCHAR(20);
```

5. Insertamos datos para ejemplares

```
INSERT INTO ejemplar (id_libro, id_estado, ubicacion_fisica, fecha_adquisicion, condicion) VALUES  
    (1, 1, 'Estante 3', '2024-01-15 10:00:00', 'Nuevo'),  
    (1, 2, 'Estante 4', '2024-01-15 10:00:00', 'Dañado'),  
    (2, 1, 'Estante 2', '2024-01-16 11:00:00', 'Usado'),  
    (3, 2, 'Estante 1', '2024-01-17 12:00:00', 'Dañado'),  
    (4, 1, 'Estante 5', '2024-01-18 13:00:00', 'Nuevo'),  
    (5, 1, 'Estante 4', '2024-01-19 14:00:00', 'Usado');
```

6. Creamos la relación libros_ejemplares

```
CREATE TABLE libros_ejemplares(  
    id_libro_ejemplar INT AUTO_INCREMENT,  
    id_libro INT NOT NULL,  
    id_ejemplar INT NOT NULL,  
    PRIMARY KEY(id_libro_ejemplar),  
    FOREIGN KEY (id_libro) REFERENCES libros(id_libro) ON DELETE CASCADE,  
    FOREIGN KEY (id_ejemplar) REFERENCES ejemplar(id_libro) ON DELETE CASCADE  
);
```

Creamos una relación entre los libros y ejemplares

```
INSERT INTO libros_ejemplares (id_libro, id_ejemplar)
VALUES
(1, 1),
(2, 2),
(3, 3);
```

7. Crear la tabla reserva libro

```
DROP TABLE IF EXISTS reservas;
```

```
CREATE TABLE reservas(
    id_reserva INT AUTO_INCREMENT,
    id_socio INT NOT NULL,
    id_libro INT NOT NULL,
    fecha_reserva DATETIME,
    fecha_expiracion_reserva DATETIME,
    estado_reserva VARCHAR(20),
    PRIMARY KEY(id_reserva,id_libro,id_socio),
    FOREIGN KEY(id_libro) REFERENCES libros(id_libro) ON DELETE CASCADE,
    FOREIGN KEY(id_socio) REFERENCES socio(id_socio) ON DELETE CASCADE
);
```

CONSULTA 1

-- Seleccionar su id,ubicacion y disponibilidad, combinar las tablas estado libro con ejemplar y usar su ID para filtrar resultados.

	id_ejemplar	ubicacion_fisica	estado
▶	3	Estante 2	Disponible

Result 18 × Result 19 prestamo 20 prestamo 21 Result 22

CONSULTA 2

Selecciones id del libro el titulo contamos los ejemplares, de la tabla libros, ingresamos a ejemplar combinado con libros, ingresamos al estado_libro combinado con ejemplar, condicionamos que este disponible, los agrupamos por su id y titulo y los mostramos. Seleccionamos de préstamo donde el id sea uno especifico

	id_libro	titulo	cantidad_disponibles
▶	1	Don Quijote de la Mancha	1
	2	La vida de Lazarillo de Tormes	1
	4	Los Juegos del Hambre	1
	5	En llamas	1

Result 18	Result 19 ×	prestamo 20	prestamo 21	Result 22
-----------	-------------	-------------	-------------	-----------

CONSULTA 3

Seleccionamos toda la información de la tabla préstamo dónde el id del socio coincida con el especificado.

	id_prestamo	id_libro	id_socio	fecha_prestamo	fecha_devolucion
▶	1	1	1	2024-10-30 16:43:21	NULL
	2	2	1	2024-10-30 16:43:21	NULL
*	NULL	NULL	NULL	NULL	NULL

Result 18 Result 19 **prestamo 20** × prestamo 21 Result 22

CONSULTA 4

Seleccionamos de prestamos todo el historial de prestamos que hayamos registrado.

	id_prestamo	id_libro	id_socio	fecha_prestamo	fecha_devolucion
▶	1	1	1	2024-10-30 16:43:21	NULL
	2	2	1	2024-10-30 16:43:21	NULL
	3	1	3	2024-10-30 16:43:21	NULL
*	NULL	NULL	NULL	NULL	NULL

CONSULTA 5

Seleccionar información a mostrar, seleccionar de libros, combinar categorías, ejemplar y libros, mostrar los libros de categoría con ID =1.

	id_ejemplar	id_libro	titulo	nombre_categoria
▶	1	1	Don Quijote de la Mancha	Ficción
	2	1	Don Quijote de la Mancha	Ficción
	NULL	6	Harry Potter y la piedra filosofal	Ficción
	NULL	7	Harry Potter y la cámara secreta	Ficción
	NULL	9	El viaje de las estrellas	Ficción

TRANSACCIONES DE CLASE

Usar procedimientos para comprobar las transacciones que estamos haciendo:

1. Agregar un nuevo libro y su autor
 - Eliminar procedimiento si existe.
 - Crear el procedimiento.
 - Comenzar el procedimiento.
 - Excepción: salir del procedimiento si ocurre un error.
 - Si ocurre la excepción: mostrar mensaje de error y realizar un rollback para no hacer los cambios en las tablas.
 - Terminar si hubo alguna excepción.
 - Si no hay ninguna excepción, comenzar con la transacción:
 - Iniciar la transacción.
 - Insertar una persona de tipo autor.
 - Insertar en la tabla autor la persona agregada anteriormente.
 - Insertar el libro con el ID del autor agregado anteriormente.
 - Mostrar mensaje de confirmación de operación.
 - Realizar los cambios.
 - Terminar el procedimiento.
2. Actualizar la información del libro "Don Quijote de la Mancha"
 - Eliminar procedimiento si existe.
 - Crear el procedimiento.
 - Comenzar el procedimiento.
 - Excepción: salir del procedimiento si ocurre un error.
 - Si ocurre la excepción: mostrar mensaje de error y realizar un rollback para no hacer los cambios en las tablas.
 - Terminar si hubo alguna excepción.
 - Si no hay ninguna excepción, comenzar con la transacción:
 - Iniciar la transacción.
 - Actualizamos libros
 - Actualizamos titulo libro
 - Donde el ID coincida
 - Mostrar mensaje de confirmación de operación.
 - Realizar los cambios.
 - Terminar el procedimiento.
3. Hacer un préstamo
 - Eliminar procedimiento si existe.
 - Crear el procedimiento.
 - Comenzar el procedimiento.
 - Excepción: salir del procedimiento si ocurre un error.
 - Si ocurre la excepción: mostrar mensaje de error y realizar un rollback para no hacer los cambios en las tablas.
 - Terminar si hubo alguna excepción.

- Si no hay ninguna excepción, comenzar con la transacción:
 - Iniciar la transacción.
 - Insertamos datos de préstamo
 - Actualizamos libros
 - Cambiamos estado donde coincida
 - Mostrar mensaje de confirmación de operación.
 - Realizar los cambios.
 - Terminar el procedimiento.
4. Hacer una devolución
- Eliminar procedimiento si existe.
- Crear el procedimiento.
 - Comenzar el procedimiento.
 - Excepción: salir del procedimiento si ocurre un error.
 - Si ocurre la excepción: mostrar mensaje de error y realizar un rollback para no hacer los cambios en las tablas.
 - Terminar si hubo alguna excepción.
 - Si no hay ninguna excepción, comenzar con la transacción:
 - Iniciar la transacción.
 - Actualizar la fecha de devolución del libro prestado
 - Cambiamos estado libro a disponible
 - Mostrar mensaje de confirmación de operación.
 - Realizar los cambios.
 - Terminar el procedimiento.
5. Eliminar autor

- Eliminar procedimiento si existe.
- Crear el procedimiento.
- Comenzar el procedimiento.
- Excepción: salir del procedimiento si ocurre un error.
- Si ocurre la excepción: mostrar mensaje de error y realizar un rollback para no hacer los cambios en las tablas.
- Terminar si hubo alguna excepción.
- Si no hay ninguna excepción, comenzar con la transacción:
 - Iniciar la transacción.
 - Eliminar autor dónde el ID coincida.
 - Mostrar mensaje de confirmación de operación.
 - Realizar los cambios.
 - Terminar el procedimiento.