

# Proyecto de Base de Datos Grupal BD

Fecha de entrega: 5 febrero 2025

## Instrucciones Generales

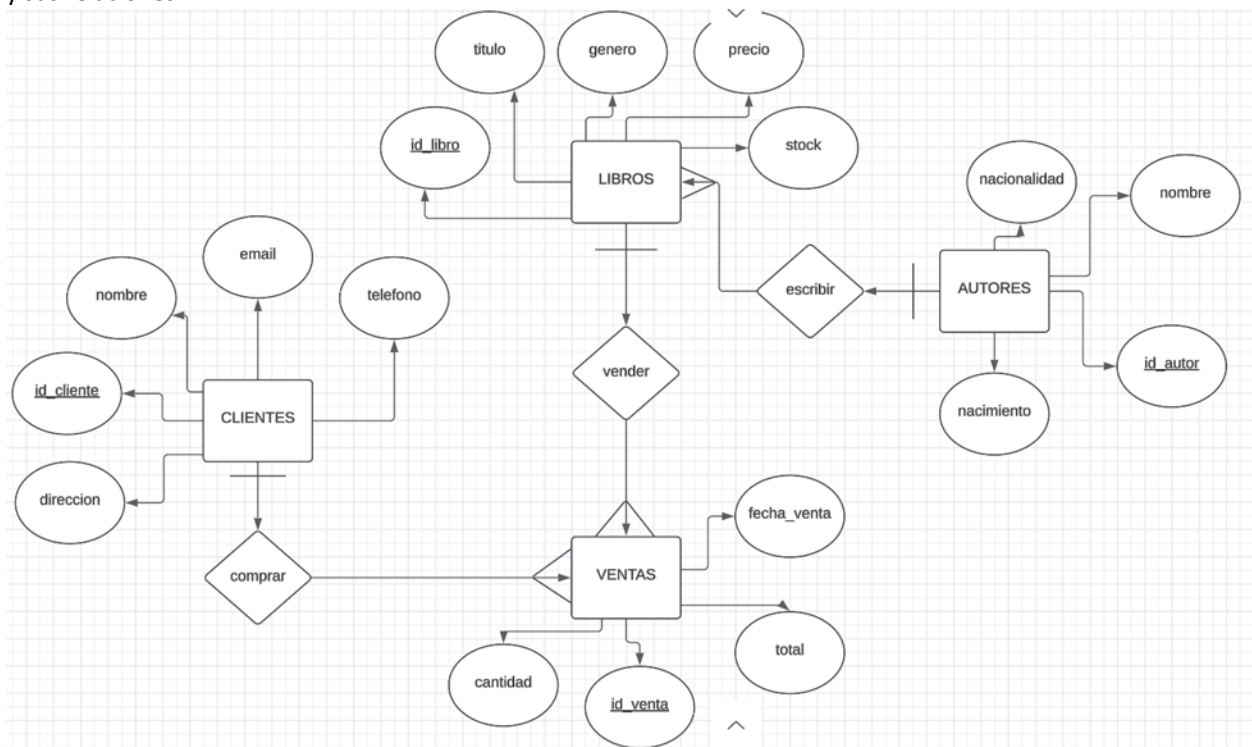
### 1. Modelado de Base de Datos y Diccionario de Datos

**Objetivo:** Crear un diseño eficiente y bien documentado para la base de datos, utilizando el modelado ER y un diccionario de datos completo.

**Actividades:**

Diseñar el modelo conceptual, lógico y físico.

**Práctica:** Crear un modelo entidad-relación que refleje las entidades clave (como Clientes, Vuelos, Reservas, Pagos) y sus relaciones.



Podemos ver que sus relaciones son de uno a muchos en estos casos:

- Un autor puede tener muchos libros.
- Un cliente puede tener muchas ventas.
- Un libro puede estar en muchas ventas.

**Investigación:** Buscar buenas prácticas sobre cómo hacer escalables los modelos de bases de datos.

De entre todos los métodos que existen para tener escalabilidad en una base de datos, principalmente tenemos que diseñar un buen esquema equilibrado, combinando la normalización para evitar redundancias, tener índices clave. Además, priorizar la escalabilidad horizontal, en otras palabras, tener la distribución de los datos en múltiples servidores.

También, aumentar la cache para acelerar las consultas frecuentes y optimizar consultas evitando operaciones costosas e implementar bases de datos NoSQL y opciones en la nube que lo hace escalable automáticamente.

**Importancia del Conocimiento:** Conocer cómo diseñar bases de datos adecuadas asegura la eficiencia y fácil mantenimiento de la aplicación.

### 1. Desarrollar un diccionario de datos detallado.

**Práctica:** Crear un diccionario que defina todas las tablas, sus campos, relaciones y restricciones.

DICCIONARIO DE DATOS					
TABLA CLIENTES					
Campo	Tipo de dato	Longitud	Descripción	Restricciones	Ejemplo
id_cliente	INT	11	Identificador unico del cliente	primary key, not null, auto-increment	100
nombre	VARCHAR	50	Nombre del cliente	not null	'Juan Pérez'
email	VARCHAR	50	Correo electronico del cliente	unique	'juan.perez@gmail.com'
telefono	VARCHAR	10	Numero de contacto	-	987654321
direccion	TEXT	-	Direccion del cliente	-	'Av. Central 123'
TABLA AUTORES					
Campo	Tipo de dato	Longitud	Descripción	Restricciones	Ejemplo
id_autor	INT	11	Identificador unico del autor	primary key, not null, auto-increment	200
nombre	VARCHAR	50	Nombre del autor	not null	'Gabriel García Márquez'
nacionalidad	VARCHAR	15	Nacionalidad del autor	not null	'Colombiana'
fecha_nacimiento	DATE	-	Fecha de nacimiento del autor	-	'1927-03-06'
TABLA LIBROS					
Campo	Tipo de dato	Longitud	Descripción	Restricciones	Ejemplo
id_libro	INT	11	Identificador unico del libro	primary key, not null, auto-increment	300
titulo	VARCHAR	50	Nombre del libro	not null	'Cien años de soledad'
genero	VARCHAR	50	Genero del libro	not null	'Novela'
precio	DECIMAL	99,99	Precio del libro	(4,2), not null	15.99
id_autor	INT	11	Identificador del autor	foreign key, delete cascade, update cascade	100
stock	INT	11	Cantidad de existencias de libros	default	10

Por ser un proyecto pequeño y no disponía de ayudas de programas experimentados preferí usar el método manual.

**Investigación:** Investigar las mejores herramientas y métodos para generar diccionarios de datos y su uso en proyectos reales.

Se tiene una gran variedad de métodos y herramientas para generar diccionarios:

- Directamente de MySQL con esquemas de bases de datos.
- Existen programas especializados como Dataedo que crea diccionarios interactivos.
- Por medio de codificación usando las librerías pymysql o sqlalchemy, su beneficio es que puede actualizar si hay un cambio en la base de datos.
- Manualmente por medio de una tabla en excel es usado cuando son proyectos pequeños y no se dispone de programas experimentados.

Sus usos son:

- Automatiza la generación con herramientas o scripts.
- Mantén la documentación actualizada.
- Usa estándares para nombres y descripciones.

- Colabora con stakeholders para capturar el contexto de los datos.

**Importancia del Conocimiento:** Un diccionario de datos permite mantener la consistencia y facilita la colaboración entre desarrolladores.

## 2. Definir las restricciones de integridad referencial y eliminación - update.

**Práctica:** Establecer claves primarias y foráneas entre las tablas, asegurando la coherencia de los datos (por ejemplo, ClienteID debe estar presente en las tablas relacionadas).



**Eliminación – Update de cascada, set null, restrict, Constrains (NO ACTION)**

```

19 • create table libros(
20     id_libro int auto_increment primary key,
21     titulo varchar (20) not null,
22     genero varchar (15) not null,
23     precio decimal (4,2) not null,
24     id_autor int,
25     stock int default 0,
26     foreign key (id_autor) references autores (id_autor)
27     on update cascade -- si se actualiza el autor se cambia sus libros
28     on delete cascade -- si se elimina el autor se eliminan sus libros
29 );
30
31 • create table ventas(
32     id_venta int auto_increment primary key,
33     id_cliente int,
34     id_libro int,
35     fecha_venta date not null,
36     cantidad int not null,
37     total decimal (5,2),
38     foreign key (id_cliente) references clientes (id_cliente)
39     on delete set null -- si un cliente se elimina se deja el valor en null en la tabla ventas
40     on update cascade, -- si se cambia el cliente se actualiza en la tabla ventas
41     foreign key (id_libro) references libros (id_libro)
42     on delete restrict -- no se permite eliminar libros si tiene ventas asociadas
43     on update cascade -- si se cambia el libro cambia en la tabla ventas tambien
44 ) auto_increment = 100;

```

Con delete set null nos aseguramos de que si eliminamos un registro en la tabla autores se eliminará en la tabla libro.

Así mismo, con update cascade sirve para que si en algún momento se cambia un registro de autores cambiara en la tabla libros.

Delete restrict nos dice que no se permite eliminar libros si tiene valores asociados. Con update cascade tiene el mismo uso que el anterior de la tabla libros, pero en ventas ahora.

**Investigación:** Investigar cómo la integridad referencial puede prevenir la pérdida de datos y mantener la consistencia.

Para prevenir la perdida de datos usamos cascade, set null, restrict. Además tenemos en cuentas que:

- Impedir que existan registro huérfanos.
- Retringir modificaciones accidentales o erróneas.

**Importancia del Conocimiento:** Las restricciones de integridad aseguran que los datos no se corrompan.

## 2. Seguridad, Auditoría y Control de Acceso

**Objetivo:** Proteger los datos sensibles y controlar el acceso a la base de datos.

**Actividades:**

### 1. Implementar políticas de acceso y seguridad.

**Práctica:** Crear roles y permisos de usuario para controlar el acceso a las tablas y vistas.

```

create role administrador;
create role usuario;
create role auditor;

grant all privileges on libreria.* to administrador;

grant select on libreria.* to usuario;
grant insert on libreria.ventas to usuario;

grant select on libreria.* to auditor;

create user 'administrador_libreria'@'%' identified by '123';
create user 'usuario_libreria'@'%' identified by '456';
create user 'auditor_libreria'@'%' identified by '789';

grant administrador to 'administrador_libreria'@'%';
grant usuario to 'usuario_libreria'@'%';
grant auditor to 'auditor_libreria'@'%';

set default role administrador to 'administrador_libreria'@'%';
set default role usuario to 'usuario_libreria'@'%';
set default role auditor to 'auditor_libreria'@'%';

set role administrador;
set role usuario;
set role auditor;

```

- Primero creamos los roles para luego seguir para otorgar privilegios.
- Ahora creamos los usuarios con el nombre y la contraseña de cada usuario para luego guardar los roles y usuarios.
- Finalmente activamos los roles.

**Investigación:** Investigar sobre los mejores enfoques para la seguridad en bases de datos en entornos de alta disponibilidad.

Existen una gran variedad de métodos para la seguridad, estos son algunos:

- El uso de usuarios y roles para controlar el acceso a la base de datos.
- El cifrado de datos sensibles para guardar la confidencialidad de los datos.
- Monitoreo y auditoria para tener un registro de quien ingresa a la base de datos.
- Tener respaldos actualizados de los registros.

**Importancia del Conocimiento:** El control adecuado de acceso previene fugas de información y mejora la seguridad general.

## 2. Cifrado de datos sensibles.

**Práctica:** Cifrar información sensible como contraseñas y detalles de pago (por ejemplo, usando AES\_ENCRYPT en MySQL).

```

use libreria;

alter table clientes add column contraseña varbinary(255);

-- CIFRAR
insert into clientes (nombre, email, telefono, direccion, contrasena)
values (
    'Juan Pérez', 'juan@example.com', '1234567890', 'Calle Falsa 123',
    AES_ENCRYPT('contraseña', 'proyecto')
);

-- DESCIFRAR
select
    id_cliente,
    nombre,
    email,
    telefono,
    direccion,
    AES_DECRYPT(contrasena, 'proyecto') as contrasena_descifrada
from clientes
where id_cliente = 100;

alter table ventas add column detalle_pago varbinary(255);

insert into ventas (id_cliente, id_libro, fecha_venta, cantidad, total, detalles_pago)
values (
    100, 300, '2023-10-01', 2, 25.50,
    AES_ENCRYPT('{\"tarjeta\":\"1234-5678-9012-3456\", \"exp\":\"12/25\"}', 'llave_secreta')
);

```

- Se agregan columnas cifradas VARBINARY para almacenar contraseñas y detalles de pago.
- Se usa AES\_ENCRYPT() para cifrar los datos antes de almacenarlos.
- Se usa AES\_DECRYPT() para descifrar los datos al consultarlos.
- Se aplican claves de cifrado ('proyecto' y 'llave\_secreta'), las cuales deben ser seguras y gestionadas adecuadamente.

**Investigación:** Explorar algoritmos de cifrado y su impacto en el rendimiento de la base de datos. Existen dos principales cifrados:

- Cifrado Simétrico:  
Utiliza la misma clave para cifrar y descifrar.
- Cifrado Asimétrico:  
Utiliza un par de claves (pública y privada).  
Generalmente más seguro, pero más lento que el cifrado simétrico.

Su impacto consta de:

- Al generar contraseñas cifradas genera más memoria y recursos.
- Ayuda a la seguridad e integridad de registros o datos sensible del usuario.

**Importancia del Conocimiento:** El cifrado es esencial para la protección de la información confidencial de los usuarios.

### 3. Habilitar auditoría y registrar eventos de base de datos.

**Práctica:** Activar los logs de acceso y auditoría para monitorear las actividades de los usuarios (por ejemplo, registrar quién accedió a qué datos).

```

C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u usuario_libreria -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use libreria;
Database changed
mysql>

C:\Users\User>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u administrador_libreria -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use libreria;

C:\Users\User>cd C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u auditor_libreria -p
Enter password: ***
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 8.0.41 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use libreria;
ERROR 1044 (42000): Access denied for user 'auditor_libreria'@'%' to database

```

Id	User	Host	DB	Command	Time	State	Threa...
5	event_sched...	localhost	None	Daemon	4199	Waiting on e...	43
7	None	None	None	Daemon	4199	Suspending	46
17	root	localhost	libreria	Sleep	26	None	57
18	root	localhost	libreria	Sleep	26	None	58
26	root	localhost	None	Query	0	executing	66
27	root	localhost	None	Sleep	1	None	67
36	administrad...	localhost	libreria	Sleep	2787	None	76
37	usuario_libre...	localhost	libreria	Sleep	2513	None	77
38	auditor_libre...	localhost	libreria	Sleep	2469	None	78

Para poder configurar monitoreo de acceso en MySQL seguimos estos sencillos pasos:

- Ingresar al commant line de mysql ingresamos con nuestro usuario administrador, para activar los usuarios de monitoreo.
- Vamos al terminal e ingresamos a C:\Program Files\MySQL\MySQL Server 8.0\bin.
- Luego ingresamos a nuestros usuarios mysql -u administrador\_libreria -p.
- Finalmente ponemos use libreria en este caso.
- Vamos a mysql y seleccionamos client connection y veremos que usuario ingreso y a que base de datos.

**Investigación:** Buscar cómo configurar herramientas de auditoría en MySQL o PostgreSQL.

- Ingresar al commant line de mysql ingresamos con nuestro usuario administrador, para activar los usuarios de monitoreo.
- Vamos al terminal e ingresamos a C:\Program Files\MySQL\MySQL Server 8.0\bin.
- Luego ingresamos a nuestros usuarios mysql -u administrador\_libreria -p.
- Finalmente ponemos use libreria en este caso.
- Vamos a mysql y seleccionamos client connection y veremos que usuario ingreso y a que base de datos.

**Importancia del Conocimiento:** La auditoría permite rastrear cambios en los datos y detectar actividades sospechosas.

### 3. RespalDOS y Recuperación de Datos

**Objetivo:** Asegurar la integridad y disponibilidad de los datos mediante técnicas de respaldo confiables.

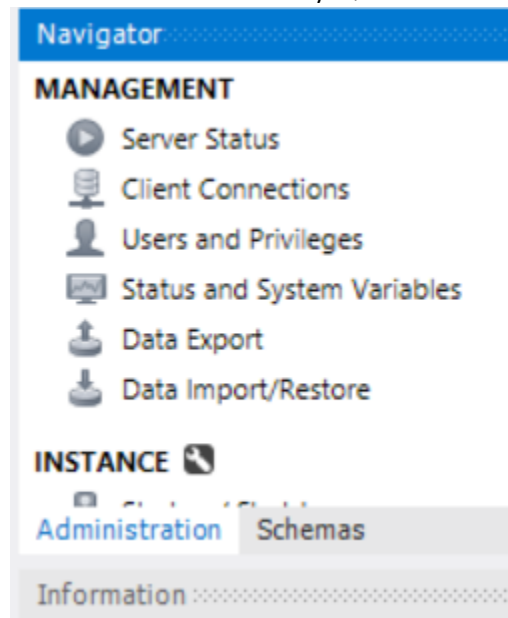
**Actividades:**

1. **Crear respaldos completos (full backups).**

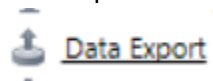
**Práctica:** Utilizar mysqldump o herramientas similares para hacer respaldos completos de la base de datos.

Para poder hacer un respaldo con mysqldump a nuestra base de datos vamos a hacer lo siguiente:

- Primero vamos a la parte de “Administration” en MySQL Workbench

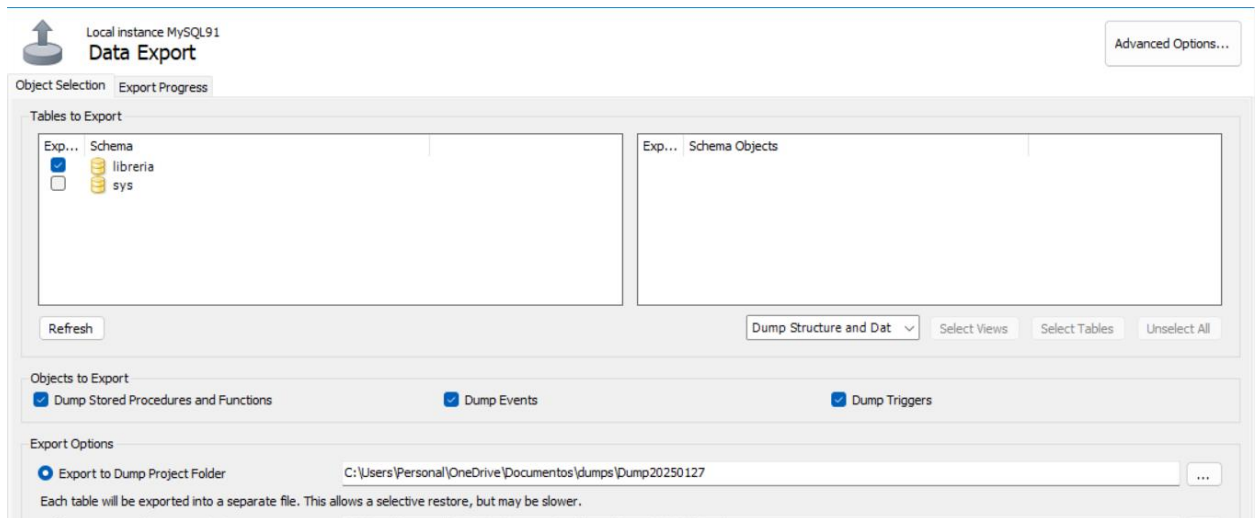


- Luego elegimos la opción de Data Export:



- Al elegir esta opción se abre esta interfaz donde vamos a poder elegir lo que queremos guardar de nuestra base de datos:



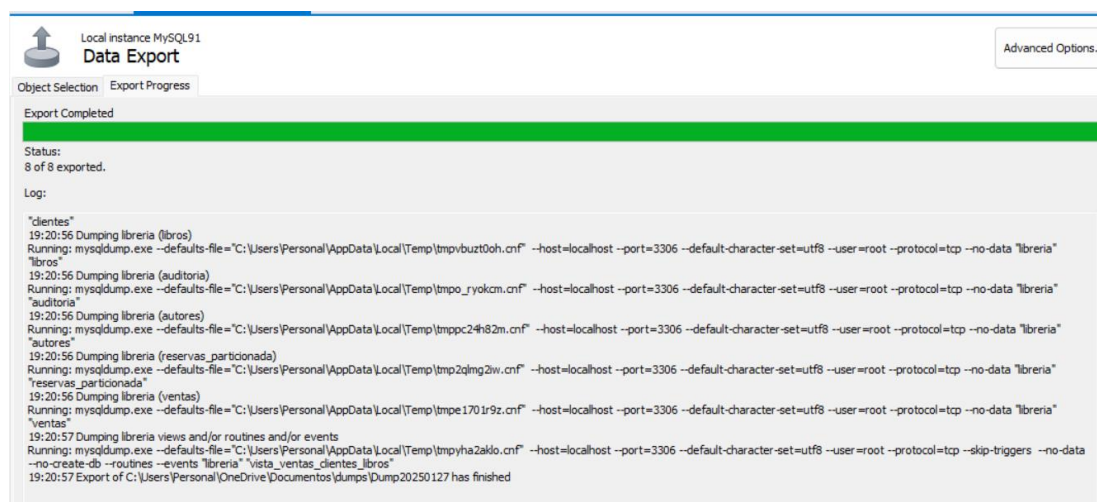


Vamos a analizar un poco esta imagen, como podemos ver se elige la base de datos llamada “libreria” luego en la parte de abajo de los cuadros esta seleccionada la opción de “Dump Structure and Data” que nos ayuda a poder guardar todo lo que se encuentra en la base de datos, y en el final hay tres casillas que esta activadas que son las de “Dump Stored Procedures and Functions”, “Dump Events”, “Dump Triggers” que nos ayudan a poder guardar los procedimientos, funciones, eventos y triggers.

Al finalizar esta la ruta donde el dump se va a guardar, para eso le damos en “Start Export” y esperamos.

Start Export

- Aquí podemos ver que se ha creado el dump:



Aquí está la carpeta donde están todos los datos:

Path: > Jordy - Personal > Documentos > dumps > Dump20250127

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
libreria_auditoria	↻	27/01/2025 19:20	SQL Text File	2 KB
libreria_autores	↻	27/01/2025 19:20	SQL Text File	2 KB
libreria_clientes	↻	27/01/2025 19:20	SQL Text File	2 KB
libreria_libros	↻	27/01/2025 19:20	SQL Text File	2 KB
libreria_reservas	↻	27/01/2025 19:20	SQL Text File	5 KB
libreria_reservas_particionada	↻	27/01/2025 19:20	SQL Text File	3 KB
libreria_routines	↻	27/01/2025 19:20	SQL Text File	6 KB
libreria_ventas	↻	27/01/2025 19:20	SQL Text File	5 KB

- Existe otra opción de poder hacer un dump y esa es guardándole en un solo archivo y solo se debe hacer un cambio cuando estamos seleccionando lo que queremos respaldar debemos seleccionar esta opción:

Export to Self-Contained File C:\Users\Personal\OneDrive\Documentos\dumps\Dump20250127.sql

All selected database objects will be exported into a single, self-contained file.

☐ Create Dump in a Single Transaction (self-contained file only) ☒ Include Create Schema

Export Completed Start Export

- Aquí esta como queda el archivo:

Path: > Jordy - Personal > Documentos > dumps >

Nombre	Estado	Fecha de modificación	Tipo
Dump20250127	↻	27/01/2025 19:20	Carpeta de
Dump20250127	↻	27/01/2025 19:26	SQL Text File

#### Recuperación de datos

- Para esto vamos a utilizar el dump que generamos antes y en la parte de “Administration” vamos a escoger esta opción:

## Data Import/Restore

- Le damos click y llenamos lo que necesitamos:

Administration - Data Import/Restore

Local instance MySQL91  
**Data Import**

Import from Disk Import Progress

**Import Options**

☒ Import from Dump Project Folder C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127 ...

Select the Dump Project Folder to import. You can do a selective restore.

☐ Import from Self-Contained File C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127.sql ...

Select the SQL/dump file to import. Please note that the whole file will be imported.

**Default Schema to be Imported To**

Default Target Schema:  New...

The default schema to import the dump into.  
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

**Select Database Objects to Import (only available for Project Folders)**

Imp...	Schema
<input checked="" type="checkbox"/>	libreria

8 tables selected

Imp...	Schema Objects
<input checked="" type="checkbox"/>	Views, routines, events etc
<input checked="" type="checkbox"/>	auditoria
<input checked="" type="checkbox"/>	autores
<input checked="" type="checkbox"/>	clientes
<input checked="" type="checkbox"/>	libros
<input checked="" type="checkbox"/>	reservas
<input checked="" type="checkbox"/>	reservas_particionada
<input checked="" type="checkbox"/>	ventas

Press [Start Import] to start...

- Le damos "Start import" y esperamos:

Administration - Data Import/Restore

Local instance MySQL91  
**Data Import**

Import from Disk Import Progress

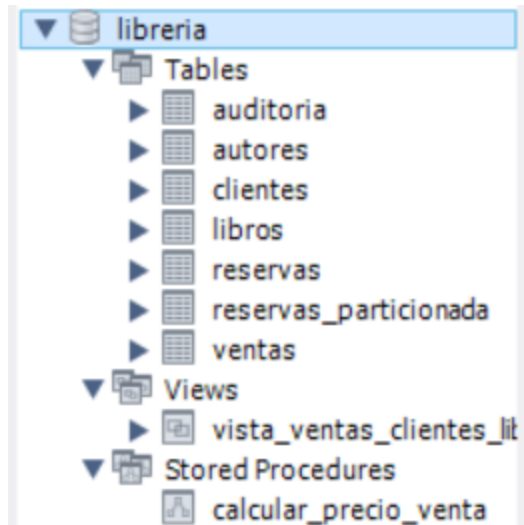
**Import Completed**

Status:  
8 of 8 imported.

Log:

```
19:33:53 Restoring libreria (autores)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmpkji-48m.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_autores.sql"
19:33:53 Restoring libreria (clientes)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmp6t8a6vgg.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_clientes.sql"
19:33:54 Restoring libreria (libros)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmp6476lp_y.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_libros.sql"
19:33:54 Restoring libreria (reservas)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmpbdf5f2l1.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_reservas.sql"
19:33:54 Restoring libreria (reservas_particionada)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmp7cu6scowr.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_reservas_particionada.sql"
19:33:54 Restoring libreria (ventas)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmpkdzsvezn.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_ventas.sql"
19:33:55 Restoring libreria (Views, routines, events etc)
Running: mysql.exe --defaults-file="C:\Users\Personal\AppData\Local\Temp\lmpqenj6f0t.cnf" --protocol=tcp --host=localhost --user=root --port=3306 --default-character-set=utf8 --comments --database=libreria < "C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127\libreria_routines.sql"
19:33:55 Import of C:\Users\Personal\OneDrive\Documents\dumps\Dump20250127 has finished
```

- Y podemos ver que se restauró de nuevo la base de datos:



**Investigación:** Buscar estrategias de respaldo para bases de datos de gran tamaño y la mejor manera de gestionarlas.

### Estrategias de Respaldo para Bases de Datos de Gran Tamaño

#### Estrategias de Almacenamiento

- **Almacenamiento Externo:** Utilizar almacenamiento en la nube o físico fuera del sitio garantiza que los datos estén protegidos incluso durante desastres locales.
- **Replicación:** Mantener copias sincronizadas en diferentes ubicaciones permite una recuperación rápida si la base de datos principal falla. La replicación puede ser sincrónica (para coherencia) o asincrónica (para rendimiento).

#### Automatización y Gestión

- **Automatización del Proceso:** Implementar soluciones que automaticen el respaldo y la recuperación simplifica la gestión y reduce el riesgo de error humano.
- **Pruebas Periódicas:** Realizar pruebas regulares del proceso de restauración asegura que los respaldos sean efectivos y que se puedan recuperar los datos sin problemas.

#### Estrategia 3-2-1

- **Tres Copias:** Mantener tres copias de los datos (una original y dos respaldos).
- **Dos Medios Diferentes:** Utilizar al menos dos tipos diferentes de almacenamiento (por ejemplo, disco duro y almacenamiento en la nube).
- **Una Copia Fuera del Sitio:** Asegurarse de que al menos una copia esté almacenada fuera del sitio físico para protegerse contra desastres locales.

#### Seguridad en el Respaldo

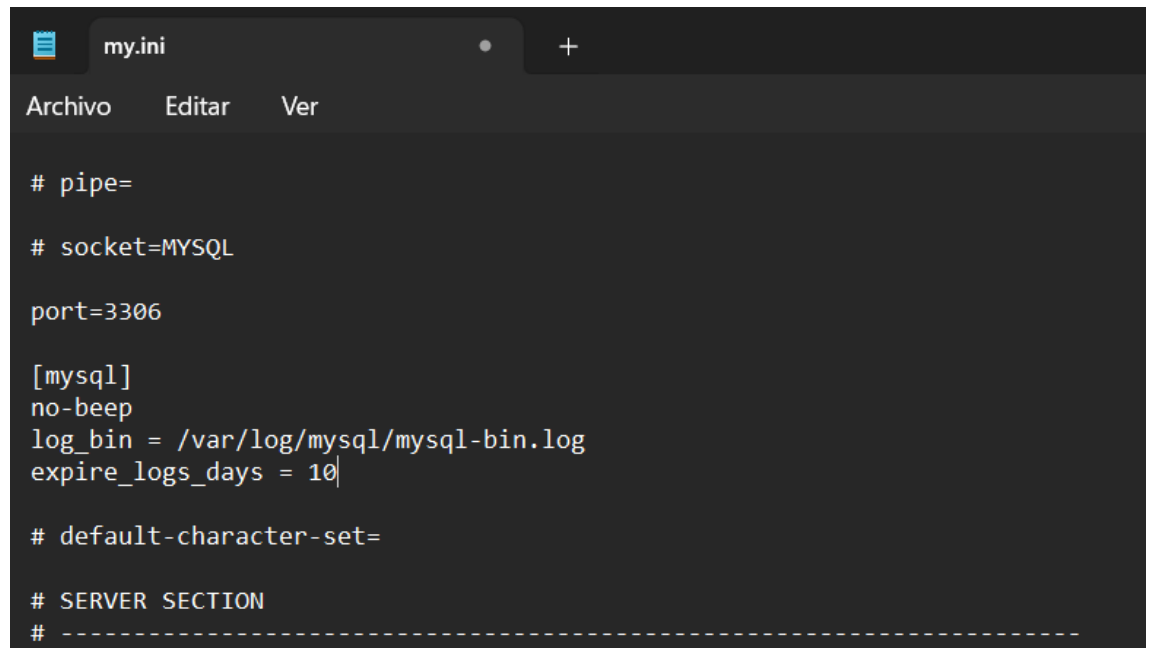
- **Cifrado:** Implementar cifrado en los respaldos es fundamental para proteger los datos contra accesos no autorizados. Utilizar claves fuertes y rotarlas regularmente mejora la seguridad.
- **Planificación ante Desastres:** Desarrollar un plan integral que incluya procedimientos claros para la recuperación ante desastres ayuda a minimizar el tiempo de inactividad y a garantizar la continuidad del negocio.

**Importancia del Conocimiento:** Los respaldos completos permiten restaurar toda la base de datos ante una falla.

## 2. Configurar respaldos incrementales.

**Práctica:** Realizar respaldos incrementales para reducir el tiempo y espacio de almacenamiento.

- Primero, necesitamos habilitar el registro binario en el servidor MySQL. Esto es crucial para poder realizar copias de seguridad incrementales.



```
my.ini
Archivo  Editar  Ver

# pipe=

# socket=MYSQL

port=3306

[mysql]
no-beep
log_bin = /var/log/mysql/mysql-bin.log
expire_logs_days = 10

# default-character-set=

# SERVER SECTION
# -----
```

Esto configurará el registro binario y eliminará los logs más antiguos después de 10 días

- Podemos comenzar a realizar respaldos incrementales. Para esto, utilizamos el comando `mysqlbinlog` junto con los logs binarios generados por MySQL. Los pasos son los siguientes:

1.- Identificar el último log binario que se generó después del respaldo completo.

```

Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW BINARY LOGS;
+-----+-----+-----+
| Log_name          | File_size | Encrypted |
+-----+-----+-----+
| DESKTOP-QI3OJS0-bin.000001 |      181 | No        |
| DESKTOP-QI3OJS0-bin.000002 |    39889 | No        |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |

```

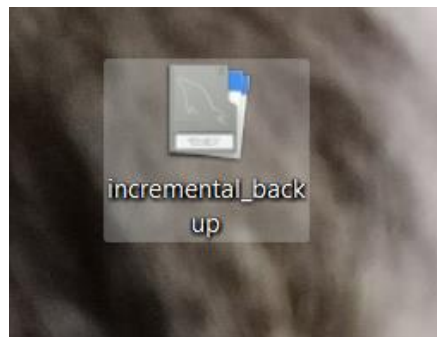
2.- Ejecutar el siguiente comando para crear un respaldo incremental:

```

mysql> mysqlbinlog --start-position=12345 C:\ProgramData\MySQL\MySQL Server 9.2\Data\DESKTOP-QI3OJS0-bin.000002 > C:\Users\Personal\OneDrive\Desktop\incremental_backup.sql

```

- Aquí podemos ver el archivo en el escritorio:



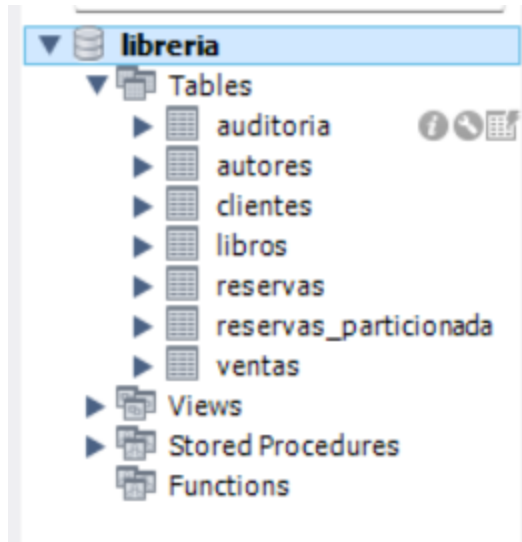
Restauramos con este comando:

```

C:\Users\Personal>mysql -u root -p < C:\Users\Personal\OneDrive\Desktop\incremental_backup.sql

```

- Aquí podemos ver cómo se restauró:



**Investigación:** Investigar cómo realizar respaldos incrementales y cuándo es más conveniente utilizarlos.

#### ¿Cómo realizarlos?

- **Respaldo Completo Inicial:** Haz un respaldo completo como base.
- **Configura el Incremental:** Usa herramientas como Cobian Backup o software de bases de datos para programar respaldos incrementales.
- **Ejecuta y Verifica:** Realiza los respaldos según la programación y revisa que se completen correctamente.

#### ¿Cuándo usarlos?

- **Frecuentes Cambios:** Bases de datos con actualizaciones constantes.
- **Espacio Limitado:** Cuando el almacenamiento es escaso.
- **Eficiencia:** Para respaldos rápidos y optimizados.

Son ideales para mantener copias actualizadas sin duplicar datos innecesarios.

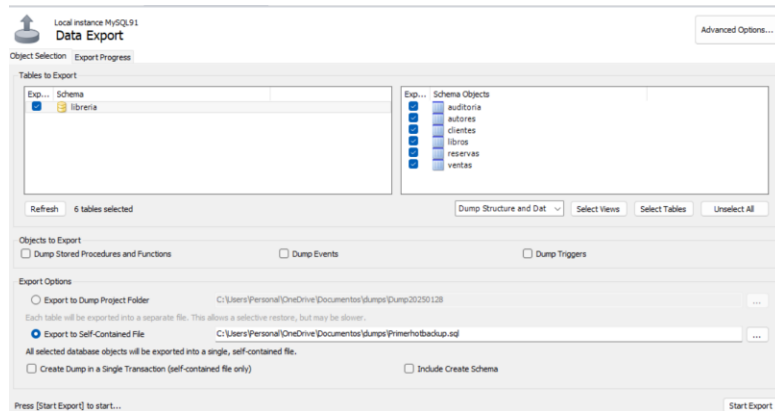
**Importancia del Conocimiento:** Los respaldos incrementales permiten optimizar los recursos y acelerar los tiempos de recuperación.

### 3. Implementar respaldos en caliente (Hot Backups).

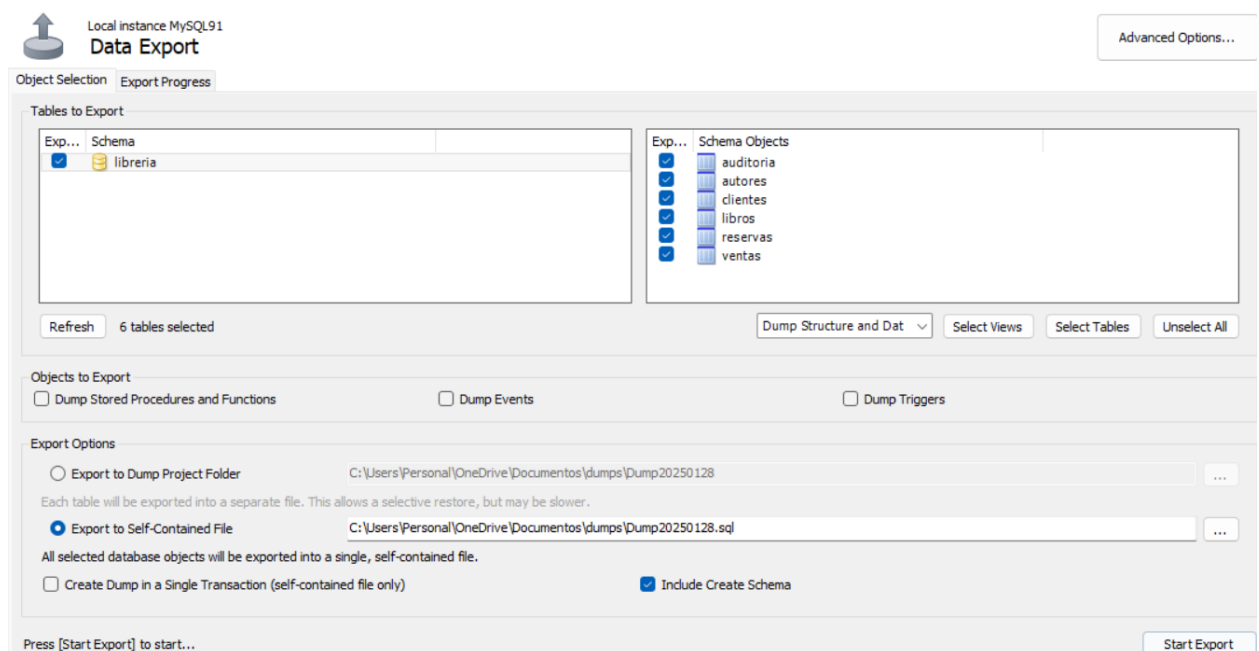
**Práctica:** Hacer respaldos sin interrumpir el servicio (por ejemplo, usando Percona XtraBackup).

Para realizar hot backups en MySQL Workbench, es importante entender que este tipo de copias de seguridad se llevan a cabo mientras la base de datos está en funcionamiento, permitiendo que los usuarios continúen accediendo a los datos.

Para poder hacer estos “Hot Backups” podemos hacerlo por medio de Workbench directamente como por ejemplo ya tenemos algunas cosas creadas en nuestra base de datos y utilizamos la herramienta “Export data” para crear el primer Hot backup con las tablas creadas:



Ahora podemos continuar trabajando en nuestra base de datos y hacer otro backup:



En este caso hemos guardado las tablas, pero también con los registros

Ingresados y así podemos ver que podemos hacer los hot backups y podemos identificarlos para saber de qué se trata:

hotbackup2	🔄	28/01/2025 8:19	SQL Text File	9 KB
Primerhotbackup	🔄	28/01/2025 8:18	SQL Text File	7 KB

**Investigación:** Investigar cómo hacer respaldos sin detener la base de datos.

**Respaldos en Caliente**



Permiten realizar copias de seguridad mientras la base de datos está activa. Esto se logra utilizando herramientas que soportan respaldos en caliente, como:

- MySQL: Utiliza la opción mysqldump con la opción --single-transaction para realizar un respaldo sin bloquear las tablas.
- SQL Server: Permite respaldos en caliente mediante el uso de SQL Server Management Studio (SSMS) o comandos T-SQL.

**Importancia del Conocimiento:** Los respaldos en caliente son esenciales para bases de datos de producción que no pueden permitirse inactividad.

#### 4. Optimización y Rendimiento de Consultas

**Objetivo:** Mejorar la eficiencia en la recuperación de datos mediante la optimización de consultas y el uso adecuado de índices.

##### Actividades:

##### 1. Crear y gestionar índices.

**Práctica:** Implementar índices en las columnas más consultadas, como VueloID, ClienteID, etc. Crear dos índices importantes

**Investigación:** Investigar sobre los tipos de índices más adecuados para bases de datos transaccionales y cómo afectan el rendimiento.

**Importancia del Conocimiento:** Los índices son cruciales para acelerar las consultas y mejorar el rendimiento general de la base de datos.

```
221 -- Índice en la columna id_cliente
222 • CREATE INDEX idx_ventas_id_cliente ON ventas (id_cliente);
223
224 -- Índice en la columna id_libro
225 • CREATE INDEX idx_ventas_id_libro ON ventas (id_libro);
226
227 -- Índice en la columna fecha_venta
228 • CREATE INDEX idx_ventas_fecha_venta ON ventas (fecha_venta);
229
230 -- Índice en la columna id_cliente
231 • CREATE INDEX idx_reservas_id_cliente ON reservas (id_cliente);
232
233 -- Índice en la columna fecha_reserva
234 • CREATE INDEX idx_reservas_fecha_reserva ON reservas (fecha_reserva);
235
```

Estos índices creados optimizan y mejoran el rendimiento de las consultas que filtran o agrupan por id\_cliente, id\_libro, fecha\_venta y fecha\_reserva.

Podemos observar los índices creados con.

```

236 -- Verificar índices
237 • SHOW INDEX FROM ventas;
238 • SHOW INDEX FROM reservas;

```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
reservas	0	PRIMARY	1	id_reserva	A	5				BTREE			YES	
reservas	1	id_libro	1	id_libro	A	5			YES	BTREE			YES	
reservas	1	idx_reservas_id_cliente	1	id_cliente	A	5			YES	BTREE			YES	
reservas	1	idx_reservas_fecha_reserva	1	fecha_reserva	A	5				BTREE			YES	

Tipos de índices más adecuados para bases de datos transaccionales:

- Índice B-tree: son eficientes para operaciones de búsqueda, inserción y eliminación.
- Índice Hash: Son más adecuados para búsquedas exactas y no son útiles para búsquedas por rango.
- Índices compuestos: Útiles cuando se realizan consultas que filtran o agrupan por varias columnas.
- Índices únicos: Se utilizan comúnmente en claves primarias y otras columnas donde se requiere unicidad.

## 2. Optimizar consultas SQL.

**Práctica:** Utilizar herramientas como EXPLAIN para identificar cuellos de botella en las consultas y optimizarlas.

**Practica: Aplicación de 3 join**

**Investigación:** Investigar cómo hacer uso eficiente de las uniones (JOIN), subconsultas, y optimizar las consultas complejas.

**Importancia del Conocimiento:** Las consultas optimizadas aseguran un sistema rápido y eficiente, especialmente en sistemas con alta demanda.

```

251 SELECT
252     v.id_venta, c.nombre AS cliente, l.titulo AS libro, v.cantidad, v.total
253 FROM
254     ventas v
255 JOIN
256     clientes c ON v.id_cliente = c.id_cliente
257 JOIN
258     libros l ON v.id_libro = l.id_libro
259 WHERE
260     v.total > 50;

```

id_venta	cliente	libro	cantidad	total
413	Ana Sánchez	Harry Potter y la piedra filosofal	3	60.00

Primero creamos una Consulta que une tres tablas que son ventas, clientes y libros y muestra la información solicitada de estas

```

249 -- uso de explain
250 • EXPLAIN
251 SELECT
252     v.id_venta, c.nombre AS cliente, l.titulo AS libro, v.cantidad, v.total
253 FROM
254     ventas v
255 JOIN
256     clientes c ON v.id_cliente = c.id_cliente
257 JOIN
258     libros l ON v.id_libro = l.id_libro
259 WHERE
260     v.total > 50;
261
262
263

```

Result Grid										
Filter Rows:										
Export: Wrap Cell Content:										
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	Extra
▶	1	SIMPLE	v	HULL	ALL	idx_ventas_id_cliente,idx_ventas_id_libro	HULL	HULL	HULL	5 33.33 Using where
	1	SIMPLE	c	HULL	eq_ref	PRIMARY	PRIMARY	4	libreria.v.id_cliente	1 100.00 HULL
	1	SIMPLE	l	HULL	eq_ref	PRIMARY	PRIMARY	4	libreria.v.id_libro	1 100.00 HULL

El comando EXPLAIN se utiliza antes de la consulta para obtener información sobre cómo el motor de base de datos ejecutará la consulta. Proporciona detalles sobre:

- Tipo de unión (JOIN): Indica cómo se unirán las tablas.
- Número estimado de filas: Muestra cuántas filas se espera que se procesen.
- Uso de índices: Informa si se utilizarán índices para optimizar la búsqueda.
- Costo estimado: Proporciona una estimación del costo asociado con la ejecución de la consulta.

Para optimizar el uso de uniones (JOIN) y subconsultas en consultas complejas, se utiliza el tipo de JOIN adecuado según las necesidades y se considera subconsultas para resultados intermedios. Primero se asegúrate de que las columnas en las condiciones de unión estén indexadas para mejorar el rendimiento. Además, se emplea el comando EXPLAIN para analizar la ejecución de las consultas y ajustar índices o estructuras según sea necesario, lo que puede aumentar significativamente la eficiencia general.

### 3. Utilizar particionamiento de tablas.

**Práctica:** Dividir tablas grandes, como Reservas, en particiones según una clave (por ejemplo, por fecha).

**Investigación:** Investigar sobre los beneficios del particionamiento y cómo implementarlo en sistemas de bases de datos grandes.

**Importancia del Conocimiento:** El particionamiento de tablas mejora la escalabilidad y el rendimiento en bases de datos con gran volumen de datos.

```

65  -- particion de tabla
66  ● ⊖ CREATE TABLE reservas_particionada (
67      id_reserva INT NOT NULL,
68      id_cliente INT,
69      fecha_reserva DATE NOT NULL,
70      cantidad INT NOT NULL,
71      total DECIMAL(5,2),
72      PRIMARY KEY (id_reserva, fecha_reserva)
73  )
74  ⊖ PARTITION BY RANGE (YEAR(fecha_reserva)) (
75      PARTITION p_2023 VALUES LESS THAN (2024),
76      PARTITION p_2024 VALUES LESS THAN (2025),
77      PARTITION p_2025 VALUES LESS THAN (2026),
78      PARTITION p_max VALUES LESS THAN MAXVALUE
79  );

```

La tabla está particionada por rango (RANGE) basado en el año de la columna fecha\_reserva. Esto significa que las filas se agrupan en particiones según el año en que se realizó la reserva.

- PARTITION p\_2023: Almacena todas las reservas con una fecha\_reserva antes del año 2024.
- PARTITION p\_2024: Almacena reservas con una fecha\_reserva antes del año 2025.
- PARTITION p\_2025: Almacena reservas con una fecha\_reserva antes del año 2026.
- PARTITION p\_max: Captura todas las reservas con años superiores a 2025.

El particionamiento de bases de datos es una técnica que divide tablas grandes en partes más pequeñas y manejables, lo que ofrece múltiples beneficios. Entre los principales beneficios se incluyen la mejora del rendimiento. Para implementar particionamiento en sistemas de bases de datos grandes, es recomendable elegir un criterio de particionamiento adecuado como fechas o rangos de valores.

## 5. Procedimientos Almacenados, Vistas y Triggers, Funciones (prácticas de cada uno)

**Objetivo:** Mejorar la eficiencia y automatizar tareas mediante el uso de procedimientos almacenados, vistas y triggers.

### Actividades:

1. Crear procedimientos almacenados.

**Práctica:** Crear un procedimiento para calcular el precio total de una reserva, aplicando descuentos y cargos adicionales, aplicar 2 ejercicios y explicar comprensión al 100%

**Investigación:** Explorar cómo los procedimientos almacenados pueden mejorar la reutilización de código y la eficiencia.

**Importancia del Conocimiento:** Los procedimientos almacenados centralizan la lógica y pueden mejorar el rendimiento al ejecutarse directamente en el servidor.

```
115     delimiter $$
116 • create procedure calcular_precio_venta (
117     IN p_id_venta int,
118     IN p_descuento decimal(5,2),
119     IN p_cargo decimal(5,2)
120 )
121 begin
122     declare precio_libro decimal(4,2);
123     declare cantidad int;
124     declare subtotal decimal(10,2);
125     declare precio_total decimal(10,2);
126     -- Obtener el precio del libro y la cantidad vendida
127     select l.precio, v.cantidad
128     into precio_libro, cantidad
129     from ventas v
130     join libros l on v.id_libro = l.id_libro
131     where v.id_venta = p_id_venta;
132     -- Calcular el subtotal
133     set subtotal = precio_libro * cantidad;
134     -- Calcular el precio total aplicando descuento y cargo adicional
135     set precio_total = subtotal - (subtotal * p_descuento / 100) + (subtotal * p_cargo / 100);
136     -- Actualizar el precio total en la tabla de ventas
137     update ventas
138     set total = precio_total
139     where id_venta = p_id_venta;
140 end $$
141 delimiter ;
```

Este Procedimiento está diseñado para calcular y actualizar el precio total de una venta en función del precio del libro, la cantidad vendida, un descuento y un cargo adicional.

Primero declaramos tres variables de entrada, luego declaramos variables para almacenar los datos de las operaciones realizadas. Después obtenemos los valores necesarios de las tablas ventas y libros, a continuación, procedemos a realizar la multiplicación del precio por la cantidad de libros añadiéndole un descuento y un cargo adicional

Ahora probaremos el funcionamiento del procedimiento almacenado con una venta que tiene como id: 411

Result Grid						
	id_venta	id_cliente	id_libro	fecha_venta	cantidad	total
▶	411	101	301	2025-01-11	1	12.50
*	NULL	NULL	NULL	NULL	NULL	NULL

Y después de llamar al procedimiento almacenado añadiéndole un 10% de descuento y un cargo de 5%, obtenemos:

```
142 • call calcular_precio_venta(411, 10.00, 5.00);
143 • select * from ventas where id_venta=411;
```

Result Grid						
	id_venta	id_cliente	id_libro	fecha_venta	cantidad	total
▶	411	101	301	2025-01-11	1	11.88
*	NULL	NULL	NULL	NULL	NULL	NULL

Este procedimiento almacenado no solo mejora la reutilización del código al centralizar la lógica de cálculo, sino que también aumenta la eficiencia operativa al reducir el tráfico entre cliente y servidor y optimizar las operaciones dentro del servidor de base de datos.

## 2. Crear vistas para simplificar consultas complejas.

**Práctica:** Crear vistas que presenten información de varias tablas de manera unificada (por ejemplo, una vista que combine datos de Vuelos, Clientes y Reservas).

**Investigación:** Investigar las ventajas de usar vistas en lugar de consultas complejas repetitivas.

**Importancia del Conocimiento:** Las vistas ayudan a simplificar el acceso a datos complejos y pueden mejorar la seguridad al limitar el acceso directo a las tablas.

```
146 • create view vista_ventas_clientes_libros as
147     select
148         v.id_venta,
149         c.nombre as cliente,
150         l.titulo as libro,
151         v.cantidad,
152         v.total,
153         v.fecha_venta
154     from ventas v
155     join clientes c on v.id_cliente = c.id_cliente
156     join libros l on v.id_libro = l.id_libro;
```

En esta vista se unen las tablas ventas, clientes y libros para combinar los datos relacionados, lo que permite obtener una visión completa de cada venta, estas uniones lo hacemos mediante los Joins.

Ahora probaremos la vista creada. Una vez llamemos la vista creada, esta mostrará información de las tres tablas relacionadas:

158 • `select * from vista_ventas_clientes_libros;`

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	id_venta	cliente	libro	cantidad	total	fecha_venta
▶	410	Juan Pérez	Cien años de soledad	2	30.38	2025-01-10
	411	María Gómez	La casa de los espíritus	1	11.88	2025-01-11
	412	Carlos Torres	La ciudad y los perros	1	18.00	2025-01-12
	413	Ana Sánchez	Harry Potter y la piedra filosofal	3	60.00	2025-01-13
	414	Pedro Martínez	El resplandor	2	29.98	2025-01-14

vista\_ventas\_clientes\_libros 6 ×

Las vistas simplifican las consultas complejas al encapsular la lógica en una tabla virtual, lo que facilita su uso y comprensión. Además, proporcionan un nivel adicional de seguridad al restringir el acceso a datos sensibles, permitiendo que los usuarios interactúen solo con la información necesaria. Las vistas también optimizan el rendimiento al reducir la cantidad de datos procesados y mejorar la velocidad de acceso a la información.

### 3. Implementar triggers para auditoría y control de cambios.

**Práctica:** Crear triggers que registren cambios en las tablas de Reservas y Pagos cada vez que un registro se actualiza o elimina., 2 ejercicios conocimiento al 100%

**Investigación:** Investigar cómo utilizar triggers para mantener un historial de cambios en la base de datos.

**Importancia del Conocimiento:** Los triggers permiten automatizar tareas como la auditoría y validación de datos.

**Acciones a realizar de forma automática, es decir si desea aplicar un calcuo de descuento y cambio del iva que se debe hacer donde se pone esos valores y como se automatiza**



```

-- tiggers de actulizaciones
delimiter $$
create trigger auditoria_update_reservas
after update on reservas
for each row
begin
    insert into auditoria (nombre_tabla, tipo_accion, id_registro, fecha_accion, cambios)
    values ('reservas', 'UPDATE', old.id_reserva, now(),
        concat('Fecha antigua: ', old.fecha_reserva, ', Fecha nueva: ', new.fecha_reserva,
            ', Cliente antiguo: ', old.id_cliente, ', Cliente nuevo: ', new.id_cliente));
end $$
delimiter ;
delimiter $$
create trigger auditoria_update_ventas
after update on ventas
for each row
begin
    insert into auditoria (nombre_tabla, tipo_accion, id_registro, fecha_accion, cambios)
    values ('ventas', 'UPDATE', old.id_venta, now(),
        concat('Cantidad antigua: ', old.cantidad, ', Cantidad nueva: ', new.cantidad,
            ', Total antiguo: ', old.total, ', Total nuevo: ', new.total));
end $$
delimiter ;

```

Estos tiggers registran la información sobre una actualización de datos en la tabla auditoria, estas se activan automáticamente después de que se actualiza un registro en las tablas reservas y ventas el comando que lo hace posible es “after update”.

Ahora probaremos el tigger:

```

215 • DELETE FROM reservas WHERE id_reserva = 500;
216 • select * from auditoria ;
217
218

```

	id_auditoria	nombre_tabla	tipo_accion	id_registro	fecha_accion	cambios
▶	1	reservas	UPDATE	502	2025-01-26 16:48:20	Fecha antigua: 2025-01-22, Fecha nueva: 202...
	2	ventas	UPDATE	411	2025-01-30 20:28:55	Cantidad antigua: 1, Cantidad nueva: 1, Total ...
	3	reservas	DELETE	500	2025-01-30 22:11:15	Fecha: 2025-01-20, Cliente: 100



```

174 -- tiggers de eliminación
175 delimiter $$
176 • create trigger auditoria_delete_reservas
177 after delete on reservas
178 for each row
179 begin
180     insert into auditoria (nombre_tabla, tipo_accion, id_registro, fecha_accion, cambios)
181     values ('reservas', 'DELETE', old.id_reserva, now(),
182           concat('Fecha: ', old.fecha_reserva, ', Cliente: ', old.id_cliente));
183 end $$
184 delimiter ;
185 delimiter $$
186 • create trigger auditoria_delete_ventas
187 after delete on ventas
188 for each row
189 begin
190     insert into auditoria (nombre_tabla, tipo_accion, id_registro, fecha_accion, cambios)
191     values ('ventas', 'DELETE', old.id_venta, now(),
192           concat('Cantidad: ', old.cantidad, ', Total: ', old.total));
193 end $$
194 delimiter ;
195

```

Estos tiggers registran la información sobre una eliminación de datos en la tabla auditoria, estas se activan automáticamente después de que se elimine un registro en las tablas reservas y ventas el comando que lo hace posible es “after delete”.

Ahora probaremos el tigger:

```

209 • UPDATE reservas
210 SET
211     fecha_reserva = '2025-02-01',
212     id_cliente = 102
213 WHERE id_reserva = 502;
214
215 • select * from auditoria ;

```

	id_auditoria	nombre_tabla	tipo_accion	id_registro	fecha_accion	cambios
▶	1	reservas	UPDATE	502	2025-01-26 16:48:20	Fecha antigua: 2025-01-22, Fecha nueva: 202...
	2	ventas	UPDATE	411	2025-01-30 20:28:55	Cantidad antigua: 1, Cantidad nueva: 1, Total ...
*	NULL	NULL	NULL	NULL	NULL	NULL

En conclusión, los triggers pueden registrar automáticamente cada vez que se realiza una operación UPDATE o DELETE en una tabla. Esto se logra creando un trigger que inserte información relevante sobre el cambio en una tabla de auditoría.

## 6. Monitoreo y Optimización de Recursos

**Objetivo:** Controlar el rendimiento de la base de datos, identificando y solucionando problemas de recursos.

**Actividades:**

1. Monitorear el rendimiento de consultas.

**Práctica:** Usar herramientas como SHOW PROCESSLIST para detectar consultas lentas y optimizarlas.

```
298 • SHOW PROCESSLIST;  
299  
300
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content: [IA](#)

	Id	User	Host	db	Command	Time	State	Info
▶	5	event_scheduler	localhost	NULL	Daemon	1263	Waiting on empty queue	NULL
	22	root	localhost:64370	NULL	Sleep	24		NULL
	23	root	localhost:64371	libreria	Query	0	init	SHOW PROCESSLIST

Como vemos en la imagen al utilizar el comando en la database de “libreria” se puede ver que no está yendo lenta porque en la columna “Time” nos da 0.

**Investigación:** Investigar las mejores prácticas para monitorear el rendimiento de las consultas en producción.

- Definir Métricas Clave: Mide tiempos de respuesta, errores y uso de recursos.
- Usar Herramientas: Utiliza soluciones como Datadog o SolarWinds para identificar consultas lentas.
- Configurar Dashboards: Visualiza métricas clave y recibe alertas personalizadas.
- Analizar Tendencias: Observa patrones para prevenir problemas futuros.
- Establecer Alertas: Configura notificaciones automáticas para detectar anomalías.
- Optimizar Regularmente: Revisa y ajusta consultas según el rendimiento observado.

**Importancia del Conocimiento:** El monitoreo proactivo puede identificar cuellos de botella antes de que afecten el rendimiento del sistema.

## 2. Realizar pruebas de carga.

**Práctica:** Simular múltiples usuarios concurrentes usando herramientas como Apache JMeter para ver cómo responde la base de datos bajo alta carga.

Para esta prueba vamos a utilizar 100 usuarios:

Montar grupo de Hilos

Nombre: Montar grupo de Hilos

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo

Número de Hilos: 100

Periodo de Subida (en segundos): 1

Contador del bucle: ☐ Sin fin 1

☒ Same user on each iteration

☐ Planificador

Duración (segundos)

Retardo de arranque (segundos)

Además, configuramos la conexión con la base de datos:

Plan de Pruebas

Montar grupo de Hilos

Configuración de la Conexión JDBC

Peticion JDBC

Ver Árbol de Resultados

Reporte resumen

Nombre Variable Enlazado al Pool

Nombre de Variable: ConexionSQL

Configuración del Pool de Conexiones

Número Máximo de Conexiones: 0

Timeout del Pool: 10000

Intervalo de Limpieza por Inactividad (ms): 60000

Auto Commit: True

Transaction Isolation: DEFAULT

Pool Prepared Statements: -1

Preinit Pool: False

Init SQL statements separated by new line:

1

Validación de Conexión por Pool

Keep-Alive: True

Edad máxima de las Conexiones (ms): 5000

Query de Validación:

Configuración de la Conexión a Base de Datos

URL de la Base de Datos: jdbc:mysql://localhost:3306/libreria

Clase del Driver JDBC: com.mysql.jdbc.Driver

Nombre de Usuario: root

Password: \*\*\*\*

Y se creó el request de jdbc para hacer ahí las consultas:

Plan de Pruebas

Montar grupo de Hilos

Configuración de la Conexión JDBC

Peticion JDBC

Ver Árbol de Resultados

Reporte resumen

Peticion JDBC

Nombre: Peticion JDBC

Comentarios

Nombre de Variable Ligada al Pool

Nombre de Variable: ConexionSQL

Query SQL

Solo Query: Select Statement

Query:

1 select \* from ventas;

Prueba utilizando el “select \* from ventas”

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Peticion JDBC	100	31	24	54	5,37	0,00%	97,1/sec	20,01	0,00	211,0
Total	100	31	24	54	5,37	0,00%	97,1/sec	20,01	0,00	211,0

**Prueba utilizando el “SELECT \* FROM reservas\_particionada ORDER BY fecha\_reserva;”**

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Bytes
Petición JDBC	100	2972	2316	3560	293,10	0,00%	27,8/sec	5,48	0,00	202,0
Total	100	2972	2316	3560	293,10	0,00%	27,8/sec	5,48	0,00	202,0

**Investigación:** Investigar cómo realizar pruebas de estrés y carga en bases de datos de alto rendimiento.

1. Definir Objetivos:
  - Escenarios: Identifica situaciones a simular (picos de tráfico).
  - Métricas: Establece qué medir (tiempos de respuesta, tasas de error).
2. Seleccionar Herramientas:
  - Apache JMeter: Para simular múltiples usuarios.
  - Gatling: Enfocada en rendimiento y escalabilidad.
  - Locust: Basada en Python, fácil para pruebas distribuidas.
3. Diseñar Casos de Prueba:
  - Carga: Simula usuarios concurrentes.
  - Estrés: Lleva la base de datos más allá de su capacidad.
4. Ejecutar Pruebas:
  - Hazlo en un entorno controlado y monitorea el rendimiento en tiempo real.
5. Analizar Resultados:
  - Identifica cuellos de botella y ajusta configuraciones según sea necesario.}
6. Repetir y Validar:
  - Realiza pruebas periódicas para asegurar mejoras continuas.

**Importancia del Conocimiento:** Las pruebas de carga aseguran que el sistema sea capaz de manejar tráfico alto y crecimiento de datos.

**3. Optimizar el uso de recursos y gestionar índices.**

**Práctica:** Identificar índices no utilizados y eliminarlos para liberar recursos y mejorar la velocidad de las operaciones de escritura.

- Estos son los índices que se utilizan en nuestra base de datos

```

302 • SHOW INDEX FROM ventas;
303 • SHOW INDEX FROM reservas;
304
305

```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible
▶ ventas	0	PRIMARY	1	id_venta	A	1	NULL	NULL		BTREE			YES
ventas	1	idx_ventas_id_cliente	1	id_cliente	A	0	NULL	NULL	YES	BTREE			YES
ventas	1	idx_ventas_id_libro	1	id_libro	A	0	NULL	NULL	YES	BTREE			YES
ventas	1	idx_ventas_fecha_venta	1	fecha_venta	A	0	NULL	NULL		BTREE			YES

```

303 • SHOW INDEX FROM reservas;

```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type
▶ reservas	0	PRIMARY	1	id_reserva	A	0	NULL	NULL		BTREE
reservas	1	id_libro	1	id_libro	A	0	NULL	NULL	YES	BTREE
reservas	1	idx_reservas_id_cliente	1	id_cliente	A	0	NULL	NULL	YES	BTREE
reservas	1	idx_reservas_fecha_reserva	1	fecha_reserva	A	0	NULL	NULL		BTREE

Identificar en que se usan los índices:

```

305 • EXPLAIN
306 SELECT *
307 FROM ventas
308 WHERE id_cliente = 100;

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶ 1	SIMPLE	ventas	NULL	ref	idx_ventas_id_cliente	idx_ventas_id_cliente	5	const	1	100.00	NULL

```

310 • EXPLAIN
311 SELECT *
312 FROM reservas
313 WHERE fecha_reserva BETWEEN '2025-01-01' AND '2025-01-31';
314
315

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶ 1	SIMPLE	reservas	NULL	ALL	idx_reservas_fecha_reserva	NULL	NULL	NULL	5	80.00	Using where

Se puede ver con las consultas que los índices si están siendo utilizados y no necesitamos borrar ningún índice.

**Investigación:** Investigar cómo ajustar el número de índices según el tipo de consulta (lectura/escritura).

- Consultas de Lectura:
  - Más Índices: Agrega índices para acelerar SELECT.

- Índices de Cobertura: Incluye columnas clave para evitar escaneos completos.
  - Tipos de Índices: Usa índices agrupados para datos únicos y no agrupados para filtros frecuentes.
2. Consultas de Escritura:
    - Menos Índices: Reduce índices en tablas con INSERT, UPDATE y DELETE frecuentes para evitar sobrecarga.
    - Índices Clave: Mantén solo los necesarios para consultas críticas.
  3. Monitoreo y Ajuste:
    - Revisa patrones de uso, elimina índices redundantes y ajusta según las necesidades.

**Importancia del Conocimiento:** La optimización de los recursos asegura un uso eficiente del hardware y mejora la escalabilidad.

## 7. Git y Control de Versiones

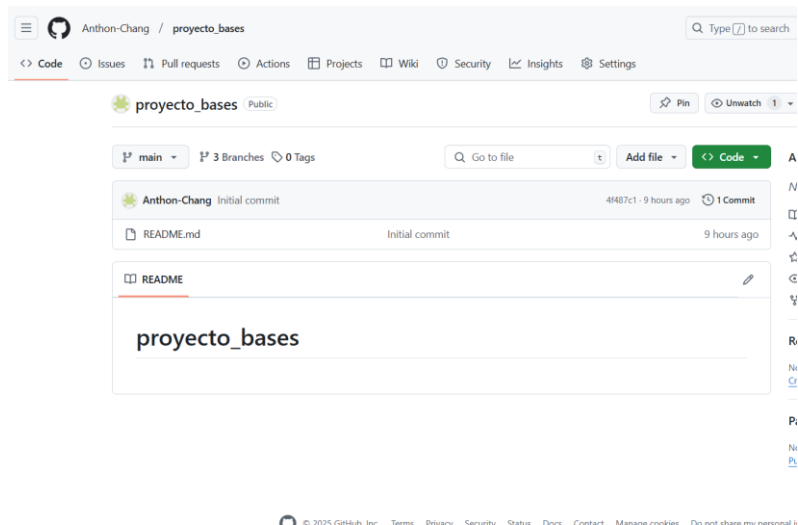
**Objetivo:** Asegurar que el código relacionado con la base de datos esté versionado y que el equipo pueda colaborar de manera eficiente.

**Actividades:**


### 1. Configurar un repositorio de Git para el proyecto.



**Práctica:** Inicializar un repositorio en Git y subir los archivos de definición de la base de datos, scripts de SQL y procedimientos almacenados.


**Repositorio en GitHub inicializado:**



**Carga de archivos:**


**Anthon-Chang** Add files via upload


 Proyecto_Libreria	Add files via upload
 README.md	Initial commit







 README

---

# proyecto\_bases

---


**Anthon-Chang** Add files via upload

Name
 ..
 Diccionario de datos
 Modelados
 Presentaciones
 Responsabilidades
 Scripts

**Investigación:** Investigar buenas prácticas de flujo de trabajo en Git (por ejemplo, uso de ramas, git merge).

Tenemos la práctica de ramas que jerarquiza la información y archivos.

Uso de Ramas: Crea ramas para cada nueva característica o corrección y elimina las ramas obsoletas después de fusionarlas.

Para integrar cambios de una rama a otra, se usan:

git merge:

- Une el historial de dos ramas creando un nuevo commit de fusión.
- Se recomienda para combinar feature con develop o hotfix con main.

**Importancia del Conocimiento:** Git permite la colaboración y el manejo eficiente de cambios en el código, especialmente cuando se trabaja en equipo.

## 2. Realizar commits frecuentes y con mensajes claros.

**Práctica:** Hacer commits regularmente, describiendo claramente los cambios realizados en los scripts SQL y la estructura de la base de datos.

```

On branch main
Your branch is up to date with 'origin/main'.
Changes not staged for commit:
(use "git add <file> ..." to update what will be committed)
(use "git restore <file> ..." to discard changes in working directory)
modified:
day1/libreria.sql
no changes added to commit (use "git add" and/or "git commit -a")
20230315 git: (main) x git add .
20230315 git: (main) x git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file> ..." to unstage)
modified:
day1/libreria.sql

```

**Investigación:** Investigar cómo utilizar git rebase y git pull para evitar conflictos.

Git rebase nos ayuda con mantener el historial limpio o sea permite mover commits de una rama sobre otra, evitando merges innecesarios y manteniendo un historial más lineal.

Git pull ayuda a mantener la rama actualizada. Cuando varios desarrolladores trabajan en la misma rama, git pull puede crear commits de fusión innecesarios. Usar --rebase evita esto.

**Importancia del Conocimiento:** Un flujo de trabajo claro en Git mejora la colaboración y la gestión de versiones.

### 3. Automatizar pruebas con GitHub Actions.

**Práctica:** Crear flujos de trabajo de CI/CD que automaticen las pruebas de las consultas SQL y otros scripts relacionados con la base de datos.

**Investigación:** Investigar sobre integración continua y cómo aplicarla en bases de datos con GitHub Actions.

La Integración Continua es una práctica en la que los cambios en el código se integran, prueban y despliegan automáticamente en un entorno controlado.

- Se activa cuando hay un push o pull request en main.
- Crea un servicio de MySQL 8 en un contenedor.
- Descarga el código del repositorio.
- Instala el cliente de MySQL en la máquina de GitHub Actions.
- Ejecuta las migraciones SQL automáticamente.

**Importancia del Conocimiento:** Las pruebas automáticas aseguran que las bases de datos se mantengan consistentes y funcionales a lo largo del tiempo.

**Enlace del repositorio del GitHub:** [https://github.com/Anthon-Chang/proyecto\\_bases](https://github.com/Anthon-Chang/proyecto_bases)

## CONSIDERACIONES

**Sugerencia para mejorar el trabajo en equipo y habilidades blandas:**

Para optimizar la colaboración, sugiero crear una **tabla de responsabilidades y capacitación**. Esta tabla permitirá monitorear quién es responsable de cada tema, qué actividades se han realizado para capacitar a los compañeros y cuándo se realizaron. Esto fomenta la responsabilidad individual y la transparencia en el equipo.



**Mejoras en habilidades blandas:**

**Comunicación efectiva:** Promover reuniones de retroalimentación para que todos los miembros intercambien ideas y soluciones. Fomentar la participación activa en las discusiones y evitar el trabajo mecánico.

**Investigación y curiosidad:** Incentivar a los miembros a investigar profundamente sobre los temas, identificando problemas no documentados y buscando soluciones innovadoras.

**Colaboración activa:** Fomentar un ambiente de colaboración, promoviendo sesiones de brainstorming y revisiones entre compañeros, y asegurando que todos estén alineados con el progreso del proyecto.

**Responsabilidad colectiva:** Asegurar que los miembros del equipo no solo sean responsables de sus tareas individuales, sino también del éxito global del proyecto. Esto incluye apoyar a los compañeros en su aprendizaje.

Responsable	Tema Asignado	Fecha de asignación	Fecha de culminación	Fecha de Capacitación
Anthon Chang	Diseñar modelos, Seguridad ,Auditoría y control de acceso	22/01/2025	04/02/2025	22/01/2025
Eddy Morales	Optimización y Rendimiento de Consultas, Procedimientos Almacenados, Vistas y Triggers, Funciones	22/1/2025	04/02/2025	22/1/2025
Paulo Cisneros	Respaldos y Recuperación de Datos, Monitoreo y Optimización de Recursos	22/01/2025	04/02/2025	22/01/2025

## Temáticas Disponibles

- El grupo es libre de elegir una temática sin repetirse con los demás grupos, seguir el ejemplo indicado

## Entregables

```
/Project-AEROLINEAS
/Presentaciones
  Proyecto.pptx
/Informe
  Informe_Proyecto.pdf
/Modelados
  Modelo_ER_Conceptual.png
  Modelo_ER_Logico.png
  Modelo_ER_Fisico.png
/Diccionario_De_Datos
  diccionario_datos.xlsx
/Responsabilidades
  responsabilidades_equipo.xlsx
/Scripts
  /Modelado
    crear_tablas.sql
    relaciones_integridad.sql
  /Seguridad
    crear_rol.sql
    cifrado_datos.sql
  /Auditoria
    activar_auditoria.sql
  /Optimización
    crear_indices.sql
    optimizar_consultas.sql
README.md
```

## EXPLICACION

### Presentación (PPT):

Crear una carpeta llamada Presentaciones donde se suba el archivo .ppt o .pptx correspondiente a la explicación del proyecto, los objetivos, las actividades, y los resultados alcanzados.

### Documento Informe:

- Subir el informe detallado del proyecto en formato .docx o .pdf, incluyendo:
  - Resumen ejecutivo
  - Descripción de cada fase del proyecto
  - Resultados obtenidos
  - Conclusiones

### Modelados (ER):

Crear una carpeta llamada Modelados para almacenar los diagramas de modelado ER. Estos pueden estar en formatos como .png, .jpg, .pdf.

- Incluir versiones del modelo conceptual, lógico y físico.

### Diccionario de Datos:

Subir un archivo en formato .xlsx o .csv que contenga el diccionario de datos. Este debe incluir detalles como:

- Nombre de la tabla
- Descripción de la tabla
- Campos (nombre, tipo de datos, restricciones, etc.)
- Relación con otras tablas

### Responsabilidades:

Subir un archivo que detalle las responsabilidades de cada miembro del equipo, indicando qué tareas corresponden a cada uno. Este archivo puede ser una tabla en formato .xlsx o .docx.

**Script Actividades a Realizar:**

Subir los scripts de las actividades, como la creación de la base de datos, la implementación de procedimientos almacenados, vistas, triggers, etc. Estos archivos pueden ser .sql o .sh (si son scripts de shell para automatizar tareas).

Estos scripts deben estar organizados en carpetas según la actividad, por ejemplo, Scripts/Modelado, Scripts/Seguridad, Scripts/Auditoría, etc.