CLASE 2

Instalación y Primeros Pasos con MySQL Workbench

SQL 1025

Manual Práctico de Configuración

En esta clase aprenderás:

- Instalar MySQL y MySQL Workbench
- Navegar por la interfaz de Workbench
- Crear tu primera base de datos con tablas
- Establecer relaciones entre tablas
- Insertar y consultar datos

Índice

1. Introducción					
2.	Instalación de MySQL y MySQL Workbench 2.1. Requisitos del Sistema	3 3 4			
3.	Primeros Pasos en MySQL Workbench 3.1. Server Status 3.2. Client Connections 3.3. Users and Privileges 3.4. Data Export 3.5. Data Import 3.6. Dashboard 3.7. Schema 3.8. Pestaña de Consultas (Query) 3.9. Cuestionario de Repaso	5 5 5 6 6 7 7 7 8 8			
4.	4.1. Proyecto: Sistema de Blog 4.2. Paso 1: Crear la Base de Datos 4.3. Creación de Tablas 4.3.1. Tabla: Usuarios 4.3.2. Tabla: Publicaciones 4.3.3. Tabla: Comentarios 4.4. Claves Foráneas 4.4.1. ¿Qué son las Claves Foráneas?	10 10 10 10 11 11 11 12 12			
	5.1. Insertar Usuarios 5.2. Verificar los Datos Insertados 5.3. Insertar Publicaciones 5.4. Insertar Comentarios 5.5. Consultas Prácticas 5.5.1. Ver todas las publicaciones con el nombre del autor 5.5.2. Ver comentarios de una publicación específica 5.5.3. Contar publicaciones por usuario	14 14 15 15 15 15			
	Buenas Prácticas 7.1. Nomenclatura de Bases de Datos	16 16 16 17 17			

8.	Solución de Problemas Comunes	18
	8.1. Error: Cannot add foreign key constraint	18
	8.2. Error: Duplicate entry for key 'PRIMARY'	18
	8.3. Error: Table doesn't exist	18
	8.4. No puedo conectarme a MySQL	18
9.	Comandos SQL de Referencia Rápida	19
	9.1. Gestión de Bases de Datos	19
	9.2. Gestión de Tablas	19
	9.3. Modificación de Tablas	19
	9.4. Manipulación de Datos	20
10	.Glosario de Términos	21
11	.Recursos Adicionales	21
	11.1. Documentación Oficial	21
	11.2. Herramientas Recomendadas	21
	11.3. Práctica Adicional	21
12	Checklist de Verificación	22
13	Resumen de la Clase	22
14	.Próximos Pasos	23

1 Introducción

Objetivos de la Clase

Al finalizar esta clase, serás capaz de:

- ✓ Instalar y configurar MySQL Server y MySQL Workbench
- ✓ Navegar por todas las secciones de MySQL Workbench
- Crear bases de datos y tablas con diferentes tipos de datos
- ✓ Comprender y aplicar claves primarias y foráneas
- ✓ Insertar datos en tablas y realizar consultas básicas

2 Instalación de MySQL y MySQL Workbench

2.1 Requisitos del Sistema

Antes de comenzar, verifica que tu sistema cumpla con los siguientes requisitos:

- Sistema Operativo: Windows 10/11, macOS 10.15+, o Linux (Ubuntu, Debian, etc.)
- RAM: Mínimo 4 GB (recomendado 8 GB)
- Espacio en Disco: Al menos 2 GB disponibles
- Procesador: Intel Core i3 o equivalente

2.2 Proceso de Instalación

∄ Pasos a Seguir

Paso 1: Descargar MySQL

- 1. Visita https://dev.mysql.com/downloads/
- 2. Selecciona MySQL Community Server
- 3. Elige tu sistema operativo
- 4. Descarga el instalador

Paso 2: Instalar MySQL Server

- 1. Ejecuta el instalador descargado
- 2. Selecciona Developer Default para instalar todas las herramientas
- 3. Configura una contraseña segura para el usuario root
- 4. Completa la instalación siguiendo las instrucciones en pantalla

Paso 3: Verificar MySQL Workbench

4

MySQL Workbench se instala automáticamente con el paquete Developer Default. Si no está instalado:

- 1. Descarga MySQL Workbench desde https://dev.mysql.com/downloads/workbench/
- 2. Instala siguiendo las instrucciones del asistente

i Nota Importante

Importante: Guarda tu contraseña de root en un lugar seguro. La necesitarás para conectarte a la base de datos.

2.3 Primera Conexión

Una vez instalado, abre MySQL Workbench y:

- 1. Haz clic en la conexión local que aparece (generalmente Local instance MySQL)
- 2. Ingresa la contraseña de root que configuraste
- 3. Haz clic en \mathbf{OK}

¡Felicidades! Ya estás conectado a tu servidor MySQL local.

3 Primeros Pasos en MySQL Workbench

MySQL Workbench tiene una interfaz completa con múltiples secciones. Vamos a explorar cada una de ellas.

3.1 Server Status

Concepto Clave

El **Server Status** muestra información en tiempo real sobre el estado de tu servidor MySQL.

Información que puedes ver:

- Estado del servidor (corriendo o detenido)
- Tiempo de actividad
- Número de conexiones activas
- ✓ Versión de MySQL instalada

¿Para qué sirve?

Esta sección es útil para:

- Monitorear el rendimiento del servidor
- Detectar problemas de conexión
- Verificar que el servidor está funcionando correctamente

3.2 Client Connections

Concepto Clave

Client Connections muestra todas las conexiones activas al servidor MySQL.

Información disponible:

- Usuario conectado
- Host de origen
- Base de datos en uso
- Tiempo de conexión
- Consultas ejecutadas

Acciones que puedes realizar:

- X Terminar conexiones inactivas
- Ver detalles de cada conexión
- Monitorear actividad de usuarios

3.3 Users and Privileges

Concepto Clave

Esta sección permite gestionar los usuarios del sistema y sus permisos de acceso.

Operaciones principales:

- 1. Crear nuevos usuarios: Define nombre de usuario y contraseña
- 2. Asignar privilegios: Determina qué puede hacer cada usuario
- 3. Limitar acceso: Restringe acceso a bases de datos específicas
- 4. **Gestionar roles:** Agrupa permisos para facilitar la administración

⟨**/>** Ejemplo Práctico

Tipos de privilegios comunes:

- SELECT: Permite leer datos
- INSERT: Permite agregar nuevos registros
- UPDATE: Permite modificar registros existentes
- DELETE: Permite eliminar registros
- CREATE: Permite crear nuevas tablas/bases de datos
- DROP: Permite eliminar tablas/bases de datos
- ALL PRIVILEGES: Otorga todos los permisos

⚠ Advertencia

Seguridad: Nunca compartas las credenciales del usuario **root**. Crea usuarios específicos con privilegios limitados para aplicaciones y otros desarrolladores.

3.4 Data Export

Concepto Clave

Data Export permite hacer copias de seguridad (backups) de tus bases de datos.

Opciones de exportación:

- Exportar bases de datos completas
- Exportar tablas específicas
- Incluir estructura (schema) y datos
- Exportar como archivo SQL o CSV

¿Cuándo usar Data Export?

- Antes de hacer cambios importantes en la base de datos
- Para crear copias de seguridad regulares
- Para migrar datos a otro servidor
- Para compartir datos con otros desarrolladores

3.5 Data Import

Concepto Clave

Data Import permite restaurar bases de datos desde archivos de respaldo o importar nuevos datos.

Formatos soportados:

- Archivos SQL (.sql)
- Archivos CSV (.csv)
- Dumps de MySQL

Proceso de importación:

- 1. Selecciona el archivo a importar
- 2. Elige la base de datos de destino
- 3. Configura opciones adicionales (codificación, delimitadores)
- 4. Ejecuta la importación
- 5. Verifica los resultados

i Nota Importante

Si importas una base de datos completa, asegúrate de que no exista una con el mismo nombre o los datos se sobrescribirán.

3.6 Dashboard

Concepto Clave

El **Dashboard** es el panel principal que muestra una vista general del estado de tu servidor MySQL.

Componentes del Dashboard:

- Gráficos de rendimiento en tiempo real
- Información del servidor
- Estado de conexiones

Métricas de uso de recursos

3.7 Schema

Concepto Clave

La sección de **Schema** (Esquema) muestra todas las bases de datos disponibles en tu servidor y su estructura.

Vista jerárquica:

- ■ Bases de datos
 - III Tablas
 - ∘ III Columnas
 - PÍndices
 - Relaciones (Foreign Keys)
 - • Vistas
 - Procedimientos almacenados
 - Funciones

Acciones disponibles:

- Hacer clic derecho en cualquier elemento para ver opciones
- Crear, modificar o eliminar objetos de la base de datos
- Ver estructura de tablas
- Examinar datos directamente

3.8 Pestaña de Consultas (Query)

Concepto Clave

La **pestaña Query** es donde escribes y ejecutas tus comandos SQL.

Características principales:

- Leditor de código con resaltado de sintaxis
- ▶ Botón para ejecutar consultas
- Historial de consultas ejecutadas
- Guardar consultas para uso futuro
- 🞛 Vista de resultados en formato tabla

Atajos de teclado útiles:

- Ctrl + Enter: Ejecutar la consulta actual
- Ctrl + Shift + Enter: Ejecutar todas las consultas
- Ctrl + /: Comentar/descomentar línea
- Ctrl + B: Formatear código SQL

3.9 Cuestionario de Repaso

Ejercicio Práctico

Responde las siguientes preguntas para verificar tu comprensión:

- 1. ¿Qué información puedes ver en Server Status?
- 2. ¿Para qué sirve la sección Users and Privileges?
- 3. ¿Cuál es la diferencia entre Data Export y Data Import?
- 4. ¿Qué muestra la sección Schema?
- 5. ¿Qué atajo de teclado usas para ejecutar una consulta SQL?

Las respuestas están en las secciones anteriores de este manual.

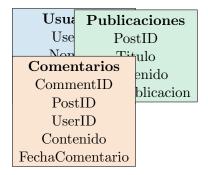
4 Crear tu Primera Base de Datos

Ahora que conoces la interfaz, es momento de crear tu primera base de datos completa con múltiples tablas relacionadas.

4.1 Proyecto: Sistema de Blog

Vamos a crear una base de datos llamada blogdb que contendrá:

- Usuarios: Personas registradas en el blog
- Publicaciones: Artículos escritos por los usuarios
- Comentarios: Comentarios de usuarios en las publicaciones



4.2 Paso 1: Crear la Base de Datos

Primero, crea la base de datos:

```
CREATE DATABASE blogdb;
2 USE blogdb;
```

i Nota Importante

CREATE DATABASE crea una nueva base de datos. USE selecciona la base de datos con la que vamos a trabajar.

4.3 Creación de Tablas

4.3.1 Tabla: Usuarios

Esta tabla almacenará la información de los usuarios registrados:

```
CREATE TABLE 'blogdb'.'usuarios' (
'UserID' INT NOT NULL AUTO_INCREMENT,
'Nombre' VARCHAR(45) NULL,
'Email' VARCHAR(45) NULL,
'FechaRegistro' DATE NULL,
PRIMARY KEY ('UserID')
');
```

Explicación de cada elemento:

UserID INT NOT NULL AUTO_INCREMENT: Identificador único que se incrementa automáticamente

- Nombre VARCHAR(45) NULL: Nombre del usuario (hasta 45 caracteres)
- Email VARCHAR(45) NULL: Correo electrónico del usuario
- FechaRegistro DATE NULL: Fecha en que se registró el usuario
- PRIMARY KEY (UserID): Define UserID como clave primaria

4.3.2 Tabla: Publicaciones

Esta tabla contendrá los artículos del blog:

```
CREATE TABLE 'blogdb'.'publicaciones' (
'PostID' INT NOT NULL AUTO_INCREMENT,
'Titulo' VARCHAR(100) NULL,
'Contenido' TEXT NULL,
'FechaPublicacion' DATE NULL,
'UserID' INT NULL,
PRIMARY KEY ('PostID')

);
```

i Nota Importante

Mejora aplicada: He cambiado Contenido de VARCHAR(45) a TEXT porque las publicaciones suelen ser más largas. También agregué UserID para saber quién escribió cada publicación.

4.3.3 Tabla: Comentarios

Esta tabla almacenará los comentarios de los usuarios:

```
CREATE TABLE 'blogdb'.'comentarios' (
'CommentID' INT NOT NULL AUTO_INCREMENT,
'PostID' INT NULL,
'UserID' INT NULL,
'Contenido' TEXT NULL,
'FechaComentario' DATE NULL,
PRIMARY KEY ('CommentID')

);
```

A Advertencia

Corrección importante: En el código original, PostID y UserID estaban definidos como VARCHAR(45), lo cual es incorrecto. Deben ser INT para poder establecer relaciones con las otras tablas.

4.4 Claves Foráneas

Concepto Clave

Las claves foráneas (Foreign Keys) establecen relaciones entre tablas y garantizan la integridad referencial de los datos.

4.4.1 ¿Qué son las Claves Foráneas?

Una clave foránea es una columna (o conjunto de columnas) en una tabla que hace referencia a la clave primaria de otra tabla.

Beneficios:

- Mantienen la consistencia de los datos
- ✓ Previenen registros huérfanos
- ✓ Facilitan consultas relacionales
- O Documentan la estructura de la base de datos

4.4.2 Agregar Claves Foráneas a Publicaciones

Establecemos que cada publicación debe pertenecer a un usuario existente:

```
ALTER TABLE publicaciones

ADD CONSTRAINT fk_publicaciones_usuarios

FOREIGN KEY (UserID)

REFERENCES usuarios(UserID)

ON DELETE CASCADE

ON UPDATE CASCADE;
```

Explicación:

- CONSTRAINT fk_publicaciones_usuarios: Nombre de la restricción
- FOREIGN KEY (UserID): Columna que actúa como clave foránea
- REFERENCES usuarios(UserID): Tabla y columna referenciada
- ON DELETE CASCADE: Si se elimina un usuario, se eliminan sus publicaciones
- ON UPDATE CASCADE: Si se actualiza el UserID, se actualiza en publicaciones

4.4.3 Agregar Claves Foráneas a Comentarios

Los comentarios tienen dos relaciones: con usuarios y con publicaciones:

```
-- Relacion con usuarios
  ALTER TABLE comentarios
  ADD CONSTRAINT fk_comentarios_usuarios
  FOREIGN KEY (UserID)
  REFERENCES usuarios (UserID)
  ON DELETE CASCADE
6
  ON UPDATE CASCADE;
  -- Relacion con publicaciones
9
 ALTER TABLE comentarios
10
 ADD CONSTRAINT fk_comentarios_publicaciones
 FOREIGN KEY (PostID)
12
 REFERENCES publicaciones (PostID)
13
14 ON DELETE CASCADE
  ON UPDATE CASCADE;
```

Concepto Clave

Cascada de eliminación:

Si eliminas un usuario:

- Se eliminan automáticamente todas sus publicaciones
- Se eliminan automáticamente todos sus comentarios

Si eliminas una publicación:

• Se eliminan automáticamente todos sus comentarios

5 Insertar Datos en una Tabla

Ahora que tenemos la estructura completa, vamos a insertar datos de prueba.

5.1 Insertar Usuarios

```
INSERT INTO usuarios(Nombre, Email, FechaRegistro)

VALUES

('Juan', 'juan.ejemplo@hotmail.com', '2024-06-01'),

('Maria', 'maria.garcia@gmail.com', '2024-06-05'),

('Carlos', 'carlos.lopez@yahoo.com', '2024-06-10');
```

5.2 Verificar los Datos Insertados

Para ver todos los usuarios registrados:

```
1 SELECT * FROM usuarios;
```

Resultado esperado:

azulprimario!20 UserID	Nombre	Email	FechaRegistro
1	Juan	juan.ejemplo@hotmail.com	2024-06-01
2	Maria	maria.garcia@gmail.com	2024-06-05
3	Carlos	carlos.lopez@yahoo.com	2024-06-10

5.3 Insertar Publicaciones

Ahora insertamos algunas publicaciones asociadas a nuestros usuarios:

```
INSERT INTO publicaciones (Titulo, Contenido, FechaPublicacion, UserID
      )
  VALUES
2
3
       ('Mi primer post',
        'Este es el contenido de mi primera publicacion en el blog',
4
        '2024-06-02', 1),
5
       ('Aprende SQL',
        'SQL es fundamental para trabajar con bases de datos
           relacionales',
       '2024-06-06', 2),
       ('Consejos de programacion',
        'Algunos consejos utiles para programadores principiantes',
10
        '2024-06-11', 3);
11
```

5.4 Insertar Comentarios

Finalmente, agregamos comentarios a las publicaciones:

```
INSERT INTO comentarios (PostID, UserID, Contenido, FechaComentario)

VALUES

(1, 2, 'Excelente primer post!', '2024-06-03'),

(1, 3, 'Muy interesante, gracias por compartir', '2024-06-04'),

(2, 1, 'Gran articulo sobre SQL', '2024-06-07'),

(2, 3, 'Me ayudo mucho esta informacion', '2024-06-08'),

(3, 1, 'Consejos muy utiles!', '2024-06-12');
```

5.5 Consultas Prácticas

5.5.1 Ver todas las publicaciones con el nombre del autor

```
p.PostID,
p.Titulo,
u.Nombre AS Autor,
p.FechaPublicacion
FROM publicaciones p
INNER JOIN usuarios u ON p.UserID = u.UserID;
```

5.5.2 Ver comentarios de una publicación específica

```
c.Contenido AS Comentario,
u.Nombre AS Usuario,
c.FechaComentario
FROM comentarios c
INNER JOIN usuarios u ON c.UserID = u.UserID
WHERE c.PostID = 1
ORDER BY c.FechaComentario;
```

5.5.3 Contar publicaciones por usuario

i Nota Importante

Estas consultas utilizan JOIN, que veremos en detalle en clases posteriores. Por ahora, observa cómo podemos relacionar datos de múltiples tablas.

6 Ejercicios Prácticos

Ejercicio Práctico

Ejercicio 1: Ampliar la Base de Datos

Agrega una nueva tabla llamada categorias con los siguientes campos:

- CategoriaID (INT, PRIMARY KEY, AUTO_INCREMENT)
- NombreCategoria (VARCHAR(50))
- Descripcion (TEXT)

Ejercicio 2: Relacionar Tablas

Modifica la tabla publicaciones para agregar una columna CategoriaID y establece una clave foránea con la tabla categorias.

Ejercicio 3: Insertar Datos

Inserta al menos 3 categorías diferentes y asigna categorías a las publicaciones existentes.

Ejercicio 4: Consulta Avanzada

Escribe una consulta que muestre:

- Título de la publicación
- Nombre del autor
- Nombre de la categoría
- Cantidad de comentarios

7 Buenas Prácticas

7.1 Nomenclatura de Bases de Datos

- ✓ Nombres descriptivos: Usa nombres claros que indiquen el propósito
- ✓ Consistencia: Mantén un estilo uniforme (camelCase, snake_case, etc.)
- ✓ Evita espacios: Usa guiones bajos en lugar de espacios
- ✓ Singular vs Plural: Decide una convención (usuarios o usuario) y mantenla

7.2 Diseño de Tablas

- Siempre define claves primarias
- 💡 Usa tipos de datos apropiados para cada campo
- Stablece restricciones de integridad (FOREIGN KEYS)
- 💡 Normaliza tus datos para evitar redundancia
- Documenta tu estructura con comentarios

7.3 Seguridad

⚠ Advertencia

Recomendaciones de seguridad:

- Nunca almacenes contraseñas en texto plano
- Usa conexiones SSL/TLS para conexiones remotas
- Limita los privilegios de los usuarios al mínimo necesario
- Realiza backups regulares de tus bases de datos
- Mantén MySQL actualizado a la última versión estable

7.4 Respaldos

© Concepto Clave

Estrategia de respaldo recomendada:

- Diarios: Para bases de datos en producción
- Semanales: Para bases de datos de desarrollo
- Antes de cambios importantes: Siempre haz un backup

Usa Data Export para crear backups manuales o configura scripts automatizados.

8 Solución de Problemas Comunes

8.1 Error: Cannot add foreign key constraint

Causa: Los tipos de datos de las columnas relacionadas no coinciden. Solución:

```
1 -- Verifica que ambas columnas tengan el mismo tipo
2 DESCRIBE tabla1;
3 DESCRIBE tabla2;
4
5 -- Si es necesario, modifica el tipo de dato
6 ALTER TABLE tabla1
7 MODIFY columna INT;
```

8.2 Error: Duplicate entry for key 'PRIMARY'

Causa: Intentas insertar un registro con una clave primaria que ya existe. Solución:

```
1 -- Usa AUTO_INCREMENT o no especifiques el ID
2 INSERT INTO tabla(campo1, campo2)
3 VALUES ('valor1', 'valor2');
4 -- No incluyas el campo ID en la insercion
```

8.3 Error: Table doesn't exist

Causa: La tabla no existe o estás en la base de datos incorrecta. Solución:

```
-- Verifica en que base de datos estas

SELECT DATABASE();

-- Cambia a la base de datos correcta

USE nombre_base_datos;

-- Verifica que la tabla exista

SHOW TABLES;
```

8.4 No puedo conectarme a MySQL

Posibles causas y soluciones:

- 1. Contraseña incorrecta: Verifica la contraseña del usuario root
- 2. Servidor detenido: Inicia el servicio MySQL desde los servicios del sistema
- 3. Puerto bloqueado: Verifica que el puerto 3306 esté disponible
- 4. Firewall: Configura excepciones para MySQL

O Comandos SQL de Referencia Rápida

9.1 Gestión de Bases de Datos

```
-- Crear base de datos
CREATE DATABASE nombre_bd;

-- Eliminar base de datos
DROP DATABASE nombre_bd;

-- Usar base de datos
USE nombre_bd;

-- Mostrar bases de datos
SHOW DATABASES;
```

9.2 Gestión de Tablas

```
-- Crear tabla
CREATE TABLE nombre_tabla (
columna1 tipo1,
columna2 tipo2
);

-- Eliminar tabla
BROP TABLE nombre_tabla;

-- Mostrar tablas
SHOW TABLES;

-- Ver estructura de tabla
DESCRIBE nombre_tabla;

SHOW COLUMNS FROM nombre_tabla;
```

9.3 Modificación de Tablas

```
-- Agregar columna
ALTER TABLE nombre_tabla
ADD columna tipo;

-- Eliminar columna
ALTER TABLE nombre_tabla
DROP COLUMN columna;

-- Modificar columna
ALTER TABLE nombre_tabla
MODIFY columna nuevo_tipo;

-- Renombrar columna
ALTER TABLE nombre_tabla
CHANGE columna_vieja columna_nueva tipo;
```

9.4 Manipulación de Datos

```
INSERT INTO tabla (columna1, columna2)

VALUES (valor1, valor2);

-- Actualizar datos

UPDATE tabla
SET columna = valor
WHERE condicion;

-- Eliminar datos
DELETE FROM tabla
WHERE condicion;

WHERE condicion;

-- Consultar datos
SELECT * FROM tabla;
SELECT columna1, columna2 FROM tabla WHERE condicion;
```

10 Glosario de Términos

AUTO_INCREMENT

Incrementa automáticamente el valor de una columna numérica

CASCADE Acción que propaga cambios o eliminaciones a tablas relacionadas

CONSTRAINT

Restricción que aplica reglas a los datos de una tabla

Foreign Key Clave foránea que referencia la clave primaria de otra tabla

Primary Key Clave primaria que identifica únicamente cada registro

Schema Estructura que define la organización de la base de datos

VARCHAR Tipo de dato para cadenas de texto de longitud variable

INT Tipo de dato para números enteros

DATE Tipo de dato para fechas (YYYY-MM-DD)

TEXT Tipo de dato para textos largos

11 Recursos Adicionales

11.1 Documentación Oficial

- MySQL Documentation: https://dev.mysql.com/doc/
- MySQL Workbench Manual: https://dev.mysql.com/doc/workbench/en/
- SQL Tutorial: https://www.w3schools.com/sql/

11.2 Herramientas Recomendadas

- MySQL Workbench: Interfaz gráfica oficial
- DBeaver: Cliente de base de datos universal
- phpMyAdmin: Administración web de MySQL
- DataGrip: IDE profesional de JetBrains

11.3 Práctica Adicional

- SQLZoo: https://sqlzoo.net/
- LeetCode Database: https://leetcode.com/problemset/database/
- HackerRank SQL: https://www.hackerrank.com/domains/sql

12 Checklist de Verificación

Marca cada elemento cuando lo hayas completado:

He instalado MySQL Server correctamente

He instalado MySQL Workbench

Puedo conectarme al servidor MySQL local

Entiendo la función de Server Status

Sé cómo ver las conexiones activas

Conozco la sección de Users and Privileges

Puedo exportar e importar datos

Navego cómodamente por el Schema

He creado la base de datos blogdb

He creado las tres tablas del proyecto

He establecido las claves foráneas correctamente

He insertado datos de prueba en todas las tablas

Puedo ejecutar consultas SELECT básicas

Entiendo cómo funcionan las relaciones entre tablas

He completado al menos uno de los ejercicios prácticos

13 Resumen de la Clase

Concepto Clave

Lo que has aprendido hoy:

- 1. Cómo instalar y configurar MySQL Server y MySQL Workbench
- 2. Las diferentes secciones de la interfaz de MySQL Workbench
- 3. Cómo crear bases de datos y tablas con diferentes tipos de datos
- 4. La importancia de las claves primarias y foráneas
- 5. Cómo establecer relaciones entre tablas
- 6. Comandos básicos para insertar y consultar datos
- 7. Buenas prácticas de diseño y seguridad

14 Próximos Pasos

En la siguiente clase aprenderás:

- → Consultas SELECT avanzadas
- → Operadores de filtrado (WHERE, AND, OR)
- → Funciones de agregación (COUNT, SUM, AVG)
- → Ordenamiento y agrupación de datos
- → JOINs entre múltiples tablas



${\bf ¡Felicitaciones!}$

Has completado la Clase 2 de SQL 1025. Ahora tienes las herramientas necesarias para trabajar con bases de datos relacionales.

Sigue practicando y nos vemos en la próxima clase.