

Proyecto Final de la materia Bases de Datos I

Implementación mínima del proyecto

Entrega

La entrega consiste subir a una carpeta del GIT del grupo de al menos cuatro archivos:

- **Justificación del diseño.** Este archivo puede ser un documento estándar (Markdown o PDF) donde se justifique el diseño de la base de datos que debe corresponder al menos hasta 3NF, de forma similar al TP5.
- **Fuente SQL.** Este archivo debe contener **todo** el código SQL necesario para crear las tablas y cargar datos iniciales para el testing del proyecto (*e.g.* INSERT's)
- **Archivo de código Python.** Este archivo debe corresponder con las consignas que se detallan abajo el proyecto elegido por el grupo.
- **Archivo de Diseño de Esquema de Tablas.** Puede ser una captura de la imagen del diseño (*e.g.* imagen JPG, PNG, etc) o alternativamente puede ser un archivo DBML con el diseño.

Características Mínimas Exigidas

El proyecto debe contener **cada una** de estas características mínimas.

- **Diseño esperado.** Identifica entidades fuertes y débiles, atributos, relaciones y cardinalidades.
- **Normalización:** Debes aplicar las formas normales (1NF, 2NF y 3NF) para optimizar el diseño de la base de datos y eliminar redundancias.
- **Restricciones de Integridad:** Define llaves primarias y foráneas. Aplica restricciones NOT NULL, UNIQUE.
- **Operaciones en cascada** Define operaciones en cascada convenientes usando ON UPDATE CASCADE, ON DELETE RESTRICT, etc. donde sea necesario para prevenir errores de consistencia.
- **Consultas Avanzadas:** Implementa INNER JOIN, LEFT JOIN y RIGHT JOIN para recuperar datos relacionados. Utiliza cuando sea necesario funciones agregadas y cláusulas como GROUP BY, HAVING, ORDER BY.
- **Transacciones y Manejo de Errores:** Utiliza transacciones para garantizar la consistencia de los datos en operaciones críticas.
- **Procedimientos Almacenados y Funciones:** Crea procedimientos para manejar operaciones complejas como préstamos y devoluciones. Implementa funciones para cálculos específicos, como multas por retraso.

- **Índices:** Crea índices para mejorar el rendimiento de las consultas más frecuentes. El diseño debe contener al menos *un* (1) índice.
- **Python y mysql.connector:** Desarrolla una aplicación en Python que interactúe con la base de datos MySQL. Implementa una interfaz de usuario (CLI) para facilitar la interacción con la aplicación.

Proyectos

El grupo debe seleccionar uno de estos proyectos para implementarlo como proyecto final.

Proyecto 1: Sistema de Gestión de Biblioteca

Descripción:

Desarrolla un sistema para gestionar una biblioteca que incluye usuarios, libros, préstamos y pagos.

- **Modelo del Sistema.** Diseña un diagrama de Esquema de Tablas que represente entidades como Usuarios, Libros, Préstamos y Pagos. El pago consiste una cuota mensual.
- **Datos iniciales** La base de datos inicial debe contener al menos diez (10) socios y diez (10) libros. La antigüedad *promedio* de los socios debe ser de diez (10) meses.

Consignas para el Menú del CLI

1. **Gestión de Usuarios:** Agregar, ver, actualizar o eliminar información de usuarios.
2. **Gestión de Libros:** Registrar nuevos libros, ver detalles, actualizar información y eliminar entradas.
3. **Manejo de Préstamos:** Calcular la multa por retraso en la devolución de un libro para un socio **dado**, bajo este criterio: por cada día se debe abonar el 3% de la cuota mensual.
4. **Búsqueda y Filtrado:** Buscar libros o usuarios mediante diferentes atributos (título, autor, ID de usuario). Tener en cuenta que la búsqueda puede ser por parte del texto a buscar.
5. **Reporte de morosos** Mostrar un reporte el promedio de meses de los socios que no estén al día con las cuotas.
6. **Modificación de la cuota:** Modificar una cuota para un mes y año dado.

Proyecto 2: Sistema de Ventas en Línea

Descripción:

Crea una plataforma de ventas en línea que gestione productos, clientes y órdenes de compras. Las órdenes de compra deben contener el producto, la fecha y la cantidad de unidades de ese producto.

- **Modelo del Sistema.** Diseña entidades como Productos, Clientes y Órdenes.
- **Datos iniciales** La base de datos inicial debe contener al menos diez (10) productos y diez (10) clientes. La cantidad *promedio* de las órdenes debe ser de diez (10) órdenes por cada cliente.

Consignas para el Menú del CLI

1. **Gestión de Productos:** Agregar, actualizar, ver o eliminar productos, incluyendo categorías y niveles de stock.
 2. **Gestión de Clientes:** Registrar, actualizar y ver detalles de clientes, gestionar contactos.
 3. **Procesamiento de Órdenes:** Mostrar las órdenes pedidas por un cliente dado.
 4. **Búsquedas Avanzadas:** Recuperar productos o clientes con filtros (e.g., productos más vendidos).
 5. **Reporte de productos más vendidos:** Generar un reporte del producto más vendido indicando la cantidad total pedida de ese producto.
 6. **Modificación de valor de un producto:** Modificar las órdenes de un producto dado para ajustarse una **cierta** cantidad máxima.
-

Proyecto 3: Sistema de Gestión de Hospital

Descripción:

Desarrolla un sistema para gestionar un hospital que incluya pacientes, médicos y turnos de consulta médica.

- **Modelo del Sistema.** Representa entidades como Pacientes, Médicos y Turnos con fecha y horario.
- **Datos iniciales** La base de datos inicial debe contener al menos diez (10) pacientes y diez (10) médicos. La cantidad *promedio* de los turnos debe ser de diez (10) turnos por cada paciente.

Ejercicio: Consignas para el Menú del CLI

1. **Gestión de Pacientes:** Registrar, actualizar, ver y eliminar información de pacientes.
2. **Gestión de Doctores:** Agregar, actualizar y ver detalles de los doctores, incluyendo especialidades.
3. **Manejo de Turnos:** Programar, actualizar o cancelar turnos.
4. **Búsquedas Avanzadas:** Recuperar pacientes o médicos mediante diferentes atributos (nombre, especialidad, ID de paciente). Tener en cuenta

que la búsqueda puede ser por parte del texto a buscar.

5. **Reporte de turnos:** Mostrar un reporte donde muestre los tres (3) médicos que más turnos tienen y la cantidad de turnos de cada uno.
6. **Cancelación de turnos:** Se debe poder cancelar para un médico **dado** y un rango de fechas **dado**.