

Enunciado del Problema: Sistema de Gestión de Empleados con MySQL

Desarrolla un sistema de gestión para una empresa que permita administrar la información de empleados, departamentos y puestos utilizando **Python** y una base de datos **MySQL**. El sistema debe permitir realizar las siguientes operaciones a través de un menú interactivo:

Requerimientos:

1. **Agregar Empleado:** El usuario debe poder añadir empleados proporcionando su nombre, el ID del puesto, el ID del departamento, y el salario. Los datos serán almacenados en la tabla empleados en la base de datos MySQL.
2. **Listar Empleados:** El sistema debe mostrar todos los empleados almacenados, incluyendo su nombre, puesto, departamento, y salario.
3. **Actualizar Salario de un Empleado:** El usuario debe poder actualizar el salario de un empleado existente proporcionando su ID y el nuevo salario.
4. **Eliminar Empleado:** El usuario podrá eliminar un empleado de la base de datos proporcionando el ID del empleado.
5. **Agregar Departamento:** Permitir al usuario añadir nuevos departamentos a la tabla departamentos ingresando el nombre del departamento y su descripción.
6. **Listar Departamentos:** Mostrar todos los departamentos almacenados en la base de datos.
7. **Agregar Puesto:** Permitir al usuario añadir nuevos puestos a la tabla puestos, proporcionando el nombre y el nivel del puesto.
8. **Listar Puestos:** Mostrar todos los puestos de trabajo disponibles en la base de datos.
9. **Salir del Sistema:** El usuario puede optar por salir del programa.

Requisitos del Sistema:

1. **Base de Datos MySQL:**
 - **Base de datos:** EmpresaDB
 - **Tablas:**
 - empleados: contiene información sobre los empleados.

- departamentos: contiene información sobre los departamentos de la empresa.
- puestos: contiene los distintos puestos de trabajo en la empresa.

2. Conexión a la Base de Datos en Python:

- Usa el módulo `mysql.connector` para conectar Python con MySQL.
- Los detalles de conexión (host, usuario, contraseña, base de datos) estarán directamente definidos dentro de la clase `Database`.

Tablas en la Base de Datos:

- **empleados:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `puesto_id`: INT (FOREIGN KEY referenciando la tabla `puestos`)
 - `departamento_id`: INT (FOREIGN KEY referenciando la tabla `departamentos`)
 - `salario`: DECIMAL(10, 2)
- **departamentos:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `descripcion`: TEXT
- **puestos:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `nivel`: VARCHAR(50)

Funcionalidades del Menú Interactivo:

- **Opción 1:** Agregar Empleado.
- **Opción 2:** Listar Empleados.
- **Opción 3:** Actualizar Salario de un Empleado.
- **Opción 4:** Eliminar Empleado.
- **Opción 5:** Agregar Departamento.
- **Opción 6:** Listar Departamentos.
- **Opción 7:** Agregar Puesto.
- **Opción 8:** Listar Puestos.
- **Opción 9:** Salir del Programa.

Ejemplo de Uso:

El usuario debe poder agregar empleados, listarlos, modificar su salario o eliminarlos del sistema. También puede gestionar departamentos y puestos, añadiendo y listando los que están registrados en la base de datos.

Al ejecutar el programa, el usuario verá un menú interactivo que le permitirá seleccionar las distintas operaciones del sistema, proporcionando los datos necesarios en cada caso. Una vez que termine, podrá cerrar la conexión con la base de datos al salir del programa.

Este sistema facilita la gestión de empleados, departamentos y puestos en una empresa de manera eficiente, con persistencia de los datos en una base de datos MySQL.

Enunciado del Problema: Sistema de Gestión de Inventarios

Desarrolla un sistema de gestión de inventarios para una tienda que permita registrar y gestionar productos, categorías, y proveedores en una base de datos MySQL. El sistema debe ofrecer un menú interactivo que permita realizar las siguientes acciones:

Requerimientos:

1. **Agregar Producto:** El usuario debe poder añadir productos proporcionando su nombre, el ID de la categoría, el ID del proveedor, la cantidad en stock y el precio de venta. Los datos serán almacenados en la tabla productos de la base de datos MySQL.
2. **Listar Productos:** El sistema debe mostrar todos los productos almacenados en la base de datos, junto con su categoría, proveedor, cantidad en stock y precio.
3. **Actualizar Stock de un Producto:** El usuario debe poder actualizar la cantidad en stock de un producto existente, proporcionando el ID del producto y la nueva cantidad.
4. **Eliminar Producto:** El usuario podrá eliminar un producto de la base de datos proporcionando el ID del producto.
5. **Agregar Categoría:** Permitir al usuario añadir nuevas categorías a la tabla categorías, proporcionando el nombre de la categoría y una descripción.
6. **Listar Categorías:** Mostrar todas las categorías almacenadas en la base de datos.
7. **Agregar Proveedor:** Permitir al usuario añadir nuevos proveedores a la tabla proveedores, proporcionando el nombre del proveedor, dirección y número de teléfono.
8. **Listar Proveedores:** Mostrar todos los proveedores registrados en la base de datos.
9. **Salir del Sistema:** El usuario puede optar por salir del programa.

Requisitos del Sistema:

1. **Base de Datos MySQL:**

- **Base de datos:** TiendaDB
- **Tablas:**
 - productos: Contiene información sobre los productos.
 - categorias: Contiene las diferentes categorías de productos.
 - proveedores: Contiene los proveedores de los productos.

2. Conexión a la Base de Datos en Python:

- Usa el módulo `mysql.connector` para conectar Python con MySQL.
- Los detalles de conexión estarán definidos dentro de la clase `Database`.

Tablas en la Base de Datos:

- **productos:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `categoria_id`: INT (FOREIGN KEY referenciando la tabla `categorias`)
 - `proveedor_id`: INT (FOREIGN KEY referenciando la tabla `proveedores`)
 - `cantidad_stock`: INT
 - `precio`: DECIMAL(10, 2)
 - **categorias:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `descripcion`: TEXT
 - **proveedores:**
 - `id`: INT (AUTO_INCREMENT, PRIMARY KEY)
 - `nombre`: VARCHAR(100)
 - `direccion`: VARCHAR(200)
 - `telefono`: VARCHAR(15)
-

Funcionalidades del Menú Interactivo:

- **Opción 1:** Agregar Producto.
- **Opción 2:** Listar Productos.
- **Opción 3:** Actualizar Stock de un Producto.
- **Opción 4:** Eliminar Producto.
- **Opción 5:** Agregar Categoría.
- **Opción 6:** Listar Categorías.
- **Opción 7:** Agregar Proveedor.
- **Opción 8:** Listar Proveedores.
- **Opción 9:** Salir del Programa.