

# Conexión a MySQL



Ayúdame a crear un programa de Python para conectarse a una base de datos de MySQL



Claro, puedo ayudarte con eso. Aquí te dejo un ejemplo básico de cómo conectarte a una base de datos MySQL usando Python. Necesitarás el paquete `mysql-connector-python`. Si aún no lo has instalado, puedes hacerlo con:

```
bash
```

```
pip install mysql-connector-python
```

Luego, puedes usar el siguiente código para conectarte a la base de datos:

```
python
```

```
import mysql.connector
from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='localhost',          # Cambia esto por la dirección de tu servidor MySQL
            user='tu_usuario',         # Cambia esto por tu usuario de MySQL
            password='tu_contraseña',  # Cambia esto por tu contraseña de MySQL
            database='tu_base_de_datos' # Cambia esto por el nombre de tu base de datos
        )

        if connection.is_connected():
            print("Conexión exitosa a la base de datos MySQL")
            return connection

    except Error as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

def close_connection(connection):
    """Cierra la conexión con la base de datos MySQL."""
    if connection.is_connected():
        connection.close()
        print("Conexión cerrada")

# Ejemplo de uso
if __name__ == "__main__":
    conn = create_connection()
    # Aquí puedes realizar operaciones con la base de datos, por ejemplo, consultar datos.
    close_connection(conn)
```

## Explicación:

- ``create_connection``: Esta función establece una conexión con la base de datos usando los parámetros especificados (host, usuario, contraseña y nombre de la base de datos). Si la conexión es exitosa, devuelve el objeto de conexión; si ocurre un error, imprime el mensaje de error.
- ``close_connection``: Esta función cierra la conexión con la base de datos si está abierta.

Asegúrate de reemplazar ``localhost``, ``tu_usuario``, ``tu_contraseña`` y ``tu_base_de_datos`` con la información correspondiente a tu entorno.



Ahora quiero consultar los registros de una tabla curso y que los convierta en formato Dataframe y los exporte a un Excel



### Memoria actualizada

Para consultar los registros de una tabla en MySQL, convertirlos en un DataFrame de ``pandas`` y luego exportarlos a un archivo Excel, puedes usar el siguiente código:

1. Primero, asegúrate de tener los paquetes necesarios instalados:

bash

```
pip install pandas mysql-connector-python openpyxl
```

2. Aquí tienes un ejemplo de código para realizar la consulta y exportar los resultados a Excel:

python

```
import mysql.connector
import pandas as pd
from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='localhost',          # Cambia esto por la dirección de tu servidor MySQL
            user='tu_usuario',         # Cambia esto por tu usuario de MySQL
            password='tu_contraseña',  # Cambia esto por tu contraseña de MySQL
            database='tu_base_de_datos' # Cambia esto por el nombre de tu base de datos
        )
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

def fetch_data(query, connection):
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
    try:
        df = pd.read_sql(query, connection)
        return df
    except Error as e:
        print(f"Error al ejecutar la consulta: {e}")
        return None
```

```
def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL
    query = "SELECT * FROM curso;" # Cambia 'curso' por el nombre de tu tabla

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_curso.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()
```

## Explicación:

- `create_connection`: Conecta a la base de datos MySQL.
- `fetch_data`: Ejecuta la consulta SQL y convierte los resultados en un DataFrame de `pandas`.
- `export_to_excel`: Exporta el DataFrame a un archivo Excel usando el motor `openpyxl`.
- `main`: Define la consulta SQL, ejecuta la consulta, exporta los datos a Excel y cierra la conexión.

Asegúrate de ajustar los detalles de conexión y la consulta según tus necesidades.



Ahora modifica el código para que se muestren igual en una tabla de excel los datos de la tabla profesor



Claro, aquí está el código modificado para consultar los datos de la tabla `profesor` en lugar de la tabla `curso` y exportarlos a un archivo Excel. He añadido una sección para permitir que el usuario elija entre `curso` y `profesor`, y he cambiado el nombre del archivo Excel a `datos_profesor.xlsx`.

python

```
import mysql.connector
import pandas as pd
from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='localhost',          # Cambia esto por la dirección de tu servidor MySQL
            user='tu_usuario',         # Cambia esto por tu usuario de MySQL
        )
    except Error as e:
        print(f"Error al crear la conexión: {e}")
    return connection

def fetch_data(query, connection):
    """Obtiene los datos de la consulta SQL y los devuelve como un DataFrame de pandas."""
    if connection is None:
        print("No se pudo crear la conexión.")
        return None
    try:
        cursor = connection.cursor()
        cursor.execute(query)
        data = cursor.fetchall()
        cursor.close()
        return pd.DataFrame(data)
    except Error as e:
        print(f"Error al obtener los datos: {e}")
        return None

def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL
    query = "SELECT * FROM profesor;" # Cambia 'curso' por el nombre de tu tabla

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_profesor.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()
```

```

        password='tu_contraseña', # Cambia esto por tu contraseña de MySQL
        database='tu_base_de_datos' # Cambia esto por el nombre de tu base de datos
    )
    if connection.is_connected():
        return connection
except Error as e:
    print(f"Error al conectar a la base de datos: {e}")
    return None

def fetch_data(query, connection):
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
    try:
        df = pd.read_sql(query, connection)
        return df
    except Error as e:
        print(f"Error al ejecutar la consulta: {e}")
        return None

def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL para la tabla profesor
    query = "SELECT * FROM profesor;" # Cambia 'profesor' por el nombre de tu tabla

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_profesor.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()

```

## Cambios realizados:

- Se cambió la consulta SQL para obtener datos de la tabla `profesor`.
- Se cambió el nombre del archivo Excel a `datos_profesor.xlsx`.

Este código realiza una consulta para obtener todos los registros de la tabla `profesor`, los convierte en un DataFrame y luego los exporta a un archivo Excel. Asegúrate de ajustar los parámetros de conexión y consulta según tu entorno y tus necesidades.



Mejora el siguiente código para que solo me de los registros en donde se cumpla que el valor en el campo "apellido1" empiece con la letra "B"



Para modificar el código y que solo obtengas los registros donde el valor en el campo `apellido1` empiece con la letra "B", puedes ajustar la consulta SQL para incluir una cláusula `WHERE` con un patrón de búsqueda. Aquí tienes el código actualizado:

python

```
import mysql.connector
import pandas as pd
from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='localhost',          # Cambia esto por la dirección de tu servidor MySQL
            user='tu_usuario',         # Cambia esto por tu usuario de MySQL
            password='tu_contraseña',  # Cambia esto por tu contraseña de MySQL
            database='tu_base_de_datos' # Cambia esto por el nombre de tu base de datos
        )
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

def fetch_data(query, connection):
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
    try:
        df = pd.read_sql(query, connection)
        return df
    except Error as e:
        print(f"Error al ejecutar la consulta: {e}")
        return None

def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL para obtener registros donde apellido1 empiece con 'B'
    query = "SELECT * FROM profesor WHERE apellido1 LIKE 'B%';" # Cambia 'profesor' por
    # el nombre de tu tabla

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_profesor_B.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()
```

## Explicación:

- **Consulta SQL:** Se añadió la cláusula ``WHERE apellido1 LIKE 'B%`` a la consulta SQL. Esto filtra los registros para que solo se incluyan aquellos donde el campo ``apellido1`` empieza con la letra "B". El ``%`` es un comodín que representa cualquier secuencia de caracteres.

Este código ahora consultará los registros de la tabla ``profesor`` donde el campo ``apellido1`` empieza con "B" y los exportará a un archivo Excel llamado ``datos_profesor_B.xlsx``. Asegúrate de ajustar los detalles de la conexión y de la tabla según sea necesario.



Adecua el siguiente código para que solo se muestren las asignaturas de 6 créditos "import

```
mysql.connector
```

```
import pandas as pd
```

```
from mysql.connector import Error
```

```
def create_connection():
```

```
    """Crea una conexión con la base de datos MySQL."""
```

```
    try:
```

```
        connection = mysql.connector.connect(
```

```
            host='195.179.238.58', # Cambia esto por la dirección de tu servidor MySQL
```

```
            user='u927419088_admin', # Cambia esto por tu usuario de MySQL
```

```
            password='#Admin12345#', # Cambia esto por tu contraseña de MySQL
```

```
            database='u927419088_testing_sql' # Cambia esto por el nombre de tu base de datos
```

```
        )
```

```
        if connection.is_connected():
```

```
            return connection
```

```
    except Error as e:
```

```
        print(f"Error al conectar a la base de datos: {e}")
```

```
    return None
```

```
def fetch_data(query, connection):
```

```
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
```

```
    try:
```

```
        df = pd.read_sql(query, connection)
```

```
        return df
```

```
    except Error as e:
```

```
        print(f"Error al ejecutar la consulta: {e}")
```

```
    return None
```

```
def export_to_excel(df, file_path):
```

```
    """Exporta el DataFrame a un archivo Excel."""
```

```
    try:
```

```
        df.to_excel(file_path, index=False, engine='openpyxl')
```

```
        print(f"Datos exportados exitosamente a {file_path}")
```

```
    except Exception as e:
```

```
        print(f"Error al exportar a Excel: {e}")
```

```
def main():
    # Define tu consulta SQL para la tabla profesor
    query = "SELECT * FROM asignatura;" # Cambia 'profesor' por el nombre de tu tabla

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_asignatura.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()
```



Para adaptar el código para que solo muestre las asignaturas de 6 créditos, necesitas ajustar la consulta SQL para filtrar las asignaturas basadas en el campo correspondiente a los créditos. Supongamos que el campo que almacena los créditos se llama `creditos`. Aquí está el código actualizado:

python

```
import mysql.connector
import pandas as pd
from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='195.179.238.58', # Dirección de tu servidor MySQL
            user='u927419088_admin', # Usuario de MySQL
            password='#Admin12345#', # Contraseña de MySQL
            database='u927419088_testing_sql' # Nombre de tu base de datos
        )
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

def fetch_data(query, connection):
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
    try:
        df = pd.read_sql(query, connection)
        return df
    except Error as e:
        print(f"Error al ejecutar la consulta: {e}")
```

```

return None

def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL para obtener asignaturas con 6 créditos
    query = "SELECT * FROM asignatura WHERE credits = 6;" # Cambia 'credits' por el
    nombre correcto del campo

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_asignatura_6_creditos.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()

```

## Cambios realizados:

- **Consulta SQL:** La consulta ahora incluye una cláusula `WHERE` para filtrar las asignaturas con 6 créditos: `SELECT \* FROM asignatura WHERE credits = 6;`.
- **Nombre del archivo Excel:** Se cambió a `datos\_asignatura\_6\_creditos.xlsx` para reflejar el nuevo filtro.

Este código ahora obtendrá solo las asignaturas que tienen exactamente 6 créditos y exportará esos datos a un archivo Excel. Asegúrate de ajustar el nombre del campo `credits` si es diferente en tu base de datos.



adecua el código para saber que asignaturas son de pimer cuatrimestre



Para adaptar el código y filtrar las asignaturas que pertenecen al primer cuatrimestre, necesitas ajustar la consulta SQL para incluir una cláusula `WHERE` que filtre las asignaturas basadas en el campo correspondiente al cuatrimestre. Supongamos que el campo que almacena la información del cuatrimestre se llama `cuatrimestre`.

Aquí tienes el código actualizado:

python

```

import mysql.connector
import pandas as pd

```



```

from mysql.connector import Error

def create_connection():
    """Crea una conexión con la base de datos MySQL."""
    try:
        connection = mysql.connector.connect(
            host='195.179.238.58', # Dirección de tu servidor MySQL
            user='u927419088_admin', # Usuario de MySQL
            password='#Admin12345#', # Contraseña de MySQL
            database='u927419088_testing_sql' # Nombre de tu base de datos
        )
        if connection.is_connected():
            return connection
    except Error as e:
        print(f"Error al conectar a la base de datos: {e}")
        return None

def fetch_data(query, connection):
    """Ejecuta una consulta y devuelve los resultados en un DataFrame."""
    try:
        df = pd.read_sql(query, connection)
        return df
    except Error as e:
        print(f"Error al ejecutar la consulta: {e}")
        return None

def export_to_excel(df, file_path):
    """Exporta el DataFrame a un archivo Excel."""
    try:
        df.to_excel(file_path, index=False, engine='openpyxl')
        print(f"Datos exportados exitosamente a {file_path}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")

def main():
    # Define tu consulta SQL para obtener asignaturas del primer cuatrimestre
    query = "SELECT * FROM asignatura WHERE cuatrimestre = 1;" # Cambia 'cuatrimestre'
    por el nombre correcto del campo

    # Conectar a la base de datos
    connection = create_connection()

    if connection:
        # Consultar datos
        df = fetch_data(query, connection)

        if df is not None:
            # Exportar a Excel
            export_to_excel(df, 'datos_asignatura_primer_cuatrimestre.xlsx')

        # Cerrar la conexión
        connection.close()

if __name__ == "__main__":
    main()

```

## Cambios realizados:

- **Consulta SQL:** La consulta ahora incluye una cláusula `WHERE` para filtrar las asignaturas que están en el primer cuatrimestre: `SELECT \* FROM asignatura WHERE cuatrimestre = 1;`.
- **Nombre del archivo Excel:** Se cambió a `datos\_asignatura\_primer\_cuatrimestre.xlsx` para reflejar el nuevo filtro.

Este código consultará solo las asignaturas que están en el primer cuatrimestre y exportará esos datos a un archivo Excel. Asegúrate de ajustar el nombre del campo `cuatrimestre` si es diferente en tu base de datos.

ChatGPT puede cometer errores. Comprueba la información importante.