

Realizar estas tablas en un SGBD:

MemSch.MemTraMDADet

Columns

- idMDA (bigint, not null)
- claNodo (PK, varchar(10), not null)
- fecha (PK, date, not null)
- hora (PK, tinyint, not null)
- pml (decimal(10,5), null)
- pml_ene (decimal(10,5), null)
- pml_per (decimal(10,5), null)
- pml_cng (decimal(10,5), null)
- FechaUltimaMod (datetime, null)
- NombrePcMod (nchar(30), null)
- ClaUsuarioMod (int, null)

MemSch.MemTraMTRDet

Columns

- idMTR (bigint, null)
- claNodo (PK, varchar(10), not null)
- fecha (PK, date, not null)
- hora (PK, tinyint, not null)
- pml (decimal(10,5), null)
- pml_ene (decimal(10,5), null)
- pml_per (decimal(10,5), null)
- pml_cng (decimal(10,5), null)
- FechaUltimaMod (datetime, null)
- NombrePcMod (nchar(10), null)
- ClaUsuarioMod (int, null)

MemSch.MemTraTcDet

Columns

- idTc (int, null)
- fecha (PK, date, not null)
- valor (decimal(10,6), null)
- FechaUltimaMod (datetime, null)
- NombrePcMod (varchar(30), null)
- ClaUsuarioMod (int, null)

MemSch.MemTraTBFinVw

Columns

- fecha (date, null)
- TbFin (numeric(38,14), null)
- TbFinTGR (numeric(38,9), null)

```
1 CREATE TABLE MemTraMDADet (
2     idMDA BIGINT NOT NULL,
3     claNodo VARCHAR(10) NOT NULL,
4     fecha DATE NOT NULL,
5     hora SMALLINT NOT NULL,
6     pml DECIMAL(10,5),
7     pml_ene DECIMAL(10,5),
8     pml_per DECIMAL(10,5),
9     pml_cng DECIMAL(10,5),
10    FechaUltimaMod TIMESTAMP,
11    NombrePcMod NCHAR(30),
12    ClaUsuarioMod INT,
13    PRIMARY KEY (claNodo, fecha, hora)
14 );
15
16 CREATE TABLE MemTraMTRDet (
17     idMTR BIGINT,
18     claNodo VARCHAR(10) NOT NULL,
19     fecha DATE NOT NULL,
20     hora SMALLINT NOT NULL,
21     pml DECIMAL(10,5),
22     pml_ene DECIMAL(10,5),
23     pml_per DECIMAL(10,5),
24     pml_cng DECIMAL(10,5),
25     FechaUltimaMod TIMESTAMP,
26     NombrePcMod NCHAR(10),
27     ClaUsuarioMod INT,
28     PRIMARY KEY (claNodo, fecha, hora)
29 );
```

```
31 CREATE TABLE MemTraTcDet (
32     idTc INT,
33     fecha DATE NOT NULL,
34     valor DECIMAL(10,6),
35     FechaUltimaMod TIMESTAMP,
36     NombrePcMod VARCHAR(30),
37     ClaUsuarioMod INT,
38     PRIMARY KEY (fecha)
39 );
40
41 CREATE TABLE MemTraTBFin (
42     fecha DATE NOT NULL,
43     TbFin NUMERIC(38,14),
44     TbFinTGR NUMERIC(38,9),
45     PRIMARY KEY (fecha)
46 )
47
```

Python:

Por medio de Python carga la información de los csv a tu Servidor de Base de datos a sus correspondientes tablas

```
1  from sqlalchemy import create_engine
2  from datetime import datetime
3  import pandas as pd
4
5  engine = create_engine('postgresql+psycopg2://usr:pwd@localhost:5432/MemSch')
6
7  def populate(obj):
8      df = pd.read_csv(obj['file_name'])
9      df.columns = [x.lower() for x in df.columns]
10
11     if 'fechaultimamod' not in df.columns and obj['index']:
12         df['fechaultimamod'] = datetime.today().strftime('%Y-%m-%d')
13     if 'nombrepcomod' not in df.columns and obj['index']:
14         df['nombrepcomod'] = 'hgm'
15     if 'clausuariomod' not in df.columns and obj['index']:
16         df['clausuariomod'] = 86
17
18     df.to_sql(obj['table_name'].lower(), engine, if_exists='append', index=obj['index'], index_label=obj['index_label']).
19
20     files = [
21 >         { ...
27 >         { ...
33 >         { ...
39 >         { ...
45     ]
46
47     if __name__ == "__main__":
48         for file in files:
49             populate(file)
50
51             print(f'finished: {file["table_name"]}')
52
53     print('finished')
```

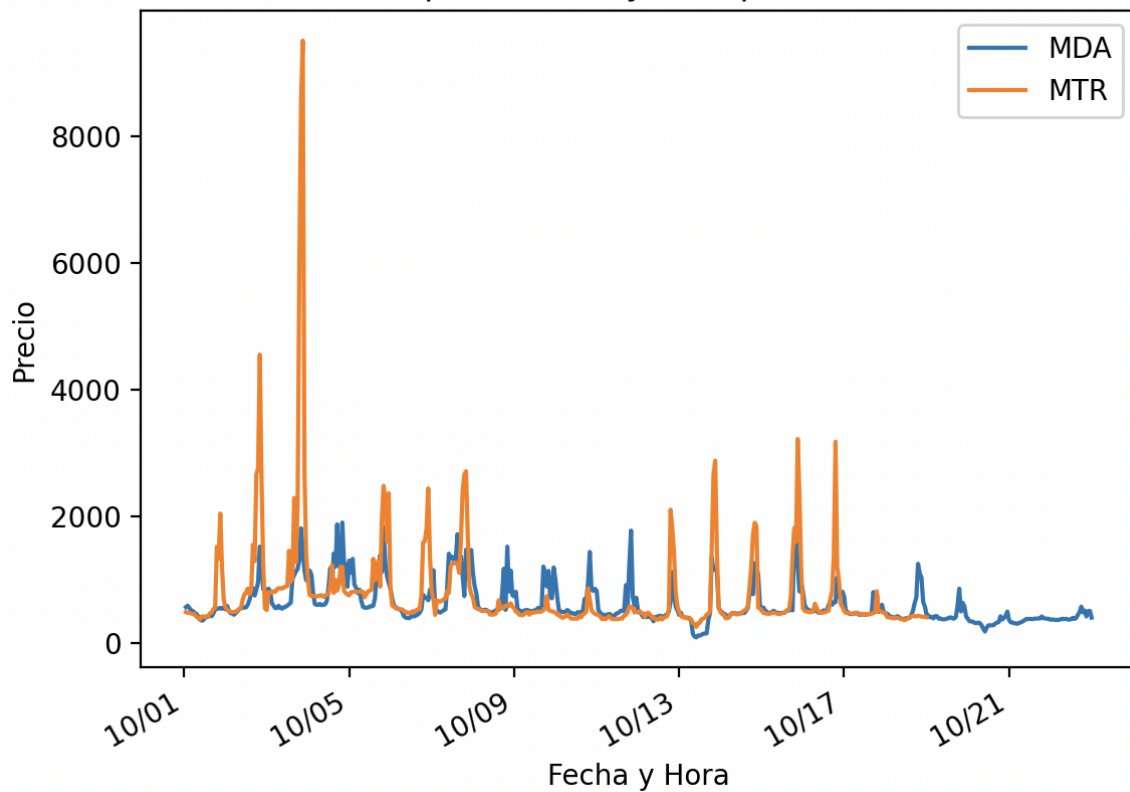
Graficar la evolución del precio MDA y MTR del nodo 01ANS-85

```

1 from sqlalchemy import create_engine, select, MetaData
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import matplotlib.dates as mdates
5
6 def main():
7     # Conexión a la base de datos
8     engine = create_engine('postgresql+psycopg2://usr:pwd@localhost/MemSch')
9     metadata = MetaData()
10    metadata.reflect(bind=engine)
11
12    # Cargar datos
13    mda = pd.read_sql(select(metadata.tables['memtramdadet']).select_from(metadata.tables['memtramdadet']), engine)
14    mtr = pd.read_sql(select(metadata.tables['memtramtrdet']).select_from(metadata.tables['memtramtrdet']), engine)
15    tc = pd.read_sql(select(metadata.tables['memtratcdet']), engine)
16    tbfin = pd.read_sql(select(metadata.tables['memtratbfin']), engine)
17
18    mda_01ans85 = mda[mda['clanodo'] == '01ANS-85']
19    mda_01ans85['datetime'] = pd.to_datetime(mda_01ans85['fecha']) + pd.to_timedelta(mda_01ans85['hora'], unit='h')
20    mtr_01ans85 = mtr[mtr['clanodo'] == '01ANS-85']
21    mtr_01ans85['datetime'] = pd.to_datetime(mtr_01ans85['fecha']) + pd.to_timedelta(mtr_01ans85['hora'], unit='h')
22
23    # Gráfica evolución del precio MDA y MTR para nodo 01ANS-85
24    fig, ax = plt.subplots()
25    ax.plot(mda_01ans85['datetime'], mda_01ans85['pml'], label="MDA")
26    ax.plot(mtr_01ans85['datetime'], mtr_01ans85['pml'], label="MTR")
27    ax.legend()
28    ax.xaxis.set_major_formatter(mdates.DateFormatter('%m/%d'))
29    fig.autofmt_xdate()
30    plt.title('Evolución del precio MDA y MTR para nodo 01ANS-85')
31    plt.xlabel('Fecha y Hora')
32    plt.ylabel('Precio')
33    plt.show()

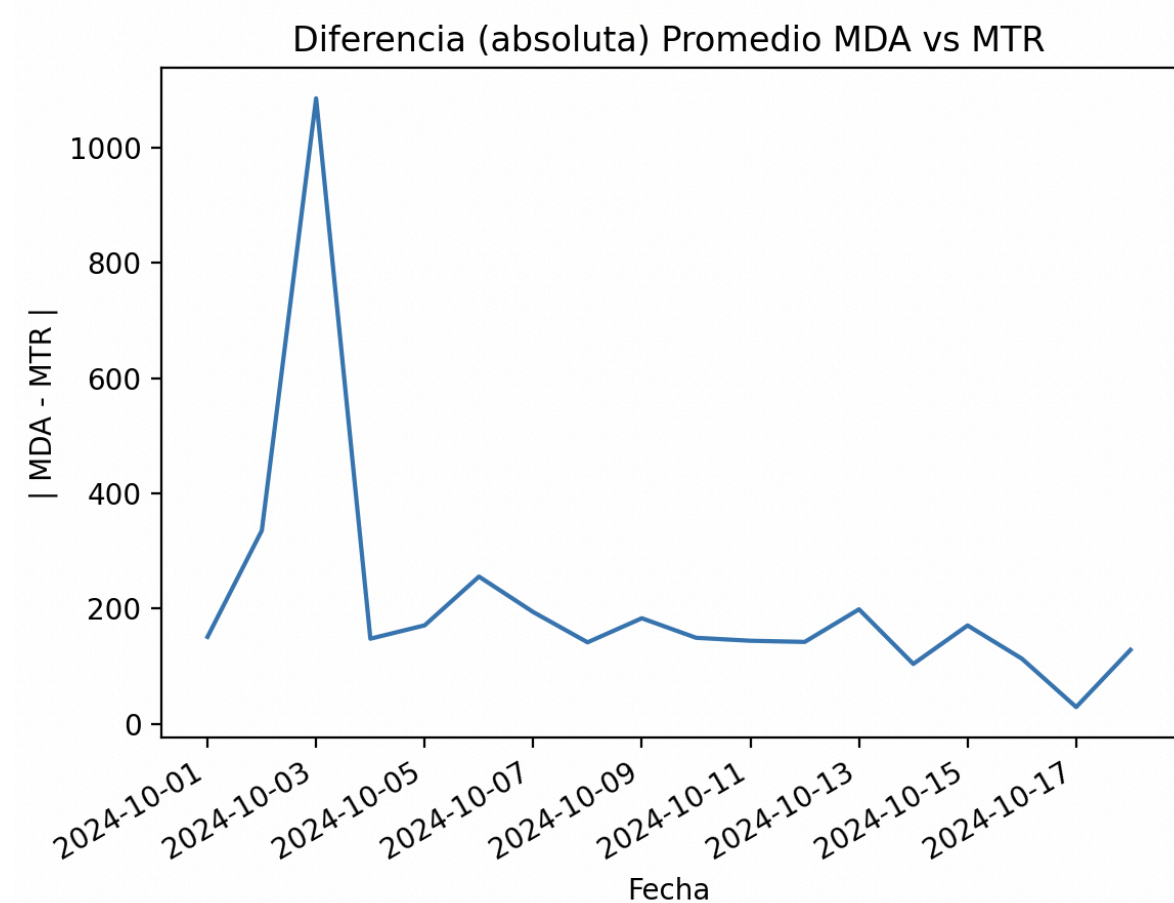
```

Evolución del precio MDA y MTR para nodo 01ANS-85



Graficar la diferencia promedio por día del precio entre el MDA y MTR de todos los nodos agrupados por día

```
35     # Diferencia promedio diaria entre MDA y MTR
36     fig, ax = plt.subplots()
37     mda_prices = mda.groupby('fecha')['pml'].mean()
38     mtr_prices = mtr.groupby('fecha')['pml'].mean()
39     diff_avg = abs(mda_prices - mtr_prices).reset_index(name='diff_avg')
40     ax.plot(diff_avg['fecha'], diff_avg['diff_avg'])
41     fig.autofmt_xdate()
42     plt.title(('Diferencia (absoluta) Promedio MDA vs MTR'))
43     plt.xlabel('Fecha')
44     plt.ylabel('| MDA - MTR |')
45     plt.show()
```



Une en una sola tabla MemTraMdaDet y MemTraMtrDet agregando una columna al inicio que se llame origen y pueda ser “MDA” o “MTR”

```

47     # Unir MDA y MTR
48     mda['origen'] = 'MDA'
49     mtr['origen'] = 'MTR'
50     mda_mtr = pd.concat([mda, mtr], ignore_index=True)

```

Une esta tabla creada en el paso anterior y la de TC, para agregar a una columna de valor para tener tambien el tc

```

52     # Unir con TC
53     merged = pd.merge(mda_mtr, tc, on='fecha', how='left', suffixes=('', '_tc'))

```

Genera un DataFrame Nodo, fecha, hora, pml, tbfin de los datos que el pml sea mayor que la tbfin

```

55     # DataFrame Nodo, fecha, hora, pml, tbfin con pml > tbfin
56     filtered = pd.merge(merged, tbfin, on='fecha', how='left')
57     filtered = filtered[filtered['pml'] > filtered['tbfin']][['clanodo', 'fecha', 'hora', 'pml', 'tbfin']]

```

Genera un dataframe que tenga el promedio diario de los precios del pml

```

59     # DataFrame promedio diario de los precios del pml
60     daily_avg_pml = merged.groupby('fecha')['pml'].mean().reset_index()

```

Grafica el precio del Nodo y el precio de la tbfin por fecha y hora

No es posible, las fechas de los registros de tbfin y los nodos no están en el mismo espacio de tiempo, la información de tbfin va de “2020-05-01” a “2023-01-18”, los registros de los nodos empiezan en “2024-10-01”

SQL:

Proporcionar el Query que me traiga el precio del nodo (pml) en MDA y el precio en MTR del nodo 01ANS-85, ordenado por nodo (ascendente) , por fecha (descendente) y ascendente por hora

```
1  -- Query 1: Precio del nodo 01ANS-85 en MDA y MTR, ordenado por nodo (asc), fecha (desc), y hora (asc)
2  SELECT mda.claNodo, mda.fecha, mda.hora, mda.pml AS pml_mda, mtr.pml AS pml_mtr
3  FROM MemTraMDADet mda
4  JOIN MemTraMTRDet mtr ON mda.claNodo = mtr.claNodo AND mda.fecha = mtr.fecha AND mda.hora = mtr.hora
5  WHERE mda.claNodo = '01ANS-85'
6  ORDER BY mda.claNodo ASC, mda.fecha DESC, mda.hora ASC;
```

	clanodo	fecha	hora	pml_mda	pml_mtr
1	01ANS-85	2024-10-18	1	460.43000	404.54000
2	01ANS-85	2024-10-18	2	431.93000	398.80000
3	01ANS-85	2024-10-18	3	410.34000	394.79000
4	01ANS-85	2024-10-18	4	405.43000	390.86000
5	01ANS-85	2024-10-18	5	397.80000	382.51000
6	01ANS-85	2024-10-18	6	402.87000	389.87000
7	01ANS-85	2024-10-18	7	417.77000	392.01000
8	01ANS-85	2024-10-18	8	400.98000	387.53000
9	01ANS-85	2024-10-18	9	381.08000	370.20000
10	01ANS-85	2024-10-18	10	378.01000	369.90000

Proporciona el Query que me traiga el precio promedio por nodo en MTR y en MDA, y la diferencia de estos 2 precios promedio, ordenado por diferencia descendente

```
8  -- Query 2: Precio promedio por nodo en MTR y MDA, y diferencia de estos dos precios promedio, ordenado por diferencia
9  SELECT mda.claNodo,
10         AVG(mda.pml) AS avg_pml_mda,
11         AVG(mtr.pml) AS avg_pml_mtr,
12         ABS(AVG(mda.pml) - AVG(mtr.pml)) AS diff_pml
13  FROM MemTraMDADet mda
14  JOIN MemTraMTRDet mtr ON mda.claNodo = mtr.claNodo AND mda.fecha = mtr.fecha AND mda.hora = mtr.hora
15  GROUP BY mda.claNodo
16  ORDER BY diff_pml DESC;
```

	clanodo	avg_pml_mda	avg_pml_mtr	diff_pml
1	02JUC-115	661.7642824074074074	814.1389351851851852	152.3746527777777778
2	02JUD-115	658.7420370370370370	810.4318518518518519	151.6898148148148149
3	02JUD-230	659.1350925925925926	810.6565277777777778	151.5214351851851852
4	02JTL-115	700.0163194444444444	850.9209490740740741	150.9046296296296297
5	02JUY-115	680.6081712962962963	827.2230555555555556	146.6148842592592593
6	02KIM-115	669.1291666666666667	813.7690740740740741	144.6399074074074074
7	02JRN-115	659.8292824074074074	784.1061111111111111	124.2768287037037037
8	01ATG-69	613.6244212962962963	734.9962962962962963	121.3718750000000000
9	05CFN-115	518.6126620370370370	637.8737731481481481	119.2611111111111111
10	05CEM-115	523.1079861111111111	641.7880324074074074	118.6800462962962963

Proporciona el precio de nodo en dlls tomando como tipo de cambio el campo valor que está en la tabla MEMTraTcDet

```
18 -- Query 3: Precio de nodo en dólares tomando el tipo de cambio de MemTraTcDet
19 SELECT mda.claNodo, mda.fecha, mda.hora, mda.pml, mda.pml / tc.valor AS pml_usd
20 FROM MemTraMDADet mda
21 JOIN MemTraTcDet tc ON mda.fecha = tc.fecha
22 ORDER BY mda.claNodo, mda.fecha, mda.hora;
```

	clanodo	fecha	hora	pml	pml_usd
1	01ALR-85	2024-10-01	1	557.13000	28.3242754083692176
2	01ALR-85	2024-10-01	2	582.21000	29.5993329842346350
3	01ALR-85	2024-10-01	3	544.69000	27.6918305820627666
4	01ALR-85	2024-10-01	4	508.12000	25.8326258153403458
5	01ALR-85	2024-10-01	5	497.76000	25.3059273908600538
6	01ALR-85	2024-10-01	6	466.82000	23.7329496636959384
7	01ALR-85	2024-10-01	7	456.06000	23.1859153927106158
8	01ALR-85	2024-10-01	8	414.40000	21.0679369792116809
9	01ALR-85	2024-10-01	9	373.70000	18.9987645973248194
10	01ALR-85	2024-10-01	10	346.34000	17.6077926963807277
11	01ALR-85	2024-10-01	11	349.30000	17.7582779605179540

Proporciona el listado de nodos por fecha, hora, de los precios de los nodos en mda y mtr, junto con el tipo de cambio y el precio de la tbfin

```
24 -- Query 4: Listado de nodos con precios en MDA y MTR, tipo de cambio y precio de la tbfin
25 SELECT mda.claNodo, mda.fecha, mda.hora, mda.pml AS pml_mda, mtr.pml AS pml_mtr, tc.valor AS tipo_cambio, tb.TbFin
26 FROM MemTraMDADet mda
27 JOIN MemTraMTRDet mtr ON mda.claNodo = mtr.claNodo AND mda.fecha = mtr.fecha AND mda.hora = mtr.hora
28 JOIN MemTraTcDet tc ON mda.fecha = tc.fecha
29 JOIN MemTraTBFin tb ON mda.fecha = tb.fecha
30 ORDER BY mda.claNodo, mda.fecha, mda.hora;
```

Query sin resultados por falta de registros de tbfin en el espacio de tiempo correcto

Nota:

Todo el código fuente utilizado está en el repositorio de github del sistema de gestión de inventario

<https://github.com/HeribertoGM/SGI>