

# Case: zonne-opbrengst vergelijken

met pandas dataframes

# Data verzamelen

Plant-data (niet de opbrengst) in MySQL:

- tabel **inverters**: o.a. id (primary key) en "inverter name"
- tabel **invstrings**: o.a. string nummers (str\_num) gekoppeld aan inverter\_id (foreign key),
- tabel **modules**: o.a. **tilt/azimuth/amount** voor de modules gekoppeld aan invstring\_id (foreign key).

Verkrijg alle data voor tilt/azimuth/amount met SQL query:

```
SELECT m.tilt, m.azimuth, m.amount, invstrings.str_num as str_num, inverters.name
```

```
FROM modules as m
```

```
INNER JOIN invstrings ON m.invstring_id = invstrings.id
```

```
INNER JOIN inverters ON invstrings.inverter_id = inverters.id and inverters.plant_name= %s and inverters.active = 'yes'
```

```
# Execute de query and save result
```

```
cursor.execute(query, params)
```

```
result = cursor.fetchall()
```

```
# Data in pandas dataframe
```

```
df = pd.DataFrame(result)
```

## Overzicht per string

	tilt	azimuth	amount	str_num	inverter_name
0	12.0	56	24	1	05-185KTL
1	12.0	56	24	2	05-185KTL
2	12.0	236	26	3	05-185KTL
3	12.0	236	26	4	05-185KTL
4	12.0	56	24	5	05-185KTL
..	...	...	...	...	...
174	12.0	56	26	13	06-185KTL
175	12.0	56	26	14	06-185KTL
176	12.0	236	26	15	06-185KTL
177	12.0	236	26	16	06-185KTL
178	12.0	56	26	17	06-185KTL

[179 rows x 5 columns]

# Verkrijgen opbrengst (power) per string en voeg kolom toe

```
from prometheus_pandas import query

p = query.Prometheus('localhost:9090') # connection with Prometheus db

power_column = []

for index,row in df.iterrows(): # iterate per string

    df_voltage = get_voltage(p,...)      # all 10 minute values in period
    df_current = get_current(p,...)      # all 10 minute values in period

    df_power = df_voltage * df_current #  $P = V * I$ 

        Voltage                Current                Power
2025-01-26 08:20:00    955.0    2025-01-26 08:20:00    0.21    2025-01-26 08:20:00    200.550
2025-01-26 08:30:00    955.2    2025-01-26 08:30:00    0.41    2025-01-26 08:30:00    391.632
2025-01-26 08:40:00    953.8    2025-01-26 08:40:00    0.41    2025-01-26 08:40:00    391.058
...                    ...                    ...

power_column.append(df_power.sum()[0] / row['amount'])
```

Voeg kolom toe:

```
df['power_per_module'] = power_column
```

## Overzicht per string met power per paneel (voor periode)

	tilt	azimuth	amount	str_num	inverter_name	Power_per_panel
0	12.0	56	24	1	05-185KTL	29107.033792
1	12.0	56	24	2	05-185KTL	31641.686792
2	12.0	236	26	3	05-185KTL	46805.352769
3	12.0	236	26	4	05-185KTL	44686.744346
..	...	...	...	...	...	...
174	12.0	56	26	13	06-185KTL	30928.503115
175	12.0	56	26	14	06-185KTL	31543.251500
176	12.0	236	26	15	06-185KTL	48194.926923
177	12.0	236	26	16	06-185KTL	45152.743577
178	12.0	56	26	17	06-185KTL	30058.958308

[179 rows x 6 columns]

En dan nog groeperen bij (tilt,azimuth):

```
dfs = {(tilt,azimuth): group for (tilt, azimuth), group in df.groupby(['tilt','azimuth'])}
```

## Meer groeperen en overzicht maken

```
for (tilt,azimuth), df in common_dfs.items():  
    grouped_strings = df.groupby('inverter_name')  
    # min(), max() en afwijking (%) rapporteren
```

## Overzicht strings met tilt 12 graden en azimuth 236 graden

Inverter	String/Min	String/Max	Deviation
01-185KTL	4/42693.75	15/48197.47	11.42 %
02-185KTL	10/42670.86	3/47909.17	10.93 %
03-185KTL	16/42242.78	2/47446.10	10.97 %
04-185KTL	10/43062.50	9/47440.16	9.23 %
05-185KTL	7/44517.79	15/48915.74	8.99 %
06-185KTL	4/43831.25	15/48194.93	9.05 %
07-185KTL	7/45220.04	15/49064.60	7.84 %
08-185KTL	17/30881.90	9/47836.83	35.44 %
09-185KTL	16/38980.92	15/44794.27	12.98 %
10-185KTL	7/40649.27	15/47892.93	15.12 %
11-185KTL	10/41409.88	9/45972.71	9.93 %

# Referentie

[https://pandas.pydata.org/Pandas\\_Cheat\\_Sheet.pdf](https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf)