

Patstat register - Exercise 1

In this exercise we will gradually build a query using ORM that gives us all the Swiss Applicants related to European patents granted after the 31 of December 2019.

The publications table

First we need to learn about the table `reg102_pat_publn`. This table contains details about european and international publications that are visible in the European Patent Register.

EP publications from 2020 onwards

We will use the filter functionality to see only EP publications published since 2020, with these filters.

- `publn_auth == 'EP'` : Only includes records where the publication authority is 'EP'.
- `publn_date > '2019-12-31'` : Only includes records where the publication date is after December 31, 2019.
- `publn_date < '2099-12-31'` : PATSTAT uses the date 2099-12-31 instead of `null` to indicate that there is no data information for a specific record. With this condition we filter out publications with an unknown date.

The results are ordered by the `publn_date` column in ascending order.

```
In [5]: # Importing the patstat client
from epo.tipdata.patstat import PatstatClient

# Initialize the PATSTAT client
patstat = PatstatClient()

# Access ORM
db = patstat.orm()
# Importing tables as models
from epo.tipdata.patstat.database.models import REG102_PAT_PUBLN
```

```
In [8]: q = db.query(
    REG102_PAT_PUBLN.publn_auth,
    REG102_PAT_PUBLN.publn_nr,
    REG102_PAT_PUBLN.publn_date
).filter(
    REG102_PAT_PUBLN.publn_kind == 'B1', # shows only granted patents
    REG102_PAT_PUBLN.publn_auth == 'EP',
    REG102_PAT_PUBLN.publn_date > '2019-12-31',
    REG102_PAT_PUBLN.publn_date < '9999-12-31' # eliminates publications without a date
).order_by(
    REG102_PAT_PUBLN.publn_date
)
# Creating a dataframe with the results
res = patstat.df(q)

res
```

Out[8]:

	publn_auth	publn_nr	publn_date
0	EP	3093161	2020-01-01
1	EP	3257446	2020-01-01
2	EP	3229093	2020-01-01
3	EP	2813186	2020-01-01
4	EP	2849021	2020-01-01
...
450042	EP	3638444	2024-03-13
450043	EP	3308606	2024-03-13
450044	EP	4047183	2024-03-13
450045	EP	3846520	2024-03-13
450046	EP	4079540	2024-03-13

450047 rows × 3 columns

Introducing the reg107_parties table

The goal of this exercise is to find out the European patents that mention an applicant or proprietor from Switzerland. For this we need to work with the `reg107_parties` table in the PATSTAT Register database. This table stores information about the parties involved in patent applications.

Finding Swiss applicants

We will make a simple query in the `reg107_parties` to find all the parties with a place of residence in Switzerland. There can be three types of parties:

- **Applicant or proprietor ("A")**
- **Inventor ("I")**
- **Agent or representative ("R")**
- **Opponent ("O")**

We will query the table 107 with these filters:

- `REG107_PARTIES.country == 'CH'` : Only includes records that specify a place of business or residence in Switzerland.
- `REG107_PARTIES.type == 'A'` : Only includes records listed as applicant or proprietor.
- `REG107_PARTIES.is_latest == 'Y'` : the parties for an application change over time. This field is a Y/N flag. 'Y' indicates that the record belongs to the latest (current or most recent) set of applicants, inventors, representatives, or opponents.

```
In [10]: from epo.tipdata.patstat.database.models import REG107_PARTIES

q = db.query(
    REG107_PARTIES.name,
    ).filter(
        REG107_PARTIES.country == 'CH',
        REG107_PARTIES.type == 'A',
        REG107_PARTIES.is_latest == 'Y'
    ).order_by(
        REG107_PARTIES.name
    )

# Creating a dataframe with the results
res = patstat.df(q)

res[0:10]
```

Out [10]:

	name
0	'PLANET' MATTHIAS JAGGI
1	'PLANET' MATTHIAS JAGGI
2	'BRUGG'-KABEL AG
3	'BRUGG'-KABEL AG
4	'Brugg' Drahtseil AG
5	'HOLDERBANK' Financière Glarus AG
6	'HOLDERBANK' Financière Glarus AG
7	'HOLDERBANK' Financière Glarus AG
8	'HOLDERBANK' Financière Glarus AG
9	'HOLDERBANK' Financière Glarus AG

Understanding duplicates in the parties table

Unfortunately there is no unique identifier for applicants, inventors, or proprietors in the field of patent data. Each patent application mentions the applicants with their names and addresses. It often happens that the same applicant is filed with variations of the same address, or different addresses, in different patent applications. The applicant name itself can sometimes be spelled differently. This typically creates multiple records in the parties table for one single legal entity or a single person.

Please take into consideration this fact for all your patent data analysis.

Joining parties to publications via the applications table

If you look at the logical model diagram of patstat in the documentation, you will see that the `reg107_parties` and `reg102_pat_publication` tables are not related. In Patstat Register the central table is `reg101_appln`, which contains data about the European and International patent applications in the register.

We will then join table 102, 101, and 107 to get the desired query.

Let's first join tables 101 and 102 to get the publications from 2020 and later, and get the application ID for each publication. This application ID will later be needed for joining tables 101 and 107.

```
In [11]: from epo.tipdata.patstat.database.models import REG101_APPLN

q = db.query(
    REG102_PAT_PUBLN.publn_auth,
    REG102_PAT_PUBLN.publn_nr,
    REG102_PAT_PUBLN.publn_date,
    REG101_APPLN.id,
    REG101_APPLN.appln_nr
).join(
    REG101_APPLN, REG102_PAT_PUBLN.id == REG101_APPLN.id
).filter(
    REG102_PAT_PUBLN.publn_kind == 'B1',
    REG102_PAT_PUBLN.publn_auth == 'EP',
    REG102_PAT_PUBLN.publn_date > '2019-12-31',
    REG102_PAT_PUBLN.publn_date < '9999-12-31'
).order_by(
    REG102_PAT_PUBLN.publn_date
)

# Creating a dataframe with the results
res = patstat.df(q)
res
```

Out[11]:

	publn_auth	publn_nr	publn_date	id	appln_nr
0	EP	3360402	2020-01-01	18151092	18151092
1	EP	3218299	2020-01-01	14796764	14796764
2	EP	3373874	2020-01-01	16791613	16791613
3	EP	2581035	2020-01-01	12006033	12006033
4	EP	3165259	2020-01-01	16197566	16197566
...
450042	EP	4217538	2024-03-13	21739107	21739107
450043	EP	4225143	2024-03-13	21782557	21782557
450044	EP	4096935	2024-03-13	21704032	21704032
450045	EP	3562396	2024-03-13	16925493	16925493
450046	EP	3893434	2024-03-13	18944755	18944755

450047 rows × 5 columns

Our final query

We are reaching the end of the exercise. We are ready now to build a query that connects the 101, 102 and 107 tables, and looks for Swiss applicants related to European patents granted from 2020 and onwards.

The query performs a double join to connect three tables: REG101_APPLN , REG102_PAT_PUBLN , and REG107_PARTIES .

1. First Join:

- Connects REG101_APPLN and REG102_PAT_PUBLN using REG102_PAT_PUBLN.id == REG101_APPLN.id .

2. Second Join:

- Connects the resulting dataset with REG107_PARTIES using REG101_APPLN.id == REG107_PARTIES.id .

```
In [13]: q = db.query(
    REG102_PAT_PUBLN.publn_auth,
    REG102_PAT_PUBLN.publn_nr,
    REG102_PAT_PUBLN.publn_date,
    REG101_APPLN.id,
    REG101_APPLN.appln_nr,
    REG107_PARTIES.name
).join(
    REG101_APPLN, REG102_PAT_PUBLN.id == REG101_APPLN.id
).join(
    REG107_PARTIES, REG101_APPLN.id == REG107_PARTIES.id
).filter(
    REG102_PAT_PUBLN.publn_kind == 'B1',
    REG102_PAT_PUBLN.publn_auth == 'EP',
    REG102_PAT_PUBLN.publn_date > '2019-12-31',
    REG102_PAT_PUBLN.publn_date < '9999-12-31',
    REG107_PARTIES.country == 'CH',
    REG107_PARTIES.type == 'A',
    REG107_PARTIES.is_latest == 'Y'
).order_by(
    REG107_PARTIES.name
)

# Creating a dataframe with the results
res = patstat.df(q)
res
```

Out[13]:

	publn_auth	publn_nr	publn_date	id	appln_nr	name
0	EP	3114461	2022-12-21	15707967	15707967	1 Drop SA
1	EP	3669190	2023-11-01	18765185	18765185	1MED SA
2	EP	3833741	2023-11-15	19769576	19769576	1MED SA
3	EP	2569154	2020-01-01	11724365	11724365	3A Composites International AG
4	EP	3718498	2021-11-24	19167161	19167161	3D MED AG
...
17743	EP	2898534	2021-11-24	13815134	13815134	École Polytechnique Fédérale de Lausanne (EPFL)
17744	EP	3697740	2022-08-17	18808105	18808105	École Polytechnique Fédérale de Lausanne (EPFL)
17745	EP	2770899	2021-03-10	12844532	12844532	École Polytechnique Fédérale de Lausanne (EPFL)
17746	EP	3884013	2024-01-10	19801607	19801607	École Polytechnique Fédérale de Lausanne (EPFL)
17747	EP	4112494	2024-01-10	22181300	22181300	éscale cosmétique

17748 rows × 6 columns

In []: