The Patstat library - Lesson 2

This notebook expands on the first lesson about Patstat. We will take a look at the applications table, which is the main table of the database schema of Patstat.

The applications table

When working with Patstat you should be familiar with its rich data structure. The goal of this course is not to explain the whole set of tables and fields, since this can be found in the official documentation. We will, however, take a look at some of the main tables and work with them to perform data analysis.

Table `tls201_appln` , that we will be refering to as the `applications table` is the central table in the databse schema for PATSTAT global. Almost all other tables in the schema have a direct relationship with the applications table. Let's take a look at this table

Viewing all the fields with SQL

We saw in lesson 1 that we can query patstat using SQL language, using the method `sql_query` . We are going to take advantage of that and query using `SELECT * FROM tls201_appln` , which gives us all the fields in a given table.

In [1]:

```
# Importing the patstat client
from epo.tipdata.patstat import PatstatClient

# Initialize the PATSTAT client
patstat = PatstatClient()

# Access ORM
db = patstat.orm()
```

In [ ]:

```
# Querying with SQL for all the fields in the applications table

res = patstat.sql_query("""
SELECT
    *
FROM
    tls201_appln
""")

# Printing the number of fields
print (f"Number of fields in the applications table", len(res[
0]))

# Showing the first result
res[0]
```

The fields in the applications table

We can see that table `tls201_appln` in the PATSTAT database contains 27 fields. Below you can see a description of each field.

1. **appln_id**: A unique identifier for the patent application. This is an internal number used to uniquely identify each application within the database.
2. **appln_auth**: The patent authority or office where the application was filed. For example, 'EP' stands for the European Patent Office.
3. **appln_nr**: The application number assigned by the patent authority. This number is unique within the context of the authority.
4. **appln_kind**: The kind of patent document, often represented by a letter (e.g., 'A' for a published application, 'B' for a granted patent).
5. **appln_filing_date**: The date on which the patent application was filed with the patent authority.
6. **appln_filing_year**: The year in which the patent application was filed, extracted from the filing date.
7. **appln_nr_epodoc**: The application number in the EPODOC format, a standardized format used by the European Patent Office.
8. **appln_nr_original**: The original application number as assigned by the patent authority.
9. **ipr_type**: The type of intellectual property right, such as 'PI' for patent of invention.
10. **receiving_office**: The receiving office for the application, which is typically used in the context of international patent applications.
11. **internat_appln_id**: Identifier for the international application, if applicable.
12. **int_phase**: Indicates whether the application is in the international phase ('Y' for yes, 'N' for no).
13. **reg_phase**: Indicates whether the application is in the regional phase ('Y' for yes, 'N' for no).
14. **nat_phase**: Indicates whether the application is in the national phase ('Y' for yes, 'N' for no).
15. **earliest_filing_date**: The earliest filing date among all related applications in the same patent family.
16. **earliest_filing_year**: The year of the earliest filing date.
17. **earliest_filing_id**: Identifier for the earliest related application.
18. **earliest_publn_date**: The earliest publication date of the application.
19. **earliest_publn_year**: The year of the earliest publication date.
20. **earliest_pat_publn_id**: Identifier for the earliest related publication.
21. **granted**: Indicates whether the application has been granted ('Y' for yes, 'N' for no).
22. **docdb_family_id**: Identifier for the DOCDB patent family, which groups related patent documents across different countries.
23. **inpadoc_family_id**: Identifier for the INPADOC patent family, a broader grouping of related patent documents.
24. **docdb_family_size**: The number of documents in the DOCDB patent family.
25. **nb_citing_docdb_fam**: The number of DOCDB patent families that cite this application.
26. **nb_applicants**: The number of applicants for the patent.
27. **nb_inventors**: The number of inventors listed on the application.

Example query: the most influencial European patents of the decade

Before moving to more complex queries, let's take a look at an example of a query using only the applications table.

We will use ORM for this example and throughout the rest of the course. Remember that for working with ORM, we need to import the table(s) we want to work with as models.

In [29]:

```
# Importing tables as models
from epo.tipdata.patstat.database.models import TLS201_APPLN
```

Our query

We will see what granted European patents have been cited the most, from the applications filed in this decade.

In [4]:

```
# Importing necessary modules
from epo.tipdata.patstat.database.models import TLS201_APPLN

# Define the query in ORM
q = db.query(
    TLS201_APPLN.appln_id,
    TLS201_APPLN.appln_nr,
    TLS201_APPLN.appln_filing_date,
    TLS201_APPLN.nb_citing_docdb_fam  # number of families citing
the application
).filter(
    TLS201_APPLN.appln_filing_year >= 2020,
    TLS201_APPLN.appln_auth == 'EP',
    TLS201_APPLN.granted == 'Y'
).order_by(
    TLS201_APPLN.nb_citing_docdb_fam.desc()
)

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

res
```

Out[4]:

	appln_id	appln_nr	appln_filing_date	nb_citing_docdb_fam
0	533200270	20182100	2020-06-24	302
1	533200253	20181956	2020-06-24	302
2	533254899	20182485	2020-06-26	207
3	543216412	20215721	2020-12-18	179
4	534806764	20186667	2020-07-20	155
...	...	...	...	...
26119	544684043	21700250	2021-01-05	0
26120	547136081	21162486	2021-03-15	0
26121	566183869	22155069	2022-02-03	0
26122	566731184	22155771	2022-02-09	0
26123	569137597	22163438	2022-03-22	0

26124 rows × 4 columns

Adding a link to the register

We will now add a link to the European patent register for the top 10 most cited granted patents. We use the application number of each record, and generate the url for the register with that application number.

In [5]:

```python
# Extract the first 10 records
top_10_records = res.head(10)

# Loop over the first 10 records and generate the URLs
urls = []
for index, row in top_10_records.iterrows():
    appln_nr = row['appln_nr']
    print (f"https://register.epo.org/application?number=EP{appln_nr}")
```

```
https://register.epo.org/application?number=EP20182100
https://register.epo.org/application?number=EP20181956
https://register.epo.org/application?number=EP20182485
https://register.epo.org/application?number=EP20215721
https://register.epo.org/application?number=EP20186667
https://register.epo.org/application?number=EP20708227
https://register.epo.org/application?number=EP21749328
https://register.epo.org/application?number=EP21802817
https://register.epo.org/application?number=EP21707816
https://register.epo.org/application?number=EP20163907
```

In [ ]: