

# PATSTAT Python Library: A Primer

## Setup

### Import the library module

```
In [1]: from epo.tipdata.patstat import PatstatClient
```

### Intantiate the client object

#### Note

As reported by the client instantiation output below, by default you will be working with a **test database**.

It's recommended to use it during the whole development phase, as it is way faster to test. You can switch anytime to the complete database (called 'PROD') without needing to apply any change to your code.

```
In [2]: patstat = PatstatClient()
```

```
This client instance is currently configured to use a test dataset with reduced number of publications (~10K).  
Use PatstatClient(env='PROD') to use the complete PATSTAT dataset (>140M publications).  
Use PatstatClient(env='TEST') to use the test dataset and avoid displaying this warning
```

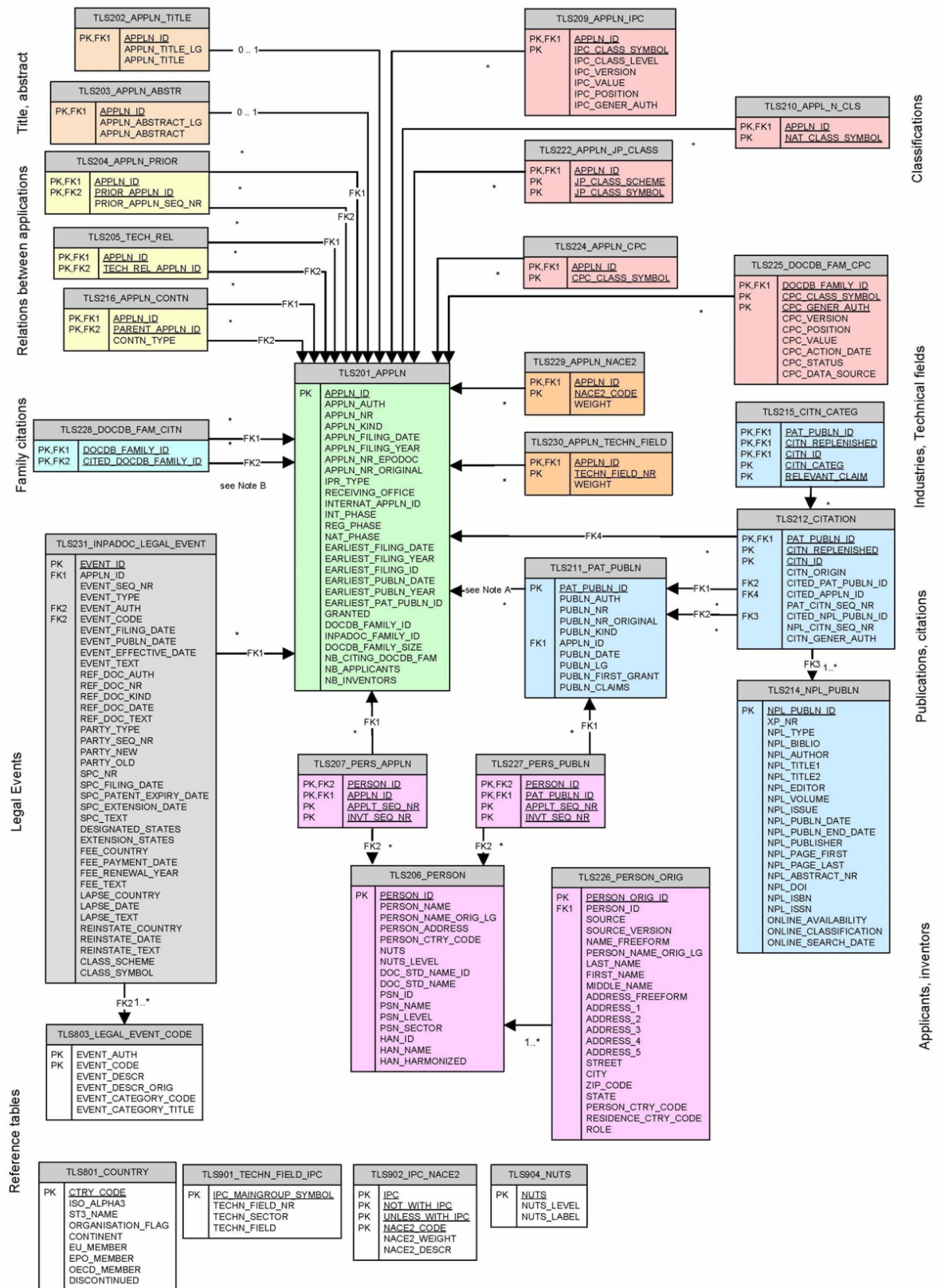
## Inspecting the PATSTAT database schema

PATSTAT is structured into two major sets of tables: *GLOBAL* and *REGISTER*

With the following commands you can get displayed the database schema, tables and relations for both of them. This can help quickly finding the needed information, and defining the relevant queries

## Getting the schema of PATSTAT GLOBAL

In [3]: *# Get the picture of the schema, as provided in the official documentation*  
 patstat.global\_schema()

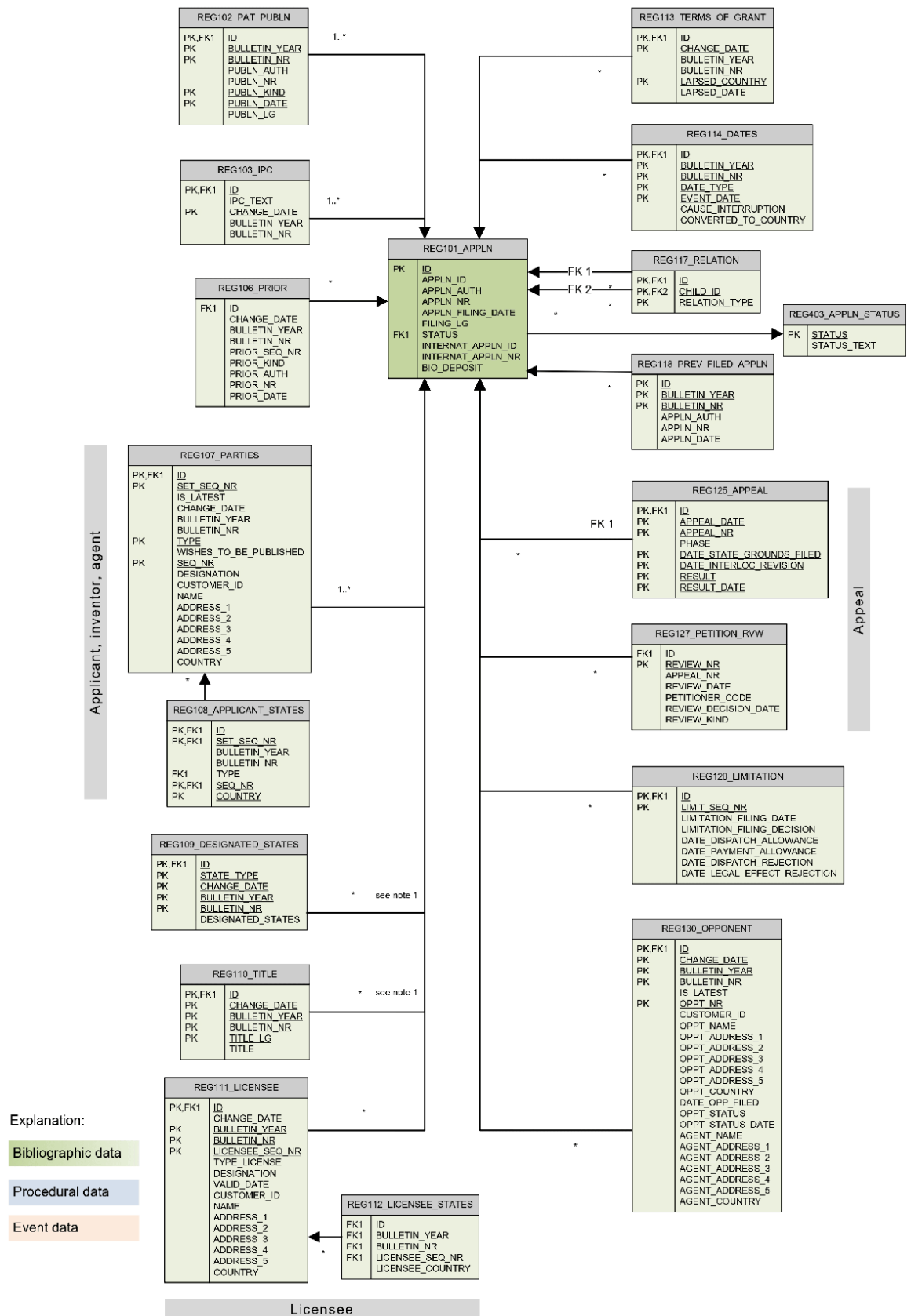


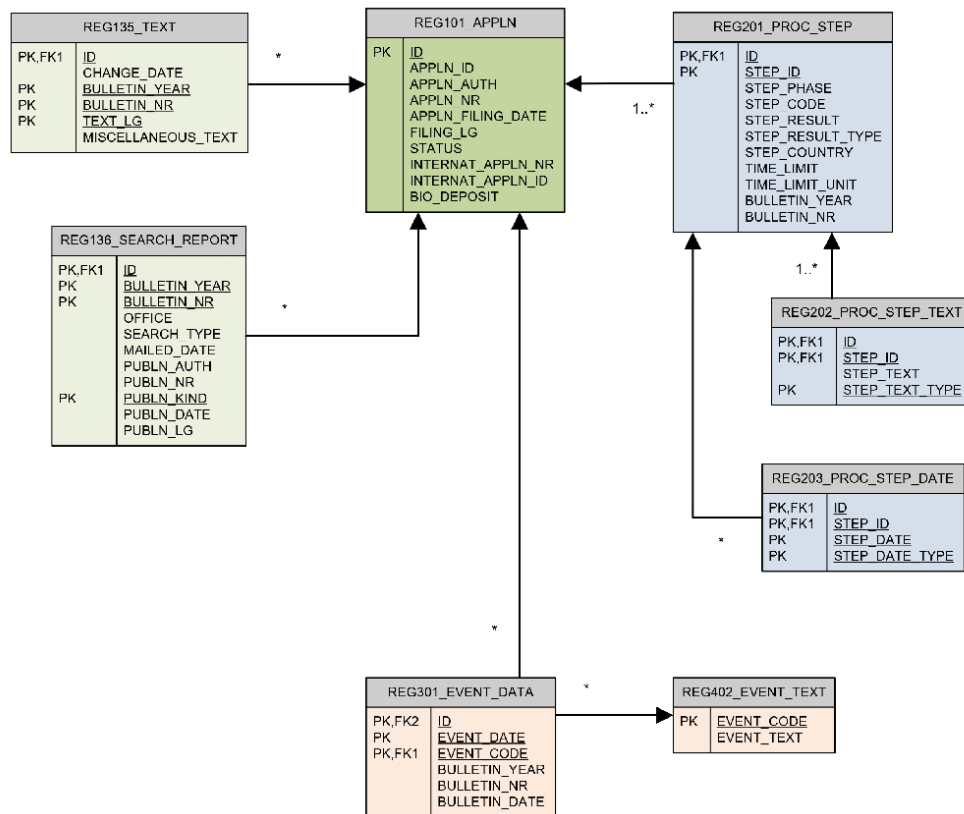
```
In [4]: # Get the list of tables also as a Python list
patstat.list_global_tables()
```

```
Out[4]: ['tls201_appln',
'tls204_appln_prior',
'tls205_tech_rel',
'tls216_appln_contn',
'tls209_appln_ipc',
'tls210_appln_n_cls',
'tls222_appln_jp_class',
'tls224_appln_cpc',
'tls225_docdb_fam_cpc',
'tls228_docdb_fam_citn',
'tls231_inpadoc_legal_event',
'tls211_pat_publn',
'tls212_citation',
'tls214_npl_publn',
'tls215_citn_categ',
'tls206_person',
'tls207_pers_appln',
'tls226_person_orig',
'tls227_pers_publn',
'tls801_country',
'tls803_legal_event_code',
'tls901_techn_field_ipc',
'tls902_ipc_nace2',
'tls903_region',
'tls229_appln_nace2',
'tls230_appln_techn_field',
'tls202_appln_title',
'tls203_appln_abstr']
```

## Getting the schema of PATSTAT REGISTER

```
In [5]: # As a picture
patstat.register_schema()
```





Explanation:

Bibliographic data

Procedural data

Event data

#### Legend:

0..1 cardinality

\* cardinality 0 ... n

PK This attribute is (part of) the Primary Key

FKn This attribute is (part of) the Foreign Key FK n;

Underlined attributes constitute the Primary Key.

```
In [6]: # As Python list
patstat.list_register_tables()
```

```
Out[6]: ['reg101_appln',
'reg102_pat_publn',
'reg110_title',
'reg118_prev_filed_appln',
'reg103_ipc',
'reg125_appeal',
'reg127_petition_rvw',
'reg128_limitation',
'reg135_text',
'reg136_search_report',
'reg117_relation',
'reg403_appln_status',
'reg107_parties',
'reg108_applicant_states',
'reg109_designated_states',
'reg111_licensee',
'reg112_licensee_states',
'reg130_opponent',
'reg106_prior',
'reg201_proc_step',
'reg202_proc_step_text',
'reg203_proc_step_date',
'reg301_event_data',
'reg402_event_text',
'reg113_terms_of_grant',
'reg114_dates',
'reg701_appln',
'reg705_places_of_business',
'reg707_parties',
'reg731_event_data',
'reg741_appln_status',
'reg742_event_text',
'reg711_licensee',
'reg712_licensee_states',
'reg714_dates',
'reg721_proc_step',
'reg722_proc_step_text',
'reg723_proc_step_date']
```

## Using the ORM

### Instantiate the ORM

```
In [7]: db = patstat.orm()
```

Here you can import all the tables that you need to use.

It's also possible to alternatively use:

```
from epo.tipdata.patstat.database import model
```

And then use `tab` -based autocompletion to get a hint on the list of the table models

```
In [8]: from epo.tipdata.patstat.database.models import TLS201_APPLN, TLS  
202_APPLN_TITLE, TLS206_PERSON, TLS207_PERS_APPLN, TLS224_APPLN_C  
PC, TLS231_INPADOC_LEGAL_EVENT
```

## 5 easy to understand SQL queries to get you started with PATSTAT

### Example 1: Granted EPO applications filed in 2010

List all granted applications filed at EPO (direct +PCT) having a year of filing = 2010.

```
In [9]: q = db.query(TLS201_APPLN.appln_id, TLS201_APPLN.appln_auth, TLS201_APPLN.appln_nr,
                    TLS201_APPLN.appln_kind, TLS201_APPLN.appln_filing_date).\
        filter(TLS201_APPLN.appln_filing_year == 2010,
               TLS201_APPLN.appln_auth == 'EP',
               TLS201_APPLN.granted == 'Y')
patstat.df(q)
```

Out[9]:

	appln_id	appln_auth	appln_nr	appln_kind	appln_filing_date
0	274222610	EP	10000313	A	2010-01-14
1	274369023	EP	10000849	A	2010-01-28
2	274681480	EP	10001469	A	2010-02-12
3	274720647	EP	10001552	A	2010-02-16
4	274875659	EP	10002051	A	2010-03-01
...	...	...	...	...	...
515	364674236	EP	10860973	A	2010-12-22
516	405968954	EP	13165754	A	2010-09-20
517	448177758	EP	16150333	A	2010-08-12
518	470108142	EP	16192322	A	2010-07-16
519	543439324	EP	20217409	A	2010-07-16

520 rows × 5 columns

### Example 2: EPO application in a range and a specific field.

List all applications filed at EPO (direct +PCT) that were filed between 2010 and 2015 (included) which have a CPC classification symbol in the field of wind energy. Applications that have multiple CPC classifications in wind energy will occur multiple times in the list.

Observe the blanks and % wildcard in the CPC classification symbol !

```
In [10]: q = db.query(TLS201_APPLN.appln_id, TLS201_APPLN.appln_auth, TLS2
01_APPLN.appln_nr,
                        TLS201_APPLN.appln_kind, TLS201_APPLN.
appln_filing_date, TLS224_APPLN_CPC.cpc_class_symbol).\
                        join(TLS224_APPLN_CPC).\
                        filter(TLS201_APPLN.appln_filing_year.between(2010,
2015),
                        TLS201_APPLN.appln_auth == 'EP',
                        TLS224_APPLN_CPC.cpc_class_symbol.like('Y02E
10/7%')).\
                        order_by(TLS201_APPLN.appln_id)
patstat.df(q)
```

Out [10]:

	appln_id	appln_auth	appln_nr	appln_kind	appln_filing_date	cpc_class_symbol
0	274088046	EP	10150465	A	2010-01-11	Y02E 10/72
1	274154890	EP	10150606	A	2010-01-13	Y02E 10/72
2	274154890	EP	10150606	A	2010-01-13	Y02E 10/72
3	274209407	EP	10000294	A	2010-01-14	Y02E 10/72
4	274209433	EP	10000317	A	2010-01-14	Y02E 10/72
...	...	...	...	...	...	...
5101	583551712	EP	22209927	A	2013-07-02	Y02E 10/72
5102	583551712	EP	22209927	A	2013-07-02	Y02E 10/76
5103	583918479	EP	22211827	A	2015-10-06	Y02E 10/72
5104	586040661	EP	23151711	A	2011-11-17	Y02E 10/72
5105	588255122	EP	23160005	A	2013-10-23	Y02E 10/72

5106 rows × 6 columns

### Example 3: EPO applications filed by a certain applicant

List all applications filed by 'NOKIA CORPORATION'.

We used the PATSTAT Standardised Name (psn\_name) which will give more (better) results then if we would have used the “person\_name” (name as published on the patent document) instead.

```
In [11]: q = db.query(TLS201_APPLN.appln_id, TLS201_APPLN.appln_auth, TLS2
01_APPLN.appln_nr,
                TLS201_APPLN.appln_kind, TLS201_APPLN.
appln_filing_date, TLS206_PERSON.psn_name,
                TLS202_APPLN_TITLE.appln_title).\
        select_from(TLS201_APPLN).\
        join(TLS207_PERS_APPLN).\
        join(TLS206_PERSON).\
        join(TLS202_APPLN_TITLE).\
        filter(TLS206_PERSON.psn_name == 'NOKIA CORPORATIO
N').\
        order_by(TLS201_APPLN.appln_filing_date.desc())
patstat.df(q)
```

Out [11]:

	appln_id	appln_auth	appln_nr	appln_kind	appln_filing_date	psn_name
0	421950346	US	201313795544	A	2013-03-12	NOKIA CORPORATION
1	411086047	US	201213443360	A	2012-04-10	NOKIA CORPORATION
2	341254279	US	84746310	A	2010-07-30	NOKIA CORPORATION
3	328781186	US	51518510	A	2010-06-24	NOKIA CORPORATION
4	266715124	US	7009708	A	2008-02-14	NOKIA CORPORATION
5	57363655	US	98551207	A	2007-11-15	NOKIA CORPORATION
6	52374000	US	63375606	A	2006-12-05	NOKIA CORPORATION
7	51265685	US	48682906	A	2006-07-14	NOKIA CORPORATION
8	24044628	WO	2006000869	W	2006-04-13	NOKIA CORPORATION
9	24039536	WO	2005003584	W	2005-11-29	NOKIA CORPORATION

10	24039536	WO	2005003584	W	2005-11-29	NOKIA CORPORATION
11	49042184	US	25434505	A	2005-10-20	NOKIA CORPORATION
12	49009025	US	25033805	A	2005-10-14	NOKIA CORPORATION
13	18585339	WO	2005000444	W	2005-10-14	NOKIA CORPORATION
14	53703468	US	83204904	A	2004-04-26	NOKIA CORPORATION
15	53496384	US	79441304	A	2004-03-04	NOKIA CORPORATION
16	52423896	US	64056603	A	2003-08-14	NOKIA CORPORATION
17	24011273	WO	0202432	W	2002-06-26	NOKIA CORPORATION
18	47850599	US	12195902	A	2002-04-12	NOKIA CORPORATION
19	45741043	US	1080801	A	2001-11-13	NOKIA CORPORATION
20	53595567	US	81219101	A	2001-03-19	NOKIA CORPORATION
21	17342174	EP	98917166	A	1998-04-28	NOKIA CORPORATION

#### Example 4: Create hitlist for applicant

Create a hitlist of Chinese applicants filing patents at the EPO (direct or PCT).

```
In [12]: from sqlalchemy import func
q = db.query(TLS206_PERSON.psn_name, TLS206_PERSON.person_ctype_code,
              func.count(TLS201_APPLN.appln_id).label('APPLICATIONS_AT_EPO')).\
        select_from(TLS206_PERSON).\
        join(TLS207_PERS_APPLN).join(TLS201_APPLN).\
        filter(TLS206_PERSON.person_ctype_code == 'CN',
               TLS207_PERS_APPLN.appln_seq_nr > 0,
               TLS207_PERS_APPLN.invt_seq_nr == 0,
               TLS201_APPLN.appln_auth == 'EP').\
        group_by(TLS206_PERSON.psn_name, TLS206_PERSON.person_ctype_code).\
        order_by(func.count(TLS201_APPLN.appln_id).desc()).\
        limit(10)
patstat.df(q)
```

Out [12]:

	psn_name	person_ctype_code	APPLICATIONS_AT_EPO
0	BEIJING GOLDWIND SCIENCE & CREATION WINDPOWER ...	CN	100
1	XINJIANG GOLDWIND SCIENCE & TECHNOLOGY COMPANY	CN	50
2	Beijing Goldwind Science & Creation Windpower ...	CN	21
3	Huawei Digital Power Technologies Co., Ltd.	CN	16
4	SINOVEL WIND GROUP COMPANY	CN	14
5	JIANGSU GOLDWIND SCIENCE & TECHNOLOGY COMPANY	CN	14
6	SUNGROW POWER SUPPLY COMPANY	CN	13
7	HUAWEI TECHNOLOGIES COMPANY	CN	10
8	GREE ELECTRIC APPLIANCES	CN	10
9	Shanghai Electric Wind Power Group Co., Ltd.	CN	7

### Example 5: Filter applicant applications by target country

Create a list of granted EP applications filed by Chinese applicants for which a renewal fee was paid in Belgium.

The “ AND appln\_auth = 'EP' ” condition in the WHERE clause is in principle redundant because the “event\_code = 'PGFP'” (Post Grant Fee Paid) condition implies that the application is a granted European patent (EP).

```

In [13]: q = db.query(TLS206_PERSON.psn_name, TLS206_PERSON.person_ctype,
                    TLS201_APPLN.appln_auth,
                    TLS201_APPLN.appln_nr, TLS201_APPLN.appln_kind,
                    TLS201_APPLN.appln_filing_date,
                    TLS231_INPADOC_LEGAL_EVENT.fee_country,
                    TLS231_INPADOC_LEGAL_EVENT.fee_payment_date,
                    TLS231_INPADOC_LEGAL_EVENT.fee_renewal_year,
                    TLS202_APPLN_TITLE.appln_title,
                    TLS201_APPLN.appln_id).\
                    select_from(TLS201_APPLN).\
                    join(TLS207_PERS_APPLN).join(TLS206_PERSON).join(TLS231_INPADOC_LEGAL_EVENT).join(TLS202_APPLN_TITLE).\
                    filter(TLS206_PERSON.person_ctype == 'CN',
                    TLS207_PERS_APPLN.applt_seq_nr > 0,
                    TLS201_APPLN.appln_auth == 'EP',
                    TLS231_INPADOC_LEGAL_EVENT.event_code == 'PGF
P',
                    TLS231_INPADOC_LEGAL_EVENT.fee_country == 'B
E').\
                    distinct().\
                    order_by(TLS231_INPADOC_LEGAL_EVENT.fee_payment_date.desc()).limit(10)
patstat.df(q)

```

Out [13]:

	psn_name	person_etry_code	appln_auth	appln_nr	appln_kind	appln_filing_da
0	JIANGSU SHENMA ELECTRIC COMPANY	CN	EP	13849544	A	2013-10-
1	Yueli Electric (Jiangsu) Co., Ltd.	CN	EP	15860005	A	2015-10-
2	BAFANG ELECTRIC (SUZHOU) COMPANY	CN	EP	07740080	A	2007-03-
3	SHANGHAI GHREPOWER GREEN ENERGY COMPANY	CN	EP	11835510	A	2011-05-
4	SHANGHAI INSTITUTE OF PHARMACEUTICAL INDUSTRY	CN	EP	12760538	A	2012-03-
5	SHANGHAI INSTITUTE OF PHARMACEUTICAL INDUSTRY	CN	EP	12761142	A	2012-03-
6	SHANGHAI GHREPOWER GREEN ENERGY COMPANY	CN	EP	11852207	A	2011-09-
7	NORTHEASTERN UNIVERSITY	CN	EP	15171679	A	2015-06-

## RAW SQL

It is of course also possible to access PATSTAT directly using SQL, without using the ORM simplification. The following examples replicate the previous cases, but the solution is given as direct SQL query

## 5 easy to understand SQL queries to get you started with PATSTAT

## Example 1

List all granted applications filed at EPO (direct +PCT) having a year of filing = 2010.

**Note** This and the next examples use the **legacy SQL** dialect, which e.g. allows to use the `+` notation in the select statement (see in particular `appln_auth +appln_nr + appln_kind number` below).

The legacy SQL dialect is selected by *default* for enabling backwards compatibility with legacy PATSTAT scripts. It can be set using the parameter `use_legacy_sql` (by default set to `True`)

```
In [14]: res = patstat.sql_query("""
SELECT appln_id, appln_auth +appln_nr + appln_kind number, appln_
filing_date

FROM tls201_appln

WHERE appln_filing_year = 2010  --application filing year = 2010

AND appln_auth = 'EP'          --patents filed at the EPO (direc
t + PCT)

AND granted = 'Y'             --the patent has been granted
""")
res[0:5]
```

```
Out [14]: [{'appln_id': 274222610,
'number': 'EP10000313A ',
'appln_filing_date': datetime.date(2010, 1, 14)},
{'appln_id': 274369023,
'number': 'EP10000849A ',
'appln_filing_date': datetime.date(2010, 1, 28)},
{'appln_id': 274681480,
'number': 'EP10001469A ',
'appln_filing_date': datetime.date(2010, 2, 12)},
{'appln_id': 274720647,
'number': 'EP10001552A ',
'appln_filing_date': datetime.date(2010, 2, 16)},
{'appln_id': 274875659,
'number': 'EP10002051A ',
'appln_filing_date': datetime.date(2010, 3, 1)}]
```

## Example 2

List all applications filed at EPO (direct +PCT) that were filed between 2010 and 2015 (included) which have a CPC classification symbol in the field of wind energy. Applications that have multiple CPC classifications in wind energy will occur multiple times in the list.

Observe the blanks and % wildcard in the CPC classification symbol !

```
In [15]: res = patstat.sql_query("""
SELECT tls201_appln.appln_id, appln_auth +appln_nr + appln_kind n
umber, appln_filing_date, cpc_class_symbol

FROM tls201_appln JOIN tls224_appln_cpc

ON tls201_appln.appln_id = tls224_appln_cpc.appln_id

WHERE appln_filing_year between 2010 and 2015    --application fil
ing between 2010 and 2015

AND appln_auth ='EP'                            --patents
filed at the EP0 (direct + PCT)

AND cpc_class_symbol like 'Y02E 10/7%'          --CPC classificat
ion for wind energy

ORDER BY tls201_appln.appln_id                  --orders the resu
lts per application
""")
res[0:5]
```

```
Out[15]: [{'tls201_appln_appln_id': 274088046,
'number': 'EP10150465A ',
'appln_filing_date': datetime.date(2010, 1, 11),
'cpc_class_symbol': 'Y02E 10/72'},
{'tls201_appln_appln_id': 274154890,
'number': 'EP10150606A ',
'appln_filing_date': datetime.date(2010, 1, 13),
'cpc_class_symbol': 'Y02E 10/72'},
{'tls201_appln_appln_id': 274154890,
'number': 'EP10150606A ',
'appln_filing_date': datetime.date(2010, 1, 13),
'cpc_class_symbol': 'Y02E 10/727'},
{'tls201_appln_appln_id': 274209407,
'number': 'EP10000294A ',
'appln_filing_date': datetime.date(2010, 1, 14),
'cpc_class_symbol': 'Y02E 10/72'},
{'tls201_appln_appln_id': 274209433,
'number': 'EP10000317A ',
'appln_filing_date': datetime.date(2010, 1, 14),
'cpc_class_symbol': 'Y02E 10/72'}]
```

### Example 3

List all applications filed by 'NOKIA CORPORATION'. We use the PATSTAT Standardised Name (psn\_name) which will give more (better) results than if we would have used the “person\_name” (name as published on the patent document) instead.

**Note** also that in this and the following examples we will use the **standard SQL** dialect (parameter `use_legacy_sql` set to `False`).

**Note 2** : As you can see, it is **NOT** necessary to enter the name of the project and the dataset in the query string (which is normally requested by standard BigQuery SQL). The `sql_query` method does this for you!

```
In [16]: res = patstat.sql_query("""
SELECT tls201_appln.appln_id, appln_auth, appln_nr, appln_kind, a
      ppln_filing_date, psn_name, appln_nr_epodoc, appln_title

FROM tls201_appln JOIN tls207_pers_appln ON tls201_appln.appln_id
      = tls207_pers_appln.appln_id

JOIN tls206_person ON tls207_pers_appln.person_id = tls206_perso
      n.person_id

JOIN tls202_appln_title ON tls202_appln_title.appln_id = tls201_a
      ppln.appln_id

WHERE psn_name like 'NOKIA CORPORATION' -- all application filed
      by = 'NOKIA CORPORATION'

ORDER BY appln_filing_date desc           -- orders the results on
      the application filing date
""", use_legacy_sql=False)
res[0:5]
```

```

Out[16]: [{ 'appln_id': 421950346,
            'appln_auth': 'US',
            'appln_nr': '201313795544',
            'appln_kind': 'A ',
            'appln_filing_date': datetime.date(2013, 3, 12),
            'psn_name': 'NOKIA CORPORATION',
            'appln_nr_epodoc': None,
            'appln_title': 'Steerable transmit, steerable receive frequency
modulated continuous wave radar transceiver'},
          { 'appln_id': 411086047,
            'appln_auth': 'US',
            'appln_nr': '201213443360',
            'appln_kind': 'A ',
            'appln_filing_date': datetime.date(2012, 4, 10),
            'psn_name': 'NOKIA CORPORATION',
            'appln_nr_epodoc': None,
            'appln_title': 'METHOD AND APPARATUS FOR PROVIDING SERVICES USI
NG CONNECTING USER INTERFACE ELEMENTS'},
          { 'appln_id': 341254279,
            'appln_auth': 'US',
            'appln_nr': '84746310',
            'appln_kind': 'A ',
            'appln_filing_date': datetime.date(2010, 7, 30),
            'psn_name': 'NOKIA CORPORATION',
            'appln_nr_epodoc': None,
            'appln_title': 'EXECUTION AND DISPLAY OF APPLICATIONS'},
          { 'appln_id': 328781186,
            'appln_auth': 'US',
            'appln_nr': '51518510',
            'appln_kind': 'A ',
            'appln_filing_date': datetime.date(2010, 6, 24),
            'psn_name': 'NOKIA CORPORATION',
            'appln_nr_epodoc': None,
            'appln_title': 'Method and Apparatus For Staged Approach Transi
ent RF Detection And Sensor Power Saving'},
          { 'appln_id': 266715124,
            'appln_auth': 'US',
            'appln_nr': '7009708',
            'appln_kind': 'A ',
            'appln_filing_date': datetime.date(2008, 2, 14),
            'psn_name': 'NOKIA CORPORATION',
            'appln_nr_epodoc': None,
            'appln_title': 'Apparatus, methods, and computer program produc
ts providing improved communication in decentralized wireless net
works'}]

```

#### Example 4

Creates a hitlist of Chinese applicants filing patents at the EPO (direct or PCT).

```
In [17]: res = patstat.sql_query("""
SELECT psn_name, person_ctry_code,

COUNT(tls201_appln.appln_id) APPLICATIONS_AT_EPO -- Counting appl
ications

FROM tls201_appln JOIN tls207_pers_appln ON tls201_appln.appln_id
= tls207_pers_appln.appln_id

JOIN tls206_person ON tls207_pers_appln.person_id = tls206_perso
n.person_id

WHERE person_ctry_code = 'CN'                --Chinese "person
s"

AND applt_seq_nr > 0 AND invt_seq_nr = 0      --only consider a
pplicants, exclude inventors

        AND appln_auth = 'EP'                --patents
filed at the EPO

GROUP BY psn_name, person_ctry_code          -- creates totals
per name/country combination

ORDER BY count(tls201_appln.appln_id) desc   -- order
in descending number of applications
""", use_legacy_sql=False)
res[0:5]
```

```
Out[17]: [{'psn_name': 'BEIJING GOLDWIND SCIENCE & CREATION WINDPOWER EQUI
PMENT COMPANY',
  'person_ctry_code': 'CN',
  'APPLICATIONS_AT_EPO': 100},
 {'psn_name': 'XINJIANG GOLDWIND SCIENCE & TECHNOLOGY COMPANY',
  'person_ctry_code': 'CN',
  'APPLICATIONS_AT_EPO': 50},
 {'psn_name': 'Beijing Goldwind Science & Creation Windpower Equi
pment Co. Ltd.',
  'person_ctry_code': 'CN',
  'APPLICATIONS_AT_EPO': 21},
 {'psn_name': 'Huawei Digital Power Technologies Co., Ltd.',
  'person_ctry_code': 'CN',
  'APPLICATIONS_AT_EPO': 16},
 {'psn_name': 'SINOVEL WIND GROUP COMPANY',
  'person_ctry_code': 'CN',
  'APPLICATIONS_AT_EPO': 14}]
```

## Example 5

Create a list of granted EP applications filed by Chinese applicants for which a renewal fee was paid in Belgium. The “ AND appln\_auth = 'EP' ” condition in the WHERE clause is in principle redundant because the “event\_code = 'PGFP'” (Post Grant Fee Paid) condition implies that the application is a granted European patent. (EP).

```
In [18]: res = patstat.sql_query("""
SELECT DISTINCT psn_name, person_ctry_code, appln_auth, appln_nr,
appln_kind number, appln_filing_date, fee_country, fee_payment_date,
fee_renewal_year, appln_title, tls201_appln.appln_id

FROM tls201_appln JOIN tls207_pers_appln ON tls201_appln.appln_id
= tls207_pers_appln.appln_id

JOIN tls206_person ON tls207_pers_appln.person_id = tls206_person.person_id

JOIN tls231_inpadoc_legal_event ON tls201_appln.appln_id = tls231_inpadoc_legal_event.appln_id

JOIN tls202_appln_title ON tls201_appln.appln_id = tls202_appln_title.appln_id

WHERE person_ctry_code = 'CN'           --Chinese “persons”

AND applt_seq_nr > 0                    --only consider applicants

      AND appln_auth = 'EP'              --patents filed at the EP0

AND event_code = 'PGFP'                 -- a Postgrant Fee was paid (implies granted EP)

AND fee_country = 'BE'                  -- fee paid in Belgium

ORDER BY fee_payment_date desc          -- order in descending fee payment date
""", use_legacy_sql=False)
res[0:5]
```

```
Out[18]: [{'psn_name': 'Yueli Electric (Jiangsu) Co., Ltd.',
'person_ctry_code': 'CN',
'appln_auth': 'EP',
'appln_nr': '15860005',
'number': 'A ',
'appln_filing_date': datetime.date(2015, 10, 30),
'fee_country': 'BE',
'fee_payment_date': datetime.date(2023, 9, 18),
'fee_renewal_year': 9,
```

```

    'appln_title': 'WHEEL-HUB DRIVEN WIND TURBINES',
    'appln_id': 456192321},
    {'psn_name': 'JIANGSU SHENMA ELECTRIC COMPANY',
     'person_ctype_code': 'CN',
     'appln_auth': 'EP',
     'appln_nr': '13849544',
     'number': 'A ',
     'appln_filing_date': datetime.date(2013, 10, 24),
     'fee_country': 'BE',
     'fee_payment_date': datetime.date(2023, 9, 18),
     'fee_renewal_year': 11,
     'appln_title': 'A COMPOSITE TOWER COMPRISING ACROSS ARM STRUCTU
RE FOR POWER TRANSMISSION LINES',
     'appln_id': 417363519},
    {'psn_name': 'BAFANG ELECTRIC (SUZHOU) COMPANY',
     'person_ctype_code': 'CN',
     'appln_auth': 'EP',
     'appln_nr': '07740080',
     'number': 'A ',
     'appln_filing_date': datetime.date(2007, 3, 28),
     'fee_country': 'BE',
     'fee_payment_date': datetime.date(2023, 3, 21),
     'fee_renewal_year': 17,
     'appln_title': 'ELECTRICALLY ASSISTED BICYCLE AND UNIT ADAPTED
FOR USE IN ELECTRICALLY ASSISTED BICYCLE AND CAPABLE OF BEING ATT
ACHED TO BICYCLE BODY FRAME',
     'appln_id': 408291},
    {'psn_name': 'SHANGHAI GHREPOWER GREEN ENERGY COMPANY',
     'person_ctype_code': 'CN',
     'appln_auth': 'EP',
     'appln_nr': '11835510',
     'number': 'A ',
     'appln_filing_date': datetime.date(2011, 5, 27),
     'fee_country': 'BE',
     'fee_payment_date': datetime.date(2020, 5, 7),
     'fee_renewal_year': 10,
     'appln_title': 'POWER SUPPLY SYSTEM WITH INTEGRATION OF WIND PO
WER, SOLAR ENERGY, DIESEL fuel generator AND MAINS SUPPLY',
     'appln_id': 353444681},
    {'psn_name': 'SHANGHAI INSTITUTE OF PHARMACEUTICAL INDUSTRY',
     'person_ctype_code': 'CN',
     'appln_auth': 'EP',
     'appln_nr': '12760538',
     'number': 'A ',
     'appln_filing_date': datetime.date(2012, 3, 22),
     'fee_country': 'BE',
     'fee_payment_date': datetime.date(2020, 2, 25),
     'fee_renewal_year': 9,
     'appln_title': 'NEW CRYSTAL FORM VII OF AGOMELATINE, PREPARATIO
N METHOD AND USE THEREOF AND PHARMACEUTICAL COMPOSITION CONTAININ
G SAME',
     'appln_id': 375672617}]

```