The Patent Application Abstract Table (TLS203_APPLN_ABSTR)

Welcome to the TLS203_APPLN_ABSTR table in PATSTAT.This table contains the English language abstract, if available. If there is no abstract in English, then it contains the most recent abstract in another language.

As always, we start creating the PATSTAT client and accessing ORM. Then we import the TLS203_APPLN_ABSTR table.

In [1]:

```
from epo.tipdata.patstat import PatstatClient

# Initialize the PATSTAT client
patstat = PatstatClient(env='PROD')

# Access ORM
db = patstat.orm()

# Importing the as models
from epo.tipdata.patstat.database.models import TLS203_APPLN_ABSTR
```

APPLN_ID

Again we find the unique identifier for each patent application as primary key of the table. Let's join this table with table TLS201.

In [2]:

```
# Import table TLS201
from epo.tipdata.patstat.database.models import TLS201_APPLN

appln_id = db.query(
    TLS203_APPLN_ABSTR.appln_id,
    TLS201_APPLN.appln_nr
).join(
    TLS201_APPLN, TLS203_APPLN_ABSTR.appln_id == TLS201_APPLN.appln_id  # Join the two table via the common appln_id attribute
).limit(20000)

appln_id_df = patstat.df(appln_id)
appln_id_df
```

Out[2]:

	appln_id	appln_nr
0	440666936	102013019442
1	572506	56098
2	55837247	102007040089
3	341255512	202011100038
4	499421084	102017105861
...	...	...
19995	23074912	9212696
19996	530841407	201921894673
19997	333355458	200980114649
19998	24053634	2007001718
19999	530842110	201922037884

20000 rows × 2 columns

APPLN_ABSTRACT_LG

Language of the abstract of the application selected for and loaded in PATSTAT.

Let's check how many applications have an abstract that is no in English. We import `func` from `sqlalchemy` to perform the `count` on the application IDs. We filter excluding the applications that have the `appln_abstract_lg` different from 'en'.

In [3]:

```
from sqlalchemy import func

lg = db.query(
    TLS203_APPLN_ABSTR.appln_abstract_lg,
    func.count(TLS203_APPLN_ABSTR.appln_id).label('total_applications')
).filter(
    TLS203_APPLN_ABSTR.appln_abstract_lg != 'en'
).group_by(
    TLS203_APPLN_ABSTR.appln_abstract_lg
).order_by(
    func.count(TLS203_APPLN_ABSTR.appln_id).desc()
)

lg_df = patstat.df(lg)
lg_df
```

Out[3]:

	appln_abstract_lg	total_applications
0	ko	1632173
1	de	1221298
2	zh	1018900
3	ja	923929
4	es	899207
5	fr	674754
6	pt	542691
7	ru	257792
8	tr	97058
9	uk	71156
10	no	62639
11	pl	51740
12	it	39172
13	hu	28315
14	el	25404
15	ro	17189
16	nl	16437
17	cs	11003
18	sh	9511
19	hr	7306
20	ar	6685

21	sr	3617
22	da	3086
23	fi	3068
24	sv	2801
25	sl	2720
26	bg	1045
27	sk	926
28	bs	472
29	me	469
30	lv	326
31	lt	78
32	et	1

We can check which application authorities have more distinct languages used for the abstracts of the applications filed therein.

In [6]:

```
lg_auth = db.query(
    TLS201_APPLN.appln_auth,
    func.count(TLS203_APPLN_ABSTR.appln_abstract_lg.distinct()).label('num_of_languages')  # Count the distinct number of title languages
).join(
    TLS201_APPLN, TLS203_APPLN_ABSTR.appln_id == TLS201_APPLN.appln_id
).group_by(
    TLS201_APPLN.appln_auth
).order_by(
    func.count(TLS203_APPLN_ABSTR.appln_abstract_lg.distinct()).desc()
)

lg_auth_df = patstat.df(lg_auth)
lg_auth_df
```

Out[6]:

	appln_auth	num_of_languages
0	CH	4
1	KR	4
2	ME	4
3	BE	4
4	FI	3
...	...	...
77	AU	1
78	PE	1
79	GB	1
80	MT	1
81	MC	1

82 rows × 2 columns

Out of curiosity, let's check which title languages are present among the applications filed at the EPO and the WIPO. As in table TLS202, we need to import `distinct` .

In [7]:

```
# Import distinct from sqlalchemy
from sqlalchemy import distinct

lg_wo_ep_ch = db.query(
    TLS201_APPLN.appln_auth,
    TLS203_APPLN_ABSTR.appln_abstract_lg.label('distinct_languages')
).distinct(  # Consider distinct appln_auth-appln_title_lg rows combinations only
).join(
    TLS201_APPLN, TLS203_APPLN_ABSTR.appln_id == TLS201_APPLN.appln_id
).filter(
    (TLS201_APPLN.appln_auth == 'WO') | (TLS201_APPLN.appln_auth == 'EP')
).order_by(
    TLS201_APPLN.appln_auth
)

lg_wo_ep_ch_df = patstat.df(lg_wo_ep_ch)
lg_wo_ep_ch_df
```

Out[7]:

	appln_auth	distinct_languages
0	EP	de
1	EP	fr
2	EP	en
3	WO	en
4	WO	de
5	WO	fr

APPLN_ABSTRACT

Abstract of the application. Multiple abstracts may be published for any application, but only one abstract will be stored in PATSTAT, according to these rules (first applicable rule is applied):

1. most recent (according to publication date) abstract in English
2. most recent abstract in language of publication
3. most recent abstract in any other language.

We can show the abstracts of the applications with `appln_abstract_lg` equal to 'en'.

In [ ]:

```
abstract = db.query(
    TLS203_APPLN_ABSTR.appln_id,
    TLS203_APPLN_ABSTR.appln_abstract_lg,
    TLS203_APPLN_ABSTR.appln_abstract
).filter(
    TLS203_APPLN_ABSTR.appln_abstract_lg == 'en'
)

abstract_df = patstat.df(abstract)
abstract_df
```

In [ ]: