

The Patent Application Title Table (TLS202_APPLN_TITLE)

Welcome to the second table in PATSTAT, namely the TLS202_APPLN_TITLE table. It comprises three attributes: one containing the application ID, and the other two provide an abstract and the language of the title of the application.

Let's start creating the PATSTAT client and accessing ORM. Then we import the TLS202_APPLN_TITLE 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 TLS202_APPLN_TITLE
```

APPLN_ID (Primary Key)

As seen in table TLS201, this is the unique identifier for each patent application in the PATSTAT database. We can use it to join this table with table TLS201.

```
In [2]: # Import table TLS201
from epo.tipdata.patstat.database.models import TLS201_APPLN

appln_id = db.query(
    TLS202_APPLN_TITLE.appln_id,
    TLS201_APPLN.appln_nr
).join(
    TLS201_APPLN, TLS202_APPLN_TITLE.appln_id == TLS201_APPLN.appln_id
).limit(20000)

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

Out [2]:

	appln_id	appln_nr
0	574743335	2020000167
1	15488449	04253483
2	15446321	639689
3	15369107	387261
4	15415692	520586
...
19995	479598911	201611234274
19996	597573164	202320457599
19997	34140771	19699096
19998	410013675	2012284942
19999	40843897	20050080144

20000 rows × 2 columns

APPLN_TITLE_LG

Language of the title of the application.

Let's see which is the most used language to file applications. Remember to import `func` from the `sqlalchemy` library in order to use aggregate functions.

```
In [3]: from sqlalchemy import func

lg = db.query(
    TLS202_APPLN_TITLE.appln_title_lg,
    func.count(TLS202_APPLN_TITLE.appln_id).label('total_applications')
).group_by(
    TLS202_APPLN_TITLE.appln_title_lg
).order_by(
    func.count(TLS202_APPLN_TITLE.appln_id).desc()
)

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

Out[3]:

	appln_title_lg	total_applications
0	en	94633136
1	de	6194803
2	fr	2216641
3	es	1183417
4	ja	1021108
5	pt	951028
6	zh	907168
7	it	713085
8	ko	697854
9	ru	522562
10	da	447370
11	fi	205902
12	sv	203598
13	nl	198866
14	no	188435
15	tr	111308
16	uk	68846
17	el	41955
18	pl	19321
19	id	14635
20	is	7440
21	et	6680

22	ar	5173
23	sh	4475
24	lv	1247
25	lt	966
26	cs	894
27	bg	743
28	ro	233
29	bs	218
30	sr	96
31	hr	59
32	hu	30
33	sk	9
34	he	5
35	sl	2
36	vi	1
37	me	1

As we could expect, the most common language is English.

Suppose that we are also interested in knowing which application authorities have more distinct languages used for the titles of the applications filed therein.

```
In [4]: lg_auth = db.query(
        TLS201_APPLN.appln_auth,
        func.count(TLS202_APPLN_TITLE.appln_title_lg.distinct()).label(
            'num_of_languages') # Count the distinct number of title languages
    ).join(
        TLS201_APPLN, TLS202_APPLN_TITLE.appln_id == TLS201_APPLN.appln_id
    ).group_by(
        TLS201_APPLN.appln_auth
    ).order_by(
        func.count(TLS202_APPLN_TITLE.appln_title_lg.distinct()).desc()
    )

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

Out [4]:

	appln_auth	num_of_languages
0	PT	10
1	SE	6
2	WO	6
3	GR	6
4	BE	6
...
101	IE	1
102	ZW	1
103	UY	1
104	CO	1
105	MC	1

106 rows × 2 columns

Out of curiosity, we can check which title languages are present among the applications filed at the EPO and the WIPO. We need to consider distinct combinations of `appln_auth` - `appln_title_lg` rows. However, we are not applying this to an aggregate function. In this case, we have to use the `distinct` operator.

```
In [5]: # Import distinct from sqlalchemy
        from sqlalchemy import distinct

        lg_wo_ep = db.query(
            TLS201_APPLN.appln_auth,
            TLS202_APPLN_TITLE.appln_title_lg.label('distinct_languages')
        ).distinct( # Consider distinct appln_auth-appln_title_lg rows combinations only
        ).join(
            TLS201_APPLN, TLS202_APPLN_TITLE.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_df = patstat.df(lg_wo_ep)
        lg_wo_ep_df
```

Out[5]:

	appln_auth	distinct_languages
0	EP	en
1	EP	de
2	WO	zh
3	WO	es
4	WO	de
5	WO	en
6	WO	el
7	WO	fr

APPLN_TITLE

Title of the application. Only one of possibly multiple abstracts is stored. See table TLS203_APPLN_ABSTRACT for details.

We can take a look to the abstracts stored in the table.

```
In [6]: title = db.query(
        TLS202_APPLN_TITLE.appln_id,
        TLS202_APPLN_TITLE.appln_title_lg,
        TLS202_APPLN_TITLE.appln_title
    ).limit(20000)

title_df = patstat.df(title)
title_df
```

Out[6]:

	appln_id	appln_title_lg	appln_title
0	476492379	ar	بواسطة المعـا ACETIC ACID طريقة لانتاج حمض الخل...
1	404900247	ar	طريقة وجهاز استعمال جسيمات البلازما في سائل وا...
2	334503647	ar	جهاز دراسة البيانات متكاملة في نظام المعلومات ...
3	597126818	ar	جهاز ضخ يستخدم في بئر عميق
4	476503309	ar	واستخدامه في عملي CATALYST MODIFIER معدل لحفاز
...
19995	579094696	de	Radialhebelfederanordnung für einen Drehschwin...
19996	511681179	de	Spracherkennungsvorrichtung
19997	15063500	de	Oberfräse
19998	6164360	de	Verfahren zur Darstellung eines basischen Farb...
19999	6227894	de	Verfahren für beidseitige Beschichtung von Kar...

20000 rows × 3 columns

In []: