The TLS227_PERS_PUBLN Table

Welcome to a comprehensive exploration of one of the key tables in the PATSTAT database: the `TLS227_PERS_PUBLN` Table. `TLS227_PERS_PUBLN` is an essential table in the PATSTAT database, designed to link individuals or entities, such as applicants and inventors, to patent publications. This table not only tracks who is involved in a patent but also how their contributions **evolve** as the patent moves through different stages of the publication process.

The `TLS227_PERS_PUBLN` table is fundamental in patent data analysis because it provides insight into the dynamics of patent authorship and applicant involvement. As patents progress from application to publication, this table allows analysts to study changes in the roles of inventors and applicants, such as:

- Identifying if a new inventor joins a patent team during its development.
- Examining if an applicant, often a company or institution, changes during the publication process.
- Detecting patterns in collaborative inventions where multiple inventors or applicants are linked to one patent publication.

Since patent applications can span several years before getting published, there may be alterations in the list of inventors or applicants due to company acquisitions, inventors moving to different organisations, or other administrative changes. `TLS227_PERS_PUBLN` captures these shifts by linking people to their respective publications, offering a detailed view of the timeline of contribution.

The table `TLS227_PERS_PUBLN` serves as a linking table that connects two primary entities:

1. **Applicants/Inventors** (via `PERSON_ID`): These could be individuals or organisations (corporations, universities, etc.) involved in the patent.
2. **Publications** (via `PAT_PUBLN_ID`): These represent the official patent publications issued by patent authorities.

By joining `TLS227_PERS_PUBLN` with other tables like `TLS206_PERSON` (for person details) and `TLS211_PAT_PUBLN` (for publication details), analysts can gain a clearer picture of who is contributing to each patent and how the contributions shift over time as new versions of the patent are published.

In [1]:

```
from epo.tipdata.patstat import PatstatClient
from epo.tipdata.patstat.database.models import TLS227_PERS_PUBLN, TLS206_PERSON, TLS211_PAT_PUBLN
from sqlalchemy import func, case, select, and_

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

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

The relationship in `TLS227_PERS_PUBLN` is more complex than a simple many-to-many structure. In fact, it is a **many-to-many-to-many** relationship, which arises due to the involvement of multiple entities. Namely:

- **Applications**: a patent application can go through multiple publications as it progresses through the patent examination and granting process.
- **Publications**: each patent application can result in multiple publications over time (e.g., as it gets published in different jurisdictions or stages like granted patents, amended versions, etc.).
- **Persons** (Applicants/Inventors): for each publication, there can be multiple persons involved (either as applicants or inventors), and each person may be involved in multiple patent publications.

A single **patent application** (represented by `APPLN_ID`) can generate multiple **publications** (represented by `AT_PUBLN_ID`). For example, an application might be published as an initial publication and later as a granted patent, or even as amended versions in different jurisdictions.

Each **publication** can involve multiple **persons** (applicants and/or inventors). For instance, a team of inventors or a group of companies may be listed as part of the publication.

Each **person** can be associated with multiple **applications** and their corresponding **publications**. For example, an inventor may file several patents, each of which can result in multiple publications.

Key Fields in the `TLS227_PERS_PUBLN` Table

PERSON_ID

This field serves as a foreign key referencing the `TLS206_PERSON` table, which stores detailed information about the person, such as their name, address, and country.

PAT_PUBLN_ID

This foreign key links to `TLS211_PAT_PUBLN` , a table that contains details about the patent publication, including publication authority, number, kind code, and date.

APPLT_SEQ_NR

This field indicates the sequence number of the applicant in a particular publication, showing the order in which applicants are listed.

INVT_SEQ_NR

This sequence number refers to the inventor's position in the publication record, helping to identify the primary inventor or additional inventors involved.

Some possible queries

This query retrieves all patent publications that a specific person (applicant or inventor) has been involved in, along with their role in each publication.

In [2]:

```
q = db.query(
    TLS227_PERS_PUBLN.person_id,
    TLS206_PERSON.person_name,
    TLS227_PERS_PUBLN.pat_publn_id,
    TLS211_PAT_PUBLN.publn_nr,    # Publication number
    TLS211_PAT_PUBLN.publn_auth,  # Publication authority
    TLS227_PERS_PUBLN.applt_seq_nr,
    TLS227_PERS_PUBLN.invt_seq_nr
).join(
    TLS206_PERSON, TLS227_PERS_PUBLN.person_id == TLS206_PERSON.p
erson_id
).join(
    TLS211_PAT_PUBLN, TLS227_PERS_PUBLN.pat_publn_id == TLS211_PA
T_PUBLN.pat_publn_id
).filter(
    TLS227_PERS_PUBLN.person_id == 5227571
).order_by(
    TLS211_PAT_PUBLN.publn_nr  # Sort by publication number
)

res1 = patstat.df(q)
res1
```

Out[2]:

	person_id	person_name	pat_publn_id	publn_nr	publn_auth	applt_seq_nr	invt_s
0	5227571	Apple Inc.	389962970	0811605	GB	1	
1	5227571	Apple Inc.	389964932	0813588	GB	1	
2	5227571	Apple Inc.	389964936	0813592	GB	1	
3	5227571	Apple Inc.	389966840	0815519	GB	1	
4	5227571	Apple Inc.	389966841	0815520	GB	1	
...	...	...	...	...	...	...	
6251	5227571	Apple Inc.	295297763	9358598	AU	1	
6252	5227571	Apple Inc.	488679292	9869030	US	1	
6253	5227571	Apple Inc.	419739654	D708632	US	2	
6254	5227571	Apple Inc.	391960377	TO20080448	IT	1	
6255	5227571	Apple Inc.	391960378	TO20080449	IT	1	

6256 rows × 7 columns

This query will find all the inventors and applicants across multiple publications of a specific patent application.

In [3]:

```
q = db.query(
    TLS227_PERS_PUBLN.person_id,
    TLS206_PERSON.person_name,
    TLS227_PERS_PUBLN.pat_publn_id,
    TLS211_PAT_PUBLN.publn_nr,
    TLS227_PERS_PUBLN.applt_seq_nr,
    TLS227_PERS_PUBLN.invt_seq_nr
).join(
    TLS206_PERSON, TLS227_PERS_PUBLN.person_id == TLS206_PERSON.person_id
).join(
    TLS211_PAT_PUBLN, TLS227_PERS_PUBLN.pat_publn_id == TLS211_PAT_PUBLN.pat_publn_id
).filter(
    TLS211_PAT_PUBLN.appln_id == 473378055
).order_by(
    TLS227_PERS_PUBLN.invt_seq_nr
)

res2 = patstat.df(q)
res2
```

Out[3]:

	person_id	person_name	pat_publn_id	publn_nr	applt_seq_nr	invt_seq_nr
0	5210554	General Electric Company	473378056	2016377051	1	0
1	5210554	General Electric Company	515503477	10337490	1	0
2	48462548	Yarbrough, Aaron A.	473378056	2016377051	0	1
3	48379676	Caruso, Christopher Daniel	515503477	10337490	0	1
4	48462548	Yarbrough, Aaron A.	515503477	10337490	0	2
5	12047856	Hynum, Daniel Alan	473378056	2016377051	0	2
6	12047856	Hynum, Daniel Alan	515503477	10337490	0	3
7	48379676	Caruso, Christopher Daniel	473378056	2016377051	0	3

This query identifies all applications where a specific person is listed across multiple publications.

In [4]:

```
q = db.query(
    TLS227_PERS_PUBLN.person_id,
    TLS206_PERSON.person_name,
    TLS211_PAT_PUBLN.appln_id,
    func.count(TLS211_PAT_PUBLN.pat_publn_id).label('publication_count')
).join(
    TLS206_PERSON, TLS227_PERS_PUBLN.person_id == TLS206_PERSON.person_id
).join(
    TLS211_PAT_PUBLN, TLS227_PERS_PUBLN.pat_publn_id == TLS211_PAT_PUBLN.pat_publn_id
).filter(
    TLS206_PERSON.person_id == 5210554
).group_by(
    TLS227_PERS_PUBLN.person_id,
    TLS206_PERSON.person_name,
    TLS211_PAT_PUBLN.appln_id
).having(
    func.count(TLS211_PAT_PUBLN.pat_publn_id) > 1  # Only show applications with multiple publications
).order_by(
    'publication_count'
)

res3 = patstat.df(q)
res3
```

Out[4]:

	person_id	person_name	appln_id	publication_count
0	5210554	General Electric Company	566240027	2
1	5210554	General Electric Company	537573598	2
2	5210554	General Electric Company	510301881	2
3	5210554	General Electric Company	378955611	2
4	5210554	General Electric Company	470850836	2
...	...	...	...	...
15051	5210554	General Electric Company	439652765	3
15052	5210554	General Electric Company	407341310	3
15053	5210554	General Electric Company	406608141	3
15054	5210554	General Electric Company	405817816	3
15055	5210554	General Electric Company	405819620	3

15056 rows × 4 columns

This query compares the sequence numbers (both for applicants and inventors) across different publications for the same application, allowing for the detection of shifts in the list of applicants and inventors.

In [5]:

```
q4 = db.query(
    TLS227_PERS_PUBLN.person_id,
    TLS206_PERSON.person_name,
    TLS211_PAT_PUBLN.appln_id,
    TLS211_PAT_PUBLN.publn_nr,
    TLS227_PERS_PUBLN.applt_seq_nr,
    TLS227_PERS_PUBLN.invt_seq_nr
).join(
    TLS206_PERSON, TLS227_PERS_PUBLN.person_id == TLS206_PERSON.person_id
).join(
    TLS211_PAT_PUBLN, TLS227_PERS_PUBLN.pat_publn_id == TLS211_PAT_PUBLN.pat_publn_id
).filter(
    TLS211_PAT_PUBLN.appln_id == 473378055
).order_by(
    TLS227_PERS_PUBLN.applt_seq_nr,  # Order by applicant sequence number
    TLS227_PERS_PUBLN.invt_seq_nr    # Order by inventor sequence number
)

res4 = patstat.df(q4)
res4
```

Out[5]:

	person_id	person_name	appln_id	publn_nr	applt_seq_nr	invt_seq_nr
0	48462548	Yarbrough, Aaron A.	473378055	2016377051	0	1
1	48379676	Caruso, Christopher Daniel	473378055	10337490	0	1
2	48462548	Yarbrough, Aaron A.	473378055	10337490	0	2
3	12047856	Hynum, Daniel Alan	473378055	2016377051	0	2
4	12047856	Hynum, Daniel Alan	473378055	10337490	0	3
5	48379676	Caruso, Christopher Daniel	473378055	2016377051	0	3
6	5210554	General Electric Company	473378055	10337490	1	0
7	5210554	General Electric Company	473378055	2016377051	1	0