

The Patent Technical Relation Table (TLS205_TECH_REL)

Welcome to the fifth table of the PATSTAT database. As the name suggests, this table is about technical relations. Technical relations are "priority-like" relations between applications which have been detected by EPO examiners, but which have not been published by a patent office.

A technical family is created manually on request when documents disclosing identical subject matter (i.e., having identical description and drawings) are not automatically grouped together because they do not claim the same priority or combination of priorities.

The reasons why applicants may decide not to claim a priority are of various kinds: for example, in some cases, the 12-month period foreseen in the Paris Convention might have been exceeded.

Let's look at an example to better understand in which cases we can end up with technical relations and why they can be useful and meaningful. The EPO received a request for technical merge of the documents GB2542582, GB2542583 and GB2542584. These three patent applications, concerning a stretcher arrangement, were filed on the same day by the same applicant. The description and figures are in all three cases identical, but the claims cover three different aspects of the same invention: a pivoting arrangement for body support panels using coaxial actuators, a stretcher wheel assembly with solenoid activated locking mechanism and a pivoting body support for a stretcher, respectively. Since they were filed on the same day, they could not claim each other's priority and originally entered the EPO search collection as first filings. This was changed by the EPO expert in order to create the technical relations which we now see in the EPO databases.

```
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 TLS205_TECH_REL
```

APPLN_ID (Primary Key)

Application ID linking this table to table TLS201. If we want to retrieve earlier applications from table TLS201 that are claimed by applications in table TLS205 we have to JOIN between the two tables imposing the match between `tech_rel_appln_id` in table TLS205 and `appln_id` in table TLS201. The reason is the same explained for table TLS204.

```
In [4]: # Import table TLS201
from epo.tipdata.patstat.database.models import TLS201_APPLN

tech_id = db.query(
    TLS201_APPLN.inpadoc_family_id,
    TLS205_TECH_REL.tech_rel_appln_id,
    TLS201_APPLN.appln_id,
    TLS201_APPLN.appln_filing_date,
    TLS201_APPLN.earliest_filing_date
).join(
    TLS201_APPLN, TLS205_TECH_REL.tech_rel_appln_id == TLS201_APPLN.appln_id # Join table TLS201 and TLS205
).limit(10000)

tech_id_df = patstat.df(tech_id)
tech_id_df
```

Out [4]:

	inpadoc_family_id	tech_rel_appln_id	appln_id	appln_filing_date	earliest_filing_date
0	11330364	907848322	907848322	9999-12-31	9999-12-31
1	11336970	908548029	908548029	9999-12-31	9999-12-31
2	9450268	908034541	908034541	9999-12-31	9999-12-31
3	9450268	908034541	908034541	9999-12-31	9999-12-31
4	18642582	908671225	908671225	9999-12-31	9999-12-31
...
9995	20761113	907669819	907669819	1954-06-30	1954-06-30
9996	20756272	908415158	908415158	1954-04-26	1954-04-26
9997	20751388	908602916	908602916	1954-03-10	1954-03-10
9998	21140451	908471878	908471878	1954-03-08	1954-03-08
9999	22226241	907481742	907481742	1954-10-22	1954-10-22

10000 rows × 5 columns

TECH_REL_APPLN_ID

This field defines the relationship between an application and a prior application (priority). If the priority-linkage-type is equal to 'T', then the priority is a technically related priority. The value of `appln_id` for this application is the `tech_rel_appln_id`.

TECH_REL_APPLN_ID is linked to INPADOC_FAMILY_ID

Let's show that applications referring to the same prior application (via technical relation) fall under the same INPADOC family but may possibly belong to different DOCDB families. In order to show this, we join table TLS201 and TLS205 via the `appln_id` attribute and retrieve the `docdb_family_id` for a fixed `inpadoc_family_id`.

```
In [3]: doc_inpa = db.query(
    TLS201_APPLN.appln_id,
    TLS205_TECH_REL.tech_rel_appln_id,
    TLS201_APPLN.inpadoc_family_id,
    TLS201_APPLN.docdb_family_id
).join(
    TLS201_APPLN, TLS205_TECH_REL.appln_id == TLS201_APPLN.appln_id # Join tables TLS201 and TLS205 via the appln_id attribute
).filter(
    TLS201_APPLN.inpadoc_family_id == 98645
).order_by(
    TLS205_TECH_REL.tech_rel_appln_id
)

doc_inpa_df = patstat.df(doc_inpa)
doc_inpa_df
```

Out[3]:

	appln_id	tech_rel_appln_id	inpadoc_family_id	docdb_family_id
0	441452953	413899147	98645	53373125
1	413899147	441452953	98645	51867932

In []: