The TLS803_LEGAL_EVENT_CODE Table

The `TLS803_LEGAL_EVENT_CODE` table serves as a comprehensive reference for legal event codes used in the European Patent Office's worldwide legal event database, also known as the INPADOC database. This table catalogs various legal event codes, each representing a distinct type of legal event, such as the payment of fees, lapses, reinstatements, and other significant actions within the lifecycle of a patent. To facilitate easier analysis, similar legal event codes are grouped into categories, providing insight into the nature of these events based on their `EVENT_CATEGORY_CODE` and `EVENT_CATEGORY_TITLE`, which are aligned with the WIPO ST.27 standard

The `TLS803_LEGAL_EVENT_CODE` table is closely connected to the `TLS231_INPADOC_LEGAL_EVENT` table, which records individual occurrences of these legal events for specific patents. Through the `EVENT_AUTH` and `EVENT_CODE` columns, `TLS803_LEGAL_EVENT_CODE` can be joined with `TLS231_INPADOC_LEGAL_EVENT` to provide more context about each event. While `TLS231_INPADOC_LEGAL_EVENT` logs events at the patent level—such as the application ID (`appln_id`), event ID (`event_id`), and event effective date (`event_effective_date`)— `TLS803_LEGAL_EVENT_CODE` enriches this information by supplying detailed descriptions and category data.

By combining the information from both tables, we can analyze legal events for specific patents with a clear understanding of what each event entails and how it fits into the broader lifecycle of a patent. This notebook will guide you through querying these tables, retrieving meaningful event data, and exploring the types and categories of legal events associated with patent applications.

The INPADOC classification scheme was developed to simplify navigation through the extensive INPADOC database, which contains over 250 million records of legal events related to patents. To address the challenge of accessing specific data efficiently, the EPO introduced a structured classification scheme that groups events based on their nature, making the data easier to retrieve and understand.

A high-level classification, following the WIPO ST.27 standard, organizes events into broad categories. There are currently 21 main categories, identified by single letters from A to Z. For example:

Category	Title	Category	Title
A	Application filing	P	Re-publication of document after modification
B	Application discontinuation	Q	Document publication
C	Application revival	R	Party data change
D	Search and examination	S	Information on licensing and similar transactions
E	Pre-grant review request	T	Administrative procedure adjustment
F	IP right grant	U	Payment
G	Protection beyond IP right term	V	Appeal
H	IP right cessation	W	Other
K	IP right revival	Y	Correction and deletion of event information
L	IP right review request	Z	Classification pending
M	IP right maintenance		

Each category groups related events, such as application filings, grant procedures, licensing, and corrections.

In [1]:
```
from epo.tipdata.patstat import PatstatClient
from epo.tipdata.patstat.database.models import (
    TLS201_APPLN,
    TLS231_INPADOC_LEGAL_EVENT,
    TLS803_LEGAL_EVENT_CODE,
    TLS224_APPLN_CPC
)
from sqlalchemy import and_, case, func, select, distinct, or_

# Initialise the PATSTAT client
patstat = PatstatClient(env="TEST")

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

Key Attributes of the TLS803_LEGAL_EVENT_CODE Table

EVENT_AUTH

This attribute indicates the authority or jurisdiction that recorded the legal event, represented as a two-letter code, following WIPO's ST.3 standard. For example, "EP" represents the European Patent Office, while "US" represents the United States Patent and Trademark Office.

EVENT_CODE

This attribute contains the unique code assigned to a specific legal event within the patent lifecycle, such as the filing of an application, payment of fees, lapse, or reinstatement. It helps categorize and distinguish between different types of events.

EVENT_DESCR

This attribute provides a description of the legal event in English, explaining the nature of the event associated with the EVENT_CODE. It offers a standardized explanation of each event, making it easier to understand its impact on a patent's status.

EVENT_DESCR_ORIG

This attribute gives the original-language description of the legal event, useful for non-English-speaking jurisdictions. It helps users interpret events in the native language of the patent authority.

EVENT_CATEGORY_CODE

This attribute groups similar legal events under a unified category code. By clustering events with similar implications, it simplifies the analysis of events by their type or effect on a patent.

EVENT_CATEGORY_TITLE

This attribute provides the title of the event category, further clarifying the nature of the grouped events in EVENT_CATEGORY_CODE. It enables users to quickly understand the general type or purpose of each legal event category.

In [2]:
```
legal_event_join_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS803_LEGAL_EVENT_CODE.event_auth,
        TLS803_LEGAL_EVENT_CODE.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS803_LEGAL_EVENT_CODE.event_descr,
        TLS803_LEGAL_EVENT_CODE.event_descr_orig,
        TLS803_LEGAL_EVENT_CODE.event_category_code,
        TLS803_LEGAL_EVENT_CODE.event_category_title
    )
    .join(
        TLS803_LEGAL_EVENT_CODE,
        (TLS231_INPADOC_LEGAL_EVENT.event_code == TLS803_LEGAL_EV
ENT_CODE.event_code) &
        (TLS231_INPADOC_LEGAL_EVENT.event_auth == TLS803_LEGAL_EV
ENT_CODE.event_auth)
    )
    .filter(TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9
999-12-31')
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)  # Order by ap
plication ID
)

legal_event_join_results = patstat.df(legal_event_join_query)
legal_event_join_results
```

Out[2]:

	appln_id	event_id	event_auth	event_code	event_effective_date	ev
0	145	263741582	AT	MM01	2012-08-31	LAPSE BE NOT PAYIN
1	145	309368459	DE	R081	2014-09-26	C APPLICANT/
2	145	106624995	EP	BERE	2009-08-31	B
3	145	74260856	EP	17Q	2008-06-03	FIRST EXA DE
4	145	162342943	EP	GBPC	2011-08-01	GB: E PATEN THRO F
...	...	...	...	...	...	
324381	606036745	1079218205	US	AS	2018-02-12	ASS
324382	606039115	1079215987	US	AS	2022-04-02	ASS
324383	606039173	1079207446	US	AS	2023-12-11	ASS
324384	606078571	1078694611	JP	A521	2023-09-28	REC AMENDM
324385	606078580	1078694618	JP	A521	2023-09-26	REC AMENDM

324386 rows × 9 columns

In [2]:
```
active_energy_patents_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_id,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.class_symbol,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS803_LEGAL_EVENT_CODE.event_category_code,
        TLS803_LEGAL_EVENT_CODE.event_category_title,
    )
    # Join TLS231 and TLS803 tables on EVENT_AUTH and EVENT_CODE
    .join(
        TLS803_LEGAL_EVENT_CODE,
        (TLS231_INPADOC_LEGAL_EVENT.event_auth == TLS803_LEGAL_EV
ENT_CODE.event_auth) &
        (TLS231_INPADOC_LEGAL_EVENT.event_code == TLS803_LEGAL_EV
ENT_CODE.event_code)
    )
    # Filter for the GENERATION, CONVERSION, OR DISTRIBUTION OF E
LECTRIC POWER (H02)
    .filter(TLS231_INPADOC_LEGAL_EVENT.class_symbol.like("H02%"),
            TLS231_INPADOC_LEGAL_EVENT.event_code != "H",
            TLS231_INPADOC_LEGAL_EVENT.event_effective_date != '9
999-12-31')

    # Order by application ID
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)


active_energy_patents_results = patstat.df(active_energy_patents_
query)
active_energy_patents_results
```

Out[2]:

	appln_id	event_id	event_effective_date	class_symbol	event_code	event_categ
0	13484302	351938249	2015-04-22	H02S0040380000	R079	
1	15035142	281734549	2014-02-05	H02S0040300000	R079	
2	15063151	281735574	2014-02-03	H02S0050000000	R079	
3	15092091	235232605	2013-04-26	H02P0029000000	R079	

4	274879479	282773254	2014-02-06	H02S0040300000	R079
5	336609934	216416128	2013-01-17	H02K0021240000	R079
6	339191728	281741756	2014-02-05	H02S0050100000	R079
7	353627993	281738197	2014-01-30	H02S0040360000	R079
8	364028575	282772837	2014-02-07	H02S0050100000	R079
9	378211830	282772943	2014-02-06	H02S0050100000	R079
10	379470397	282769765	2014-02-06	H02S0040300000	R079
11	405070606	282764440	2014-02-10	H02S0010120000	R079
12	405070606	282774179	2012-06-25	H02N0006000000	R079
13	406804874	281740098	2014-02-05	H02S0030200000	R079
14	409464819	295284012	2014-06-18	H02K0021020000	R079
15	420727754	333924861	2015-01-30	H02S0020000000	R079

Some Possible Analysis

This query is specifically identifying applications with the "Unitary Effect Registered" status under the European Patent Office (EPO), indicated by the event code "U07" and the authority "EP." By selecting the application IDs, event details, effective dates, designated states, and IPC classifications, we aim to see which applications have attained this unitary effect status. This helps us analyze which patents, particularly those in a specified IPC class, have been registered with unitary effect, giving insight into the geographic and legal coverage they possess across designated states.

In [3]:
```
unitary_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS231_INPADOC_LEGAL_EVENT.designated_states,
        TLS231_INPADOC_LEGAL_EVENT.class_symbol
    )
    .filter(
        and_(
            TLS231_INPADOC_LEGAL_EVENT.event_code == "U07",
            TLS231_INPADOC_LEGAL_EVENT.event_auth == "EP"
        )
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)


unitary_results = patstat.df(unitary_query)
unitary_results
```

Out[3]:

	appln_id	event_auth	event_code	event_effective_date	
0	332577328	EP	U07	2023-06-26	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
1	379298573	EP	U07	2023-07-17	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
2	404886357	EP	U07	2023-08-22	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
3	407716238	EP	U07	2023-08-04	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
4	414640625	EP	U07	2023-08-24	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
...	...	...	...	...	
254	557589536	EP	U07	2023-12-15	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
255	558477883	EP	U07	2023-10-17	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
256	559363921	EP	U07	2023-09-27	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
257	564066674	EP	U07	2023-09-25	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI
258	568000314	EP	U07	2023-07-31	AT,BE,BG,DE,DK,EE,FI,FR,IT,LI

259 rows × 6 columns

In this query, we are expanding our analysis of the "Unitary Effect Registered" event (code "U07") for patents under the European Patent Office (EPO), but we are now incorporating data from the CPC (Cooperative Patent Classification) codes. This allows us to examine not only which applications have the unitary effect status but also the specific technological areas or classifications they fall under, as indicated by their CPC class symbols.

In [4]:
```
unitary_cpc_query = (
    db.query(
        TLS231_INPADOC_LEGAL_EVENT.appln_id,
        TLS231_INPADOC_LEGAL_EVENT.event_auth,
        TLS231_INPADOC_LEGAL_EVENT.event_code,
        TLS231_INPADOC_LEGAL_EVENT.event_effective_date,
        TLS224_APPLN_CPC.cpc_class_symbol # add table TLS224
    )
    .join(
        TLS224_APPLN_CPC,
        TLS224_APPLN_CPC.appln_id == TLS231_INPADOC_LEGAL_EVENT.a
ppln_id  # Join on application ID
    )
    .filter(
        and_(
            TLS231_INPADOC_LEGAL_EVENT.event_code == "U07",
            TLS231_INPADOC_LEGAL_EVENT.event_auth == "EP"
        )
    )
    .order_by(TLS231_INPADOC_LEGAL_EVENT.appln_id)
)


unitary_cpc_results = patstat.df(unitary_cpc_query)
unitary_cpc_results
```

Out[4]:

	appln_id	event_auth	event_code	event_effective_date	cpc_class_symbol
0	332577328	EP	U07	2023-06-26	F03D 9/255
1	332577328	EP	U07	2023-06-26	F05B2270/337
2	332577328	EP	U07	2023-06-26	H02J 3/1885
3	332577328	EP	U07	2023-06-26	H02J 3/46
4	332577328	EP	U07	2023-06-26	H02J2300/28
...	...	...	...	...	...
2601	568000314	EP	U07	2023-07-31	F03B 13/1815
2602	568000314	EP	U07	2023-07-31	F03B 13/22
2603	568000314	EP	U07	2023-07-31	F03B 17/02
2604	568000314	EP	U07	2023-07-31	F05B2210/12
2605	568000314	EP	U07	2023-07-31	Y02E 10/30

2606 rows × 5 columns

This query counts the number of European patents under unitary effect (event code "U07") for each CPC class. By joining the `TLS224_APPLN_CPC` table (which contains CPC classifications) with `TLS231_INPADOC_LEGAL_EVENT` (which records legal events), we filter to include only records where the event authority is the EPO and the event is specifically the unitary effect.

The result is grouped by `cpc_class_symbol` to tally how often each CPC class appears with unitary effect, and the counts are sorted in descending order to show the CPC classes with the highest frequency. This analysis reveals which technological classes have the most patents benefiting from the unitary effect.

In [5]:
```
cpc_count_query = (
    db.query(
        TLS224_APPLN_CPC.cpc_class_symbol,
        func.count(TLS231_INPADOC_LEGAL_EVENT.appln_id).label("co
unt")
    )
    .join(
        TLS231_INPADOC_LEGAL_EVENT,
        TLS224_APPLN_CPC.appln_id == TLS231_INPADOC_LEGAL_EVENT.a
ppln_id
    )
    .filter(
        and_(
            TLS231_INPADOC_LEGAL_EVENT.event_code == "U07",
            TLS231_INPADOC_LEGAL_EVENT.event_auth == "EP"
        )
    )
    .group_by(TLS224_APPLN_CPC.cpc_class_symbol)
    .order_by(func.count(TLS231_INPADOC_LEGAL_EVENT.appln_id).lab
el("count").desc())
)

cpc_count_results = patstat.df(cpc_count_query)
cpc_count_results
```

Out[5]:

	cpc_class_symbol	count
0	Y02E 10/72	197
1	Y02P 70/50	40
2	F03D 17/00	38
3	F03D 1/0675	32
4	F03D 13/10	30
...	...	...
1096	F03D 7/044	1
1097	G06F 15/163	1
1098	H02J 7/0048	1
1099	H02J 7/0071	1
1100	H02J 7/04	1

1101 rows × 2 columns

You can use the site https://www.cooperativepatentclassification.org/home (https://www.cooperativepatentclassification.org/home) to explore CPC classes and gain a deeper understanding of the types of technologies each class represents.