

The REG201_PROC_STEP: Procedural step Table

Welcome to the REG201_PROC_STEP table overview. This table contains essential procedural data that is part of the European Patent (EP) register. While not all of this information is published in the bulletin, it provides insights into various procedural steps throughout the patent application process. Each record represents a specific procedural step in a patent's lifecycle, capturing details such as the step phase, code, result, affected countries, and any relevant time limits. This data is crucial for understanding the procedural aspects of patents, particularly those that are not fully disclosed in the public bulletin. Let's dive into the attributes that shape the flow of patent applications within the EP system.

The REG201_PROC_STEP table plays a fundamental role in documenting the procedural steps a patent application undergoes throughout its lifecycle at the European Patent Office. Each step is associated with a publication event recorded in the EPO Bulletin, and the BULLETIN_YEAR and BULLETIN_NR fields serve as key timestamps for tracking these events.

```
In [6]: from epo.tipdata.patstat import PatstatClient
from epo.tipdata.patstat.database.models import REG201_PROC_STEP,
REG101_APPLN
from sqlalchemy import select, func, case, select, and_
patstat = PatstatClient(env='PROD')
db = patstat.orm()
```

```
In [ ]: q = db.query(
    REG201_PROC_STEP.id,
    REG201_PROC_STEP.step_id,
    REG201_PROC_STEP.step_phase,
    REG201_PROC_STEP.step_code,
    REG201_PROC_STEP.step_result,
    REG201_PROC_STEP.step_result_type,
    REG201_PROC_STEP.step_country,
    REG201_PROC_STEP.time_limit,
    REG201_PROC_STEP.time_limit_unit,
    REG201_PROC_STEP.bulletin_year,
    REG201_PROC_STEP.bulletin_nr
)
res = patstat.df(q)
res
```

Key Fields in the REG201_PROC_STEP Table

ID (Primary Key)

The ID field serves as a technical identifier that uniquely connects patent applications across various tables.

```
In [3]: q = db.query(  
    REG201_PROC_STEP.id  
).limit(100)  
  
res = patstat.df(q)  
res
```

Out [3]:

	id
0	17706159
1	9734294
2	20855522
3	6733636
4	90120043
...	...
95	17160700
96	95934862
97	22195714
98	90905999
99	12158527

100 rows × 1 columns

STEP_ID (Primary Key)

The `STEP_ID` is a unique identifier assigned to each procedural step within the patent application process. It is present in multiple tables, including `REG201_PROC_STEP`, `REG202_PROC_STEP_TEXT`, `REG203_PROC_STEP_DATE`, `REG721_PROC_STEP`, `REG722_PROC_STEP_TEXT`, and `REG723_PROC_STEP_DATE`. This attribute serves as a key reference for identifying and linking procedural steps across various datasets. With a domain of up to 30 characters, it ensures precise tracking and organization of procedural events associated with patent applications.

STEP_PHASE

The `STEP_PHASE` attribute represents the phase during which a procedural step occurred in the lifecycle of a patent application. Found in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables, it categorizes procedural actions into specific stages using standardized codes of up to 5 ASCII characters.

Defaulting to UNDEF if undefined, the possible values include:

- EXAMN: Examination phase
- APEXA: Appeal in examination
- OPPOS: Opposition phase
- APOPP: Appeal in opposition
- LIMIT: Limitation phase
- REGEN: Entry into the regional phase
- INTEX: International examination
- PROPP: Petition for review in opposition
- REVOC: Revocation phase

```
In [9]: q = db.query(  
    REG201_PROC_STEP.id,  
    func.count(REG201_PROC_STEP.step_id).label("num_steps")  
).group_by(REG201_PROC_STEP.id)  
  
res = patstat.df(q)  
res
```

Out[9]:

	id	num_steps
0	6797255	43
1	2726636	28
2	7012424	32
3	5777180	34
4	1952657	27
...
6933139	5104353	24
6933140	10709479	24
6933141	98955671	24
6933142	11001053	24
6933143	2001337	24

6933144 rows × 2 columns

In the following, the different types of steps.

```
In [10]: q = db.query(  
    REG201_PROC_STEP.step_phase  
).distinct()  
  
res = patstat.df(q)  
res
```

Out [10]:

	step_phase
0	REGEN
1	APEXA
2	PROPP
3	REVOC
4	APOPP
5	UNDEF
6	EXAMN
7	INTEX
8	LIMIT
9	OPPOS

Let's count how many occurrences we have for each step phase.

```
In [15]: q = db.query(
    REG201_PROC_STEP.step_phase,
    func.count(REG201_PROC_STEP.step_phase).label("phase_count")
).group_by(REG201_PROC_STEP.step_phase)

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

Out [15]:

	step_phase	phase_count
0	EXAMN	44656555
1	REGEN	15542642
2	APEXA	10730
3	PROPP	96
4	REVOC	1
5	APOPP	25103
6	UNDEF	14485582
7	INTEX	891749
8	LIMIT	254
9	OPPOS	378510

STEP_CODE

The `STEP_CODE` attribute represents a mnemonic identifier for a procedural step in the patent process. This code is essential for categorizing and identifying specific actions associated with procedural events. It is found in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables and provides a concise alphanumeric representation of each procedural step. The attribute does not have a default value, allowing it to adapt to various scenarios. Codes can be up to 10 characters long, offering flexibility while maintaining clarity. Examples of such codes include OREX and AREX, which correspond to distinct procedural actions. In the 2014 Autumn Edition of the database, 42 unique procedural step codes were identified. The `STEP_CODE` ensures a consistent and standardized way to reference procedural steps, enhancing data management and accessibility for users analyzing procedural details.

```
In [16]: q = db.query(
    REG201_PROC_STEP.step_code
).distinct()

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

Out [16]:

step_code

0	EXRE
1	ADWI
2	IGRA
3	TRAN
4	DEST
5	ORAL
6	REJO
7	OREX
8	REES
9	OPPC
10	WINT
11	EXPT
12	DOBS
13	LIRE
14	REJR
15	EXAM_PCT
16	REVO
17	PART
18	ISAT
19	IGRE
20	CDEC
21	PMAP
22	FFEE
23	RFPR
24	IDOP
25	OBSO
26	PFEE
27	SFEE
28	PROL
29	PREX
30	RAEX
31	REFU
32	OBAP
33	ACOR

34	RFEE
35	LOPR
36	AGRA
37	OPEX
38	VAPT
39	DDIV
40	ABEX
41	RDEC
42	TIPA
43	CORS

STEP_RESULT

The `STEP_RESULT` attribute captures the outcome of a procedural step within the patent process. This attribute is included in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables and provides valuable information about the resolution or status of specific procedural actions. The attribute's domain allows for a variety of predefined values, such as:

- "yes"
- "no"
- "Request accepted"
- "Request deemed not to be filed"
- "Request granted"
- "Request procedure closed"
- "Request rejected"
- "Request withdrawn" It also permits an empty string as a valid entry, which serves as the default value when no result is specified. The `STEP_RESULT` attribute plays a key role in documenting the procedural journey of a patent application, offering clarity and transparency regarding the decisions made at various stages.

```
In [17]: q = db.query(
    REG201_PROC_STEP.step_result
).distinct()

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

Out [17]:

	step_result
0	Request withdrawn
1	no
2	Request granted
3	Request procedure closed
4	Request deemed not to be filed
5	
6	Request accepted
7	Request rejected
8	yes

STEP_RESULT_TYPE

The `STEP_RESULT_TYPE` attribute defines the type of result associated with a procedural step in the patent process. This attribute is present in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables and categorizes the result of procedural actions into specific types. The domain for this attribute includes the values accepted(yes/no), RESULT, or an empty string, which serves as the default when no result type is specified. By distinguishing between these types, the `STEP_RESULT_TYPE` attribute enhances the granularity of procedural documentation, providing clearer insight into the nature and classification of outcomes in the patent process.

```
In [18]: q = db.query(
    REG201_PROC_STEP.step_result_type
).distinct()

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

Out [18]:

	step_result_type
0	
1	accepted(yes/no)
2	RESULT

STEP_COUNTRY

The `STEP_COUNTRY` attribute represents the office code of the state affected by a specific procedural step. Found in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables, it identifies the office responsible for actions such as conducting preliminary examinations. The attribute follows the WIPO ST.3 standard, using two-character codes (A-Z) to denote countries or regions.

For example, in international applications under Chapter 2 of the PCT, the procedural step code "PREX" indicates that the preliminary examination was conducted, with the `STEP_COUNTRY` value specifying the office responsible (e.g., "EP" for the European Patent Office). If no office is implicated, the attribute defaults to an empty value. By providing this information, the `STEP_COUNTRY` attribute ensures traceability of procedural steps and their jurisdictional relevance within the patent examination process.

```
In [20]: q = db.query(
    REG201_PROC_STEP.step_country,
    func.count(REG201_PROC_STEP.step_id).label("steps_per_country")
).group_by(REG201_PROC_STEP.step_country)

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

Out [20]:

	step_country	steps_per_country
0	CN	10388
1	DK	2
2	SG	708
3	CA	4842
4	RU	3685
5	IT	28
6	LU	150
7	JP	75140
8	TR	256
9	LI	11
10	PH	2
11	US	188593
12	SE	34659
13	DE	50322
14	EG	26
15	CH	3

16		75104903
17	EP	465377
18	IN	507
19	PT	1
20	IL	851
21	XV	29
22	XN	555
23	AU	25507
24	KR	9250
25	FI	1598
26	BE	2151
27	BR	810
28	AT	4816
29	ES	1538
30	GB	3161
31	CL	98
32	FR	1134
33	UA	48
34	NL	73

TIME_LIMIT

The `TIME_LIMIT` attribute specifies a time limit associated with a procedural step in the patent process. Found in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables, it provides critical timing information for actions required or deadlines imposed during procedural steps. The domain allows up to 10 characters, with values such as empty (default), numeric indicators (e.g., "01", "06", "12"), or specific formats like "M04" (representing a 4-month deadline).

Occasionally, more complex values like "[1986/51]" are used, indicating a deadline set for week 51 of the year 1986. By recording time constraints in procedural steps, the `TIME_LIMIT` attribute ensures structured tracking of deadlines, enabling efficient management of time-sensitive tasks in the patent lifecycle.

TIME_LIMIT_UNIT

The `TIME_LIMIT_UNIT` attribute defines the unit of measurement for the time limit associated with a procedural step. It is included in the `REG201_PROC_STEP` and `REG721_PROC_STEP` tables to provide clarity on the nature of the deadline. The default value is an empty string, indicating no specific unit.

The domain allows up to 6 characters, with "months" being the only commonly used non-default value. This attribute ensures that the duration specified in the corresponding `TIME_LIMIT` attribute is unambiguously interpreted, supporting accurate scheduling and adherence to procedural deadlines in the patent process.

BULLETIN_YEAR

In the PATSTAT database, the `BULLETIN_YEAR` field captures the year when an action or event related to a patent application was published in the EPO Bulletin. This field plays a critical role in tracking the timeline of patent events, ensuring chronological accuracy in analyses.

The `BULLETIN_YEAR` is a 4-digit numeric field (formatted as YYYY), with a default value of 0 to indicate cases where no bulletin publication is known. For entries where publication in the EPO Bulletin is confirmed, `BULLETIN_YEAR` reflects the corresponding year of publication. It is used in conjunction with `BULLETIN_NR`, which specifies the bulletin issue number.

BULLETIN_NR

The `BULLETIN_NR` attribute represents the issue number of the EPO Bulletin in which a specific action has been published. This number indicates the calendar week during which the Bulletin was released. It serves as a reference for identifying the exact edition of the EPO Bulletin where actions such as patent grants, publications, or other significant events are announced.

If the action was not published in the Bulletin or if the information is unknown, the default value of 0 is used for the `BULLETIN_NR`, which corresponds to the absence of a known bulletin number. This value is only used when the associated `BULLETIN_YEAR` is also set to 0.

```
In [8]: q = db.query(  
    REG201_PROC_STEP.id,  
    REG201_PROC_STEP.bulletin_year,  
    REG201_PROC_STEP.bulletin_nr  
).filter(  
    REG201_PROC_STEP.bulletin_year != 0  
)  
  
res = patstat.df(q)  
res
```

Out[8]:

		id	bulletin_year	bulletin_nr
0		17935369	2020	53
1		13174475	2015	53
2		7012494	2009	53
3		17741955	2020	53
4		11722787	2015	53
...	
3463190		11183783	2013	52
3463191		21839705	2023	52
3463192		18943878	2021	52
3463193		19957037	2022	52
3463194		17887076	2019	52

3463195 rows × 3 columns

In []: