

Tracking Reproducible Revisions in PostgreSQL with Psycopg2

Miroslav Šedivý

 @eumiro

1 / 44




Tracking Reproducible Revisions in PostgreSQL with Psycopg2

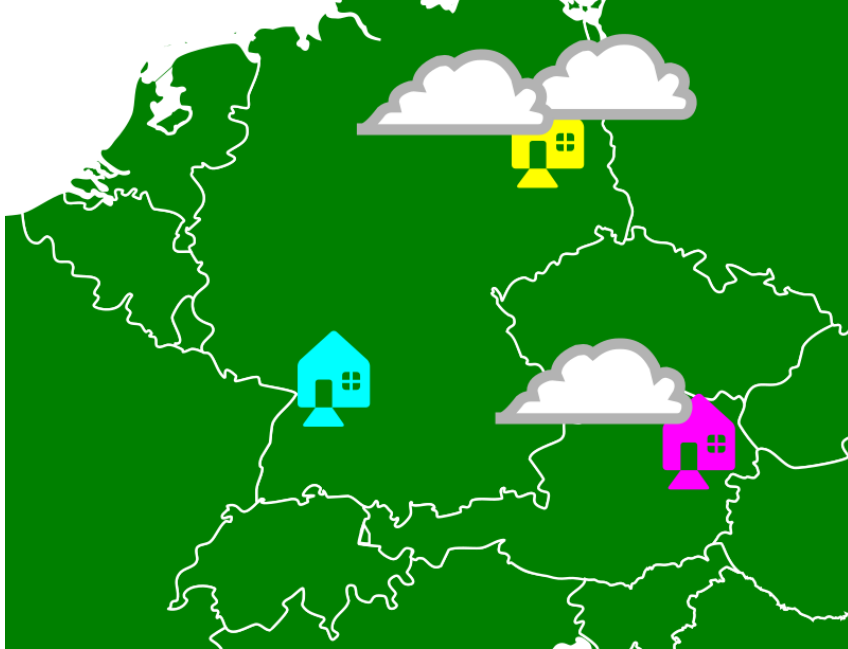
INSERT, UPDATE and DELETE without losing information

- Time-versioning entities with attributes
- RANGE types in PostgreSQL 9.x+
- GIST extension
- Python and Psycopg2
- Modifying data (concurrently)
- Reading data (consistently)

Miroslav Šedivý

[ˈmɪrɔslav ˈʃɛɪvɪː]

-   @eumiro
-  šedivý
- born in Bratislava, Czechoslovakia
- M.Sc. at INSA Lyon, France
- used MySQL and Oracle before PostgreSQL
- came to Python 2.5 from Perl/Java in 2008
- Senior Software Architect at UBIMET GmbH in Karlsruhe, Germany
- using Python to make the sun shine and the wind blow



Status Quo

```
CREATE TABLE generator (generator_id SERIAL,  
name TEXT,  
coords GEOMETRY,  
power REAL);
```

Status Quo

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL);
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power
1	KA	8.4/49.0	4
2	BER	13.4/52.5	6

Status Quo

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL);
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power
1	KA	8.4/49.0	4
2	BER	13.4/52.5	6

```
INSERT INTO generator (name, coords, power) VALUES ('VIE', '16.4/48.2', 8);
```

generator_id	name	coords	power
1	KA	8.4/49.0	4
2	BER	13.4/52.5	6
3	VIE	16.4/48.2	8

When was the entry updated?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```


When was the entry updated?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	updated_on
1	KA	8.4/49.0	4	2018-01-01
2	BER	13.4/52.5	6	2018-01-01

When was the entry updated?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	updated_on
1	KA	8.4/49.0	4	2018-01-01
2	BER	13.4/52.5	6	2018-01-01

```
INSERT INTO generator (name, coords, power) VALUES ('VIE', '16.4/48.2', 8);
```

generator_id	name	coords	power	updated_on
1	KA	8.4/49.0	4	2018-01-01
2	BER	13.4/52.5	6	2018-01-01
3	VIE	16.4/48.2	8	2018-06-29

When was the entry updated?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	updated_on
1	KA	8.4/49.0	4	2018-01-01
2	BER	13.4/52.5	6	2018-01-01

```
INSERT INTO generator (name, coords, power) VALUES ('VIE', '16.4/48.2', 8);
```

generator_id	name	coords	power	updated_on
1	KA	8.4/49.0	4	2018-01-01
2	BER	13.4/52.5	6	2018-01-01
3	VIE	16.4/48.2	8	2018-06-29

- UPDATE / DELETE ?

When is the entry valid?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          activity TSTZRANGE DEFAULT TSTZRANGE(NOW(), NULL),  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```

When is the entry valid?

```
CREATE TABLE generator (generator_id SERIAL,  
                          name TEXT,  
                          coords GEOMETRY,  
                          power REAL,  
                          activity TSTZRANGE DEFAULT TSTZRANGE(NOW(), NULL),  
                          updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	activity	updated_on
1	KA	8.4/49.0	4	[2018-01-01, NULL)	2018-01-01
2	BER	13.4/52.5	6	[2018-01-01, NULL)	2018-01-01

When is the entry valid?

```
CREATE TABLE generator (generator_id SERIAL,  
name TEXT,  
coords GEOMETRY,  
power REAL,  
activity TSTZRANGE DEFAULT TSTZRANGE(NOW(), NULL),  
updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	activity	updated_on
1	KA	8.4/49.0	4	[2018-01-01, NULL)	2018-01-01
2	BER	13.4/52.5	6	[2018-01-01, NULL)	2018-01-01

```
INSERT INTO generator (name, coords, power) VALUES ('VIE', '16.4/48.2', 8);
```

generator_id	name	coords	power	activity	updated_on
1	KA	8.4/49.0	4	[2018-01-01, NULL)	2018-01-01
2	BER	13.4/52.5	6	[2018-01-01, NULL)	2018-01-01
3	VIE	16.4/48.2	8	[2018-06-29, NULL)	2018-06-29

When is the entry valid?

```
CREATE TABLE generator (generator_id SERIAL,  
name TEXT,  
coords GEOMETRY,  
power REAL,  
activity TSTZRANGE DEFAULT TSTZRANGE(NOW(), NULL),  
updated_on TIMESTAMPTZ DEFAULT NOW());
```

```
SELECT * FROM generator;
```

generator_id	name	coords	power	activity	updated_on
1	KA	8.4/49.0	4	[2018-01-01, NULL)	2018-01-01
2	BER	13.4/52.5	6	[2018-01-01, NULL)	2018-01-01

```
INSERT INTO generator (name, coords, power) VALUES ('VIE', '16.4/48.2', 8);
```

generator_id	name	coords	power	activity	updated_on
1	KA	8.4/49.0	4	[2018-01-01, NULL)	2018-01-01
2	BER	13.4/52.5	6	[2018-01-01, NULL)	2018-01-01
3	VIE	16.4/48.2	8	[2018-06-29, NULL)	2018-06-29

- update an entry?

```
CREATE EXTENSION btree_gist;
```



```
CREATE EXTENSION btree_gist;
```

```
CREATE TABLE generator (  
  generator_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL UNIQUE,  
  coords GEOMETRY NOT NULL,  
  power REAL NOT NULL,  
  activity TSTZRANGE NOT NULL,  
  EXCLUDE USING GIST (generator_id WITH =,  
                      activity WITH &&)  
);
```

```
CREATE EXTENSION btree_gist;
```

```
CREATE TABLE generator (  
  generator_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL UNIQUE,  
  coords GEOMETRY NOT NULL,  
  power REAL NOT NULL,  
  activity TSTZRANGE NOT NULL,  
  EXCLUDE USING GIST (generator_id WITH =,  
                      activity WITH &&)  
);
```

generator_id	name	coords	power	activity
1	KA	8.4/49.0	4	[2018-01-01, NULL)
2	BER	13.4/52.5	6	[2018-01-01, 2018-05-01)
2	BER	13.4/52.5	12	[2018-05-01, NULL)

```
CREATE EXTENSION btree_gist;
```

```
CREATE TABLE generator (  
  generator_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL UNIQUE,  
  coords GEOMETRY NOT NULL,  
  power REAL NOT NULL,  
  activity TSTZRANGE NOT NULL,  
  EXCLUDE USING GIST (generator_id WITH =,  
                      activity WITH &&)  
);
```

generator_id	name	coords	power	activity
1	KA	8.4/49.0	4	[2018-01-01, NULL)
2	BER	13.4/52.5	6	[2018-01-01, 2018-05-01)
2	BER	13.4/52.5	12	[2018-05-01, NULL)

- when was this entry valid?

```
CREATE TABLE generator (  
  generator_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL UNIQUE,  
  coords GEOMETRY NOT NULL,  
  power REAL NOT NULL,  
  activity TSTZRANGE NOT NULL,  
  validity TSTZRANGE NOT NULL,  
  EXCLUDE USING GIST (generator_id WITH =,  
                      activity WITH &&,  
                      validity WITH &&)  
);
```

```

CREATE TABLE generator (
  generator_id INTEGER PRIMARY KEY,
  name TEXT NOT NULL UNIQUE,
  coords GEOMETRY NOT NULL,
  power REAL NOT NULL,
  activity TSTZRANGE NOT NULL,
  validity TSTZRANGE NOT NULL,
  EXCLUDE USING GIST (generator_id WITH =,
                      activity WITH &&,
                      validity WITH &&)
);

```

generator_id	name	coords	power	activity	validity
1	KA	8.4/49.0	4	[2018-01-01, NULL)	[2018-01-01, NULL)
2	BER	13.4/52.5	6	[2018-01-01, NULL)	[2018-01-01, 2018-06-29)
2	BER	13.4/52.5	6	[2018-01-01, 2018-05-01)	[2018-06-29, NULL)
2	BER	13.4/52.5	12	[2018-05-01, NULL)	[2018-06-29, NULL)

```
CREATE TABLE generator (  
  generator_id INTEGER PRIMARY KEY,  
  name TEXT NOT NULL UNIQUE);  
  
CREATE TABLE generator_rev(  
  generator_id INTEGER REFERENCES generator NOT NULL,  
  coords GEOMETRY NOT NULL,  
  power REAL NOT NULL,  
  activity TSTZRANGE NOT NULL,  
  validity TSTZRANGE NOT NULL,  
  EXCLUDE USING GIST (generator_id WITH =,  
                      activity WITH &&,  
                      validity WITH &&)  
);
```

```

CREATE TABLE generator (
  generator_id INTEGER PRIMARY KEY,
  name TEXT NOT NULL UNIQUE);

CREATE TABLE generator_rev(
  generator_id INTEGER REFERENCES generator NOT NULL,
  coords GEOMETRY NOT NULL,
  power REAL NOT NULL,
  activity TSTZRANGE NOT NULL,
  validity TSTZRANGE NOT NULL,
  EXCLUDE USING GIST (generator_id WITH =,
                      activity WITH &&,
                      validity WITH &&)
);

```

generator_id	name
1	KA
2	BER

generator_id	coords	power	activity	validity
1	8.4/49.0	4	[2018-01-01, NULL)	[2018-01-01, NULL)
2	13.4/52.5	6	[2018-01-01, NULL)	[2018-01-01, 2018-06-29)
2	13.4/52.5	6	[2018-01-01, 2018-05-01)	[2018-06-29, NULL)
2	13.4/52.5	12	[2018-05-01, NULL)	[2018-06-29, NULL)

```
CREATE TABLE revision (  
  revision_id INTEGER PRIMARY KEY,  
  activity TSTZRANGE DEFAULT TSTZRANGE(NOW()), NULL),  
  desc TEXT NOT NULL UNIQUE,  
  EXCLUDE USING GIST (activity WITH &&)  
);
```

revision_id	activity	desc
0	(NULL, NULL)	In the beginning, the Earth was without form, and void...


```

CREATE TABLE generator (
  generator_id INTEGER PRIMARY KEY,
  name TEXT NOT NULL UNIQUE);

CREATE TABLE generator_rev(
  generator_id INTEGER REFERENCES generator NOT NULL,
  coords GEOMETRY NOT NULL,
  power REAL NOT NULL,
  activity TSTZRANGE NOT NULL,
  revs INT4RANGE NOT NULL,
  EXCLUDE USING GIST (generator_id WITH =,
                      activity WITH &&,
                      revs WITH &&)
);

```

generator_id	name
1	KA
2	BER

generator_id	coords	power	activity	revs
1	8.4/49.0	4	[2018-01-01, NULL)	[1, NULL)
2	13.4/52.5	6	[2018-01-01, NULL)	[1, 2)
2	13.4/52.5	6	[2018-01-01, 2018-05-01)	[2, NULL)
2	13.4/52.5	12	[2018-05-01, NULL)	[2, NULL)

generator_id	name
1	KA
2	BER

generator_id	coords	power	activity	revs
1	8.4/49.0	4	[2018-01-01, NULL)	[1, NULL)
2	13.4/52.5	6	[2018-01-01, NULL)	[1, 2)
2	13.4/52.5	6	[2018-01-01, 2018-05-01)	[2, NULL)
2	13.4/52.5	12	[2018-05-01, NULL)	[2, NULL)

revision_id	activity	desc
0	(NULL, 2018-01-01)	In the beginning, the Earth was without form, and void..
1	[2018-01-01, 2018-06-29)	Add KA, BER
2	[2018-06-29, NULL)	Double BER power

Writing into DB

- table revision
 - close previous activity
 - insert a new one
- other tables
 - close previous revs
 - insert new rows

Writing into DB

- table revision
 - close previous activity
 - insert a new one
- other tables
 - close previous revs
 - insert new rows

```
with Revision(desc='update from #123') as rev:  
  rev.modify(...)  
  rev.modify(...)
```

```

class Revision:
    def __init__(self, desc):
        self.desc = desc
        self.revision_id = None
        self.todos = []
        self.conn = psycopg.connect('postgres://USER@HOST:PORT/DBNAME')

    def __enter__(self):
        with self.conn.cursor() as cur:
            cur.execute("""UPDATE revision SET activity = TSTZRANGE(LOWER(activity), NOW())
                           WHERE UPPER_INF(activity) RETURNING revision_id""")
            self.revision_id = next(cur)[0] + 1 # raise StopIteration?
            cur.execute("""INSERT INTO revision (revision_id, desc) VALUES (%s, %s)""",
                       (self.revision_id, self.desc))

        return self

    def __exit__(self, exc_type, exc_value, exc_traceback):
        if exc_type is None and self.todos:
            while True:
                confirm = input(f'{len(self.todos)} todos. commit? (yes/no)')
                if confirm == 'yes':
                    self.conn.commit()
                    return
                elif confirm == 'no':
                    break
            self.conn.rollback()

    def modify(self, ...):
        ...

```

Reading from DB

- consistent at any time

```
SELECT MAX(revision_id)
FROM revision;
```

```
SELECT ...
FROM generator
WHERE revs @> %(revision_id)s
AND ...
```

Reading from DB

- consistent at any time

```
SELECT MAX(revision_id)
FROM revision;
```

```
SELECT ...
FROM generator
WHERE revs @> %(revision_id)s
AND ...
```

- direct SQL or service?

Reading from DB

- consistent at any time

```
SELECT MAX(revision_id)
FROM revision;
```

```
SELECT ...
FROM generator
WHERE revs @> %(revision_id)s
AND ...
```

- direct SQL or service?
 - logging, usage statistics (upgrades?)
 - caching
 - updates in service?

Consistent serial reading

What is “now”?

```
today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
yesterday = (datetime.datetime.utcnow() - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

Consistent serial reading

What is “now”?

```
today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
yesterday = (datetime.datetime.utcnow() - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

```
now = datetime.datetime.utcnow()
today = now.strftime('%Y-%m-%d')
yesterday = (now - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

Consistent serial reading

What is “now”?

```
today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
yesterday = (datetime.datetime.utcnow() - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

```
now = datetime.datetime.utcnow()
today = now.strftime('%Y-%m-%d')
yesterday = (now - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

What revision are we looking at?

```
data = service.method_one(*args)
data = service.method_two(*args)
```

Consistent serial reading

What is “now”?

```
today = datetime.datetime.utcnow().strftime('%Y-%m-%d')
yesterday = (datetime.datetime.utcnow() - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

```
now = datetime.datetime.utcnow()
today = now.strftime('%Y-%m-%d')
yesterday = (now - datetime.timedelta(1)).strftime('%Y-%m-%d')
```

What revision are we looking at?

```
data = service.method_one(*args)
data = service.method_two(*args)
```

```
revision_id, data = service.method_one(*args)
revision_id, data = service.method_one(*args, revision_date=datetime(2018, 4, 1, 13, ...))
revision_id, data = service.method_one(*args, revision_id=17)
_, data = service.method_two(*args, revision_id=revision_id)
```

Revisions like commits in VCS?

- single branch
- “test” branches through clones?

Lessons learned

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST
- psycopg2 blocking transactions on modified rows

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST
- psycopg2 blocking transactions on modified rows
- use context managers for “transactions” in code

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST
- psycopg2 blocking transactions on modified rows
- use context managers for “transactions” in code
- read external sources (time, DB, ...) independently from your runtime

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST
- psycopg2 blocking transactions on modified rows
- use context managers for “transactions” in code
- read external sources (time, DB, ...) independently from your runtime
- Python3, psycopg2 and PostgreSQL are cool

Lessons learned

- INT4RANGE, TSTZRANGE, EXCLUDE USING GIST
- psycopg2 blocking transactions on modified rows
- use context managers for “transactions” in code
- read external sources (time, DB, ...) independently from your runtime
- Python3, psycopg2 and PostgreSQL are cool
- come to PyCon.DE in Karlsruhe (24–26 Oct)

Miroslav Šedivý

[ˈmɪrɔslav ˈʃɛɟɪviː]

 eumiro  eumiro  inšedivý