Replication lab

Setup

Tworzymy bazy (komendy w katalogu `/usr/lib/postgresql/12/bin`):

```
./initdb -D ~/pub_db
./initdb -D ~/sub_db
```

Zmieniamy w konfiguracji publishera ustawiając `wal_level` na logical a port na 5433. Zmieniamy port subscribera na 5434.

Uruchamiamy:

```
./pg_ctl -D /home/przemek/pub_db/ -l ~/pub_db/logfile start
./pg_ctl -D /home/przemek/sub_db/ -l ~/sub_db/logfile start
```

Łączymy się:

```
psql -p 5433 -U przemek -d postgres
```

Tworzymy nową baze i łaczymy się:

```
postgres=# create database pub_db;
CREATE DATABASE
postgres=# exit

psql -p 5433 -U przemek -d pub_db
```

Tworzymy tabelę i wypełniamy:

```
pub_db=# create table pub_tbl (id int primary key, name varchar);
CREATE TABLE

pub_db=# insert into pub_tbl (id, name) select generate_series(1,10), 'some
random name';
INSERT 0 10
pub_db=# select * from pub_tbl;
 id |       name
----+------------------
  1 | some random name
  2 | some random name
```

```
  3 | some random name
  4 | some random name
  5 | some random name
  6 | some random name
  7 | some random name
  8 | some random name
  9 | some random name
 10 | some random name
(10 rows)
```

Łaczymy się z instancją subscriber, tworzymy nową bazę `sub_db` i łaczymy się z nią:

```
psql -p 5434 -U przemek -d postgres
postgres=# create database sub_db;
psql -p 5434 -U przemek -d sub_db
```

Przerzucamy schemat danych (-s to schema only, czyli bez danych):

```
./pg_dump -p 5433 -s pub_db | psql -p 5434 -U przemek -d sub_db
```

Towrzymy publishera i subscribera:

```
pub_db=# create publication pub1 for all tables;
CREATE PUBLICATION

sub_db=# create subscription sub1 connection
'postgresql://przemek@localhost:5433/pub_db' publication pub1;
NOTICE:  created replication slot "sub1" on publisher
CREATE SUBSCRIPTION
```

Widzimy dane na subscriberze:

```
sub_db=# select * from pub_tbl;
 id |       name
----+------------------
  1 | some random name
  2 | some random name
  3 | some random name
  4 | some random name
  5 | some random name
  6 | some random name
  7 | some random name
  8 | some random name
  9 | some random name
```

```
 10 | some random name
(10 rows)
```

Testy

Wstawiamy wiećej wierzy na publisherze:

```
pub_db=# insert into pub_tbl (id, name) select generate_series(11,20),
'some other random name';
INSERT 0 10
```

I patrzymy czy się zreplikowały:

```
sub_db=# select * from pub_tbl;
 id |          name
----+------------------------
  1 | some random name
  2 | some random name
  3 | some random name
  4 | some random name
  5 | some random name
  6 | some random name
  7 | some random name
  8 | some random name
  9 | some random name
 10 | some random name
 11 | some other random name
 12 | some other random name
 13 | some other random name
 14 | some other random name
 15 | some other random name
 16 | some other random name
 17 | some other random name
 18 | some other random name
 19 | some other random name
 20 | some other random name
(20 rows)
```

Mamy to!

Update:

```
pub_db=# update pub_tbl set name='Przemek' where id=2;
UPDATE 1

sub_db=# select * from pub_tbl where id=2;
 id |  name
```

```
----+---------
  2 | Przemek
(1 row)
```

Usuwanie:

```
pub_db=# delete from pub_tbl where id > 15;
DELETE 5

sub_db=# select * from pub_tbl where id>10;
 id |          name
----+------------------------
 11 | some other random name
 12 | some other random name
 13 | some other random name
 14 | some other random name
 15 | some other random name
(5 rows)
```

Usuwanie wszystkich danych:

```
pub_db=# truncate table pub_tbl;
TRUNCATE TABLE

sub_db=# select * from pub_tbl;
 id | name
----+------
(0 rows)
```

Dodanie kolumnty:

```
pub_db=# alter table pub_tbl add age int;
ALTER TABLE
pub_db=# insert into pub_tbl (id, name, age) select generate_series(1,10),
'some random name', generate_series(20,29);
INSERT 0 10

sub_db=# select * from pub_tbl;
 id | name
----+------
(0 rows)
```

Musimy zmienić także tabelę subskrybenta:

```
sub_db=# alter table pub_tbl add age int;
ALTER TABLE
sub_db=# select * from pub_tbl;
 id |       name       | age
----+------------------+-----
  1 | some random name |  20
  2 | some random name |  21
  3 | some random name |  22
  4 | some random name |  23
  5 | some random name |  24
  6 | some random name |  25
  7 | some random name |  26
  8 | some random name |  27
  9 | some random name |  28
 10 | some random name |  29
(10 rows)
```

```
pub_db=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 45952
usesysid         | 10
usename          | przemek
application_name | sub1
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 36380
backend_start    | 2024-05-13 16:07:25.563879+02
backend_xmin     |
state            | streaming
sent_lsn         | 0/16C0490
write_lsn        | 0/16C0490
flush_lsn        | 0/16C0490
replay_lsn       | 0/16C0490
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-05-13 16:09:26.097396+02



sub_db=# select * from pg_stat_replication;
(0 rows)
```

Na publisherze mamy dane dotyczące replikacji, na subskriberze nie.

Zatryzmujemy subskrypcje i patrzymy znowu:

```
sub_db=# alter subscription sub1 disable;
ALTER SUBSCRIPTION

pub_db=# select * from pg_stat_replication;
(0 rows)
```

Tym razem brak danych o replikacji i słusznie!

```
sub_db=# alter subscription sub1 enable;
ALTER SUBSCRIPTION
```

Dodatkowe konfiguracje

Utworzymy z naszych instancji zabawną strukturę drzewiastą i w ten sposób obejdziemy oba podpunkty na raz, ha!

sub2 i sub3 będą podpięte pod publishera, a sub4, będzie podpięte pod oryginalny sub (cascade)

```
./initdb -D ~/sub2_db
./initdb -D ~/sub3_db
./initdb -D ~/sub4_db
```

Zmieniamy porty i ustawienia wal_level na sub (on też będzie publishował). Uruchamiamy:

```
./pg_ctl -D /home/przemek/sub_db/ -l ~/sub_db/logfile restart

./pg_ctl -D /home/przemek/sub2_db/ -l ~/sub2_db/logfile start
./pg_ctl -D /home/przemek/sub3_db/ -l ~/sub3_db/logfile start
./pg_ctl -D /home/przemek/sub4_db/ -l ~/sub4_db/logfile start
```

Kopiujemy schemy:

```
./pg_dump -p 5433 -s pub_db | psql -p 5435 -U przemek -d postgres
./pg_dump -p 5433 -s pub_db | psql -p 5436 -U przemek -d postgres
./pg_dump -p 5434 -s sub_db | psql -p 5437 -U przemek -d postgres
```

Tworzymy publishera (w sub) i subscriberów w reszcie.

```
sub_db=# create publication pub2 for all tables;
CREATE PUBLICATION

baza sub2:
```

```
postgres=# create subscription sub2 connection
'postgresql://przemek@localhost:5433/pub_db' publication pub1;
NOTICE:  created replication slot "sub2" on publisher


baza sub3:
postgres=# create subscription sub3 connection
'postgresql://przemek@localhost:5433/pub_db' publication pub1;
NOTICE:  created replication slot "sub3" on publisher
CREATE SUBSCRIPTION

baza sub4:
postgres=# create subscription sub3 connection
'postgresql://przemek@localhost:5434/sub_db' publication pub2;
NOTICE:  created replication slot "sub3" on publisher
CREATE SUBSCRIPTION
```

Kopiuje się:

```
postgres=# select * from pub_tbl;
 id |       name       | age
----+------------------+-----
  1 | some random name |  20
  2 | some random name |  21
  3 | some random name |  22
  4 | some random name |  23
  5 | some random name |  24
  6 | some random name |  25
  7 | some random name |  26
  8 | some random name |  27
  9 | some random name |  28
 10 | some random name |  29
```