PostgreSQL インサイド

論理レプリケーションにおけるパブリケーションの行フィルター PostgreSQL 15 でコミットされた機能の先行紹介

技術者 Blog

Peter Smith
Fujitsu Australia Software Technology
Senior Software Development Engineer
PostgreSQL Contributor

はじめに

通常、論理レプリケーションのパブリッシュとサブスクライブモデルを使用する場合、パブリッシュされたテーブルのすべてのデータ変更が適切なサブスクライバーにレプリケートされます。新しい行フィルター機能を使用すると、指定したフィルター式に一致する行データのみがレプリケートされるよう制限できるようになります。

機能の概要

行フィルターは、CREATE PUBLICATION 構文において、テーブルごとにオプションで指定します。行フィルター「WHERE 句」を、行フィルターを適用するテーブルごとに追加します。各パブリケーションには、0 個以上の行フィルターを持つことができます。

CREATE PUBLICATION 構文の改良

次に、簡略化した CREATE PUBLICATION 構文を示します。PostgreSQL 15 で追加された新しい行フィルターの WHERE 句を強調表示しています。

```
CREATE PUBLICATION <pub-name> FOR TABLE <table-list>
<table-list> ::= <table-list-item> [, <table-list>]
<table-list-item> ::= <table-name> [ WHERE (<row-filter-expression>) ]
```

行フィルター式

行フィルターの WHERE 句では、基本的な関数と論理演算子を含む単純な式のみを使用できます。また、所属するテーブルのカラムのみを参照できます。パブリケーションが UPDATE または DELETE の操作をパブリッシュする場合、行フィルター式には、テーブルに「レプリカアイデンティティ（REPLICA IDENTITY）」が設定されているカラムを指定する必要があります。パブリケーションが INSERT の操作のみをパブリッシュする場合、行フィルター式にはテーブル内のどのカラムでも使用できます。

例 1

```
CREATE PUBLICATION cheap_widgets
FOR TABLE widget WHERE (price < 1.99);
```

例 2

```
CREATE PUBLICATION sales_people
FOR TABLE employees WHERE (role <> 'manager' AND dept = 'sales');
```


例 3

```
CREATE PUBLICATION rate_usa    FOR TABLE exchange_rates WHERE (country = 'us');
CREATE PUBLICATION rate_uk     FOR TABLE exchange_rates WHERE (country = 'uk');
CREATE PUBLICATION rate_france FOR TABLE exchange_rates WHERE (country = 'fr');
```

行フィルターの適用

行フィルターが適用されるタイミングは、データの変更をパブリッシュするかどうかを決定する前です。なお、行フィルター式が NULL または false と評価された場合、その行はレプリケートされません。また、行フィルターにより、データを変更するコマンドは以下のように処理されます。

TRUNCATE TABLE コマンド

行フィルターは無視されます。

INSERT コマンド

INSERT としてレプリケートされます。

DELETE コマンド

DELETE としてレプリケートされます。

UPDATE コマンド

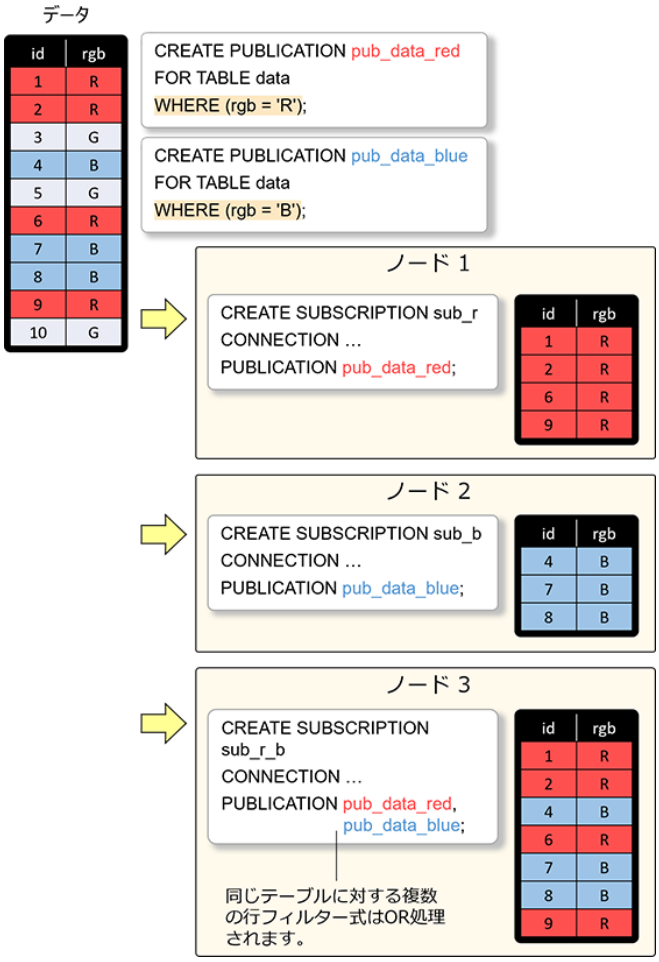
サブスクライバー側でレプリケートされたテーブルデータが、パブリッシャーで定義された行フィルター式に従う必要があるため、より複雑な動きになります。これを実現するために必要なアクションを決定するには、UPDATE が処理されるたびに、「古い」 行データと 「新しい」 行データの両方（つまり、UPDATE の前後）に対して行フィルター式を評価する必要があります。

- 「古い」 行データは行フィルター式を満たしていないが（この行はサブスクライバーに存在しないものとします）、「新しい」 行データは行フィルター式を満たす場合、データの一貫性を保つため、 サブスクライバーに「新しい」 行を追加する必要があります。つまり、UPDATE は INSERT に変換されます（下表の A）。
- 「古い」 行データは行フィルター式を満たすが（この行はすでにサブスクライバーに送信されていると断言します）、「新しい」 行データは行フィルター式を満たさない場合、データの一貫性を保つため、 サブスクライバーから「古い」 行を削除する必要があります。 つまり、UPDATE は DELETE に変換されます（下表の B）。

フィルターの一致有無		レプリケーション方法
古い行	新しい行	
無	無	何もレプリケーションしません。
無	有	サブスクライバーに新しい行を INSERT します。**（A）**
有	無	サブスクライバーの古い行を DELETE します。**（B）**
有	有	サブスクライバーの UPDATE をレプリケートします。

複数の行フィルターの組み合わせ

1 つのサブスクリプションが、複数のパブリケーションによってサブスクライブする場合があります。そのパブリケーションは、同じテーブルが異なる行フィルターの WHERE 句でパブリッシュされたときです。この場合、これらの式は OR 処理され、いずれかの式を満たす行がレプリケートされます。
これは、そのテーブルのサブスクライブされたパブリケーションの 1 つに行フィルターがない（全行が対象になる）場合、そのテーブルの他のすべての行フィルターが無視されることをも意味します。



テーブルの初期同期

ユーザーが CREATE SUBSCRIPTION 構文でデフォルトの「copy_data=true」のパラメーターを選択すると、パブリケーションの行フィルターを満たす既存のデータのみがコピーされます。

psql コマンドの拡張

この新機能の一部として、PostgreSQL 15 の psql コマンドについても行フィルター式に関する有用な情報を提供するように拡張されました。
テーブルを表示するメタコマンド（¥d）では、そのテーブルがメンバーであるパブリケーションの WHERE 句が表示されるようになりました。パブリケーションを一覧表示するメタコマンド（¥dRp+）では、任意のテーブルの WHERE 句が表示されるようになりました。

例 4

```
postgres=# CREATE TABLE data(id int, rgb text);
CREATE TABLE
postgres=# CREATE PUBLICATION pub_data_all FOR TABLE data;
CREATE PUBLICATION
postgres=# CREATE PUBLICATION pub_data_blue FOR TABLE data WHERE (rgb = 'B');
CREATE PUBLICATION
postgres=# CREATE PUBLICATION pub_data_red FOR TABLE data WHERE (rgb = 'R');
CREATE PUBLICATION
postgres=# ¥d data
               Table "public.data"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           |          |
 rgb    | text    |           |          |
Publications:
    "pub_data_all"
    "pub_data_blue" WHERE (rgb = 'B'::text)
    "pub_data_red" WHERE (rgb = 'R'::text)

postgres=# ¥dRp+
                        Publication pub_data_all
  Owner   | All tables | Inserts | Updates | Deletes | Truncates | Via root
----------+------------+---------+---------+---------+-----------+----------
 postgres | f          | t       | t       | t       | t         | f
Tables:
    "public.data"

                       Publication pub_data_blue
  Owner   | All tables | Inserts | Updates | Deletes | Truncates | Via root
----------+------------+---------+---------+---------+-----------+----------
 postgres | f          | t       | t       | t       | t         | f
Tables:
    "public.data" WHERE (rgb = 'B'::text)

                        Publication pub_data_red
  Owner   | All tables | Inserts | Updates | Deletes | Truncates | Via root
----------+------------+---------+---------+---------+-----------+----------
 postgres | f          | t       | t       | t       | t         | f
Tables:
    "public.data" WHERE (rgb = 'R'::text)

postgres=#
```


効果

ユーザーが論理レプリケーションの行フィルター定義を利用することに、いくつかの理由があります。

例 5：パフォーマンスの向上

大規模なデータテーブルの小さなサブセットのみをレプリケートすることで、ネットワークトラフィックを削減させる

```
CREATE PUBLICATION pub_data_2205
FOR TABLE bank_transactions WHERE (year = 2022 AND month = 'May');
```

例 6：不要データの削減

データのノイズ（不要な部分）を低減させたり、不良データを除去したりする

```
CREATE PUBLICATION pub_data_good
FOR TABLE sensor WHERE (value IS NOT NULL AND value > 0);
```

例 7：データの特定

サブスクライバーノードに関連するデータのみを提供する

```
CREATE PUBLICATION pub_data_new_york
FOR TABLE weather WHERE (state = 'NY');
```

例 8：情報漏洩対策

機密情報を非表示（レプリケートしない）にすることで、セキュリティの一形態として動作させる

```
CREATE PUBLICATION pub_data_salary
FOR TABLE employees WHERE (role <> 'manager');
```

今後に向けて

行フィルターの追加は、PostgreSQL 15 の論理レプリケーションの主要な機能です。この機能により、ユーザーのツールボックスにまた 1 つ便利な機能が増え、より洗練され、カスタマイズされた論理レプリケーション・ソリューションを作成できるようになります。今後も、富士通の OSS チームは、PostgreSQL 論理レプリケーションの機能強化と機能追加を支援していきます。PostgreSQL 15 で追加された行フィルターに関連する機能として、「論理レプリケーションの列リスト」があります。これは、私の同僚が今後ブログで取り上げる予定です。

詳細情報

本ブログで解説した「パブリケーション行フィルター」についての詳細は、PostgreSQL 文書または GitHub に投稿したコミット情報をご覧ください。

行フィルターを記述するための情報や例について

- https://www.postgresql.org/docs/15/logical-replication-row-filter.html（PostgreSQL オフィシャルのページへ）

新しい行フィルター「WHERE 句」を記述するための CREATE PUBLICATION について

- https://www.postgresql.org/docs/15/sql-createpublication.html（PostgreSQL オフィシャルのページへ）

「レプリカアイデンティティ」について

- https://www.postgresql.org/docs/current/logical-replication-publication.html（PostgreSQL オフィシャルのページへ）
- https://www.postgresql.org/docs/current/sql-altertable.html#SQL-ALTERTABLE-REPLICA-IDENTITY（PostgreSQL オフィシャルのページへ）

新しい行フィルター機能を追加するための GitHub ソースコードについて

- https://github.com/postgres/postgres/commit/52e4f0cd472d39d07732b99559989ea3b615be78（GitHub.com のページへ）

2022 年 9 月 2 日