

# アプリケーションエンジニアから見た PostgreSQL15 の新機能

2022-10-01 OSC 2022 オンライン広島  
日本 PostgreSQL ユーザー会 中国支部長 高橋 一騎

# 注意事項

- スライドは公開しています。
- 質問は #osc22hi に投稿してもらえれば、後ほど拾う事もできるかと思いますので是非活用してください。

# おしながき

- 自己紹介
- PostgreSQLとは
- PostgreSQL15の新機能
- まとめ

# 自己紹介

- 高橋 一騎 (@ikkitang)
- 岡山在住
- スターフェスティバル株式会社  
TechPdM 兼 アプリケーションエンジニア
- 日本PostgreSQLユーザー会 中国地方支部長



# 自己紹介

- 日本PostgreSQLユーザー会中国地方支部長ではありますが、普段は普通にWebアプリケーションのコードを書いてていわゆるDBA的な仕事はしていません。
  - 割とPostgreSQLやMySQLはAWSのAmazon RDSでシュッと導入して使っています。
- 本セッションでは、今回のPostgreSQL15に入る機能をアプリケーションエンジニアとしての立場からピックアップをした上でご紹介をさせていただければと思います。

## 2. PostgreSQLとは

## 2. PostgreSQLとは

- 代表的なオープンソースのRDBMSの一つ
- もともと、大学の研究用に開発された研究用のRDBMSの`ingress`が元となっている。
- PostgreSQL開発コミュニティによって開発が行われていて、約1年弱の開発期間を経た後、毎年9~10月頃にメジャーバージョンがリリースされている。
- 今年は`PostgreSQL15`のリリースに向けて開発が行われており、2022-10-06にGAの予定が発表された



Noriyoshi Shinoda

@nori\_shinoda · Follow



PostgreSQL 15 : The release team is planning to release PostgreSQL 15 RC1 on 2022-09-29, and they are planning for 2022-10-06 to be the GA date.



postgresql.org

PostgreSQL 15 RC1 + GA dates

10:44 PM · Sep 19, 2022



10



Reply



Copy link

Explore what's happening on Twitter

# PostgreSQLの特徴

## 複数のIndexアルゴリズムをサポート

- B-Tree Index
- Hash Index
- GiST Index, SP-GiST index, GIN Index
- BRIN Index

## 豊富なデータ型をサポート

- 数値型, 文字型, boolean型, 列挙型
- UUID型
- JSON型
- 配列型
- 範囲型
- 幾何データ型
  - 座標点
  - 直線
  - 円
- IPアドレス型



# PostgreSQLのバージョンニング

- 10より前は `x.y.z` の `x.y` の部分がメジャーバージョン
  - 8.4 => 9.1 => 9.2 => 9.3 (年単位でバージョンアップ)
- 10移行は `x.y` の `x` の部分がメジャーバージョン
  - 10.0 => 11.0 => 12.0 (年単位でバージョンアップ)
  - EOLはリリースから5年間と定められているので今ではサポートされているのはすべて `x.y` の形式

今日はそんなPostgreSQL15の話

### 3. PostgreSQL15の新機能

### 3. PostgreSQL15の新機能（抜粋）

- Merge文のサポート
- ロジカルレプリケーションの機能拡張
- パラレルクエリの強化
- バージョン非互換対応（新機能ではないけど）
  - PublicスキーマのCreate権限がデフォルトからなくなる

# Merge文のサポート

# Merge文のサポート

- INSERT・UPDATE・DELETEを一括で処理できる。
- SQL:2003で標準SQLとして定義されていて、OracleやSQL Serverではすでにサポートされている。
- 条件に合致したとき（`WHEN MATCHED` 句）
  - UPDATE
  - DELETE
  - DO NOTHING: 何も処理しない
- 条件に合致しなかったとき（`WHEN NOT MATCHED` 句）
  - INSERT
  - DO NOTHING: 何も処理しない

例

```
MERGE INTO members

  USING (VALUES (1, 'test@example.com', 'test name'))
    AS i(member_id, email, user_name)
  ON members.id = i.member_id

  WHEN MATCHED THEN
    UPDATE SET user_name = i.user_name
  WHEN NOT MATCHED THEN
    INSERT (member_id, email, user_name)
    VALUES (i.member_id, i.email, i.user_name);
```

`WHEN MATCHED` 句では条件を複数記述できる

```
WHEN MATCHED AND hoge.flag = 1 THEN
```

# UPSERT文

- INSERT・UPDATEを組み合わせた操作を行う事からUPSERTと呼ばれる。
- PostgreSQLにはすでに`UPSERT`相当の機能がある
- 以下2つは成功時の結果が同じ

## MERGE文

```
MERGE INTO members
USING (VALUES (1, 'test@example.com', 'test name'))
  AS i(member_id, email, user_name)
ON members.id = i.member_id
WHEN MATCHED THEN
  UPDATE SET user_name = i.user_name
WHEN NOT MATCHED THEN
  INSERT (member_id, email, user_name)
  VALUES (i.member_id, i.email, i.user_name);
```

## INSERT ON CONFLICT句

```
INSERT INTO members (member_id, email, user_name)
VALUES (1, 'test@example.com', 'test name')
ON CONFLICT(member_id)
DO UPDATE SET user_name = 'test name';
```

# UPSERT文の違い

項目	MERGE	INSERT ON CONFLICT
処理速度	簡易比較ではこちらの方が10~100%程高速	
実装方法	targetとsourceをJOINし、MATCHED句に応じて処理 JOINの結果で予め処理を決めてから実行するのでJOIN時と実データに差があればエラー出ることもある	INSERTして、制約に違反したらUPDATEする 処理をした結果に応じてUPDATEできるので並行性能が高い
対応範囲	DELETEに対応, 条件の比較に等号・不等号を扱える	DELETE未対応, 条件の比較は等号により比較
実行の注意事項		ON CONFLICT に指定したカラムに必ずユニーク制約が必要

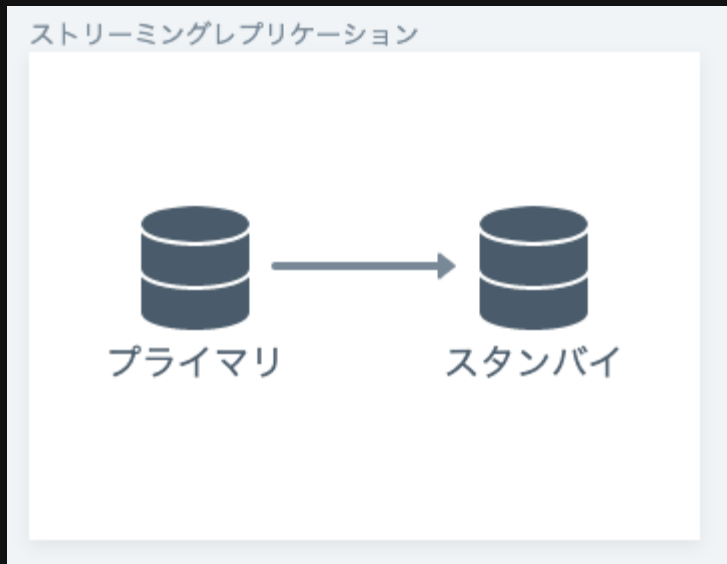


ロジカル（論理）レプリケーションの機能拡張

# PostgreSQLのレプリケーションについておさらい

## ストリーミングレプリケーション

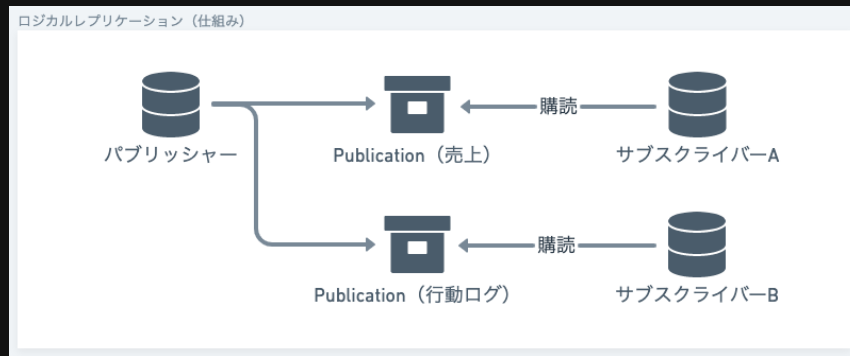
- データベースクラスタ全体を  
WAL転送によってレプリケーションする
- レプリケーション元をプライマリ
- レプリケーション先をスタンバイ
- スタンバイは参照のみ
- 主な利用用途は  
リードレプリカ, フェイルオーバー



# PostgreSQLのレプリケーションについておさらい

## ロジカルレプリケーション

- テーブルやデータベース単位でWALを操作の情報に変換した情報をレプリケーションする
- レプリケーション元をパブリッシャー
- レプリケーション元をサブスクライバーといい、Publicationを購読する仕組み
- サブスクライバーに書き込みをしても良い  
(完全同期の必要がない)
- OSやメジャーバージョンが異なってもレプリケーションできる
- 主な利用用途は  
分析目的のデータベースを作る, バージョンアップ



# ロジカル（論理）レプリケーションの機能拡張

## 行フィルタ機能

- PUBLICATION定義にWHERE句を指定して条件を満たす行だけのPublisherが作成できるようになった

```
CREATE PUBLICATION chugoku_members
  FOR TABLE app.members WHERE address IN ('広島','岡山','島根','鳥取','山口');
```

## 列フィルタ機能

- PUBLICATION定義でカラムを絞る事で特定のカラムだけのPublisherが作成できるようになった

```
CREATE PUBLICATION member_emails
  FOR TABLE app.members (email);
```

## スキーマ対象でテーブルを一括指定

```
CREATE PUBLICATION app_schemas
  FOR ALL TABLES IN SCHEMA app;
```

# ロジカル（論理）レプリケーションの機能拡張列挙

- サブスクリプション側でエラーになった時に自動で購読を止めるオプション追加
- サブスクリプション側でLSN（WALログID）を指定して処理をスキップする事ができる機能追加

これまで、煩雑な処理の後に調整していたり、ずっとエラーログが出続けるような問題についてサポートが入った



# パラレルクエリーの性能向上

- `SELECT DISTINCT` 文でパラレルクエリが有効化！
  - 私個人的に好きな機能の一つです

# その他機能

- 正規表現関数の追加
- ログをJSON形式で出力できるように
- NOT IN句のアルゴリズムを改善して高速化
- ソートアルゴリズムが改善
- ウィンドウ関数 ( row\_number, rank, count ) の性能改善
- JSON関数の追加 ... は残念ながら見送られた



バージョン非互換の変更

データベース作成時のデフォルト権限の変更

# データベース作成時のデフォルト権限の変更 元々

- `public` スキーマに誰でもテーブルなどのオブジェクトを作成可能だった
- これによって、脆弱性が発生したり、攻撃の対象となっていた（CVE-2018-1058 とか）

## PostgreSQL15から

- データベース所有者かスーパーユーザ以外がオブジェクトを作成する権限がデフォルトから除外された
  - 上記の脆弱性の危険性と今後も付き合っていく必要があるが、デフォルトから除外なので追加すれば元に戻す事はできる
- `public` スキーマの所有者が `pg_database_owner` に変更された

データベース作成時のデフォルト権限の変更

## バージョンアップ時に想定される影響

- PostgreSQL14でデータベースの中にあるpublicスキーマを運用している場合に PostgreSQL15にバージョンアップした時にどのような影響が発生するか.
- バージョンアップ方法が2種類あるので、それぞれ比較

## dump & restore によるバージョンアップ

- `pg_dump` で旧バージョンからダンプして `pg_restore` で新バージョンにリストアを行う方式
- 新しい空のデータベースを作った上でリストアするので、publicスキーマの書き込み権限は無く publicスキーマを使用している場合は影響を受ける

## pg\_upgrade

- すべてが以前のバージョンと同じようにコピーされる
- 旧バージョン時点でpublicスキーマへの書き込み権限があれば、そのまま継続利用できる

データベース作成時のデフォルト権限の変更

## AWSでアップグレードをする時の挙動

- RDS for PostgreSQLではバージョンアップをコンソールから実行できる
- そこでは `pg_upgrade` を利用してバージョンアップが行われているので、旧バージョンのまま利用できる
- ちなみに...
  - GCP (Cloud SQL for PostgreSQL) もコンソールでバージョンアップでき、`pg_upgrade` が使われてる
  - Azureは、コンソールでメジャーバージョンアップには対応してないので、`dump & restore` 方式を使う必要がある
- 旧バージョンのまま利用できる とはいいつつも **publicスキーマの書き込み権限が脆弱性の起因となる** 事についてはちゃんと考慮の上、運用していきましょう。

# 最後に...

- PostgreSQL15では、他RDBで使えてる機能を積極サポートしながらも、PostgreSQL自体の強み機能も改善していくようなバージョンアップと感じています。
  - Cloud移行に伴って商用DBからPostgreSQLに乗り換えて頂く事も多い事から他RDBで使えている機能の積極サポートはありがたいそう。
  - Docker公式イメージでPostgreSQL15 beta版を利用する事もできるので興味があれば試してみてください！
- PostgreSQL15のGAリリースは 2022-10-06です！！ お楽しみに！！

# 参考文献

- PostgreSQL15 検証レポート SRAOSS
  - [https://www.sraoss.co.jp/tech-blog/wp-content/uploads/2022/08/pg15\\_report\\_20220906.pdf](https://www.sraoss.co.jp/tech-blog/wp-content/uploads/2022/08/pg15_report_20220906.pdf)
- ロジカルレプリケーションの紹介 (1) SRA OSS Tech Blog
  - <https://www.sraoss.co.jp/tech-blog/pgsql/logical-replication-1/>
- Changes to the public schema in PostgreSQL 15 and how to handle upgrades
  - <https://andreas.scherbaum.la/blog/archives/1120-Changes-to-the-public-schema-in-PostgreSQL-15-and-how-to-handle-upgrades.html>
- PostgreSQL14文書
  - <https://www.postgresql.jp/document/14/html/index.html>
- PostgreSQL15Document
  - <https://www.postgresql.org/docs/15/>

# 参考文献

- PostgreSQL 15 開発最新情報
  - <https://www.slideshare.net/masahikosawada98/postgresql-15>
- PostgreSQL 15 最新情報解説
  - [https://www.sraoss.co.jp/wp-content/uploads/files/event\\_seminar/material/2022/PG15\\_sraoss\\_techwebinar\\_20220902.pdf](https://www.sraoss.co.jp/wp-content/uploads/files/event_seminar/material/2022/PG15_sraoss_techwebinar_20220902.pdf)
- PostgreSQL の INSERT ON CONFLICT と MERGE の簡易性能比較
  - [https://qiita.com/fujii\\_masao/items/462bac9f6a107d6134c4](https://qiita.com/fujii_masao/items/462bac9f6a107d6134c4)
- PostgreSQL 15 新機能検証結果 (Beta 1)
  - [https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/lcc/pdf/PostgreSQL\\_15\\_Beta\\_1\\_1.pdf](https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/lcc/pdf/PostgreSQL_15_Beta_1_1.pdf)
- Wikipedia MERGE (SQL)
  - [https://ja.wikipedia.org/wiki/MERGE\\_\(SQL\)](https://ja.wikipedia.org/wiki/MERGE_(SQL))

ご清聴ありがとうございました