

PostgreSQL & リレーショナルデータベース 超入門

2025-02-22 11:00-11:45

203 号室

オープンソースカンファレンス 2025 Tokyo Spring

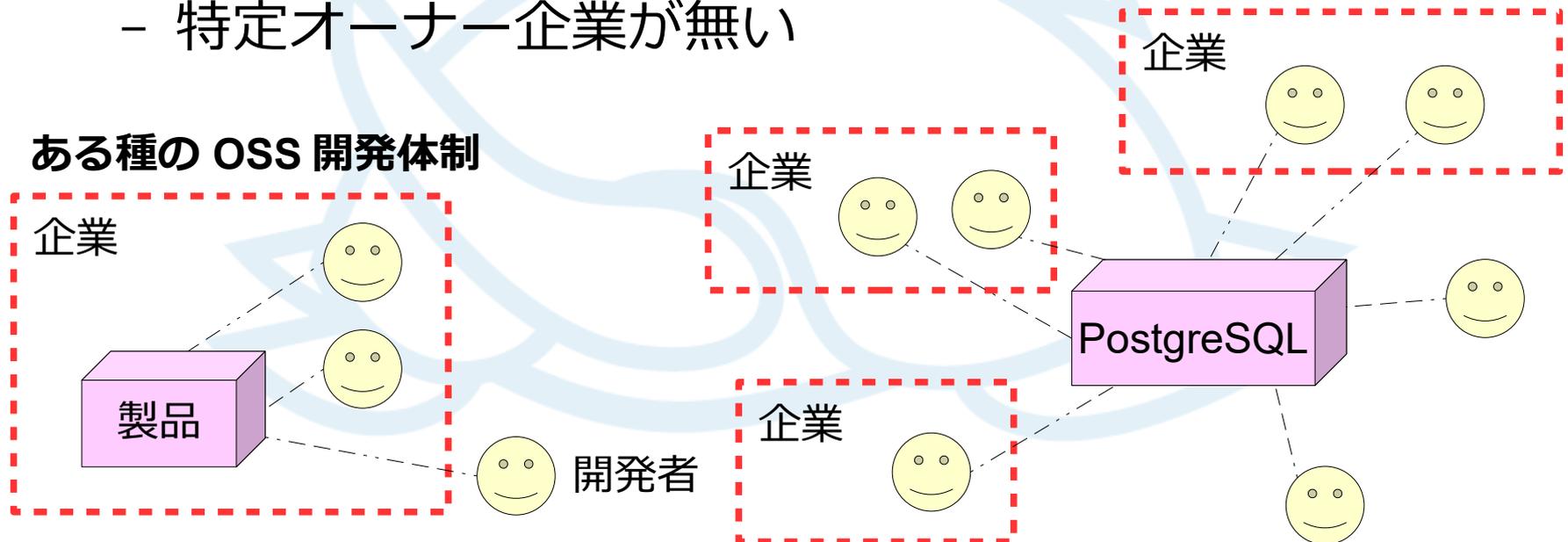
日本 PostgreSQL ユーザ会 (JPUG) 理事 高塚 遥

はじめに

- 「PostgreSQL 近況」と「入門者むけ」の講演を交互にやっています - 今回は入門者むけ
- 本講演の（わたしたちの）目的：
「リレーショナルデータベースも SQL も初めての人の PostgreSQL を使ってもらおう」

PostgreSQL とは

- 多機能、高性能、かつオープンソースのリレーショナルデータベース管理システム ソフトウェア
 - INGRES('70), POSTGRES('80) 由来の長い歴史
 - BSD タイプのライセンス
 - 特定オーナー企業が無い



日本 PostgreSQL ユーザ会 (JPUG) とは

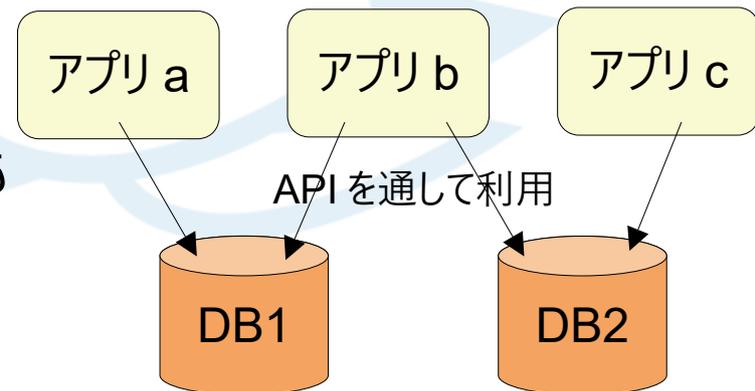
- PostgreSQL の普及を促進し、技術情報を公開し、ユーザ相互の情報交換を支援する
 - メーリングリストや Slack スペースの運営
 - 勉強会、カンファレンスの主催や開催支援
 - マニュアル翻訳
 - PostgreSQL グローバル開発グループ (PGDG) の日本における advocacy 代表窓口

データベース / DBMS とは

- データベース（DB）：
 - 管理されたデータの山
 - 通常は DBMS で管理されている、ので、DBMS を意味することも
- データベース管理システム（DBMS）：
 - データベースを管理するシステム
 - それを実現しているソフトウェアやサービスを指す
 - PostgreSQL, MySQL, Cassandra, MongoDB, ...
 - DBMS が提供するもの：
 - データの追加・削除・変更、検索ができる
 - アプリケーションから独立している
 - 耐障害性の実現、同時実行制御

DBMS が提供するもの (1)

- データの追加・削除・変更、検索
 - データモデルと検索方式は DBMS 種別によって異なる
 - key⇒value (value には 文字列, JSON, XML, など)
 - 文章 (→ 全文検索エンジン)
 - グラフ (→ 知識検索エンジン、「何と何は何関係」)
 - ディレクトリ (→ 階層構造、DNS や Active Directory)
 - 表 (→ リレーショナルデータベース)
- アプリケーションから独立
 - 1つのアプリ配下とは限らない
 - アプリ更改でも DB は残ることも
 - 下位レイヤを委譲



DBMS が提供するもの (2)

- 耐障害性
 - 途中で失敗したら？（アプリクラッシュ，N/W 障害，...）
 - サーバが急にダウンしたら？（停電，物理故障，...）
 - 確定（コミット）済の書き込み内容を保証する
 - さもなくば、やりかけた処理を正常にキャンセルする
 - 同時実行制御
 - 同時に {同じ／関連する} データを更新したら？
 - 更新している途中のデータを参照したら？
 - 一貫性のある動作を保証する
- ★何でも保証するわけではなく、保証できる範囲が定義されていて、その範囲において正しく動いてくれる。

リレーショナルデータベース (RDB) とは

- 複数の表 (テーブル) でデータを扱う DBMS
 - 実用システムで最もよく使われている
 - 高い汎用性 (→ 他のデータモデルを包含できている)
 - SQL 言語による強力な問い合わせ
 - 活用法について理論・ノウハウが蓄積されている

社員表

社員番号	氏名	アカウント名	在籍
1011	喜田 紘介	kida	true
1012	高塚 遥	harukat	false
1013	片岡 裕生	kataoka	true

社員イベント表

イベントID	社員番号	イベント種別	年月日
10001	1011	入社	2017-06-01
10002	1012	入社	2018-04-01
10003	1013	入社	2018-09-01
10004	1012	退社	2020-03-31

★定義域 (ドメイン) 群の直積の有限部分集合をリレーションと呼ぶ、といった話からリレーショナルデータベースと呼ばれているが、そこは気にしなくても良い

SQL とは

- RDB に対する問い合わせ言語
 - データ分析用インタフェースとしても人気
- 在籍者の社員番号と氏名を社員番号昇順に並べた表をください

```
SELECT 社員番号, 氏名 FROM 社員表  
WHERE 在籍 = true  
ORDER BY 社員番号 ASC;
```

社員表

社員番号	氏名	アカウント名	在籍
1011	喜田 紘介	kida	true
1012	高塚 遥	harukat	false
1013	片岡 裕生	kataoka	true

《問い合わせ結果》

社員番号	氏名
1011	喜田 紘介
1013	片岡 裕生

SQL - テーブル結合

- アカウント名が harukat である社員が退社したのはいつ？

```
SELECT 年月日 FROM 社員イベント表
JOIN 社員表 ON 社員表.社員番号 = 社員イベント表.社員番号
WHERE アカウント名 = 'harukat' AND イベント種別 = '退社' ;
```

《問い合わせ結果》

年月日
2020-03-31

本例は
「内部結合」

《結合結果》

イベント ID	社員番号	アカウント名	イベント種別	年月日
10001	1011	kida	入社	2017-06-01
10002	1012	harukat	入社	2018-04-01
10003	1013	kataoka	入社	2018-09-01
10004	1012	harukat	退社	2020-03-31

SQL - 集約問い合わせ

- 在籍社員数は？

```
SELECT count(*) FROM 社員表 WHERE 在籍 = true;
```

社員表

社員番号	氏名	アカウント名	在籍
1011	喜田 紘介	kida	true
1012	高塚 遥	harukat	false
1013	片岡 裕生	kataoka	true

《問い合わせ結果》

count
2

SQL - 集約問い合わせ

- 年ごとの社員イベント件数を表にして

```
SELECT extract(year FROM 年月日) , count(*)  
FROM 社員イベント表 GROUP BY extract(year FROM 年月日);
```

社員イベント表

イベントID	社員番号	イベント種別	年月日
10001	1011	入社	2017-06-01
10002	1012	入社	2018-04-01
10003	1013	入社	2018-09-01
10004	1012	退社	2020-03-31

《問い合わせ結果》

extract	count
2017	1
2018	2
2020	1

SQL - データ分析

- 2024 年の売上を何時台に売れたか、何月に売れたか、で集計

```
SELECT 時, 月, sum(売上) FROM 売上表 WHERE 年 = 2024  
GROUP BY ROLLUP (時, 月)  
ORDER BY 時, 月;
```

売上表

売上ID	売上	店舗	年	月	日	時	..
10001	1,800	A店	2024	1	1	1	..
10002	620	B店	2024	1	1	1	..
10003	11,000	A店	2024	1	1	2	..
10004	1,055	C店	2024	1	1	2	..
:	:	:	:	:	:	:	

《問い合わせ結果》

時	月	sum
1	1	367,010
1	2	169,620
1	3	267,235
:	:	:
1	null	7,167,325
:	:	:
null	null	294,718,805

SQL - テーブル定義、データ操作

- テーブル定義

```
CREATE TABLE 社員表 (  
  社員番号      integer,  
  氏名          varchar(30),  
  アカウソク名  varchar(30),  
  在籍          boolean,  
  PRIMARY KEY (社員番号)  
);
```

SQLは
データ操作命令
については
そっけない

- 追加、更新、削除

```
INSERT INTO 社員表  
VALUES  
(1011, '喜田 紘介', 'kida', NULL);
```

多くのRDBは
他に非標準の
データロード機能を持つ

```
UPDATE 社員表  
SET   在籍 = true  
WHERE 社員番号 = 1011;
```

```
DELETE FROM 社員表  
WHERE 社員番号 = 1011;
```

SQL - トランザクション操作

- 複数の処理を1つトランザクションとして実行

```
BEGIN;  
INSERT INTO 社員表 VALUES  
  (1012, '高塚 遥', 'harukat', NULL);  
UPDATE 社員表 SET 在籍 = true  
  WHERE 社員番号 = 1012;  
COMMIT;
```

BEGIN 相当が
暗黙に発効される
RDBもある

COMMIT が
成功して、
変更が確定する

- トランザクションをロールバック (=取り消し)

```
BEGIN;  
DELETE FROM 社員表  
  WHERE 社員番号 = 1012;  
ROLLBACK;
```

途中で障害が
起きた場合も
ロールバック

SQL - その他の機能

ほかにも、
多数の機能が
あります

- インデックス
 - テーブルの列に対して作る
 - その列について条件を付けた問い合わせを高速化
- 様々な列データ型
 - 整数、浮動小数点、文字列、日付時刻、 boolean
 - 配列、 JSON 、 XML 、 地理・幾何データ
- 再帰問い合わせ
 - グラフデータベースでの問い合わせと等価 - ただし構文は煩雑
- ビュー、サブクエリ
 - 問い合わせ結果を問い合わせ先の表として扱う

RDB/SQL 界における PostgreSQL

- OSS 実装の 2 巨頭の 1 つ（他方は MySQL 系）
- 「PostgreSQL 互換」の商用製品・サービスが多数
 - EDB Postgres, Fujitsu Enterprise Postgres, YugabyteDB, Aurora, Redshift, AlloyDB, Spanner
 - 「互換」の程度はそれぞれ異なる
- SQL 標準への準拠度が高く、癖が強くない
 - 初学者の学習用に適している
- 拡張性が高い
 - 1000 を超える拡張モジュール

PostgreSQL を触ってみる方法は？

- POSTGRES PALYGROUND (ブラウザ上で)
 - クラウドサービス
 - ローカルマシンにコンテナイメージでインストール
 - ローカルマシンに PostgreSQL をインストール
-
- クライアントに何をを使う？
 - 標準の psql コマンドコンソール
 - GUI クライアントツール
 - PgAdmin4 (PostgreSQL 専用、OSS)
 - RDB 汎用の各種 SQL クライアント

POSTGRES PALYGROUND

Open source license info

```
Target width is unset.  
Expanded display is used automatically.  
postgres=# CREATE TABLE t1 (id int primary key, txt varchar(10));  
CREATE TABLE  
postgres=# INSERT INTO t1 VALUES (1, 'AAA');  
INSERT 0 1  
postgres=# INSERT INTO t1 VALUES (2, 'BBB');  
INSERT 0 1  
postgres=# SELECT * FROM t1;  
 id | txt  
----+----  
  1 | AAA  
  2 | BBB  
(2 rows)  
  
postgres=#
```

登録も何もしずに
自由に遊べる

チュートリアル
文書が
用意されている
(英語)

似たサービスとして以下も：
<https://wasm.supabase.com/>

クラウドサービスで

お金はかかる

- クラウドサービスで DB インスタンス作成
 - (AWS) RDS for PostgreSQL
 - Azure Database for PostgreSQL
 - (GCP) Cloud SQL for PostgreSQL
 - EDB BigAnimal など
- クライアントをローカルに導入 (PgAdmin4 など)
 - Linux であれば postgresql パッケージから psql 導入が簡単
- クラウドサービスの文書を参考に DB 接続する
 - 多くのクラウドサービスで DB 接続方法の文書は親切丁寧
 - たいていは、ホスト/ユーザ名/パスワードを指定して接続可
 - しばしば「接続元 IP アドレスを許可」も必要

PgAdmin4 から
クラウド DB インスタンス
を作ることできる

pgAdmin4

The screenshot displays the pgAdmin 4 web interface. On the left, the Object Explorer shows a tree view of the database structure, with the 'db1' database selected. The main panel shows a query editor with the following SQL query:

```
1 SELECT * FROM 社員表;
```

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table format. The table has four columns: '社員番号' (Employee ID), '氏名' (Name), 'アカウント名' (Account Name), and '在籍' (Status). The results are as follows:

	社員番号 [PK] integer	氏名 text	アカウント名 text	在籍 boolean
1	1011	喜田 紘介	kida	true
2	1012	高塚 遥	harukat	false
3	1013	片岡 裕生	kataoka	true

At the bottom of the interface, the status bar indicates 'Total rows: 3 of 3', 'Query complete 00:00:00.076', and 'Ln 1, Col 19'.

ローカルインストール

- Linux
 - RHEL 系、Debian 系とも、公式のリポジトリ/パッケージ有
 - OS 付属パッケージも有
 - インストール後、データベースインスタンス作成とサービス起動
- Windows
 - バイナリ配布あり（インストーラまたは zip セット）
 - インストーラなら、データベースインスタンス作成とサービス登録もしてくれる
 - sc や「サービス」で起動

```
# su - postgres
$ initdb -D /var/lib/pgsql/16/data \
        -U postgres \
        --locale=C --encoding=UTF8
$ pg_ctl start -D /var/lib/pgsql/16/data
```

ローカルインストール

サーバログを
log/ 以下の
ファイルに溜める

- 設定ファイル
 - postgresql.conf
 - pg_hba.conf

```
logging_collector = on
```

#TYPE	DB	USER	ADDRESS	METHOD
local	all	all		trust
host	all	all	127.0.0.1/32	trust
host	all	all	:::1/128	trust

- psql で接続

```
# su - postgres
$ psql -U postgres -d postgres -h localhost
psql (16.4)
Type "help" for help.

postgres=#
```

ローカル接続には
認証を要求しない
(学習環境用)

\d テーブル一覧
\d テーブル名 定義表示
\q 終了

アプリケーションから使う

- 各開発環境ごとにクライアントライブラリがある

libpq	標準付属のクライアントライブラリ (各種言語むけにラッパーあり)
pgJDBC	JDBCドライバ (Java)
Npgsql	.NET データプロバイダ
psqlodbc	ODBCドライバ (C/C++)

pq、go-pg	GO 言語むけクライアントライブラリ
DBD::Pg	Perl 言語むけクライアントライブラリ
Psycopg	Python 言語むけクライアントライブラリ
ruby-pg	Ruby 言語むけクライアントライブラリ
rust-postgres	Rust 言語むけクライアントライブラリ

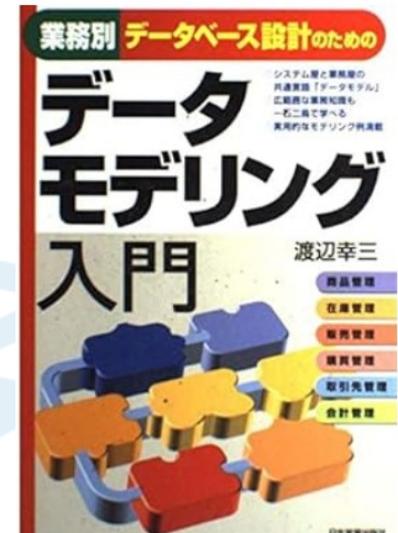
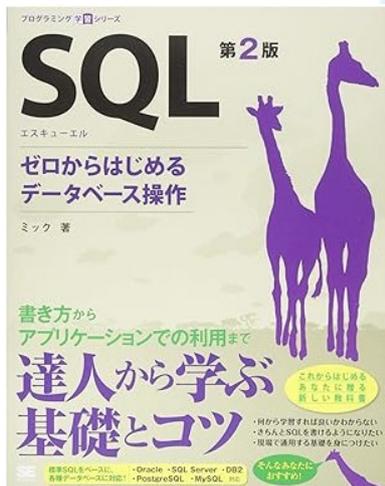
- 今日では SQL 発行のための API を意識しないことも：
 - OR マッパー 経由
 - テーブル定義と、手動または自動での SQL 作成
 - ローコード開発ツールで利用
 - テーブル定義だけ頑張れば、あとは開発ツールで SQL 生成

さらに学ぶためには

- SQL の教本は良書が沢山ある / テーブル設計の良書も
 - 近著なら大抵 PostgreSQL に対応している
- PostgreSQL 入門書は「最新版対応」が少ない
 - が、v12 以降対応ならそれほど不都合はない
- 動画コンテンツは色々ある
 - 有償教材 - Udemy PostgreSQL 入門など、多くは廉価
 - 無料動画 - OSS-DB 教材など、多数あり
- 日本語マニュアル
 - 完成度の高い日本語訳を JPUG 活動で継続作成

参考文献

- 本資料作成のための参考文献：
 - 増永 良文「データベース入門 [第 2 版]」 (サイエンス社)
 - PGDG / JPUG 訳「PostgreSQL 16 文書」
- さらに SQL を学ぶための入門書籍：



書籍紹介

- 「SQL 第2版」ミック (翔泳社) 2016
- 「オープンソースデータベース標準教科書 - PostgreSQL - Ver3.0」
LPI-JAPAN (エルピーアイジャパン) 2024
- 「楽々 ERD レッスン」羽生 章洋 (翔泳社) 2006
- 「データモデリング入門」渡辺 幸三 (日本実業出版社) 2001
- 「これからはじめる PostgreSQL 入門」高塚 遥 (技術評論社) 2018

