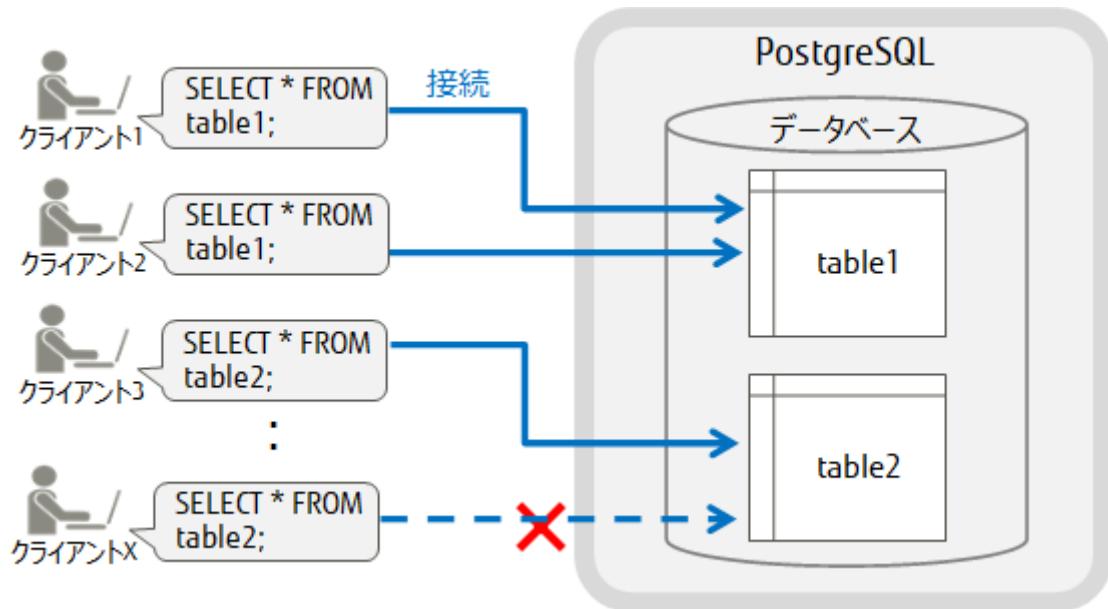


# データベースへの接続数が超過するエラーを 解決したい 技術を知る

- |                                   |                             |                                |  |   |
|-----------------------------------|-----------------------------|--------------------------------|--|---|
| <input type="checkbox"/> 導入／環境設定  | <input type="checkbox"/> 移行 | <input type="checkbox"/> 性能    | <input checked="" type="checkbox"/> チューニング | <input type="checkbox"/> バックアップ／リカバリー   |
| <input type="checkbox"/> 冗長化／負荷分散 | <input type="checkbox"/> 監視 | <input type="checkbox"/> データ連携 | <input type="checkbox"/> 災害対策              | <input checked="" type="checkbox"/> 豆知識 |

## 実現方法

業務運用中のデータベースに起因するトラブルの中で多いものとして、クライアントからデータベースサーバへの接続数の超過があります。クライアントからの接続要求数が、データベースサーバ側で設定した最大接続数を超えると、接続エラーが発生します。



その際、以下のようなエラーメッセージが出力され、データベースサーバへ接続できません。

```
FATAL: sorry, too many clients already
```

```
FATAL: remaining connection slots are reserved for non-replication superuser connections
```

```
FATAL: too many connections for database "xxxx"
```

日本語環境では以下のようなエラーメッセージが出力され、データベースサーバへ接続できません。

```
FATAL: 現在クライアント数が多すぎます
```

```
FATAL: 残りの接続スロットはレプリケーションユーザーではないスーパーユーザー用に予約されています
```

```
FATAL: データベース"xxxx"への接続が多すぎます
```

このような場合、データベースサーバへの最大接続数を増やすなどの対処を行う必要があります。なお、この記事は、PostgreSQL 14.0 をベースに作成しています。

## 対処例

PostgreSQL のデータベースシステムでは、クライアントからデータベースへの最大接続数は、設定ファイル `postgresql.conf` の `max_connections` パラメーターで設定します。このパラメーターは、単独で値を決めるのではなく、関連するパラメーター値と調整して決めることをお勧めします。

### `max_connections`

PostgreSQL に同時に接続できるクライアントの最大数です。デフォルト値は、100 です。値を大きくするとメモリー使用量が増え、性能に影響する可能性があります。

```
max_connections = インスタンスへの最大同時接続数 + superuser_reserved_connections
```

### `superuser_reserved_connections`

PostgreSQL データベースの保守のために予約しておく、スーパーユーザーによる接続数です。デフォルト値は 3 です。クライアントからの同時接続数が `max_connections` から `superuser_reserved_connections` を差し引いた数以上のときは、新規接続はスーパーユーザーのみが許可され、新たなレプリケーション接続は受け入れられません。この値は `max_connections` より小さくなくてはなりません。

## 補足

`max_connections` パラメーターに関連する主なパラメーターについては、「データベースチューニング」を参照してください。

### 最大接続数の確認

`max_connections` パラメーター値を見直すために、現状値の確認方法について説明します。

データベースは `mydb` とします。メタコマンドで拡張テーブル形式モード (`\x`) を有効にし、縦表示とします。

1. **最大接続数 (`max_connections`) の値を確認します。**

```
mydb=# SHOW max_connections;
-[ RECORD 1 ]---+---+
max_connections | 100  <-- デフォルト値の 100
```

2. **`pg_stat_activity` ビューを使用して、現在の同時接続数を確認します。**

```
mydb=# SELECT count(*) FROM pg_stat_activity WHERE backend_type = 'client backend' ;
-[ RECORD 1 ]
count | 92  <-- 同時接続数が最大接続数に近い
```

## 最大接続数の変更

最大接続数の値を 200 に変更する例について説明します。設定ファイル `postgresql.conf` をテキストエディタで編集する方法もありますが、ここでは SQL コマンドを使います。

1. 最大接続数の値を「200」に変更します。

```
mydb=# ALTER SYSTEM SET max_connections = 200 ;      <-- パラメーター値の変更
ALTER SYSTEM
```

2. パラメーター値の変更を反映するためにデータベースシステムを再起動して、値が変更されたことを確認します。

```
$ pg_ctl restart      <-- データベースシステムの再起動
...
$ psql -d mydb
mydb=# SHOW max_connections;
-[ RECORD 1 ]-----+
max_connections | 200  <-- 変更が反映されました
```

## ポイント

最大接続数の課題を解決する上で、参考になる情報を示します。

### コネクションの状態を確認して切断する

データベースへの接続数が超過する原因の 1 つに、アプリケーションが異常終了するなどして、長時間待ち状態にあるコネクションが残存しているケースがあります。なお、この状態は、メモリーや CPU などの資源が完全に解放されないため、パフォーマンスに悪影響を及ぼす弊害もあります。このようなケースにおいては、不要なコネクションを強制的に切断して、資源を回収します。

1. `pg_stat_activity` ビューを使用して、以下の検索条件に該当するコネクションを表示します。

- A) バックエンドの種別 (backend\_type) : client backend
- B) プロセスの状態 (state) : idle in transaction(トランザクション中でアイドル状態)
- C) 問合せが開始した時刻 (query\_start) : 実行に 60 分以上経過した問合せ (現在時刻から算出)

```
mydb=# SELECT * FROM pg_stat_activity WHERE backend_type = 'client backend' and state='idle in transaction'
and current_timestamp > cast(query_start + interval '60 minutes' as timestamp);
-[ RECORD 1 ]-----+
datid      | 16384
datname    | mydb
pid        | 39122          <-- プロセス ID
leader_pid | 
usesysid   | 10
username   | fsepuser
application_name | apl01
client_addr | 192.33.44.15
client_hostname | 
client_port | 27500
backend_start | 2022-04-25 17:52:11.043695+09
xact_start   | 2022-04-25 17:54:13.270685+09
query_start  | 2022-04-25 17:54:13.270685+09  <-- 検索条件 C
```

state_change	2022-04-25 17:54:13.270794+09	
wait_event_type	Client	
wait_event	ClientRead	
state	idle in transaction	--- 検索条件 B
backend_xid		
backend_xmin		
query_id		
query	begin;	
backend_type	client backend	--- 検索条件 A

この条件に該当するコネクションのプロセス ID が「39122」であることが確認できました。

- 不要なコネクションのプロセス ID を指定して、`pg_terminate_backend` 関数を呼び出すことでコネクションを切断します。

```
mydb=# SELECT pg_terminate_backend(39122) ; --- コネクションの切断
```

なお、`pg_terminate_backend` 関数の使用は、対象の権限が付与されているユーザー、または、スーパーユーザーに制限されていますので、注意が必要です。

#### 補足：コネクションの状態確認と切断をまとめて実施する

1回の SQL コマンドで、対象のコネクションをまとめて切断することもできます。

`pg_stat_activity` ビューにおいて、`backend_type` が「client backend」、`state` が「idle in transaction」、そして 60 分以上経過した問合せを持つコネクションをまとめて切断します。

```
mydb=# SELECT pid, usename, application_name, client_addr, pg_terminate_backend(pid)  FROM pg_stat_activity
WHERE backend_type = 'client backend' and state = 'idle in transaction'
and current_timestamp > cast(query_start + interval '60 minutes' as timestamp) ;
-[ RECORD 1 ]-----
pid          | 12345
usename      | fsepuser
application_name | apl01
client_addr   | 192.33.44.15
pg_terminate_backend | t    --- コネクションの切断が正常終了
-[ RECORD 2 ]-----
pid          | 12367
usename      | fsepuser
...
...
```

#### アプリケーションのコネクション切断漏れを確認する

不要なコネクションが残存し、最大接続数を超える原因の 1 つに、アプリケーションにてデータベースへの接続を切断していないケースも考えられます。

必要に応じてアプリケーションのエラー処理などを見直し、不要なコネクションが残らないよう対処してください。

## 参考

---

### PostgreSQL 14 文書

- Documentation (PostgreSQL オフィシャル)  
<https://www.postgresql.org/docs/>
  - Chapter 20. Server Configuration
    - 20.3. Connections and Authentication
      - 20.3.1. Connection Settings
- PostgreSQL 日本語ドキュメント (日本 PostgreSQL ユーザ会)  
<https://www.postgresql.jp/document/>
  - 第 20 章 サーバの設定
    - 20.3. 接続と認証
      - 20.3.1. 接続設定

2023 年 1 月 20 日