



PostgreSQL チートシート

文字列操作

1. 文字列の結合

```
select 'a' || 'b';      a' と 'b' を結合
```

2. 文字数の取得 (全半角の違いなし)

```
select length('文字列');
```

3. 文字列のバイト数を取得 (全角=3バイト計算)

```
select octet_length('文字列');
```

4. 文字列の一部を取得

```
select substring('文字列', 開始位置, 取得文字数);
```

```
select left('文字列', 取得文字数);
```

```
select right('文字列', 取得文字数);
```

5. 検索文字位置の取得

```
select strpos('文字列', '探す文字');
```

6. 前後の半角スペース除去 (全角は除去されない)

```
select * from trim('文字列');
```

7. 大文字・小文字の変換

```
select upper('文字列');
```

```
select lower('文字列');
```

8. 置換

```
select replace('文字列', ' ', 'E');
```

9. 0埋め

```
select lpad(cast(数値 as character varying), 桁数, '0');
```

10. 後ろスペース埋め

```
select rpad('文字列', 桁数, ' ');
```

11. Null の場合の処理

```
select COALESCE(対象文字列, '変換文字列');
```

対象の文字列がnullの場合に変換文字列に変換する

数値操作

1. 数値の書式設定

```
select to_char(数値, 'FM999,999.999');
```

2. 文字数の取得 (全半角の違いなし)

```
select abs(-5);
```

3. 割り算の余りを取得

```
select mod(数値, 割る数値);
```

4. 四捨五入

```
select round(数値, 3);      小数第3位四捨五入
```

```
select trunc(数値, 3);     小数第3位切り捨て
```

5. ランダム値

```
select * from random();
```

日付操作

1. 現在日時取得

```
select current_date;      日付のみ
```

```
select current_time;     時刻のみ
```

```
select now();            日付 + 時刻
```

```
select current_timestamp; 日付 + 時刻
```

2. 日付の書式設定

```
select * from abs(-5);
```

3. 月、週、日、時刻を加算

```
select current_date + cast('5 months' as INTERVAL);
```

```
select current_date + cast('5 weeks' as INTERVAL);
```

```
select current_date + cast('5 days' as INTERVAL);
```

```
select current_date + cast('5 hours' as INTERVAL);
```

```
select current_date + cast('5 minutes' as INTERVAL);
```

```
select current_date + cast('5 seconds' as INTERVAL);
```

4. 曜日を取得する

```
select date_part('dow', current_date);      0: 日-6: 土
```

配列

1. 数値の書式設定

```
select array_length(array['A', 'B', 'C'], 1);
```

2. 配列の要素数分ループ

```
for i in 1..array_length(arr, 1) loop
```

```
end loop;
```

型の変換

1. 文字 → 日付

```
select cast('20190401' as date);
```

2. 文字 → timestamp

```
select cast('20190401 123456' as timestamp);
```

3. 文字 → 数値

```
select cast('20190401' as date);
```



PostgreSQL チートシート

型の変換

4. 数値 ⇒ 文字

```
select cast('20190401 123456' as timestamp);
```

5. 数値 ⇒ 日付

```
select cast('20190401 123456' as timestamp);
```

コメントアウト

1. コメントアウト (1行)

「--」をつける

2. コメントアウト (複数行)

「/*」、「*/」で囲む

データベース・テーブル情報の取得

1. データベース一覧

```
select datname, pg_encoding_to_char(encoding)
from pg_database;
```

2. テーブル一覧

```
select tablename from pg_tables
where schemaname != 'pg_catalog' order by tablename;
```

3. テーブルの列一覧

```
select * from information_schema.columns
where table_name='テーブル名' order by ordinal_position;
```

4. ビュー一覧

```
select * From pg_views where schemaname = 'public';
```

5. シーケンス一覧

```
SELECT relname FROM pg_class JOIN pg_namespace ON
(pg_class.relnamespace = pg_namespace.oid)
WHERE relkind = 'S';
```

6. シーケンスの初期化

```
select SETVAL('seq_name', 1 , false);
```

テーブル・列の変更・ビューの作成

1. 列の追加・変更・削除

列の追加

```
ALTER TABLE テーブル名 add 列 型;
```

Not Null制約の追加

```
ALTER TABLE テーブル名 ALTER COLUMN 列 SET NOT NULL;
```

デフォルト値の追加

```
ALTER TABLE テーブル名 ALTER COLUMN 列名 SET
DEFAULT デフォルト値;
```

列の削除

```
ALTER TABLE テーブル名 DROP COLUMN 列名;
```

インデックスの作成

```
CREATE INDEX インデックス名 ON テーブル名 (列1,列2,...);
```

インデックスの削除

```
DROP INDEX インデックス名;
```

2. データベース・テーブル・列にコメントを付加

```
COMMENT ON DATABASE データベース名 IS 'コメント';
```

```
COMMENT ON TABLE テーブル名 IS 'コメント';
COMMENT ON COLUMN テーブル名.列名 IS 'コメント';
COMMENT ON COLUMN テーブル名.列名 IS NULL;
```

関数

1. 変数の型

character(文字) , character varying(文字) ,
integer(数値) , numeric(数値) , boolean(論理値) ,
date(日付) , timestamp(日付時刻) , record(レコード)

2. 定数

変数名 constant character varying(10) := 'abc';
変数名 constant numeric := 1.08;

3. イコール、ノットイコール、大なり小なり

a = b , a != c ,
a > b , a < c , a >= b , a <= c

4. IF文

```
if 条件式 then 値
elseif 条件式 then 値
else 値
end if;
```

5. For文

```
for i in 1..10 loop
  IF i > 5 THEN EXIT;END IF; EXITでループを抜ける
end loop;
```



PostgreSQL チートシート

関数

6. While

```
while 条件文 loop
    --条件文がtrueの時に処理を繰り返す
end loop;
```

7. SELECT結果を変数へ代入

```
select 列名 into 変数 from テーブル名;
select * into 変数 from テーブル名;
```

8. 変数の中身を表示する

```
raise info '%', 変数;
```

コマンド

1. PostgreSQLへのログイン

```
psql -h localhost -p 5432 -U postgres -d データベース名
```

2. サービスの開始・停止 (PostgreSQL12の例)

```
net start postgresql-x64-12
net stop postgresql-x64-12
```

3. CSVからデータを登録する

```
¥copy テーブル名 from 'CSVのフルパス' with csv
¥copy テーブル名 from 'CSVのフルパス' with csv header
```

4. バックアップ

```
pg_dump -U ユーザー名 -f バックアップファイルパス -Fc DB名
```

5. リストア

```
pg_restore -U postgres -d DB名 バックアップファイルパス
'pg_restore -U ユーザー名 -C -d postgres バックアップファイルパス
```

メンテナンス

1. 手動VACUUM

```
VACUUM テーブル名;
```

2. 手動ANALYZE

```
ANALYZE テーブル名;
```

3. CLUSTER

```
CLUSTER テーブル名 USING インデックス名;
```

4. EXPLAIN

```
EXPLAIN SELECT文;
```

5. 接続中セッション

```
select * from pg_stat_activity where datname = 'DB名';
```

6. 接続中セッションの切断

```
select pg_terminate_backend(pid);
```