

# 第5回：全体を集計する - COUNT, SUM, AVG

---

表全体の件数や合計、平均値を計算できるようになる

# ● 本日のゴール

---

- ✓ 表全体の件数や合計、平均値を計算できるようになる

# ● 座学パート

---

## 1. 集計関数とは？

複数の行をまとめて1つの値を計算する関数。

**COUNT**：件数を数える

**SUM**：合計を計算する

**AVG**：平均を計算する

その他：MAX（最大）、MIN（最小）

## 2. 基本構文

-- 件数を数える

```
SELECT COUNT(*) FROM 'data/sales.csv';
```

-- 合計を計算

```
SELECT SUM(quantity) FROM 'data/sales.csv';
```

-- 平均を計算

```
SELECT AVG(price) FROM 'data/products.csv';
```

## 3. ポイント

`COUNT(*)` : 全行数を数える (NULLも含む)

`COUNT(列名)` : その列でNULL以外の値を数える

SUMやAVGは数値列にのみ使用可能

結果は1行だけ返ってくる

## 4. 集計の利点

計算結果がすぐに確認できる

複数の集計を同時に実行して比較できる

エラーがあればすぐに分かる

# ● 演習パート

---

## 演習1：sales.csvの売上データは全部で何件？

```
SELECT COUNT(*) FROM 'data/sales.csv';
```

結果：15件

## 応用：特定条件の件数を数える

```
-- C001さんの購入回数は？  
SELECT COUNT(*) FROM 'data/sales.csv'  
WHERE customer_id = 'C001';
```



## 演習2：売上数量の合計を計算

```
SELECT SUM(quantity) FROM 'data/sales.csv';
```

結果を確認：

全部で何個売れた？

1件あたり平均何個？（合計÷件数）

## 演習3：商品の平均価格を計算

```
SELECT AVG(price) FROM 'data/products.csv';
```

## さらに詳しく

-- 最高価格、最低価格、平均価格を一度に

**SELECT**

**MAX**(price) **AS** 最高価格,

**MIN**(price) **AS** 最低価格,

**AVG**(price) **AS** 平均価格

**FROM** 'data/products.csv';

# ● 応用練習

---

## 1. 複数の集計を同時に実行

```
SELECT
    COUNT(*) AS 売上件数,
    SUM(quantity) AS 総販売数,
    AVG(quantity) AS 平均販売数,
    MAX(quantity) AS 最大販売数,
    MIN(quantity) AS 最小販売数
FROM 'data/sales.csv';
```

## 2. 条件付き集計

```
-- 10個以上の大量購入の統計
SELECT
    COUNT(*) AS 大量購入回数,
    SUM(quantity) AS 大量購入総数,
    AVG(quantity) AS 大量購入平均
FROM 'data/sales.csv'
WHERE quantity >= 10;
```

### 3. 計算式と組み合わせ

-- 仮想的な売上金額の計算（商品価格を10,000円と仮定）

**SELECT**

**SUM**(quantity) **AS** 総販売数,

**SUM**(quantity) \* 10000 **AS** 推定売上金額

**FROM** 'data/sales.csv';

# ● 実践的な使い方

---

## ケース1：在庫管理

```
-- 商品マスタの統計情報
SELECT
    COUNT(*) AS 商品数,
    AVG(price) AS 平均価格,
    SUM(price) AS 全商品合計価格
FROM 'data/products.csv'
WHERE category = '電子機器';
```

## ケース2：売上分析

```
-- 期間を絞った売上分析
SELECT
    COUNT(*) AS 取引回数,
    SUM(quantity) AS 総販売数
FROM 'data/sales.csv'
WHERE order_date >= '2024-01-20';
```



## ケース3：顧客分析

```
-- アクティブ顧客数（購入履歴のある顧客）  
SELECT COUNT(DISTINCT customer_id) AS アクティブ顧客数  
FROM 'data/sales.csv';
```

# ● 効果的な分析

---

## 段階的な分析アプローチ

### 1. まず全体を把握

```
SELECT COUNT(*) AS 全データ数 FROM 'data/sales.csv';
```

### 2. 条件を加えて絞り込み

```
SELECT COUNT(*) AS 特定顧客数  
FROM 'data/sales.csv'  
WHERE customer_id = 'C001';
```

### 3. 割合を計算

```
-- C001さんの購入割合
SELECT
    COUNT(CASE WHEN customer_id = 'C001' THEN 1 END) AS C001の購入回数,
    COUNT(*) AS 全購入回数,
    ROUND(COUNT(CASE WHEN customer_id = 'C001' THEN 1 END) * 100.0 / COUNT(*), 2) AS 購入割合
FROM 'data/sales.csv';
```

# ● 集計関数の使い分け

関数	用途	例
COUNT(*)	全行数を数える	売上件数、顧客数
COUNT(列)	NULL以外を数える	有効データ数
SUM	合計値	売上総額、在庫総数
AVG	平均値	平均単価、平均購入数
MAX	最大値	最高価格、最大購入数
MIN	最小値	最低価格、最小購入数

# ● 注意点と活用Tips

---

## NULL値の扱い

COUNT(\*)はNULLも数える

COUNT(列名)はNULLを除外

SUM、AVGもNULLを除外して計算

## 0件の場合

```
-- 該当データが0件の場合
SELECT COUNT(*) FROM 'data/sales.csv'
WHERE customer_id = 'C999'; -- 存在しない顧客
-- 結果：0
```

```
SELECT SUM(quantity) FROM 'data/sales.csv'
WHERE customer_id = 'C999';
-- 結果：NULL (0ではない！)
```

# ● 本日のまとめ

---

今日学んだこと：

- ✓ `COUNT(*)` で件数を数える
- ✓ `SUM(列名)` で合計を計算
- ✓ `AVG(列名)` で平均を計算
- ✓ `MAX` で最大値、`MIN` で最小値
- ✓ `WHERE` 句と組み合わせて条件付き集計も可能

# よく使うパターン

-- 基本的な統計情報

**SELECT**

**COUNT**(\*) **AS** 件数,  
**SUM**(数値列) **AS** 合計,  
**AVG**(数値列) **AS** 平均,  
**MAX**(数値列) **AS** 最大,  
**MIN**(数値列) **AS** 最小

**FROM** テーブル;

-- 条件付き集計

**SELECT** 集計関数

**FROM** テーブル

**WHERE** 条件;



# ● 次回予告

---

第6回では、GROUP BYを使って「顧客ごと」「商品ごと」といったグループ単位での集計を学びます。

より実践的な分析ができるようになります！

# ● 追加演習問題

---

## 問題1：基本的な集計関数

以下の集計を実行してください：

- 1. products.csvの最高価格を表示
- あなたの答えをここに書いてください
  
- 2. sales.csvの最小購入数量を表示
- あなたの答えをここに書いてください
  
- 3. customers.csvの登録顧客数をカウント
- あなたの答えをここに書いてください

## 問題2：条件付き集計

WHERE句と組み合わせた集計：

-- 1. 電子機器カテゴリの商品数と平均価格

**SELECT**

**COUNT**(\*) **AS** 商品数,

**AVG**(\_\_\_\_) **AS** 平均価格

**FROM** 'data/products.csv'

**WHERE** \_\_\_\_ = '\_\_\_\_';

-- 2. 2024年1月25日以降の売上件数と合計数量

-- あなたの答えをここに書いてください

-- 3. 顧客C002の購入回数と平均購入数量

-- あなたの答えをここに書いてください

## 問題3：複数の集計を組み合わせ

一つのクエリで複数の情報を取得：

```
-- products.csvの統計情報を一度に表示
SELECT
    ____ AS 商品総数,
    ____ AS 最高価格,
    ____ AS 最低価格,
    ____ AS 平均価格,
    ____ AS 価格の合計
FROM 'data/products.csv';
```

## 問題4：実務シナリオ

ビジネスで使える集計：

- 1. 日次売上サマリー（特定日の売上統計）
- 2024年1月20日の売上について以下を集計：
  - - 取引件数
  - - 販売商品の種類数（ヒント：DISTINCT）
  - - 総販売数量
- あなたの答えをここに書いてください
  
- 2. 在庫価値の計算
- 全商品の合計価格（在庫価値と仮定）
- あなたの答えをここに書いてください

## チャレンジ問題

- sales.csvから以下を計算してください：
- 1. 全取引の平均購入数量（小数点以下1桁）
- 2. 購入数量が平均以上の取引件数
- ヒント：サブクエリを使わずに、2回に分けて実行



## 実践問題：売上分析レポート

- 以下の分析を行ってください：
- 1. 期間売上分析（1月15日～1月20日）
  - - 期間中の総売上件数
  - - 期間中の総販売数量
  - - 1日あたりの平均取引件数
- 2. 顧客別の購入分析（特定顧客C001）
  - - 総購入回数
  - - 総購入数量
  - - 1回あたりの平均購入数量

# デバッグ練習

以下のクエリの問題点を見つけて修正：

-- エラー1：集計関数の使い方

```
SELECT customer_id, COUNT(*)  
FROM 'data/sales.csv';
```

-- エラー2：文字列にSUMを使用

```
SELECT SUM(customer_name)  
FROM 'data/customers.csv';
```

-- エラー3：集計関数とWHERE句の組み合わせ

```
SELECT COUNT(*)  
FROM 'data/sales.csv'  
WHERE COUNT(*) > 5;
```





## 応用：NULL値の扱い

- NULL値がある場合の集計の違いを理解：
- 1. COUNT(\*)とCOUNT(列名)の違い

**SELECT**

COUNT(\*) **AS** 全行数,

COUNT(customer\_id) **AS** 顧客ID数,

COUNT(**DISTINCT** customer\_id) **AS** ユニーク顧客数

**FROM** 'data/sales.csv';

- 2. NULLを0として扱う (COALESCE関数)
- 仮にNULL値があった場合の対処法

**SELECT**

SUM(COALESCE(quantity, 0)) **AS** 合計数量

**FROM** 'data/sales.csv';



## 総合演習

- sales.csv、products.csv、customers.csvそれぞれについて
- 以下の「データヘルスチェック」クエリを作成：
- 1. レコード数
- 2. 主要な数値列の最小・最大・平均値
- 3. 異常値の可能性（例：価格が0の商品、数量が極端に多い売上など）

# ● FAQ

---

**Q: COUNT(1)とCOUNT(\*)の違いは？**

A: 実質的に同じです。COUNT(\*)の方が一般的。

**Q: 小数点以下を丸めたい**

A: ROUND関数を使います。 `ROUND(AVG(price), 2)` で小数点以下2桁。

**Q: 文字列は集計できる？**

A: COUNTは可能ですが、SUMやAVGは数値列のみです。