

# 第3回：行を絞る - WHERE -

---

特定の条件に合うデータだけを表示できるようになる

# ● 本日のゴール

---

- ✓ 特定の条件に合うデータだけを表示できるようになる

# ● 座学パート

---

## 1. WHERE句の役割

膨大なデータから必要な情報だけを取り出す「**フィルター**」の役割。

例：1万件の売上データから「今日の売上」だけを見たい

## 2. WHERE句の基本構文

```
SELECT 列名 FROM 'ファイル名' WHERE 条件;
```

## 3. 条件の書き方

```
-- 数値の比較
WHERE price > 1000      -- 1000より大きい
WHERE price >= 1000     -- 1000以上
WHERE price = 1000      -- 1000と等しい
WHERE price < 1000      -- 1000より小さい
WHERE price <= 1000     -- 1000以下
```

## 条件の書き方（続き）

-- 文字列の比較

**WHERE** customer\_id = 'C001' -- 完全一致（' 'で囲む）

**WHERE** category = '電子機器' -- 日本語もOK

-- 複数条件

**WHERE** price > 1000 **AND** category = '電子機器' -- 両方満たす

**WHERE** price > 50000 **OR** category = '家具' -- どちらか満たす

## 4. WHERE句を使うメリット

- 条件を変えながら結果をすぐに確認できる
- 件数が表示されるので、絞り込みの効果が分かる
- エラーがあればすぐに気づける

# ● 演習パート

---

## 演習1：価格が1000円以上の商品を探そう

1. 以下クエリを実行：

```
SELECT * FROM 'data/products.csv'  
WHERE price >= 1000;
```

2. 結果を確認

- 何件表示されたか？
- どの商品が条件を満たしたか？



## 演習2：顧客IDが「C001」の売上データだけを表示

```
SELECT * FROM 'data/sales.csv'  
WHERE customer_id = 'C001';
```

 **注意**：文字列は必ずシングルクォート（`'`）で囲む！

結果を見て確認：

- C001の田中太郎さんの購入履歴だけが表示される
- 何回購入しているか数えてみよう

## 演習3：商品IDが「P003」で、かつ数量が10個以上のデータ

```
SELECT * FROM 'data/sales.csv'  
WHERE product_id = 'P003' AND quantity >= 10;
```

結果の解釈：

- メカニカルキーボード（P003）を10個以上買った記録
- 該当するデータは何件？

# ● 応用練習

---

## 1. OR条件を使ってみよう

```
-- 電子機器または価格が30000円以上の商品
SELECT product_name, price, category
FROM 'data/products.csv'
WHERE category = '電子機器' OR price >= 30000;
```

## 2. 条件を組み合わせよう

```
-- 2024年1月20日以降の、数量5個以上の売上
SELECT * FROM 'data/sales.csv'
WHERE order_date >= '2024-01-20'
      AND quantity >= 5;
```

## 3. 否定条件を使ってみよう

```
-- C001以外の顧客の購入履歴  
SELECT * FROM 'data/sales.csv'  
WHERE customer_id != 'C001';
```

# ● 実践的な使い方

---

## ケース1：在庫確認

```
-- 高額商品（3万円以上）のリスト
SELECT
    product_id AS 商品コード,
    product_name AS 商品名,
    price AS 価格
FROM 'data/products.csv'
WHERE price >= 30000;
```

## ケース2：顧客分析

-- 特定顧客の大量購入（10個以上）履歴

**SELECT**

customer\_id **AS** 顧客ID,

product\_id **AS** 商品ID,

quantity **AS** 購入数,

order\_date **AS** 購入日

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

**WHERE** quantity >= 10;

# ● 効率的なデバッグ

---

## よくあるエラーと対処法

### 1. 文字列のクォート忘れ

-- ✕ エラーになる

```
WHERE customer_id = C001
```

-- ✓ 正しい

```
WHERE customer_id = 'C001'
```



## 2. 比較演算子の間違い

-- ✕ 代入演算子 (SQLでは使えない)

**WHERE** price := 1000

-- ✓ 比較演算子

**WHERE** price = 1000

### 3. ANDとORの優先順位

-- 括弧を使って明確に

```
WHERE (price > 10000 OR category = '家具')  
      AND quantity > 5
```

# ● 条件による絞り込みの確認

結果の件数がすぐ分かるので、以下を試してみましょう：

-- 全データ

```
SELECT COUNT(*) FROM 'data/sales.csv'; -- 結果：15件
```

-- C001の購入履歴

```
SELECT COUNT(*) FROM 'data/sales.csv'  
WHERE customer_id = 'C001'; -- 結果：？件
```






-- 10個以上の大量購入

```
SELECT COUNT(*) FROM 'data/sales.csv'  
WHERE quantity >= 10; -- 結果：？件
```

# ● 本日のまとめ

---

今日学んだこと：

-  WHERE 句で条件に合うデータだけを抽出
-  数値の比較（ > , >= , = , < , <= ）
-  文字列の比較（シングルクォートで囲む）
-  AND で複数条件をすべて満たす
-  OR でいずれかの条件を満たす

# よく使うパターン

-- 数値の範囲指定

```
WHERE price >= 1000 AND price <= 5000
```

-- 特定の値のリスト

```
WHERE customer_id = 'C001'  
      OR customer_id = 'C002'  
      OR customer_id = 'C003'
```

-- 日付の範囲

```
WHERE order_date >= '2024-01-15'  
      AND order_date <= '2024-01-20'
```

# ● 次回予告

---

第4回では、ORDER BY句を使ってデータを並び替える方法を学びます。

価格の高い順、日付の新しい順など、見たい順番でデータを表示できるようになります！

# ● 追加演習問題

---

## 問題1：様々な比較演算子

以下の条件でデータを抽出してください：

- 1. products.csvから価格がちょうど29999円の商品
- あなたの答えをここに書いてください
  
- 2. sales.csvから数量が5個未満のデータ
- あなたの答えをここに書いてください
  
- 3. customers.csvから2023年8月1日以降に登録した顧客
- あなたの答えをここに書いてください

## 問題2：複合条件の練習

ANDとORを使って複雑な条件を作成：

-- 1. 電子機器カテゴリで、かつ価格が10000円以上の商品

```
SELECT * FROM 'data/products.csv'  
WHERE ____ AND ____;
```

-- 2. 顧客IDがC001またはC002の売上データ

```
SELECT * FROM 'data/sales.csv'  
WHERE ____ OR ____;
```

-- 3. 2024年1月20日以降で、かつ数量が5個以上、かつ商品IDがP001の売上

-- あなたの答えをここに書いてください



## 問題3：実務シナリオ

以下のビジネス要件をSQLで表現：

- 1. 在庫確認：1万円未満の低価格商品リスト
- あなたの答えをここに書いてください
  
- 2. VIP顧客抽出：10個以上購入したことがある取引
- あなたの答えをここに書いてください
  
- 3. 期間限定分析：1月25日から1月29日の売上
- あなたの答えをここに書いてください

## 問題4：エラーを見つけて修正

以下のクエリの間違いを修正してください：

```
SELECT * FROM 'data/sales.csv' WHERE customer_id = C001;
```

```
SELECT * FROM 'data/products.csv' WHERE price => 10000;
```

```
SELECT * FROM 'data/sales.csv'  
WHERE customer_id = 'C001' OR customer_id = 'C002' AND quantity > 10;  
-- 意図：C001またはC002の、10個超の購入
```

## チャレンジ問題

- 「購入していない商品」を見つけるヒントになるクエリを書いてください
- 例：特定の商品IDがsales.csvに存在しないことを確認する方法
- ヒント：!=（不等号）を使ってみましょう



## 実践問題：販売戦略分析

- 以下の分析クエリを作成してください：
- 1. 高額商品（3万円以上）の購入履歴
- 2. 少量購入（3個以下）の顧客を特定
- 3. 特定期間（1月15日～1月20日）の特定顧客（C001）の活動

## デバッグ練習

-- 以下のクエリを実行し、なぜ結果が0件か考えてください：

```
SELECT * FROM 'data/products.csv'  
WHERE price > 100000;
```

```
SELECT * FROM 'data/sales.csv'  
WHERE order_date = '2024/01/15'; -- ヒント：日付フォーマット
```

# ● FAQ

---

**Q: 日付の比較はどうすればいい？**

A: 文字列として比較できます。 `WHERE order_date >= '2024-01-20'` のように。

**Q: NULLのデータを探すには？**

A: `WHERE 列名 IS NULL` を使います（`= NULL` ではないので注意）。

**Q: 大文字小文字は区別される？**

A: 文字列の値は区別されます。'C001'と'c001'は別物です。