

11章 より複雑なデータを検索しよう

11章 より複雑なデータを検索しよう

WHERE句を使ったより複雑な条件で特定のデータを検索する方法を学びます。

本章の目標

- AND演算子とOR演算子で複数の条件を指定する方法を知ること
- 指定範囲のデータを抽出する方法を知ること
- 複数の値のうちいずれかに一致するデータの抽出方法を知ること
- 文字列のあいまい検索（「～を含む」など）を行う方法を知ること

11章 より複雑な検索の必要性

前章ではWHERE address = '神奈川県'（住所が神奈川県）やWHERE age <= 30（年齢が30歳以下）など、WHERE句を使って特定のデータを検索しました。

しかし実際のアプリやサービスでは、より複雑な条件で検索することが多いです。

- 「商品価格が3000～5000円」
- 「商品名に『紅茶』を含む」

そこで、本章では前章に引き続きWHERE句を使い、より複雑な条件で特定のデータを検索する方法を学びます。

11章 WHERE句について復習しよう

WHERE句は、SQLの基本的な構文（SELECT、UPDATE、DELETE）と組み合わせて使います。

-- 条件式に当てはまるデータを検索し、取得する

SELECT カラム名 **FROM** テーブル名 **WHERE** 条件式;

-- 条件式に当てはまるデータを検索し、セットされた値で更新する

UPDATE テーブル名 **SET** カラム名1 = 値1, カラム名2 = 値2 **WHERE** 条件式;

-- 条件式に当てはまるデータを検索し、削除する

DELETE FROM テーブル名 **WHERE** 条件式;

11章 演算子について復習しよう

WHERE句でよく使われる演算子は主に以下の2種類です。

比較演算子

2つの値を比較し、その結果をTRUEまたはFALSEで返す演算子

- `>` `<=` `=` `!=` など

論理演算子

複数の条件式を組み合わせて「AかつB」「AまたはB」といった演算を行う演算子

- `AND` `OR` など

11章 比較演算子の復習

比較演算子	使用例	意味
=	A = B	AとBが等しいとき、TRUEを返す
!=	A != B	AとBが等しくないとき、TRUEを返す
<	A < B	AがBよりも小さいとき、TRUEを返す
<=	A <= B	AがB以下のとき、TRUEを返す
>	A > B	AがBよりも大きいとき、TRUEを返す
>=	A >= B	AがB以上のとき、TRUEを返す
IS NULL	A IS NULL	AがNULLのとき、TRUEを返す
IS NOT NULL	A IS NOT NULL	AがNULLでないとき、TRUEを返す

11章 論理演算子の復習

論理演算子	使用例	意味
AND	A AND B	AとBがいずれもTRUEのとき、TRUEを返す
&&	A && B	AとBがいずれもTRUEのとき、TRUEを返す
OR	A OR B	AかBの少なくとも1つがTRUEのとき、TRUEを返す
	A B	AかBの少なくとも1つがTRUEのとき、TRUEを返す
XOR	A XOR B	AかBのどちらか1つがTRUEのとき、TRUEを返す
NOT	NOT A	AがFALSEのとき、TRUEを返す
!	!A	AがFALSEのとき、TRUEを返す

11章 AND演算子とOR演算子で複数の条件を指定しよう

アプリやサービスでは「高度な検索」や「詳細検索」など、複数の条件を指定して検索したい場面が出てきます。

例：「言語が日本語かつ『SQL』というワードを含む」投稿を検索する

そこでよく使うのが、SQLのAND演算子とOR演算子です。

```
SELECT * FROM users WHERE age >= 30 AND address = '神奈川県';
```


11章 AND演算子とOR演算子の書き方

基本的な書き方は以下のとおりです。

-- 「AかつB」という条件に当てはまるデータを検索し、取得する
`SELECT` カラム名 `FROM` テーブル名 `WHERE` A `AND` B;

-- 「AまたはB」という条件に当てはまるデータを検索し、取得する
`SELECT` カラム名 `FROM` テーブル名 `WHERE` A `OR` B;

-- 「Aかつ『BまたはC』」という条件に当てはまるデータを検索し、取得する
`SELECT` カラム名 `FROM` テーブル名 `WHERE` A `AND` (B `OR` C);

11章 評価の優先順位

普通の計算と同じようにSQLでも丸括弧()をつけると優先順位が高くなります。

A AND (B OR C) という式の評価順序：

1. B OR C を評価する
 - TRUEだった場合 → 2に進む
 - FALSEだった場合 → A AND FALSE となり、FALSEを返す
2. A を評価する
 - TRUEだった場合 → TRUE AND TRUE となり、TRUEを返す
 - FALSEだった場合 → FALSE AND TRUE となり、FALSEを返す

11章 AND演算子とOR演算子を使ってみよう

「年齢が30歳以上かつ『住所が神奈川県または大阪府』」という条件を指定してみます。

```
SELECT * FROM users  
WHERE age >= 30 AND (address = '神奈川県' OR address = '大阪府');
```

実行結果

usersテーブルには「年齢が30歳以上かつ『住所が神奈川県または大阪府』」のユーザーが4人いることがわかります。

11章 BETWEEN句で指定範囲のデータを抽出しよう

アプリやサービスでは3000～5000円の商品を検索するときなど、下限値と上限値を指定して検索したい場合があります。

そこでよく使うのが、SQLのBETWEEN句です。

BETWEENは「(2つの) 間で」という意味で、カラムの値が下限値以上、上限値以下の場合にTRUEを返すコマンドです。

例： `WHERE price BETWEEN 3000 AND 5000`

11章 BETWEEN句の書き方

BETWEEN句の基本的な書き方は以下のとおりです。

-- カラムの値が下限値以上、上限値以下のデータを抽出する

SELECT カラム名 **FROM** テーブル名 **WHERE** カラム名 **BETWEEN** 下限値 **AND** 上限値;

注意点

- BETWEEN句は比較演算子の一種
- 本教材では「BETWEEN句」という呼び方で統一

11章 BETWEEN句を使ってみよう

「年齢が30歳以上、40歳以下」という条件を指定してみます。

```
SELECT * FROM users WHERE age BETWEEN 30 AND 40;
```

実行結果

usersテーブルには「年齢が30歳以上、40歳以下」のユーザーが7人いることがわかります。

11章 IN句で複数の値から一致するデータを抽出しよう

30歳、40歳、50歳、60歳のユーザーデータを取得したいとき...

`WHERE age = 30 OR age = 40 OR age = 50 OR age = 60` **と書いてもよいのですが、これでは書くのが大変なうえ、読みにくくなってしまう。**

このように複数の値のうち、いずれかに一致するデータを抽出したいときは、IN句を使えば簡単です。

`WHERE age IN (30, 40, 50, 60)` のようにリスト形式で簡潔に記述できます。

11章 IN句とOR演算子の比較

IN句の有無	WHERE句の書き方（一例）
IN句なし	WHERE age = 30 OR age = 40 OR age = 50 OR age = 60
IN句あり	WHERE age IN (30, 40, 50, 60)

INは「～の中に」という意味で、複数の値のうちいずれかに一致するデータを抽出するコマンドです。

11章 IN句の書き方

IN句の基本的な書き方は以下のとおりです。

-- カラムの値がリスト（複数の値）のうちいずれかに一致するデータを抽出する
SELECT カラム名 **FROM** テーブル名 **WHERE** カラム名 **IN** (値1, 値2,);

IN句の強み

- 値をカンマで区切ってリスト形式で記述
- BETWEEN句のように連続した値でなくてもよい

11章 IN句を使ってみよう

年齢を5歳から75歳まで、10刻みでリストを作ってみます。

```
SELECT * FROM users WHERE age IN (5, 15, 25, 35, 45, 55, 65, 75);
```

実行結果

usersテーブルには「35歳、55歳、65歳、75歳」のユーザーが1人ずついることがわかります。

このように、IN句を使えば複数の値のうちいずれかに一致するデータを抽出できます。

11章 LIKE句で文字列のあいまい検索をしよう

WHERE furigana = 'たろう' と記述すれば、ふりがなが「たろう」と「完全に一致する」データは検索できます。

しかし、「たろう」を「含む」データや「たろう」で「終わる」データなど、あいまいな検索はできません。

そこで使うのがLIKE句です。

LIKEは「～のような」という意味で、その文字列を「含む」データやその文字列で「始まる」「終わる」データなど、あいまい検索を行うコマンドです。

11章 LIKE句の使い方

例： `WHERE furigana LIKE '%たろう%'` と記述すれば、ふりがなに「たろう」を「含む」データを検索できます。

比較演算子の `=` を `LIKE` に置き換えて「『たろう』のような」とあいまいにするイメージです。

活用場面

アプリやサービスで商品名や投稿などの検索機能を作るときは、このLIKE句が使われることが多いです。

11章 LIKE句の書き方

LIKE句の基本的な書き方は以下のとおりです。

-- 指定した文字列であいまい検索を行い、一致するデータを取得する
`SELECT` カラム名 `FROM` テーブル名 `WHERE` カラム名 `LIKE` '検索文字列';

ワイルドカードとは

トランプにおけるジョーカーのような存在で、`%` や `_` のように1つの文字だけであらゆる文字を表せる特殊文字のことです。

11章 LIKE句で使えるワイルドカード

ワイルドカード文字	意味
%	0文字以上の任意の文字列
_	任意の1文字

例

- '%たろう' : 「しょうたろう」「いちたろう」「こたろう」「たろう」のように、「たろう」の前に何文字あっても検索条件にマッチ
- '_たろう' : 「こたろう」「きたろう」のように、「たろう」の前に1文字ある場合しか検索条件にマッチしない

11章 LIKE句を使った代表的な検索方法

検索方法	書き方	例
完全一致	'文字列'	'さむらい たらう' (「さむらい たらう」と完全に一致するデータを検索する)
前方一致	'文字列%'	'さむらい%' (「さむらい」で始まるデータを検索する)
後方一致	'%文字列'	'%たらう' (「たらう」で終わるデータを検索する)
部分一致	'%文字列%'	'%むらい%' (「むらい」を含むデータを検索する)

11章 LIKE句を使ってみよう

氏名に「八」を含むデータを検索してみます。

```
SELECT * FROM users WHERE name LIKE '%八%';
```

実行結果

usersテーブルには氏名に「八」を含むユーザーが1人いることがわかります。

演習の内容

1. データベースとテーブルを作成する
2. WHERE句でデータを検索する
 - 「年齢が50歳以下」かつ「住所が埼玉県」という条件で検索
3. IN句でデータを検索する
 - 「住所が九州の7県（福岡県、佐賀県、長崎県、熊本県、大分県、宮崎県、鹿児島県）」という条件で検索

WHERE句でデータを検索

-- 「年齢が50歳以下」かつ「住所が埼玉県」のデータを検索
`SELECT * FROM users WHERE age <= 50 AND address = '埼玉県';`

解説

「AかつB」という複数の条件に当てはまるデータを検索する場合、AND演算子を使います。

11章 演習の解答例 (2/2)

IN句でデータを検索

```
-- 「住所が九州の7県」のデータを検索
SELECT * FROM users
WHERE address IN ('福岡県', '佐賀県', '長崎県', '熊本県',
                  '大分県', '宮崎県', '鹿児島県');
```

解説

複数の値のうち、いずれかに一致するデータを抽出するには、IN句を使います。

11章 まとめ (1/3)

- 普通の計算と同じように、SQLでも丸括弧()をつけると評価の優先順位が高くなる
- **BETWEEN**は「(2つの) 間で」という意味で、カラムの値が下限値以上、上限値以下の場合にTRUEを返すコマンド
- **IN**は「～の中に」という意味で、複数の値のうちいずれかに一致するデータを抽出するコマンド

11章 まとめ (2/3)

- **LIKE**は「～のような」という意味で、その文字列を「含む」データやその文字列で「始まる」「終わる」データなど、あいまい検索を行うコマンド
- **ワイルドカード**とは、`%`や`_`のように1つの文字だけであらゆる文字を表せる特殊文字のこと

ワイルドカード文字	意味
<code>%</code>	0文字以上の任意の文字列
<code>_</code>	任意の1文字

11章 まとめ (3/3)

-- 「AかつB」という条件に当てはまるデータを検索し、取得する

SELECT カラム名 **FROM** テーブル名 **WHERE** A **AND** B;

-- 「AまたはB」という条件に当てはまるデータを検索し、取得する

SELECT カラム名 **FROM** テーブル名 **WHERE** A **OR** B;

-- カラムの値が下限値以上、上限値以下のデータを抽出する

SELECT カラム名 **FROM** テーブル名 **WHERE** カラム名 **BETWEEN** 下限値 **AND** 上限値;

-- カラムの値がリスト（複数の値）のうちいずれかに一致するデータを抽出する

SELECT カラム名 **FROM** テーブル名 **WHERE** カラム名 **IN** (値1, 値2,);

-- 指定した文字列であいまい検索を行い、一致するデータを取得する

SELECT カラム名 **FROM** テーブル名 **WHERE** カラム名 **LIKE** '検索文字列';