Woodland Academy

5.6 SQL Index

- インデックス
- 実践練習

Shape Your Future

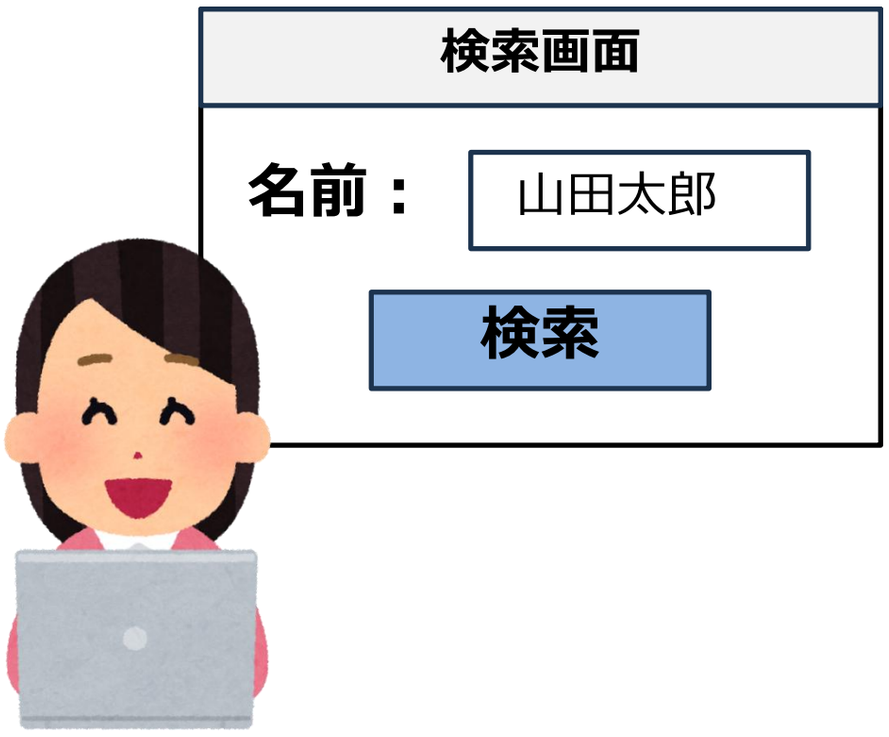
目次

インデックスとは

- インデックスとは、**本棚にある「索引カード」みたいなもの**とイメージすると理解しやすいでしょう。

- 例えば、図書館で「ハリーポッター賢者の石」を探したいとき、1冊ずつ本を開くよりも、**索引カード**を見て書棚の場所がわかれば速いです。

- **データベース**でも同じように考えることができます。
  - テーブル　　：「本棚」
  - データ　　：「本」
  - インデックス：「索引カード」



大量にデータが登録されているテーブル

社員番号	名前	住所
20200001	鈴木一郎	東京都渋谷区
20200002	山田太郎	大阪府大阪市
20200003	田中一郎	神奈川県横浜市
20200004	斎藤五郎	北海道札幌市
…	…	…

インデックスを設定

- よく使う列にインデックスをつけると、**索引用のデータが**
  テーブルとは別に作成され、検索が**高速**になります。

次へ

インデックスの作り方

- 基本構文：
  CREATE INDEX インデックス名 **ON** テーブル名(カラム名);

```
1 -- テーブルの特定の列にインデックスを作成する
2 CREATE INDEX index_name ON table_name (column_name);
3
```

（例）家計簿テーブルの項目「メモ」にインデックスを作成

```
1 --  「memo」列にインデックスを作成
2 CREATE INDEX idx_memo ON kakeibo (memo);
3
```

- これで「memo = ‘○○'」を探すとき、1件ずつ見なくても**パッと見つかる**ようになります！

Woodland Academy

インデックスの使いどころ①

- **WHERE句**による検索

（例）

```sql
-- memo列にインデックスがあると、この検索が高速になる
SELECT * FROM kakeibo WHERE memo = '1月の電気代';

```

- 例えるなら：索引カードで「1月の電気代」を引いて、すぐ見つけることができます。

Woodland Academy

インデックスの使いどころ②

- **ORDER BY**句による並び替え

（例）

```
1 -- date列にインデックスがあると、並び替えが効率的
2 SELECT * FROM kakeibo ORDER BY date DESC;
3
```

- 例えるなら：すでに日付順に並んでる棚を眺めるだけで済みます。

インデックスの使いどころ③

- **JOIN**条件の高速化

（例）

```sql
-- 結合キーにインデックスがあると処理が速い
SELECT *
FROM kakeibo
JOIN himoku ON kakeibo.himoku_id = himoku.id;
```

- 例えるなら：2つの図書館の目録番号で簡単に本を突き合わせられる状態です。

インデックスを付けるべき例

- 先ほどの使いどころをふまえて、どのような場合にインデックスを付けるべきか確認しましょう。

列の用途	理由
よく検索される列（WHERE）	絞り込みが速くなる。
並び替えに使う列（ORDER BY）	ソート処理を軽減。
結合に使う列（JOIN）	JOINが高速になる。
主キーやユニークキー	自動でインデックスが付く。（例：IDなど）

Woodland Academy

インデックスの注意点

- **問題①容量を消費する。**
　⇒**インデックス自体もデータとして保存される**ため、テーブルが増えるほど容量を圧迫します。とくに複数のカラムにインデックスを付けると、ストレージ負担が大きくなります。

- **問題②書き込みが遅くなる。**
　⇒**INSERT / UPDATE / DELETE などの操作時に、インデックスも同時に更新される**ため、処理に余分な時間がかかります。読み込みは速くなっても、書き込みが多い場面では逆にパフォーマンスが悪化することがあります。

- **問題③インデックスが多すぎると逆効果になる。**
　⇒**必要以上にインデックスを作ると、どれを使うか判断する時間や管理コストがかかる**ため、検索速度が落ちることがあります。ちょうど「図書館の索引カードが多すぎて、探すのが大変になる」ようなイメージです。

Woodland Academy

Woodland Academy

Q&A

目次

テーブルの作成

- インデックス実践練習前に、まずは、使用するテーブルとデータの作成をします。今回は家計簿テーブルを作成してもらいます。

Try

index.sql

インデックスの効果（インデックスなし）

- インデックスの効果を確認するために、まずはインデックスなしの状態で処理速度を計測しましょう。これは、**実行計画**で確認することができます。

- 実行計画とは：**データベースがクエリを「どうやって実行するか」を示した設計図**です。**検索や並び替えの方法・順番・効率**がわかります。主にSQLの**速さや重さを確認・改善**したいときに使います。

Woodland Academy

次へ

インデックスの効果（インデックスなし）

- **EXPLAIN** を使って**実行計画**を確認することができます。

```
1 -- 「電気代」というメモを検索
2 EXPLAIN SELECT * FROM kakeibo WHERE memo = '1月の電気代';
```

	QUERY PLAN text
1	Seq Scan on kakeibo (cost=0.00..23.38 rows=5 width=48)
2	Filter: (memo = '1月の電気代'::text)

```
1 -- 出金額が高い順に表示
2 EXPLAIN SELECT * FROM kakeibo ORDER BY expense DESC;
```

	QUERY PLAN text
1	Sort (cost=74.54..77.21 rows=1070 width=48)
2	Sort Key: expense DESC
3	-> Seq Scan on kakeibo (cost=0.00..20.70 rows=1070 width=48)

次へ

Woodland Academy

インデックスの作成

- 今回は、**検索、並び替えを高速化**するために、下記のようにインデックスを作成します。

```
1 -- メモ列に対する検索用インデックス
2 CREATE INDEX idx_memo ON kakeibo (memo);
3
4 -- 日付での並び替え用インデックス
5 CREATE INDEX idx_date ON kakeibo (date);
6
```

次へ

インデックスの効果（インデックスあり）

- **EXPLAIN** を使って**実行計画**を確認します。

```
1 -- 「電気代」というメモを検索
2 EXPLAIN SELECT * FROM kakeibo WHERE memo = '1月の電気代';
```

	QUERY PLAN text
1	Seq Scan on kakeibo (cost=0.00..1.07 rows=1 width=48)
2	Filter: (memo = '1月の電気代'::text)

インデックスなしと比較
⇒約 22.31 コスト削減！

```
1 -- 出金額が高い順に表示
2 EXPLAIN SELECT * FROM kakeibo ORDER BY expense DESC;
```

	QUERY PLAN text
1	Sort (cost=1.14..1.15 rows=6 width=48)
2	Sort Key: expense DESC
3	-> Seq Scan on kakeibo (cost=0.00..1.06 rows=6 width=48)

インデックスなしと比較
⇒並び替え ： 約 267倍 軽い！
テーブル読み込み ： 約 19.64
コスト削減！

Woodland Academy

インデックスまとめ

- **検索を速くするための「目次」や「地図」の役割**
  ⇒大量データから目的の情報をすばやく探すことができます。

- **検索が早くなるメリットが大きい**
  ⇒特に頻繁に検索・並び替えに使う列に効果的です。

- **付けすぎは逆効果**
  ⇒容量を消費したり、書き込み（追加・更新・削除）が遅くなったり、混乱の元になる可能性もあるため、使う目的に合わせて必要なところだけに絞るのがコツです。

次へ

Woodland Academy

Woodland Academy

Q&A

まとめ

Sum Up

1.インデックスについて
　使いどころや注意点を確認

2.インデックスの作成方法と実行結果の確認方法

Woodland Academy

Woodland Academy

Thank you!

From Seeds to Woodland — Shape Your Future.

Shape Your Future