

# 生成AI基礎

## 6 データベースサーバーの基本

# データベースサーバーの役割

- クライアントからのデータベースクエリを処理し、適切なデータを返す。
- データの整合性を保持。

# データベースの種類

- リレーショナルデータベース: テーブルとしてデータを整理し、SQLで操作。
- NoSQLデータベース: 非リレーショナルで、柔軟なデータモデルを提供。例にはドキュメント、キーバリュー、グラフベースがある。

# リレーショナルデータベースの基本概念

# テーブル、行（レコード）、列（フィールド）の概念

- テーブル：データベース内のデータを構造化する基本的な単位。
- 行（レコード）：テーブル内の個々のデータアイテム。
- 列（フィールド）：テーブル内のデータの特定の属性。

# 主キーと外部キーの説明

- 主キー：テーブル内の各レコードを一意に識別するための列。
- 外部キー：異なるテーブルの行を関連付けるための列。

# データベースの正規化の概要

- 冗長性を排除し、データの一貫性を保つための設計プロセス。
- 第1正規形から第3正規形までの各レベルが存在。

# 仮想テーブルの説明

- ビュー（仮想テーブル）：物理的には存在しないが、一つ以上のテーブルから派生した結果として定義されるテーブル。



# SQL基礎

# SQLの概要と基本構文

- SQL（Structured Query Language）：データベースから情報を取得、更新、操作するための標準言語。
- 基本的な構文には、SELECT、FROM、WHEREなどが含まれる。

利用したデータセット：

『江戸料理レシピデータセット』（CODH作成） 『日本古典籍データセット』（国文学研究資料館蔵）を翻案

提供：ROIS-DS人文学オープンデータ共同利用センター(<http://codh.rois.ac.jp/>)

# SQLiteについて

- **ファイルベース:** 一つのディスクファイルに全てのデータを保存します。これにより、簡単に扱うことができ、ポータビリティが高まります。
- **軽量:** システムリソースをほとんど消費せず、小規模なアプリケーションに最適です。
- **無料かつオープンソース:** パブリックドメインのソフトウェアで、誰でも自由に使用、改変、配布が可能です。

# SELECT文によるデータの取得

- 特定の列や全列のデータを取得。
- 特定の条件に基づいたデータの抽出。

```
SELECT name, main_ingredients, primary_cooking_methods  
FROM recipes;
```

# WHERE句によるデータの絞り込み

- 指定された条件に一致するデータのみを選択。
- 比較演算子や論理演算子を用いる。

```
SELECT name, food_category_1, ingredients  
FROM recipes  
WHERE food_category_1 = '野菜';
```

# ORDER BY句によるデータの並び替え

- 結果セットのデータを特定の列で昇順または降順に並べ替え。

```
SELECT name, seasonings  
FROM recipes  
ORDER BY name DESC;
```

# INSERT文によるデータの挿入

- 新しい行をテーブルに追加。
- 必要な値を明確に指定。

```
INSERT INTO recipes (name, abstract, ingredients, steps)
VALUES ('新しい料理', '料理の概要', '材料のリスト', '調理手順');
```

# UPDATE文によるデータの更新

- 既存のデータを修正。
- 条件を指定して特定のレコードのみを更新。

```
UPDATE recipes  
SET abstract = 'パーティー向け'  
WHERE name = '新しい料理';
```



# DELETE文によるデータの削除

- 条件に一致するレコードをテーブルから削除。

```
DELETE FROM recipes  
WHERE name = '新しい料理';
```

# JOIN文によるデータの結合

- 二つ以上のテーブルから関連するデータを結合して一つの結果セットを生成。

```
SELECT * FROM recipes
JOIN (SELECT name, like FROM like WHERE like = True)
AS like ON recipes.name = like.name;
```

# 全文検索

# 全文検索の概要

- 文書全体からテキストを検索する技術。
- インデックスを利用して高速な検索を実現。

# 全文検索クエリ

- 特定のキーワードに基づいた検索を実行。

```
SELECT name, steps FROM recipes_fts  
WHERE steps MATCH 'レンジ';
```

- 検索精度を高めるためのオペレータ（AND, OR, NOT）の使用。

```
SELECT name, steps FROM recipes_fts  
WHERE steps MATCH 'レンジ 片栗粉';
```

```
SELECT name, steps FROM recipes_fts  
WHERE steps MATCH 'レンジ OR フライパン';
```

参考:(<https://www.sqlite.org/fts5.html>)

# 日本語の前処理（分かち書き）

- 日本語テキストを単語単位に分割する処理。
- 検索やテキスト解析の精度を高めるために重要。
- 日本語用の形態素解析器（MeCab、Kuromoji等）がある

# まとめ

- データベースサーバーは、データの整合性を保持し、クライアントからのクエリを処理する。
- リレーショナルデータベースは、テーブル、行、列の概念を持ち、SQLで操作。
- SQLを用いてデータの取得、更新、削除、結合が可能。
- 全文検索は、文書全体からテキストを検索する技術で、日本語の前処理が重要。