

1. 概要

本システムは、CSVデータおよびJSONデータを基にして、SQLのINSERT文や各種ファイル形式（CSV、JSON）の生成を行います。ユーザーが出力ファイル形式を選択し、変換ボタンを押下すると、指定された形式のファイルを出力します。

2. 機能要件

2.1. ファイル入力

- **対応フォーマット:**
 - SQLのINSERT文
 - CSVファイル
 - JSONファイル
- **処理内容:**
 - ファイルを読み込み、選択した出力形式に変換。

2.2. ファイル出力

- **出力形式:**
 - SQLのINSERT文
 - CSVファイル
 - JSONファイル
- **ユーザー操作:**
 - ユーザーが出力形式を選択し、変換ボタンをクリックする。

3. システム構成

3.1. ユーザーインターフェース

- **ファイル選択:**
 - ファイル選択ダイアログを使用してCSVまたはJSONファイルを選択。
- **出力形式選択:**
 - ドロップダウンメニューで出力形式を選択。
- **変換ボタン:**
 - ユーザーが設定を確定し、変換処理を開始するボタン。

4. 処理フロー

4.1. ファイルの読み込み

1. ユーザーが入力ファイルを選択。

4.2. 出力形式の選択

1. ユーザーが出力形式を選択。
2. システムが選択された形式を内部で保持。

4.3. 変換処理

1. ユーザーが変換ボタンをクリック。
2. システムが内部データ構造を選択された出力形式に変換。
3. エクスポートファイルを生成し、ユーザーに保存場所を指定させる。

5. クラス設計

5.1. ファイル管理クラス

- **FileHandler**
 - ファイルの読み込みおよび書き込みを担当。
 - メソッド: `ReadCSV()`, `ReadJSON()`, `ExportAsCSV()`, `ExportAsJSON()`, `GenerateSQLFromCSV()`, `GenerateSQLFromJSON()`

5.2. データ管理クラス

- **DataManager**
 - 読み込んだデータを内部データ構造に変換および保持。
 - メソッド: `ParseCSV()`, `ParseJSON()`, `ConvertToInternalStructure()`, `ConvertToCSV()`, `ConvertToJSON()`, `GenerateInsertSQL()`

5.3. ユーザーインターフェースクラス

- **UserInterface**
 - ユーザーとの対話を担当。
 - メソッド: `SelectFile()`, `SelectOutputFormat()`, `StartConversion()`

6. インターフェース設計

6.1. ファイル選択ダイアログ

- **操作:** ファイルシステムから入力ファイルを選択。
- **表示:** 選択されたファイル名を表示。

6.2. 出力形式選択

- **操作:** ドロップダウンメニューから希望の形式を選択。
- **表示:** 選択された形式を表示。

6.3. 変換ボタン

- **操作:** 変換処理を開始。

7. エラーハンドリング

- **ファイル読み込みエラー:**
 - 対応策: エラーメッセージを表示し、再試行を促す。
- **形式変換エラー:**

- 対応策: 変換エラーの場合は詳細メッセージを表示。

8. ユーザーマニュアル

1. ファイル選択:

- ファイル選択ボタンをクリックし、入力ファイルを選択。

2. 出力形式選択:

- ドロップダウンメニューから希望の形式を選択。

3. 変換開始:

- 変換ボタンをクリックし、エクスポートファイルの保存場所を指定。