

# 人材マッチング調査オープンデータに関するユーザーガイド

---

このガイドでは、e-CSTI が提供する人材マッチング調査オープンデータを用いて、MySQL または PostgreSQL 上に分析データベースを構築する方法を説明します。

- 文中の p4ds2023 とは、「2023 年調査までのデータを格納した人材マッチング調査分析データベース」の名称です。導入時には、ご自身の環境に合わせて任意の名称に変更し、以降のスクリプトで一貫してご使用ください。
- ファイル名やスクリプト名の \_v[version] は、実際のバージョン番号（例：v1.0, v1.3 など）に置き換えてご利用ください。

## 1 分析用データ

- p4ds2023\_tables\_csv\_v[version].zip
  - 分析用に正規化・整形された 44 テーブル分の CSV ファイルが含まれています。以下の手順でご使用ください。
    1. ZIP ファイルを展開し、CSV ファイルを データロード用の保存先フォルダ に配置する。
    2. データロード用スクリプトのデータ保存先パス（your\_upload\_data\_location と記述されている部分）を、ご自身の環境に応じて修正（置換）する。
    3. 先にテーブル構造作成用スクリプトを実行し、次にデータロード用スクリプトを実行する。

- survey\_unified\_canonical\_v[version].zip（加工前データ）

※ このファイルはデータベース構築には使いません

- 過去の人材マッチング調査データを縦に連結した CSV 形式ファイルです。
- 複数選択式の設問に関する応答が複数列にまたがって格納されています。
- Excel などでの閲覧や加工前のデータ確認にご利用いただけます。

## 2 データベース作成用スクリプト

- MySQL と PostgreSQL 向けのテーブル構造作成用とデータロード用のスクリプト例を提供しています。
- スクリプトは Windows 環境（BOM付きUTF-8）で作成されています。
- 先にテーブル構造作成用スクリプトを、次にデータロード用スクリプトを実行する前提で作成されています。

### 2.1 MySQL 用

- p4ds2023\_Schema\_MySQL\_v[version].sql (テーブル構造作成用スクリプト)
- p4ds2023\_LoadData\_MySQL\_v[version].sql (データロード用スクリプト)

### 2.2 PostgreSQL 用

- p4ds2023\_Schema\_PostgreSQL\_v[version].sql (テーブル構造作成用スクリプト)
- p4ds2023\_LoadData\_PostgreSQL\_v[version].sql (データロード用スクリプト)

## 3 オープンデータ説明資料

### 3.1 人材マッチング調査に関する概要資料

- p4ds2023\_Document\_v[version].pdf
- p4ds2023\_Document\_v[version].md
  - 人材マッチング調査全体の概要、目的、調査方法などを網羅した報告書です。
  - PDF形式とMarkdown形式で提供します。

### 3.2 人材マッチング調査オープンデータに関するユーザーガイド

- p4ds2023\_UserGuide\_v[version].pdf
- p4ds2023\_UserGuide\_v[version].md
  - 人材マッチング調査を使った分析者向けに、オープンデータの使い方を説明したユーザーガイドです。
  - PDF形式とMarkdown形式で提供します。

### 3.3 人材マッチング調査設問リスト

- p4ds2023\_QuestionList\_v[version].xlsx:
  - 人材マッチング調査各回の設問を一覧形式でまとめたExcelファイルです。
  - 設問の文言が調査時の表現のまま掲載されています。

## 4 本ユーザーガイドが想定する環境

### 4.1 MySQL の場合

- MySQL Server 8.0 がインストール済み
- MySQL クライアントツールを使用（例：MySQL Workbench、コマンドラインクライアント）

### 4.2 PostgreSQL の場合

- PostgreSQL がインストール済み（ローカルまたはリモート環境）
- psql インターフェース または GUI ツールを使用（例：pgAdmin）

## 5 MySQL を用いたデータベース構築手順

### 5.1 MySQL のセットアップと接続

1. MySQL クライアントツールを開く（Workbench または コマンドライン）
2. MySQL Server に接続

### 5.2 p4ds2023 データベースの作成

本ガイドで扱うデータは 日本語を含むため、文字化けを防ぐためにもデータベース作成時に UTF-8（utf8mb4） の設定を指定してください。以下は作成スクリプト例です。

```
CREATE DATABASE p4ds2023db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;  
USE p4ds2023db;
```

- p4ds2023db はデータベース名の一例です。任意の名前に変更可能です。
- utf8mb4 は絵文字なども扱える UTF-8 の上位互換です。
- utf8mb4\_unicode\_ci を使用することで、日本語を含む文字列比較において適切な順序や一致判定が行えます。

### 5.3 テーブル構造の作成

スクリプト（例）

```
source /path/to/p4ds2023_Schema_MySQL_v[version].sql;
```

※ /path/to/ は、ご自身の環境に合わせて適宜変更してください。

### 5.4 データのロード

1. p4ds2023\_csv\_v[version].zip を解凍し、データロード用ディレクトリに配置
2. p4ds2023\_LoadData\_MySQL\_v[version].sql を開き、データの保存先パスを環境に合わせて変更
3. スクリプトを実行してデータをロードする

スクリプト（例）

```
source /path/to/p4ds2023_LoadData_MySQL_v[version].sql;
```

※ /path/to/ は、ご自身の環境に合わせて適宜変更してください。

## 6 PostgreSQL を用いたデータベース構築手順

### 6.1 PostgreSQL のセットアップと接続

1. psql で接続する場合

スクリプト（例）

```
psql -U ユーザー名 -d データベース名
```

2. pgAdmin を利用する場合

- PostgreSQL Server に接続
- クエリエディタを開く

### 6.2 データベースとスキーマの作成

本ガイドで扱うデータは日本語を含むため、文字化けを防ぐためにもデータベース作成時に UTF-8 エンコーディングを指定してください。以下は作成スクリプト例です。

```
CREATE DATABASE p4ds2023db
WITH ENCODING 'UTF8'
LC_COLLATE='ja_JP.UTF-8'
LC_CTYPE='ja_JP.UTF-8'
TEMPLATE=template0;
```

- p4ds2023db はデータベース名の一例です。任意の名前に変更可能です。
- ja\_JP.UTF-8 を設定することで、日本語データの並び順や大文字・小文字の扱いも日本語に適したものになります。

作成したデータベースに接続した後、スキーマを作成します。以下は作成スクリプト例です。

```
CREATE SCHEMA p4ds2023;
```

- p4ds2023db はスキーマ名の一例です。任意の名前に変更可能です。

### 6.3 テーブル構造の作成

スクリプト（例）

```
\i /path/to/p4ds2023_Schema_PostgreSQL_v[version].sql
```

※ /path/to/ は、ご自身の環境に合わせて適宜変更してください。

### 6.4 データのロード

1. p4ds2023\_csv\_v[version].zip を解凍し、データロード用ディレクトリに配置
2. p4ds2023\_LoadData\_PostgreSQL\_v[version].sql を開き、データの保存先パスを環境に合わせて変更
3. スクリプトを実行してデータをロードする

スクリプト（例）

```
\i /path/to/p4ds2023_LoadData_PostgreSQL_v[version].sql
```

※ /path/to/ は、ご自身の環境に合わせて適宜変更してください。

## 7 データ分析

### 7.1 Python を使う

- お手元の Python 分析環境からデータベースに接続して、Pandas, matplotlib, seaborn などで行って分析と可視化を行ってください。

### 7.2 BI ツールを使う（Tableau の場合）

- Tableau から MySQL / PostgreSQL を利用する方法についての詳細は、Tableau の公式ドキュメントを参照してください。ここでは、カスタム SQL を使用する方法の例を示します。
- e-CSTI では、人材ミスマッチ調査データの分析に際してカスタム SQL を用いた運用を行っており、以下の設定は実際に使用されているものです。

#### Tableau におけるカスタム SQL の例（t\_core\_attribute テーブルを読み込む場合）

- 下記のカスタムSQLでは、t\_core\_attribute テーブルに格納された応答データの各列（code）を読み込み、コード番号に対応するカテゴリー（日本語のラベル）を格納するマスターテーブル（m）を左結合しています。
- カスタムSQLは、テーブル本体と、列に対応するラベルデータを下記のように可読性の高い状態で読み込むことができます。
- カスタムSQLの詳しい使い方は、Tableau の公式ドキュメント等を参照してください。

```
SELECT
    core.year,
    core.response_id,
    core.gender_code,
    core.university_major_code,
    core.age_code,
    core.employment_status_code,
    core.number_of_employees_code,
    core.industry_code,
    core.job_title_code,
    core.annual_income_code,
    core.highest_education_level_code,
    gender.gender_jp,
    major.university_major_jp,
    age.age_jp,
    age.age_value,
    employment.employment_status_jp,
    employees.number_of_employees_jp,
    industry.industry_jp,
    job.job_title_jp,
    income.annual_income_jp,
    income.annual_income_value,
    education.highest_education_level_jp
FROM
    t_core_attribute core
LEFT JOIN m_gender gender
    ON core.gender_code = gender.gender_code
LEFT JOIN m_university_major major
```

```
    ON core.university_major_code = major.university_major_code
LEFT JOIN m_age age
    ON core.year = age.year AND core.age_code = age.age_code
LEFT JOIN m_employment_status employment
    ON core.year = employment.year AND core.employment_status_code =
employment.employment_status_code
LEFT JOIN m_number_of_employees employees
    ON core.year = employees.year AND core.number_of_employees_code =
employees.number_of_employees_code
LEFT JOIN m_industry industry
    ON core.year = industry.year AND core.industry_code = industry.industry_code
LEFT JOIN m_job_title job
    ON core.year = job.year AND core.job_title_code = job.job_title_code
LEFT JOIN m_annual_income income
    ON core.year = income.year AND core.annual_income_code =
income.annual_income_code
LEFT JOIN m_highest_education_level education
    ON core.year = education.year AND core.highest_education_level_code =
education.highest_education_level_code;
```

202503271600