

w585●従業員管理データベース構築

w585●従業員管理データベース構築

PostgreSQLのmrpデータベースにEMPLOYEESテーブルを作成します。

1. データベースサーバ(PostgreSQL)に接続

- Windowsクライアントのcseでmrpデータベースに接続確認

 cse.exe

Windowsクライアントの「cse161フォルダ」にあるcse.exeをダブルクリックを起動して「mrpデータベース」に接続します。



接続

DBMS: PostgreSQL

データソース:

ユーザ名: postgres

パスワード: *****

サーバー名: localhost

DB名: mrp

ポート番号: 5432

オプション:

標準出力先:

UNIX/ケット:

圧縮プロトコルを使用(O)

※未入力項目にはデフォルトの値が適用されます

OK

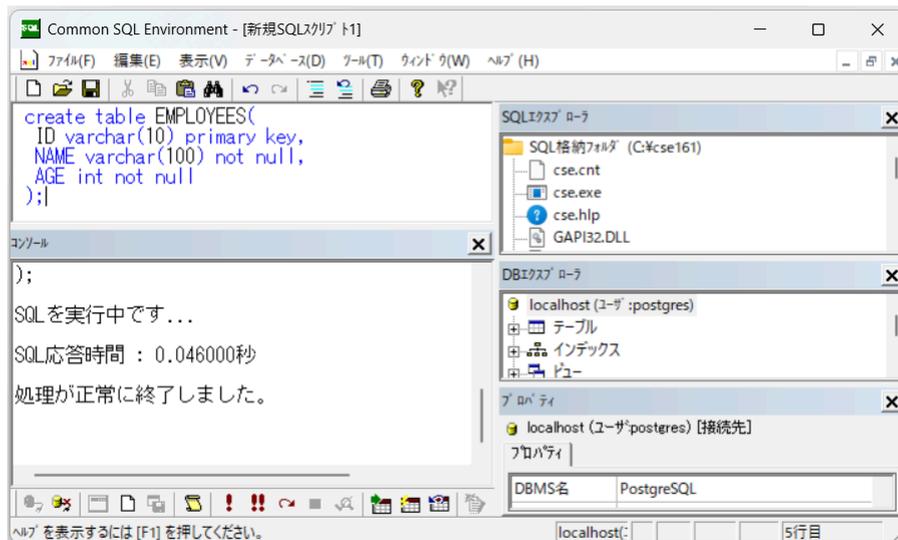
キャンセル

接続リスト

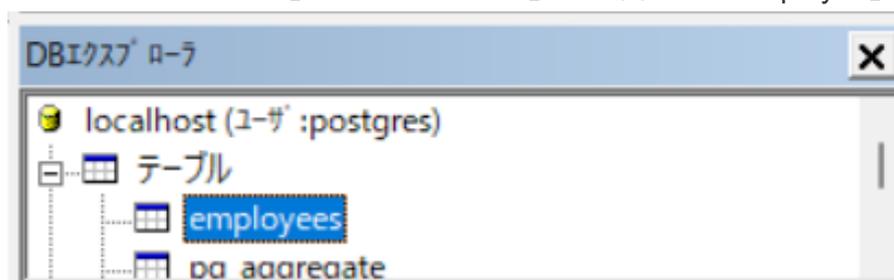
2. テーブルの作成とデータ入力

- WindowsクライアントのcseからEMPLOYEESテーブルを作成

以下のSQL-DDLを入力して ! コマンドで実行します。

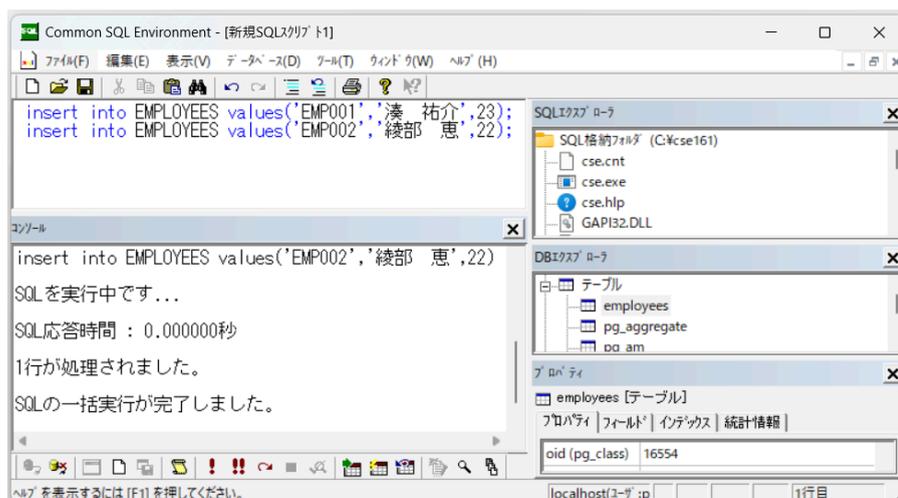


「DBエクプローラ」から「テーブル」を展開して「employee」テーブルの存在を確認します。



• WindowsクライアントのCSEからEMPLOYEEテーブルに初期データを入力

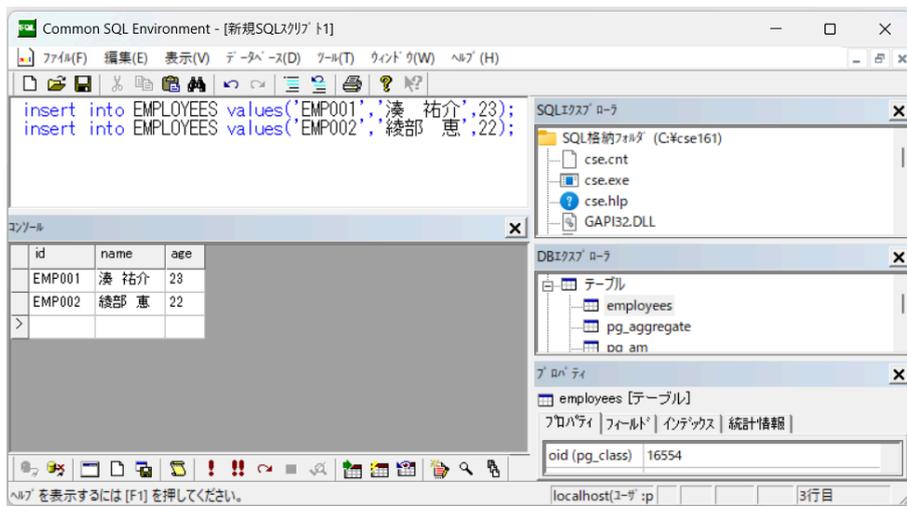
以下のSQL-DMLを入力して !!コマンドで実行します。



2件のSQLが実行されないといけません→失敗したら一旦データを削除してやり直してください
※データの削除の仕方「delete from employees」

• WindowsクライアントのCSEからEMPLOYEESテーブルのデータを確認

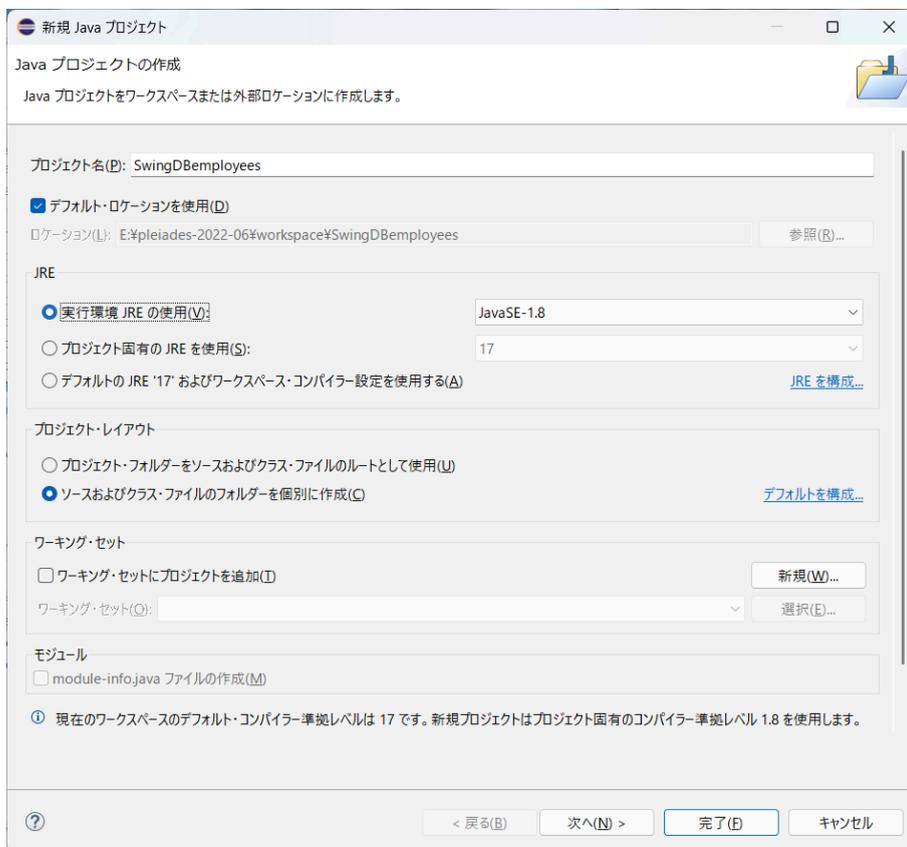
「DBエクプローラ」の「テーブル」内の
「employee」テーブルを右クリックして「全データを開く」のクリックでデータが入力されていることを確認します。



3. SwingDBEmployeeプロジェクトの作成

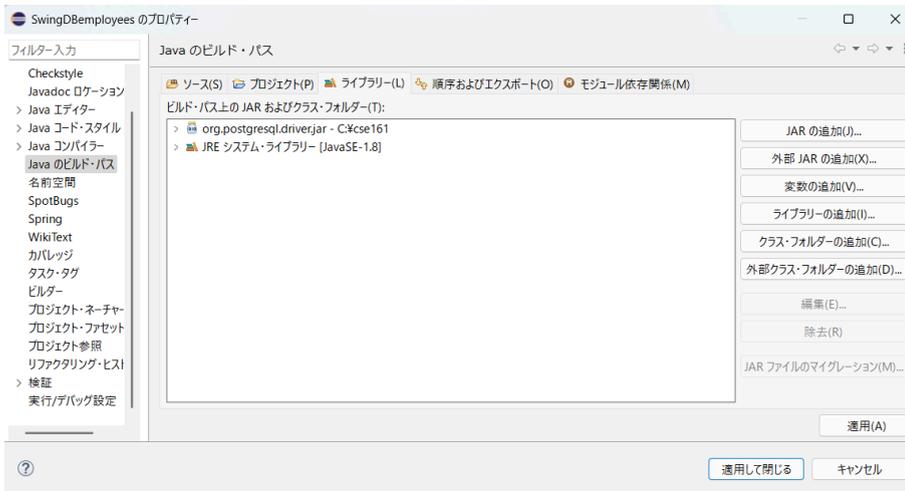
• EclipseよりSwingプロジェクトを作成します

ファイルメニュー → 新規 → Javaプロジェクト



• プロジェクトにDBアクセス用ドライバを登録します

SwingDBEmployeesプロジェクトを右クリック → ビルド・パス → ビルドパスの構成'
→ 「ライブラリー」タブ → 外部JARの追加 → c:¥cse161内のorg.potgresql.driver.jarを選択
→ 開く → 適用して閉じる

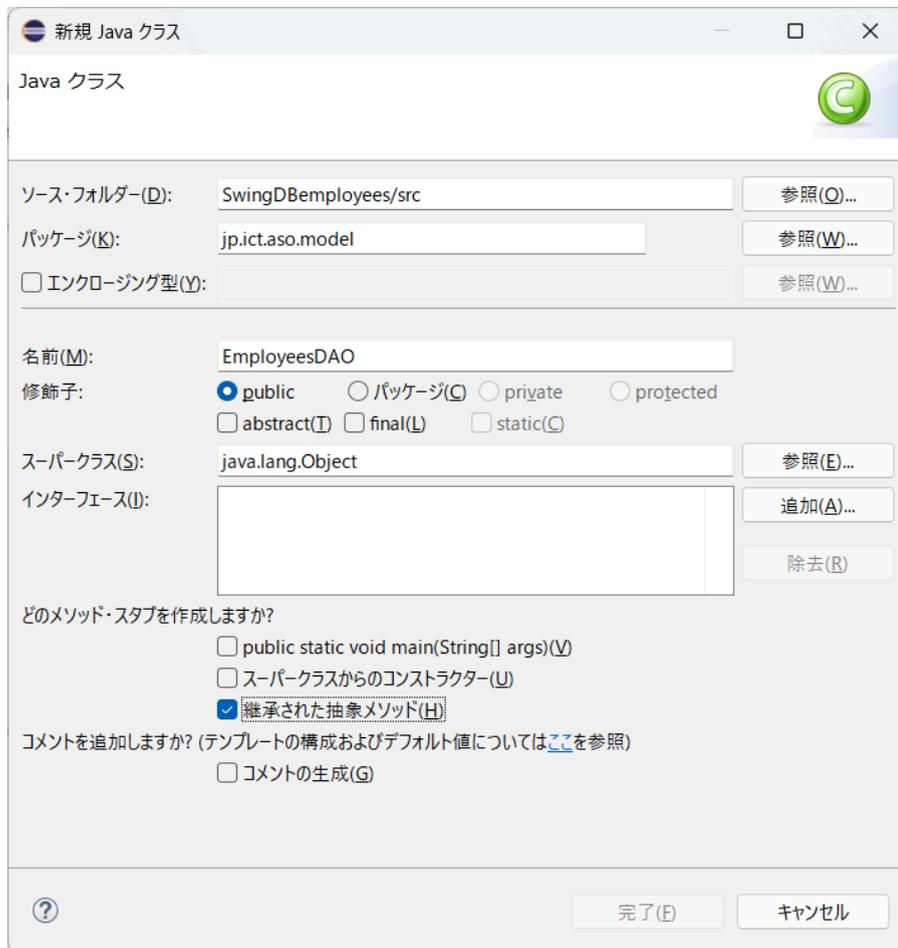


4. Javaプログラム版DBアクセスプログラムを作成

SwingDBEmployeesプロジェクトを右クリック → 新規 → クラス

パッケージ：jp.ict.aso.model

名前：EmployeesDAO



【ソースコード】

```
package jp.ict.aso.model;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```

import java.sql.ResultSet;
import java.sql.SQLException;

public class EmployeesDAO {
    // データベース接続に使用する情報
    private final String JDBC_URL = "jdbc:postgresql://localhost/mrp";
    private final String DB_USER = "postgres";
    private final String DB_PASS = "postgres";
    private final String DB_DRIVER = "org.postgresql.Driver";
    private String resultSet;

    public void dataList(){
        // 結果はhtml形式で保存するためヘッダーを書き込む
        resultSet="<table border=1><tr><th width=150>ID</th><th width=150>名前</th><th width=150>年齢
</th></tr>";

        // JDBCドライバを読み込む
        try {
            Class.forName(DB_DRIVER);
        } catch (ClassNotFoundException e) {
            throw new IllegalStateException("JDBCドライバを読み込めませんでした");
        }
        // データベースに接続
        try (Connection conn = DriverManager.getConnection(JDBC_URL, DB_USER, DB_PASS)) {

            // SELECT文を準備
            String sql = "SELECT ID,NAME,AGE FROM EMPLOYEES";
            PreparedStatement pStmt = conn.prepareStatement(sql);

            // SELECTを実行し、結果表 (ResultSet) を取得
            ResultSet rs = pStmt.executeQuery();

            // 結果表に格納されたレコードの内容を表示
            while (rs.next()) {
                String id = rs.getString("ID");
                String name = rs.getString("NAME");
                int age = rs.getInt("AGE");

                // 取得したデータを出力
                System.out.println("ID:" + id);
                System.out.println("名前:" + name);
                System.out.println("年齢:" + age + "\n");

                // 取得したデータをhtml形式で変数に保存
                resultSet=resultSet+"<tr><td>"+id+"</td><td>"+name+"</td><td>"+age+"</td></tr>";
            }
        }
    }
}

```

```

    }
    resultSet=resultSet+"</table>";
    } catch (SQLException e) {
    e.printStackTrace();
    }
}
public String getResultSet() {
    return resultSet;
}
public static void main(String[] args) {
    EmployeesDAO ed=new EmployeesDAO();
    ed.dataList();
    System.out.println(ed.getResultSet());
}
}

```

【参考】

●PostgreSQLに接続するコード（重要なのはココ）

```

Class.forName("org.postgresql.Driver");
Connection conn = DriverManager.getConnection("jdbc:[postgresql://192.168.56.200:5432/mrp]
(postgresql://192.168.56.10:5432/mrp)", "postgres", "postgres")

```

・DBアクセスプログラムを実行して確認

SelectEmployees.javaをコンパイルして実行します。

プログラムエディタ内で右クリック → 実行 → Javaアプリケーション

左下コンソールに実行結果が出力されます。

ID:EMP001

名前:湊 祐介

年齢:23

ID:EMP002

名前:綾部 恵

年齢:22

```

<table><tr><th>ID</th><th>名前</th><th>年齢</th></tr><tr><td>EMP001</td><td>湊 祐介</td>
<td>23</td></tr><tr><td>EMP002</td><td>綾部 恵</td><td>22</td></tr></table>

```

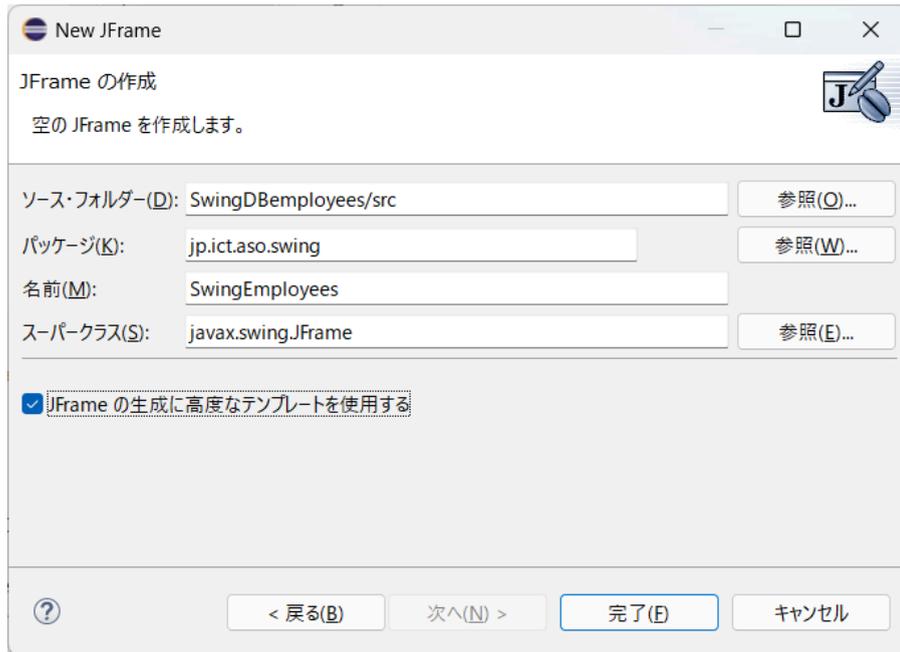
(実行確認後結果を一旦報告してください)

5. Swing版DBアクセスプログラムを作成

SwingDBemployeesプロジェクトを右クリック → 新規 → その他
→ WindowBuilder → Swing デザイナー → JFrame

パッケージ: `jp.ict.aso.swing`

名前: `SwingEmployees`



• デザインタブでGUIを設計する

BorderLayout → contentPane

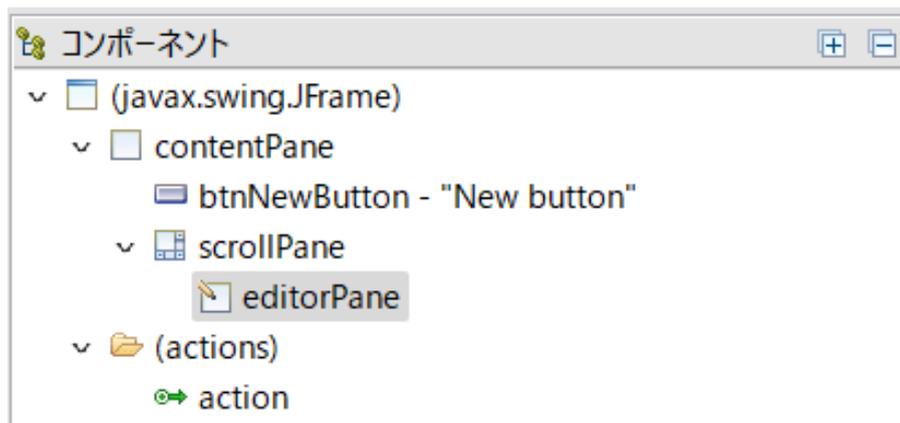
JButton → contentPane

JScrollPane → contentPane → Center

→ HORIZONTAL_SCROLLBAR_ALWAYS

→ VERTICAL_SCROLLBAR_ALWAYSJEditorPane → ViewPoint位置

SwingActions → 新規 → JButton位置



editorPane → contentType を `text/html` に変更

Variable	editorPane
Class	javax.swing.JEditorPane
background	<input type="checkbox"/> 255,255,255 ...
contentType	text/html ...
dropMode	USE_SELECTION
editable	<input checked="" type="checkbox"/> true
enabled	<input checked="" type="checkbox"/> true
font	MS UI Gothic 12 ...
foreground	■ 0,0,0 ...
text	...
toolTipText	...

・ソースタブでActionを実装する

① actionPerformed の内容を追加 (イベント処理の実装)

```
public void actionPerformed(ActionEvent e) {
    EmployeesDAO ed=new EmployeesDAO(); ↓
    ed.dataList(); ↓
    editorPane.setText(ed.getResultSet()); ↓
} ↓
```

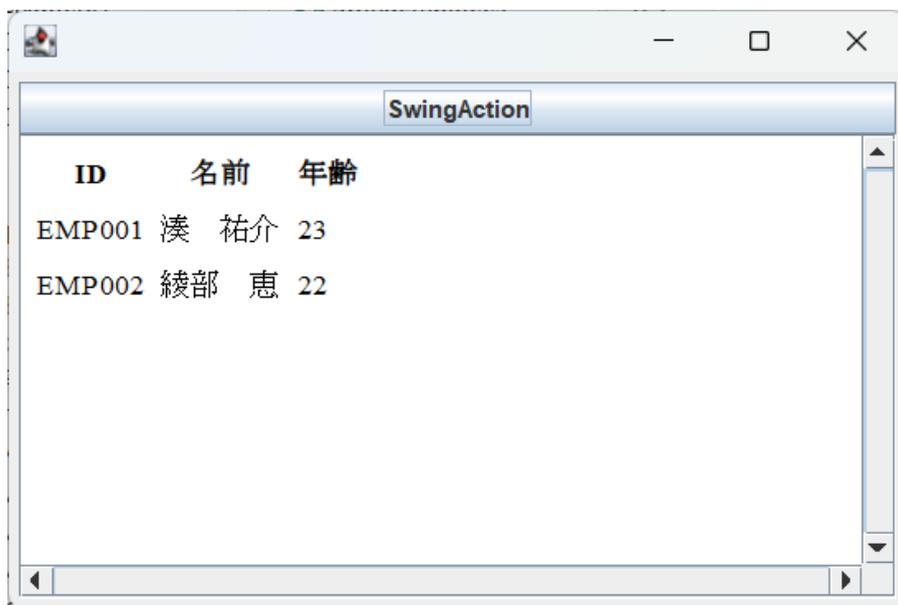
② //koko の部分を追加 (インスタンス変数の宣言追加)

```
private JPanel contentPane; ↓
private final Action action = new SwingAction()
private JEditorPane editorPane; || //koko ↓
```

③ //koko の部分を変更 (ローカル変数の宣言削除)

```
editorPane = new JEditorPane(); || //koko ↓
editorPane.setContentType("text/html"); ↓
scrollPane.setViewportView(editorPane); ↓
```

・実行してみましょう

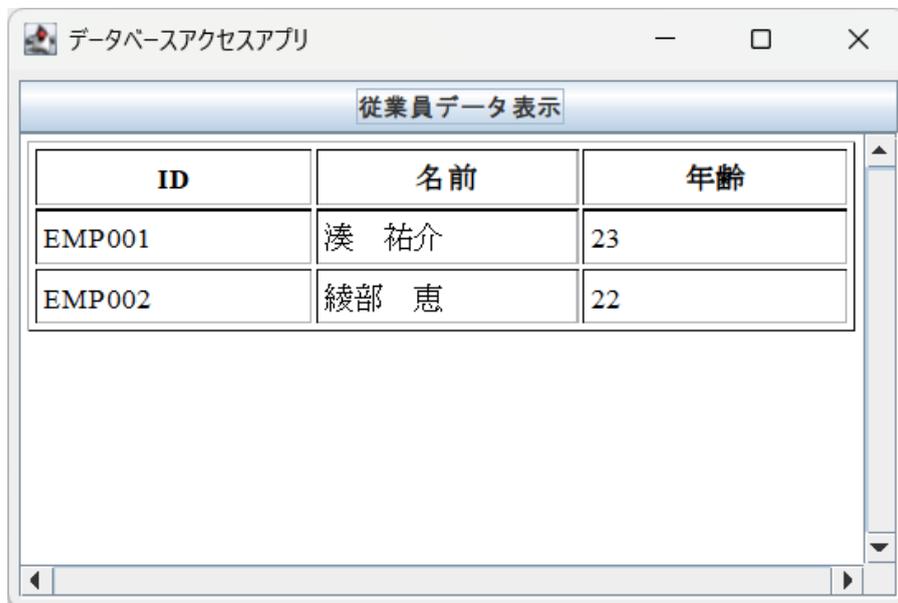


【演習 1】

以下の 4 点を実装してください。

1. フレームのタイトルを設定する
2. ボタンのラベルを設定する
3. データの枠をつける
4. データの表示幅を均等にする

(実行結果)



【演習 2】

データの追加と削除の機能を実装してください。

※SQLのinsert文やdelete文は戻り値のResultSetはありません。よって executeUpdate() メソッドを使います。

追加の例

```
// insert文を準備
String sql = "insert into EMPLOYEES (ID, NAME, AGE) values ('" + id + "','" + name + "','" +
age+"')";
PreparedStatement pstmt = conn.prepareStatement(sql);

// sqlを実行
pstmt.executeUpdate();
```

削除の例

```
// delete文を準備
String sql = "delete from EMPLOYEES where (ID='" + id + "')";
PreparedStatement pstmt = conn.prepareStatement(sql);

// sqlを実行
pstmt.executeUpdate();
```

実行例



ID	名前	年齢
EMP001	湊 祐介	23
EMP002	綾部 恵	22
emp003	福岡 太郎	34

削除