

Link to Github:

<https://github.com/VUGANEZAPATRICE/ihuzoNewAssignment>

The screenshot displays a database management tool interface with a top toolbar containing icons for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Tracking. The main content area shows the results of an SQL execution. A green status bar at the top indicates "1 row inserted. Inserted row id: 2 (Query took 0.0654 seconds.)". Below this, the SQL statement `INSERT INTO Promotion (employeeid) VALUES (1);` is shown, followed by links for [ Edit inline ], [ Edit ], and [ Create PHP code ]. A yellow warning bar states: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available." Below the warning, another green status bar shows "Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)". The SQL statement `SELECT e.name as Employees FROM Employees e LEFT JOIN Promotion p ON e.id = p.employeeId WHERE p.id IS NULL;` is displayed, with links for [ Edit inline ], [ Edit ], [ Explain SQL ], [ Create PHP code ], and [ Refresh ]. Below the SQL, there are controls for "Show all", "Number of rows" (set to 25), "Filter rows" (with a search box "Search this table"), and "Sort by key" (set to None). A section titled "Extra options" is visible, containing a table with the following data:

Employees
Henry
Max

At the bottom left, there is a "Console" tab.

## **MYSQL**

```
CREATE DATABASE Assign_database;
```

```
USE Assign_database;
```

```
CREATE TABLE employees (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(255)  
);
```

```
CREATE TABLE Promotion (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  employeeid INT,  
  FOREIGN KEY (employeeid) REFERENCES Employees(id)  
);
```

```
INSERT INTO employees (name) VALUES ("Joe");  
INSERT INTO employees (name) VALUES ("Henry");  
INSERT INTO employees (name) VALUES ("Sam");  
INSERT INTO employees (name) VALUES ("Max");
```

```
INSERT INTO Promotion (employeeid) VALUES (3);  
INSERT INTO Promotion (employeeid) VALUES (1);
```

```
SELECT e.name as Employees  
FROM employees e  
LEFT JOIN Promotion p ON e.id = p.employeeId  
WHERE p.id IS NULL
```

# PYTHON SQLITE3

```
import sqlite3

conn = sqlite3.connect('Assign_database')
c = conn.cursor()

c.execute('''
    CREATE TABLE IF NOT EXISTS employees (
        id INT AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(255)
    )

''')
c.execute('''
    CREATE TABLE IF NOT EXISTS promotion (
        id INT AUTO_INCREMENT PRIMARY KEY,
        employeeid INT,
        FOREIGN KEY (employeeid) REFERENCES Employees(id)
    )

''')
# c.execute('''
#     CREATE TABLE IF NOT EXISTS prices
#     ([product_id] INTEGER PRIMARY KEY, [price] INTEGER)
# ''')
# c.execute('''INSERT INTO employees (name) VALUES ("Joe")''')
# c.execute('''INSERT INTO employees (name) VALUES ("Henry")''')
# c.execute('''INSERT INTO employees (name) VALUES ("Sam")''')
# c.execute('''INSERT INTO employees (name) VALUES ("Max")''')
# c.execute('''INSERT INTO employees (name) VALUES ("Patrice Vuganeza")''')

# c.execute('''INSERT INTO promotion (employeeid) VALUES ("Joe")''')
# c.execute('''INSERT INTO promotion (employeeid) VALUES ("Joe")''')
c.execute('''
    SELECT e.name as employees
    FROM employees e
    LEFT JOIN promotion p ON e.id = p.employeeId
    WHERE p.id IS NULL
''')

conn.commit()
```