Database Management System With Redis and PHP

CS 51550, Fall 2024
Purdue University Northwest
12/08/2024.

Name	**Email**	**Github**
Shivam Pandya	pandya23@pnw.edu	https://github.com/shivampandya67/MYSQL-to-Redis-Implementation
Vishva Gor	vgor@pnw.edu	https://github.com/Vishvagor/redis-php-management-system

DemoVideo:

Importing MySQL Data Across Containers via Python to Redis| PHP Web Pa…

1. Abstract

This project demonstrates the development of a web-based Department and Employee Management System using **Redis** as the database backend and **PHP** for server-side scripting. The application, deployed in a **Dockerized environment**, facilitates the efficient management of organizational data such as department information, employee details, projects, and department locations. The project highlights the benefits of in-memory databases like Redis for real-time data retrieval, replacing traditional RDBMS systems like MySQL.

2. Introduction

In the modern era of web applications, scalability and performance are critical for efficiently managing large volumes of data. Traditional relational database management systems (RDBMS) like MySQL have served as reliable backends for decades. However, the emergence of NoSQL databases, such as Redis, has revolutionized data handling by offering in-memory storage for ultra-fast read and write operations. Redis stands out due to its ability to handle diverse data types, such as strings, hashes, lists, and sets, making it ideal for real-time applications.

This project explores the transition of a Department and Employee Management System from MySQL to Redis, demonstrating Redis's effectiveness in managing structured and semi-structured data. The application provides a user-friendly interface to perform various operations, including viewing department details, fetching employee information by SSN, listing all departments, and displaying projects associated with departments. The system leverages Redis's hashing capabilities to store and retrieve data efficiently.

The project is deployed in a Dockerized environment to ensure portability and scalability, where separate containers manage Redis and PHP. This approach encapsulates each service, minimizing compatibility issues and streamlining development workflows. The Docker setup enhances the deployment process, ensuring consistent performance across different environments.

The project aims to highlight Redis's capability as a high-performance database solution while maintaining the functionality and integrity of the original application. By integrating Redis with PHP, the project bridges the gap between cutting-edge NoSQL technologies and traditional web application frameworks, demonstrating the potential of modern database systems in organizational data management.

3. Project Objectives

1. The transition from an existing MySQL-based management system to Redis.
2. Develop a Dockerized environment with separate containers for PHP and Redis.
3. Implement web interfaces to:
    - List all departments.
    - View detailed information about a specific department.
    - Fetch and display employee details using their SSN.
4. Ensure scalability, efficiency, and real-time data retrieval using Redis.
5. Overcome challenges related to data migration and application redesign.

4. ER Diagram

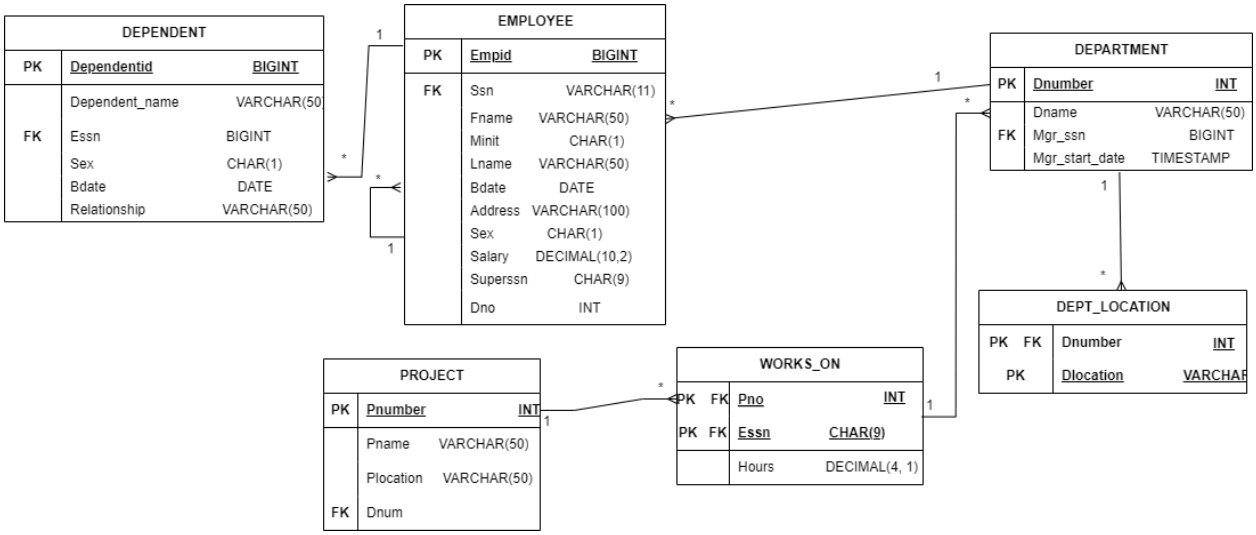


Figure 1: ER Diagram

The ERD (Entity-Relationship Diagram) models a company's organizational database structure. It includes key entities such as **EMPLOYEE**, **DEPARTMENT**, **PROJECT**, **DEPENDENT**, and **WORKS_ON**, showing their attributes, primary keys, and relationships:

- **EMPLOYEE** manages employee details and connects to departments (`Dno`) and supervisors (`Superssn`).
- **DEPARTMENT** tracks department details and has a one-to-many relationship with employees, projects, and department locations.
- **PROJECT** links to departments and has a many-to-many relationship with employees via the `WORKS_ON` table.
- **DEPENDENT** stores employee-dependent information, with a foreign key to the employee's `Essn`.
- Composite keys are used for tables like `WORKS_ON` and `DEPT_LOCATION` to manage relationships efficiently.

This ERD ensures data consistency using primary and foreign keys and models real-world organizational scenarios effectively.

5. Implementation

5.1 Dockerized Environment

Dockerized Environment:

- The application is deployed using Docker Compose with two containers:
  - PHP container: Hosts the web application.
  - Redis container: Manages the in-memory database.

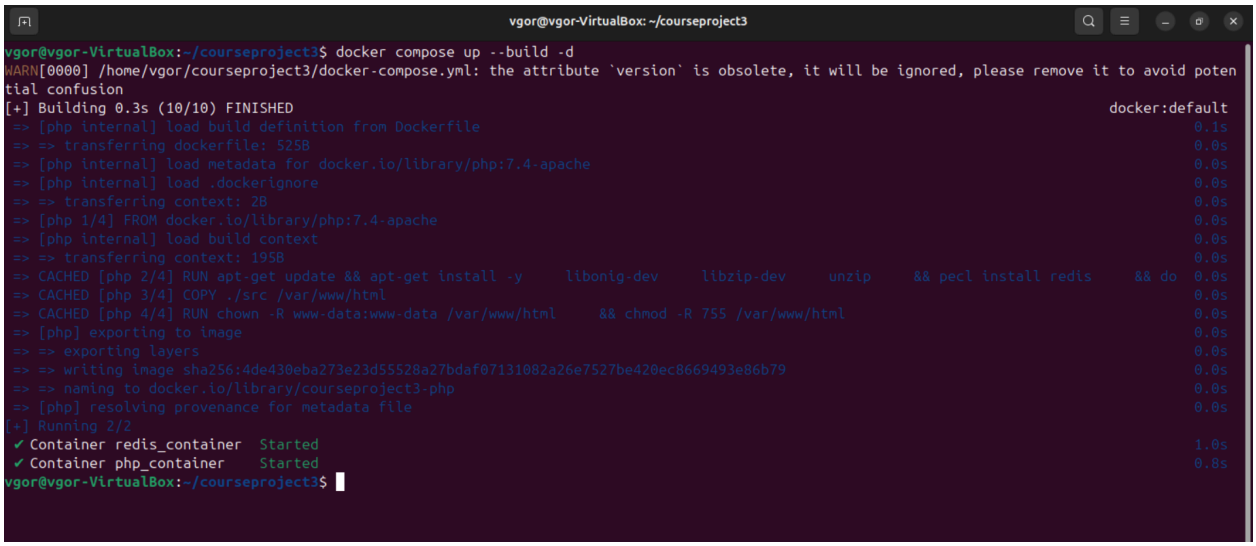


Figure 2: Docker Setup

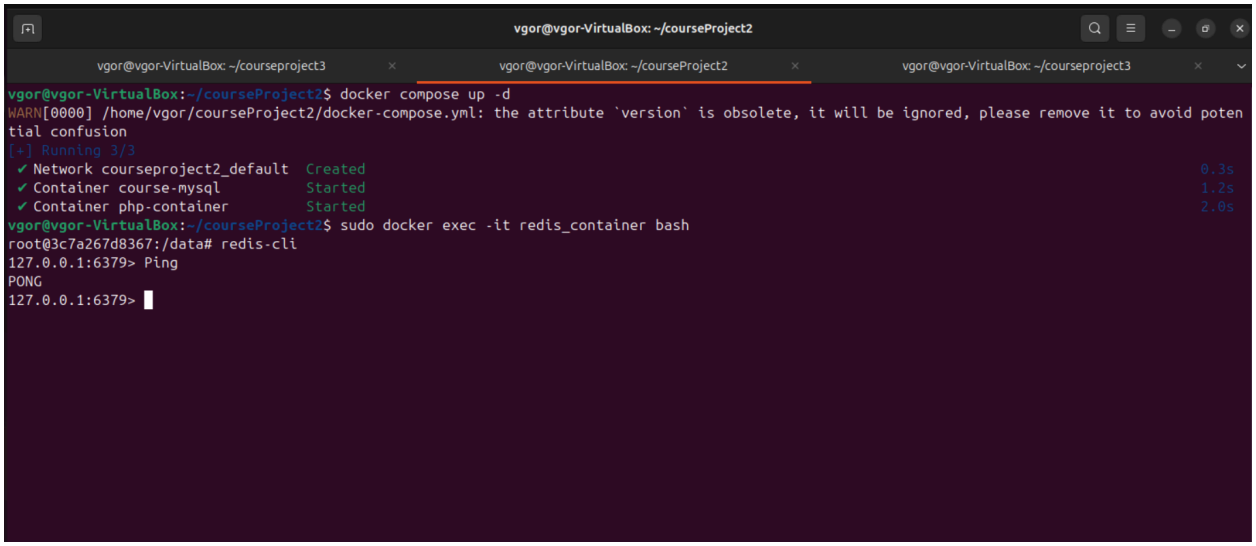


Figure 3: Redis service is running

5.2 Data Migration

Data was extracted from MySQL and stored in Redis using a Python script. The Python script serves as a bridge between MySQL and Redis. To achieve this, we built a MySQL container and a Redis container. We then executed the Python script, which successfully extracted data from MySQL and loaded it into Redis.

The project employs a custom Python script to efficiently migrate data from MySQL to Redis. This script extracts structured data from MySQL tables, processes it to Redis-compatible formats, and stores it using Redis's hash data structure. Each table in MySQL is converted into a series of Redis keys with hash mappings for the respective rows.

Special care was taken to handle MySQL data types such as `DATE`, `DECIMAL`, and `NULL`, converting them into Redis-friendly formats. For instance, `DATE` values were transformed into ISO-compliant strings, and `DECIMAL` values were stored as floating-point numbers to maintain precision.

The migration process ensures data consistency and integrity, maintaining relationships between entities like employees, departments, and projects. By adopting a structured key-naming convention (e.g., `EMPLOYEE:<Ssn>` and

DEPARTMENT:<Dnumber>), the system facilitates intuitive and efficient data retrieval in Redis.

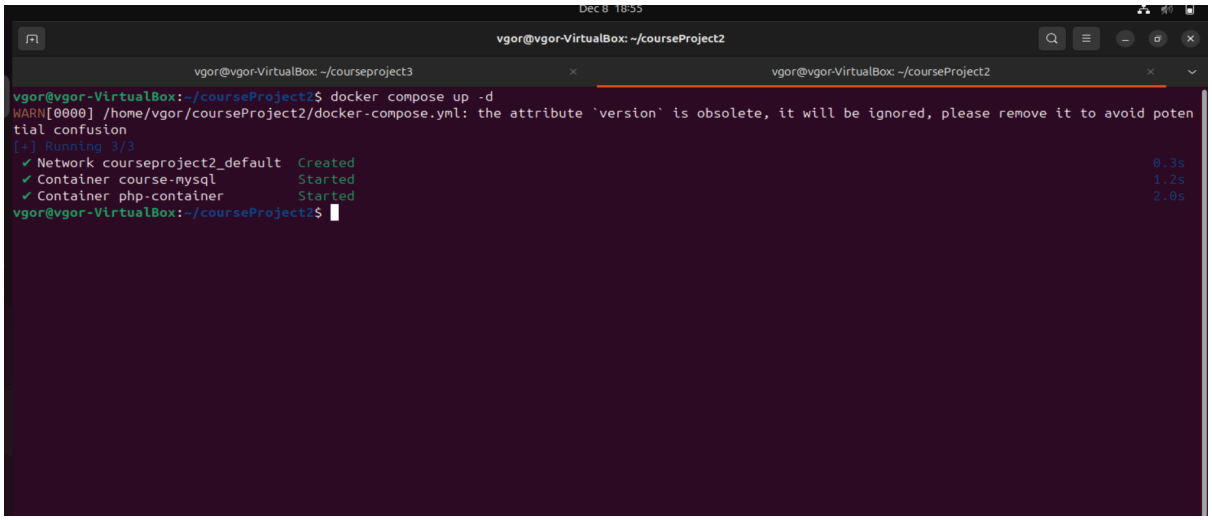
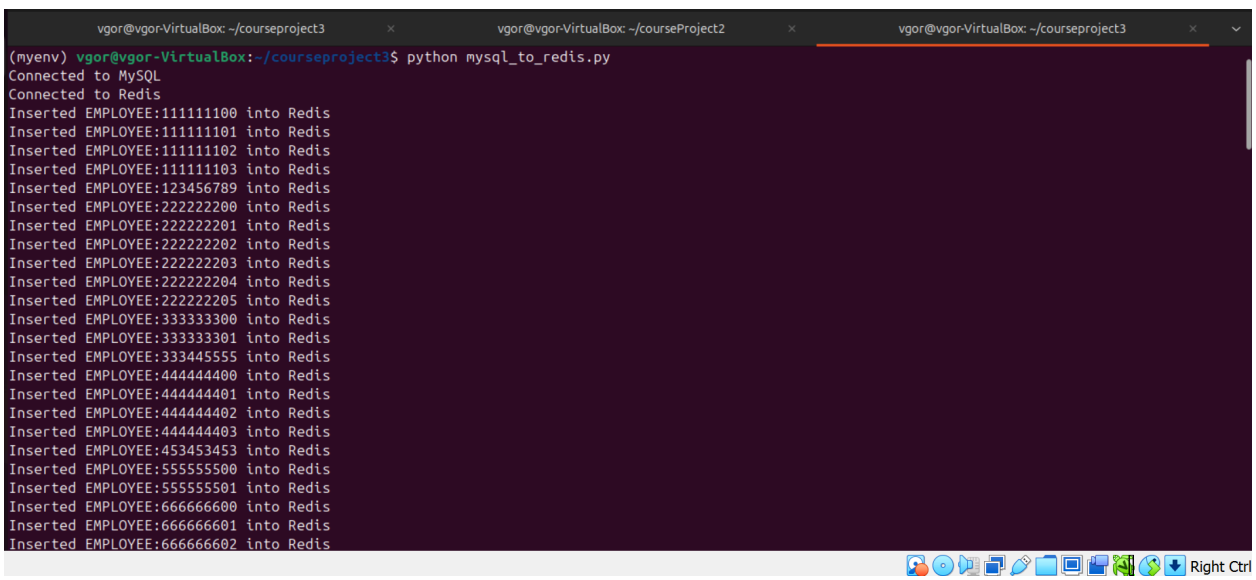


Figure 4: MYSQL container building



```
vgor@vgor-VirtualBox: ~/courseproject3
vgor@vgor-VirtualBox: ~/courseproject3    vgor@vgor-VirtualBox: ~/courseProject2    vgor@vgor-VirtualBox: ~/courseproject3
Inserted WORKS_ON:555555500:92 into Redis
Inserted WORKS_ON:555555501:92 into Redis
Inserted WORKS_ON:666666601:91 into Redis
Inserted WORKS_ON:666666603:91 into Redis
Inserted WORKS_ON:666666604:91 into Redis
Inserted WORKS_ON:666666605:92 into Redis
Inserted WORKS_ON:666666606:91 into Redis
Inserted WORKS_ON:666666607:61 into Redis
Inserted WORKS_ON:666666608:62 into Redis
Inserted WORKS_ON:666666609:63 into Redis
Inserted WORKS_ON:666666610:61 into Redis
Inserted WORKS_ON:666666611:61 into Redis
Inserted WORKS_ON:666666612:61 into Redis
Inserted WORKS_ON:666666613:61 into Redis
Inserted WORKS_ON:666666613:62 into Redis
Inserted WORKS_ON:666666613:63 into Redis
Inserted WORKS_ON:666884444:3 into Redis
Inserted WORKS_ON:888665555:20 into Redis
Inserted WORKS_ON:987654321:20 into Redis
Inserted WORKS_ON:987654321:30 into Redis
Inserted WORKS_ON:987987987:10 into Redis
Inserted WORKS_ON:987987987:30 into Redis
Inserted WORKS_ON:999887777:10 into Redis
Inserted WORKS_ON:999887777:30 into Redis
Data loaded into Redis successfully.
(myenv) vgor@vgor-VirtualBox:~/courseproject3$
```

Figure 5: Running Python script

```
127.0.0.1:6379> HGETALL EMPLOYEE:123456789
 1) "Fname"
 2) "John"
 3) "Minit"
 4) "B"
 5) "Lname"
 6) "Smith"
 7) "Ssn"
 8) "123456789"
 9) "Bdate"
10) "1955-01-09"
11) "Address"
12) "731 Fondren, Houston, TX"
13) "Sex"
14) "M"
15) "Salary"
16) "30000.0"
17) "Super_ssn"
18) "333445555"
19) "Dno"
20) "5"
127.0.0.1:6379>
```

Figure 6: Data into Redis

5.3 PHP Integration

Implemented index.php, p1.php, deptView.php, and companyBrowse.php to interact with Redis and provide a seamless user interface.

Additionally, the PHP integration in the project highlights the versatility and simplicity of combining Redis with web technologies. Each PHP file is designed with a specific function to interact with Redis, utilizing its efficient data retrieval mechanisms. The files collectively form a comprehensive system:

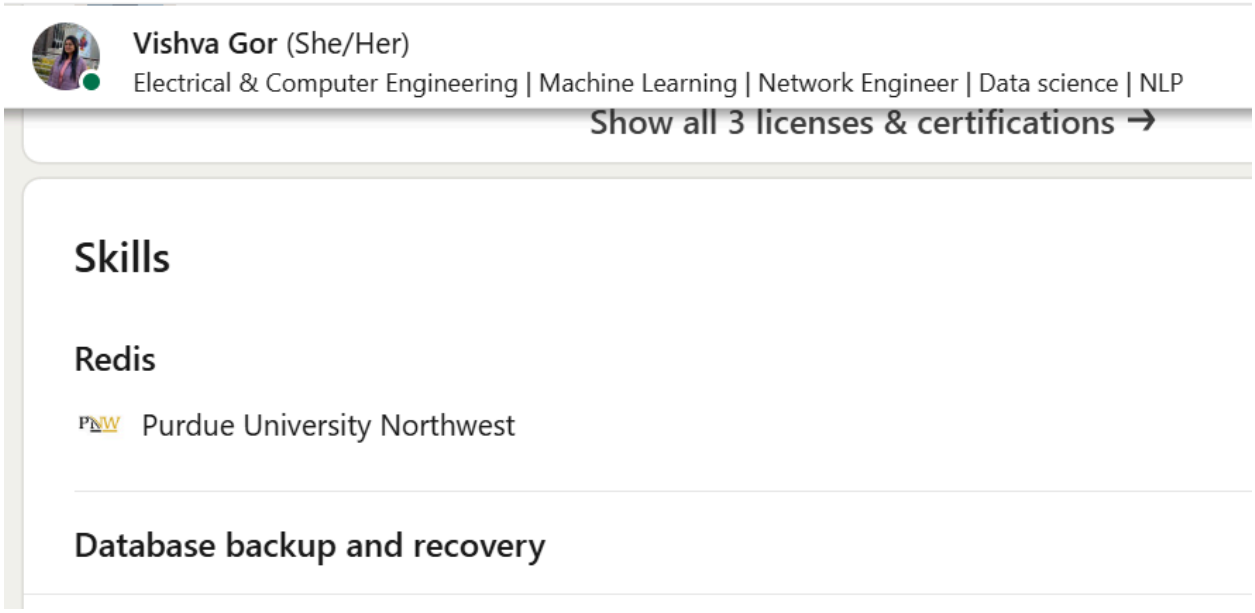
- **index.php** provides an entry point where users can select employees from a dropdown populated dynamically with data from Redis.
- **p1.php** processes the employee selection and retrieves detailed information such as first name, middle initial, and last name by querying Redis keys and values.
- **deptView.php** offers a detailed view of a department, including its associated manager, employees, projects, and locations, all fetched efficiently using Redis hashes.
- **companyBrowse.php** displays a list of all departments, providing links to detailed department views, leveraging Redis's rapid key search capabilities.

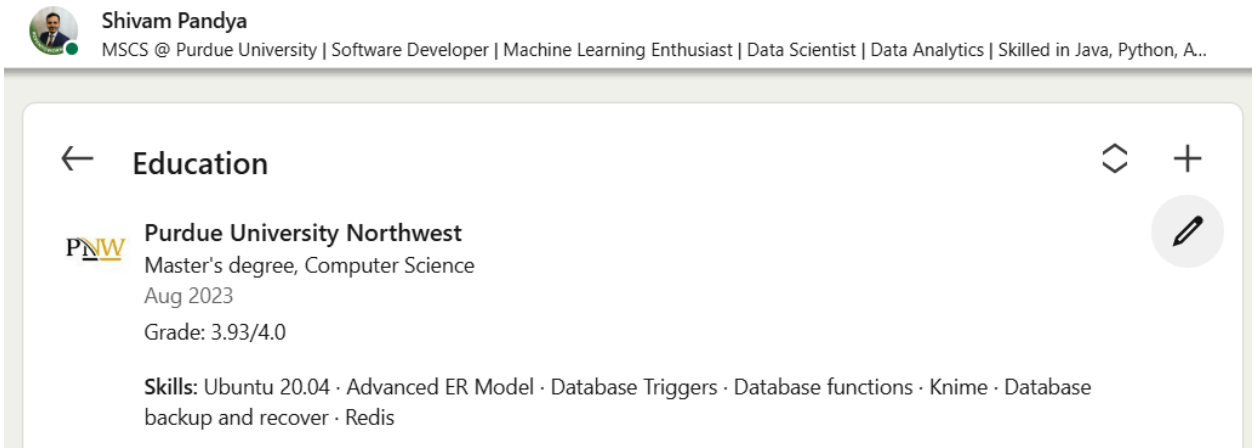
5.4 Web Pages

- **index.php**: Lists employees with their SSNs in a dropdown.
- **p1.php**: Displays detailed employee information.
- **deptView.php**: Shows department details, projects, and employees linked to a specific department.
- **companyBrowse.php**: Lists all departments in tabular form.

Figure 7: An Example of a web page

5.5 LinkedIn Update





6. Challenges

Data Migration:

- Converting MySQL data types (e.g., DATE, DECIMAL) to Redis-compatible formats required meticulous handling.

Redis Integration:

- Initial challenges in establishing connections between the Redis container and PHP were resolved by debugging Redis configurations and PHP Redis extensions.

Application Redesign:

- Transitioning SQL-based queries to Redis's key-value model necessitated significant code refactoring.

7. Discussion

The project successfully demonstrates Redis's capability to replace traditional RDBMS in scenarios requiring high-speed data access. The use of Docker ensures portability and isolated development environments, making the system easy to deploy. By adopting Redis, the application achieves:

- Real-time data retrieval.
- Simplified scalability for large datasets.
- Reduced latency in web application responses.

However, Redis's lack of native relational capabilities posed challenges during implementation. The application mitigated these challenges by structuring keys and data appropriately.

8. Conclusion

This project highlights Redis's potential as a robust and efficient alternative to traditional databases for web applications. The migration from MySQL to Redis was successfully achieved while retaining the application's core functionality. By integrating Docker, Redis, and PHP, the project serves as a modern, scalable solution for organizational data management.

9. Acknowledgement

I would like to express my sincere gratitude to **Professor Dr. David Dai** for his invaluable guidance, support, and encouragement throughout this project. His insights and feedback were instrumental in shaping the success of this endeavor.

I would also like to acknowledge my teammate **Vishva**, whose collaboration, dedication, and efforts were vital in overcoming the technical challenges and ensuring the smooth completion of this project. This achievement would not have been possible without his contributions and teamwork.

Lastly, I am grateful for the resources and documentation provided by the Redis and PHP communities, which greatly assisted in the implementation and troubleshooting of this system.

10. Appendix

10.1 Docker commands

```
version: '3.8'

services:
  redis:
    image: redis:latest
    container_name: redis_container
    ports:
      - "6379:6379"

  php:
    build:
      context: .  # Use the directory with the Dockerfile
    container_name: php_container
    ports:
      - "8081:80"  # Change the port here (PHP will now be accessible on port 8081)
    volumes:
      - ./src:/var/www/html
    depends_on:
      - redis
    environment:
      REDIS_HOST: redis
```

- Build and start containers:

docker-compose up --build -d

- Stop containers:

docker-compose down

10.2 Python Code

```
import mysql.connector
import redis
from datetime import datetime, date
```

```
from decimal import Decimal

# MySQL connection setup
mysql_config = {
    'host': 'localhost',          # MySQL container hostname (or 'localhost' if running locally)
    'port': 3307,                 # Port mapped to MySQL container
    'user': 'root',
    'password': 'rootpassword',   # MySQL root password
    'database': 'mydatabase'      # The database name
}

# Redis connection setup
redis_conn = redis.Redis(host='localhost', port=6379, decode_responses=True)

# Function to convert MySQL data types into Redis-compatible formats
def convert_mysql_to_redis(row):
    for key, value in row.items():
        if isinstance(value, (datetime, date)):  # Handle both datetime and date
            row[key] = str(value)  # Convert datetime/date to string
        elif isinstance(value, Decimal):  # Handle DECIMAL type (MySQL)
            row[key] = float(value)  # Convert DECIMAL to float
        elif isinstance(value, float):  # For DECIMAL type, store as float
            row[key] = float(value)
        elif isinstance(value, int):  # For INT type, store as int (already valid)
            row[key] = int(value)
        elif isinstance(value, str):  # For VARCHAR type, ensure it's a string (valid in Redis)
            row[key] = str(value)
        elif value is None:  # Handle NULL values (MySQL NULL values get converted to "NULL"
string in Redis)
            row[key] = "NULL"
    return row

# Connect to MySQL
try:
    mysql_conn = mysql.connector.connect(**mysql_config)
    mysql_cursor = mysql_conn.cursor(dictionary=True)
    print("Connected to MySQL")

    # Check Redis connection
    try:
        redis_conn.ping()  # Ping Redis to check the connection
        print("Connected to Redis")
    except redis.ConnectionError:
        print("Failed to connect to Redis")
```

```
    # Fetch data from MySQL and load into Redis

    # Load EMPLOYEE table into Redis
    mysql_cursor.execute("SELECT * FROM EMPLOYEE")
    employees = mysql_cursor.fetchall()

    for employee in employees:
        employee = convert_mysql_to_redis(employee)  # Convert MySQL types to
Redis-compatible types
        key = f"EMPLOYEE:{employee['Ssn']}"  # Use SSN as the Redis key
        redis_conn.hset(key, mapping=employee)
        print(f"Inserted EMPLOYEE:{employee['Ssn']} into Redis")

    # Load DEPARTMENT table into Redis
    mysql_cursor.execute("SELECT * FROM DEPARTMENT")
    departments = mysql_cursor.fetchall()
    for department in departments:
        department = convert_mysql_to_redis(department)  # Convert MySQL types
        key = f"DEPARTMENT:{department['Dnumber']}"  # Use department number as the Redis
key
        redis_conn.hset(key, mapping=department)
        print(f"Inserted DEPARTMENT:{department['Dnumber']} into Redis")

    # Load DEPENDENT table into Redis
    mysql_cursor.execute("SELECT * FROM DEPENDENT")
    dependents = mysql_cursor.fetchall()
    for dependent in dependents:
        dependent = convert_mysql_to_redis(dependent)  # Convert MySQL types
        key = f"DEPENDENT:{dependent['Essn']}:{dependent['Dependent_name']}"  # Use SSN
and dependent name as the Redis key
        redis_conn.hset(key, mapping=dependent)
        print(f"Inserted DEPENDENT:{dependent['Essn']}:{dependent['Dependent_name']} into
Redis")

    # Load DEPT_LOCATION table into Redis
    mysql_cursor.execute("SELECT * FROM DEPT_LOCATION")
    dept_locations = mysql_cursor.fetchall()
    for dept_location in dept_locations:
        dept_location = convert_mysql_to_redis(dept_location)  # Convert MySQL types
        key = f"DEPT_LOCATION:{dept_location['Dnumber']}:{dept_location['Location']}"  # Use
Dnumber and Location as the Redis key
        redis_conn.hset(key, mapping=dept_location)
```

```
        print(f"Inserted DEPT_LOCATION:{dept_location['Dnumber']}:{dept_location['Location']}
into Redis")

    # Load PROJECT table into Redis
    mysql_cursor.execute("SELECT * FROM PROJECT")
    projects = mysql_cursor.fetchall()
    for project in projects:
        project = convert_mysql_to_redis(project)  # Convert MySQL types
        key = f"PROJECT:{project['Pnumber']}"  # Use project number as the Redis key
        redis_conn.hset(key, mapping=project)
        print(f"Inserted PROJECT:{project['Pnumber']} into Redis")

    # Load WORKS_ON table into Redis (Updated)
    mysql_cursor.execute("SELECT * FROM WORKS_ON")
    works_on = mysql_cursor.fetchall()
    for work in works_on:
        work = convert_mysql_to_redis(work)  # Convert MySQL types
        key = f"WORKS_ON:{work['Essn']}:{work['Pno']}"  # Use Essn and Pno as the Redis key
        redis_conn.hset(key, mapping=work)
        print(f"Inserted WORKS_ON:{work['Essn']}:{work['Pno']} into Redis")

    print("Data loaded into Redis successfully.")
    mysql_cursor.close()
    mysql_conn.close()

except Exception as e:
    print(f"Error: {e}")
```

10.3 Redis commands

Sudo docker exec -it redis_container bash

Redis-cli

HGETALL EMPLOYEE:123456789

10.4 PHP Code

- Index.php

```
<!DOCTYPE html>
```

```html
<html>
<head>
    <title>Select Employee SSN</title>
    <style>
        body {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            font-family: Arial, sans-serif;
            background-color: #f4f4f9;
            margin: 0;
        }
        .container {
            text-align: center;
            background-color: #ffffff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            max-width: 400px;
            width: 100%;
        }
        h3 {
            color: #333;
            margin-bottom: 20px;
        }
        select {
            padding: 8px;
            width: 100%;
            font-size: 16px;
            border: 1px solid #ccc;
            border-radius: 4px;
            margin-bottom: 15px;
        }
        input[type="submit"] {
```

```
            padding: 10px 15px;
            font-size: 16px;
            background-color: #4CAF50;
            color: white;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            width: 100%;
        }
        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>

<div class="container">
    <h3>Employee Details for:</h3>
    <form method="GET" action="p1.php">
        <label for="ssn">Select SSN:</label>
        <select name="ssn" id="ssn" required>
            <?php
                // Connect to Redis
                $redis = new Redis();
                $redis->connect('redis', 6379);  // Redis container hostname

                // Fetch all employee keys
                $keys = $redis->keys('EMPLOYEE:*');  // Get all employee SSNs

                if (empty($keys)) {
                    echo "<option disabled>No employees found</option>";
                } else {
                    foreach ($keys as $key) {
                        // Extract SSN from the key (EMPLOYEE:<ssn>)
                        $ssn = str_replace('EMPLOYEE:', '', $key);
```

```
                echo "<option value=\"$ssn\">$ssn</option>";
            }
        }
    ?>
    </select>
    <input type="submit" value="Get Employee Details">
  </form>
</div>

</body>
</html>
```

- p1.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Employee Information</title>
  <style>
    body {
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      font-family: Arial, sans-serif;
      background-color: #f4f4f9;
      margin: 0;
    }
    .container {
      text-align: center;
      background-color: #ffffff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
      max-width: 400px;
      width: 100%;
```

```
        }
        h4 {
            color: #333;
            margin-bottom: 20px;
        }
        .employee-info {
            font-size: 18px;
            font-weight: bold;
            color: #333;
            margin-top: 15px;
        }
    </style>
</head>
<body>

<div class="container">
    <h4>Employee Information</h4>
    <div class="employee-info">
        <?php
        try {
            // Connect to Redis
            $redis = new Redis();
            $redis->connect('redis', 6379);  // Use Redis container hostname

            // Check if 'ssn' is passed via GET
            if (isset($_GET['ssn'])) {
                $ssn = $_GET['ssn'];

                // Build the Redis key for the employee
                $key = "EMPLOYEE:" . $ssn;

                // Check if the key exists in Redis
                if ($redis->exists($key)) {
                    // Fetch employee details from Redis
                    $employee = $redis->hGetAll($key);
```

```
                // Display employee information
                echo "<div style='text-align:center; padding: 20px; border: 1px solid
#ddd; border-radius: 5px; width: 300px; margin: auto;'>";
                echo "<h2>Employee Information</h2>";
                echo "Name: " . htmlspecialchars($employee['Fname']) . " " .
                    htmlspecialchars($employee['Minit']) . " " .
                    htmlspecialchars($employee['Lname']) . "<br>";
                echo "SSN: " . htmlspecialchars($employee['Ssn']) . "<br>";
                echo "Date of Birth: " . htmlspecialchars($employee['Bdate']) . "<br>";
                echo "Address: " . htmlspecialchars($employee['Address']) . "<br>";
                echo "Sex: " . htmlspecialchars($employee['Sex']) . "<br>";
                echo "Salary: $" . htmlspecialchars($employee['Salary']) . "<br>";
                echo "</div>";
            } else {
                echo "No employee found with SSN: $ssn in Redis.";
            }
        } else {
            echo "SSN parameter is missing in the request.";
        }
    } catch (Exception $e) {
        // Handle Redis connection error
        echo "Could not connect to Redis: " . htmlspecialchars($e->getMessage());
    }
    ?>
  </div>
</div>

</body>
</html>
```

- p2.php

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Department Employee Details</title>
<style>
    body {
        display: flex;
        justify-content: center;
        align-items: center;
        height: 100vh;
        font-family: Arial, sans-serif;
        background-color: #f4f4f9;
        margin: 0;
    }
    .container {
        text-align: center;
        background-color: #ffffff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        max-width: 400px;
        width: 100%;
    }
    h3, h4 {
        color: #333;
        margin-bottom: 20px;
    }
    form {
        margin-bottom: 15px;
    }
    label {
        font-weight: bold;
        display: block;
        margin-bottom: 5px;
    }
    input[type="number"] {
        padding: 8px;
        width: 100%;
```

```
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
    margin-bottom: 15px;
}
input[type="submit"] {
    padding: 10px 15px;
    font-size: 16px;
    background-color: #4CAF50;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    width: 100%;
}
input[type="submit"]:hover {
    background-color: #45a049;
}
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
th, td {
    padding: 10px;
    border: 1px solid #ddd;
    text-align: left;
}
th {
    background-color: #4CAF50;
    color: white;
}
td {
    background-color: #f9f9f9;
}
```

```
    </style>
</head>
<body>

<div class="container">
    <h3>Enter Department Number</h3>
    <form method="GET" action="p2.php">
        <label for="dno">Department Number:</label>
        <input type="number" id="dno" name="dno" required>
        <input type="submit" value="Get Employee Details">
    </form>

    <?php
    try {
        // Connect to Redis
        $redis = new Redis();
        $redis->connect('redis', 6379);  // Use Redis container hostname

        if (isset($_GET['dno'])) {
            $dno = $_GET['dno'];

            // Debugging: Show the department number
            echo "<h4>Employees in Department $dno</h4>";

            // Redis keys are in the format EMPLOYEE:<ssn>
            $keys = $redis->keys('EMPLOYEE:*');  // Get all employee keys

            $found = false;  // Flag to check if employees are found

            echo "<table>";
            echo "<tr><th>Last Name</th><th>Salary</th></tr>";

            foreach ($keys as $key) {
                // Get the employee data
                $employee = $redis->hGetAll($key);
```

```
            // Check if the employee belongs to the requested department
            if (isset($employee['Dno']) && $employee['Dno'] == $dno) {
                $found = true;
                echo "<tr>";
                echo "<td>" . htmlspecialchars($employee['Lname']) . "</td>";
                echo "<td>$" . htmlspecialchars($employee['Salary']) . "</td>";
                echo "</tr>";
            }
        }

        if (!$found) {
            echo "<p>No employees found in department $dno.</p>";
        }

        echo "</table>";
    } else {
        echo "<p>Please enter a department number above.</p>";
    }
} catch (Exception $e) {
    echo "Could not connect to Redis: " . htmlspecialchars($e->getMessage());
}
?>
</div>

</body>
</html>
```

- deptView.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Department View</title>
    <style>
        body {
```

```
    font-family: Arial, sans-serif;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    background-color: #f4f4f9;
    margin: 0;
}
.container {
    width: 80%;
    max-width: 600px;
    background-color: #ffffff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    text-align: center;
}
h3, h4 {
    color: #333;
    margin-bottom: 15px;
}
form {
    margin-bottom: 20px;
}
label {
    font-weight: bold;
    margin-right: 10px;
}
input[type="number"], input[type="submit"] {
    padding: 8px;
    font-size: 16px;
}
input[type="submit"] {
    background-color: #4CAF50;
    color: white;
```

```
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
input[type="submit"]:hover {
    background-color: #45a049;
}
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
}
th, td {
    padding: 10px;
    text-align: left;
    border: 1px solid #ddd;
}
th {
    background-color: #4CAF50;
    color: white;
}
td a {
    color: #4CAF50;
    text-decoration: none;
}
td a:hover {
    text-decoration: underline;
}
.section-title {
    font-weight: bold;
    color: #555;
    margin-top: 20px;
}
.no-data {
    color: #777;
```

```
            margin-top: 10px;
        }
    </style>
</head>
<body>

<div class="container">
    <h3>Enter Department Number</h3>
    <form method="GET" action="deptView.php">
        <label for="dno">Department Number:</label>
        <input type="number" id="dno" name="dno" required>
        <input type="submit" value="Submit">
    </form>

    <?php
    try {
        // Connect to Redis
        $redis = new Redis();
        $redis->connect('redis', 6379); // Connect to Redis container

        if (isset($_GET['dno'])) {
            $dno = $_GET['dno'];

            // Build the Redis key for the department
            $dept_key = "DEPARTMENT:$dno";

            // Check if the department exists in Redis
            if ($redis->exists($dept_key)) {
                $department = $redis->hGetAll($dept_key);

                // Display department information
                $dname = htmlspecialchars($department["Dname"]);
                $mssn = htmlspecialchars($department["Mgr_ssn"]);
                $mstart = htmlspecialchars($department["Mgr_start_date"]);
```

```
            echo "<h4>Department: $dname</h4>";
            echo "<p>Manager SSN: <a
href=\"p1.php?ssn=$mssn\">$mssn</a></p>";
            echo "<p>Manager Start Date: $mstart</p>";

            // Query department locations
            echo "<h4 class='section-title'>Department Locations:</h4>";
            $locations = $redis->keys("DEPT_LOCATION:$dno:*");

            if (!empty($locations)) {
                foreach ($locations as $location_key) {
                    $location_data = $redis->hGetAll($location_key);
                    echo htmlspecialchars($location_data['Location']) . "<br>";
                }
            } else {
                echo "<p class='no-data'>No locations found.</p>";
            }

            // Query employees in the department
            echo "<h4 class='section-title'>Employees:</h4>";
            $employee_keys = $redis->keys("EMPLOYEE:*");
            $found_employees = false;

            echo "<table>";
            echo "<tr><th>Employee SSN</th><th>Last Name</th><th>First
Name</th></tr>";
            foreach ($employee_keys as $emp_key) {
                $employee = $redis->hGetAll($emp_key);
                if ($employee['Dno'] == $dno) {
                    $found_employees = true;
                    echo "<tr>";
                    echo "<td><a href=\"p1.php?ssn=" .
htmlspecialchars($employee['Ssn']) . "\">" . htmlspecialchars($employee['Ssn']) .
"</a></td>";
                    echo "<td>" . htmlspecialchars($employee['Lname']) . "</td>";
```

```
                echo "<td>" . htmlspecialchars($employee['Fname']) . "</td>";
                echo "</tr>";
            }
        }
        echo "</table>";
        if (!$found_employees) {
            echo "<p class='no-data'>No employees found.</p>";
        }

        // Query projects in the department
        echo "<h4 class='section-title'>Projects:</h4>";
        $project_keys = $redis->keys("PROJECT:*");
        $found_projects = false;

        echo "<table>";
        echo "<tr><th>Project Number</th><th>Project
Name</th><th>Location</th></tr>";
        foreach ($project_keys as $proj_key) {
            $project = $redis->hGetAll($proj_key);
            if ($project['Dnum'] == $dno) {
                $found_projects = true;
                echo "<tr>";
                echo "<td>" . htmlspecialchars($project['Pnumber']) . "</td>";
                echo "<td>" . htmlspecialchars($project['Pname']) . "</td>";
                echo "<td>" . htmlspecialchars($project['Plocation']) . "</td>";
                echo "</tr>";
            }
        }
        echo "</table>";
        if (!$found_projects) {
            echo "<p class='no-data'>No projects found.</p>";
        }
    } else {
        echo "<p class='no-data'>No department found with number $dno.</p>";
    }
```

```
        } else {
            echo "<p class='no-data'>Please enter a department number above.</p>";
        }
    } catch (Exception $e) {
        echo "Could not connect to Redis: " . htmlspecialchars($e->getMessage());
    }
    ?>
</div>

</body>
</html>
```

- companybrows.php

```
<!DOCTYPE html>
<html>
<head>
    <title>All Departments</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f4f4f9;
            margin: 0;
        }
        .container {
            text-align: center;
            background-color: #ffffff;
            padding: 20px;
            border-radius: 8px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
        }
        h4 {
```

```
            color: #333;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin-top: 10px;
        }
        th, td {
            padding: 10px;
            text-align: left;
            border: 1px solid #ddd;
        }
        th {
            background-color: #4CAF50;
            color: white;
        }
        td a {
            color: #4CAF50;
            text-decoration: none;
        }
        td a:hover {
            text-decoration: underline;
        }
    </style>
</head>
<body>

<div class="container">
    <h4>Departments of the Company</h4>
    <table>
        <tr>
            <th>Department Number</th>
            <th>Department Name</th>
        </tr>
```

```
    <?php
    try {
        // Connect to Redis
        $redis = new Redis();
        $redis->connect('redis', 6379); // Use the Redis container hostname

        // Fetch all department keys
        $dept_keys = $redis->keys("DEPARTMENT:*");

        if (!empty($dept_keys)) {
            foreach ($dept_keys as $dept_key) {
                // Get department details
                $department = $redis->hGetAll($dept_key);

                // Extract department number and name
                $dno = htmlspecialchars($department['Dnumber']);
                $dname = htmlspecialchars($department['Dname']);

                // Display department details in a table row
                echo "<tr>
                        <td><a href=\"deptView.php?dno=$dno\">$dno</a></td>
                        <td>$dname</td>
                      </tr>";
            }
        } else {
            echo "<tr><td colspan='2'>No departments found.</td></tr>";
        }
    } catch (Exception $e) {
        echo "<tr><td colspan='2'>Error: " . htmlspecialchars($e->getMessage()) .
"</td></tr>";
    }
    ?>
  </table>
</div>
```

```
</body>
</html>
```