Database management System and PHP

CS51550, Fall 2024
Purdue University Northwest
10/31/2024.

Name	Email
Shivam Pandya	pandya23@pnw.edu
Vishva Gor	vgor@pnw.edu

YouTube View of the Demo	https://youtu.be/qQMt4FMMgUI
GitHub Repository	https://github.com/shivampandya67/Company-Management

1. Abstract

This project demonstrates the integration of MySQL with PHP in a Docker environment to create a web-based interface for managing organizational data. By leveraging Docker containers, the project isolates services and constructs a dedicated database, named '`mydatabase`', with PHP scripts that provide interactive database access. Additionally, the project utilizes KNIME for efficient data loading and integrates PLSQL scripts to enhance database functionality and data management. This report covers the project's setup, design, and implementation phases, detailing the use of Docker for containerization, KNIME for data preparation, and PLSQL to expand database operations.

2. Introduction

Efficient data management is critical for modern organizations to maintain accurate records and enable seamless information access. This project focuses on creating a robust, web-based database management system using MySQL and PHP, hosted within a Dockerized environment for enhanced modularity, consistency, and deployment ease. The system is designed to allow users to interact with organizational data through a user-friendly web interface, with each service isolated within Docker containers to ensure stability and ease of maintenance.

To optimize data handling, KNIME is employed for loading datasets into the MySQL database, offering an efficient method to preprocess and integrate large volumes of data. Additionally, the project leverages PL/SQL scripting to add advanced functionalities to the database, such as automated triggers and stored procedures, enhancing the system's capabilities and reliability. This report outlines the steps taken to design, implement, and integrate each component of the system, covering environment setup, database and interface design, and the integration of KNIME and PL/SQL. Together, these elements form a comprehensive data management solution that is flexible, scalable, and easy to deploy in diverse organizational settings.

3. Project Objective

The primary objective of this project is to build a Dockerized environment with two interconnected containers: a PHP container and a MySQL container. This setup is designed to streamline the process of storing and managing .dat file records within a structured MySQL database. The project further involves developing various PHP pages that interact with the database to retrieve and display data in an organized and user-friendly format. In addition, PL/SQL and KNIME are employed to facilitate efficient data loading and manipulation, ensuring data accuracy and scalability for complex queries and analyses.

4. ER Diagram

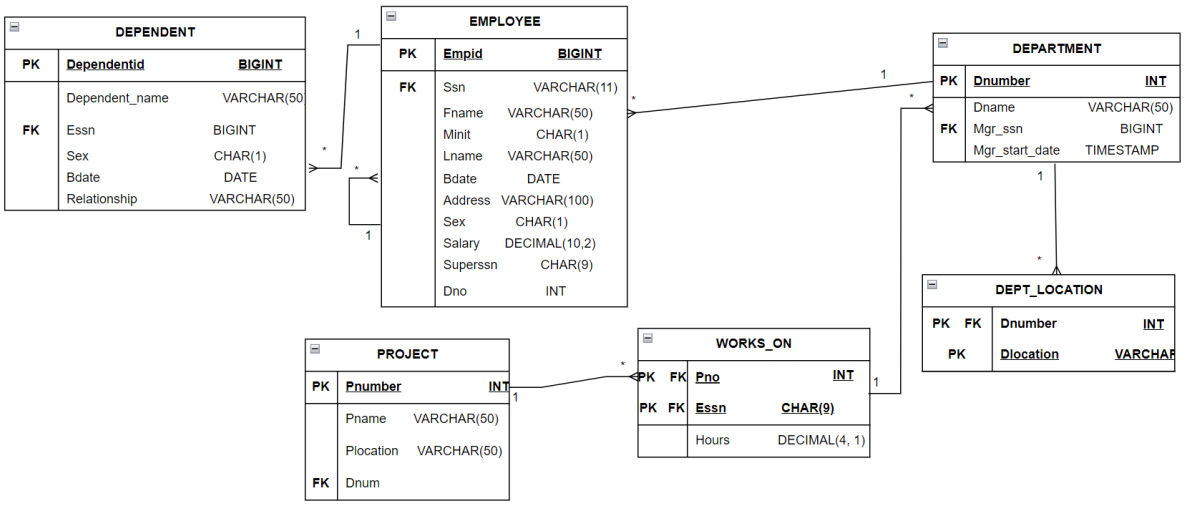


Figure 1: ER diagram

Explanation of the ER Diagram

The mydatabase database includes the following primary entities:

- **Department**: Stores department information, including Dname, Dnumber, and Mgrssn (manager's SSN).
- **Employee**: Represents individual employees, with attributes like Ssn, Fname, Lname, Salary, and Dno (department number).
- **Project**: Contains project details, with Pname, Pnumber, Dnum (department number), and Plocation.
- **Dept_Locations**: Maps departments to multiple locations using Dnumber and Dlocation.
- **Works_On**: Associates employees with projects, specifying hours worked.
- **Dependent Table**: Contains information about employee dependents such as Dependentid, Dependent_name, Essn, Sex, Bdate, and Relationship.

Relationships

- **Department-Employee**: One-to-many relationship. Each department is managed by one employee, while each employee can be part of only one department.
- **Employee-Project**: Many-to-many relationship via the Works_On table, allowing employees to work on multiple projects and vice versa.
- **Department-Dept_Locations**: One-to-many relationship, as each department can have multiple locations.

5. Implementation

5.2 Docker Setup

Docker containers are used to run MySQL and PHP as separate services. The following steps were taken:

1. **MySQL Container**: Configured to initialize with MYSQL_ROOT_PASSWORD and MYSQL_DATABASE=mydatabase.
2. **PHP Container**: Set up to serve PHP files and connect to the MySQL container.

```
shivam@shivam-virtual-machine:~$ mkdir courseProject2
shivam@shivam-virtual-machine:~$ cd courseProject2
shivam@shivam-virtual-machine:~/courseProject2$ mkdir src
shivam@shivam-virtual-machine:~/courseProject2$ touch docker-compose.yml
shivam@shivam-virtual-machine:~/courseProject2$ touch Dockerfile
shivam@shivam-virtual-machine:~/courseProject2$ ls
docker-compose.yml  Dockerfile  src
```

Figure 2: Making courseProject2 directory with docker-compose.yml and Dockerfile

5.2 YAML File Configuration

```
shivam@shivam-virtual-machine:~/courseProject2$ sudo docker-compose up -d
Creating network "courseproject2_default" with the default driver
Creating volume "courseproject2_db-data" with default driver
Building php-apache
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   5.12kB
Step 1/4 : FROM php:8.1-apache
8.1-apache: Pulling from library/php
a480a496ba95: Pull complete
95ab1cc5ca33: Pull complete
78ee5e1490ca: Pull complete
e807ae4973d0: Pull complete
8a1846dfbe9a: Pull complete
27f1d0bbde81: Pull complete
8fac5e585cd6: Pull complete
667a65346814: Pull complete
4a4bd62fe024: Pull complete
548fb97305e4: Pull complete
791621d160a8: Pull complete
87a490ccd81f: Pull complete
e6a802f5636b: Pull complete
```

Figure 3: Launching docker container by using docker-compose command

5.3 Database Initialization

The database is initialized with SQL scripts:

- **Table Creation**: SQL scripts to create DEPARTMENT, EMPLOYEE, PROJECT, etc., with constraints and relationships.
- **Data Loading**: Data is imported from .dat files to populate tables with test data for various departments, employees, and projects.

```
mysql> CREATE TABLE DEPARTMENT (
    ->     Dname VARCHAR(20) NOT NULL,
    ->     Dnumber INT PRIMARY KEY,
    ->     Mgr_ssn CHAR(9),
    ->     Mgr_start_date DATE
    -> );
Query OK, 0 rows affected (0.09 sec)

mysql> CREATE TABLE EMPLOYEE (
    ->     Fname VARCHAR(20),
    ->     Minit CHAR(1),
    ->     Lname VARCHAR(20),
    ->     Ssn CHAR(9) PRIMARY KEY,
    ->     Bdate DATE,
    ->     Address VARCHAR(50),
    ->     Sex CHAR(1),
    ->     Salary DECIMAL(10, 2),
    ->     Super_ssn CHAR(9),
    ->     Dno INT,
    ->     FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
    -> );
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> CREATE TABLE PROJECT (
    ->     Pname VARCHAR(20) NOT NULL,
    ->     Pnumber INT PRIMARY KEY,
    ->     Plocation VARCHAR(20),
    ->     Dnum INT
    -> );
Query OK, 0 rows affected (0.20 sec)

mysql>
mysql>
mysql> CREATE TABLE DEPT_LOCATION (
    ->     Dnumber INT,
    ->     Location VARCHAR(20),
    ->     PRIMARY KEY (Dnumber, Location)
    -> );
Query OK, 0 rows affected (0.11 sec)

mysql>
mysql> CREATE TABLE WORKS_ON (
    ->     Essn CHAR(9),
    ->     Pno INT,
    ->     Hours DECIMAL(3, 1),
    ->     PRIMARY KEY (Essn, Pno)
    -> );
Query OK, 0 rows affected (0.22 sec)
```

```
mysql>
mysql> CREATE TABLE DEPENDENT (
    ->     Essn CHAR(9),
    ->     Dependent_name VARCHAR(20),
    ->     Sex CHAR(1),
    ->     Bdate DATE,
    ->     Relationship VARCHAR(20),
    ->     PRIMARY KEY (Essn, Dependent_name)
    -> );
Query OK, 0 rows affected (0.12 sec)

mysql>
```

Figure 4: Creating Tables

```
mysql> ALTER TABLE DEPARTMENT
    -> ADD CONSTRAINT fk_department_mgr_ssn
    -> FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn);
Query OK, 0 rows affected (0.26 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
mysql> ALTER TABLE EMPLOYEE
    -> ADD CONSTRAINT fk_employee_dno
    -> FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);
Query OK, 0 rows affected (0.24 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql>
```

Figure 5: Adding Constraint

```
mysql> ALTER TABLE PROJECT
    -> ADD CONSTRAINT fk_project_dnum
    -> FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber);
Query OK, 11 rows affected (0.26 sec)
Records: 11  Duplicates: 0  Warnings: 0
```

Figure 6: Constraint for the table PROJECT

```
mysql> ALTER TABLE EMPLOYEE
    -> ADD CONSTRAINT fk_employee_dno
    -> FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);
Query OK, 40 rows affected (0.24 sec)
Records: 40  Duplicates: 0  Warnings: 0
```

Figure 7: Constraint for the table EMPLOYEE

```
mysql> ALTER TABLE DEPARTMENT
    -> ADD CONSTRAINT fk_department_mgr_ssn
    -> FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn);
Query OK, 6 rows affected (0.19 sec)
Records: 6  Duplicates: 0  Warnings: 0
```

Figure 8: Constraint for the table DEPARTMENT

```
mysql> ALTER TABLE DEPT_LOCATION
    -> ADD CONSTRAINT fk_dept_locations_dnumber
    -> FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber);
Query OK, 13 rows affected (0.30 sec)
Records: 13  Duplicates: 0  Warnings: 0
```

Figure 9: Constraint for the table DEPARTMENT

```
mysql> ALTER TABLE WORKS_ON
    -> ADD CONSTRAINT fk_works_on_essn
    -> FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
    -> ADD CONSTRAINT fk_works_on_pno
    -> FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber);
Query OK, 48 rows affected (0.22 sec)
Records: 48  Duplicates: 0  Warnings: 0
```

Figure 10: Constraint for the table WORKS_ON

```
mysql> use mydatabase;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> LOAD DATA LOCAL INFILE '/var/lib/mysql-files/department.dat'
    -> INTO TABLE DEPARTMENT
    -> FIELDS TERMINATED BY ','
    -> OPTIONALLY ENCLOSED BY '"';
Query OK, 6 rows affected (0.04 sec)
Records: 6  Deleted: 0  Skipped: 0  Warnings: 0

mysql> select * from DEPARTMENT;
+----------------+-------------+-----------+----------------+
| dept_name      | dept_number | mgr_ssn   | mgr_start_date |
+----------------+-------------+-----------+----------------+
| Headquarters   |           1 | 888665555 | 1971-06-19     |
| Administration |           4 | 987654321 | 1985-01-01     |
| Research       |           5 | 333445555 | 1978-05-22     |
| Software       |           6 | 111111100 | 1999-05-15     |
| Hardware       |           7 | 444444400 | 1998-05-15     |
| Sales          |           8 | 555555500 | 1997-01-01     |
+----------------+-------------+-----------+----------------+
6 rows in set (0.00 sec)

mysql>
```

Figure 11: Data loading

5.4 PHP Scripts

Five primary PHP files were created:

1. **index.php**: Acts as the entry point, providing navigation to all other scripts.
2. **p1.php**: Accepts an SSN input to find and display specific employee details.
3. **p2.php**: Takes a department number as input and lists all employees in that department.
4. **companyBrowse.php**: Displays a table listing all departments within the organization.
5. **deptView.php**: Offers a comprehensive view of a department, including manager information, employees, and associated projects.

This setup provides an organized and efficient way to interact with the data, offering detailed views and streamlined access to information within each department.

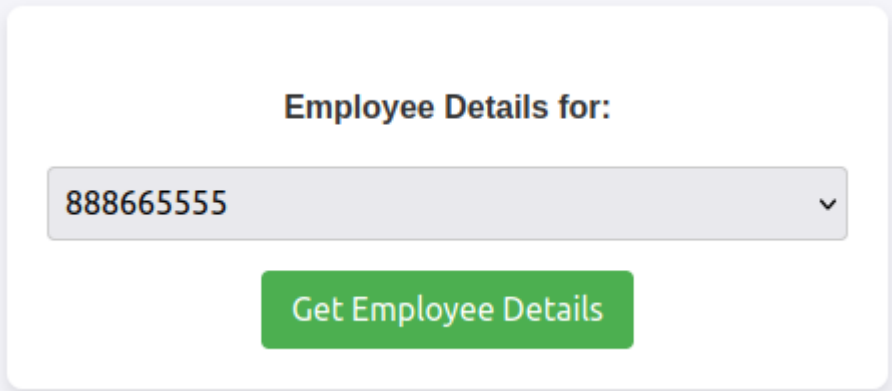


Figure 12: UI where all the employee ssn numbers are shown and you need to select to get the name of that employee

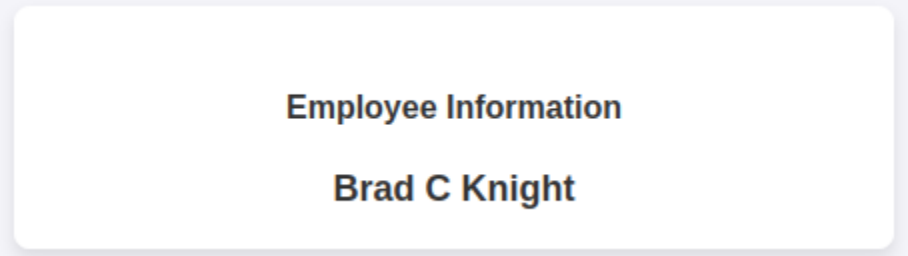


Figure 13: UI when we select a ssn number

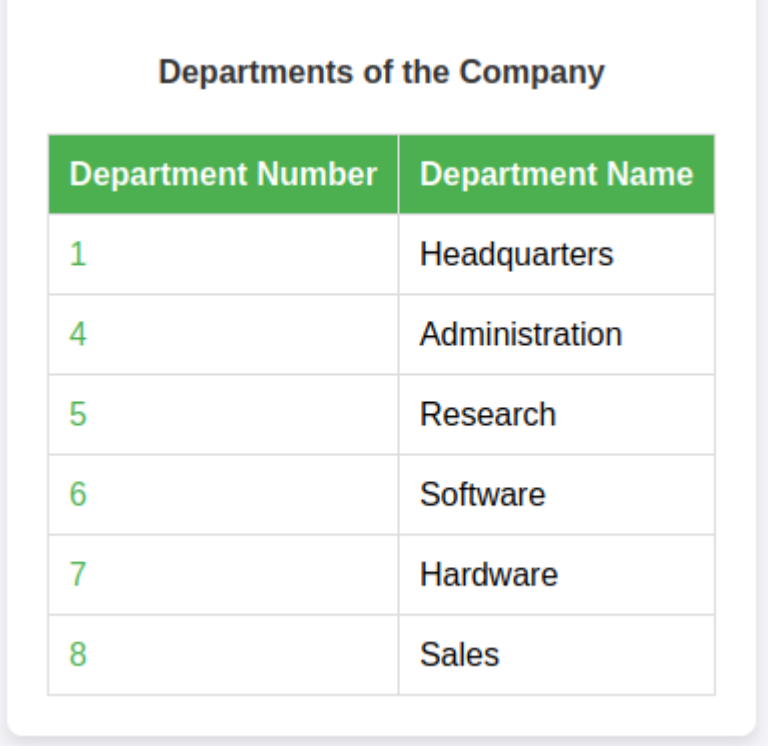


Figure 14: UI of department with its allocated department number

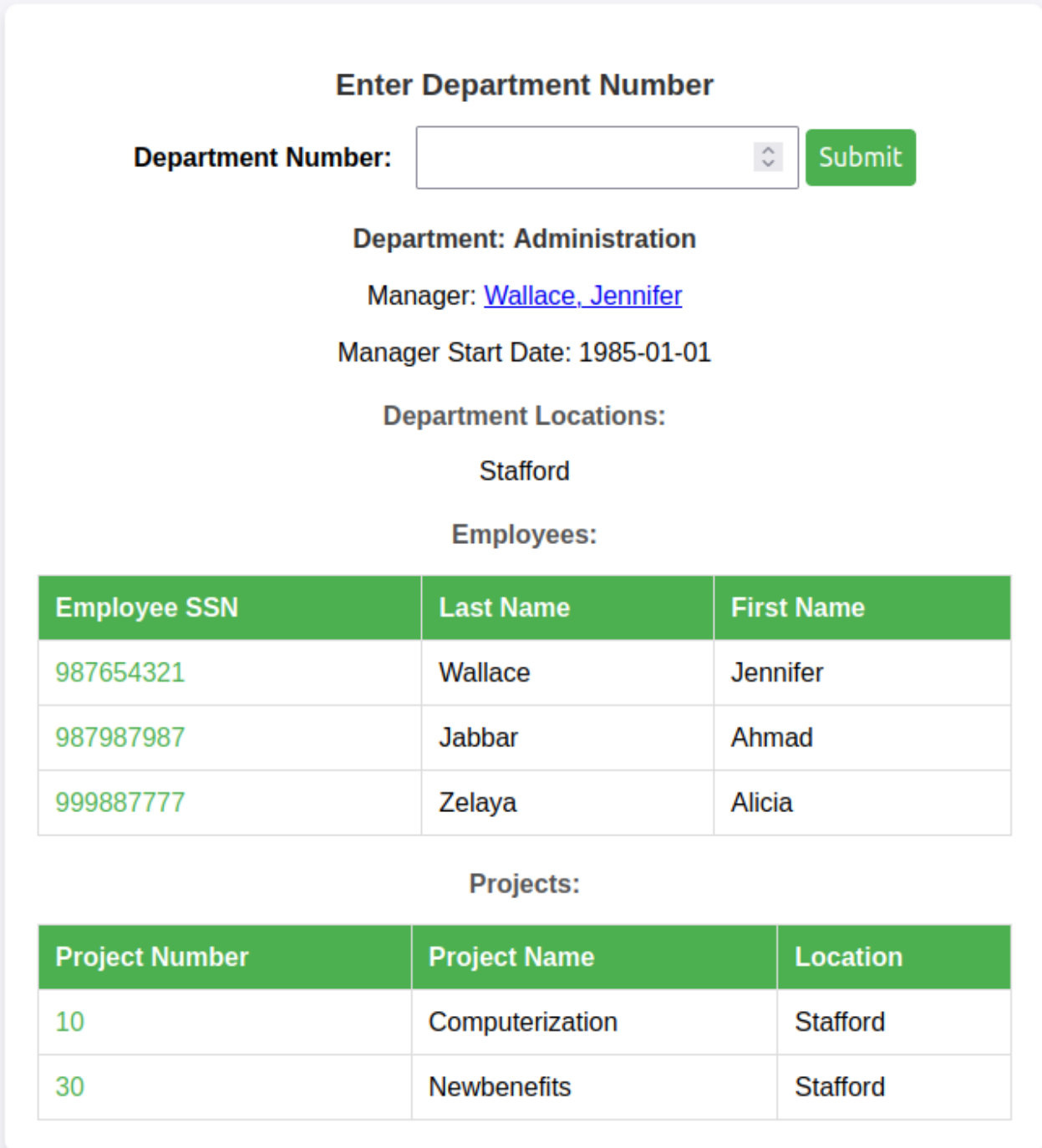


Figure 15: UI when you press one of the department no, it will display the manager, the location, employees and the projects

5.5 PLSQL

```
mysql> CREATE PROCEDURE assign_employee_to_project (
    ->     IN p_essn CHAR(9),  -- Use CHAR(9) for employee SSN
    ->     IN p_pno INT,       -- Use INT for project number
    ->     IN p_hours DECIMAL(3,1)  -- Use DECIMAL for hours
    -> )
    -> BEGIN
    ->     DECLARE record_count INT; -- Declare a variable to hold the count of records
    ->
    ->     -- Check if the record already exists
    ->     SELECT COUNT(*)
    ->     INTO record_count
    ->     FROM WORKS_ON
    ->     WHERE Essn = p_essn AND Pno = p_pno;
    ->
    ->     -- If record exists, update it
    ->     IF record_count > 0 THEN
    ->         UPDATE WORKS_ON
    ->         SET Hours = p_hours
    ->         WHERE Essn = p_essn AND Pno = p_pno;
    ->     ELSE
    ->         -- If record does not exist, insert a new row
    ->         INSERT INTO WORKS_ON (Essn, Pno, Hours)
    ->         VALUES (p_essn, p_pno, p_hours);
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> DELIMITER ;
mysql> CALL assign_employee_to_project('453453453', 1, 23.5);
Query OK, 0 rows affected (0.04 sec)

mysql> CALL assign_employee_to_project('123456789', 1, 20.5);
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> SELECT * FROM WORKS_ON WHERE Essn = '123456789' AND Pno = 1;
+-----------+-----+-------+
| Essn      | Pno | Hours |
+-----------+-----+-------+
| 123456789 |   1 |  20.5 |
+-----------+-----+-------+
1 row in set (0.00 sec)
```

Figure 16: created and called the procedure assign_employee_to_project

Figure 17: displayed the procedure

```
mysql> CREATE PROCEDURE add_employee (
    ->     p_ssn CHAR(9),
    ->     p_fname VARCHAR(20),
    ->     p_minit CHAR(1),
    ->     p_lname VARCHAR(20),
    ->     p_bdate DATE,
    ->     p_address VARCHAR(50),
    ->     p_sex CHAR(1),
    ->     p_salary DECIMAL(10, 2),
    ->     p_super_ssn CHAR(9),
    ->     p_dno INT,
    ->     p_project_number INT,
    ->     p_hours DECIMAL(4, 1)
    -> )
    -> BEGIN
    ->     DECLARE dept_exists INT DEFAULT 0;
    ->     DECLARE project_exists INT DEFAULT 0;
    ->
    ->     -- Step 1: Insert into EMPLOYEE table
    ->     INSERT INTO EMPLOYEE (
    ->         Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
    ->     ) VALUES (
    ->         p_ssn, p_fname, p_minit, p_lname, p_bdate, p_address, p_sex, p_salary, p_super_ssn, p_dno
    ->     );
    ->
    ->     -- Step 2: Check if a project assignment is provided
    ->     IF p_project_number IS NOT NULL THEN
    ->         -- Verify the project exists
    ->         SELECT COUNT(*) INTO project_exists
    ->         FROM PROJECT
    ->         WHERE Pnumber = p_project_number;
    ->
    ->         IF project_exists > 0 THEN
    ->             -- Insert a record into WORKS_ON
    ->             INSERT INTO WORKS_ON (Essn, Pno, Hours)
    ->             VALUES (p_ssn, p_project_number, p_hours);
    ->         ELSE
    ->             SIGNAL SQLSTATE '45000'
    ->             SET MESSAGE_TEXT = 'Project does not exist.';
    ->         END IF;
    ->     END IF;
    -> END //
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> DELIMITER ;
```

Figure 18: Created a procedure which add an employee

```
mysql> CALL add_employee('111223333', 'John', 'A', 'Doe', '1985-07-14', '123 Maple St', 'M', 55000, '987654321', 1, NULL, NULL);
Query OK, 1 row affected (0.07 sec)
```

Figure 19: Adding one employee by calling the procedure

```
mysql> SHOW PROCEDURE STATUS WHERE Db = 'mydatabase';
+------------+---------------------------+-----------+---------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
| Db         | Name                      | Type      | Definer       | Modified            | Created             | Security_type | Comment | character_set_client | collation_connection | Database Collation |
+------------+---------------------------+-----------+---------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
| mydatabase | add_employee              | PROCEDURE | root@localhost | 2024-11-01 19:20:25 | 2024-11-01 19:20:25 | DEFINER       |         | latin1               | latin1_swedish_ci    | utf8mb4_0900_ai_ci |
| mydatabase | assign_employee_to_project | PROCEDURE | root@localhost | 2024-11-01 16:12:45 | 2024-11-01 16:12:45 | DEFINER       |         | latin1               | latin1_swedish_ci    | utf8mb4_0900_ai_ci |
+------------+---------------------------+-----------+---------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
2 rows in set (0.14 sec)
```

Figure 20: Showcasing the procedures

5.6 KNIME

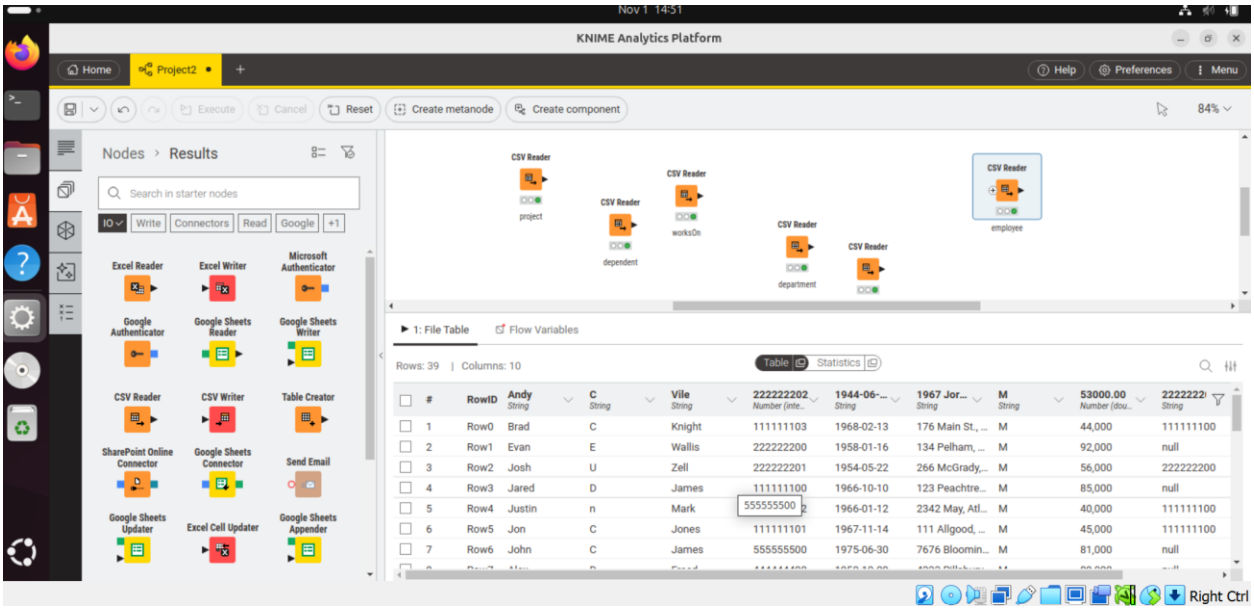
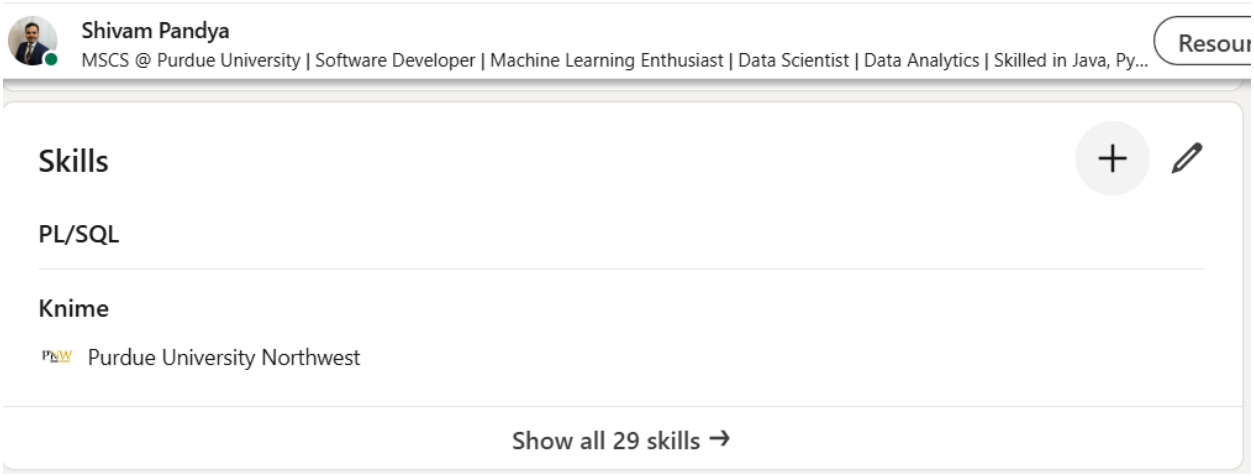


Figure 21: Loading data into KNIME

5.7 LinkedIn Update



6. Challenges

Throughout the project, several challenges were encountered and addressed, particularly related to the integration of Docker, MySQL, and PHP, as well as the handling of data import from the .dat file.

Setting Up Docker Containers: Configuring Docker to run both the MySQL and PHP containers simultaneously required careful attention to the YAML configuration file. Ensuring correct mapping and linking between the containers was crucial to maintaining seamless communication between the database and PHP files.

Importing Data from the .dat File: Loading data from the .dat file into the MySQL database posed unique challenges. The process involved creating a Docker volume to store the .dat file and mapping it to the container's file system. Then, the MySQL container was configured to execute a custom command to import data from the .dat file upon initialization. Managing file paths and ensuring compatibility between the file structure and database schema was essential for a successful import.

Database Synchronization and Updates: Synchronizing the database with PHP scripts within a Docker environment required rigorous testing to confirm that each query and data retrieval operation functioned as intended. Configurations such as user permissions, table structures, and query compatibility had to be continuously monitored and adjusted.

7. Discussion

This project leverages containerization with Docker to create a modular and scalable environment for managing and visualizing data stored in a MySQL database. By setting up separate containers for PHP and MySQL, we achieve a clean separation of concerns, which simplifies deployment and enhances portability. The PHP container acts as the front-end interface, handling user requests and executing queries to interact with the MySQL container, which manages the database. This setup enables a seamless flow from data storage to data presentation, while also allowing for easier maintenance and updates of each container independently.

A key part of the project involves importing records from `.dat` files into the MySQL database, ensuring that data from external sources is accurately and efficiently integrated. We utilize PL/SQL for database management tasks, including complex queries and data transformations, which adds robustness to the data handling process. Additionally, KNIME is used to streamline data loading and preprocessing, offering a powerful tool for data integration and transformation tasks that might otherwise be time-consuming.

The PHP pages developed in this project serve as the user interface, displaying the retrieved data in a well-organized and visually appealing format. These pages facilitate a user-friendly experience for interacting with the database, making it easy for users to view, analyze, and navigate the information. By combining Docker, MySQL, PHP, PLSQL, and KNIME, this project not only showcases an integrated approach to data management but also demonstrates effective use of modern tools and frameworks to create a flexible and scalable solution.

8. conclusion

This project successfully demonstrates the integration of multiple technologies—Docker, MySQL, PHP, PL/SQL, and KNIME—to create a streamlined, efficient, and scalable data management environment. By containerizing the PHP and MySQL components, we achieved modularity and simplified deployment, enhancing portability and ease of maintenance. The process of importing `.dat` files into a structured MySQL database, combined with the use of PL/SQL for database operations, ensures efficient data handling and reliability. KNIME further contributes to data preprocessing and loading, making the overall workflow adaptable to complex datasets. The PHP-driven interface provides a user-friendly means of displaying and interacting with database content, fulfilling the project's objective of presenting data in an accessible, visually organized format. Altogether, this project underscores the power of combining containerization, data management, and visualization tools to create a robust, flexible solution for modern data applications.

9. Acknowledgment

I would like to extend my heartfelt gratitude to Dr. Prof. David Dei for his invaluable guidance in creating the ER model for this project and for his continuous support throughout each stage. His insights and advice have been instrumental in overcoming challenges and achieving the project's objectives. I would also like to thank the community at Stack Overflow for providing solutions and insights that helped address various technical challenges. Additionally, the KNIME YouTube channel and W3Schools have been beneficial resources, offering clear tutorials on KNIME workflows and PLSQL, which greatly contributed to the project's success. I would like to express my sincere gratitude to my team member Shivam who contributed his time, skills, and support throughout the development of this project. Their dedication and teamwork were invaluable in achieving our goals.

10. Appendix

10.1 Creating the tables, inserting data, and adding a constraint

```
CREATE TABLE DEPARTMENT (
   Dname VARCHAR(20) NOT NULL,
   Dnumber INT PRIMARY KEY,
   Mgr_ssn CHAR(9),
   Mgr_start_date DATE
);
CREATE TABLE EMPLOYEE (
   Fname VARCHAR(20),
   Minit CHAR(1),
   Lname VARCHAR(20),
   Ssn CHAR(9) PRIMARY KEY,
   Bdate DATE,
   Address VARCHAR(50),
   Sex CHAR(1),
   Salary DECIMAL(10, 2),
   Super_ssn CHAR(9),
   Dno INT
);

CREATE TABLE PROJECT (
   Pname VARCHAR(20) NOT NULL,
   Pnumber INT PRIMARY KEY,
   Plocation VARCHAR(20),
   Dnum INT
```

```
);
CREATE TABLE DEPT_LOCATION (
   Dnumber INT,
   Location VARCHAR(20),
   PRIMARY KEY (Dnumber, Location)
);
CREATE TABLE WORKS_ON (
   Essn CHAR(9),
   Pno INT,
   Hours DECIMAL(3, 1),
   PRIMARY KEY (Essn, Pno)
);
CREATE TABLE DEPENDENT (
   Essn CHAR(9),
   Dependent_name VARCHAR(20),
   Sex CHAR(1),
   Bdate DATE,
   Relationship VARCHAR(20),
   PRIMARY KEY (Essn, Dependent_name)
);
LOAD DATA LOCAL INFILE '/var/lib/mysql-files/dependent.dat'
INTO TABLE DEPENDENT
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/worksOn.dat'
INTO TABLE WORKS_ON
```

```
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/dloc.dat'
INTO TABLE DEPT_LOCATION
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/project.dat'
INTO TABLE PROJECT
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/employee.dat'
INTO TABLE EMPLOYEE
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/department.dat'
INTO TABLE DEPARTMENT
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"';

ALTER TABLE DEPARTMENT
ADD CONSTRAINT fk_department_mgr_ssn
FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn);
ALTER TABLE EMPLOYEE
```

```
ADD CONSTRAINT fk_employee_dno
FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber);

ALTER TABLE PROJECT
ADD CONSTRAINT fk_project_dnum
FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber);

ALTER TABLE DEPT_LOCATION
ADD CONSTRAINT fk_dept_locations_dnumber
FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber);

ALTER TABLE WORKS_ON
ADD CONSTRAINT fk_works_on_essn
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
ADD CONSTRAINT fk_works_on_pno
FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber);
```

10.2 index.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Data Access</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 {
    margin-bottom: 20px;
    color: #333;
}
select {
    padding: 8px;
    width: 100%;
    margin-bottom: 15px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 4px;
}
input[type="submit"] {
```

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

<div class="container">
    <h4>Employee Details for:</h4>
    <form method="post" action="p1.php">
        <select name="ssn">
            <?php
                $servername = "course-mysql";
                $username = "user";
                $password = "userpassword";
                $dbname = "mydatabase";

                // Create connection
                $conn = new mysqli($servername, $username, $password, $dbname);
```

```
            // Check connection
            if ($conn->connect_error) {
                die("Connection failed: " . $conn->connect_error);
            }

            $query = "SELECT ssn FROM EMPLOYEE";
            $result = $conn->query($query);

            if ($result === false) {
                echo "Error: " . $conn->error;
                exit();
            }

            while ($row = $result->fetch_assoc()) {
                echo "<option value=\"" . htmlspecialchars($row['ssn']) . "\">" .
htmlspecialchars($row['ssn']) . "</option>";
            }

            $conn->close();
        ?>
    </select>
    <input type="submit" value="Get Employee Details">
  </form>
</div>

</body>
```

```
</html>
```

10.3 p1.php

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Data Access</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 {
```

```
            margin-bottom: 20px;
            color: #333;
        }
        .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
            $servername = "course-mysql";
            $username = "user";
            $password = "userpassword";
            $dbname = "mydatabase";

            $ssn = $_POST['ssn'];
            $conn = new mysqli($servername, $username, $password, $dbname);

            if ($conn->connect_error) {
                die("Connection failed: " . $conn->connect_error);
```

```
        }

        $ssn = $conn->real_escape_string($ssn);
        $query = "SELECT fname, minit, lname FROM EMPLOYEE WHERE ssn = '$ssn'";
        $result = $conn->query($query);

        if ($result && $result->num_rows == 1) {
            $row = $result->fetch_assoc();
            echo htmlspecialchars($row['fname']) . " " . htmlspecialchars($row['minit']) . " " .
htmlspecialchars($row['lname']);
        } else {
            echo "No employee found with the given SSN.";
        }

        $conn->close();
    ?>
  </div>
</div>

</body>
</html>
```

10.4 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
$servername = "course-mysql";
$username = "user";
$password = "userpassword";
$database = "mydatabase";

if (isset($_GET['dno'])) {
    $dno = $_GET['dno'];
    $conn = mysqli_connect($servername, $username, $password, $database);

    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }

    $query = "SELECT lname, salary FROM EMPLOYEE WHERE dno = ?";
    $stmt = mysqli_prepare($conn, $query);
    mysqli_stmt_bind_param($stmt, 'i', $dno);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    echo "<h4>Employees in Department $dno</h4>";

    if (mysqli_num_rows($result) > 0) {
?>
    <table>
        <tr>
            <th>Last Name</th>
```

```
            <th>Salary</th>
        </tr>

        <?php
            while ($row = mysqli_fetch_assoc($result)) {
                $lname = htmlspecialchars($row["lname"]);
                $salary = htmlspecialchars($row["salary"]);
        ?>
        <tr>
            <td><?php echo $lname; ?></td>
            <td><?php echo $salary; ?></td>
        </tr>
        <?php
            }
        ?>
    </table>
<?php
    } else {
        echo "<p>No employees found in department $dno.</p>";
    }

    mysqli_stmt_close($stmt);
    mysqli_close($conn);
} else {
    echo "<p>Please enter a department number above.</p>";
}
?>
```

```
</div>

</body>
</html>
```

10.5 companyBrowse.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">
  <?php
    $servername = "course-mysql";
    $username = "user";
    $password = "userpassword";
    $dbname = "mydatabase";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Check connection
    if ($conn->connect_error) {
      die("Connection failed: " . $conn->connect_error);
    }

    // Query the database
    $query = "SELECT dnumber, dname FROM DEPARTMENT ORDER BY dnumber";
    $result = $conn->query($query);
  ?>

  <h4>Departments of the Company</h4>
  <table>
    <tr>
      <th>Department Number</th>
      <th>Department Name</th>
    </tr>
```

```
    <?php
      if ($result && $result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
          $dno = htmlspecialchars($row['dnumber']);
          $dname = htmlspecialchars($row['dname']);
          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>";
      }
      $conn->close();
    ?>
  </table>
</div>

</body>
</html>
```

10.6 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
$servername = "course-mysql";
$username = "user";
$password = "userpassword";
$database = "mydatabase";

if (isset($_GET['dno'])) {
    $dno = $_GET['dno'];
    $conn = mysqli_connect($servername, $username, $password, $database);

    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }

    // Query for department information
    $query = "SELECT Dname, Mgr_ssn, Mgr_start_date, Lname, Fname
            FROM DEPARTMENT, EMPLOYEE
            WHERE Dnumber = ? AND Mgr_ssn = Ssn";
    $stmt = mysqli_prepare($conn, $query);
    mysqli_stmt_bind_param($stmt, 'i', $dno);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);
```

```
    if ($row = mysqli_fetch_assoc($result)) {
        $dname = htmlspecialchars($row["Dname"]);
        $mssn = htmlspecialchars($row["Mgr_ssn"]);
        $mstart = htmlspecialchars($row["Mgr_start_date"]);
        $mlname = htmlspecialchars($row["Lname"]);
        $mfname = htmlspecialchars($row["Fname"]);

        echo "<h4>Department: $dname</h4>";
        echo "<p>Manager: <a href=\"empView.php?ssn=$mssn\">$mlname, $mfname</a></p>";
        echo "<p>Manager Start Date: $mstart</p>";
    } else {
        echo "<p class='no-data'>No department found with number $dno.</p>";
    }

    // Query for department locations
    $query = "SELECT Location FROM DEPT_LOCATION WHERE Dnumber = ?";
    $stmt = mysqli_prepare($conn, $query);
    mysqli_stmt_bind_param($stmt, 'i', $dno);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    echo "<h4 class='section-title'>Department Locations:</h4>";
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            echo htmlspecialchars($row["Location"]) . "<br>";
```

```
        }
    } else {
        echo "<p class='no-data'>No locations found.</p>";
    }

    // Query for employees
    $query = "SELECT Ssn, Lname, Fname FROM EMPLOYEE WHERE Dno = ?";
    $stmt = mysqli_prepare($conn, $query);
    mysqli_stmt_bind_param($stmt, 'i', $dno);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    echo "<h4 class='section-title'>Employees:</h4>";
    if (mysqli_num_rows($result) > 0) {
        echo "<table>";
        echo "<tr><th>Employee SSN</th><th>Last Name</th><th>First Name</th></tr>";
        while ($row = mysqli_fetch_assoc($result)) {
            $ssn = htmlspecialchars($row["Ssn"]);
            $lname = htmlspecialchars($row["Lname"]);
            $fname = htmlspecialchars($row["Fname"]);
            echo "<tr><td><a
href=\"empView.php?ssn=$ssn\">$ssn</a></td><td>$lname</td><td>$fname</td></tr>";
        }
        echo "</table>";
    } else {
        echo "<p class='no-data'>No employees found.</p>";
    }
```

```
        // Query for projects
        $query = "SELECT Pnumber, Pname, Plocation FROM PROJECT WHERE Dnum = ?";
        $stmt = mysqli_prepare($conn, $query);
        mysqli_stmt_bind_param($stmt, 'i', $dno);
        mysqli_stmt_execute($stmt);
        $result = mysqli_stmt_get_result($stmt);

        echo "<h4 class='section-title'>Projects:</h4>";
        if (mysqli_num_rows($result) > 0) {
            echo "<table>";
            echo "<tr><th>Project Number</th><th>Project Name</th><th>Location</th></tr>";
            while ($row = mysqli_fetch_assoc($result)) {
                $pnum = htmlspecialchars($row["Pnumber"]);
                $pname = htmlspecialchars($row["Pname"]);
                $ploc = htmlspecialchars($row["Plocation"]);
                echo "<tr><td><a
href=\"projView.php?pnum=$pnum\">$pnum</a></td><td>$pname</td><td>$ploc</td></tr>
";
            }
            echo "</table>";
        } else {
            echo "<p class='no-data'>No projects found.</p>";
        }

        mysqli_stmt_close($stmt);
        mysqli_close($conn);
    } else {
```

```
        echo "<p class='no-data'>Please enter a department number above.</p>";
    }
    ?>

</div>

</body>
</html>
```

10.7 docker-compose.yml

```
version: '3.8'

services:
  php-apache:
    build: .
    container_name: php-container
    ports:
      - "8080:80"            # Expose the PHP/Apache server on port 8080
    volumes:
      - ./src:/var/www/html   # Map the local src folder to the container's web root
    depends_on:
      - course-mysql         # Ensure MySQL is ready before starting PHP
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost || exit 1"]
      interval: 30s
      timeout: 10s
      retries: 5
```

```
  course-mysql:
    image: mysql:8.0
    container_name: course-mysql
    environment:
      MYSQL_ROOT_PASSWORD: rootpassword
      MYSQL_DATABASE: mydatabase
      MYSQL_USER: user
      MYSQL_PASSWORD: userpassword
    ports:
      - "3307:3306"          # Expose MySQL on port 3306
    volumes:
      - db-data:/var/lib/mysql
      - ./my.cnf:/etc/mysql/conf.d/my.cnf
      - ./data:/var/lib/mysql-files
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 30s
      timeout: 10s
      retries: 5

volumes:
  db-data:
```

10.8 Dockerfile

```
# Dockerfile

# Use the official PHP image with Apache installed
```

```
FROM php:8.1-apache

# Install the MySQL extensions
RUN docker-php-ext-install mysqli pdo pdo_mysql

# Copy the PHP source code from the src folder to the Apache document root
COPY ./src/ /var/www/html/

# Expose port 80 for the web server
EXPOSE 80
```

10.9 Procedures

```
DELIMITER //

CREATE PROCEDURE assign_employee_to_project (
    IN p_essn CHAR(9),  -- Use CHAR(9) for employee SSN
    IN p_pno INT,       -- Use INT for project number
    IN p_hours DECIMAL(3,1)  -- Use DECIMAL for hours
)
BEGIN
    DECLARE record_count INT; -- Declare a variable to hold the count of records

    -- Check if the record already exists
    SELECT COUNT(*)
    INTO record_count
    FROM WORKS_ON
    WHERE Essn = p_essn AND Pno = p_pno;
```

```
    -- If record exists, update it
    IF record_count > 0 THEN
        UPDATE WORKS_ON
        SET Hours = p_hours
        WHERE Essn = p_essn AND Pno = p_pno;
    ELSE
        -- If record does not exist, insert a new row
        INSERT INTO WORKS_ON (Essn, Pno, Hours)
        VALUES (p_essn, p_pno, p_hours);
    END IF;
END //

DELIMITER ;

CALL assign_employee_to_project('453453453', 1, 23.5);
CALL assign_employee_to_project('123456789', 1, 20.5);

SELECT * FROM WORKS_ON WHERE Essn = '123456789' AND Pno = 1;

CREATE PROCEDURE add_employee (
    p_ssn CHAR(9),
    p_fname VARCHAR(20),
    p_minit CHAR(1),
    p_lname VARCHAR(20),
    p_bdate DATE,
    p_address VARCHAR(50),
    p_sex CHAR(1),
```

```
    p_salary DECIMAL(10, 2),
    p_super_ssn CHAR(9),
    p_dno INT,
    p_project_number INT,
    p_hours DECIMAL(4, 1)
)
BEGIN
    DECLARE dept_exists INT DEFAULT 0;
    DECLARE project_exists INT DEFAULT 0;

    -- Step 1: Insert into EMPLOYEE table
    INSERT INTO EMPLOYEE (
        Ssn, Fname, Minit, Lname, Bdate, Address, Sex, Salary, Super_ssn, Dno
    ) VALUES (
        p_ssn, p_fname, p_minit, p_lname, p_bdate, p_address, p_sex, p_salary, p_super_ssn,
p_dno
    );

    -- Step 2: Check if a project assignment is provided
    IF p_project_number IS NOT NULL THEN
        -- Verify the project exists
        SELECT COUNT(*) INTO project_exists
        FROM PROJECT
        WHERE Pnumber = p_project_number;

        IF project_exists > 0 THEN
            -- Insert a record into WORKS_ON
```

```
        INSERT INTO WORKS_ON (Essn, Pno, Hours)
        VALUES (p_ssn, p_project_number, p_hours);
    ELSE
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Project does not exist.';
    END IF;
  END IF;
END //

DELIMITER ;

CALL add_employee('111223333', 'John', 'A', 'Doe', '1985-07-14', '123 Maple St', 'M', 55000,
'987654321', 1, NULL, NULL);

SHOW PROCEDURE STATUS WHERE Db = 'mydatabase';
```