

CSLR51 – Database Management Systems Laboratory #Session: 10 || Date: 16/10/2024
Viva Due: Question 2 (16/10/2024) Moodle Due: 23/10/2024 at 11 PM

1. Embedded SQL using C Develop an application package using C embedding SQL for the Airline Travel Schema you have explored in Session 5. Your package should simulate every command in DDL and DML.

//To install mysql header files -> apt-get install libmysqlclient-dev

//To run embedded sql in c -> gcc test1.c `mysql_config --cflags --libs`

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <mysql/mysql.h>

void finish_with_error(MYSQL *conn) {
    fprintf(stderr, "%s\n", mysql_error(conn));
    mysql_close(conn);
    exit(EXIT_FAILURE);
}

void create_employee_table(MYSQL *conn) {
    const char *create_table =
        "CREATE TABLE IF NOT EXISTS Employees ("
        "eid INT PRIMARY KEY, "
        "ename VARCHAR(100), "
        "salary INT)";

    if (mysql_query(conn, create_table)) {
        finish_with_error(conn);
    }
    printf("Employees table created successfully.\n");
}

void insert_employee(MYSQL *conn) {
    int eid, salary;
    char ename[100];

    printf("Enter employee ID: ");
    scanf("%d", &eid);
    printf("Enter employee name: ");
    scanf("%s", ename);
    printf("Enter salary: ");
    scanf("%d", &salary);
```

```

char query[512];
snprintf(query, sizeof(query),
    "INSERT INTO Employees (eid, ename, salary) VALUES (%d, '%s', %d)",
    eid, ename, salary);

if (mysql_query(conn, query)) {
    finish_with_error(conn);
}
printf("Employee added successfully.\n");
}

void update_employee(MYSQL *conn) {
    int eid, salary;
    char ename[100];

    printf("Enter employee ID to update: ");
    scanf("%d", &eid);
    printf("Enter new employee name: ");
    scanf("%s", ename);
    printf("Enter new salary: ");
    scanf("%d", &salary);

    char query[512];
    snprintf(query, sizeof(query),
        "UPDATE Employees SET ename='%s', salary=%d WHERE eid=%d",
        ename, salary, eid);

    if (mysql_query(conn, query)) {
        finish_with_error(conn);
    }
    printf("Employee updated successfully.\n");
}

void delete_employee(MYSQL *conn) {
    int eid;
    printf("Enter employee ID to delete: ");
    scanf("%d", &eid);

    char query[128];
    snprintf(query, sizeof(query), "DELETE FROM Employees WHERE eid=%d", eid);

    if (mysql_query(conn, query)) {
        finish_with_error(conn);
    }
}

```

```

    }
    printf("Employee deleted successfully.\n");
}

void fetch_employees(MYSQL *conn) {
    if (mysql_query(conn, "SELECT * FROM Employees")) {
        finish_with_error(conn);
    }

    MYSQL_RES *res = mysql_store_result(conn);
    if (res == NULL) {
        finish_with_error(conn);
    }

    int num_fields = mysql_num_fields(res);
    MYSQL_ROW row;

    printf("\nEmployees:\n");
    while ((row = mysql_fetch_row(res))) {
        for (int i = 0; i < num_fields; i++) {
            printf("%s ", row[i] ? row[i] : "NULL");
        }
        printf("\n");
    }

    mysql_free_result(res);
}

void alter_employee_table(MYSQL *conn) {
    int choice;
    printf("1. Add a new column\n");
    printf("2. Drop a column\n");
    printf("Choose an option: ");
    scanf("%d", &choice);

    char query[256];
    if (choice == 1) {
        char new_col_name[100];
        char new_col_type[100];
        printf("Enter new column name: ");
        scanf("%s", new_col_name);
        printf("Enter new column type (e.g., VARCHAR(100)): ");
        scanf("%s", new_col_type);
    }
}

```

```

        snprintf(query, sizeof(query), "ALTER TABLE Employees ADD COLUMN %s %s",
new_col_name, new_col_type);
    } else if (choice == 2) {
        char col_name[100];
        printf("Enter column name to drop: ");
        scanf("%s", col_name);
        snprintf(query, sizeof(query), "ALTER TABLE Employees DROP COLUMN %s", col_name);
    } else {
        printf("Invalid choice.\n");
        return;
    }

    if (mysql_query(conn, query)) {
        finish_with_error(conn);
    }
    printf("Table altered successfully.\n");
}

```

```

void drop_employee_table(MYSQL *conn) {
    char query[128] = "DROP TABLE IF EXISTS Employees";
    if (mysql_query(conn, query)) {
        finish_with_error(conn);
    }
    printf("Employees table dropped successfully.\n");
}

```

```

int main(int argc, char *argv[]) {
    MYSQL *conn;

    if (argc < 5) {
        fprintf(stderr, "Usage: %s <host> <user> <password> <database>\n", argv[0]);
        return EXIT_FAILURE;
    }

    conn = mysql_init(NULL);
    if (conn == NULL) {
        fprintf(stderr, "mysql_init() failed\n");
        return EXIT_FAILURE;
    }

    if (mysql_real_connect(conn, argv[1], argv[2], argv[3], argv[4], 0, NULL, 0) == NULL) {
        finish_with_error(conn);
    }
}

```

```

int choice;
do {
    printf("\nMenu:\n");
    printf("1. Create Employees Table\n");
    printf("2. Insert Employee\n");
    printf("3. Update Employee\n");
    printf("4. Delete Employee\n");
    printf("5. Fetch Employees\n");
    printf("6. Alter Employees Table\n");
    printf("7. Drop Employees Table\n");
    printf("8. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            create_employee_table(conn);
            break;
        case 2:
            insert_employee(conn);
            break;
        case 3:
            update_employee(conn);
            break;
        case 4:
            delete_employee(conn);
            break;
        case 5:
            fetch_employees(conn);
            break;
        case 6:
            alter_employee_table(conn);
            break;
        case 7:
            drop_employee_table(conn);
            break;
        case 8:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice! Please try again.\n");
    }
} while (choice != 8);

```

```
mysql_close(conn);  
return EXIT_SUCCESS;  
}
```

Output:

reethi@DESKTOP-8744EFO:~/dir1/dbms\$./emp 172.21.176.1 test test Lab10

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 5

Employees:

458 Shanti 13000

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 2

Enter employee ID: 123

Enter employee name: Walter

Enter salary: 12345

Employee added successfully.

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 3

Enter employee ID to update: 123

Enter new employee name: Vicky

Enter new salary: 123456

Employee updated successfully.

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee

5. Fetch Employees

6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 5

Employees:

123 Vicky 123456

458 Shanti 13000

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee

5. Fetch Employees

6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 4

Enter employee ID to delete: 123

Employee deleted successfully.

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee

5. Fetch Employees

6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 5

Employees:

458 Shanti 13000

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 6

1. Add a new column
2. Drop a column

Choose an option: 1

Enter new column name: Age

Enter new column type (e.g., VARCHAR(100)): INT

Table altered successfully.

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 5

Employees:

458 Shanti 13000 NULL

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 6

1. Add a new column

2. Drop a column

Choose an option: 2

Enter column name to drop: Age

Table altered successfully.

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee

5. Fetch Employees

6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 5

Employees:

458 Shanti 13000

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee

5. Fetch Employees

6. Alter Employees Table

7. Drop Employees Table

8. Exit

Enter your choice: 6

1. Add a new column

2. Drop a column

Choose an option: 1

Enter new column name: Age

Enter new column type (e.g., VARCHAR(100)): INT

Table altered successfully.

Menu:

1. Create Employees Table

2. Insert Employee

3. Update Employee

4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 2

Enter employee ID: 1245

Enter employee name: Sunny

Enter salary: 8908

Employee added successfully.

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 5

Employees:

458 Shanti 13000 NULL

1245 Sunny 8908 NULL

Menu:

1. Create Employees Table
2. Insert Employee
3. Update Employee
4. Delete Employee
5. Fetch Employees
6. Alter Employees Table
7. Drop Employees Table
8. Exit

Enter your choice: 8

Exiting...

```
mysql> show tables;
+-----+
| Tables_in_lab10 |
+-----+
| employees       |
+-----+
1 row in set (0.00 sec)

mysql> select * from employees;
+-----+-----+-----+-----+
| eid | ename | salary | Age |
+-----+-----+-----+-----+
| 456 | Shanti | 13000 | NULL |
| 1245 | Sunny | 8908 | NULL |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |
```

2. Embedded SQL using PHP and HTML Develop an application package using PHP and HTML embedding SQL for the employee schema you have explored in Session 4. Your package should simulate every command in DDL and DML

I have used XAMPP. XAMPP is a free and open source cross platform web server solution stack. It helps developers by enabling them to locally run and test web applications before deploying. It uses MySQL database, PHP (server side scripting language) and Apache server.

Listed below are the files used:

0.db.php

```
<?php
$host = 'localhost'; // XAMPP MySQL host
$user = 'test1';     // Default username
$password = 'Tooth9240'; // Default password (usually empty)
$dbname = 'employee_db'; // Your database name

$conn = new mysqli($host, $user, $password, $dbname);

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

try {
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
```

```

        die("Could not connect to the database: " . $e->getMessage());
    }

    ?>

```

1.index.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Employee Management System</title>
</head>
<body>
    <h1>Employee Management System</h1>
    <ul>
        <li><a href="create_table.php">Create Employee Table</a></li>
        <li><a href="drop_table.php">Drop Employee Table</a></li>
        <li><a href="alter_table.php">Alter Employee Table</a></li>
        <li><a href="create_employee.php">Add Employee</a></li>
        <li><a href="read_employee.php">View Employees</a></li>
        <li><a href="update_employee.php">Update Employee</a></li>
        <li><a href="delete_employee.php">Delete Employee</a></li>
    </ul>
</body>
</html>

```

2.create_table.php

```

<?php
include 'db.php';

try {
    // Check if the table exists
    $tableCheckQuery = $pdo->query("SHOW TABLES LIKE 'EMPLOYEE'");

    if ($tableCheckQuery->rowCount() > 0) {
        echo "Table 'EMPLOYEE' already exists!";
    } else {
        // Create the table if it doesn't exist
        $sql = "CREATE TABLE EMPLOYEE (
            Fname VARCHAR(15),
            Minit CHAR(1),
            Lname VARCHAR(15),
            Ssn CHAR(9) NOT NULL PRIMARY KEY,
            Bdate DATE,

```

```

        Address VARCHAR(30),
        Sex CHAR(1),
        Salary DECIMAL(10,2),
        Super_ssn CHAR(9),
        Dno INT
    );
    $pdo->exec($sql);
    echo "Table 'EMPLOYEE' created successfully.";
}
} catch (PDOException $e) {
    echo "Error creating table: " . $e->getMessage();
}
?>

```

3.drop_table.php

```

// drop_table.php
<?php
include 'db.php';
$sql = "DROP TABLE IF EXISTS EMPLOYEE";
$pdo->exec($sql);
echo "Table EMPLOYEE dropped successfully.";
?>

```

4.alter_table.php

```

// alter_table.php
<?php
include 'db.php';
$sql = "ALTER TABLE EMPLOYEE ADD COLUMN Phone VARCHAR(15)";
$pdo->exec($sql);
echo "Table EMPLOYEE altered successfully.";
?>

```

5.create_employee.php

```

<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $fname = $_POST['fname'];
    $minit = $_POST['minit'];
    $lname = $_POST['lname'];
    $ssn = $_POST['ssn'];
    $bdate = $_POST['bdate'];
    $address = $_POST['address'];
    $sex = $_POST['sex'];
    $salary = $_POST['salary'];
    $super_ssn = $_POST['super_ssn'];
}

```

```

$dno = $_POST['dno'];

$sql = "INSERT INTO EMPLOYEE (Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary,
Super_ssn, Dno)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
$stmt = $pdo->prepare($sql);

try {
    $stmt->execute([$fname, $minit, $lname, $ssn, $bdate, $address, $sex, $salary, $super_ssn,
$dno]);
    echo "Employee added successfully! <a href='index.php'>Go back</a>";
} catch (PDOException $e) {
    echo "Error adding employee: " . $e->getMessage();
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Add Employee</title>
</head>
<body>
    <h1>Add Employee</h1>
    <form method="POST">
        <label>First Name: <input type="text" name="fname" required></label><br>
        <label>Middle Initial: <input type="text" name="minit" maxlength="1"></label><br>
        <label>Last Name: <input type="text" name="lname" required></label><br>
        <label>SSN: <input type="text" name="ssn" required maxlength="9"></label><br>
        <label>Birth Date: <input type="date" name="bdate" required></label><br>
        <label>Address: <input type="text" name="address" required></label><br>
        <label>Sex: <input type="text" name="sex" maxlength="1" required></label><br>
        <label>Salary: <input type="number" name="salary" step="0.01" required></label><br>
        <label>Supervisor SSN: <input type="text" name="super_ssn" maxlength="9"></label><br>
        <label>Department Number: <input type="number" name="dno" required></label><br>
        <button type="submit">Add Employee</button>
    </form>
    <a href="index.php">Back</a>
</body>
</html>

```

6.read_employee.php

```

<?php
include 'db.php';

```

```

$sql = "SELECT * FROM EMPLOYEE";
$stmt = $pdo->query($sql);
$employees = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>View Employees</title>
</head>
<body>
    <h1>Employee List</h1>
    <table border="1">
        <tr>
            <th>SSN</th>
            <th>Name</th>
            <th>Address</th>
            <th>Salary</th>
        </tr>
        <?php foreach ($employees as $employee): ?>
            <tr>
                <td><?= htmlspecialchars($employee['Ssn']) ?></td>
                <td><?= htmlspecialchars($employee['Fname'] . ' ' . $employee['Minit'] . ' ' .
$employee['Lname']) ?></td>
                <td><?= htmlspecialchars($employee['Address']) ?></td>
                <td><?= htmlspecialchars($employee['Salary']) ?></td>
            </tr>
        <?php endforeach; ?>
    </table>
    <a href="index.php">Back</a>
</body>
</html>

```

7.update_employee.php

```

<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $ssn = $_POST['ssn'];
    $fname = $_POST['fname'];
    $minit = $_POST['minit'];
    $lname = $_POST['lname'];
    $address = $_POST['address'];
    $salary = $_POST['salary'];
}

```

```
$sql = "UPDATE EMPLOYEE SET Fname = ?, Minit = ?, Lname = ?, Address = ?, Salary = ?  
WHERE Ssn = ?";
```

```
$stmt = $pdo->prepare($sql);  
$stmt->execute([$fname, $minit, $lname, $address, $salary, $ssn]);
```

```
echo "Employee updated successfully! <a href='index.php'>Go back</a>";  
}
```

```
$employee = null;  
if (isset($_GET['ssn'])) {  
    $sql = "SELECT * FROM EMPLOYEE WHERE Ssn = ?";  
    $stmt = $pdo->prepare($sql);  
    $stmt->execute($_GET['ssn']);  
    $employee = $stmt->fetch(PDO::FETCH_ASSOC);  
}  
?>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Update Employee</title>  
</head>  
<body>  
    <h1>Update Employee</h1>  
    <?php if ($employee): ?>  
        <form method="POST">  
            <input type="hidden" name="ssn" value="<?= htmlspecialchars($employee['Ssn']) ?>">  
            <label>First Name: <input type="text" name="fname" value="<?= htmlspecialchars($employee['Fname']) ?>" required></label><br>  
            <label>Middle Initial: <input type="text" name="minit" value="<?= htmlspecialchars($employee['Minit']) ?>" maxlength="1"></label><br>  
            <label>Last Name: <input type="text" name="lname" value="<?= htmlspecialchars($employee['Lname']) ?>" required></label><br>  
            <label>Address: <input type="text" name="address" value="<?= htmlspecialchars($employee['Address']) ?>" required></label><br>  
            <label>Salary: <input type="number" name="salary" step="0.01" value="<?= htmlspecialchars($employee['Salary']) ?>" required></label><br>  
            <button type="submit">Update Employee</button>  
        </form>  
    <?php else: ?>  
        <p>Employee not found!</p>  
    <?php endif: ?>  
    <a href="index.php">Back</a>  
</body>  
</html>
```


8.delete_employee.php

```
<?php
include 'db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $ssn = $_POST['ssn'];

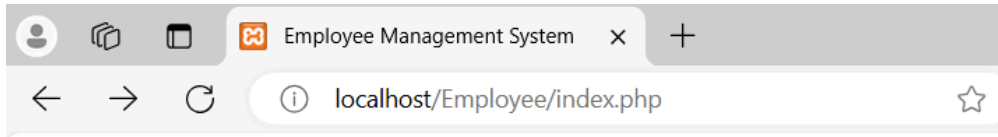
    $sql = "DELETE FROM EMPLOYEE WHERE Ssn = ?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$ssn]);

    echo "Employee deleted successfully! <a href='index.php'>Go back</a>";
}

$employees = $pdo->query("SELECT * FROM EMPLOYEE")->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Delete Employee</title>
</head>
<body>
    <h1>Delete Employee</h1>
    <form method="POST">
        <label>Select Employee:
            <select name="ssn" required>
                <option value="">--Select Employee--</option>
                <?php foreach ($employees as $employee): ?>
                    <option value="<?= htmlspecialchars($employee['Ssn']) ?>"><?=
htmlspecialchars($employee['Fname'] . ' ' . $employee['Lname']) ?></option>
                <?php endforeach; ?>
            </select>
        </label>
        <button type="submit">Delete Employee</button>
    </form>
    <a href="index.php">Back</a>
</body>
</html>
```

Outputs:



Employee Management System

- [Create Employee Table](#)
- [Drop Employee Table](#)
- [Alter Employee Table](#)
- [Add Employee](#)
- [View Employees](#)
- [Update Employee](#)
- [Delete Employee](#)

A screenshot of a web browser window showing the 'Add Employee' form. The address bar shows 'localhost/Employee/create_employee.php'. The page title is 'Add Employee'. The form contains the following fields:

- First Name:
- Middle Initial:
- Last Name:
- SSN:
- Birth Date: (with a calendar icon)
- Address:
- Sex:
- Salary:
- Supervisor SSN:
- Department Number: (with a dropdown arrow)

At the bottom of the form, there is an 'Add Employee' button and a 'Back' link.

View Employees

×

+

←

↻

localhost/Employee/read_employee.php

Employee List

SSN	Name	Address	Salary
1234567	Manoj K Tiwari	No:3,Church Street,Guindy,Chen	25000.00

[Back](#)

Update Employee

×

+

←

↻

localhost/Employee/update_employee.php?ssn=1234

Update Employee

First Name:

Middle Initial:

Last Name:

Address:

Salary:

[Back](#)

