

Shift Scheduling System – README

Objective

To build a fully functional **Shift Scheduling System** using **MySQL only** (without any frontend or backend). This system efficiently manages employees, shifts, availability, and leaves, and ensures conflict-free shift assignments using only SQL.

Core Features

1. Employee Management
 - Add, update, delete employee records.
 - Stores contact details, department, role, skills, and preferred shifts.
2. Shift Management
 - Create and manage different shift types like Morning, Evening, and Night with time ranges.
3. Shift Assignment
 - Assign employees to shifts on specific dates.
 - Prevents double-booking using constraints and triggers.
4. Conflict Detection
 - Prevents assignment of shifts if employees are already assigned or on leave.
5. Availability Tracking
 - Track which employees are available on which dates.
6. Shift Swap Requests
 - Allows employees to request shift swaps.
 - Track status: 'Pending', 'Approved', 'Rejected'.
7. Shift Coverage Reporting
 - Generate reports of who is assigned to what shift on which date.
 - Identify unassigned employees.
8. Audit Logs
 - Automatically logs every assignment and shift update via triggers.

Bonus Features

- **Automated Shift Assignment**:
 - Assign available employees automatically to the Morning shift using stored procedure 'auto_assign_shift'.
- **Role-Based Scheduling**:
 - Leverage employee role/department data to structure future logic for skill-based shift planning.
- **Leave Management**:
 - Prevents shift assignment during the leave period using triggers.
- **Historical Data Analysis**:
 - Query workload reports and analyze how many shifts each employee handled.

Technologies Used

- Database: MySQL (version 5.7 or above)
 - SQL Concepts: Schema Design, Constraints, Foreign Keys, Triggers, Views, Stored Procedures
-

Database Tables Overview

Table Name

- employees - Store employee records
 - shifts - Define different shift timings
 - availability - Track which employee is available when
 - leave_requests - Record leave periods for employees
 - shift_assignments - Assign specific employees to shifts
 - shift_swap_requests - Track employee shift swap requests
 - audit_log - Track changes in shift assignments
-

Triggers

1. trg_log_shift_assignment
 - Logs all new shift assignments into 'audit_log'.
 2. trg_prevent_assignment_on_leave
 - Blocks shift assignment if the employee is on leave during that date.
-

Stored Procedure

'auto_assign_shift(date)'

- Automatically assigns available and eligible employees to the Morning shift for a given date.
 - Checks for:
 - Availability
 - Existing assignments
 - Approved leave periods
-

Views

1. 'view_shift_coverage'
 - Shows which employee is assigned to what shift on which date.
 2. 'view_unassigned_employees'
 - Lists available employees who are not yet assigned for a specific date.
-

How to Run

1. Install MySQL & Set Up Your Environment

Make sure you have one of the following installed:

MySQL Workbench (recommended for GUI-based interaction)

XAMPP/phpMyAdmin (for browser-based SQL interface)

MySQL CLI (for command-line execution)

2. Create and Open Your SQL File

Open your SQL IDE (e.g., MySQL Workbench), then:

Create a new SQL file

Copy and paste the entire content of your .sql project (the one we just finished)

3. Execute the Script

Your script already contains:

```
sql
Copy
Edit
CREATE DATABASE IF NOT EXISTS shift_scheduler_system;
USE shift_scheduler_system;
So when you run it:
```

The database will be created if it doesn't exist

All tables, triggers, stored procedures, views, and sample data will be initialized

In MySQL Workbench, press Ctrl + Shift + Enter or click the `Execute` button.

4. Verify It's Working

Try some basic tests:

```
sql
Copy
Edit
-- See available databases
SHOW DATABASES;

-- Use your new database
USE shift_scheduler_system;

-- See all employees
SELECT * FROM employees;

-- View shift coverage report
SELECT * FROM view_shift_coverage;

-- Try assigning shifts automatically
CALL auto_assign_shift('2025-06-26');
```

5. Explore, Modify, and Extend

You can now:

Add more employees or shift types

Insert availability or leave records

Test triggers by trying to assign a shift during a leave period

Approve or reject swap requests manually

Use the views for daily shift coverage reports or admin dashboards

Useful Queries

- ****Employee's shifts****:

```
““sql
SELECT * FROM shift_assignments WHERE employee_id = 1;
““
```

- ****Pending swap requests****:

```
““sql
SELECT * FROM shift_swap_requests WHERE status = 'Pending';
““
```

- ****Employees on leave today****:

```
““sql
SELECT name FROM employees e
JOIN leave_requests l ON e.employee_id = l.employee_id
WHERE CURDATE() BETWEEN l.leave_start AND l.leave_end;
““
```

- ****Employee workload****:

```
““sql
SELECT e.name, COUNT(sa.assignment_id) AS shift_count
FROM employees e
LEFT JOIN shift_assignments sa ON e.employee_id = sa.employee_id
GROUP BY e.employee_id;
““
```

Notes

- All business logic is encapsulated within MySQL.
 - This system is fully portable and backend-independent.
-

