An Example of Hotel Maintenance System built with SQL Database

Abstract: This paper presents the design and implementation of an automated hotel maintenance system aimed at integrating the maintenance of hotel operations, from a request is sent, to an employee is assigned the task, till the job is accomplished and the log record is updated, all tasks are cascaded because the system utilizes a relational database ensuring timely repairs, and minimizing service disruptions. All work orders are streamlined, tracking asset conditions, and to schedule preventive maintenance — to keep maintaining data accuracy and accessibility.

System Brief Structure

The hotel maintenance system built with an **SQL database** focuses on organizing data related to:

- Maintenance requests
- Staff assignments
- Rooms and facilities
- Equipment/assets
- Maintenance schedules

Database Schema Components

Here is a simplified SQL schema that supports core maintenance functionalities:

1. Rooms Table

sql
CopyEdit

```
CREATE TABLE Rooms (
   RoomID INT PRIMARY KEY,
   RoomNumber VARCHAR(10),
   Floor INT,
   RoomType VARCHAR(50)
);
```

2. MaintenanceRequests Table

sql
CopyEdit

```
CREATE TABLE MaintenanceRequests (
   RequestID INT PRIMARY KEY,
   RoomID INT,
   IssueDescription TEXT,
   Status VARCHAR(20), -- e.g., 'Pending', 'In Progress', 'Completed'
   RequestDate DATE,
   CompletionDate DATE,
   FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)
);
```

3. Staff Table

sql

CopyEdit

```
CREATE TABLE Staff (
    StaffID INT PRIMARY KEY,
    Name VARCHAR(100),
    Role VARCHAR(50)
);
```

4. Assignments Table

sql

CopyEdit

```
CREATE TABLE Assignments (
    AssignmentID INT PRIMARY KEY,
    RequestID INT,
    StaffID INT,
    AssignedDate DATE,
    FOREIGN KEY (RequestID) REFERENCES MaintenanceRequests(RequestID),
    FOREIGN KEY (StaffID) REFERENCES Staff(StaffID)
);
```

Benefits of SQL-Based Maintenance System

- **Efficient Tracking:** Keeps logs of issues and repairs.
- **Scheduling:** Enables preventive maintenance planning.
- **Accountability:** Links tasks to specific staff members.
- **Reporting:** Facilitates data-driven decisions with SQL queries.

Sample Query: Open Requests Per Staff Member

sql

CopyEdit

```
SELECT s.Name, COUNT(*) AS OpenRequests
FROM Staff s
JOIN Assignments a ON s.StaffID = a.StaffID
JOIN MaintenanceRequests r ON a.RequestID = r.RequestID
WHERE r.Status != 'Completed'
GROUP BY s.Name;
```

Reference

Priyadharshini, S., & Joy, C. R. (2021). Design and implementation of an automated hotel management system. *International Journal of Engineering and Advanced Technology (IJEAT)*, Volumn 10, Issue 5, 256–259. https://doi.org/10.35940/ijeat.E2569.0610521

Objectives

To understand the fundamental principles of how data is structured, stored, processed, and used effectively in various systems, to build up a foundation for a data scientist career after graduation.

Learning outcomes for Data Concept

1. SQL(Structured Query Language) is the fundamental tool to work on relational database, which can create tables, insert data, create views and fetch data to specific requirements.
2. Entity, Primary key, foreign key concepts and their relationships.
3. Transaction is not an entity; therefore, it should not be a table.
4. Entity relationship graphic structure is a central part of a relational database
5. How to explore data integrity and security
6. The difference between the OLTP and OLAP as below comparison

Feature	**OLTP (Online Transaction Processing)**	**OLAP (Online Analytical Processing)**
Purpose	Handles **day-to-day operations** (e.g., booking, sales)	Supports **data analysis and decision making**
Data Type	Current, real-time transactional data	Historical, aggregated, and analytical data
Users	**Clerks, front-line staff, customers**	**Analysts, managers, executives**
Operations	Simple queries: INSERT, UPDATE, DELETE, SELECT	Complex queries: GROUP BY, JOIN, CUBE
Speed	Fast for small transactions	Optimized for large, complex queries
Database Design	**Normalized** (to reduce redundancy)	**Denormalized** (to improve read speed)
Examples	Banking systems, hotel reservations, inventory	Business dashboards, trend analysis, reports
Transaction Volume	High (many small transactions per second)	Low (fewer but heavier analytical queries)

⍰ **OLTP = Real-time operations** (fast, many transactions)

⍰ **OLAP = Strategic analysis** (deep insights, fewer but complex queries)

Edward Zhou