Transport Management System

27.11.2023

—

Anshika 2021CSB1069

Darakshinda 2021CSB1130

Chaitanya 2021CSB1121

B.Tech 3rd Year

Department of Computer Science and Engineering, IIT Ropar

Github Link

Overview

The Transport Management System project simplifies and streamlines bus and taxi transportation services through the integration of a robust Database Management System (DBMS). The project leverages SQL for database management and Flask and Python for scripting and backend logic. The system focuses on enhancing efficiency, transparency, and user experience in managing transportation services.

Features

1. **Add and manage passenger details.**
2. **Add and manage employee details.**
3. **Make bus reservations and track transactions.**
4. **Import and export data to/from CSV files.**
5. **Clear all data in the database.**

Goals

6. **Automation of Operations:** Develop a comprehensive TMS Transport Management Systemto automate various aspects of transportation management, including route planning, scheduling, ticketing, and reporting.
7. **Database Management**: Implement a reliable and scalable database using SQL to store and manage data related to vehicles, routes, schedules, passengers, and transactions.
8. **User-Friendly Interface**: Create an intuitive and user-friendly interface for both administrators and end-users to facilitate easy interaction with the system.

Specifications

The proposed system utilizes a three-tier architecture:

1. **Presentation Layer:** Developed using Python-based frameworks (e.g., Flask) for a dynamic and responsive user interface.
2. **Application Layer**: Python scripts for backend logic, business rules, and seamless integration with the database.

3. **Data Layer**: A MySQL database managed using SQL to store and retrieve information related to buses, passengers, bus stops and transactions.

Contributions

- ### Database Structure and Queries in MySQL

  Schema Tables Design: Anshika, Darakshinda,Chaitanya

  Schema Tables Creation Queries: Anshika, Darakshinda

  Schema Data Manipulation Queries: Anshika, Darakshinda, Chaitanya

- ### Backend using Flask

  Backend design: Anshika, Darakshinda,Chaitanya

  Authorization: Anshika

  Backend Functions: Anshika, Darakshinda

- ### Frontend

  Frontend Design: Darakshinda,Chaitanya

  Frontend HTML: Anshika

- ### Report and ER Diagram

  Anshika, Darakshinda

Normalization Analysis

- ### First Normal Form

  A. **Atomic values:** Each table column has atomic (indivisible) values. The Passenger table's passen_fname and passen_lname columns include first and last names.

  B. **No Group Repetition:** No column has recurring groups or arrays. Each table cell has an indivisible value.

  C. **Unique Column Names**: Table columns have distinct names. No Passenger or other table has two columns with the same name.

- ## Second Normal Form

A. **Meets 1NF**: As noted, the schema meets first normal form criteria. Atomic values, no repeated groupings, and unique column names are in each column.

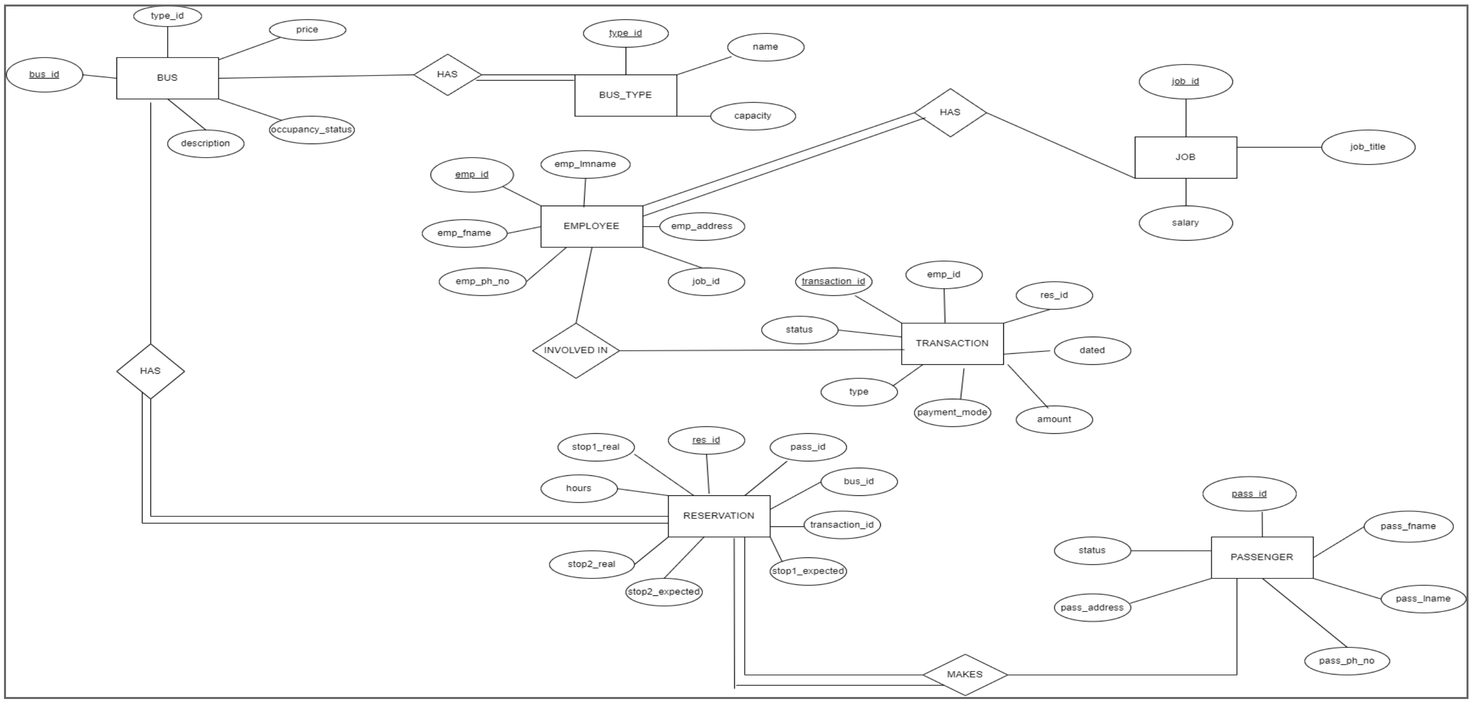
B. **No Partial Dependencies**: 2NF ensures that all non-prime characteristics (attributes not in any candidate key) are functionally reliant on the primary key, eliminating partial dependencies. Passenger table main key is passen_id.

The main key controls all other properties (passen_fname, lname, address, ph_no, status).Res_id is the Reservation table's main key.

No partial dependencies exist between the main key and the attributes (passen_id, bus_id, transaction_id, stop1_expected, stop1_real, stop2_expected, stop2_real, hours).Other tables like Bus, Bus_Type, Transactions, Employees, Job, and User have characteristics that are functionally reliant on their main keys.

C. **Proper Use of Foreign Keys**: Tables are linked via foreign keys. The Employees table's job_id foreign key references the Job table. This links the Employees and Job tables using the Job table's main key..

ER Diagram


BUS
bus_id
type_id
price
description
occupancy_status
HAS
BUS_TYPE
type_id
name
capacity
HAS
JOB
job_id
job_title
salary
EMPLOYEE
emp_id
emp_lmname
emp_fname
emp_address
emp_ph_no
job_id
INVOLVED IN
TRANSACTION
transaction_id
emp_id
res_id
status
dated
type
payment_mode
amount
HAS
RESERVATION
res_id
stop1_real
pass_id
hours
bus_id
transaction_id
stop2_real
stop2_expected
stop1_expected
PASSENGER
pass_id
pass_fname
status
pass_lname
pass_address
pass_ph_no
MAKES