**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Database Management System**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program :- BTech AIML |  | |
| Semester:- IVth |  | |
| Name of the Project:- NextGen Airport Control System |  | |
|  | | |
| Details of Project Members |  |  |
| Batch : | Roll No. | Name |
| B | C264 | Rohit Badgujar |
| B | C229 | Vaibhav Gujrathi |
| B | C243 | Muzammil Khatik |
| Date of Submission: | | |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Project Report**

**NextGen Airport Control System**

**By**

**Rohit Badgujar C264**

**Vaibhav Gujrathi C229**

**Muzammil Khatik C243**

**Course: DBMS**

**AY: 2024-25**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline | 1 |
| **2** | Components of Database Design | 1 |
| **3** | Entity Relationship Diagram | 2 |
| **4** | Relational Model | 2 |
| **5** | Normalization | 3 |
| **6** | SQL Queries | 3 |
| **7** | Learning from the Project | 8 |
| **8** | Project Demonstration | 13 |
| **9** | Self-learning beyond classroom | 13 |
| **10** | Learning from the project | 13 |
| **8** | Challenges faced | 13 |
| **9** | Conclusion | 13 |

**I. Storyline**

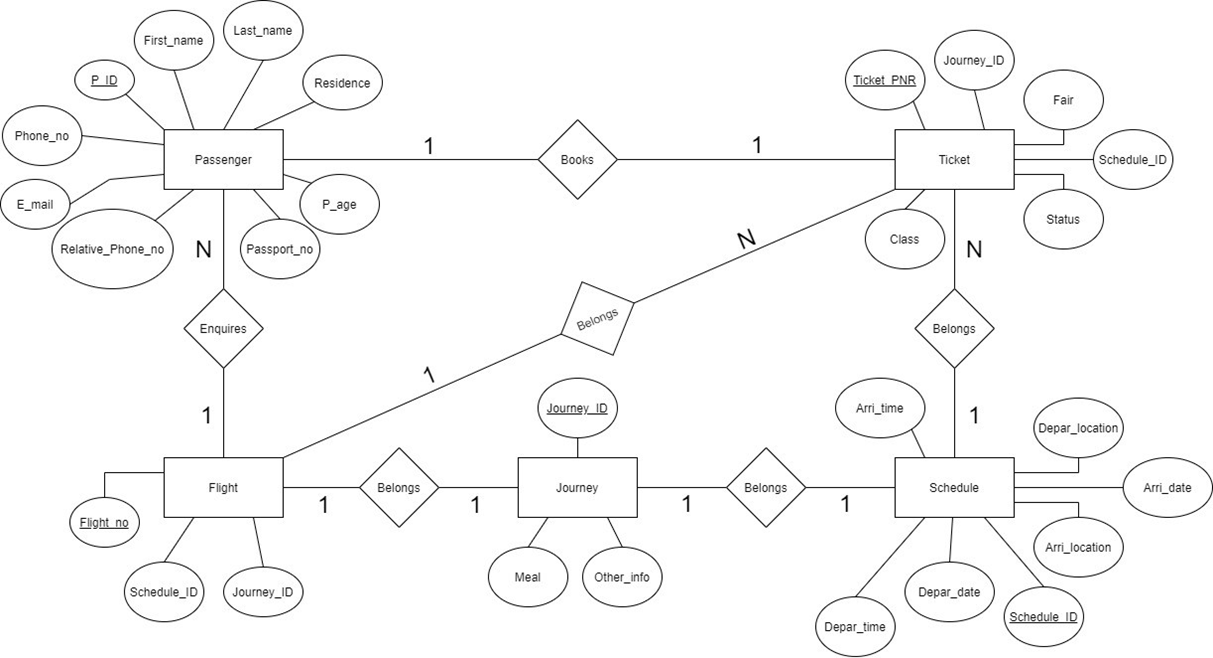
The Airport Management System (AMS) facilitates airline operations by managing flight schedules, passenger information, ticketing, and journey details. It allows passengers to book flights, stores their personal information, and tracks their travel history. The system links tickets, flights, and schedules to efficiently manage airport traffic and resources

**II. Components of Database Design**

**Entities and Attributes:**

* **Passenger**
  + P\_ID (PK)
  + First\_Name
  + Last\_Name
  + P\_age
  + Residence
  + Phone\_no
  + Relative\_phone\_no
  + Email
  + Passport\_no (Unique)
  + Ticket\_PNR (FK)
* **Journey**
  + Journey\_ID (PK)
  + Meal
  + Other\_Info
* **Schedule**
  + Schedule\_ID (PK)
  + Depar\_location
  + Arri\_location
  + Depar\_time
  + Arri\_time
  + Depar\_date
  + Arri\_date
* **Flight**
  + Flight\_no (PK)
  + Schedule\_ID (FK)
  + Journey\_ID (FK)
* **Ticket**
  + Ticket\_PNR (PK)
  + Journey\_ID (FK)
  + Schedule\_ID (FK)
  + Fair
  + Class
  + Status

**III. Entity Relationship Diagram**



**IV. Relational Model**

Each entity has been translated into a relational schema:

Passenger(P\_ID, First\_Name, Last\_Name, P\_age, Residence, Phone\_no, Relative\_phone\_no, Email, Passport\_no, Ticket\_PNR)

Journey(Journey\_ID, Meal, Other\_Info)

Schedule(Schedule\_ID, Depar\_location, Arri\_location, Depar\_date, Arri\_date, Depar\_time, Arri\_time)

Flight(Flight\_no, Schedule\_ID, Journey\_ID)

Ticket(Ticket\_PNR, Journey\_ID, Schedule\_ID, Fair, Class, Status)

**V. Normalization**

Third Normal Form (3NF) is a database design principle that ensures:

No transitive dependencies exist.

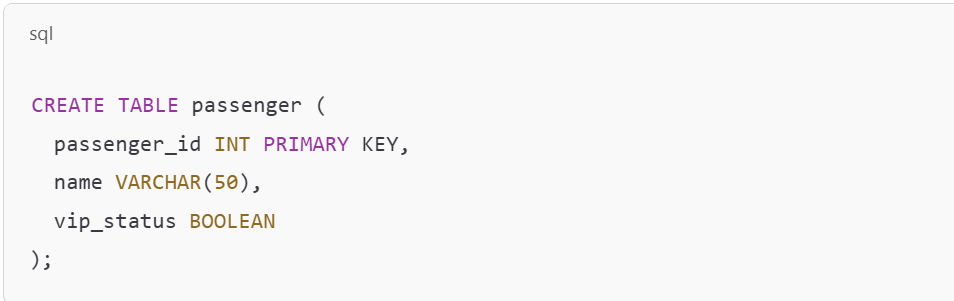
Every non-key attribute is fully dependent only on the primary key.

The database is structured in a way that eliminates redundancy and promotes data integrity.

📦 How 3NF is Applied in Our Project

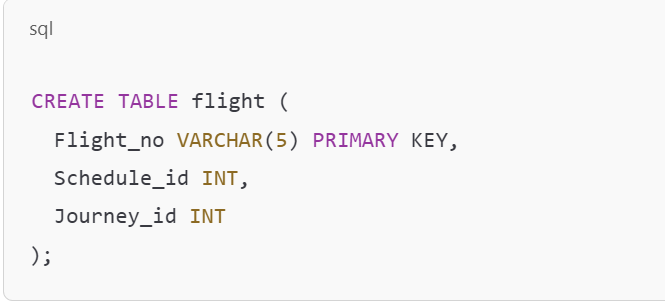
Our Airport Management System database strictly follows 3NF by organizing data across multiple related tables, each focused on a single entity. Here’s how:

1. Passenger Table



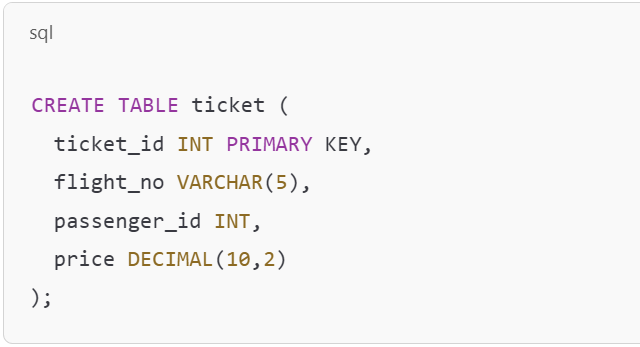
* Each attribute depends only on the primary key passenger\_id.
* Passenger-related data is not mixed with tickets or flights.
* No transitive dependencies.

2. Flight Table



* Flight\_no is the primary key.
* Foreign keys Schedule\_id and Journey\_id link to other normalized tables.
* All attributes depend only on the primary key.

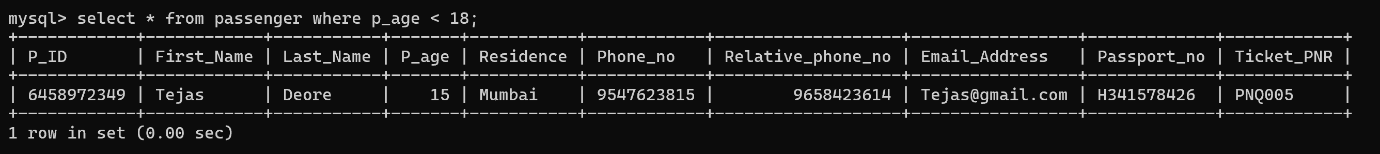
3. Ticket Table



* Connects passengers and flights through foreign keys.
* Ticket table does not store passenger names or flight times.
* Ensures data is only stored once, avoiding duplication.

**VI. SQL Queries**

1. Find the information of the passengers who are below 18 years.

select \* from passenger where p\_age < 18;

A screenshot of a computer

AI-generated content may be incorrect.

1. Find the departure time and date ofticket PNR = PNQ002.

select depar\_date, depar\_time from schedule where schedule\_id=(select schedule\_id from ticket where ticket\_PNR='PNQ002');

A black screen with white text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Find the ticket\_pnr whose ticket is not confirmed.

A black screen with white text

AI-generated content may be incorrect.select ticket\_pnr from ticket where status='NOT CONFIRMED';

A screenshot of a computer

AI-generated content may be incorrect.

1. Show the passenger’s name along with the class in which they will travel.

select passenger.First\_Name as "Passenger Name", ticket.class as "Class" from ticket inner join passenger on passenger.Ticket\_PNR=ticket.ticket\_PNR;

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

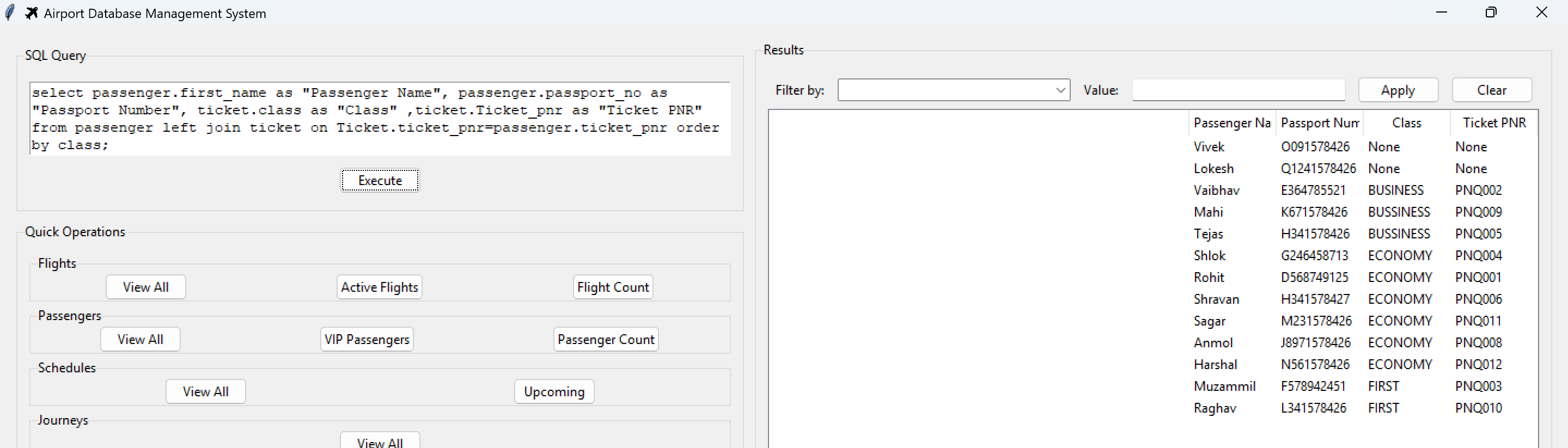
AI-generated content may be incorrect.

1. Show name, passport number, class and ticket PNR of all the passengers.

select passenger.first\_name as "Passenger Name", passenger.passport\_no as "Passport Number", ticket.class as "Class" ,ticket.Ticket\_pnr as "Ticket PNR" from passenger left join ticket on Ticket.ticket\_pnr=passenger.ticket\_pnr order by class;

A screenshot of a computer

AI-generated content may be incorrect.

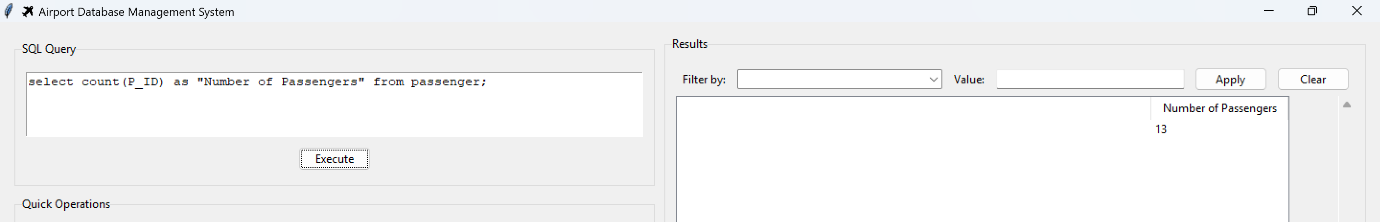


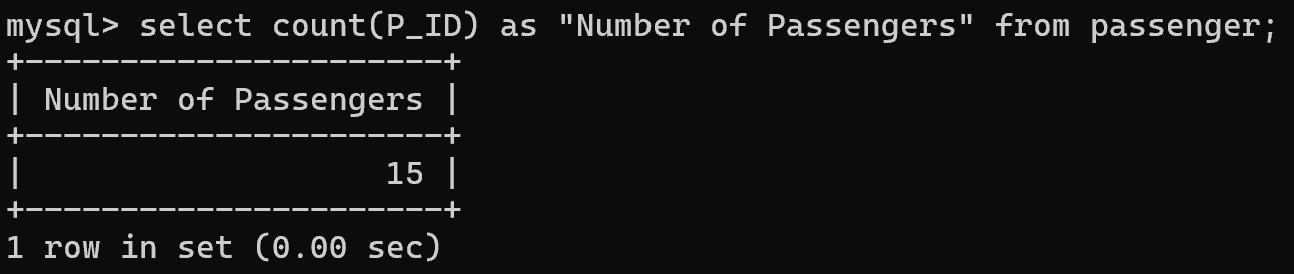
1. Find the Phone number and name of the passenger who is going to Dubai select first\_name as "Name", phone\_no as "Phone Number" from passenger where ticket\_pnr=(select ticket\_pnr from ticket where schedule\_id=(select schedule\_id from schedule where arri\_location='Dubai'));

A black screen with white text

AI-generated content may be incorrect.

1. Find total number of passengers.

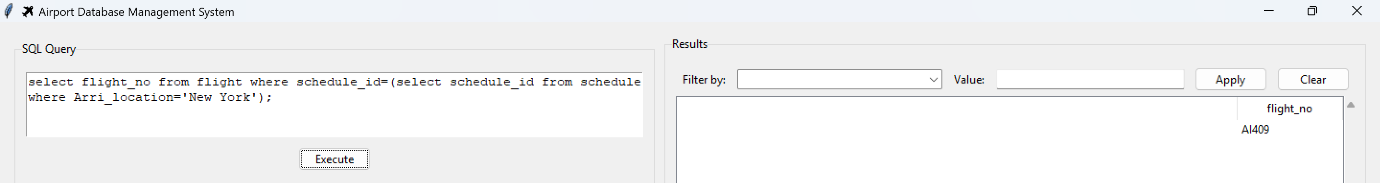
select count(P\_ID) as "Number of Passengers" from passenger;



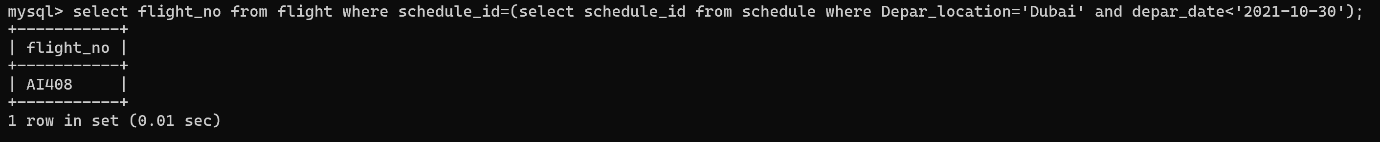
1. Find the flight number going to New York.

A black screen with white text

AI-generated content may be incorrect.select flight\_no from flight where schedule\_id=(select schedule\_id from schedule where Arri\_location='New York');



1. Find the Flights coming from Dubai, before 30/10/21.

select flight\_no from flight where schedule\_id=(select schedule\_id from schedule where Depar\_location='Dubai' and depar\_date<'2021-10-30');

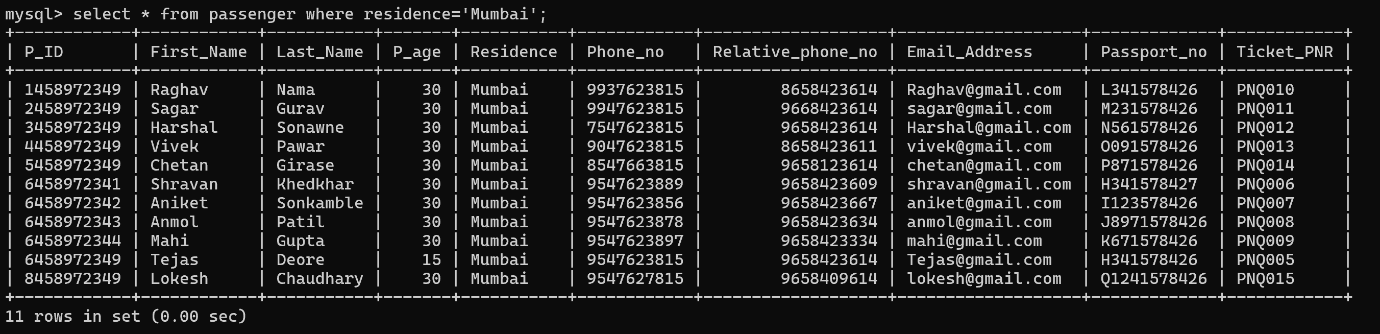
A screenshot of a computer

AI-generated content may be incorrect.

1. Find passengers residing in Mumbai.

select \* from passenger where residence='Mumbai';

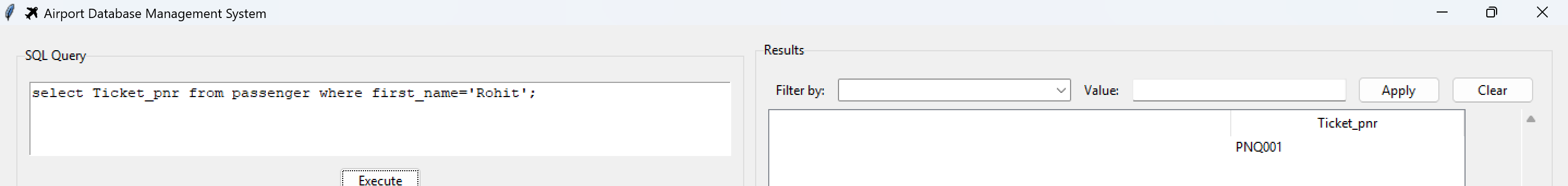
A screenshot of a computer

AI-generated content may be incorrect.

1. Find the ticket PNR of Raghav.

A black screen with white text

AI-generated content may be incorrect.select Ticket\_pnr from passenger where first\_name='Rohit';



1. List the details of customer

whose phone number start with 64.

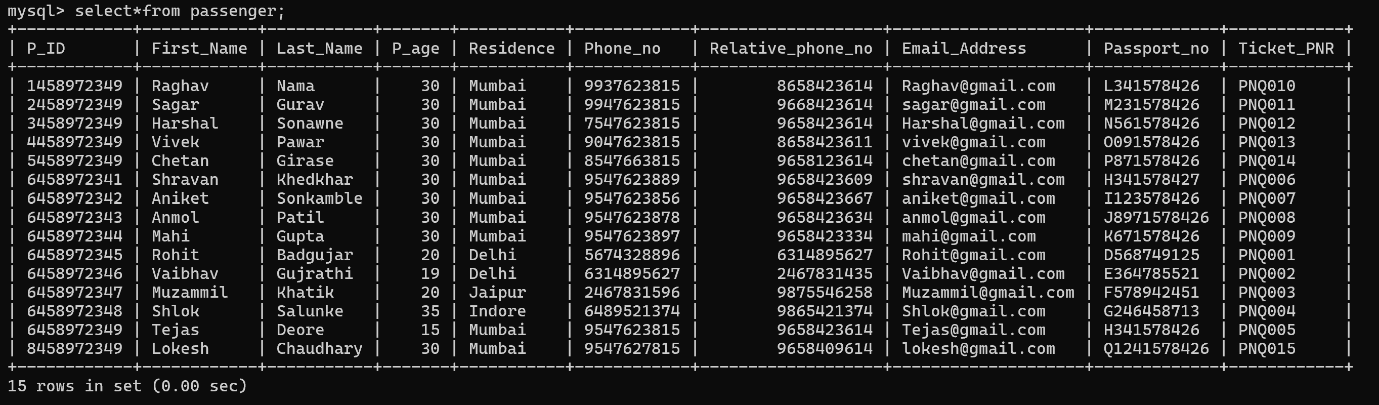
select \* from passenger where phone\_no like '64%';

A screenshot of a computer

AI-generated content may be incorrect.

**VI. Project demonstration**

* **Tools Used:** MySQL 5.5 Command Line Clint
* **Technologies:** SQL, ER Diagram Tools
* **Screenshots:**



A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**VII. Self -Learning beyond classroom**

:

Use of real-life airline system modeling

Learned advanced SQL joins and subqueries

Explored normalization beyond BCNF

Used ER diagram tools for clear design representation

**VIII. Learning from the Project**

Practical understanding of DBMS principles

Team collaboration and project documentation SQL

proficiency and database design experience

**IX. Challenges Faced**

Mapping entity relationships with multiple dependencies

Data insertion errors due to foreign key constraints

Time formatting issues in SQL Server

**X. Conclusion**

The Airport Management System successfully demonstrates a real-world DBMS application. The project enhanced our understanding of relational databases, normalization, and SQL querying, and laid a foundation for more advanced data systems development.