UNIVERSITY OF ASIA PACIFIC PACIc PACIFIC

Department of Computer Science and Engineering (CSE)

**Course Code:** CSE 212

|  |  |
| --- | --- |
| Submitted by: | |
| Name | **Reg No** |
| Raj Kumar Saha | 23101065 |
| Farhana Chowdhury | 23101066 |
| Tithi Ghosh | 23101071 |

**Submitted to:**

**Alif Ruslan**

**Lecturer of CSE, UAP**

**Date of Submission:** 12.06.25

**Project Name: Ticket Management System**

**Section: B1**

**Course Title:** Database Systems Lab

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| Topic | Page No |
| Introduction | 1-2 |
| Tables and Entities | 2-4 |
| ER Diagram | 4-5 |
| Schema Diagram | 6-7 |
| Queries and Outputs | 7-13 |
| CEP |  |
| Conclusion |  |

**Ticket Management System**

Ticket Management System is a database-driven platform developed to streamline and modernize the process of booking, managing, and reselling tickets for various modes of transportation, including buses, trains, and planes. This system reduces manual intervention, minimizes ticketing errors, and improves user convenience by offering real-time access to ticket data and transaction histories. The Ticket Management System provides an intuitive interface for both customers and administrators, with clear roles and permissions to ensure smooth operation.

Each table in the system has been carefully designed to support fast, reliable access to critical information, such as ticket availability, purchase status, travel schedules, and user details. The database is structured in a way that enhances operational efficiency, supports resale functions, and maintains data consistency across the platform. The system is easy to navigate even for users who have limited technical skills. This management system supports:

* **Centralized Ticketing and User-Friendly Management:**

1. Tables like Ticket, Bus, Train, and Plane allow users to book, view, and manage multiple types of transport from a single platform.
2. The system interface is simple and intuitive, making it accessible even for users with limited technical skills.

* **Role-Based Access and Security:**

1. The Admin and Customer tables define user roles, ensuring that only authorized users can perform specific actions like managing users or reselling tickets.
2. Login credentials and foreign key-based control (Control table) provide structured user access and secure role-based interactions.

* **Enhanced Tracking and Resolution:**

1. Every ticket and resale transaction are logged in the Ticket and Resell tables, ensuring full traceability of user activity.
2. Admins can easily monitor and resolve disputes related to ticket ownership or payment using centralized ticket records.

* **Efficient Financial Handling:**

1. The Resell table tracks resale prices, buyer and seller IDs, and ensures correct financial transfers between users.
2. Pricing details in the Ticket table help customers understand travel costs clearly and reduce billing errors.

* **Challenges Faced During Development:**

1. Implementing many-to-many relationships between Ticket and various transport modes (Bus, Train, Plane) required complex database design.
2. Ensuring proper ticket ownership transfer and data integrity during resale was a difficult but crucial part of development.

* **Future Enhancements:**

1. Add QR code-based digital tickets, mobile alerts for schedule changes, and contactless check-ins to improve user experience.
2. Integrate digital wallets and user rating systems to increase trust, speed up transactions, and enhance overall system usability.

Having a Ticket Management System is essential to make travel easier, quicker, and more efficient for everyone. It removes the need for time-consuming booking procedures, paperwork, or dealing with unorganized schedules. Passengers can easily book, manage, or even resell their tickets, while admins can oversee transactions and transport services effectively.

Ultimately, it's about streamlining the process, minimizing confusion, and delivering a hassle-free experience so travelers can focus on their journey, and service providers can ensure smooth operations.

**Tables and Entities:**

To make this database system a success, we have used several tables to structure the system.

They are:

1. **Admin:**

* **ad\_username** (PK)
* **ad\_name, ad\_password, ad\_email**
* **Primary Entity:** Represents the administrators of the system who are responsible for managing tickets and overseeing customer interactions. This table stores login credentials and essential contact details for access control.

1. **Customer:**

* **cus\_id** (PK)
* **cus\_name, cus\_email, cus\_pass, cus\_phone**
* **Primary Entity:** Represents individual users of the system who can book or resell tickets. It holds customer details such as name, email, phone number, and login information.

1. **Bus:**

* **Bus\_id** (PK)
* **Bus\_name, Capacity, Bus\_time, Bus\_type**
* **Primary Entity:** Contains information about available buses in the system. It stores data such as the bus name, schedule, capacity, and type of service.

1. **Train:**

* **train\_id** (PK)
* **train\_name, train\_type, train\_time, capacity**
* **Primary Entity:** Stores data related to train services offered in the system. Includes details like train type, timing, and seat capacity for bookings.

1. **Plane:**

* **plane\_id** (PK)
* **plane\_name, plane\_time**
* **Primary Entity:** Represents available plane services in the system. It includes flight names and scheduled times for travel.

1. **Ticket:**

* **ticket\_id** (PK)
* **cus\_id, ad\_username, Bus\_id, train\_id, plane\_id, ticket\_from, ticket\_to, ticket\_date, ticket\_class, seat\_number, ticket\_price, ticket\_type**
* **Foreign Keys:** cus\_id references **Customer.cus\_id**, ad\_username references **Admin.ad\_username,** Bus\_id references **Bus.Bus\_id**, train\_id references **Train.train\_id**, plane\_id references **Plane.plane\_id**
* **Primary Entity:** Central table representing each travel ticket. Stores journey details, transport type, ticket class, seat info, pricing, and links to customer and admin for tracking and management.

1. **Control:**

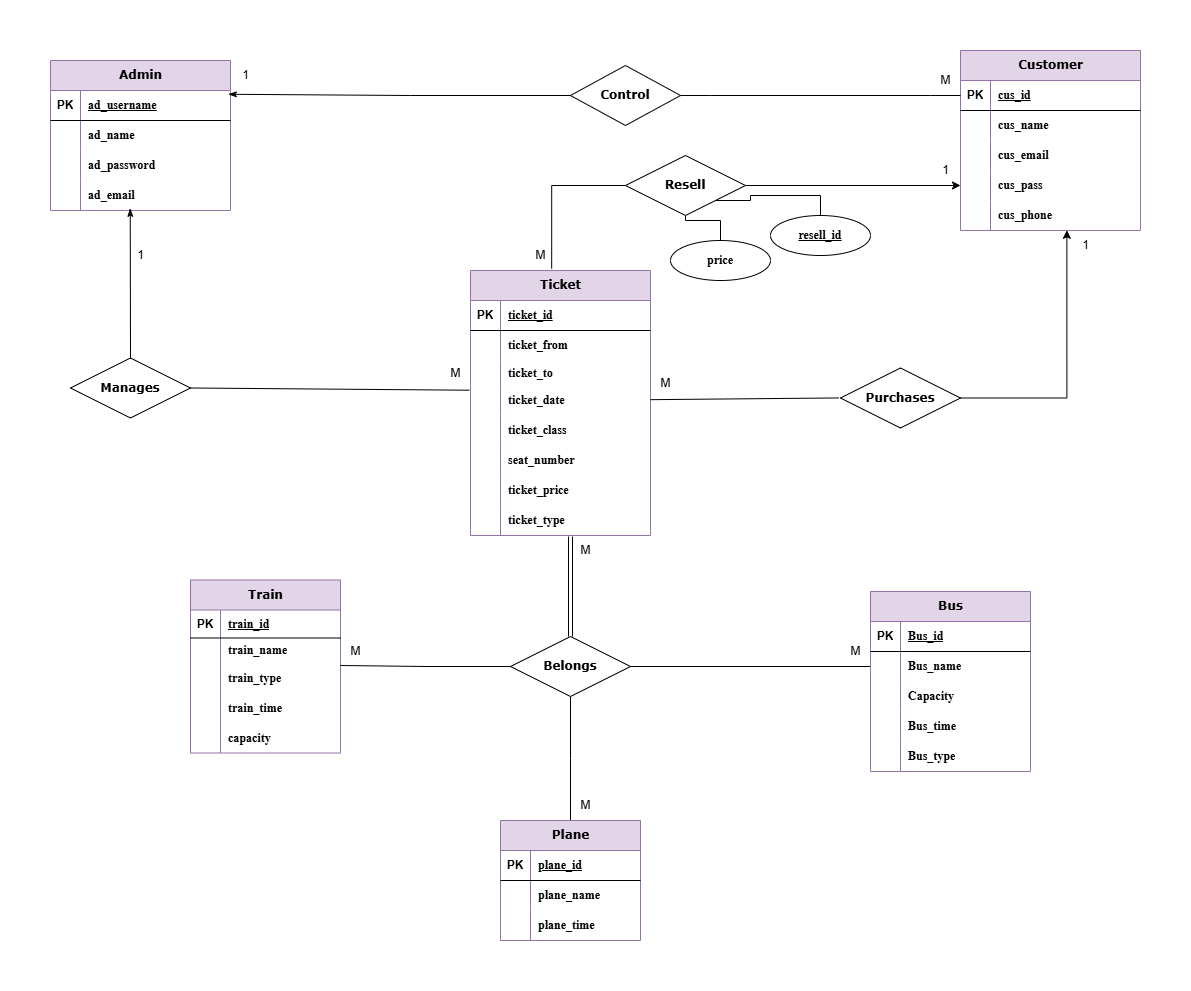
* **cus\_id, ad\_username**
* **Foreign Keys:** cus\_id references **Customer.cus\_id**, ad\_username references **Admin.ad\_username**
* **Relationship Table:** Establishes the connection between customers and their assigned admins. Implements a one-to-many relationship where each admin manages several customers.

1. **Reselling:**

* **resell\_id** (PK)
* **buyer\_cus\_id, seller\_cus\_id, ticket\_id, price**
* **Foreign Keys:** buyer\_cus\_id references **Customer.cus\_id**, seller\_cus\_id references **Customer.cus\_id**, ticket\_id references **Ticket.ticket\_id**
* **Primary Entity:** Records details of ticket resale transactions between customers. Stores buyer and seller information, ticket being resold, and updated pricing for transparency and ownership tracking.

**ER Diagram:**

This ER diagram visually represents the relationships between key entities such as Customers, Tickets, Trains, Buses, Planes, and Admins. It illustrates how customers purchase tickets for various modes of transportation, managed by admins, and how ticket reselling is controlled. The diagram ensures efficient management of ticket bookings, transportation schedules, and customer information by defining the data flow within the system. Entities like Customers and Trains are linked through Tickets, while Admins manage the overall system.

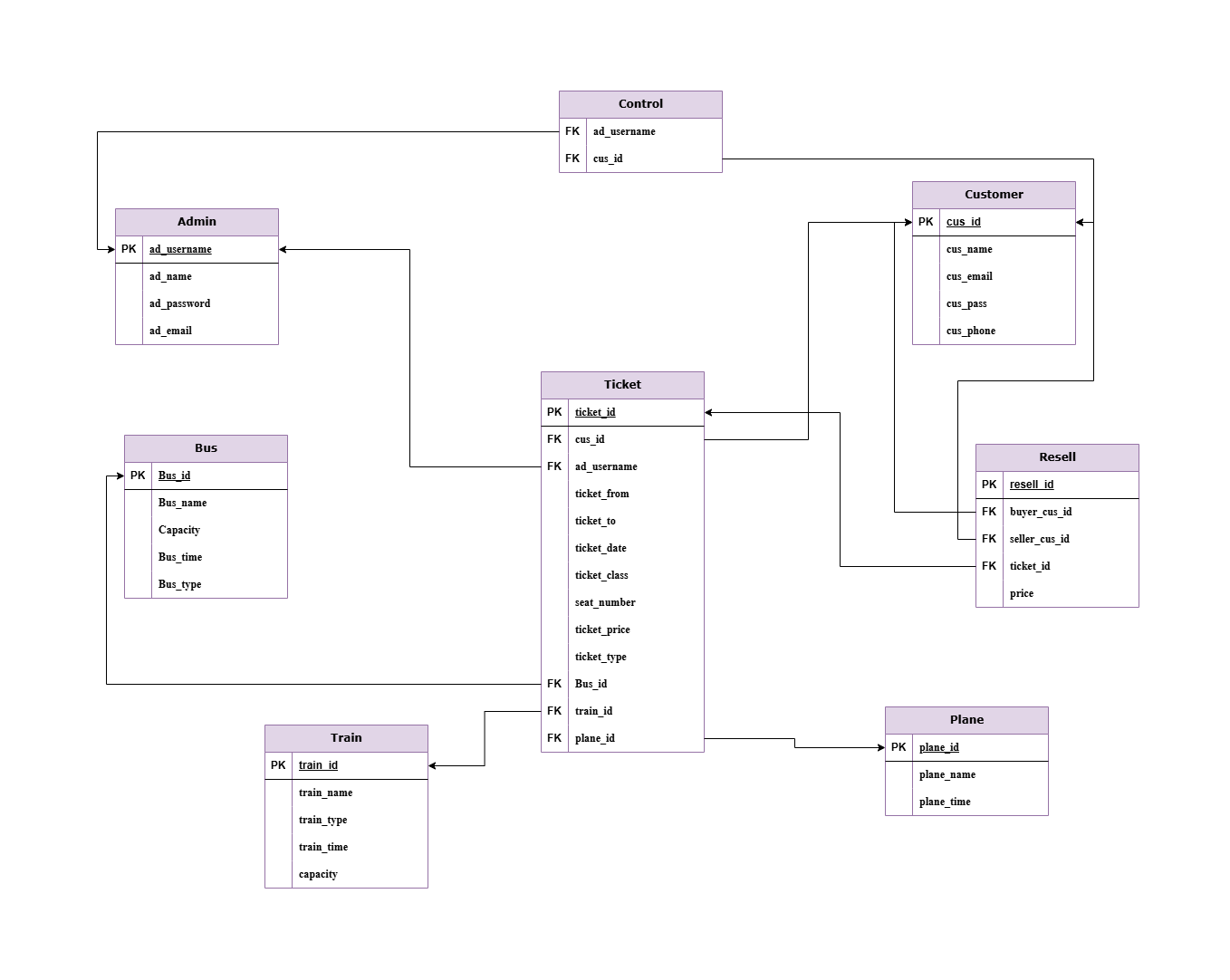


For a better view, visit the link:

<https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&dark=auto#G1v7Q-2NBdkmXOBlYbaW01EB5WN6qUjqK6>

**Schema Diagram:**

This schema diagram provides a detailed structural view of the Ticket Management System’s database. It defines the tables such as Customer, Admin, Ticket, Bus, Train, Plane, Resell, and Control and their key fields, primary keys, and foreign key relationships. The diagram shows how ticket information is connected to customers and various transportation modes, while also managing admin control and ticket reselling. By clearly outlining the organization of data and the links between tables, the schema diagram ensures efficient data storage, retrieval, and integrity within the system.



For a better view, visit the link:

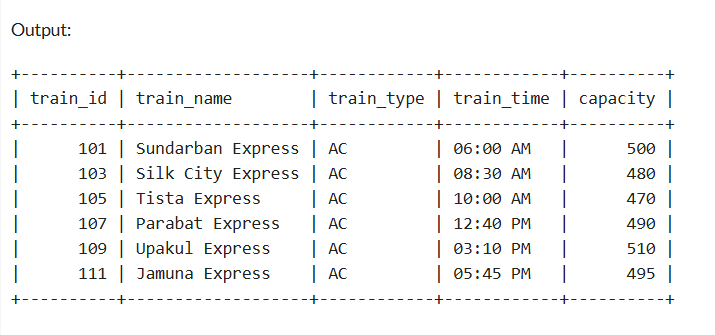
<https://viewer.diagrams.net/?tags=%7B%7D&lightbox=1&highlight=0000ff&edit=_blank&layers=1&nav=1&dark=auto#G1Kp8OAmd68SSQtEbcuqX80c-YkDbdd-7g>

**Queries and Outputs:**

**Simple Queries:**

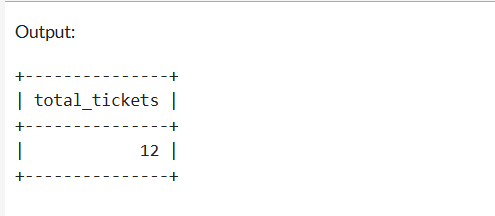
1. **Find all trains that are of AC type.**

**=** SELECT \* FROM Train WHERE train\_type = 'AC';



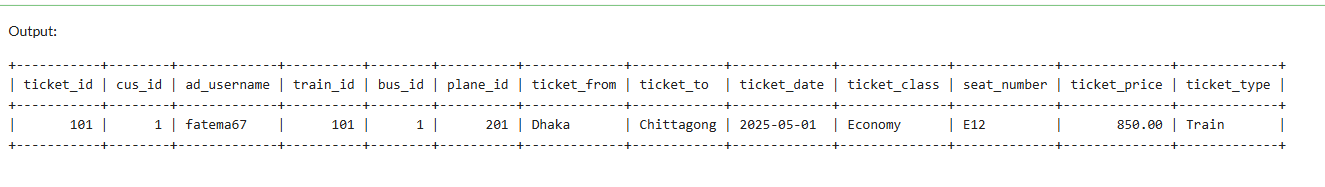
1. **Count total number of tickets.**

**=** SELECT COUNT(\*) AS total\_tickets FROM Ticket;



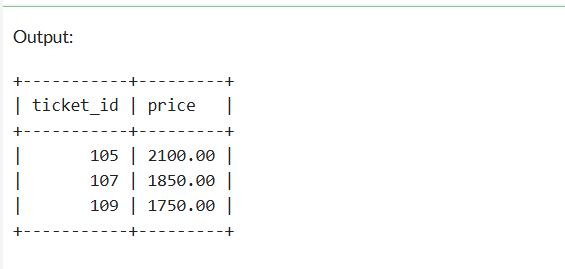
1. **List the tickets sold by Admin 'fatema67’.**

**=** SELECT \* FROM Ticket WHERE ad\_username = 'fatema67';



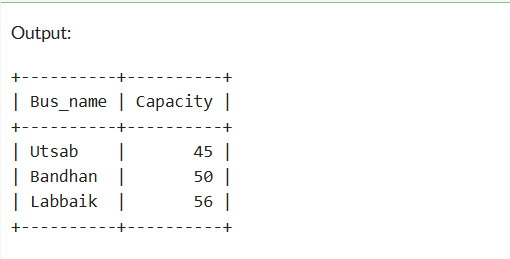
1. **List the ticket IDs and prices for tickets resold for more than 1000 BDT.**

**=** SELECT ticket\_id, price FROM Resell WHERE price > 1000;



1. **Find all the 'Non-AC' buses that have a capacity greater than 40.**

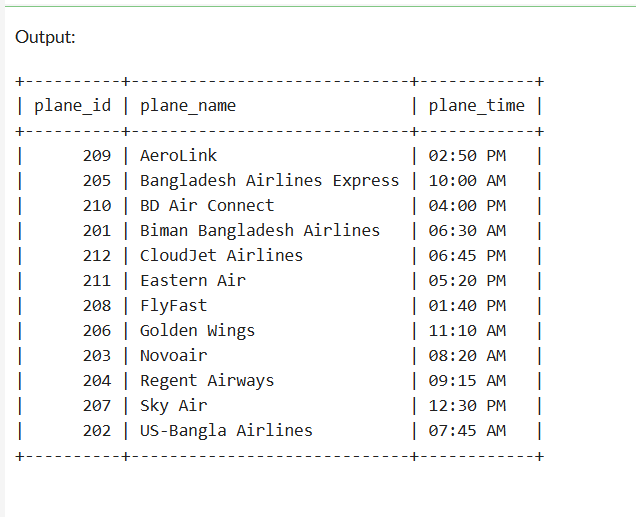
**=** SELECT Bus\_name, Capacity FROM Bus WHERE Bus\_type = 'Non-AC' AND Capacity > 40;



**Intermediate Queries:**

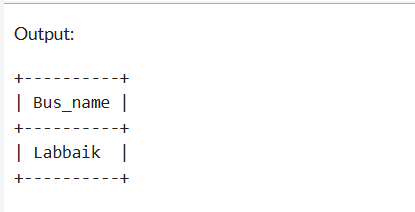
1. **Show all the details of planes from the Plane table and sort them by the plane name in ascending order.**

**=** SELECT \* FROM Plane order by plane\_name asc;

****

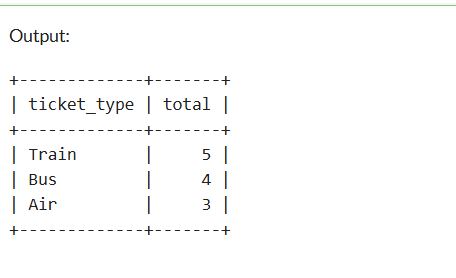
1. **Find the name of the bus with the highest capacity.**

**=** SELECT Bus\_name FROM Bus WHERE Capacity = (SELECT MAX(Capacity) FROM Bus);

****

1. **Total number of tickets sold for each transport type.**

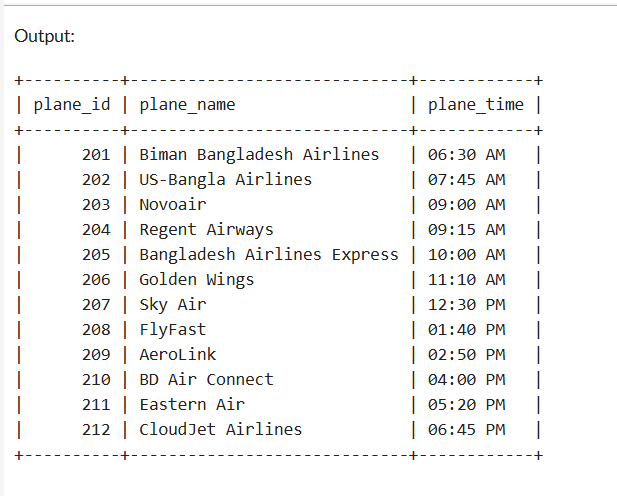
**=** SELECT ticket\_type, COUNT(\*) AS total FROM Ticket GROUP BY ticket\_type;

****

1. **Update plane time for plane\_id 203.**

**=** UPDATE Plane SET plane\_time = '09:00 AM' WHERE plane\_id = 203;

SELECT \* FROM Plane;

****

1. **Add gender column to Customer table and show all data.**

**=** ALTER TABLE Customer ADD gender VARCHAR(100);

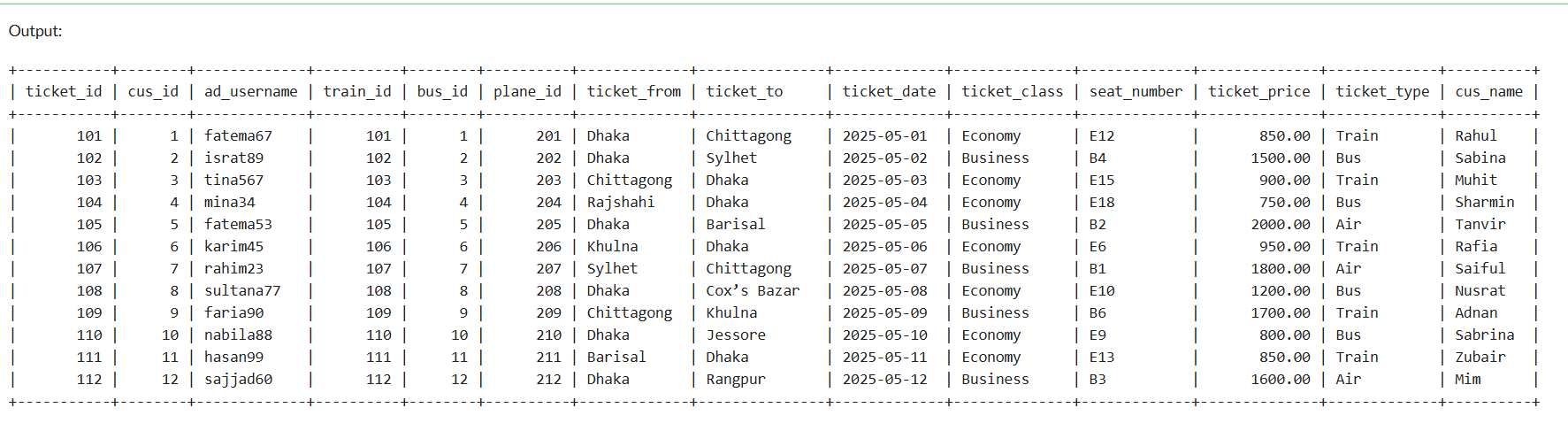
SELECT \* FROM Customer;

****

**Hard Queries:**

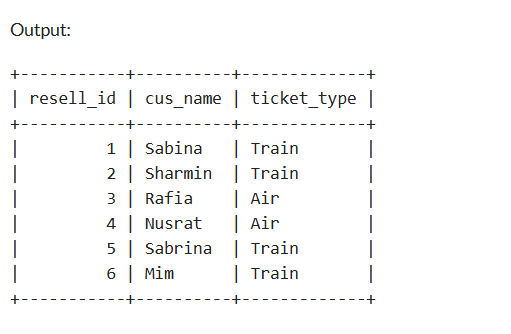
1. **List all ticket details with customer names.**

**=** SELECT Ticket.\*, Customer.cus\_nameFROM Ticket JOIN Customer ON Ticket.cus\_id = Customer.cus\_id;

****

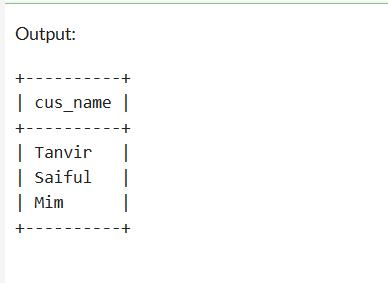
1. **Show resold tickets with ticket type and buyer name.**

**=** SELECT Resell.resell\_id, Customer.cus\_name, Ticket.ticket\_type FROM Resell JOIN Customer ON Resell.buyer\_cus\_id = Customer.cus\_id JOIN Ticket ON Resell.ticket\_id = Ticket.ticket\_id;



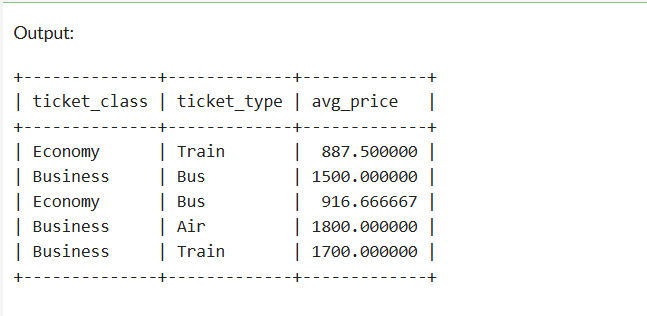
1. **Find names of customers who bought plane tickets.**

**=** SELECT c.cus\_name FROM Customer cJOIN Ticket t ON c.cus\_id = t.cus\_id WHERE t.ticket\_type = 'Air';

****

**14. Get the average ticket price for each class (Economy/Business) and transportation type (Train/Bus/Air).**

**=** SELECT ticket\_class, ticket\_type, AVG(ticket\_price) AS avg\_price FROM Ticket GROUP BY ticket\_class, ticket\_type;

****

**15. Find the customer names and their phone numbers who have a ticket for the plane 'Biman Bangladesh Airlines’.**

**=** SELECT Customer.cus\_name, Customer.cus\_phone FROM Customer JOIN Ticket ON Customer.cus\_id = Ticket.cus\_id JOIN Plane ON Ticket.plane\_id = Plane.plane\_id WHERE Plane.plane\_name = 'Biman Bangladesh Airlines';

