**NIYONZIMA MAURICE 31/01/2025**

**2301000186**

**TICKET MANAGEMENT SYSTEM PROJECT**

**ASSIGNMENT**

**DATABASE MANAGEMENT SYSTEM**

**LECTURE Dr BUGINGO Emmanuel**

**TICKET MANAGEMENT SYSTEM PROJECT**

**DESCRIPTION OF DATABASE TICKET MANAGEMENT SYSTEM**

**Database System: Ticket Management System**

**Overview:**

This database system is designed to manage a ticketing system for various transportation methods (such as flights, buses, and trains). It stores data related to passengers, trips, and tickets. It provides functionality to:

* Manage passenger details
* Record trip information
* Book and manage tickets for passengers
* Support operations such as adding passengers, booking tickets, and managing trips

**Entities & Relationships:**

1. **Passenger:**
   * **Description:** The Passenger table stores information about individuals who will book tickets for trips. This includes personal details like name, date of birth, gender, email, and phone number.
   * **Key Fields:**
     + Passenger\_ID: A unique identifier for each passenger.
     + Name: Full name of the passenger.
     + Date\_of\_Birth: The birth date of the passenger.
     + Gender: The gender of the passenger.
     + Email: A unique email address for communication.
     + Phone\_Number: Contact phone number.
   * **Business Use:** This table allows you to store and retrieve passenger information. Passengers will be referenced when booking tickets for trips.
2. **Trip:**
   * **Description:** The Trip table contains details about trips, including departure and arrival cities, dates, transportation methods (flight, train, bus), and the price per seat.
   * **Key Fields:**
     + Trip\_ID: A unique identifier for each trip.
     + Departure\_City: City where the trip begins.
     + Arrival\_City: City where the trip ends.
     + Departure\_Date: Date and time when the trip is scheduled to depart.
     + Arrival\_Date: Date and time when the trip is expected to arrive.
     + Train\_Bus\_Flight: The mode of transportation (train, bus, or flight).
     + Price\_Per\_Seat: The price for one seat on the trip.
   * **Business Use:** This table helps define and store the trip details. It is used to look up available trips for passengers when booking tickets.
3. **Ticket:**
   * **Description:** The Ticket table stores the details of tickets purchased by passengers for specific trips. It includes ticket status, seat number, price, and the issue date.
   * **Key Fields:**
     + Ticket\_ID: A unique identifier for each ticket.
     + Passenger\_ID: A foreign key referencing the Passenger table (to link the ticket to a specific passenger).
     + Trip\_ID: A foreign key referencing the Trip table (to associate the ticket with a specific trip).
     + Issue\_Date: The date and time when the ticket was issued.
     + Price: The total price paid for the ticket.
     + Seat\_Number: The seat assigned to the passenger for the trip.
     + Status: The current status of the ticket (e.g., "Booked", "Cancelled").
   * **Business Use:** This table manages ticket bookings, allowing you to track which passengers have booked tickets for which trips. It also keeps track of seat assignments and ticket statuses.

**Database Operations and Procedures:**

The system also supports various stored procedures and triggers that help manage the operations and maintain data integrity:

* **Stored Procedures:**
  + **AddPassenger:** Adds a new passenger to the system.
  + **BookTicket:** Allows a passenger to book a ticket for a specific trip.
  + **CancelTicket:** Allows a ticket to be canceled.
  + **AddTrip:** Adds a new trip to the system.
  + **UpdateTrip:** Updates details of an existing trip.
  + **DeleteTrip:** Deletes a trip from the system.
* **Triggers:**
  + **AfterInsertPassenger:** After a new passenger is added, log the action into a log table.
  + **AfterInsertTrip:** After a new trip is added, log the action into a log table.
  + **AfterInsertTicket:** After a new ticket is booked, log the action into a log table.

**Use Cases and Workflow:**

1. **Passenger Registration:**
   * When a new passenger registers, their details are entered into the Passenger table. The system tracks their personal information, allowing them to later book tickets.
2. **Trip Management:**
   * Administrators can add new trips (flights, buses, or train rides) to the system. Each trip contains essential information such as departure and arrival cities, dates, and prices.
3. **Ticket Booking:**
   * When a passenger books a ticket, the Ticket table is updated with the relevant details, including the passenger ID, trip ID, price, seat number, and status.
4. **Ticket Cancellation:**
   * If a passenger cancels a ticket, the ticket's status is updated to "Cancelled," and any necessary business rules are triggered.
5. **Logging & Auditing:**
   * The system uses triggers to log important actions (like adding passengers, trips, and tickets) into a log table, which can be used for auditing purposes.

**Data Integrity and Constraints:**

1. **Foreign Keys:**
   * Ticket.Passenger\_ID references Passenger.Passenger\_ID (ensuring each ticket is linked to a valid passenger).
   * Ticket.Trip\_ID references Trip.Trip\_ID (ensuring each ticket is linked to a valid trip).
2. **Unique Constraints:**
   * Passenger.Email is unique to ensure no two passengers can share the same email address.
3. **Cascading Deletes:**
   * If a passenger or a trip is deleted, the associated tickets are automatically deleted, ensuring referential integrity.

**Possible Extensions:**

1. **Notifications:**
   * The system could integrate with an external notification service to send email or SMS notifications to passengers when their tickets are booked, canceled, or updated.
2. **Seat Availability Management:**
   * You could extend the system to track seat availability dynamically, ensuring no double bookings for the same seat.
3. **Pricing and Discounts:**
   * The system can be extended to handle dynamic pricing or promotional discounts, which could modify the Price\_Per\_Seat value in the Trip table.

This **Ticket Management System Database** serves as the backbone of any transportation booking system, allowing for efficient management of passengers, trips, and tickets. It supports essential operations such as booking tickets, managing trips, and keeping track of ticket statuses while ensuring data integrity through foreign keys, unique constraints, and cascading deletes

ERD (Entities Relationship diagram)

**Entities:**

1. **Passenger**
   * **Attributes**:
     + Passenger\_ID (Primary Key)
     + Name
     + Date\_of\_Birth
     + Gender
     + Email (Unique)
     + Phone\_Number
2. **Trip**
   * **Attributes**:
     + Trip\_ID (Primary Key)
     + Departure\_City
     + Arrival\_City
     + Departure\_Date
     + Arrival\_Date
     + Train\_Bus\_Flight (Mode of transport)
     + Price\_Per\_Seat
3. **Ticket**
   * **Attributes**:
     + Ticket\_ID (Primary Key)
     + Passenger\_ID (Foreign Key referencing Passenger)
     + Trip\_ID (Foreign Key referencing Trip)
     + Issue\_Date
     + Price
     + Seat\_Number
     + Status

**Relationships:**

* **Passenger to Ticket**:
  + **One-to-Many (1:N)**: A passenger can have multiple tickets, but each ticket is associated with exactly one passenger. This is established through the Passenger\_ID in the Ticket table, which references the Passenger\_ID in the Passenger table.
* **Trip to Ticket**:
  + **One-to-Many (1:N)**: A trip can have multiple tickets associated with it, but each ticket is linked to exactly one trip. This is represented by the Trip\_ID in the Ticket table, which references the Trip\_ID in the Trip table.

Passenger Ticket Trip

Passenger\_ID (PK) |<------| Ticket\_ID (PK) |------>| Trip\_ID (PK)

Name Passenger\_ID (FK) Departure\_City

Date\_of\_Birth Trip\_ID (FK) Arrival\_City

Gender Issue\_Date Departure\_Date

Email Price Arrival\_Date

Phone\_Number Seat\_Number Train\_Bus\_Flight

Status Price\_Per\_Seat

**Key Points in the ERD:**

1. **Passenger and Ticket**:
   * **One-to-Many Relationship**: A passenger can have multiple tickets, but each ticket corresponds to one specific passenger. The relationship is represented by the Passenger\_ID in the Ticket table (Foreign Key).
2. **Trip and Ticket**:
   * **One-to-Many Relationship**: A trip can have multiple tickets, but each ticket belongs to a single trip. The relationship is represented by the Trip\_ID in the Ticket table (Foreign Key).
3. **Primary Keys (PK)**:
   * Passenger\_ID in the Passenger table.
   * Trip\_ID in the Trip table.
   * Ticket\_ID in the Ticket table.
4. **Foreign Keys (FK)**:
   * Passenger\_ID in the Ticket table references Passenger\_ID in the Passenger table.
   * Trip\_ID in the Ticket table references Trip\_ID in the Trip table.

**ERD Creation Tools:**

If you'd like to generate this ERD visually, you can use online tools like:

* **draw.io (diagrams.net)**: Allows you to drag and drop entities, define primary and foreign keys, and create the relationships between them.
* **Lucidchart**: Another popular tool for designing ERDs with an easy-to-use interface.
* **MySQL Workbench**: If you're using MySQL, you can generate an ERD directly from your database schema using this tool.

PDM PHYSICAL DATA MODEL

**Physical Data Model for Passenger Table:**

**Table Name: Passenger**

| **Column Name** | **Data Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| Passenger\_ID | INT | Unique identifier for the passenger | PRIMARY KEY, AUTO\_INCREMENT |
| Name | VARCHAR(255) | Full name of the passenger | NOT NULL |
| Date\_of\_Birth | DATE | Birth date of the passenger | NOT NULL |
| Gender | VARCHAR(10) | Gender of the passenger (e.g., Male, Female, Other) | NULLABLE |
| Email | VARCHAR(100) | Email address of the passenger | NOT NULL, UNIQUE |
| Phone\_Number | VARCHAR(15) | Contact phone number of the passenger | NULLABLE |

DATA DICTIONARY OF TICKET MANAGEMENT SYSTEM

**1. Passenger Table**

| **Field Name** | **Data Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| Passenger\_ID | INT | Unique identifier for each passenger. | PRIMARY KEY, AUTO\_INCREMENT |
| Name | VARCHAR(255) | Full name of the passenger. | NOT NULL |
| Date\_of\_Birth | DATE | Birth date of the passenger. | NOT NULL |
| Gender | VARCHAR(10) | Gender of the passenger (e.g., Male, Female, etc.) | NULLABLE |
| Email | VARCHAR(100) | Email address of the passenger. | NOT NULL, UNIQUE |
| Phone\_Number | VARCHAR(15) | Phone number of the passenger. | NULLABLE |

**Description:**  
This table stores information about passengers who will book tickets. The Passenger\_ID is the unique key used to identify each passenger. The Email column is required and must be unique for each passenger. The Phone\_Number and Gender columns are optional.

**2. Trip Table**

| **Field Name** | **Data Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| Trip\_ID | INT | Unique identifier for each trip. | PRIMARY KEY, AUTO\_INCREMENT |
| Departure\_City | VARCHAR(100) | The city where the trip departs from. | NOT NULL |
| Arrival\_City | VARCHAR(100) | The city where the trip arrives at. | NOT NULL |
| Departure\_Date | DATETIME | The scheduled departure date and time of the trip. | NOT NULL |
| Arrival\_Date | DATETIME | The scheduled arrival date and time of the trip. | NOT NULL |
| Train\_Bus\_Flight | VARCHAR(50) | Mode of transportation (e.g., Train, Bus, Flight). | NOT NULL |
| Price\_Per\_Seat | DECIMAL(10, 2) | The price of a single seat on the trip. | NOT NULL |

**Description:**  
This table stores information about trips available for booking. The Trip\_ID is the unique key for each trip. The table includes details such as the departure and arrival cities, dates, transportation mode, and price per seat. All fields are required.

**3. Ticket Table**

| **Field Name** | **Data Type** | **Description** | **Constraints** |
| --- | --- | --- | --- |
| Ticket\_ID | INT | Unique identifier for each ticket. | PRIMARY KEY, AUTO\_INCREMENT |
| Passenger\_ID | INT | The ID of the passenger who purchased the ticket. | FOREIGN KEY (references Passenger.Passenger\_ID) |
| Trip\_ID | INT | The ID of the trip that the ticket is for. | FOREIGN KEY (references Trip.Trip\_ID) |
| Issue\_Date | DATETIME | The date and time when the ticket was issued. | NOT NULL |
| Price | DECIMAL(10, 2) | The price paid for the ticket. | NOT NULL |
| Seat\_Number | VARCHAR(10) | The seat number assigned to the passenger. | NULLABLE |
| Status | VARCHAR(50) | The current status of the ticket (e.g., "Booked", "Cancelled"). | NULLABLE |

**Description:**  
This table stores information about the tickets purchased by passengers. It links to both the Passenger and Trip tables via foreign keys. The Ticket\_ID is the primary key. Other fields include the issue date, price, seat number, and ticket status.

**Relationships Between Tables:**

1. **Passenger → Ticket (1:N)**
   * A passenger can have multiple tickets, but each ticket is associated with exactly one passenger. This is enforced through the Passenger\_ID in the Ticket table, which is a foreign key that references the Passenger\_ID in the Passenger table.
2. **Trip → Ticket (1:N)**
   * A trip can have multiple tickets associated with it, but each ticket corresponds to exactly one trip. This is enforced through the Trip\_ID in the Ticket table, which is a foreign key that references the Trip\_ID in the Trip table.

**Example Data for the Tables:**

**Passenger Table:**

| **Passenger\_ID** | **Name** | **Date\_of\_Birth** | **Gender** | **Email** | **Phone\_Number** |
| --- | --- | --- | --- | --- | --- |
| 1 | Alice Johnson | 1985-06-15 | Female | alice.johnson@email.com | 555-1234 |
| 2 | Bob Smith | 1990-09-10 | Male | bob.smith@email.com | 555-5678 |
| 3 | Charlie Brown | 1995-01-25 | Male | charlie.brown@email.com | 555-8765 |
| 4 | Diana Ross | 1980-03-02 | Female | diana.ross@email.com | 555-2345 |
| 5 | Eva Green | 1992-11-30 | Female | eva.green@email.com | 555-3456 |

**Trip Table:**

| **Trip\_ID** | **Departure\_City** | **Arrival\_City** | **Departure\_Date** | **Arrival\_Date** | **Train\_Bus\_Flight** | **Price\_Per\_Seat** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | New York | Los Angeles | 2025-03-01 08:00:00 | 2025-03-01 14:00:00 | Flight | 300.00 |
| 2 | London | Paris | 2025-03-05 09:00:00 | 2025-03-05 11:00:00 | Train | 150.00 |
| 3 | Berlin | Amsterdam | 2025-03-10 10:00:00 | 2025-03-10 12:00:00 | Bus | 50.00 |
| 4 | Tokyo | Osaka | 2025-03-15 06:00:00 | 2025-03-15 08:00:00 | Flight | 250.00 |
| 5 | Sydney | Melbourne | 2025-03-20 07:00:00 | 2025-03-20 09:00:00 | Flight | 200.00 |

**Ticket Table:**

| **Ticket\_ID** | **Passenger\_ID** | **Trip\_ID** | **Issue\_Date** | **Price** | **Seat\_Number** | **Status** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 2025-02-01 10:00:00 | 300.00 | A1 | Booked |
| 2 | 2 | 2 | 2025-02-02 12:00:00 | 150.00 | B3 | Booked |
| 3 | 3 | 3 | 2025-02-03 14:00:00 | 50.00 | C5 | Cancelled |
| 4 | 4 | 4 | 2025-02-04 16:00:00 | 250.00 | D2 | Booked |
| 5 | 5 | 5 | 2025-02-05 18:00:00 | 200.00 | E4 | Booked |

**Constraints and Rules:**

* **Primary Key Constraints**:
  + Passenger\_ID (Passenger)
  + Trip\_ID (Trip)
  + Ticket\_ID (Ticket)
* **Foreign Key Constraints**:
  + Ticket.Passenger\_ID references Passenger.Passenger\_ID
  + Ticket.Trip\_ID references Trip.Trip\_ID
* **Unique Constraints**:
  + Passenger.Email is unique to prevent duplicate emails.
* **Cascading Deletes**:
  + If a Passenger or Trip is deleted, any associated Ticket records are automatically deleted.