

DBMS

Mini Project

Assignment

Semester -4th sem

Subject code -css453

Submission date -8th April, 2025

Group-16

# Group Members

-23CS8077 P.Surya Teja

-23CS8078 B.Nikhileswar Reddy

-23CS8079 K.Jaswika

-23CS8080 M.Vindhya

# Project

# Name:

# Database for Online Ticket Booking System

# Statement:

The database should manage users, ticket bookings and payment details

# Objective:

The objective of an online ticket booking database is to create well structured tables and perform operations on the Relations to get required data

# Goal of the Project

It is to design and implement a relational database for Online Ticket Booking System that supports users to perform some operations on the data to extract a structured information

# System Design

The online Ticket Booking System is designed using the principles of relational database management.

The key Relations of the system are Users, Tickets and Payments. These entities are modeled as individual tables and operations are done such that they ensure data consistency

# Tables

-Users

For unique identification of the user

-UserID

Other additional details about the user

-Name

-Email

-Phone Number

-Tickets

For unique identification of Ticket

-TicketID

Use of foreign key to identify user

-UserID

Specifications of the ticket

-EventName

-SeatNumber

-Price

-Payments

For unique identification of Payment

-PaymentID

Use of foreign key to identify user and ticket

-UserID

-TicketID

Details

-Amount

-PaymentDate

# Operations

• Retrieve all ticket bookings by a specific user

• List all upcoming events

• Update ticket price for a specific event

• Find the total revenue from ticket sales

• Delete a user’s booking and payment details

• Retrieve all payments made within a specific date range

• Find the most frequently booked event

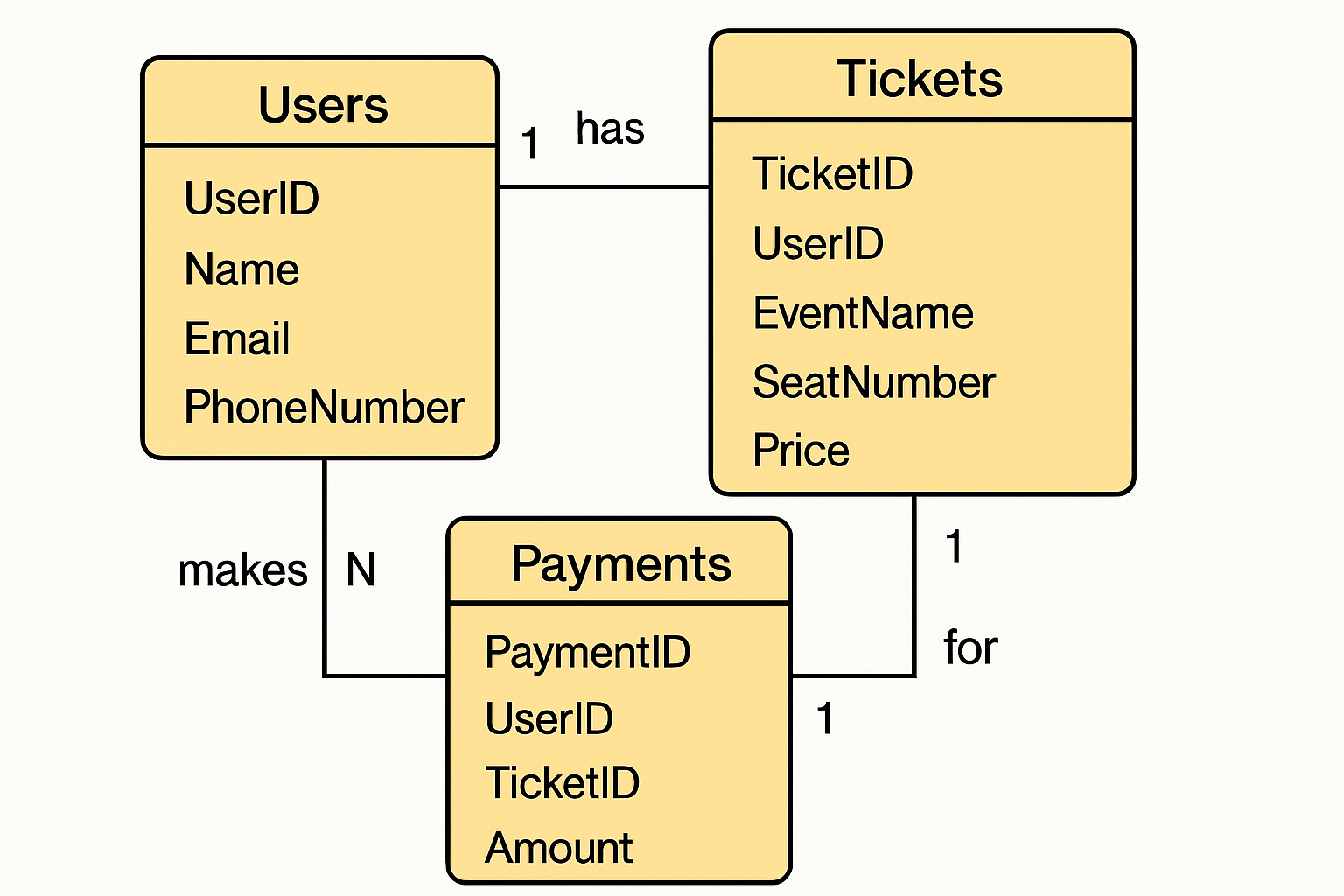
• Identify users who booked more than three tickets

• Generate a report of total revenue by event

• Retrieve the top five highest-paying customers

# ER Diagram

Let’s understand the problem with an ER(Entity-Relation) Diagram



We use MySQL for table creation and query execution

# Table Creation Scripts

# Users Table

create table Users

(

UserID int primary key,

Name varchar(30),

Email varchar(30),

PhoneNumber char(10) check(length(PhoneNumber)=10)

);

# Tickets Table

create table Tickets

(

TicketID int primary key,

UserID int,

EventName varchar(30),

SeatNumber int,

Price int,

Foreign key (UserID) references Users(UserID)

on delete cascade

on update cascade

);

# Payments Table

create table Payments

(

PaymentID int primary key,

UserID int,

TicketID int,

Amount int,

PaymentDate date check(PaymentDate <=current\_date),

Foreign key (UserID) references Users(UserID)

on delete cascade

on update cascade,

Foreign key (TicketID) references Tickets(TicketID)

on delete cascade

on update cascade

);

# Working on sample data

# Inserting values into Users table

insert into Users values

(1,'surya','surya@gmail.com','9398294616'),

(2,'nikhileswar','nikhileswar@gmail.com','7893615681'),

(3,'jaswika','jaswika@gmail.com','6301780659'),

(4,'vindhya','vindhya@gmail.com','9888933372');



# Inserting values into Tickets table

insert into Tickets values

(1234,1,"Dgp\_Hwh",131,345),

(2345,2,"Dgp\_Hwh",156,165),

(2680,3,"Dgp\_Hwh",119,210),

(2681,4,"Dgp\_Hwh",120,210),

(3456,3,"Hwh\_Krv",80,1740),

(4567,4,"Hwh\_Krv",44,1740),

(7890,1,"Hwh\_Che",51,1460),

(7892,2,"Hwh\_Che",78,1450),

(5796,3,"Krv\_Hyb",79,1100),

(9753,2,"Che\_Bza",78,900),

(9953,2,"Bza\_bha",10,100);

# 

# Inserting values into Payments table

insert into Payments values

(654789,1,1234,385,"2025-03-02"),

(123456,2,2345,205,"2025-03-02"),

(891230,3,2680,250,"2025-03-01"),

(891302,4,2681,250,"2025-03-01"),

(234567,3,3456,1800,"2025-03-01"),

(345678,4,4567,1800,"2025-03-01"),

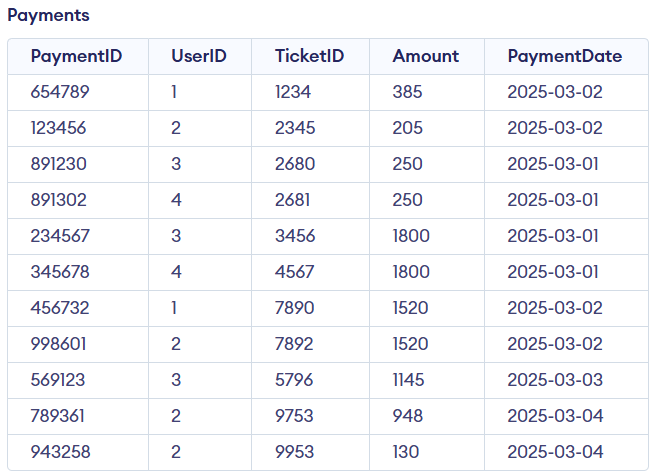
(456732,1,7890,1520,"2025-03-02"),

(998601,2,7892,1520,"2025-03-02"),

(569123,3,5796,1145,"2025-03-03"),

(789361,2,9753,948,"2025-03-04"),

(943258,2,9953,130,"2025-03-04");



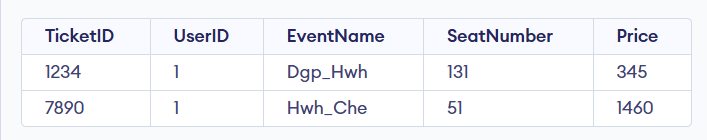
# Queries

## • Retrieve all ticket bookings by a specific user

select \* from Tickets

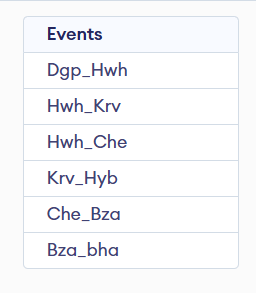
where UserID=<User\_ID>;

for User\_ID=1



## • List all upcoming events

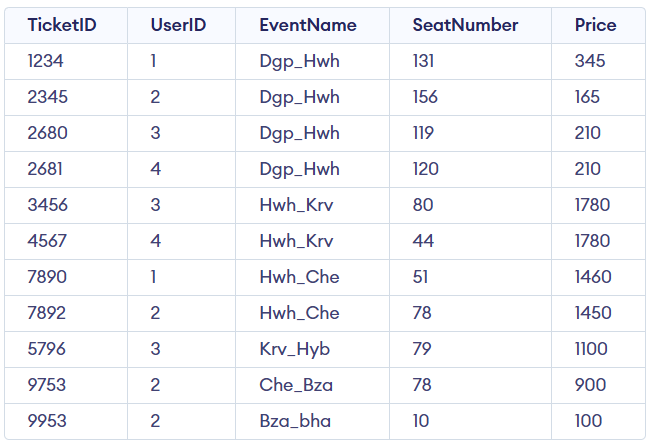
select distinct EventName as Events from Tickets;



# • Update ticket price for a specific event

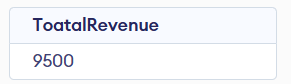
update Tickets set price=<new\_price> where EventName=<Name\_of\_Event>;

for new\_price=1780 and Name\_of\_Event=”Hwh\_Krv”



## • Find the total revenue from ticket sales

select sum(price) as ToatalRevenue from Tickets;

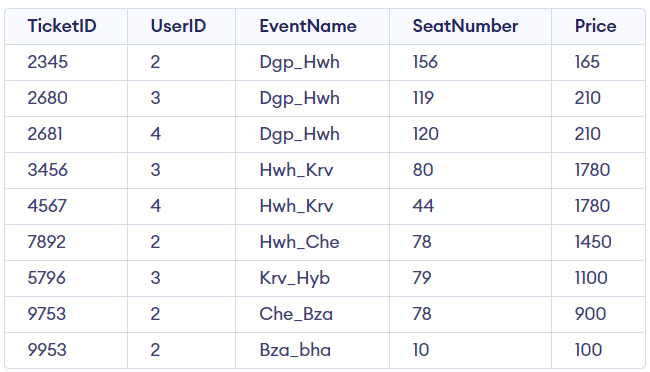


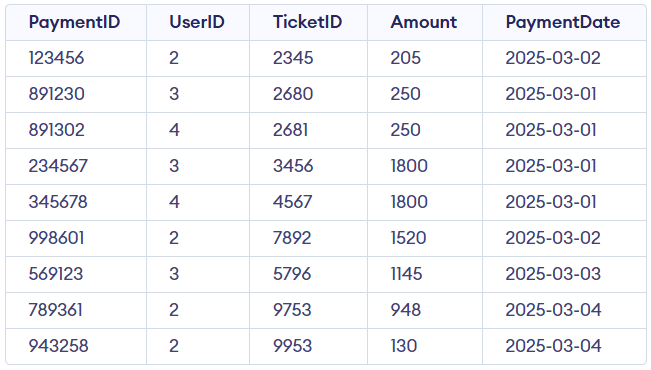
## • Delete a user’s booking and payment details

delete from Payments where UserID=<User\_ID>;

delete from Tickets where UserID=<USER\_ID>;

for User\_ID=1



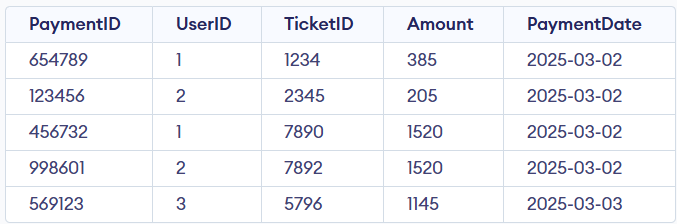


## • Retrieve all payments made within a specific date range

select \* from Payments

where PaymentDate between <start\_date> and <end\_date>;

for strat\_date=’2025-03-02’ and end\_date=’2025-03-03’



## • Find the most frequently booked event

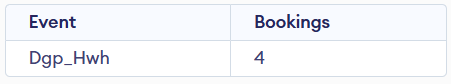
select EventName as Event,count(EventName) as Bookings

from Tickets

group by EventName

order by Bookings desc

limit 1;



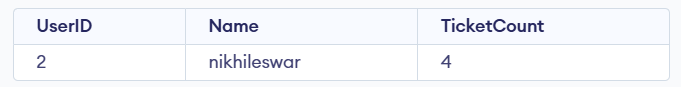
## • Identify users who booked more than three tickets

select U.UserID,U.Name,count(T.EventName) as TicketCount

from Tickets as T join Users as U on U.UserID=T.UserID

group by U.UserID

having TicketCount>3;

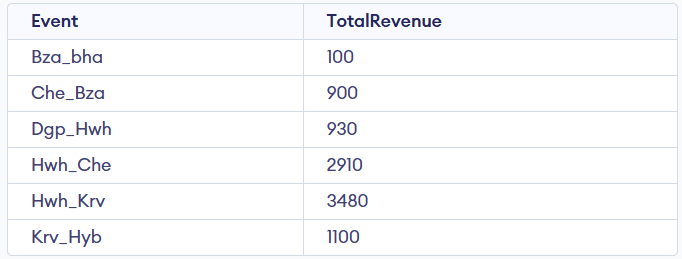


## • Generate a report of total revenue by event

select EventName as Event,sum(price) as TotalRevenue

from Tickets

group by EventName;



## • Retrieve the top five highest-paying customers

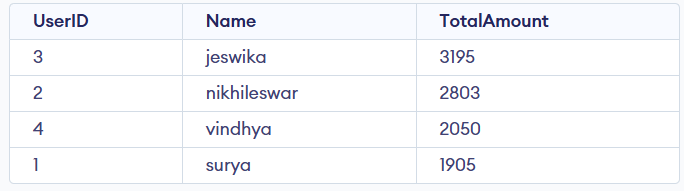
select U.UserID,U.Name,sum(P.Amount) as TotalAmount

from Payments as P join Users as U on P.UserID=U.UserID

group by U.UserID

order by TotalAmount desc

limit 5;



# Implementation and Github

The above tables creation, adding values to the tables and implementation of operations on the tables are saved in

# Github Repository:

# <https://github.com/nit-dgp-cse-gp-16/Database-for-Online-Ticket-Booking-System>

# Group Members:

[Suryatejapotnuru](https://github.com/Suryatejapotnuru)

[b-nikhileswar-reddy (B.Nikhileswar Reddy)](https://github.com/b-nikhileswar-reddy)

[JaswikaKilaru (jaswika kilaru)](https://github.com/JaswikaKilaru)

[VindhyaMuvva](https://github.com/VindhyaMuvva)