# Database Design Assignment

```
CREATE DATABASE IF NOT EXISTS BikeRental:
USE BikeRental;
CREATE TABLE Users (
  UserID INT AUTO_INCREMENT PRIMARY KEY,
  Name VARCHAR(100) NOT NULL,
  Email VARCHAR(100) NOT NULL UNIQUE.
  Phone VARCHAR(20) NOT NULL,
  Address VARCHAR(255),
  RegistrationDate DATE,
  Password VARCHAR(255) NOT NULL,
  CONSTRAINT chk_email CHECK (Email LIKE '%@%.%')
SELECT * FROM Users;
 Edit: 🚰 📆 📙 Export/Import: 📳 📸 | Wrap Cell Content: 🏗
     UserID Name
                                                                     Address
                                                                                               RegistrationDate Password
                                                      123-456-7890 123 Main St, City, Country
             John Smith
                             john.smith@email.com
                                                                                               2025-03-14
                                                                                                               password 123

        John Smith
        john.smith@email.com
        123-456-7890
        123 Main St, City, Country
        2025-03-14

        Emma Johnson
        emma.johnson@email.com
        234-567-8901
        456 Elm St, City, Country
        2025-03-14

                                                                                                              securepass456
                            michael.brown@email.com 345-678-9012 789 Oak St. City. Country
                                                                                                               brownie 789
             Michael Brown
                                                                                              2025-03-14
                            sophia.davis@email.com 456-789-0123 101 Pine St, City, Country 2025-03-14
                                                                                                           sophia2023
            Sophia Davis
             William Wilson
                             william.wilson@email.com
                                                     567-890-1234 202 Maple St, City, Country
                                                                                              2025-03-14
                                                                                                               willnower555
                           olivia.taylor@email.com 678-901-2345 303 Cedar St, City, Country 2025-03-14 oliveoi777
       Olivia Taylor
             James Anderson james.anderson@email.com 789-012-3456 404 Birch St, City, Country
                                                                                               2025-03-14
                                                                                                               jamesrules888
             Ava Martinez ava.martinez@email.com 890-123-4567 505 Walnut St, City, Country 2025-03-14
                                                                                                              avamartinez999
             Benjamin Lee
                            benjamin.lee@email.com
                                                      901-234-5678
                                                                     606 Spruce St, City, Country 2025-03-14
                                                                                                               benjilee 101
             Mia Rodriguez mia.rodriguez@email.com 012-345-6789
                                                                    707 Fir St, City, Country
                                                                                               2025-03-14
     10
                                                                                                               miarodriguez202
    NULL
DELIMITER //
CREATE TRIGGER before_insert_users
BEFORE INSERT ON Users
FOR EACH ROW
BEGIN
 IF NEW.RegistrationDate IS NULL THEN
   SET NEW.RegistrationDate = CURDATE();
 FND IF:
END//
DELIMITER;
-- Create Bikes Table
CREATE TABLE Bikes (
  BIKEID INT AUTO_INCREMENT PRIMARY KEY,
  Model VARCHAR(100) NOT NULL,
  Type VARCHAR(50) NOT NULL,
  HourlyRate DECIMAL(10, 2) NOT NULL,
  Status ENUM('Available', 'Rented', 'Maintenance') DEFAULT 'Available',
  Location VARCHAR(100) NOT NULL,
  YearManufactured INT NOT NULL,
  CONSTRAINT chk_year CHECK (YearManufactured >= 2000)
select * from Bikes;
-- Create Bookings Table
CREATE TABLE Bookings (
  BookingID INT AUTO_INCREMENT PRIMARY KEY,
  UserID INT NOT NULL,
  BikeID INT NOT NULL,
  StartTime DATETIME NOT NULL,
  EndTime DATETIME NOT NULL,
  Status ENUM('Pending', 'Active', 'Completed', 'Cancelled') DEFAULT 'Pending',
  TotalAmount DECIMAL(10, 2) NOT NULL,
  Notes TEXT.
  FOREIGN KEY (UserID) REFERENCES Users(UserID) ON DELETE CASCADE,
  FOREIGN KEY (BikeID) REFERENCES Bikes(BikeID) ON DELETE CASCADE,
  CONSTRAINT chk_time CHECK (EndTime > StartTime)
 -- Create Payments Table
CREATE TABLE Payments (
  PaymentID INT AUTO_INCREMENT PRIMARY KEY,
  BookingID INT NOT NULL,
  PaymentDate DATETIME DEFAULT CURRENT_TIMESTAMP,
  Amount DECIMAL(10, 2) NOT NULL,
  PaymentMethod ENUM('Credit Card', 'Debit Card', 'PayPal', 'Cash') NOT NULL,
  TransactionID VARCHAR(100) UNIQUE,
  Status ENUM('Pending', 'Completed', 'Failed', 'Refunded') DEFAULT 'Pending',
  FOREIGN KEY (BookingID) REFERENCES Bookings (BookingID) ON DELETE CASCADE
```

New Section 5 Page 1

-- Insert Data into Users Table

INSERT INTO Users (Name, Email, Phone, Address, RegistrationDate, Password) VALUES ('John Smith', 'john.smith@email.com', '123-456-7890', '123 Main St, City, Country', CURDATE(), 'nassword123')

('Emma Johnson', 'emma.johnson@email.com', '234-567-8901', '456 Elm St, City, Country', CURDATE(). 'securepass456').

('Michael Brown', 'michael.brown@email.com', '345-678-9012', '789 Oak St, City, Country', CURDATE(). 'brownie789').

('Sophia Davis', 'sophia.davis@email.com', '456-789-0123', '101 Pine St, City, Country', CURDATE(), 'sophia2023'),

('William Wilson', 'william.wilson@email.com', '567-890-1234', '202 Maple St, City, Country', CURDATE(), 'willpower555'),

('Olivia Taylor', 'olivia taylor@email.com', '678-901-2345', '303 Cedar St, City, Country', CURDATE(), 'oliveoil777'),

('James Anderson', 'james.anderson@email.com', '789-012-3456', '404 Birch St, City, Country', CURDATE(), 'jamesrules888'),

('Ava Martinez', 'ava.martinez@email.com', '890-123-4567', '505 Walnut St, City, Country', CURDATE(), 'avamartinez999'),

('Benjamin Lee', 'benjamin.lee@email.com', '901-234-5678', '606 Spruce St, City, Country', CURDATE(). 'benjilee101').

('Mia Rodriguez', 'mia.rodriguez@email.com', '012-345-6789', '707 Fir St, City, Country', CURDATE(), 'miarodriguez202');

#### -- Insert Data into Bikes Table

INSERT INTO Bikes (Model, Type, HourlyRate, Status, Location, YearManufactured) VALUES ('Trek FX 2', 'City Bike', 12.50, 'Available', 'North Station', 2021),

('Specialized Rockhopper', 'Mountain Bike', 15.00, 'Available', 'East Station', 2020),

('Cannondale Synapse', 'Road Bike', 18.00, 'Available', 'South Station', 2022),

('Giant Talon', 'Mountain Bike', 14.50, 'Available', 'West Station', 2021),

('Schwinn Cruiser', 'Cruiser Bike', 10.00, 'Available', 'Central Park', 2019),

('Rad Power RadRunner', 'Electric Bike', 25.00, 'Available', 'Downtown Hub', 2023),

('Brompton M6L', 'Folding Bike', 20.00, 'Available', 'Train Station', 2022),

('Surly Long Haul Trucker', 'Touring Bike', 16.00, 'Available', 'Bike Shop', 2020),

('Kona Dew', 'Hybrid Bike', 13.00, 'Available', 'University', 2021),

('Norco Bigfoot', 'Fat Bike', 22.00, 'Maintenance', 'Mountain Resort', 2022);

#### -- Insert Data into Bookings Table

INSERT INTO Bookings (UserID, BikeID, StartTime, EndTime, Status, TotalAmount, Notes) VALUES

 $(1,3,'2025\text{-}03\text{-}10\ 10\text{:}00\text{:}00','2025\text{-}03\text{-}10\ 14\text{:}00\text{:}00','\text{Completed'},72.00,'\text{Weekend ride'}),$ 

(2, 1, '2025-03-11 09:00:00', '2025-03-11 12:00:00', 'Completed', 37.50, 'City tour'),

(3, 6, '2025-03-12 14:00:00', '2025-03-12 17:00:00', 'Completed', 75.00, 'Electric bike trial'),

(4, 2, '2025-03-13 08:00:00', '2025-03-13 12:00:00', 'Completed', 60.00, 'Mountain trail ride'),

 $(5, 4, '2025-03-14\ 11:00:00', '2025-03-14\ 15:00:00', 'Active', 58.00, 'Group\ ride'),\\$ 

 $(6, 7, '2025-03-15\ 13:00:00', '2025-03-15\ 16:00:00', 'Pending', 60.00, 'Try\ folding\ bike'),$ 

(7, 8, '2025-03-16 07:00:00', '2025-03-16 19:00:00', 'Pending', 192.00, 'Full day tour'),

(8, 5, '2025-03-17 10:00:00', '2025-03-17 12:00:00', 'Pending', 20.00, 'Short ride'),

(9, 9, '2025-03-18 15:00:00', '2025-03-18 18:00:00', 'Cancelled', 39.00, 'Changed plans'),

 $(10, 1, '2025-03-19\ 09:00:00', '2025-03-19\ 13:00:00', 'Pending', 50.00, 'Business\ commute');$ 

#### -- Insert Data into Payments Table

 $INSERT\ INTO\ Payments\ (BookingID,\ PaymentDate,\ Amount,\ PaymentMethod,\ TransactionID,\ Status)$  VALUES

(1, '2025-03-10 09:30:00', 72.00, 'Credit Card', 'TXN123456789', 'Completed'),

(2, '2025-03-11 08:45:00', 37.50, 'PayPal', 'PP987654321', 'Completed'),

(3, '2025-03-12 13:50:00', 75.00, 'Debit Card', 'DC456789123', 'Completed'),

(4, '2025-03-13 07:55:00', 60.00, 'Credit Card', 'TXN789123456', 'Completed'),

(5, '2025-03-14 10:45:00', 58.00, 'PayPal', 'PP456123789', 'Completed'),

(6, '2025-03-15 12:30:00', 60.00, 'Credit Card', 'TXN321654987', 'Pending'),

(7, '2025-03-16 06:45:00', 192.00, 'Debit Card', 'DC159753468', 'Pending'),

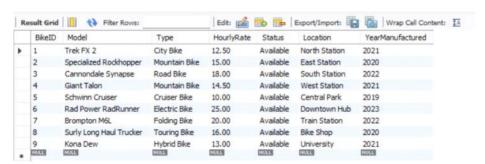
(8, '2025-03-17 09:50:00', 20.00, 'Cash', 'CASH001002', 'Pending'),

(9, '2025-03-18 14:55:00', 39.00, 'Credit Card', 'TXN753159852', 'Refunded'),

(10, '2025-03-19 08:40:00', 50.00, 'PayPal', 'PP258369147', 'Pending');

### -- 1. Select all available bikes

SELECT \* FROM Bikes WHERE Status = 'Available':



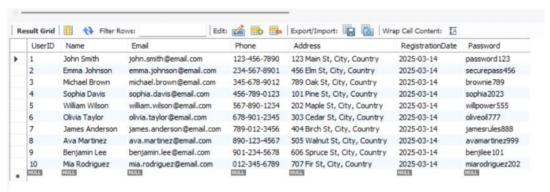
## -- 2. Select all active bookings

SELECT \* FROM Bookings WHERE Status = 'Active';



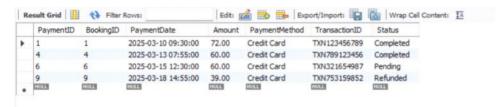
-- 3. Select all users who registered in 2025

SELECT \* FROM Users WHERE YEAR(RegistrationDate) = 2025;



-- 4. Select all payments made via Credit Card

SELECT \* FROM Payments WHERE PaymentMethod = 'Credit Card';



-- 1. Update bike status when rented

UPDATE Bikes SET Status = 'Rented' WHERE BikeID = 3;

- -- DELETE Operations
- -- 1. Delete a cancelled booking

DELETE FROM Bookings WHERE Status = 'Cancelled' AND BookingID = 9;

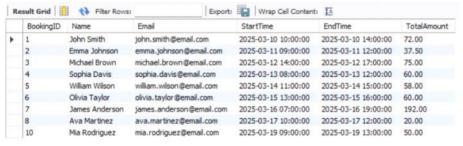
-- 2. Delete a user account

DELETE FROM Users WHERE UserID = 10;

-- 1. INNER JOIN: Get all bookings with user details

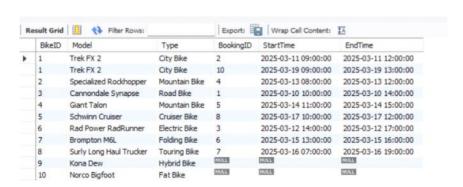
 ${\tt SELECT~b.BookingID,~u.Name,~u.Email,~b.StartTime,~b.EndTime,~b.TotalAmount~FROM~Bookings~b}$ 

INNER JOIN Users u ON b.UserID = u.UserID;



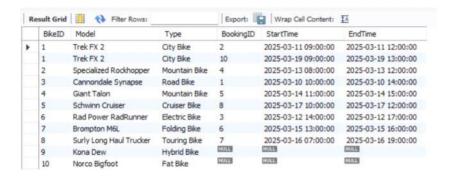
-- 2. LEFT JOIN: Get all bikes and their booking details (including bikes with no bookings) SELECT bk.BikeID, bk.Model, bk.Type, b.BookingID, b.StartTime, b.EndTime

FROM Bikes bk LEFT JOIN Bookings b ON bk.BikeID = b.BikeID;



-- 3. RIGHT JOIN: Get all payments and their corresponding bookings SELECT p.PaymentID, p.Amount, p.PaymentMethod, p.Status, b.BookingID, b.StartTime, b.EndTime FROM Bookings b

RIGHT JOIN Payments p ON b.BookingID = p.BookingID;



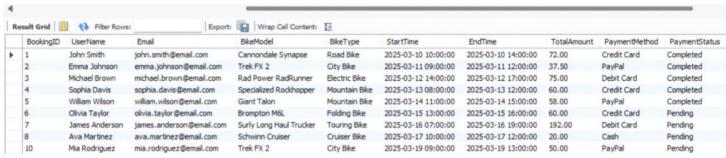
-- 4. Multiple JOINs: Get complete booking information with user, bike, and payment details SELECT b.BookingID, u.Name AS UserName, u.Email, bk.Model AS BikeModel, bk.Type AS BikeType,

 $b. StartTime, \, b. EndTime, \, b. Total Amount, \, p. PaymentMethod, \, p. Status \, AS \, PaymentStatus \, FROM \, Bookings \, b$ 

JOIN Users u ON b.UserID = u.UserID

JOIN Bikes bk ON b.BikeID = bk.BikeID

JOIN Payments p ON b.BookingID = p.BookingID;



-- 1. Find users who have spent more than the average amount on bookings SELECT u.UserID, u.Name, u.Email FROM Users u WHERE u.UserID IN ( SELECT b.UserID FROM Bookings b GROUP BY b.UserID HAVING SUM(b.TotalAmount) > ( SELECT AVG(TotalAmount) FROM Bookings Edit: 🕍 📆 Export/Import: 🏣 👸 | Wrap Cell Content: 🔣 Hearth Name John Smith john.smith@email.com 3 Michael Brown michael.brown@email.com James Anderson iames.anderson@email.com HULL DUCTOR -- 2. Find the most expensive bike type based on average hourly rate SELECT Type, AVG(HourlyRate) as AvgRate **GROUP BY Type** HAVING AvgRate >= ALL ( SELECT AVG(HourlyRate) FROM Bikes **GROUP BY Type** Result Grid Filter Rows: Export: Wrap Cell Content: IA Type AvgRate 25.000000 Electric Bike -- 3. Find bikes that have never been booked SELECT BikeID, Model, Type FROM Bikes WHERE BikeID NOT IN ( SELECT DISTINCT BikeID FROM Bookings



# **ER Diagram**

