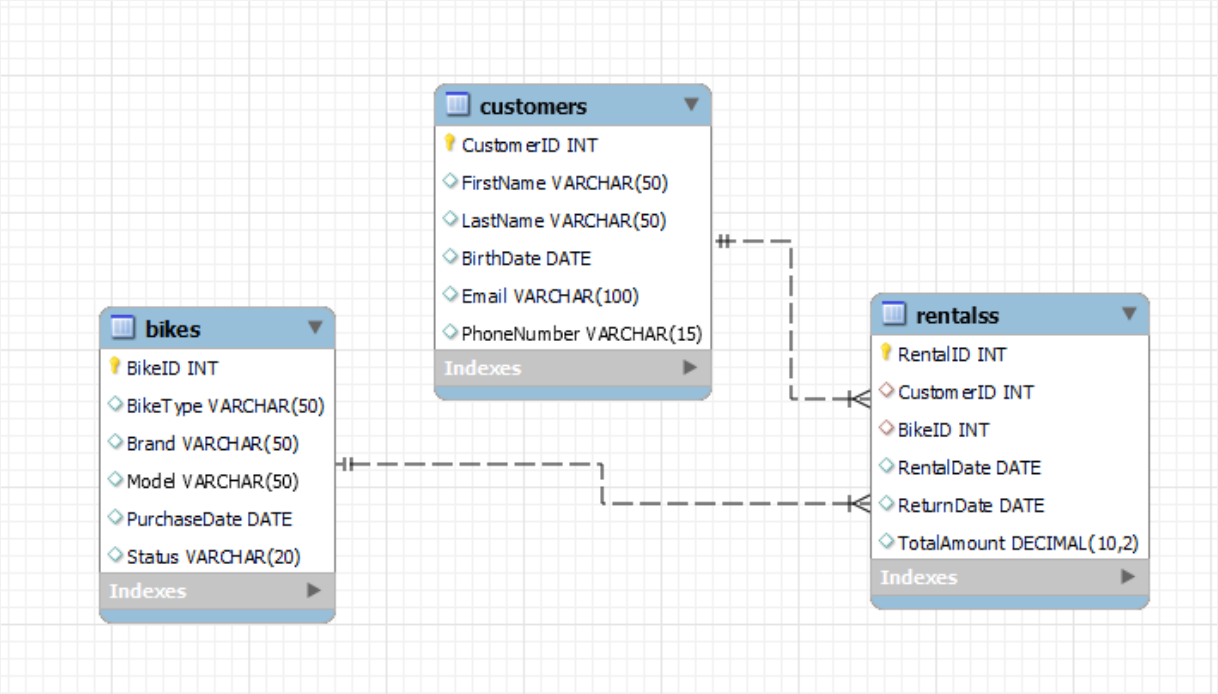
CASE STUDY OF BIKE RENTALS

Introduction

In this case study, we will develop a comprehensive Bike Rental Management System using SQL databases. The system is divided into three key components, each represented by a separate database: Customer Management, Bike Inventory, and Rental Records. Each component focuses on managing different aspects of the bike rental process, ensuring efficient handling of customer information, bike details, and rental transactions.

ENTITY RELATIONSHIP DIAGRAM



THEY ARE THREE DATABASE USED IN CASE STUDY

1)BIKES

2)CUSTOMERS

3)RENTALS

DATASET

1) CREATE DATABASE CustomerManagement;

USE CustomerManagement;

SELECT * FROM Customers;

2) CREATE DATABASE BikeInventory;

USE BikeInventory;

SELECT * FROM Bikes;

3) CREATE DATABASE RentalRecords;

USE RentalRecords;

SELECT * FROM Rentals;

```sql
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    BirthDate DATE,
    Email VARCHAR(100),
    PhoneNumber VARCHAR(15)
);

-- Insert records into the Customers table
INSERT INTO Customers (CustomerID, FirstName, LastName, BirthDate, Email, PhoneNumber)
VALUES
    (1, 'John', 'Doe', '1985-05-15', 'john.doe@example.com', '555-1234'),
    (2, 'Jane', 'Smith', '1990-08-22', 'jane.smith@example.com', '555-5678'),
    (3, 'Alice', 'Johnson', '1982-12-30', 'alice.johnson@example.com', '555-8765'),
    (4, 'Bob', 'Brown', '1995-03-10', 'bob.brown@example.com', '555-4321'),
    (5, 'Carol', 'White', '1988-07-18', 'carol.white@example.com', '555-6789');



-- Create the Bikes table
CREATE TABLE Bikes (
    BikeID INT PRIMARY KEY,
    BikeType VARCHAR(50),
    Brand VARCHAR(50),
    Model VARCHAR(50),
    PurchaseDate DATE,
    Status VARCHAR(20)
);
```

```sql
INSERT INTO Bikes (BikeID, BikeType, Brand, Model, PurchaseDate, Status)
VALUES

  (1, 'Mountain', 'Giant', 'Talon 3', '2022-05-01', 'Available'),
    (2, 'Road', 'Trek', 'Domane AL 2', '2022-06-15', 'Available'),
    (3, 'Hybrid', 'Cannondale', 'Quick 4', '2022-07-20', 'Available'),
    (4, 'Electric', 'Specialized', 'Turbo Vado SL', '2022-08-10', 'Available'),
    (5, 'BMX', 'Mongoose', 'Legion L100', '2022-09-05', 'Available');


-- Create the Rentals table
CREATE TABLE Rentalss (
    RentalID INT PRIMARY KEY,
    CustomerID INT,
    BikeID INT,
    RentalDate DATE,
    ReturnDate DATE,
    total DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES CustomerManagement.Customers(CustomerID),
    FOREIGN KEY (BikeID) REFERENCES BikeInventory.Bikes(BikeID)
);

INSERT INTO Rentalss (RentalID, CustomerID, BikeID, RentalDate, ReturnDate, TotalAmount)
VALUES
    (1, 1, 1, '2023-07-01', '2023-07-03', 30.00),
    (2, 2, 2, '2023-07-02', '2023-07-04', 25.00),
    (3, 3, 3, '2023-07-03', '2023-07-05', 27.50),
    (4, 4, 4, '2023-07-04', '2023-07-06', 35.00),
    (5, 5, 5, '2023-07-05', '2023-07-07', 22.50);
```

CASE STUDY QUESTION WITH ANSWER

1)LIST ALL CUSTOMERS

```
SELECT * FROM Customers;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CustomerID	FirstName	LastName	BirthDate	Email	PhoneNumber
1	John	Doe	1985-05-15	john.doe@example.com	555-1234
2	Jane	Smith	1990-08-22	jane.smith@example.com	555-5678
3	Alice	Johnson	1982-12-30	alice.johnson@example.com	555-8765
4	Bob	Brown	1995-03-10	bob.brown@example.com	555-4321
5	Carol	White	1988-07-18	carol.white@example.com	555-6789

Customers 1

2)FIND CUSTOMERS BORN AFTER JANUARY 1990

```
SELECT * FROM Customers WHERE BirthDate > '1990-01-01';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TotalCustomers
5

3)COUNT THE TOTAL NUMBER OF CUSTOMERS

```
SELECT COUNT(*) AS TotalCustomers FROM Customers;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CustomerID	FirstName	LastName	BirthDate	Email	PhoneNumber
2	Jane	Smith	1990-08-22	jane.smith@example.com	555-5678
4	Bob	Brown	1995-03-10	bob.brown@example.com	555-4321
NULL	NULL	NULL	NULL	NULL	NULL

Customers 2

-- 4. Find customers with a specific email domain (e.g., @example.com).

```
SELECT * FROM Customers WHERE Email LIKE '%@example.com'
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CustomerID	FirstName	LastName	BirthDate	Email	PhoneNumber
1	John	Doe	1985-05-15	john.doe@example.com	555-1234
2	Jane	Smith	1990-08-22	jane.smith@example.com	555-5678
3	Alice	Johnson	1982-12-30	alice.johnson@example.com	555-8765
4	Bob	Brown	1995-03-10	bob.brown@example.com	555-4321
5	Carol	White	1988-07-18	carol.white@example.com	555-6789

Customers 6

---5. List customers by their birth month.

```
SELECT MONTH(BirthDate) AS BirthMonth, FirstName, LastName FROM Customers ORDER BY BirthMonth;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BirthMonth	FirstName	LastName
3	Bob	Brown
5	John	Doe
7	Carol	White
8	Jane	Smith
12	Alice	Johnson

-- 6. List all rentals along with customer names.

```
SELECT r.RentalID, r.RentalDate, r.ReturnDate, c.FirstName, c.LastName
FROM Rentalss r
JOIN CustomerManagement.Customers c ON r.CustomerID = c.CustomerID;
```

Result Grid | Filter Rows: | Export:

RentalID	RentalDate	ReturnDate	FirstName	LastName
1	2023-07-01	2023-07-03	John	Doe
2	2023-07-02	2023-07-04	Jane	Smith
3	2023-07-03	2023-07-05	Alice	Johnson
4	2023-07-04	2023-07-06	Bob	Brown
5	2023-07-05	2023-07-07	Carol	White

Result 1

-- 7. Find all rentals for 'Mountain' bikes.

```
SELECT r.RentalID, r.RentalDate, r.ReturnDate, b.BikeType
FROM Rentals r
JOIN BikeInventory.Bikes b ON r.BikeID = b.BikeID
WHERE b.BikeType = 'Mountain';
```

Result Grid | Filter Rows: | Export

RentalID	RentalDate	ReturnDate	BikeType
1	2023-07-01	2023-07-03	Mountain

-- 8. Find all customers who have rented 'Electric' bikes.

```
SELECT DISTINCT c.CustomerID, c.FirstName, c.LastName
FROM Rentals r
JOIN CustomerManagement.Customers c ON r.CustomerID = c.CustomerID
JOIN BikeInventory.Bikes b ON r.BikeID = b.BikeID
WHERE b.BikeType = 'Electric';
```

Result Grid | Filter Rows:

CustomerID	FirstName	LastName
4	Bob	Brown

-- 20. List all bikes that have been rented out more than once.

```
SELECT b.BikeID, b.BikeType, COUNT(r.RentalID) AS TotalRentals
FROM Rentals r
JOIN BikeInventory.Bikes b ON r.BikeID = b.BikeID
GROUP BY b.BikeID, b.BikeType
HAVING COUNT(r.RentalID) > 1;
```

Result Grid | Filter Rows:

BikeID	BikeType	TotalRentals

-- 10. List the total amount earned from rentals for each bike type.

```
SELECT b.BikeType, SUM(r.TotalAmount) AS TotalEarnings
FROM Rentals r
JOIN BikeInventory.Bikes b ON r.BikeID = b.BikeID
GROUP BY b.BikeType;
```

Result Grid | Filter Rows:

BikeType	TotalEarnings
Mountain	30.00
Road	25.00
Hybrid	27.50
Electric	35.00
BMX	22.50

-- 12. Find all rentals by 'John Doe'.

```
SELECT * FROM Rentals WHERE CustomerID = (SELECT CustomerID FROM
CustomerManagement.Customers WHERE FirstName = 'John' AND LastName = 'Doe');
```

Result Grid | Filter Rows: | Edit: | Exp

RentalID	CustomerID	BikeID	RentalDate	ReturnDate	TotalAmount
1	1	1	2023-07-01	2023-07-03	30.00
NULL	NULL	NULL	NULL	NULL	NULL

-- 13. Find the average rental duration.

```
SELECT AVG(DATEDIFF(ReturnDate, RentalDate)) AS AverageRentalDuration FROM Rentals;
```

Result Grid | Filter Rows:

AverageRentalDuration
2.0000

-- 14. Find the bike with the earliest purchase date.

```
SELECT * FROM Bikes ORDER BY PurchaseDate ASC LIMIT 1;
```

Result Grid | Filter Rows: | Edit:

BikeID	BikeType	Brand	Model	PurchaseDate	Status
1	Mountain	Giant	Talon 3	2022-05-01	Available
NULL	NULL	NULL	NULL	NULL	NULL

THANKING YOU

PRESENTED BY

RAGUL S