

Design and Implementation of a Relational Database System

ACCA4023

MARVIN ALEXANDER VALENCIA ALFARO

Table of Contents

INTRODUCTION	3
1. Scenario Overview and Requirements	4
2. ERD (Entity Relationship Diagram)	5
3. Relationship table	6
4. Table Design	7
4.1 Datasheet View	9
4.2 Maintenance Records	10
5. Queries and testing	11
Query Q1 – Currently Rented Bikes	11
Query Q2 – Available Bikes per Station	12
Query Q3 – Rentals per Rider	13
Query Q5 – Maintenance History by Bike	14
Conclusion	16
References	17

INTRODUCTION

The purpose of this report is to implement and design a database for the EcoRide electric bike rental scheme. This company aims to provide a service that allows people in the city to get around more quickly and efficiently, helping the environment and promoting physical activity. We are trying to determine how the database can help us obtain information about the users who use our service and get an idea of how many bicycles are available at each station, both to provide quality service to our users and to keep track of the necessary maintenance of the bicycles. Therefore, the main objective is to design and implement a database relationship in the sector. This report covers scenario analysis, ERD, implementation in Access, queries, and testing.

The objective of this database is to replace the current manual processes consisting of paper forms and spreadsheets with a structured system capable of storing rider details, tracking rentals, maintaining bike status, and supporting maintenance operations. This report covers the scenario analysis, Entity Relationship Diagram (ERD), database implementation in Microsoft Access, sample data population, query development, and system testing. [1]

1. Scenario Overview and Requirements

EcoRide is a green transport initiative that allows users to rent electric bikes from different stations located at parks, train stations and universities campuses. At present, rider details, rental transaction and maintenance information are captured manually using paper form and simple spreadsheets. This leads to data inconsistency, missing information and limited visibility of bike status across the network.

From the scenario description, several key requirements were identified. The system must store detail of riders and staff, register each bike and its current location, and record every rental including the start and end station, dates and cost. In addition, the database must support logging maintenance activities, including the bike involved, the issue reported and the staff member responsible. Finally, the system should allow staff to run queries to find available bikes, monitoring usage and identify bikes under repair or overdue rentals.

2. ERD (Entity Relationship Diagram)

The Entity Relationship Diagram (ERD) in figure 1 illustrates the final logical design of EcoRide database. It shows six entities with their primary keys, key attributes and the foreign key relationships between them. The diagram includes a double relationship from station to rental, representing the stationOUT and StationRETURN fields, which allow bikes to be picked up and dropped off at different locations. The Bike Entity also contains a foreign key called “CurrentLocation”, linking to station, so that the system can track where each bike is stored when it is not on hire. Overall, the ERD provides a clear visual representation of how data is structured and interconnected in the system [2].

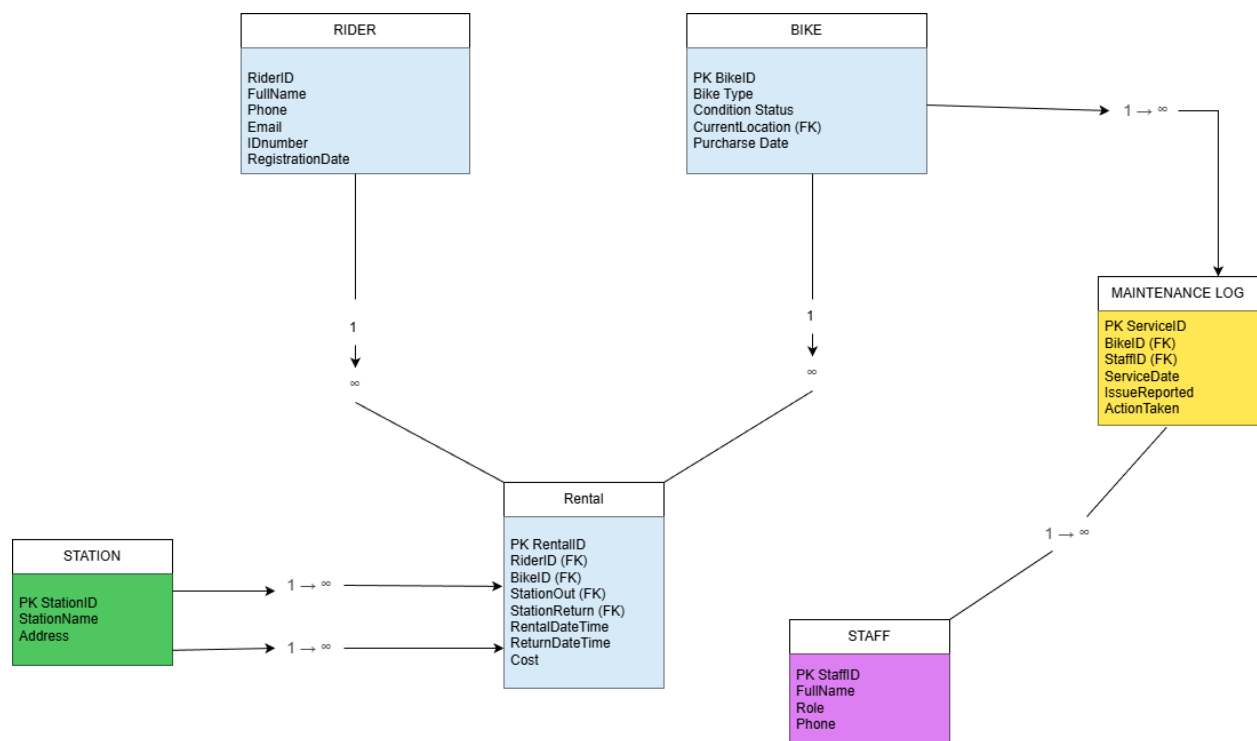


Figure 2.1 – ERD for the EcoRide database system.

3. Relationship table

Referential integrity was enforced for all primary and foreign keys using the Access Relationships view. One-to-many relationships were defined as follows:

- Rider → Rental
- Bike → Rental
- Station → Rental (both StationOUT and StationRETURN)
- Station → Bike (via CurrentLocation)
- Staff → MaintenanceLog
- Bike → MaintenanceLog

Enforcing referential integrity prevents invalid entries such as orphan rentals referencing bikes or stations that do not exist, contributing to consistent and reliable data management.

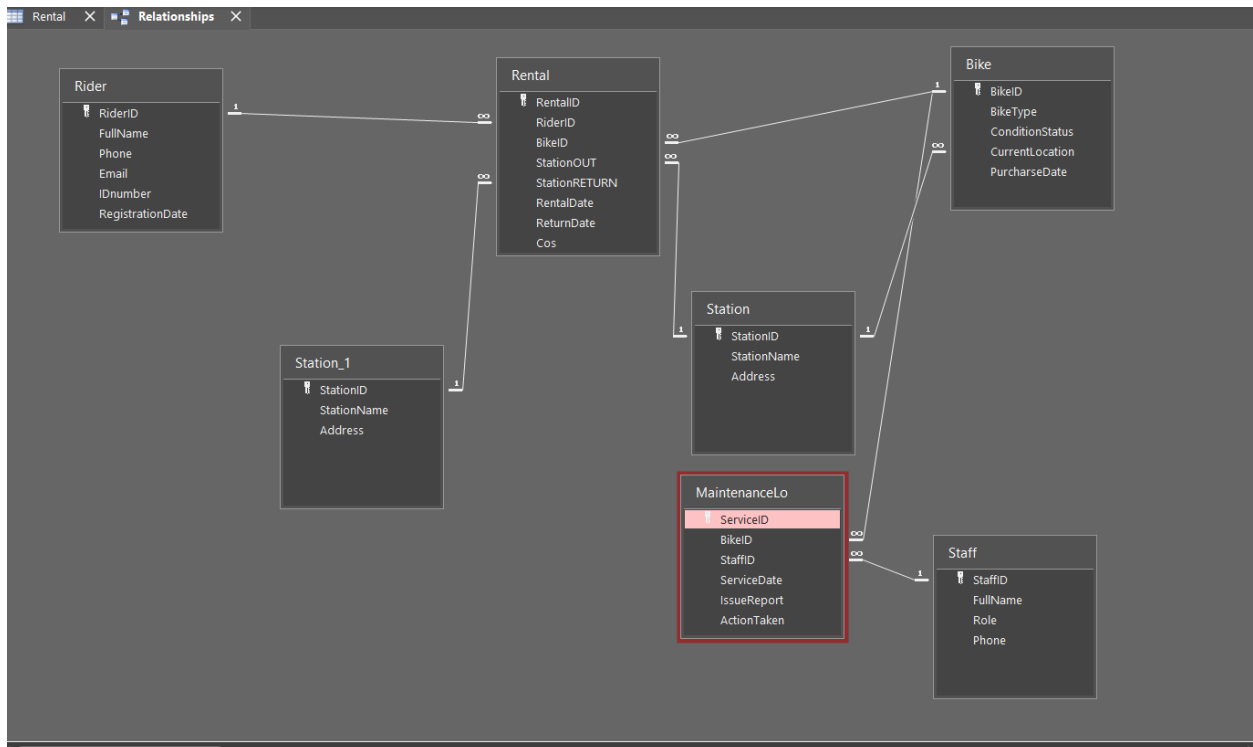


Figure 3.1 shows the implemented relationships in Access, enforcing referential integrity between the main entities.

4. Table Design

Figures X–Y show the design of the main entities in Design View and Datasheet View. Field types, primary keys, and foreign keys were configured according to the ERD to ensure consistency across the system.

This figure shows the configuration of the Rider and Bike tables, including key attributes and field types.

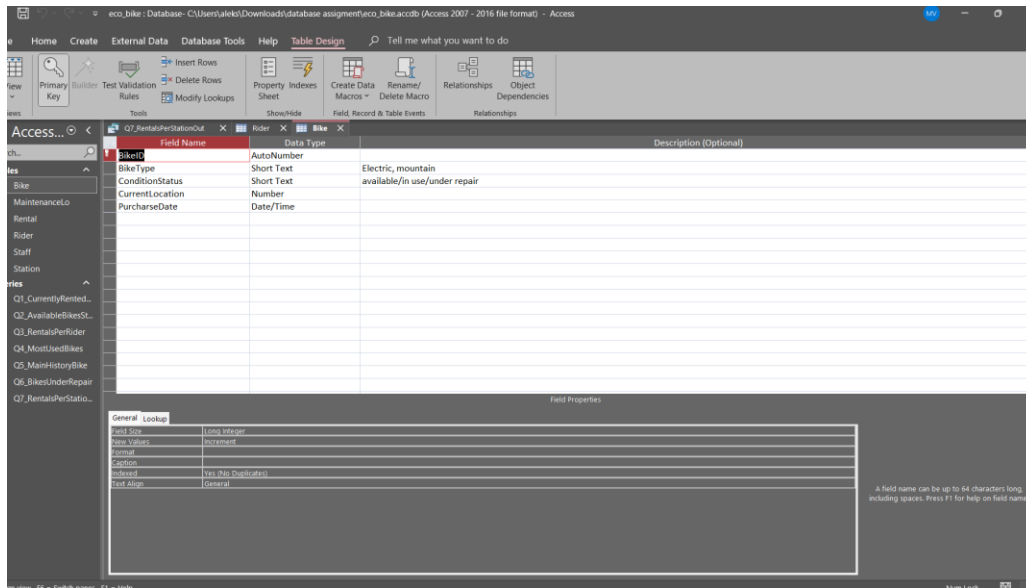


Figure 4.1 – Bike

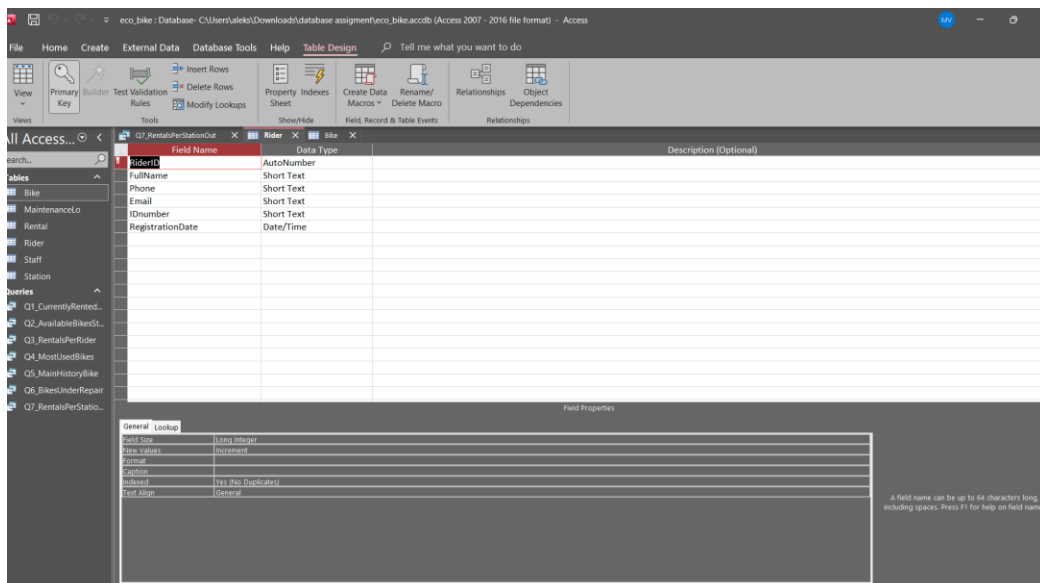


Figure 4.2 - Rider

Rental and Station (Design View)

This section illustrates the Rental and Station table structures, highlighting the foreign keys linking to Rider, Bike, and Station.

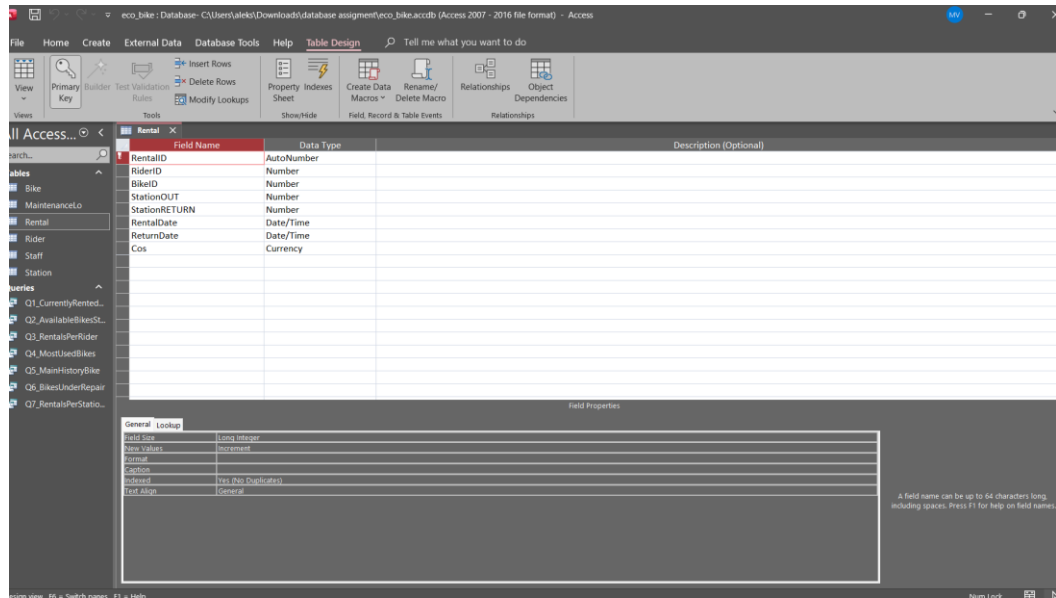


Figure 4.3 - Rental

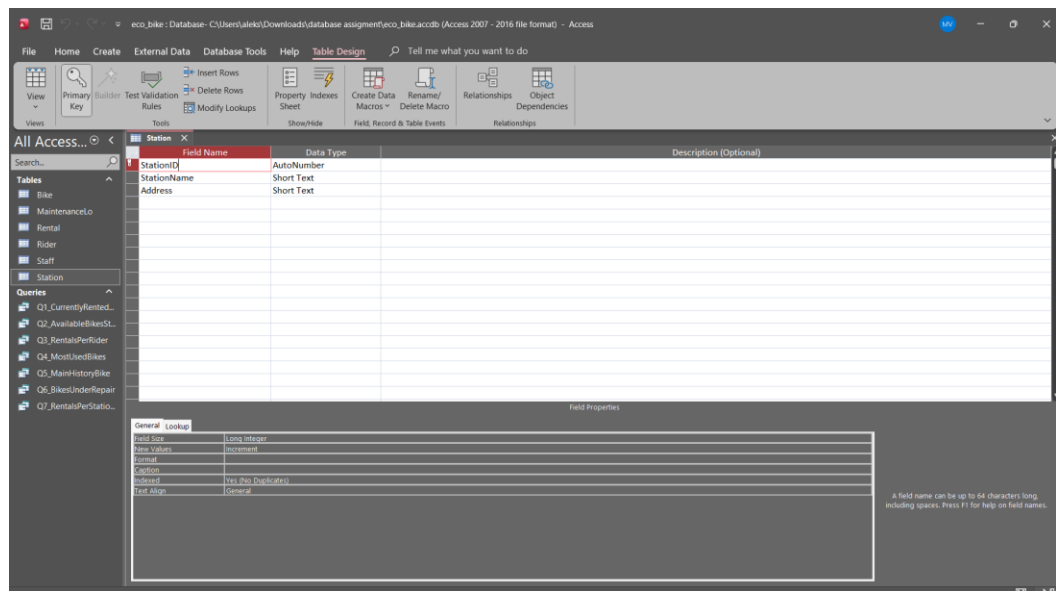


Figure 4.4 - Station

4.1 Datasheet View

This figure displays sample data inserted into the Rider, Bike, and Station tables for system testing and validation.

The figure consists of three screenshots of the Microsoft Access application, each showing a different table in the 'All Access' database. The first screenshot shows the 'Station' table with three records. The second screenshot shows the 'Rider' table with five records. The third screenshot shows the 'Bike' table with five records. Each screenshot includes the standard Access ribbon and a left-hand pane showing the database structure.

Station Table Data:

StationID	StationName	Address
1	Central Park	Park Lane, London
2	University Hub	Campos Road, London
3	Riverside Point	River Street, London

Rider Table Data:

RiderID	FirstName	Phone	Email	RiderNumber	RegistrationDate
1	Daniel Rivera	849112312312	daniel.r@gmail.com	RID001	3/10/2024
2	Sarah Thompson	002024319485	sarah.t@gmail.com	RID002	4/18/2024
3	Michael Johnson	02015484555	michael.j@gmail.com	RID003	2-5-2024
4	Laura White	020164893333	laura.w@gmail.com	RID004	12/6/2024
5	Sam Martinez	020154645454	samuel.m@gmail.com	RID005	1/7/2024

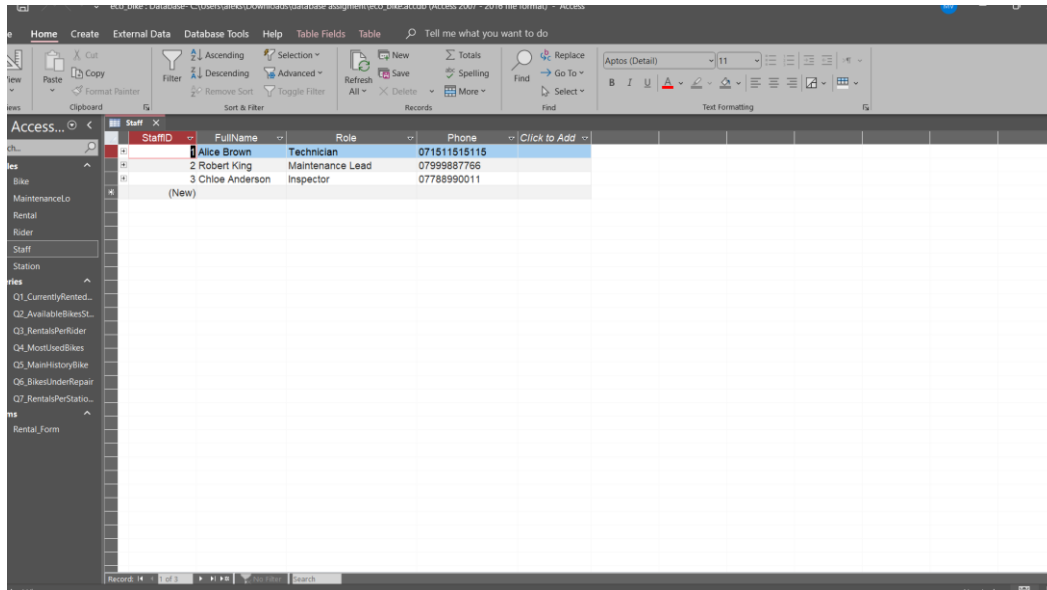
Bike Table Data:

BikeID	BikeType	ConditionStatus	CurrentLocation	PurchaseDate
1	Electric City	Available	1	2/15/2023
2	Electric City	In Use	2	3/21/2023
3	Mountain E-Bike	Available	3	10/5/2023
4	Electric Road	Under Repair	2	6/6/2023
5	Cargo E-Bike	Available	1	7/25/2023

Figure- Sample data for Rider, Bike and Station tables.

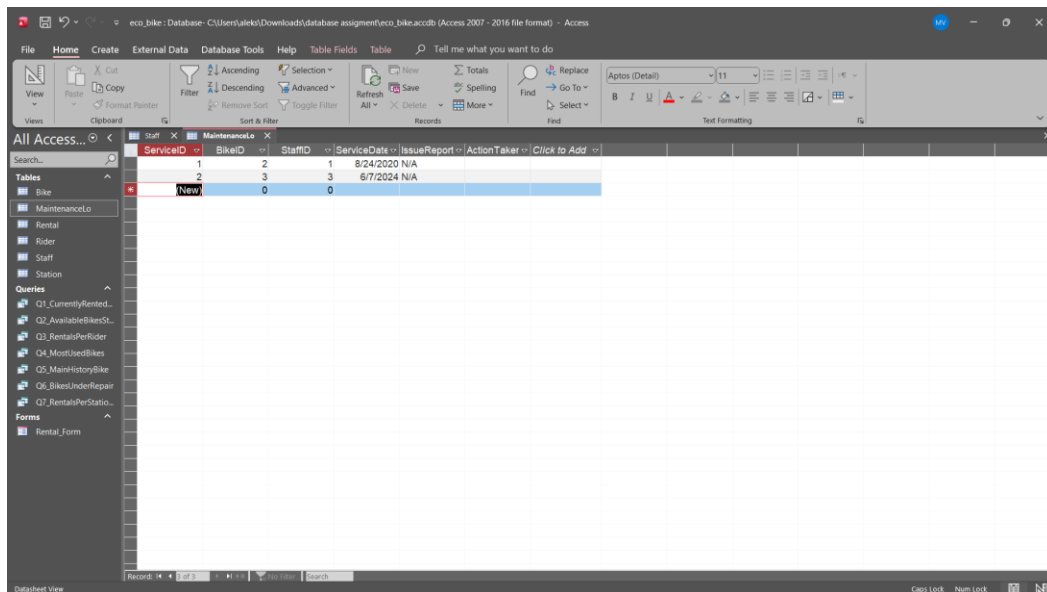
4.2 Maintenance Records

This figure presents the sample MaintenanceLog data used to validate the maintenance tracking functionality.



The screenshot shows the Microsoft Access interface with the 'Staff' table selected in the 'All Access Objects' pane. The table is displayed in Datasheet view. The table has five columns: StaffID, FullName, Role, Phone, and Click to Add. There are three records listed.

StaffID	FullName	Role	Phone	Click to Add
1	Alice Brown	Technician	07151151515	
2	Robert King	Maintenance Lead	07999887766	
3	Chloe Anderson	Inspector	07788990011	



The screenshot shows the Microsoft Access interface with the 'MaintenanceLog' table selected in the 'All Access Objects' pane. The table is displayed in Datasheet view. The table has six columns: ServiceID, BikeID, StaffID, ServiceDate, IssueReport, and ActionTaken. There are three records listed, with the first two being existing records and the third being a new record.

ServiceID	BikeID	StaffID	ServiceDate	IssueReport	ActionTaken
1	2	1	8/24/2020	N/A	
2	3	3	6/7/2024	N/A	
New	0	0			

Figure- Sample Rental and Maintenance records uses for testing

5. Queries and testing

The following queries demonstrate the practical use of the system and confirm that the data model supports the operations required by EcoRide.

Query Q1 – Currently Rented Bikes

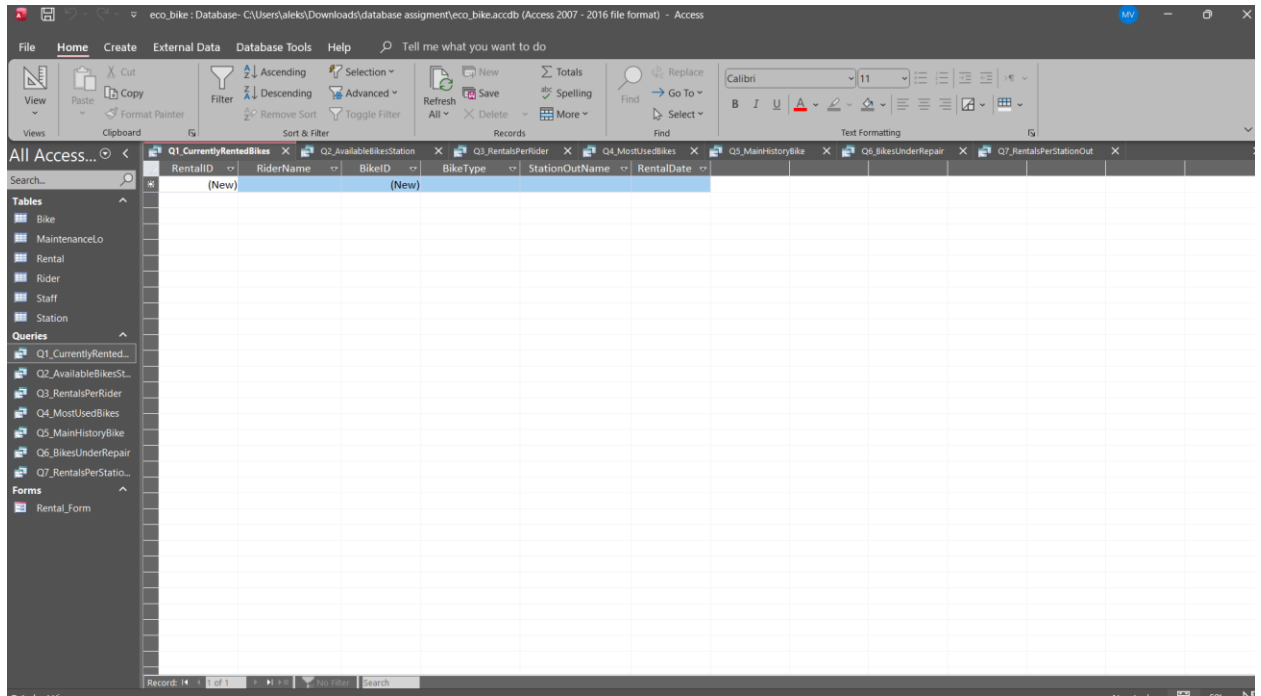


Figure 5.1 – Design view of query Q1_CurrentlyRentedBikes. This query returns all bikes that have an active rental. It supports monitoring of real time bike usage.

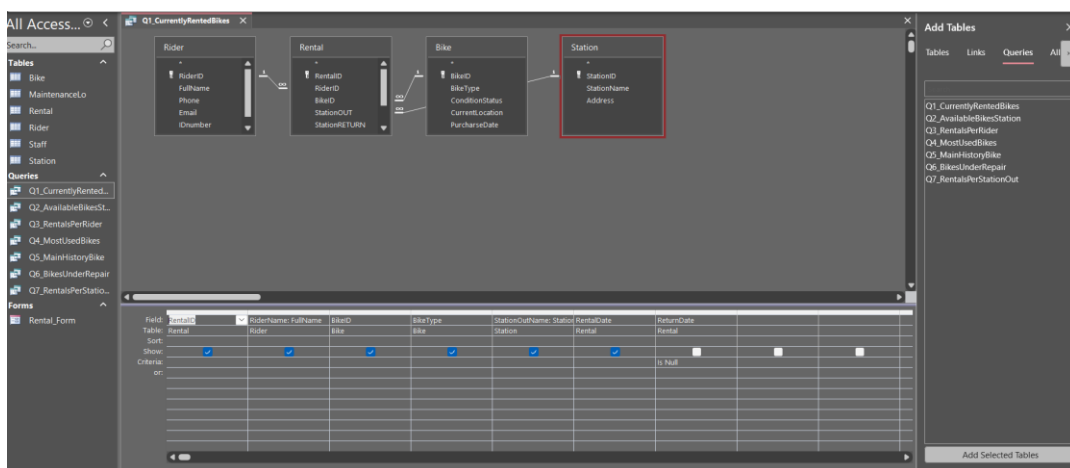


Figure 5.2 – Results of query Q1_CurrentlyRentedBikes showing bikes that are currently on hire. The results show which bikes are currently on hire, along with the associated rider details.

Query Q2 – Available Bikes per Station

This query retrieves all bikes marked as 'Available' and displays the station where each bike is located, supporting daily operational planning for station capacity.

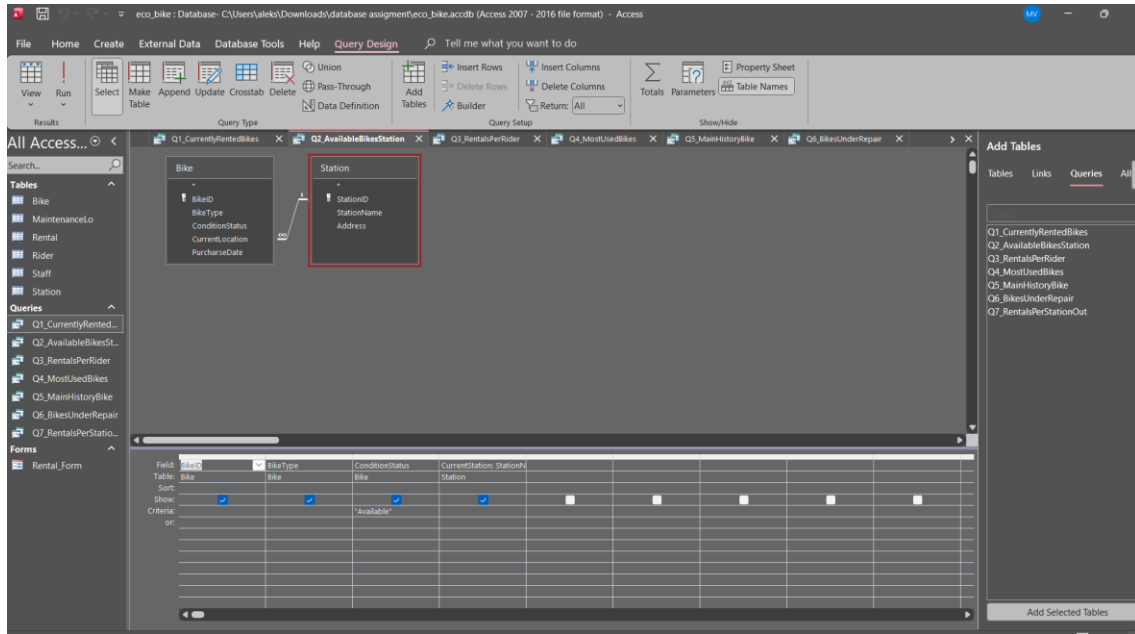


Figure 5.3 - Design view of query Q2_AvailableBikesStation which returns all bikes whose status is 'Available' together with the station where each bike is currently located.

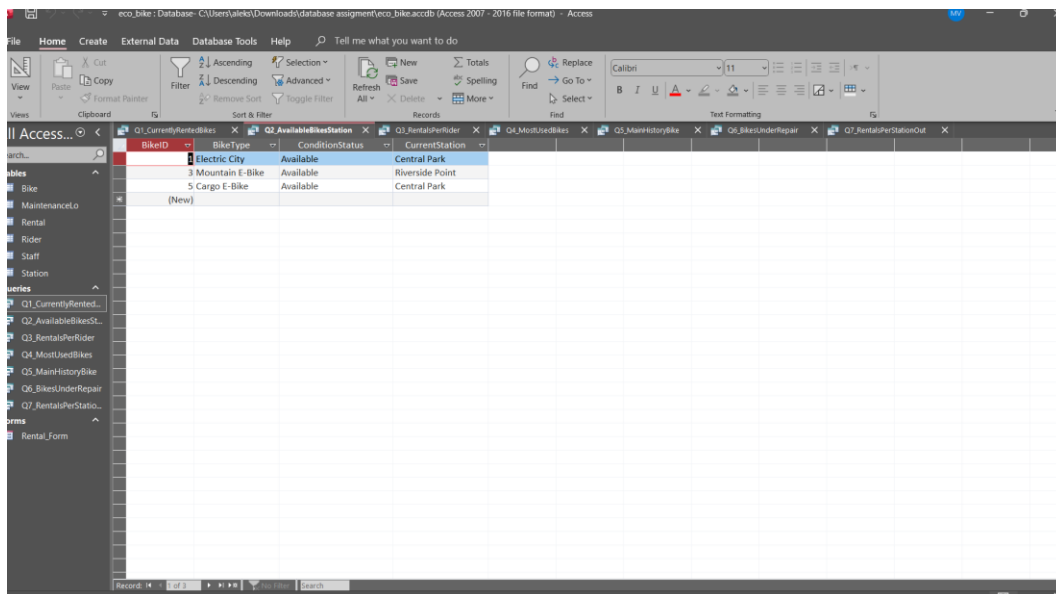


Figure 5.4 - Results of query Q2_AvailableBikesStation showing the available bikes at each station, supporting day to day operational planning.

This query aggregates rental records by rider to calculate the total number of rentals and the most recent rental date, enabling usage analysis.

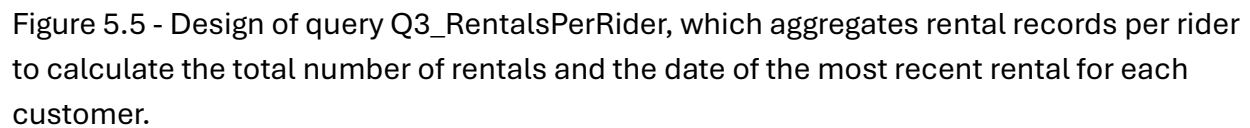


Figure 5.6 - Results of query Q3_RentalsPerRider highlighting the most active riders in the EcoRide system based on historical rental activity.

Query Q5 – Maintenance History by Bike

This query joins Bikes, MaintenanceLog, and Staff to produce a full service history for each bike, including issue reported, action taken, and staff involved. It supports maintenance planning and safety compliance.

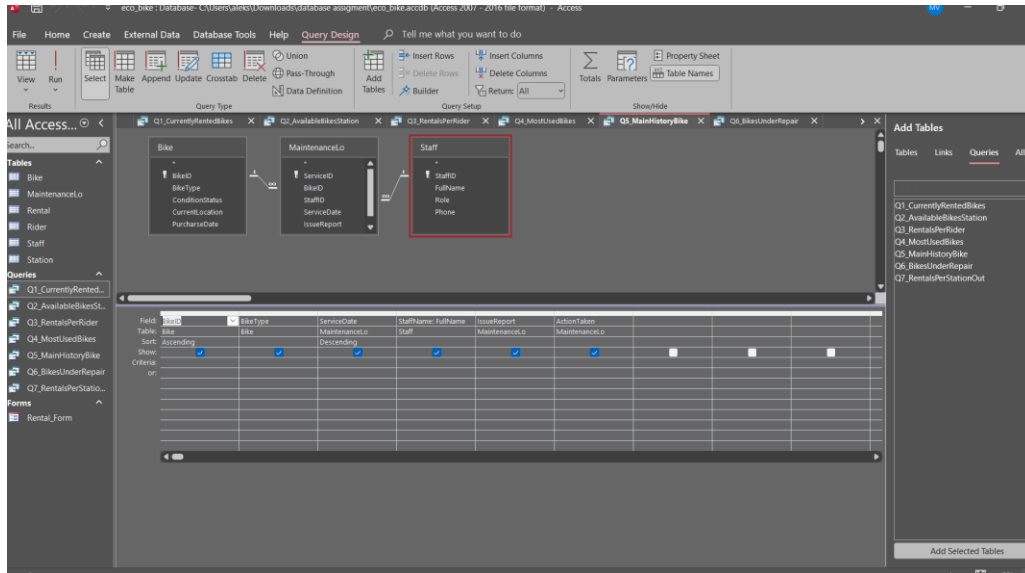


Figure 5.7 - Design of query Q5_MaintenanceHistoryByBike, which joins bikes, maintenance records and staff to produce a detailed maintenance history per bike.

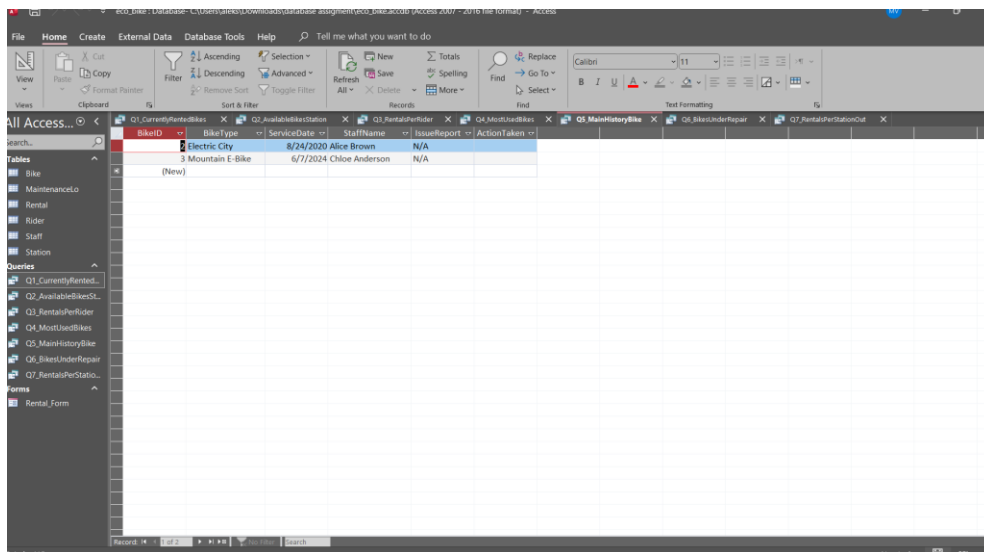


Figure 5.8 - Results of query Q5_MaintenanceHistoryByBike listing maintenance events for each bike, including the issue reported, the action taken and the staff member responsible.

Validation and Testing

The system was tested by populating each table with sample data and executing the previously designed queries. The results confirmed that:

- Referential integrity prevents invalid or inconsistent entries.
- Queries produce accurate summaries of bike availability, current rentals, rider activity, and maintenance history.
- The database structure meets Third Normal Form (3NF), avoiding redundancy and ensuring efficient data updates.
- The system fulfills all functional requirements of the EcoRide scenario.

Conclusion

The development of the EcoRide relational database system successfully meets the operational and analytical requirements identified in the scenario. By replacing manual paper-based processes with a structured and normalised database design, the organisation can now maintain accurate and consistent records of riders, bikes, rentals, stations, and maintenance activities.

The implemented data model supports reliable tracking of bike availability, rider activity, and maintenance history, enabling staff to make informed decisions and manage daily operations more efficiently. The use of referential integrity and Third Normal Form (3NF) principles ensures data accuracy, reduces redundancy, and improves overall system reliability.

Furthermore, the queries developed demonstrate the practical functionality of the database by providing meaningful insights into system usage, operational performance, and maintenance needs. Through testing with sample data, the system proved to be robust, scalable, and aligned with the intended purpose of supporting a sustainable and efficient electric bike rental service.

Overall, this database solution provides EcoRide with a strong foundation for data management and operational optimisation, enhancing both service delivery and long-term organisational performance.

References

- [1] R. E. a. S. Navathe, Fundamentals of Database Systems, Boston, Mass: Pearson Addison-Wesley, 2007.
- [2] T. C. a. C. Begg, Database Systems : a practical approach to design, implementation and management, Sixth, Global Edition: Boston: Pearson, 2015.