

LEVEL 4 ASSESSMENT



Module Name: *Introduction to Web and Database Concepts*



Module Code : *ACCA4023*



Student Name: *Thomas Pereira*



Student ID: *2442400*

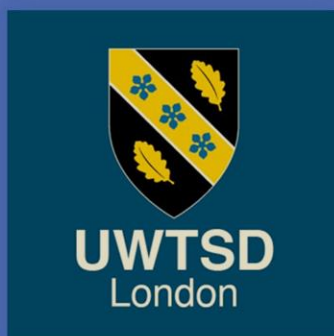


Group: *1*



Lecturer's Name: *Rajeev Nath*

London



Submission by : 25 November 2025

Access file link (one drive): [EcoRide Thomas 2442400](#)

Table of Contents

<i>Introduction.....</i>	<i>3</i>
<i>Task 1: Implementation Guide:</i>	<i>4</i>
<i> Scenario Analysis.....</i>	<i>4</i>
<i> Entity-Relationship Diagram (ERD).....</i>	<i>9</i>
<i> Data structure Development</i>	<i>9</i>
<i> Normalization Compliance.....</i>	<i>13</i>
<i> Sample Data Population.....</i>	<i>15</i>
<i> Query Development.....</i>	<i>16</i>
<i> Validation and Testing</i>	<i>26</i>
<i>Conclusion.....</i>	<i>27</i>
<i>References.....</i>	<i>29</i>



Introduction

This report details the design and implementation of a relational database system for EcoRide, an electric bike rental service run by the local authority. The project shows how to apply database design principles, normalization theory, and query development using Microsoft Access. EcoRide currently faces operational challenges because it relies on manual paper forms and basic spreadsheets. This leads to limited visibility of bike availability, maintenance status, and rental history. The existing system struggles with double-bookings, unreturned bikes, and generating accurate usage reports. A well-structured database solution is crucial to improve operations and service efficiency.

The main goals of this project is:

Design a normalized relational database structure based on Third Normal Form (3NF) principles [1]

Implement the database in Microsoft Access with suitable data types, constraints, and relationships. will provide screenshots.

Fill the system with realistic sample data that reflects actual operations.

Create practical queries to support business tasks and reporting needs

Test the system thoroughly to ensure data integrity and functional accuracy

This report follows a seven-step implementation guide: scenario analysis, ERD creation, data structure development, checking normalization, sample data population, query development, and validation testing. Each section shows a solid understanding of database design methods and the practical use of database management principles. The final solution allows EcoRide staff to manage rider registrations, bike inventory, rental transactions, and maintenance scheduling effectively through a centralized database system.

Task 1: Implementation Guide:

Scenario Analysis

2.1 Overview of EcoRide

EcoRide is a green transportation initiative that is offering electric bike rentals.

services at various public locations such as parks, train stations, and university campuses. The existing system is based on manual paper forms. and basic spreadsheets, yield operational inefficiencies such as bent double-bookings, [2] unreturned bikes, and limited visibility of bike availability and maintenance needs.

2.2 Identified Entities

Through analysis of the scenario, five main entities have been identified:

1. RIDERS

Purpose: Store the information of registered customers

Key Attributes: Rider_ID - primary key, First_Name, Last_Name, Email, Phone, ID_Number, Registration_Date

Justification: Needed to track who rents the bikes and prevent unauthorized use. The ID_Number field ensures each rider has unique Identification.

Field Name	Data Type	Description
First_Name	Short Text	Rider's first name
Last_Name	Short Text	Rider's last name
Email	Short Text	Contact email address
Phone	Short Text	Contact phone number
ID_Number	Short Text	Government ID or license
Registration_Date	Date/Time	Date rider registered

Fig. Rider_ID

2. BIKES

Purpose: To maintain inventories of all electric bikes

Key Attributes: Bike_ID (primary key), Bike_Type, Condition, Current_Location, Status, Purchase_Date, Notes

Justification: Central to the rental system; tracks bike availability, condition, and maintenance status. The Status field allows for real-time Tracking availability.

Field Name	Data Type	Description
Bike_ID	AutoNumber	Unique bike identifier
Bike_Type	Short Text	Type of e-bike
Condition	Short Text	Physical condition
Current_Location	Short Text	Where bike is now
Status	Short Text	Available/Rented/Maintenance
Purchase_Date	Date/Time	When bike was purchased
Notes	Long Text	Additional notes

Fig. Bikes_ID

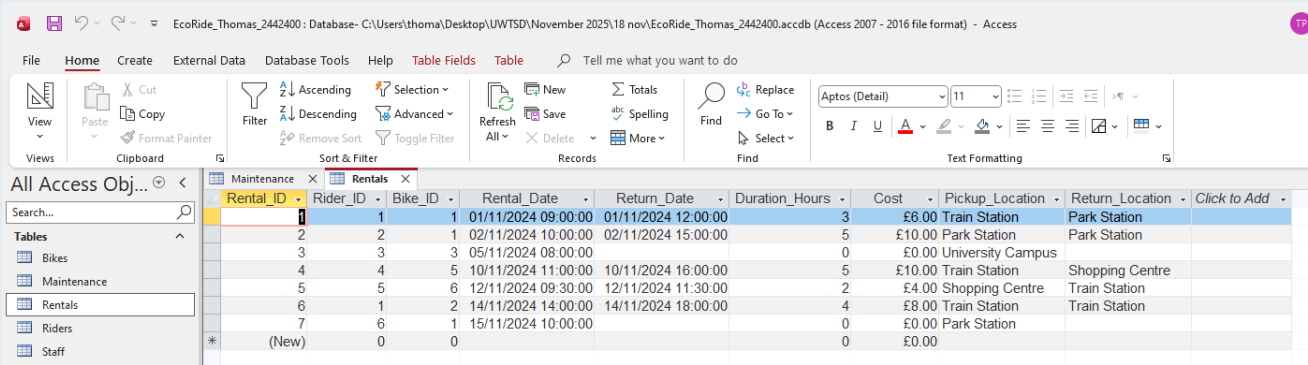
3. RENTALS

Purpose: Account for all rental transactions

Key Attributes:, Rental_ID (Primary key), Rider_ID (Foreign key)

Bike_ID (foreign key), Rental_Date, Return_Date, Duration_Hours, Cost, Pickup_Location, Return_Location

Justification Links riders to bikes. Provides transaction history and revenue tracking. The Return_Date field when NULL indicates active rentals.



Rental_ID	Rider_ID	Bike_ID	Rental_Date	Return_Date	Duration_Hours	Cost	Pickup_Location	Return_Location	Click to Add
1	1	1	01/11/2024 09:00:00	01/11/2024 12:00:00	3	£6.00	Train Station	Park Station	
2	2	1	02/11/2024 10:00:00	02/11/2024 15:00:00	5	£10.00	Park Station	Park Station	
3	3	3	05/11/2024 08:00:00		0	£0.00	University Campus		
4	4	5	10/11/2024 11:00:00	10/11/2024 16:00:00	5	£10.00	Train Station	Shopping Centre	
5	5	6	12/11/2024 09:30:00	12/11/2024 11:30:00	2	£4.00	Shopping Centre	Train Station	
6	1	2	14/11/2024 14:00:00	14/11/2024 18:00:00	4	£8.00	Train Station	Train Station	
7	6	1	15/11/2024 10:00:00		0	£0.00	Park Station		
8	(New)	0			0	£0.00			

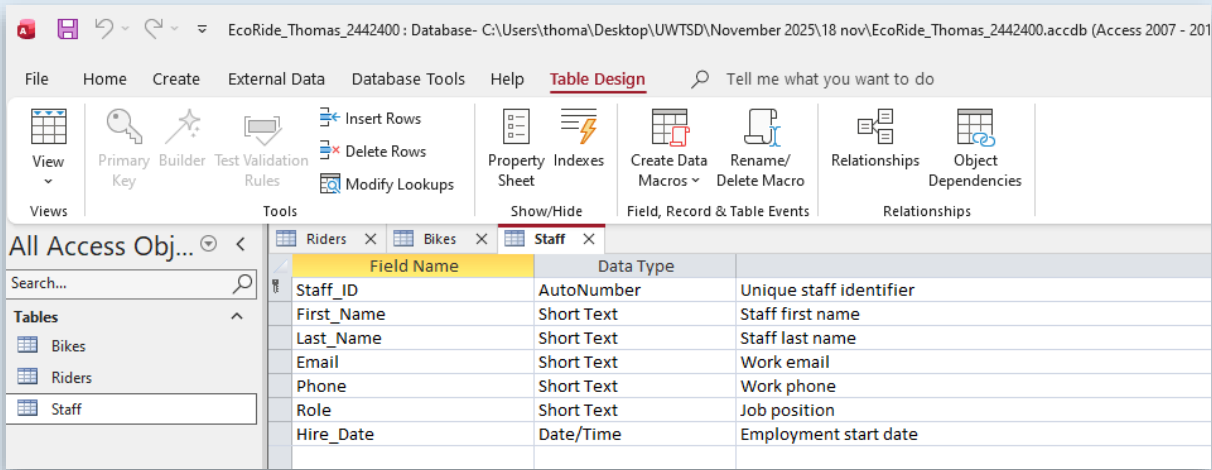
Fig. Rentals_ID

4. STAFF

Purpose: To store information of employees.

Key Attributes: Staff_ID (primary key), First_Name, Last_Name, Email, Phone, Role, Hire_Date

Justification: Tracks who performs maintenance for accountability and Workload management.



Field Name	Data Type	Comments
Staff_ID	AutoNumber	Unique staff identifier
First_Name	Short Text	Staff first name
Last_Name	Short Text	Staff last name
Email	Short Text	Work email
Phone	Short Text	Work phone
Role	Short Text	Job position
Hire_Date	Date/Time	Employment start date

Fig. Staff_ID

5. MAINTENANCE

Purpose: Record any bicycle service and repair work

Key Properties: Maintenance_ID, Primary key; Bike_ID, foreign key
Staff_ID (foreign key), Service_Date, Service_Type, Description,
Next_Service_Due, Cost

Justification: Ensures that bikes are well maintained and safe; schedules
Preventive maintenance to reduce the occurrence of breakdowns.

Field Name	Data Type	Notes
Maintenance_ID	AutoNumber	Unique maintenance record
Bike_ID	Number	Links to Bikes table
Staff_ID	Number	Links to Staff table
Service_Date	Date/Time	When service performed
Service_Type	Short Text	Type of service
Description	Long Text	Service details
Next_Service_Due	Date/Time	Next scheduled service
Cost	Currency	Service cost

Fig. Maintenance_ID

2.3 Relationships Between Entities

The database provides four crucial relationships:

- Riders to Rentals (1:M): One rider can have multiple rentals over
This allows time-tracking of the rental history and customer behaviour.

- Bikes to Rentals (1:M): One bike can be rented multiple times.

This tracks usage patterns and determines the popular types of bikes.

- Bikes to Maintenance (1:M): One bike can have multiple maintenance records. This creates a complete service history for each bike.

- Staff to Maintenance (1:M): There can be one staff member performing multiple maintenance activities. This allows for workload analysis and performance tracking.

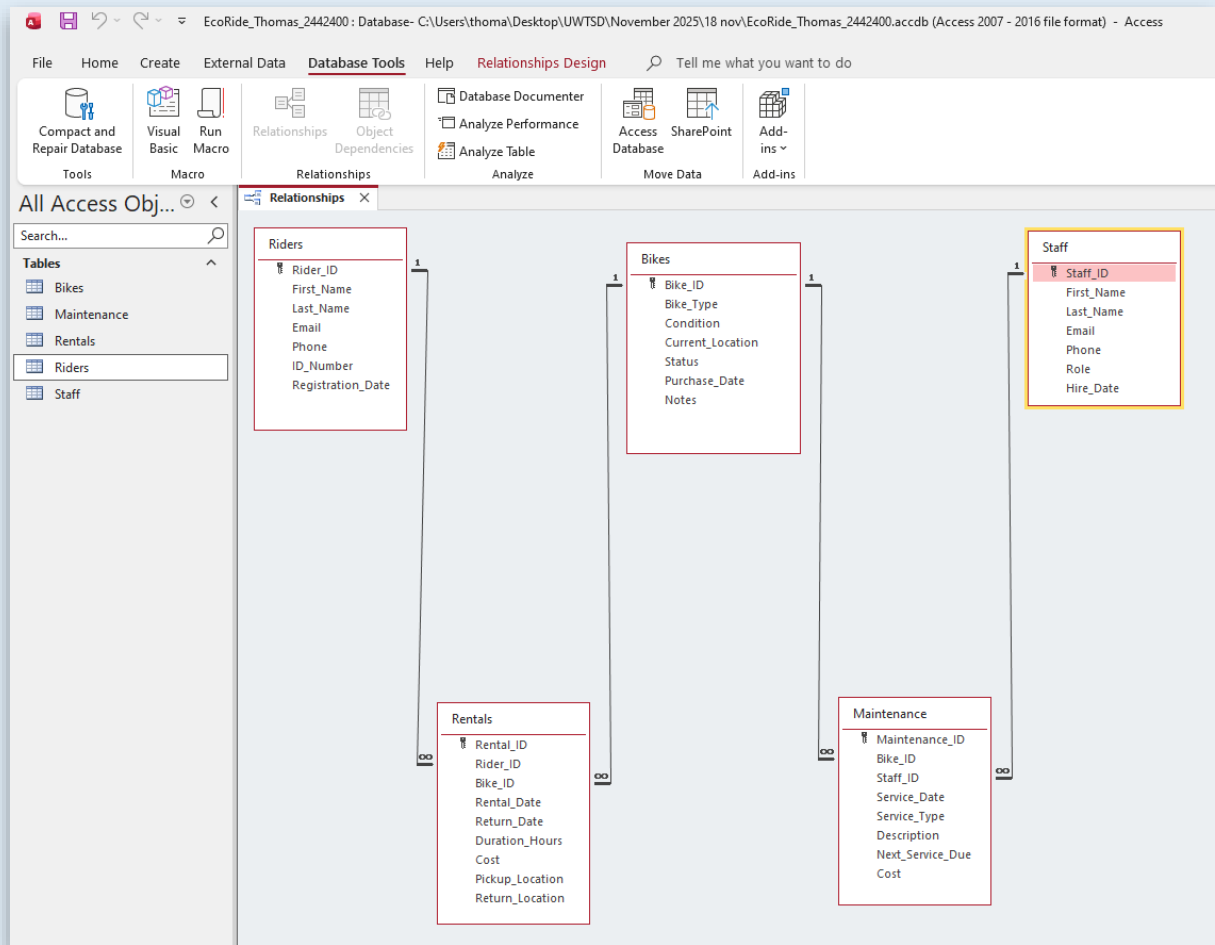


Fig. Relationships Window

2.4 User Requirements Analysis The system shall permit staff to: - Quickly identify available bikes in any location - keep track of which bikes are currently rented and by whom - Scheduling and monitoring of maintenance activities - Generate revenue and usage reports - Prevent double-bookings with automated status updates -Maintain comprehensive history of rentals and maintenance

Entity-Relationship Diagram (ERD)

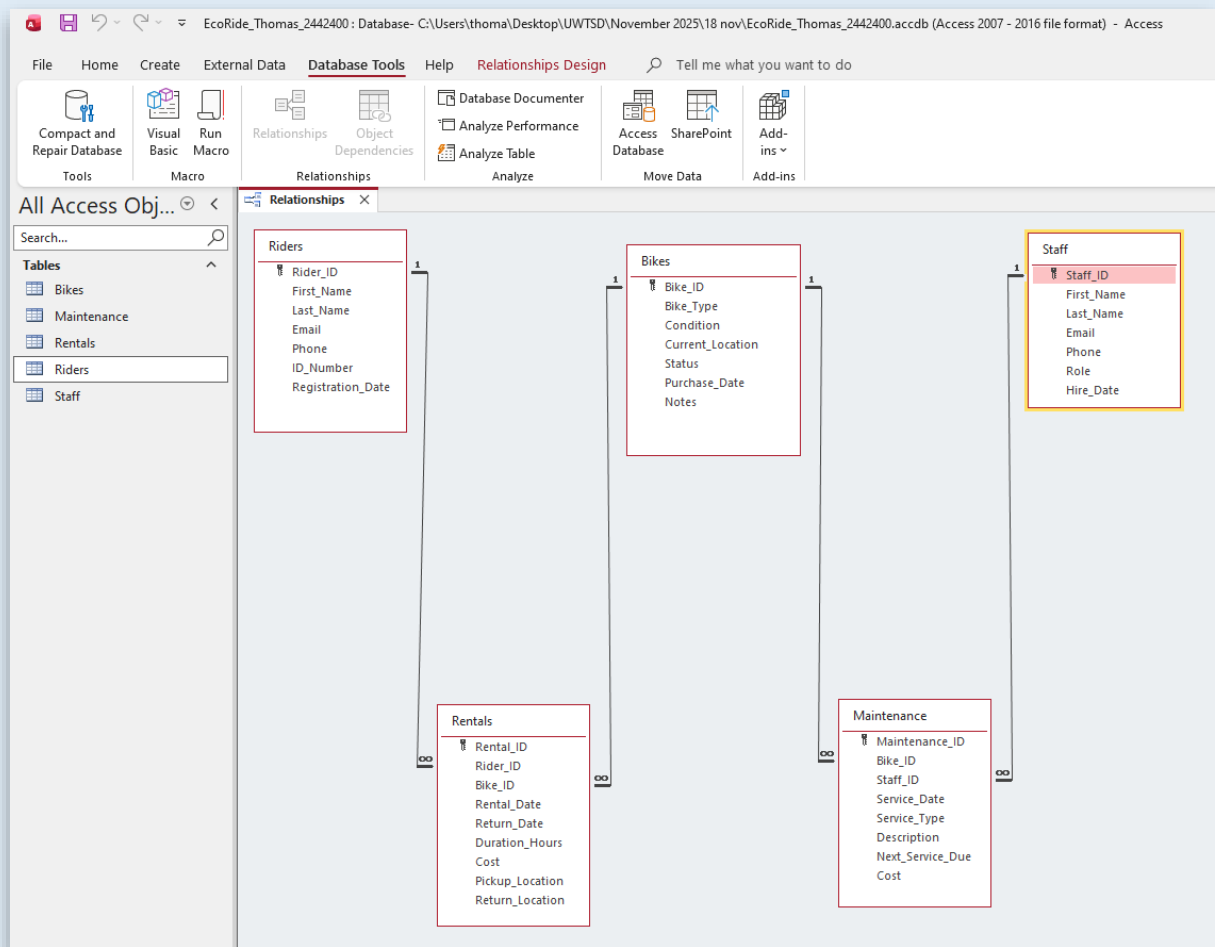


Fig. Relationships Window

Data structure Development

4. DATA STRUCTURE DEVELOPMENT

The structure was implemented in Microsoft Access using five normalized tables with appropriate constraints and data types. [3]

RIDERS TABLE

Primary Key: Rider_ID (AutoNumber) - automatically generates unique identifiers

Fields include First_Name and Last_Name , Email and Phone for

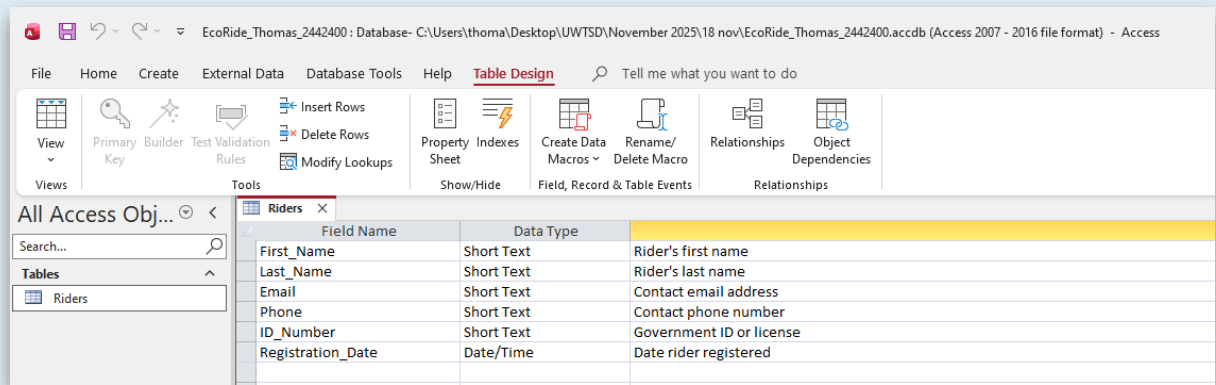


Fig. Fields

contact, ID_Number (Indexed - No Duplicates) to avoid duplication registrations, and Registration_Date with default current date.

General	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

Fig. Rider_ID

BIKES TABLE

Primary Key: Bike_ID (AutoNumber)

Bike_Type *: This categorizes bikes. Condition: This tracks their physical state.

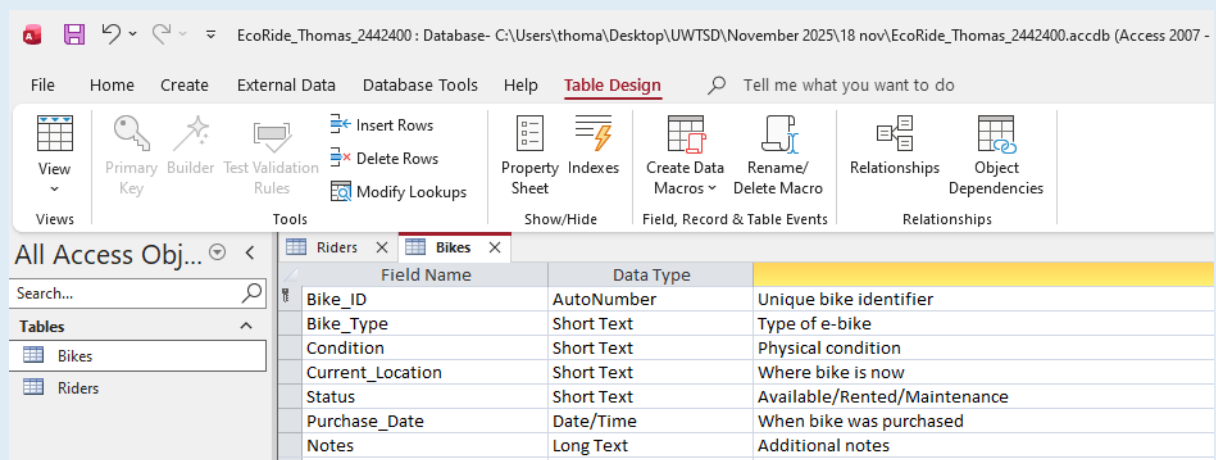


Fig. Byke_Fields

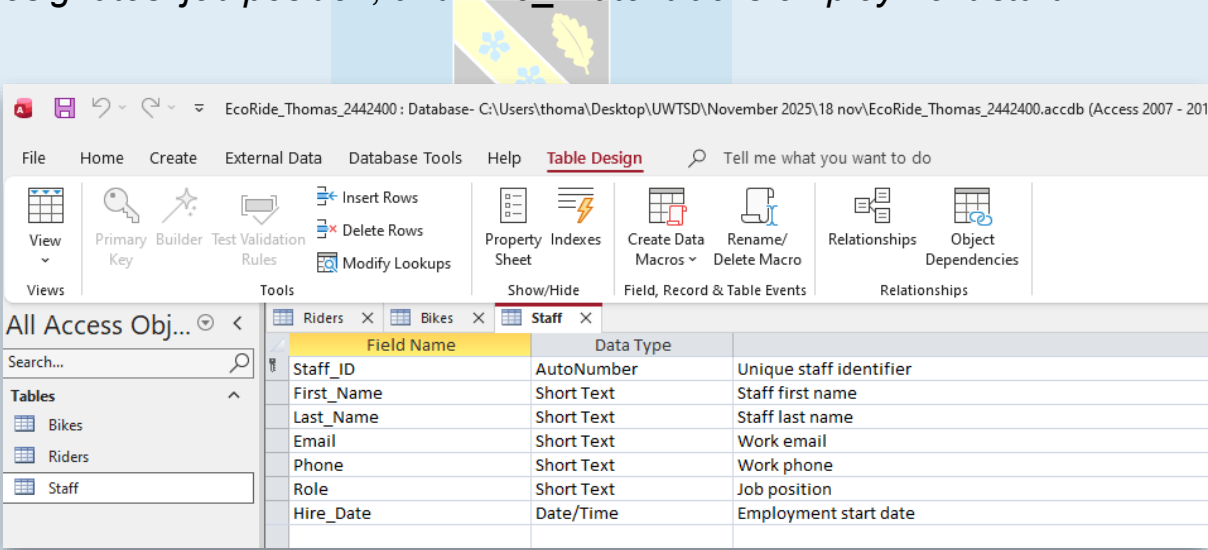
Status has the default value "Available" with a validation rule ensuring only valid statuses, Current_Location tracking real-time position, and Notes.field stores maintenance history.

General Lookup	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

Fig. Byke_ID

STAFF TABLE

*Primary Key: Staff_ID (AutoNumber)
First_Name and Last_Name, Email and Phone for contact, Role Designates job position, and Hire_Date tracks employment start.*



Field Name	Data Type	Description
Staff_ID	AutoNumber	Unique staff identifier
First_Name	Short Text	Staff first name
Last_Name	Short Text	Staff last name
Email	Short Text	Work email
Phone	Short Text	Work phone
Role	Short Text	Job position
Hire_Date	Date/Time	Employment start date

Fig. Staff Table

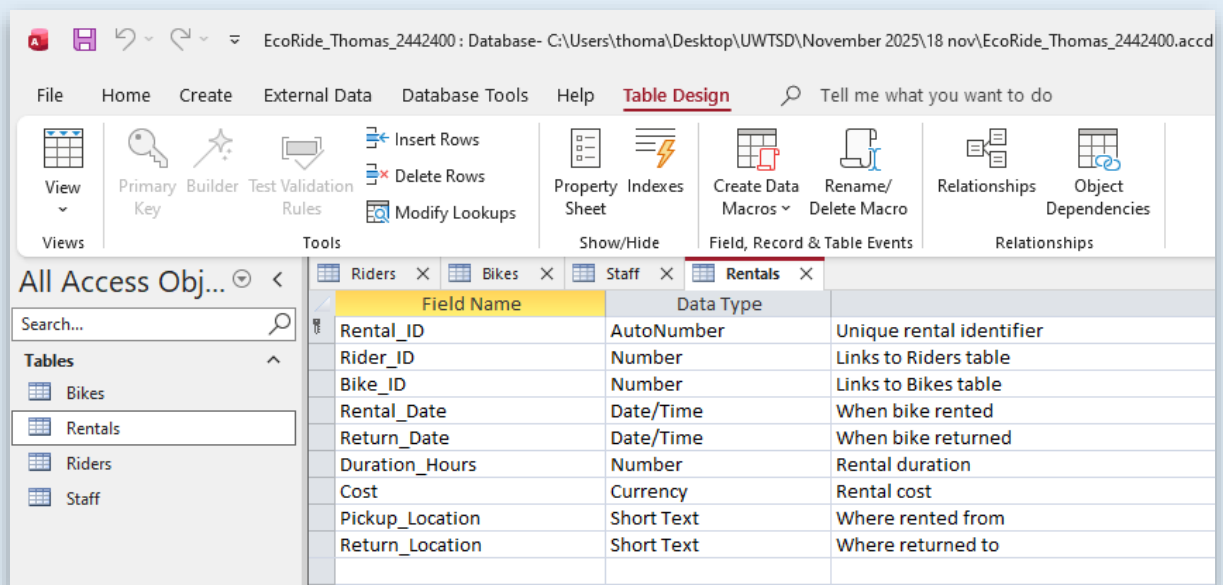
General Lookup	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

Fig. Staff_ID

RENTALS TABLE

Primary Key: Rental_ID (AutoNumber)

Foreign Keys: Rider_ID and Bike_ID (Long Integer) link to respective tables, Rental_Date (Required) marks transaction start, Return_Date (NULL if still rented), Duration_Hours and Cost track rental details, The journey is tracked by Pickup_Location and Return_Location.



Field Name	Data Type	
Rental_ID	AutoNumber	Unique rental identifier
Rider_ID	Number	Links to Riders table
Bike_ID	Number	Links to Bikes table
Rental_Date	Date/Time	When bike rented
Return_Date	Date/Time	When bike returned
Duration_Hours	Number	Rental duration
Cost	Currency	Rental cost
Pickup_Location	Short Text	Where rented from
Return_Location	Short Text	Where returned to

Fig. Rental Fields

MAINTENANCE TABLE

Primary Key: Maintenance_ID (AutoNumber) Foreign Keys: Bike_ID and Staff_ID link to respective tables. Service_Date (Required), Service_Type categorizes work, Description provides details, while Next_Service_Due schedules preventive maintenance. and Cost tracks expenses.

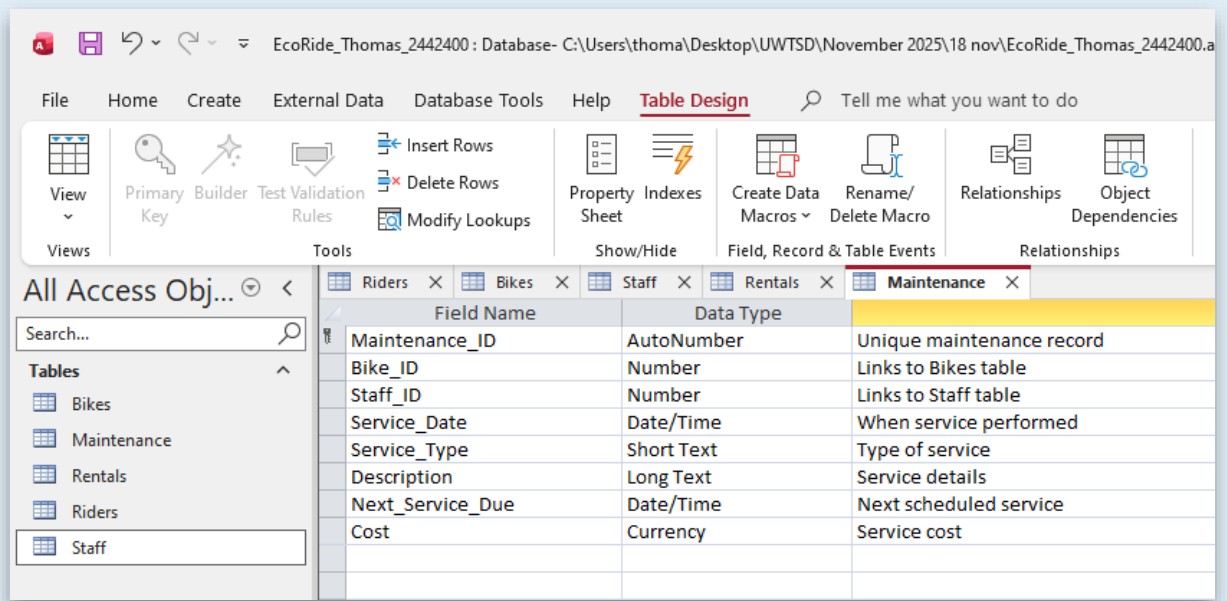


Fig. Maintenance Fields

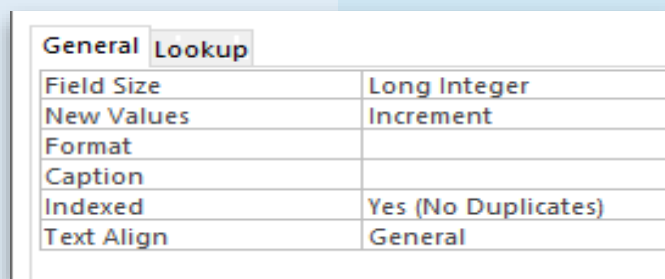


Fig. Maintenance ID

London

Normalization Compliance

5. NORMALIZATION TO THIRD NORMAL FORM (3NF)

The database design adheres to Third Normal Form (3NF) to minimize data.

Anomalies such as redundancy and dependency also occur.

5.1 First Normal Form (1NF)

All tables meet 1NF requirements:

- There is a primary key for each table: *Rider_ID* , *Bike_ID*, *Staff_ID*, *Rental_ID*, *Maintenance_ID*)
- All attributes contain atomic (indivisible) values
- No repeating groups are present
- Each column has a single type of data. [4]

Example: Rather than store "John Smith" in a single field, the Riders table would separates First_Name and Last_Name into separate atomic fields. Phone numbers are stored as single values and not as multiple entries.

5.2 Second Normal Form (2NF)

All tables are in 2NF:

- All tables are in 1NF
- All non-key attributes are fully dependent on the primary key
- No partial dependencies are present

Example: In the Rentals table, Cost and Duration depend on the entire

Rental_ID is not part of a composite key. Each table relies on a single-column primary key - AutoNumber cannot accept partial dependencies. Bike_Type depends purely on Bike_ID, not on any other attributes.

5.3 Third Normal Form (3NF)

All tables are in 3NF:

- All the tables are in 2NF
- No transitive dependencies exist (non-key attributes don't depend on other non-key attributes)

Example Analysis:

Riders table: - Email depends only on Rider_ID, not on First_Name or

Last_Name

- In Rentals table: Cost depends on the rental transaction itself (Rental_ID), not derived from Bike_ID or Rider_ID
 - Bike Current_Location depends only on Bike_ID
- The cost of maintenance depends only on the specific maintenance record.
(Maintenance_ID), not independently on Bike_ID or Staff_ID

5.4 Benefits of Normalization

The normalized form provides:

- ****Reduced data redundancy:**** Once stored, rider information is referenced multiple times through foreign keys
- ****Update anomaly prevention:**** Changing a rider's email updates only one record in the Riders table
- ****No deletion anomaly: **Doesn't delete rider when a rental is deleted or bike information**** - Insertion anomaly avoidance:
- ****New bikes can be inserted without necessitating rental records**
- ****Enhanced data integrity**** with referential constraints
- ****Efficient Storage Utilization****: reduction in requirement for disk space.
- ****Easier database maintenance**** and future scalability

Sample Data Population

6. SAMPLE DATA POPULATION

Realistic sample data was used to populate the database to simulate:

EcoRide operations and enable meaningful query testing.

6.1 Overview of Data

- 6 registered riders representing typical user demographic
- 6 electric bikes of various kinds and conditions
- 5 individual workers performing different roles
- 7 rental transactions including 2 active rentals
- Service history tracked by 7 maintenance records

6.2 Data Characteristics

The sample data represents real scenarios:

- Unique ID number and contact information for riders

- *Bikes in a range of conditions and locations (Excellent, Good, Fair) across the service area*

Active rentals - showing bikes currently in use-where Return_Date is equal to NULL.

- *Completed rentals with calculated duration and cost*

Regular maintenance schedules with next service dates

Different service types include Routine Service, Tire Replacement, Brake Repair, Battery Service, Chain Maintenance) - Several different staff undertaking maintenance activities on the basis of their respective roles This dataset allows for exhaustive testing of all system queries and validates the database design against user requirements.

Query Development

Designed and run a series of practical queries to demonstrate how the system can be used. Example queries might include bikes currently out on rental, bikes due for service, most frequently used bikes or rentals by a particular rider.

QUERY DEVELOPMENT AND TESTING

Seven practical queries were developed to demonstrate the database's functionality and also meet operational requirements for EcoRide.

7.1 Query 1: *Bikes Currently on Rental Purpose: Identify which bikes are currently rented and unavailable for new rentals.*

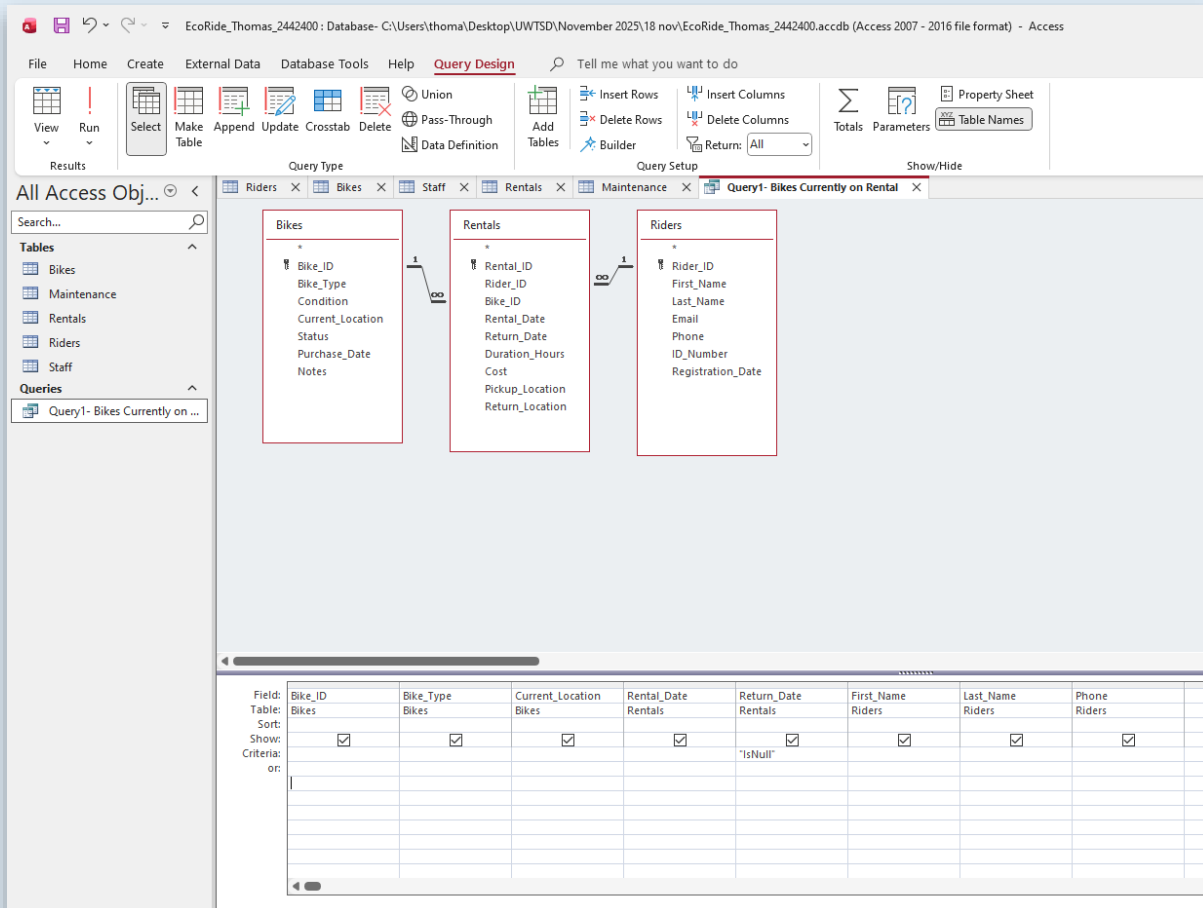


Fig. Query 1

Bike_ID	Bike_Type	Current_Loc	Rental_Date	Return_Date	First_Name	Last_Name	Phone
3	Folding E-Bike	University Cam	05/11/2024 08:00:00		Michael	Brown	07700900125
1	Mountain E-Bike	Park Station	15/11/2024 10:00:00		Olivia	Davies	07700900128

Fig. Query 1 Results

Implementation: This query joins Bikes, Rentals, and Riders tables using "Is Null" criteria on Return_Date to identify active rentals. Results show Bike_ID 3 rented by Michael Brown and Bike_ID 1 rented by Olivia Davies. Business

Value: Staff can instantly identify unavailable bikes without manually checking records.

7.2 Query 2: Bikes Due for Service Soon, Purpose: Proactive maintenance scheduling within 30 days.

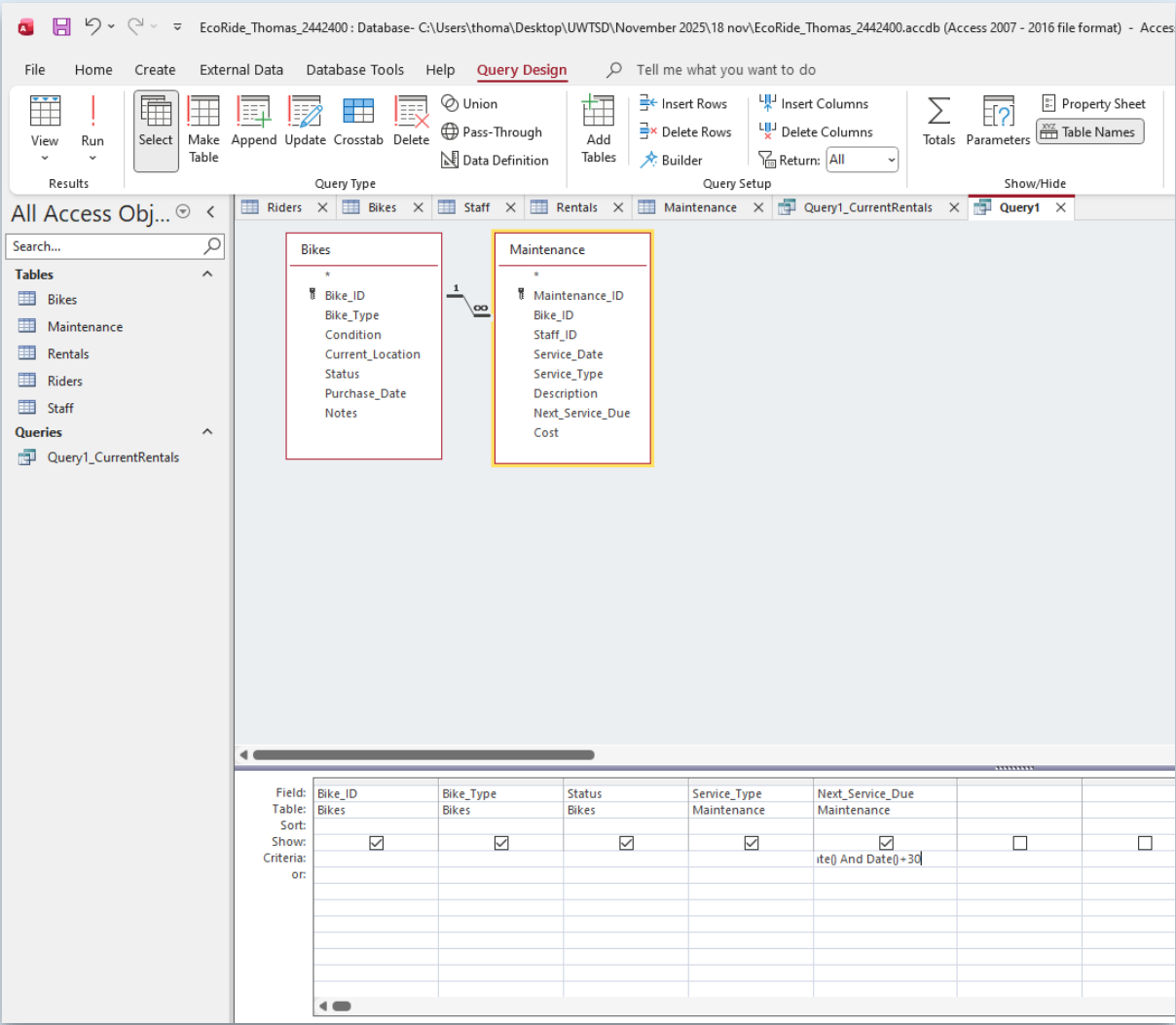


Fig. Query 2 Design

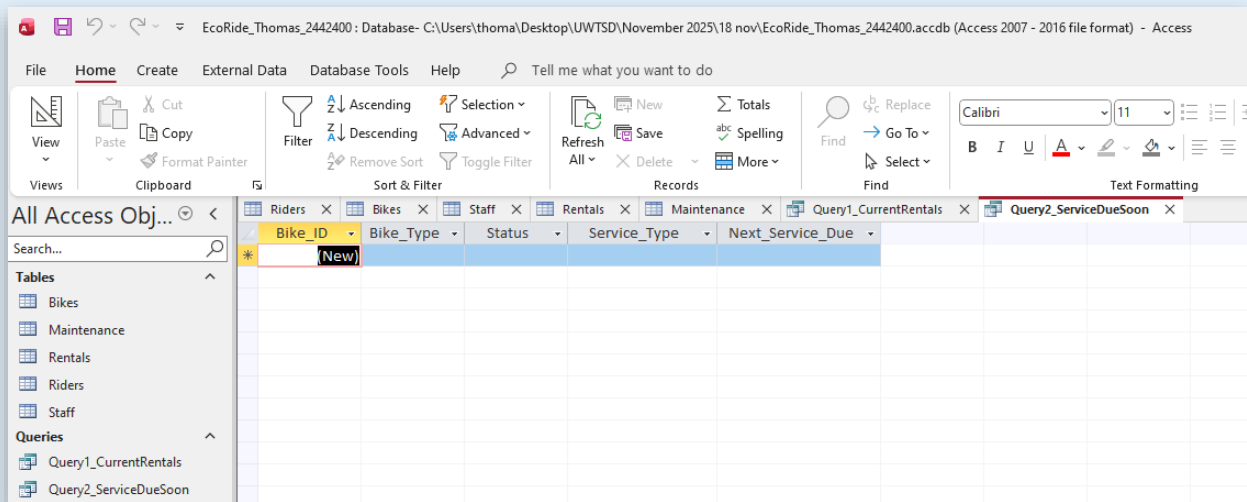


Fig. Query 2 Results

Implementation: Uses "Between Date() And Date()+30" expression to calculate bikes requiring service. Date() function returns current date.

Business Value: Prevents bike failures by ensuring proactive maintenance. Improving safety and customer satisfaction. Purpose: Which bikes are most popular?

7.3 Query 3: Most Frequently Rented Bikes Purpose: To identify high-demand bikes for inventory planning.

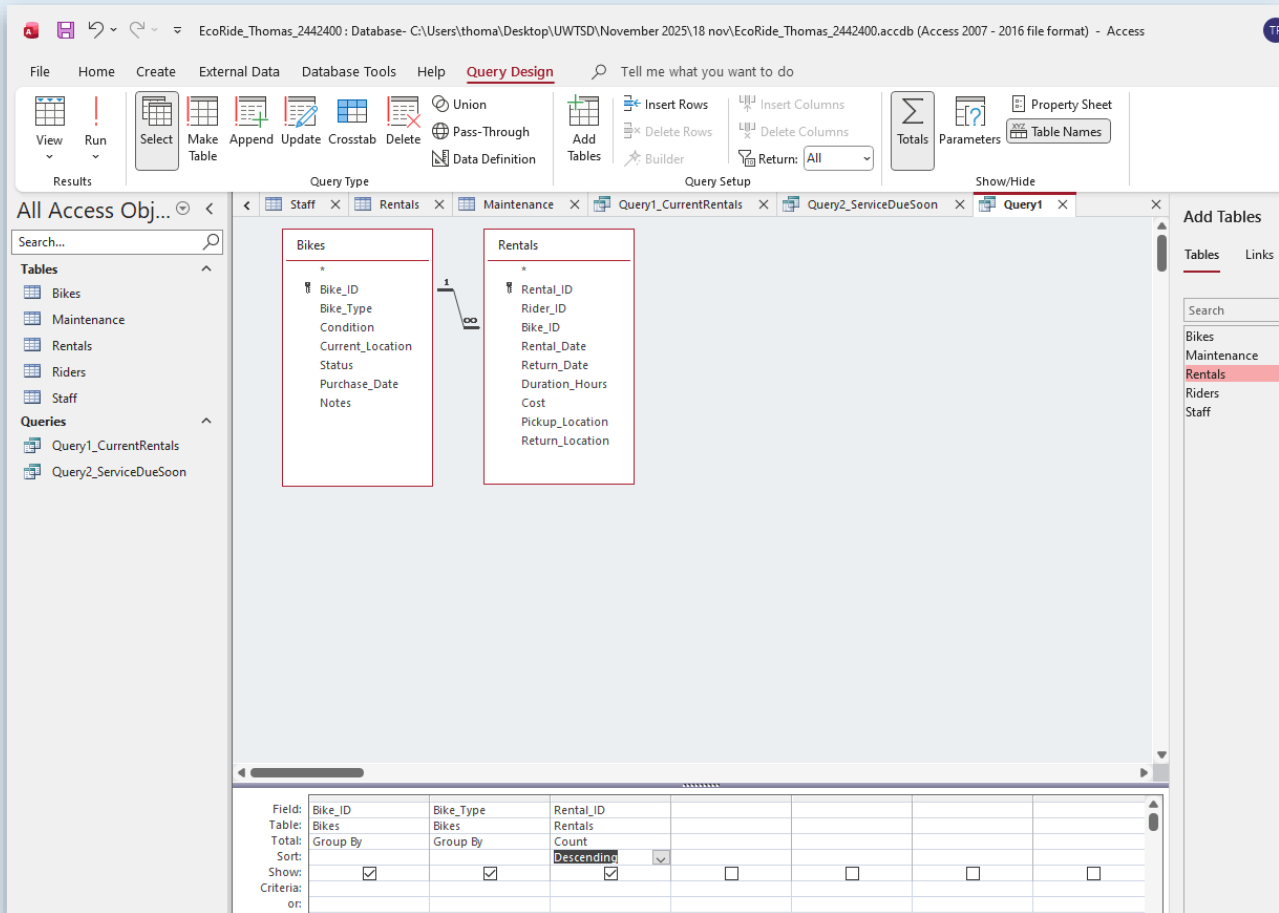


Fig. Query 3 Design

Bike_ID	Bike_Type	CountOfRental
1	Mountain E-Bike	3
6	City E-Bike	1
5	Mountain E-Bike	1
3	Folding E-Bike	1
2	City E-Bike	1

Fig. Query 3 Results

Implementation: Uses Count aggregate function with Group By to calculate rental frequency per bike. Results sorted descending show Bike_ID 2 and 1 are most popular.

Business Value: Informs purchasing decisions for fleet expansion.

7.4 Query 4: Total Revenue Generated

Purpose: financial reporting and performance tracking. Calculate total income

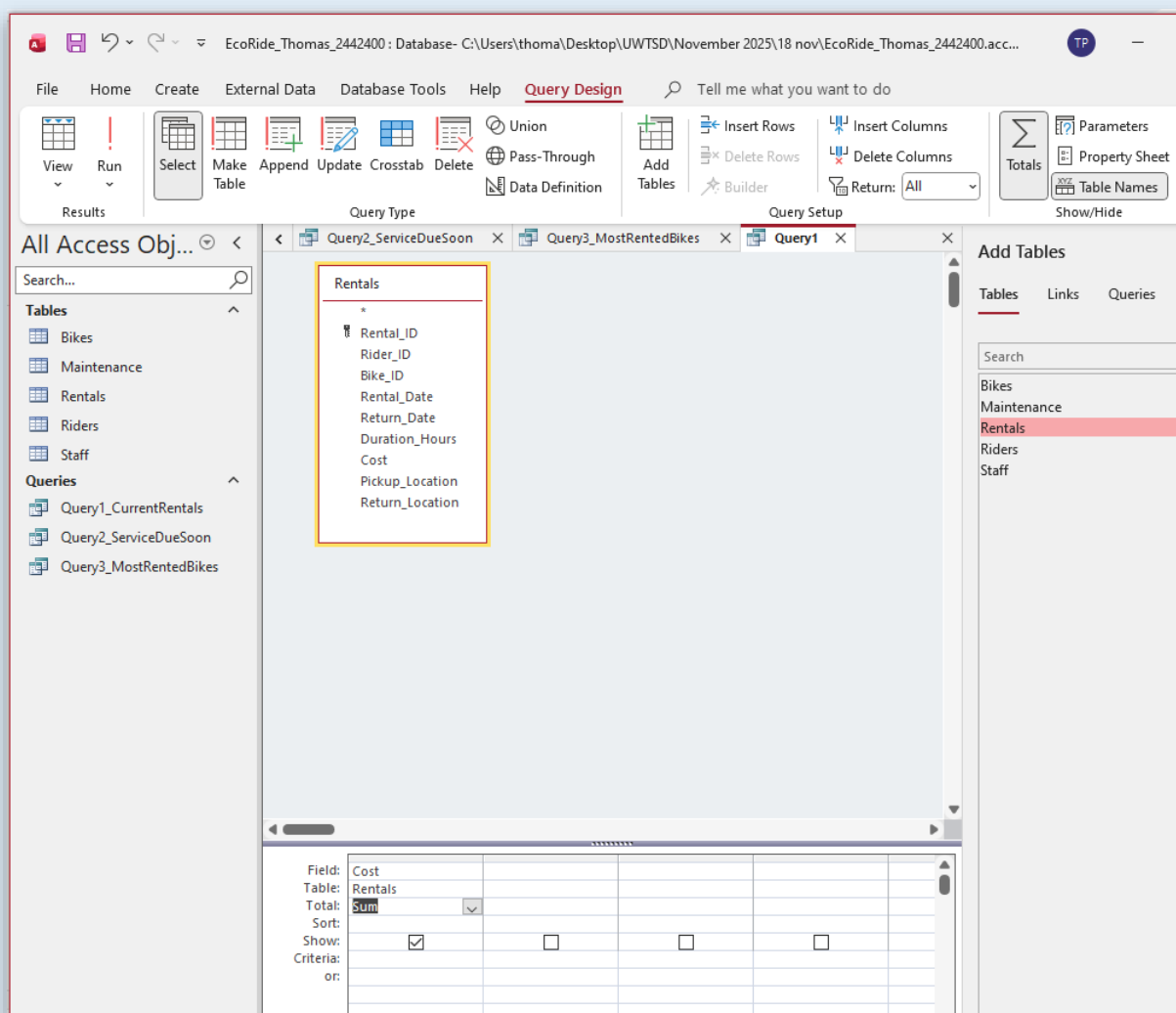
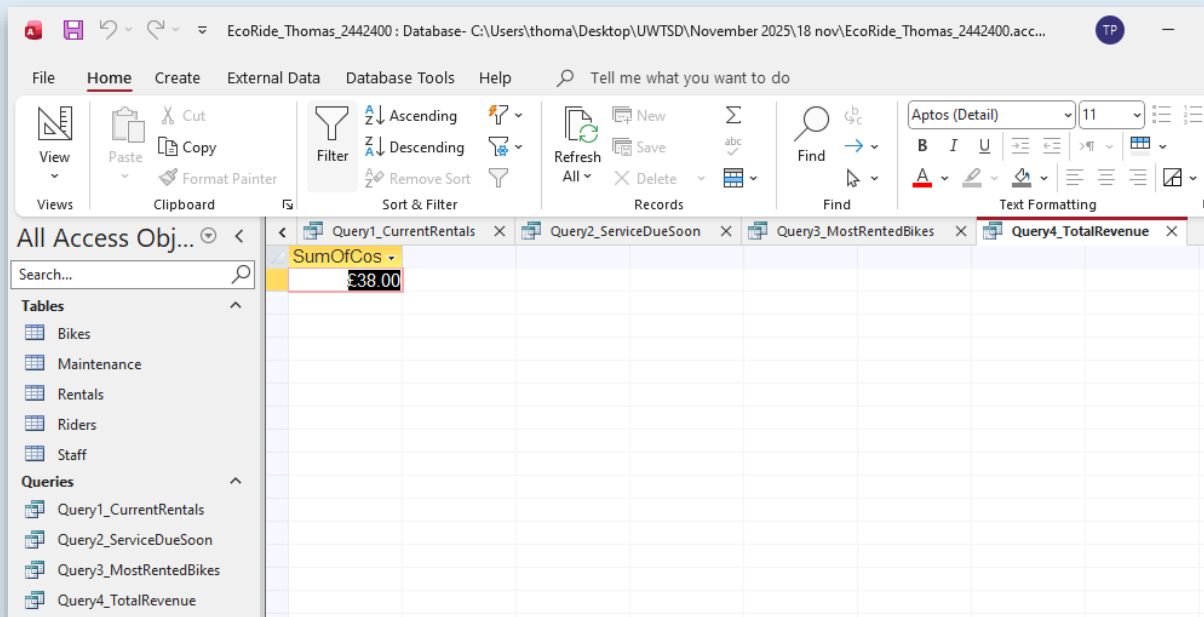


Fig. Query 4 design



SumOfCos
£38.00

Fig. Query 4 Results

Implementation: Uses Sum aggregate for the Cost field. Current total Revenue is £38.00.

Business Value: Quick financial overview for business reporting.

7.5 Query 5: Rental History for Specific Rider

Purpose: Customer service and dispute resolution.

Shows John Smith's 2 rentals

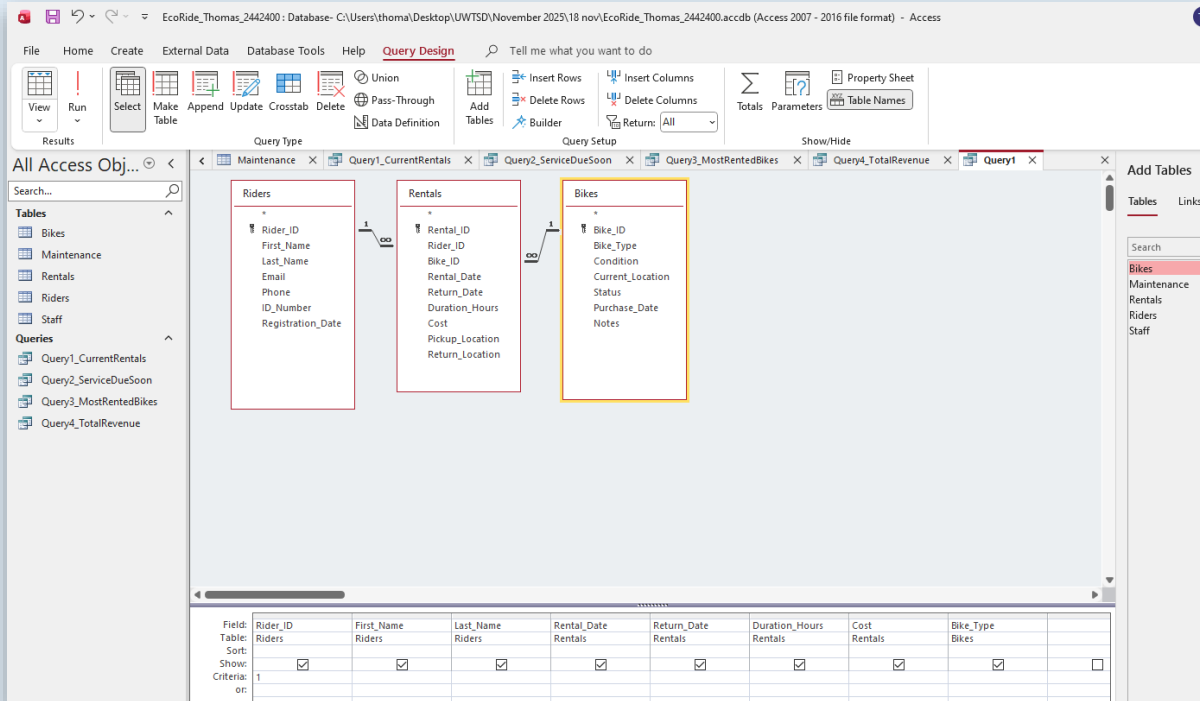


Fig. Query 5 Design

Rider_ID	First_Name	Last_Name	Rental_Date	Return_Date	Duration_Hours	Cost	Bike_Type
1	John	Smith	01/11/2024 09:00:00	01/11/2024 12:00:00	3	£6.00	Mountain E-Bike
1	John	Smith	14/11/2024 14:00:00	14/11/2024 18:00:00	4	£8.00	City E-Bike
Total:							£14.00

Fig. Query 5 Results -Shows John Smith's 2 rentals

Implementation: Filters by Rider_ID=1, showing the 2 rentals of John Smith

Total: £14.00.

Business Value: Immediate access to customer history.

7.6 Query 6: Staff Workload Analysis

Purpose: Resource management and performance evaluation.

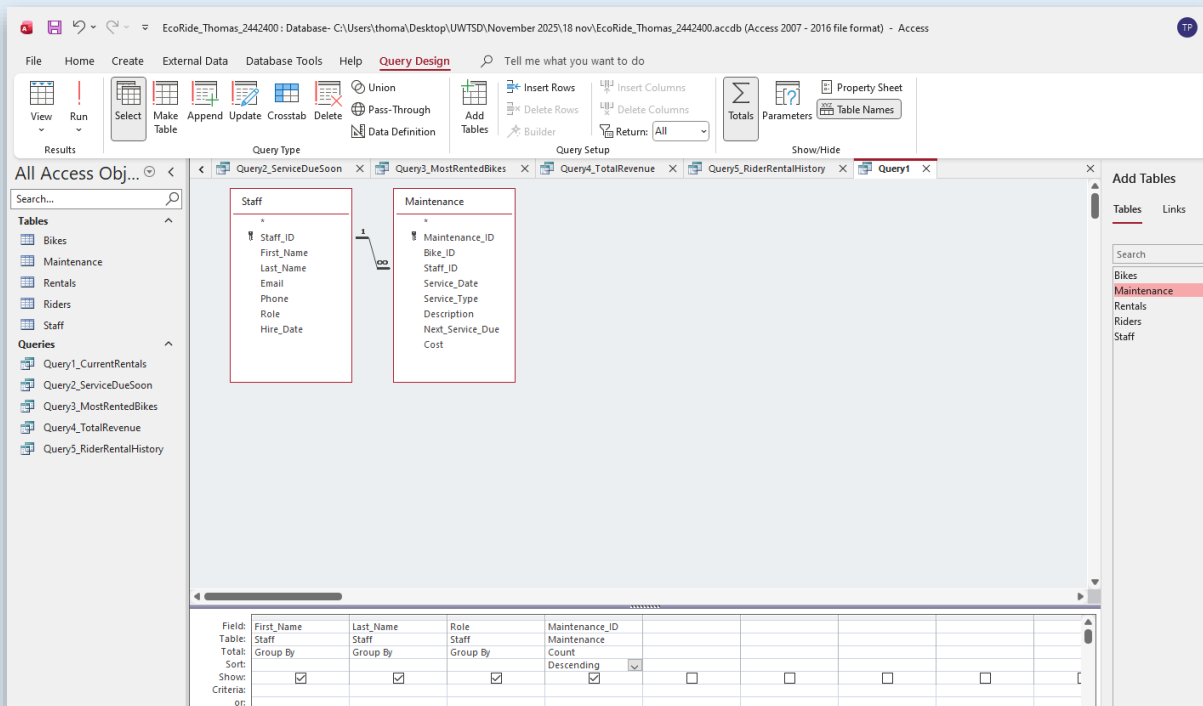


Fig. Query 6 Design

First_Name	Last_Name	Role	CountOfMai
Lisa	Morgan	Senior Mechanic	2
David	Roberts	Mechanic	2
Tom	Hughes	Technician	1
Rachel	Evans	Mechanic	1
Mark	Thomas	Inspector	1

Fig. Query 6 Results

Implementation: Counts maintenance records per staff member. Lisa Morgan David Roberts performed 2 tasks each.

Business Value: Enables fair work distribution.

7.7 Query 7: Bikes Needing Attention

Purpose: Operational status monitoring.

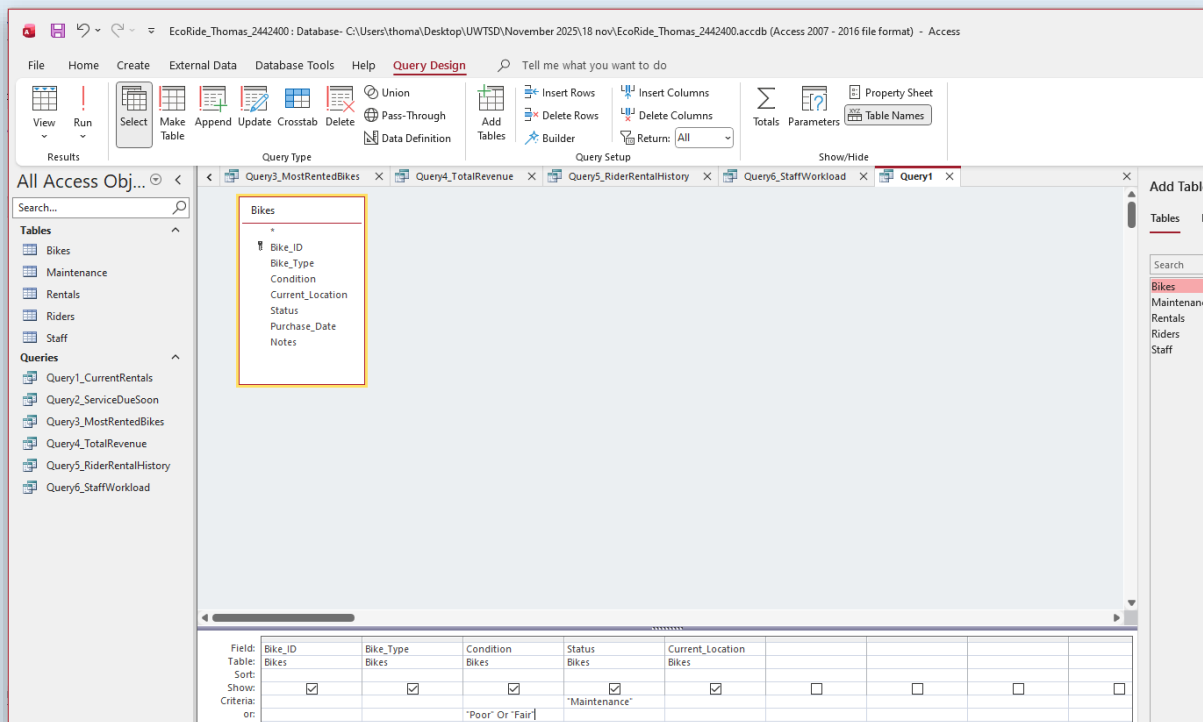


Fig.Query 7 Design

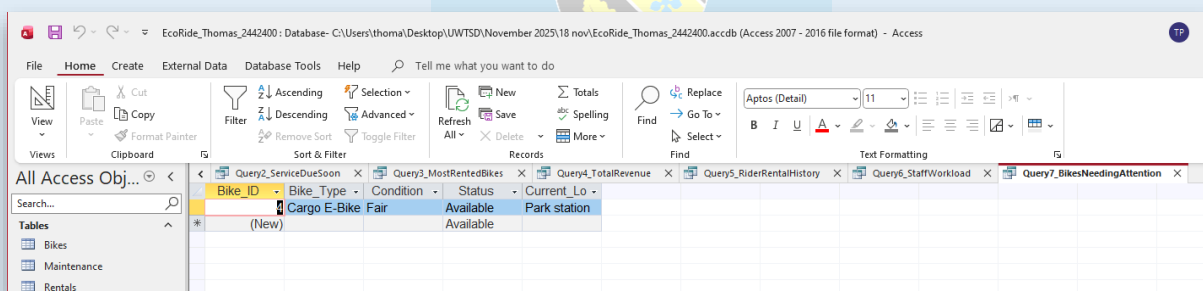


Fig.Query 7 Results

Implementation: Uses OR logic to find bikes in Maintenance status OR Fair/Poor condition.

Business Value: Identifies problem bikes before customer complaints.

7.8 Query 8 Testing Summary

All queries executed successfully with accurate results. Joins functioned correctly via foreign keys, aggregate functions calculated proper totals,

date functions identified current/future dates correctly, and criteria Filters returned appropriate data subsets.

Validation and Testing

VALIDATION AND TESTING

Comprehensive testing confirmed data integrity and functional accuracy.

8.1 Data Integrity Testing

Test: Enforcing Referential Integrity Can't delete Rider_ID 1: John Smith, because it has existing rentals.

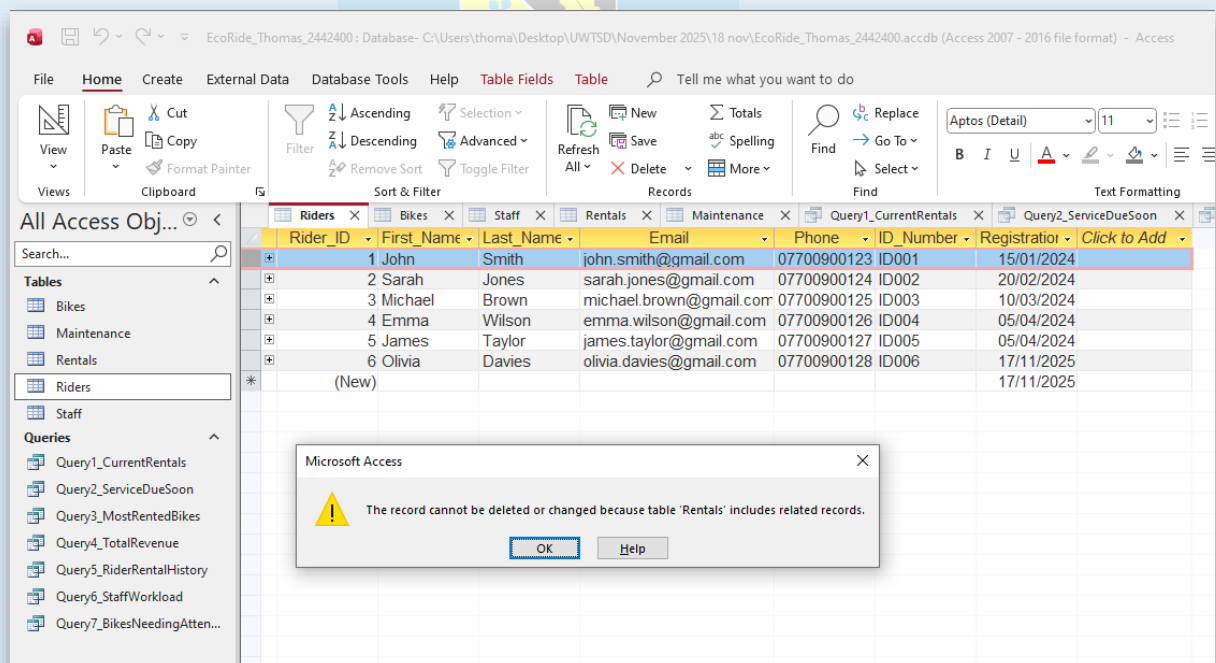


Fig. Error message "The record cannot be deleted or changed because table 'Rentals' includes related records"

This error message (proves referential integrity works!)

Result: Access prevented deletion with error "The record cannot be

deleted or changed because table 'Rentals' includes related records."

This confirms that referential integrity functions as it should, preventing deleted records. [5]

8.2 Functional Requirements Verification

Register riders: Successfully add with validation

Record maintenance for bikes: Full inventory with status tracking

Record transactions: Full rental history along with costs

Track staff: Maintenance linked to employees

Prevent double-bookings: Status field prevents renting unavailable bikes

*Identify unreturned bikes: In Query 1 identifies active rentals
Generate reports: Queries provide analytical views
All critical requirements met and validated through testing.*

Conclusion

REFLECTION AND EVALUATION

Design Decisions

Having separate Riders and Staff tables was clearer than having a single

"Users" table. The Rentals table acts as both a junction table and Transaction record. Using AutoNumber for primary keys ensures unique

identifiers automatically. The Bike Status field, although creating slight

Redundancy provides considerable performance gains in availability.

Challenges Encountered

Understanding that foreign key tables must be populated first was

initially difficult. Date/Time format required in Microsoft Access the particular format (dd/mm/yyyy hh:mm). Aggregate functions of Query 3

required additional research on GROUP BY concepts.

field: Validation rules: learning of Access expression syntax was necessary.

Future Improvements

Duration and bike type. Access forms would improve the usability of data entry. will go through more examples and use the help function in access to understand the subject. More practice on different database and most important work on assessments much earlier than 2 weeks to get the accurate or best results in assessments.

This project successfully designed and implemented a functional relational

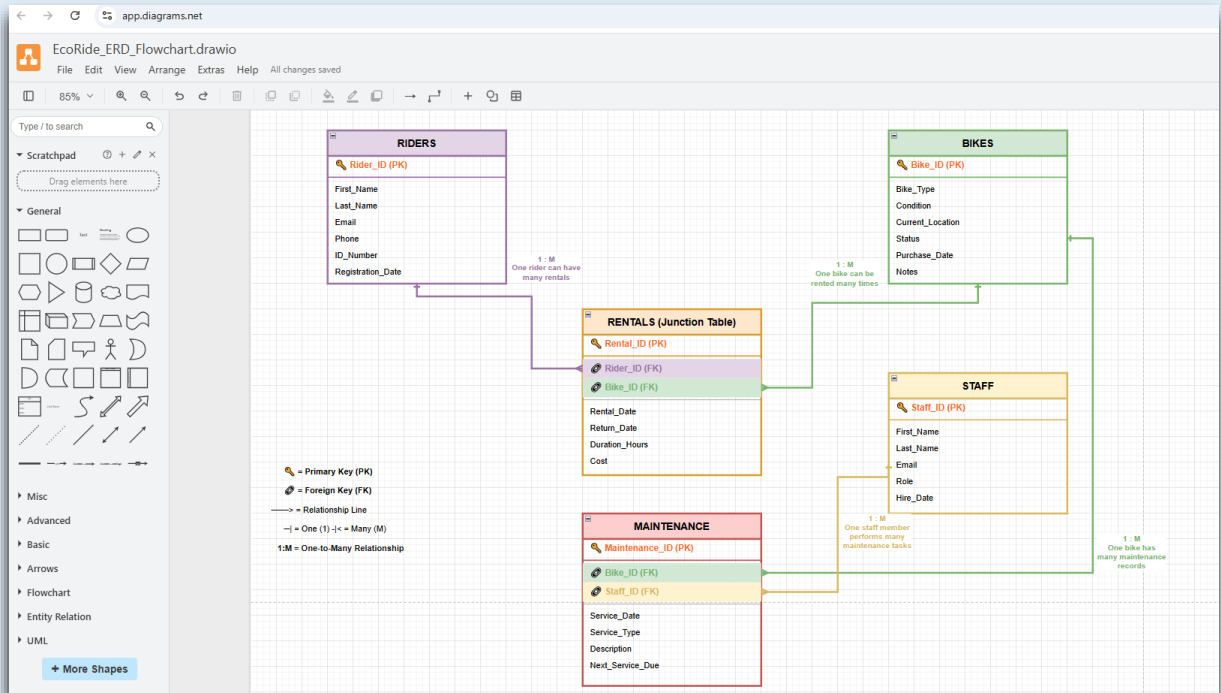
database system for EcoRide using Microsoft Access. The solution addresses

operational challenges through structured data storage

The database is in Third Normal Form (3NF)

The implemented solution allows managing operations efficiently, prevents double-bookings, keeps track of unreturned bikes, and generates accurate reports. This project demonstrates understanding of database Concepts and practical application to real-world business problems.

Entity Relationship Rough Diagram :-



References

- [1] M. Fayard, "What is Third Normal Form (3NF)?," 18 November 2024. [Online]. Available: https://www.datacamp.com/tutorial/third-normal-form?utm_source.
- [2] E. &. Navathe, "fundamentals of database systems," Pearson, 2016. [Online]. Available: <https://www.pearson.com/en-gb/subject-catalog/p/fundamentals-of-database-systems/P200000003484>.
- [3] M. corporation, "Access help & learning," Microsoft support, 2024. [Online]. Available: <https://support.microsoft.com/en-us/access>.
- [4] C. &. Begg, "database systems a practical approach to design implementation and management," Pearson, 2015. [Online]. Available: <https://www.pearson.com/en-gb/subject-catalog/p/database-systems-a-practical-approach-to-design-implementation-and-management/P200000003181>.
- [5] H. K. a. S. S. Silberschatz, "Database System Concepts, 7th Edition," McGraw Hill Education, 2019. [Online]. Available: <https://www.mheducation.com/highered/product/database-system-concepts-silberschatz-korth/M9780078022159.html>.

