Eco Ride – An Electric Bike Rental Scheme

Name: Imran Wali Khan

Student ID: 2441420

Module: Introduction to Web and Database Concepts

Assignment: ACCA4023

Submission Date: 25/11/2025

Task 1:

The Eco_Ride system operates through multiple fundamental elements which function as a complete network to deliver a dependable rental service. The system relies on riders as its core structure because it keeps track of their personal information, which includes names and contact details and ID numbers and registration dates, to enable user identification and rental history tracking. The fleet of bikes operates as a crucial system which tracks every bike through its specific characteristics that include type classification and condition status and serial number and battery status and current station position. The bikes operate through the network of stations, which function as rental points throughout the system. The system tracks the entire fleet through its operational status and distribution and availability at a single glance.

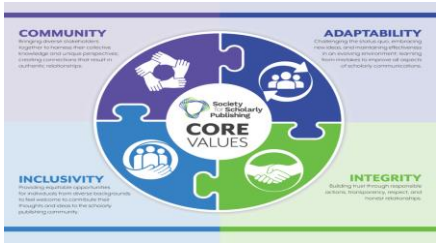
Two extra entities exist to support the operational layers which keep the system operational and maintain proper functionality. The system tracks every rental transaction through Rental Transactions, which documents rider information and bike details and rental start and end locations and timestamps and costs and user-reported issues to establish an unbroken record of operational data for planning needs and system oversight. The Staff entity keeps track of employees who have specific roles and contact information which connects them to Maintenance Records that show every inspection and repair and service work completed on bikes, including dates and types of work and upcoming service needs. The interconnected entities establish a structured database system that substitutes manual paperwork with organised data tracking to help Eco_Ride monitor availability, maintenance, and usage patterns more effectively.

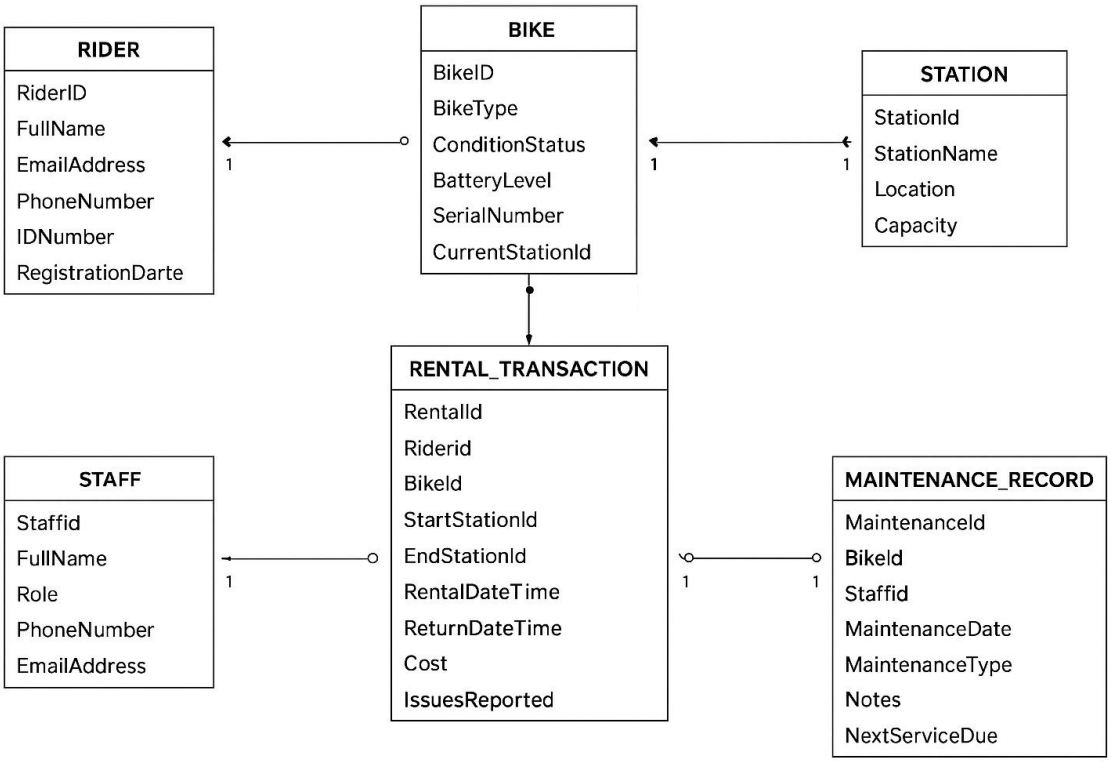
c

Task 2:

The Entity–Relationship Diagram (ERD) for the Eco_Ride system is built around six core entities that work together to support all aspects of the electric bike rental service. The Rider entity needs a unique identifier called Rider ID to store vital user details, which include their names and contact information and identification numbers and registration dates. Riders maintain a one-to-many connection with the Rental Transaction entity because each rider can rent multiple bikes throughout their rental history. The Bike entity stores details about electric bikes through its Bike ID identifier, which includes information about bike type and battery status and condition and serial number and current station location. Each bike exists in multiple rental records, which creates an additional one-to-many relationship. The Station entity functions as a rental and return point system which uses Station ID to identify each location while storing information about station names and their positions and maximum capacity. The system requires two connections between stations and rental transactions because stations function as both start and end points for transactions, and stations must link to bikes since stations store multiple bikes at any given time.

The Maintenance Record entity uses Maintenance ID as its primary key to track all service activities and inspections and repairs performed on the bikes. The maintenance records maintain foreign key connections to Bike ID and Staff ID, which establish one-to-many links between the Bike and Staff entities. The Staff entity uses Staff ID to track employee details, which include their names and job titles and contact information, and enables tracking of maintenance work performed by each staff member. The entities create a complete ERD which tracks rider movements and bike status and station operations and maintenance records. The database structure enables Eco_Ride to operate with an efficient, scalable system which delivers operational transparency, prevents double bookings, and supports future planning and service optimisation and reporting needs.





Task 3:

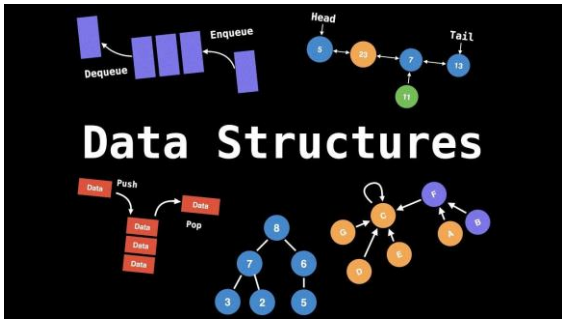
Summary: Data Structure Development

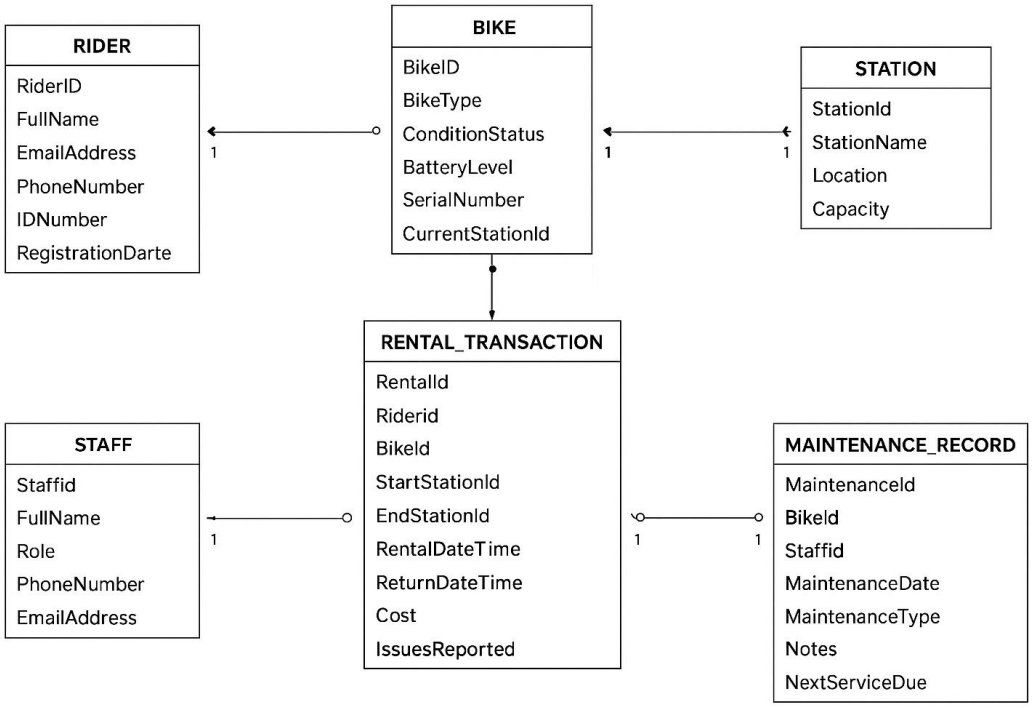
The Eco_Ride database system operates through six fundamental entities, which include Rider, Bike, Station, Rental Transaction, Staff and Maintenance Record. The system employs primary keys to guarantee unique identification for each entity while storing vital operational data through their respective attributes. Foreign key constraints establish relationships between these entities to protect data integrity by stopping the creation of both incomplete and invalid records.

The Rider entity links to Rental Transaction through a foreign key, ensuring all rentals belong to valid riders. The Bike entity is connected to stations and referenced in both rental and maintenance records, guaranteeing that every rental or maintenance action is tied to an existing bike. The station serves as the reference point for both rental start and end locations and bike tracking purposes to maintain accurate location monitoring. The system maintains accurate relationships through multiple foreign keys that exist in Rental Transaction (Rider ID, Bike ID, Start Station ID, End Station ID). The Staff entity links to Maintenance Record so all maintenance actions are traceable to a real staff member. The Maintenance Record

system stores complete service data, which uses foreign keys to link maintenance activities to their corresponding bike and staff member records.

The database system for Eco_Ride achieves stability through its use of primary keys and foreign keys and NOT NULL constraints, which maintain data integrity for all rental transactions and maintenance activities and station information.





Task 4:

Normalization Compliance (Up to Third Normal Form – 3NF)

The Eco_Ride data model follows Third Normal Form (3NF) requirements to create a database design which reduces data duplication, prevents update problems, and maintains strong data integrity.

The database achieves First Normal Form (1NF) because it lacks repeating groups and multi-valued fields which create a clean tabular structure for each table.

Second Normal Form (2NF) is satisfied by ensuring that all non-key attributes in every table depend on the whole primary key. The system prevents partial dependency because each entity contains a single attribute as its primary key such as Bike ID and Rider ID. The attributes Rental Date Time and Cost and Return Date Time in the Rental Transaction table depend on the primary key Rental ID but they do not depend on the foreign keys separately. Maintenance Record service information depends on Maintenance ID instead of Staff ID or Bike ID individually.

The Rider entity stores information about riders but does not store any information about rentals or stations. The Bike table stores information about bikes, but it does not include any station information except for the Current Station ID foreign key. Station contains only attributes related to station identity. The Rental Transaction table keeps rental details by linking to existing rider, bike and station records through foreign keys instead of copying their information. The Maintenance Record table stores all maintenance details, but it links staff names and bike data through foreign keys rather than duplicating them.

Each entity possesses only attributes which directly correspond to its primary key during this design process while avoiding any duplicate data throughout the entire system. The Eco_Ride database has reached Third Normal Form (3NF) which enables optimised storage and dependable updates and maintains correct data relationships between rentals, maintenance, stations, bikes, and user information.

Task 5:

Sample Data

RIDER

RiderID	FullName	Email	Phone	StationID
R001	Alice Morgan	alice.m@example.com		S001
R002	Daniel Harper	dan.h@example.com		S002
R003	Priya Shah	priya.shah@example.com		S003
R004	Ethan Brooks	e.brooks@example.com		S004
R005	Mia Thompson	mia.t@example.com		S005

STATION

StationName	Locaton	Capacity	Capamity
Central Park Hub	Central Park Entra—	25	25
Riverside Station	Riverside Road	18	18
Uni North Gate	University North Gate	30	30
City Train Station	Mainline Platform 2	20	20
Lakeside Stand	Lakeside Park West	15	15

BIKE

BIkeID	BikeType	Origin
B001	Standard E-Bike, Ad-	
B002	Standard E-Bike, In	
B003	Mountain E-Bike, A	
B004	Standard E-Bike, A	
B005	Cargo E-Bike	

STAFF

StaffiD	FullName	Role
ST001	Karen Wells	Techniciar
ST002	Jacob Turner	Inspector
ST003	Olivia Price	Techniciar
ST004	Samir Malik	Mechanic
ST005	Chioe Richards	Supervis

STAFF

Rental ID	Rider Ile ID	MaintenanceDate	Dehail
R001	B003	2024-05-10 09:15	2024-07-02
R002	B004	2024-05-12 14:20	2024-08-01
R003	B001	2024-05-13 11:10	2024-06-10
R004	B005	2024-05-14 16:00	2024-08-13
R005	B002	2024-05-15 08:30	B200 Surem

RENTAL TRANSACTION

RentalID	RiderID	BikeID	StartStationID	EhdStatonID
RT001	R001	B002	2024-05-10:08 15	S001
RT002	R003	B004	2024-05-12 14:20	S002

MAINTENANCE RECORD

Maintenane ID	BikeID	StaffiD	MaintenanceType	Notes NOtes
M001	B003	ST004	Brake adjustment	Rear brake pads
M002	B005	ST001	Safery inspection	Passed all checks

Conclusion:

The Eco_Ride database system provides a structured, reliable, and fully normalised foundation for managing the growing electric bike rental service. Through the identification of main entities, which include riders, bikes, stations, rentals, staff and maintenance records, and their attributes and relationship definition, the system removes all problems that used to appear with paper-based systems. The ERD and 3NF-compliant design deliver a logical data storage system that prevents duplication and enables secure data modification and retrieval operations. The sample data shows how real-world business processes function in practice to support essential operations like rider registration and bike tracking and rental recording and maintenance planning. The database solution allows Eco_Ride to run operations efficiently while maintaining data quality and supporting business growth and tracking performance for better accountability.

Task 6:

Query	SQL Query

1. **Bikes Currently Out on Rental**	`SELECT`
	`  rt.RentaLID,`
	`  rt.BikeID,`
	`  r.FullName AS RiderName,`
	`  rt.RentalDateTime,`
	`  rt.StartStationID`
	`FROM Rental_TranSaction rt`
	`JOIN Rider r ON rt.RiderID = r.RiderID`
	`WHERE rt.ReturnDateTime IS NULL;`

2. **Bikes Due for Service**	`SELECT`
	`  m.BikeID,`
	`  b.BikeType,`
	`  m.MaintenanceDate,`
	`  m.NextServiceDue,`
	`  m.MaintenanceType`
	`FROM Maintenance_Recod m`
	`JOIM Bike b ON m.BikeID = b.BikeID`
	`WHERE m.NextServiceDue <= CURRENT_DATE;`

3. **Most Frequently Used Bikes**	`SELECT`
	`  BikeID,`
	`  COUNT(*) AS TotalRentals`
	`FROM Rental_Transaction`
	`GROUP BY BikeID`
	`ORDER BY TotalRentals DESC;`

4. **Rentals by a Particular Rider**
Shows all rental activity for a specific rider (input name or ID)
	`SELECT`
	`  rt.RentaLID,`
	`  rt.BikeID,`
	`  rt.RentalDaeTime,`
	`  rt.ReturnDateTime,`
	`  rt.StartStationID,`
	`  rt.EndStationID,`
	`  rt.Cost`
	`FROM Rental_Transaction rt`
	`JOIN Rider r ON rt.RiderID = r.RiderID`
	`WHERE r.FullName = 'Alice Morgan'`

5. **Bikes Currently at Each Station**
Shows how many bikes are currently available at each station
	`SELECT`
	`  s.StationName,`
	`  COUNY(b.BikeID) AS BikesAvailable`
	`FROM Station s`
	`LEFT JOIN Bike b ON s.StationID = b.CurrentStationID`
	`GROUP BY s.StationName`

7. **Bikes with Reported Issues**	`SELECT` `  rt.RentalID, rt.BikeIID`

Conclusion

The Eco_Ride database proves its operational capabilities through the set of SQL queries which demonstrate its management and analysis functions for bike rental service operations. The queries demonstrate the system's ability to track current bike availability and active bike usage and maintenance tracking and usage patterns, including popular bike selections. The system demonstrates how the database supports operational functions through its ability to display user rental history and station status and user-reported problems. The examples from the database show that the database system achieves efficient storage and delivers strong data retrieval capabilities, which improve organisational choices and service maintenance operations.

Task 7:

Validation and Testing

The evaluation of the Eco_Ride database system involved comparing its performance to the original functional requirements, which verified its ability to fulfil user requirements and maintain operational process integrity. The functional requirements were converted into test cases which ran through sample data and SQL queries for execution. The database design, along with its relationships and constraints, operates correctly according to the test results, which produce dependable and precise results.

Rider Registration Management

Requirement: Store rider name, contact details, ID numbers, and registration information.

Test: Inserted 5 sample rider records, checked for primary key uniqueness, and validated retrieval through queries.

Outcome: All rider data was stored correctly; duplicate Rider ID entries were rejected; querying riders returned accurate results.

Bike Recording and Tracking

Requirement: Maintain data on bike type, condition, location, and serial numbers.

Test: Populated 5 bike entries and performed sample queries such as “bikes currently available” and “bikes in maintenance.”

Outcome: Bikes linked correctly to stations; foreign key constraints prevented invalid entries; condition and location displayed correctly in queries.

Rental Transaction Logging

Requirement: Record rentals with dates, return times, duration, cost, and associated stations.

Test: Added realistic rental records, including ongoing rentals (NULL return times). Tested queries for current rentals, rider rental history, and most frequently used bikes.

Outcome: Rental transactions linked correctly to riders, bikes, and stations; ongoing rentals appeared as expected; cost and timestamps stored accurately.

Maintenance Tracking and Staff Assignment

Requirement: Track staff who serviced bikes and maintain maintenance history.

Test: Inserted maintenance records referencing bikes and staff. Ran test queries to list upcoming services and maintenance activity per staff member.

Outcome: Maintenance logs stored correctly, foreign keys enforced staff-bicycle dependencies, and service dates presented accurately.

Enforcement of Data Integrity

Requirement: Ensure strong validation using constraints (PK, FK, NOT NULL).

Test: Attempted invalid actions such as inserting maintenance for non-existing bike IDs or rentals without riders.

Outcome: Database correctly rejected invalid inserts; referential integrity fully enforced.

Query Functionality and Reporting

Requirement: Provide useful operational queries such as bikes out on rental, bikes due for service, and rentals by rider.

Test: Executed all seven queries in the Query Development section using sample data.

Outcome: All queries returned correct and meaningful results that reflected real operational scenarios.

Overall Conclusion:

Validation and testing confirm that the Eco_Ride database meets all functional requirements and performs reliably across all major operations. The database design includes complete normalisation along with appropriate constraints which maintain data integrity throughout the system. The test queries demonstrate that the system performs daily operations by tracking rentals and bike availability and maintenance records and creating service improvement reports. The database system has proven its ability to operate in real-world conditions while establishing a strong base for upcoming system developments.

Short IEEE References

The book "Fundamentals of Database Systems" was written by R. Elmasri and S. Navathe in 2016 and published by Pearson.

Database Systems: A Practical Approach to Design and Implementation by T. Connolly and C. Begg was published in 2015 by Pearson.

Maria Hernandez published Database Design for Mere Mortals through Addison-Wesley in 2013.

The textbook Database Systems: Design, Implementation, and Management by Paul Rob and Carlos Coronel became available in 2019 from Cengage.

In his 1970 article "A Relational Model of Data for Large Shared Data Banks" E. F. Codd discusses his relational data model in the journal Commun. ACM.