

Module Name: Introduction to Web and Database Concepts

Module Code: (ACCA4023)

**Assignment Title: Design and Implementation
Of a Relational Database System**

Student Name: Bonssa Abdulaziz

Student ID: 2510399

Submission Date: 18/07/2026

Table of Contents

1. Introduction.....	3
2. Scenario Analysis.....	3
3. Entity Relationship Diagram (ERD).....	4
4. Database Structure Development.....	6
5. Database Normalization.....	7
6. Database Implementation.....	8
7. Sample Data Population.....	10
8. Query Development.....	11
9. Validation and Testing.....	17
10. Design Reflection.....	18
11. Conclusion	19
12. References.....	20

1. Introduction

AutoPark Management System is a parking service provider for a city with different parking sites. The organisation is currently using a manual system, which records vehicle entry, parking spaces, customer details, and payment information, leading to inconsistencies, duplicate data, and inefficient parking management. This report aims to design and build a relational database system in the Microsoft Access database to enhance the ease of database operations and performance. The report will provide an understanding of designing and implementing a database through entity identification, normalisation and query construction. It also explains the importance of a central database for accurate parking data storage, retrieval, and management. The scenario analysis, database design, implementation, and testing are discussed in the following sections.

2. Scenario Analysis

2.1 Purpose and Context

The scenario is a parking service company that needs to replace the manual records. There are opportunities to have duplicate records, errors, and parking problems at multiple sites with manual processes. In this way, the database provides reliable data storage, efficient retrieval, and reliable reporting that can help with daily parking operations [1].

2.2 Identified Entities and Attributes

Based on an analysis of the scenario, the following five entities are identified:

- **Customer:** Store the information about the owner of the vehicle, such as customer ID, name, contact number, address etc.
- **Vehicle:** Holds the vehicle registration number, vehicle type and a reference to the owner of the vehicle.
- **ParkingSlot:** Slot is identified by a slot ID, a location for the site, a status of the slot (used or available), and a type of the slot.
- **Booking:** It saves booking ID, booking time, booking off time, booking reference to the customer, vehicle and slot.
- **Payment:** It collects payment ID, amount, payment date and a reference to the booking to which the payment is applied.

2.3 Relationships between Entities

One Customer may have several Vehicles, so the Customer and Vehicle are related in a one-to-many manner. A vehicle may also have several bookings over time, and each booking will use one parking slot in a parking site at a particular time [2]. Each booking, however, is associated

with exactly one payment record, which means that the relationship between Booking and Payment is one-to-one. Together, these relationships help to ensure that accurate log of vehicle movement and revenue is maintained for the different parking facilities.

3. Entity Relationship Diagram (ERD)

3.1 Purpose of the ERD

The Entity Relationship Diagram is a visual model that represents the data structure that is identified during scenario analysis. It identifies entities, attributes and relationships prior to any physical implementation, which helps to detect structural errors early.

3.2 Entities, Keys and Attributes

Entity	Primary Key	Foreign Key(s)	Attributes
Customer	CustomerID	—	CustomerName, ContactNumber, Address
Vehicle	VehicleID	CustomerID	RegistrationNumber, VehicleType
ParkingSlot	SlotID	—	SiteLocation, SlotStatus, SlotType
Booking	BookingID	VehicleID, SlotID	EntryTime, ExitTime
Payment	PaymentID	BookingID	Amount, PaymentDate

3.3 Relationship Cardinality

- **Customer to Vehicle:** one-to-many (a single customer may register more than one vehicle, but each vehicle is registered by only one customer.)
- **Vehicle to Booking:** one-to-many (one vehicle used across multiple bookings over time, each booking points to a particular vehicle.)
- **ParkingSlot to Booking:** one to many (one parking slot can have multiple bookings at different times, but each booking uses only one parking slot at a time.)
- **Booking to Payment:** one-to-one (Each booking creates one payment record; if a user parks one vehicle, their payment is for that one vehicle).

Entity Relationship Diagram (ERD) of the AutoPark Management System

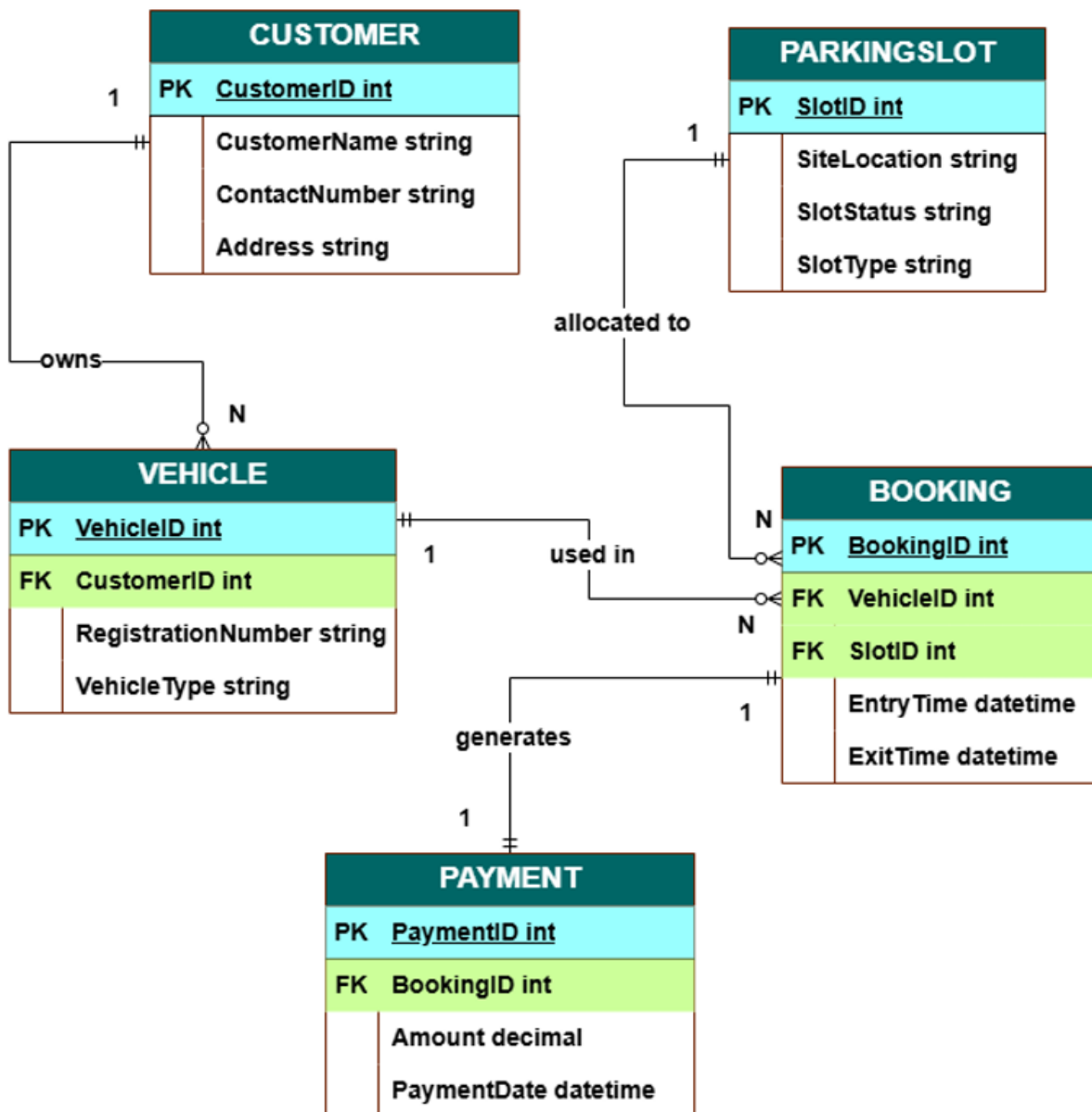


Figure 1: The ERD of AutoPark Management System

(Source: Self-created using draw.io)

It represents each entity as a box labeled with the entity name, and includes the attributes of that entity, including the primary keys. Cardinality is represented using connecting lines with crow's foot notation.

4. Database Structure Development

4.1 Overall Design Approach

Database Structure Development involves creating tables in Microsoft Access that reflect the ER model. Each entity is converted into a table and each attribute is converted to a field [3].

4.2 Field Selection and Data Types

Field Category	Example Fields	Data Type Used	Reason
Identifiers	CustomerID, BookingID	AutoNumber	Guarantees unique record identification
Text details	CustomerName, SiteLocation	Short Text	Stores alphanumeric data of limited length
Date and time	EntryTime, ExitTime, PaymentDate	Date/Time	Supports duration calculation and ordering
Financial values	Amount	Currency	Ensures accurate decimal representation

4.3 Relationships and Referential Integrity

- Foreign key fields are used to establish the relationship between tables based on ERD cardinality.
- Referential integrity is the feature that does not allow any child record to reference a non-existent parent record.
- There must be a corresponding vehicle and slot record for each booking record.

4.4 Constraints and Significance

- Required fields prevent incomplete entries.
- Unique indexes on primary keys allow no duplicate records.
- The date fields have validation rules to ensure that date entries are valid and that exit and entry times are not illogical.

It is a structured approach that is significant not only for accuracy. It affects the manual error and loss of data in the scenario directly and thus offers a sound foundation for queries and reports, as mentioned below in this report.

5. Database Normalization

5.1 Purpose of Normalization

Normalization reduces data redundancies and data dependencies in the database design. By normalizing the AutoPark Management System, each table will contain only the necessary data for its respective entity, avoiding duplication and the possibility of inconsistencies when updating the customer, vehicle, slot, booking, and payment records [4].

5.2 First Normal Form (1NF)

In this stage, each table has atomic values; that is, one record, or field, does not contain two or more values. As a result, each table also has a specific primary key that distinguishes each row.

- The VehicleType and RegistrationNumber are not combined, they are separate single-value fields
- Every table has a distinct identifier like CustomerID, VehicleID or BookingID
- There are no multiple data segments in a record.

5.3 Second Normal Form (2NF)

Partial dependency is eliminated in 2NF - when an attribute is dependent on only one part of a composite key. Each table has a single column primary key so attributes are completely dependent on the primary key.

- ContactNumber and Address are completely dependent on CustomerID.
- SlotStatus and SlotType are dependent on SlotID.
- There are no attributes that depend on only a portion of a key, as they should not have composite keys.

5.4 Third Normal Form (3NF)

Third Normal Form eliminates transitive dependency, in which a non-key attribute is dependent on a non-key attribute instead of the primary key.

- The Amount and PaymentDate is dependent on PaymentID, and not on BookingID indirectly.
- The EntryTime and ExitTime are directly dependent on BookingID
- No dependency on another non-key attribute, attributes are grouped in their own table.

5.5 Final Normalized Structure

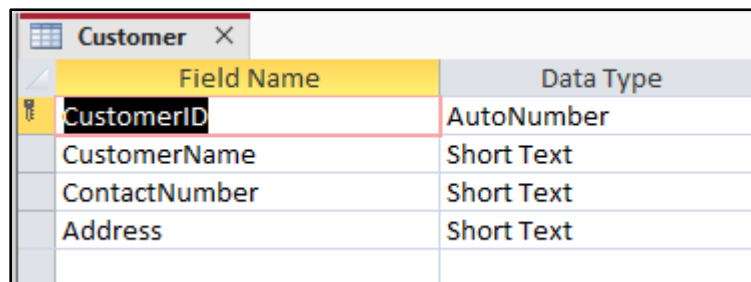
Normal Form	Requirement Met
1NF	Atomic values, unique primary key
2NF	No partial dependency
3NF	No transitive dependency

This helps to ensure data integrity and to produce reliable reports based on the data across all parking sites [5].

6. Database Implementation

6.1 Database and Table Creation

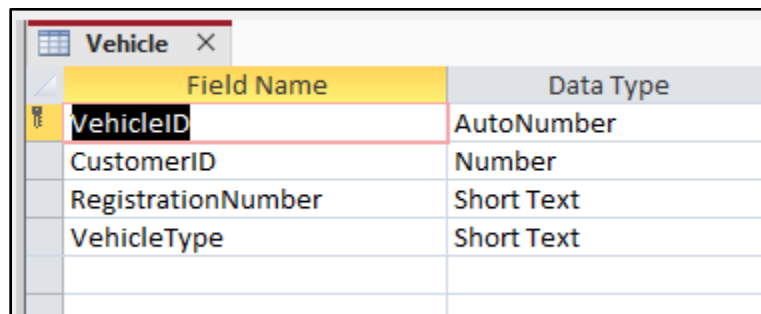
The AutoPark Management System is implemented in Microsoft Access, in which a new database file is created to store all five tables that were designed in the design phase. All tables are created one by one and each table field name and data type correspond to those defined in Section 4 [6]. AutoNumber fields are used as Primary Keys, Number, Short Text, Date/Time, and Currency data types are used based on the type of the field.



Field Name	Data Type
CustomerID	AutoNumber
CustomerName	Short Text
ContactNumber	Short Text
Address	Short Text

Figure 2: Customer Table Structure

(Source: Microsoft Access)



Field Name	Data Type
VehicleID	AutoNumber
CustomerID	Number
RegistrationNumber	Short Text
VehicleType	Short Text

Figure 3: Vehicle Table Structure

(Source: Microsoft Access)

Field Name	Data Type
SlotID	AutoNumber
SiteLocation	Short Text
SlotStatus	Short Text
SlotType	Short Text

Figure 4: ParkingSlot Table Structure

(Source: Microsoft Access)

Field Name	Data Type
BookingID	AutoNumber
VehicleID	Number
SlotID	Number
EntryTime	Date/Time
ExitTime	Date/Time

Figure 5: Booking Table Structure

(Source: Microsoft Access)

Field Name	Data Type
PaymentID	AutoNumber
BookingID	Number
Amount	Currency
PaymentDate	Date/Time

Figure 6: Payment Table Structure

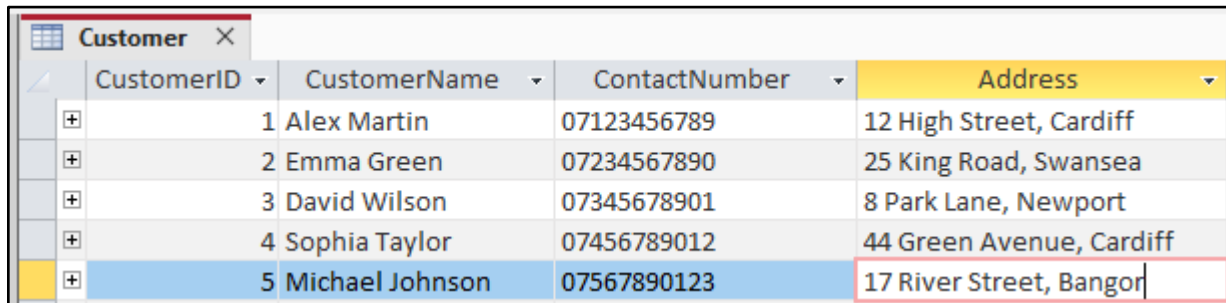
(Source: Microsoft Access)

6.2 Relationships Between Tables

After creating the tables, relationships are established between the tables on the basis of foreign key fields, linking Customer to Vehicle, Vehicle and ParkingSlot to Booking, and Booking to Payment. These links use referential integrity to keep there being no orphaned records within the system.

7. Sample Data Population

Sample data used in each table is realistic and not random and this allows for meaningful testing of the system [7].

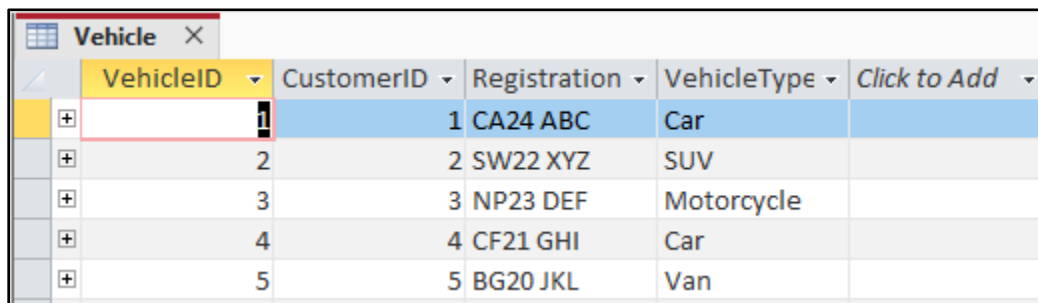


CustomerID	CustomerName	ContactNumber	Address
1	Alex Martin	07123456789	12 High Street, Cardiff
2	Emma Green	07234567890	25 King Road, Swansea
3	David Wilson	07345678901	8 Park Lane, Newport
4	Sophia Taylor	07456789012	44 Green Avenue, Cardiff
5	Michael Johnson	07567890123	17 River Street, Bangor

Figure 7: Customer Table Sample Data

(Source: Microsoft Access)

Represents 5 customers details each with a name, contact number, and address representing the owners of various vehicles.

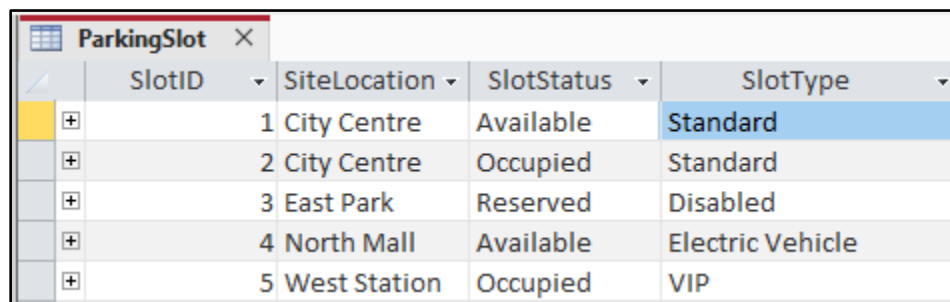


VehicleID	CustomerID	Registration	VehicleType	Click to Add
1	1	CA24 ABC	Car	
2	2	SW22 XYZ	SUV	
3	3	NP23 DEF	Motorcycle	
4	4	CF21 GHI	Car	
5	5	BG20 JKL	Van	

Figure 8: Vehicle Table Sample Data

(Source: Microsoft Access)

Presents 5 case histories of vehicles, each with a related customer record, with various vehicle numbers and types.



SlotID	SiteLocation	SlotStatus	SlotType
1	City Centre	Available	Standard
2	City Centre	Occupied	Standard
3	East Park	Reserved	Disabled
4	North Mall	Available	Electric Vehicle
5	West Station	Occupied	VIP

Figure 9: ParkingSlot Table Sample Data

(Source: Microsoft Access)

It lists five slot records from various locations with different slot statuses and slot types, such as standard, disabled, electric vehicle and VIP.

Booking					
	BookingID	VehicleID	SlotID	EntryTime	ExitTime
+	1	1	2	15/07/2026 08:30:00	15/07/2026 12:30:00
+	2	2	5	15/07/2026 09:00:00	15/07/2026 13:00:00
+	3	3	3	15/07/2026 10:15:00	15/07/2026 11:45:00
+	4	4	1	16/07/2026 08:00:00	16/07/2026 10:00:00
+	5	5	4	16/07/2026 09:30:00	16/07/2026 12:00:00

Figure 10: Booking Table Sample Data

(Source: Microsoft Access)

It shows five booking records which each connect a vehicle with a parking slot and entry/exit time, revealing that various vehicles are using different parking slots throughout the day.

Payment				
	PaymentID	BookingID	Amount	PaymentDate
+	1	1	£12.50	15/07/2026
+	2	2	£15.00	15/07/2026
+	3	3	£8.00	15/07/2026
+	4	4	£10.00	16/07/2026
+	5	5	£14.50	16/07/2026

Figure 11: Payment Table Sample Data

(Source: Microsoft Access)

This table holds 5 payment records, each one associated with a booking, to list the amounts and the payment dates of completed payments.

8. Query Development

Six queries are created in the system to showcase the use of stored data for practical monitoring and decision-making throughout this System [8].

Q1: Customer Vehicles

This query will return the name of the customers, contact number and registered vehicle information. It links to the Customer and Vehicle tables to show which vehicle is owned by which customer. As such, it helps to facilitate rapid identification of vehicle ownership during operations.

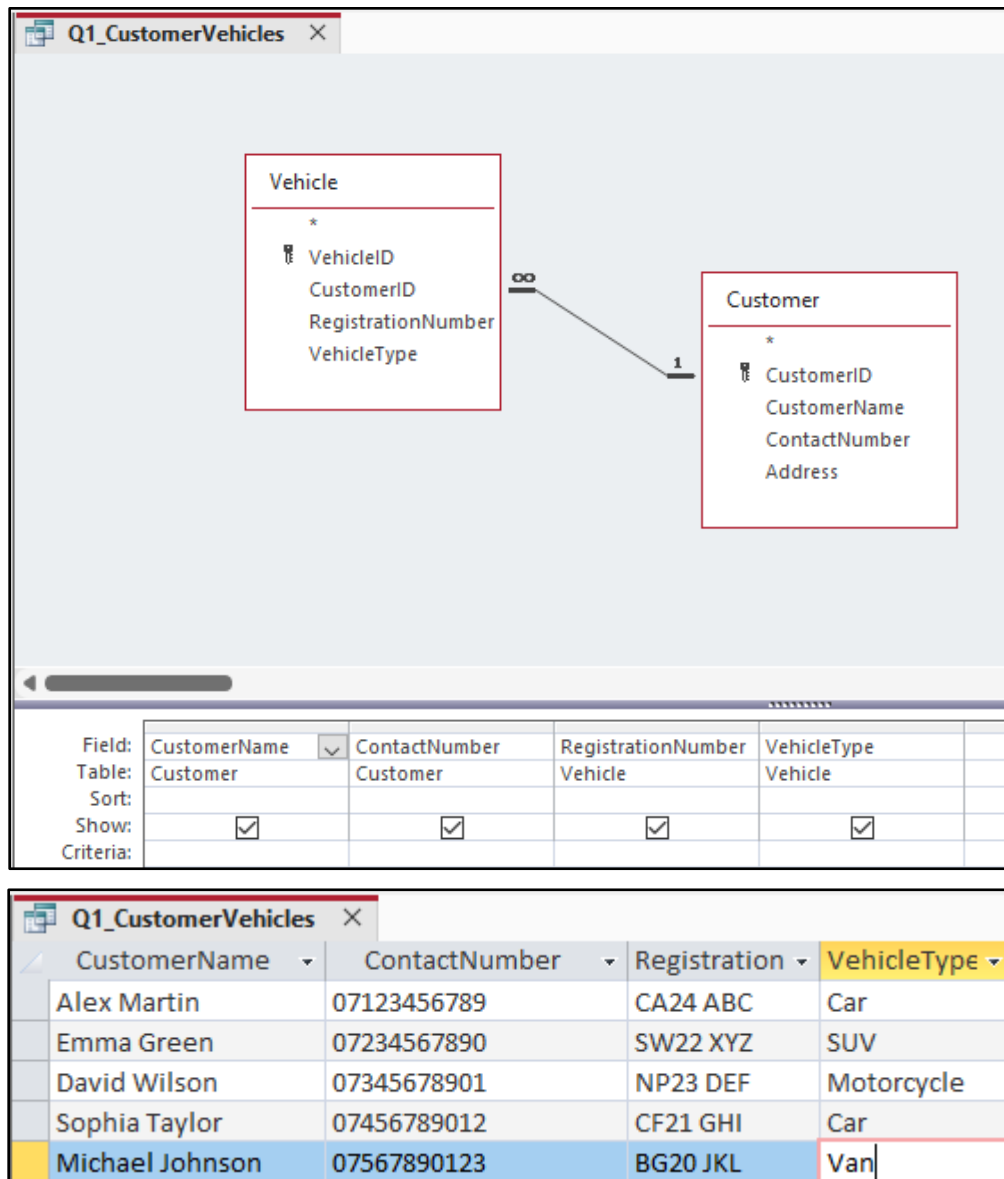


Figure 12: The result of Q1

(Source: Microsoft Access)

Q2: Current Bookings

This is a query that shows booking identifiers and vehicle registration numbers, entry and exit times. Shows active or completed parking sessions by aggregating Vehicle and Booking data. Hence, management can keep track of the movement of vehicles throughout the system.

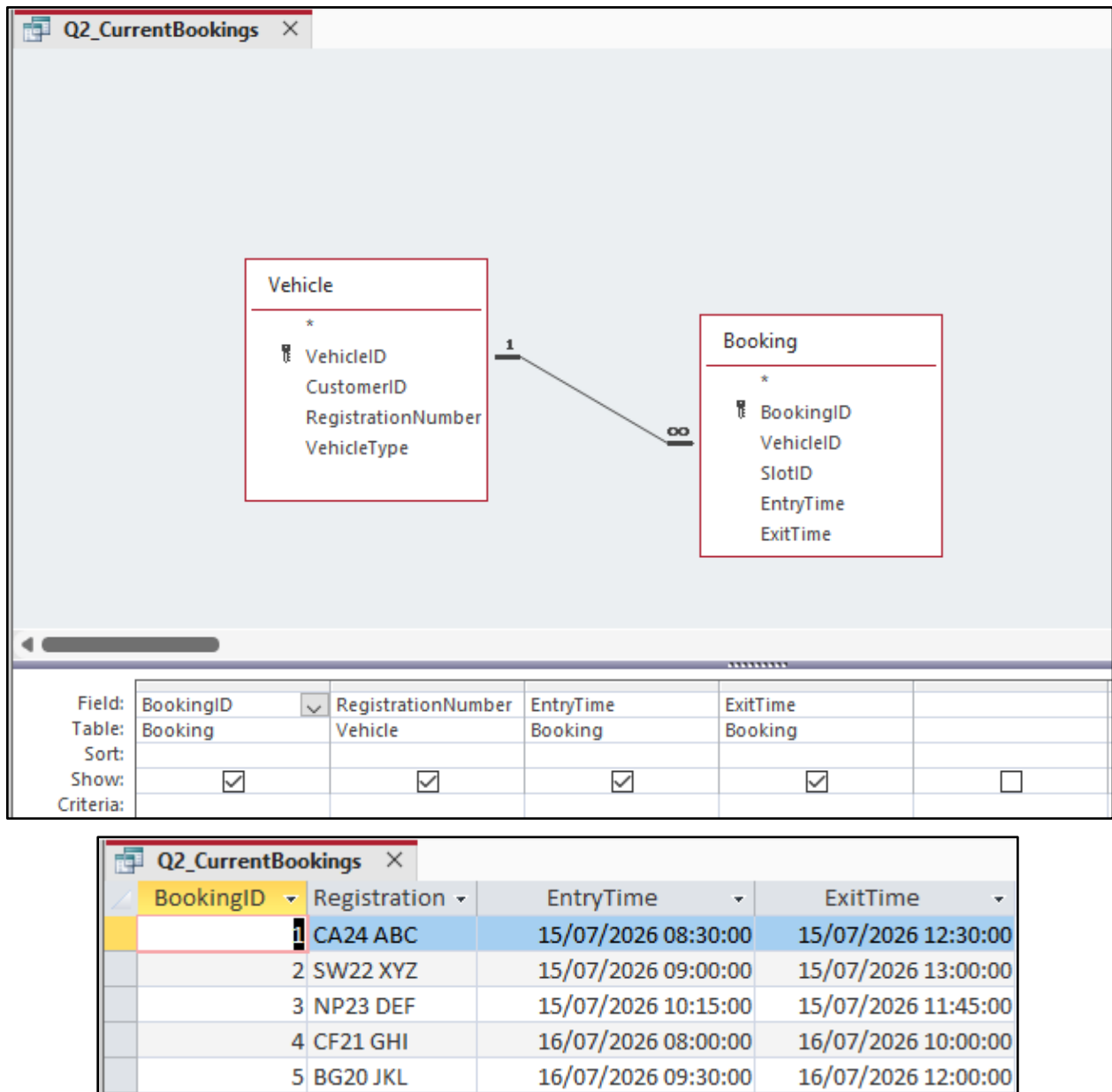


Figure 13: The result of Q2

(Source: Microsoft Access)

Q3: Parking Slot Allocation

This query connects the tables Vehicle, Booking, and ParkingSlot and returns the vehicle that is parked in each of the parking slots as well as the type of parking slot and the status. It offers information on the slots used on various websites, which can help with decisions about space administration.

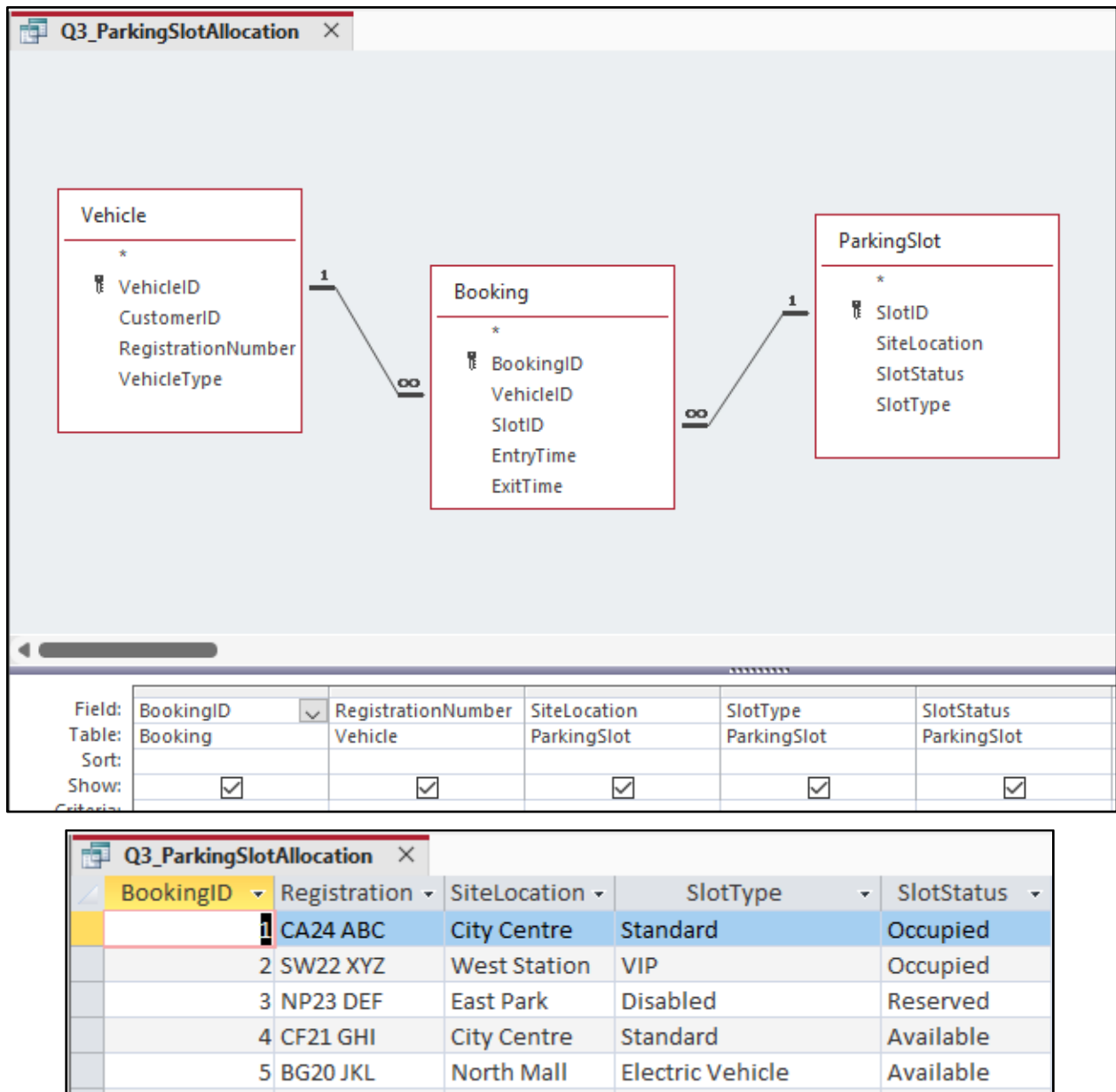


Figure 14: The result of Q3

(Source: Microsoft Access)

Q4: Payments Above Ten

This query will only be applied to the payment record whose amount is more than 10 pounds. It fetches payment identifiers, payment booking references, and payment dates that match this condition to enable financial monitoring of higher value transactions.

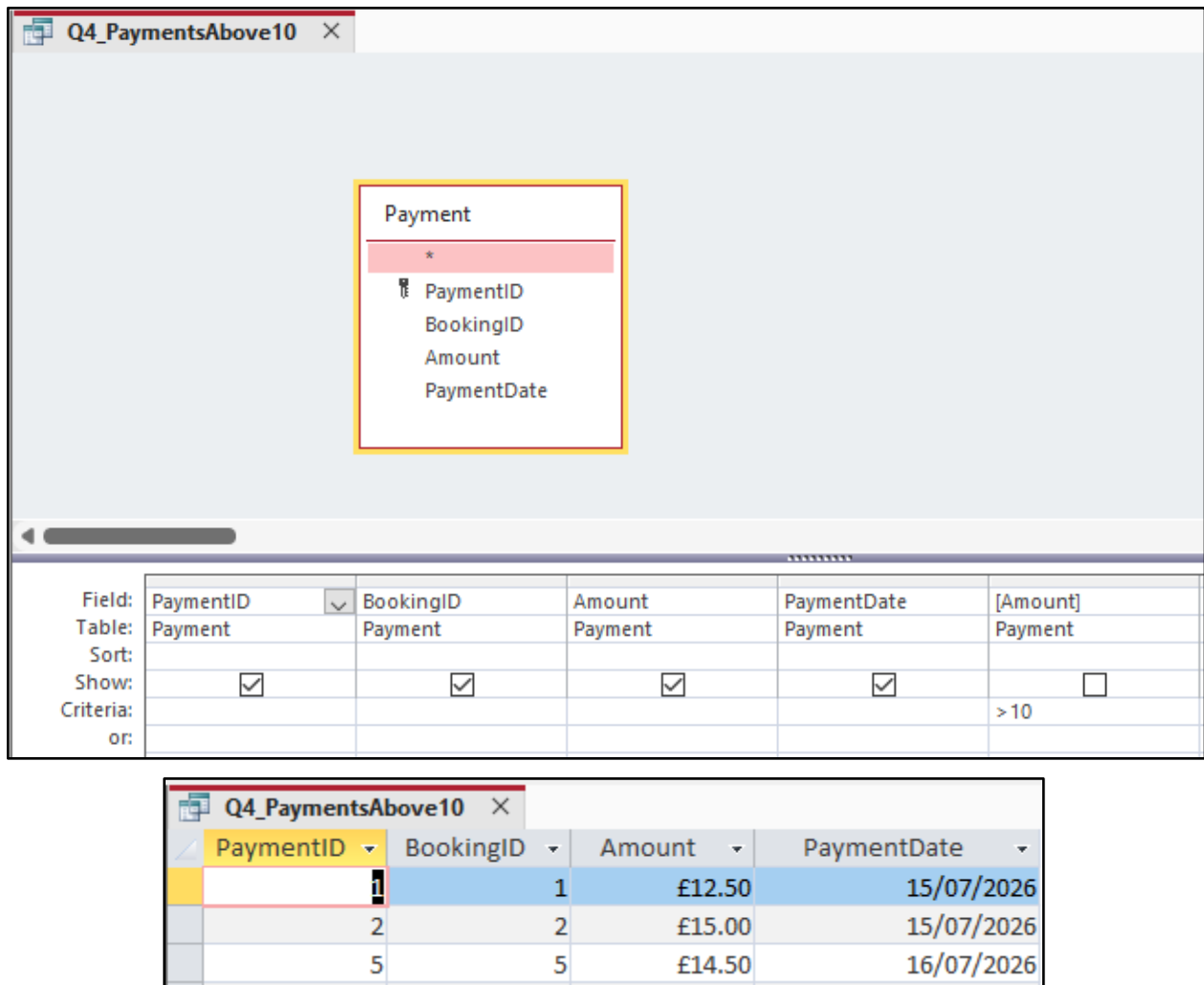


Figure 15: The result of Q4

(Source: Microsoft Access)

Q5: Daily Revenue

This query sorts payments by date and sums up the revenue per day. It shows the financial reporting capability of the system and how it can keep track of the income trends over time from management's perspective.

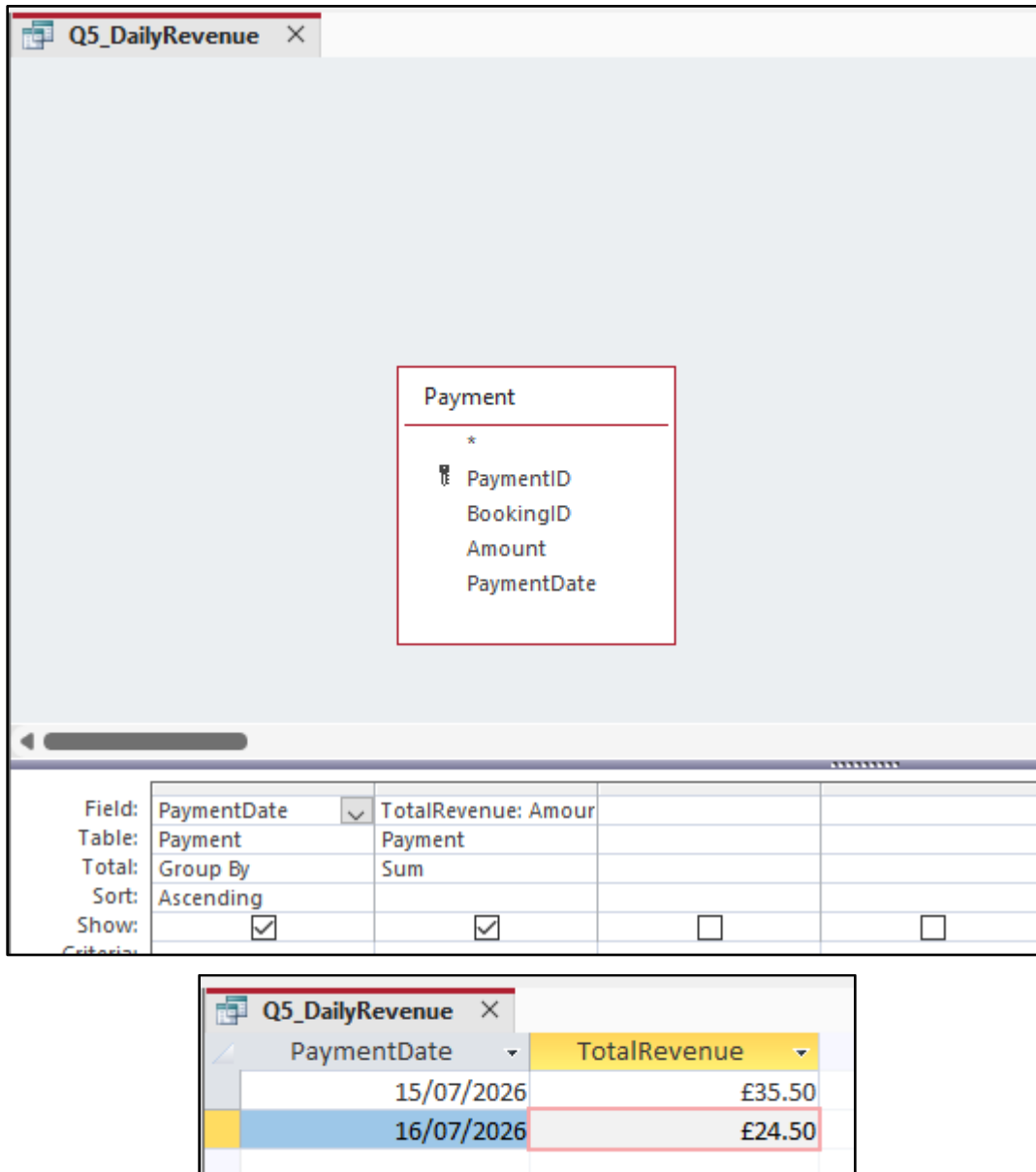


Figure 16: The result of Q5

(Source: Microsoft Access)

Q6: City Centre Bookings

This query will show bookings on City Centre site including Registration Numbers, Entry Time and Exit Time. It provides the ability to monitor parking activity and parking pattern on the site on a specific basis.

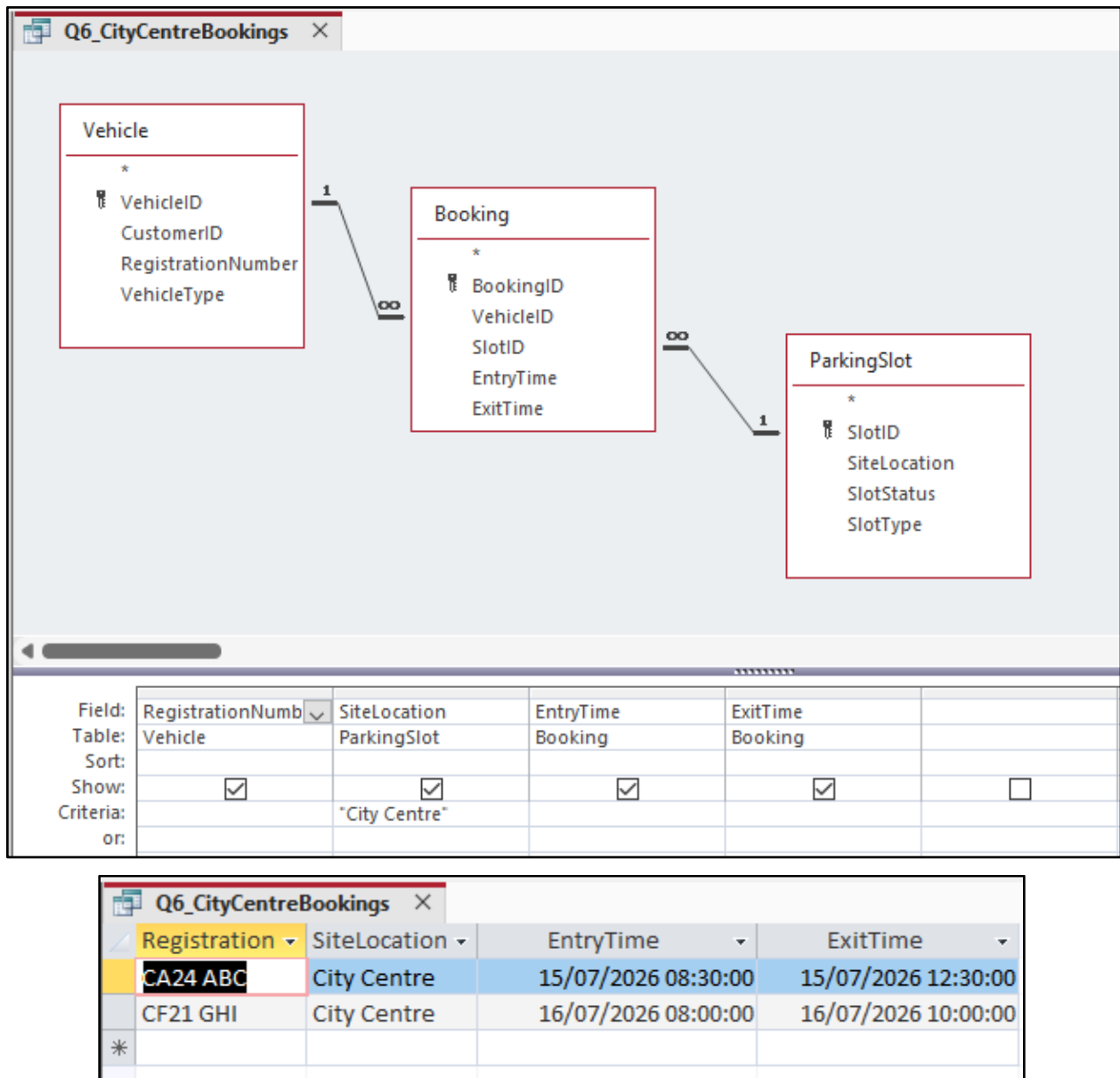


Figure 17: The result of Q6

(Source: Microsoft Access)

9. Validation and Testing

The validation is to ensure that the AutoPark Management System database is operating correctly based on the scenario. The correctness and behaviour of all tables and queries are checked for relationships.

Test	Expected Outcome	Actual Outcome	Result
------	------------------	----------------	--------

Insert customer and linked vehicle record	Vehicle correctly linked to CustomerID	Record linked correctly	Pass
Run Q1 (Customer and Vehicle join)	Displays owner alongside vehicle details	Correct records displayed	Pass
Run Q3 (Parking Slot Allocation)	Displays correct slot, type, and status	Correct data returned	Pass
Run Q4 (Payments Above Ten)	Displays only payments over £10	Correct filtered results	Pass
Run Q5 (Daily Revenue)	Displays total revenue per date	Correct totals calculated	Pass
Run Q6 (City Centre Bookings)	Displays only City Centre records	Correct filtered results	Pass

10. Design Reflection

10.1 Design Decisions

The database has a relational structure, meaning that it is made up of a series of tables containing information about customers, vehicles, parking spaces, bookings and payments. Consistency is maintained and duplication of data is avoided with the implementation of normalization, and the tables are linked by primary and foreign keys. The design is structured for data storage and retrieval, ensuring that the data is stored and retrieved accurately.

10.2 Challenges Encountered

The primary issue in building a database is to make sure that related tables are properly linked together and that referential integrity is maintained. The second is to design queries properly to return correct data from multiple related tables and to use correct data types. These problems will be solved with proper designing and testing of the database.

10.3 Future Improvements

There are many features that can be added in the future, such as online parking reservations and real-time parking space availability, as well as the automatic payment feature. Such enhanced options can improve user parking convenience.

11. Conclusion

The AutoPark Management System database is correctly structured as a relational database that holds customers' information, vehicle information, parking information, booking information, and payment information. The design is completed, it is normalized, designed, implemented, queried and tested, and it achieves the project objectives and helps manage parking efficiently.

12. References

- [1] Acharya, K., 2024. Automobile Management System Project Report. Authorea Preprints.
- [2] Babirye, G., Alayo, R., Kalanzi, M., Atim, D. and Ecuman, P., 2022. Parking management system (Doctoral dissertation, Makerere University).
- [3] Ahammed, F.A.Y.S.A.L., 2025. Design and implementation of parking management system (PMS). The Science Post, 1(1).
- [4] Kaplanović, N.D., 2024. Information system to support the operation of the car park. Vojnotehnički glasnik/Military Technical Courier, 72(1), pp.388-407.
- [5] Hardini, M., Agarwal, V., Apriani, D., Widjaya, I.A., Setiawaty, E. and Nurasiah, N., 2025. Application of database normalization in increasing data storage efficiency. International Transactions on Artificial Intelligence, 3(2), pp.201-211.
- [6] Simon, M., 2023. Getting Started with SQL and Databases. Cham, Switzerland: Springer.
- [7] DeBarros, A., 2022. Practical SQL: A Beginner's Guide to Storytelling with Data. no starch Press.
- [8] Khudair, H.H., 2024. Employment the database of software (Microsoft Access) in implementing and printing official documents. Eximia, 13, pp.87-98.