



THE IMPLEMENTATION OF AN MS ACCESS DATABASE

AutoPark Management System



21 JULY 2026

Lecturer: Suresh Shanmugasundaram

Student Name: Hamed Huffi

Student Number: 2528907

Group 1

Table of Contents

Task 1: Describe the main entities of the scenario	3
Task 2: Entity Relationship Diagram (ERD)	3
Figure 1: Entity Relationship Diagram.....	3
Figure 2: Cardinality between the Carparks and the Parking sessions tables.	4
Figure 3: Cardinality between the Parking sessions and the Payments tables.....	5
Figure 4: Cardinality between the Season-Pass-Holders and the Parking Sessions tables.....	5
Figure 5: Cardinality between the Customers and the Parking sessions tables.	5
Figure 6: Cardinality between the Customers and the Payments tables.	5
Figure 7: Cardinality between the Staff and the Carparks tables.....	6
Task 3: Data Structure	6
Figures 8&9: Carparks Table	6
Figures 10&11: Parking Sessions Table.....	7
Figures 12&13: Payments Table	8
Figures 14&15: Season Pass Holders Table.....	9
Figures 16&17: Staff Users Table	9
Task 4: Normalisation Compliance	10
Figure 18: Unnormalized Data (Source MS Excel)	11
Figure 19: First Normal Form (1NF) (Source MS Excel)	12
Figures 20,21,22: Second Normal Form (2NF) (Source MS Excel)	12
Figures 23,24,25: Third Normal Form (3NF) (Source MS Excel)	14
Task 5: Population of Data in the Database	16
Figure 26: Population of Data (Source MS Access)	17
Task 6: Query Development.....	17
Figures 27&28: Daily Revenue Per Carpark Query (Source MS Access)	18
Figures 29&30: Most Frequent Customers Query (Source MS Access).....	19
Figures 31&32: Staff Per Carpark Query (Source MS Access)	20
Figures 33&34: Season-Pass-Holders Duration Query (Source MS Access).....	21
Task 7: Validation And Testing	21
Figures 35&36: Validation and Testing for Entry Time	22

Figures 37&38: Validation and Testing for Capacity Limit 23

Figures 39&40 Validation and Testing for Customer Contact Number 24

Sources: 24

Task 1: Describe the main entities of the scenario.

AutoPark Management Systems currently records data manually, resulting in errors, data loss, and the wasteful use of parking spaces. That's why the company wants to fully automate it.

Entities: I recognised six main entities that are the core tables of my relational database for Autopark Management system.

- 1) The first entity is the Car-Parks table, which contains all the information about the 5 facilities run by the AutoPark management system and connects with the CarParkID (Primary Key) to the Parking-Sessions table and the Staff table, who manage them.
- 2) Next entity I made is the Parking Sessions table, which logs the usage of the facilities. (Primary Key is Session-ID), which establishes the relationships among three tables: the Car-Park-table, the Customer table, and the Season-Pass-Holders table.
- 3) Third is the Season-Pass-Holders table, which is only for clients wanting to purchase a season pass instead of paying as they park each time (Season-Pass-ID is the Primary key). It is connected to the Parking-Sessions table.
- 4) Fourth is the **Payments table**, which records the financial entries and the methods used (Payment-ID is the Primary Key) that links to the Parking-Sessions and Customer tables.
- 5) The second last is the **Customer table, which has the essential details of potential clients (Customer-ID is the Primary Key)**. It is linked to the payments and parking sessions tables.
- 6) The last table I created is the Staff-User table, where all the staff details are documented. The relationship is formed with the Car-Parks table through the Foreign Key Car-Park-ID

Additional note: (If you are an in-house MS developer, you would be an entity that comes under the staff category. However, in this task, I have chosen to serve as an external MS developer, creating a sturdy database system to fully automate the current manual system.)

Task 2: Entity Relationship Diagram (ERD)

Here is a screenshot showing what the database will look like in an ERD form and how the entity's primary and foreign keys connect between them to form relationships with cardinality.

Figure 1: Entity Relationship Diagram

(Source Lucid charts): https://lucid.app/lucidchart/04171359-cf43-41c9-ac32-6ac096aec54b/edit?invitationId=inv_c620a475-1b66-462f-b124-642ff6bc2a2f&page=0_0#

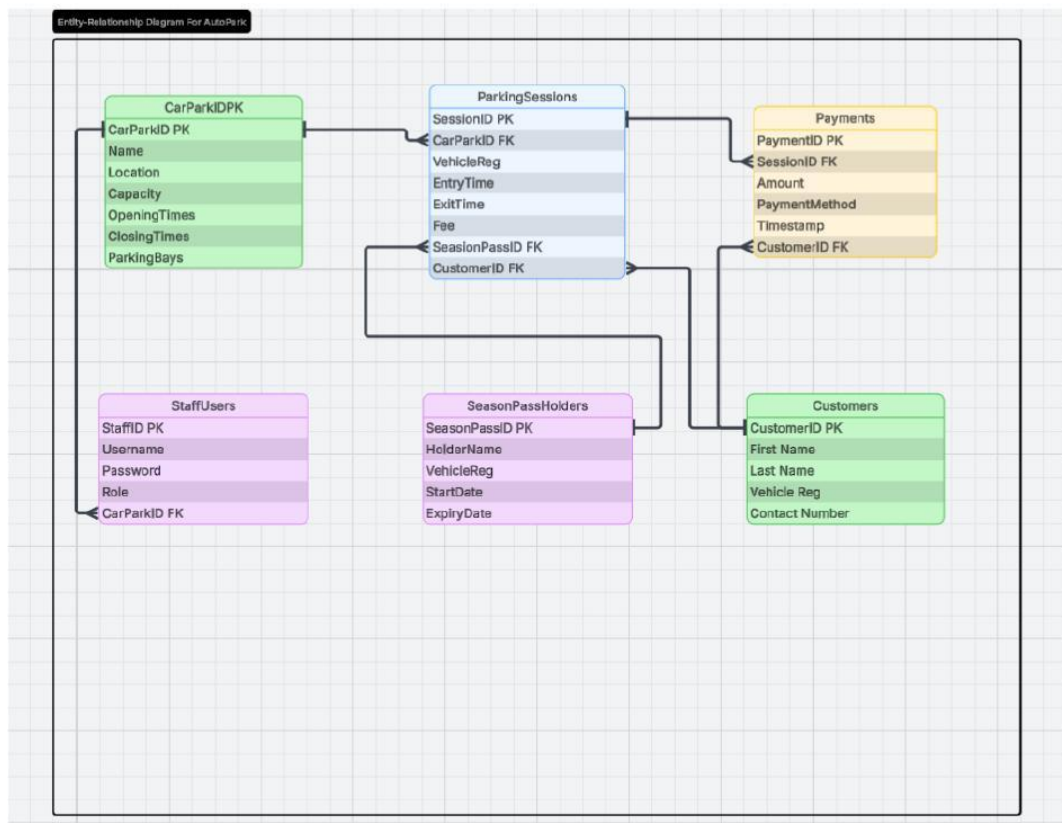


Figure 2: Cardinality between the Carparks and the Parking sessions tables. (One-To-Many) (Source MS Access)

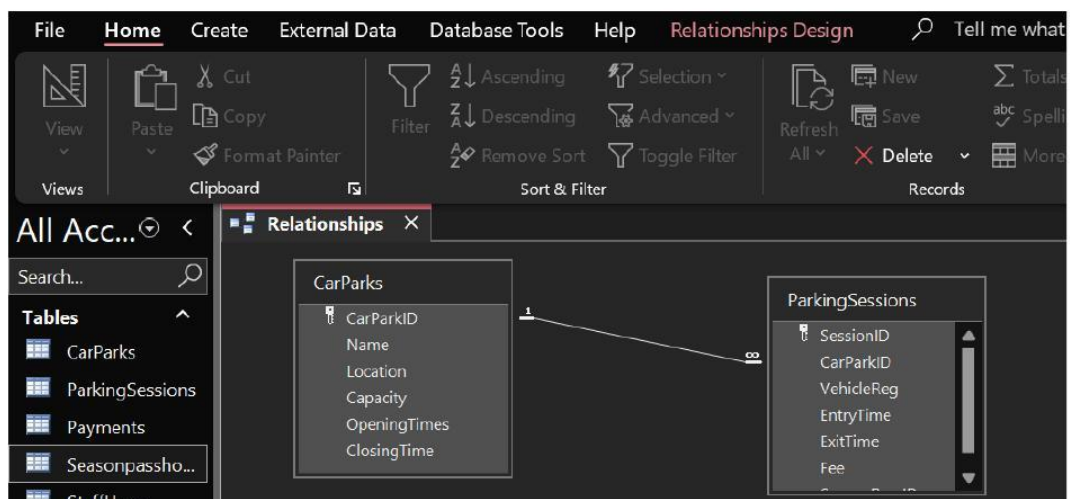


Figure 3: Cardinality between the Parking sessions and the Payments tables. (One-To-Many) (Source MS Access)

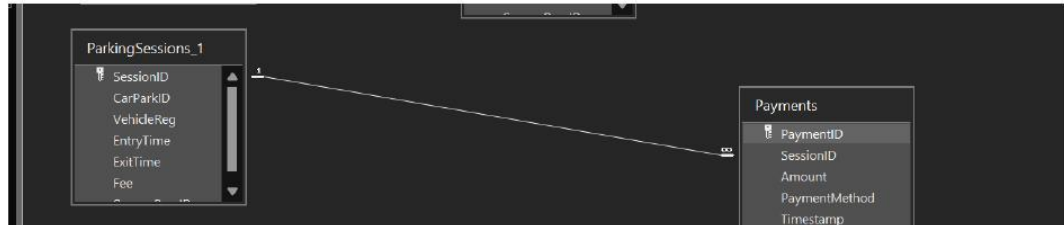


Figure 4: Cardinality between the Season-Pass-Holders and the Parking Sessions tables. (One-To-Many) (Source MS Access)

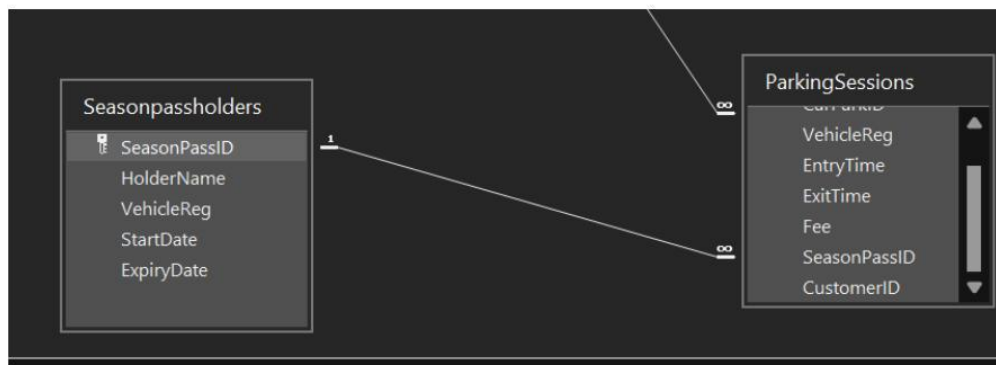


Figure 5: Cardinality between the Customers and the Parking sessions tables. (One-To-Many) (Source MS Access)

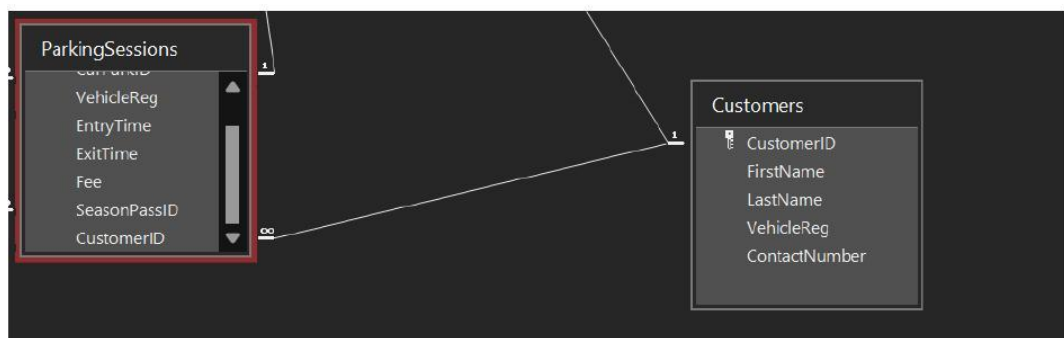


Figure 6: Cardinality between the Customers and the Payments tables. (One-To-Many) (Source MS Access)

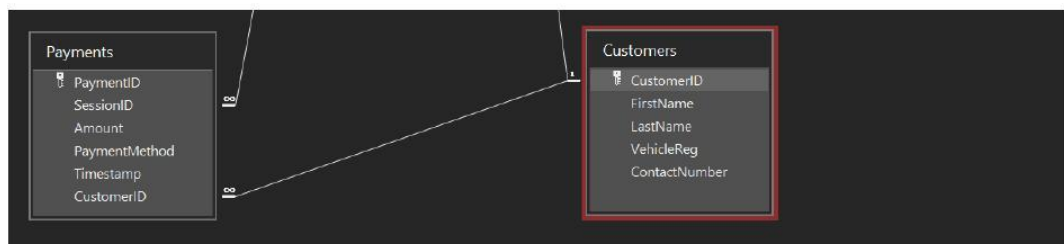
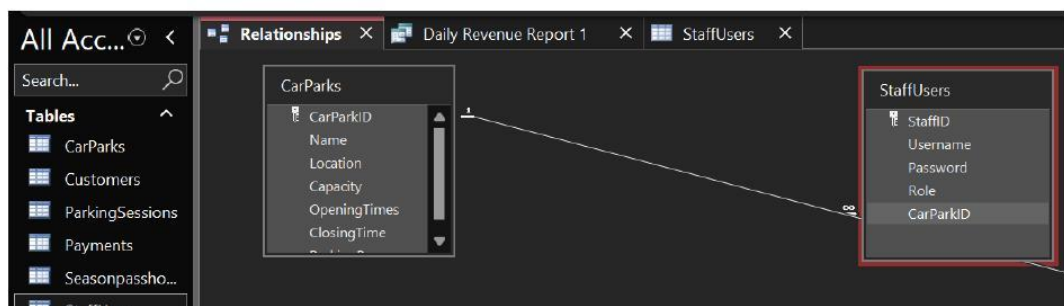


Figure 7: Cardinality between the Staff and the Carparks tables.
(One-To-Many) (Source MS Access)



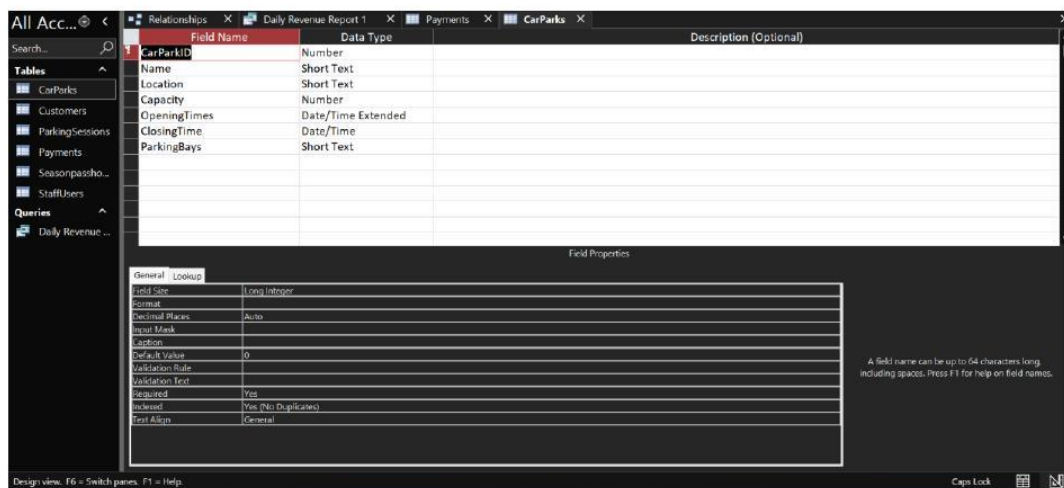
Task 3: Data Structure

The new database blueprint I built will look like this in data and design views, labelling each entity, its attributes, and sub-categories, and showing their data types, field properties, and the inserted data.

Figures 8&9: Carparks Table (Source MS Access) Design and Data view.

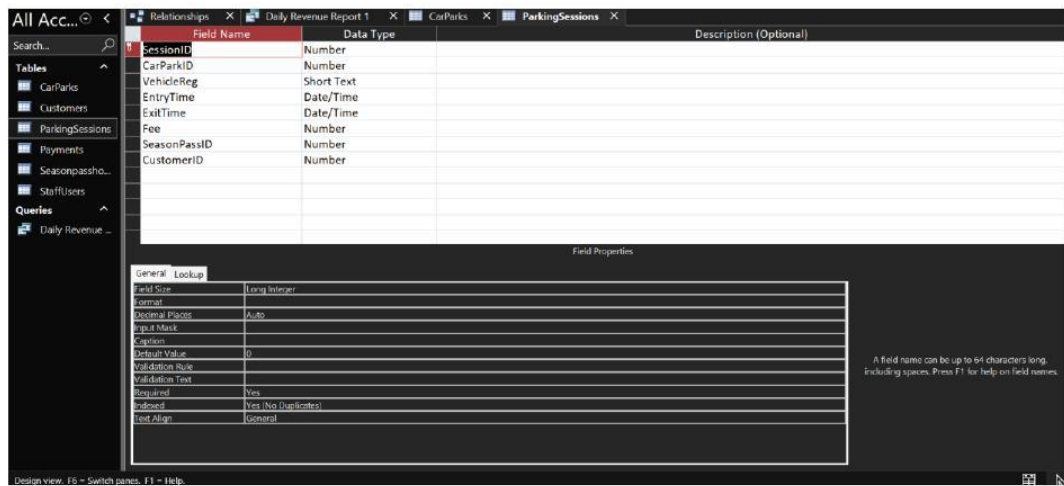
The screenshot shows the Microsoft Access Data view of the **CarParks** table. The table has the following columns: CarParkID, Name, Location, Capacity, OpeningTime, ClosingTime, ParkingBays, and Click to Add. The data is as follows:

CarParkID	Name	Location	Capacity	OpeningTime	ClosingTime	ParkingBays	Click to Add
1	NorthLondon	Edmonton	40	06:00:00	12:00:00	N1-N40	
2	WestLondon	WestbournePe	40	06:00:00	12:00:00	W1-W40	
3	EastLondon	CanaryWharf	40	06:00:00	12:00:00	E1-E40	
4	SouthLodon	Victoria	40	06:00:00	12:00:00	S1-S40	
5	CentraLondor	Soho	40	06:00:00	12:00:00	C1-C40	
0			0				



Figures 10&11: Parking Sessions Table (Source MS Access) Design and Data view.

SessionID	CarParkID	VehicleReg	EntryTime	ExitTime	Fee	SeasonPassID	CustomerID	Click to Add
11	1	LK75DNF	12:00:00	03:00:00	15	112	101	
12	2	AS24TGF	09:00:00	19:00:00	50	123	201	
13	3	WQ74KJL	14:00:00	11:00:00	45	134	301	
14	4	TR25SDD	07:00:00	15:00:00	40	145	401	
15	5	UW26TSD	10:00:00	13:00:00	25	156	501	
0	0				0	0	0	



Figures 12&13: Payments Table (Source MS Access) Design and Data view.

PaymentID	SessionID	Amount	PaymentMethod	Timestamp	CustomerID	Click to Add
1	14	40	BACS	15:00:00	401	
2	12	50	TicketMachine	19:00:00	201	
3	11	15	BACS	03:00:00	101	
4	13	45	BACS	11:00:00	301	
5	15	25	TicketMachine	13:00:00	501	
0	0	0			0	

Field Name	Data Type	Description (Optional)
PaymentID	Number	
SessionID	Number	
Amount	Number	
PaymentMethod	Short Text	
Timestamp	Date/Time	
CustomerID	Number	

Field Properties	
General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

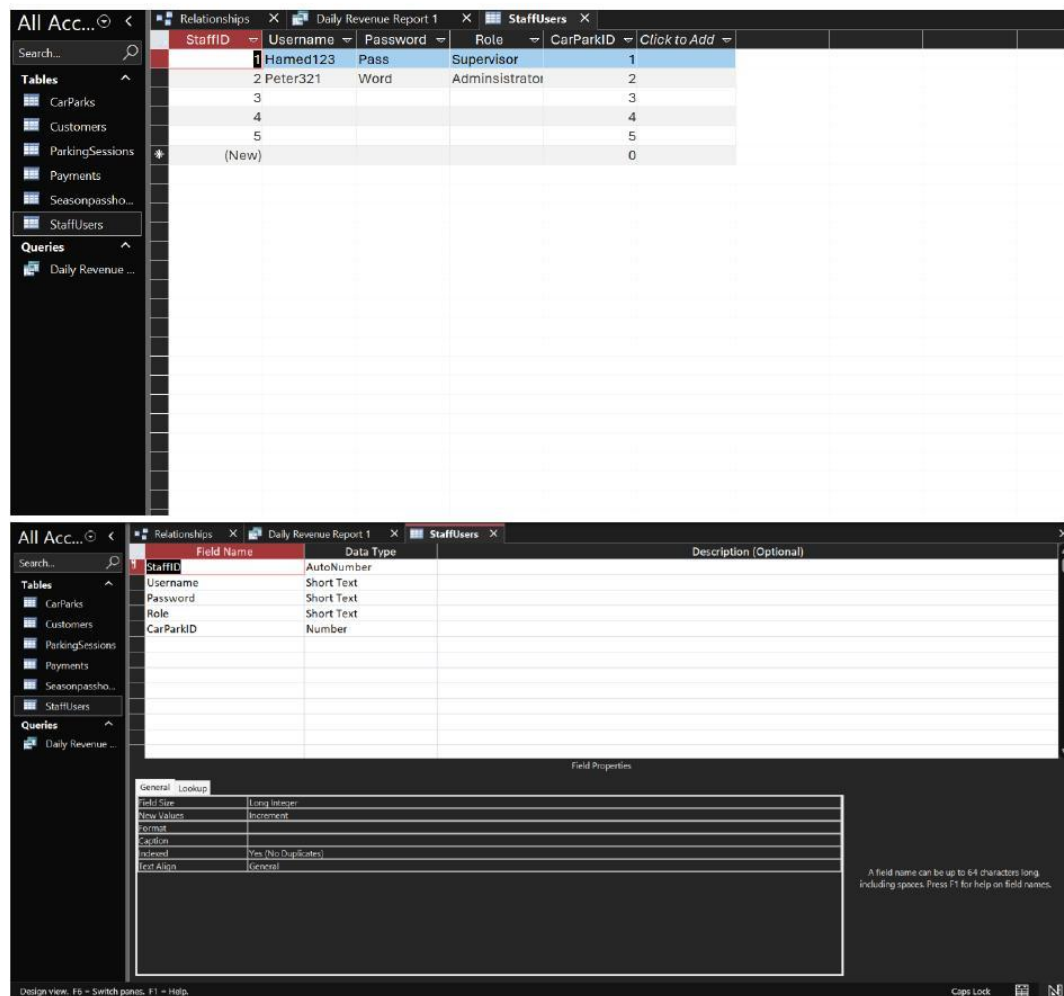
Figures 14&15: Season Pass Holders Table (Source MS Access) Design and Data view.

SeasonPassID	HolderName	VehicleReg	StartDate	ExpiryDate
112	Jake	LK75DNF	12/03/2026	12/06/2026
123	Mo	AS24TGF	17/06/2026	23/10/2026
134	Ahmed	WQ74KJL	20/04/2026	30/09/2026
145	Hana	TR25SDD	01/04/2026	01/08/2026
156	Hamed	UW26TSD	02/02/2026	17/07/2026
0				

Field Name	Data Type	Description (Optional)
SeasonPassID	Number	
HolderName	Short Text	
VehicleReg	Short Text	
StartDate	Date/Time	
ExpiryDate	Date/Time	

Field Properties	
General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Text Align	General

Figures 16&17: Staff Users Table (Source MS Access) Design and Data view.



Task 4: Normalisation Compliance

1) Unnormalized Data: Issue Repeated Groups and more than one value in fields. (slot numbers have multiple values, not atomic)

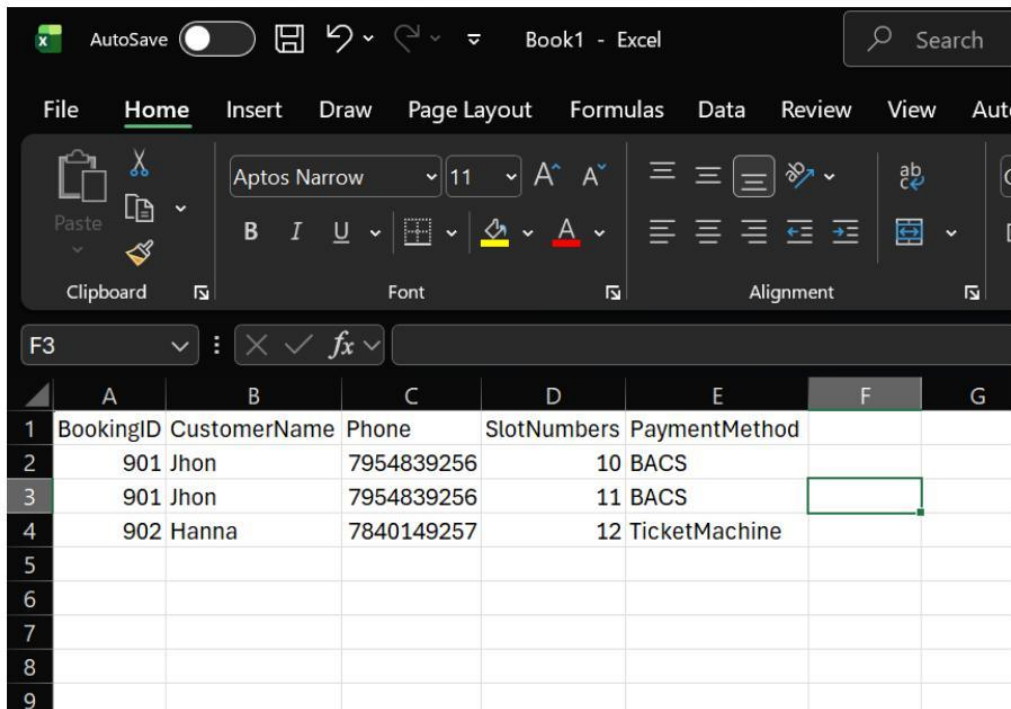
Figure 18: Unnormalized Data (Source MS Excel)

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	BookingID	CustomerName	Phone	SlotNumbers	PaymentMethod	
2	901	Jhon	7954839256	10,11	BACS	
3	902	Hanna	7840149257	12	TicketMachine	
4						
5						
6						
7						
8						
9						

2) First Normal Form (1NF): Each slot becomes its own row (this is now atomic and in First Normal Form)

Figure 19: First Normal Form (1NF) (Source MS Excel)



	A	B	C	D	E	F	G
1	BookingID	CustomerName	Phone	SlotNumbers	PaymentMethod		
2	901	Jhon	7954839256	10	BACS		
3	901	Jhon	7954839256	11	BACS		
4	902	Hanna	7840149257	12	TicketMachine		
5							
6							
7							
8							
9							

3) Second Normal Form (2NF): Splitting the data into 2 tables, now every attribute depends on the Primary Key.

Figures 20,21,22: Second Normal Form (2NF) (Source MS Excel)

(Bookings-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review View

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

D11

	A	B	C	D	E
1	BookingID	CustomerName	Phone	PaymentMethod	
2	901	Jhon	7954839256	BACS	
3	902	Hanna	7840149257	TicketMachine	
4					
5					
6					
7					
8					

(Booking-Slots-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

C5

	A	B	C	D	
1	BookingID	SlotNumbers			
2	901	10			
3	901	11			
4	902	12			
5					
6					
7					
8					
9					
10					
11					

(Payment-Method-Table)

	A	B	C
1	PaymentMethod	Amount	
2	BACS	£30	
3	TicketMachine	£10	
4			
5			
6			
7			
8			
9			

Before transforming into 3NF, the issue is that the amount depends on the Payment Method rather than directly on the Booking ID.

4) Third Normal Form (3NF): So, to convert into 3NF, you would split the tables again.

Figures 23,24,25: Third Normal Form (3NF) (Source MS Excel)

(Bookings-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review View

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

D11

	A	B	C	D	E
1	BookingID	CustomerName	Phone	PaymentMethod	
2	901	Jhon	7954839256	BACS	
3	902	Hanna	7840149257	TicketMachine	
4					
5					
6					
7					
8					

(Booking-Slots-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles Styles

C5

	A	B	C	D	E
1	BookingID	SlotNumbers			
2	901	10			
3	901	11			
4	902	12			
5					
6					
7					
8					
9					
10					
11					

(Payment-Methods-Table)

	A	B	C
1	PaymentMethod	Amount	
2	BACS	£30	
3	TicketMachine	£10	
4			
5			
6			
7			
8			
9			

Now, there are no repeated groups, no partial dependencies (Composite Key), and no transitive dependencies (directly depend on Primary Key), which meet the requirements for Third Normal Form (3NF).

Task 5: Population of Data in the Database

I populated the database with 5 clients, detailing their first and last names, their unique IDs, Vehicle-Reg, and contact numbers.

Figure 26: Population of Data (Source MS Access)

CustomerID	FirstName	LastName	VehicleReg	ContactNur	Click to Add
101	Jake	Harrow	LK75DNF	756763927	
201	Mo	Khalifa	AS24TGF	737120973	
301	Ahmed	Elbadawi	WQ74KJL	796556773	
401	Hanna	Rahim	TR25SDD	738276432	
501	Hamed	Mohammed	UW26TSD	787629321	
0				0	

Task 6: Query Development

Step-by-step sequence: 1. Open Query Design 2. Add Tables 3. Form Relationships if not created already 4. Select the relevant Fields 5. Then ensure all is sorted in your accurate preference 6. Finally, click Run to see your outcome.

1) Daily Revenue Per Carpark (Tables: Payments, Parking-Sessions, CarParks) (Design and Data View).

Figures 27&28: Daily Revenue Per Carpark Query (Source MS Access)

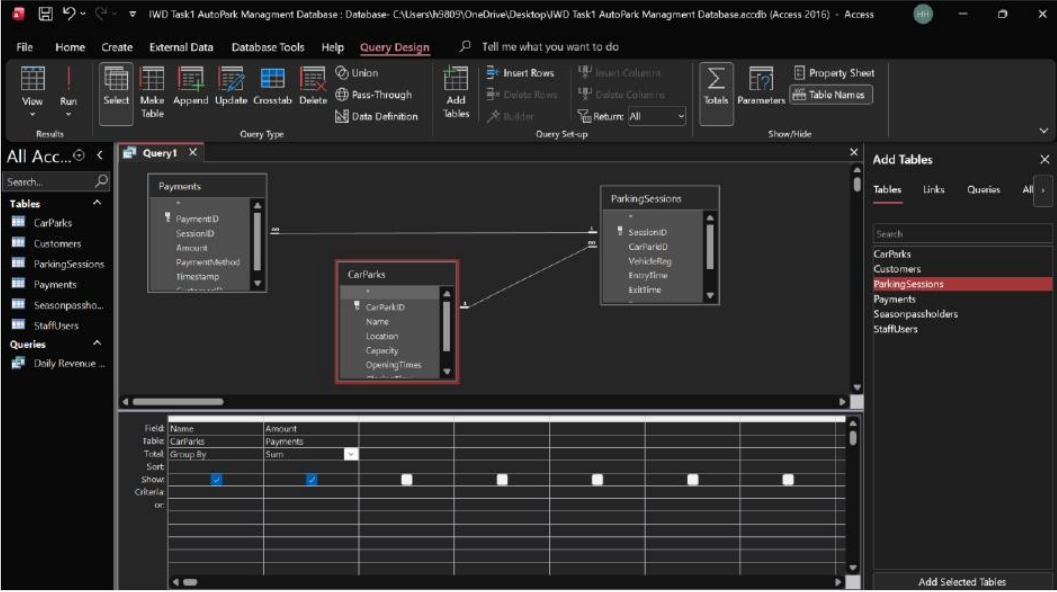
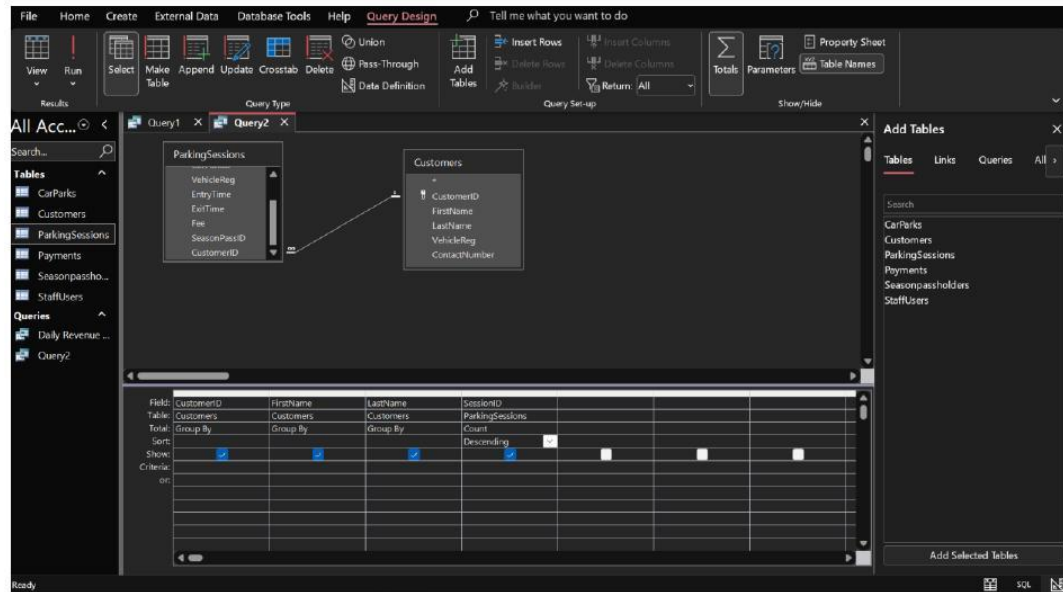


Figure 28 shows the Microsoft Access Query View for 'Query1'. The query results are displayed in a table with columns 'Name' and 'SumOfAmou'. The results show the daily revenue for five car parks: CentralLondon (25), EastLondon (45), NorthLondon (15), SouthLodon (40), and WestLondon (50).

Name	SumOfAmou
CentralLondon	25
EastLondon	45
NorthLondon	15
SouthLodon	40
WestLondon	50

2) Most frequent customers (Tables: Parking-Sessions, Customers) (design and data view).

Figures 29&30: Most Frequent Customers Query (Source MS Access)



CustomerID	FirstName	LastName	CountOfSess
501	Hamed	Mohammed	1
401	Hanna	Rahim	1
301	Ahmed	Elbadawi	1
201	Mo	Khalifa	1
101	Jake	Harrow	1

3) Staff Per Carpark (Tables: Staff, CarParks) (design and data view).

Figures 31&32: Staff Per Carpark Query (Source MS Access)

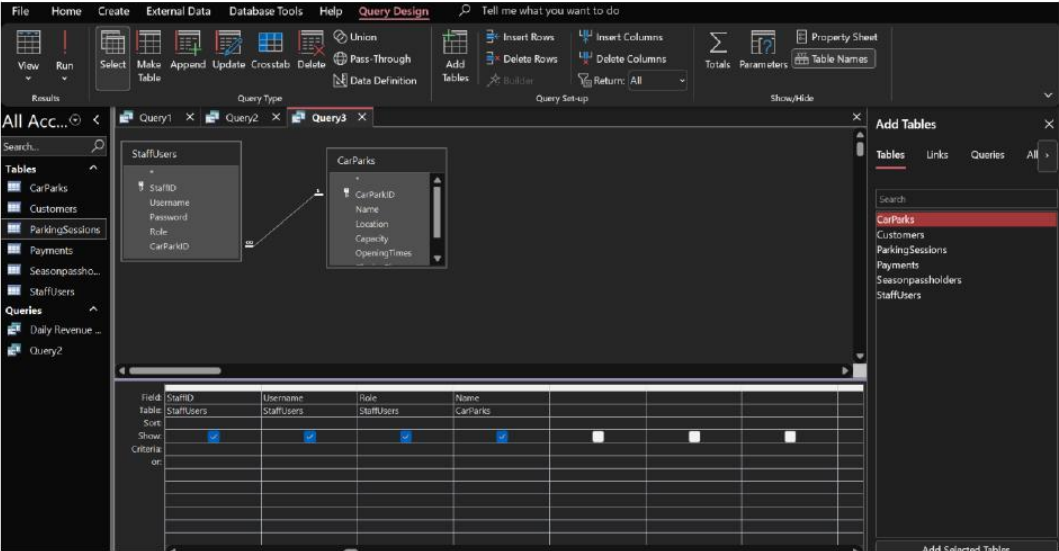
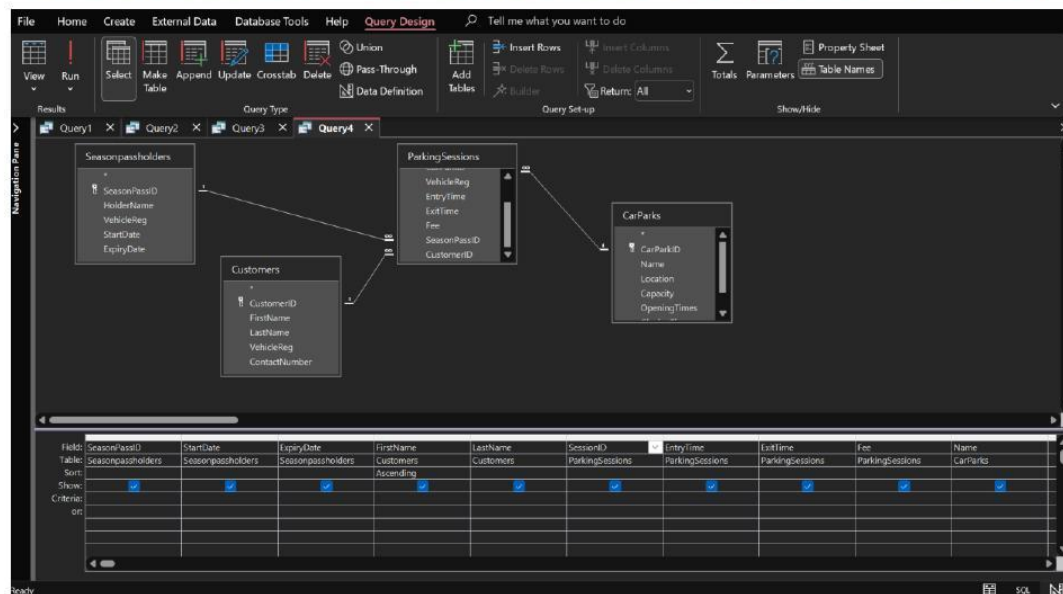


Figure 32 shows the Datasheet view of 'Query3'. The table displays data for StaffID, Username, Role, and Name. The first five rows are: 1 Hamed123 Supervisor NorthLondon, 2 Peter321 Administrator WestLondon, 3 EastLondon, 4 SouthLodon, 5 Centrallondon. A sixth row is marked with an asterisk and '(New)'.

StaffID	Username	Role	Name
1	Hamed123	Supervisor	NorthLondon
2	Peter321	Administrator	WestLondon
3			EastLondon
4			SouthLodon
5			Centrallondon
*	(New)		

4) Season-Pass-Holders Duration (Tables: Season-Pass-Holders, Customers, Parking-Sessions, Carparks) (design and data view).

Figures 33&34: Season-Pass-Holders Duration Query (Source MS Access)

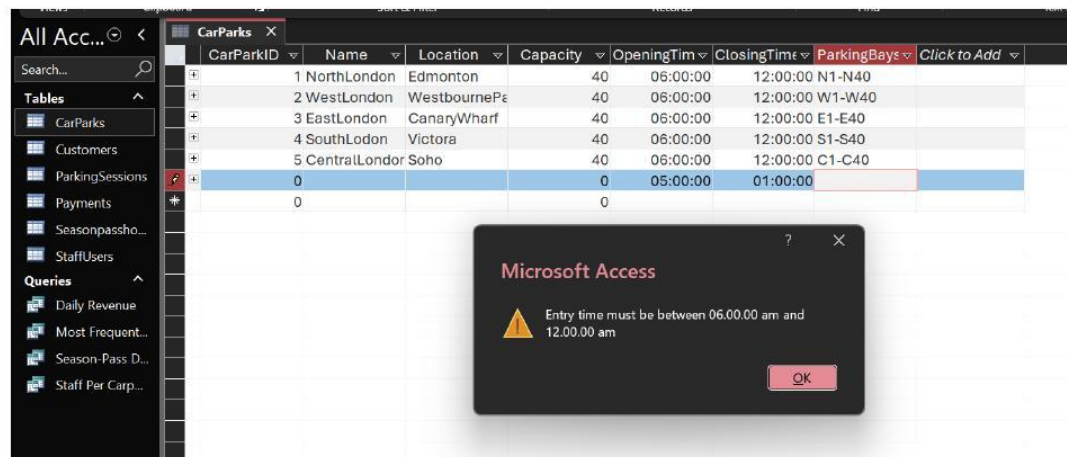
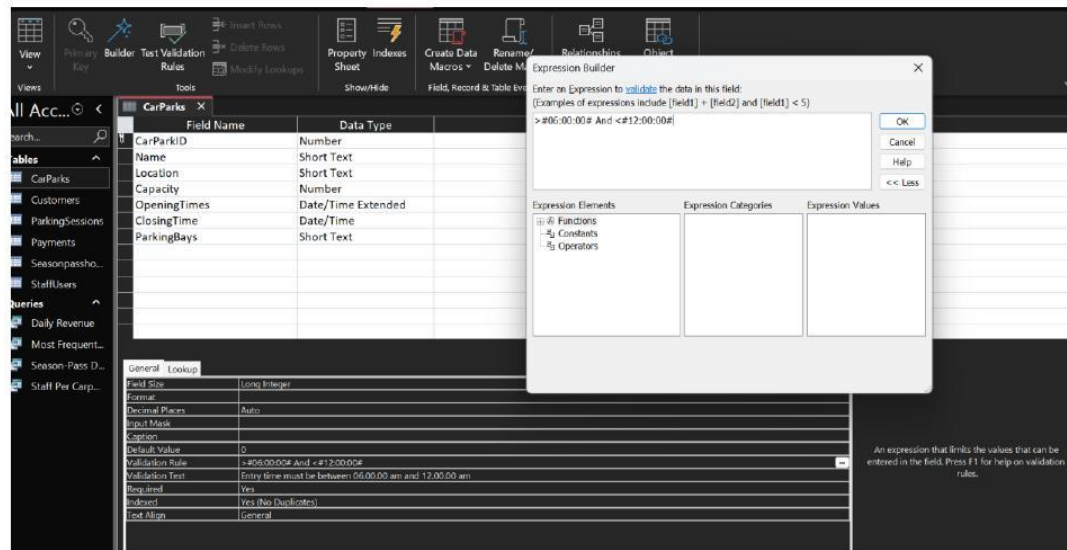


SeasonPassID	StartDate	ExpiryDate	FirstName	LastName	SessionID	EntryTime	ExitTime	Fee	Name
134	20/04/2026	30/09/2026	Ahmed	Elbadawi	13	14:00:00	11:00:00		45 EastLondon
156	02/02/2026	17/07/2026	Hamed	Mohammed	15	10:00:00	13:00:00		25 Centrallondon
145	01/04/2026	01/08/2026	Hanna	Rahim	14	07:00:00	15:00:00		40 SouthLondon
112	12/03/2026	12/06/2026	Jake	Harrow	11	12:00:00	03:00:00		15 NorthLondon
123	17/06/2026	23/10/2026	Mo	Khalifa	12	09:00:00	19:00:00		50 WestLondon

Task 7: Validation And Testing

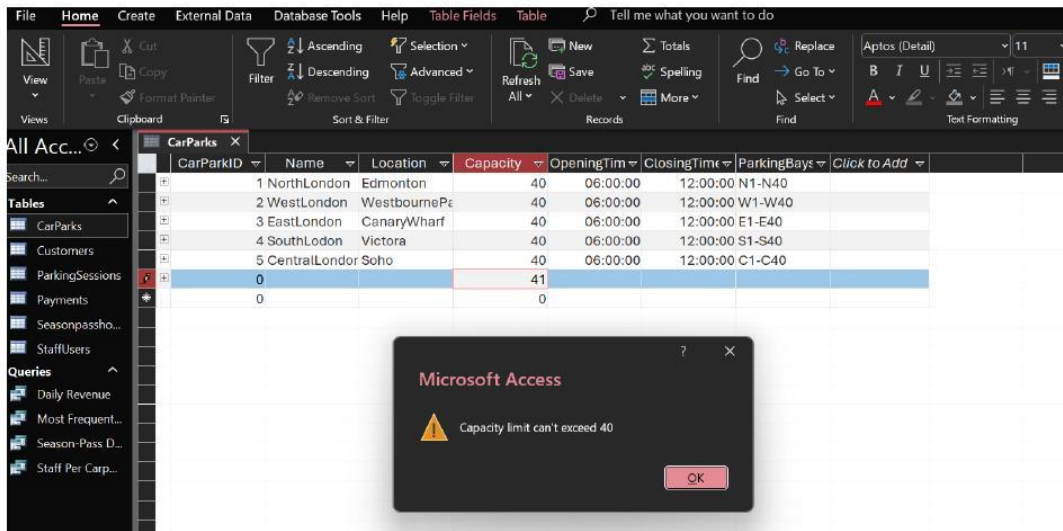
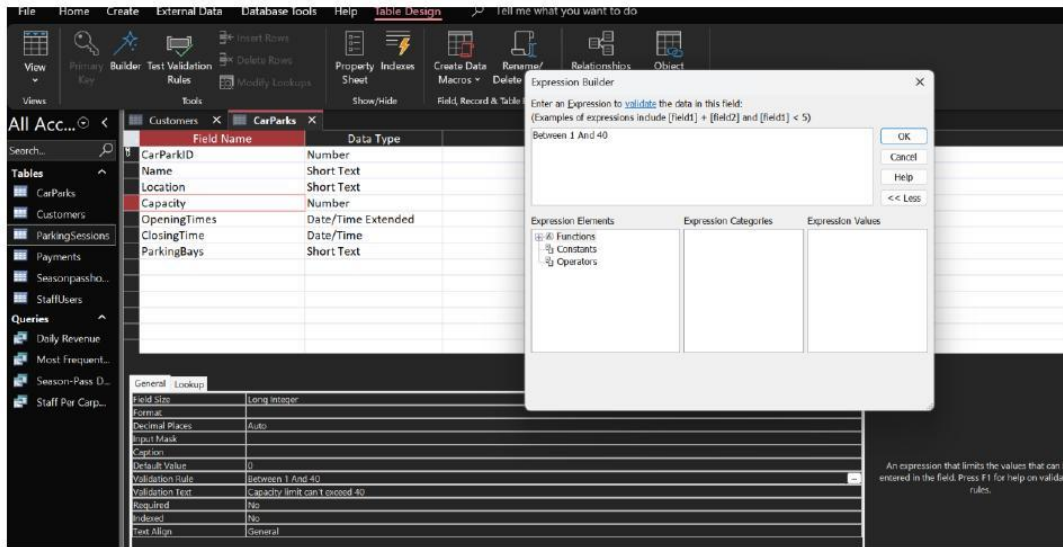
1) The first validation I created is that the entry time to carparks must be between 6 am and 12 am.

Figures 35&36: Validation and Testing for Entry Time



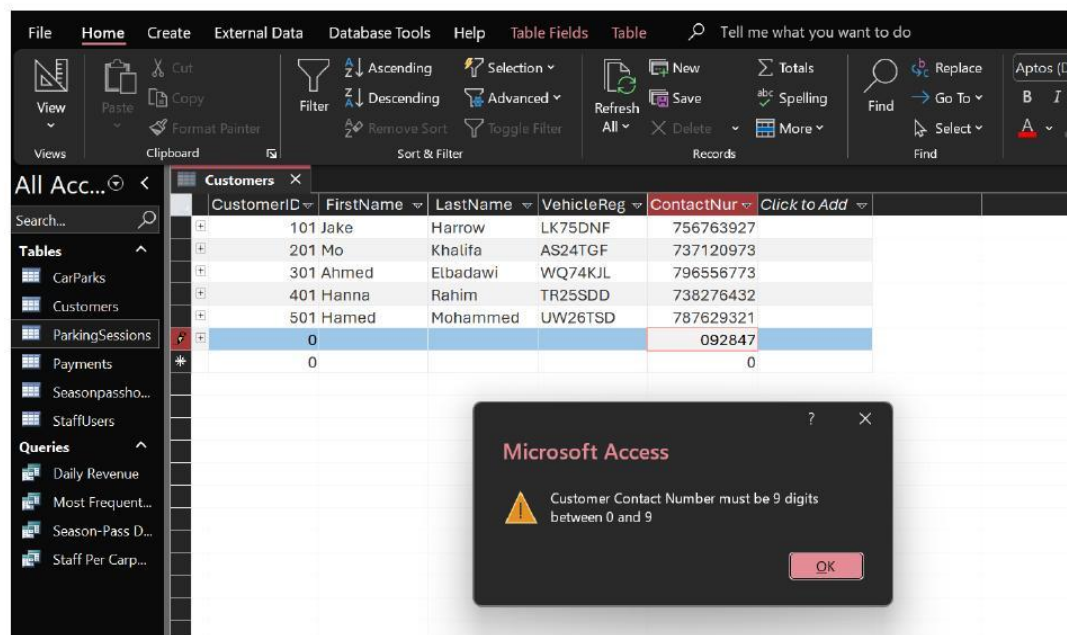
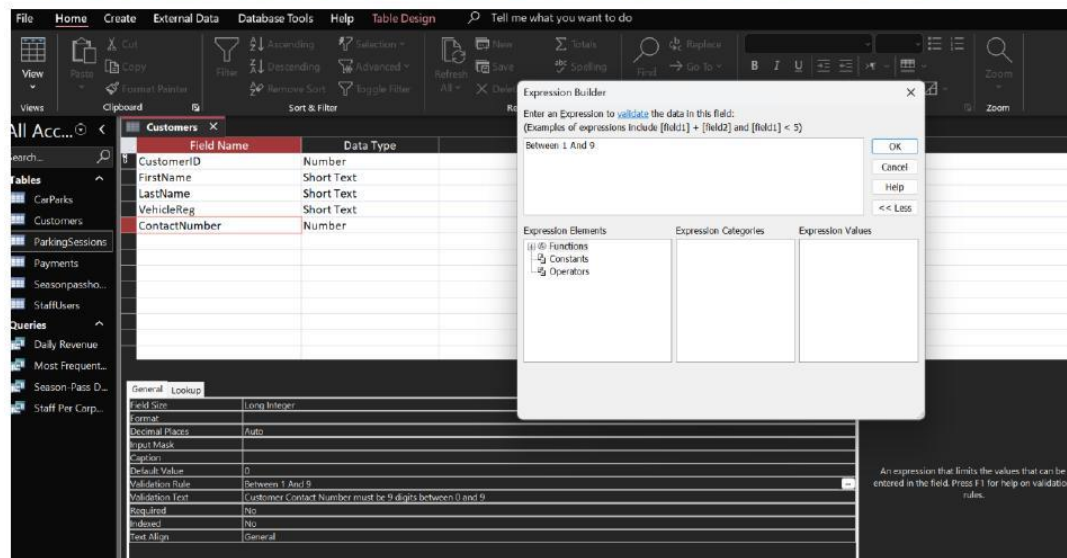
2) The second validation I created is that the capacity limit can't exceed 40 spaces.

Figures 37&38: Validation and Testing for Capacity Limit



3) The third validation I created is that the number of digits for customer contact numbers must be at least 9 digits.

Figures 39&40 Validation and Testing for Customer Contact Number



Sources:

Lucid Charts: https://lucid.app/lucidchart/04171359-cf43-41c9-ac32-6ac096aec54b/edit?invitationId=inv_c620a475-1b66-462f-b124-642ff6bc2a2f&page=0_0#

Design and Implementation of a Relational Database System

Lecturer: Suresh Shanmugasundaram

Student Name: Hamed Huffi

Student Number: 2528907

Group 1

Date: 21/07/2026

AutoPark
Database
System

Table of Contents

Introduction	2
Scenario	2
AutoPark Management System:.....	2
Screenshots of Task 1	3
Entity-Relationship-Diagram and Cardinality:.....	3
Data Structure:	7
Normalisation:	11
Population of Data:.....	17
Query Development:	18
Validation And Testing:	21
Design decisions.....	25
Database:.....	25
Cardinalities:	26
ERD:	26
Data Structure:.....	27
Normalisation:	27
Populating Data:.....	27
Query Development:	27
Validating And Testing:	28
Conclusion	28
References and Sources List.....	28

Introduction

In this assessment, I'm serving as an external MS Access developer creating a database solution for a company called **AutoPark Management System**. I will describe the scenario, provide a solution, and elaborate on and demonstrate how the six sections and core entities helped build my database. These sections are an Entity-Relationship Diagram, Data Structure, Normalisation Compliance, Population of Database, Query Development, and Validation and Testing. Core entities are Carparks, Parking-sessions, Payments, Customers, Season-Pass-Holders, and Staff-Users.

Scenario

AutoPark Management System:

The operations director for AutoPark Management System is seeking to transform their current manually managed model into a fully automated, reliable, and consistent database system for multiple (5) carparks around the city of London using MS Access. The organisation currently documents its information and data by hand, which has led to the disappearance of data, errors, and uneconomical allocation of parking bays. Hence, the company wants to fully automate it and rely on a resilient system to manage its multiple (5) facilities throughout the city of London.

The existing system stores data such as vehicle entries, parking bays, client information, and payment transactions. As these are hand-entered entries, there are bound to be inconsistencies. This has resulted in a loss of profits for the organisation, confusion, and frustrated customers who would prefer to use a better, more convenient service.

Therefore, a unified, durable database solution that accurately logs information immediately is urgently needed to improve the service. As the external Microsoft Access developer, I created a database to coordinate operations across five car parks in the city of London. It consists of six main entity tables that form the database system. These are the Carparks Table, Parking Sessions Table, Season-Pass-Holders Table, Payments Table, Customers Table, and Lastly the Staff User table.

The relationship between AutoPark Management System and its potential consumers is a business-to-consumer (B2C) relationship, in other words, a parking service provider built on accessibility, customer convenience, assurance, and fee settlement [1]. So, providing this solution will assist immensely and avoid any future discrepancies.

Screenshots of Task 1

Entity-Relationship-Diagram and Cardinality:

Task 2: Entity Relationship Diagram (ERD)

Here is a screenshot showing what the database will look like in an ERD form and how the constraints, primary and foreign keys, connect across tables to form relationships with cardinality.

Figure 1: Entity Relationship Diagram

(Source Lucid charts): https://lucid.app/lucidchart/04171359-cf43-41c9-ac32-6ac096aec54b/edit?invitationId=inv_c620a475-1b66-462f-b124-642ff6bc2a2f&page=0_0#

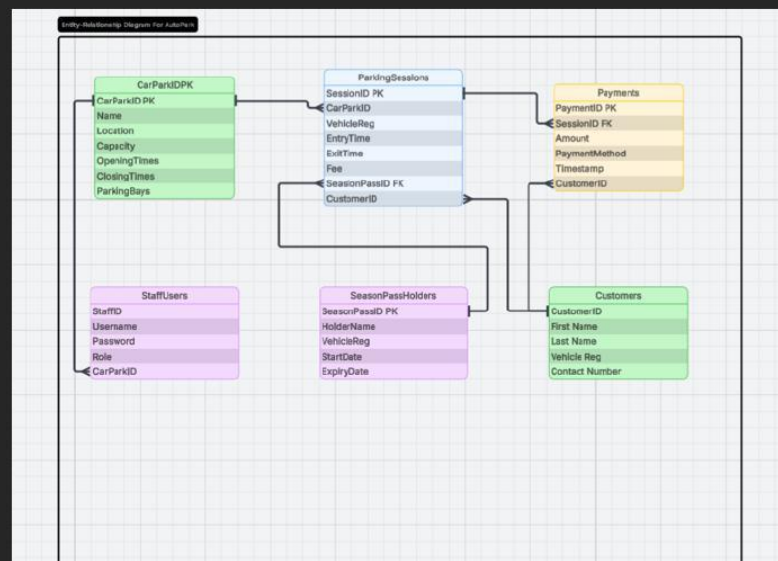


Figure 2: Cardinality between the Carparks and the Parking sessions tables. (One-To-Many) (Source MS Access)

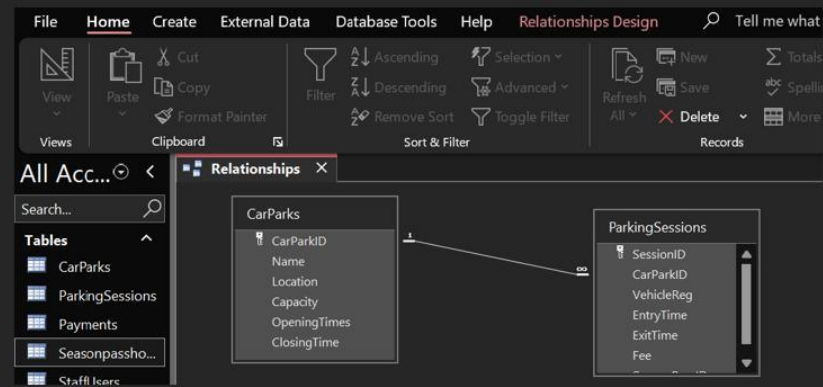


Figure 3: Cardinality between the Parking sessions and the Payments tables. (One-To-Many) (Source MS Access)

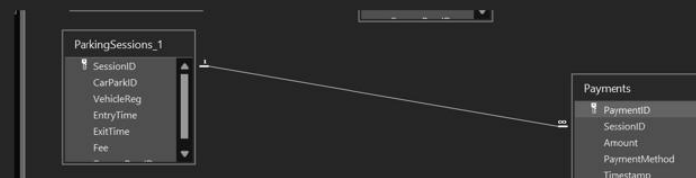


Figure 4: Cardinality between the Season-Pass-Holders and the Parking Sessions tables. (One-To-Many) (Source MS Access)

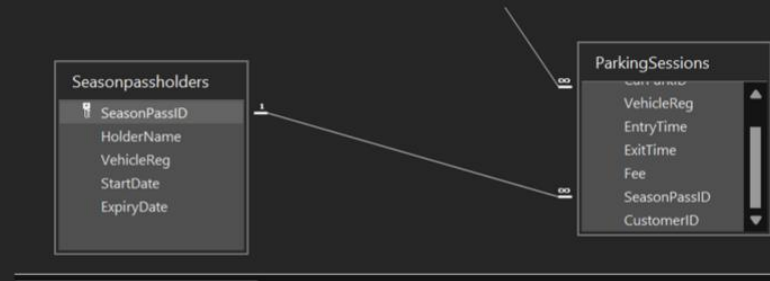


Figure 5: Cardinality between the Customers and the Parking sessions tables. (One-To-Many) (Source MS Access)

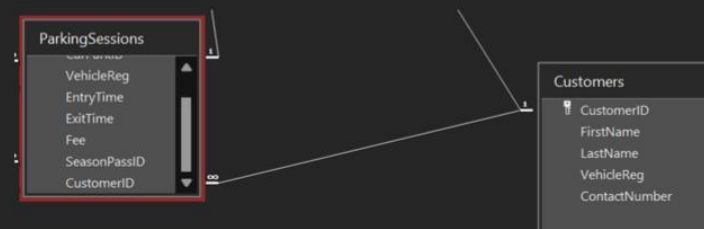
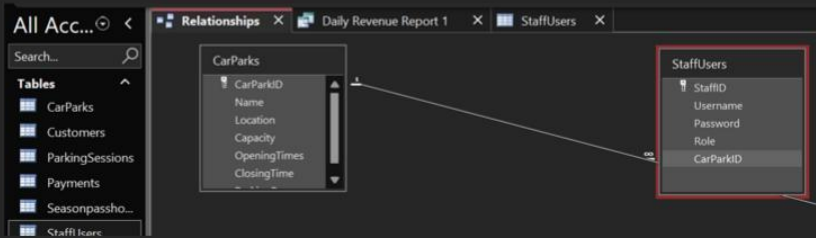


Figure 6: Cardinality between the Customers and the Payments tables. (One-To-Many) (Source MS Access)



Figure 7: Cardinality between the Staff and the Carparks tables. (One-To-Many) (Source MS Access)



Data Structure:

Task 3: Data Structure

The new database blueprint I built will look like this, labelling each entity, its constraints, and sub-categories, and showing how they connect through primary keys and foreign keys to guarantee data consistency.

Figures 8&9: Carparks Table (Source MS Access) Design and Data view.

The image displays two screenshots of the Microsoft Access interface for the 'CarParks' table.

Top Screenshot (Data View): Shows the table with the following data:

CarParkID	Name	Location	Capacity	OpeningTime	ClosingTime	ParkingBays
1	NorthLondon	Edmonton	40	06:00:00	12:00:00	N1-N40
2	WestLondon	WestbournePc	40	06:00:00	12:00:00	W1-W40
3	EastLondon	CanaryWharf	40	06:00:00	12:00:00	E1-E40
4	SouthLondon	Victoria	40	06:00:00	12:00:00	S1-S40
5	CentralLondon	Soho	40	06:00:00	12:00:00	C1-C40
0			0			

Bottom Screenshot (Design View): Shows the table structure with the following fields and data types:

Field Name	Data Type	Description (Optional)
CarParkID	Number	
Name	Short Text	
Location	Short Text	
Capacity	Number	
OpeningTimes	Date/Time Extended	
ClosingTime	Date/Time	
ParkingBays	Short Text	

Below the field list, the 'Field Properties' pane shows settings for the 'CarParkID' field:

Property	Value
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Default Value	0
Validation Rule	
Validation Text	
Indexed	Yes (No Duplicates)
Primary Key	Yes
Text Align	General

Figures 10&11: ParkingSessions Table (Source MS Access) Design and Data view.

All Acc... Relationships x Daily Revenue Report 1 x CarParks x ParkingSessions x

Search...

Tables

- CarParks
- Customers
- ParkingSessions
- Payments
- SeasonPasses...
- StaffUsers

Queries

- Daily Revenue ...

SessionID	CarParkID	VehicleReg	EntryTime	ExitTime	Fee	SeasonPassID	CustomerID
1	1	LK75DNF	12:00:00	03:00:00	15	112	101
12	2	AS24TGF	09:00:00	19:00:00	50	123	201
13	3	WQ74KJL	14:00:00	11:00:00	45	134	301
14	4	TR26GDD	07:00:00	15:00:00	40	145	401
15	5	UW26TSD	10:00:00	13:00:00	25	156	501
0	0				0	0	0

All Acc... Relationships x Daily Revenue Report 1 x CarParks x ParkingSessions x

Search...

Tables

- CarParks
- Customers
- ParkingSessions
- Payments
- SeasonPasses...
- StaffUsers

Queries

- Daily Revenue ...

Field Name	Data Type	Description (Optional)
SessionID	Number	
CarParkID	Number	
VehicleReg	Short Text	
EntryTime	Date/Time	
ExitTime	Date/Time	
Fee	Number	
SeasonPassID	Number	
CustomerID	Number	

Field Properties

General

Field Size	Long Integer
Primary	Yes
Normal Status	Yes
Field Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Not Null	Yes

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F5 = Switch panels. F1 = Help.

Figures 12&13: Payments Table (Source MS Access) Design and Data view.

All Accounts

Search...

Tables

- CarParks
- Customers
- ParkingSessions
- Payments
- Seasonpassho...
- StaffUsers

Queries

- Daily Revenue ...

Relationships

Daily Revenue Report 1

Payments

PaymentID	SessionID	Amount	PaymentMe	Timestamp	CustomerID	Click to Add
1	14	40	BACS	15:00:00	401	
2	12	50	TicketMachine	19:00:00	201	
3	11	15	BACS	03:00:00	101	
4	13	45	BACS	11:00:00	301	
5	15	25	TicketMachine	13:00:00	501	
0	0	0			0	

All Accounts

Search...

Tables

- CarParks
- Customers
- ParkingSessions
- Payments
- Seasonpassho...
- StaffUsers

Queries

- Daily Revenue ...

Relationships

Daily Revenue Report 1

Payments

CarParks

Field Name	Data Type	Description (Optional)
PaymentID	Number	
SessionID	Number	
Amount	Number	
PaymentMethod	Short Text	
Timestamp	Date/Time	
CustomerID	Number	

General

Lookup

Field Type	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	No
Not Null	General

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. R6 = Switch panel. F1 = Help.

Copy Lock

Figures 14&15: Season Pass Holders Table (Source MS Access)
Design and Data view.

SeasonPassID	HolderName	VehicleReg	StartDate	ExpiryDate
112	Mike	LK75DNF	12/03/2026	12/06/2026
123	Mo	AS24TGF	17/06/2026	23/10/2026
134	Ahmed	WQ78KJL	20/04/2026	30/09/2026
145	Hanna	TR25DD	01/04/2026	01/08/2026
156	Hamed	UW26TSD	02/02/2026	17/07/2026
0				

Field Name	Data Type	Description (Optional)
SeasonPassID	Number	
HolderName	Short Text	
VehicleReg	Short Text	
StartDate	Date/Time	
ExpiryDate	Date/Time	

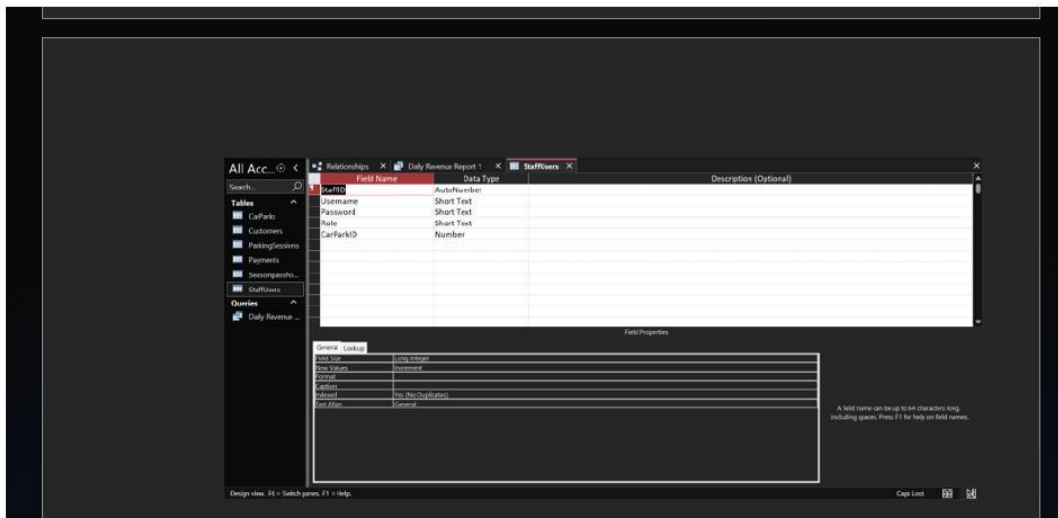
Field Properties

Field Name	Field Size	Field Properties
SeasonPassID	Byte Integer	
HolderName	Short Text	
VehicleReg	Short Text	
StartDate	Date/Time	
ExpiryDate	Date/Time	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Figures 16&17: Staff Users Table (Source MS Access) Design and Data view.

StaffID	Username	Password	Role	CarParkID
1	Hamed123	Pass	Supervisor	1
2	Peter321	Word	Administrato	2
3				3
4				4
5				5
(New)				0



Normalisation:

Task 4: Normalisation Compliance

1) Unnormalized Data: Issue Repeated Groups and more than one value in fields. (slot numbers have multiple values, not atomic)

Figure 19: Unnormalized Data (Source MS Excel)

The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	BookingID	CustomerName	Phone	SlotNumbers	PaymentMethod	
2	901	Jhon	7954839256	10,11	BACS	
3	902	Hanna	7840149257	12	TicketMachine	
4						
5						
6						
7						
8						

2) First Normal Form (1NF): Each slot becomes its own row (this is now atomic and in First Normal Form)

Figure 20: First Normal Form (1NF) (Source MS Excel)

BookingID	CustomerName	Phone	SlotNumbers	PaymentMethod
901	Jhon	7954839256	10	BACS
901	Jhon	7954839256	11	BACS
902	Hanna	7840149257	12	TicketMachine

3) Second Normal Form (2NF): Splitting the data into 2 tables, now every attribute depends on the Primary Key.

Figures 21,22,23: Second Normal Form (2NF) (Source MS Excel)

(Bookings-Table)

AutoSave ☐ Book1 -...

File Home Insert Draw Page Layout Formulas Data Review View

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles

D11

	A	B	C	D	E
1	BookingID	CustomerName	Phone	PaymentMethod	
2	901	Jhon	7954839256	BACS	
3	902	Hanna	7840149257	TicketMachine	
4					
5					
6					
7					
8					

(Booking-Slots-Table)

AutoSave ☐ Book1 -...

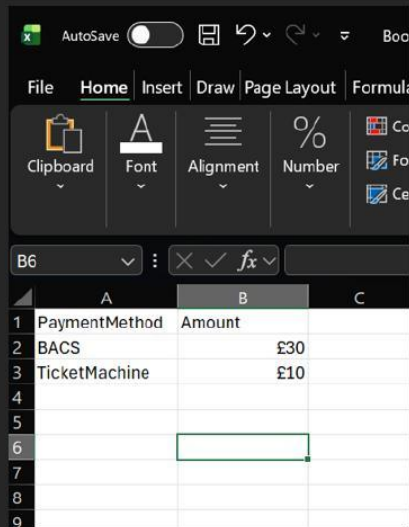
File Home Insert Draw Page Layout Formulas Data Review

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles

C5

	A	B	C	D	E
1	BookingID	SlotNumbers			
2	901	10			
3	901	11			
4	902	12			
5					
6					
7					
8					
9					
10					
11					
12					

(Payment-Method-Table)



The screenshot shows the Microsoft Excel interface. The ribbon is set to 'Home'. The active cell is B6. The table below is displayed in the worksheet.

	A	B	C
1	PaymentMethod	Amount	
2	BACS	£30	
3	TicketMachine	£10	
4			
5			
6			
7			
8			
9			

Before transforming into 3NF, the issue is that the amount depends on the Payment Method rather than directly on the Booking ID.

4) Third Normal Form (3NF): So, to convert into 3NF, you would split the tables again.

Figures 24,25,26: Third Normal Form (3NF) (Source MS Excel)

(Bookings-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review View

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles

D11

	A	B	C	D	E
1	BookingID	CustomerName	Phone	PaymentMethod	
2	901	Jhon	7954839256	BACS	
3	902	Hanna	7840149257	TicketMachine	
4					
5					
6					
7					
8					

(Booking-Slots-Table)

AutoSave Book1 -...

File Home Insert Draw Page Layout Formulas Data Review

Clipboard Font Alignment Number Conditional Formatting Format as Table Cell Styles

C5

	A	B	C	D	E
1	BookingID	SlotNumbers			
2	901	10			
3	901	11			
4	902	12			
5					
6					
7					
8					
9					
10					
11					
12					

(Payment-Methods-Table)

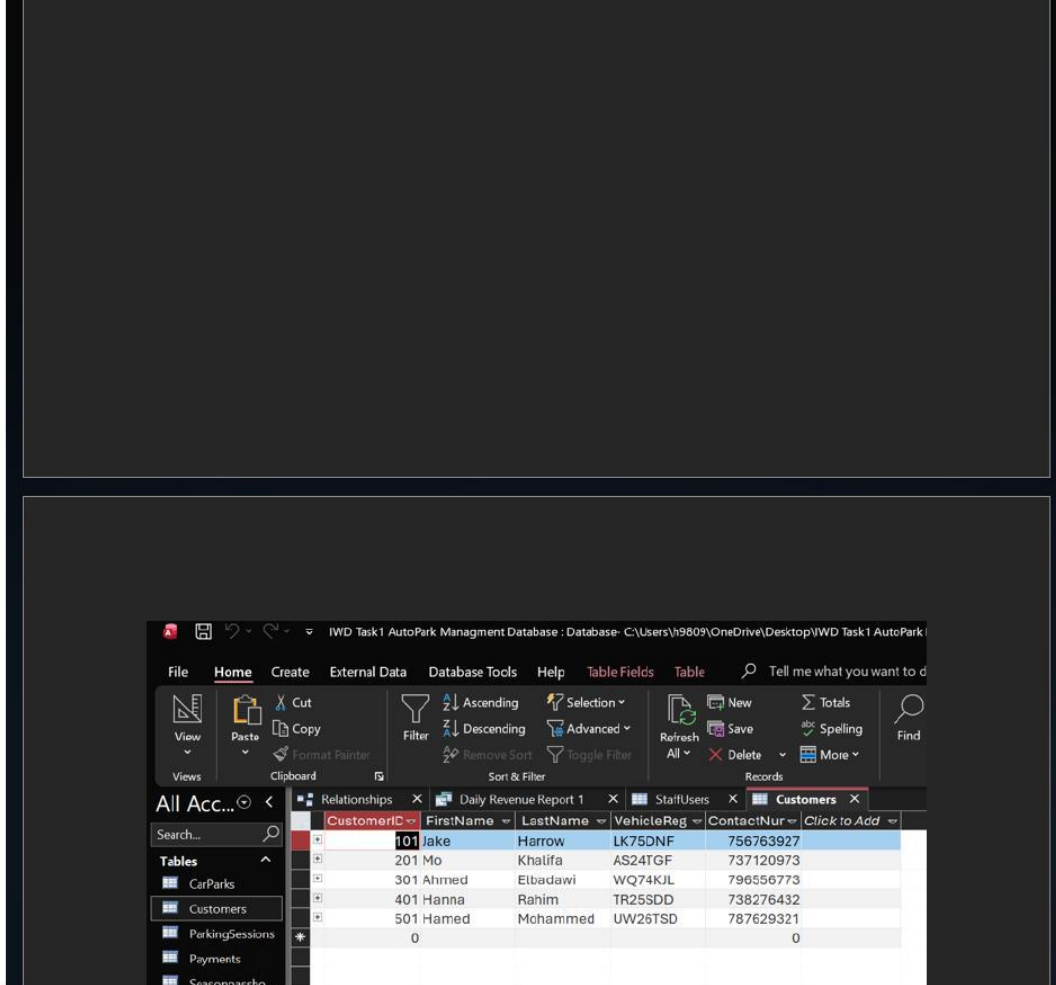
AutoSave                                 <

Population of Data:

Task 5: Population of Data in the Database

I populated the database with 5 clients, detailing their first and last names, their unique IDs, Vehicle-Reg, and contact numbers.

Figure 27: Population of Data (Source MS Access)



The screenshot displays the Microsoft Access interface for a database named 'IWD Task1 AutoPark Management Database'. The 'Customers' table is open, showing a list of 5 records. The table has columns for 'CustomerID', 'FirstName', 'LastName', 'VehicleReg', and 'ContactNur'. The records are as follows:

CustomerID	FirstName	LastName	VehicleReg	ContactNur
101	Jake	Harrow	LK75DNF	756763927
201	Mo	Khalifa	AS24TGF	737120973
301	Ahmed	Elbadawi	WQ74KJL	796556773
401	Hanna	Rahim	TR25SDD	738276432
501	Hamed	Mohammed	UW26TSD	787629321

Query Development:

Task 6: Query Development

Step-by-step sequence: 1. Open Query Design 2. Add Tables 3. Form Relationships if not created already 4. Select the relevant Fields 5. Then ensure all is sorted in your accurate preference 6. Finally, click Run to see your outcome.

1) Daily Revenue Per Carpark (Tables: Payments, Parking-Sessions, CarParks) (Design and Data View).

Figures 28&29: Daily Revenue Per Carpark Query (Source MS Access)

The top screenshot shows the Query Design view in Microsoft Access. The 'Query Design' ribbon is active. The 'Add Tables' task pane on the right shows the 'CarParks' table selected. The 'Query Design' grid shows the following fields:

Table	Field	Criteria	Criteria2
Payments	PaymentID		
Payments	Amount		
ParkingSessions	SessionID		
ParkingSessions	CarParkID		
ParkingSessions	VehicleReg		
ParkingSessions	EntryTime		
ParkingSessions	ExitTime		
CarParks	CarParkID		
CarParks	Name		
CarParks	Location		
CarParks	Capacity		
CarParks	OpeningHours		

The bottom screenshot shows the Query1 Data View. The 'Query1' ribbon is active. The 'Sort & Filter' group shows 'Ascending' selected. The results are as follows:

Name	SumOfAmount
Centrallondon	25
EastLondon	45
NorthLondon	15
SouthLondon	40
WestLondon	50

2) Most frequent customers (Tables: Parking-Sessions, Customers) (design and data view).

The screenshot displays the Microsoft Access interface. The top ribbon includes 'File', 'Home', 'Create', 'External Data', 'Database Tools', and 'Help'. The 'Query Design' tab is active, showing a query design grid with two tables: 'ParkingSessions' and 'Customers'. The 'Customers' table is linked to the 'ParkingSessions' table via the 'CustomerID' field. The 'Query Results' view is also visible, showing the results of the query. The results table has columns: 'CustomerID', 'FirstName', 'LastName', and 'CountOfSess'. The data is sorted by 'CustomerID' in descending order.

CustomerID	FirstName	LastName	CountOfSess
501	Hamed	Mohammed	1
401	Hanna	Rahim	1
301	Ahmed	Elbadawi	1
201	Mo	Khalifa	1
101	jake	Harrow	1

3) Staff Per Carpark (Tables: Staff, CarParks) (design and data view).

Figures 32&33: Staff Per Carpark Query (Source MS Access)

The screenshot displays the Microsoft Access interface for the 'Staff Per Carpark' query. The top half shows the 'Query Design' view, where the 'StaffUsers' and 'CarParks' tables are linked. The bottom half shows the 'Query3' data view, which lists staff members and their assigned carpark locations.

Query Design View:

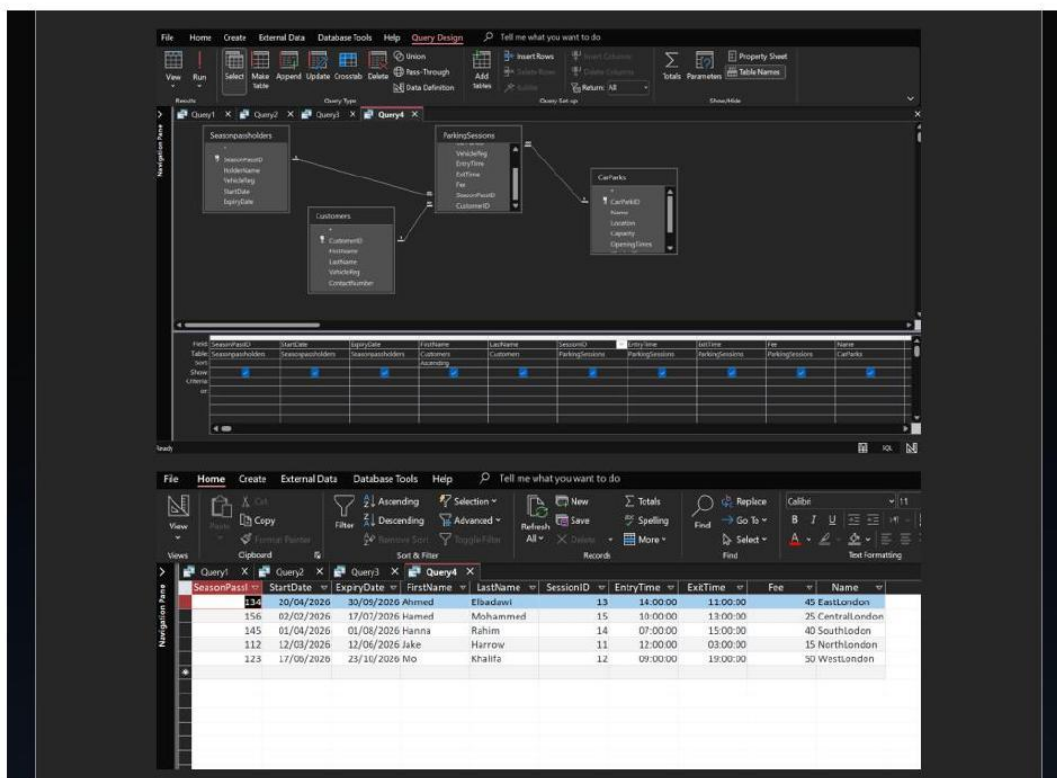
- Tables: StaffUsers, CarParks
- Fields: StaffID, Username, Role, Name, Location, Capacity, OpeningHours

Query3 Data View:

StaffID	Username	Role	Name
1	Hamed123	Supervisor	NorthLondon
2	Peter321	Adminsitrator	WestLondon
3			EastLondon
4			SouthLodon
5			CentralLondon
(New)			

4) Season-Pass-Holders Duration (Tables: Season-Pass-Holders, Customers, Parking-Sessions, Carparks) (design and data view).

Figures 34&35: Season-Pass-Holders Duration Query (Source MS Access)

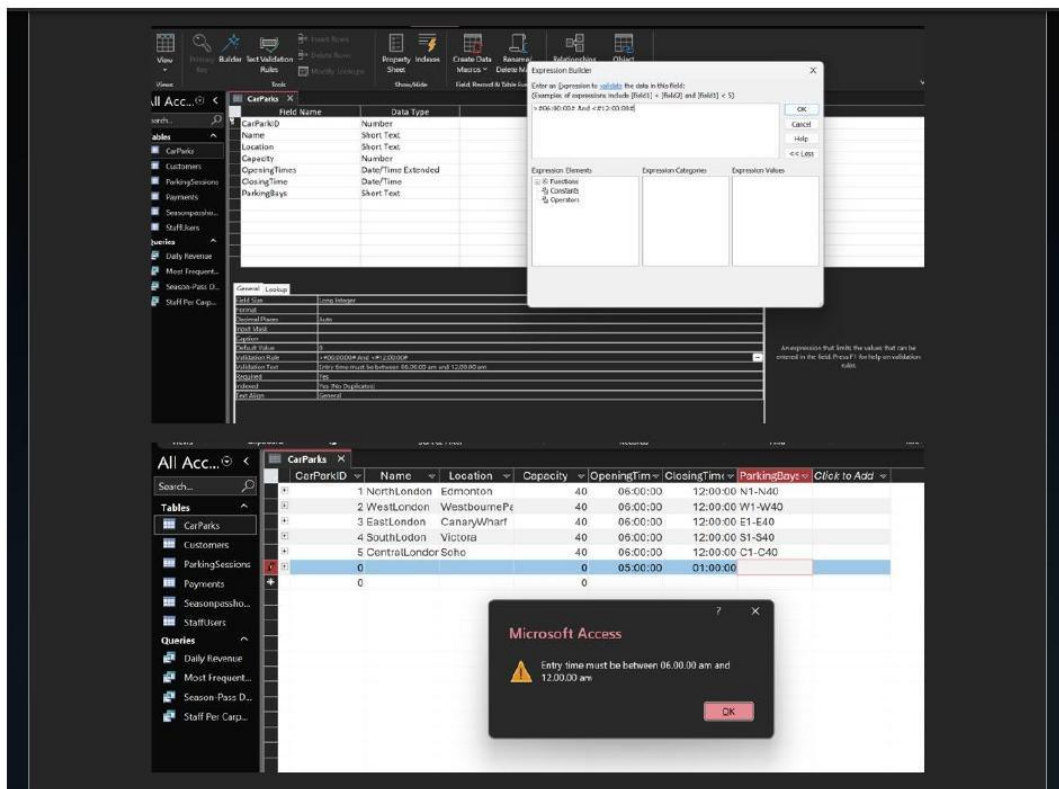


Validation And Testing:

Task 7: Validation And Testing

1) The first validation I created is that the entry time to carpark must be between 6 am and 12 am.

Figures 35&36: Validation and Testing for Entry Time



2) The second validation I created is that the capacity limit can't exceed 40 spaces.

Figures 37&38: Validation and Testing for Capacity Limit

Figure 37: Microsoft Access - Table Design View for CarParks

The screenshot shows the Table Design View for the 'CarParks' table. The fields and their data types are:

Field Name	Data Type
CarParkID	Number
Name	Short Text
Location	Short Text
Capacity	Number
OpeningTimes	Date/Time Extended
ClosingTime	Date/Time
ParkingDays	Short Text

The 'Capacity' field has a validation rule of 'Between 1 And 40'. The 'OpeningTimes' field has a validation rule of 'Between 1 And 40'. The 'ClosingTime' field has a validation rule of 'Between 1 And 40'. The 'ParkingDays' field has a validation rule of 'Between 1 And 40'. The 'CarParkID' field has a validation rule of 'Between 1 And 40'. The 'Name' field has a validation rule of 'Between 1 And 40'. The 'Location' field has a validation rule of 'Between 1 And 40'. The 'Capacity' field has a validation rule of 'Between 1 And 40'. The 'OpeningTimes' field has a validation rule of 'Between 1 And 40'. The 'ClosingTime' field has a validation rule of 'Between 1 And 40'. The 'ParkingDays' field has a validation rule of 'Between 1 And 40'.

Figure 38: Microsoft Access - Table View for CarParks

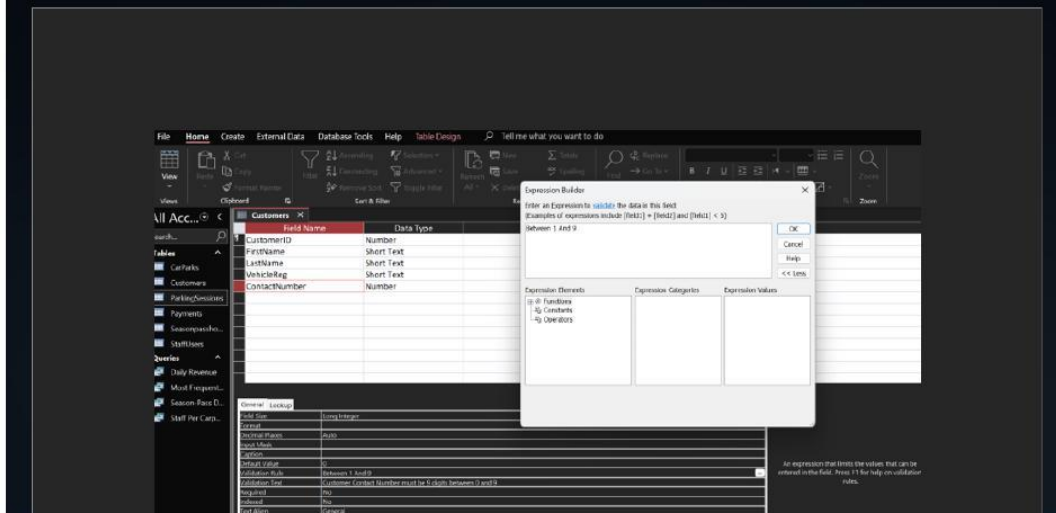
The screenshot shows the Table View for the 'CarParks' table. The data is as follows:

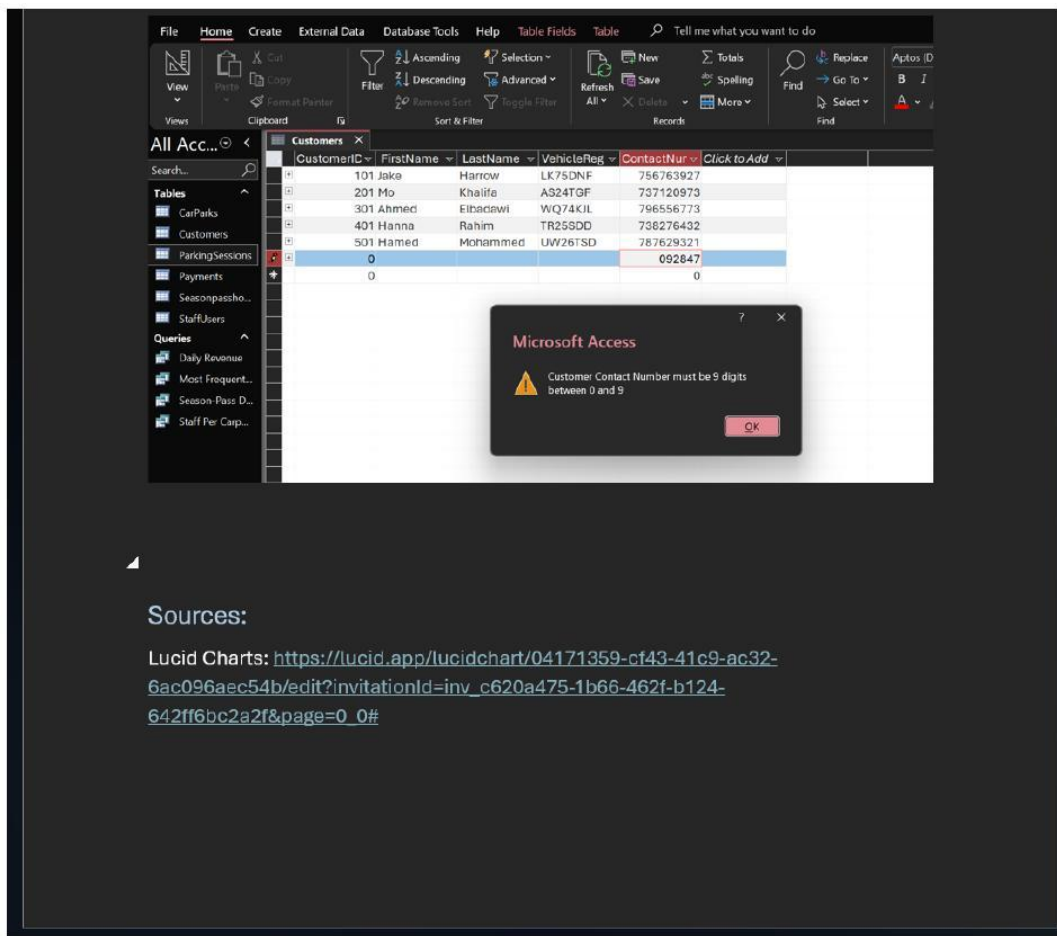
CarParkID	Name	Location	Capacity	OpeningTimes	ClosingTime	ParkingDays	Click to Add
1	NorthLondon	Epsomton	40	06:00:00	12:00:00	N1-N40	
2	WestLondon	WestbournePz	40	06:00:00	12:00:00	W1-W40	
3	EastLondon	CanaryWharf	40	06:00:00	12:00:00	E1-E40	
4	SouthLondon	Victoria	40	06:00:00	12:00:00	S1-S40	
5	CentralLondon	Soho	40	06:00:00	12:00:00	C1-C40	
6			41				
7			0				

A Microsoft Access error message box is displayed, stating: "Capacity limit can't exceed 40".

3) The third validation I created is that the number of digits for customer contact numbers must be at least 9 digits.

Figures 39&40 Validation and Testing for Customer Contact Number





Sources:

Lucid Charts: https://lucid.app/lucidchart/04171359-cf43-41c9-ac32-6ac096a54b/edit?invitationId=inv_c620a475-1b66-462f-b124-642ff6bc2a2f&page=0_0#

Design decisions

In examining closely, the scenario given describes that the current manual system in place has these entities (vehicle entries, parking slots, customer details, and payments). I have chosen six tables that mirror those entities, as well as an optional one (Season-Pass table). Contributing to a robust database solution that will not only be customer-centric but also become trendy among drivers wanting convenience and efficiency.

Database:

- 1) First is the **Car-Parks table**, which has 7 columns. They are in the following order: CarParkID, Name, Location, Capacity, Opening Times, Closing Times, and Parking-Bays. This table manages five facilities across the city, each with a capacity of 40 parking spaces, and all operate between the hours of 6 am and 12 am. The first column, Car-Park-ID, is the primary key that forms the relationship with the Parking-Sessions table and the Staff-Table, which manages the allocation of the parking bays.
- 2) Second is the **Parking Sessions table**, which contains 8 columns that record each potential client's use of the facility. The first column is the Session ID, which is the primary key for this

table. Next, we have the CarParkID (1st Foreign Key), which connects to the first entity (Car-Parks-Table).

And the remaining columns include the customer's vehicle registration number, the fee paid for the session, carpark usage time, and two foreign keys, one nullable and one non-nullable. (These are Vehicle-Registration, Entry-Time, Exit-Time, Fee, CustomerID (2nd foreign key), and Season-Pass ID (3rd foreign key, for season-pass-holders), which is the optional nullable foreign key creating the relation to the Season-Pass-holders table, as not all customers buy a season pass and may purchase a one-off session for an unlimited-hour period, depending on slot availability. Which is why I also have a CustomerID (non-nullable foreign key) to form a relationship with the customer's table.

3) Third is the **Season-Pass-Holders table**, which stores all essential details of season-pass-holders/clients. It contains 5 columns: Season-Pass-ID (Primary Key), Holder Name, Vehicle Registration, Start Date, and Expiry Date.

4) Fourth is the **Payments table**, where the payment methods/documents used by clients are stored, the amount (£5-per-hour), the time purchased and the Payment/Session/Customer IDs. The first column, Payment-ID, is the (Primary Key), and the second column, Session-ID, is the (1st foreign key) linking to the Parking Sessions table. The last column, Customer-ID, is the (2nd foreign key), creating the relationship between the Customers table.

5) The fifth is the **Customers Table**, which contains all the vital information of consumers, such as CustomerID (Primary Key), which is the link forming the relationship between the Payments and Parking-Sessions tables, and the other attributes are First and Last names, Vehicle Registration number, and Customer Contact Numbers.

6) Lastly, the sixth is the **Staff User table** that holds all staff details, such as usernames, passwords, roles, and Staff ID (Primary Key). The last column, Car-Park-ID, is the foreign key that establishes the relationship with the CarParks table, allowing staff members to administer the parking bays.

Cardinalities:

- **One** carpark can have **many** parking sessions
- **One** parking session can have **many** payments,
- **One** customer can have **many** payments or parking sessions
- **One** season-pass-holder can have **many** parking sessions
- **One** carpark can have **many** staff members

ERD:

A relational data diagram, often referred to as an Entity-Relationship Diagram (ERD), is an organised illustration framework that portrays datasets, core entities, and the relationships connecting them in the system blueprint [2]. It enables developers to interpret the movement of information and the ways modules communicate. A common Entity-Relationship Diagram consists of tables, their attributes and constraints like Primary and Foreign Keys to uphold data accuracy [2]. Relationships represent the connections between entities by defining

cardinalities. For example, the student table has a one-to-many cardinality, and the lectures table has a one-to-one cardinality. These relational data illustrations are fundamental when it comes to the development stage of the database, as they minimise discrepancies among essential elements and deliver a robust foundation [2].

In creating my database for the AutoPark Management System, I took these characteristics into consideration, as they will form the core logic of a consistent, reliable database solution.

Data Structure:

In MS Access, a data structure means the organisation, placement, and relational mapping of a dataset inside an entity-based data management system [3]. The primary method MS Access uses to build and structure its database is through tables [3]. Every single table serves as the corresponding entity, like staff, payments, or clients. Within each table, there are fields that identify properties, and every entry represents a specific record for that entity. MS Access applies form by defining data-type rules and enforcing consistent and correct data entries [3]. Associations between tables are implemented via primary and foreign keys, allowing data to maintain relationships without repeating information.

This organised setup enabled me to create a database structure that ensures consistent data control for AutoPark Management System.

Normalisation:

Data normalisation, also known as modelling, is an organised method for arranging data to eliminate unnecessary repetition in databases and boost data integrity [4]. It applies step-by-step rules in sequence, First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), to guarantee one value per field, correct partial dependencies (composite keys), and resolve transitive dependencies (dependency) [4].

Implementing these three forms and splitting tables demonstrated to me that normalisation rules improve process reliability and productivity.

Populating Data:

Inserting records in your database tables includes essential information about entities. Allowing developers to validate forms, reports, queries, and to make a general analysis by enforcing integrity checks and data format rules on attributes to preserve alignment within your MS Access database.

I have populated my database with 5 clients, each with their own Customer-IDs and personal details. This information helped me create queries such as the most frequent customers and the duration of season-pass holders.

Query Development:

Query construction is the step-by-step method of building, tuning, and executing database queries to produce the information you need [5]. It requires you to understand the user demands, identify the required tables to be used, and add query constraints, thereby enhancing system throughput. An impactful query development maintains data integrity, minimises execution delays, and contributes to making better decisions [5].

Following the step-by-step method, I created four queries based on my database solution. This will assist AutoPark Management System in identifying its daily revenue per carpark, its most frequent customers, how many season-pass holders utilise its facilities, and how many staff are required to run a carpark.

Validating And Testing:

MS Access implements validation constraints and data format rules to prevent inaccurate data that does not meet the required criteria [6]. Testing those validations proves that queries, forms, and reports execute precisely. Collectively, they uphold data soundness, stop inconsistency and support stability daily [6].

In my database solution, I created three validation rules that will assist AutoPark Management System. In recognising vehicle entry times, checking that facility capacity doesn't exceed 40 spaces, and ensuring that clients' contact numbers are entered with at least 9 digits.

Conclusion

In summary, I have created a resilient database solution for AutoPark Management System. It is an upgrade over the previous manual model and will ensure there is no parking inefficiency, information loss, or mistakes.

References and Sources List

References

- [1] Q. Siluo, "An Analysis of B2C and B2B Dynamics in the US Consumer Goods Market," vol. 11, no. 02, pp.105–117, 2025,
- [2] M. R. Syarifatunnisa and Y. Firmansyah, "Creating Of Electronic Filing System Based Microsoft Access at PT Mitra Rajawali Banjaran," vol. 17, no. 2, pp.178–189, 2023,
- [3] R. McFadyen, "Relational Databases and Microsoft Access 365," 2022,
- [4] M. Hardini, V. Agarwal, D. Apriani, I. A. Widjaya, E. Setiawaty and N. Nurasiah, "Application of database normalization in increasing data storage efficiency," vol. 3, no. 2, pp.201–211, 2025,
- [5] T. L. Dim and W. P. P. Kyaw, "No title,"
- [6] Microsoft Corporation. "Restrict data input by using validation rules." Accessed: 08/07 2026. [Online]. Available: https://support.microsoft.com/en-us/access/restrict-data-input-by-using-validation-rules?utm_source=copilot.com

Lucid Charts: https://lucid.app/lucidchart/04171359-cf43-41c9-ac32-6ac096aec54b/edit?invitationId=inv_c620a475-1b66-462f-b124-642ff6bc2a2f&page=0_0#

Note: Total word count: 2198 (2000 + 10% - 2 words)