Module: CSCU9B3

Assessment: ASSIGNMENT

Due Date/Time: 17 November 2025/17:00

AIAS Levels Allowed: 2

	Please tick the boxes/include appropriate information below
Student ID Number	3455375
Word Count (penalties apply for exceeding the stated limit)	2677 words
I have read and understand the severity of academic misconduct – see link below	✓
I give consent for my work to be used as an exemplar to future students.	✓
I have checked my submitted document to ensure it complies with module requirements.	✓
Link to version-controlled file (i.e on OneDrive, Google Docs, Github, or other) which contain evidence of the process I undertook to complete this assignment. Information on how to create a Microsoft 365 OneDrive folder is available HERE. *Please see notes below	3455375 - CSCU9B3.docx
I understand that if there is a concern about potential academic misconduct, including inappropriate use of AI tools, then I could be asked to provide evidence of my drafting process during an academic integrity meeting if I have not done so using the link above. Not providing evidence of my drafting process could prejudice the outcome of academic misconduct cases.	✓
Tailored feedback. If you would like tailored feedback on a specific aspect (or aspects) of your work (e.g., referencing, writing style, grammar), then please give details here.	I would like to get feedback on report presentation
If you used AI at (or below) the level allowed, please explain briefly which AI, how you used it, and for what purpose.	I used Chat GPT to understand the structure of the report and for brainstorming.

**This may include (but is not limited to) drafts, versions of the finished document, notes, references, AI output, and AI prompts. These materials are not marked or graded, but they are simply a way to demonstrate how your work was created and to confirm that any AI use in your final submission is within the permitted AIAS scale for your assessment. Providing this helps safeguard you, showing your authentic process, and protecting you should any academic integrity questions arise.*

https://www.stir.ac.uk/about/professional-services/student-academic-and-corporate-services/academic-registry/policy-and-procedure/

For more information, please see the Coversheet FAQ.

CSCU9B3

ASSIGNMENT REPORT

Assignment Title: Food Delivery Management System

Student ID: 3455375

Course: BSc Data Science and AI

Module Code: CSCU9B3

Table of Contents

1. INTRODUCTION

This report is all about developing the design and implementation of a Relational Database Management System named “Food Delivery Management System”. The objective of this project is to develop a database capable of managing data related to customers, delivery person, food items, and customer orders.

The Food Delivery Management System report outlines the entire process from conceptual design to SQL implementation. It includes the Entity-Relationship Diagram (ER Diagram), a discussion on normalization, table creation, sample data, SQL queries that show key functionalities, and a brief discussion of the complexity related to the system.

2. SCENARIO DESCRIPTION

The Food Delivery Management System is designed to make ordering food online easier and more organized. Customers can browse through a list of food items, place their orders, and have their orders delivered by assigned delivery staff. All the food items come from a single centralized restaurant, so restaurant need not to be a single separate entity.

Each customer can place several orders. Every order can include multiple dishes. The system keeps detailed information about which dishes are in each order.

The system tracks important data such as customer details, order date, total amount, delivery status, and the assigned delivery person. This ensures clear communication between the customer and the delivery staff. The database also allows for easy retrieval of records, order tracking, and efficient delivery management.

2.1 Assumptions

- Customer must register before placing an order
- Food items come only from a single restaurant
- Delivery handling is not included in the system
- A food price is fixed at the time of ordering
- Order cannot be created without a customer
- Restaurant is assumed to be open and available to take the order

2.2 Functional Rules

- Each customer can place many orders
- Each order belongs to one single restaurant
- Each order contains many food items
- System calculates subtotal per item and total per order
- Each food item belongs to one single restaurant
- One order can have multiple order details

3. NORMALIZATION DISCUSSION

Normalization is a process in database design that organizes data efficiently by reducing repetitive data and improve data integrity. It involves breaking large, complex tables into smaller, related ones and defining relationships between them. This ensures that each data is stored only once, which makes updates, insertions, and deletions more consistent and reliable. Normalization helps maintains a clean structured database, avoids data issues, and improves the performance and accuracy of queries.

3.1 Sample Data (Unnormalized Data)

CustomerID	CustomerName	Phone_No	Address	Email	DeliveryPersonID	DeliveryPersonName	Contact_No	FoodID	FoodName	Price	Quantity	TotalAmount	OrderID	OrderDate	Delivery_Status
C001	Aisha Khan	551111111	Al Nakheel, RAK	aisha.khan@gmail.com	D001	Bilal Ahmed	501110001	F001	Biryani	20	2	40	O001	01/01/2025	Delivered
C002	Meera Shah	551111112	Al Nakheel, RAK	meera.shah@gmail.com	D002	Sarah Malik	501110002	F002	Karak Tea	5	1	5	O002	02/01/2025	Delivered
C003	Yousuf Ali	551111113	Al Dhait, RAK	yousuf.ali@gmail.com	D003	Ahmed Noor	501110003	F003	Burger	18	1	18	O003	03/01/2025	Pending
C004	Riya Das	551111114	Al Mairid, RAK	riya.das@gmail.com	D004	Faris Khan	501110004	F004	Shawarma	10	3	30	O004	04/01/2025	Out for Delivery
C005	Fahad Khan	551111115	Al Seer, RAK	fahad.khan@gmail.com	D005	Maria Faisal	501110005	F005	Juice	6	1	6	O005	05/01/2025	Delivered
C006	Noor Fatima	551111116	Al Seer, RAK	noor.fatima@gmail.com	D006	Zain Ali	501110006	F006	Water	2	1	2	O006	06/01/2025	Out for Delivery
C007	Hani Omar	551111117	Al Rams, RAK	hani.omar@gmail.com	D007	Zoya Rehan	501110007	F007	Sandwich	12	2	24	O007	07/01/2025	Pending
C008	Hassan Ibrahim	551114324	Khuzam, RAK	hassan@gmail.com	D008	Faris Khan	557421138	F008	Chicken Mandi	30	1	30	O008	08/01/2025	Delivered
C009	Amir Malik	551111119	Qusaidath, RAK	amir.malik@gmail.com	D009	Aysha Rahman	501110009	F009	Pasta	22	1	22	O009	09/01/2025	Delivered
C010	Lina Joseph	551111120	Al Mamourah, RAK	lina.joseph@gmail.com	D010	Omar Waleed	501110010	F010	Ice Cream	8	2	16	O010	10/01/2025	Delivered

fig 1: Sample Dataset

3.2 Initial ER Diagram (Before Normalization)

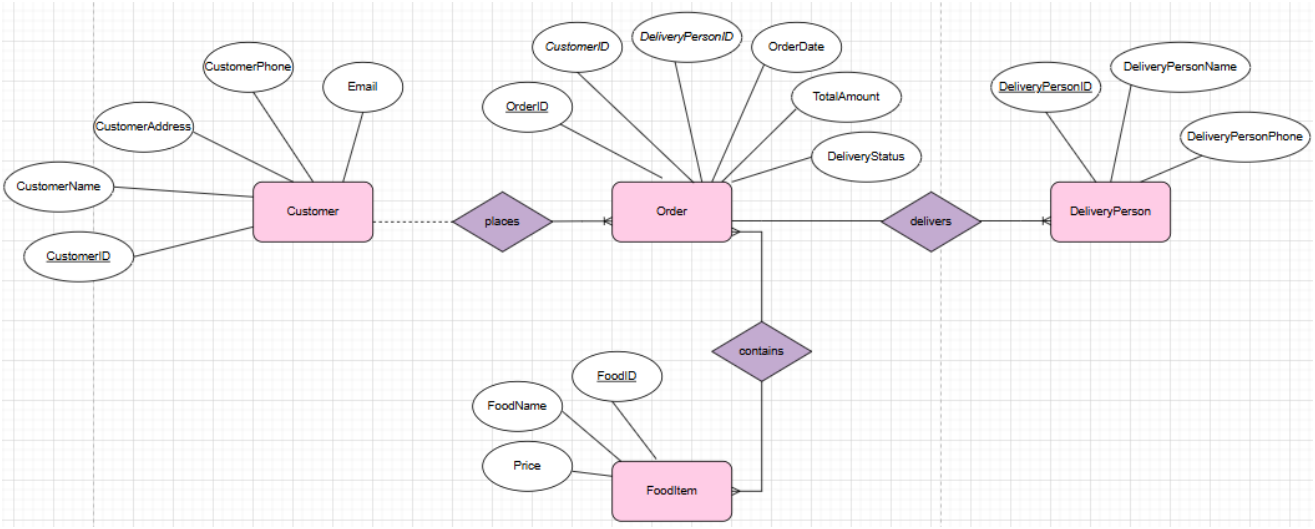

fig 2: Rough ER Diagram [3]

3.3 First Normal Form (1NF)

The dataset used in this project is already in First Normal Form (1NF). This means all tables have atomic values and do not contain repeating groups or multivalued attributes. Each record is uniquely identified by a primary key, and every attribute holds a single value relevant to that entity.

3.4 Second Normal Form (2NF)

The Second Normal Form focuses on removing partial dependencies, where a non-key attribute depends only on a part of a composite key. A table is in 2NF if it is already in 1NF and all non-key attributes depend fully on the entire primary key.

The table design is not currently in 2NF because it has partial dependencies. In this design, the composite key is (Customer ID, Food ID, Order ID)

Functional Dependencies

- Customer_ id ⟶ Customer Name, Customer Address, Customer Phone, Email

- Delivery_ Person_ id ⟶ Delivery Person Name, Delivery Person Phone
- Food_ id ⟶ Food Name, Price
- Order_ id ⟶ Customer id, Delivery Person id, Order Date, Total Amount
- Order details table: (Order id, Food id) ⟶ quantity

Partial Dependencies

A partial dependency occurs when a non-key attribute depends only on part of the composite key. Partial dependencies for this database are:

- Food Name, price depends only on food_ id and not on the full composite key (order id, food id)
- Customer_ id, customer_ name, depends only on order id and not on the full composite key (order_ id, food_ id)
- Delivery Person ID, delivery person name depend only on order_ id and not on the full composite key (order id, food id)
- Total Amount depends only on order id and not on the full composite key (order_ id, food_ id)

Since not all non-key attributes are fully dependent on the composite key, the data is highly redundant and not in 2NF.

Decomposition to Achieve 2NF

To achieve 2NF, we remove these partial dependencies by separating the attributes into distinct relations, ensuring every non-key attribute is fully functionally dependent to the primary key of its new table.
This results in the five new tables:

Table 1: Customer (customer id (PK), customer name, customer address, customer phone, email_ id)
Table 2: Delivery Person (delivery person id (PK), delivery person name, delivery person phone)
Table 3: Food Item (food id (PK), food name, price)
Table 4: Order (order id (PK), order date, total amount, delivery status, customer id (FK), delivery person id (FK))
Table 5: Order Details ((order id, food id: (composite PK)), quantity, total price)

This decomposition successfully separates all the partial dependencies. Now, all the non-key attributes in the new tables are fully functionally dependent to their primary keys.

- Customer info depends on Customer id
- Delivery Person info depends on Delivery Person id
- Food item info depends on Food id
- Order info depends on Order id
- Order Details info depends on the composite keys such as Order id and Food id

3.5 Third Normal Form (3NF)

A table is in 3NF if it is in 2NF and there are no transitive dependencies (non-key attributes depending on other non-key attributes)

Transitive Dependencies:

- Customer id ⟶ customer name (it depends on customer id, which is not the primary key of this combined table)
- Delivery Person id ⟶ delivery person name (it depends on delivery person id)

After achieving 2NF, the next step was to remove these transitive dependencies. As these were moved to their respective tables so that each non-key attribute depends only on its own primary key. With all the transitive dependencies removed, now the database is fully in Third Normal Form (3NF)

The five new tables after normalization are as follows:

Customer Table

CustomerID	CustomerName	CustomerPhone	CustomerAddress	Email
C001	Aisha Khan	551111111	Al Nakheel, RAK	aisha.khan@gmail.com
C002	Meera Shah	551111112	Al Nakheel, RAK	meera.shah@gmail.com
C003	Yousuf Ali	551111113	Al Dhait, RAK	yousuf.ali@gmail.com
C004	Riya Das	551111114	Al Mairid, RAK	riya.das@gmail.com
C005	Fahad Khan	551111115	Al Seer, RAK	fahad.khan@gmail.com
C006	Noor Fatima	551111116	Al Seer, RAK	noor.fatima@gmail.com
C007	Hani Omar	551111117	Al Rams, RAK	hani.omar@gmail.com
C008	Hassan Ibrahim	551114324	Khuzam, RAK	hassan@gmail.com
C009	Amir Malik	551111119	Qusaidath, RAK	amir.malik@gmail.com
C010	Lina Joseph	551111120	Al Mamourah, RAK	lina.joseph@gmail.com

fig 3: Customer Table

Delivery Person Table

DeliveryPersonID	DeliveryPersonName	DeliveryPersonPhone
D001	Bilal Ahmed	501110001
D002	Sarah Malik	501110002
D003	Ahmed Noor	501110003
D004	Faris Khan	501110004
D005	Maria Faisal	501110005
D006	Zain Ali	501110006
D007	Zoya Rehan	501110007
D008	Faris Khan	557421138
D009	Aysha Rahman	501110009
D010	Omar Waleed	501110010

fig 4: Delivery Person Table

Food Item Table

FoodID	FoodName	Price
F001	Biryani	20
F002	Karak Tea	5
F003	Burger	18
F004	Shawarma	10
F005	Juice	6
F006	Water	2
F007	Sandwich	12
F008	Chicken Mandi	30
F009	Pasta	22
F010	Ice Cream	8

fig 5: Food Item Table

Orders Table

OrderID	CustomerID	DeliveryPersonID	OrderDate	TotalAmount	DeliveryStatus
O001	C001	D001	01/01/2025	40	Delivered
O002	C002	D002	02/01/2025	5	Delivered
O003	C003	D003	03/01/2025	18	Pending
O004	C004	D004	04/01/2025	30	Out for Delivery
O005	C005	D005	05/01/2025	6	Delivered
O006	C006	D006	06/01/2025	2	Out for Delivery
O007	C007	D007	07/01/2025	24	Pending
O008	C008	D008	08/01/2025	30	Delivered
O009	C009	D009	09/01/2025	22	Delivered
O010	C010	D010	10/01/2025	16	Delivered

fig 6: Orders Table

Order Detail Table

OrderID	FoodID	Quantity	TotalPrice
O001	F001	2	40
O002	F002	1	5
O003	F003	1	18
O004	F004	3	30
O005	F005	1	6
O006	F006	1	2
O007	F007	2	24
O008	F008	1	30
O009	F009	1	22
O010	F010	2	16

fig 7: Order Details Table

3.6 The final ER diagram after Normalization

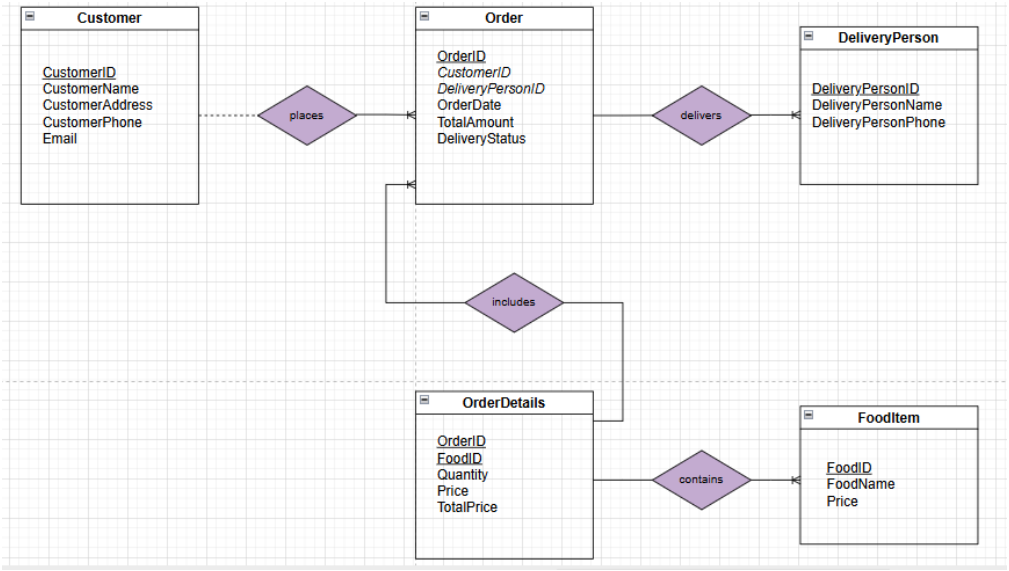

fig 8: Final ER Diagram [3]

The ER diagram shows the key parts of the Food Delivery Management System and how they are connected. It features five main entities such as: Customer, Order, Delivery Person, Food Items and Order Details. The diagram illustrates how orders move from customers to the system and how the assigned delivery person delivers the ordered food items to the customers. The Order Details table was introduced after normalization to eliminate the many-to-many relationship between Order and Food Items, ensuring the database is properly structured and avoids redundancy.

Cardinality

- Customer ⟶ Order (1:M) : One customer can place many orders, but an order contains only 1 customer

- Delivery Person ⟶ Order (1:M): One delivery person delivers many orders but one order is only delivered by a single person
- Orders ⟶ Order Details (1:M): One order can include multiple food items, represented through several order detail records
- Food Item ⟶ Order Details (1:M): A single food item may appear in many order details if customers order it multiple times
- Order ⟷ Food Item (M:N resolved): The original many-to-many relationship between order and food item is resolved by the order details table. This tables separates it into two one-to-many relationships

4. ENTITIES AND ATTRIBUTES

TABLE I

Entity-Key Relationship

Entity	**Attributes**	**Key Type**
Customer	Customer ID, Customer Name, Customer Address, Customer Phone, Email	**Primary key**: Customer ID
Order	Order ID, Customer ID, Delivery Person ID, Order Date, Total Amount, Delivery Status	**Primary key**: Order ID **Foreign key**: Customer ID (references customer)
Delivery_ Person	Delivery Person ID, Delivery Person Name, Delivery Person Phone	**Primary key**: Delivery Person ID
Food Item	Food ID, Food Name, Price	**Primary key**: Food ID
Order Details	Order ID, Food ID, Quantity, Total Amount	**Composite Primary keys**: Order ID, Food ID **Foreign Keys**: Order ID (references order), Food ID (references Food Item)

4.1 Brief Description of Each Entity

- **Customer**: Store information about customer like customer id, customer name, customer address, customer phone, and email and is linked to order via customer id
- **Order**: Stores information about each order placed by a customer, including the total amount, order date, and is linked to order details
- **Delivery Person:** Stores information about the delivery staffs and is linked to order via Delivery Person ID
- **Food items**: Stores information about different food items including price, and is linked to orders via order details
- **Order Details**: Stores information about individual items in each order, including the quantity and total price, and is linked to orders and food items.

5. SQL Table Creation

This section shows the queries used to create all the tables in Food Delivery Management Database. Each table represents an entity in the system, primary keys, foreign keys, data types and the relationships with other tables.

5.1 Queries

Customer: CREATE TABLE Customer (Customer ID VARCHAR (10) PRIMARY KEY, Customer Name VARCHAR (50) NOT NULL, Customer Phone VARCHAR (15) NOT NULL, Customer Phone VARCHAR (15) NOT NULL, Customer Address VARCHAR (100) NOT NULL, Customer Email VARCHAR (50) NOT NULL)

INSERT INTO Customer SELECT DISTINCT Customer ID, Customer Name, Customer Phone, Customer Address FROM Food_ Delivery_ Dataset;

←T→	CustomerID	CustomerName	CustomerPhone	CustomerAddress	Email
Edit Copy Delete	C001	Aisha Khan	551111111	Al Nakheel, RAK	aisha.khan@gmail.com
Edit Copy Delete	C002	Meera Shah	551111112	Al Nakheel, RAK	meera.shah@gmail.com
Edit Copy Delete	C003	Yousuf Ali	551111113	Al Dhait, RAK	yousuf.ali@gmail.com
Edit Copy Delete	C004	Riya Das	551111114	Al Mairid, RAK	riya.das@gmail.com
Edit Copy Delete	C005	Fahad Khan	551111115	Al Seer, RAK	fahad.khan@gmail.com
Edit Copy Delete	C006	Noor Fatima	551111116	Al Seer, RAK	noor.fatima@gmail.com
Edit Copy Delete	C007	Hani Omar	551111117	Al Mamourah, RAK	hani.omar@gmail.com
Edit Copy Delete	C008	Hassan Ibrahim	557421388	Khuazam, RAK	hassan@gmail.com
Edit Copy Delete	C009	Amir Malik	551111118	Qusaidat, RAK	amir.malik@gmail.com
Edit Copy Delete	C010	Lina Joseph	551111119	Al Mamourah, RAK	lina.joseph@gmail.com

fig 9: Customer Table Creation

Delivery Person: CREATE TABLE Delivery Person (Delivery person ID VARCHAR (10) PRIMARY KEY, Delivery Person Name VARCHAR (50) NOT NULL, Delivery Person Phone VARCHAR (15) NOT NULL)

INSERT INTO Delivery Person SELECT DISTINCT Delivery Person ID, Delivery Person Name, Delivery Person Phone FROM Food_ Delivery_ Dataset;

	DeliveryPersonID	DeliveryPersonName	DeliveryPersonPhone
Edit Copy Delete	D001	Bilal Ahmed	501110001
Edit Copy Delete	D002	Sarah Malik	501110002
Edit Copy Delete	D003	Ahmed Noor	501110003
Edit Copy Delete	D004	Faris Khan	501110004
Edit Copy Delete	D005	Maria Faisal	501110005
Edit Copy Delete	D006	Zain Ali	501110006
Edit Copy Delete	D007	Zoya Rehan	501110007
Edit Copy Delete	D008	Faris Khan	557421388
Edit Copy Delete	D009	Aysha Rahman	501110009
Edit Copy Delete	D010	Omar Waleed	501110010

fig 10: Delivery person Table Creation

Orders: CREATE TABLE Orders (Order ID VARCHAR (10) PRIMARY KEY, Order Date DATE NOT NULL, Total Amount DECIMAL (8,2), Customer ID VARCHAR (10) NOT NULL, Delivery Person ID VARCHAR (10), FOREIGN KEY (Customer ID) References Customer (Customer ID), FOREIGN KEY (Delivery Person ID) References Delivery Person (Delivery Person ID))

INSERT INTO Orders SELECT DISTINCT Order ID, Order Date, Total Amount, Customer ID, Delivery Person ID FROM Food_ Delivery_ Dataset;

	OrderID	OrderDate	TotalAmount	CustomerID	DeliveryPersonID	Delivery Status
Edit Copy Delete	O001	2025-01-01	40.00	C001	D001	Delivered
Edit Copy Delete	O002	2025-01-02	5.00	C002	D002	Delivered
Edit Copy Delete	O003	2025-03-03	18.00	C003	D003	Pending
Edit Copy Delete	O004	2025-04-01	30.00	C004	D004	Out for Delivery
Edit Copy Delete	O005	2025-05-05	6.00	C005	D005	Delivered
Edit Copy Delete	O006	2025-06-01	2.00	C006	D006	Out for Delivery
Edit Copy Delete	O007	2025-07-01	24.00	C007	D007	Pending
Edit Copy Delete	O008	2025-08-01	30.00	C008	D008	Delivered
Edit Copy Delete	O009	2025-09-01	22.00	C009	D009	Delivered
Edit Copy Delete	O010	2025-01-10	16.00	C010	D010	Delivered

fig 11: Orders Table Creation

Food Item: CREATE TABLE Food Item (Food ID VARCHAR (10) PRIMARY KEY, Food Name VARCHAR (50) NOT NULL, Price DECIMAL (6,2) NOT NULL)

INSERT INTO Food Item SELECT DISTINCT Food ID, Food Name, Price FROM Food_ Delivery_ Dataset;

	FoodID	FoodName	Price
Edit Copy Delete	F001	Biryani	20.00
Edit Copy Delete	F002	Karak Tea	5.00
Edit Copy Delete	F003	Burger	18.00
Edit Copy Delete	F004	Shawarma	10.00
Edit Copy Delete	F005	Juice	6.00
Edit Copy Delete	F006	Water	2.00
Edit Copy Delete	F007	Sandwich	12.00
Edit Copy Delete	F008	Chicken Mandi	30.00
Edit Copy Delete	F009	Pasta	22.00
Edit Copy Delete	F010	Ice Cream	8.00

fig 12: Food Item Table Creation

Order Details: CREATE TABLE Order Details (Order ID VARCHAR (10) NOT NULL, Food ID VARCHAR (10) NOT NULL, Quantity INT NOT NULL, Total Price DECIMAL (8,2), PRIMARY KEY (Order ID, Food ID), FOREIGN KEY (Order ID) References Orders (Order ID) ON DELETE CASCADE, FOREIGN KEY (Food ID) References Food Item (Food ID))

INSERT INTO Order Details SELECT DISTINCT Order ID, Food ID, Quantity, Total Price FROM Food_ Delivery_ Dataset;

	OrderID	FoodID	Quantity	TotalPrice
Edit Copy Delete	O001	F001	2	40.00
Edit Copy Delete	O002	F002	1	5.00
Edit Copy Delete	O003	F003	1	18.00
Edit Copy Delete	O004	F004	3	30.00
Edit Copy Delete	O005	F005	1	6.00
Edit Copy Delete	O006	F006	1	2.00
Edit Copy Delete	O007	F007	2	24.00
Edit Copy Delete	O008	F008	1	30.00
Edit Copy Delete	O009	F009	1	22.00
Edit Copy Delete	O010	F010	2	16.00

fig 13: Order Details Table Creation

6. SQL QUERIES

Query 1: Using LIKE on an Alphanumeric field

SELECT * FROM Customer WHERE Customer Name LIKE ‘H%’;

	CustomerID	CustomerName	CustomerPhone	CustomerAddress
Edit Copy Delete	C007	Hani Omar	551111117	Al Mamourah, RAK
Edit Copy Delete	C008	Hassan Ibrahim	557421388	Khuazam, RAK

fig 14: Query 1- Using LIKE

This query finds all the customers whose name start with the letter H. It helps the system quickly filter customers alphabetically.

Query 2: JOIN across multiple entities

SELECT Orders. Order ID, Customer. Customer Name, Food Item. Food Name, Order Detail. quantity FROM Orders

JOIN Customer ON Orders. Customer ID= Customer. Customer ID

JOIN Order Details ON Orders. Order ID = Order Details. Order ID

JOIN Food Item ON Order Details. Food ID= Food Item. Food ID;

OrderID	CustomerName	FoodName	Quantity
O001	Aisha Khan	Biryani	2
O002	Meera Shah	Karak Tea	1
O003	Yousuf Ali	Burger	1
O004	Riya Das	Shawarma	3
O005	Fahad Khan	Juice	1
O006	Noor Fatima	Water	1
O007	Hani Omar	Sandwich	2
O008	Hassan Ibrahim	Chicken Mandi	1
O009	Amir Malik	Pasta	1
O010	Lina Joseph	Ice Cream	2

fig 15: Query 2- Using JOIN

This query is shows which customer ordered which food item in what quantity, combining four tables. It is useful for generating order summaries.

Query 3: Using > operator

SELECT * FROM Food Item WHERE Price>10;

	FoodID	FoodName	Price
Edit Copy Delete	F001	Biryani	20.00
Edit Copy Delete	F003	Burger	18.00
Edit Copy Delete	F007	Sandwich	12.00
Edit Copy Delete	F008	Chicken Mandi	30.00
Edit Copy Delete	F009	Pasta	22.00

fig 16: Query 3- Using > Operator

This query selects food items priced more than 10 AED. It helps to identify premium or higher-priced food items.

Query 4: Using GROUP BY (Count)

SELECT Customer ID, COUNT (Order ID) AS Total Orders FROM Orders GROUP BY Customer ID;

CustomerID	TotalOrders
C001	1
C002	1
C003	1
C004	1
C005	1
C006	1
C007	1
C008	1
C009	1
C010	1

fig 17: Query 4- Using GROUP BY

This query counts how many orders each customer has placed. It helps to track customer ordering analysis.

Query 5: Using IN

SELECT * FROM Food item WHERE Food Name IN ('Biryani', 'Shawarma', 'Chicken Mandi');

	FoodID	FoodName	Price
Edit Copy Delete	F001	Biryani	20.00
Edit Copy Delete	F004	Shawarma	10.00
Edit Copy Delete	F008	Chicken Mandi	30.00

fig 18: Query 5- Using IN

This query selects only specific food items (Biryani, Pasta, Shawarma). It is useful for checking availability of popular menu items.

Query 6: UPDATE multiple rows

UPDATE Orders SET Delivery Status= “Delivered”

WHERE Order ID IN (‘O003’, ‘O006’, ‘O008’);

	OrderID	OrderDate	TotalAmount	CustomerID	DeliveryPersonID	Delivery Status
Edit Copy Delete	O001	2025-01-01	40.00	C001	D001	Delivered
Edit Copy Delete	O002	2025-01-02	5.00	C002	D002	Delivered
Edit Copy Delete	O003	2025-03-03	18.00	C003	D003	Delivered
Edit Copy Delete	O004	2025-04-01	30.00	C004	D004	Out for Delivery
Edit Copy Delete	O005	2025-05-05	6.00	C005	D005	Delivered
Edit Copy Delete	O006	2025-06-01	2.00	C006	D006	Delivered
Edit Copy Delete	O007	2025-07-01	24.00	C007	D007	Pending
Edit Copy Delete	O008	2025-08-01	30.00	C008	D008	Delivered
Edit Copy Delete	O009	2025-09-01	22.00	C009	D009	Delivered
Edit Copy Delete	O010	2025-01-10	16.00	C010	D010	Delivered

fig 19: Query 6- Using UPDATE

This query updates the delivery status of three orders. It demonstrates how the system maintains accurate and up-to-date data about each order.

7. Discussion of Complexity

One challenging part of the Food Delivery Management System in SQL is dealing with many-to-many relationships, like those between Orders and Food Items. To represent this, we need a junction table called Order Details with multiple foreign keys. Queries to calculate total or gather order-specific details involve complicated JOINs and aggregates.

Another issue is managing real-time data, such as delivery status or staff assignments. SQL alone cannot automatically manage workflows, like finding the nearest delivery person or updating status steps. This requires triggers, or external application logic.

8. References

1. University of Stirling,https://canvas.stir.ac.uk/courses/18076/pages/session-4-normalisation?module_item_id=933556
2. University of Stirling, https://canvas.stir.ac.uk/courses/18076/pages/page-8-dot-1-distributed-databases-2
3. Draw.io, https://app.diagrams.net/?src=about