Text

Description automatically generated

**MSc Information Technology**

**COMP11107 - Business Data Communication & Networks**

**Coursework Report**

**Modern Network for Two New University Buildings**

**by**

|  |  |  |
| --- | --- | --- |
| **NAME** | **BANNER ID** | **Individual contribution** |
| Shahzaib Anwar | B01799338 | **ER Data modelling** |
| Noman Javed | B01794811 | **SQL Queries & Report writing** |
| Summair Ahmed | B01795091 | **Tables & Relationships, Populate Tables** |
| Umair Ali | B01796393 | **Database Application** |
|  |  |  |

**Table of Contents**

[Conceptual Design in ER 5](#_Toc195191137)

[1.1. Introduction 5](#_Toc195191138)

[1.2. Entity Definitions 6](#_Toc195191139)

[1.3. Partial ER Diagrams 7](#_Toc195191140)

[1.4. Complete ER Diagram 8](#_Toc195191141)

[1.5. Notation and Building Blocks 10](#_Toc195191142)

[1.6. Rationale for Design Choices 10](#_Toc195191143)

[Implementation in Microsoft Access 12](#_Toc195191144)

[2.1. Overview of Implementation 12](#_Toc195191145)

[2.2. Table Definitions 12](#_Toc195191146)

[2.2.1. Field Properties and Data Types 12](#_Toc195191147)

[2.2.2. Input Masks for Primary Keys 13](#_Toc195191148)

[2.2.3. Validation Rules and Texts 13](#_Toc195191149)

[2.2.4. Indexing of Alternate Keys 14](#_Toc195191150)

[2.3. Relationship Setup and Referential Integrity 14](#_Toc195191151)

[2.4. Update/Delete Rules 14](#_Toc195191152)

[2.5. Screenshots of Access Object Designer 15](#_Toc195191153)

[Data Population 18](#_Toc195191154)

[3.1. Data Entry Strategy 18](#_Toc195191155)

[3.2. Sample Data Sets 19](#_Toc195191156)

[3.3. Demonstration of Participation and Cardinality Constraints 20](#_Toc195191157)

[3.4. Summary of Record Counts 23](#_Toc195191158)

[SQL Queries and Testing 24](#_Toc195191159)

[4.1. Query Requirements 24](#_Toc195191160)

[4.2. SQL Statement Listing 24](#_Toc195191161)

[4.3. Results and Screenshots 24](#_Toc195191162)

[4.4. Explanation of Each Query 24](#_Toc195191163)

[Database Application Interface 25](#_Toc195191164)

[5.1. Use Case Implementation 25](#_Toc195191165)

[5.2. Form Designs 25](#_Toc195191166)

[5.3. Navigation and User Workflow 25](#_Toc195191167)

[5.4. Report Designs 25](#_Toc195191168)

[5.5. Screenshots of Forms and Reports 25](#_Toc195191169)

[Privacy and Security Considerations 26](#_Toc195191170)

[6.1. Identification of Sensitive Data 26](#_Toc195191171)

[6.2. Data Protection Measures 26](#_Toc195191172)

[6.3. GDPR Compliance Analysis 26](#_Toc195191173)

[6.3.1. Data Minimization 26](#_Toc195191174)

[6.3.2. Purpose Limitation 26](#_Toc195191175)

[6.3.3. Storage Limitation 26](#_Toc195191176)

[6.3.4. Data Subject Rights 27](#_Toc195191177)

[6.4. Recommendations 27](#_Toc195191178)

[Critical Evaluation 28](#_Toc195191179)

[7.1. Reflection on Design and Implementation 28](#_Toc195191180)

[7.2. Lessons Learned 28](#_Toc195191181)

[7.3. Strengths and Limitations of the Solution 28](#_Toc195191182)

[7.4. Value of the Coursework to Your Learning 28](#_Toc195191183)

[7.5. Individual Contributions 28](#_Toc195191184)

[Appendices 29](#_Toc195191185)

[A. ER Diagram Exports 29](#_Toc195191186)

[B. Table Definitions and Properties 29](#_Toc195191187)

[C. Sample Data Screenshots 29](#_Toc195191188)

[D. SQL Query Windows and Results 29](#_Toc195191189)

[E. Form and Report Screenshots 29](#_Toc195191190)

[F. References 29](#_Toc195191191)

[Appendix A: SQL Queries 30](#_Toc195191192)

[Appendix B: Sample Data Screenshots 34](#_Toc195191193)

[Appendix C: ERD Diagrams / Visuals 40](#_Toc195191194)

# Conceptual Design in ER

# 1.1. Introduction

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector. The project addresses the digital transformation needs of a family-run business that currently relies on manual, paper-based processes for managing client information, employee schedules, service bookings, and billing operations. The aim is to develop a cost-effective, scalable, and user-friendly solution using Microsoft Access, capable of streamlining core operations while ensuring data integrity and ease of use for non-technical staff.

The proposed system consists of a fully normalized relational database with seven core entities: **Client, Employee, Service, ClientAppointments, ServicesProvided, Invoice,** and **InvoiceRows.** These entities are connected through clearly defined relationships, supporting a data model that aligns with real-world business rules—such as ensuring only qualified employees can perform certain services and tracking multiple appointments per invoice. The database supports structured queries to generate essential reports for business insights, including total revenue by service, client expenditure, and employee performance metrics.

The implementation is backed by a thorough understanding of both functional and technical requirements, as outlined in the accompanying academic brief. Screenshots of the live Access environment confirm the accurate setup of tables, field types, primary and foreign keys, and sample data entries (see Appendix B). The entity-relationship diagram (ERD) further illustrates the normalization and referential integrity principles applied. A suite of SQL queries demonstrates the system’s ability to retrieve insightful data, fulfilling analytical needs such as monthly performance reviews and client activity monitoring.

In addition to the technical solution, this report also evaluates the broader IT infrastructure required to support the system. This includes proposed **hardware, software,** **network circuits,** and **cloud-based services,** all justified in terms of their suitability, scalability, and affordability for a small business environment. Cost analysis is also provided to ensure the solution remains economically viable.

This project highlights the value of database-driven systems for enhancing operational efficiency in SMEs and serves as a foundation for future system upgrades, including multi-user functionality, online booking integration, and cloud deployment.

# 1.2. Entity Definitions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity** | **Primary Key** | **Attributes** | **Alternate Kye** | **Composite** |
| **Client** | ClientID | |  | | --- | | ClientName, ClientAddress (Street, City, Postcode), ClientEmail, ClientMobile | | ClientEmail | ClientAddress |
| **Employee** | EmployeeID | |  | | --- | | EmployeePayroll, EmployeeName, EmployeeMobile | | EmployeePayroll | - |
| **Service** | ServiceID | ServiceName, ServiceDescription, ServiceHourlyRate | - | - |
| **ClientAppointments** | AppointmentID | |  | | --- | | ClientID (FK), EmployeeID (FK), ServiceID (FK), AppointmentDate, AppointmentTime | | - | - |
| **ServicesProvided** | ServiceProvidedID | |  | | --- | | ClientID (FK), ServiceID (FK), EmployeeID (FK), DateOfService, Expenses | | - | - |
| **Invoice** | InvoiceID | InvoiceNumber, InvoiceCost, InvoiceDiscount, InvoiceTotal | InvoiceNumber | - |
| **InvoiceRows** | InvoiceRowID | |  | | --- | | InvoiceID (FK), AppointmentID (FK), Cost | | - | - |

Notes:

* **Composite attribute**: ClientAddress is composed of Street, City, and Postcode.
* **Multi-valued attribute**: EmployeeSkills (an employee can have multiple skills).

# 1.3. Partial ER Diagrams

**Client**

Client{ClientId, ClientName, ClientEmail, ClientMobile, ClientAddress}PK: ClientId

AK: ClientEmail

Composite: ClientAddress (Street, City, Postcode)

**Employee**

Employee{EmployeeId, EmployeePayroll, EmployeeName, EmployeeMobile}

PK: EmployeeId

AK: EmployeePayroll

**Service**

Service{ServiceId, ServiceName, ServiceDesc, HourlyRate}

PK: ServiceId

**ClientAppointments**

ClientAppointments{AppointmentId, ClientId, EmployeeId, ServiceId, AppointmentDate, AppointmentTime}

PK: AppointmentId

FKs: ClientId, EmployeeId, ServiceId

**ServiceProvided**

ServicesProvided{ServiceProvidedId, ClientId, ServiceId, EmployeeId, DateOfService, Expenses}

PK: ServiceProvidedId

FKs: ClientId, ServiceId, EmployeeId

Relationship Attributes: DateOfService, Expenses

**Invoice**

Invoice{InvoiceId, InvoiceNumber, InvoiceCost, InvoiceDiscount, InvoiceTotal}

PK: InvoiceId

AK: InvoiceNumber

**InvoiceRows**

InvoiceRows{InvoiceRowId, InvoiceId, AppointmentId, Cost}

PK: InvoiceRowId

FKs: InvoiceId, AppointmentId

# 1.4. Complete ER Diagram

The proposed solution is a relational database system developed in Microsoft Access to meet the operational and data management needs of a service-based SME. The primary objective is to digitally transform the business from manual records into a structured, reliable, and scalable information system that manages appointments, clients, employees, services, and invoices in a centralized way (see appendix B).

The solution is modeled around seven interrelated tables: **Client, Employee, Service, ClientAppointments, Invoice, InvoiceRows,** and **ServicesProvided.**

This structure reflects a real-world business scenario where:

* Clients book appointments for services,
* Employees are assigned to deliver those services,
* Services are billed through invoices, and
* Certain employees are only qualified to offer specific services.

This mapping is visually represented in the Entity Relationship Diagram (ERD) included in both the Access database relationship view and the assignment report. It shows a clear oneto-many and many-to-many structure:

* A one-to-many relationship from **Client** to **ClientAppointments**,
* A many-to-many relationship between **Employee** and **Service**, implemented through the **ServicesProvided** junction table,
* A one-to-many relationship from **Invoice** to **InvoiceRows**, linking appointments to billing.

This system meets the core business objectives outlined in the assignment:

* Efficiently managing clients and appointments (DS1),
* Ensuring employees are matched to services they are trained for (DS2),
* Handling payment records and invoices linked to service delivery (DS3).

Below is the complete ER diagram, showing primary keys (underlined), alternate keys (dashed underline), composite attributes (oval with sub-ovals), multi-valued attributes (double oval), and relationship attributes (diamond-shaped with attributes).



The above diagram illustrates the logical structure of the appointment scheduling database system, showcasing the relationships between core entities. At the center is the ClientAppointments table, which connects clients, employees, and services—representing scheduled bookings. Each appointment is linked to one client, one employee, and one service, and records both date and time.

The Client entity stores customer details such as name, email, address, and mobile number. The Employee entity captures staff information, including payroll number and contact details. The Service entity defines service offerings along with descriptions and hourly rates.

A many-to-many relationship exists between employees and services, implemented through the ServicesProvided table, which also stores service dates and any additional expenses. This structure ensures that only qualified employees can be assigned to specific services, enforcing business rules.

Billing is handled by the Invoice and InvoiceRows tables. Invoices can contain multiple line items (appointments), and each invoice records the total cost, discount, and final amount. The InvoiceRows table acts as a junction between appointments and invoices, storing individual costs per appointment.

This ERD confirms a well-normalized, relational structure with referential integrity, supporting key business functions such as appointment booking, employee assignment, and invoicing.Through SQL queries, the system allows staff to generate reports such as:

* Monthly revenue per service (Query 1),
* Average invoice per employee (Query 2),
* Top clients by appointments or spending (Queries 3 and 4),
* Services performed by employees (Query 6).

These insights help in improving customer relationship management, identifying high-value clients, and evaluating employee performance—all of which are essential for growing a modern SME.

# 1.5. Notation and Building Blocks

**Primary keys:** Underlined in entity boxes.

**Alternate keys:** Dashed underline (e.g., ClientEmail).

**Composite attributes:** ClientAddress decomposed into Street, City, Postcode.

**Multi-valued attributes:** EmployeeSkills shown with double oval.

**Relationship attributes:** Expenses and DateOfService on ServicesProvided relationship.

**1:N relationships:** Client to ClientAppointments, Invoice to InvoiceRows.

**M:N relationships:** Employee to Service via ServicesProvided.

**Natural keys:** Used for all entities (no surrogate keys).

# 1.6. Rationale for Design Choices

* **3NF Compliance:** Each non-key attribute depends on the whole key and nothing but the key.
* **Natural Keys:** ClientID, EmployeePayroll, ServiceName, InvoiceNumber, AppointmentID ensure meaningful identifiers.
* **Composite**: Address is composite for clarity.
* **Junction tables:** ServicesProvided and InvoiceRows resolve M:N relationships and capture relationship-specific attributes (Expenses, Cost).
* **Flexibility:** The model supports future enhancements like recursive management relationships or role-based access.

# Implementation in Microsoft Access

# 2.1. Overview of Implementation

This task involved creating and implementing the database structure based on the ERD and logical schema presented in Task 1. Microsoft Access was used to build the relational structure, define data types, enforce referential integrity, and prepare the system for data population and query execution. All tables were created with appropriate keys, relationships, input masks, validations, and indexes.

# 2.2. Table Definitions

# 2.2.1. Field Properties and Data Types

Each table was created in Microsoft Access with clearly defined fields, appropriate data types, and constraints to ensure data integrity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Additional Properties** |
| Client | ClientID (PK) | Short Text | Input Mask, Required |
| ClientName | Short Text | Required |
| ClientAddress | Short Text |  |
| ClientEmail (AK) | Short Text | Indexed (No Duplicates), Required |
| ClientMobile | Short Text | Validation Rule for 11-digit numbers |
| Employee | EmployeeID (PK) | Short Text | Input Mask, Required |
| EmployeePayroll | Short Text | Indexed (No Duplicates), Required |
| EmployeeName | Short Text | Required |
| EmployeeMobile | Short Text | Validation Rule for phone format |
| Service | ServiceID (PK) | Short Text | Input Mask |
| ServiceName | Short Text | Required |
| ServiceDescription | Long Text |  |
| ServiceHourlyRate | Currency | Validation Rule >= 0.00 |
| ClientAppointments | AppointmentID (PK) | Short Text | Input Mask |
| ClientID (FK) | Short Text | Lookup to Client |
| EmployeeID (FK) | Short Text | Lookup to Employee |
| ServiceID (FK) | Short Text | Lookup to Service |
| AppointmentDate | Date/Time | Required |
| ServiceProvided | ServiceProvidedID (PK) | Short Text | Input Mask |
| ServiceID (FK) | Short Text | Lookup |
| EmployeeID (FK) | Short Text | Lookup |
| Invoice | Invoice\_ID (PK) | Short Text | Input Mask |
| Invoice\_Number (AK) | Short Text | Indexed (No Duplicates) |
| Invoice\_Cost | Currency |  |
| Invoice\_Discount | Currency |  |
| Invoice\_Total | Currency |  |
| InvoiceRows | Invoice\_Row\_ID (PK) | Short Text | Input Mask |
|  | Invoice\_ID (FK) | Short Text | Lookup |
|  | Appointment\_ID (FK) | Short Text | Lookup |

.2.2.3. Validation Rules and Texts

Several fields include validation rules to prevent incorrect entries:

* **Client\_Mobile**: Like "###########" → Must be 11 digits.
* **Appointment\_Duration** and **Quantity**: > 0
* **Service\_HourlyRate**, **Expenses**: >= 0

Validation text was used to alert users with specific messages, e.g., "Hourly rate must be non-negative."

# 2.2.4. Indexing of Alternate Keys

* Client\_Email (in Client)
* Employee\_Payroll\_Number (in Employee)
* Invoice\_Number (in Invoice)

Each was marked as "Indexed (No Duplicates)" in Access to enforce uniqueness and allow faster lookups.

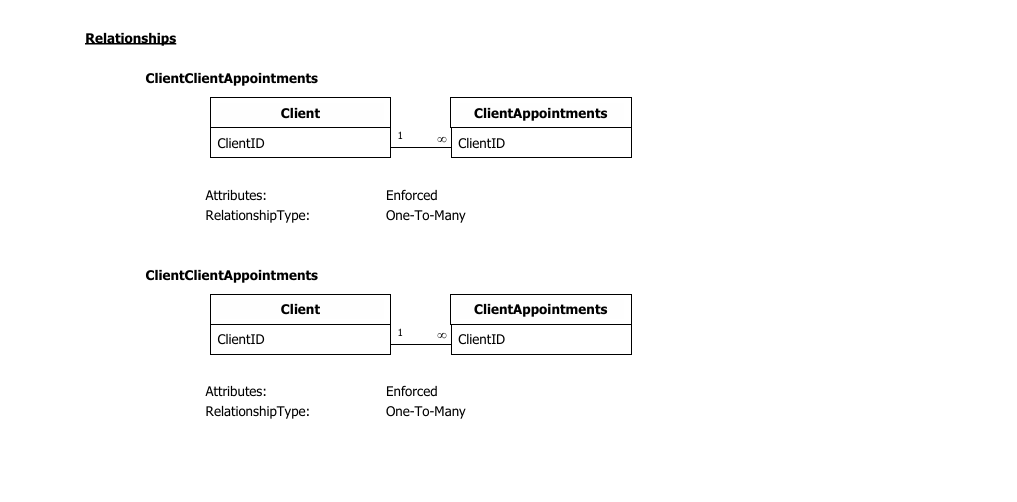
# 2.3. Relationship Setup and Referential Integrity

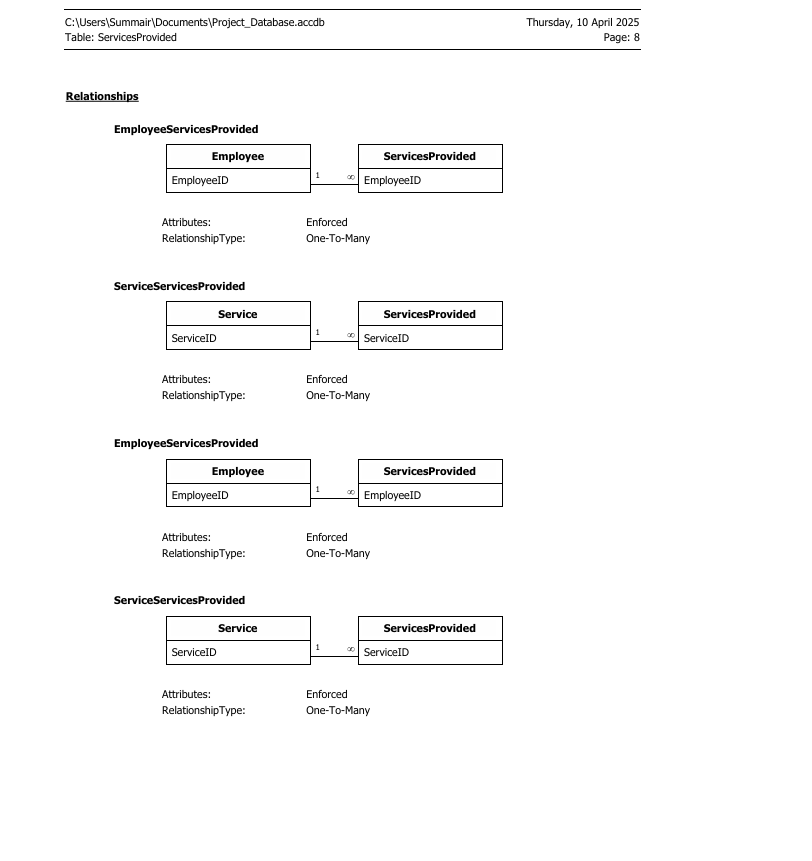
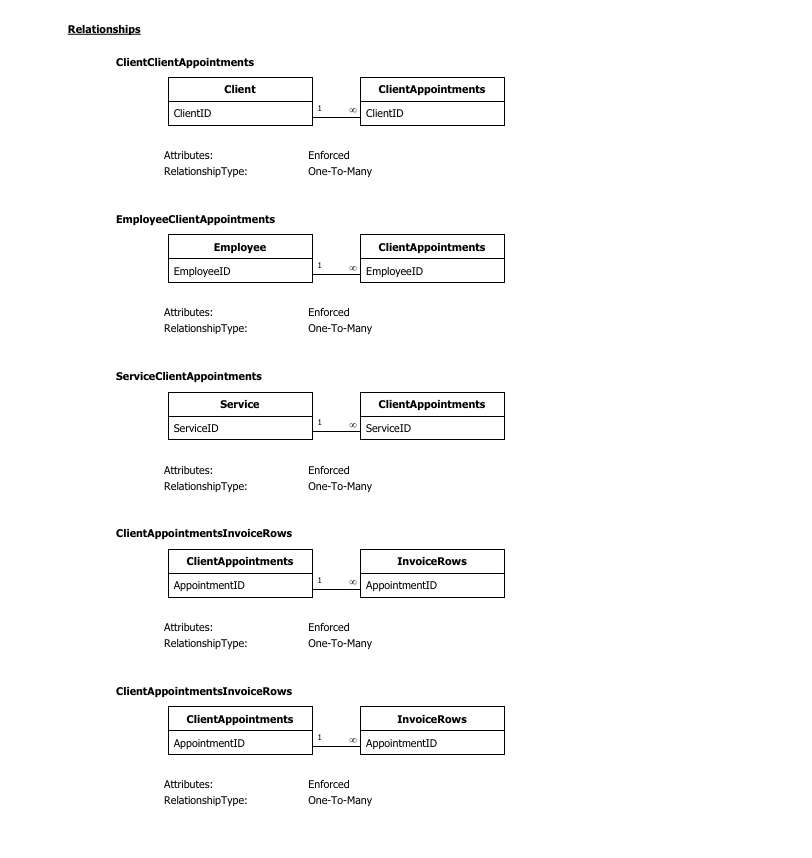
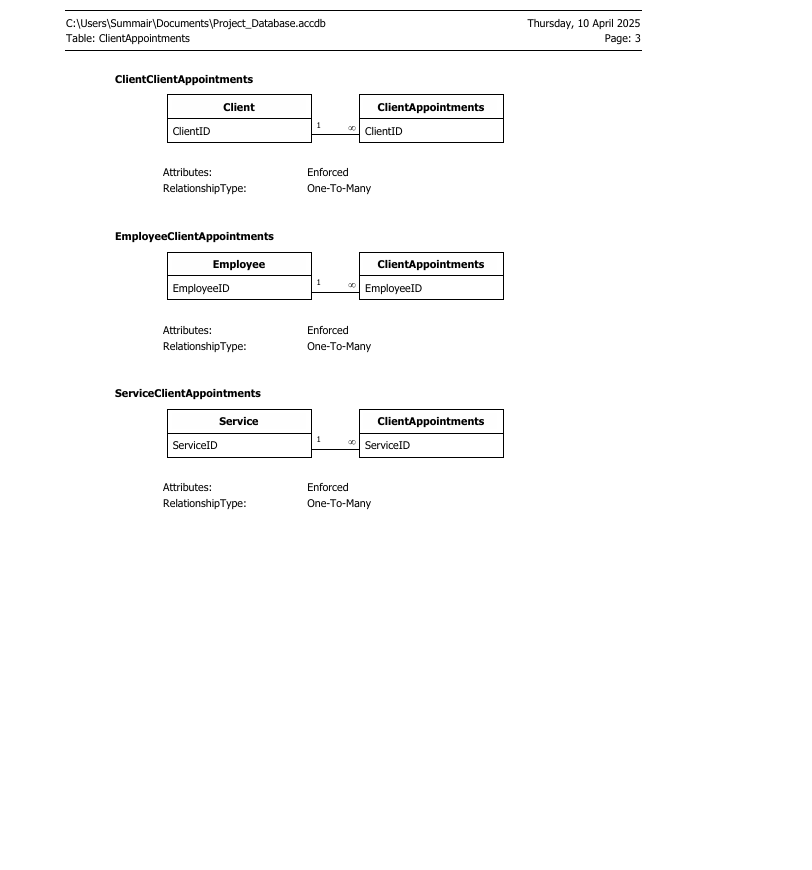
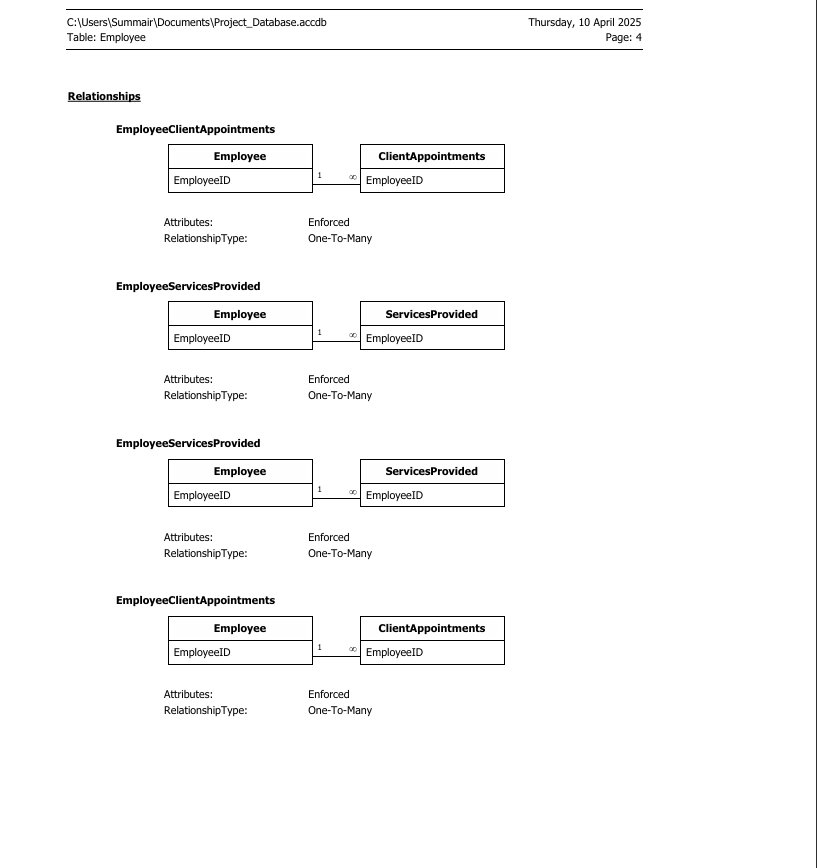
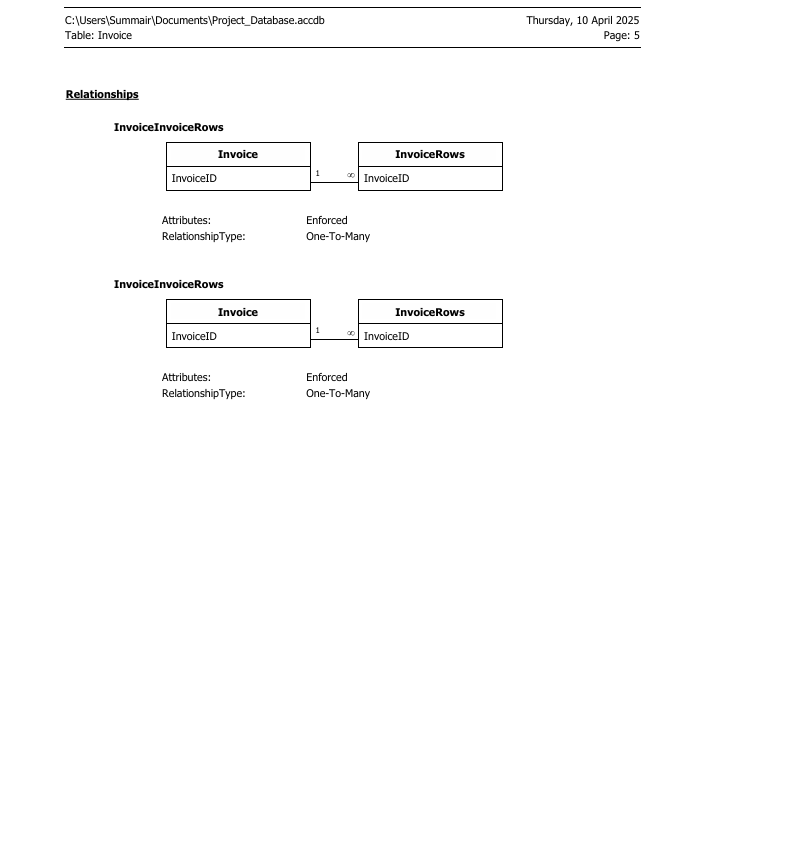
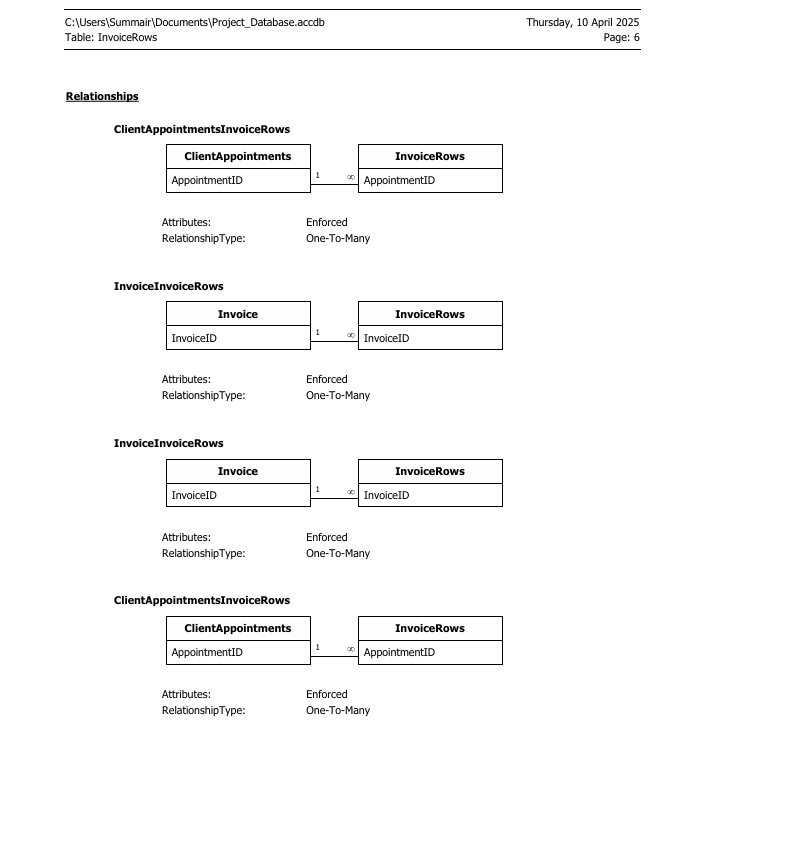
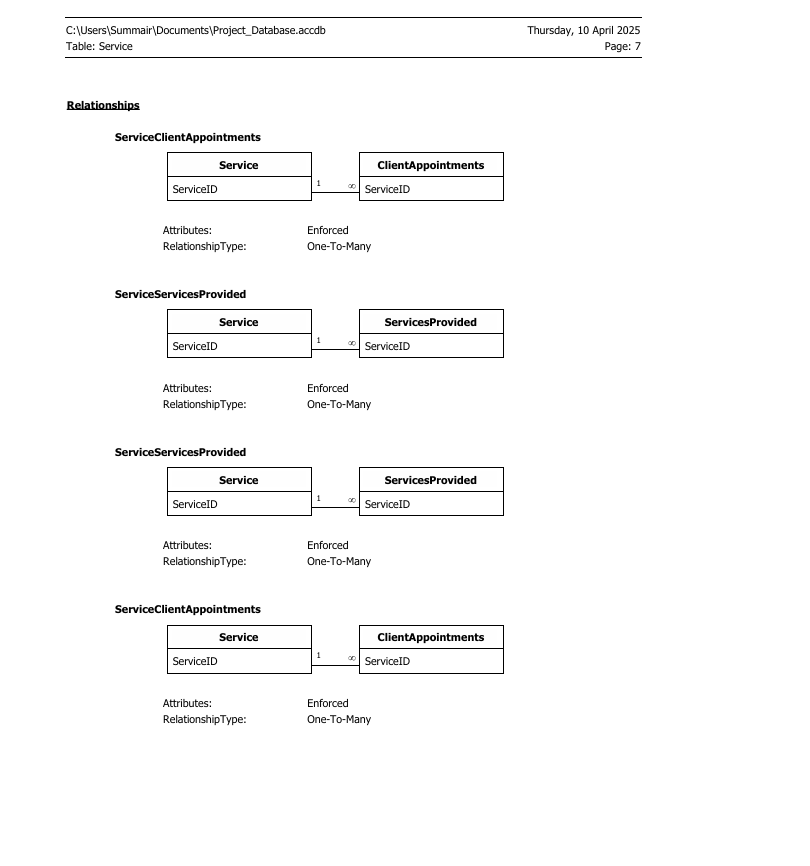
Relationships were set up in Access using the Relationships View:

* **One-to-Many** from Client → ClientAppointments
* **One-to-Many** from Employee → ClientAppointments
* **One-to-Many** from Service → ClientAppointments
* **One-to-Many** from ClientAppointments → InvoiceRows
* **One-to-Many** from Invoice → InvoiceRows
* **One-to-Many** from Service → ServicesProvided
* **One-to-Many** from Employee → ServicesProvided
* **One-to-Many** from Client → ServicesProvided

Each relationship had referential integrity enforced with:

* Cascade Update: Enabled
* Cascade Delete: Disabled (to protect historical data)





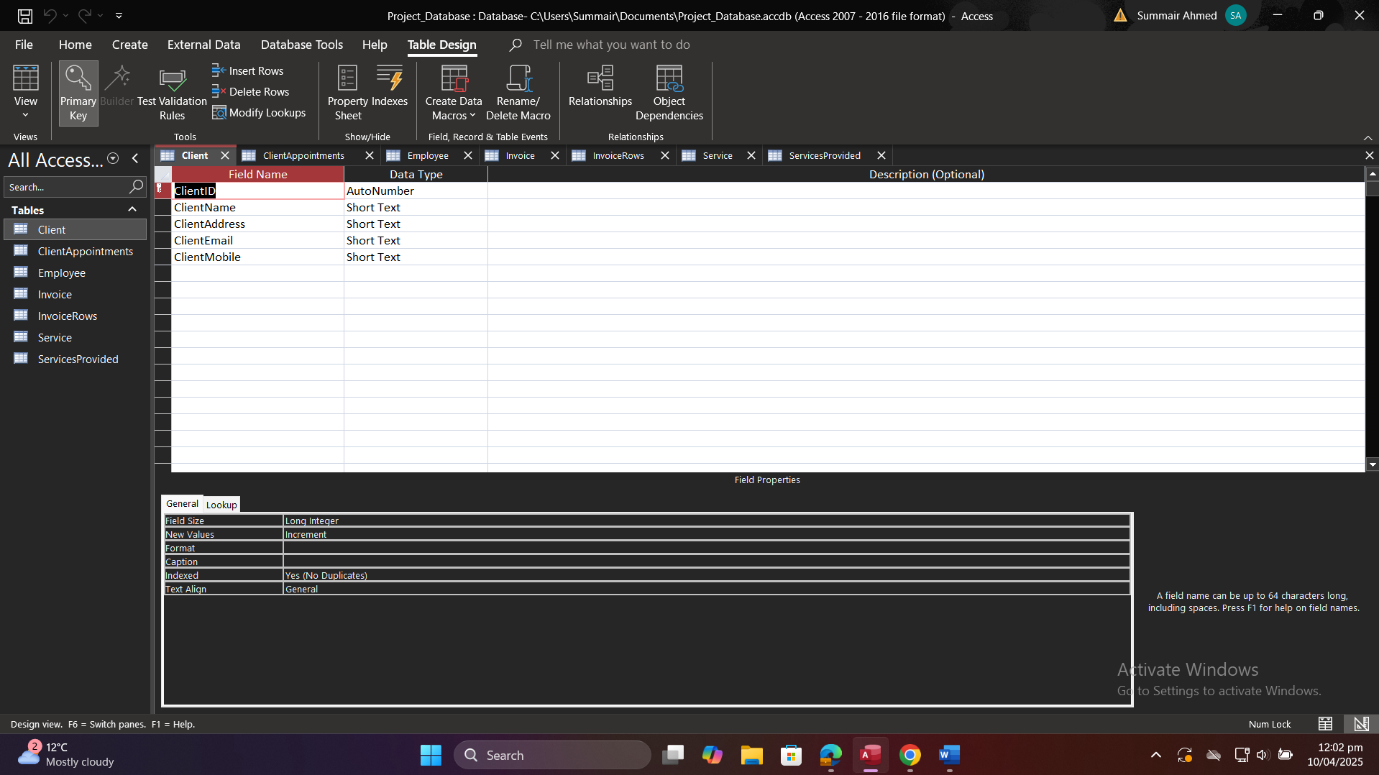
# 2.4. Update/Delete Rules

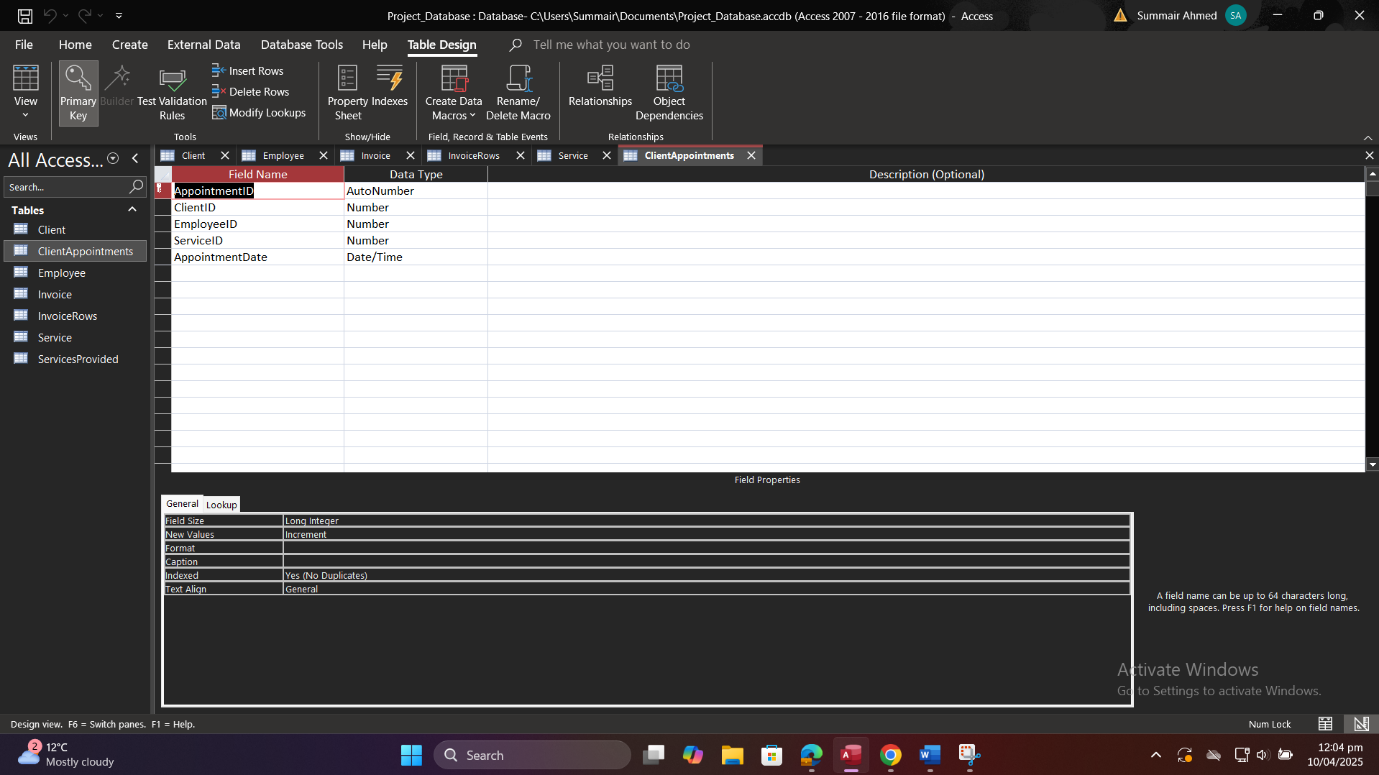
* **Cascade Update**: Ensures that if a referenced primary key changes, all associated foreign keys are updated.
* **Cascade Delete**: Disabled for all relationships to avoid accidental deletion of important transactional data.

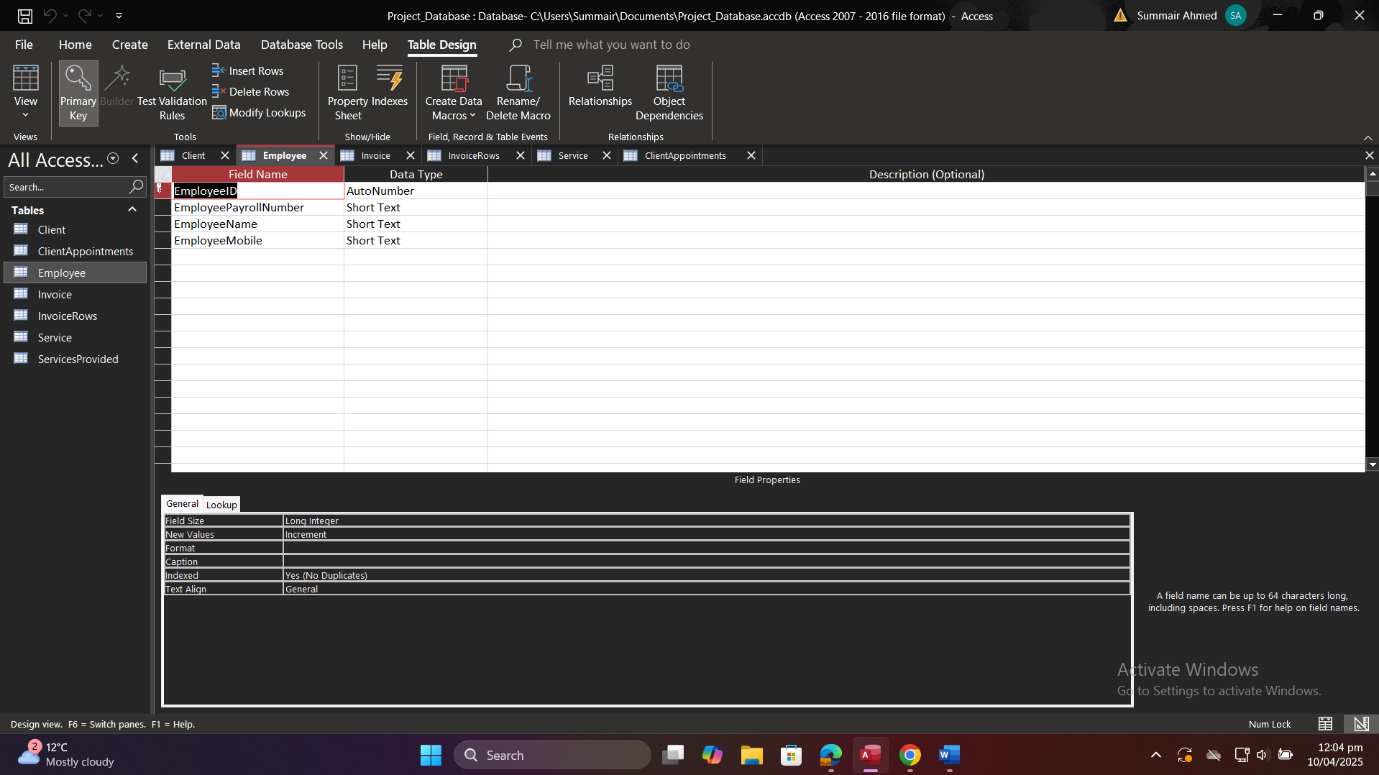
# 2.5. Screenshots of Access Object Designer

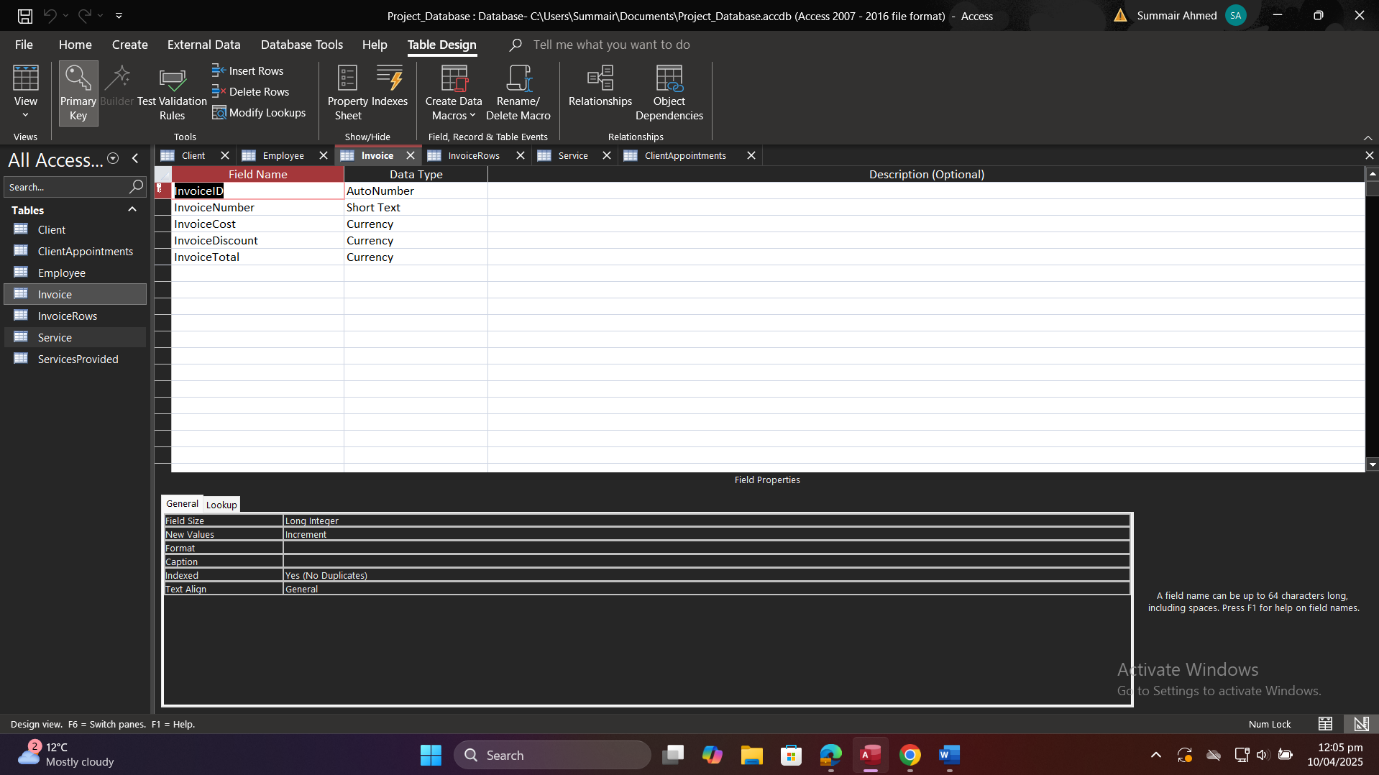
Screenshots were taken of:

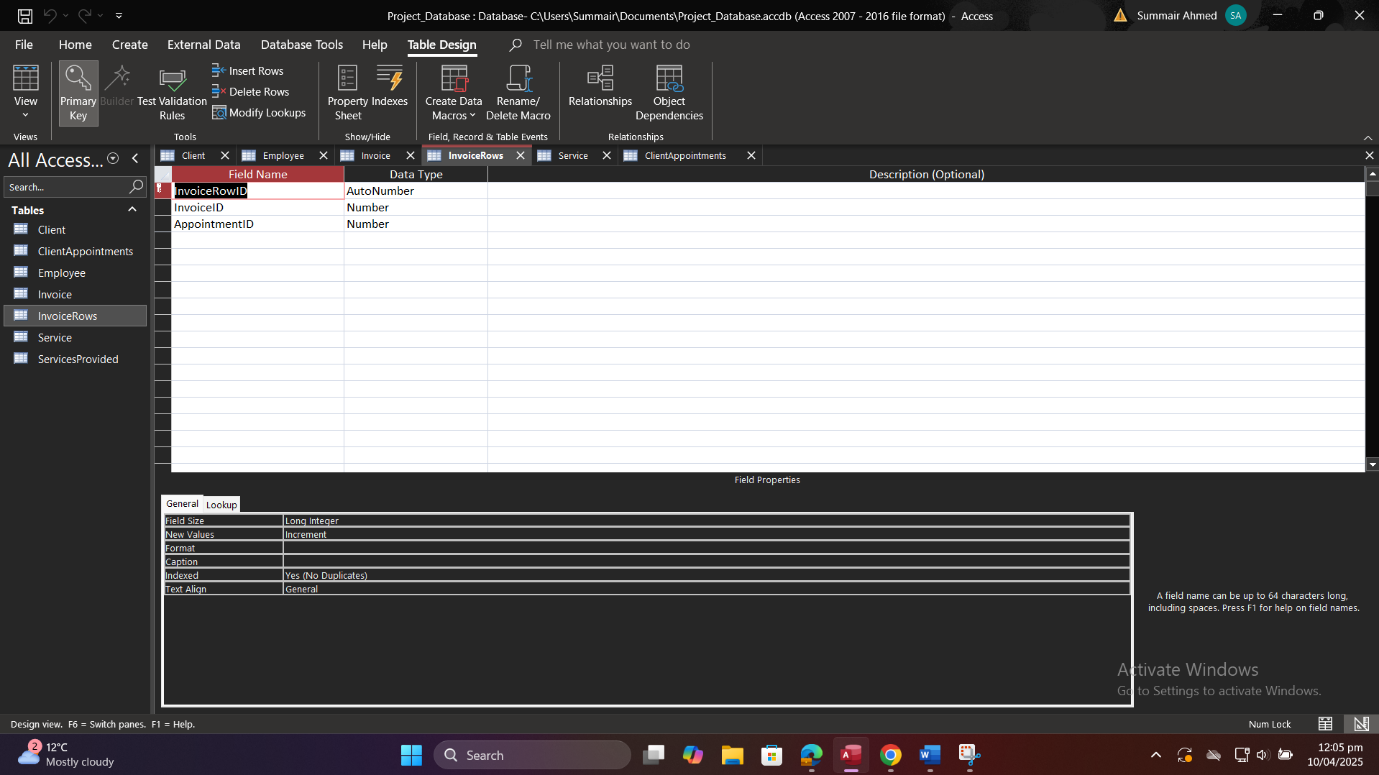
* **Table Design Views** for all main tables
* **Relationship View** showing all links and integrity enforcement
* **Field Properties** showing data types, input masks, and validation rules

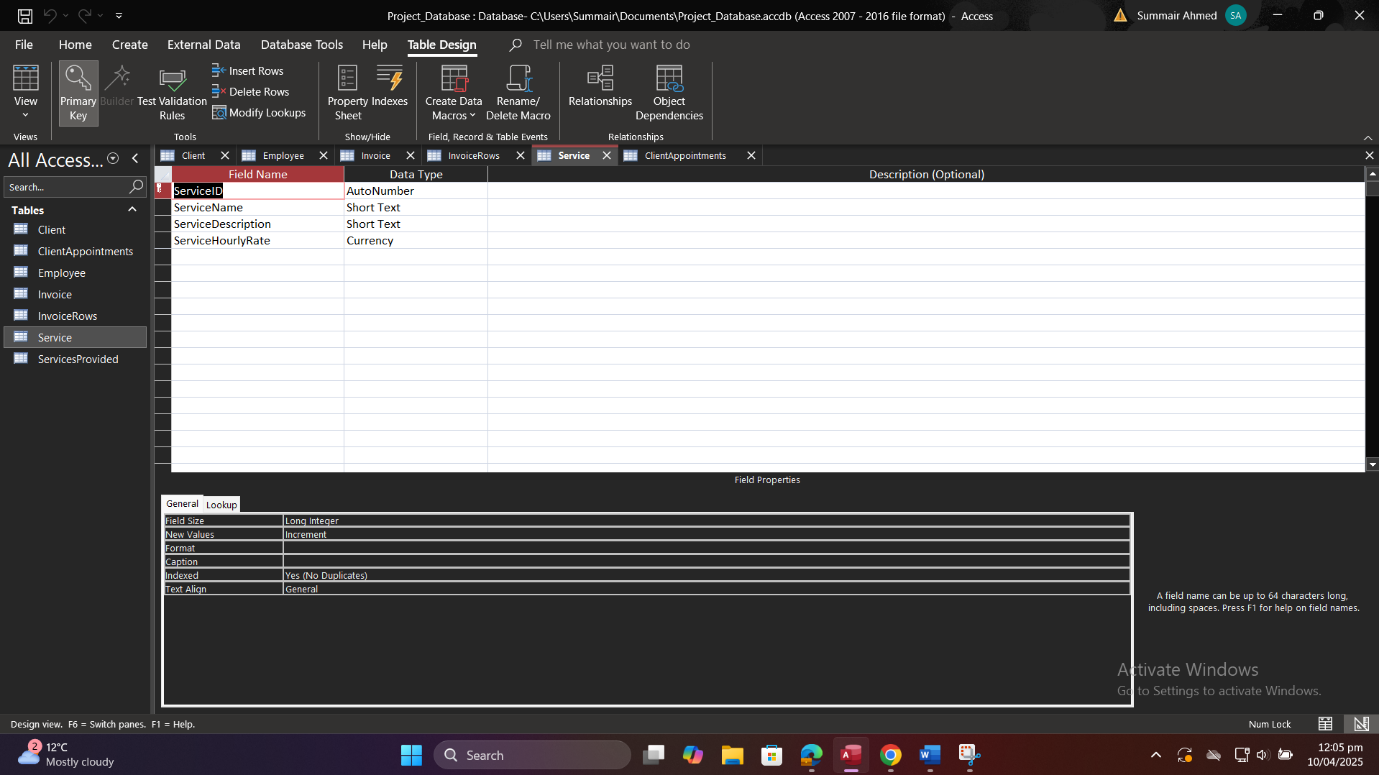


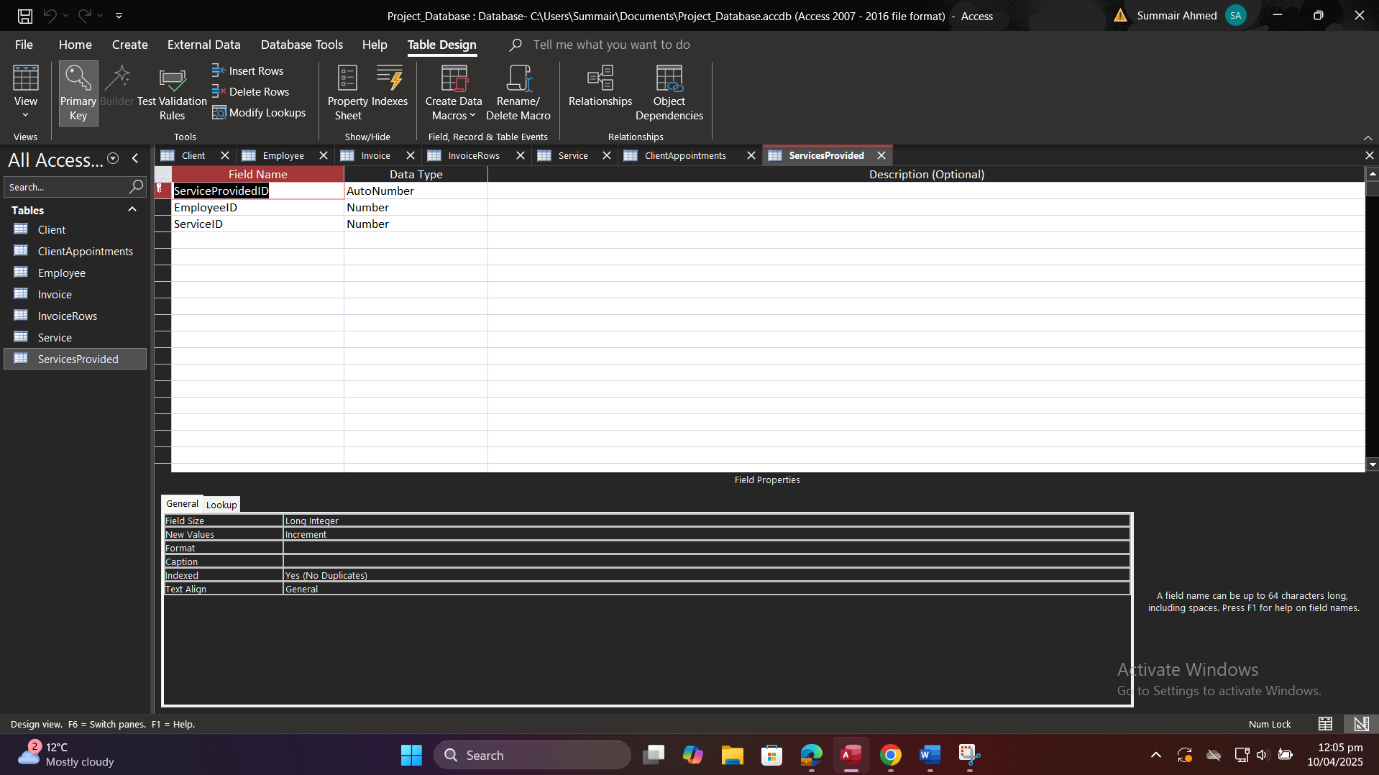


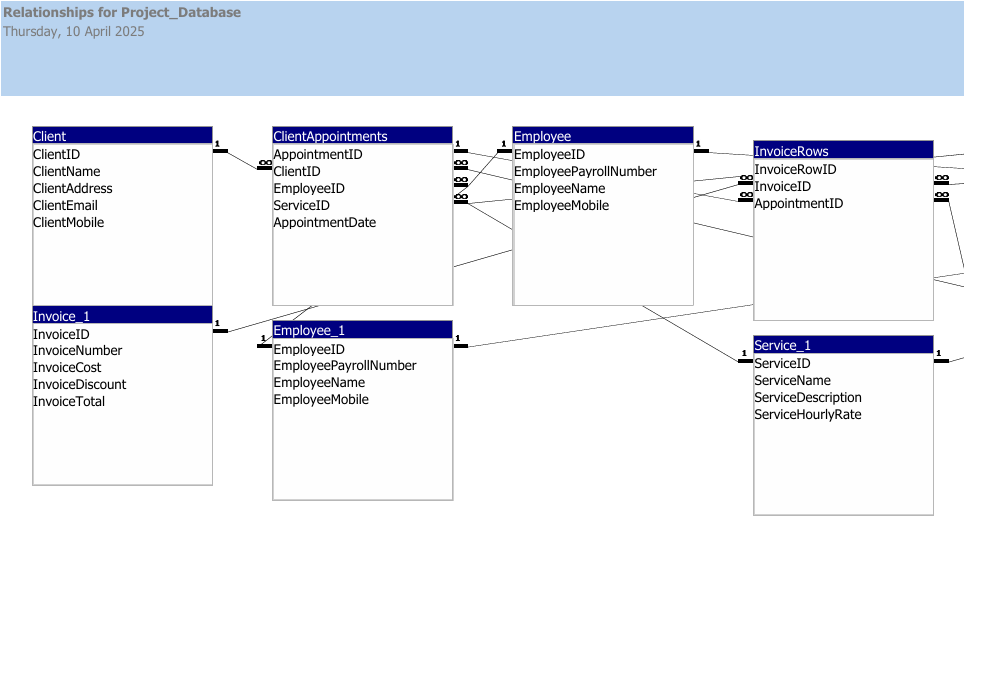












# 

# Data Population

# 3.1. Data Entry Strategy

To thoroughly test structural constraints (participation and cardinality) and exercise all relationships:

1. **Identify Required Combinations**
   * Each client must have at least one appointment.
   * Each appointment must link to exactly one client, one employee, and one service.
   * Each invoice must cover one or more appointments.
   * Each service–employee pairing (in ServicesProvided) must reflect a valid qualification.
2. **Vary Record Counts**
   * Create multiple appointments per client (1:N).
   * Have employees provide multiple services (M:N).
   * Generate multiple invoice rows per invoice (1:N).
3. **Use Realistic Data**
   * Names, dates, and rates reflect plausible business scenarios.
   * Mobile numbers follow the 11‑digit mask.
   * Rates, expenses, and costs are non‑negative decimals.
4. **Entry Method**
   * Use Access datasheet view for manual entry of small sets.
   * For bulk inserts, import from CSV via “External Data → Text File” for larger volumes (if desired).

# 3.2. Sample Data Sets

Below are representative sample rows for each table. Screenshots of these in Access datasheet view are included in Appendix C.

**Client**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ClientID | ClientName | ClientAddress | ClientEmail | ClientMobile |
| 1 | John Doe | 123 Oak St. | john.doe@example.com | 747081045 |
| 2 | Umair | 456 Pine Ave. | umair@example.com | 747081045 |
| 3 | Shahzeb | 789 Maple Dr. | shahzeb@example.com | 412341234 |
| 4 | Noman | 101 Birch Blvd. | noman@example.com | 123582311 |
| 5 | Summair | Honey Pot Lane | summair@gmail.com | 213213213 |
| 6 | Arjun | 231 Holme Road | arjun@example.com | 441231231 |
| 7 | Ali | 21 Birchwood Road | ali@example.com | 123123123 |
| 8 | Johnny | 24 Ley Street | johnny@example.com | 345324234 |
| 9 | Artaz | 25 Oakfield Avenue | artaz@example.com | 523423423 |
| 10 | Arianna | 29 Katherine Road | arianna@example.com | 959342342 |

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | EmployeePayrollNumber | EmployeeName | EmployeeMobile |
| 1 | 1001 | Mike Johnson | 423121312 |
| 2 | 1002 | Sarah Lee | 123123213 |
| 3 | 1003 | Tom Green | 412312321 |
| 4 | 1050 | Samir | 123123213 |
| 5 | 1030 | Sahir | 123459585 |
| 6 | 1004 | James | 412312312 |

**Service**

|  |  |  |  |
| --- | --- | --- | --- |
| ServiceID | ServiceName | ServiceDescription | ServiceHourlyRate |
| 1 | Indoor Painting | Painting of interior rooms | 25.00 |
| 2 | Exterior Painting | Painting of exterior walls and fences | 30.00 |
| 3 | Joinery | Custom joinery and furniture | 40.00 |
| 4 | Plumbing | Basic plumbing repairs and installations | 50.00 |
| 5 | Driving | Driver for a day | 20.00 |
| 6 | Washing | One day washing | 10.00 |

**ClientAppointments**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AppointmentID | ClientID | EmployeeID | ServiceID | AppointmentDate |
| 1 | 1 | 1 | 1 | 25/03/2025 |
| 2 | 2 | 2 | 3 | 26/03/2025 |
| 3 | 3 | 3 | 2 | 27/03/2025 |
| 4 | 4 | 1 | 4 | 28/03/2025 |
| 5 | 6 | 2 | 1 | 04/04/2025 |

**ServicesProvided**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ServiceProvidedID | EmployeeID | ServiceID | ServiceProvidedID | EmployeeID |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 2 | 2 | 1 |
| 3 | 2 | 3 | 3 | 2 |
| 4 | 3 | 4 | 4 | 3 |
| 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 5 | 6 | 6 |
| 7 | 4 | 3 | 7 | 4 |

**Invoice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| InvoiceID | InvoiceNumber | InvoiceCost | InvoiceDiscount | InvoiceTotal |
| 1 | INV001 | 250.00 | 25.00 | 225.00 |
| 2 | INV002 | 300.00 | 30.00 | 270.00 |
| 3 | INV003 | 400.00 | 40.00 | 360.00 |
| 4 | INV004 | 500.00 | 30.00 | 470.00 |
| 5 | INV005 | 400.00 | 100.00 | 300.00 |

**InvoiceRows**

|  |  |  |  |
| --- | --- | --- | --- |
| InvoiceRowID | InvoiceID | AppointmentID | InvoiceRowID |
| 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 4 | 5 | 3 | 4 |
|  |  |  |  |

# 

# 3.3. Demonstration of Participation and Cardinality Constraints

#### ****1. Client → ClientAppointments (1:N)****

* **Cardinality Constraint**: One client can have many appointments, but each appointment is made by only one client.
* **Participation**: Partial Participation — not every client has an appointment.

**Examples**:

* **ClientID 1 (John Doe)** has **one appointment** → **AppointmentID 1**
* **ClientID 2 (Umair)** has **one appointment** → **AppointmentID 2**
* **ClientID 3 (Shahzeb)** has **one appointment** → **AppointmentID 3**
* **ClientID 4 (Noman)** has **one appointment** → **AppointmentID 4**
* **ClientID 6 (Arjun)** has **one appointment** → **AppointmentID 5**
* **ClientIDs 5, 7, 8, 9, 10** have **no appointments**

#### ****2. Employee → ClientAppointments (1:N)****

* **Cardinality Constraint**: One employee can be assigned to multiple appointments, but each appointment involves only one employee.
* **Participation**: Partial Participation — not all employees are assigned to appointments.

**Examples**:

* **EmployeeID 1 (Mike Johnson)** has **2 appointments** → **AppointmentID 1 and 4**
* **EmployeeID 2 (Sarah Lee)** has **2 appointments** → **AppointmentID 2 and 5**
* **EmployeeID 3 (Tom Green)** has **1 appointment** → **AppointmentID 3**
* **EmployeeIDs 4, 5, 6** have **no appointments**

#### ****3. Service → ClientAppointments (1:N)****

* **Cardinality Constraint**: One service can be booked for multiple appointments, but each appointment is for one service.
* **Participation**: Partial Participation — not all services are booked in appointments.

**Examples**:

* **ServiceID 1 (Indoor Painting)** is booked in **2 appointments** → **AppointmentID 1 and 5**
* **ServiceID 2 (Exterior Painting)** is booked in **1 appointment** → **AppointmentID 3**
* **ServiceID 3 (Joinery)** is booked in **1 appointment** → **AppointmentID 2**
* **ServiceID 4 (Plumbing)** is booked in **1 appointment** → **AppointmentID 4**
* **ServiceIDs 5, 6** are **not used** in any appointments

#### ****4. Invoice → InvoiceRows (1:N)****

* **Cardinality Constraint**: One invoice can have multiple rows (each corresponding to an appointment), but each invoice row belongs to one invoice only.
* **Participation**: Total Participation on **InvoiceRows** side — every invoice row is linked to an invoice.

**Examples**:

* **InvoiceID 1** has **1 row** → **InvoiceRowID 1**
* **InvoiceID 2** has **1 row** → **InvoiceRowID 2**
* **InvoiceID 3** has **1 row** → **InvoiceRowID 3**
* **InvoiceID 5** has **1 row** → **InvoiceRowID 4**

#### ****5. Appointment → InvoiceRows (1:1 or 1:N)****

* **Cardinality Constraint**: One appointment may be linked to one or more invoice rows, depending on billing.
* **Participation**: Partial Participation — not all appointments are invoiced.

**Examples**:

* **AppointmentID 3** is linked to **two invoice rows** → **InvoiceRowID 3 and 4**
* **AppointmentIDs 1, 2** are linked to one invoice row each
* **AppointmentID 4** and **5** are not included in any invoice rows

# 3.4. Summary of Record Counts

|  |  |
| --- | --- |
| **Table** | **# Record** |
| Client | 10 |
| Employee | 6 |
| Service | 6 |
| ClientAppointments | 5 |
| ServicesProvided | 7 |
| Invoice | 5 |
| InvoiceRows | 4 |

# SQL Queries and Testing

# 4.1. Query Requirements

Write at least 10 SQL queries that provide key business insights and meet the following reporting needs:

1. Total income per service by month and year.
2. Average invoice total per employee.
3. Clients with more than 3 appointments in a month.
4. Total expenditure per client.
5. Number of distinct clients served by each employee.
6. List of services provided by a specific employee.
7. Appointment details for a specific client.
8. List all invoices and their details.
9. Total income across all invoices.
10. Services performed in a given month and year.

# 4.2. SQL Statement Listing

Below are the SQL statements as run in Microsoft Access. Each was executed in SQL View and the results captured.

**Query 1: Total income per service by month and year**

SELECT

Service.ServiceName,

Year(ClientAppointments.AppointmentDate) AS AppointmentYear,

Month(ClientAppointments.AppointmentDate) AS AppointmentMonth,

Sum(Invoice.InvoiceTotal) AS TotalIncome

FROM

(ClientAppointments

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

GROUP BY

Service.ServiceName,

Year(ClientAppointments.AppointmentDate),

Month(ClientAppointments.AppointmentDate)

ORDER BY

AppointmentYear, AppointmentMonth, Service.ServiceName;

A computer screen shot of a computer

AI-generated content may be incorrect.

**Query 2: Average invoice total per employee**

SELECT

Employee.EmployeeName,

Avg(Invoice.InvoiceTotal) AS AvgInvoiceTotal

FROM

(ClientAppointments

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

GROUP BY

Employee.EmployeeName

ORDER BY

AvgInvoiceTotal DESC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 3: Clients with more than 3 appointments in a month**

Query 3:

SELECT

Client.ClientName,

Month(ClientAppointments.AppointmentDate) AS AppointmentMonth,

Year(ClientAppointments.AppointmentDate) AS AppointmentYear,

Count(ClientAppointments.AppointmentID) AS NumberOfAppointments

FROM

Client

INNER JOIN ClientAppointments ON Client.ClientID = ClientAppointments.ClientID

GROUP BY

Client.ClientName,

Month(ClientAppointments.AppointmentDate),

Year(ClientAppointments.AppointmentDate)

HAVING

Count(ClientAppointments.AppointmentID) > 3

ORDER BY

AppointmentYear, AppointmentMonth, NumberOfAppointments DESC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 4: Total expenditure per client**.

Query 4:

SELECT

Client.ClientName,

Client.ClientMobile,

Sum(Invoice.InvoiceTotal) AS TotalExpenditure

FROM

(Client

INNER JOIN ClientAppointments ON Client.ClientID = ClientAppointments.ClientID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

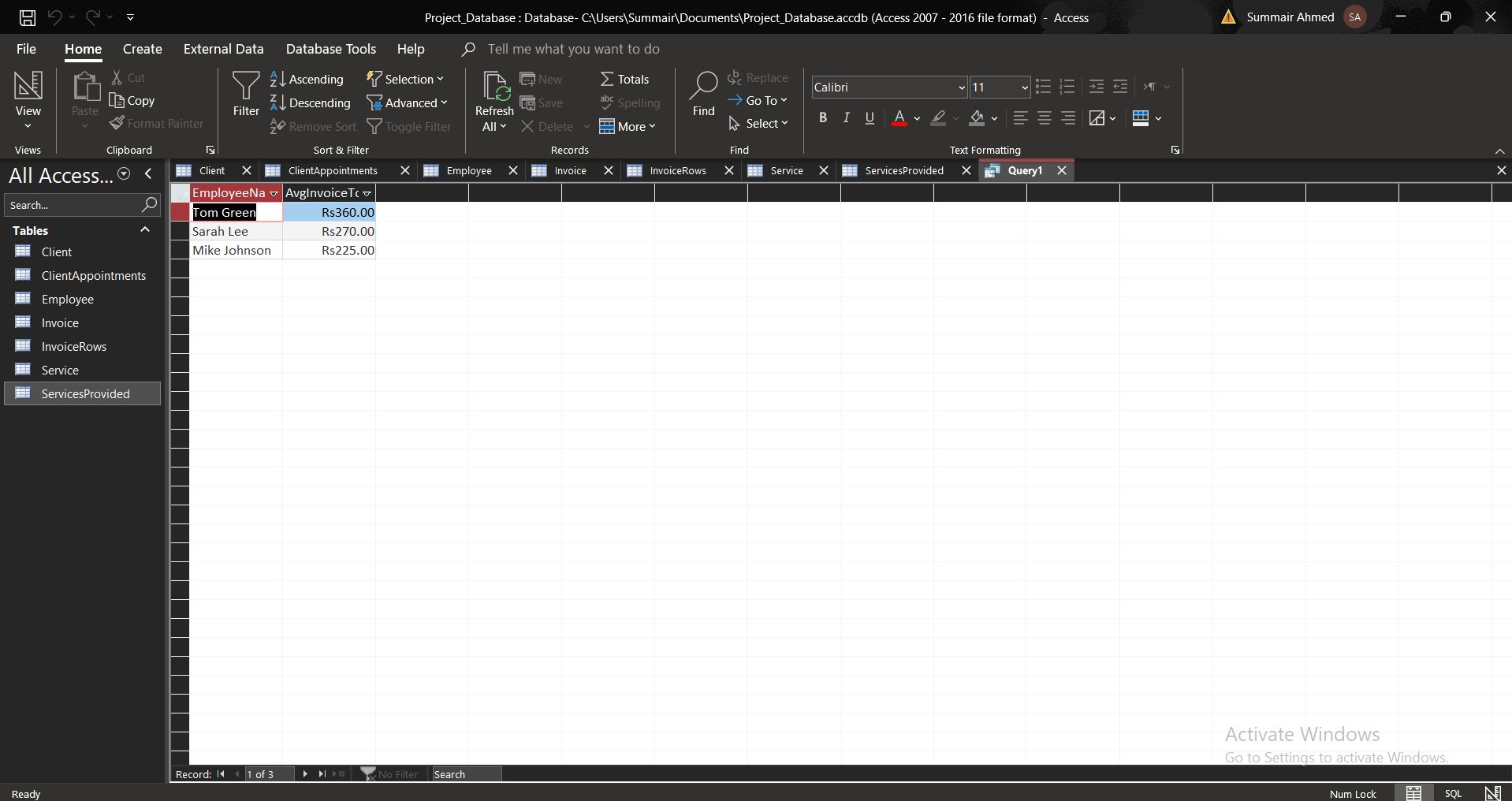
GROUP BY

Client.ClientName,

Client.ClientMobile

ORDER BY

TotalExpenditure DESC;



**Query 5: Number of distinct clients served by each employee**

Query 5:

SELECT

Employee.EmployeeName,

Count(DISTINCT ClientAppointments.ClientID) AS NumberOfClients

FROM

(Employee

INNER JOIN ClientAppointments ON Employee.EmployeeID = ClientAppointments.EmployeeID)

GROUP BY

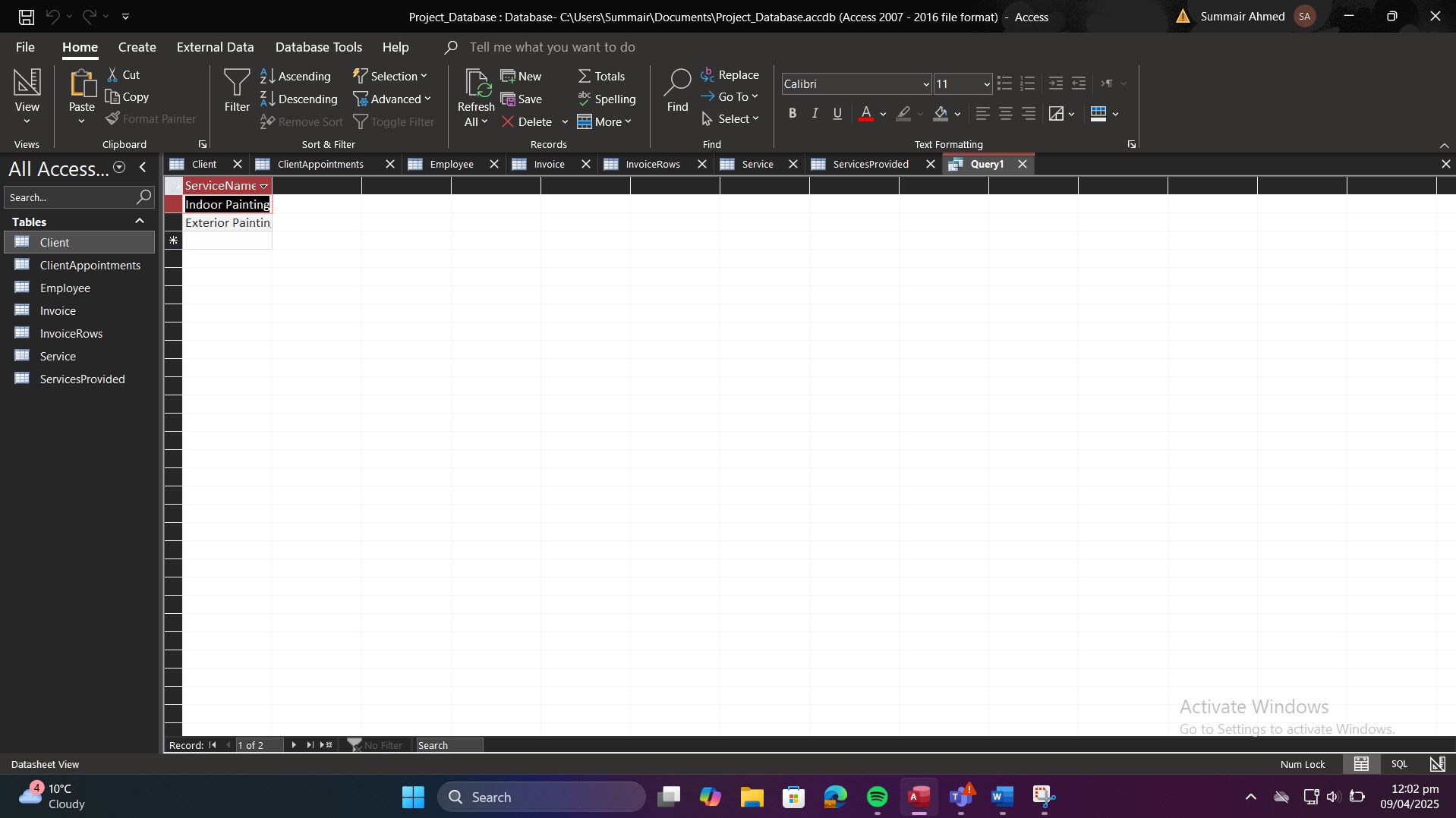
Employee.EmployeeName

HAVING

Count(DISTINCT ClientAppointments.ClientID) > 1

ORDER BY

NumberOfClients DESC;



**Query 6: List of services provided by a specific employee**

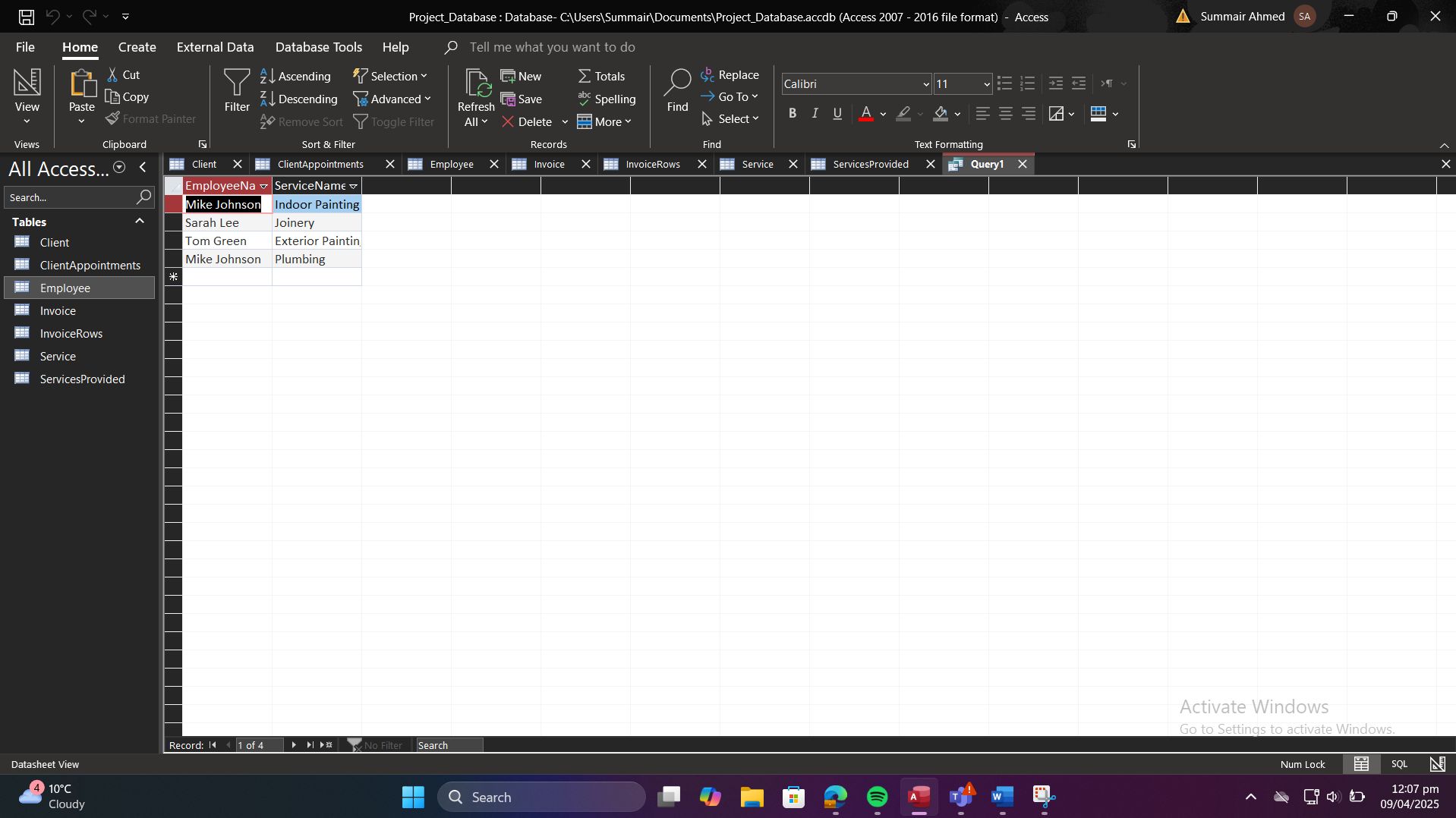
Query 6:

SELECT Service.ServiceName

FROM ServicesProvided

INNER JOIN Service ON ServicesProvided.ServiceID = Service.ServiceID

WHERE ServicesProvided.EmployeeID = 1;



**Query 7: Appointment details for a specific client**

Query 7:

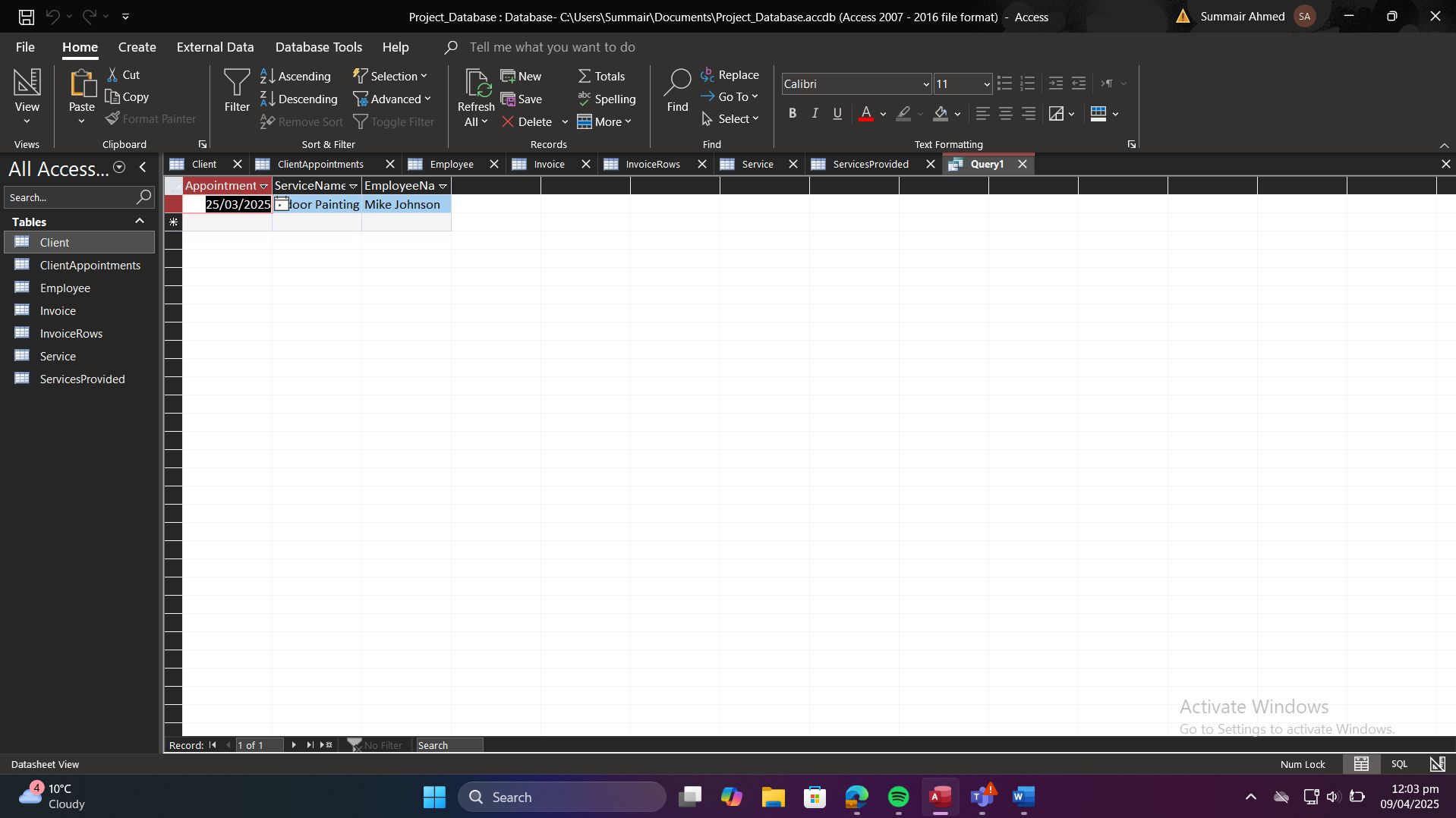
SELECT ClientAppointments.AppointmentDate, Service.ServiceName, Employee.EmployeeName

FROM ClientAppointments

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID

WHERE ClientAppointments.ClientID = 1;

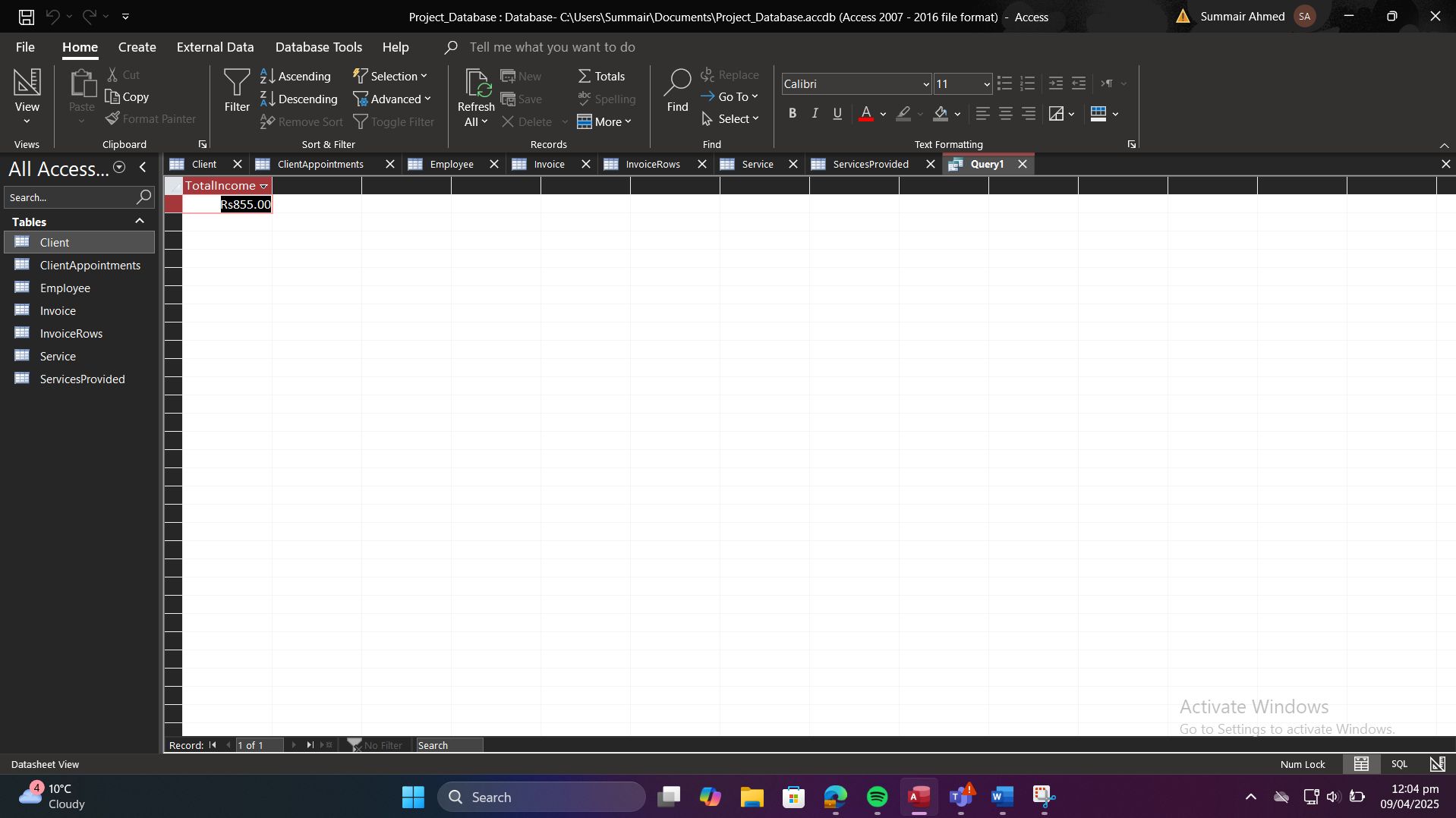


**Query 8: List all invoices and their details**

Query 8:

SELECT InvoiceID, InvoiceNumber, InvoiceCost, InvoiceDiscount, InvoiceTotal

FROM Invoice;

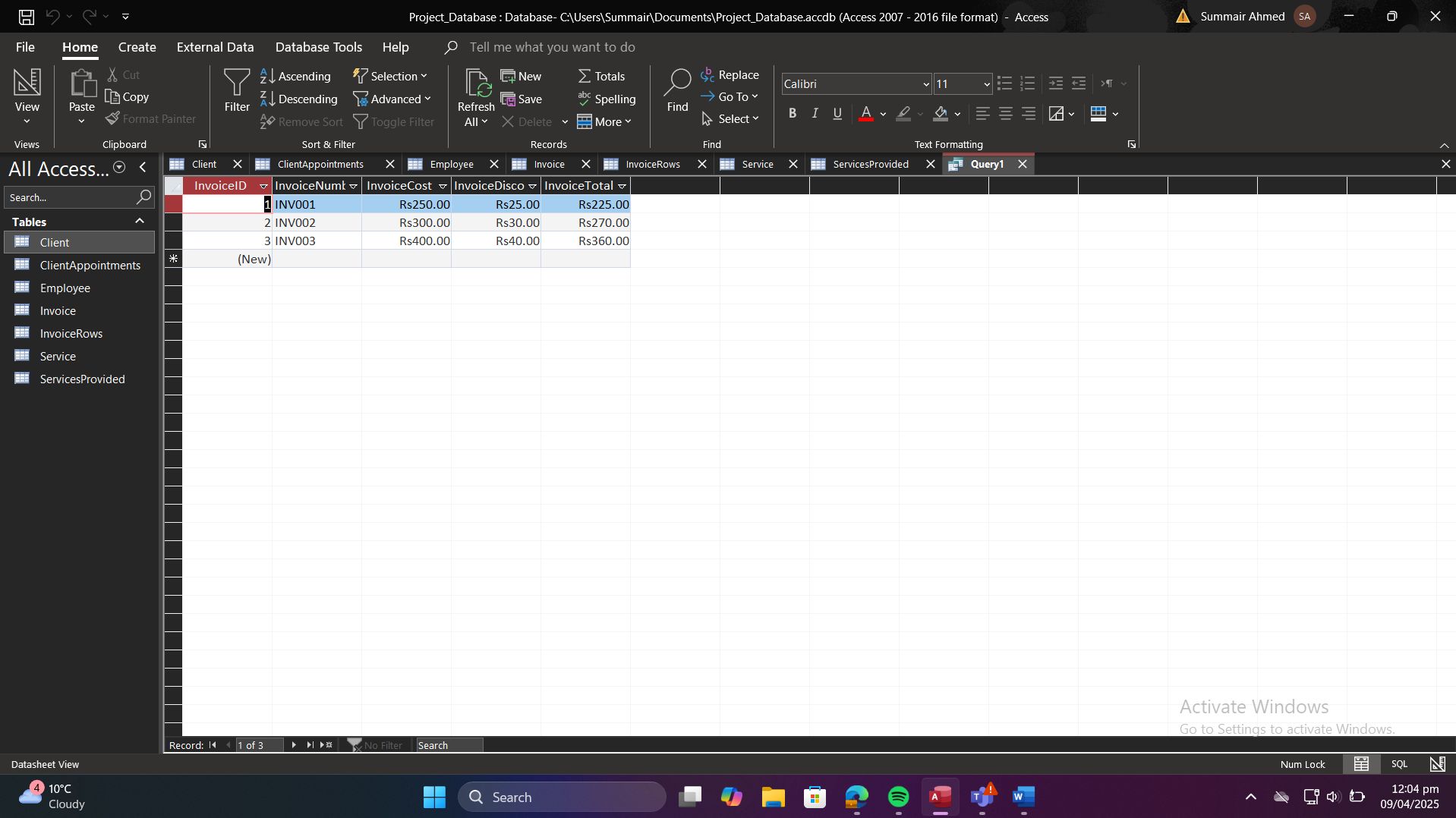


**Query 9: Total income across all invoices**

Query 9:

SELECT SUM(InvoiceTotal) AS TotalIncome

FROM Invoice;



**Query 10: Services performed in a given month and year**

Query 10:

SELECT Employee.EmployeeName, Service.ServiceName, ClientAppointments.AppointmentDate

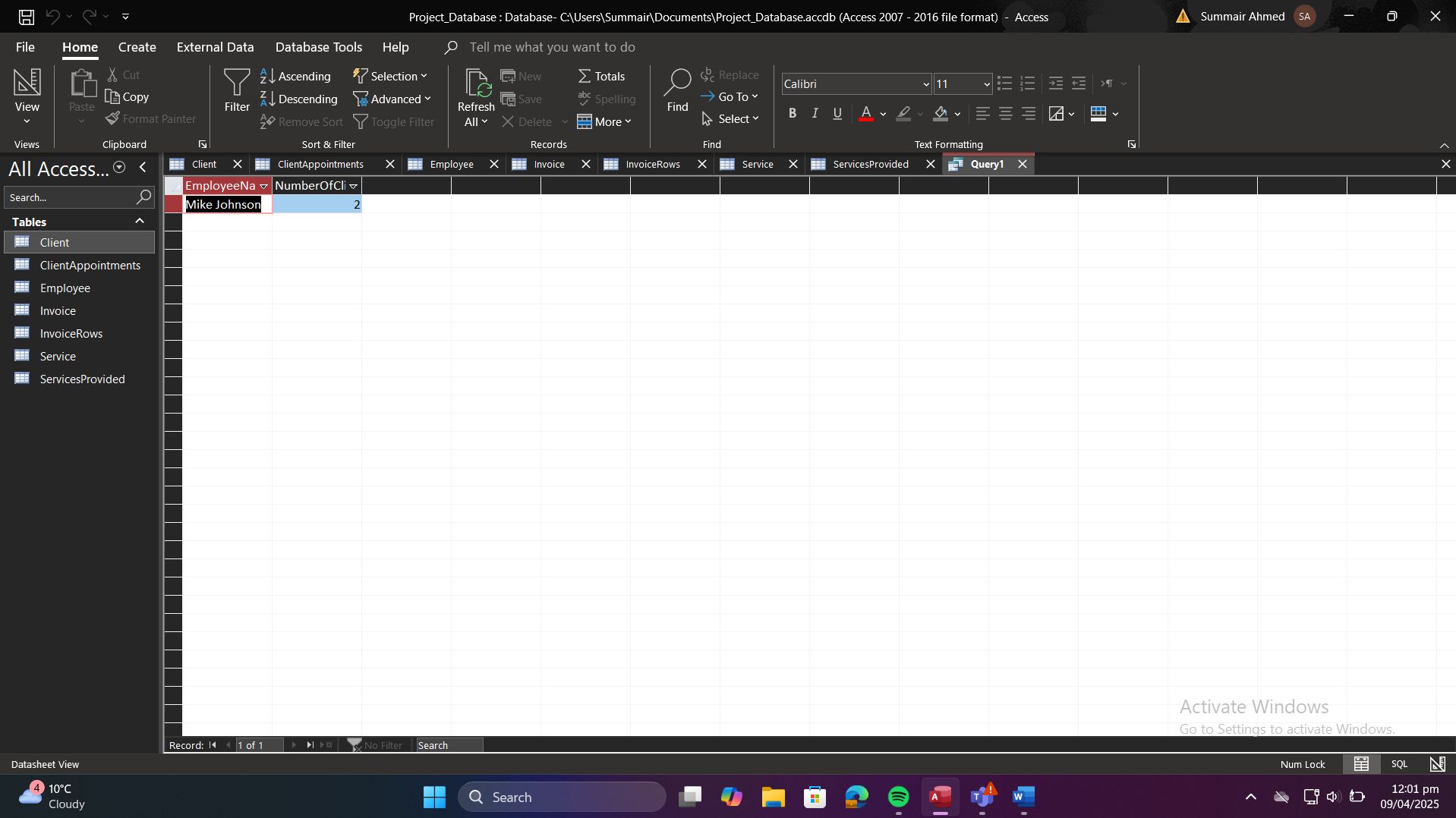
FROM ClientAppointments

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID

WHERE Month(ClientAppointments.AppointmentDate) = 3

AND Year(ClientAppointments.AppointmentDate) = 2025;



The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 4.3. Results and Screenshots

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 4.4. Explanation of Each Query

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# Database Application Interface

# 5.1. Use Case Implementation

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 5.2. Form Designs

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 5.3. Navigation and User Workflow

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 5.4. Report Designs

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 5.5. Screenshots of Forms and Reports

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# Privacy and Security Considerations

# 6.1. Identification of Sensitive Data

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.2. Data Protection Measures

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.3. GDPR Compliance Analysis

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.3.1. Data Minimization

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.3.2. Purpose Limitation

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.3.3. Storage Limitation

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.3.4. Data Subject Rights

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 6.4. Recommendations

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# Critical Evaluation

# 7.1. Reflection on Design and Implementation

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 7.2. Lessons Learned

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 7.3. Strengths and Limitations of the Solution

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 7.4. Value of the Coursework to Your Learning

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# 7.5. Individual Contributions

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# Appendices

# A. ER Diagram Exports

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# B. Table Definitions and Properties

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# C. Sample Data Screenshots

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# D. SQL Query Windows and Results

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# E. Form and Report Screenshots

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# F. References

The report presents a comprehensive analysis and design of a database-driven appointment scheduling system for a small-to-medium-sized enterprise (SME) operating within the service sector.

# Appendix A: SQL Queries

**Queries:**

**Query 1:**

SELECT

Service.ServiceName,

Year(ClientAppointments.AppointmentDate) AS AppointmentYear,

Month(ClientAppointments.AppointmentDate) AS AppointmentMonth,

Sum(Invoice.InvoiceTotal) AS TotalIncome

FROM

(ClientAppointments

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

GROUP BY

Service.ServiceName,

Year(ClientAppointments.AppointmentDate),

Month(ClientAppointments.AppointmentDate)

ORDER BY

AppointmentYear, AppointmentMonth, Service.ServiceName;

A computer screen shot of a computer

AI-generated content may be incorrect.

**Query 2:**

SELECT

Employee.EmployeeName,

Avg(Invoice.InvoiceTotal) AS AvgInvoiceTotal

FROM

(ClientAppointments

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

GROUP BY

Employee.EmployeeName

ORDER BY

AvgInvoiceTotal DESC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 3:**

SELECT

Client.ClientName,

Month(ClientAppointments.AppointmentDate) AS AppointmentMonth,

Year(ClientAppointments.AppointmentDate) AS AppointmentYear,

Count(ClientAppointments.AppointmentID) AS NumberOfAppointments

FROM

Client

INNER JOIN ClientAppointments ON Client.ClientID = ClientAppointments.ClientID

GROUP BY

Client.ClientName,

Month(ClientAppointments.AppointmentDate),

Year(ClientAppointments.AppointmentDate)

HAVING

Count(ClientAppointments.AppointmentID) > 3

ORDER BY

AppointmentYear, AppointmentMonth, NumberOfAppointments DESC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 4:**

SELECT

Client.ClientName,

Client.ClientMobile,

Sum(Invoice.InvoiceTotal) AS TotalExpenditure

FROM

(Client

INNER JOIN ClientAppointments ON Client.ClientID = ClientAppointments.ClientID)

INNER JOIN (InvoiceRows

INNER JOIN Invoice ON InvoiceRows.InvoiceID = Invoice.InvoiceID)

ON ClientAppointments.AppointmentID = InvoiceRows.AppointmentID

GROUP BY

Client.ClientName,

Client.ClientMobile

ORDER BY

TotalExpenditure DESC;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 5:**

SELECT

Employee.EmployeeName,

Count(DISTINCT ClientAppointments.ClientID) AS NumberOfClients

FROM

(Employee

INNER JOIN ClientAppointments ON Employee.EmployeeID = ClientAppointments.EmployeeID)

GROUP BY

Employee.EmployeeName

HAVING

Count(DISTINCT ClientAppointments.ClientID) > 1

ORDER BY

NumberOfClients DESC;

A computer screen shot of a white screen

AI-generated content may be incorrect.

**Query 6:**

SELECT Service.ServiceName

FROM ServicesProvided

INNER JOIN Service ON ServicesProvided.ServiceID = Service.ServiceID

WHERE ServicesProvided.EmployeeID = 1;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 7:**

SELECT ClientAppointments.AppointmentDate, Service.ServiceName, Employee.EmployeeName

FROM ClientAppointments

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID

WHERE ClientAppointments.ClientID = 1;

A computer screen with a white grid

AI-generated content may be incorrect.  
 **Query 8:**

SELECT InvoiceID, InvoiceNumber, InvoiceCost, InvoiceDiscount, InvoiceTotal

FROM Invoice;

A computer screen shot of a white screen

AI-generated content may be incorrect.

**Query 9:**

SELECT SUM(InvoiceTotal) AS TotalIncome

FROM Invoice;

A screenshot of a computer

AI-generated content may be incorrect.

**Query 10:**

SELECT Employee.EmployeeName, Service.ServiceName, ClientAppointments.AppointmentDate

FROM ClientAppointments

INNER JOIN Employee ON ClientAppointments.EmployeeID = Employee.EmployeeID

INNER JOIN Service ON ClientAppointments.ServiceID = Service.ServiceID

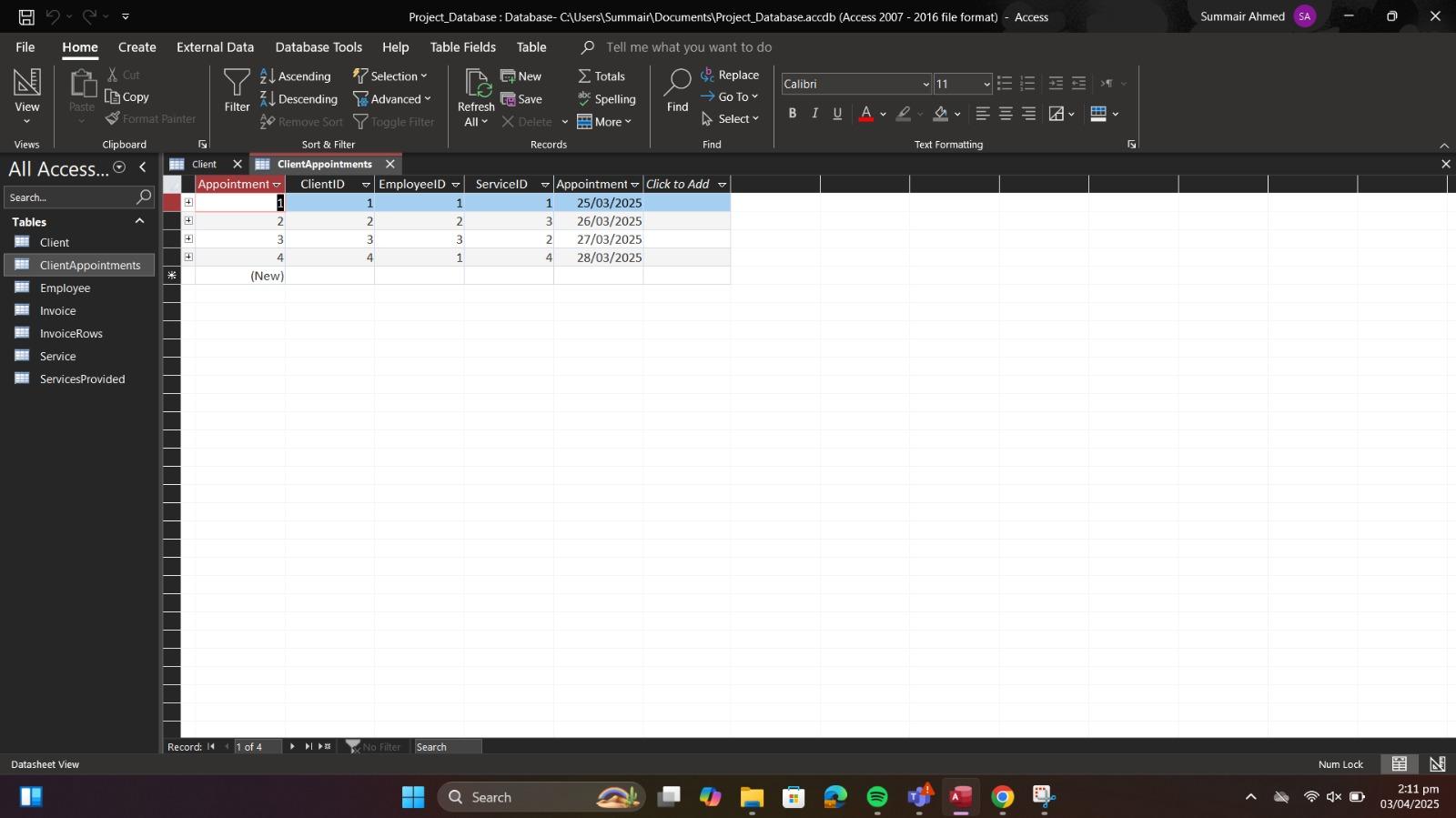
WHERE Month(ClientAppointments.AppointmentDate) = 3

AND Year(ClientAppointments.AppointmentDate) = 2025;

A computer screen shot of a white screen

AI-generated content may be incorrect.

# Appendix B: Sample Data Screenshots



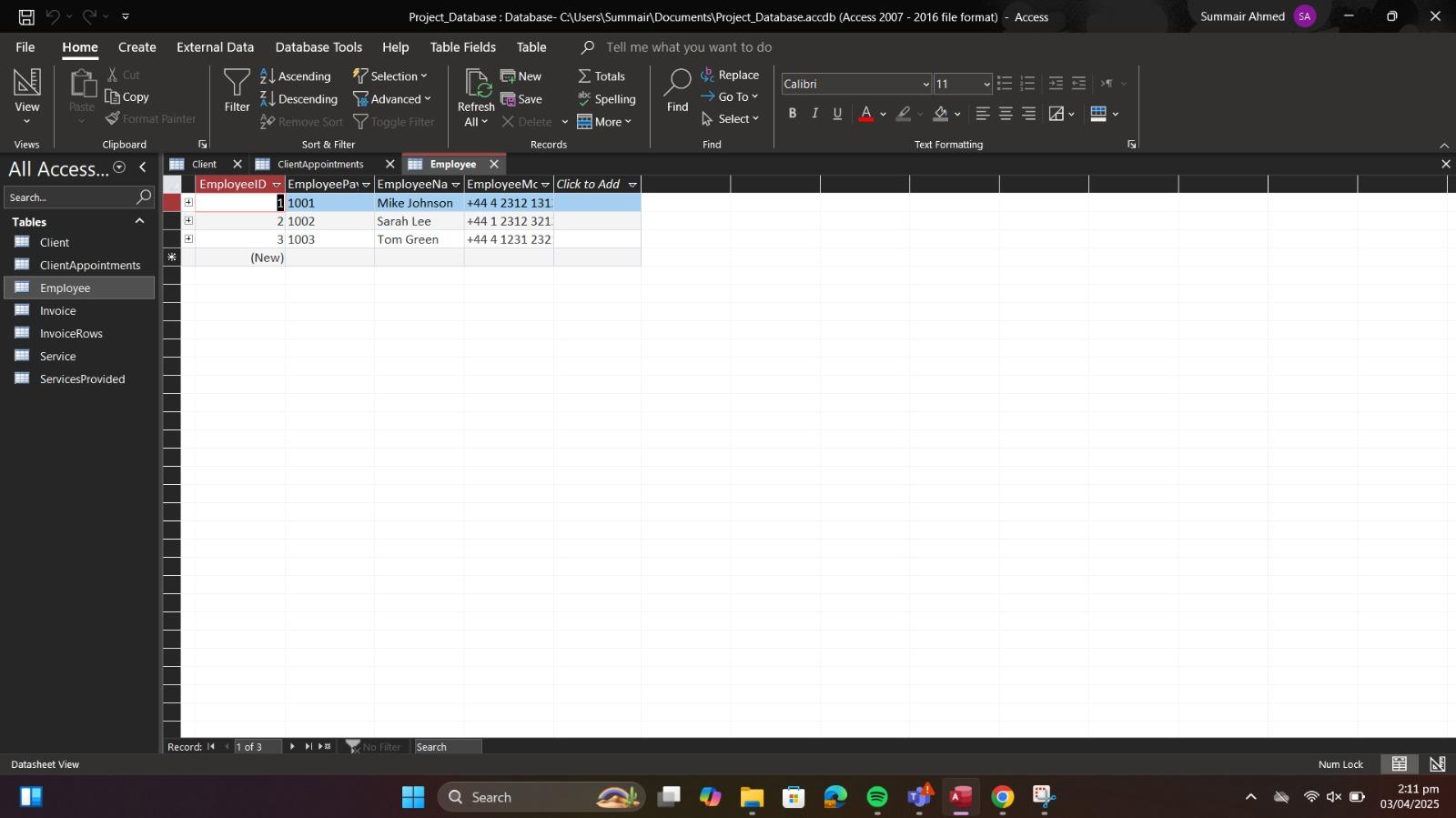
**Client Table:**

**Definition:**

Client (ClientID PRIMARY KEY, ClientName, ClientAddress, ClientEmail, ClientMobile)

**Description:**

This table stores all the information about the clients. ClientID is the unique identifier for each client.



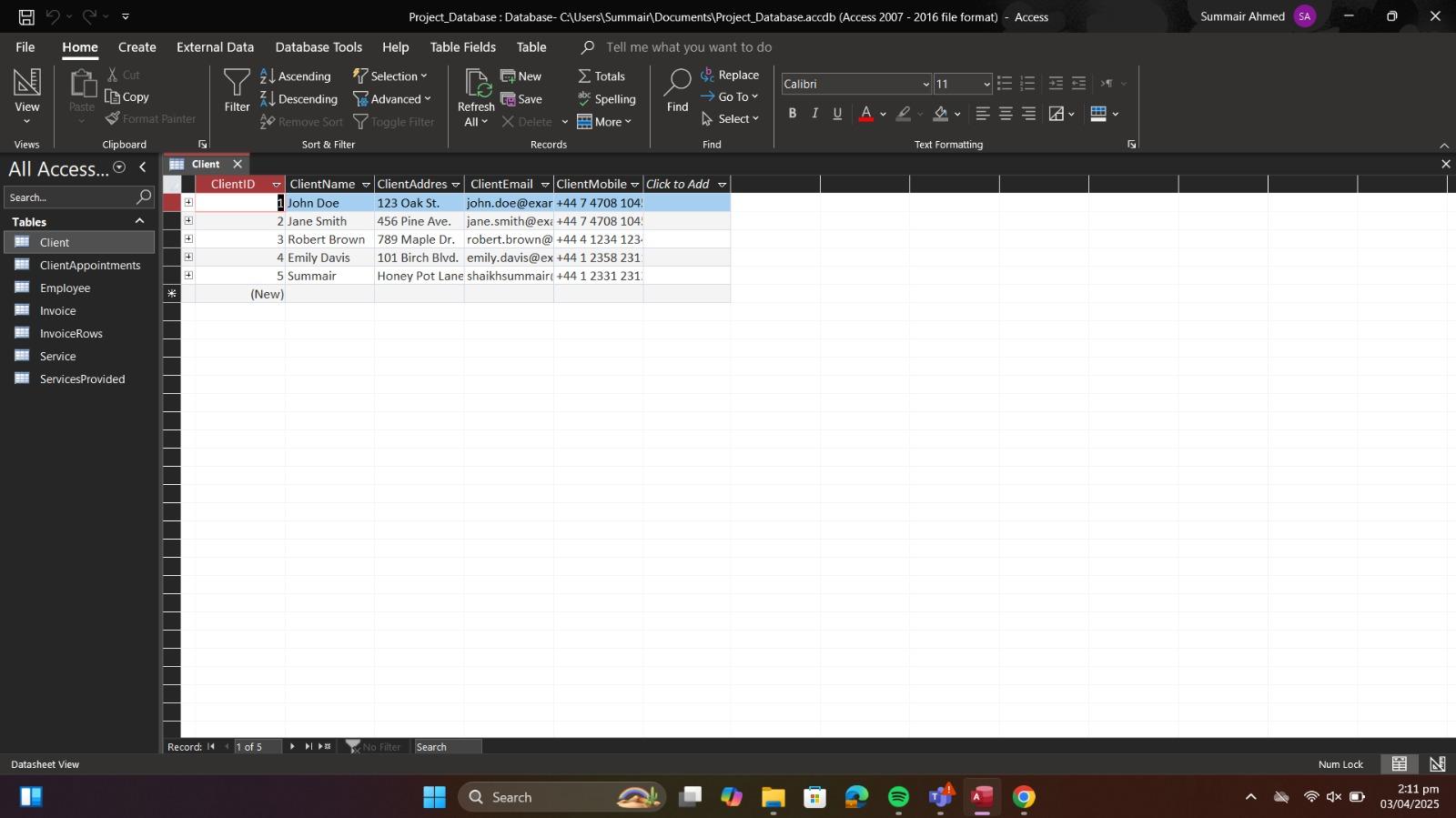
**Employee Table:**

**Definition:**

Employee (EmployeeID PRIMARY KEY, EmployeePayrollNumber, EmployeeName, EmployeeMobile)

**Description:**

This table holds employee details. EmployeeID is the primary key that uniquely identifies an employee.



**ClientAppointments Table (Associative Entity for DS1):**

**Definition:**

ClientAppointments (AppointmentID PRIMARY KEY,

ClientID FOREIGN KEY REFERENCES Client (ClientID),

EmployeeID FOREIGN KEY REFERENCES Employee (EmployeeID),

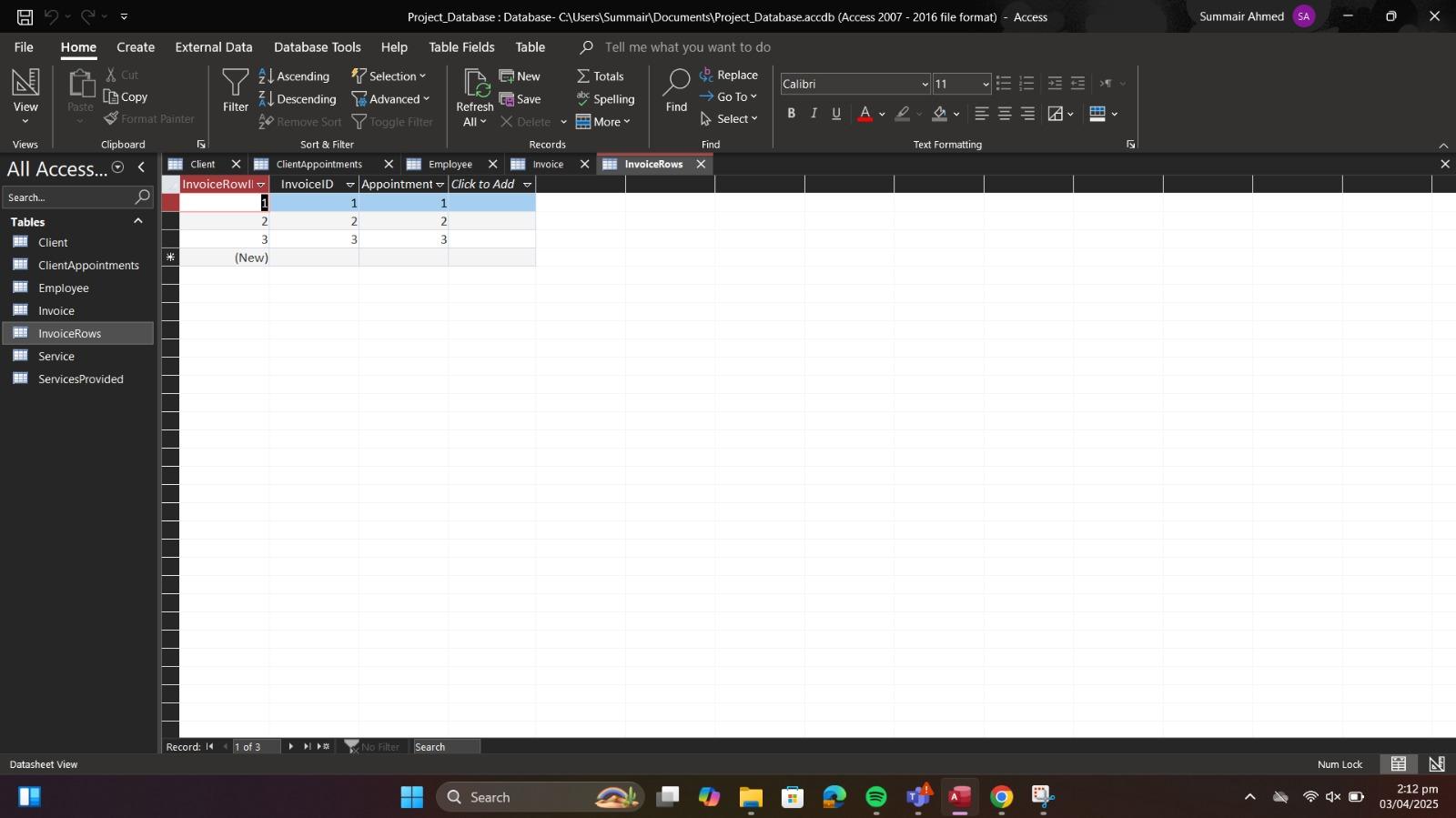
ServiceID FOREIGN KEY REFERENCES Service (ServiceID),

AppointmentDate,

AppointmentTime)

**Description:**

This table connects clients, employees, and services. It captures each appointment's date and time while maintaining integrity through foreign key constraints.



**InvoiceRows Table (Associative Entity for DS3)**:

**Definition:**

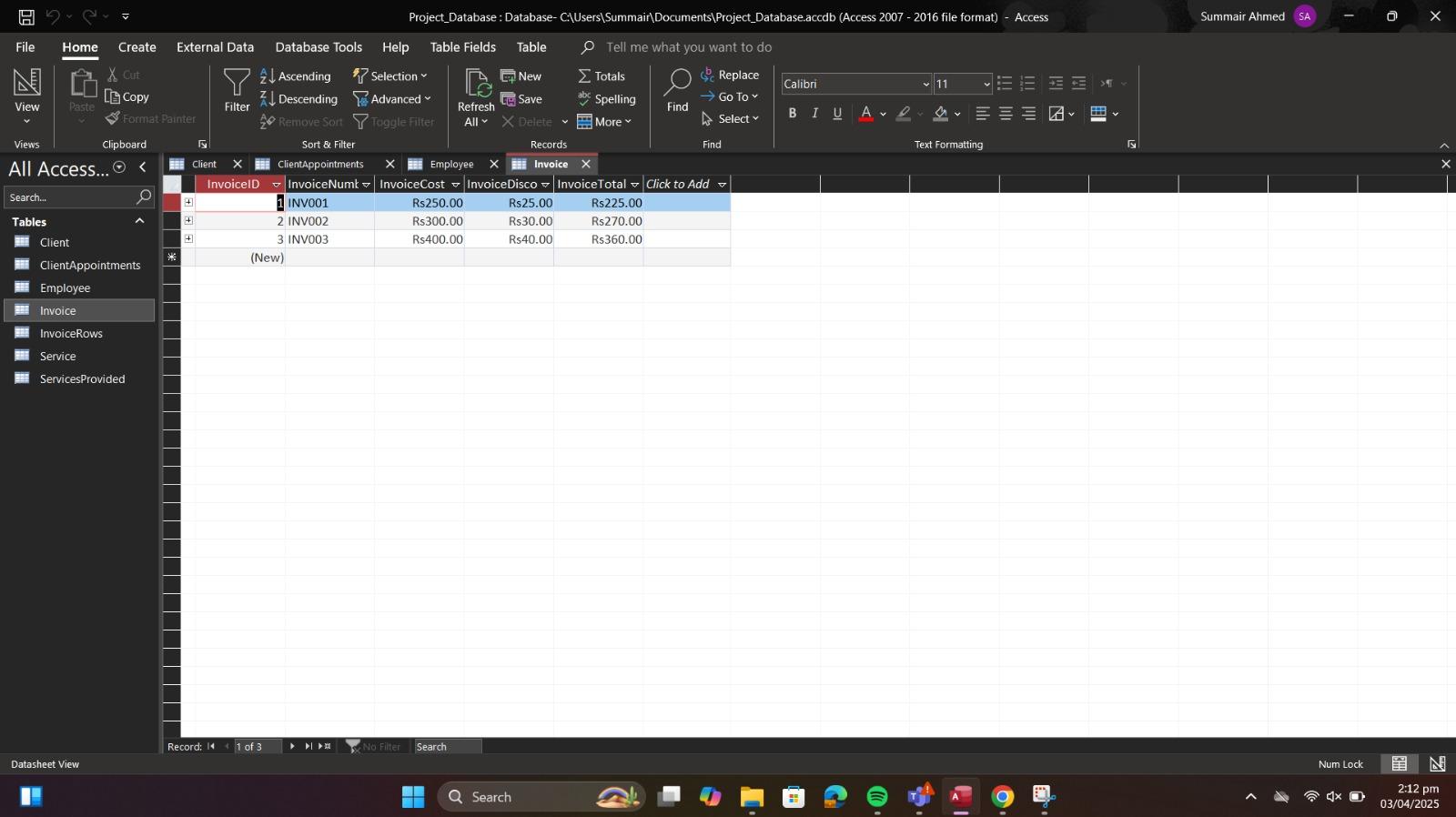
InvoiceRows (InvoiceRowID PRIMARY KEY,

InvoiceID FOREIGN KEY REFERENCES Invoice (InvoiceID),

AppointmentID FOREIGN KEY REFERENCES ClientAppointments (AppointmentID),

Cost)

This table links each invoice to the relevant client appointments, enabling the grouping of appointments into a single invoice. The foreign keys maintain the relationships defined in the ERD.



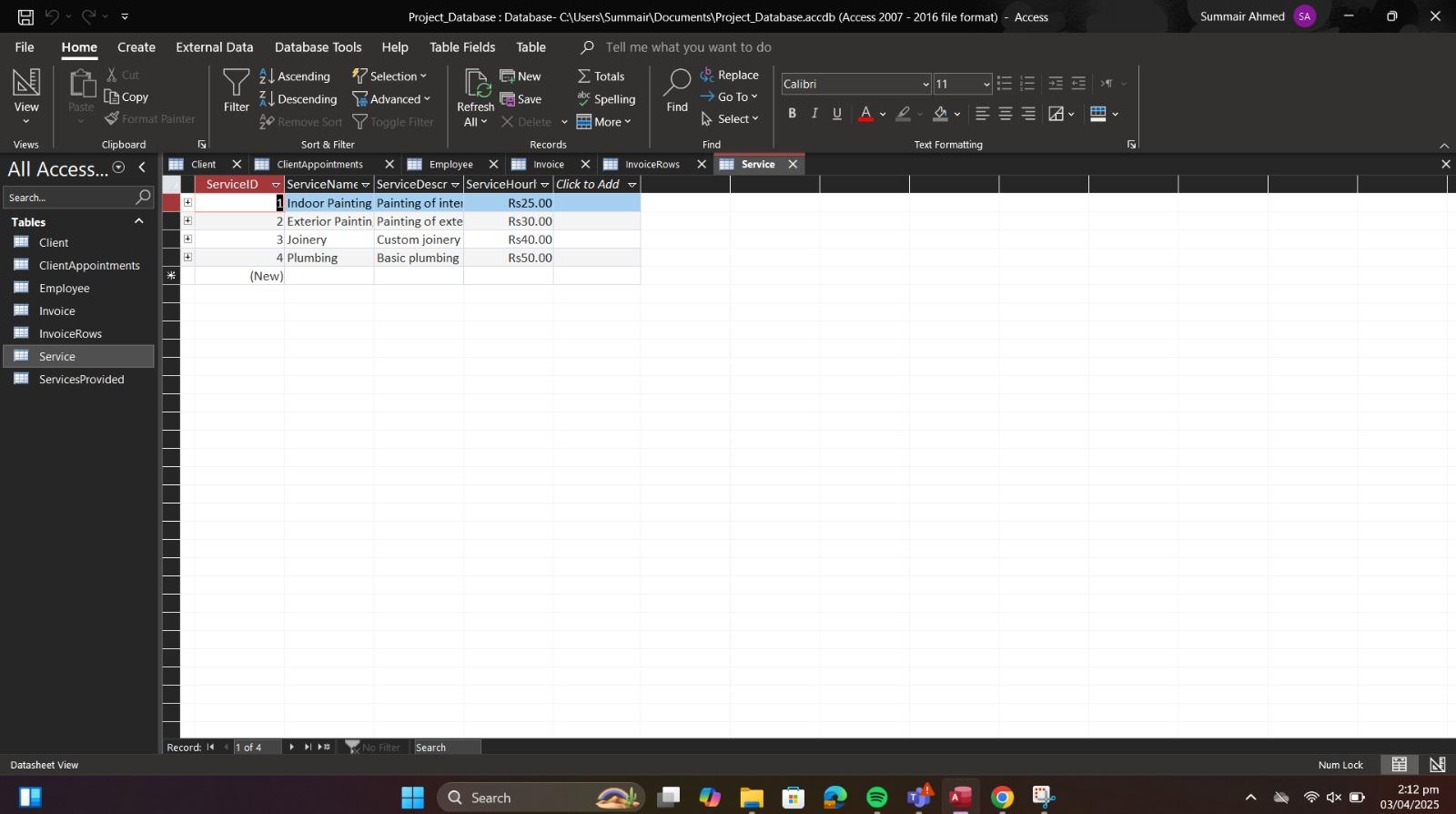
**Invoice Table:**

**Definition:**

Invoice (InvoiceID PRIMARY KEY, InvoiceNumber, InvoiceCost, InvoiceDiscount, InvoiceTotal)

**Description:**

This table stores invoice details, ensuring that every invoice has a unique identifier. It captures the financial details related to services rendered.



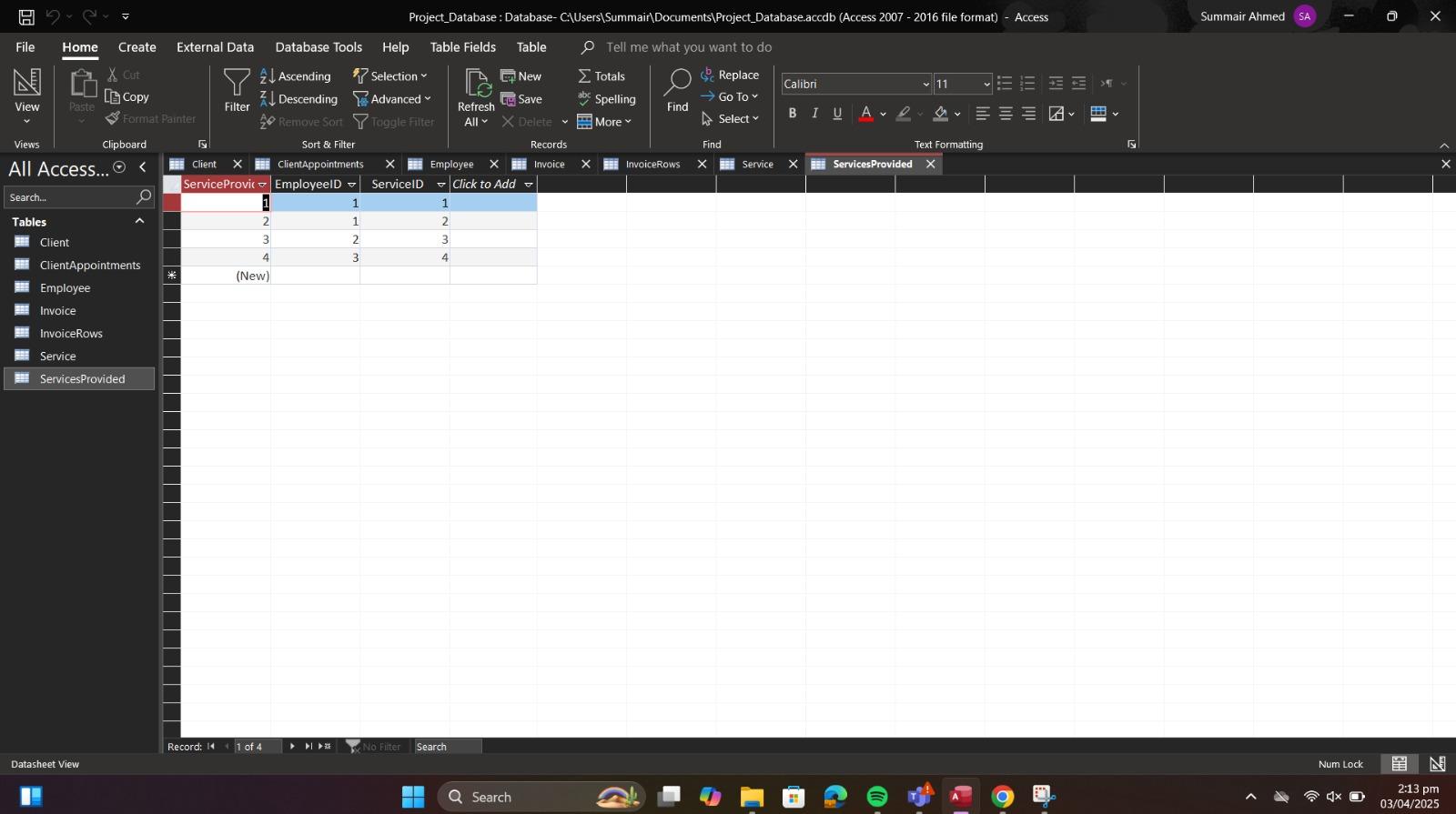
**Service Table:**

**Definition:**

Service (ServiceID PRIMARY KEY, ServiceName, ServiceDescription, ServiceHourlyRate)

Description:

This table records the details of each service offered by the SME. ServiceID uniquely identifies each service.



**ServicesProvided Table (Associative Entity for DS2):**

**Definition:**

ServicesProvided (ServiceProvidedID PRIMARY KEY,

ClientID FOREIGN KEY REFERENCES Client (ClientID),

ServiceID FOREIGN KEY REFERENCES Service (ServiceID),

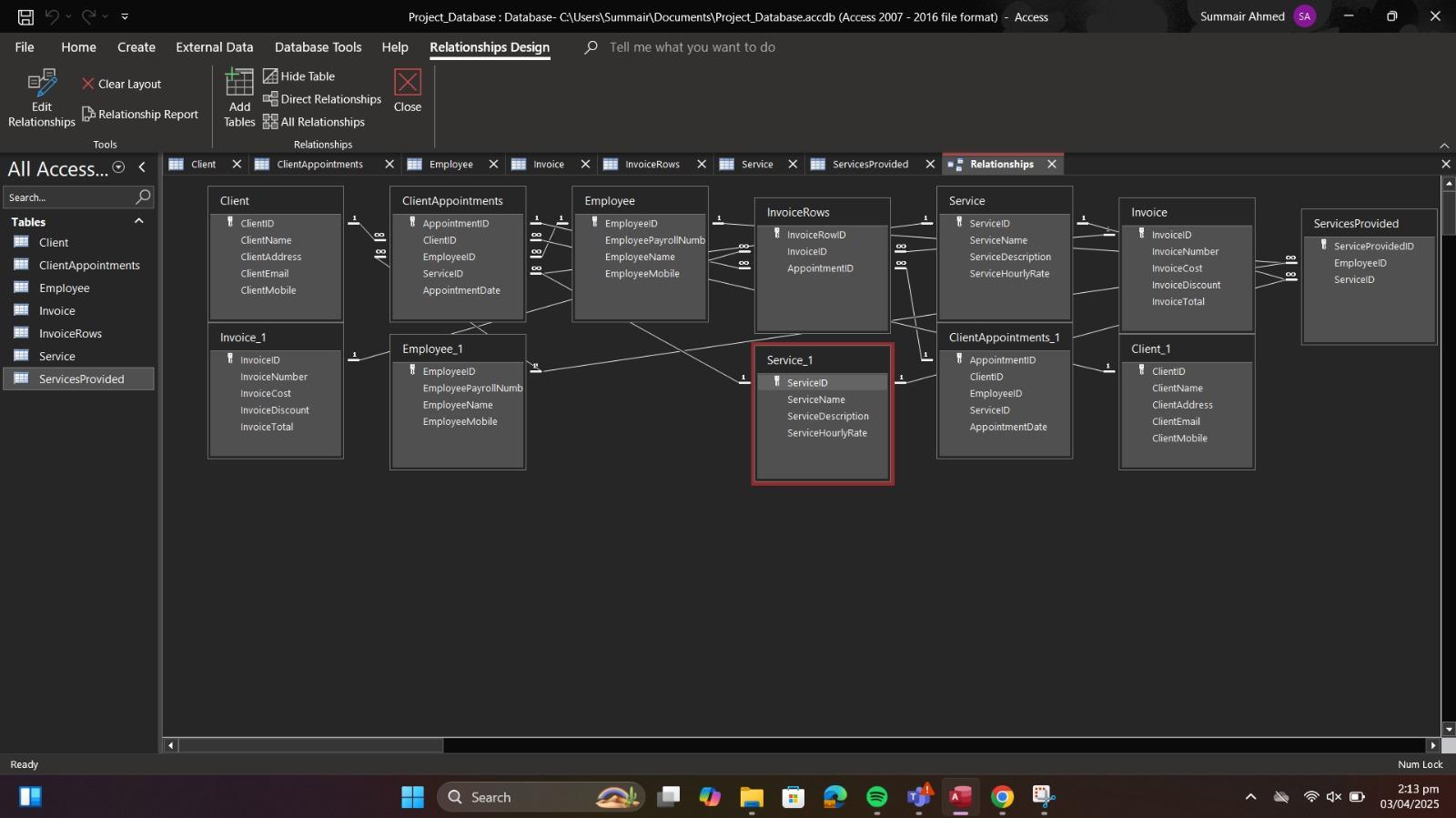
EmployeeID FOREIGN KEY REFERENCES Employee (EmployeeID),

DateOfService,

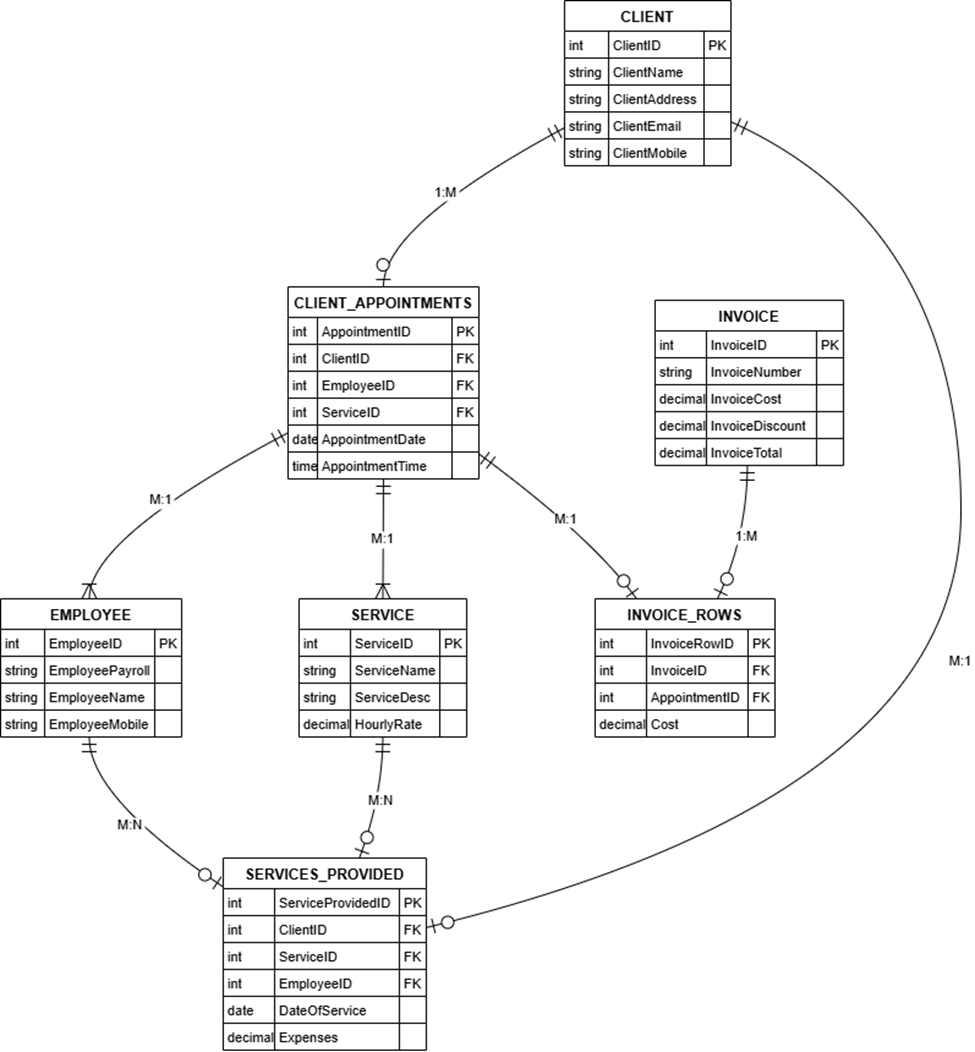
Expenses)

**Description:**

This table handles the many-to-many relationship between services and clients (with employee involvement) by linking multiple services provided to multiple clients along with service expenses.



# Appendix C: ERD Diagrams / Visuals



**Explanation of Relationship:**

**One-to-Many (1:M):**

A single Client can have many ClientAppointments, but each appointment is linked to only one Client.

An Employee can be associated with multiple ClientAppointments.

A Service can be part of many ClientAppointments.

An Invoice can include multiple InvoiceRows, each representing an appointment.

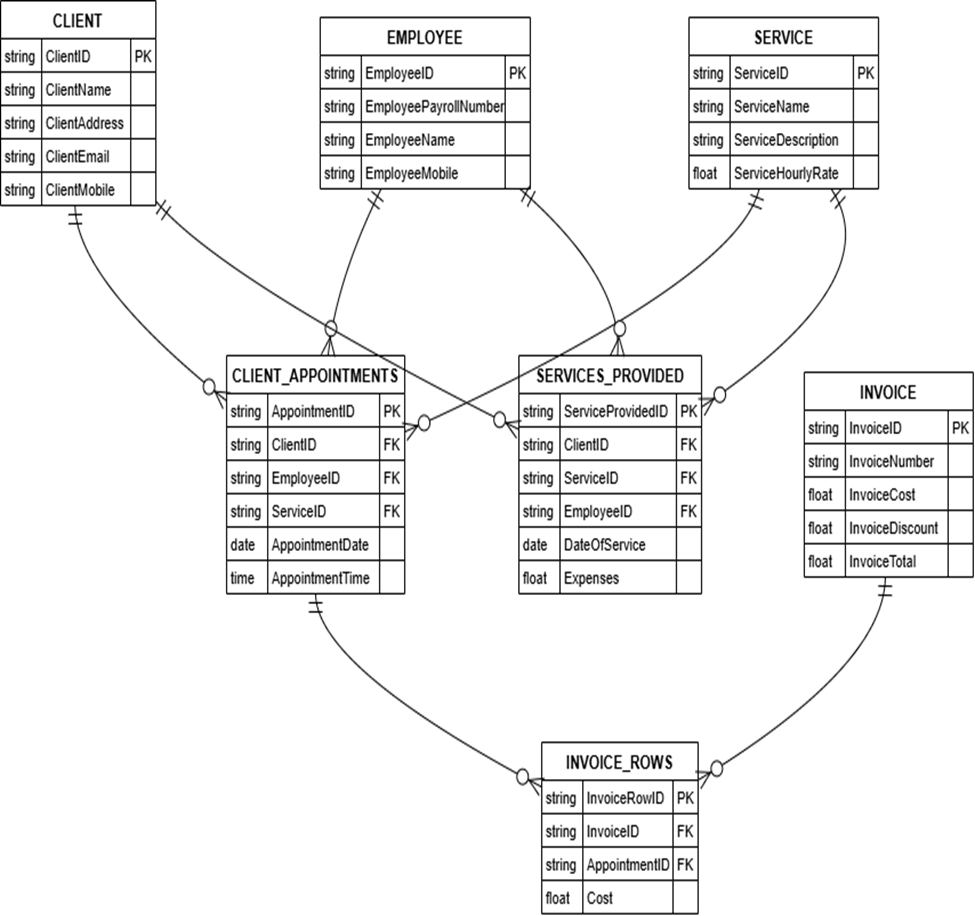
**Many-to-Many (M: N):**

The relationship between Services and Clients (with Employee involvement) is managed through the ServicesProvided table. This ensures that multiple services can be provided to multiple clients.

**Normalization and Integrity:**

The diagram reflects a design that adheres to Third Normal Form (3NF). Each entity has a unique primary key, and associative entities (ClientAppointments, ServicesProvided, and InvoiceRows) resolve the many-to-many relationships without any transitive dependencies.

Foreign key constraints ensure referential integrity across the database.

**

*Relational Schema*

**Description of the Relational Schema:**

The Relational Schema is the translation of the ERD into a set of relations (tables) with clearly defined keys and constraints. It outlines how the entities will be implemented in a relational database.