

# Math6147 - Database Modelling Coursework

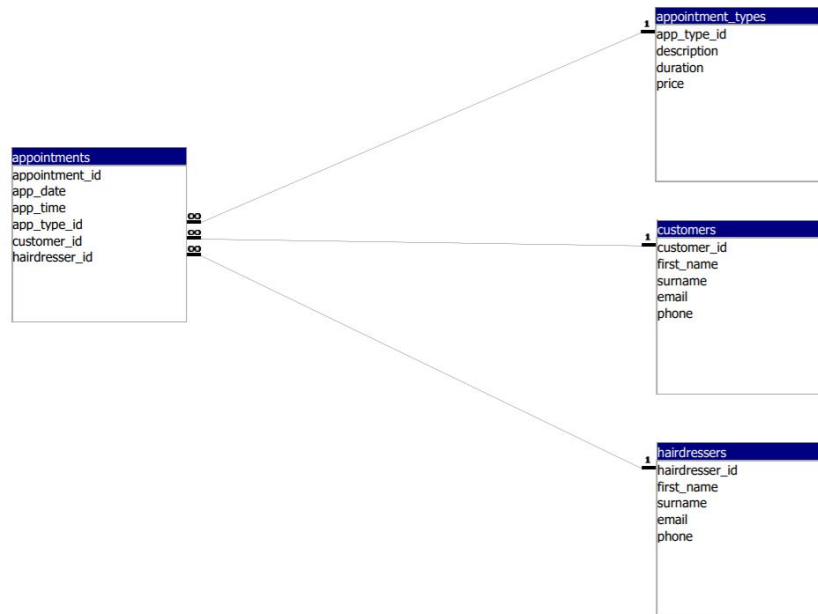
Student ID: 2940 4045

October 2021

## Introduction

The following report is intended to demonstrate the structure and design of the provided SQL code. In this assignment, we endeavour to build a relational database and then produce solutions to a series of nine query prompts.

## Entity-Relationship Diagram for our Database



# Technical Documentation

## Chosen Data Types

For modelling reasons, the following data-types were chosen for the fields in our data tables:

- Strictly positive non-null integers for all primary and foreign keys.
- Character strings for all instances of first names, surnames, email addresses and numbers.
- Currency type for appointment type prices.
- Date type for appointment dates.
- Time type for appointment times.
- Positive integers for appointment durations.

For further information, the table creation queries in the provided Microsoft Access File lay-out all data-types and keys clearly.

## Primary and Foreign Keys

The following table fields are keys for their given tables:

- Appointment Types
  - app\_type\_id is the primary key
- Customers
  - customer\_id is the primary key
- Hairdressers
  - hairdresser\_id is the primary key
- Appointments
  - appointment\_id is the primary key
  - app\_type\_id a foreign key
  - customer\_id a foreign key
  - hairdresser\_id is a foreign key

## Coding Style

Please note that the solutions in the provided Microsoft Access file will not be formatted identically to the below, this due to the file type not saving text formatting. For formatting reasons and ease of reading, I have observed the following coding-style conventions:

- Indentation to indicate scope where nesting occurs, except where this would render a line too large for the page-width of this document.
- Line-spacing to clearly separate key sections of the code.
- White-space within a given code line to separate key words and keep all symbols visible.
- Syntax highlighting has been used to improve readability.

## A Note on the Provided Data

The data for the database was imported directly into the database using the excel import wizard. Though, one particular record was unable to be directly imported into the database. In particular, under the “appointments” table in the excel spreadsheet, there is an error in the data: record 1757 is missing a hairdresser ID value, instead simply having ‘x’ entered into the cell. We may infer that this is an error produced by the hairdresser’s admin team. Based on the surrounding values, it seems intuitive to declare that  $x = 3$  as this would fit the surrounding pattern of the data. However, as we are unable to verify this with the data source, it seems reckless to edit any of the data directly. As such, record 1757 has been excluded from the following analyses.

## Query A

```
SELECT first_name, surname, app_time
FROM customers
    INNER JOIN appointments
        ON customers.customer_id = appointments.customer_id
WHERE hairdresser_id = 3
AND app_date = #26/11/2021#;
```

## Query B

```
SELECT appointment_id, app_date, app_time, customer_id,
    hairdresser_id, customer_id, description
FROM appointments
    INNER JOIN appointment_types
        ON appointments.app_type_id = appointment_types.app_type_id
WHERE customer_id = 257
AND description = "Perm";
```

## Query C

```
SELECT *
FROM (SELECT app_date, count(*) AS total_customers

FROM (SELECT app_date FROM (
    SELECT *
    FROM appointments
        INNER JOIN appointment_types
            ON appointments.app_type_id = appointment_types.app_type_id
    WHERE description = "Highlights"
))

GROUP BY app_date
ORDER BY app_date);
```

## Query D

```
SELECT TOP 1 total_spend, first_name, surname
FROM(

    SELECT total_spend, first_name, surname
    FROM (

        SELECT customer_id, SUM(price) AS total_spend
        FROM (

            SELECT customer_id, appointment_id, price
            FROM appointments
            INNER JOIN appointment_types
            ON appointment_types.app_type_id = appointments.app_type_id

        )
        GROUP BY customer_id
    ) AS price_table

    INNER JOIN customers
    ON customers.customer_id = price_table.customer_id
    ORDER BY total_spend DESC
);
```

## Query E

```
SELECT DISTINCT customers.customer_id, first_name, surname, description
FROM (

    SELECT customer_id, description
    FROM appointments
    INNER JOIN appointment_types
    ON appointments.app_type_id = appointment_types.app_type_id
    WHERE description = "Cut and Blow Dry"

) AS blow_dry_table
INNER JOIN customers
ON blow_dry_table.customer_id = customers.customer_id
ORDER BY customers.customer_id;
```

## Query F

```
SELECT time_period, SUM(price) AS total_revenue
FROM (

SELECT time_period, appointment_id, price
FROM (

SELECT appointment_id, price,
IIF ( app_date < DateValue("01-11-2021"), "Previous",

    IIF (app_date < DateValue("01-12-2021"), "November 2021",

        IIF (app_date < DateValue("01-01-2022"), "December 2021",

            IIF (app_date < DateValue("01-02-2022"), "January 2022",

                IIF (app_date >= DateValue("01-02-2022"), "Later"
            )
        )
    )
)
) AS time_period

FROM (

SELECT appointment_id, app_date, price
FROM appointments
INNER JOIN appointment_types
ON appointments.app_type_id = appointment_types.app_type_id

)
)
WHERE time_period = "November 2021"
OR time_period = "December 2021"
OR time_period = "January 2022"

)
GROUP BY time_period;
```

## Query G

```
SELECT totals_table.hairdresser_id, first_name, surname, total_appointments
FROM (

    SELECT hairdresser_id, COUNT(appointment_id) AS total_appointments
    FROM (

        SELECT hairdresser_id, appointment_id, app_date
        FROM appointments

        WHERE app_date = #18/10/2021#
        OR app_date = #19/10/2021#
        OR app_date = #20/10/2021#
        OR app_date = #21/10/2021#
        OR app_date = #22/10/2021#
        OR app_date = #23/10/2021#
        OR app_date = #24/10/2021#

    )

    GROUP BY hairdresser_id

) AS totals_table

INNER JOIN hairdressers
ON hairdressers.hairdresser_id = totals_table.hairdresser_id;
```

## Query H

```
SELECT TOP 1 total_perms, first_name, surname
FROM (

SELECT hairdressers.hairdresser_id, first_name, surname, total_perms
FROM (

SELECT hairdresser_id, COUNT(appointment_id) AS total_perms
FROM (

SELECT appointment_id, hairdresser_id, description, time_period
FROM (

SELECT appointment_id, hairdresser_id, app_type_id,
    IIF ( app_date < DateValue("01-12-2021"), "Previous",
        IIF (app_date < DateValue("01-01-2022"), "December 2021",
            IIF (app_date >= DateValue("01-01-2022"), "Later"
                )
            )
        ) AS time_period
FROM appointments

) AS times_table

INNER JOIN appointment_types
ON times_table.app_type_id = appointment_types.app_type_id

WHERE time_period = "December 2021"
AND description = "Perm"

)

GROUP BY hairdresser_id

) AS totals_table

INNER JOIN hairdressers
ON hairdressers.hairdresser_id = totals_table.hairdresser_id

)

ORDER BY total_perms DESC;
```



## Query I

```
SELECT customers.customer_id, first_name, surname, total_owed
FROM (

SELECT customer_id, SUM(price) AS total_owed
FROM (

SELECT appointment_id, customer_id, time_period, price
FROM (

SELECT *
FROM (

SELECT appointment_id, customer_id, app_type_id,
IIF ( app_date < DateValue("01-12-2021"), "Previous",
      IIF (app_date < DateValue("01-01-2022"), "December 2021",
            IIF (app_date >= DateValue("01-01-2022"), "Later"
                )
            )
        ) AS time_period
FROM appointments

)

WHERE time_period = "December 2021"

) AS times_table

INNER JOIN appointment_types
ON appointment_types.app_type_id = times_table.app_type_id

)

GROUP BY customer_id

) AS debt_table

INNER JOIN customers
ON customers.customer_id = debt_table.customer_id;
```