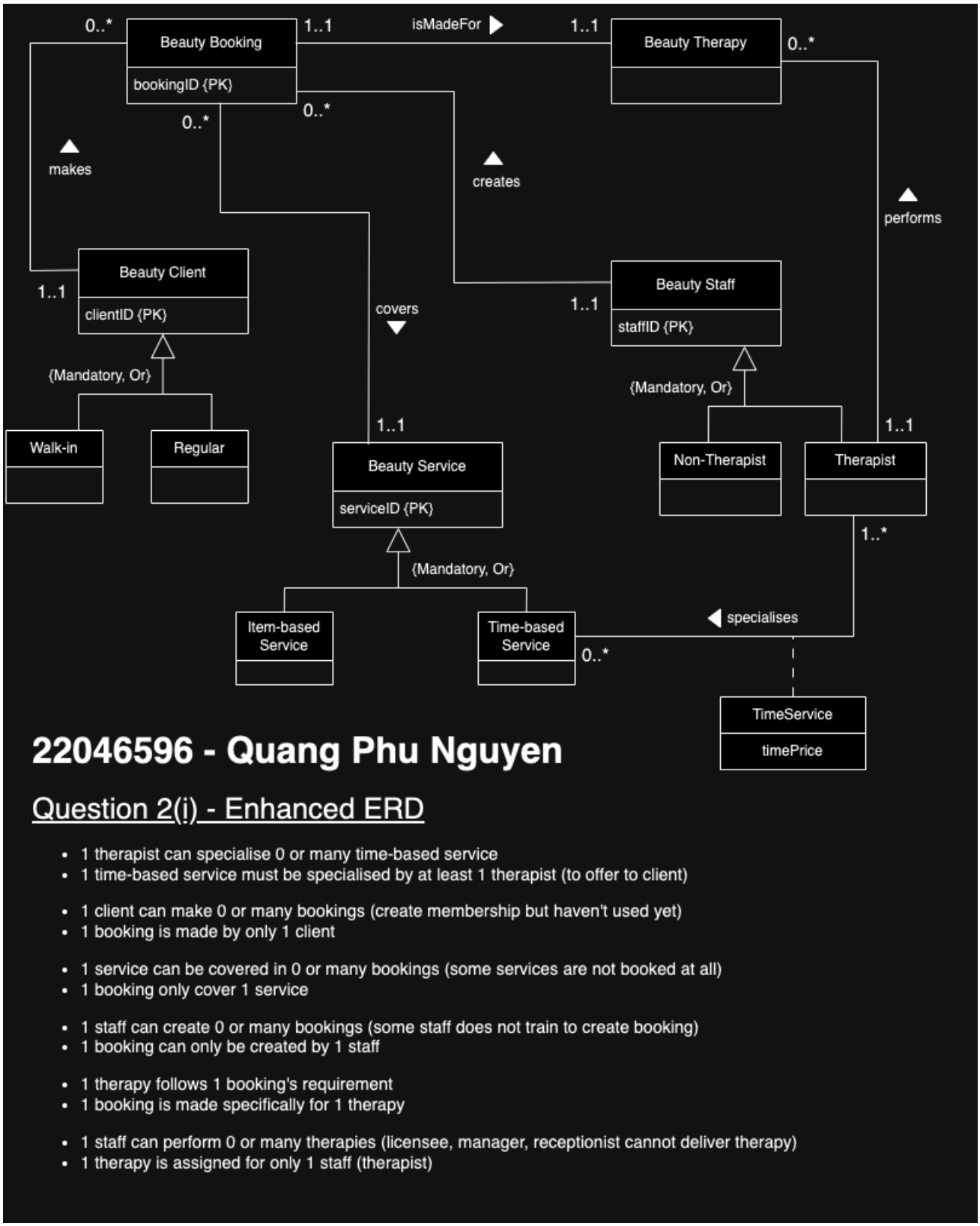
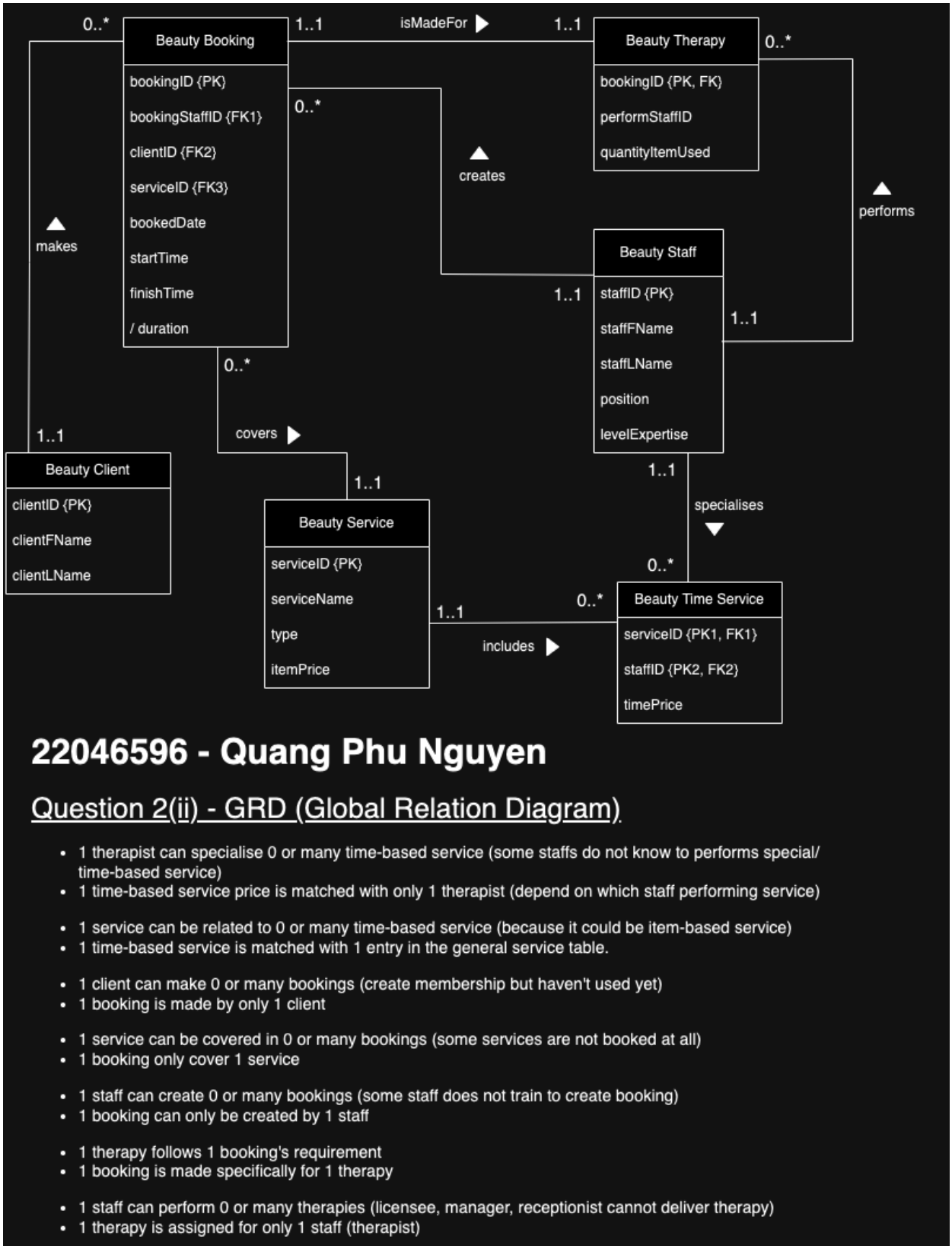
Database modelling

i. (Enhanced) Entity-Relation Diagram (E-ERD)



ii. Global Relation Diagram (GRD)



iii. Some note (without any requirement)

More analysis and SQL

i. Create database tables (Screenshot)

Filename: Q3_Table Creation

Filename: Q3_Data Insertion

Filename: Q3_Query

```
/*
DROP TABLE Beauty_Therapy;
DROP TABLE Beauty_Booking;
DROP TABLE Beauty_TimeService;
DROP TABLE Beauty_Staff;
DROP TABLE Beauty_Client;
DROP TABLE Beauty_Service;
*/

CREATE TABLE Beauty_Staff (
	staffID varchar(8) PRIMARY KEY,
	staffFname varchar(20) NOT NULL,
	staffLname varchar(20) NOT NULL,
	position varchar(17)
		CHECK (position IN ('Licensee', 'Manager', 'Receptionist', 'Beauty
Therapist')),
	levelExpertise INT, -- year performing service
);

CREATE TABLE Beauty_Client (
	clientID varchar(8) PRIMARY KEY,
	clientFname varchar(20),
	clientLname varchar(20),

	CHECK ( -- Only 1 entry 'Walk-in' for all Walk-in client
	 (clientID = 'Walk-in' AND clientFname IS NULL AND clientLname IS NULL) OR
	 (clientID != 'Walk-in' AND clientFname IS NOT NULL AND clientLname IS NOT NULL)
    ));

CREATE TABLE Beauty_Service (
	serviceID varchar(8) PRIMARY KEY,
	serviceName varchar(20) NOT NULL,
	type CHAR(10) CHECK (type IN ('Item-based', 'Time-based')),
	itemPrice numeric(2) NOT NULL,

	CHECK ((type = 'Time-based' AND itemPrice = 0.00) OR type = 'Item-based') -- 0 when
it is time-based
);

CREATE TABLE Beauty_TimeService ( -- LEFT JOIN with table Service for overall information
	qualifiedStaffID varchar(8),
	serviceID varchar(8),
	timePrice numeric(3), -- Price per hour

	PRIMARY KEY (qualifiedStaffID, serviceID),
	FOREIGN KEY (serviceID) REFERENCES Beauty_Service,
	FOREIGN KEY (qualifiedStaffID) REFERENCES Beauty_Staff
);

CREATE TABLE Beauty_Booking (
	bookingID varchar(8) PRIMARY KEY,
	bookingStaffID varchar(8) NOT NULL,
```

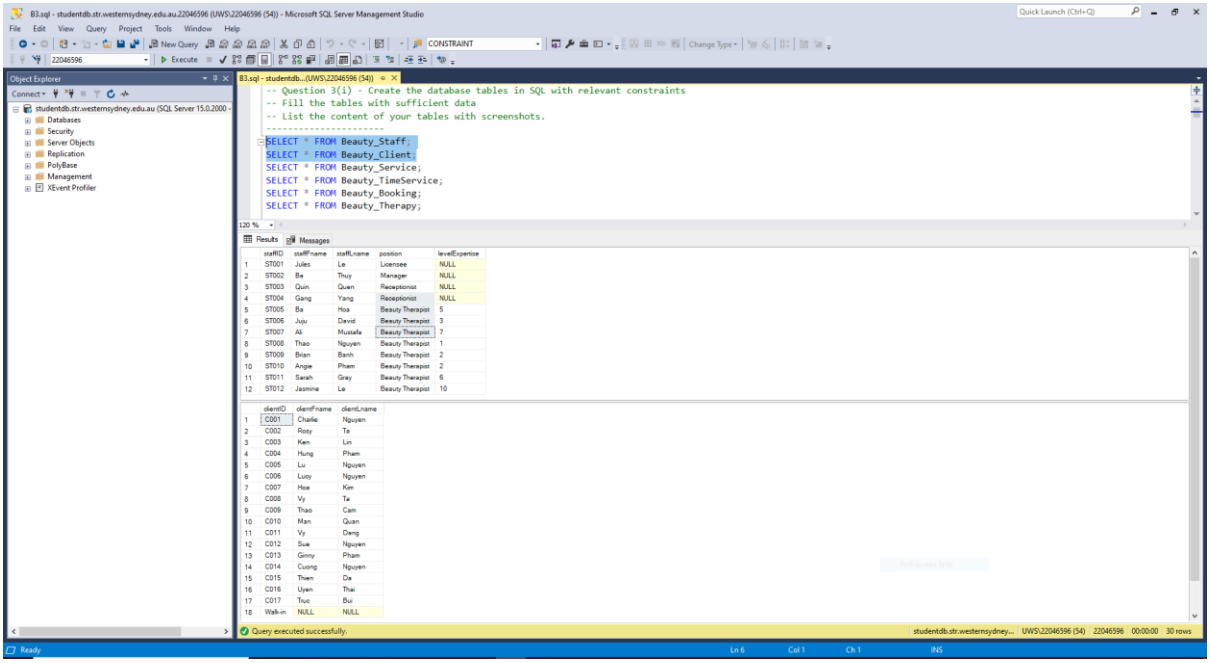
```
        clientID varchar(8) NOT NULL,
        serviceID varchar(8) NOT NULL,
        bookedDate DATE NOT NULL,
        startTime TIME NOT NULL,
        finishTime TIME NOT NULL,
        duration AS DATEDIFF(MINUTE, startTime, finishTime), -- calculate the duration

        CHECK (startTime < finishTime),
        UNIQUE (clientID, bookedDate, startTime, finishTime),

        FOREIGN KEY (bookingStaffID) REFERENCES Beauty_Staff,
        FOREIGN KEY (clientID) REFERENCES Beauty_Client,
        FOREIGN KEY (serviceID) REFERENCES Beauty_Service
);

CREATE TABLE Beauty_Therapy (
        bookingID varchar(8),
        performStaffID varchar(8),
        quantityItemUsed INT,

        PRIMARY KEY (bookingID),
        FOREIGN KEY (bookingID) REFERENCES Beauty_Booking,
        FOREIGN KEY (performStaffID) REFERENCES Beauty_Staff,
);
```

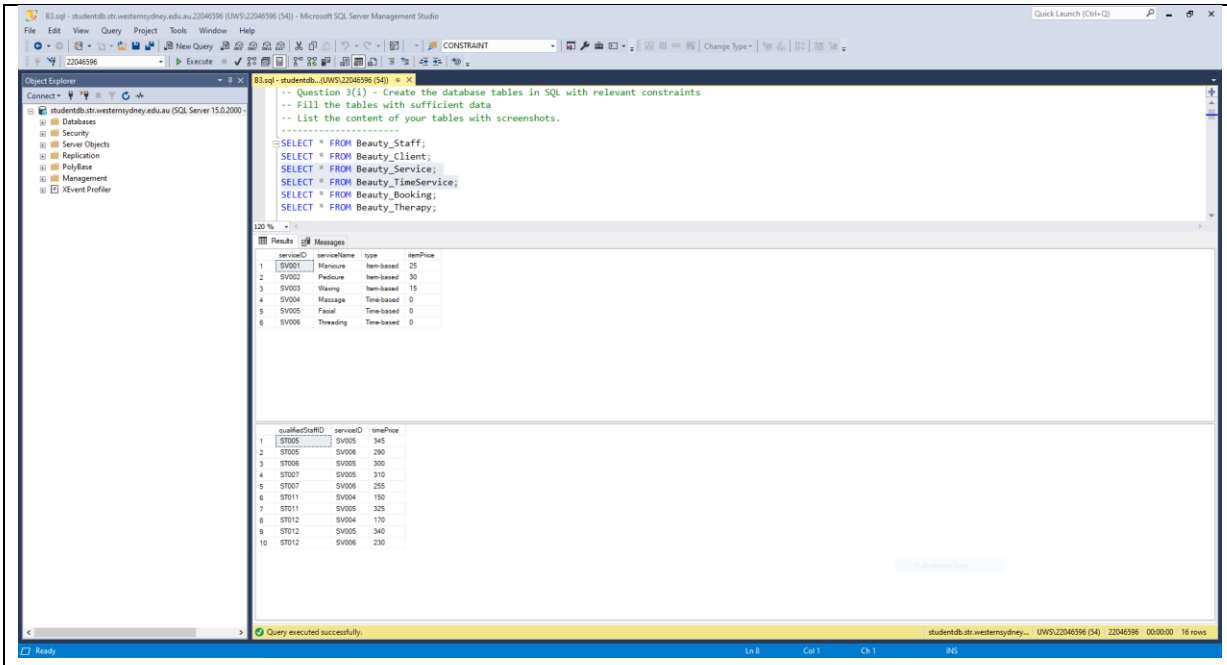
```
INSERT INTO Beauty_Staff VALUES -- staffID, staffFname, staffLname, position, levelExpertise
('ST001', 'Jules', 'Le', 'Licensee', NULL), -- Non-Beauty Therapist
('ST002', 'Ba', 'Thuy', 'Manager', NULL),
('ST003', 'Quin', 'Quen', 'Receptionist', NULL),
('ST004', 'Gang', 'Yang', 'Receptionist', NULL),
('ST005', 'Ba', 'Hoa', 'Beauty Therapist', 5), -- Beauty Therapist from this line
('ST006', 'Juju', 'David', 'Beauty Therapist', 3),
('ST007', 'Ali', 'Mustafa', 'Beauty Therapist', 7),
('ST008', 'Thao', 'Nguyen', 'Beauty Therapist', 1),
('ST009', 'Brian', 'Banh', 'Beauty Therapist', 2),
('ST010', 'Angie', 'Pham', 'Beauty Therapist', 2),
('ST011', 'Sarah', 'Gray', 'Beauty Therapist', 6),
('ST012', 'Jasmine', 'Le', 'Beauty Therapist', 10);

INSERT INTO Beauty_Client VALUES -- clientID, clientFname, clientLname
('Walk-in', NULL, NULL),
('C001', 'Charlie', 'Nguyen'),
('C002', 'Rosy', 'Ta'),
('C003', 'Ken', 'Lin'),
('C004', 'Hung', 'Pham'),
('C005', 'Lu', 'Nguyen'),
('C006', 'Lucy', 'Nguyen'),
('C007', 'Hoa', 'Kim'),
('C008', 'Vy', 'Ta'),
('C009', 'Thao', 'Cam'),
('C010', 'Man', 'Quan'),
('C011', 'Vy', 'Dang'),
('C012', 'Sue', 'Nguyen'),
('C013', 'Ginny', 'Pham'),
('C014', 'Cuong', 'Nguyen'),
('C015', 'Thien', 'Da'),
('C016', 'Uyen', 'Thai'),
('C017', 'Truc', 'Bui');

INSERT INTO Beauty_Service VALUES -- serviceID, serviceName, type, itemPrice
('SV001', 'Manicure', 'Item-based', 25.00),
('SV002', 'Pedicure', 'Item-based', 30.00),
```

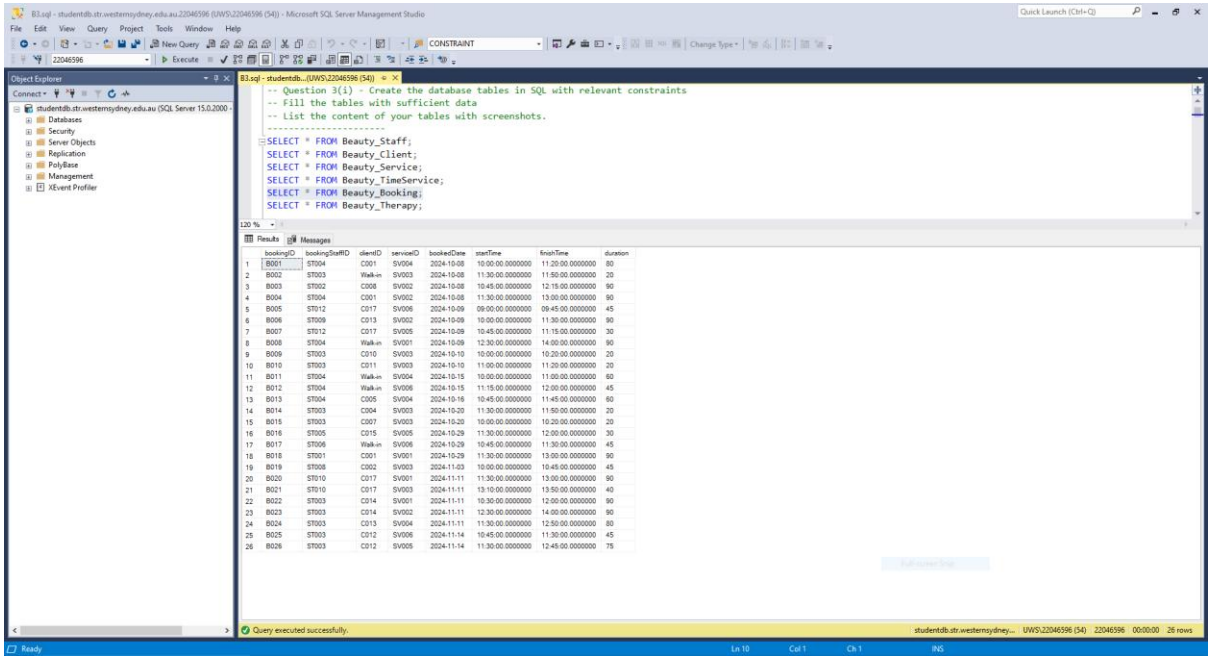
```
('SV003', 'Waxing', 'Item-based', 15.00),
('SV004', 'Massage', 'Time-based', 0.00), -- LEFT JOIN with TimeService For more information
price
('SV005', 'Facial', 'Time-based', 0.00),
('SV006', 'Threading', 'Time-based', 0.00);

INSERT INTO Beauty_TimeService VALUES -- qualifiedStaffID, serviceID, timePrice
('ST011', 'SV004', 150.00), -- Service 004
('ST012', 'SV004', 170.00),
('ST005', 'SV006', 290.00), -- Service 006
('ST007', 'SV006', 255.00),
('ST012', 'SV006', 230.00),
('ST005', 'SV005', 345.00), -- Service 005
('ST007', 'SV005', 310.00),
('ST006', 'SV005', 300.00),
('ST011', 'SV005', 325.00),
('ST012', 'SV005', 340.00);

INSERT INTO Beauty_Booking VALUES
-- bookingID, bookingStaffID, clientID, isWalkin, serviceID, bookedDate, startTime,
finishTime, duration
-- but duration is automatically calculated
('B001', 'ST004', 'C001', 'SV004', '2024-10-08', '10:00:00', '11:20:00'),  -- 2024-10-08
('B002', 'ST003', 'Walk-in', 'SV003', '2024-10-08', '11:30:00', '11:50:00'),
('B003', 'ST002', 'C008', 'SV002', '2024-10-08', '10:45:00', '12:15:00'),
('B004', 'ST004', 'C001', 'SV002', '2024-10-08', '11:30:00', '13:00:00'),
('B005', 'ST012', 'C017', 'SV006', '2024-10-09', '09:00:00', '09:45:00'),  -- 2024-10-09
('B006', 'ST009', 'C013', 'SV002', '2024-10-09', '10:00:00', '11:30:00'),
('B007', 'ST012', 'C017', 'SV005', '2024-10-09', '10:45:00', '11:15:00'),
('B008', 'ST004', 'Walk-in', 'SV001', '2024-10-09', '12:30:00', '14:00:00'),
('B009', 'ST003', 'C010', 'SV003', '2024-10-10', '10:00:00', '10:20:00'),  -- 2024-10-10
('B010', 'ST003', 'C011', 'SV003', '2024-10-10', '11:00:00', '11:20:00'),
('B011', 'ST004', 'Walk-in', 'SV004', '2024-10-15', '10:00:00', '11:00:00'),  -- 2024-10-15
('B012', 'ST004', 'Walk-in', 'SV006', '2024-10-15', '11:15:00', '12:00:00'),
('B013', 'ST004', 'C005', 'SV004', '2024-10-16', '10:45:00', '11:45:00'),  -- 2024-10-16
('B014', 'ST003', 'C004', 'SV003', '2024-10-20', '11:30:00', '11:50:00'),  -- 2024-10-20
('B015', 'ST003', 'C007', 'SV003', '2024-10-20', '10:00:00', '10:20:00'),
('B016', 'ST005', 'C015', 'SV005', '2024-10-29', '11:30:00', '12:00:00'),  -- 2024-10-29
('B017', 'ST006', 'Walk-in',  'SV006', '2024-10-29', '10:45:00', '11:30:00'),
('B018', 'ST001', 'C001', 'SV001', '2024-10-29', '11:30:00', '13:00:00'),
('B019', 'ST008', 'C002', 'SV003', '2024-11-03', '10:00:00', '10:45:00'),  -- 2024-11-03
('B020', 'ST010', 'C017', 'SV001', '2024-11-11', '11:30:00', '13:00:00'),  -- 2024-11-11
('B021', 'ST010', 'C017', 'SV003', '2024-11-11', '13:10:00', '13:50:00'),
('B022', 'ST003', 'C014', 'SV001', '2024-11-11', '10:30:00', '12:00:00'),
('B023', 'ST003', 'C014', 'SV002', '2024-11-11', '12:30:00', '14:00:00'),
('B024', 'ST003', 'C013', 'SV004', '2024-11-11', '11:30:00', '12:50:00'),
('B025', 'ST003', 'C012', 'SV006', '2024-11-14', '10:45:00', '11:30:00'),  -- 2024-11-14
('B026', 'ST003', 'C012', 'SV005', '2024-11-14', '11:30:00', '12:45:00');

INSERT INTO Beauty_Therapy VALUES -- bookingID, performStaffID, quantityItemUsed
('B001', 'ST011', 0),
('B002', 'ST005', 8),
('B003', 'ST010', 5),
('B004', 'ST006', 10),
('B005', 'ST007', 0),
('B006', 'ST008', 7),
('B007', 'ST007', 0),
('B008', 'ST009', 9),
('B009', 'ST008', 3),
('B010', 'ST008', 4),
('B011', 'ST012', 0),
('B012', 'ST012', 0),
```

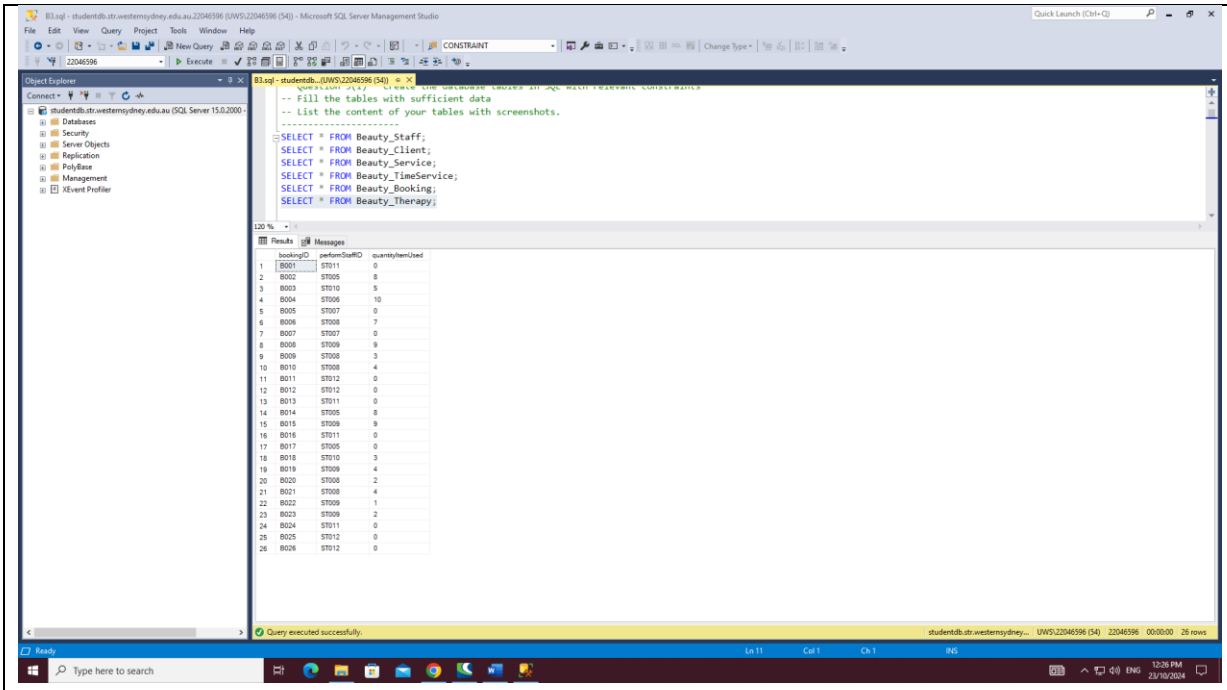
```
('B013', 'ST011', 0),
('B014', 'ST005', 8),
('B015', 'ST009', 9),
('B016', 'ST011', 0),
('B017', 'ST005', 0),
('B018', 'ST010', 3),
('B019', 'ST009', 4),
('B020', 'ST008', 2),
('B021', 'ST008', 4),
('B022', 'ST009', 1),
('B023', 'ST009', 2),
('B024', 'ST011', 0),
('B025', 'ST012', 0),
('B026', 'ST012', 0);
```

```
SELECT * FROM Beauty_Staff;
SELECT * FROM Beauty_Client;
SELECT * FROM Beauty_Service;
SELECT * FROM Beauty_TimeService;
SELECT * FROM Beauty_Booking;
SELECT * FROM Beauty_Therapy;
```




-- Question 3(i) - Create the database tables in SQL with relevant constraints
-- Fill the tables with sufficient data
-- List the content of your tables with screenshots.
SELECT * FROM Beauty_Staff;
SELECT * FROM Beauty_Client;
SELECT * FROM Beauty_Service;
SELECT * FROM Beauty_TimeService;
SELECT * FROM Beauty_Booking;
SELECT * FROM Beauty_Therapy;
Query executed successfully.

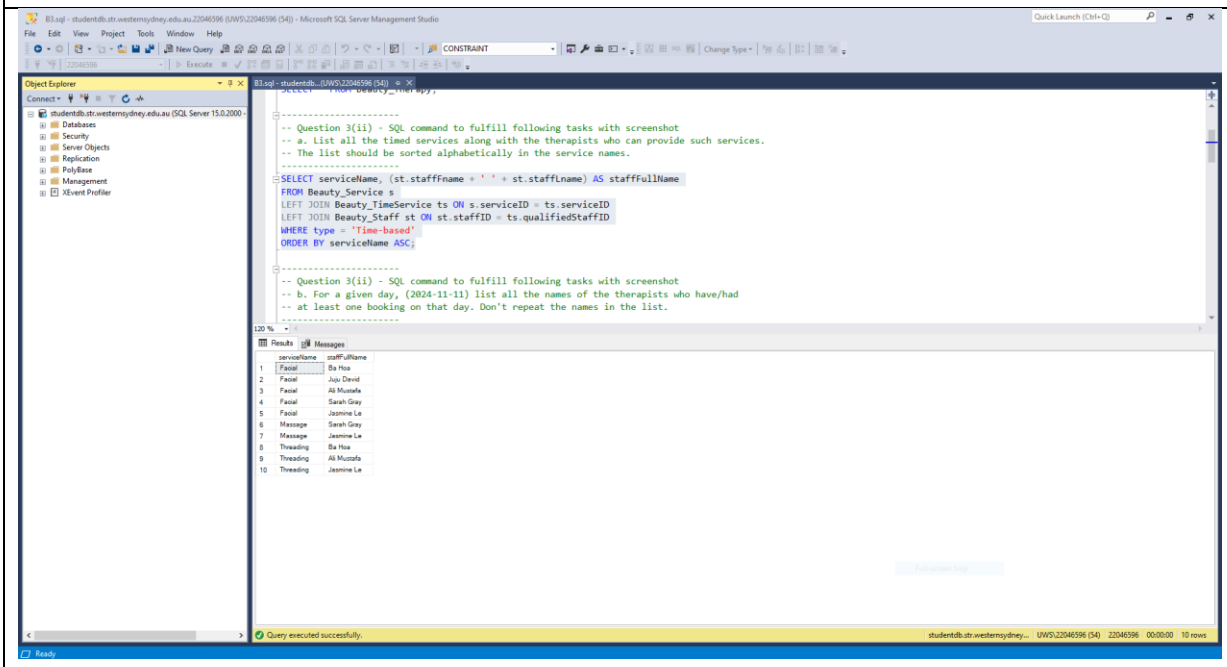

-- Question 3(i) - Create the database tables in SQL with relevant constraints
-- Fill the tables with sufficient data
-- List the content of your tables with screenshots.
SELECT * FROM Beauty_Staff;
SELECT * FROM Beauty_Client;
SELECT * FROM Beauty_Service;
SELECT * FROM Beauty_TimeService;
SELECT * FROM Beauty_Booking;
SELECT * FROM Beauty_Therapy;
Query executed successfully.



ii. SQL to complete specific requirement (giving extra 1 query) (Screenshot)

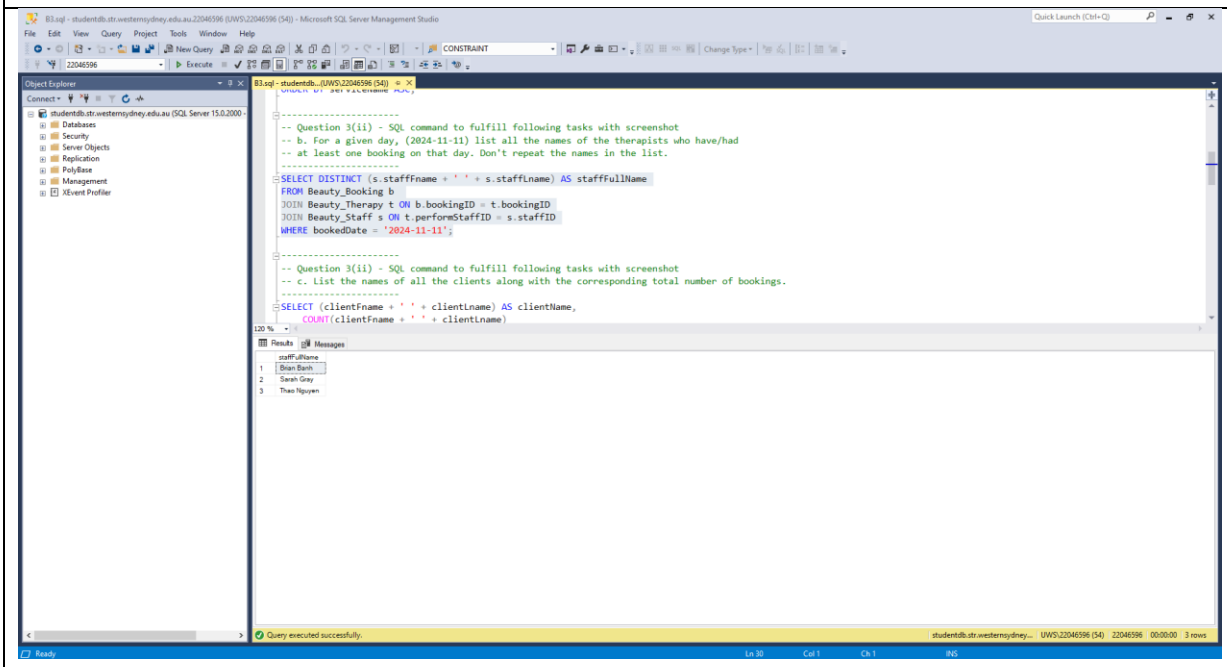
a. List all timed services with all qualified therapists. List alphabetically according to service names.

```
SELECT serviceName, (st.staffFname + ' ' + st.staffLname) AS staffFullName
FROM Beauty_Service s
LEFT JOIN Beauty_TimeService ts ON s.serviceID = ts.serviceID
LEFT JOIN Beauty_Staff st ON st.staffID = ts.qualifiedStaffID
WHERE type = 'Time-based'
ORDER BY serviceName ASC;
```



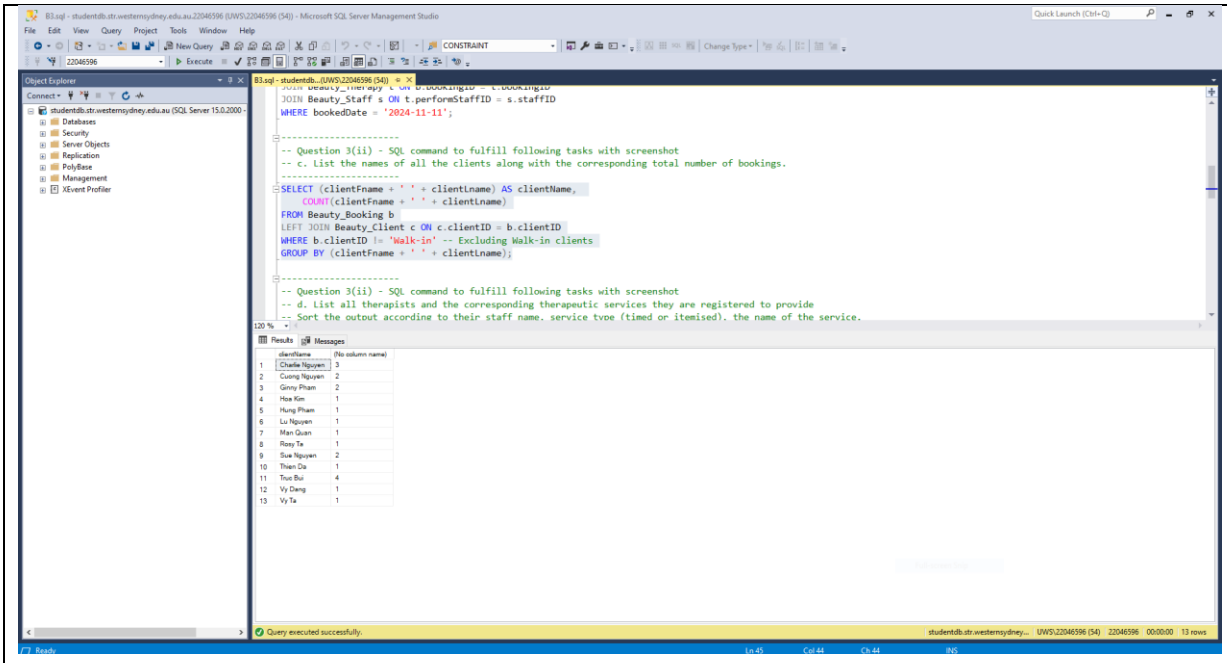
b. For a given day (2024-11-11), list all names of therapists (without duplication) who have at least 1 appointment/therapy

```sql
SELECT DISTINCT (s.staffFname + ' ' + s.staffLname) AS staffFullName
FROM Beauty_Booking b
JOIN Beauty_Therapy t ON b.bookingID = t.bookingID
JOIN Beauty_Staff s ON t.performStaffID = s.staffID
WHERE bookedDate = '2024-11-11';
```



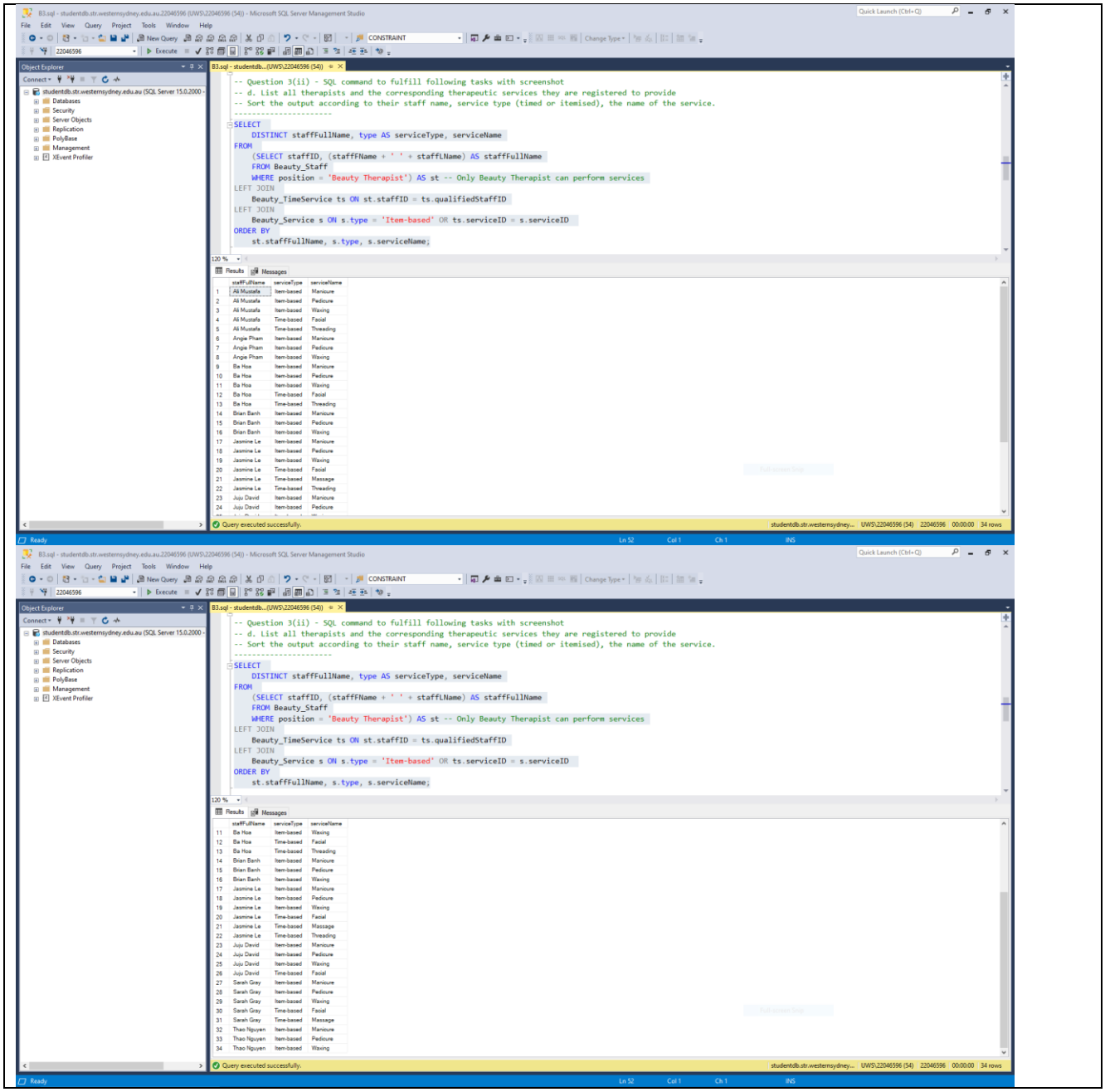
c. List all the names of the clients along with the corresponding total numbers of bookings.

```sql
SELECT (clientFname + ' ' + clientLname) AS clientName,
       COUNT(clientFname + ' ' + clientLname)
FROM Beauty_Booking b
LEFT JOIN Beauty_Client c ON c.clientID = b.clientID
WHERE b.clientID != 'Walk-in' -- Excluding Walk-in clients
GROUP BY (clientFname + ' ' + clientLname);
```



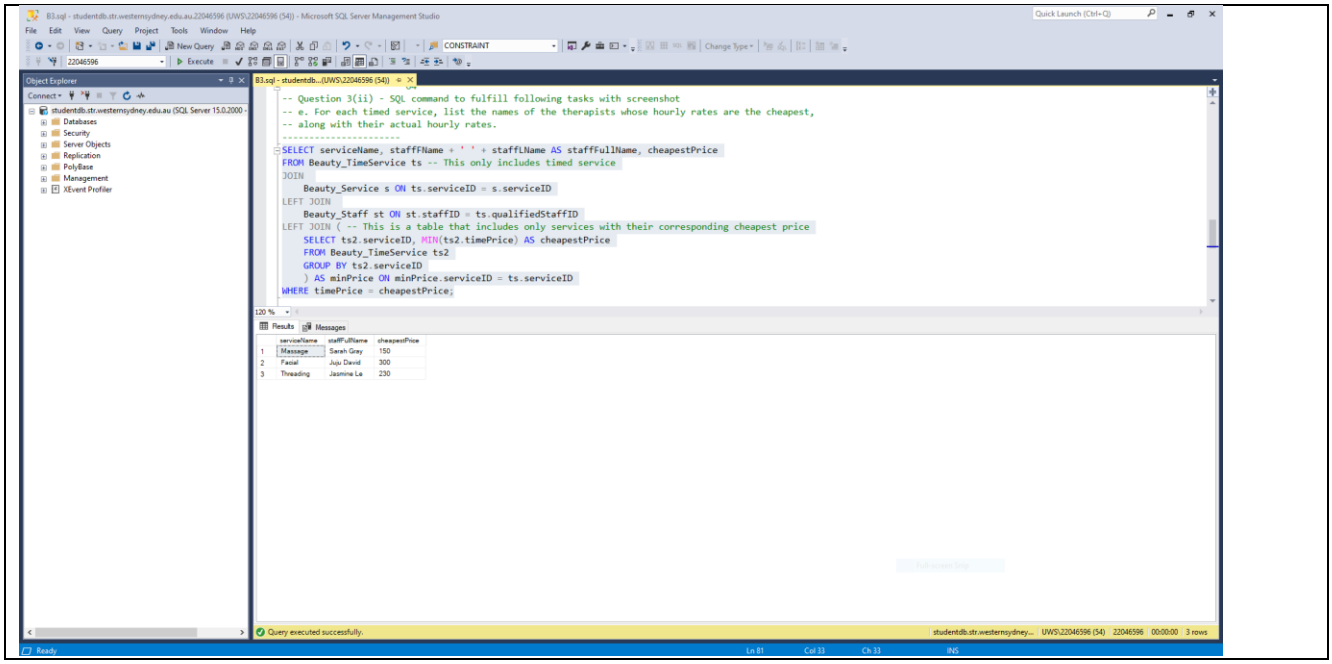
d. List all therapist and all services that they can perform. List alphabetically according to staff name, service type, name of service.

```
SELECT
    DISTINCT staffFullName, type AS serviceType, serviceName
FROM
    (SELECT staffID, (staffFName + ' ' + staffLName) AS staffFullName
       FROM Beauty_Staff
       WHERE position = 'Beauty Therapist') AS st -- Only Beauty Therapist can perform
services
LEFT JOIN
    Beauty_TimeService ts ON st.staffID = ts.qualifiedStaffID
LEFT JOIN
    Beauty_Service s ON s.type = 'Item-based' OR ts.serviceID = s.serviceID
ORDER BY
    st.staffFullName, s.type, s.serviceName;
```



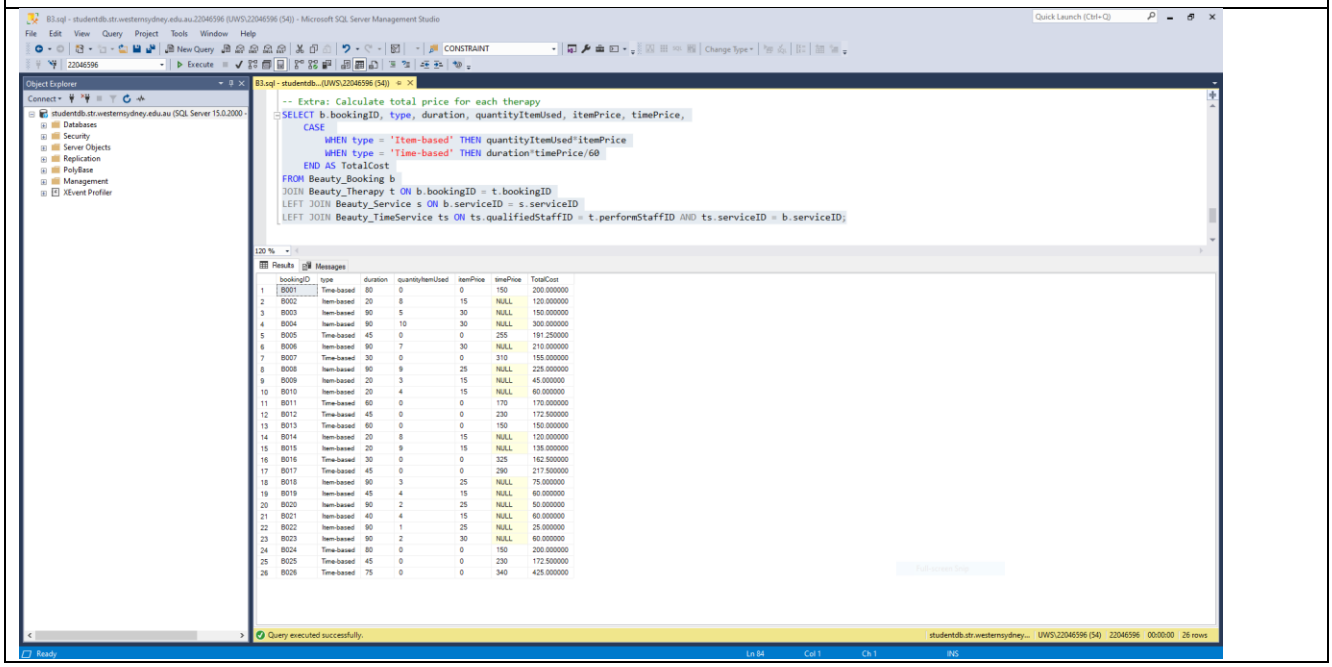
e. For each timed service, list the names of the therapist whose hourly rate are the cheapest along with their actual hourly rates.

```
SELECT serviceName, staffFName + ' ' + staffLName AS staffFullName, cheapestPrice
FROM Beauty_TimeService ts -- This only includes timed service
JOIN
	Beauty_Service s ON ts.serviceID = s.serviceID
LEFT JOIN
	Beauty_Staff st ON st.staffID = ts.qualifiedStaffID
LEFT JOIN ( -- This is a table that includes only services with their corresponding cheapest price
	SELECT ts2.serviceID, MIN(ts2.timePrice) AS cheapestPrice
	FROM Beauty_TimeService ts2
	GROUP BY ts2.serviceID
	) AS minPrice ON minPrice.serviceID = ts.serviceID
WHERE timePrice = cheapestPrice;
```



Extra query: Calculate the cost of the therapy (base for later use)

```
SELECT b.bookingID, type, duration, quantityItemUsed, itemPrice, timePrice,
       CASE
        WHEN type = 'Item-based' THEN quantityItemUsed*itemPrice
        WHEN type = 'Time-based' THEN duration*timePrice/60
    END AS TotalCost
FROM Beauty_Booking b
JOIN Beauty_Therapy t ON b.bookingID = t.bookingID
LEFT JOIN Beauty_Service s ON b.serviceID = s.serviceID
LEFT JOIN Beauty_TimeService ts ON ts.qualifiedStaffID = t.performStaffID AND ts.serviceID =
b.serviceID;
```



iii. Find a scenario in which a relatively prominent business data integrity can not be ensured by current and foreign keys. Write a SQL statement that will determine is such a problem exists or not. (Screenshot)

Scenario 1: Assigning 'not' qualified therapist to perform a time-based service.

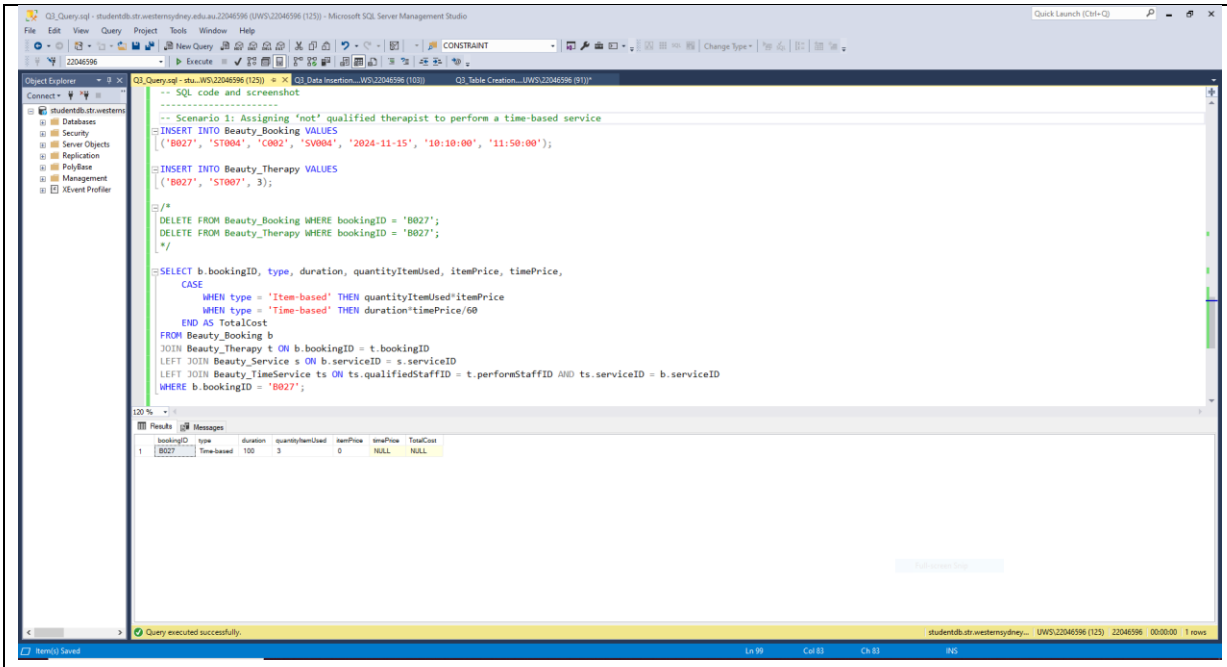
Reason:

- Table Beauty_Booking use Beauty_Service table for referencing serviced, that could be either item-based or time-based
- Meanwhile, table Beauty_Therapy use Beauty_Staff table for referencing performStaffID, that could be either staff can perform all services or staff can do specific services
- When joining Beauty_Booking and Beauty_Therapy, it could appear a time-based service carried on by an incapable staff (manager, receptionist, licensee) or staff who hasn't been trained for that time-service.
- This will lead to a crash in the query that calculate bill for a booking.

In the following code, when I create a new booking and assign staff to that booking, I make a mistake.

- Service 'SV004' can only be carried on by 'ST011' and 'ST012'
- However, I put Staff 'ST007' to this booking.
- This leads to an impossibility in delivering service as well as calculating final bill.
- Meanwhile, we can anticipate this error by running the final bill.
- If the bill is shown as below (NULL cost), assigned staff is not capable to deliver service that client wants.

```
INSERT INTO Beauty_Booking VALUES
('B027', 'ST004', 'C002', 'SV004', '2024-11-15', '10:10:00', '11:50:00');

INSERT INTO Beauty_Therapy VALUES
('B027', 'ST007', 3);

/*
DELETE FROM Beauty_Booking WHERE bookingID = 'B027';
DELETE FROM Beauty_Therapy WHERE bookingID = 'B027';
*/

SELECT b.bookingID, type, duration, quantityItemUsed, itemPrice, timePrice,
       CASE
        WHEN type = 'Item-based' THEN quantityItemUsed*itemPrice
        WHEN type = 'Time-based' THEN duration*timePrice/60
    END AS TotalCost
FROM Beauty_Booking b
JOIN Beauty_Therapy t ON b.bookingID = t.bookingID
LEFT JOIN Beauty_Service s ON b.serviceID = s.serviceID
LEFT JOIN Beauty_TimeService ts ON ts.qualifiedStaffID = t.performStaffID AND ts.serviceID =
b.serviceID
WHERE b.bookingID = 'B027';
```



Scenario 2: Booking staff allow client to book overlapping timeslot.

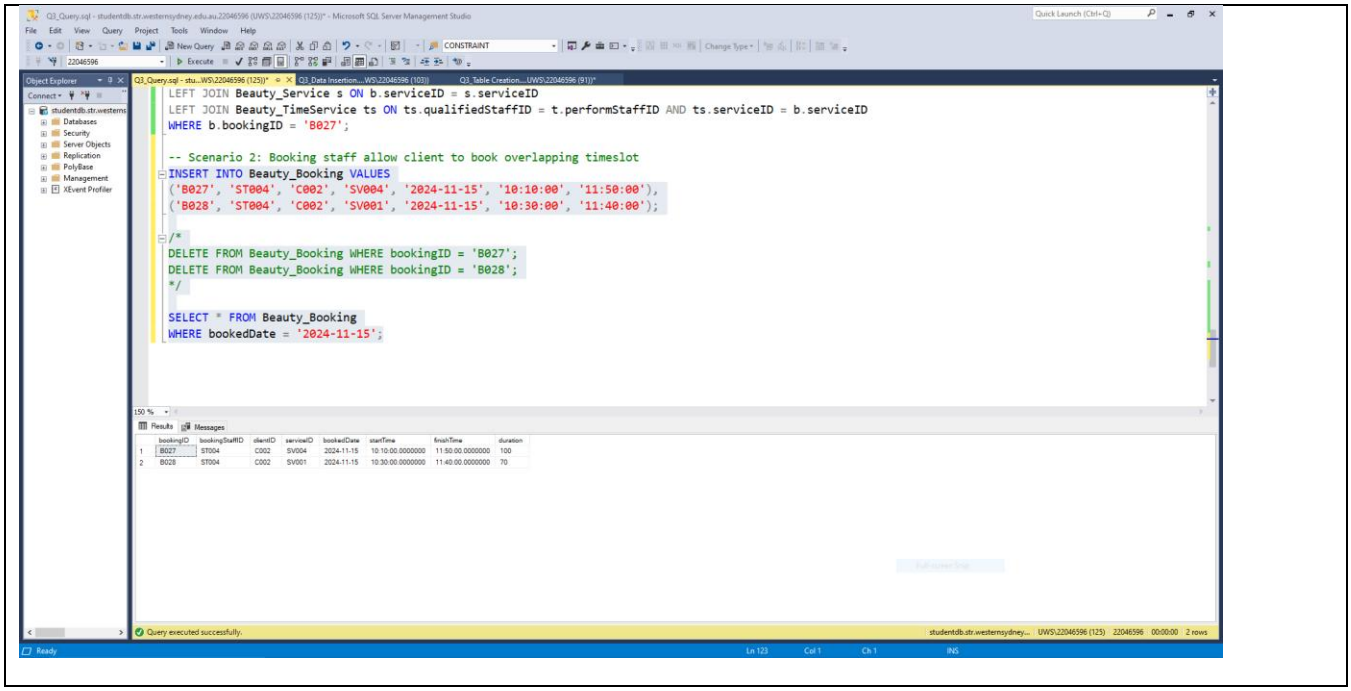
Reason:

- The current constraint in table Beauty_Booking is "`UNIQUE (clientID, bookedDate, startTime, finishTime)`"
- This allow to have unique time start in a day and finish time for that particular session
- However, this one does not provide any limit in booking another service within that timeslot in the previous booking.
- Although there is no error warning shown and still let staff put wrong timeslot in, this would result in that a client cannot experience 2 services at the same time.

In the following code, when I create two new booking for the same client but insert the wrong timeslot for them.

- It is obvious that the timeslot in the booking 'B028' is conflicted with the timeslot in 'B027'
- This leads to the confusion when I assign staff for that client.
- When the staff of booking 'B028' show up, the client is still in service, leading to the staff has no idea what to do next.

```
INSERT INTO Beauty_Booking VALUES
('B027', 'ST004', 'C002', 'SV004', '2024-11-15', '10:10:00', '11:50:00'),
('B028', 'ST004', 'C002', 'SV001', '2024-11-15', '10:30:00', '11:40:00');

/*
DELETE FROM Beauty_Booking WHERE bookingID = 'B027';
DELETE FROM Beauty_Booking WHERE bookingID = 'B028';
*/

SELECT * FROM Beauty_Booking
WHERE bookedDate = '2024-11-15';
```


LEFT JOIN Beauty_Service s ON b.serviceID = s.serviceID
LEFT JOIN Beauty_TimeService ts ON ts.qualifiedStaffID = t.performStaffID AND ts.serviceID = b.serviceID
WHERE b.bookingID = 'B027';
-- Scenario 2: Booking staff allow client to book overlapping timeslot
INSERT INTO Beauty_Booking VALUES
('B027', 'ST004', 'C002', 'SV004', '2024-11-15', '10:10:00', '11:50:00'),
('B028', 'ST004', 'C002', 'SV001', '2024-11-15', '10:30:00', '11:40:00');
/*
DELETE FROM Beauty_Booking WHERE bookingID = 'B027';
DELETE FROM Beauty_Booking WHERE bookingID = 'B028';
*/
SELECT * FROM Beauty_Booking
WHERE bookedDate = '2024-11-15';
Results
Messages
Query executed successfully.
Ready