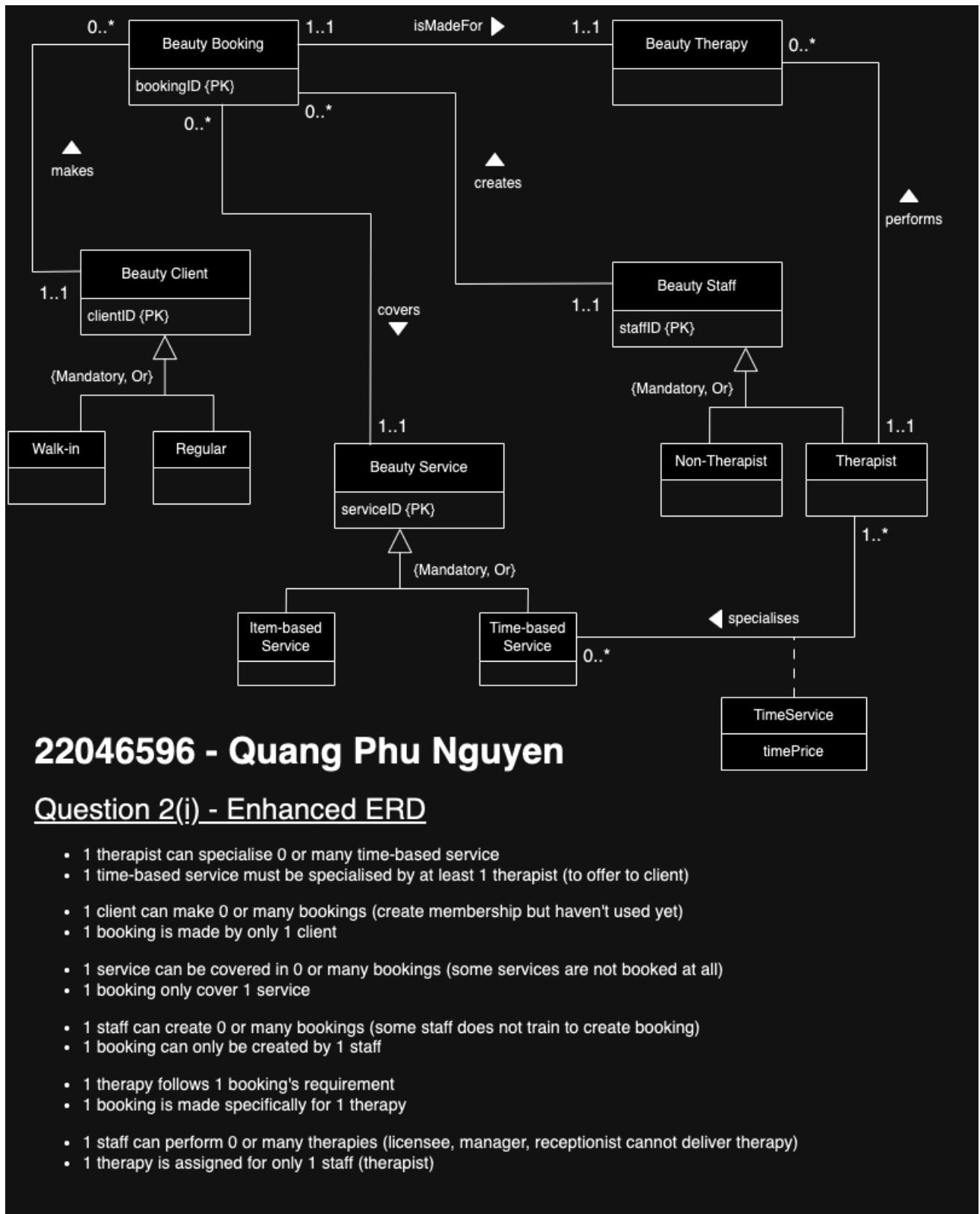
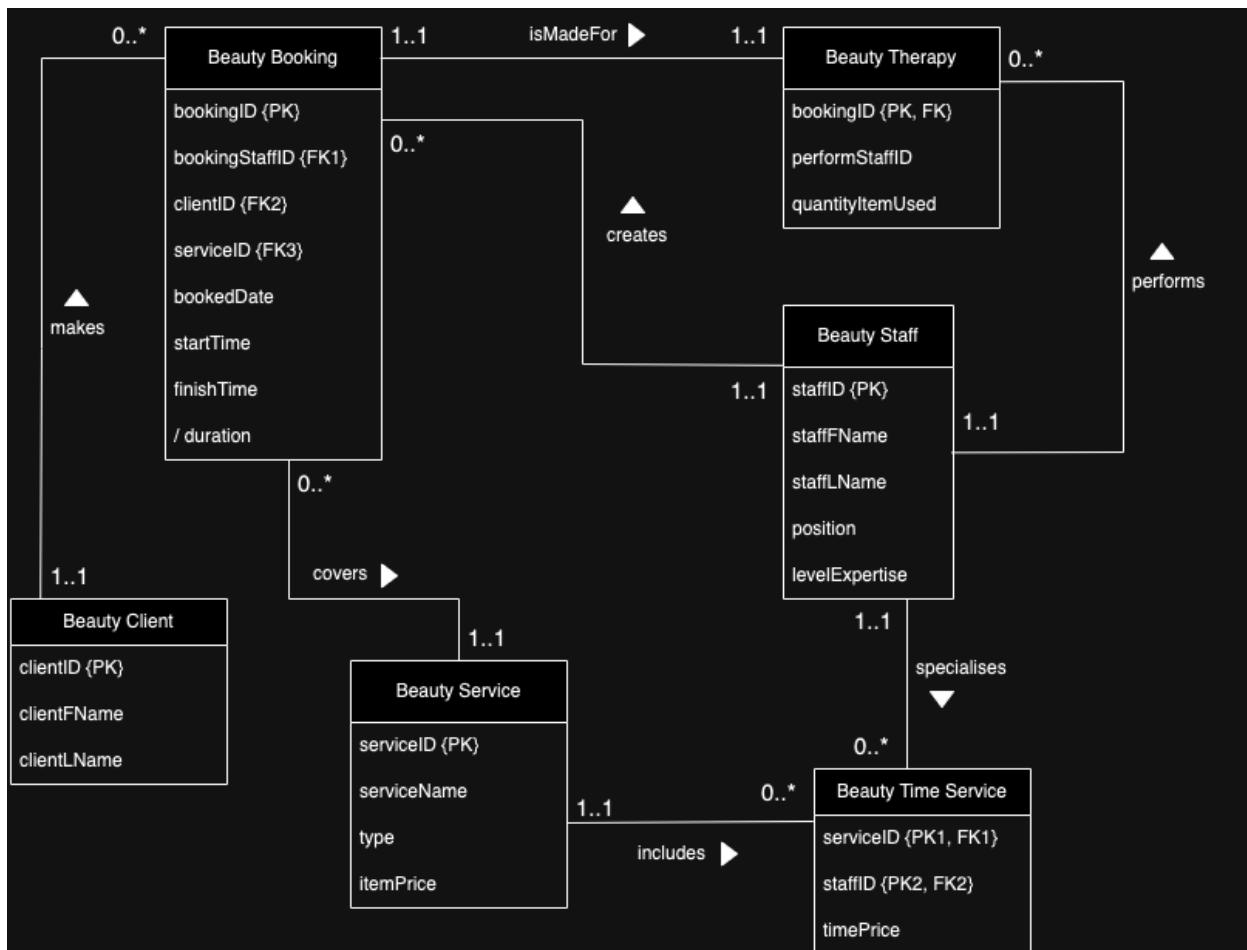


Database modelling

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



ii. Global Relation Diagram (GRD)



22046596 - Quang Phu Nguyen

Question 2(ii) - GRD (Global Relation Diagram)

- 1 therapist can specialise 0 or many time-based service (some staffs do not know to performs special/time-based service)
 - 1 time-based service price is matched with only 1 therapist (depend on which staff performing service)
 - 1 service can be related to 0 or many time-based service (because it could be item-based service)
 - 1 time-based service is matched with 1 entry in the general service table.
 - 1 client can make 0 or many bookings (create membership but haven't used yet)
 - 1 booking is made by only 1 client
 - 1 service can be covered in 0 or many bookings (some services are not booked at all)
 - 1 booking only cover 1 service
 - 1 staff can create 0 or many bookings (some staff does not train to create booking)
 - 1 booking can only be created by 1 staff
 - 1 therapy follows 1 booking's requirement
 - 1 booking is made specifically for 1 therapy
 - 1 staff can perform 0 or many therapies (licensee, manager, receptionist cannot deliver therapy)
 - 1 therapy is assigned for only 1 staff (therapist)

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;
```

The screenshot shows the Microsoft SQL Server Management Studio interface with two result grids:

Beauty_Staff

staffID	staffName	position	levelExperience
1	Le	Licenced	NULL
2	Bé	Thuy	Manager
3	Quin	Quen	Receptionist
4	Gang	Yang	Receptionist
5	ST005	Ba	Beauty Therapist
6	ST006	David	Beauty Therapist
7	ST007	Ak	Mustela
8	ST008	Thao	Beauty Therapist
9	ST009	Brian	Banh
10	ST010	Angie	Pham
11	ST011	Sarah	Gray
12	ST012	Jasmine	Le

Beauty_Client

clientID	clientName	clientName
1	Chadie	Nguyen
2	C0002	Ts
3	C0003	Ken
4	C0004	Lin
5	C0005	Hung
6	C0006	Lu
7	C0007	Nguyen
8	C0008	Le
9	C0009	Vy
10	C0010	Thao
11	C0011	Cam
12	C0012	Man
13	C0013	Quan
14	C0014	De
15	C0015	Vy
16	C0016	Tran
17	C0017	Rui
18	Walk-in	NULL

B3.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect + studentdb.str.westernsydney.edu.au (SQL Server 15.0.2000...)

Databases Security Server Objects Replication PolyBase Management Event Profiler

Results Messages

```
-- 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;
```

serviceID	serviceName	type	itemPrice
SV001	Manicure	Item-based	25
SV002	Pedicure	Item-based	30
SV003	Waxing	Item-based	15
SV004	Massage	Time-based	0
SV005	Facial	Time-based	0
SV006	Threading	Time-based	0

qualifiedStaffID	serviceID	timePrice
ST005	SV005	345
ST005	SV006	290
ST006	SV005	300
ST007	SV003	310
ST007	SV006	395
ST011	SV004	150
ST011	SV005	325
ST012	SV004	170
ST012	SV005	340
ST012	SV006	230

Query executed successfully.

studentdb.str.westernsydney... UWS:22046596 (54) 22046596 00:00:00 16 rows

Ln 8 Col 1 Ch 1 IN5

B3.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (54)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect + studentdb.str.westernsydney.edu.au (SQL Server 15.0.2000...)

Databases Security Server Objects Replication PolyBase Management Event Profiler

Results Messages

```
-- 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;
```

bookingID	bookingStaffID	clientID	serviceID	bookedDate	startTIme	endTIme	duration	
B001	ST004	CD01	SV004	2024-10-08	11:00:00.0000000	11:20:00.0000000	20	
2	ST003	CD02	SV003	2024-10-08	11:30:00.0000000	12:30:00.0000000	60	
3	ST003	CD03	SV002	2024-10-08	13:45:00.0000000	12:15:00.0000000	90	
4	ST004	CD04	SV005	2024-10-08	11:30:00.0000000	13:00:00.0000000	90	
5	ST005	ST009	CD17	SV006	2024-10-09	09:00:00.0000000	09:45:00.0000000	45
6	ST006	ST009	CD13	SV002	2024-10-09	09:00:00.0000000	11:30:00.0000000	90
7	ST007	ST009	CD12	SV006	2024-10-09	09:45:00.0000000	12:30:00.0000000	90
8	ST008	ST004	SV001	2024-10-09	12:30:00.0000000	14:00:00.0000000	90	
9	ST009	ST003	CD10	SV003	2024-10-10	10:00:00.0000000	10:20:00.0000000	20
10	ST010	ST003	CD11	SV003	2024-10-10	11:00:00.0000000	11:20:00.0000000	20
11	ST011	ST004	SV004	2024-10-10	10:00:00.0000000	11:00:00.0000000	60	
12	ST012	ST009	CD08	SV006	2024-10-10	12:00:00.0000000	12:45:00.0000000	45
13	ST013	ST004	SV004	2024-10-10	13:45:00.0000000	14:45:00.0000000	60	
14	ST014	ST003	CD04	SV005	2024-10-10	11:30:00.0000000	11:50:00.0000000	20
15	ST015	ST003	CD07	SV005	2024-10-10	10:00:00.0000000	10:20:00.0000000	20
16	ST016	ST005	CD15	SV005	2024-10-10	11:30:00.0000000	12:00:00.0000000	30
17	ST017	ST006	CD04	SV001	2024-10-10	12:00:00.0000000	12:45:00.0000000	45
18	ST018	ST003	CD01	SV001	2024-10-10	13:30:00.0000000	13:00:00.0000000	90
19	ST019	ST005	CD02	SV003	2024-10-10	10:45:00.0000000	10:45:00.0000000	45
20	ST020	ST010	CD17	SV001	2024-11-11	11:30:00.0000000	13:00:00.0000000	90
21	ST021	ST010	CD17	SV003	2024-11-11	13:10:00.0000000	13:50:00.0000000	40
22	ST022	ST003	CD17	SV001	2024-11-11	12:00:00.0000000	12:45:00.0000000	45
23	ST023	ST004	SV002	2024-11-11	13:30:00.0000000	13:00:00.0000000	90	
24	ST024	ST003	CD13	SV004	2024-11-11	11:30:00.0000000	12:30:00.0000000	80
25	ST025	ST003	CD12	SV006	2024-11-14	10:45:00.0000000	11:30:00.0000000	45
26	ST026	ST003	CD12	SV005	2024-11-14	11:30:00.0000000	12:45:00.0000000	75

Query executed successfully.

studentdb.str.westernsydney... UWS:22046596 (54) 22046596 00:00:00 26 rows

Ln 10 Col 1 Ch 1 IN5

B3.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (54)) - Microsoft SQL Server Management Studio

-- Question 3(i) : SQL command to fulfill following tasks with screenshot
-- a. 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;

```

Results

bookingID	performStaffID	quantityItemUsed
1	ST001	0
2	ST002	9
3	ST003	5
4	ST004	10
5	ST005	0
6	ST006	7
7	ST007	0
8	ST008	9
9	ST009	3
10	ST010	4
11	ST011	0
12	ST012	0
13	ST013	0
14	ST014	8
15	ST015	9
16	ST016	0
17	ST017	0
18	ST018	3
19	ST019	4
20	ST020	2
21	ST021	4
22	ST022	1
23	ST023	2
24	ST024	0
25	ST025	0
26	ST026	0

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;

```

B3.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (54)) - Microsoft SQL Server Management Studio

-- Question 3(ii) : SQL command to fulfill following tasks with screenshot
-- a. List all the timed services along with the therapists who can provide such services.
-- The list should be sorted alphabetically in the 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;

```

-- Question 3(iii) : SQL command to fulfill following tasks with screenshot
-- b. For a given day, (2024-11-11) list all the names of the therapists who have/had
-- at least one booking on that day. Don't repeat the names in the list.

Results

serviceName	staffFullName
Facial	Ba Hoo
Facial	Joule Smith
Facial	Ali Mustafa
Facial	Sarah Gray
Facial	Jasmine Le
Massage	Sarah Gray
Massage	Jasmine Le
Threading	Ba Hoo
Threading	Ali Mustafa
Threading	Jasmine Le

Query executed successfully.

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

```
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';
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window is open with the following content:

```
-- Question 3(i) - SQL command to fulfill following tasks with screenshot
-- b. For a given day, (2024-11-11) list all the names of the therapists who have/had
-- at least one booking on that day. Don't repeat the names in the list.

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';
```

Below the query, the results pane shows the output:

staffFullName
Brian Banks
Sarah Gray
Thao Nguyen

The status bar at the bottom indicates "Query executed successfully." and "3 rows".

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

```
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);
```

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "B3.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (54)) - Microsoft SQL Server Management Studio". The main area contains a query window with the following SQL code:

```

-- Question 3(iii) - SQL command to fulfill following tasks with screenshot
-- c. list the names of all the clients along with the corresponding total number of bookings.

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);

-- Question 3(iv) - SQL command to fulfill following tasks with screenshot
-- d. List all therapists and the corresponding therapeutic services they are registered to provide
-- Sort the output according to their staff name, service type (timed or itemised), the name of the service.

```

The results pane displays a table with 13 rows of client names and their counts:

clientName	(No column name)
1 Charlie Nguyen	3
2 Cuong Nguyen	2
3 Dang Ngan	2
4 Hoa Kim	1
5 Hung Pham	1
6 Lu Nguyen	1
7 Man Quan	1
8 Nhat	1
9 Sue Nguyen	2
10 Thien Da	1
11 Tuoi Bai	4
12 Vy Dang	1
13 Vy Ta	1

At the bottom of the results pane, it says "Query executed successfully." and shows statistics: In 45, Cn 44, Ch 44, INS.

- 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;

```

```

-- Question 3(iii) - SQL command to fulfill following tasks with screenshot
-- d. List all therapists and the corresponding therapeutic services they are registered to provide
-- Sort the output according to their staff name, service type (timed or itemised), the name of the 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;

```

staffFullName	serviceType	serviceName
All Mustafa	Item-based	Manicure
All Mustafa	Item-based	Pedicure
All Mustafa	Item-based	Waxing
All Mustafa	Time-based	Facial
All Mustafa	Time-based	Threading
Angie Pham	Item-based	Manicure
Angie Pham	Item-based	Pedicure
Angie Pham	Item-based	Waxing
Ba Hoi	Item-based	Manicure
Ba Hoi	Item-based	Pedicure
Ba Hoi	Item-based	Waxing
Ba Hoi	Time-based	Facial
Ba Hoi	Time-based	Threading
Brian Banks	Item-based	Manicure
Brian Banks	Item-based	Pedicure
Brian Banks	Item-based	Waxing
Brian Banks	Time-based	Facial
Jasmine Le	Item-based	Manicure
Jasmine Le	Item-based	Pedicure
Jasmine Le	Item-based	Waxing
Jasmine Le	Time-based	Facial
Jasmine Le	Time-based	Threading
Juju David	Item-based	Manicure
Juju David	Item-based	Pedicure
Juju David	Time-based	Facial
Sarah Gray	Item-based	Manicure
Sarah Gray	Item-based	Pedicure
Sarah Gray	Item-based	Waxing
Sarah Gray	Time-based	Facial
Thao Nguyen	Item-based	Manicure
Thao Nguyen	Item-based	Pedicure
Thao Nguyen	Item-based	Waxing


```

-- Question 3(iii) - SQL command to fulfill following tasks with screenshot
-- d. List all therapists and the corresponding therapeutic services they are registered to provide
-- Sort the output according to their staff name, service type (timed or itemised), the name of the 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;

```

staffFullName	serviceType	serviceName
Ba Hoi	Item-based	Waxing
Ba Hoi	Time-based	Facial
Ba Hoi	Time-based	Threading
Brian Banks	Item-based	Manicure
Brian Banks	Item-based	Pedicure
Brian Banks	Item-based	Waxing
Brian Banks	Time-based	Facial
Jasmine Le	Item-based	Manicure
Jasmine Le	Item-based	Pedicure
Jasmine Le	Item-based	Waxing
Jasmine Le	Time-based	Facial
Jasmine Le	Time-based	Threading
Juju David	Item-based	Manicure
Juju David	Item-based	Pedicure
Juju David	Time-based	Facial
Sarah Gray	Item-based	Manicure
Sarah Gray	Item-based	Pedicure
Sarah Gray	Item-based	Waxing
Sarah Gray	Time-based	Facial
Thao Nguyen	Item-based	Manicure
Thao Nguyen	Item-based	Pedicure
Thao Nguyen	Item-based	Waxing

- 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;

```

```

-- Question 3(iii) - SQL command to fulfill following tasks with screenshot
-- e. For each timed service, list the names of the therapists whose hourly rates are the cheapest,
-- along with their actual hourly rates.

SELECT serviceName, staffName = t.staffName 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;

```

Results

	serviceName	staffName	cheapestPrice
1	Massage	Sara Grey	150
2	Facial	Juju David	300
3	Threading	Jasmine Lee	250

Query executed successfully.

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;

```

```

-- Extra: Calculate total price for each therapy
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;

```

Results

bookingID	duration	quantityItemUsed	itemPrice	timePrice	TotalCost
1	8001	Time-based	80	15	200.00000
2	8002	Item-based	20	8	120.00000
3	8003	Item-based	90	5	150.00000
4	8004	Item-based	90	10	300.00000
5	8005	Time-based	75	0	25.00000
6	8006	Item-based	90	7	30.00000
7	8007	Time-based	30	0	310.00000
8	8008	Item-based	90	9	25.00000
9	8009	Item-based	20	3	45.00000
10	8010	Time-based	75	4	15.00000
11	8011	Time-based	60	0	170.00000
12	8012	Time-based	45	0	172.50000
13	8013	Time-based	60	0	150.00000
14	8014	Item-based	20	8	150.00000
15	8015	Time-based	75	15	15.00000
16	8016	Time-based	30	0	325.00000
17	8017	Time-based	45	0	290.00000
18	8018	Item-based	90	3	25.00000
19	8019	Item-based	45	4	15.00000
20	8020	Time-based	75	2	15.00000
21	8021	Item-based	40	4	15.00000
22	8022	Item-based	90	1	25.00000
23	8023	Item-based	90	2	30.00000
24	8024	Time-based	80	0	150.00000
25	8025	Time-based	45	0	250.00000
26	8026	Time-based	75	0	340.00000

Query executed successfully.

- 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 if such a problem exists or not. (Screenshot)

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

Reason:

- Table Beauty_Booking uses Beauty_Service table for referencing service, that could be either item-based or time-based
- Meanwhile, table Beauty_Therapy uses 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 calculates 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';

```

The screenshot shows a Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "Q3_Query.sql - studentdb.str.westernsydney.edu.au:22046596 (UWS:22046596 (125)) - Microsoft SQL Server Management Studio". The main area contains a query window with the following T-SQL code:

```

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

```

The results pane shows a single row of data:

bookingID	type	duration	quantityItemUsed	itemPrice	timePrice	TotalCost
1	B027	100	3	0	NULL	NULL

At the bottom of the results pane, it says "Query executed successfully." and "1 rows".

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

Reason:

- The current constraint in table `Beauty_Booking` is “`UNIQUE (clientID, bookedDate, startTime, endTime)`”
- 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';

```

Q3_Query.sql - studentdb.str.westernsydney.edu.au:22046598 (UWS:22046598 (125)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Execute Save As Constraint Change Type

Object Explorer Q3_Query.sql - stu...WS:22046598 (125) Q3_Data Insertion... WS:22046598 (103) Q3_Table Creation... UWS:22046598 (91)*

Connect Connect to database Connect to instance Connect to server Connect to master Connect to system master Connect to system catalog Connect to system stored procedure Connect to linked server Connect to linked master Connect to replicated database Connect to replicated master Connect to replicated catalog Connect to Polybase Connect to Management Connect to XEvent Profiler

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

	bookingID	bookingStaffID	clientID	serviceID	bookedDate	startTime	endTime	duration
1	B027	ST004	C002	SV004	2024-11-15	10:10:000000000	11:50:000000000	100
2	B028	ST004	C002	SV001	2024-11-15	10:30:000000000	11:40:000000000	70

Query executed successfully.

studentdb.str.westernsydney... UWS:22046598 (125) 22046598 00:00:00 2 rows

Ln 123 Col 1 Ch 1 INS

Ready