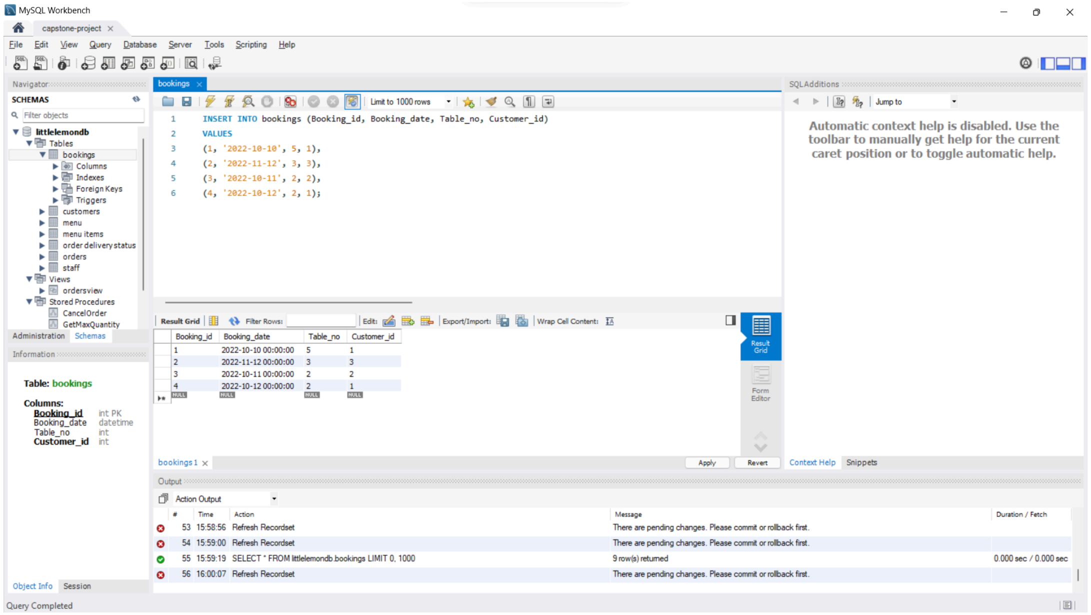
CREATE SQL QUERIES TO CHECK AVAILABLE BOOKINGS BASED ON USER INPUT

Task 1:

Your first task is to replicate the list of records in the following table by adding them to the Little Lemon booking table.



```sql
INSERT INTO bookings (Booking_id, Booking_date, Table_no, Customer_id)
VALUES
(1, '2022-10-10', 5, 1),
(2, '2022-11-12', 3, 3),
(3, '2022-10-11', 2, 2),
(4, '2022-10-12', 2, 1);
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Booking_id	Booking_date	Table_no	Customer_id
1	2022-10-10 00:00:00	5	1
2	2022-11-12 00:00:00	3	3
3	2022-10-11 00:00:00	2	2
4	2022-10-12 00:00:00	2	1
NULL	NULL	NULL	NULL

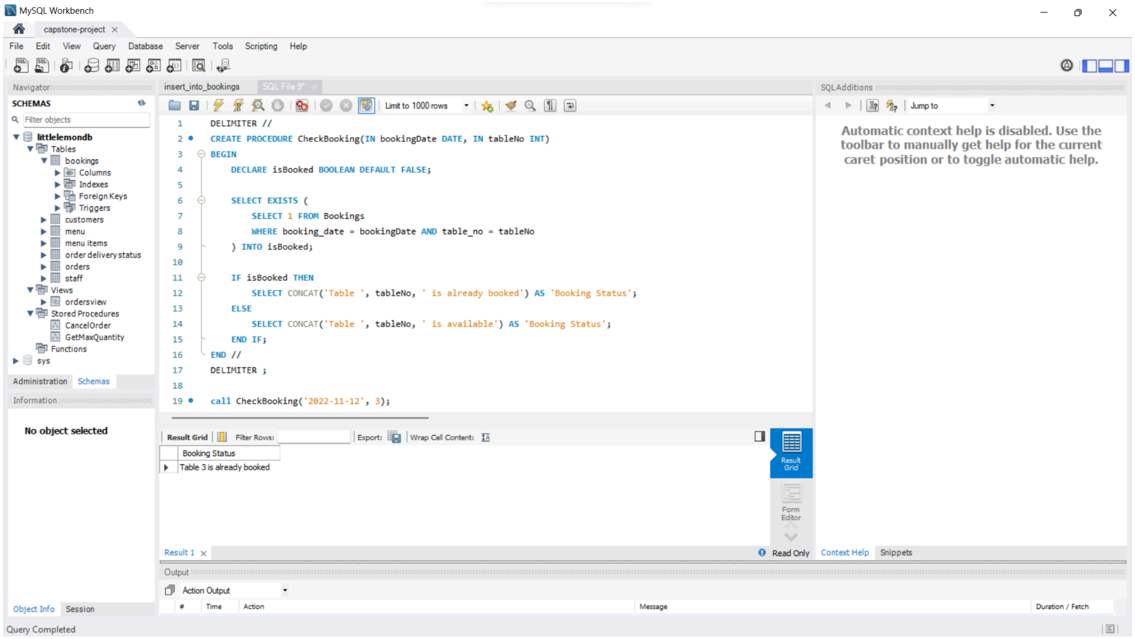
Table: bookings

Columns:
Booking_id int PK
Booking_date datetime
Table_no int
Customer_id int

```
#   Time      Action                                          Message                                            Duration / Fetch
53  15:58:56  Refresh Recordset                               There are pending changes. Please commit or rollback first.
54  15:59:00  Refresh Recordset                               There are pending changes. Please commit or rollback first.
55  15:59:19  SELECT * FROM littlelemondb.bookings LIMIT 0, 1000   9 row(s) returned                            0.000 sec / 0.000 sec
56  16:00:07  Refresh Recordset                               There are pending changes. Please commit or rollback first.
```

Task 2:

Little Lemon needs you to create a stored procedure called CheckBooking to check whether a table in the restaurant is already booked.



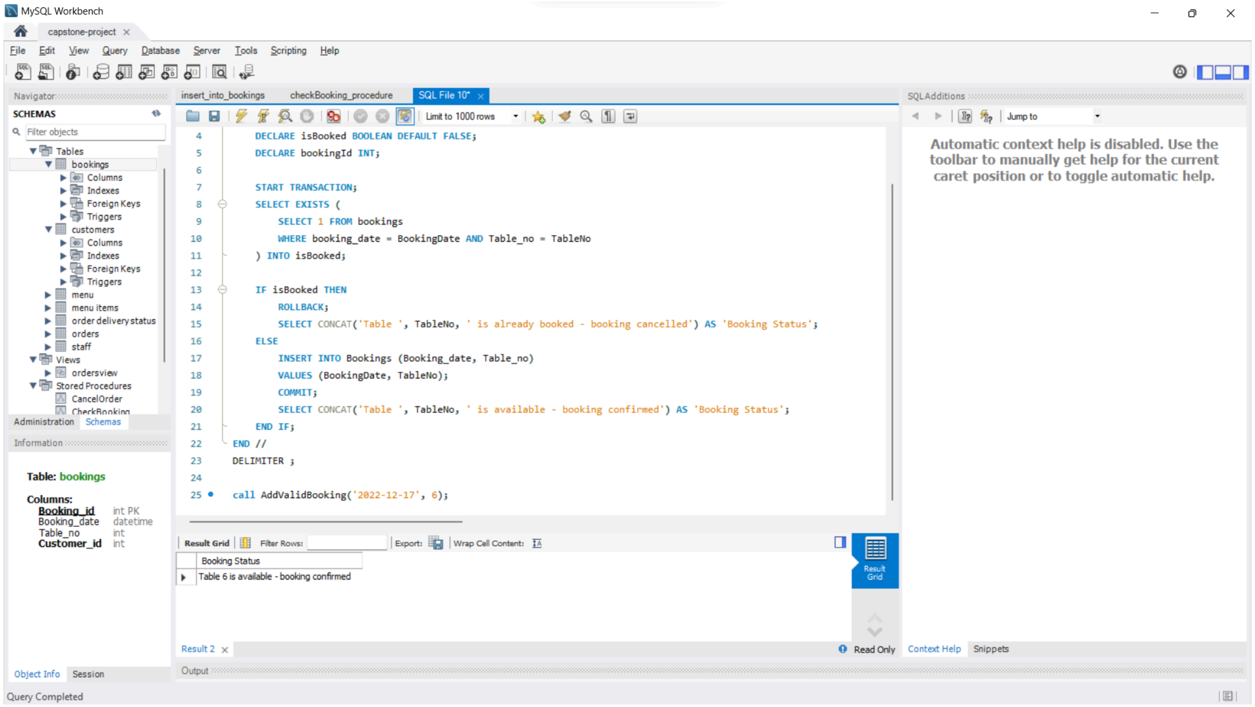
```sql
DELIMITER //
CREATE PROCEDURE CheckBooking(IN bookingDate DATE, IN tableNo INT)
BEGIN
    DECLARE isBooked BOOLEAN DEFAULT FALSE;

    SELECT EXISTS (
        SELECT 1 FROM Bookings
        WHERE booking_date = bookingDate AND table_no = tableNo
    ) INTO isBooked;

    IF isBooked THEN
        SELECT CONCAT('Table ', tableNo, ' is already booked') AS 'Booking Status';
    ELSE
        SELECT CONCAT('Table ', tableNo, ' is available') AS 'Booking Status';
    END IF;
END //
DELIMITER ;

call CheckBooking('2022-11-12', 3);
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Booking Status
Table 3 is already booked

Task 3:

You need to create a new procedure called AddValidBooking. This procedure must use a transaction statement to perform a rollback if a customer reserves a table that's already booked under another name.



```sql
    DECLARE isBooked BOOLEAN DEFAULT FALSE;
    DECLARE bookingId INT;

    START TRANSACTION;
    SELECT EXISTS (
        SELECT 1 FROM bookings
        WHERE booking_date = BookingDate AND Table_no = TableNo
    ) INTO isBooked;

    IF isBooked THEN
        ROLLBACK;
        SELECT CONCAT('Table ', TableNo, ' is already booked - booking cancelled') AS 'Booking Status';
    ELSE
        INSERT INTO Bookings (Booking_date, Table_no)
        VALUES (BookingDate, TableNo);
        COMMIT;
        SELECT CONCAT('Table ', TableNo, ' is available - booking confirmed') AS 'Booking Status';
    END IF;
END //
DELIMITER ;

call AddValidBooking('2022-12-17', 6);
```

Booking Status
Table 6 is available - booking confirmed

Table: bookings

Columns:

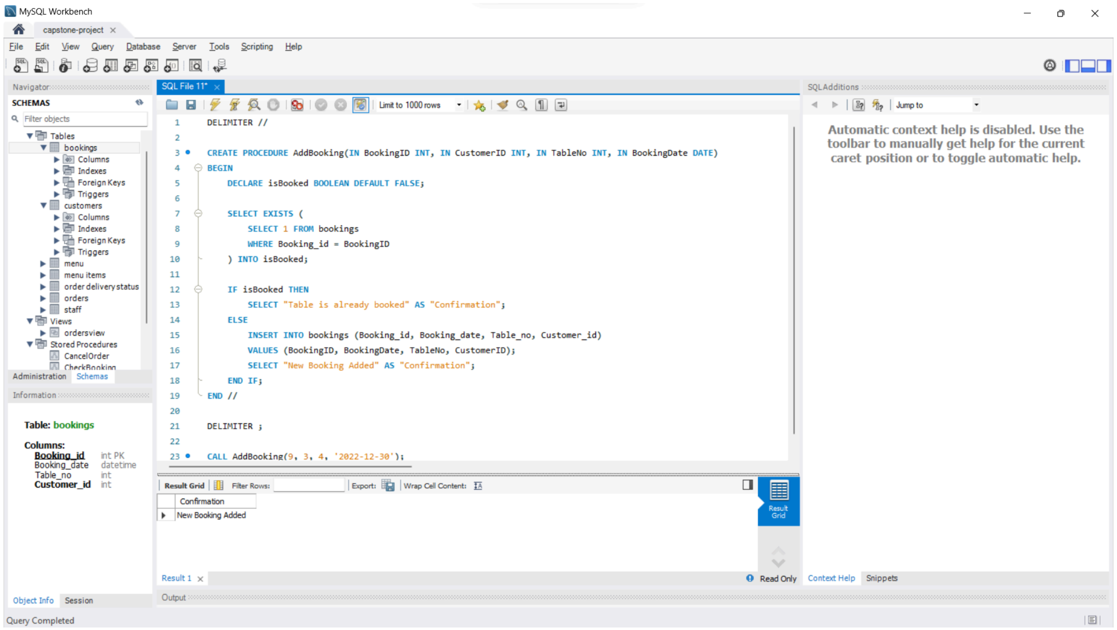
Booking_id	int PK
Booking_date	datetime
Table_no	int
Customer_id	int

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

CREATE SQL QUERIES TO ADD AND UPDATE BOOKINGS

Task 1:

In this first task you need to create a new procedure called AddBooking to add a new table booking record.



```
DELIMITER //

CREATE PROCEDURE AddBooking(IN BookingID INT, IN CustomerID INT, IN TableNo INT, IN BookingDate DATE)
BEGIN
    DECLARE isBooked BOOLEAN DEFAULT FALSE;

    SELECT EXISTS (
        SELECT 1 FROM bookings
        WHERE Booking_id = BookingID
    ) INTO isBooked;

    IF isBooked THEN
        SELECT "Table is already booked" AS "Confirmation";
    ELSE
        INSERT INTO bookings (Booking_id, Booking_date, Table_no, Customer_id)
        VALUES (BookingID, BookingDate, TableNo, CustomerID);
        SELECT "New Booking Added" AS "Confirmation";
    END IF;
END //

DELIMITER ;

CALL AddBooking(9, 3, 4, '2022-12-30');
```

```
Confirmation
New Booking Added
```

Table: bookings

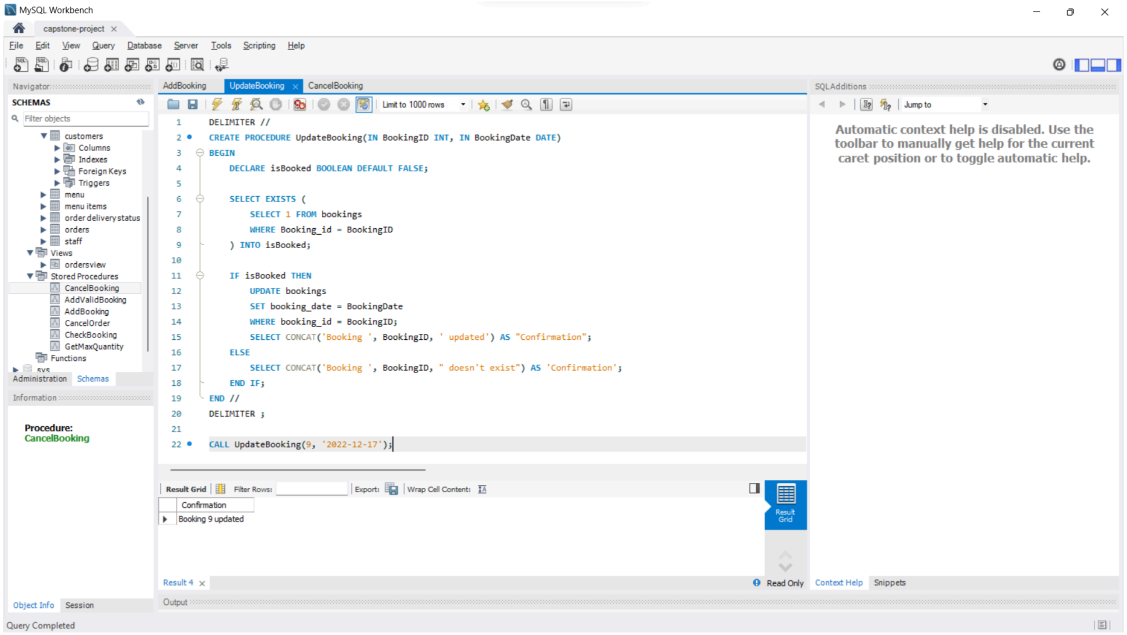
Columns:

Booking_id	int PK
Booking_date	datetime
Table_no	int
Customer_id	int

Query Completed

Task 2:

Little Lemon needs you to create a new procedure called UpdateBooking that they can use to update existing bookings in the booking table.



```
DELIMITER //
CREATE PROCEDURE UpdateBooking(IN BookingID INT, IN BookingDate DATE)
BEGIN
    DECLARE isBooked BOOLEAN DEFAULT FALSE;

    SELECT EXISTS (
        SELECT 1 FROM bookings
        WHERE Booking_id = BookingID
    ) INTO isBooked;

    IF isBooked THEN
        UPDATE bookings
        SET booking_date = BookingDate
        WHERE booking_id = BookingID;
        SELECT CONCAT('Booking ', BookingID, ' updated') AS "Confirmation";
    ELSE
        SELECT CONCAT('Booking ', BookingID, " doesn't exist") AS 'Confirmation';
    END IF;
END //
DELIMITER ;

CALL UpdateBooking(9, '2022-12-17');
```

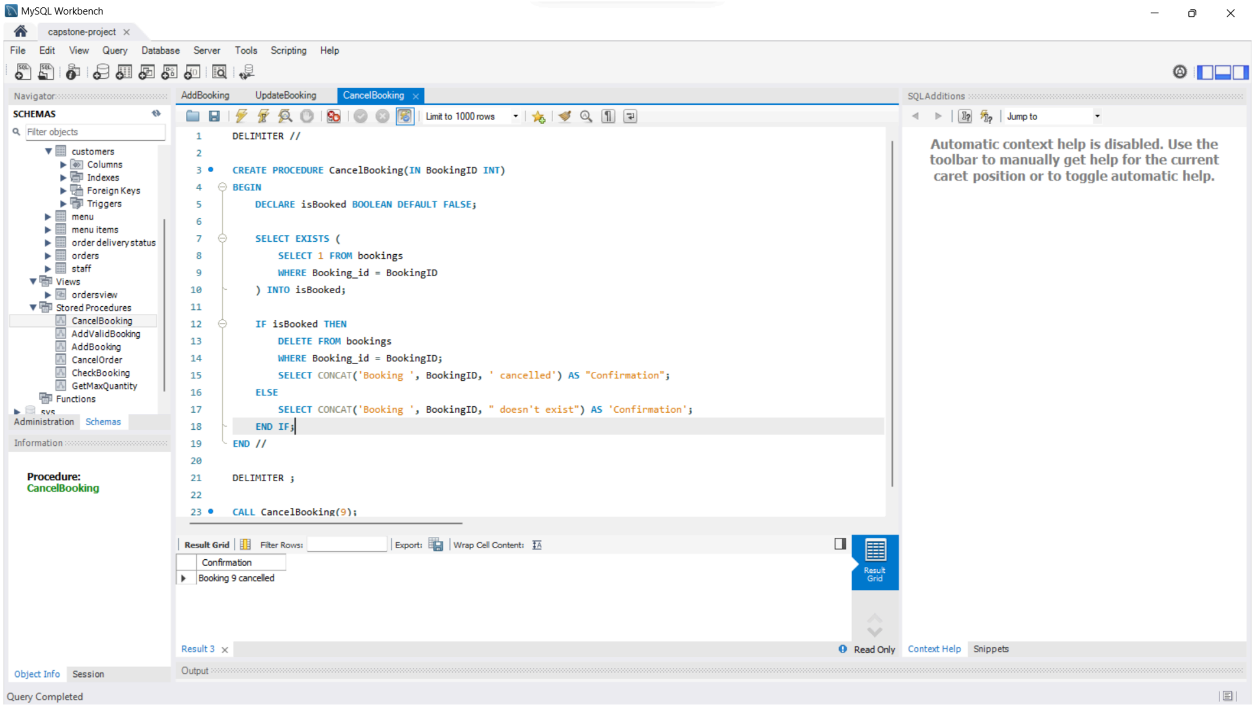
```
Confirmation
Booking 9 updated
```

Procedure:
CancelBooking

Query Completed

Task 3:

Little Lemon needs you to create a new procedure called CancelBooking that they can use to cancel or remove a booking.



```sql
DELIMITER //

CREATE PROCEDURE CancelBooking(IN BookingID INT)
BEGIN
    DECLARE isBooked BOOLEAN DEFAULT FALSE;

    SELECT EXISTS (
        SELECT 1 FROM bookings
        WHERE Booking_id = BookingID
    ) INTO isBooked;

    IF isBooked THEN
        DELETE FROM bookings
        WHERE Booking_id = BookingID;
        SELECT CONCAT('Booking ', BookingID, ' cancelled') AS "Confirmation";
    ELSE
        SELECT CONCAT('Booking ', BookingID, " doesn't exist") AS 'Confirmation';
    END IF;
END //

DELIMITER ;

CALL CancelBooking(9);
```

Confirmation
Booking 9 cancelled

Procedure: CancelBooking

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.