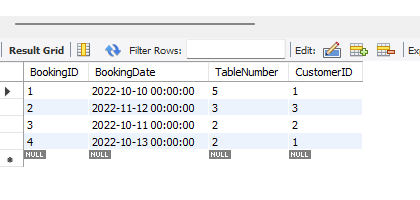
#Task1



#Task 2 create a stored procedure called **CheckBooking**

**CREATE DEFINER=`root`@`localhost` PROCEDURE `CheckBooking`(IN Booking\_Date DATETIME,IN Table\_Number INT)**

**BEGIN**

**DECLARE booking\_status INT;**

**-- Check if the table is already booked for the given date**

**SELECT COUNT(\*) INTO booking\_status**

**FROM littlelemon.bookings**

**WHERE littlelemon.bookings.BookingDate = Booking\_Date AND littlelemon.bookings.TableNumber=Table\_Number;**

**-- Display the status**

**IF booking\_status > 0 THEN**

**SELECT CONCAT('Table ', Table\_Number, ' is already booked. ') AS BookingStatus;**

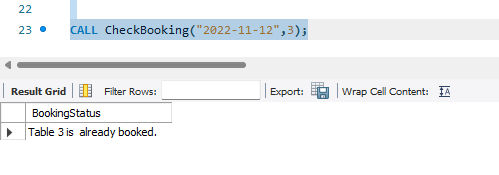
**ELSE**

**SELECT CONCAT('Table ', Table\_Number, ' is available.') AS BookingStatus;**

**END IF;**

**END**

**CALL CheckBooking("2022-11-12",3);**

****

#TASK 3 create a new procedure called **AddValidBooking**

**CREATE DEFINER=`root`@`localhost` PROCEDURE `AddValidBooking`(**

**IN booking\_date DATE,**

**IN table\_number INT**

**)**

**BEGIN**

**DECLARE booking\_status INT;**

**-- Start a transaction**

**START TRANSACTION;**

**-- Check if the table is already booked**

**SELECT COUNT(\*) INTO booking\_status**

**FROM littlelemon.bookings**

**WHERE littlelemon.bookings.BookingDate = booking\_date AND littlelemon.bookings.TableNumber = table\_number;**

**-- Conditional logic to handle the booking**

**IF booking\_status > 0 THEN**

**-- If the table is already booked, rollback the transaction**

**ROLLBACK;**

**SELECT CONCAT('Table ', table\_number, ' is already booked - booking cancelled.') AS BookingStatus;**

-- If the table is available, insert the new booking and commit the transaction

INSERT INTO bookings (BookingDate, TableNumber)

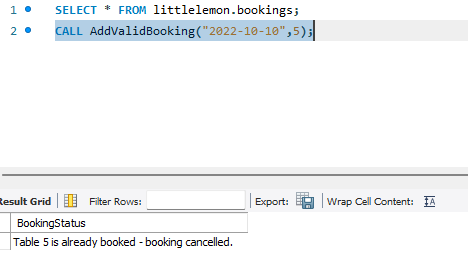
VALUES (booking\_date, table\_number);

COMMIT;

SELECT CONCAT('Table ', table\_number, ' is successfully booked.') AS BookingStatus;

**END IF;**

**END**

****