**Stored procedure to book tickets**

DROP PROCEDURE IF EXISTS SEAT\_BOOKING;

DELIMITER $$

# Assumption that only available seats are visible to the customer

CREATE PROCEDURE SEAT\_BOOKING (IN seats INTEGER, IN seat\_num VARCHAR(50), IN showid INTEGER, IN theatreid INTEGER, IN custid INTEGER)

BEGIN

DECLARE counter1 INTEGER;

DECLARE counter2 INTEGER;

DECLARE seatprice DECIMAL(8,2);

DECLARE book\_price DECIMAL(8,2);

DECLARE rowno VARCHAR(20);

DECLARE seatno INTEGER;

DECLARE seat VARCHAR(20);

DECLARE seatid INTEGER;

DECLARE bookid INTEGER;

DECLARE screenid INTEGER;

DECLARE rowno\_seat1 VARCHAR(20);

DECLARE old\_rowno VARCHAR(20);

DECLARE old\_seatno INTEGER;

DECLARE old\_seatid INTEGER;

SET counter1 = 1;

SET counter2 = 1;

SET book\_price = 0;

# Fetch the screen\_id for the given show\_id

SET screenid = (SELECT SCREEN\_ID FROM MOVIE\_SHOW WHERE SHOW\_ID = showid);

# Calculate the price for the given seats

seatloop: LOOP

IF counter1 <= seats THEN

SET seat = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat\_num,',',counter1), ',', -1);

SET rowno = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat,':',1), ':', -1);

SET seatno = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat,':',2), ':', -1);

# Fetch the seat id for the given combination of row no and seat no

SET seatid = (SELECT SEAT\_ID FROM SEAT WHERE ROW\_No = rowno AND SEAT\_No = seatno

AND SCREEN\_ID = screenid);

# Fetch the price for each price and add to the total booking price

SET seatprice = (SELECT SEAT\_Price FROM SEAT WHERE SEAT\_ID = seatid);

SET book\_price = book\_price + seatprice;

ELSE

LEAVE seatloop;

END IF;

SET counter1 = counter1 + 1;

END LOOP;

# Insert the booking details for the given number of seats

INSERT INTO BOOKING (No\_Of\_Seats,BOOKING\_STATUS,PRICE,BOOKING\_Time,SHOW\_ID,CUST\_ID) VALUES

(seats,'Success', book\_price, CURRENT\_TIMESTAMP, showid, custid);

# Fetch the book\_id for the inserted booking details

SET bookid = (SELECT BOOKING\_ID FROM BOOKING WHERE CUST\_ID = custid AND

DATE\_FORMAT(BOOKING\_Time,'%y-$m-$d %h:%i') = DATE\_FORMAT(NOW(),'%y-$m-$d %h:%i')

AND No\_Of\_Seats = seats AND SHOW\_ID = showid);

# update the payment table with price and the booking id to process the payment

CALL MAKE\_PAYMENT (bookid,book\_price);

# Loop is used to split the seat\_num to the necessary fields in the tables.

# seat\_num will be in the combination of format of 'rowno:seatno'

seatloop: LOOP

IF counter2 <= seats THEN

SET seat = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat\_num,',',counter2), ',', -1);

SET rowno = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat,':',1), ':', -1);

SET seatno = SUBSTRING\_INDEX(SUBSTRING\_INDEX(seat,':',2), ':', -1);

# Fetch the seat id for the given combination of row no and seat no

SET seatid = (SELECT SEAT\_ID FROM SEAT WHERE ROW\_No = rowno AND SEAT\_No = seatno

AND SCREEN\_ID = screenid);

# Insert the entries into show\_seat table to block the given seats for the corresponding show

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid,showid,bookid);

# Logic to implement blocking of adjacent seats for the given booking starts...

# Assume all the given seats are booked for the rowno of the first given seat.

IF counter2 = 1 THEN

SET rowno\_seat1 = rowno;

END IF;

IF rowno = rowno\_seat1 THEN

# Block the adjacent seats on the left side of the first seat of the given booking

IF counter2 = 1 THEN

IF seatno <> 1 AND seatno <> 2 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-1,showid,bookid);

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-2,showid,bookid);

ELSE

IF seatno = 2 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-1,showid,bookid);

END IF;

END IF;

END IF;

# Block the adjacent seats on the right side of the last seat of the given booking

IF counter2 = seats THEN

IF seatno <> 9 AND seatno <> 10 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+1,showid,bookid);

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+2,showid,bookid);

ELSE

IF seatno = 9 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+1,showid,bookid);

END IF;

END IF;

END IF;

# Store the previous row-seat combination for futher processing if required

SET old\_rowno = rowno;

SET old\_seatno = seatno;

SET old\_seatid = seatid;

ELSE

# Logic to implement if the customer books seats in different rows.

# Set the new rowno to the temporary variable

SET rowno\_seat1 = rowno;

# Block the adjacent seats on the right side of the previous row

IF old\_seatno <> 9 AND old\_seatno <> 10 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,old\_seatid+1,showid,bookid);

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,old\_seatid+2,showid,bookid);

ELSE

IF old\_seatno = 9 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,old\_seatid+1,showid,bookid);

END IF;

END IF;

# Block the adjacent seats on the left side of the seat in the new row

IF seatno <> 1 AND seatno <> 2 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-1,showid,bookid);

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-2,showid,bookid);

ELSE

IF seatno = 2 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid-1,showid,bookid);

END IF;

END IF;

# If this is the only seat in the new row, then block the adjacent seats on the right side

# of the seat.

IF counter2 = seats THEN

IF seatno <> 9 AND seatno <> 10 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+1,showid,bookid);

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+2,showid,bookid);

ELSE

IF seatno = 9 THEN

INSERT INTO SHOW\_SEAT (SHOW\_SEAT\_STATUS,SEAT\_ID,SHOW\_ID,BOOKING\_ID) VALUES

(0,seatid+1,showid,bookid);

END IF;

END IF;

END IF;

END IF;

ELSE

LEAVE seatloop;

END IF;

SET counter2 = counter2 + 1;

END LOOP;

END$$

DELIMITER ;