**CUSTOMER QUERIES**

1. **ACCOUNT PAGE**

Inputs: @username, @password, @passwordagain, @name, @birthdate, @passportno, @membershipname

**CREATE ACCOUNT**

INSERT INTO customer

VALUES(@username, @password, @name, @birthdate, GETDATE() - @birthdate, @passportno, 0, 0)

WHERE @password = @passwordagain

**ADD MEMBERSHIP**

INSERT INTO membership

VALUES(@username, @membershipname)

**DELETE MEMBERSHIP**

DELETE FROM membership

WHERE membership\_name = @membershipname AND

user\_name = @username

Inputs: @username, @newusername, @password, @passwordagain, @name, @birthdate, @passportno, @creditcard

**VIEW ACCOUNT INFO**

SELECT \*

FROM customer

WHERE user\_name = @username

**MANAGE ACCOUNT**

Inputs: @username, @newusername, @password, @passwordagain, @name, @birthdate, @passportno, @membershipname

**CHANGE USERNAME**

UPDATE customer

SET user\_name = @newusername

WHERE user\_name = @username

**CHANGE PASSWORD**

UPDATE customer

SET password = @password

WHERE user\_name = @username AND  
 @password = @passwordAgain

**CHANGE NAME**

UPDATE customer

SET name = @name

WHERE user\_name = @username

**CHANGE BIRTHDATE**

UPDATE customer

SET birthdate = @birthdate

WHERE user\_name = @username

**CHANGE PASSPORT NO**

UPDATE customer

SET passport\_no = @passportno

WHERE user\_name = @username

1. **SEARCH FLIGHT SCREEN**

**LIST AVAILABLE AIRPORTS**

SELECT airport\_name, city\_name

FROM airport

**VIEW SCHEDULED FLIGHTS –TWO-WAY = RESULT OF SEARCH**

Inputs: @to, @from, @departuredate, @arrivaldate, @oneWay

( SELECT flight\_id, date, departure\_time, price

FROM CustomerFlightView C

WHERE departs = @to AND

arrives = @from AND

depatureDate = @departuredate ) UNION

( SELECT flight\_id, date, departure\_time, price

FROM CustomerFlightView C

WHERE departs = @from AND

arrives = @to AND

depatureDate = @arrivaldate )

**VIEW SCHEDULED FLIGHTS –ONE-WAY = RESULT OF SEARCH**

SELECT flight\_id, date, departure\_time, price

FROM CustomerFlightView C

WHERE departs = @to AND

arrives = @from AND

depatureDate = @departuredate

1. **MAKE RESERVATION SCREEN**

MAKE RESERVATION USE CASE

Inputs: @to, @from, @departuredate, @departuretime, @class, @username

INSERT INTO reservation

FROM flight NATURAL JOIN route

SELECT (user\_name, plane\_name, route\_id, date, departure\_time, INDEX(reservation\_no), @class, 0 )

WHERE user\_name = @username AND

route\_id in (SELECT route\_id FROM route where departs = @from and arrives = @to) AND

date = @departuredate AND

time = @departuretime AND

(plane\_name, route\_id, date, departure\_time) not in

(SELECT plane\_name, route\_id, date, departure\_time

FROM reservation NATURAL JOIN flight NATURAL JOIN plane NATURAL JOIN plane\_type

GROUP BY plane\_name, route\_id, date, departure\_time

HAVING count(user\_name) >= capacity )

1. **PURCHASE TICKET SCREEN**

Inputs: @username, @to, @from, @departuredate, @departuretime, @selectedPromotion, @extraluggage, @meal, @class, @seat\_no

INSERT INTO ticket

FROM flight NATURAL JOIN route

SELECT (user\_name, plane\_name, route\_id, date, departure\_time, INDEX(ticket\_no), @extraluggage, @meal, @class, 0, @seat\_no, price)

WHERE user\_name = @username AND

route\_id in (SELECT route\_id FROM route where departs = @from and arrives = @to) AND

date = @departuredate AND

time = @departuretime AND

price <= (SELECT total\_money FROM customer WHERE user\_name = @username) )

AFTER: Trigger Price&Total Money

USE PROMOTION USE CASE

UPDATE ticket

SET price = CASE

WHEN @selectedPromotion in (SELECT promotion\_id FROM sale WHERE user\_name = @username) THEN price \* sale\_amount

WHEN @selectedPromotion in (SELECT promotion\_id FROM campaign WHERE user\_name = @username) THEN 0

ELSE price

END

1. **RESERVATIONS & TICKETS SCREEN**

VIEW RESERVATIONS&TICKETS

SELECT \*

FROM CustomerReservationView

SELECT \*

FROM CustomerTicketView

CANCEL RESERVATION

Inputs: @reservationno, @username

UPDATE reservation

SET cancelled = 1

WHERE reservation\_no = @reservationno AND  
 user\_name = @username

REFUND TICKET

Inputs: @ticketno, @username

DELETE FROM ticket

WHERE ticket\_no = @ticketno AND  
 user\_name = @username

AFTER: Cancel reservation trigger

+ Refund Money Trigger

+ Pay Penalty Trigger