

Travel Reservation Service

CS 4400: Introduction to Database Systems
Course Project: Fall 2021 Semester

Version History

Version	Date	Notes
0	November 9 th , 2021	Initial Release
1	November 10 th , 2021	Added to 1a and 1b description and added their sample calls

Main Use Case

The following is a text description of how the system (i.e., the database tables, foreign keys, and related structures) that you've developed will be used. This information will clarify how data flows through the system, and how the views and stored procedures will allow the system operators to observe and modify the state of the system (i.e., database), respectively.

Below is a detailed list of the use cases for the system along with the functions and views that support those actions:

[1] Users Register for an Account

```
create procedure register_customer (  
    in i_email varchar(50),  
    in i_first_name varchar(100),  
    in i_last_name varchar(100),  
    in i_password varchar(50),  
    in i_phone_number char(12),  
    in i_cc_number varchar(19),  
    in i_cvv char(3),  
    in i_exp_date date,
```

```
in i_location varchar(50))
```

This procedure is to register a new customer if (and only if) the following conditions are met:

- The new customer's email and phone number will be unique in the system.
- The new customer's credit card number will be unique in the system.
- The new customer's phone number will be unique in the system.
- If the customer to be added already exists as an account and client, but not as a customer, we will add them as a customer to indicate that they are now both an owner and a customer.

```
create procedure register_owner (  
    in i_email varchar(50),  
    in i_first_name varchar(100),  
    in i_last_name varchar(100),  
    in i_password varchar(50),  
    in i_phone_number char(12))
```

This procedure is to register a new owner if (and only if) the following conditions are met:

- The new owner's email and phone number will be unique in the system.
- The new owner's phone number will be unique in the system.
- If the owner to be added already exists as an account and client, but not as an owner, we will add them as an owner to indicate that they are now both a customer and an owner.

```
create procedure remove_owner (  
    in i_owner_email varchar(50))
```

This procedure is to delete an owner if (and only if) the following conditions are met:

- The owner has no listed properties.
- If an owner is deleted, their reviews of customers should be deleted as well
- If an owner is deleted, customer reviews of the owner should be deleted as well
- Note: Only the owner should be removed from the system – if the owner is also a customer, then the customer should remain in the system. If the owner is not also a customer, then the client and account associated with this owner should be removed as well.

[2] Admins Schedule and Remove Flights

```
create procedure schedule_flight (  
    in i_flight_num char(5),  
    in i_airline_name varchar(50),  
    in i_from_airport char(3),  
    in i_to_airport char(3),  
    in i_departure_time time,  
    in i_arrival_time time,  
    in i_flight_date date,  
    in i_cost decimal(6, 2),  
    in i_capacity int,  
    in i_current_date date)
```

This procedure is used for an airline adding a new flight if (and only if) the following conditions are met:

- The new flight numbers must be combined with the airline's name to be uniquely identifiable
- The flight cannot have the same to_airport and from_airport
- The flight date must be in the future (use current date for comparison)

```
create procedure remove_flight (  
    in i_flight_num char(5),  
    in i_airline_name varchar(50),  
    in i_current_date date)
```

This procedure is used for an airline to cancel an existing flight if (and only if) the following conditions are met:

- The flight must be scheduled to depart at a date in the future compared to the current_date passed in

- When a flight is cancelled, all bookings associated with the flight should also be deleted to reflect its cancellation

[3] Customers Book and Cancel Flights

```
create procedure book_flight (  
    in i_customer_email varchar(50) ,  
    in i_flight_num char(5) ,  
    in i_airline_name varchar(50) ,  
    in i_num_seats int ,  
    in i_current_date date)
```

This procedure allows customers to book an existing flight hosted by an airline if (and only if) the following conditions are met:

- The number of seats left on the flight is greater than or equal to the number of seats to be booked.
- The date of the flight must be in the future (use current date for comparison)
- The combination of customer_email, flight_num, and airline_name should be unique in the system.
- A customer can't have more than one non-cancelled flight booked per day
- Note: if the combination of customer_email, flight_num, and airline_name already exists as a booking, the number of seats for that booking should be updated to add the additional seats if there are enough available seats remaining. If the already existing booking was cancelled, the customer will not be allowed to book this flight.

```
create procedure cancel_flight_booking (  
    in i_customer_email varchar(50) ,  
    in i_flight_num char(5) ,  
    in i_airline_name varchar(50) ,  
    in i_current_date date)
```

This procedure allows customers to cancel a booking with an existing flight hosted by an airline if (and only if) the following conditions are met:

- The customer must currently be booked for this flight
- The flight date must be at a future date (use the passed in current date for comparison)
- To cancel a booking, the was_cancelled attribute in the book table should be set to 1

create or replace view view_flight

This view displays the flight id, date, airline, destination, seat cost, number of empty seats (empty seats = capacity – total booked seats for this flight), and the total amount of money spent on the flight, for all flights. Note: the total spent on the flight should be the full cost of all current bookings plus the cancellation fees of all cancelled bookings (if nothing was spent on the flight, total spent should be 0).

[4] Owners List and Remove Properties

```
create procedure add_property (
    in i_property_name varchar(50) ,
    in i_owner_email varchar(50) ,
    in i_description varchar(500) ,
    in i_capacity int,
    in i_cost decimal(6, 2) ,
    in i_street varchar(50) ,
    in i_city varchar(50) ,
    in i_state char(2) ,
    in i_zip char(5) ,
    in i_nearest_airport_id char(3) ,
    in i_dist_to_airport int)
```

This procedure is used for an owner adding a new property if (and only if) the following conditions are met:

- The new property's address is unique
- The new property's name must be combined with the owner's email to be uniquely identifiable

- If the nearest_airport_id does not exist in the system, the property should still be created, but an entry will not be created in the is_close_to table
- If the property_name, owner_combination is not unique, no entry should be added to the is_close_to table
- Note: The nearest airport and distance to nearest airport inputs are optional. If they are given and valid, a new entry should be created in the is_close_to table. If they are not given, NULL will be passed in for both inputs and a new entry should not be created

```
create procedure remove_property (
    in i_property_name varchar(50) ,
    in i_owner_email varchar(50) ,
    in i_current_date date)
```

This procedure is used for an owner to remove a property that is currently being listed available for reservation by customers. It can only be unlisted if (and only if) the following conditions are met:

- The property is not reserved for the current date (unless the reservation is cancelled: if the current reservation was cancelled, proceed with removal)
- All past and future reservations of this property should also be deleted
- The reviews for the property must be removed when the property is unlisted
- The amenities of the property must be removed when the property is unlisted
- The property must be unlisted from being in proximity to nearby airports

[5] Customers Reserve, Cancel, and Review Properties

```
create procedure reserve_property (
    in i_property_name varchar(50) ,
    in i_owner_email varchar(50) ,
    in i_customer_email varchar(50) ,
    in i_start_date date ,
    in i_end_date date ,
    in i_num_guests int,
```

in i_current_date date)

This procedure allows customers to reserve an available property advertised by an owner if (and only if) the following conditions are met:

- The combination of property_name, owner_email, and customer_email should be unique in the system
- The start date of the reservation should be in the future (use current date for comparison)
- The guest has not already reserved a property that overlaps with the dates of this reservation
- The available capacity for the property during the span of dates must be greater than or equal to i_num_guests during the span of dates provided
- Note: for simplicity, the available capacity of a property over a span of time will be defined as the capacity of the property minus the total number of guests staying at that property during that span of time

```
create procedure cancel_property_reservation (  
    in i_property_name varchar(50) ,  
    in i_owner_email varchar(50) ,  
    in i_customer_email varchar(50) ,  
    in i_current_date date)
```

This procedure allows a customer to cancel an existing property reservation if (and only if) the following conditions are met:

- The customer must already have reserved this property
- If the reservation is already cancelled, this procedure should do nothing
- The date of the reservation must be at a date in the future (use the current date passed in for comparison)
- To cancel a reservation, the was_cancelled attribute in the reserve table should be set to 1

```
create procedure customer_review_property (  
    in i_property_name varchar(50) ,  
    in i_owner_email varchar(50) ,  
    in i_customer_email varchar(50) ,
```

```

in i_content varchar(500) ,
in i_score int,
in i_current_date date)

```

This procedure allows customers to leave a review for a property at which they stayed if (and only if) the following conditions are met:

- The customer must have started a stay at this property at a date in the past that wasn't cancelled (current date must be equal to or later than the start date of the reservation at this property)
- The combination of property_name, owner_email, and customer_email should be distinct in the review table (a customer should not be able to review a property more than once)

create or replace view view_properties

This view displays the name, average rating score, description, concatenated address, capacity, and cost per night of all properties. Note: The concatenated address should have a comma and space (', ') between each part of the address (ie: "Blackhawks St, Chicago, IL, 60176").

```

create procedure view_individual_property_reservations (
in i_property_name varchar(50) ,
in i_owner_email varchar(50))

```

This procedure creates a table that displays a single property's reservations such as the name, start date, end date, customer email, customer phone number, the total cost of the booking, the property rating score from the customer if it exists (null if it doesn't exist), and the property review from the customer if it exists (null if it doesn't exist), if (and only if) the following conditions are met:

- The property name and owner email must exist in the system. If they do not exist in the system, a table should be created with no entries.
- For calculating total cost, include the start and end date in the number of days. The cost of a single reservation should be number of days * property cost, and if the reservation is cancelled only take 20% of this calculation.
- Note: within this procedure we will create a **table** titled: "view_individual_property_reservations" that will display the data mentioned above.

[6] Customers and Owners Rate Each Other

```
create procedure customer_rates_owner (  
    in i_customer_email varchar(50) ,  
    in i_owner_email varchar(50) ,  
    in i_score int ,  
    in i_current_date date)
```

This procedure allows a customer to give an owner a rating score if (and only if) the following conditions are met:

- The customer has stayed at a property owned by the owner at some point in the past (use current date for comparison) and the reservation was not cancelled
- The customer and owner must both exist in the database
- The combination of customer_email and owner_email should be distinct in the customers_rate_owners table (a customer should not be able to rate an owner more than once)

```
create procedure owner_rates_customer (  
    in i_owner_email varchar(50) ,  
    in i_customer_email varchar(50) ,  
    in i_score int ,  
    in i_current_date date)
```

This procedure allows an owner to give a customer a rating if (and only if) the following conditions are met:

- The customer has stayed at a property owned by the owner at some point in the past (use current date for comparison) and the reservation was not cancelled
- The customer and owner must both exist in the database
- The combination of owner_email and customer_email should be distinct in the owners_rate_customers table (an owner should not be able to rate a customer more than once)

[7] Airports Host Airlines and Service Flights

create or replace view view_airports

This view displays the airport id, airport name, time zone, total number of flights that have arrived at this airport (0 if no arriving flights), total number of flights that have left this airport (0 if no departing flights), and the average flight cost of all flights leaving this airport (null if no departing flights), for all airports.

create or replace view view_airlines

This view displays the airline name, airline rating, total number of flights for this airline (if an airline has no flights, total flights should be 0), and the minimum cost for a flight through this airline (if an airline has no flights, min cost should be null), for all airlines.

[8] Admin Views Customers and Owners

create or replace view view_customers

This view displays the customer's concatenated name, average rating (leave null if the customer has no ratings), current location, if the customer is also an owner (1 if true, 0 if false), and the total number of seats the customer has purchase on flights (past and future, including cancelled flights. If the customer has purchased no seats this should be 0), for each customer.

create or replace view view_owners

This view displays the owner's concatenated name, average rating (leave null if the owner has no ratings), number of properties owned (0 if no owned properties), and average property rating (if a property isn't rated, don't include it in the calculation and if no properties are rated leave null), for all owners.

[9] Admins Process Process Dates

```
create procedure process_date (  
    in i_current_date date)
```

This procedure updates the database based on the current date by updating all customers' locations who are taking a flight on that date to the state of their destination airport if (and only if) the following conditions are met:

- If a user cancels their flight for that date, their location should not be updated
- Note: a customer should not have more than one non-cancelled flight in a single day

Main Use Case – Sample Procedure Calls

The below sample calls are positive test cases, meaning that the desired outcome for the procedure should occur and the database state should be updated accordingly without error. Note that these are only example calls and aren't sufficient to check that the procedures are fully functioning so additional testing should be conducted.

```
1a: call register_customer('falcon@gmail.com', 'Samuel',  
    'Wilson', 'password22', '777-469-5347', '9121 2762 7467 5215',  
    '809', '2022-05-11', 'Baton Rouge');
```

```
1b: call register_owner('worldchampion@gmail.com', 'Magnus',  
    'Carlsen', 'password25', '404-720-5367');
```

```
1c: call remove_owner('jseinfeld@gmail.com');
```

```
2a: call schedule_flight('3', 'Southwest Airlines', 'MIA',  
    'DFW', '130000', '160000', '2021-10-18', 350, 125, '2021-11-  
    04');
```

```
2b: call remove_flight('2', 'Southwest Airlines', '2021-08-  
    01');
```

```
3a: call book_flight('scooper3@gmail.com', '2', 'Southwest  
    Airlines', 122, '2021-10-01');
```

```
3b: call cancel_flight_booking('bshelton@gmail.com', '4',  
    'United Airlines', '2021-10-01');
```

```
4a: call add_property('Dunder Mifflin', 'mscott22@gmail.com',  
    'A great paper company for an overnight stay!', 15, 50.00,  
    'Slough Avenue', 'Scranton', 'PA', 18503, 'LGA', 135);
```

```

4b: call remove_property('LA Lakers Property',
'lebron6@gmail.com', '2021-10-22');

5a: call reserve_property('Beautiful San Jose Mansion',
'arthurread@gmail.com', 'johnthomas@gmail.com', '2021-10-19',
'2021-10-22', 1, '2021-10-01');

5b: call cancel_property_reservation('Beautiful Beach
Property', 'msmith5@gmail.com', 'cbing10@gmail.com', '2021-10-
01');

5c: call customer_review_property('Beautiful Beach Property',
'msmith5@gmail.com', 'cbing10@gmail.com', 'This property was
amazing!', 5, '2021-12-19');

5e: call view_individual_property_reservations('New York City
Property', 'cbing10@gmail.com');

6a: call customer_rates_owner('cbing10@gmail.com',
'msmith5@gmail.com', 3, '2021-10-18');

6b: call owner_rates_customer('msmith5@gmail.com',
'cbing10@gmail.com', 4, '2021-10-18');

9a: call process_date('2021-10-19');

```

Main Use Case – Global Views Expected Results

3c) view_flight

flight_id	flight_date	airline	destination	seat_cost	num_empty_seats	total_spent
▶ 1	2021-10-18	Delta Airlines	JFK	400.00	148	800.000
10	2021-10-20	Delta Airlines	ATL	700.00	110	0.000
11	2021-10-20	Southwest Airlines	ORD	600.00	95	0.000
12	2021-10-20	United Airlines	ATL	275.00	115	0.000
2	2021-10-18	Southwest Airlines	MIA	350.00	123	1050.000
3	2021-10-18	American Airlines	DFW	350.00	125	0.000
4	2021-10-18	United Airlines	LGA	400.00	96	1600.000
5	2021-10-19	JetBlue Airways	ATL	400.00	127	1520.000
6	2021-10-19	Spirit Airlines	ATL	650.00	140	0.000
7	2021-10-19	WestJet	SJC	700.00	96	3360.000
8	2021-10-19	Interjet	ORD	350.00	123	700.000
9	2021-10-20	Delta Airlines	ATL	375.00	150	0.000

5d) view_properties

property_name	average_rating_score	description	address	capacity	cost_per_night
▶ Atlanta Great Property	NULL	This is right in the middle of Atlanta near many attractions!	2nd St, ATL, GA, 30008	4	600.00
Beautiful Beach Property	NULL	You can walk out of the house and be on the beach!	456 Beach Ave, Miami, FL, 33101	2	975.00
Beautiful San Jose Mansion	3.0000	Huge house that can sleep 12 people. Totally worth it!	Golden Bridge Pkwt, San Jose, CA, 90001	12	900.00
Chicago Blackhawks House	5.0000	This is a great property!	Blackhawks St, Chicago, IL, 60176	3	775.00
Chicago Romantic Getaway	NULL	This is a great property!	23rd Main St, Chicago, IL, 60176	2	1050.00
Family Beach House	NULL	You can literally walk onto the beach and see it from the patio!	1132 Beach Ave, Miami, FL, 33101	6	850.00
House near Georgia Tech	5.0000	Super close to bobby dodde stadium!	North Ave, ATL, GA, 30008	3	275.00
LA Kings House	NULL	This house is super close to the LA kinds stadium!	Kings St, La, CA, 90011	4	750.00
LA Lakers Property	2.0000	This house is right near the LA lakers stadium. You might ev...	Lebron Ave, LA, CA, 90011	4	850.00
Los Angeles Property	4.0000		10th St, LA, CA, 90008	3	700.00
New York City Property	4.5000	A view of the whole city. Great property!	123 Main St, NYC, NY, 10008	2	750.00
Statue of Liberty Property	4.0000	You can see the statue of liberty from the porch	1st St, NYC, NY, 10009	5	1000.00
Texas Longhorns House	NULL	You can walk to the longhorns stadium from here!	1125 Longhorns Way, Dallas, TX, 75001	10	600.00
Texas Roadhouse	NULL	This property is right in the center of Dallas, Texas!	17th Street, Dallas, TX, 75043	3	450.00

7a) view_airports

airport_id	airport_name	time_zone	total_arriving_flights	total_departing_flights	avg_departing_flight_cost
▶ ATL	Atlanta Hartsfield Jackson Airport	EST	5	2	400.000000
DFW	Dallas International Airport	CST	1	0	NULL
JFK	John F Kennedy International Airport	EST	1	1	375.000000
LAX	Lost Angeles International Airport	PST	0	2	650.000000
LGA	Laguardia Airport	EST	1	2	550.000000
MIA	Miami International Airport	EST	1	3	325.000000
ORD	O'Hare International Airport	CST	2	1	350.000000
SJC	Norman Y. Mineta San Jose International Airport	PST	1	1	650.000000

7b) view_airlines

airline_name	rating	total_flights	min_flight_cost
▶ American Airlines	4.6	1	350.00
Delta Airlines	4.7	3	375.00
Interjet	3.7	1	350.00
JetBlue Airways	3.6	1	400.00
Southwest Airlines	4.4	2	350.00
Spirit Airlines	3.3	1	650.00
United Airlines	4.2	2	275.00
WestJet	3.9	1	700.00

8a) view_customers

customer_name	avg_rating	location	is_owner	total_seats_purchased
▶ Addison Ray	5.0000		0	2
Bob Lee	NULL		0	0
Blake Shelton	3.0000		0	8
Chandler Bing	NULL		1	2
Charlie Demilio	NULL		0	0
Howard Wolowitz	NULL		1	5
John Thomas	NULL		0	0
Jerry Seinfeld	1.0000		0	4
Luke Bryan	4.0000		0	2
Madison Smith	2.0000		0	2
Monica Geller	NULL		1	0
Sheldon Cooper	NULL		1	0
Samantha Wilson	5.0000		0	3
Taylor Swift	4.0000		0	2

8b) view_owners

	owner_name	avg_rating	num_properties_owned	avg_property_rating
►	Arthur Read	4.0000	3	3.5000
	Chandler Bing	5.0000	1	4.5000
	Ellie Johnson	NULL	1	NULL
	George Burdell	5.0000	1	5.0000
	Howard Wolowitz	2.0000	1	5.0000
	John Wayne	NULL	0	NULL
	Lebron James	1.0000	1	2.0000
	Monica Geller	3.0000	1	4.0000
	Michael Jordan	NULL	1	NULL
	Michael Scott	NULL	2	NULL
	Michael Smith	NULL	1	NULL
	Sheldon Cooper	NULL	1	NULL