

Oregon State University

Introduction to Databases

CS340-400 Fall 2019

Group 25

Tourism Database

Step 7: Project Portfolio

Aishwarya Manicka Ravichandran
Amy Robertson

Table of Contents

Website	2
Summary	2
Project Outline	3
Entities	3
Relationships	5
Entity Relationship Diagram	6
Schema	7
Screen Captures	8
Home	8
Customers	9
Bookings	12
Assignments	14
Tour Guides	17
Ratings	23
Payment	25
Feedback & Changes from Previous Steps	27
Project Step 1	27
Project Step 2	27
Project Step 3 Draft Version: Design HTML Interface	27
Project Step 3 Review	28
Project Step 4 Draft Version: DML and DDL Queries - Peer Reviews	29
Project Step 5 Draft Version: Implement CREATE + READ Operations	31
Project Step 6 Draft Version: Implement UPDATE + DELETE Operations	32

Website

<http://flip3.engr.oregonstate.edu:9810/>

Summary

Overall, the basic structure of our project remained essentially unchanged from the start of the project to now. Majority of the feedback we received was centered around readability and usability, but not as much regarding the structure of the database. However, one change that we did implement based on feedback was the addition of the Ratings table. A fellow student suggested that it would be interesting for the customer to be able to add ratings for the trips. We agreed and added that as an entity.

The other structural change that we made was adding the price per adult and price per child attributes to the Travel Location table. We realized that in order to generate tour price amounts for the bookings, and consequently, the payments table, that we needed a price per adult and per child. We implemented this as a simple calculation but realized that should be an attribute for each location and the calculation should be done by a SQL query.

We had also originally planned to have a user and admin portal with different views but decided against pursuing this front-end design as it appeared beyond the scope of the project and could potentially take too much time to implement. We also changed our HTML design some to improve the readability of our site, removing the image background and instead adding color background to the tables so that it was more clearly distinguished from the background image. We clarified the project by adding the descriptions on the main page based on feedback. We also implemented more dropdowns in several instances where we originally had an ID as the search parameters, such as using customer email instead of customer ID. As mentioned in feedback, ID numbers are not typically identifiable by the user, so we displayed more meaningful indicators.

At the end of implementation, we also realized that we did not have a relationship with partial participation. We decided that it made the most sense for the assignment relationships to allow partial participation with the tour guide entity. An assignment must have an associated booking, but the tour guide can be updated to NULL (no guide) by the user.

The most value from the feedback came during steps 5 and 6. We did have a couple implementation blocks that we posted and received very constructive and helpful feedback that helps us solve the issues. One that was resolved with peer feedback was the implementation of dropdowns including adding a "NULL" or blank at the top of the dropdown, formatting issues and displaying the current date.

Overall, the process of submitting each step for peer review was very positive. All feedback was clearly intended to be positive and was overall very constructive. Perhaps the feedback did not cause us to make significant changes to the structure of our database but it did help with the process of refining and clarifying the data and its presentation.

Project Outline

The outline of the project is to design a database system for a generic tourism company. The project will focus on the purchase organization (guides, customer, etc) as related to each destination purchase. We would imagine this as a mini version of Expedia website, where a customer can make bookings for different travel locations and make payment based on the bookings made. The customer will also be assigned tour guides for the locations, he has booked to travel. The customer can also give ratings to the locations, he visited and also select the travel locations based on the ratings given by other customers.

Entities

(We assume that all attributes as NOT NULL except middle_name in Customer entity and tourGuide_ID in Assignment Entity. number_children in Booking entity is given as DEFAULT NULL)

Customer (Entity 1):

Attributes	Data Type
Customer_ID	INT(11) [Primary Key, Auto Increment]
First_Name	Varchar(255)
Last_Name	Varchar(255)
Middle_Name(optional)	Varchar(255)
Street_Number	INT(11)
City	Varchar(255)
State	Varchar(255)
Country	Varchar(255)
Postal_Code	Varchar(5)
Phone_Number	Varchar(12)
Emai_ID	Varchar(255)
Passport_Number	Varchar(10)
Passport_CountryOfIssue	Varchar(255)
Passport_ExpiryDate	Date

Bookings (Entity 2):

Attributes	Data Type
Booking_ID	INT(11) [Primary Key, Auto Increment]
Booking_Date	Date
Departure_Date	Date
Arrival_Date	Date
Number_Adults	INT(11)
Number_Children	INT(11)
TravelLocation_ID	INT(11) [Foreign Key]
Customer_ID	INT(11) [Foreign Key]

Travel Location (Entity 3):

Attributes	Data Type
TravelLocation_ID	INT(11) [Primary Key, Auto Increment]

City	Varchar(255)
Country	Varchar(255)
Amount_perAdult	Decimal(8,2)
Amount_perChild	Decimal(8,2)

Tour Guide (Entity 4):

Attributes	Data Type
TourGuide_ID	INT(11) [Primary Key, Auto Increment]
First_Name	Varchar(255)
Last_Name	Varchar(255)

Payment (Entity 5):

Attributes	Data Type
Payment_ID	INT(11) [Primary Key, Auto Increment]
Booking_ID	INT(11) [Foreign Key]
Payment_Amount	Decimal(9,2)
Payment_Date	Date
Payment_Description	Varchar(255)

Assignment (Relationship Entity):

Attributes	Data Type
TourGuide_TravelLocationID	INT(11) [Primary Key, Auto Increment]
TravelLocation_ID	INT(11) [Foreign Key]
TourGuide_ID	INT(11) [Foreign Key]
Booking_ID	INT(11) [Foreign Key]

Ratings (Relationship Entity):

Attributes	Data Type
Rating_ID	INT(11) [Primary Key, Auto Increment]
TravelLocation_ID	INT(11) [Foreign Key]
Customer_ID	INT(11) [Foreign Key]
Rating	INT(11)
Review	varchar(255)

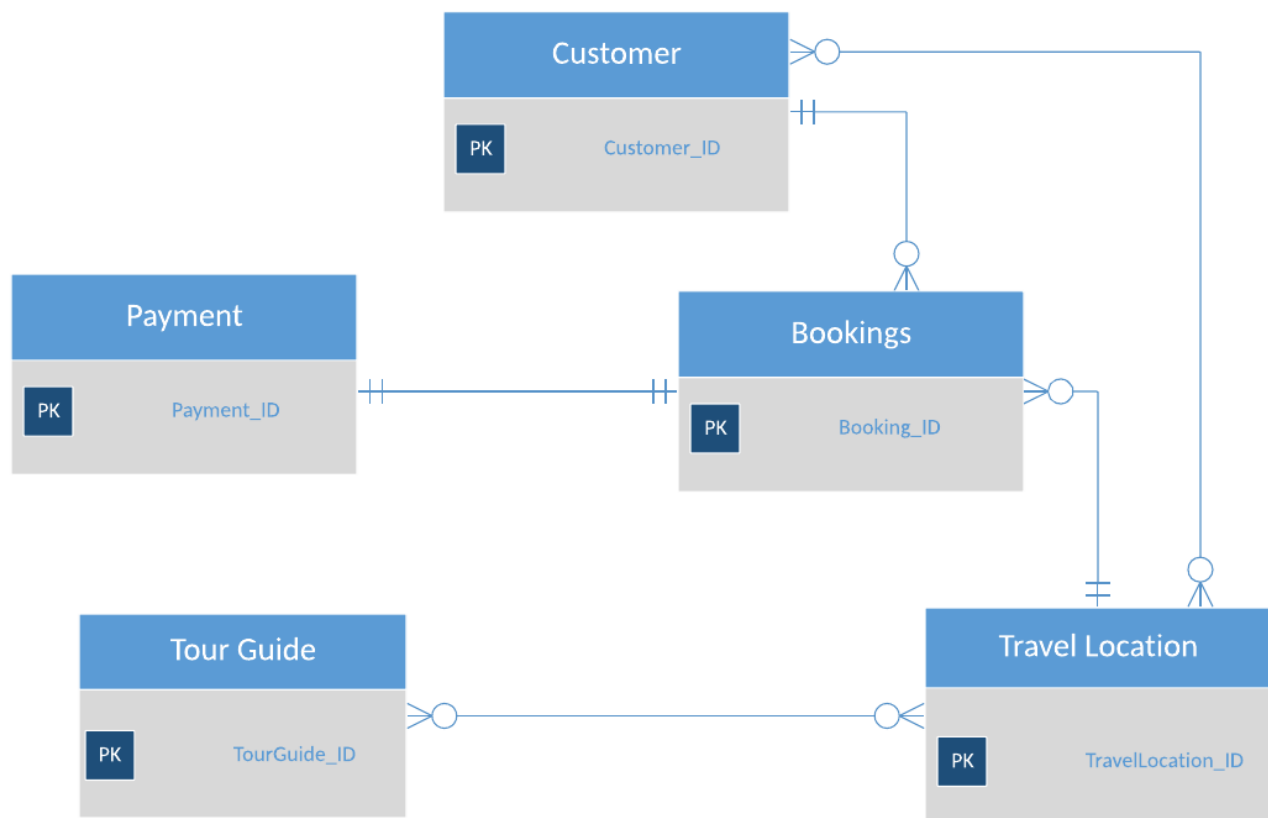
Unique Constraints:

- **In Tour_Guide Entity**
'Full_name' UNIQUE('first_name', 'last_name')
- **In Travel_Location Entity**
'location' UNIQUE ('city', 'country')
- **In Assignment Entity**
'uniquebooking' UNIQUE ('booking_ID');

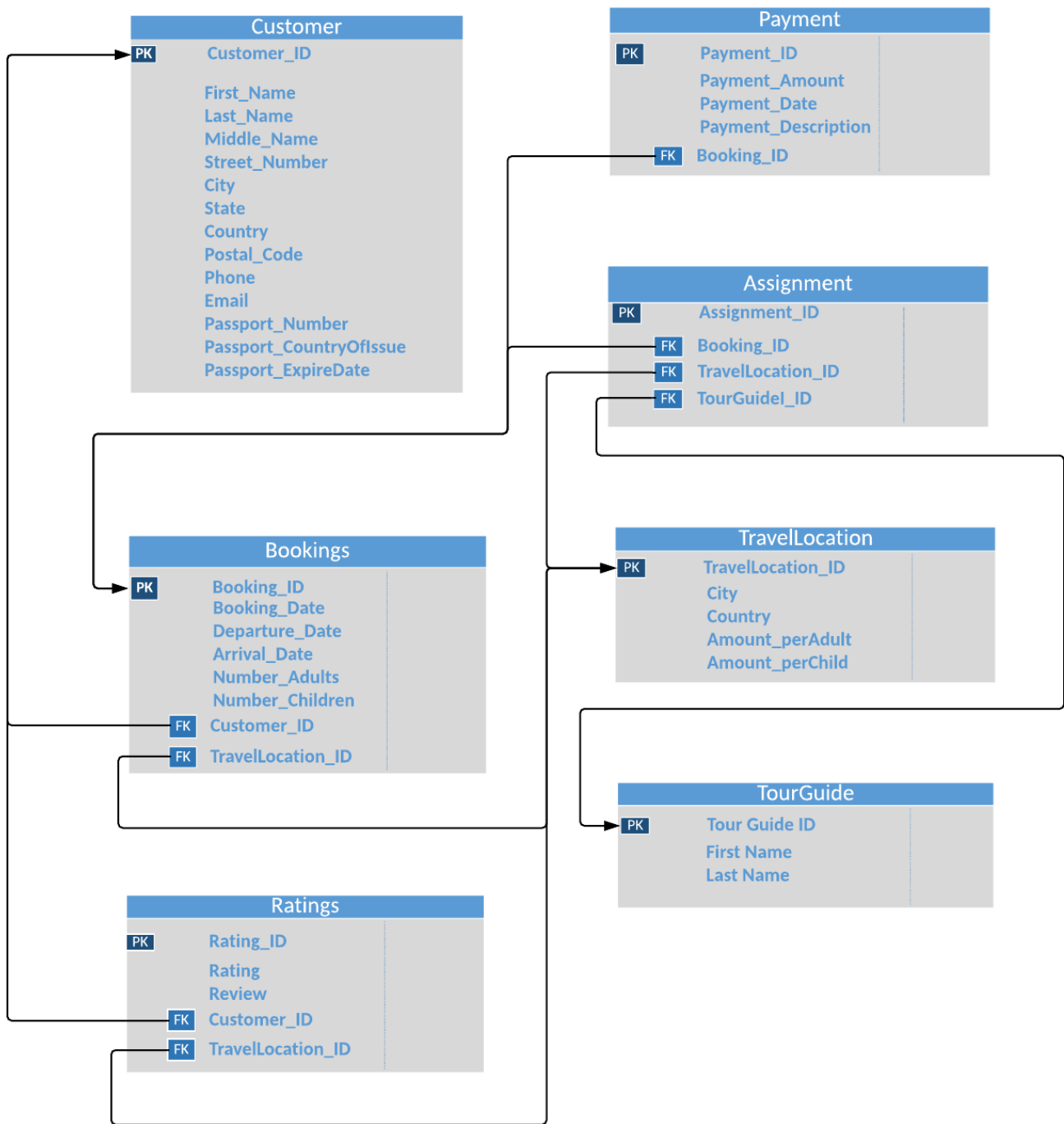
Relationships

- Customer and Bookings:
 - Customer to Bookings:
 - ❖ **Zero to Many** relationship
 - ❖ A customer can have zero or many bookings done
 - Bookings to Customer:
 - ❖ **One to One** Relationship
 - ❖ A booking can only be associated to one customer
 - ❖ A booking must have a customer
- Booking and Travel Location:
 - Booking to Travel Location:
 - ❖ **One to One** relationship
 - ❖ A booking can only be associated to one travel location
 - ❖ A Travel location can be associated with zero to many bookings
 - Travel Location to Bookings:
 - ❖ **Zero to Many** relationship
 - ❖ A travel location can have many bookings, i.e, a trip can have many bookings
- Bookings and Payment:
 - Bookings to Payment:
 - ❖ **One to One** relationship
 - ❖ A booking can be associated to only one payment
 - ❖ A payment must have at least one booking.
 - Payment to Bookings:
 - ❖ **One to One** relationship
 - ❖ A payment can be done for only one booking
- Tour Guide and Travel Location
 - **Many to Many** Relationship
 - A tour guide can be associated zero to many travel locations
 - A travel location can have zero to many tour guides.
 - Assignment Table will hold M-M relationship between tour guide and travel location
- Customer and Travel Location:
 - **Many to Many** Relationship
 - A customer would have travelled to zero or many travel locations. Likewise, a travel location would be visited by zero or many customers.
 - The Ratings table will have attributes for the rating and reviews posted by different customers, which will help the customers choose the travel location based on the rating.
 - ❖ This table will resolve the M-M relationship between Customer and Travel Location with respect to reviews.
 - ❖ Travel Location can have zero or more ratings
 - ❖ A customer can give ratings for zero or more locations.

Entity Relationship Diagram



Schema



Screen Captures

Home

Main page with description of functionality and purpose for each other page.

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Welcome to our website!

Customers

Search & edit existing customers, plus add new customers. Customers cannot be deleted from the database.

Bookings

Search & delete existing bookings, plus add new bookings. Bookings must be associated with an existing customer. Bookings can be deleted and upon deletion, the associated payment and assignment will also be deleted. When a new booking is created, a new payment must be added. A new assignment will also be added, but the Tour Guide in the relationship can be NULL. This can then be updated/managed from the Tour Guide page.

Assignments

Search, edit, delete & add new assignments. New assignments require only the Booking Id and Tour Guide name. The location Id is take from the booking information as the 3rd foreign key. Assignments manage the M:M relationship between Location and Tour Guide. The guide for a given assignment can be set to NULL (no guide), but an assignments is always linked to a booking.

Tour Guides

Search, edit & add new tour guides to the database. If a Tour Guide is deleted from the database, any associated assignments are also deleted.

Locations

Search, edit & add new locations to the database. Locations cannot be deleted from the database.

Ratings

Search & add new ratings to the database. Ratings must be associated with a customer and with a location, managing the M:M relationship between the two.

Payments

Search, delete & add new payments. Payments must be associated with a booking.

Customers

Main Customer Page – Displays all customers in database. From this page, user can filter and add new customers

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Search Customers

Email ID

Select email ID

Filter

Clear

Add New Customer

#	First Name	Middle Name	Last Name	Address	City	State	Country	Postal Code	Phone number	Email	Passport#	Passport CountryOfIssue	
1	Kenny		Sebastian	22	Seattle	Washington	USA	98101	111-222-3333	fakemail@random.com	O902323	USA	Edit
2	Judith		Aniston	48	Las Vegas	California	USA	98901	222-222-3333	judith@random.com	J682423	USA	Edit

Add New Customer – Input all information necessary to add new customer to database.

Tourism Database[Home](#)[Customer](#)[Bookings](#)[Assignments](#)[Tour Guides](#)[Locations](#)[Ratings](#)[Payment](#)

Add New Customer:

First Name:

Your First Name

Last Name:

Your Last Name

Middle Name:

Your Middle Name (optional)

Street Number:

Street Number

City:

Eg:Seattle

State:

Eg:Washington

Country:

Eg:USA

Postal Code:

Eg:98101

Phone Number

xxx-xxx-xxxx

Email ID

Eg:abcd@gmail.com

Passport Number:

Eg:P2309876

Passport CounrtyOfissue:

Eg:USA

Passport Expiry Date:

10/24/2019

Submit

Edit Customer – From this form, the user can update information for a specific customer. Form is prepopulated with existing customer information.

Tourism Database[Home](#) [Customer](#) [Bookings](#) [Assignments](#) [Tour Guides](#) [Locations](#) [Ratings](#) [Payment](#)

Update Customer:

First Name	<input type="text" value="Kenny"/>
Last Name	<input type="text" value="Sebastian"/>
Street	<input type="text" value="22"/>
City	<input type="text" value="Seattle"/>
State	<input type="text" value="Washington"/>
Country	<input type="text" value="USA"/>
Postal Code:	<input type="text" value="98101"/>
Phone Number	<input type="text" value="111-222-3333"/>
Email ID	<input type="text" value="fakemail@random.com"/>
<input type="button" value="Update"/>	

Bookings

Main Bookings page – Displays all bookings in database. From this page, user can filter, add new and delete bookings.

Tourism Database

HomeCustomerBookingsAssignmentsTour GuidesLocationsRatingsPayment

Manage Bookings

Customer Name

Select Customer Name

Filter

Clear

Add New Booking

Booking ID#	Booking Date	Departure Date	Arrival Date	Number of Adults	Number of children	Travel Location ID #	Customer ID #	Delete
1	12/12/2019	05/01/2020	05/10/2020	2	0	1	2	Delete
2	05/12/2019	05/02/2020	05/14/2020	4	1	2	1	Delete
11	12/08/2019	02/11/2020	02/20/2020	2	0	7	2	Delete
12	12/08/2019	12/09/2019	12/19/2019	2	0	3	2	Delete

Add New Booking – Input all information necessary to add new booking to database

Tourism Database[Home](#)[Customer](#)[Bookings](#)[Assignments](#)[Tour Guides](#)[Locations](#)[Ratings](#)[Payment](#)

Add New Booking

Customer Name	<input type="text" value="Select customer"/>
Travel Location	<input type="text" value="Paris"/>
Departure Date	<input type="text" value="mm/dd/yyyy"/>
Arrival Date	<input type="text" value="mm/dd/yyyy"/>
Number of Adults	<input type="text"/>
Number of Children	<input type="text"/>

Add

Assignments

Main Assignment Page – Displays all assignments in database. From this page, user can filter, add new and delete assignments.

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Manage Assignments

Add New Assignment

Assignment ID#	Booking ID#	Guide	Destination	Customer Name	Departure Date	Arrival Date	# Adults	# Children		
15	1	Joey Tribbiani	Paris, France	Judith Aniston	05/01/2020	05/10/2020	2	0	Delete	Edit
17	2	Ross Geller-Jones	Bangkok, Thailand	Kenny Sebastian	05/02/2020	05/14/2020	4	1	Delete	Edit
19	11	Elizabeth Taylor	Edinburgh, Scotland	Judith Aniston	02/11/2020	02/20/2020	2	0	Delete	Edit

Add New Assignment – Input all information necessary to add new assignment to database

Tourism Database[Home](#) [Customer](#) [Bookings](#) [Assignments](#) [Tour Guides](#) [Locations](#) [Ratings](#) [Payment](#)

Add New Assignment:
One assignment per booking. Only unassigned bookings will appear in dropdown.

Booking ID#

12

Tour Guide

Chandler Bing

Add

Return to Manage Assignments

Edit Assignment – From this form, the user can update information for a specific assignment. Form is prepopulated with existing

Tourism Database Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Manage Assignment #15:

Booking ID 1 ▼

Booking ID is unique for each assignment. Cannot be updated. Shown here for display purpose!

Tour Guide Joey Tribbiani ▼

Update

Tour Guides

Main Tour Guide Page – Displays all tour guides in database. From this page, the user can search by tour guide first name, and add new and delete existing tour guides.

Tourism Database

HomeCustomerBookingsAssignmentsTour GuidesLocationsRatingsPayment

Manage Tour Guides

Search By First Name

Search

Clear

Add New Tour Guide

Tour Guide ID#	First Name	Last Name	# Assignments	# Locations	Update	Delete
2	Ross	Geller-Jones	1	1	<div>Edit</div>	<div>Delete</div>
3	Joey	Tribbiani	1	1	<div>Edit</div>	<div>Delete</div>
4	Chandler	Bing	0	0	<div>Edit</div>	<div>Delete</div>
10	Elizabeth	Taylor	1	1	<div>Edit</div>	<div>Delete</div>

Add New Tour Guide – Input all information necessary to add new tour guide to database.

Tourism Database[Home](#)[Customer](#)[Bookings](#)[Assignments](#)[Tour Guides](#)[Locations](#)[Ratings](#)[Payment](#)

Add New Tour Guide:

First Name

Last Name

Add

Return to Manage Tour Guides

Edit Tour Guide – From this form, the user can update information for a specific tour guide name. Form is prepopulated with existing tour guide information.

Tourism Database Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Update Ross Geller-Jones:

First Name

Last Name

Travel Locations

Main Travel Location Page – Displays all travel locations in database. From this page, the user can search locations by country, add new locations and update a specific location.

Tourism Database									
Home Customer Bookings Assignments Tour Guides Locations Ratings Payment									
Manage Locations									
Search By Country <input type="text"/>									
<div>Search Clear Add New Location</div>									
Location ID#	City	Country	Price Per Adult	Price Per Child	# Bookings	# Tour Guides	# Adults + Kids	Average Review	
1	Paris	France	100.00	50.00	1	1	2	5	Edit
2	Bangkok	Thailand	200.00	150.00	1	1	5	4	Edit
3	Las Vegas	USA	300.00	250.00	1	0	2		Edit
4	Hawaii	USA	200.00	150.00	0	0			Edit
5	Melbourne	Australia	150.00	200.00	0	0			Edit
6	Rome	Italy	800.00	700.00	0	0			Edit
7	Edinburgh	Scotland	900.00	500.00	1	1	2		Edit
8	Venice	Italy	900.00	450.00	0	0		5	Edit

Add New Location – Input all information necessary to add new location to database

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Add New Travel Location:

City

Country

Price Per Adult

Price Per Child

Edit Location – From this form, the user can update price information for a specific location. Form is prepopulated with existing location price information.

Tourism Database[Home](#)[Customer](#)[Bookings](#)[Assignments](#)[Tour Guides](#)[Locations](#)[Ratings](#)[Payment](#)

Update Prices for Paris, France:

Price Per Adult

100

Price Per Child

50

Update

Ratings

Main Ratings Page – Displays all customers in database. From this page, the user can filter and add new customers.

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Manage Ratings

Rating

Select Rating

Filter

Clear

Add New Rating

Rating ID#	Travel Location ID#	Travel Location	Customer ID#	Customer Name	Rating	Review
1	1	Paris	1	Kenny Sebastian	5	Excellent
2	2	Bangkok	2	Judith Aniston	4	Good
10	8	Venice	1	Kenny Sebastian	5	Outstanding! Best place on Earth!

Add New Rating – Input all information necessary to add new rating to database.

Tourism Database

HomeCustomerBookingsAssignmentsTour GuidesLocationsRatingsPayment

Add Ratings

Customer ID

Select Customer Name ▾

Location

Rating

Select Rating ▾

Review

Add

Payment

Main Payments Page – Displays all payments in database. From this page, user can filter and add new payments.

Tourism Database

Home Customer Bookings Assignments Tour Guides Locations Ratings Payment

Manage Payment

Booking ID

Filter

Clear

Add Payment

Payment ID#	Booking ID#	Payment Amount	Payment Date	Payment Description	Delete
1	1	1800	12/07/2019	Payment for Paris tour	<div>Delete</div>
10	2	11400	12/08/2019	Full payment.	<div>Delete</div>

Add New Payment – Input all information necessary to add new payment to database.

Tourism Database[Home](#)[Customer](#)[Bookings](#)[Assignments](#)[Tour Guides](#)[Locations](#)[Ratings](#)[Payment](#)

Add Payment

One payment per booking. Only unpaid bookings will appear in dropdown.

Booking ID

Select BookingID ▾

Payment Description

Pay

Feedback & Changes from Previous Steps

Project Step 1

- 1) The datatype of "Postal Code" should be varchar(5) or varchar(10). The disadvantages of INT here are: you cannot store a dash and the leading zero will be missed. Also need to change the datatype of "Phone" because of the same reason.
- 2) The "Tour Departure ID" is associated to "Travel Location ID", right? If you have an attribute "Assignment" (INT), there should be an ID in "Assignment" entity. Please add some description about these two attributes "Tour Departure ID" and "Assignment".
- 3) Based on my understanding of the data structure, the "Travel Location" attribute of Bookings is a FK. If so, it would be nice if you renamed it to "Travel Location ID". It is more straightforward to present its relation with "Travel Location" entity.

Actions based on Feedback:

- 1) Postal code and phone attributes for the Customer entity changed to varchar(5) and varchar(10) respectively.
- 2) Assignment ID attribute added to Assignment entity and labeled as Primary Key.
- 3) Tour Departure changed to Travel Location.
- 4) The Travel Location attribute of Bookings was renamed to Travel Location ID and noted as a foreign key.

Project Step 2

No feedback was received between submission of Step 2 and Step 3 Draft Version.

Project Step 3 Draft Version: Design HTML Interface

- 1) Project Outline and Database Outline, ERD and Schema Updated Version

“Well Done!” feedback received for Step 2 Final Version.

- 2) Fixes based on Feedback from Previous Steps:

No applicable fixes for this step.

Feedback by the peer reviewer

Include verbatim the review that your Draft submission received in the Group Review. If you did not receive any reviews, mention that. You would include feedback that both members of the Project Group received.

- 1) Maybe I am missing it somewhere, but what is your project? I didn't see a description in the document. A tourist company of sorts? Also, regarding your ERD, you have your relationship tables, or "entities". I was under the impression that we don't put those for ERDs. We just use notation to represent the relationships, then we create relation tables as you did in your Schema. But I could be way off. Otherwise, good stuff!
- 2) Great Work! I think that your idea has been properly executed and explained. That being said I do have a few minor suggestions:
 - a. I would create tables for each of the entities to make it more readable. It is a bit annoying to match a given attribute to its corresponding data type when there are no horizontal lines to give you.
 - b. I noticed you only specified the size of a few of you varchar(#) attributes. I think it would be best if you specified the size for all of them. Is there a specific reason as to why not all of the sizes were specified?
 - c. I would add to you tables a bit information about the participation of the relationship (e.g. partial or total participation).

- d. I would name your relationship tables a combination of all the entities that are represented in them. This would make it easier to identify the specific entities and their relationships. Well done, keep it up
- 3) I think a bookings website is a great idea for this project. Your document is fairly well organized and easy to follow.
 - a. I think your table is a clear way to convey your various entities and attributes. You've clearly put a lot of time into nailing down the various details you'll need for the database as a whole. I don't see much to critique there. Your ERD is equally well organized and helps to make your relationships very clear.
 - b. I do think the schema could be a bit neater. I'm not sure you need to have the types for your attributes listed. Additionally, the overlapping of the arrow makes some relationships unclear and hard to follow.
 - c. Overall, I think this is well thought out and presented well. My biggest concern is the lack of an overview of the project. After reading through the tables and diagrams, I understand you are building some sort of trip booking application and there will be tour guides involved.
 - d. I'm not sure if this will be the website for a travel agency that gives tours of an area or if this is something more like Expedia. It could be a little bit of both, but unfortunately that's not clear.
 - e. I think the database setup seems sound, but I'd like to know a little more about the application overall.
 - 4) Hey there, guys. Interesting idea. I really like the addition of the tour guide and assignment entities. A few things:
 - a. Booking should have a one to many relationship to Customer in the outline, but it is correct in the ER diagram.
 - b. This may be more of a personal thing, but I prefer underscores to spaces in table names and attributes. I'm not sure if the way you did it would cause any problems, but it might in some DBMSs.
 - c. Perhaps a cool addition would be a rating relation. This relation would hold customer_id, location_id, number_of_stars (out of 5), and review. It would be a cool way to search for most popular locations on your website.

Actions based on the feedback

List briefly the actions that you chose to take based on the above feedback. If you decided not to act on a specific suggestion, you need to describe in detail your reasoning.

- 1) We created separate tables for each entity, to make it look clear and organized.
- 2) We added description of our project in the outline.
- 3) We added the size of the varchar attributes. Clarified Primary Keys as auto increment.
- 4) Added more detail regarding participation of relationships between tables. Changed outline to bullet form.
- 5) Removed relationship tables from ERD.
- 6) Reworked schema to improve readability. Removed data types.
- 7) Added Ratings table for rating relation, with Rating_ID as primary key and Customer_ID & TravelLocation_ID as foreign keys. Also added rating and review attributes.
- 8) Renamed attributes in all entities with underscores and camelcase.
- 9) We did not rename the relationship tables as a combination of the entities that are involved, because it didn't give much meaning to the table and we felt that it was not clear.

Upgrades to the Draft version

If you are making any changes to the files based on your own changed design decisions, they should be listed under this section.

- 1) We removed foreign keys, Booking_ID and TravelLocation_ID from the Travel Location entity, since, the purpose of the Assignment table is to connect the bookings, travel locations and tour guides. So, we added the Booking_ID in the Assignment table as a Foreign Key.
- 2) We changed the datatype of Payment_Amount in Payment Entity from Currency to Float.

Project Step 3 Review

Feedback by the peer reviewer

Include verbatim the review that your Draft submission received in the Group Review. If you did not receive any reviews, mention that. You would include feedback that both members of the Project Group received.

- 1) Gabrielle Hatfield 3 days ago Hello! Neat project idea and great looking site!

A few questions -> what functionalities will the customer have once signed in? I'm guessing once they are signed in, they will get a different view where they can search, book or cancel trips? Will their view also allow additions to the payment table and the ratings table? (I think that will complete the inserts)

For the admin view for the list of the manage tour guides, have you considered having a delete button for guides, in addition to edit? Also, will the admin be able to see payment info for customers?

Also, could an administrator manually add customers to the site? Overall, great start!

- 2) Hey Group 25! From what I can tell you've covered all of the database requirements that are listed on the project assignment page. It looks like you have the necessary SELECT INSERT UPDATE DELETE functions built into your UI. Only thought I had would be to add dropdowns for certain search attributes to make searching database items a little easier for the user. That way they wouldn't need to match the search criteria exactly how it's stored in the database. For examples, on you Search Assignments page, you could have a dropdown of all available Travel Locations instead of a text input that the user would need to type in exactly hows its stored in your database. I don't think this is necessary for all search inputs, but for some attributes having dropdowns with available options may make your user experience a little better. Nice work so far!
- 3) Haochong Pan 16 hours ago Hello guys, The main page looks so pretty! Here are some thoughts and suggestions:
 - a) I think it will be better if the 'Booking' table can also be viewed from Admin site since it's a two side website for Admins and Customers.
 - b) I think you might forgot to put customer_id attributes in your 'Booking' entity to refer to the customer information.
 - c) I am not sure if I missed some details in your pdf, but when I was looking through your file, I didn't see the 'Assignment' on the ERD diagram. So you might just need to take a little bit time to add it up.
Good job guys! :)

Actions based on the feedback

List briefly the actions that you chose to take based on the above feedback. If you decided not to act on a specific suggestion, you need to describe in detail your reasoning.

- 1) Added Ratings page with with SEARCH, ADD buttons
- 2) Added Bookings page with with ADD, SEARCH & DELETE buttons/corresponding pages
- 3) Added Payment page with ADD, SEARCH, EDIT & DELETE buttons/corresponding pages
- 4) Updated select fields to be dropdowns based on current database content instead of user input.
- 5) Add text describing function of each page.
- 6) Decided to not add DELETE to guide and customer as do not want user to be able to delete guides or customers entirely from the database.
- 7) Added ability for administrator to manually ADD customers to database

Upgrades to the Draft version

If you are making any changes to the files based on your own changed design decisions, they should be listed under this section.

- 3) Remove the customer page and just have home and admin page
- 4) Updated fields on Customer, Bookings, Tour Guides & Assignment pages.
- 5) Added Search button to each page.
- 6) Created html pages to update Customers, Tour Guides and Bookings.

Project Step 4 Draft Version: DML and DDL Queries - Peer Reviews

Feedback by the peer reviewer

- 1) Elliot Hershberg 5 days ago
Hi Amy and Aishwarya,
The website looks great! Your design is very easy to navigate, and captures a lot of useful functionality. I noticed that your group chose to host on Github. How has that been going? Is it pretty straightforward. Our group chose to host on

Heroku and it has been good and easy so far, but I'd be curious to try Github pages for personal projects down the road.

Both SQL files look good. My critique for the code is more aesthetic than functional. There are several locations in the code that I think could be split into several lines. Again, the code will run the same, but I think that it makes it much easier to read and quickly see what is going on.

For example,

```
SELECT tour_guideID, first_name, last_name FROM tour_guide WHERE tour_guideID = :tour_guideID_from_browse_form
```

I find harder to parse than

```
SELECT tour_guideID, first_name, last_name  
FROM tour_guide  
WHERE tour_guideID = :tour_guideID_from_browse_form
```

With some additional lines, the logic really jumps out. Again, great work so far!

2) John Sy 5 days ago

First of all I can definitely tell your group has spent a lot of time on the SQL statements, even going above what's required by conveniently grouping the results by customer even though you didn't have to. I know since it's a rough draft and there isn't a whole slew of example data to try out the search bar I can already see how it would behave. The only thing I would say since I didn't see it specifically, is integrating wildcard functionality on that since as of now I can only see the user input being assigned as is from the form straight to :email_idSearchInput, just in case your user input is not an exact match to something in the table.

The dropdown boxes for subsequent input using data existing in other tables is well thought out in pages such as Assignments where you narrow down the user's choice in Booking ID/Location/Tour Guide. Normally I'd suggest having the actual names as opposed to the ID/Primary Key as the choices but since this is a travel/booking management system, I'd assume that anything comparable used in the real world would likely use id numbers as well, so it makes sense in this context.

Other than that, just minor aesthetic adjustments regarding the lighter color choices against the background in your html table elements make it a bit hard to see some of the information but at the end of the day that's nothing major.

3) Justin Parks 4 days ago

Hi! Right off the bat, your SQL queries are really extensive! It looks like you all put a lot of time into making these as user-friendly as possible. I have a little trouble reading the lines however. Perhaps separating the SELECT from the FROM lines, etc. could make it a bit more readable. It's not a huge deal, since most users don't ever see these raw queries anyways.

Looking at your site, it's very easy to see where things are and where a user can interact with the data. One thing I would suggest is to have a background element behind your text that provides a bit of contrast between the text and the background. Making it just transparent enough could make the text easier to read without losing the look of the original background.

4) Megan Liles 3 days ago

Your website is great, some things I would note is that wherever you have text with an image in the background, make the background of the text a solid color so it's easier to read.

As far as your SQL queries go, it looks like you have everything covered pretty good. Some things I would suggest would be to make your user input as "error free" as possible so for example when users enter a date, make it a date selector rather than just text because it could be entered wrong and mess up your database. As well as changing this of the admin side of things because you could accidentally delete something you didn't mean to.

Actions based on the feedback:

- 1) Improved readability of SQL statements to be implemented in javascript files.
- 2) Will look at background / text visibility but no changes being made at this point.

Project Step 5 Draft Version: Implement CREATE + READ Operations

Feedback by the peer reviewer

- 1) Allan Slocum
Hey Amy and Aishwarya,
Your link to your website appears to be directing to a previously hosted version. It says :8806, but directs to :8805.

I also noticed that you only have a main2.js file. Although it's not impossible to write everything into one .js file, it's probably best to split it up and use terms like these from the sample database to link them:

```
app.use('/people', require('./people.js'));  
app.use('/planets', require('./planets.js'));
```

There doesn't seem to be any html in your .zip file so I can't comment on it too much other than inspecting your website for what's rendered.

For you "blocked" portions, let's start with drop-down selection. These are important to ensure that an entity added to your database that requires a relationship will have the relationship established at its inception with an existing entity. Another thing to mention is that it's generally bad form (no pun intended) to require your user to input ID's into the forms. You should be able to handle these with drop-down menus that display the name of the selection but store the value via a SQL query. The form should look something like this, where homeworld is the fk relationship entity with "person" (this is from the sample bsg_db).

```
<form id="addperson" action="/people" method="post">  
First name: <input type="text" name="fname"><br>  
Last name: <input type="text" name="lname"><br>  
Homeworld: <select name="homeworld">  
  {{#each planets}}  
    <option value="{{id}}">{{name}}</option>  
  {{/each}}  
</select><br>  
Age: <input type="number" name="age"><br>  
<input type="submit" value="Submit">  
</form>
```

In order for this to display properly in your page, you need to add a function to your router.get (for your add-page render) for getting all your drop-down entities to display in the drop-menu (and don't forget to increment your callback count). IMPORTANT TO NOTE: the value for the homeworld selection is the id, not the name, for the homeworld. This means that you don't need to add any fancy sql selection query when adding that data using a app.post in main.js, you simply just need to add the value.

Let's take a look at part of your add-booking as an example:


```
var sql = "INSERT INTO bookings(customer_ID, travelLocation_ID, departure_date, arrival_date, number_adults, number_children) VALUES(?, (SELECT travelLocation_ID FROM travel_location WHERE city = ?),?, ?, ?, ?);";
```

```
var inserts = [req.body.customer_ID, req.body.travelLocation, req.body.departure_date, req.body.arrival_date, req.body.number_adults, req.body.number_children];
```

```
sql = mysql.pool.query(sql,inserts,function(err, result, fields){
```

If implemented as above for the form, the value for travelLocation should be the id, therefore you don't need a SELECT query to obtain the id, you just need to add the value passed in by the form for that selection.

One more thing -- if you want to have an option in your drop-down menu for selecting NULL, here's how to do it:

In your form, similar to above, you need to add a selection that indicates a lack of a relationship, let's say you want the choice to say "There is no booking for this location". The value you assign to that selection can be whatever you want, so long as you handle it in the app.post accordingly to change it's value to actual null before it's sent off the to the database.

Here's an example using your main.js function for app.post('/add-rating').

Let's assume that your select-location value came from a drop-down selection menu and you had an option to that meant no location was chosen. If you set the value for that selection to "NO_LOCATION" in your html form, then in your app.post, in-between your sql and your inserts for the post you would write a way to convert "NO_LOCATION" to null like so:

```
if (req.body.select-location == "NO_LOCATION"){  
  req.body.select-location = null;  
}
```

I hope this helps and best of luck with your website!

-Josh

Actions based on the feedback

- 1) Added 'NULL' as empty first value in dropdowns on Tour Guide, Assignment and Location pages.
- 2) Will display Customers, Guides, Locations by the user-appropriate name of the entity, but for bookings and assignments, feel that the ID number is most appropriate to display to user as there is not another name that we can display that would have meaning to the user, while the ID number clearly indicates the assignment or booking that resolves the M:M relationship.
 - a) Will implement on Bookings page and changing the Customer ID and Travel Location ID fields to be drop downs of more meaningful names for user, but was not able to finish for current step.
- 3) We understand the benefits of breaking up code into different files and will likely do that for the final submission, but ran out of time.

Upgrades to the Draft version

- 1) We realized that we needed price per adult and price per child attributes in the Travel Location entity in order to calculate the price for each booking. This was added after Step 5.

Project Step 6 Draft Version: Implement UPDATE + DELETE Operations

Feedback by the peer reviewer

- 1) Shawn H Hi Group 25,
I like the summary page, easy to understand what the pages are for. Nice use of error handling / output in your node files, and your code is clean and easy to read. No feedback on the SQL statements

On your blockers:

The edit pages aren't displaying correctly because of a change in context - e.g. customer/ versus customer/1 . We ran into the same issue, still debating on how to solve it though, since a few potential methods require a change in how we're working with the express routers. The following article might help as a reference:

<https://stackoverflow.com/questions/19035373/how-do-i-redirect-in-expressjs-while-passing-some-context>

If you want to use JS, you might consider `Date.now()` - depending on where you implement it (e.g. client or server side, server time or client local time, etc.), that might change how you'd use it. I believe that there's a similar function in SQL if you want to insert it directly, but I've not used it.

Also, assuming you want to format it, and do so on the data you pass from your express api, you can trim / format your departure and arrival date/time information using `DATE_FORMAT`. The MYSQL reference has the flag listings.

2) Arjun Kahlon
Hello Aishwarya and Amy,

Your website seems to be down. I'm trying to connect and it's giving me an error so I'm assuming you're making changes and restarting your flip process. When you restart it, can you let me know so I can view your URL?

To answer your questions:

How do you insert the current day's date using js for booking instead of asking the user to input a date?

Use `new Date()` to generate a Date object which stores both the current time and current date.

```
var today = new Date();  
var currentDate = String(today.getDate().padStart(2, '0'));  
console.log(currentDate)
```

Why is the CSS on the edit pages not displaying correctly?

I'm not seeing your CSS code. Currently browsing through your zipped file and can't seem to locate it.

Looks like you got the meat of your project working correctly which is great. Try messing with the current time code to see if that works for you. If you want it in the "2019-09-03", you can try

```
today = new Date().toISOString().slice(0, 10)  
console.log(today)
```

Actions based on the feedback:

- 1) Updated the css for update pages
- 2) Changed the date format for departure date, arrival date, booking date and payment date to mm/dd/yyyy
- 3) Used javascript to parse current date directly on adding new booking and payment

Upgrades to the Draft version:

- 1) Added the `booking_ID` in the assignment page as UNIQUE constraint to avoid multiple assignments for a single booking.
- 2) Added date validation in add-booking forms for departure date and arrival date.