Final Project

Sammy Pardes
IST659 M401
12/16/19

PART 1

Project Summary ……………………………………………………………………………………… 2

Data Questions ……………………………………………………………………………………… 2

Business Rules ……………………………………………………………………………………… 2

Glossary ……………………………………………………………………………………… 3

ERD ……………………………………………………………………………………… 4

Normalized Logical Model ……………………………………………………………………… 5

PART 2

Physical Database Design ……………………………………………………………………… 6

Data Creation ……………………………………………………………………………………… 9

Data Manipulation Code & Answering Data Questions ……………………………………… 11

Implementation ……………………………………………………………………………………… 13

Reflection ……………………………………………………………………………………… 14

Final Summary ……………………………………………………………………………………… 15

Project Summary

In the travel industry, there are a lot of considerations when it comes to booking a trip. Many choose to enlist the help of a travel agency to get the job done right. From managing clients to partnerships with hotels, travel agencies have a lot to keep track of. An organized database would make the life of a travel agency much easier. Borten Overseas is such an agency. As they continue to grow, Borten Overseas will need a database to help schedule and maintain all their bookings. Agents will use the database to efficiently cover for when their co-workers are out of office, correctly record their compensation, and easily manage information about their respective destination vendors.

Data Questions

1. How many bookings is each agent responsible for?
2. What destination(s) does each agent specializes in?
3. Where is each client going?
4. Where will the client be staying on his or her trip?

Business Rules

Agents have uniquely identifying employee IDs. They also keep their names, emails, addresses, and tenure on file. Each agent may be responsible for more than one booking, but bookings can only be handled by one agent. Agents may specialize in several destinations. They are responsible for scheduling bookings on behalf of their clients.

Clients may take more than one trip with the agency. They may purchase as many bookings as they would like. The agency keeps track of client names, mailing addresses, email addresses (which must be unique), and, if desired, the number of people in the client's party.

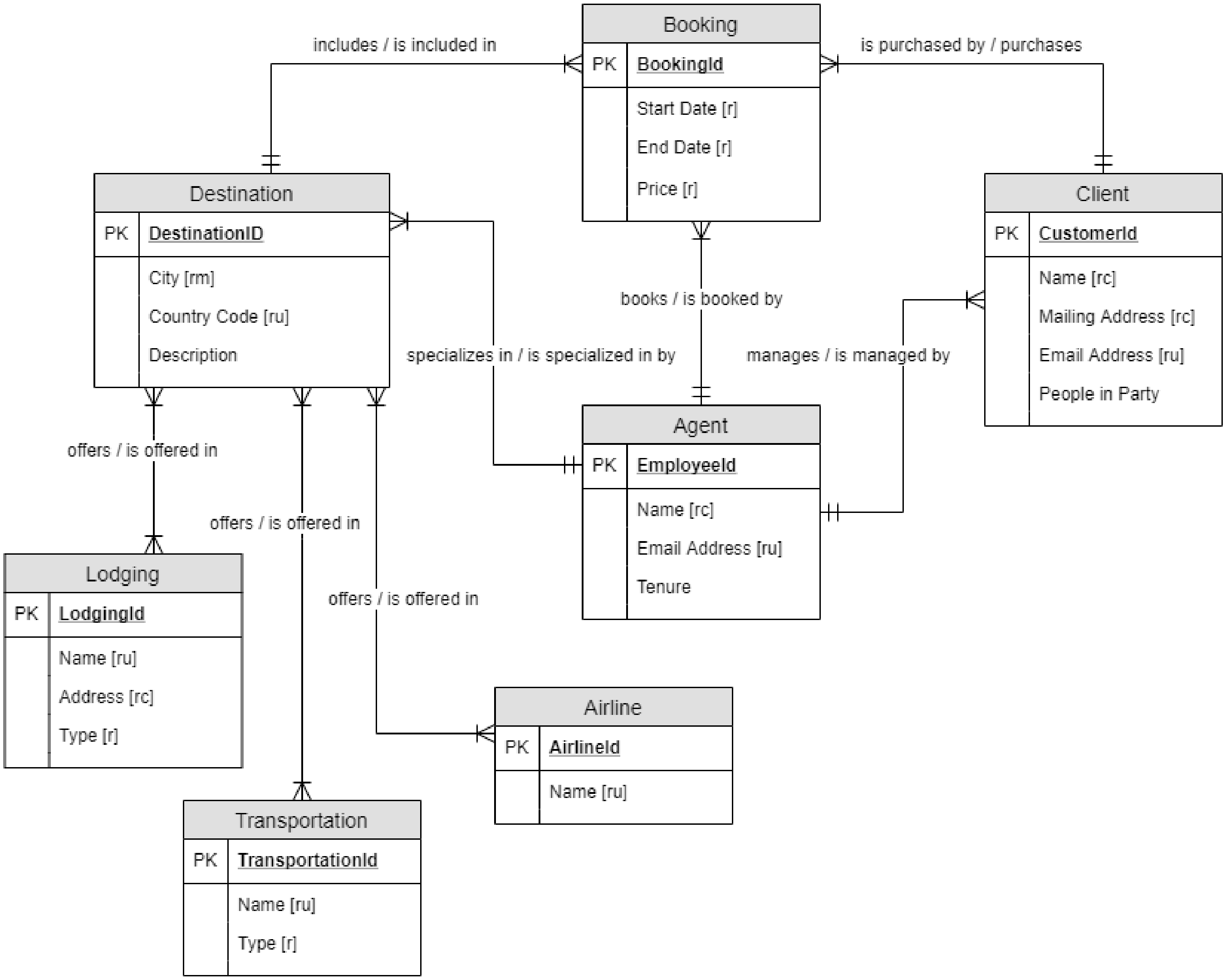
Every booking must have a start date, end date, and a total price. Bookings are scheduled for one and only one destination. Bookings belong to one and only one client.

Each country, or destination that is available will have an optional description and one or more cities of interest. For each destination, lodging, transportation, and airlines are available. Each must have a unique name. Lodgings will additionally list their address and accommodation type. Transportation must specify the mode of ground transit.

Glossary

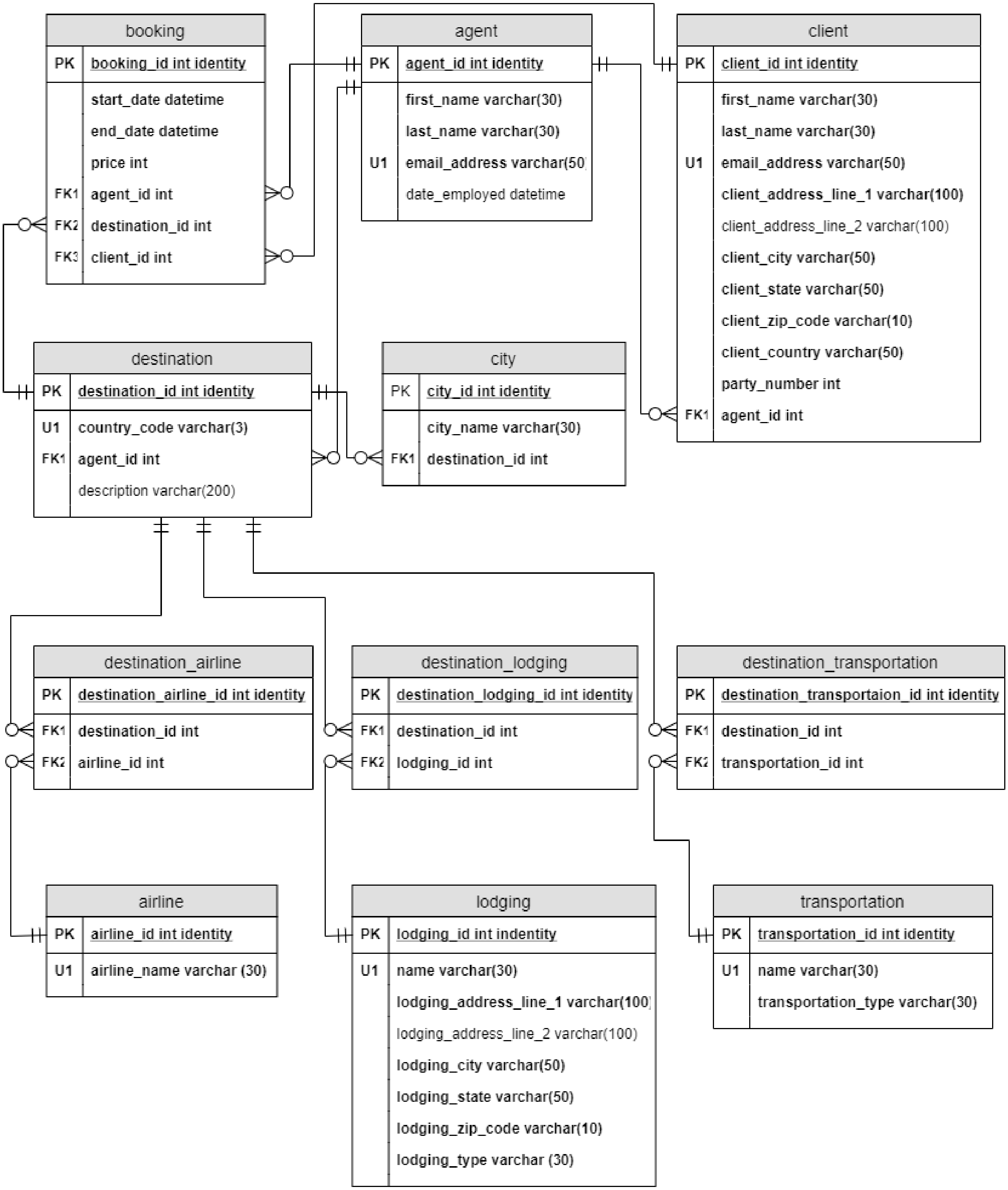
Entity	Attribute	Properties
Booking	Start Date	Required
	End Date	Required
	Price	Required
Client	Name	Required, composite of first and last name
	Mailing Address	Required, composite of address, city, state, zip, and country
	Email Address	Required, must be unique
	People in Party	Optional
Agent	Name	Required, composite of first and last name
	Email Address	Required, must be unique
	Tenure	Optional
Destination	City	Required, can be multi-valued
	Country Code	Required, must be unique
	Description	Optional
Lodging	Name	Required, must be unique
	Address	Required, composite of address, city, state, zip, and country
	Type	Required
Transportation	Name	Required, must be unique
	Type	Required
Airline	Name	Required, must be unique

Conceptual Model



Normalized Logical Model

- All primary keys implemented as surrogate keys.


booking
PK booking_id int identity
start_date datetime
end_date datetime
price int
FK1 agent_id int
FK2 destination_id int
FK3 client_id int
agent
PK agent_id int identity
first_name varchar(30)
last_name varchar(30)
U1 email_address varchar(50)
date_employed datetime
client
PK client_id int identity
first_name varchar(30)
last_name varchar(30)
U1 email_address varchar(50)
client_address_line_1 varchar(100)
client_address_line_2 varchar(100)
client_city varchar(50)
client_state varchar(50)
client_zip_code varchar(10)
client_country varchar(50)
party_number int
FK1 agent_id int
destination
PK destination_id int identity
U1 country_code varchar(3)
FK1 agent_id int
description varchar(200)
city
PK city_id int identity
city_name varchar(30)
FK1 destination_id int
destination_airline
PK destination_airline_id int identity
FK1 destination_id int
FK2 airline_id int
destination_lodging
PK destination_lodging_id int identity
FK1 destination_id int
FK2 lodging_id int
destination_transportation
PK destination_transportaion_id int identity
FK1 destination_id int
FK2 transportation_id int
airline
PK airline_id int identity
U1 airline_name varchar (30)
lodging
PK lodging_id int indentity
U1 name varchar(30)
lodging_address_line_1 varchar(100)
lodging_address_line_2 varchar(100)
lodging_city varchar(50)
lodging_state varchar(50)
lodging_zip_code varchar(10)
lodging_type varchar (30)
transportation
PK transportation_id int identity
U1 name varchar(30)
transportation_type varchar(30)

Physical Database Design

```
--drop statements
--drop tables
drop table if exists Borten_DestinationAirline
drop table if exists Borten_DestinationLodging
drop table if exists Borten_DestinationTransportation
drop table if exists Borten_Airline
drop table if exists Borten_Lodging
drop table if exists Borten_Transportation
drop table if exists Borten_Booking
drop table if exists Borten_City
drop table if exists Borten_Destination
drop table if exists Borten_Client
drop table if exists Borten_Agent

--drop procedures
drop procedure if exists borten_ChangeClientEmail

--drop functions
drop function if exists borten_AgentBookingsCount

drop view if exists borten_AgentBookingReport
drop view if exists borten_AgentDestinationReport
drop view if exists borten_ClientDestinationReport
drop view if exists borten_ClientLodgingReport
-- end drop statements

-- create table statements
create table Borten_Agent (
       agent_id int identity primary key,
       agent_first_name varchar(30) not null,
       agent_last_name varchar(30) not null,
       agent_email_address varchar(50) not null unique,
       date_employed datetime default getdate()
)

create table Borten_Destination (
       destination_id int identity primary key,
       country_code varchar(3) unique not null,
       agent_id int not null foreign key references Borten_Agent(agent_id),
       destination_description varchar(200)
)

create table Borten_City (
       city_id int identity primary key,
       city_name varchar(30) not null unique,
       destination_id int not null foreign key references Borten_Destination(destination_id)
)

create table Borten_Client (
       client_id int identity primary key,
       first_name varchar(30) not null,
       last_name varchar(30) not null,
       client_email_address varchar(50) not null unique,
       client_address_line_1 varchar(50) not null,
       client_address_line_2 varchar(50),
       client_city varchar(30) not null,
       client_state varchar(30) not null,
       client_zip_code varchar(10) not null,
       client_country varchar(30) not null,
       party_number int,
       agent_id int not null foreign key references Borten_Agent(agent_id)
)

create table Borten_Booking (
       booking_id int identity primary key,
       begin_date datetime not null,
       end_date datetime not null,
       price int not null,
       agent_id int not null foreign key references Borten_Agent(agent_id),
       destination_id int not null foreign key references Borten_Destination(destination_id),
       client_id int not null foreign key references Borten_Client(client_id)
)

create table Borten_Airline (
       airline_id int identity primary key,
       airline_name varchar(30) not null unique
)

create table Borten_Lodging (
       lodging_id int identity primary key,
       lodging_name varchar(30) not null unique,
       lodging_address_line_1 varchar(50) not null,
       lodging_address_line_2 varchar(50),
       lodging_city varchar(30) not null,
       lodging_state varchar(30) not null,
       lodging_zip varchar(10) not null,
       lodging_type varchar(30) not null
)

create table Borten_Transportation (
       transportation_id int identity primary key,
       transportation_name varchar(30) not null unique,
       transportation_type varchar(30) not null
)

create table Borten_DestinationAirline(
       destination_airline_id int identity primary key,
       destination_id int not null foreign key references Borten_Destination(destination_id),
       airline_id int not null foreign key references Borten_Airline(airline_id),
       constraint u1_Borten_DestinationAirline unique(destination_id, airline_id)
)

create table Borten_DestinationLodging (
       destination_lodging_id int identity primary key,
       destination_id int not null foreign key references Borten_Destination(destination_id),
       lodging_id int not null foreign key references Borten_Lodging(lodging_id),
       constraint u1_Borten_DestinationLodging unique(destination_id, lodging_id)
)

create table Borten_DestinationTransportation (
       destination_transportation_id int identity primary key,
       destination_id int not null foreign key references Borten_Destination(destination_id),
	transportation_id int not null foreign key references
Borten_Transportation(transportation_id),
	constraint u1_Borten_DestinationTransportation unique(destination_id, transportation_id)
)
-- end create table statements

--create a procedure to update a client's email address given first and last name
GO
CREATE PROCEDURE borten_ChangeClientEmail (@clientFirstName varchar(30), @clientLastName
varchar(30), @newEmail varchar(50))
AS
BEGIN
	UPDATE Borten_Client SET client_email_address = @newEmail
	WHERE last_name = @clientLastName AND first_name = @clientFirstName
END
GO
--end create client email update procedure

--create views and functions
--create function to count bookings for each agent
GO
CREATE FUNCTION dbo.borten_AgentBookingsCount(@agentID int)
RETURNS int AS
BEGIN
	DECLARE @returnValue int
	SELECT @returnValue = COUNT(Borten_Booking.agent_id) FROM Borten_Booking
	WHERE Borten_Booking.agent_id = @agentID
	RETURN @returnValue
END
GO
--end create count bookings function

--create destinations for each agent view
GO
CREATE VIEW borten_AgentDestinationReport AS
	SELECT
	Borten_Agent.agent_first_name,
	Borten_Destination.destination_description
	FROM Borten_Destination
	JOIN Borten_Agent ON Borten_Agent.agent_id = Borten_Destination.agent_id
GO
--end create destination view

--create client destinations view
GO
CREATE VIEW borten_ClientDestinationReport AS
	SELECT
	Borten_Client.first_name,
	Borten_Client.last_name,
	Borten_Destination.destination_description,
	Borten_Agent.agent_first_name
	FROM Borten_Client
	JOIN Borten_Booking ON Borten_Booking.client_id = Borten_Client.client_id
	JOIN Borten_Destination ON Borten_Destination.destination_id =
Borten_Booking.destination_id
	JOIN Borten_Agent ON Borten_Agent.agent_id = Borten_Booking.agent_id
GO
--end create client destination view
```

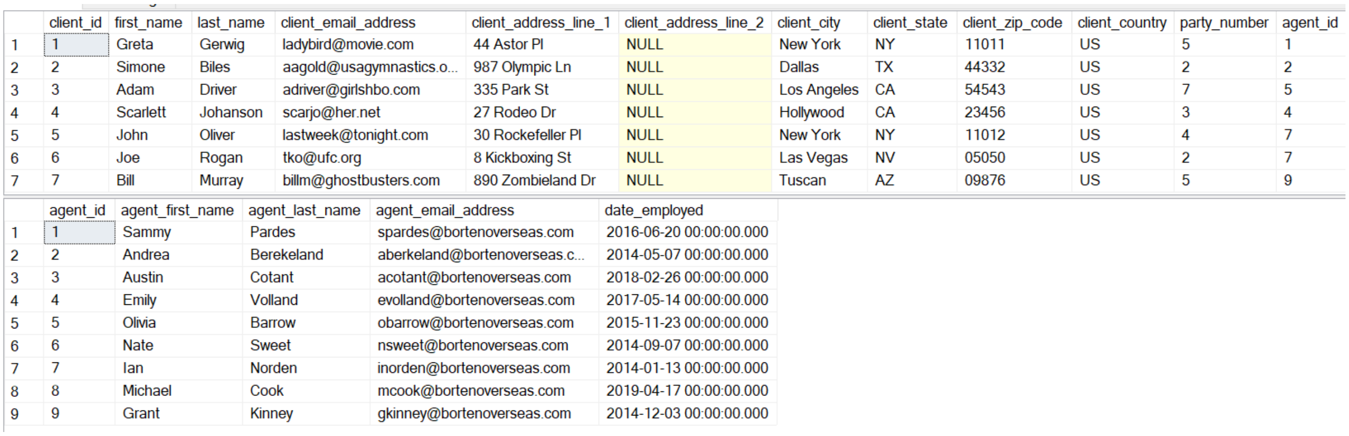
```
--create view for client lodging
GO
CREATE VIEW borten_ClientLodgingReport AS
       SELECT
       Borten_Client.first_name,
       Borten_Client.last_name,
       Borten_Destination.destination_description,
       Borten_Lodging.lodging_name
       FROM Borten_Client
       JOIN Borten_Booking on Borten_Booking.client_id = Borten_Client.client_id
       JOIN Borten_Destination ON Borten_Destination.destination_id = Borten_Booking.booking_id
       JOIN Borten_DestinationLodging ON Borten_DestinationLodging.destination_id =
Borten_Destination.destination_id
       JOIN Borten_Lodging on Borten_Lodging.lodging_id = Borten_DestinationLodging.lodging_id
GO
--end create views
```

Data Creation Code

```
-- begin instert statements
INSERT INTO Borten_Agent
              (agent_first_name, agent_last_name, agent_email_address, date_employed)
VALUES ('Sammy', 'Pardes', 'spardes@bortenoverseas.com', '6/20/2016'),
              ('Andrea', 'Berekeland', 'aberkeland@bortenoverseas.com', '5/7/2014'),
              ('Austin', 'Cotant', 'acotant@bortenoverseas.com', '2/26/2018'),
              ('Emily', 'Volland', 'evolland@bortenoverseas.com', '5/14/2017'),
              ('Olivia', 'Barrow', 'obarrow@bortenoverseas.com', '11/23/2015'),
              ('Nate', 'Sweet', 'nsweet@bortenoverseas.com', '9/7/2014'),
              ('Ian', 'Norden', 'inorden@bortenoverseas.com', '1/13/2014'),
              ('Michael', 'Cook', 'mcook@bortenoverseas.com', '4/17/2019'),
              ('Grant', 'Kinney', 'gkinney@bortenoverseas.com', '12/3/2014')
GO

INSERT INTO Borten_Destination
              (country_code, agent_id, destination_description)
VALUES ('AUS','1','Australia'), ('DNK', '2', 'Denmark'), ('ISL', '3', 'Iceland'),
              ('IRL', '4', 'Ireland'),  ('JPN', '5', 'Japan'), ('NLD', '6', 'Netherlands'),
              ('GBR', '7', 'United Kingdom'), ('FRA', '8', 'France'), ('MEX', '9', 'Mexico'),
              ('SWE', '1', 'Sweden'), ('NZL', '2', 'New Zealand'), ('IND', '3', 'India'),
              ('FJI', '4', 'Fiji'), ('BRA', '5', 'Brazil'), ('ZAF', '6', 'South Africa'),
              ('TCA', '7', 'Turks and Caicos'), ('BHS', '8', 'Bahamas'), ('AIA', '9',
'Anguilla')
GO

INSERT INTO Borten_City
              (city_name, destination_id)
VALUES ('Sydney', '1'), ('Copenhagen', '2'), ('Reykjavik', '3'), ('Dublin', '4'), ('Tokyo',
'5'),
              ('Amsterdam', '6'), ('London', '7'), ('Paris', '8'), ('Cancun', '9'),
('Stockholm', '10'),
              ('Auckland', '11'), ('New Delhi', '12'), ('Suva', '13'), ('Rio de Janeiro', '14'),
('Johannesburg', '15'),
              ('Cockburn Town', '16'), ('Nassau', '17'), ('The Valley', '18'), ('Melbourne',
'1'), ('Nice', '8'),
              ('Osaka', '5'), ('Manchester', '7'), ('Perth', '1'), ('Rotterdam', '6'), ('Mexico
City', '9')
GO
```

```
INSERT INTO Borten_Airline
              (airline_name)
VALUES ('Delta'), ('JetBlue'), ('SouthWest'), ('American Airlines'), ('United'), ('Japan
Airlines'), ('Emirates'), ('Caribbean Airlines')
GO

INSERT INTO Borten_Lodging
              (lodging_name, lodging_address_line_1, lodging_city, lodging_state, lodging_zip,
lodging_type)
VALUES ('Four Seasons Sydney','199 George St','The Rocks','New South Wales','2000','Hotel'),
              ('Steel House Copenhagen','Herholdtsgade 6','Kobenhavn','Copenhagen','','Hostel'),
              ('Galaxy Pod Hostel','Laugavegur 172','Reykjavík','Reykjavík','','Hostel'),
              ('Clontarf Castle Hotel','Castle Ave','lontarf East','Dublin','','Hotel')
GO

INSERT INTO Borten_DestinationLodging
              (destination_id, lodging_id)
VALUES ('1', '1'),
              ('2', '2'),
              ('3', '3'),
              ('4', '4')
GO

INSERT INTO Borten_Client
              (first_name, last_name, client_email_address, client_address_line_1, client_city,
client_state, client_zip_code, client_country, party_number, agent_id)
VALUES ('Greta', 'Gerwig', 'ladybird@movie.com','44 Astor Pl', 'New York', 'NY', '11011',
'US','5','1'),
              ('Simone', 'Biles', 'aagold@usagymnastics.org', '987 Olympic Ln', 'Dallas','TX',
'44332', 'US', '2', '2'),
              ('Adam', 'Driver', 'adriver@girlshbo.com', '335 Park St', 'Los Angeles','CA',
'54543', 'US', '7', '5'),
              ('Scarlett', 'Johanson', 'scarjo@her.net', '27 Rodeo Dr', 'Hollywood','CA',
'23456', 'US', '3', '4'),
              ('John', 'Oliver', 'lastweek@tonight.com', '30 Rockefeller Pl', 'New York','NY',
'11012', 'US', '4', '7'),
              ('Joe', 'Rogan', 'tko@ufc.org', '8 Kickboxing St', 'Las Vegas','NV', '05050',
'US', '2', '7'),
              ('Bill', 'Murray', 'billm@ghostbusters.com', '890 Zombieland Dr', 'Tuscan','AZ',
'09876', 'US', '5', '9')
GO

INSERT INTO Borten_Booking
              (begin_date, end_date, price, agent_id, destination_id, client_id)
VALUES ('1/22/20', '2/5/20','7000','1','3','1'),
              ('4/7/20', '4/27/20', '3250', '2', '7', '2'),
              ('9/4/20', '10/1/20', '8750', '5', '17', '3'),
              ('8/16/20', '8/25/20', '2500', '4', '8', '4'),
              ('3/9/20', '4/9/20', '5525', '7', '4', '5'),
              ('3/27/20', '5/2/20', '10500', '7', '10', '6'),
              ('5/14/20', '5/29/20', '2300', '9', '12', '7')
GO
--end insert statements

--begin select statements to show inserts were successful
SELECT * FROM Borten_Client
SELECT * FROM Borten_Agent
```

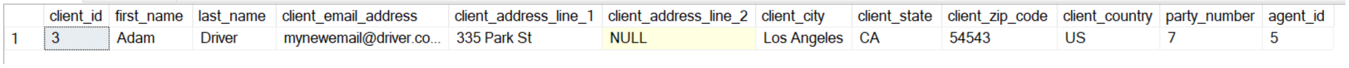
```
--end select statements
```



	client_id	first_name	last_name	client_email_address	client_address_line_1	client_address_line_2	client_city	client_state	client_zip_code	client_country	party_number	agent_id
1	1	Greta	Gerwig	ladybird@movie.com	44 Astor Pl	NULL	New York	NY	11011	US	5	1
2	2	Simone	Biles	aagold@usagymnastics.o...	987 Olympic Ln	NULL	Dallas	TX	44332	US	2	2
3	3	Adam	Driver	adriver@girlshbo.com	335 Park St	NULL	Los Angeles	CA	54543	US	7	5
4	4	Scarlett	Johanson	scarjo@her.net	27 Rodeo Dr	NULL	Hollywood	CA	23456	US	3	4
5	5	John	Oliver	lastweek@tonight.com	30 Rockefeller Pl	NULL	New York	NY	11012	US	4	7
6	6	Joe	Rogan	tko@ufc.org	8 Kickboxing St	NULL	Las Vegas	NV	05050	US	2	7
7	7	Bill	Murray	billm@ghostbusters.com	890 Zombieland Dr	NULL	Tuscan	AZ	09876	US	5	9

	agent_id	agent_first_name	agent_last_name	agent_email_address	date_employed
1	1	Sammy	Pardes	spardes@bortenoverseas.com	2016-06-20 00:00:00.000
2	2	Andrea	Berekeland	aberkeland@bortenoverseas.c...	2014-05-07 00:00:00.000
3	3	Austin	Cotant	acotant@bortenoverseas.com	2018-02-26 00:00:00.000
4	4	Emily	Volland	evolland@bortenoverseas.com	2017-05-14 00:00:00.000
5	5	Olivia	Barrow	obarrow@bortenoverseas.com	2015-11-23 00:00:00.000
6	6	Nate	Sweet	nsweet@bortenoverseas.com	2014-09-07 00:00:00.000
7	7	Ian	Norden	inorden@bortenoverseas.com	2014-01-13 00:00:00.000
8	8	Michael	Cook	mcook@bortenoverseas.com	2019-04-17 00:00:00.000
9	9	Grant	Kinney	gkinney@bortenoverseas.com	2014-12-03 00:00:00.000

```
--use email update procedure
EXEC borten_ChangeClientEmail 'Adam', 'Driver', 'mynewemail@driver.com'
SELECT * FROM Borten_Client WHERE first_name='Adam' and last_name ='Driver'
--end email update procedure
--end borten_ChangeClientEmail procedure
```



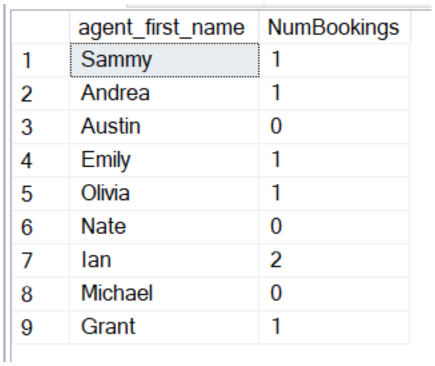
	client_id	first_name	last_name	client_email_address	client_address_line_1	client_address_line_2	client_city	client_state	client_zip_code	client_country	party_number	agent_id
1	3	Adam	Driver	mynewemail@driver.co...	335 Park St	NULL	Los Angeles	CA	54543	US	7	5

Data Manipulation Code & Answering Data Questions

How many bookings is each agent responsible for?

To determine how many bookings each agent is responsible for at any given time, I've created a function called "borten_AgentBookingsCount" that takes an Agent's ID as input and outputs the total number of bookings the specified agent has taken on. Below is a screenshot of a SELECT statement using the borten_AgentBookingsCount function to display each agent and their respective number of bookings. As you can easily, see Ian is currently the agent with the most bookings with a total of two. Austin, Nate, and Michael each have zero bookings and everyone else has one. While we want our agents to stay busy, we don't want them to be overwhelmed with too many clients.

```
--use borten_AgentBookingsCount function
SELECT Borten_Agent.agent_first_name, dbo.borten_AgentBookingsCount(Borten_Agent.agent_id) AS
NumBookings FROM Borten_Agent
--end count bookings for each agent function
```

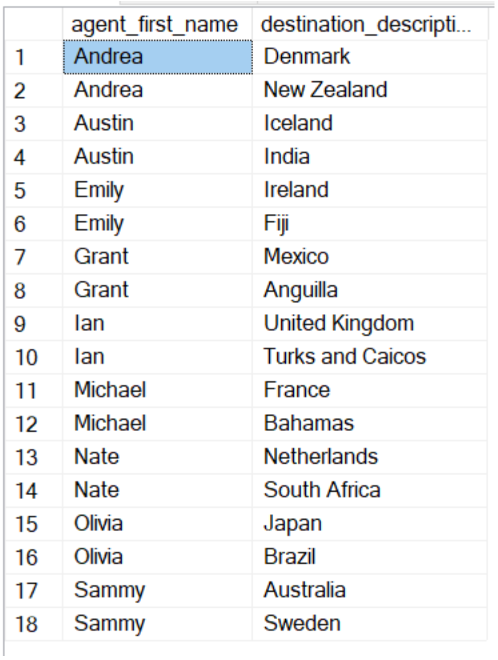


	agent_first_name	NumBookings
1	Sammy	1
2	Andrea	1
3	Austin	0
4	Emily	1
5	Olivia	1
6	Nate	0
7	Ian	2
8	Michael	0
9	Grant	1

What destination(s) does each agent specializes in?

Using the view entitled "borten_AgentDestinationsReport", we can easily find out the locations of expertise for each agent. Each agent at Borten Overseas specializes in two different travel destinations from our wide array of offerings.

```
--test out agent destinations view
SELECT * FROM borten_AgentDestinationReport ORDER BY agent_first_name
-- end agent destinations view
```



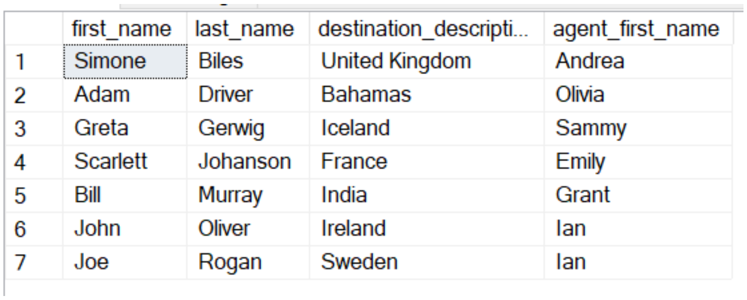
	agent_first_name	destination_descripti...
1	Andrea	Denmark
2	Andrea	New Zealand
3	Austin	Iceland
4	Austin	India
5	Emily	Ireland
6	Emily	Fiji
7	Grant	Mexico
8	Grant	Anguilla
9	Ian	United Kingdom
10	Ian	Turks and Caicos
11	Michael	France
12	Michael	Bahamas
13	Nate	Netherlands
14	Nate	South Africa
15	Olivia	Japan
16	Olivia	Brazil
17	Sammy	Australia
18	Sammy	Sweden

Where is each client going?

To quickly view where each client is headed and the agent they are working with, the view called "borten_ClientDestinationReport" comes in handy. Here we can see that each of our clients is traveling to a unique destination and have been paired with one of our talented agents.

```
--test out client destinations view
```

```
SELECT * FROM borten_ClientDestinationReport ORDER BY last_name
--end client destinations view
```

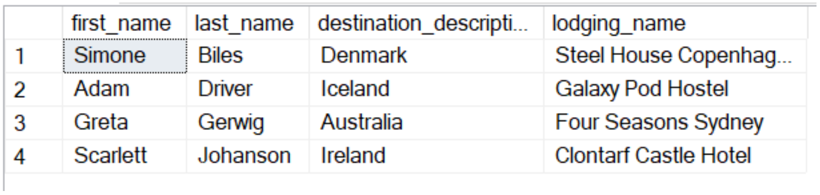


	first_name	last_name	destination_descripti...	agent_first_name
1	Simone	Biles	United Kingdom	Andrea
2	Adam	Driver	Bahamas	Olivia
3	Greta	Gerwig	Iceland	Sammy
4	Scarlett	Johanson	France	Emily
5	Bill	Murray	India	Grant
6	John	Oliver	Ireland	Ian
7	Joe	Rogan	Sweden	Ian

Where will the client be staying on his or her trip?

For our clients who have settled on lodging at their respective destinations, we can use the "borten_ClientLodgingReport" view to pull up the latest information on they will be staying for the duration of each of their trips.

```
--test out client lodging view
SELECT * FROM borten_ClientLodgingReport ORDER BY last_name
--end create views
```



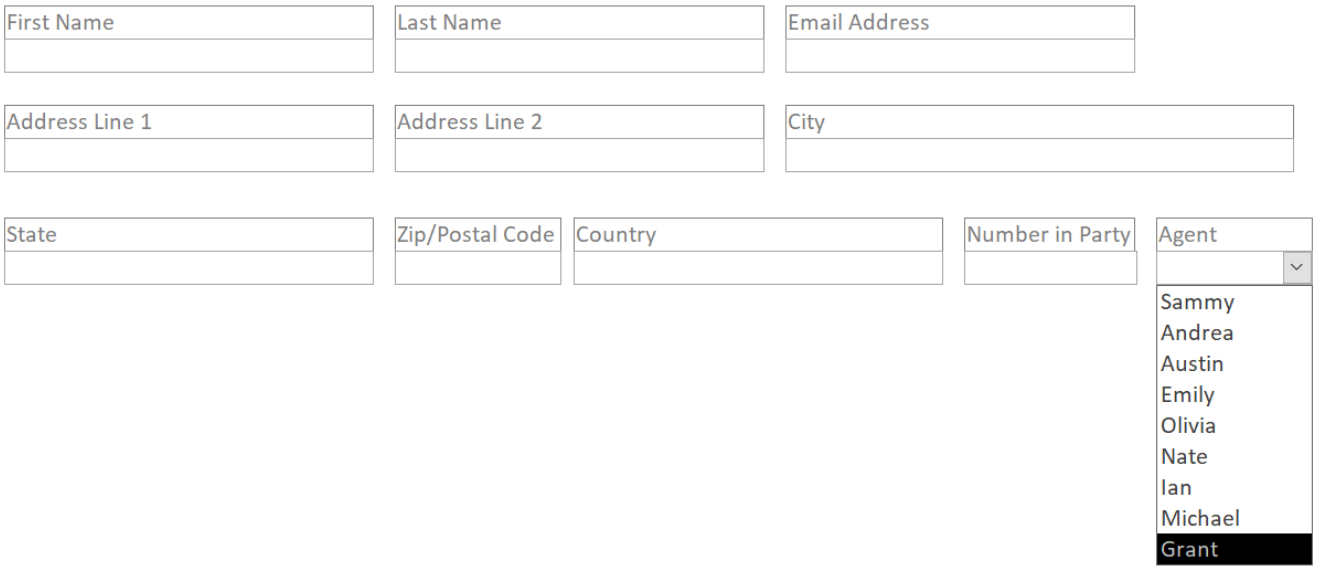
	first_name	last_name	destination_descripti...	lodging_name
1	Simone	Biles	Denmark	Steel House Copenhag...
2	Adam	Driver	Iceland	Galaxy Pod Hostel
3	Greta	Gerwig	Australia	Four Seasons Sydney
4	Scarlett	Johanson	Ireland	Clontarf Castle Hotel

Implementation

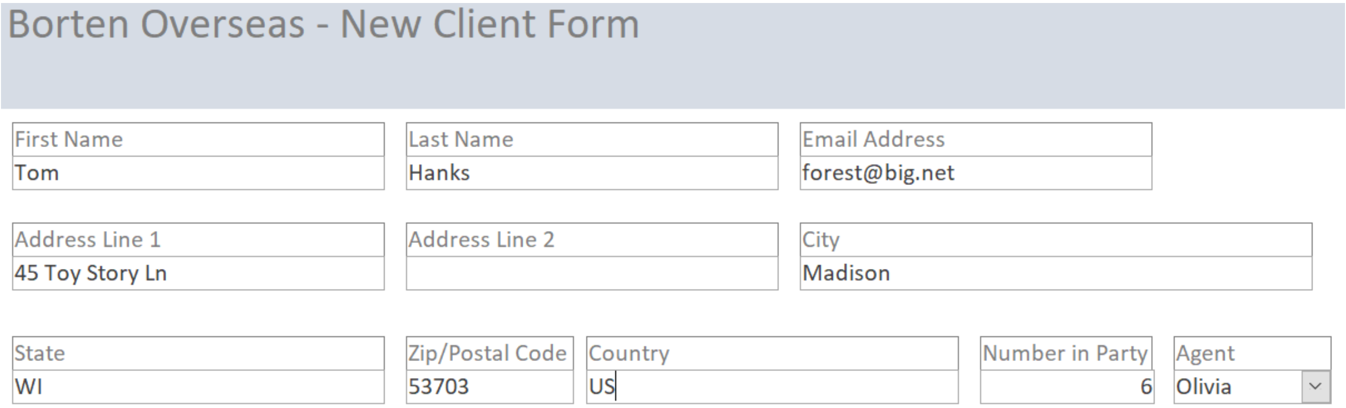
Microsoft Access provides an easy and efficient way for the Borten Overseas agents to input client and booking information. Since our agents are not SQL experts, they will use Access to fill out the New Client Form and insert incoming client data directly into the Borten Overseas database.

Below, our newest client was successfully entered into the database using the New Client Form I set up in Access.

```
--create form in Access to enter new client information
--agent set to dropdown menu
```

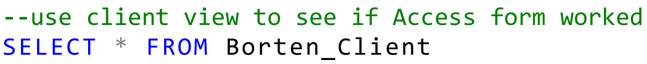


First Name	Last Name	Email Address		
Address Line 1	Address Line 2	City		
State	Zip/Postal Code	Country	Number in Party	Agent

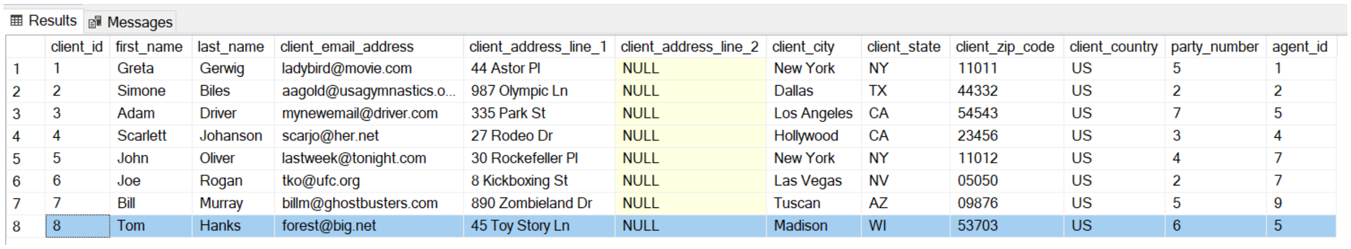


Borten Overseas - New Client Form

First Name	Last Name	Email Address		
Tom	Hanks	forest@big.net		
Address Line 1	Address Line 2	City		
45 Toy Story Ln		Madison		
State	Zip/Postal Code	Country	Number in Party	Agent
WI	53703	US	6	Olivia



```
--use client view to see if Access form worked
SELECT * FROM Borten_Client
```



	client_id	first_name	last_name	client_email_address	client_address_line_1	client_address_line_2	client_city	client_state	client_zip_code	client_country	party_number	agent_id
1	1	Greta	Gerwig	ladybird@movie.com	44 Astor Pl	NULL	New York	NY	11011	US	5	1
2	2	Simone	Biles	aagold@usagymnastics.o...	987 Olympic Ln	NULL	Dallas	TX	44332	US	2	2
3	3	Adam	Driver	mynewemail@driver.com	335 Park St	NULL	Los Angeles	CA	54543	US	7	5
4	4	Scarlett	Johanson	scarjo@her.net	27 Rodeo Dr	NULL	Hollywood	CA	23456	US	3	4
5	5	John	Oliver	lastweek@tonight.com	30 Rockefeller Pl	NULL	New York	NY	11012	US	4	7
6	6	Joe	Rogan	tko@ufc.org	8 Kickboxing St	NULL	Las Vegas	NV	05050	US	2	7
7	7	Bill	Murray	billm@ghostbusters.com	890 Zombieland Dr	NULL	Tuscan	AZ	09876	US	5	9
8	8	Tom	Hanks	forest@big.net	45 Toy Story Ln	NULL	Madison	WI	53703	US	6	5

Reflection

While I am overall pleased with the implementation of the Borten Overseas database, there are a few things I would do differently if I could re-do this assignment. Though the database is functional as is, there are a few steps that could improve the data entering process.

Since Lodging_Type and Transportation_Type will have a finite number of responses, I think creating lookup tables for the fields could improve my database. Additionally, I would have liked to come up with a better way to enter information into the DestinationLodging table. I had to ensure that I correctly matched up each destination and lodging ID for every row. This could easily lead to human error, especially if entering in a lot of data. Perhaps a better way would be to set up another form in Access that would use dropdown tables.

Final Summary

Thanks to the work done in Microsoft SQL Server and Microsoft Access, Borten Overseas now has a centralized database for all their business needs. Detailed records will be easily managed for new and old clients alike. Changes to client information will now be simple to do and universally updated within the Borten Overseas system. Now that agents are empowered to keep track of their individual bookings and their corresponding destinations, Borten Overseas has been brought into the 21st century and can continue to thrive!