TRAVEL COMPANY DATABASE: WORLD WANDERERS

Paige Gonzales

CS 5318

Spring 2023

Video Link: Final Project - Paige Gonzales.mp4

Table of Contents

Abstract

World Wanderers is a travel agency specializing in European vacations for American travelers. The company has grown significantly in the last year and would like to implement a database system to better keep track of staff, offices, available travel packages and trips, customer information and customer satisfaction. Upon implementation, the staff of World Wanderers will be able to update information on customer purchases, customer surveys, popular travel packages, staff information, office information, and trips booked.

Mission Statement

The *WorldWanderers* database system will efficiently update, manage, and deliver customer information, travel package booking history, customer survey feedback, and available travel packages so that the company can understand and cater to customers' needs, ensure the company is staffed appropriately for tours, and manage payments of booked travel packages.

Mission Objectives

- To maintain data on travel packages offered
- To maintain data on travel package costs
- To maintain data on trip availability
- To maintain data on travel packages booked
- To maintain data on trip countries
- To maintain data on travel dates offered
- To maintain data on staff employed
- To maintain data on customer information
- To maintain data on customer purchases
- To maintain data on customer feedback

- To perform searches on travel packages offered
- To perform searches on travel package costs
- To perform searches on trip availability
- To perform searches on travel packages booked
- To perform searches on trip countries

- To perform searches on travel dates offered
- To perform searches on staff employed
- To perform searches on customer information
- To perform searches on customer purchases
- To perform search on customer feedback

- To track the status of available travel packages
- To track the status of customer payments due

- To report on travel packages booked
- To report on staff availability
- To report on popular travel dates
- To report on customer trends
- To report on customer purchases
- To report on customer feedback

Use Cases

Use Case 1: A marketer wants to add a new travel package for Italy into the database.

Actor: Staff

Steps:

1. A member of staff selects “Travel Packages.”
2. Member of Staff clicks on “New Travel Package.”
3. A new window opens asking for package name, package country location, duration, max capacity, and managing staff ID to be filled in by the staff member.
4. The staff member inputs required information and clicks “Save.”

Use Case 2: A member of staff wants to delete a travel package from the database as the company will no longer offer it.

Actor: Staff

Steps:

1. A member of staff selects “Travel Packages.”

2. A member of staff selects option to “Search for Travel Package.”
3. The staff member is prompted to type the package name that needs to be searched for in the database.
4. The member selects the package name and selects “Delete Package.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 3: A member of staff wants to update a travel package’s max capacity.

Actor: Staff

Steps:

1. A member of staff selects “Travel Packages.”
2. A member of staff selects option to “Search for Travel Package.”
3. The staff member is prompted to type the package name to be edited in the search bar.
4. A member of staff selects option to “Edit Package.”
5. The staff member types in the new max capacity and selects “Okay.”

Use Case 4: A member of staff wants to view how many staff members are marketing travel packages for each country.

Actor: Staff

Steps:

1. A member of staff selects “Travel Packages.”
2. The member selects option: “Travel Packages by Country” to display travel packages grouped by country.
3. The application will display statistics for countries, including how many staff members are marketing for each country.

Use Case 5: A new customer has signed up and needs to be added to the database.

Actor: Customer

Steps:

1. The customer is prompted to enter their name and contact details of first name, last name, address, phone, and email. First name, last name, phone number and email are required, and the customer cannot continue unless these are fields are filled in.
2. The customer selects “Sign Up.”

3. The information is added to the database along with a generated customer ID and staff ID.

Use Case 6: A member of staff wants to remove a customer from the database.

Actor: Staff

Steps:

1. A member of staff selects “Customers.”
2. A member of staff selects option to “Search for Customer.”
3. The member is promoted to type in the customer’s ID in the search bar.
4. The member finds the customer and selects “Delete Customer.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 7: A customer has changed their phone number since booking the trip and wants to update their contact information.

Actor: Customer

Steps:

1. A member of staff selects option to “Customers.”
2. A member of staff selects option to “Search for Customer.”
3. The member is prompted to type in the customer’s ID in the search bar.
4. The staff member selects option to “Edit Customer.”
5. The staff member types in the new phone number and selects “Okay.”

Use Case 8: A member of staff wants to know how many customers live in a certain state.

Actor: Staff

Steps:

1. A member of staff selects “Customers” where all customer details will be displayed.
2. The staff member will filter by state to only show customers who live in the desired state(s).
3. The application will display statistics for customers, including how many customers there are based on the filters applied.

Use Case 9: A member of staff has been hired and needs to be added to the database.

Actor: Staff

Steps:

1. A member of staff selects “Staff.”
2. Member of Staff clicks on “New Staff Member.”
3. A new window opens asking staffID, first name, last name, date of birth, email, office, and primary role to be filled in by the staff member. All the fields except primary roles are required.
4. The staff member inputs all information and clicks “Save.”

Use Case 10: A long-time staff member has retired and needs to be removed from the database.

Actor: Staff

Steps:

1. A member of staff selects “Staff.”
2. A member of staff selects option to “Search for Staff Member.”
3. The staff member is prompted to type the staff ID in the search bar.
4. The member locates the staff member and selects “Delete Staff Member.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 11: A member of staff has gotten married and needs to change their last name.

Actor: Staff

Steps:

1. A member of staff selects “Staff.”
2. A member of staff selects option to “Search for Staff Member.”
3. The staff member is prompted to type the staff ID in the search bar.
4. A member of staff selects option to “Edit Staff Member.”
5. The staff member types in the new last name and selects “Okay.”

Use Case 12: A member of staff wants to see the number of staff for each primary role: Marketer, Admin, and Guide.

Actor: Staff

Steps:

1. A member of staff selects “Staff” where all staff details will be displayed.

2. The staff member will filter by primary role to show how many staff members there are in each.

Use Case 13: A new office has opened and needs to be added to the database.

Actor: Staff

Steps:

1. A member of staff selects “Office.”
2. Member of Staff clicks on “New Office.”
3. A new window opens asking the staff member to fill in the office number, address, and phone number. Only office numbers are required.
4. The staff member inputs all information and clicks “Save.”

Use Case 14: An office has closed and needs to be removed from the database.

Actor: Staff

Steps:

1. A member of staff selects “Office.”
2. A member of staff selects option to “Search for Office.”
3. The staff member is prompted to type the office number in the search bar.
4. The member locates the office and selects “Delete Office.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 15: An existing office has relocated and needs to update the address.

Actor: Staff

Steps:

1. A member of staff selects “Office.”
2. A member of staff selects option to “Search for Office.”
3. The staff member is prompted to type the office number in the search bar.
4. A member of staff selects option to “Edit Office.”
5. The staff member types in the new address and selects “Okay.”

Use Case 16: A member of staff wants to see the number of offices located in each country.

Actor: Staff

Steps:

1. A member of staff selects “Office” where all staff details will be displayed.
2. The staff member will select the option to “Display offices by country” which will show how many offices are in each country.

Use Case 17: A new trip is becoming available next summer and needs to be added to the database.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability.”
2. Member of Staff clicks on “New Trip.”
3. A new window opens asking the staff member to fill in the trip ID, trip date, price, associated travel package, and staff ID who will guide the trip.
4. The staff member inputs all information and clicks “Save.”

Use Case 18: A trip has been cancelled and needs to be removed from the database.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability.”
2. A member of staff selects option to “Search for Trip.”
3. The staff member is prompted to type the trip ID in the search bar.
4. The member locates the trip and selects “Delete Trip.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 19: A price has changed for a trip and needs to be updated.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability.”
2. A member of staff selects option to “Search for Trip.”
3. The staff member is prompted to type the trip ID that needs to be searched for.
4. A member of staff selects option to “Edit Trip.”
5. The staff member types in the new price and selects “Okay.”

Use Case 20: A member of staff wants to see the average price of a trip in August 2023.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability” where all trip details will be displayed.
2. The staff member will filter by trip date to select month of August and year of 2023.
3. The display will show all trips available in August 2023 and their details.
4. The staff member will click “Show Statistics” which will show average trip price based on filter.

Use Case 21: A new booking has been made and needs to be added to the database.

Actor: Customer

Steps:

1. A customer purchases a trip by clicking the “Purchase” button.
2. The customer enters how many people they want to book for the trip.
3. The customer selects “Book.”
4. The transactionID, dateBooked, dueDate, totalCharge, tripID, X and customerID auto populate based on customer account information and search criteria and are added to the database.

Use Case 22: A booking has been cancelled and needs to be removed from the database.

Actor: Staff

Steps:

1. A member of staff selects “Bookings.”
2. A member of staff selects option to “Search for Booking.”
3. The staff member is prompted to type the transactionID in the search bar.
4. The member locates the trip and selects “Delete Booking.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 23: A customer wants to change the number of people they have booked for a trip.

Actor: Staff

Steps:

1. A member of staff selects “Bookings.”
2. A member of staff selects option to “Search for Booking.”

3. The staff member is prompted to type the transaction ID in the search bar.
4. A member of staff selects option to “Edit Booking.”
5. The staff member types in the new number of people to book and updates the total charge and selects “Okay.”

Use Case 24: A member of staff wants to see what the booking charges are for bookings with a due date of May 2023.

Actor: Staff

Steps:

1. A member of staff selects “Bookings” where all booking details are displayed.
2. The staff member will filter by booking date to select May 2023.
3. The display will show all trips booked in May 2023 and their details.
4. The staff member will click “Show Statistics” which will show total charges based on filter.

Use Case 25: A customer has returned from a trip and has been prompted to take a customer satisfaction survey.

Actor: Customer

Steps:

1. A customer follows the link in the customer outreach email to take the survey.
2. The customer enters their ratings on overall satisfaction, comfort, staff, facilities, value, and location on a scale of 1-10. The survey date is populated according to the date the survey was completed.
3. The customer selects “Submit.”

Use Case 26: A survey was prematurely submitted by a customer and needs to be deleted.

Actor: Staff

Steps:

1. A member of staff selects “Surveys.”
2. A member of staff selects option to “Search for Survey.”
3. The staff member is prompted to type the transactionID of the trip that needs to be searched for.

4. The member locates the trip and selects “Delete Survey.”
5. The member is prompted to confirm the deletion and selects “Okay.”

Use Case 27: A customer wants to change their survey responses.

Actor: Customer

Steps:

1. A customer follows the link in the customer outreach email to take the survey.
2. The customer selects “Edit Survey.”
3. The customer edits their ratings on overall satisfaction, comfort, staff, facilities, value, and location on a scale of 1-10. The survey date is updated according to the date the survey was edited.
4. The customer selects “Submit.”

Use Case 28: A member of staff wants to see what the average overall satisfaction rating is for surveys taken in the year 2022.

Actor: Staff

Steps:

1. A member of staff selects “Surveys” where all survey details are displayed.
2. The member of staff will filter to show only year 2022.
3. The staff member will click “Show Statistics” which will show the average survey rating by category based on filter selected.

Use Case 29: A member of staff wants to display the trip details and travel package details of trips being offered that have a duration of less than 14 days.

Actor: Staff

Steps:

1. A member of staff selects “Trips Available.”
2. The trip and package details will be displayed, including cost per trip.
3. The staff member will filter by number of days to only show trips less than 14 days.

Use Case 30: A member of staff wants to see the customer details of customers who are assigned to administrative staff.

Actor: Staff

Steps:

1. The staff member selects option “Customers.”
2. The staff member filters by staff member primary role as “Admin.”

Use Case 31: A staff member wants to view the travel package details of packages that are marketed by staff in office number 9.

Actor: Staff

Steps:

1. A member of staff selects “Travel Packages.”
2. The travel package details will be displayed, including the staff ID associated with each travel package.
3. The staff member will filter by staff member located in office 9.

Use Case 32: A member of staff wants to view the staff details of the staff members who work at a branch in Italy.

Actor: Staff

Steps:

1. A member of staff selects “Staff” where all staff details are displayed.
2. The staff member will filter by office country to only show only staff members in Italy offices.

Use Case 33: A member of staff wants to display all trip availability details for bookings that have been made in July 2022.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability” where all trip details will be displayed.
2. The staff member will filter by booking date of July 2022.
3. The display will show all trips booked in July 2022 and their details.

Use Case 34: A member of staff wants to display all customer details for bookings that have been made in December 2022.

Actor: Staff

Steps:

1. A member of staff selects “Bookings” where all booking details will be displayed.
2. The staff member will filter by booking date of December 2022.
3. The display will show all trips booked in December 2022 and their details.
4. The staff member will select the option “View Customer Details on Selection” which will show all customer information associated with the booking filter.

Use Case 35: A member of staff wants to display all booking details for surveys that have an overall satisfaction rate of less than 5.

Actor: Staff

Steps:

1. A member of staff selects “Bookings” where all booking details will be displayed.
2. The staff member will filter by survey overall satisfaction of less than 5.
3. The display will show all bookings with overall satisfaction of less than 5 and their details.

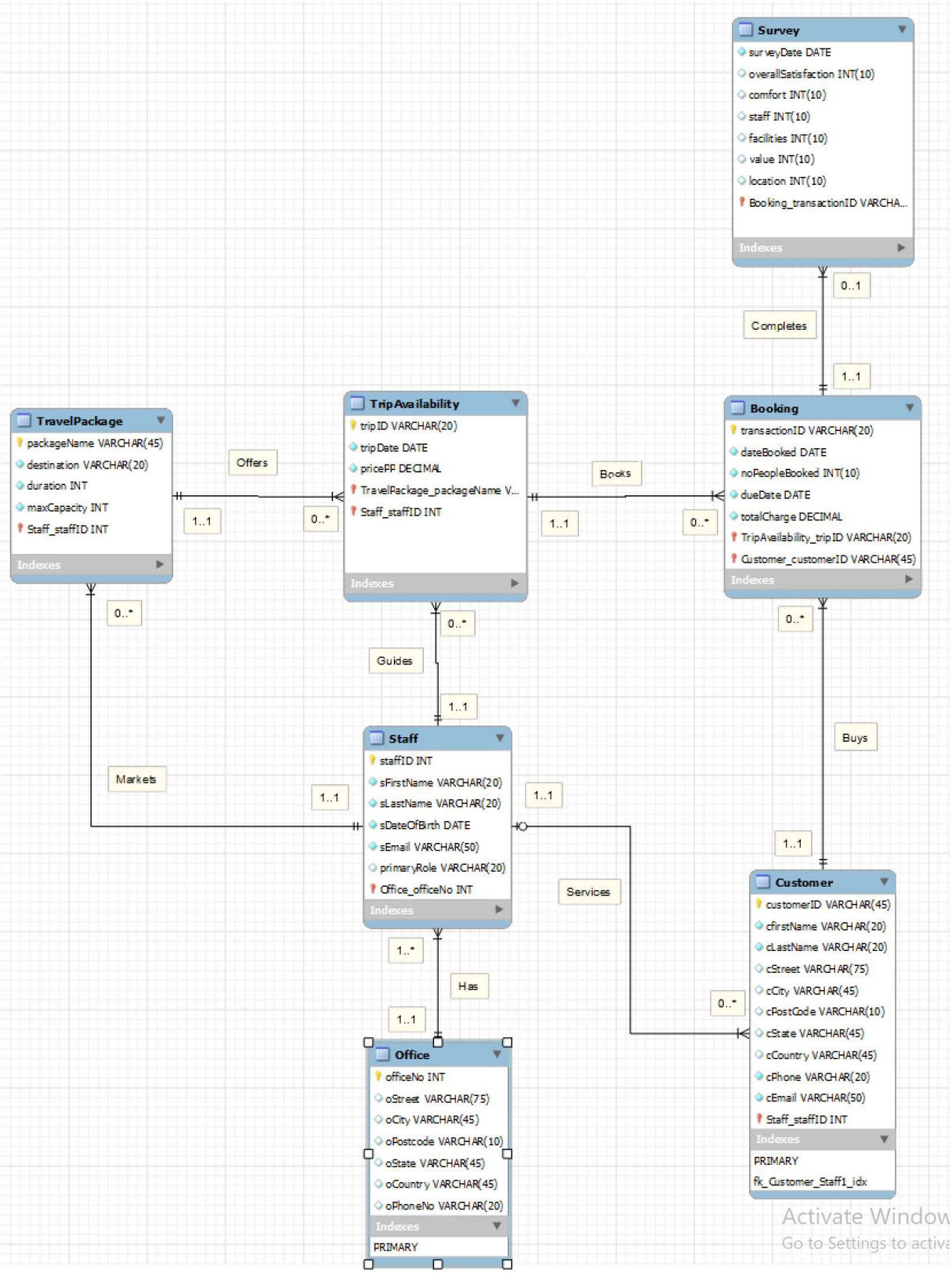
Use Case 36: A member of staff wants to display all staff details for staff that have been assigned a trip in September 2023.

Actor: Staff

Steps:

1. A member of staff selects “Trip Availability” where all trip details will be displayed.
2. The staff member will filter by trip date of September 2023.
3. The display will show all trips taking place in September 2023 and their details.

E/R Diagram


Survey
surveyDate DATE
overallSatisfaction INT(10)
comfort INT(10)
staff INT(10)
facilities INT(10)
value INT(10)
location INT(10)
Booking_transactionID VARCHA...
Indexes
0..1
Completes
1..1
TravelPackage
packageName VARCHAR(45)
destination VARCHAR(20)
duration INT
maxCapacity INT
Staff_staffID INT
Indexes
Offers
1..1
0..*
TripAvailability
tripID VARCHAR(20)
tripDate DATE
pricePP DECIMAL
TravelPackage_packageName V...
Staff_staffID INT
Indexes
Books
1..1
0..*
Booking
transactionID VARCHAR(20)
dateBooked DATE
noPeopleBooked INT(10)
dueDate DATE
totalCharge DECIMAL
TripAvailability_tripID VARCHAR(20)
Customer_customerID VARCHAR(45)
Indexes
0..*
0..*
Guides
1..1
Buys
1..1
Staff
staffID INT
sFirstName VARCHAR(20)
sLastName VARCHAR(20)
sDateOfBirth DATE
sEmail VARCHAR(50)
primaryRole VARCHAR(20)
Office_officeNo INT
Indexes
Markets
1..1
1..1
Services
1..*
Has
1..1
Customer
customerID VARCHAR(45)
cfirstName VARCHAR(20)
cLastName VARCHAR(20)
cStreet VARCHAR(75)
cCity VARCHAR(45)
cPostCode VARCHAR(10)
cState VARCHAR(45)
cCountry VARCHAR(45)
cPhone VARCHAR(20)
cEmail VARCHAR(50)
Staff_staffID INT
Indexes
PRIMARY
fk_Customer_Staff1_idx
0..*
Office
officeNo INT
oStreet VARCHAR(75)
oCity VARCHAR(45)
oPostcode VARCHAR(10)
oState VARCHAR(45)
oCountry VARCHAR(45)
oPhoneNo VARCHAR(20)
Indexes
PRIMARY
Activate Window
Go to Settings to activa

Relational Model

Staff(staffID, sFirstName, sLastName, sDateOfBirth, sEmail, primaryRole, officeNo)

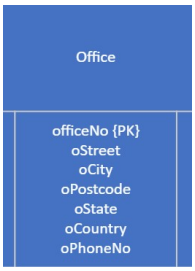
Primary Key: staffID

Foreign Key: officeNo references **Office**(officeNo)

Office(officeNo, oStreet, oCity, oPostcode, oState, oCountry, oPhoneNo)

Primary Key: officeNo

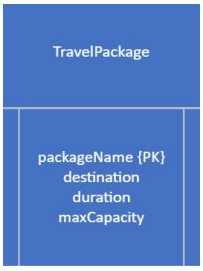
Foreign Key: None



TravelPackage(packageName, destination, duration, maxCapacity, staffID)

Primary Key: packageName

Foreign Key: staffID references **Staff**(staffID)



TripAvailability(tripID, tripDate, pricePP, packageName, staffID)

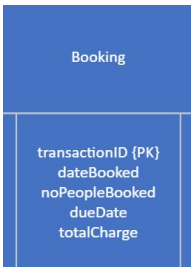
Primary Key: tripID

Foreign Keys: packageName references **TravelPackage**(packageName), staffID references **Staff**(staffID)

Booking(transactionID, dateBooked, noPeopleBooked, dueDate, totalCharge, tripID, customerID)

Primary Key: transactionID

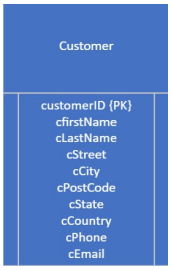
Foreign Keys: tripID references **TripAvailability**(tripID), customerID references **Customer**(customerID)



Customer(customerID, cFirstName, cLastName, cStreet, cCity, cPostcode, cState, cCountry, cPhone, cEmail, staffID)

Primary Key: customerID

Foreign Key: staffID references **Staff**(staffID)



Survey(surveyDate, overallSatisfaction, comfort, staff, facilities, value, location, transactionID)

Primary Key: transactionID

Foreign Key: transactionID references **Booking**(transactionID)

Normalization

Staff(staffID, sFirstName, sLastName, sDateOfBirth, sEmail, primaryRole, officeNo) has a primary key of staffID. The attribute officeNo is a foreign key referencing the **Office** entity. Each instance of **Staff** has one and only one office only. An office has at least one staff, but can be many staff. The relation is already in 3NF since each table has a single value, there are no partial dependencies, and there are no transitive dependencies.

Office(officeNo, oStreet, oCity, oPostcode, oState, oCountry, oPhoneNo) has a primary key of officeNo. Each instance of **Office** has one or more staff, so officeNo is a foreign key in **Staff.** The relation is already in 3NF since each table has a single value, there are no partial dependencies, and there are no transitive dependencies.

Customer(customerID, cFirstName, cLastName, cStreet, cCity, cPostcode, cState, cCountry, cPhone, cEmail, staffID) has a primary key of customerID. Each instance of **Customer** has exactly one staff servicing them, so staffID is a foreign key referencing the **Staff** entity. The relation is already in 3NF since each attribute instance has a single value, there are no partial dependencies, and there are no transitive dependencies.

TravelPackage(packageName, destination, duration, maxCapacity, staffID) has a primary key of packageName. Each instance of **TravelPackage** has exactly one member of staff marketing the package, so staffID is a foreign key referencing the member of staff marketing the package. The relation is already in 3NF since each attribute instance has a single value, there are no partial dependencies, and there are no transitive dependencies.

TripAvailability(tripID, tripDate, pricePP, packageName, staffID) has a primary key of tripID. Each instance of **TripAvailability** has exactly one staff member guiding it, so staffID is a foreign key referencing a member of staff in the **Staff** entity. The relation is already in 3NF since each attribute instance has a single value, there are no partial dependencies, and there are no transitive dependencies.

Booking(transactionID, dateBooked, noPeopleBooked, dueDate, totalCharge, tripID, customerID) has a primary key of transactionID. Each instance of **Booking** has exactly one customer who purchased it, so customerID is a foreign key referencing **Customer** entity. Each booking instance is associated with one and only one available trip, so tripID is a foreign key referencing the **TripAvailability** entity. The relation is already in 3NF since each attribute instance has a single value, there are no partial dependencies, and there are no transitive dependencies.

Survey(surveyDate, overallSatisfaction, comfort, staff, facilities, value, location, transactionID) has a primary key of transactionID. Each instance of **Survey** is associated with exactly one instance of **Booking**, but an instance of **Booking** does not require an instance of **Survey**. For this reason, the foreign key of transactionID from the **Booking** entity is also the primary key in **Survey**. The relation is already in 3NF since each attribute instance has a single value, there are no partial dependencies, and there are no transitive dependencies.

Use Case Implementation

Use Case 1: A marketer wants to add a new travel package for Italy into the database.

```
INSERT INTO TravelPackages VALUES('ITALY4', 'Italy', 14, 30, 423125);
```

Use Case 2: A member of staff wants to delete a travel package from the database as the company will no longer offer it.

```
DELETE FROM TravelPackage
WHERE packageName = 'SWEDEN3';
```

Use Case 3: A member of staff wants to update a travel package's max capacity.

```
UPDATE TravelPackage
SET maxCapacity = 35
```

WHERE packageName = ‘PORTUGAL1’;

Use Case 4: A member of staff wants to view how many staff members are marketing travel packages for each country.

SELECT destination, COUNT (DISTINCT staffID)

FROM TravelPackage

GROUP BY destination;

Use Case 5: A new customer has signed up and needs to be added to the database.

INSERT INTO Customer VALUES('1SC3CGINF3F', 'Paige', 'Gonzales', '3833 Dunlavy St Apt 507', 'Houston', '77006', 'TX', 'USA', '14097954207', 'paige@paige.com', 773884);

Use Case 6: A member of staff wants to remove a customer from the database.

DELETE FROM Customer

WHERE customerID = ‘0ETGRWY6JB’;

Use Case 7: A customer has changed their phone number since booking the trip and wants to update their contact information.

UPDATE Customer

SET cPhone = 13334445555

WHERE customerID = '1SC3CFME2E';

Use Case 8: A member of staff wants to know how many customers live in a certain state.

SELECT COUNT(DISTINCT customerID) AS Total

FROM Customer

WHERE cState = 'CA';

Use Case 9: A member of staff has been hired and needs to be added to the database.

INSERT INTO Staff VALUES('239119', 'Paige', 'Gonzales', '1992-06-01', 'PaigeGonzales@worldwanderers.net', 'Guide', 9);

Use Case 10: A long-time staff member has retired and needs to be removed from the database.

```
DELETE FROM Staff

WHERE staffID = '239119';
```

Use Case 11: A member of staff has gotten married and needs to change their last name.

```
UPDATE Staff

SET sLastName = 'Martin'

WHERE staffID = '155227';
```

Use Case 12: A member of staff wants to see the number of staff for each primary role: Marketer, Admin, and Guide

```
SELECT primaryRole, COUNT(DISTINCT staffID) AS Total

FROM Staff

GROUP BY primaryRole;
```

Use Case 13: A new office has opened and needs to be added to the database.

```
INSERT INTO Office VALUES(11, '123 Steven Lane', 'Madrid', '567891', 'Madrid', 'Spain', '8888888888');
```

Use Case 14: An office has closed and needs to be removed from the database.

```
DELETE FROM Office

WHERE officeNo= '11';
```

Use Case 15: An existing office has relocated and needs to update the address.

```
UPDATE Office

SET oStreet = '123 Steven Lane'

WHERE officeNo = '9';
```

Use Case 16: A member of staff wants to see the number of offices located in each country.

```
SELECT oCountry, COUNT(DISTINCT officeNo) AS Total

FROM Office

GROUP BY oCountry;
```

Use Case 17: A new trip is becoming available next summer and needs to be added to the database.

```
INSERT INTO TripAvailability VALUES('16ECA9TGP', '2024-08-04', 2400, 'ITALY2', '231728');
```

Use Case 18: A trip has been cancelled and needs to be removed from the database.

```
DELETE FROM TripAvailability

WHERE tripID= '05DB8Z8SHO';
```

Use Case 19: A price has changed for a trip and needs to be updated.

```
UPDATE TripAvailability

SET pricePP= 3500

WHERE tripID = '07RSGQBYZ1';
```

Use Case 20: A member of staff wants to see the average price per person of a trip in August 2023.

```
SELECT AVG(pricePP) AS Average

FROM TripAvailability

WHERE tripDATE LIKE '%2023-08%';
```

Use Case 21 A new booking has been made and needs to be added to the database.

```
INSERT INTO Booking VALUES(' 0458SYCDHOSYW9RK8W2R', '2024-07-15', 3, '2023-09-18' , 8898, '2HZT6YIMTG',  'TX6Z7EHPTQ');
```

Use Case 22: A booking has been cancelled and needs to be removed from the database.

```
DELETE FROM Booking

WHERE transactionID = '0458SYCDHOSYW9RK8W2R ';
```

Use Case 23: A customer wants to change the number of people they have booked for a trip.

```
UPDATE Booking

SET noPeopleBooked = 3
```

WHERE transactionID = '0KSX2C8O8CYHZPODXR6H';

Use Case 24: A member of staff wants to see what the booking charges are for bookings with a due date of May 2023.

SELECT SUM(totalCharge) AS Total

FROM Booking

WHERE dueDATE LIKE '%2023-05%';

Use Case 25: A customer has returned from a trip and has been prompted to take a customer satisfaction survey.

INSERT INTO Survey VALUES('2022-03-16', 3, 4, 10, 6, 6, 3, '0ZGZOZOO8BEO1G08VE5D');

Use Case 26: A survey was prematurely submitted by a customer and needs to be deleted.

DELETE FROM Survey

WHERE Booking_transactionID = '0ZGZOZOO8BEO1G08VE5D';

Use Case 27: A customer wants to change their survey responses.

Input data:

UPDATE Survey

SET overallSatisfaction = 6

WHERE Booking_transactionID = '12GUKPU9X5JB2EZ9IFS3';

Use Case 28: A member of staff wants to see what the average overall satisfaction rating is for surveys taken in the year 2022.

SELECT AVG(overallSatisfaction) AS AverageRating

FROM Survey

WHERE surveyDATE LIKE '%2022%';

Use Case 29: A member of staff wants to display the trip details and travel package details of trips being offered that have a duration of less than 14 days.

SELECT t*

FROM TravelPackage p, TripAvailability t

WHERE p.packageName = t.packageName AND duration < 14

SORT BY duration;

Use Case 30: A member of staff wants to see the customer details of customers who are assigned to administrative staff.

SELECT c.*, s.primaryRole

FROM Customer c, Staff s

WHERE c.Staff_staffID = s.staffID AND primaryRole = ‘Admin’;

Use Case 31: A staff member wants to view the travel package details of packages that are marketed by staff in office number 9.

SELECT p.*, s.Office_officeNo

FROM travelPackage p, Staff s

WHERE p.Staff_staffID = s.staffID AND Office_officeNo = 9;

Use Case 32: A member of staff wants to view the staff details of the staff members who work at a branch in Italy.

SELECT s.*, o.oCountry

FROM Staff s, Office o

WHERE s.Office_officeNo = o.officeNo AND oCountry = 'Italy';

Use Case 33: A member of staff wants to display all trip availability details for bookings that have been made in July 2022.

SELECT t.*, b.dateBooked

FROM TripAvailability t, Booking b

WHERE b.TripAvailability_tripID = t.tripID AND dateBooked LIKE '%2022-07%';

Use Case 34: A member of staff wants to display all customer details for bookings that have been made in December 2022.

```
SELECT c.*, b.dateBooked

FROM Customer c, Booking b

WHERE c.customerID = b.Customer_customerID AND dateBooked LIKE '%2022-12%';
```

Use Case 35: A member of staff wants to display all booking details for surveys that have an overall satisfaction rate of less than 5.

```
SELECT b.*, s.overallSatisfaction

FROM Booking b, Survey s

WHERE b.transactionID = s.Booking_transactionID AND overallSatisfaction < 5;
```

Use Case 36: A member of staff wants to display all staff details for staff that have been assigned a trip in September 2023.

```
SELECT s.*, t.tripDate

FROM tripAvailability t, Staff s

WHERE t.Staff_staffId = s.staffID AND tripDate LIKE '%2023-09%';
```

Testing

Use Case 1: A marketer wants to add a new travel package for Italy into the database.

Input data:

```
INSERT INTO worldwanderers.TravelPackages VALUES('ITALY4', 'Italy', 14, 30, 423125);
```

Expected output: A new record will be added to the TravelPackages table with the input values of packageName = ITALY4, destination = Italy, duration = 14, maxCapacity = 30, staffID = 423125.

Actual output: The query functioned as expected.

packageName	destination	duration	maxCapacity	Staff_staffID
ITALY4	Italy	14	30	423125

Use Case 2: A member of staff wants to delete a travel package from the database as the company will no longer offer it.

Input data:

DELETE FROM worldwanderers.TravelPackage

WHERE packageName = 'SWEDEN3';

Expected output: Any travel package named ‘SWEDEN3’ should be removed from the database along with its associated rows in child tables.

Actual output: The query functioned as expected.

Before:

packageName	destination	duration	maxCapacity	Staff_staffID
SWEDEN1	Sweden	5	35	303111
SWEDEN2	Sweden	10	35	303111
SWEDEN3	Sweden	15	35	303111
NULL	NULL	NULL	NULL	NULL

After:

packageName	destination	duration	maxCapacity	Staff_staffID
SWEDEN1	Sweden	5	35	303111
SWEDEN2	Sweden	10	35	303111

Use Case 3: A member of staff wants to update a travel package’s max capacity.

Input data:

UPDATE worldwanderers.TravelPackage

SET maxCapacity = 35

WHERE packageName = ‘PORTUGAL1’;

Expected output: The PORTUGAL1 package should show a max capacity of 35.

Actual output: The query functioned as expected.

packageName	destination	duration	maxCapacity	Staff_staffID
PORTUGAL1	Portugal	5	35	813679

Use Case 4: A member of staff wants to view how many staff members are marketing travel packages for each country.

Input data:

SELECT destination, COUNT (DISTINCT Staff_staffID)

FROM worldwanderers.TravelPackage

GROUP BY destination;

Expected output: The output should display a single value representing the number of staff members marketing travel packages for each country.

Actual output: The query functioned as expected.

destination	StaffNo
Austria	1
Belgium	1
France	2
Germany	1
Italy	2
Netherlands	1
Portugal	1
Spain	1
Sweden	1
Switzerland	1

Use Case 5: A new customer has signed up to World Wanderers and needs to be added to the database.

Input data:

INSERT INTO worldwanderers.Customer VALUES('1SC3CGINF3F', 'Paige', 'Gonzales', '3833 Dunlavy St Apt 507', 'Houston', '77006', 'TX', 'USA', '14097954207', 'paige@paige.com', 773884);

Expected output: A new row of the Customer table should be added with all the information about the customer where customerID =1SC3CGINF3F, cfirstName = Paige, cLastName = Gonzales, cStree = 3833 Dunlavy St Apt 507, cCity = Houston, cPostCode = 77006, cState = TX, cCountry = USA, cPhone = 14097954207, cEmail = paige@paige.com, staffID = 773884.

Actual output: The query functioned as expected.

customerID	cfirstName	cLastName	cStreet	cCity	cPostCode	cState	cCountry	cPhone	cEmail
1SC3CGINF3F	Paige	Gonzales	3833 Dunlavy St Apt 507	Houston	77006	TX	USA	14097954207	paige@p

Use Case 6: A member of staff wants to remove a customer from the database.

Input data:

DELETE FROM worldwanderers.Customer

WHERE customerID = ‘0ETGRWY6JB’;

Expected output: The customer with ID 0ETGRWY6JB will be removed from the database along with any associated child tables.

Actual output: The query functioned as expected.

Before:

customerID	cfirstName	cLastName	cStreet	cCity	cPostCode	cState	cCountry	cPhone	cEmail
0ETGRWY6JB	Alanna	Lozano	5378 Allison Street	Arvada	80002	CO	USA	17167758403	AlannaLoz

After:

customerID	cfirstName	cLastName	cStreet	cCity	cPostCode	cState	cCountry	cPhone	cEmail	Staff_stat
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Use Case 7: A customer has changed their phone number since booking the trip and wants to update their contact information.

Input data:

UPDATE worldwanderers.Customer

SET cPhone = 13334445555

WHERE customerID = '1SC3CFME2E';

Expected output: The customer with ID 1SC3CFME2E should have their phone number changed to 13334445555.

Actual output: The query functioned as expected.

customerID	cfirstName	cLastName	cStreet	cCity	cPostCode	cState	cCountry	cPhone
1SC3CFME2E	Frederic	May	1710 Massachusetts Avenue Southeast	Washington	20003	AR	USA	13334445555

Use Case 8: A member of staff wants to know how many customers live in a certain state.

Input data:

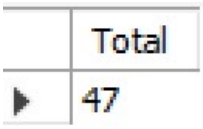
SELECT COUNT(DISTINCT customerID) AS Total

FROM worldwanderers.Customer

WHERE cState = 'CA';

Expected output: The output should show one value, which is the number of customers living in California.

Actual output: The query functioned as expected.



	Total
▶	47

Use Case 9: A member of staff has been hired and needs to be added to the database.

Input data: INSERT INTO worldwanderers.Staff VALUES('239119', 'Paige', 'Gonzales', '1992-06-01', 'PaigeGonzales@worldwanderers.net', 'Guide', 9);

Expected output: A new row of the Staff table should be added with all the information about the new staff member including staffID = 239119, sFirstName = Paige, sLastName = Gonzales, sDateOfBirth = 1992-06-01, sEmail = PaigeGonzales@worldwanderers.net, primaryRole= Guide, officeNo = 9.

Actual output: The query functioned as expected.

	staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo
▶	239119	Paige	Gonzales	1992-06-01	PaigeGonzales@worldwanderers.net	Guide	9

Use Case 10: A long-time staff member has retired and needs to be removed from the database.

Input data:

DELETE FROM worldwanderers.Staff

WHERE staffID = '239119';

Expected output: The staff member with ID 239119 will be removed from the database.

Actual output: The query functioned as expected.

Before:

staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo
239119	Paige	Gonzales	1992-06-01	PaigeGonzales@worldwanderers.net	Guide	9

After:

staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Use Case 11: A member of staff has gotten married and needs to change their last name.

Input data:

```
UPDATE worldwanderers.Staff

SET sLastName = 'Martin'

WHERE staffID = '155227';
```

Expected output: The staff member with staffID 155227 will show their last name as Martin.

Actual output: The query functioned as expected.

staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo
155227	Alice	Martin	1975-02-27	AliceSmith@worldwanderers.net	Admin	9

Use Case 12: A member of staff wants to see the number of staff for each primary role: Marketer, Admin, and Guide

Input data:

```
SELECT primaryRole, COUNT(DISTINCT staffID) AS Total

FROM worldwanderers.Staff

GROUP BY primaryRole;
```

Expected output: The output will show one value for each of the three primary roles: Admin, Guide, and Marketer representing the total number of staff in each role.

Actual output: The query functioned as expected.

primaryRole	Total
Admin	14
Guide	32
Marketer	12

Use Case 13: A new office has opened and needs to be added to the database.

Input data: INSERT INTO worldwanderers.Office VALUES(11, '123 Steven Lane', 'Madrid', '567891', 'Madrid', 'Spain', '8888888888');

Expected output: A new row of the Office table should be added with all the information about the new office location including officeNo = 11, oStreet = 123 Steven Lane, oCity = Madrid, oPostcode = 567891, oState = Madrid, oCountry = Spain, oPhoneNo = 8888888888.

Actual output: The query functioned as expected.

officeNo	oStreet	oCity	oPostcode	oState	oCountry	oPhoneNo
11	123 Steven Lane	Madrid	567891	Madrid	Spain	8888888888

Use Case 14: An office has closed and needs to be removed from the database.

Input data:

DELETE FROM worldwanderers.Office

WHERE officeNo= '11';

Expected output: The office with officeNo of 11 will be removed from the database.

Actual output: The query functioned as expected.

Before:

officeNo	oStreet	oCity	oPostcode	oState	oCountry	oPhoneNo
11	123 Steven Lane	Madrid	567891	Madrid	Spain	8888888888

After:

officeNo	oStreet	oCity	oPostcode	oState	oCountry	oPhoneNo
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Use Case 15: An existing office has relocated and needs to update the address.

Input data:

UPDATE worldwanderers.Office

SET oStreet = '123 Steven Lane'

WHERE officeNo = '9';

Expected output: Office number 9 will have its street address shown as '123 Steven Lane.'

Actual output: The query functioned as expected.

officeNo	oStreet	oCity	oPostcode	oState	oCountry	oPhoneNo
9	123 Steven Lane	Stockholm	11120	Soderma...	Sweden	46168165757

Use Case 16: A member of staff wants to see the number of offices located in each country.

Input data:

```
SELECT oCountry, COUNT(DISTINCT officeNo) AS Total
FROM worldwanderers.Office
GROUP BY oCountry;
```

Expected output: The output will show one value representing the number of office in each of the countries where the offices are located.

Actual output: The query functioned as expected.

oCountry	Total
Austria	1
Belgium	1
France	1
Germany	1
Italy	1
Netherlands	1
Portugal	1
Spain	1
Sweden	1
Switzerland	1

Use Case 17: A new trip is becoming available next summer and needs to be added to the database.

Input data:

```
INSERT INTO worldwanderers.TripAvailability VALUES('16ECA9TGP', '2024-08-04', 2400, 'ITALY2', '231728');
```

Expected output: A new row of the Trip Availability table should be added with all the information about the new trip being offered including tripID = 16ECA9TGP, tripDate = 2024-08-04, 2400, packageName = ITALY2, staffID = 231728.

Actual output: The query functioned as expected.

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID
16ECA9TGP	2024-08-04	2400	ITALY2	231728

Use Case 18: A trip has been cancelled and needs to be removed from the database.

Input data:

```
DELETE FROM worldwanderers.TripAvailability

WHERE tripID= '05DB8Z8SHO';
```

Expected output: The trip with tripID 05DB8Z8SHO will be removed from the database.

Actual output: The query functioned as expected.

Before:

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID
05DB8Z8SHO	2023-08-15	2430	ITALY1	898045

After:

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID
NULL	NULL	NULL	NULL	NULL

Use Case 19: A price has changed for a trip and needs to be updated.

Input data:

```
UPDATE worldwanderers.TripAvailability

SET pricePP= 3500

WHERE tripID = '07RSGQBYZ1';
```

Expected output: The trip with ID 07RSGQBYZ1 will show its pricePP as 3500.

Actual output: The query functioned as expected.

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID
07RSGQBYZ1	2023-11-08	3500	FRANCE3	951437

Use Case 20: A member of staff wants to see the average price per person of a trip in August 2023.

Input data:

SELECT AVG(pricePP) AS Average

FROM worldwanderers.TripAvailability

WHERE tripDATE LIKE '%2023-08%';

Expected output: The output will show one value for the average price of all trips that take place in August 2023.

Actual output: The query functioned as expected.

	Average
▶	2319.6000

Use Case 21: A new booking has been made and needs to be added to the database.

Input data:

INSERT INTO worldwanderers.Booking VALUES(' 0458SYCDHOSYW9RK8W2R', '2024-07-15', 3, '2023-09-18' , 8898, '2HZT6YIMTG', 'TX6Z7EHPTQ');

Expected output: A new row of the Booking table should be added with all the information about the booking including transactionID = 0458SYCDHOSYW9RK8W2R', dateBooked = 2024-07-15, noPeopleBooked = 3, dueDate = 2023-09-18, totalCharge = 8898, tripID = 2HZT6YIMTG, and customerID = TX6Z7EHPTQ.

Actual output: The query functioned as expected.

	transactionID	dateBooked	noPeopleBooked	dueDate	totalCharge	TripAvailability_tripID	Customer_customerID
▶	0458SYCDHOSYW9RK8W2R	2024-07-15	3	2023-09-18	8898	2HZT6YIMTG	TX6Z7EHPTQ

Use Case 22: A booking has been cancelled and needs to be removed from the database.

Input data:

DELETE FROM worldwanderers.Booking

WHERE transactionID = '0458SYCDHOSYW9RK8W2R ';

Expected output: The trip with transactionID 0458SYCDHOSYW9RK8W2R will be removed from the database.

Actual output: The query functioned as expected.

Before:

transactionID	dateBooked	noPeopleBooked	dueDate	totalCharge	TripAvailability_tripID	Customer_customerID
0458SYCDHOSYW9RK8W2R	2024-07-15	3	2023-09-18	8898	2HZT6YIMTG	TX6Z7EHPTQ

After:

transactionID	dateBooked	noPeopleBooked	dueDate	totalCharge	TripAvailability_tripID	Customer_customerID
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Use Case 23: A customer wants to change the number of people they have booked for a trip.

Input data:

```
UPDATE worldwanderers.Booking

SET noPeopleBooked = 3

WHERE transactionID = '0KSX2C8O8CYHZPODXR6H';
```

Expected output: Transaction with ID 0KSX2C8O8CYHZPODXR6H will have the number of people booked changed to 3.

Actual output: The query functioned as expected.

transactionID	dateBooked	noPeopleBooked	dueDate	totalCharge	TripAvailability_tripID	Customer_customerID
0KSX2C8O8CYHZPODXR6H	2021-07-28	3	2023-08-07	15438	7U0D7HC9N8	K8CG3NOIAP

Use Case 24: A member of staff wants to see what the booking charges are for bookings with a due date of May 2023.

Input data:

```
SELECT SUM(totalCharge) AS Total

FROM worldwanderers.Booking

WHERE dueDATE LIKE '%2023-05%';
```

Expected output: The output will show one value for the total charges with a due date in May 2023.

Actual output: The query functioned as expected.

Total
126135

Use Case 25: A customer has returned from a trip and has been prompted to take a customer satisfaction survey.

Input data:

INSERT INTO worldwanderers.Survey VALUES('2022-03-16', 3, 4, 10, 6, 6, 3, '0ZGZOZOO8BEO1G08VE5D');

Expected output: A new row of the Survey table should be added with all the information about the survey associated with transaction 0ZGZOZOO8BEO1G08VE5D.

Actual output: The query functioned as expected.

surveyDate	overallSatisfaction	comfort	staff	facilities	value	location	Booking_transactionID
2022-03-16	3	4	10	6	6	3	0ZGZOZOO8BEO1G08VE5D

Use Case 26: A survey was prematurely submitted by a customer and needs to be deleted.

Input data:

DELETE FROM worldwanderers.Survey

WHERE Booking_transactionID = '0ZGZOZOO8BEO1G08VE5D';

Expected output: The survey with transactionID 0ZGZOZOO8BEO1G08VE5D will be removed from the database.

Actual output:

Before:

surveyDate	overallSatisfaction	comfort	staff	facilities	value	location	Booking_transactionID
2022-03-16	3	4	10	6	6	3	0ZGZOZOO8BEO1G08VE5D

After:

surveyDate	overallSatisfaction	comfort	staff	facilities	value	location	Booking_transactionID
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Use Case 27: A customer wants to change their survey responses.

Input data:

UPDATE worldwanderers.Survey

SET overallSatisfaction = 6

WHERE Booking_transactionID = '12GUKPU9X5JB2EZ9IFS3';

Expected output: Transaction with ID 12GUKPU9X5JB2EZ9IFS3 will have the number of overall satisfaction rating changed to 6.

Actual output: The query functioned as expected.

	surveyDate	overallSatisfaction	comfort	staff	facilities	value	location	Booking_transactionID
▶	2022-10-04	6	7	5	2	3	7	12GUKPU9X5JB2EZ9IFS3

Use Case 28: A member of staff wants to see what the average overall satisfaction rating is for surveys taken in the year 2022.

Input data:

SELECT AVG(overallSatisfaction) AS AverageRating

FROM worldwanderers.Survey

WHERE surveyDATE LIKE '%2022%';

Expected output: The output will show one value for the average overall satisfaction rating for surveys taken in 2022.

Actual output: The query functioned as expected.

	AverageRating
▶	4.8769

Use Case 29: A member of staff wants to display the trip details and travel package details of trips being offered that have a duration of less than 14 days.

Input data:

SELECT t.*, p.*

FROM worldwanderers.TravelPackage p, worldwanderers.TripAvailability t

WHERE p.packageName = t.TravelPackage_packageName AND duration < 14

ORDER BY duration DESC;

Expected output: All trip availability and travel package details for trips less than 14 days in travelPackage and tripAvailability tables should be displayed.

Actual output: The query functioned as expected.

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID	packageName	destination	duration	maxCapacity	Staff_staffID
2OR463WN8L	2023-03-09	3108	NETHERLANDS3	596709	NETHERLANDS3	Netherlands	12	35	261561
JP5L0S8X0N	2022-03-09	3015	NETHERLANDS3	596709	NETHERLANDS3	Netherlands	12	35	261561
M7NGZP5SIT	2023-03-05	3270	NETHERLANDS3	512433	NETHERLANDS3	Netherlands	12	35	261561
NETD0O9MYH	2022-11-17	3564	NETHERLANDS3	668957	NETHERLANDS3	Netherlands	12	35	261561
YU4AMLS8IX	2023-11-17	3674	NETHERLANDS3	668957	NETHERLANDS3	Netherlands	12	35	261561

Use Case 30: A member of staff wants to see the customer details of customers who are assigned to administrative staff.

Input data:

SELECT c.*, s.primaryRole

FROM worldwanderers.Customer c, worldwanderers.Staff s

WHERE c.Staff_staffID = s.staffID AND primaryRole = ‘Admin’;

Expected output: All customer details should display who are assigned to staff that has a primary role of Admin, along with the column primaryRole to verify all are Admin.

Actual output: The query functioned as expected.

	cCity	cPostCode	cState	cCountry	cPhone	cEmail	Staff_staffID	primaryRole
Road	undefined	5647	VT	USA	17276095952	ArjanMacdonald@msn.com	128008	Admin
ue	Worcester	1604	MA	USA	1201709293	GinaLee@aol.com	128008	Admin
Street	Fayetteville	72701	AR	USA	17774122214	AlastairNavarro@aol.com	128008	Admin
o Avenue	Mesa	85208	AZ	USA	1764310827	RyanChen@gmail.com	128008	Admin
Court	Crofton	21114	MD	USA	1924835585	CurtisAllen@msn.com	128008	Admin
Circle	Savannah	31415	GA	USA	17344327376	RayKim@gmail.com	128008	Admin
t	Manchester	6042	CT	USA	15032014491	VioletCobb@msn.com	128008	Admin

Use Case 31: A staff member wants to view the travel package details of packages that are marketed by staff in office number 9.

Input data:

SELECT p.*, s.Office_officeNo

FROM worldwanderers.travelPackage p, worldwanderers.Staff s

WHERE p.Staff_staffID = s.staffID AND Office_officeNo = 9;

Expected output: The output will show all package details for packages that are marketed by members of staff who work in office number 9.

Actual output: The query functioned as expected.

packageName	destination	duration	maxCapacity	Staff_staffID	Office_officeNo
SWEDEN1	Sweden	5	35	303111	9
SWEDEN2	Sweden	10	35	303111	9

Use Case 32: A member of staff wants to view the staff details of the staff members who work at a branch in Italy.

Input data:

```
SELECT s.*, o.oCountry
FROM worldwanderers.Staff s, worldwanderers.Office o
WHERE s.Office_officeNo = o.officeNo AND oCountry = 'Italy';
```

Expected output: The output will show all the staff details for staff members that work at an office in Italy, along with the office country to verify it is Italy.

Actual output: The query functioned as expected.

staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo	oCountry
128008	Kevin	Kim	1976-01-12	KevinKim@worldwanderers.net	Admin	5	Italy
231728	David	Kim	1983-09-12	DavidKim@worldwanderers.net	Guide	5	Italy
361924	David	Kim	1966-06-05	DavidKim@worldwanderers.net	Admin	5	Italy
423125	Rachel	Lee	1957-09-11	RachelLee@worldwanderers.net	Marketer	5	Italy
618551	Mia	Johnson	1992-02-04	MiaJohnson@worldwanderers.net	Guide	5	Italy
673039	Park	Min	1954-05-10	ParkMin@worldwanderers.net	Marketer	5	Italy

Use Case 33: A member of staff wants to display all trip availability details for bookings that have been made in July 2022.

Input data:

```
SELECT t.*, b.dateBooked
FROM worldwanderers.TripAvailability t, worldwanderers.Booking b
WHERE b.TripAvailability_tripID = t.tripID AND dateBooked LIKE '%2022-07%';
```

Expected output: The output will show all the trip availability details for trips that have been booked in July 2022.

Actual output: The query functioned as expected.

tripID	tripDate	pricePP	TravelPackage_packageName	Staff_staffID	dateBooked
OQDN1HVPCT	2023-06-09	3749	SWITZERLAND3	988352	2022-07-30
O5FW1C6QUP	2023-06-11	3369	ITALY3	898045	2022-07-17
FXNTS483WD	2022-12-24	1574	SWITZERLAND1	988352	2022-07-13
VMOI1KZKFA	2022-11-20	1626	ITALY1	618551	2022-07-22
8WD741V70B	2023-02-13	2820	NETHERLANDS2	596709	2022-07-03
RAXDATOUJI	2023-03-21	2490	FRANCE2	622951	2022-07-13
CFJ1YGL24Y	2023-04-16	2484	NETHERLANDS2	512433	2022-07-10

Use Case 34: A member of staff wants to display all customer details for bookings that have been made in December 2022.

Input data:

SELECT c.*, b.dateBooked

FROM worldwanderers.Customer c, worldwanderers.Booking b

WHERE c.customerID = b.Customer_customerID AND dateBooked LIKE '%2022-12%';

Expected output: The output will show all the customer details for customers that made a booking in December 2022 and the dateBooked to confirm they are all in December 2022.

Actual output: The query functioned as expected.

	cCity	cPostCode	cState	cCountry	cPhone	cEmail	Staff_staffID	dateBooked
Juan Avenue	Fremont	94536	CA	USA	15149421435	GiuliaRossi@aol.com	623596	2022-12-28
	NULL	NULL	NULL	NULL	15817475610	DavidLee@yahoo.com	502899	2022-12-10
ood Drive	Anchorage	99502	AK	USA	12144397472	SamiOneal@yahoo.com	502703	2022-12-02
ic Avenue	Berkeley	94709	CA	USA	13635493049	SarahLee@msn.com	623596	2022-12-21
cliff Road	Panama City	32405	FL	USA	166159837	CerysArcher@hotmail.com	128008	2022-12-06
	NULL	NULL	NULL	NULL	17548215279	AliceSmith@outlook.com	361924	2022-12-01
k Drive	Louisville	40214	KY	USA	18519818005	JohnSmith@yahoo.com	816353	2022-12-17
Street	Manchester	6042	CT	USA	15032014491	VioletCobb@msn.com	128008	2022-12-04

Use Case 35: A member of staff wants to display all booking details for surveys that have an overall satisfaction rate of less than 5.

Input data:

SELECT b.*, s.overallSatisfaction

FROM worldwanderers.Booking b, worldwanderers.Survey s

WHERE b.transactionID = s.Booking_transactionID AND overallSatisfaction < 5;

Expected output: The output will show all booking details for surveys that have an overall satisfaction rate less than 5, along with the overall satisfaction rating.

Actual output: The query functioned as expected.

transactionID	dateBooked	noPeopleBooked	dueDate	totalCharge	TripAvailability_tripID	Customer_customerID	overallSatisfaction
1TMKFUIGNESAQU3EQJL8	2021-08-01	4	2022-03-22	4284	OK5QB9QEIX	Z31OZ5GQY4	1
1TP7EAUHISB6XMFGXBWI	2021-03-08	2	2022-06-27	6317	CQHQP769FP	9TU8ZYTWJT	2
1YGIVXPE2UIXENGM8CB1	2022-03-26	4	2023-01-02	11279	8WD741V70B	CP5JT2AVHQ	1
2EQINKNU65SZRALQOI3A	2022-08-31	3	2023-09-07	3416	6MS7SFXP3G	379YUAH0TY	1
32P496GF8M4J28GCFTSP	2021-10-21	3	2022-06-27	9475	CQHQP769FP	TS2XWQ6G0G	4
3ATO4LUU54IIZM2QS8DJ	2022-10-27	3	2022-11-05	5040	2TITJ4NK4F	9T1S36VAHK	2

Use Case 36: A member of staff wants to display all staff details for staff that have been assigned a trip in September 2023.

Input data:

```
SELECT s.*, t.tripDate

FROM worldwanderers.tripAvailability t, worldwanderers.Staff s

WHERE t.Staff_staffId = s.staffID AND tripDate LIKE '%2023-09%';
```

Expected output: The output will show all the staff details for staff that are assigned to a trip taking place in September 2023.

Actual output: The query functioned as expected.

staffID	sFirstName	sLastName	sDateOfBirth	sEmail	primaryRole	Office_officeNo	tripDate
728161	Lily	Wong	1979-06-19	LilyWong@worldwanderers.net	Guide	9	2023-09-09
743784	James	Wilson	1983-11-16	JamesWilson@worldwanderers.net	Guide	5	2023-09-22
668957	Peter	Brown	1951-12-25	PeterBrown@worldwanderers.net	Guide	4	2023-09-25
618551	Mia	Johnson	1992-02-04	MiaJohnson@worldwanderers.net	Guide	5	2023-09-06

Conclusion

The WorldWanderers database has been created to store data on customers, staff, trip packages, and customer satisfaction using seven entities that have been thoroughly reviewed for any potential data redundancy or update anomalies. An ER diagram has been created to show relationships and multiplicities, a relational model has been created to map relationships, and finally normalization has been performed to show all relations are in 3NF. Lastly, the database has been evaluated using update, insert, delete, and aggregation queries during use case implementation and testing, and all tests return expected results. The WorldWanderers database is ready to be put into production by the company and fulfills all mission objectives required.

References

Begg, T. C. C. (2019). *Database systems: A practical approach to design, implementation, and management*. Pearson.

Rattanaporn, K. (2023, March 6). *Traveler trip dataset*. Kaggle. Retrieved April 21, 2023, from https://www.kaggle.com/datasets/rkiattisak/traveler-trip-data

Random US street address - fake address generator. Random Lists. (n.d.). Retrieved April 21, 2023, from https://www.randomlists.com/random-addresses