INFOST 410-201
Final Project Requirements

Contents

Wilma's Wild Wisconsin
Project Description

Background

The Organization
Wilma's Wild Wisconsin (WWW) is an adventure tour company owned and operated by Wilma Carson. Wilma started the business in 2010 based on her family's love of kayaking and canoeing and Wilma's dream of starting her own business. Wilma's staff includes her husband, Bud, and a collection of people who share Wilma's love for kayaking, canoeing, and leading others as they learn and perfect their abilities in these sports.

While Wilma has dreams of significant business expansion, the operations of WWW are currently limited to one-day river trips within Wisconsin during the summer season. Current trip destinations include the Upper Wisconsin River, the Lower Wisconsin River, and the Wolf River. Trips are currently scheduled for only one weekend day (Saturday) allowing the possibility to postpone these trips by one day (to Sunday) when required by unfavorable weather conditions.

WWW Operations
Wilma runs WWW from her home in Milwaukee. She takes care of reservations, cancellations, payment, confirmations, employee scheduling, payroll, and other administrative duties. Wilma is a qualified guide and acts as a substitute when regular guides are sick or otherwise unavailable. In these instances, the WWW office is staffed by one of the workers who is qualified to take reservations.

Guides must be qualified to lead the type of trip to which they are assigned (kayak, or canoe). Wilma is planning to add rafting trips next year, so she would like the system to support rafting as a trip type and as a guiding expertise.

When trips are particularly large, or when the preponderance of guests are inexperienced, WWW assigns one of the staff to join the trip as a Gear Assistant. This also depends on the level of experience of the guide assigned to the trip.

A typical trip starts when the guide (and possibly a gear assistant) meet the guests at the Guest Gathering Point. This is a parking facility near the take-out point for the trip. Guests are expected to arrive by the agreed Latest Guest Arrival Time. Then, the party boards a locally hired bus for transport to the put-in point. Equipment and box lunches are provided to the group by third party contractors at the put-in point.

Upon arrival at the put-in point, the guide provides basic training for any inexperienced guests, and then conducts an enjoyable trip. The group makes several stops during the day (including a lunch stop). When the group arrives at the take-out point, the trip is over. If the Guest Gathering Point is not located at the take-out point, the group is bused back to the Guest Gathering Point so that they may pick up their cars. Equipment handling at the take-out point is also handled by a third-party contractor.

WWW Information Technology

Wilma uses Microsoft Excel spreadsheets, a laptop, and a printer to meet the information technology needs of the business. This approach works fairly well because Wilma is a former bookkeeper and is particularly talented at keeping all of the duplicate information in the spreadsheets synchronized. Nevertheless, she does make the occasional embarrassing mistake.

Wilma is planning significant growth for WWW starting next summer. Ideas under consideration include a longer season (to include spring), rafting trips, trips on weekdays, and more river destinations in Wisconsin. Most important, Wilma plans to turn over the record keeping and reservations work to office staff so that Wilma can get back to guiding (her first love) on a more full-time basis. When these changes take place in the business, Wilma expects that the current information systems approach will result in many mistakes. She is concerned that these mistakes will lead to lost customers, lower employee morale, and bad financial performance.

New WWW Information Technology Initiatives

Wilma was pleased to learn that the University of Wisconsin – Milwaukee, School of Information Studies (UWM/SOIS) works with area businesses to solve business information needs while training information technology professionals for the future. During the past several months, Wilma has been working with a team of systems analysis students from UWM/SOIS to document the requirements for a new systems solution. That has led to the current project to which you have been assigned.

The Approach

The decision has been made that the reports created by the spreadsheet-based system that Wilma has in place represent a good starting point for system capabilities. The weaknesses of the system are that it is limited to a single user at a time, and that the integrity of the data depends upon Wilma's extraordinary bookkeeping skill. A decision has been made to create a multi-user, Web-based system in which the data are stored and maintained in a MySQL database. Based upon Wilma's existing reports, Wilma and the systems analysis team from UWM/SOIS have created a conceptual data model (see Appendix B). The next steps are to create a physical database schema, create a physical sample database, populate the sample database, and write SQL scripts that can be used to simulate the reports and transactions that are currently implemented by Wilma's Excel-based system. The result of these efforts will be considered a completed proof of concept project. It is anticipated that this project will be followed later by a Web Development project in which the rest of the system components are designed and built.

Your Assignment

Your assignment is to use the samples of existing spreadsheet-based reports (see Appendix A) and the completed conceptual data model (see Appendix B) as input to a project where you:

1. Design the physical MySQL database/schema including:
    a. Table definitions
    b. Column definitions that enforce appropriate data types
    c. Foreign keys that implement table-to-table relationships
    d. Database constraints that enforce the referential integrity of foreign keys
    e. Database constraints that enforce uniqueness in columns other than the primary key.
    f. Indexes that facilitate the efficient querying of the data tables based upon their primary key and any other identified secondary keys.

    I am expecting you to implement the conceptual data model as directly as possible. This means that I expect the tables in your database to correspond very closely to those in the conceptual data model.

Note that you may use the MySQL Workbench tool to create the database/schema definitions directly. Or, you may prefer to create these definitions in the more traditional way by coding SQL Data Definition Language (DDL) statements explicitly and executing them via scripts. **The schema name that you use for the database must be "www".**

2. Populate the sample database based upon the values found in the copies of the current spreadsheet-based reports created by Wilma (see Appendix B). Note that Wilma's reports contain a fairly large number of planned trips with a substantial number of registered guests. This will require you to create more sample guests than are shown on the one Trip Roster report that is provided. Feel free to create any number of pretend guests (with names like Guest1, Guest2, etc.) and to reuse them for multiple trips in order to make the reservation counts correct on the query results. As you will see from the conceptual data model, the relationships between trips and guests is many-to-many, so you can re-use your pretend guests from trip to trip in order to create the proper guest counts on the Booking Summary query. Just remember to use each guest no more than once per trip. This restriction will be enforced by the application software that will be created in a future project. Nevertheless, we want your sample data to conform to this rule.

   Note that **any fields that are shown as blank on the sample reports should be represented with a NULL value in the database**. This means that you need to allow for NULL values in the relevant database columns when you are defining the columns in your tables.

3. Use MySQL Workbench to dump your sample database, creating (as a result of this dump) a script for recreating the sample database in its pristine state. You will need this to submit your work and for your testing.

4. Create and test the series of scripts described below in the *Report Query Scripts* section. Run these scripts to test their correctness

and to test the correctness of the sample database contents that you created in an earlier step. **Please remember to pay particular attention to the number of rows expected and the order in which rows should be sorted in the results set**.

5. Create and test the series of scripts described below in the *Database Maintenance Transaction Scripts* section. Use the *Report Queries* created above and the general features of MySQL Workbench to test your work.

6. Submit your project following the directions listed below in *Submitting Your Project*. Make sure that you follow the directions for naming each script file, naming the scripts directory, and naming the .zip file with which you submit your files.

Create Database Script

Create Database (`create_www_database.sql`)
You will not be expected to submit individual scripts to create the database, its tables, constraints, indexes, etc. You may create the database, tables, constraints, indexes etc. using the MySQL Workbench graphical user interface. Whichever approach you choose, this Create Database script will be the script that we run to create your database before we test the work that you submit.

You will also not be expected to populate the sample database using individual scripts that insert data values. You may populate the tables using the MySQL Workbench graphical user interface. Whichever approach you choose, this Create Database script should include all of the INSERT statements that populate the database with sample data.

You can create this script using the MySQL utility program as described in Chapter 19 of the Murach text. Alternatively, you can use the **Server > Data Export** menu option of the MySQL Workbench tool to create this script.

This script should create a fresh copy of the www database/schema on the MySQL server. It should drop the www database and re-create it. The script should drop all existing tables and recreate them. This will allow the script to be used to create the original copy of the database. It also will be useful for resetting the database to its original state during testing. The script should insert sample data into each of the tables.

Make sure to test this script well before submitting your work. When we run your script, we may or may not have already dropped the www database before running your script. So, this script must run successfully whether there is a pre-existing www database or not.

Report Query Scripts

Note: Please pay close attention to the requirements listed below for each script. While these scripts produce result sets that are similar to the previously existing WWW spreadsheets, they are different in a number of details.

Phone List (`phone_list.sql`)
(In order of Employee Nickname. One row per employee.)
Employee Id
Employee Nickname
Employee Name
Mobile Phone
Home Phone

Guide Roles List (`guide_roles_list.sql`)
(In order of Employee Nickname, Role Name. One row per employee, per role.)
Employee Nickname
Employee Name
Role Name

Employee Availability List (`employee_availability_list.sql`)
(In order of Employee Nickname. One row per employee.)
Employee Nickname
Employee Name
Availability Notes

Booking Summary (`booking_summary.sql`)
(In order of Destination Name, Trip Type Name, Trip Date, Trip Number. One row per trip.)
Destination Name
Trip Type Name
Trip Number
Trip Date
Guide Full Name

Trip Guest Capacity
Trip Guests Booked (Note: This field must be calculated based upon current reservation count)
Trip Positions Available (Note: This field must be calculated based upon current reservation count)

Trip Roster (`trip_roster.sql`)
(In order of Destination Name, Trip Type Name, Trip Date, Guest Full Name. One row per trip per guest.)
Destination Name
Trip Type Name
Trip Number
Trip Date
Guide Full Name (format is last name, first name)
Guest Full Name (format is last name, first name)
Guest Experience Name
Guest Age
Guest Weight
Guest Is Swimmer
Guest Mobile Phone

Trip Detail Sheet (`trip_detail_sheet.sql`)
(In order of Destination Name, Trip Type Name, Trip Date. One row per trip.)
Destination Name
Trip Type Name
Trip Number
Trip Date
Guide Full Name
Guide Mobile Phone
Wilma's Wild Wisconsin Office Phone *(414-555-1212)*
Latest Guest Arrival Time
Trip Departure Time
Estimated Trip Return Time
Guest Gathering Point Description

Database Maintenance Transaction Scripts

Some Notes Regarding Database Maintenance Transactions
A properly defined database implements relations between tables with foreign keys. Foreign key relationships are implemented by foreign key constraints. Foreign key constraints assure that no transaction will leave the database in a state of damaged integrity. This means that when you create scripts that INSERT rows, you may need to insert rows into related tables before you can insert the intended row into the target table. Likewise, DELETE scripts can leave the database in a state of damaged integrity unless related rows in other tables are deleted or updated. In these cases, we are expecting each of your scripts to issue more than one SQL statement in order to achieve the job at hand while maintaining relational database integrity. When creating a multi-statement script, remember to end each SQL statement with a semicolon.

When coding DELETE and UPDATE statements, you may identify the row that you want to DELETE or UPDATE using a primary key value. When you use these hard coded key values, your queries will only work properly if run in a particular order. When we test your project, we will be running the queries in the same order that they appear in this document. So, make sure that your queries work when run in this order:

1. Create Database
2. Phone List
3. Guide Roles List
4. Employee Availability List
5. Booking Summary
6. Trip Roster
7. Trip Detail Sheet
8. Add A New Reservation
9. Transfer an Existing Reservation to a Different Trip
10. Delete a Reservation
11. Add A New Employee
12. Delete an Existing Employee
13. Add A New Trip
14. Delete an Existing Trip

Add A New Reservation (`add_a_new_reservation.sql`)

This script will both create a new guest and it will add a new reservation for that guest on existing trip #562. Don't worry about the new reservation putting the trip over its capacity limit. The following are the details of the new guest:

Name	Lilly Ludsen
Experience	Intermediate
Age	25
Weight	120
Is-Swimmer	Yes
Mobile Phone	(none available)

Transfer an Existing Reservation to a Different Trip (`transfer_a_reservation.sql`)

This script will delete the reservation for Lamar Lincoln from trip #562, and it will add a new reservation for Lamar Lincoln to trip #564. As an alternative, you may accomplish the same end with an update transaction.

Delete a Reservation (`delete_a_reservation.sql`)

This script will delete the reservation of Bart Samuels, Jr. on trip #562.

Add A New Employee (`add_a_new_employee.sql`)

This script will insert rows into the tables of the www database in order to create a new guide employee with the following characteristics:

Employee Name	Patrick L. Patterson
Employee Nickname	Pat
Mobile Phone	847-555-9706
Home Phone	(none available)
Availability Notes	All Saturdays and Sundays in June, July, and August
Roles	Guide, Gear
Guide Qualifications	Kayak, Canoe, Raft

Delete an Existing Employee
(`delete_an_existing_employee.sql`)
This script should delete the employee Summer Simms.

Add A New Trip (`add_a_new_trip.sql`)
This script will add a new trip into the database with the following particulars:

Destination Name	Upper Wisconsin River
Trip Type Name	Kayak
Trip Number	(any unused number)
Trip Date	8/2/2014
Guide	Pat Patterson

Delete an Existing Trip (`delete_an_existing_trip.sql`)
This script will cancel trip #576.

Coding and Testing Strategy

Having created the www database/schema and populated it using the MySQL workbench tool, you will then be ready to test each of the transactions and the queries detailed above.   An approach like the following should serve you well:

1. Construct the **`create_www_database.sql`** script so that a fresh copy of the database may be created at any time by running that script.
2. Run the **`create_www_database.sql`** script to refresh the database.
3. Write each query script and test it against the fresh copy of the database. Compare your output to the spreadsheet samples provided to make sure that your queries a generating the expected output.
4. When the query scripts are all written and tested, then begin writing the maintenance transaction scripts.

5. After writing each transaction script, run the query scripts to confirm that the proper changes have taken place in the database.
6. Whenever you reach an unrecoverable state in the database, re-run the **`create_www_database.sql`** script to refresh the database.
7. Continue writing transaction scripts and running report queries until you are satisfied that you have identified and corrected as many errors as you possibly can.
8. Remember to include a USE statement at the beginning of each of your queries: USE www;
9. Remember to run a final test of all of your SQL scripts in the order provided in the section above to make sure that you get expected results when running the same test that we will run on your submitted work.
10. When running your test, it is acceptable to set MySQL Workbench to *Safe Mode* in order to allow your scripts to run properly.

Submitting Your Project

Collect all of your **`.sql`** scripts in a directory named according to the following scheme:

```
yourLastName_yourFirstName_final_project
```

Use a zip utility to create one zip file that contain the contents of this directory. The zip file should be named according to the following scheme:

```
yourLastName_yourFirstName_final_project.zip
```

Submit this one file to the assignment activity made available for this purpose on our Moodle site.

APPENDIX A
Wilma's Existing Spreadsheet-Based Reports

Wilma's Wild Wisconsin
Phone and Availability

Employee Name	Employee Nickname	Roles	Guide Qualifications	Mobile Phone	Home Phone	Availability
Wilma C. Carson	Wilma	Reservations, Guide, Gear	Kayak, Canoe	414-555-1234	414-555-9876	Year round, full-time
Walter C. Carson	Bud	Guide, Gear	Kayak, Canoe	414-555-1235	414-555-9876	Year round, full-time, except for deer hunting season
Anne B. Schultz	Annie	Reservations, Guide, Gear	Kayak	414-555-1266	414-555-2121	Year round, full-time, except for August vacation
Joshua Green	Josh	Guide, Gear	Canoe	414-555-1268		Summers, full-time
William R. Wagner, III	Billy	Gear		414-555-1271	414-555-3259	Summers, full-time, returns to school on August 19th
Constance Simms	Summer	Reservations, Gear		414-555-1288		Year round, part-time, as needed
Blair Q. McArthur	Blinky	Guide, Gear	Canoe	414-555-1289	414-555-2121	Year round, full-time, except for May vacation

Wilma's Wild Wisconsin
Booking Summary

Trip Destination	Trip Type	Trip Number	Trip Date	Guide	Capacity	Booked	Available
Lower Wisconsin River	Canoe	561	8/2/2014	Josh	10	10	0
Lower Wisconsin River	Kayak	562	8/2/2014	Annie	10	9	1
Lower Wisconsin River	Canoe	563	8/9/2014	Josh	10	8	2
Lower Wisconsin River	Kayak	564	8/9/2014	Annie	10	6	4
Lower Wisconsin River	Canoe	565	8/16/2014	Josh	10	6	4
Lower Wisconsin River	Kayak	566	8/16/2014	Annie	10	4	6
Lower Wisconsin River	Canoe	567	8/23/2014	Josh	10	2	8
Lower Wisconsin River	Kayak	568	8/23/2014	Annie	10	0	10
Lower Wisconsin River	Canoe	569	8/30/2014	Josh	10	4	6
Lower Wisconsin River	Kayak	570	8/30/2014	Annie	10	1	9
Upper Wisconsin River	Kayak	571	8/2/2014	Bud	7	7	0
Upper Wisconsin River	Kayak	572	8/9/2014	Bud	7	7	0
Upper Wisconsin River	Kayak	573	8/16/2014	Bud	7	3	4
Upper Wisconsin River	Kayak	574	8/23/2014	Bud	7	2	5
Upper Wisconsin River	Kayak	575	8/30/2014	Bud	7	2	5
Wolf River	Kayak	576	8/2/2014	Wilma	10	8	2
Wolf River	Canoe	577	8/2/2014	Blinky	10	10	0
Wolf River	Kayak	578	8/9/2014	Wilma	10	6	4
Wolf River	Canoe	579	8/9/2014	Blinky	10	9	1
Wolf River	Kayak	580	8/16/2014	Wilma	10	2	8
Wolf River	Canoe	581	8/16/2014	Blinky	10	5	5
Wolf River	Kayak	582	8/23/2014	Wilma	10	6	4
Wolf River	Canoe	583	8/23/2014	Blinky	10	7	3
Wolf River	Kayak	584	8/30/2014	Wilma	10	1	9
Wolf River	Canoe	585	8/30/2014	Blinky	10	0	10

Wilma's Wild Wisconsin
Trip Roster

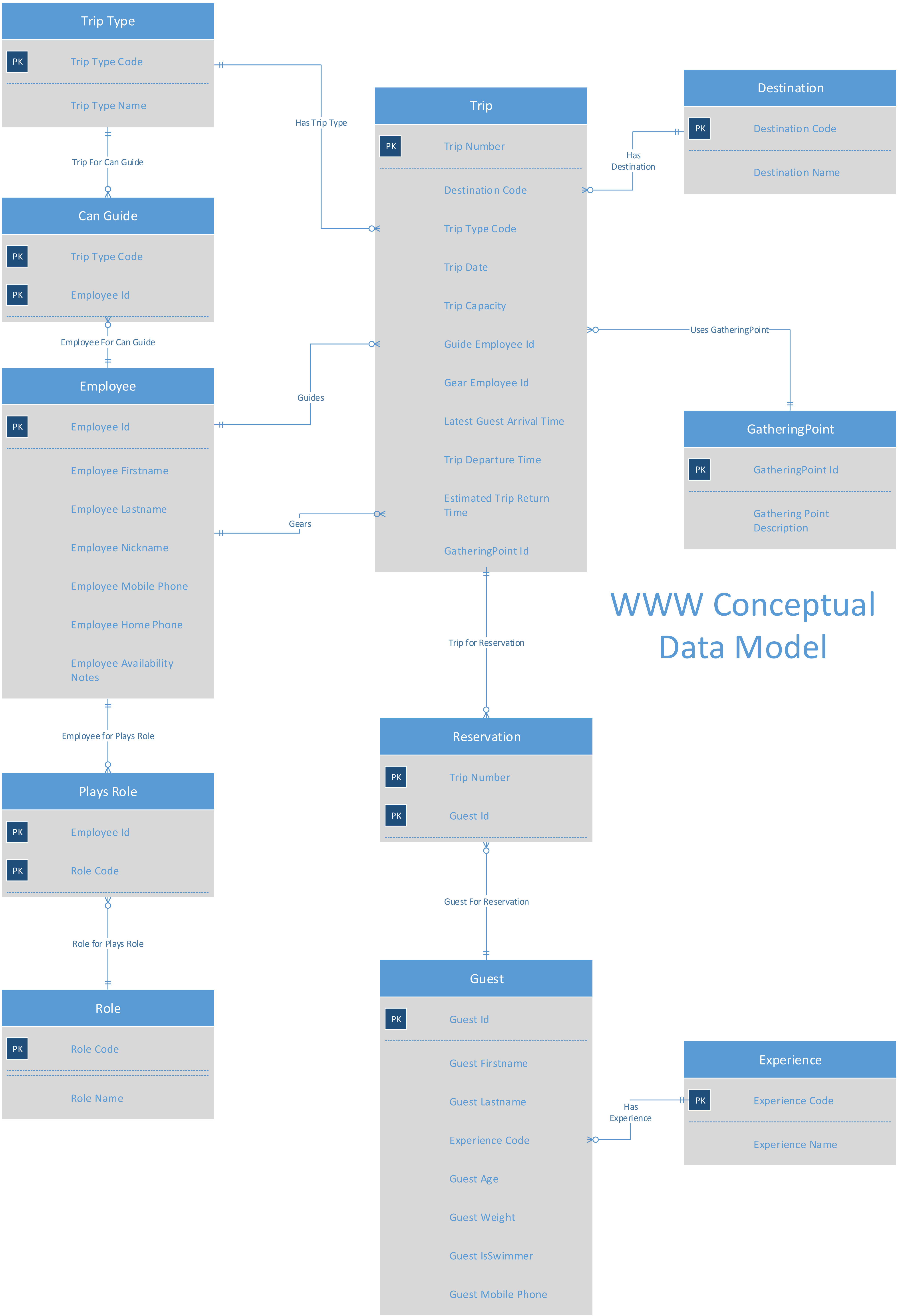
Trip Destination	Trip Type	Trip Number	Trip Date	Guide	Guest Name	Experience	Age	Weight	Swimmer	Mobile Phone
Lower Wisconsin River	Kayak	562	8/2/2014	Annie	Margie Samuels	Novice	31	130	Yes	847-555-5201
					Bart Samuels	Novice	44	195	Yes	
					Lilly Samuels	Novice	18	101	Yes	
					Bart Samuels, Jr.	None	6	60	No	
					George Fogerty	Expert	28	160	Yes	414-555-7654
					Bruce Simmons	Expert	5	150	Yes	
					Marion Glaston	Novice	61	135	Yes	815-555-4453
					Marie Glaston	Novice	61	135	Yes	
					Lamar Lincoln	None	25	175		414-555-9988

Wilma's Wild Wisconsin
Trip Detail Sheet

Information Category	Important Trip Details
Trip Destination	Lower Wisconsin River
Trip Type	Kayak
Trip Number	562
Trip Date	8/2/2014
Guide	Annie
Guide Mobile Phone	414-555-1266
Gear Assistant	
Wilma's Wild Wisconsin Office Phone	414-555-6666
Latest Guest Arrival Time	7:30 AM
Trip Departure Time	8:15 AM
Estimated Trip Return Time	4:00 PM
Guest Gathering Point	River Valley High School (Parking Lot), 660 Varsity Blvd, Spring Green, WI 53588

Appendix B
Conceptual Data Model

WWW Conceptual Data Model


Trip Type
PK Trip Type Code
Trip Type Name
Trip For Can Guide
Can Guide
PK Trip Type Code
PK Employee Id
Employee For Can Guide
Employee
PK Employee Id
Employee Firstname
Employee Lastname
Employee Nickname
Employee Mobile Phone
Employee Home Phone
Employee Availability Notes
Employee for Plays Role
Plays Role
PK Employee Id
PK Role Code
Role for Plays Role
Role
PK Role Code
Role Name
Has Trip Type
Guides
Gears
Trip
PK Trip Number
Destination Code
Trip Type Code
Trip Date
Trip Capacity
Guide Employee Id
Gear Employee Id
Latest Guest Arrival Time
Trip Departure Time
Estimated Trip Return Time
GatheringPoint Id
Has Destination
Destination
PK Destination Code
Destination Name
Uses GatheringPoint
GatheringPoint
PK GatheringPoint Id
Gathering Point Description
Trip for Reservation
Reservation
PK Trip Number
PK Guest Id
Guest For Reservation
Guest
PK Guest Id
Guest Firstname
Guest Lastname
Experience Code
Guest Age
Guest Weight
Guest IsSwimmer
Guest Mobile Phone
Has Experience
Experience
PK Experience Code
Experience Name