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 raft 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 has been working with a systems analyst to design a more robust system for WWW based upon the MySQL relational database management system. At this point, Wilma and the systems analyst have agreed on the logical database design for the new system. They have also agreed on the specifications for a series of scripts to be written that will implement various system features.

Your Assignment

Your assignment is to use the designs and specification below to create and test the first implementation of the new WWW system. Your work should include the following:

1. Create 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. Database constraints that prevent rows being added to tables with important column values set to NULL.
    g. Indexes that facilitate the efficient querying of the data tables based upon their primary key and any other identified secondary keys.

We are expecting you to implement the logical database design presented in Appendix A. Table names and column names in the database that you create should follow those shown in the logical database design.

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 database, including:
    a. Use the test data values presented in the Appendix B .CSV files to populate the database tables.
    b. You may use the .CSV files provided to populate the tables by any reasonable means. Possible approaches include:
        i. Hand-coding INSERT statements that include these values in a script. Run the scripts to insert the data.
        ii. Entering the test values into the database directly using the MySQL Workbench graphical user interface.
        iii. Using any available MySQL feature to directly import the data from the .CSV files into the tables.
    c. Remember to represent the NULL values shown in the test data as true NULL values.

3. Use MySQL Workbench to dump your populated database, creating (as a result of this dump) a script for recreating the fully populated database in its pristine state. More details are provided below under the heading **`01_create_www_database.sql.`**

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 populated 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 Scripts* section. Use the *Report Queries* created above and the general features of MySQL Workbench to check that your scripts leave the database in the expected state.

6. When all of your scripts are complete, 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.

Database Creation Script

`01_create_www_database.sql`
Before creating this script, be sure to create the database / schema and populate it using the methods described above.

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 populated www database 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 shown in Appendix C, they are different in a number of details. So, please concentrate on meeting the requirements described here.

`02_phone_list.sql`
(In order of nickname. One row per employee.)

- employee_id
- nickname
- employee name *(format is first_name + last_name)*
- mobile_phone
- home_phone

03_guide_roles_list.sql

(In order of nickname, role_name. One row per employee, per role.)

- nickname
- employee name *(format is first_name + last_name)*
- role_name

04_employee_availability_list.sql

(In order of nickname. One row per employee.)

- nickname
- employee name *(format is first_name + last_name)*
- availability_notes

05_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 name *(format is first_name + last_name)*
- capacity
- guests booked *(Note: This field must be calculated based upon current reservation count)*
- positions available *(Note: This field must be calculated based upon current reservation count)*

06_trip_roster.sql

(In order of destination_name, trip_type_name, trip_date, guest_name. One row per trip per guest.)

- destination_name
- trip_type_name
- trip_number
- trip_date
- guide name *(format is first_name + last_name)*
- guest name *(format is first_name + last_name)*
- experience_name
- age
- weight
- IsSwimmer
- guest mobile_phone

07_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 name *(format is first_name + last_name)*
- guide mobile_phone
- Wilma's Wild Wisconsin Office Phone *(always 414-555-1212)*
- latest_guest_arrival_time
- departure_time
- estimated_return_time
- gathering point description

Database Maintenance Scripts

Some Notes Regarding Database Maintenance Scripts
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:

- **01_create_www_database.sql**
- **02_phone_list.sql**
- **03_guide_roles_list.sql**
- **04_employee_availability_list.sql**
- **05_booking_summary.sql**
- **06_trip_roster.sql**
- **07_trip_detail_sheet.sql**
- **08_add_a_new_reservation.sql**
- **09_transfer_a_reservation.sql**

- **`10_delete_a_reservation.sql`**
- **`11_add_a_new_employee.sql`**
- **`12_delete_an_existing_employee.sql`**
- **`13_add_a_new_trip.sql`**
- **`14_delete_an_existing_trip.sql`**

`08_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	True
Mobile Phone	(none available)
Experience	Novice

`09_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.

`10_delete_a_reservation.sql`
This script will delete the reservation of Bart Samuels, Jr. on trip #562.

`11_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

`12_delete_an_existing_employee.sql`
This script should delete the employee Summer Simms.

`13_add_a_new_trip.sql`
This script will add a new trip into the database with the following particulars:

Trip Number	666
Trip Date	8/2/2014
Capacity	7
Latest Guest Arrival Time	7:30 AM
Departure Time	8:15 AM
Estimated Return Time	4:00 PM
Destination Name	Upper Wisconsin River
Trip Type Name	Kayak
Guide Employee	Pat Patterson
Gear Employee	(none assigned)
Gathering Point	Omar's Live Bait and Bridal Salon, 3421 Highway KZ, Casino Springs, WI 54776

`14_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 scripts described above. An approach like the following should serve you well:

1. Construct the **`01_create_www_database.sql`** script so that a fresh copy of the populated database may be created at any time by running that script.
2. Run the **`01_create_www_database.sql`** script to refresh the database.
3. Write each query script and test it against the fresh copy of the database. Check your result sets for correctness. Pay particular attention to the number of rows and the order of rows in each result set.
4. When the query scripts are all written and tested, then begin writing the maintenance scripts.
5. After writing each maintenance 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 **`01_create_www_database.sql`** script to refresh the database.
7. Continue writing maintenance 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:

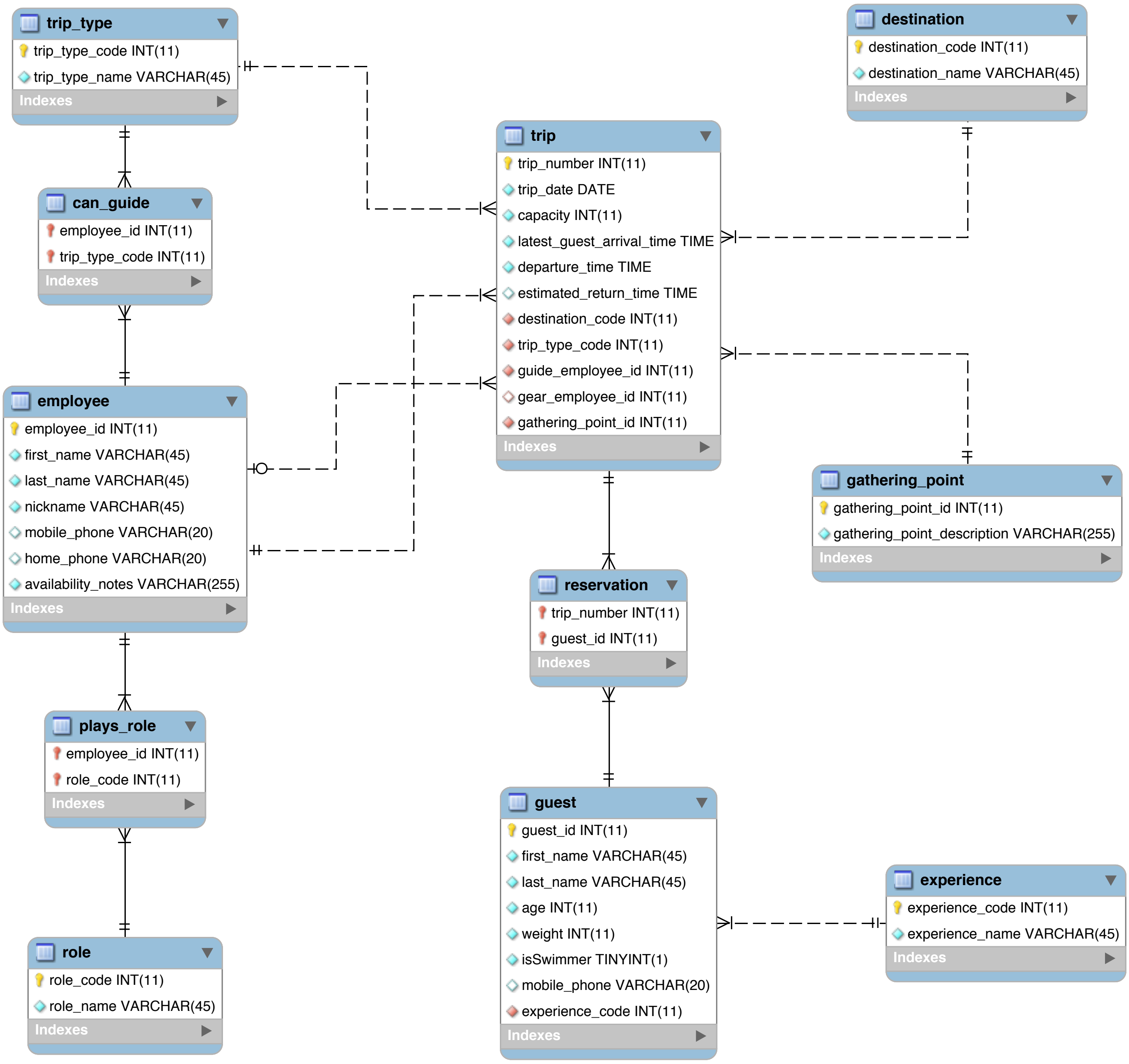
`yourLastName_yourFirstName_final_project`

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 identified in the Weekly Schedule.

Appendix A
WWW Database Logical Design


trip_type
trip_type_code INT(11)
trip_type_name VARCHAR(45)
Indexes
destination
destination_code INT(11)
destination_name VARCHAR(45)
Indexes
trip
trip_number INT(11)
trip_date DATE
capacity INT(11)
latest_guest_arrival_time TIME
departure_time TIME
estimated_return_time TIME
destination_code INT(11)
trip_type_code INT(11)
guide_employee_id INT(11)
gear_employee_id INT(11)
gathering_point_id INT(11)
Indexes
can_guide
employee_id INT(11)
trip_type_code INT(11)
Indexes
employee
employee_id INT(11)
first_name VARCHAR(45)
last_name VARCHAR(45)
nickname VARCHAR(45)
mobile_phone VARCHAR(20)
home_phone VARCHAR(20)
availability_notes VARCHAR(255)
Indexes
gathering_point
gathering_point_id INT(11)
gathering_point_description VARCHAR(255)
Indexes
reservation
trip_number INT(11)
guest_id INT(11)
Indexes
plays_role
employee_id INT(11)
role_code INT(11)
Indexes
guest
guest_id INT(11)
first_name VARCHAR(45)
last_name VARCHAR(45)
age INT(11)
weight INT(11)
isSwimmer TINYINT(1)
mobile_phone VARCHAR(20)
experience_code INT(11)
Indexes
experience
experience_code INT(11)
experience_name VARCHAR(45)
Indexes
role
role_code INT(11)
role_name VARCHAR(45)
Indexes

APPENDIX B
Test Data

Please note that test data has been provided as a series of .CSV files. You may use the values in these files to craft INSERT statements. Or, you can use the values in these files as the source for test data that you enter into tables using the MySQL Workbench graphical user interface. Or, you can use various automated features of MySQL to load test data from the .CSV files into tables automatically.

The test data files are provided in a zip file named:

`test_data_files.zip`

There is one test data file for each of the tables in the WWW Logical Database Design:

- **`can_guide.csv`**
- **`destination.csv`**
- **`employee.csv`**
- **`experience.csv`**
- **`gathering_point.csv`**
- **`guest.csv`**
- **`plays_role.csv`**
- **`reservation.csv`**
- **`role.csv`**
- **`trip_type.csv`**
- **`trip.csv`**

APPENDIX C
Wilma's Existing Spreadsheet-Based Reports

Please note that these reports are provided as examples. They have served as the inspiration for the test data provided. But, they don't match the test data exactly. For test data, please see Appendix B, Test Data.

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