Final Project Instructions

Contents

- Assignment Overview
- Review WWW Case Description
- Review Logical Database Design Document
- Reset MySQL Settings
- Create a MySQL Database Schema
- Populate the Database by Loading Test Data
- Create Database Restore Script
- Code and Test Report Query SQL Scripts
- Code and Test Database Maintenance SQL Scripts
- Retest the Entire System Flow
- Package Files and Submit Your Work
- Appendix A: WWW Logical Database Design
- Appendix B: Test Data
- Appendix C: Wilma's Spreadsheet-Based Reports
- Appendix D: Wilma's Wild Wisconsin Case Description

Assignment Overview

The Final Project Assignment is to create a MySQL-based business system solution for a fictitious small business, Wilma's Wild Wisconsin (WWW). You may accomplish this by following the steps below:

1. Review the WWW case description provided in Appendix D to understand the client for whom the work is being performed and their needs.

2. Review the logical database design document provided in Appendix A to understand the database design upon which this solution is based.

3. Reset the settings of the MySQL database product to make sure they are proper for this project.

4. Create a MySQL database schema for the solution.

5. Populate the database by running the database loading script provided as a starter file for this assignment in the Weekly Schedule. Test data details are provided in Appendix B.

6. Create a database restore script that can be used to create a fresh, populated copy of the database whenever required.

7. Code and test the report query SQL scripts.

8. Code and test the database maintenance SQL scripts.

9. Retest the entire system flow beginning with the database restore script and ending with the database maintenance scripts.

10. Package up the files in your solution and submit your work.

More detail is presented below on each of these steps.

1 Review WWW Case Description

The case description for WWW is provided in Appendix D. Please read these details even if you have seen the WWW case before. The case evolves over time and you want to understand this current version of the case.

2 Review Logical Database Design Document

A logical database design for the WWW project is presented as an ERD in Appendix A. You are expected to implement this design exactly as presented and without deviation. That includes using the table names, column names, and data types provided.

3 Reset MySQL Settings

During the course, we have been running a script that I provided in order to set the SQL modes for MySQL Server. Another copy of this script is provided in the starter files for this assignment:

- `set_sql_modes_for_database_course.sql`

Please be sure to run this script again before starting your Final Project. We will be running this same script before grading your submission. Failing to run this script may make you vulnerable to missing mistakes in your work that will lower your grade.

4 Create a MySQL Database Schema

Creating a MySQL database schema is the physical database design step. Based upon the logical database design provided, you are expected to create the schema while making appropriate choices for the following:

a. Database encoding: make sure that your schema specifies encoding for your database as utf8mb4.
b. Table definitions
c. Column definitions with appropriate data types
   Please make sure that your columns are defined in the same order as shown in the logical database design ERD. Otherwise, test data may not load properly.
d. Primary keys identified in all tables
e. Default values for columns whenever appropriate
f. Auto numbering for columns whenever appropriate
g. Uniqueness constraints that enforce uniqueness in columns other than the primary key whenever appropriate. This is particularly important for name values on lookup tables.
h. NOT NULL constraints that prevent rows being added to tables with important column values set to NULL
i. Foreign key constraints that implement the table-to-table relationships shown in the logical database design

j. Additional indexes created for any column or combination of columns that you believe require them.

Use MySQL Workbench to create your schema. You may use the ERD diagrammer feature to document the design and then generate the schema from the ERD diagram. If using this approach, use the **Database > Forward Engineer** menu options to display the schema script and to generate the database on the server. If you generate the database using forward engineering, make sure that your database encoding is utf8mb4.

Alternatively, you may create your schema SQL script by hand using the SQL Data Definition Language (DDL). In that case, you will need to run your schema script to generate your database on the server. Feel free to use whichever combination of methods you find most effective. Whichever method you use to create your schema, make sure that your schema is named "www".

5 Populate the Database by Running Test Data Loading Script

I have provided a test data loading script as part of the starter files for this assignment:

- **`load_www_test_data_using_inserts.sql`**

I have also provided a script that you may use to check row counts after you have loaded the test data:

- **`display_row_counts_by_table.sql`**

A link to the starter files is available in the Weekly Schedule under the Final Project assignment. See *Appendix B* for more details. These details include expected row counts for each table.

Provided that you have properly created your database schema in the prior step, this script can be used to populate the database with the test data that will be used during the remainder of the final project. Please make sure that your columns are defined in the same order as shown in the logical database design. Otherwise, test data may not load properly.

Remember that you will probably need to revisit this database population step several times. Every time that you find a problem in your schema, you will be expected to create a revised version of the schema. When you use the schema script to create a new database, this new database will NOT be populated. So, you will need to re-run the test data loading script.

Please note that this test data loading script will not be submitted as part of your final project. The first script that you will submit will be the **`01_restore_populated_www_database.sql`** script that you will create in the next step (see below).

6 Create Database Restore Script

`01_restore_populated_www_database.sql`
This script is the first deliverable that will be submitted with your project. Before creating this script, be sure that you have an up-to-date copy of the populated www database. This copy of the database may have been the result of several iterations of steps 4 and 5.

You can create this script using the **Server > Data Export** menu option of the MySQL Workbench tool. Each time you create a script in this manner, **be sure to include the data as well as the structure**. Ignoring this advice can lead to submitting a project that contains no data.

Your schema should be named "www". Also, make sure that this script will run equally well whether or not there is a www schema already defined. To accomplish this, the script should begin with the following code:

- DROP SCHEMA IF EXISTS `www`;
- CREATE SCHEMA IF NOT EXISTS `www`;

This script is the first deliverable for your project. It is the first script that we will run when testing your project submission. It should be the first script that you run when you are testing your project submission. This script is also handy for restoring the database to a reliable state after fixing a problem found during testing.

7 Code and Test Report Query SQL 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 respects. 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 *(first_name + space + 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 *(first_name + space + last_name)*
- role_name

`04_employee_availability_list.sql`
(In order of nickname. One row per employee.)

- nickname
- employee name *(first_name + space + last_name)*
- availability_notes

`05_booking_summary.sql`
(In order of destination_name, trip_type_name, trip_date, trip_number. One row per trip. Make sure that trips without a gear employee and trips without reservations are present in the result set.)

- destination_name
- trip_type_name
- trip_number
- trip_date
- guide nickname
- gear nickname
- 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 reservation. Do NOT include rows for trips that do not have reservations.)

- destination_name
- trip_type_name
- trip_number
- trip_date
- guest full name *(first_name + space + last_name)*
- experience_name
- age
- weight
- IsSwimmer *(Display value 0 as "False". Display value 1 as "True")*
- guest mobile_phone

07_trip_detail_sheet.sql
(*In order of Destination Name, Trip Type Name, Trip Date. One row per trip. Make sure that trips without a gear employee are present in the result set.)*

- destination_name
- trip_type_name
- trip_number
- trip_date
- guide nickname
- guide mobile_phone
- gear nickname
- gear mobile phone
- Wilma's Wild Wisconsin Office Phone *(always 414-555-1212)*
- latest_guest_arrival_time
- departure_time
- estimated_return_time
- gathering point description

08_trip_assignments_view.sql
This script will have two parts. The first part will create a view named *trip_assignments_view*. The second part will be a SELECT that uses *trip_assignments_view*.

The following are the requirements for *trip_assignments_view*:
(*One row per trip. Make sure that trips without a gear employee are present in the result set. Also, make sure that trips without reservations are present in the result set.)*

- trip_date
- trip_number
- destination_name
- trip_type_name
- reservation_count
- guide_nickname
- gear_nickname

The following are the requirements for the SELECT statement:
(*Use trip_assignments_view. In order of trip_date, trip_number. One row per trip.)*

- show all columns from the view

8 Code and Test Database Maintenance SQL Scripts

Note: 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 referential 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 referential 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 referential 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 hard-coded 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 (01 through 15). So, make sure that your queries work when run in that order.

`09_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)

`10_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. Don't worry about the transferred reservation putting the new trip over its capacity limit. As an alternative, you may accomplish the same result using UPDATE.

`11_delete_a_reservation.sql`

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

`12_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

`13_delete_an_existing_employee.sql`
This script should delete the employee Summer Simms.

`14_add_a_new_trip.sql`
This script will add a new trip into the database with the following particulars:

Trip Number	666
Trip Date	8/2/2021
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

`15_delete_an_existing_trip.sql`
This script will DELETE trip #576.

9 Retest the Entire System Flow

The scripts that you will be submitting (especially the database maintenance scripts) are very sensitive to being run in the correct order. To make sure that you will get full credit for your work, it is important to run a final test of the entire system flow. To accomplish this, drop the WWW database, then run the scripts in order (01 through 15). Check that you get the expected results.

10 Package Files and Submit Your Work

Collect all of your **`.sql`** scripts (01 through 15) in a directory named according to the following scheme:

`surname_givenname_final_project`

If this were my own project, I would name the directory as follows:

`trainor_kevin_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:

`surname_givenname_final_project.zip`

If this were my own project, I would name the zip file as follows:

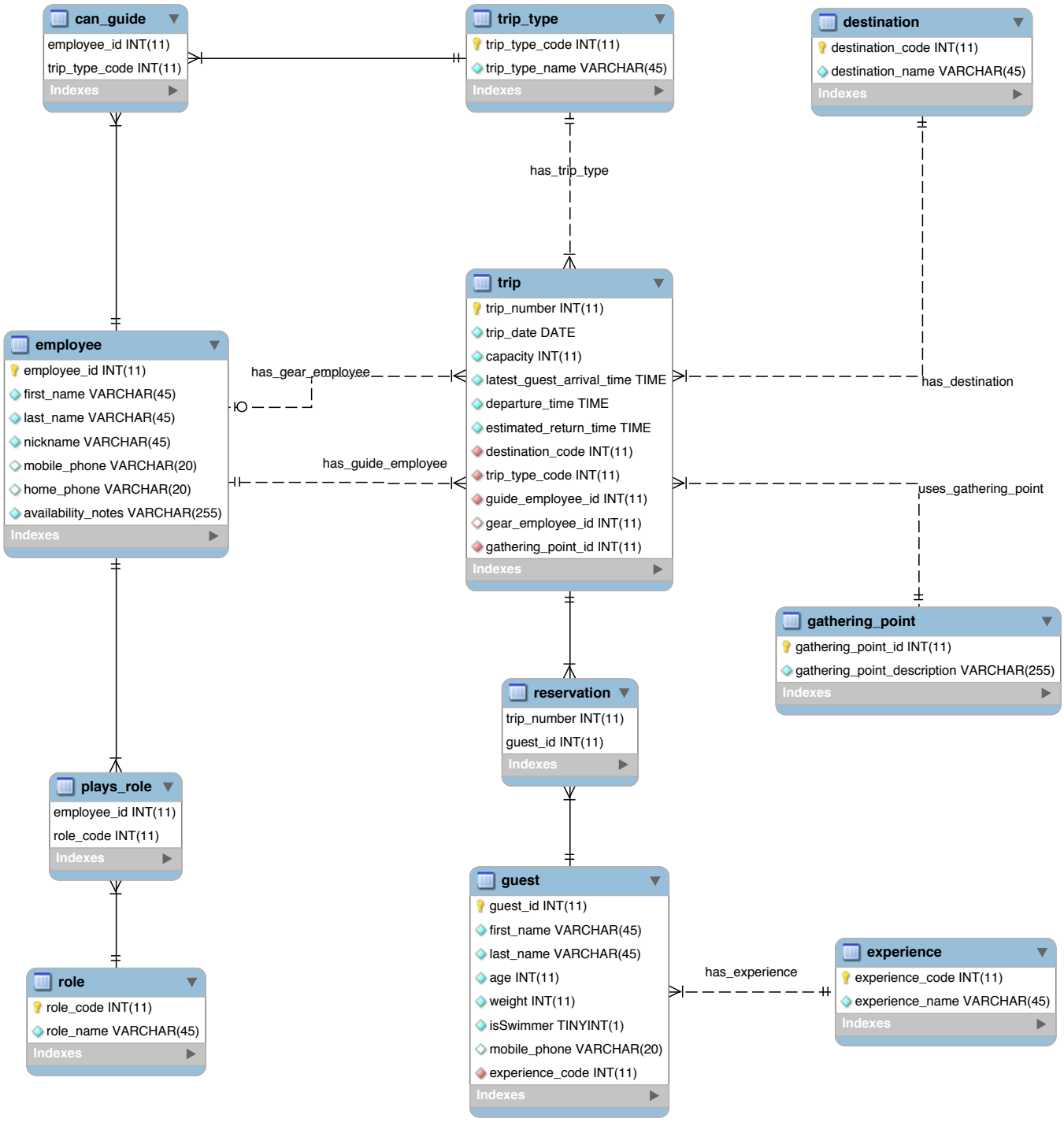
`trainor_kevin_final_project.zip`

Submit this one file to the assignment activity identified in the Weekly Schedule.

Last Revised
2021-03-26

Appendix A
WWW Logical Database Design

Appendix A: Logical Database Design



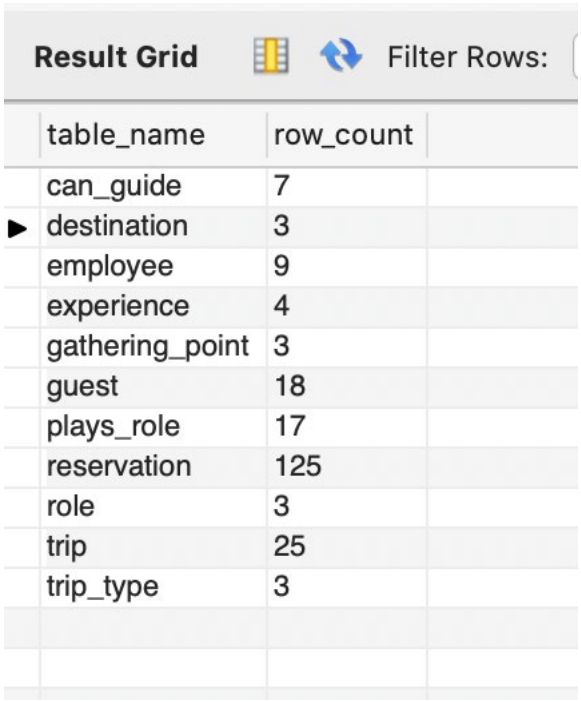
Appendix B
Test Data

The following SQL scripts have been provided as part of the starter files for this project:

- **`load_www_test_data_using_inserts.sql`**
- **`display_row_counts_by_table.sql`**

You may use the first of these scripts to load the test data into your unpopulated database schema. Please make sure that your columns are defined in the same order as shown in the logical database design. Otherwise, test data may not load properly.

When test data loading is complete, you may run the second script to check that the proper number of rows has been added to each table. If loading has been successful, The following result set should be displayed.



table_name	row_count
can_guide	7
destination	3
employee	9
experience	4
gathering_point	3
guest	18
plays_role	17
reservation	125
role	3
trip	25
trip_type	3

Appendix C
Wilma's Spreadsheet-Based Reports

Please note that these copies of Wilma's existing spreadsheet-based reports are provided as examples. They have served as the inspiration for the test data provided. Yet, they don't match the test data exactly. For test data, please see Appendix B.

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 D
Wilma's Wild Wisconsin Case Description

This final project is based upon the Wilma's Wild Wisconsin case that I use in several of my courses. As we move from semester to semester and from course to course, I often make changes to the case. So, please read the material below to get a full understanding of the current version of this case.

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.