Campground Management System

CIS 3050

Summer 2023

Overview

This project involves working with a dataset from a campground business. Your task is to design a logical model ERD based on the provided data, create the corresponding database, and execute queries to extract meaningful insights and address specific business questions. The dataset and part of the database structure as well as instructor's demo videos are provided to you.

Business Rules

1. The system stores camper's last name, first name, address, city, state, zip code, drivers license, email.
2. The system stores camping spot's name, length, pull through (yes or no for back in spot), electric amperage, water (yes or no), sewer (yes or no), peak and normal rates.
3. A camper may reserve a spot on a specific date for a number of nights at either peak or normal rate with deposit.
4. Each camper may reserve multiple spots. Each spot must be reserved by one and only one camper on a specific date.
5. Each spot may be reserved by multiple campers on different dates. Each reservation must be associated with one and only one spot.

Dataset

You are provided with un-normalized data and normalized data for the database containing CAMPER, RESERVATIONS, and SPOT (Canvas – Modules – Individual Project -> campground.xlsx).

Task 1: Create an ERD for the campground database

You are provided with a database structure containing table names, field names, data types, primary keys, foreign keys, and the cardinality of relationships (Canvas – Modules – Individual Project -> campground.xlsx). You can follow the instructor's videos (links in Canvas – Modules – Individual Project) to create the ERD containing CAMPER, RESERVATIONS, and SPOT using either ERWin or LucidChart.

Task 2: Create the campground database

Then you will export the SQL queries from Erwin or LucidChart (you may need to manually create some extra SQL queries as shown in the demo video) and create the database in either Microsoft SQL Server or MySQL by running the SQL queries.

Task 3: Modify the database design to add a new business extension of camping equipment rental

Now the campground business wants to extend its business to renting camping equipment to campers.

The new business rules:

- A camper may rent multiple equipment during a reservation.
- A equipment can be rented to multiple campers during their reservations.

The rental records:

- When camper Pat Schmidt made Reservation 3, she rented 1 camp stove and 2 kayaks.
- When camper Clifford Williams made Reservation 5, he rented 1 camping tent, 2 sleeping bags, and 2 bicycles.

Hint:

- Is Equipment more relate to CAMPER or RESERVATION?
- Is it an one-to-many or many-to-many relationship?

Modify the ERD to include necessary entities and attributes, update the database structure, and populate the database with the new data given above.

Task 4: Complete the data dictionary and the referential integrity

- Complete the following **Table 1 Data Dictionary**. List ***entities*** and their ***attributes***, mark if Primary Keys (PK) and/or Foreign Keys (FK), and their data types, as the examples shown.
- Then complete **Table 2 Referential Integrity** below, specify the relationships and their cardinalities, as the example shows.
- Make sure to maintain 3rd Normal Form for each relation, i.e. no repeating groups, partial dependencies or transitive dependencies.

Table 1: Data Dictionary (to be completed by you)

Entity	Key (PK and/or FK)	Attribute or Field Name	Data Type* and Field Length
CAMPER	PK	CAMPER_ID	Number / Integer
		CAMPER_LAST_NAME	varchar(50)
		CAMPER_FIRST_NAME	varchar(50)
		CAMPER_ADDRESS	varchar(100)
		CAMPER_CITY	varchar(50)
		CAMPER_STATE	varchar(2)
		CAMPER_ZIP_CODE	varchar(10)
		CAMPER_DRIVERS_LICENSE	varchar(20)
		CAMPER_EMAIL	varchar(50)
SPOT	PK	SPOT_NUMBER	int
		SPOT_NAME	varchar(50)
		SPOT_LENGTH	int
		SPOT_PULLTHRU	int
		SPOT_ELECTRIC_AMPS	int
		SPOT_WATER	int
		SPOT_SEWER	int
		SPOT_RATE_NORMAL	decimal(10,2)
		SPOT_RATE_PEAK	decimal(10,2)
RESERVATION	PK	RESV_NUMBER	int
		RESV_DATE	date

		RESV_NIGHTS	int
	FK	SPOT_NUMBER	int
	FK	CAMPER_NUMBER	int
		RESV_RATE_PEAK	int
		RESV_DEPOSIT	decimal(10,2)
Equipment	PK	EQ_NUMBER	int
		EQ_NAME	varchar(50)
		EQ_DAILY_PRICE	decimal(10,2)
		INVENTORY	int
	FK	RESV_NUMBER	int
EquipmentRental	PK	EQ_NUMBER	int
		RENTAL_DATE	date
		RETURN_DATE	date
	FK	CAMPER_NUMBER	int
		RENTAL_FEE	decimal(10,2)

**** When you create a new attribute or field in ERD you should assign its data type.***

- You can choose ***one from the four domains***: Blob, Date/Time, Number, String.
- Under each ***domain*** there are various choices, for example, **Number** can be Integer or Decimal. Use Number type only for attributes that are meaningful for arithmetic calculations, such as quantity or price. For Decimal (p,s), p is precision, s is scale. Precision refers to number of digits in a number, minimum 1, maximum 39. Scale refers to the number of digits to the right of the decimal point. The scale of a decimal value cannot exceed its precision. For example, Decimal (9,5) can store 1234.56789. https://docs.actian.com/ingres/10s/index.html#page/SQLRef/Decimal_Data_Type.htm
- If you don't specify the data type for each attribute appropriately, you may encounter errors during the ERwin forward engineering process because SQL Server will not accept inappropriate data types, such as precision and scales for decimal type of data. For example, use Decimal (10,2) for prices instead of Decimal (). Similarly, if you don't specify the field width for each attribute/column as described in the table, data may be truncated during the data populating process.
- In the case you are using Lucid Chart, not specifying the appropriate data type can also cause issues when populating the database with data later.

Table 2: Referential Integrity (to be completed by you)

Relationship	**Cardinality Constraints****
CAMPER -> RESERVATION	Optional Many
RESERVATION -> CAMPER	Mandatory One
SPOT -> RESERVATION	Optional Many
RESERVATION -> SPOT	Mandatory One
RESERVATION -> EQUIPMENT	Optional Many
EQUPMENT -> RESERVATION	Mandatory One
EQUIPMENT -> EQUIPMENTRENTAL	Optional Many
EQUIPMENTRENTAL -> EQUIPMENT	Mandatory One
CAMPER -> EQUIPMENTRENTAL	Optional Many
EQUIPMENTRENTAL -> CAMPER	Mandatory One

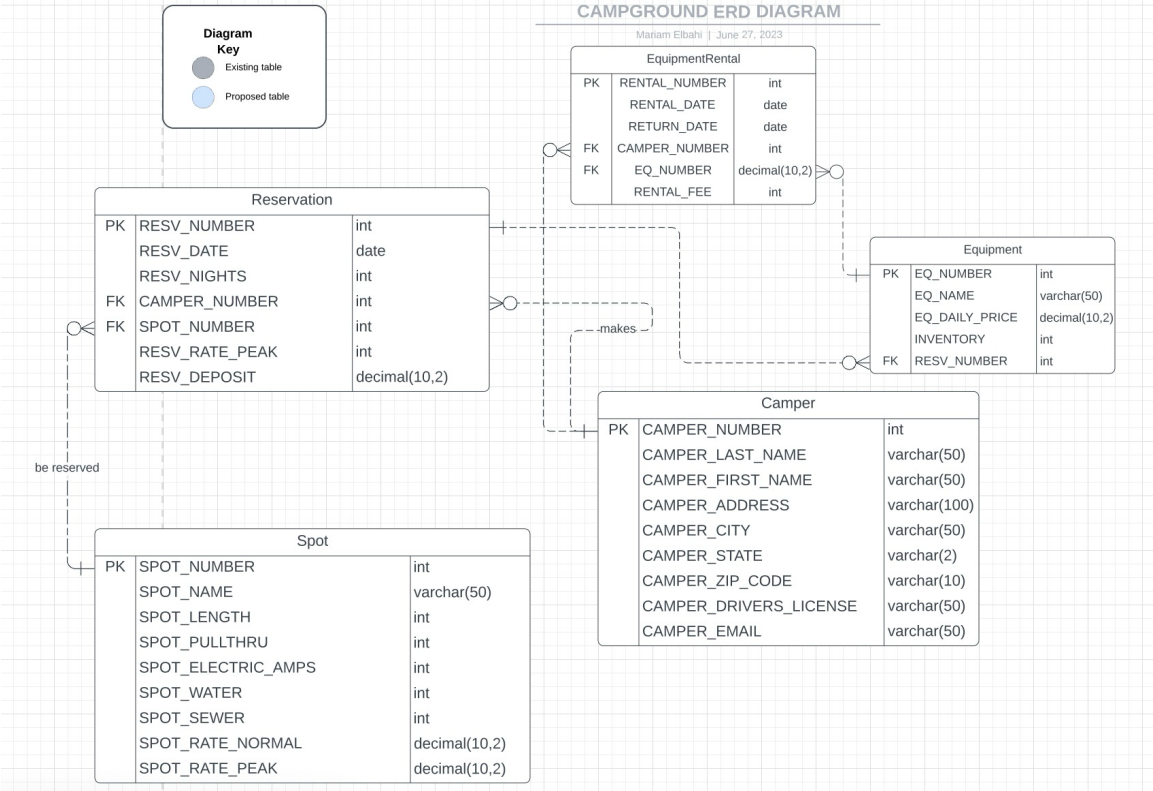
***** Cardinality Constraints can be:***

- Optional Many - means 1 to many with the minimum 0.
- Mandatory Many – means 1 to many with the minimum 1.
- Optional One - means 1 to 1 with the minimum 0.
- Mandatory One – means 1 to 1 with the minimum 1.

Task 5: Provide a screenshot of the final ERD

Take a screenshot of the final ERD including CAMPER, RESERVATION, SPOT, and the new entities and relationships for the equipment rental business extension and paste it below.

- Make sure you have the correct primary keys, foreign keys, data types, relationships, and cardinalities for these tables.
- Make sure the ERD is consistent with Table 1 & 2.
- Make sure the screenshot is readable to the level of each relationship description and cardinality constraint.



Hints:

- You will need to create one new associative entity/table. Associative entity is usually used in ERD design to normalize i.e. reduce duplicate records, when there is a many-to-many relationship (M:N). As the result, instead of the M:N relationship between original two entities, there will be two one-to-many relationships (1:N) between each of the original entities and the new associative entity. I suggest you review Chapter 4 Associative (Composite) Entities and the existing ERD (CAMPER-Reservation-SPOT).

- Other than primary keys and foreign keys, there is no need to create additional attributes that you don't see in the business rules. You do not need to create anything like rent total since that can be calculated based on data stored in the database. You do not need to worry about tax rate either.

- You only need to create sub entities if there are unique attributes or relationships for each sub entities that's not covered by super entities, however this project didn't indicate any, so there shouldn't be any sub entities in your ERD.

- Make sure that you provide all cardinalities for all the relationships in Task 2, there should be two cardinalities for each relationship, one at each end. For example, I already provide that from CAMPER to RESERVATION is optional many because of the business rule "Each camper may reserve multiple spots. " From RESERVATION to CAMPER is mandatory one because of the business rule " Each reservation must be associated with one and only one spot. "

- In ERD there should be a relationship between any two entities; there shouldn't be any "orphan" entity that doesn't link to anyone else. Also, an entity/table is a collection of many instances or records, if you find only one possible instance/record that could be put into a table, that means you shouldn't create this table at all, for example, the company PSC shouldn't be an entity or table itself, there is only one PSC - one name, address, and phone

Task 6: Provide all the SQL queries you used to create the Database in MS SQL Server or MySQL

Use ERwin to create database in MS SQL Server:

Use ERwin forward engineering function to automatically create the database schema in Microsoft SQL Server. Save the .ere file generated during the process. Watch my demo video (link posted on Canvas -> Modules -> Individual Project) to learn how to forward engineer ERwin model to SQL Server.

Make sure your SQL Server is turned on (by default it should be on, if not you can reboot your computer or turn it on in SQL Server Configuration Manager) and connected to the Management Studio (In “Connect to Server” window, use Server type: Database Engine, Server name: localhost\SQLExpress or you can replace localhost with your computer name, Authentication: Windows Authentication). This is also explained in the demo video “ERwin Forward Engineering to MS SQL Server Management Studio” (link posted on Canvas -> Modules -> Individual Project).

Use Lucid Chart to create database in MySQL or MS SQL Server:

If you use **Lucid Chart** then copy and paste the SQL for creating each table including its primary key. Then write the SQL query for establish the relationships between tables by adding the referential integrity i.e. matching PK with FK. Watch my demo video “How to create an ERD diagram using Lucid

Chart and export SQL to create the database in MySQL Links to an external site" (link posted on Canvas -> Modules -> Individual Project) to learn how to.

Then run these queries in the DBMS such as MySQL or SQL Server to create the database.

If you used ERwin to create ERD, you can **copy the SQL queries** when generating the database schema as demonstrated in the video "ERwin Forward Engineering to MS SQL Server Management Studio" at 3:08. Then Copy and paste the **SQL queries for creating the database** below.

If you used Lucid Chart to create ERD, you can **export SQL in Lucid Chart** and you may need to manually create some extra SQL queries as shown in the demo video "How to create an ERD diagram using Lucid Chart and export SQL to create the database in MySQL." Copy and paste both the Lucid Chart exported SQL queries and manually created SQL queries below.

Lucid Chart Exported SQL:

```
CREATE TABLE `Camper` (
  `CAMPER_NUMBER` int,
  `CAMPER_LAST_NAME` varchar(50),
  `CAMPER_FIRST_NAME` varchar(50),
  `CAMPER_ADDRESS` varchar(100),
  `CAMPER_CITY` varchar(50),
  `CAMPER_STATE` varchar(2),
  `CAMPER_ZIP_CODE` varchar(10),
  `CAMPER_DRIVERS_LICENSE` varchar(50),
  `CAMPER_EMAIL` varchar(50),
  PRIMARY KEY (`CAMPER_NUMBER`)
);

CREATE TABLE `Spot` (
  `SPOT_NUMBER` int,
  `SPOT_NAME` varchar(50),
  `SPOT_LENGTH` int,
  `SPOT_PULLTHRU` int,
  `SPOT_ELECTRIC_AMPS` int,
  `SPOT_WATER` int,
  `SPOT_SEWER` int,
  `SPOT_RATE_NORMAL` decimal(10,2),
  `SPOT_RATE_PEAK` decimal(10,2),
  PRIMARY KEY (`SPOT_NUMBER`)
);

CREATE TABLE `Reservation` (
  `RESV_NUMBER` int,
  `RESV_DATE` date,
  `RESV_NIGHTS` int,
  `CAMPER_NUMBER` int,
  `SPOT_NUMBER` int,
  `RESV_RATE_PEAK` int,
  `RESV_DEPOSIT` decimal(10,2),
  PRIMARY KEY (`RESV_NUMBER`),
```

```
  FOREIGN KEY (`CAMPER_NUMBER`) REFERENCES
`Camper`(`CAMPER_NUMBER`),
  FOREIGN KEY (`SPOT_NUMBER`) REFERENCES `Spot`(`SPOT_NUMBER`)
);

CREATE TABLE `Equipment` (
  `EQ_NUMBER` int,
  `EQ_NAME` varchar(50),
  `EQ_DAILY_PRICE` decimal(10,2),
  `INVENTORY` int,
  `RESV_NUMBER` int,
  PRIMARY KEY (`EQ_NUMBER`),
  FOREIGN KEY (`RESV_NUMBER`) REFERENCES `Reservation`(`RESV_NUMBER`)
);

CREATE TABLE `EquipmentRental` (
  `RENTAL_NUMBER` int,
  `RENTAL_DATE` date,
  `RETURN_DATE` date,
  `CAMPER_NUMBER` int,
  `EQ_NUMBER` int,
  `RENTAL_FEE` decimal(10,2),
  PRIMARY KEY (`RENTAL_NUMBER`),
  FOREIGN KEY (`CAMPER_NUMBER`) REFERENCES
`Camper`(`CAMPER_NUMBER`),
  FOREIGN KEY (`EQ_NUMBER`) REFERENCES `Equipment`(`EQ_NUMBER`)
);
```

Manually Created SQL:

```
--
-- Database: `Campground`
--
CREATE DATABASE IF NOT EXISTS `Campground` DEFAULT CHARACTER SET
utf8 COLLATE utf8_general_ci;
USE `Campground`;
CREATE TABLE `Camper` (
  `CAMPER_NUMBER` int,
  `CAMPER_LAST_NAME` varchar(50),
  `CAMPER_FIRST_NAME` varchar(50),
  `CAMPER_ADDRESS` varchar(100),
  `CAMPER_CITY` varchar(50),
  `CAMPER_STATE` varchar(2),
  `CAMPER_ZIP_CODE` varchar(10),
  `CAMPER_DRIVERS_LICENSE` varchar(50),
  `CAMPER_EMAIL` varchar(50),
  PRIMARY KEY (`CAMPER_NUMBER`)
);

CREATE TABLE `Spot` (
```

```
  `SPOT_NUMBER` int,
  `SPOT_NAME` varchar(50),
  `SPOT_LENGTH` int,
  `SPOT_PULLTHRU` int,
  `SPOT_ELECTRIC_AMPS` int,
  `SPOT_WATER` int,
  `SPOT_SEWER` int,
  `SPOT_RATE_NORMAL` decimal(10,2),
  `SPOT_RATE_PEAK` decimal(10,2),
  PRIMARY KEY (`SPOT_NUMBER`)
);

CREATE TABLE `Reservation` (
  `RESV_NUMBER` int,
  `RESV_DATE` date,
  `RESV_NIGHTS` int,
  `CAMPER_NUMBER` int,
  `SPOT_NUMBER` int,
  `RESV_RATE_PEAK` int,
  `RESV_DEPOSIT` decimal(10,2),
  PRIMARY KEY (`RESV_NUMBER`),
  FOREIGN KEY (`CAMPER_NUMBER`) REFERENCES
`Camper`(`CAMPER_NUMBER`),
  FOREIGN KEY (`SPOT_NUMBER`) REFERENCES `Spot`(`SPOT_NUMBER`)
);

CREATE TABLE `Equipment` (
  `EQ_NUMBER` int,
  `EQ_NAME` varchar(50),
  `EQ_DAILY_PRICE` decimal(10,2),
  `INVENTORY` int,
  `RESV_NUMBER` int,
  PRIMARY KEY (`EQ_NUMBER`),
  FOREIGN KEY (`RESV_NUMBER`) REFERENCES `Reservation`(`RESV_NUMBER`)
);

CREATE TABLE `EquipmentRental` (
  `RENTAL_NUMBER` int,
  `RENTAL_DATE` date,
  `RETURN_DATE` date,
  `CAMPER_NUMBER` int,
  `EQ_NUMBER` int,
  `RENTAL_FEE` decimal(10,2),
  PRIMARY KEY (`RENTAL_NUMBER`),
  FOREIGN KEY (`CAMPER_NUMBER`) REFERENCES
`Camper`(`CAMPER_NUMBER`),
  FOREIGN KEY (`EQ_NUMBER`) REFERENCES `Equipment`(`EQ_NUMBER`)
);
```

Hints:

When you populate the data, be aware that existing integrity constraints will force you to enter data in certain orders. For example, the Camper ID and Spot ID are required in Reservation table so you cannot populate the Camper and Spot tables before you populate Reservation table.

One way to populate the data is to use INSERT SQL statements (SQL Server Management Studio -> New Query). When you use SQL insert statements to populate the data, be aware of the columns that do not have any data, enter a pair of empty quotes (,'',) for empty string type of columns, and null (,,) for empty number type of columns, otherwise, the SQL Server will not execute your insert statement. Note that date is text-based too so when you insert a date value don't forget to use '', otherwise incorrect value will be inserted. Alternatively, you can insert values into selected columns instead of all columns.

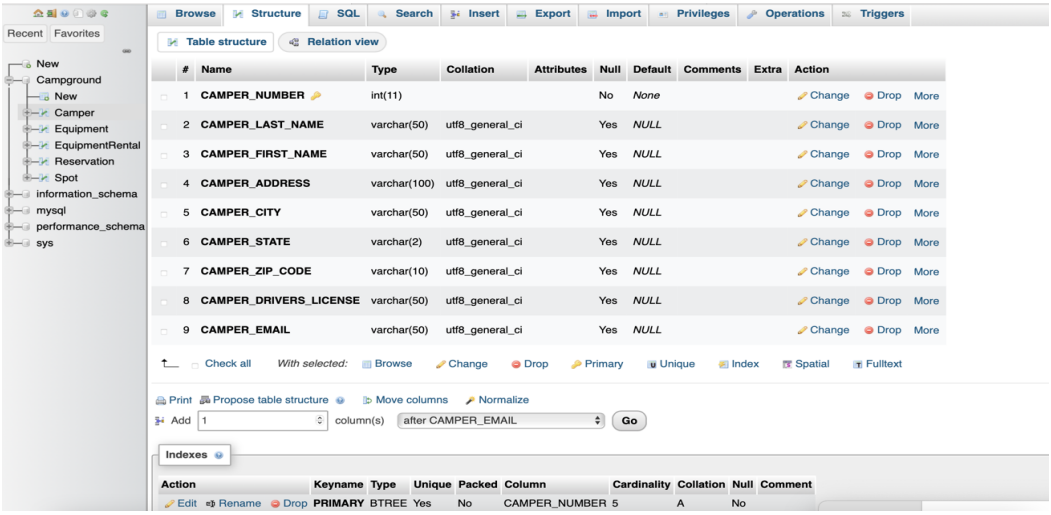
Sometimes you will realize your database structure is built incorrectly once you started to populate data into it, and because you created the database structure using ERwin Forward Engineering function, it may have limitations on what you can change afterward – such as changing the data type for a non-key column in the table design in Management Studio. You may first need to turn off the default "Prevent saving changes that require table re-creation", by going to the top menu, selecting Tools -> Options -> Designer, and uncheck the "Prevent saving changes that require table re-creation" option. In most cases you probably will find it easier to detach and delete the database, correct your ERwin model and re-create the SQL Server database from ERwin.

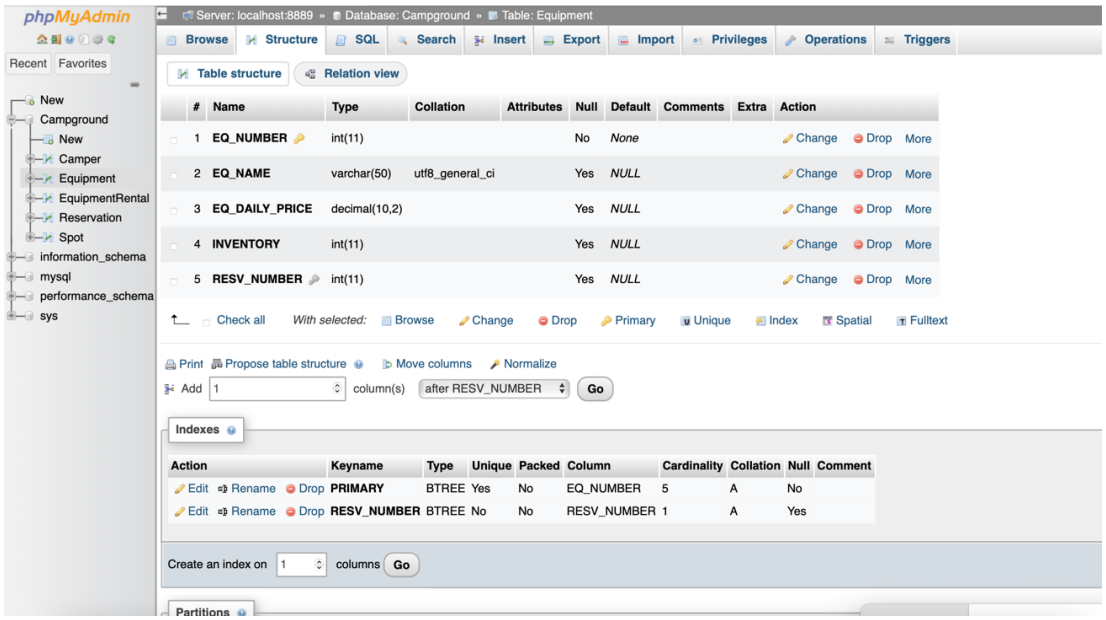
Another way to populate the data is to import the data from an Excel spreadsheet using the Query Wizard, however, the same integrity constraints apply, so you will still need to populate the tables in order.

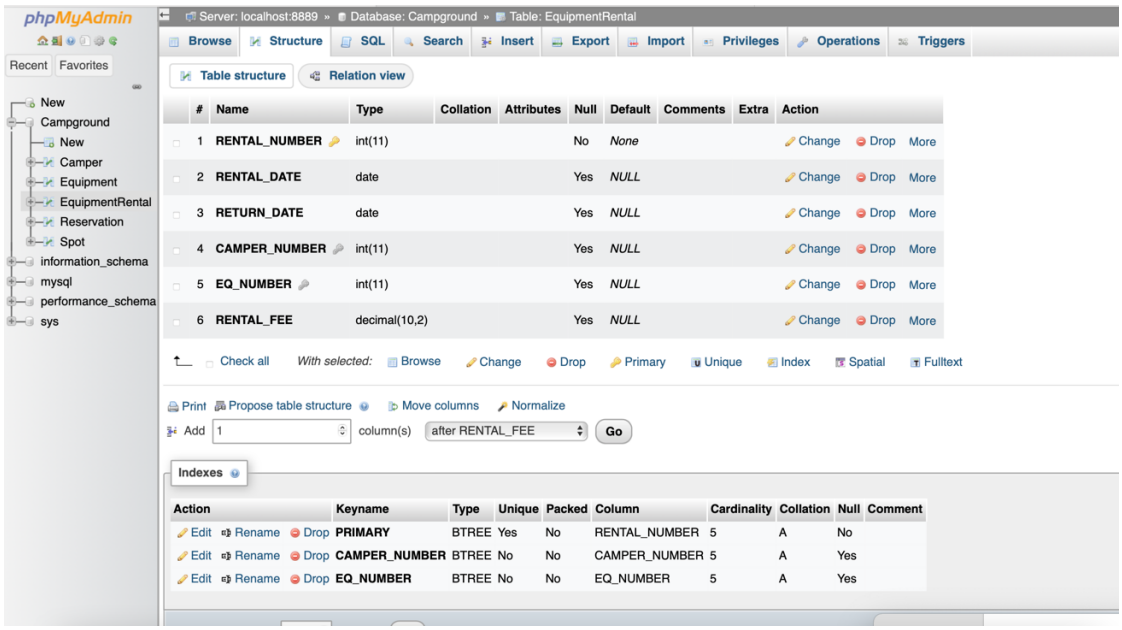
Lastly, you are able to populate the data manually in Design View in Management Studio – this is not preferred because it is highly labor-intensive and subject to human error. The method may work with this class project but not realistic in a real-world scenario. Also, when you do this do not turn on the auto-generated identifier option.

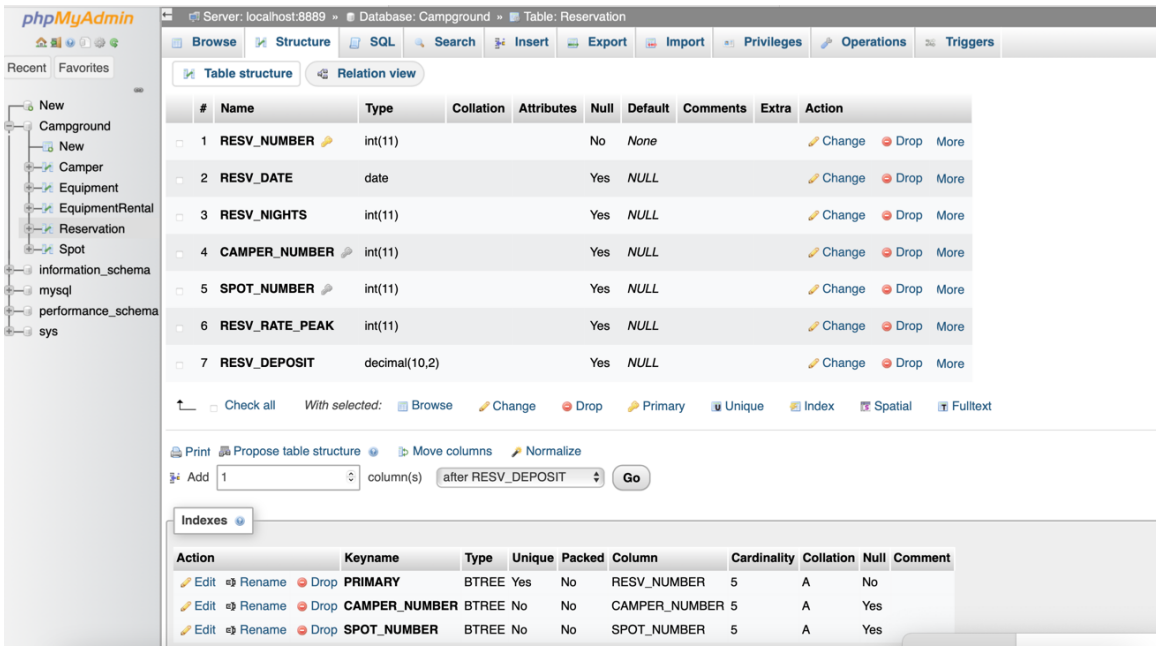
Task 7: Provide the database design and data populated.

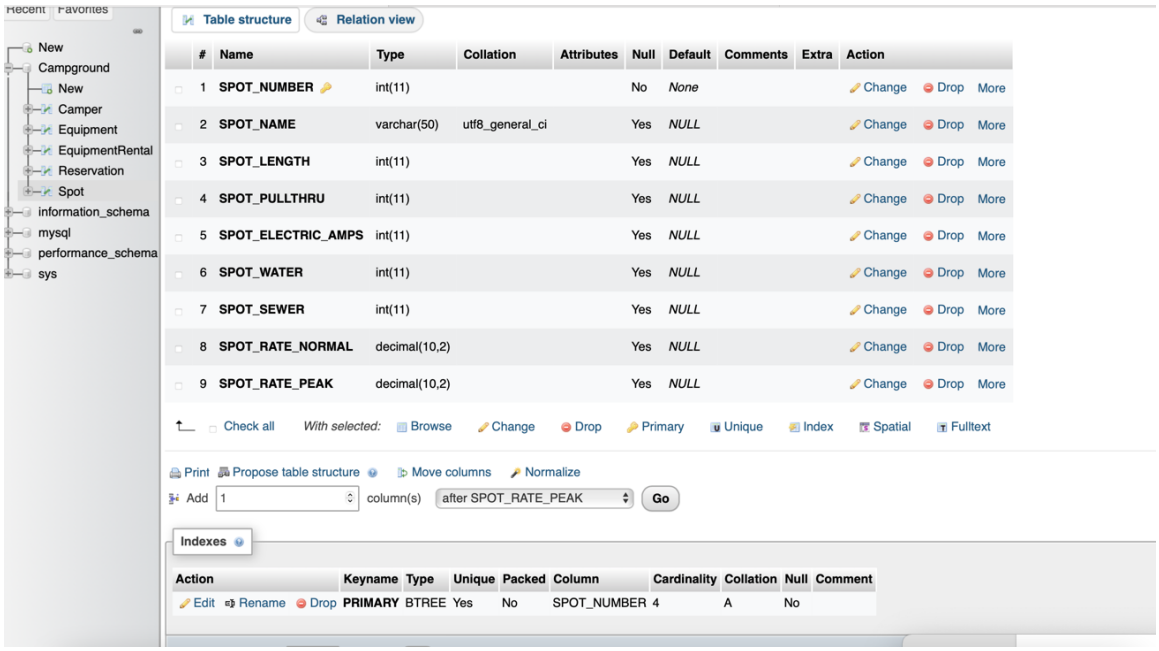
- **Take a screenshot of the database design or structure for each table** by right click on the table name and select "Design" or use Design View and paste them below. Make sure to include table's name, and structure of all columns and all rows.



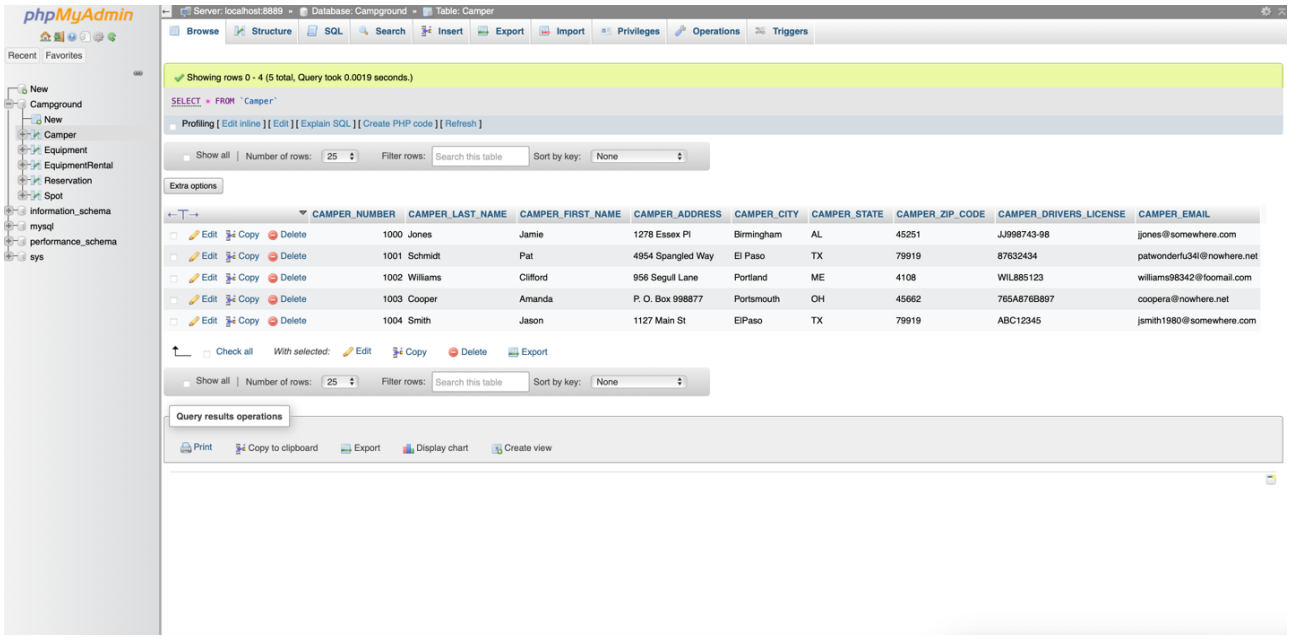

phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: Equipment
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 EQ_NUMBER int(11) No None Change Drop More
2 EQ_NAME varchar(50) utf8_general_ci Yes NULL Change Drop More
3 EQ_DAILY_PRICE decimal(10,2) Yes NULL Change Drop More
4 INVENTORY int(11) Yes NULL Change Drop More
5 RESV_NUMBER int(11) Yes NULL Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
Print Propose table structure Move columns Normalize
Add 1 column(s) after RESV_NUMBER Go
Indexes
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Rename Drop PRIMARY BTREE Yes No EQ_NUMBER 5 A No
Edit Rename Drop RESV_NUMBER BTREE No No RESV_NUMBER 1 A Yes
Create an index on 1 columns Go
Partitions

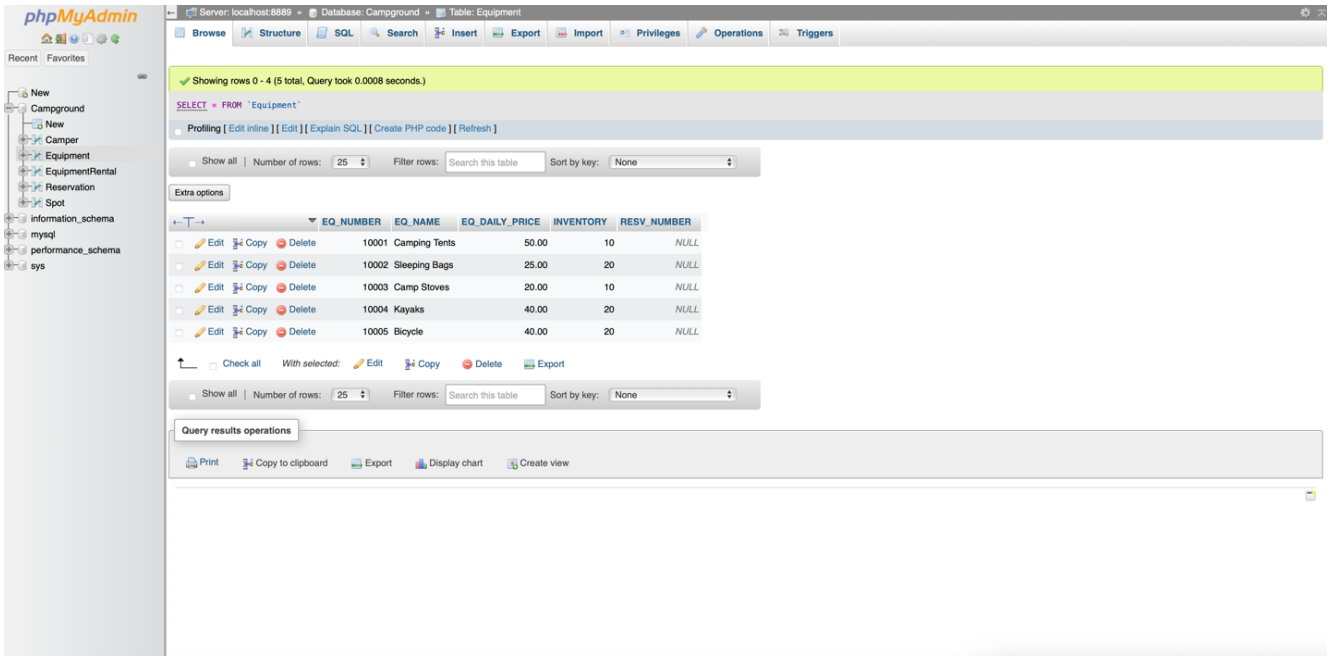

phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: EquipmentRental
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Table structure Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 RENTAL_NUMBER int(11) No None Change Drop More
2 RENTAL_DATE date Yes NULL Change Drop More
3 RETURN_DATE date Yes NULL Change Drop More
4 CAMPER_NUMBER int(11) Yes NULL Change Drop More
5 EQ_NUMBER int(11) Yes NULL Change Drop More
6 RENTAL_FEE decimal(10,2) Yes NULL Change Drop More
Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext
Print Propose table structure Move columns Normalize
Add 1 column(s) after RENTAL_FEE Go
Indexes
Action Keyname Type Unique Packed Column Cardinality Collation Null Comment
Edit Rename Drop PRIMARY BTREE Yes No RENTAL_NUMBER 5 A No
Edit Rename Drop CAMPER_NUMBER BTREE No No CAMPER_NUMBER 5 A Yes
Edit Rename Drop EQ_NUMBER BTREE No No EQ_NUMBER 5 A Yes

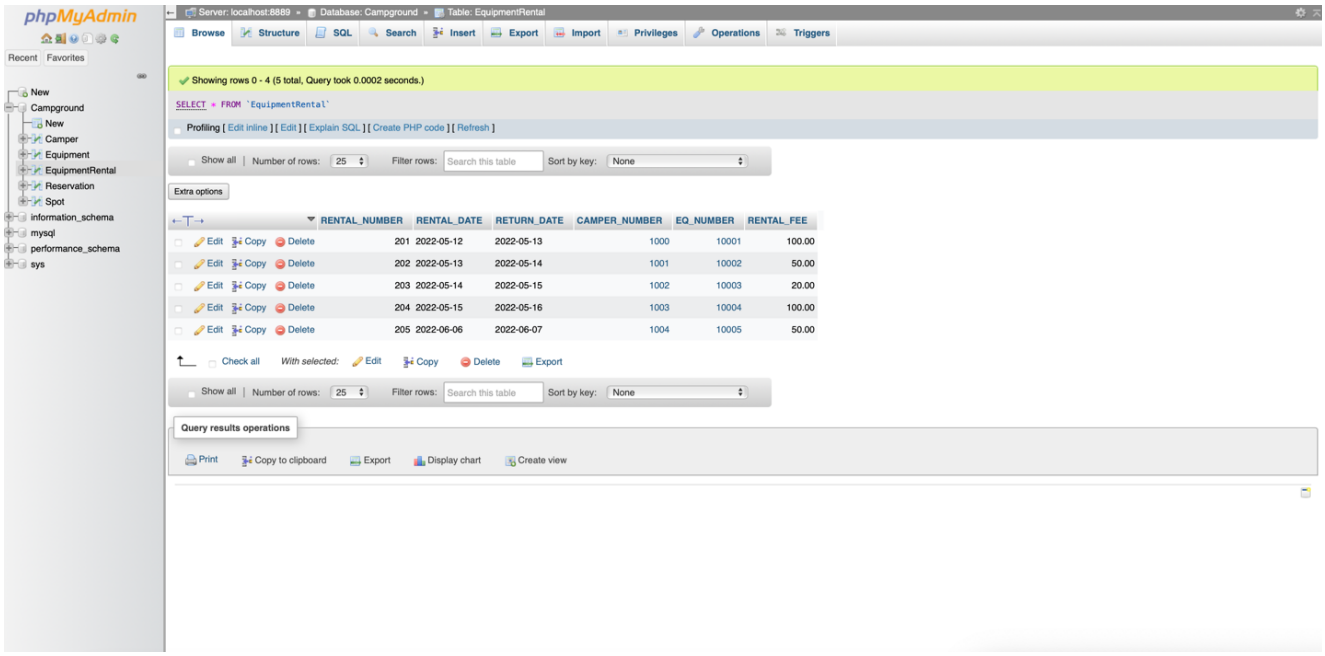


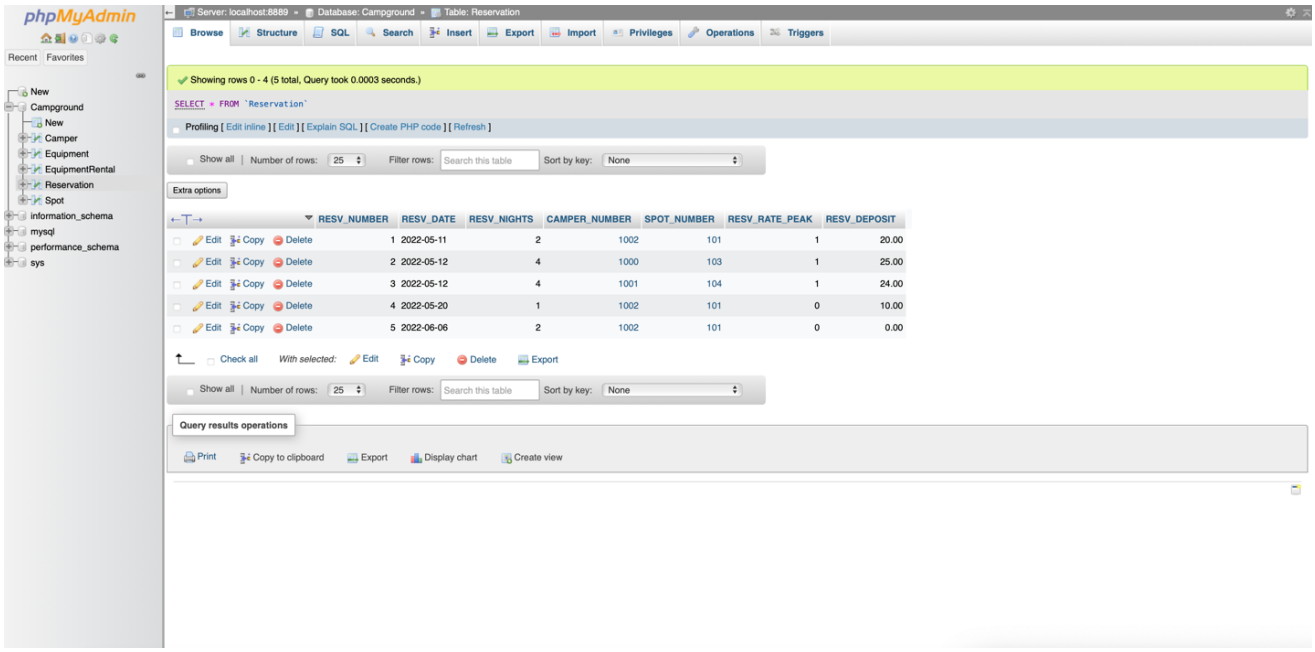


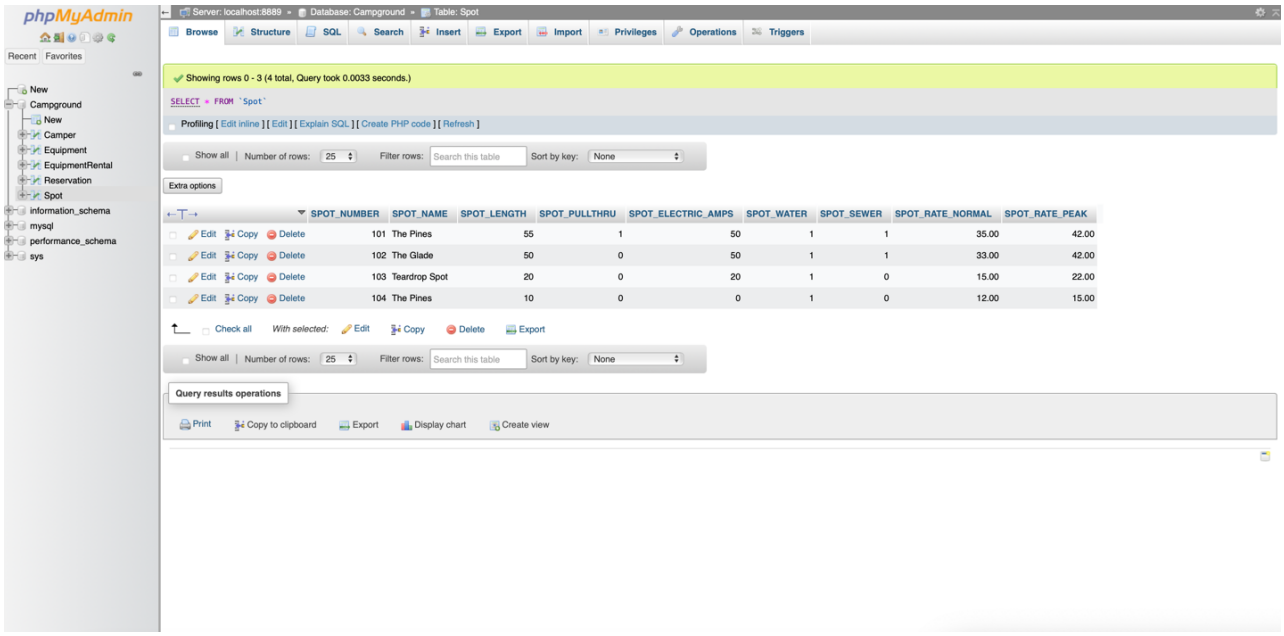
- **Take a screenshot of all the data in each table by using SELECT * FROM [TABLE_NAME] query and paste them below**. Make sure to include table's name, and data in all columns and all rows.


phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: Camper
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 4 (5 total, Query took 0.0019 seconds.)
SELECT * FROM `Camper`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
CAMPER_NUMBER CAMPER_LAST_NAME CAMPER_FIRST_NAME CAMPER_ADDRESS CAMPER_CITY CAMPER_STATE CAMPER_ZIP_CODE CAMPER_DRIVERS_LICENSE CAMPER_EMAIL
Edit Copy Delete 1000 Jones Jamie 1278 Essex Pl Birmingham AL 45251 JJ998743-98 jjones@somewhere.com
Edit Copy Delete 1001 Schmidt Pat 4954 Spangled Way El Paso TX 79919 87632434 patwonderfu34l@nowhere.net
Edit Copy Delete 1002 Williams Clifford 956 Segull Lane Portland ME 4108 WIL885123 williams98342@foomail.com
Edit Copy Delete 1003 Cooper Amanda P. O. Box 998877 Portsmouth OH 45662 765A876B897 coopera@nowhere.net
Edit Copy Delete 1004 Smith Jason 1127 Main St ElPaso TX 79919 ABC12345 jsmith1980@somewhere.com
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view


phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: Equipment
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)
SELECT * FROM `Equipment`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
EQ_NUMBER EQ_NAME EQ_DAILY_PRICE INVENTORY RESV_NUMBER
Edit Copy Delete 10001 Camping Tents 50.00 10 NULL
Edit Copy Delete 10002 Sleeping Bags 25.00 20 NULL
Edit Copy Delete 10003 Camp Stoves 20.00 10 NULL
Edit Copy Delete 10004 Kayaks 40.00 20 NULL
Edit Copy Delete 10005 Bicycle 40.00 20 NULL
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view


phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: EquipmentRental
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 4 (5 total, Query took 0.0002 seconds.)
SELECT * FROM `EquipmentRental`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
RENTAL_NUMBER RENTAL_DATE RETURN_DATE CAMPER_NUMBER EQ_NUMBER RENTAL_FEE
Edit Copy Delete 201 2022-05-12 2022-05-13 1000 10001 100.00
Edit Copy Delete 202 2022-05-13 2022-05-14 1001 10002 50.00
Edit Copy Delete 203 2022-05-14 2022-05-15 1002 10003 20.00
Edit Copy Delete 204 2022-05-15 2022-05-16 1003 10004 100.00
Edit Copy Delete 205 2022-06-06 2022-06-07 1004 10005 50.00
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view


phpMyAdmin
Recent Favorites
New
Campground
New
Camper
Equipment
EquipmentRental
Reservation
Spot
information_schema
mysql
performance_schema
sys
Server: localhost:8889 » Database: Campground » Table: Reservation
Browse Structure SQL Search Insert Export Import Privileges Operations Triggers
Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)
SELECT * FROM `Reservation`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options
RESV_NUMBER RESV_DATE RESV_NIGHTS CAMPER_NUMBER SPOT_NUMBER RESV_RATE_PEAK RESV_DEPOSIT
Edit Copy Delete 1 2022-05-11 2 1002 101 1 20.00
Edit Copy Delete 2 2022-05-12 4 1000 103 1 25.00
Edit Copy Delete 3 2022-05-12 4 1001 104 1 24.00
Edit Copy Delete 4 2022-05-20 1 1002 101 0 10.00
Edit Copy Delete 5 2022-06-06 2 1002 101 0 0.00
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Query results operations
Print Copy to clipboard Export Display chart Create view



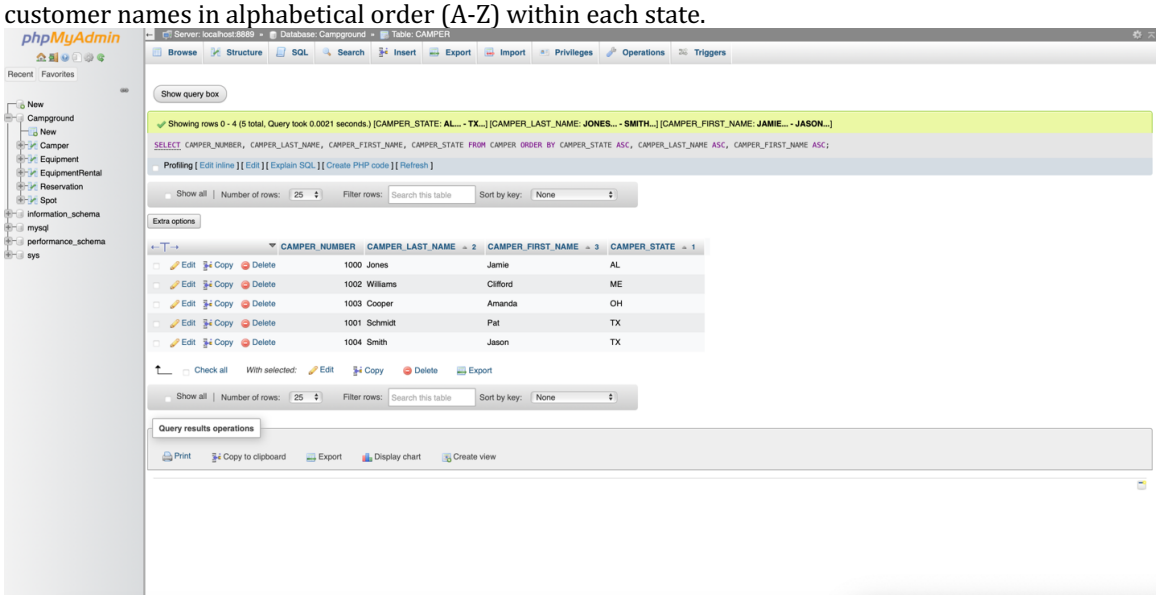
Task 8: Use SQL to retrieve data from databases

Please read the requirements document carefully, only use the criteria that were given, and only display what was asked for in your query results. There are many ways to query to get the same results, but you should deliver the simplest and most efficient way.

You must use one and only one SQL Statement to get results for each of the following items.

Query # **Description**

1. Select all campers and display camper's id, name, and state, sort the results by the state, with customer names in alphabetical order (A-Z) within each state.



2. Select all campers and display camper's id, name, Reservation id, date, and Total due per reservation (in dollar amount, calculated field, use alias), sort the results by Total due (the highest amount first).

[Include a screenshot showing your complete SQL Query and complete results returned]

3. Select the camper(s) who reserved the spot that has the word "Pines" in it after 5/11/2022, display the camper(s)' name(s), spot name, # of reservation nights, and the deposit, sort the results by date ascending, meaning the oldest ones come first and the most recent ones last.

[Include a screenshot showing your complete SQL Query and complete results returned]

4. Select the camper(s) that made more than one reservations, display camper's id and name, average deposit (use alias), and # of reservations (use alias), sort the results by the number of reservations descending, meaning the camper with the highest number of reservations first.

[Include a screenshot showing your complete SQL Query and complete results returned]

5. Select the spot(s) that didn't get reserved by any camper, display spot id and name. **Must use a subquery.**