

Historical Background of the Database

I'm not sure about the details of the original database set up. I do know that Ellen Mutzl and others (Mary Williams?) created a spreadsheet in 2002 that indicated the years that members current at that time had been members. There was no record of mileage in the spreadsheet, and no record of earlier members who were no longer members in 2002.

Jane Hilaire took that information and created a MS Access database that tracked members, mileage and hikes. Summary spreadsheets of the first year of the database (2002-2003) are in the "Archives."* Jane knew MS Access inside and out and created a lovely database.

In December 2012, Jane was diagnosed with cancer. She knew that I had database experience so she had Damon Evans call me to see if I could possibly take it over. I had to say yes. She sent me the files and current paperwork. It didn't take long to figure it out, and she was able to answer my questions.

For the next several years, I kept up the MS Access database which required manually entering every hiker on every hike. That was essentially around 88,000 data entries a year. No one in their right mind would take that on, so a committee was formed (date?) to try to make it easier for hike leaders to enter the mileage. That committee was chaired by Phil Smith. I transferred the MS Access data to MySQL on the IHC server, and Bob Hackenberg did a fantastic job of writing scripts that made it possible for hike leaders and members to enter and view data. The first reports from the MySQL database were created in the 2018-2019 club year.

Now it is time for someone else to rework the system. I have done the set up the best I know how, but I am not a MySQL expert by any means. Perhaps someone(s) will be able to turn this into a more efficient system. I think the goal should be to keep reports simple—just because something is possible, it doesn't mean it's necessary. We are all volunteers with other commitments on our time.

*The database archives are on an external drive, but they are not very well organized. Someone might be able to make sense of them. There is also a full run of Trail Blazers (Mervyn has a copy of those). You'd think that a former librarian/digital archivist would do a better job keeping the archives in good order, but....it is what it is.

Karen Zimmerman, 8 November 2022

MILEAGE/MEMBERSHIP DATABASE ADMINISTRATOR

Special Characteristics Required for this Position

- a) **Must understand what relational databases are and how they work**
- b) **Must have computer access to the online database**
- c) **Must be able to create and maintain data tables and queries in a phpMyAdmin/MySQL environment or be able to migrate the database to a different system**
- d) **Must develop an understanding of Club policies in order to create appropriate reports.**

Duties Associated with Position

- a) **Collaborate with Pathfinder, Membership and Treasurer to maintain membership, mileage and hike schedule data for each active club member's participation throughout the Club calendar year (October-September)**
- b) **Monitor data entered by hike leaders, Membership and Treasurer, and resolve any data discrepancies**
- c) **Create a Mileage Awards Report for each 2 month period and provide to Membership for use in ordering Mileage badges and patches and to Pathfinder for inclusion in each schedule/newsletter**
- d) **Create annual Trail Blazer Report and Membership Roster for Publication to print and distribute in January**
- e) **Provide updated address labels for use by Publication in distribution of schedule, annual dues notice, annual banquet forms, Trail Blazer Report, and January Membership Roster and any other mailings that may be required**
- f) **Receive data from Conservation for Achievement Award assessment**
- g) **Provide Membership a list of members qualifying for yearly 25-Year award**
- h) **Provide Webmaster with data for updating awards and Club activities posted on the Club website**
- i) **Collaborate with Web Programmer [i.e. Bob Hackenberg] to integrate database functions via online access for appropriate members**
- j) **Backup all database tables (data and structures) regularly and use backup tables to test draft report queries**
- k) **Maintain detailed documentation of database structure, report queries and processes**

IHC Database Structure

(Note: YYYY = the club year being recorded. 2022 = club year Oct 1 2021 through Sept 30 2022)

There are two standard tools available through most web hosting services that are used to provide our online database and web interface. These are resident on our web server and are what we are using for our online data system. Most computer techies working on web development will be familiar with these tools.

MySQL : This is an open-source (i.e. free) *relational database management system* (RDMS). [Microsoft Access, which we were using previously, is another example of an RDMS.] “SQL” stands for *Structured Query Language* which is used for managing data held in an RDMS.

phpMyAdmin : This is a free software tool and is one of the most popular MySQL administration tools, especially for web hosting.

A relational database management system is different from Microsoft Excel and other spreadsheet software. An RDMS presents data in a series of tables. Each table consists of rows and columns. The user can manipulate the data using relational operators, in our case the SQL statements.

For example, our IHC database consists of three major tables (plus two other tables used for awards processing) which become linked by SQL statements.

1. “**MilesYYYY**” This table has three columns (plus one generated automatically as a unique counter). It contains a row for every hiker’s miles for every hike as they accumulate. This is the table that receives the miles entered through Bob’s mileage entry form.

The three columns are HikeID, MemberID and Miles (+the automatic counter column). By the end of the the 2020-2021 club year there were 15,030 rows of data in this table=a row for each hike by each hiker.

HikeID	MemberID	Miles	id
1	1494	5	46
1	175	5	47
1	545	3	49
1	544	3	50
1	413	3	51
4	214	6	52
4	1230	6	53
4	591	6	54
4	544	5	55
4	331	6	56
4	1145	6	57
4	336	4	58
4	233	6	59
4	1337	6	60
4	1338	6	61
4	491	6	62
4	52	6	63
14	369	6	64
14	368	6	65

2. **“HikeDetailYYYY”** is the table that holds all the detailed information about each hike. The columns for HikeDetail are:

- HikeID
- Day
- Tm (Time)
- Dt (Date)
- HikeName
- LeaderID = this field is the same as MemberID in the MembersYYYY table
- LeaderName = this field should match “PrintName” in the MembersYYYY table
- Members = number on hike [inserted from Bob’s code based on the Mileage Entry process]
- Cancelled (Yes/No)
- Comments
- Description
- (plus the automatic counter called “myunique” in this table).

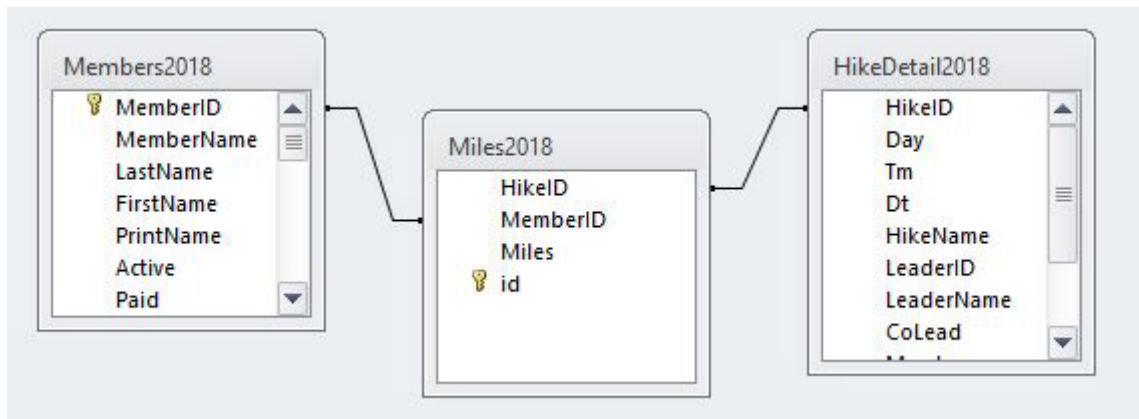
Notice the **HikeID** column. This is where our first relationship occurs. This column relates to the HikeID column in the Miles table. By using an SQL statement matching the HikeID column in Miles to the Hike ID column in HikeDetail, we now have a way to JOIN the two tables together.

3. **“MembersYYYY”** is the third table. It holds all the detailed information about the club members. This table is the most complex because it includes yes/no columns that help when preparing specific reports. It includes the following columns:

- **Member ID** = you’ll notice also relates to the Miles Table. This also serves at the unique key for this table
- **MemberName** = format is lastname, firstname
- **LastName**
- **FirstName**
- **PrintName** = somewhat redundant but useful for relating to the LeaderName in the HikeDetail table; format firstname lastname
- **Active** Yes/No = is the member currently active (remain active even if unpaid until Jan 1)
- **Paid** Yes/No = has the member paid for the new club year
- **MailTo** = Yes/No column = only one member of a family is checked so each family member does not receive the mailing (only one mailing per household)
- **MailToName** = either a single member name or joint names for families
- **Address** = street address
- **City**
- **State**
- **Zip**
- **Phone**
- **Email**
- **Reinstated** = is the member rejoining after an absence
- **YRJoinedLatest**
- **YRJoinedOrig**
- **YRSoff**
- **AccumYRSCurrent**
- **YR25Award** = yes/no Marked yes if member is due or has received this award
- **AccumMilesOct1** = used as a baseline for award calculations; updated during the EndOfYear processes

- **MilesYTD** = updated every two months for award list
- **TotMiles** = AccumMilesOct1 + MilesYTD
- **Award** = the latest award a member has received = used for award calculations
- **NoBadge** = yes/no = member has requested not to receive badges; see Comments2 field for limitations of the NoBadge request
- **Milestone** = yes/no = member has reached 10,000 miles
- **Publish** = yes/no = member agrees to have membership info published
- **Web** = schedule from web yes/no
- **Comments2** = for admin comments

Summary of the basic structural connections among the three main tables:



In addition, MemberID links to LeaderID and PrintName links to LeaderName.

The power of these tables lies in the SQL statements. Here's an example of a very simple SQL statement:

```
SELECT `MailToName`, `Address`, `City`, `State`, `Zip` FROM `Members2018` WHERE (`Paid`=1 AND `MailTo`=1 AND `Web`=0)
```

This statement creates the address label data for mailing schedules to those who don't get a copy from the website.

Statements like this one which are used frequently can be saved into "Views" to be run live at any time. This is like a separate table created from current data at the current moment. These views create data sets for various reports. An example of a View table used in creating the bimonthly Mileage Awards is "XXXXX"

Other Tables:

There are two other unchanging tables that are used for determining mileage awards. One is "Award Type" which simply defines the three award types: B = Badge; P=Patch; and R=Recognition. The other is "AwardsCriteria," which defines the mileage breaks for the minimum and maximum for each award. These are called upon every two months in SQL queries when selecting members who have met the requirements for a mileage award.

Another table used in creating the Mileage Awards list is UpdateAward. The content of this table is created during the MileageAwards process and changes each time the process is completed. Another table which changes bimonthly is NewSchedule which holds data about upcoming hikes entered by the

Pathfinders. An example of a View table used in creating the bimonthly Mileage Awards is **"MemberYYYYMilesYTD"**

Data is entered into the three main tables as follows;

- Hike Leaders enter the hikers' miles into MilesYYYY through a MileageEntry cgi code developed by Bob Hackenberg.
- MembersYYYY is updated by the Treasurer and the Membership person through a web-based Membership form also developed by Bob Hackenberg.
- HikeDetailYYYY relies on data entered into NewSchedule by the Pathfinders (Ed Wright and Jean Ballinger). This data is then used to update HikeDetailYYYY in schedule months.

Caveats:

It is significant to know that a change in the structure and/or data of one table might adversely affect the data in one or more tables and SQL statements, so caution is a word to the wise. Backup, backup, backup!

Automated processes can certainly make updating tables faster and easier, but it still takes a human to check for errors and make sure the results were accurate (e.g. importing the new hike schedules) and for troubleshooting problems. The Database Administrator needs to continually check for data consistency and needs to know the workings of the database thoroughly in order to handle troubleshooting when something goes wrong.

If there are several people working on the same tables, there has to be coordination to make sure one person doesn't overwrite in error what someone else has entered correctly. Communication is key.

Updated 16 Nov 2021 by Karen Zimmerman

MembersYYYY Table

- **Member ID** = This column relates to Member ID in the Miles Table. This also serves as the unique key for this table. **Each new member receives the next consecutive unique number.**
- **MemberName** = format is **lastname, firstname** so it will sort for reports. This should be consistent with the name used by the treasurer/membership persons.
- **LastName** = this column is convenient for searching for members, though it is somewhat redundant
- **FirstName** = this column is convenient for searching for members, though it is somewhat redundant
- **PrintName** = somewhat redundant but currently necessary for relating to the LeaderName in the HikeDetail table firstname lastname format
- **Active** = is the member currently active. Members continue to be "1" until January 1 and then switched to "0" if they have not yet paid. This field is a qualifier for legitimacy membership and used in several report queries.
- **Paid** = has the member paid for the new club year. This column is set to "0" for everyone September 1 in the new club year table.
- **MailTo** = yes/no . All individual members are "yes." Only one member of a family member is "yes" so the family receives only one copy of the mailing. This can also be used as a counter of how many memberships are individual, and how many are family.
- **MailToName** = Format of name(s) for mailing labels
- **Address** = street address with unit # if necessary
- **City**
- **State**
- **Zip**
- **Phone**
- **Email**
- ~~**Reinstated** = Has the member rejoined after an absence. Reinstatements will no longer be recorded after the 2021 membership year~~
- ~~**YRJoinedLatest** = This is the most recent club year a member has joined, and is significant for reinstated members~~
- **YRJoinedOrig** = This is the club year the member joined originally
- ~~**YRSoff** = This is used to calculate accumulated years for reinstated members~~
- **AccumYRSCurrent** = This is used to track years of membership. Current data for 25-year members are grandfathered in as is. New members receive 1 year for the current year they are joining. This field is carried over to the new club year and updated +1 when the member renews.
- **YR25Award** = yes/no = used to check members who've been members for 25 accumulated years. This column could be eliminated and calculated by a query at any time or set up as a View Table. It is not used for any other data merges.
- ~~**AccumMilesOct1** = used as the club beginning year mileage. Entered as "0" for new members. (and reinstated members if earlier miles not known.)~~
- **MilesYTD** = This field tracks member mileage for the current club year. It is currently updated in schedule months when mileage awards are determined. There is a query used to update the column.
- **TotMiles** = This is also updated for schedule months using a query that adds AccumMilesOct1 and MilesYTD columns. This field is carried over to the new club year as an update to the new year's AccumMilesOct1 column.
- **Award** = the latest award a member has received. This is necessary for having a base award for calculating the next mileage award a person is due.
- **NoBadge** = yes/no = There are a few members who have chosen not to receive any badges. 1 = No Badge; 0 = Yes Badge. For members who decide "No badge until X

Miles," I put a note in the comment field and check this column for "no." This means manually checking for the exceptions when creating the report.

- **Milestone** = yes/no. This indicates those who've reached 10,000 miles. This column could also be eliminated and calculated by a query at any time or set up as a View Table. It is not used for any other data merges.
- **Publish** = yes/no. This indicates members who do not want their data published. There have occasionally been people who don't want miles published, but want to be in the roster. This would be a note for the exception in the comments2 column which must be manually checked.
- **Web** = schedule from web yes/no. 1 for those who get the schedule online; 0 for those who want printed copies.
- **Comments2** = I used this to track exceptions and to remind myself when address, phone or email has been updated in order to sync with treasurer's list.
- **Hikes** = used at the end of the year for Trail Blazer report

Thanks to Bob Hackenberg's work, there is now a live web form that the membership chair and the treasurer use to add new members and edit member data.

Membership chair enters most of the data, and marks new members Active and Paid immediately following the board meetings. All three of us coordinate to check that we're all in agreement. Treasurer has her own set of records from current and past members. [NOTE: Marti & Kathy can decide who is marking members Active and Paid.]

This web form is password protected and only a few people have access. Here is an image of the main menu:



For the Database Process Manual 2022-01-21

HikeDetailYYYY Table

- **HikeID** = autoincrement key; numerical ID of each hike. Matches HikeID in the MilesYYYY table
- **Day** = day of the week in ddd format
- **Tm** = time of the hike in 24-hour clock format
- **HikeName** = title of the hike in all caps
- **LeaderID** = MemberID of the hike leader
- **LeaderName** = FirstName LastName of hike leader (should match PrintName in the MembersYYYY table)
- **Members** = automatically updates the number of hikers on a hike when the hike leader enters the mileage data
- **Comments** = not used much, but hike leader has the option of entering a comment for the database admin
- **Description** = the text description of the hike

This is now all updated by the "Import Hike Schedule" Process. There is also a web form that Ed can use to edit this table and to add more new hikes.

Karen Zimmerman April 1, 2021

MilesYYYY Table

- **HikeID** = this links to the HikeID in the HikeDetailYYYY table
- **MemberID** = this links to the MemberID in the MembersYYYY table
- **Miles** = collects miles per hiker from Bob's Mileage entry form
id = unique key

Karen Zimmerman April 1, 2021

Note added September 4, 2022: By the end of a membership year, there are usually over 15,000 lines of data. As of today, there are 14,661 lines of data recording each hiker's miles on each hike.

IHC Membership Year Database Management Calendar

October

New membership year begins. This is a **major Transition Process**, which begins in September of the previous club year. This process is documented in the Database Manual and should be coordinated with Bob Hackenberg and Ed Wright. This includes setting up the new year database table and finishing up reports for the previous year—achievement awards, website updates, etc. Some of these reports will be saved to be used in the **Trail Blazer** process.

Backup.

November

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Print mailing labels for the new schedule mailing at the time of the board meeting.

After the November board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup.

December

Start the **draft of the Trail Blazer** following the manual.

Either on New Year's Eve or early New Year's day, change all unpaid members' status to Inactive. This timing affects who will be able to earn mileage. Coordinate with Treasurer.

Backup.

January

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Shortly before the board meeting (coordinate with Membership and Treasurer) create the **Annual Membership Roster.**

Print mailing labels for the new schedule mailing at the time of the board meeting. Also print labels for mailing the Roster.

By the time of the board meeting, complete the **Trail Blazer.**

After the board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup.

March

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Print mailing labels for the new schedule mailing at the time of the board meeting.

After the board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup.

May

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Print mailing labels for the new schedule mailing at the time of the board meeting.

After the board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup.

July

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Print mailing labels for the new schedule mailing at the time of the board meeting.

After the board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup.

September

On the first day or two of the month, **Run Mileage Awards report** to send immediately to Membership, and to Pathfinder for the schedule and to the board.

Print mailing labels for the new schedule mailing at the time of the board meeting. Also print labels for mailing out dues notices and Awards Banquet invitations.

After the board meeting when Ed sends out the final copy of the new hike schedule, follow the **Upload New Schedule** process.

Backup and prepare for **End of Year Transition Process.**

Mileage Awards Process

Run this report in Schedule Months on the first or second day of the month: **November, January, March, May, July September**. It is timed so the Membership person has time to order badges to arrive before the board meeting.

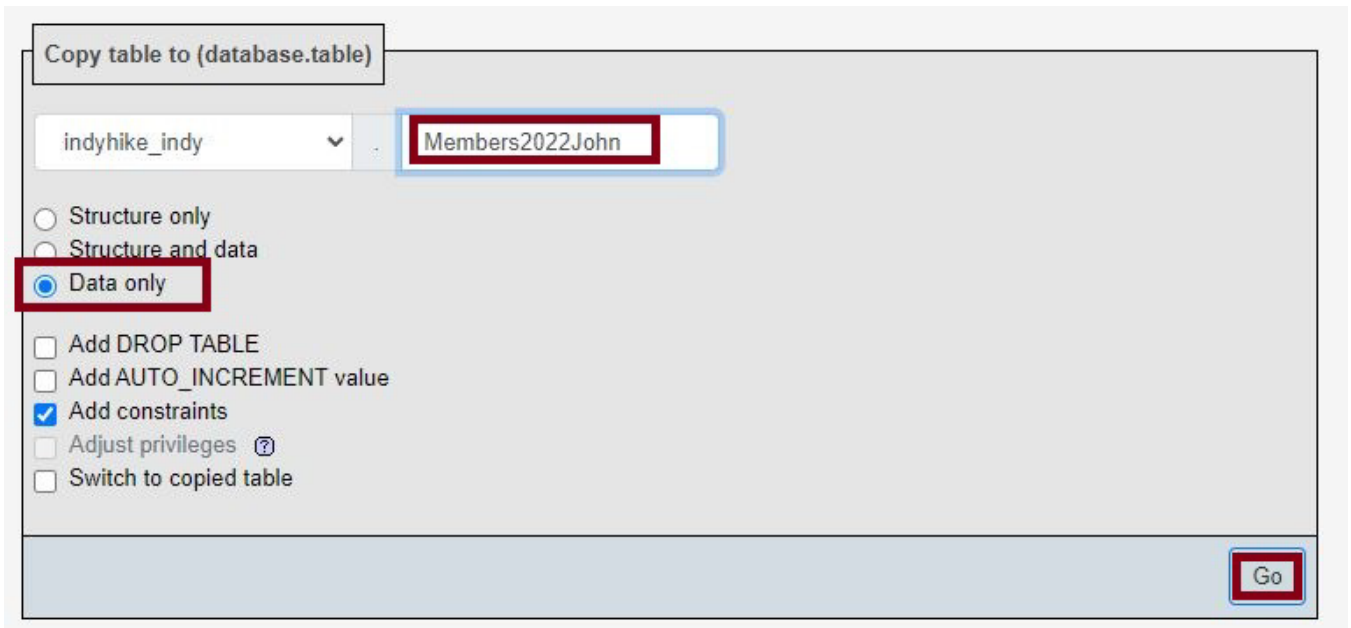
Use table Members2022John first as a trial.

1.

Truncate (clears data but keeps structure) Members2022John.
Go to the **Operations Tab** and scroll down to the Bottom section.



Then scroll to the Copy table section of the Operations menu list.



2.

Copy the mileage data from the view named **MemberMiles2022YTD** into the MilesYTD column in your Members2022John table.

```
UPDATE `Members2022John`  
LEFT JOIN `Member2022MilesYTD` ON `Members2022John`.`Member.ID` =  
`Member2022MilesYTD`.`MemberID`  
SET `Members2022John`.`MilesYTD` = `Member2022MilesYTD`.`MilesYTD`
```

3.

Add the current miles hiked year to date (YTD) to the members' beginning year (Oct 1) mileage thereby giving the member total lifetime mile accumulation.

UPDATE `Members2022John`

SET `TotMiles` = `AccumMilesOct1` + `MilesYTD`

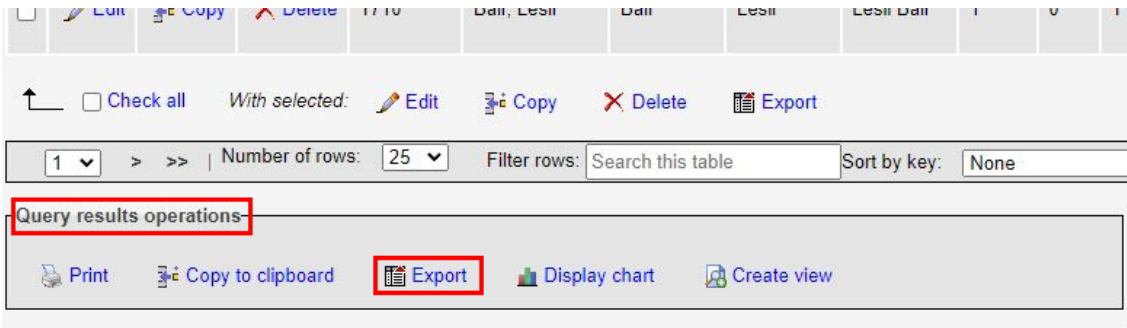
If steps 2 and 3 appear to work in Members2022John, repeat steps 2 & 3 in Members2022. Then continue in Members2022:

4.

Create the Awards List and export as a .csv file for creating reports.

```
SELECT `MemberID`, `MemberName`, `PrintName`, `Active`, `NoBadge`, `Award`, `TotMiles`, `Type`, `Min`
AS `AwardDue`, `Max`
FROM `Members2022`, `AwardsCriteria`
WHERE (`Award` < `Min` OR `Award` = '0') AND `TotMiles` >= `Min` AND `Active` = '1'
```

Export the query results as a .csv file to use in a spreadsheet for sorting and creating the lists.



In selecting and formatting the data, remember to keep track of the people who do not want badges. Check the comments area of those members to see if there are certain limits to their request. This is a sample format to use for the report emailed to the membership person. A few days before the board meeting, email a copy to all board members.

HC MILEAGE AWARDS	
January 16, 2018	
BADGES	
David Kincaid	24000
Martin Dadel	12000
Jeff Edmondson	11000
Kae Ramey	10000
Mike Khalil	8000
Priscilla Dick	4000
Mervyn Cohen	3000
Sue Brutkiewicz	2000
*NO BADGE	
PATCHES	
Linda McGlothlin	500
Jim Goulding	300
Mark Copher	200
Judie Hansen	200
Mary Ann Loehr	100
Robert Dickinson	100
RECOGNITION	
Glee Crowder	41500
Rick Kinnaman	29500
Cherie Voegel	25500
Julie Litten	14500
Pat Lawler	11500
Curtis Hinds	9500
John Gaebler	7500
Mary Bruss	6500
Jo Anne Starzyk	5500
Phil Smith	5500
Donna Peyton	3500

The format of the report to email to Ed for the Hike Schedule looks like this:

Glee Crowder	33500	Edeltraud Evans	4000	Donna Schuster	500
Allan Roberts	33000	Richard Evans	4000	Cathy Harms	500
Jill McFall	31000	Jim Jones	3500	Carole Canull	500
Cherie Voege	22000	Mike Khalil	3500	Mike Schultz	500
Michele Kestle	20500	Susanne McNeely	3500	Polly Lybrook	500
David Kincaid	19000	Judy Torrence	3000	Phil Coons	500
Ed Wright	17500	Reba McFarland	3000	Sue Brutkiewicz	500
Tish Brafford	16000	Mike Seeman	3000	Linda Crawford	400
Mary Ann Layman	12000	Jerry Green	2500	David Kollmeyer	400
Julie Litten	11500	Anne Heighway	2500	Linda B Griffin	400
Martin Dadel	10000	Jo Anne Starzyk	2500	Sheryl VanDeMark	400
Rita Bymaster	9500	Curtis Hinds	2500	Deb Barnes	400
Mary Williams	9500	Carole Gartner	2000	Sue Brutkiewicz	400
Nan Tate	8000	Marsha Webster	2000	Larry Hough	300
Joan Griffitts	7000	Marty Burton	2000	Janet Selby	300
Rory Wyss	6500	Phil Smith	2000	Barb Clisham	300
Donna Maurer	6500	Sally Sandman	1500	Tom Ladd	200
Sue Bullock	6000	Paula Chappell	1500	Ned Lewis	200
Kate Curtiss	6000	Mary K Johnson	1500	L Neal Percy	200
Mary Ladd	5000	Mick Parker	1500	Tim Killion	100
Terry Roesch	5000	Uschi Nicola	1000		

5.

Using the spreadsheet of the current Awards List, delete all columns except **MemberID**, **MemberName**, **AwardDue** and add an empty column called **AutoKey**. **If a member is due more than one award, delete the lower awards and keep only the highest award.** Example: member is due 200 and 300 mile patches, delete the 200 row and keep the 300 row. Save this as a .csv file.

Truncate the **UpdateAward** table if it still has previous award data in it. Then choose the **Import tab** and import the prepared .csv file.

Server: localhost » Database: indyhike_indy » Table: UpdateAward

Importing into the table "UpdateAward"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: Mileage Awards List.csv (Max: 20MiB)

You may also drag and drop a file on any page.

Character set of the file:

Partial import:

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. (This might be a good way to import large files, however it can break transactions.)

Skip this number of queries (for SQL) starting from the first one:

Other options:

Enable foreign key checks

Format:

Note: If the file contains multiple tables, they will be combined into one.

6.

Next update the Award column in the Members table. Using Members2022John again, run the following query:

```
UPDATE `Members2022John`  
LEFT JOIN `UpdateAward` on `Members2022John`.`MemberID`=`UpdateAward`.`MemberID`  
SET `Members2022John`.`Award`=`UpdateAward`.`AwardDue`  
WHERE `UpdateAward`.`AwardDue`>`Members2022John`.`Award`;
```

If it goes well, repeat the query in Members2022.

```
UPDATE `Members2022`  
LEFT JOIN `UpdateAward` ON `Members2022`.`MemberID` = `UpdateAward`.`MemberID`  
SET `Members2022`.`Award` = `UpdateAward`.`AwardDue`  
WHERE `UpdateAward`.`AwardDue`>`Members2022`.`Award`;
```

Updated 28 February 2022

Marti, Kathy and John:

You discovered some data errors! This is a confusing data field. If someone does NOT want a badge, the column should be marked 1 = "YES, this person does NOT want a badge"

So, Mary Bruss, David Kincaid, Rick Kinnaman and Joe Anne Starzyk should be 1 in this field.

Also, you will notice that Mary Bruss, David Kincaid, Rick Kinnaman and Jo Anne Starzyk have already passed their "No Badge until x miles" limitation, so they should be contacted to see what limitation they now want. I like the way Cherie Voege says "until 5000 milestones." Barb usually contacted the ones who passed their limitation.

MemberID	MemberName	NoBadge	Comments2	TotMiles
52	Brafford, Tish	1	No badge until 25,000	24627
73	Bruss, Mary	0	ph 11/18; no badge until 10,000	11193
80	Bullock, Sue	1	no badge until 10,000; email 11/17	7175
134	Crowder, Glee	1	no badge until 55000	52271
332	Khalil, Mike	1	No badge until 15000 miles	12030
334	Kincaid, David	0	no badge until 30,000	30473
337	Kinnaman, Rick	0	no badge until 35000	35615
657	Mutzi, Ron	1	email 7/16	12603
738	Roesch, Tom	1		4742
1605	Smith, Amanda	1		66
491	Starzyk, Jo Anne	0	no badge until 10000	10411
504	Sumner, Sandy	1		70
1418	Vetor, Cara	1	no roster; no patches	522
544	Voege, Cherie	1	no badge until 5000 milestones	31590
598	Zappia, Linda	1	email 10/17	5243

```
SELECT `MemberID`, `MemberName`, `NoBadge`, `Comments2`, TotMiles
FROM Members2022
WHERE `NoBadge`='1' OR `Comments2` LIKE '%badge%'
```

There are also several people who **never** want badges. In a few cases, there is no comment. It might be a good idea to add a comment "Never wants badges" or something similar just so you know for sure this is deliberate.

More about the Comments2 field: I used the field in the past to indicate when particular changes were made to phone numbers, addresses, etc. so I would remind myself (e.g. the note in Ron Mutzi dated 11/17). I didn't need to keep doing that in recent years, so you can ignore those comments or even delete them. Use the comments field for what you might want to track.

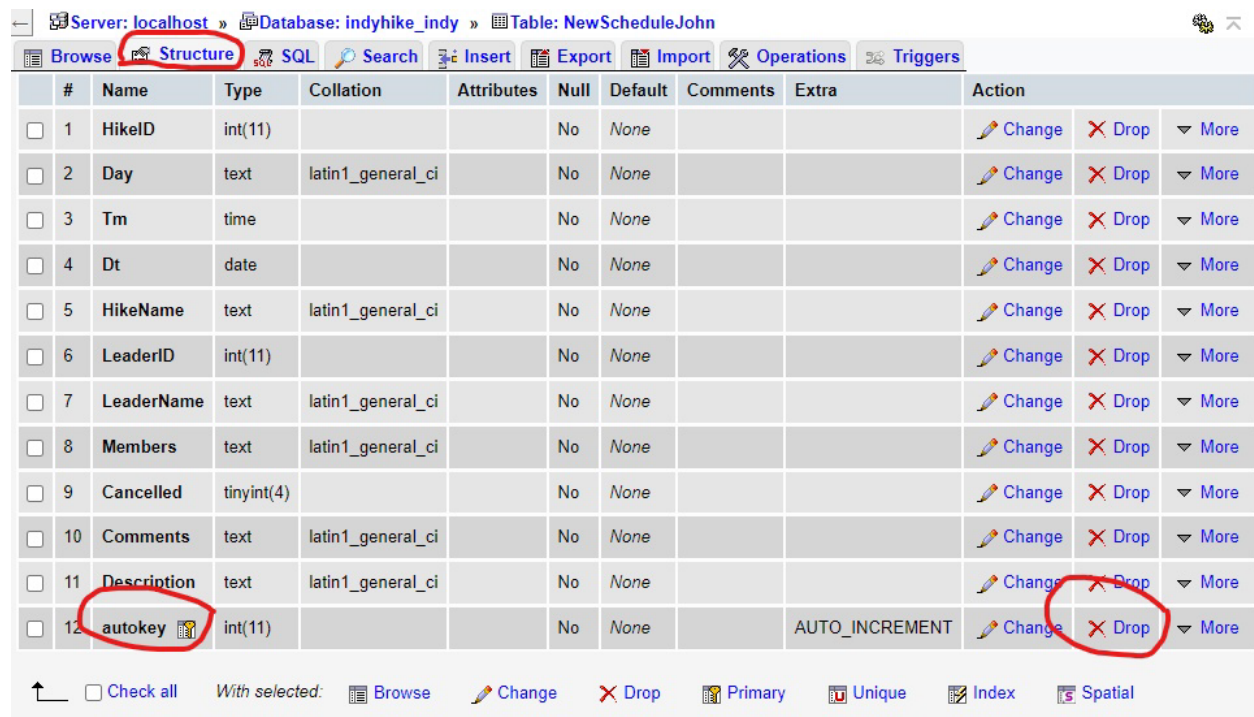
Process: Import New Hike Schedule

When Ed Wright uploads the PDF copy of the new Hike Schedule to the website, it's time to run the following process. This inserts the data of the new hike schedule into HikeDetailYYYY so it will show in the mileage entry calendar and in the hike schedule calendar on the website home page.

1 First Truncate HikeDetail2022John and copy data only from HikeDetail2022 to HikeDetail2022John

2 Truncate NewScheduleJohn and copy data only from NewSchedule to NewScheduleJohn

3 In NewScheduleJohn drop (delete) the autokey field



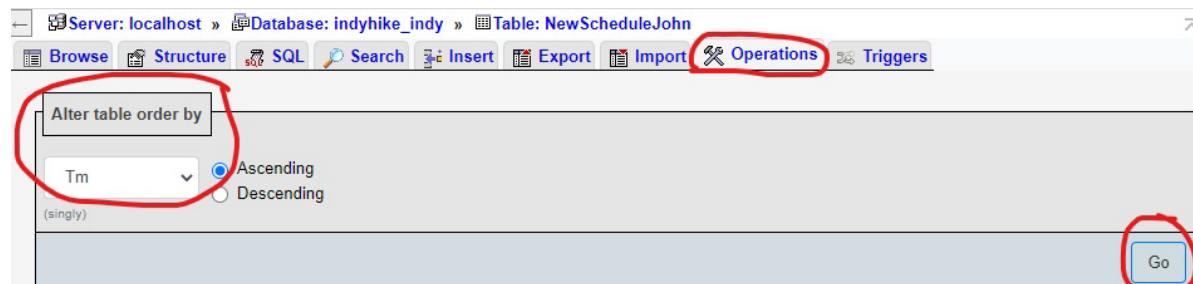
Server: localhost » Database: indyhike_indy » Table: NewScheduleJohn

Browse Structure SQL Search Insert Export Import Operations Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	HikeID	int(11)		No	None			Change Drop More
<input type="checkbox"/>	2	Day	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	3	Tm	time		No	None			Change Drop More
<input type="checkbox"/>	4	Dt	date		No	None			Change Drop More
<input type="checkbox"/>	5	HikeName	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	6	LeaderID	int(11)		No	None			Change Drop More
<input type="checkbox"/>	7	LeaderName	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	8	Members	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	9	Cancelled	tinyint(4)		No	None			Change Drop More
<input type="checkbox"/>	10	Comments	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	11	Description	text	latin1_general_ci	No	None			Change Drop More
<input type="checkbox"/>	12	autokey	int(11)		No	None		AUTO_INCREMENT	Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial

4 Sort NewScheduleJohn first by time and then by date.



Server: localhost » Database: indyhike_indy » Table: NewScheduleJohn

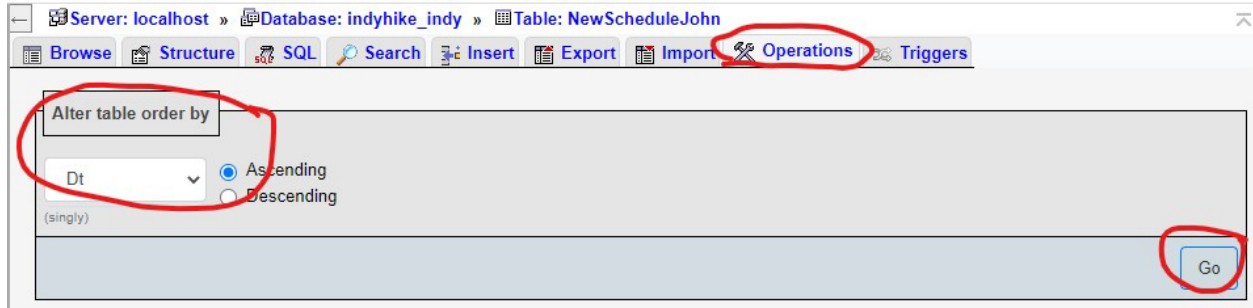
Browse Structure SQL Search Insert Export Import Operations Triggers

Alter table order by

Tm Ascending Descending

(singly)

Go



5 Run query in HikeDetail2020John (as a test run):

INSERT INTO `HikeDetail2022John` SELECT * FROM `NewScheduleJohn`

The HikeID is automatically added during the insert process. Scroll to the end of the browse pages to see that it worked properly. If so, repeat the insertion query into HikeDetail2022

6 INSERT INTO `HikeDetail2022` SELECT * FROM `NewScheduleJohn`

Check the online hike calendar on the website to see that the new hikes are listed, and check the Mileage Entry form to be sure the new hikes are listed. Notify Bob Hackenberg and Ed Wright that the schedule has been uploaded and have them check also.

7 TRUNCATE NewSchedule so it is ready for Ed and Jean to add the next schedule data

Process for Compiling The Trail Blazer

This report is produced annually. It used to be formatted by the Publications officer with data provided by the Mileage officer. Over time, the Mileage officer took on this task. It is to be prepared **ahead of the January board meeting and will be sent to all members with the updated Hike Schedule.**

* * *

Save a copy of the previous year's Word document with the new year's file name.

Begin by changing the Header to the correct year:

THE INDIANAPOLIS HIKING CLUB THE TRAIL BLAZER **2021 EDITION** PAGE 2

Edit the Volume/Edition and the Celebrating Years to the correct numbers:

THE TRAIL BLAZER

VOLUME **FIFTY-THREE – 2020 EDITION**

Celebrating **64** Years of Sponsoring Hikes and Outings

Retain the three paragraphs of text. Update the Officers to the list for the report year (not the list for the current year).

<u>ELECTED OFFICERS</u>	<u>APPOINTED OFFICERS</u>	<u>DIRECTORS</u>
PRESIDENT – Marti Burton VICE PRESIDENT – Mary Barbara Miller SECRETARY – Konnie Schlechte TREASURER – Kathy Whalen PATHFINDERS – Jean Ballinger & Ed Wright	CONSERVATION – John Bacone ELECTRONIC MEDIA – Phil Coons HISTORIAN – Mervyn Cohen MEMBERSHIP – Barb Strite MILEAGE – Karen Zimmerman PUBLICITY – Susan Sievers PUBLICATIONS – Kae Ramey SOCIAL – Pat Lawler WEBMASTER – Ed Wright	Mervyn Cohen Harold Crooks Bob Hackenberg Phil Smith

Retain the horizontal line and edit the Year in Review heading:

2020 YEAR IN REVIEW

Delete all the text in the Year In Review Section to the horizontal break for the Reports section. Copy Ed's new review text into this space.

Reports

Enter selected reports of interest into this space using examples in the previous year's reports.

To create the data for some of the common reports:

HIKE LEADERS - Those who led # or more hikes:

```
SELECT `LeaderName` AS `LeaderName`, COUNT(0) AS `COUNT(*)`  
FROM `HikeDetail2021`  
WHERE `HikeDetail2021`.`Cancelled` = '0' and `HikeDetail2021`.`Members` > '0'  
GROUP BY `HikeDetail2021`.`LeaderName`  
ORDER BY `COUNT(*)` desc
```

This has been saved as a View: CountLeaderHikes2021

Export to .csv file to use in Excel spreadsheet to choose the leaders with # or more hikes.

THE "MILE-A-DAY" HONOR ROLL - The honor roll recognizes members who have hiked 365 miles or more during the year—an average of at least a mile a day.

```
SELECT `PrintName`  
FROM `Members2021`  
WHERE `MilesYTD` >= '365'
```

MEMBERSHIP MILEAGE RECORD

For the year ending September 30, 2021 plus accumulated mileage since October 1, 1967

** New or reinstated member 2021*

We no longer track the number of hikes for each member in the MembersYYYY table, so that needs to be queried as follows:

First add a field `Hikes` to the MembersYYYY table. Use the structure tab and add the field at the end of the list. At this point, be sure to truncate your "John" members table and copy the live members table to "John."

Then set up a **View such as `2021MemberNumberHikes`** with the following query:

```
SELECT `Mileage2021`.`MemberID` AS `MemberID`, `Mileage2021`.`MemberName` AS  
`MemberName`, count(0) AS `COUNT(*)`  
FROM `Mileage2021`  
GROUP BY `Mileage2021`.`MemberID`
```

Next add the Number of Hikes to the Members in the MembersYYYY table: **(AS ALWAYS, USE YOUR "JOHN" TABLES FIRST TO TEST THE RESULTS OF THE UPDATE QUERY)**

```
UPDATE `Members2021`  
LEFT JOIN `2021MemberNumberHikes` ON `Members2021`.`MemberID`=  
`2021MemberNumberHikes`.`MemberID`
```

```
SET `Members2021`.`Hikes` = `2021MemberNumberHikes`.`COUNT(*)`
```

Then you can finally do the full list report as follows:

```
SELECT `MemberName`, `Hikes`, `MilesYTD`, `TotMiles`  
FROM `Members2021`  
ORDER BY `MemberName` ASC
```

Saved as View 2021TrailBlazerList

New Members are selected by the following query:

```
SELECT `MemberName`  
FROM `Members2021`  
WHERE `YRJoinedLatest`='2021'  
ORDER BY `MemberName` ASC
```

Saved as View AddedMembers2021

Send a copy of the New member List to the Membership Person and the Treasurer to make sure it's accurate. Then add an asterisk to the new members in the Trail Blazer Member Miles list.

NOTE: Several of these queries for 2021 have been saved as "Views" in the database. These views can be edited the following year by changing "2021" to "2022" in the View Structure options.

Updated December 28, 2021

Annual ROSTER (January)

SELECT `MemberName`, `Address`, `City`, `State`, `Zip`, `Phone` FROM `Members2023` ORDER BY `Members2023`.`MemberName` ASC

[Created View "2023 Member Roster"]

Be sure to check for the "No Publish" people in the Members Table and drop them from the list.

Change years on front page and on headers for other pages.

Change names of officers to the 2022 list.



The Indianapolis Hiking Club

www.indyhike.org

Happiness - A Step at a Time



ROSTER OF ACTIVE MEMBERS

January 2021

ELECTED OFFICERS

PRESIDENT – Mary Barbara Miller
VICE PRESIDENT – Lori Showley
SECRETARY – Konnie Schlechte
TREASURER – Kathy Whalen
PATHFINDERS – Jean Ballinger
& Ed Wright

APPOINTED OFFICERS

CONSERVATION – Marti Burton
ELECTRONIC MEDIA – Phil Coons
HISTORIAN – Mervyn Cohen
MEMBERSHIP – Barb Strite
MILEAGE – Karen Zimmerman
PUBLICITY – Susan Sievers
PUBLICATIONS – Kae Ramey
SOCIAL – Pat Lawler
WEBMASTER – Ed Wright

DIRECTORS

Harold Crooks
Ann Heighway
Phil Smith
Rick Wortman

The use of the information in this directory for commercial or personal solicitations and the conversion of directory information to digital media is strictly prohibited.

Name	Address	City	State	Zip	Phone
Abbott, Tony	7410 N Jewel Ln	Indianapolis	IN	46250	317-261-0900
Aber, Cheryl	947 Silver Creek Way	Greenwood	IN	46113	317-883-2287