

Monitoring Analysis Steps

Tables are orange Columns are blue SQL commands are in pink

Table of Contents

[Table of Contents](#)

Why not store the data as one spreadsheet?

[I can calculate the relative abundance using this formula:](#)

[I can import A3 as a csv file into postgres like this:](#)

[Find the average hits per quadrat at each plot using this code:](#)

[Analyzing an individual plant species](#)

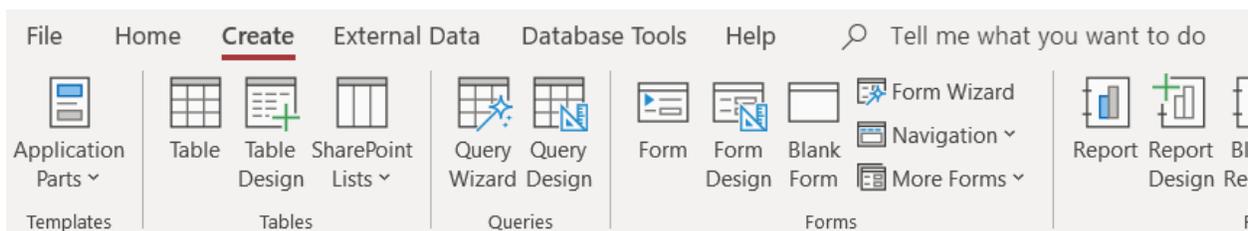
[Select all species in a season and plot](#)

[Analyzing average hits per quadrat at each plot](#)

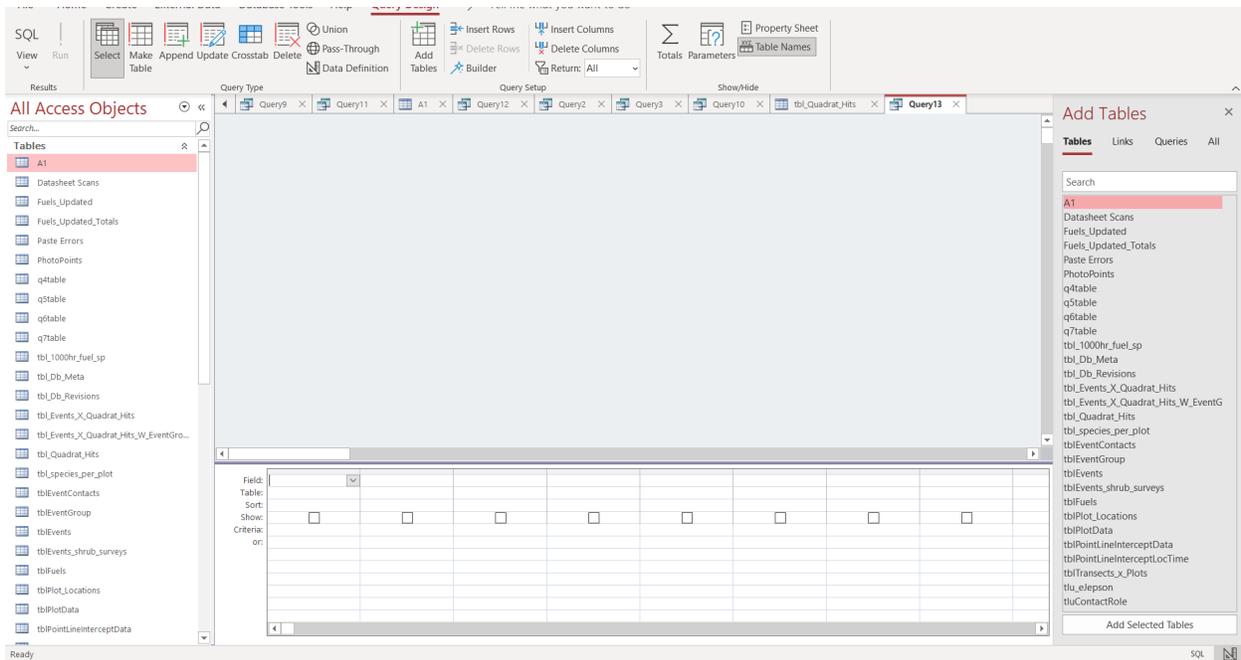
[Analyzing which species stay the same between two species lists](#)

[Analyzing the difference between two species lists](#)

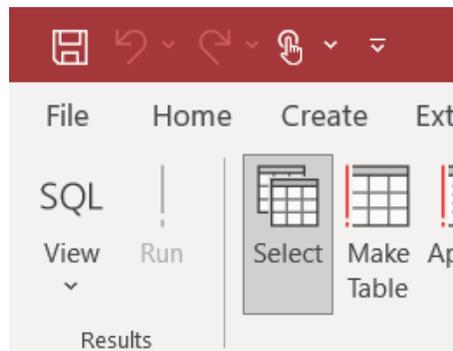
First, you will need to enter the SQL edit mode using Microsoft Access. To do this, go to the **Create** tab and click on the icon that says **Query Design**. This icon will have a tabular grid, and a couple little rulers in the bottom left.



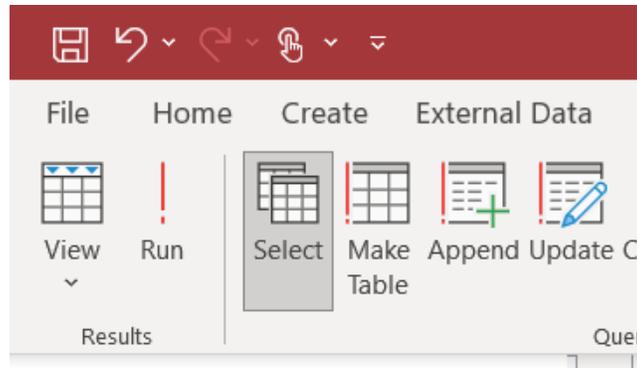
Microsoft Access will first show you the visual query design maker.



On the top left, click "SQL view".



We will be using the INTO command to create new tables with combined data from multiple tables. You can preview the result with "View", and you can actually create the new table using "Run."



This first command is creating the first analysis table **A1**, which is a new analysis table that is created by attaching (*using an SQL JOIN*) the Jepson Species information to table containing information about the number of hits per plant in each plot.

The join is connected using the **species ID**, meaning the species code stored in the **tbl_Quadrat_Hits** table is also stored on the lookup table **tlu_eJepson**.

```
SELECT
tbl_Quadrat_Hits.EventID,
tbl_Quadrat_Hits.Dead,
tbl_Quadrat_Hits.Unrooted,
tbl_Quadrat_Hits.HitsinQuadrat,
tlu_eJepson.family,
tlu_eJepson.genus,
tlu_eJepson.species,
tlu_eJepson.nativity

INTO
A1

FROM
tbl_Quadrat_Hits

LEFT JOIN
tlu_eJepson

ON
tlu_eJepson.PlantSpeciesID = tbl_Quadrat_Hits.SpeciesID;
```

After you hit "Run" you should see a pop-up asking if you are sure, then when you approve, you will see the new table visible in the list of tables (All Access Objects) on the left-hand side of the screen.

Here is my table A1:

EventID	Dead	Unrooted	HitsInQuadra	family	genus	species	nativity
7	<input type="checkbox"/>	<input type="checkbox"/>		0 BARE GROUND	BARE GROUND	BARE GROUND	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 LICHEN	LICHEN	LICHEN	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 LITTER	LITTER	LITTER	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 MOSS	MOSS	MOSS	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 ROCK	ROCK	ROCK	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 WATER	WATER	WATER	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 WOOD	WOOD	WOOD	NOTPLANT
7	<input type="checkbox"/>	<input type="checkbox"/>		0 POACEAE	Briza	maxima	NATURALIZED
7	<input type="checkbox"/>	<input type="checkbox"/>		0 POACEAE	Brachypodium	distachyon	NATURALIZED
7	<input type="checkbox"/>	<input type="checkbox"/>		0 POACEAE	UNKNOWN POA	UNKNOWN POA	UNKNOWN
7	<input type="checkbox"/>	<input type="checkbox"/>		0 POACEAE	UNKNOWN POA	UNKNOWN POA	UNKNOWN

Why not store the data as one spreadsheet?



Storing the actual name and citation info about each species in the `tlu_eJepson` table will make it easier for future updates to be made to the species list. Because species are collected in multiple types of surveys, there are many tables that use the species data. If there is a change, storing the species data as a code in each of the individual tables and storing the actual data in a lookup table means that if there is an update, the species data do not have to be repeatedly changed in multiple tables.

The next command brings in information from `tblEvents`, to form the next temporary analysis table `A2`. This table will look similar to `A1`, but also include the `transect`, `plot ID`, and `Event Group ID`.

```
SELECT
tblEvents.transect,
tblEvents.LocationID,
tblEvents.EventGroup,
A1.Dead,
A1.Unrooted,
A1.HitsInQuadrat,
A1.family,
A1.genus,
A1.species,
A1.nativity

INTO
A2

FROM
A1

LEFT JOIN
tblEvents

ON
A1.EventID = tblEvents.EventID;
```

Here is my table A2:

transect	LocationID	EventGroup	Dead	Unrooted	HitsInQuadrat	family	genus	species	nativity
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	0	ASTERACEAE	Hypochaeris	radicata	NATURALIZED
C	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	0	POACEAE	Briza	maxima	NATURALIZED
C	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	14	LITTER	LITTER	LITTER	NOTPLANT
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	2	BARE GROUND	BARE GROUND	BARE GROUND	NOTPLANT
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	2	LITTER	LITTER	LITTER	NOTPLANT
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	9	POACEAE	Briza	maxima	NATURALIZED
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	12	POACEAE	Rytidosperma	penicillatum	NATURALIZED
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	6	PLANTAGINACE	Plantago	lanceolata	NATURALIZED
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	14	CAPRIFOLIACEA	Lonicera	hispidula	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	5	ROSACEAE	Fragaria	vesca	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	1	AGAVACEAE	Chlorogalum	pomeridianum	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	5	ASTERACEAE	Baccharis	pilularis	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	1	POACEAE	Avena	barbata	NATURALIZED
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	8	POACEAE	Danthonia	californica	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	0	POACEAE	Bromus	vulgaris	NATIVE
B	MGP_03_POL	2	<input type="checkbox"/>	<input type="checkbox"/>	7	POACEAE	POACEAE	UNKNOWN POA	UNKNOWN

There is still one column with unhelpful data. The [EventGroup](#) column has numbers that reference the table [tblEventGroup](#), where data about each monitoring season is stored:

EventGroupID	StartDate	EndDate	Protocol ver	EventGroupName	EventGroupDesc	EventGroupNotes	EventGroupReport	Click to A
1	5/22/2020	6/5/2020	trial Protocol	Spring_2020	Trial season, in sprin	This season is missing		
2	4/30/2021	5/25/2021	version 1 Prot	Spring_2021	Spring 2021 monitori			
(New)								

A third join is needed to show the [EventGroupName](#) in the table.

```

SELECT
tblEventGroup.EventGroupName,
A2.transect,
A2.LocationID,
A2.Dead,
A2.Unrooted,
A2.HitsInQuadrat,
A2.family,
A2.genus,
A2.species,
A2.nativity

INTO
A3

FROM
A2

LEFT JOIN
tblEventGroup

```

```
ON
A2.EventGroup = tblEventGroup.EventGroupID;
```

Here is my table A3:

EventGroupName	transect	LocationID	Dead	Unrooted	HitsInQuadra	family	genus	species	nativity
Spring_2021	C	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>		0 ASTERACEAE	Hypochaeris	radicata	NATURALIZED
Spring_2021	C	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>		0 POACEAE	Briza	maxima	NATURALIZED
Spring_2021	C	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	14	LITTER	LITTER	LITTER	NOTPLANT
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	2	BARE GROUND	BARE GROUND	BARE GROUND	NOTPLANT
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	2	LITTER	LITTER	LITTER	NOTPLANT
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	9	POACEAE	Briza	maxima	NATURALIZED
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	12	POACEAE	Rytidosperma	penicillatum	NATURALIZED
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	6	PLANTAGINACE	Plantago	lanceolata	NATURALIZED
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	14	CAPRIFOLIACEA	Lonicera	hispidula	NATIVE
Spring_2021	B	MGP_03_POL	<input type="checkbox"/>	<input type="checkbox"/>	5	ROSACEAE	Fragaria	vesca	NATIVE

This table can now be analyzed using filters, sorts, and aggregate functions.

I want to create lists comparing the species present at each spot, from year to year, and comparing the relative abundance.

I can calculate the relative abundance using this formula:

$$RA = TS / TP * 100$$

- Where RA is the relative abundance of species (%)
- TS is the total number of species in an area
- TP is the total sum of the populations of all species in the area

<https://calculator.academy/relative-abundance-calculator/>

<https://www.biology-fieldwork.org/a-level/fieldwork-techniques/vegetation-sampling/using-quadrats/>

I can import A3 as a csv file into postgres like this:

<https://www.postgresqltutorial.com/import-csv-file-into-posgresql-table/>

I use postgres for complex SQL queries because it will clearly print out exactly where there is an error, if the code does not work.

Find the average hits per quadrat at each plot using this code:

```
select genus, species, nativity, (sum(hitsinquadrat)/3) as plantavg from tbl_A3
where eventgroupname = 'Spring_2020'
and locationid= 'MGP_01_HWY'
group by genus, species, nativity
order by plantavg desc;
```

My results:

	genus character varying	species character varying	nativity character varying	plantavg bigint
1	Brachypodium	distachyon	NATURALIZED	29
2	UNKNOWN POACEAE	UNKNOWN POAC...	UNKNOWN	19
3	Plantago	lanceolata	NATURALIZED	11
4	UNKNOWN MONOCOT	UNKNOWN MONO...	UNKNOWN	10
5	LITTER	LITTER	NOTPLANT	9
6	Briza	maxima	NATURALIZED	4
7	Hypochaeris	radicata	NATURALIZED	3
8	Baccharis	pilularis	NATIVE	3
9	UNKNOWN FLOX	UNKNOWN FLOX	UNKNOWN	2
10	Plantago	erecta	NATIVE	2
11	Danthonia	californica	NATIVE	1

Change the `eventgroupname` to assess a different year, and change the `locationid` to assess a different location.

Analyzing an individual plant species

```
select locationid, eventgroupname, (sum(hitsinquadrat)/3) as hits
from tbl_A3
where genus = 'Brachypodium'
and species = 'distachyon'
group by locationid, eventgroupname
```

```
order by eventgroupname
;
```

This returns the average hits at each quadrat per plot for each year, per species.

	locationid character varying (50)	eventgroupname character varying (50)	hits bigint
1	MGP_01_HWY	Spring_2020	29
2	MGP_02_OLH	Spring_2020	39
3	MGP_03_POL	Spring_2020	24
4	MGP_01_HWY	Spring_2021	39
5	MGP_03_POL	Spring_2021	18

- In order to make it easier to analyze data by species, I need to fix the concatenation issue in my spreadsheet. Currently, unknown plant species data aren't stored in the very useful name_minus_authors column in [tbl_eJepson](#). This will also allow me to analyze data by subspecies, and by individual "unknowns" rather than the currently very unhelpful and identical UNKNOWN POACEAEs floating around.

Select all species in a season and plot

```
select distinct genus, species, nativity from tbl_A3
where eventgroupname = 'Spring_2021'
and locationid= 'MGP_02_OLH'
order by nativity asc;
```

Select **genus**, **species**, and **nativity** from the desired season and plot, and they will be displayed in group according to their nativity status.

Analyzing average hits per quadrat at each plot

```
select nativity, eventgroupname, locationid, (sum(hitsinquadrat)/3) as hits
from tbl_A3
where eventgroupname = 'Spring_2020'
and locationid = 'MGP_01_HWY'
```

```
group by nativity, eventgroupname, locationid
order by eventgroupname, locationid
;
```

	nativity character varying	eventgroupname character varying (50)	locationid character varying (50)	hits bigint
1	NATURALIZED	Spring_2020	MGP_01_HWY	52
2	NOTPLANT	Spring_2020	MGP_01_HWY	9
3	UNKNOWN	Spring_2020	MGP_01_HWY	34
4	NATIVE	Spring_2020	MGP_01_HWY	8

Analyzing which species stay the same between two species lists

```
with table2020 as (
select genus, species from tbl_a3
where eventgroupname = 'Spring_2020'
and locationid = 'MGP_01_HWY'
),

table2021 as(
select genus, species from tbl_a3
where eventgroupname = 'Spring_2021'
and locationid = 'MGP_01_HWY'
)

select distinct table2021.genus, table2021.species from table2021
inner join table2020
on table2021.genus = table2020.genus and table2021.species = table2020.species;
```

Using the **WITH** command, create two temporary tables for the two species lists. Then do an **inner join** on the two tables.

Analyzing the difference between two species lists

```
with table2020 as (
select genus, species from tbl_a3
where eventgroupname = 'Spring_2020'
and locationid = 'MGP_01_HWY'
),
```

```

table2021 as (
select genus, species from tbl_a3
where eventgroupname = 'Spring_2021'
and locationid = 'MGP_01_HWY'
)

SELECT DISTINCT table2021.genus, table2021.species
FROM   table2021
      LEFT JOIN
      table2020 ON table2020.species=table2021.species and
table2020.genus=table2021.genus
WHERE  table2020.species IS NULL;

```

Using the **with** command, create two temporary tables for the two species list, as above. This time, do a **left join** and return the NULL values from this join. The left join will join two tables, and any table on the righthand table that does not match values in the lefthand table will be returned as NULL. This returns the new species found in 2021 (where table2020.species is NULL on the join).

Review this code, because I don't entirely understand how it works. I found it on this post:

Compare two tables and give the output record which does not exist in 1st table

I want an SQL code which should perform the task of data scrubbing. I have two tables both contain some names I want to compare them and list out only those name which are in table 2 but not in ta...

 <https://stackoverflow.com/questions/18649500/compare-two-tables-and-give-the-output-record-which-does-not-exist-in-1st-table/18649525>

