Hunting for Trends in Data Using Microsoft Access

Jennifer LaFleur
The Dallas Morning News

When Bob Imrie, an Associated Press reporter in Wausau, WI, went searching for information about hunting accidents, he ended up right on target. The state filed hard-copy reports on every accident detailing information such as weather, topography and the number of pieces of orange the victim was wearing.

Imrie had no way of digging out trends from the voluminous hard-copy reports and he had never used a computer database. But with some help over the phone from some nerdy colleagues, he created his own database to look at trends in hunting accidents in Wisconsin.

Here's what his database ended up looking like:

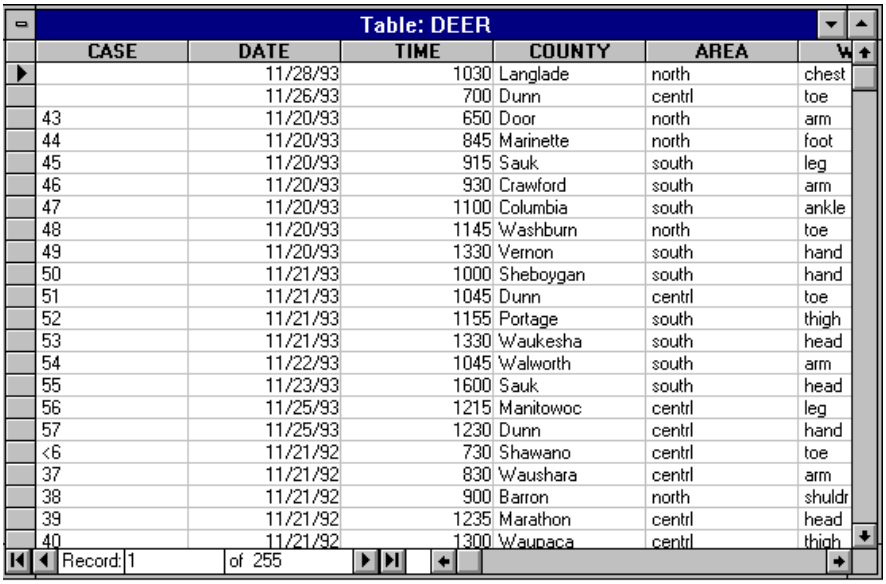


Table: DEER

CASE	DATE	TIME	COUNTY	AREA	W
	11/28/93	1030	Langlade	north	chest
	11/26/93	700	Dunn	centrl	toe
43	11/20/93	650	Door	north	arm
44	11/20/93	845	Marinette	north	foot
45	11/20/93	915	Sauk	south	leg
46	11/20/93	930	Crawford	south	arm
47	11/20/93	1100	Columbia	south	ankle
48	11/20/93	1145	Washburn	north	toe
49	11/20/93	1330	Vernon	south	hand
50	11/21/93	1000	Sheboygan	south	hand
51	11/21/93	1045	Dunn	centrl	toe
52	11/21/93	1155	Portage	south	thigh
53	11/21/93	1330	Waukesha	south	head
54	11/22/93	1045	Walworth	south	arm
55	11/23/93	1600	Sauk	south	head
56	11/25/93	1215	Manitowoc	centrl	leg
57	11/25/93	1230	Dunn	centrl	hand
<6	11/21/92	730	Shawano	centrl	toe
37	11/21/92	830	Waushara	centrl	arm
38	11/21/92	900	Barron	north	shuldr
39	11/21/92	1235	Marathon	centrl	head
40	11/21/92	1300	Waupaca	centrl	thigh

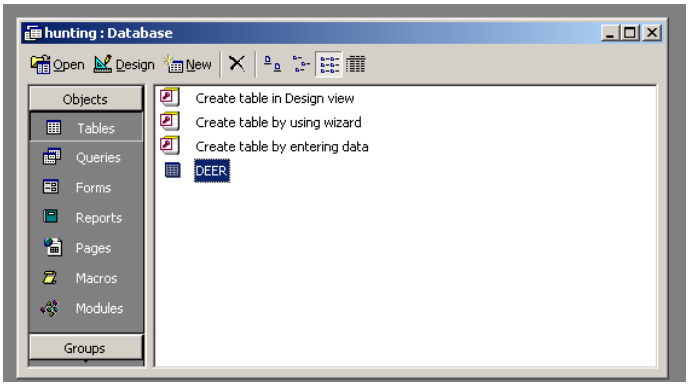
Record: 1 of 255

The record layout, which is the map to any database, is below. It shows the fields Imrie created in the database, whether they were characters or numbers and the width of each field:

#	Name	Type	Width	Description
1	CASE	Character	10	An identifier he created

2	DATE	Date		Date of accident
3	TIME	Number	4	Time of accident
4	COUNTY	Character	11	County of accident
5	AREA	Character	6	Region of the state (he created this)
6	WOUND	Character	6	Part of body wounded
7	INJURY	Character	5	Severity of injury: minor, major or fatal
8	TYPE	Character	2	si = self-inflicted ; sp = second person
9	CAUSE	Character	30	Description of cause
10	SAGE	Number	2	Shooter's age
11	VAGE	Number	2	Victim's age
12	FIREARM	Character	7	Type of firearm
13	FACTION	Character	6	Faction of gun
14	ALCOHOL	Character	3	Alcohol involved
15	ALCOLEV	Number	5	Blood alcohol level
16	WEATHER	Character	11	Weather conditions
17	TOPOGRO	Character	10	Topography
18	SEXPER	Number	2	Shooter's years of experience
19	VEXPER	Number	2	Victim's years of experience
20	SGRADUATE	Character	3	Shooter graduate of safety school?
21	VGRADUATE	Character	3	Victim graduate of safety school?
22	SSEX	Character	1	Shooter sex
23	VSEX	Character	1	Victim sex
24	GUNBRND	Character	13	Brand of gun
25	GUNGUAGE	Character	10	Gauge of gun
26	TEMP	Number	2	Temperature
27	MUZDIS	Number	5	Muzzle distance from victim
28	LAND	Character	3	pub = public land; pri = private land
29	VORANGE	Number	1	# pieces of orange worn by victim
30	GUNSIGHT	Character	5	Open, scope or other
31	VACTIVITY	Character	10	Victim's activity
32	LOCATION	Character	6	Location of victim
33	PRECIP	Character	3	Precipitation: Yes or blank

To bring up the data yourself, go to FILE/OPEN and in the data directory, open a file called Hunting.mdb.

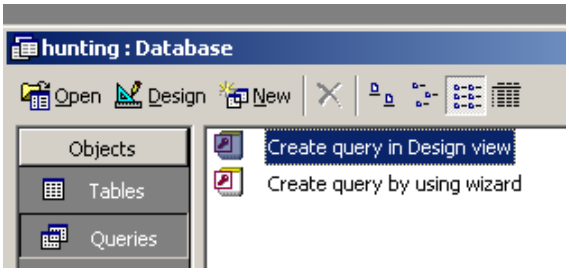


In Access a database may contain several individual data sets called “tables.” To open the Deer table, select it and click OPEN. Access will open the database to a browse window.

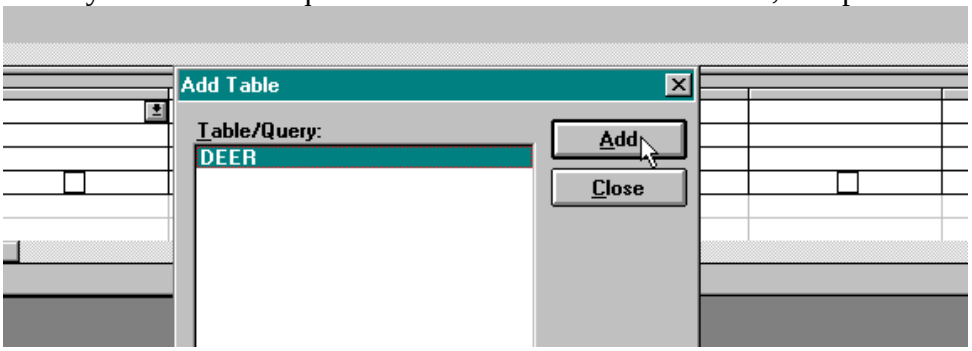
Interview your data.

Just as you do with people, you must get to know your data. Study it, look for patterns, look for special codes, upper case, lower case: all this will help when it’s time to ask questions.

Just looking at this database triggers many questions. But let’s start off with some basic queries. To bring up the query form, click on the Query tab in the main database window, and press the New button. You’ll then see the following window:

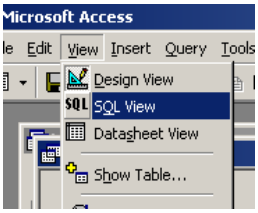


Click New Query, then select design view. Access will bring you to a new window, and ask which tables you want to ask questions of. Choose the DEER table, and press the Add button ONCE!



Then press the Close button.

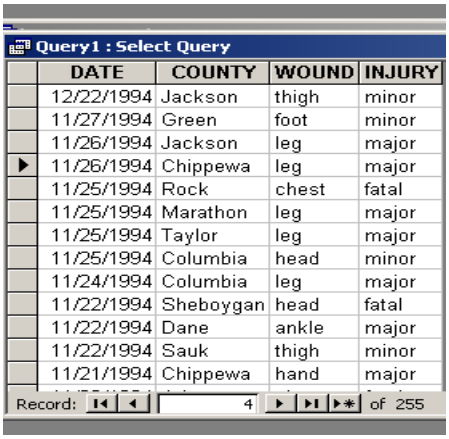
You now find yourself in Design View, or the grid. If you want to write queries in SQL, simply select "View", then "SQL". In Access, you can easily switch back and forth from the Design, SQL and Table views.



WARM-UP

1. Let's practice doing a query to show the following fields: DATE, COUNTY, WOUND, INJURY. Placing these fields in the SELECT line will guarantee that they will appear in the query results. Also, sort your results chronologically.

SELECT DATE, COUNTY, WOUND, INJURY
FROM DEER
ORDER BY DATE DESC


Query1 : Select Query

DATE	COUNTY	WOUND	INJURY
12/22/1994	Jackson	thigh	minor
11/27/1994	Green	foot	minor
11/26/1994	Jackson	leg	major
11/26/1994	Chippewa	leg	major
11/25/1994	Rock	chest	fatal
11/25/1994	Marathon	leg	major
11/25/1994	Taylor	leg	major
11/25/1994	Columbia	head	minor
11/24/1994	Columbia	leg	major
11/22/1994	Sheboygan	head	fatal
11/22/1994	Dane	ankle	major
11/22/1994	Sauk	thigh	minor
11/21/1994	Chippewa	hand	major

Record: 4 of 255

1. **Limit the Records**

Say you didn't really care about all the deer hunting accidents in Wisconsin, you just wanted to see them for Jackson County. That's where the CRITERIA line in design view and the WHERE line of SQL view come in handy.

SELECT date, county, wound, injury
FROM deer
WHERE county="JACKSON"
ORDER by date desc

Query1 : Select Query

date	county	wound	injury
12/22/1994	Jackson	thigh	minor
11/26/1994	Jackson	leg	major
11/30/1991	Jackson	foot	major
11/25/1991	Jackson	chest	major
11/24/1991	Jackson	abdomn	major
11/23/1990	Jackson	arm	major
11/26/1988	Jackson	foot	major
11/27/1987	Jackson	thigh	minor

If you're screening your data, but you don't know the whole word in a field, Access will let you estimate using a *, which is called a "wildcard." A wildcard stands for any combination of letters or numbers.

For example, this filter would pull out all causes beginning with "careless":
SELECT date, county, wound, injury
FROM deer
WHERE cause like "CARELESS*"

This would pull out all causes containing the word wife: *WIFE*

2. **Let's get Boolean**

Sometimes you want more from a selection criteria than just one county. You might want the records for Jackson and Door counties. That's where OR comes in handy. Type OR between each item.
SELECT date, county, wound, injury
FROM deer
WHERE county="JACKSON" or county="DOOR"
ORDER BY date desc

date	county	wound	injury
12/22/1994	Jackson	thigh	minor
11/26/1994	Jackson	leg	major
11/20/1993	Door	arm	minor
11/29/1992	Door	foot	major
11/30/1991	Jackson	foot	major
11/25/1991	Jackson	chest	major
11/24/1991	Jackson	abdomn	major
11/23/1990	Jackson	arm	major
11/19/1989	Door	foot	major
11/26/1988	Jackson	foot	major
11/27/1987	Jackson	thigh	minor
11/22/1987	Door	hand	major

Record: 1 of 12

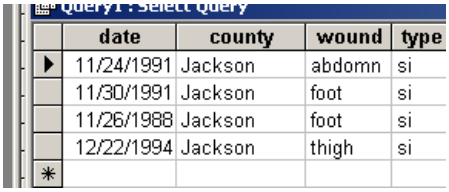
You told Access you wanted all the records where the COUNTY was either Jackson or Door. This might be useful if you covered a multiple county area. Using the OR to connect criteria is called Boolean logic. Another Boolean word is AND. AND is more restrictive than OR: with AND both lines must be true in order for a record to be selected. We could use the AND if we wanted all causes containing both the word wife AND careless.

SELECT date, county, wound, injury
FROM deer
WHERE cause like "*WIFE*" and cause like "*CARELESS*"

date	county	wound	injury	cause
11/18/1990	Trempealeau	foot	major	careless handling-wife involed

You can also filter on two fields at the same time. For example, we might want just self-inflicted accidents from Jackson County:

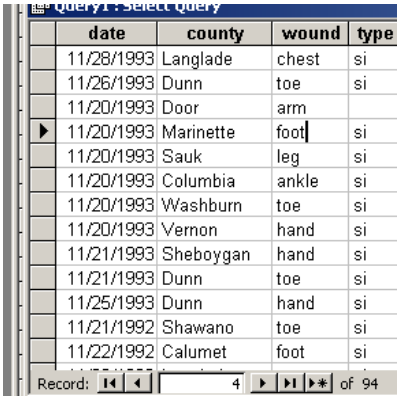
SELECT date, county, wound, type
FROM deer
WHERE county = "JACKSON" and type = "SI"



date	county	wound	type
11/24/1991	Jackson	abdomn	si
11/30/1991	Jackson	foot	si
11/26/1988	Jackson	foot	si
12/22/1994	Jackson	thigh	si

If you wanted Jackson or Door county, OR self-inflicted wounds, you would change the query to this:

SELECT date, county, wound, type
FROM deer
Where county = "JACKSON" or county = "DOOR" or type = "SI"



date	county	wound	type
11/28/1993	Langlade	chest	si
11/26/1993	Dunn	toe	si
11/20/1993	Door	arm	
11/20/1993	Marinette	foot	si
11/20/1993	Sauk	leg	si
11/20/1993	Columbia	ankle	si
11/20/1993	Washburn	toe	si
11/20/1993	Vernon	hand	si
11/21/1993	Sheboygan	hand	si
11/21/1993	Dunn	toe	si
11/25/1993	Dunn	hand	si
11/21/1992	Shawano	toe	si
11/22/1992	Calumet	foot	si

Record: 4 of 94

Notice that many more records are returned when using OR statements than AND statements.

PRACTICE EXERCISE # 1 – HUNTING DATABASE

1. Do a query to display all of the injuries occurring from the ankle down. (Hint: Here's a list of all of the body parts: *abdomen, ankle, arm, back, butt, chest, ear, eye, face, finger, foot, groin, hand, head, hip, knee, leg, neck, scrotum, shoulder, stomach, thigh, thumb, toe.)*

2. Show all self-inflicted accidents caused by shooters older than 50.

3. How old is the oldest victim in Jackson County? What part of the body was hurt? How old was the youngest? The part of the body?

4. Display all of the accidents that occurred on the first day of hunting season: Nov. 19, 1994

ANSWERS: HUNTING DATABASE EXERCISE #1

1. *Ankle down – you can choose whatever fields you'd like to see:*

   SELECT DATE, COUNTY, WOUND, INJURY, CAUSE
   FROM DEER
   WHERE WOUND = "ankle" OR WOUND = "toe" OR WOUND = "foot"

52 | 12/1/1993 | 1155 | Portage | south | thigh | minor | sp | ricochet-tree

Query1 : Select Query

DATE	COUNTY	WOUND	INJURY	CAUSE
11/26/1993	Dunn	toe	minor	careless handling-swinging gun
11/20/1993	Marinette	foot	major	unloading firearm-gloves
11/20/1993	Columbia	ankle	major	trigger caught on object
11/20/1993	Washburn	toe	minor	careless handling-rifl on foot
11/21/1993	Dunn	toe	major	unloading firearm
11/21/1992	Shawano	toe	minor	careless handling-trigger caut
11/22/1992	Calumet	foot	major	careless handling-fell asleep
11/22/1992	Langlade	toe	major	nonhunter handed loaded gun
11/23/1992	Adams	foot	major	uncasing loaded gun
11/25/1992	Portage	toe	major	unloading gun
11/29/1992	Door	foot	major	Bullet strikes victim in car
11/23/1991	Dunn	toe	minor	shooter stumbled and fell
11/24/1991	Manitowoc	foot	major	careless handling-tree involvd

Record: 1 of 46

   The answer contains 46 items.

2. *All self-inflicted accidents by shooters over 50:*
   SELECT CASE, DATE, COUNTY, VAGE , WOUND, INJURY, CAUSE
   FROM DEER
WHERE VAGE > 50 AND TYPE = "si"

Query1 : Select Query

CASE	DATE	COUNTY	VAGE	WOUND	INJURY	CAUSE
51	11/24/1991	Jackson	55	abdomn	major	careless handling-tree involvd
35	11/24/1990	Oneida	52	arm	major	loaded firearm in vehicle
20	11/19/1989	Chippewa	65	face	minor	careless handling-setting down
8	11/21/1988	Adams	55	thigh	major	careless handling-tree involvd
17	11/22/1987	Door	56	hand	major	unloading firearm-defective
28	11/24/1987	Adams	66	leg	major	firearm fell, insecure rest
19	11/22/1994	Dane	58	ankle	major	firearm fell, insecure rest

Record: 1 of 7

   There are seven injuries.

3. *Oldest and youngest victims in Jackson County with wounds?*
   SELECT CASE, DATE, COUNTY, VAGE , WOUND, INJURY, CAUSE
   FROM DEER
   WHERE COUNTY = "Jackson"
   ORDER BY VAGE DESC

Query1 : Select Query

CASE	DATE	COUNTY	VAGE	WOUND	INJURY	CAUSE
43	11/27/1987	Jackson	62	thigh	minor	victim out of sight of shooter
51	11/24/1991	Jackson	55	abdomn	major	careless handling-tree involvd
57	11/25/1991	Jackson	42	chest	major	victim out of sight of shooter
17	12/22/1994	Jackson	25	thigh	minor	careless handling-strange gun
27	11/26/1994	Jackson	20	leg	major	shooter stumbled and fell
34	11/23/1990	Jackson	20	arm	major	victim in line of fire
67	11/30/1991	Jackson	15	foot	major	careless handling-gun at foot
19	11/26/1988	Jackson	14	foot	major	careless handling-gun in lap

Record: 6 of 8

(The oldest is at the top of the list; the youngest is at the bottom)
Oldest: 62, thigh. Youngest: 14, foot

4. *First day of hunting season:*
   SELECT CASE, DATE, COUNTY, VAGE , WOUND, INJURY, CAUSE
   FROM DEER
   WHERE DATE = #11/19/94#
   (The # signs tell Access it's a date, not a word or a number.)

Query1 : Select Query

CASE	DATE	COUNTY	VAGE	WOUND	INJURY	CAUSE
1	11/19/1994	Chippewa	47	neck	fatal	victim mistaken for deer
2	11/19/1994	Buffalo	42	shuldr	major	victim mistaken for deer
3	11/19/1994	Outagamie	26	thigh	minor	victim out of sight of shooter
4	11/19/1994	Marathon	58	leg	major	victim mistaken for deer
5	11/19/1994	Calumet	26	abdomn	major	victim in line of fire
6	11/19/1994	Green Lake	33	chest	fatal	victim in line of fire
7	11/19/1994	Waupaca	14	foot	minor	careless handling-gloves
8	11/19/1994	Washburn	21	abdomn	fatal	victim in line of fire
9	11/19/1994	Iowa	22	arm	major	loaded gun in vehicle

Record: 1 of 9

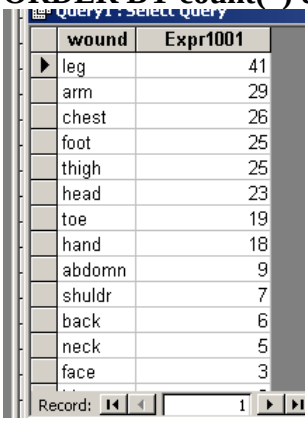
PERFORMING CALCULATIONS

Just as in a spreadsheet, sometimes you want to do calculations between columns or "fields." Because a database has no cell addresses, you have to use the names of fields. In Access SQL view, you type a calculation right into the Select line, and in design view you type the calculation right into the field name line in a fresh column.

SELECT sage-vage
FROM deer

Data Groupies:
GROUP BY lest you summarize your entire database. For example, in our deer database, we might want to know what the most common injury is. Before you do the query, think about what your result might look like. (I always think about what the chart I print in the newspaper would look like.) In this case I would have two columns: the wounds and the number of times each wound occurred. My headline would be: What Gets Wounded in All Hunting Accidents.

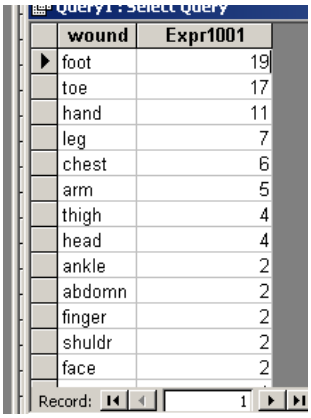
SELECT wound, count(*)
FROM deer
GROUP BY wound
ORDER BY count(*) desc



Leg injuries are most common, with 41 shot up. *Note: You can use this same technique for computing average, minimum, maximum and sum*

Now, let's make a minor change to the query. Say, we didn't care about all wounds, let's change that headline to Most Common Types of Self-Inflicted Wounds. This would require changing just one thing, and often we try to make this harder than it should be.

SELECT wound, count(*)
FROM deer
WHERE type = "SI"
GROUP BY wound
ORDER BY count(*) desc



wound	Expr1001
foot	19
toe	17
hand	11
leg	7
chest	6
arm	5
thigh	4
head	4
ankle	2
abdomn	2
finger	2
shuldr	2
face	2

Record: 1

Multi-Groupies

Another approach to the counting up self-inflicted wounds might be to group by both WOUND and TYPE.

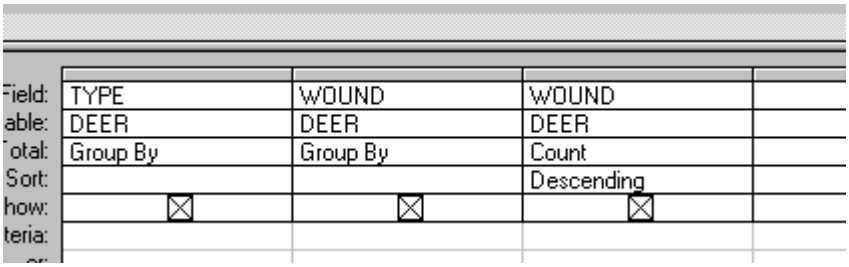
The steps:

1. Select WOUND, TYPE and COUNT of ?????. It can be difficult to decide what to count, but here's a hint: count will total the different combinations of everything you group by. But it will only count records that are filled in for the field you select. In our database, WOUND is always filled out. So let's just ask for COUNT in a new WOUND column.

2. Order by: Do it so the biggest count is on top
3. Selection criteria: I know, I know, we just pulled out self-inflicted. But now we want both self-inflicted, plus the other types. So remove the selection criteria.

4. Group by: group by both WOUND and TYPE.

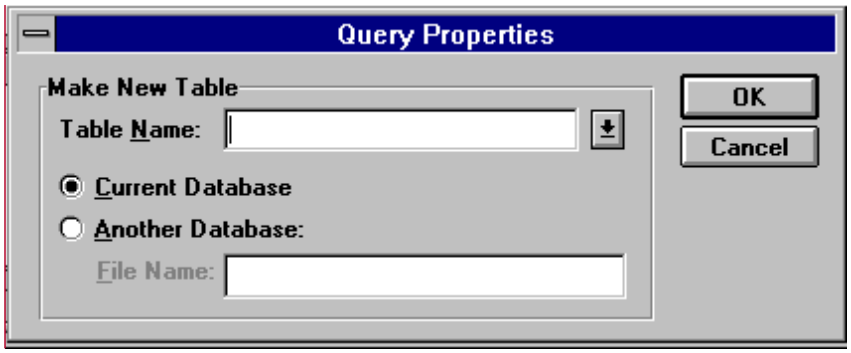
Run the query. Your answer will show all the possible combinations of TYPE and WOUND and count them. Answer: 30 leg injuries by second person



Field:	TYPE	WOUND	WOUND
Table:	DEER	DEER	DEER
Total:	Group By	Group By	Count
Sort:			Descending
Show:	☒	☒	☒
Criteria:			

SELECT type, wound, count(*)
FROM deer
GROUP BY type, wound
ORDER BY count(*) desc

A note about output: On the main query form, the output is usually set to browse. This means that your answer pops up in a window. You can change this output to save your answer to a separate file. To do this go to QUERY on the main menu and select MAKE TABLE, Access will then prompt you for the name of the table. Then click OK.



PRACTICE EXERCISE #2—HUNTING DATABASE

1. What was the most common cause of hunting accidents? How many? What's wrong?

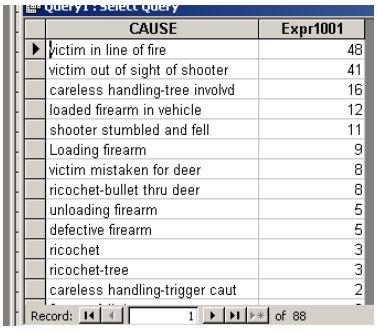
2. Which brand of gun is involved in the most hunting accidents? How many? Can you say it's the most dangerous?

3. Which county had the most accidents? How many? Can you say it's the most dangerous?

4. What is the average age of shooters in butt-injury accidents?

ANSWERS: HUNTING DATABASE EXERCISE #2

1. Most common cause of accidents:
   SELECT CAUSE, COUNT(*)
   FROM DEER
   GROUP BY CAUSE
   ORDER BY COUNT(*) DESC

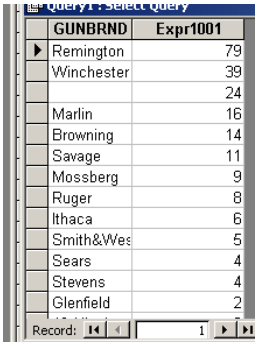


CAUSE	Expr1001
victim in line of fire	48
victim out of sight of shooter	41
careless handling-tree involvd	16
loaded firearm in vehicle	12
shooter stumbled and fell	11
Loading firearm	9
victim mistaken for deer	8
ricochet-bullet thru deer	8
unloading firearm	5
defective firearm	5
ricochet	3
ricochet-tree	3
careless handling-trigger caut	2

Record: 1 of 88

Answer: Victim in the line of fire (duh), with 48. But the causes aren't entered consistently, so you don't really know.

2. Gun brand:
   SELECT GUNBRND, COUNT(*)
   FROM DEER
   GROUP BY GUNBRND
   ORDER BY COUNT(*) DESC

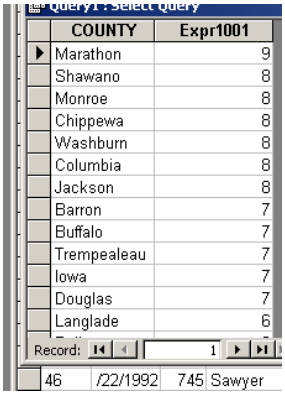


GUNBRND	Expr1001
Remington	79
Winchester	39
	24
Marlin	16
Browning	14
Savage	11
Mossberg	9
Ruger	8
Ithaca	6
Smith&Wes	5
Sears	4
Stevens	4
Glenfield	2

Record: 1

Answer: Remington, with 79. You also need to know the number of guns in use by brand.

3. County:
   SELECT COUNTY, COUNT(*)
   FROM DEER
   GROUP BY COUNTY
   ORDER BY COUNT(*) DESC



COUNTY	Expr1001
Marathon	9
Shawano	8
Monroe	8
Chippewa	8
Washburn	8
Columbia	8
Jackson	8
Barron	7
Buffalo	7
Trempealeau	7
Iowa	7
Douglas	7
Langlade	6

Record: 1

46	/22/1992	745	Sawyer

Answer: Marathon, with 9. You also don't know how many hunters hunt in each county.

4. Average age:
   SELECT WOUND, AVG(SAGE)
   FROM DEER
   WHERE SAGE>0 AND WOUND = "butt"
   GROUP BY WOUND

WOUND	Expr1001
butt	16.5

Answer: 16.5

PRACTICE EXERCISE #3—HUNTING DATABASE

1. Did more accidents occur in bad weather?

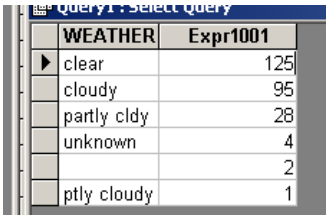
2. What was the most common cause of fatal accidents? Were most of those inflicted by the shooter or someone else?

3. Do accidents of certain causes have higher or lower rates of orange-wearing?

ANSWERS: HUNTING DATABASE EXERCISE #3

1. Bad weather?

   SELECT WEATHER, COUNT(*)
   FROM DEER
   GROUP BY WEATHER
   ORDER BY COUNT(*) DESC



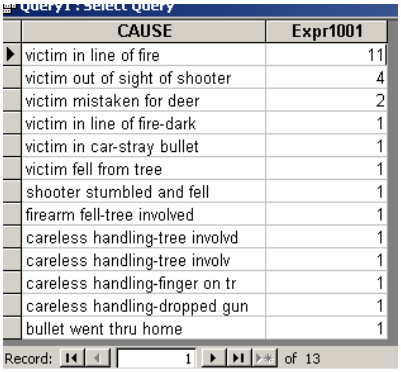
WEATHER	Expr1001
clear	125
cloudy	95
partly cldy	28
unknown	4
	2
ptly cloudy	1

Answer: You don't know. First, is partly cloudy considered bad weather? And something seems wrong with the categories: there's no indication of stormy weather. Most accidents occurred in clear weather.

2. Fatal accidents?

   This is really two queries:

   SELECT CAUSE, COUNT(*)
   FROM DEER
   WHERE INJURY = "fatal"
   GROUP BY CAUSE
   ORDER BY COUNT(*) DESC

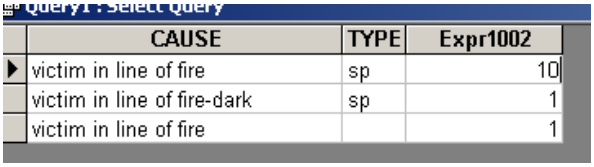


CAUSE	Expr1001
victim in line of fire	11
victim out of sight of shooter	4
victim mistaken for deer	2
victim in line of fire-dark	1
victim in car-stray bullet	1
victim fell from tree	1
shooter stumbled and fell	1
firearm fell-tree involved	1
careless handling-tree involvd	1
careless handling-tree involv	1
careless handling-finger on tr	1
careless handling-dropped gun	1
bullet went thru home	1

Record: 1 of 13

Then repeat to find type::

SELECT CAUSE, TYPE, COUNT(*)
FROM DEER
WHERE INJURY = "fatal" AND CAUSE LIKE "victim in line*"
GROUP BY CAUSE, TYPE
ORDER BY COUNT(*) DESC



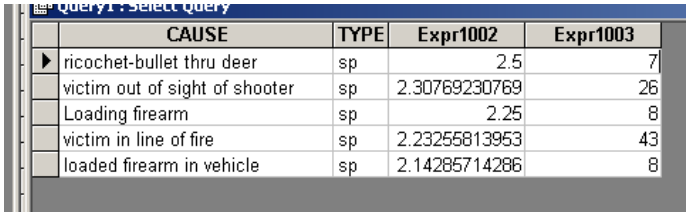
CAUSE	TYPE	Expr1002
victim in line of fire	sp	10
victim in line of fire-dark	sp	1
victim in line of fire		1

None are self-inflicted.

3. Rate of orange-wearing.

   This is an exercise in changing the language from reporting questions to query questions. It's also about refining your thoughts enough to make sense of a question.
   It's the same as an average number of pieces of orange. It also makes sense to limit the query only to second-person shootings, since the number of pieces of orange is irrelevant when you shoot yourself. I'm also limiting mine to causes with more than 5 accidents, since rates for very small counts can be misleading.

   SELECT CAUSE, TYPE, AVG(VORANGE), COUNT(*)
   FROM DEER
   WHERE TYPE = "sp"
   GROUP BY CAUSE, TYPE
   HAVING COUNT(*) > 5
   ORDER BY AVG(VORANGE) DESC



CAUSE	TYPE	Expr1002	Expr1003
ricochet-bullet thru deer	sp	2.5	7
victim out of sight of shooter	sp	2.30769230769	26
Loading firearm	sp	2.25	8
victim in line of fire	sp	2.23255813953	43
loaded firearm in vehicle	sp	2.14285714286	8

   Answer: Depends on what you call "different".