SQL CHEAT SHEET:

Here are the five basic lines in Structured Query Language, in the order they need to be used in a query, and denoted in parentheses is whether it is required or optional.

SELECT (required)
FROM (required)
WHERE (optional)
GROUP BY (optional)
ORDER BY (optional)

About each one:

SELECT -- this is where you list the fields (columns) you want displayed in your answer. If you want to see all the fields, you can simply type "SELECT *" (asterisk represents all the fields). The things you put here don't necessarily need to be a field exactly as you have it displayed in the table. It can be a calculation based on a field (the sum of all the values in a field or the average of all the values), it can be a math calculation between two or more fields (i.e. number of students divided by number of teachers to get a student-teacher ratio), it can be a combination of two text fields (i.e. firstname + lastname). Note: if any of your field names have spaces in them or start with a number or are the same as a "reserved" word in SQL-- such as the word "select" -- then you will need to put brackets around the field name. Here's an example:

SELECT county, [accident type], date, injury

FROM -- this is the name of the table(s) or query(s) that holds the data you are querying. If you are just pulling data from one table it will be simply the word FROM and the name of your table. If you are pulling data from more than one table -- this is referred to as a "join" -- the syntax for the FROM line will be more complex. (see other tipsheets for more on this)

WHERE -- this is how you filter which records/rows you want to appear in your answer or be calculated as part of your answer. The syntax of this line will vary greatly depending on what criteria you are using to filter. (see options below)

GROUP BY -- this is used when you want to summarize or aggregate your data based on one or more fields in your table. For example, if your data is one record for each hunting injury and there was a field indicating which county it occurred in, you could use group by to return a list of the counties and the number of injuries that occurred in each one. In this instance you would "group by" the field called County.

ORDER BY -- this is an optional line that you can use for setting the order of the records in your answer based on one or more of the fields in your table. The default is that it will sort ascending (youngest to oldest or 1 to 100 or A to Z). If you want it to go the opposite way -- descending -- you simply add the word "DESC" after the field name.

You can also do a multiple order by using more than one field. For example, the following query would first order them by the date and then when it finds multiple records on the same date, it would then order those records (on that one date) by the county name, alphabetically:

```
select county, date, injury, type
from deer
order by date, county
```

Most database programs, including SQL Lite, allow you to use a little shortcut in the order by line. Instead of referring to the name of the column, you can identify which "column number" (of your answer) that you want to order by.

So in the above example, we could change it to:

```
Select county, date, injury, type
from deer
Order by 2, 1
```

COMMAND	SYMBOL USED	EXAMPLE
Wildcard:	% and Like	Cause like "careless%"
Text:	Single quotes	county= 'Washington'
Date:	Single quotes	Date = '12/1/2004'
Number:	no special symbols	SAGE = 5
Blanks:	NULL	date is NULL
Don't include Blanks:	NOT NULL	date is NOT NULL
Does not equal:	<>	county <> 'Washington'
Does not equal:	NOT	County is not 'Washington'
Greater than:	>	SAGE>5
Less than:	<	SAGE<5
Greater than or equal to:	>=	Sage >= 20
Less than or equal to:	<=	SAGE<=20