Homework 5: Reading Assignment

1. Describe the differences between the where and having clauses in SQL queries.

> The WHERE clause pertains to individual tuples within a database instance, ie “rows where this clause is true”. The HAVING clause pertains to a group of rows, formed by the GROUP BY clause. Usually the HAVING clause can only be used with aggregate functions over a group, ie “groups where this aggregate function meets this requirement”.

2. Create an example query based on your schema for homework 3 that involves a group by, with both a WHERE and HAVING clause.

```
SELECT company_name, COUNT(area_of_business)
FROM area_of_business_info
WHERE area_of_business <> “Internet”
GROUP BY company_name
HAVING count(area_of_business) >= 2;
```

> This query will return the company name, and number of areas of business of that company that aren’t internet, for each company that has more than 1 area of business, in our area_of_business_info table. The WHERE clause specifically excludes all rows where area_of_business is “Internet”, because we don’t care about those for whatever reason. Then, it groups all remaining companies, and excludes all companies with under 2 areas of business, since we’re only interested in companies that have multiple areas of business. The WHERE clause filters out specifics, and the HAVING clause lets us exclude groups not meeting a general criterion.

3. Briefly describe, in your own words, the difference between a view and a stored procedure. Also, compare views and stored procedures to constructs in programming languages.

> In my mind, a view is sort of a “fake table”, wherein one can show a subset of another relation instance. This fake table can be used in other queries, but can’t do things like modify the table it’s representing. A stored procedure is more like a function - if you’re going to use a specific query or operation many times, you put it in a stored procedure to use whenever you want. A stored procedure, in my mind, is almost exactly like a function. It can take parameters, has logic inside, can change tables, and returns things. A view is tougher to describe - to me, it’s sort of like a toString, or a getter method, for a class. In a sense, we’re hiding somethings, and showing/returning only what we need to. You don’t give a toString or getter any parameter (usually), and you can’t really use a getter or toString method to change a class itself; however, it does provide specific, useful information that you can use in another context.