Matthew Lonis

Jeff Whitmer

CSCI-A 290

March 23rd, 2017

***MINI 2***

**Topic 1: What are SQL statements, expressions, and queries? Are these the terms that are most commonly used? What makes each different from the others? Do they have important features in common?**

**3 URLs**

1. <http://stackoverflow.com/questions/4735856/difference-between-a-statement-and-a-query-in-sql>
2. <http://www.informit.com/articles/article.aspx?p=29661>
3. <https://www.tutorialspoint.com/sql/sql-expressions.htm>

**Precis of Sources**

A statement, in SQL, is the general term to describe or define a complete piece of correct SQL that can be sent to a DBMS. A statement has a persistent effect on the database. A query is a subdivision of a statement. A query is, in general terms, a statement that retrieves data and returns it. This makes queries a special kind of statement. A query is often synonymous with the keyword SELECT because it is always needed in a query to select which data to be pulled from which table.

An expression, in SQL, is a combination of values, operators, and SQL keywords that will evaluate to a value. A common way to use expressions is to select specific data from a database using defined parameters. In this way, a Boolean expression is used to retrieve data that matches the Boolean expression. So, for example, if there exists a student table with columns named: firstName, lastName, address, etc. A Boolean expression can be used to retrieve all the data from the table whose first name is “Chris.” Other types of expression are numeric expressions. Numeric expressions can be used to determine averages and sums of numerical data from a database.

**Summary of Topic**

In SQL, statements are a broadly defined term which is anything that is valid and can be passed to a query. Statements can be used to make tables, update a table, etc. Queries are statements that only retrieve data from a table. It’s a special kind of statement but different because it is only defined as a statement that retrieves and returns data. An expression is also a special kind of statement but it can be used to perform more complex tasks or queries. Statements can be used numerically to determine sums or averages over numerical data in a database. Expressions can also be used to run advanced queries that specify specific data to be returned based on Boolean expression. This allows for more fined tuned queries that go beyond just singling out a specific table, column or row to be returned and instead only returns data that satisfies the Boolean expression.

**JIT 2**

Are expressions and queries specific types of statements?

**Topic 2: What, if any, are the key issues regarding the one-to-one, one-to-many, and many-to-many relationships that can be created between tables? Be sure to investigate both what makes each one important and unique and what important issues/concerns/”gotchas” are involved with each. You will have to research “Linking Tables” in the context of the many-to-many relationship.**

**3 URLs**

1. <http://stackoverflow.com/questions/7296846/how-to-implement-one-to-one-one-to-many-and-many-to-many-relationships-while-de>
2. <http://www.tech-recipes.com/rx/56738/one-to-one-one-to-many-table-relationships-in-sql-server/>
3. <http://www.tomjewett.com/dbdesign/dbdesign.php?page=manymany.php>

**Precis of Sources**

A one-to-one (1-1) relationship is a relationship between two tables where each table is associated with each other based on only one matching row. This relationship is setup using primary keys for both tables and having one of the tables use a foreign key referencing to the primary key of the other table. Both keys need to be unique. This allows data from one table to appear in another table. Once this relationship is established, you cannot enter in detail in the table that references back to a different table that has the same data because it will cause a unique key violation.

A one-to-many (1-M) relationship is a relationship where one row in a table can have multiple matching rows in another table. This relationship is also setup using primary and foreign keys. In this relationship, a row in one table can correspond to multiple rows in another table instead of just one. A many-to-many (M-M) relationship is best described by visualizing a database table flow chart. In words, however, a many-to-many relationship relates data in a table to many rows in another table which is also related in multiple ways to another table. In this manner, the tables are linked and joined together. The relationship is called M-M because a table’s max multiplicity is in any direction is many.

**Summary of Topic**

There are three types of relationships in SQL: one-to-one (1-1), one-to-many (1-M) and many-to-many (M-M). A 1-1 relationship is a relationship where one row of a table is related to exactly one row of another table. This allows the second table to use data from the first table but the data from the original table only corresponds to exactly one row in the new table. A 1-M relationship is where one row in a table can correspond to multiple rows in another table. This is like 1-1 except that a row from the first table can be used more than once in a second table. A M-M relationship is a relationship where a table incorporates many 1-M or 1-1 relationships within itself, allowing for its multiplicity in any direction to be “many.”

**JIT 2**

Are we going to get to work with complex M-M relationships?

**Topic 3: What can you find out about these KEYWORDS below? What does each one do for us and why is each one important?**

1. **SELECT**
2. **FROM**
3. **WHERE**
4. **GROUP BY**
5. **HAVING**

**3 URLs**

1. <https://www.w3schools.com/sql/sql_select.asp>
2. <https://www.techonthenet.com/sql/from.php>
3. <https://en.wikipedia.org/wiki/Where_(SQL)>
4. <https://www.tutorialspoint.com/sql/sql-group-by.htm>
5. <https://www.tutorialspoint.com/sql/sql-having-clause.htm>

**Precis of Sources**

The SELECT statement in SQL is the most common and often required statements to be used, especially to run queries. SELECT is used to “select” data from within a database and more specifically, specify which columns to select the data from. The FROM clause is used in conjunction with other statements and allows for the specification of a specific table to select data from as well as any joins that are required. The WHERE keyword is much like a Boolean expression and when combined with SELECT and FROM, allows for the retrieval of data that will result in true in the where clause. The GROUP BY clause is used with the SELECT statement and allows for the arrangement of data into groups. The GROUP BY must come after the WHERE clause in a proper SQL statement. HAVING works just like WHERE by creating a Boolean expression that must be true if data wants to be returned by the SQL statement. HAVING must come after GROUP BY in a proper SQL statement. This is because HAVING specifies conditions on the groups created by GROUP BY whereas WHERE puts conditions on the selected columns.

**Summary of Topic**

In SQL, the SELECT keyword is generally the first keyword to be used in any SQL statement. SELECT selects specific columns to return the data from in a database. The FROM keyword further specifies which table to select from within the database. The WHERE clause places conditional clauses on the selected columns and only return the data that satisfies the condition. The GROUP BY keyword groups the returned data based on a specified attribute or column. The HAVING keyword is much like the WHERE keyword however, instead of placing conditions on the columns that are selected, the HAVING clause places conditions on the groups created by GROUP BY.

**JIT 2**

Will we work with more than these three keywords?