SQL INJECTION, ACID, BASE:
Importance in Database Design

SQL injections, BASE, and ACID are all important topics to know when working with databases. By learning about SQL injections, a developer will be better able to produce a secure database. A developer that learns about BASE and ACID can then make a better decision about which one to pick to use for the database.

The first thing to learn is SQL injection. The website w3schools says that SQL is a code injection technique, that may have the capability to destroy the database, and that it is a common hacking technique. The whole point of SQL injection is to manipulate a database by using an SQL statement for some kind of input, like a username. This essentially means that people can use malicious code, or really well-crafted SQL statements, through web page input to alter or affect the database in some way.

These SQL injections can be a problem because you can ask for a username and password, and if the user is smart enough they can type in an SQL statement that gains them access to essential information in the database. W3schools has an example in which the user inputs “105 OR 1=1” and this would produce a valid SQL statement, which may allow them to gain access to all the usernames and passwords for that database. According to w3schools this is just one kind of injection, based on an always true approach. This is what the example above shows, 1=1 is always true. There is also the approach using batched SQL statements, which is a group of SQL statements separated by a semicolon. This method can also be used to access information in the database. This is a problem for any database, since it is important to keep user names and passwords out of someone else’s hands.

There are ways to protect the database from these SQL injections. The website w3schools has some suggestions for protecting the database. One way is to use SQL parameters, which are values added in a controlled manner to the SQL query at execution time, to secure the database. These are represented by an @ marker and are interpreted literally and not to be executed. Since it is interpreted literally it will prevent some user from entering an SQL statement in order to trick the database into giving them private data. Another thing a developer who is creating a database needs to learn is what ACID and BASE means.

Knowing what the acronym ACID means is important in database design. According to Wikipedia ACID, which is an acronym that means Atomicity, Consistency, Isolation, and Durability. ACID is essentially a set of properties and there are a sequence of database operations that must satisfy the ACID properties. These sequence of database operations can be perceived as a single logical operation, if they follow and satisfy the ACID properties, on the data, which is called a transaction. To better understand what it means it will be necessary to define each part of the acronym.

The first thing is to define all four parts of the acronym. SearchSQLServer defines the four attributes starting with Atomicity. Atomicity is a transaction involving two or more discrete pieces of information and either all the pieces of the transaction are committed or none of the pieces are committed. This means that if all the information is not put into the database, then none of it will be put into the database.

The next part of the acronym, that is defined by SearchSQLServer is consistency. Consistency is when a transaction either creates a new and valid state of data or when

a failure happens it returns all data to before the transaction. This basically means that the transaction will either succeed or it will not occur at all.

The third word in the acronym, which is defined by SearchSQLServer, is isolation. Isolation is a transaction that is in process and not yet committed, so it needs to remain isolated from other transactions. This means while a transaction is taking place it will not be interrupted.

The final word in the acronym, defined by SearchSQLServer, is durability. Durability is when committed data is saved by the system so there can be no data lose, even with an event failure or system restart. This means that the data is safe no matter what happens to the database. Also, according to SearchSALServer these four attributes, that make up the acronym, are ensured to any transaction by a transaction manager.

An example of why ACID is useful, according to the Service Architecture website, is a computer system that allows people to buy products would need some way to prevent people from buying the same product at the same time as everyone else. ACID helps by preventing the transactions from overlapping each other. This means that no two people can buy the same product at once, which can help the retailer a lot with things like inventory and item availability.

The last thing that a developer of a database needs to learn is what BASE means. According to neo4j the BASE acronym means Basic Availability, Soft-state, Eventual consistency. The website neo4j also defines all the terms.

The first term defined by neo4j is basic availability. Basic Availability means that the database seems to work most of the time. This means that it does not have to be as reliable, unlike ACID. This offers a greater a greater availability for the database.

The second term that is defined by neo4j is soft-state. Soft-state means stores do not have to be write-consistent. Basically this kind of database does not value consistency like an ACID database. This means the data is not as consistent then in the ACID database.

The last term to be defined by neo4j is eventual consistency. Eventual consistency stores exhibit consistency at some later point. This means that the database that uses BASE stores the data in a consistent manner at a later date rather then immediately. Essentially database based on BASE values availability over guaranteed consistency.

A good thing to do is compare the two kinds of databases in order to see which one would be useful for the database being created. According to the neo4j website, when figuring out if to use ACID or BASE, a developer needs to select the appropriate one on a case-by-case basis. When using BASE a developer needs more knowledge about consistent data. It is also essential that the developer be familiar with their chosen aggregate. Although ACID is much simpler than BASE, especially when the developer must plan around the limitations of BASE. ACID is also useful for when data consistency is essential.

Essentially SQL injections, BASE, and ACID are major topics when dealing with databases. The SQL injections are a good thing for developers of databases to learn to

avoid. The BASE and ACID are also good for a developer of a database to know, so the developer knows which one to pick for the particular database they are creating.

Works Citied

"ACID." *Wikipedia*. Wikimedia Foundation, 21 Mar. 2017. Web. 10 Apr. 2017. <https://en.wikipedia.org/wiki/ACID>.

Bryce Merkl Sasaki, Aspiring Graphista | September 4, 2015, Neo4j Community

Disclaimer, Reach out and Connect with the Neo4j Staff., and More Events →.

"Graph Databases for Beginners: ACID vs. BASE Explained." *Neo4j Graph Database*. N.p., 18 July 2016. Web. 10 Apr. 2017. <https://neo4j.com/blog/acid-vs-base-consistency-models-explained/>.

"SQL Injection." *SQL Injection*. Refsnes Data, n.d. Web. 10 Apr. 2017. <https://www.w3schools.com/sql/sql_injection.asp>.

"What is ACID (atomicity, consistency, isolation, and durability)? - Definition from WhatIs.com." *SearchSQLServer*. N.p., n.d. Web. 10 Apr. 2017. <http://searchsqlserver.techtarget.com/definition/ACID>.