CMP2060M

DATABASE SYSTEMS ASSIGNMENT 2 REPORT

Daniel Dixon

Student ID: DIX16602092

1. NoSQL

Provide a response to the following question: "Describe what is meant by the NoSQL term 'Dynamic Schema', and what benefits could the use of a Dynamic Schema have over relational database Fixed Schemas, such as that currently used in your SQL database solution?" You must justify and critique your response beyond a simple description, and contextualised to your final database solution;

As stated in the CAP Theorem, "there is a fundamental trade-off between consistency, availability and partition tolerance" (Gilbert and Lynch, 2012, 1) when applied to distributed systems. Consistency involves any changes to databases being expresses across the entire system. Availability involves any server request sent receives a response even if it is a slow one. Partition-tolerance is focussed on to make sure only a network failure can cause an incorrect system response. Both NoSQL and MySQL, the database management system used in this assessment, cater to these trade-offs in different ways.

MySQL is a Static SQL Schema which focusses on consistency and availability in its structure. A Static Schema is considered hardcoded and unchangeable once executed, it also has fixed datatypes that every piece of data inserted must adhere to. Finally, MySQL is known as a relational database language in which each table in a database has to be connected in some way, in this case using primary, foreign or composite keys, in order to interact with each other.

Not Only SQL (NoSQL) usually attempts to give both consistency and Partition tolerance using a Dynamic Schema. Dynamic Schemas involve the code being "Constructed at runtime" (Goodson and Steward, 2009), they don't need the databases to be related for them to interact and the data that is inputted can be of any datatype.

When comparing both static and dynamic schemas both have their own advantages and disadvantages.

"Static SQL provides performance advantages over dynamic SQL because static SQL is pre-processed, which means the statements are parsed, validated, and optimized only once" (Goodson and Steward, 2009). It has issues however, such as a high level of work to change parts of the schema, this work only increases further with the increase of the schema's complexity, this added complexity continues to cause issues in several other areas such as data queries where more are needed to find similar data. SQL can be used for smaller databases that are ensured to be inputting the same datatypes every time.

Dynamic SQL may be considered slower for performance but it has benefits too such as attempting to solve the issues previously mentioned for Static schemas including the removal of the need for database relationships to interact and the complexity of data queries is decreased, making it much more searchable. This makes it much better for a company who sells multiple different items that need separate data kept about them as a separate storage structure isn't needed.

In terms of the database solution used for this assessment, several issues came about that could have been solved using dynamic schemas. The need for keys slowed down the creation process as the whole schema structure and relationships had to be devised before the input of data was allowed, this could have been solved by the use of an object ID in NoSQL. There was also a lack of future proofing as if the garden centre diversified its products in the future a lot of changes would need to be made to make the database usable.

2. DATABASE SECURITY

Provide a response to the following question: "Using your own relational database solution as context, describe what an SQL Injection Attack is, and what steps you can take to protect your database solution from such an attack?" You must support your response by providing an example(s) of hypothetical SQL queries that could be used as part of a targeted SQL injection attack against your implemented database solution.

An SQL Injection Attack "exploits a security vulnerability occurring in the database layer of an application and a service" (Sajjadi and Pour, 2013, 1). Using SQL against the database, for example the one used in the database solution for this project, the attack seeks to gain access to systems or data stored within the software. "Injection attacks generally take advantage of improper validation over input/output data" (Sajjadi and Pour, 2013, 1), there is no user interface created as part of this project and hence no input or output data but the code infrastructure in place can allow this to be developed in the future.
If a person attempting to access the project database system to find out information on the system's customer information but they were not a user with any privileges, they could gain these privileges using an injection attack. If not previously protected a user could load the customer login page that uses a SELECT statement and user input being plugged directly into said statement and manipulate the underlying code. They could enter SQL code into the username location such as ' or 1=1 -- to change the statement and gain access to all customers stored in the customer table. Once complete this attack would give all the customers names, emails and addresses, which they could then sell on to others who will use it for advertising or malicious purposes. The attacker could also use this data to commit other attacks such as phishing emails using the customers email and their known affiliation with the garden centre to send them targeted emails to give up money. If the staff's postcodes were again linked to the address table then the previously acquired data could be used to eventually find their login information, and that could then be further escalated to gain access to the admin account and subsequent privileges. To protect this from occurring several methods can be implemented, one of these is a prepared statement, "SQL statements that separate statement structure from statement input" (Thomas, 2009, 590). The prepared statement can be used to make sure when the attacker tries to enter the SQL code into the customer username area it will be disallowed as the datatype users can enter are severely restricted, the previously mentioned code would cause the attacker to be told his username is incorrect and try again, thus the attack has be prevented from working.

3. REFERENCES

Gilbert, S and N. Lynch (2012) *Perspectives on the CAP Theorem*. Singapore: IEEE. https://groups.csail.mit.edu/tds/papers/Gilbert/Brewer2.pdf [accessed 2 January 2017]

Goodson, J. and Steward, R. (2009) *The Data Access Handbook*, 1st edition. New Jersey: Prentice Hall Professional.

Sajjadi, S and B Pour (2013) *Study of SQL Injection Attacks and Countermeasures*. Singapore: IJCCE http://www.ijcce.org/papers/244-E091.pdf [accessed 3 January 2017]

Thomas, S., L. Williams and T. Xie. (2009) On automated prepared statement generation to remove SQL injection vulnerabilities. *Information and Software Technology*, (51), 589-598