Database Architecture

Q1)What are cursors? Why are they useful in application architectures? What are some of the benefits? Are there drawbacks to cursors? How do MySQL and SQLite support cursors and how would you (in one of them) use them? Provide a 300-500 word explanation of cursors.

A1) A cursor holds multiple rows returned by a SQL statement. They are used to store database tables. There are two types of cursors :

1) Implicit cursurs - They are default cursors that are already creted. These are allocated when the user performs DML operations

2) explicit cursors - They are created by the user whenever required. It fetches the data from the table in a row-by-row manner.

Cursors are useful in applications because of their advantages like :

Row by row processing which in turn can help us perform operation on each row.

Can provide first few rows prior to assembling the whole result set.

For updating changes in the database we do not need to send seprate queries to the SQL server.

Cursors are faster in performance.

As everything has advantanges as well as diadvnantages so is the case for cursors. #### It requires more memory for storage.

It uses more bandwidth compared to the execution of a sing SQL statement.

There are five steps to create an explicit cursor

1) Declare the cursor object

2) Open the cursor connection

3) Fethch data from cursor

4) Close the cursor connection

5) Deallocate the memory

There are six methods to fethc the data from the cursor:

FIRST - Fetches only first row

LAST - Fetches only last row

NEXT - Fetches data in forwrad direction

PRIOR - Fetches data in backward direction

ABSOLUTE - Fetches an exact row

RELATIVE - Fetch via increment or decrement.

MySQL cursors have three properties

1. Read Only - The data in the table cannot be modified but can only be read.

2. Non_Scrollable - Only rows that are asked in the SELECT statement are displayed. One cannot skip rows or jump to any row required.

3. Asensitive and Insensitive Cursor - A asensitive cursor points to actua data and the sensitive cursors uses a copy of the data.

In SQLite it acts as a middleware between the db connection and the SQL query.

To use a cursor in MySQL four steps are executed

- Declare the cursor

- Open the cursor

- Fetch the cursor

and finally,

- Close the cursor.

Q2) What are connection pools? Why are they useful in application architectures? What are some of the benefits? Are there any potential drawbacks? What are some of the main issues with using connection pools? Provide a 300-500 word explanation of connection pools.

A2) A database package by default has a basic connection pool. There is very little abitlity to control it. Connection pooling might open up two connections and execute them seperately and this may result in unexpected behaviour of the queries. One can execute various connections but if a lot of things are being addressed at once it could lead to various connection and the server can throw an error stating there are too many connections. It is always advised to disconnect your db connection because keeping if kept idle for longer time can cause timeout issues.

We do not need to create a connection pool as it is already created depending on the language being used for the database but we do have to size them accordingly. We need to make sure that the number of idle connection are less and it has the appropriate number of maximum connections. Making the pool too small or too big can introduce introduce latency. To determine the perfect size of pool all the factors need to be considered.

Whenever a new connection is created a new connection pool is created based on matching criteria that associates the pool with the connection string in the connection. The connection pool is associated with a unique connection string and when the connection string does not exactly match with the existing pool a new connection and a new pool is created. Connections are pooled per process, per application and per connection string.

ADVANTAGES

Database connection pooling reduces the cost by mainiataing a pool of the open connetions. Database connection pooling can improve the application performance.

It reduces the times the user has to create a new connection every time as the connection already exists it just needs to open and close when and if required.

The connection can be reused and can be quickly used.

Requires minimum effort to manage the connection

Controls the cost and resources spend on maintaining various connections.

DISADVANTAGES

It also has some drawbacks like :

There are costs associate with opening closing as well as maintaining a db connection.

The connection pool can end up being stale this happens when the inactive connections are timed out.

Q3) In this question you will venture into the literature (as important skill for graduate students). Start by reading this paper:

Atoum, J. O., & Qaralleh, A. J. (2014). A hybrid technique for SQL injection attacks detection and prevention. International Journal of Database Management Systems, 6(1), 21.

Once you have finished reading the article, select a second article from the references in the first article or closely related to the topics in the first article and then write an 800 - 1000 word review of the two articles (combine the two reviews; do not write two separate reviews). What are the main issues they address? List three things that you learned or three key take-aways (from either of the two papers). Are there statements that you do not agree with?

A2) A COMBINED REVIEW OF BOTH PAPERS

The research paper talks about SQL injection attacks. SQL injection attacks(SQLIAs) are used used to gain, manipulate, or delete information in any data-driven system. SQLIAs are simple to be learned and used hence even unexperienced hackers can use it. Different researchers have covered about various topics to prevent it but there has been no solutions that can help the entire system to be prevented from such attacks. SQL attacks are more prone to happen if the systems are not updated. Always try to increase the security of your system by monitoring, logging, validation and other operations. The first step towards building a secure database is that the access is well controlled. The secure data should be encrypted so that even if it is hacked the sensitive data is not leaked. The most used gateway when SQLIAs are performed

is browser variables, user inputs, and injection HTTP header. Some of the aspects that came up to prevent SQLIAs :

- Static Analysis which is the most widely used.

- Runtime Analysis detects specific type of attacks.

- Static and Runtime Analysis combines two techniques to come up with a more effective solution.

HYBRID TECHNIQUE

- Normal Data Exchanging Strategy : This introduces the normal three tier execution of dividing the system into three tiers :

1) Presentation Tier 2) Logic Tier 3)Storage Tier

- Suggested Approach Strategy : This approach provides an extra defense line Data-Tier to ensure that this side will not execute any abnormal codes

- Suggested Approach Stages : This approach is based on different stages to reject any malicious query from being passed through the database engine before its execution.

The second paper mentions that SQL injections has been described as one of the most serious threats for all the web applications and the hacker can get complete access to the database which has all the important information. To describe the attack two important characteristics of SQLIAs are :

1. INJECTION MECHANISM

Injection through user input : Attackers inject SQL commands by crafting user inputs. User input usually comes from GET, POST, DELETE requests.

Injection through cookies : It is advised not to accept the cookies on the web pages they contain state information and are stored on client machine.

Injection through server variables : Server variables contains a collection of variables that have HTTP, network headers, and environmental variables. These are used in variety of ways.

Second-order injection : In here hackers seed malicious inputs in the system which will trigger whenever the user enters any input

2. ATTACK INTENT

Identifying injectable parameters: The attacker tries to find out what user input fields are vulnerable to SQL injection attack

Performing database finger-printing: The attacker tries to find out what type or version of the database is used for the web application and then tries too create a SQLIA related to the database.

Determining database schema: The attacker tries to find out the details of the entity relationship model and tries to find the column names, row names and total number of entities etc.

Extracting data: This allows to extract the values of the data.

Adding or modifying data: This attack basically focuses on changing the database.

Performing denial of service : This attack denies user the access and shuts the database down

Evading detection: This attack disallows the systems protection mechanism to work.

Bypassing authentication : This type of attack allows the attacker to bypass the database.

Executing remote commands : These type of attack can execute commands on the database.

Performing privilege escalation: These type of attacks taKe advantages of errors like logical or syntax errors in the database.

DIFFERENT TYPES OF SQLIAs

1. Tautologies

2. Union Query

3. Stored Procedures

4. Piggy-Backed Queries

5. Inference,

6. Alternate Encodings etc.

PREVENTION OF SQLIAs

1. DEFENSIVE CODING PRACTICES

Encoding of inputs : Using functions that encodes a string such that all the meta characters are encoded and presented as normal characters to the database.

Positive Pattern Matching : This helps in identifying good or bad input which is called positive validation. As every type of attack can be identified by a developer a safer choice is to make sure that there is positive validation.

Identification of all input sources : The input sources can be used to perform attacks hence developer should be aware about all the inputs that can be given to the web application and all the sources where user can enter an input should be checked.

2. DETECTION AND PREVENTION TECHNIQUES

Black Box Testing - This techniques uses a web crawler and identifies all the points where SQLIAs can take place in the web app. It then uses machine learning approach to prevent such attack although there is no 100% completeness

Static Code Checkers. - There exists a JDBC checker that checks the type correctness of the SQL queries

Combined Static and Dynamic Analysis - AMNESIA uses models and checks all queries using the model before they are being redirected to the database.

Taint Based Approaches - The WebSSARI detects the points where the required need are not met and then suggests filters and sanitization techniques

New Query Development Paradigms - Safe Query Objects provide a way to solve the problem by changing the query building process. It can be changed into one using an API.

Intrusion Detection Systems - The system is based on machine learning techniques that can be trained using a set of pa reticular application queries.

Proxy Filters - SPDL provide constraints and they can be applied to an application parameter when flowing from the web to server.

Instruction Set Randomization - SQLrand allows user to create queries using random instructions.

THREE KEY TAKE-AWAYS

1. As Computer Science students there is no scope for mistakes as any minute or minor mistake can lead to hackers taking advantage of the data present in the system. Developers need to make sure they are aware about all the techniques that can be used to perform an SQLIA and should write their code keeping all this in mind

2. Always make sure not to accept Cookies on web pages as you might just be making your data prone to hackers and always update the new versions of softwares.

3. Although all the research has been done and is still progress there is not yet any solution that can give 100% completeness or prevention from SQLIA attack.

SOMETHING I THOUGHT ABOUT

There was a point mentioned in the research paper that stated the sensitive data should be encrypted and if that is acheived even if the data is attacked the sensitive data won't be leaked. I still wonder why not all the data in the database can be encrypted. Maybe something like this can't be done because this might be cost effective for the industries or maybe it will make it difficult for the SQL users to perform queries on a regular basis.

REFERENCES

!("https://www.geeksforgeeks.org/what-is-cursor-in-sql/")

!("https://www.mysqltutorial.org/mysql-cursor/")

!("https://www.c-sharpcorner.com/UploadFile/b5eb6a/use-cursor-in-sql/")

!("https://docs.oracle.com/en/database/oracle/oracle-database/21/jjucp/intro.html#GUID-F9DE0E14-C0D9-40E7-99A3-AC207CE9DF14")

!("https://blog.pythian.com/sql-server-understanding-and-controlling-connection/")

!("https://www.cockroachlabs.com/blog/what-is-connection-pooling/")

WORD COUNTER AND PLAGIARISM CHECKER

!("https://wordcounter.net/")

!("https://www.turnitin.com/")