IJARSCT

ISSN (Online) 2581-9429

IJARSCT

Impact Factor: 6.252

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 2, February 2022

Inverse SQL Query Generation Algorithm in the DB Learn Adaptive E-Learning System

Prof. Deepika R. Thakare, Prof. Pramila S. Gaidhani, Prof. Chndrabhan R. Ghuge

Department of Computer Engineering

Guru Gobind Singh Polytechnic, Nashik, Maharashtra, India

deepika.thakare@ggsf.edu.in, pramila.gaidhani@ggsf.edu.in, chandrabhan.ghughe@ggsf.edu.in

Abstract: *Currently, individuals who learn and instruct SQL orders need to get hands-on training with genuine climate to make the learning successful. The making SQL practice is a tedious undertaking for educators. Thus, understudies probably won't get an adequate number of inquiries to meet their requests. The converse SQL question age calculation (ISQLG) is created to take care of this issue. The ISQLG has capacity to naturally produce SQL practice for understudies. The ISQLG can turn around the manual inquiry creation process what begins from making question to begin making inquiry answer first all things being equal. The ISQLG thinks about the current information also data set design by utilizing different requirements. The teachers additionally can determine the language, design and clarification of the inquiries. The ISQLG upholds DML orders – SELECT, INSERT, UPDATE and DELETE – furthermore support information recovery from different tables which perform by JOIN and sub query tasks. The calculation has capacity to produce mass inquiries with less exertion. The educator isn't needed to composing prerequisites and approving the inquiries. The ISQLG can be executed in e-figuring out how to improve feasible practices and further develop learning result for understudies.*

Keywords: SQL; automated question generator; e-learning; database learning; reverse question generation.

I. Introduction

Organized inquiry language (SQL) is language planned for overseeing information held in social data set administration framework (RDBMS) and furthermore basic information to all of software engineering understudies. Conventional method of educating SQL has two ways to deal with get ready SQL works out. The first approach is to set up the activities by considering on as it were information base pattern. The benefit is simple for educators since they don't need to check the accuracy reply which is gotten back from the framework. Nonetheless, understudies don't have freedom to rehearse in genuine climate and need to envision the outcomes without anyone else. The subsequent methodology is to make practices worried on both blueprint and informational index. This methodology makes understudies to learn and attempt the inquiry with genuine data set. The understudies have more viable learning from this methodology. However this methodology makes issues for teachers since it devoured a great deal of time and exertion. This activity made from outline and dataset is handled from three stages which are SQL question composing (inquiry clarification), answer composing (SQL inquiry) and reply approval then the teacher invests a great deal of energy and exertion to get ready a lot of activities. With time constraint, the teacher needed to reuse the old activities which have little amount of activities with less assortment in SQL orders. It may be an underlying driver of issues on understudy angle too. The understudies probably won't have an adequate number of activities to meet their want. Nonetheless, rehearsing with similar activities over and again can't further develop understudy learning result. This paper introduced the first RSQLG (Reverse SQL Question Generation) Algorithm to settle the referenced issues. The RSQLG calculation has a switched cycle to make reply (SQL inquiry) first then, at that point, produces question (inquiry clarification). This calculation can create various practices by contributing the source data set. Utilizing this calculation, understudies can rehearse their SQL ability with variety practices and the current inquiry can be kept away from. The RSQLG can be incorporated with e-learning framework. The understudies foster their taking in outcome from this generator.

DOI: 10.48175/IJARSCT-2790

II. Literature Review

2.1 Automated Question Generator Algorithm

The Automated inquiry generator is a device which can ease up the educator's responsibilities and improve the learning experience via programmed producing mass inquiries from an input source. The information source can be differed like a pool of information, the accuracy answers and questions. The age calculations are examined in related region. The fascinating works can be shown as follows:

Lee et al. fostered an age calculation for English language course by producing mistake ID question.

This calculation produces potential mistakes in a sentence [1].Funabaki et al. fostered the calculation to produce investigating question for programming courses. The calculation makes bugs in JAVA source codes with these techniques which are order cancellation; variable trading and order inclusion [2].Khalek et al. fostered the calculation to produce SQL questions from the information base pattern for testing data set execution [3]. Notwithstanding, their calculation doesn't consider existing information in tables or presents the technique to produce inquiry clarification which is shown as question in the works out. Past research papers fundamentally convert a source component (contribution to) an outcome component (work out). The source component can be either reply, for example, linguistic right sentence, sans bug source codes or information base mappings. The RSQLG generator applied this idea in calculation to produce SQL questions and replies. The source component is an info information base. The RSQLG can remove information from the information base to produce SQL inquiries and increases the results with question clarifications.

2.2 SQL Practice Questions

The fundamental information control language (DML) is engaged in this paper. The crucial DML orders are SELECT, INSERT, UPDATE and DELETE. The design of the inquiries from the SQL practice official book is took for investigation. [4] – The model inquiry can be addressed in Table I. In rundown, the SELECT objective purposes to show the information which is trailed by field names, table names and the conditions. The UPDATE objective is also to SELECT target however the initial segment of UPDATE purpose to change the information as opposed to showing the information. The INSERT objective purposes to add new information which is followed by table name then the information is added into referenced tables. The DELETE question contains the conditions which follow by the table name.

Objective	Command	Question Example
SELECT statement	SELECT,FROM ,WHERE	Display Student ID, Name ,Dept. and whose name start with ‘D’
Aggregate Function	SUM,AVG,COUNT	Write a query to display name of student fees
Update		Update student fees from student table
Insert		Insert new record in student table

Table 1: Example Questions from SQL Practice Book

III. RSQLG Framework

The conventional technique for SQL practices creation starts from the data set investigation then an inquiry clarification (question) is made. The educator composes the SQL question for SQL clarification and approves result with the genuine information base which addressed in Fig. 1. The RSQLG switches the conventional interaction by making a SQL inquiry (reply) first.The inquiry from initial step is a contribution to produce question clarification (question) and is approved at the same time as Fig. 2. The progression of RSQLG calculation is addressed in Fig. 3.The data set has a pre-handling module (A) in initial step. This module takes an information data set and most extravagant the info with metadata. The inquiry metadata generator (B) takes the expanded info and question setup to create question metadata and store the result in the inquiry metadata bank (C) in second step. The SQL answer generator (D) develops question metadata from the inquiry metadata bank then, at that point, convert into a SQL question (the appropriate response) when an practice is mentioned. The SQL question generator (E) takes a similar developed inquiry metadata and question format to produce an inquiry clarification (the inquiry) in the equivalent time. The subtleties can be shown as follows:

DOI: 10.48175/IJARSCT-2790

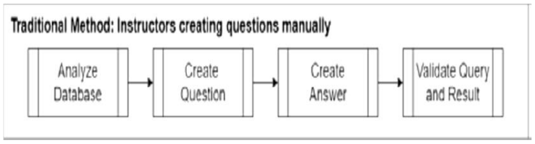
Traditional Method: Instructors creating questions manually
Analyze Database
Create Question
Create Answer
Validate Query and Result

Figure 1. Process of manually generating SQL exercises.

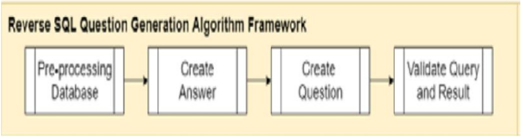
Reverse SQL Question Generation Algorithm Framework
Pre-processing Database
Create Answer
Create Question
Validate Query and Result

Figure 2. Concept in generating SQL exercises using RSQLG.

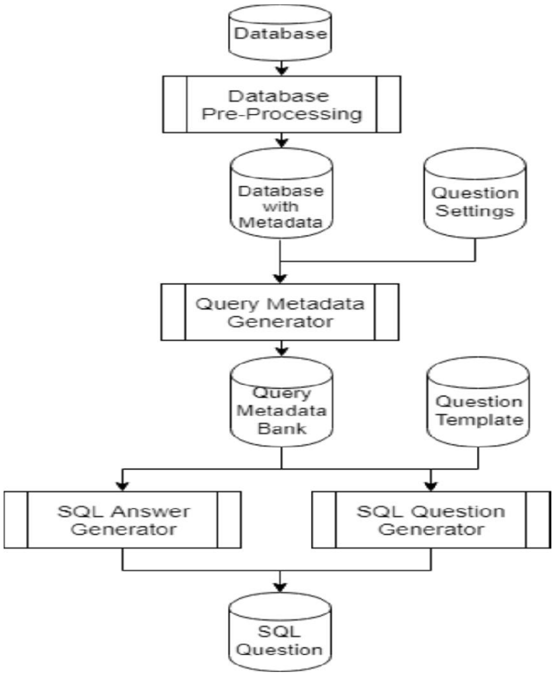

Figure 3. Flow of RSQLG algorithm.

3.1 Database Pre-processing

The RSQLG pre-handling removes the information and metadata which is essential for SQL question creation, for example data set diagram, table blueprint, trait information type, quality information length, trait limitations, relationship and key. The pre-handling creates some amassed information which can be shown in Table II. This information is accommodated inquiry metadata generator. The metadata generator works without genuine data set association. In current cycle, the educator still needs to give the clarification of table and field names in data set metadata physically.

3.2 Query Metadata Generator

The question metadata generator produces the metadata of yield question. The SQL answer generator and SQL question generator utilize this metadata for their cycles. The metadatais changed over to be SQL question (reply) and inquiry clarification (question) separately. The order age stream contains different modules following the SQL question execution request [5] which are; FROM, WHERE, Bunch BY, HAVING, SELECT, ORDER BY and LIMIT. The requirement is portrayed in section IV.

Property	Data Type	Constraints
Existing Values	All	Query for all existing value
Aggregate	All	Query for AVG,MIN,MAX
Table Schema	All	Explain query

The SELECT order group triggers all modules in the whole stream. The calculation picks related tables then, at that point, makes conditions to inquiry the outcome. The INSERT order group contains FROM, WHERE, SELECT and LIMIT modules. The calculation makes the new records which attach the current information with a similar section arranges or do a few changes, for example adding prefix, postfix and erasing a few words. The new records are result from Insert order which come from understudy's reply.

The UPDATE order group incorporates something very similar modules as the INSERT objective. The calculation recovers (at least one) record which be altered by understudy. The interaction utilizes approach same as WHERE proviso in SELECT order age then, at that point, produce new information in one (or more) segment by utilizing a similar methodology as INSERT order age.

The DELETE order group contains FROM, WHERE and LIMITS modules. The calculation makes a condition which performs to erase records with something very similar approach as WHERE statement in SELECT orders generation.

3.3 Query Metadata Bank

Question metadata bank stores the produced metadata in measured articles as opposed to putting away the entire result which can decrease the result size. The inquiry metadata bank put away three item types which are addressed in Fig. 4. The first object is FROM object which put away the table and joined table information. The subsequent item is SELECT article which put away the selectable segments and new qualities (for INSERT, Refresh and DELETE question). The last is WHERE object which put away the legitimate conditions. The question metadata bank keeps the relationship between different objects. The relationship can be explained as the SELECT object can display data from which FROM object, the SELECT object can display data from which WHERE object and the WHERE object can query data from which FROM object.

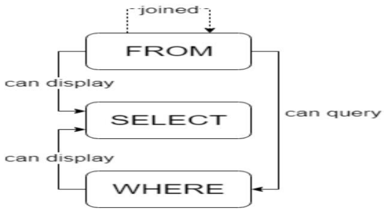

Figure 4. Structure of query metadata bank.

3.4 SQL Answer Generator

To generate an exercise, The SQL answer generator retrieves the data from query metadata bank which associated to the defined objective in question setting. The data from Query metadata bank is combined as a complete query metadata. An example of query metadata which represents in JSON object is shown in Fig. 5. The SQL answer generator generates the SQL query by arranging each part of query from metadata into the format according to the command. The query metadata from Fig. 5 is processed and generated to be the SQL query as example below:

SELECT name, city FROM student WHERE city = “pune”;

```
“command”: “SELECT”,
“columns”: [
                    {“table”:”student”,”attr”:”name”},
                     {“table”:”student”,”attr”:”city”}
             ],
 "condition" : {
                        "exp":
                          [
                           {     "table": "student",
                                     "attr":"name",
                                     "op" : "="
                                     "value" : "pune"}
                          ],
                      "lop" : NULL
                    },
 "order"   :  [
                         {"table": "student",
                         "attr":"name",
                         "order": "DESC"
                         }
                 ]
}
```

Figure 5. The complete query metadata example.

Breakdown	**SELECT**	**Name, city**	**FROM student**	**WHERE**	**city = "pune"**	**;**
Explanation	display	name, city	of all student	whose	city is pune	
SQL Questions	Display name and city of all students from student whose city is pune.					

Figure 6. Example of output from query extraction and translation.

3.5 SQL Question Generator

The SQL question generator accumulates a similar inquiry metadata from SQL answer generator with the inquiry clarification for SQL practice creation. This part is a whiz language clarification. Each piece of inquiry is converted into regular language. A model is shown in Fig. 6. The teachers can indicate example or language of clarification in view of their inclinations by altering the inquiry layout. The inquiry age for JOIN, GROUP BY and sub query requires extra examples. The JOIN has two extra examples which are the primary table is referenced in the table name of inquiry clarification and the subsequent table is referenced in the field name. For GROUP BY design, the clarification contains data about technique to assemble the yield collectively. For sub queries, the condition in each sub query is referenced. A model is addressed in Table III. The underlined words are the extra examples.

Objective	**Query Example**
Join	Displays student last name and department, city from all student whose city is pune.
subquery	Displays student name of from student who has first name is Abc.

IV. Query Generation Constraints

Each piece of the inquiry is created with the imperatives which characterized in question metadata age. The order age requirements show in Table IV. These requirements are significant as a result of the accompanying reasons: -"all column returns" limitation forestalls the outcome set containing each record in the table. -"no line returns" limitation forestalls an unfilled outcome set gotten back from the question. -"deceiving" requirement forestalls the disarray of the objective order. A few orders are tradable, for example IN and equivalent are at some point tradable. -"superfluous order" limitation forestalls question age with order which has no impact. -"DB imperative" requirement forestalls infringement of information base requirements.

Operators / Commands	Constraints	Reason
WHERE clause		
>, >=, <, <=	Must not use with first ordered value.	All row returns, No row returns
UPDATE clause		
Update	INSERT rules apply, but do not change primary key.	DB constraint
INSERT clause		
Primary key	Must equals to highest value plus row number of new data.	DB constraint

V. RSQLG Results and Evaluation

An amount of RSQLG result relies upon the info information base (both construction and informational collection). The MySQL-changed over variant of Oracle's HR test construction scripts and articles [6] are brought to test in this paper and utilized in current customary study hall. The three inquiries from various objective are converted into both English and Thai variants. The inquiries were approved by IT-arranged specialists which assessed all inquiry clarifications of all destinations in part of clarity and reasonable, and the produced answers are right. The case of result is displayed in Table

VI. Conclusion

This paper introduced the first switched SQL question age calculation (RSQLG) which switches the physically composing exercise process from making the inquiry, to starting with an age of reply. The RSQLG took a data set as information then it produced question metadata by investigating data set outline and informational index. The created inquiry metadata needs to coordinate with characterized imperatives. The inquiry metadata can be changed over to substantial SQL question (reply) and clarification of the inquiry in normal language (question). The educators set the language, design and question clarification without anyone else. RSQLG can be utilized to take care of issues in educating and learning SQL. The understudies can pick the target of training without getting copy questions. This benefit further develop opportunity forSQL expertise upgrade. Besides, the educators can diminish their responsibility to make SQL activities to work on their impact since they have the opportunity to create or investigate the newadvancement and innovation to show their understudy moreover. At long last, the activities from the RSQLG meet the learning reason likewise then the taking in outcome from understudy is more productivity. The future work centers around executing this RSQLG calculation in e-learning framework – DBLearn [7] – to assess the learning result and understandability of the inquiries. Besides, the RSQLG actually requires some improvement in complex activities, language support which has more complex language structure and more unsupported orders, for example DDL orders.

<table>
<tr><th>Objective</th><th>Query Question</th><th>SQL Query</th></tr>
<tr><td>SELECT</td><td>Display all data of student who has roll no is 09</td><td>SELECT * FROM student WHERE rollno = 09;</td></tr>
<tr><td>Aggregate</td><td>Display sum of fees of student.</td><td>SELECT AVG(fees) FROM student;</td></tr>
<tr><td>Update</td><td>Changes the fees to 10000 whose name is Abc</td><td>UPDATE fees SET fees = 10000 WHERE Name=”Abc”;</td></tr>
<tr><td>Insert</td><td>Insert data into student table
<table>
<tr><td>Name</td><td>City</td><td>fees</td></tr>
<tr><td>Abc</td><td>Pune</td><td>10000</td></tr>
</table></td><td>INSERT INTO student(Name,City,fees) VALUES ('Abc','Pune', 10000);</td></tr>
</table>

References

[1]. K. Lee, S.-O. Kweon, H. Seo, and G. G. Lee, "Generating grammar questions using corpus data in L2 learning," in Spoken Language Technology Workshop (SLT), 2012 IEEE, 2012, pp. 443-448.

[2]. N. Funabiki, T. Mohri, and S. Yamaguchi, "Toward personalized learning in JPLAS: Generating and scoring functions for debugging questions," in Consumer Electronics, 2016 IEEE 5th Global Conference on, 2016, pp. 1-4.

[3]. S. Abdul Khalek and S. Khurshid, "Automated SQL query generation for systematic testing of database engines," in Proceedings of the IEEE/ACM international conference on Automated software engineering, 2010, pp. 329-332.

[4]. N. Greenberg and P. Nathan, Introduction to Oracle9i: SQL: Oracle University, 2001.

[5]. B.-G. Itzik, L. Kollár, D. Sarka, and S. Kass, Inside Microsoft SQL Server 2008-T-SQL Querying: Microsoft Pr., 2009.

[6]. HR Sample Schema Scripts and Objects. Available:https://docs.oracle.com/en/database/oracle/oracledatabase/18/comsc/HR-sample-schema-scripts-and-objects.html

[7]. S. Nalintippayawong, K. Atchariyachanvanich, and T. Julavanich,"DBLearn: Adaptive e-learning for practical database course—An integrated architecture approach," in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing(SNPD), 2017 18th IEEE/ACIS International Conference on, 2017, pp. 109-114.