



University of
Strathclyde
Glasgow

AI-Assisted Generation of SQL Comprehension Questions

Martin Goodfellow, Alasdair Lambert, Andrew Fagan, Robbie
Booth

DataEd 2026
24th March

Motivation

- Students frequently write functionally correct SQL, yet may not fully understand how those queries operate [1]
- Comprehension-oriented tasks have been shown to provide insight into how students interpret and reason about program behaviour [2]
- Individually questioning students or manually creating questions is time consuming and doesn't easily scale

[1] D. Miedema, E. Aivaloglou, G. Fletcher, Identifying SQL Misconceptions of Novices: Findings from a Think-Aloud Study. ICER 2021

[2] Carsten Schulte, Tony Clear, Ahmad Taherkhani, Teresa Busjahn, and James H. Paterson. 2010. An introduction to program comprehension for computer science educators. ITiCSE-WGR '10

Motivation

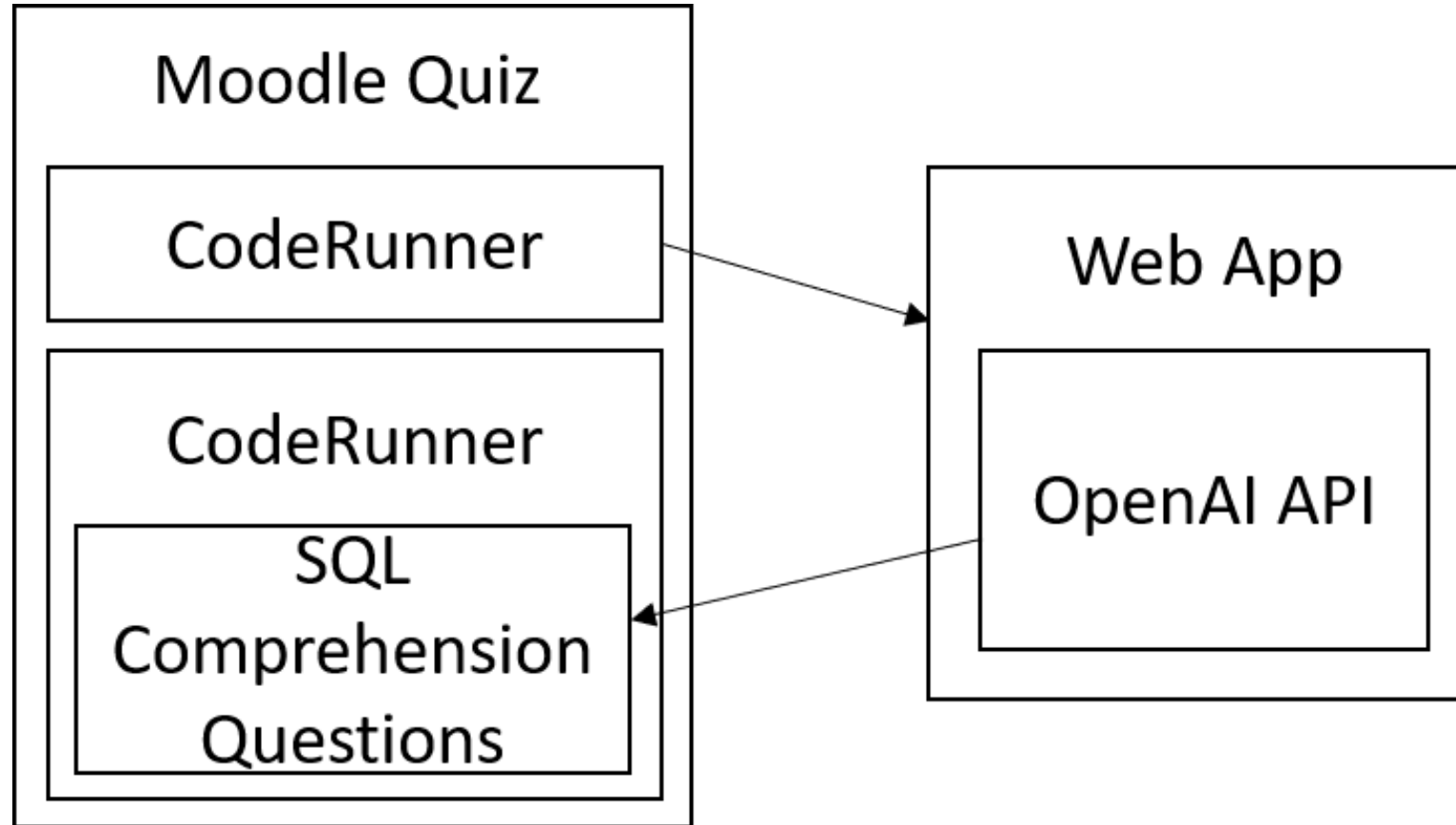
- Recent advances in GenAI have enabled the automatic generation of code-based comprehension questions [3]
- Prior research on automatic SQL generation focuses on producing syntactically valid queries [4]
- Existing automated SQL assessment tools compare student queries to model solutions or generate structural hints [5]

[3] M. Goodfellow, R. Booth, A. Fagan, A. Lambert, AutoMCQ - Automatically Generate Code Comprehension Questions using GenAI. ITiCSE 2025

[4] D. R. Quan Do, Rajeev K. Agrawal, V. N. Gudivada, Automatic Generation of SQL Queries. ASEE Annual Conference 2014

[5] C. Kleiner, F. Heine, Enhancing Feedback Generation for Autograded SQL Statements to Improve Student Learning. ITiCSE 2024

Approach



[6] Martin Goodfellow, Andrew Abel, Konstantinos Liaskos, and John Levine. 2024. Automated Marking in Undergraduate Programming Classes. CEP '24

Example - Prompt

- System prompt: You are an educational assistant specializing in computer science. Your task is to analyse students' SQL code for the beginner database class and generate thoughtful multiple-choice questions that can help them understand and improve their SQL skills. You should try and make good distractor options to really test students' understanding.
- Parameters: question_count, question, question_topics, database_schema, student_query
- How to structure the response – JSON

SQL Questions

- Self-join: List all pairs of customer IDs who live on a street with the same name but in a different city
- Natural join: List all customer IDs, dates and quantities of transactions containing products named Apples
- Group By: Return a list of the number of stores per city

Table name	Attributes
customer	<u>cID</u> , cName, street, city
store	<u>sID</u> , sName, street, city
product	<u>pID</u> , pName, suffix
shoppinglist	<u>cID</u> , <u>pID</u> , quantity, date
purchase	<u>tID</u> , <u>cID</u> , <u>sID</u> , <u>pID</u> , date, quantity, price
inventory	<u>sID</u> , <u>pID</u> , date, quantity, unit-price

List all pairs of customer IDs who live on a street with the same name but in a different city.

Customer(cID, cName, street, city)

Answer: (penalty regime: 10, 20, ... %)

```

1 SELECT a.cID, b.cID
2 FROM customer a
3 JOIN customer b
4 ON a.street = b.street AND a.city <> b.city AND a.cID < b.cID;

```

	Expected		Got		
✓	cID	cID	cID	cID	✓
	---	---	---	---	
	0	3	0	3	
	1	3	1	3	

Passed all tests! ✓

cID	cName	street	city
0	Martin	High Street	Glasgow
1	Alasdair	High Street	Glasgow
2	Andrew	George Street	Edinburgh
3	Robbie	High Street	Edinburgh
4	Clemens	Duke Street	Glasgow

Example - Prompt

```
{  
  "question_count": 2,  
  "question": "List all pairs of customer IDs who live on a street with the same  
              name but in a different city. Customer (cID, CName, street, city)"  
  "question_topics": "Self-join",  
  "database_schema": "expressed in SQL",  
  "student_query": "str - the student's final query"  
}
```

Example: Self-Join

```
SELECT a.cID, b.cID
FROM customer a
JOIN customer b
  ON a.street = b.street
  AND a.city <> b.city
  AND a.cID < b.cID;
```

```
SELECT cID
FROM customers
WHERE street = street
AND city <> city;
```

Q: Why is `a.city <> b.city` included in the self-join condition?

- (a) To include only customers from the same city.
- (b) To ensure that the paired customers are from different cities.**
- (c) To return customers regardless of their city.
- (d) To filter out customers who live on the same street.

Q: What is the correct syntax for referencing the same column in a self-join condition?

- (a) Use table aliases to distinguish between the two instances of the column.**
- (b) Use a single table reference without aliases.
- (c) Use different column names for each instance.
- (d) Reference the column using only its name without specifying the table.

Example: Natural join

```
SELECT cID, date, quantity
FROM purchase
NATURAL JOIN product
WHERE pName = 'Apples';
```

```
SELECT s.cID, s.date, s.quantity
FROM shoppinglist s, product p
WHERE p.pName = "Apples";
```

Q: What is the purpose of using NATURAL JOIN in the student's SQL query?

- (a) To restrict the results to only matching records in both tables.
- (b) To create a subquery that rearranges the results based on product names.
- (c) To join all columns from both tables regardless of matching names.
- (d) To automatically join tables based on columns with the same name.**

Q: What is the main issue with the query when trying to retrieve transactions containing products named Apples?

- (a) The query does not include a join condition to link the 'shoppinglist' and 'product' tables.**
- (b) The query uses the wrong delimiter for string values.
- (c) The query selects the wrong columns.
- (d) The query does not filter out products that are not in stock.

Example: Group By

```
SELECT city, COUNT(*) AS num_stores  
FROM store  
GROUP BY city;
```

```
SELECT city, COUNT(city)  
FROM store  
GROUP BY city;
```

Q: What will be the output of the COUNT(*) function in the context of the given SQL code?

- (a) **The total number of stores for each city.**
- (b) The average number of stores across all cities.
- (c) The number of unique cities in the store table.
- (d) The total inventory of all items in the store.

Q: In the query, what will happen if there are entries in the store table without a city specified?

- (a) Those entries will be included with a null city.
- (b) The query will fail due to invalid data.
- (c) **Those entries will be excluded from the results.**
- (d) The city will be counted as a blank entry.

Conclusions and Future Work

- Our tool can generate coherent, query-specific, multiple-choice questions based on student-submitted SQL at low cost
- Can be used to identify areas of weak understanding and provide a useful study tool in a scalable and cost-effective manner
- Future Work:
 - improve detection of invalid or irrelevant questions
 - empirical evaluation of the quality and educational effectiveness of the generated questions

References

- [1] D. Miedema, E. Aivaloglou, G. Fletcher, Identifying SQL Misconceptions of Novices: Findings from a Think-Aloud Study. ICER 2021
- [2] Carsten Schulte, Tony Clear, Ahmad Taherkhani, Teresa Busjahn, and James H. Paterson. 2010. An introduction to program comprehension for computer science educators. ITiCSE-WGR '10
- [3] M. Goodfellow, R. Booth, A. Fagan, A. Lambert, AutoMCQ - Automatically Generate Code Comprehension Questions using GenAI. ITiCSE 2025
- [4] D. R. Quan Do, Rajeev K. Agrawal, V. N. Gudivada, Automatic Generation of SQL Queries. ASEE Annual Conference 2014
- [5] C. Kleiner, F. Heine, Enhancing Feedback Generation for Autograded SQL Statements to Improve Student Learning. ITiCSE 2024
- [6] Martin Goodfellow, Andrew Abel, Konstantinos Liaskos, and John Levine. 2024. Automated Marking in Undergraduate Programming Classes. CEP '24



University of
Strathclyde
Glasgow