Let's Ask the Experts: Insights into Student Misconceptions about SQL

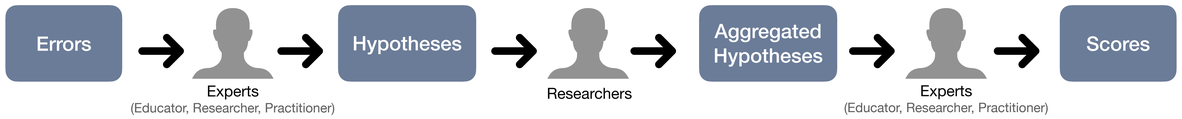
Daphne Miedema
d.e.miedema@tue.nl

Efthimia Aivaloglou
e.aivaloglou@liacs.leidenuniv.nl

George Fletcher
g.h.l.fletcher@tue.nl

The Structured Query Language (SQL) has been introduced in 1970, and quickly became the standard query language for Relational Databases. It is ubiquitous in both practice and education. Various researchers have published works on the errors that novices make while writing queries. However, until recently, not much attention was paid to the underlying reasons why novices made so many mistakes. In a previous paper, we explored these underlying reasons, also called misconceptions, from the student perspective. We gave students query formulation problems and asked them to solve those while explaining to us what they were thinking. We followed this up by exploring the perspectives of SQL experts -researchers, practitioners and educators-. Their expertise on errors and solutions gave us new insights into other misconceptions that may lead to the same error. The result are 11 lists of misconceptions that novices might hold. Below, we present the top-ranked expert hypothesis and juxtapose it with the student hypothesis.

Delphi protocol



1. Using ≠ instead of != or <>

```
SELECT *
FROM store
WHERE city ≠ "Eindhoven";
```

Expert
Students are familiar with ≠ and find it more intuitive than <> and !=.

Student
Interference of previous course knowledge from either Mathematics or Relational Algebra.

2. AVG in WHERE instead of HAVING

```
GROUP BY t.sID
FROM transaction as t
WHERE AVG(t.quantity) > 4;
```

Expert
Students associate the application of a filter with WHERE.

Student
Misunderstanding SQL syntax and its internal (in)consistency.

3. JOIN in all tables that you need the primary keys of

```
SELECT c.cID, t.date,
     t.quantity
FROM customer as c,
transaction as t,
product as p
WHERE t.pID = p.pID
AND c.cID = t . cID
AND p.pName = "Apples"
```

Expert and Student
Students have created a template query/pattern and apply it without understanding it.

4. Missing second table for self-join

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

Expert
Students do not think of a second access as a separate instance of a table.

Student
Misunderstanding the scope of elements in a query.

5. Lack of understanding of when to apply GROUP BY

```
SELECT city, street,
   sName, sID
FROM store
GROUP BY city
ORDER BY city ASC;
```

Expert
Students do not understand the implication of grouping

Student
Misunderstanding SQL syntax and its internal (in)consistency.

Hypotheses types

- Previous course knowledge
- Generalisations
- Incorrect or incomplete mental model
- Language related

6. Missing parts of keywords

```
SELECT *
FROM store AS s
ORDER s.city ASC
```

Expert
Students forget the correct syntax because most other SQL keywords are single-valued.

Student
Remembering only the core part of a keyword as it captures the main meaning.

For more information about student misconceptions in SQL, find our open-access paper here:

7. Extra condition when not required

```
SELECT c1.cID, c2.cID
FROM customer c1, customer c2
WHERE c1.cID <> c2.cID
AND c1.street = c2.street
AND c1.city <> c2.city ;
```

Expert
Students do not realise that being a different customer follows from the different cities.

Student
Lacking knowledge on primary keys.

8. Ambiguous column name because of missing alias

```
SELECT unit_price
FROM inventory
INNER JOIN product
ON pID = pID
WHERE pName = "Bananas"
```

Expert
Student knew which pID they meant and feel it is self-explanatory.

Student
Lack of knowledge or experience.

9. Using comma instead of AND in WHERE clause

```
SELECT p.pName
FROM inventory as I,
  product as p
WHERE
([subquery]) < i.unit_price,
p.pID= i.pID;
```

Expert
Student was sloppy in writing the query.

Student
Misunderstanding SQL syntax and its internal (in)consistency.

10. Thinking DISTINCT will take the first item of a list

"If I would have used a DISTINCT before, on the unit_price, then you'd have taken the first unit_price and compared it with the unit_price of the primary query. [. . .] If I used DISTINCT over this C.unit_price, I'd have given only one unit_price and probably that would have been the first one, because that is what it will find first. And then it would just compare [the price] with that one."

Expert
Students mix the semantics of 'a single' and 'the first'.

Student
Believing DISTINCT will take the first item of a list (because of being presented with very selective examples of DISTINCT queries.)

11. Lack of understanding subqueries

```
SELECT t.cID, t.date,
t.quantity
FROM transaction as t
WHERE (
   SELECT pName
   FROM product
   WHERE pID = t.pID
   AND pName = "Apples")
```

Expert
Students prefer a loop-style query, as you would do in programming.

Student
Lack of knowledge or experience.

Conclusion

Research Question 1: *How do experts interpret the underlying causes of student errors in SQL?*
The varying backgrounds of the experts meant that they approached the problem from different angles. We found up to 8 different hypotheses per error, some were deemed more likely than others. Sometimes all hypotheses were closely related, but the experts typically considered a wide spectrum of underlying causes.

Research Question 2: *What are the differences in the perspectives of students and experts on errors in SQL?*
Although sometimes there was a difference in focus, overall the students misconception hypotheses were ranked highly by the experts too. For example, for error 2, the wordings are different but the core of the hypotheses for this error is the same. The only error where the students and experts were not in agreement was error 6.