

Visual Query Representations for Supporting SQL Learners

Daphne Miedema, George Fletcher

Writing SQL queries is difficult.

Students struggle with writing correct SQL queries [1]:

1. writing overly complex queries (example in Figure 1)
2. losing track of writing due to complexity
3. incorrectly interpreting commands (Figure 2)

This is due to the cognitive load associated with writing queries [2]. Therefore, novice SQL users should receive support while working on problems. In this poster we present a visual representation of SQL queries that can alleviate these issues by:

1. increasing understanding of complex SQL commands
2. simplifying the writing of nested queries
3. easing interpretation of existing queries
4. reducing the number of errors in table and column names

```
query5 = '''
SELECT pName, sName
FROM purchase
WHERE price = MAX (price) and pName = (
SELECT pName
FROM product
WHERE pID = (
SELECT pID
FROM purchase
WHERE price = MAX (price)) and sName = (
SELECT sName
FROM store
WHERE sID = (
SELECT sID
FROM purchase
WHERE price = MAX (price))))'''
```

```
query2 = '''
SELECT pur.cID, pur.date
FROM purchase pur, product prod
WHERE prod.pID = pur.pID
GROUP BY pur.cID, pur.date
HAVING prod.pName = 'Coffee'
AND prod.pName = 'Onions'
'''
```

Figure 1: Participant 99 writes complex queries.

Figure 2: Participant 15 interprets incorrectly.

How can we help?

Before settling on a design, we ran a user test, asking six PhD students in the Database Research Group at TU/e to represent a set of SQL queries in any visual way they found intuitive. Their representations can be grouped in two categories:

1. Technical details such as \forall and \exists (Figure 3).
2. Graph characteristics, such as arrows and links (Figure 4).

This latter category of results is supported by literature, as the data characteristics that are implicit in relational databases become explicit in graph visualizations. Based on the designs by the participants, existing literature, and our intuitions, we designed visual representations as in Figures 5 to 7.

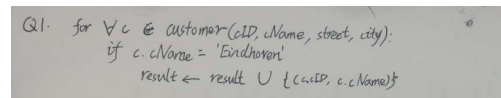


Figure 3: An example of a technical representation

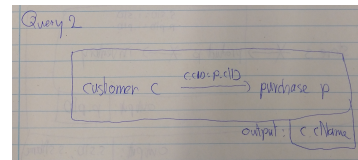


Figure 4: An example of a graph representation.

How is our solution evaluated?

With these visual representations in hand, we ran another user study in which we asked five people to use a database system that incorporated our visualizations for answering query formulation problems [1]. These participants were disjoint from the previous set, but all had some SQL experience.

The system was found to be highly usable and clean. Participants were particularly positive about the representation [1]:

- Participant 3: "[The representation] is a good idea because it helps to keep track of what you are doing"
- Participant 5: "For very complex queries, this may help find mistakes in the query."

In practice, the visual representation was underutilized, possibly because no similar functionality exists. This may lead participants to have low exposure, which causes them to forget to use this mechanism [3]. One solution is to make the visualization reactive to the query text field.

Some examples of the visual SQL representation as we designed it:

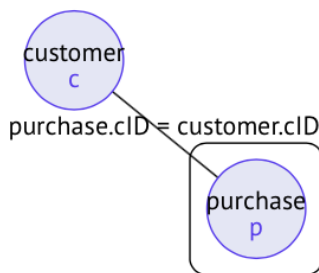


Figure 5: People who have made a purchase.

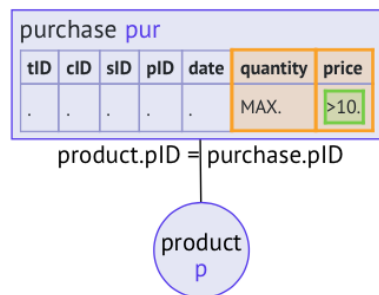


Figure 6: The max amount of items in a purchase above 10 euros.

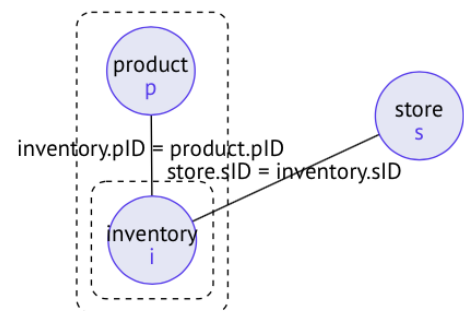


Figure 7: Stores that have all products in stock (there is no item not in inventory).

References

[1] Miedema, D. (2019). *Towards successful interaction between Humans and Databases*. MSc Thesis, Eindhoven University of Technology.
 [2] Mitrovic, A. (1998). Learning SQL with a computerized tutor. In *ACM SIGCSE bulletin*, 307-311.
 [3] Zajonc, R. (1968). Attitudinal effects of mere exposure. *Journal of Personality and Social Psychology*, 9(2), 1-27.

