A Proposal for Revising SQL Error Taxonomies Based on Automated Detection

Davide Ponzini[1,2]
davide.ponzini@edu.unige.it

Giovanna Guerrini[1]
giovanna.guerrini@unige.it

Barbara Catania[1]
barbara.catania@unige.it

[1]Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi (DIBRIS)
University of Genoa, Italy

[2]Dipartimento di Lingue e Culture Moderne (DLCM)
University of Genoa, Italy

March 24th, 2026

Context

SQL in CS Education

- SQL is a core topic in data systems education
- Students often struggle with SQL despite its apparently intuitive declarative nature [3, 5, 9]
  - Declarative nature contrasts with imperative languages learned earlier
- Existing DBMS error messages are usually not pedagogically helpful [6, 7, 9]

Context

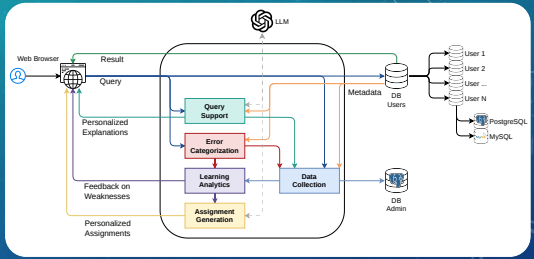
SQL Error Taxonomies

- Research on SQL education has addressed:
  - Student error classification [8]
  - Misconceptions behind errors [3]
  - Automated feedback and repair tools [1, 2]
- Widely used taxonomy: Taipalus et al. [8]
  - Syntax errors, Semantic errors, Logic errors, and Complications
- Limitations:
  - Designed for **human annotation**
  - Not directly suitable for **automation**

Motivation (1)

From Taxonomy to Automation

- We are developing **LenSQL** [4]
  - Automatic SQL error detection and classification
  - Adaptive feedback
  - Learning analytics
  - Personalized exercise generation
- Taxonomies are the backbone of these functionalities



Motivation (2)

From Taxonomy to Automation

- Problems:
  - Hard to translate into algorithmic rules
  - Ambiguous categories
  - Some errors → multiple / no categories
- **Goal:** revise taxonomy for automated classification

Revision Principles

- **Clear naming**
  - Avoid ambiguous or overlapping definitions
- **Behavior-based definitions**
  - Based only on observable query properties
  - Not on inferred student intent
- **Reduce ambiguity**
  - Merge or reorganize overlapping categories
- **Pedagogy preserved**
  - Changes improve both detection and interpretation

Taxonomy Revision (1)

What We Changed

- **Clearer naming**
  - "Join" → **Table reference** vs **Join condition**
- **Merged / removed ambiguous categories**
  - e.g., multiple definitions → **Ambiguous column**
- **Split overly broad categories**
  - e.g., wildcard errors → **wrong** vs **invalid**

Taxonomy Revision (2)

What We Changed

- **Added missing errors**
  - Subqueries: **missing quantifier**
  - Set operations: **different tuples**
  - Clauses: missing / extraneous / incorrect
- **Reclassified errors**
  - Moved some **semantic** $\rightarrow$ **logical**
  - Focus on query behavior, not intent

Example (1)

- **Query:** `SELECT unit_price > 10 FROM inventory;`

- Original taxonomy:
  - Syntax error *("restriction in SELECT clause")*
- Revised taxonomy:
  - Valid SQL $\rightarrow$ no syntax error
  - Classification depends on **data demand**
    - E.g. *"Extraneous expression"* or *"Missing WHERE clause"*

Example (2)

- **Query:** `SELECT * FROM store, inventory;`
- **Expected:**
```
SELECT * FROM store s
    JOIN inventory i ON s.sID = i.sID
    JOIN product p ON p.pID = i.pID;
```

- Original taxonomy:
  - Semantic error *("omitting a join")*
  - Logic error *("missing join")*
- Revised taxonomy:
  - *"omitting a join"* $\rightarrow$ **"missing join condition"**
  - *"missing join"* $\rightarrow$ **"missing table reference"**

Why It Matters

- Taxonomy as a **bridge**:
  - Query $\rightarrow$ Error $\rightarrow$ Misconception
- Behavior-based errors are:
  - Detectable
  - Interpretable
  - Mappable to misconceptions
- This enables:
  - Reliable automatic error detection
  - More precise feedback
  - Learning analytics and personalization

Take-Home Message

- Existing taxonomies are not designed for automation
- We propose a **practice-driven revision**
- Core shift:
  - From *intent-based* $\rightarrow$ **behavior-based** errors
- Goal:
  - Errors that are **detectable, interpretable, and pedagogically useful**

References I

[1] Bikash Chandra et al. "Data Generation for Testing and Grading SQL Queries". In: *The VLDB Journal* 24.6 (2015), pp. 731–755.

[2] Yihao Hu et al. "Qr-Hint: Actionable Hints Towards Correcting Wrong SQL Queries". In: *Proc. of the ACM Conf. on Management of Data* 2.3 (2024), pp. 1–27.

[3] Daphne Miedema, Efthimia Aivaloglou, and George Fletcher. "Identifying SQL Misconceptions of Novices: Findings from a Think-aloud Study". In: *ACM Inroads* 13.1 (2022), pp. 52–65.

References II

[4] Davide Ponzini, Barbara Catania, and Giovanna Guerrini. "Enhancing SQL Learning Through Generative AI and Student Error Analysis". In: *New Trends in Database and Information Systems*. Springer, 2025, pp. 118–128.

[5] Toni Taipalus. "Explaining Causes behind SQL Query Formulation Errors". In: *2020 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2020, pp. 1–9.

[6] Toni Taipalus. "SQL: A Trojan Horse Hiding a Decathlon of Complexities". In: *Proc. of the 2nd Int'l Workshop on Data Systems Education*. 2023, pp. 9–13.

[7] Toni Taipalus and Hilkka Grahn. "Framework for SQL Error Message Design: A Data-Driven Approach". en. In: *ACM Trans. on Software Engineering and Methodology* (2023).

References III

[8] Toni Taipalus, Mikko Siponen, and Tero Vartiainen. "Errors and Complications in SQL Query Formulation". In: *ACM Trans. on Computing Education* 18.3 (2018), pp. 1–29.

[9] Jun Yang et al. "What Teaching Databases Taught Us about Researching Databases: Extended Talk Abstract". In: *Proc. of the 3rd Int'l Workshop on Data Systems Education*. 2024, pp. 1–6.

Our revised taxonomy

https://github.com/DavidePonzini/sql_error_taxonomy/tree/revision

THANK YOU FOR YOUR ATTENTION

Any questions?