Do You Know SQL?
About Semantic Errors in SQL Queries

Christian Goldberg
goldberg@informatik.uni-halle.de

Institut für Informatik
Martin-Luther-Universität Halle-Wittenberg

TLAD 2009
Birmingham, 6th July 2009
Contents

1. Introduction

2. Classes of Semantic Errors
   - Unnecessary Complications
   - Violation of Standard Patterns
   - Further Classes of Semantic Errors

3. Base Data

4. Statistics and Evaluation
   - Error Distribution
   - Most frequent semantic errors
   - Possible Causes and Solutions
Classification of Errors

Errors in SQL queries

- Syntactic errors
  - Task must be known
- Semantic errors
  - Task independent
Example (Inconsistent Condition)

```sql
SELECT ENAME
FROM EMP
WHERE JOB = 'CLERK' AND JOB = 'MANAGER'
```

- Empty result in all database states (certainly not intended)
- Inconsistent condition is a frequent student error
- In general not decidable
DB schema

SQL query

sqllint

semantic error warnings
Unnecessary Complications

- Unnecessarily complicated query
  → “probably not intended”

Situation:

1. User wrote query \( A \).
2. \( B \) exists equivalent to \( A \).
3. \( B \) is significantly simpler than \( A \):
   - \( B \) results from \( A \) by deleting parts of the query.
Complications possible in all query parts:

- **SELECT**  Constant / duplicate output columns
- **FROM**    Unused tuple vars, Unnecessary joins
- **WHERE**   Implied, tautological or inconsistent sub-conditions, Unnecessary general comparison operator
- **GROUP BY** singleton groups, only one group

... 

**Entire query unnecessary** (Inconsistent Condition)
Example (singleton groups)

SELECT EMPNO, MAX(SAL)
FROM EMP
WHERE JOB = 'MANAGER'
GROUP BY EMPNO

Example (comparison operator)

SELECT ENAME, SAL
FROM EMP
WHERE SAL >= (SELECT MAX(SAL) FROM EMP)
Violation of Standard Patterns

- Missing join conditions
- Uncorrelated EXISTS-subqueries
- SELECT clause of subquery uses no tuple variable from the subquery
- Conditions in subquery that can be moved up
- Comparison between different domains
- HAVING without GROUP BY
- DISTINCT in SUM and AVG
- Wildcards without LIKE
Further Classes of Semantic Errors

- Duplicates (Unnecessary DISTINCT, Many duplicates)
- Inefficient Formulations (Inefficient HAVING/UNION)
- Possible Runtime Errors (SELECT INTO that might return more than one tuple)
- “Bad Style” (Inconsistent use of defaults)

A quite complete list of over 40 semantic errors can be found in:
Five “Databases I” exams analyzed with 22 SQL exercises

<table>
<thead>
<tr>
<th>Exam</th>
<th>Part.</th>
<th>Exercises SQL</th>
<th>Points SQL</th>
<th>Points Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final 02/03</td>
<td>67</td>
<td>4</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Midterm 03/04</td>
<td>153</td>
<td>3</td>
<td>9</td>
<td>23</td>
</tr>
<tr>
<td>Final 03/04</td>
<td>148</td>
<td>3</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Final 05/06</td>
<td>40</td>
<td>6</td>
<td>18</td>
<td>37</td>
</tr>
<tr>
<td>Final 08/09</td>
<td>53</td>
<td>6</td>
<td>15</td>
<td>35</td>
</tr>
</tbody>
</table>

Classification according to degree of difficulty:
- beginner (6 exercises)
- intermediate (9 exercises)
- advanced (7 exercises)
Analytical Result for 1411 queries in five exams

- Correct: 28%
- Semantic: 18%
- Both: 14%
- Syntax: 8%
- Wrong Task: 8%
- Not Counted: 8%

Do You Know SQL?
• Normalized distribution per difficulty class:

<table>
<thead>
<tr>
<th>Difficulty Class</th>
<th>Correct</th>
<th>Sem.</th>
<th>Syntax</th>
<th>Both</th>
<th>N.C.</th>
<th>W.T.</th>
</tr>
</thead>
<tbody>
<tr>
<td>beginner</td>
<td>40.71</td>
<td>27.67</td>
<td>10.47</td>
<td>8.5</td>
<td>2.37</td>
<td>10.28</td>
</tr>
<tr>
<td>intermediate</td>
<td>28.22</td>
<td>22.63</td>
<td>17.56</td>
<td>13.26</td>
<td>9.88</td>
<td>8.45</td>
</tr>
<tr>
<td>advanced</td>
<td>13.44</td>
<td>21.15</td>
<td>26.87</td>
<td>22.03</td>
<td>12.11</td>
<td>4.41</td>
</tr>
</tbody>
</table>
## Most frequent semantic errors

<table>
<thead>
<tr>
<th>Ratio</th>
<th>Semantic Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>15%</td>
<td>Missing join condition</td>
</tr>
<tr>
<td>13%</td>
<td>Many duplicates</td>
</tr>
<tr>
<td>11%</td>
<td>Unnecessary join</td>
</tr>
<tr>
<td>8%</td>
<td>Inconsistent condition</td>
</tr>
<tr>
<td>6%</td>
<td>Unnecessary argument of COUNT</td>
</tr>
<tr>
<td>5%</td>
<td>Implied, tautological or inconsistent subcondition</td>
</tr>
<tr>
<td>5%</td>
<td>Unnecessary DISTINCT</td>
</tr>
</tbody>
</table>

- Percentages are relative to all detected semantic errors
Possible Causes and Solutions

“I thought it will be joined if I type it under FROM.”

- Lack of preparation
- Absence from lectures and exercises
- Improperly reading of given tasks
- Insufficient experience in programming SQL
- But also: Insufficient comprehension of underlying DB schema

Sometimes fewer errors by making use of connection graphs and discussion
Possible Causes and Solutions

“I thought it will be joined if I type it under FROM.”

- Lack of preparation
- Absence from lectures and exercises
- Improperly reading of given tasks
- Insufficient experience in programming SQL
- But also: Insufficient comprehension of underlying DB schema

Sometimes fewer errors by making use of connection graphs and discussion
Current database systems print no warnings, only error messages if query is not executable.

We develop a semantic checker for SQL called sqllint.

The paper gives a survey of how often and which semantic errors appear.

Sensible error message possible in nearly a quarter of all cases.

For detailed exam descriptions and sqllint prototype, see:

http://dbs.informatik.uni-halle.de/sqllint/
Further Literature


- Stefan Brass, Christian Goldberg: Proving the Safety of SQL Queries. In: Proceedings of the Fifth International Conference on Quality Software (QSIC’05), 197–204, 2005