Semantic Errors in SQL Queries: Exam Evaluation 2008-01

Christian Goldberg
Martin-Luther-Universität Halle-Wittenberg
goldberg@informatik.uni-halle.de

Abstract

We investigate classes of SQL queries which are syntactically correct, but certainly not intended, no matter for which task the query was written. For instance, queries that are contradictory, i.e. always return the empty set, are obviously not intended. Current database management systems, e.g. Oracle, execute such queries without any warning.

In this evaluation, we give a statistic of such errors for one special exam and list the concerning SQL exercises and their possible solutions. Section 1 contains important data of the analyzed exam. In section 2 we explain the database scheme(s) that is/are used in the listed exercises together with their possible solutions in section 3. Section 4 conducts a survey on the number and sorts of occurred semantic errors.

1 Exam Data

Lecture Title	:	Database Systems I
Term	:	Winter term 2008/2009
Lecturer	:	Prof. Dr. Stefan Brass
University	:	Martin-Luther-University Halle, Germany
Analysis	:	Christian Goldberg
Date of Analysis	:	April 2009
Error Code Reference	:	[1]

2 Underlying Database Scheme

In the following exercises, we use a simplified database scheme for pyrotechnic articles which stores information about articles, fireworks and ignitions:

```
ARTICLE(ANO, PROD, DESC, TYPE, DUR, HEIGHT, PRICE)
FIREWORK(FID, TITLE, DATE)
IGNITION(FID→FIREWORK, CHANNEL, TIME, ANO→ARTICLE, AMOUNT)
```

PROD contains the producer of a pyrotechnic article and DESC its description. Both together form a compound alternate key. The column TYPE may contain a “F” (fountain), “B” (bouquet shell) or “R” (roman candle). DUR is the duration in seconds, HEIGHT is given in meters. IGNITION contains the points in time (channels), when pyrotechnic articles have to be ignited. The time when a particular channel is ignited is given in seconds.

3 Analyzed Exercises and Possible Solutions

The exam “Database Systems I” in winter term 2008/2009 contained 14 exercises about logic, relational calculus, SQL, database modelling, transactions and locking. The first 5 analyzed SQL queries resulted in 15 out of 35 points. The last exercise was a bonus question for which the students could get two bonus points. The 53 participating students had 120 minutes to solve the exercises and were allowed to use the lecture script or other notes but no electronic resources.

It was pointed out that unnecessary complications, unnecessary `DISTINCT` and many duplicates may result in a deduction of points.

3.1 Exercise 3a)

List all fireworks with title and date in which volcanos or roman candles were ignited, i.e. which contain articles with the type “V” or “R”.

```
SELECT DISTINCT F.TITLE, F.DATE
FROM   FIREWORK F, IGNITION I, ARTICLE A
WHERE  F.FID=I.FID
AND    I.ANO=A.ANO
AND    (A.TYPE=’V’ OR A.TYPE=’R’)
```

3.2 Exercise 3b)

Request the producer and description of all articles that are never ignited in a firework (i.e. that do not appear in `IGNITION`).

```
SELECT A.PROD, A.DESC
FROM   ARTICLE A
WHERE  NOT EXISTS (SELECT *
                   FROM   IGNITION I
                   WHERE  I.ANO=A.ANO)
```

3.3 Exercise 3c)

Print out the safety clearance for every firework (given by `FID` and `TITLE`). The (simplified) safety clearance is calculated by the maximum height of all ignited articles multiplied by 0.8 (80 percent of the height). This column shall be named `CLEARANCE`.

```
SELECT   F.FID, F.TITLE, MAX(A.HEIGHT)*0.8 AS CLEARANCE
FROM     FIREWORK F, IGNITION I, ARTICLE A
WHERE    F.FID=I.FID
AND      I.ANO=A.ANO
GROUP BY F.FID, F.TITLE
```

3.4 Exercise 3d)

Which are the articles with the maximum price? Specify producer, description and price.

```
SELECT PROD, DESC, PRICE
FROM   ARTIKEL
WHERE  PRICE=(SELECT MAX(PRICE)
              FROM   ARTICLE)
```

3.5 Exercise 3e)

Generate a list of all articles by specifying PROD and DESC and the number of fireworks (column named FIREWORKS) they are ignited in, including the 0 (zero) if an article was never ignited. Rank the output according to this number, with the highest number first, producer and description.

```
SELECT   A.PROD, A.DESC, COUNT(*) AS FIREWORKS
FROM     ARTICLE A, IGNITION I
WHERE    A.ANO=I.ANO
GROUP BY A.PROD, A.DESC
UNION ALL
SELECT   A.PROD, A.DESC, 0 AS FIREWORKS
FROM     ARTICLE A
WHERE    NOT EXISTS (SELECT *
                     FROM   IGNITION I
                     WHERE  I.ANO=A.ANO)
ORDER BY FIREWORKS DESC, PROD ASC, DESC ASC
```

Here, a more elegant solution with an outer join is possible, too:

```
SELECT   A.PROD, A.DESC, COUNT(FID) AS FIREWORKS
FROM     ARTICLE A LEFT OUTER JOIN IGNITION I
         ON A.ANO=I.ANO
GROUP BY A.PROD, A.DESC
ORDER BY FIREWORKS DESC, PROD ASC, DESC ASC
```

Note, that in this case the argument of count is essential as null values must not be counted.

3.6 Bonus question

Generate a list of ignited articles for every firework by specifying FID, ignition time and article description. The end of each firework shall also be listed. The end time is the maximum sum of ignition time and duration over all channels. Use the string 'end' as description. Rank the output according to FID and the ignition time.

```
SELECT   I.FID, I.TIME AS TIME, A.DESC AS DESCRIPTION
FROM     IGNITION I, ARTICLE A
WHERE    I.ANO=A.ANO
UNION ALL
SELECT   I.FID, MAX(I.TIME+A.DUR) AS TIME, 'end' AS DESCRIPTION
FROM     IGNITION I, ARTICLE A
WHERE    I.ANO=A.ANO
GROUP BY I.FID
ORDER BY I.FID, TIME
```

4 Statistics

The list of error types mentioned in [1] is based on our experience from grading a large number of exams and homework. (Error 1a is new and not mentioned in [1], it means: Unnecessary outer query.) After this error taxonomy was finished, we analyzed the solutions of the SQL exercises in several exams of the course “Databases I” at the University of Halle The results for the final exam in winter term 2008/2009 are shown in Figure 1. The exercises are numbered with the

Error	3a	3b	3c	3d	3e	bonus	$\sum$
1	-	7	1	-	1	-	9
1a	-	-	-	-	6	9	15
2	2	3	3	3	2	3	16
3	-	-	-	-	1	-	1
5	-	-	-	-	3	-	3
6	-	8	1	-	11	13	33
8	-	-	8	4	3	-	15
11	-	-	-	2	-	-	2
13	-	-	-	1	-	-	1
15	-	-	2	-	-	-	2
17	-	-	-	-	23	-	23
19	-	-	1	6	1	-	8
21	-	-	-	-	9	-	9
22	-	-	-	2	-	-	2
25	-	-	1	-	-	-	1
26	-	-	-	-	10	18	28
27	1	-	2	-	3	4	10
28	-	1	-	-	-	-	1
36	-	-	1	-	-	-	1
37	14	-	4	-	1	-	19
39	-	-	-	1	-	1	2
Correct	27	41	29	29	3	-	40.6%
Only Semantic	13	10	9	18	15	20	26.7%
Syntax and Semantic	4	1	2	-	26	11	13.8%
Only Syntax	8	-	9	3	4	3	8.5%
Not Counted	-	-	1	-	3	17	6.6%
Wrong Task	1	1	3	3	2	2	3.8%

Figure 1: Error statistics for final exam, winter term 2008/2009

numbers and letters from section 3, Further course material and exam exercises are available from the project web page ([4]).

In 16.6 % of all solutions we did count several unrelated semantic errors in the same exercise, but in most cases they did not interact, thus almost all semantic errors (that did not occur simultaneously with syntactic errors) could have been found by our methods, described in [2]. Otherwise, if an error occurred more than once in the same exercise it was counted only once in our statistics. Error 27 (Missing join condition) almost always involves Error 37 (Many duplicates). Thus, in our statistics Error 37 is only counted if it occurs independent from Error 27.

The number of exams that contained at least one semantic error is the sum of the entries "Only semantics" and "Syntax and Semantic". Of course we counted only semantic errors from our list in [1], i.e. that are detectable without knowing the task of the query. "Wrong task" lists the number of exams that can only be detected as incorrect if the goal of the query is known. "Not counted" lists exams that did not try the particular exercise, or that contained so severe syntax errors that looking at semantic errors in detail was not possible. In this exam that we analyzed with this error taxonomy, the occurred semantic errors are (percentages are relative to all detected semantic errors):

1.	16.4 %	Error 6: Unnecessary join
2.	13.9 %	Error 26: Inefficient UNION
3.	11.4 %	Error 17: Unnecessary argument of COUNT
4.	9.5 %	Error 37: Many duplicates
5.	8.0 %	Error 2: Unnecessary DISTINCT
6.	7.5 %	Error 8: Implied, tautological or inconsistent subcondition
7.	7.5 %	Error 1a: Unnecessary outer query

References

[1] Stefan Brass and Christian Goldberg. Semantic Errors in SQL Queries: A Quite Complete List. In: *Elsevier's Journal of Systems and Software 79(5)*, 2006.

[2] Stefan Brass and Christian Goldberg. Proving the Safety of SQL Queries. In: *Fifth International Conference on Quality Software (QSIC'05)*, IEEE Computer Society Press, 2005.

[3] Stefan Brass and Christian Goldberg. Detecting Logical Errors in SQL Queries. In: *16th Workshop on Foundations of Databases (GvD'04)*, 2004.

[4] Stefan Brass and Christian Goldberg. SQLLint: Detecting Logical Errors in SQL Queries. Project website: http://dbs.informatik.uni-halle.de/sqllint/