Semantic Errors in SQL Queries: Exam Evaluation 2003-04

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 2003/2004 - final exam
Lecturer	:	Prof. Dr. Stefan Brass
University	:	Martin-Luther-University Halle, Germany
Analysis	:	Christian Goldberg
Date of Analysis	:	September 2006
Error Code Reference	:	[1]

2 Underlying Database Scheme

In the following exercises, we use a database scheme for a families budget which stores information about planned and actual expenditures:

```
PLAN(COST_CODE, BUDGET, FIX)
COST(ID, DAY, MONTH, COST_CODE→PLAN, AMOUNT, PURPOSE)
```

The column `FIX` may only contain '`Y`' or '`N`', depending whether the cost code has a fix budget or not. The table `COST` contains expenditures that does not have a fix budget only.

3 Analyzed Exercises and Possible Solutions

The final exam "Database Systems I" in winter term 2003/2004 contained 7 exercises about SQL, database modelling, data definition language, functional dependencies and normalization. The 3 analyzed SQL queries resulted in 9 out of 20 points. The 148 participating students had 60 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 may result in a deduction of points.

3.1 Exercise 1a)

Write a query that list the actual balance per cost code assuming that the actual month now is February. Thus, the sum of all cost has to be subtracted from the double budget for every cost code. Disregard all cost codes with a fix budget.

```
SELECT   P.COST_CODE, (P.BUDGET*2 - SUM(C.AMOUNT)) AS BALANCE
FROM     PLAN P, COST C
WHERE    P.COST_CODE=C.COST_CODE
AND      P.FIX='N' AND C.MONTH<=2
GROUP BY P.COST_CODE, P.BUDGET
```

3.2 Exercise 1b)

Now, list the balance for every cost code without a fix budget that has no cost so far. Again, assume that the actual month now is February. Thus, the balance has to be the double budget. Rank the output according to the balance, with the smallest balance first.

```
SELECT   P.COST_CODE, P.BUDGET*2 AS BALANCE
FROM     PLAN P
WHERE    P.FIX='N'
AND      NOT EXISTS (SELECT *
                     FROM   COST C
                     WHERE  C.COST_CODE=P.COST_CODE
                     AND    C.MONTH<=2)
ORDER BY P.BUDGET
```

3.3 Exercise 1c)

Request the handle and budget for the cost code with the biggest planned budget.

```
SELECT P.COST_CODE, P.BUDGET
FROM   PLAN P
WHERE  P.BUDGET=(SELECT MAX(P.BUDGET)
                 FROM   PLAN P)
```

4 Statistics

The list of error types mentioned in [1] is based on our experience from grading a large number of exams and homeworks. (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 2003/2004 are shown in Figure 1. The exercises are numbered

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

Error	1a	1b	1c	$\sum$
1	1	14	2	17
1a	4	-	-	4
2	2	2	-	4
3	-	1	-	1
5	11	9	1	21
6	2	5	2	9
7	-	-	1	1
8	4	1	1	6
11	1	-	3	4
14	2	2	1	5
17	1	-	-	1
19	1	2	16	19
20	1	-	-	1
21	1	1	1	3
22	1	1	-	2
25	3	-	1	4
27	2	5	-	7
28	-	6	-	6
30	1	3	-	4
Only Syntax	82	30	57	38.1%
Correct	17	45	54	26.1%
Only Semantic	9	31	18	13.1%
Syntax and Semantic	21	15	7	9.7%
Wrong Task	6	23	8	8.3%
Not Counted	13	4	4	4.7%

Figure 1: Error statistics for final exam, winter term 2003/2004

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.	17.7 %	Error 5: Unused tuple variable
2.	16.0 %	Error 19: `GROUP BY` with singleton groups
3.	14.2 %	Error 1: Inconsistent condition
4.	7.6 %	Error 6: Unnecessary join
5.	5.9 %	Error 27: Missing join condition
6.	5.0 %	Error 8: Implied, tautological or inconsistent subcondition
6.	5.0 %	Error 28: Uncorrelated `EXISTS`-subquery

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/