Project Report

Data Storage Paradigms, IV1351

Date

Project members:
[Elias Tosteberg, eliasto@ug.kth.se]

Declaration:

By submitting this assignment, it is hereby declared that all group members listed above have contributed to the solution. It is also declared that all project members fully understand all parts of the final solution and can explain it upon request.

It is furthermore declared that the solution below is a contribution by the project members only, and specifically that no part of the solution has been copied from any other source (except for lecture slides at the course IV1351), no part of the solution has been provided by someone not listed as a project member above, and no part of the solution has been generated by a system.

1 Introduction

This project is to create queries for use internally by Soundgood musicschool. There will be 4 queries that will be executed on a database. This project was made on my own based on Seminar 1 and 2. The seminar one part was made with a group but seminar 2 and 3 were made alone.

2 Literature Study

I have watched the SQL lecture. Where I learned the basics of writing queries. I learned how To SELECT what parts of tables that is needed. It also contains information on how to group things together. JOIN and UNION were very useful for my database. The tips and tricks document explained well how the subqueries and joins were supposed to be done. The Postgres documentation for syntax.

3 Method

When developing the queries I used Visual Studio Code to write the queries due to its automatic formatting and spellcheck. The queries were then pasted into a terminal window were the database ran. When making a queries the first step was creating a version that strips out all the unused data from the original table. When it runs correctly any missing data was then added by joining other tables. Last step is to convert the data into the required columns by using COUNT and similar. When verifying if a queries is working first we check for errors in the terminal. If that is correct we manually calculate the expected result from the database data and compare it to the result in the terminal.

4 Result

The queries are stored as queries.sql in https://github.com/Sailet03/IS1351-Seminar. The database is almost the same as in Seminar 2.

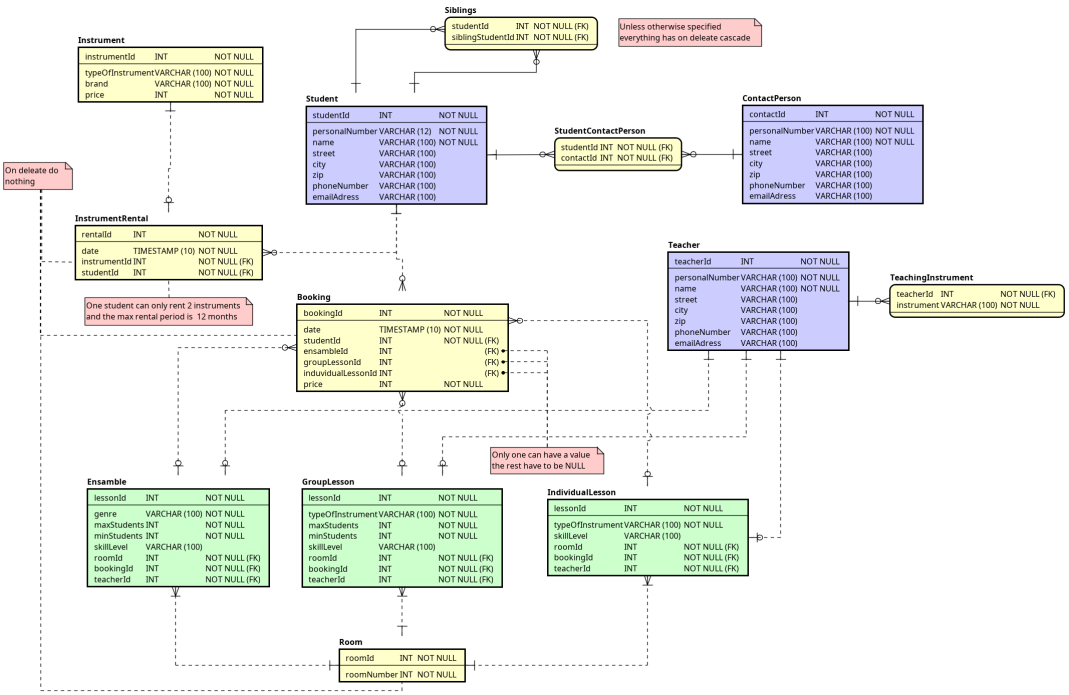

Figure 1: A sample diagram, included to illustrate caption (this text), numbering and reference in text.

The first query is the one that searches the database for how many lessons have been given during a particular year. The query works by selecting table booking and then counting how many of FK id are NOT NULL. The table booking has an id field for every single lesson type, but only one can have a value. So by counting the number of times

the FK are not null we can count the number of lessons used. The output is grouped by month and only entries in the specified year are considered.

Month	Total	Induvidual	Group	Ensamble
June	4	0	0	4
September	1	0	1	0
December	2	0	0	2

The second query is for counting how many students have how many siblings. It is based on multiple subqueries working together. First both the columns in the sibling table and the studentId of the student table are combined using UNION ALL. This will create a table that contains all the ids from all locations. The next subquery counts the number of times every id exits in this list and subtracts 1 from the final count. This will give an accurate representation of how many siblings every student has. The outermost subquery then formats the final answer.

No of Siblings	No of Students
0	3
1	2
2	3

The third query is for getting the number of lessons given by all teachers during a certain month. This query works by extracting all the lessonsId and date from every booking. It is then filtered by what month and year it is from. All remaining lessons are counted and displayed.

id	name	Total Lessons
1	Myles Gamble	4

The forth query is for listing the free seat of the Ensambles of a given week. This query works by joining the tables Booking and Ensemble together. The wanted information is then filtered by week and is then displayed.

Day	Genre	No of Free Seats
Thursday	Folkmusic	3
Sunday	Clasic	17

5 Discussion

The database is almost unchanged from seminar 2. The only one change was that the sibling table only had one FK from the student table. The other id was just an attribute. It is changed to a relation to self with two separate FK. This simplified the sibling query since half of the siblings weren't connected to the sibling table. This table should have been this way from the beginning. There are multiple subqueries but they don't contain Correlated Subqueries. They dont have any nested loops anywere. There are multiple UNION clauses in the table. They were necessary since multiple tables with mostly similar data had to be merged together. This is most evident on the 3 different lesson types. EXPLAIN ANALYZE on the forth query. It requires multiple rounds of sorting. Planing time is 0.285 ms while the execution time is 0.332 ms. This seems reasonably short. When the tables grow however the sorting might be more expensive and give a longer execution time.