

LAB 3

Lab Overview

A lot of the time when querying databases, you will need to handle large amounts of data. The client has collected a lot of data and now wants to see what this data can tell us about their current LMS. In this lab, you are provided with two files containing Create table and Insert statements. The provided database has some issues, but the client wants to query the data before the database is polished.

Rules

1. The labs must be done in groups of exactly two people. No larger groups are allowed, and if you have extraordinary extenuating circumstances that force you to do the labs alone, you must obtain permission to do so from the course leader. Both students in a group must be able to present all of the lab for the group to pass. Lab assistants do not record partial labs.
2. You must present correct and valid solutions to all the given tasks in order to pass the lab.
3. Presenting P+ assignments are optional for a higher grade if the given tasks are completed and passed.
4. This is a PSQL lab. No other programming languages, either embedded in the database or external to it, are allowed.
5. Please refrain from creating any [functions](#) since this lab is designed to assess query programming languages.
6. You are not allowed to hard-code anything except that which has been explicitly given to you in the problems. In particular, this means that constructs like limit 1 or similar artificial ways of reducing the output are forbidden.
7. You must utilize nothing but a single top-level SELECT statement to answer every problem(although you are, of course, allowed to use any number of sub-selects and joins required within that top level statement). You are specifically forbidden from referencing any temporary data structures like views or temporary tables from your solutions. Except on question 5 where you must use the recursive method.
8. No alterations of the database and data is allowed in this laboration.

LAB 3

Loading data into the database

OBS! The data from your previous labs **will not** be used for this or further labs. The reason for this is to formalize correct answers to the queries below, otherwise different databases would have different solutions, which could cause unfair correction.

Consider dropping these relations from the database if you have passed the previous lab, alternatively use your partner's database when setting up the new database.

1. For the lab environment, please read the "Postgre installation guide" available on Canvas.
2. Download the files LMS-schema.psql and LMS-inputs.psql from Canvas.
3. One file at a time, move the downloaded files to a folder in your home catalogue. Suggested method for doing this remotely is available under the section "Moving files local → KTH-remote" in the installation guide.
4. Run the `psql -h psql-dd1368-ht21.sys.kth.se` command and connect to the database
5. Run the `\i <filename>.psql` command with your moved files and remember to create the schema before the inputs.

Lab Presentation

Course related terminology is expected during the lab presentation and make sure you have the following ready to be presented to the TA:

- All queries asked for in the tasks, ready to be executed.
- Motivations for how the solution for the requested queries are sensible. (Tasks may have several ways to solve them, but they have to be good enough for e.g. a client to accept.)

LAB 3

Before you start

Make sure you're saving your queries. PostgreSQL uses the ".psql" fileformat. You can use your preferred text editing software or DataGrip by JetBrains which has database specific features. If you go with DataGrip you can use it for free with the student option.

Tip: Take your time and make sure you've read the questions carefully as some are quite dense.

Querying the Database

Queries for the following questions are to be completed and should show a satisfactory result. Lists and tables are expected to be alphabetical or numerical in order.

1. Display the title and combined genres for all books and sort the titles alphabetically.

Tips: Use the PSQL function [string_agg](#)

Expected output format:

title	genre
20000 Leagues Under the Sea	Northern
A Clash of Kings	Thriller
A Dance with Dragons	Northern
A Feast for Crows	Northern
A Game of Thrones	Horror
A Storm of Swords	Northern
Age of Discontinuity	Spooky
Age of Wrath	Western, Fantasy
Age of the Warrior	Sci-fi
Aghal Paghal	Southern, Thriller
Ahe Manohar Tari	Western, Thriller
All the Presidents Men	Western, Thriller
Amulet of Samarkand	Eastern, Course litterature
Analysis	Sci-fi, Romantic
Angels & Demons	Western, Thriller, Course litterature

2. Rank the top 5 of the all time most popular books in the 'RomCom' genre based on the amount of times they've been borrowed.

Note: The list may be longer than 5 if the books share rank

Tips: Use the PSQL function [rank](#), there is also an alternate version of this function called `dense_rank` with a slightly different format.

Expected output format:

LAB 3

title	rank
New Markets & Other Essays	1
India from Midnight to Millennium	2
Wealth of Nations	3

- Present a weekly report from week 1 to 30 on how many books were borrowed, returned and late.

Note:

- A week's late books are those books returned that week and which have $dor > doe$
- PSQL counts weeks a little differently, the year starts at week 53.

Tip: Use the PSQL function [date_part](#)

Expected output format:

week	borrowed	returned	late
1	14	5	2
2	11	7	3
3	20	9	3
4	12	6	2
5	14	4	4

- We want to see books that do and don't have prequels as a true/false list, however we only care about the books borrowed in the month of February. The first column says the book title, the second column tells us whether this book has a prequel or not as true or false and the third column shows when the book was borrowed. This should be sorted by titles alphabetically.

Expected output format:

title	every	dob
A Dance with Dragons	t	2021-02-12
A Dance with Dragons	t	2021-02-18
A Storm of Swords	t	2021-02-14
Age of the Warrior	f	2021-02-11
Ahe Manohar Tari	f	2021-02-11

LAB 3

5. Use the [recursive method](#) to present a list of the titles of each book in the Harry Potter Series.

***Note:** In this query you are allowed to directly reference an ID once in order to identify the Harry Potter series. Using an ID from a different series should identify that series instead. The last book in the Harry Potter series is Deathly Hallows.*

Expected output format:

title	bookid	prequelid
Harry Potter and the Deathly Hallows	8713	74532
Harry Potter and the Half-Blood Prince	74532	90293
Harry Potter and the Order of the Phoenix	90293	37822
Harry Potter and the Goblet of Fire	37822	3747
Harry Potter and the Prisoner of Azkaban	3747	51843
Harry Potter and the Chamber of Secrets	51843	76418
Harry Potter and the Philosophers Stone	76418	

(7 rows)

P+

For each book series, use the [recursive method](#) to present the name and sum of all pages of each book in all series in order.

***Note:** subsets of bookseries are not allowed. Meaning you may not have a list with LOTR1-LOTR2 and also LOTR1-LOTR2-LOTR3. You may only have each series once.*

Expected output format:

series	pages
Book1 -> Book2	333
Book1 -> Book2 -> Book3	999

(2 rows)