Lab 1: PostgreSQL, Data Independence, Keys

University of Toronto Mississauga

Due: Friday, January 28th, 2022 by 12PM ET

Released: Wednesday, January 19th, 2022 || Version: 1.1

Setup

1. Logging In

In this section, you need to complete the following steps:

1. Login to the remote department server (cslinux/cs.utm/lab pc) with your utorid. Using `scp` to upload the `lab1.ddl` and `lab1_drop.ddl` from your local PC that downloaded from Quercus Lab page. Then using `ls -la` to show that they have sit in the remote folder.
2. Login to PostgreSQL and use your database.

2. Running a script

In this section, you need to provide a screenshot that depicts each one of the following steps:

1. Import the `lab1.ddl` file to your database and show (by describing) the structure of all of the tables in your database.
2. Show all the tables in your database.
3. Import the `lab1_drop.ddl` to your database.
4. Show all tables in your database

Hint: there is no need to take four screenshots, one/two screenshots may suffice as they can show multiple steps.

Database Independence

Michael has decided to move the CSC343H5 student submission database from his local machine to a cloud service that is hosted at the University of Toronto. Before he transferred the data, Michael checked the tables in the database and noticed that there are some duplicate records (with different primary keys). In an attempt to fix the issue, Michael wrote a script to combine all of the duplicated records. The script was stored on his local drive. Michael also realized that the DBMS on the cloud service supports the LZMA algorithm which can compress the data. He applied the algorithm to his database first and then he transferred the data to the cloud service.

1. Identify the logical data independence from the description above and explain what affected it.
2. Identify the physical data independence from the description above and explain what affected it.

Keys

Keys and Referential Integrity

Consider the relations for a mid-size local supermarket database given by the following schema:

1. **Department (did, name)**: this table records the department name and ID.
2. **Employee (eid, name, gender, position, password, permission_level, sin, did)**: employee table stores the basic employee information such as the name, gender, position, sin, as well as the login password for the system and system permission.
3. **Providers (pid, pname, address, postcode, ein, swiftcode, branch, branch_address, contact_name, contact_num, notes)**: this records all item providers for the supermarket which contains the providers name, address info, ein, swiftcode and banking information for wire transfer money, provider contact information, and notes for the provider.
4. **Customer (cid, name, gender, total_spend, points)**: this table records the customer who registered a membership in the supermarket. Each customer has an unique cid, general personal information for the person, their total spend in the store, and the points that they earned.
5. **Product (productid, category, stock, price, name, ispromo, promotionid)**: it provides the product information and stock availability in the store. Also, the product may have promotion. If there is no promotion, then the promotionid will be null.
6. **Warehousing (date, productid, unit, price, saleprice, total_price, pid, eid)**: here is the table containing the warehousing details of items. It provides the date to do the warehousing, how many units there are, sale and import price, total amount for the warehousing, the provider of the item, and the employee who performed the actual warehousing operation. Each product can only be stocked once per day.
7. **Promotion (promotionid, promotion_type, description, discount, scriptcheck)**: here is all information about the promotion which includes the type of the promotion (e.g., discount, buy 1 get one free, etc.), description, discount if it is a discount promo, and scriptcheck to verify the validity of the promotion.
8. **Sale (sid, date, total_sale, sale_method, cid, sale_end_id, receipt_num, eid)**: records all sales that occurred in the store, including who made the sale and the sale machine identifier (i.e. sale_end_id).
9. **Receipt (receipt_num, productid, sid, quantity, total)**: here is receipt information for each unique item sold.

Tasks:

1. Identify all of the superkey(s) of each relation.
2. Identify the candidate key(s) of each relation. For each key, briefly state the assumptions or conditions under which each key would be valid.
3. Identify the foreign key in each relation (if applicable). For every foreign key, you must references the relation it is a primary key for.

Requirements and Submission

This lab is to be completed in partners (pairs of 2) unless written permission is given by the Course Coordinator. You and your partner are required to work together, equally contribute to, and understand all parts of your submission. Please refer to the syllabus for additional details on groups and the "Minimum Standards for Submitted Work".

All submission are required to be written in LaTeX. I would strongly suggest using Overleaf as a means of collaborative editing LaTeX documents. You are not permitted to use another word processor.

All files are to be submitted using the MarkUs platform (https://markus108.utm.utoronto.ca/csc343s22/). You or your partner must create the "group" and the other must accept the invitation to join. Once your group is formed on MarkUs, only one person from each group is required to submit the file(s). You may submit as many times as you like, in fact you are encouraged to do so! Please ensure your answers are typed and submissions is clearly legible/understandable.

You must include your, and your partner's, full name and student ID number in both in the PDF. Submit your answers in one file called **`lab1.pdf`** and the source LaTeX file in **`lab1.tex`**.