CS5200: Homework 1: Relational Algebra

Fall 2024

This assignment asks you to practice with writing and analyzing queries written in the variant of relational algebra discussed in class. For this assignment, please submit a single PDF containing your responses to the questions below. Label each of your answers with the part and question numbers. As a reminder, your submission must be your own work, completed individually, although you may discuss your approach to these problems, in general terms, with your classmates.

As your solutions may include the various mathematical symbols that we use to denote the relational algebra operations, you may submit either a typeset document (there are notes below on entering these symbols into your editor or word processor) or a scanned copy of a handwritten document. If you choose the latter, you must ensure that your writing is clear and legible to the grader.

Many of these questions are adapted from Database Systems: The Complete Book, 2nd edition, by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom.

1 Questions

1.1 Part 1

For this part, use the schemas and sample data from the "Product Database" section below.

There are 7 questions:

1. What is the output of the following relational algebra expression?

   $$\pi_{\mathsf{model}}(\sigma_{\mathsf{speed} \geq 3.0}(\mathsf{PC}))$$

   Include both a table containing the resulting rows and an English description of the data returned. Your English description must avoid a restatement of the relational algebra operators but should instead categorize the data in terms that would make sense to a user of the application, such as "the speed of every laptop whose price is less than 2000."

2. Write a relational algebra expression to compute the names of the manufacturers (makers) that sell PCs with at least 2048MB of memory. You may assume that all values in the ram column are in megabytes. Include a table containing the output of your query.

3. Write a relational algebra expression to find the model numbers of all black-and-white laser printers. Include a table containing the output of your query.

4. Write a relational algebra expression to list all manufacturers that sell printers but not laptops. Include a table containing the output of your query.

5. Write a relational algebra expression that lists the model numbers and prices of all PCs made by manufacturer B. Include a table containing the output of your query.

6. Extend your answer from the previous question to create a relational algebra expression that lists the model numbers and prices of all products, of any type, made by manufacturer B. Include a table containing the output of your query.

7. Write a relational algebra query that lists the price of the most expensive PC sold by each manufacturer. If a manufacturer does not sell any PCs, that manufacturer should not be included in the result. Include a table containing the output of your query.

8. What is the output of the following relational algebra expression?

$$\pi_{\mathsf{maker}}(\mathsf{Product} \ltimes \sigma_{\mathsf{price}<750}(\mathsf{PC}))$$

   Include both a table and a one-sentence English description of the data returned.

1.2 Part 2

There are 4 questions in this part.

1. What is the schema of the result of the following relational algebra expression?

$$\sigma_{\mathsf{numPostsCommented}\geq 3}({}_{\mathsf{userID,firstName,lastName}}\gamma_{\mathsf{numPostsCommented}\leftarrow\mathsf{count(distinct\ postID)}}(\mathsf{User} \bowtie \mathsf{Comment}))$$

   Be sure to include both attribute names and their types.

2. Is the following relational algebra expression well-typed? If so, what is the result's schema? If not, identify the type error.

$$\sigma_{\mathsf{year(posted)}=2022}({}_{\mathsf{userID,firstName,lastName}}\gamma_{\mathsf{numPostsCommented}\leftarrow\mathsf{count(postID)}}(\mathsf{User} \bowtie \mathsf{Comment}))$$

3. Consider the following two queries that are intended to list all of the users who have posted fewer than 5 times since March 1, 2023. One of the two queries is well-typed, and the other is not. Identify which of the two queries is not well-typed and explain why.

   (a) $\mathsf{User} \ltimes \sigma_{\mathsf{numPosts}<5}({}_{\mathsf{userID}}\gamma_{\mathsf{numPosts}\leftarrow\mathsf{count(postID)}}(\sigma_{\mathsf{posted}\geq\texttt{'2023-03-01'}}(\mathsf{Posts})))$

   (b) $\sigma_{\mathsf{numPosts}<5}(\mathsf{User} \ltimes {}_{\mathsf{userID}}\gamma_{\mathsf{numPosts}\leftarrow\mathsf{count(postID)}}(\sigma_{\mathsf{posted}\geq\texttt{'2023-03-01'}}(\mathsf{Posts})))$

4. Write a relational algebra expression that produces a table with two columns, userID (a string) and numPosts (an integer). For each user in the system, numPosts should be the number of posts that user has written. If a user has not written any posts, that user should appear in the output with a numPosts value of 0.

2 Note on Entering Mathematical Symbols

Writing relational algebra expressions requires you to enter various mathematical symbols into your document. The steps for doing this vary from program to program and operating system to operating system, and I cannot provide instructions for all possible combinations.

To insert these characters when using macOS, you can press command-control-space to bring up the Character Viewer window; many of the relevant symbols can be found under Math Symbols. You can also use the search box in the upper left to search for specific symbols. Helpful search terms include "sigma," "pi," "bowtie," and others.

If you have questions or suggestions about the best way to insert these characters on a different operating system, I encourage you to post to Piazza.

3 Product Database

This database tracks information about computer products available for purchase. It consists of four relations, with the following schemas:

- Product(maker : string, <u>model</u> : integer)
- PC(<u>model</u> : integer, speed : float, ram : integer, hd : integer, price : integer)
- Laptop(<u>model</u> : integer, speed : float, ram : integer, hd : integer, screen : float, price : integer)
- Printer(<u>model</u> : integer, color : string, type : string, price : integer)

Constraints:

- PC.model, Laptop.model, and Printer.model are each foreign keys referring to Product.model. It therefore follows that each record in PC, Laptop, and Printer must have a matching record in Product.
- For every tuple in the Product relation, there exists a corresponding tuple (with matching foreign key value) in *exactly one* of the PC, Laptop, and Printer relations.

Table 1: Product

maker	**model**
A	1001
A	1002
A	1003
A	2004
A	2005
A	2006
A	3001
B	1004
B	1005
B	1006
B	2007
B	3002
B	3003
C	1007
D	1008
D	1009
D	1010
D	3004
D	3005
E	2001
E	2002
E	2003

Table 2: PC

model	**speed**	**ram**	**hd**	**price**
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649
1005	3.20	4096	250	1495
1006	3.20	1024	320	1049
1007	2.20	1024	200	510
1008	2.20	2048	250	770
1009	2.00	1024	250	650
1010	2.80	2048	300	770

Table 3: Laptop

model	speed	ram	hd	screen	price
2001	2.00	2048	240	20.1	3673
2002	1.73	1024	80	17.0	949
2003	1.80	512	60	15.4	549
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500
2006	2.00	2048	80	15.4	1700
2007	1.60	1024	100	15.4	1429

Table 4: Printer

model	color	type	price
3001	true	ink-jet	99
3002	false	laser	239
3003	true	laser	899
3004	true	ink-jet	120
3005	false	laser	120

4 Blog Application

This database contains all of the data required for a blogging website. We provide only schemas here, as the questions do not depend on actual data.

- User(userID : string, firstName : string, lastName : string, accountCreated : date)
- Post(postID : integer, userID : string, title : string, body : string, posted : date)
- Comment(commentID : integer, userID : string, postID : integer, posted : date, body : string)
- Reshare(userID : string, postID : integer, created : date)
- Vote(userID : string, postID : integer)

Foreign key relationships:

- Post.userID → User.userID
- Comment.userID → User.userID
- Comment.postID → Post.postID
- Reshare.userID → User.userID
- Reshare.postID → Post.postID
- Vote.userID → User.userID
- Vote.postID → Post.postID

These relationships indicate requirements between relations. By "Comment.author → User.email", for example, we mean that for every tuple in the Comment relation, there must exist a tuple in the User relation such that the value of the author attribute in the Comment tuple equals the value of the email attribute in the User relation.