

Introduction to Databases

Relational Algebra

under set semantics

Dr Paolo Guagliardo

(v26.0)

CHANGELOG

v26.0 Initial version.

Problem 1. Given the database below:

CUSTOMER			ACCOUNT			
ID	Name	City	Number	Branch	CustID	Balance
1	John	London	111	London	1	120
2	Mary	Edinburgh	222	Edinburgh	1	62
3	Jeff	London	333	London	3	76
4	Jane	Cardiff	444	London	2	200

compute the answer to the following query (without rewriting it):

$$\text{CUSTOMER} \bowtie (\pi_{\text{ID}, \text{City}}(\text{CUSTOMER}) \cap \rho_{\text{CustID} \rightarrow \text{ID}, \text{Branch} \rightarrow \text{City}}(\pi_{\text{Branch}, \text{CustID}}(\text{ACCOUNT})))$$

To this end, first work out all intermediate results on paper and by hand; then verify them, and the final answer, using the Relational Algebra interpreter [real](https://git.ecdf.ed.ac.uk/pguaglia/real)¹ (see [usage instructions](#)) by appropriately rewriting the natural join ($\bowtie$) operation.

Problem 2. Using the definition of natural join and intersection, rewrite the query of Problem 1 into an equivalent expression that uses only primitive operations.

Problem 3. Write the following queries in relational algebra:

- (1) “ID and name of customers who own an account in a branch in their city.”
- (2) “ID and name of customers who do **not** own any account.”
- (3) “ID and name of customers who own an account in **every** branch.”
- (4) “ID and name of customers who own an account with a balance which is no less than the balance of any other account.”

Problem 4. Can query (4) of Problem 3 ever return more than one tuple? If yes, show a database (over the given schema) on which that happens; otherwise, explain why it cannot happen.

¹<https://git.ecdf.ed.ac.uk/pguaglia/real>