

Supplementary Questions and Exercises

Nested Relations

1. The following *nested relation* has two tuples, each containing two relations, X and Y .

R				
$(A$	X	Y		$)$
	$(J$	$K)$	$(J$	$L)$
1	a	1	a	9
	a	2	a	7
	b	1		
2	a	1	b	8
	a	2	b	9
	b	1		

- (a) Using the same way of writing a nested relation, show the data that would result from the Aldat code:
- ```
let Z be X ijoin Y;
S <- [A, Z] in R;
```
- (b) Describe (in words with examples if needed) two *different* ways of building an implementation for the above Aldat code. At least one of these ways should take into account the facts that a) joins are expensive, and b) the relation  $R$  might have thousands or millions of tuples.
2. In a relation,  $Names(SIN, family, given)$ , a person (identified by Social Insurance Number) has one family name and possibly many given names. Rewrite this relation as a nested relation and write the code to find all people who have all the given names of anyone else in their families. (Assume same family name means same family.)
3. In the nested relation  $agri$ , shown below, the second  $y$ -coordinate in the corn polygon is wrong and must be updated from 4 to 3. Write the update statement to do this.

| $agri$ |       |         |     |      |     |
|--------|-------|---------|-----|------|-----|
| $(p$   | $c$   | $shape$ |     |      | $)$ |
|        |       | $(seq$  | $x$ | $y)$ |     |
| 1      | corn  | 1       | 1   | 3    |     |
|        |       | 2       | 3   | 4    |     |
|        |       | 3       | 2   | 4    |     |
| 2      | wheat | 1       | 3   | 3    |     |
|        |       | 2       | 1   | 3    |     |
|        |       | 3       | 2   | 1    |     |

4. In the nested relation  $agri$ , shown below, the second  $y$ -coordinate in the corn polygon is wrong and must be updated from 4 to 3. Write the update statement to do this.

|             |          |              |          |            |
|-------------|----------|--------------|----------|------------|
| <i>agri</i> |          |              |          |            |
| ( <i>p</i>  | <i>c</i> | <i>shape</i> |          | )          |
|             |          | ( <i>seq</i> | <i>x</i> | <i>y</i> ) |
| 1           | corn     | 1            | 1        | 3          |
|             |          | 2            | 3        | 4          |
|             |          | 3            | 2        | 4          |
| 2           | wheat    | 1            | 3        | 3          |
|             |          | 2            | 1        | 3          |
|             |          | 3            | 2        | 1          |

5. A retail chain is interested in what items customers are likely to buy together, so it can place them on the shelves so as to maximize the time the customer spends in the stores looking for them. So it collects data at the checkout points for the relation  $sales(item, cart)$ . (For instance, *cart* 1 contains **bread**, **butter**, and **coffee**, *cart* 2 contains **bread**, **butter**, and **milk**, and so on.)

Write the Aldat code to create a nested version of this data,  $cartsets(item, carts(cart))$ , associating sets of carts with each item. Show what this does to the above data.

6. Continue the previous question to convert *item* to the singleton nested relation  $items(item)$ . Now write code to create a new relation,  $salesets2(items'', carts'')$ , which gives pairs of *items* and the sets of *carts* where both items in the pair are in all carts in the set. Eliminate tuples with empty sets of *carts*.

Thus, in the above, the set {1,2} of *carts* is associated with the set {**bread**, **butter**} of *items*. Show all the results for the data given above.

For the relation,  $Checkout(item, cart)$ , shown in matrix form, write relational algebra (and domain algebra if needed) to find:

7. —all items that are in all carts (do this in three completely different ways); and  
 —all carts with an odd number of items.

|                           |   |   |   |
|---------------------------|---|---|---|
| <i>item</i> \ <i>cart</i> | 1 | 2 | 3 |
| bread                     | • | • | • |
| butter                    | • | • |   |
| jam                       | • | • | • |
| milk                      |   | • | • |

8. Write the view that represents the relation  $Checkout$  of question 7 as two tuples and a Cartesian product. Write also the statements that define the three components of the view.

Write the statements that convert the flat relation of question 7 to the nested relation shown.

9. (You may use constructs defined but not yet implemented in the language.)

|                  |                 |
|------------------|-----------------|
| <i>Checksets</i> |                 |
| ( <i>items</i>   | <i>carts</i> )  |
| ( <i>item</i> )  | ( <i>cart</i> ) |
| {bread}          | {1,2,3}         |
| {butter}         | {1,2}           |
| {jam}            | {1,2,3}         |
| {milk}           | {2,3}           |

6. Write code to flatten any nested relation on the same attributes as  $R$ , shown, to  $Rflat$ . Show  $Rflat$  for the example data. Write code to nest  $Rflat$  back to the same attributes as  $R$  and show the result for the example data.

|       |         |
|-------|---------|
| $R(A$ | $G(B))$ |
| 1     | 1       |
|       | 2       |
| 1     | 1       |
| 2     | 1       |

components of this work owned by others than T. H. Merrett must be honoured. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to republish from: T. H. Merrett, School of Computer Science, McGill University, fax 514 398 3883.

The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.