# CS 410- QL Database App Dev Assignment: Relational Algebra - Bag semantics, complex expressions, and extended Algebra

I, Baba Avinash Puppala (bpuppala), declare that I have completed this assignment completely and entirely on my own, without any consultation with others. I understand that any breach of the UAB Academic Honor Code may result in severe penalties.

## Solution to Question 1: Bag Projection and Intersection

Given relation $R(A, B, C)$ with the following tuples:

| $A$ | $B$ | $C$ |
|---|---|---|
| 1 | 2 | 3 |
| 1 | 2 | 3 |
| 2 | 3 | 1 |
| 3 | 1 | 2 |
| 2 | 2 | 3 |
| 2 | 3 | 3 |

We need to compute the intersection:

$$\text{PROJ}_{A,B}(R) \cap \text{RENAME}_{S(A,B)}(\text{PROJ}_{B,C}(R)),$$

where $S$ is used to rename the attributes of $\text{PROJ}_{B,C}(R)$ to match the schema of $A, B$.

### Step 1: Projection

- The projection $\text{PROJ}_{A,B}(R)$ results in:

$$(1,2),(1,2),(2,3),(3,1),(2,2),(2,3).$$

Since we are considering a bag (multiset), duplicates are preserved.

- The projection $\text{PROJ}_{B,C}(R)$ results in:

$$(2,3),(2,3),(3,1),(1,2),(2,3),(3,3).$$

### Step 2: Renaming

Rename $\text{PROJ}_{B,C}(R)$ to have the same schema as $A, B$:

$$(2,3),(2,3),(3,1),(1,2),(2,3),(3,3).$$

### Step 3: Intersection

Find the intersection between $\text{PROJ}_{A,B}(R)$ and the renamed $\text{PROJ}_{B,C}(R)$:

$$(1,2),(2,3),(3,1).$$

The result contains only the tuples that appear in both projections.

## Solution to Question 2: Set and Bag Operations

Let $R(A,B,C)$ and $S(A,B,C)$ be two relations where no attributes can have a null value. Determine whether the given equations hold for both sets and bags.

### Part (a): $(R \cap S) = (RS)$

- **Set Operations:** In general, $R \cap S$ returns the common tuples in both $R$ and $S$, while $RS$ produces a join result based on matching tuples across all attributes. The join is equivalent to $R \cap S$ only if $R$ and $S$ contain exactly the same tuples with matching attributes. - **Bag Operations:** Even when considering bags, a join operation can produce multiple occurrences based on matching criteria, whereas $R \cap S$ counts occurrences in both relations. Hence, they are not guaranteed to be equal.

### Part (b): $((R - S) \cup (S - R)) \cap (R \cap S) = \emptyset$

- **Set Operations:** This equation is always true because $(R - S) \cup (S - R)$ represents the symmetric difference, which includes elements not common to both sets. Since $R \cap S$ contains elements common to both, their intersection will indeed be an empty set. - **Bag Operations:** For bags, even if there are duplicate elements, the symmetric difference and intersection will remain disjoint. Therefore, the equation still holds true.

### Part (c): $\Delta(R) = \Gamma_{A,B,C}(R)$

- **Set Operations:** The operator $\Delta$ returns the distinct tuples from $R$. When applying the grouping operator $\Gamma$, the resulting grouped tuples are the same as the distinct elements in $R$, thus making $\Delta(R)$ and $\Gamma_{A,B,C}(R)$ equivalent. - **Bag Operations:** In the case of bags, both operations will return the distinct tuples without considering duplicates, hence they are equal.

### Part (d): $R \cup S = R$

- **Set Operations:** This equation holds if and only if $S \subseteq R$. In other words, all tuples in $S$ must also be present in $R$. - **Bag Operations:** For bags, each tuple in $S$ must appear in $R$ at least as many times as in $S$ for this equation to hold.

## Solution to Question 3: Database Queries Using Relational Algebra

Given the following relations:

- **Person** (*ID*, *name*, *age*, *gender*) where *ID* is a key.
- **Shop_list** (*ID*, *item*) where (*ID*, *item*) is the key.
- **Sell** (*supermarket*, *item*, *price*) where (*supermarket*, *item*) is the key.

### Part (a): Find all the items on the shop_list of at least one person over the age of 18

$$\text{PROJ}_{item}(\sigma_{age>18}(\text{Person}) \bowtie \text{Shop\_list})$$

This expression filters the persons who are over 18, joins the result with the shop_list, and projects the items.

## Part (b): Find the names of all males who have at least one item in their shop_list sold at Walmart

$$\text{PROJ}_{name}(\sigma_{gender='M'}(\text{Person}) \bowtie \text{Shop\_list} \bowtie \sigma_{supermarket='Walmart'}(\text{Sell}))$$

The query first filters for male persons, joins with their shopping lists, then joins with items sold at Walmart, and finally projects their names.

## Part (c): Find for each person, the lowest price of items on his/her shop_list

$$\Gamma_{ID,\text{MIN}(price)}(\text{Person} \bowtie \text{Shop\_list} \bowtie \text{Sell})$$

The expression performs a join across the three relations and then uses a grouping to find the minimum price for each person based on their ID.