## Chapter 5.2 Algebra Law used for Query Project Improvement

This chapter will show some Algebra Laws and these laws are used to convert one Expression Tree to another equal Expression Tree, and the latter may has the more effective Physics Query Plan.

*The result of applying these Algebra Expressions is the Logical Query Plan, it is the output of Query Rewrite Phase.*

### Chapter 5.2.1 Commutation Law and Association Law

***Definition:***

1. Commutation Law: The results are always the same even there have some sequential changes of the parameters.
2. Association Law: The calculation can start from the left, also it can start from the right.

***Laws:***

Multi - Operators of Relation Algebra satisfy the Commutation Law and Association Law.

* R \* S = S \* R; (R \* S) \* T = R \* (S \* T)
* R join S = S join R; (R join S) join T = R join (S join T)
* R union S = S union R; (R union S) union T = R union (S union T)
* R intersection S = S intersection R; (R intersection S) intersection T = R intersection (S intersection T)

*(Attention: These laws are established for Set and Package.)*

***Example:***

Verify the Commutation Law: R join S = S join R:

* Assume that the tuple t exists in the result of R join S, which is to say tuple t exists in the left expression. Then tuple r must exists in Relation R, and the tuple s exists in Relation S, they must be have the same value on the common property t. Therefore when we calculate the right expression S join R, then tuple s and r will combine as the tuple t.
* Because our Relation Algebra is a package, but not set, so we must verify that if tuple t appears in the left for n times, then t should also appears in the right for n times.
* Assume that tuple t appears in the left for n times ,then tuple r in Relation R must appears for nr times, while tuple s in Relation S must appears for ns times, nr \* ns = n.
* When we calculate the right expression S join R, tupe s should appear ns times, tuple r should appear nr times, then we can get nr\*ns times t copies, then n tuple t.

***Supplement:***

Theta Join is changeable. R join c S = S join c R, as long as the condition is meaningful, then Theta Join also satisfies the Association Law.

***Example:***

Assume that we have three Relation R(a, b), S(b, c), T(c, d), the expression:

[ R join (R.b > S.b) S ] join (a < d) T

Here we can not calculate Relation S join T first, since attribute a and d do not belong to Relation S and Relation T. So when we use the Theta Join, we need to pay attention to it.

### Chapter 5.2.2 Law Selection

### Chapter 5.2.3 Push Down Selection

### Chapter 5.2.4 Projection Law

### Chapter 5.2.5 Join and Product Law

### Chapter 5.2.6 Eliminate Duplication Law

### Chapter 5.2.7 Grouping and Aggregation Law