## Chapter 5.4 The Estimation of Calculation Cost

***Definition:***

After we analyze the Grammar of one Query and convert to one Logic Query Plan, the next step is that we must convert the Logic Query Plan to the Physical Plan.

Normally, when we check multiple Physical Plan and evaluate each Physical Plan, or just evaluate the cost of this conversion. After through this kind of evaluation, normally is called *Enumeration based on Cost*, however we choose the Physical Query Plan with the least estimation cost.

For each Physical Plan, we choose:

* For each Law that satisfied the *Combination Law* and *Distribution Law*, for example decide the sequence of Join, Union, Intersection and Grouping Operator.
* For Algorithm of each Operator in the Logical Plan, for example, decide whether to use *Nested Loop Join* or *Hash Join*.
* For other Operators, such as Scan, Sort and so on, they are required in the Physical Plan but *not visible* in Logical Plan.
* The method that pass one parameter to the next parameter, such as, through using *the intermediate result* or through using *iterator*.Then pass one tuple of one parameter each time or one main memory buffer area.

In order to make each selection, we need to know the cost of each Physical Plan. Also if there has no execution plan, we can not know it’s cost. However, executing one Query Project is much more than select a Project. So, in order to not execute multi projects for one Query, this forced us to estimate the cost of this plan.

Of course, before we discuss the Physical Plan Enumeration, we must consider how could we estimate the cost of these Plan.

### Chapter 5.4.1 The Estimation of Intermediate Relation Size

***Definition:***

Here we choose from the Physical Plan is to minimize the estimation cost when executing Query. Here, no matter we use which method to execute the Query Plan, and no matter how the cost of Query Plan is estimate, then the size of intermediate Relation has big influence on the cost.

Normally, we can use the rule to estimate the tuple numbers in the Intermediate Relation, make these rule:

1. Give the Exact Estimation.
2. Easy to Calculation.
3. Logical Consistent, find out one way which is to say the estimation of one intermediate Relation Size do not depend on this Relation. *(For example, the estimation size of multi Relation Join should not depend on the sequence when we calculate.)*

***Supplement:***

We need to mention that, there do not exist one method which can satisfy all these three conditions. Here we give one simple rule that satisfies most situation.

*Luckily, here we just want to help find one Physical Query Plan but not estimate the accurate size.* Although this inaccurate estimation do has some error with the actual estimation, but as long as it help decide the final best Physical Query Plan, then we can let this tiny error go.

### Chapter 5.4.2 The Estimation of Projection Operator

***Instruction:***

Here we makes the traditional De - Duplication Projection as the Package Projection followed with the De - Duplication Operator. Normally, the Projection Operator does not the same as other Operators, since the size of result can be calculated precisely. If the size of Projection decreased, then just because of some of the fields are eliminated. However, enlarge Projection Operator will cause generate the new fields, and they are the combinations that had before, under this circumstance, Projection Operator actually adds the size of Relation.

***Example1:***

Assume that Relation R(a, b, c) is one relation, among which a, b are the integer that occupies 4 bytes length, c is the string with 100 bytes length. Also assume that the tuple head needs 12 bytes. Then each tuple of Relation R will occupies 12 + 4 + 4 + 100 = 120 bytes.

Assume that the block is 1024 bytes length, and among this the block head occupies 24 bytes. ( 1024 - 24 ) / 120 = 8, then each block can be used to store 8 tuples. By the way, assume that T(R) = 10 000, which is to say, there has 10 000 tuples in Relation R, then B(R) = 10 000 / 8 = 1250.

***Example2:***

Consider the S = Projection a + b -> x, c (R), then we use the sum of a and b to substitute the a and b. The tuple of Relation S occupies 4 + 100 + 12 = 116 bytes. Although the total bytes of tuple of Relation S is much smaller, but we can still store 8 tuples in one block. Therefore, T(S) = 10 000 and B(S) = 1250.

***Example3:***

Now let’s consider U = Projection a, b (R), then in this Relation, the tuple occupies 4 + 4 + 12 = 20 bytes. Also T(U) = 10 000, then (1024 - 24 ) / 20 = 50 can be stored in one block. B(U) = 10 000 / 50 = 200.

### Chapter 5.4.3 The Estimation of Selection Operator

### Chapter 5.4.4 The Estimation of Join Operator

### Chapter 5.4.5 Natural Join of Multi - Join - Property

### Chapter 5.4.6 Join of Multi - Relation

Chapter 5.4.7 The Estimation of Other Operators