Relational algebra operators – combining operations

Relational algebra is the mathematical basis for performing queries against a relational database. Operations are performed against relations – resulting in relations. Because the result of relational algebra operation is a relation, operations can be stacked up against each other. This becomes the basis for multiple selection criteria.

For the following examples, we are going to be using relations Employees & Parking with the following schemas:

Employee (Emp_id:int, Emp_name:string, Emp_office:int)
Parking (Emp_id:int, Parking_lot:string, Parking_space:int)

Combining

Imagine that we want the list of Employee ids that are in office 10. Let's look at the following relation instances of Employee & Parking:

Employee		
Emp_id	Emp_name	Emp_office
1001	Bob	10
1002	Alice	11
1003	Sandy	10
1004	Larry	11
1005	Susan	11

Parking		
Emp_id	Parking_lot	Parking_space
1001	A	6
1002	A	14
1003	B	17
1004	B	6
1005	A	12

If we apply the selection operation to this instance of Employee, we are saying we want a specific set of rows out of this instance. The statement:

$\sigma_{Emp\_offce=10}$ Employee

is saying, select all of the rows whose Emp_office is equal to 10 from this instance of the relation Employee. The result of this operation is a relation instance of Employee that looks like this:

Employee		
Emp_id	Emp_name	Emp_office
1001	Bob	10
1003	Sandy	10

We can then apply the projection operation to the result relation as follows:

$\Pi_{Emp\_id}(\sigma_{Emp\_offce=10}$ Employee)

Which says to project out the attribute Emp_id from the result relation of the sigma operation. This returns the following relation:

Emp_id
1001
1003

Now - what if the question you were asking was: What parking is used by the employees in office 10? With the above, we have identified the employee ids of the employees in office 10, if we perform a natural join with the Parking relation now, we will get the associated parking spaces. The statement:

$(\Pi_{Emp\_id}(\sigma_{Emp\_offce=10}$ Employee)) |X| Parking

Will give us the following relation:

Emp_id	Parking_lot	Parking_space
1001	A	6
1003	B	17

Clearly precedence is important here, liberal use of parenthesis will help you keep straight what is happening.

One last twist, what if I wanted the names of the employees in office 10 and their parking spots? The selection of employees (the inner query) doesn't change, but we need to project out both the Emp_id and the name if we want it in the result:

$(\Pi_{Emp\_id,\ Emp\_name}(\sigma_{Emp\_offce=10}$ Employee)) |X| Parking

The result of this is:

Emp_id	Emp_name	Parking_lot	Parking_space
1001	Bob	A	6
1003	Sandy	B	17

And again, if we then DIDN'T want the emp_id showing in the result, we project out the columns we do want.

$\Pi_{Emp\_name,\ Parking\_lot,\ Parking\_space}(\Pi_{Emp\_id,\ Emp\_name}(\ \sigma_{Emp\_offce=10}$ Employee)) |X| Parking

To give us:

Emp_name	Parking_lot	Parking_space
Bob	A	6
Sandy	B	17