Nested Query Processing

Kyuseok Shim
Seoul National Univ.

Optimization of Nested Queries

- Sample Schema

  Supplier (sno, sname, sloc, sbudget)
  Part (pno, pname, dim, price, color)
  Project (jno, jname, pno, jbudget, jloc)
  Shipment (sno, pno, jno, qty)

- Outer query block and inner query block (Q)

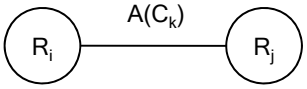
```
select   sno
from     shipment
where    pno = ( select  max(pno)
                 from    part
                 where price > 25 )
```

2/10

Optimization of Nested Queries

- Type-A nesting if
  - the inner query block Q does not contain a join predicate that references the relation in outer query block
  - select clause of Q indicates an aggregate function associated with the column name

```
select    sno
from      shipment
where     pno = ( select  max (pno)
                  from    part
                  where price > 25 )
```



3/10

Optimization of Nested Queries

- Type-A nesting if
  - the inner query block Q does not contain a join predicate that references the relation in outer query block
  - select clause of Q indicates an aggregate function associated with the column name

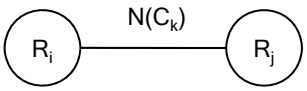
```
select    sno
from      shipment
where     pno = ( select  max (pno)
                  from      part
                  where     price > 25 )
select    sno
from      shipment,
          ( select  max (pno)
            from    part
            where price > 25 ) as M(pno)
where      shipment.pno = M.pno
```

4/10

Optimization of Nested Queries

- Type-N nesting if
  - Q does not contain a join predicate that references the relation of the outer query block
  - the column name in the select clause of Q does not have an aggregate function associated with it

```
select    sno
from      shipment
where     pno in ( select    pno
                   from      part
                   where     price > 25 )
```



5/10

Optimization of Nested Queries

- Type-N nesting if
  - Q does not contain a join predicate that references the relation of the outer query block
  - the column name in the select clause of Q does not have an aggregate function associated with it

```
select    sno
from      shipment
where     pno in ( select  pno
                   from    part
                   where   price > 25 )
select    sno
from      shipment,
          ( select pno
            from   part
            where price > 25 ) as M(pno)
where     shipment.pno = M.pno
```

6/10

Optimization of Nested Queries

- Type-N nesting if
  - Q does not contain a join predicate that references the relation of the outer query block
  - the column name in the select clause of Q does not have an aggregate function associated with it

```
select    sno
from      shipment
where     pno in ( select  pno
                   from part
                   where       price > 25 )
```

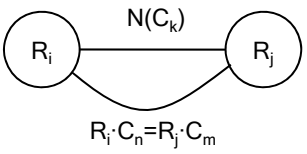
```
select    sno
from      shipment, part
where     shipment.pno = part.pno
   and    price > 25
```

7/10

Optimization of Nested Queries

- Type-J nesting if
  - The where clause of Q contains a join predicate that references the relation of the outer query block
  - the column name in the select clause of Q is not associated with aggregate function

```
select    sno
from      shipment
where     pno in ( select  pno
                   from     project
                   where    shipment.jno = project.jno
                      and jloc = 'New York' )
```



8/10

Optimization of Nested Queries

- Type-J nesting if
  - The where clause of Q contains a join predicate that references the relation of the outer query block
  - the column name in the select clause of Q is not associated with aggregate function

```
select    sno
from      shipment
where     pno in ( select  pno
                   from    project
                   where   shipment.jno = project.jno
                     and jloc = 'New York' )
```

```
select    sno
from      shipment, project
where     shipment.pno = project.pno
  and     shipment.jno = project.jno
  and     project.jloc = 'New York'
```

Need to remove duplicate of pno from part!

9/10

Optimization of Nested Queries

- Type-JA nesting if
  - The query graph exhibits at least one circular arc and at least one straight arc labeled 'A'

```
select    sno
from      shipment
where     pno = ( select   max (pno)
                  from     project
                  where    shipment.jno = project.jno
                    and    jloc = 'New York' )
```

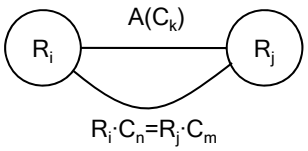
```
select    sno
from      shipment
where     pno = ( select   M.mpno
                  from     ( select          jno, max (pno)
                             from            project
                             group by        jno ) as M (jno, mpno)
                   where    shipment.jno = M.jno
                     and    jloc = 'New York' )
```

10/10

Optimization of Nested Queries

- Type-JA nesting if
  - The query graph exhibits at least one circular arc and at least one straight arc labeled 'A'

```
select    sno
from      shipment
where     pno = ( select          max (pno)
                  from    project
                  where shipment.jno = project.jno
                    and           jloc = 'New York' )
```



11/10

Optimization of Nested Queries

- Type-JA nesting if
  - The query graph exhibits at least one circular arc and at least one straight arc labeled 'A'

```
select    sno
from      shipment
where     pno = ( select    max (pno)
                  from      project
                  where     shipment.jno = project.jno
                    and     jloc = 'New York' )
select    sno
from      shipment,
          ( select          jno, max (pno)
            from            project
            where           jloc = 'New York'
            group by        jno ) as M (jno, mpno)
where     shipment.jno = M.jno
  and     shipment.pno = M.mpno
```

12/10

Optimization of Nested Queries

- Count Bug for Type-JA nesting

```
select   R.a
from     R
where    R.b OP ( select        count (S.*)
                  from   S
                  where R.c = S.c )
select   R.a
from     R,
         ( select       S.c, count (S.*)
           from         S
           group by     S.c ) as T (c, count)
where    R.c = T.c
  and    R.b OP T.count
```

Problem when count is zero!

13/10

Optimization of Nested Queries

- To resolve the Count Bug for Type-JA nesting

```
select   R.a
from     R
where    R.b OP ( select        count (S.*)
                  from   S
                  where R.c = S.c )
select   R.a
from     R, S
where    R.c = S.c          // Outer Join
group by R.#                // primary key
having   R.b OP count (S.*)
```

Do we really want to implement outer join?

14/10

Optimization of Nested Queries

- Type-D nesting if
  - The join predicate in either Q_i or Q_j (or both) references the relation of the outer query block

```
select   sname
from     supplier
where    ( select   pno
           from     shipment
           where    shipment.sno = supplier.sno)
                 contains
         ( select   pno
           from     part
           where    color = 'red' and price > 25)
```

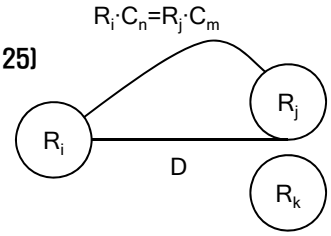
Too complex!

15/10

Optimization of Nested Queries

- Type-D nesting if
  - The join predicate in either Q_i or Q_j (or both) references the relation of the outer query block

```
select   sname
from     supplier
where    ( select pno
           from   shipment
           where shipment.sno = supplier.sno)
                  contains
         ( select pno
           from   part
           where color = 'red' and price > 25)
```



16/10

When we only allow temporary relation in FROM clause!

- Sample Schema
  - Table1 (cola, colb)
  - Table2 (cola, colc)

```
Q: select   cola
   from     ( select cola, colb
              from   table1
              where colb < 4 ) as a
   where    a.cola > 3

Q1: select  cola, colb
    from    table1
    where   colb < 4

Q2: select  cola
    from    a
    where   cola > 3
```

17/10

Move the subquery in the WHERE clause to the FROM clause

- Joined Subqueries
  - Non-correlated EXISTS or NOT EXISTS
  - The subquery can simply be inserted in the FROM cluase
  - Cartesian product takes place between outer tables and subquery
    - If subquery result is empty, no result is generated
    - Otherwise, multiple copies of the tuples in outer query may result

```
select      column-list1
from        table-list1
where       criteria-list1 and [exists | not exists]
              ( select column-list2
                from   table-lis2
                where criteria-list2 )

select      column-list1
from        table-list1,
            ( select   count(*)
              from     table-lis2
              where    criteria-list2 ) as SQ (ct)
where       criteria-list1 and SQ.ct  [ != | = ] 0
```

18/10

Move the subquery in the WHERE clause to the FROM clause

- Joined Subqueries
  - Non-correlated aggregate subqueries
  - The subquery can simply be inserted in the FROM clause and the resulting table to the tables in the outer query
  - Since the result of the subquery is guaranteed to have only one row, no duplicates can be introduced

```
select      column-list1
from        table-list1
where       criteria-list1 and
            parent-tbl.col OP ( select    aggr-column
                                from      table-lis2
                                where     criteria-list2 )

select      column-list1
from        table-list1,
            ( select   aggr-column
              from     table-lis2
              where    criteria-list2 ) as SQ (aggr-column)
where       criteria-list1 and
            parent-tbl.col OP SQ.aggr-column
```

19/10

Move the subquery in the WHERE clause to the FROM clause

- Joined Subqueries
  - Correlated EXITS/NOT EXISTS subquries
  - Outer join of relation R and an empty relation results in all tuples in R padded with NULLs

```
select      column-list1
from        table-list1
where       criteria-list1 and [EXISTS | NOT EXISTS]
          ( select    column-list2
            from      table-list2
            where     criteria-list2  and table.column OP corr-value)

select      column-list1
from        ( select   column-list1, one-of-table-list.#
            from       table-list1 LEFT OUTER JOIN
                       ( select   column-list2
                         from     table-list2
                         where    criteria-list2) as SQ
                       ON SQ.table.column OP corr-value
            where      criteria-list1
            group by one-of-table-list.#, column-list1
            having counts(SQ.table.column) [ != | = ] 0) as Q
```

20/10