CIS 631 – Database Management System Design

SQL Exercise (3)- Relational Algebra Operations

Dr. H. Assadipour

In the following examples, the SQL queries for the relational algebra exercises discussed in class are presented. The database used for the first 7 queries is the supplier-part database. The remaining queries are on supplier-part-project database.

1. **Get suppliers names who supply part "P2".**

```
SQL> select sname
  2      from s, sp
  3*     where s.s#=sp.s# and sp.p#='P2';

SNAME
---------------
Smith
Jones
Blake
Clark
```

Or:

```
SQL> select distinct s.sname
  2  from s
  3  where s.s# IN
  4     (select sp.s#
  5             from sp
  6             where sp.p# ='P2');

SNAME
---------------
Blake
Clark
Jones
Smith
```

2. **Get suppliers names who supply at least one red part.**

```
SQL> select distinct sname
  2  from s, sp, p
  3* where p.color='Red' and s.s#=sp.s# and p.p#=sp.p#

SNAME
---------------
Clark
Jones
Smith
```

Or:

```
select distinct s.sname
from s
where s.s# IN
      (select sp.s#
      from sp
      where sp.p# IN
            (select p.p#
            from p
            where p.Color = 'Red') );
```

```
SNAME
---------------
Clark
Jones
Smith
```

3. **Get the supplier names for suppliers who do not supply part 'P2'.**

```
SQL> select distinct s.sname
  2  from s
  3  where s.s# NOT IN
  4     (select sp.s#
  5     from sp
  6     where sp.p#='P2');

SNAME
---------------
Adams
```

Or:

```
SQL> select distinct s.sname
  2  from s
  3  where NOT EXISTS
  4     (select *
  5     from sp
  6     where sp.s# = s.s#
  7     and sp.p# = 'P2');

SNAME
---------------
Adams
```

4. **Get the supplier names for suppliers who supply all parts.**

```
select distinct s.sname
from s
where NOT EXISTS
      (select *
```

```
        from p
        where NOT EXISTS
              (select *
              from sp
              where sp.s# = s.s#
              and   sp.p# = p.p#) );
```

```
SNAME
----------
Smith
```

5. **Get supplier numbers who supply at lease one of the parts supplied by supplier 'S2'.**

```
SQL> select  distinct s.s#
2     from s, sp
3     where s.s# = sp.s# and p#  IN
4               (select p#
5               from sp
6               where sp.s# = 'S2')
```

```
S#
---
S1
S2
S3
S4
```

6. **Get all pairs of supplier numbers such that two suppliers are "colocated" (located in the same city).**

```
SQL> l
  1  select A.s# AS SA, B.S# AS SB
  2  from S  A, S  B
  3  where A.city = B. city
  4* and A.s# < B.S#
```

```
SA  SB
--- ---
S2  S3
S1  S4
```

7. **Join the three tables and find the result of natural join with selected attributes.**

```
SQL> select distinct  s.s#, sname, p.p#, p.pname, s.city, status, QTY
  2  from s, sp, p
  3* where s.s#=sp.s# and p.p#=sp.p# and s.city=p.city
```

```
S#  SNAME         P#  PNAME      CITY         STATUS       QTY
--- ---------------  --- ---------- - ---------  ----------  ----------
S1  Smith            P1  Nut        London        20           300
S1  Smith            P4  Screw     London        20           200
S1  Smith            P6  Cog       London        20           100
S2  Jones            P2  Bolt      Paris         10           400
S3  Blake            P2  Bolt      Paris         30           200
S4  Clark            P4  Screw     London        20           300
```

6 rows selected.

8. **Get all shipments where the quantity is in the range 300 to 750 inclusive.**

```
SQL> select spj.*
  2  from spj
  3  where spj.QTY>=300
  4  and   spj.QTY<=750;

S# P# J#       QTY
-- -- -- ----------
S1 P1 J4       700
S2 P3 J1       400
S2 P3 J4       500
S2 P3 J5       600
S2 P3 J6       400
S3 P4 J2       500
S4 P6 J3       300
S4 P6 J7       300
S5 P5 J5       500
S5 P5 J4       400
S5 P6 J4       500
```

11 rows selected.

9. **Get all supplier-number/part-number/project-number triples such that the indicated supplier, part, and project are all colocated (i.e., all in the same city).**

```
SQL> select s.s#, p.p#, J.j#
  2  from s, p, j
  3  where s.city = p.city
  4  and   p.city = j.city;

S#  P#  J#
--- --- --
S1  P1  J5
S4  P1  J5
```

```
S1  P4  J5
S4  P4  J5
S1  P6  J5
S4  P6  J5
S1  P1  J7
S4  P1  J7
S1  P4  J7
S4  P4  J7
S1  P6  J7

S#  P#  J#
--- --- --
S4  P6  J7

12 rows selected.
```

10. **Get all pairs of city names such that a supplier in the first city supplies a project in the second city.**

```
SQL> select distinct s.city as scity, j.city as jcity
  2  from s, j
  3  where exists
  4     (select *
  5     from spj
  6     where spj.s# = s.s#
  7*    and   spj.j# = j.j#);

SCITY      JCITY
---------- ----------
Athens     Athens
Athens     London
Athens     Rome
London     Athens
London     Paris
London     London
Paris      Athens
Paris      Paris
Paris      London
Paris      Oslo
Paris      Rome

11 rows selected.
```

11. **Get all cities in which at least one supplier, part, or project is located.**

```
SQL> select s.city
```

```
  2  from s
  3 union
  4 select p.city
  5 from p
  6 union
  7 select j.city
  8 from j;

CITY
----------
Athens
London
Oslo
Paris
Rome
```

Notice the use of Union operator when the query involves “either or”.

In the next query, the Difference operator (minus) is utilized to provide the pair of supplier #/part # where the indicated supplier does not supply the indicated part. This is done by subtracting all supplier #/part # (from SPJ table) from all possible supplier #/part # (from supplier and pert tables).

12. **Get supplier-number/part-number pairs such that the indicated supplier does not supply the indicated part.**

```
SQL> select s.s#, p.p#
  2  from s, p
  3  minus
  4  select spj.s#, spj.p#
  5  from spj;

S#  P#    S#  P#
--- ---    --- ---

S1  P2    S3  P5
S1  P3    S3  P6
S1  P4    S4  P1
S1  P5    S4  P2
S1  P6    S4  P3
S2  P1    S4  P4
S2  P2    S4  P5
S2  P4
S2  P6
S3  P1
S3  P2    18 rows selected.
```

13. **Get all pairs of part numbers and supplier numbers such that some supplier supplies both**

indicated parts.

```
SQL> l
  1  select distinct spjx.s#, spjx.p# as PA, spjy.p# as PB
  2  from spj  spjx, spj  spjy
  3  where spjx.s# = spjy.s#
  4* and   spjx.p# < spjy.p#;

S#   PA   PB
--   --   --
S2   P3   P5
S3   P3   P4
S5   P1   P2
S5   P1   P3
S5   P1   P4
S5   P1   P5
S5   P1   P6
S5   P2   P3
S5   P2   P4
S5   P2   P5
S5   P2   P6
S5   P3   P4
S5   P3   P5
S5   P3   P6
S5   P4   P5
S5   P4   P6
S5   P5   P6

17 rows selected.
```