

Colours Schema

Customer	
cust#	PK
cname	
fav_colour	
phone#	

Item	
item#	PK
prod#	FK to Product
cust#	FK to Customer
colour	
date_sold	

Product	
prod#	PK
pname	
cost	
maker	FK to Company

Avail_Colours	
prod#	PK, FK to Product
colour	PK

Query 1.

Show, for each customer (reporting the customer's name), the products by name that come in the customer's favourite colour.

```
select C.cname, P.pname
       from Customer C, Avail_Colours A,
          Product P
       where C.fav_colour = A.colour
          and A.prod# = P.prod#;
```

Query 2.

Show, for each customer (reporting the customer's name), the products by name that *do not* come in the customer's favourite colour.

```
select C.cname, P.pname
       from Customer C, Product P
       where C.fav_colour not in (
           select A.colour
                from Avail_Colours A
                where A.prod# = P.prod#
       );
```

Query 2. (B)

```
select C.cname, P.pname
  from Customer C, Product P,
  (   select distinct Q.prod#, A.colour
      from Product Q,
      Avail_Colours A
      except
      select prod#, colour
      from Avail_Colours
  ) as N
 where C.fav_colour = N.colour
    and P.prod# = N.prod#;
```

Query 2. (C)

with

```
    NotAvail (prod#, colour) as (  
        select distinct Q.prod#, A.colour  
        from Product Q,  
             Avail_Colours A
```

```
    except
```

```
        select prod#, colour  
        from Avail_Colours
```

```
)
```

```
select C.cname, P.pname
```

```
    from Customer C, Product P, NotAvail N
```

```
    where C.fav_colour = N.colour
```

```
        and P.prod# = N.prod#;
```

Query 3.

List pairs of customers (columns: first_cust#, first_cname, second_cust#, second_cname) such that the two customers own at least two products in common.

```
select distinct C.cust#, C.cname, D.cust#, D.cname
  from Customer C, Customer D, Item IC, Item JC, Item ID, Item JD
 where C.cust# = IC.cust# and C.cust# = JC.cust# and
       D.cust# = ID.cust# and D.cust# = JD.cust# and
       IC.prod# = ID.prod# and JC.prod# = JD.prod# and
       IC.prod# <> JC.prod# and
       C.cust# < D.cust#;
```

Query 4.

List customers who own items in all the available colours. That is, for every available colour, the customer owns some item in that colour.

```
select cust#, cname
  from Customer
except
select C.cust#, C.cname
  from ( select D.cust#, A.colour
         from Customer D,
              Avail_Colours A
       except
         select I.cust#, I.colour
         from Item I
       ) as M,
  Customer C
where C.cust# = M.cust#;
```

Query 5.

List each customer by name, paired with the product(s) by name that he or she has bought that was the most expensive (cost) of all the products he or she has bought.

```
select C.cname, P.pname
  from ( select distinct cust#, prod#
         from Item
       except
       select I.cust#, I.prod#
         from Item I, Item J, Product Q, Product R
        where I.cust# = J.cust# and
              I.prod# = Q.prod# and J.prod# = R.prod# and
              Q.cost < R.cost
       ) as M,
  customer C, Product P
 where C.cust# = M.cust# and P.prod# = M.prod#;
```


Query 6.

Show, for each customer, the total cost he or she has paid for products in his or her favourite colour.

```
select C.cust#, C.name,  
       sum(P.cost) as total  
from Customer C, Item I,  
       Product P  
where C.cust# = I.cust#  
       and I.prod# = P.prod#  
       and C.fav_colour = I.colour  
group by C.cust#, C.cname;
```

Query 7.

Report with columns `cust#` and `colour` for each customer which colour he or she has spent more on products of that colour than on products of any other colour.

with

```
Colours (cust#, colour, total) as (  
    select I.cust#, I.colour, sum(P.cost)  
    from Item I, Product P  
    where I.prod# = P.prod#  
    group by I.cust#, I.colour  
) ,  
:
```

Query 7. (p.2)

with

⋮

```
Most (cust#, highest) as (  
    select C.cust#, max(C.total)  
    from Colours C  
    group by cust#
```

)

⋮

Query 7. (p.3)

⋮

```
select C.cust#, C.cname, R.colour, M.highest
from Customer C, Colour R, Most M
where C.cust# = R.cust#
      and C.cust# = M.cust# and
      R.total = M.highest;
```

Query 8.

What is the total each customer has spent on items *since* his or her most expensive purchase?

In case of ties for the most expensive purchase, count since the *first* most expensive purchase.

with

```
Expensive (cust#, cost) as (  
    select I.cust#, max(P.cost)  
        from Item I, Product P  
        where I.prod# = P.prod#  
        group by I.cust#  
    ),  
    :
```

Query 8. (p.2)

with

⋮

```
First (cust#, when) as (  
    select I.cust#, min(date_sold)  
        from Item I,  
             Expensive E,  
             Product P  
    where I.cust# = E.cust#  
          and I.prod# = P.prod#  
          and P.cost = E.cost  
    group by I.cust#
```

)

⋮

Query 8. (p.3)

⋮

```
select C.cust#, C.cname, sum(P.cost) as total
  from Customer C, Item I,
       Product P, First F
 where C.cust# = I.cust#
       and C.cust# = F.cust#
       and I.prod# = P.prod#
       and I.date_sold > F.when
 group by C.cust#, C.cname;
```

Query 9.

Which pairs of customers own at least twelve products in common?

with

```
Owned (cust#, prod#) as (  
    select distinct cust#, prod#  
    from Item  
)  
:
```


Query 9. (p.2)

⋮

```
select C.cust#, C.cname,  
       D.cust#, D.cname  
from Customer C, Customer D,  
       Owned P, Owned Q  
where C.cust# = P.cust#  
       and D.cust# = Q.cust#  
       and P.prod# = Q.prod#  
       and C.cust# < D.cust#  
group by C.cust#, C.cname,  
         D.cust#, D.cname  
having count(*) >= 12;
```

Query 10.

Query 5 again: List each customer by name, paired with the product(s) by name that he or she has bought that was the most expensive (cost) of all the products he or she has bought.

Hey, but you have aggregation now!

with

```
Expensive (cust#, highest) as (  
    select I.cust#, max(P.cost)  
        from Item I, Product P  
        where I.prod# = P.prod#  
        group by I.cust#  
    )  
    :
```

Query 10. (p.2)

⋮

```
select C.cname, P.pname
       from Customer C, Item I,
           Product P, Expensive E
where C.cust# = I.cust#
      and C.cust# = E.cust#
      and I.prod# = P.prod#
      and P.cost = E.highest;
```

Recursion: E.g., Bosses

with

```
Boss (emp#, boss#) as (  
    select emp#, boss#  
        from Employee  
union all  
    select E.emp#, B.boss#  
        from Boss B, Employee E  
        where E.boss# = B.emp#  
) ,
```

```
select E.emp#, E.name as ename,  
       B.boss#, F.name as bname  
from Boss B, Employee E, Employee F  
where B.emp# = E.emp#  
and B.boss# = F.emp#;
```

Counting (w/o Aggregation!)

with

```
First (a, b, r#) as (  
    select a, b, 0  
        from Data  
except  
select Y.a, Y.b, 0  
    from Data Y, Data Z  
    where (Y.a > Z.a) or  
          ((Y.a >= Z.a) and (Y.b > Z.b))  
) ,  
:
```

Counting (p.2)

with

⋮

Counter (a, b, r#) as (

 select a, b, r#

 from First

union all

 select D.a, D.b, C.r# + 1

 from Counter C, Data D

 where (D.a > C.a) or

 ((D.a >= C.a) and (D.b > C.b))

)

⋮

Counting (p.3)

```
select a, b, r#  
      from Counter  
except  
select M.a, M.b, M.r#  
      from Counter M, Counter X  
      where M.a = X.a  
            and M.b = X.b  
            and M.r# < X.r#  
order by a, b;
```