Table of Contents

These queries will use set operators to look at which products were ordered in different months. To keep the output down to something we can understand more easily I will limit this to Houseware products(catg_id = 'HW') and to items purchased in the last three months of 2015.

1. Demos for Housewares

Demo 01: Query to show the relevant orders

```
select order_id, prod_id, catg_id, prod_name,  order_date
from oe_orderHeaders  OH
join oe_orderDetails  OD using(order_id)
join prd_products PR using (prod_id)
where catg_id = 'HW'
and extract( month from order_date) in (10,11,12)
and extract(year from order_date) = 2015
order by order_date;
```

```
  ORDER_ID    PROD_ID CATG_ID PROD_NAME                  ORDER_DATE
---------- ---------- ------- ------------------------- ----------
       108       1080 HW      Cornpopper                02-OCT-15
       107       1110 HW      Pancake griddle           02-OCT-15
       110       1090 HW      Gas grill                 12-OCT-15
       113       1080 HW      Cornpopper                08-NOV-15
       112       1110 HW      Pancake griddle           08-NOV-15
       115       1000 HW      Hand Mixer                08-NOV-15
       115       1080 HW      Cornpopper                08-NOV-15
       115       1100 HW      Blender                   08-NOV-15
       408       1071 HW      Iron                      20-NOV-15
       119       1070 HW      Iron                      28-NOV-15
       126       1100 HW      Blender                   15-DEC-15
       127       1110 HW      Pancake griddle           15-DEC-15
       127       1080 HW      Cornpopper                15-DEC-15
       127       1100 HW      Blender                   15-DEC-15
       130       1090 HW      Gas grill                 30-DEC-15

 15 rows selected
```

I could use this query as the basis for the rest of the demos but it is rather long; I could put it in a CTE and make the rest of the query simpler to read. In this case since I want to use it several time, I am going to create it as a view. I will also add a calculated column for the month since I will use that in the demos.

Demo 02: The view definition.

```
create view orderData as
   select order_id, prod_id, catg_id, prod_name,  order_date
   , extract( month from order_date) as order_month
   from oe_orderHeaders  OH
   join oe_orderDetails  OD using(order_id)
   join prd_products PR using (prod_id)
   where catg_id = 'HW'
   and extract( month from order_date) in (10,11,12)
   and extract(year from order_date) = 2015
   order by order_date;;
```

Demo 03: Union for items purchased in either November or December

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
UNION
select  prod_id, catg_id, prod_name
from orderData
where order_month = 12;
```

```
    PROD_ID CATG_ID PROD_NAME
----------- ------- -------------------------
       1000 HW      Hand Mixer
       1070 HW      Iron
       1071 HW      Iron
       1080 HW      Cornpopper
       1090 HW      Gas grill
       1100 HW      Blender
       1110 HW      Pancake griddle
```

Demo 04: Union for items purchased in both November and December

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
INTERSECT
select  prod_id, catg_id, prod_name
from orderData
where order_month = 12;
```

```
    prod_id catg_id prod_name
----------- ------- -------------------------
       1080 HW      Cornpopper
       1100 HW      Blender
       1110 HW      Pancake griddle
```

Demo 05: Orders which contained both a Blender(product 1100) and Pancake Griddle (product 1110)

```
select order_id, order_month
from orderData
where prod_id = 1100
INTERSECT
select order_id,order_month
from orderData
where prod_id = 1110;
```

```
   ORDER_ID ORDER_MONTH
----------- -----------
        127          12
```

Demo 06: Why do we get no rows if we also display the product id?

```
select order_id, order_month, prod_id
from orderData
where prod_id = 1100
INTERSECT
select order_id,order_month, prod_id
from orderData
where prod_id = 1110;
```

```
no rows selected
```

When we use a view (or a CTE) these queries become as simple as the ones in the previous document.

2. Casting to handle syntax rules

The rules for Set operations is that the various select sets must have the same number of columns and the columns must be type compatible. You can use casting functions to handle this.

Demo 07: Using a Union query to display two types of data. Note the use of the CAST function to make the first column union compatible.
The first part of the query gives us product id and prices and the second gives us a descriptive text and an average price.

```
   select cast(prod_id as varchar(6) ) AS "Product ID"
   , prod_list_price as "List Price"
   from Product.products
   where catg_id = 'APL'
UNION  ALL
   select
     '---- avg Price for all Appliances ----'
   , avg(prod_list_price)
   from Product.products
   where catg_id = 'APL'
;
```

```
Product ID                             List Price
-------------------------------------- ----------
4569                                       349.95
1120                                       549.99
1125                                          500
1126                                          850
1130                                       149.99
---- avg Price for all Appliances ----    479.986
```

If you do not use the cast in the first select, you get an error message.

```
SQL Error: ORA-01790: expression must have same datatype as corresponding expression
```

The first select written without the cast `-- select prod_id  AS "Product ID"` produces integer values and then Oracle tries to convert the literal in the second select to an integer. The first select statement sets the data types for the results columns.

3. Using multiple set operators

When you use multiple set operators you need to be concerned about the order of precedence for the operators. This is the same situation as when you use + and * in the same arithmetic expression- which operator is done first.

The rule for Oracle is that the operators are executed in the order in which they appear (top to bottom).

This is not standard ANSI, and Oracle says this might be changed in the future. Use parentheses to change the default order.

Here are a few examples. There are not many rows in the view- compare these row by row.

Demo 08: Items orders in Nov but not in Oct and not in Dec

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
minus
select  prod_id, catg_id, prod_name
from orderData
where order_month = 12
minus
select  prod_id, catg_id, prod_name
from orderData
where order_month = 10;
```

```
    PROD_ID CATG_ID PROD_NAME
----------- ------- -------------------------
       1000 HW      Hand Mixer
       1070 HW      Iron
       1071 HW      Iron
```

Demo 09: Items order in Nov but (not in both Oct and Dec)

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
minus
 (
  select  prod_id, catg_id, prod_name
  from orderData
  where order_month = 12
  intersect
  select  prod_id, catg_id, prod_name
  from orderData
  where order_month = 10
  )
;
```

```
    prod_id catg_id prod_name
----------- ------- -------------------------
       1000 HW      Hand Mixer
       1070 HW      Iron
       1071 HW      Iron
       1100 HW      Blender

(4 row(s) affected)
```

Demo 10: items that were ordered in Nov; take out the items order in Dec; add in the items order in Oct.

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
minus
select  prod_id, catg_id, prod_name
from orderData
where order_month = 12
union
select  prod_id, catg_id, prod_name
from orderData
where order_month = 10
;
```

```
    prod_id catg_id prod_name
----------- ------- -------------------------
       1000 HW      Hand Mixer
       1070 HW      Iron
       1071 HW      Iron
       1080 HW      Cornpopper
       1090 HW      Gas grill
       1110 HW      Pancake griddle

(6 row(s) affected)
```

Demo 11: Why is the result the same as one of the previous demos in this set?

```
select prod_id, catg_id, prod_name
from orderData
where order_month = 11
minus
 (
  select  prod_id, catg_id, prod_name
  from orderData
  where order_month = 12
  union
  select  prod_id, catg_id, prod_name
  from orderData
  where order_month = 10
  );
```

```
    prod_id catg_id prod_name
----------- ------- -------------------------
       1000 HW      Hand Mixer
       1070 HW      Iron
       1071 HW      Iron

(3 row(s) affected)
```