Table of Contents

We will look at views in more detail soon, but the assignment for this unit is easier with a view. A view is a select query that is given a name and the view definition is stored in the database. You can then use the view as a data source in a From clause. For the assignment you can just think of the view as a table expression and you can

- select columns from the view
- use filters against the columns in the view
- join the view to base tables using the same syntax you always use

The view we will use in the assignment and also in the subqueries demo contains a join of several tables and the purpose of the view is to hide that complexity from the queries we want to run.

1. Creating a Simple View

Demo 01: This is a very simple view that just exposes three columns for the products table for some of the rows. With Oracle you can use a "Create or Replace View" syntax

```
create or replace View prd_HW_APL
as (
  select
    prod_id
  , prod_name
  , prod_list_price
  from prd_products
  where catg_id in ('APL', 'HW')
);
```

2. Using the View

Demo 02: select data from the view

```
select *
from prd_HW_APL;
```

PROD_ID	PROD_NAME	PROD_LIST_PRICE
1000	Hand Mixer	125
1070	Iron	25.5
1071	Iron	25.5
1072	Iron	25.5
1080	Cornpopper	25
1090	Gas grill	149.99
1100	Blender	49.99
1160	Mixer Deluxe	149.99
4569	Mini Dryer	349.95
4575	Electric can opener	49.95
1110	Pancake griddle	49.99
1120	Washer	549.99
1125	Dryer	500
1126	WasherDryer	850
1130	Mini Freezer	149.99

Demo 03: select data from the view with a filter

```
select *
from prd_HW_APL
where prod_list_price > 100;
```

PROD_ID	PROD_NAME	PROD_LIST_PRICE
1000	Hand Mixer	125
1090	Gas grill	149.99
1160	Mixer Deluxe	149.99
4569	Mini Dryer	349.95
1120	Washer	549.99
1125	Dryer	500
1126	WasherDryer	850
1130	Mini Freezer	149.99

Demo 04: select data from the view with a join to a base table

```
select
  PR.prod_id
, quantity_ordered * quoted_price as extPrice
, order_id
from prd_HW_APL PR
join oe_OrderDetails OD On PR.prod_id = OD.prod_id
where order_id Between 110 and 115;;
```

PROD_ID	EXTPRICE	ORD_ID
1090	149.99	110
1130	149.99	110
1110	99.98	112
1080	22.5	113
1130	625	114
1000	200	115
1120	1900	115
1080	25	115
1100	180	115

3. A more complex view

The next view is more complex, including several joins and filters and renaming the column names

Demo 05: a more complex view

```
create or replace View oe_customer_orders
as (
  select
    OH.order_id as invoice
  , OH.order_date as orderDate
  , OH.customer_id as custID
  , PR.catg_id as category
  , OD.prod_id as itemPurchased
  from oe_orderHeaders OH
  join oe_OrderDetails OD On OH.order_id = OD.order_id
  join prd_products PR On OD.prod_id = PR.prod_id
  where OD.quoted_price > 0
  and OD.quantity_ordered > 0
);
```

Demo 06: When you use the second view, refer to the column names defined in the view. Note that if an order has more than one detail line, the view returns more than one row

```
select
  invoice
, itemPurchased
from oe_customer_orders
where extract(month from orderDate) = 6;
```

```
   INVOICE ITEMPURCHASED
---------- -------------
       301          1100
       302          1140
       302          1040
       303          1000
       306          1120
       306          1125
       307          1120
       307          1125
       312          1040
       312          1060
       312          1060
       312          1050
       313          1000
       324          4576
       378          1125
       378          1120
       390          1010
       395          1010
       540          1110
       540          1080
       540          1152
```