

1. Create Materialized View for 2011 Shipments

CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2011

BUILD IMMEDIATE

REFRESH COMPLETE ON DEMAND AS

SELECT inventory_fact.custvendorkey, date_dim.datekey, SUM(extcost) AS ExtCostSum, SUM(quantity)
AS QuantSum, COUNT(*) AS TransTotal

FROM inventory_fact, cust_vendor_dim, date_dim

WHERE inventory_fact.custvendorkey = cust_vendor_dim.custvendorkey

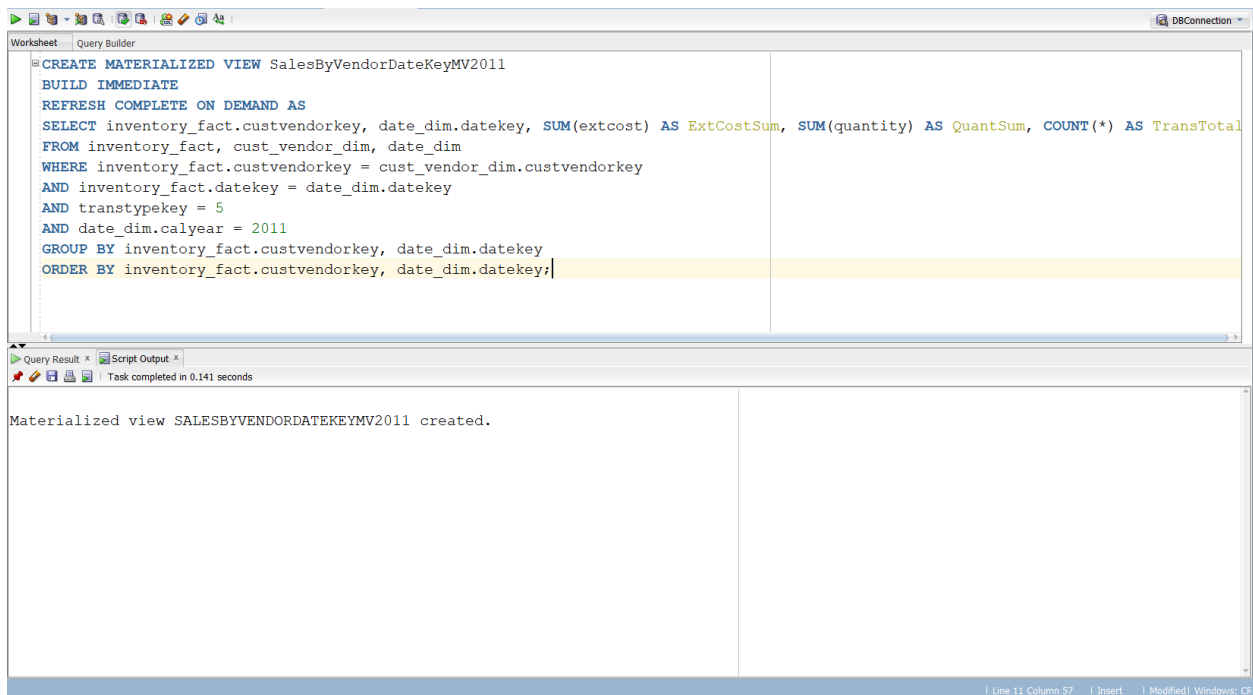
AND inventory_fact.datekey = date_dim.datekey

AND transtypekey = 5

AND date_dim.cyear = 2011

GROUP BY inventory_fact.custvendorkey, date_dim.datekey

ORDER BY inventory_fact.custvendorkey, date_dim.datekey;



2. Create Materialized View for 2012 Shipments

```
CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2012
```

```
BUILD IMMEDIATE
```

```
REFRESH COMPLETE ON DEMAND AS
```

```
SELECT inventory_fact.custvendorkey, date_dim.datekey, SUM(extcost) AS ExtCostSum, SUM(quantity)  
AS QuantSum, COUNT(*) AS TransTotal
```

```
FROM inventory_fact, cust_vendor_dim, date_dim
```

```
WHERE inventory_fact.custvendorkey = cust_vendor_dim.custvendorkey
```

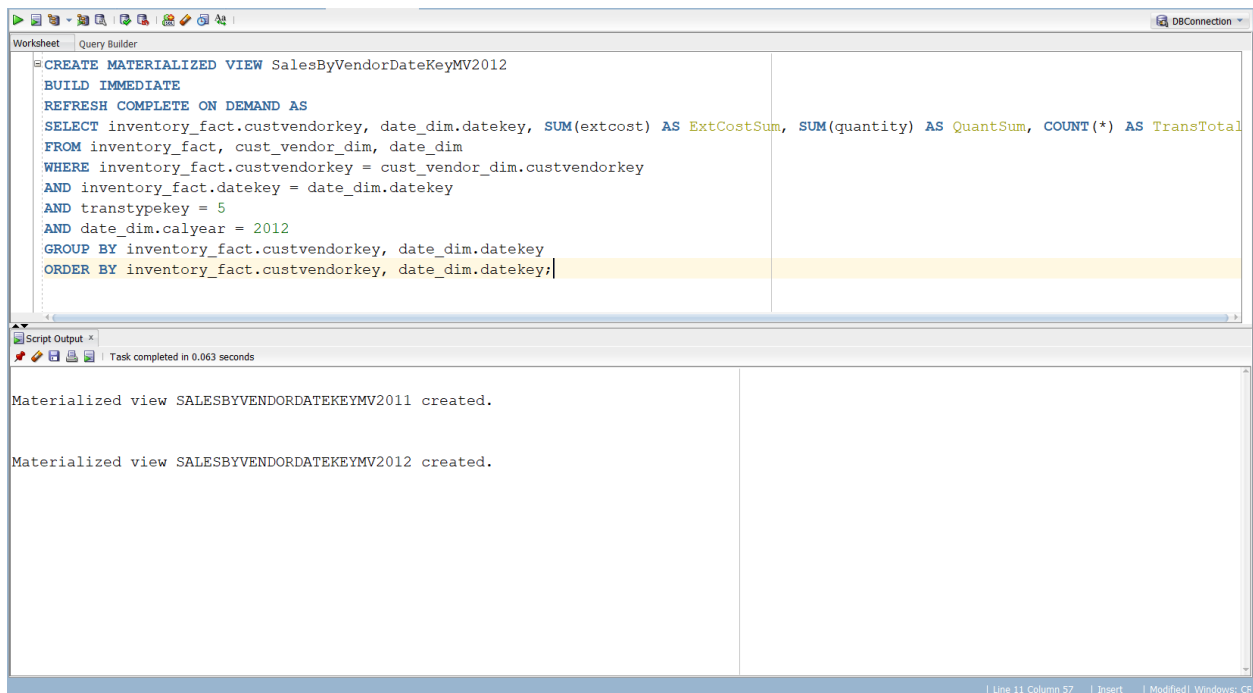
```
AND inventory_fact.datekey = date_dim.datekey
```

```
AND transtypekey = 5
```

```
AND date_dim.cyear = 2012
```

```
GROUP BY inventory_fact.custvendorkey, date_dim.datekey
```

```
ORDER BY inventory_fact.custvendorkey, date_dim.datekey;
```



3. Rewrite Query 1 of the Module 2 Assignment

```

SELECT date_dim.calmonth, addrcatcode1, SUM(QuantSum) QuantSum2, SUM(ExtCostSum) AS
ExtCostSum2

FROM SalesByVendorDateKeyMV2011, cust_vendor_dim, date_dim

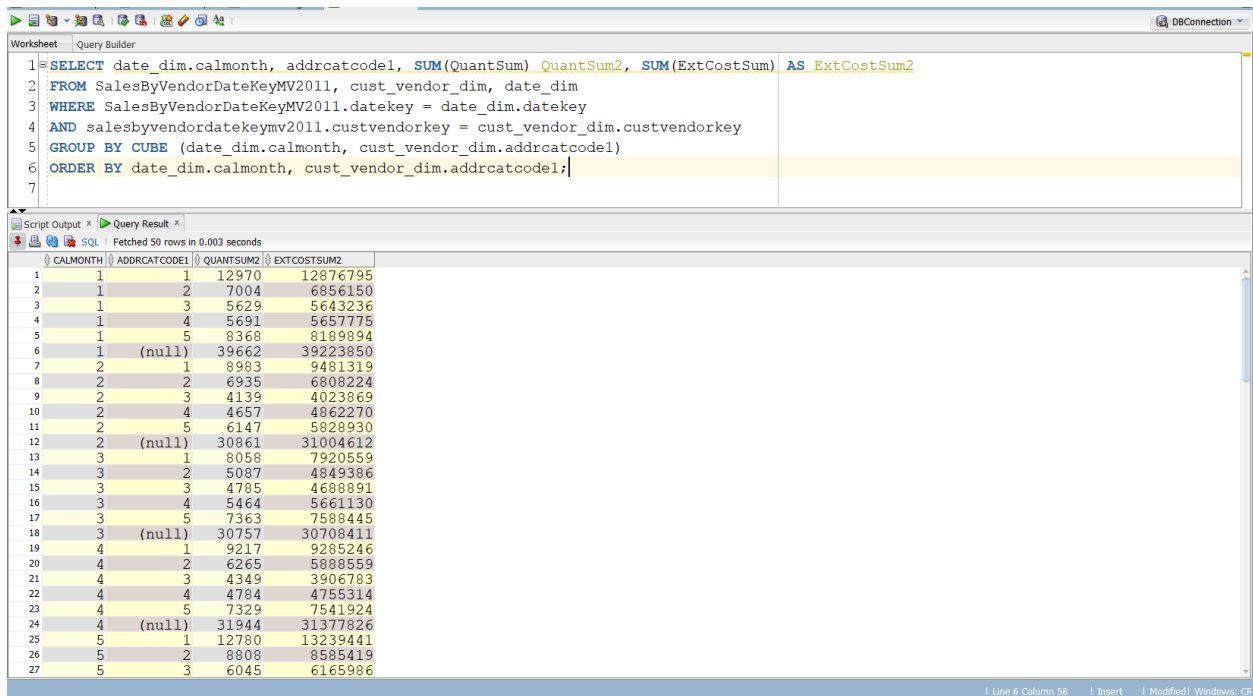
WHERE SalesByVendorDateKeyMV2011.datekey = date_dim.datekey

AND salesbyvendordatekeymv2011.custvendorkey = cust_vendor_dim.custvendorkey

GROUP BY CUBE (date_dim.calmonth, cust_vendor_dim.addrcatcode1)

ORDER BY date_dim.calmonth, cust_vendor_dim.addrcatcode1;

```



The screenshot shows a SQL query builder window with a query and its results. The query is as follows:

```

1 SELECT date_dim.calmonth, addrcatcode1, SUM(QuantSum) QuantSum2, SUM(ExtCostSum) AS ExtCostSum2
2 FROM SalesByVendorDateKeyMV2011, cust_vendor_dim, date_dim
3 WHERE SalesByVendorDateKeyMV2011.datekey = date_dim.datekey
4 AND salesbyvendordatekeymv2011.custvendorkey = cust_vendor_dim.custvendorkey
5 GROUP BY CUBE (date_dim.calmonth, cust_vendor_dim.addrcatcode1)
6 ORDER BY date_dim.calmonth, cust_vendor_dim.addrcatcode1;

```

The results are displayed in a table with 4 columns: CALMONTH, ADDRCATCODE1, QUANTSUM2, and EXTCOSTSUM2. The table contains 27 rows of data, showing the sum of QuantSum and ExtCostSum for each combination of calmonth and addrcatcode1.

	CALMONTH	ADDRCATCODE1	QUANTSUM2	EXTCOSTSUM2
1	1	1	12970	12876795
2	1	2	7004	6856150
3	1	3	5629	5643236
4	1	4	5691	5657775
5	1	5	8368	8189894
6	1	(null)	39662	39223850
7	2	1	8983	9481319
8	2	2	6935	6808224
9	2	3	4139	4023869
10	2	4	4657	4862270
11	2	5	6147	5828930
12	2	(null)	30861	31004612
13	3	1	8058	7920559
14	3	2	5087	4849386
15	3	3	4785	4688891
16	3	4	5464	5661130
17	3	5	7363	7588445
18	3	(null)	30757	30708411
19	4	1	9217	9285246
20	4	2	6265	5888559
21	4	3	4349	3906783
22	4	4	4784	4755314
23	4	5	7329	7541924
24	4	(null)	31944	31377826
25	5	1	12780	13239441
26	5	2	8808	8585419
27	5	3	6045	6165986

4. Rewrite Query 2 of the Module 2 Assignment

```

SELECT calquarter, zip, name, SUM(extcost) AS SumExtCost, SUM(transtotal) AS TransTotal

FROM (

SELECT calquarter, zip, name, SUM(extcostsum) AS extcost, SUM(transtotal) AS transtotal

FROM SalesByVendorDateKeyMV2011, cust_vendor_dim, date_dim

WHERE salesbyvendordatekeymv2011.custvendorkey = cust_vendor_dim.custvendorkey

AND SalesByVendorDateKeyMV2011.datekey = date_dim.datekey

```

GROUP BY calquarter, zip, name

UNION

SELECT calquarter, zip, name, SUM(extcostsum) AS extcost, SUM(transtotal) AS transtotal

FROM SalesByVendorDateKeyMV2012, cust_vendor_dim, date_dim

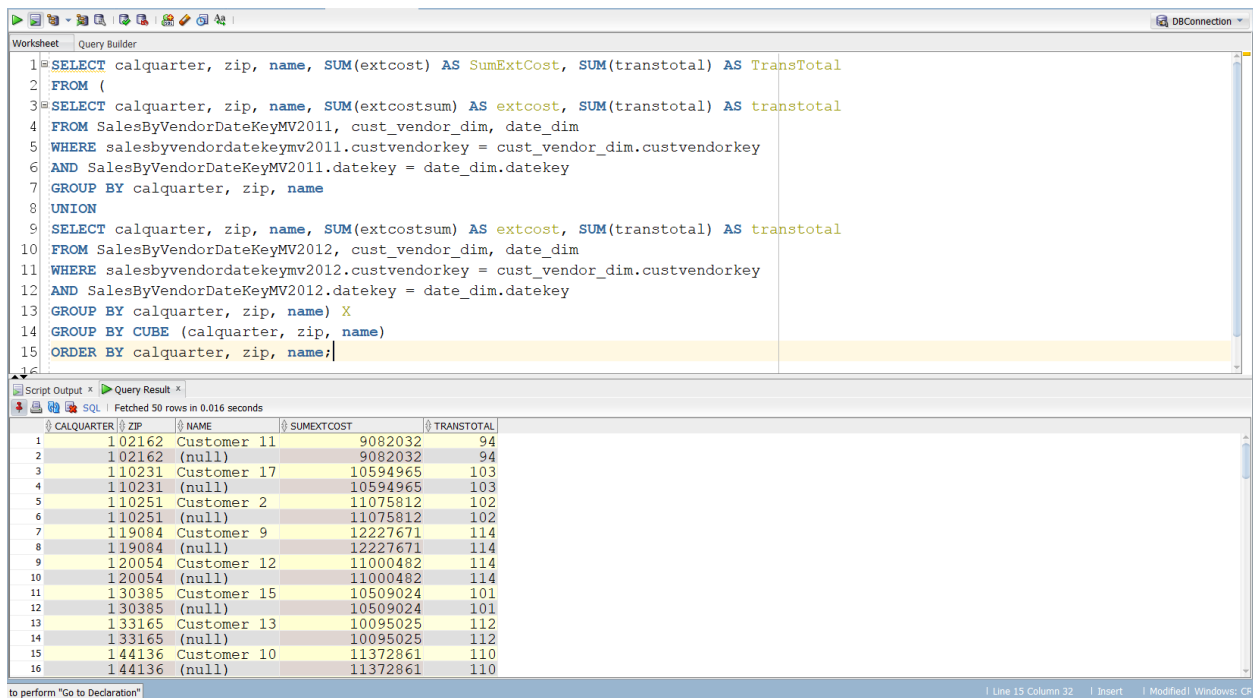
WHERE salesbyvendordatekeymv2012.custvendorkey = cust_vendor_dim.custvendorkey

AND SalesByVendorDateKeyMV2012.datekey = date_dim.datekey

GROUP BY calquarter, zip, name) X

GROUP BY CUBE (calquarter, zip, name)

ORDER BY calquarter, zip, name;



Worksheet - Query Builder

```
1 SELECT calquarter, zip, name, SUM(extcost) AS SumExtCost, SUM(transtotal) AS TransTotal
2 FROM (
3 SELECT calquarter, zip, name, SUM(extcostsum) AS extcost, SUM(transtotal) AS transtotal
4 FROM SalesByVendorDateKeyMV2011, cust_vendor_dim, date_dim
5 WHERE salesbyvendordatekeymv2011.custvendorkey = cust_vendor_dim.custvendorkey
6 AND SalesByVendorDateKeyMV2011.datekey = date_dim.datekey
7 GROUP BY calquarter, zip, name
8 UNION
9 SELECT calquarter, zip, name, SUM(extcostsum) AS extcost, SUM(transtotal) AS transtotal
10 FROM SalesByVendorDateKeyMV2012, cust_vendor_dim, date_dim
11 WHERE salesbyvendordatekeymv2012.custvendorkey = cust_vendor_dim.custvendorkey
12 AND SalesByVendorDateKeyMV2012.datekey = date_dim.datekey
13 GROUP BY calquarter, zip, name) X
14 GROUP BY CUBE (calquarter, zip, name)
15 ORDER BY calquarter, zip, name;
```

Script Output x Query Result x

SQL | Fetched 50 rows in 0.016 seconds

	CALQUARTER	ZIP	NAME	SUMEXTCOST	TRANSTOTAL
1	102162	Customer 11		9082032	94
2	102162	(null)		9082032	94
3	110231	Customer 17		10594965	103
4	110231	(null)		10594965	103
5	110251	Customer 2		11075812	102
6	110251	(null)		11075812	102
7	119084	Customer 9		12227671	114
8	119084	(null)		12227671	114
9	120054	Customer 12		11000482	114
10	120054	(null)		11000482	114
11	130385	Customer 15		10509024	101
12	130385	(null)		10509024	101
13	133165	Customer 13		10095025	112
14	133165	(null)		10095025	112
15	144136	Customer 10		11372861	110
16	144136	(null)		11372861	110

to perform "Go to Declaration"

Line 15 Column 32 | Insert | Modified | Windows: CR