

Solution for the Module 2 Assignment

1: Sales Order Shipments by Month and Category Code1

Write an SQL statement to display the sum of the extended cost and the sum of the quantity. The results should include data for shipments (transaction type 5) in calendar year 2021. Summarize the result by calendar month and Address Category Code 1. The result should include the grouped columns and the full totals for every combination of grouped columns. Do not use the GROUPING SETS and UNION operators.

```
-- Assignment requires CUBE solution
select CalMonth, AddrCatCode1, sum(ExtCost) as tot_cost,
       sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
       d.Calyear = 2021 and
       i.CustVendorKey = c.CustVendorKey and
       i.DateKey = d.DateKey
group by CUBE(AddrCatCode1, d.calmonth);
```

```
-- GROUPING SETS solution
-- Solution not in the problem specification
select CalMonth, AddrCatCode1, sum(ExtCost) as tot_cost,
       sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
       d.Calyear = 2021 and
       i.CustVendorKey = c.CustVendorKey and
       i.DateKey = d.DateKey
group by GROUPING SETS((AddrCatCode1, d.calmonth), AddrCatCode1, d.calmonth,
());
```

2: Sales Order Shipments by Name, Zip, and Quarter

Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2021 and 2022. Summarize the result by calendar quarter, customer zip code, and customer name. The result should include the grouped columns and the full set of subtotals for every combination of grouped columns. Do not use the CUBE and UNION operators.

```
-- Assignment requires GROUPING SETS solution
select Name, Zip, CalQuarter, sum(ExtCost) as tot_cost, count(*) as Cnt
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
       d.Calyear BETWEEN 2021 AND 2022 AND
       d.datekey = i.datekey and
       i.CustVendorKey = c.CustVendorKey AND
       i.DateKey = d.DateKey
group by GROUPING SETS((c.Name, c.Zip, d.CalQuarter), (c.Name, c.Zip),
```

```
(c.Name, d.CalQuarter), (c.Zip, d.CalQuarter),
c.Name, c.Zip, d.CalQuarter, ());

-- CUBE solution
-- Solution not in the problem specification
select Name, Zip, CalQuarter, sum(ExtCost) as tot_cost, count(*) as Cnt
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear BETWEEN 2021 AND 2022 AND
      d.datekey = i.datekey and
      i.CustVendorKey = c.CustVendorKey AND
      i.DateKey = d.DateKey
group by CUBE(c.Name, Zip, d.CalQuarter);
```

3: Transfers by Company and Branch Plant

Write an SQL statement to display the sum of the extended cost and the sum of the quantity. The results should include data for transfers (transaction type 2). Summarize the result by company name and branch plant name. The result should include the grouped columns and a partial set of subtotals in order of the grouped columns (company name and branch plant name). Transfer quantities by design should sum to zero across all companies so that the grand total should be 0 for the sum of quantity and extended cost. Do not use the GROUPING SETS and UNION operators.

```
-- Assignment requires ROLLUP solution.
select CompanyName, BPName, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, company_dim cd, branch_plant_dim bp
where TransTypeKey = 2 and
      bp.CompanyKey = cd.CompanyKey and
      i.branchplantkey = bp.branchplantkey
group by ROLLUP (CompanyName, BPName);

-- GROUPING SETS solution
select CompanyName, BPName, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, company_dim cd, branch_plant_dim bp
where TransTypeKey = 2 and
      bp.CompanyKey = cd.CompanyKey and
      i.branchplantkey = bp.branchplantkey
group by GROUPING SETS ((CompanyName, BPName), CompanyName, ());
```

4: Inventory Transactions by Transaction Description, Company, and Branch Plant

Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for all transaction types. Summarize the result by transaction description, company name, and branch plant name. The result should include the grouped columns and partial totals in order of the grouped columns (transaction description, company name, and branch plant name). Do not use the ROLLUP and UNION operators.

```
-- Assignment requires GROUPING SETS solution
select TransDescription, CompanyName, BPName, sum(ExtCost) as Tot_cost,
       count(*) as Count
from inventory_fact i, trans_type_dim tt, branch_plant_dim bp, company_dim c
where i.TransTypeKey = tt.TransTypeKey and
      i.BranchPlantKey = bp.BranchPlantKey and
      bp.CompanyKey = c.CompanyKey
group by GROUPING SETS((TransDescription, CompanyName, BPName),
                        (TransDescription, CompanyName),
                        (TransDescription, BPName),
                        (TransDescription, CompanyName, BPName));

-- ROLLUP solution
select TransDescription, CompanyName, BPName, sum(ExtCost) as Tot_cost,
       count(*) as Count
from inventory_fact i, trans_type_dim tt, branch_plant_dim bp, company_dim c
where i.TransTypeKey = tt.TransTypeKey and
      i.BranchPlantKey = bp.BranchPlantKey and
      bp.CompanyKey = c.CompanyKey
group by ROLLUP(TransDescription, CompanyName, BPName);
```

5: Adjustments by Part Number

Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2021 and 2022. Summarize the result by calendar year, calendar quarter, and customer name. The result should show the grouped columns and the normal set of group by results plus partial subtotals for year and quarter concatenated with customer name. Do not use the GROUPING SETS operator and UNION operators. (Hint: see partial ROLLUP example in lesson 5).

```
-- Partial ROLLUP solution
select Name, CalYear, CalQuarter, sum(ExtCost) as tot_cost, count(*) as Cnt
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.CalYear BETWEEN 2021 AND 2022 AND
      d.datekey = i.datekey and
      i.CustVendorKey = c.CustVendorKey AND
      i.DateKey = d.DateKey
group by c.name, ROLLUP(CalYear, d.CalQuarter);
```

6: Rewritten Query 1 without the CUBE or GROUPING SETS operators

```
select CalMonth, AddrCatCode1, sum(ExtCost) as tot_cost,
       sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.CalYear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey
group by d.calmonth, AddrCatCode1
UNION ALL
select CalMonth, NULL, sum(ExtCost) as tot_cost,
       sum(Quantity) as tot_qty
```

```

from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey
group by d.calmonth
UNION ALL
select NULL, AddrCatCode1, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey
group by AddrCatCode1
UNION ALL
select NULL, NULL, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey;

```

7: Rewritten Query 3 without the ROLLUP or GROUPING SETS operators

```

-- UNION and UNION ALL produce the same result
select CompanyName, BPName, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, company_dim cd, branch_plant_dim bp
where TransTypeKey = 2 and
      bp.CompanyKey = cd.CompanyKey and
      i.branchplantkey = bp.branchplantkey
group by CompanyName, BPName
UNION ALL
select CompanyName, NULL, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, company_dim cd, branch_plant_dim bp
where TransTypeKey = 2 and
      bp.CompanyKey = cd.CompanyKey and
      i.branchplantkey = bp.branchplantkey
group by CompanyName
UNION ALL
select NULL, NULL, sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, company_dim cd, branch_plant_dim bp
where TransTypeKey = 2 and
      bp.CompanyKey = cd.CompanyKey and
      i.branchplantkey = bp.branchplantkey;

```

Query 8: Sales Order Shipments by Name and Combination of Year and Quarter

Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2021 and 2022. Summarize the result by calendar year, calendar quarter, and customer name. The result should include the grouped columns and the full set of subtotals for customer name and the combination of year and quarter. Do not use the GROUPING SETS operator. (Hint: see the composite column example in lesson 5).

```
-- CUBE with composite columns
select Name, CalYear, CalQuarter, sum(ExtCost) as tot_cost, count(*) as Cnt
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear BETWEEN 2021 AND 2022 AND
      d.datekey = i.datekey and
      i.CustVendorKey = c.CustVendorKey AND
      i.DateKey = d.DateKey
group by CUBE(c.Name, (CalYear, d.CalQuarter));
```

Query 9: Sales Order Shipments by Month and Category Code1 with Group Number

Write an SQL statement to display the sum of the extended cost and the sum of the quantity. The results should include data for shipments (transaction type 5) in calendar year 2021. Summarize the result by calendar month and Address Category Code 1. The result should include the grouped columns and the full set of subtotals for every combination of grouped columns along with the hierarchical group number for both grouping columns. Do not use the GROUPING SETS operator. (Hint: see the group functions slide in lesson 5).

```
-- CUBE with GROUPING_ID function
-- Oracle solution
select CalMonth, AddrCatCode1,
      GROUPING_ID(AddrCatCode1, d.CalMonth) AS GroupNo,
      sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey
group by CUBE(AddrCatCode1, d.calmonth);
```

```
-- PostgreSQL solution
select CalMonth, AddrCatCode1,
      GROUPING(AddrCatCode1, d.CalMonth) AS GroupNo,
      sum(ExtCost) as tot_cost,
      sum(Quantity) as tot_qty
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear = 2021 and
      i.CustVendorKey = c.CustVendorKey and
      i.DateKey = d.DateKey
group by CUBE(AddrCatCode1, d.calmonth);
```

Query 10: Sales Order Shipments with Subtotals by Name and Partial Subtotals for Year and Quarter

Write an SQL statement to display the sum of the extended cost and the number of inventory transactions. The results should include data for shipments (transaction type 5) in calendar years 2021 and 2022. Summarize the result by calendar year, calendar quarter, and customer name. The result should include the grouped columns and subtotals for customer name along with partial subtotals for year and quarter. Do not include the normal GROUP BY totals in the result. (Hint: see the nested rollup example in lesson 5). Do not use the UNION operator.

```
-- Nested ROLLUP inside GROUPING SETS operator
select Name, CalYear, CalQuarter, sum(ExtCost) as tot_cost, count(*) as Cnt
from inventory_fact i, cust_vendor_dim c, date_dim d
where TransTypeKey = 5 and
      d.Calyear BETWEEN 2021 AND 2022 AND
      d.datekey = i.datekey and
      i.CustVendorKey = c.CustVendorKey AND
      i.DateKey = d.DateKey
group by GROUPING SETS (c.Name, ROLLUP(CalYear, d.CalQuarter));
```