Solutions for the Module 3 Assignment

Query 1: Ranking within the entire result

Use the RANK function to rank customers by the sum of external cost for shipments (transaction type 5). You should use the entire result as a single partition. The result should include the customer name, sum of the external cost, and rank.

```
SELECT c.Name, SUM(ExtCost) AS SumExtCost,
  RANK() OVER (ORDER BY SUM(ExtCost) DESC) ExtCostRank
  FROM inventory_fact i, cust_vendor_dim c
  WHERE i.CustVendorKey = c.CustVendorKey AND TransTypeKey = 5
 GROUP BY c.Name;
```

Query 2: Ranking within a partition

Use the RANK function to rank customers by the sum of external cost for shipments (transaction type 5). You should partition the rank values by customer state. The result should include the customer state, customer name, sum of the external cost, and rank. You should order the result by customer state.

```
SELECT c.State, c.Name, SUM(ExtCost) AS SumExtCost,
  RANK() OVER (PARTITION BY c.State ORDER BY SUM(ExtCost) DESC) ExtCostRank
  FROM inventory_fact i, cust_vendor_dim c
  WHERE i.CustVendorKey = c.CustVendorKey AND TransTypeKey = 5
 GROUP BY  c.State, c.Name
 ORDER BY c.State;
```

Query 3: Ranking and dense ranking

Use both RANK and DENSE_RANK functions to rank by the count of inventory transactions for shipments (transaction type 5). You should use the entire result as a single partition. The result should include the customer name, count of transactions, rank, and dense rank.

```
SELECT c.Name, COUNT(*) AS ShipmentCount,
  RANK() OVER (ORDER BY COUNT(*) DESC) ShipmentCounttRank,
  DENSE_RANK() OVER (ORDER BY COUNT(*) DESC) ShipmentCounttDenseRank
  FROM inventory_fact i, cust_vendor_dim c
  WHERE i.CustVendorKey = c.CustVendorKey AND TransTypeKey = 5
 GROUP BY c.Name;
```

Query 4: Cumulative external costs for the entire result

Calculate the cumulative sum of external cost ordered by customer zip code, calendar year, and calendar month. The result should include the customer zip code, calendar year, calendar month, sum of the external cost, and cumulative sum of the external cost. Note that the cumulative external cost is the sum of the external cost in the current row plus the cumulative sum of external costs in all previous rows.

```
select C.Zip, CalYear, CalMonth, sum(ExtCost) as tot_cost,
      SUM(SUM(ExtCost)) OVER (ORDER BY C.Zip, CalYear, CalMonth
       ROWS UNBOUNDED PRECEDING) AS CumSumExtCost
FROM inventory_fact i, cust_vendor_dim c, date_dim d
  WHERE i.CustVendorKey = c.CustVendorKey
    AND TransTypeKey = 5  AND i.DateKey = d.DateKey
group by c.Zip, CalYear, CalMonth;
```

Query 5: Cumulative external costs for a partition

Calculate the cumulative sum of external cost ordered by customer zip code, calendar year, and calendar month. Restart the cumulative external cost after each combination of zip code and calendar year. The result should include the customer zip code, calendar year, calendar month, sum of the external cost, and cumulative sum of the external cost. Note that the cumulative external cost is the sum of the external cost in the current row plus the cumulative sum of external costs in all previous rows of the store zip code and years. The value of cumulative external cost resets in each partition (new value for zip code and year).

```
select C.Zip, CalYear, CalMonth, sum(ExtCost) as tot_cost,
      SUM(SUM(ExtCost)) OVER (PARTITION BY C.Zip, CalYear
      ORDER BY C.Zip, CalYear, CalMonth ROWS UNBOUNDED PRECEDING) AS
CumSumExtCost
FROM inventory_fact i, cust_vendor_dim c, date_dim d
  WHERE i.CustVendorKey = c.CustVendorKey
    AND TransTypeKey = 5  AND i.DateKey = d.DateKey
group by c.Zip, CalYear, CalMonth;
```

Query 6: Ratio to report applied to the entire result

Calculate the ratio to report of the sum of external cost for adjustments (transaction type 1). You should sort on descending order by sum of external cost. The result should contain the second item id, sum of external cost, and ratio to report.

Oracle solution

```
select SecondItemId, sum(ExtCost) as tot_cost,
      RATIO_TO_REPORT(SUM(ExtCost)) OVER () AS SumExtCostRatio
from inventory_fact i, item_master_dim im
where TransTypeKey = 1 and
     i.ItemMasterKey = im.ItemMasterKey
```

```
group by SecondItemId
ORDER BY SUM(ExtCost) DESC;
```

PostgreSQL solutions

Using SUM as an analytic function

```
SELECT SECONDITEMID, SUM(EXTCOST) AS TOTAL_COST, (SUM(EXTCOST) /
SUM(SUM(EXTCOST)) OVER ()) AS RATIO_TO_REPORT
FROM INVENTORY_FACT F, ITEM_MASTER_DIM I
WHERE TRANSTYPEKEY = 1
    AND F.ItemMasterKey = I.ItemMasterKey
GROUP BY SECONDITEMID
ORDER BY TOTAL_COST DESC;
```

Not using analytic functions

```
SELECT X1.seconditemid, ItemExtCost, ItemExtCost/SumExtCost AS ItemExtCostRatio
FROM
  ( SELECT seconditemid, CAST(SUM(extcost) AS NUMERIC) AS ItemExtCost
        FROM inventory_fact, item_master_dim
        WHERE item_master_dim.itemmasterkey = inventory_fact.itemmasterkey
            AND transtypekey = 1
        GROUP BY seconditemid) X1,
  ( SELECT CAST(SUM(extcost) AS NUMERIC) AS SumExtCost
        FROM inventory_fact, item_master_dim
        WHERE item_master_dim.itemmasterkey = inventory_fact.itemmasterkey
            AND transtypekey = 1) X2
ORDER BY ItemExtCost DESC;
```

Query 7: Ratio to report applied to a partition

Calculate the ratio to report of the sum of external cost for adjustments (transaction type 1) with partitioning on calendar year. You should sort on descending order by calendar year and sum of external cost. The result should contain the calendar year, second item id, sum of external cost, and ratio to report.

Oracle solution

```
select CalYear, SecondItemId, sum(ExtCost) as tot_cost,
      RATIO_TO_REPORT(SUM(ExtCost))
       OVER (PARTITION BY CalYear) AS SumExtCostRatio
from inventory_fact i, item_master_dim im, date_dim d
where TransTypeKey = 1 AND
     i.ItemMasterKey = im.ItemMasterKey AND i.DateKey = d.DateKey
group by CalYear, SecondItemId
ORDER BY CalYear, sum(ExtCost) DESC;
```

PostgreSQL solutions
Using SUM as an analytic function

```
SELECT CalYear, SECONDITEMID, SUM(EXTCOST) AS TOTAL_COST,
(SUM(EXTCOST) / SUM(SUM(EXTCOST)) OVER (PARTITION BY CalYear)) AS
RATIO_TO_REPORT
FROM INVENTORY_FACT F, ITEM_MASTER_DIM I, Date_Dim D
WHERE TRANSTYPEKEY = 1
    AND F.ItemMasterKey = I.ItemMasterKey
    AND F.DateKey = D.DateKey
GROUP BY CalYear, SECONDITEMID
ORDER BY CalYear, TOTAL_COST DESC;
```

Not using analytic functions

```
SELECT X1.CalYear, X1.seconditemid, ItemExtCost,
          ItemExtCost/SumExtCost AS ItemExtCostRatio
FROM
  ( SELECT D.CalYear, seconditemid, CAST(SUM(extcost) AS NUMERIC) AS ItemExtCost
        FROM inventory_fact F, item_master_dim I, Date_Dim D
        WHERE I.itemmasterkey = F.itemmasterkey
            AND F.DateKey = D.DateKey
            AND transtypekey = 1
        GROUP BY CalYear, seconditemid) X1,
  ( SELECT D2.CalYear, CAST(SUM(extcost) AS NUMERIC) AS SumExtCost
        FROM inventory_fact F2, item_master_dim I2, Date_Dim D2
        WHERE I2.itemmasterkey = F2.itemmasterkey
            AND F2.DateKey = D2.DateKey
            AND transtypekey = 1
        GROUP BY CalYear) X2
 WHERE X1.CalYear = X2.CalYear
ORDER BY X1.CalYear, ItemExtCost DESC;
```

Query 8: Cumulative distribution functions for carrying cost of branch plants

Calculate the rank, percent_rank, and cume_dist functions of the carrying cost in the branch_plant_dim table. The result should contain the BPName, CompanyKey, CarryingCost, rank, percent_rank, and cume_dist.

```
SELECT BPName, CompanyKey, CarryingCost,
     RANK() OVER (ORDER BY CarryingCost) As RankCarryCost,
     PERCENT_RANK() OVER (ORDER BY CarryingCost) As PercentRankCarryCost,
     CUME_DIST() OVER (ORDER BY CarryingCost) As CumDistCarryCost
FROM branch_plant_dim;
```

Query 9: Determine worst performing plants

Determine the branch plants with the highest carrying costs (top 15%). The result should contain the BPName, CompanyKey, CarryingCost, and cume_dist.

```
SELECT BPName, CompanyKey, CarryingCost, CumDistCarryCost
FROM ( SELECT BPName, CompanyKey, CarryingCost,
     CUME_DIST() OVER (ORDER BY CarryingCost) As CumDistCarryCost
FROM branch_plant_dim ) X
WHERE CumDistCarryCost >= 0.85;
```

Query 10: Cumulative distribution of external cost for Colorado inventory

Calculate the cumulative distribution of external cost for Colorado inventory (condition on customer state). The result should contain the external cost and cume_dist, ordered by external cost. You should eliminate duplicate cumulative distribution values in the result.

```
SELECT DISTINCT ExtCost,
     CUME_DIST() OVER (ORDER BY ExtCost) As CumDistExtCost
FROM cust_vendor_dim cvd, inventory_fact if
WHERE cvd.CustVendorKey = if.CustVendorKey AND State = 'CO'
ORDER BY ExtCost;
```