Solution for the Module 4 Assignment

1. CREATE MATERIALIZED VIEW Statement for 2021 Shipments

```
-- Oracle solution
CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2021
  PCTFREE 0
  BUILD IMMEDIATE AS
  SELECT f.custvendorkey, D.DateKey,
         SUM(quantity) AS SumQty, SUM(extcost) AS ExtCost,
         COUNT(*) AS NumTrans
   FROM date_dim d, inventory_fact f
   WHERE f.datekey = d.datekey
     AND f.transtypekey = 5
     AND d.CalYear = 2021
   GROUP BY f.custvendorkey, D.DateKey;

-- PostgreSQL solution
CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2021 AS
  SELECT f.custvendorkey, D.DateKey,
         SUM(quantity) AS SumQty, SUM(extcost) AS ExtCost,
         COUNT(*) AS NumTrans
   FROM date_dim d, inventory_fact f
   WHERE f.datekey = d.datekey
     AND f.transtypekey = 5
     AND d.CalYear = 2021
   GROUP BY f.custvendorkey, D.DateKey;
```

2. CREATE MATERIALIZED VIEW Statement for 2022 Shipments

```
-- Oracle solution
CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2022 AS
  PCTFREE 0
  BUILD IMMEDIATE AS
  SELECT f.custvendorkey, D.DateKey,
         SUM(quantity) AS SumQty, SUM(extcost) AS ExtCost,
         COUNT(*) AS NumTrans
   FROM date_dim d, inventory_fact f
   WHERE f.datekey = d.datekey
     AND f.transtypekey = 5
     AND d.CalYear = 2022
   GROUP BY f.custvendorkey, D.DateKey;

-- PostgreSQL solution
CREATE MATERIALIZED VIEW SalesByVendorDateKeyMV2022 AS
  SELECT f.custvendorkey, D.DateKey,
         SUM(quantity) AS SumQty, SUM(extcost) AS ExtCost,
         COUNT(*) AS NumTrans
   FROM date_dim d, inventory_fact f
   WHERE f.datekey = d.datekey
     AND f.transtypekey = 5
     AND d.CalYear = 2022
   GROUP BY f.custvendorkey, D.DateKey;
```

```
-- Optional drop statements
DROP MATERIALIZED VIEW SalesByVendorDateKeyMV2021;
DROP MATERIALIZED VIEW SalesByVendorDateKeyMV2022;
```

3. Rewritten Query 1 using SalesByVendorDayMV2021

```
-- CUBE solution
SELECT CalMonth, AddrCatCode1, SUM(ExtCost) as tot_cost,
       SUM(sumqty) as tot_qty
FROM SalesByVendorDateKeyMV2021 MV, cust_vendor_dim c,
     date_dim d
WHERE MV.CustVendorKey = c.CustVendorKey
  AND d.DateKey = MV.DateKey
GROUP BY CUBE(AddrCatCode1, d.calmonth);

-- GROUPING SETS solution
SELECT CalMonth, AddrCatCode1, SUM(ExtCost) as tot_cost,
       SUM(sumqty) as tot_qty
FROM SalesByVendorDateKeyMV2021 MV, cust_vendor_dim c,
     date_dim d
WHERE MV.CustVendorKey = c.CustVendorKey
  AND d.DateKey = MV.DateKey
group by GROUPING SETS((AddrCatCode1, d.calmonth), AddrCatCode1, d.calmonth,
());
```

4. Rewritten Query 2 using both MVs

```
-- This solution can be done with a view or using a UNION operation in
-- the FROM clause. The CUBE should occur after the UNION
-- operation.

-- Solution using UNION operation in the FROM clause
-- UNION operation in the FROM clause Solution 1
SELECT Name, Zip, CalQuarter, SUM(ExtCost) AS TotCost,
       SUM(SumQty) AS TotQty
FROM date_dim, cust_vendor_dim,
      (SELECT * FROM SalesByVendorDateKeyMV2021
      UNION
      SELECT * FROM SalesByVendorDateKeyMV2022) MV
WHERE MV.datekey = date_dim.datekey AND
      MV.custvendorkey = cust_vendor_dim.custvendorkey
GROUP BY CUBE(name, zip, calquarter);

-- UNION operation in the FROM clause Solution 2
-- Note the PostgreSQL requires the alias (MVX) for the subquery
SELECT Zip, CalQuarter, Name, SUM(ExtCost), SUM(SumQty)
FROM
(
SELECT Zip, CalQuarter, Name, ExtCost, SumQty
FROM SalesByVendorDateKeyMV2021 MV1, Date_Dim, Cust_Vendor_Dim
WHERE Date_Dim.DateKey = MV1.DateKey  AND
      Cust_Vendor_Dim.CustVendorKey = MV1.CustVendorKey
UNION
SELECT Zip, CalQuarter, Name, ExtCost, SumQty
FROM SalesByVendorDateKeyMV2022 MV2, Date_Dim, Cust_Vendor_Dim
```

```
WHERE Date_Dim.DateKey = MV2.DateKey AND
      Cust_Vendor_Dim.CustVendorKey = MV2.CustVendorKey
) MVX
GROUP BY CUBE (Zip, CalQuarter, Name);

-- GROUPING SETS solution with UNION in FROM clause
SELECT Name, Zip, CalQuarter, SUM(ExtCost) AS TotCost,
       SUM(SumQty) AS TotQty
FROM date_dim, cust_vendor_dim,
      (SELECT * FROM SalesByVendorDateKeyMV2021
      UNION
      SELECT * FROM SalesByVendorDateKeyMV2022) MV
WHERE MV.datekey = date_dim.datekey AND
      MV.custvendorkey = cust_vendor_dim.custvendorkey
GROUP BY GROUPING SETS((Name, Zip, CalQuarter), (Name, Zip),
                       (Name, CalQuarter), (Zip, CalQuarter), Name,
                        Zip, CalQuarter, ());
```