### 1. Star Schema Design:

#### **Fact Table:**

* **RevenueFacts**
  + TransactionID (Unique identifier for each sales transaction)
  + MemberID (reference to the Member dimension table)
  + Date (from SalesTransactions)
  + ItemID (could be a reference to the Merchandise table)
  + Quantity (from SoldVia)
  + TotalPrice (Quantity \* ItemPrice/MembershipPrice/PassPrice as applicable)

#### **Dimension Tables:**

* **MemberDim**
  + MemberID (from Member)
  + MemberName
  + MembershipType (from Membership table)
  + MembershipPrice (from Membership table)
* **MerchandiseDim**
  + ItemID (from Merchandise)
  + ItemName
  + ItemPrice
* **OneDayPassDim**
  + PassID (combined from OneDayGuestPass and OneDayPassCategory)
  + PassDate (from OneDayGuestPass)
  + PassCategory (from OneDayPassCategory)
  + PassPrice (from OneDayPassCategory)

### 2. Queries to Address the Tasks:

-- Merchandise Revenue

SELECT 'Merchandise' as Category, SUM(rf.TotalPrice) as MerchandiseRevenue

FROM RevenueFacts rf

INNER JOIN MerchandiseDim md ON rf.ItemID = md.ItemID

WHERE md.ItemID IS NOT NULL;

-- Membership Revenue

SELECT 'Membership' as Category, SUM(m.MembershipPrice) as MembershipRevenue

FROM MemberDim m

INNER JOIN RevenueFacts rf ON m.MemberID = rf.MemberID

WHERE m.MembershipType IS NOT NULL;

-- One Day Pass Revenue

SELECT 'One Day Pass' as Category, SUM(odp.PassPrice) as OneDayPassRevenue

FROM OneDayPassDim odp

INNER JOIN RevenueFacts rf ON odp.PassID = rf.ItemID

WHERE odp.PassID IS NOT NULL;

#### **b. Total Revenue:**

**SELECT SUM(TotalPrice) as TotalRevenue**

**FROM RevenueFacts;**

### 3. Conceptual Model Design:

* The conceptual schema will focus on a main fact table (RevenueFacts) that collects data on all revenue transactions. Surrounding dimensions, such as MemberDim, MerchandiseDim, and OneDayPassDim, will provide additional detail about the entries in the fact table.
* This star design allows for multidimensional queries and analysis of revenue. The dimensions provide context for the revenue data collected in the fact table.

### 4. Implementation in SQL Server:

* Once you have the star design ready, you can begin implementing it in SQL Server. This would involve creating the tables based on the design, followed by inserting the provided sample data.
* Once the tables are in place and populated with data, you could begin writing and executing queries to meet the specific requirements mentioned in the tasks.

This design and associated consultations are geared toward meeting the specific requirements mentioned above. It's important to remember that this is a high-level design and may require adjustments based on additional requirements or specific business details.

**Here is the SQL needed to create the tables based on the star schema design I provided, and to insert the sample values ​​you provided:**

**-- Creating Member Dimension Table**

**CREATE TABLE MemberDim (**

**MemberID INT PRIMARY KEY,**

**MemberName VARCHAR(255),**

**MembershipType VARCHAR(255),**

**MembershipPrice DECIMAL(10, 2)**

**);**

**INSERT INTO MemberDim (MemberID, MemberName, MembershipType, MembershipPrice)**

**VALUES (111, 'May', 'Platinum', 100.00),**

**(222, 'Joe', 'Gold', 300.00),**

**(333, 'Sue', NULL, NULL);**

**-- Creating Merchandise Dimension Table**

**CREATE TABLE MerchandiseDim (**

**ItemID VARCHAR(255) PRIMARY KEY,**

**ItemName VARCHAR(255),**

**ItemPrice DECIMAL(10, 2)**

**);**

**INSERT INTO MerchandiseDim (ItemID, ItemName, ItemPrice)**

**VALUES ('AP1', 'Tshirt', 11.00),**

**('AP2', 'Hat', 9.00),**

**('E01', 'PumpRope', 12.00);**

**-- Creating One Day Pass Dimension Table**

**CREATE TABLE OneDayPassDim (**

**PassID INT PRIMARY KEY,**

**PassDate DATE,**

**PassCategory VARCHAR(255),**

**PassPrice DECIMAL(10, 2)**

**);**

**INSERT INTO OneDayPassDim (PassID, PassDate, PassCategory, PassPrice)**

**VALUES (1, '2004-01-01', 'Adult', 20.00),**

**(2, '2004-01-02', 'Kid', 3.00),**

**(3, '2004-01-02', 'Senior', 10.00);**

**-- Creating Revenue Facts Table**

**CREATE TABLE RevenueFacts (**

**TransactionID INT PRIMARY KEY IDENTITY(1, 1),**

**MemberID INT,**

**Date DATE,**

**ItemID VARCHAR(255),**

**Quantity INT,**

**TotalPrice DECIMAL(10, 2)**

**);**

**-- Example Insert Statement for RevenueFacts (you might want to fill this in with actual data):**

**-- INSERT INTO RevenueFacts (MemberID, Date, ItemID, Quantity, TotalPrice) VALUES (...);**

Note: The RevenueFacts table has an automatic identifier (IDENTITY(1, 1)) for the TransactionID column, meaning it will be automatically generated when new records are inserted.

You can adapt this SQL to your needs. Once the tables are created and populated, you can proceed to run the queries required for your analyses or adjust the tables as needed.

I'll use the given information to generate the inserts for the RevenueFacts table. Since in the star design, the RevenueFacts table must contain transaction information, I'll take the data from the SalesTransactions and SoldVia tables to generate the inserts:

**-- Inserting data into RevenueFacts Table**

**INSERT INTO RevenueFacts (MemberID, Date, ItemID, Quantity, TotalPrice)**

**VALUES**

**(111, '2004-01-01', 'AP1', 1, 11.00), -- Calculated by multiplying Quantity (1) with AP1 price (11.00)**

**(222, '2004-01-02', 'AP2', 1, 9.00), -- Calculated by multiplying Quantity (1) with AP2 price (9.00)**

**(333, '2004-01-02', 'E01', 3, 36.00); -- Calculated by multiplying Quantity (3) with E01 price (12.00)**

**-- Note: The TotalPrice column values are computed based on the Quantity and respective product's price.**

**-- You might want to implement this calculation dynamically in your application or database triggers.**

The TotalPrice value in the insert was calculated based on the number of items and their price (taken from the MerchandiseDim table). You can make this calculation dynamic in your application or even at the database level using stored procedures or triggers.