## **Your Planned Setup**

1. **Existing Power BI Model (Referenced via DQFM)**
   1. **Dimension Tables:** **Dual Mode** (Can act as either Import or DirectQuery)
   2. **Aggregation Table:** **Import Mode** (Fully cached in memory)
2. **New Power BI Model**
   1. Uses **DirectQuery for Another Model (DQFM)** to reference dimensions & aggregations.
   2. Contains a **new Detail Table in DirectQuery mode** pointing to **Databricks**.

## **🛠 How Query Execution and Folding Work in This Setup**

### **Scenario 1: Querying Dimensions Only (From DQFM)**

* Since dimension tables are **Dual**, Power BI will decide whether to use **Import (cached)** or **DirectQuery** depending on the query context.
* If the **query does not include the DirectQuery detail table**, it will **run from cache (Import Mode)** for fast performance.
* If the **query includes a DirectQuery table**, the dimensions will behave as **DirectQuery** and push filters to the source.

**Folding:** If used in DirectQuery mode, filters on dimensions should fold back to Databricks.

### **Scenario 2: Querying Aggregations Only (From DQFM)**

* Since the **aggregated table is Import Mode**, all calculations on it will run in memory.
* If the user only queries **pre-aggregated data**, no DirectQuery calls will be triggered.

**Folding:** No need, since this runs in memory.

### **Scenario 3: Querying Detail Table + Dimensions**

* If a report uses both the **new detail table (DirectQuery from Databricks)** and **dimensions (from DQFM in Dual mode)**:
  + Power BI sends a **DirectQuery request to Databricks** for the detail table.
  + If a dimension is used **with a filter**, Power BI tries to **push the filter to Databricks** (folding it).
  + If the filter comes from an **Import table**, it must be applied in memory.

**Folding:**

* If the dimension is used in DirectQuery mode, **filtering will fold back to Databricks**.
* If the dimension is used in Import mode, **filtering happens in Power BI memory** (which can be less efficient).

### **Scenario 4: Querying Aggregations + Detail Table**

* If a report **combines the aggregated table (Import Mode) with the detail table (DirectQuery)**:
  + Power BI will **run two separate queries**:
    - One **in memory for the aggregation**.
    - One **against Databricks for the detail table**.
  + The results are **joined in Power BI memory**.

**Folding Issue:** Since Import and DirectQuery data cannot be joined in SQL, **Power BI does the join in memory**, which can be slow.

🔹 **Workaround:**

* Try to **avoid cross-source joins** in Power BI.
* If necessary, **use pre-aggregations in Databricks** instead of aggregating in Power BI.

### **Scenario 5: Querying Aggregations + Detail Table Using Dimension Filters**

1. If a user **filters a report using a dimension (Dual mode)**:
   1. If the query **only involves Import tables**, the filter is applied in memory.
   2. If the query **includes the DirectQuery detail table**, Power BI tries to **push the filter to Databricks**.

**Folding:** If the filter is applied at the database level (not in Power BI memory), it will fold.

## **Optimizing Performance**

To ensure the best performance:

1. **Ensure dimensions used in DirectQuery mode** are pushed to Databricks instead of being applied in memory.
2. **Avoid joining Import and DirectQuery tables in Power BI** when possible.
3. **Use pre-aggregations in Databricks** for queries that need both aggregated and detail data.
4. **Test SQL folding** by using Power Query’s "View Native Query" feature.