

Continue



When working with Power BI, many users encounter the challenge of managing multiple fact tables in their data models. These tables often contain additive values like sales amounts or enrollment counts but may operate at different granularities. Understanding how to effectively manage these tables is crucial for accurate reporting and efficient data processing. Understanding the Challenges Upon connecting to a data warehouse or building a model from another source, you might find yourself faced with several tables that house important metrics. The primary concern is how to handle these tables effectively without complicating your model or impacting refresh times. Common Pitfalls Before diving into solutions, let's discuss two common mistakes I see when dealing with multiple fact tables: Duplicating Dimensions: Many users create separate dimension tables for each fact table. For instance, if you have reseller and internet sales, you might end up with two calendar tables and two product tables. This duplication not only increases the size of your model but can also slow down refresh times. Creating Consolidated Fact Tables: Another approach is to stack fact tables on top of each other, creating a consolidated fact table. While this may seem convenient, it often leads to blank values in reports due to mismatched keys across fact tables. Avoiding Common Mistakes To avoid these pitfalls, it's essential to understand the implications of duplicating dimensions and consolidating fact tables. Why Duplicating Dimensions is Problematic When you duplicate dimensions, you increase the size of your model unnecessarily. This can lead to longer refresh times and complications in reporting. For example, if you try to create a report that includes both internet and reseller sales, you might find yourself confused about which calendar table to use. Problems with Consolidated Fact Tables When users choose to consolidate fact tables by appending them, they often encounter blank values due to unmatched keys. This can be frustrating for report consumers, who may see unexpected blank values in their reports. Additionally, such a large table can take longer to refresh and complicate measure calculations. The Right Approach So, what should you do instead? The answer lies in effectively utilizing the schema provided by your data warehouse and employing a star schema design. Here's how: Use Shared Dimensions Instead of duplicating dimensions, use a single set of shared dimensions for all fact tables. For example, create a centralized date table that relates to both internet sales and reseller sales. This way, you can filter and aggregate data across both tables seamlessly. Conformed Dimensions Utilize conformed dimensions that can be shared across different fact tables. This ensures that all fact tables use the same definitions and relationships, simplifying your model and enhancing usability. Implementing the Star Schema The star schema is a powerful design for organizing your data model. It consists of a central fact table connected to dimension tables, forming a star-like structure. Here's how to implement it: Identify Shared Dimensions: Identify dimensions like time, product, and location that can be shared across your fact tables. Create Relationships: Establish one-to-many relationships between the shared dimensions and fact tables. For instance, your product table should relate to both internet sales and reseller sales. Designate Specific Relationships: For dimensions that are specific to one fact table, like employee details for reseller sales, keep those relationships intact. Enhancing User Experience To improve the experience for report authors and consumers, consider adding descriptions to your tables. This way, users will know which dimensions to use with which fact tables, reducing confusion. Conclusion In conclusion, managing multiple fact tables in Power BI doesn't have to be a daunting task. By avoiding common pitfalls like duplicating dimensions and consolidating fact tables, and by employing a star schema design, you can create a more efficient and user-friendly data model. This approach allows for seamless reporting and analysis, ensuring that your data delivers the insights your business needs. Have you encountered challenges with multiple fact tables in Power BI? Share your experiences in the comments below! For those looking to enhance their Power BI skills, consider exploring expert-led training to further your understanding of data modeling and reporting. Boost your data skills with expert-led Power BI training. We have partnered with Microsoft to bring high quality Power BI training. Skip to main content Reddit and its partners use cookies and similar technologies to provide you with a better experience. By accepting all cookies, you agree to our use of cookies to deliver and maintain our services and site, improve the quality of Reddit, personalize Reddit content and advertising, and measure the effectiveness of advertising. By rejecting non-essential cookies, Reddit may still use certain cookies to ensure the proper functionality of our platform. For more information, please see our Cookie Notice and our Privacy Policy. A BI application must never issue SQL that joins two fact tables together across the fact table's foreign keys. It is impossible to control the cardinality of the answer set of such a join in a relational database, and incorrect results will be returned to the BI tool. For instance, if two fact tables contain customer's product shipments and returns, these two fact tables must not be joined directly across the customer and product foreign keys. Instead, the technique of drilling across two fact tables should be used, where the answer sets from shipments and returns are separately created, and the results sort-merged on the common row header attribute values to produce the correct result. Scenario 1: Multiple fact tables thar are not related to each otherSolution: Relate every dimension with their respective fact tables.Identify the shared dimensions.Relate them with every fact.Sometimes this model is called constellation schema, where each fact table individually is a perfect star schema design because each one is related with a dimension.AdvantagesStar schemas are easy to read, understand and use.Best performance.You can filter both fact tables using the shared dimensions.Disadvantages:As the fact tables are not related to each other if you try to add columns from both tables in the same visual that do not have a shared dimension you will get the following error:Bad Practices:Duplicate dimensions in the model:You will increase the size of your model which could impact the refresh time.You can't use the same dimension to filter the data of both tables.Consolidate the fact tables: Join fact tables through a union (append in Power Query) will create a huge table that can take a long time to refresh, in addition to a lot of blank values in the unmatched columns. Scenario 2: Multiple fact tables related to each other with different granularitiesRemember that the granularity represents the level of detail in each row of your fact table. In the header/detail models you must be careful to use the correct granularity in your calculations and try to maintain a good performance considering that they are not a perfect star schema.An example of header/detail models is when you have an order (header) and order details. The order table has an order number, date, a customer, and much more useful information at the order level. The detail table contains the product, the price, the quantity, and detailed information related to the individual order line.Solution 1: Flatten the header table into the detail table: With this solution you will get a denormalized model, where you locate the data from the header table to the detail table increasing the granularity.Advantages:You will get a perfect star schema that offers all its advantages.The values will get the correct granularity.The model will be easier to use because it can be summed and sliced by any dimension.Disadvantages:Increase the data model size.Categorical values repeated in the denormalized model.Solution 2: Relate the header and detail tableBe careful with this solution because you could get incorrect results due to the granularity differences between both tables. You can use it in the following scenario:You need to use columns from the different fact tables in a same visual table.The header detail table had a big number of column and flattening it into the detail table would create a huge table.You need to make few calculations over the header.The reports just needed to be sliced using common dimensions.Considering these points, apply the following solution:Identify the shared dimensions.Relate them with every fact to get a star schema model for every fact.Relate the header and detail table through an inactive relationship:Use a Many to one Cardinality and a "Both" cross filter direction:Finally, to summarize or bring the columns from the header table, use the USERELATIONSHIP DAX formula:Contract Amount = CALCULATE (SUM ('Fact Header'[Contract Amount]), USERELATIONSHIP ('Fact Detail'[WIP PK], 'Fact Header'[PK]))In this solution, linking the header and the detail table breaks the rules of the star schema because both tables act as a dimension and a fact table at the same time.It's a dimension when you need to slice the detail and it's a fact table when you summarize values at the header granularity.So, you will get a snowflake when the header table acts as dimension, although you should avoid it, sometimes as in this specific scenario could be a good solution to representing the data.Advantages:Avoid increase the size the data model having fact tables separated.You can use the star schemas to create reports that don't need to include in a same visual, columns from different fact tablesDisadvantages:This solution only works if you slice the data using the common dimensions, because otherwise you will get incorrect results due to the granularity differences between both tables.Possible performance degradation. Data AnalyticsPower BILearning Selection