DSI310: Data Exploration and Preprocessing - Chapter 6: Dimensional Modeling - Fact Tables

1. The Engine of Analysis: An Introduction to Fact Tables

Welcome back to DSI310. For the past two weeks, we have been building a strong foundation for our data warehouse. We began by differentiating between OLTP (Online Transactional Processing) and OLAP (Online Analytical Processing) systems, establishing that our goal is to transform data from the former to serve the needs of the latter.[1] Last week, we delved into the design of dimension tables, the descriptive "who, what, where, when, and how" of our business.[2] We learned how to handle disparate data sources and manage changes over time using powerful techniques like Slowly Changing Dimensions (SCD) to ensure our data warehouse remains a single source of truth for historical analysis.[2]

This week, we complete our exploration of the star schema by focusing on its central and most voluminous component: the **fact table**. If dimension tables provide the context, the fact table provides the quantitative measures—the "what happened"—that power all of our business intelligence and analytical reporting. A dimension table on its own is a list of attributes, but a fact table, linked to those dimensions, is the engine that drives business insights. It allows us to aggregate, compare, and analyze the core metrics of our business, such as sales, production yields, or website clicks.[2]

The central challenge of the OmniCorp project is to unify the data from the Chinook and Northwind databases into a cohesive data warehouse.[3] Our work this week will culminate in the design of the

FactSales table, which will serve as the single, authoritative source for all sales-related analysis across both business units. This design is not merely a technical exercise; it is a strategic decision that will determine our ability to answer critical business questions, such as "What is our total revenue across all business units?" or "Which products are our top sellers?" [3] A well-designed fact table is the key that unlocks the full analytical potential of our data.

2. Anatomy of a Fact Table: Measures and Foreign Keys

The fact table is the primary table in a dimensional model. Its structure is simple, yet its purpose is profound: to record business events and provide the quantitative data needed for analysis.[2] A fact table is characterized by two types of columns:

measures and **foreign keys**.[2]

- **Measures (The "What"):** Measures are the numeric values that represent the quantitative metrics of a business process.[2] They are the numbers that we want to aggregate, sum, average, count, and analyze.[2] For our
  FactSales table, the key measures are SalesQuantity (the number of items sold) and SalesAmount (the total monetary value for that line item).[2] These values are at the heart of our business analysis.
- **Foreign Keys (The "Context"):** These are the columns that link the fact table to the surrounding dimension tables.[2] A foreign key in the fact table corresponds to the primary key of a dimension table, creating the logical join that provides all the descriptive context for each business event.[2] For example, a single row in our
  FactSales table will have a foreign key that links to the DimCustomer table, allowing us to see which customer was involved in that specific transaction.[2]

A fact table typically does not contain any descriptive information; its primary purpose is to hold the measures and the keys that connect those measures to the dimensions. This design ensures that the fact table remains as compact as possible, which is crucial for maximizing query performance on massive datasets.[2]

3. The Central Design Principle: Grain and Granularity

The single most important decision in designing a fact table is determining its **grain**, or its level of granularity.[3] The grain defines exactly what a single row in the fact table represents. A well-defined grain is critical because it dictates the types of business questions the data model can and cannot answer.

For our FactSales table, the chosen grain is **one row per line item per invoice or order**.[3] This means that if a customer purchased five different products in a single transaction, that transaction would be represented by five separate rows in the

FactSales table.

Let's illustrate why this decision is so important by considering the consequences of a common mistake: modeling the grain at the invoice or order level.[3] If our

FactSales table had one row per invoice, it would contain the total sales amount for that invoice, but we would lose the detail for each individual product purchased within that transaction.[3] We would be unable to answer a critical business question like "Which products (tracks/items) are our top sellers?" because the fact table would not contain the product-level data needed for this analysis.[3] Our choice of a line-item grain ensures that we can drill down into the details of every sale, enabling granular analysis by product, category, or genre.[3]

A single row in our FactSales table, based on the line-item grain, expresses a specific business event in a business-friendly sentence:

> *On **January 15th, 2024**, customer **Bob Johnson** purchased **10 units** of **Chai Tea** for a total of **$180** from the **Northwind** business, with the transaction handled by employee **Nancy Davolio**.* [3]

This sentence clearly demonstrates the power of the star schema, where the fact table's measures (10 units, $180) are directly linked to the context provided by five different dimensions (DimTime, DimCustomer, DimProduct, DimSourceSystem, and DimEmployee).

4. Types of Fact Tables: A Broader Perspective

While our FactSales table is a **transactional fact table**, it is important to understand that other types of fact tables exist, each designed to address a different type of business event or analytical need.[2] The course materials introduce several types of fact tables as a part of a comprehensive data warehousing toolkit.[2]

- **Transactional Fact Tables:** This is the most common and fundamental type of fact table.[2] Each row in a transactional fact table represents a single event or a business transaction at the most granular level. Our
  FactSales table is a perfect example: each row represents a single line item from an invoice or an order.[2] The measures in this type of table are typically
  additive, meaning they can be summed up across any dimension (e.g., you can sum SalesAmount by customer, by product, or by date).
- **Periodic Snapshot Fact Tables:** Unlike a transactional fact table that records a single event, a periodic snapshot table records the state of something at a regular, predefined interval (e.g., daily, weekly, or monthly).[2] These are often used to track changes over time that are not tied to a specific transaction. For example, a periodic snapshot fact table might record the number of active customers or the total inventory level at the end of

each month. The measures in these tables are often **semi-additive** (can be summed over some dimensions but not all) or **non-additive** (cannot be summed at all, e.g., a balance).
- **Accumulating Snapshot Fact Tables:** This type of fact table is designed to track the progression of a process with a clear beginning and end, such as an order fulfillment or a customer's journey from application to approval. Each row represents the entire lifecycle of the process, and the measures are updated as new milestones are reached.
- **Factless Fact Tables:** These tables contain no measures at all. Their purpose is to track events or relationships between dimensions that do not have a numeric value associated with them. For example, a factless fact table could be used to track the relationship between customers and the products they have viewed on a website, which can be useful for a "Top N Analysis" of website interest.
- **Aggregated Fact Tables (or Cubes):** These tables pre-summarize data to improve query performance.[2] For example, we could create an aggregated fact table that contains the total SalesAmount by DateKey and ProductID. This allows for lightning-fast queries for pre-defined reports, but at the cost of losing the granular, line-item detail from the original fact table.

For our OmniCorp project, the **Transactional Fact Table** is the correct choice, as it provides the most granular level of detail, which is essential for our business analysis objectives.

5. Designing for Cross-Business Analytics with a Unified Fact Table

The true power of our data model lies in its ability to support business intelligence across both the music and food businesses. Our FactSales table, designed at the line-item grain, enables this by linking to a set of **conformed dimensions** that unify the disparate data from Chinook and Northwind.[3] This design allows a business analyst to perform a wide range of analyses with simple, high-performance queries.

5.1 Enabling Top N Analysis

A common business question is to identify the top-performing entities.[3] Our schema is perfectly designed to answer this by using a combination of the

FactSales table and our dimension tables.

- **Top Selling Products:** To find the top-selling products, a business analyst would simply aggregate the SalesQuantity measure from the FactSales table and group the results by the ProductName attribute from the DimProduct dimension.[3]
- **Top Spending Customers:** Similarly, to find the top customers by total spend, the analyst would aggregate the SalesAmount measure and group by CustomerName from the DimCustomer dimension.[3] The assignment requires a query that can identify the top 5 customers from the music business (Chinook), which would simply involve an additional filter on the
  DimSourceSystem dimension to select only "Chinook" records before performing the aggregation.[3]

5.2 Cross-Business Comparison

One of the key deliverables of the OmniCorp project is the ability to compare performance between the two new business units.[3] Our schema handles this seamlessly through the

DimSourceSystem dimension, which we designed last week.

- The FactSales table contains a foreign key that links to the DimSourceSystem dimension. This allows a business analyst to aggregate measures from the FactSales table (e.g., SalesAmount) and GROUP BY the SourceSystemName attribute (e.g., 'Chinook', 'Northwind').[3] This single, simple query provides a direct comparison of total revenue generated by each business unit.

5.3 Employee Performance Analysis

Our schema is also designed to support human resources and performance-related analytics. By linking the FactSales table to the DimEmployee dimension, we can analyze sales performance from various perspectives. For example, we can slice and dice sales data by employee title, city, or even by the number of sales they made. This directly addresses the assignment's test case of answering the question, "What is the total sales amount for all 'Sales Support Agent' employees from Chinook?" [3]

6. The Final Step in the Data Pipeline: From Ingestion to Integration

Our journey so far has taken us from the raw, messy data in our source systems to a clean, structured analytical model. The fact table is the final destination for our transactional data. The process of building and populating this table is the final, crucial step in the ETL (Extract, Transform, Load) process.

The data for our FactSales table comes from two different source tables: Chinook.InvoiceLine and Northwind."Order Details". Populating our unified FactSales table involves a multi-step ETL process:

1. **Extract:** We start by extracting the raw transactional data from the two source systems into our data lake's **Landing Zone**, as we learned in Week 2.[2]
2. **Transform & Load:** In the **Staging Zone** and **Integration Zone**, we perform the complex transformations. This is where we will:
    - Read the raw data from the Landing Zone.
    - Join the transactional data to the cleaned and prepared dimension data (e.g., joining Chinook.InvoiceLine to the new DimProduct and DimCustomer tables we designed last week).
    - Derive the SalesAmount measure by multiplying UnitPrice and Quantity for each line item.[3]
    - Use the surrogate keys from our dimension tables as the foreign keys for our FactSales table.
    - Finally, load the completed FactSales table into our data warehouse, where it is ready for analysis.[2]

7. The Deeper Meaning: Facts as Assertions (A-Boxes)

To conclude, let's revisit the conceptual framework we introduced in the last lecture. The star schema is not just a technical design; it is a profound method for representing business knowledge. This concept is captured in the relationship between the T-Box and A-Box from the field of knowledge graphs and ontology.[4]

- **Dimension tables are the T-Boxes:** They define the permanent terminology, concepts, and hierarchies of our business domain.[4] Our DimCustomer table, for example, is our T-Box that defines what a customer *is* in our analytical world.
- **Fact tables are the A-Boxes:** They hold the specific, transactional, and dynamic **assertions** or "facts" about the business.[4] A single row in our FactSales table is an assertion of a specific event—a sales transaction—that happened at

a particular time.

The star schema, therefore, is a logically sound and robust way to bring together the permanent conceptual knowledge (T-Box) with the specific factual observations (A-Box), enabling us to perform powerful analysis and derive new knowledge from our data.

8. Key Theories and Keywords

- **Fact Table:** The central table in a dimensional model. It contains the quantitative measures of a business event and foreign keys that link it to the dimension tables. It is typically a very large and highly voluminous table.[2]
- **Measures:** The numeric quantities or metrics of a business process that can be aggregated or analyzed. Examples include SalesAmount and SalesQuantity.[2]
- **Foreign Key:** A column in a fact table that links to the primary key of a dimension table, providing the descriptive context for each business event.[2]
- **Grain:** The lowest level of detail represented in a fact table. It is the most important design decision in dimensional modeling, as it determines the specific level of granularity for analysis.[3] Our
  FactSales table is designed at the line-item grain.
- **Transactional Fact Table:** A fact table in which each row represents a single, individual transaction or business event. This type of table is ideal for high-volume, granular analysis.
- **Periodic Snapshot Fact Table:** A fact table that records the state of a business at a regular, predefined interval.
- **Top N Analysis:** An analytical query used to find the top N records based on a specific measure, such as finding the top 10 customers by sales amount.
- **Cross-Business Comparison:** A type of analysis that compares key performance indicators (KPIs) or measures between different business units or sources, often enabled by a SourceSystem dimension.
- **A-Box (Assertion Box):** A concept from knowledge graphs and ontology that represents the specific, dynamic, and transactional facts or events that occur within a domain. Our fact tables are the A-Boxes of our data model.[4]
- **T-Box (Terminological Box):** A concept from knowledge graphs and ontology that represents the permanent, conceptual schema and hierarchies of a domain. Our dimension tables are the T-Boxes of our data model.[4]

Works cited

1. OLTP vs. OLAP: Differences and Applications - Snowflake, accessed September 10, 2025,

https://www.snowflake.com/en/fundamentals/olap-vs-oltp-the-differences/

2. dsi310_แบบฟอร์มเค้าโครงการบรรยาย 2025
3. dsi310: Data Lake Modeling
4. 2025 dsi310 week02