Heehwan Soul

Mehmet Görkem Basar

Professor Patrick Erdelt

Datenbanksysteme II - 1.Project

11.06.2024

Project Report

Introduction

In this project we aimed to create a data warehouse, perform data quality checks, and analyze sales data using Tableau. This included designing a star schema, implementing ETL procedures, conducting data quality assessments, and visualizing data insights using Tableau. The implementation can be found in the `stud_soul` or `stud_basar` database.

1. Data Source

This part details the process of creating a comprehensive database view named `bestelldaten` from the `sose24_dbs_oltp` database. The view consolidates relevant information from multiple tables, ensuring efficient data retrieval and reduced query execution time.

Database Schema Overview

The `sose24_dbs_oltp` database comprises several tables interconnected through primary and foreign key relationships. The tables involved in this view creation are:

- `person`
- `kundengruppe`
- `bestellung`
- `verkaufskanal`
- `versandart`
- `bestellposition`
- `liefert`
- `artikel`
- `lieferant`
- `artikelgruppe`
- `hersteller`

Explanation of the View Creation

The SQL code to create the `bestelldaten` view is in the file `Datenbanksysteme2-Projekt-Skript.sql`. The view utilizes a series of `JOIN` operations to combine data from multiple tables. This ensures that all related data is brought together in a single, unified view. All fields from each table are selected and aliased for clarity.

Noteworthy Aspects

1. Utilization of Additional Connections: While initially, only one connection was apparent between the `bestellposition` and `liefert` tables, further exploration revealed an additional connection through individual columns in the tables on the ER Diagram. By leveraging both connections, the view ensures more efficient data retrieval. This approach reduces the total number of rows in the `bestelldaten` view, enhancing query performance and speeding up execution times. Consequently, any code utilizing the `bestelldaten` view benefits from improved efficiency.
2. Upon exploring the data, we discovered that rows with a `gesamtwert` (total value) of 0 were not appearing in the `bestelldaten` view. This discrepancy arose from orders placed by customers whose `kundengruppe` (customer group) value is 'NA', resulting in a `gesamtwert` of 0 for these orders. When joining all the tables, the row with the 'NA' value is not included, leading to the omission of orders with a `gesamtwert` of 0 from the `bestelldaten` view.

2. Data Quality Checks

Ensure the reliability, accuracy, and completeness of the data in the data warehouse. Data quality checks are crucial to ensure that the data used for analysis is accurate and reliable. High-quality data leads to better decision-making, operational efficiency, and trustworthy analytical results. Poor data quality can result in incorrect insights, flawed strategies, and wasted resources. We

conducted three distinct types of data quality check: functional dependency check, completeness check, and range check.

Functional Dependency Check

A functional dependency check assesses the relationship between attributes in a dataset, ensuring that one attribute's value uniquely determines another's. Two functional dependency checks were carried out:

1. **Between `bestellung_bestellnummer` and `bestellung_bestelldatum`**: As both attributes reside in the same table, with `bestellung_bestellnummer` serving as the primary key, a full functional dependency (1) was expected and observed.
2. **Between `bestellung_bestellnummer` and `verkaufskanal_name`**: With `bestellung_bestellnummer` being the primary key in the `bestellung` table and the `verkaufskanal` table being joined with `bestellung`, a full functional dependency (1) was anticipated and confirmed.

Completeness Check

A completeness check was conducted specifically for the field `Straße`. This involved examining each entry in the database to ensure that both a street name and a house number were present. To quantify this completeness, a metric was created, ranging from 0 to 1, representing the percentage of entries that contained both a street name and a house number.

This metric provides a measure of the completeness of the `Straße` field, indicating the proportion of entries that have sufficient address information. A value of 1 indicates that all

entries include both a street name and a house number, while a value closer to 0 suggests a higher incidence of missing or incomplete address data.

Noteworthy Aspects

The result of the completeness check for the `Straße` field was close to 1 but not exactly 1. Upon further examination of the data, it was observed that several entries contained the value "NA." This likely contributed to the slightly lower completeness score.

We used two different SQL queries to calculate the completeness ratio for the `Straße` field:

1. "The first query, labeled as 'Completeness Ratio version 1' in the file `Datenbanksysteme2-Projekt-Skript.sql`, counts entries that contain both a character and a number in the `person_strasse` field, assuming that a character represents the street name and a number represents the house number.
2. Similarly, the second query, identified as 'Completeness Ratio version 2' in the same file, counts entries where the `person_strasse` field is not NULL or blank. This query operates under the assumption that if an entry is filled, it contains both a street name and a house number."

We concluded that the first query is more robust because the second query does not account for cases where only the house number or only the street name exists. While our current dataset may not contain such entries, it's important to consider the possibility for future data uploads.

Notably, despite the different approaches, both queries produced similar results, suggesting that there are few, if any, entries with only a street name or only a house number.

Range Check

The range check examines the `artikel_preis` column for negative values. This validation ensures the accuracy and integrity of price data, guarding against errors in financial reporting and profitability analysis. A result of 0 signifies that all entries have non-negative prices, while a result closer to 1 indicates a higher frequency of negative prices within the dataset.

Noteworthy Aspects

The result of the range check for the `artikel_preis` column was 0, indicating that all prices were non-negative.

Automated Data Quality Monitoring

The objective is to automate the periodic execution of data quality checks and store the results in a dedicated table.

1. **Data Quality Table**: A table named "datenqualitaet" will be utilized to store the outcomes of data quality assessments.
2. **Stored Procedures**: The stored procedure "update_datenqualitaet" has been developed to automate the execution of data quality checks and subsequently insert the results into the "datenqualitaet" table on an hourly basis.

3. **Event Scheduler**: An Event Scheduler has been configured to execute the "update_datenqualitaet" stored procedure every hour, ensuring regular and automated monitoring of data quality over time.

3. ETL(Extract, Transform, Load) and Data Warehouse

The objective of this implementation is to create a data warehouse system that is updated hourly with current data, using a 5-dimensional analysis cube modeled in a star schema.

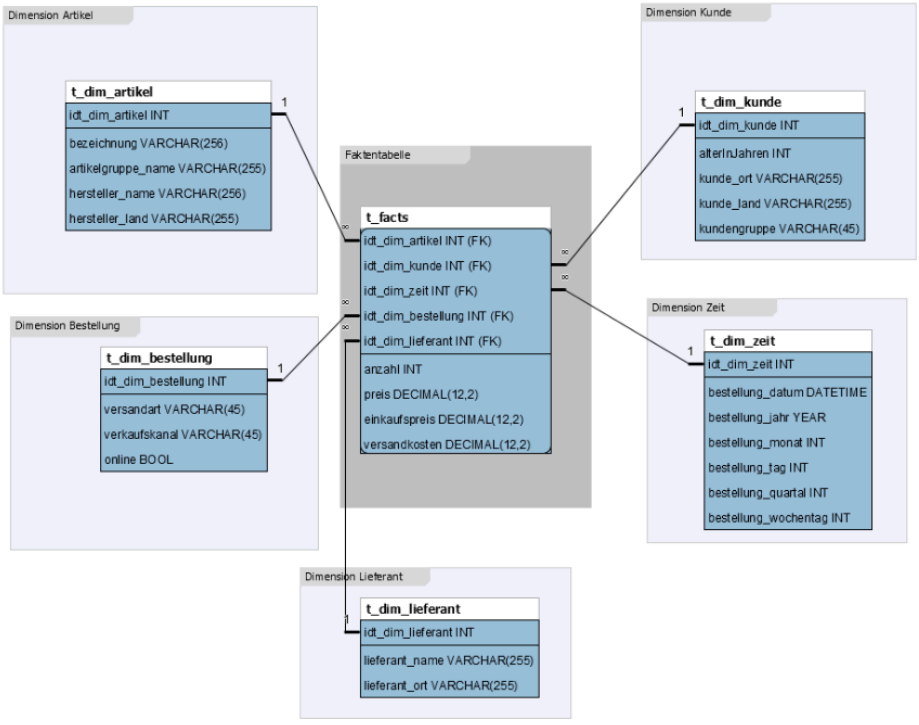


Abbildung 1: Data Warehouse

ETL Procedure

A stored procedure named `etl_update()` was developed to automate the Extract, Transform, Load (ETL) process. This procedure copies new data from the source (`bestelldaten` view) into the data warehouse tables. **The procedure handles exceptions using a transaction, ensuring data integrity by rolling back changes in case of errors during the insertion process.**

Noteworthy Aspects

1. WHERE bestelldaten.lieferant_id NOT IN (SELECT idt_dim_lieferant FROM t_dim_lieferant): this is WHERE clause of a code block to insert data into t_dim_lieferant from our view `bestelldaten`. If a `lieferant_id` value in `bestelldaten` does not exist in the `t_dim_lieferant` table, we include the information from that row to update the dimension table with new information. The same logic applies to all other WHERE clauses used to insert new data into the other dimension tables.

2. WHERE t_dim_bestellung.idt_dim_bestellung NOT IN (SELECT idt_dim_bestellung FROM t_facts): this is WHERE clause of a code block to insert data into `t_facts`. Although the primary key of `t_facts` comprises 5 fields, checking only the `idt_dim_bestellung` field is sufficient, as it determines all other 4 fields.

3. In the code for inserting data into `t_facts`, initially, we joined `bestelldaten` only with `t_dim_zeit`, as all the required information for the `t_facts` table was

contained within these two tables. However, we encountered an error due to foreign key constraints. Consequently, we opted to join all 5 dimensional tables.

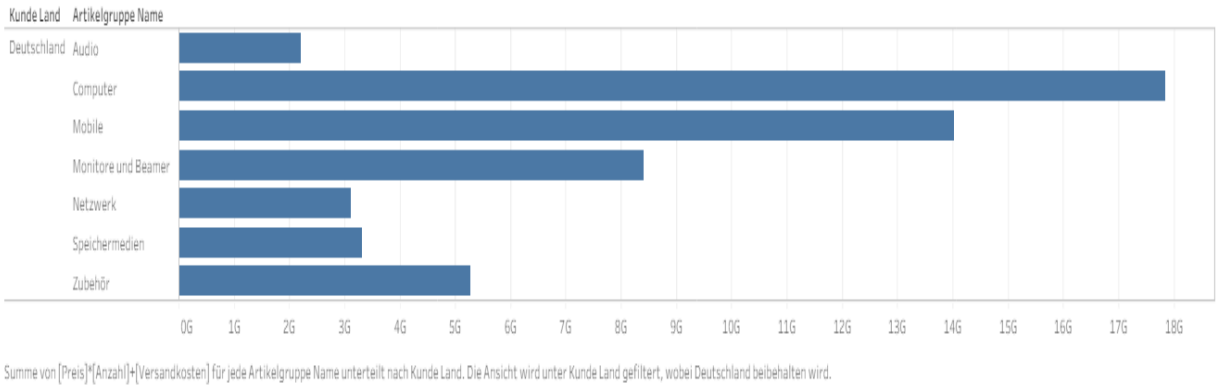
Event Scheduler

An Event Scheduler named `hourly_etl_event` was configured to execute the `etl_update()` procedure every hour. This ensures that the data warehouse is regularly updated with fresh data, maintaining its relevance and accuracy for analytical purposes.

4. Data Analysis Using Tableau

Here we have analyzed the sales data in two dimensions. We have done 2 analysis;

Total Sales by Product Category in Germany: A bar chart displaying total sales for different product categories in Germany. `Computer` and `Mobile` categories dominate the German market, while other categories show moderate sales. This can guide inventory and marketing strategies towards high-demand products.



Computer Sales Trends Over Time: A line chart showing the trend of computer sales prices from 2012 to 2022. The trend shows a significant rise in computer sales starting in 2015, with stable growth continuing through 2022. This indicates a strong and growing market for computers, useful for forecasting and strategic planning.