INTERNATIONAL UNIVERSITY OF APPLIED SCIENCES

DLMDSEBA01 Business Intelligence I

CASE STUDY

Sylt Fish Specialties

Author: Yashwant Sawant

Enrollment Number: 32210034

Study Program: Computer Science

Table of Contents

Introduction

The world is changing rapidly with data being everywhere. Every business, from small startups to large enterprises, generates huge amounts of data every day. But stored data alone is not useful, it needs to be collected, processed, and analyzed to extract meaningful information. This is where Business Intelligence (BI) plays an important role. BI helps organizations make sense of data, discover patterns, and helps in decision-making. Companies that effectively use BI can identify opportunities and gain a competitive edge over their competition.

Sylt Fish Specialties is one such company that relies on data to stay ahead in the fast-food industry. With 400 branches spread across Germany, they have built a strong reputation for offering fresh fish through one of the fastest inland logistics networks. This model ensures quality and freshness, setting them apart from competitors. Their success has placed them among the top five fast-food chains in the country. However, maintaining this position requires continuous improvement and growth. They must understand how their stores are performing, which products are driving the most sales, and how customer preferences are evolving.

To achieve this, Sylt Fish Specialties needs a structured approach to managing and analyzing data. Business Intelligence provides the tools and methods to extract valuable insights from data gathered from various components of Sylt Fish Specialties like sales division, customer division or logistics. By studying data from all these systems at single place, executives at Sylt Fish can make informed decisions about inventory management, product offerings, and marketing strategies. For example, understanding which products perform best in different locations can help optimize stock levels and reduce waste. Similarly, identifying peak sales periods can improve staffing and logistics planning.

To make this work, the company needs a solid data pipeline. First, they pull raw data from different operational systems—things like sales transactions, inventory logs, and customer databases. Next, they clean and harmonize the data, ensuring everything is accurate and standardized. Finally, this processed data is stored in a central Data Warehouse (DWH), designed for analysis and reporting. From here, managers and executives can generate reports, visualize trends, and spot opportunities for improvement.

By leveraging the full power of BI, Sylt Fish Specialties can not only maintain its position among the top five but also find new opportunities for growth and expansion.

Main Body

Operational Systems and Data generated from them

- Operational Systems

Operational systems are core of any business. They are part of the day-to-day activities of the business from ground level. From product manufacturing to product usage by end consumer every component or system involved in an operational system of that business. For Sylt Fish Specialties restaurants chain, following are the operational systems.

1. Customer management system: This operational system is responsible for handling the complete information about every customer of Sylt Fish Specialties. They maintain customer information like their name, address, birth date etc. This system is also responsible for maintaining subscription or membership information of the customers.
2. Inventory management system: For Sylt Fish Specialties, the inventory is all the items necessary to fulfil the customer orders. Inventory management system store information like how much stock of each type fish is available at each restaurant location, how much is available at central warehouse, how many desert available how many are about to expire etc.
3. Point of Sale (POS) System: Point of sales system is responsible for the all the business happening in the restaurant. Internally POS system relies on
4. Ordering System: Ordering system is responsible for order of the restaurants. It depends on the inventory system before creating order for customer.
5. Payment System: This system is responsible for all the payment transactions. This system stores the customer payment information and users use it for payments for the order.
6. Invoice System: Once payment is completed Invoice system generates the invoice for the order.

This is the list of the operational systems that we will using for creating this prototype BI application, but this list no means exhaustive list of operational systems involved in the restaurant business.

- Python environment and Faker package

Since we do not have access to the actual operational data of Sylt Fish Specialties and all the

operational systems have a particular function that they are responsible for like customer management system has responsibility of the maintaining the customer information like customer name, customer contact information, date of birth and address. Each operational system maintains its own database or data store where they store the information. These systems can perform operation related to their system locally by using their data, but the operational level decisions can only be taken when complete view of the entire business is taken by analyzing complete data.

For the data stored by operational system we will be using python program and Faker library. Faker is python package that generate fake or synthetic data based on the application requirement. For the Sylt Fish Specialties restaurants operational systems mentioned above, Faker will be used to generate the system specific data. Also, for some preprocessing another python package called Pandas will be used.

Initially, Faker and Pandas is installed on the python environment using 'PIP'.

```
yrsawant@yrsawant-mac ~ % source myenv/bin/activate  # On macOS/Linux

(myenv) yrsawant@yrsawant-mac ~ % pip install faker pandas

Collecting faker
  Downloading faker-37.1.0-py3-none-any.whl.metadata (15 kB)
Collecting pandas
  Using cached pandas-2.2.3-cp313-cp313-macosx_11_0_arm64.whl.metadata (89 kB)
Collecting tzdata (from faker)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting numpy>=1.26.0 (from pandas)
  Using cached numpy-2.2.4-cp313-cp313-macosx_14_0_arm64.whl.metadata (62 kB)
Collecting python-dateutil>=2.8.2 (from pandas)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
Collecting pytz>=2020.1 (from pandas)
  Using cached pytz-2025.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Using cached six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Downloading faker-37.1.0-py3-none-any.whl (1.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 30.6 MB/s eta 0:00:00
Using cached pandas-2.2.3-cp313-cp313-macosx_11_0_arm64.whl (11.3 MB)
Using cached numpy-2.2.4-cp313-cp313-macosx_14_0_arm64.whl (5.1 MB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached pytz-2025.1-py2.py3-none-any.whl (507 kB)
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: pytz, tzdata, six, numpy, python-dateutil, faker, pandas
Successfully installed faker-37.1.0 numpy-2.2.4 pandas-2.2.3 python-dateutil-2.9.0.post0 pytz-2025.1 six-1.17.0 tzdata-2025.2

[notice] A new release of pip is available: 25.0 -> 25.0.1
[notice] To update, run: pip install --upgrade pip
(myenv) yrsawant@yrsawant-mac ~ %
```

Figure 1: Setting python virtual environment and installing Faker

Using this python environment and Faker package, Customer data, inventory data, payment data, invoice data and sales data is generated. Faker can generate fake data for field like customer name, city name, company name etc. Other data like product names which are specific to fish and deserts we used the a list of hardcodes names of the products. Following code python snippet shows how operational system data is generated using Faker.

```python
import faker
from datetime import datetime, timedelta

fake = faker.Faker()

# Define products (15 fish-based, 5 desserts)
products = [ ...
]

# Generate Stores (400 stores)
stores = [{"store_id": i, "name": fake.company(), "city": fake.city()} for i in range(1, 401)]

# Generate Customers (1000 customers)
customers = [{"customer_id": i, "name": fake.name(), "zip_code": fake.zipcode()} for i in range(1, 1001)]
```

Figure 2: Python Code snippet to generate fake operational system data

Each order can have multiple products ordered

```
sales_orders = []
order_details = []
invoices = []
payments = []
inventory = {}

for order_id in range(1, 5001):
    store = random.choice(stores)
    customer = random.choice(customers)
    order_date = fake.date_time_this_year()  # Format: YYYY-MM-DD HH:MM:SS
    total_amount = 0
    products_in_order = random.sample(products, random.randint(1, 5))

    for product in products_in_order:
        product_id = products.index(product) + 1
        quantity = random.randint(1, 3)
        unit_price = product[2]
        total_price = quantity * unit_price
        total_amount += total_price
        order_details.append({"order_id": order_id, "product_id": product_id, "quantity": quantity, "unit_price": unit_price, "t
```

Figure 3: Python code snippet to store random data in list

- Generated Data:

The script ran successfully, creating a large dataset with realistic orders, customers, and transactions. Below is a snapshot of the synthetic data generated by the python script

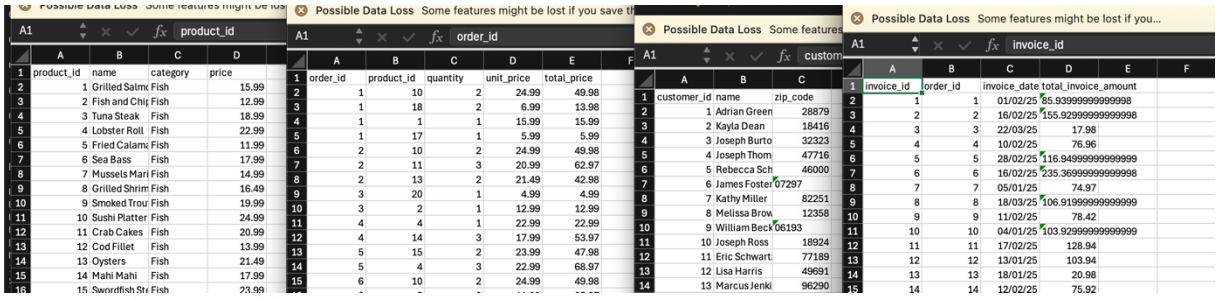

Figure 4: Faker generated data

The synthetic data generated, and python script used for the generation is available in the GitHub.

Design and implement a data model for the DWH.

- Definition and Architecture

Data warehouse (DWH) is defined as “A data warehouse is a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management’s decisions” (Inmon, 2005, p. 31). For Sylt Fish Specialties, the prototype BI application will use these principles to organize and analyze data effectively across its 400 branches.

1. Subject-oriented: As Sylt fish business is centered around sales, customers, payments and inventory, the Sylt fish DWH is created around these subjects. This helps when subject specifics analysis is to be done like bestselling product etc.

2. Integrated: Source of the data stored in DWH originated from various heterogenous systems like point of sales system, customer management system, payment system etc. Each system has its own way to store data but when data is added to DWH, data

is moved using ETL to standardize data. This allows holistic analysis regardless of origin is from which restaurant branch or which operational systems.

3. Nonvolatile: Unlike transactional databases where records are frequently updated or deleted, data in the Sylt Fish DWH is stored permanently. Once sales, customer, or inventory data is loaded into the warehouse, it remains unchanged.

4. Time-Variant: The DWH maintains historical snapshots of data. This will help in performing analysis like which product is popular in summer vs winter and based on that analysis better inventory management can be done.

For this prototype BI application, central data warehouse (C-DWH) architecture is used. Data collected from various operational systems is processed and stored at central data warehouse. This architecture ensures data consistency and simplified management. DWH is created using Oracle RBDMS database which ensures consistency through ACID transactions. Relational Storage (ROLAP) is data storage model used for DWH.

- Data Model for Sylt Fish Specialties

To support the analytics needs of Sylt Fish Specialties, DWH contains following tables

1. Few Dimensions table: DT_Products: Stores product related information DT_Payments: Store payment related information

2. Fact Table: Stores sales transactions, linking to all dimension tables, and containing metrics like quantity sold, location information revenue, total amount etc.

Following figure shows entity relation diagram of the C_DWH created for the Sylt Fish Specialties.

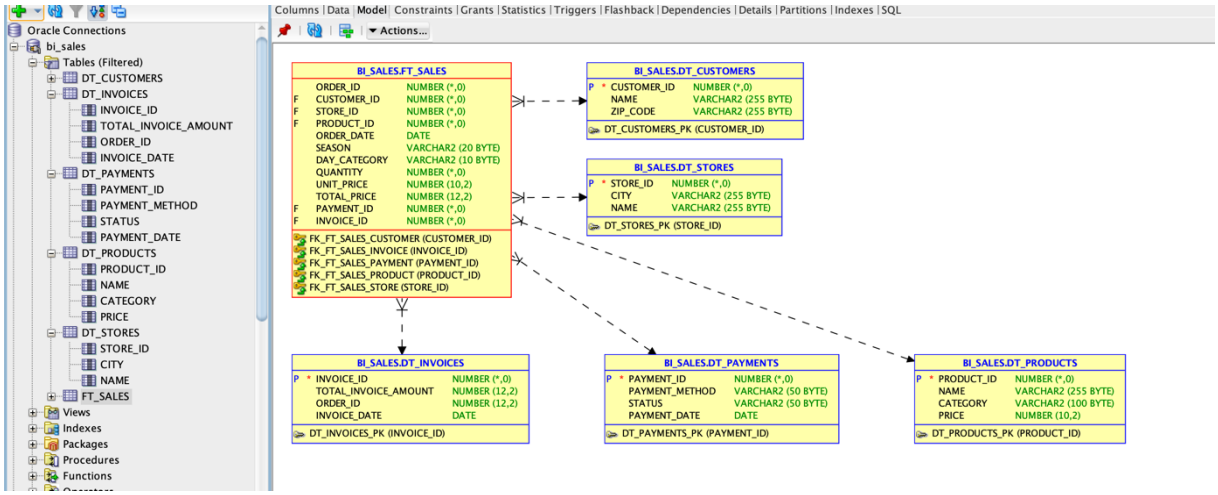

Figure 5:Sylt Fish Specialties C-DWH Entity Relationship Diagram

Load the test data into the DWH using a suitable tool

The generated synthetic test data in CSV format using Python Faker is loaded into Oracle-based Data Warehouse (DWH). This follows the standard Extract, Transform, Load (ETL) process, ensuring the data is properly structured before being stored in the database. To implement this, pETL package, a powerful Python library for ETL operations is used. The process consists of three key steps:

- Extract Data

The first step involves reading the generated CSV files. These files contain raw data for different operational systems of Sylt Fish Specialties. Data from the 5 operational systems is in the CSV format. pETL python package is used to load these CSV files for further processing.

```
# ---------------------- 1. Extract & Store Raw Data ----------------------
raw_data = {
    "customers": etl.fromcsv("customers.csv"),
    "products": etl.fromcsv("products.csv"),
    "stores": etl.fromcsv("stores.csv"),
    "inventory": etl.fromcsv("inventory.csv"),
    "payments": etl.fromcsv("payments.csv"),
    "invoices": etl.fromcsv("invoices.csv"),
    "order_details": etl.fromcsv("order_details.csv"),
    "sales_orders": etl.fromcsv("sales_orders.csv")
}
```

Figure 6: Python ETL - Extract data

- Transform Data

Transformation is the process of cleaning, modifying, and enriching raw data before loading it into the Data Warehouse (DWH). This ensures that the data is structured correctly for analysis, reporting, and decision-making.

First transformation is filtering in which data is freed from the defects in 2 steps of extracting and cleansing. During filtering syntactic and semantic defects will be cleaned up from the loaded data. This python in memory is acting as the staging area. After filtering data harmonization is done for reconciliation of data coming from multiple heterogenous systems. After this stage data from multiple systems will be normalized form. For Sylt Fish Specialties harmonization will unify dates, normalize text like payment statues and convert numeric payment fields to float values. Third and fourth transformation is aggregation and enrichment which will create new columns features which will help faster analysis.

Figure 7: Python ETL Transformation phase

- Loading Data

The transformed and enriched data will be stored in Centralized Data Warehouse (C-DWH) architecture, where Oracle DWH will store in dimension tables and fact tables. As Oracle is the base database, python cx_Oracle package is used to load the ETL data to oracle based Centralized Data Warehouse. Following snippet show INSERT statement used to load the processed data.

Figure 8: Loading data to DWH

Create five suitable analyses using appropriate BI tools.

To start with the analysis, Data was enriched and cleaned the sales data which includes details like product, store, customer, payment, and seasonal information. This cleaned dataset was then loaded into Oracle database This DWH is connected Tableau, pulled in the data in Tableau for further analysis, reporting and visualization.

The idea behind this analysis is to understand sales trends, product performance, and

customer behavior for Sylt Fish Specialties. By breaking down the data and visualizing it from different angles, Goal is to get actionable insights that can help improve business decisions. Below are the detailed analyses performed on the data stored in DWH.

Analysis 1 – Product-wise Sales Contribution:

The first analysis is focused on understanding how each product is performing in terms of sales. The idea is to see which products are bringing in the most revenue and which ones aren't selling much. This kind of analysis is important for any business, and for Sylt Fish Specialties, it can help management decide which dishes to promote, which ones to keep in stock, or even which ones to possibly remove from the menu.

For this, we created a Bar Chart in Tableau. On the Y-axis, we used the "Product id" so that we can clearly see each item. On the Y-axis, we used the "Total Sales Amount". This shows us how much money each product has earned overall.

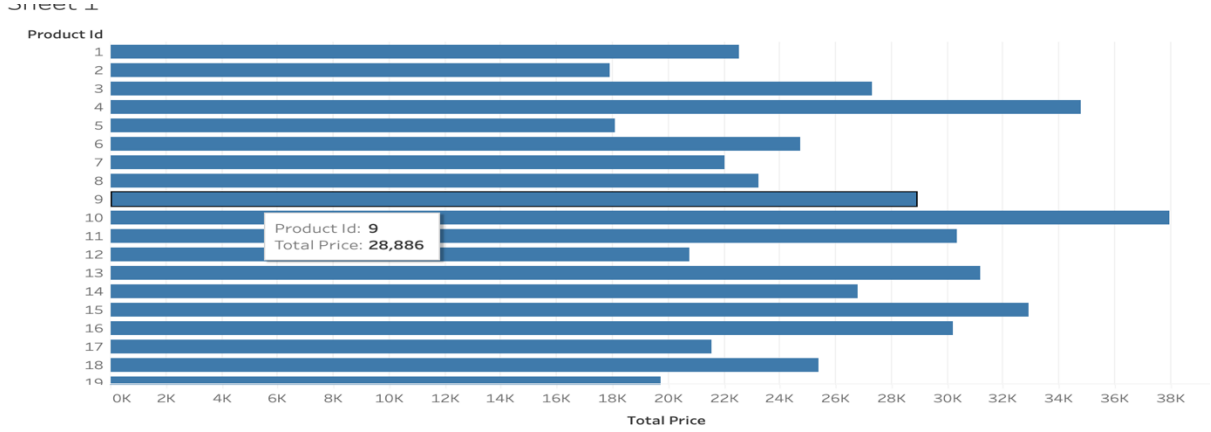

Figure 9: Report of each products contribution to Price

This chart gives a very clear picture. Just by looking at it, the team at Sylt Fish can quickly spot their bestsellers and their weaker products. This can help in managing inventory of best seller and at same can suggest offering some discounts to customer for products which are not selling very good

Analysis 2 – Seasonality Analysis of Product Sales

This analysis focuses understanding whether the sales of Sylt Fish Specialties' products are influenced by seasons. Reporting tools plots seasonal sales data to check if certain products are sold more during specific times of the year. The goal was to see if there is any clear seasonal pattern in customer buying behavior.

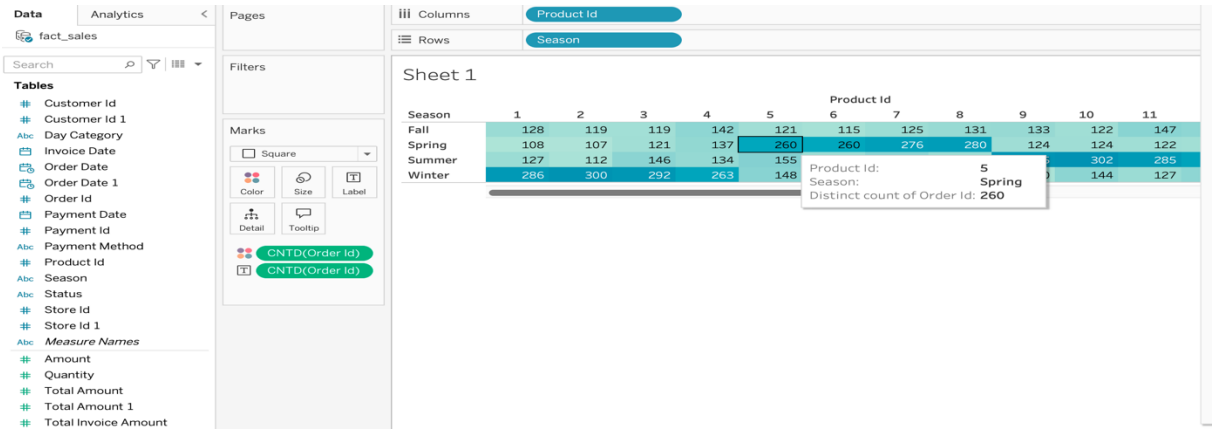

Figure 10: Seasonal distribution of products

Above reporting shows the seasonal distribution of products ordered by customers. This analysis shows that few of the products are very popular in summer while some are very popular in winter season for example product 1 is ordered by 286 customers time during summer but only 108 times in spring. This analysis can help Sylt Fish Specialties in many ways like It can help them plan their stock for seasonal products or They can run marketing offers during low-sales seasons.

Analysis 3 – Store-wise Sales Performance

This analysis shows which stores are performing well and which ones are not. Sylt Fish Specialties has 400 store locations, and it is important to compare each stores sales to identify the top-performing and low-performing stores. This helps the management to understanding strategies that are working well with top performers and can be applied to low performer to improve the overall revenue.

For this analysis, BI prototype application created a bar chart using Tableau. On the Y-axis, we placed the Store IDs. On the X-axis, we plotted the total sales amount made by each store. Each bar in the chart represents one store, and the length of the bar shows how much revenue that store has generated

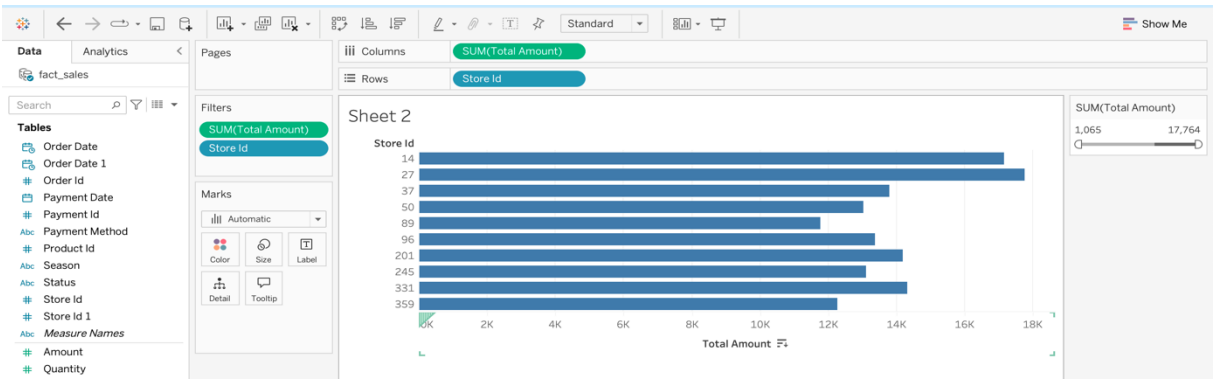

Figure 11: Top performers stores

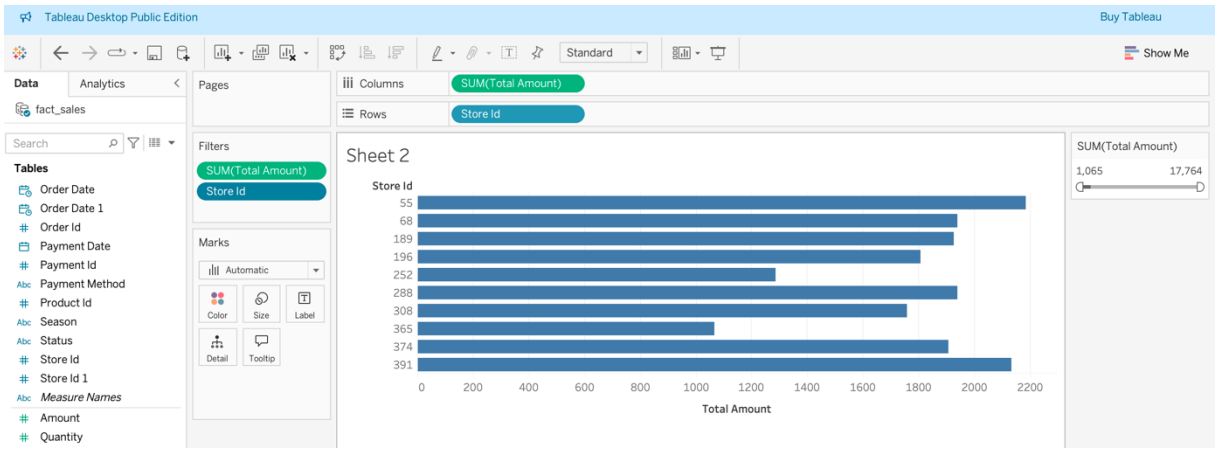

Figure 12: Bottom performers stores

Analysis 4 – Payment Method Preference Analysis

In this analysis, Prototype BI application tries to understand how customers of Sylt Fish Specialties prefer to pay for their orders. Payment behaviour is a crucial business insight because it can reveal customer preferences, payment method performance, and even highlight operational challenges.

For this, Application prepared a bar chart. On the X-axis, we placed the Payment Method (Credit Card, Cash, Mobile Payment). On the Y-axis, we plotted the Total Payment Amount received for each method.

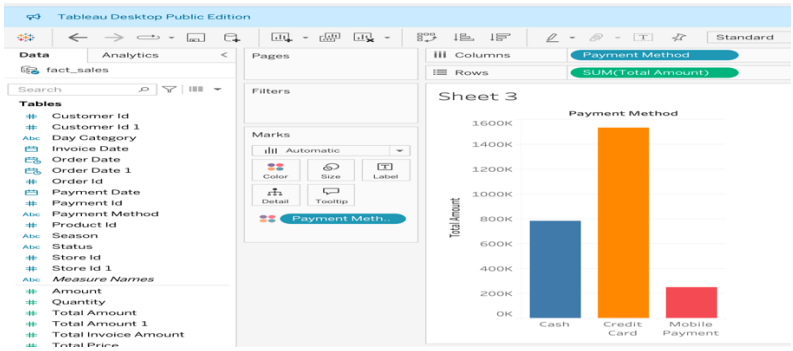

Figure 13: Payment Modes contribution to total revenue

This analysis can help Sylt Fish Specialties in multiple ways like Identify which payment method is most used by customers and contributes maximum revenue, decide whether to promote certain payment methods (like Mobile Payment discounts) based on customer trends etc.

Analysis 5 –Products orders on Weekdays vs Weekends

In this analysis, Bi application understands how customer preferences for different products change between weekdays and weekends. To do this, Bi application used the enriched sales data in Tableau Public, where during ETL phase, data was already categorized each order by “Day_Category” with value either Weekday or Weekend. This classification was made possible

during the ETL process, where enrichment phase added this extra column while loading and transforming the data. Such enrichment allowed us to easily differentiate between orders placed on working days and weekends.

For this, Application prepared a bar chart. On the X-axis is product id and day category and Y-axis average quantity of orders per day. Now the data generated script created almost uniform data over week, so following plot shows uniform trend for the weekday vs weekend analysis but management can use this graph for future if this trend changes to better manage inventory and staffing requirement for weekdays vs weekends.

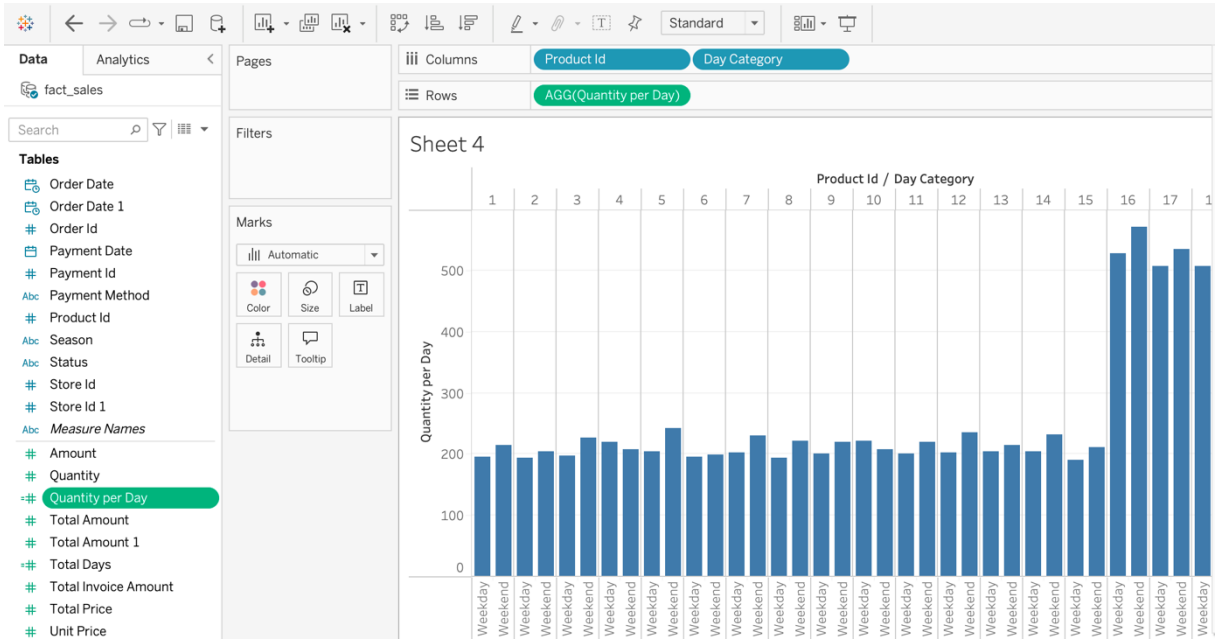

Figure 14:Weekdays vs weekend analysis

Conclusion

Business Intelligence (BI) is about using data to understand how a business is doing. It helps companies track their performance, spot trends, and make better decisions. BI tools turn raw data into useful reports and charts that are easy to understand.

In this project, we collected sales and store data from Sylt Fish Specialties' systems. The raw data was cleaned, enriched, and stored in a Data Warehouse. As part of the enrichment, we added new fields like Day Category to mark if an order was on a weekday or weekend. We also made sure product names, store details, and customer information were correctly mapped.

After the data was ready, we used Tableau Public to connect to the data warehouse. We created different charts and reports — like product-wise sales, seasonal patterns, top and worst-performing stores, payment method trends, and order behaviour on weekdays vs weekends. These charts made it easy to spot patterns and find areas where Sylt Fish Specialties could improve or focus more.

Bibliography

Faker Küttner, R. (2024). *Faker: Python library for generating fake data* (Version 24.x) [Computer software]. https://faker.readthedocs.io/

Petl Alistair, M. (2024). *petl: Extract, Transform and Load data in Python* (Version 1.x) [Computer software]. https://petl.readthedocs.io/

Tableau Public Tableau Software, LLC. (2024). *Tableau Public* (Version 2024.x) [Computer software]. https://public.tableau.com/

Inmon, W. H. (2005). Building the data warehouse (4th ed.). Wiley.