DATA ENGINEERING: DATA WAREHOUSE DESIGN

Presented By : Cathrine J A Xavier

BUSINESS REQUIREMENT

What We Want

- Best-selling item
- Average price of an item
- Customers with the highest revenue
- Delivery reliability of the freight carrier

How It Is Achieved

Based on the given data, a design using multiple fact tables, a bridge table, and well-defined dimension tables offers the most flexibility and accuracy.

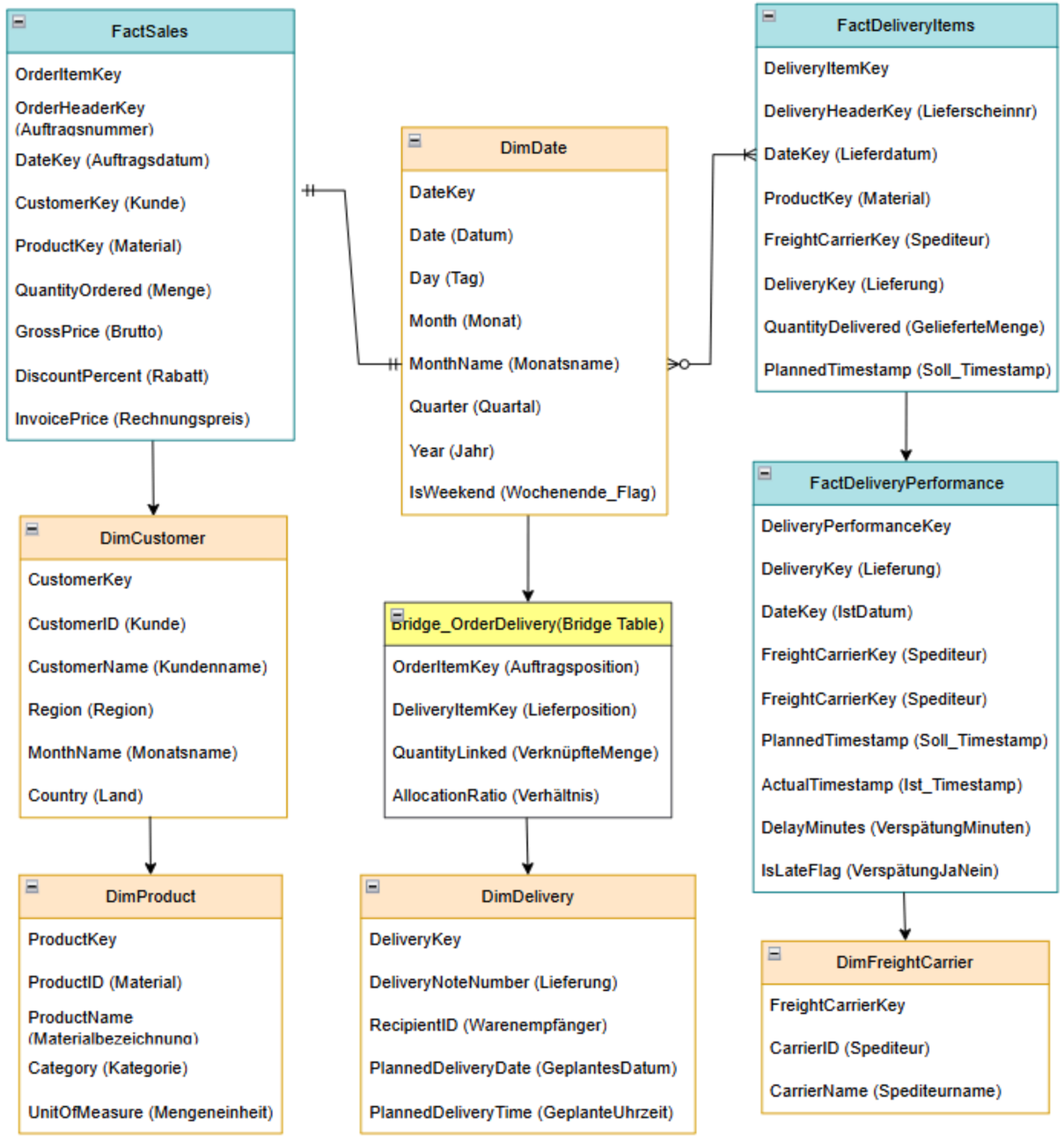
Star Schema with Many-to-Many Relationships

A star schema is designed to handle the complex many-to-many relationship between orders and deliveries.

Bridge Table for Order-Delivery Relationship

A bridge table (Bridge_OrderDelivery) is the solution for
- One order being fulfilled in multiple deliveries
- One delivery fulfilling items from multiple orders

STAR SCHEMA DESIGN



WHY STAR SCHEMA ?

Use of Multiple Fact Tables

Facts Table	Purpose	Aligns to
1. FactSales 2. FactDelivery 3. FactDeliveryPerformance	1. Sales-related KPIs per order item 2. Delivery quantities per item 3. Planned vs. actual delivery timestamps	1. Best-selling item, average price, top customers 2. Quantity shipped per delivery 3. Delivery reliability metrics

✓ This is a best practice when facts come from different business processes (sales vs. delivery vs. carrier performance).

✓ It also keeps each fact table at a consistent grain, making it easier to reason about and maintain.

MODEL MAPPING

1 Best-selling item

- Source: FactSales
- Metric: SUM(Menge) or SUM(Rechnungspreis) by Product
- Dimensions: DimProduct, DimDate

2 Average item price

- Source: FactSales
- Metric: AVG(Rechnungspreis / Menge) by Product
- Caveat: Exclude rows with 0 quantity to avoid division errors

MODEL MAPPING

3 Customers with the Highest Revenue

- Source: FactSales
- Metric: SUM(Rechnungspreis) by CustomerKey
- Join with DimCustomer for names

4 Delivery reliability

- Source: FactDeliveryPerformance
- Metric: % of deliveries where Ist_Timestamp <= Soll_Timestamp
- Group by FreightCarrierKey

ARCHITECTURE OVERVIEW

Concept for the Architecture

Ingestion Layer

Tool:
FTP/API

Description:
Load Excel, Delivery, Feedback (FTP)

Staging Layer:
Raw delivery, order, truck feedback

- Data from ERP system (e.g. SAP) into Order_Header, Order_Items, Delivery_Header, Delivery_Items
- Truck Feedback ingested daily from **FTP server**

Transformation Layer (ETL / ELT)

Tool:
Python/Spark/DBT

Description:
Clean, generate SKs, join Referenzbeleg/Pos

Transformation Layer:
Mapping and surrogate key generation

- Clean raw data
- Join delivery positions with truck feedback
- Explode many-to-many delivery–order relations
- Generate surrogate keys
- Handle late-arriving dimensions (e.g., actual delivery after delivery note)

Data Warehouse Layer

Tool:
Redshift, BigQuery, Snowflake

Description:
Into star schema

Data Warehouse:
Star schema model

- Dimensions as described
- Indexes on surrogate keys, partitioning by delivery date

BI / Analytics Layer

Tool:
Power BI / Tableau dashboards

BI/Reporting:
Power BI, Tableau

- KPIs: Delivery reliability %,
- average item price,
- customer revenue, etc

CHALLENGES AND HOW TO SOLVE THEM

1.Many-to-Many between Orders and Deliveries

Example: Order 00001234, Position 020 delivered in:

- Delivery 004323, Pos 020, Menge 2
- Delivery 004211, Pos 020, Menge 1

Fix: Use BridgeOrderDelivery to avoid double counting in measures

2.Data Linking via Natural Keys

Linking 'Auftrag' + Position from 'Auftragspositionen' to 'Lieferung Positionen'
via 'Referenzbeleg', 'ReferenzPos'

Fix: Carefully parse and validate these keys, generate surrogate keys in ETL

CHALLENGES AND HOW TO SOLVE THEM

3.Delivery Timeliness Calculation

Compare Planned (Datum + Uhrzeit) vs Actual (from "Truck Feedback")

Example:Lieferung 4323 planned at 2/10/22 10:00, actual 2/10/22 10:10 → Delay: 10 minutes

Fix: Create calculated columns like DelayMinutes = Actual - Planned

4.No Explicit Freight Carrier Info

No carrier name in the source

Fix: Assumption: one carrier per Lieferung, or enrich with lookup table

5.Partial Deliveries

Order position 00001234/030 is delivered in full in one delivery
Position 020 is split across two deliveries

Fix: Must store delivered quantity per delivery in BridgeOrderDelivery

OPEN QUESTIONS

- **Q 1**

  How frequently is "Truck Feedback" updated? (Impacts latency of delivery analytics)

- **Q 2**

  Is there a formal freight carrier table, or do we derive from Lieferung?

- **Q3**

  How should unmatched delivery items (orphan records) be handled?

- **Q 4**

  Are returns or cancelled deliveries part of the scope?

- **Q 5**

  Will more granular product details (e.g., category), Customer Details, be added later?

THANK YOU