Praktikum Business Intelligence - Global Bike Data Warehouse

Gruppe: Gruppe 10
Studierende:
Luca Stehle
Thieu, Quang Trung
Achour, Ahmed

25. Oktober 2025

Zusammenfassung

Dieses Dokument beschreibt die Umsetzung eines Data Warehouses für die fiktive Firma Global Bike im Rahmen des Business Intelligence Praktikums. Es umfasst die Analyse der Datenstruktur, die Erstellung eines ER-Modells in der 3. Normalform, die Implementierung in PostgreSQL und die Entwicklung eines ETL-Prozesses mit Bonobo zur Datenintegration und Fehlerbehandlung.

Inhaltsverzeichnis

1 Aufgabe 1: Konzeption und Modellierung

1.1 Global Bike – Kurzüberblick

Global Bike ist eine fiktive Firma aus dem Schulungsmaterial der SAP SE, die als inhaltlicher Rahmen für dieses Praktikum dient.

- **Fusion (2001):** Aus Frankenstein Bikes (USA, John Davis) und Heidelberg Composites (DE, Peter Schwarz).
- **Zwei Gesellschaften:** Global Bike Inc. (USA) & Global Bike Germany GmbH (DE).
- **Standorte:** Dallas, Miami, San Diego, Heidelberg, Hamburg.
- **Produkte:** Rennräder (Deluxe/Professional), Mountainbikes (Damen/Herren), Zubehör, Rohstoffe, Halbfabrikate.
- **Geschäftspartner:** Kunden & Lieferanten in USA und Deutschland.
- **Prozesse:** Vertrieb, Einkauf, Produktion, Finanzen, Controlling, HR, Lager, Service.
- **Strategie:** Fokus auf Qualität, Stärke und Leistung im Sport- und Freizeitbereich.

1.2 ER-Modell für das Global-Bike-Data-Warehouse (3NF)

Das ER-Modell wurde für ein einfaches Data Warehouse in der 3. Normalform (3NF) erstellt, um Datenredundanz zu minimieren und die Datenintegrität zu gewährleisten. Es besteht aus einer zentralen Faktentabelle und mehreren Dimensionstabellen.

1.2.1 Faktentabelle: FactSales

Die Faktentabelle speichert die Verkaufsmetriken und verweist über Fremdschlüssel auf die Dimensionstabellen:

- `OrderItem` (PK, laufende Nummer je Bestellung)
- `OrderNumber` (FK → Order)
- `ProductID` (FK → Product)
- `CustomerID` (FK → Customer)
- `DateID` (FK → Date)
- `SalesQuantity`
- `UnitOfMeasure`
- `RevenueUSD`
- `DiscountUSD`
- `CostsUSD`

1.2.2 Dimensionstabellen

Order

- `OrderNumber` (PK)
- `SalesOrgID` (FK → SalesOrg)
- `Currency`
- `Revenue` (Original Currency)
- `Discount` (Original Currency)

Customer

- `CustomerID` (PK)
- `CustDescr` (Name, z. B. Cruiser Bikes)
- `City`
- `CountryCode` (FK → Country)

Product

- `ProductID` (PK)
- `ProdDescr`
- `ProdCatID` (FK → ProductCategory)
- `DivisionCode` (z. B. BI/AS)

ProductCategory

- `ProdCatID` (PK)
- `CatDescr`

SalesOrg

- `SalesOrgID` (PK)
- `SalesOrgCode` (z. B. DN00, DS00)
- `CountryCode` (FK → Country)

Country

- `CountryCode` (PK)
- `CountryName`

Date

- `DateID` (PK)
- `Date`
- `Year`
- `Month`
- `Day`

1.2.3 Primär-/Fremdschlüssel-Beziehungen

- `FactSales.OrderNumber` → `Order.OrderNumber`
- `FactSales.ProductID` → `Product.ProductID`
- `FactSales.CustomerID` → `Customer.CustomerID`
- `FactSales.DateID` → `Date.DateID`
- `Order.SalesOrgID` → `SalesOrg.SalesOrgID`
- `Customer.CountryCode` → `Country.CountryCode`
- `SalesOrg.CountryCode` → `Country.CountryCode`
- `Product.ProdCatID` → `ProductCategory.ProdCatID`

1.2.4 Begründung / Hinweise

- **3NF erreicht:** Jedes Attribut ist voll funktional abhängig vom Primärschlüssel.
- **Kein Redundanzproblem:** Länder, Kategorien und Organisationen sind ausgelagert.
- **Einfache Übertragbarkeit:** Das Modell ist einfach auf ein späteres Star-Schema übertragbar (Fakt + Dimensionen).

1.2.5 ER-Diagramme

Das Data Warehouse wurde mit zwei komplementären Diagrammen modelliert:

Class Diagram (UML-Klassendiagramm) Das UML-Klassendiagramm zeigt die Struktur der Entitäten und ihre Beziehungen in Abbildung 1.

Abgeleitetes ER-Diagramm Das daraus abgeleitete ER-Diagramm zeigt die finale Datenbankstruktur in Abbildung 2.

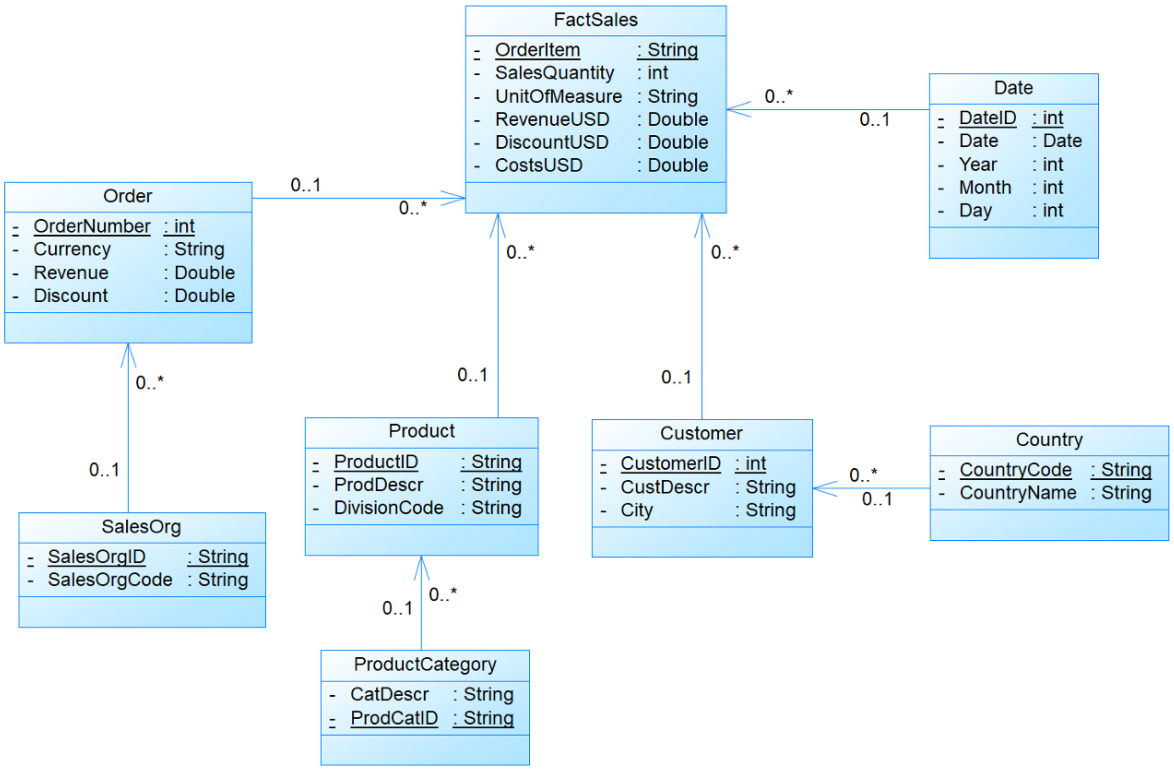

Abbildung 1: UML-Klassendiagramm des Global Bike Data Warehouses

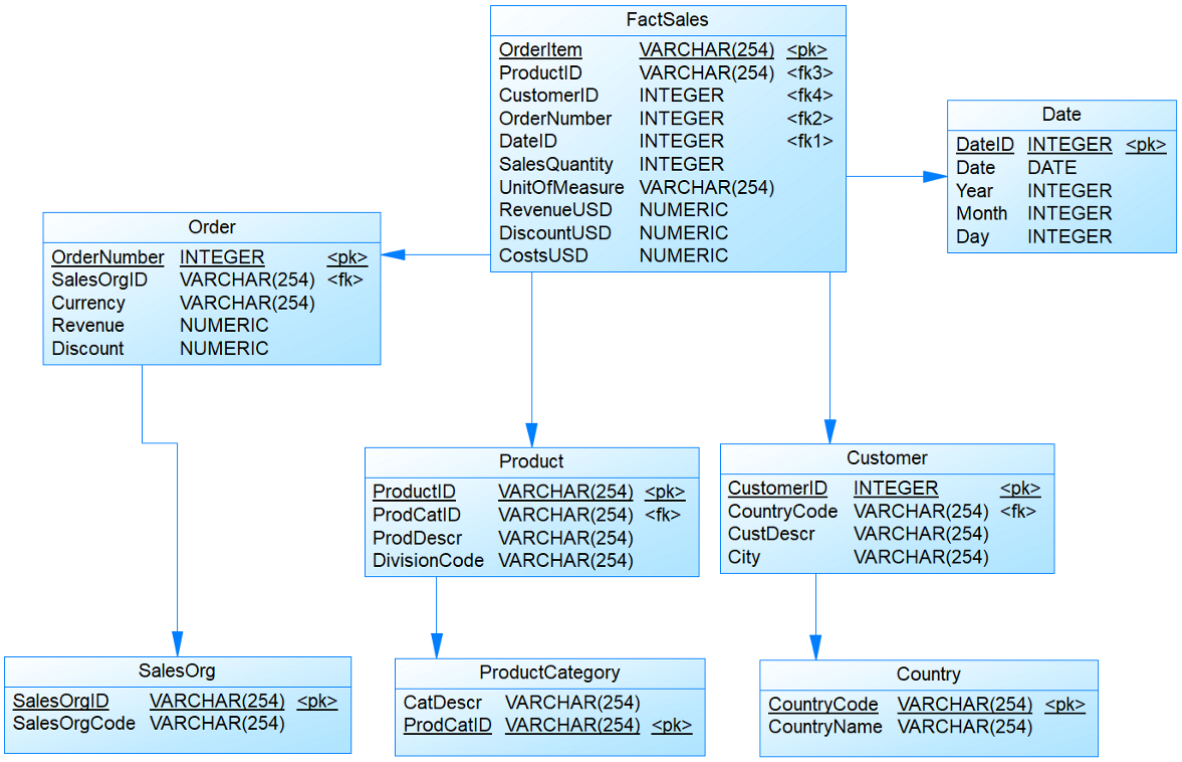

Abbildung 2: Abgeleitetes ER-Modell des Global Bike Data Warehouses (3NF)

2 Aufgabe 2: Implementierung und ETL-Prozess

2.1 Implementierung des Datenmodells in PostgreSQL

Das in Aufgabe 1.b entworfene ER-Modell wurde in einer PostgreSQL-Datenbank implementiert. Die SQL-Befehle zur Erstellung der Tabellen, Primärschlüssel und Fremdschlüssel sind in der Datei `crebas2.sql` enthalten. Die Implementierung erfolgte über das Query Tool in pgAdmin auf dem Server `postgres.fbi.h-da.de`.

2.2 Vertrautmachen mit Bonobo ETL

Für den ETL-Prozess wurde das Bonobo-Framework verwendet. Bonobo ermöglicht die Definition von Datenflüssen als Graphen, was eine klare Strukturierung der Extraktions-, Transformations- und Lade-Schritte (ETL) ermöglicht.

2.3 Laden der Daten mit Bonobo ETL

Die Daten aus der bereitgestellten `SalesData.csv`-Datei wurden mittels eines Bonobo ETL-Prozesses in das Data Warehouse geladen. Der Python-Code für diesen Prozess ist in der Datei `etl_process.py` enthalten.

2.4 Fehlerbehandlung im ETL-Prozess

Der ETL-Prozess wurde so konzipiert, dass er potenzielle Fehler in den Quelldaten identifiziert und, wo möglich, automatisch korrigiert. Ein detaillierter Fehlerbericht wird am Ende des Prozesses generiert.

2.4.1 Ergebnisse des ETL-Prozesses

Der ETL-Prozess wurde erfolgreich mit Python 3.9 ausgeführt. Die folgende Tabelle zeigt die geladenen Datenmengen:

Dimensionstabelle	**Anzahl Datensätze**
Länder	2
Kunden	24
Datumswerte	4.631
Vertriebsorganisationen	4
Bestellungen	36.847
Produktkategorien	6
Produkte	29
Verkaufstransaktionen (FactSales)	**171.010**

Tabelle 1: Zusammenfassung der geladenen Daten

2.4.2 Terminal-Output des ETL-Prozesses

```
PS C:\Users\Luca\OneDrive\Dokumente\Informatik\Master\
   Semester 3\Business Intelligence\Prakitkum\P1> py -3.9
   etl_process.py

============================================================

GLOBAL BIKE SALES DATA - ETL PROCESS
============================================================


Dieser ETL-Prozess l dt die Daten aus SalesData.csv
in das normalisierte Data Warehouse.

Datenbank-Passwort eingeben:
 - extract_csv in=1 out=171010 [done]
 - validate_and_transform in=171010 out=171010 [done]
 - load_dimension_tables in=171010 out=171010 [done]

============================================================

DATENBANK-IMPORT STARTET
============================================================


    Verbindung zur Datenbank hergestellt

Lade 2 L nder...
    2 L nder geladen

Lade 4 Vertriebsorganisationen...
    4 Vertriebsorganisationen geladen

Lade 24 Kunden...
    24 Kunden geladen

Lade 4631 Datumswerte...
    4631 Datumswerte geladen

Lade 36847 Bestellungen...
    36847 Bestellungen geladen

Lade 6 Produktkategorien...
    6 Produktkategorien geladen

Lade 29 Produkte...
    29 Produkte geladen

Lade 171010 Verkaufstransaktionen...
    171010 Verkaufstransaktionen geladen

    Alle Daten erfolgreich in die Datenbank geschrieben!
```

```
============================================================


============================================================

FEHLERBERICHT
============================================================


3 fehlerhafte Datens tze gefunden:

      Zeile 20: W hrung '   ' automatisch zu 'EUR'
     korrigiert
      Zeile 51122: L ndercode 'GER' automatisch zu 'DE'
     korrigiert
      Zeile 66705: L ndercode 'USA' automatisch zu 'US'
     korrigiert

Vollst ndiger Fehlerbericht wurde in '
   fehlerhafte_datensaetze.txt' gespeichert
============================================================


============================================================

ZUSAMMENFASSUNG
============================================================


Geladene Dimensionen:
      2 L nder
      24 Kunden
      4631 Datumswerte
      4 Vertriebsorganisationen
      36847 Bestellungen
      6 Produktkategorien
      29 Produkte
      171010 Verkaufstransaktionen

Fehler: 3 fehlerhafte Datens tze
============================================================
```

Listing 1: Terminal-Output des ETL-Prozesses

2.4.3 Gefundene und korrigierte Fehler

Während des ETL-Prozesses wurden 3 fehlerhafte Datensätze identifiziert und automatisch korrigiert:

- **Zeile 20:** Währung '€' wurde automatisch zu 'EUR' korrigiert.

- **Zeile 51122:** Ländercode 'GER' wurde automatisch zu 'DE' korrigiert.
- **Zeile 66705:** Ländercode 'USA' wurde automatisch zu 'US' korrigiert.

Ein vollständiger Fehlerbericht wurde in der Datei `fehlerhafte_datensaetze.txt` gespeichert.

3 Fazit

Der ETL-Prozess wurde erfolgreich implementiert und ausgeführt. Insgesamt wurden 171.010 Verkaufstransaktionen aus der CSV-Datei in das normalisierte Data Warehouse geladen. Die Datenqualität war sehr hoch, da nur 3 fehlerhafte Datensätze gefunden und automatisch korrigiert wurden. Das entwickelte ER-Modell in 3NF ermöglicht eine effiziente Speicherung und Analyse der Verkaufsdaten ohne Redundanzen.