# Zensus

# ANALYSE DES DATENSATZES

# 

# Dennis Reimer

# August 2024

**Zusammenfassung**

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc140062912)

[2 Kickstarter 3](#_Toc140062913)

[3 Daten und Methoden 4](#_Toc140062914)

[4 Ergebnisse 6](#_Toc140062915)

[5 Empfehlung 9](#_Toc140062916)

[Literaturverzeichnis 10](#_Toc140062917)

# 1 Einleitung

**Beschreibung der Ausgangsdaten und der Zielsetzung der Arbeit**

Die Ausgangsdaten dieser Arbeit stammen aus dem Zensus 2022, einer umfassenden statistischen Erhebung in Deutschland, die detaillierte Informationen über die Bevölkerung, Haushalte und Wohnungen erfasst. Der Zensus liefert essenzielle demografische Daten, die für politische, wirtschaftliche und soziale Planungen unverzichtbar sind. Die Daten umfassen verschiedene Dimensionen wie Schulabschlüsse, Erwerbsstatus, berufliche Abschlüsse und Altersverteilung.

Die Zielsetzung dieser Arbeit besteht darin, die bereitgestellten Zensusdaten zu bereinigen, in die zweite Normalform zu bringen und anschließend in ein Sternschema zu überführen. Dadurch soll eine strukturierte und effiziente Grundlage für umfangreiche Datenanalysen geschaffen werden. Im weiteren Verlauf der Arbeit wird die Datenbank so gestaltet, dass sie sowohl relationale als auch multidimensionale Analysen unterstützt, um fundierte und präzise Erkenntnisse aus den Daten zu gewinnen.

**Kurze Einführung in relationale und multidimensionale Datenbanken**

Relationale Datenbanken basieren auf dem relationalen Modell, das von Edgar F. Codd in den 1970er Jahren entwickelt wurde. In einer relationalen Datenbank werden Daten in Tabellen organisiert, wobei jede Tabelle aus Zeilen und Spalten besteht. Die Spalten definieren die Attribute der Daten, während jede Zeile einen einzelnen Datensatz darstellt. Relationale Datenbanken verwenden Schlüssel, um Beziehungen zwischen Tabellen herzustellen und Integrität zu gewährleisten. SQL (Structured Query Language) ist die primäre Sprache zur Verwaltung und Abfrage relationaler Datenbanken. Beispiele für relationale Datenbanksysteme sind PostgreSQL, MySQL und Microsoft SQL Server.

Multidimensionale Datenbanken hingegen sind darauf ausgelegt, Daten aus verschiedenen Perspektiven oder Dimensionen zu analysieren. Sie werden häufig in Data Warehousing und Business Intelligence verwendet. Das multidimensionale Modell organisiert Daten in Form von „Cubes“, die Dimensionen und Kennzahlen enthalten. Dimensionen sind die Perspektiven oder Kategorien, durch die die Daten betrachtet werden können, wie Zeit, Geografie oder Produktkategorien. Kennzahlen sind numerische Werte, die analysiert werden, wie Umsatz oder Stückzahlen. OLAP (Online Analytical Processing) ist eine Technologie, die in multidimensionalen Datenbanken verwendet wird, um schnelle und interaktive Analysen zu ermöglichen. Beispiele für multidimensionale Datenbanksysteme sind Microsoft Analysis Services und Oracle OLAP.

Durch die Überführung der Zensusdaten in ein Sternschema, einem hybriden Ansatz, der Aspekte sowohl relationaler als auch multidimensionaler Modelle integriert, wird die Basis für effiziente Abfragen und Analysen geschaffen. Dies ermöglicht eine flexible und leistungsstarke Datenanalyse, die sowohl detaillierte Abfragen als auch umfassende multidimensionale Analysen unterstützt.

# 2 ETL-Prozess

**Erklärung unseres ETL-Prozesses (Extract, Transform, Load)**

**Extract**: Der Extraktionsprozess bestand darin, Daten aus einer Excel-Datei zu laden, die verschiedene Tabellenblätter mit spezifischen Datensätzen enthielt. Jedes relevante Tabellenblatt wurde identifiziert und in einen Pandas DataFrame geladen. Dies ermöglichte eine strukturierte und programmatische Verarbeitung der Rohdaten.

**Transform**: Die Transformationsphase beinhaltete mehrere Schritte zur Datenbereinigung und Anpassung:

1. **Konsistenz der Spaltennamen**: Alle Spaltennamen wurden in Kleinbuchstaben umgewandelt, um eine einheitliche Benennung zu gewährleisten.
2. **Ersetzen von Sonderzeichen**: Fehlerhafte Werte wie '.', '/', '-' und '–' wurden durch 0 ersetzt, um Datenintegrität sicherzustellen.
3. **Numerische Konvertierung**: Daten in numerische Typen konvertiert und fehlende Werte durch 0 ersetzt, insbesondere für Spalten ab der fünften.
4. **Bereinigung der 'Regionalebene'-Spalte**: Spezielle Bereinigung für die 'Regionalebene'-Spalte, einschließlich Entfernung von Leerzeichen und Schrägstrichen.
5. **Erstellung neuer Tabellen**: Erstellung von Tabellen wie stadt\_typen, um spezifische Informationen zu extrahieren und zu organisieren.
6. **Aufbau von Regionalebenen**: Separate DataFrames für verschiedene regionale Ebenen erstellt und die entsprechenden Schlüssel extrahiert.

Diese Transformationen wurden vor dem Laden durchgeführt, um sicherzustellen, dass nur bereinigte und konsistente Daten in die Datenbank geladen werden. Der ETL-Prozess wurde gewählt, um die Datenqualität zu maximieren und Transformationen vor dem Laden vorzunehmen, was insbesondere bei heterogenen und potenziell fehlerhaften Datenquellen wie Excel-Dateien von Vorteil ist.

**Load**: Der Ladeprozess bestand darin, die transformierten Daten in eine PostgreSQL-Datenbank zu laden. Hierbei wurden Dimensionstabellen und eine Faktentabelle erstellt und mit den notwendigen Fremdschlüsselbeziehungen versehen, um die Datenintegrität zu gewährleisten.

**b. Kurze Beschreibung der verwendeten Tools und Technologien**

1. **Python**: Die Hauptprogrammiersprache zur Implementierung des ETL-Prozesses. Python wurde wegen seiner Flexibilität und der umfangreichen Bibliotheken für Datenverarbeitung und Datenbankintegration gewählt.
2. **pandas**: Eine leistungsstarke Datenanalysebibliothek in Python, die für das Laden, Bereinigen und Transformieren der Daten verwendet wurde.
3. **SQLAlchemy**: Ein SQL-Toolkit und Object-Relational-Mapping (ORM) Bibliothek für Python, die verwendet wurde, um eine Verbindung zur PostgreSQL-Datenbank herzustellen und die Daten effizient zu laden.
4. **PostgreSQL**: Ein fortschrittliches objektrelationales Datenbankmanagementsystem (ORDBMS), das als Ziel-Datenbank für die transformierten Daten diente. PostgreSQL ermöglicht komplexe Abfragen und unterstützt starke Datenintegrität.
5. **psycopg2**: Ein PostgreSQL-Adapter für Python, der zusammen mit SQLAlchemy verwendet wurde, um die Verbindung zur PostgreSQL-Datenbank herzustellen.
6. **SQLTool**: Eine PostgreSQL-Erweiterung, die verwendet wurde, um SQL-Skripte effizient zu verwalten und auszuführen.

**Begründung für die Wahl eines ETL-Prozesses**: Der ETL-Prozess wurde dem ELT-Prozess vorgezogen, da die Transformation der Daten vor dem Laden in die Datenbank eine bessere Kontrolle über die Datenqualität und -konsistenz ermöglicht. Insbesondere bei heterogenen und potenziell fehlerhaften Datenquellen wie Excel-Dateien ist es entscheidend, die Daten vor dem Laden zu bereinigen und zu transformieren, um sicherzustellen, dass die Datenbank nur valide und konsistente Daten enthält. Dies reduziert den Aufwand für nachträgliche Bereinigungen und Transformationen in der Datenbank erheblich.

# 3 Beschreibung der Normalisierung

# Erklärung der Normalisierung und der 2. Normalform (2NF)

# Normalisierung ist ein Prozess in der Datenbankentwicklung, bei dem Daten in eine Struktur gebracht werden, die Redundanzen minimiert und Datenintegrität maximiert. Die zweite Normalform (2NF) wird erreicht, wenn eine Tabelle in der ersten Normalform (1NF) ist und alle Nicht-Schlüsselattribute vollständig von jedem Kandidatenschlüssel abhängen. Dies bedeutet, dass jedes Nicht-Schlüsselattribut von der gesamten Primärschlüssel-Kombination abhängt und nicht nur von einem Teil davon.

# Darstellung und Begründung der Normalisierung der Daten in der praktischen Umsetzung

# In unserer Studienarbeit haben wir die Ausgangsdaten aus einer Excel-Datei extrahiert und in ein bereinigtes und normalisiertes Format gebracht. Der Prozess umfasste mehrere Schritte:

# Datenbereinigung: Zunächst wurden die Daten aus den Excel-Tabellenblättern eingelesen und bereinigt. Dazu gehörte das Ersetzen von Sonderzeichen durch Nullen, die Umwandlung von Datentypen und die Sicherstellung einheitlicher Formatierungen.

# Spaltennamen und Schlüssel: Alle Spaltennamen wurden in Kleinbuchstaben umgewandelt, um eine konsistente und fehlerfreie Verarbeitung sicherzustellen. Darüber hinaus wurden Schlüsselspalten identifiziert und korrekt benannt, um die Beziehungen zwischen den Tabellen klar zu definieren.

# Erstellen der Dimensionstabellen: Die Daten wurden in verschiedene Dimensionstabellen aufgeteilt, wobei jede Dimensionstabelle Attribute enthielt, die spezifisch für diese Dimension waren (z.B. "dim\_gemeinde" für Gemeindedaten). Diese Tabellen wurden so strukturiert, dass jede Tabelle eine eindeutige Identifikation hatte und keine Datenredundanz aufwies.

# Vermeidung von Redundanzen: Durch die Aufteilung der Daten in mehrere spezialisierte Tabellen wurde sichergestellt, dass keine redundanten Daten gespeichert wurden. Jede Information wurde nur einmal in der passenden Tabelle abgelegt.

# Sicherstellung der Datenintegrität: Durch das Festlegen von Primärschlüsseln und das Definieren von Fremdschlüsselbeziehungen zwischen den Tabellen wurde die Datenintegrität gewährleistet. Dies bedeutet, dass Änderungen in einer Tabelle automatisch in den verknüpften Tabellen berücksichtigt werden, wodurch inkonsistente Daten vermieden werden.

# Diese Schritte der Normalisierung halfen uns, eine saubere und effiziente Datenstruktur zu erstellen, die den Prinzipien der 2. Normalform entspricht. Dies erleichtert die Wartung und Erweiterung der Datenbank und verbessert die Genauigkeit und Verlässlichkeit der Daten für analytische Zwecke

# 4 Sternschema und Olap

**Erklärung des Sternschemas, der Faktentabellen und der Dimensionstabellen**

Das Sternschema ist ein Modell zur Organisation von Daten in Data Warehouses und wird hauptsächlich für OLAP-Anwendungen verwendet. Es besteht aus einer zentralen Faktentabelle, die durch mehrere Dimensionstabellen ergänzt wird. Die Faktentabelle enthält numerische Fakten oder Messgrößen der Geschäftsprozesse, während die Dimensionstabellen beschreibende Attribute enthalten, die Kontext zu den Fakten liefern.

* **Faktentabellen**: Diese Tabellen enthalten die quantitativen Daten, die analysiert werden sollen. Jede Zeile in einer Faktentabelle stellt eine einzelne Messung oder ein Ereignis dar, z.B. Verkaufszahlen oder Anzahl der Besucher. In unserer Implementierung enthält die Faktentabelle "faktentabelle" Messgrößen wie "gesamtbevölkerung", "erwerbstätige", "arbeitslose", "schulabschlüsse" und "beruflicheabschlüsse". Jede dieser Messgrößen ist durch Fremdschlüssel mit den entsprechenden Dimensionstabellen verknüpft.
* **Dimensionstabellen**: Diese Tabellen enthalten die deskriptiven Attribute, die den Fakten Kontext geben und helfen, die Daten zu analysieren. Typische Dimensionen sind Zeit, Geographie, Produkt oder Kunde. In unserer Implementierung gibt es verschiedene Dimensionstabellen wie "dim\_land" (Länder), "dim\_gemeinde" (Gemeinden), "dim\_gemeindeverband" (Gemeindeverbände) und weitere. Jede Dimensionstabelle enthält Attribute, die die Dimensionen beschreiben, z.B. "name" und verschiedene Schlüssel.

Das Sternschema ist besonders effektiv für Abfragen, die große Datenmengen aggregieren, da die Struktur einfach und die Beziehungen zwischen den Tabellen klar definiert sind.

**Erläuterung von OLAP (Online Analytical Processing) und ROLAP (Relational OLAP)**

* **OLAP (Online Analytical Processing)**: OLAP bezeichnet Technologien, die es ermöglichen, Daten aus einem Data Warehouse effizient und interaktiv zu analysieren. OLAP-Systeme bieten die Möglichkeit, komplexe Abfragen schnell durchzuführen und Daten aus verschiedenen Perspektiven zu betrachten, z.B. durch Drill-Down, Roll-Up oder Slicing and Dicing. Dies ermöglicht es Benutzern, tiefgehende Einblicke in die Daten zu gewinnen und fundierte Geschäftsentscheidungen zu treffen.
* **ROLAP (Relational OLAP)**: ROLAP ist eine Form von OLAP, die auf relationalen Datenbanken basiert. Bei ROLAP werden die Daten in relationalen Tabellen gespeichert, und die OLAP-Funktionalität wird durch SQL-Abfragen auf diesen Tabellen bereitgestellt. Der Hauptvorteil von ROLAP ist die Skalierbarkeit, da es große Datenmengen effizient verarbeiten kann. Außerdem nutzt ROLAP die bestehenden Fähigkeiten von relationalen Datenbanken, einschließlich Transaktionen, Sicherheit und Datenintegrität.

In dieser Arbeit wurde das Sternschema in einer relationalen Datenbank (PostgreSQL) implementiert, was ein klassisches Beispiel für ROLAP ist. Durch die Nutzung von SQL-Tools und Erweiterungen wie sqltool konnten die Daten effizient extrahiert, transformiert und geladen werden(ETL-Prozess). Das resultierende Sternschema ermöglicht es, OLAP-Abfragen durchzuführen und komplexe Datenanalysen zu unterstützen, indem es eine strukturierte und optimierte Datenorganisation bietet.

# Literaturverzeichnis