Digital Business University of Applied Sciences

Data Science und Business Analytics

DAI92 SP III Modul 1: Dateninfrastruktur

Uwe Scholl

Aufbau einer Datenarchitektur auf Basis des Zensus 2022 zu Bildung und Erwerbstätigkeit in Deutschland

Studienarbeit

Eingereicht von: Mareike Lass-Hennemann

Matrikelnummer: 190190

30.08.2024

Inhaltsverzeichnis

Abbildungsverzeichnis

Einführung

a. Beschreibung der Ausgangsdaten und der Zielsetzung der Arbeit

Dieser Arbeit liegen Daten des Zensus 2022 zugrunde. Der Zensus ist wie eine moderne Form der Volkszählung. Auf Basis von Interviews mit weniger als zehn Prozent der Bevölkerung Deutschlands, amtlichen Melderegisterdaten, sowie postalischen Befragungen von Besitzer*innen von Wohngebäuden und Eigentumswohnungen wird ein von den statistischen Ämtern des Bundes und der Länder entwickeltes Verfahren angewandt. So können verlässliche Zahlen zu den Themen Bevölkerung, Demografie, Gebäude und Wohnungen, Haushalte und Familien sowie Bildung und Erwerbstätigkeit erhoben werden, ohne dass jeder Mensch in Deutschland befragt werden muss (Zensus, 2022). In dieser Arbeit wurden die Daten zum Thema Bildung und Erwerbstätigkeit verwendet.

Die Ausgangsdaten liegen als Excel-Arbeitsmappe vor. Es handelt sich hierbei um zehn aufbereitete Arbeitsblätter mit Filterfunktion sowie zehn einfache Tabellenblätter (gekennzeichnet durch die Benennung „CSV-…“) ohne Funktionen. Zusätzlich sind ein Titelblatt, das Impressum, eine Erläuterung zur Methodik und eine Erläuterung zu den mit „CSV-“ gekennzeichneten Blättern vorhanden. Jedes Arbeitsblatt enthält die Attribute „Amtlicher Gemeindeschlüssel (AGS)“ (in den „CSV-“Blättern „_RS“), „Name“ und „Regionalebene“. In den „CSV-“Blättern ist zusätzlich das Attribut „Berichtszeitpunkt“ vorhanden. Neben diesen beschreibenden Attributen enthält jedes Arbeitsblatt die Zählungen von Personen fokussiert auf eine Dimension (zum Beispiel „Klassenstufe“) mit verschiedenen Ausprägungen (zum Beispiel „Klasse 1 bis 4“, „Klasse 5 bis 10“, „Gymnasiale Oberstufe“). Die Zählungen werden neben den genannten Attributen außerdem nach Geschlecht unterteilt und es sind immer auch Gesamtwerte angegeben. Die genannten Dimensionen sind thematisch den zwei Blöcken Bildung und Erwerbstätigkeit zuzuordnen. Im Block Bildung werden Personen in schulischer Ausbildung nach Klassenstufe und nach Schulform gezählt sowie Personen ab fünfzehn Jahren nach höchstem schulischem Abschluss und höchstem beruflichen Abschluss. Zum Block Erwerbstätigkeit gehört die Zählung von allen Personen nach ihrem Erwerbsstatus und die Zählung aller erwerbstätigen Personen nach Altersgruppe, nach höchstem beruflichem Abschluss, nach Berufsgruppe, nach Stellung im Beruf und nach Wirtschaftszweig.

Ziel ist es nun, eine Datenarchitektur aufzubauen, die es ermöglicht, schnell und übersichtlich Zahlen für die verschiedenen Dimensionen und ihre jeweiligen Ausprägungen auf Ebene der Bundesländer oder Kreise abrufen und z.B. in Visualisierungen weiterverarbeiten zu können. Hierzu werden die einzelnen Datenblätter in einer lokalen Postgres-Datenbank gespeichert, normalisiert und in ein Sternschema überführt.

b. Kurze Einführung in relationale und multidimensionale Datenbanken

In einer relationalen Datenbank sind die Daten in Tabellen organisiert. Jede Tabelle besteht aus Zeilen (Datensätzen) und Spalten (Attribute), wobei jede Zeile durch einen Primärschlüssel eindeutig identifiziert wird. Relationale Datenbanken nutzen SQL (Structured Query Language) zur Datenabfrage und -manipulation.

Neben relationalen Datenbanken gibt es außerdem multidimensionale Datenbanken. Anstatt Daten in Tabellen zu speichern, werden die Daten in "Würfeln" (Cubes) organisiert,

die Dimensionen und Metriken enthalten. Jede Dimension repräsentiert eine Perspektive (z.B. Zeit, Standort, Produkt), als Metriken dienen (Zahlen-)Werte wie z.B. Umsatz oder Menge. Multidimensionale Datenbanken nutzen z.B. MDX oder DAX zur Datenabfrage und -manipulation.
Für dieses Projekt wurde die relationale Datenbank Postgres genutzt.

ETL-Prozess

a. Erklärung des ETL-Prozesses (Extract, Transform, Load)

Es werden die Prozesse ETL und ELT unterschieden, wobei beim ETL-Prozess die Transformation im ETL-Tool vorgenommen wird und beim ELT-Prozess die Transformation erst im Datenbanksystem stattfindet. Für dieses Projekt wird der ETL-Prozess angewendet. Die Daten werden aus Excel in Dataframes in einem Jupyter Notebook in VSCode geladen, dort transformiert und dann in die Datenbank gespeichert. Eine vorherige „ingestion“ im Sinne von Sammlung, Übertragung und Speicherung von Daten in einem Zielsystem muss nicht stattfinden, da die Daten bereits gesammelt und gebündelt zur Verfügung gestellt wurden.

Vorweg werden die Daten über die aufbereiteten Excel-Arbeitsblätter grob erkundet. Hierbei fällt auf, dass zwar für verschiedene Regionalebenen jeweils Zahlen vorliegen, diese jedoch nicht für jede Ebene vollständig sind. So stimmen die Gesamtsummen der Bundesebene, Landesebene und der Ebene Stadtkreis/kreisfreie Stadt/Landkreis immer überein, diejenigen der Ebenen Gemeinde, Gemeindeverband und Regierungsbezirk jedoch nicht, sodass diese Ebenen nicht mit in die Datenbank aufgenommen werden sollten. Zudem sind Zellen mit Zeichen anstatt Zahlen vorhanden, wobei ein „/“ für „keine Angabe, da Zahlenwert nicht sicher genug“ steht. Alle anderen Zeichen sind nicht mehr relevant, da sie durch den Wegfall der unvollständigen Regionalebenen nicht mehr auftauchen. Eine weitere Besonderheit liegt im Format der Ausprägungen des Attributes „_RS“ bzw. Gemeindeschlüssel: hier gibt es Werte, die mit „0“ anfangen. Um diese Null zu Beginn der Zeichenkette zwecks späterer Weiterverarbeitung zu behalten, müssen diese Werte als string übernommen werden. Weiter zu beachten ist, dass die Daten in einem breiten pivot-ähnlichen Format dargestellt sind, sodass die Zählungen pro Arbeitsblatt und für jede Dimensionsausprägung jeweils in drei Spalten für männlich, weiblich, insgesamt vorliegen.

Mit diesem Vorwissen werden im ersten Prozessschritt „extract“ die Daten aus den „CSV-“Blättern mittels Python und Jupyter Notebooks pro Blatt in jeweils ein Dataframe geladen und alle Dataframes in einem Dictionary gespeichert, sodass über alle Dataframes gemeinsam iteriert werden kann. Die Werte aller Spalten werden wie oben erwähnt explizit als strings übernommen.

Es folgt nun der umfangreichere Prozessschritt „transform“. Es werden zunächst die überflüssigen Regionalebenen (s.o.) aus allen Dataframes gelöscht und die Spalten, die die Zählwerte enthalten wieder in integer umgewandelt – es bleiben „_RS“, „Regionalebene“ und „Name“ im Format string, sodass für „_RS“ die führenden Nullen erhalten bleiben. Nun beginnt die Transformation der einzelnen Dataframes, die beispielhaft am Dataframe „df_klassenstufe“ beschrieben werden soll: mittels der Funktion pd.melt() wird das bereits erwähnte breite Format in ein langes Format umgewandelt, sodass statt zwölf Spalten mit Zählwerten zu verschiedenen Geschlechts- und Klassenstufenkombinationen nur noch eine Spalte mit Zählwerten vorhanden ist, sowie eine neue Spalte, in der die vorherigen Spaltenüberschriften nun zeilenweise als Dimensionsausprägung zum zugehörigen Zählwert eingetragen sind. Im umgewandelten Dataframe gibt es nun also nur noch die Spalten „Berichtszeitpunkt“, „_RS“, „Name“, „Regionalebene“, „Subtyp_0“ (entspricht der Kombination Klassenstufe/Geschlecht) und „Anzahl“. In einem weiteren Transformationsschritt werden mittels selbst definierter Python-Funktionen „derive_klassenstufe“ und „derive_geschlecht“ die Ausprägungen der Dimension Klassenstufe und die verschiedenen Geschlechter extrahiert und in jeweils eigene Spalten „Subtyp“ und „Geschlecht“ überführt, sowie die ursprüngliche gemeinsame Spalte „Subtyp_0“ gelöscht. Außerdem in diesem Zuge gelöscht, werden alle aggregierten Zahlen, da diese jederzeit neu berechnet werden können. Die Zeilen mit aggregierten Zahlen wurden innerhalb der zuvor genannten Funktionen anhand fehlender Klassenstufen und/oder Geschlechtszuordnung mit „NaN“ in den Spalten „Subtyp“ und/oder „Geschlecht“ sozusagen markiert, sodass dann alle Zeilen mit „NaN“ in diesen beiden Spalten gelöscht werden können. Diese Schritte werden auf alle Dataframes separat angewendet, da insbesondere die Befüllung der Spalte mit den thematischen Dimensionsausprägungen durch eine jeweils individuelle, nach der jeweiligen Kodierung im Blatt „Erläuterung zu CSV-Tabellen“ aufgelösten Funktion erfolgt. Es werden alle bearbeiteten Dataframes in ein neues Dictionary gespeichert, sodass nochmals über alle

gemeinsam iteriert werden kann. Es wird eine weitere Spalte „einheit“ zugefügt, die für jeden Dataframe die thematische Beschreibung enthält z.B. für den Dataframe mit den Daten zur Klassenstufe, kommt in „einheit“ der Wert „klassenstufe“. Dies geschieht bereits in Vorarbeit für das spätere Sternschema. In einem weiteren Schritt werden sämtliche Großbuchstaben in Spalten und Zeilen in Kleinbuchstaben umgewandelt.

Es folgt dann der letzte Prozessschritt „load“, in dem die Übertragung aller Dataframes jeweils als eigene Tabelle mit dem suffix „raw_“ in eine lokale Postgres Datenbank stattfindet.

b. Kurze Beschreibung der verwendeten Tools und Technologien

* Postgres Datenbank: Zur Speicherung der Daten wird eine lokale Postgres Datenbank verwendet. PostreSQL ist ein kostenloses Open-Source-Managementsystem für objektrelationale Datenbanken.

*PG Admin: Zur Verwaltung der Datenbank wird PG Admin genutzt. Für diese Arbeit ist dies insbesondere nützlich, um schnell fehlerhafte Tabellen wieder löschen zu können.

*VS Code: Für die Entwicklung des Codes wird VS Code genutzt.

*Jupyter Notebook / Python: Für den ETL-Prozess sowie für die abschließenden Visualisierungen wird die Erweiterung Jupyter Notebooks in VS Code genutzt und innerhalb dieser mit einem Python Kernel gearbeitet.

*PostgreSQL Explorer wird ebenfalls als Erweiterung in VS Code installiert, um eine Verbindung zur Datenbank aufzubauen. Es können dann mittels .sql-Dateien Abfragen an die Datenbank gestellt und direkt in VS Code ausgeführt und angezeigt werden. Die komplette Normalisierung und die Überführung ins Sternschema finden auf diese Weise statt.

Beschreibung der Normalisierung

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

Die Normalisierung ist ein Vorgehen, bei dem eine Datenbank so strukturiert wird, dass sie effizient und fehlerfrei arbeitet. Ziel ist es, Redundanz zu minimieren und Inkonsistenzen zu vermeiden. Normalisierte Datenbanken können einfacher aktualisiert werden, da Änderungen nur an einer Stelle vorgenommen werden müssen. Die Normalisierung erfolgt

in mehreren Stufen oder „Normalformen“, wobei jede Stufe zusätzliche Anforderungen an die Struktur der Datenbank stellt. In der ersten Normalform befindet sich eine Tabelle, wenn alle Spalten atomar sind - d.h. es sind keine mehrfachen Werte oder Listen in einer Zelle vorhanden - und jede Zeile eindeutig identifiziert werden kann. Eine Tabelle befindet sich in der zweiten Normalform, wenn sie sich in der ersten Normalform (1NF) befindet und zusätzlich alle nicht-schlüsselbezogenen Attribute vollständig funktional vom gesamten Primärschlüssel abhängen. Das bedeutet, dass ein Attribut (Spalte) in der Tabelle vollständig vom gesamten Primärschlüssel abhängen muss und nicht nur von einem Teil des Primärschlüssels.

b. Darstellung und Begründung der Normalisierung der Daten in der praktischen Umsetzung

Die raw_Tabellen aus dem ETL-Prozess liegen bereits in der ersten Normalform vor: Jede Zelle enthält nur einen Wert. Jede Zeile kann durch den Verbundschlüssel aus den Attributen „_rs“, „subtyp“ und „geschlecht“ eindeutig identifiziert werden.

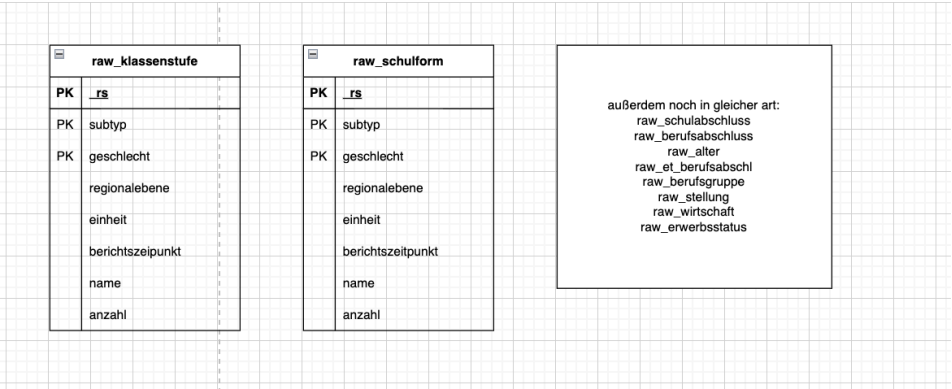

Abbildung 1: Normalisierung 1NF

Für die Überführung in die zweite Normalform werden die Tabellen noch einmal umgestaltet. Zunächst werden die Attribute „name“ (als region_name) und „regionalebene“ in eine eigene Tabelle „kreis“ mit dem Primärschlüssel „region_id“ (die Werte entsprechen denen in _rs“) geschrieben, da diese nicht vom gesamten Verbundschlüssel abhängen, sondern allein von „_rs“. Um Redundanz zu vermindern und Übersichtlichkeit zu gewinnen, werden Kreise und Länder nun getrennt: dafür werden in die Tabelle zunächst vier weitere Spalten „kreis_id“, „kreis_name“ sowie „land_id“ und „land_name“ aufgenommen. Für das Attribut „kreis_id“ werden alle Werte aus „region_id“ aufgenommen, an denen „regionalebene“ dem Wert „stadtkreis/kreisfreie

stadt/landkreis“ entspricht – nach demselben Vorgehen werden für „kreis_name“ alle Werte aus „region_name“ aufgenommen. Nun wird die Spalte „land_id“ mit den ersten zwei Zeichen der „kreis_id“ befüllt und ein mapping aufgesetzt, welches jeder Zeichenkombination ein Bundesland zuweist, das nun entsprechend in „land_name“ eingetragen wird. Nun können die Spalten „region_id“, „region_name“ und „regionalebene“ gelöscht werden und nach Erstellung einer weiteren Referenztabelle „land“ auch die Spalte „land_name“. „kreis_id“ wird in der Tabelle „kreis“ der neue Primärschlüssel.

Außerdem wird für jede „raw_“-Tabelle eine passende „-typ“-Tabelle gebildet, die den Primärschlüssel „_id“ passend zu „_typ“ (entspricht den Werten von „subtyp“) enthält, sowie das Attribut „einheit“, da dieses allein von „subtyp“ abhängig ist, aber ebenfalls nicht von allen drei Teilen des Verbundschlüssels. Es werden noch weitere Referenztabellen mit ID’s gebildet, damit potentielle Änderungen der Attributwerte, die häufig vorkommen, weniger fehleranfällig sind. Hierfür entstehen die Tabellen „geschlecht“, „land“ und „zeitpunkt“. Zuletzt wird aus jeder „raw_“Tabelle jeweils eine Haupttabelle gemacht, die nun einen synthetischen Primärschlüssel „haupt_id“ enthält sowie die Werte aus dem jeweiligen Attribut „anzahl“ und die Fremdschlüssel zu den Tabellen „_typ“, „kreis“, „geschlecht“ sowie „zeitpunkt“ (siehe Abb. 2)

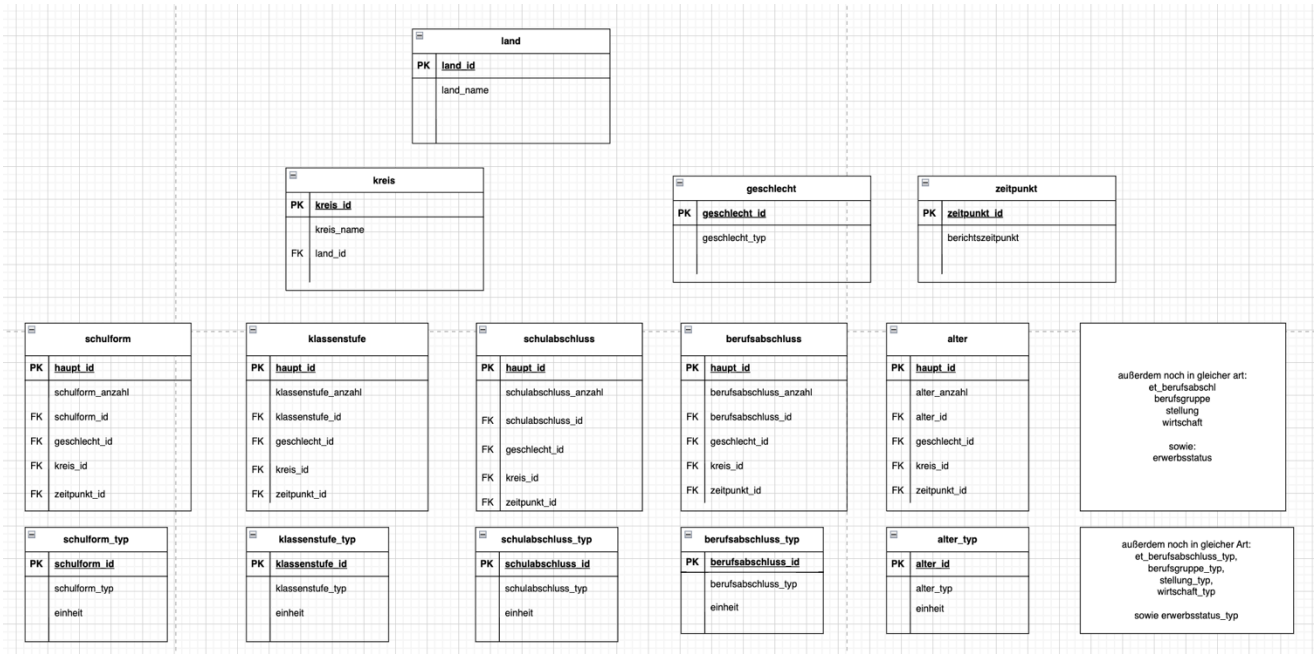

Abbildung 2: Normalisierung 2NF

Sternschema und OLAP

a. Erklärung des Sternschemas, der Faktentabellen und der Dimensionstabellen

Das Sternschema bietet eine klare Trennung von quantitativen Fakten in der zentralen Faktentabelle und zugehörigen Kontextinformationen in Dimensionstabellen. Es erleichtert Abfragen und Analysen, enthält aber auch mehr Redundanzen, sodass Aktualisierungen komplexer sind und eventuell mehr Speicherplatz benötigt wird. Das hier entworfene Sternschema besteht aus einer Faktentabelle „zensus_fakten“ und vier Dimensionstabellen „zeitpunkt“, „geschlecht“, „regionalebenen“ und „einheiten“. Die Faktentabelle enthält vor allem das Attribut „anzahl“, in der sämtliche Zählwerte enthalten sind, sowie die Fremdschlüssel zu allen Dimensionstabellen. Die Dimensionen enthalten jeweils ihre ID und maximal zwei beschreibende Attribute. Die Tabellen „geschlecht“ und „zeitpunkt“ sind gegenüber dem normalisierten Datenmodell unverändert geblieben. Die Tabelle „regionalebenen“ enthält weiter das Attribut „kreis_id“ wie in der normalisierten Tabelle „kreis“, aber es ist keine ID und Referenztabelle für „land“ und „land_name“ mehr vorhanden, sondern „land_name“ steht schon direkt als Attribut in „regionalebenen“. In der Tabelle „einheiten“ werden alle „_typ“-Tabellen des normalisierten Datenmodells zusammengeführt, hier sind nun wieder die Attribute „subtyp“ und „einheit“ vorhanden. Mittels der Dimensionstabellen kann über einen Join die Faktentabelle gefiltert oder über „fakten_id“ oder mittels gezielter Spaltenauswahl gesliced werden (siehe Abb.3).

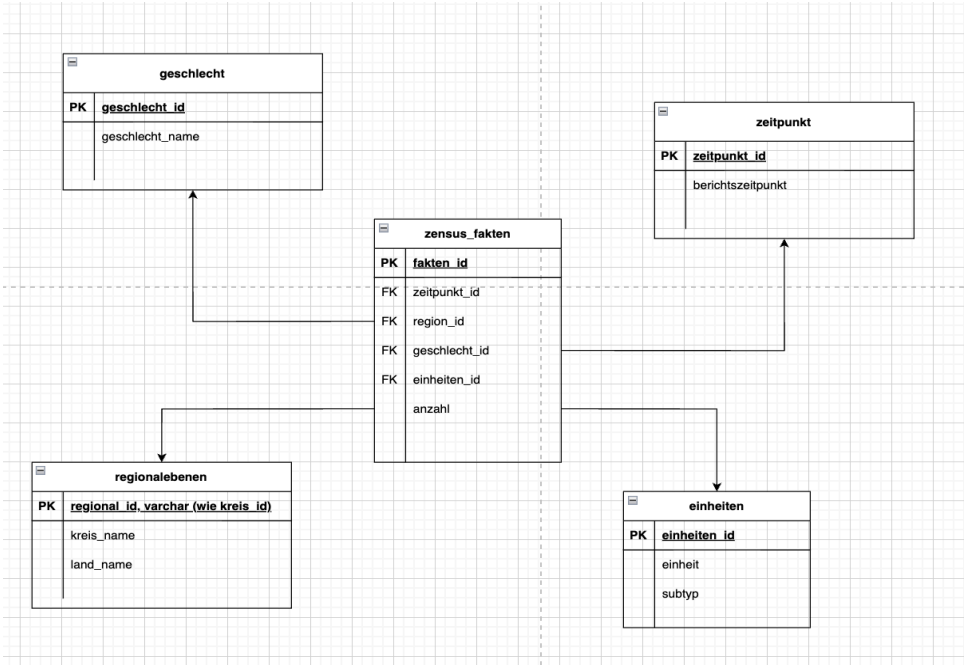

Abbildung 3: Sternschema

b. Erläuterung von OLAP (Online Analytical Processing) und ROLAP (Relational OLAP)

OLAP ist eine Technologie, die darauf abzielt, komplexe Abfragen und Analysen großer Datenmengen effizient und schnell durchzuführen. OLAP-Systeme sind für multidimensionale Abfragen ausgelegt und ermöglichen es Benutzern, Daten aus verschiedenen Perspektiven und Dimensionen zu analysieren (z. B. nach Zeit, Ort, Produktkategorie).

ROLAP ist eine spezielle Art von OLAP, die relationalen Datenbanken (z. B. SQL-basierte Datenbanken) verwendet, um die multidimensionale Analyse durchzuführen. Anstatt Daten in einem multidimensionalen Cube zu speichern, speichert ROLAP die Daten in relationalen Tabellen und simuliert die multidimensionale Analyse durch komplexe SQL-Abfragen.

Das vorliegende Datenmodell im Sternschema ist in einer relationalen Datenbank gespeichert und eignet sich ideal für ROLAP. Die Faktentabelle enthält die Kennzahlen, während die Dimensionstabellen Kontext und Struktur für die Analyse bieten. SQL-Abfragen können geschrieben werden, um die Tabellen zu verbinden und komplexe, multidimensionale Analysen durchzuführen.

Literaturverzeichnis / Quellen

Zensus, 2022. Abgerufen am 23.08.2024 von https://www.zensus2022.de/DE/Wie-funktioniert-der-Zensus/_inhalt.html#_njpegw1b6