Regional Economics Database for NRW

Database Structure and Design Guide

Version:	1.0
Database Name:	regional_economics
Architecture:	Star Schema (Data Warehouse)
Created:	December 2024
Last Updated:	December 17, 2025

Executive Summary

This database stores economic, demographic, and labor market indicators for North Rhine-Westphalia (NRW) regions using a star schema design. All metrics from 36+ different source tables are stored in a unified structure with shared dimension tables for geography, time, and indicators.

Key Benefits:

- Single query pattern works for all data types
- Easy to add new indicators without schema changes
- Optimized for analytical queries and reporting
- Consistent data structure across all categories
- Scales efficiently with growing data volumes

1. Database Architecture

Star Schema Overview

The database follows a star schema pattern where dimension tables (reference data) surround fact tables (measurements). This design optimizes analytical queries and provides a consistent structure across different data types.

Architecture Diagram:

```
DIMENSION TABLES (Reference/Lookup Data)
═══════════════════════════════════════════

┌─────────────────┐         ┌─────────────────┐
│ dim_geography   │         │    dim_time     │
├─────────────────┤         ├─────────────────┤
│ • geo_id (PK)   │         │ • time_id (PK)  │
│ • region_code   │         │ • year          │
│ • region_name   │         │ • reference_date│
│ • region_type   │         │ • quarter       │
│ • ruhr_area     │         └─────────────────┘
└─────────────────┘

      ┌─────────────────┐
      │  dim_indicator  │
      ├─────────────────┤
      │ • indicator_id  │
      │ • indicator_code│
      │ • category      │
      │ • source_table  │
      └─────────────────┘

FACT TABLES (Measurement Data)
═══════════════════════════════

┌───────────────────────────────────────┐
│      fact_demographics                │
│  (Population & Employment Data)       │
├───────────────────────────────────────┤
│ • geo_id (FK) → dim_geography        │
│ • time_id (FK) → dim_time            │
│ • indicator_id (FK) → dim_indicator  │
│ • value (THE MEASUREMENT)            │
│ • gender, nationality, age_group     │
└───────────────────────────────────────┘
```

2. Dimension Tables

Dimension tables contain descriptive attributes and reference data that provide context for the measurements in fact tables.

2.1 dim_geography (Geographic Dimension)

Purpose: Defines all geographic regions in the database

Current Count: 60 regions

Column	Type	Description	Example
geo_id	SERIAL (PK)	Unique identifier	1, 2, 3...
region_code	VARCHAR(20)	Official region code	05112, 05, DG
region_name	VARCHAR(255)	Region name (German)	Duisburg, NRW
region_type	VARCHAR(50)	Type of region	urban_district, state
ruhr_area	BOOLEAN	Part of Ruhr?	TRUE/FALSE
latitude	DECIMAL	GPS coordinate	51.4344
longitude	DECIMAL	GPS coordinate	6.7623
area_sqkm	DECIMAL	Area in km^2	232.82

Region Types:

- district - Rural district (Kreis)
- urban_district - Independent city (Kreisfreie Stadt)
- administrative_district - Regional government area (Regierungsbezirk)
- state - Federal state (Bundesland)
- country - National level (Deutschland)

2.2 dim_time (Temporal Dimension)

Purpose: Defines temporal periods for data

Current Count: 17 years (2008-2024)

Column	Type	Description	Example
time_id	SERIAL (PK)	Unique identifier	1, 2, 3...
year	INTEGER	Calendar year	2024
reference_date	DATE	Specific date	2024-06-30
reference_type	VARCHAR(50)	Type of reference	mid_year
quarter	INTEGER	Quarter (1-4)	2
month	INTEGER	Month (1-12)	6

2.3 dim_indicator (Indicator Dimension)

Purpose: Defines what each measurement represents

Current Count: 4 indicators (40+ planned)

Column	Type	Description
indicator_id	SERIAL (PK)	Unique identifier
indicator_code	VARCHAR(100)	Short code (e.g., pop_total)
indicator_name	VARCHAR(255)	Full name (German)
indicator_category	VARCHAR(100)	Category (e.g., demographics)
source_table_id	VARCHAR(50)	GENESIS table (e.g., 12411-03-03-4)
unit_of_measure	VARCHAR(50)	Unit (e.g., persons, employees)
update_frequency	VARCHAR(50)	How often updated (e.g., annual)

Currently Loaded Indicators:

ID	Code	Name	Source Table	Records
1	pop_total	Population total	12411-03-03-4	17,556
2	employment_workplace	Employment at workplace	13111-01-03-4	798
9	employment_sector	Employment by sector	13111-07-05-4	19,134
3	employment_scope	Employment by scope	13111-03-02-4	~8,700

3. Fact Tables

Fact tables contain measurements (the actual numbers) with foreign keys linking to dimension tables. Each record represents a specific measurement for a particular region, time period, and indicator.

3.1 fact_demographics

Purpose: Population and demographic indicators

Current Records: 45,000+

Column	Type	Description
geo_id	INTEGER (FK)	Links to dim_geography → Which region?
time_id	INTEGER (FK)	Links to dim_time → Which year/period?
indicator_id	INTEGER (FK)	Links to dim_indicator → What metric?
value	NUMERIC(20,4)	THE MEASUREMENT - the actual number
gender	VARCHAR(20)	male, female, total
nationality	VARCHAR(50)	german, foreign, total
age_group	VARCHAR(50)	0-5, 6-17, 18-64, 65+, total
notes	TEXT	Additional info (sector, scope)
data_quality_flag	VARCHAR(20)	V=Validated, E=Estimated, P=Provisional

Example Record:

```
geo_id = 5            → Duisburg (from dim_geography)
time_id = 15          → Year 2024, June 30 (from dim_time)
indicator_id = 9      → Employment by sector (from dim_indicator)
value = 156,999.00    → THE ACTUAL NUMBER
gender = 'total'
nationality = 'total'
notes = 'Sector: Dienstleistungsbereiche (G-U)'

INTERPRETATION: In Duisburg on June 30, 2024, there were
156,999 employees in the service sector.
```

4. How It Works

Traditional vs Star Schema Approach

Traditional Approach (Complex):

- tbl_population → Unique structure
- tbl_employment → Different structure
- tbl_unemployment → Different structure
- tbl_gdp → Different structure
- ... 36 different table structures with different query patterns

Star Schema Approach (Unified):

- ALL data → fact_demographics, fact_labor_market, etc.
- Same query pattern for everything
- indicator_id tells you what the data means
- Add new indicators without changing schema

5. Example Queries

5.1 Get Duisburg Population for 2024

```
SELECT
    g.region_name,
    t.year,
    i.indicator_name,
    f.value
FROM fact_demographics f
JOIN dim_geography g ON f.geo_id = g.geo_id
JOIN dim_time t ON f.time_id = t.time_id
JOIN dim_indicator i ON f.indicator_id = i.indicator_id
WHERE g.region_code = '05112'           -- Duisburg
  AND t.year = 2024
  AND i.indicator_code = 'pop_total';
```

Result:

region_name	year	indicator_name	value
Duisburg	2024	Bevölkerung insgesamt	502,270

5.2 Employment Trend for Duisburg (2020-2024)

```
SELECT
    t.year,
    SUM(f.value) as total_employment
FROM fact_demographics f
JOIN dim_geography g ON f.geo_id = g.geo_id
JOIN dim_time t ON f.time_id = t.time_id
WHERE g.region_code = '05112'           -- Duisburg
  AND f.indicator_id = 9                 -- Employment by sector
  AND t.year BETWEEN 2020 AND 2024
GROUP BY t.year
ORDER BY t.year;
```

6. Data Flow: From Source to Database

ETL Pipeline Process:

STEP 1: EXTRACT

- Source Table: e.g., 13111-07-05-4 (Employment by sector)
- Output: Raw CSV data (~7,500 rows per year)

STEP 2: TRANSFORM

- Filter to NRW regions only
- Map region_code → geo_id (lookup in dim_geography)
- Map year → time_id (lookup in dim_time)
- Assign indicator_id
- Validate and clean values

STEP 3: LOAD

- Bulk insert for performance
- Validate foreign key constraints
- Update table registry

7. Current Database Status

As of December 17, 2025:

Component	Status	Details
Dimension Tables	Complete	4 tables fully functional
dim_geography	Populated	60 NRW regions loaded
dim_time	Populated	17 years (2008-2024)
dim_indicator	Partial	4 of 40+ indicators defined
Fact Tables	In Progress	
fact_demographics	Active	45,000+ records
fact_labor_market	Pending	Using demographics table
Other fact tables	Pending	Schema created, not populated

Category	Tables	Status
Demographics	1 table	100% complete (17,556 records)
Labor Market	3 tables	3 completed, 9 pending
Economic Activity	8 tables	Not started
Healthcare	6 tables	Not started
Public Finance	3 tables	Not started
Infrastructure	1 table	Not started
Mobility/Commuters	2 tables	Not started

8. Frequently Asked Questions

Why use a star schema instead of separate tables?

Star schema provides flexibility and consistency. Adding new indicators doesn't require creating new tables, just new rows in dim_indicator. All queries follow the same pattern, making the database easier to learn and use.

How do I know which fact table to query?

Check the indicator_category in dim_indicator. Demographics → fact_demographics, Labor Market → fact_labor_market (or fact_demographics currently), Business → fact_business_economy.

What's the difference between geo_id and region_code?

geo_id is the internal database key (1, 2, 3...), while region_code is the official GENESIS code (05112, 05, DG). Use region_code in WHERE clauses for readability.

Can I add calculated/derived indicators?

Yes! Insert into dim_indicator with is_derived = TRUE, then calculate and insert the values into the appropriate fact table.

How is data quality tracked?

Each fact record has a data_quality_flag (V=Validated, E=Estimated, P=Provisional) and optional confidence_score. Check these fields when data accuracy is critical.

9. Getting Started

Step 1: Connect to Database

Database: regional_economics
Host: localhost (or your server address)
Port: 5432
Username: (your username)
Password: (your password)

Step 2: Explore the Data

Count total records:

```
SELECT COUNT(*) FROM fact_demographics;
```

List available regions:

```
SELECT region_code, region_name FROM dim_geography ORDER BY region_name;
```

List loaded indicators:

```
SELECT indicator_id, indicator_code, indicator_name FROM dim_indicator;
```

Check year coverage:

```
SELECT DISTINCT year FROM dim_time ORDER BY year;
```

Regional Economics Database for NRW

Database Structure Guide v1.0

For questions or support, contact: Kanyuchi