

## Project Description

End-to-end data warehouse solution designed to digitalize and model complex organizational processes for a company (example case: a Zoo).

The project demonstrates the full lifecycle of modern data engineering: conceptual modelling, schema design, ETL automation, data validation, and analytics-ready warehouse creation.

## Key Features & Demonstrated Skills

- Conceptual and physical data modelling (ER diagrams) for multi-entity business environments
- Implementation of a SQL-based data warehouse (SQLite prototype) with a normalized and analytics-ready structure
- Python-based ETL automation, including data loading, transformation, validation, and quality checks
- Documentation of data lineage and warehouse processes for long-term maintainability
- SQL queries illustrating entity relationships, process flows, and reporting logic
- Example data and data dictionary enabling full reproducibility
- Data quality assessment and migration strategy design

This project simulates real-world data engineering work: building a complete system that integrates business logic, database architecture, and automated data operations.

## Repository Structure

– Operative Datenbank/	# SQL-based warehouse (SQLite) with example data
– 01_Source_data/	# Example CSV files used as input for ETL
– 02_Create_Tables_Zoo.sql	# SQL Scripts for tables creation
– 03_Python_Programs/	# Python scripts for automated ETL, data loading, and validation
– 04_Queries/	# SQL queries illustrating entity relationships and analytics logic
– Aufgabe/	# Original project task and instructions
– Zoo.db	# SQL warehouse prototype containing all modeled tables and sample data. File “Zoo.sqlite” is a SQLite project copy
– Data_Dictionary.xlsx	# Detailed description of tables, attributes, and data types
– ERM-Schema	# Entity-relationship model showing relationships among all parties
– Project_Presentation.pdf	# Presentation: objectives, tools, ERM schema, SQL warehouse, data migration, next steps

## Main Files

- Zoo.db and Zoo.sqbpro (Zoo.db with all the queries) - SQLite-based data warehouse filled with sample data representing all operational processes.
- Data\_Dictionary.xlsx - Documented table descriptions, field meanings, and data types.
- ERM-Schema - Complete entity-relationship diagram showing all participants and relationships.
- Project\_Presentation.pdf - Explains scope, objectives, modelling decisions, ETL workflow, warehouse structure, data quality checks, and recommended next development steps.

## Usage Notes

- SQL queries can be run directly from Zoo.sqbpro file.
- ETL can be reproduced by executing Python scripts inside Python\_Programs (please check that all the csv files and Zoo.db are placed in "C:\Source\_data\")
- Warehouse inspection can be done using DB Browser for SQLite or any compatible SQL tool.