Data Warehousing Guide - Core Concepts - 1/4

qimia.io/en/blog/Data-Warehousing-Guide-Core-Concept

Introduction

Welcome to our blog and to our new tutorial series on the topic of data warehousing! In our previous tutorial series, we focused on the data engineering part of building a data warehouse.

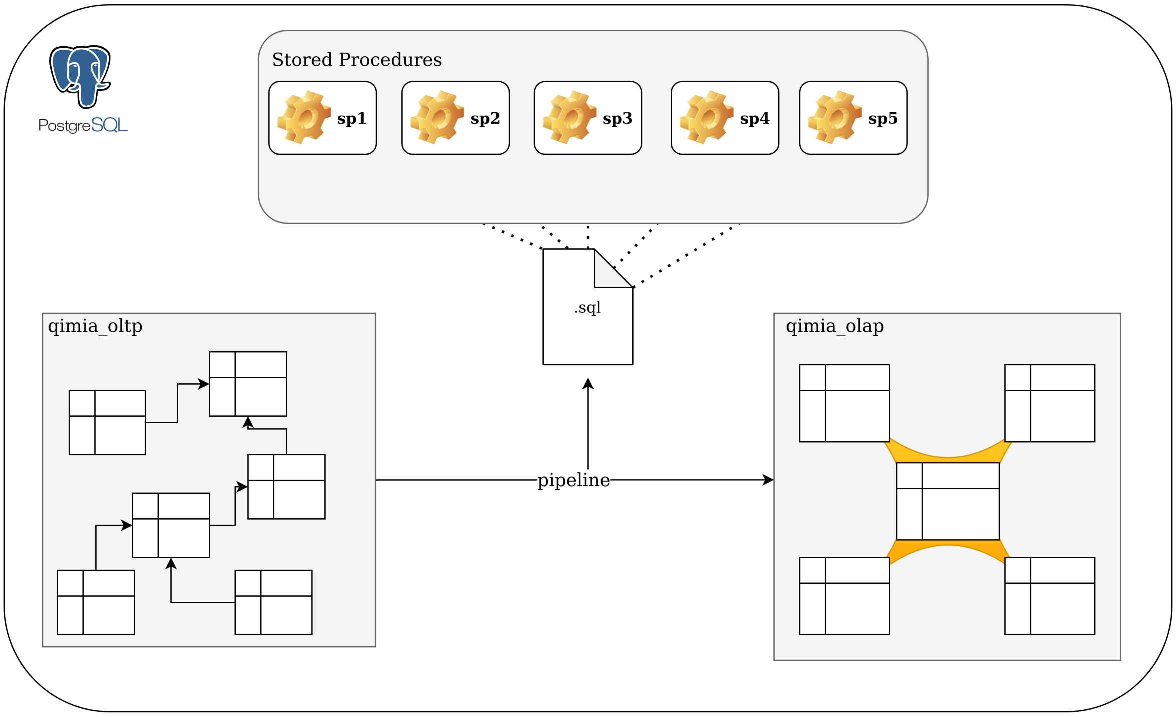
This series will go into more detail on how to design a data warehouse with an existing OLTP (Online Transaction Processing) database as its origin. Be aware that data warehouse design is a huge topic and we cannot possibly cover all of the techniques that exist. Anyway, we are going to teach you some of the most important parts and include examples with artificial data that we provide.

The series will consist of the following parts:

1. Core concepts: star schema, facts, dimensions, data warehouse design process, bus matrix
2. Planning: source schema, defining dimension tables, and fact tables building the bus matrix
3. Design: defining the schema for dimension tables and fact tables
4. Implementation: building the data pipelines in SQL

Target

In this series, we will focus mostly on the data warehousing principles and try to simplify the other aspects as much as possible. Thus, our target architecture including the source system will look like the following.

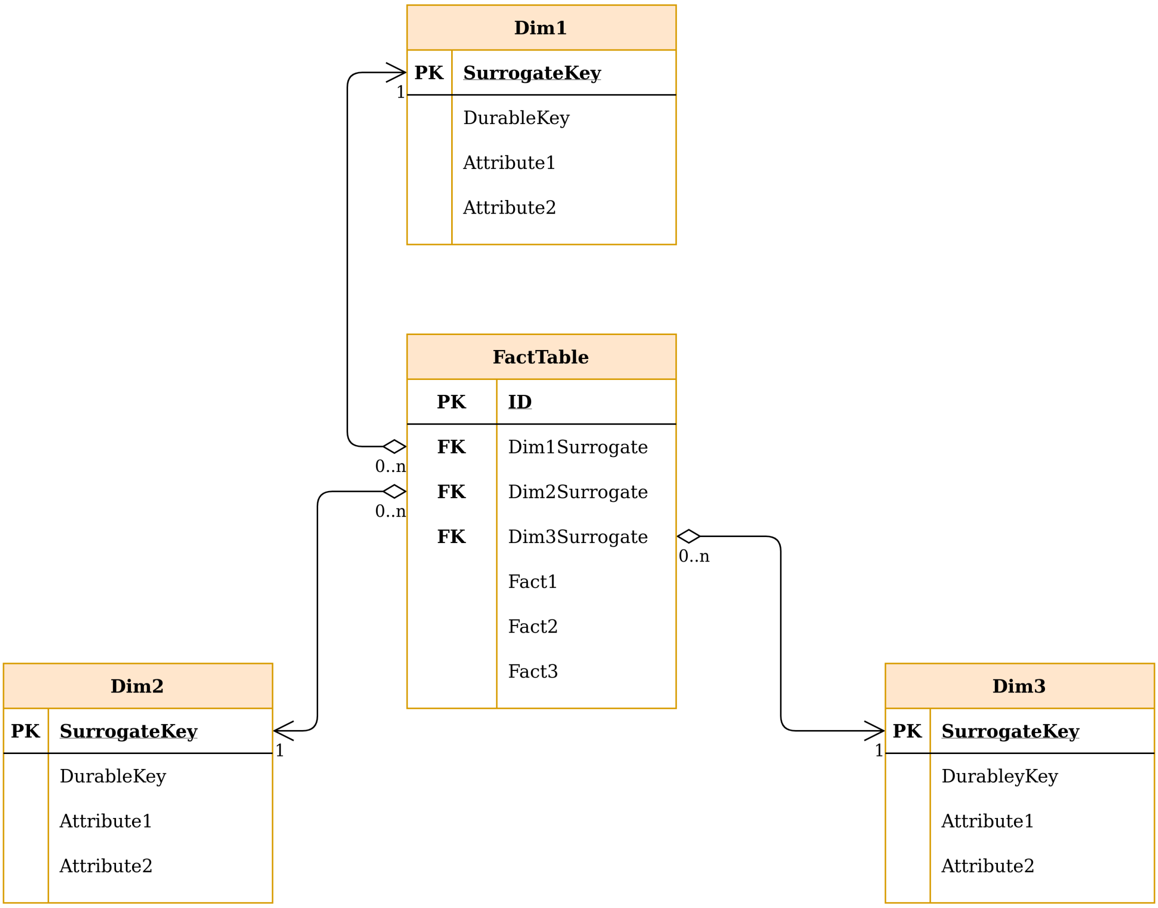


Target Architecture
We will simply use the same postgres database with two different schemas, one containing our operational tables and one containing the tables in the star schema. To transfer the data from the operational schema to the data warehousing schema, we define stored procedures that are executed from an SQL file that acts as the ETL pipeline. If you are searching for a more realistic architecture, make sure to check out our guide for Data Engineering on MS Azure.

1. Star Schema

Star schemas are the core concept of data warehouse design. Unlike the normalized schemas that we use for our operational OLTP databases, star schemas put way less value on reducing redundancy and way more value on performance and simplicity. Another way of thinking about it is that OLTP databases focus on single entries and lean more towards fast writing while OLAP (Online Analytical Processing) databases focus on ranges and lean more towards fast reading.

To enable fast reading for range queries, the star schema organizes the tables so that usually only two tables have to be joined to answer a business question. This is effective because joins are very costly operations. To keep the number of joins low, a star schema will always contain one fact table and two or more dimension Tables that are linked to the fact table through foreign keys but usually not to each other (exception: outrigger dimensions).



Example Star Schema

1.1. Fact Tables

Fact tables are at the center of the star schema and are usually the depiction of a business process. They contain a lot of metrics, also called facts, that are interesting to business analysts because they are key factors of profitability. Other than metrics, fact tables contain foreign keys to the dimension tables. Since fact tables are concerned with processes/ actions like sales, purchases, or shipments they often contain very many rows. To keep the size of the database manageable, descriptive details about the process are mostly extracted to dimensions.

1.2. Dimension Tables

Dimension tables contain data that can enrich the information about the process, typically involving the W-questions. From a business analyst perspective, the dimension attributes will mostly be used for filtering or grouping the process. Unlike fact tables which describe processes, dimension tables describe entities that are mostly static, like products, stores, or customers. Since the dimension tables usually contain quite a manageable amount of rows, they are often enriched with further columns to make the querying more comfortable for business analysts. A good example of this is the date dimension, where it is possible to compute a lot of additional attributes like a holiday, weekday, etc. based on the original value. Furthermore, due to their small size, it might be feasible to replicate the dimension tables among the nodes if you are using distributed databases like AWS Redshift, Azure Synapse, or Google BigQuery.

2. Data Warehousing Process

The process of data warehousing is a delicate topic since it is often tied to the strategy and the processes of the whole company and thus multiple departments. Due to the high amount of contributors and stakeholders needed to build an efficient and integrated data warehouse, it is very important to have a clear process and tool to coordinate and execute the efforts.

2.1. Business Process Steps

There is a four-step process that should be used to cut the huge task of building a data warehouse into smaller, manageable chunks.

1. Select Business Process

A company has many different processes like procurement, manufacturing, or sales. Both the measured facts and the applicable dimensions heavily depend on the business process, so focusing on one business process at a time is inevitable. There are also dimensions that are applicable to multiple business processes. These dimensions should be implemented as conformed dimensions, meaning that they are shared by fact tables so business analysts can perform an integrated analysis along the value chain.

2. Declare the Grain

The grain at which the process is analyzed also has an impact on which dimensions are applicable. In the case of a retail shop, a customer might buy multiple products in the same transaction. If the grain is on the level of the product, it does make sense to connect a product dimension table to the fact table. This is not possible if the grain is at the transaction level. Because it is hardly predictable for the data engineering team which questions business analysts will try to answer with the data warehouse, one should always have a fact table at the most granular/atomic level. Aggregated fact tables speed up the queries, but they also limit the questions that can be answered by the data warehouse. Thus, they should always be the optional add-on to the mandatory atomic fact table.

3. Identify the Dimensions

When it comes to selecting dimensions, the trickiest part is usually to define what qualifies as a standalone dimension and what should be integrated into another dimension. That decision is use-case specific and should be made according to the magnitude of dependency between the tables. For example, if the products offered to depend heavily on the store, the stores and the products should be captured in one dimension. If the products are independent of the stores, they should be captured in two different dimensions.

Dimensions should usually be independent of one another!

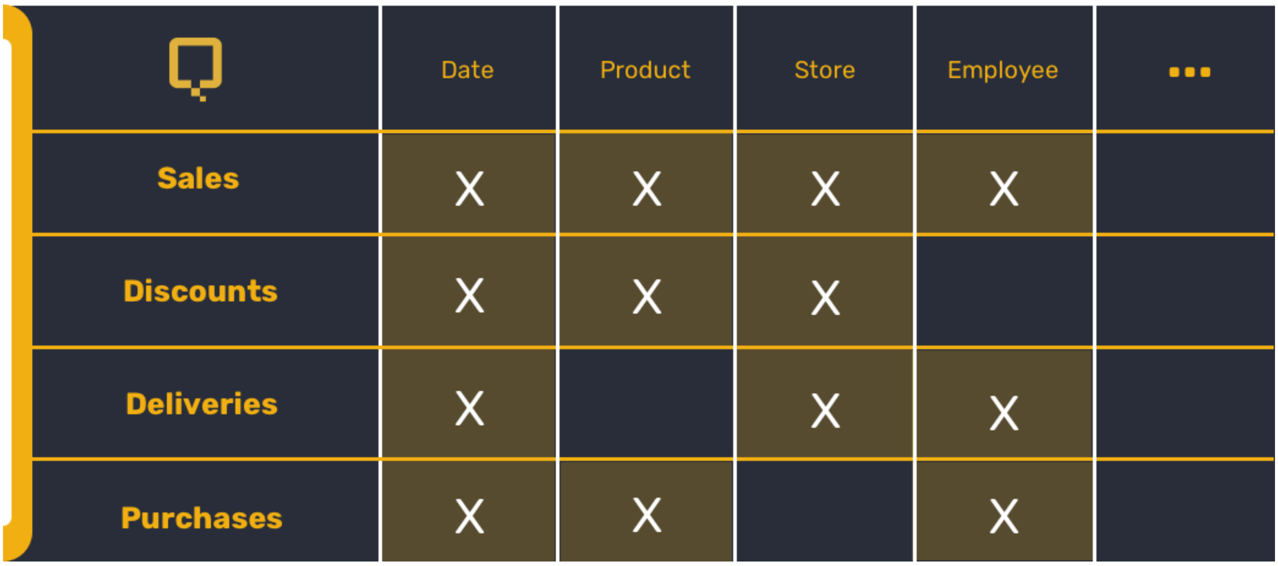
Sometimes, attributes from different dimensions are also thrown into one junk dimension, to reduce the number of overall dimensions.

4. Select Facts

The facts in the fact table are usually metrics that are essential to the success of the company. There are three different types of facts: additive, semi-additive and non-additive. This type determines whether the facts still make sense when they get summed up in the context of an aggregation. E.g. a sales quantity will be additive while a ratio like revenue margin is not additive. Facts should have a one-to-one relationship to the grain of the fact table. Also, facts should be directly linked to the observable event, such as the sales transaction.

2.2. Bus Matrix

The bus matrix is an integral tool both for the planning of efforts and the building of the data warehouse. It usually contains a row for each of the business processes that will result in a fact table. The columns contain all of the dimensions that are applicable to at least one of the business processes. Each cell is then shaded or somehow marked to indicate whether the dimension is applicable to the process. The data engineering team should always focus on building the schema and pipeline for one process at a time. This principle enables an incremental integration of new processes into the existing infrastructure and thus increases acceptance in the business since progress is visible very quickly.



	Date	Product	Store	Employee	...
Sales	X	X	X	X	
Discounts	X	X	X		
Deliveries	X		X	X	
Purchases	X	X		X	

Bus Matrix

Wrapping It Up

In this article, we learned about the basics of data warehousing. We learned about the star schema with its fact- and dimension tables, the data warehousing process, and the bus matrix.

In the next article, we will present the OLTP schema that will be our use case for the series. Also, we will apply the data warehousing process to our OLTP schema to build our very own bus matrix. See you in part2, Data Warehousing Guide - Following the Data Warehousing Process!

Source

- Wiley: The Data Warehouse Toolkit

For the next part of the tutorial click here