# Learned DBMS

## How to adapt Zero-Shot Models to different Database Management Systems?

## Abstract

The main tasks of a database management system (DBMS) are to manage incoming data, organize it, and then to provide ways to modify or retrieve the data. DBMS contain different components that can be learned. For example, an index structure could be trained using machine learning instead of handcrafting it. This offers high potentials to optimize complex structures while avoiding manual effort. Previous approaches (e.g. data-driven learning) are limited as they need to run additional training queries and to retrain a model when looking at unseen data. Zero-shot learning introduces an approach to train machine learning models which generalize to different DBMS components of unseen databases without the need to retrain the model. This poster proposes the idea to adapt the concept of using zero-shot models even among different DBMS.

Figure 1: Examples of different DBMS.

## Problem Statement

When looking at the technology landscape of DBMS, one encounters multiple **different technical implementations**, for example database types, query operators and hash join implementations. The main advantage of zero-shot learning is that it offers a way to generalize trained models to different DBMS components of unseen databases without the need to retrain them. Although these components are trained for one certain DBMS, they are not directly usable from one platform to another. Additionally, not only various DBMS are differently implemented, but also regular database systems evolve over time, so the question arises how to adapt learned zero-shot models to those engineering changes?

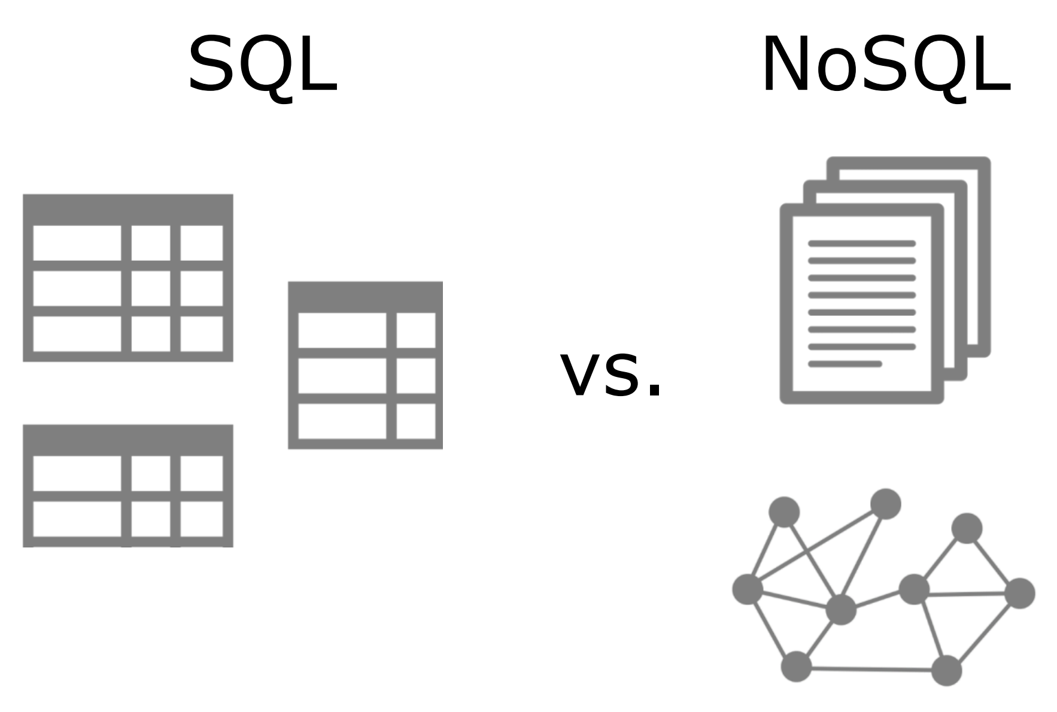

Figure 2: Example of different database types as a technical implementation that is not identical in all DBMS.

## Research Proposal

The goal behind this research proposal is to introduce an approach of using individual engineering features of DBMS to **adapt the initial zero-shot models to evolving DBMS and to generalize among other DBMS**.

## Approach

One would look at extracting the **technical engineering features** of the different DBMS. The zero-shot model could be relearned on those extracted features. For example, zero-shot learning uses graph encodings, which contain the plan operators, predicates and tables etc. as graph nodes. The engineering features could be considered in the graph encoding as well. The zero-shot model could then also be transformed to a **few-shot model** when retraining it with the additional engineering features.

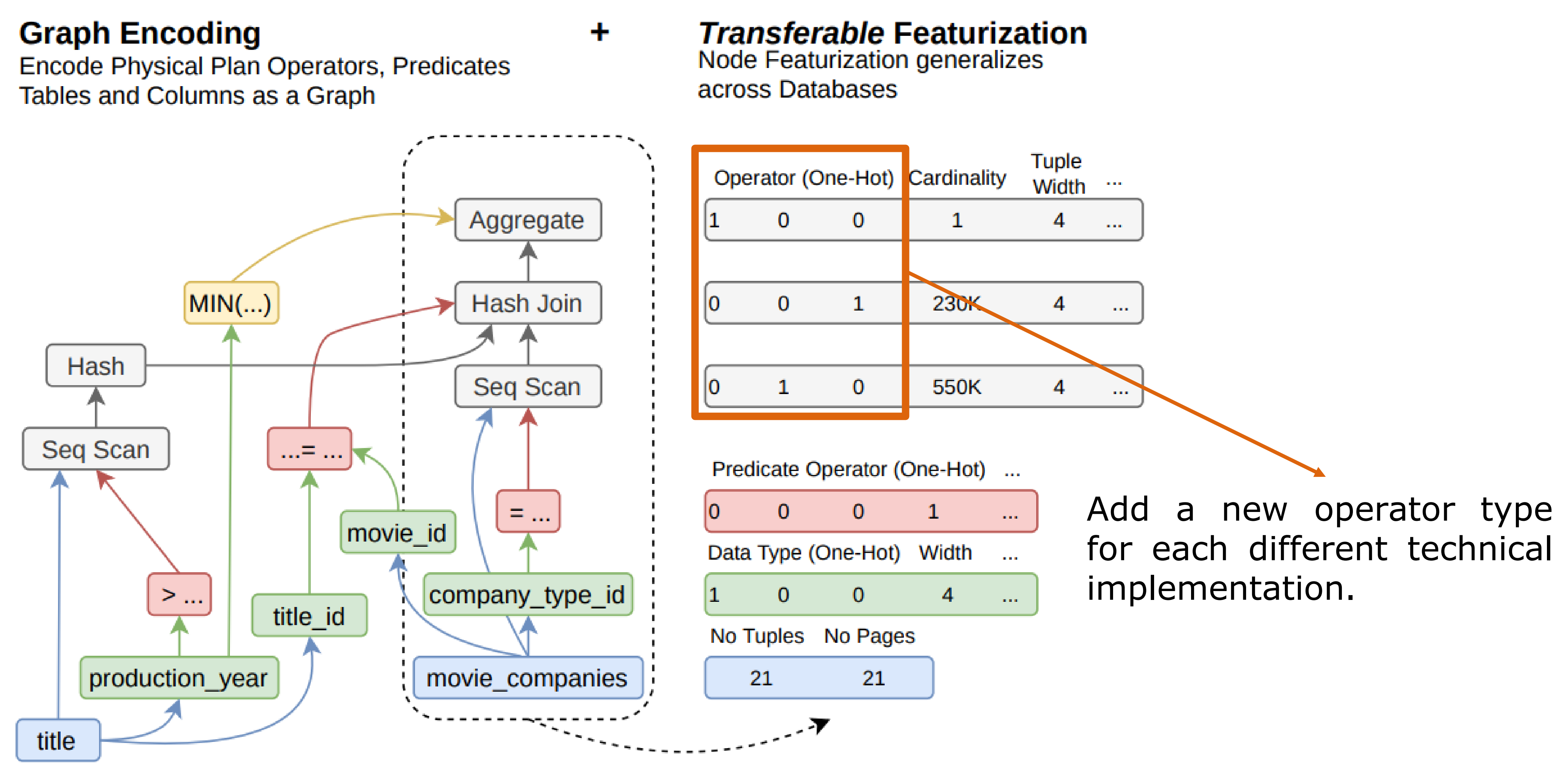

Figure 3: Example of how to extend the graph encodings and their transferable features [1] with technical engineering features.

**Larissa Hoffäller**
**M.Sc. Student IT-Systems Engineering**
**Lecture Series on Database Research (WS 21/22)**

Hasso Plattner Institute, Potsdam, Germany

E-Mail: larissa.hoffaeller@student.hpi.de

**Connection to the Lecture:**
Based on the talk "Learned DBMS Components" by Carsten Binnig, TU Darmstadt.

**Source:**
[1] B. Hilprecht, and C. Binnig. One Model to Rule them All: Towards Zero-Shot Learning for Databases, May 2021.

HPI Hasso Plattner Institut