

Technische Hochschule Brandenburg

IT Security  
Computerscience and Media  
Prof. Dr. Oleg Lobachev

Embedding Generative Pretrained Transformers into PostgreSQL

Bachelor Thesis

Summer semester 2025

March 22, 2025

Mara Schulke – Matr-Nr. 20215853

## Abstract

This thesis explores the integration of Generative Pretrained Transformers (GPT) into PostgreSQL database systems. The research focuses on implementation strategies, performance optimization, and practical applications of this integration.

## Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Problem Statement and Motivation . . . . .	5
1.2	Objectives of the Thesis . . . . .	5
1.3	Research Questions . . . . .	5
1.4	Methodological Approach . . . . .	5
1.5	Structure of the Thesis . . . . .	5
<b>2</b>	<b>Theoretical Foundations</b>	<b>5</b>
2.1	Generative Pretrained Transformers (GPT) . . . . .	5
2.1.1	Architecture and Functionality . . . . .	5
2.1.2	Training and Fine-tuning . . . . .	5
2.1.3	Application Areas . . . . .	5
2.2	PostgreSQL as a Database System . . . . .	5
2.2.1	Architecture of PostgreSQL . . . . .	5
2.2.2	Extension Capabilities . . . . .	5
2.2.3	PostGIS and Other Extensions as Examples . . . . .	5
2.3	Embedding AI Models in Database Systems . . . . .	5
2.3.1	Current Approaches and Solutions . . . . .	5
2.3.2	Technical Challenges . . . . .	5
2.3.3	Benefits of Integration . . . . .	5
<b>3</b>	<b>Conceptual Design of GPT Embedding in PostgreSQL</b>	<b>5</b>
3.1	Requirements Analysis . . . . .	5
3.1.1	Functional Requirements . . . . .	5
3.1.2	Non-functional Requirements . . . . .	5
3.2	Architecture Design . . . . .	5
3.2.1	Interface Design . . . . .	5
3.2.2	Data Model . . . . .	5
3.2.3	Integration into PostgreSQL . . . . .	5
3.3	Technical Implementation Strategies . . . . .	5
3.3.1	Foreign Data Wrapper . . . . .	5
3.3.2	Extension Using C/C++ . . . . .	5
3.3.3	PL/Python or Other Procedural Languages . . . . .	5
3.3.4	Comparison of Approaches . . . . .	5
<b>4</b>	<b>Implementation</b>	<b>5</b>
4.1	Development Environment and Tools . . . . .	5
4.2	Integration of the GPT Model . . . . .	5
4.2.1	Model Selection and Optimization . . . . .	5
4.2.2	API Connection or Local Embedding . . . . .	5
4.3	Development of the PostgreSQL Extension . . . . .	5

4.3.1	SQL Functions for GPT Interactions . . . . .	5
4.3.2	Data Type Conversion and Processing . . . . .	5
4.3.3	Error Handling and Logging . . . . .	5
4.4	Optimization . . . . .	5
4.4.1	Performance Tuning . . . . .	5
4.4.2	Memory Usage . . . . .	5
4.4.3	Parallelization . . . . .	5
<b>5</b>	<b>Evaluation</b>	<b>5</b>
5.1	Test Environment and Methodology . . . . .	5
5.2	Performance Tests . . . . .	5
5.2.1	Latency . . . . .	5
5.2.2	Throughput . . . . .	5
5.2.3	Scalability . . . . .	5
5.3	Use Cases . . . . .	5
5.3.1	Natural Language Queries . . . . .	5
5.3.2	Text Generation Within the Database . . . . .	5
5.3.3	Semantic Search and Text Classification . . . . .	5
5.4	Comparison with Alternative Approaches . . . . .	5
<b>6</b>	<b>Discussion</b>	<b>5</b>
6.1	Interpretation of Results . . . . .	5
6.2	Limitations of the Implementation . . . . .	5
6.3	Ethical and Data Privacy Considerations . . . . .	5
6.4	Potential Future Developments . . . . .	5
<b>7</b>	<b>Summary and Outlook</b>	<b>5</b>
7.1	Summary of Results . . . . .	5
7.2	Addressing the Research Questions . . . . .	5
7.3	Outlook for Future Research and Development . . . . .	5
	<b>References</b>	<b>5</b>
	<b>List of Figures</b>	<b>6</b>
	<b>List of Tables</b>	<b>6</b>
<b>A</b>	<b>Installation Guide</b>	<b>6</b>
<b>B</b>	<b>API Documentation</b>	<b>6</b>
<b>C</b>	<b>Code Examples</b>	<b>6</b>
<b>D</b>	<b>Test Data and Results</b>	<b>6</b>
	<b>List of Figures</b>	

GPT    Generative Pretrained Transformer  
SQL    Structured Query Language  
API    Application Programming Interface



## 1 Introduction

### 1.1 Problem Statement and Motivation

### 1.2 Objectives of the Thesis

### 1.3 Research Questions

### 1.4 Methodological Approach

### 1.5 Structure of the Thesis

## 2 Theoretical Foundations

### 2.1 Generative Pretrained Transformers (GPT)

#### 2.1.1 Architecture and Functionality

#### 2.1.2 Training and Fine-tuning

#### 2.1.3 Application Areas

### 2.2 PostgreSQL as a Database System

#### 2.2.1 Architecture of PostgreSQL

#### 2.2.2 Extension Capabilities

#### 2.2.3 PostGIS and Other Extensions as Examples

### 2.3 Embedding AI Models in Database Systems

#### 2.3.1 Current Approaches and Solutions

#### 2.3.2 Technical Challenges

#### 2.3.3 Benefits of Integration

## 3 Conceptual Design of GPT Embedding in PostgreSQL

### 3.1 Requirements Analysis

#### 3.1.1 Functional Requirements

#### 3.1.2 Non-functional Requirements

### 3.2 Architecture Design

#### 3.2.1 Interface Design

#### 3.2.2 Data Model

#### 3.2.3 Integration into PostgreSQL

### 3.3 Technical Implementation Strategies

#### 3.3.1 Foreign Data Wrapper

#### 3.3.2 Extension Using C/C++

#### 3.3.3 PL/Python or Other Procedural Languages

#### 3.3.4 Comparison of Approaches

## 4 Implementation

5

### 4.1 Development Environment and Tools

### 4.2 Integration of the GPT Model

#### 4.2.1 Model Selection and Optimization

#### 4.2.2 API Connection or Local Embedding

### 4.3 Development of the PostgreSQL Extension

#### 4.3.1 SQL Functions for GPT Interactions

#### 4.3.2 Data Type Conversion and Processing

**List of Figures**

**List of Figures**

**List of Tables**

**List of Tables**

**A Installation Guide**

**B API Documentation**

**C Code Examples**

**D Test Data and Results**