

Bachelorstudiengang  
Informationsmanagement und Unternehmenskommunikation  
Hochschule für angewandte Wissenschaften Neu-Ulm

**BACHELORARBEIT**

Thema

**Die Erstellung und Überprüfung von SQL-Statements mit  
und ohne KI**

Verfasser:	Arthur Sipple
Matr.-Nr.:	263228
Geburtsdatum:	11.04.1998
Erstbetreuer:	Prof. Dr. Thomas Bauer
Zweitbetreuer:	Prof. Dr. Klaus Lang
Thema erhalten:	25.03.2024
Arbeit abgeliefert:	25.07.2024
Sperrvermerk:	Nein
Anlagen:	ZIP- Datei

## **Zusammenfassung**

In der Bachelorarbeit wird die Erstellung von SQL-Statements sowohl manuell als auch durch den Einsatz von Künstlicher Intelligenz (KI) untersucht. Der Fokus liegt auf der Frage, ob KI die Effizienz und Genauigkeit in der SQL-Statement-Generierung verbessern kann. Die Arbeit beginnt mit einer detaillierten Einführung in die Bedeutung und die Grundlagen von SQL in modernen Datenbanksystemen und diskutiert die Rolle von KI in der Datenverarbeitung. Es folgt eine umfassende Literaturrecherche, die die neuesten Entwicklungen und Anwendungen von KI im Kontext von SQL und Datenbankmanagement beleuchtet. Der experimentelle Teil der Arbeit umfasst die Entwicklung einer Datenbankstruktur für eine fiktive Universität, in der SQL-Statements unter Verwendung von KI-Technologien generiert und getestet werden. Es wird eine systematische Analyse der von KI generierten SQL-Statements durchgeführt und ihre Genauigkeit verglichen. In der Diskussion werden die Ergebnisse kritisch bewertet. Abschließend werden die wichtigsten Erkenntnisse zusammengefasst, Forschungsfragen beantwortet und Empfehlungen für zukünftige Forschungen gegeben, um die Integration von KI in Datenbankmanagementsysteme weiter zu verbessern. Die Arbeit hebt hervor, dass KI die Präzision der SQL-Statement-Generierung verbessern und die Geschwindigkeit von Datenbankoperationen steigern kann, während gleichzeitig die menschlichen Fehler reduziert werden.

Schlüsselwörter: Datenbanken, SQL, Künstliche Intelligenz, Test, Datenbankmanagementsystem

## **Abstract**

In the thesis, the creation of SQL statements is examined both manually and through the use of Artificial Intelligence (AI). The focus is on whether AI can improve efficiency and accuracy in SQL statement generation. The work begins with a detailed introduction to the significance and fundamentals of SQL in modern database systems and discusses the role of AI in data processing. This is followed by a comprehensive literature review that highlights the latest developments and applications of AI in the context of SQL and database management. The experimental part of the thesis includes the development of a database structure for a fictional university, where SQL statements are generated and tested using AI technologies. A systematic analysis of the AI-generated SQL statements is conducted, and their accuracy is compared. In the discussion, the results are critically evaluated. In conclusion, the most important findings are

summarized, research questions answered, and recommendations for future research are provided to further improve the integration of AI into database management systems. The thesis emphasizes that AI can enhance the precision of SQL statement generation and increase the speed of database operations, while simultaneously reducing human errors.

Key words: Databases, SQL, Artificial Intelligence, Test, Database Management System

## Inhaltsverzeichnis

<b>1 Einleitung</b> .....	1
<b>1.1 Bedeutung von SQL in modernen Datenbanksystemen</b> .....	1
<b>1.2 Motivation und Relevanz des Themas</b> .....	2
<b>1.3 Zielsetzung der Arbeit</b> .....	3
<b>1.4 Aufbau der Arbeit</b> .....	5
<b>2 Grundlagen</b> .....	6
<b>2.1 Literaturrecherche</b> .....	6
<b>2.2 Grundkonzepte relationaler Datenbanken</b> .....	11
<b>2.3 Künstliche Intelligenz</b> .....	12
<b>2.3.1 Grundkonzepte und Definitionen</b> .....	13
<b>2.3.2 KI in der Datenverarbeitung und Programmierung</b> .....	15
<b>3 Manuelle Erstellung und Überprüfung von SQL-Statements</b> .....	17
<b>3.1 Methoden und Werkzeuge</b> .....	17
<b>3.2 Herausforderungen und gängige Fehlerquellen</b> .....	18
<b>4 Experiment</b> .....	20
<b>4.1 Zielsetzung des Experiments</b> .....	20
<b>4.2 Vorarbeit des Experiments</b> .....	20
<b>4.3 Definition der Fremdschlüsselbeziehungen</b> .....	22
<b>4.4 Erstellung der Datenbank in MySQL</b> .....	23
<b>4.5 Einfügen von Testdaten</b> .....	24
<b>4.6 Schreiben von Testfällen</b> .....	26
<b>4.7 Testen der KIs</b> .....	26
<b>5 Testergebnisse</b> .....	28
<b>6 Diskussion</b> .....	31
<b>6.1 Bewertung der Ergebnisse</b> .....	31
<b>6.2 Kritische Reflexion der KI-Integration in SQL-Prozesse</b> .....	31
<b>7 Zukunftsperspektiven und Potenziale</b> .....	33
<b>7.1 Herausforderungen und Risiken der KI-Integration in Datenbanksysteme</b> .....	33
<b>7.2 Weitere Anwendungsbereiche von Künstlicher Intelligenz</b> .....	34
<b>8 Fazit</b> .....	36
<b>8.1 Zusammenfassung der wichtigsten Erkenntnisse</b> .....	36
<b>8.2 Beantwortung der Forschungsfragen</b> .....	36
<b>8.3 Ausblick und Empfehlungen für zukünftige Forschungen</b> .....	36
<b>Literaturverzeichnis</b> .....	V

## 1 Einleitung

In der heutigen digitalen Welt sind Daten das Herzstück zahlreicher Geschäftsprozesse und Entscheidungssysteme. Die Fähigkeit, diese Daten effizient und präzise zu verwalten, ist für Unternehmen und Organisationen von hoher Relevanz. Structured Query Language (SQL) hat sich dabei als unverzichtbares Werkzeug zur Verwaltung und Abfrage von Datenbanken etabliert. SQL-Statements ermöglichen es, erhebliche Mengen an Daten zu durchsuchen, zu analysieren und zu manipulieren, um wertvolle Aufschlüsse zu erlangen und fundierte Entscheidungen zu treffen.

Mit der rasanten Entwicklung der Künstlichen Intelligenz (KI) eröffnen sich neue Möglichkeiten zur Automatisierung und Optimierung der Erstellung und Überprüfung von SQL-Statements. KI-basierte Systeme versprechen nicht nur eine erhöhte Effizienz, sondern auch eine Verringerung von Fehlerquellen, die bei der manuellen Erstellung von SQL-Befehlen auftreten können. Gleichzeitig stellt sich die Frage, inwieweit menschliche Expertise und kreative Problemlösungsfähigkeiten in diesen Prozess weiterhin eine Rolle spielen.

### 1.1 Bedeutung von SQL in modernen Datenbanksystemen

Structured Query Language (SQL) ist das Fundament moderner relationaler Datenbanksysteme und spielt eine entscheidende Rolle in der Datenverarbeitung. Seit seiner Entwicklung in den 1970er Jahren hat sich SQL als unverzichtbares Instrument für Datenmanagement und Datenmanipulation erwiesen. Die Relevanz von SQL in der heutigen digitalen Wirtschaft wird durch mehrere Schlüsselfaktoren verdeutlicht („SQL“, o. J.):

1. Standardisierung und Portabilität: SQL wird durch ANSI und ISO standardisiert, was eine konsistente Syntax und Funktionalität über verschiedene Datenbanksysteme hinweg gewährleistet. Dies ermöglicht die Portabilität von Anwendungen zwischen verschiedenen Systemen, ohne die Notwendigkeit umfangreicher Code-Neuschreibungen. Die Einheitlichkeit und die breite Akzeptanz von SQL erleichtern die Interoperabilität zwischen verschiedenen Datenbanksystemen und tragen zur systemübergreifenden Effizienz bei (Groff & Weinberg, 1999, S. 12).

2. Effiziente Datenoperationen: SQL optimiert die Ausführung von Datenoperationen wie Abfragen, Updates, Inserts und Deletes. Mechanismen, wie die Bündelung von SQL-Anweisungen, können die Netzwerklasten in Client/Server-Umgebungen signifikant

reduzieren, was die Gesamteffizienz verbessert. Die Auswahl optimaler Ausführungspläne durch das DBMS steigert die Effizienz von SQL-Abfragen (Groff & Weinberg, 1999, S. 32).

3. Flexibilität und Mächtigkeit: SQL unterstützt komplexe Abfragen, Subqueries, Joins und Aggregationsfunktionen, die es Entwicklern ermöglichen, sehr spezifische Daten effizient abzurufen. Diese Flexibilität macht SQL zu einem mächtigen Werkzeug für Datenanalysten und Softwareentwickler, was die Nutzung in vielfältigen Anwendungsfällen ermöglicht (Groff & Weinberg, 1999, S. 14, 185, 194, 214).

4. Umfangreiche Unterstützung und Ressourcenverfügbarkeit: Aufgrund seiner weiten Verbreitung und breiten Akzeptanz, wird SQL von einer Vielzahl von Plattformen und Technologien unterstützt. Entwickler und Datenbankadministratoren haben Zugang zu einer umfangreichen Menge an Dokumentationen, Tools, Gemeinschaftsressourcen und Best Practices. Dies erleichtert das Lernen und die effektive Nutzung von SQL, sowie die Lösung spezifischer Probleme oder die Implementierung komplexer Datenbanklösungen. Die umfangreiche Unterstützung erleichtert auch die Integration von SQL in unterschiedliche Softwareumgebungen, was besonders für komplexe Unternehmensanwendungen und internetbasierte Architekturen entscheidend ist (Groff & Weinberg, 1999, S. 33, 609).

5. Sicherheit und Zugriffskontrolle: SQL bietet eingebaute Mechanismen zur Verwaltung der Datenzugriffskontrolle. Durch die Vergabe und Entziehung von Rechten können Administratoren steuern, wer Zugang zu welchen Daten hat. Dies ist für die Aufrechterhaltung der Datensicherheit und -integrität unerlässlich, besonders in Umgebungen, in denen sensible oder regulierte Informationen verwaltet werden. Diese Sicherheitsfunktionen von SQL helfen, die Compliance mit Datenschutzstandards und Sicherheitsrichtlinien zu gewährleisten (Groff & Weinberg, 1999, S. 303–308). Diese Punkte verdeutlichen, wie SQL als eine robuste, flexible und sichere Sprache die Grundlagen für moderne datengetriebene Systeme bildet.

## **1.2 Motivation und Relevanz des Themas**

Die Überprüfung und Erstellung von SQL-Statements kann ein entscheidender Prozess in der Verwaltung moderner Datenbanksysteme werden, der die Integrität und Leistung dieser Systeme direkt beeinflusst. In Anbetracht der zentralen Bedeutung dieser Aufgaben und der steigenden Komplexität von Datenbankanwendungen in Unternehmen, rückt die Potenzialausschöpfung durch die Nutzung von Künstlicher Intelligenz (KI) in den Fokus.

Besonders interessant ist dabei die Frage, wie genau und effizient KI-Technologien bei der Erstellung und Überprüfung von SQL-Statements agieren können.

Die Motivation für dieses Forschungsthema ergibt sich aus der wachsenden Präsenz von KI in zahlreichen Geschäftsprozessen und der Notwendigkeit, ihre Anwendbarkeit und Effektivität in spezifischen technischen Domänen zu verstehen. (Dr. habil. Nizar Abdelkafi et al., o. J., S. 12–15). Die Integration von KI-Methoden zur Automatisierung und Verbesserung der SQL-Statement-Erstellung und -Überprüfung ist ein hochaktuelles Thema, das sowohl technische als auch betriebswirtschaftliche Innovationen verspricht. Dieses Thema untersucht, inwieweit KI die Präzision und Effizienz in Prozessen steigern kann, die traditionell eine hohe Fehleranfälligkeit aufweisen und oft arbeitsintensiv sind.

Die Relevanz dieses Forschungsvorhabens liegt in der Möglichkeit, durch den Einsatz von KI-Technologien die Genauigkeit der SQL-Statement-Generierung und Fehlererkennung zu erhöhen. Dies könnte nicht nur zu einer Verringerung von menschlichen Fehlern führen, sondern auch die Geschwindigkeit und Effizienz von Datenbankoperationen verbessern. Unternehmen könnten dadurch erheblich von reduzierten Wartungskosten und einer verbesserten Datenqualität profitieren.

Weiterhin bietet die Untersuchung von KI-gestützten Methoden zur SQL-Überprüfung und -Erstellung die Möglichkeit, tiefere Einblicke in die Grenzen und Potenziale dieser Technologien zu gewinnen. Es ist essenziell zu verstehen, in welchen Szenarien KI zuverlässig funktioniert und wo möglicherweise Risiken oder Herausforderungen bestehen. Die Ergebnisse dieser Forschung können dazu beitragen, zukünftige Richtungen für die Entwicklung von Datenbankmanagementsystemen zu bestimmen und Informationen darauf zu geben, wie KI effektiv in bestehende Technologieinfrastrukturen integriert werden kann (Durkin, 2023).

Angesichts der rasanten Fortschritte in der KI-Forschung und der zentralen Rolle von Daten in der heutigen Wirtschaft ist dieses Thema von strategischer Bedeutung für Unternehmen, die darauf abzielen, ihre Datenverarbeitungsprozesse zu optimieren und zukunftssicher zu gestalten (Kapelle, o. J.).

### **1.3 Zielsetzung der Arbeit**

Das Ziel dieser Bachelorarbeit ist es, die Effektivität von den KI-Modellen ChatGPT und text2SQL.ai bei der Erstellung und Überprüfung von SQL-Statements zu untersuchen. Dabei

soll festgestellt werden, ob und wie gut eine KI in der Lage ist, korrekte SQL-Statements zu generieren und zu überprüfen.

Detaillierte Zielsetzung

1. Untersuchung der Korrektheit von SQL-Statements:

- Der erste Schritt besteht darin, die generierten SQL-Statements auf ihre syntaktische und logische Korrektheit zu überprüfen. Dies schließt die Prüfung ein, ob die Abfragen die richtigen Datenbankstrukturen (Tabellen, Spalten) verwenden und ob die Abfragen korrekte Ergebnisse liefern.

2. Fehleranalyse und -kategorisierung:

Die Bachelorarbeit wird die Fehler, die bei der Generierung der SQL-Statements auftreten, systematisch kategorisieren. Dies beinhaltet:

- Syntaktische Fehler: Fehler, die durch falsche Syntax entstehen (z.B. fehlende Kommas, falsche Schlüsselwörter).
- Logische Fehler: Fehler, die durch eine falsche logische Struktur der Abfrage entstehen (z.B. falsche Join-Bedingungen, inkorrekte WHERE-Klauseln).
- Zu viel Information: Fehler, die durch eine falsche Bedeutung der Abfrage entstehen (z.B. Verwendung falscher Tabellen oder Spalten). Diese Fehler beinhalten auch das Ausgeben von Spalten, die nicht gefragt wurden, oder das Hinzufügen von ORDER BY-Klauseln, die in der ursprünglichen Anfrage nicht verlangt wurden. Diese unnötigen zusätzlichen Informationen können die Abfrageergebnisse unnötig verkomplizieren und den Fokus von den tatsächlich geforderten Daten ablenken.

Durch die Beantwortung dieser Fragen soll ein umfassendes Verständnis dafür entwickelt werden, wie gut KI-Modelle in der Lage sind, SQL-Statements korrekt zu generieren. Dies schließt die Identifikation von Stärken und Schwächen der KI-Modelle ein. Ein tieferes Verständnis der Leistungsfähigkeit dieser Modelle ermöglicht es, gezielt Empfehlungen für die weitere Forschung und Entwicklung zu erarbeiten.



## 1.4 Aufbau der Arbeit

Der Lesbarkeit wegen wird in der vorliegenden Bachelorarbeit, wie gewohnt die männliche Sprechform bei personenbezogenen Begrifflichkeiten verwendet. Dies soll jedoch keine Benachteiligung des weiblichen Geschlechts andeuten, sondern soll im Sinne der sprachlichen Vereinfachung als geschlechtsneutral zu verstehen sein. In diesem Kapitel wird der Aufbau der Bachelorarbeit beschrieben, um dem Leser eine strukturierte Übersicht zu geben und den inhaltlichen Verlauf der Arbeit darzustellen.

Die Einleitung bietet einen Überblick über die Bedeutung von SQL in modernen Datenbanksystemen und die Motivation sowie Relevanz des Themas. Zudem werden die Zielsetzungen der Arbeit definiert. Das zweite Kapitel widmet sich den theoretischen Grundlagen. Es umfasst eine umfassende Literaturrecherche und eine Einführung in relationale Datenbanksysteme sowie deren Grundkonzepte. Ein weiterer Schwerpunkt liegt auf der Künstlichen Intelligenz (KI) und deren Anwendung in der Datenverarbeitung und Programmierung.

Das dritte Kapitel beschreibt die Methoden und Werkzeuge zur manuellen Erstellung und Überprüfung von SQL-Statements. Hier werden gängige Herausforderungen und typische Fehlerquellen analysiert und durch Fallbeispiele verdeutlicht. Im vierten Kapitel wird das Experiment vorgestellt. Dies beinhaltet die Zielsetzung, die Vorarbeiten, die Erstellung der Datenbank in MySQL, das Einfügen von Testdaten und die Definition der Fremdschlüsselbeziehungen. Es wird auch die Erstellung und Durchführung von Testfällen beschrieben, um die Genauigkeit und Effizienz der SQL-Statements zu bewerten. Das fünfte Kapitel umfasst die Methodik der vergleichenden Analyse. Hier werden die Effizienz und Genauigkeit der SQL-Erstellung sowie die Fehlererkennung und -korrektur analysiert. Zudem werden Anwendungsbeispiele und Fallstudien präsentiert.

In der Diskussion werden die Ergebnisse der Arbeit bewertet und eine kritische Reflexion der KI-Integration in SQL-Prozesse vorgenommen. Es werden auch die Grenzen der aktuellen KI-Technologien aufgezeigt. Das siebte Kapitel beschäftigt sich mit den Weiterentwicklungen der KI-Technologien und möglichen neuen Anwendungsbereichen. Der Einfluss auf die Datenbankverwaltung und -sicherheit wird ebenfalls thematisiert.

Im abschließenden Kapitel werden die wichtigsten Erkenntnisse zusammengefasst, die Forschungsfragen beantwortet und ein Ausblick sowie Empfehlungen für zukünftige Forschungen gegeben.

## 2 Grundlagen

### 2.1 Literaturrecherche

Die automatisierte Erstellung und Überprüfung von SQL-Abfragen mithilfe von KI, insbesondere großen Sprachmodellen wie ChatGPT, stellt eine innovative Schnittstelle zwischen natürlicher Sprachverarbeitung und Datenbankverwaltung dar. Dieser Bereich hat aufgrund seines Potenzials, Datenbankinteraktionen zu vereinfachen und sie auch für nicht-technische Benutzer zugänglich zu machen, erhebliches Interesse geweckt. Da dieses Gebiet jedoch noch relativ neu ist, befindet sich die vorhandene wissenschaftliche Literatur noch im Entstehen und bestehende Studien konzentrieren sich oft eher auf Machbarkeitsnachweise als auf umfassende Bewertungen.

In einem Paper wird beschrieben, wie eine auf ChatGPT basierende Zero-Shot-Text-to-SQL-Methode entwickelt wurde. Text-to-SQL bezieht sich auf den Prozess der Transformation von natürlichen Sprachfragen in SQL-Abfragen, mit dem Ziel, Benutzern zu ermöglichen, mit Datenbanken zu interagieren, ohne die SQL-Syntax kennen zu müssen, und so den Zugang zu Daten zu erleichtern. Zero-Shot-Lernen wiederum bedeutet, dass Modelle darauf trainiert werden, Aufgaben zu erfüllen, ohne explizite, aufgabenspezifische Trainingsbeispiele zu erhalten. Im Kontext von Text-to-SQL versuchen Zero-Shot-Modelle wie ChatGPT, genaue SQL-Abfragen allein auf Basis der Eingabeaufforderung zu generieren, ohne ähnliche Beispiele während des Trainings gesehen zu haben (Dipl.-Ing Luber, 2022).

Die von Xuemei Dong und Kollegen beschriebene Methode, genannt C3, integriert drei Schlüsselkomponenten: Clear Prompting (CP), Calibration with Hints (CH) und Consistent Output (CO). Der C3-Ansatz hat eine neue Benchmark in der Zero-Shot-Text-to-SQL-Leistung gesetzt und erzielt eine Ausführungsgenauigkeit von 82,3 % auf dem Spider-Datensatz, einem weit verbreiteten Benchmark zur Bewertung von Text-to-SQL-Systemen. Dies zeigt das Potenzial von ChatGPT zur Generierung von SQL-Abfragen in einem Zero-Shot-Kontext.

#### **Clear Prompting (CP)**

Clear Prompting (CP) bezieht sich auf die Technik, klare und präzise Anweisungen an das Modell zu geben, um sicherzustellen, dass die generierten SQL-Abfragen den Erwartungen entsprechen. Durch die Bereitstellung strukturierter und spezifischer Prompts kann das Modell besser verstehen, welche Informationen abgefragt werden sollen und wie diese in

eine korrekte SQL-Abfrage umzusetzen sind. Dies reduziert die Wahrscheinlichkeit von Missverständnissen und verbessert die Genauigkeit der generierten Abfragen.

**Beispiel:** Anstatt das Modell mit einer vagen Anfrage wie "Finde die Daten der Verkäufer" zu füttern, wird eine klar formulierte Anfrage wie "Liste die Namen der Verkäufer auf, die in New York arbeiten" verwendet.

### **Calibration with Hints (CH)**

Calibration with Hints (CH) beinhaltet das Geben von zusätzlichen Hinweisen oder Kontextinformationen während des Promptings, um die Generierung von SQL-Abfragen weiter zu verbessern. Diese Hinweise können spezifische Details über die Datenbankstruktur oder Beispielabfragen umfassen, die dem Modell helfen, die richtige Syntax und Logik anzuwenden. Durch diese Kalibrierung wird die Konsistenz und Präzision der generierten Abfragen erhöht.

**Beispiel:** Wenn das Modell eine komplexe Abfrage generieren soll, kann man zusätzliche Informationen wie "Verwende eine JOIN-Operation, um Daten aus den Tabellen VENDOR und SALE zu kombinieren" geben. Dies hilft dem Modell, die richtige Abfrage zu erstellen.

### **Consistent Output (CO)**

Consistent Output (CO) zielt darauf ab, die Konsistenz der generierten SQL-Abfragen sicherzustellen, indem wiederholbare und vorhersagbare Ergebnisse produziert werden. Dies kann durch die Implementierung von Techniken erreicht werden, die die Ausgabe des Modells stabilisieren und zufällige Abweichungen minimieren. Eine konsistente Ausgabe ist besonders wichtig, um sicherzustellen, dass die generierten Abfragen zuverlässig und für den praktischen Einsatz geeignet sind.

**Beispiel:** Um sicherzustellen, dass das Modell bei jeder Anfrage ähnliche Abfragen erzeugt, kann man es darauf trainieren, spezifische Muster zu verwenden, wie die regelmäßige Nutzung der SELECT DISTINCT-Klausel, um doppelte Einträge zu vermeiden.

Eine der Hauptbeschränkungen, die in der Forschung von Dong et al. identifiziert wurden, ist, dass die zur Bewertung verwendeten SQL-Abfragen möglicherweise nicht komplex genug sind, um die Fähigkeiten von ChatGPT in realen Szenarien umfassend zu testen. Beispielsweise wurden einfache Abfragen wie "Welche Staaten haben sowohl Besitzer als auch Fachleute, die dort leben?" verwendet, was nicht ausreichend ist, um die Leistungsfähigkeit des Modells bei komplexeren SQL-Abfragen zu bewerten. Dies bedeutet, dass das wahre Potenzial und die Einschränkungen von ChatGPT bei der Generierung von

SQL-Abfragen für komplexere Datenbankstrukturen und Abfragen nicht vollständig erforscht wurden.

Zusätzlich weist die Studie auf inhärente Verzerrungen in der Ausgabe von ChatGPT und Inkonsistenzen aufgrund der Zufälligkeit im Generierungsprozess des Modells hin. Trotz der Einführung von Kalibrierungstechniken zur Minderung dieser Probleme bleiben einige Verzerrungen und Inkonsistenzen bestehen. Beispielsweise tendiert ChatGPT dazu, zusätzliche Spalten in die Ausgabe-SQL-Abfragen aufzunehmen, die nicht explizit in der Anfrage gefordert wurden, was die Genauigkeit der Abfragen beeinträchtigen kann

(Dong et al., 2023, S. 3–6).

Eine weitere umfassende Untersuchung der Text-to-SQL-Fähigkeiten von Sprachmodellen wurde im Paper "Battle of the Large Language Models: Dolly vs LLaMA vs Vicuna vs Guanaco vs Bard vs ChatGPT - A Text-to-SQL Parsing Comparison" durchgeführt. In dieser Studie wurden sechs große Sprachmodelle (Dolly, LLaMA, Vicuna, Guanaco, Bard und ChatGPT) hinsichtlich ihrer Text-to-SQL-Leistung auf neun Benchmark-Datensätzen verglichen. Die Ergebnisse zeigen, dass open source Modelle, wie Dolly und Vicuna in der Regel schlechter abschneiden als closed source Modelle wie GPT-3.5. Dies unterstreicht die Notwendigkeit weiterer Forschungen, um die Leistungslücke zwischen open source und closed source Modellen zu schließen.

Ein interessantes Ergebnis dieser Studie ist, dass die meisten Modelle zwar syntaktisch gültige SQL-Abfragen generieren können, jedoch oft Schwierigkeiten haben, semantisch korrekte Abfragen zu erstellen. Zum Beispiel erzeugte das Modell Guanaco auf die Anfrage "Gebe mir die Homepage von PVLDB zurück", direkt die Antwort "Die Website ist <https://www.vldb.org/pvldb>", anstatt eine SQL-Abfrage zu generieren, was die Notwendigkeit einer besseren kontextuellen Anpassung der Modelle anzeigt.

Die Studie zeigt auch, dass die Effektivität von Prompting-Strategien stark vom Stil der Prompts abhängt. Während die Informal Schema (IS) Strategie für GPT-3.5 und Bard effektiv ist, zeigt sich, dass Modelle wie Dolly und LLaMA bessere Ergebnisse mit der Select 3 (S3) Strategie erzielen.

Die Select 3 (S3) Strategie beinhaltet drei Beispielzeilen für jede Tabelle in der Datenbank. Diese zusätzlichen Informationen sollen konkrete Beispiele der in den Tabellen enthaltenen Daten liefern und somit das Schema ergänzen. Diese Strategie hilft dem Modell, sich ein besseres Bild von der Struktur und dem Inhalt der Datenbank zu machen, was zu präziseren und relevanteren Abfragen führt (Sun et al., 2023).

Eine dritte relevante Arbeit ist die von Adithya Bhaskar et al. im Paper "Benchmarking and Improving Text-to-SQL Generation under Ambiguity". Diese Studie befasst sich mit der Problematik der Mehrdeutigkeit in natürlichen Sprachabfragen und stellt den AmbiQT-Benchmark vor, der über 3000 Beispiele umfasst, bei denen jede Textanfrage aufgrund lexikalischer und/oder struktureller Mehrdeutigkeit in zwei plausible SQL-Abfragen übersetzt werden kann. Die Studie zeigt, dass bestehende Text-to-SQL-Systeme Schwierigkeiten haben, mit dieser Mehrdeutigkeit umzugehen, und schlägt einen neuen Decoding-Algorithmus namens LogicalBeam vor, der signifikante Verbesserungen bei der Handhabung mehrdeutiger Anfragen zeigt (Bhaskar, Tomar, Sathe, & Sarawagi, 2023, S. 1–8).

In einer weiteren Arbeit, "Verification of Relational Database Languages Codes Generated by ChatGPT", wird die Fähigkeit von ChatGPT zur Generierung korrekter SQL-Abfragen untersucht. Diese Studie überprüft SQL-Abfragen, die von ChatGPT generiert wurden, auf ihre Übereinstimmung mit relationalen Algebra Operatoren. Anhand von Beispielen wie der Abfrage "Liste die Namen der Anbieter, die sich in Bangkok befinden" zeigt die Studie, dass ChatGPT in der Lage ist, korrekte SQL-Abfragen zu generieren:

- **SQL:** SELECT VNAME FROM VENDOR WHERE LOCATION = 'Bangkok';

Weitere Abfragen aus dieser Studie beinhalten:

- **Abfrage:** Liste den Namen, den Artikel und die Menge der Lieferungen von Anbietern, die sich in Bangkok befinden.
  - **SQL:** SELECT V.VNAME, I.INAME, S.AMOUNT FROM VENDOR V JOIN SALE S ON V.V# = S.V# JOIN ITEM I ON S.I# = I.# WHERE V.LOCATION = 'Bangkok';
- **Abfrage:** Liste die Einträge von Anbietern, die sich in Bangkok oder Paris befinden.
  - **SQL:** SELECT \* FROM VENDOR WHERE LOCATION = 'Bangkok' UNION SELECT \* FROM VENDOR WHERE LOCATION = 'Paris';
- **Abfrage:** Liste die Anbieter, die sowohl I1 als auch I2 liefern.
  - **SQL:** SELECT DISTINCT s1.V# FROM SALE s1 WHERE s1.I# = 'I1' AND EXISTS (SELECT 1 FROM SALE s2 WHERE s2.V# = s1.V# AND s2.I# = 'I2');

Diese Beispiele verdeutlichen, dass ChatGPT die grundlegenden SQL-Operationen korrekt umsetzen kann. (Pornphol & Chittayasothorn, 2023, S. 18–21)

Weitere Forschung zu verwandten Themen bezieht sich zum Beispiel darauf, mit welcher Methodik Modelle trainiert werden sollten. Die Arbeit "A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions" von Qin et.al., bietet eine umfassende Übersicht über die Entwicklungen und Herausforderungen im Bereich der Text-to-SQL-Generierung.

Text-to-SQL-Parsing zielt darauf ab, natürliche Sprachfragen in entsprechende SQL-Abfragen zu übersetzen, die gegen relationale Datenbanken ausgeführt werden können. Frühe Systeme basierten stark auf manueller Codierung und Interaktion mit Benutzern, was zu hohen Kosten und eingeschränkter Flexibilität führte. Mit dem Aufkommen tiefer neuronaler Netzwerke und großer vortrainierter Sprachmodelle (PLMs) hat sich das Text-to-SQL-Parsing jedoch erheblich weiterentwickelt.

Ein wichtiger Aspekt zur Verbesserung von Sprachmodellen für die Text-to-SQL-Generierung ist die effektive Tokenisierung von Wörtern. Die Tokenisierung teilt den Text in kleinere Einheiten (Tokens) auf, die das Modell verarbeiten kann. Eine präzise Tokenisierung ist entscheidend, um die semantische Bedeutung und Struktur des Textes korrekt zu erfassen. Besonders herausfordernd ist dies bei der Verarbeitung von Fachbegriffen und komplexen Datenbankstrukturen, wo eine ungenaue Tokenisierung zu Missverständnissen und Fehlern in der SQL-Abfrage führen kann. Moderne Modelle nutzen oft Byte-Pair Encoding (BPE) oder WordPiece Tokenisierung, um eine feinere und kontextabhängigere Zerlegung von Texten zu erreichen.

Die Bewertung von Text-to-SQL-Parsers erfolgt hauptsächlich durch den Vergleich der generierten SQL-Abfragen mit den korrekten Antworten. Hierbei werden Metriken wie die "Exact Set Match Accuracy" (EM) und die "Execution Accuracy" (EX) verwendet. Die Arbeit stellt auch verschiedene Datensätze vor, die für das Training und die Evaluierung von Text-to-SQL-Parsers genutzt werden, wie z.B. WikiSQL, Spider und SParC. Diese Datensätze variieren hinsichtlich ihrer Komplexität und Domänenvielfalt, was unterschiedliche Anforderungen an die Generalisierungsfähigkeiten der Modelle stellt (Qin et al., 2022, S. 1-5,16).

Die aktuelle Forschung konzentriert sich oft auf spezifische Datensätze wie Spider, die, obwohl umfassend, nicht das gesamte Spektrum möglicher SQL-Abfragekomplexitäten in diversen realen Anwendungen abdecken. Daher besteht Bedarf an umfassenderem Benchmarking gegen eine breitere Vielfalt von SQL-Abfragen, einschließlich solcher, die verschachtelte Abfragen, mehrere Joins und erweiterte SQL-Funktionen beinhalten.

Die Anwendung dieser Modelle auf reale Datenbanken und Anwendungsfälle würde wertvolle Einblicke in ihre praktische Nützlichkeit und Einschränkungen bieten. Fallstudien

mit komplexen Unternehmensdatenbanken könnten helfen, die Robustheit dieser Modelle zu verstehen. Zukünftige Forschung sollte zudem die nahtlose Integration von KI-generierten SQL-Abfragen mit bestehenden Datenbankmanagementsystemen (DBMS) untersuchen, um sicherzustellen, dass die generierten Abfragen in Bezug auf Leistung und Sicherheit optimiert sind.

Der Einsatz von KI, insbesondere großen Sprachmodellen wie ChatGPT, zur Generierung und Überprüfung von SQL-Abfragen ist ein vielversprechendes, Feld. Die aktuelle Forschung bietet eine solide Grundlage, hebt jedoch auch mehrere Bereiche hervor, die weiterer Untersuchung bedürfen. Während sich diese Technologie weiterentwickelt, birgt sie das Potenzial, die Art und Weise, wie der Umgang mit Datenbanken gepflegt wird, zu revolutionieren und Daten für ein breiteres Publikum zugänglicher und nutzbarer zu machen.

## 2.2 Grundkonzepte relationaler Datenbanken

Relationale Datenbanken basieren auf dem relationalen Modell, das von Edgar F. Codd in den 1970er Jahren entwickelt wurde und nutzen Tabellen (Relationen) zur Darstellung von Daten und deren Beziehungen. Die Grundkonzepte relationaler Datenbanken umfassen mehrere Schlüsselaspekte, die für die effiziente Datenverwaltung und -abfrage entscheidend sind:

1. **Tabellen (Relationen):** Das zentrale Element jeder relationalen Datenbank. Eine Tabelle besteht aus Zeilen und Spalten, wobei jede Zeile einen Datensatz (Tupel) repräsentiert und jede Spalte eine bestimmte Eigenschaft (Attribut) dieser Datensätze beschreibt.
2. **Primärschlüssel (Primary Key):** Jede Tabelle sollte einen Primärschlüssel haben, der jede Zeile eindeutig identifiziert, um Datenredundanz und Integritätsprobleme zu verhindern. Ein Primärschlüssel kann aus einem oder mehreren Feldern bestehen, sofern die Kombination der Felder eindeutig ist.
3. **Fremdschlüssel (Foreign Key):** Ein Fremdschlüssel ist ein Attribut oder eine Gruppe von Attributen in einer Tabelle, die auf den Primärschlüssel einer anderen Tabelle verweisen. Fremdschlüssel sind entscheidend für die Definition von Beziehungen zwischen Tabellen und ermöglichen relationale Operationen wie Joins.
4. **Normalisierung:** Ein Prozess zur Strukturierung einer Datenbank, um Redundanz zu reduzieren und die Datenintegrität zu verbessern. Normalisierung beinhaltet die

Aufteilung einer großen Tabelle in kleinere Tabellen und das Definieren von Beziehungen zwischen ihnen.

5. **Integritätsregeln:** Diese Regeln helfen, die Genauigkeit und Verlässlichkeit der in der Datenbank gespeicherten Informationen zu gewährleisten. Zu diesen Regeln gehören die Entity-Integrity-Regel, die sicherstellt, dass jeder Primärschlüssel eindeutig und nicht null ist, sowie die Referential-Integrity-Regel, die die Gültigkeit jeder Fremdschlüsselreferenz sichert. (Codd, 1970)

Künstliche Intelligenz (KI) kann in mehreren dieser Grundkonzepte nützlich sein, insbesondere bei der Automatisierung und Optimierung von Datenbankoperationen. KI-Technologien können beispielsweise verwendet werden, um die Integrität von Daten zu überwachen, Muster in großen Datenmengen zu erkennen und automatisch Optimierungsvorschläge für Datenbankabfragen zu generieren. Darüber hinaus kann KI in der Normalisierung eingesetzt werden, um Datenstrukturen effizienter zu organisieren und Redundanzen zu identifizieren, die menschlichen Analysten möglicherweise entgehen.

In dieser Bachelorarbeit liegt der Schwerpunkt auf der Untersuchung, wie genau KI bei der Erstellung und Überprüfung von SQL-Statements sein kann. Hierbei wird exploriert, inwieweit KI-gestützte Systeme in der Lage sind, korrekte und effiziente SQL-Statements zu generieren und bestehende Statements auf Fehler zu überprüfen. Dieser Fokus trägt dazu bei, das Potenzial von KI in der praktischen Anwendung innerhalb relationaler Datenbanksysteme zu bewerten und liefert Einsichten, die für die zukünftige Integration von KI-Technologien in Datenbankmanagementsysteme entscheidend sein können.

### 2.3 Künstliche Intelligenz

Künstliche Intelligenz (KI) stellt ein multidisziplinäres Forschungsfeld innerhalb der Computerwissenschaften dar, welches das Ziel verfolgt, technologische Systeme zu entwickeln, die fähig sind, Aufgaben zu bewältigen, die traditionell der menschlichen Intelligenz vorbehalten sind. Dies schließt Fähigkeiten ein wie das Lernen, Problemlösen, perzeptive Wahrnehmung und die Verarbeitung natürlicher Sprache. Der Begriff "Künstliche Intelligenz" wurde 1956 von John McCarthy geprägt und hat seitdem eine rasante Entwicklung durchlaufen. Heute ist KI ein zentraler Bestandteil moderner Technologien und beeinflusst zahlreiche Branchen und Lebensbereiche. (Nilsson, 2009, S. 13)



### 2.3.1 Grundkonzepte und Definitionen

Künstliche Intelligenz (KI) ist ein weites Feld, das zahlreiche Technologien und Methoden umfasst, die darauf abzielen, Maschinen und Computersysteme so zu gestalten, dass sie menschenähnliche Intelligenz zeigen. Einige der grundlegendsten und einflussreichsten Konzepte und Technologien in der KI sind Maschinelles Lernen, Deep Learning, neuronale Netze, natürliche Sprachverarbeitung und Computer Vision. Diese Technologien sind von entscheidender Bedeutung in der modernen KI und haben jeweils spezifische Anwendungsbereiche und Eigenschaften.

**Maschinelles Lernen (ML)** ist eine Kerntechnologie der KI, die es Computern erlaubt, aus Daten zu lernen und Prognosen zu erstellen oder Entscheidungen zu treffen, ohne explizit programmiert zu werden. ML-Algorithmen erkennen Muster in großen Datensätzen und nutzen diese Muster, um zukünftige Daten zu analysieren und zu interpretieren („Was ist Machine Learning?“, o. J.). Es gibt verschiedene Ansätze im maschinellen Lernen, darunter überwachtes Lernen, unüberwachtes Lernen und bestärkendes Lernen. Beim überwachten Lernen wird das Modell mit einem Datensatz geschult, der Eingabedaten und die entsprechenden richtigen Ausgaben enthält. Das Ziel ist es, eine Funktion zu lernen, die neue Eingaben korrekt vorhersagen kann. Unüberwachtes Lernen hingegen arbeitet mit Datensätzen, die nur Eingabedaten enthalten, und versucht, verborgene Muster oder Strukturen in den Daten zu entdecken. Bestärkendes Lernen basiert auf dem Prinzip der Belohnung und Bestrafung und wird häufig für Entscheidungsprozesse und Steuerungsaufgaben eingesetzt („Was ist Künstliche Intelligenz?“, o. J.).

**Deep Learning (DL)** ist eine spezielle Art des maschinellen Lernens, die auf künstlichen neuronalen Netzen basiert. Diese Netze bestehen aus mehreren Schichten von Knoten (Neuronen), die miteinander verbunden sind und durch Gewichte verstärkt oder abgeschwächt werden („Was ist Künstliche Intelligenz?“, o. J.). Deep Learning ist besonders effektiv bei der Verarbeitung großer und komplexer Datensätze und hat in den letzten Jahren bedeutende Fortschritte in Bereichen wie der Bilderkennung, der Spracherkennung und der Verarbeitung natürlicher Sprache gemacht. Ein tiefes neuronales Netz (Deep Neural Network, DNN) kann komplexe Muster und Zusammenhänge in Daten erkennen, indem es Schichten von Abstraktionen bildet („Deep Neural Network - an overview | ScienceDirect Topics“, o. J.). Convolutional Neural Networks (CNNs) sind eine spezielle Art von tiefen neuronalen Netzen, die besonders gut für die Verarbeitung von Bilddaten geeignet sind, da sie lokale Muster in

Bildern erkennen und verarbeiten können (Erfassung et al., o. J.). Recurrent Neural Networks (RNNs) hingegen sind darauf ausgelegt, Sequenzen von Daten zu verarbeiten, was sie ideal für Aufgaben wie Sprachverarbeitung und Zeitreihenvorhersagen macht („Was sind rekurrente neuronale Netze?“, 2023).

**Neuronale Netze** sind inspiriert von der Struktur und Funktionsweise des menschlichen Gehirns. Sie bestehen aus einer großen Anzahl von Neuronen, die in Schichten organisiert sind. Jede Verbindung zwischen Neuronen hat ein Gewicht, das angepasst wird, um das Netzwerk zu trainieren und Muster in den Daten zu erkennen. Ein neuronales Netz lernt, indem es die Gewichte basierend auf den Fehlern anpasst, die es bei der Vorhersage von Ergebnissen macht. Dieser Prozess wird als Training bezeichnet und erfordert in der Regel große Mengen an Daten und Rechenleistung (Roscher, Guderitz, & Hengl, o. J.).

**Natürliche Sprachverarbeitung (NLP)** befasst sich mit der Interaktion zwischen Computern und menschlicher Sprache. NLP umfasst eine Reihe von Techniken, die es Maschinen ermöglicht, natürliche Sprache zu verstehen, zu interpretieren und darauf zu reagieren. Dies umfasst Aufgaben wie die maschinelle Übersetzung, die Textklassifikation, die Sentiment-Analyse und die Spracherkennung. Moderne NLP-Modelle, wie das Transformer-Modell und seine Varianten (z.B. BERT, GPT-3), haben die Fähigkeit, kontextuelle Beziehungen in Texten zu verstehen und sehr natürlich klingende Texte zu generieren. Diese Modelle basieren auf tiefen neuronalen Netzen und nutzen Techniken wie Attention-Mechanismen, um relevante Teile des Textes zu identifizieren und zu verarbeiten (Litzel & Luber, 2022).

**Computer Vision** ist ein weiteres wichtiges Teilgebiet der KI, das sich mit der automatischen Extraktion, Analyse und Interpretation von Informationen aus Bildern und Videos beschäftigt. Techniken der Computer Vision ermöglichen es Maschinen, visuelle Daten zu verstehen und zu verarbeiten, ähnlich wie es das menschliche Auge und Gehirn tun. Anwendungen der Computer Vision umfassen die Gesichtserkennung, die Objekterkennung, die Bilderkennung und die Bildsegmentierung. Technologien wie CNNs sind auch in der Computer Vision weit verbreitet, da sie besonders gut darin sind, Muster und Strukturen in Bildern zu erkennen und zu verarbeiten. Diese verschiedenen Technologien und Konzepte bilden das Fundament der modernen KI und haben zahlreiche Anwendungen in der realen Welt. Sie ermöglichen es Computern, Aufgaben zu erfüllen, die früher nur Menschen vorbehalten waren, und eröffnen neue Möglichkeiten in Bereichen wie der Medizin, der Automobilindustrie, der Finanzanalyse und vielen anderen. Die kontinuierliche Weiterentwicklung dieser

Technologien wird die Fähigkeit von KI-Systemen weiter verbessern und ihre Anwendungsbereiche erweitern („Computer Vision - Fraunhofer IAIS“, o. J.).

### **2.3.2 KI in der Datenverarbeitung und Programmierung**

Künstliche Intelligenz (KI) ist in den letzten Jahren auf ein neues Level gehoben worden und wird zunehmend in der Datenverarbeitung und Programmierung eingesetzt. Diese Entwicklungen beeinflussen die Art und Weise, wie Software entwickelt, getestet und gewartet wird, und bieten zahlreiche Vorteile für Entwickler und Unternehmen.

#### **Automatisierte Code-Generierung und -Überprüfung**

Eine der beeindruckendsten Anwendungen von KI in der Programmierung ist die automatisierte Code-Generierung. Tools wie GitHub Copilot, basierend auf OpenAI's Code, ermöglichen es Entwicklern, aus einfachen Textbeschreibungen funktionsfähigen Code zu erzeugen. Dies beschleunigt den Entwicklungsprozess erheblich und reduziert die Notwendigkeit für repetitive Programmieraufgaben. Entwickler können Vorschläge für Code-Snippets erhalten, Boilerplate-Code automatisch generieren und komplexe Algorithmen effizienter implementieren (Kyle Daigle, 2023).

#### **Fehlererkennung und Debugging**

KI-gestützte Tools spielen auch eine wichtige Rolle beim Debugging. Sie können Quellcode analysieren, häufige Fehler identifizieren und Vorschläge zur Behebung dieser Fehler machen. Dies hilft Entwicklern, ihre Programme schneller und effizienter zu debuggen, wodurch die Gesamtqualität der Software verbessert wird. Zum Beispiel verwendet das Tool DeepCode maschinelles Lernen, um Code auf Sicherheitslücken und Bugs zu überprüfen und entsprechende Korrekturen vorzuschlagen (Shani, 2023).

#### **Optimierung und Refactoring von Code**

Ein weiterer wichtiger Anwendungsbereich von KI ist die Optimierung und das Refactoring von Code. Durch maschinelles Lernen können Tools wie Sourcery Vorschläge zur Verbesserung der Code-Performance, zur Reduzierung des Speicherverbrauchs und zur Erhöhung der Lesbarkeit und Wartbarkeit machen. Diese Tools erkennen redundante Codeblöcke, schlagen effizientere Algorithmen vor und strukturieren den Code sauberer, was die langfristige Wartung erleichtert (Ruiz, 2024).

#### **Natürliche Sprachverarbeitung (NLP) in der Softwareentwicklung**

Natürliche Sprachverarbeitung (NLP) verbessert die Interaktion zwischen Entwicklern und Softwaretools erheblich. Durch die Integration von NLP können Entwickler auf natürliche Weise mit ihren Entwicklungsumgebungen kommunizieren, was die Benutzerfreundlichkeit und Effizienz deutlich erhöht. Moderne AI-Technologien transformieren die Art und Weise, wie Entwickler mit ihren Tools interagieren, indem sie eine nahtlose und intuitive Kommunikation ermöglichen. Diese Entwicklungen führen zu einer erheblich gesteigerten Produktivität und einem verbesserten Workflow, da Entwickler schneller und effektiver arbeiten können („The State of AI in 2023“, 2023).

### **Automatisierte Tests und Qualitätssicherung**

KI-gestützte Tools verbessern auch die Testabdeckung und die Effizienz der Qualitätssicherung. Durch maschinelles Lernen und prädiktive Analysen können diese Tools automatisch Testfälle generieren, die schwer zu testende Bereiche des Codes abdecken. Sie erkennen Muster in Fehlerberichten und Testdaten, um potenzielle Problemstellen frühzeitig zu identifizieren und zu beheben. Dies trägt dazu bei, die Zuverlässigkeit und Stabilität von Softwareprodukten zu erhöhen („AI in Quality Assurance“, o. J.).

### **Aktuelle Entwicklungen und Trends**

Aktuelle Berichte zeigen, dass generative KI, wie die von OpenAI und anderen großen KI-Anbietern entwickelten Modelle, die Art und Weise, wie Entwickler arbeiten, tiefgreifend verändert hat. Eine Umfrage unter Entwicklern ergab, dass 92 % der Befragten bereits KI-Coding-Tools verwenden, sowohl innerhalb als auch außerhalb ihrer Arbeit. Diese Tools bieten Vorteile wie verbesserte Codequalität, schnellere Fertigstellung und effizienteres Beheben von Vorfällen. Darüber hinaus haben führende Unternehmen gezeigt, dass sie durch den Einsatz von KI in Produkt- und Serviceentwicklung sowie Risikomanagement signifikante Werte erzielen können (Shani, 2023).

Zusammenfassend lässt sich sagen, dass KI-Technologien die Programmierung erheblich verändern und verbessern können. Sie ermöglichen es Entwicklern, effizienter zu arbeiten, komplexe Probleme schneller zu lösen und qualitativ hochwertigere Softwareprodukte zu erstellen. Die kontinuierliche Weiterentwicklung dieser Technologien wird die Art und Weise, wie Software entwickelt und gewartet wird, weiter transformieren und neue Möglichkeiten für Innovation und Effizienzsteigerung eröffnen.

### 3 Manuelle Erstellung und Überprüfung von SQL-Statements

Die Erstellung und Überprüfung von SQL-Statements ist eine zentrale Aufgabe in der Datenbankverwaltung. Sie umfasst die Entwicklung und Validierung von Abfragen, die auf relationalen Datenbanken ausgeführt werden, um Daten zu manipulieren und abzurufen. Dieser Abschnitt beschreibt detailliert die Methoden und Werkzeuge zur manuellen Erstellung und Überprüfung von SQL-Statements sowie die Herausforderungen und gängigen Fehlerquellen.

#### 3.1 Methoden und Werkzeuge

Bei der manuellen Erstellung und Überprüfung von SQL-Statements kommen verschiedene Methoden und Werkzeuge zum Einsatz, um die Genauigkeit und Effizienz der Abfragen zu gewährleisten. Hier sind einige der wichtigsten:

SQL-Editoren und IDEs wie SQL Server Management Studio (SSMS), MySQL Workbench und Oracle SQL Developer bieten benutzerfreundliche Oberflächen zur Erstellung, Bearbeitung und Ausführung von SQL-Statements. Diese Werkzeuge unterstützen Funktionen wie Syntax-Highlighting, Autovervollständigung und Fehlerüberprüfung, was die Erstellung von korrektem und effizientem Code erleichtert („Die besten Online-SQL-Editoren“, 2024).

SQL Server Management Studio (SSMS): Diese IDE ist besonders für SQL Server und Azure SQL Datenbanken nützlich und bietet umfassende grafische Werkzeuge und visuelle Designer, die es einfacher machen, Datenbankschemata, Tabellen und Indizes zu erstellen und zu ändern. SSMS unterstützt auch Debugging und Profiling, um die Leistung von Abfragen zu optimieren (Shivakumar, o. J.)

MySQL Workbench: Diese plattformübergreifende IDE bietet umfassende Tools für SQL-Entwicklung, Datenmodellierung und Serveradministration. Sie ermöglicht die visuelle Gestaltung und Verwaltung von Datenbankschemata und bietet Werkzeuge zur Performance-Optimierung und Datenbankmigration (Ostrowska, 2022)

Oracle SQL Developer: Diese kostenlose IDE bietet umfassende Werkzeuge für die Verwaltung und Entwicklung von Oracle-Datenbanken. Sie unterstützt Funktionen wie SQL-Formatierung, Skriptausführung und einen visuellen Abfrage-Builder. Diese Funktionen machen es einfacher, komplexe SQL-Abfragen zu erstellen und zu verwalten („SQL Developer“, o. J.).

**Abfrageoptimierungstools:** Tools wie der SQL Server Profiler oder der MySQL Query Analyzer helfen dabei, die Performance von SQL-Abfragen zu überwachen und zu optimieren. Diese Werkzeuge analysieren die Ausführungspläne und bieten Vorschläge zur Verbesserung der Abfrageeffizienz, wie z.B. die Verwendung von Indizes oder die Umstrukturierung von Abfragen („MySQL :: MySQL Query Analyzer“, o. J.).

**Test- und Validierungstechniken:** Zur Sicherstellung der Korrektheit und Effizienz von SQL-Statements werden verschiedene Testmethoden eingesetzt, darunter Unit-Tests, Integrations-tests und Lasttests. Diese Tests überprüfen, ob die Abfragen die erwarteten Ergebnisse liefern und unter verschiedenen Bedingungen performant bleiben („Data Validation Testing: Techniques, Examples, & Tools“, o. J.).

**Manuelle Überprüfung:** Die manuelle Überprüfung von SQL-Statements durch erfahrene Entwickler oder Datenbankadministratoren bleibt eine wichtige Methode zur Identifizierung und Korrektur von Fehlern. Dies umfasst das Lesen und Verstehen des Codes, das Testen der Abfragen mit unterschiedlichen Datensätzen und die Analyse der Abfrageergebnisse.

### 3.2 Herausforderungen und gängige Fehlerquellen

Die manuelle Erstellung und Überprüfung von SQL-Statements ist nicht ohne Herausforderungen. Zu den häufigsten Fehlerquellen und Herausforderungen gehören:

**Syntaxfehler:** Syntaxfehler sind eine häufige Fehlerquelle bei der Erstellung von SQL-Statements. Diese Fehler entstehen durch Tippfehler, fehlende oder falsche Schlüsselwörter, Klammern oder Anführungszeichen. Syntaxfehler können leicht durch die Verwendung von SQL-Editoren mit Syntax-Highlighting und Autovervollständigung minimiert werden. (Petrovic, 2014)

**Logische Fehler:** Logische Fehler treten auf, wenn die Abfrage nicht die gewünschten Ergebnisse liefert. Dies kann durch falsche Join-Bedingungen, fehlerhafte WHERE-Klauseln oder unsachgemäße Gruppierungen und Aggregationen verursacht werden. Solche Fehler erfordern eine gründliche Analyse der Abfrage und der zugrunde liegenden Datenstruktur („Debugging SQL query logic“, o. J.)

**Komplexität und Wartbarkeit:** Komplexe SQL-Statements sind oft schwer zu lesen und zu warten. Dies kann zu Fehlern führen, wenn Änderungen vorgenommen werden müssen. Eine gute Praxis ist es, Abfragen so einfach und modular wie möglich zu halten und klare

Kommentare zu verwenden, um die Logik und den Zweck der Abfragen zu erläutern („Common Database Management Challenges and How to Fix Them“, o. J.)

Diese Herausforderungen und Fehlerquellen verdeutlichen die Notwendigkeit sorgfältiger Planung, gründlicher Tests und kontinuierlicher Optimierung bei der manuellen Erstellung und Überprüfung von SQL-Statements. Die Verwendung geeigneter Werkzeuge und Methoden kann dazu beitragen, die Qualität und Effizienz der Abfragen zu verbessern und die Gesamtleistung der Datenbank zu steigern. Künstliche Intelligenz (KI) kann dabei in vielen Bereichen unterstützen, beispielsweise durch automatisierte Fehlererkennung und -behebung, Optimierungsvorschläge und Unterstützung bei der Abfrageerstellung, was die Effizienz weiter erhöht und Fehler reduziert.

## 4 Experiment

### 4.1 Zielsetzung des Experiments

Das Hauptziel dieses Experiments ist es, die Genauigkeit von SQL-Statements zu bewerten, durch den Einsatz von Künstlicher Intelligenz (KI) erstellt werden. Die für dieses Vorhaben entwickelte Datenbankstruktur einer fiktiven Universität dient als Grundlage, um die Wirksamkeit KI-gestützter Methoden in einem standardisierten Datenbankmanagementsystem, nämlich MySQL, direkt zu vergleichen. Durch das Experiment sollen Einblicke gewonnen werden, wie KI-Technologien die Überprüfung von SQL-Statements optimieren können. Es soll außerdem die spezifischen Szenarien identifizieren, in denen KI den größten Nutzen bietet, und mögliche Herausforderungen oder Einschränkungen aufzeigen, die bei der Anwendung von KI in der SQL-Statement-Generierung entstehen können.

### 4.2 Vorarbeit des Experiments

Die Universitätsdatenbank wurde entworfen, um alle gängigen Beziehungstypen und Datentypen zu integrieren. Dies gewährleistet eine solide und vergleichbare Basis für das Experiment zur Überprüfung von SQL-Statements mit und ohne KI. Durch die Einbeziehung verschiedener Beziehungen wie 1:1, 1

, N:1 und M

sowie Datentypen wie INT, VARCHAR, und DATE bietet die Datenbankstruktur eine umfassende Umgebung für Tests unter realen Bedingungen. Auf diesen grundlegenden Überlegungen entstand folgendes ER-Diagramm, welches in Abbildung 1 zu sehen ist.



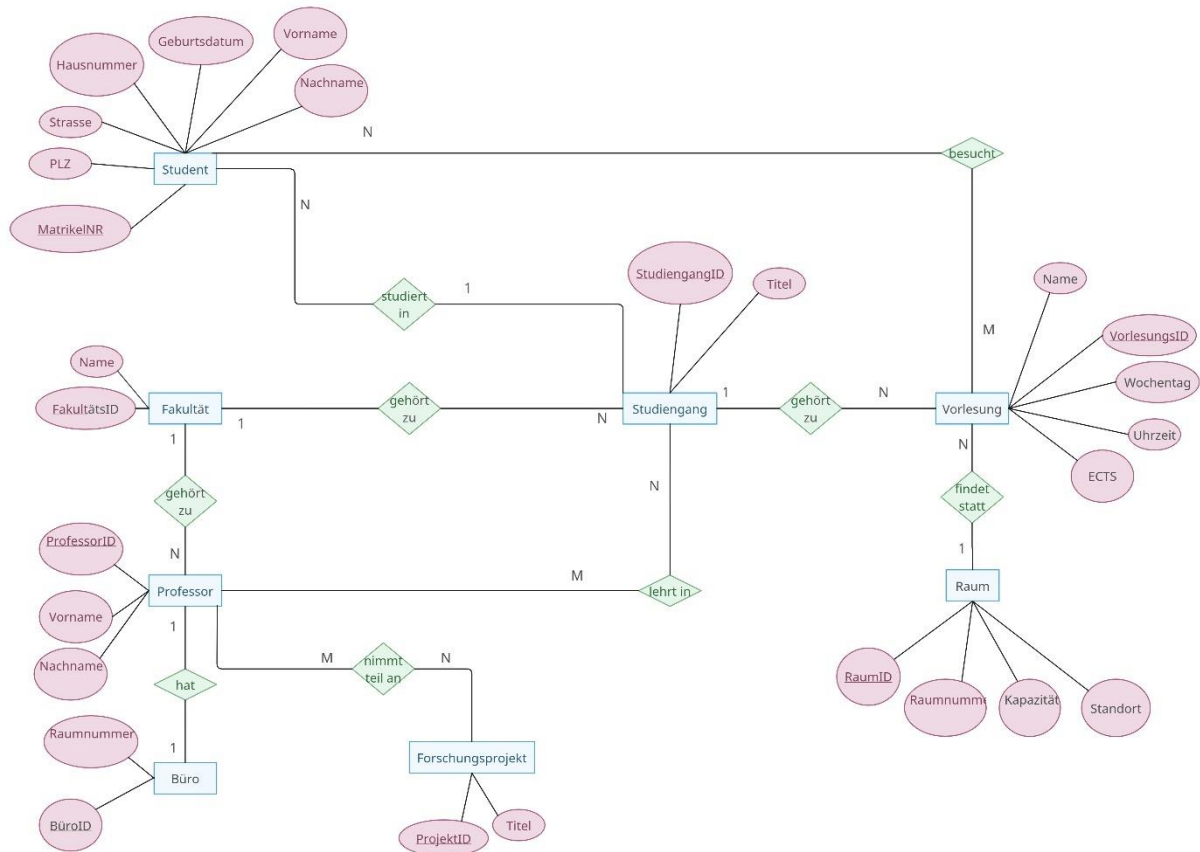


Abbildung 1: ER-Diagramm der Universitätsdatenbank (Quelle: Eigene Darstellung)

Das ER-Diagramm besteht aus 8 Entitäten. Die Entität Student hat die Attribute MatrikelNR als Primärschlüssel, PLZ, Straße, Hausnummer, Geburtsdatum, Vorname und Nachname. Die Entität Fakultät hat die Attribute FakultätsID als Primärschlüssel und den Namen. Die Entität Professor hat die Attribute ProfessorID, Vorname und Nachname. Die Entität Büro hat die Attribute BüroID als Primärschlüssel und die Raumnummer. Die Entität Forschungsprojekt hat die Attribute ProjektID als Primärschlüssel und den Titel. Die Entität Studiengang hat die Attribute StudiengangID als Primärschlüssel und den Titel. Die Entität Vorlesung hat die Attribute VorlesungsID als Primärschlüssel, Name, Wochentag, Uhrzeit und ECTS. Die Entität Raum hat die Attribute RaumID als Primärschlüssel, Raumnummer, Kapazität und den Standort. Die Kardinalitäten zwischen den Entitäten sehen folgendermaßen aus. Büro hat eine 1 zu 1 Beziehung zu Professor, denn ein Professor hat nur ein Büro und in einem Büro ist nur ein Professor. Forschungsprojekt hat eine N zu M Beziehung zu Professor, denn in einem Forschungsprojekt können mehrere Professoren teilnehmen und ein Professor kann an mehreren Forschungsprojekten teilnehmen. Professor hat eine N zu 1 Beziehung zu Fakultät, denn ein Professor gehört zu genau einer Fakultät und zu einer Fakultät gehören mehrere Professoren. Außerdem hat Professor eine N zu M Beziehung zu Studiengang, denn ein Professor lehrt in mehreren Studiengängen und in einem Studiengang lehren mehrere

Professoren. Fakultät hat neben der Beziehung zu Professor noch eine 1 zu N Beziehung zu Studiengang, denn zu einer Fakultät gehören mehrere Studiengänge und ein Studiengang gehört zu genau einer Fakultät. Student hat eine N zu 1 Beziehung zu Studiengang, denn ein Student gehört zu genau einem Studiengang und zu einem Studiengang gehören mehrere Professoren. Außerdem hat Student eine N zu M Beziehung zu Vorlesung, denn ein Student besucht mehrere Vorlesungen und eine Vorlesung wird von mehreren Studenten besucht. Studiengang hat neben der Beziehung zu Fakultät noch eine 1 zu N Beziehung zu Vorlesung, denn zu einem Studiengang gehören mehrere Vorlesungen und eine Vorlesung gehört zu genau einem Studiengang. Vorlesung hat neben den Beziehungen zu Studiengang und Student noch eine N zu 1 Beziehung zu Raum, denn eine Vorlesung findet in genau einem Raum statt und in einem Raum können mehrere Vorlesungen stattfinden.

Nachdem das ER-Diagramm erstellt wurde, das die Beziehungen und Entitäten der Universitätsdatenbank visualisiert, ist der nächste Schritt die Umsetzung dieses Diagramms in ein relationales Datenbankmodell. Dieser Schritt konvertiert das konzeptionelle Modell des ER-Diagramms in eine Struktur, die in einem relationalen Datenbankmanagementsystem in diesem Fall MySQL Workbench implementiert werden kann.

Basierend auf den Entitäten des ER-Diagramms wird für jede Entität eine entsprechende Tabelle im relationalen Modell erstellt. Die Attribute der Entitäten werden zu Spalten der Tabellen, wobei der Datentyp für jede Spalte festgelegt wird, entsprechend den Anforderungen und Eigenschaften, die im ER-Diagramm definiert wurden.

### **4.3 Definition der Fremdschlüsselbeziehungen**

Die Beziehungen zwischen den Entitäten, die im ER-Diagramm als Linien dargestellt sind, werden im relationalen Modell durch Fremdschlüssel implementiert. Diese Fremdschlüssel stellen die Verbindungen zwischen den Tabellen her und sorgen dafür, dass die Beziehungen konsistent und die Datenintegrität gewahrt bleiben. Jede Beziehungstyp wird entsprechend durch Fremdschlüssel und gegebenenfalls zusätzliche Verknüpfungstabellen umgesetzt.

In dem Fall der Universitätsdatenbank wird die BüroID Fremdschlüssel in der Tabelle Professor. Die FakultätsID wird Fremdschlüssel in der Tabelle Studiengang und in der Tabelle Professor. Die RaumID wird Fremdschlüssel in der Tabelle Vorlesungen. Die StudiengangID wird Fremdschlüssel in der Tabelle Student. Da zwischen Forschungsprojekt und Professor eine N zu M Beziehung besteht, muss hier die Zuordnungstabelle `nimmt_teil_an` erstellt werden. Die beiden Primärschlüssel ProfessorID und ProjektID werden Fremdschlüssel in der Tabelle

nimmt\_teil\_an und bilden den zusammengesetzten Primärschlüssel dieser Tabelle. Da zwischen Vorlesung und Student eine N zu M Beziehung besteht muss hier die Zuordnungstabelle besucht erstellt werden. Die beiden Primärschlüssel MatrikelNR und VorlesungsID werden Fremdschlüssel in der Tabelle besucht und bilden den zusammengesetzten Primärschlüssel dieser Tabelle. Da zwischen Studiengang und Professor eine N zu M Beziehung besteht muss hier die Zuordnungstabelle lehrt\_in erstellt werden. Die beiden Primärschlüssel ProfessorID und StudiengangID werden Fremdschlüssel in der Tabelle lehrt\_in und bilden den zusammengesetzten Primärschlüssel dieser Tabelle. Das relationale Datenmodell sieht dann folgendermaßen, wie in Abbildung 2 dargestellt, aus:

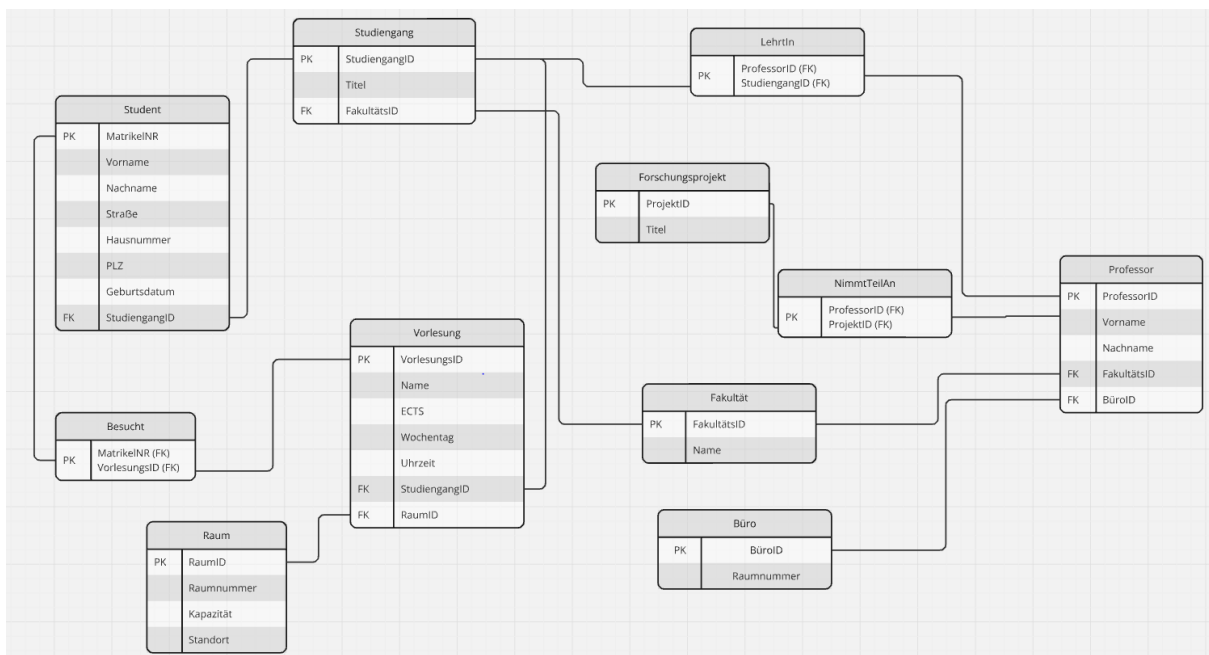


Abbildung 2 Relationales Datenbankmodell der Universitätsdatenbank (Quelle: Eigene Darstellung)

#### 4.4 Erstellung der Datenbank in MySQL

Um die Universitätsdatenbank in MySQL zu erstellen, gibt es zwei Methoden: die Eingabe von DDL (Data Definition Language) Statements und die Verwendung manueller Tools. Bei der ersten Methode werden DDL-Statements geschrieben und ausgeführt, um Tabellen zu erstellen und die Beziehungen zwischen diesen Tabellen zu definieren. Dies ermöglicht eine präzise Kontrolle über die Datenbankstruktur und bietet die Flexibilität, komplexe Konfigurationen zu implementieren. Die zweite Methode beinhaltet die manuelle Erstellung der Tabellen mit MySQL Workbench, einer benutzerfreundlichen Point-and-Click-Oberfläche. Mit MySQL Workbench können Sie Tabellen und Beziehungen manuell erstellen, ohne SQL-Skripte schreiben zu müssen. Zunächst öffnen Sie MySQL Workbench und verbinden sich mit Ihrem

MySQL-Server. Dann klicken Sie im Navigator auf den Tab „Tables“ der Datenbank, in der Sie arbeiten möchten. Mit einem Rechtsklick wählen Sie „Create Table“ und geben den Tabellennamen ein. Anschließend fügen Sie die Spalten hinzu, einschließlich des Primärschlüssels, und speichern die Tabelle. Um Fremdschlüssel zu definieren, klicken Sie auf die Tabelle, der Sie einen Fremdschlüssel hinzufügen möchten, und gehen zum Reiter „Foreign Keys“. Dort fügen Sie eine neue Fremdschlüsselbeziehung hinzu, indem Sie die referenzierte Tabelle und die entsprechenden Spalten auswählen und die Änderungen speichern. Diese Schritte wiederholen Sie für alle Tabellen in Ihrem Schema, wobei Sie die erforderlichen Fremdschlüsselbeziehungen definieren. Die manuelle Methode in MySQL Workbench bietet mehrere Vorteile. Sie ist benutzerfreundlich, da die visuelle Oberfläche das Erstellen von Tabellen und Beziehungen ohne tiefgehende SQL-Kenntnisse ermöglicht. Darüber hinaus bietet MySQL Workbench eine grafische Darstellung des Datenbankmodells, was das Verständnis und die Überprüfung der Datenbankstruktur erleichtert. Durch die Verwendung von Dropdown-Menüs und grafischen Eingabefeldern wird zudem die Wahrscheinlichkeit von Syntaxfehlern reduziert. Außerdem muss beim Schreiben der DDL-Statements eine bestimmte Reihenfolge eingehalten werden bei der Erstellung der Tabellen, was eine zusätzliche Fehlerquelle bietet.

#### **4.5 Einfügen von Testdaten**

Nachdem die Tabellen und die Fremdschlüsselbeziehungen in der MySQL-Datenbank erstellt wurden, müssen nun Testdatensätze eingefügt werden. Dieser Schritt ist wichtig, um die Datenbankstruktur zu validieren und sicherzustellen, dass alle Beziehungen korrekt funktionieren. Dabei ist es wichtig, die Reihenfolge der Dateneingabe zu beachten, um sicherzustellen, dass alle Fremdschlüsselbeziehungen erfüllt werden können. Zunächst sollten die Datensätze für die Fakultäten eingefügt werden, da diese von den Professoren und Studiengängen referenziert werden. Danach werden die Räume eingefügt, da diese von den Vorlesungen referenziert werden. Im nächsten Schritt sollten die Büros hinzugefügt werden, bevor die Datensätze der Professoren eingefügt werden, da die Professoren auf die Büros verweisen. Als nächstes müssen die Studiengänge eingefügt werden, bevor die Datensätze der Professoren und Studenten hinzugefügt werden, da beide Entitäten auf Studiengänge verweisen. Nachdem die Büros und Fakultäten hinzugefügt wurden, können die Professoren eingefügt werden. Forschungsprojekte können unabhängig von anderen Entitäten eingefügt werden. Vorlesungen sollten nach den Studiengängen und Räumen eingefügt werden, da diese

Tabellen vorher befüllt sein müssen. Nachdem die Vorlesungen hinzugefügt wurden, können die Studenten eingefügt werden, da die Studiengänge bereits existieren. Zum Schluss werden die N-M Beziehungstabellen (`nimmt_teil_an`, `besucht`, `lehrt_in`) befüllt, da diese auf die bereits eingefügten Datensätze der Professoren, Studenten, Vorlesungen, Studiengänge und Forschungsprojekte referenzieren. Beispielsweise könnten die Datensätze für die Fakultäten mit folgendem SQL-Befehl eingefügt werden: `INSERT INTO Fakultät (FakultätsID, Name) VALUES (1, 'Informatik'), (2, 'Mathematik');`. Für die Räume könnte ein SQL-Befehl wie `INSERT INTO Raum (RaumID, Raumnummer, Kapazität, Standort) VALUES (1, 'A101', 30, 'Gebäude A'), (2, 'B201', 50, 'Gebäude B');` verwendet werden. Büros könnten mit `INSERT INTO Büro (BüroID, Raumnummer) VALUES (1, 'C301'), (2, 'D401');` hinzugefügt werden.

Die Studiengänge könnten mit `INSERT INTO Studiengang (StudiengangID, Titel, FakultätsID) VALUES (1, 'Informatik', 1), (2, 'Mathematik', 2);` hinzugefügt werden, während die Professoren mit `INSERT INTO Professor (ProfessorID, Vorname, Nachname, BüroID, FakultätsID) VALUES (1, 'Max', 'Mustermann', 1, 1), (2, 'Erika', 'Musterfrau', 2, 2);` eingefügt werden könnten. Forschungsprojekte könnten mit `INSERT INTO Forschungsprojekt (ProjektID, Titel) VALUES (1, 'KI-Forschung'), (2, 'Algorithmus-Entwicklung');` eingefügt werden.

Die Vorlesungen könnten mit `INSERT INTO Vorlesung (VorlesungsID, Name, Wochentag, Uhrzeit, ECTS, StudiengangID, RaumID) VALUES (1, 'Datenbanken', 'Montag', '10:00:00', 5, 1, 1), (2, 'Analysis', 'Dienstag', '12:00:00', 6, 2, 2);` hinzugefügt werden. Studenten könnten mit `INSERT INTO Student (MatrikelNR, PLZ, Straße, Hausnummer, Geburtsdatum, Vorname, Nachname, StudiengangID) VALUES (1, '12345', 'Hauptstraße', '1', '2000-01-01', 'Anna', 'Schmidt', 1), (2, '67890', 'Nebenstraße', '2', '1999-05-05', 'Ben', 'Meier', 2);` hinzugefügt werden.

Zum Schluss könnten die N-M Beziehungstabellen mit folgenden SQL-Befehlen befüllt werden: `INSERT INTO nimmt_teil_an (ProfessorID, ProjektID) VALUES (1, 1), (2, 2);`, `INSERT INTO besucht (MatrikelNR, VorlesungsID) VALUES (1, 1), (2, 2);`, und `INSERT INTO lehrt_in (ProfessorID, StudiengangID) VALUES (1, 1), (2, 2);`. Durch diese Reihenfolge der Dateneingabe wird sichergestellt, dass alle Fremdschlüsselbeziehungen eingehalten werden und die Datenintegrität in der Datenbank gewährleistet ist. Dieser Prozess ist ein wesentlicher Schritt, um die Funktionalität und Zuverlässigkeit der Datenbankstruktur zu überprüfen.

## 4.6 Schreiben von Testfällen

Die Testfälle beginnen mit einfachen Abfragen, die grundlegende Bedingungen beinhalten, um die Fähigkeit der KI zu prüfen, korrekte und effiziente SQL-Befehle für Standarddatenabfragen zu formulieren. Dies bildet die Basis, auf der komplexere Szenarien aufgebaut werden.

Weitergehend umfassen die Testfälle Abfragen, die Joins zwischen zwei Tabellen erfordern. Hier wird die KI daraufhin getestet, ob sie in der Lage ist, korrekte Verknüpfungen zwischen Tabellen herzustellen und dabei spezifische Bedingungen zu berücksichtigen. Dies ist entscheidend, um zu beurteilen, ob die KI die Datenstruktur und die Beziehungen zwischen den Tabellen effektiv versteht. Für tiefere Einblicke und komplexere Datenbeziehungen beinhaltet die dritte Kategorie Testfälle, die Joins über drei oder mehr Tabellen erfordern. Diese Abfragen sind besonders herausfordernd, da sie ein fortgeschrittenes Verständnis der Datenbankstruktur und optimierte Abfragestrategien erfordern. Die vierte Kategorie fokussiert sich auf die Anwendung von Aggregatsfunktionen ohne Joins, um zu testen, wie gut die KI mit Berechnungen innerhalb der Datenbank umgehen kann. Schließlich kombiniert die letzte Kategorie Joins und Aggregatsfunktionen in denselben Abfragen. Diese komplexen Abfragen testen das höchste Niveau der KI-Fähigkeiten, da sie ein tiefes Verständnis der Datenverknüpfungen und gleichzeitig das korrekte Anwenden von Aggregationslogiken erfordern. Das sorgfältige Entwerfen dieser Testfälle ermöglicht eine detaillierte Bewertung der KI, um ihre Fähigkeit zu messen, in einem realen Umfeld präzise und effektive SQL-Statements zu generieren. Diese Tests sind nicht nur entscheidend für die Beurteilung der aktuellen Leistung der KI, sondern auch für die Identifikation von Bereichen, in denen weitere Verbesserungen notwendig sind. Durch diese methodische Testung kann man fundierte Aussagen über die Praxistauglichkeit der KI in Bezug auf die automatisierte Erstellung von SQL-Statements machen. Die erwähnten Testfälle sind in der Anlage zu finden.

## 4.7 Testen der KIs

Nachdem die Testfälle für die Bewertung festgelegt wurden, beginnt der umfassende Testprozess der künstlichen Intelligenzen ChatGPT und Text2SQL.ai, zwei führende Technologien in ihren jeweiligen Anwendungsgebieten. Ziel ist es, ihre Fähigkeit zur Generierung von SQL-Statements zu evaluieren, indem sie unter zwei unterschiedlichen Informationsbedingungen getestet werden. Zum einen erhalten die KIs die vollständigen DDL-Befehle, die detailliert die Struktur und Beziehungen innerhalb der Datenbank darlegen. Zum anderen wird die Datenbankstruktur lediglich in Worten beschrieben, was die KIs

herausfordert, aus weniger strukturierten Informationen die notwendigen SQL-Befehle abzuleiten. Für jeden der sorgfältig konstruierten Testfälle werden die Ergebnisse akribisch in einer Excel-Tabelle erfasst. Dies ermöglicht eine präzise Analyse und Bewertung der Antworten der KIs. Die erfassten Fehler werden in drei Hauptkategorien eingeteilt: Syntaxfehler, die technische Ungenauigkeiten in den SQL-Befehlen betreffen; Logikfehler, die fehlerhafte Verknüpfungen oder Anwendungen von Datenbanklogik beinhalten, wie beispielsweise das Joinen nicht zusammengehöriger Tabellen; sowie Fehler durch zu viel Information, bei denen die KIs mehr Daten abfragen, als eigentlich benötigt. Diese systematische Herangehensweise dient dazu, nicht nur die Genauigkeit der KIs in der SQL-Generierung zu messen, sondern auch deren Robustheit und Anpassungsfähigkeit an verschiedene Arten von Eingabeinformationen zu testen. Besonders wichtig ist dabei, dass nach jedem Testfall ein neuer Dialog mit der KI initiiert wird. Dies stellt sicher, dass kein Lerneffekt der KI die Ergebnisse verfälscht und dass jede Testrunde unter identischen Ausgangsbedingungen stattfindet. Durch diesen methodischen Ansatz wird nicht nur die unmittelbare Leistungsfähigkeit der KIs verlässlich evaluiert, sondern es werden auch wertvolle Einblicke in die praktische Anwendbarkeit dieser Technologien in realen Einsatzszenarien gewonnen. Die Ergebnisse dieser Tests bieten nicht nur Aufschluss über die gegenwärtigen Fähigkeiten der KIs, sondern legen auch mögliche Entwicklungspfade für zukünftige Verbesserungen in der Automatisierung von SQL-Generierungsprozessen dar.

## 5 Testergebnisse

Die Testfälle deckten verschiedene SQL-Abfragen ab, darunter einfache Abfragen mit Bedingungen, Joins zwischen mehreren Tabellen und komplexere Logikabfragen. Die Bewertungskriterien umfassten die Richtigkeit der Abfragen, das Vorhandensein von Syntaxfehlern, Logikfehlern und das Problem von zu viel gelieferten Informationen. Die Datenbankstruktur ist wie bereits in der Vorbereitung und der Erläuterung der Beziehungen durch Fremdschlüssel definiert.

Die Ergebnisse der Analyse zeigen, dass ChatGPT bei der Angabe aller DDL-Statements der Tabellen insgesamt 50 Testfälle bearbeitet hat, von denen 43 korrekt beantwortet wurden. Es gab keine Syntaxfehler, aber 4 Logikfehler und 3 Fälle, in denen zu viel Information geliefert wurde. Text2SQL.ai bearbeitete ebenfalls 50 Testfälle, von denen 46 korrekt beantwortet wurden. Auch hier gab es keine Syntaxfehler, aber 4 Fehler bei denen wieder zu viel Information geliefert wurde

Bei einer detaillierteren Betrachtung der Fehler zeigt sich, dass ChatGPT in 4 Fällen Logikfehler aufwies, die auf Missverständnisse oder fehlerhafte Interpretationen der Anfragen zurückzuführen waren. Ein bemerkenswerter Fall ist die Abfrage „Zeige alle Studenten mit Vorname und Nachname die nach dem Jahr 2000 geboren sind“. Hier bestand der Fehler darin, dass auch das Jahr 2000 selbst inkludiert wurde, während nur Studenten ab dem Jahr 2001 einbezogen werden sollten. Diese Art von Fehler deutet auf eine ungenaue Interpretation des Ausdrucks „nach dem Jahr 2000“ hin. Ein weiterer Fehler trat bei der Abfrage „Zeige mir alle Professoren mit Vorname und Nachname und die Namen der Vorlesungen, die sie halten“ auf. Hier wurde nicht korrekt berücksichtigt, dass die Tabelle `lehrt_in` nicht direkt mit der Tabelle `Studiengang` verknüpft werden kann. Solche Fehler zeigen, dass komplexe Bedingungen und Verknüpfungen zwischen Tabellen eine Herausforderung darstellen können.

Bei Angabe aller DDL-Statements hatte text2SQL.ai 2 Logikfehler und 2 Fehler bei denen zu viel Information geliefert wurde. Ein Beispiel für einen Logikfehler ist die Abfrage „Zeige alle Räume mit Raumnummer am Standort Gebäude A“, hier wurde die abgefragte Spalte `Raumnummer` nicht mit in die `SELECT` Anweisung aufgenommen.

Bei wörtlicher Beschreibung der Datenbank zeigte Text2SQL.ai wieder 3 Fehler bei denen zu viel Information geliefert wurde ein Beispiel ist die Abfrage „Zeige mir die Anzahl der Vorlesungen pro Professor“. Hier wurde eine `ORDER BY` Klausel hinzugefügt, obwohl nicht danach gefragt wurde.



Bei wörtlicher Beschreibung der Datenbank hatte ChatGPT in 5 Fällen das Problem, zu viele Informationen zu liefern in allen Fällen wurde eine Order BY Klausel hinzugefügt obwohl nicht danach gefragt wurde. Ein Beispiel hierfür ist die Abfrage „Zeige mir alle Vorlesungen mit Name und die Raumnummer, in denen sie stattfinden“. Diese Tendenz zu umfassenderen Ergebnissen kann in explorativen Kontexten nützlich sein, in spezifischen Abfragekontexten jedoch problematisch. Die Unterschiede in der Präzision zwischen der DDL-Form und der beschreibenden Wortform könnten die Art und Anzahl der Fehler in den generierten SQL-Abfragen beeinflussen. DDL-Statements bieten eine präzise und strukturierte Definition der Datenbanktabellen und ihrer Beziehungen. Diese exakte Beschreibung der Datenstruktur, einschließlich der Datentypen und Constraints, hilft dabei, klare und spezifische SQL-Abfragen zu generieren. Beide Systeme profitierten wahrscheinlich von der Präzision und Klarheit der DDL-Statements, was zu weniger Missverständnissen und Fehlern führte. Interessanterweise zeigte text2SQL.ai bei der wortwörtlichen Beschreibung einen Fehler weniger als bei der Angabe der DDL-Statements. Das deutet darauf hin, dass text2SQL.ai möglicherweise besser mit weniger strukturierten, natürlichsprachlichen Beschreibungen zurechtkommt und die Kontextinformationen besser verarbeiten kann. ChatGPT hingegen hatte bei der wortwörtlichen Beschreibung mehr Fehler als bei den DDL-Statements. Dies legt nahe, dass ChatGPT stärker von der Präzision und Struktur der DDL-Statements profitiert, um genaue SQL-Abfragen zu generieren. Die beschreibende Wortform bietet eine allgemeinere Übersicht der Datenbankstruktur. Während sie für das menschliche Verständnis ausreichend sein mag, können Details fehlen, die für die exakte Generierung von SQL-Abfragen wichtig sind. Ungenauigkeiten oder fehlende Details in der beschreibenden Wortform könnten zu den beobachteten Logikfehlern und überflüssigen Informationen geführt haben. Beide Systeme zeigten eine hohe Genauigkeit in der Generierung von SQL-Abfragen, wobei Text2SQL.ai mit 93 korrekt beantworteten Testfällen geringfügig besser abschnitt als ChatGPT mit 84 korrekt beantworteten Testfällen. Diese Differenz könnte jedoch statistisch nicht signifikant sein und bedarf weiterer Untersuchungen mit einer größeren Stichprobe. Bemerkenswert ist, dass beide Systeme keine Syntaxfehler aufwiesen. Dies zeigt, dass beide Modelle in der Lage sind, syntaktisch korrekte SQL-Abfragen zu generieren, was eine wichtige Grundlage für die weitere Bewertung der Logik und Genauigkeit der Abfragen ist.

Für zukünftige Arbeiten wäre es sinnvoll, die Anzahl und Vielfalt der Testfälle zu erweitern, um feinere Unterschiede in der Leistung beider Systeme zu identifizieren. Zusätzliche Trainingsdaten, die häufige Logikfehler adressieren, könnten beiden Systemen helfen, ihre Genauigkeit weiter zu verbessern. Weitere Untersuchungen könnten sich auf spezifische

Anwendungsfälle konzentrieren, um die Stärken und Schwächen der Systeme in unterschiedlichen Kontexten zu bewerten. Diese detaillierte Analyse zeigt, dass sowohl ChatGPT als auch Text2SQL.ai beeindruckende Fähigkeiten in der Generierung von SQL-Abfragen haben, jedoch mit unterschiedlichen Stärken und Schwächen, die je nach Anwendungsfall berücksichtigt werden sollten. Die genauen Ergebnistabellen befinden sich im Anhang, zusammen mit Screenshots der Ausgabe der jeweiligen KIs.

## **6 Diskussion**

### **6.1 Bewertung der Ergebnisse**

Die Ergebnisse dieser Untersuchung zeigen deutlich, dass beide getesteten KI-Modelle – ChatGPT und Text2SQL.ai – in der Lage sind, eine hohe Genauigkeit bei der Erstellung von SQL-Statements zu erreichen. Diese Erkenntnis hebt die Potenz von KI-Modellen in der automatisierten SQL-Generierung hervor, insbesondere wenn es darum geht, syntaktisch korrekte und funktional genaue Abfragen zu erstellen.

Darüber hinaus deuten die Ergebnisse darauf hin, dass die Verbesserung der Fähigkeit von KI-Modellen, logische Fehler zu minimieren und ungefragt zu viele Informationen zu liefern ein entscheidender Schritt in ihrer Weiterentwicklung sein wird. Dies könnte durch erweiterte Trainingsdaten, die spezifischere und komplexere Abfragen enthalten, oder durch fortschrittlichere Lernmethoden erreicht werden. Zudem könnten hybride Ansätze, die regelbasierte Systeme mit maschinellem Lernen kombinieren, dazu beitragen, die Präzision und Zuverlässigkeit der SQL-Generierung weiter zu erhöhen.

Zusammengefasst zeigen die Ergebnisse dieser Untersuchung, dass KI-Modelle wie ChatGPT und Text2SQL.ai bereits heute eine wertvolle Unterstützung bei der Erstellung von SQL-Statements darstellen. Sie unterstreichen jedoch auch die Notwendigkeit weiterer Forschung und Entwicklung, um die Modelle in ihrer Fähigkeit zu stärken.

### **6.2 Kritische Reflexion der KI-Integration in SQL-Prozesse**

Obwohl die Integration von KI in SQL-Prozesse vielversprechend ist, gibt es mehrere Herausforderungen und potenzielle Risiken. Eine der Hauptherausforderungen ist die Sicherstellung der Datenintegrität und -sicherheit, insbesondere wenn KI-Modelle autonom arbeiten. Die Datenintegrität muss gewährleistet sein, damit alle Daten korrekt und konsistent bleiben, selbst wenn KI-Modelle komplexe Abfragen und Datenmanipulationen durchführen. Dies erfordert, dass die KI-Modelle strengen Validierungsprozessen unterzogen werden, um sicherzustellen, dass sie keine fehlerhaften oder unvorhergesehenen Änderungen an den Daten vornehmen (Kaufmann & Meier, 2023, S. 11,141).

Ein weiterer wichtiger Aspekt der Sicherheit betrifft den Umgang mit sensiblen Daten von Unternehmen. In vielen Branchen, wie beispielsweise der Finanz- und Gesundheitsbranche, werden große Mengen hochsensibler Daten verarbeitet, die strengen

Datenschutzbestimmungen unterliegen. Der Einsatz von KI-gestützten Systemen zur Verarbeitung dieser Daten erfordert zusätzliche Sicherheitsvorkehrungen, um unbefugten Zugriff und Datenlecks zu verhindern. KI-Modelle müssen so konzipiert und implementiert werden, dass sie den Datenschutzbestimmungen entsprechen und gleichzeitig die Vertraulichkeit, Integrität und Verfügbarkeit der Daten gewährleisten (Wecke, 2024a, S. 28–32).

Ein spezifisches Risiko besteht darin, dass KI-Modelle selbst zu Zielen von Cyberangriffen werden könnten. Angreifer könnten versuchen, die Modelle zu manipulieren, um unbefugten Zugriff auf Daten zu erhalten oder die Funktionsweise der Modelle zu stören. Daher müssen robuste Sicherheitsmaßnahmen, wie Verschlüsselung, Authentifizierung und kontinuierliche Überwachung, implementiert werden, um solche Angriffe zu verhindern. Darüber hinaus sollten Mechanismen zur Anomalieerkennung integriert werden, um verdächtige Aktivitäten frühzeitig zu erkennen und zu stoppen (Wecke, 2024b, S. 19–20).

Ein weiteres Risiko besteht in der Abhängigkeit von der KI, die zu einem Verlust an menschlicher Expertise führen könnte. Wenn Unternehmen zunehmend auf KI-gestützte Systeme zur Erstellung und Überprüfung von SQL-Statements setzen, könnten menschliche Datenbankadministratoren und Entwickler ihre Fähigkeiten und ihr Verständnis für die zugrunde liegenden Datenbankstrukturen und -prozesse verlieren. Dies könnte problematisch sein, wenn die KI-Systeme ausfallen oder unerwartete Fehler auftreten, da es dann an qualifiziertem Personal mangelt, das in der Lage ist, diese Probleme zu beheben (Ahmad et al., 2023, S. 2–11).

## **7 Zukunftsperspektiven und Potenziale**

Die Zukunft der Künstlichen Intelligenz (KI) im Datenmanagement ist vielversprechend und birgt enorme Potenziale für verschiedenste Anwendungsbereiche. Mit der stetigen Weiterentwicklung von KI-Technologien und -Methoden können Unternehmen in den kommenden Jahren mit signifikanten Verbesserungen in der Effizienz, Genauigkeit und Sicherheit ihrer Datenverarbeitungssysteme rechnen.

### **7.1 Herausforderungen und Risiken der KI-Integration in Datenbanksysteme**

Die Integration von Künstlicher Intelligenz (KI) in Datenmanagementsysteme bringt zahlreiche Vorteile, aber auch erhebliche Risiken mit sich. Es ist wichtig, diese Risiken zu verstehen und entsprechende Maßnahmen zu ergreifen, um sie zu minimieren. Hier sind einige der Hauptprobleme, die bei der Implementierung von KI in Datenmanagementsystemen auftreten können, basierend auf aktuellen Quellen:

#### **1 Datenqualität und -zuverlässigkeit**

Eine der größten Herausforderungen bei der Nutzung von KI im Datenmanagement ist die Sicherstellung der Datenqualität. Die Leistungsfähigkeit von KI-Systeme ist nur so gut wie die Qualität ihrer zugrundeliegenden Daten, die sie verarbeiten. Wenn die Daten fehlerhaft, unvollständig oder verzerrt sind, können die Ergebnisse der KI-Modelle ebenfalls fehlerhaft sein. Dies kann zu falschen Vorhersagen, ungenauen Analysen und fehlerhaften Geschäftsentscheidungen führen (Ehrlinger & Wöß, 2022)

#### **2 Sicherheitsrisiken**

KI-Systeme im Datenmanagement sind anfällig für verschiedene Sicherheitsrisiken. Dazu gehören Datenlecks, Hacking und Missbrauch von Daten. Angreifer könnten KI-Modelle manipulieren, um unbefugten Zugang zu sensiblen Informationen zu erhalten oder um das System zu stören. Es ist daher wichtig, robuste Sicherheitsmaßnahmen zu integrieren, um diese Bedrohungen zu minimieren (Hintersdorf, Struppek, & Kristian, 2024)

#### **3 Ethik und Bias**

KI-Modelle können unbeabsichtigte Voreingenommenheiten (Bias) in den Daten verstärken. Wenn die Trainingsdaten Vorurteile enthalten, kann das Modell diskriminierende Entscheidungen treffen. Dies kann schwerwiegende ethische und rechtliche Konsequenzen

haben, insbesondere in Bereichen wie Personalwesen, Strafjustiz und Finanzdienstleistungen. Es ist entscheidend, Mechanismen zur Erkennung und Korrektur von Bias in KI-Modellen zu entwickeln (Ferrara, 2023)

#### **4 Mangel an Transparenz**

KI-Modelle, insbesondere Deep-Learning-Modelle, sind oft als "Black Boxes" bekannt, weil ihre Entscheidungsprozesse für Menschen schwer nachvollziehbar sind. Dieser Mangel an Transparenz kann das Vertrauen in die KI-Systeme beeinträchtigen und es schwierig machen, ihre Entscheidungen zu erklären oder zu rechtfertigen. Es ist wichtig, Methoden zur Verbesserung der Interpretierbarkeit und Transparenz von KI-Modellen zu entwickeln (Qamar & Bawany, 2023).

### **7.2 Weitere Anwendungsbereiche von Künstlicher Intelligenz**

Künstliche Intelligenz (KI) hat in den letzten Jahrzehnten immense Fortschritte gemacht und findet heute Anwendung in einer Vielzahl von Bereichen. Diese Anwendungen erstrecken sich über verschiedene Industrien und haben das Potenzial, sowohl die Effizienz zu steigern als auch innovative Lösungen für komplexe Probleme zu bieten.

Ein bedeutender Anwendungsbereich der KI ist die Gesundheitsbranche. KI wird zur Diagnose und Prognose von Krankheiten genutzt, indem sie umfangreiche medizinische Daten auswertet und Muster erkennt, die Ärzte möglicherweise übersehen. Studien haben gezeigt, dass KI-Algorithmen bei der Erkennung von Krebs, insbesondere bei der Analyse von Mammographien und Hautkrebs, eine Genauigkeit erreichen können, die mit der von menschlichen Experten vergleichbar ist (Esteva et al., 2017).

Im Finanzsektor revolutioniert KI die Art und Weise, wie Finanzinstitute arbeiten. Algorithmen für maschinelles Lernen werden zur Betrugserkennung eingesetzt, indem sie ungewöhnliche Transaktionsmuster in Echtzeit identifizieren (Ngai, Hu, Wong, Chen, & Sun, 2011). Zudem nutzen Investmentfirmen KI, um Handelsstrategien zu entwickeln und Markttrends vorherzusagen. Durch die Analyse vergangener Daten und die Anwendung komplexer Modelle können KI-Systeme Handelsentscheidungen automatisieren und so die Effizienz und Rentabilität erhöhen (Heaton, Polson, & Witte, 2018, S. 18–19).

Diese Beispiele verdeutlichen die breite Palette der Anwendungsbereiche von KI und das transformative Potenzial dieser Technologie. Die kontinuierliche Forschung und Entwicklung

in diesem Bereich wird voraussichtlich noch mehr innovative Anwendungen hervorbringen, die verschiedene Aspekte unseres Lebens verbessern und revolutionieren können.

## **8 Fazit**

### **8.1 Zusammenfassung der wichtigsten Erkenntnisse**

Die Untersuchung hat gezeigt, dass KI-Modelle wie ChatGPT und Text2SQL.ai in der Lage sind, eine hohe Genauigkeit bei der Erstellung von SQL-Statements zu erreichen. Während beide Modelle ihre Stärken und Schwächen haben, bietet Text2SQL.ai eine leicht bessere Leistung bei der Vermeidung von überflüssigen Informationen und der genauen Interpretation spezifischer Abfragen. Die Ergebnisse unterstreichen das Potenzial der KI, die Effizienz und Genauigkeit in der SQL-Generierung zu verbessern, während sie gleichzeitig die Notwendigkeit weiterer Forschung und Entwicklung aufzeigen.

### **8.2 Beantwortung der Forschungsfragen**

Die Forschungsfragen wurden wie folgt beantwortet: (1) Beide KI-Modelle zeigten eine hohe Genauigkeit in der SQL-Generierung, wobei Text2SQL.ai geringfügig besser abschnitt. (2) Die häufigsten Fehlerkategorien waren Logikfehler und überflüssige Informationen, die auf Missverständnisse bei der Interpretation komplexer Bedingungen zurückzuführen waren. (3) KI-Modelle benötigen präzise und strukturierte Informationen, um optimale Ergebnisse zu erzielen, wobei DDL-Statements eine bessere Basis als beschreibende Wortformen bieten.

### **8.3 Ausblick und Empfehlungen für zukünftige Forschungen**

Zukünftige Forschungen sollten sich auf die Weiterentwicklung der KI-Modelle konzentrieren, insbesondere auf die Verbesserung ihrer Fähigkeit, komplexe logische Beziehungen und Bedingungen korrekt zu interpretieren. Es wäre auch sinnvoll, die Integration von KI in verschiedene Datenbankmanagementsysteme zu untersuchen, um die Anwendbarkeit und Effizienz in unterschiedlichen technischen Umgebungen zu bewerten. Darüber hinaus sollten hybride Ansätze, die regelbasierte Systeme mit maschinellem Lernen kombinieren, erforscht werden, um die Stärken beider Ansätze zu nutzen und die Schwächen zu minimieren.

Zukünftige Forschungen sollten ebenfalls prüfen, inwieweit sich KIs dazu nutzen lassen, die Performance von SQL-Abfragen zu optimieren. Die Optimierung von SQL-Abfragen ist ein wesentlicher Bestandteil der Datenbankverwaltung, da sie die Effizienz der Datenverarbeitung und -abfrage erheblich verbessern kann. KI-Modelle könnten dazu eingesetzt werden, komplexe Abfragepläne zu analysieren und zu optimieren, indem sie automatisch die besten



Indizes auswählen, den Abfragepfad anpassen und potenzielle Engpässe identifizieren. Ein solcher Ansatz könnte die manuellen Bemühungen zur Abfrageoptimierung erheblich reduzieren und gleichzeitig die Leistung der Datenbank erhöhen.

Darüber hinaus sollte erforscht werden, ob und wie KIs in anderen Programmiersprachen effizient genutzt werden können. Die Verwendung von KI in Programmiersprachen wie Python, Java oder C++ könnte neue Möglichkeiten eröffnen, um Anwendungen zu entwickeln, die intelligentere und effizientere Algorithmen enthalten. Insbesondere in der Softwareentwicklung könnte KI genutzt werden, um Code zu analysieren und zu optimieren, Fehler zu identifizieren und Vorschläge zur Verbesserung der Codequalität zu machen.

Insgesamt könnten diese Forschungen dazu beitragen, die Einsatzmöglichkeiten von KI in verschiedenen Bereichen der Informatik weiter zu erweitern und zu vertiefen. Sie könnten auch dazu beitragen, die Akzeptanz und Integration von KI-Technologien in bestehenden Systemen und Prozessen zu erhöhen, indem sie deren Nutzen und Effizienzsteigerung aufzeigen.

## Literaturverzeichnis

Ahmad, S. F., Han, H., Alam, M. M., Rehmat, Mohd. K., Irshad, M., Arraño-Muñoz, M., & Ariza-Montes, A. (2023). Impact of artificial intelligence on human loss in decision making, laziness and safety in education. *Humanities and Social Sciences Communications*, 10(1), 311. <https://doi.org/10.1057/s41599-023-01787-8>

AI in Quality Assurance: From Manual to Autonomous Testing. (o. J.). Abgerufen 25. Juli 2024, von Katalon.com website: <https://katalon.com/resources-center/blog/ai-in-quality-assurance>

Bhaskar, A., Tomar, T., Sathe, A., & Sarawagi, S. (2023, Oktober 20). *Benchmarking and Improving Text-to-SQL Generation under Ambiguity*. arXiv. Abgerufen von <http://arxiv.org/abs/2310.13659>

Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*.

Common Database Management Challenges and How to Fix Them. (o. J.). Abgerufen 25. Juli 2024, von <https://blog.transformhub.com/common-database-management-challenges-and-how-to-fix-them>

Computer Vision—Fraunhofer IAIS. (o. J.). Abgerufen 25. Juli 2024, von Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS website: <https://www.iais.fraunhofer.de/de/geschaeftsfelder/computer-vision.html>

Data Validation Testing: Techniques, Examples, & Tools. (o. J.). Abgerufen 25. Juli 2024, von <https://www.montecarloata.com/blog-data-validation-testing/>

Debugging SQL query logic. (o. J.). Abgerufen 25. Juli 2024, von Metabase | Business Intelligence, Dashboards, and Data Visualization website: <https://www.metabase.com/learn/debugging-sql/sql-logic>

Deep Neural Network—An overview | ScienceDirect Topics. (o. J.). Abgerufen 25. Juli 2024, von <https://www.sciencedirect.com/topics/engineering/deep-neural-network>

Die besten Online-SQL-Editoren. (2024, März 15). Abgerufen 25. Juli 2024, von LearnSQL.de website: <https://learnsql.de/blog/die-besten-online-sql-editoren/>

Dipl.-Ing Luber, S. (2022, Februar 23). Was ist Zero-Shot-Lernen? Abgerufen von BigData Insider website: <https://www.bigdata-insider.de/was-ist-zero-shot-lernen-a-1097200/>

Dong, X., Zhang, C., Ge, Y., Mao, Y., Gao, Y., Chen, lu, ... Lou, D. (2023, Juli 14). *C3: Zero-shot Text-to-SQL with ChatGPT*. arXiv. Abgerufen von <http://arxiv.org/abs/2307.07306>

Dr. habil. Nizar Abdelkafi, Inga Döbel, Johannes David Drzewiecki, Anja Meironke, Andreas Niekler, & Sonja Ries. (o. J.). KÜNSTLICHE INTELLIGENZ (KI) IM UNTERNEHMENSKONTEXT. *FRAUNHOFER-ZENTRUM FÜR INTERNATIONALES MANAGEMENT UND WISSENSÖKONOMIE, IMW*.

Durkin, M. (2023, September 20). KI für die Datenanalyse: Wie die Fusion von KI und SQL Ihren Workflow beschleunigen kann. Abgerufen 24. Juli 2024, von LearnSQL.de website: <https://learnsql.de/blog/ki-fuer-die-datenanalyse-wie-die-fusion-von-ki-und-sql-ihren-workflow-beschleunigen-kann/>

Ehrlinger, L., & Wöß, W. (2022). A Survey of Data Quality Measurement and Monitoring Tools. *Frontiers in Big Data*, 5. <https://doi.org/10.3389/fdata.2022.850611>

Erfassung, V. der heutigen neuen und innovativen A. für künstliche I. C. D. L.-T. zur, Video-, I. und A. verschiedener A. von, CNNs, A. T. A. was sind, Gebildet, W. W. S., KI, und was ist ihre V. zum D. L. E. S. mehr über die G. von, Erfassung, D. L. und C. V. und wie I. T. dies alles möglich macht V. der heutigen neuen und innovativen A. für künstliche I. C. D. L.-T. zur, ... macht, D. L. und C. V. und wie I. T. dies alles möglich. (o. J.). Convolutional Neural Networks (CNN) und Deep Learning. Abgerufen 25. Juli 2024, von Intel website: <https://www.intel.com/content/www/de/de/internet-of-things/computer-vision/convolutional-neural-networks.html>

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639), 115–118. <https://doi.org/10.1038/nature21056>

Ferrara, E. (2023). *FAIRNESS AND BIAS IN ARTIFICIAL INTELLIGENCE: A BRIEF SURVEY OF SOURCES, IMPACTS, AND MITIGATION STRATEGIES*. Abgerufen von <https://arxiv.org/pdf/2304.07683v1>

Groff, J. R., & Weinberg, P. N. (1999). *SQL: The Complete Reference*. Osborne/McGraw-Hill.

Heaton, J. B., Polson, N. G., & Witte, J. H. (2018, Januar 14). *Deep Learning in Finance*. arXiv. Abgerufen von <http://arxiv.org/abs/1602.06561>

Hintersdorf, D., Struppek, L., & Kristian, K. (2024). *Balancing Transparency and Risk: The Security and Privacy Risks of Open-Source Machine Learning Models*. Abgerufen von <https://ar5iv.labs.arxiv.org/html/2308.09490>

Kapelle, F.-J. (o. J.). Daten als wertvolle Ressource: Die Data-Value-Perspektive für Unternehmen. Abgerufen 24. Juli 2024, von Horváth Management Consultants website: <https://www.horvath-partners.com/de/media-center/artikel/daten-als-wertvolle-ressource-die-data-value-perspektive-fuer-unternehmen>

Kaufmann, M., & Meier, A. (2023). *SQL- & NoSQL-Datenbanken: 9. erweiterte und aktualisierte Auflage*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-67092-7>

Kyle Daigle. (2023, November 8). Octoverse: The state of open source and rise of AI in 2023. Abgerufen 24. Juli 2024, von The GitHub Blog website: <https://github.blog/news-insights/research/the-state-of-open-source-and-ai/>

Litzel, N., & Luber, S. (2022, Mai 10). Was ist BERT? Abgerufen 25. Juli 2024, von BigData-Insider website: <https://www.bigdata-insider.de/was-ist-bert-a-1116088/>

MySQL :: MySQL Query Analyzer. (o. J.). Abgerufen 25. Juli 2024, von <https://www.mysql.com/products/enterprise/query.html>

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559–569. <https://doi.org/10.1016/j.dss.2010.08.006>

Nilsson, N. J. (2009). *The Quest for Artificial Intelligence* (1. Aufl.). Cambridge University Press. <https://doi.org/10.1017/CBO9780511819346>

Ostrowska, K. (2022, April 26). Top 5 MySQL Query Tools (Updated for 2024). Abgerufen 25. Juli 2024, von LearnSQL.com website: <https://learnsql.com/blog/mysql-query-tools/>

Petrovic, M. (2014, April 2). Common SQL syntax errors and how to resolve them. Abgerufen 25. Juli 2024, von SQL Shack—Articles about database auditing, server performance, data recovery, and more website: <https://www.sqlshack.com/sql-syntax-errors/>

Pornphol, P., & Chittayasothorn, S. (2023). Verification of Relational Database Languages Codes Generated by ChatGPT. *Proceedings of the 2023 4th Asia Service Sciences and Software*

*Engineering Conference*, 17–22. Aizu-Wakamatsu City Japan: ACM.  
<https://doi.org/10.1145/3634814.3634817>

Qamar, T., & Bawany, N. Z. (2023). Understanding the black-box: Towards interpretable and reliable deep learning models. *PeerJ Computer Science*, 9, e1629.  
<https://doi.org/10.7717/peerj-cs.1629>

Qin, B., Hui, B., Wang, L., Yang, M., Li, J., Li, B., ... Li, Y. (2022, August 29). *A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions*. arXiv. Abgerufen von <http://arxiv.org/abs/2208.13629>

Roscher, K., Guderitz, A., & Hengl, H.-T. (o. J.). Künstliche Intelligenz (KI) und maschinelles Lernen—Fraunhofer IKS. Abgerufen 25. Juli 2024, von Fraunhofer-Institut für Kognitive Systeme IKS website: <https://www.iks.fraunhofer.de/de/themen/kuenstliche-intelligenz.html>

Ruiz, J. (2024, Februar 22). How AI code generation works. Abgerufen 24. Juli 2024, von The GitHub Blog website: <https://github.blog/ai-and-ml/generative-ai/how-ai-code-generation-works/>

Shani, I. (2023, Juni 13). Survey reveals AI's impact on the developer experience. Abgerufen 24. Juli 2024, von The GitHub Blog website: <https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

Shivakumar, D. (o. J.). Fascinating Best SQL IDE & Tools for Increasing Productivity. Abgerufen 25. Juli 2024, von <https://www.turing.com/kb/best-sql-developer-ide-and-tools>

SQL. (o. J.). Abgerufen 24. Juli 2024, von <https://www.tenmedia.de/de/glossar/sql>

SQL Developer. (o. J.). Abgerufen 25. Juli 2024, von <https://www.oracle.com/database/sqldeveloper/>

Sun, S., Zhang, Y., Yan, J., Gao, Y., Ong, D., Chen, B., & Su, J. (2023, Oktober 16). *Battle of the Large Language Models: Dolly vs LLaMA vs Vicuna vs Guanaco vs Bard vs ChatGPT -- A Text-to-SQL Parsing Comparison*. arXiv. Abgerufen von <http://arxiv.org/abs/2310.10190>

The State of AI in 2023: Latest Releases, Current Trends, and Future Developments. (2023, Juli 24). Abgerufen 24. Juli 2024, von Big Human website: <https://www.bighuman.com/blog/whats-happening-now-in-artificial-intelligence>, <https://www.bighuman.com/blog/whats-happening-now-in-artificial-intelligence>

Was ist Künstliche Intelligenz? KI einfach erklärt. (o. J.). Abgerufen 25. Juli 2024, von Mittelstand-Digital website: <https://digitalzentrum-augsburg.de/kuenstliche-intelligenz-einfach-erklaert/>

Was ist Machine Learning? | Definition, Arten, Beispiele. (o. J.). Abgerufen 25. Juli 2024, von SAP website: <https://www.sap.com/germany/products/artificial-intelligence/what-is-machine-learning.html>

Was sind rekurrente neuronale Netze? | IBM. (2023, Dezember 14). Abgerufen 25. Juli 2024, von <https://www.ibm.com/de-de/topics/recurrent-neural-networks>

Wecke, B. (2024a). *Generative KI als neues Teammitglied im Marketing: Ein Leitfaden für Marketingmanger:innen*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-44179-1>

Wecke, B. (2024b). *Wachstum durch den Einsatz von Generativer KI: Funktionsweise und Anwendungsgebiete im Marketing*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-44041-1>