applied sciences

Systematic Review

The Role of Artificial Intelligence in Computer Science Education: A Systematic Review with a Focus on Database Instruction

Alkmini Gaitantzi and Ioannis Kazanidis *

Department of Informatics, Democritus University of Thrace, 65404 Kavala, Greece; algaita@cs.duth.gr
* Correspondence: kazanidis@cs.duth.gr

Abstract: The integration of artificial intelligence (AI) into computer science (CS) education is evolving, yet its specific application in database instruction remains underexplored. This systematic review analyzes 31 empirical studies published between 2020 and 2025, examining how AI applications support teaching and learning in CS, with an emphasis on database education. Following the PRISMA methodology, the review categorizes AI applications according to instructional design models, roles, actions, benefits, and challenges. Findings indicate that AI tools, particularly chatbots, intelligent tutoring systems, and code generators, effectively support personalized instruction, immediate feedback, and interactive problem-solving across CS and database-specific contexts. However, challenges persist, including AI inaccuracies, biases, student dependency in AI, and academic integrity risks. The review also identifies a shift in programming education as AI reshapes software development practices, prompting a need to align curricula with evolving industry expectations. Despite growing attention to AI applications in programming education, database-related research remains limited. This review highlights the necessity for further empirical investigations specifically in database instruction, including more extensive studies addressing AI-driven pedagogical strategies and their long-term impacts. The results suggest that careful integration of AI tools can complement traditional instruction, emphasizing the critical role of human educators in achieving meaningful and effective learning outcomes.

Keywords: artificial intelligence; generative AI; ChatGPT; computer science education; programming education; database instruction; AI in education; systematic review

check for updates

Academic Editor: Christos Bouras

Received: 23 February 2025
Revised: 31 March 2025
Accepted: 2 April 2025
Published: 3 April 2025

Citation: Gaitantzi, A.; Kazanidis, I. The Role of Artificial Intelligence in Computer Science Education: A Systematic Review with a Focus on Database Instruction. *Appl. Sci.* **2025**, *15*, 3960. https://doi.org/10.3390/app15073960

1. Introduction

The rapid growth of artificial intelligence (AI) has led to a proliferation of groundbreaking possibilities, with AI now being present in many areas of human activity. The development and deployment of generative pre-trained transformers (GPTs), notably OpenAI's ChatGPT (GPT-3.5 version), has attracted much interest. These models are designed to facilitate the generation of meaningful text that resembles human-written content [1]. All AI-based models offer unprecedented capabilities aimed at solving obscure problems, automating certain tasks, and enhancing efficiency.

In recent years, the presence of artificial intelligence has become increasingly pervasive in various aspects of our daily lives, notably including education. The use of AI holds the potential to transform educational practice [2]. It has the potential to revolutionize the way students learn and teachers teach [3] by making learning and teaching more personalized, engaging, and effective. The advent of AI technologies has given rise to sophisticated learning tools that can personalize education and enhance learning outcomes [3]. These technologies hold the potential to transform traditional teaching methodologies by providing personalized learning experiences and real-time feedback to students [3]. For example, AI can generate assignments and presentations. Additionally, it can facilitate class discussions, questions, and personalized feedback [1]. Therefore, AI tools such as ChatGPT have the potential to serve as valuable educational resources.

Technology, artificial intelligence, and the field of computer technology and science, in general, are increasingly penetrating our daily lives. Consequently, there is an imperative for society to adapt to new technologies [4], and there is a great need to educate students in computer science (CS) [5]. The integration of technology in the teaching of CS has become imperative for all individuals, as it is a critical skill set. However, students face difficulties in areas such as algorithms and data structures during the teaching and learning process in computing [6]. Programming is also difficult for some students to understand, especially if they do not have a well-developed algorithmic mindset. Quite a few students describe some computing courses as "dry" and "not very interesting" [5]. The use of artificial intelligence as a technology can facilitate computer science education [5]. For example, these systems can be used to assist a student in solving problems that involve both practical and abstract reasoning [6]. In addition, ChatGPT can be used to perform a range of programming tasks, making it easier for students to learn [1].

Databases (DB) form a distinct and essential subfield of computer science, characterized by specialized concepts and complex problem-solving requirements. Students often face challenges in mastering SQL syntax, query optimization, and relational database principles, necessitating innovative teaching approaches. Interactive learning methods, real-time feedback, and AI-driven explanations could significantly enhance the learning experience by making abstract concepts more accessible. Among AI technologies, chatbots offer a promising solution by providing instant support, generating adaptive exercises, and facilitating personalized learning pathways.

While numerous studies have explored the use of AI and chatbots in education, research on their specific application in computer science, particularly in database instruction, remains limited. The benefits, challenges, and pedagogical implications of integrating AI into database education have yet to be thoroughly examined. This paper aims to bridge this gap by providing a systematic review of AI's role in computer science education, with a particular focus on its impact on database learning. It explores AI's applications, advantages, limitations, and instructional design considerations to ensure its effective and pedagogically sound implementation.

At the same time, the transformative impact of AI on programming as a discipline cannot be overlooked. Beyond its role in education, generative AI has begun reshaping the very nature of software development by automating key tasks such as coding, debugging, and testing. This shift has led to the emergence of "AI-native software development", where developers increasingly act as curators and collaborators with AI systems, rather than solely writing code themselves. As this transformation redefines professional practices, it also necessitates a rethinking of how programming is taught. Educational strategies must adapt to prepare students for a landscape where AI tools are integral to the development process. This review therefore not only considers how AI enhances instructional design, but also examines the broader impact of AI on software engineering and the profession itself, expanding the literature review to incorporate emerging research on AI-assisted development.

1.1. Literature Review

Several reviews have examined the role of artificial intelligence in computer science education, each focusing on different aspects of its application and impact. Table 1 presents the identified reviews in descending order of the examined time range.

Table 1. Reviews on the use of AI in computer science education.

Reference	Number of Sample Papers (Time Range)	Field in Focus	Objective
[3]	54 (2013–2024)	AI and STEM Education	Analysis of how AI and ML can improve educational outcomes, personalize learning experiences, and address critical issues in STEM fields
[5]	28 (2011–2024)	AI and Computer Science Education	Examine how the integration of AI affects learning outcomes in CS education
[2]	115 (2022–2023)	AI and Education	Explore the potential and challenges of using advanced AI models in education
[1]	15 (2012–2023)	AI, ChatGPT, and Education and Programming	Explore the possibilities of ChatGPT for education, programming, content creation, and more
[6]	26 (2009–2021)	ITS, AI, and Computer Science Education	Analysis of the development and use of ITS in CS education, with a focus on AI-based techniques
[4]	25 (2010–2020)	Educational AI and Computational Sciences	Analysis of the impact of Al and computational science on student achievement

Tlili [5] examines how the integration of artificial intelligence affects learning outcomes in computer science education and the potential variables that mediate this effect. Expanding on this, ref. [4] quantitatively and qualitatively analyze the impact of AI and computational science tools on student performance. Their review explores how educational artificial intelligence (EAI) enhances learning, the most common AI applications in education, and their effectiveness across different educational levels.

A more specialized perspective is provided by [6], who focus on the development and use of intelligent tutoring systems (ITS) in computer science education. Their analysis highlights the challenges in advancing ITS research and making these systems widely accessible to students and educators. Additionally, they investigate how artificial intelligence in ITS can be used to teach computer science. Similarly, ref. [3] provide a thorough analysis to understand how artificial intelligence and machine learning (ML) can improve educational outcomes, personalize learning experiences, and address critical issues in STEM education.

Beyond broad educational applications, some reviews examine more specific AI implementations. For instance, ref. [2] discuss the opportunities and challenges of advanced AI models in education, providing an overview of current views on their potential and limitations. Lastly, ref. [1] focus on the functions of ChatGPT, including reinforcement learning, and various applications and limitations. In this analysis, we learned about the use of ChatGPT in areas such as education, programming, and content production.

1.2. Research Gap

Together, these studies present a comprehensive overview of AI's role in computer science education, yet they reveal critical gaps in the research. While AI has been widely explored in programming education and intelligent tutoring systems, its broader role in computer science instruction remains underexamined. Most existing studies provide a superficial exploration of AI in computer science, without a deeper analysis of its pedagogical frameworks or long-term impact. Notably, only three studies [4–6] primarily focus on computer science education, while others either address computational sciences partially, as in [1,3], or do not explore them at all, as in [2]. This fragmented approach leaves important questions unanswered regarding how AI effectively supports learning and teaching strategies beyond programming.

Another notable gap is the lack of a unified pedagogical framework for integrating AI into IT education, which raises questions about the consistency and validity of existing approaches.

Another critical research gap concerns AI-assisted programming and the educational shift it brings [4]. While existing studies explore AI's role in programming education, they often fail to account for the broader transformation occurring in computer science as a discipline [5,6]. The emergence of generative AI has ushered in novel paradigms, such as "AI-native software development", where programmers transition from their traditional roles as coders to becoming AI curators [1,3]. However, there is limited research on how these changes should be reflected in educational frameworks. Specifically, the pedagogical implications of AI-driven automation in debugging, testing, and software development remain under-explored [2,6]. While there has been some examination of AI in the context of intelligent tutoring and adaptive learning, there is a need for more research on how AI can support students in transitioning from conventional programming practices to AI-augmented development workflows [1–3,6].

Furthermore, despite the significance of databases as a core discipline in computer science, none of the reviewed studies explicitly examine AI's role in database instruction. While programming education has received substantial attention, databases, relational modeling, and query optimization remain underexplored areas in AI-driven education. Given the complexity of database concepts and students' difficulties in mastering SQL, AI-driven learning tools, such as chatbots or automated tutors, could provide valuable support, yet their effectiveness in this domain has not been systematically evaluated.

To address these gaps, this systematic review aims to analyze existing applications of AI in CS education, identify its pedagogical uses, and evaluate its benefits and limitations, with a focus on database education.

In order to clearly define the scope of the systematic review and to address the issues involved, the following research questions have been formulated:

RQ1 What research methodologies have been used to evaluate the impact of AI in computer science education?
RQ2 How is artificial intelligence integrated into the instructional design of computer science education?
RQ3 What are the roles and actions of using generative AI in the instruction of computer science?
RQ4 What are the key benefits, challenges, and limitations of generative AI in computer science education?
RQ5 How does AI-assisted programming transform educational approaches?
RQ6 How is artificial intelligence applied in database instruction?

While only RQ6 explicitly addresses the role of AI in database instruction, the broader research questions (RQ1–RQ5) were designed to encompass the entire context of computer science education, including database instruction as an integral component. Thus, database instruction is inherently considered within RQ1–RQ5, given that methodologies, instructional designs, roles, benefits, and challenges identified broadly in computer science are also directly applicable to database education.

2. Methodology

This systematic review follows the PRISMA 2020 guidelines to ensure transparency and reproducibility. The review process included the following key steps: defining research questions, establishing inclusion and exclusion criteria, conducting a comprehensive literature search, selecting relevant studies, extracting data, and synthesizing results. A systematic screening process ensured reliability by removing duplicates, screening titles and abstracts, and evaluating full texts. The focus was on the analysis of AI applications in the domain of computer science education, with a particular emphasis on their role in database instruction.

Studies published between 2020 and 2025 were included, provided that they focused on AI in computer science and database education and employed empirical research methods such as experimental studies or mixed-method approaches. Inclusion and exclusion criteria are presented in Table 2. Exclusion criteria included studies published before 2020, non-English publications, and those lacking a clear research methodology. The PRISMA flowchart is used to visually represent the selection process, illustrating the number of records identified, screened, and ultimately included in the review.

Table 2. Inclusion and exclusion criteria.

Inclusion Criteria	Exclusion Criteria
Studies published between 2020 and 2024	Studies published before 2020
Studies in English language	Studies in a language other than English
Studies using AI in education and CS and database teaching and learning Studies using AI tools in computer engineering or programming development	Studies that examine the use of AI theoretically rather than through experimental application
Studies using AI with research methods	

The collected data were then categorized based on three distinct research methods: empirical studies, qualitative analyses, and systematic reviews. Additionally, the AI applications were categorized into three groups: programming, databases, and computational thinking, among others. The educational design was analyzed, identifying AI's function as a tutor, assistant, or automation tool. The reported benefits, challenges, and limitations of AI in computer science and databases education were also extracted.

The literature search was conducted in February 2025, with the final data extraction completed in March 2025. All searches were performed using two major academic databases, Google Scholar and Scopus, to ensure a comprehensive collection of relevant studies.

The search strategy involved predefined Boolean search queries incorporating key terms related to AI, computer science, database instruction, and education. The exact keyword combinations used for both databases were ("generated AI" OR "Artificial Intelligence" OR "chatbots" or "AI" OR "LLM" OR "LLMs" OR "AI-driven" OR "AI-augmented" OR "AI-assisted") AND ("database" OR "Computer Science" OR "Relational Databases" OR "Informatics" or "Computational Thinking" OR "Programming" OR "Algorithms") AND ("education" OR "training" OR "Learning" OR "Instruction" OR "Tutoring" OR "Intervention"). The Boolean search queries were applied explicitly to the title, abstract, and keyword fields when searching the Scopus database. Due to the inherent limitations of the Google Scholar platform, the same Boolean queries were applied broadly across the available metadata (including titles and abstracts). Given that Google Scholar does not allow fine-grained control of search fields, results from this source were limited to the first 1000 records returned by relevance ranking.

The initial screening of titles and abstracts, as well as the subsequent full-text review, was performed independently by two authors. Disagreements regarding the inclusion or exclusion of studies were resolved through detailed discussion until consensus was reached. Although inter-rater reliability was not calculated, this consensus-driven approach helped ensure consistency and accuracy in the selection process.

Once relevant studies were selected, data extraction followed a standardized process. The extracted data included study details (authors, year, publication type), research methods, AI applications in education, instructional approaches, reported benefits and challenges, and study outcomes. The reviewers worked separately during the initial extraction phase to ensure accuracy and reduce bias. Any discrepancies in extracted data were discussed and resolved through consensus. No automation tools were used in this process. Where necessary, additional clarifications were obtained by cross-referencing details from the original study reports.

The review focused on extracting data related to AI applications in programming and database instruction, specifically examining instructional approaches, AI roles, learning benefits, and challenges. Outcomes sought included improvements in student engagement, personalized learning, and skill development, as well as challenges like AI accuracy, biases, and academic integrity concerns. All relevant results reported in the selected studies were considered, prioritizing those with quantitative evidence of AI effectiveness or direct pedagogical comparisons. Studies lacking explicit educational outcomes or focusing solely on AI technology development without instructional application were excluded from synthesis.

In addition to educational outcomes, data were extracted on study characteristics, including participant demographics (e.g., student level, sample size), AI intervention types, and instructional settings. If key details were missing or unclear, assumptions were made based on the study context, or information was inferred from related sections. Studies lacking essential methodological details were excluded from the synthesis to maintain data reliability.

A risk of bias assessment was performed on criteria such as sample size, research design, data collection methods, and potential conflicts of interest, and each study was classified by the two reviewers independently assigning a low, moderate, or high risk of bias classification.

Due to variability in study designs and outcome reporting, a meta-analysis was not conducted and findings were synthesized narratively. Studies were deemed eligible for synthesis based on their relevance to AI applications in computer science education, particularly in programming and database instruction. Each study's intervention characteristics, such as AI tool type, instructional approach, and target learning outcomes, were tabulated and compared against the predefined inclusion criteria. Studies that focused solely on AI development without an educational application were excluded. Eligible studies were then grouped by AI roles, instructional strategies, and reported learning outcomes, allowing for a structured synthesis of findings.

Synthesis of the data was then used to address the research questions. Results were organized into tables and figures, summarizing study characteristics, AI applications, instructional approaches, and reported outcomes. Studies were grouped based on AI roles, benefits, challenges, and educational contexts to facilitate comparison. Visual representations, such as charts, were used to highlight trends and distributions across the included studies.

The risk of bias due to missing results was assessed by examining publication sources and the completeness of reported outcomes. Studies lacking sufficient methodological details or clear reporting of findings were noted as having a potential reporting bias. No formal statistical methods were applied to detect publication bias, but the narrative synthesis accounted for possible underreporting by considering study limitations.

3. Results

The study selection process followed the PRISMA 2020 guidelines to ensure a systematic and transparent approach. It was conducted in multiple stages, which are visually presented in the PRISMA flowchart (Figure 1). The initial search across Google Scholar and Scopus yielded 37,800 results across both databases. It should be noted that Google Scholar provides only the first 1000 records of the list. After removing duplicates, 200 articles remained for title and abstract screening. At this stage, 120 studies were selected based on the inclusion criteria, focusing on AI applications in computer science education. A full-text assessment was then conducted, leading to the exclusion of 31 studies that did not meet the eligibility criteria. Reasons for exclusion included a lack of empirical research (n = 12), studies with a broad focus on general AI in education rather than computer science (n = 10), and papers that lacked relevant findings on AI's impact on learning outcomes (n = 9). Ultimately, 31 studies were included in this review for synthesis.

The 31 studies included in this systematic review were categorized based on their publication year and type (scientific journal or conference proceedings), as presented in Table 3. The analysis revealed that 17 studies (55%) were published in scientific journals, indicating long-term contributions to the field. The remaining 14 studies (45%) were conference papers published in peer-reviewed conference proceedings. This suggests that research on AI in computer science education is actively evolving, with conferences serving as a platform for early dissemination of findings and emerging trends.

Table 3. Summary of study year and type.

Year	Reference	Title	Type of Publication	Journal/Conference Title
2024	[7]	Iris: An AI-Driven Virtual Tutor for Computer Science Education	Conference Proceedings	Innovation and Technology in Computer Science Education
2024	[8]	A Comparative Study of AI-Generated (Gpt-4) And Human-Crafted MCQs in Programming Education	Conference Proceedings	26th Australasian Computing Education Conference
2024	[9]	Teachers' And Students' Perceptions of AI-Generated Concept Explanations: Implications for Integrating Generative AI in 2023 Computer Science Education	Scientific Journal	Computers and Education: Artificial Intelligence
2024	[10]	Evaluating the Effectiveness of Large Language Models in Introductory Computer Science Education: A Semester-Long Field Study	Conference Proceedings	Eleventh ACM Conference on Learning@ Scale

Table 3. *Cont.*

Year	Reference	Title	Type of Publication	Journal/Conference Title
2024	[11]	Enhancing Educational Efficiency: Generative AI Chatbots and DevOps in Education 4.0	Scientific Journal	Computer Applications in Engineering Education
2024	[12]	Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education	Conference Proceedings	46th International Conference on Software Engineering: Software Engineering Education and Training
2024	[13]	Would ChatGPT-facilitated programming mode impact college students' programming behaviors, performances, and perceptions? An empirical study	Scientific Journal	International Journal of Educational Technology in Higher Education
2024	[14]	Chatbot-Based Learning Platform for SQL Training	Scientific Journal	International Journal of Interactive Multimedia and Artificial Intelligence
2023	[15]	My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises	Conference Proceedings	25th Australasian Computing Education Conference
2023	[16]	On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree?	Conference Proceedings	2023 Conference on Innovation and Technology in Computer Science Education
2023	[17]	Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors	Conference Proceedings	2023 ACM Conference on International Computing Education Research–Volume 2
2023	[18]	The Robots are Here: Navigating the Generative AI Revolution in Computing Education	Conference Proceedings	2023 Working Group Reports on Innovation and Technology in Computer Science Education
2023	[19]	ChatGPT for Education and Research: Opportunities, Threats, and Strategies	Scientific Journal	Applied Sciences
2023	[20]	Information Systems Professors' Perspectives on AI-Assisted Programming	Scientific Journal	Issues in Information Systems
2023	[21]	The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation	Scientific Journal	Computers and Education: Artificial Intelligence
2023	[22]	Augmented intelligence in programming learning: Examining student views on the use of ChatGPT for programming learning	Scientific Journal	Computers in Human Behavior: Artificial Humans

Table 3. *Cont.*

Year	Reference	Title	Type of Publication	Journal/Conference Title
2023	[23]	ChatbotSQL: Conversational agent to support relational database query language learning	Scientific Journal	SoftwareX
2022	[24]	The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming	Conference Proceedings	24th Australasian Computing Education Conference
2020	[25]	e-JAVA Chatbot for Learning Programming Language: A Post-Pandemic Alternative Virtual Tutor	Scientific Journal	International Journal of Emerging Trends in Engineering Research
2023	[26]	An Experiment on Leveraging ChatGPT for Online Teaching and Assessment of Database Students	Conference Proceedings	IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)
2024	[27]	An LLM-Driven Chatbot in Higher Education for Databases and Information Systems	Scientific Journal	IEEE Transactions on Education
2024	[28]	Exploring the Effectiveness of AI-Enabled Microlearning in Database Design and Programming Course	Conference Proceedings	IEEE Frontiers in Education Conference (FIE)
2025	[29]	Teaching Tip: Incorporating AI Tools Into Database Classes	Scientific Journal	Journal of Information Systems Education
2024	[30]	Integrating LLMs into Database Systems Education	Conference Proceedings	Proceedings of the 3rd International Workshop on Data Systems Education: Bridging education practice with education research
2024	[31]	Programming education and learner motivation in the age of generative AI: student and educator perspectives	Scientific Journal	Information and Learning Sciences
2023	[32]	Computing Education in the Era of Generative AI	Scientific Journal	Communications of the ACM
2024	[33]	Kattis vs ChatGPT: Assessment and Evaluation of Programming Tasks in the Age of Artificial Intelligence	Conference Proceedings	14th Learning Analytics and Knowledge Conference
2024	[34]	Authoring Programming Exercises for Automated Assessment Assisted by Generative AI	Conference Proceedings	5th International Computer Programming Education Conference (ICPEC 2024)
2025	[35]	The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance	Scientific Journal	International Journal of STEM Education

Table 3. *Cont.*

Year	Reference	Title	Type of Publication	Journal/Conference Title
2024	[36]	AI-assisted programming and AI literacy in Computer Science Education	Scientific Journal	Effective Practices in AI Literacy Education: Case Studies and Reflections
2023	[37]	The Psychological Effects of AI-Assisted Programming on Students and Professionals	Conference Proceedings	2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)

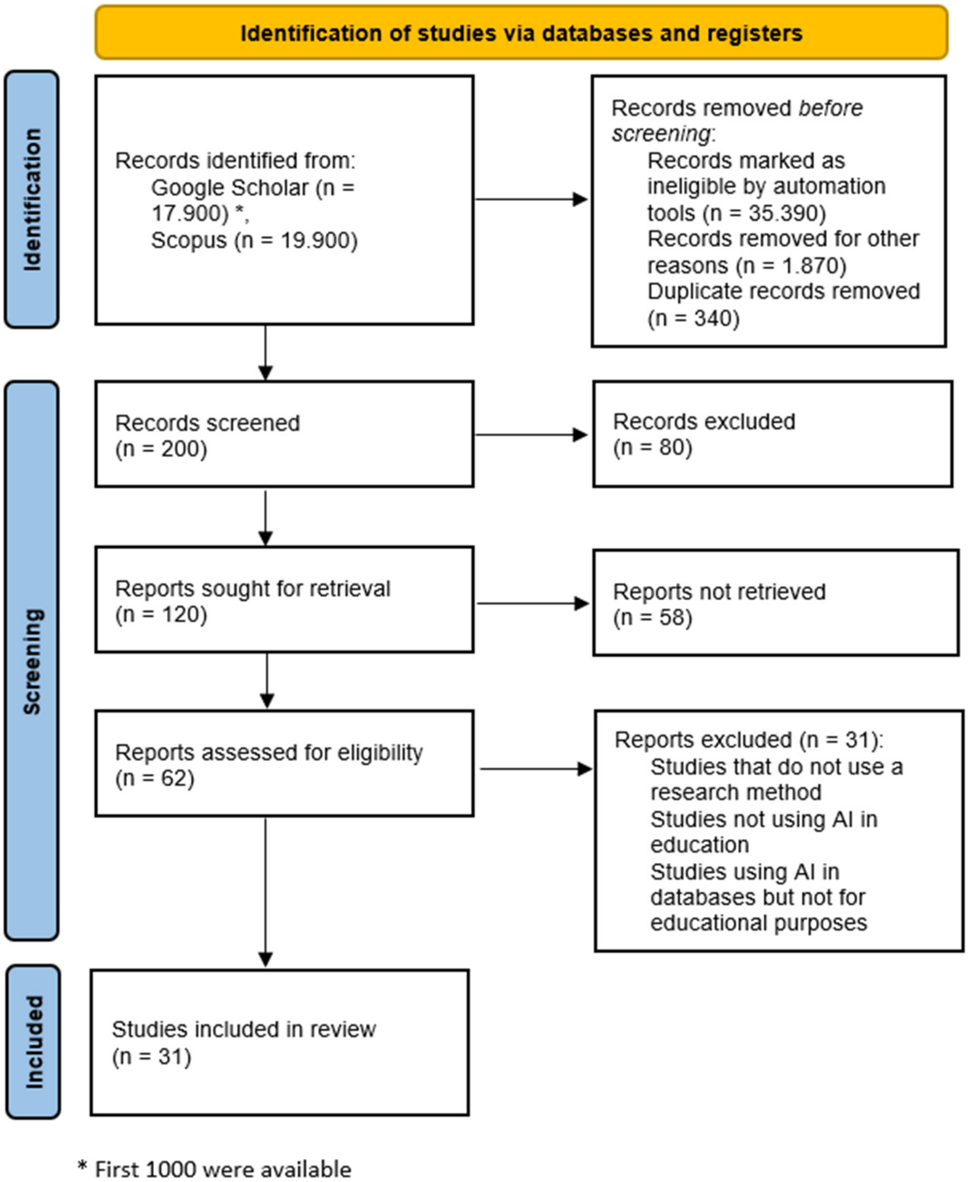

Figure 1. Prisma flowchart.

3.1. Research Method

The analysis of the thirty-one (31) empirical studies included in this systematic review reveals a variety of research methods. Of the total number of studies, nine (9) used a quantitative research methodology [11,12,14,15,23,24,27,33,35]. They usually involved experimental design, data collection through trials, surveys and analysis, and numerical–statistical evaluation of the performance, interactions, and behaviors between participants and AI tools. In addition, three (3) studies used a qualitative research methodology approach [26,30,36]. These studies involved thematic data analysis, observational data, and discussions. Furthermore, nineteen (19) studies used a mixed methodology approach [7–10,13,16–22,25,28,29,31,32,34,37]. These studies combined both quantitative (e.g., statistical analyses) and qualitative methods (e.g., thematic data analysis). Figure 2 provides an overview of the research methodologies adoption by the analyzed studies.

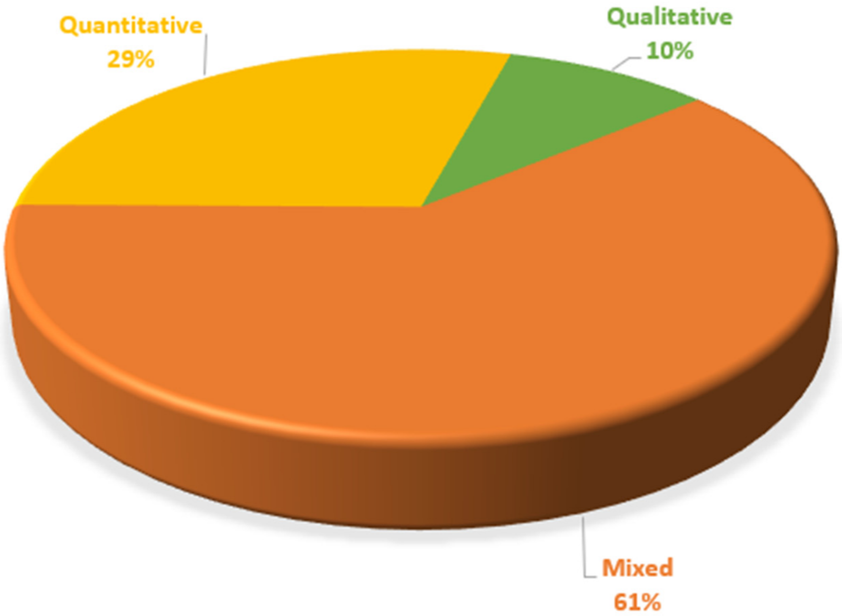

Figure 2. Research methodologies applied in the selected studies.

3.2. Instructional Design

This section presents the instructional design of the 31 studies along with the AI's educational applications in computer science and database instruction. Table 4 provides a comprehensive summary of the research design of these studies, outlining key aspects such as learning topic, research objectives, instructional approaches, AI tools used, participant characteristics, and experiment duration. This structured analysis offers insights into how AI has been integrated into computer science education and highlights trends in its implementation. However, some studies had small sample sizes or lacked details on participant demographics and experiment duration, which affects the reliability and generalizability of the findings.

The instructional design approaches used in the selected studies highlight the diverse ways in which AI is integrated into computer science education. As shown in Figure 3, most studies focus on AI applications in programming education, with significantly fewer addressing areas such as database instruction or computational thinking.

In terms of learning approaches, Figure 4 illustrates that the predominant instructional strategies include problem-based learning (PBL) and collaborative learning. These approaches encourage active student engagement and promote critical thinking and problem solving in real-world learning contexts. Additionally, experiential learning, project-based learning and inquiry-based learning models are frequently applied, as they provide hands-on experiences that enhance students' practical skills and deepen their understanding of theoretical concepts.

Table 4. Research design, AI tools, and instructional approaches.

Reference	Topic	Research Objectives	Instructional Approach	AI Tool	Participants (No./Type)	Duration
[7]	Programming	Reducing the cognitive load on students and more seamless, accessible, and effective interaction through Iris	Problem-Based Learning (PBL), Inquiry-Based Learning (IBL)	Iris, an AI-based virtual teacher designed to support CS education	n/m	10 days
[25]	JAVA Programming	Evaluation of a virtual teaching tool to support students with JAVA programming	PBL, Competency-Based Learning (CBL)	e-JAVA chatbot	43 first semester students	n/m
[8]	Python Programming	Investigate whether and how GPT-4 could generate high-quality MCQ assessments for higher education programming courses	CBL, Drill and Practice (DP)	GPT-4	Students in Python courses	n/m
[15]	Programming	Investigate how well Codex performs in complex programming problems used in the CS2 exam	PBL, CBL	Codex	264 students in the CS2 course	4 h
[24]	Programming	Investigating how Codex works in typical introductory programming exercises, comparing its performance with that of real students	PBL, IBL	Codex	71 students in CS1 course	n/m
[9]	Programming basics	Compared perceived effectiveness of AI-generated concept explanations vs. teacher-generated explanations, focusing on the programming concepts of sequence, choice, and repetition	Experiential Learning (EL), Collaborative Learning (CL)	GPT-3.5	11 elementary school teachers and 70 6th grade students	n/m
[10]	Computer Science and Python Programming	Evaluating the practicality of large language models (LLM) in programming learning frameworks, and understanding their long-term impact on learning and teaching methodologies	PBL, DP	CodeTutor, a web application that employs GPT-3.5 model	42 students in an introductory CS course	1 semester

Table 4. *Cont.*

Reference	Topic	Research Objectives	Instructional Approach	AI Tool	Participants (No./Type)	Duration
[16]	Computer Security	Evaluation of the influence of ChatGPT in university education, with a main focus on computer security	PBL, Project-Based Learning	ChatGPT	n/m	n/m
[11]	Content Management Systems and Web Development	Evaluation teaching effectiveness before and after the adoption of AI chatbots and benefits analysis	Project-Based Learning, CL, EL	ChatGPT 3.5	49 students from 4 academic years CS courses	15 weeks
[12]	AI Detectors and Computer Science	Evaluation of the performance of different AIGCs in different contexts and syntax variants	PBL, EL, Project-Based Learning	ChatGPT	n/m	n/m
[17]	Programming	Integrated evaluation of state-of-the-art LLM for different scenarios in programming education and comparison of their performance against trainers	PBL, CL	ChatGPT (based on GPT-3.5), GPT-4	n/m	n/m
[18]	LLMs and Generative AI in IT education	Exploring computer education to set an agenda for researchers and gather effective practices for teachers	Project-Based Learning, CL	ChatGPT, GPT-3, GPT-4, and similar systems	57 trainers and 171 students in CS or related fields	n/m
[19]	Programming	Exploring the use of ChatGPT in learning and teaching	EL, CL	ChatGPT	Students and teachers	n/m
[20]	Computer programming and education	Investigating the impact of AI in higher education, especially on information systems and CS teachers	EL, CL	ChatGPT, SQL Query Builder, Polycoder, Turing by Borealis AI	110 CS courses professors	n/m
[13]	Programming Python	Investigating the impact of ChatGPT-facilitated programming on college students' programming behaviors, performance, and cognition	IBL, PBL	ChatGPT	82 students	5 weeks

Table 4. *Cont.*

Reference	Topic	Research Objectives	Instructional Approach	AI Tool	Participants (No./Type)	Duration
[21]	Programming	Investigating the impact of programming education using ChatGPT on students' computational thinking skills, self-efficacy, and motivation	Flipped Classroom, EL	ChatGPT (GPT-3.5 architecture)	45 under-graduate students	5 weeks
[22]	Programming	Exploring students' views on the use of ChatGPT in programming	PBL, EL	ChatGPT	Students	8 weeks
[14]	SQL and Relational Database Management Systems	Supporting students in learning and training with SQL using a chatbot-based learning platform	DP, CBL, Game-Based Learning	Chatbot-based learning platform, specifically designed for SQL	59 students in DB course	2 weeks
[23]	SQL and Complex Questions	Supporting standalone learning of SQL language for database queries	PBL, EL, DP	ChatbotSQL, based on IBM Watson Assistant	64 students in DB course	1 year
[26]	SQL education	Evaluate ChatGPT's effectiveness in solving SQL assignments, its potential for enabling cheating in online assessments, and as a grading tool	PBL, CBL, Project-Based Learning	ChatGPT	Students (DB Class)	1 semester
[27]	Databases and information systems	Explore the benefits and challenges of integrating an LLM-driven chatbot, into higher education LMS	IBL, CBL, CL	MoodleBot, a chatbot integrated into the Moodle	46 students	n/m
[28]	Database design and programming course	Explore effectiveness of the AI-enabled microlearning approach and identify the benefits and challenges of using AI chatbot in microlearning	Flipped Classroom, PBL	SQLPal, an AI tool that provides personalized feedback	23 students	15 weeks
[29]	Databases	Explore integration of AI tools in database education for instructors and students	PBL, Project-Based Learning, CL	ChatGPT, Copilot, Gemini	35 students	16 weeks

Table 4. *Cont.*

Reference	Topic	Research Objectives	Instructional Approach	AI Tool	Participants (No./Type)	Duration
[30]	Databases	Integrate LLMs into undergraduate database education and enhance learning outcomes while addressing students' unique needs	IBL, CL, Project-Based Learning	ChatGPT, Gemini, Claude	Students and instructors	n/m
[31]	Programming education and learner motivation	Explore the experiences of programming educators and students regarding the impact of generative AI tools, particularly large language models (LLMs), on learner motivation and educational practices	-	ChatGPT, Copilot	44 participants (educators and students) from various fields related to programming education	About one year (2023)
[32]	Generative AI in Computing Education	Evaluate the performance of code generation models on introductory programming problems, investigate the quality and novelty of learning resources they generate, and assess their impact on teaching practices in computing education	PBL, CL, EL	Codex	Students and educators	n/m
[33]	Programming	Evaluate ChatGPT-3.5's ability to solve programming tasks generated and assessed by Kattis in introductory programming education	-	ChatGPT	n/m	1 month
[34]	Programming	Explore the use of generative AI to automate the creation of programming exercises for educational purposes	PBL, IBL	OpenAI GPT API	n/m	n/m

Table 4. *Cont.*

Reference	Topic	Research Objectives	Instructional Approach	AI Tool	Participants (No./Type)	Duration
[35]	AI-assisted Programming	Investigate the effects of AI-assisted pair programming on undergraduate students in comparison to human–human pair programming and individual programming approaches	CL	GPT-3.5 Turbo, Claude 3 Opus	234 undergraduate students in a Java web application development course	2 academic years (2023-2024)
[36]	AI-assisted Programming	Explore AI's role in enhancing computer science education, personalized learning, competitive programming, ethical concerns, and ongoing innovation	CL, Flipped Classroom	GitHub Copilot, GPT-3, GPT-4	CS students and educators	n/m
[37]	AI-assisted Programming	Investigate the programmer behavior change and psychological effects of AI-assisted programming on professionals and undergraduates	CL, EL	ChatGPT 4.0, GitHub Copilot	38 undergraduate students (software engineering classrooms) and 5 software engineering professionals	n/m

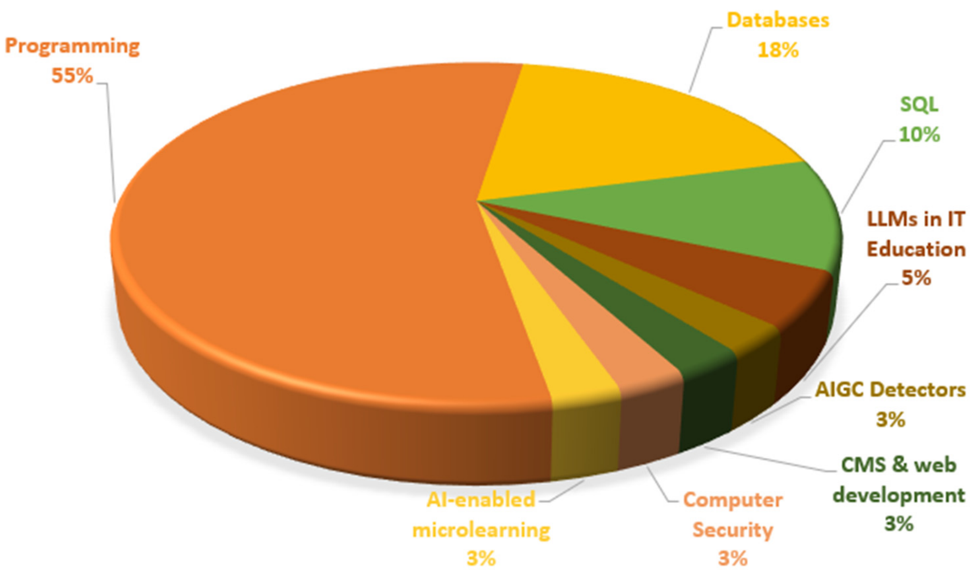

Figure 3. Learning content distribution.

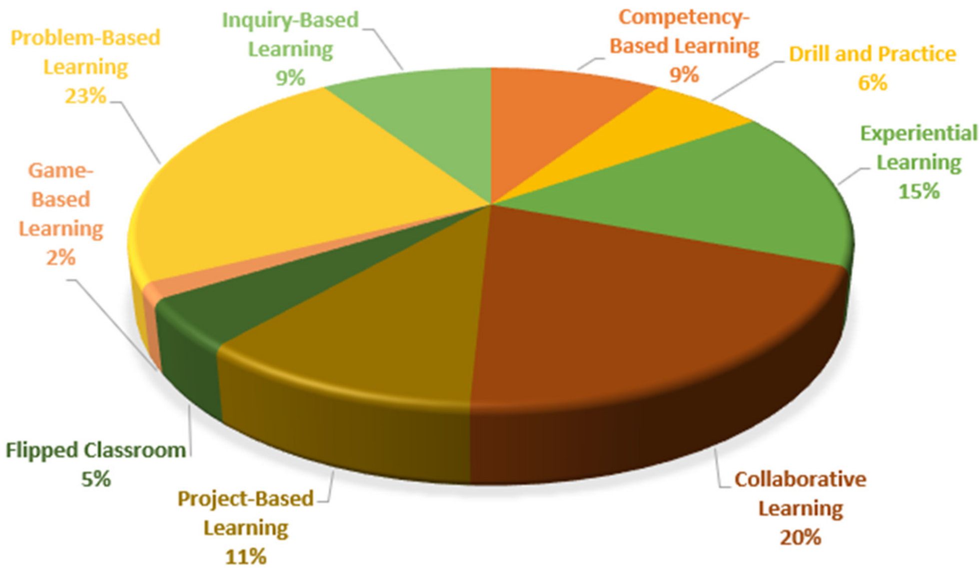

Figure 4. Learning approaches distribution.

3.3. Roles and Actions of Artificial Intelligence

The studies included in this systematic review identify several distinct roles that artificial intelligence plays in computer science education, particularly in instructional support and student assessment. As illustrated in Figure 5, AI is primarily used for guidance, code writing, question generation, progress monitoring, enhancing and accelerating learning, facilitating personalized learning, and performance/quality assessment. These roles highlight AI's growing influence in adaptive learning environments, where it provides real-time support, automated feedback, and tailored learning experiences.

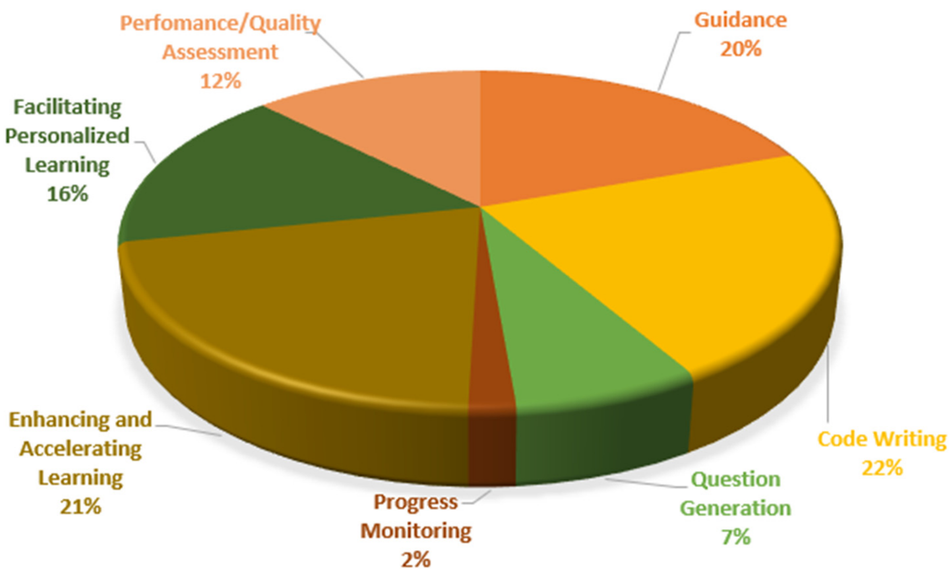

Figure 5. Roles of AI distribution.

Among the identified roles, code writing is the most frequently applied, with 25 studies [10–13,15–22,24–27,29–37], where AI tools automatically generate ready-to-use code to assist students in solving coding problems efficiently. Additionally, 24 studies [7–12,14–16,18–23,25,27,29–32,34,36,37] explored enhancing and accelerating learning. AI achieves this by responding directly to student queries, clarifying concepts, and streamlining the learning process.

Another significant role is guidance, used in 22 studies [7,9–11,13,14,16–18,20–23,25–31,35,36], utilizing AI to provide real-time instructional support as students complete assignments. This role enhances student engagement by offering on-demand assistance and explanations, particularly in programming tasks. Additionally, 18 studies [7,8,10,13,14,18–25,27,28,30,31,34] identified AI as a tool for facilitating personalized learning, where AI adapts responses and recommendations to align with individual student needs and progress levels.

AI also plays a role in educational assessment and evaluation. The role of performance/quality assessment was applied in 14 studies [8,12,15,17,19,23,24,26–30,32,33], where AI was used to evaluate the quality and effectiveness of student solutions. Additionally, eight studies [8,11,18–20,27,29,30] utilized AI for question generation, where AI automatically formulates evaluation questions based on student interactions and learning patterns. However, progress monitoring was the least explored AI function, with only two studies [14,30] employing AI to track student activity and generate statistical insights on learning progress.

To provide a structured overview of these findings, Table 5 presents a detailed breakdown of how AI is utilized across the selected studies, aggregating the AI actions of each role. This analysis highlights AI's most impactful contributions to computer science education and identifies underutilized areas that require further exploration.

Table 5. AI roles and actions in computer science education.

AI Role	Actions	Reference
Guidance	Help and guidance on programming tasks, creates explanations	[7,9,10,13,20,22,25–30]
	Help, guidance, and advice on solving problems with explanations	[11,16–18,21,26–30,35,36]
	Solutions and examples to difficult programming tasks	[13,20,22,29,30]
	Real-time answers, help, advice, and explanations	[14,21,26–31,35]
	Guides and advises students on the correct formulation of SQL questions	[23,26–30]
	Provides comparison and iterative feedback on different solutions to programming problems	[26–30]
Code Writing	Creates ready-made code	[11–13,15–22,24–27,29–37]
	Modifying incomplete or incorrect code	[10,11,13,15,17–19,21,26,27,29–32]
	Provides examples of best practices in SQL coding	[29,30]
Question Generation	Creates exercises and quizzes for better understanding	[11,18–20,27,29,30]
	Creates multiple-choice questions	[8,27,29,30]
	Generates true/false questions to assess comprehension	[29,30]
Progress Monitoring	Monitors student progress and generates statistics	[14,30]
Enhancing and Accelerating Learning	Immediate feedback and help	[7,9,11,15,19,21–23,25,27,29–32,34–36]
	Better understanding of concepts	[7,9,11,12,27,29,30,32,34,35,37]
	Efficient learning and teaching	[8,11,20,27,29,30,32,34–37]
	Improved performance and creativity	[10,16,29–32,37]
	Real-time gamified experience	[14]

Table 5. *Cont.*

AI Role	Actions	Reference
Facilitating Personalized Learning	Personalized answers to questions and support	[7,10,14,18–20,23,25,27,28,30,31,34]
	Customized solutions to the needs of each student	[10,13,19,22–25,27,28,30,34]
	Personalized questions to achieve specific learning outcomes	[8,21,30,31]
Performance/Quality Assessment	Evaluate the performance and quality of solutions accurately and efficiently	[15,17,19,24,27–30,32,33]
	Evaluates the quality, clarity, validity and alignment	[8,12,26,27,29,32]
	Evaluates and provides feedback on the correctness of SQL queries	[23,26–30]
	Comparative analysis of query performance	[26]

The most frequently utilized AI action is help and guidance on code writing, reinforcing AI's role in providing real-time explanations, debugging support, correct formulation of SQL questions, and problem-solving assistance. Similarly, generating ready-made code is a dominant AI action in code writing, enabling students to efficiently develop and refine coding solutions. AI also plays a significant role in personalized learning, where it adapts responses and support to individual students' needs.

For the enhancing and accelerating learning role, the most cited action is providing immediate feedback and real-time assistance, helping students correct mistakes and refine their understanding on demand. AI-driven performance/quality assessment, while less frequently, it is primarily focused on evaluating the correctness and effectiveness of students' solutions, ensuring that responses are accurate and aligned with learning objectives. It also assists with solution quality assessment and SQL query feedback.

Progress monitoring, the least explored function, is limited to two studies [14,30], indicating that AI's potential in tracking student learning progress, generating analytics, and providing data-driven insights is largely untapped. Automated question generation remains an emerging application, with AI used to create quizzes and multiple-choice questions for better assessment alignment.

3.4. Benefits, Challenges, and Limitations

Through this review, it was discovered that the use of artificial intelligence in the instruction of computer science and databases has several benefits in improving the learning experience. However, when utilizing it, a number of challenges may need to be addressed, whether successfully or not. Also, apart from the benefits, which are many, there are several limitations to the use of AI in the empirical studies analyzed.

3.4.1. Benefits

The findings reveal that AI provides numerous benefits in computer science education, improving learning efficiency, engagement, and instructional support. Table 6 presents a detailed overview of these advantages, ranking them based on their frequency across the selected studies.

The most commonly cited benefit is student support and guidance, where AI provides real-time assistance, problem-solving strategies, and explanations that help students navigate complex programming challenges [10,16,17,19,23,26–30]. AI also enhances efficiency and productivity, reducing manual effort for both students and instructors through automated grading, assessment generation, and content recommendations, a benefit that is particularly pronounced in IT education and SQL learning [8,9,12,20,22,26,27,29,30].

Table 6. Benefits of AI utilization in computer science instruction.

Benefits	Reference
Student support and guidance	[10,16,17,19,23,26–30]
Enhanced learning experience	[7,12,16,25–30]
Efficiency and productivity	[8,9,12,20,22,26,27,29,30]
Code writing and debugging	[11,13,15,20,22,24,26,27]
Real-time feedback	[13,17,22,23,26–28,30]
Personalized learning	[12,19,23,25–28,30]
Teacher assistance	[8,9,12,19,25,29]
Skill development	[14,21,23,27,30]
Creativity and critical thinking	[9,22,24,29,30]

Furthermore, AI contributes to enhanced learning experiences, making education more interactive, engaging, and accessible, especially in database education and in areas like AI-enabled microlearning [7,12,16,25–30]. Code writing and debugging is another major advantage, where AI tools assist students in automating code generation, identifying errors, and refining syntax, making coding tasks more efficient, particularly in fields such as programming and web development [11,13,15,20,22,24,26,27].

Another key benefit is real-time feedback, where AI-powered systems provide instant responses to student queries, enabling faster learning and quicker error correction [13,17,22,23,26–28,30]. Additionally, AI supports personalized learning by adapting instruction to individual learning styles, progress levels, and needs, a benefit that has been particularly noted in database and SQL courses [12,19,23,25–28,30]. In similar vein, teacher assistance is a closely related benefit, as AI tools help educators streamline lesson planning, track student progress, and provide personalized feedback, thereby easing their workload [8,9,12,19,25,29].

Another emerging benefit of AI in CS education is skill development, particularly in areas such as computational thinking, SQL learning, and problem-solving [14,21,23,27,30]. Additionally, some studies highlight AI's potential to support students' creativity and critical thinking, as students use AI tools for problem synthesis, solution evaluation, and project planning [9,22,24,29,30].

3.4.2. Challenges

The integration of generative AI tools, such as ChatGPT, presents challenges related to content accuracy, problem complexity, and evaluation reliability. AI systems often struggle with diverse topics, especially with erroneous data inputs or complex queries [22,25,28]. In coding tasks, ChatGPT frequently generates incorrectly formatted outputs, causing syntactical and logical errors in code [9,24]. Despite advancements, it occasionally fails to evaluate programming solutions correctly, causing misinterpretations of expected outputs and flawed debugging assistance [17,24].

A significant concern is the inability of AI content detectors (AIGC) to distinguish between human-written and AI-generated code due to varying syntax structures in programming languages. Tools fail to detect AI submissions, raising concerns about academic integrity [12]. This highlights the risk of academic dishonestly, as students may rely on AI to generate SQL code, undermining the authenticity of their work and the integrity of assessments [26]. Additionally, concerns about biases and overreliance on AI hinder the development of critical thinking and problem-solving skills in students [18,19,22]. AI tools face technical constraints, including high computational requirements, slow response times, and integration challenges, which limit their scalability in education [11,13].

In the context of database education, AI tools encounter notable limitations, particularly in terms of providing contextually appropriate explanations. This deficiency can lead to misconceptions and inaccurate learning outcomes [14,23]. For instance, ChatGPT faces challenges in interpreting SQL queries and determining if different queries yield identical results, largely due to the absence of a clear theoretical foundation for SQL equivalence [26]. This can lead to errors in grading and misinterpretations of logical relationships between queries [26].

Additionally, an excessive reliance on AI tools in database education has the potential to hinder students' development of critical thinking and problem-solving skills, which are essential for comprehending intricate database structures and SQL logic [29]. It is crucial to strike a balance between AI utilization and conventional learning methodologies to foster independent thinking and problem-solving skills [29].

However, biases in the training data, particularly those related to newly proposed solutions and curricula, may lead to LLMs reinforcing outdated or biased content, potentially impacting student learning [30]. Data privacy and security are also significant issues, with concerns over the misuse of both training data and student input [30].

Addressing these concerns is crucial for the successful integration of AI-driven systems in education. The ongoing refinement of AI-driven tutoring systems, particularly in areas beyond programming assistance, is essential to ensure the effective and ethical use of these technologies in educational settings.

3.4.3. Limitations

Despite its benefits, ChatGPT and AI tools face several limitations in educational and programming contexts. Issues with validity and bias limit their generalizability, particularly in complex coding tasks like Java and SQL [7,14,23,25]. Usability challenges also arise for learners with minimal technical skills, restricting AI's accessibility [25].

AI-generated assessments, including multiple-choice questions, often lack completeness and alignment with learning objectives, raising concerns about rater bias and reliability [8]. Additionally, reliance on preexisting datasets affects assessment originality and academic integrity [24]. Research findings are further constrained by small sample sizes and short study durations, limiting insights into long-term AI impact [9,10,21]. Studies often face methodological challenges, such as low participation, which may skew results, as students predisposed to AI technology are more likely to engage. Furthermore, student familiarity with course content influences the quality of chatbot queries, affecting AI-generated responses. Manual feedback evaluation by a single teaching assistant can also introduce accuracy concerns. Another limitation is the exclusive use of GPT-4 in some studies, without comparison to alternative LLMs [27].

Ethical concerns include plagiarism risks, student over-reliance, and AI's role in academic dishonesty, particularly among first-year learners [16,22]. Computational demands, adaptation challenges, and inconsistent AI-generated responses affect reliability and scalability [18–20]. Moreover, AI's limited application in database learning and computational thinking remains a research gap [23].

To enhance AI's effectiveness in computer science education, improvements are needed in personalization, adaptability, and ethical safeguards [14,23]. Addressing these challenges and limitations is crucial for sustainable and responsible AI adoption in education.

Is it inevitable that these limitations, such as difficulties in recognizing correct answers, understanding context, and processing complex queries, contribute to reduced trust in AI tools. These issues intensify concerns around the reliability of AI in supporting educational and programming tasks, especially in database design courses. The frequent errors and inconsistencies in responses limit the overall effectiveness of AI tools, making it challenging for them to meet the demands of both learners and educators in complex learning environments [28].

3.5. The Use of AI-Assisted Programming in Educational Approaches

The integration of AI in programming education is leading to significant changes in teaching methodologies and learning experiences. Large language models and AI-driven code generation tools have been shown to provide personalized support, automate routine tasks, and generate educational resources, thereby fundamentally altering the landscape of programming education [10–13,15–20,24–27,29–37].

AI-assisted tools have been shown to enhance learner autonomy by offering tailored learning experiences, reducing negative emotions such as frustration and insecurity, and fostering engagement. Educators can leverage these tools to provide personalized feedback and resources, allowing them to focus on social learning opportunities that further motivate students [31]. The capacity of AI to automatically generate programming exercises, explanations, and feedback contributes to a reduction in instructors' workload while expanding the variety of available learning materials [32,34].

Moreover, AI tools improve the clarity and actionability of programming error messages, facilitate multiple solution approaches, and enable students to concentrate on higher-level problem-solving by automating lower-level coding tasks. These features enable students to engage with specification-focused learning, emphasizing problem definition over manual coding [32]. Additionally, studies have demonstrated that AI-assisted programming can enhance intrinsic motivation, reduce programming anxiety, and improve performance, particularly in collaborative settings such as AI-assisted pair programming [35].

Recent empirical studies provide further evidence of AI's impact on programming education. For instance, a semester-long study revealed that students utilizing CodeTutor, an LLM-powered assistant, demonstrated notable enhancements in their final scores compared to a control group [10]. The study also found that students with no prior experience with AI tools benefited the most, suggesting that AI-assisted programming can help bridge gaps in foundational programming skills [10]. Similarly, another research study revealed that while educators recognize AI's potential in enhancing instruction and automating grading, concerns remain about student reliance on AI tools and academic integrity [20].

The use of AI chatbots, particularly ChatGPT, has been explored as a means to support programming education in various ways. Studies have shown that AI tools can generate code snippets, debug errors, and provide structured explanations, which have been beneficial for both novice and experienced programmers [11,13]. Research indicates that AI-generated learning materials can enhance comprehension by offering adaptive learning tasks and real-time feedback [17,19].

Despite these advantages, challenges remain regarding the reliability of AI-generated content and the potential for misuse. A primary concern is that students may become overly dependent on AI-generated solutions, potentially undermining their ability to develop critical problem-solving skills. The quality and reliability of AI-generated content vary, particularly in complex problem scenarios, raising concerns about the accuracy of solutions provided by LLMs [15,16]. Additionally, the current state of AI-generated content detectors often fails to effectively differentiate between human-written code and AI-generated code, underscoring existing limitations in enforcing academic integrity [12]. Studies also show that while tools like ChatGPT or Codex can produce syntactically correct code, their outputs may contain conceptual errors or security vulnerabilities that students may fail to identify without proper training [24]. Furthermore, the variability and occasional inconsistency in AI responses require developers to be equipped with the skills to assess, validate, and refine AI-generated code. Therefore, it is essential that educational programs include structured guidance and explicit instruction on how to interact with AI tools critically and responsibly. This training should cover both technical usage and also promote reflective practices, ethical awareness, and an understanding of AI limitations—ensuring that students are not merely consumers of AI assistance, but also informed and capable collaborators.

Research is ongoing to evaluate the accuracy of AI-generated solutions and explore best practices for integrating these tools into educational settings while maintaining academic integrity [33]. At the same time, concerns have also been raised regarding the potential for over-reliance on AI, which could result in a superficial understanding of fundamental programming concepts. A balanced approach integrating AI tools while ensuring fundamental knowledge acquisition is recommended [37].

The application of AI-driven programming tools goes beyond traditional educational frameworks, promoting self-regulated learning and competitive programming skill development. These tools provide immediate support through natural language interfaces, automated error diagnosis, and dynamic difficulty modulation tailored to individual learner progression. The integration of AI into computer science education provides students with hands-on experience with intelligent programming assistance, aligning with current industry standards [36].

Despite the challenges mentioned above, AI-driven tools continue to enhance programming education by strengthening self-regulated learning and aligning with industry standards. The application of AI-driven programming tools transcends conventional educational frameworks, promoting self-regulated learning and the cultivation of competitive programming skills. These tools offer immediate assistance through natural language interfaces, automated error diagnosis, and dynamic difficulty modulation tailored to individual learner progression. The integration of AI into computer science education furnishes students with practical experience in intelligent programming assistance, aligning with current industry standards [36]. For instance, AI tools have been successfully integrated into database courses, enhancing student engagement with SQL query generation and database schema design [26,29]. Furthermore, LLM-based virtual tutors have exhibited promise in delivering customized learning experiences through real-time problem-solving and academic guidance [30].

The continuous improvement of AI models suggests a growing role for AI in programming development and education, leading to the design and application of innovative teaching methods and the transformation of traditional learning models. When integrated into software engineering curricula, AI has the potential to optimize learning outcomes while minimizing potential adverse effects [37]. However, a balanced approach is essential, leveraging AI's strengths while ensuring students develop a strong foundation in programming fundamentals.

3.6. The Use of Artificial Intelligence in Databases Instruction

The analysis of the selected studies reveals that seven out of the twenty-four studies specifically investigate the use of generative AI in database education [14,23,26–30]. The emergence of AI, especially large language models (LLMs) such as ChatGPT, is changing teaching and learning paradigms, especially in online education [26–28,30]. AI tools are being integrated into database education to enhance the learning experience and provide personalized support. These tools offer benefits such as personalized learning, increased student engagement, and reduced administrative burden for educators [23,29,30]. For example, AI-powered chatbots, such as MoodleBot [27] and ChatbotSQL [23], can assist students by providing 24/7 support, answering questions, and generating exercises. ChatGPT can help generate SQL queries, assess student work, and provide feedback on student solutions. AI can also support self-paced learning by providing immediate feedback and explanations [14,23,26,27]. In addition, instructors can use AI to create course materials and design assignments that promote deeper understanding and problem-solving skills [29].

However, the integration of AI in database instruction also poses challenges. Concerns include the accuracy and consistency of AI responses [28], the potential for bias and hallucinations [27], and the risk of over-reliance on AI [30], which may hinder critical thinking [29]. There are also challenges regarding data privacy, security, and the potential for cheating and misuse of AI tools [26,30]. The effectiveness of AI responses can be sensitive to prompting, requiring careful design of questions and prompts. Moreover, students have reported issues with the accuracy of AI feedback, with correct answers being marked wrong and AI struggling with complex queries [28].

To address these challenges, ongoing research and refinement of AI tools are necessary. This includes improving the accuracy and contextual understanding of AI, implementing automated fact-checking mechanisms, and incorporating interactive features and human support to enhance effectiveness [27,28]. Educators are encouraged to design assignments that balance AI assistance with the development of critical thinking and problem-solving skills, and to train students in prompt engineering and ethical considerations in AI use [29]. Future research could explore broader applications of AI in various educational contexts and focus on creating more interactive and personalized learning experiences, such as through virtual tutor systems [30]. These systems should be designed to incorporate features such as implicit query execution, visual step-throughs, data personalization, pop quizzes, and learning outcomes reports to provide comprehensive support and enhance mastery of database concepts [14,30].

4. Discussion

This systematic review highlights the growing role of AI in computer science education, emphasizing its prevalence in programming instruction and its relatively limited application in database learning. The findings indicate that AI is primarily used to enhance learning efficiency, provide personalized learning and automate feedback, and support student engagement, aligning with previous studies [38–41]. However, while AI has been extensively studied for code writing, debugging, and personalized learning, its application in database instruction remains significantly unexplored, suggesting an important gap in the field, with only seven studies addressing in database education, SQL learning, and query optimization.

To further contextualize these findings, the discussion is structured around the key research questions that guided this review.

4.1. Research Methodologies

The reviewed studies employed a variety of research methodologies, with a strong emphasis on quantitative and mixed methods approaches. The majority of studies utilized quantitative methods, particularly surveys, experimental designs, and statistical analysis, to assess the effectiveness of AI tools in computer science education [3,4]. The dominance of quantitative methods aligns with broader trends in AI and education research, where empirical validation is prioritized to measure AI's impact on learning outcomes [42].

Mixed methods research, combining both quantitative and qualitative approaches, was also widely used, offering a more comprehensive understanding of AI's impact by integrating numerical data with qualitative insights [4,5]. This methodological approach enables a multidimensional evaluation, capturing both measurable learning outcomes and student experiences.

Notably, the majority of studies relied on quantitative methods (experiments, surveys) to measure AI's impact on student learning, while mixed methods research provided a broader evaluation of both outcomes and student experiences. However, no studies exclusively employed qualitative methods, limiting insights into student perceptions, cognitive engagement, and classroom interactions. Prior literature suggests that qualitative approaches, such as ethnographic studies or longitudinal interviews, could provide deeper understanding of AI's influence on learning behaviors and pedagogical strategies [43]. Future studies should explore how students and educators experience AI-driven instruction beyond performance metrics.

4.2. Instructional Design for AI Integration in Computer Science Education

Research findings indicate that the integration of AI into CS education has been achieved through a variety of pedagogical approaches. AI applications span across multiple areas of computer science, with a predominant focus being on programming instruction. Research highlights AI's role in teaching specific programming languages (e.g., Java, Python) and general programming concepts [3,6]. Given the fundamental role of programming in computer science, AI tools play a crucial role in enhancing instruction by providing real-time coding assistance, generating explanations for complex concepts, and offering multiple solution approaches. These capabilities align with existing literature emphasizing the effectiveness of automated tutoring and real-time debugging in coding instruction [44].

Beyond programming, the integration of AI extends to database instruction, where studies [4,23,26–30] have explored the use of AI-driven tools to support both students and instructors and enhance learning and teaching. Chatbot-based platforms have been utilized to facilitate SQL learning and assist students in practicing complex database queries independently [14,23]. Additionally, large language models (LLMs) such as ChatGPT have been evaluated for their effectiveness in solving SQL assignments, their potential implications for academic integrity in online assessments, and their use as grading tools [26]. Research also emphasizes the role of LLM-driven chatbots in higher education database courses, highlighting their potential to enhance engagement and comprehension [27]. Furthermore, investigations have been conducted into AI-driven microlearning methodologies, which have been shown to offer benefits such as customized instruction and the segmentation of learning material into manageable units [28–30]. Given the conceptual complexity of databases, expanding research into AI-driven tutoring systems could enhance students' understanding of complex query structures, normalization processes, and database design principles.

Several instruction models have been identified in AI-supported computer science education. Problem-based learning is the most frequently applied approach [3], encouraging students to engage in real-world problem-solving. This approach fosters critical thinking and collaboration as students debug code, optimize algorithms, and explore alternative solutions. Similarly, collaborative learning leverages AI as a support tool for discussions, troubleshooting, and knowledge sharing, enhancing both technical and social learning experiences [1,38]. Another widely implemented approach is experiential learning, which emphasizes hands-on engagement through interactive AI-based platforms that provide real-time feedback and adaptive learning pathways.

Additional instructional models include inquiry-based learning, where students pose and investigate their own questions, and competency-based learning, which focuses on skill mastery through AI-driven personalized assessments. Additionally, drill and practice, project-based learning, flipped classroom, and game-based learning have been incorporated to enhance engagement and provide diverse learning experience [4]. AI-powered platforms such as Codex, CodeTutor, and ChatbotSQL have demonstrated effectiveness in supporting autonomous learning.

However, the review also indicates that most AI-driven studies focus on higher education settings, with limited research on AI's use in K-12 computer science instruction [5]. This may be due to the fact that computer science courses are mostly taught in higher education, although there are many attempts to teach some courses, such as programming or AI, at earlier educational levels (e.g., elementary or K-12) [5]. The use of AI in earlier educational stages, particularly for teaching fundamental programming and computational thinking concepts, is growing [4]. Expanding research in this area could help design more accessible and age-appropriate AI-driven instructional models for a broader range of learners.

4.3. Roles and Actions of Generative AI in the Instruction of Computer Science?

The results showed that there are different roles for generative AI when used in computer science education. The most frequently used role encountered was that of enhancing and accelerating learning. Through direct support and feedback in an interactive and creative environment, AI achieves faster learning and deeper understanding. Also, due to the fact that AI is applied based on a deep understanding of computer science pedagogy, it contributes better to the learning performance of computer science education compared to other educational subjects [5].

In addition, artificial intelligence used in computer science education can "guide" users and "write code". Students enjoy real-time assistance for any questions and explanations of concepts for better understanding. In addition, generative AI tools are able to provide answers even with an entire code, code fragments, and even modify an incorrect code, often pointing out many times where the errors are [1]. Debugging code can take many hours, but if there is a help from a chatbot, for example, then learning is accelerated at the same time. There are also tools that have the potential to offer students "personalized learning" according to their level and needs. In addition, there are tools that have the potential to offer students "personalized learning" according to their level and requirements [3]. Personalized responses can lead to better results in understanding concepts, and consequently, there is more freedom to express questions and answers always according to the needs of each student, while achieving a deeper understanding. Furthermore, some studies have shown that AI can "create questions" in order to check the level of understanding, and also "evaluate the performance/quality" for the solutions given [4]. Teachers can be allowed the possibility to control the actions and activity of their students through the role of "progress monitoring" for a better situational awareness [6].

We conclude that the roles of generative AI are diverse and critical to supporting students and teachers in computer science education. However, these roles should be integrated into teaching with consistency, moderation, and balance as an additional means of learning [5]. It is not possible for AI tools to replace the vital importance of the teacher in the educational and learning process.

While AI effectively facilitates immediate learning tasks, its ability to develop students' conceptual understanding and problem-solving skills remains unclear. Another critical finding is the limited implementation of AI in progress monitoring, with only one study employing AI to track student performance over time [14]. This is significant because adaptive learning models, which adjust instruction based on student progress, are a key feature of AI-enhanced education. Most AI applications in the reviewed studies focus on short-term, task-based assistance rather than long-term skill development. Future research should explore AI's potential for tracking learning trajectories, providing adaptive interventions, and supporting competency-based education.

4.4. Benefits, Challenges, and Limitations of Generative AI in Computer Science

The integration of generative AI in computer science education offers multiple benefits for both students and educators. The learning experience is enhanced and made more engaging through an interactive mode of learning [1,3]. Students benefit from hands-on, experiential learning, which increases motivation and fosters a deeper understanding of complex concepts. Additionally, AI provides varied explanations and multiple solution pathways, catering to different learning styles and improving overall comprehension. Beyond student engagement, AI also supports educators by facilitating professional development and instructional efficiency. Teachers can leverage AI-driven analytics to track student progress, gaining insights into individual learning patterns and adapting their teaching strategies accordingly [6]. This allows teachers to obtain a better picture of each student's level and abilities.

Artificial intelligence can, therefore, boost pedagogical efficiency, and it is useful to integrate it into the computer science instruction. However, it should be noted that AI in education should be seen as a complementary tool rather than a replacement for instructors [45].

Like any emerging technology, generative AI comes with inherent challenges and limitations. These include topic coverage constraints, biases in AI-generated content, inaccuracies in responses, and difficulty in solving highly complex problems [1,4]. Given that AI tools are typically designed for specific educational tasks, they may struggle with contextual understanding or misinterpret ambiguous queries. Furthermore, excessive reliance on AI can diminish students' critical thinking and problem-solving abilities, leading to cognitive dependency rather than fostering independent learning [5], which would be very unpleasant. Additionally the review showed that AI may produce inaccurate or misleading responses, which is in accordance with the literature [46].

While future AI advancements may address some of these limitations, its responsible and moderated use is crucial to prevent overdependence and misuse [3]. AI's integration into computer science education should be carefully managed, ensuring that it supports learning without replacing essential human cognitive processes.

4.5. AI-Assisted Programming and Transformation of Educational Approaches

The integration of AI-assisted programming, particularly through LLMs and generative AI tools, is transforming programming education [10–13,15–22,24–27,29–37]. These technologies facilitate personalized learning experiences by adapting to diverse student needs, offering support across different programming languages, learning styles, and proficiency levels [10,11,31,35]. By providing real-time feedback and customized explanations, AI tools enhance student engagement, promote self-efficacy, and provide a more interactive and autonomous learning environment [31,35].

AI-assisted programming has been shown to mitigate common learning barriers such as a lack of confidence, frustration, and programming anxiety [31,35]. The non-judgmental nature of AI support encourages experimentation and reduces fear of failure, allowing students to develop problem-solving skills in a supportive setting [35]. Additionally, AI has been shown to enhance learning by automating routine tasks such as debugging, syntax correction, and code explanation, thereby enabling students to focus on conceptual understanding and algorithmic thinking rather than on rote code implementation [32,37].

Furthermore, AI technologies are transforming the development of educational materials. The automated generation of programming exercises, detailed explanations, and personalized feedback reduces educators' workload while making curricula more adaptive and responsive [32,34]. These tools demonstrate multiple solutions to coding problems, reinforcing diverse problem-solving strategies and improving comprehension [32].

Additionally, AI-powered tutoring systems and chatbots facilitate dynamic assessment methods, enhancing educational outcomes and promoting greater accessibility to programming [25,27].

However, the integration of AI in programming education presents challenges that require careful consideration. A significant concern is the potential for overreliance on AI-generated solutions, which may hinder the development of critical thinking and problem-solving skills [10,16,24,30]. The effectiveness of AI tools is heavily dependent on the quality of the prompts and user interactions, emphasizing the need for educators to guide students in effectively leveraging these technologies [10]. Ethical considerations, such as academic integrity and the risk of plagiarism, also necessitate pedagogical strategies that prioritize comprehension, debugging, and algorithmic reasoning over mere code generation [20].

Is it evident that programming education is undergoing a shift towards higher-order skills such as code analysis, debugging strategies, and AI utilization [15,17,18]. This evolution underscores the need to strike a balance between leveraging AI assistance and fundamental learning, ensuring that students develop a robust understanding of programming principles [37]. By strategically integrating AI-assisted programming into curricula, educators can maximize its benefits while maintaining essential skill development and a critical approach to AI in software engineering [29,30].

AI-assisted programming has been identified as a promising avenue for enhancing education through personalized support, automated learning resources, and interactive engagement [18–22]. However, its implementation must be meticulously managed to preserve essential programming competencies and encourage independent problem-solving [10–13,15–17]. A thoughtful integration of AI in programming education has the potential to empower students and prepare them for the evolving technological landscape [29,30].

4.6. AI in Database Instruction

The integration of AI in database instruction is still in its early stages [14,23,26,28–30], despite its widespread adoption in programming education [4,14,23,27–30]. AI-powered tools, such as digital assistants, have demonstrated considerable potential in providing personalized support [3,14,23,27,28,30], interactive learning [1,14,23,27–30], and progress monitoring [5,14,30], benefiting both students and educators. However, challenges such as student dependency on AI tools and concerns regarding academic integrity must be carefully addressed [1,14,23,26–30].

Database education encompasses complex subjects, including data modeling, normalization, and query optimization, which extend beyond basic syntax correction [14,23,29]. While AI has been employed for SQL learning through chatbot-based platforms and automated assessment tools, a gap remains in AI-driven solutions designed to visualize database relationships, detect logical errors, and provide adaptive feedback. This underscores a significant research opportunity [14,23,27–30]. Expanding AI's role beyond syntax assistance to include concept-based learning and intelligent tutoring systems has the potential to enhance students' comprehension of abstract database principles.

Recent advancements, particularly in LLMs such as ChatGPT, have introduced novel opportunities for integrating AI into DB instruction. These models assist students by generating queries from natural language inputs, improving their understanding of complex database concepts [14,23,26–30]. These models also facilitate flexible grading and independent learning, particularly in online settings. However, pilot studies on AI-enabled microlearning have revealed mixed student feedback [28]. While students have expressed appreciation for personalization and real-time interaction [28,30], concerns regarding inconsistent responses and inaccuracies have also been raised [23,26,27,29,30]. These findings underscore the need for the ongoing refinement of AI tools [23] to ensure their optimal performance and effectiveness in supporting learning. Integrating AI tools into database courses could enhance teaching efficiency by automating tasks such as quiz creation and grading, allowing instructors to focus more on personalized student engagement [23,26,28–30]. Additionally, students have reported that AI-integrated assignments have helped them to develop problem-solving and analytical skills, particularly in database design and programming [14,23,27–30].

According to the results of the systematic review, the primary applications of AI in database education are through three key approaches. One approach involves the use of AI-powered chatbots specifically designed for database instruction. These chatbots enable students to inquire about database concepts and SQL queries, providing immediate feedback and explanations. This form of support helps students navigate complex topics such as relational database structures, normalization, and query optimization, facilitating a more interactive and accessible learning experience.

Another significant approach is the use of intelligent self-assessment tools, which provide personalized feedback based on each student's learning progress and individual characteristics. These tools analyze students' database-related responses and offer personalized recommendations, adapting to their unique learning needs.

A further method of AI integration involves its incorporation into learning management systems, such as Moodle, where AI-driven tools assist students through interactive question-and-answer sessions. These systems provide real-time hints, explanations, and recommendations, enabling students to engage with the content in a more structured and supportive environment.

In addition to the utilization of structured AI-driven platforms, a more flexible approach has emerged. In this approach, students employ general-purpose AI-chatbots, such as ChatGPT, to verify their SQL solutions, identify errors, and receive immediate feedback. This method requires minimal instructor intervention while still promoting self-directed learning and analytical reasoning. However, it is imperative to ensure that students critically assess the AI-generated responses they receive, rather than passively accepting them.

While the present systematic review emphasizes the impact of AI on educational programming and database instruction, it is important to recognize that generative AI is also reshaping the nature of programming as a professional discipline. The advent of AI-assisted and AI-native software development is effecting a paradigm shift in industry practices and necessitating the evolution of skill sets. Consequently, there is an imperative for computer science education to evolve to address these fundamental shifts by incorporating competencies such as prompt engineering, critical assessment of AI-generated code, and collaboration with intelligent systems. While the present review focused on pedagogical contexts, acknowledging these broader disciplinary transformations underscores a crucial area for future educational innovation and research.

In conclusion, AI has the potential to transform database education by providing more interactive and personalized learning experiences [23]. Nevertheless, further research is imperative to refine AI tools and address challenges such as over-reliance and the necessity for critical thinking skills [14,23,26,28–30]. As AI continues to evolve, it has the potential to play a pivotal role in shaping the future of database instruction, better preparing students for an AI-driven economy while enhancing their mastery of database concepts.

4.7. Implication of Practice

The findings of this systematic review have several implications for the practice of integrating AI in computer science education and in the context of database instruction.

More comprehensive research is needed on the use of AI in computer science education and assessing its long-term impact. While the existing studies suggest promising results, there is a lack of large-scale, longitudinal studies that can provide a more holistic understanding of the impact of AI on student learning outcomes, engagement, and the overall educational experience.

The integration of AI in computer science education should be guided by sound pedagogical principles and educational theories. Specifically, the design and implementation of AI-powered tools and platforms should be informed by learning theories, such as constructivism, experiential learning, and collaborative learning, to ensure that they effectively support student-centered and active learning approaches.

Ethical considerations must be addressed. Institutions must also establish ethical guidelines to address student dependency, academic integrity, and AI biases.

Educators must be equipped to evaluate AI tools critically and integrate them effectively into teaching. Without proper training, AI's potential to personalize learning and improve instruction may not be fully realized. Equally important, future programmers also require structured training on the effective and ethical use of AI tools such as chatbots and code generators. This includes understanding the capabilities and limitations of generative AI, recognizing errors or biases in outputs, and maintaining critical engagement with AI-generated content.

5. Conclusions

The findings of this systematic review indicate the important role that artificial intelligence plays in the learning process of computer science and database instruction. The review has highlighted the growing body of research and innovation in this field, with AI-powered tools and platforms demonstrating the potential to enhance student learning, engagement, and outcomes. It revealed the instructional approaches for the utilization of generative AI, the roles and actions of AI in database and computer science instruction, and the benefits of AI, along with the limitations and challenges that must be addressed.

The review underscores the diverse array of educational designs through which AI can be integrated into computer science instruction. Although AI's use has been primarily focused on programming, its applicability extends to other disciplines, such as web development [11] and cyber security [16]. The predominance of AI research in higher education suggests that its development is more feasible at advanced academic levels, where students possess higher cognitive maturity and technical proficiency [4].

It is clear that the integration of AI in computer science instruction and programming as a profession can offer significant benefits for programmers, students, and educators alike. Through this, pedagogical effectiveness can be achieved as the learning experience is enhanced in an interactive way of learning and understanding becomes deeper [19,21]. Educators benefit from reduced time expenditure and the ability to obtain a comprehensive understanding of students' progress through monitoring their activity [14]. However, there are several challenges that must be addressed. The limited range of topics, biases, wrong answers leading to incorrect learning, lack of critical thinking, and inability to find solutions to complex problems are just some of them [20,22].

The systematic review showed that AI can bring about significant changes in education by improving the educational experience, but its use does not replace the importance of the human teacher in the process. Human interaction plays a pivotal role in facilitating effective learning.

This paper also touches on the vital industry of databases. Through the found studies related to databases [14,23,26–30], it was observed that artificial intelligence in database education and SQL can bring great benefits. A digital assistant that provides guidance will help improve and accelerate learning. Moreover, since databases is a subject with specific terminology that often makes it difficult for students to learn learner, personalized learning according to each student's needs, abilities and requirements can exponentially improve comprehension levels. The proper use of generative AI can, therefore, benefit both students and educators.

Limitations and Future Work

Despite the comprehensive nature of this systematic review, several limitations should be noted. Specifically, the scope of the review was limited to publications in journals and conference proceedings, after searching only two databases (Google Scholar, SCOPUS). In addition, the literature review focused on studies published in the last five years (2020–2025) and was limited to one language, English.

The limited number of studies on AI in database instruction prevents a comprehensive evaluation of its effectiveness in this domain, highlighting a gap that needs further exploration.

The scope of this study was mainly focused on the educational application of AI in computer science, focusing specifically on its role in teaching programming and database instruction, with one research question to examine the broader impact of AI on software engineering as a profession and the transformation of education. The rise of AI-assisted software development, along with the paradigm of AI-native software development, is reshaping the skills sets required in the industry; therefore, further research is essential to investigate these shifts and the changes bring in the educational process.

Another limitation of this study is the absence of a formal inter-rater reliability calculation, such as Cohen's kappa, to quantify agreement between reviewers during the study selection process. While discrepancies were resolved through discussion and consensus, the inclusion of statistical agreement measures in future systematic reviews would enhance methodological transparency and ensure greater consistency in the selection process.

To address these limitations, future research should expand the scope of the review to include research results from more diverse literature sources. This will cover a wider range of research being conducted on the topic. Furthermore, due to the limited study on the utilization of generative AI usage in the field of databases, it is imperative to conduct relevant research in the field of computer science. Further studies should explore AI-driven tools for database instruction, expanding AI's role beyond syntax correction to conceptual learning, query optimization, and relational modeling. Additionally, research on ethical concerns, academic integrity, and biases in AI-generated content should be prioritized to develop responsible AI frameworks that promote fair, transparent, and unbiased learning environments. Given the increasing influence of AI-driven practices—such as automated coding, debugging, testing, and AI-assisted software development—future research should explore how these developments are transforming professional practices and the necessary skill sets within software engineering education, along with their educational implications.

Author Contributions: Conceptualization, I.K. and A.G.; methodology, A.G.; validation, I.K. and A.G.; validation A.G.; formal analysis, A.G.; investigation, A.G.; resources, A.G.; writing—original draft preparation, A.G.; writing—review and editing, I.K.; supervision, I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
AIGC	AI-generated content
CBL	Competency-based learning
CL	Collaborative learning
CS	Computer science
DP	Drill and practice
EAI	Educational artificial intelligence
EL	Experiential learning
GPTs	Generative pre-trained transformers
IBL	Inquiry-based learning
IT	Information technology
ITS	Intelligent tutoring systems
LLMs	Large language models
LMS	Learning management system
ML	Machine learning
PBL	Problem-based learning
RDBMS	Relational database management systems
SQL	Structured query language

References

1. Kim, T.W. Application of Artificial Intelligence Chatbots, Including ChatGPT, in Education, Scholarly Work, Programming, and Content Generation and Its Prospects: A Narrative Review. *J. Educ. Eval. Health Prof.* **2023**, *20*, 38. [CrossRef] [PubMed]
2. Grassini, S. Shaping the Future of Education: Exploring the Potential and Consequences of AI and ChatGPT in Educational Settings. *Educ. Sci.* **2023**, *13*, 692. [CrossRef]
3. Joseph, O.B.; Uzondu, N.C. Integrating AI and Machine Learning in STEM Education: Challenges and Opportunities. *Comput. Sci. IT Res. J.* **2024**, *5*, 1732–1750. [CrossRef]
4. García-Martínez, I.; Fernández-Batanero, J.M.; Fernández-Cerero, J.; León, S.P. Analysing the Impact of Artificial Intelligence and Computational Sciences on Student Performance: Systematic Review and Meta-Analysis. *J. New Approaches Educ. Res.* **2023**, *12*, 171–197. [CrossRef]
5. Tlili, A. ¿Puede Ayudar La Inteligencia Artificial (IA) En La Educación En Ciencias de La Computación? Un Enfoque Metaanalítico. *Rev. Esp. Pedagog.* **2024**, *82*, 469–490. [CrossRef]
6. Francisco, R.; Silva, F. Intelligent Tutoring System for Computer Science Education and the Use of Artificial Intelligence: A Literature Review. In Proceedings of the 14th International Conference on Computer Supported Education; SCITEPRESS—Science and Technology Publications, Virtual, 22–24 April 2022; pp. 338–345.
7. Bassner, P.; Frankford, E.; Krusche, S. Iris: An AI-Driven Virtual Tutor for Computer Science Education. In Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1, Milan, Italy, 8–10 July 2024; pp. 394–400.
8. Doughty, J.; Wan, Z.; Bompelli, A.; Qayum, J.; Wang, T.; Zhang, J.; Zheng, Y.; Doyle, A.; Sridhar, P.; Agarwal, A.; et al. A Comparative Study of AI-Generated (GPT-4) and Human-Crafted MCQs in Programming Education. In Proceedings of the 26th Australasian Computing Education Conference, Sydney, Australia, 29 January–2February 2024; pp. 114–123.
9. Lee, S.; Song, K.-S. Teachers' and Students' Perceptions of AI-Generated Concept Explanations: Implications for Integrating Generative AI in Computer Science Education. *Comput. Educ. Artif. Intell.* **2024**, *7*, 100283. [CrossRef]
10. Lyu, W.; Wang, Y.; Chung, T.; Sun, Y.; Zhang, Y. Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. In Proceedings of the Eleventh ACM Conference on Learning @ Scale, Atlanta, GA USA, 18–20 July 2024; pp. 63–74.
11. Mekić, E.; Jovanović, M.; Kuk, K.; Prlinčević, B.; Savić, A. Enhancing Educational Efficiency: Generative AI Chatbots and DevOps in Education 4.0. *Comput. Appl. Eng. Educ.* **2024**, *32*, e22804.
12. Pan, W.H.; Chok, M.J.; Wong, J.L.S.; Shin, Y.X.; Poon, Y.S.; Yang, Z.; Chong, C.Y.; Lo, D.; Lim, M.K. Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training, Lisbon, Portugal, 14–20 April 2024.
13. Sun, D.; Boudouaia, A.; Zhu, C.; Li, Y. Would ChatGPT-Facilitated Programming Mode Impact College Students' Programming Behaviors, Performances, and Perceptions? An Empirical Study. *Int. J. Educ. Technol. High. Educ.* **2024**, *21*, 14. [CrossRef]

14. Balderas, A.; Baena-Pérez, R.; Person, T.; Mota, J.M.; Ruiz-Rube, I. Chatbot-Based Learning Platform for SQL Training. *Int. J. Interact. Multimed. Artif. Intell.* **2024**, *8*, 135. [CrossRef]
15. Finnie-Ansley, J.; Denny, P.; Luxton-Reilly, A.; Santos, E.A.; Prather, J.; Becker, B.A. My AI Wants to Know If This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. In Proceedings of the 25th Australasian Computing Education Conference, Melbourne, Australia, 30 January–3 February 2023; pp. 97–104.
16. Malinka, K.; Peresíni, M.; Firc, A.; Hujnák, O.; Janus, F. On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree? In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, Turku, Finland, 7–12 June 2023; pp. 47–53.
17. Phung, T.; Pădurean, V.-A.; Cambronero, J.; Gulwani, S.; Kohn, T.; Majumdar, R.; Singla, A.; Soares, G. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. In Proceedings of the 2023 ACM Conference on International Computing Education Research, Chicago, IL, USA, 7–11 August 2023.
18. Prather, J.; Denny, P.; Leinonen, J.; Becker, B.A.; Albluwi, I.; Craig, M.; Keuning, H.; Kiesler, N.; Kohn, T.; Luxton-Reilly, A.; et al. The Robots Are Here: Navigating the Generative AI Revolution in Computing Education. In Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education, Turku, Finland, 7–12 July 2023; pp. 108–159.
19. Rahman, M.M.; Watanobe, Y. ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Appl. Sci.* **2023**, *13*, 5783. [CrossRef]
20. Rose, K.; Massey, V.; Marshall, B.; Cardon, P. Is Professors' Perspectives on Ai-Assisted Programming. *Issues Inf. Syst.* **2023**, *24*, 178–190. [CrossRef]
21. Yilmaz, R.; Karaoglan Yilmaz, F.G. The Effect of Generative Artificial Intelligence (AI)-Based Tool Use on Students' Computational Thinking Skills, Programming Self-Efficacy and Motivation. *Comput. Educ. Artif. Intell.* **2023**, *4*, 100147. [CrossRef]
22. Yilmaz, R.; Karaoglan Yilmaz, F.G. Augmented Intelligence in Programming Learning: Examining Student Views on the Use of ChatGPT for Programming Learning. *Comput. Hum. Behav. Artif. Hum.* **2023**, *1*, 100005. [CrossRef]
23. Pérez-Mercado, R.; Balderas, A.; Muñoz, A.; Cabrera, J.F.; Palomo-Duarte, M.; Dodero, J.M. ChatbotSQL: Conversational Agent to Support Relational Database Query Language Learning. *SoftwareX* **2023**, *22*, 101346. [CrossRef]
24. Finnie-Ansley, J.; Denny, P.; Becker, B.A.; Luxton-Reilly, A.; Prather, J. The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In Proceedings of the 24th Australasian Computing Education Conference, Virtual, Australia, 14–18 February 2022; pp. 10–19.
25. Mad Daud, S.H. E-JAVA Chatbot for Learning Programming Language: A Post-Pandemic Alternative Virtual Tutor. *Int. J. Emerg. Trends Eng. Res.* **2020**, *8*, 3290–3298. [CrossRef]
26. Carr, N.; Shawon, F.R.; Jamil, H.M. An Experiment on Leveraging ChatGPT for Online Teaching and Assessment of Database Students. In Proceedings of the 2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), Auckland, New Zealand, 28 November–1 December 2023; pp. 1–8.
27. Neumann, A.T.; Yin, Y.; Sowe, S.; Decker, S.; Jarke, M. An LLM-Driven Chatbot in Higher Education for Databases and Information Systems. *IEEE Trans. Educ.* **2025**, *68*, 103–116. [CrossRef]
28. Sankaranarayanan, R.; Mithun, S. Exploring the Effectiveness of AI-Enabled Microlearning in Database Design and Programming Course. In Proceedings of the 2024 IEEE Frontiers in Education Conference (FIE), Washington, DC, USA, 13–16 October 2024; pp. 1–7.
29. Zhang, X. Teaching Tip: Incorporating AI Tools Into Database Classes. *J. Inf. Syst. Educ.* **2025**, *36*, 37–52. [CrossRef]
30. Prakash, K.; Rao, S.; Hamza, R.; Lukich, J.; Chaudhari, V.; Nandi, A. Integrating LLMs into Database Systems Education. In Proceedings of the 3rd International Workshop on Data Systems Education: Bridging Education Practice with Education Research, New York, NY, USA, 9–15 June 2024; pp. 33–39.
31. Boguslawski, S.; Deer, R.; Dawson, M.G. Programming Education and Learner Motivation in the Age of Generative AI: Student and Educator Perspectives. *Inf. Learn. Sci.* **2025**, *126*, 91–109. [CrossRef]
32. Denny, P.; Prather, J.; Becker, B.A.; Finnie-Ansley, J.; Hellas, A.; Leinonen, J.; Luxton-Reilly, A.; Reeves, B.N.; Santos, E.A.; Sarsa, S. Computing Education in the Era of Generative AI. *Commun. ACM* **2024**, *67*, 56–67. [CrossRef]
33. Dunder, N.; Lundborg, S.; Wong, J.; Viberg, O. Kattis vs ChatGPT: Assessment and Evaluation of Programming Tasks in the Age of Artificial Intelligence. In Proceedings of the 14th Learning Analytics and Knowledge Conference, Kyoto, Japan, 18–22 March 2024; pp. 821–827.
34. Bauer, Y.; Leal, J.P.; Queirós, R. Authoring Programming Exercises for Automated Assessment Assisted by Generative AI. *OASIcs* **2024**, *122*, 21:1–21:8. [CrossRef]
35. Fan, G.; Liu, D.; Zhang, R.; Pan, L. The Impact of AI-Assisted Pair Programming on Student Motivation, Programming Anxiety, Collaborative Learning, and Programming Performance: A Comparative Study with Traditional Pair Programming and Individual Approaches. *Int. J. STEM Educ.* **2025**, *12*, 16. [CrossRef]

36. Tan, C.W.; Khan, M.A.M.; Yu, P.-D. AI-Assisted Programming and AI Literacy in Computer Science Education. In *Effective Practices in AI Literacy Education: Case Studies and Reflections*; O'Dea, X., Ng, D.T.K., Eds.; Emerald Publishing Limited: Leeds, UK, 2024; pp. 189–198, ISBN 978-1-83608-852-3.
37. Valový, M.; Buchalcevova, A. The Psychological Effects of AI-Assisted Programming on Students and Professionals. In Proceedings of the 2023 IEEE International Conference on Software Maintenance and Evolution (ICSME), Bogotá, Colombia, 1–6 October 2023; pp. 385–390.
38. Sajja, R.; Sermet, Y.; Cwiertny, D.; Demir, I. Integrating AI and Learning Analytics for Data-Driven Pedagogical Decisions and Personalized Interventions in Education. *arXiv* **2024**, arXiv:2312.09548.
39. Alyoshyna, Y. AI in Programming Education: Automated Feedback Systems for Personalized Learning. Available online: https://essay.utwente.nl/101073/ (accessed on 23 February 2025).
40. Azoulay, R.; Hirst, T.; Reches, S. Large Language Models in Computer Science Classrooms: Ethical Challenges and Strategic Solutions. *Appl. Sci.* **2025**, *15*, 1793. [CrossRef]
41. Lai, C.-H.; Lin, C.-Y. Analysis of Learning Behaviors and Outcomes for Students with Different Knowledge Levels: A Case Study of Intelligent Tutoring System for Coding and Learning (ITS-CAL). *Appl. Sci.* **2025**, *15*, 1922. [CrossRef]
42. Zawacki-Richter, O.; Marín, V.I.; Bond, M.; Gouverneur, F. Systematic Review of Research on Artificial Intelligence Applications in Higher Education—Where Are the Educators? *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 39. [CrossRef]
43. Selwyn, N. *Should Robots Replace Teachers? AI and the Future of Education*; Digital futures series; MA Polity: Cambridge, UK; Medford, MA, USA, 2019; ISBN 978-1-5095-2898-1.
44. Denny, P.; Leinonen, J.; Prather, J.; Luxton-Reilly, A.; Amarouche, T.; Becker, B.A.; Reeves, B.N. Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize AI Code Generators. *arXiv* **2023**, arXiv:2307.16364.
45. Kamalov, F.; Santandreu Calonge, D.; Gurrib, I. New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution. *Sustainability* **2023**, *15*, 12451. [CrossRef]
46. Kazanidis, I.; Pellas, N. Harnessing Generative Artificial Intelligence for Digital Literacy Innovation: A Comparative Study between Early Childhood Education and Computer Science Undergraduates. *AI* **2024**, *5*, 1427–1445. [CrossRef]