

Assignment3_DongminKim.pdf

저자 Dong Min Kim

제출일: 2025년 09월 15일 AM 12:50 (UTC-0400)

제출 아이디: 2751330754

파일명: Assignment3_DongminKim.pdf (228.18K)

단어 수: 2621

글자 수: 12822

Assignment 3

Dong min Kim
dkim614@gatech.edu

1 RESEARCH LOG

1.1 Background

In the last two weeks, my research moved from general ideas about intelligent tutoring systems (ITS) to more concrete strategies that make learning both effective and fun. In my first Research Log, I looked at the basics of ITS like step-by-step learning, mastery learning, adaptive feedback, and metacognition. These ideas gave me the core structure for my SQL tutor app, where learners can move forward only after they really understand the basics.

In my second Research Log, I started to focus more on motivation. I read papers about problem-based learning and game-based learning. They showed that when students work on real problems and take on roles, they keep going longer and enjoy the process more. This made me think of my SQL app not just as a tutor but also as a place where learners act like data analysts, solving missions with real challenges. I also learned that game features should have meaning, not just points or badges with no purpose.

This week, I want to read more about computer science education and also how large language models (LLMs) can help tutoring. I will look at how LLMs can give feedback, hints, and help with errors. My goal is to find out how to use them in a safe and helpful way for my SQL tutor so that they really support learning.

1.2 Papers

1) ChatGPT for good? On opportunities and challenges of large language models for education (Kasneci et al., 2023)

This paper talks about how LLMs can be useful in education but also bring problems like bias and mistakes. It says teachers and schools need to set rules and use them carefully. For my SQL app, I think I need to make sure the AI feedback is checked and safe, not just random answers.

2) Generative Artificial Intelligence and the Education Sector (Ahmad, Murugesan, & Kshetri, 2023)

This one shows how generative AI can change education, making it more creative and faster, but also says there are issues with trust and security. It makes me think I should design my app to be private and responsible. AI is helpful but it should be used slowly and tested.

3) ChatGPT and large language models in academia: opportunities and challenges (Meyer et al., 2023)

The authors write about using ChatGPT in schools and research. They show the good side like saving time but also risks like plagiarism or wrong info. For my project, it means I should connect SQL feedback to real results, so students know the answer is true.

4) Language Models are Unsupervised Multitask Learners (Radford et al., 2019)

This report shows GPT-2 can do many tasks without special training. It was the start of using LLMs for many things. For me, it shows LLM can give SQL hints, but I need to add rules and schema so it does not give nonsense.

5) Large Language Models in Medical Education: Opportunities, Challenges, and Future Directions (Abd-alrazaq et al., 2023)

This paper is about using LLMs in medical learning. It says they are powerful but always need human check because it's a sensitive area. For SQL it is not as dangerous, but I still should have checks like running the query to make sure the feedback is right.

6) A multi-national, multi-institutional study of assessment of programming skills of first-year CS students (McCracken et al., 2001)

This study looked at many schools and found that beginners in CS have trouble solving problems and coding. It shows students need small steps and help. That makes me want my SQL app to teach one small piece at a time and give feedback at each step.

7) A Media Computation Course for Non-Majors (Guzdial, 2003)

This course used media like pictures and sounds to make programming fun for students who are not CS majors. It helped them stay motivated. I think SQL can also use interesting datasets like movies or sports so learning feels more useful.

8) "Georgia Computes!": Improving the Computing Education Pipeline (Bruckman et al., 2009)

This project tried to get more people into computing by giving workshops, new lessons, and teacher support. It worked better when teachers had tools to help them. For my app, I could add reports or dashboards so teachers can also use it in class.

9) Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation (Papastergiou, 2009)

This study shows that games made CS classes more fun and helped students learn better than normal classes. It says clear goals and fast feedback are important. For my SQL app, I should make levels and show results right away so learners stay motivated.

10) Questioning Video Games! Influence on CS Interest (DiSalvo & Bruckman, 2009)

This paper checked if playing games makes students more interested in CS. They found the effect is not big. For me, it means I should not just add game points but make sure SQL tasks feel useful and connected to real skills.

11) Learning in Context: Digital Games and Young Black Men (DiSalvo, Crowley, & Norwood, 2008)

The authors studied how young Black men learn with games and found that context and culture are important. They learn better when the content matches their interests. This gives me the idea that my SQL app should use data people care about, like music or sports.

12) Teaching the Nintendo Generation to Program (Guzdial & Soloway, 2002)

This paper says new students need quick feedback and fun interaction to stay interested. They don't like waiting too long to see results. For my app, I should let users run SQL queries instantly and show the output clearly so they feel progress fast.

1.3 Synthesis

After reviewing these 12 papers, I can see two main directions coming together for my SQL tutor idea. The first is about computer science education. Many papers show that beginners struggle if the steps are too big, so lessons should be broken down and use interesting data or stories to keep students engaged (McCracken et al., 2001; Guzdial, 2003). Teachers also need support, so my app should have features that make it easier to use in class, like dashboards or reports (Bruckman et al., 2009).

The second direction is about large language models. LLMs have big potential to give hints, feedback, and explanations, but they also make mistakes (Kasneci et al., 2023; Meyer et al., 2023). This means I cannot just let the AI answer alone. Instead, I should combine AI with checks, like running the SQL query or comparing to expected results. Some papers show that even in sensitive areas like medicine, human oversight is always needed (Abd-alrazaq et al., 2023). For my project, this means I should design a safe pipeline where the AI gives a suggestion but the system verifies it before showing it to the learner.

Overall, these papers tell me that my SQL tutor should be built on small steps, interesting contexts, and strong validation. LLMs can make the app feel more personal and adaptive, but they need guardrails. If I combine CS education principles with careful AI use, I can make a tool that is both effective and engaging.

1.4 Reflection

This week, reading papers about CS education and LLMs gave me a clearer picture of how to balance teaching basics and using new AI tools. Some papers were a bit heavy and full of technical or policy language, but I tried to rewrite the ideas in my own words. That process helped me understand what is really important for my project.

I realized that I sometimes get too excited about adding AI features, but the research reminded me that the core of learning is still clear steps and good feedback. AI should help, not replace, that process. I also thought about my own learning experiences, where I quit some apps because they were boring or too confusing. These papers gave me confidence that I can design an app that avoids those problems by keeping tasks meaningful and giving safe, fast feedback.

1.5 Planning

Now that the next stage is the Qualifier and Project Proposal, I want to move from collecting research into shaping the project itself in detail. Using what I have learned from my past assignments, I will focus on turning my SQL tutor idea into a clear plan that covers not only the main features but also the structure and scope.

I plan to outline the project in a more concrete way, including the flow of lessons, the role of LLM-based feedback, and the game-like missions with real datasets. I will also sketch the basic system structure, such as how the tutor will check results, how learners will see progress, and how data will be logged for improvement.

Another goal is to prepare a simple project timeline. This will help me decide what to build first, what features can wait for later, and how I can test the early version. By making an outline and schedule now, I can write a stronger proposal that is realistic and grounded in both research and practical planning.

2 ACTIVITY

2.1 Problem Statement

1) Background Information

SQL is an important skill in computer science and jobs with data. Many students need to learn it, but for beginners it is not very easy. At first the syntax looks simple, but when the problems get bigger, they get confused. Most apps or tutorials just give small exercises and check if the answer is right. They don't really explain the steps. Because of that, many learners get stuck and don't feel motivated to keep going.

2) General Problem Statement

The big problem is that students have a hard time learning SQL in a way that feels effective and also interesting. They can practice writing queries, but many times they don't understand why it is correct or wrong. They also don't see how SQL is used in real situations.

3) Scholarly Support

Some research shows that beginners fail when the steps are too big or when there is no feedback (McCracken et al., 2001; VanLehn, 2006). Other papers say motivation is higher when students work on real problems or when tasks feel like a story or game (Barab et al., 2009; Papastergiou, 2009). New tools like ChatGPT can give hints and explanations, but sometimes they are wrong or not safe to trust (Kasneci et al., 2023). Together these studies show there is still a gap between what students need and what current systems give.

4) Specific Problem Statement

The specific problem is that current SQL learning tools don't really give step-by-step guidance, don't give enough real or meaningful practice, and don't use AI feedback in a safe way. Because of this, students lose interest, keep making the same mistakes, and do not build strong SQL skills. This affects not only individual students but also teachers who want to use SQL in their classes.

5) Closing Commentary

If this problem stays the same, students will keep getting frustrated and some may even avoid computer science or data jobs. But if we solve it, we can make a tutor that teaches SQL step by step, uses real datasets to make it feel useful, and gives feedback with AI while checking that the answers are right. I think this kind of system can make SQL learning more fun and more effective.

2.2 Research Questions

1) Main Research Question

How can an SQL tutor use step-by-step guidance, real or game-like tasks, and AI support to help beginners learn better?

2) Sub-Questions

1. What kind of scaffolding, like small steps or hints, helps beginners understand SQL concepts better?
2. How do real-world or game-like tasks change student motivation compared to normal SQL exercises?
3. Can AI tools, including LLMs, give reliable feedback if the system also checks the query result?

3) Justification

These questions are simple to read but still complex to answer. For example, scaffolding can be studied by comparing student progress in different lesson designs. Motivation can be checked by surveys or by seeing if students keep working longer when tasks are like missions. AI feedback can be tested by looking at accuracy and also by comparing how much students improve with AI hints versus static hints.

They are arguable because some people believe games or AI don't help much, and others believe they really do. By asking these questions we can collect data and see what is true. The answers will be both numbers, like accuracy or completion rate, and also student experiences.

Together these questions look at pedagogy (scaffolding), engagement (game/problem tasks), and technology (AI/LLM). It is a more balanced view and not just about one part. I think this makes it useful for building an SQL tutor that matches what students actually need.

3 REFERENCES

1. Abd-alrazaq, A., AlSaad, R., Ahmed, A., Aziz, S., Healy, P. M., Latifi, S., Sheikh, J., Alhuwail, D., Damseh, R., & Alrazak, S. A. (2023). Large language models in medical education: Opportunities, challenges, and future directions. *BMC Medical Education*, 23(1), 1–12.
2. Ahmad, N., Murugesan, S., & Kshetri, N. (2023). Generative artificial intelligence and the education sector. *IT Professional*, 25(5), 17–24. <https://doi.org/10.1109/MITP.2023.3290022>
3. Bruckman, A., Biggers, M., Ericson, B., McKlin, T., Dimond, J., DiSalvo, B., Hewner, M., Ni, L., & Yardi, S. (2009). Georgia Computes!: Improving the computing education pipeline. *ACM SIGCSE Bulletin*, 41(1), 86–90.
4. DiSalvo, B., & Bruckman, A. (2009). Questioning video games: Influences on CS interest. *Communications of the ACM*, 52(4), 142–146.
5. DiSalvo, B., Crowley, K., & Norwood, R. (2008). Learning in context: Digital games and young Black men. *Games and Culture*, 3(2), 131–141.
6. Guzdial, M. (2003). A media computation course for non-majors. *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education*, 104–108.
7. Guzdial, M., & Soloway, E. (2002). Teaching the “Nintendo generation” to program. *Communications of the ACM*, 45(4), 17–21.
8. Kasneci, E., Sessler, K., Bannert, M., Dementieva, D., Gasser, U., Groh, G., Krusche, S., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Seidel, T., Fischer, F., Hüllermeier, E., Küchemann, S., Kutyniok, G., Sailer, M., Schmidt, A., Stadler, M., ... Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274.
9. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B., Laxer, C., Thomas, L., Utting, I., & Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4), 125–140.
10. Meyer, J. G., Urbanowicz, R. J., Martin, P. C. N., Li, R., Peng, P. C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G., Moore, J. H., & O’Connor, K. (2023). ChatGPT and large language models in academia: Opportunities and challenges. *BioData Mining*, 16(1), 1–14.
11. Papastergiou, M. (2009). Digital game-based learning in high school computer science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1), 1–12.
12. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI Blog.

13. VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3), 227–265.
14. Barab, S. A., Gresalfi, M., & Ingram-Goble, A. (2009). Transformational play: Using games to position person, content, and context. *Educational Researcher*, 39(7), 525–536.