Can ChatGPT Generate Code Tasks? An Empirical Study on Using ChatGPT for Generating Tasks for SQL Queries

Ole Werger[a], Stefan Hanenberg[b], Ole Meyer, Nils Schwenzfeier and Volker Gruhn[c]
University of Duisburg–Essen, Essen, Germany

Keywords: Large Language Model, ChatGPT, Empirical Study, User Study.

Abstract: It is now widely accepted that ML models can solve tasks that deal with the generation of source code. Now it is interesting to know whether the related tasks can be generated as well. In this paper, we evaluate how well ChatGPT can generate tasks that deal with generating simple SQL statements. To do this, ChatGPT generated for 10 different database schemas tasks with three different difficulty levels (easy, medium, hard). The generated tasks are then evaluated for suitability and difficulty by exam-correction-experienced raters. With a substantial raters agreement (α=.731), 90.67% of the tasks were considered appropriate (p<.001). However, while raters agreed that tasks, that ChatGPT considers as more difficult, are actually more difficult (p<.001), there is in general no agreement between ChatGPT's task difficulty and rated difficulty (α=.310). Additionally, we checked in an N-of-1 experiment, whether the use of ChatGPT helped in the design of exams. It turned out that ChatGPT increased the time required to design an experiment by 40% (p=.036; d=-1.014). Altogether the present study rather raises doubts whether ChatGPT is in its current version a practical tool for the design of source code tasks.

1 INTRODUCTION

Large language models such as ChatGPT are today quite often used for code generation for given tasks (Le et al., 2020; Chen et al., 2021; Li et al., 2022; Barke et al., 2023; Wang et al., 2023). While solving tasks is an obvious goal, there is a different application of ChatGPT: the generation of examples, respectively tasks. Task generation plays a role whenever one needs to explain some non-trivial technology to someone - and one is just looking for some examples that could be used as it is done in the field of education. In education, lecturers need to create examples to teach and also to test or examine content with students. In such cases, teachers need to design tests or exams from time to time, which requires time. I.e., task generation is an essential, recurring duty and thus it is valid to ask whether ChatGPT is a helpful technology for this.

Our motivation comes from a specific teaching perspective. All authors are frequently involved in the correction of database management exams of a German university where a database management course is part of the curriculum. The exam consists of 50% SQL tasks and other tasks related to topics such as relational algebra, synchronization, recovery, etc. A majority of the tasks on SQL are queries, i.e., SELECT statements. One of the authors designs such exams since more than ten years and a recurring problem is to find appropriate tasks for a given relational schema. The examiner's challenge is to find an appropriate number of easy, medium, and hard tasks and to integrated them into the exam.

Taking into account today's quite enthusiastic perception on ChatGPT[1], it seems plausible to check, if ChatGPT in its current state is able to provide SQL tasks. If this is the case, task generation could be done by ChatGPT and students can decide on their own how many tasks they want to solve: whenever they see the need to practice more, ChatGPT could be an almost infinite source of possible tasks.

While it is plausible to test whether ChatGPT is helpful for the design of SQL tasks, it is not that

[a] https://orcid.org/0009-0007-3226-1271
[b] https://orcid.org/0000-0001-5936-2143
[c] https://orcid.org/0000-0003-3841-2548

[1] See for example:
- https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app
- https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/

Werger, O., Hanenberg, S., Meyer, O., Schwenzfeier, N. and Gruhn, V.
Can ChatGPT Generate Code Tasks? An Empirical Study on Using ChatGPT for Generating Tasks for SQL Queries.
DOI: 10.5220/0012076900003538
In *Proceedings of the 18th International Conference on Software Technologies (ICSOFT 2023)*, pages 303-312
ISBN: 978-989-758-665-1; ISSN: 2184-2833

trivial to determine what helpful actually means and how this can be tested using the scientific method. Actually, we see two plausible perspectives on that. The first perspective comes from the perspective of students who might ask, whether they can ask ChatGPT to generate questions for them that they can use for learning SQL. For them, it is desirable to have a high probability that a task generated by ChatGPT is a valid task, i.e., a task that can be understood and solved (without too much freedom for interpretation or misunderstandings). The other perspective comes from lecturers: It is desirable to reduce the effort to design tasks no matter whether ChatGPT's tasks could be directly used or whether ChatGPT just gives enough inspiration for tasks: In the end, ChatGPT is a valuable tool if it reduces the overall effort.

The present paper studies the usability of ChatGPT as an SQL task generator. Thereto, the paper introduces two different experiments. The first one studies to what extent tasks generated by ChatGPT are considered as appropriate and to what extent a desired difficulty is actually achieved in the generated tasks. The second one studies, to what extent the use of ChatGPT for task generation is efficient or not.

The results of the experiments are as follows. First, ChatGPT is able to provide appropriate tasks: independent raters (with a substantial interrater agreement of α=.731) agreed that 90.67% of the tasks are rather appropriate (than inappropriate) on a 2-point scale (while they hardly agreed on appropriateness on a 4-point scale). Further, the more difficult ChatGPT's tasks are intended to be, the more difficult the raters considered the tasks. Still, on average all tasks are considered as medium tasks from the raters' perspective. I.e., although ChatGPT is able to generate tasks with different difficulties, its notion of difficult tasks does not match the raters' notion of difficult tasks ($p<.001$). Second, in terms of overall effort, we ran an N-of-1 experiment where the effort with ChatGPT was 40% higher than without ($p=.036$).

We see the contribution of this paper in two ways:

- First, there is a technical contribution with respect to the question whether or not ChatGPT can be used as a task generator for SQL tasks.
- Second, the paper can be used as a proposal for studying ChatGPT in a rigorous way that leaves room for detecting potential biases of experiment participants (application of raters agreements, and the execution of a randomized control trial).

For reproducibility, we make our data and the application used for the rating publicly available[2]. The paper is structured as follows. Section 2 gives some background information on ChatGPT. Afterwards, we introduce the first experiment on the appropriateness and difficulty of tasks (Section 3). Then, we introduce the second experiment on the efficiency of exam creation with and without ChatGPT (Section 4). After describing related work in Section 5, we describe threats of validity (Section 6). Finally, we summarize, discuss, and conclude the present paper.

2 BACKGROUND

The focus of the present paper is on ChatGPT as well as on SQL, but we assume that reader is familiar with SQL. Hence, we do not describe SQL here. But since the paper runs a so-called N-of-1 trial, we give some background information on that.

2.1 Large Language Models

Large language models (LLMs) are computer programs that can interpret and generate natural language. These models are trained using machine learning to understand and respond to human speech. An example of a large language model is OpenAI's GPT-3 (Generative Pre-trained Transformer 3), which consists of hundreds of billions of parameters (Brown et al., 2020; Ouyang et al., 2022). LLMs use deep learning, which is based on neural networks. These networks are trained on large text corpora to recognize patterns in sentences. They have been applied in many areas, including machine translation, chatbots, text generation, and even in the creation of creative works such as poems and song lyrics (Mikolov et al., 2013; Vaswani et al., 2017; Liu et al., 2018; Radford et al., 2019; Ouyang et al., 2022).

2.2 Usage of Language Models

When using LLMs, it is especially important that certain models perform particularly well in certain fields of activity. These are, for example, GPT-3 for text generation (Brown et al., 2020) and the BERT model for question-answering tasks (Devlin et al., 2019). In addition, when a request is made to the model (to perform the given task), a process called prompting is used. Prompting uses a piece of text to get the model to generate a certain output type. The process for using LLMs can be described as follows: First, the 'prompt' text, such as a few words or longer paragraphs, is created and sent to the model. The LLM then generates an output based on the given prompt and its internal weights and parameters. The output

[2]The data is available at https://drive.google.com/drive/folders/1MbhhhI77Ixlh-pAT_b3OzvrN2GXPqUfV

can be a single word, a sentence, a paragraph, or even a long text, depending on how the model has been developed and trained. If the output does not match the desired results, the prompt provided can be adjusted to control the model more precisely and to generate the rather expected result on the next try (Brown et al., 2020; Liu et al., 2019; Radford et al., 2019).

2.3 OpenAI ChatGPT

On November 30, 2022, OpenAI unveiled a chatbot powered by the GPT-3 model. In the form of a dialog, the bot can be asked questions or prompted to generate a text in natural language. It was trained using reinforcement learning with human feedback and thus can also produce source code in various programming languages. Source code files were included in its training data, as it was done with its predecessor model InstructGPT (Ouyang et al., 2022). The version used of ChatGPT was released on January 9, 2023[3]. In the meantime, there is also the possibility to request the underlying model ('gpt-3.5-turbo') via API request[4] in order to not be dependent on the chat format. Further, OpenAI states to transform the chat structure into a custom format called ChatML[5] to pass one long sequence of tokens to the LLM.

2.4 Crossover and N-of-1 Trials in Experimentation

The present paper executes in one of its experiments a so-called N-of-1 trial, a special experimental procedure. In experimentation, respectively in empirical software engineering (see for example (Wohlin et al., 2012; Kitchenham et al., 2008)) it is quite common to run so-called AB experiments, where a number of participants do the experiment under treatment A while others do it under treatment B. However, there are so-called crossover trials, where subjects are tested under more than one treatment combination (see (Senn, 2002)). Actually, it is quite common to run crossover trials in software science: according to Vegas et al. more than $\frac{1}{3}$ of experiments in software engineering from 2012–2014 were crossover trials (Vegas et al., 2016, p.123): Testing subjects on multiple treatments is quite common.

A special experimental design among crossover trials are the so-called N-of-1 trials where a single subject is tested on all treatment combinations (see (Mirza et al., 2017)), an experimental procedure that was already documented in 1676 (Wiseman, 1676, p. 123). While such experiments do not appear that often in software science (see (Hanenberg and Mehlhorn, 2022)), they are standard experimental procedures in traditional empirical disciplines. For example, Perdices et al. found that 39% of the studies in the PsycBITE evidence database were single-case studies (Perdices et al., 2006).

When designing N-of-1 trials (respectively crossover-trials in general), it is necessary to consider potential carry-over effects (see (Kitchenham et al., 2003; Madeyski and Kitchenham, 2018)), i.e., effects when a previous treatment influences measurements of later treatments. Such carry-over effects are for example learning effects, fatigue effects, or novelty effects. Strategies to reduce such undesired effects are the use of breaks in the protocol (so-called washout periods, see for example (Evans, 2010, p. 10)), or small modifications in the treatment combinations.

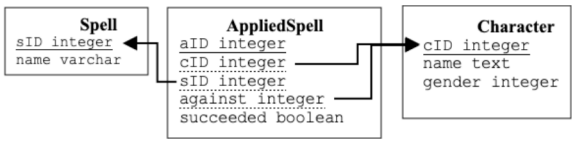

Figure 1: Example schema from exam.

3 FIRST EXPERIMENT: RATINGS

The general question for the first experiment is if the tasks generated by ChatGPT for different difficulties are appropriate tasks. We used the difficulty levels simple, medium, and hard.

Since we had access to database exams from 2017 until today, our goal was to design tasks for these already existing schemas where each schema consists of only three to four tables. In each year, there are two database exams, i.e., we have access to 12 exams in total. We removed two exams (2021/1 and 2022/2) as they used a similar schema. Figure 1 illustrates such a schema as used in one exam (the schema comes from the Harry Potter world where a character can cast a spell on some other character).

3.1 Experiment Layout

The experiment consists of the following variables.

- **Dependent Variables:**
  - **Rated Appropriateness of Tasks (1-4):** Each rater rates a task whether or not it is considered as a valid task on a Likert-scale from 1-4 (1=

[3]https://help.openai.com/en/articles/6825453-chatgpt-release-notes

[4]https://openai.com/blog/introducing-chatgpt-and-whisper-apis

[5]https://github.com/openai/openai-python/blob/main/chatml.md

"*yes, can be used*", 2="*can be used but requires minor revisions*", 3="*can rather not be used*", 4="*cannot be used*").

- **Rated Difficulty (1-3):** Each rater rated a task with respect to its difficulty (1=*simple*, 2=*medium*, 3=*hard*).

- **Independent Variables:**
  - **ChatGPT's Difficulty (1-3):** The difficulty ChatGPT was asked to generated (1=*simple*, 2=*medium*, 3=*hard*).
  - **Exam:** 10 exams (2017/1, 2017/2, ..., 2022/1).[6]
- **Fixed Variables:**
  - **Number of Repetitions:** 10 repetitions for each treatment combination (i.e., ChatGPT's difficulty and exam).
  - **Ordering:** The tasks were randomly ordered before showing to the raters (each rater in the same ordering).

The task in the experiment was: "*Rate the given task for the given schema according to its appropriateness (*1="*yes, can be used*", 2="*can be used but required minor revisions*", 3="*can rather not be used*", 4="*cannot be used*"*) and difficulty (*1=*simple*, 2=*medium*, 3=*hard)*".

3.2 Experiment Execution

From the chosen 10 exams the relational schemas were taken and ChatGPT was requested to generate 10 easy, 10 medium, and 10 hard tasks (each request was done in a separated prompt where we passed each time the corresponding schema to). The resulting 300 tasks were then randomly ordered. Then, a simple application was given to the three raters that showed the relational schema (as text) and the question. Then, the raters rated the appropriateness and the difficulty. ChatGPT's difficulty was not shown to the raters. Figure 2 illustrates the situation inside the app, when the rater was shown task 42, but did not decide yet.

Randomizing the order of the tasks was done to potentially reduce carryover effects. Raters were given as much time and breaks as they needed.

3.3 Tasks Appropriateness

Computing Krippendorf's alpha[7] directly on the dependent variable appropriateness leads to rather disillusioning results: Krippendorff's alpha for ordinal scaled data (α=.4972, N=300, 3 raters)[8] as well as Fleiss' Kappa for nominal scaled data (κ=.311) reveal no acceptable interrater agreement.[9] Although the previous statistics are common in the literature, there are still no standard ways to report the interrater agreement. Because of that, we follow the guidelines by Kottner et al. (Kottner et al., 2010, p.241) who propose to report proportions of agreements. Table 1 illustrates the results. While 138/300 ratings were identical and 150 ratings differed by one rater, 12 ratings were completely different between raters.[10]

However, a closer look at the ratings reveals that

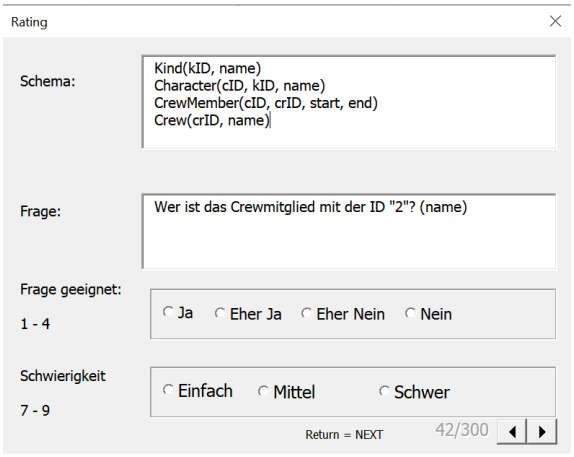

Figure 2: Used application for rating tasks (the application was written in German, "*Frage geeignet*" = "*question appropriate*", "*Schwierigkeitsgrad*" = "difficulty").

Table 1: Rated Appropriateness: Absolute results of interrater agreement on 300 tasks (three raters). *Complete* agreement describes all raters rated gave the identical rating, *None* describes all raters had different ratings, 2/3 described two raters agreed and one had a different rating.

Complete	1 only	2 only	3 only	4 only	Sum
	115	16	6	1	138

None	1/2/3	1/2/4	1/3/4	2/3/4	Sum
	3	2	4	3	12

2/3	1/2	1/3	1/4	2/3	2/4	3/4	Sum
	116	6	1	6	3	18	150

[6]Two exams do not occur in the present study: 2022/2 and 2021/1, because both exams used schemes from a previous exam.

[7]All statistical analyses in the present work were done using SPSS v27.

[8]Krippendorff's alpha was computed in SPSS (Hayes and Krippendorff, 2007) using the script that is available via Hayes' webpage http://afhayes.com/spss-sas-and-r-macros-and-code.html.

[9]Landis and Koch classify a κ with $.21 \leq \kappa \leq .4$ as fair (which does not sound bad, but it is the third lowest class of agreement on a 6-point scale from *almost perfect* to *poor*, see (Landis and Koch, 1977, p. 165)).

[10]The figure is compressed in a way that no details are shown for the 2/3 agreements. For example, 1/3 contains all ratings where two raters voted for 1 while one rater voted for 3 and additionally all ratings where two raters voted for a 3 and one voted for a 1.

Table 2: Simplified Appropriateness: *Complete* means all reviewers considered a task as rather appropriate or inappropriate, *None* describes that not all raters agreed.

Complete	Rather appropriate	Rather inappropriate	Sum
	247	25	272
None	**2x rather appropriate**	**2x rather inappropriate**	**Sum**
	11	17	28

the reviewers varied between the classification 1 and 2 as well as and 3 and 4 a lot. But it looks like there are not many disagreements between the classes *rather appropriate* (i.e., rated as 1 or 2) and *rather inappropriate* (i.e., rated as 3 or 4). Thus, we repeated the calculation of the interrater agreement, this time only on a 2-point scale (1 = rather appropriate, 2 = rather inappropriate). The result is a substantial agreement with respect to Fleiss' Kappa (κ=.731)[11] as well as Krippendorff's alpha (α=.731).[12] In the following, we call this new dependent variable *simplified appropriateness*. The results of the absolute ratings can be found in Table 2: 272 ratings were identical (from which 247 were considered as appropriate).

Because of the previous analysis, we consider the distinction between rather appropriate and rather inappropriate as substantial.

Next, we check whether the *assumed difficulty* by ChatGPT determines the simplified appropriateness. Thereto, we mark all datasets where the raters agreed or disagreed and call this variable *simplified appropriateness agreement*. I.e., we apply a χ^2-test on the variables *simplified appropriateness agreement* (yes/no) and *assumed difficulty* (simple/medium/hard). The test does not show significant differences in the agreement between (χ^2(2, N=300) = 1.497, p=.473).

To test if there are differences between the exams and the *simplified appropriateness agreement* we apply a χ^2-test on both variables. Again, we find no significant difference (χ^2(9, N=300) = 10.084, p=.344).

We conclude from the previous findings that the *simplified appropriateness agreement* does not neither depend on the exams nor on the assumed difficulty.

To check if the number of tasks considered as appropriate differ from the number of tasks that are considered as inappropriate, we run a paired t-test for each exam (i.e. we compare the number of appropriate respectively inappropriate tasks) which reveals for the tasks that were considered appropriate (M=27.2=90.67%, SD=1.68) and those that were considered not appropriate (M=2.8=9.33%, SD=1.68) a significant and large difference (t(9)=22,875, p<.001, d=3.373, $CI_{96\%}$=[21.99; 26.81]).[13] I.e., with 95% confidence the difference between appropriate and inappropriate tasks was between $\frac{21.99}{30}$=73.3% and $\frac{26.81}{30}$=89.4% in each exam.[14]

Whether the level of appropriateness is considered as good or bad lies in the eye of the beholder: while one could argue that 90.67% appropriate tasks are not bad, one could also argue that 10% inappropriate tasks are a quite high error rate.

3.4 Exploratory Experiment on Appropriateness

A recurring statement from the raters after the ratings was that ChatGPT had problems with relational schemas where the table name *kind* was used. It appeared twice in schemas (2019/2 and 2022/1) expressing different characters and their kind (e.g. Harry Potter is of kind Human, while Dobby is of kind House Elf). The identical word in German has a different meaning (*Kind* in German is translated to *child* in English), which caused ChatGPT to generate some meaningless questions (such as "*show the name of all children who are students*").

Due to this, we checked, whether the existence of the word kind in the schema influenced the appropriateness of the tasks, i.e., we applied a χ^2-test on the variables *simplified appropriateness agreement* (yes/no) and *has child in scheme* (yes/no). The test reveals significant differences (χ^2(1, N=300) = 4.766, p=.029): without the word kind in the schema $\frac{18}{240} = 7.5\%$ of the tasks were considered inappropriate, while for a schema with the word kind in it $\frac{10}{50} = 20\%$ of the tasks were considered inappropriate.

3.5 Task Difficulty

In addition to the question whether a task is considered appropriate, raters also determined the rated difficulty. Computing the rater agreement directly on the

[11]The terminology substantial comes again from Landis and Koch (Landis and Koch, 1977, p. 165) and represents the second highest rater agreement on the 6-point scale.

[12]Both results are identical, because for a 2-point scale the distinction between nominal and ordinal data is not given. It should be emphasized that Krippendorf's interpretation is slightly more reserved and following his interpretation reliabilities between $\alpha = .667$ and $\alpha = .800$ should only be used for "*for drawing tentative conclusions*" (Krippendorff, 2004, p. 241).

[13]The Shapiro-Wilk test for normality is not significant with p=.283. But even if one considers the non-parametric test as better suited due to the small sample size, the result is the same (Z=-2.825, p=.005).

[14]Instead of t-test a non-parametric Wilcoxon-test could be used, but it has the same results (Z=-2.825, p=.005).

raw data leads again to frustrating results: Krippendorff's alpha for ordinal scaled data (α=.589, N=300, 3 raters) as well as Fleiss' Kappa for nominal scaled data (κ=.498) reveal again no interrater agreement that meets the standard bars.[15]

We removed 53 cases from the dataset, in which the raters agreed that they were inappropriate, so the difficulty could not be considered, or in which the raters disagreed on appropriateness. The remaining tasks are called *rather appropriate tasks*.

Repeating the previous analysis on the remaining dataset does not change much: Krippendorff's alpha for ordinal scaled data (α=.618, N=247, 3 raters) as well as Fleiss' Kappa for nominal scaled data (κ=.529) reveal at least a moderate agreement.[16] However, at least with respect to Krippendorff's alpha the value is above .6 which implies that we can use the values at least for drawing tentative conclusions. I.e., in the following we use the rater values under the assumption of a substantial agreement, being aware that we should handle the results with more caution than the previous rated appropriateness.

Again, the goal is to check whether the level of agreement depends on the other factors such as the *assumed difficulty* or the *exams*. Hence, we mark all datasets where the raters agreed or disagreed on and call the resulting variable *difficulty agreement*. In order to check whether the difficulty agreement depends on the *assumed difficulty* or the *exams* we run (likewise to Section 3.3) a χ^2-test – and receive noteworthy differences to Section 3.3: The χ^2-test on the *difficulty agreement* and *assumed difficulty* reveals significant differences ($\chi^2(2, N=247) = 8.600$, p=.014) as well as comparison between *difficulty agreement* and *exams* ($\chi^2(9, N=247) = 20.685$, p=.014).

With respect to *assumed difficulty* we see a comparable agreement for easy and medium tasks ($\frac{66}{24} = 2.75$, $\frac{50}{31} = 1.61$), for hard tasks the number of agreements and disagreements is almost equal ($\frac{39}{37} = 1.05$): The more difficult ChatGPT's tasks are intended to be, the less often do raters agree on the difficulty.

With respect to the exams, it is hard to detect any regularity in the comparisons: while there are exams with a clear majority of agreements (2019/1: $\frac{25}{5} = 5$), there are exams where the number of agreements and disagreements are almost equal (2021/2: $\frac{14}{15} = .93$) as well as exams where the disagreements are much higher than the agreements (2019/2: $\frac{6}{13} = .46$).

While it is already noteworthy that agreements between raters are not that high as assumed, it is more interesting to know what the relationship between *ChatGPT's difficulty* and the *rated difficulty* is.

A question with this comparison is how it should be done, because one must not forget that in the sample of *rather appropriate tasks* there are still a few tasks where the raters did not agree on the difficulty.

One way we see is to compute an average rating of all three raters (called *"average rated difficulty"*) and to compare it with ChatGPT's difficulty. I.e., we run a one-way ANOVA on the independent variable *ChatGPT's difficulty* (with the treatments "easy", "medium", and "hard") and the dependent variable *average rated difficulty*. The result shows a significant impact of *ChatGPT's difficulty* on *average rated difficulty* ($F(2, 244)=26.515$, p<.001, η_p^2=.179).[17] Running a Tukey post-hoc test reveals that the differences between the ChatGPT difficulties lead to significant different ratings (see Table 3).

Figure 3 shows that as ChatGPT's difficulty increases, the average rated difficulty also increases. However, upon closer examination, the mean for medium and hard tasks is still below 2, indicating that human raters do not consider any of the tasks to be

Table 3: Results of the one-way ANOVA on the independent variable *average rated difficulty* and the dependent variable *ChatGPT difficulty*. Below the ANOVA are the results of the Tukey post-hoc test.

Variable	df	F	p	η_p^2	treatments	M
ChatGPT's Difficulty	2	26.515	<.001	.179	Easy	1.23
					Medium	1.53
					Hard	1.74

Difficulty	Difficulty	Diff	$CI_{95\%}$	p
Easy	Medium	-.301	[-.457, -0.135]	<.001
	Hard	-.515	[-.684, -.347]	<.001
Medium	Hard	-.214	[-.387, -.042]	.010

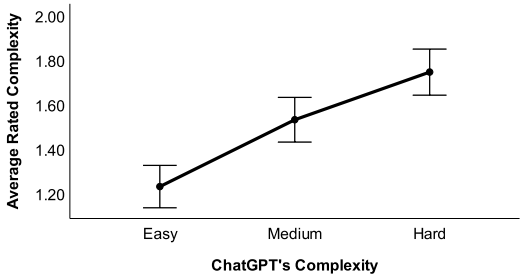

Figure 3: Effect of ChatGPT's difficulty on the average rated difficulty (including 95% confidence intervals).

[15] Landis and Koch classify κ with $.4 \leq \kappa \leq .6$ as a moderate agreement (Landis and Koch, 1977, p.165).

[16] Again, we follow here Landis' and Koch's terminology (Landis and Koch, 1977, 165).

[17] Actually, it is disputable whether the results of the ANOVA should be reported here, because the data violates the assumption of homogeneity of variance tested via Levene's test for ($F(2, 244)=9.858$, p<.001). However, running the non-parametric Kruskall-Wallis test reveals comparable results as the ANOVA ($\chi^2(2, N=247)=44.14$, p<.001). Due to this, we report keep the ANOVA's result in the paper and just report additionally the non-parametric test here.

difficult. One can simply check the statement "the average ratings are below 2" by running a one-sample t-test against the constant 2 and one gets significant results (M=1.483, SD=.504, t(246)=-16.197, p<.001).

Considering the previous results, it is plausible to test if the rated difficulties match the difficulties of ChatGPT. Actually, the result is already clear from the previous results, but for completeness it makes sense to make it explicit. Checking the interrater agreement between ChatGPT and the human raters can be done by using ChatGPT as a separate rater. Doing so leads to a low interrater agreement (α=.4265, κ=.328).

Comparing the human ratings with ChatGPT's rating, we used majority rating to avoid mixing ordinal scale and decimal numbers, so the most frequent rating is selected. We call the resulting variable the *majority rated difficulty*. We are aware that this comparison increases the potential agreement, since a disagreement between raters is rounded towards the majority. We removed three tasks where not at least two out of three raters did agree, because majority voting is then not applicable.

However, even under these optimistic assumptions, which tend to increase the raters agreement, there was hardly any agreement between ChatGPT and the reviewer (α=.310).

4 SECOND EXPERIMENT: USER PERFORMANCE

The previous experiment showed ChatGPT tasks may not be trustworthy without manual inspection before inserting into teaching tools.

From our teaching motivation, the question is also whether ChatGPT could help designing exams. Tasks which can not be used will not be a problem as long as other appropiate tasks are generated. Further, if tasks need some manual effort, they can be helpful in terms of inspiration. This second experiment focuses on user performance, we ask if the use of ChatGPT improves the time required to design SQL exams.

One of the authors of the present paper designs the database examples for his university since more than 10 years, while the others did not in the past. Because of that, we consider only him as an appropriate expertised participant for the experiment. Therefore, from our point of view, it seemed quite natural to conduct an N-of-1 trial with this single participant.

For running an N-of-1 trial it is necessary to determine the ordering of treatments as well as the exact definition under which a treatment is given. Removing too similar schemas from those used in the previous experiment and adding a new one, with also three tables leads to seven schemas for this experiment.[18] We randomly ordered these seven schemas and gave the participant the schemas and asked the participant to design an exam with 35 points alternatively without any help and with ChatGPT. After the seven exams were done, we repeated the same procedure, but changed the treatments (i.e., this time the first exam with ChatGPT, the second without, etc.), so that we counterbalanced each treatment with its alternative.

4.1 Experiment Layout

The experiment consists of the following variables.

- **Dependent Variable:** Time to finish (The total time required by the design of an SQL exam).
- **Independent Variables:** Techniques (with the treatments Manually/ChatGPT).
- **Fixed Variables: Number of Exams** (7 exams)

The task in the experiment was: "*Design an SQL exam with 35 points*".

4.2 Experiment Execution

The participant was permitted to decide on his own when to do a break, whereby a break was only permitted between two exams. When working with ChatGPT, the participant was asked to start the design of an exam with asking ChatGPT for 10 easy, 10 medium, and 10 hard tasks within one prompt. Additionally, the participant was permitted to interact with ChatGPT later on if desired. It was left to the participant to decide how the results of ChatGPT were used.

4.3 Results and Informal Interview

The experiment was analyzed using a paired-sample t-test (see Table 4). The manual design of SQL exams needed significantly less time than the ChatGPT supported design of the exams. With support it took 40% more time (p=.036, Cohen's d=-1.014).[19]

In an informal interview, the participant articulated that the request to ChatGPT did not take much time and the response times of ChatGPT were not disturbing. However, he felt that he used only a small

[18]The motivation for this additional schema lies in the fact that 6 measurements would only permit to show significant differences if in all cases one treatment requires more time than the other.

[19]We ran the t-test because the Shapiro-Wilk test for normality was not significant (p=.908). However, even the non-parametric Wilcoxon test reveals significant differences (Z=-2.028; p=.043).

Table 4: Raw measurements and statistical results of the second N-of-1 trial (with Treatments (tr.) M=treatment manually and ChatGPT=treatment with ChatGPT). Measurements are second rounded to full seconds.

	Schema						
Tr.	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**
M	1401	705	1200	866	1008	1062	1064
GPT	1193	1822	1377	1310	1658	1479	1373

T-Test	**df**	**T**	**M**	**$CI_{95\%}$**	**p**
M - GPT	6	-2.684	-415	[-794; 37]	.036

Tr.	**M**	**SD**	**$CI_{95\%}$**	**Ratio**
M	1044	223.76	[836; 1250]	$\frac{M_{GPT}}{M_M} = 1.40$
GPT	1459	215.58	[1259; 1658]	

number of tasks from ChatGPT per exam. He also articulated that there were situations where he was willing to use a task from ChatGPT, but it was not usable (using attributes that were not in the given schema). The participant felt that reading tasks from ChatGPT, thinking about them (and finally leaving such tasks out), and finding either better tasks or slightly better task formulations took more time than designing an exam manually. He felt that context switching (between reading, thinking, and writing) using ChatGPT was more stressful than designing an exam manually, where one only has to think about possible tasks and write them down.

5 RELATED WORK

These days, some researchers are concerned with the evaluation of ChatGPT. Comparable works to the presented one and some additional are listed below.

Kung et al. (Kung et al., 2023) have confronted the chatbot with the tasks from the United States Medical Licensing Exam (USMLE) exam. In doing so, the chatbot scored average, but this is equivalent to passing the exam. The authors conclude from their experiment that, on the one hand, the language models are getting better and better (earlier models had not yet passed) and, on the other hand, that in the education of adolescent medical students it may be helpful to support them with AI tools (Kung et al., 2023).

Another group around Gao et al. (Gao et al., 2023) worked on abstract/text generation for medical publications. They only specified the title and style of the publishing journal and asked ChatGPT to generate a suitable abstract. The generated texts and the original texts written by humans were checked in three ways: Both an 'AI output detector' and a 'plagiarism detector' were applied by machine. The first one detected the generated texts applicable while the second one could not detect any plagiarism of existing texts. In addition, some human reviewers were asked to distinguish between the generated and original texts. The humans found it much more difficult to recognize the generated texts (about 68%), while they also classified original texts as generated. The authors conclude that it is possible to recognize generic texts and yet ChatGPT is already able to generate abstracts based on the given titles (Gao et al., 2023).

Many different aspects of ChatGPT are considered by Bang et al. (Bang et al., 2023). They run a lot of experiments on existing datasets to evaluate the response of ChatGPT. To compare the results, appropriate metrics are used for each dataset so that the authors do not have to decide on task fulfillment. In language processing tasks, ChatGPT seems to outperform other models in most cases, while it still has problems especially with non-Latin script languages. The authors see a problem in the reasoning generation, which often fails or is misleading. However, the interactivity with the model is praised in both quantitative and qualitative evaluations (Bang et al., 2023).

6 THREATS TO VALIDITY

From our perspective, the following threats exist.
ChatGPT Output: We used the generated tasks exactly as they were, without making corrections such as spelling errors. We believe that these potential errors have an impact on the appropriateness rating.
Ratings: The rating of the ChatGPT output was performed by three raters with no prior training. We deliberately did this because the raters were already experienced database examiners. Therefore, we cannot provide any formal training material or rating guidelines to help others to repeat the rating.
Rating Scales: The rating scales used are based on assumptions and we have shown in one experiment that these had to be adjusted/reframed. In particular, the definition of the degrees of difficulty is not necessarily unambiguous, so that discrepancies may arise.
Usage of ChatGPT: Using ChatGPT, there is basically no restriction on the exact text formulations. We have used almost the same syntax in each query to produce similar results. Other formulations or application methods may achieve different results. In the performance experiment, we did not require any specific use of ChatGPT, but advised the participant to start with the task generation when using ChatGPT.
Used Database Schemas: We used the existing database schemas from the last 10 university exams. We think they are simple enough to generate easily understandable tasks. Nevertheless, we found problems with the use of the word 'kind', which has dif-

ferent meanings in German and English, possibly due to the fact that German is the natural language used but English is used for the database schemas.

N-of-1 Experiment: We believe that it is necessary to have a person who is trained in designing exams. Due to this, we decided to have only one participant do this task. The participant has many years of experience in exactly this task and thus we think that he is a valid expert for such task. However, we do not know how his performance can be compared to other universities or other lecturers.

7 DISCUSSION AND CONCLUSION

The present paper performed two experiments on the generation of SQL tasks by ChatGPT. Both experiments do not support the rather positive perception of ChatGPT's usability that one finds today. I.e., we consider the results of the study rather as a reminder that it is currently not as productive as is might appear.

While the first experiment revealed that most tasks generated by ChatGPT are rather appropriate tasks (the raters agreement on a 4-point scale was too low and had to be reduced to a 2-point scale), and while more difficult tasks from ChatGPT's perspective are also more difficult tasks from the raters' perspective, ChatGPT was mainly able to generate easy tasks: ChatGPT's perspective on difficulty differed substantially from the raters' perspectives. These results suggest that ChatGPT should currently not be used in a way where generated tasks are automatically integrated without additional manual inspections. The second experiment addresses rather the performance issue of using ChatGPT, i.e., the question whether or not a certain work (in our case the design of SQL exams) can done more time efficient with the help of ChatGPT. The N-of-1 experiment (where the participants was an expert for that work) revealed that the expert required 40% more time with ChatGPT.

We are aware that the presented experiments can be seen only as a first step towards the evaluation of ChatGPT. First, none of the experiments took either complex or more controlled interactions with ChatGPT into account: in the first experiment, the ChatGPT's generated tasks were directly used, in the second experiment the participant was not advised to follow certain, specific interaction patterns. However, it is quite plausible to us that the code generated from ChatGPT is not only a question of how good some text is written into the prompt, but how users interact with ChatGPT via the prompt by asking questions, reacting on the answers, ask follow-up questions, etc.

We definitely see a need for more studies on ChatGPT, including more complex interactions. However, we still think that the given study design – which relies not only on the precision of answers but also on the overall performance – could be used in follow-up studies to evaluate the possible benefits or drawbacks.

Hence, the present study can be summarized with the sentence: ChatGPT in its current form where users rather apply simple interactions with ChatGPT is so far not a performance boost as one might expect – at least not for generating tasks.

REFERENCES

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., et al. (2023). A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Barke, S., James, M. B., and Polikarpova, N. (2023). Grounded copilot: How programmers interact with code-generating models. *Proc. ACM Program. Lang.*, 7(OOPSLA1).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Chen, M., Tworek, J., Jun, H., Yuan, Q., et al. (2021). Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Evans, S. R. (2010). Clinical trial structures. *Journal of experimental stroke & translational medicine*, 3(1):8–18. PMC3059315[pmcid].

Gao, C. A., Howard, F. M., Markov, N. S., Dyer, E. C., et al. (2023). Comparing scientific abstracts generated by chatgpt to real abstracts with detectors and blinded human reviewers. *npj Digital Medicine*, 6(1):75.

Hanenberg, S. and Mehlhorn, N. (2022). Two n-of-1 self-trials on readability differences between anonymous inner classes (aics) and lambda expressions (les) on java code snippets. *Empirical Softw. Engg.*, 27(2).

Hayes, A. F. and Krippendorff, K. (2007). Answering the call for a standard reliability measure for coding data. *Communication Methods and Measures*, 1(1):77–89.

Kitchenham, B., Fry, J., and Linkman, S. (2003). The case against cross-over designs in software engineering. In *Eleventh Annual International Workshop on Software Technology and Engineering Practice*, pages 65–67.

Kitchenham, B. A., Al-Kilidar, H., Babar, M. A., Berry, M., et al. (2008). Evaluating guidelines for reporting empirical software engineering studies. *Empir. Softw. Eng.*, 13(1):97–121.

Kottner, J., Audigé, L., Brorson, S., Donner, A., et al. (2010). Guidelines for reporting reliability and agreement studies (GRRAS) were proposed. *J Clin Epidemiol*, 64(1):96–106.

Krippendorff, K. (2004). *Content Analysis: An Introduction to Its Methodology (second edition)*. Sage Publications.

Kung, T. H., Cheatham, M., Medenilla, A., Sillos, C., et al. (2023). Performance of chatgpt on usmle: Potential for ai-assisted medical education using large language models. *PLOS Digital Health*, 2(2):1–12.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1).

Le, T. H. M., Chen, H., and Babar, M. A. (2020). Deep learning for source code modeling and generation: Models, applications, and challenges. *ACM Comput. Surv.*, 53(3).

Li, Y., Choi, D., Chung, J., Kushman, N., et al. (2022). Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., et al. (2018). Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Liu, Y., Ott, M., Goyal, N., Du, J., et al. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Madeyski, L. and Kitchenham, B. A. (2018). Effect sizes and their variance for AB/BA crossover design studies. *Empir. Softw. Eng.*, 23(4):1982–2017.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Mirza, R., Punja, S., Vohra, S., and Guyatt, G. (2017). The history and development of n-of-1 trials. *Journal of the Royal Society of Medicine*, 110(8):330–340. PMID: 28776473.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., et al. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., et al., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Perdices, M., Schultz, R., Tate, R., Mcdonald, S., et al. (2006). The evidence base of neuropsychological rehabilitation in acquired brain impairment (abi): How good is the research? *Brain Impairment - BRAIN IMPAIR*, 7:119–132.

Radford, A., Wu, J., Child, R., Luan, D., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Senn, S. (2002). *Cross-over Trials in Clinical Research*. Statistics in Practice. Wiley.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., et al. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., et al., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vegas, S., Apa, C., and Juristo, N. (2016). Crossover designs in software engineering experiments: Benefits and perils. *IEEE Transactions on Software Engineering*, 42(2):120–135.

Wang, X., Liu, X., Zhou, P., Liu, Q., et al. (2023). Test-driven multi-task learning with functionally equivalent code transformation for neural code generation. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ASE '22, New York, NY, USA. Association for Computing Machinery.

Wiseman, R. (1676). *Eight Chirurgical Treatises*. B.Tooke, J. Knapton, T. Horne, forth edition.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., et al. (2012). *Experimentation in Software Engineering*. Computer Science. Springer Berlin Heidelberg.