CSIRO-LT at SemEval-2025 Task 8: Answering Questions over Tabular Data using LLMs

Tomas Turek[1,2*] and **Shakila Tonni**[1*] and **Vincent Nguyen**[1]
Huichen Yang[1] and **Sarvnaz Karimi**[1]
CSIRO's Data61[1], Sydney, Australia
RMIT University[2], Melbourne, Australia
firstname.lastname@csiro.au
[*] indicates co-first authors

Abstract

Question Answering over large tables is challenging due to the difficulty of reasoning required in linking information from different parts of a table, such as heading and metadata to the values in the table. We investigate using Large Language Models (LLM) for tabular reasoning, where, given a pair of a table and a question from the *DataBench* benchmark, the models generate answers. We experiment with three techniques that enable symbolic reasoning through code execution: (1) a direct code prompting (DCP) approach, DCP_{Py}, which uses Python; (2) Multi-Step Code (MSC) prompting MSC_{SQL+FS} using SQL and ReAct prompting; and, (3) MSR_{Py+FS}, which combines multi-step reasoning (MSR), few-shot (FS) learning and Python tools. We also conduct an analysis exploring the impact of answer types, data size, and multi-column dependencies on LLMs' answer generation performance, including an assessment of the models' limitations and the underlying challenges of tabular reasoning in LLMs.

1 Introduction

Table Question Answering (Table QA)—answering questions on a table—has multiple applications in different domains, such as in finance (Nararatwong et al., 2025; Nararatwong et al., 2024; Papicchio et al., 2023; Chen et al., 2022; Zhao et al., 2022; Zhu et al., 2021; Chen et al., 2021) and scientific literature (Zhao et al., 2024a: Ghosh et al., 2024; Korkmaz and Del Rio Chanona, 2024; Katsis et al., 2022; Moosavi et al., 2021). In some scientific domains, such as medicine, tables contain information that is not present in the text of a research paper (Bardhan et al., 2024; Johnson et al., 2023; Bardhan et al., 2022; Park et al., 2021). Answering questions based on information hidden in tables faces challenges such as dealing with table structure, size, linking headings to the content and in the case of numerical values, may require mathematical operations over multiple cells (Deng et al., 2024; Wu et al., 2024; Nahid and Rafiei, 2024; Wu et al., 2023; Pal et al., 2023; Cheng et al., 2022).

Table QA in natural language processing or information retrieval often includes identifying a table in a text that can answer a question and then reasoning and generating an answer (Pramanick et al., 2024; Ji et al., 2024; Dong et al., 2024; Zhao et al., 2024a; Wan et al., 2024; Herzig et al., 2021). However, we focus on a subproblem where a table is provided along with the question. We report on our participation in SemEval shared task[1] where given a question and table, the task is to determine the answer and the type of answer within the required columns. The question must only be answered with tabular data from the provided dataset, *DataBench* benchmark.

We investigate using LLMs to generate answers over tables with symbolic reasoning through code execution to select the appropriate columns and rows for a given question. Namely, a direct code prompting approach, DCP_{Py}; multi-step code prompting with few-shot learning, MSC_{SQL+FS}; and, ReAct (Yao et al., 2023) prompting, MSR_{Py+FS}, which uses multi-step reasoning with few-shot learning. Furthermore, we analyse the connection between the LLMs' answering capacity and the answer types, data sizes and cases when multiple columns are required to answer a question.

2 Related Work

Transformer-based models Some existing table QA models are transformer-based and pre-trained on tables and texts from Wikipedia. For example, TaPas (Herzig et al., 2020) with BERT encoder for parsing tabular structures, TableFormer with modified TaPas by learnable attention bias for en-

[1] https://semeval.github.io/SemEval2025

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1690–1701

coding tables (Yang et al., 2022), TaBERT with BERT encoder for a joint understanding of textual and tabular data (Yin et al., 2020), TaPEx with pre-trained BART encoder-decoder on executable SQL queries (Liu et al., 2022), and OmniTab with pre-trained TaPEx on natural language questions (Jiang et al., 2022). However, the performance of these models decreases with out-of-domain distributions and adversarial data, e.g., variations in table header and content (Zhao et al., 2023c).

Open-source generalist LLMs Several studies investigated these models for Table QA (Pal et al., 2024; Zhang et al., 2024a; Zhao et al., 2023a; Zhao et al., 2023b). These models have yet to demonstrate generalisation abilities on out-of-domain datasets and tasks. Zhang et al. (2024c) fine-tuned Llama-2-7b on the instruction data for several table-based tasks without incorporating task-specific designs. Their experiments revealed that instruction tuning improved performance with transformer-based models in in-domain settings but not for out-of-domain settings.

Osés Grijalba et al. (2025) experimented with direct and code prompting to answer the questions. For direct prompting, they used LLAMA2 models, and for direct code prompts, they used Code-LLAMA models with 7B and 13B parameters and found the Code-LLAMA to be the overall best model on *DataBench* benchmark.

Table structure One promising direction to improve table QA is to develop capabilities for understanding table structure, such as the table schema and row and column semantics. When fine-tuned, models can better locate answers over tables to predict the probability of containing the answer to a question in the rows and columns of tables (Glass et al., 2021). Retrieving the relevant columns and rows from tables with millions of tokens can boost the performance of LLMs while reducing computational complexity (Chen et al., 2024).

Zhao et al. (2024b) shows that modular and synergistic approaches can also improve the quality of generated responses. LLM hallucinations on tabular data can be mitigated with modular approaches for generating faithful and interpretable answers, e.g., conditioning answers on a QA-based plan of sub-questions with extracted relevant table data. The final answer can be selected from candidate answers generated by both pre-trained table QA and text-to-SQL models (Zhang et al., 2023; Chemmengath et al., 2021), which predict SQL queries to represent the questions before executing them on tables to find the answers (Zhang et al., 2024b). The semantic parsing and question decomposition methods help generate SQL queries based on the table schema and questions (Eyal et al., 2023; Lin et al., 2020).

Table QA Datasets and Evaluation Most benchmark datasets for table reasoning tasks are mainly based on factual questions with short-form answers (Kweon et al., 2023; Li et al., 2023; Chen et al., 2021, 2020; Parikh et al., 2020; Yu et al., 2018; Novikova et al., 2017; Pasupat and Liang, 2015). A common metric for evaluating models across all related tasks is the exact match accuracy and ROUGE-L (Lin, 2004). General benchmark datasets representing human interactions, reasoning about structured knowledge retrieval, and longer free-form responses are currently unavailable for table QA (Nan et al., 2022).

3 DataBench

The dataset in this shared task is from the *DataBench* Osés Grijalba et al. (2025)[2], a comprehensive collection of tabular data designed for evaluating question answering over tables in English. *DataBench* consists of 65 datasets covering five domains of business, health, social, sports, and travel, with varying numbers of rows and columns and varied data types.

Table 1 provides an overview of the number of datasets collected, as detailed by Osés Grijalba et al. (2025), along with information on the rows and columns they contain. The corpus includes 65 real-world datasets with 3,269,975 rows and 1,615 columns, designed to evaluate language models on the task of QA over tabular data. It includes a total of 1,300 questions, each paired with a gold-standard answer, along with additional metadata such as answer type (i.e., true/false, categorical values from the dataset, numerical values, or lists), the corresponding data columns and their types.

Expected answer types are:

- Boolean. The answer can be True/False, Y/N, or Yes/No.
- Category. The answer contains a value from one cell or part of a cell.

[2] https://huggingface.co/datasets/cardiffnlp/databench

Domain	Datasets	Rows	Columns
Business	26	1,156,538	534
Health	7	98,032	123
Social	16	1,189,476	508
Sports	6	398,778	177
Travel	10	427,151	273
Total	65	3,269,975	1615

Table 1: *DataBench* domain taxonomy from Osés Grijalba et al. (2025).

- Number. The answer is a numerical value from one or multiple data table cells, which can also represent statistics (e.g., average, maximum and minimum).
- List[category]. Multiple values from one or more table cells are listed as the answer.
- List[number]. A list of numbers as the answer.

4 Task Description and Setup

For each tuple of a question and the relevant table, we must answer the question in two different settings:

1. Task A (*DataBench*): questions are answered with a given dataset; and,
2. Task B (*DataBenchLite*): questions are answered with a sampled version of the given dataset containing a maximum of 20 rows.

During the development phase, the training and development set of the data tables is made available for training or fine-tuning. The testing phase has only the test set of the *DataBench*.

5 Methods

Following Osés Grijalba et al. (2025), we explore three symbolic reasoning approaches through code execution to tabular reasoning for QA using LLMs—DCP$_{Py}$, a direct prompting approach to generate executable Python code and MSR$_{Py+FS}$, an agentic multi-step few-shot learning approach using Python tools, and MSC$_{SQL+FS}$, direct prompting approach for generating SQL statements. In this section, we detail our methodologies along with their implementation details.

5.1 DCP$_{Py}$

DCP$_{Py}$ uses direct code prompting (Figure 1 in appendix) to generate executable Python query code. This code is executed to obtain the relevant information from the corresponding table for post-processing. Specifically, (1) we directly prompted GPT-4o (2024-10-21) to generate executable Python code given the table columns, table column types, and the question, (2) the code is then executed returning the raw results from the table, and (3) this result is converted to the required format, e.g., changed 'yes' or 'no' answers to boolean types, changed the categories to a list of categories, etc., based on the expected answer types.

5.2 MSR$_{Py+FS}$

MSR$_{Py+FS}$ uses a ReAct (Yao et al., 2023) prompt (See Appendix, Figure 2) containing *DataBenchLite* as a sample table, and few-shot exemplars from the training set of *DataBench* for multi-step reasoning. Specifically, we (1) prompted Claude Sonnet 3.5 v2 (2024-10-22) with the question, sample table and few-shot exemplars to generate Python code to execute and the reasoning behind the code; (2) we then executed the code inside an isolated environment that contained the entire table; and then, (3) passed the result back to the model for observation. After observation, the model decides whether to generate the final answer or continue from step 1.

5.3 MSC$_{SQL+FS}$

MSC$_{SQL+FS}$ uses a multi-step prompt with few-shot learning by: (1) generating an SQL query statement—adding the list of table columns in the prompt; (2) Executing the generated query—looping until the LLM retrieves at least one record; and, (3) then, prompting the LLM to generate answer—prompt contains the question, SQL query, retrieved rows and few-shot exemplars from *DataBenchLite*. On the dev set, we experiment using Gemma2-9B and GPT4o-mini models, and on the test set, in our final submission, we submit the predictions obtained using the Gemma2-9B model.

6 Results and Discussions

The obtained accuracy scores and final rankings are presented in Table 2. Overall, the method with the highest performance is MSR$_{Py+FS}$ with an accuracy score of 88.12 % on *DataBench* and 87.70 % on *DataBenchLite*. DCP$_{Py}$ is the second and

	Task A		Task B	
	DataBench	Rank	*DataBenchLite*	Rank
DCP_{Py}	80.46	19	76.05	24
MSR_{Py+FS}	88.12	5	88.70	3
MSC_{SQL+FS}	64.94	36	69.16	30

Table 2: Final competition exact match accuracy scores and ranking.

MSC_{SQL+FS} is the lowest-performing ones. All methods, except the MSC_{SQL+FS} achieved better performance in some cases with the *DataBench* data compared to its smaller *DataBenchLite* subset.

Compared to the leaderboard, our highest-performing model, MSR_{Py+FS} achieves competitive results compared to the top scorer (Team TeleAI) of 95.1% on *DataBench* and 92.91% on *DataBenchLite*. The best-performing proprietary model (Team AILS-NTUA) has an accuracy of 89.85% and 88.89% on *DataBench* and *DataBenchLite*, respectively.

6.1 Error Analysis: Validation Set

The errors in our best-performing method, MSR_{Py+FS}, are analyzed on 320 question-answer pairs from the validation set. Differences between ground truth and predicted values are evaluated using the exact match accuracy metric with boolean outputs of the basic evaluation function for the *DataBench* corpus.[3]

Tabular question answering challenges. The main reasoning challenges in tabular QA (Table 3) are understanding and reasoning over the table data, which in our case is to understand the columns and their data types and enumerate over multiple columns to produce an answer that may have a list of strings or numbers.

Ground truth data quality. The ground truth answers are manually verified by inspecting the questions and executing the required queries on the validation data. At least 10% of the ground truth answers and questions are poorly defined, making an automatic evaluation of the models less objective and more challenging.

Poorly defined questions. The questions do not specify how to deal with the possibility of multiple or repeated values in the answers. Instructions for dealing with data quality, such as duplicated and empty values in datasets, are also missing in the questions.

[3]eval code: https://tinyurl.com/3wcxfvbm

Challenge — table schema understanding
Example — Recognizing the table schema and the data types of the table **Question —**"Did any respondent indicate that they will not vote?" requires models to identify a single 'Vote Intention' column and understand its list[category] type and 'I will not vote' textual values to correctly respond with a Boolean value
Challenge — question & answer type understanding
Example — Defining unambiguous instructions **Question —** "What are the three least commonly ordered quantities?" offers multiple interpretations to include or exclude the rows with returned purchases based on the meaning of invoices with negative quantities
Challenge — multi-column integration
Example — Enumerating over number lists in columns **Question —** "Which 5 patents (by ID) have the most targets associated?" requires models to identify two columns ('id' and 'target'), understand a list[number] column type, and count the items in number list for each ID to correctly respond with a list[number] value
Challenge — numeric answer generation
Example — Defining numeric precision and list order **Question —** "What are the highest 5 levels of Extraversion?" lacks definitions of answers with specified numerical precision and sorting of numbers in a list

Table 3: Examples of model reasoning and benchmarking challenges in tabular QA.

Unsuitable generic evaluation metric. Our findings indicate that questions about tables in benchmark datasets must be defined following the underlying table data and the desired evaluation methods. For example, the exact match metric leaves almost no room for interpretation of questions and requires an explicit definition of answers. The quality and structure of data need to be considered and disclosed to ensure models can arrive at the same ground truth answers.

6.2 Tabular Reasoning: Test Set

The tabular reasoning capabilities of our methods were analyzed with the 522 pairs of questions and answers in the test set of the *DataBench* data. Our focus was on understanding the effects of table sizes and answer types on the model performance under a few-shot in-context learning setting.

Different answer types. The type of answers (for example, *boolean, list[number] and list[category]*) influences the performance of the models, as shown in Table 4. The most accurate LLM predictions were for *boolean* answers, with 90.07% on *DataBench* and 93.02% on *DataBenchLite*. In contrast, the least accurate answers are generated for

	DCP_{Py}		MSR_{Py+FS}		MSC_{SQL+FS}	
	DB	DBL	DB	DBL	DB	DBL
Boolean	86.05	87.60	**90.07**	**93.02**	66.67	71.32
Category	85.14	82.43	**87.84**	**89.19**	74.42	81.08
Number	79.49	75.00	**85.26**	**87.82**	59.62	67.95
List[cat.]	69.94	68.06	**77.78**	**83.33**	51.39	50.00
List[num.]	58.24	56.04	**76.92**	**81.32**	64.84	68.13

Table 4: Accuracy per answer type in *DataBench* (DB) and *DataBenchLite* (DBL) test data.

	$\text{DB}_{Largest}$	$\text{DB}_{Smallest}$
Overall	88.24	90.24
Boolean	87.50	100.00
Category	100.00	100.00
Number	85.71	92.86
List[category]	100.00	85.71
List[number]	66.67	71.43

Table 5: Accuracy of MSR_{Py+FS} further split into the answer types for the largest and smallest table of *DataBench* test data.

number-lists with 76.92% and 81.32% accuracy, respectively, on *DataBench* and *DataBenchLite*.

LLMs struggle when generating list-type responses when tested for exact match accuracy, as the models might not extract all the items in the specific order. With *number-lists* being harder to produce than *category-lists*, this outcome resonates with our error analysis findings. Unlike the list of categories, each item in a list of numbers might require further aggregation (operations such as sum and average) after retrieval. However, for MSC_{SQL+FS}, the accuracy of the *number-lists* (68.13% on *DataBenchLite*) is higher than *category-list* (50% on *DataBenchLite*), as the intermediate SQL query generated by the LLMs already applies the aggregation operator.

Effect of data size. In this analysis, we only consider the question-answer pairs over the largest and the smallest table from *DataBench*. In the test set, the largest table is *068_WorldBank_Awards* with 4,789,220 cells in 20 columns and 239,461 rows (88.24% accuracy), and the smallest table is the 080_Books table with 520 cells in 13 columns and 40 rows (90.24% accuracy). Using the MSR_{Py+FS} method, we observe, as illustrated in Table 5, the size of the tables does not significantly influence the LLM predictions and the overall performance is slightly better on the smallest table (90.24%) than the largest (88.24%), showing the method's robustness to increases in the data size. As before, the answers with a number and list of numbers are challenging answer types for the model, more pronounced for the largest table.

We observed that when the number of columns increased, QA performed better, despite it adding to the complexity of column-wide reasoning. The accuracy when using the table with the least number of columns (Table: 074_Lift with 5 columns and 3,000 rows) and the table with the most columns (Table: 066_IBM_HR with 35 columns and 1,470 rows) is 74.29% and 94.87%, respectively.

Multi-column questions. On a manually sampled 202 pairs of questions and answers (38.70% of the *DataBench* test set), where the models need more than one column to produce an answer, we found the accuracy of our best method MSR_{Py+FS} is 80.20% with an 8% drop compared to the overall accuracy of 88.12% (Table 2). In contrast, for the rest of the records requiring single columns to answer, the accuracy is 87.19 %, which closely aligns with the overall accuracy, implying the model's difficulty in understanding and reasoning for multi-column answers. Examples of sampled multi-column QA are in the app. Table 6.

7 Conclusions

Table question answering is challenging because of how information can be organised in tables, with relevant information being located in columns from diverse data types, requiring integration of information across columns. We proposed three different methods for Table QA, with our best model, MSR_{Py+FS}, which uses Reasoning and Acting (ReAct) prompting, led to our team ranking in the top-5 among over 100 submissions in the shared task. Through our analysis, we found that numbers and list answer types, and questions requiring answers over multiple columns of the table are the most challenging factors for table QA.

Future research may focus on advanced prompting strategies, e.g., chain-of-thought or building answer-type specific pipelines.

Limitations

In our work, we only consider few-shot prompting without fine-tuning the models on the downstream QA tasks to address some reasoning problems using the available training data in the development phase. We did not do any data cleaning or pre-processing of the tables considering real-world conditions where the data tables might contain erroneous or missing data, which could improve the

overall performance. Model bias was also found as LLMs refused to answer some questions, including questions about pregnant people on table data without a specified 'female' gender, which requires further investigation. Ensembling our results from the three approaches could also improve our outcome. Furthermore, there is always the scope of using more advanced few-shot selection methods or stronger models such as `Llama-405b`, `Deepseek R1`, `OpenAI O3`, or `O1` models.

References

Jayetri Bardhan, Anthony Colas, Kirk Roberts, and Daisy Zhe Wang. 2022. DrugEHRQA: A question answering dataset on structured and unstructured electronic health records for medicine related queries. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1083–1097, Marseille, France. European Language Resources Association.

Jayetri Bardhan, Kirk Roberts, and Daisy Zhe Wang. 2024. Question answering for electronic health records: Scoping review of datasets and models. *Journal of Medical Internet Research*, 26:e53636.

Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Jaydeep Sen, Mustafa Canim, Soumen Chakrabarti, Alfio Gliozzo, and Karthik Sankaranarayanan. 2021. Topic transferable table question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4159–4172, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Si-An Chen, Lesly Miculicich, Julian Martin Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. 2024. TableRAG: Million-token table understanding with language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 74899–74921. Curran Associates, Inc.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 407–426, Bangkok, Thailand. Association for Computational Linguistics.

Haoyu Dong, Jianbo Zhao, Yuzhang Tian, Junyu Xiong, Mengyu Zhou, Yun Lin, José Cambronero, Yeye He, Shi Han, and Dongmei Zhang. 2024. Encoding spreadsheets for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20728–20748, Miami, Florida, USA. Association for Computational Linguistics.

Ben Eyal, Moran Mahabi, Ophir Haroche, Amir Bachar, and Michael Elhadad. 2023. Semantic decomposition of question and SQL for text-to-SQL parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13629–13645, Singapore. Association for Computational Linguistics.

Akash Ghosh, Venkata Sahith Bathini, Niloy Ganguly, Pawan Goyal, and Mayank Singh. 2024. How robust are the QA models for hybrid scientific tabular data? a study using customized dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8258–8264, Torino, Italia. ELRA and ICCL.

Michael Glass, Mustafa Canim, Alfio Gliozzo, Saneem Chemmengath, Vishwajeet Kumar, Rishav Chakravarti, Avi Sil, Feifei Pan, Samarth Bharadwaj, and Nicolas Rodolfo Fauceglia. 2021. Capturing row and column semantics in transformer based question answering over tables. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1212–1224, Online. Association for Computational Linguistics.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question

answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. 2024. TARGET: Benchmarking table retrieval for generative tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Alistair Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei Lehman, Leo Celi, and Roger Mark. 2023. MIMIC-IV, a freely accessible electronic health record dataset. *Nature, Scientific Data*, 10:1.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2022. AIT-QA: Question answering dataset over complex tables in the airline industry. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 305–314, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Buse Sibel Korkmaz and Antonio Del Rio Chanona. 2024. Integrating table representations into large language models for improved scholarly document comprehension. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 293–306, Bangkok, Thailand. Association for Computational Linguistics.

Sunjun Kweon, Yeonsu Kwon, Seonhee Cho, Yohan Jo, and Edward Choi. 2023. Open-WikiTable : Dataset for open domain question answering with complex reasoning over table. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8285–8297, Toronto, Canada. Association for Computational Linguistics.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Ma Chenhao, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2023. Can LLM already serve as a database interface? A BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, pages 42330–42357. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xi Victoria Lin, Richard Socher, and Caiming Xiong. 2020. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4870–4888, Online. Association for Computational Linguistics.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: table pre-training via learning a neural SQL executor. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Nafise Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. SciGen: a dataset for reasoning-aware text generation from scientific tables. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Md Mahadi Hasan Nahid and Davood Rafiei. 2024. TabSQLify: Enhancing reasoning capabilities of LLMs through table decomposition. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5725–5737, Mexico City, Mexico. Association for Computational Linguistics.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Rungsiman Nararatwong, Chung-Chi Chen, Natthawut Kertkeidkachorn, Hiroya Takamura, and Ryutaro Ichise. 2024. DBQR-QA: A question answering dataset on a hybrid of database querying and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15169–15182, Bangkok, Thailand. Association for Computational Linguistics.

Rungsiman Nararatwong, Natthawut Kertkeidkachorn, Hiroya Takamura, and Ryutaro Ichise. 2025. Fin-

DBQA shared-task: Database querying and reasoning. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 385–391, Abu Dhabi, UAE. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Jorge Osés Grijalba, L. Alfonso Ureña-López, Eugenio Martínez Cámara, and Jose Camacho-Collados. 2025. SemEval-2025 Task 8: Question Answering over Tabular Data. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1015–1022, Vienna, Austria. Association for Computational Linguistics.

Vaishali Pal, Evangelos Kanoulas, Andrew Yates, and Maarten de Rijke. 2024. Table question answering for low-resourced Indic languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 75–92, Miami, Florida, USA. Association for Computational Linguistics.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. 2023. MultiTabQA: Generating tabular answers for multi-table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6322–6334, Toronto, Canada. Association for Computational Linguistics.

Simone Papicchio, Paolo Papotti, and Luca Cagliero. 2023. QATCH: Benchmarking SQL-centric tasks with table representation learning models on your data. In *Advances in Neural Information Processing Systems*, volume 36, pages 30898–30917. Curran Associates, Inc.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. 2021. Knowledge graph-based question answering with electronic health records. In *Proceedings of the 6th Machine Learning for Healthcare Conference*, volume 149 of *Proceedings of Machine Learning Research*, pages 36–53.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. SPIQA: A dataset for multimodal question answering on scientific papers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118807–118833. Curran Associates, Inc.

Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. OmniParser: A unified framework for text spotting, key information extraction and table recognition. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15641–15653.

Jian Wu, Yicheng Xu, Yan Gao, Jian-Guang Lou, Börje Karlsson, and Manabu Okumura. 2023. TACR: A table alignment-based cell selection method for HybridQA. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6535–6549, Toronto, Canada. Association for Computational Linguistics.

Xueqing Wu, Rui Zheng, Jingzhen Sha, Te-Lin Wu, Hanyu Zhou, Mohan Tang, Kai-Wei Chang, Nanyun Peng, and Haoran Huang. 2024. DACO: Towards application-driven and comprehensive data analysis via code generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 90661–90682. Curran Associates, Inc.

Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. 2022. TableFormer: Robust transformer modeling for table-text encoding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018*

Conference on Empirical Methods in Natural Language Processing, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Haowei Zhang, Shengyun Si, Yilun Zhao, Lujing Xie, Zhijian Xu, Lyuhao Chen, Linyong Nan, Pengcheng Wang, Xiangru Tang, and Arman Cohan. 2024a. OpenT2T: An open-source toolkit for table-to-text generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 259–269, Miami, Florida, USA. Association for Computational Linguistics.

Siyue Zhang, Anh Tuan Luu, and Chen Zhao. 2024b. SynTQA: Synergistic table-based question answering via mixture of text-to-SQL and E2E TQA. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2352–2364, Miami, Florida, USA. Association for Computational Linguistics.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024c. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M. Patel. 2023. Reactable: Enhancing react for table question answering. *CoRR*, abs/2310.00815.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Binghong Wu, Lei Liao, Shu Wei, Yongjie Ye, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. 2024a. TabPedia: Towards comprehensive visual table understanding with concept synergy. In *Advances in Neural Information Processing Systems*, volume 37, pages 7185–7212. Curran Associates, Inc.

Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. 2024b. TaPERA: Enhancing faithfulness and interpretability in long-form table QA by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, Bangkok, Thailand. Association for Computational Linguistics.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Yilun Zhao, Boyu Mi, Zhenting Qi, Linyong Nan, Minghao Guo, Arman Cohan, and Dragomir Radev. 2023a. OpenRT: An open-source framework for reasoning over tabular data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 336–347, Toronto, Canada. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, Dragomir Radev, and Arman Cohan. 2023b. QTSumm: Query-focused summarization over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1172, Singapore. Association for Computational Linguistics.

Yilun Zhao, Chen Zhao, Linyong Nan, Zhenting Qi, Wenlin Zhang, Xiangru Tang, Boyu Mi, and Dragomir Radev. 2023c. RobuT: A systematic study of table QA robustness against human-annotated adversarial perturbations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6064–6081, Toronto, Canada. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Appendix

A Prompting Approaches

The prompts used for DCP$_{Py}$, MSR$_{Py+FS}$, and MSC$_{SQL+FS}$ are shown in Figures 1, 2, and 3, respectively.

B Additional Analysis

Table 6 shows question and answer pairs from *DataBench* that we sample for the evaluation of answering questions that require information from multiple columns and illustrate the differences between the ground truth and the predictions of the MSR$_{Py+FS}$ method.

Question	Required Columns	Table	Ground Truth	Prediction
Which rating given to the stated purpose of students is associated with the highest accumulated grade point average?	University Rating (uint8) CGPA (float64)	072_Admissions	4.5	5
What is the single label associated with the most products? Answer with a single category.	labels_en (object) product_name (object)	070_OpenFoodFacts	'No gluten'	'Green Dot'
What cause corresponds to the lowest mortality rate?	Cause (category) Rate (float64)	075_Mortality	'Suicide'	'Nephritis'
What is the name of the windiest day on average?	wind (float64) day (uint8) calendar_names_2 (object)	078_Fires	'Monday'	'March'
What is the product type of the transactions with yielded the most money in revenue? Answer with a category.	product_type (category) Revenue (category)	079_Coffee	'Premium Beans'	'Barista Espresso'
Is any entry in the third tier a (direct or otherwise) descendant of 150?	Tier 3 (category) Parent (category)	069_Taxonomy	TRUE	FALSE
Is Barbados considered overall more expensive than the country ranked in the 10th place?	Country (category) Rank (uint8) Local Purchasing Power Index (float64)	071_COL	TRUE	FALSE
What are the top 5 total lifts by Weight Class?	Amount Lifted (kg) (uint16) Weight Class (category)	074_Lift	[88849, 88071, 87862, 83245, 81271]	['93 kg', 'Open', '59 kg', '83 kg', '52 kg']
List the weights of women with a height of exactly 1m and 45cm.	Weight (uint8) Height (uint8)	077_Gestational	[49.0, 50.0, 55.0, 66.0, 61.0]	[49, 50, 55, 66, 61, 71, 95, 55, 67, 69, 89, 55, 90, 58, 61, 78, 80]
List the ratings of the top four books with the most reviews?	Ratings (float64) Reviews (float64)	080_Books	[73.0, 85.0, 50.0, 30.0]	[30, 39, 27, 25]
List the 5 players with the least games played.	PLAYER (category) GP (uint8)	076_NBA	['Harrison Barnes', 'DeMar DeRozan', 'Jeff Green', 'P.J. Tucker', 'Andre Drummond']	['Quentin Richardson', 'Andrew Goudelock', 'Darko Milicic', 'Matt Carroll', 'Vladimir Radmanovic']

Table 6: Examples of multi-column questions based on manual inspection of table schema in the test data. The examples show the prediction errors from MSR_{Py+FS} and the corresponding ground truth.

DCP$_{Py}$ Prompt

Convert the given question to executable Python code based on the table columns and table column types. The dataframe is already given and the Python code should print out the answer only.

Table columns are: {list of table columns}
For each column types are: {list of table type}
Question: {question}
Python code:

Figure 1: DCP$_{Py}$ prompt.

MSR$_{Py+FS}$ Prompt

You are a helpful AI assistant that can execute Python code to analyse tables.
The user will ask a question that is related to a markdown table of which you are given a sample of.
The user's question can pertain to one or more rows within the table, so ensure you take this into account. You will be given a set of example questions and answers.
Only provide the answer in the form of a boolean, list[category or number], category or number and do not write anything else. Do not write anything aside from the direct answer.
You must use the Python tool to get your answer, and you must use Python's builtin print in order to see your results.
Assume that the full table is located in a parquet file at /sandbox/all.parquet

###Example Questions and Answers
{{Few-shot examples}}

###SAMPLE TABLE
{{markdown_sample_table}}

Figure 2: MSR$_{Py+FS}$ prompt.

MSC$_{SQL+FS}$ Prompt

QUESTION: {input}

You have to generate an answer to the above question from a table. The SQL query below is executed on the table: {query}

Output of the SQL query: {result}

Based on the above SQL query output, generate an appropriate answer to the given question. Check if the answer is below rules:

Rule #1: If a question that starts with 'are there..', 'is there..', the answer would be whether the records exist, in such cases, return True or false.
Rule #2: If the question is asking about a count of items or maximum, minimum or average of the items, apply that aggregation (count, min, max, average) on the items and return a numeric value, not a list of items. For example, "How many distinct HHS regions are present ?" is asking about the total number of regions and the answer is 10.
Rule #3: Avoid repetition in the answer.
Rule #4: Always put multiple values in the answer within a [] bracket. If the list items are strings, add single quotations around the strings.

EXAMPLES:
{sampleqa}

Your response should only contain the question's answer.

RESPONSE:

Figure 3: MSC$_{SQL+FS}$ prompt.