SQALE: A Large Text-to-SQL Corpus Grounded in Real Schemas

Cornelius Wolff[1,2] **Daniel Gomm**[1,2] **Madelon Hulsebos**[1]
[1]Centrum Wiskunde & Informatica [2]University of Amsterdam
{cornelius.wolff, daniel.gomm, madelon.hulsebos}@cwi.nl

Abstract

Advances in large language models have accelerated progress in text-to-SQL, methods for converting natural language queries into valid SQL queries. A key bottleneck for developing generalizable text-to-SQL models is the lack of large-scale datasets with sufficient schema and query complexity, domain coverage, and task diversity. We introduce SQALE, a large-scale semi-synthetic text-to-SQL dataset built on 135,875 relational database schemas expanded from a collection of real-world schemas, SchemaPile. We establish a principled generation pipeline which combines schema sampling, question synthesis, and SQL construction, and produce 517,676 high-quality (question, schema, query) triples. The SQALE dataset captures realistic schema size variability, diverse query patterns, and natural language ambiguity while maintaining execution validity. We provide an analysis of its contents and characteristics, and find that SQALE introduces the most realistic large-scale text-to-SQL dataset to date in comparison with existing benchmarks and datasets. We discuss how SQALE enables our vision for data scaling and model generalization in text-to-SQL research. The dataset is accessible at: https://huggingface.co/datasets/trl-lab/SQaLe-text-to-SQL-dataset.

1 Introduction

When it comes to the retrieval of insight from large document stores, large language models (LLMs) and retrieval-augmented generation (RAG) have significantly advanced natural language understanding and question answering by reducing bottlenecks in unstructured document retrieval [20, 11]. However, when it comes to structured data, the development of dedicated models for accessing structured data through text-to-SQL remains constrained by the limited availability of large, diverse, and high-quality datasets [23]. Although LLMs have enabled rapid progress in text-to-SQL generation, building new and generalizable models from scratch still faces major challenges due to insufficient data scale and schema diversity [10]. This problem is fundamental to enabling natural language interfaces for databases.

Over the past years, the field has evolved through successive generations of datasets that attempt to balance realism, schema complexity, and linguistic variety. The first common benchmark was *Spider 1.0* [32], which introduced cross-database generalization and multi-table reasoning, containing around 10,000 questions across hundreds of databases. It quickly became the de facto standard for Text-to-SQL evaluation. It was later largely replaced by *BIRD* [22], which placed greater emphasis on realism by incorporating practical database contexts. Other corpora such as *KaggleDBQA* [17] contributed similar aims, while domain-specific datasets including *EHRSQL* [18] and others [29] targeted specialized applications. More recently, *Spider 2.0* [19] has extended this line of work to enterprise settings, featuring more diverse schemas and queries, and is rapidly becoming a new common benchmark. Despite these advances, the overall scale of available resources remains orders of magnitude smaller than standard NLP corpora [24], making them insufficient under established scaling laws for training large models [13, 12]. While synthetic and semi-synthetic approaches have begun to address these gaps, only a few datasets, such as *SynSQL-2.5M* [21], have reached million-scale coverage, relying primarily on artificial schemas and programmatic question generation.

AI for Tabular Data workshop at EurIPS 2025

Yet, the used schemas often lack the scale, complexity, and other real-world characteristics needed to fully capture the challenges of practical Text-to-SQL tasks [10, 31, 23].

To address the limitations of scale, diversity, and realism, we construct a large-scale Text-to-SQL dataset with broad schema coverage and naturally phrased questions. The SQALE dataset contains 135,875 relational database schemas extracted from SchemaPile's 22,989 real-world database schemas [8] and contains 517,676 (question, schema, query) triples. Unlike fully synthetic resources, our generation pipeline is grounded in real schemas and guided by principled linguistic and structural criteria to ensure both realism and diversity in queries, resulting in distinctive characteristics such as . We also provide a descriptive analysis of SQALE in comparison with existing datasets, and discuss its implications for training generalizable text-to-SQL models. Our contributions are as follows: **(1)** a data generation pipeline for synthesizing text-to-SQL data from real schemas; **(2)** the creation of a large-scale text-to-SQL dataset; and **(3)** an in-depth characterization of the dataset, highlighting its scalability potential and relevance for model training and evaluation.

2 Creating Text-to-SQL Data at Scale

Here, we describe our methodology for producing SQALE: a principled, scalable pipeline that synthesizes representative (schema, question, query) triples corresponding with real database schemas.

2.1 Dataset Design Criteria

The design of large-scale text-to-SQL datasets should follow principles that ensure both representational realism and linguistic-semantic diversity. We distinguish between schema-level and query-level considerations. These criteria are grounded in empirical analyses of production databases and informed by benchmarks such as Spider 2.0 [19], BIRD [22], and SQLStorm [27]. All of these criteria were furthermore confirmed through expert interviews with data engineers and practitioners.

2.1.1 Schema-Level Criteria

Schema Size (C1). A realistic dataset should include schemas ranging from small, domain-specific databases to large, enterprise-scale systems containing hundreds of tables. This variety reflects the scale of modern data ecosystems observed in industry and academic corpora [4, 5, 19].

Schema Density and Normalization (C2). Text-to-SQL datasets should capture variation in schema density, the average number of columns per table, as a proxy for normalization depth and granularity, as observed in real-world databases [8, 2]. Furthermore, including both highly normalized and denormalized schemas reflects different relational schema principles[15].

Foreign Key Integrity (C3). Datasets should represent realistic variability in referential integrity. Many real-world databases contain incomplete or implicit foreign key relationships; including such cases ensures that models are exposed to typical challenges of schema reasoning and foreign-key inference. This aspect of real-world databases has been a research topic for years and therefore should be reflected in the dataset [26, 33, 14].

Naming Conventions (C4). Schema naming should reflect authentic, heterogeneous practices, including inconsistent abbreviations, mixed casing, and domain specific terms. Preserving this variability enhances validity and prevents overfitting to overly sanitized or idealized schemas [8, 19].

2.1.2 Query-Level Criteria

Join Complexity (C5). A well-designed text-to-SQL dataset should span a distribution of SQL join complexities, from single-table to multi-join analytical queries. This range allows for balanced evaluation of compositional reasoning and structural generalization [19, 27, 3].

Operator Diversity (C6). Queries should include diverse SQL constructs such as aggregations, comparisons, nested subqueries, set operations, and logical combinations to represent the breadth of real analytical workloads and support model training on varied syntactic and semantic forms [23, 16].

Intent Diversity (C7). Natural language inputs should encompass a wide spectrum of user intents, such as lookup, aggregation, filtering, comparison, and ranking. Coverage of distinct intent categories enables more comprehensive assessment of a model's semantic understanding capabilities [7, 16].

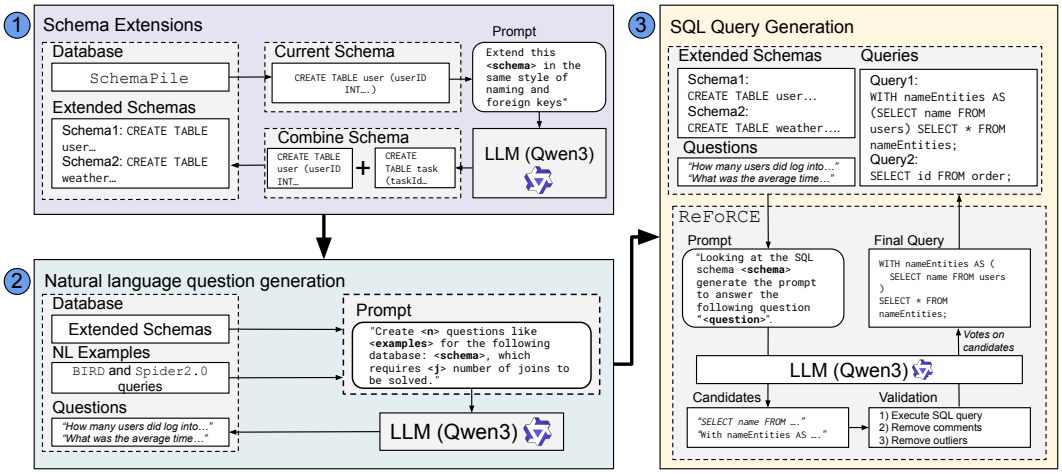

Figure 1: Overview of the data creation pipeline for the SQALE dataset. The displayed prompts are simplified for clarity.

Query Ambiguity (C8). High-quality datasets should contain controlled linguistic and semantic ambiguity, reflecting the uncertainty that stems from a user's incomplete understanding of the underlying data, such as unfamiliarity with the schema or the specific values of interest [1, 9, 28].

2.2 The SQALE Generation Pipeline

The SQALE generation pipeline (Figure 1) follows a structured, multi-stage approach that integrates schema extension, question synthesis, and SQL generation. It begins with a repository of real-world database schemas, SchemaPile [8], which serves as the foundation for realistic structural diversity (**C2**). Using a large language model (Qwen3 30B [30]), existing schemas are **extended incrementally**, for example by including 5 to 25 synthetic tables, until the target query complexity distribution is balanced around a median of 4 joins per query, informed by SQLStorm metrics [27] (**C1**). The targeted schema size, in number of tables, follows a smooth gamma-like probability distribution, where the mode is set to 100 and the maximum value to 350 based on the maximum in Spider 2.0 [19]. During the extension process, the LLM is prompted to maintain style, naming conventions and level of normalization of the original schema from SchemaPile (**C2, C4**). All **intermediate schema sizes** generated during the iterative extension process are also retained in the dataset, ensuring balanced representation across schema scales. Furthermore, for all schemas inferred from schemas without foreign key relations, any foreign keys added during extension process are removed (**C3**).

In the next stages, the pipeline performs **question generation** and **SQL query creation** using the same LLM with guided prompting. For each extended schema, natural language questions are synthesized from examples drawn from BIRD and Spider2.0, ensuring linguistic diversity, while questions are selected in line with the targeted join distribution (**C6–C8**). We chose this open question generation over a completely template based approach like in [25] in order to maximize the variety of questions in terms of style, length and complexity. Example selection also balances easy and difficult joins to ensure diversity (**C5**). These questions are then paired with candidate SQL statements in a **query creation phase** using the ReFoRCE text-to-SQL framework [6], which generates multiple SQL candidates, applies a voting process to select the best one, and validates executability by executing the SQL statement against the schema and filter for outliers in terms of prompt and query length. Finally, we manually randomly checked 300 samples from the dataset to ensure quality and correctness.

3 The SQALE Dataset

Using the outlined generation pipeline, the SQALE dataset represents a large-scale, semi-synthetic Text-to-SQL resource grounded in real-world database structures. The generation pipeline, executed with up to 100x H100 GPUs in parallel, results in 517,676 validated (schema, question, query) triples.

Table 1: Schema statistics for datasets. M. stands for median.

Dataset	#Schemas	M. #cols/schema	M. #tables/schema	#FKs
BIRD (train & dev)	80	39	5.0	526
EHRSQL	2	92	13.5	34
Spider2	236	89	7.0	0
SynSQL	16,575	72	10.0	159,547
SQALE	135,875	435.0	91.0	13,201,052

Table 2: Query statistics for datasets. Where and Join columns indicate the share of queries using those operators and the Func column contains multiple operators[2]. M. indicates the median.

Dataset	#queries	M. #tables/query	M. #tokens/query	Func. (%)	Where (%)	Join (%)
BIRD (train & dev)	10,962	2	41	13.8	88.1	76.2
EHRSQL	9,270	2	83	76.9	99.9	19.7
Spider2	250	3	229.5	45.2	94.4	72.0
SynSQL	2,544,390	3	85	12.1	75.6	89.4
SQALE	517,676	3	61	26.9	78.9	76.1

Compared to existing datasets, SQALE offers a substantially broader and more realistic schema distribution (Table 1 and Figure 2). With 135,875 schemas, a median of 91 tables and 435 columns/schema, and over 13 million Foreign Keys (FK) relations, it far exceeds the structural diversity of Spider 2.0, SynSQL and BIRD. While not as large as SynSQL in terms of total number of queries, our dataset features vastly more and more complex schemas, emphasizing depth and structural realism over volume. In terms of query composition (Table 2 and Figure 2), SQALE shows rich operator diversity and join complexity, with 76.1% involving joins. These proportions closely match the complexity of BIRD and Spider 2.0 while extending further into multi-join and nested queries.

The scale of SQALE is deliberately chosen to support the development of new small-scale text-to-SQL models and fine-tuning of larger foundation models. Grounded in authentic schema patterns and validated through the rigorous ReFoRCE-based SQL validation process, the dataset balances realism and efficiency, providing high-quality data for building and adapting text-to-SQL models. It addresses current inefficiencies seen in leaderboards [19, 22], where large models and complex pipelines dominate, and even ReFoRCE requires 12 seconds on an H100 to generate a single SQL query. The dataset can be accessed at https://huggingface.co/datasets/trl-lab/SQaLe-text-to-SQL-dataset.

Toward scaling text-to-SQL with SQALE. Scaling laws in machine learning suggest that larger datasets yield larger models that will continue to perform better [13, 12]. Informed by task-specific scaling studies, we hypothesize that through significantly larger, more complex and diverse text-to-SQL datasets we can achieve proportional gains in model performance as a function of dataset scale. To explore the effectiveness of large-scale pretraining of text-to-SQL models, we introduce the text-to-SQL dataset SQALE consisting of 517,676 semi-synthetic (schema,query,SQL) triples grounded in real-world schemas and queries. A sample of the dataset can be seen in appendix A.

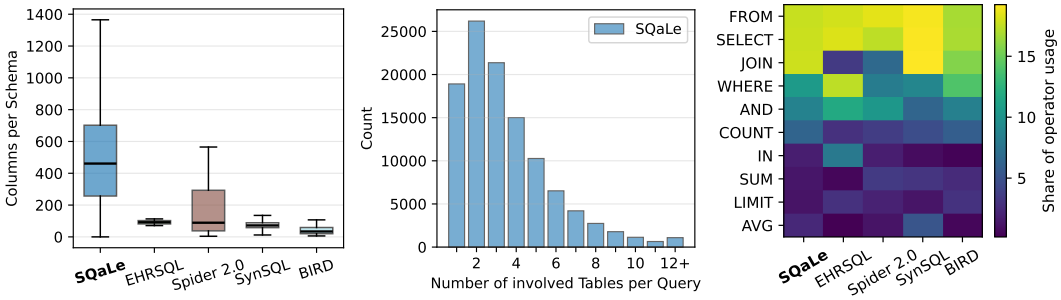

Figure 2: SQALE data statistics. Left: distribution of column counts. Center: Distribution of the number of tables involved in queries. Right: Share of specific operators among all operators.

[2]CAST, COALESCE, SUBSTR, LENGTH, UPPER, LOWER, ROUND, ABS, DATE, DATETIME, STRFTIME.

Acknowledgments

This work was partially funded by grants from NWO (NGF.1607.22.045) and SAP.

References

[1] Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. Benchmarking and Improving Text-to-SQL Generation under Ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7053–7074, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.436. URL https://aclanthology.org/2023.emnlp-main.436.

[2] Zouhaier Brahmia, Fabio Grandi, and Barbara Oliboni. A Literature Review on Schema Evolution in Databases. *Computing Open*, 02:2430001, January 2024. ISSN 2972-3701. doi: 10.1142/S2972370124300012. URL https://www.worldscientific.com/doi/10.1142/S2972370124300012.

[3] Peter Baile Chen, Fabian Wenz, Yi Zhang, Devin Yang, Justin Choi, Nesime Tatbul, Michael Cafarella, Çağatay Demiralp, and Michael Stonebraker. BEAVER: An Enterprise Benchmark for Text-to-SQL, January 2025. URL http://arxiv.org/abs/2409.02038. arXiv:2409.02038 [cs].

[4] Anthony Cleve, Maxime Gobert, Loup Meurice, Jerome Maes, and Jens Weber. Understanding database schema evolution: A case study. *Science of Computer Programming*, 97:113–121, January 2015. ISSN 01676423. doi: 10.1016/j.scico.2013.11.025. URL https://linkinghub.elsevier.com/retrieve/pii/S0167642313003092.

[5] Eli Cortez, Philip A. Bernstein, Yeye He, and Lev Novik. Annotating database schemas to help enterprise search. *Proceedings of the VLDB Endowment*, 8(12):1936–1939, August 2015. ISSN 2150-8097. doi: 10.14778/2824032.2824105. URL https://dl.acm.org/doi/10.14778/2824032.2824105.

[6] Minghang Deng, Ashwin Ramachandran, Canwen Xu, Lanxiang Hu, Zhewei Yao, Anupam Datta, and Hao Zhang. REFORCE: A TEXT-TO-SQL AGENT WITH SELF- REFINEMENT, FORMAT RESTRICTION, AND COLUMN EXPLORATION. 2025.

[7] Catalina Dragusin, Katsiaryna Mirylenka, Christoph Miksovic Czasch, Michael Glass, Nahuel Defosse, Paolo Scotton, and Thomas Gschwind. Grounding LLMs for Database Exploration: Intent Scoping and Paraphrasing for Robust NL2SQL.

[8] Till Döhmen, Radu Geacu, Madelon Hulsebos, and Sebastian Schelter. SchemaPile: A Large Collection of Relational Database Schemas. *Proceedings of the ACM on Management of Data*, 2(3):1–25, May 2024. ISSN 2836-6573. doi: 10.1145/3654975. URL https://dl.acm.org/doi/10.1145/3654975.

[9] Avrilia Floratou, Fotis Psallidas, Fuheng Zhao, Shaleen Deep, Gunther Hagleither, Joyce Cahoon, Rana Alotaibi, Jordan Henkel, Abhik Singla, Alex van Grootel, Kai Deng, Katherine Lin, Marcos Campos, Venkatesh Emani, Vivek Pandit, Wenjing Wang, and Carlo Curino. NL2SQL is a solved problem... Not! 2024.

[10] Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. *Proceedings of the VLDB Endowment*, 17(5):1132–1145, January 2024. ISSN 2150-8097. doi: 10.14778/3641204.3641221. URL https://dl.acm.org/doi/10.14778/3641204.3641221.

[11] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey, March 2024. URL http://arxiv.org/abs/2312.10997. arXiv:2312.10997 [cs].

[12] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, and Katie Millican. Training Compute-Optimal Large Language Models.

[13] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs].

[14] George Katsogiannis-Meimarakis, Katsiaryna Mirylenka, Paolo Scotton, Francesco Fusco, and Abdel Labbi. In-depth Analysis of LLM-based Schema Linking, 2026. URL https://openproceedings.org/2026/conf/edbt/paper-24.pdf.

[15] Ryosuke Kohita. Exploring Database Normalization Effects on SQL Generation, October 2025. URL http://arxiv.org/abs/2510.01989. arXiv:2510.01989 [cs].

[16] Wuwei Lan, Zhiguo Wang, Anuj Chauhan, Henghui Zhu, Alexander Li, Jiang Guo, Sheng Zhang, Chung-Wei Hang, Joseph Lilien, Yiqun Hu, Lin Pan, Mingwen Dong, Jun Wang, Jiarong Jiang, Stephen Ash, Vittorio Castelli, Patrick Ng, and Bing Xiang. UNITE: A Unified Benchmark for Text-to-SQL Evaluation, July 2023. URL http://arxiv.org/abs/2305.16265. arXiv:2305.16265 [cs].

[17] Chia-Hsuan Lee, Oleksandr Polozov, and Matthew Richardson. KaggleDBQA: Realistic Evaluation of Text-to-SQL Parsers. 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 2261–2273, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.176. URL https://aclanthology.org/2021.acl-long.176.

[18] Gyubok Lee, Hyeonji Hwang, Seongsu Bae, Yeonsu Kwon, Woncheol Shin, Seongjun Yang, Minjoon Seo, Jongyeup Kim, and Edward Choi. EHRSQL: A Practical Text-to-SQL Benchmark for Electronic Health Records.

[19] Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin SU, ZHAOQING SUO, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, et al. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. In *The Thirteenth International Conference on Learning Representations*, 2025.

[20] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, April 2021. URL http://arxiv.org/abs/2005.11401. arXiv:2005.11401 [cs].

[21] Haoyang Li, Shang Wu, Xiaokang Zhang, Xinmei Huang, Jing Zhang, Fuxin Jiang, Shuai Wang, Tieying Zhang, Jianjun Chen, Rui Shi, Hong Chen, and Cuiping Li. OmniSQL: Synthesizing High-quality Text-to-SQL Data at Scale, July 2025. URL http://arxiv.org/abs/2503.02240. arXiv:2503.02240 [cs].

[22] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM Already Serve as A Database Interface? A BIg Bench for Large-Scale Database Grounded Text-to-SQLs, November 2023. URL http://arxiv.org/abs/2305.03111. arXiv:2305.03111 [cs].

[23] Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuxin Zhang, Ju Fan, Guoliang Li, Nan Tang, and Yuyu Luo. A Survey of Text-to-SQL in the Era of LLMs: Where are we, and where are we going?, June 2025. URL http://arxiv.org/abs/2408.05109. arXiv:2408.05109 [cs].

[24] Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. Datasets for Large Language Models: A Comprehensive Survey, February 2024. URL http://arxiv.org/abs/2402.18041. arXiv:2402.18041 [cs].

[25] Simone Papicchio, Paolo Papotti, and Luca Cagliero. Qatch: Benchmarking sql-centric tasks with table representation learning models on your data. *Advances in Neural Information Processing Systems*, 36:30898–30917, 2023.

[26] Alexandra Rostin, Oliver Albrecht, Jana Bauckmann, Felix Naumann, and Ulf Leser. A Machine Learning Approach to Foreign Key Discovery.

[27] Tobias Schmidt, Viktor Leis, Peter Boncz, and Thomas Neumann. SQLStorm: Taking Database Benchmarking into the LLM Era. *Proceedings of the VLDB Endowment*, 18(11):4144–4157, July 2025. ISSN 2150-8097. doi: 10.14778/3749646.3749683. URL https://dl.acm.org/doi/10.14778/3749646.3749683.

[28] Bing Wang, Yan Gao, Zhoujun Li, and Jian-Guang Lou. Handling Ambiguous and Unanswerable Questions for Text-to-SQL.

[29] Ping Wang, Tian Shi, and Chandan K. Reddy. Text-to-SQL Generation for Question Answering on Electronic Medical Records. In *Proceedings of The Web Conference 2020*, pages 350–361, Taipei Taiwan, April 2020. ACM. ISBN 978-1-4503-7023-3. doi: 10.1145/3366423.3380120. URL https://dl.acm.org/doi/10.1145/3366423.3380120.

[30] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report, May 2025. URL http://arxiv.org/abs/2505.09388. arXiv:2505.09388 [cs].

[31] Jiaxi Yang, Binyuan Hui, Min Yang, Jian Yang, Junyang Lin, and Chang Zhou. Synthesizing Text-to-SQL Data from Weak and Strong LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7864–7875, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.425. URL https://aclanthology.org/2024.acl-long.425.

[32] 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. 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, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1425. URL http://aclweb.org/anthology/D18-1425.

[33] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc, and Divesh Srivastava. On multi-column foreign key discovery. *Proceedings of the VLDB Endowment*, 3 (1-2):805–814, September 2010. ISSN 2150-8097. doi: 10.14778/1920841.1920944. URL https://dl.acm.org/doi/10.14778/1920841.1920944.

A Dataset Sample

Below are two representative entries from the SQALE dataset illustrating the DDL of real schemas, the corresponding natural-language questions, the validated SQL queries, and concise metadata (token counts, join counts, involved tables, and columns) like it can be found on HuggingFace.

Schema (DDL)	Question	Query	Token Count	Joins	Tables	Cols
`CREATE TABLE courses (course_id TEXT, name TEXT, teacher_id TEXT...`	List all tasks with course names and task states.	`SELECT tasks.name, courses.name FROM tasks JOIN courses ON ...;`	{q:57, s:338, tot:505}	3	6	25
`CREATE TABLE employees (id INT, name TEXT, dept TEXT, salary ...`	Find total salary by department.	`SELECT dept, SUM(salary) FROM employees GROUP BY dept;`	{q:22, s:120, tot:142}	0	2	5

Table 3: Example entries illustrating the structure of the SQL generation dataset.