

BRING TEXT-TO-SQL TO BI PRODUCTION IN LARGE ENTERPRISE

Big Data Platform Department | PCG **Tencent 腾讯** Jun 2024

TARGET AUDIENCE

The intended audience:

- data platform product managers, product strategists,
- data engineers, data scientists, and ML/AI practitioners

The audience will benefit:

- First-hand experience from Tencent's large-scale practice adapting the latest LLM to BI.
- Combining the latest open source and commercial LLM, RAG pattern/framework.
- Build the next-generation BI capabilities with a natural language interface.

ABSTRACT

1. Text-to-SQL capability is available with **fine-tuning** and **prompt engineering** under the hood.

2. Attempt to bridge the gap between POC Text-to-SQL tools and querying with natural language for users in Tencent.

3. We chose DeepSeekCoder-33B as the foundational model for fine-tuning using LORA.

4. We developed a training instruction set with two primary goals:

- 1. query pattern and syntax coverage.
- 2. representing how business context is referenced, especially for multi-table queries.
- 5. Optimizing the performance and cost-effectiveness of the inference process.

6. Our Model is evaluated and compared with GPT3.5/4 using BIRD and a set of complex real-life queries from Tencent.
7. The model performs better (i.e., more accurately) than GPT4 in cases requiring table joins, though not as capable of dealing with ambiguous expressions of query intentions.

CONTENT OUTLINE

[Why]-[How]-[Use Case]

1.[Why] Fine-tuning LLMs to improve Text-to-SQL

1.1 The Reason We Choose to Fine-tune?1.2 Benchmark Testing & Model Selection

2.[How] Fine-tuning Processes

2.1 Data Preparation2.2 Fine-tuning Process2.3 Benchmark Testing

3.[Use Case]

3.1 Tencent Video & NBA News3.2 RAG & Query Rewriting3.3 BI Demo

1.[WHY] FINE-TUNE LLMS TO IMPROVE TEXT-TO-SQL

1.1 THE REASON WHY TO FINE-TUNE

Data Privacy, Cost, and Performance

1. GPT4/3.5 performs well, but Tencent needs tighter control on data privacy and data security.

2. The response time of GPT4/3.5 is long, and **the APIs are not cheap**; we are looking for **cost-effective models that respond faster**.

3. The **training dataset** for the open source model is **relatively basic**. As for specific domain knowledge and business jargon/pattern, in-context learning does not always provide the best performance.

4. The SQL generation quality of open models and GPT are **not optimized** for **predicate push down** and **column pruning**, which could not generate high performance SQL clauses when querying big data.

1.1 THE REASON WHY TO FINE-TUNE

The Objective And Steps Of Fine-tuning

1. Objective

- Fine-tune a more general, vertical Text-to-SQL model that can serve multiple business use cases.
- To solve problems with high complexity in SQL and Query (hard/extra hard)
- Train users/business patterns in specific Data Lake / Data Warehouse, and achieve best practice in *memorization* and *generalization*.
- Learn general knowledge of Query-to-SQL and business knowledge/jargon through context-learning.
- Generate stable, correct, high-performing SQL clauses, which is quite different from chat models.

2. Steps to Fine-tune

- Selection of the Foundation Model
- Selection of Training Dataset
- Data Cleaning
- Instruct Tuning
- Benchmark Testing and Evaluation

1.2 BENCHMARK TEST & MODEL SELECTION Foundation Model Benchmark

We used **BIRD Dataset & Tencent Dataset** to evaluate model performance, focused on **Accuracy**. We chose higher scored models as our foundational model.

Model	Accuracy-BIRD Dataset(No Retrieval)	Accuracy-In Tencent Dataset
ChatGLM-6b	2.4%	4%
ChatGLM2-6b	2.4%	4%
Belle-13B-ext	0%	0%
StarCodeBase-15.5B	2.1%	2%
WizardCoder-15B	14.9%	10%
WizardCoder-34B	24.3%	18%
WizardCoder-33B	39.2%	32%
DeepSeekCoder-33B-instruct	35%	34%
SQLCoder-34B	17.4%	28%

1.2 BENCHMARK TEST & MODEL SELECTION

Choose Foundation Models With Good Performance On Code Generation



StarCodeBase-15.5B--> WizardCoder-15B--> WizardCoder-34B --> WizardCoder-33B --> DeepSeekCoder-33B-instruct --> SQLCoder-34B

- With the development of LLMs in 2023, various specialized models emerged--they are different from general base models because of the pretraining dataset and the instruction set.
- We compared the performance of general models (ChatGLM, Baichuan, Belle), which perform well in Chinese, and specialized open-sourced models (StarCoder, WizardCoder).
- Initially, we used Belle and StarCoder as our base model. We picked WizardCoder, DeepSeekCoder, and SQLCoder as our potential choices when they were available.

DATA⁺AI SUMMIT

2.[HOW] FINE-TUNING PROCESSES

Challenge: Open-source Dataset & Tencent Dataset

- The quality of training data is a crucial factor in fine-tuning.
- The common datasets from the open community (BIRD, Spider, Chase, DuSQL) are in English, which requires translation.
- Considering the dataset quality and actual usage from reality, we use Spider (Chinese) & Tencent datasets as training data.

•	The following compares the (CSpider (Spider	Chinese) dataset and tr	ie Tencent dataset.

Characteristic	Spider(Chinese) Dataset	Tencent Dataset
table fields	few	many
table size	small	large
table count	few	many
table quality	good	some are good, some are poor
calculation methods	few	many
SQL quality	some are good, some are poor	some are good, some are poor



DATA'AI SUMMIT

Prepare The Training Data - Part 1

1. Self-instruction: Align Training Data with Actual Data Distribution

The training sample needs to be comprehensive enough to cover SQL at various difficulty levels. Tencent samples and CSpider samples do not meet these requirements. To improve it, we randomly select manually written, high-quality instructions as seeds and format them into few-shot templates.

This allows LLMs to generate samples with specified features. The new samples generated by the LLM will be added to the sample pool, serving as an alternative set for the next round of model selection.

2. SQL Syntax Distribution

Training data should cover comprehensive SQL. Public datasets cover well on basic SQL such as querying, filtering, aggregating, and sorting.

We need to generate more samples containing join conditions and subqueries to improve the coverage of advanced SQL.



2.1 DATA PREPARATION Prepare The Training Data - Part 2

1. Enhancing Query Coverage by Query Rewriting

Asking the same question in various ways can improve the robustness of Text-To-SQL performance. We use the open-sourced LLM to rewrite the original query to diversify the sample. The rewriting need to conform to the actual user query norms, and the rewritten query combined with table information will become the new training samples.

2. Covering More Complexity Levels

The queries and SQL in public datasets are relatively simple. The right figure shows the difficulty level distribution of samples in public datasets. In real business scenarios, there are many hard and extra-hard questions, and GPT-4 has an outstanding performance in solving them. To improve our model, we particularly extract samples covering hard and extra-hard levels from the Tencent dataset as the training set (right-bottom figure). This allows us to solve more practical problems.

3. Fixing Bad Cases in Foundation Models

Foundation Models contain bad cases, which could influence the effectiveness of fine-tuning. To address the case, we use enhanced samples to correct the foundation model and achieve better performance.

4. Including Real Users' Habits and Usage Patterns

We include samples from real users' habit--conventions in specific business, time representations, metrics calculation--to improve models' memory and performance.



Prepare The Training Data - Part 3

Instruct-Tuning Template

 ### Instruction:	Тад	Description
<user query=""></user>	User query	User's questions in natural language, including hints (relationships between fields/time & data formatetc.)
### Input:		
TableName: <table name=""> ,</table>	Table Name	<table_name></table_name>
TableFields: <table field="" schema=""></table>	Table Name	
		The collection of fields in a table, representing as follows:
### Response:	Table Field Schema	<field_original_name>: original field name in the table <field_name>: display name of the field <field_type>: unified field type</field_type></field_name></field_original_name>

Prepare The Training Data - Part 4

Instruct-Tuning Template

- Business users often configure a dictionary (enum) for certain dimensional fields, and they usually do not explicitly specify these field names when querying.
- To solve the problem, we input such field enums as complementary dataset and insert them in the tuning instruct. By in-context learning from the instruct dataset, models could auto-fill what is implied by users from the field enums.

Instruct Sample

(The training corpus is in Chinese corpus, and the prompt template is shown as it is)

Instruction:

找出每个可以容纳100多名学生的宿舍的设施数量, join条件为dormid和id (find out all amenities that can host more than 100 students, *join dormid with id*)

Input:

表: dorm 表字段: Dormid(宿舍编号): bigint, dorm_name:宿舍名称:string,

dorm_name:宿舍名称:string, student_capacity:学生容量:bigint

表: has_amenity, 表字段: Id(编号):bigint, amenid:设施编号:bigint

Response:

SELECT COUNT(*), t1.dormid FROM (SELECT dormid FROM dorm WHERE student_capacity > 100) AS t1 JOIN (SELECT id FROM has_amenity) AS t2 ON t1.dormid = t2.id GROUP BY t1.dormid;

Data Cleaning And The Final Dataset

Challenge of Dirty Data

Neither open dataset nor Tencent dataset are perfectly reliable
 Datasets generated by GPT are not 100% correct

Examples of Dirty Data

1. Unalignment of table/field names between Query and SQL, upper/lower case, missing fields

- 2. Fields used in Select and Group never appeared in the Prompt
- 3. SQL section used but never defined
- 4. Clearly requested for predicate push down, but SQL generated does not contain optimization
- 6. Fields defined do not align with what is used in the SQL
- 7. Rewriting queries to increase the variety, but these queries are actually not the same
- 8. Querying multiple tables, which could be queried in a single table
- 9. Can't tell the attributes between tables, e.g., used attributes in Table A for Table B
- 10. Time range, e.g. , last week had 7 days, but the data sample only has 6 days
- 11. Missing filtering and joining; Adding unnecessary filtering
- 12. Bad cases on special fields caused by base models require fixing.



We prepared a total of 16K+ refined samples for model training. Sample distribution as above.

2.2 FINE-TUNING PROCESSES

Hot Take-aways For Fine-tuning

Dataset	LM odel FT) Fine-tuned Model
Training	Inference
 finetune_type=LoRA LoRA Model Name=q_proj,v_proj,k_proj,o_proj LoRA dim=16 max token length=4096 optimizer=AdamW learning_rate=0.0001 epoch=3 	To increase the stability of returned results, config the params as follows: • temperature=0 • top_p=0 • top_k=0 • repetition_penalty=1 • length_penalty=1 • num_beams=1

1. Model with more params: Full-parameter tuning works better on larger sample size, LORA works better on smaller sample size.

2. Model with less params: Full-parameter tuning works better.

3. Full-parameter tuning might cause a huge problem on "forgetting", but LORA might save it.

4. Compared to sample size, sample quality is more important. A high-quality sample with a small size can achieve good performance.



2.3 BENCHMARK TESTING

Our Fine-tuned Model vs. GPT-4 On Benchmark

Model	BIRD Dataset Accuracy(No RAG Process)	Tencent Dataset Accuracy
DeepSeekCoder-33B(not fine-tuned)	35%	34%
GPT-4	45%	52%
Our Model (fine-tuned DeepSeekCoder-33B)	45%	54%



- 1. The fine-tuned model has the same accuracy as GPT-4 on the BIRD dataset.
- 2. Fine-tuned model is more accurate than GPT-4 on Tencent dataset.
- 3. The fine-tuned model has a better result than GPT-4 on medium-level complexity.

2.3 BENCHMARK TESTING

Our Fine-tuned Model vs. GPT-4 On Benchmark

SQL Complexity	Prompt	GPT-4	Our Model
Predicate Push Down & Column Pruning	### Instruction: 查询2016年2月内运送重量最大的订单的城市名称和人口数量 ### Input: 表: text2dsl_stdtest.shipping_shipment表字段: auto_id:auto_id:bigint,ship_id:运输编号:bigint,cust_id:客户编 号:bigint,weight:重量:double,truck_id:卡车编号:bigint,driver_id:司机编 号:bigint,city_id:城市编号:bigint,ship_date:发货日期:string] 表: text2dsl_stdtest.shipping_city表字段: auto_id:auto_id:bigint,city_id:城 市编号:bigint,city_name:城市名称:string,population:人口数量:bigint ### Response:	SELECT c.city_name, c.population FROM text2dsl_stdtest.shipping_shipment s JOIN text2dsl_stdtest.shipping_city c ON s.city_id = c.city_id WHERE s.ship_date BETWEEN '2016-02-01' AND '2016-02-29' ORDER BY s.weight DESC LIMIT 1; No predicate push down No column pruning. Problems in readability and SQL performance	<pre>SELECT tl.city_name, tl.population FROM SELECT city_id, city_name, population FROM text2dsl_stdtest.shipping_city) AS t1 JOIN (SELECT city_id, weight FROM text2dsl_stdtest.shipping_shipment WHERE ship_date >= '2016-02-01' AND ship_date >= '2016-02-29' ORDER BY weight DESC LIMIT 1) AS t2 ON tl.city_id = t2.city_id used predicate push down column pruning</pre>
In-context Learning	<pre>### Instruction: 查询2023年8月DataInsight的工单数量,按照工单接单人聚合,返回前2条 数据 ### Input: 表: bds_analysis.t_incident_detailed,表字段: incident_label:工单问题标签 :string,incident_id:工单id:bigint,status:工单状态:string,create_by:工单创建 人:string,create_at:工单创建时间:string,ack_by:工单接单人:string,ack_at: 工单接单时间:string,resolve_by:工单结单人:string,resolve_at:工单结单时 问:string,assign_by:工单响应人/指派人:string,incident_class1:工单一级分 类(Datainsight, Datatalk, Datamanager):string #### Response:</pre>	<pre>SELECT ack_by, COUNT(*) as count FROM bds_analysis.t_incident_detailed WHERE incident_class1 = 'DataInsight' AND create_at >= '2023-08-01' AND create_at < '2023-09-01' GROUP BY ack_by ORDER BY count DESC LIMIT 2;</pre>	<pre>SELECT ack_by, COUNT(*) AS order_count FROM bds_analysis.t_incident_detailed WHERE incident_class1 = 'DataInsight' AND create_at >= '20230801' AND create_at <= '20230831' GROUP BY ack_by ORDER BY order_count DESC LIMIT 2</pre>

3. [USE CASE] TEXT-TO-SQL IN **PRODUCTION WITH** QUERY REWRITING

3.1 TENCENT VIDEO & NBA NEWS Business Needs For Text-to-SQL

Tencent Video & Tencent News

- 1. Tencent is the exclusive agency for NBA online, and need for querying score/stats are high. Tencent video has similar needs for data-driven operations.
- 2. By using natural language to query sports data, will lower the hurdle for sports content creators increase the scale.

Business Background

- 1. Users' SQL skills, and familiarity with the BI tool prevent them from generating business insights.
- 2. The skills required for users to independently conduct data analytics are quite high.
- 3. If Text-to-SQL reaches a high accuracy, and users could interact with the database directly through natural language, data analysis would be much more available.

Sample Users' Queries

- Sports Content Creation: How has LeBron James performed in the past 10 games?
- Film and television IP consumption analysis: Show me the click-through rate of users who have watched film and television show after a (specific) advertisement page.
- News content consumption analysis: Consumption views of articles in recent days, categorized by content type.

When evaluating public and Tencent datasets, the fine-tuned model's performance can compete with GPT-4 and even surpass GPT-4 in some difficult problems. However, to use Text-To-SQL online, the model's accuracy needs to reach over 80%. We found that there is a lack of domain knowledge and a need to use context learning to bridge the gap between users' queries and schemas. We use an enhanced data process to address this to optimize the Text-to-SQL performance.



3.2 RAG & QUERY REWRITING

RAG + Query Rewriting: Enhancing Text-to-SQL in production



Combine with other techniques like Query Rewriting.

- 1. NBA News contains various professional terms and domain knowledge. For example, in the NBA Finals, the two teams must win four out of seven games to win the championship.
- 2. To achieve better results, passing the NBA domain knowledge to the Text-to-SQL model through prompts is necessary. Solely relying on the training of the Text-to-SQL model is not enough to solve business problems.
- 3. To identify which table and data to solve the user's question, we enhance the retrieval through query rewriting & RAG. At last, we run Text-To-SQL on that table.

3.2 RAG & QUERY REWRITING

Rewriting Queries To Locate Data Resource More Accurately



Challenge

- The user's query is usually bounded with business scope, carrying heavy business knowledge.
- The query expression is usually colloquial. Similarly, different users' expressions of the same concept are not always the same and often not clear enough.

Solution

- The input queries need to include the actual intent of users' queries and combine domain knowledge to obtain a precise understanding of their intent.
- LLM has excellent reasoning and generalization capabilities, we can rewrite queries using a general open-sourced LLM with few-shots.

DATA⁺AI SUMMIT



3.2 RAG & QUERY REWRITING WITH/WITHOUT QUERY REWITES

The query accuracy is **only 29.2% when solely using Text-to-SQL** Service, and it **improves to 87.8%** when **combined with RAG & query rewriting**.

Without Query Rewriting	SQL	Correctness	With Query Rewriting	SQL	Correctness
Who won last year's finals?	SELECT team_name FROM team_match_stats WHERE type = 'Finals' AND is_win = 1	False	<i>Which team won the most games in the 2023 season of the Grand Finals?</i>	SELECT team_name, COUNT(*) FROM player_match_stats WHERE season_id='2023' AND season_type='Finals' GROUP BY team_name ORDER BY COUNT(*) DESC LIMIT 1	True
<i>Who are the players who have hit 50+ three-pointers this season?</i>	SELECT player_name FROM player_match_stats WHERE season_id = '2023' AND three_point_goals_attempted >= 50	False	<i>In the 2023 season, which players have a cumulative total of more than 50 three-point shots?</i>	SELECT player_name, sum(three_point_goals) as total_three_point_goals FROM player_match_stats WHERE season_id = '2023' GROUP BY player_name HAVING total_three_point_goals >= 50	True
<i>What is the Pacers' best season record?</i>	SELECT MAX(score) , season_id FROM team_match_stats GROUP BY season_id	False	<i>What season did the team Pacers win the most games?</i>	SELECT season_id FROM team_match_stats WHERE team_name = 'Pacers' AND is_win = 1 GROUP BY season_id ORDER BY count(*) DESC LIMIT 1	True

3.2 RAG & QUERY REWRITING Text-to-SQL In Production



3.3 BI DEMO **Demo: Text Queries & Diagrams Summarization**

								service in	2. SQL (Seneration
Fabric * INKI2	总结图卡关键信息	Fabric	查询业务数据					114,00		
	• • • • • • • • • • • • • • • • • • •			1.1	Vatura	al langu	age guery	true -	an a	146, 10, 17 (Hold), 187 (H
C REAL	编标算总统用因图+10.8	 RETREE 	查询20230801业务有为	的各个空间的;	8成本、查询成功、1	(请失数次数以及对应的标准值)	调次数分别有多少? 返回前10条数据	815,10		r, nie, e, pil
	Tableoux & KU, KU, R. R. (2019) Tableoux & KU, KU, R. R. (2017) Tableoux & KU, KU, R. R. (2017)	📹 🕒 Markdown文本						100.00	te se "Discourses"	
A B Markdown (18)*		● 巴总结图卡关键信息	查询SQL	2 1	Fext-t	o-SOI		-	a data + "mineser"	
< C 272508	2020001 2020000 2020001 2020001 2020000 2020001	• 巴 查询业务数据				OGGL			Table_18 = "?"	
···	ARE © CLORIOUSE © STARDONS © I « MP ARE © CLORIOUSE © STARDONS © I « IN+	Markdown JyBA						serves by		
 巴里市业务数据 ア 品類第十-0908 	###### 2 Summarization	(2) 重次主务数据						414,12		
9		₩ 巴 宣衛业务数据	-		1.10			and the second se		
•	 · 时周边图: 家族的时间地造机从2013年9月30日至2013年9月6日。 · 植废信息: 家族的中的地皮为可能(天)", 表示每天的资源。 	· 巴 查询业务数据						timit		
	 · 简称或语: 资源中的简称为 'pd0', 表示重调和时的自分之50, // 第一次重调和时的自分之50, // 第一次重调和时间和分子的一次重调和时间的自分之50, // 第一次重调和时间和分子的一次重调和时间的自分之50, // 第一次重调和时间和分子的一次重调和时间和分子的一次重调和分子的一个不同的一个不同的一个不同的一个不同的一个不同的一个不同的一个不同的一个原本在一个目的一个不同的一个不同的一个不同的一个不同的一个原本在一个原本在一个目的一个不同的一个不同的一个不同的一个不同的一个不同的一个不同的一个不同的一个不同	◎ 四 总结图卡-0908								
	消費な力が打 したおかから、使用には使用できた使用をなかきませんの支援があるそのとなか、可以に見た、不同に使用ななかきまだかな予想にはななのおし、可能かかた使用なられたのも、までラークトを用いて用たまれな、します「		86112	3.1	Fable	return		这个SQL查询的	目的是找出最近(2024年5月24日 日本口公士98日年	日至2024年6月2日)腾讯视
	ETERTER		\$7.32					90APP38	取预门的干部剧味	1。以下定SQL的详细解释·
	2. 27日後の方で、市場路線加減温度後、当い方ができな原用には19日前時の20日本、当いたいできな原用になると、共同当時時代を加入、共同当時期の時の加減温。 3. 自然交叉分析: 北京近時美術語意識意見、可以在具体建築上分析室用時的行為現現実、時間、可以分析性特定的成点的特定的展示な新生活空間時時以現実、技工型構成分析的語言。		10.1070	10100.000	1000.0000.00	100,000,00,00	1000,000,0001,0001	1. SELECT	title, setter, storat, setter,	as as a standard an
	建议信息		8110				and the second s	:选	译需要展示的字段,即剧集的标题	和有效播放次数
	希察周察分析的结果,可以综合结论以下建议:				-		-	的总和	有效播放	文次数是指正片播放完成度>=
	1. 时方面间接到收到的数据波莱灵,可以考虑你活在面包里。 (其27月1号将约回。 2. 对于面询用时较低的数据源类型,可以进一步研究其优势和特点,以便在需要快速面询的场景中更好地利用这盘数据源。		MARCH AND	100010			127824	0.14且非正片	播放时长>=30s的播放结束次数,	这是一个衡量剧集热度的重
	3.针对查询用的存在透动的时间点,可以进一步分析波动的原因,并采取服益的指施来代化重消性能,以循保系统的指定性和可靠性。		Scener-	-	1001		100000	要指标。		
	具他要求		*******		-		and the second s	0. 5000		山谷合約支
	 在通過公分局基本時,通具有描述時,通規共同指定時,通見規模描述。 不需要發展用其他爆发的存在,只開機機變換的設施进行分析。 		Roomerson and Real Property lies	_		-		2. FRUM	中苏取教授 这个声句:	:从结正时农
	• 不需要要能出具体的数量使位。随等指标的单位在数据中的确结出。 以上更可能但你教室进行相近性状计、须愿当物系建立位意的已经再高、将或具体主条用系统教服装置、可以进一步等入分析和过程数程中的增压问题和规律。			-	-		-		甲状联蚁病,这十农巴;	古」优频 增 成时相关统计语
			PROBATION OF THE OWNER OWNER OF THE OWNER OF THE OWNER OWNE OWNER OWNE OWNE OWNE OWNE OWNE OWNE OWNE OWNE	-	-	-	2019030	A25. o		
	上方是"品结图卡关键信息"模板的尚著样例,意可以在下方"导入总结图卡"的功能成中,选择你关注的图卡进行总结吧-		SQL解释					3. WHERE	AT	: 过滤条
			普先,我们使用SELECT语句)	た理察委的列, 奴	括空房名、总成本、月	(會演次数、总會演失数次数考)	总标准查询次数。然后,我们使用WHERE子句】	BNIB#	024年5月24日至2024年6月2日則	J间的数据,这是为了获取最
	输入"广查者用绘制版功能,也面接输入处本法行音通问面。		组、以便我们可以计算每个空	间的总成本、总重	该次数、总查询失数;	文數和总标准查询次数。最后,	我们使用ORDER BY子句按总成本簿序排列,并	ttmux 近时间段内的	数据。	
	© 9.12825 1925 1925 ×		△ 根据远的提问,重向SQL	4.5	SQLi	nterpret	ation	4.00	D 24-12 PU4r 41 BU	
	aut · · · · · · · · · · · · · · · · · · ·							4. AND 約伊吉由加尼	: 穴远洋剧果天空	为Z的店来,这可
	795009 00 Kul Ala ana ana asan asan asan asan asan asan			T	ovt_T	0-801	M OlaCh	net coverta	或电影,英仲载大于 的	JEX.
					576-1			5. GROUP BY	: 按照剧集标题	进行分组,这样可以计算
								每个剧集的总	有效播放次数。	
	FEEADINE RE FEEADERJINHEADER ENTERJUN									
								6. ORDER BY	desc :	按照有效播放次数的总和降
	(28.08%)							序排序,这样	可以得到播放次数最多的剧集。	
								7 LIMIT 10 1	双出排序后的前10条记录 即最近	最少的十部剧集.
	-R# F\$6528005 #6 6478280.42							7. CINTI 10.1	A REAL PROPERTY AND A REAL	MAY SHY I LIPACIZES
57		הבי את 🐠	T+#					这个查询的目的	是为了找出这段时间内腾讯视频A	PP上最受欢迎的电视剧,以
8	3 Selecting more cards	🛛 💆 腾 饥发	」冶					便分析和推荐第	8. 3 SOL II	nternretation
8	o. Deleting more tards	Dachhar	rd Summ	ori-	otion			0.54	0.002	nerpretation
💤 🧘 kunkcheng 🔯		Dashboa	ra summ	Iariz	auor			C' 重新生成	回月王内谷	凸 有帮助 P 无帮助
	©2024	Databricks Inc.	— All right	s res	erved			And A second second		
								请输入你的问题		

DAT/

1

88入群享好礼 ×

(U) OlaChat 内测版

1. NL Query

根据您的需求为您生成SQL:

3.3 BI DEMO

Demo: BI Copilot Enables Natural Language Queries

Image: Contrast of the state	U OlaChat ×		U OlaChat ×	🥶 OlaChat 🛛 🕹
 A state C intention inference: Text-to-SQL C interpretation of SQL C interpretation of	腾讯视频半熟恋人3最近的DAU	1. Natural language query	酸讯视频半熟恋人DAU趋势	顾讯视频半熟恋人DAU趋势
 C Brithersterstersterstersterstersterstersterst	查看返回结果		查看近回结果	查看思考过程
 A start interference: Text-to-SQL A intertation inference: Text-to-SQL A most related tables A most related tables A solution A solution	▶ 查询"腾讯视频半熟恋人3最近的DAU"SQL语句		▶ 分析"腾讯视频半熟恋人DAU趋势"	(*) 公析"職道加格半職本 ↓ DAI 14執"
Instructure decode and d	 ○ 意图拆解 	2. Intention Inference:	② 意图拆解 ∨	
 Interpretation of SQL S. Interpretation SQL <li< td=""><td>当前意图为找文字生成SQL,理解后需要分析的问题是:</td><td>Text-to-SOI</td><td>当前意图为找数据表,理解后需要分析的问题是:</td><td>腾讯视频半熟恋人DAU趋势 ▽ 〔〕</td></li<>	当前意图为找文字生成SQL,理解后需要分析的问题是:	Text-to-SOI	当前意图为找数据表,理解后需要分析的问题是:	腾讯视频半熟恋人DAU趋势 ▽ 〔〕
<complex-block></complex-block>	③ 查找腾讯视频DAU表 理解错了	Text-to-OQL	③ 查找腾讯视频DAU表 理解错了	
○ SERSE ○ SERSE <t< td=""><td>根据您的需求,需要将选出深圳市的人口,并分析他们在年龄维度下的 分布情况。由于您没有指定其他感觉,平均给会直接使用年龄作为分析 编度。很不不。我们将使用于给约人就分析工具造调深圳市人口在年龄 建度下的分布情况,并输出结果。</td><td></td><td>根据您的需求,需要等适出深圳市的人口,并分析他们在年龄推覆下的 分布情况。由于远没有加过其他推进,平县将会直接使用年龄作为分析 维度,即下来,我们将你用什么的人都分析工具者场深圳市人口在年龄 推度下的分布情况,并输出信用。</td><td>~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~</td></t<>	根据您的需求,需要将选出深圳市的人口,并分析他们在年龄维度下的 分布情况。由于您没有指定其他感觉,平均给会直接使用年龄作为分析 编度。很不不。我们将使用于给约人就分析工具造调深圳市人口在年龄 建度下的分布情况,并输出结果。		根据您的需求,需要等适出深圳市的人口,并分析他们在年龄推覆下的 分布情况。由于远没有加过其他推进,平县将会直接使用年龄作为分析 维度,即下来,我们将你用什么的人都分析工具者场深圳市人口在年龄 推度下的分布情况,并输出信用。	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Comment of SQL Commen	◎ 数据探索 ~	3 Most related tables	 ○ 数据探索 	
 Constr Constr< Constr< 	找到 2 张相关数据表	3. MOST related tables	找到2张相关数据表	2/20 2/25 3/2 3/2 3/7 3/12 3/17 3/21
・ 生物のに の 生物のに の 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	· page and the data prove private angle distribute data at			
● 生物30. Su				· 就照米源 · 回 会页数 · E 综乙内容 · E 难度_难度_
 C SLARC C SCALE C SCALE	O Helton		() 查询执行	其他辅助说明
O SOL NG Fille UP CO C SOL NG Fille UP CO State UP CO		4. SQL return	音楽地行日完成	使用持续分析快速发起自助查询 →
Image: State of the state	ວິດແ ເ+ ເ/ບ ເງ			将自助查询保存的图表添加到仪表盘汇中 →
Image: Solar interpretation of SQL S. Interpretation of SQL Solar interpretation of SQL S. Interpretation of SQL Stock interpretation of SQL S. Interpretation of SQL <td></td> <td></td> <td>◎ 可视化出图 ~</td> <td>白色八纪外国上在方法国亦是</td>			◎ 可视化出图 ~	白色八纪外国上在方法国亦是
Image: Construction of SQL S. Interpretation of SQL Image: Construction of SQL I			腾讯视频半熟恋人DAU趋势	日初方有的因下怎么说用支重
Image: Solution of SQL 5. Interpretation of SQL Image: SQL of SQL o				
③ SOL版程 ⑤ SOL版程 ⑤ SOL版程 ⑤ SOL版程 ⑥ SOL版程 ⑥ SOL版程 ⑥ SOL版程 ⑧ SOL DE			~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	1. Natural language query
	⊘ SQL解读 ~	5. Interpretation of SQL		2. Toyt to COL Execution
E2計算了开出值,即 (今天播放时长 - 昨天播放时长) / 昨天播放时长 - 昨天播放时 - 100000000000000000000000000000000000	这个SQL宣询用于生成暗讯视频的日总播放时长环比情况。留先, 它从 表中提取了 在 20240509 和 20240510 2回的资源。通过使用SUM通 数和CASEE的, 计描述 702340510 (今天) 和20240509 (時天) 的 总播放时长((平均))		2/20 2/25 3/2 3/2 3/7 3/12 3/17 3/21	 Text-to-SQL – Execution Card Visualization Hints
if输入你的问题 1 Text-to-SQL if 输入你的问题 1 Text-to-Viz 0 OlaCh	它计算了环比值,即(今天播放时长 – 昨天播放时长)/ 昨天播放时长 * 100%。结果命名为			
	请输入你的问题	Text-to-SQL	请输入你的问题	Text-to-Viz 🝈 OlaCha

DATA⁺AI SUMMIT

©2024 Databricks Inc. — All rights reserved

腾讯灯塔 29

THANK YOU FOR LISTENING! Q&A;)



Hehuan Liu Senior AI/ML Scientist (+86) 13910972658 liuhehuan@gmail.com | mermaidliu@tencent.com www.linkedin.com/in/hehuanliu/



Kun Cheng Product Manager (+86) 13679263697 | (+1) 617-599-7131 kchengdusp@outlook.com www.linkedin.com/in/kun-cheng-3040701b3/