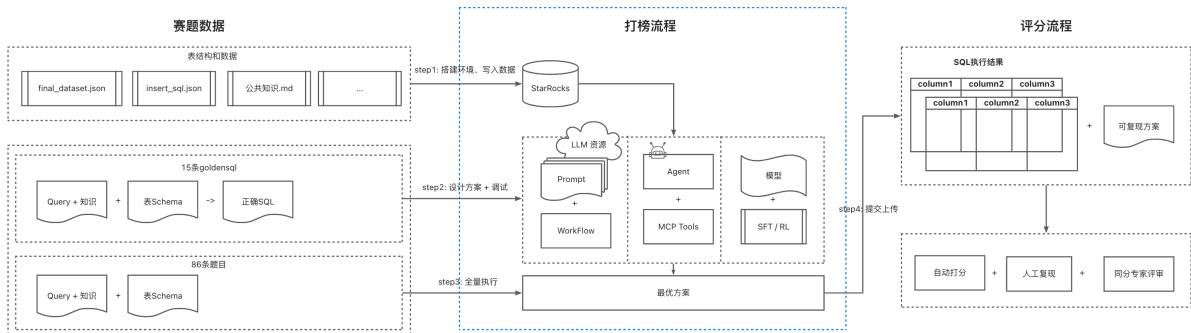


一、比赛流程



- 1、下载赛题资料，包括：数据库表结构与数据、包含15条正确SQL的goldensql文件、以及86道比赛题目；
- 2、在本地或云端搭建StarRocks，完成数据环境准备；
- 3、基于goldensql选择和调试方案，确定最终执行方案；
- 4、对86道题目逐一执行并提交结果，系统将自动评分并在排行榜中展示成绩，同时通过邮件反馈每条SQL的执行正确情况。

环境和技术栈限说明：

- 1、数据库限定使用StarRocks
- 2、后端代码限定使用Python
- 3、如果方案需要使用类dify的AI工作流平台，限定使用腾讯云智能体开发平台

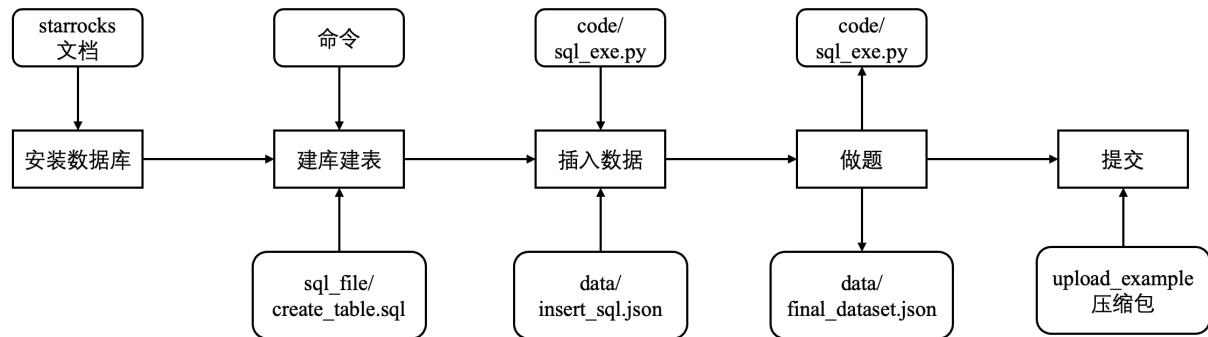
二、下载文件说明

注：本次赛题均为构造需求和数据，与游戏真实情况无关。赛题所有权、知识产权及其他权益归腾讯所有，参赛者不得擅自泄露、对外发布或者进行非本次大赛目的的使用。

```
final_for_student/
├── code/
│   └── sql_exe.py
├── data/
│   ├── final_dataset.json
│   ├── insert_sql.json
│   ├── common_knowledge.md
│   └── schema.json
└── sql_file/
    ├── create_table.sql
    └── drop_table.sql
└── upload_example.zip
```

- 1、code文件夹中sql_exe.py包含 数据插入与sql执行功能
- 2、data文件夹中包含本次大赛的题目 (final_dataset.json)、数据信息 (insert_sql.json 和 schema.json) 以及公共知识 (common_knowledge.md)，作为游戏领域知识信息补充。
- 3、sql_file文件中包含了批量建表和删表语句，方便快速构建数据环境
- 4、upload_example.zip中包含了本次大赛规定的赛题提交样式

使用流程：



三、方案和baseline

业内Text2SQL领域分为3个方向：workflow、agent、微调，大家可以选择合适自己的方向，同时也鼓励同学们探索出更多有意思、有效果的新方向。

1、Workflow

- 如果需要使用AI工作流平台，仅可以使用[腾讯云智能体开发平台](#)（登陆方式见附录）。我们在默认空间搭建了[DIN-SQL](#)作为baseline，Pass@1准确率 **15/101**。可以将赛题文件中export-din-sql.zip导入自己的空间查看实现细节。
- 如果自行开发脚本，限定技术栈使用python，需要提交可被复现的源码，作为同分情况下方案对比依据。

2、Agent

推荐使用[Datus](#)或其他专注在数据领域的开源agent框架。

如果选择该方向，限定技术栈使用python，需要提交可被复现的源码。

3、微调

微调的核心思想是增强模型（尤其是小模型）在Text2SQL上的能力，从而让其达到甚至超越一些大参数规模模型。

可以基于开源小模型先进行全参数微调，再进行强化学习训练。

- 微调部分，可尝试预训练以及后训练SFT两种方式，主流的训练框架有：
[GitHub - hiyouga/LLaMA-Factory: Unified Efficient Fine-Tuning of 100+ LLMs & VLMs \(ACL\)](#)

[2024\)](#)

[GitHub - NVIDIA/Megatron-LM: Ongoing research training transformer models at scale](#)

- 强化学习部分，可以尝试PPO GRPO DAPO 等算法，主流训练框架有：

[GitHub - huggingface/open-r1: Fully open reproduction of DeepSeek-R1](#)

[GitHub - volcengine/verl: verl: Volcano Engine Reinforcement Learning for LLMs](#)

- 训练数据部分，业界的一些公开SQL数据集有Spider，Bird等

四、提交事项

1、提交内容

upload_example.zip解压后目录结构：

```
1   └── upload_example/
2       ├── code/
3       ├── dataset_exe_result.json
4       └── 方案介绍.pdf
5           └── 代码使用说明.pdf
```

dataset_exe_result.json为sql执行结果，评分接口将检测该文件名的文件作为评分文件，请同

同学们：1、**提交zip文件**；2、**zip文件解压后保障有且仅有1个dataset_exe_result.json**

否则将无法正确判分。

code文件夹存放可复现源码。请大家将跑出的dataset_exe_result.json、源码、方案介绍和代
码使用说明，按 队伍名称.zip 压缩后提交。

2、详细说明

2.1、workflow & Agent

workflow: 如果是个人开发的workflow，请提交完整版代码，以及python环境构建指引;如果是使
用腾讯云ADP平台开发的workflow，可被作为最终提交代码版本。

Agent: 要求提交完整版代码以及python环境构建指引

```
1   code ( 个人workflow or ADP空间名称-应用名-工作流名 )
2   dataset_exe_result.json
3   文档 ( 方案、python环境构建 )
```

2.2 微调

采用该方案的同学需提交 **微调代码、方案与微调实验细节**

- 1 code
- 2 dataset_exe_result.json
- 3 文档（方案、微调实验结果以及python环境构建指引）

注意：请各位同学重视文档书写，尽可能提供足够的信息使得可以成功复现方案！

五、评分准则

本赛道将于2025.12.16结束，最终排名将截取比赛结束时实际排名前10的队伍（如果出现同分往下顺延，截取10个不同分数）。同时对其方案进行复现，如复现后发现无作弊情况则按照顺序颁奖，如果出现同分的情况，则由专家基于 **方案创新性** 以及 **方案实用性** 对同分方案进行评审，确定最终排序。

最终分数计算方式如下

- 1 最终分数 = result正确分数
- 2
- 3 # 同分情况
- 4 最终分数 = result正确分数 + 专家评分

注意：若复现中发现某队伍作弊，则取消其成绩，并由下一名队伍递补。递补队伍也需接受复现审核，此过程持续至前十名队伍均被确认无作弊且排名有效为止。

如果提交内容代码信息不全或者无法上传，我们会在比赛时间结束后通过邮件的方式为前10的队伍提供最终版本文件上传地址以及方式，请各位届时留意具体信息。

附录

数据库安装部署教程

该教程方便大家部署自己的数据环境，仅供参考。

1、StarRocks数据库环境搭建

硬件设备

本次环境搭建使用腾讯云标准型SA9 4核8GB的CVM（Cloud Virtual Machine），该机器内置docker以及mysql

```
1 # 设备os信息
2 NAME="TencentOS Server"
3 VERSION="3.2 (Final)"
4 ID="tencentos"
5 ID_LIKE="rhel fedora centos"
6 CENTOS_MANTISBT_PROJECT="CentOS-8"
7 CENTOS_MANTISBT_PROJECT_VERSION="8"
8 NAME_ORIG="CentOS Linux"
```

```
1 # 文档
2 腾讯云官网 : <https://cloud.tencent.com/?Is=sdk-topnav>
3 腾讯云docker文档 : <https://cloud.tencent.com/document/product/1207/45596>
4 starrocks官方文档 : <https://docs.starrocks.io/zh/docs/3.1/quick\_start/deploy\_with\_
```

注意：由于starrocks 2.5.12的文档已经被官方下架，建议参考starrocks 3.1的文档，不限制版本。

镜像部署

在完成了docker安装以及mysql安装后，可以使用docker安装starrocks数据库

```
1 sudo systemctl start docker # 启动docker
2 docker ps # 查看docker服务是否启动
3 docker pull starrocks/allin1-ubuntu:2.5.12 # 拉取starrocks镜像
```

注意事项1：在拉取镜像的时候容易出现如下报错 “**docker pull 命令报错 “Get <https://registry-1.docker.io/v2/>": xxxxx (Client.Timeout exceeded while awaiting headers)” 则说明是网络问题”此时建议参考 [云服务器 腾讯云软件源加速软件包下载安装和更新_腾讯云](#) 更换源。

注意事项2：关于镜像拉取的版本，本次大赛的试题构建的starrocks版本为starrocks 2.5.12作为测试环境。对于不同的linux操作系统应该都可以拉取starrocks/allin1-ubuntu:2.5.12这个版本的镜像进行安装，上述机器的os为centos8，但是实际测试发现在上述机器上安装starrocks/allin1-ubuntu:2.5.12没有问题。

2、建库与建表

建库

在插入数据前，需要链接starrock数据库并且建库，使用如下命令进行建库

```
1 mysql -P9030 -h127.0.0.1 -uroot # 链接数据库，注意端口对称，需要与docker启动命令  
2 CREATE DATABASE database_name; # 建库命令，其中database_name为建库的库名  
3 show databases; # 查看库
```

当使用show databases命令后，命令行出现所建的库即表示建库成功

建表

建库完毕后切换到具体的库，并且使用create_table.sql进行建表，具体的命令如下

```
1 use database_name; # 切换到具体的数据库  
2 source create_table.sql # 执行建表语句  
3 show tables; # 查看所建的表
```

上述执行代码并运行show tables;命令，应该可以看到全部的84张表信息

3、插入数据与sql执行

插入数据

推荐使用code/sql_exe.py中的insert_data_with_pymysql函数进行数据插入，从而确保与实验环境执行情况保持一致。由于code/sql_exe.py需要使用pymysql库进行数据库链接，请同学们搭建python环境，并使用代码前请先使用如下命令安装。

```
1 pip install pymysql
```

注意：请确保insert过程只执行一次，多次插入可能会导致后续执行结果出现偏差，从而影响判断结果，如果出现多次插入的情况，可使用sql_file/drop_table.sql删除表以及表数据，然后重新按照上述过程重新建表并插入数据

sql执行

推荐使用code/sql_exe.py中的execute_sql_with_pymysql函数执行sql，该函数包含了本赛道针对结果的小数数值处理逻辑

```
1 针对假小数（即小数点后面只有0），将其转化为整型 eg: 3.0 -> 3  
2 针对真小数（即小数点后存在多位有效数），将其保留两位小数 eg: 3.3333 -> 3.33
```

由于评分结果也是用该函数跑出的来的，所以使用execute_sql_with_pymysql可以最大程度确保所执行结果与评分规则保持一致

注意：如果不使用上述代码进行result处理，在实际运行过程中需要注意处理Decimal数据类型，将Decimal数据类型也保留为两位小数！

腾讯云智能体开发平台 (ADP) 操作说明

1、账号登陆腾讯云平台

- 登陆地址：[登录 - 腾讯云](#)
- 账号：邮箱前缀@100044597907 (比如赛事官网填写的邮箱信息是：zhyu@seu.edu.cn，那么账号名称填 zhyu@100044597907)
- 初始密码：TGAC.data105
- 初始登陆后自行设置密码并绑定手机号

2、进入ADP平台

控制台搜索ADP产品

The screenshot shows the Tencent Cloud Control Console interface. At the top, there is a navigation bar with tabs for Product Management, Product Monitoring, Security Center, Architecture Governance, and Management Dashboard. A search bar is located at the top right. Below the navigation bar, a message box displays a warning about AccessKey expiration. The main search results area has a heading 'ADP' and a search input field. A card titled '腾讯云智能体开发平台' (highlighted with a red box) lists '消息队列 Pulsar 版', '容器服务 - TKE Insight - Workload Map', '访问管理 - 身份提供商 - 角色SSO', and '访问管理 - 身份提供商 - 用户SSO'. To the right, there are sections for 'API Explorer' and '访问密钥'. Below this, there is a grid of service cards:

域名注册 提供域名注册、查询、购买到移动解析一站式满足...	短信 快速稳定、简单易用、触达全球的短信服务，支持...	对象存储 安全稳定、海量、便捷、低延迟、低成本的云端存...
弹性公网 IP 稳定、安全、弹性、高性能的云端计算服务，实时...	边缘安全加速平台 EO 面向海外市场提供边缘一体化服务，聚焦流媒体与...	ICP备案 快速初审、免费部署、备案补缴的网站备案办理服务
轻量应用服务器 简单易用的云服务器，助您快速构建并管理轻量级...		

默认空间选择din-sql查看baseline

The screenshot shows the 'Application Baseline Management' page for the 'din-sql' space. The top navigation bar includes tabs for Application Settings, Knowledge Management, Workload Management (highlighted), Application Testing, Application Release, and Application Operation. A search bar is at the top right. The main content area shows a table of baseline information for the 'din-sql' space, which was automatically generated on 2025-10-30 00:00:00 and is pending release. The table columns include Workload Name, Modifier, Update Time, Release Status, and Whether Enabled. The 'din-sql' entry is listed with 'mxyuman' as the modifier, '2025-10-31 11:51' as the update time, '待发布' as the release status, and '已启用' as the enable status. Action buttons for Edit, Delete, and More are available for each row.

3、添加模型

官方预计会在比赛中期提供免费deepseek-v3模型，前期大家可以使用自行购买的模型。

进入自己队名对应的空间，添加第三方模型。

deepseek-v3和r1添加示例：

<p>模型类型 <input checked="" type="radio"/> LLM</p> <p>模型名称* <input type="text" value="deepseek-chat"/></p> <p>模型ID* <input type="text" value="deepseek-chat"/></p> <p>调用地址* <input type="text" value="http://i.deepseek.com/chat/completions"/></p> <p>密钥 <input type="text"/></p> <p>模型上下文长度* <input type="text" value="64"/> k</p> <p>输入长度限制* <input type="text" value="64"/> k</p>	<p>模型类型 <input checked="" type="radio"/> LLM</p> <p>模型名称* <input type="text" value="deepseek-reasoner"/></p> <p>模型ID* <input type="text" value="deepseek-reasoner"/></p> <p>调用地址* <input type="text" value="http://i.deepseek.com/chat/completions"/></p> <p>密钥 <input type="text"/></p> <p>模型上下文长度* <input type="text" value="128"/> k</p> <p>输入长度限制* <input type="text" value="1280"/> k</p>
---	--

4、开发workflow

新建应用：

工作流管理 - 批量导入，上传 export-din-sql.zip

配置数据库信息。注意有两个节点需要配置：获取表schema 和 SQL执行

```

    107     def original_create_connection(self):
    108         """重载父类方法，返回原生连接对象
    109         """
    110         # -----
    111         # 数据库连接配置
    112         # -----
    113         import pymysql
    114         db_config = {
    115             'host': '', # 目标数据库地址
    116             'user': '', # 用户名
    117             'password': '', # 密码
    118             'database': '', # 数据库名
    119             'charset': 'utf8mb4',
    120             'cursorclass': pymysql.cursors.DictCursor,
    121             'port': 9030,
    122             'connect_timeout': 10,
    123         }
    124
  
```

点击调试，输入final_dataset.json中某一条，点击“去调试”。在对话框输入任意内容。

The screenshot shows the AI Debugging interface. On the left, under '启动工作流的输入变量' (Input variables for starting the workflow), there is a JSON object with the following content:

```

1
2   {
3     "sql_id": "sql_1",
4     "question": "统计2025.07.24的手游全量用户且
      标签为其他，在竞品业务下2025.05.30-2025.07.
      24的在线时长。\\n输出: suserid, sgamedcode,
      ionlinetime\\n\\n",
5     "复杂度": "中等",
6     "table_list": [

```

On the right, the results section shows a blue circular icon with a white 'S' and the text '我的智能体应用01'. Below it is a large button labeled '123'.

得到返回结果

The screenshot shows the AI Debugging interface with a detailed execution flow. The flow starts with a 'sub_questions' node, followed by a '条件判断1' (Conditional Judgment) node, then an 'LLM_medium' node, another 'LLM_medium' node, a '变量聚合1' (Variable Aggregation 1) node, and finally a '获致SQL' (Obtain SQL) node. The '获致SQL' node has an output of 'sql'.

On the right, the results section shows the generated SQL query:

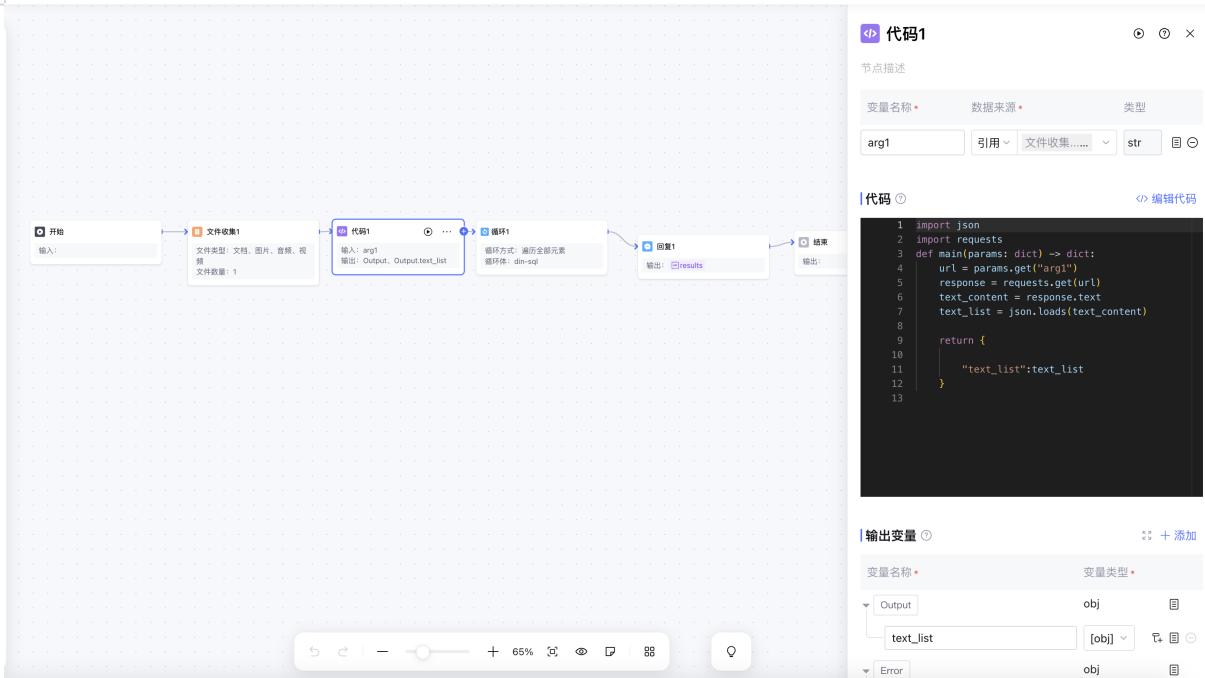
```

{
  "result": [
    "sql": "\nSELECT \n  t1.suserid,\n  t2.sgamedcode,\n  t2.onlinetime\nFROM \n  dim_userid\nWHERE \n  t1.dtstatdate = '20250724'\n  AND \n  t1.tag = '其他'\n  AND \n  t1.vgameappid IN\n  ('vie1','vie2','vie3','vie4','vie5','vie6','vie7','vie8','vie9','vie10')\n  AND \n  t2.dtstatdate BETWEEN '20250530' AND\n  '20250724'\n  AND \n  t2.sgamedcode IN\n  ('initiated','jordass','esports','allianceforce','strategy','playzone','su')\n  AND \n  t2.saccounttype = '-100'\n  AND \n  t2.suseridtype IN\n  ('qq','wxid')\n  AND \n  t2.splattype = '-100'\n  AND \n  t2.splat = '-100'\n",
    "sql_id": "sql_1"
  ]
}

```

Below the results, there is a status message: '65.5s, 16134tokens- 工作流已完成'.

如果希望一次性跑完goldensql，可以将final_dataset.json中的15条golden.json单独保存为txt，新建工作流配置“文件收集”节点，并且循环组件中调用din-sql



详细参考：[腾讯云智能体开发平台 操作指南-文档中心-腾讯云](#)

注意：

- 自行搭建的数据库建议放开所有ip，如果必须有安全规则限制，联系赛事方提供ip白名单

写在最后

在这次比赛中，我们为赛题悄悄埋下了一些“隐藏关卡”。它们或许会暂时带来困惑，却恰恰是通往真正“hard fun”的钥匙。正如类魂游戏不回避一次次“战败”，却在最终突破难关时迸发出巨大的满足感——我们希望你在代码与数据的碰撞中，也能体会到这种由坚持而生的喜悦。

虽然算法大赛无法像游戏那样提供即时反馈，但当你在深夜里忽然洞察解法，或默契地感受到出题者的设计意图，那种豁然开朗的清醒与快乐，正是独属于解题者的高峰体验。

每个人生来都是“游戏的人”。在这个略显异化的时代，愿同学们始终保持游戏者的心态：不惧难关、享受过程，在每一次挑战中找回创造的初心。

祝大家玩得愉快，赛得尽兴！