CSDI SUMMIT

2023CSDI中国软件研发创新科技峰会

2023 CSDI China Software development Innovation Technology Summit

算力 +智能

科技未来·数字时代的进化升级

基于自然语言问答的数据服务架构实践

分享嘉宾：张俊

CSDI SUMMIT

目录

CONTENTS

1 BI新范式的思考

2 超音数建设实践

3 未来规划与总结

01 SECTION

BI新范式的思考

数据架构正在发生的范式变革

Data Service	Headless BI(Semantic Layer) 统一数据口径	Chatbot BI 自然语言问答
Data Model	Streaming 实时数仓、批流一体	Data Build Tool 版本控制、代码复用、单元测试
Data Infra	Cloud-native 存算分离、按需计费	Lakehouse 湖仓一体、存储统一

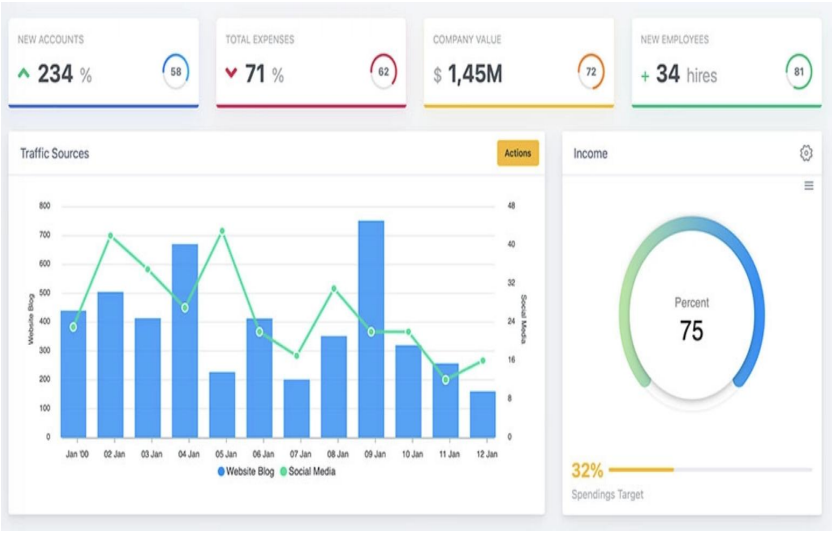
传统数据服务孵化出三类BI平台

形成产品矩阵，建设的先后顺序、使用占比因环境而异，没有统一标准

Dashboard

普适性：★ ★ ★ ★

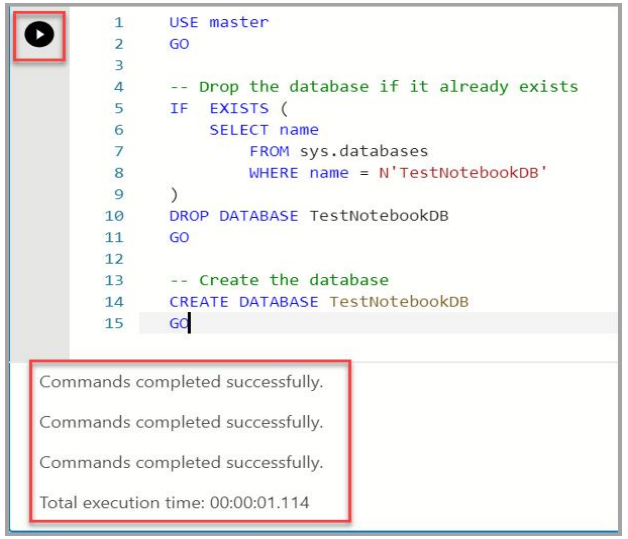
灵活性：★



Notebook

普适性：★

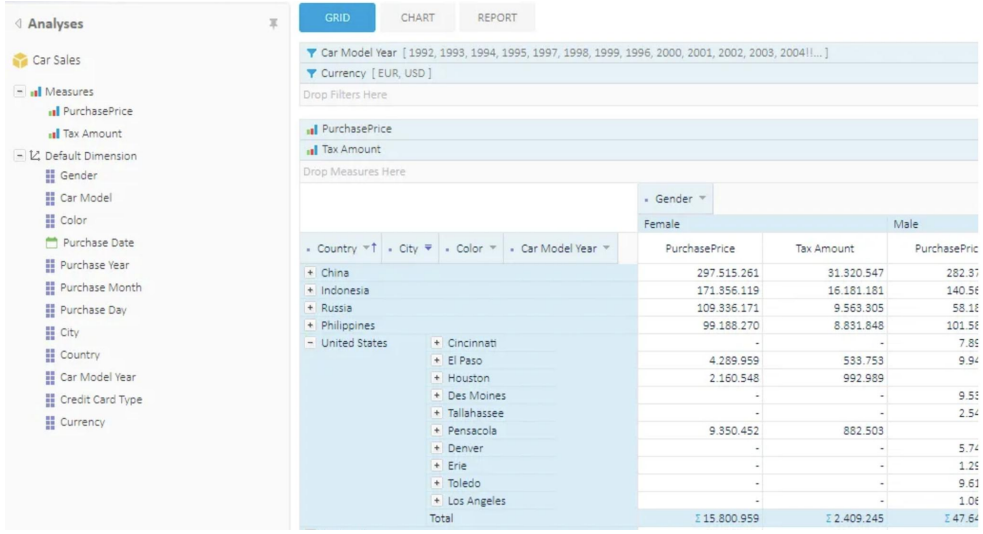
灵活性：★ ★ ★ ★



Drag & Drop

普适性：★ ★ ★

灵活性：★ ★



多BI平台并存所面临的难题

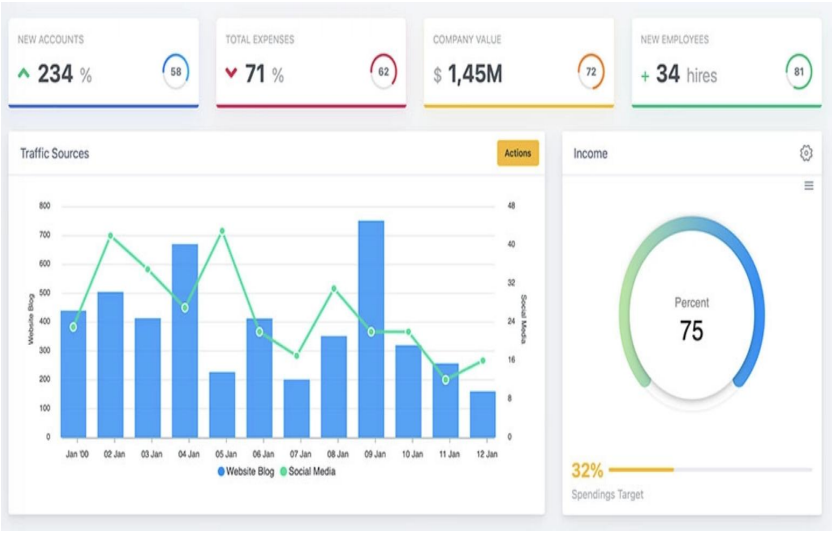
信任危机：相同语义的指标在不同的产品口径不一致

长尾困境：长尾场景同时需要较高的普适性和灵活性

Dashboard

普适性：★ ★ ★ ★

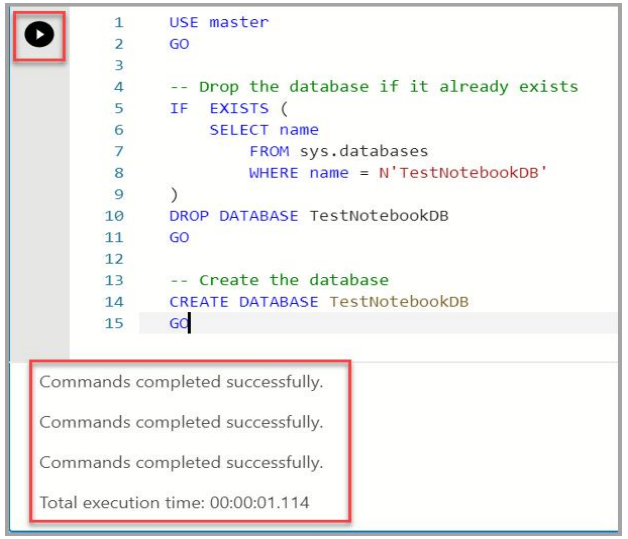
灵活性：★



Notebook

普适性：★

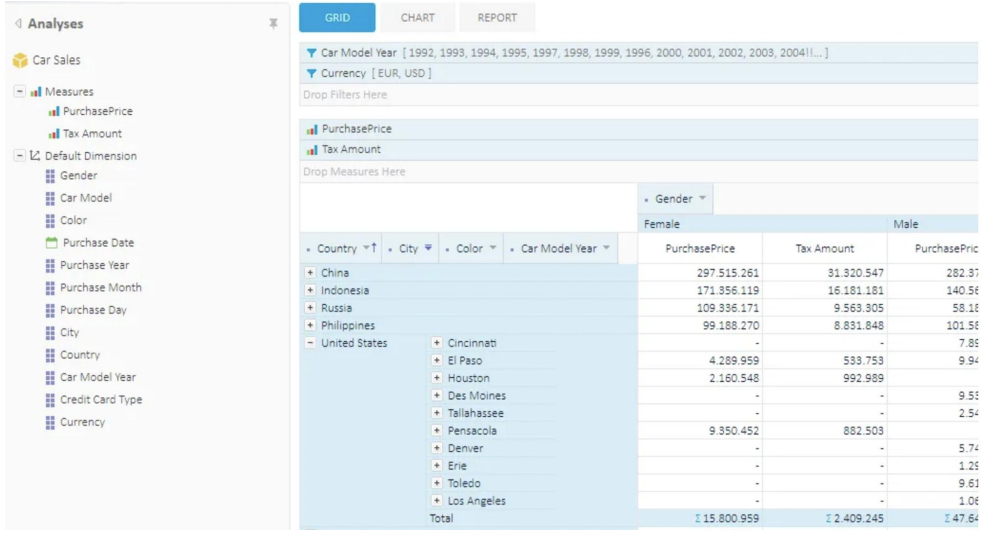
灵活性：★ ★ ★ ★



Drag & Drop

普适性：★ ★ ★

灵活性：★ ★



BI新范式入场解题

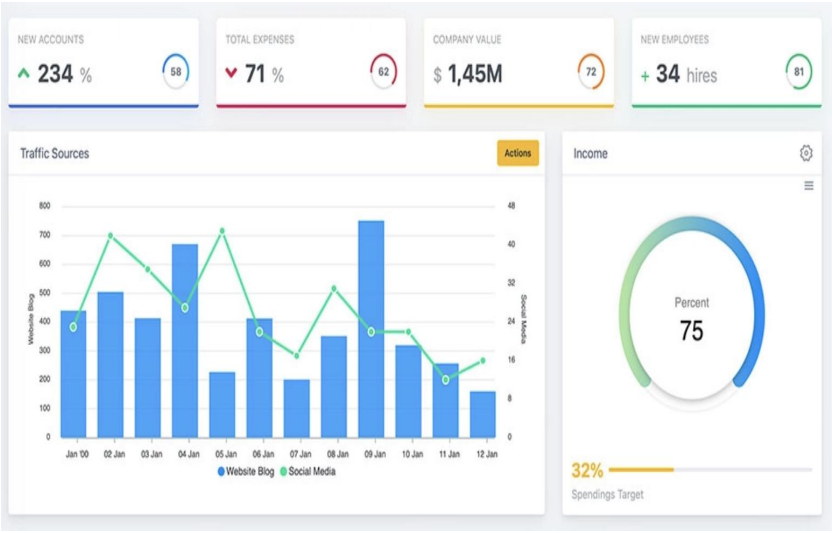
信任危机：相同语义的指标在不同产品口径存在差异 ⇨ **Headless BI**

长尾困境：长尾场景同时需要较高的普适性和灵活性 ⇨ **Chatbot BI**

Dashboard

普适性：★ ★ ★ ★

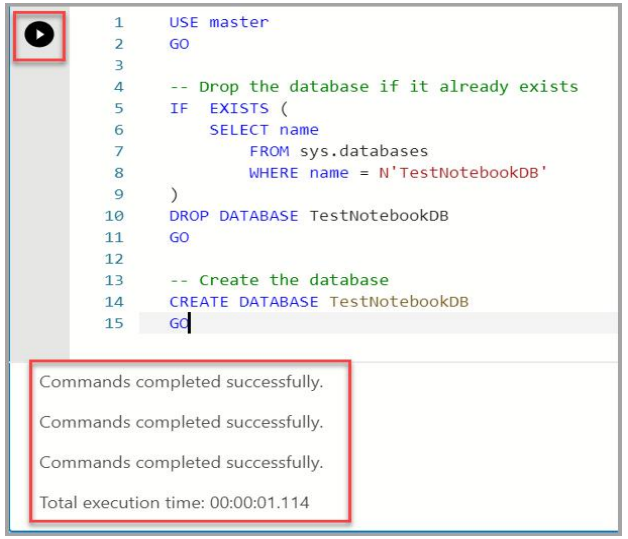
灵活性：★



Notebook

普适性：★

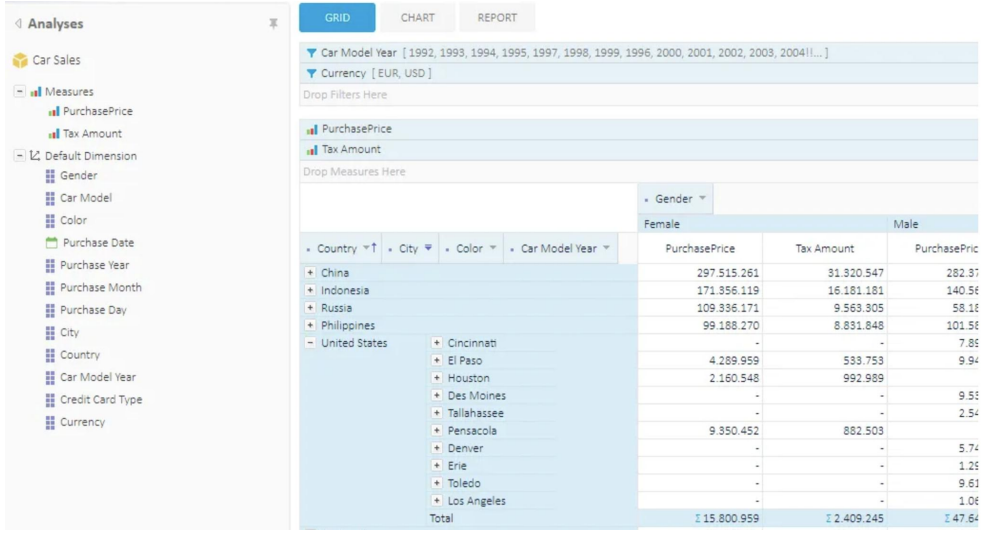
灵活性：★ ★ ★ ★



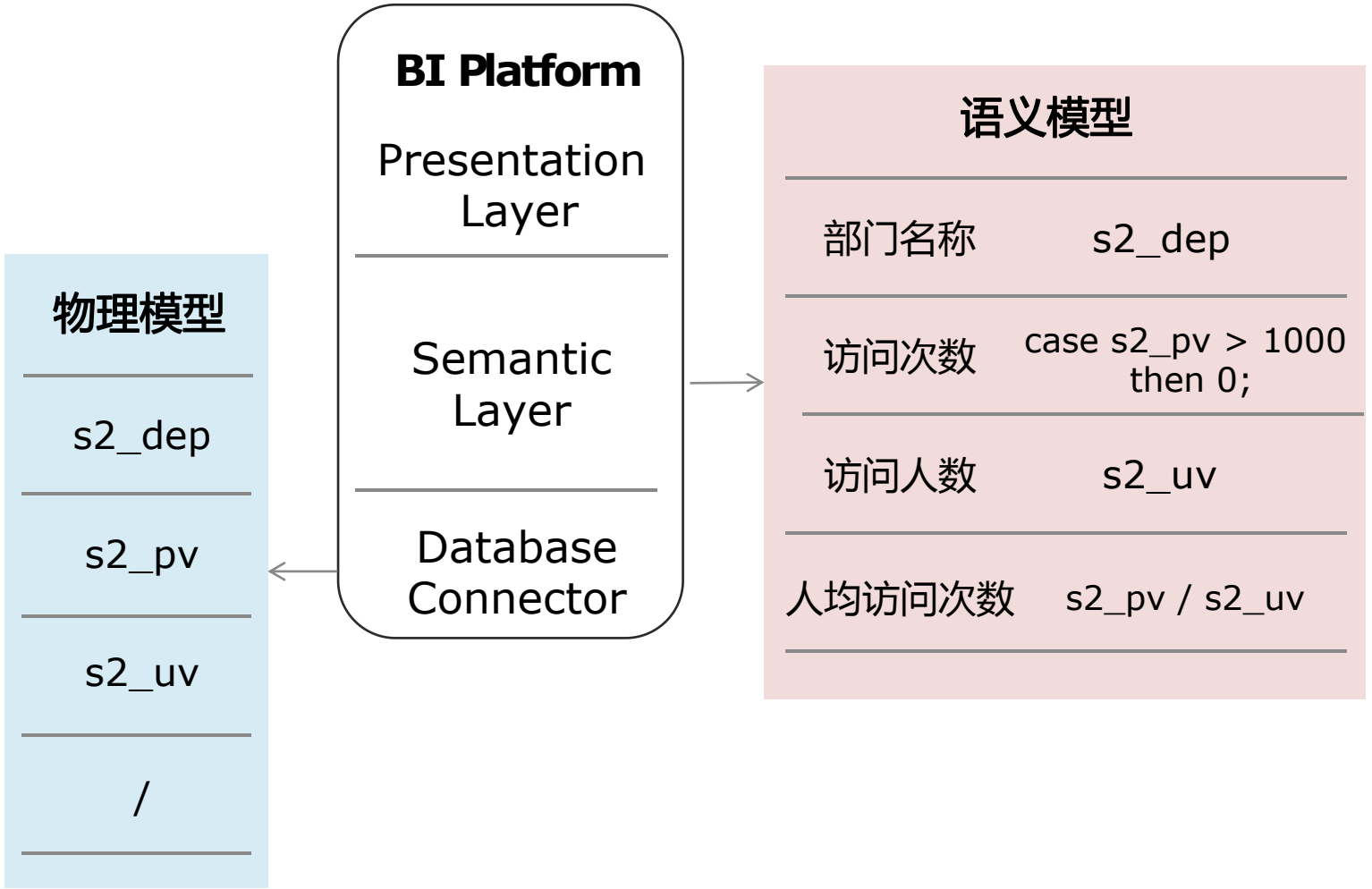
Drag & Drop

普适性：★ ★ ★

灵活性：★ ★



传统BI平台通常内置语义层（Semantic Layer）



语义层基于物理模型做二次建模

名词翻译，概念易懂

将技术名词（表名/列名）翻译成业务术语（维度名/指标名），以便于业务用户理解

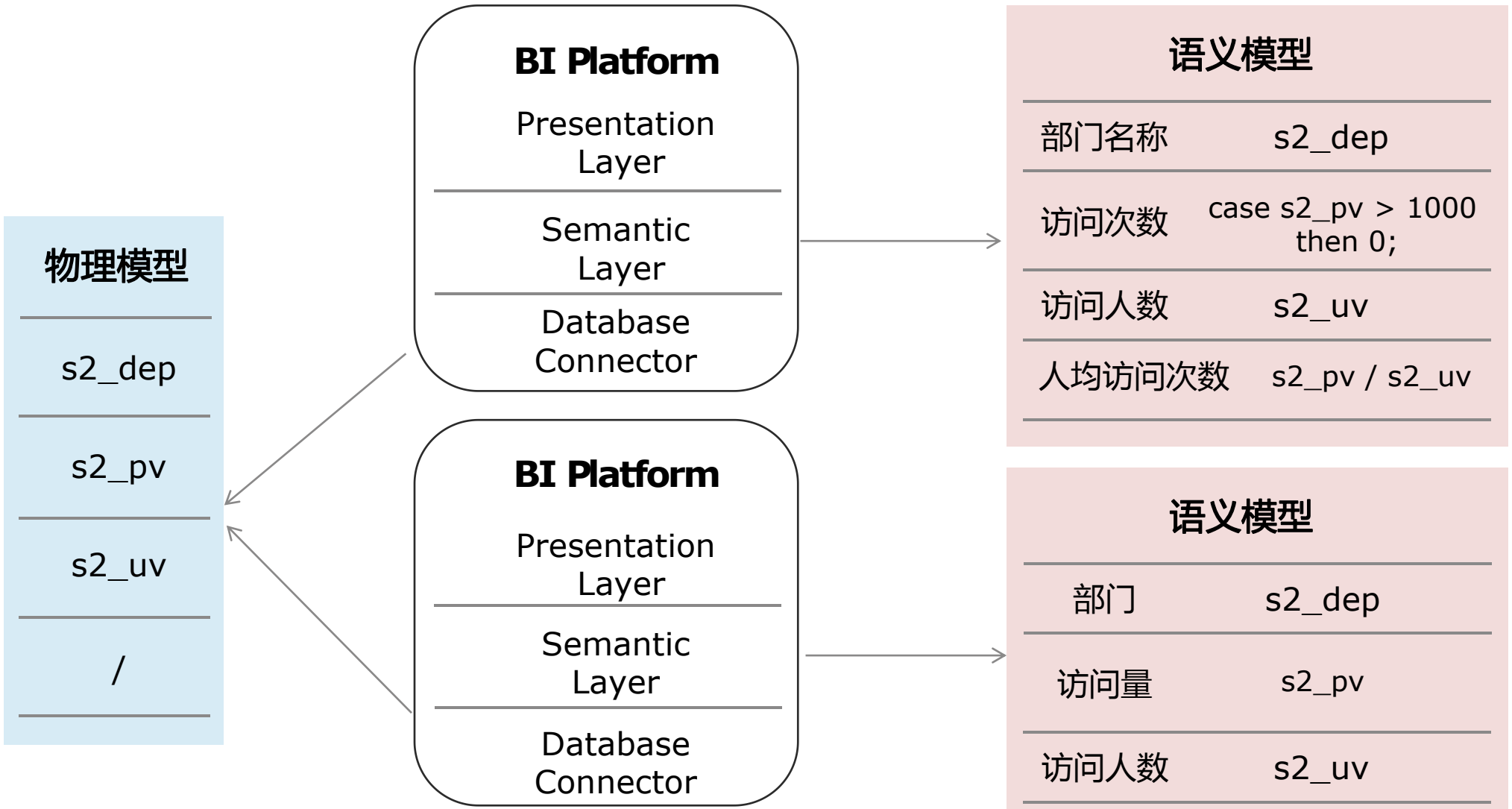
细节屏蔽，模型易用

将表间的关联关系、指标/维度的计算公式等技术细节封装，以便于业务直观使用

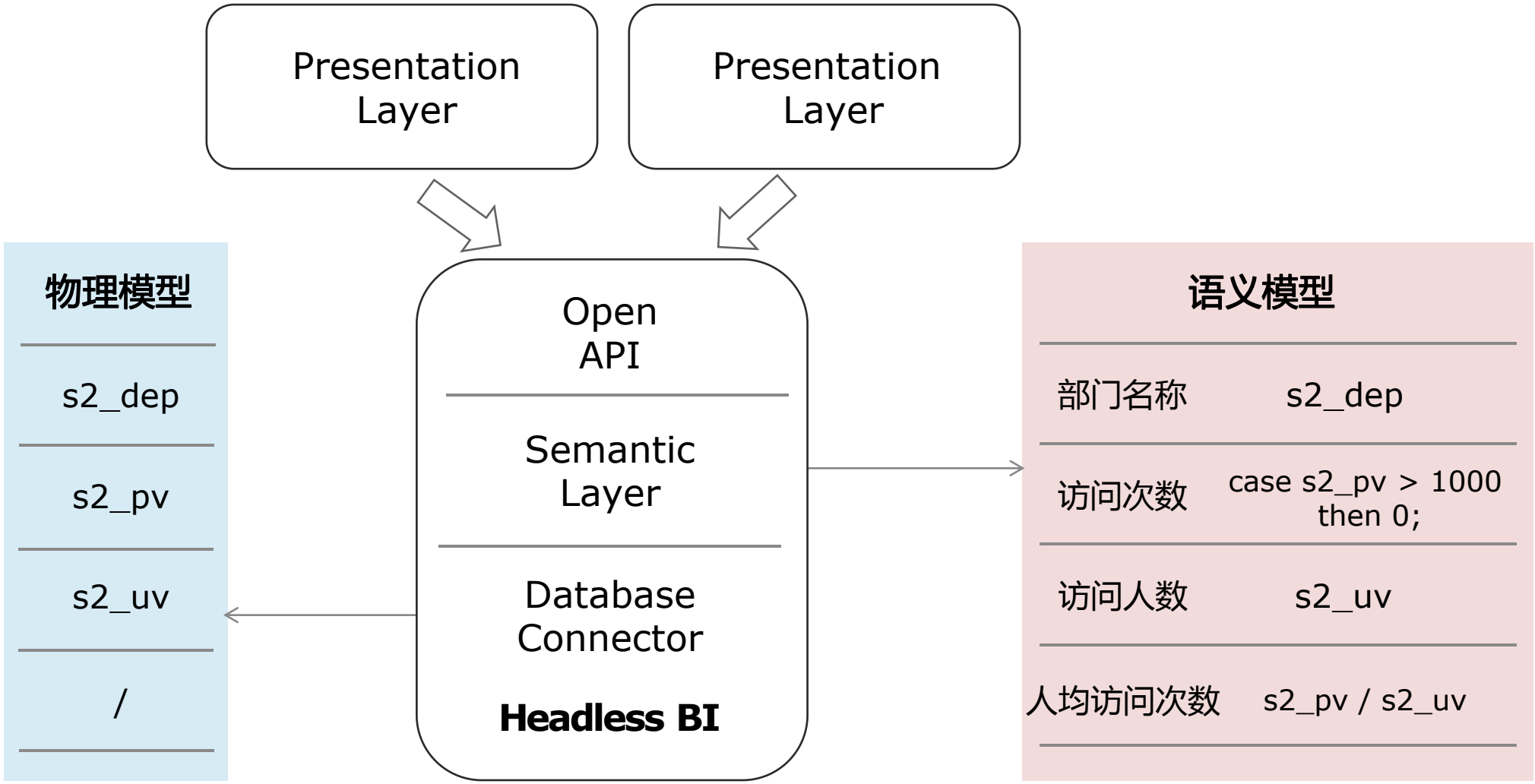
逻辑建模，口径易改

根据物理数据模型中的原子字段，定义出新的运算逻辑公式，以便于业务敏捷变更

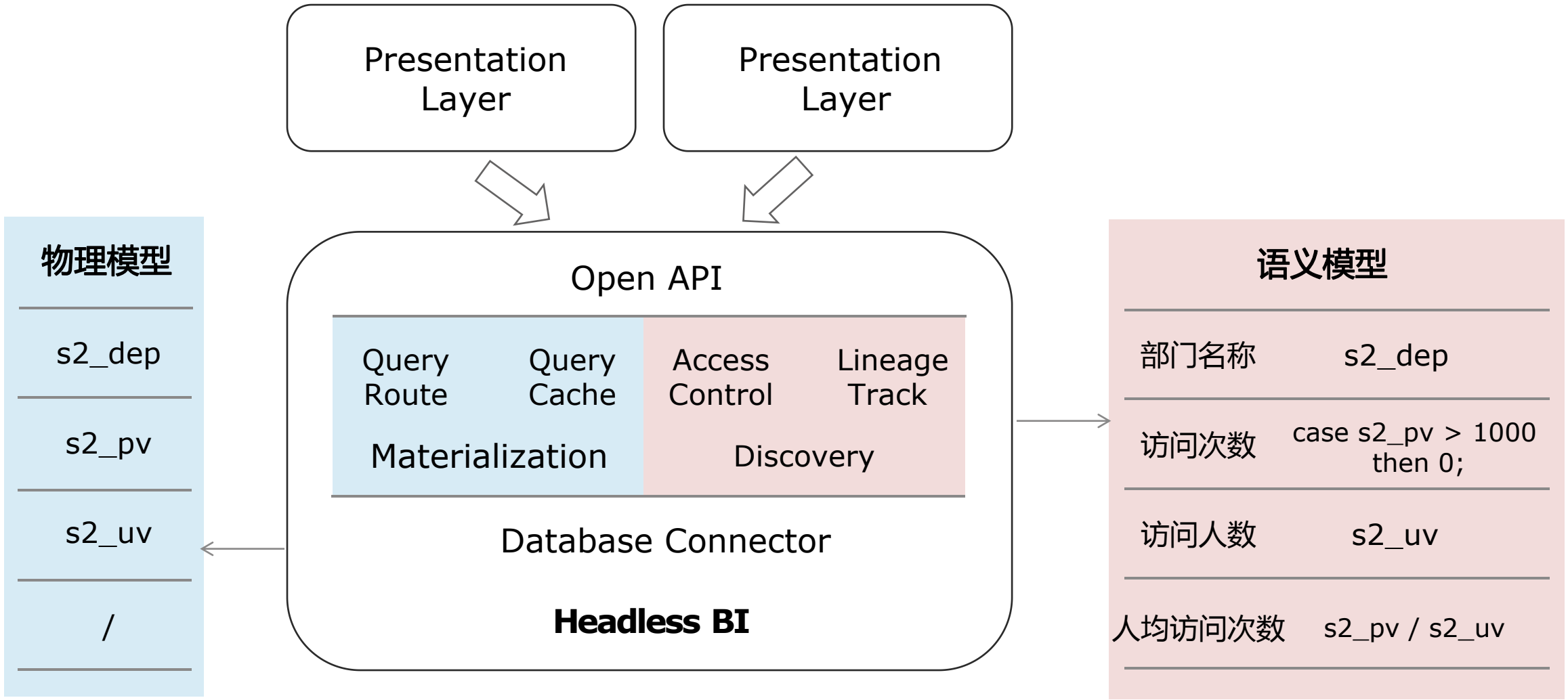
不同BI平台定义出不一致的语义模型



Headless BI的主旨：一次定义、多次复用，消除二义性



Headless BI的附加收益：集中优化、统一管理，降本增效



Headless BI选择开源项目还是自研

群雄争霸，尚未形成标准

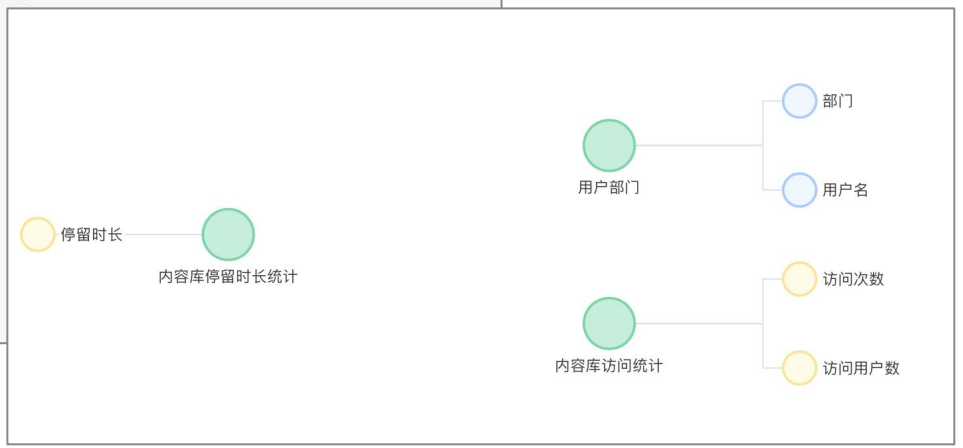
生态：dbt, looker, cube, malloy, etc

- 新的建模语言，有学习成本
- 新的查询接口，有迁移成本

选择自研，实现自主可控

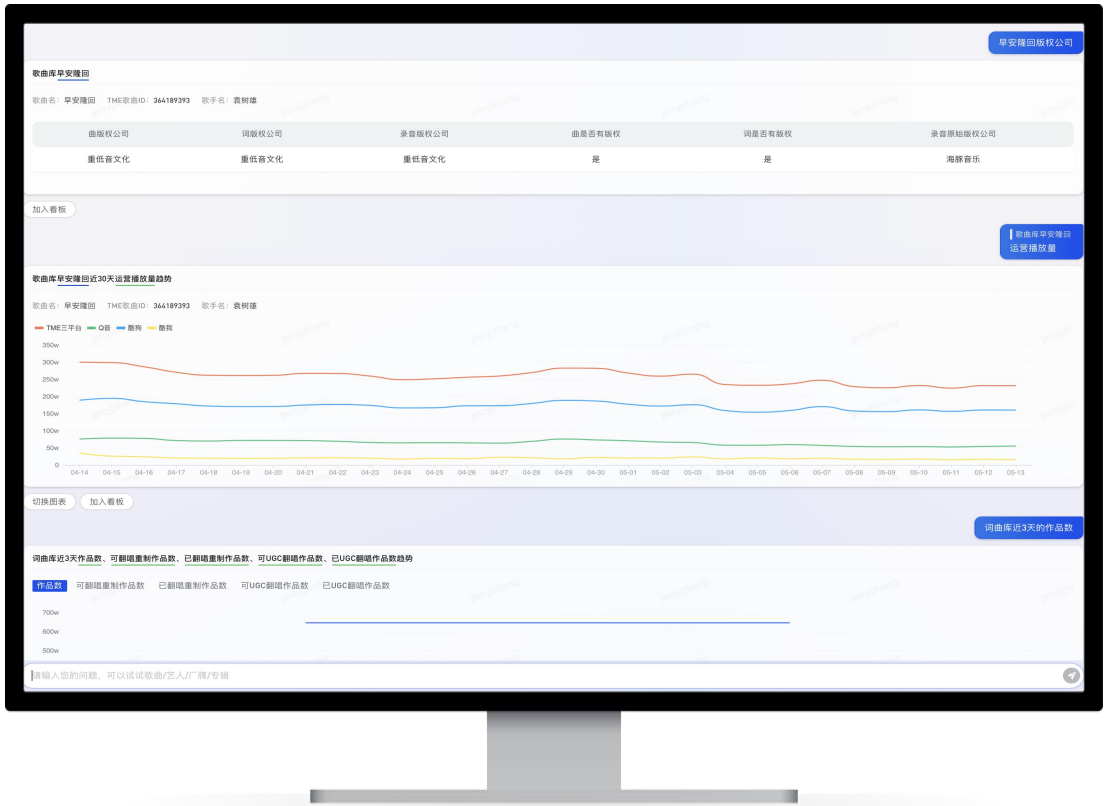
- Java开发，便于定制
- GUI交互，便于管理
- 协议自由，便于开源

```
cubes:
  - name: active_users
    sql: SELECT user_id, timestamp FROM events

    measures:
      - name: weekly_active
        sql: id
        type: count_distinct
        rolling_window:
          trailing: 7 day
          offset: start

    dimensions:
      - name: time
        type: time
        sql: timestamp
```



Chatbot BI的主旨：通过自然语言对话实现零门槛数据分析

兼具普适性和灵活性，PC和移动端体验一致



Chatbot BI如何落地的思考

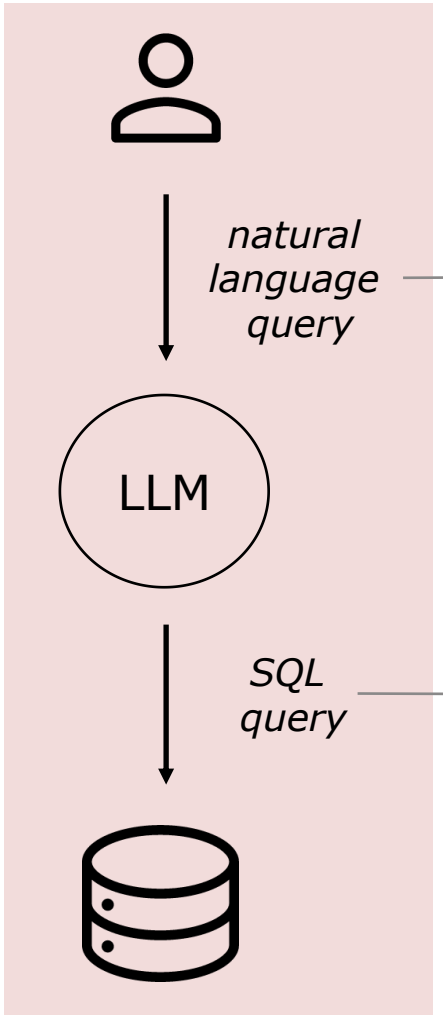
	半年前	现在	实践
Chatbot可以完全替代传统BI交互吗			Chat UI + Graphical UI
Chatbot可以直接通过LLM来实现吗			LLM + 配套工程化组件

02

SECTION

超音数建设实践

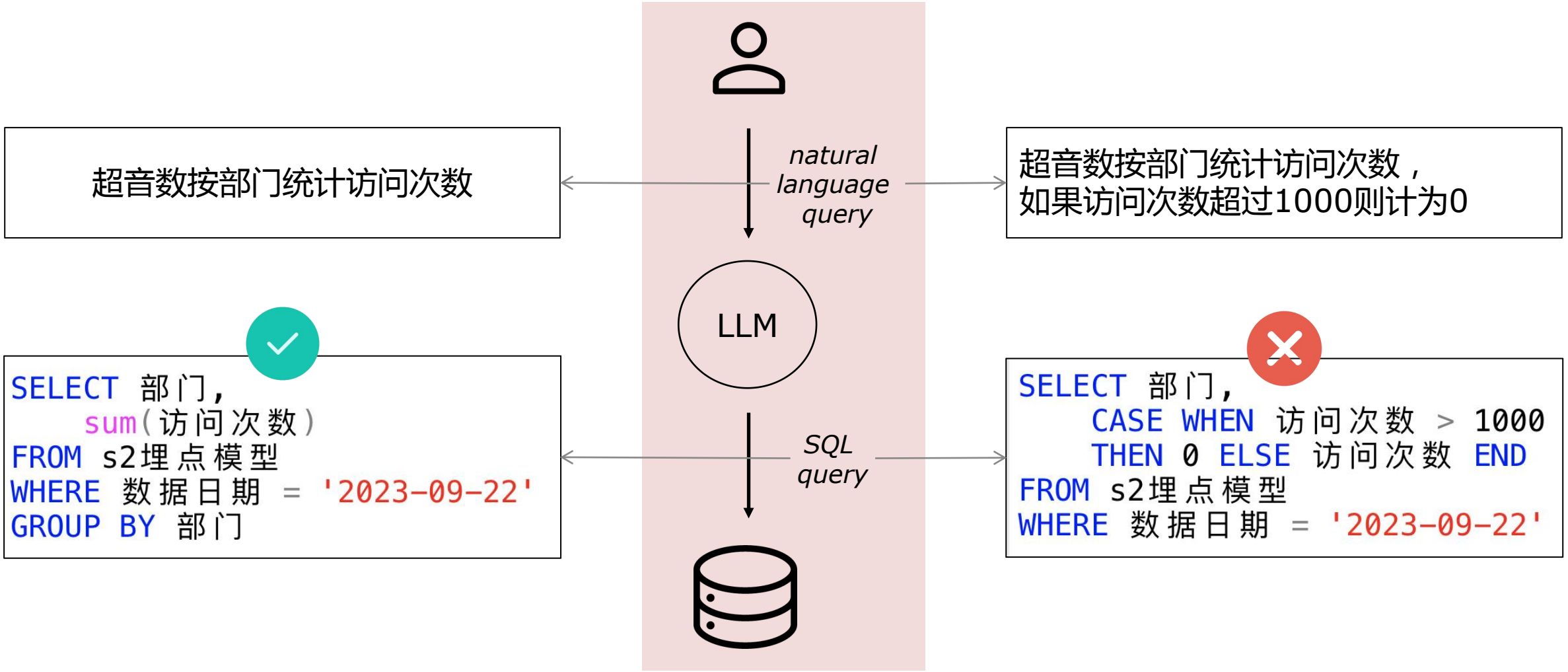
理想的实现方案：经过LLM直接实现Text2SQL



超音数按部门统计访问次数，且如果访问次数超过1000则计为0

```
SELECT
  u.dep,
  SUM (CASE
    WHEN s.pv > 1000 THEN 0
    ELSE s.pv)
FROM
  s2_log as s, user_dim as u
WHERE
  s.user_id = u.user_id
GROUP BY u.dep
```

LLM-based Text2SQL面临的主要挑战：稳定性



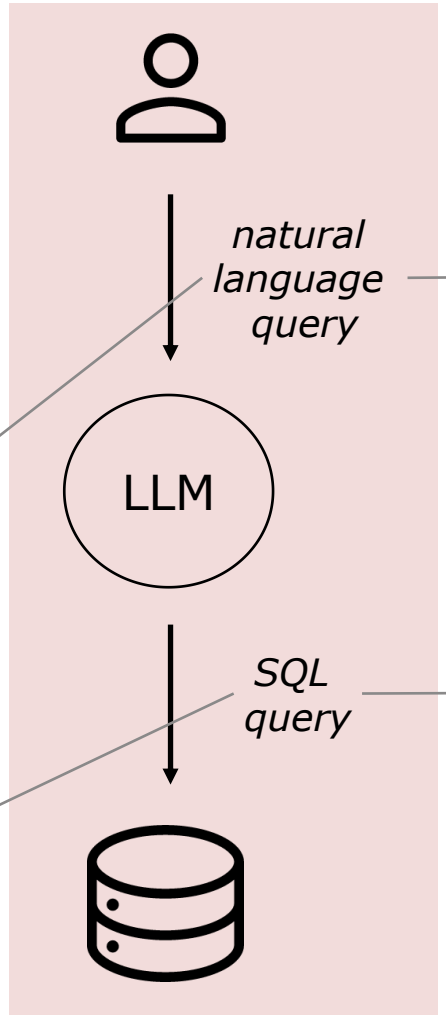
LLM-based Text2SQL面临的主要挑战：稳定性

语义复杂性导致不稳定

- 数据语义：多表关联、计算公式、时间函数、特殊方言等，依赖LLM生成能力
- 问询语义：通过自然语言表述来描述数据语义，依赖用户的表达能力

超音数按部门统计访问次数

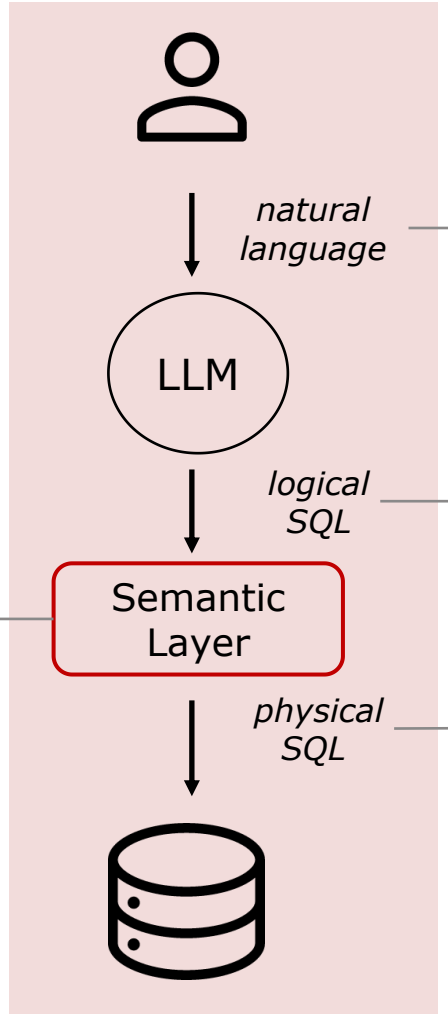
```
SELECT 部门,
    sum(访问次数)
FROM s2埋点模型
WHERE 数据日期 = '2023-09-22'
GROUP BY 部门
```



超音数按部门统计访问次数，
如果访问次数超过1000则计为0

```
SELECT 部门,
    CASE WHEN 访问次数 > 1000
    THEN 0 ELSE 访问次数 END
FROM s2埋点模型
WHERE 数据日期 = '2023-09-22'
```

引入翻译器：Semantic Layer



超音数按部门统计访问次数

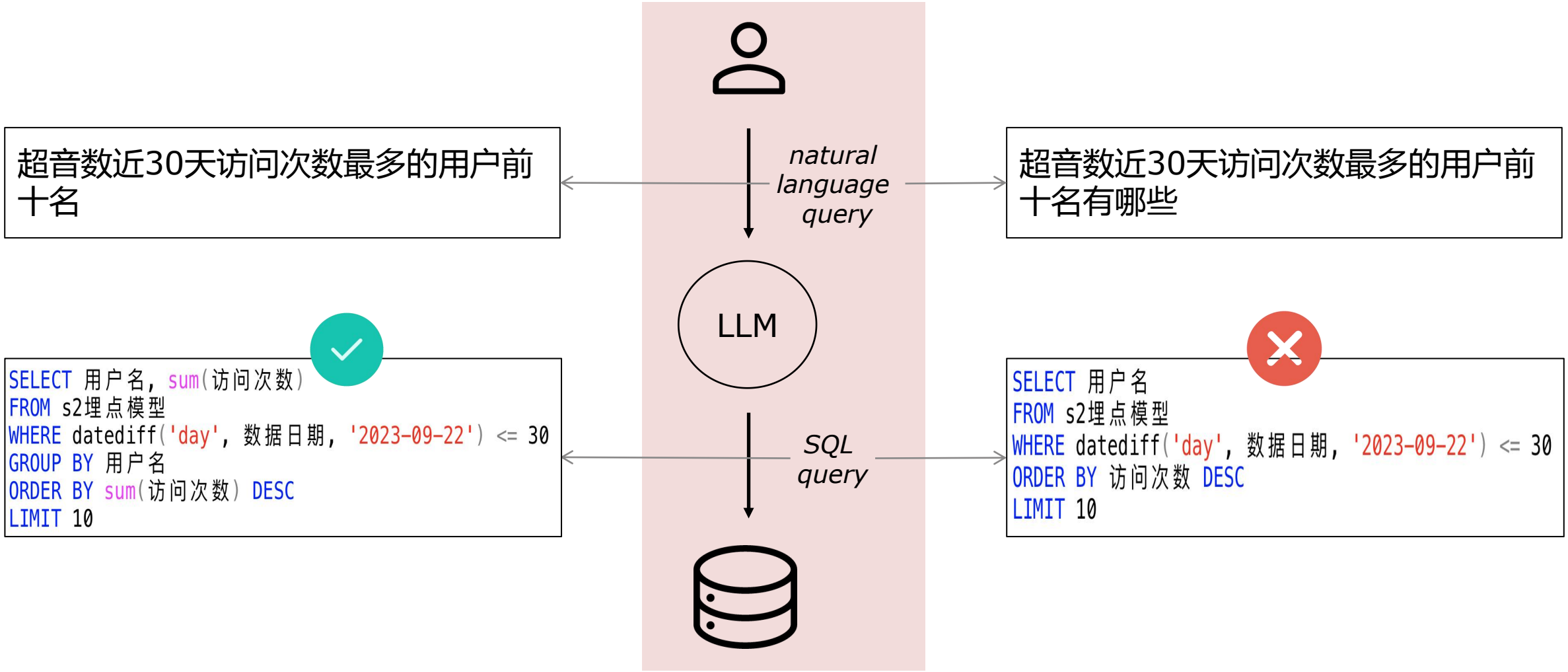
```
SELECT 部门, SUM(访问次数)
FROM s2埋点模型
GROUP BY 部门
```

```
SELECT
  u.dep,
  SUM (CASE
    WHEN s.pv > 1000 THEN 0
    ELSE s.pv)
FROM
  s2_log as s, user_dim as u
WHERE
  s.user_id = u.user_id
GROUP BY u.dep
```

封装复杂语义，化繁为简

- 多表关联、计算公式、时间函数、特殊方言由语义层处理，简化LLM代码生成
- 数据口径的定义都在语义层固化，简化用户的问询表达

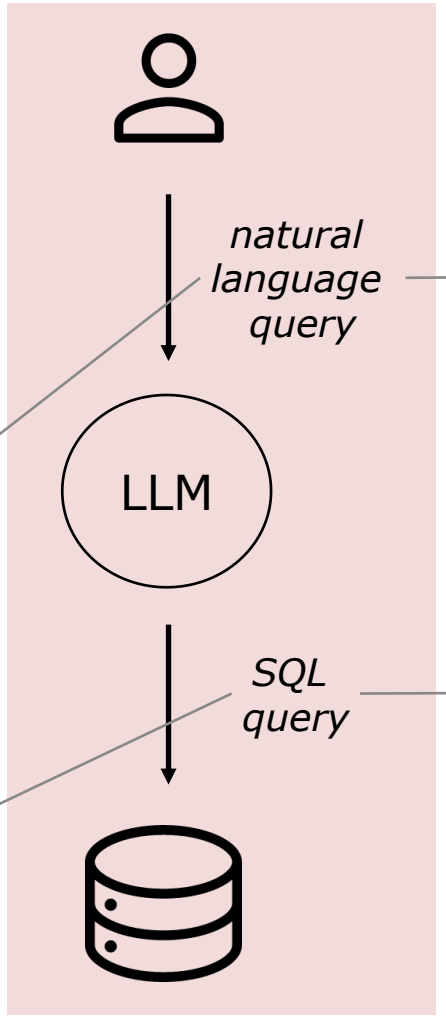
LLM-based Text2SQL面临的主要挑战：稳定性



LLM-based Text2SQL面临的主要挑战：稳定性

语言表达多样性导致不稳定

- 语义相同，但语言表达有细微差别，也可能导致完全不一样的结果



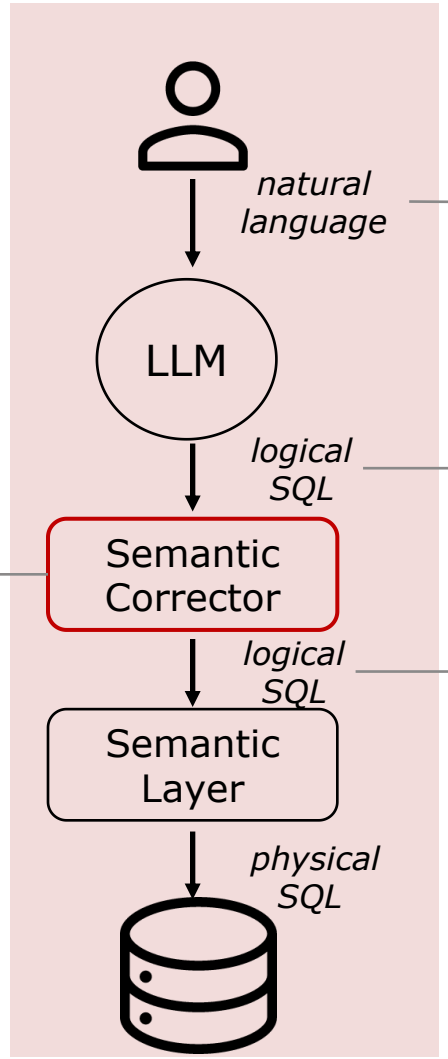
超音数近30天访问次数最多的用户前十名

```
SELECT 用户名, sum(访问次数)
FROM s2埋点模型
WHERE datediff('day', 数据日期, '2023-09-22') <= 30
GROUP BY 用户名
ORDER BY sum(访问次数) DESC
LIMIT 10
```

超音数近30天访问次数最多的用户前十名有哪些

```
SELECT 用户名
FROM s2埋点模型
WHERE datediff('day', 数据日期, '2023-09-22') <= 30
ORDER BY 访问次数 DESC
LIMIT 10
```

引入修正器：Semantic Corrector



超音数近30天访问次数最多的用户前十名有哪些

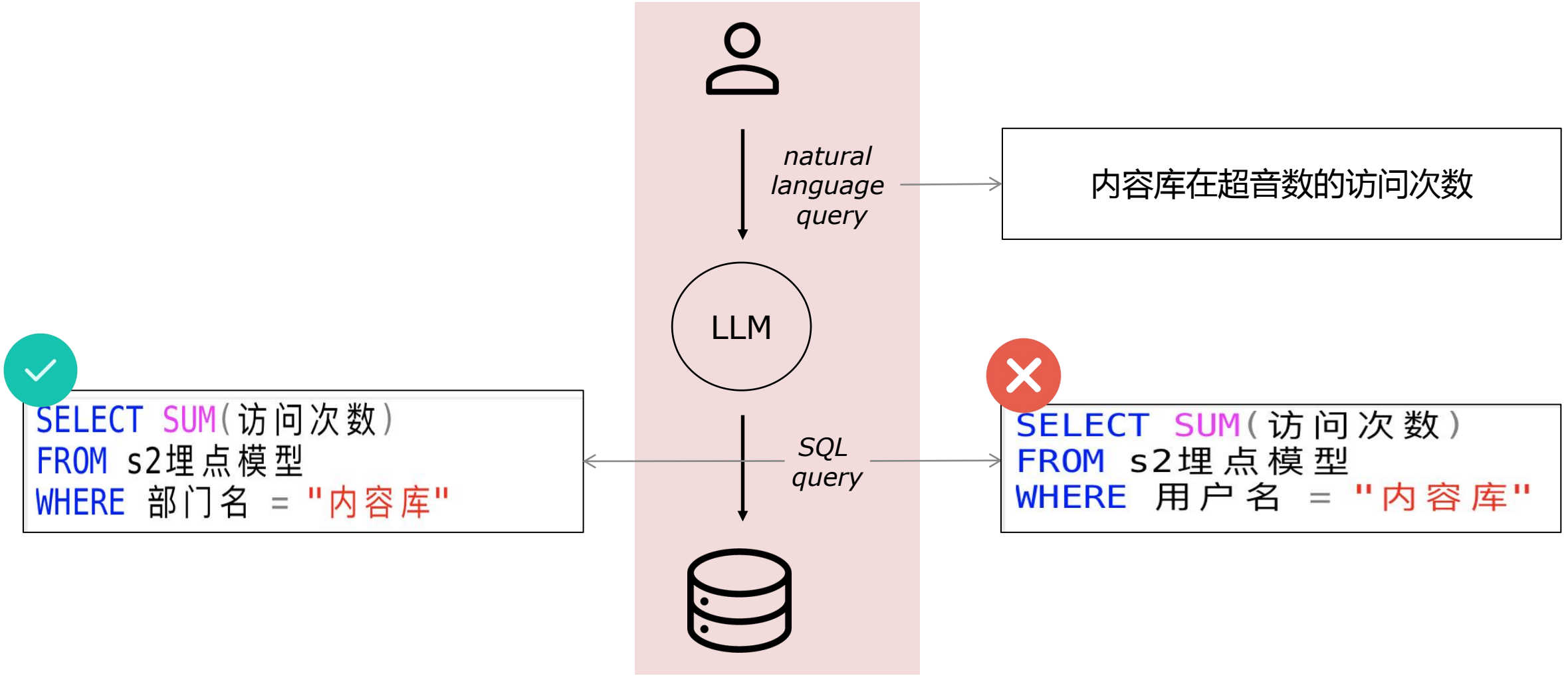
```
SELECT 用户名
FROM s2埋点模型
WHERE
  datediff('day', 数据日期, '2023-09-22') <= 30
ORDER BY 访问次数 desc
LIMIT 100
```

```
SELECT 用户名, SUM(访问次数)
FROM s2埋点模型
WHERE
  datediff('day', 数据日期, '2023-09-22') <= 30
GROUP BY 用户名
ORDER BY SUM(访问次数) desc
LIMIT 10
```

二次语义校验，尝试修正

- 字段名/表名是否存在
- 指标是否需要聚合/排序
- 维度是否需要分组
- 时间范围是否合法

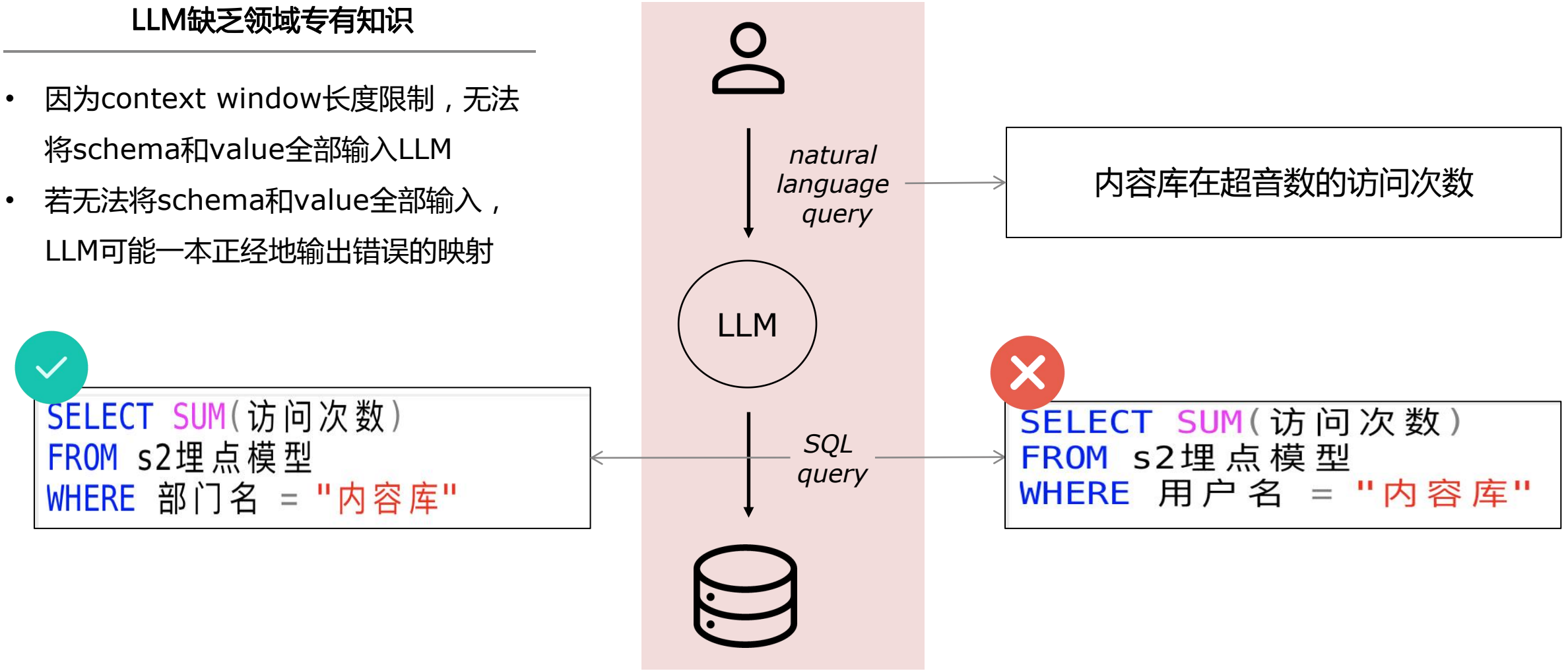
LLM-based Text2SQL面临的另一挑战：幻觉



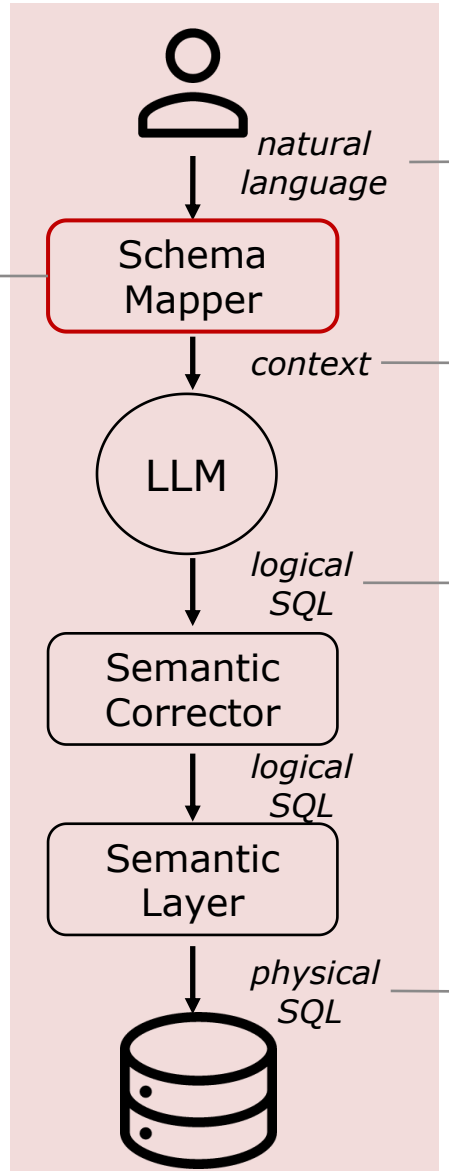
LLM-based Text2SQL面临的另一挑战：幻觉

LLM缺乏领域专有知识

- 因为context window长度限制，无法将schema和value全部输入LLM
- 若无法将schema和value全部输入，LLM可能一本正经地输出错误的映射



引入筛选器：Schema Mapper



只将相关的词语输入Context

- 文本处理：分词、n-gram探测
- 文本匹配：前缀、后缀、编辑距离
- TopN术语：指标、维度、取值

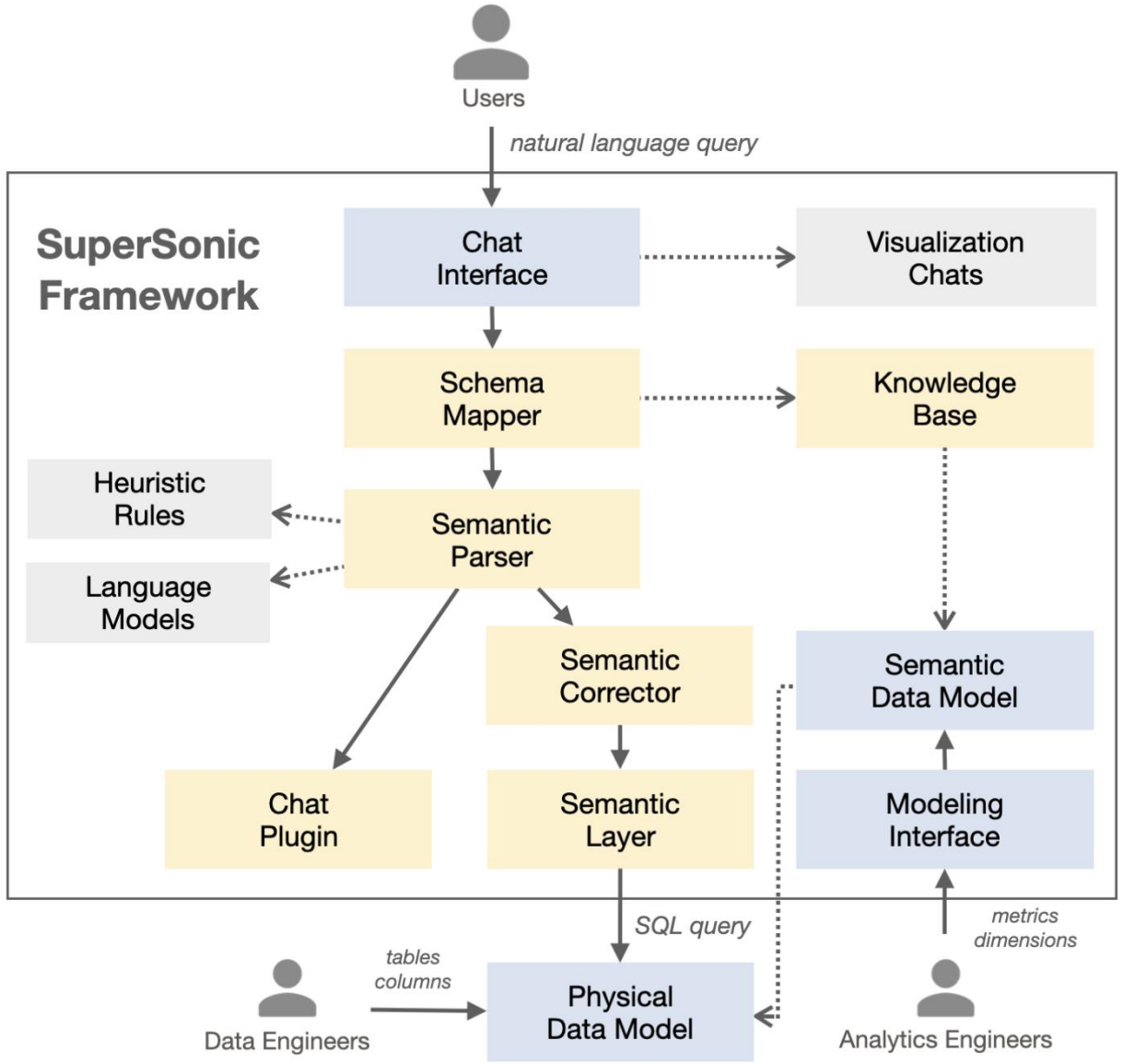
内容库在超音数的访问次数

维度::部门名 = 内容库
指标 = 访问次数

```
SELECT SUM(访问次数)
FROM s2埋点模型
WHERE 部门名 = "内容库"
```

```
SELECT
  SUM (CASE
    WHEN s.pv > 1000 THEN 0
    ELSE s.pv)
FROM
  s2_log as s, user_dim as u
WHERE
  s.user_id = u.user_id
  AND u.dep = '内容库'
```

超音数(SuperSonic) 开源框架

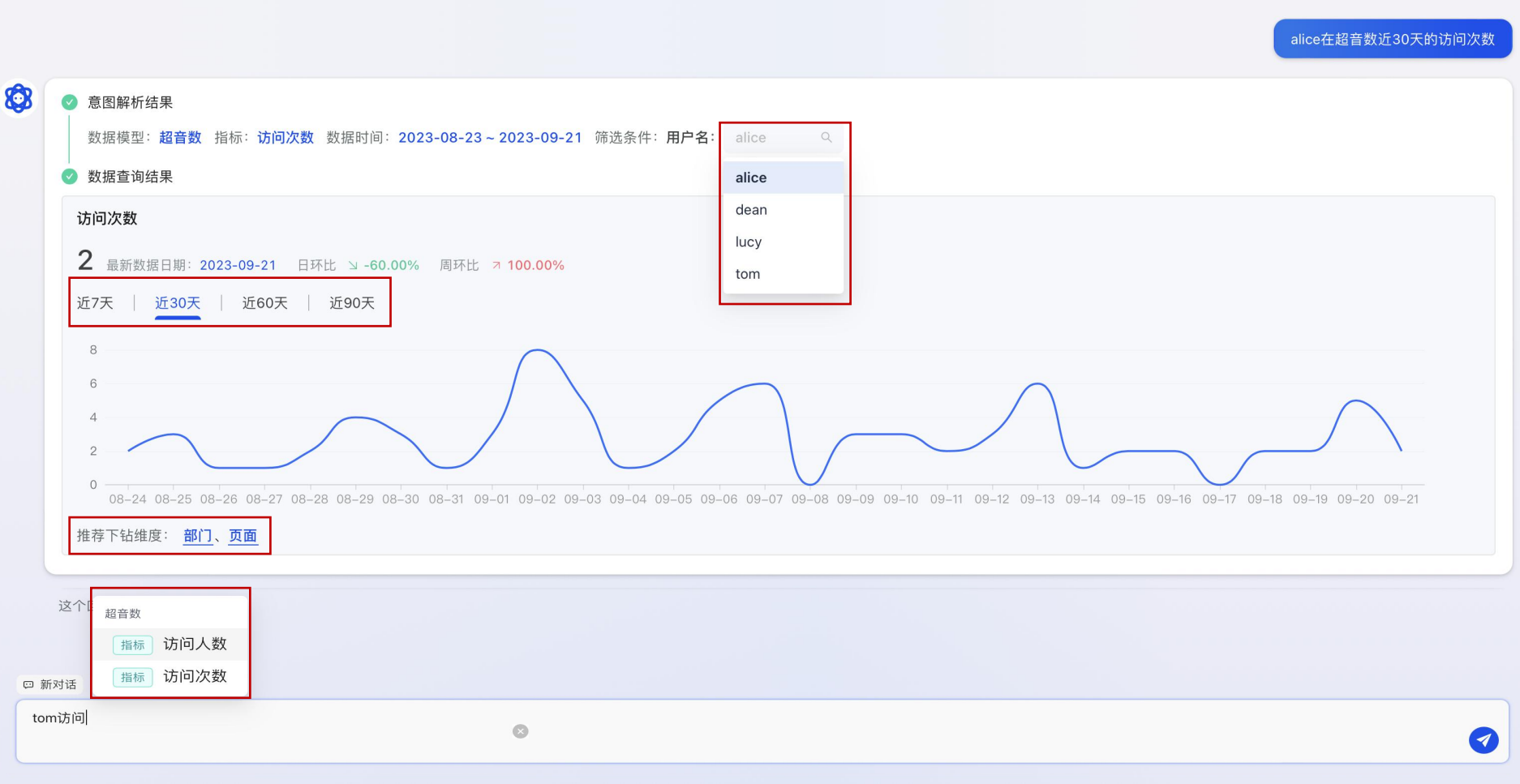


http://github.com/tencentmusic/supersonic

- **开箱即用：**“自然语言提问–可视化图表回答”简洁交互，“输入联想+多轮对话”轻松上手
- **简单上手：**无需修改和拷贝数据库中的物理模型，定义好语义模型即可开启问答
- **即插即用：**通过插件机制来做语义解析的增强或者定制功能的扩展

超音数交互设计：需要考虑固有的使用习惯和便捷性

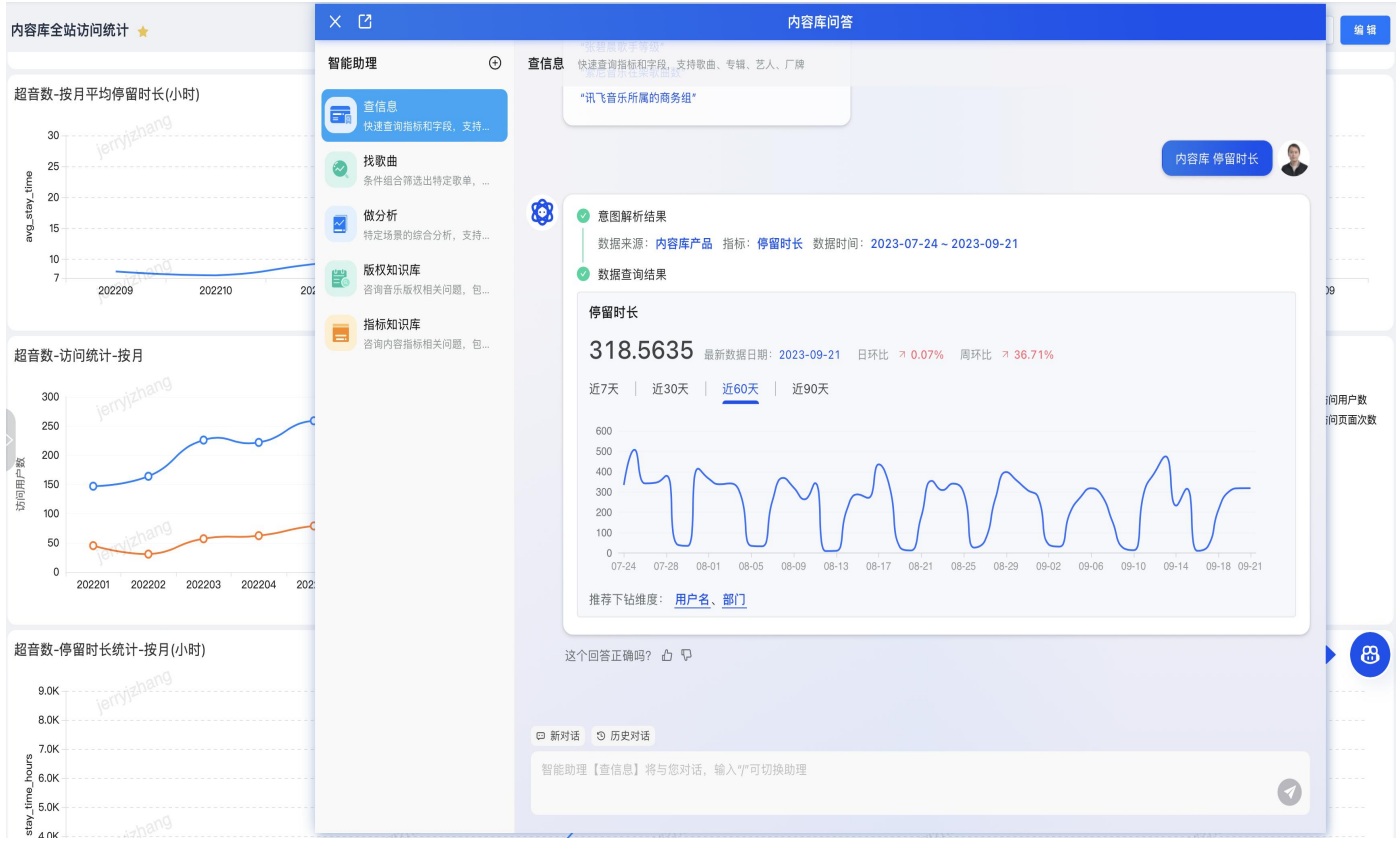
Chat UI + Graphical UI



超音数交互设计：如何与传统看板做融合

CUI作为主驾

CUI作为副驾



03

SECTION

未来规划与总结

未来规划

- 场景驱动，持续打磨数据问答的稳定性与复杂度
- 体验驱动，持续优化交互体验的便捷性与友好度
- 模型拓展，尝试引入更多的LLM
- 功能拓展，探索生成数据分析结论

总结

- Headless BI和Chatbot BI是数据服务的新范式
- Headless BI通过语义模型带来数据的易懂、易用、易改
- Chatbot BI通过工程化组件（筛选器、修正器、翻译器）的引入，可以一定程度解决LLM的稳定性和幻觉问题
- Chat UI需要与传统Graphical UI、Dashboard融合，取长补短，才能真正释放Chatbot BI的价值

2023CSDI 算力+智能

科 技 未 来 ： 数 字 时 代 的 进 化 升 级

thanks

CSDI SUMMIT