****

数据库原理实验报告

|  |  |
| --- | --- |
| 题 目： | 新型开源云原生数据库PolarDB实践 |
| 学生姓名： | 陈康坤 |
| 指导教师： | 赵颖 |
| 学 院： | 计算机学院 |
| 专业班级： | 计科2202 |

本科生院制

2024年5月10日

目录

[新型开源云原生数据库PolarDB实践 I](#_Toc21914)

[实验五 新型开源云原生数据库PolarDB实践 2](#_Toc3594)

[一、实验目的 4](#_Toc2465)

[二、实验内容 4](#_Toc17745)

[三、实验步骤及结果 5](#_Toc15208)

[四、实验中遇到的难题及解决办法 16](#_Toc9038)

[五、实验总结 16](#_Toc17956)

# 一、实验目的

了解国内自主研发的数据库

了解国内商业数据库应用现状

了解数据库的前沿技术

尝试使用云数据库及其相关的数据可视化与人工智能技术

# 二、实验内容

**（1）了解阿里云PolarDB基本特性**

具体资料请参看“实验五-附件1：阿里云PolarDB与云工开物计划.ppt”, “实验五-附件2：PolarDB快速入门操作指南.doc

**（2）创建数据库**

使用PolarDB创建一个学生选课数据库

**（3）创建四张数据表**

1． 学生表（xsb）包括如下字段：

学号（xh）：普通编码可变长字符型，10位长，主码

姓名（xm）：普通编码可变长字符型，20位长，非空

2． 课程表（kcb）包括如下字段：

课程号（kch）：普通编码可变长字符型，9位长，主码

课程名（kcm）：普通编码可变长字符型，20位长，非空

3． 学生选课表（xskcb）包括如下字段：

学号（xh）：主码

课程号（kch）：主码

分数（fs）：整型，取值范围为0-100，非空

其中：学号为引用学生表的外码，课程号为引用课程表的外码。

4. 补考表（bkb）包括如下字段：

序号（id）:整型，主码

是否通过（has\_pass）:tinyint,值为1代表通过，值为0代表未通过

学号（student\_id）: 普通编码可变长字符型，10位长

课程号（course\_id）: 普通编码可变长字符型，9位长

**（4）体验PolarDB特色功能**

**1. 对查询结果进行数据可视化**

DMS（Data Management Studio）提供了可视化模块，如柱形图、折线图、散点图等，让用户可以更容易地洞察数据之间的关系，比如分析趋势、增长对比等，进行相应的决策。在本次实验中，我们需要对SQL语句查询的查询结果进行数据可视化，观察可视化图表的效果。

**2. 对学生补考场景任务编排**

数据管理DMS的任务编排主要用于编排各类任务节点并进行周期或事件调度执行，通过创建一个或多个任务节点组成的任务流，实现复杂的任务调度，提高数据开发效率，其为一个自动化流程来确保数据处理、分析、备份等多种需求的顺序和正确性。在本次实验中，我们对学生补考场景进行任务编排，通过筛选补考表中补考通过的学生，修改其学生课程表中的成绩字段。

**3. PolarDB for** **AI**

PolarDB for AI是基于PolarDB MySQL版的一个数据库内的分布式机器学习组件，其内置了两大机器学习模型：通义千问大模型和NL2SQL大模型，通义千问是一个能理解和生成人类语言的AI模型，而NL2SQL能将自然语言转化为SQL语言。在本次实验中，我们可以通过扩展的SQL语句在PolarDB上使用通义千问大模型，对通义千问提出要求，让其完成我们传递给它的问题。此外，我们也会使用NL2SQL大模型，让它为我们生成问题的SQL语句，并与我们自己所写的语句进行对比。

# 三、实验步骤及结果

**（1）创建数据库**

登录PolarDB控制台,在控制台左上角，选择集群所在地域,找到目标集群，单击集群ID。在左侧导航栏中，单击配置与管理 > 数据库管理,单击创建数据库。数据库创建成功结果如下图所示：



**（2）使用DMS连接集群**

在基本信息页面右上角，单击登录数据库。在弹出的对话框中，输入PolarDB MySQL版集群中创建的数据库账号和数据库密码，单击登录。登录DMS后，在左侧导航栏的已登录实例列表中，单击目标集群名称，找到并双击目标数据库名称，即可切换到目标数据库进行管理。

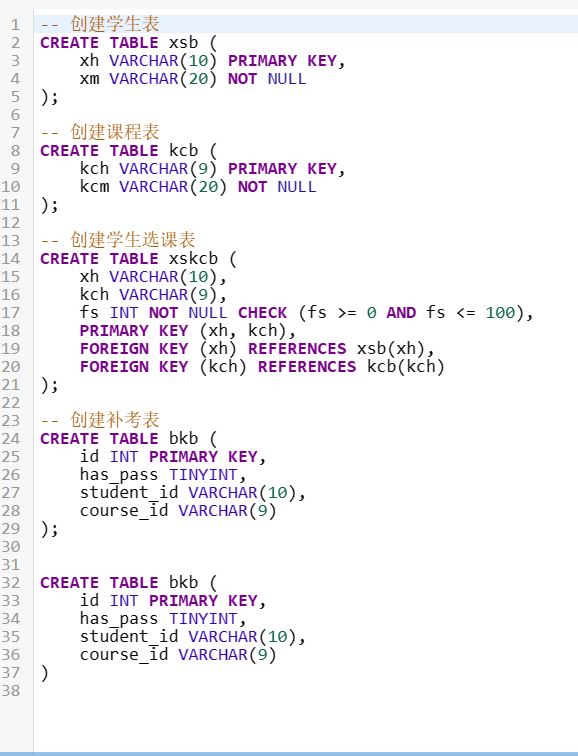
**（3）创建数据表和插入数据**

使用sql语句或者在表界面右键创建表，进行创建数据表，对学生表、课程表、学生课程表、补考表分别进行创建数据表操作。

在学生表（图1）、课程表（图2）、学生课程表（图3）分别插入以下数据：



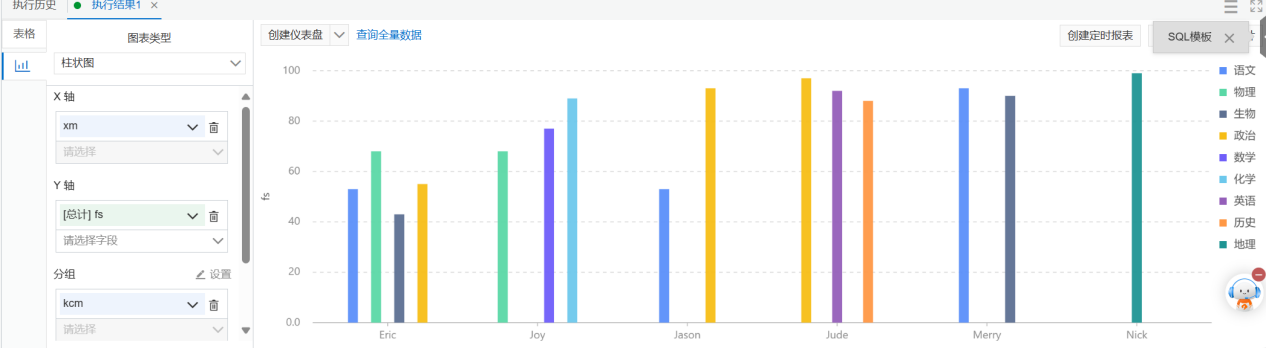
图1 图2 图3



**（4）对sql查询结果进行数据可视化**

练习1：用sql语句查询“各个学生所选的课程以及每门课程的成绩”。对查询结果创建柱形图，x轴选择学生的姓名，y轴选择课程的成绩，分组采用课程名，得到可视化的柱形图，如下图所示。

实验截图



实验代码

SELECT xsb.xh, xsb.xm, kcb.kch, kcb.kcm, xskcb.fs

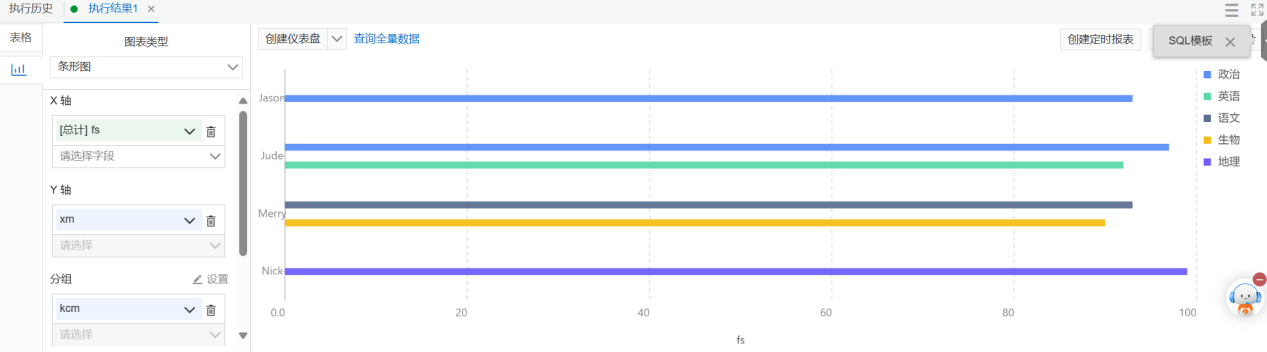
FROM xsb

INNER JOIN xskcb ON xsb.xh = xskcb.xh

INNER JOIN kcb ON xskcb.kch = kcb.kch;

练习2：查询“选择的课程成绩都大于等于90分的学生的学号和姓名、以及其选择的所有课程名和成绩”，自定义样式进行数据可视化。

实验截图



实验代码

SELECT xsb.xh, xsb.xm, kcb.kcm, xskcb.fs

FROM xsb

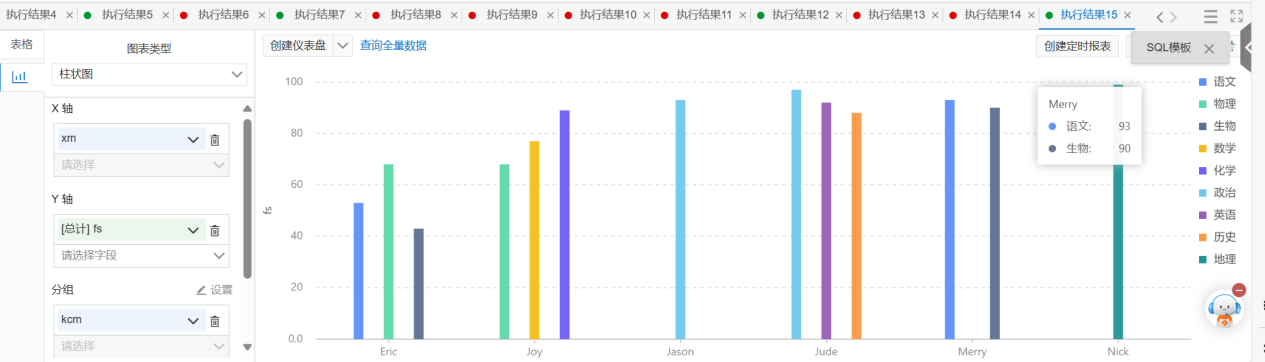
INNER JOIN xskcb ON xsb.xh = xskcb.xh

INNER JOIN kcb ON xskcb.kch = kcb.kch

WHERE xskcb.fs >= 90;

练习3：查询“每个课程分数最高的前二人的学号、姓名以及课程名、分数”，自定义样式进行数据可视化。

实验截图



实验代码

SELECT xskcb.xh, xsb.xm AS student\_name, kcb.kcm AS course\_name, xskcb.fs

FROM xskcb

INNER JOIN xsb ON xskcb.xh = xsb.xh

INNER JOIN kcb ON xskcb.kch = kcb.kch

WHERE (xskcb.xh, xskcb.kch) IN (

SELECT xh, kch

FROM (

SELECT xh, kch, fs,

ROW\_NUMBER() OVER (PARTITION BY kch ORDER BY fs DESC) AS row\_num

FROM xskcb

) AS ranked

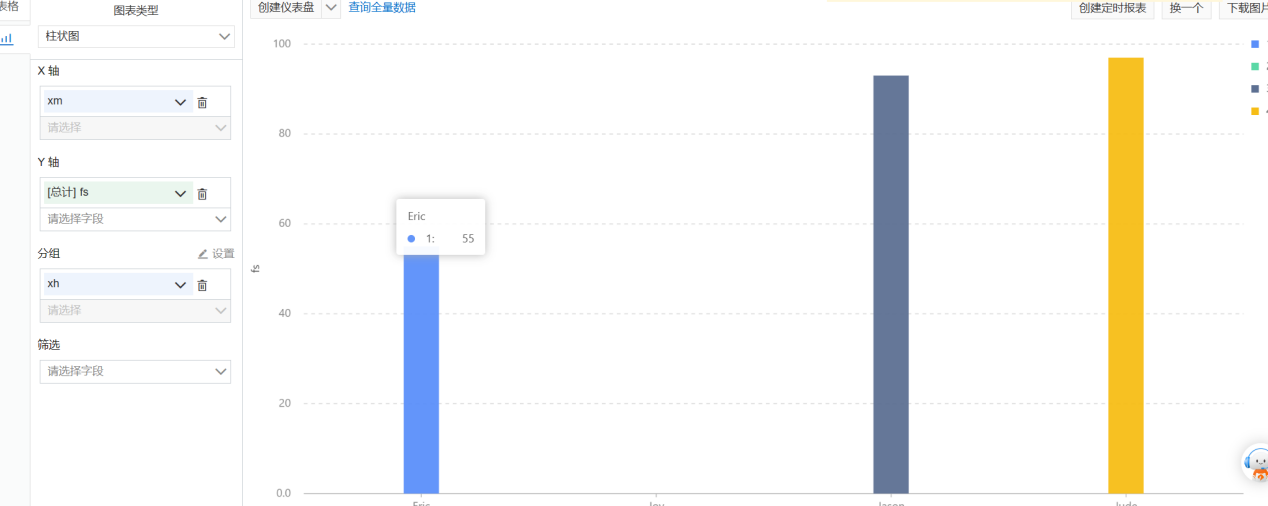
WHERE row\_num <= 2

);

练习4：请同学们自己尝试其他查询，并对查询结果进行可视化。

查询选择物理的学号、姓名以及分数

实验截图



实验代码

SELECT xsb.xh, xsb.xm, xskcb.fs

FROM xsb

INNER JOIN xskcb ON xsb.xh = xskcb.xh

INNER JOIN kcb ON xskcb.kch = kcb.kch

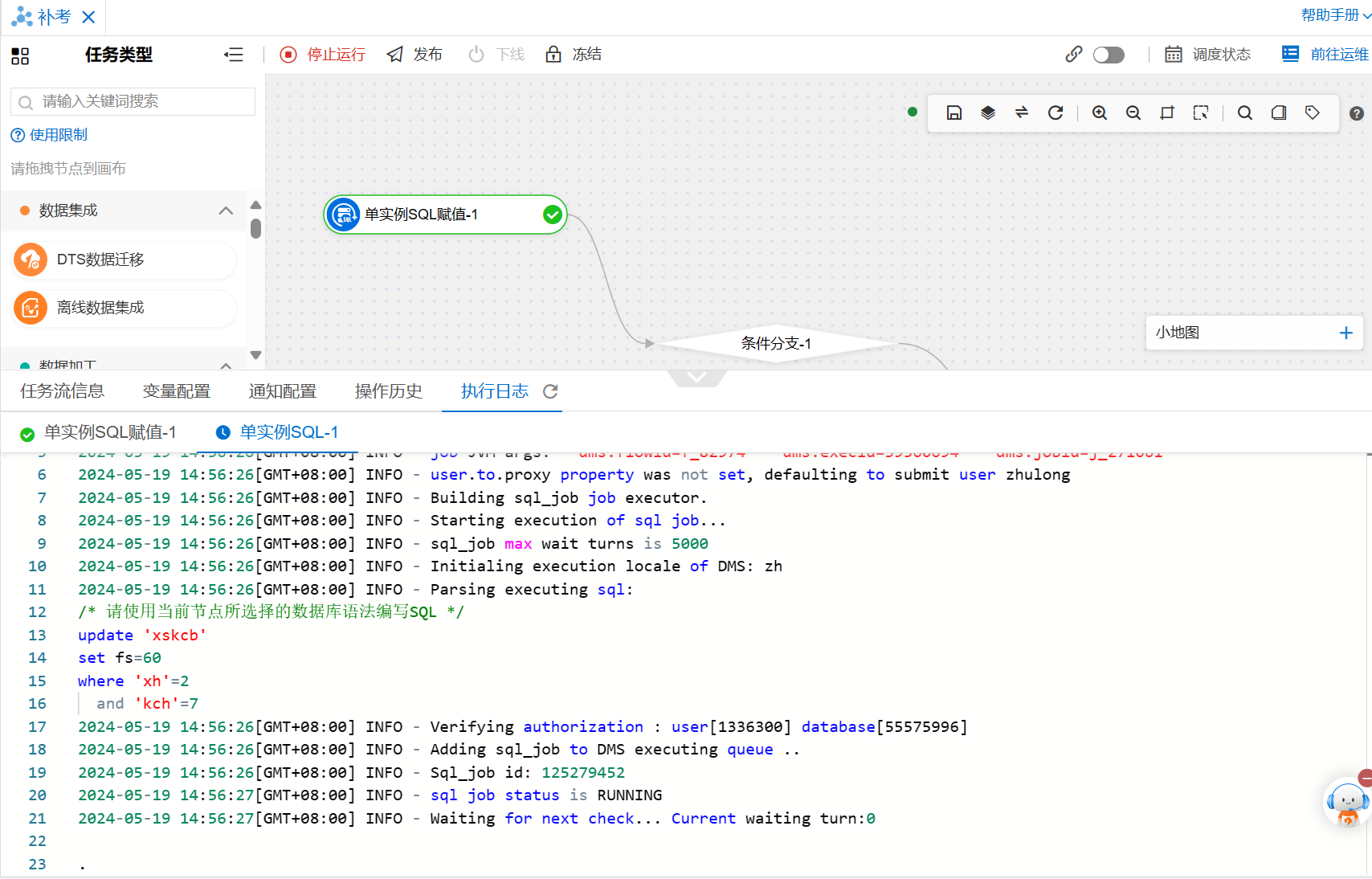
WHERE kcb.kcm = '政治';

**（5）任务编排**

任务编排是通过创建一个或多个任务节点组成的任务流，来实现复杂的任务调度。

本次实验中我们将使用到单实例SQL赋值、条件分支、单实例SQL三种任务节点。

实验要求创建一个任务流，这个任务流的功能为：依据补考表学生是否通过补考，来修改学生课程表中成绩字段，修改的成绩要从原本的不及格的分数变为60分。



**（6）开启PolarDB for AI功能**

* **创建普通权限账号**

在集群页面，左侧导航栏，单击配置与管理>账号管理，创建一个普通权限的账号



* **开通PolarDB for AI**

单击配置与管理>NL2SQL，在PolarDB NL2SQL页面，单击开通正式版本。单击+增加一个db4ai节点，选中服务协议，单击立即购买（可使用代金券免费购买）。

* **增加AI节点**

在集群基本信息页，等待节点新增成功后，选中节点，单击设置连接数据库账号，即可体验PolarDB for AI功能。



**（7）使用通义千问模型**

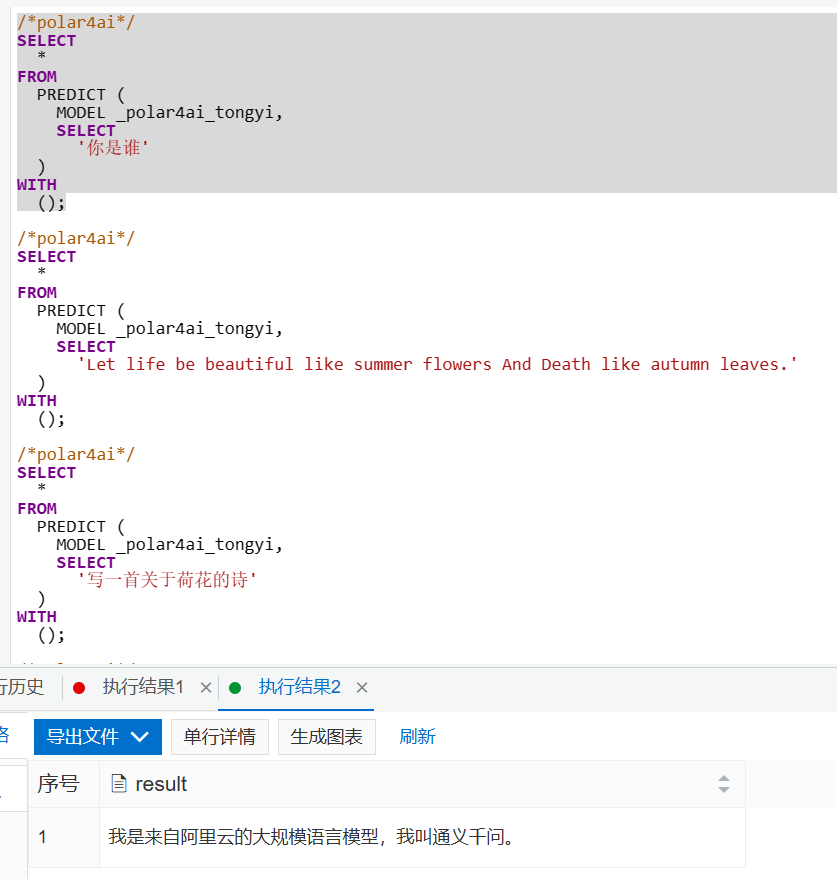
通义千问是一个能生成人类语言的ai模型，我们可以使用sql语句来对其进行提问，通

义千问模型就会对我们的问题进行回答，并将回答作为结果返回。

例如我们使用sql语句“/polar4ai/SELECT FROM PREDICT (MODEL \_polar4ai\_tongyi, SELECT '你是谁') with ();”对通义千问模型进行提问，其便会回答“我是来自阿里云的大规模语言模型，我叫通义千问。”

练习1：编写sql语句，使得通义千问模型将问题“Let life be beautiful like summer flowers And Death like autumn leaves.”翻译为中文

实验截图



实验代码

/polar4ai/

SELECT

FROM

PREDICT (

MODEL \_polar4ai\_tongyi,

SELECT

'你是谁'

)

WITH

();

练习2：编写sql语句，让通义千问模型写一首关于荷花的诗

实验截图



实验代码

/polar4ai/

SELECT

FROM

PREDICT (

MODEL \_polar4ai\_tongyi,

SELECT

'写一首关于荷花的诗'

)

WITH

();

练习3：通义千问模型依据MODEL后面的模型名称\_polar4ai\_tongyi的不同，有着不同的作用，如\_polar4ai\_tongyi\_sa为情感分析，\_polar4ai\_tongyi\_summarize为总结，请同学们使用的不同的模型名称参数进行体验

实验截图



实验代码

/polar4ai/

SELECT

FROM

PREDICT (

MODEL \_polar4ai\_tongyi\_sa,

SELECT

'我刚分手了'

)

WITH

();

**（8）使用NL2SQL模型**

NL2SQL是一种能将自然语言转化为SQL语句的模型，我们可以使用扩展的SQL语句向NL2SQL模型提出问题，NL2SQL便会依据问题生成对应的SQL语句。

* **NL2SQL的数据准备**

在使用NL2SQL模型之前，我们需要进行构建相关的数据表进行数据准备

1. 构建数据表的检索索引表

SQL语句如下：

*/polar4ai/*CREATE TABLE schema\_index(id integer, table\_name varchar, table\_comment text\_ik\_max\_word, table\_ddl text\_ik\_max\_word, column\_names text\_ik\_max\_word, column\_comments text\_ik\_max\_word, sample\_values text\_ik\_max\_word, vecs vector\_768,ext text\_ik\_max\_word, primary key (id));

1. 将数据表中的信息导入检索索引表

SQL语句如下：

*/polar4ai/*SELECT FROM PREDICT (MODEL \_polar4ai\_text2vec, select '') WITH (mode='async', resource='schema') into schema\_index;

* **使用NL2SQL模型**

请你尝试使用NL2SQL模型来对问题“id为1的内容是什么”进行回答，使模型返回相应的SQL语句。

实验截图



实验代码

/polar4ai/

CREATE TABLE

schema\_index (

id integer,

table\_name varchar,

table\_comment text\_ik\_max\_word,

table\_ddl text\_ik\_max\_word,

column\_names text\_ik\_max\_word,

column\_comments text\_ik\_max\_word,

sample\_values text\_ik\_max\_word,

vecs vector\_768,

ext text\_ik\_max\_word,

primary key (id)

);

/polar4ai/SELECT FROM PREDICT (MODEL \_polar4ai\_text2vec, select '') WITH (mode='async', resource='schema') into schema\_index;

/polar4ai/SELECT FROM PREDICT(MODEL \_polar4ai\_nl2sql,select 'id为1的内容是什么')

with (basic\_index\_name='schema\_index');

/polar4ai/

SELECT

FROM

PREDICT (

MODEL \_polar4ai\_nl2sql,

select

'查询数据不为空的内容'

)

with

(basic\_index\_name = 'schema\_index');

# 四、实验中遇到的难题及解决办法

. 问题一：SQL语句创建表时出错

问题描述

在使用SQL语句创建数据表时，遇到语法错误或约束冲突，导致表无法创建。

解决办法

1. 检查语法：仔细检查SQL语句的语法是否正确，包括字段类型、约束条件等。例如，确保使用正确的数据类型和长度。

2. 参考示例：参考PolarDB提供的示例SQL语句，确保格式和语法正确无误。

3. 简化测试：逐步简化SQL语句，逐行测试，找出出错的具体部分，然后逐步添加其他部分，确保每一步都正确。

4. 使用界面创建：如果SQL语句问题难以解决，可以尝试使用PolarDB的图形化界面进行创建操作。

问题二：数据可视化图表显示异常

问题描述

在DMS中对查询结果进行数据可视化时，生成的图表显示异常，如图表空白或数据不匹配。

解决办法

1. 检查查询结果：确保SQL查询结果正确无误，数据格式和内容符合可视化需求。

2. 调整图表设置：在可视化模块中，调整图表的设置，如X轴、Y轴和分组字段，确保匹配正确的数据列。

3. 重新加载数据：有时可能是数据加载的问题，尝试重新执行查询并重新加载数据进行可视化。

4. 参考文档：查阅DMS的使用文档，了解如何正确配置和生成图表。

# 五、实验总结

通过本次实验，我们对国内自主研发的数据库阿里云PolarDB有了深入了解，掌握了其基本特性和实际应用。通过创建数据库和数据表、使用SQL语句进行数据操作以及数据可视化，我们熟悉了PolarDB的基本操作和数据管理功能。同时，通过任务编排和使用PolarDB for AI功能，我们体验了数据处理的自动化流程和AI模型在数据库中的应用，为未来的数据分析和管理工作提供了有力的技术支持。总之，本次实验不仅增强了我们对数据库技术的理解，也提升了我们在实际项目中应用数据库和AI技术的能力。