

ALIBABA CLOUD

阿里云

数据管理
数据库实验室

文档版本：20220704

 阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 确定 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.数据库实验室入门介绍	05
2.数据库实验室问答篇	19
3.通过应用程序访问数据库	21

1.数据库实验室入门介绍

在数据库实验室，您可以免费开通不同类型的数据库，并对数据库进行探索和使用。

视频介绍

数据库开通

1. 登录[数据库实验室](#)，左侧将呈现目前实验室支持的数据库类型（后续会支持更多类型的数据库，敬请期待），双击希望开通的数据库类型（例如PolarDB for MySQL），将弹出开通数据库的弹窗。



2. 输入您数据库名称，单击**开通数据库**，就可以免费拥有您自己的数据库。



进入SQL Console

开通数据库之后，您就可以在功能强大的SQL Console里进行您想做的数据操作了。您可以通过以下几种方式进入SQL Console：

- 方式一：在首页左侧菜单栏，单击数据库类型将其展开后，双击目标数据库即可进入。



- 方式二：右键单击目标数据库，单击单库查询。



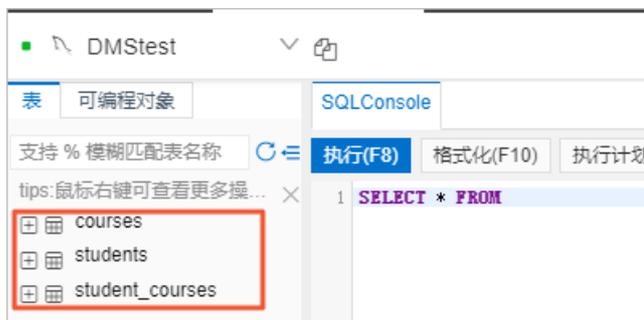
- 方式三：单击顶部菜单栏SQL Console > 单库查询。



数据查询

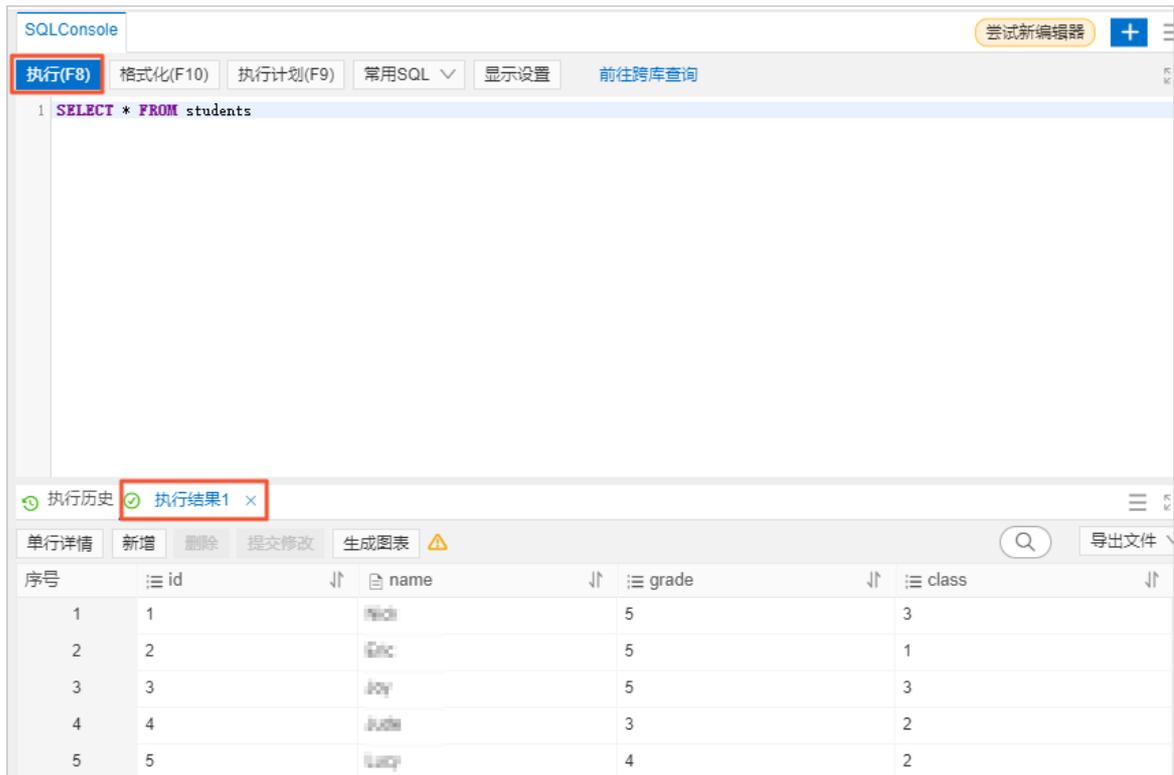
1. 进入SQL Console后，您可以在左侧看到数据库下面的全部数据表。

说明 DMS实验室为您开通数据库时，会自动为您创建示例表和示例数据。

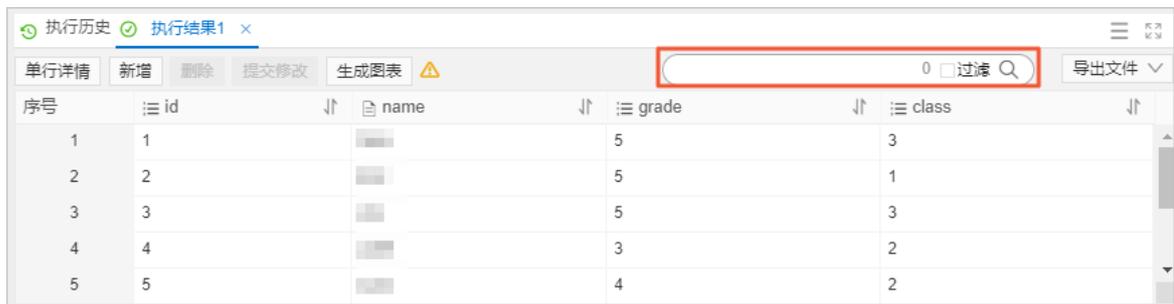


2. 在SQL Console控制台中输入SQL语句，单击执行按钮。

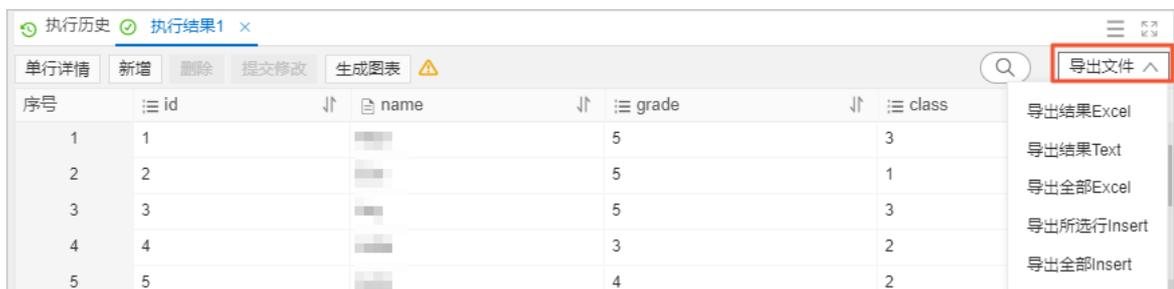
说明 编辑SQL语句时提供了智能提示和高亮功能，您可以在下方的执行结果中查看查询到的数据。



3. 您可以在搜索框对结果集进行检索。



4. 单击执行结果右上角的导出文件，您可以导出查询结果。



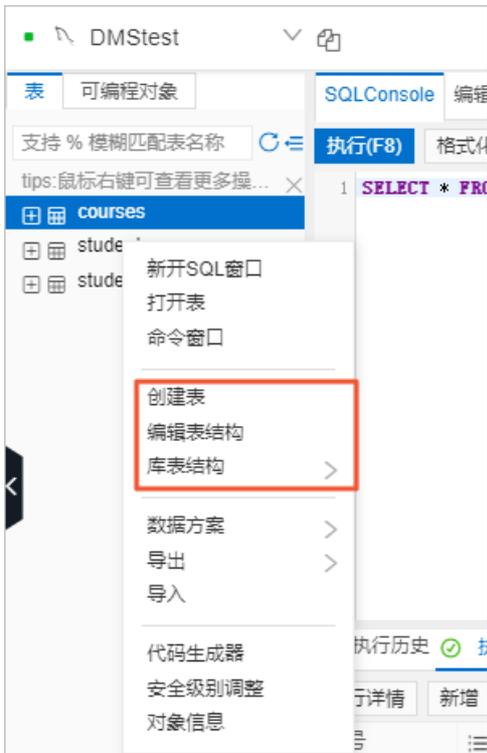
关于数据查询的详细介绍，请参见[数据查询](#)。

表操作

您可以在SQL Console直接通过执行SQL语句来进行一系列的表操作，例如创建表、编辑表、删除表等。

您也可以通过我们提供的可视化编辑功能对表进行操作，具体操作为：在SQL Console中，右键单击目标表，单击编辑表结构，您可以通过可视化编辑直接修改表的基本信息、列信息、索引和外键信息。

说明 本案例以编辑表结构为例进行演示。

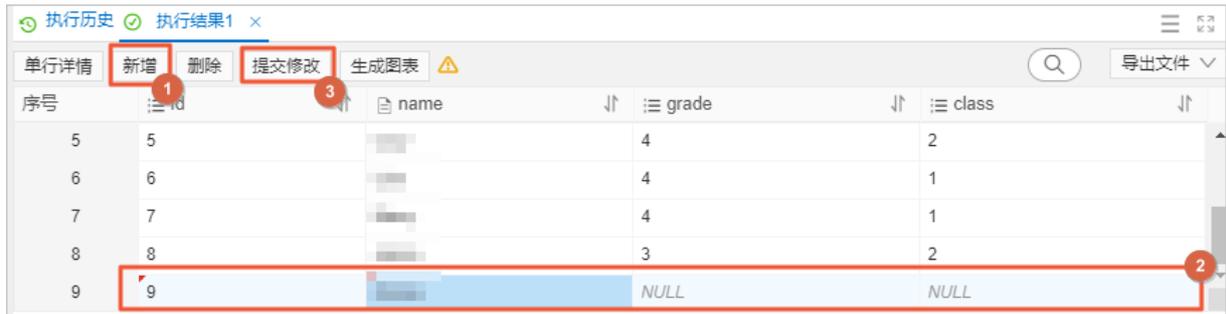


数据变更

您可以在SQL Console直接通过执行SQL语句来进行一系列的数据变更操作，例如插入、删除、修改数据等。

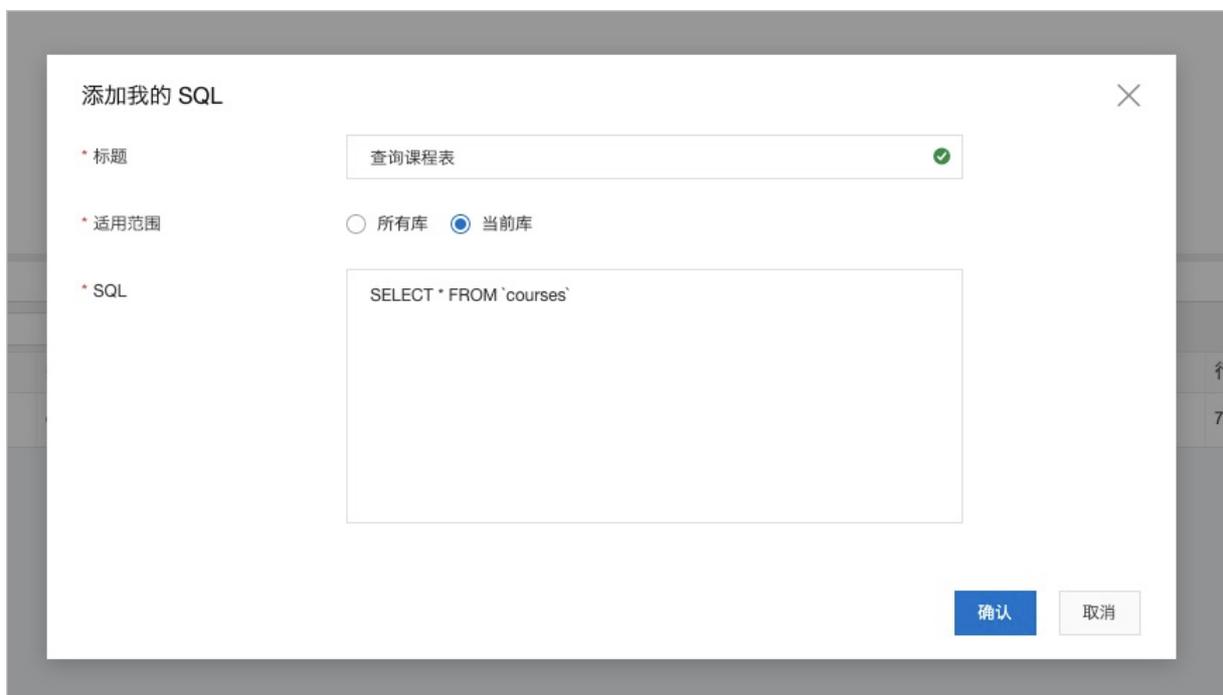
您也可以直接在查询到的结果集中对数据直接进行修改，具体操作为：在**执行结果**页下，单击**新增**，在出现的空白行中填写新数据，单击**提交修改**即可。

说明 本案例以插入数据为例进行演示，数据单元格左上角的红色三角形是为了方便您辨认这条数据是新增或者修改过的。



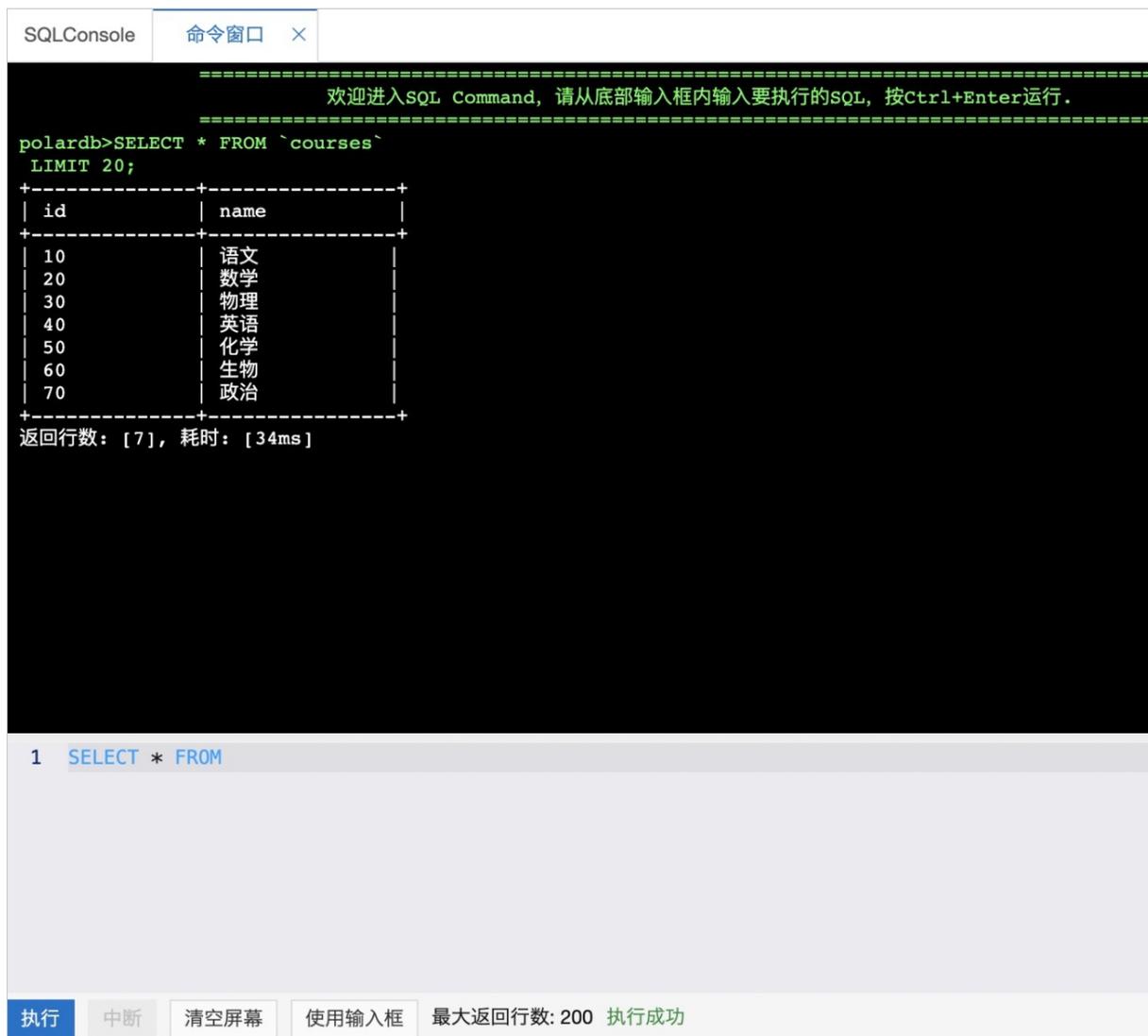
常用SQL管理

在数据库实验室，您可以对您常用的SQL进行管理。在SQL Console中，单击常用SQL，您可以添加、编辑、删除和选择（将SQL直接添加到执行界面中）您常用的SQL。



命令窗口模式

除了在SQL Console中直接执行SQL以及通过可视化操作进行表和数据的变更之外，数据库实验室还为您提供命令窗口模式。在SQL console中，右键单击目标表，单击命令窗口，即可进入命令模式。



导入和导出

数据库实验室支持数据和表结构的导入和导出功能。在SQL console中，右键单击目标表，单击导出 > 导出该表，您可以选择导出表中的数据、表的结构，或者两者同时导出。

 说明 本案例以导出该表为例进行介绍，关于数据导入和导出的详情介绍，可参见[数据导入](#)和[数据导出](#)。



数据库导出

数据库实验室支持将整个数据库进行导出。在控制台首页，单击顶部导航栏的数据方案 > 数据导出 > 数据库导出。



您也可以在SQL Console中单击导出 > 导出该库，将数据库进行导出。



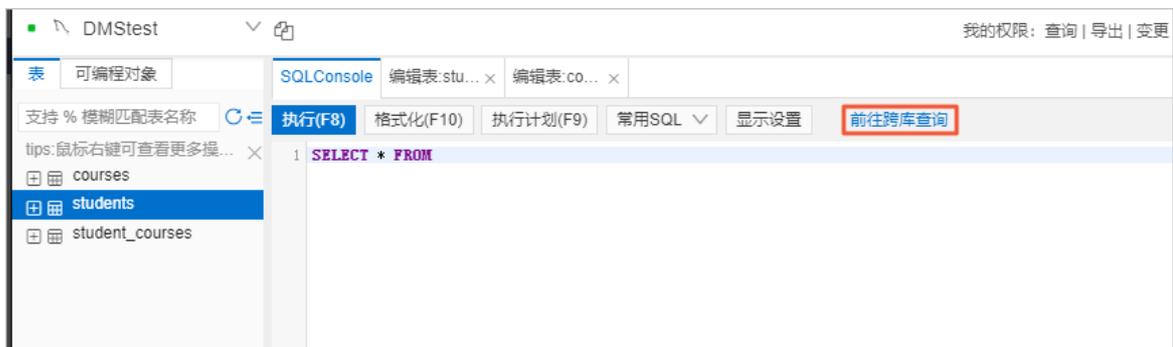
跨库查询

数据库实验室的跨库查询功能为您提供了跨不同数据库类型进行关联查询的能力。

1. 在控制台首页，单击顶部导航栏的SQLConsole > 跨库查询进入。



您也可以在SQL Console中单击前往跨库查询进入。



2. 在跨库查询界面的左侧，您可以看到您已开通实例的DBLink列表。每个DBLink都会指向一个实例。



3. 在查询SQL中，您通过DBLink指代您希望查询的实例。

例如，如果希望查询PolarDB for MySQL实例下面dms_polardb_mysql这个库的courses表，您可以这样写SQL：



4. 通过跨库查询，您可以关联查询不同类型的数据库。

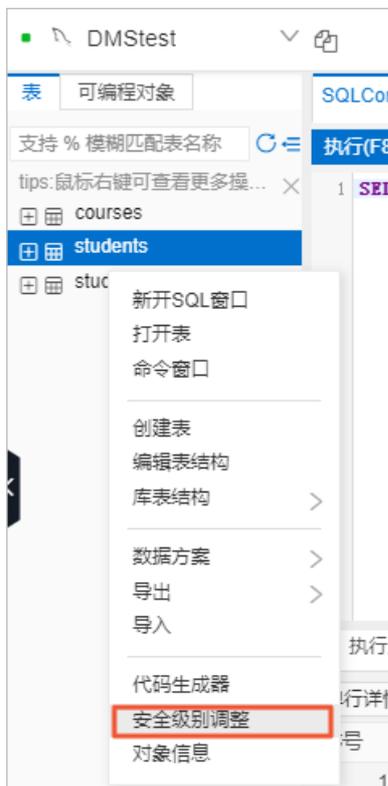
例如，下面这个SQL将PolarDB for MySQL和 RDS for SQLServer这两个实例进行关联查询：

```
1 SELECT *
2 FROM `POLARDB_MYSQL_DBLINK`.`dms_polardb_mysql`.`courses` t1,
3     `RDS_SQLSERVER_DBLINK`.`dms_rds_sqlserver`.`student_courses` t2
4 WHERE t1.id= t2.course_id;
```

序号	id	name	id	student_id	course_id	start_time	score
1	10	语文	57	8	10	2017-09-01 00:00:00	100
2	10	语文	50	7	10	2017-09-01 00:00:00	93
3	10	语文	43	6	10	2017-09-01 00:00:00	83
4	10	语文	36	5	10	2017-09-01 00:00:00	61
5	10	语文	29	4	10	2017-09-01 00:00:00	99
6	10	语文	15	3	10	2017-09-01 00:00:00	89
7	10	语文	8	2	10	2017-09-01 00:00:00	53
8	10	语文	1	1	10	2017-09-01 00:00:00	97
9	20	数学	58	8	20	2017-09-01 00:00:00	99

设置敏感列

在数据表中，有些数据可能是敏感字段，不希望被其他人看到具体的值。数据库实验室提供给敏感字段进行加密的功能。在SQL console中，右键单击目标表，单击安全级别调整，就可以对该表中的字段设置或者取消敏感列设置。



例如，我们希望对 `students` 这张表中的 `grade` 字段进行加密，我们将 `grade` 这个列的级别调整为机密，然后单击提交流程到安全部门，在弹框中单击确认即可。

安全级别调整 ×

表名: `students` 安全级别解释

	字段名	描述	原级别	新级别(仅需要调整变更的字段)	操作状态
1	<code>id</code>	ID	内部	<input checked="" type="radio"/> 内部 <input type="radio"/> 敏感 <input type="radio"/> 机密	
2	<code>name</code>	姓名	内部	<input checked="" type="radio"/> 内部 <input type="radio"/> 敏感 <input type="radio"/> 机密	
3	<code>grade</code>	年级	内部	<input type="radio"/> 内部 <input type="radio"/> 敏感 <input checked="" type="radio"/> 机密	提升
4	<code>class</code>	班级	内部	<input checked="" type="radio"/> 内部 <input type="radio"/> 敏感 <input type="radio"/> 机密	

再次对 `students` 这张表进行查询时，`grade` 这个字段已经被加密。

执行历史 ✓ 执行结果1 ×

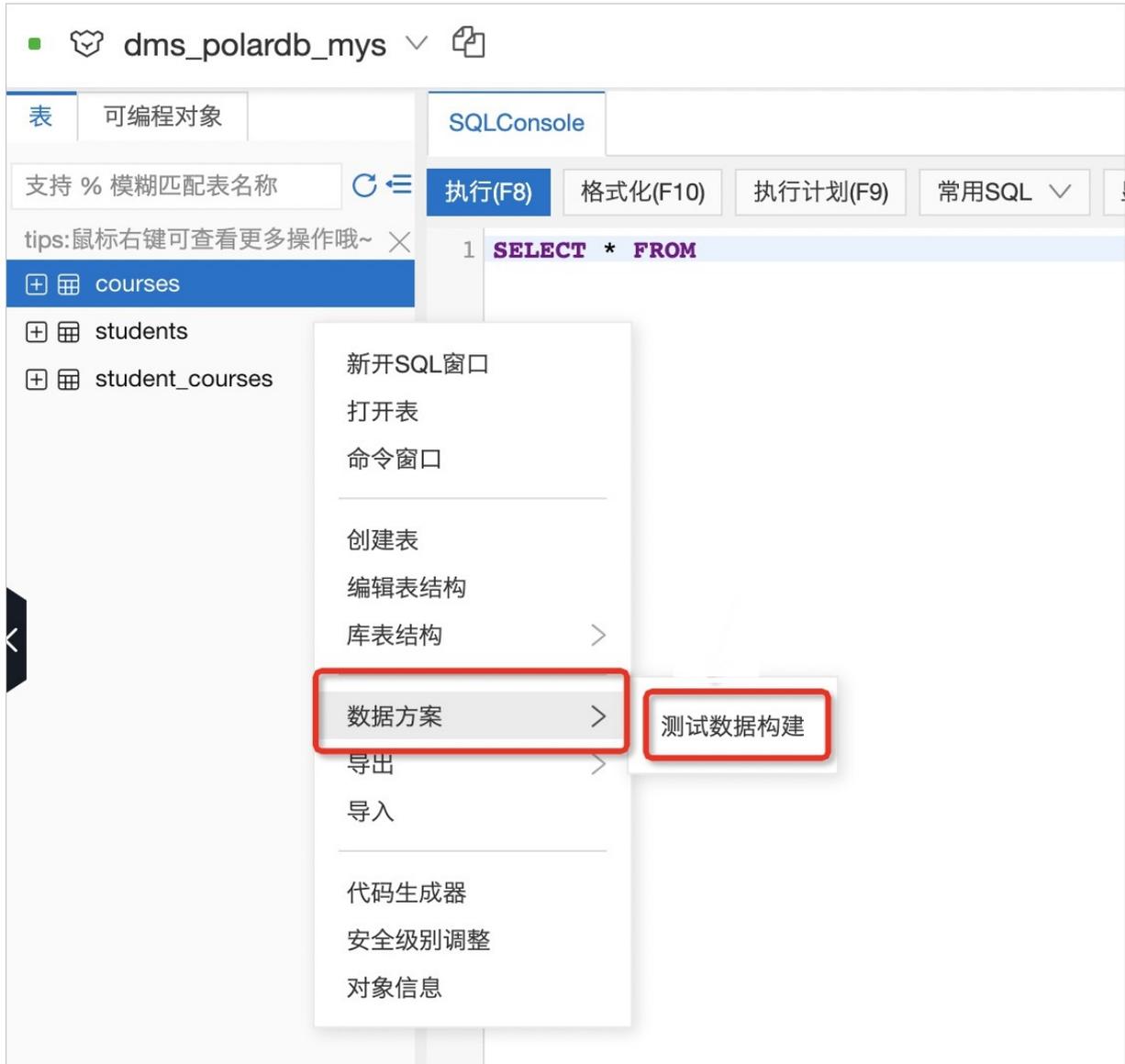
单行详情 ⚠ Q 导出文件

序号	id	name	grade	class
1	1	██████████	*****	3
2	2	██████████	*****	1
3	3	██████████	*****	3
4	4	██████████	*****	2
5	5	██████████	*****	2

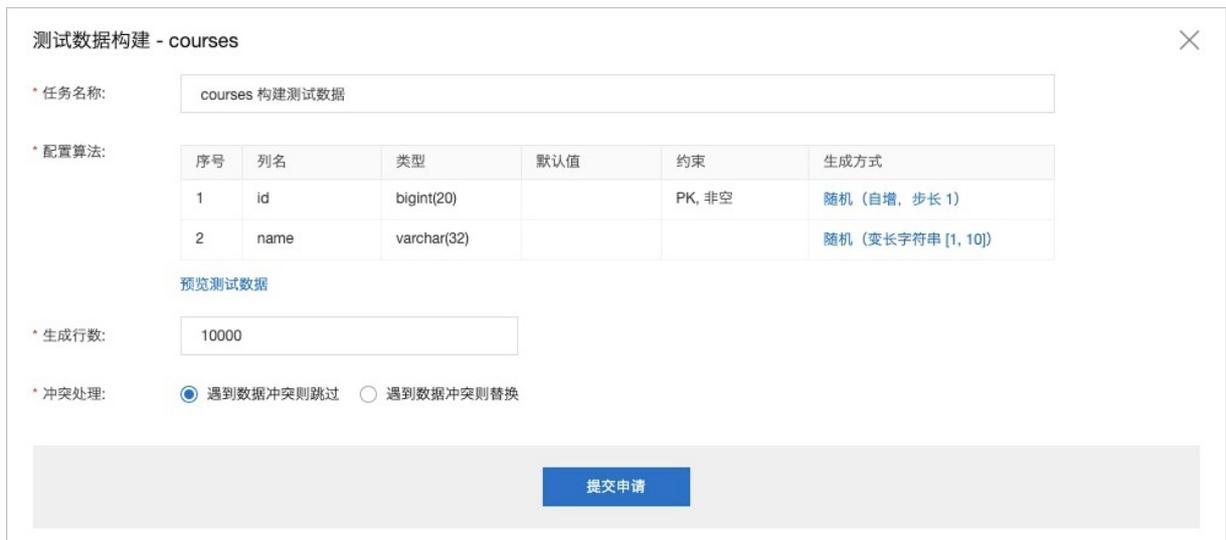
关于设置敏感列的详情介绍，请参见 [字段安全级别调整](#)。

测试数据构建

数据库实验室可以为您的表构建测试数据。在SQL console中，右键单击目标表，单击数据方案 > 测试数据构建即可进入。



您可以根据需要来构建这张表的测试数据。



关于测试数据构建的详细介绍，请参见[测试数据构建](#)。

应用程序或命令行直连

如果您希望通过应用程序或者命令行的方式直接连接数据库，目前MySQL数据库已支持该功能。在控制台首页左侧，右键单击目标实例，单击[获取连接信息](#)，您就可以看到目标实例的连接地址以及您的账号名和密码。



获取帮助

如果您对数据库实验室的使用有什么疑问或者建议，可以通过以下方式咨询：

- [提交工单](#)。
- 使用钉钉搜索钉钉群号21991247，加入钉钉群反馈。

2. 数据库实验室问答篇

DMS数据库实验室已上线，本文对数据库实验室进行简单介绍。

- 什么是数据库实验室？

数据库实验室是DMS打造的一站式数据库体验馆，为免费体验不同类型的阿里云数据库产品提供统一的入口。在数据库实验室，每个人都可以拥有免费的数据库，体验不同的云数据库产品。

- 在数据库实验室我可以做什么？

在数据库实验室，您可以一键开通数据库，创建、编辑和删除表，对表里的数据进行增删改查。您还可以免费体验DMS的高阶功能，例如跨库查询、测试数据构建、数据库导出、批量数据导入和导出、敏感列设置、操作日志管理等。

- 数据库实验室收费吗？

数据库实验室是完全免费的，不会收取任何费用。

- 哪些用户可以使用数据库实验室？

数据库实验室面向全体阿里云用户开放，只要您有一个阿里云账号，就可以使用数据库实验室。

- 数据库实验室的入口在哪里？

您可以通过域名访问：[DMS数据库实验室](#)。

- 在哪里可以看到数据库实验室目前支持的数据库类型？

您可以在数据库实验室左侧的导航栏里看到不同类型的实例名称，这些都是数据库实验室目前已经支持的数据库类型（目前已支持9种数据库类型：PolarDB for MySQL、PolarDB for PostgreSQL、PolarDB O引擎、RDS for MySQL、RDS for SQLServer、RDS for PostgreSQL、ADB for MySQL、ADB for PostgreSQL以及MongoDB）。接下来数据库实验室也会不断去支持更多的数据库类型，敬请期待。

- 怎么开通数据库？

您可以在左侧导航栏双击您想开通的实例，或者右键单击实例名称，单击**开通实验室数据库**，就可以开通实验室数据库。您可以免费开通控制台的导航栏中所有类型的数据库。

- 我想重新开通一个数据库，怎么办？

如果您已经开通了一个数据库并希望删除该库重新开通，您只需要在左侧导航栏中，右键单击您想删除的实例，选择**释放实例**，就可以删除该数据库。然后您即可开通新的数据库。

- 我可以在哪里对数据进行操作？

数据库开通后，会帮您自动打开**SQL Console**，您也可以在左侧导航栏双击数据库名称，进入**SQL Console**。

 **说明** SQL Console是一个集成了众多重要功能的藏宝地。

- 您可以通过图形化操作、写SQL语句等方式创建、编辑、删除表数据，对数据进行增删改查。
- 您可以进行一键导出数据库、批量导入导出数据、自动生成测试数据等操作。
- 您可以打开命令窗口，直接对结果集进行编辑保存。这里还有很多丰富而强大的功能，快来亲自体验一下吧。

- 什么是跨库查询？

跨库查询功能可以让您用一条SQL实现不同类型数据库之间的关联查询。您可以单击数据库实验室顶部菜单栏的**SQLConsole > 跨库查询**进入查询页面。

- 可以获取数据库的账号密码，并通过应用程序或者命令行直连数据库吗？

目前对于MySQL数据库是可以的，右键单击目标实例，单击获取连接信息处即可看到您的数据库的连接地址、用户名和密码。

- 数据库实验室适用于哪些场景呢？

数据库实验室对全体阿里云用户开放，适用于任何想免费体验不同类型数据库的小伙伴。

情景举例：

- 学生A：数据库课程上刚教了SQL Server，我自己也想有个SQL Server数据库可以体验一下，有什么便捷的方法呢？来数据库实验室吧。
 - 学生B：我有个小应用需要连接数据库，从哪里可以获得一个免费的数据库并且可以让我通过应用程序访问呢？数据库实验室就可以呀。
 - 老师：作为《数据库导论》这门课的任课老师，我希望学生都可以有自己的数据库，方便他们动手实践和进行课程实验，怎么样才能方便地进行指导和管理？让学生们直接登录到数据库实验室吧。
 - 开发者A：这两个数据库类型在语法支持上面有什么区别？有什么办法可以让我快速验证一下？通过数据库实验室就可以轻松搞定呀。
 - 开发者B：听过阿里云新推出了PolarDB系列，业界传得很火，我也想试试，该去哪里呢？当然是数据库实验室啦。
- 数据库实验室的资源有使用期限吗？

原则上数据库实验室是可以无限期免费使用的。您每进入一次实验室，就可以免费续期30天。如果连续30天都没有进入到实验室，会回收之前的资源，再次进入时会重新分配资源。

- 数据库实验室的使用有什么限制吗？

目前每位用户每个库的使用空间上限为512M，每个数据库创建的表数量上限为1000个。如果您有特别的需求，可以联系我们。

3.通过应用程序访问数据库

本文档指导您如何通过应用程序访问数据库。

前提条件

已访问数据库实验室，并创建PolarDB for MySQL或RDS for MySQL实例。

背景信息

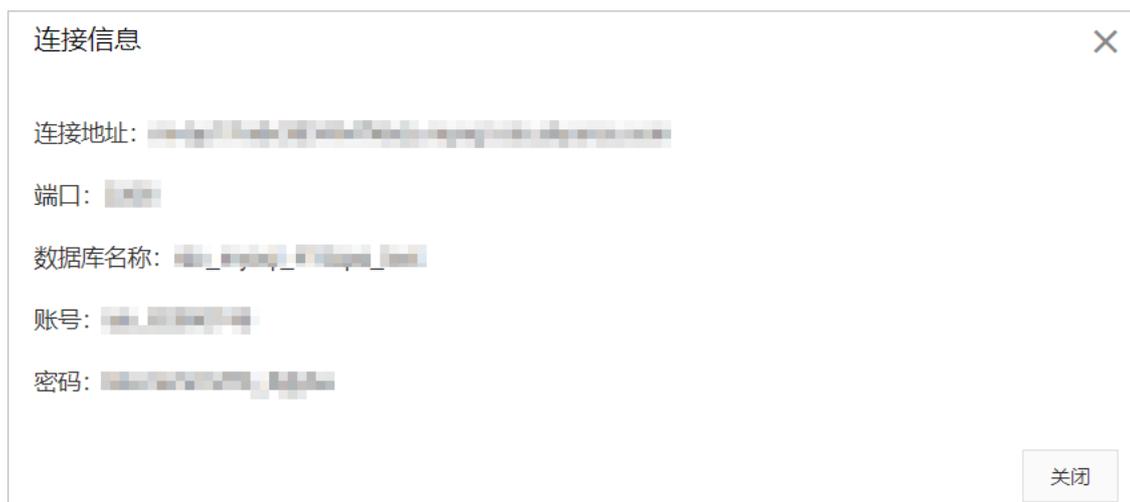
数据库实验室是DMS打造的一站式数据库体验馆，为免费体验不同类型的阿里云数据库产品提供统一的入口。在数据库实验室，您可以免费体验不同的云数据库产品，详情请参见[数据库实验室入门介绍](#)。

步骤一：获取数据库连接信息

1. 登录[数据库实验室](#)。
2. 在控制台首页左侧的实例列表中，右键单击目标数据库实例。
3. 从弹出的列表中，选择获取连接信息，即可查看对应数据库的连接信息。



连接信息如下图所示。



连接信息

连接地址: [Redacted]

端口: [Redacted]

数据库名称: [Redacted]

账号: [Redacted]

密码: [Redacted]

关闭

步骤二：通过应用程序访问数据库

获取数据库连接信息后，您就可以在应用程序中通过代码访问数据库。本文档向您提供了3种语言示例：Java、Python和C语言。

② 说明 需替换的参数及说明：

- *Host*: 连接地址
- *Port*: 端口
- *myDatabase*: 数据库名称
- *myUsername*: 账号
- *myPassword*: 密码

- Java程序访问：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
public class DatabaseConnection
{
    public static void main(String args[]) {
        String connectionUrl= "jdbc:mysql://Host:Port/myDatabase";
        ResultSet resultSet;
        try (Connection connection=DriverManager.getConnection(connectionUrl,"myUsername"
, "myPassword");
            Statement statement = connection.createStatement()) {
            String selectSql = "SELECT * FROM `courses`";           // 输入希望执行的SQL
            resultSet = statement.executeQuery(selectSql);
            while (resultSet.next()) {
                System.out.println(resultSet.getString("name"));
            }
        }
        catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

- Python程序访问:

```
import pymysql
connection = pymysql.connect(host='Host',
                             port=Port,
                             user='myUsername',
                             passwd='myPassword',
                             db='myDatabase')

try:
    with connection.cursor() as cursor:
        sql = "SELECT * FROM `courses`"           // 输入希望执行的SQL
        cursor.execute(sql)
        for result in cursor:
            print(result)
finally:
    connection.close()
```

- C语言:

```
#include <stdio.h>
#include <mysql.h>
#include <string.h>
void main(void)
{
    MYSQL *t_mysql;
    MYSQL_RES *res = NULL;
    MYSQL_ROW row;
    char *query_str = NULL;
    int rc, i, fields;
    int rows;
    char select[] = "select * from courses"; // 输入希望执行的SQL
    t_mysql = mysql_init(NULL);
    if(NULL == t_mysql){
        printf("init failed\n");
    }
    if(NULL == mysql_real_connect(t_mysql, Host, myUsername, myPassword, myDatabase,
        Port, NULL, 0)){
        printf("connect failed\n");
    }
    if(mysql_real_query(t_mysql, select, strlen(select)) != 0){
        printf("select failed\n");
    }
    res = mysql_store_result(t_mysql);
    if (NULL == res) {
        printf("mysql_restore_result(): %s\n", mysql_error(t_mysql));
        return -1;
    }
    fields = mysql_num_fields(res);
    while ((row = mysql_fetch_row(res))) {
        for (i = 0; i < fields; i++) {
            printf("%s\t", row[i]);
        }
        printf("\n");
    }
    mysql_close(t_mysql);
}
```