ALIBABA CLOUD

阿里云

云原生数据仓库AnalyticDB MySQL版

连接数据库

文档版本：20220712

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 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.申请和释放公网地址

云原生数据仓库AnalyticDB MySQL版集群支持VPC地址和公网地址两种类型的地址，如果需要通过公网连接集群，您需要先申请公网地址。

公网和VPC网络

网络类型	说明
VPC网络	• 一个VPC就是一个隔离的网络环境。VPC的安全性较高，推荐您使用VPC网络。 • 创建集群时，指定集群所属VPC。 • 支持自定义VPC中的路由表、IP地址范围和网关。 • 支持通过专线或者VPN的方式，将自建机房与阿里云VPC组合成一个虚拟机房，实现应用平滑上云。

公网	• 公网地址需要手动申请，不需要时也可以释放。
	• 公网地址适用场景：
	  ◦ 阿里云以公的设备需要通过公网地址访问AnalyticDB MySQL集群。
	  ◦ 需要访问的AnalyticDB MySQL集群位于不同的地域或者网络类型不同。

申请公网地址

1. 登录云原生数据仓库AnalyticDB MySQL控制台。
2. 在页面左上角，选择集群所在地域。
3. 在左侧导航栏，单击**集群列表**。
4. 根据您的集群类型，选择**湖仓版（3.0）**或者**数仓版（3.0）**。
5. 单击目标**集群ID**，进入**集群信息**页面。
6. 在**集群信息**页面的**网络信息**卡片中，单击**申请公网**。
7. 在**申请公网**提示框中单击**确定**，生成公网地址。

> **说明** 公网地址生成后，如果需要使用公网地址访问AnalyticDB MySQL集群，您还需要将待访问AnalyticDB MySQL集群的设备IP加入白名单。

释放公网地址

1. 登录云原生数据仓库AnalyticDB MySQL控制台。
2. 在页面左上角，选择集群所在地域。
3. 在左侧导航栏，单击**集群列表**。
4. 根据您的集群类型，选择**湖仓版（3.0）**或者**数仓版（3.0）**。
5. 单击目标**集群ID**，进入**集群信息**页面。
6. 在**集群信息**页面，单击**释放公网**。
7. 在**释放公网**提示框中单击**确定**，释放公网地址。

2.工具兼容性说明

2.1. 兼容性概览

以下列出了云原生数据仓库AnalyticDB MySQL版支持的客户端或ETL工具，以及这些客户端或工具与云原生数据仓库AnalyticDB MySQL版在数据库连通性、列举数据库、建表、查询表数据等方面的兼容性。

客户端与AnalyticDB MySQL版兼容性

客户端	连接数据库	列举数据库	建表	列举表	查看表结构	写表	查表	建视图	查看视图结构	查询视图
Navicat Premium	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
DBeaver	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
DBVisualizer	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
SQL WorkBench/J	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
DataGrip	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持
Tableau	支持	支持	无该功能	支持	无该功能	无该功能	支持	无该功能	无该功能	支持
QlikView	支持	支持	无该功能	支持	支持	无该功能	支持	无该功能	支持	支持
永洪BI	支持	支持	无该功能	支持	支持	无该功能	支持	无该功能	支持	支持
FineBI 5.0	支持	支持	无该功能	支持	支持	无该功能	支持	无该功能	支持	支持
FineReport 10.0	支持	支持	无该功能	支持	支持	无该功能	支持	无该功能	支持	支持
Smartbi	支持	支持	无该功能	支持	支持	支持	支持	无该功能	支持	支持

ETL工具与AnalyticDB MySQL版兼容性

ETL工具	连接数据库	查询视图	查表	写表	列举数据库	建表	列举表	查看表结构	查看视图结构	建视图
Oracle Golden Gate	支持	支持	支持	支持	• OGG专注链路同步、不支持该功能。 • 使用OGG时，需要提前准备源端Oracle、目的端AnalyticDB MySQL版、OGG同步链路以及存储OGG的元数据表MySQL表。					
Scriptella	支持	支持	支持	支持	支持	支持	支持	支持	支持	支持

2.2. Navicat Premium

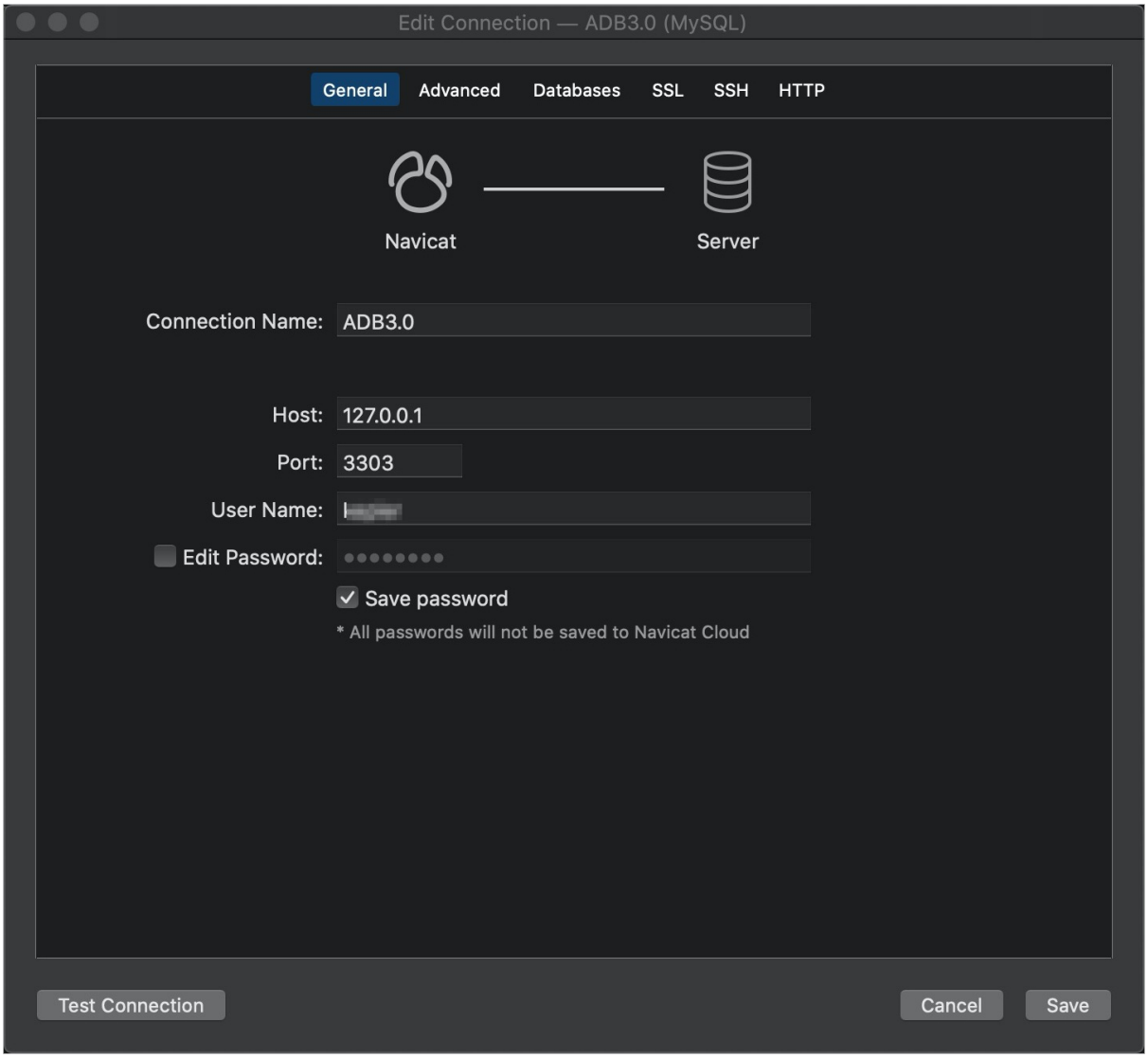
本文测试了Navicat Premium 12.1.27版本与AnalyticDB for MySQL在连通性、列举数据库、创建表等方面的兼容性，并给出测试结果图。

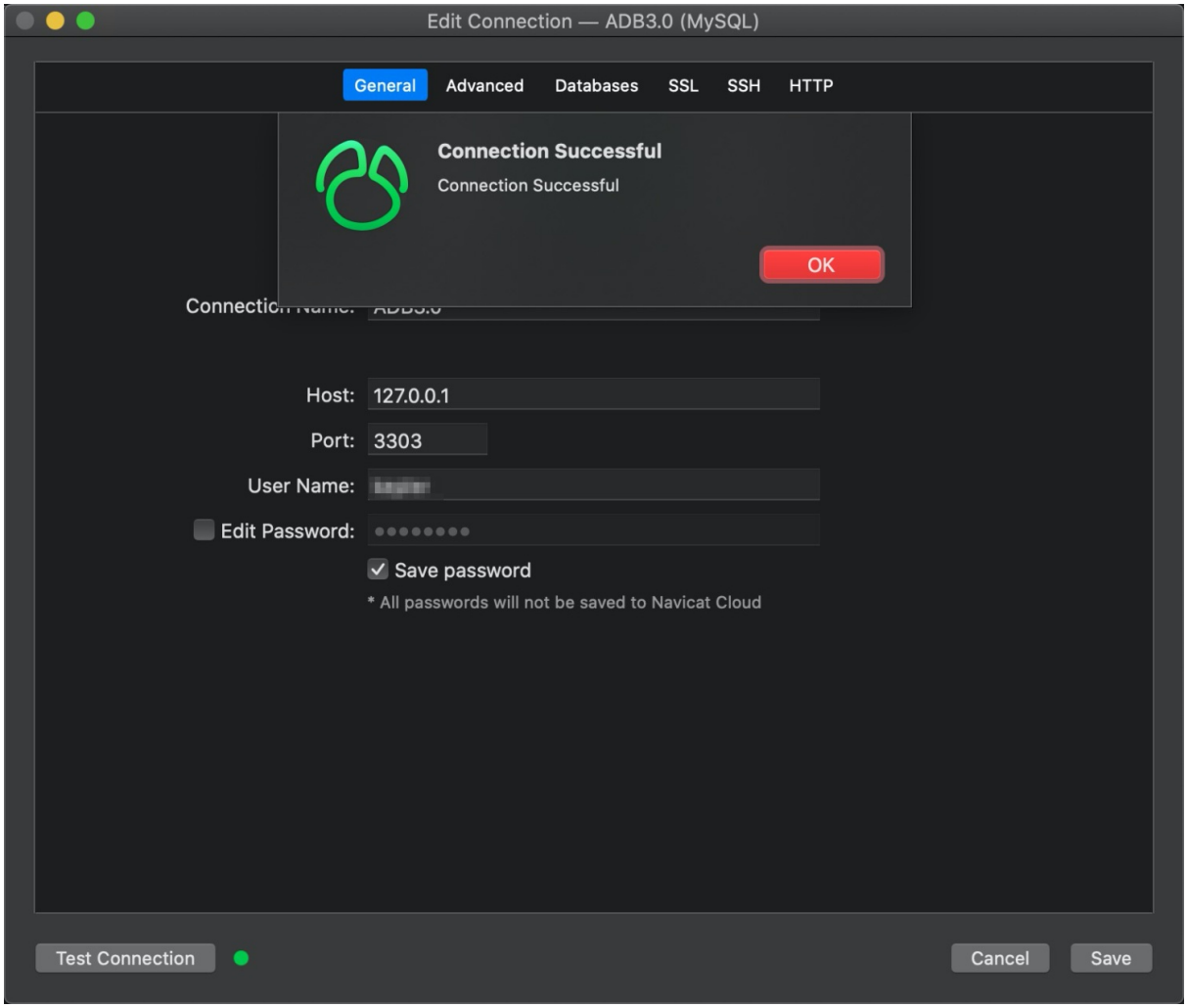
测试环境

Java	MySQL	Navicat Premium
• java version "1.8.0_161" • Java(TM) SE Runtime Environment (build 1.8.0_161-b12) • Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)	mysql Ver5.6.46 for osx10.13 on x86_64 (Homebrew)	版本为Navicat Premium 12.1.27

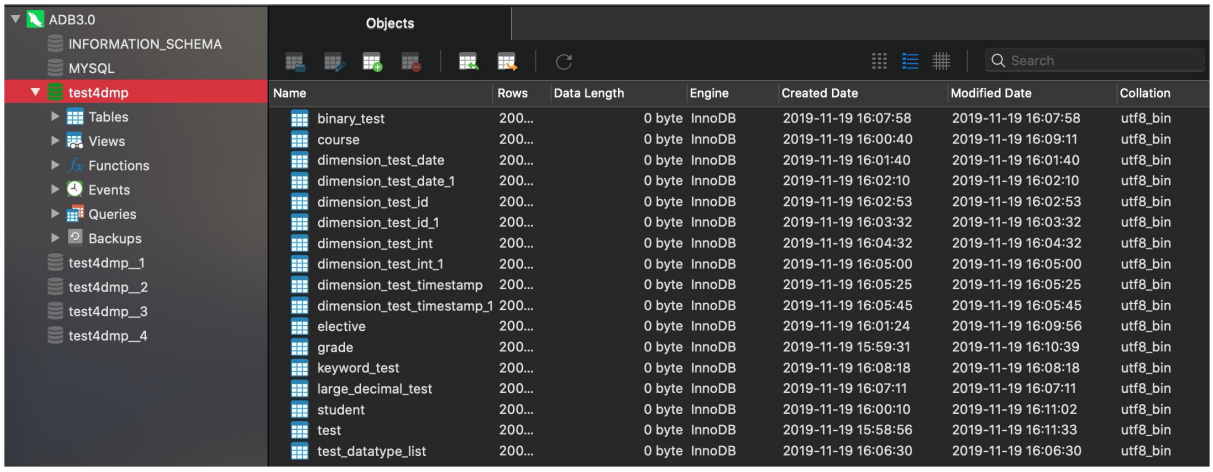
测试范围

- 连通性

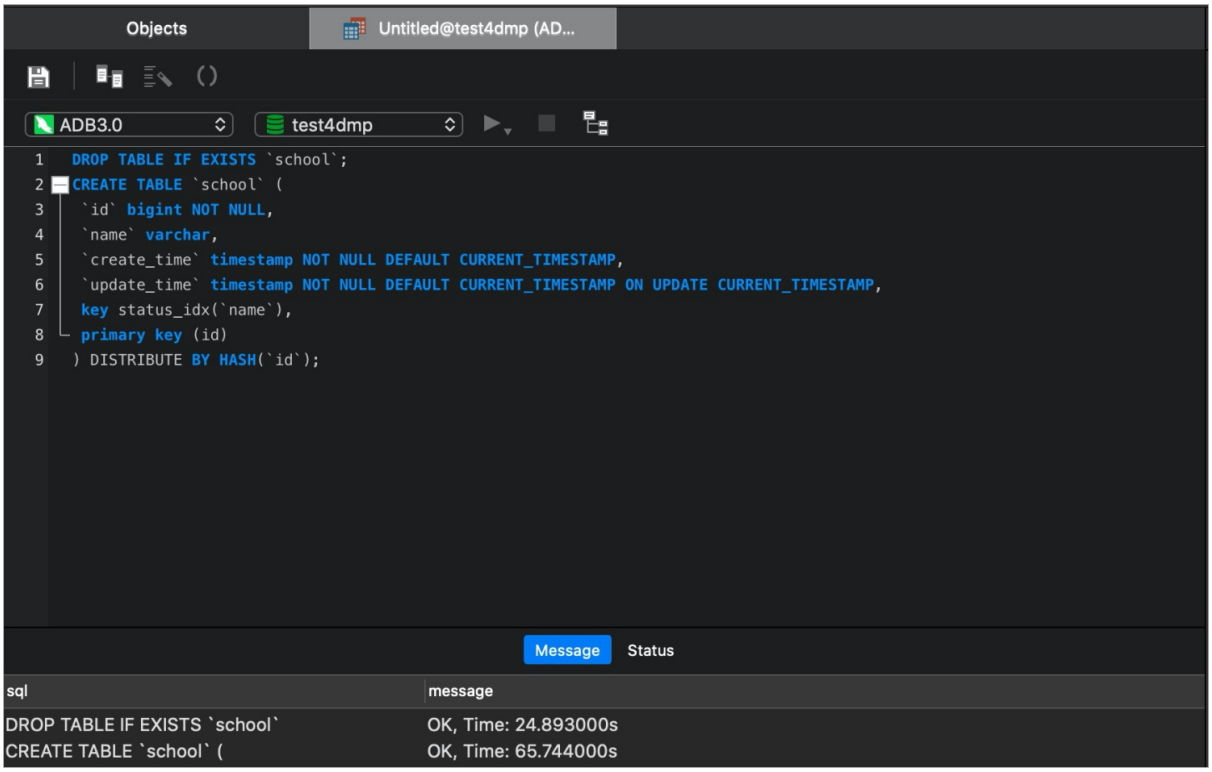

Edit Connection — ADB3.0 (MySQL)
General
Advanced
Databases
SSL
SSH
HTTP
Navicat
Server
Connection Name:
ADB3.0
Host:
127.0.0.1
Port:
3303
User Name:
Edit Password:
Save password
* All passwords will not be saved to Navicat Cloud
Test Connection
Cancel
Save


Edit Connection — ADB3.0 (MySQL)
General
Advanced
Databases
SSL
SSH
HTTP
Connection Successful
Connection Successful
OK
Host: 127.0.0.1
Port: 3303
User Name:
Edit Password:
Save password
* All passwords will not be saved to Navicat Cloud
Test Connection
Cancel
Save

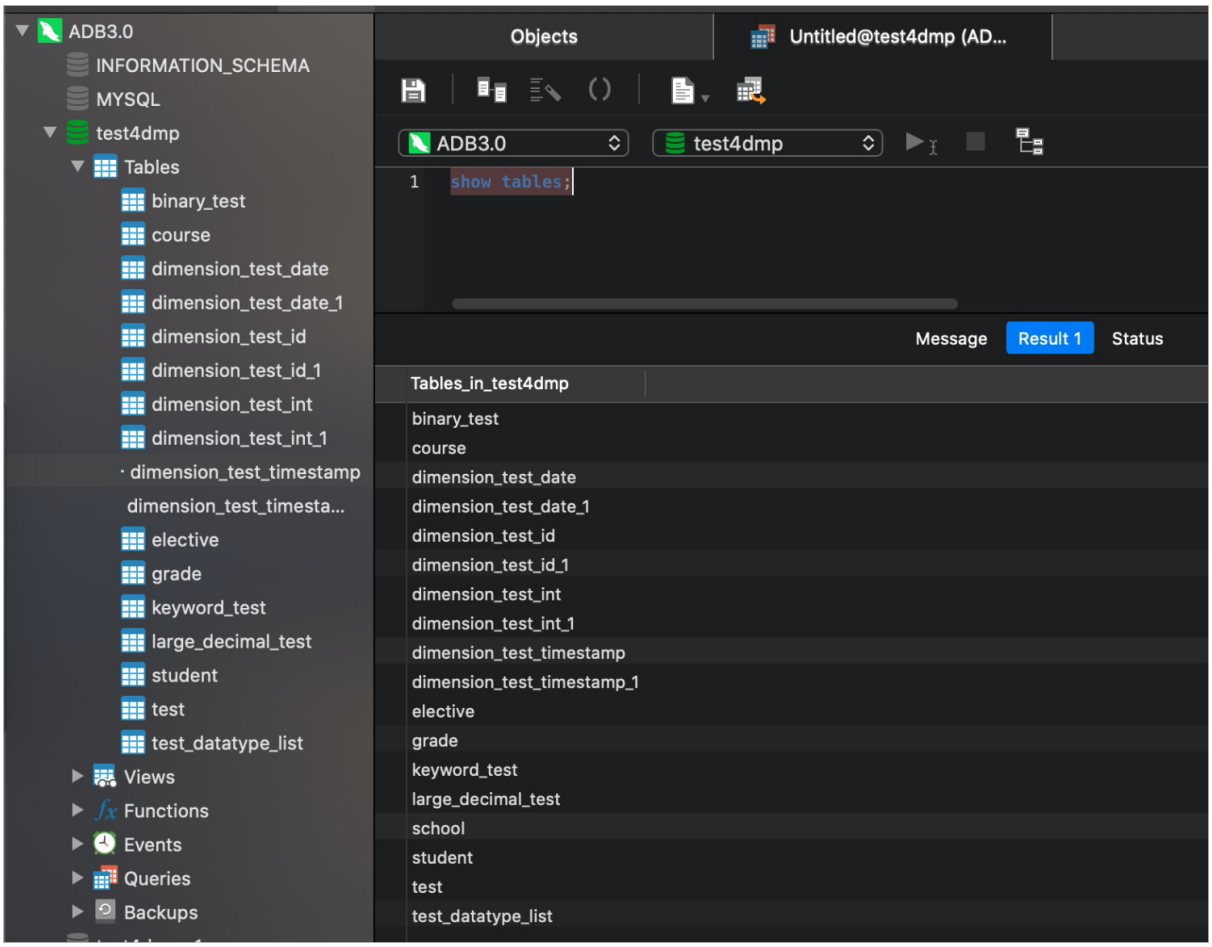
- 列举数据库


ADB3.0
INFORMATION_SCHEMA
MYSQL
test4dmp
Tables
Views
Functions
Events
Queries
Backups
test4dmp_1
test4dmp_2
test4dmp_3
test4dmp_4
Objects
Search
Name Rows Data Length Engine Created Date Modified Date Collation
binary_test 200... 0 byte InnoDB 2019-11-19 16:07:58 2019-11-19 16:07:58 utf8_bin
course 200... 0 byte InnoDB 2019-11-19 16:00:40 2019-11-19 16:09:11 utf8_bin
dimension_test_date 200... 0 byte InnoDB 2019-11-19 16:01:40 2019-11-19 16:01:40 utf8_bin
dimension_test_date_1 200... 0 byte InnoDB 2019-11-19 16:02:10 2019-11-19 16:02:10 utf8_bin
dimension_test_id 200... 0 byte InnoDB 2019-11-19 16:02:53 2019-11-19 16:02:53 utf8_bin
dimension_test_id_1 200... 0 byte InnoDB 2019-11-19 16:03:32 2019-11-19 16:03:32 utf8_bin
dimension_test_int 200... 0 byte InnoDB 2019-11-19 16:04:32 2019-11-19 16:04:32 utf8_bin
dimension_test_int_1 200... 0 byte InnoDB 2019-11-19 16:05:00 2019-11-19 16:05:00 utf8_bin
dimension_test_timestamp 200... 0 byte InnoDB 2019-11-19 16:05:25 2019-11-19 16:05:25 utf8_bin
dimension_test_timestamp_1 200... 0 byte InnoDB 2019-11-19 16:05:45 2019-11-19 16:05:45 utf8_bin
elective 200... 0 byte InnoDB 2019-11-19 16:01:24 2019-11-19 16:09:56 utf8_bin
grade 200... 0 byte InnoDB 2019-11-19 15:59:31 2019-11-19 16:10:39 utf8_bin
keyword_test 200... 0 byte InnoDB 2019-11-19 16:08:18 2019-11-19 16:08:18 utf8_bin
large_decimal_test 200... 0 byte InnoDB 2019-11-19 16:07:11 2019-11-19 16:07:11 utf8_bin
student 200... 0 byte InnoDB 2019-11-19 16:00:10 2019-11-19 16:11:02 utf8_bin
test 200... 0 byte InnoDB 2019-11-19 15:58:56 2019-11-19 16:11:33 utf8_bin
test_datatype_list 200... 0 byte InnoDB 2019-11-19 16:06:30 2019-11-19 16:06:30 utf8_bin

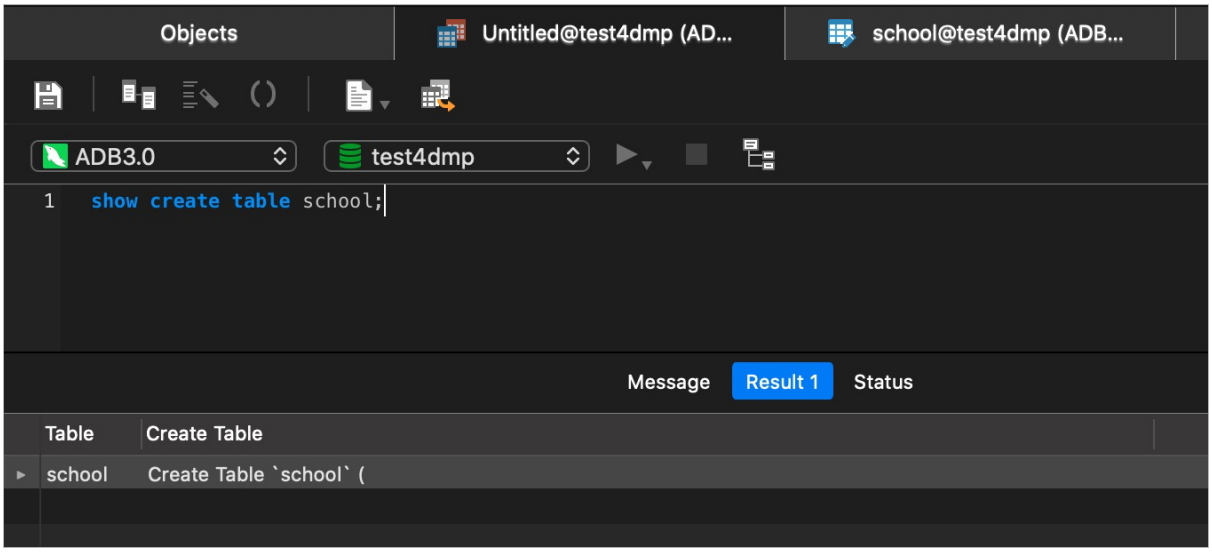
- 创建表

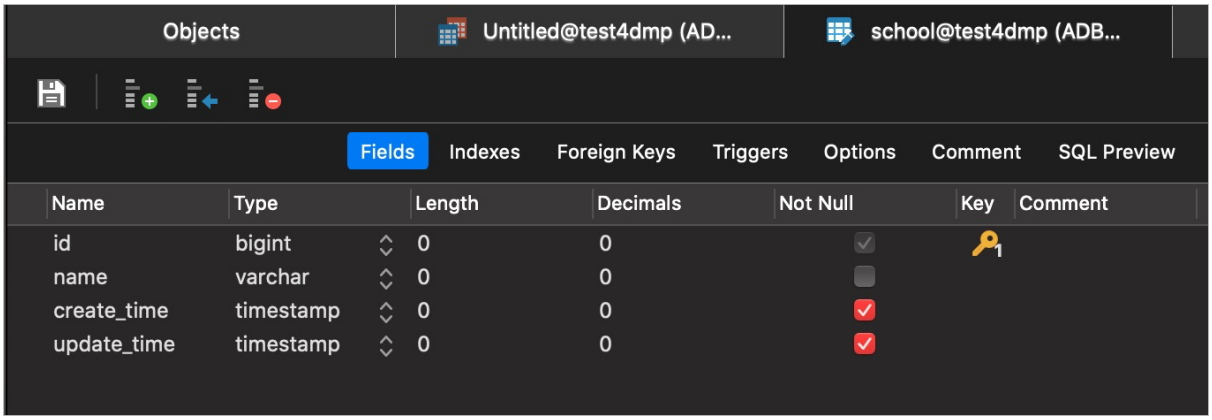

Objects
Untitled@test4dmp (AD...
ADB3.0
test4dmp
1 DROP TABLE IF EXISTS `school`;
2 CREATE TABLE `school` (
3 `id` bigint NOT NULL,
4 `name` varchar,
5 `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
6 `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
7 key status_idx(`name`),
8 primary key (id)
9) DISTRIBUTE BY HASH(`id`);
Message
Status
sql
message
DROP TABLE IF EXISTS `school`
OK, Time: 24.893000s
CREATE TABLE `school` (
OK, Time: 65.744000s

- 列举所有表

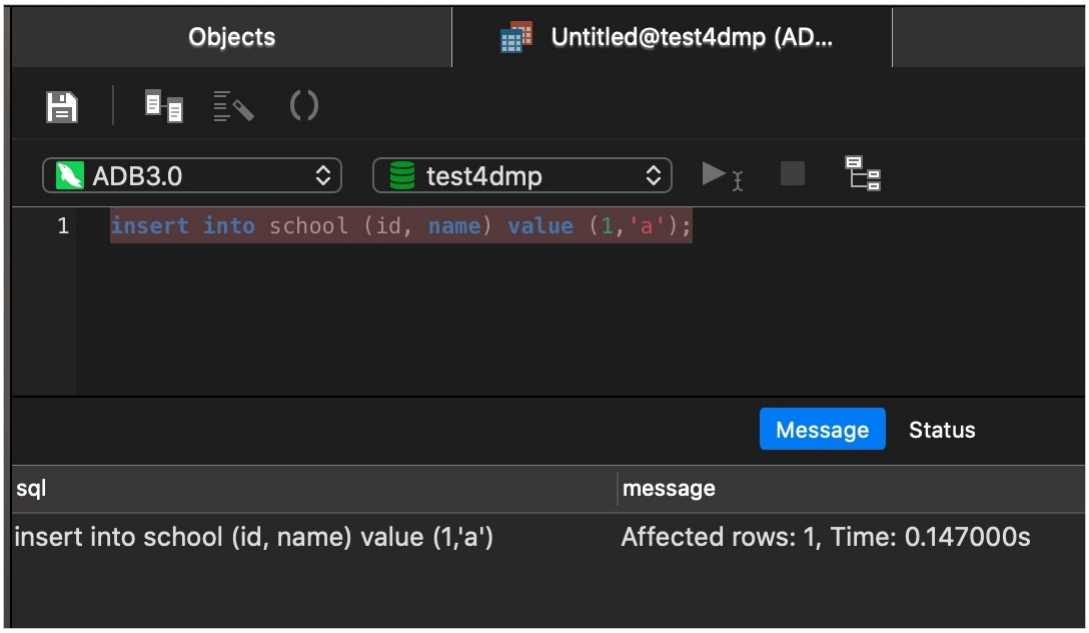

ADB3.0
INFORMATION_SCHEMA
MYSQL
test4dmp
Tables
binary_test
course
dimension_test_date
dimension_test_date_1
dimension_test_id
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timesta...
elective
grade
keyword_test
large_decimal_test
student
test
test_datatype_list
Views
Functions
Events
Queries
Backups
Objects
Untitled@test4dmp (AD...
ADB3.0
test4dmp
1 show tables;
Message
Result 1
Status
Tables_in_test4dmp
binary_test
course
dimension_test_date
dimension_test_date_1
dimension_test_id
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timestamp_1
elective
grade
keyword_test
large_decimal_test
school
student
test
test_datatype_list

- 查看表结构

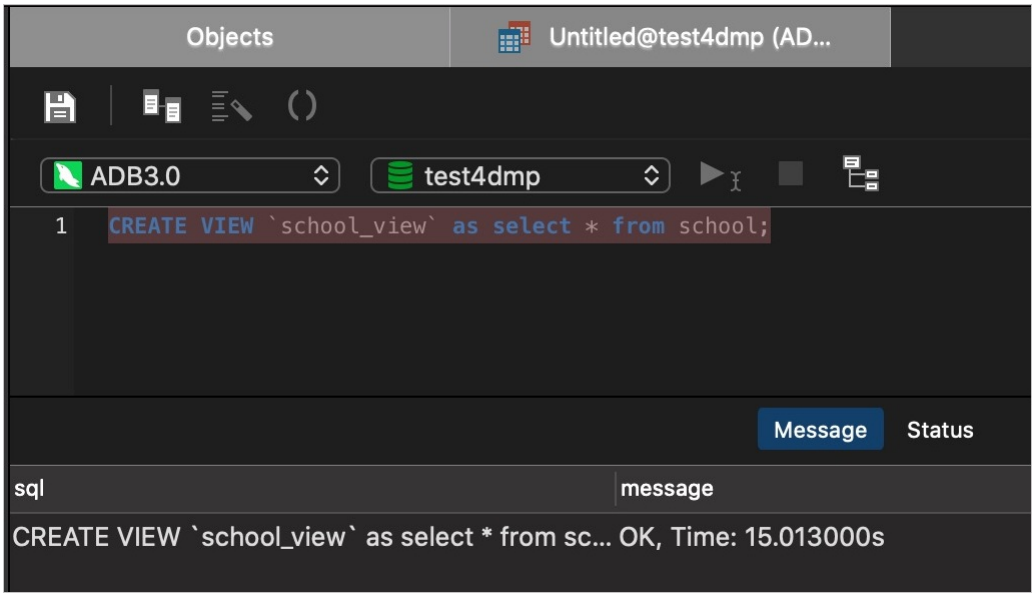




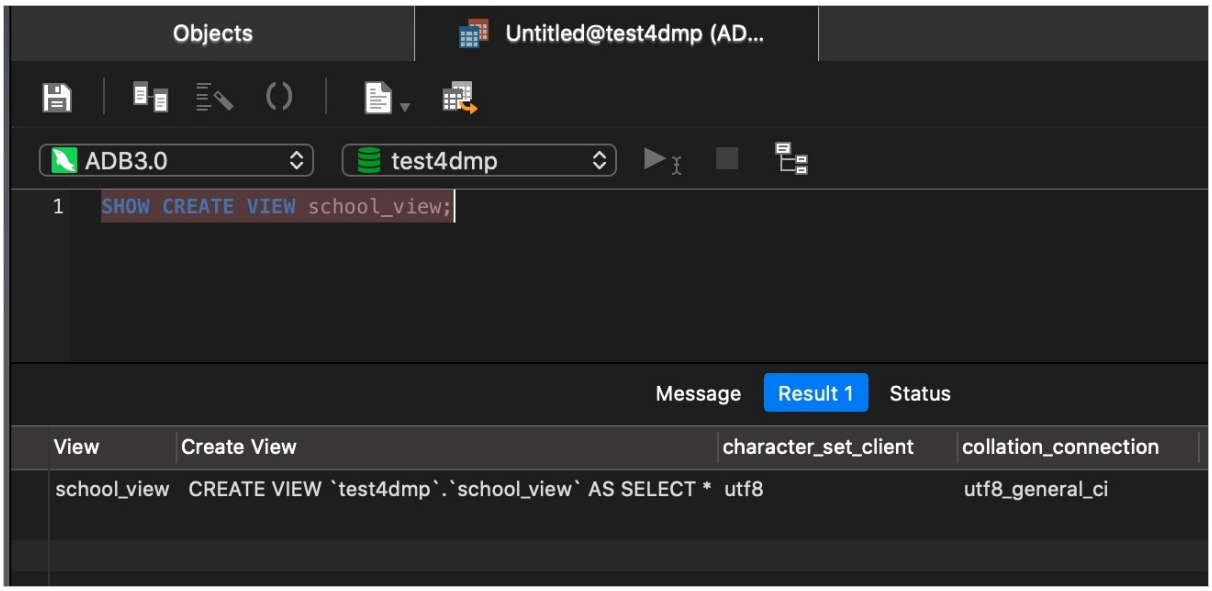
- 向表中写入数据



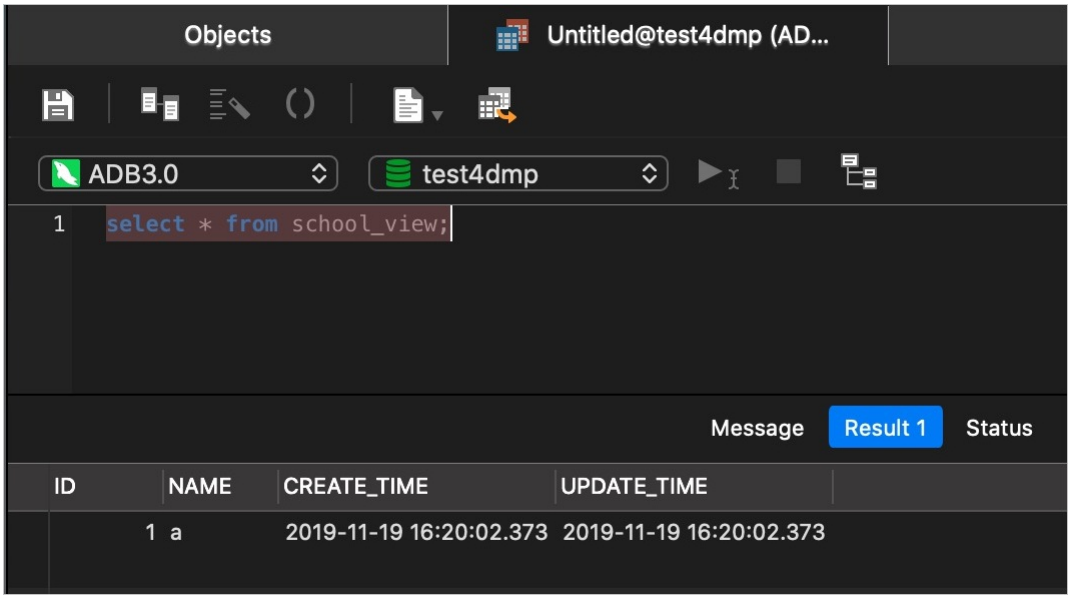
- 创建视图


Objects
Untitled@test4dmp (AD...
ADB3.0
test4dmp
1 CREATE VIEW `school_view` as select * from school;
Message
Status
sql
message
CREATE VIEW `school_view` as select * from sc... OK, Time: 15.013000s

- 查看视图结构


Objects
Untitled@test4dmp (AD...
ADB3.0
test4dmp
1 SHOW CREATE VIEW school_view;
Message
Result 1
Status
View
Create View
character_set_client
collation_connection
school_view
CREATE VIEW `test4dmp`.`school_view` AS SELECT *
utf8
utf8_general_ci

- 查询视图



2.3. DBeaver

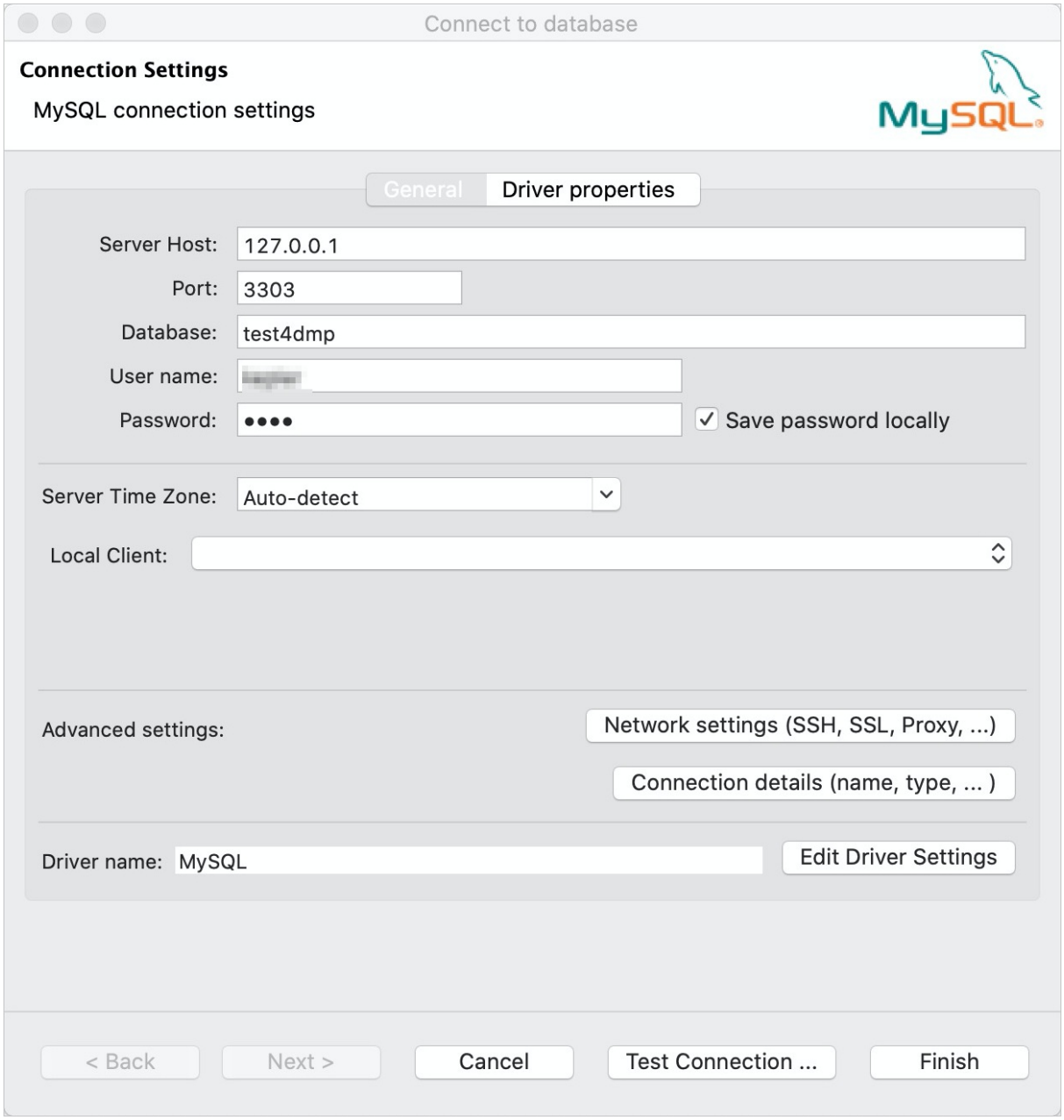
本文测试了DBeaver 6.1.2 Community Edition版本与AnalyticDB for MySQL在连通性、列举数据库、创建表等方面的兼容性，并给出测试结果图。

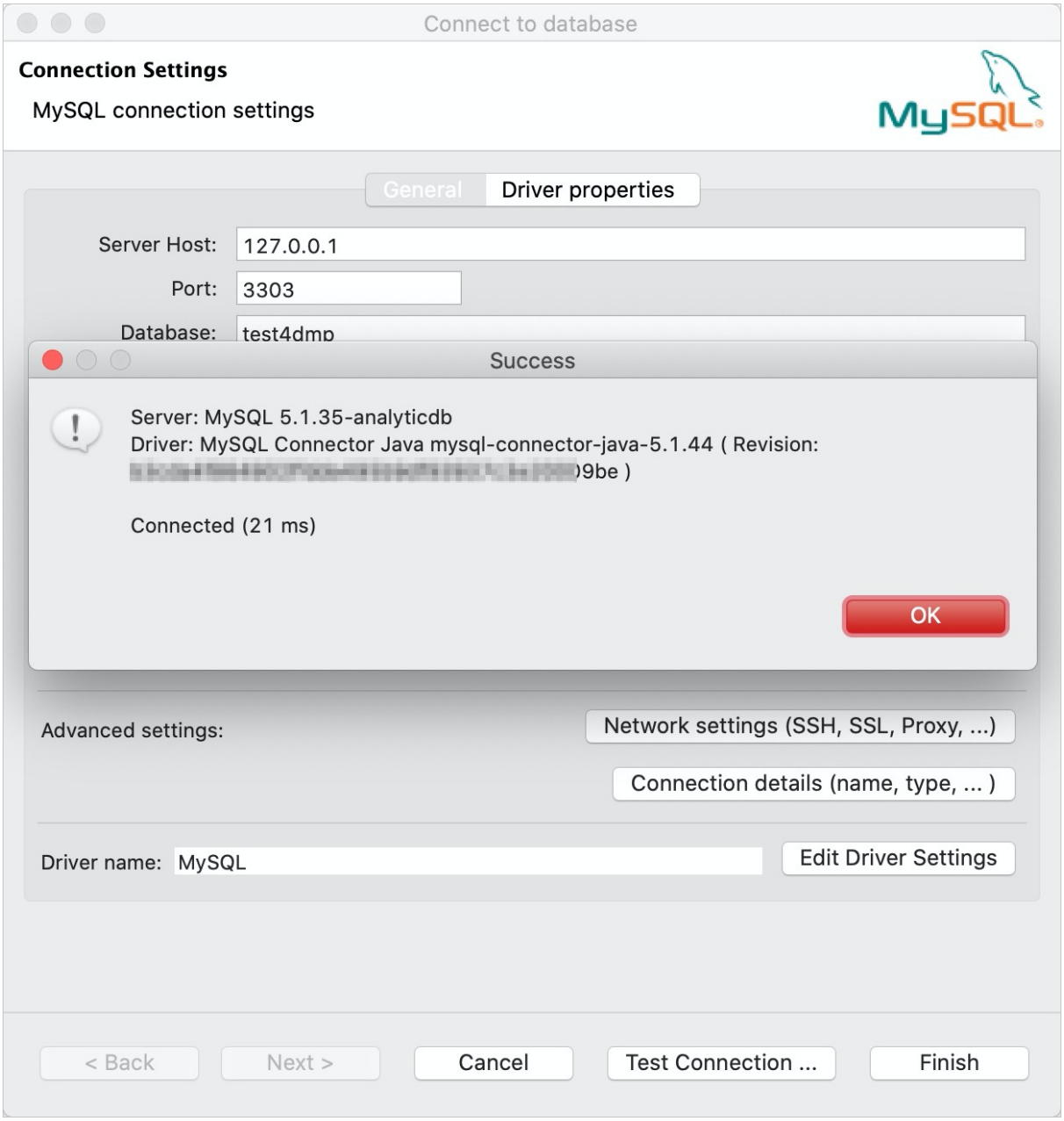
测试环境

Java	MySQL	AnalyticDB for MySQL	DBeaver
• java version "1.8.0_161" • Java(TM) SE Runtime Environment (build 1.8.0_161-b12) • Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)	mysql Ver5.6.46 for osx10.13 on x86_64 (Homebrew)	AnalyticDB for MySQL V3.1.0以上版本	版本为DBeaver 6.1.2 Community Edition

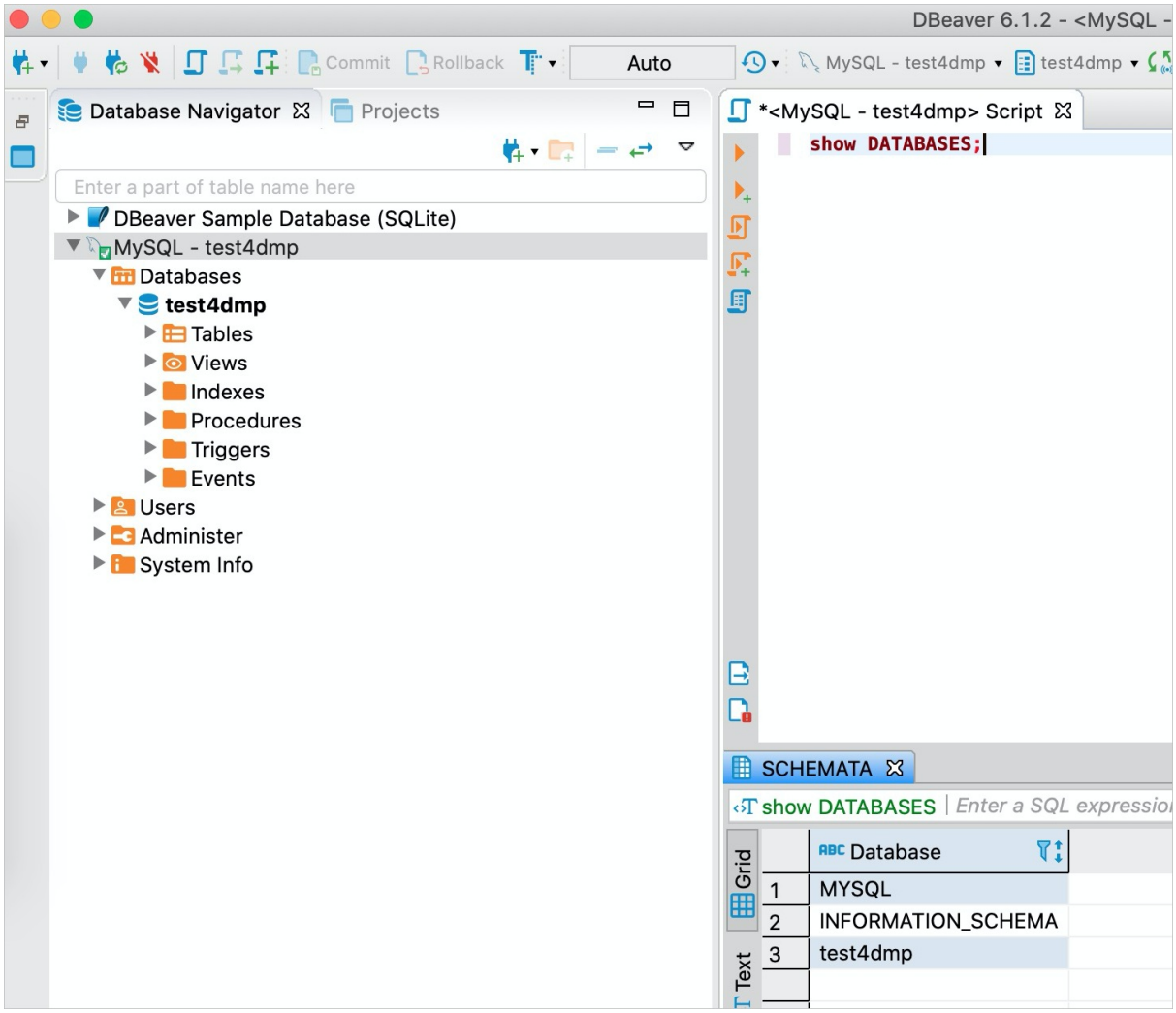
测试范围

- 连通性

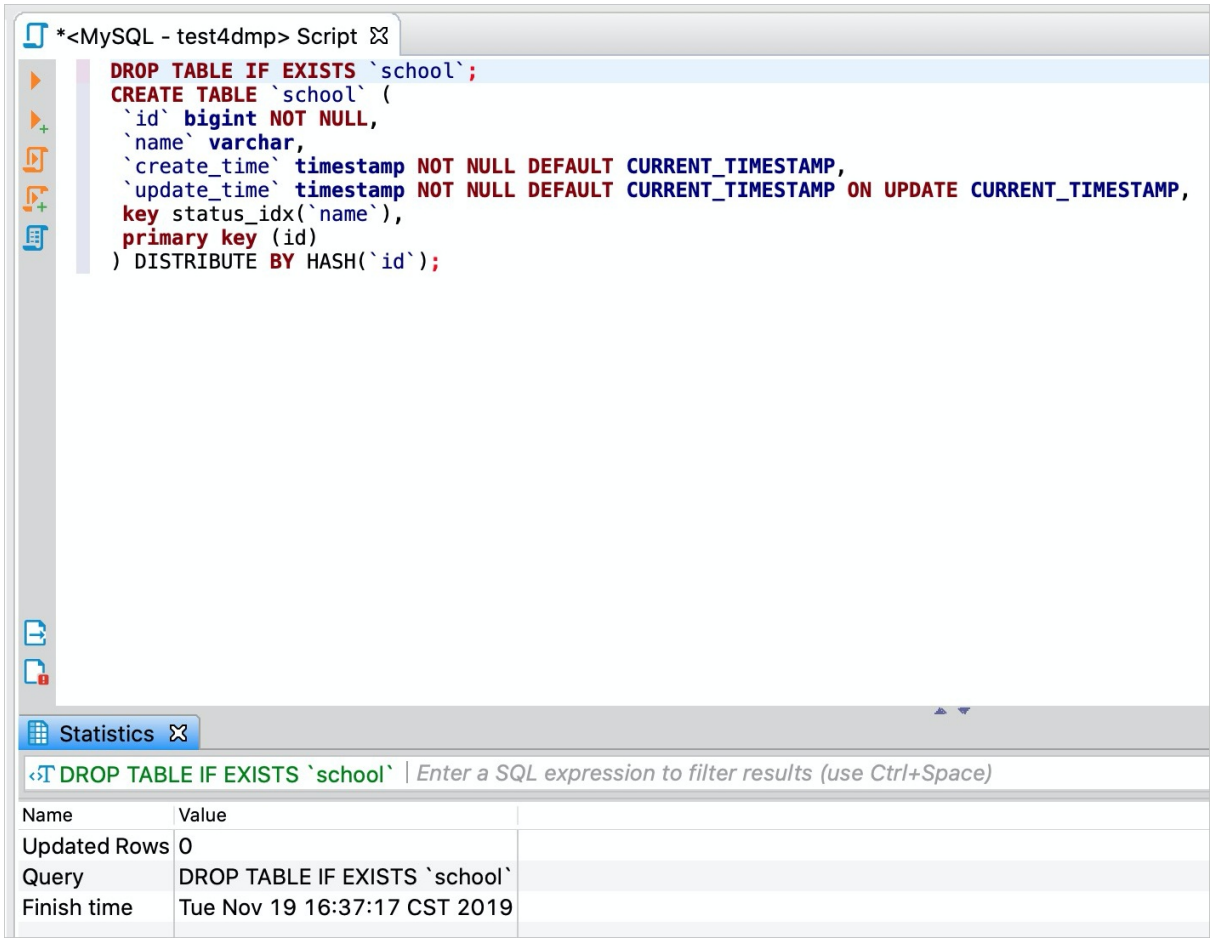

Connect to database
Connection Settings
MySQL connection settings
MySQL
General
Driver properties
Server Host:
127.0.0.1
Port:
3303
Database:
test4dmp
User name:
Password:
Save password locally
Server Time Zone:
Auto-detect
Local Client:
Advanced settings:
Network settings (SSH, SSL, Proxy, ...)
Connection details (name, type, ...)
Driver name:
MySQL
Edit Driver Settings
< Back
Next >
Cancel
Test Connection ...
Finish

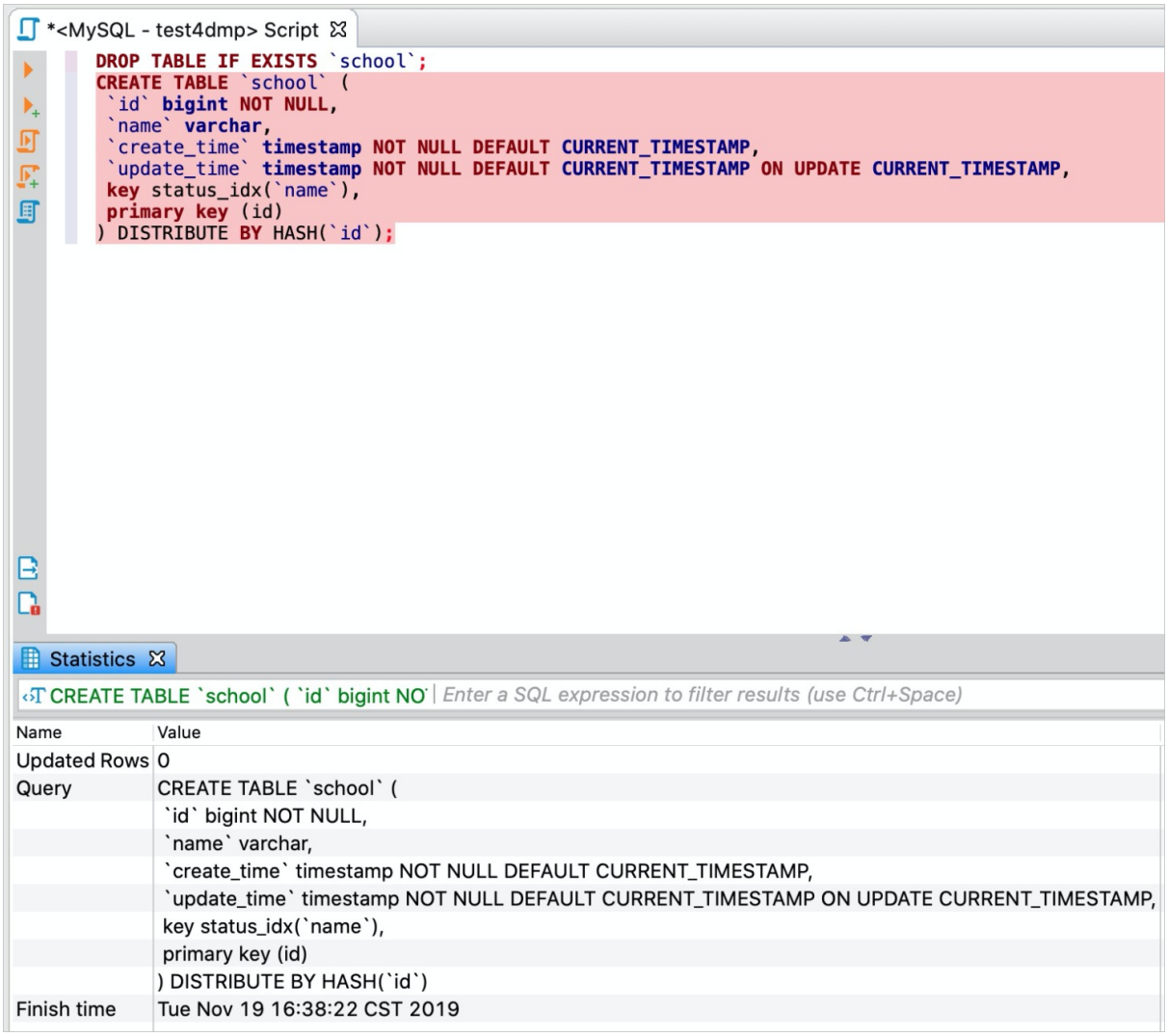

Connect to database
Connection Settings
MySQL connection settings
MySQL
General
Driver properties
Server Host:
127.0.0.1
Port:
3303
Database:
test4dmp
Success
Server: MySQL 5.1.35-analyticdb
Driver: MySQL Connector Java mysql-connector-java-5.1.44 (Revision:
9be)
Connected (21 ms)
OK
Advanced settings:
Network settings (SSH, SSL, Proxy, ...)
Connection details (name, type, ...)
Driver name:
MySQL
Edit Driver Settings
< Back
Next >
Cancel
Test Connection ...
Finish

- 列举数据库

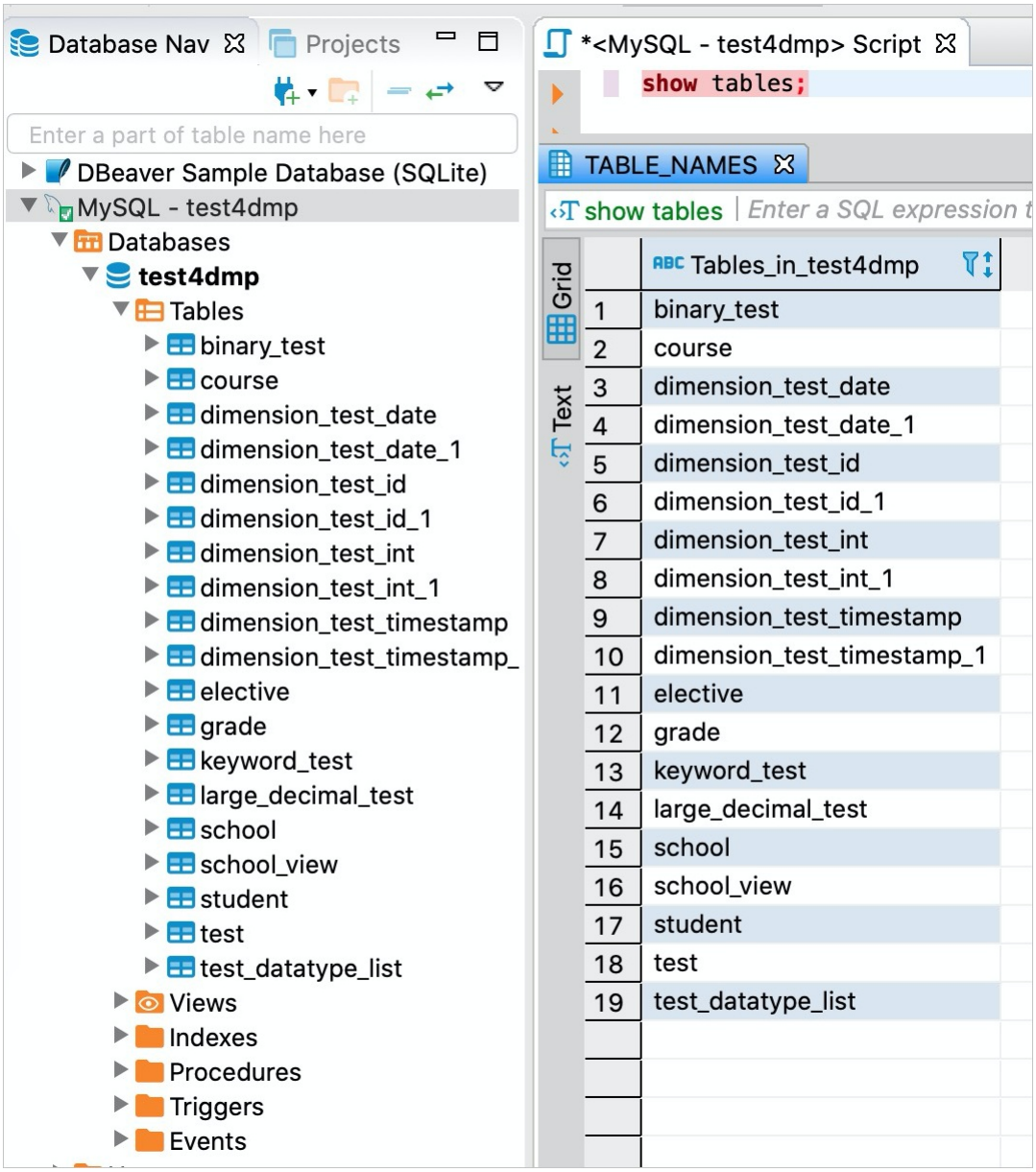

DBeaver 6.1.2 - <MySQL -
Commit
Rollback
Auto
MySQL - test4dmp
test4dmp
Database Navigator
Projects
Enter a part of table name here
DBeaver Sample Database (SQLite)
MySQL - test4dmp
Databases
test4dmp
Tables
Views
Indexes
Procedures
Triggers
Events
Users
Administer
System Info
*<MySQL - test4dmp> Script
show DATABASES;
SCHEMATA
show DATABASES
Enter a SQL expressio
Grid
Text
Database
1 MYSQL
2 INFORMATION_SCHEMA
3 test4dmp

- 创建表

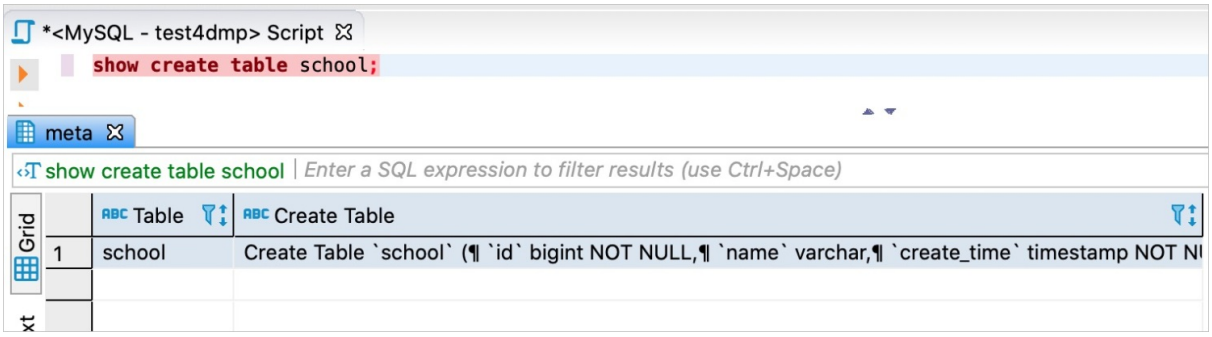

*<MySQL - test4dmp> Script
DROP TABLE IF EXISTS `school`;
CREATE TABLE `school` (
`id` bigint NOT NULL,
`name` varchar,
`create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
key status_idx(`name`),
primary key (id)
) DISTRIBUTE BY HASH(`id`);
Statistics
DROP TABLE IF EXISTS `school`
Enter a SQL expression to filter results (use Ctrl+Space)
Name
Value
Updated Rows
0
Query
DROP TABLE IF EXISTS `school`
Finish time
Tue Nov 19 16:37:17 CST 2019

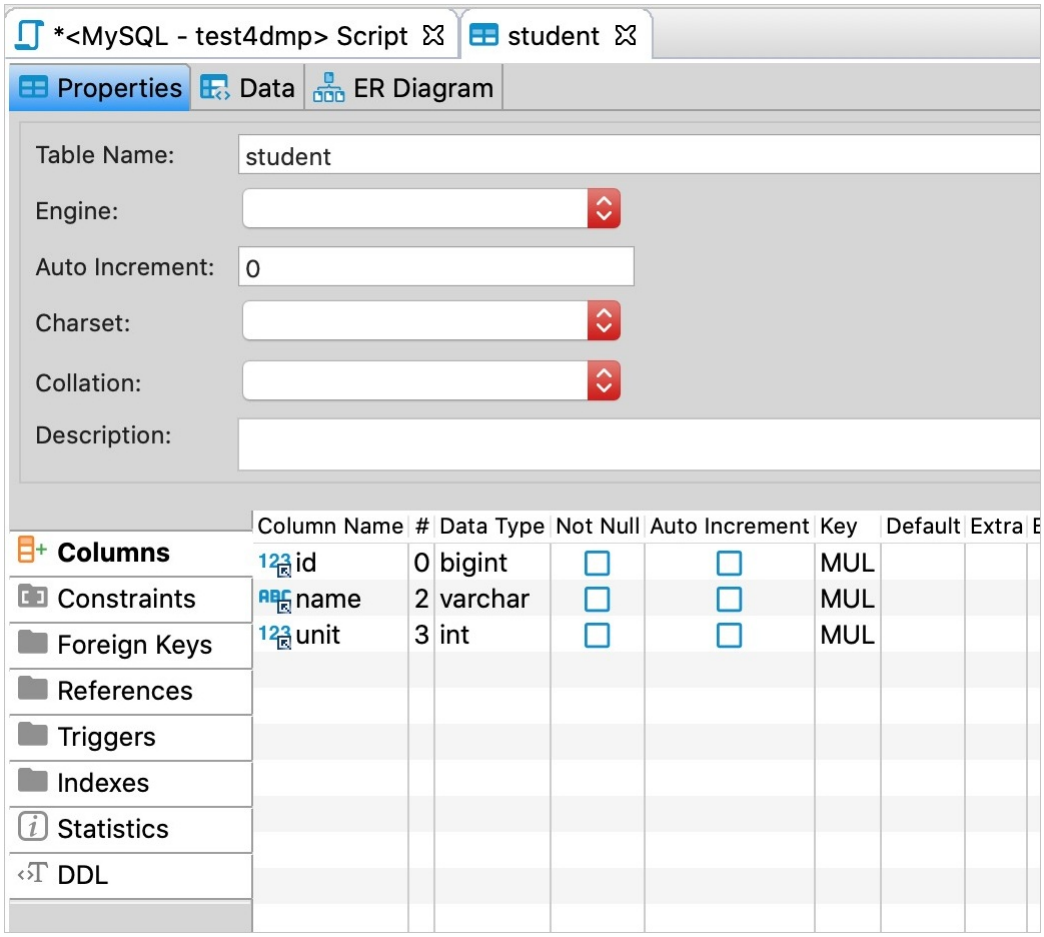

*<MySQL - test4dmp> Script
DROP TABLE IF EXISTS `school`;
CREATE TABLE `school` (
`id` bigint NOT NULL,
`name` varchar,
`create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
key status_idx(`name`),
primary key (id)
) DISTRIBUTE BY HASH(`id`);
Statistics
CREATE TABLE `school` (`id` bigint NO
Enter a SQL expression to filter results (use Ctrl+Space)
Name
Value
Updated Rows
0
Query
CREATE TABLE `school` (
`id` bigint NOT NULL,
`name` varchar,
`create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
key status_idx(`name`),
primary key (id)
) DISTRIBUTE BY HASH(`id`)
Finish time
Tue Nov 19 16:38:22 CST 2019

- 列举所有表

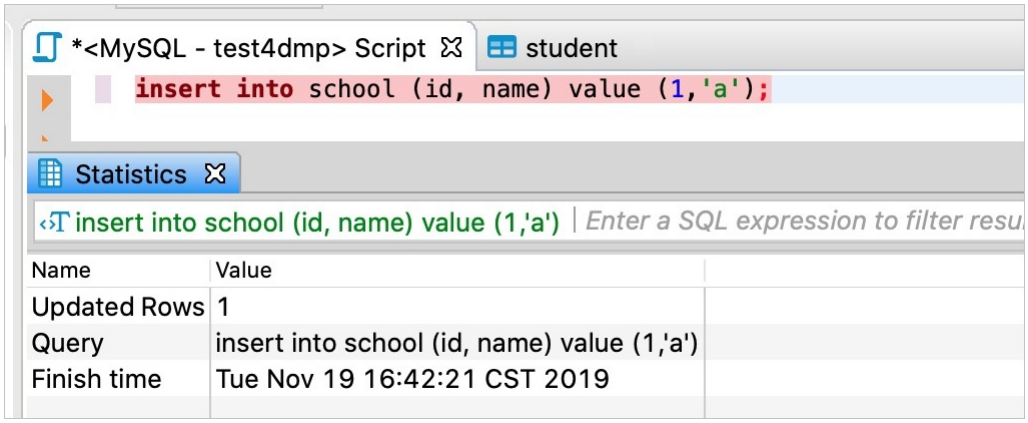

Database Nav
Projects
Enter a part of table name here
DBeaver Sample Database (SQLite)
MySQL - test4dmp
Databases
test4dmp
Tables
binary_test
course
dimension_test_date
dimension_test_date_1
dimension_test_id
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timestamp_
elective
grade
keyword_test
large_decimal_test
school
school_view
student
test
test_datatype_list
Views
Indexes
Procedures
Triggers
Events
*<MySQL - test4dmp> Script
show tables;
TABLE_NAMES
show tables
Enter a SQL expression t
Grid
Text
Tables_in_test4dmp
1 binary_test
2 course
3 dimension_test_date
4 dimension_test_date_1
5 dimension_test_id
6 dimension_test_id_1
7 dimension_test_int
8 dimension_test_int_1
9 dimension_test_timestamp
10 dimension_test_timestamp_1
11 elective
12 grade
13 keyword_test
14 large_decimal_test
15 school
16 school_view
17 student
18 test
19 test_datatype_list

- 查看表结构

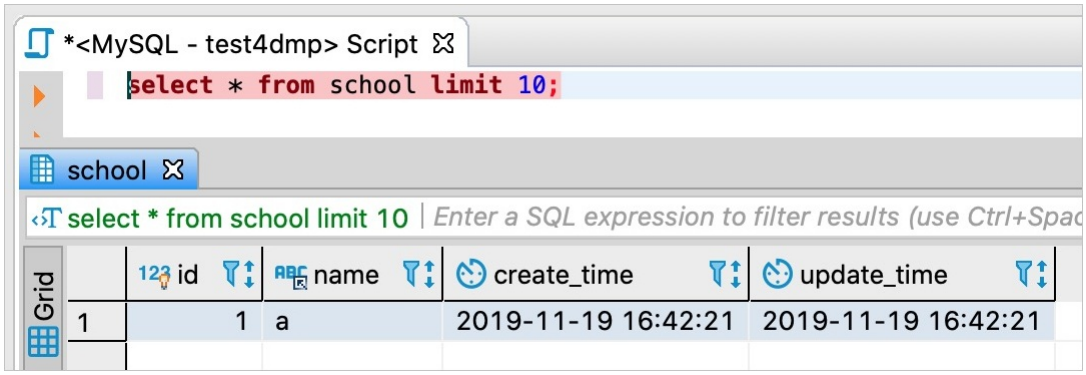

*<MySQL - test4dmp> Script
show create table school;
meta
show create table school
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Table
Create Table
1 school
Create Table `school` (¶ `id` bigint NOT NULL,¶ `name` varchar,¶ `create_time` timestamp NOT NU

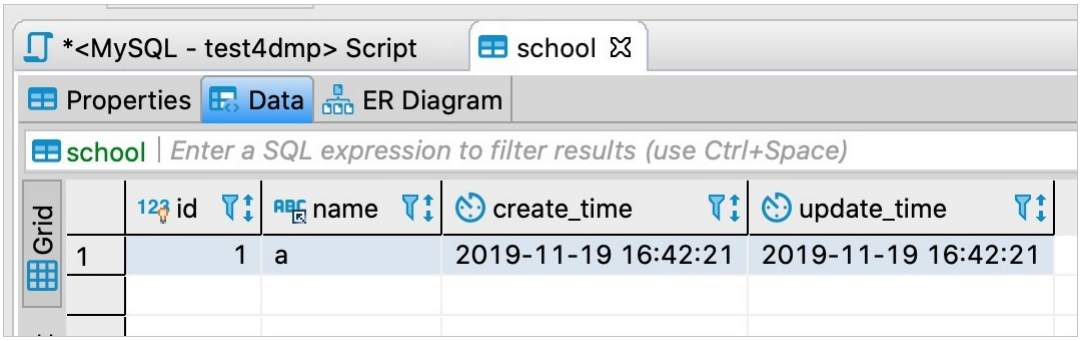


- 向表中写入数据

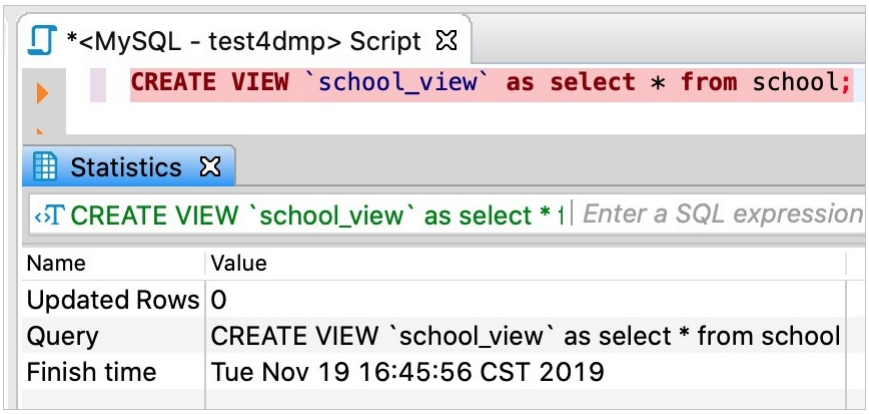


- 查看表数据

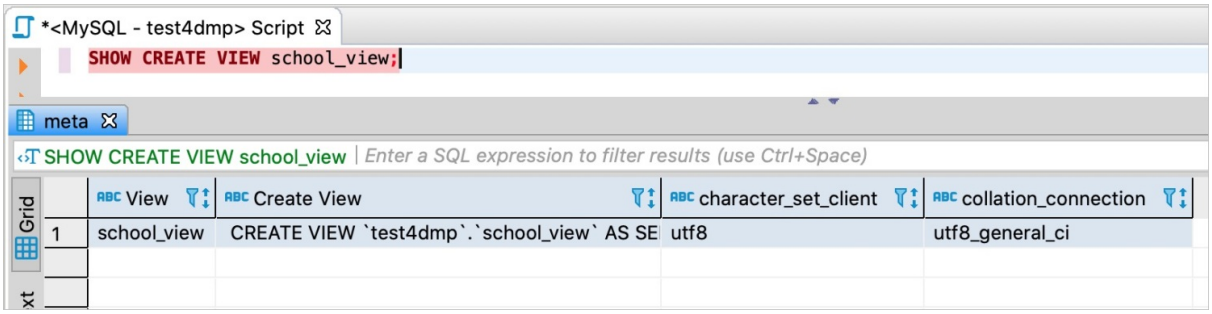


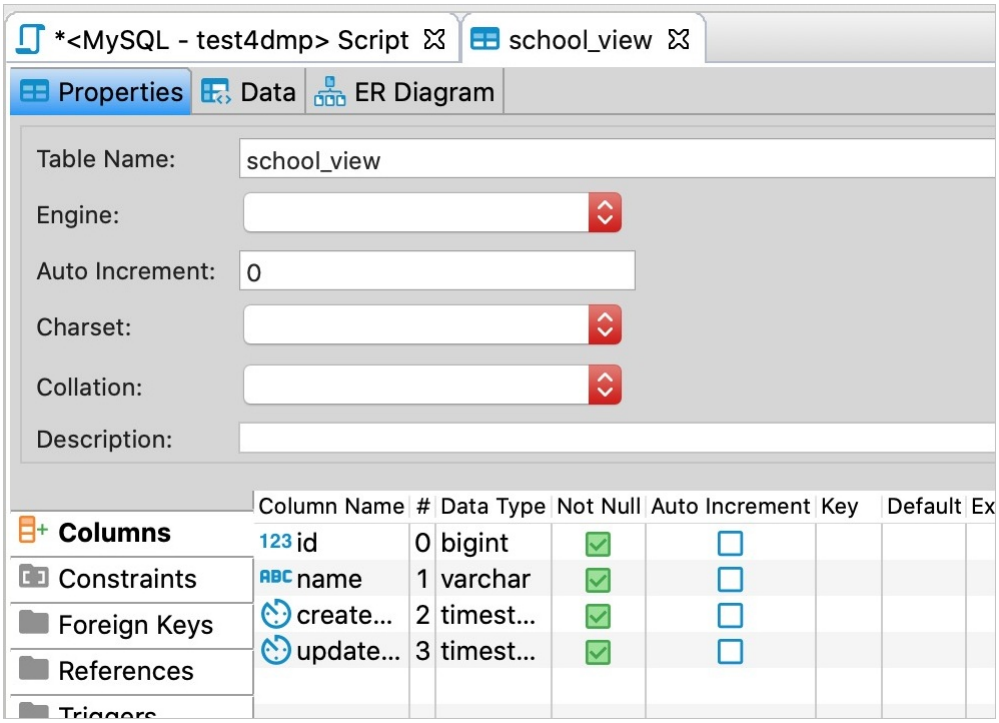

*<MySQL - test4dmp> Script
school
Properties
Data
ER Diagram
school
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
id
name
create_time
update_time
1
1
a
2019-11-19 16:42:21
2019-11-19 16:42:21

- 创建视图

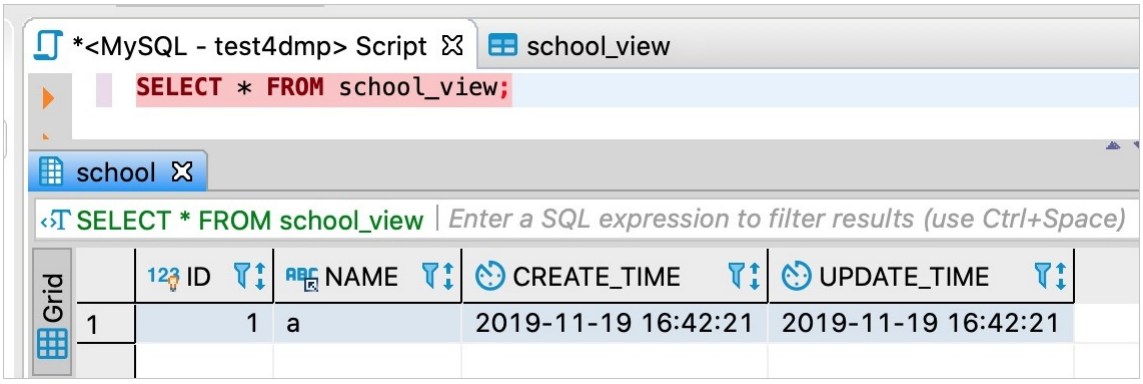

*<MySQL - test4dmp> Script
CREATE VIEW `school_view` as select * from school;
Statistics
CREATE VIEW `school_view` as select * f
Enter a SQL expression
Name
Value
Updated Rows
0
Query
CREATE VIEW `school_view` as select * from school
Finish time
Tue Nov 19 16:45:56 CST 2019

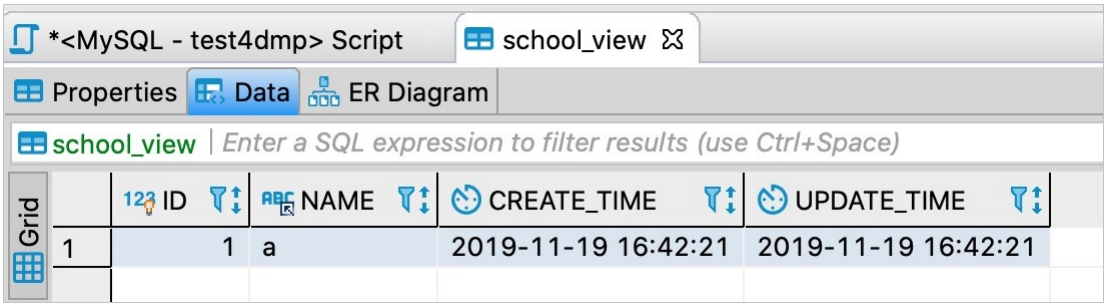
- 查看视图结构


*<MySQL - test4dmp> Script
SHOW CREATE VIEW school_view;
meta
SHOW CREATE VIEW school_view
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
View
Create View
character_set_client
collation_connection
1
school_view
CREATE VIEW `test4dmp`.`school_view` AS SE
utf8
utf8_general_ci



- 查询视图





2.4. DBVisualizer

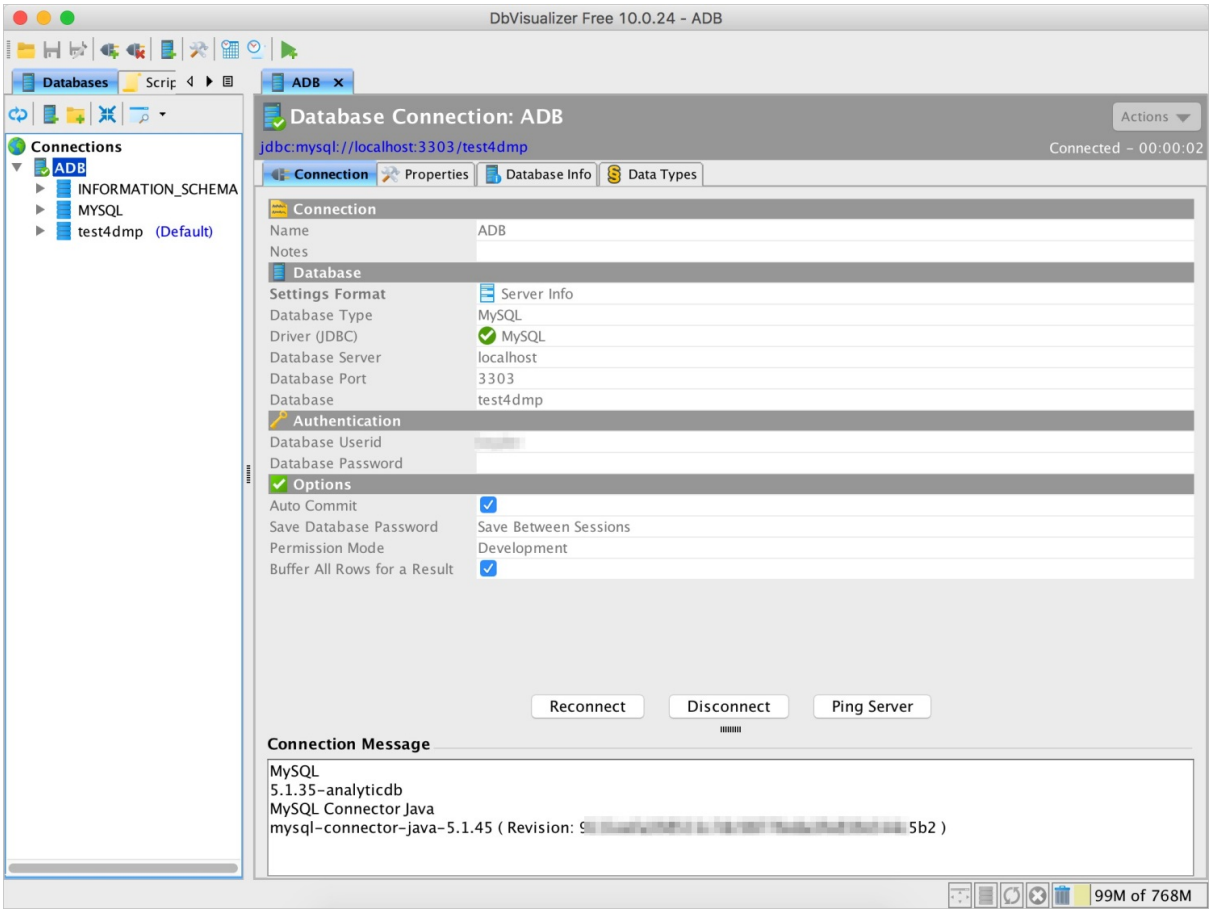
本文测试了DBVisualizer 10.0.24 版本与AnalyticDB for MySQL在连通性、列举数据库、创建表等方面的兼容性，并给出测试结果图。

测试环境

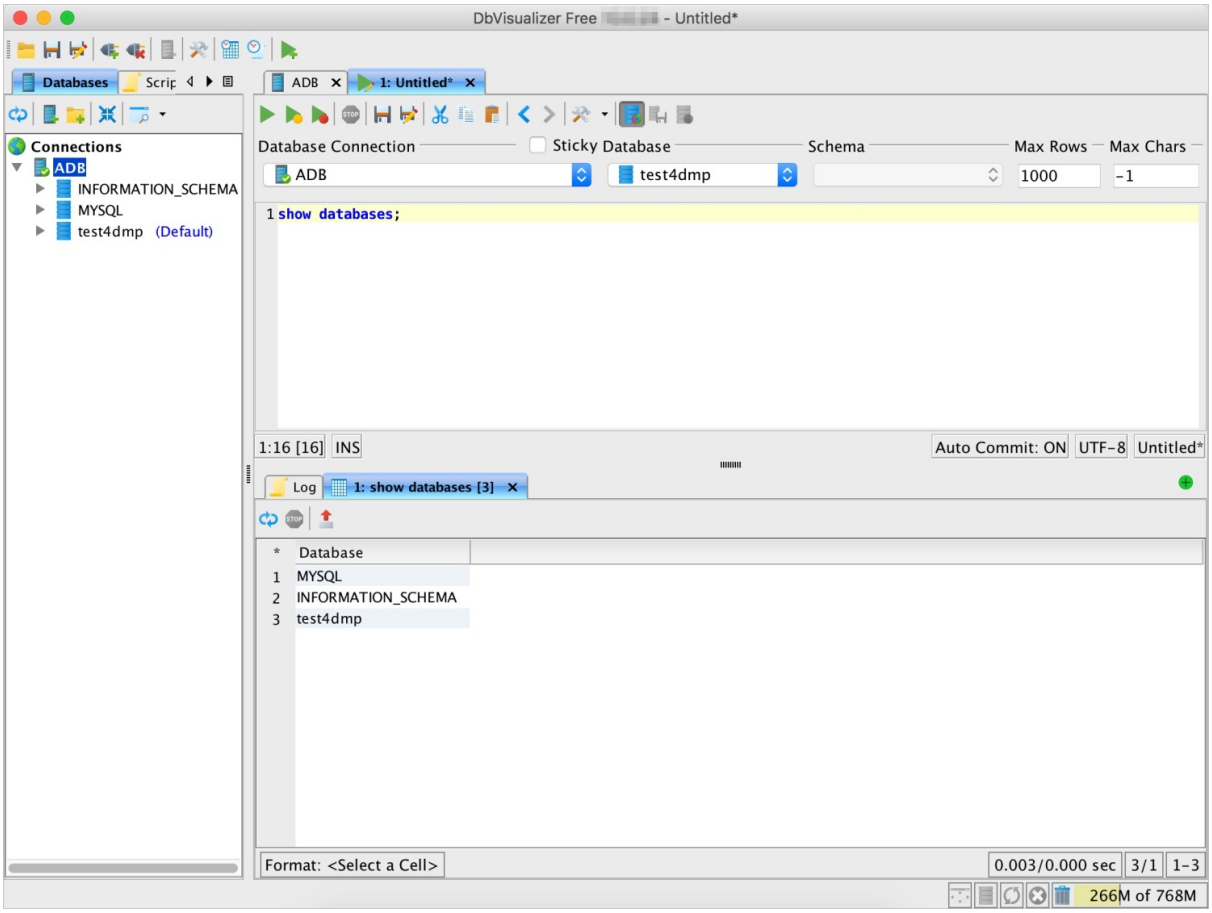
Java	MySQL
• java version "1.8.0_161" • Java(TM) SE Runtime Environment (build 1.8.0_161-b12) • Java HotSpot(TM) 64-Bit Server VM (build 25.161-b12, mixed mode)	mysql Ver5.6.46 for osx10.13 on x86_64 (Homebrew)

测试范围

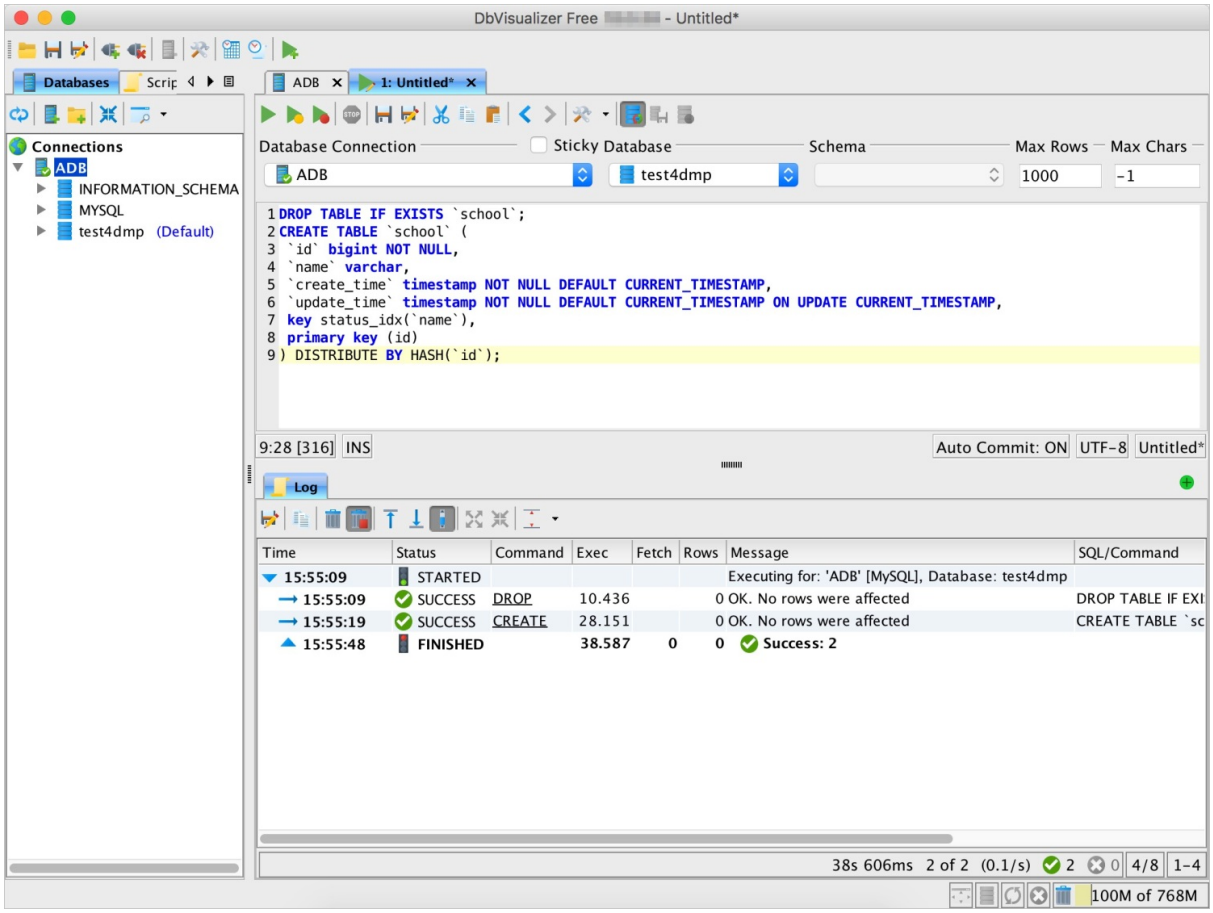
- 连通性


DbVisualizer Free 10.0.24 - ADB
Databases
Scrip
ADB
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
Database Connection: ADB
Actions
jdbc:mysql://localhost:3303/test4dmp
Connected - 00:00:02
Connection
Properties
Database Info
Data Types
Connection
Name
ADB
Notes
Database
Settings Format
Server Info
Database Type
MySQL
Driver (JDBC)
MySQL
Database Server
localhost
Database Port
3303
Database
test4dmp
Authentication
Database Userid
Database Password
Options
Auto Commit
Save Database Password
Save Between Sessions
Permission Mode
Development
Buffer All Rows for a Result
Reconnect
Disconnect
Ping Server
Connection Message
MySQL
5.1.35-analyticdb
MySQL Connector Java
mysql-connector-java-5.1.45 (Revision:
5b2)
99M of 768M

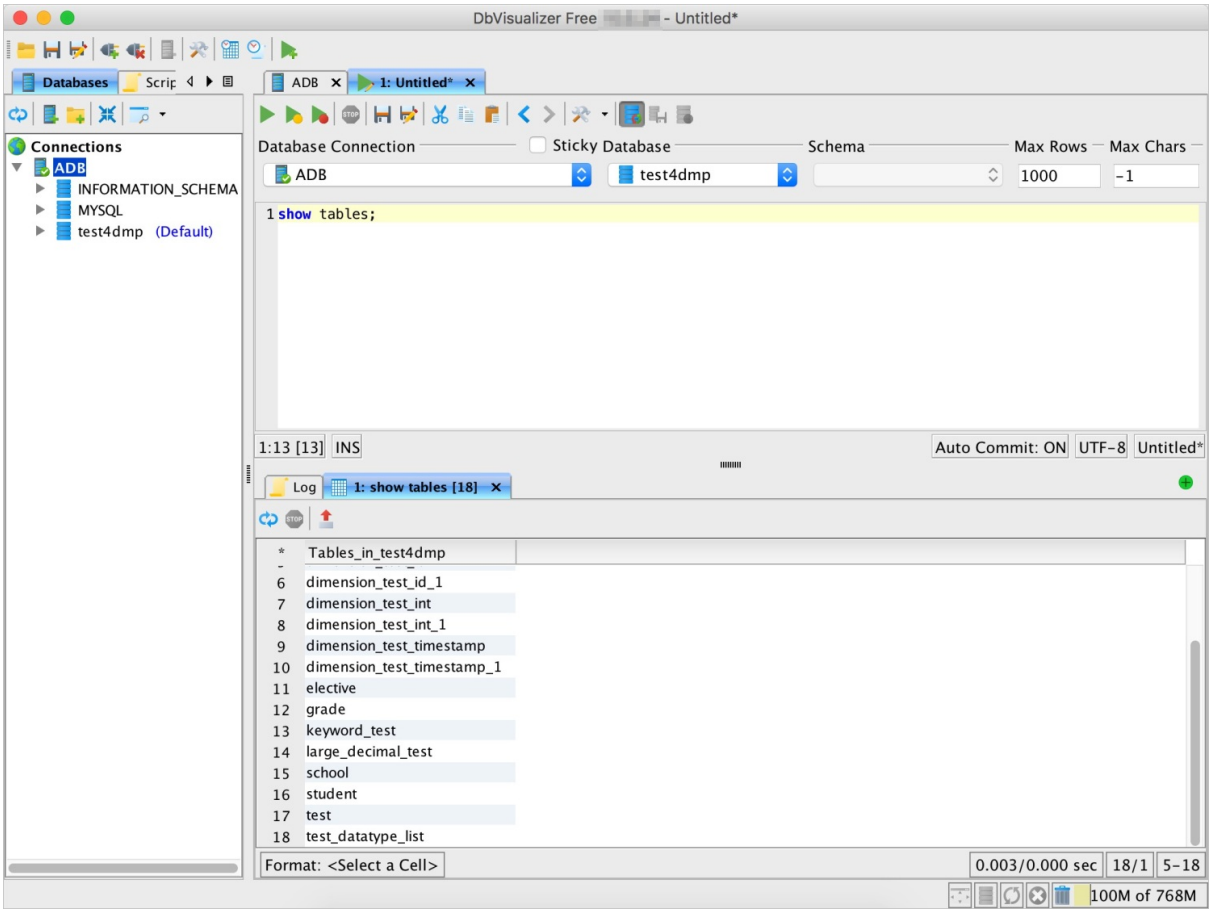
- 列举数据库


DbVisualizer Free - Untitled*
Databases
Scrip
ADB
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 show databases;
1:16 [16]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
1: show databases [3]
Database
1 MYSQL
2 INFORMATION_SCHEMA
3 test4dmp
Format: <Select a Cell>
0.003/0.000 sec
3/1
1-3
266M of 768M

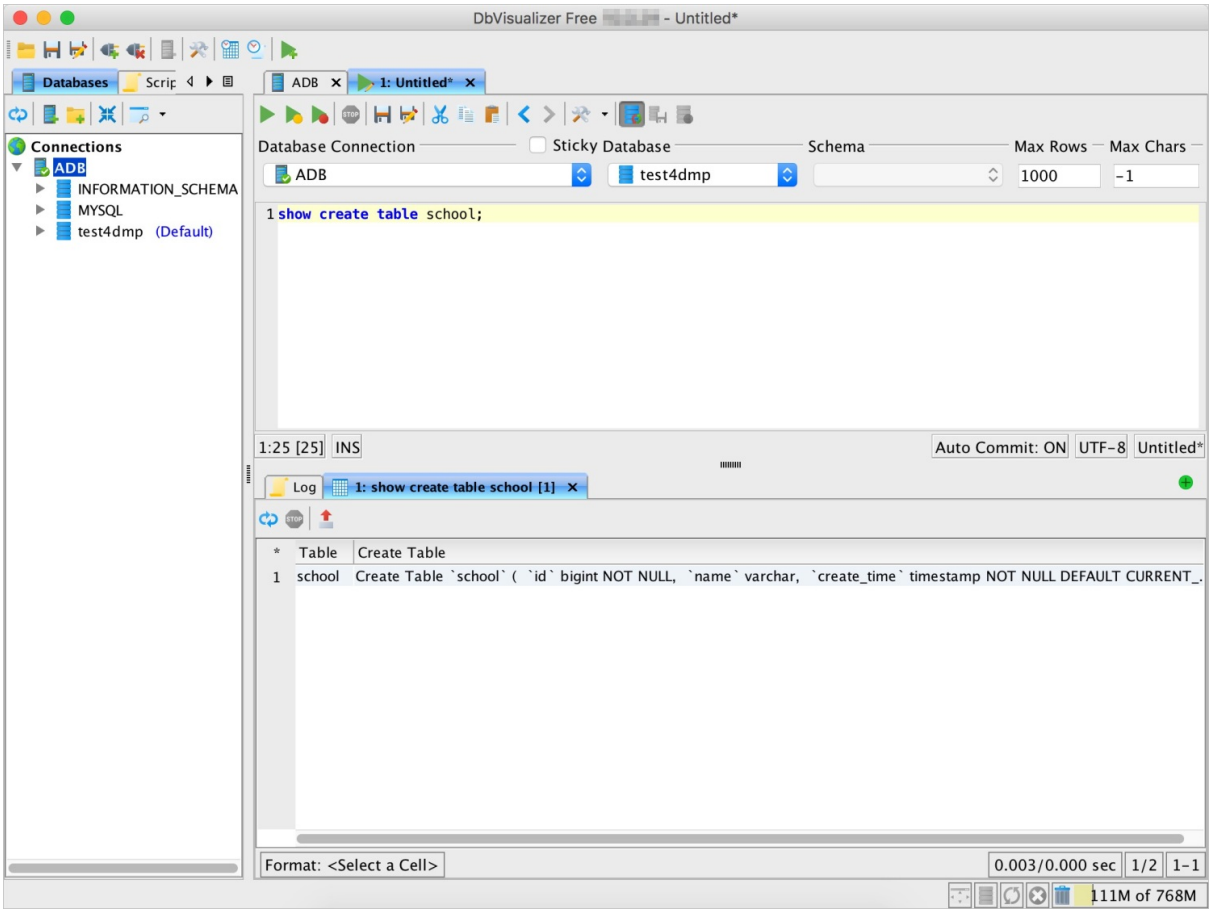
- 创建表


DbVisualizer Free - Untitled*
Databases
ADB
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 DROP TABLE IF EXISTS `school`;
2 CREATE TABLE `school` (
3 `id` bigint NOT NULL,
4 `name` varchar,
5 `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
6 `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
7 key status_idx(`name`),
8 primary key (id)
9) DISTRIBUTE BY HASH(`id`);
9:28 [316]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
Time
Status
Command
Exec
Fetch
Rows
Message
SQL/Command
15:55:09 STARTED Executing for: 'ADB' [MySQL], Database: test4dmp
15:55:09 SUCCESS DROP 10.436 0 OK. No rows were affected DROP TABLE IF EXI
15:55:19 SUCCESS CREATE 28.151 0 OK. No rows were affected CREATE TABLE `sc
15:55:48 FINISHED 38.587 0 0 Success: 2
38s 606ms 2 of 2 (0.1/s) 2 0 4/8 1-4
100M of 768M

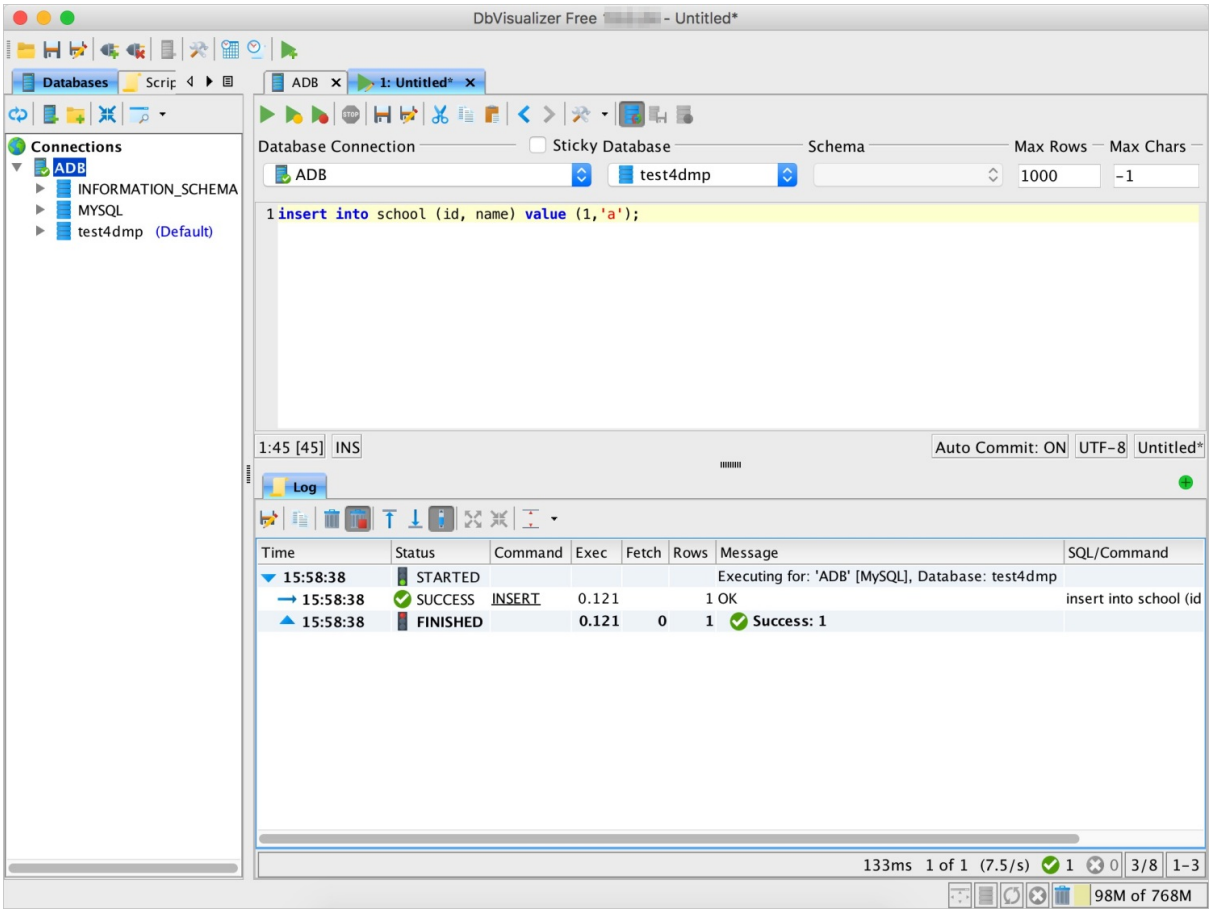
- 列举所有表


DbVisualizer Free - Untitled*
Databases
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
ADB
1: Untitled*
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 show tables;
1:13 [13]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
1: show tables [18]
Tables_in_test4dmp
6 dimension_test_id_1
7 dimension_test_int
8 dimension_test_int_1
9 dimension_test_timestamp
10 dimension_test_timestamp_1
11 elective
12 grade
13 keyword_test
14 large_decimal_test
15 school
16 student
17 test
18 test_datatype_list
Format: <Select a Cell>
0.003/0.000 sec
18/1
5-18
100M of 768M

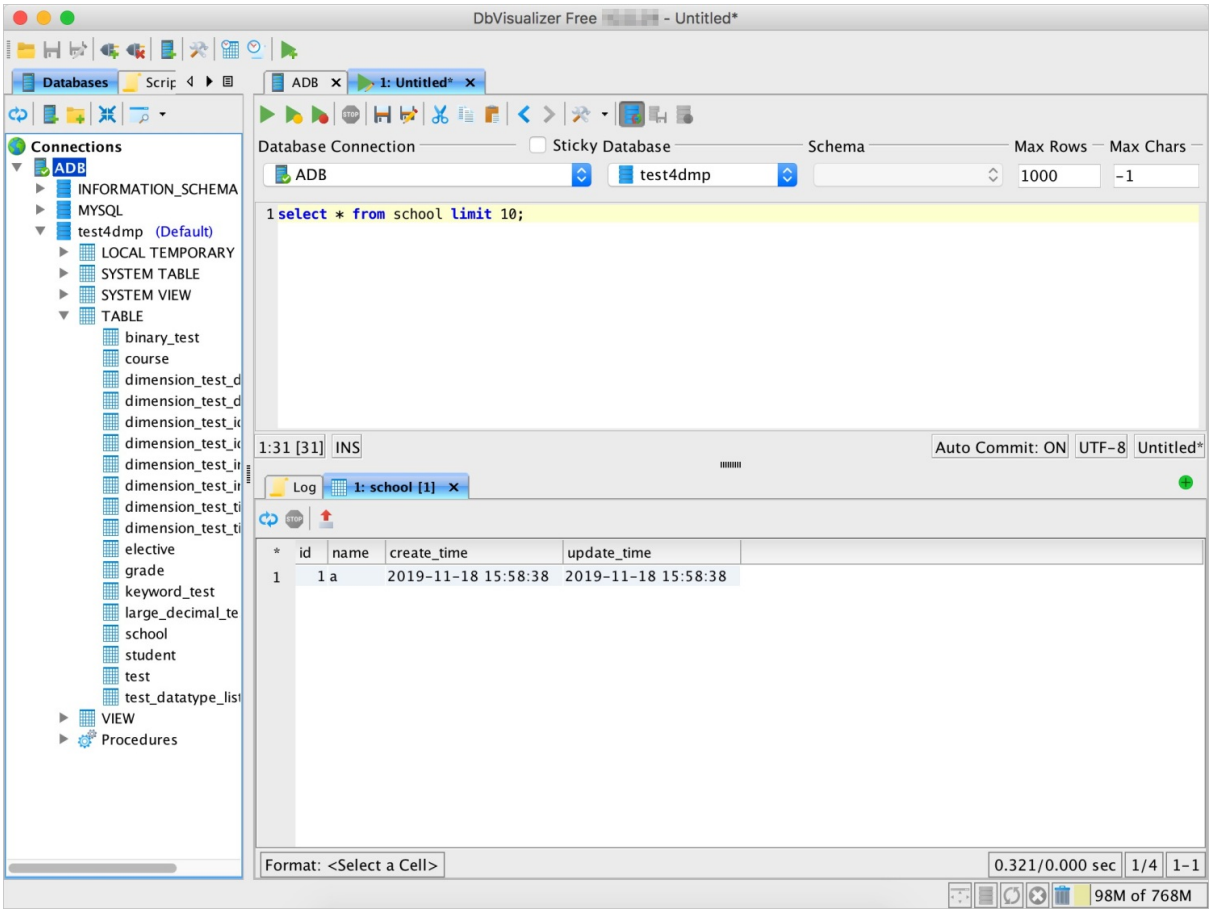
- 查看表结构


DbVisualizer Free - Untitled*
Databases
Scrip
ADB
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 show create table school;
1:25 [25]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
1: show create table school [1]
Table
Create Table
1
school
Create Table `school` (`id` bigint NOT NULL, `name` varchar, `create_time` timestamp NOT NULL DEFAULT CURRENT_.
Format: <Select a Cell>
0.003/0.000 sec
1/2
1-1
111M of 768M

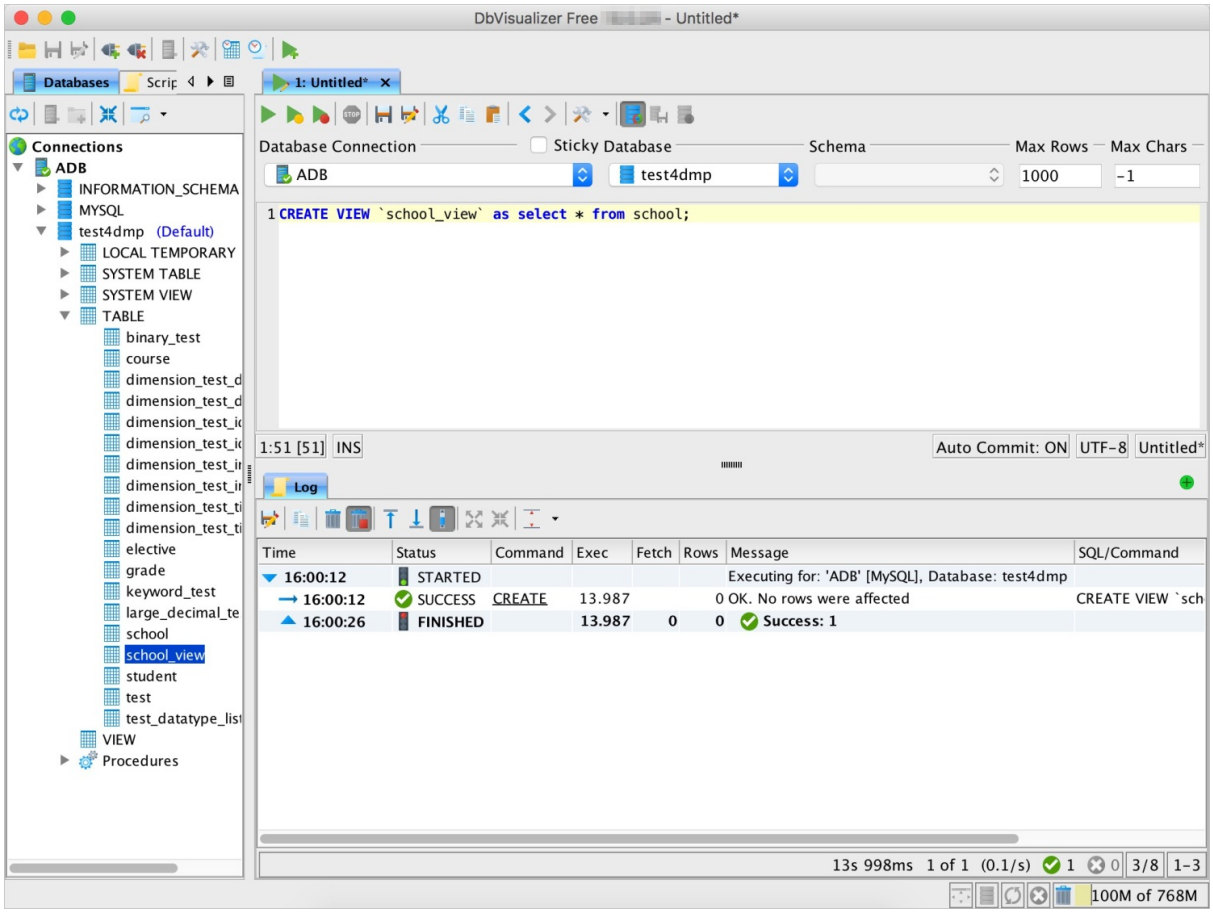
- 向表中写入数据


DbVisualizer Free - Untitled*
Databases
ADB
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 insert into school (id, name) value (1,'a');
1:45 [45]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
Time
Status
Command
Exec
Fetch
Rows
Message
SQL/Command
15:58:38 STARTED Executing for: 'ADB' [MySQL], Database: test4dmp
15:58:38 SUCCESS INSERT 0.121 1 OK insert into school (id
15:58:38 FINISHED 0.121 0 1 Success: 1
133ms 1 of 1 (7.5/s) 1 0 3/8 1-3
98M of 768M

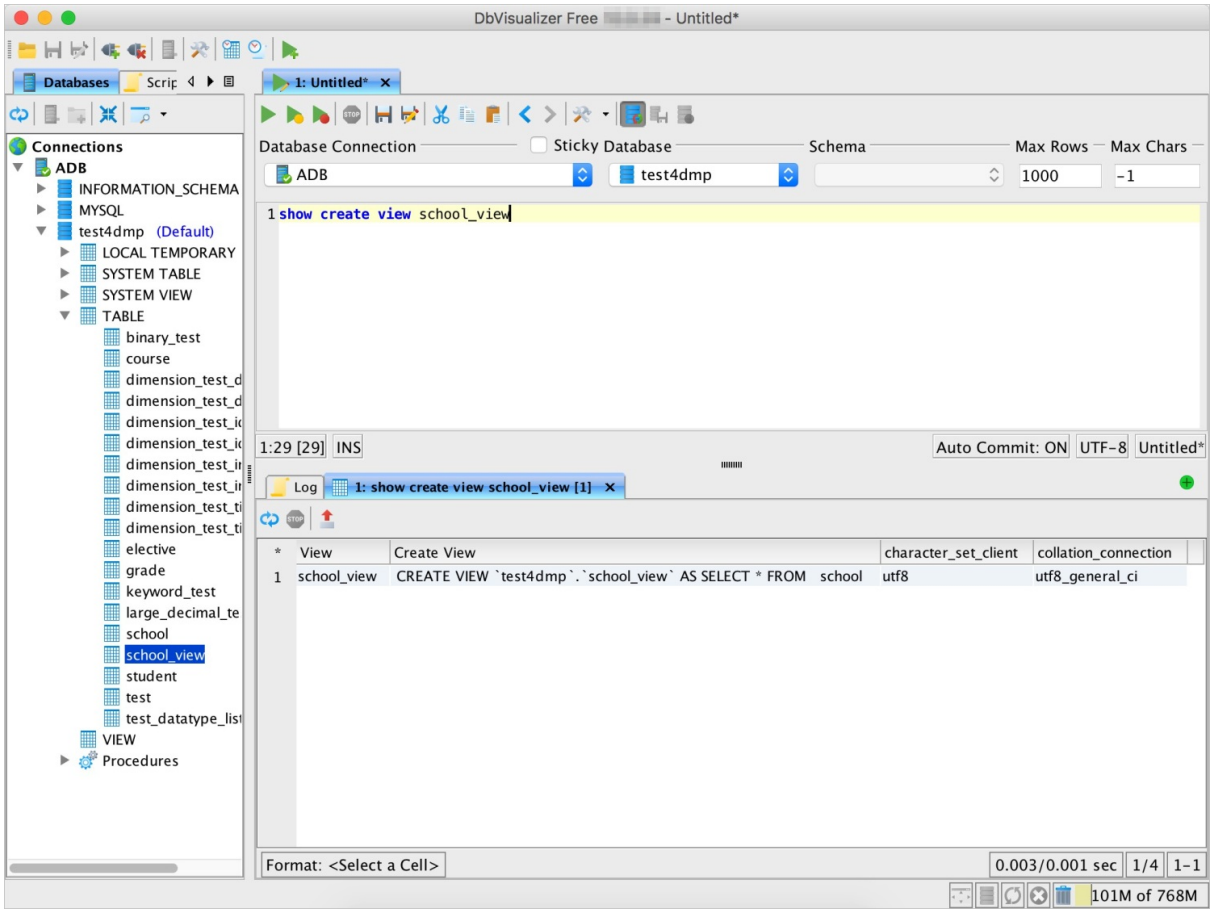
- 查看表数据


DbVisualizer Free - Untitled*
Databases
ADB
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
LOCAL TEMPORARY
SYSTEM TABLE
SYSTEM VIEW
TABLE
binary_test
course
elective
grade
keyword_test
large_decimal_te
school
student
test
VIEW
Procedures
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
1000
-1
1 select * from school limit 10;
1:31 [31]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
1: school [1]
id
name
create_time
update_time
1 a
2019-11-18 15:58:38
2019-11-18 15:58:38
Format: <Select a Cell>
0.321/0.000 sec
1/4
1-1
98M of 768M

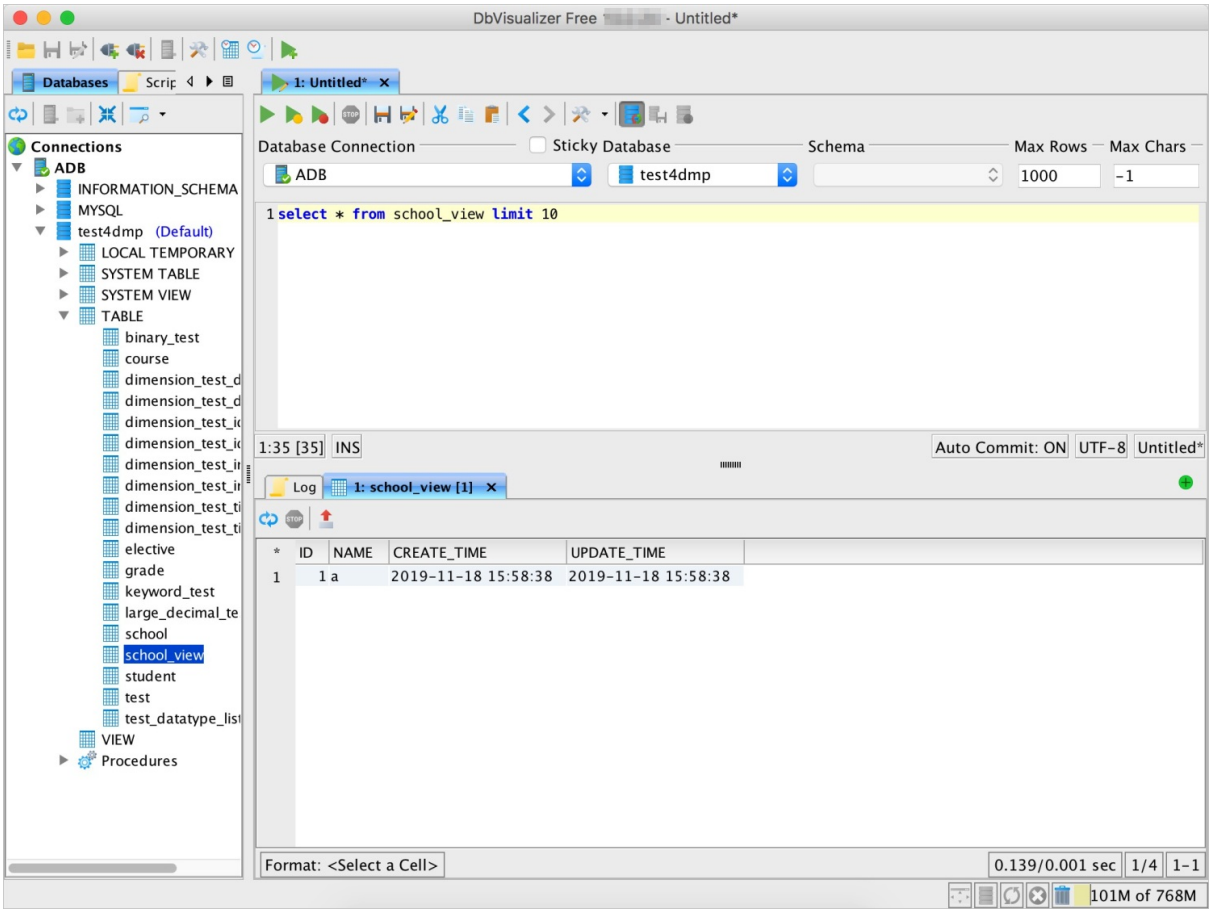
- 创建视图


DbVisualizer Free - Untitled*
Databases
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
LOCAL TEMPORARY
SYSTEM TABLE
SYSTEM VIEW
TABLE
binary_test
course
elective
grade
keyword_test
large_decimal_te
school
school_view
student
test
VIEW
Procedures
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 CREATE VIEW `school_view` as select * from school;
1:51 [51]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
Time
Status
Command
Exec
Fetch
Rows
Message
SQL/Command
16:00:12 STARTED Executing for: 'ADB' [MySQL], Database: test4dmp
16:00:12 SUCCESS CREATE 13.987 0 OK. No rows were affected CREATE VIEW `sch
16:00:26 FINISHED 13.987 0 0 Success: 1
13s 998ms 1 of 1 (0.1/s)
3/8
1-3
100M of 768M

- 查看视图结构


DbVisualizer Free - Untitled*
Databases
1: Untitled*
Connections
ADB
INFORMATION_SCHEMA
MYSQL
test4dmp (Default)
LOCAL TEMPORARY
SYSTEM TABLE
SYSTEM VIEW
TABLE
binary_test
course
elective
grade
keyword_test
large_decimal_te
school
school_view
student
test
test_datatype_list
VIEW
Procedures
Database Connection
Sticky Database
Schema
Max Rows
Max Chars
ADB
test4dmp
1000
-1
1 show create view school_view
1:29 [29]
INS
Auto Commit: ON
UTF-8
Untitled*
Log
1: show create view school_view [1]
View
Create View
character_set_client
collation_connection
1
school_view
CREATE VIEW `test4dmp`.`school_view` AS SELECT * FROM school
utf8
utf8_general_ci
Format: <Select a Cell>
0.003/0.001 sec
1/4
1-1
101M of 768M

- 查询视图



2.5. SQL WorkBench/J

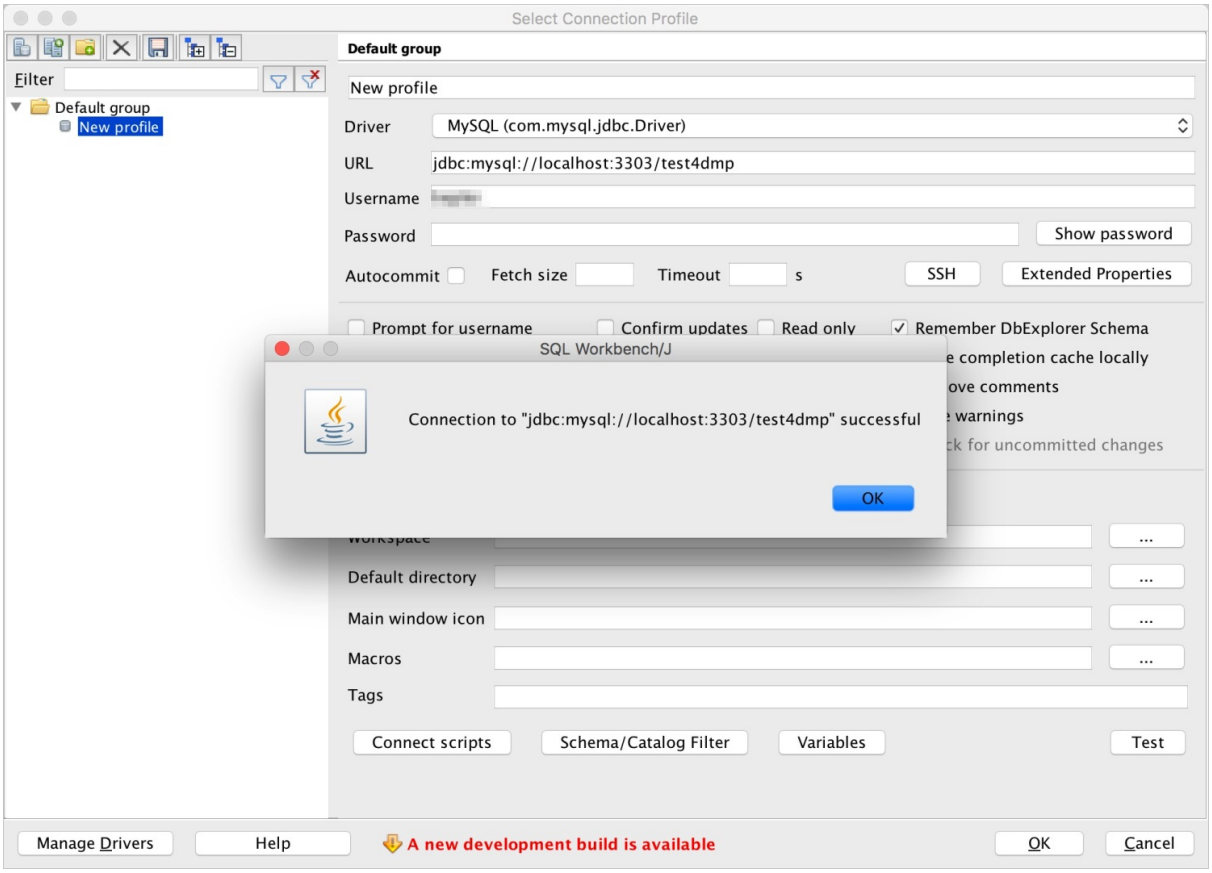
本文测试了SQL WorkBench/J与AnalyticDB MySQL版在连通性、列举数据库、创建表等方面的兼容性，并给出测试结果图。

测试环境

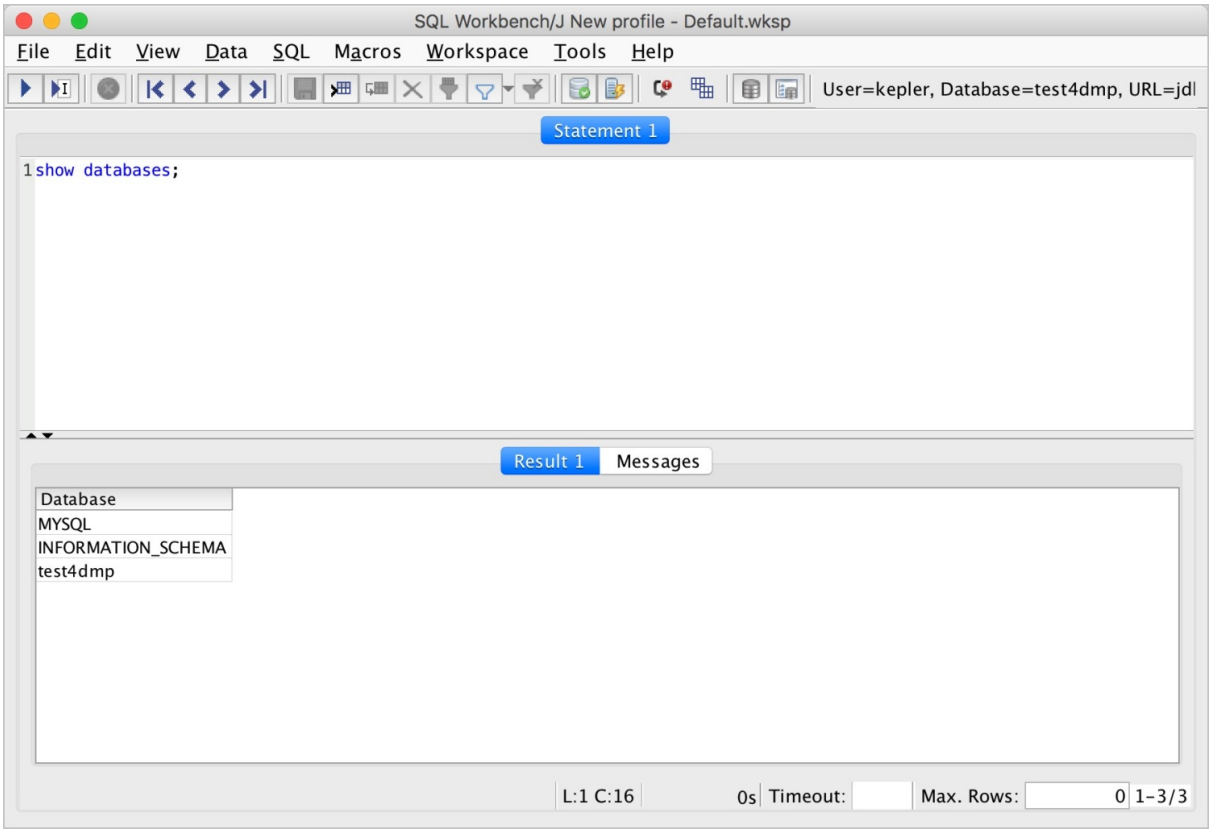
MySQL JDBC Driver	SQL WorkBench/J
MySQL JDBC Driver 5.1.48 (Platform Independent)，下载地址为MySQL JDBC Driver。	下载地址为SQL WorkBench/J。

测试范围

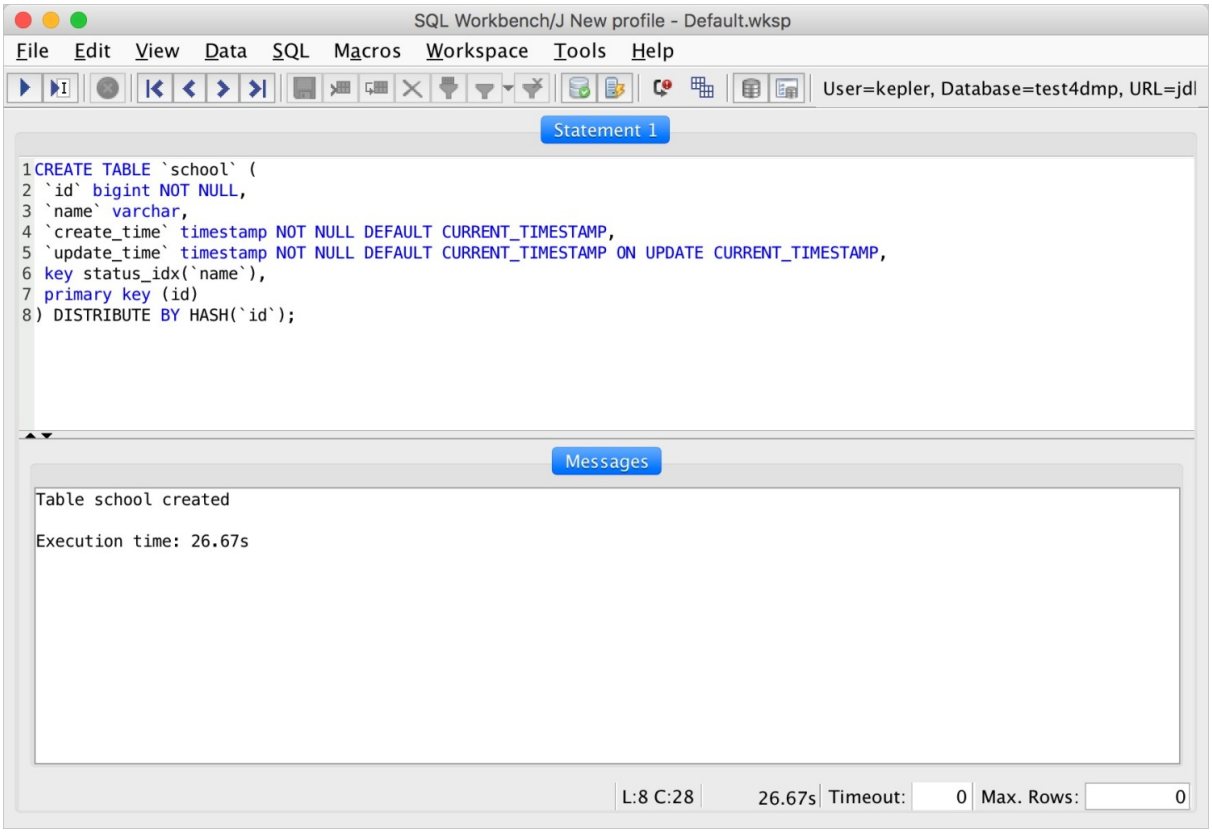
- 连通性


Select Connection Profile
Filter
Default group
New profile
Default group
New profile
Driver
MySQL (com.mysql.jdbc.Driver)
URL
jdbc:mysql://localhost:3303/test4dmp
Username
Password
Show password
Autocommit
Fetch size
Timeout
s
SSH
Extended Properties
Prompt for username
Confirm updates
Read only
Remember DbExplorer Schema
SQL Workbench/J
Connection to "jdbc:mysql://localhost:3303/test4dmp" successful
OK
Default directory
Main window icon
Macros
Tags
Connect scripts
Schema/Catalog Filter
Variables
Test
Manage Drivers
Help
A new development build is available
OK
Cancel

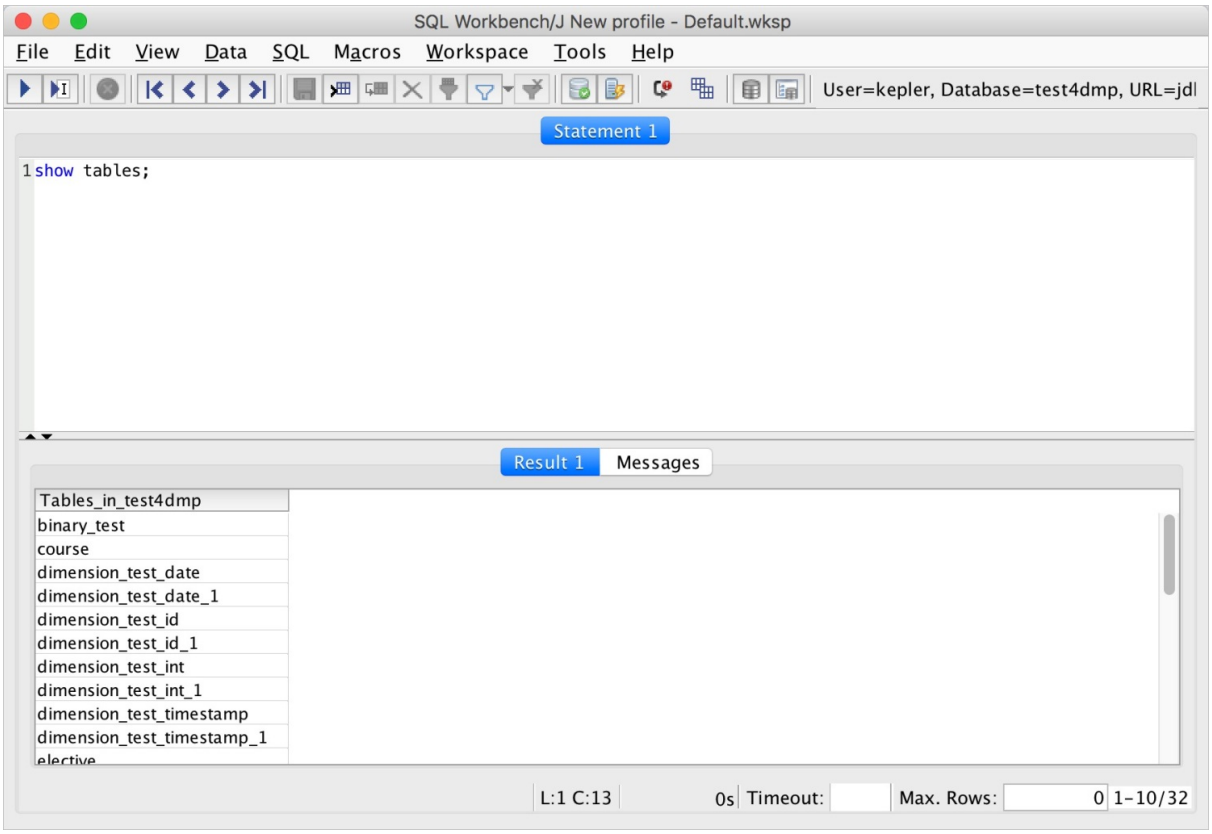
- 列举数据库


SQL Workbench/J New profile - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
User=kepler, Database=test4dmp, URL=jdl
Statement 1
1 show databases;
Result 1
Messages
Database
MYSQL
INFORMATION_SCHEMA
test4dmp
L:1 C:16
0s
Timeout:
Max. Rows:
0
1-3/3

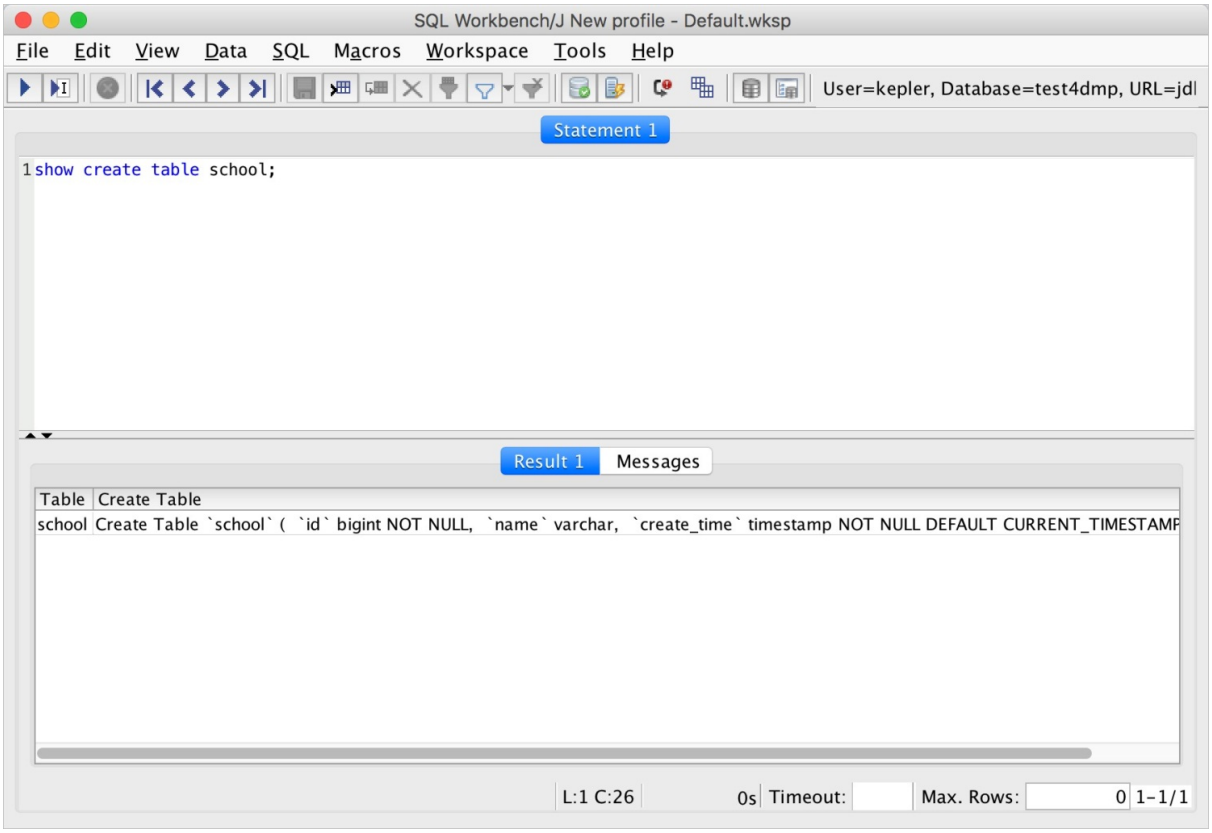
- 创建表


SQL Workbench/J New profile - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
User=kepler, Database=test4dmp, URL=jdl
Statement 1
1 CREATE TABLE `school` (
2 `id` bigint NOT NULL,
3 `name` varchar,
4 `create_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
5 `update_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
6 key status_idx(`name`),
7 primary key (id)
8) DISTRIBUTE BY HASH(`id`);
Messages
Table school created
Execution time: 26.67s
L:8 C:28 26.67s Timeout: 0 Max. Rows: 0

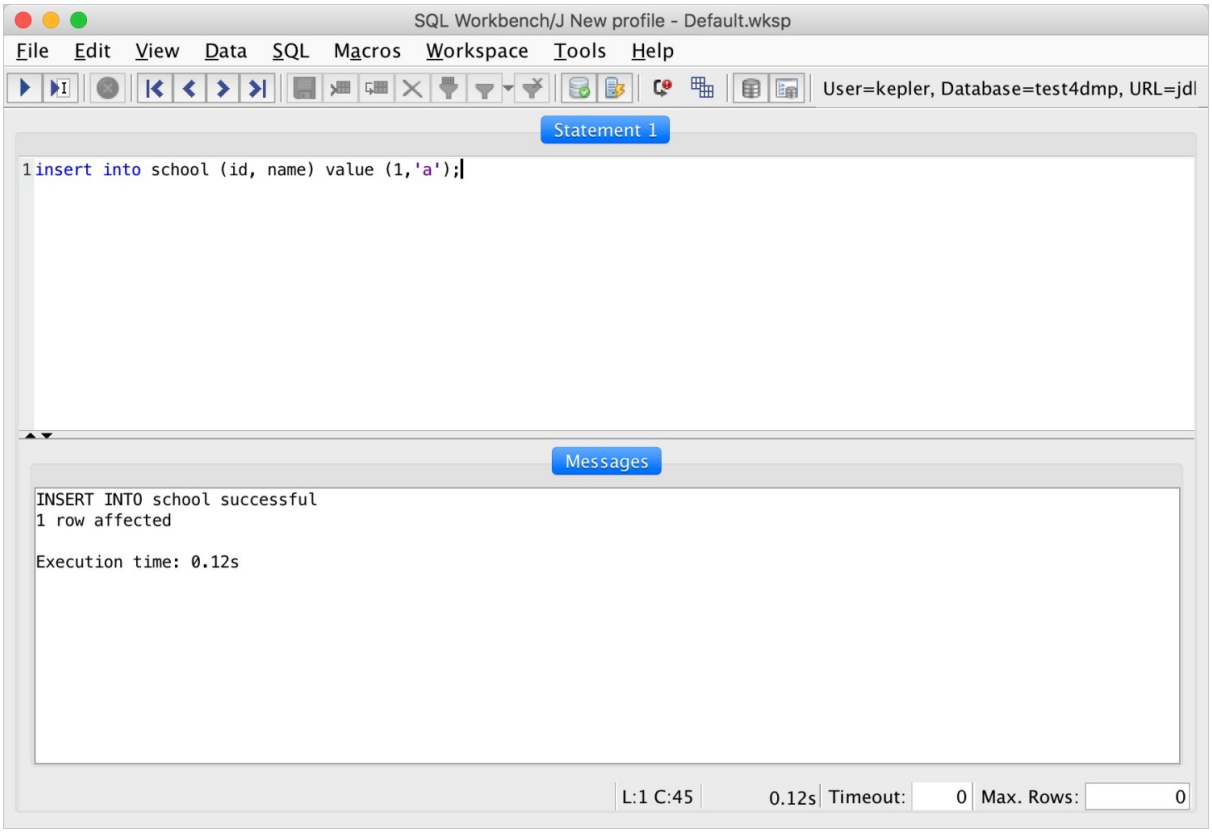
- 列举所有表


SQL Workbench/J New profile - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
User=kepler, Database=test4dmp, URL=jdl
Statement 1
1 show tables;
Result 1 Messages
Tables_in_test4dmp
binary_test
course
dimension_test_date
dimension_test_date_1
dimension_test_id
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timestamp_1
elective
L:1 C:13 0s Timeout: Max. Rows: 0 1-10/32

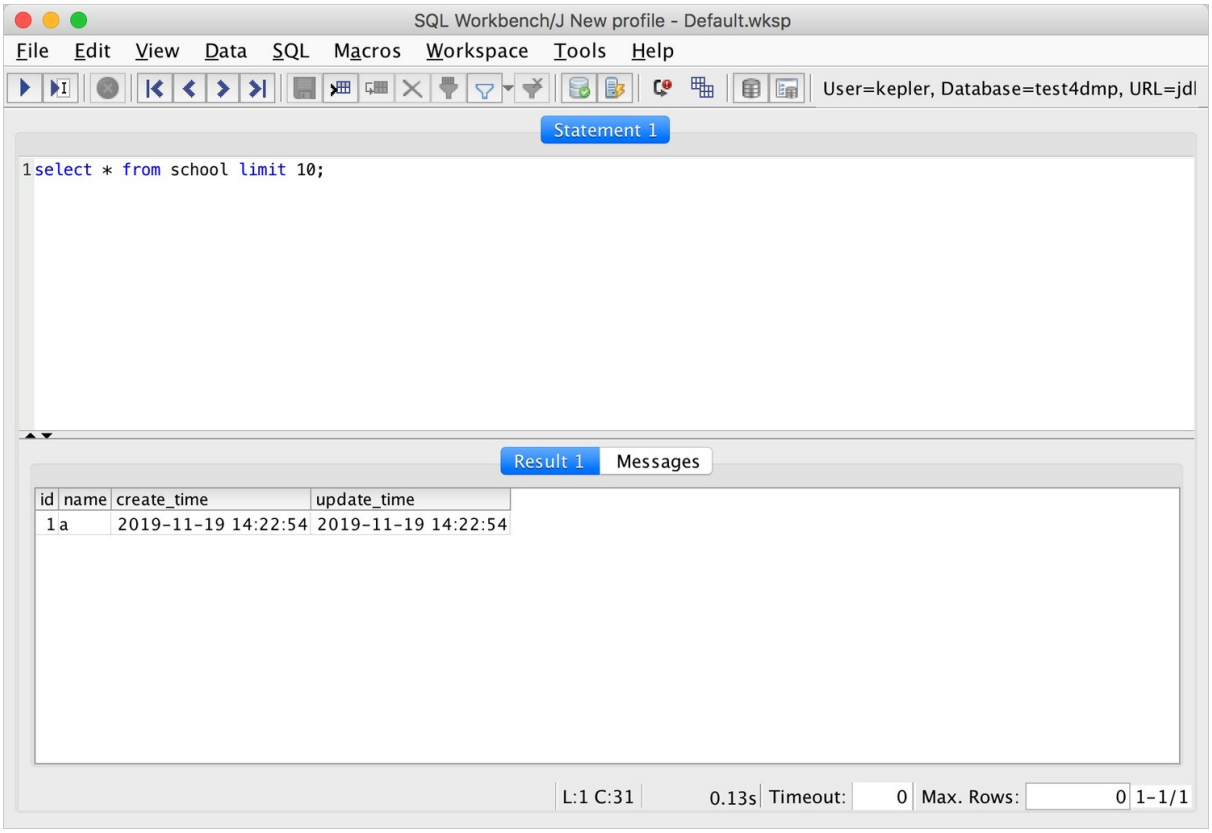
- 查看表结构



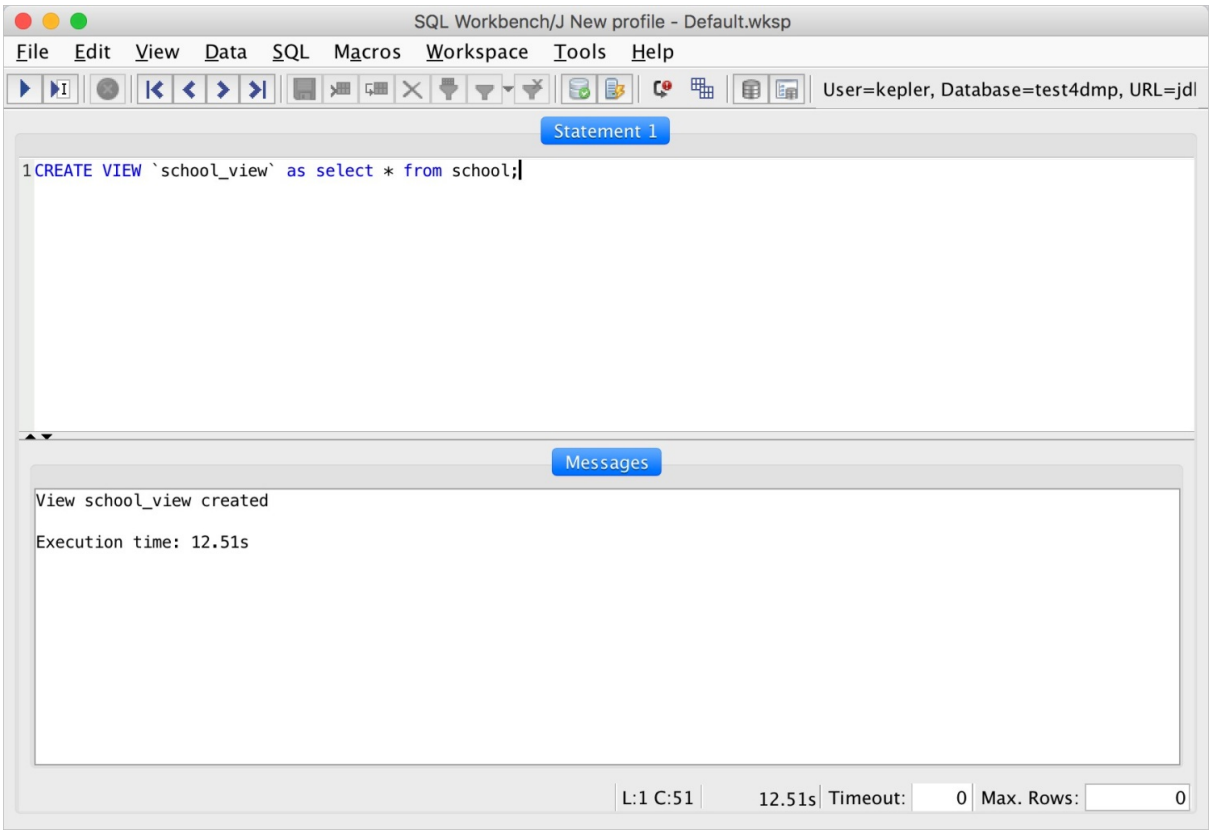
- 向表中写入数据



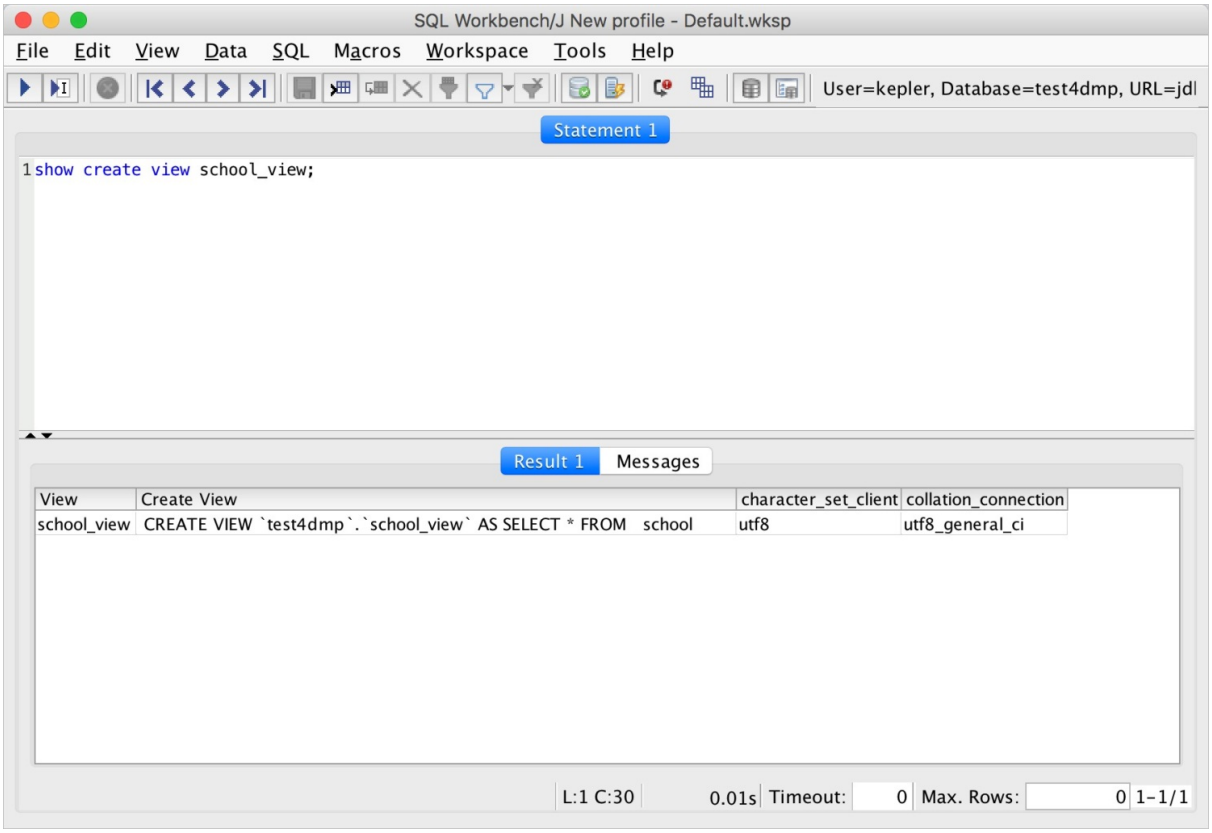
- 查看表数据


SQL Workbench/J New profile - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
User=kepler, Database=test4dmp, URL=jdl
Statement 1
1 select * from school limit 10;
Result 1 Messages
id name create_time update_time
1 a 2019-11-19 14:22:54 2019-11-19 14:22:54
L:1 C:31 0.13s Timeout: 0 Max. Rows: 0 1-1/1

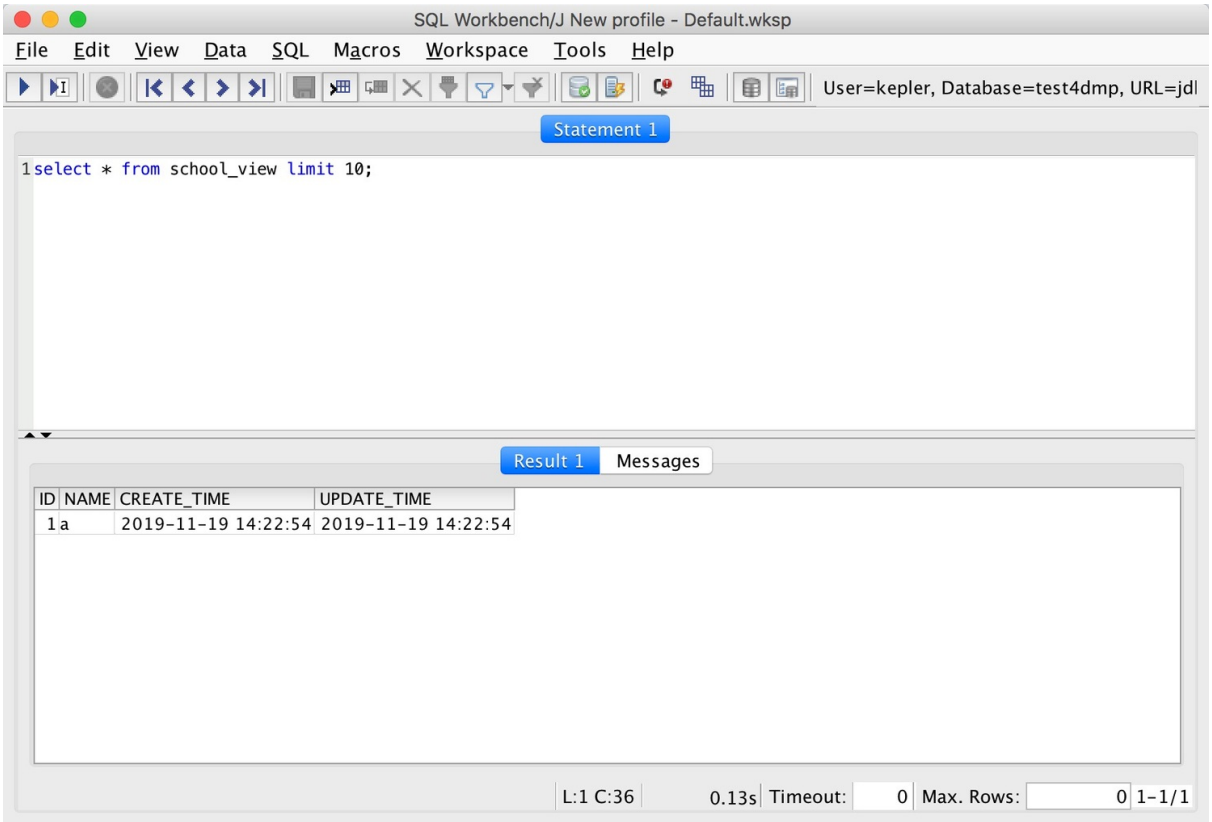
- 创建视图


SQL Workbench/J New profile - Default.wksp
File Edit View Data SQL Macros Workspace Tools Help
User=kepler, Database=test4dmp, URL=jdl
Statement 1
1 CREATE VIEW `school_view` as select * from school;
Messages
View school_view created
Execution time: 12.51s
L:1 C:51 12.51s Timeout: 0 Max. Rows: 0

- 查看视图结构



- 查询视图



2.6. DataGrip

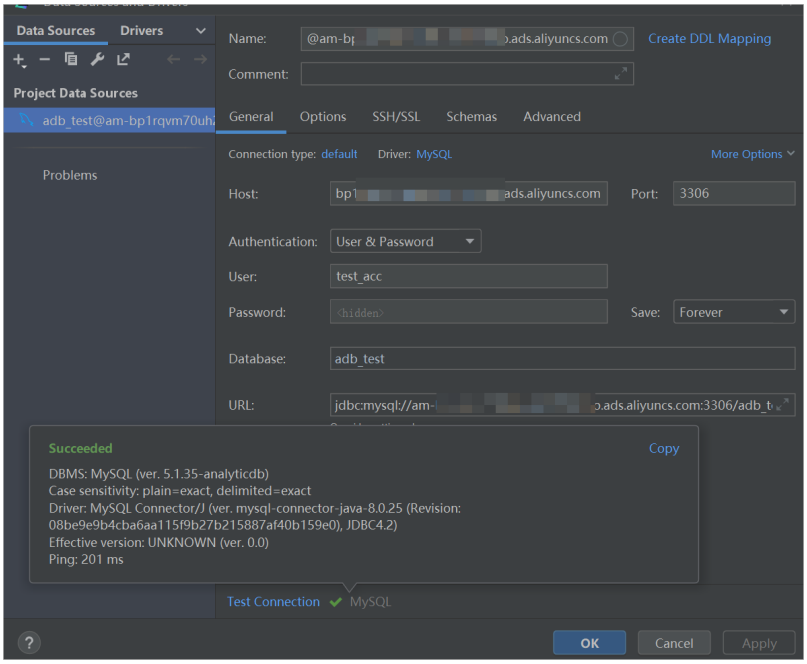
本文测试了DataGrip与云原生数据仓库AnalyticDB MySQL版在连通性、列举数据库、创建表等方面的兼容性，并给出测试结果图。

测试环境

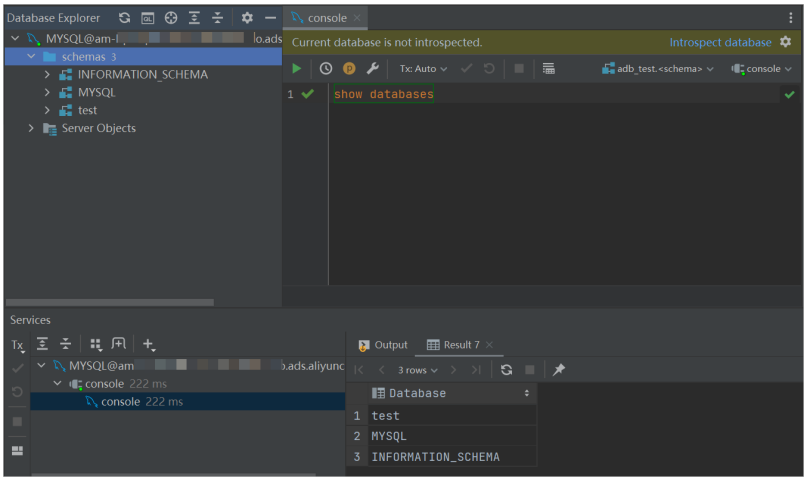
MySQL JDBC Driver	DataGrip
mysql-connector-java-8.0.15	下载地址为DataGrip。

测试范围

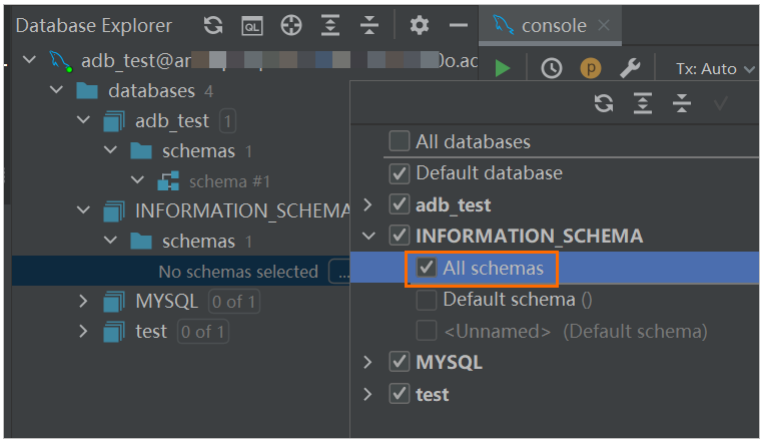
- 连通性



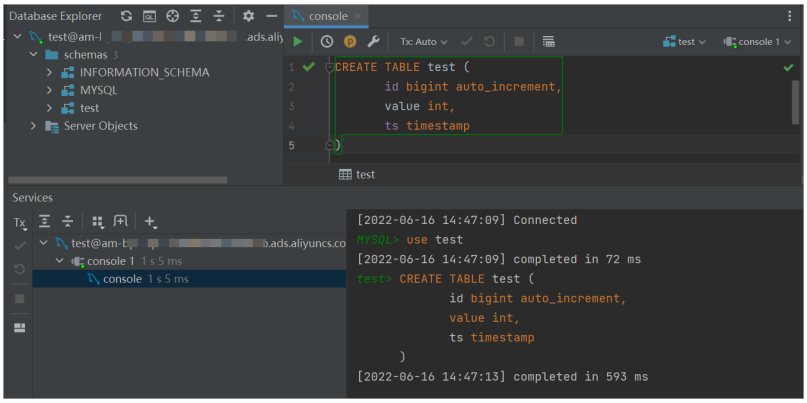
- 列举数据库



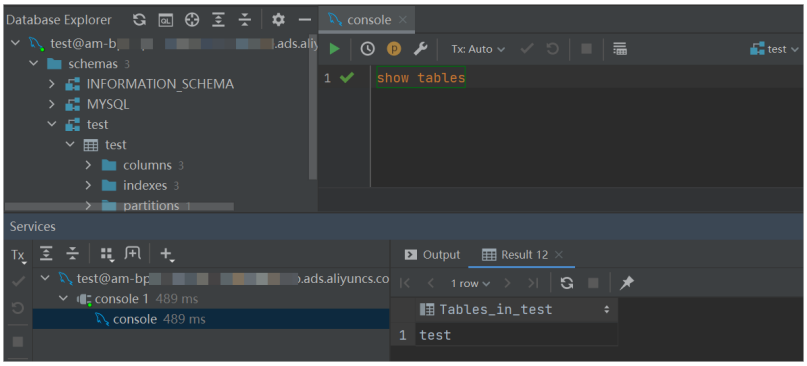
如果需要展示所有的Schemas，请在连接配置中选中All schemas。



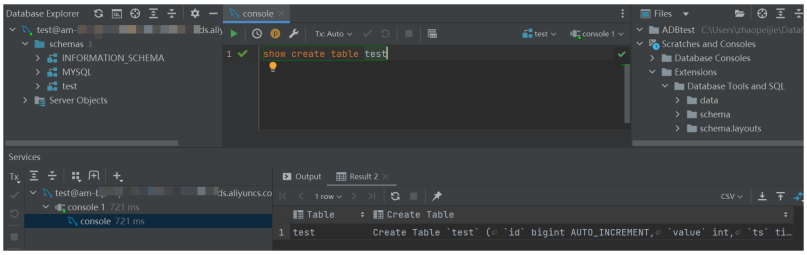
- 创建表



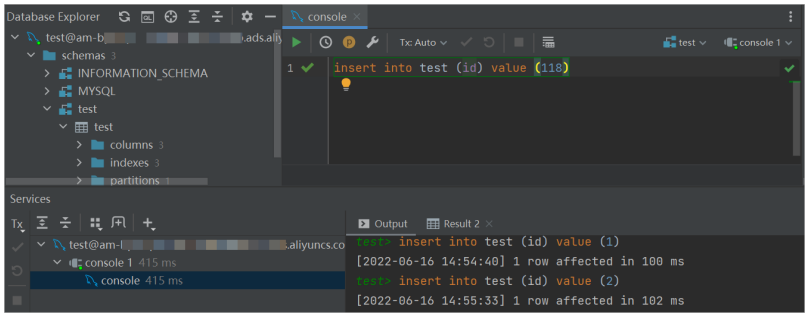
- 列举所有表



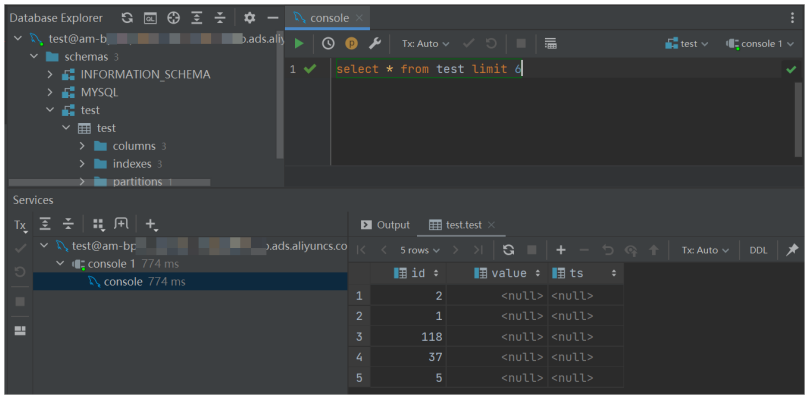
- 查看表结构



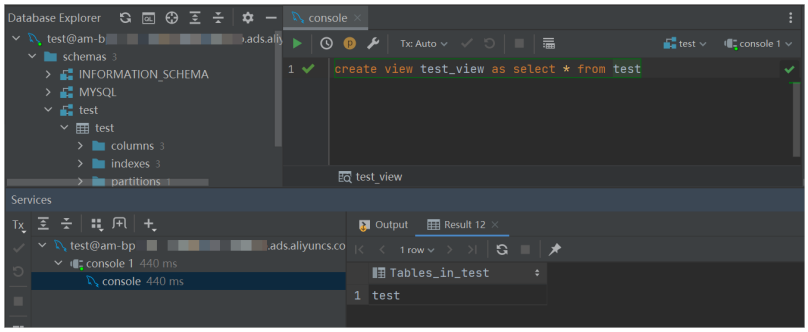
- 向表中写入数据



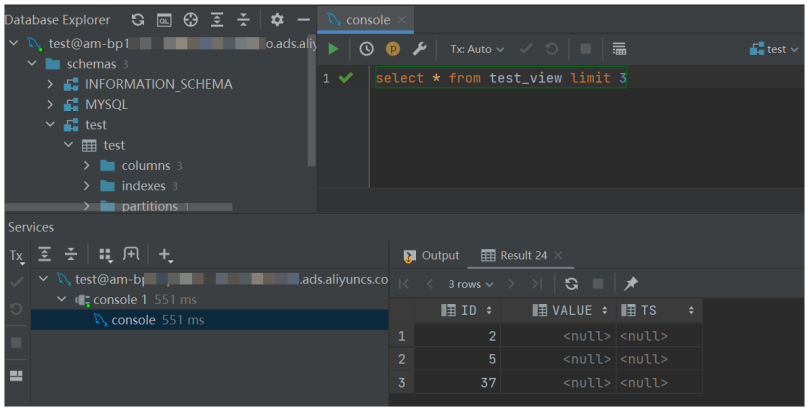
- 查看表数据



- 创建视图



- 查询视图



2.7. Oracle Golden Gate

Oracle Golden Gate（OGG）可用于将Oracle中的数据同步到AnalyticDB for MySQL中。

Oracle Golden Gate数据类型与AnalyticDB for MySQL兼容性

源端	目标端		同步行为		
Oracle	MySQL	AnalyticDB for MySQL	INSERT	UPDATA	DELETE
number(3)	bool或tinyint(1)	boolean	支持	支持	支持

源端	目标端		同步行为		
Oracle	MySQL	AnalyticDB for MySQL	INSERT	UPDATA	DELETE
number(3)	tinyint	tingyint	支持	支持	支持
number(5)	smallint	smallint	支持	支持	支持
number(10)	int	int/integer	支持	支持	支持
number(19)	bigint	bigint	支持	支持	支持
float(24)	float	float	支持	支持	支持
float(24)	double	double	支持	支持	支持
float(24)	decimal	decimal	支持	支持	支持
varchar2(128)	char	varchar(128)	支持	支持	支持
varchar2(2000)	varchar(255)	varchar(255)	支持	支持	支持
varchar2(4000)	text	varchar(65535)	支持	支持	支持
date	date	date	支持	支持	支持
date	time	time	不支持该功能、MySQL不适用	不支持该功能、MySQL不适用	不支持该功能、MySQL不适用
date	datetime	datetime	支持	支持	支持
date	timestamp	timestamp	支持	支持	支持

通过OGG将Oracle数据同步到AnalyticDB for MySQL

1. 使用OGG账号登录ECS。

```
sqlplus ogg/ogg
```

2. 执行以下SQL在Oracle中创建源表。

```
drop table users.xqtest15;
create table users.xqtest15 (
  c1 number(10),
  c2 number(1),
  c3 number(3),
  c4 number(5),
  c5 number(19),
  c6 float(24),
  c7 float(24),
  c8 float(24),
  c9 char(1),
  c10 varchar2(2000),
  c11 varchar2(4000),
  c12 date,
  c13 date,
  c14 date,
  c15 date,
  primary key(c1)
);
```

3. Oracle建表成功后，在OGG源端添加trandata。

```
[OGG账号登录ECS] cd /odata/ogg_o_12202
[启动OGG] ./ggsci
ggsci> dblogin userid goldengate, password ogg
ggsci> add trandata users.xqtest15
```

> **说明** 上述命令末尾不要加分号（;），否则将报错No viable tables matched specification。

4. 执行以下SQL在MySQL中创建表，存储Oracle中的元数据。

```
-- checkpoint表
CREATE TABLE `ckpt1220` (
  `group_name` varchar(8) NOT NULL,
  `group_key` decimal(19,0) NOT NULL,
  `seqno` decimal(10,0) DEFAULT NULL,
  `rba` decimal(19,0) NOT NULL,
  `audit_ts` varchar(29) DEFAULT NULL,
  `create_ts` datetime NOT NULL,
  `last_update_ts` datetime NOT NULL,
  `current_dir` varchar(255) NOT NULL,
  `log_bsn` varchar(128) DEFAULT NULL,
  `log_csn` varchar(128) DEFAULT NULL,
  `log_xid` varchar(128) DEFAULT NULL,
  `log_cmplt_csn` varchar(128) DEFAULT NULL,
  `log_cmplt_xids` varchar(2000) DEFAULT NULL,
  `version` decimal(3,0) DEFAULT NULL,
  PRIMARY KEY (`group_name`,`group_key`)
) DISTRIBUTE BY HASH(`group_key`) INDEX_ALL='Y';
```

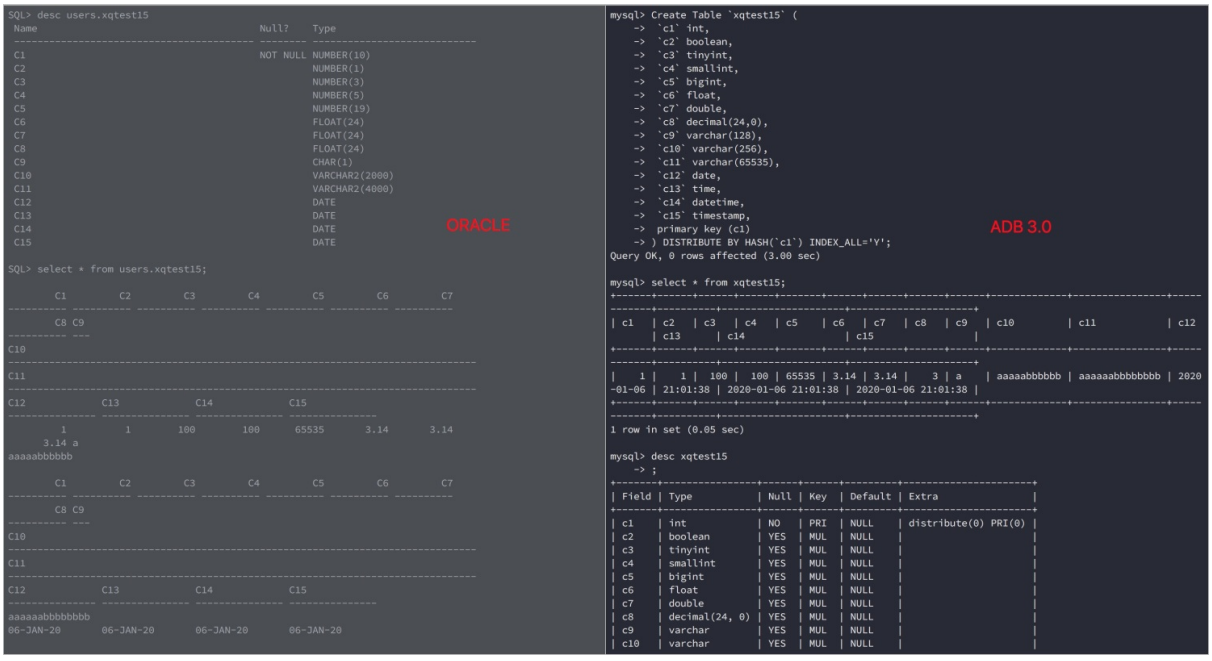
```
-- checkpoint表
CREATE TABLE `ckpt1220_lox` (
  `group_name` varchar(8) NOT NULL,
  `group_key` decimal(19,0) NOT NULL,
  `log_cmplt_csn` varchar(128) NOT NULL,
  `log_cmplt_xids_seq` decimal(5,0) NOT NULL,
  `log_cmplt_xids` varchar(2000) NOT NULL,
  PRIMARY KEY (`group_name`,`group_key`,`log_cmplt_csn`,`log_cmplt_xids_seq`)
) DISTRIBUTE BY HASH(`group_key`) INDEX_ALL='Y';
```

5. 执行以下SQL在AnalyticDB for MySQL中创建目标表，存储从Oracle同步过来的数据。

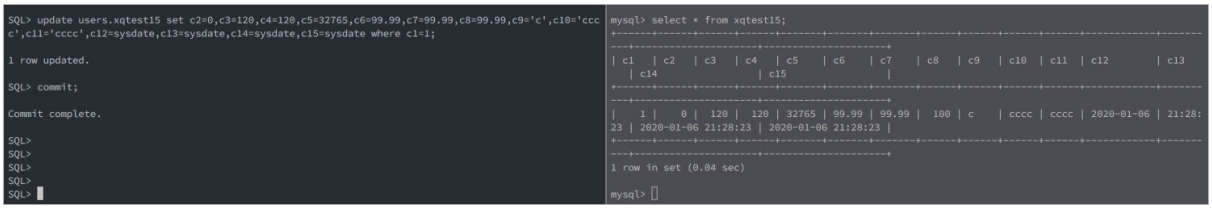
```
Create Table `xqtest15` (
 `c1` int,
 `c2` boolean,
 `c3` tinyint,
 `c4` smallint,
 `c5` bigint,
 `c6` float,
 `c7` double,
 `c8` decimal(24, 0),
 `c9` varchar(128),
 `c10` varchar(255),
 `c11` varchar(65535),
 `c12` date,
 `c13` time,
 `c14` datetime,
 `c15` timestamp,
 primary key (c1)
) DISTRIBUTE BY HASH(`c1`) INDEX_ALL='Y'
```

附录

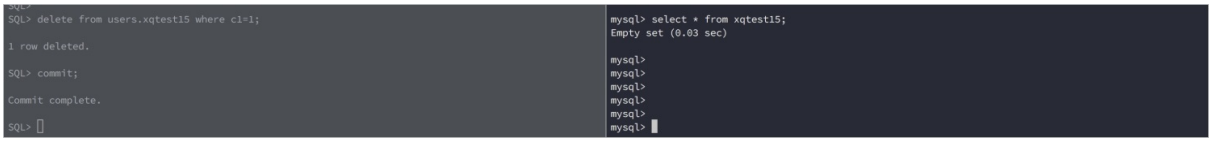
- INSERT同步



- UPDATE同步



- DELETE同步



2.8. Tableau

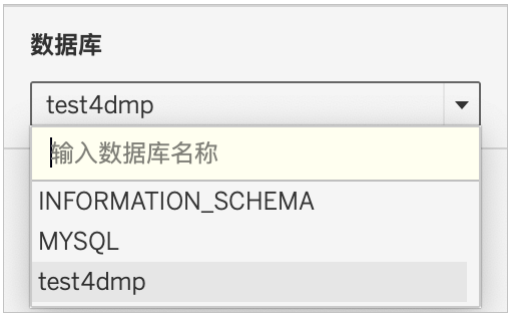
本文测试了Tableau与AnalyticDB for MySQL在列举数据库、查看表表等方面的兼容性，并给出测试结果图。

测试环境

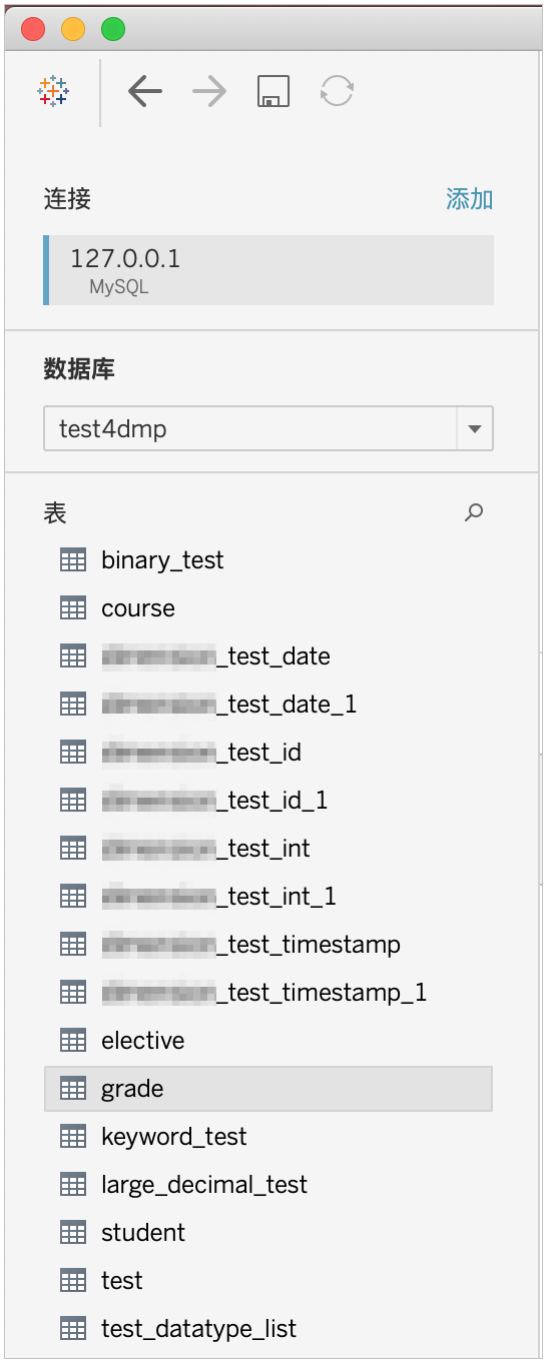
MySQL	Tableau
mysql Ver5.6.46 for osx10.13 on x86_64 (Homebrew)	Tableau Desktop 2019.4

测试范围

- 列举数据库



- 列举所有表


连接
添加
127.0.0.1
MySQL
数据库
test4dmp
表
binary_test
course
_test_date
_test_date_1
_test_id
_test_id_1
_test_int
_test_int_1
_test_timestamp
_test_timestamp_1
elective
grade
keyword_test
large_decimal_test
student
test
test_datatype_list

- 查看表数据

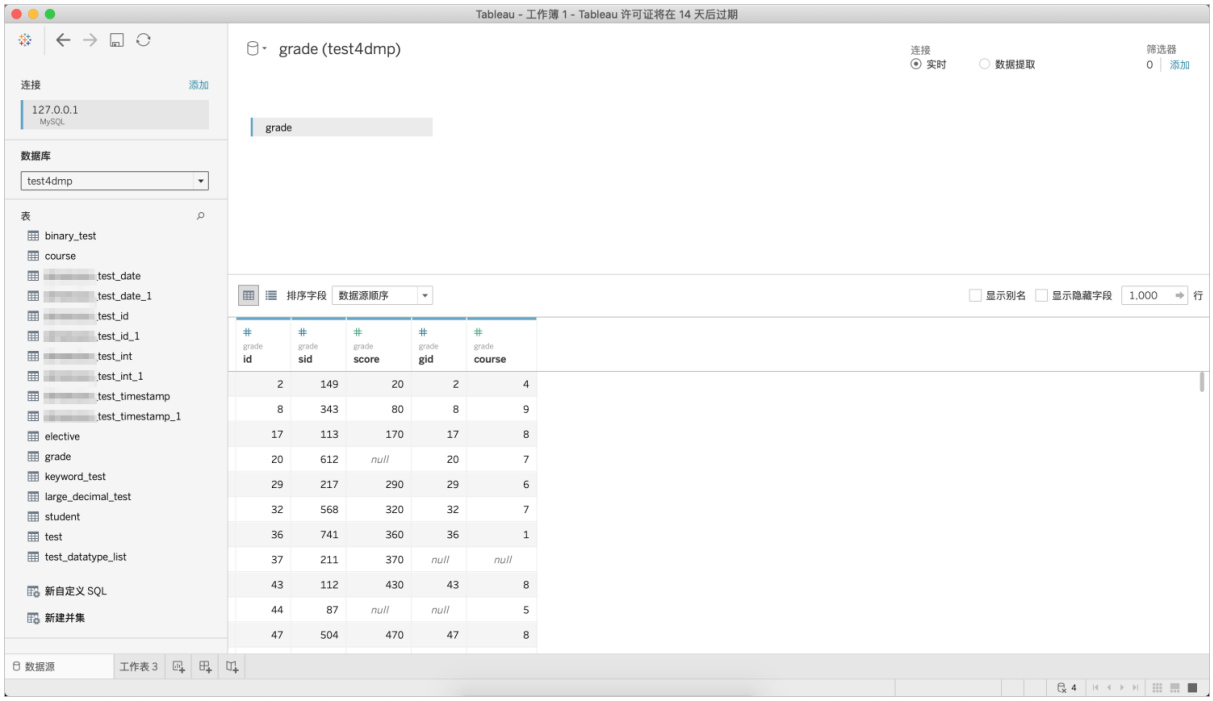


Tableau - 工作簿 1 - Tableau 许可证将在 14 天后过期
grade (test4dmp)
连接
实时
数据提取
筛选器
0 添加
连接
添加
127.0.0.1
MySQL
数据库
test4dmp
表
binary_test
course
test_date
test_date_1
test_id
test_id_1
test_int
test_int_1
test_timestamp
test_timestamp_1
elective
grade
keyword_test
large_decimal_test
student
test
test_datatype_list
新自定义 SQL
新建并集
grade
排序字段 数据源顺序
显示别名
显示隐藏字段
1,000
行
id
sid
score
gid
course
2 149 20 2 4
8 343 80 8 9
17 113 170 17 8
20 612 null 20 7
29 217 290 29 6
32 568 320 32 7
36 741 360 36 1
37 211 370 null null
43 112 430 43 8
44 87 null null 5
47 504 470 47 8
数据源
工作表 3

- 突出显示表

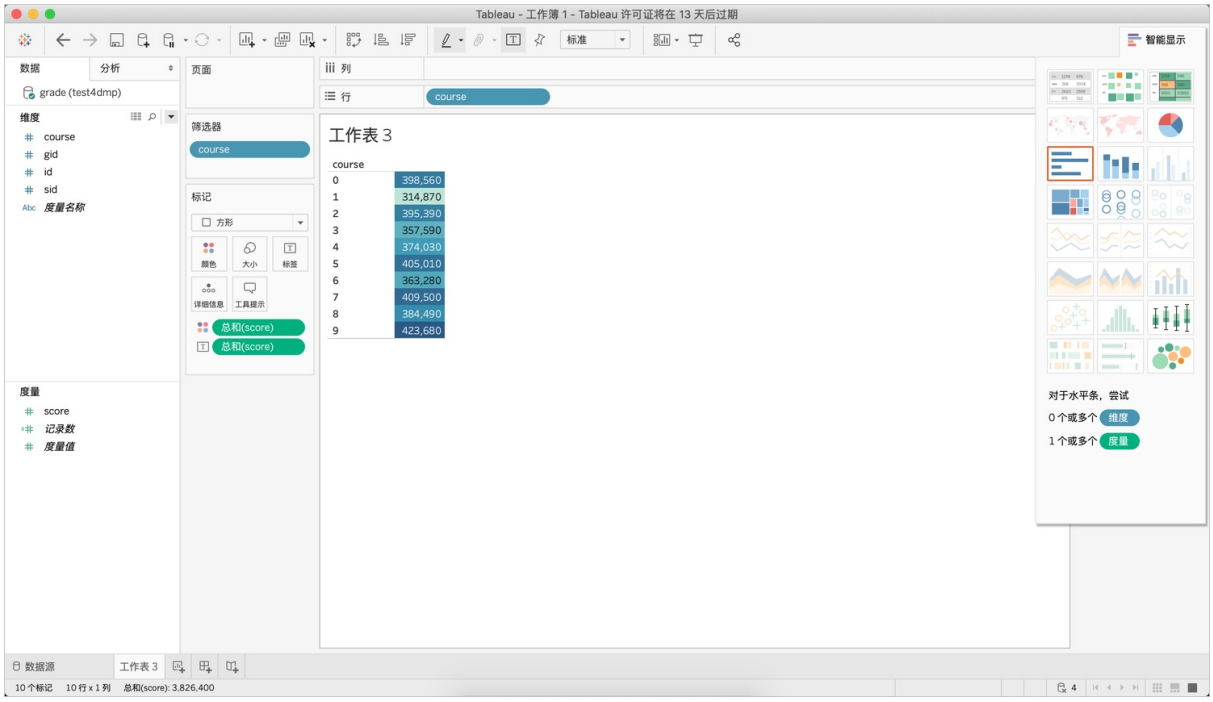
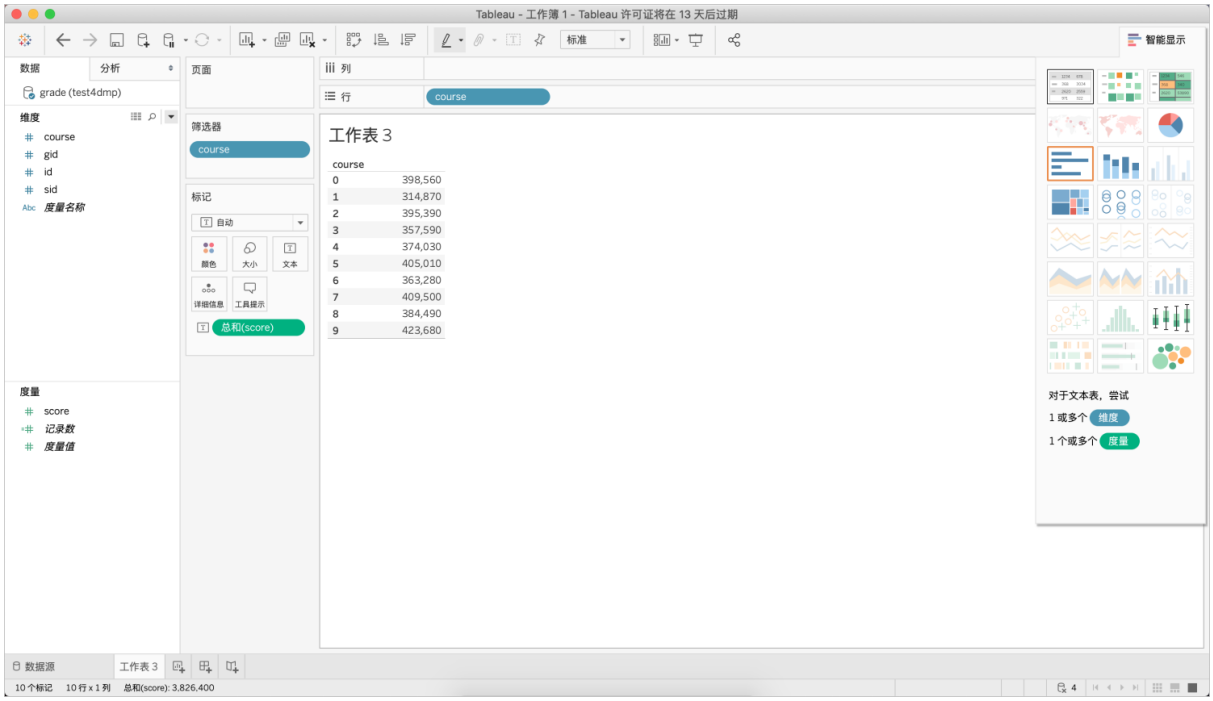
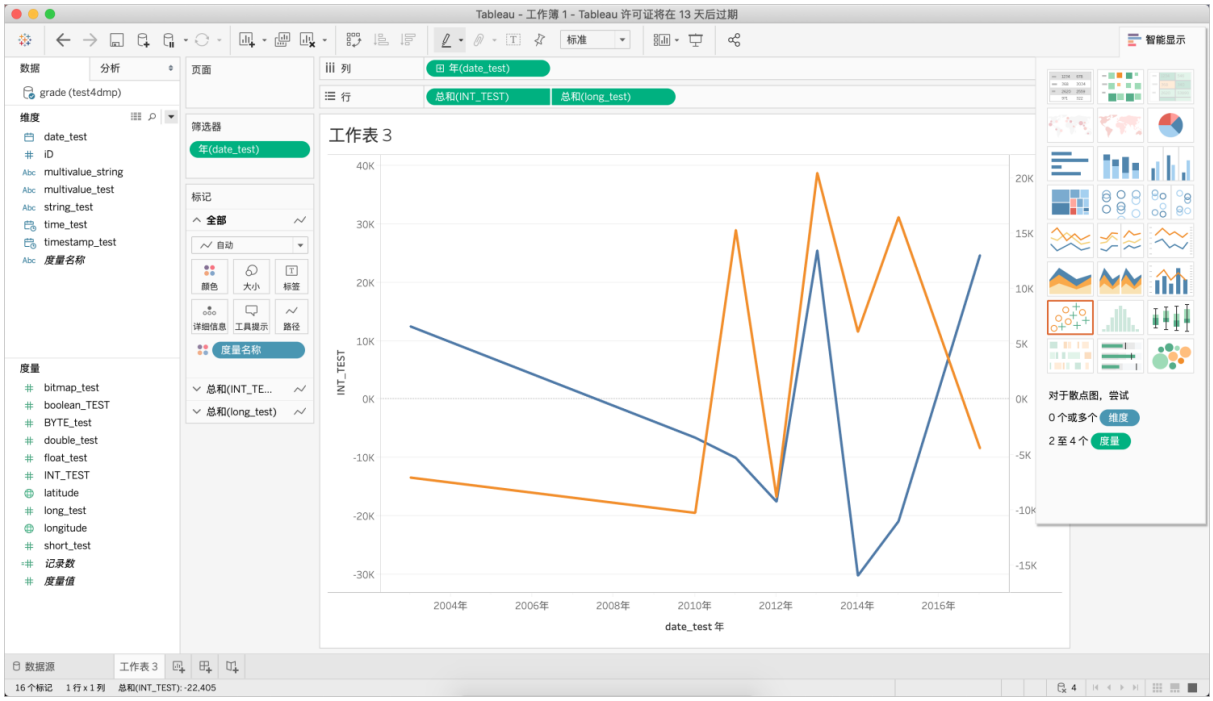


Tableau - 工作簿 1 - Tableau 许可证将在 13 天后过期
智能显示
数据
分析
grade (test4dmp)
维度
course
gid
id
sid
度量名称
页面
筛选器
course
标记
方形
颜色
大小
标签
详细信息
工具提示
总和(score)
总和(score)
列
行
course
工作表 3
course
0 398,560
1 314,870
2 395,390
3 357,590
4 374,030
5 405,010
6 363,280
7 409,500
8 384,490
9 423,680
度量
score
记录数
度量值
对于水平条，尝试
0 个或多个 维度
1 个或多个 度量
数据源
工作表 3
10 个标记 10 行 x 1 列 总和(score): 3,826,400

- 文本表



- 折线图



2.9. QlikView

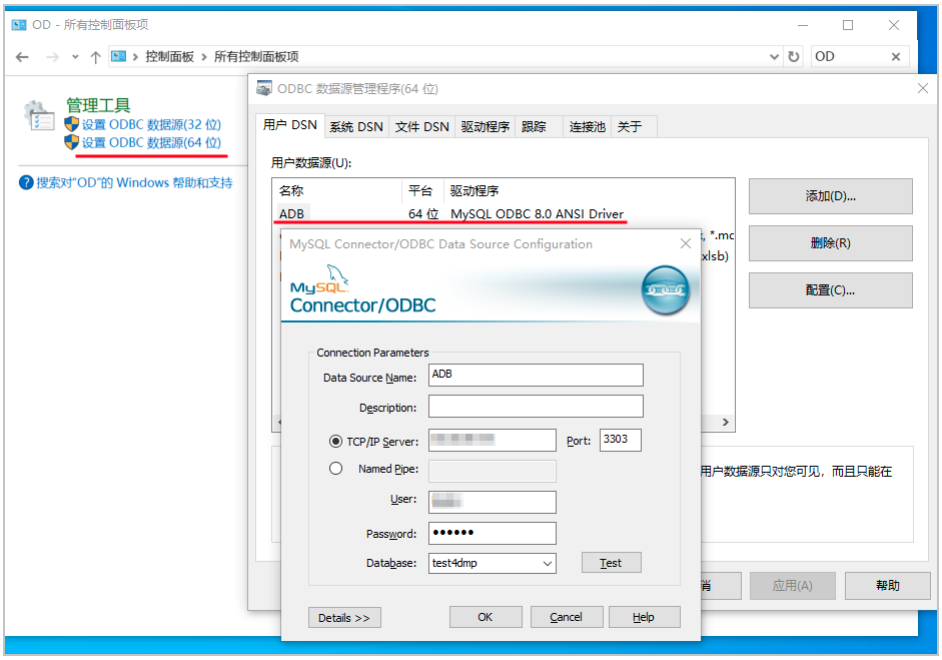
本文测试了QlikView与AnalyticDB for MySQL在连通性和显示表等方面的兼容性，并给出测试结果图。

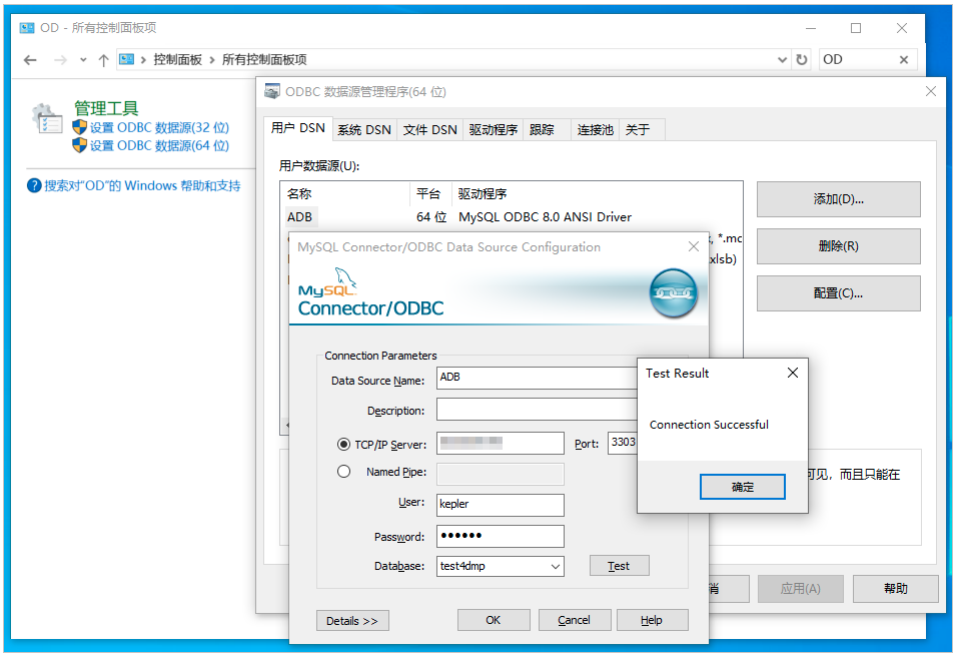
测试环境

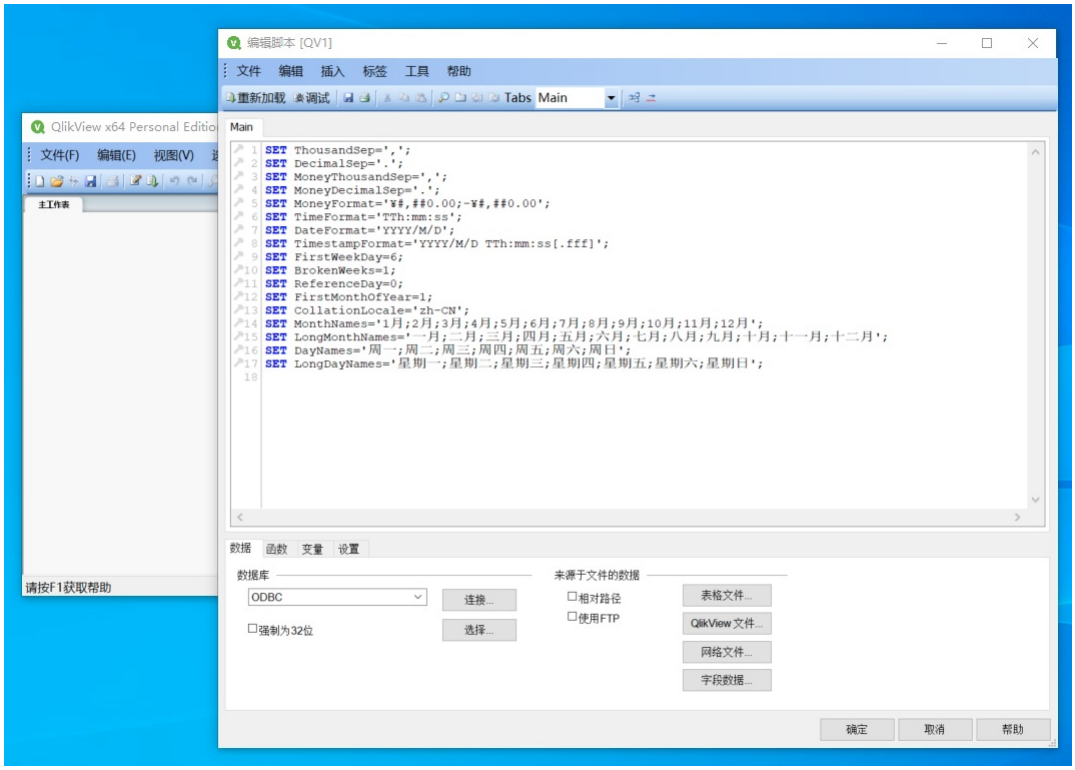
MySQL ODBC Driver	QlikView
MySQL ODBC 8.0 ANSI Driver x64	QlikView 12 on Windows (Personal Edition)

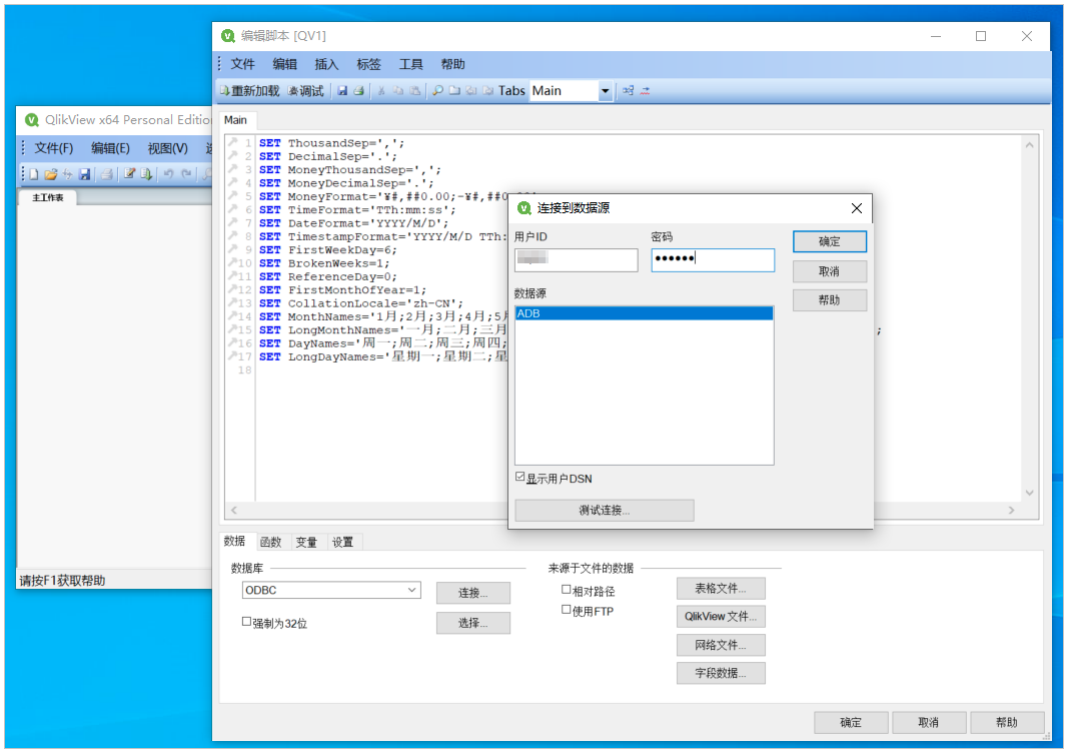
测试范围

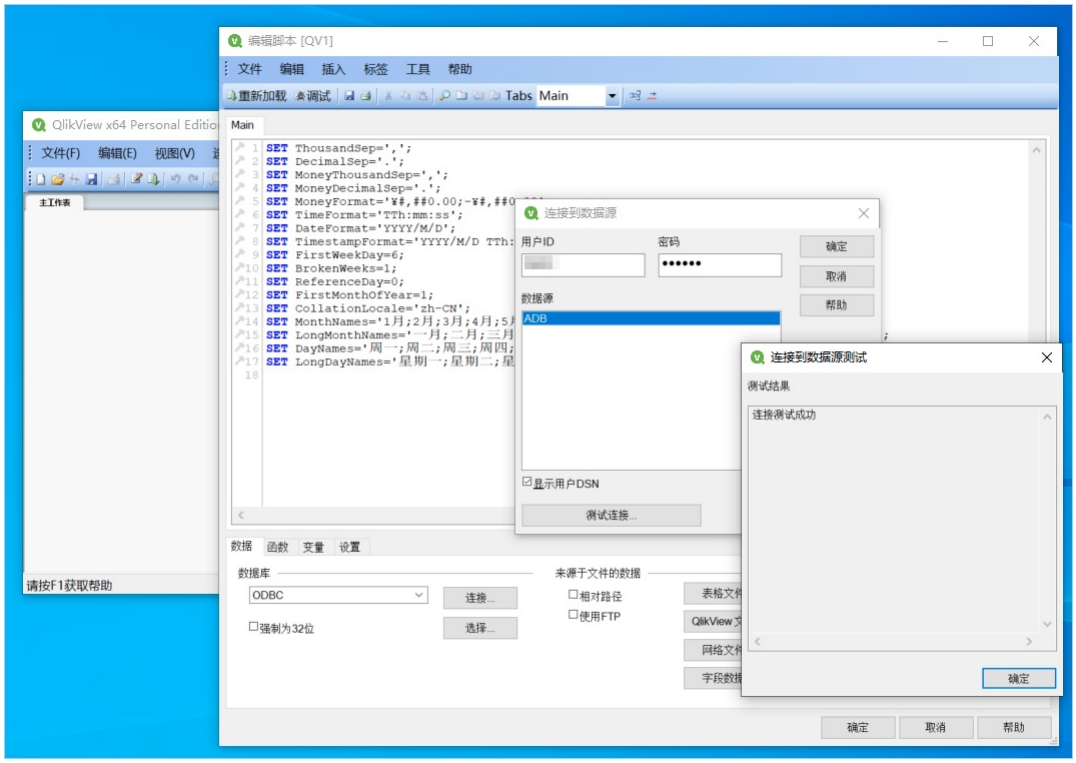
- 连通性


OD - 所有控制面板项
控制面板 › 所有控制面板项
管理工具
设置 ODBC 数据源(32 位)
设置 ODBC 数据源(64 位)
搜索对"OD"的 Windows 帮助和支持
ODBC 数据源管理程序(64 位)
用户 DSN
系统 DSN
文件 DSN
驱动程序
跟踪
连接池
关于
用户数据源(U):
名称
平台
驱动程序
ADB
64 位
MySQL ODBC 8.0 ANSI Driver
添加(D)...
删除(R)
配置(C)...
MySQL Connector/ODBC Data Source Configuration
MySQL
Connector/ODBC
Connection Parameters
Data Source Name:
ADB
Description:
TCP/IP Server:
Port:
3303
Named Pipe:
User:
Password:
Database:
test4dmp
Test
Details >>
OK
Cancel
Help
应用(A)
帮助

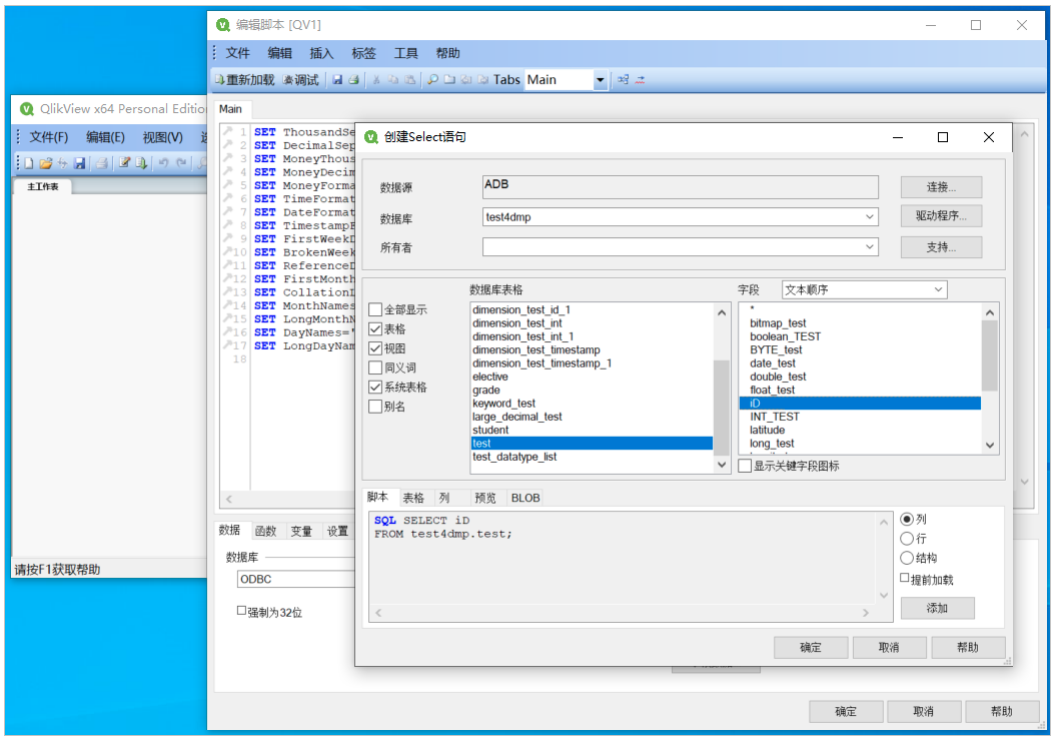

OD - 所有控制面板项
控制面板 › 所有控制面板项
管理工具
设置 ODBC 数据源(32 位)
设置 ODBC 数据源(64 位)
搜索对"OD"的 Windows 帮助和支持
ODBC 数据源管理程序(64 位)
用户 DSN
系统 DSN
文件 DSN
驱动程序
跟踪
连接池
关于
用户数据源(U):
名称
平台
驱动程序
ADB
64 位
MySQL ODBC 8.0 ANSI Driver
添加(D)...
删除(R)
配置(C)...
MySQL Connector/ODBC Data Source Configuration
MySQL
Connector/ODBC
Connection Parameters
Data Source Name:
ADB
Description:
TCP/IP Server:
Port:
3303
Named Pipe:
User:
kepler
Password:
Database:
test4dmp
Test
Test Result
Connection Successful
确定
Details >>
OK
Cancel
Help
应用(A)
帮助


编辑脚本 [QV1]
文件 编辑 插入 标签 工具 帮助
重新加载 调试 Tabs Main
Main
SET ThousandSep=',';
SET DecimalSep='.';
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='¥#,##0.00;-¥#,##0.00';
SET TimeFormat='TTh:mm:ss';
SET DateFormat='YYYY/M/D';
SET TimestampFormat='YYYY/M/D TTh:mm:ss[.fff]';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;
SET FirstMonthOfYear=1;
SET CollationLocale='zh-CN';
SET MonthNames='1月;2月;3月;4月;5月;6月;7月;8月;9月;10月;11月;12月';
SET LongMonthNames='一月;二月;三月;四月;五月;六月;七月;八月;九月;十月;十一月;十二月';
SET DayNames='周一;周二;周三;周四;周五;周六;周日';
SET LongDayNames='星期一;星期二;星期三;星期四;星期五;星期六;星期日';
数据 函数 变量 设置
数据库
ODBC
连接...
选择...
强制为32位
来源于文件的数据
相对路径
使用FTP
表格文件...
QlikView 文件...
网络文件...
字段数据...
确定
取消
帮助
QlikView x64 Personal Editio
文件(F) 编辑(E) 视图(V)
主工作表
请按F1获取帮助

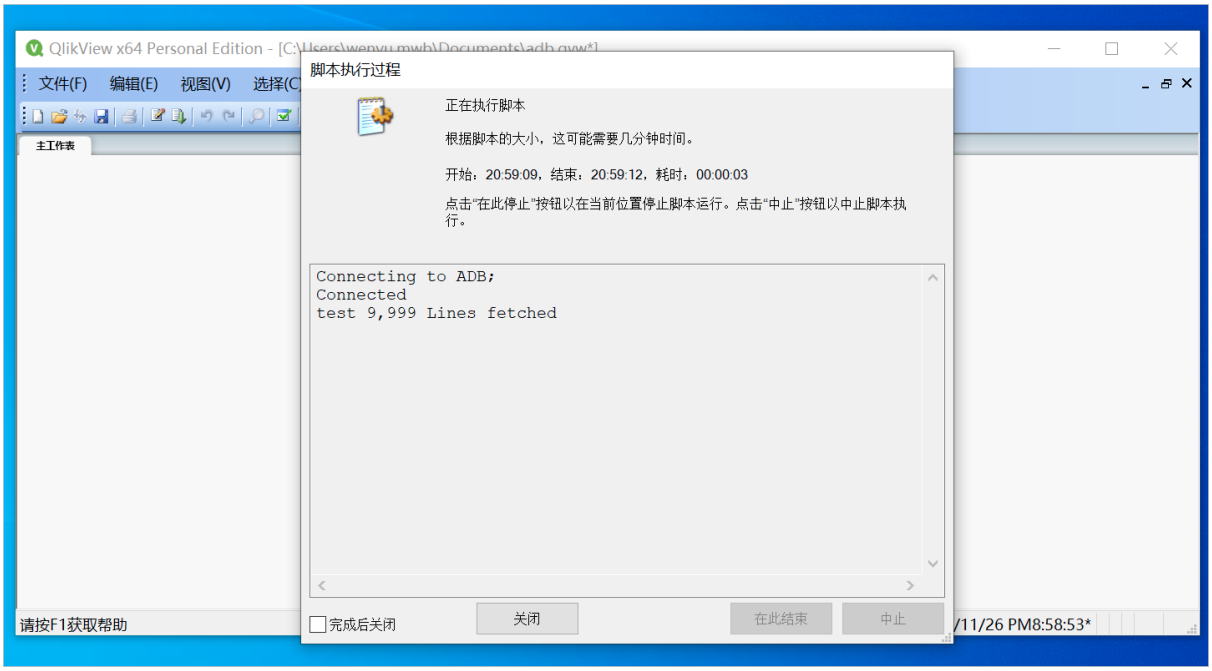

编辑脚本 [QV1]
文件 编辑 插入 标签 工具 帮助
重新加载 调试 Tabs Main
Main
连接到数据源
用户ID
密码
确定
取消
帮助
数据源
ADB
显示用户DSN
测试连接...
数据 函数 变量 设置
数据库
ODBC
连接...
选择...
强制为32位
来源于文件的数据
相对路径
使用FTP
表格文件...
QlikView 文件...
网络文件...
字段数据...
确定
取消
帮助
QlikView x64 Personal Editio
文件(F) 编辑(E) 视图(V)
主工作表
请按F1获取帮助

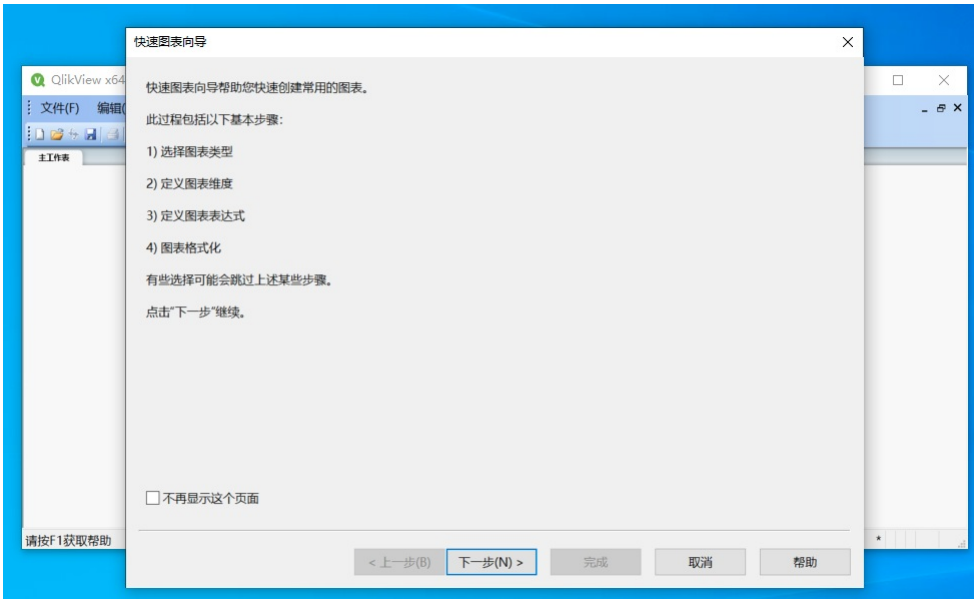

编辑脚本 [QV1]
文件 编辑 插入 标签 工具 帮助
重新加载 调试 Tabs Main
Main
连接到数据源
用户ID
密码
确定
取消
帮助
数据源
ADB
显示用户DSN
测试连接...
连接到数据源测试
测试结果
连接测试成功
确定
数据 函数 变量 设置
数据库
ODBC
连接...
选择...
强制为32位
来源于文件的数据
相对路径
使用FTP
确定 取消 帮助

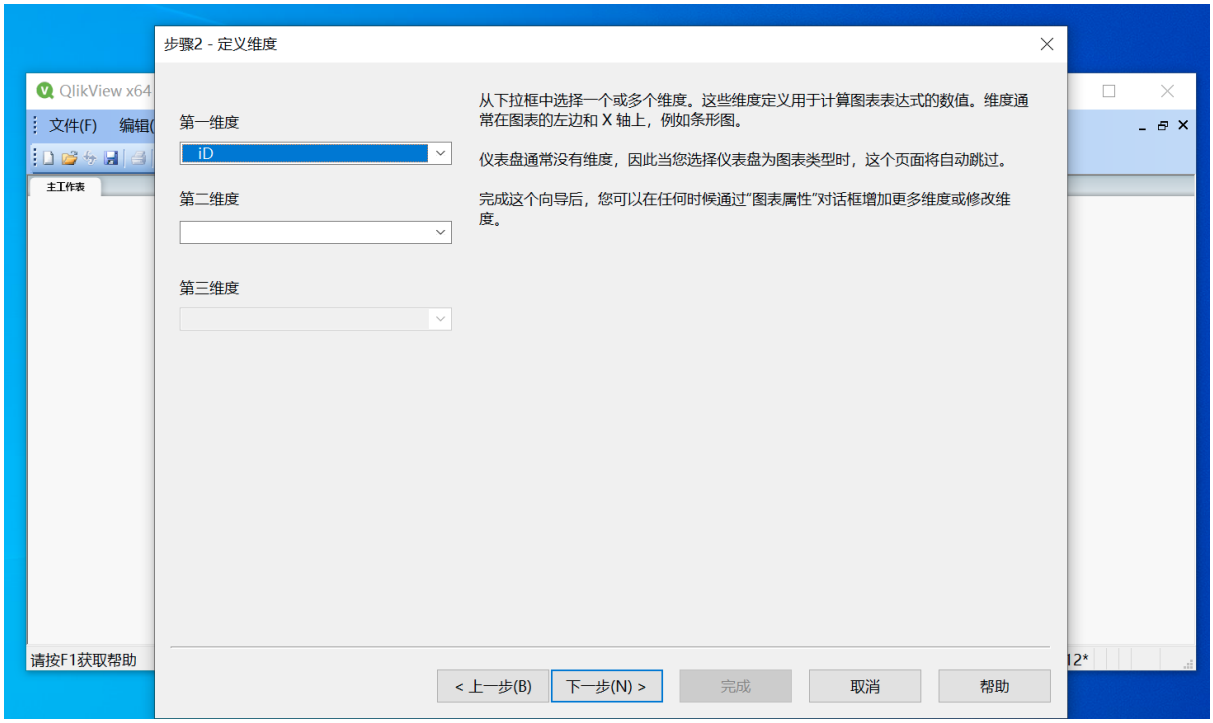
- 显示表

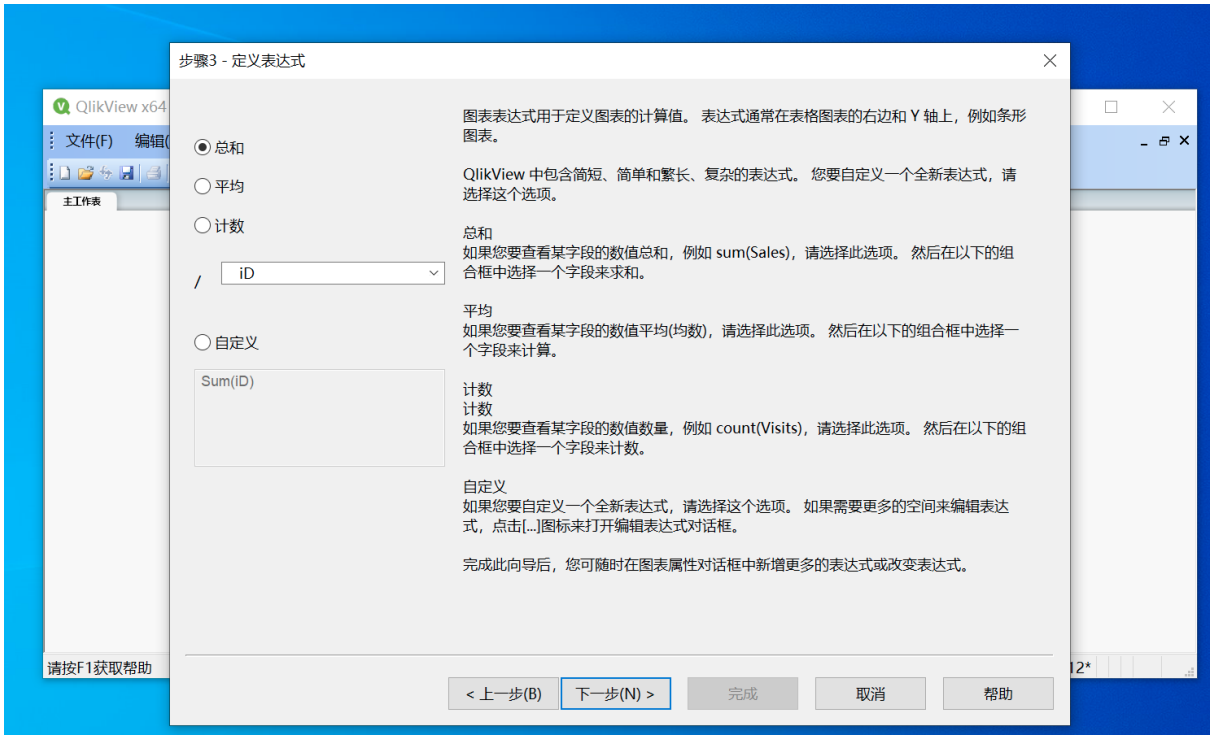

编辑脚本 [QV1]
文件 编辑 插入 标签 工具 帮助
重新加载 调试 Tabs Main
Main
创建Select语句
数据源
ADB
连接...
数据库
test4dmp
驱动程序...
所有者
支持...
数据库表格
字段
文本顺序
全部显示
表格
视图
同义词
系统表格
别名
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timestamp_1
elective
grade
keyword_test
large_decimal_test
student
test
test_datatype_list
bitmap_test
boolean_TEST
BYTE_test
date_test
double_test
float_test
iD
INT_TEST
latitude
long_test
显示关键字段图标
脚本 表格 列 预览 BLOB
SQL SELECT iD
FROM test4dmp.test;
列
行
结构
提前加载
添加
确定 取消 帮助
数据 函数 变量 设置
数据库
ODBC
强制为32位
确定 取消 帮助

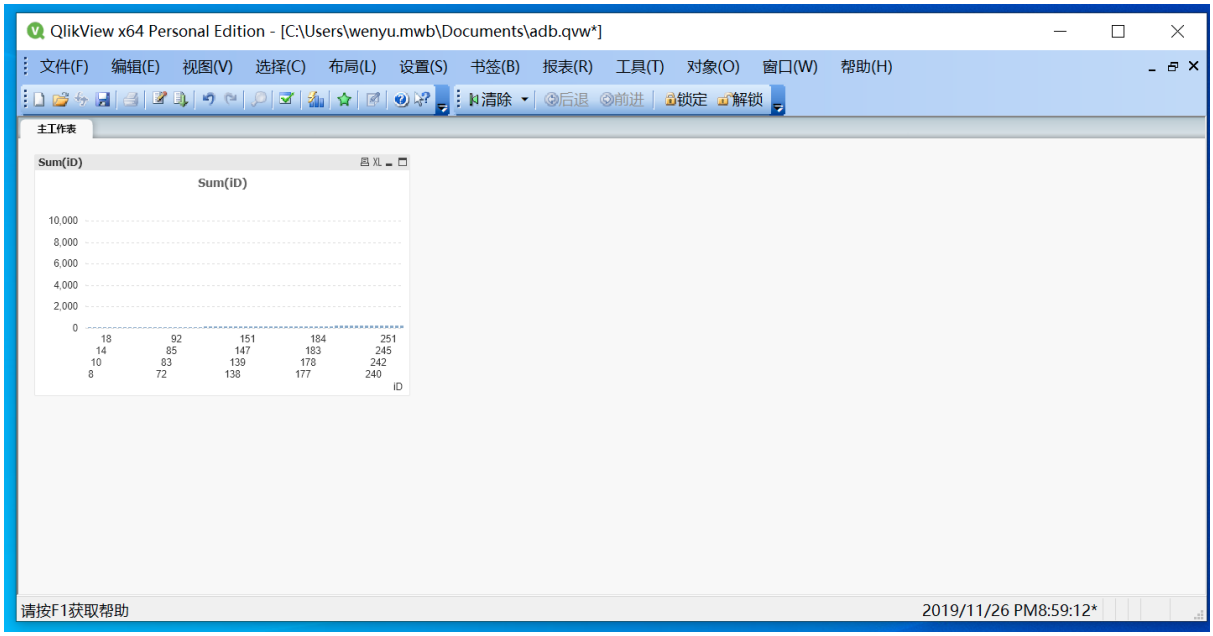
- 查询


QlikView x64 Personal Edition
文件(F)
编辑(E)
视图(V)
选择(C)
主工作表
脚本执行过程
正在执行脚本
根据脚本的大小，这可能需要几分钟时间。
开始：20:59:09，结束：20:59:12，耗时：00:00:03
点击"在此停止"按钮以在当前位置停止脚本运行。点击"中止"按钮以中止脚本执行。
Connecting to ADB;
Connected
test 9,999 Lines fetched
完成后关闭
关闭
在此结束
中止
请按F1获取帮助


快速图表向导
快速图表向导帮助您快速创建常用的图表。
此过程包括以下基本步骤:
1) 选择图表类型
2) 定义图表维度
3) 定义图表表达式
4) 图表格式化
有些选择可能会跳过上述某些步骤。
点击"下一步"继续。
不再显示这个页面
< 上一步(B)
下一步(N) >
完成
取消
帮助


步骤2 - 定义维度
QlikView x64
文件(F) 编辑(
主工作表
第一维度
iD
第二维度
第三维度
从下拉框中选择一个或多个维度。这些维度定义用于计算图表表达式的数值。维度通常在图表的左边和 X 轴上，例如条形图。
仪表盘通常没有维度，因此当您选择仪表盘为图表类型时，这个页面将自动跳过。
完成这个向导后，您可以在任何时候通过"图表属性"对话框增加更多维度或修改维度。
< 上一步(B)
下一步(N) >
完成
取消
帮助
请按F1获取帮助


步骤3 - 定义表达式
QlikView x64
文件(F) 编辑(
主工作表
总和
平均
计数
/
iD
自定义
Sum(iD)
图表表达式用于定义图表的计算值。 表达式通常在表格图表的右边和 Y 轴上，例如条形图表。
QlikView 中包含简短、简单和繁长、复杂的表达式。 您要自定义一个全新表达式，请选择这个选项。
总和
如果您要查看某字段的数值总和，例如 sum(Sales)，请选择此选项。 然后在以下的组合框中选择一个字段来求和。
平均
如果您要查看某字段的数值平均(均数)，请选择此选项。 然后在以下的组合框中选择一个字段来计算。
计数
计数
如果您要查看某字段的数值数量，例如 count(Visits)，请选择此选项。 然后在以下的组合框中选择一个字段来计数。
自定义
如果您要自定义一个全新表达式，请选择这个选项。 如果需要更多的空间来编辑表达式，点击[...]图标来打开编辑表达式对话框。
完成此向导后，您可随时在图表属性对话框中新增更多的表达式或改变表达式。
< 上一步(B)
下一步(N) >
完成
取消
帮助
请按F1获取帮助



2.10. 永洪BI

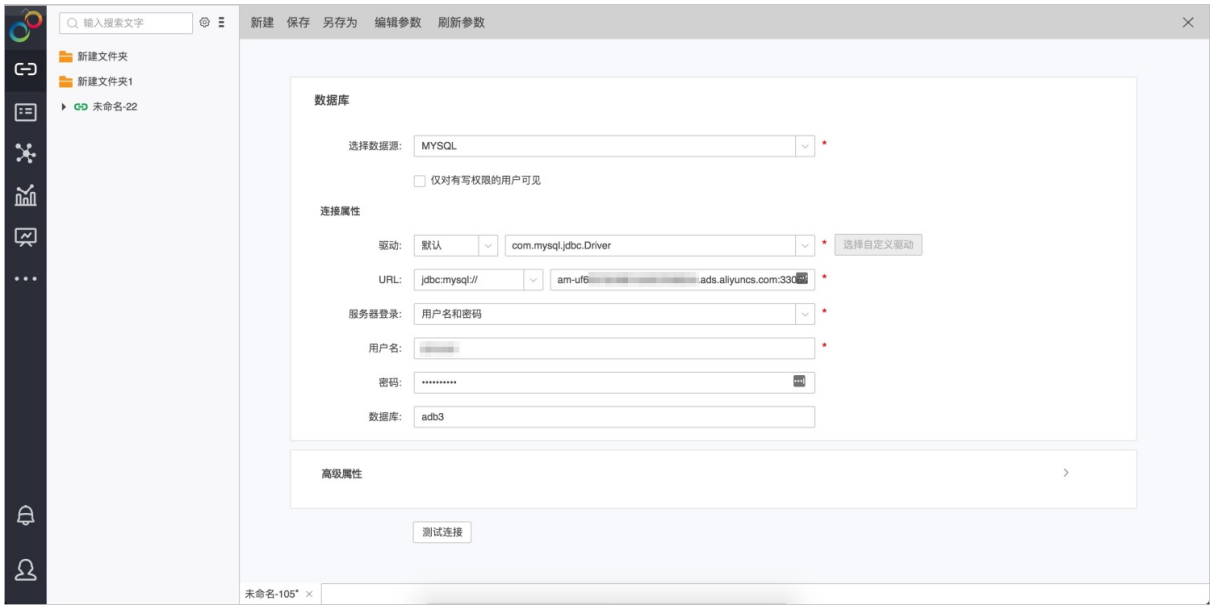
本文测试了永洪BI与AnalyticDB for MySQL在连通性、列举表等方面的兼容性，并给出测试结果图。

测试环境

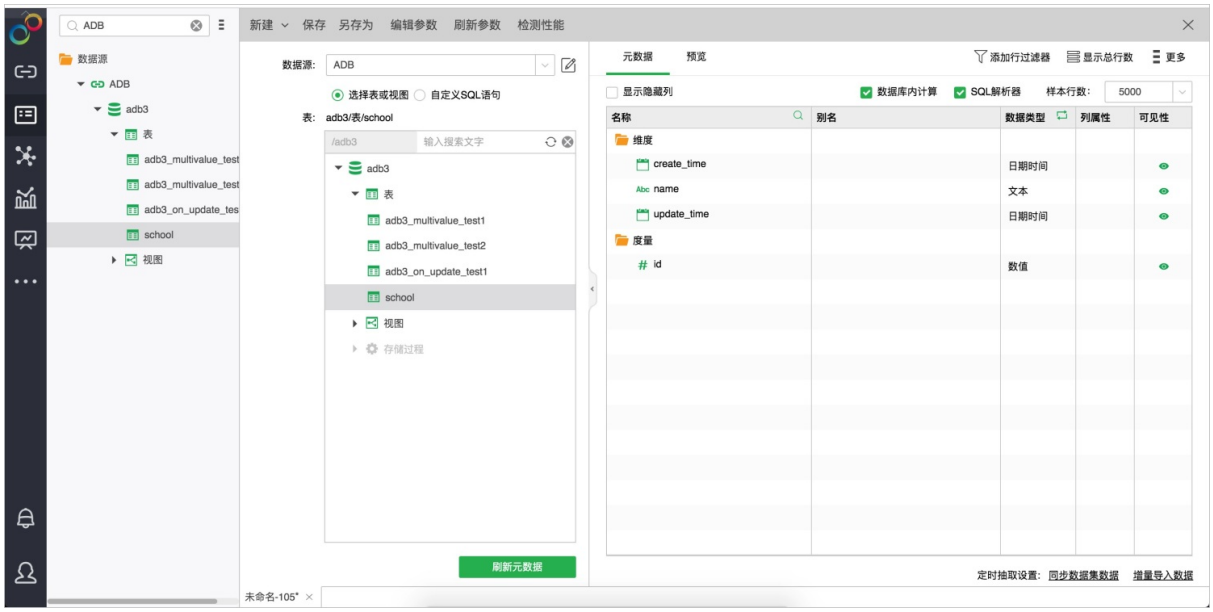
永洪BI试用地址，请参见登录永洪BI。

测试范围

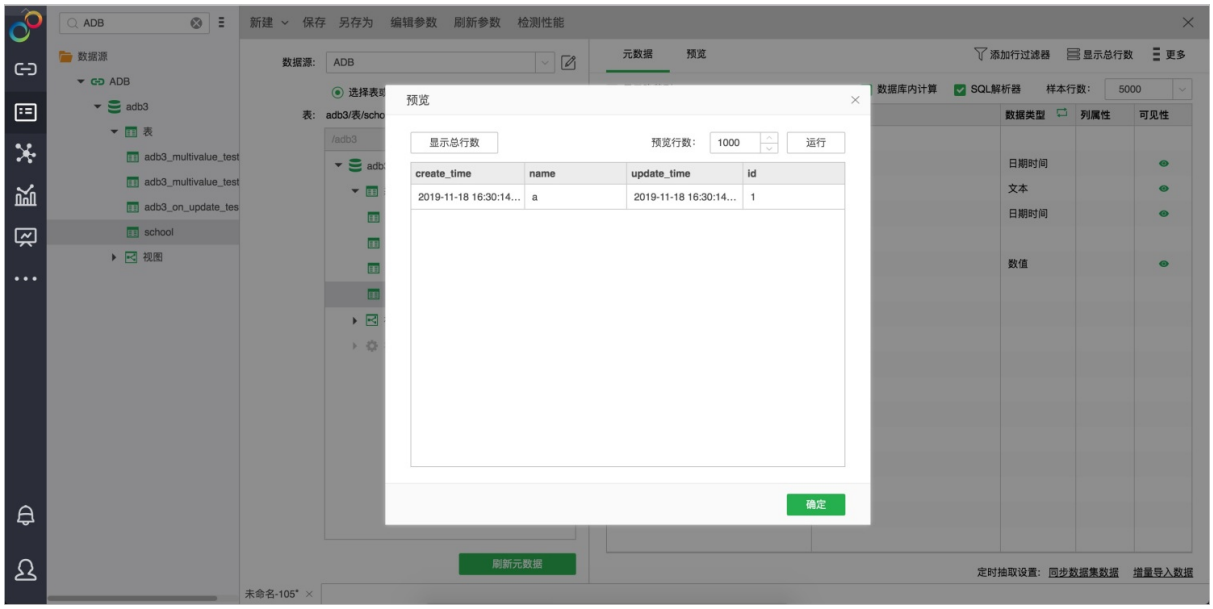
- 连通性



- 查看表结构



- 查看表数据



2.11. FineBI 5.0

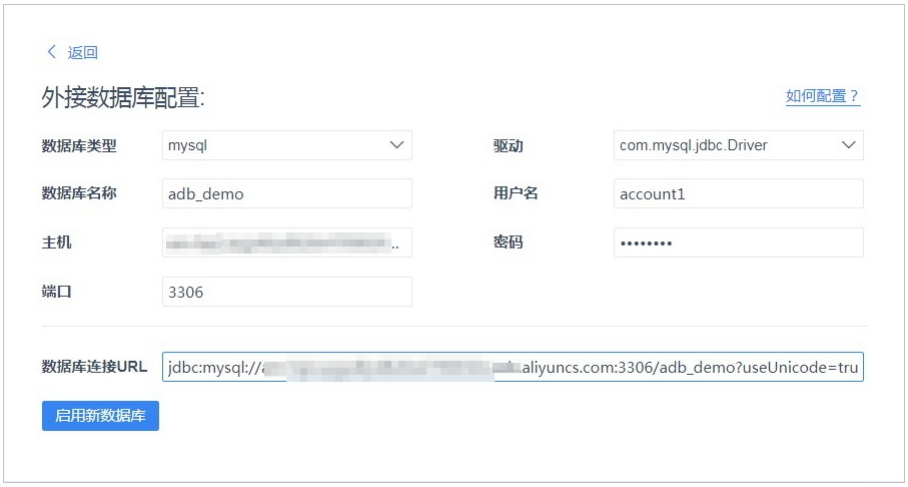
本文测试了FineBI 5.0与AnalyticDB for MySQL在连通性、列举表、查看表数据等方面的兼容性，并给出测试结果图。

测试环境

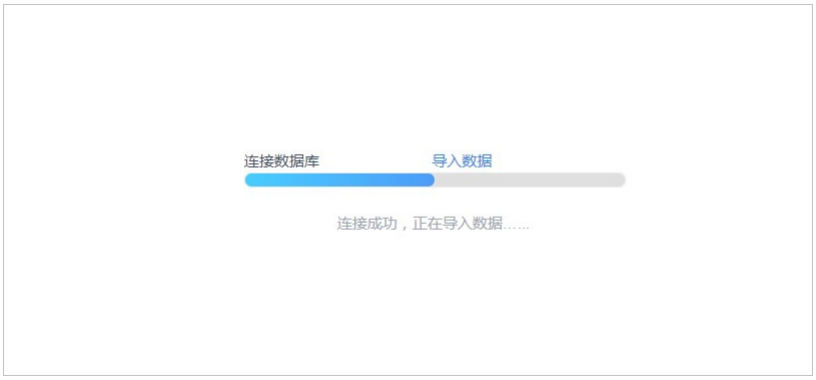
FineBI 5.0下载地址为FineBI 5.0。

测试范围

- 连接FineBI



说明 此处应填写MySQL数据库的连接信息，而不是AnalyticDB for MySQL数据库连接信息。

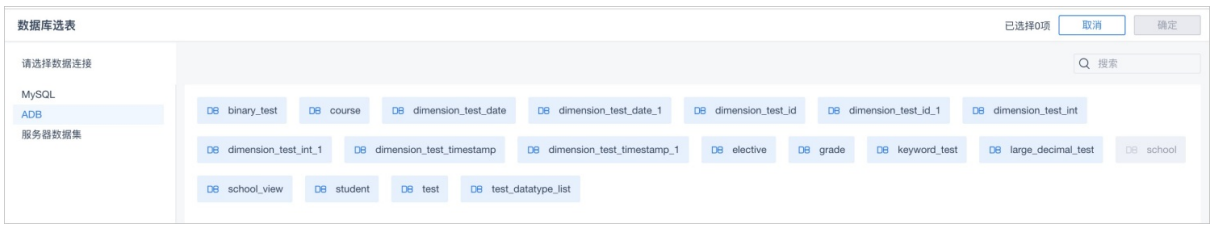


- 新建数据库连接

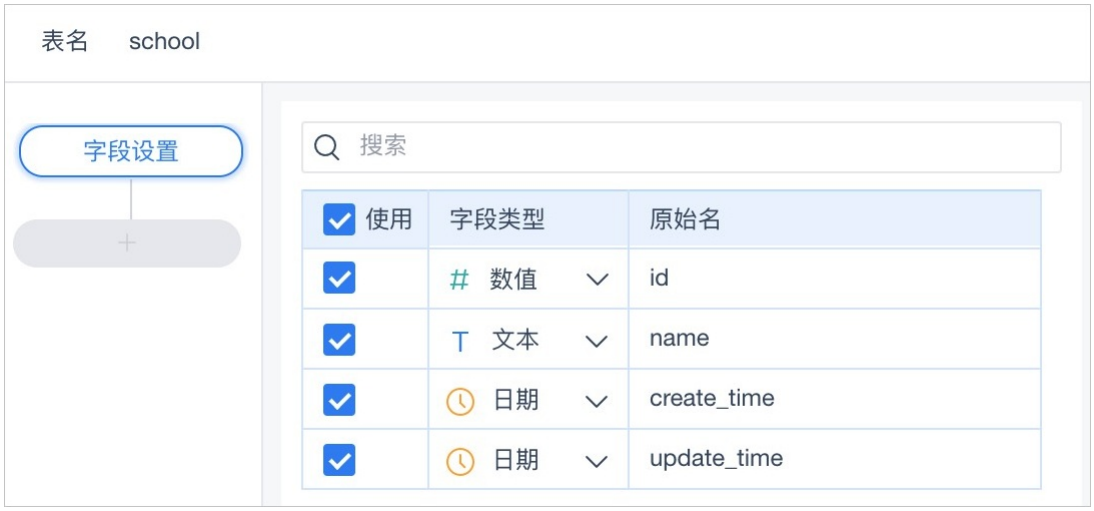

FineBI商业智能
目录
仪表板
数据准备
管理系统
目录管理
用户管理
权限管理
模板管理
外观配置
系统管理
定时调度
移动平台
注册管理
智能运维
数据连接
数据连接管理
连接池状态
新建数据连接
APACHE KYLIN
DERBY
HP Vertica
IBM DB2
INFORMIX
Microsoft SQL Server
MySQL
Oracle
Pivotal Greenplum Databa...
Postgresql
GaussDB 200
更多数据连接...
其他


数据连接(MySQL)
数据连接名
AnalyticDB for MySQL
驱动器
com.mysql.jdbc.Driver
URL
jdbc:mysql://am-bp ads...
编码
自动
用户名
密码
连接池属性
SQL验证查询
select 1
获取连接前校验
是
最大活动连接数
50
测试连接

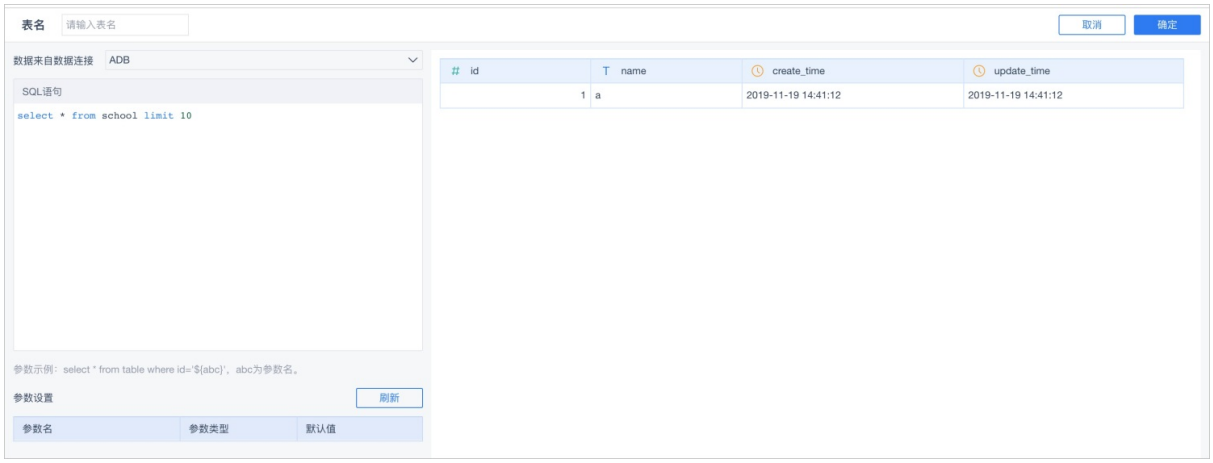
- 列举表


数据库选表
已选择0项
取消
确定
请选择数据连接
MySQL
ADB
服务器数据集
搜索
binary_test
course
dimension_test_date
dimension_test_date_1
dimension_test_id
dimension_test_id_1
dimension_test_int
dimension_test_int_1
dimension_test_timestamp
dimension_test_timestamp_1
elective
grade
keyword_test
large_decimal_test
school
school_view
student
test
test_datatype_list

- 查看表结构



- 查看表数据



2.12. FineReport 10.0

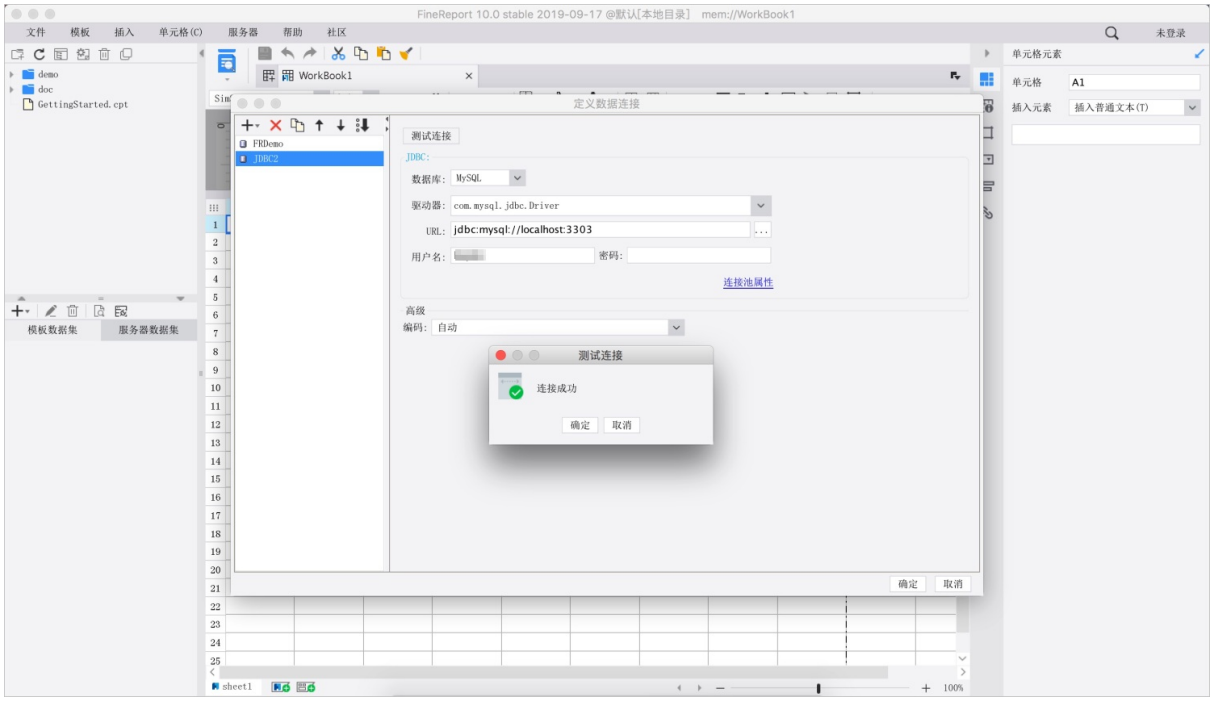
本文测试了FineReport 10.0与AnalyticDB for MySQL在连通性、列举表、查看表数据等方面的兼容性，并给出测试结果图。

测试环境

FineReport 10.0官网地址为FineReport 10.0。

测试范围

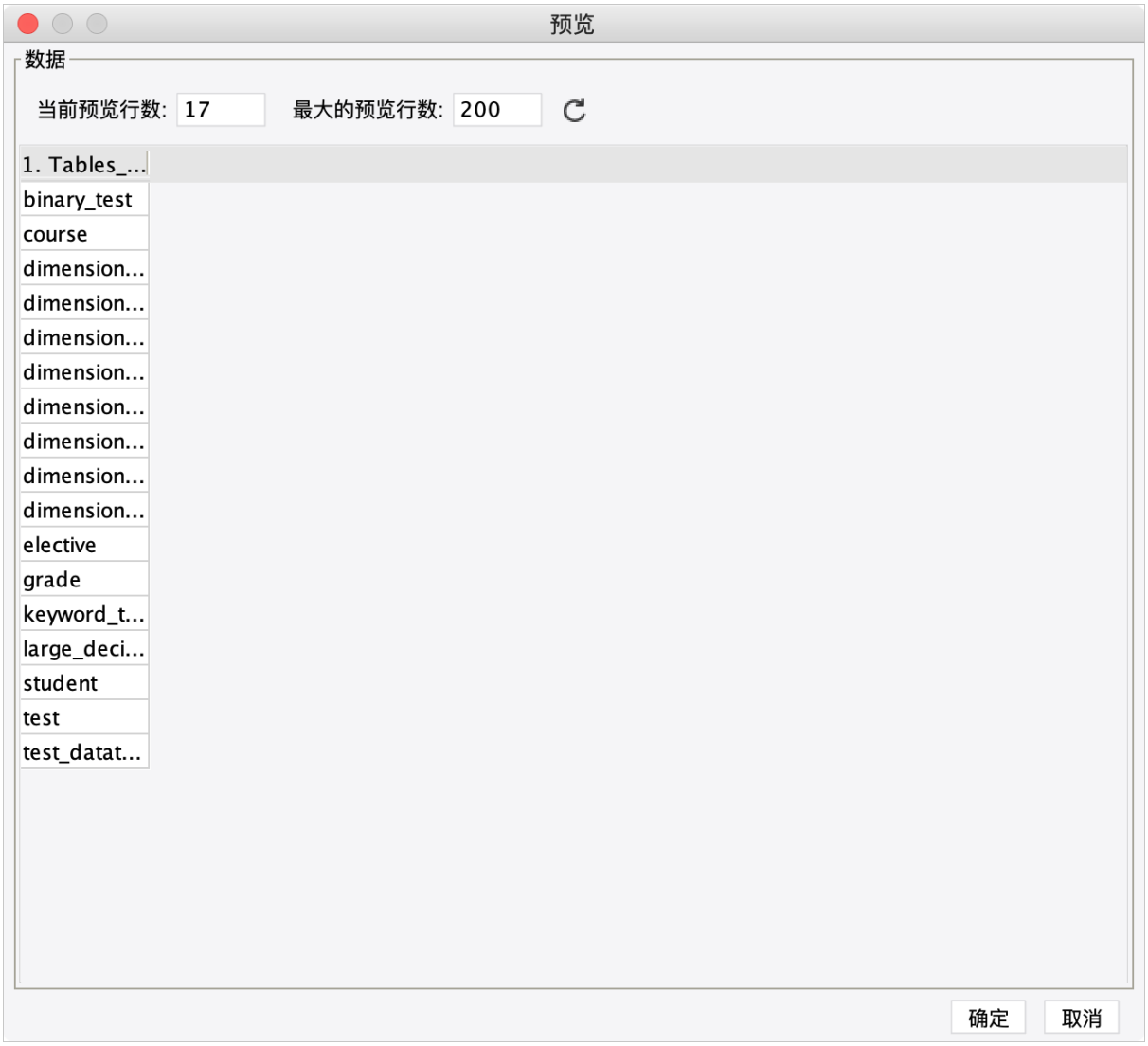
- 连接FineReport 10.0


FineReport 10.0 stable 2019-09-17 @默认[本地目录] mem://WorkBook1
文件
模板
插入
单元格(C)
服务器
帮助
社区
未登录
demo
doc
GettingStarted.cpt
模板数据集
服务器数据集
WorkBook1
定义数据连接
FRDemo
JDBC2
测试连接
JDBC:
数据库: MySQL
驱动器: com.mysql.jdbc.Driver
URL: jdbc:mysql://localhost:3303
用户名:
密码:
连接池属性
高级
编码: 自动
测试连接
连接成功
确定
取消
确定
取消
单元格元素
单元格 A1
插入元素 插入普通文本(T)
sheet1
100%

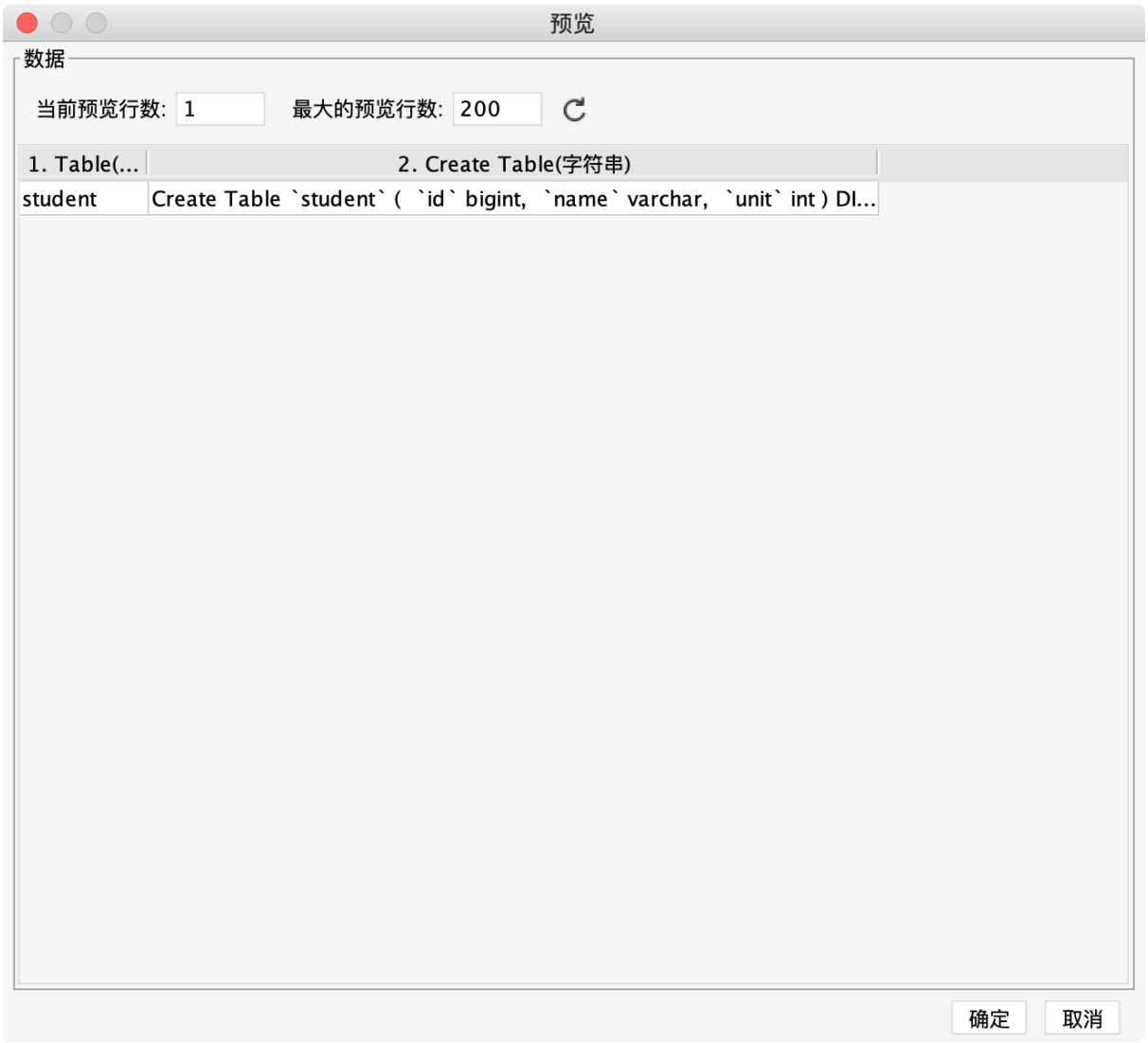
- 列举数据库


预览
数据
当前预览行数: 3
最大的预览行数: 200
1. Databa...
MYSQL
INFORMA...
test4dmp
确定
取消

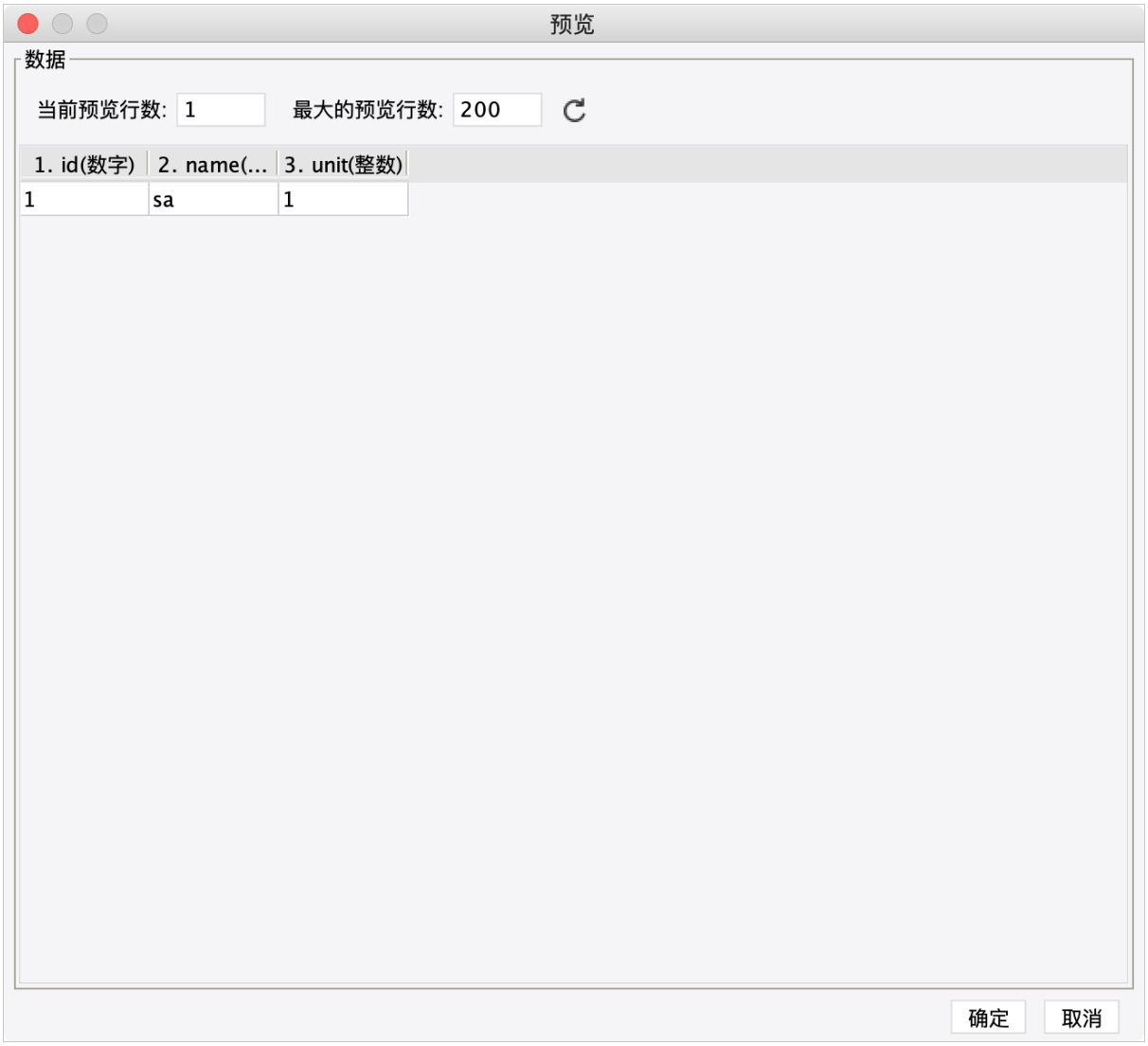
- 列举表


预览
数据
当前预览行数: 17
最大的预览行数: 200
1. Tables_...
binary_test
course
dimension...
dimension...
dimension...
dimension...
dimension...
dimension...
dimension...
dimension...
elective
grade
keyword_t...
large_deci...
student
test
test_datat...
确定
取消

- 查看表结构


预览
数据
当前预览行数: 1
最大的预览行数: 200
1. Table(...
2. Create Table(字符串)
student
Create Table `student` (`id` bigint, `name` varchar, `unit` int) DI...
确定
取消

- 查看表数据



- 查看视图

2.13. Scriptella

本文列出了Scriptella与AnalyticDB for MySQL在连通性、创建表、查询表数据、写入数据、创建视图等方面的兼容性。

测试环境

Java	JDBC	Scriptella
Java版本1.8.0_231	JDBC版本5.1.48	Scriptella下载地址为Scriptella，版本为1.2。

测试范围

- 执行命令

```
java -jar scriptella.jar -debug etl.xml
```

- ETL脚本

```
<!DOCTYPE etl SYSTEM "http://scriptella.org/dtd/etl.dtd">
<etl>
    <connection id="adb" driver="mysql" url="jdbc:mysql://127.0.0.1:3303/test4dmp" user="k*****" password="" classpath="/Library/scriptella-1.2/mys
ql-connector-java-5.1.48.jar;/Library/scriptella-1.2/mysql-connector-java-5.1.48-bin.jar" />
    <!-- DROP TABLE -->
    <script connection-id="adb">
        drop table if exists `student_etl`;
    </script>
    <!-- CREATE TABLE -->
    <script connection-id="adb">
        Create Table `student_etl` (`id` bigint, `name` varchar, `unit` int ) DISTRIBUTE BY HASH(`id`) INDEX_ALL='Y';
    </script>
    <!-- QUERY -->
    <query connection-id="adb">
        SELECT * FROM student
    </query>
    <!-- INSERT TABLE -->
    <script connection-id="adb">
        insert into student_etl select * from student;
    </script>
    <!-- CREATE VIEW -->
    <script connection-id="adb">
        create view student_view as select * from student;
    </script>
</etl>
```

- 返回值

```
2019-12-4 15:02:31 <信息> Execution Progress.Initializing properties: 1%
2019-12-4 15:02:31 <详细> registerDriver: com.mysql.jdbc.Driver@6f539caf
2019-12-4 15:02:31 <详细> Found driver class com.mysql.jdbc.Driver
2019-12-4 15:02:31 <详细> DriverManager.getConnection("jdbc:mysql://127.0.0.1:3303/test4dmp")
2019-12-4 15:02:31 <详细>     trying com.mysql.jdbc.Driver
2019-12-4 15:02:32 <详细> getConnection returning com.mysql.jdbc.Driver
2019-12-4 15:02:32 <详细> jdbc:mysql://127.0.0.1:3303/test4dmp: Statement cache is enabled (cache size 64). Statement separator ';'. Autocommit: fal
se.
2019-12-4 15:02:32 <信息> Execution Progress.Initialized connection JdbcConnection{com.mysql.jdbc.JDBC4Connection}, Dialect{MySQL 5.1.35-analyticdb}
, properties {}: 5%
2019-12-4 15:02:32 <信息> Execution Progress./etl/script[2] prepared: 6%
2019-12-4 15:02:32 <信息> Execution Progress./etl/script[3] prepared: 7%
2019-12-4 15:02:32 <信息> Execution Progress./etl/script[4] prepared: 10%
2019-12-4 15:02:32 <信息> Registered JMX mbean: scriptella:type=etl,url="file:/Library/scriptella-1.2/etl.xml"
2019-12-4 15:02:32 <详细> Executing script /etl/script[1]
2019-12-4 15:02:33 <详细>     Executed statement drop table if exists `student_etl`. Update count: 0
2019-12-4 15:02:33 <详细> Script /etl/script[1] completed
2019-12-4 15:02:33 <信息> Execution Progress./etl/script[1] executed: 27%
2019-12-4 15:02:33 <详细> Executing script /etl/script[2]
2019-12-4 15:02:34 <详细>     Executed statement Create Table `student_etl` (`id` bigint, `name` varchar, `unit` int ) DISTRIBUTE BY HASH(`id`) IND
EX_ALL='Y'. Update count: 0
2019-12-4 15:02:34 <详细> Script /etl/script[2] completed
2019-12-4 15:02:34 <信息> Execution Progress./etl/script[2] executed: 44%
2019-12-4 15:02:34 <详细> Executing query /etl/query[1]
2019-12-4 15:02:34 <详细> Processing row #1 for query /etl/query[1]
2019-12-4 15:02:34 <详细> Processing row #2 for query /etl/query[1]
2019-12-4 15:02:34 <详细>     Executed statement SELECT * FROM student
2019-12-4 15:02:34 <详细> Query /etl/query[1] processed.
2019-12-4 15:02:34 <信息> Execution Progress./etl/query[1] executed: 61%
2019-12-4 15:02:34 <详细> Executing script /etl/script[3]
2019-12-4 15:02:34 <详细>     Executed statement insert into student_etl select * from student. Update count: 2
2019-12-4 15:02:34 <详细> Script /etl/script[3] completed
2019-12-4 15:02:34 <信息> Execution Progress./etl/script[3] executed: 78%
2019-12-4 15:02:34 <详细> Executing script /etl/script[4]
2019-12-4 15:02:35 <详细>     Executed statement create view student_view as select * from student. Update count: 0
2019-12-4 15:02:35 <详细> Script /etl/script[4] completed
2019-12-4 15:02:35 <信息> Execution Progress./etl/script[4] executed: 95%
2019-12-4 15:02:35 <信息> Execution Progress.Complete
2019-12-4 15:02:35 <详细> Commiting connection JdbcConnection{com.mysql.jdbc.JDBC4Connection}
2019-12-4 15:02:35 <详细> Closing JdbcConnection{com.mysql.jdbc.JDBC4Connection}
2019-12-4 15:02:35 <信息> Execution statistics:
Executed 1 query, 4 scripts, 5 statements
/etl/script[1]: Element successfully executed (1 statement). Working time 897 milliseconds. Avg throughput: 1.11 statements/sec.
/etl/script[2]: Element successfully executed (1 statement). Working time 821 milliseconds. Avg throughput: 1.22 statements/sec.
/etl/query[1]: Element successfully executed (1 statement). Working time 36 milliseconds. Avg throughput: 27.27 statements/sec.
/etl/script[3]: Element successfully executed (1 statement). Working time 702 milliseconds. Avg throughput: 1.42 statements/sec.
/etl/script[4]: Element successfully executed (1 statement). Working time 303 milliseconds. Avg throughput: 3.3 statements/sec.
Total working time: 3.17 seconds
2019-12-4 15:02:35 <信息> Successfully executed ETL file /Library/scriptella-1.2/etl.xml
```

```
mysql> select * from student_etl;
--------------
select * from student_etl
--------------

+------+------+------+
| id   | name | unit |
+------+------+------+
|    1 | a    |    1 |
|    2 | b    |    2 |
+------+------+------+
2 rows in set (0.06 sec)

mysql> select * from student_view;
--------------
select * from student_view
--------------

+------+------+------+
| ID   | NAME | UNIT |
+------+------+------+
|    2 | b    |    2 |
|    1 | a    |    1 |
+------+------+------+
2 rows in set (0.04 sec)
```

2.14. Smartbi

本文列出了Smartbi与AnalyticDB MySQL版在连通性、列举数据库、查询表数据等方面的兼容性。

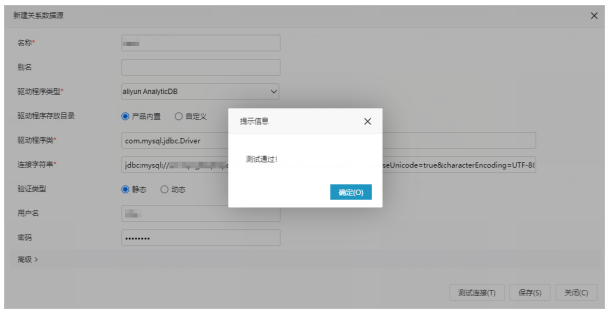
测试环境

Smartbi试用地址为登录Smartbi。

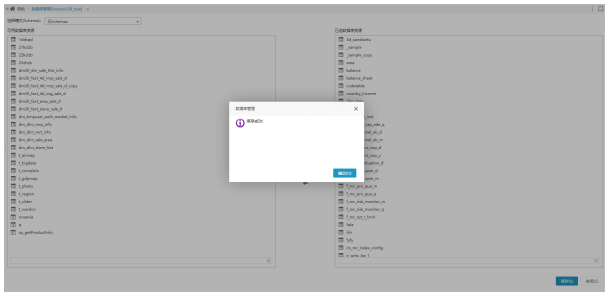
如何通过Smartbi连接AnalyticDB MySQL版集群，请参见Smartbi。

测试范围

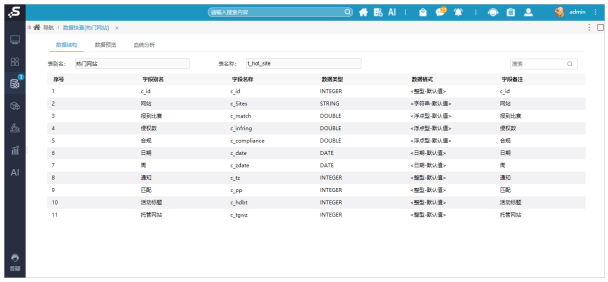
- 连接AnalyticDB MySQL版集群数据库



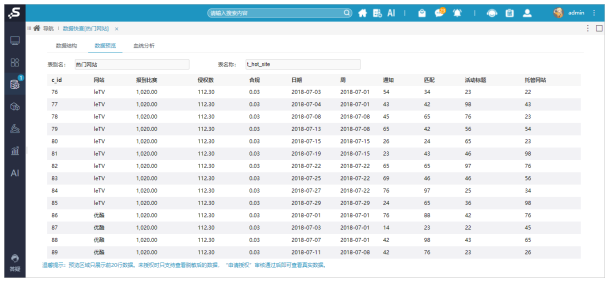
- 列举数据库



- 查看表结构



- 查看表数据



3.MySQL命令行连接AnalyticDB for MySQL

本文介绍如何通过MySQL命令行工具连接AnalyticDB for MySQL。

语法

```
mysql -hadb_url -P3306 -uadb_user -padb_password
```

参数

- `adb_url`：AnalyticDB for MySQL集群的连接地址，通过控制台**集群信息**页面中的**网络信息**区域获取连接地址。
- `3306`：端口为`3306`。
- `adb_user`：AnalyticDB for MySQL集群中的高权限账号或者拥有相关权限的普通账号。
- `adb_password`：账号对应的密码。

示例

```
mysql -ham-bp****.ads.aliyuncs.com -P3306 -utest -pTest123
```

4.业务系统连接AnalyticDB for MySQL

4.1. Java

本文介绍如何在Java中通过MySQL JDBC连接AnalyticDB for MySQL集群。

MySQL JDBC驱动版本

AnalyticDB for MySQL支持以下版本的MySQL JDBC驱动。

- 5.0版本系列：5.0.2，5.0.3，5.0.4，5.0.5，5.0.7，5.0.8。
- 5.1版本系列：
  5.1.1，5.1.2，5.1.3，5.1.4，5.1.5，5.1.6，5.1.7，5.1.8，5.1.11，5.1.12，5.1.13，5.1.14，5.1.15，5.1.16，5.1.17，5.1.18，5.1.19，5.1.20，5.1.21，5.1.22，5.1.23，5.1.5.1.32, 5.1.33, 5.1.34。
- MySQL 8.0。

注意事项

Java中创建MySQL JDBC连接依赖于MySQL-JDBC驱动包，您需要手动将MySQL-JDBC驱动包（mysql-connector-java-x.x.x.jar）加入到 `CLASSPATH` 中，否则无法创建MySQL JDBC连接。

不带重试的JDBC连接示例

您可以在业务系统的Java代码中添加以下代码，通过MySQL JDBC连接AnalyticDB for MySQL数据库。

```
Connection connection = null;
Statement statement = null;
ResultSet rs = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    //adb_url是AnalyticDB for MySQL集群的连接地址URL，可以在控制台的集群信息页面获取连接URL，3306是端口号。
    //db_name是AnalyticDB for MySQL集群中的数据库名称。
    String url = "jdbc:mysql://adb_url:3306/db_name?useUnicode=true&characterEncoding=UTF-8";
    Properties connectionProps = new Properties();
    //account_name是AnalyticDB for MySQL集群中的用户账号：高权限账号或者普通账号。
    connectionProps.put("user", "account_name");
    //account_password是AnalyticDB for MySQL集群中用户账号对应的密码。
    connectionProps.put("password", "account_password");
    connection = DriverManager.getConnection(url, connectionProps);
    statement = connection.createStatement();
    String query = "select count(*) from information_schema.tables";
    rs = statement.executeQuery(query);
    while (rs.next()) {
        System.out.println(rs.getObject(1));
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    if (rs != null) {
        try {
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (statement != null) {
        try {
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (connection != null) {
        try {
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

带重试的JDBC连接示例

在JDBC中通过配置参数可以实现连接重试机制。

```
public static final int MAX_QUERY_RETRY_TIMES = 3;
public static Connection conn = null;
public static Statement statement = null;
public static ResultSet rs = null;
public static void main(String[] args) throws ClassNotFoundException {
    //AnalyticDB for MySQL集群中的数据库名称。
```

```
    String yourDB = "db_name";
    //AnalyticDB for MySQL集群中的用户账号：高权限账号或者普通账号。
    String username = "account_name";
    //AnalyticDB for MySQL集群中用户账号对应的密码。
    String password = "account_password";
    Class.forName("com.mysql.jdbc.Driver");
    //adb_url是AnalyticDB for MySQL集群的连接地址URL，可以在控制台的集群信息页面获取连接URL，3306是端口号。
    String url = "jdbc:mysql://adb_url:3306/" + yourDB + "?useUnicode=true&characterEncoding=UTF-8";
    Properties connectionProps = new Properties();
    connectionProps.put("user", username);
    connectionProps.put("password", password);
    String query = "select id from test4dmp.test limit 10";
    int retryTimes = 0;
    // 通过循环自动重试。
    while (retryTimes < MAX_QUERY_RETRY_TIMES) {
        try {
            getConn(url, connectionProps);
            execQuery(query);//执行query。
            break; // query执行成功后，结束整个循环。
        } catch (SQLException e) {
            System.out.println("Met SQL exception: " + e.getMessage() + ", then go to retry task ...");
            try {
                if (conn == null || conn.isClosed()) {
                    retryTimes++;
                }
            } catch (SQLException e1) {
                if (conn != null) {
                    try {
                        conn.close();
                    } catch (SQLException e2) {
                        e.printStackTrace();
                    }
                }
            }
        }
    }
    // Clear connection resource.
    closeResource();
}
/**
 * Get connection.
 *
 * @param url
 * @param connectionProps
 * @throws SQLException
 */
public static void getConn(String url, Properties connectionProps) throws SQLException {
    conn = DriverManager.getConnection(url, connectionProps);
}
/**
 * Query task execution logic.
 *
 * @param sql
 * @throws SQLException
 */
public static void execQuery(String sql) throws SQLException {
    Statement statement = null;
    ResultSet rs = null;
    statement = conn.createStatement();
    for (int i = 0; i < 10; i++) {
        long startTs = System.currentTimeMillis();
        rs = statement.executeQuery(sql);
        int cnt = 0;
        while (rs.next()) {
            cnt++;
            System.out.println(rs.getObject(1) + " ");
        }
        long endTs = System.currentTimeMillis();
        System.out.println("Elapse Time: " + (endTs - startTs));
        System.out.println("Row count: " + cnt);
        try {
            Thread.sleep(160000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
/**
 * Close connection resource.
 */
public static void closeResource() {
    if (rs != null) {
        try {
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
```

```
        }
    }
    if (statement != null) {
        try {
            statement.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

4.2. Druid连接池配置

本文介绍如何通过JDBC Druid连接池连接云原生数据仓库AnalyticDB MySQL版。

注意事项

- 仅支持Druid 1.1.16以上的版本。
- 使用Druid连接池连接AnalyticDB MySQL时，建议配置 `<property name="keepAlive" value="true" />` ，可以复用连接和避免短连接。
- AnalyticDB MySQL前端接入节点使用SLB做负载均衡，客户端连接前端节点。为了将客户端请求较为均匀地发送到各个前端节点，建议连接池里有稍多一些连接，且连接保持的时间不宜太长。
- 根据业务情况合理调整连接池配置，并确保配置生效。您可以在程序中通过 `DruidDataSource#getStatData()` 和 `DruidDataSource#dump()` 方法定期获取生效的配置与连接池的信息，并在日志中查看、核对相关配置信息。
- 执行查询前从连接池获取连接，执行完查询后要及时调用 `conn.close()` 将连接返回到连接池中。再次使用时，请从连接池中获取新连接，避免长时间使用一个连接导致Druid无法及时检查到连接失效。
- 复杂网络情况下，当遇到网关性能达到瓶颈、网络链路长、网络抖动、重传率或丢包率高等情形，可能会导致连接中断。在确保连接池合理配置都生效的情况下，如果仍然有连接中断的情况，建议业务代码侧通过重试进行优化。

配置Druid连接池

```
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
    <!-- jdbc_url是AnalyticDB for MySQL集群的连接地址URL，可以在控制台的集群信息页面获取连接URL。-->
    <property name="url" value="${jdbc_url}" />
    <!-- jdbc_user是AnalyticDB for MySQL集群中的用户账号：高权限账号或者普通账号。 -->
    <property name="username" value="${jdbc_user}" />
    <!-- jdbc_password是AnalyticDB for MySQL集群中用户账号对应的密码。 -->
    <property name="password" value="${jdbc_password}" />
    <!-- 配置初始化连接池大小、最小连接数、最大连接数。 -->
    <property name="initialSize" value="5" />
    <property name="minIdle" value="10" />
    <property name="maxActive" value="20" />
    <!-- 配置获取连接等待超时的时间。 -->
    <property name="maxWait" value="60000" />
    <!-- 配置一个连接在连接池中的最小生存时间、最大生存时间，超过最大生存时间会被移除，单位毫秒。 -->
    <property name="minEvictableIdleTimeMillis" value="600000" />
    <property name="maxEvictableIdleTimeMillis" value="900000" />
    <!-- 配置间隔多久进行一次检测，检测需要关闭的空闲连接，单位毫秒。 默认是60s，太长可能会导致无法及时检测到连接中断。 -->
    <property name="timeBetweenEvictionRunsMillis" value="2000" />
    <!-- 配置从连接池获取连接时，当连接空闲时间大于timeBetweenEvictionRunsMillis时是否检查连接有效性，true每次都检查；false不检查。 -->
    <property name="testWhileIdle" value="true" />
    <!-- 配置从连接池获取连接时和向连接池归还连接时，是否检查连接有效性。 -->
    <!-- 每次获取或归还连接都检测太频繁，除非特别重要或网络特别不可靠等情况，建议用testWhileIdle + timeBetweenEvictionRunsMillis代替。 -->
    <property name="testOnBorrow" value="false" />
    <property name="testOnReturn" value="false" />
    <!-- 配置是否定期探活、探活间隔。 -->
    <property name="keepAlive" value="true" />
    <property name="keepAliveBetweenTimeMillis" value="3000" /> <!-- 默认120s。 -->
    <!-- 探活、验证链接有效性的查询，新版本默认使用mysqlPing代替。 -->
    <property name="validationQuery"  value="select 1" />
    <!-- 配置监控统计拦截的filters。 -->
    <property name="filters" value="stat" />
</bean>
```

生效ValidationQuery

Druid新版本默认使用mysqlPing代替执行ValidationQuery进行探活，如果需要执行ValidationQuery可以进行如下设置：

```
...
druidDataSource.init();
Properties properties = new Properties();
properties.setProperty("druid.mysql.usePingMethod", "false");
druidDataSource.setConnectProperties(properties);
```

获取连接池信息

使用 `dataSource.dump()` 可以获取连接池和连接的基本信息。

```
{
    CreateTime:"2022-06-01 15:28:10",    # 连接池创建时间
    ActiveCount:0,      # 从连接池取出来在用的连接数
    PoolingCount:2,     # 在连接池中，未取出去使用的连接数
    CreateCount:2,      # 创建过的连接数，连接销毁后重新创建会增加该数量
    DestroyCount:0,     # 销毁过的连接数，连接异常或到期销毁会增加该数量
    CloseCount:4,       # 调用conn.close()将连接还到连接池的次数
    ConnectCount:4,     # 调用dataSource.getConnection()获取连接的次数
    Connections:[
        {
            ID:525571,     # 连接ID
            ConnectTime:"2022-06-01 15:28:11",        # 连接创建时间
            UseCount:0,                     # 连接被获取使用的次数，所有UseCount加起来等于上面的ConnectCount
            LastActiveTime:"2022-06-01 15:28:11"     # 连接最后被取出来使用的时间，未使用等于ConnectTime
        },
        {ID:1263877414, ConnectTime:"2022-06-01 15:28:11", UseCount:4, LastActiveTime:"2022-06-01 15:28:11"}
    ]
}
```

获取连接池生效配置

使用 `dataSource.getStatData()` 可以获取连接池生效的配置。

```
{
    Identity=85777802,
    Name=DataSource-85777802,
    DbType=mysql,
    DriverClassName=com.mysql.jdbc.Driver,
    URL=jdbc:mysql://host:port/db_name,
    UserName=haicen,
    FilterClassNames=[],
    WaitThreadCount=0,
    NotEmptyWaitCount=0,
    NotEmptyWaitMillis=0,
    PoolingCount=2,
    PoolingPeak=2,
    PoolingPeakTime=Wed Jun 01 16:08:15 CST 2022,
    ActiveCount=0,
    ActivePeak=1,
    ActivePeakTime=Wed Jun 01 16:08:15 CST 2022,
    InitialSize=1,
    MinIdle=2,
    MaxActive=3,
    QueryTimeout=0,
    TransactionQueryTimeout=0,
    LoginTimeout=0,
    ValidConnectionCheckerClassName=com.alibaba.druid.pool.vendor.MySqlValidConnectionChecker,
    ExceptionSorterClassName=com.alibaba.druid.pool.vendor.MySqlExceptionSorter,
    TestOnBorrow=true,
    TestOnReturn=true,
    TestWhileIdle=true,
    DefaultAutoCommit=true,
    DefaultReadOnly=null,
    DefaultTransactionIsolation=null,
    LogicConnectCount=14,
    LogicCloseCount=14,
    LogicConnectErrorCount=0,
    PhysicalConnectCount=6,
    PhysicalCloseCount=4,
    PhysicalConnectErrorCount=0,
    DiscardCount=0,
    ExecuteCount=14,
    ExecuteUpdateCount=0,
    ExecuteQueryCount=14,
    ExecuteBatchCount=0,
    ErrorCount=0,
    CommitCount=0,
    RollbackCount=0,
    PSCacheAccessCount=0,
    PSCacheHitCount=0,
    PSCacheMissCount=0,
    StartTransactionCount=0,
    TransactionHistogram=[
        J@6a472554,
        ConnectionHoldTimeHistogram=[
            J@7ff2a664,
            RemoveAbandoned=true,
            ClobOpenCount=0,
            BlobOpenCount=0,
            KeepAliveCheckCount=332,
            KeepAlive=true,
            FailFast=false,
            MaxWait=6000,
            MaxWaitThreadCount=-1,
            PoolPreparedStatements=false,
            MaxPoolPreparedStatementPerConnectionSize=10,
            MinEvictableIdleTimeMillis=600000,
            MaxEvictableIdleTimeMillis=900000,
            LogDifferentThread=true,
            RecycleErrorCount=0,
            PreparedStatementOpenCount=0,
            PreparedStatementClosedCount=0,
            UseUnfairLock=false,
            InitGlobalVariants=false,
            InitVariants=false
}
```

4.3. Python

本文介绍如何在Python中通过MySQLdb的module连接AnalyticDB for MySQL集群。

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
import MySQLdb
# 打开数据库连接。
# host是AnalyticDB for MySQL集群的URL或IP。
# port是AnalyticDB for MySQL集群的URL对应的端口。
# user是AnalyticDB for MySQL集群的用户账号：高权限账号或者普通账号。
# passwd是AnalyticDB for MySQL集群的用户账号对应的密码。
# db是AnalyticDB for MySQL集群中的数据库名。
db = MySQLdb.connect(host='am-bp***.ads.aliyuncs.com', port=3306, user='account_name', passwd='account_password', db='db_name')
# 使用cursor()方法获取操作游标。
cursor = db.cursor()
# 使用execute方法执行SQL语句。
cursor.execute("SELECT VERSION()")
# 使用 fetchone() 方法获取一条数据。
data = cursor.fetchone()
print "Database version : %s " % data
# 关闭数据库连接。
db.close()
```

4.4. PHP

本文介绍如何在PHP程序中连接云原生数据仓库AnalyticDB MySQL版集群。

注意事项

- 操作系统为Linux时，需要安装php-mysql 5.1.x模块。
- 操作系统为Windows时，需要安装php_MySQL.dll。
- 如果使用公网地址连接AnalyticDB MySQL集群，您还需要将待访问AnalyticDB MySQL集群的设备IP加入白名单。

使用mysqli连接AnalyticDB MySQL

```
//AnalyticDB MySQL集群的连接地址，可以在控制台的集群信息页面获取连接地址。
$ads_server_name="am-bp***.ads.aliyuncs.com";
//AnalyticDB MySQL集群的用户账号：高权限账号或者普通账号。
$ads_username="account_name";
//AnalyticDB MySQL集群的用户账号的密码。
$ads_password="account_password";
//AnalyticDB MySQL集群的数据库名称。
$ads_database="db_name";
//AnalyticDB MySQL集群的连接端口号。
$ads_port=3306;
//连接AnalyticDB MySQL。
$ads_conn=mysqli_connect($ads_server_name,$ads_username,$ads_password,$ads_database, $ads_port);
$strsql="SELECT user_id FROM my_ads_db.my_first_table limit 20;";
$result=mysqli_query($ads_conn, $strsql);
while($row = mysqli_fetch_array($result)) {
 //user_id为列名
  echo $row["user_id"];
}
```

使用PDO连接AnalyticDB MySQL

> **说明** 如果需要在PDO中开启PrepareStatement，请参见PDO中开启PrepareStatement。

```
//AnalyticDB MySQL集群的连接地址，可以在控制台的集群信息页面获取连接地址。
$ads_server_name = "am-bp***.ads.aliyuncs.com";
//AnalyticDB MySQL集群的用户账号：高权限账号或者普通账号。
$ads_username = "account_name";
//AnalyticDB MySQL集群的用户账号的密码。
$ads_password = "account_password";
//AnalyticDB MySQL集群的数据库名称。
$ads_database = 'db_name';
//AnalyticDB MySQL集群的连接端口号。
$ads_port = 3306;
$dsn = "mysql:host={$ads_server_name};dbname={$ads_database};port={$ads_port}";
try {
    $dbh = new PDO($dsn, $ads_username, $ads_password);
    echo 'PDO Success !';
} catch (PDOException $e) {
    echo 'PDO Connection failed: ' . $e->getCode() ."\n" . $e->getMessage() ."\n". $e->getTraceAsString();
}
```

4.5. C#（Mac）

ADB支持在Visual Studio for Mac中通过C#连接ADB。

前提条件

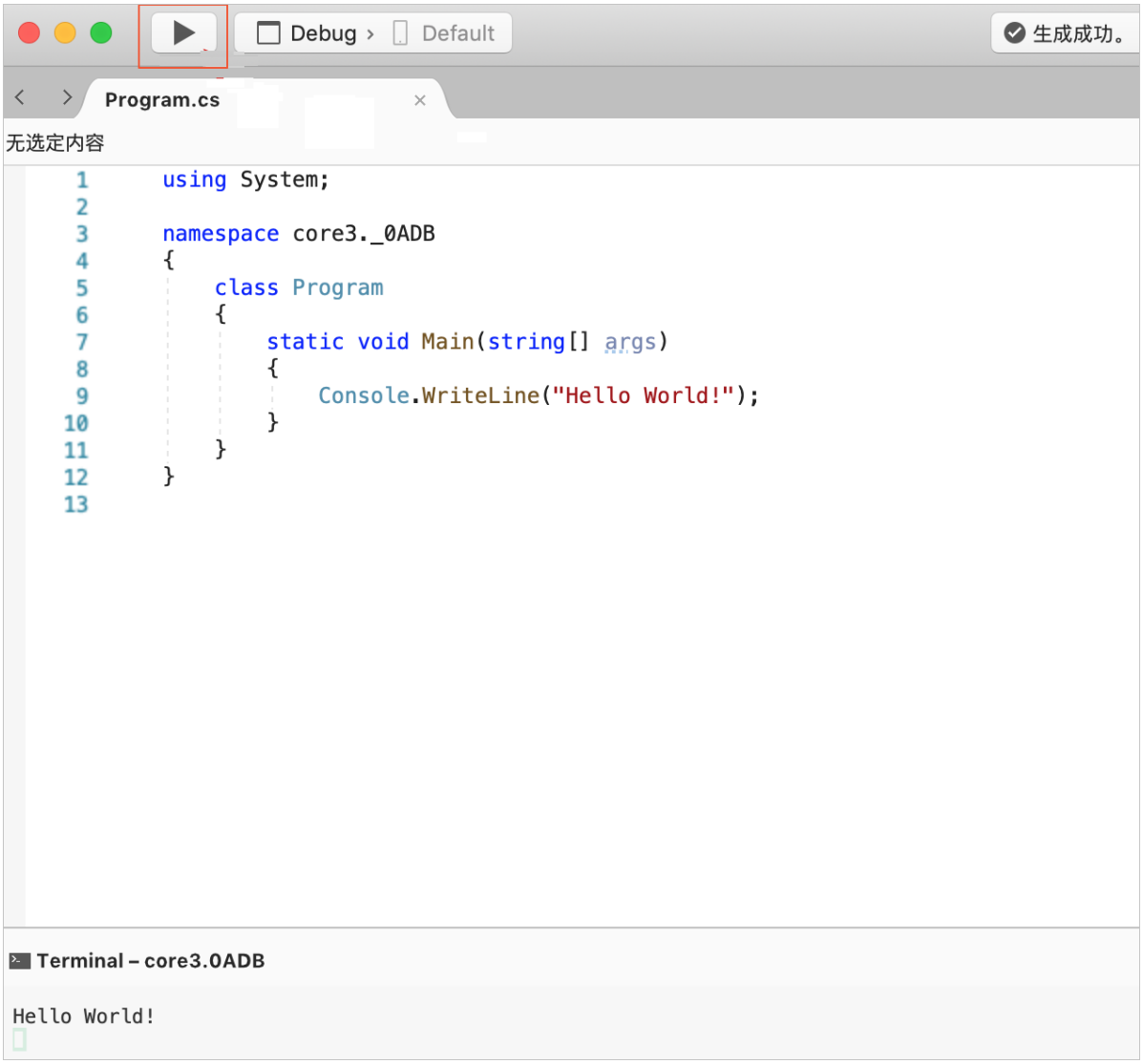
- 下载和安装Visual Studio for Mac，本文测试使用的是Visual Studio for Mac 8.6.5版本。
- 根据ADB快速入门，在ADB中准备测试数据。

```
create table t1 (a int, s1 varchar)DISTRIBUTE BY HASH(`a`) ENGINE='CSTORE';
insert into t1 values (11, 'test1'), (22, 'test2'), (33, 'test3'), (44, 'test4');
```

```
create user test identified by 'test_123456';
grant select on test.* to test;
```

操作步骤

1. 打开Visual Studio。
2. 单击**文件 > 新建解决方案 > 控制台应用程序**，然后单击**下一步**。
3. 根据系统提示输入项目名称，创建一个名为hello world的示例项目，然后单击左上角的**运行**，系统输出运行结果。



4. 修改上述示例代码，增加连接ADB并输出t1表结果相关代码。

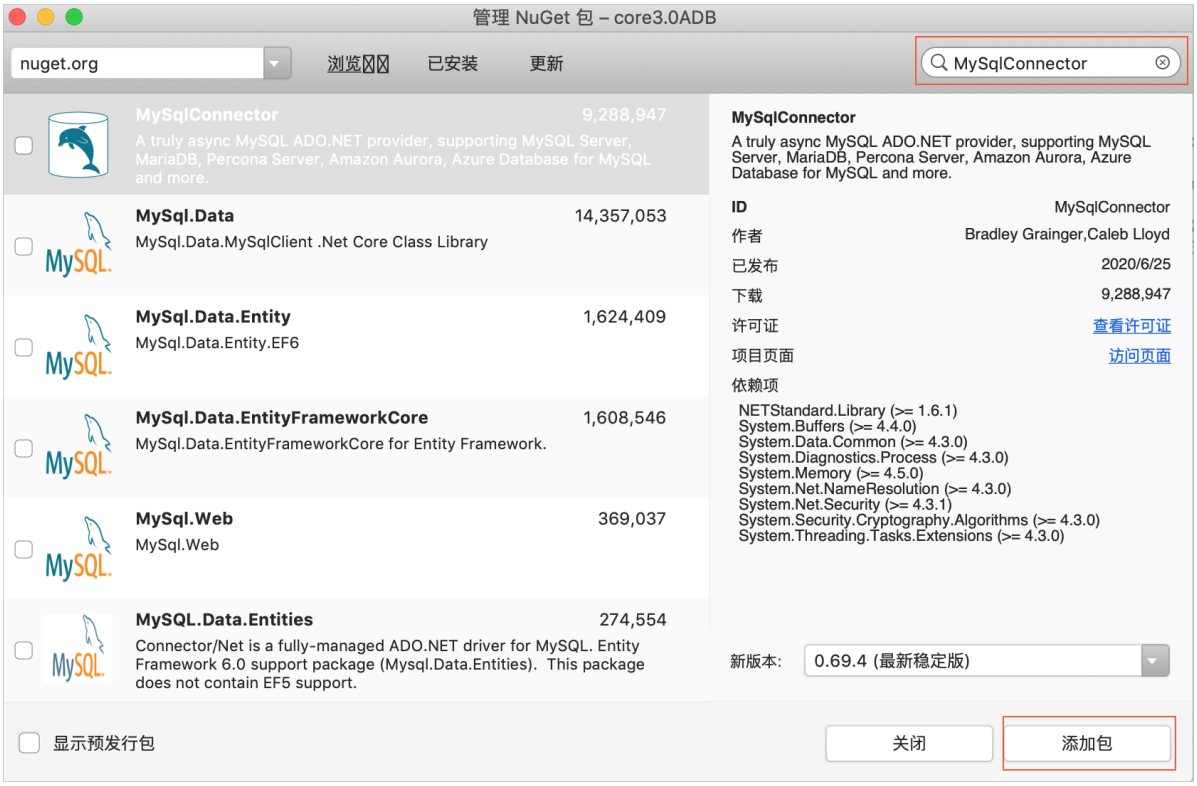
```
using System;
using MySql.Data.MySqlClient;
namespace connectADB
{
    class Program
    {
        static void Main(string[] args)
        {
            string connStr = "server=127.0.0.1;UID=test;database=test;port=3303;password=test_123456;SslMode=none;";
            MySqlConnection conn = new MySqlConnection(connStr);
            try
            {
                Console.WriteLine("Connecting to MySQL...");
                conn.Open();
                string sql = "select * from `t1`";
                MySqlCommand cmd = new MySqlCommand(sql, conn);
                MySqlDataReader rdr = cmd.ExecuteReader();
                while (rdr.Read())
                {
                    Console.WriteLine(rdr[0] + " --- " + rdr[1]);
                }
                rdr.Close();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
            }
            conn.Close();
            Console.WriteLine("Done.");
        }
    }
}
```

代码修改完成后，系统提示错误，需要引入MySqlConnector包。

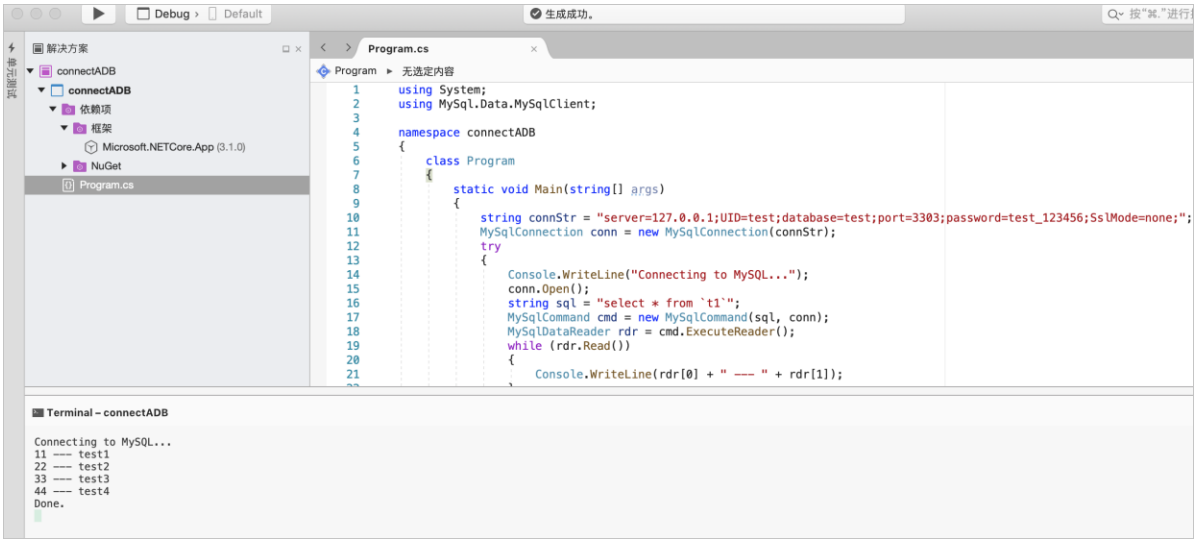
5. 右键单击**解决方案**，选择**管理NuGet包**。

> **说明** C#连接ADB时依赖MySqlConnector包。

6. 在**管理NuGet包**页面，在搜索框中输入MySqlConnector，然后单击**添加包**。



7. MySqlConnector包添加成功后，错误提示消失，单击左上角的**运行**系统输出正确结果。



4.6. Golang

本文介绍如何使用Golang连接AnalyticDB MySQL版。

前提条件

- 已安装了Git，并且已将Git路径添加到Path环境变量中。
- 下载并安装Golang。
- 安装Golang MySQL Driver。
  i. 下载 Golang MySQL Driver。
  ii. 使用Shell中的go工具将驱动包安装到 `$GOPATH` 中。

```
go get github.com/go-sql-driver/mysql
```

连接AnalyticDB MySQL版

```
package main
import (
    "database/sql"
    "fmt"
    _ "github.com/go-sql-driver/mysql"
)
const (
   //user是AnalyticDB MySQL版集群中的用户账号：高权限账号或者普通账号。
    user = "adb_test"
    //password是AnalyticDB MySQL版集群中用户账号对应的密码。
    password = "xxx"
    //host是AnalyticDB MySQL版集群的连接地址，可以在控制台的集群信息页面获取连接地址。
    host = "127.0.xx.xx"
    //3306是端口号。
    port = 3306
    //database是AnalyticDB MySQL版集群中的数据库名称。
    database = "database_name"
    //数据库连接的超时时间。
    connectTimeout = "10s"
)
func main() {
    //打开数据库连接。
    url := fmt.Sprintf("%s:%s@tcp(%s:%d)/%s?timeout=%s", user, password, host, port, database, connectTimeout)
    db, err := sql.Open("mysql", url)
    if err != nil {
        panic(err.Error())
    }
    //设置可打开连接数的最大值，默认值为0，表示不限制。
    db.SetMaxOpenConns(2)
    //设置最大闲置连接数。
    db.SetMaxIdleConns(1)
    // 设置连接的最大生命周期，默认连接总是可重用。
    // 该设置无法保证连接在连接池中完整存在一小时。
    // 该设置项不是空闲超时时间，即连接会在第一次创建后一小时过期。
    // 理论上，连接的最大生命周期越短，从0开始创建连接的频率就会越高。
    db.SetConnMaxLifetime(time.Hour)
    // defer the close till after the main function has finished
    // executing
    defer db.Close()
    rows, err := db.Query("show tables")
    if err != nil {
        panic(err.Error())
    }
    for rows.Next() {
        var tableName string
        err := rows.Scan(&tableName)
        if err != nil {
            panic(err.Error())
        }
        fmt.Println(tableName)
    }
}
```

开启服务端的Prepared Statement

```
package main
import (
    "database/sql"
    "fmt"
    _ "github.com/go-sql-driver/mysql"
    "time"
)
const (
   //user是AnalyticDB MySQL版集群中的用户账号：高权限账号或者普通账号。
    user = "adb_test"
    //password是AnalyticDB MySQL版集群中用户账号对应的密码。
    password = "xxx"
    //host是AnalyticDB MySQL版集群的连接地址，可以在控制台的集群信息页面获取连接地址。
    host = "127.0.xx.xx"
    //3306是端口号。
    port = 3306
    //database是AnalyticDB MySQL版集群中的数据库名称。
    database = "database_name"
    //数据库连接的超时时间。
    connectTimeout = "10s"
)
func main() {
    // open the database connection
    url := fmt.Sprintf("%s:%s@tcp(%s:%d)/%s?timeout=%s", user, password, host, port, database, connectTimeout)
    db, err := sql.Open("mysql", url)
    if err != nil {
        panic(err.Error())
    }
    // 设置最大打开的连接数，默认值为0，表示不限制。
    db.SetMaxOpenConns(2)
    // 设置最大闲置的连接数
    db.SetMaxIdleConns(1)
    // 设置连接的最大生命周期，默认连接总是可重用。
    // 该设置无法保证连接在连接池中完整存在一小时。连接可能会因为某些原因无法使用而自动关闭。
    // 该设置项不是空闲超时时间，即连接会在第一次创建后一小时过期，而不是空闲后一小时过期。
    // 理论上，连接的最大生命周期越短，从0开始创建连接的频率就会越高。
    db.SetConnMaxLifetime(time.Hour)
    defer db.Close()
    stmt, err := db.Prepare("select * from student where id = ?")
    if err != nil {
        panic(err.Error())
    }
    defer stmt.Close()
    rows, err := stmt.Query(9)
    if err != nil {
        panic(err.Error())
    }
    defer rows.Close()
    for rows.Next() {
        var id string
        var name string
        var unit string
        err := rows.Scan(&id, &name, &unit)
        if err != nil {
            panic(err.Error())
        }
        fmt.Println(fmt.Sprintf("%s, %s, %s", id, name, unit))
    }
}
```

开启客户端的PrepareStatement

在Go MySQL Driver中开启PrepareStatement时，需要配置参数 `interpolateParams=true` 开启客户端的PrepareStatement，如下所示。

> **注意** db.Prepare和stmt.Query无法感知参数 `interpolateParams=true`，必须使用db.Query才可开启客户端的PrepareStatement。

```
package main
import (
    "database/sql"
    "fmt"
    _ "github.com/go-sql-driver/mysql"
    "time"
)
const (
    //user是AnalyticDB MySQL版集群中的用户账号：高权限账号或者普通账号。
    user = "adb_test"
    //password是AnalyticDB MySQL版集群中用户账号对应的密码。
    password = "xxx"
    //host是AnalyticDB MySQL版集群的连接地址，可以在控制台的集群信息页面获取连接地址。
    host = "127.0.xx.xx"
    //3306是端口号。
    port = 3306
    //database是AnalyticDB MySQL版集群中的数据库名称。
    database = "database_name"
    //数据库连接的超时时间。
    connectTimeout = "10s"
)
func main() {
    //打开数据库连接。
    url := fmt.Sprintf("%s:%s@tcp(%s:%d)/%s?timeout=%s&interpolateParams=true", user, password, host, port, database, connectTimeout)
    db, err := sql.Open("mysql", url)
    if err != nil {
        panic(err.Error())
    }
    //设置最大打开的连接数，默认值为0，表示不限制。
    db.SetMaxOpenConns(2)
    //设置最大闲置的连接数。
    db.SetMaxIdleConns(1)
    //设置连接的最大生命周期，默认是连接总是可重用。
    //该设置无法保证连接在连接池中完整存在一小时。连接可能会因为某些原因无法使用而自动关闭。
    //该设置项不是空闲超时时间，即连接会在第一次创建后一小时过期，而不是空闲后一小时过期。
    //理论上，连接的最大生命周期越短，从0开始创建连接的频率就会越高。
    db.SetConnMaxLifetime(time.Hour)
    defer db.Close()
    rows, err := db.Query("select * from student where id = ?", 9)
    if err != nil {
        panic(err.Error())
    }
    defer rows.Close()
    for rows.Next() {
        var id string
        var name string
        var unit string
        err := rows.Scan(&id, &name, &unit)
        if err != nil {
            panic(err.Error())
        }
        fmt.Println(fmt.Sprintf("%s, %s, %s", id, name, unit))
    }
}
```

5.客户端连接AnalyticDB MySQL

5.1. DBeaver

本文介绍如何通过DBeaver连接AnalyticDB for MySQL集群。

背景信息

DBeaver是一款免费、开源（GPL）的专门为开发人员和数据库管理员提供的通用数据库工具。DBeaver支持MySQL、PostgreSQL、Oracle、DB2、MSSQL、Sybase以及其他兼容JDBC的数据库。您可以通过DBeaver的图形界面查看数据库结构、执行SQL查询和脚本、浏览和导出数据、处理BLOB/CLOB数据以及修改数据库结构等。

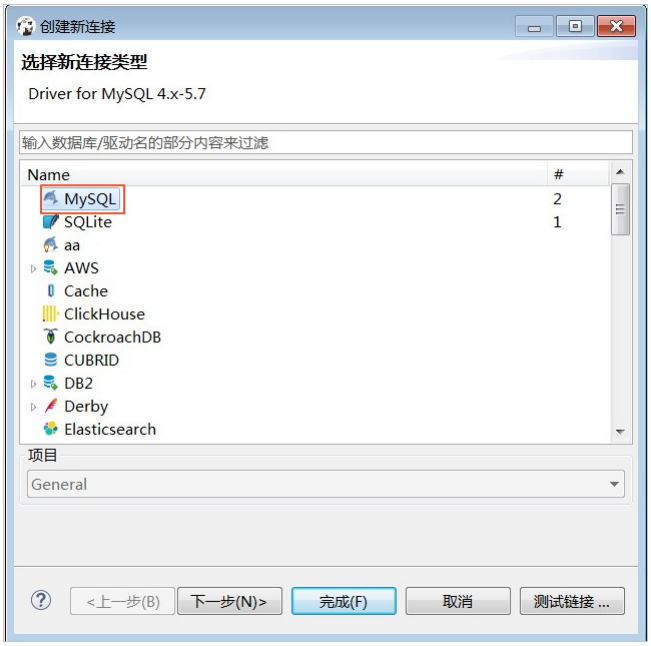
准备工作

开始使用DBeaver前，您需要完成以下准备工作。

- 安装DBeaver软件。
- 安装MySQL JDBC驱动。
- 将安装了DBeaver软件的设备IP添加到AnalyticDB for MySQL集群白名单中。
- 如果您需要通过外网连接AnalyticDB for MySQL集群，请先申请外网地址。

操作步骤

1. 打开**DBeaver**，在菜单栏单击**数据库** > **新建连接**。
2. 在**创建新连接**页面，连接类型选择**MySQL**，单击**下一步**。



3. 在**创建新连接**页面，进行参数配置。

参数	说明
服务器地址	AnalyticDB for MySQL集群的连接地址，通过控制台**集群信息**页面，查看连接信息。
端口	3306。
数据库	AnalyticDB for MySQL集群中数据库的名字。
用户名	AnalyticDB for MySQL集群中创建的账号： ○ 高权限账号。 ○ 普通账号。
密码	账号对应的密码。

4. 完成上述参数配置后，单击**测试连接**测试连通性，测试成功后单击**完成**连接至集群。

5.2. DBVisualizer

本文介绍如何通过DBVisualizer连接AnalyticDB for MySQL集群。

准备工作

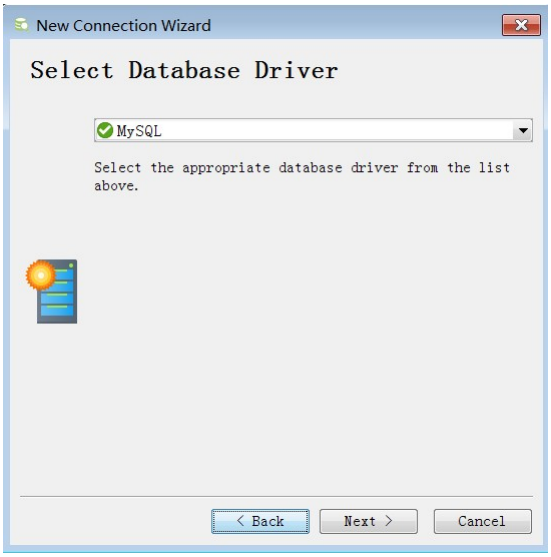
开始使用DBVisualizer前，您需要完成以下准备工作。

- 安装MySQL JDBC驱动。
- 安装DBVisualizer。
- 将安装了DBVisualizer软件的设备IP添加到AnalyticDB for MySQL集群白名单中。
- 如果您需要通过外网连接AnalyticDB for MySQL集群，请先申请外网地址。

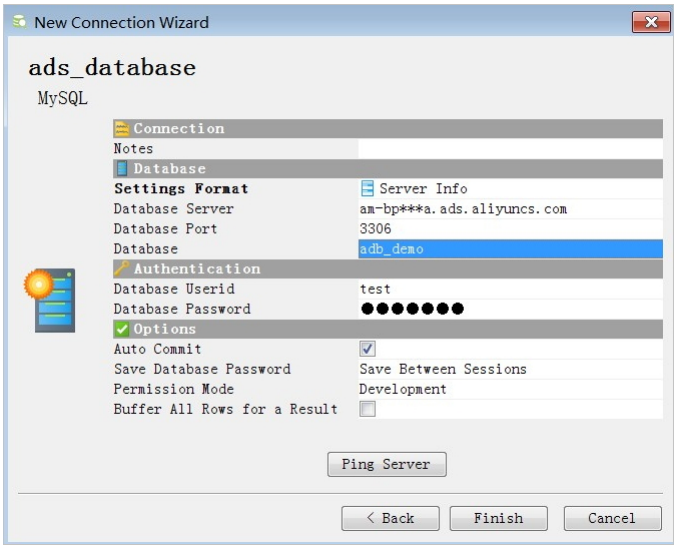
操作步骤

1. 打开DBVisualizer，在菜单栏单击**Tools > Connection Wizard** ，进入**New Connection Wizard**页面，您需要为连接输入一个名字，便于后续管理。

2. 单击**Next**，选择**MySQL**作为Database Driver。



3. 单击**Next**，配置连接参数。



参数	说明
Notes	备注信息

参数	说明
Database Server	AnalyticDB for MySQL集群的连接地址，通过控制台**集群信息**页面，查看连接信息。
Database Port	3306。
Database	AnalyticDB for MySQL集群中数据库的名字。
Database Userid	AnalyticDB for MySQL集群中创建的账号： ○ 高权限账号。 ○ 普通账号。
Database Password	账号对应的密码。

4. 完成上述参数配置后，单击**Ping Server**测试连通性，测试通过后，单击**Finish**。

   成功连接AnalyticDB for MySQL后，您就可以通过DBVisualizer进行数据管理。

5.3. Navicat

Navicat是一套快速、可靠且价格相宜的数据库管理工具，专为简化数据库的管理及降低系统管理成本而设。Navicat提供图形化用户界面，您可以简单、方便地创建本机到AnalyticDB MySQL版集群的远程连接，然后使用Navicat进行数据管理。

前提条件

开始使用Navicat前，您需要完成以下准备工作：

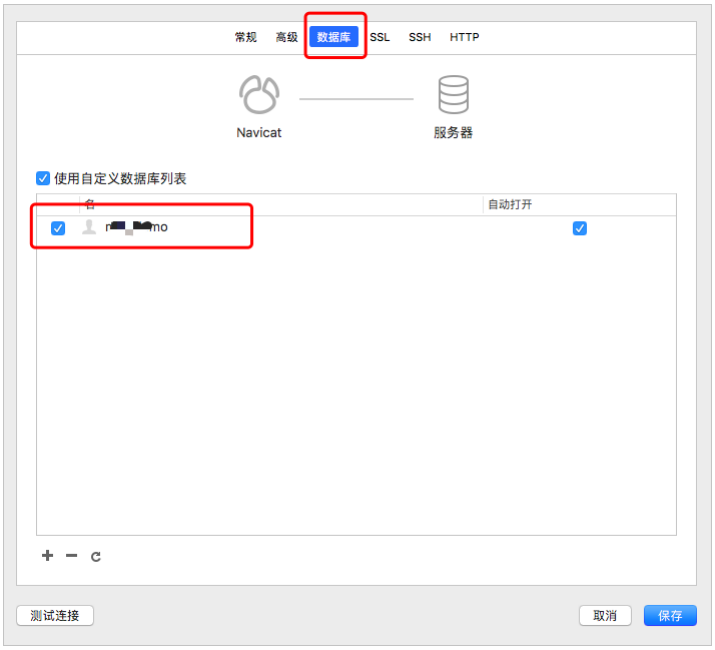
- 了解Navicat与AnalyticDB MySQL版之间的兼容性，请参见兼容性概览。
- 安装Navicat软件，Navicat Premium和Navicat for MySQL推荐15.0.10及以上版本，本文将以Navicat for MySQL软件为例说明。
- 将安装了Navicat软件的设备IP添加到AnalyticDB MySQL版集群白名单中。
- 如果您需要通过外网连接AnalyticDB MySQL版集群，请先申请外网地址。

操作步骤

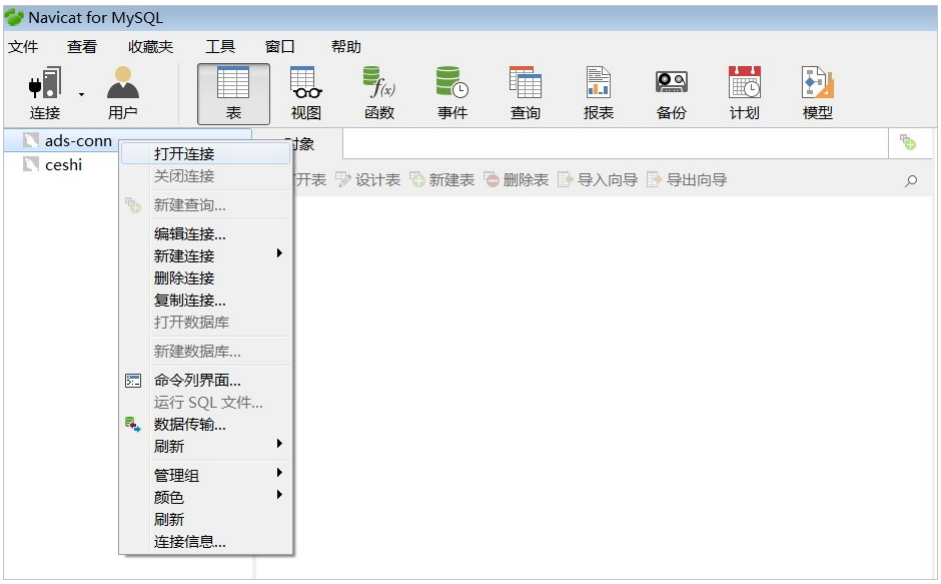
1. 打开Navicat for MySQL，单击**文件 > 新建连接 > MySQL**，在**新建连接**页面，进行参数配置。

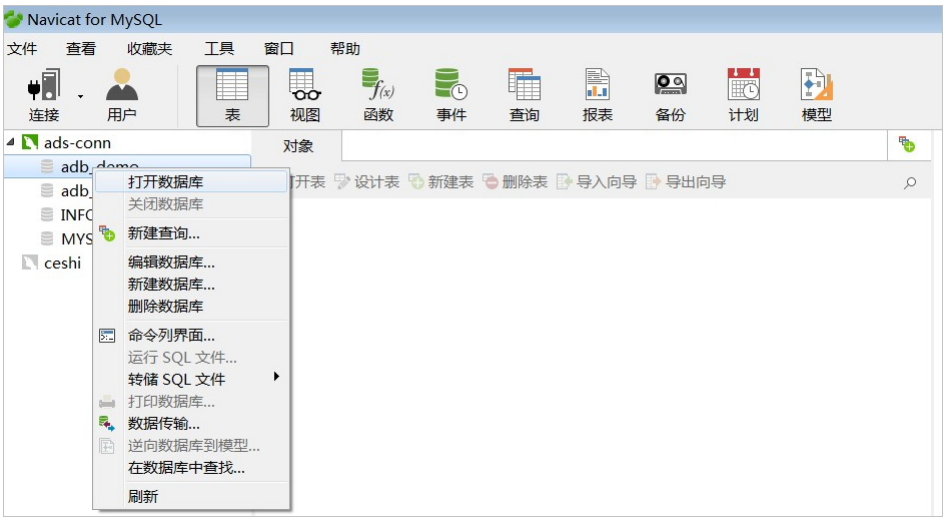
参数	说明
连接名	为连接设置一个名字，便于后续管理。
主机名或IP地址	AnalyticDB MySQL版集群的连接地址，通过控制台**集群信息**页面，查看连接信息。
端口	3306
用户名	AnalyticDB MySQL版集群中创建的账号： ○ 高权限账号。 ○ 普通账号。
密码	账号对应的密码。

如果您的操作系统是macOS，配置完连接信息后，需要加上数据库名字。



2. 单击**连接测试**测试连通性，测试成功后单击**确定**。至此已成功建立集群连接，但连接处于关闭状态需要您手动打开连接。
3. 右键单击**连接名 > 打开连接**，然后右键单击**数据库名**，打开某个数据库连接，接下来您就可以利用Navicat for MySQL进行数据管理。





5.4. SQL WorkBench/J

SQL Workbench/J是一个独立于DBMS，跨平台的SQL查询分析工具，您可以通过SQL WorkBench/J连接AnalyticDB for MySQL集群。

前提条件

开始使用SQL WorkBench/J前，您需要完成以下准备工作。

- 了解SQL WorkBench/J与AnalyticDB for MySQL之间的兼容性，请参见兼容性概览。
- 安装MySQL JDBC驱动。
- 安装SQL WorkBench/J。将安装了SQL WorkBench/J软件的设备IP添加到AnalyticDB for MySQL集群白名单中。
- 如果您需要通过外网连接AnalyticDB for MySQL集群，请先申请外网地址。

注意事项

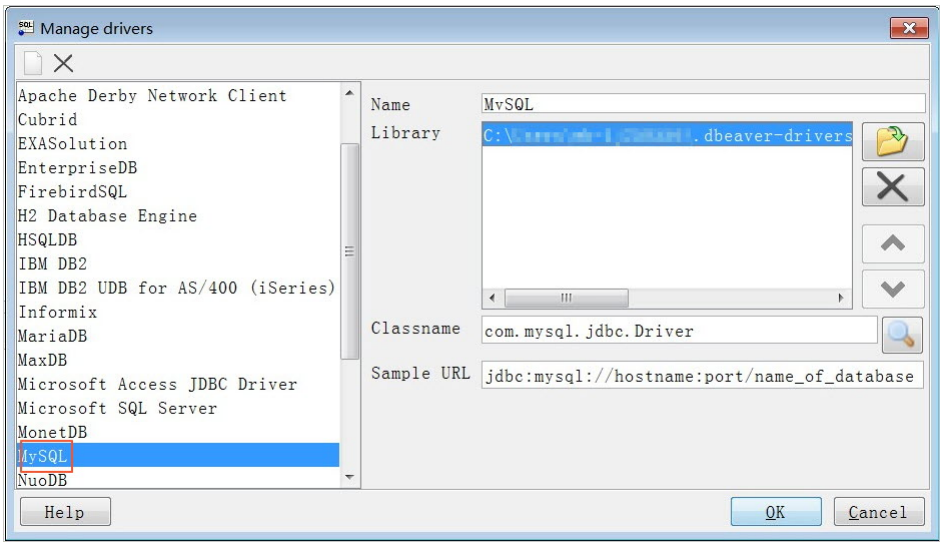
如果需要在SQL WorkBench/J中开启PrepareStatement，请参见不同编程语言中如何开启客户端的PrepareStatement。

操作步骤

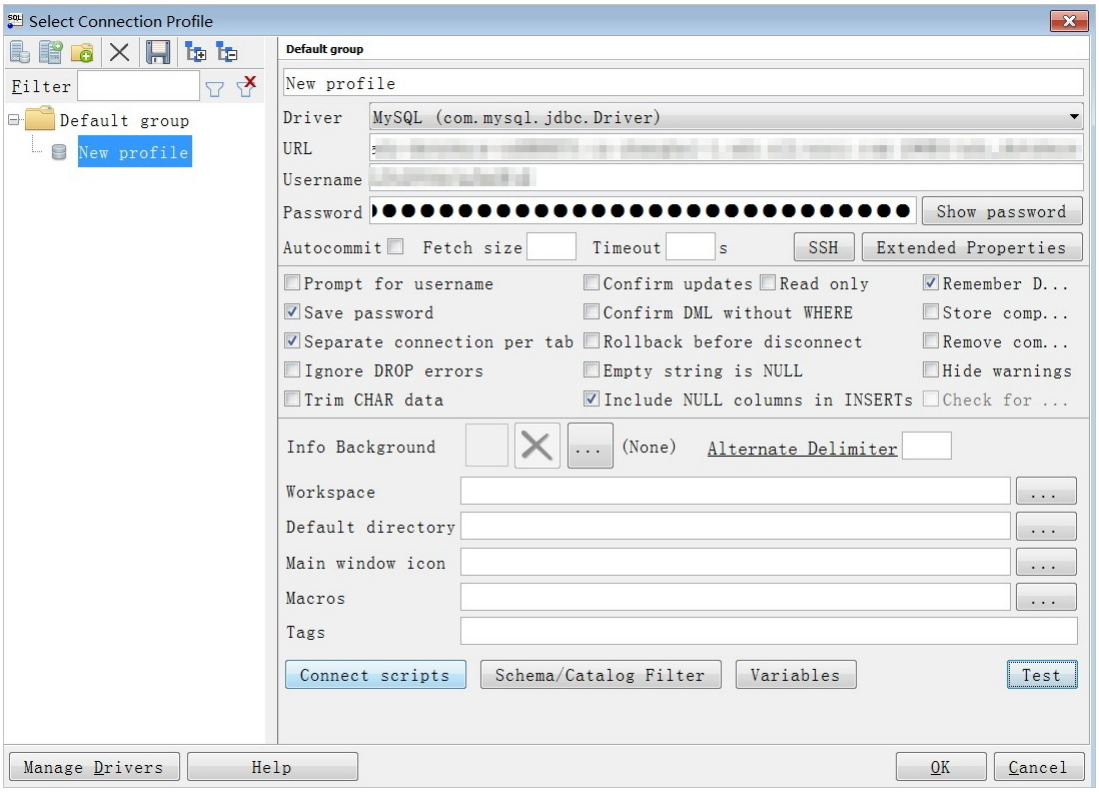
1. 打开SQL WorkBench/J，单击**File > Manage Drivers**。

   > **说明** 首次使用SQL WorkBench/J时，需要添加JDBC驱动和jar包，后续使用SQL WorkBench/J时可直接从步骤三开始。

2. 在**Manage drivers**页面，选择驱动为**MySQL**，添加驱动jar包，单击**OK**。



3. 单击**File > Connect window**，在**Select Connection Profile**页面进行参数配置。



参数	说明
New profile	为连接设置一个名字，便于后续管理。
Driver	选择**MySQL**
URL	AnalyticDB for MySQL集群的连接地址，格式为：`jdbc:mysql://hostname:port/name_of_database`。 ○ hostname：集群的外网地址或者VPC地址。 ○ port：3306。 ○ name_of_database：数据库名字，可选项。
Username	AnalyticDB for MySQL集群中创建的账号： ○ 高权限账号。 ○ 普通账号。
Password	账号对应的密码。

4. 完成上述参数配置后，单击**Test**测试连通性，测试通过后单击**OK**，连接至AnalyticDB for MySQL集群。

6.不同编程语言中如何开启客户端的PrepareStatement

本文将介绍如何在不同编程语言中开启客户端的PrepareStatement。

大多数数据库中，依靠服务器端预处理语句可以提高数据库性能。AnalyticDB for MySQL数据库自身具备强大的查询计算能力和计划缓存功能，无需依靠服务器端预处理语句获得大部分性能优势。

AnalyticDB for MySQL数据库目前不支持服务器端预编译协议，大部分开发语言中支持通过配置开启客户端预编译（PrepareStatement），也称之为客户端准备语句仿真或参数插值。

MySQL Connector/J (JDBC) driver

在MySQL Connector/J (JDBC) driver中开启PrepareStatement时，需要配置`useServerPrepStmts=false`参数，详情请参见Configuration Properties for Connector/J。

> **说明** 无需配置`useCursorFetch=true`参数，否则将覆盖`useServerPrepStmts=false`参数，无法开启PrepareStatement。

MariaDB Connector/J

在MariaDB Connector/J中开启PrepareStatement时，需要配置`useServerPrepStmts=false`参数，详情请参见About MariaDB Connector/J。

Go MySQL driver

在Go MySQL driver中开启PrepareStatement时，需要配置`interpolateParams=true`参数，详情请参见Go-MySQL-Driver。

PDO

在PDO中使用PrepareStatement时，需要配置`PDO::ATTR_EMULATE_PREPARES=TRUE`参数，详情请参见setAttribute。