阿里云 MaxCompute

工具及下载

文档版本: 20190701

为了无法计算的价值 | [-] 阿里云

<u>法律声明</u>

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。

- 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。
- 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
•	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。	禁止: 重置操作将丢失用户配置数据。
A	该类警示信息可能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,恢复业务所需 时间约10分钟。
Ê	用于补充说明、最佳实践、窍门等,不 是用户必须了解的内容。	道 说明: 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	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。	单击 确定。
courier 字体	命令。	执行 cd /d C:/windows 命令,进 入Windows系统文件夹。
##	表示参数、变量。	bae log listinstanceid Instance_ID
[]或者[a b]	表示可选项,至多选择一个。	ipconfig [-all -t]
	表示必选项,至多选择一个。	<pre>swich {stand slave}</pre>

目录

法律声明I
通用约定
1 安白兴 1
2 MaxCompute Studio4
2.1 认识Studio
2.2 项目空间连接管理9
2.3 工具安装与版本信息14
2.3.1 IntelliJ IDEA安装步骤14
2.3.2 IntelliJ IDEA插件安装步骤19
2.3.3 查看和更新版本
2.4 管理数据和资源
2.4.1 浏览表及 UDF29
2.4.2 导入导出数据34
2.4.3 可视化创建、修改和删除表42
2.5 开发SQL程序46
2.5.1 创建MaxCompute Script Module46
2.5.2 编写SQL脚本53
2.5.3 提交SQL脚本58
2.6 开发Java程序61
2.6.1 创建MaxCompute Java Module61
2.6.2 开发和调试UDF63
2.6.3 开发MapReduce68
2.6.4 非结构化开发72
2.6.5 开发Graph76
2.6.6 打包、上传和注册78
2.7 开发Python程序81
2.7.1 Python开发使用须知81
2.7.2 开发Python UDF84
2.7.3 开发PyODPS脚本89
2.8 管理MaxCompute作业90
2.8.1 作业浏览90
2.8.2 作业实例详情94
2.9 工具集成106
2.9.1 与MaxCompute客户端集成106
2.10 配置选项
2.10.1 配置 MaxCompute Studio108
2.11 Studio视频介绍112
2.12 常见问题(FAQ)112
2.12.1 如何通过 Studio 开发 UDF112
2.12.2 如何通过 Studio 管理 MaxCompute 元数据

3 Eclipse开发插件	
3.1 安装Eclipse插件	
3.2 创建MaxCompute工程	
3.3 MapReduce开发插件介绍	
3.4 UDF开发插件介绍	
3.5 Graph开发插件介绍	
4 相关下载	146

1客户端

本文为您介绍如何使用客户端命令行工具使用MaxCompute服务的基础功能。

在使用MaxCompute客户端前,请首先安装并配置客户端。



- ・请不要依赖客户端的输出格式来做任何的解析工作。客户端的输出格式不承诺向前兼容,不同 版本间的客户端命令格式及行为有差异。
- ·关于客户端的基本命令介绍,请参见基本命令。
- ·新版客户端:点击此处即可下载新版客户端。关于不同环境下客户端的安装配置信息请参见安装并配置客户端。
- ·客户端从0.28.0版开始支持JDK 1.9,这之前的版本只能用JDK 1.8。
- ・客户端从0.27.0版本开始支持MaxCompute 2.0新数据类型。

安装并配置好客户端后,您可借助命令行工具进行以下操作。

获取帮助

若想显示客户端的帮助信息,命令格式如下所示。

odps@ >./bin/odpscmd -h;

您也可以在交互模式下键入h;或help;(不区分大小写)。

客户端还提供了help [keyword];命令,可获取到与关键字有关的命令提示。例如:输入help table;可以得到与table操作相关的命令提示,如下所示。

启动参数

在启动时,您可指定一系列参数,如下所示。

-r set retry times
-f <"file_path;"> execute command in file
-e <"command;[command;]..."> execute command, include sql command
-C will display job counters

以-f参数为例,操作如下:

1. 准备本地脚本文件script.txt,假设存放在D盘,文件内容如下所示。

```
DROP TABLE IF EXISTS test_table_mj;
CREATE TABLE test_table_mj (id string, name string);
DROP TABLE test_table_mj;
```

2. 打开您的CMD命令行工具,进入odpscmd客户端所在路径,运行以下命令。

```
odpscmd\bin>odpscmd -f D:/script.txt;
```

3. 您也可以使用上述命令批量建表或删表。

交互模式

直接运行客户端即可进入到交互模式,如下所示。

```
[admin: ~]$odpscmd
Aliyun ODPS Command Line Tool
Version 1.0
@Copyright 2012 Alibaba Cloud Computing Co., Ltd. All rights reserved.
odps@ odps> INSERT OVERWRITE TABLE DUAL SELECT * FROM DUAL;
```

在光标位置输入命令(以分号作为语句的结束标志),回车即可运行。

续跑

- ・ 在用-e或-f 模式运行时,如果有多条语句,想从中间某条语句开始运行,可以指定参数-k,表 示忽略前面的语句,从指定位置的语句开始运行。当指定参数<=0 时,从第一条语句开始执行。
- · 每个以分号分隔的语句被视为一条有效语句,在运行时会打印出当前运行成功或者失败的是第几条语句。

示例

假设文件/tmp/dual.sql中有三条SQL语句,如下所示。

drop table dual; create table dual (dummy string); insert overwrite table dual select count(*) from dual;

如果想忽略前两条语句,直接从第三条语句开始执行,则命令格式如下所示。

odpscmd -k 3 -f dual.sql

获取当前登录用户

若想获取当前登录用户,命令格式如下所示。

whoami;

示例

```
odps@ hiveut>whoami;
Name: odpstest@aliyun.com
End_Point: http://service.odps.aliyun.com/api
Project: lijunsecuritytest
```

通过以上命令,即可获取当前登录用户的云账号、使用的End_Point配置和项目名。

退出

若想退出客户端,命令格式如下所示。

odps@ > quit;

您也可输入以下命令退出客户端。

odps@ > q;

2 MaxCompute Studio

2.1 认识Studio

MaxCompute Studio是阿里云MaxCompute平台提供的安装在开发者客户端的大数据集成开发环境工具,是一套基于流行的集成开发平台*IntelliJ IDEA*的开发插件,帮助您便捷、快速地进行数据 开发。本文将为您介绍MaxCompute Studio的功能界面和常用的应用场景。

基本用户界面

MaxCompute Studio是IntelliJ IDEA平台上的一套插件,共享了IntelliJ IDEA的基本开发界面。

MaxCompute Studio在IntelliJ的基础上提供以下功能:

・ SQL编辑器(SQL Editor):提供SQL语法高亮、代码补全、实时错误提示、本地编译、作业 提交等功能。

编译器视图(Compiler View):显示本地编译的提示信息和错误信息,在编辑器中定位代码。

・项目空间浏览器(Project Explorer): 连接MaxCompute项目空间,浏览项目空间表结构、自定义函数、资源文件。

表详情视图(Table Details View):提供表、视图等资源的详情显示和示例数据(Sample Data)。

- · 作业浏览器(Job Explorer): 浏览、搜索MaxCompute的历史作业信息。
 - 作业详情视图(Job Details View):显示作业的运行详细信息,包括执行计划和每个执行 任务的详细信息,Logview工具能够显示的全部信息。
 - 作业输出视图(Job Output View):显示正在运行的作业的输出信息。
 - 作业结果视图(Job Result View):显示SELECT作业的输出结果。
- MaxCompute控制台(MaxCompute Console):集成了MaxCompute客户端,可以输入和 执行MaxCompute客户端命令。

连接MaxCompute项目空间

Studio的大部分功能需要您首先创建项目空间连接。建立项目空间连接后,即可在项目空间浏览 器中查看相关的数据结构和资源信息。Studio会自动为每一个项目空间连接建立一个本地的元数据 备份,以提高对MaxCompute元数据的访问频率和降低延时。 **送** 说明:

- 您需要先指定项目空间链接,才可通过Studio进行编辑SQL脚本、提交作业、查看Job信息、 打开MaxCompute控制台等操作,因此首先创建一个MaxCompute项目空间的连接是非常必要的。
- · MaxCompute项目空间的更多详情请参见项目空间。
- · 在Studio中管理项目空间的更多详情请参见项目空间连接。

管理数据

您可以通过Studio的项目空间浏览器快速浏览项目空间的表结构、自定义函数、资源文件。通过树 形控件,可以列出所有项目空间连接下的数据表、列、分区列、虚拟视图、自定义函数名称、函数 签名、资源文件及类型等,并支持快速定位。

双击某个数据表,即可打开表详情视图,查看数据表的元信息、表结构和示例数据。如果您没有项 目空间的相应权限,Studio会提示对应的错误信息。

Studio集成了MaxCompute Tunnel工具,可以支持本地数据的上传和下载,更多详情请参见导入并导出数据。

编写SQL脚本

您可以在Studio中编写MaxCompute SQL脚本,非常方便。

- 1. 打开Studio, 导航至File > New > Project或者File > New > Module…。
- 2. 创建一个MaxCompute Studio类型的项目或者模块。
- 导航至File > New > MaxCompute Script 或者右击菜单New > MaxCompute Script,即可 创建一个MaxCompute SQL脚本文件。

📕 说明:

创建MaxCompute SQL脚本时,Studio会提示您选择一个关联的MaxCompute项目空间,您也可以通过SQL编辑器上的工具条最右侧的项目空间选取器进行更改,编辑器会根据SQL脚本关联的项目空间对SQL语句自动进行元数据(比如表结构等)的检查并汇报错误,提交运行时也会发送到关联的项目空间执行。更多详情请参见_{编写}SQL脚本。

SQL代码智能提示

Studio提供的SQL编辑器可以根据您写入的代码,智能提示SQL语句的语法错误、类型匹配错误或 者警告等,实时地标注在代码上。如下图所示:



通过代码补全功能,Studio可以根据代码上下文,提示您项目空间名称、表、字段、函数、类型、 代码关键词等,并根据您的选择,自动补全代码。如下图所示:



编译和提交作业

・编译作业

单击SQL编辑器工具条上的

图标,可以对SQL脚本执行本地编译,如果有语法或者语义错

误,编译器窗口会报告错误。



・提交作业

单击SQL编辑器工具条上的图标,会在本地编译之后,把SQL脚本提交

到MaxCompute指定的项目空间排队执行。

查看历史作业

打开作业浏览器,您即可查看指定项目空间上近期执行的作业。

📋 说明:

这个列表只能显示以当前连接使用的用户ID提交的作业。

MaxCompute Job Explorer 🛛 🌞 🗜							
Project: s	ql_optimiz	er 🔻	Days: 2	t)			
		Q (40/47883)	Table 🔻			
Instanceld	Status	Owner	StartTime	EndTime			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	SUCCESS	ODPS	2017	2017			
20170	FAILED	ODPS	2017	2017			

双击其中一个作业,便可查看作业的的详情信息。如下图所示:



如果知道一个任务的Logview URL,可以导航至MaxCompute > Open Logview,打开该任务的详情页面。

开发MapReduce和UDF

Studio还支持MapReduce和Java UDF的开发。

连接MaxCompute客户端

Studio集成了最新版本的MaxCompute_{客户端},您也可以在Studio的<mark>配置页面</mark>中指定本地已经安装好的MaxCompute客户端路径。

您在项目空间浏览器中选定一个项目空间,右键单击菜单选择Open in Console即可打 开MaxCompute控制台窗口。



后续步骤

现在,您已经学习了MaxCompute Studio的功能界面和常用的应用场景,您可以继续学习下一个 教程。在该教程中您将学习如何安装MaxCompute Studio。详情请参见安装IntelliJ IDEA。

2.2 项目空间连接管理

MaxCompute Studio的一个核心功能是浏览MaxCompute项目空间(Project)的资源,包括Table、UDF、Resource等。要想实现这一功能,首先需要创建项目空间连接。

前提条件

由于使用IntelliJ的Tool Windows必须先打开某个IntelliJ project,而配置MaxCompute Project需要进入IntelliJ界面Tool Windows中的MaxCompute Project Explorer,所以在 创建MaxCompute Project连接前,您可以首先添加或者导入一个IntelliJ project。本文将以 在Windows下新增project为例。

运行IntelliJ IDEA后,单击New > Project,选择弹出页面中的MaxCompute Studio,单 击Next。



UNew Project		2%
Java	Additional Libraries and <u>F</u> rameworks:	
Java FX		
👾 Android		
🔮 IntelliJ Platform Plugin		
<i>M</i> Maven	Nothing to show	
📀 Gradle		
G Groovy		
🗐 Griffon		
Kotlin		
Empty Project		
🔥 MaxCompute Java		
🕺 MaxCompute Studio		
	Previous <u>N</u> ext Cancel Hel	Р

填写Project name,单击Finish。

🖳 New Project		X
Project n <u>a</u> me: MyFir	stProject	
Project location: C:\Us	ers\Administrator\IdeaProjects\MyFirstProject	
 More Settings 		
Module na <u>m</u> e:	MyFirstProject	
Content <u>r</u> oot:	C:\Users\Administrator\IdeaProjects\MyFirstProject	
Module file location	: C:\Users\Administrator\IdeaProjects\MyFirstProject	
Project <u>f</u> ormat:	.idea (directory based)	\sim
	Previous <u>F</u> inish Cancel Help	þ

创建MaxCompute项目链接

建议您根据自己的Region配置MaxCompute项目连接,否则会出现无法访问等错误。您可以参考<mark>配置Endpoint</mark>配置自己的Endpoint和Region。

操作步骤:

1. 单击菜单中的View选项,选择Tool Windows。

Project [C:\Users\wb-wd419990\IdeaProjects\N	lyFirstProject] - IntelliJ IDEA	
<u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e <u>R</u> efactor	<u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indo	w MaxCompute <u>H</u> elp
<u>T</u> ool Windows	Project Alt+1	
Recent Files Ctrl+E	★ Favorites Alt+2	
Recently Changed Files Ctrl+Shift+E	Run Alt+4	
Recent Changes Alt+Shift+C	∰ Debug Alt+5	
Quick Switch Scheme Ctrl+`	TODO Alt+6	
Toolbar	Structure Alt+7	
Tool Buttons	Version Control Alt+9	
	The second secon	
✓ Na <u>v</u> igation Bar	Capture Analysis	
Bidi Text Direction		rch Everywhere Double Shift
Enter Presentation Mode	Designer	
Enter Distraction Free Mode	Event Log	to File Ctrl+Shift+N
Enter Full Screen	V Image Lavers	
	V Job Explorer	ent Files Ctrl+E
	<i>m</i> Maven Projects	instice Dev Alter Lines
	🕍 Palette	All Home
	🕍 Palette	n files here to open
	🔥 Project Explorer	p mes nere to open
	Terminal Alt+F12	
	Theme Preview	
	📝 UI Designer	

2. 单击弹出页面中的Project Explorer

3. 单击左上角的+, 选择AddprojectfromaccessId/Key模式, 添加一个MaxCompute Project。



4. 在Add MaxCompute Project对话框中,填入相关配置选项。

🖳 Add MaxCompu	te project				Σ	3	
Connection Set	ting					_	
Properties File:	odps_config.ini	表初始化AK/	Endpoint	等配置项			
AK Account:	AK账号来初始化 K Account	AK。可点击	右侧+号》		+		
* Access Id:							
* Access Key:	连接MaxCom 等配置信息,	npute Proj 必填。可	ject时Al 「手工填:	K/Endp 写,也ī	oint 可通		
* Project Name:	过上述Prope	erties File	:或AK Ao ቛ	count	未初		
* End Point:			<u>n</u> •				
					~		
0			OK	Ca	ncel		
📋 说明:							
・単击dialog左下	角的?即可链接到a	车线文档。					
 ・ 若同步出现招时 	错误。可以在sett	ing标签页函	情延长元	数据同步	到本国	11111111111111111111111111111111111111	时间。
和署会成后 单击0	K 左侧MaxCon	nute Proje	et Fynlo	rer由会	显示M	lavCom	nute
Project的信息 可	以通过鼠标单击查	看该nroiect	中的表	和图 屎	i数D17	B 密源等	pute 信息
	从远之 俄你中田重	Idea Dreieste M					
<u>File Edit View Nav</u>	rigate <u>C</u> ode Analy	<u>ze R</u> efactor	<u>B</u> uild R <u>u</u>	n <u>T</u> ools	VC <u>S</u>	<u>W</u> indow	MaxCompute
MyFirstProject	<u> </u>						
Project Explorer				\$-⊪			
ê + - ↓ €	: 😤 🙆 🗊 ?						
to VA http://service	odps.aliyun.com/ap	i					
> Table	rojectwann s & Views						
> 🔂 Funct	ions)					
5							

查看/修改MaxCompute项目链接

在Project Explorer中,对需要修改的MaxCompupte项目右键选择Show|Modify project properties。

IJ	MyFirst	Project	[C:\	A-1408	19890 (Me	aProjects\I	MyFirst	Projec	t] - Intel	IJ IDE	А
<u>F</u> ile	e <u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indow
V	MyFire	stProje	$\left \mathbf{ct} ight angle$								
rer	Projec	t Explo	rer						* - ŀ-		
oldxi	+ -	ſĵ	₹		D ?						
tg	~ 🔥 h	ttp://se	ervice.odps	aliyun.c	om/api						
Proj	ໂ	P MyF > □	irstProjectV Tables & Vi	Vdh ew: Ad	d project			>			
N.		> 🚡	Functions	Rei	move proj	ect					
Б		s ^a	Resources	Sho	ow Modify	/ project pi	ropertie	es			
- Del				Ret	fresh meta	3					
Ě				Ор	en in Con	sole					
9				Ne	w sql edit	or					
¥1				Cre	eate a new	/ table					
📑 1: Project											

在弹出框中可以查看或修改该MaxCompute project的Connection和Setting。

后续操作

现在,您已经学习了如何新建、管理项目空间连接,您可以继续学习下一个教程。在该教程中您将 学习如何进行元数据查询、清理数据、上传下载数据等操作,来管理数据和资源。详情请参见 管理 数据和资源。

2.3 工具安装与版本信息

2.3.1 IntelliJ IDEA安装步骤

本文将为您介绍如何安装MaxCompute Studio的基础平台IntelliJ IDEA。

操作步骤

 单击此链接,根据操作系统(Windows、macOS、Linux)下载对应IntelliJ IDEA版本。此 处我们以Windows操作系统安装为例,下载IntelliJ IDEA 14.1.4以上版本。
 支持Ultimate版本或免费的Community版本,PyCharm也支持。

2. 下载完成后,双击安装程序,进入安装界面,单击Next。



3. 指定安装目录后,单击Next。

IntelliJ IDEA Co		
	ommunity Edition Setup	
	Choose Install Location	
JIJ	Change the folder in which to install Intelli	IDEA Community
	Edition.	IDEA Community
Setup will install Ir folder, click Brows	ntelliJ IDEA Community Edition in the following folder. T e and select another folder. Click Next to continue.	o install in a different
Destination Fold	ler	
Destination Fold	ler IetBrains\IntelliJ IDEA Community Edition 2018.1.4	Browse
Destination Fold rogram Files\2 Space required: 9	ler letBrains\IntelliJ IDEA Community Edition 2018.1.4 26.9 MB	Browse
Destination Fold rogram Files\J Space required: 9 Space available: 1	ler letBrains\IntelliJ IDEA Community Edition 2018.1.4 26.9 MB 121.8 GB	Browse
Destination Fold rogram Files\ Space required: 9 Space available: 1	ler letBrains\IntelliJ IDEA Community Edition 2018.1.4 26.9 MB 121.8 GB	Browse

4. 根据本地操作系统的版本选择需要安装32位或者64位的IntelliJ IDEA。

如何查看本地操作系统的版本:

a) 打开windows资源管理器, 右击计算机, 选择 属性。

									×
	计算机	•			•	· * 7	搜索 计算机		٩
组织 ▼ 系统)	属性	卸載或更改程序	映射网络驱动器	打开控制面板					0
 ▲ ☆ 收藏夹 ↓ 下载 ■ 桌面 20 最近访问的 	位置	▲ 硬盘 (2) OSDisk	: (C:) 3 可用 , 共 97.6 GB		DataDisk (D:) 120 GB 可用,共 14	40 GB			
▷ 💪 WPS网盘			盘 里手机也能看到						
 ↓ □ Git ▶ □ Git ▶ □ QIL ▶ □ QLL ▶ □ QLL ▶ □ 文档 ▶ □ 音乐 									
▲ 1型 计算机 ▷ 🏭 OSDis	折叠(A)							
) 💼 DataD 🌍	管理(G 在新窗) 口中打开(E)							
▷ 🗣 网络	映射网 断开网	络驱动器(N) 络驱动器(C)							
	添加— 删除(D	个网络位置(L)	Core(TM) i5-6	内存: 8.00 GB					
[重命名 属性(R	(M)) opyright © 2018 Gid	Dealization & Conte	ent Service All Rig	hts Reserved				

b) 在弹出界面中查看操作系统位数。

	統和安全 🕨 系統	✓ ⁴ ₇	搜索控制面板
控制面板主页 设备管理器 过程设置 近程设置 系统保护 家統设置	查看有关计算机的基 Windows 版本 Windows 7 企业版 版权所有 © 2009 Mi Service Pack 1	本信息 crosoft Corporation。保留所有权利。	
	 系统 分级: 处理器: 安装内存(RAM): 系统类型: 笔和触摸: 计算机名称 域和工作组页 	 系统分级不可用 Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz 8.00 GB (7.88 GB 可用) 64 位操作系统 没有可用于此显示器的笔或触控输入 	3.20 GHz
另请参阅 操作中心 Windows Update 性能信息和工具	计算机名: 计算机名: 计算机全名: 计算机描述: 工作组:	MININT-AM7EIIM MININT-AM7EIIM 1	●更改设置

5. 选择相应的系统类型,单击Next。如下图所示:

IntelliJ IDEA Community Edition Setup
Installation Options Configure your IntelliJ IDEA Community Edition installation
Create Desktop Shortcut 32-bit launcher Create Associations
.java .groovy .kt
< Back Next > Cancel

6. 单击Install,开始安装。如下图所示:

IntelliJ IDEA C	ommunity	Edition Setup			
IJ		Choose Start M Choose a Start N Edition shortcuts	lenu Folder 1enu folder for t h 5.	ne Intellij IDE/	A Community
Select the Start I can also enter a	Menu folder name to cre	in which you wou ate a new folder.	ld like to create t	he program's s	shortcuts. You
JetBrains					
Accessories Administrative T Cisco Eclipse Games Git Java Maintenance Microsoft Office MySQL-Front Notepad++ Oxygen XML Edi	ools 2013	1-bit)			
Oxygen XML Edi	itor 18.1 (64	+-DIT)			*
					_
			< Back	Install	Cancel
intelliJ IDEA Co	ommunity	Edition Setup	< Back	Install	Cancel
intelliJ IDEA Co	ommunity I	Edition Setup nstalling Please wait while installed.	< Back	Install	Cancel
intelliJ IDEA Co	ommunity I gle_off_focu	Edition Setup Installing Please wait while installed. used_holo_dark.9	< Back	Install	Cancel
intelliJ IDEA Co	ommunity I gle_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co IJ Extract: btn_togg Show details	ommunity I gle_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity l gle_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity l gle_off_focu	Edition Setup Installing Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity le_off_focu	Edition Setup Installing Please wait while installed. Ised_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity le_off_focu	Edition Setup Installing Please wait while installed. Ised_holo_dark.9	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity J gle_off_focu	Edition Setup Installing Please wait while installed. Ised_holo_dark.9.	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity Je_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9	< Back	Install	Cancel
IntelliJ IDEA Co	ommunity gle_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel
ntelliJ IDEA Co	ommunity gle_off_focu	Edition Setup nstalling Please wait while installed. used_holo_dark.9.	< Back	Install	Cancel

7. 安装完成后,单击Finish。



后续操作

现在,您已经学习了如何安装IntelliJ IDEA,可以继续学习如何安装MaxCompute Studio插件。详情请参见安装 Studio 插件。

2.3.2 IntelliJ IDEA插件安装步骤

MaxCompute Studio是IntelliJ IDEA的插件,本文为您介绍如何安装MaxCompute Studio。

环境要求

IntelliJ IDEA支持在Windows, Mac, Linux操作系统上安装,硬件及系统环境要求请参见 Requirements for IntelliJ IDEA。基于IntelliJ IDEA平台的MaxCompute Studio也可以安装在这 些操作系统的客户端上。

MaxCompute Studio对用户环境有以下要求:

- · Windows, Mac OS, 或者Linux系统客户端。
- ・安装IntelliJ IDEA v18.2.4以上版本 (支持Ultimate版本、PyCharm版本和免费的 Community 版本)。
- ・已安装JRE 1.8 (最新的IntelliJ IDEA版本捆绑了JRE 1.8)。
- · 已安装JDK 1.8 (可选:如果需要开发和调试Java UDF,则需安装 JDK)。

📙 说明:

客户端从0.28.0版开始支持JDK 1.9,在这之前的版本只支持JDK 1.8。

安装方式

MaxCompute Studio是IntelliJ IDEA的插件,有以下两种安装方式:

- ·通过插件库在线安装(推荐)。
- ・通过本地文件安装。

在线安装(推荐)

MaxCompute Studio插件已对全部公网用户开放,您可以通过IntelliJ官方插件库安装。

操作步骤

- 1. 在IntelliJ IDEA中打开插件配置页面 (Windows/Linux用户导航至 File > Settings > Plugins, Mac用户导航至IntelliJ IDEA > Preferences > Plugins)。
- 2. 单击Browse repositories…按钮,然后搜索MaxCompute Studio。
- 3. 找到MaxCompute Studio插件页面,单击绿色Install按钮进行安装。
- 4. 确认安装后,重新启动IntelliJ IDEA,完成安装。

Settings			Browse Repositories	Browse Repositories		
*	Plugins		🔍 🖉 Category: All 🗸			
Appearance & Behavior	Q* Show: All plugins ~		Sort by: name FRAMEWORK INTEGRATION			
Keymap Editor	Sort by:	name 🔻	env files support <u>962,252</u>			
Plugins	C And Support		ignore 8,364,746 ★★★★☆ ♥ @ Work NUTCONT			
Version Control Build, Execution, Deployment	G Bytecode Viewer G Copyright	Ø	VCS INTEGRAII 4 months ago Updated 2018/8/25 v0.7 Image: Strate			
Languages & Frameworks Tools	Coverage		ANDROID one month ago I01 Header 3,341	a for		
MaxCompute Studio	CVS Integration	\square	✓ FORMATTING 7 months ago PHP, JavaScript, Python, Go and ☑ 1024 Tools 5,592 ★★★★★★	Ruby		
	EditorConfig		CODE TOOLS 3 months ago GODE TOOLS 3 months ago 360 FireLine Plugin 35 236	ick o		
	G GitHub	2	CODE TOOLS one month ago	on o		
	Gradle Groovy	2	A Powerful Java Stri 7,780 A Powerful Java Stri 7,780 Change Notes OBFUSCATION one year ago A powerful power	nv*		
	📲 I18n for Java		A8Translate 15,184 ******* files support a sup	from		
	 IntelliLang Java Bytecode Decompiler 	\square	ABTranslate 13,775			
	Java Stream Debugger		Addi Fayzakhmanov http://addi.fayzakhmanov http://addi.fayzakhmanov			
	Chark or unchark a plugin to enable or disable it		ABACUS Plugin 905 ***** 98.8 K			
	Install JetBrains plugin Browse repositories	Ins	Ins			

本地安装

MaxCompute Studio也可以在本地环境中进行安装。

操作步骤

1. 进入MaxCompute Studio 插件页面下载插件包。

- 2. 运行IntelliJ IDEA。
 - ·如果是第一次运行,会出现欢迎界面,单击欢迎界面中的configure(配置),选中弹出菜 单中的Plugins(插件),如下图所示:



 ・如果不是第一次运行,可以依次单击菜单File > Settings > Plugins进入相同的界面,如下 图所示:



U Settings			
Q_	Plugins		
> Appearance & Behavior	Q,*	Show: All plugins $ \smallsetminus $	
Keymap		Sort by: name 🔻	Android Support
> Editor	🖷 Android Support		
Plugins	📑 Ant Support		and Android Studio.
> Version Control	📲 Bytecode Viewer		
> Build, Execution, Deployment	🗲 Copyright	\checkmark	
> Languages & Frameworks	🗲 Coverage	\checkmark	
> 100IS	📲 CVS Integration	\checkmark	
Vim Emulation	📲 Eclipse Integration	\checkmark	
	📲 EditorConfig		
	📲 Git Integration		
	📲 GitHub		
	🖷 Gradle		
	📲 Groovy		
	📲 I18n for Java		
	📲 IdeaVim		
	📑 IntelliLang	\checkmark	
	📑 Java Bytecode Decompiler	\checkmark	
	🕞 Java Stream Debugger	\checkmark	
	🔍 lavaFX		
	Check or uncheck a plugin to enable	or disable it.	
	Install JetBrains plugin	rowse repositories Instal	ll plugin from <u>d</u> isk
			OK Carrel
V			OK Cancel Apply

3. 在插件页面,	单击Install plugin from disk…	(从本地磁盘安装插件),	如下图所示:
-----------	-----------------------------	--------------	--------

Plugins		
Q,▼ Show:	All plugins $ \smallsetminus $	
	Sort by: name 🔻	Android Support
🖷 Android Support		
📲 Ant Support		and Android Studio.
📲 Bytecode Viewer		
Copyright	\checkmark	
Coverage	\checkmark	
CVS Integration		
🗲 Eclipse Integration		
📲 EditorConfig		
📲 Git Integration		
📲 GitHub		
📲 Gradle		
📲 Groovy		
📲 I18n for Java	\checkmark	
📲 IdeaVim	\checkmark	
📲 IntelliLang		
📲 Java Bytecode Decompiler	\checkmark	
📲 Java Stream Debugger	\checkmark	
🖳 JavaFX		
Check or uncheck a plugin to enable or disable	e it.	
Install <u>J</u> etBrains plugin <u>B</u> rowse re	epositories Insta	ll plugin from <u>d</u> isk
		OK Cancel Apply

4. 在弹出窗口中,通过单击目录名称前的灰色图标进行导航,找到插件文件并选中,单击OK。

Le Choose Plugin File
JAR and ZIP archives are accepted
🕋 🖾 🐂 🐂 🗶 😋 😭 Hide path
恵下载∖IntellJ IDEA∖odps-studio-intellij-3.0.2_20181126174744.zip 上
> infunctionstudio
> 🖿 jar包
> Ksoftware
> 🖿 KwDownload
> MCStudio截图
> MININT
> mysql
> Program Files
> Program Files (x86)
> Users
> 1 作业
> eclipse插件
odps-studio-intellij-3.0.2_20181126174744.zip
Drag and drop a file into the space above to quickly locate it in the tree
OK Cancel

5. 回到插件首页后,单击OK,开始安装本地插件。

P Settings			X
Q,*	Plugins		Rese
> Appearance & Behavior	Q, [*]	Show: All plugins \smallsetminus	
Keymap		Sort by: name 🔻	MaxCompute Studio
> Editor	Coverage		C Restart IntelliJ IDEA
Plugins	Eclinea Integration		New version will be available after restart
> Version Control			MaxCompute is a big data processing platform developed by
> Build, Execution, Deployment			Alibaba Cloud independently. It is a fast and cloud-based big
> Languages & Frameworks	Git Integration		and processing models, which can provide data storage
> Tools	GitHub		warehouse and big data modeling service.
Vim Emulation	Gradie		MaxCompute Studio is a plugin for IntelliJ platform allowing data developers works with MaxCompute platform including
	Groovy		authoring SQL scripts, UDF extensions, MapReduce
	II8n for Java		browsing and unleading/downloading, job browsing and provide a state
	Carl IdeaVim		Features include:
	C IntelliLang		MaxCompute SQL language support
	Java Bytecode Decompiler	\checkmark	MaxCompute function development MaxCompute data management
	🗲 Java Stream Debugger	\checkmark	 MaxCompute job management
	🗲 JavaFX	\checkmark	Change Notes
	C JUnit	\checkmark	Changes in Version 3.0.0:
	🗲 Kotlin	\checkmark	New Feature:
	📲 Maven Integration		 support write authorization statement, show user privileges and diagnosis auth exception message
	MaxCompute Studio		use web view to draw sql pot graph and performance
	Check or uncheck a plugin to enable	e or disable it.	
	Install <u>J</u> etBrains plugin	Browse repositories Insta	all plugin from <u>d</u> isk
0			OK Cancel Apply

6. 安装完成后,弹出重新启动的提示窗口,单击Restart ,重新启动IntelliJ IDEA。



7. 重新启动后, 界面如下所示:



后续步骤

现在,您已经学习了如何安装MaxCompute Studio插件,您可以继续学习下一个教程。 在该教程中您将学习如何配置MaxCompute Project连接管理数据和资源。详情请参见 新 建MaxCompute项目空间连接。

2.3.3 查看和更新版本

查看Studio版本信息

通过以下步骤查看Studio的版本信息:

- 1. 打开Settings/Preferences页面 (Windows Ctrl-Alt-S, Mac 光,)
- 2. 在对话框左侧边栏选择Plugins, 然后搜索MaxCompute Studio
- 3. 可以看到MaxCompute Studio的版本号,以及版本的发布信息。

也可以在Setting页面左侧边栏选择MaxCompute Studio,再找到当前版本号。

检查新版本

缺省配置下, MaxCompute Studio会自动检测新版本, 当有新的可用版本时, 会自动通知用户。

MaxCompute Studio
 发现新版本: 3.0.2
 安装.或者配置版本更新检查器

收到更新提示后,用户可以选择:

- · 安装: 点击更新提示中的安装链接,将会自动下载并安装此新版本,安装完成后重启IntelliJ IDEA
- · 配置: 点击更新提示中的配置链接, 你可以配置是否自动检查新版本

如果关闭了自动更新功能,用户通过以下步骤可以检查MaxCompute Studio的版本更新并选择安装:

- 1. 打开Settings/Preferences页面 (Windows Ctrl-Alt-S, Mac 光,)。
- 2. 在对话框左侧边栏选择MaxCompute Studio。
- 3. 在Studio配置页面上,点击按钮Check new versions。
- 如果检测到新的可用版本,会提示用户新的版本号。点击按钮Install new version可以安装,重新启动 IntelliJ 完成安装。

可以通过Automatically checks for new versions复选框控制自动检查版本更新的开关。

] 说明:

Studio当前最新版本对应的JAVA SDK版本是0.28.4及以上。

下一步

建立MaxCompute项目连接

2.4 管理数据和资源

2.4.1 浏览表及 UDF

查看表及函数

在项目空间浏览器(Project Explorer)窗口中可以快速浏览已添加连接的表、函数、资源等。使 用前提是添加MaxCompute项目连接。

浏览表和函数

要浏览项目空间的表和函数,使用以下步骤:

1. 打开项目空间浏览器 (Project Explorer),即可以查看已添加的 Project节点树。



节点树上方是工具栏,包括:

- · 增加Project:新增一个到MaxCompute项目空间的连接
- · 删除Project: 删除一个项目空间浏览器中的项目连接, 对服务器端项目空间无影响
- ·刷新元数据:从服务器端项目空间刷新元数据信息,刷新本地元数据缓存
- · 展开节点:展开全部树节点
- · 折叠节点: 折叠全部树节点
- ·用户反馈:提交用户反馈
- ・ 在线文档: 打开在线文档

- 2. 双击或点击下拉箭头展开Tables节点,可列出该项目下的所有表(包括虚拟视图)。这里 的表名列表与用户执行show tables命令相等,需要用户在project下有list table权限。函 数(Functions)和资源(Resources)节点类似:
 - P odps_studio_dev
 - Tables & Views
 - alitrip_dest_relation
 - array_map_test2
 - haoping_sdk_log_copy
 - haoping_test_part
 - hy_test1
 - ▶ **⊞** key
 - 🕨 🧰 table test
 - Functions
 - Resources
- 3. Studio会将服务上的项目元数据下载到本地,当服务端元数据有更新时,如新增了一张表,需 手动触发一次刷新,将变化的元数据重新加载到本地。可以选择在项目(Project)或表(Table)级别做刷新,步骤如下:
 - a. 选中相应的节点。
 - b. 点击工具栏上的刷新图标或在右键菜单中选择刷新菜单项。



查看表详细信息

用户可以通过Studio的 表详情视图(Table Details View) 查看数据表详细信息。
- 1. 在节点树中,展开某个表名节点,可快速查看列名和类型:
 - P odps_studio_dev
 - Tables & Views W. alitrip_dest_relation ► array_map_test2 id :STRING arrays :ARRAY<STRING> maps :MAP<STRING,STRING> haoping_sdk_log_copy Þ haoping_test_part ► hy_test1 ► key ► table_test ► Functions ►.
 - Resources
- 2. 双击某个表或右键菜单Show Table Detail可以查看表的详细信息,包

括owner, size, column等元数据, 表结构信息, 以及data preview:

U I I I I I I I I I I I I I I I I I I I	Table information	Table information		Table schema					
a dev	key	value		name		type	comments		
101020223 stocontrol 395	Name:	b hour	hour		STRING				
listest1 test1	Comments:		c		STRING				
notest1_test1	Owner:	1830820614534790	d		STRING				
Tables & Views	Created time:	2016-02-18 15:20:46	e		STRING				
b_hour	Meta Modified time:	2016-04-18 16:48:18	f		STRING				
hour :STRING	Data Modified time:	2016-08-05 10:18:33	q		STRING				
C STRING	Size:	63048	h		STRING				
	Physical size:	189144							
	Partitions:	0							
e :STRING	Partition columns								
f :STRING	Sharding:								
g :STRING	Sharding columns:								
# h STRING									
h second	Data president								
	Data preview	1							
big	hour	c		d	e	r	9		
big2	00		null	null	null	Inull	null		
cc	01		Inull	Inull	Inull	null	Inull		
dual	02		null	null	Inull	Inull	null		
haoning asa2	03		null	Inull	Inull	null	null		
hasping_uuuz	04		null	Inull	Inull	Inull	null		
naoping_test4	05		null	null	null	null	null		
haoping_test_part	06		null	null	null	null	null		
hy_test	07		null	null	null	null	null		
limiao testabo	08		null	null	null	null	null		
mydual	09		null	null	null	null	null		
- here	10		null	null	null	null	null		
skew	11		null	null	null	null	null		
lt	12		null	inull.	null	null	null		
t_view	13		null	null	null	null	null		
t_yuyuan	14		null	Inull	null	null	null		
test1	15		null	null	Inull	Inull	null		
testa	16		null	null	Inull	Inull	Inull		
l testz	17		null	null	Inull	Inull	null		
] unit_test_sdk	18		null	Inul	Inull	Inull	null		
unit test sdk2	19		null	null	null	null	null		

Μ

3. 通过在Tables&Views右键菜单项Open specific entity,可以指定表名显示详情(注意要完 整表名称)。另外如果用户没有project的list权限,而只有具体某张表的权限,也可以通过这 种方式将该表抓取下来。函数(Functions)及资源(Resources)类似。

MaxCompute Project Explorer
• ① ① ▲ 至 □ ① ↓
▼ [®] P odps_studio_dev
Func Add project
Resc Refresh meta
▶ 'P wpp_de Open specific entity ▶ 'P cdo_mea
P studiotest1_test1
Open specific table
Name:
ОК Сапсе

Intellij IDE默认支持搜索,可展开表后直接敲击键盘模糊搜索。

4. studio也支持快速搜索某张表,可通过快捷键(win:Ctrl+Alt+Shift+N mac:#+\H+O)唤出 navigate symbol, 输入表名后回车即可。

Enter symbol name: Include non-project s	ymbols (乀策O) 🍞 💉
Q hy	\otimes
hy_partition_test odps_studio_dev	
<pre>hy_test odps_studio_dev</pre>	
hy_view odps_studio_dev	封河社区 yq.aliyun.com

可通过前置关键字(table:或function:或resource:)缩小搜索范围,譬如想搜索函数count,可 输入function:count。

5. 当想知道某张表在哪些script中用到时,可以右键该表,使用Find Usages功能。



查看函数详细信息

 在Functions树UserDefined节点下可以展开某个函数节点,以显示该函数的方法签名。双 击某个函数节点(或在Resources下双击改函数对应的源码资源),可打开该函数对应的的代 码。



1 说明:

Java代码通过反编译jar获取,并非源码。Python UDF解析签名需要安 装pyodps(MaxCompute python sdk),具体的先安装pip: sudo /usr/bin/python get-pip.py(请自行google下载get-pip.py),然后安装pyodps: sudo /usr/bin/ python -m pip install pyodps。 2. 在Functions树BuiltIn节点下分类显示了系统内置函数,展开显示签名,双击显示函数文档。



2.4.2 导入导出数据

Studio可以将CSV,TSV等格式的本地数据文件导入到MaxCompute表中,也可 将MaxCompute中表数据导出数据到本地文件。Studio是通过MaxCompute平台提供的批量数 据通道(Tunnel)功能完成的。

使用须知

- · 导入导出数据采用MaxCompute Tunnel服务,因此要求Studio中添加的MaxCompute Project必须开通或配置了Tunnel服务。
- ・导入导出表必须具备相应权限。

导入数据

1. 打开项目空间浏览器 (Project Explorer)窗口,在表名上点击右键或在表详细页面的Data Preview中字段属性上右键,选择Import Data Into Table。

里 Local Settings [C:\Users	\Administ	trator\Loc	al Settings]] - Intel	IJ IDE	A		
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>N</u> avigate	e <u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indow
Local Settings								
Project Explorer						₽ - I=		
+ - 🗘 🗉 🕏 🕷	2	¥ 🕘 🤉	୲ ?					
Multip://service.odps.al MySecondProject Tables & View Image: Tables &	iyun.com, t2 vs new nfo_d Q Find u Genera Genera Genera Genera Trunca Drop t Impor Export	ages rivilege h table m table deta table edit ate select ate DDL s table from t data into t data from	eta ail or statement tatement data n server o table m table					

2. 在弹出的Import Data对话框中,选择导入数据文件的路径,列分隔符(可自定义输入),大小限制,错误容忍行数等参数,点击按钮OK。

☑ Importing data to result_tabl X						
Input File:	D:\搜狗高速下载\MaxCompute客户端\					
File charset:	UTF-8 ~					
Column Separator:	Comma(',')		~			
Record Limit:	100	Size(MB) Limit: 10				
☑ Include Column Header Error Record Limit: 0						
0		OK Cano	:el			
Column Separator: Record Limit: Include Column ?	Loomma(,') 100 Header	Size(MB) Limit: 10 Error Record Limit: 0 OK Cano	v :el			

3. 提示Import Data Success, 表示数据导入成功, 可在表中查看导入的数据。

导出数据

- 1. 启动导出表数据有两种方式:
 - · 在表名上右键,选择Export Data From Table。

+ - 🗊 🖩 🕏 🕷	፻ 😤 🙆 🛈 ?	
 ✓ M http://service.odps.aliyu ✓ P MySecondProject2 ✓ Tables & Views > ■ bank_data 	n.com/api	
> emp_test_nev	v	
> cds_user_into > result_tabl > result_table	🔍 Find usages 🌀 User Privilege	
> IIII table_1 > IIII wc_in1 > III Functions	∫∫ Refresh table meta ॡ Show table detail	
> 🔚 Resources	Open table editor Generate select statement Generate DDL statement	
	Truncate table data O Drop table from server	
	Import data into table Export data from table	

如果您导出的是分区表,在导出时可输入分区值。

Exporting data to ods_user_info_d						
Input File:	D:\搜狗高速下载\Ma	D:\搜狗高速下载\MaxCompute客户端				
File charset:	UTF-8	~				
Partition:	pt='20180903'	~				
Column Separator:	Comma(',')	~				
Record Limit:	100	Size(MB) Limit: 10				
🗹 Include Column	Header	Error Record Limit: 0				
0		OK Cancel				

· 在表详细页面的Data Preview中字段属性上右键,选择Export Data From Table。

≜hy_test1 ×					
Table information			Table schema		
key	value		name	type	comments
Name:	hy_test1		id	BIGINT	
Comments:			name	STRING	
Owner:	ALIYUN\$dxp_0689468	4@aliyun.com	age	BIGINT	
Created time:	2016-12-29 11:50:22				
Meta Modified time:	2016-12-29 11:50:22				
Data Modified time:	2017-01-22 20:42:37				
Size:	2840				
Partitions:	0				
Partition columns	•				
Sharding:					
Sharding columns:					
			8 I		
Data preview					
- Sud preview			-		
1	2	Export data from table	10	11	age
1	a	Termont data inte table		1	

2. 弹出Export Data对话框,选择导出数据文件的保存路径,列分隔符(可自定义输入),大小限制,错误容忍行数等,填写完成后点击OK。

Exporting data from result_tabl							
Output File:	D:\搜狗高速下载\Max	D:\搜狗高速下载\MaxCompute客户端					
File charset:	UTF-8 ~						
Column Separator	: Comma(',')	~]				
Record Limit:	100	Size(MB) Limit: 10					
✓ Include Column Header Error Record Limit: 0							
0		OK Cancel]				

3. 提示Export Data Success, 表示数据导出成功, 可在目标文件中看到导出的数据。

用户也可以在Table的Data Preview窗格中选择右键菜单Export Grid Data导出数据。

₄hy_test1 ×								
Table information				Table schema				
key	value			name	type	comments		
Name:	hy_test1			id	BIGINT			
Comments:				name	STRING			
Owner:	ALIYUN\$dxp_06894	684@aliyun.com		age	BIGINT			
Created time:	2016-12-29 11:50:2	2						
Meta Modified time:	2016-12-29 11:50:2	2						
Data Modified time:	2017-01-22 20:42:3	7						
Size:	1280							
Physical size:	3840							
Partitions: 0								
Partition columns								
Sharding:								
Sharding columns:								
▲			§					
Data preview								
id		n	am	e		age		
1		a			11			
1		a			1			
		Copy cell content						
		Export grid data						
		Export grid data						
		View cell text						



在Data Preview窗格的导出数据功能仅导出显示在数据示例(Data Sample),而不一定是表中 的全部数据。

新类型导入导出

只需按照约定格式生成文本并存储为csv或tsv格式,就可通过studio导入到table。

下面将详细介绍各个数据类型的转换规则。

・基本类型

- 1. TINYINT, SMALLINT, INT, BIGINT直接存储为整型字符串,数值超过类型边界会报错。
- 2. FLOAT, DOUBLE存储小数字符串或浮点形式,如: 2.342 1E+7。
- 3. VARCHAR直接存储为字符串,超过上限会自动截断,不会报错。
- 4. STRING直接存储为字符串。
- 5. DECIMAL支持整形或浮点型的字符串。
- 6. BINARY需要将二进制数据编码为base64 string。
- 7. DATETIMEdate time需import dialog中指定的format格式保持一致,格式不匹配将报错。
- TIMESTAMPtimestamp需要按照yyyy-[m]m-[d]d hh\:mm\:ss[.f...]格式存储为 字符串。
- 9. BOOLEANtrue or false字符串。

1. ARRAY需存储为JSON数组,数组元素按照本文约定规则转换成字符串,数组元素支持任意 类型。

工具及下载 / 2 MaxCompute Studio

- 2. MAP需存储为JSON对象, map key、value按照本文约定规则转换为字符串, value支持任 意类型嵌套。
- 3. STRUCT需存储为JSON对象, struct字段名为string, 转换为JSON对象的key, struct字段值转换为JSON对象的value, 字段值以本文定义规则转换。

示例

· array类型

对于如下所示表结构:

列名	列数据类型
c_1	ARRAY <tinyint></tinyint>
c_2	ARRAY <int></int>
c_3	ARRAY <float></float>
c_4	ARRAY <datetime></datetime>
c_6	ARRAY <timestamp></timestamp>
c_7	ARRAY <string></string>

可通过下面所示csv格式导入数据:

```
c_1,c_2,c_3,c_4,c_6,c_7
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00",""2017-11-11 00:00:00"",""2017-11-11
00:00:00"]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"","ccc""]"
"[""1"",""2"",""3""]","[""1"",""2"",""3"",""4""]","[""1.2"",""2.0
""]","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
"[""1"","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
"["1"","[""2017-11-11 00:00:00"",""2017-11-11 00:00:00"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb
"",""ccc""]"
"[","[""2017-11-11 00:00:00.123456789"",""2017-11-11 00:00
:00.123456789"",""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789"",""2017-11-11
00:00:00""]","[""2017-11-11 00:00:00.123456789""]","[""aaa"",""bbb"",""ccc""]"
```

📕 说明:

CSV格式需要对双引号进行转义,通过两个双引号来表示双引号,具体可参考CSV格式规范。

· map类型

对于如下所示表结构:

列名	列数据类型
c_1	MAP <tinyint,string></tinyint,string>
c_2	MAP <string,int></string,int>
c_3	MAP <float,string></float,string>
c_4	MAP <string,datetime></string,datetime>
c_5	MAP <string,string></string,string>
c_6	MAP <timestamp,string></timestamp,string>

可通过下面所示csv格式导入数据:

c_1,c_2,c_3,c_4,c_5,c_6 "{1:"2345""}","{"123"":"2",""3":"4""}","{2.0:"223445"",1.2 :""1111"}","{"aaa":"2017-11-11 00:00:00"",""ccc"":"2017-11-11 00:00:00"",""bbb"":"2017-11-11 00:00:00""}","{"ckey"":"cvalue ""}","{"2017-11-11 01:00:00.123456789"":"'dddd"","2017-11-11 00:00 :00.123456789"":"aaa"","2017-11-11 00:01:00.123456789"":"'ddd""}" "{1:"2345""}","{"123"":"2",""3":"4""}","{2.0:"223445"",1.2 :""1111""}","{"aaa"":"2017-11-11 00:00:00"",""ccc"":"2017-11-11 00:00:00"",""bbb"":"2017-11-11 00:00:00"",""ccc"":"2017-11-11 00:00:00"",""bbb"":"2017-11-11 00:00:00""}","{"ccc"":"2017-11-11 00:00:00",""bbb"":"2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789"":""aaa",""2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789":""aaa",""2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789":""aaa",""2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789":""aaa",""2017-11-11 00:00:00",""ccc"":"2017-11-11 00:00:00",""bbb":""2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789"::""aaa",""2017-11-11 00:01:00.123456789"":""ddd""}" "{1:"23456789"::""aaa",""2017-11-11 00:00:00",""ccc"::"2017-11-11 00:00:00",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00"]",""bbb"::""2017-11-11 00:00:00"]",""ccc"::""2017-11-11 00:00:00"]",""bbb"::""2017-11-11 00:00"]"]",""ccc"::""2017-11-11 00:00 :00.123456789"::""ddd"",""2017-11-11 00:00

・ struct类型

对于如下所示表结构:

列名	列数据类型
c_struct	<ruct<x:int,y:varchar(256),z: STRUCT<a:tinyint,b:string>></a:tinyint,b:string></ruct<x:int,y:varchar(256),z:

可通过下面所示csv格式导入数据:

```
c_struct
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b
"":""stringdemo""}}"
```

```
"{""x"":""1000"",""y"":""varchar_test"",""z"":{""a"":""123"",""b
"":""stringdemo""}}"
```

2.4.3 可视化创建、修改和删除表

Studio的project explorer提供了可视化表结构编辑器,可进行创建、修改表。

可视化建表

操作步骤

1. 右击需要创建表的project,选择Create a new table:



2. 弹框里填写对应的表名、列等信息,然后点击Generate CreateTable Statement生成对应的DDL语句,点击Execute,执行建表。

및 创建表/视图 - [MySecondProject2]				X
创建表: 注释: 生命周期:		不存在时创建外部表[_	视图	
□以 AliOrc 格式存储 □以后表为模板	创建	~		1
数据列 分区列 索引 AS SELECT	附加属性			
d name	type	not null	comment	+
SQL DDL SCRIPT				+
1 CREATE TABLE MySecon	dProject2`.`` (<u>0</u>
col1 BIGINT)	1		4	
操作: 提交到 MaxCompute 执行 🛛 🗸		取	消执行	行开编辑器

表/列名、生命周期、列类型等限制请遵循MaxCompute相关要求,可参见_{表操作}。

道 说明:	
studio可视化建表并没有选择flag的设置,默认使用以下两个flag:	
<pre>a. odps.sql.submit.mode : script</pre>	
<pre>b. odps.sql.type.system.odps2: true</pre>	
表创建成功后,可以在Drojoct Explorer的tableSwjew本看到元物提	芋坮 て列港Do

3. 表创建成功后,可以在Project Explorer的table&view查看到元数据,若找不到请Refresh meta。

可视化修改表

操作步骤

1. 在Project Explorer的table&view列表中,右击需要修改的表,选择Open table editor:



9. 弹框中对表进行编辑,可以修改表comments、lifecycle,修改列名、列描述,新增列。具体规则遵循MaxCompute table相关要求,可参看表操作。

nsisca≊ - [nni					
長名:	table_2		转为存档		
ì释:					
- 今田知.	1				
上叩问别:	-1				
数据列 分区	区列 索引 附加	加属性			
na	ame	type	not null	comment	+
col1		BIGINT			_
col2		BIGINT			
col4		BIGINT			1
col3		BIGINT			+
col5		BIGINT			
SQL DDL SC	CRIPT				
	CRIPT	able_2 ADD COLUMNS	· · · · · · · · · · · · · · · · · · ·		
SQL DDL SC	CRIPT LTER TABLE to Col5' BIGIN	able_2 ADD COLUMNS T):			
SQL DDL SC 1 AL 2 AL 3	CRIPT TER TABLE to co15 BIGIN	able_2 ADD COLUMNS T):			
SQL DDL SC 1 P AI 2 P 3	CRIPT LTER TABLE to Col5' BIGIN	able_2 ADD COLUMNS T):			<u>e</u>
SQL DDL SC	CRIPT LTER TABLE to co15 BIGIN	able_2 ADD COLUMNS T);	(
SQL DDL SC 1 AI	CRIPT TER TABLE to co15 BIGIN	able_2 ADD COLUMNS T):			
SQL DDL SC	CRIPT TER TABLE to Col5 BIGIN	able_2 ADD COLUMNS T):	(9
SQL DDL SC 1 AI 2 3	CRIPT LTER TABLE to col5 BIGIN	able_2 ADD COLUMNS T);	(<u> </u>
SQL DDL SC 1 P AI 2 - 3	CRIPT LTER TABLE to col5' BIGIN	able_2 ADD COLUMNS T):	(<u>e</u>

3. 填写完成修改项后,点击Alter Table Statement 生成具体的Alter语句,点击Execute执行表修改操作。执行成功可以对表的元数据进行查看。

可视化删除表

在Project Explorer的table&view列表中,右击需要修改的表,选择Drop table from server:



弹框中选择OK即可将表从MaxCompute服务上删除。

2.5 开发SQL程序

2.5.1 创建MaxCompute Script Module

开发MaxCompute Script前,需要创建一个MaxCompute Script Module,创建时存在以下两种情况:

本地没有script文件

本地没有script文件时,可通过Intellij创建一个全新的Module。

操作步骤

 打开或者新建一MaxCompute Studio Project。本文以新建为例,单击菜单栏中的File,导航 至New > Project。如下图所示:

IJ	Local Se	ettings	[C:\Users\A	Administ	trator\Loc	al Settings]	- C:\U	sers∖A	dminist	rator\.	IdeaIC
Eile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> ind
	New				>	<u>P</u> roject					
	Open					Project	from E	Existing	g Source	es	
	Open <u>F</u>	Recent			>	Project	from \	Versio	n Contro	ol	>
	Close F	roject				<u>M</u> odul	e				
۶	Se <u>t</u> ting	s		Ct	rl+Alt+S	Modul	e from	Existir	ng Sourd	es	
Ē.	Project	Struct	ure (Ctrl+Alt-	+Shift+S	揚 Scratch	n File	Ctr	l+Alt+S	hift+In	sert
	Other 9	Setting	s		>	븕 FXML F	ile				
	Import	Setting	gs							5	
	<u>E</u> xport	Setting	js							6	
	Export	to Ecli	pse							7	Ę
	Export	to Zip	File							8	
	Setting	s Repo	sitory							9	
H	<u>Save</u> A	I			Ctrl+S						
9	Synchro	onize		Ct	rl+Alt+Y						
	Invalida	ate Cao	ches / Resta	art							
Ð	Print										
	Power	Save N	1ode								
	E <u>x</u> it										
						_					

New Project		8						
Java	Additional Libraries and <u>F</u> rameworks:							
Java FX								
🖷 Android								
뜬 IntelliJ Platform Plugin								
<i>M</i> Maven	Nothing to show							
📀 Gradle								
G Groovy								
🗐 Griffon								
【 Kotlin								
Empty Project								
<mark> MaxCompute Java</mark>								
<mark>M</mark> MaxCompute Studio								
	Previous <u>N</u> ext Cancel Help							

2. 选择左侧导航栏中的MaxCompute Studio,单击Next。

3. 填写Project Name, 单击Finish。

Project	
Project n <u>a</u> me:	MySQLProject
Project location:	C:\Users\Administrator\IdeaProjects\MySQLProject
► Mor <u>e</u> Settings -	
	Previous Finish Cancel Help
道 说明:	
如果之前有打法	FF的Project,将会提示您是否在当前窗口中打开(即关掉之前的Project),选
择This Wind	ow₀
r	
New Project	
New How	projects can either be opened in a new window or replace the project in the existing window. would you like to open the project?

<u>R</u>emember, don't ask again

New Window

<u>T</u>his Window

4. 创建完成后,页面如下图所示,您即可在此项目中开发SQL脚本。

MySQLProje	ct [C:\Users\	Administ	trator\Ide	aProjects\I	MySQL	Project	t] - Intel	IJ IDE	A		
<u>F</u> ile <u>E</u> dit <u>V</u> iew	v <u>N</u> avigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indow	MaxCompute	<u>H</u> e
🔥 MySQLProj	ect										
🗊 Project 🔹			\odot	≑ ☆ -	+						
AvsQLPr Mill External L D D Scratches	oject C:\Use ibraries and Consol	ers\4 dm	inistrator ¹	(IdeaProjec	cts\						
										Search Ev	/ery
										Go to File	e Ct
										Recent Fi	iles

本地已有script文件

假如本地某个文件夹下已经有很多script,此时需要用MaxCompute studio来编辑这些 script

,您可直接打开一个Module。

操作步骤

1. 在scripts文件夹下创建一个MaxCompute的连接配置文件 *odps_config.ini*,在里面配置 与MaxCompute连接的鉴权信息:

- project_name=xxxxxxxx
- · access_id=xxxxxxxxx
- · access_key=xxxxxxxx
- end_point=xxxxxxxxx

								Users/liuyi	/scripts
				* ~	Ê	\odot	<u>5</u>		
	路径	显示	排列	操作	共享	添加标记	连接		
	名称								
	a.osq	ĮI							
Ш	odps_	_config.ini							
	script	tMode.osql							
	test.c	lpsql							

2. 打开IDEA, 导航至File > open, 选择您的scripts文件夹。



3. MaxCompute Studio会探测该文件夹下是否存在odps_config.ini文件,根据这个文件里的配置信息抓取服务端的meta,然后编译文件夹下的所有script。

2.5.2 编写SQL脚本

MaxCompute Studio模块创建完成后,即可开始编写MaxCompute SQL脚本。

操作步骤

1. 右击scripts, 导航至New > MaxCompute Script。

IJ	Myfire	stProject	: [C:\L	Jsers∖	Administ	trator\Ide	aProjects\N	/ yfirstP	roject]	- Int	elliJ IDB	A		
Eil	e <u>E</u> di	t <u>V</u> iew	<u>N</u> av	vigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	<u>T</u> oo	ls VC <u>S</u>	<u>W</u> indow	MaxCompute	<u>H</u> elp
V	Myfi	rstProje	$ \mathbf{ct}\rangle $	scri	ipts $ angle$									
ñ	Proje	ect 🔻				Θ	⇒ ☆・	←						
	M M	fire+Dre	viect	CALLER	ars\Adm	inistrator		+c)						
Ť		Lidea		0.105	ers (Autri	iniisti ator	udeariojed							
	>	MvFris	tMod	lule										
	>	scripts			_									
		target	_	N	ew)		Java Cla	ass		
	>	wareho	use	Жc	u <u>t</u>			0	Ctrl+X		Kotlin F	ile/Class		
	4	Myfirst	Proje		ору			0	Ctrl+C		File			
>	IIII Ext	ternal Lil	brarie	C	<u>o</u> py Path	ı		Ctrl+S	nift+C	=0	Scratch	File (Ctrl+Alt+Shift+In	sert
>	Sci	ratches a	and C	C	op <u>y</u> Refe	erence	Ctrl	+Alt+Sl	nift+C		Packag	e 		
	_			<u>Ге</u>	aste			0	Ctrl+V	<>	FXML F	ile		
				Fi	nd <u>U</u> sag	es		A	lt+F7		packag	e-info.java		
				Fi	nd in <u>P</u> a	th		Ctrl+S	hift+F		module	-info.java	_	
				R	epl <u>a</u> ce ir	n Path		Ctrl+S	nift+R	뿸	MaxCo	mpute SQL	. 脚本	
				Α	naly <u>z</u> e)	쀨	MaxCo	mpute 交互	式查询	
				R	efactor)	8	从 D2 カ	載		
					dd to Fa	vorites			3	H	HTML F	ile		
				s	how Ima	ae Thumh	onails	Ctrl+S	r hift+T		JavaFX/	Application		
							, and the second s	Culu	A 14 - 1		Singleto	on 		
				<u>K</u>	etormat	Lode		Ctrl+	AIT+L	G	Gradle	Kotlin DSL	Build Script	
				0	ptimi <u>z</u> e -lata	Imports		Ctri+i	Alt+O	G	Gradle	Kotlin DSL	Settings	
				<u> </u>	elete			L	Pelete		XSLT St	ylesheet		
F -1		_		B	uild <u>M</u> oo	dule 'Myfi	rstProject'				Edit File	e Template	s	
EV	ent Lo	g		R	ebuild '«	<default></default>	(Ctrl+Sh	itt+F9	省	GUI Fo	rm		
9	1			S	how in E	xplorer				省	Dialog			
	1			0 🗉	pen in te	erminal					Form S	napshot		
4 =5				뿔 Se	et MaxC	ompute p	roject			í.	Resour	ce Bundle		
				Lo	ocal <u>H</u> ist	ory)	e	Plugin (DevKit		>

2. 填写弹出框中的相关内容,点击OK。

UNew MaxCompute SQL Script	X
Script Name: MyFirstSQL MaxCompute Project: MySecondProject2 (service.odps.aliyun.co	m) ~ +
ОК	Cancel

- · Script Name: 脚本名称。
- · Script type: 脚本类型。
- Target Project:目标MaxCompute项目。单击后面的+即可新建一个MaxCompute Project,配置详情请参见新建项目空间连接。
- 3. 在SQL文件编辑界面中,编写SQL。





 a. 实际SQL请根据自己的MaxCompute Project中的表进行编写。可单击toolbar右上 角切换绑定的不同的MaxCompute项目,也支持跨project资源依赖。例如script绑定 了ProjectA,同时还会用到ProjectB.table1,这时Studio会自动使用ProjectA的账号去 抓取ProjectB的元数据。表的元数据Studio会保存在本机中类似下图的位置:

```
liuyi-MBP:hy_test liuyi$ pwd
/Users/liuyi/.odps.studio/meta/odps_studio_dev/tables/hy_test
liuyi-MBP:hy_test liuyi$ cat schema.ddl
CREATE TABLE IF NOT EXISTS `odps_studio_dev`.`hy_test` (
   `id` BIGINT,
   `name` STRING);
```

b. 新建sql script时的代码模板是可以在Intellij Preferences页修改的: P Settings 83 Editor > File and Code Templates Q. > Appearance & Behavior Scheme: Default \sim Keymap ✓ Editor Files Includes Code Other > General ---name:\${NAME} + - 🖬 🖻 Font author: \$ {USER} 븕 JUnit5 Test Class > Color Scheme --create time:\${TEAR}-\${NONTH}-\${DAT} \${HOUR}:\${MINUTE} 📒 JUnit5 Test Method Code Style G 描 MaxCompute Aggregator AxCompute Combiner Java ē 📒 MaxCompute Extractor Groovy ē MaxCompute GraphLoader HTML G MaxCompute新建文件时相关的代码模板 都在这里了,可根据自己的需要修改。 JSON ē 📇 MaxCompute Mapper 着 MaxCompute MR Driver Kotlin 6 AaxCompute Outputer ē OdpsQL 🚔 MaxCompute PyODPS Script Properties ē 🚔 MaxCompute Python UDAF MaxCompute Python UDF XML 6 MaxCompute Python UDTF Reformat according to style Enable Live Templates YAML MaxCompute Reducer Description Other File Types G MaxCompute Sql Script MaxCompute StorageHandler Inspections ē Apache Velocity template language is used 着 MaxCompute UDAF File and Code Templates HaxCompute UDF File Encodinas G 着 MaxCompute UDJ AxCompute UDTF Live Templates 📒 MaxCompute Vertex File Types 📇 MaxCompute VertexResolve Android Layout Editor New Kotlin Function Body > Copyright G OK Cancel Apply 2

MaxCompute Studio功能

MaxCompute Studio不仅提供语法高亮,智能提醒,错误提示,还支持以下功能:

- · schema annotator: 当鼠标悬停在表上,可显示其schema。悬停在列上,可显示其类 型;悬停在函数上,可显示其签名。
- · code folding: 可以将子查询等折叠起来, 方便长SQL的阅读。
- · brace matching: 鼠标单击高亮左括号,其匹配的右括号也会高亮,反之亦然。
- · go to declaration:按住Ctrl键,单击table,即可查看table详情。单击function,即可显 示其源码。

 code formatting:支持对当前脚本格式化,快捷键(Ctrl + Alt + L)。可在如下页面自定义 格式化规则,譬如关键字大小写,是否换行等。

MyfirstProject [C:\Users\Administrator\Ide	aProjects\MyfirstProject]\scripts\My	FirstSQL.osql [MyfirstProject] - IntelliJ IDEA			
File Edit View Navigate Code Analyze	Refactor Build Run Tools VCS	<u>V</u> indow MaxCompute <u>H</u> elp			C-3 Circo in
	Q* > Appearance & Benavior	Editor > Code Style > OdpsQL	·*	<u></u>	ondProject
Ctrl+Alt+S Ctrl+Alt+Shift+S Other Settings	Keymap V Editor	General Tabs and Indents Spaces Wrappin	g and Braces Blank	Set from	
Import Settings Export Settings Export to Eclipse Export to Zip File Settings Repository	Font > Color Scheme ~ Code Style Java	V Word case Keywords Types Identifiers Functions Vew line before Column type	To upper As keywords Do not change Do not change	<pre>SFI = = b:SFI = obs. rol.maper MERGE LIMIT mire = 04;CREA id BIGLIT, mane STRING)PARTITIONED BI (partition_column STRING):CREATE TABLE INSERT INTO 11 PARTITION (partition_column. = 'he')' SELECT 11.id, 12.id.,tl mame, 12.mame</pre>	-
Image: Save All Ctrl+S Image: Synchronize Ctrl+Alt+Y Invalidate Caches / Restart	Groovy HTML JSON	Select clause From clause Join expression Join condition		<pre>FROM t1 JOIN t2 OF (t1.id = t2 id AND t1.name = 'abcdefg FROM (SELECT id + 100 AS 'aVeryLongIdentifierToTestWrap', concat (name, 'a'),</pre>	
Export to HTML Print Add to favorites File Encoding Line Separators Make File Read-only	Kotin OdpsQL Properties XML YAML Other File Tunes	Indie captersion(and]or) Group by clause Having clause Values clause Other clause(order byllimit) Query expression inside parenthesis Query expression inside DML statements		88 +2000, id + 7; FROM +1 WHERE (id BETWEEN 1 AND 100 OR id = 110))tmp SELECT aVeryLangidentifierToTestWrap GROUP BT aVeryL	
Power save Mode Egit Event Log © © © ©	Inspections File and Code Templates File Encodings Live Templates File Types Android Layout Editor	Multi insert V New line after Each select item New line around semicolon Select item 'as' inside select statement 'as' inside with clause 'a' inside set clause 'a' inside set clause Types inside create table			¢- ±
	> Copyright	Inside query expression	Ø	OK Cancel Apply	

code inspect:支持对当前脚本进行代码检查,某些检查还会提供quick fix,可通过快捷
 键Alt + Enter唤出。另外,可在Settings > Editor > Inspections > MaxCompute处修改某
 条规则。



- find usages:选中editor中的某张表(或函数),右键菜单选Find Usages,则会在当前IntelliJ project下寻找所有使用该表的脚本。
- · live template: Studio内置了一些SQL live template,可以在编辑器中使用Ctrl + J
 (Command + J on Mac OS X) 快捷键唤出(例如忘记了insert into table的语法,便可唤出live template popup后搜索insert table)。

 builtin documentation: 支持在系统内置函数处通过Ctrl + Q (Ctrl + J on Mac OS X) 唤 出帮助文档。



· sql history: 所有通过studio提交运行的sql我们都记录在本地了,你可以点击toolbar上的图标,弹出sql history窗口,查询你曾经执行过的SQL。

·stSQL.osql ×									
单句模式 V MaxCompute类型系统 V 默认编译器 V 〔〕 ↓ 》 👔 📷 📴 (SQL History)									
name:MyFirstSQL									
author: Administrator									
create time:2018-12-06 17:36									
CREATE TABLE `MySecondProject2`.`table_4` (
☑ Sql History)	
Q		Project: ALL			✓ File:	Sql:	@, « » 1		
	Pro	oject Name	File Name	Sql Text	Start Time	End Time	Instance Id		
	MySe	econdProjec	MyFirstSQL.osql	CREATE TABLE `	2018-12-07 14:5	2018-12-07 14:5	20181207065230		
8	MySe	econdProjec	MyFirstSQL.osql	insert into table	2018-12-07 14:4	2018-12-07 14:4	20181207064858		
N N N N N N N N N N N N N N N N N N N	MySe	econdProjec	MySecondProject2	select * from My	2018-12-06 16:5	2018-12-06 16:5	20181206085502		
•	iviyse	econdProjec	wysecondProject2	select " from iviy	2018-12-00 10:5	2018-12-00 10:5	20181206085445		
				\					
	单击一行,下方显示其soL文本								
	单击一行,下方显示其sqL文本								
	_								
1	1 CREATE TABLE 'MySecondProject2'.'table_4' (
2		coll BIGI	NT,				_		
3		col2 BIGI	NT,				-		
4		col3 BIGI	NT)				-		
5		1;							
2									

详细的SQL编写命令介绍请参见表操作。

2.5.3 提交SQL脚本

MaxCompute Studio可直接将MaxCompute SQL提交到服务端运行,并显示查询结果,执行计 划等详细信息。它在提交前会进行编译,能够有效避免提交到服务端后才发现编译错误。

前提条件

- · 首先创建MaxCompute项目连接,并绑定目标项目。
- · 创建 MaxCompute Studio Module。
- · 在提交前需根据自身需求进行相关设置。MaxCompute Studio提供了丰富的设置功能,可 在Editor编辑页面上方的Tool Bar工具栏中快速设置。设置主要分为以下三种:
 - 编辑器模式:编译器模式设定包括两种模式,脚本模式和单步模式。
 - 单步模式会将提交的脚本文件按;分隔,逐条提交到服务端执行。
 - 脚本模式为最新开发模式,可将整条脚本一次提交到服务端,由服务端提供整体优化,效率更高,推荐使用。
 - 类型系统:类型系统主要解决 SQL 语句的兼容性问题,分为以下三种类型:
 - 旧有类型系统:原有 MaxCompute 的类型系统。
 - MaxCompute 类型系统: MaxCompute 2.0 引入的新的类型系统。
 - Hive 类型系统: MaxCompute 2.0 引入的 Hive 兼容模式下的类型系统。
 - 编译器版本: MaxCompute Studio提供稳定版编译器和实验性编译器两种模式:
 - 默认编译器:稳定版本。

■ 实验性编译器: 包含编译器最新特性。

▋ 说明:

脚本提交设置可采用全局设定,导航至File > Settings > MaxCompute,选择MaxCompute SQL,其中Compiler > Submit中可以设置以上属性。

提交SQL脚本

Editor上方工具栏中提供同步,编译,提交功能。

* **同步功能**:更新SQL脚本中使用的元数据,包括表名、UDF等。如果Studio提示表或函数找不到,而服务端又明确存在时,可尝试使用该功能更新元数据。 * **编译、提交**:按MaxCompute SQL预发规则编译或提交到服务端,编译错误会在 MaxCompute Compiler窗口中显示详细信息。

具体操作

编写好SQL语句后,单击工具栏或侧边栏上的绿色运行图标,或者右键Script Editor,选择Run MaxCompute SQL Script,即可提交到服务端。当SQL中存在变量时(如下图的\${bizdate}),会弹出对话框,提示您输入变量值。

MyFir	rstSQL.osql ×								
	单句模式 ~	MaxCompute类型系统 🗸 默认编译器 🗸 ᡗ 📲 🕨 🎓 🔤 💣							
1	name:MyFir	stSQL							
2	author:Adm	inistrator							
3	create tim	create time:2018-12-06 17:36							
4 🕨 🤇	CREATE TABL	CREATE TABLE MySecondProject2. table_5 (
5	coll' BIGINT,								
6	col2 BIG	col2 BIGINT,							
7	col3 BIG	INI)							
8 0	8 -:								
	insert into table_2 values (1, 'a', '201908908', '20180907');								
10	select * from ods_user_info_d where ds='\${bizdate}';								
		Specify script extra info bizdate : 20181206 OK Cancel							

在您的SQL任务运行前,IDEA会向您提示预估的SQL费用。

Run: 🐪 billing.osql 🗵	\$- <u>+</u>
	Confirmation
■ 1> 2019-01-25 14:15:56 Status: ESTIMATE_COST	Cost Estimate:
=	Inputs:0 Bytes
2	Complexity:1.0
×	Do not show this dialog again
	Cancel OK

2. SQL会先被本地编译(依赖于你在Project Explore窗口中添加的项目元数据),无编译错误后就会提交到服务端执行。SQL执行过程中会显示运行日志,当已经开始在服务端运行时,会打开job detail页,显示作业运行的基本信息以及执行图:



3. 可以在结果页查看SQL结果,单句模式下当存在多条语句时,会显示每条语句的结果。可以选择 表格中的一些行或列, Ctrl + C到剪切板中。



2.6 开发Java程序

2.6.1 创建MaxCompute Java Module

MaxCompute Studio能支持用户开发Java UDF和MR,首先需要新建一个MaxCompute Java Module。

创建module

依次单击File > new > module, module类型为MaxCompute Java, 配置Java JDK。单击next, 输入module名, 单击finish。studio会帮用户自动创建一个maven module,并引入MaxCompute相关依赖, 具体请查看pom文件。

module结构说明

至此,一个能开发MaxCompute Java程序的module已建立,如下图的mDev。主要目录包括:

- · src/main/java:用户开发java程序源码。
- · examples: 示例代码,包括单测示例,用户可参考这里的例子开发或编写UT。
- ·warehouse:本地运行时需要的schema和data。

🗊 Project 👻	Θ	+	÷	
🗸 🐼 MyfirstProject C:\Users\Administrat	or\Io	deaPi	rojec	ts∖M
> 🖿 .idea				
MyFristModule				
> 🖿 examples				
Y src				
Y main				
> 🖿 java				
resources				
> 🖿 test				
> target				
MyFristModule.iml				
m pom.xml				
> scripts				
target				
warehouse				
example_project				
v tables				
>ads log				
> kmeans in				
> kmeans out				
> Dagerank_in				
> pagerank_out				
> 🖿 rs_out				
> 🖿 sssp_in				
> 🖿 sssp_out				
> wc_in1				
> 🖿 wc_in2				
> wc_out				
📲 MyfirstProject.iml				
> IIII External Libraries				
Scratches and Consoles				

2.6.2 开发和调试UDF

创建完成MaxCompute Java Module 后,即可开发UDF。

操作步骤

1. 展开已创建的MaxCompute Java Module目录, 导航至src > main > java > new, 单 击MaxCompute Java 。如下图所示:



2. 填写name和kind, 单击OK。如下图所示:



- · Name:填写创建的MaxCompute Java Class名称,如果还没创建package,可以在此处 填写 packagename.classname,会自动生成package。
- Kind:选择类型。目前支持的类型有:自定义函数(UDF/UDAF/UDTF)、MapReduce (Driver/Mapper/Reducer)、非结构化开发(StorageHandler/Extractor)等。
- 3. 创建成功后,即可进行开发、编辑、测试Java程序。



说明:

这里的代码模板可在Intellij中自定义,具体的: Preference > Editor > File > Code Templates,然后在Code标签页中寻找MaxCompute对应的模板修改。

通常情况下, JAVA UDF的开发可以通过以下几种方式:

- · 使用MaxCompute Studio完成JAVA UDF开发整个流程。
- ·使用Eclipse插件开发和调试JAVA UDF,导出Jar包,然后通过命令或者DataWorks添加资源后 再注册函数。

了解详细开发步骤,请参见开发Java UDF#可选#。

调试UDF

开发UDF完成后,即可通过单元测试和本地运行两种方式进行测试,看是否符合预期。

单元测试

在examples目录下有各种类型的单元测试示例,可参考示例编写自己的Unit Test。



IDE and Plugin Updates: IntelliJ IDEA is ready to update. (today 16:45)

本地运行

本地运行UDF时,需要指定运行数据源,有以下两种方式设定测试数据源:

- · MaxCompute Studio通过Tunnel服务自动下载指定项目下的表数据到warehouse目录下。
- ·提供Mock项目及表数据、您可参考warehouse下的example_project自行设置。

操作步骤

 右击UDF类,单击运行,弹出run configuration对话框。UDF/UDAF/UDTF一般作用 于select子句中表的某些列,需要配置MaxCompute project,table和column(元数据来源 于project explorer和warehouse下的Mock项目)。复杂类型的调试也是支持的,如下图:



2. 单击OK。


- ·如果指定项目下的表数据未被下载到warehourse中,需要先下载数据,默认下载100条。 默认下载100条,如需更多数据,可配置Download record limit项。
- ·如果采用Mock项目或已下载数据,则直接运行。
- · UDF的local run框架会将warehouse中指定列的数据作为UDF的输入,开始本地运行 UDF,您可以在控制台看到日志输出和结果打印。

本地warehouse目录

本地warehouse用于存储表(包括meta和数据)或资源,用于本地执行UDF或MR。 warehouse目录如下图所示:



说明:

- · warehouse目录下依次是项目名, tables, 表名, 表schema和sample data。
- schema文件依次配置项目名,表名,以及列名和类型(冒号分隔),分区表还需配置分区
 列(非分区表参考wc_in1,分区表参考wc_in2)。

- · data文件采用标准csv格式存储表的sample数据:
 - 特殊字符为逗号, 双引号和换行(\n或\r\n)
 - 列分隔符为逗号, 行分隔符为\n或\r\n
 - 如果列内容里包含特殊字符,需要在该列内容前后加上双引号,例如: 3,No -> "3, No"
 - 如果列内容包含双引号,则每个双引号转义成两个双引号,例如: a" b" c -> "a"" b" " c"
 - \N表示该列为null, 如果该列内容(string 类型)就是\N, 需要转换为""" \N"""
 - 文件字符编码为UTF-8

2.6.3 开发MapReduce

创建完成 MaxCompute Java Module后,即可以开始开发MR了。

开发MR

- 1. 在Module的源码目录即src > main上右键new > java,选择MaxCompute Java。
- 2. 分别创建Driver、Mapper、Reducer。

U Create	new MaxCompute java class	23
<u>N</u> ame:		ţ↑
<u>K</u> ind:	Oriver	\sim
	OK Cance	I

3. 模板已自动填充框架代码,只需要设置输入/输出表, Mapper/Reducer类等即可。

c	C HelloDriver.java ×			
	HelloDriver			
1			pack	age mymr.myudf;
2				
3		+	impo	ort
11	•		publ	ic class HelloDriver
13			pube	
14		E	J	<pre>public static void main(String[] args) throws OdpsException {</pre>
15		ľ		
16				<pre>JobConf job = new JobConf();</pre>
17				
18				<pre>// TODO: specify map output types</pre>
19				job.setMapOutputKeySchema(SchemaUtils. <i>fromString</i> (<u>,?)</u>);
20				job.setMapOutputValueSchema(SchemaUtils. <i>fromString(_?))</i> ;
21				
22				<pre>// TODO: specify input and output tables</pre>
23				InputUtils.add/able(lableInfo.builder().tableName(_?).build(), job);
24				outputotils.addrable(rableinto.builder().tableName(_?).build(), job_;
25				// TODO: specify a manner
20				ioh setMannerClass(2);
28				// TODO: specify a reducer
29				iob.setReducer(lass(?):
30				
31				RunninaJob ri = JobClient. <i>runJob</i> (job):
32				<pre>rj.waitForCompletion();</pre>
33				
34		Ę		}
35			_	
36			}	

开发MR详情请参见编写MapReduce#可选#。

调试MR

MR开发好后,下一步就是要测试自己的代码,看是否符合预期,目前支持以下两种方式:

· 单元测试: 在examples目录下有WordCount的单测实例, 可参考例子编写自己的UT。

© Project ▼	C HelloD	river.java × 🕜 WordCount.java × 🔇	WordCountTest.java ×	
✓ ✓ MyfirstProject C:\Users\Administrator\IdeaProjects\M	Project SI)K is not defined		
> 🔝 .idea	1	package com. aliyun. odps. examples. mr. tes	st;	
MyFristModule	2			
 examples com alivun odas avamalas 	3	import		
Contailyunoupstexamples	18	muhlis alege WardfauntTast extends Wi		
> in comalivur.odps.examples.oraph	20	// 完义输入输出表的 schema	λ Cu <u>t</u>	Ctrl+X
> im com.alivun.odps.examples.mr	21	private final static String INPUT_	<u>в</u> <u>С</u> ору	Ctrl+C
com.alivun.odps.examples.mr.test	22	private final static String OUTPUT	Copy as Plain Text	
> C WordCountTest	23	private JobConf job;	Copy Reference	Ctrl+Alt+Shift+C
> com.aliyun.odps.examples.udf	24		I <u>F</u> aste	Ctrl+V
> com.aliyun.odps.examples.udf.test	25	public WordCountTest() throws Except	Paste from History	Ctrl+Shift+V
> 🖿 com.aliyun.odps.examples.unstructured	26	IestUtil.init#arehouse(); //\中告在北部墨	Paste Simple	Ctrl+Alt+Shift+V
> 🖿 com.aliyun.odps.examples.unstructured.test	28	iob = new JobConf():	Column Selection Mode	Alt+Shift+Insert
> 🖿 data.ambulance_csv	29	,	Find <u>U</u> sages	Alt+F7
> 🛅 data.speech_wav	30	job. setNapperClass (WordCount. Tokeni	Refactor	>
> m src	31	job. setCombinerClass (WordCount. SunC	Folding	>
> 🖿 target	32	job. setReducerClass (WordCount. SumRe	Analyze	>
MyFristModule.iml	33		Search with Google	
m pom.xml	34	job.setWapOutputKeySchema(SchemaUt)	Go To	>
scripts	36	Job. settapourpurvaruesenema (Senema	Generate	Alt+Insert
> 🖿 .idea	37	InputUtils. addTable(IableInfo. builde	Recompile 'WordCountTest.java'	Ctrl+Shift+F9
쪰MyFirstSQLosql	38	OutputUtils. addTable(TableInfo. build	Run 'WordCountTest'	Ctrl+Shift+F10
target	39 6	3 }	₩ <u>D</u> ebug 'WordCountTest'	
✓ Warehouse	40		Run 'WordCountTest' with Coverage	je
example_project	41	@SuppressWarnings ("deprecation")	Save 'WordCountTest'	
>resources	42	public woid testMan() throws TOFree	Show in Explorer	
tables	44	MapUIContext mapContext = new MapUI	Open in terminal	
✓ ads_log	45	mapContext.setInputSchema(INPUT_SC	Local History	>
i _scnema	46	mapContext.setOutputSchema(001P07_	Compare with Cliphoard	
	47	///准备现试数据	File Encoding	
kmeans_in	48	Record record = mapContext.createIng	A constant	
Kmeans_out	49	record set (new Text[] {new Text (st	Create Gist	

- ·本地运行MR:本地运行时,需要指定运行数据源,有两种方式设定测试数据源:
 - Studio通过Tunnel服务自动下载指定MaxCompute Project的表数据到warehouse目录
 下。默认下载100条,如需更多数据测试,请自行使用Console的Tunnel命令或者Studio的表下载功能。
 - 提供mock项目(example_project)及表数据,用户可参考warehouse下example_pr oject自行设置。
 - 1. 运行MR:在Driver类上右键,点击运行菜单,弹出run configuration对话框,配置MR需要在哪个MaxCompute Project上运行即可。

및 Run/Debug Configurations + - 대 비 양 수 »	Name: WordCountTest	ShareSingle_instance c
✓ ● JUnit	Configuration Code Cove Iest kind: Class Class: com.alin	rage Logs vage Logs Fork mode: none Repeat: Once 1
WordCountTest MaxCompute Java MaxCompute SQL	VM options:	-ea
> 🐕 Defaults	Program a <u>r</u> guments: <u>W</u> orking directory:	%MODULE_WORKING_DIR%
	Environment variables:	
	JRE: Shorten command line:	Default (<no jre=""> - module not specified) v user-local default: none - java [options] classname [args] v</no>
	 <u>B</u>efore launch: Build, Activate 	e tool window
	+ - ∕ ↑ ↓ ↓ ⁰¹ ₀₁ Build	
	☐ Show this page ☑ Activa	ite tool window
0	Warning: No JDK specifie	ed for module 'MyFristModule' OK Cancel Apply

2. 单击OK,如果指定MaxCompute Project的表数据未被下载到warehouse中,则首先下载数据;如果采用mock项目或已被下载则跳过。MR Local Run框架会读取warehouse中

指定表的数据作为MR的输入,开始本地运行MR。用户可以在控制台看到日志输出和结果打印。



生产运行MR

本地调试通过后,把MR发布到服务端,在MaxCompute分布式环境下运行MR。

- 1. 首先,将自己的MR程序打成jar包,并发布到服务端。详细操作步骤请参见打包、上传和注册。
- 通过MaxCompute Console(无缝集成于Studio),打开Project Explorer Window,右键 单击project,选择Open in Console,可在Console命令行中输入类似如下的jar命令。

jar-libjars wordcount.jar -classpath D:\odps\clt\wordcount.jar com. aliyun.odps.examples.mr.WordCount wc_in wc_out;

详细命令输入参请见jar命令。

2.6.4 非结构化开发

MaxCompute2.0新增了一套<mark>非结构化数据处理框架</mark>,支持通过外部表的方式直接访问OSS、OTS等。Studio对此提供了一些代码模板支持,方便用户快速开发。

编写StorageHandler/Extractor/Outputter

- 创建MaxCompute Java Module(在examples目录下的unstructured文件夹有示例代码供参考)。
- 2. 在module的源码目录即src > main上右键new > java,选择MaxCompute Java。
- 3. 输入包名.类名,如myun.MyExtractor,选择类型Extractor,点击OK。

U Create	new MaxCompute java class	23
<u>N</u> ame:	myun.Extractor] †↓
<u>K</u> ind:	E Extractor	\sim
	OK Cano	el

4. 模板已自动填充框架代码,只需要编写自己的逻辑代码即可。

5. 类似上述步骤可分别完成Outputter和StorageHandler的编写。

单元测试

可参考examples目录下的例子编写unit test,测试自己的Extractor/Outputter。

🗊 Project 👻 😌 😤 👫	C ExtractorTest.java ×
studioTest ~/IdeaProjects/studioTest	ExtractorTest ambulanceFullSchema
▶ Idea	3 import
Jdev	18
examples	19 G public class ExtractorTest {
com.aliyun.odps.examples	20 private String ambulance FullSchema =
🕨 🖿 mr	21 venicle:bigint;id:bigint;patient:bigint;calls:bigint;latitude:oc
▶ C∎ udf	22 private String speechbataruttSchema = Sentence_sin: doubte; id:string
	24 GTest
	25 G public void testTextExtractor() throws Exception {
	26 /**
C & Extractor lest	27 * Equivalent to the following SQL:
C 🔓 OutputerTest	28 CREATE EXTERNAL TABLE ambulance_data_external
🕨 💿 SpeechSentenceSnrExtractor.java	29 (vehicle bigint, id bigint, patient bigint, calls bigint,
💿 🔓 SpeechStorageHandler	30 Latitude double, Longitude double, time string, direction string)
c 🔓 TextExtractor	31 STORED BT COM. allyun. oups. uur. example. text. TextStoragenanuter
C & TextOutputer	33 USING 'iar file name.jar':
	34
	35 SELECT * FROM ambulance_data_external;
	36 */
data	37 Column[] externalTableSchema = UnstructuredUtils.parseSchemaString

打包上传

StorageHandler/Extractor/Outputter写好后,可以参考打包发布将已写好的java程序打成jar包,并作为resource上传到服务端。

创建外部表



1. 在scripts目录右键new > MaxCompute Script。

2. 输入SQL脚本名, Target Project选择脚本将要在哪个MaxCompute project下执行, 点 击OK。

3. editor中唤出create external table live template, 能快速插入创建外部表脚本模板:

te	ē P	roje	ect 🔻	⊕ ≑ ≱- ⊮	C E	xtractorTest	java⇒	MySecondSQL.osql	×		
: Proj	~ 🗸	Му	firstProject C:\Users\Administrate	or\IdeaProjects\	м	单句模式	t ~	MaxCompute类型系统 ~	默认编译器 ~	∫j ↓‼	
	> ~		.idea MyFristModule		1	name:/	lySeco:	ndSQL			
ē		>	examples		3	create	e time	:2018-12-10 16:58			
plo		\sim	src		4	с					
1 U U			> 🖿 main			cr				reate role	
<u>G</u> i			> 🖿 test			<i>P</i> create	(Key	word)			
\$		>	🖿 target			PCREATE	DATAR	BASE (Keyword)			
			MyFristModule.iml			CREATE	DIM 1	TABLE (Keyword)			
ē			<i>m</i> pom.xml			CREATE	DIMEI	NSION TABLE (Keyword)			
娩	~		scripts			CREATE	EXTER	RNAL TABLE (Keyword)			
e P		>	idea			<i>₿</i> create	FUNCI	(Keyword)			
,			🕾 MyFirstSQL.osql			<i>₿</i> create	INDEX	Keyword)			
			MySecondSQL.osql			<i>P</i> create	PACK	AGE (Keyword)			
			target			CREATE	RESUI	LT TABLE (Keyword)			
	>		warehouse			CREATE	ROLE	(Keyword)			
		1	MyfirstProject.iml			Ctrl+向下箭头	and Ct	trl+向上箭头 will move caret dow	n and up in the edito	or <u>>></u>	Α

然后修改外部表名称、列及类型、StorageHanlder类路径、配置参数、外部路径、jar名等,修改完成后点击运行脚本,创建该外部表。

ee osd.c	osql × Run MaxCompute SQL Script: "osd" (企業S)
	单句模式 ▼ MaxCompute类型系统 ▼ 默认编译器 ▼ ↓ ↓ ↓
1	name:osd
2	author:liuyi
3	create time:2017-03-07 17:06
4	
5	<pre>set odps.service.mode=off;</pre>
6	CREATE EXTERNAL TABLE IF NOT EXISTS myun_src_external
7	
8	vehicleId bigint ,
9	recordId bigint ,
10	patientId bigint,
11	calls bigint ,
12	locationLatitute double,
13	locationLongtitue double,
14	recordTime string,
15	direction string
16)
17	STORED BY 'myun.MyStorageHandler'
18	WITH SERDEPROPERTIES('delimiter'=' ')
19	LOCATION 'oss://oss-cn-hangzhou-zmt.aliyuncs.com/074799/demo/SampleData/CSV/src/'
20	USING 'myun.jar';
21	

4. 接下来,就可以查询该外部表了,类似:

ser osq	.osql × 興 a.osql × Run MaxCompute SQL Script: "a" (企業S)
	单句模式 ▼ MaxCompute类型系统 ▼ 默认编译器 ▼ ↓ ↓ ↓
1	<pre>set odps.task.major.version=unstructured_data;</pre>
2	<pre>select * from myun_src_external where patientId > 25;</pre>

2.6.5 开发Graph

创建完成 MaxCompute Java Module后,即可以开始开发Graph了。

代码示例

在examples目录下有graph的一些代码示例,可参考示例熟悉Graph程序的结构。



编写Graph

- 1. 在module的源码目录即src > main > java上右键new,选择MaxCompute Java。
- 2. 选择GraphLoader/Vertex等类型,Name框中输入类名(支持包名.类名)。点击OK,模板会自动填充框架代码,可在此基础上继续修改。



本地调试Graph

Graph开发好后,下一步就是要测试自己的代码,看是否符合预期。我们支持本地运行Graph,具体的:

1. 运行Graph: 在驱动类(有main函数且调用GraphJob.run方法)上右键,点击运行菜单,弹 出run configuration对话框,配置Graph需要在哪个MaxCompute Project上运行即可。



2. 点击OK,如果指定MaxCompute project的表数据未被下载到warehouse中,则首先 下载数据;如果采用mock项目或已被下载则跳过。接下来,graph local run框架会读 取warehouse中指定表的数据作为输入,开始本地运行Graph,用户可以在控制台看到日志输 出。每运行一次本地调试,都会在Intellij工程目录下新建一个临时目录,见下图:

🗊 Project 👻 😌 崇 🌾 🗠	C PageRank.java ×
✓ sandbox ~//deaProjects/sandbox ► i.dea Jeaph Jout Scripts Temp ✓ graph_20180322144903_634_1476 Fusbigling Fusbiglin	<pre> 2</pre>
Run N PageRank Eds: Start to write table scheme : exam 日 2, 2018 2:49:03 下午 com.aliyun.odg 信息: generate schema file: /Users/Liuyi 日 2, 2018 2:49:03 下午 com.aliyun.odg 信息: Finished to write table scheme : e 日 20 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	<pre>uple_project.pagerank_out>/Users/liuy1/IdeaProjects/sandbox/temp/graph_20180322144903_634_1476/output/_default os.local.common.utils.SchemaUtils generateSchemaFile /IdeaProjects/sandbox/temp/graph_20180322144903_634_1476/output/_default_/_schema_ os.local.common.utils.SchemaUtils generateSchemaFile xample_project.pagerank_out>/Users/liuy1/IdeaProjects/sandbox/temp/graph_20180322144903_634_1476/output/_default os.graph.local.mmaster.Master init</pre>
道 说明:	

关于warehouse的详细介绍请参考开发UDF中本地warehouse目录部分。

生产运行Graph

本地调试通过后,接下来就可以把Graph发布到服务端,在MaxCompute分布式环境下运行了:

- 1. 首先,将自己的Graph程序打成jar包,并发布到服务端。如何打包发布?
- 2. 通过Studio无缝集成的MaxCompute Console(在Project Explorer Window的Project上 右键,选择Open in Console),在Console命令行中输入类似如下的 *jar* 命令:

jar -libjars xxx.jar -classpath /Users/home/xxx.jar com.aliyun.odps. graph.examples.PageRank pagerank_in pagerank_out;

更详细的Graph开发介绍请参见编写Graph#可选#。

2.6.6 打包、上传和注册

完成UDF或 MR开发后,需要打包发布到MaxCompute系统。

打包UDF或MR

一个UDF或MR要想发布到服务端供生产使用,要经历打包>上传>注册三个步骤。针对此,我
们提供了一键发布功能(studio会依次执行mvn clean package,上传jar和注册UDF三个
步骤,一次完成)。具体的,在UDF或MR类上右键(该类必须在src>main>java子目录
下,maven module里编译成功),选择Deploy to server…菜单,会弹出如下对话框,选择要部
署的MaxCompute project,输入资源名和函数名,点击ok等待后台完成即可。

₽ Package a jar and s	ubmit resource	83
*MaxCompute project	t: MySecondProject2 (service.odps.aliyun.com)	~ +
*Resource name: MyF	ristModule-1.0-SNAPSHOT.jar	
*Main class: MyFirstU	DF	
*Function name:		
✓ Force update if al	ready exists	
0	ОК	Cancel

📕 说明:

当然,如果你有特殊的打包需求,那么你可以自行修改pom.xml打包相关配置。打好包后,再依次手工操作如下的上传jar和注册UDF即可。

上传jar

打包成功后,就可以将该jar包上传到MaxComptute服务端:

1. 在MaxCompute菜单选择Add Resource菜单项。



2. 选择要上传到哪个MaxCompute project上, jar包路径, 要注册的资源名, 以及当资源或函数 已存在时是否强制更新, 然后点击OK。

P Add Resource
*MaxCompute project: MySe + 2 (service.odps.aliyun.com) V
*Resource file: /MyfirstProject/MyFristModule/src/main/java/MyFirstUDF.java
*Resource name: MyFirstUDF.java
✓ Force update if already exists
OK Cancel

3. 上传成功后,可以在project explorer窗口的resources节点下看到该资源。



注册UDF

jar包上传完成后,就可以注册UDF函数了。

1. 在MaxCompute菜单选择Create Function菜单项。



2. 选择需要使用的资源jar、jar的主类,输入函数名,然后点击OK。

📱 Create Function
*MaxCompute project: MySecondProject2 (service.odps.aliyun.com) v +
*Function name:
mylower.jar
MyModule-1.0-SNAPSHOT.jar
*Using resources:
*Main class: MyFirstUDF ~
✓ Force update if already exists
function name can't be empty
OK Cancel

3. 注册成功后,可以在project explorer窗口的functions节点下看到该函数。



使用UDF

・接下来就可以在SQL中使用新编写的UDF完成后续开发。

G ↓	<u>n</u>	ieta_de							
name:m									
author:Liuyi									
set odos.sql.nlanner.narser.odos2 = true:									
-set odps.sql.planner.mode = "lot";									
set odps.sql.preparse.odps2 = "lot";									
set odps.compiler.output.format = "lot,pot";									
set odps.sql.runtime.mode = "executionengine";									
ELECT P	transmit with ()								
P									
PADSE UPL (STRING) (ODF FUNCTION)	STRING								
PARSE_URL(STRING, STRING) (Builtin Function)	DOURLE								
PERCENT_RANK() (Builtin Function)	DOUBLE								
PI() (Builtin Function)	DOUBLE								
PHOD(DECIMAL, DECIMAL) (Builtin Function)	DECIMAL								
POSITIVE(T) (Builtin Function)	DOURIE								
POW(DOUBLE, DOUBLE) (Builtin Function)	DOUBLE								
POWER(DECIMAL, DECIMAL) (Builtin Function)	DECIMAL								
PRINTE(STRING ANY) USUALTAN PUNCTIONA	STRING								

2.7 开发Python程序

2.7.1 Python开发使用须知

MaxCompute Studio支持您在Intellij IDEA中完成Python相关的开发,包括UDF和PyODPS脚本,但使用前必须安装Python、PyODPS和IDEA的Python插件。

在开始Python项目开发前,我们需要下载相关版本的Python配置Python环境变量。本文使用的 Python版本为Python2.7。

安装PyODPS

PyODPS是MaxCompute的PyODPS SDK,详情请参见Python SDK。

在Windows环境下,请务必在File > Settins > MaxCompute Studio保证您的Python安装路径 设置正确,用于解析Python UDF,举例如下。

Partings	straggy (highes, DOC) - Insels (DOA	1		X
Q.	MaxCompute Studio			
Appearance & Behavior Keymap	General Settings			
► Editor	Local meta store location:			
Plugins Version Control 	Table preview rows:		(max preview 500	
Build, Execution, Deployment	本地作业保存目录:	C:\Users\odps.studio\job		
 Languages & Frameworks Tools 	Python path to resolve UDF:	C:\Python27\python.exe		
▼ MaxCompute Studio				
SDK & Console				
MaxCompute SQL				
MaxCompute Job				
Accounts				
	Updates Checking			
	 Automatically checks for r 	new version		
	Last checked:			
	Current version:		3.0.3	
	Latest version:			
		Check new versions		
			ок	

安装Python插件

在Intellij IDEA的插件仓库中搜索Python或者Python Community Edition插件并安装。



配置Python依赖

配置Studio Module对Python的依赖,即可进行MaxCompute Python的开发。

1. 导航至File > Project Structure, 添加Python SDK。

Project Structure			8
⇔ ⇒	+ -	Name: Python 2.7	
Project Settings	1.8		
Project	Kotlin SDK	Python SDK home path: C:\Python27\python.exe	
Modules	Python 2.7	Classpath Packages	
Libraries		C:\Python27\DLLs	
Facets		C:\Python27\Lib	
Artifacts		C:\Python27\Lib\lib-tk	_
Platform Settings		C:\Python27	
SDKs		C:\Python2/\Lib\site-packages	
Global Libraries		C:\Users\Administrator\.IdeaIC2018.1\config\plugins\python_stabs\-72214762	
	-	C:\Users\Administrator\.IdeaIC2018.1\config\plugins\python-ce\helpers\typeshed\stdlib\2	
Problems		C:\Users\Administrator\.IdeaIC2018.1\config\plugins\python-ce\helpers\typeshed\stdlib\2and3	
		C:\Users\Administrator\.IdeaIC2018.1\config\plugins\python-ce\helpers\typeshed\third_party\2	
		C:\Users\Administrator\.IdealC2018.1\config\plugins\python-ce\helpers\typeshed\third_party\2and3	
0		OK Cancel App	ly

2. 导航至File > Project Structure, 添加Python Facets。



Project Structure				X
◆ ⇒	+ - 1	Python Interpreter:	Python 2.7 (MySecondProject) C:\Python27\python.exe	×
Project Settings	✓ MyFirstModule			
Project	Python			
Modules	MySecondProject			
Libraries	Python			
Facets				
Artifacts				
Platform Settings				
SDKs				
Global Libraries				

3. 导航至File > Project Structure, 配置Module依赖Python Facets。

2.7.2 开发Python UDF

MaxCompute Studio支持Python UDF开发,但需要您做好准备工作。

开发

1. 右键单击New > MaxCompute Python。



如果没有MaxCompute Python选项说明没有Python插件,请确认是否安装成功。配置和安装过程请参见Python开发使用须知。

Pro	×	V	My	firstProj	ect C:\Users\Administrator\IdeaProjects\M				Proj	ect	SDK is not defined	
		>		.idea					19			
	> MyFristModule 20								≫	<pre>public class ExtractorTest {</pre>		
orer		~		scripts		New			>	С	Java Class	
1 dx			>	idea.	\vee	Cut		Ctr	d L V	1	Kotlin File/Class	
gt				門 MyFin	000 6	Conv		Ctr		싙	File	
Proj				曾 MySe	1	Copy Path	Ctr	l∓shif	i+C	4	Scratch File Ctrl+Alt+Shift+Ins	ert
M				target		Copy Path Copy Reference	Ctrl+Alt	++Shif	t+C	b	Package	
		>	4	warehou	r i	Paste	CarrAn	Ctr	d+V	Dı	Python Package	
lorei		uh.		MytirstPr		Cad Use see				4	FXML File	
Е Б	Ľ		Car	ernal Libr		Find <u>U</u> sages	Ctu	Alt	+F/	4II J	package-info.java	
q	ľ		SCR	atches an		Poplace in Dath	Ctr	li chit	11.+F	4II J	module-info.java	
N						Analyze	Cu	17 SHIII	(TR)	Þ	MaxCompute Python	
						Pafastar.			1	2	MaxCompute SQL 脚本	
						Refactor	:			5	MaxCompute 交互式查询	
						Clean Python Compiled P	lies			8	从 D2 加载	
						Add to Favorites			>	í,	Python File	
						Show Image Thumbnails	Ctr	rl+Shit	t+T	.⊿ IPIy1	Jupyter Notebook	
						<u>R</u> eformat Code	C	Ctrl+A	lt+L	⊿ ∎ H	HTML File	
						Optimi <u>z</u> e Imports	C	trl+Al	t+O	4II J	JavaFXApplication	
						<u>D</u> elete		De	lete	J	Singleton	
						Build Module 'MyfirstPro	ject'			G	Gradle Kotlin DSL Build Script	
						R <u>e</u> build ' <default>'</default>	Ctrl	+Shift	+F9	G	Gradle Kotlin DSL Settings	
						Show in Explorer				o	XSLT Stylesheet	
an					2.	Open in terminal					Edit File Templates	

2. 输入类名,如Hello,选择类型,此处选择Python UDF。填写完成后单击OK。

P Create	new MaxCompute python class	8
<u>N</u> ame:	Hello] †↓
<u>K</u> ind:	🚾 python UDF	\sim
	OK Cano	el :

3. 模板已自动填充框架代码,您只需要编写UDF的入参出参,以及函数逻辑。

i 🍓 H	łello.py ×
1	from odps. udf import annotate
2	
3	
4	@annotate("bigint, bigint=>bigint")
5	class Hello(object):
6	
7	<pre>def evaluate(self, arg0, arg1):</pre>
8	if None in (arg0, arg1):
9	return None
10	P P return arg0+arg1
11	

测试

UDF开发完成后,需要测试自己的代码,看是否符合预期。我们支持下载表的部分sample数据到本地运行,进行DeBug,操作如下:

1. 右键单击Editor中的UDF类,单击RUN,弹出Run/Debug Configurations对话框。



UDF|UDAF|UDTF一般作用于Select子句中表的某些列,此处需配置MaxCompute project、table和columns,元数据来源于project explorer窗口和warehouse下的Mock 项目。

ч	Run/Debug Configurations		ß					
	+ — 👔 🗄 🐓 ↑ + 🖿 ↓3	Name: Hello	<u>Share</u> Single instance only					
	IUnit	Configuration Logs						
	MaxCompute Python	*Script:	C:\Users\Administrator\IdeaProjects\MyfirstProject\scripts\Hello.py					
	Hello	Script Parameter:	۳					
	9 Defaults	*MaxCompute project:	local					
		*MaxCompute table:	wc_in1 ~					
		*Table columns:	col1 ie:c1,c2					
		Download Record limit	t: 100					
		 Environment Environment variables: Python Interpreter: 	PYTHONUNBUFFERED=1 Image: SDK of module: Image: SDK of module: Image: Omega: Discussion of the second secon					
		Interpreter <u>o</u> ptions:	κ ²					
		Working directory:	C:\Users\Administrator\IdeaProjects\MyfirstProject\scripts					
		Add content roots to	PYTHONPATH					
		Add source roots to	PYTHONPATH					
		 ▶ Before launch: Activate tool window + - ▲ There are no tasks to run before launch □ Show this page Activate tool window 						
			OK Cancel Apply					

2. 单击OK后,通过Tunnel自动下载您指定表的sample数据到本地warehouse目录。

📕 说明:

- ·如果已经下载过,则不会再次重复下载,否则利用Tunnel服务下载数据。
- ·默认下载100条,如果需要更多数据测试,请自行使用console的Tunnel命令或者Studio的表下载功能。

 下载完成后,您可以在warehouse目录看到下载的sample数据。您也可以使用Mock data(即warehouse中的数据自己mock,详情请参见开发和调试UDF中的#####warehouse ##模块。

Project ▼ ⊕ ≑ ♣ ↓ I ^{+−}	👘 Hello	o.py × 🔮 data ×
 mexample_project 	1	399200, H. N. Harritan adapt is up
resources	2	399(50), 1. (headed)), headle (
> 🖿 table_resource1	3	382 m M. Terlil comments
> table_resource2	4	382148, S. I. Maintler Banart, a
a file_resource.txt	5	106 Th. W. T. Assays. Prime, bdcd
kmeans centers	6	906 HI Mini III. 2. Martine. Planetarium, platium
	7	351
✓ ■_tables_		
✓ ■ ads_log		
ੂschema		
🗐 data		

4. 本地运行框架会根据您指定的列,获取data文件中指定列的数据,调用UDF本地运行。

📕 说明:

本地运行是通过Pyodps的pyou脚本实现的,命令如pyou hello.Plus<data。安装 完pyodps后可以使用相应的命令检查该脚本是否存在。

- ·如果您是windows系统,请运行\${python}/../Scripts/pyou。
- ·如果您是mac系统,请运行\${python}/../pyou。

您可以在控制台看到结果打印,也可以在UDF上打断点调试。

🗊 Project 🔹 😌 寺 🕸・ 🖿	😼 hello.py ×
 studioTest ~/IdeaProjects/studioTest idea cripts hello.py target warehouse studioTest.iml External Libraries 	<pre>1 from odps.udf import annotate 2 3 4 @annotate("bigint.bigint->bigint") 5 Oclass Plus(object): 6 def evaluate(self, arg0, arg1): self: <hello.plus 0x113530f90="" at="" object=""> arg0: 1 arg1: None 7 if None in (arg0, arg1): 8 9 return None 9 return None 9 10</hello.plus></pre>
Debug 🚔 hello	
😭 Debugger 🗷 Console 📲 🛅 💆 🎽 🎽	差 物 坂 ■
Frames	→* 🗮 Variables
II MainThread II Image: second s	

注册发布

Python UDF测试通过后,即可注册发布到生产上进行使用。Add Resource后, Create Function即可。详情请参见<u>打包、上传和注册</u>。



2.7.3 开发PyODPS脚本

PyODPS是MaxCompute Python版本的SDK, 它提供了对MaxCompute对象的基本操作,并 提供了DataFrame框架,您可以在MaxCompute上进行数据分析。

开发PyODPS脚本

1. 使用PyODPS开发脚本,首先需要安装PyODPS及Python插件,安装完成后即可直接新建一

个MaxCompute PyODPS脚本。

Proj	2 V MyfirstProject C:\Users\Administrator\IdeaProjects\M							C:\Users\Administrator\Io	Pro	Project SDK is not defined			
E E			>		∣.idea						19		
	L		>		MyFr	istM	lod	lule			20	⊳	<pre>public class ExtractorTest {</pre>
ē			\sim		script	s							lava Class
l e				>	ic.ic	lea		New					Java class
L Å					₽M	yFir	Ж	Cu <u>t</u>		Ct	rl+X		Kotiin File/Class
je.					Μ	ySe	Ē	<u>С</u> ору		Ct	rl+C		File
4				h	targe	t		C <u>o</u> py Path		Ctrl+Shi	ft+C	Ð	Scratch File Ctrl+Alt+Shift+Insert
N N			>	Ŀ.	ware	nou		Copy Reference	Ctrl	+Alt+Shi	ft+C		Package
5	L			4	Myfir	stPr	ĥ	<u>P</u> aste		Ct	rl+V		Python Package
- je		>	uh	Ex	ternal	Libr		Find Usages		A	t+F7	<>	FXML File
<u>ا</u>	,	>	60	Sc	ratche	sar		Find in Path		Ctrl+Sh	ift+F		package-info.java
힉	ľ							Replace in Path		Ctrl+Shi	ft+R	J	module-info.java
N								Analyze		ourrom	>	ද	MaxCompute Python
	L							Refector			>	5	MaxCompute SQL 脚本
	L							Clean Puthon Compiled F	iler		-	5	MaxCompute 交互式查询
	L							clean Fython Compiled P	lies			5	从 D2 加载
	L							Add to Favorites			>	1 🍋	Python File
	L							Show Image Thumbnails		Ctrl+Shi	ift+T	d PIy1	Jupyter Notebook
	L							<u>R</u> eformat Code		Ctrl+A	Alt+L	 H	HTML File
	L							Optimi <u>z</u> e Imports		Ctrl+A	lt+O	ᆀ	JavaFXApplication
	L							<u>D</u> elete		De	elete	ᆀ	Singleton
	L							Build Module 'MyfirstPro	ject'			G	Gradle Kotlin DSL Build Script
	L							R <u>e</u> build ' <default>'</default>	-	Ctrl+Shif	t+F9	G	Gradle Kotlin DSL Settings
	L							Show in Explorer					XSLT Stylesheet
2								Open in terminal					Edit File Templates
1 3													
ĺ .		_											1
Ľ	-	C	re	ate	e new	Ma	nx(compute python class	5		X		
	N	Ja	m								† I		
	-	d									1.4		

Kind:

python UDF
 python UDF
 python UDAF
 python UDTF
 PyODPS Script

2. 新建成功后,模板会通过PyODPS room对象,自动初始化odps和o这两个对象。



在公共云Dataworks上运行时会自动在后台创建,所以为了IDEA编译通过,需要显式创建。



2.8 管理MaxCompute作业

2.8.1 作业浏览

通过Studio可以方便查看当前用户提交的MaxCompute的运行实例,包括运行状态、作业类型、 起止时间等。

打开Job Explorer

如果Job Explorer View没有在左侧Dock上显示,可以通过View > Tool Windows > Job Explorer 打开。

IJ	MyfirstProject [C:\Users\Administrator\IdeaProjects\MyfirstProject]\scripts\HelloTes											
Eile	e <u>I</u>	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	Analy <u>z</u> e	<u>R</u> efactor	<u>B</u> uil	d R <u>u</u> n	<u>T</u> ools	VC <u>S</u>	<u>W</u> indo
_	м	yfirs	Ī	ool Window	s		2		Project			Alt+1
	5		Q	uic <u>k</u> Definiti	on	C	trl+Shift+I	*	Favorite	es -		Alt+2
ojec		P Pro	Sł	now Siblings	;				Run			Alt+4
L D	ľ	V I	Q	uick <u>D</u> ocum	entatior	ı	Ctrl+Q	₩£	Debug			Alt+5
		>	<u>P</u> a	arameter Inf	fo		Ctrl+P	2	TODO			Alt+6
		>	<u>E</u> x	pression Ty	/pe	Ct	trl+Shift+P	5	Structur	e		Alt+7
orer		~	<u>C</u> c	ontext Info			Alt+Q	V	Version	Control		Alt+9
l 🛱			Re	ecent Files			Ctrl+E	∦	Ant Buil	d		
ช			Re	ecently Char	naed Fil	es Ct	trl+Shift+E	Q,	Capture	Analysi	s	
P.o.			Re	ecent Chano	ies	A	lt+Shift+C		Capture	e Tool		
M			* C		, h		Ctrl+D	- M	Console	9		
Ι.				mpare with	u Clinho	ard	CUITD		Designe	er		
			ipr Co	mpare wit	1 Clip <u>b</u> o	aru			Education	onal.Che	ckDet	ails
8			Q	uick Switch	Scheme		Ctrl+		Event Lo	og		
<u>ि</u> द		1	To	olbar				N	FAQ Ro	bot		
M		ulter	√ <u>⊺</u> o	ool Buttons				Y	Image l	ayers	_	
	Ľ		√ <u>S</u> t	atus Bar				W	Job Exp	olorer		
	ľ	8- 3	✓ N	a <u>v</u> igation Ba	ar			m	Maven	Projects		
			Ac	tive Editor			2	2	Palette			
			Bi	Bidi Text Direction			2	2	Palette			
			Er	nter Present	ation M	ode		N	Project	Explore		
			Er	nter Distract	ion Free	e Mode		2.	Termina	al	A	t+F12
			Er	nter Full Scre	een			۲	Theme	Preview		
									UI Desig	gner		

查看项目下所有作业实例

Job Explorer支持按照状态查询提交的作业列表,如下图所示,查看过去24小时内执行失败的作业:

ಕ	Joł	o Explorer								
Proj	∫∫ MySecondProject2 ∨		状态: 失败 ~	提交者:	任何人 ~	日期: 12/13	~ 现在 ~	用时:任意	i v	
- AI	Id与名称 提交者			全部作业		时间	总时长	完成	时间	
	8	.console_q	ALVUNIC	活动中	ı	09:42	00:00:02	12/14 00:0)9:44	
e.	8	.console_q	ALPRINKI	全部活动作	<u>V</u>	08:37	00:00:02	12/14 00:0)8:39	
檀	8	.console_q	console_q ALMUNE1			09:01	00:00:01	12/13 00:0)9:02	
Ĕ	8	.console_q	ALTRUNCS	● 排队中		07:56	00:00:02	12/13 00:0)7:58	
rojec				③ 运行中						
ů,				——— 历史作	<u> </u>	_				
-				全部历史作	44					
ē				🕑 成功						
음				😣 失败						
с Р Е				🕛 已取消						

单击日期筛选框,选择其他日期,拖动选择时间。如下图。

t	Jo	b Explorer								∳ - I+		
Proj	ſĵ	MySecondProject2 ∨	状态: 失败 ~	提交者:任何人 >	日期: 1	12/14 ~ 现在 ~ 用时	†:任意 ~					
A		Id与名称 提交者	类型	提交时间		日期。以	20	18-12-14	포미	现在		
-8	8	console_q	SQL	12/14 00:09:42		L170-7A		5:00:00	353	AULT		
b	8	console_q IIIIIII	SQL	12/14 00:08:37								
흘	8	console_q	SQL	12/13 00:09:01	1	I.	1	I	_	I		Y
roject Exp	8	console_q alm.m1	SQL	12/13 00:07:56		18-12-7 1周前		18-12-1 2天前	2	18-12-13 昨天	18-12-14 今天	现在
Ň					取消	ă						确认

点击刷新按钮获取作业列表。

〕 说明:

默认只显示符合条件的前1000条作业,若超过1000条的作业将无法显示,请考虑更新过滤条件。

作业列表排序

可以通过点击job列表的列名对job进行简单排序,如下图按提交时间倒序:

\$P. 1.
时:任意 ~
]
8
2
9
4

作业排队队列

活动状态作业如果正在排队队列中等待调度,作业列表中会展示当前排队的位置和全局优先级:

Job Explorer 🔅 🖓 - 👔 🗠										
dal	da									
	提交者	类型	Cpu用量	内存用量	优先级	队列位置	提交时间	配额组	集群	ld
3	F	SQL	12900/1	797736/	0	0	2017-12	27	A 38A	20171213
3.		SQL	431100/	8915148	3	0	2017-12	27	AY ISA	20171213
(ii)		SQL	//	//	0	1	2017-12	27	AY BA	20171213
8.		SQL	//	//	0		2017-12	27	AYSJA	20171213

🗾 说明:

在Running Instances Tab下的作业状态、队列位置等会自动更新,作业结束后会从列表中移除。

作业日志保存功能

目前job的logview日志默认保存7天,有些重要的logview希望能保留更长时间以便后期进行查看,可以将logview保存到本地:

双击列表中的job,在右侧打开job的详细信息,点击工具栏上的save按钮,将日志保存到本地:

ţ	Job	Explorer						÷. ⊩	a2018121316	0941836gpisz192 ×	a20181213160837553g6rpe62m ×
Proj	ſĵ	MySecond	Project2 🗸	状态: 失败 ~	提交者: 任何人 ~	日期: 12/13	- 现在 ~ 用时: 任意	~	<> • •	G 🗇 🛱 🖸 🗆	Enable auto refresh
		Id与名称	提交者	类型	提交时间	总时长	完成时间		属性名	值	执行图 送债 脚木 结果
- 0	8	console_q	ALIYUN\$1	SQL	12/14 00:09:42	00:00:02	12/14 00:09:44		作业 Id	2018121316094	A A D D D D D D D D D D D D D D D D D D
5	8	console_q	ALIYUN\$1	SQL	12/14 00:08:37	00:00:02	12/14 00:08:39		名称	console_query	MaxCompute task detail is empty !
힡	😣 .	console_q	ALIYUN\$1	SQL	12/13 00:09:01	00:00:01	12/13 00:09:02		创建人	ALPUPEIAR230	
Ä	₿.	console_q	ALIYUN\$1	SQL	12/13 00:07:56	00:00:02	12/13 00:07:58		状态	FAILED	
t									类型	SQL	
ō									开始时间	2018-12-14 00:	
-									结束时间	2018-12-14 00:	
*r									耗时(H:m:s)	00:00:02	
									 Service 		
orer									End Point	http://www.inn.o.u	
-dx									项目空间	Mylacondhroje	
e e											
9											
N											

若需要自定义保存文件的目录,可以在Studio的Setting配置项中进行配置:

U Settings				
Qr	MaxCompute Studio			
 > Appearance & Behavior Keymap > Editor Plugins > Version Control > Build, Execution, Deployment > Languages & Frameworks 	General Settings Local meta store location: Table preview rows: 本地作业保存目录: Python path to resolve UDF:	(max preview 5000 rows)		
> Tools				
SDK & Console				
MaxCompute SQL				
MaxCompute Job				
Accounts	Updates Checking			
	Automatically checks for ne	w version		
	Last checked:			
	Current version:			3.0.2
	Latest version:	Charle new versions		1
		Check new versions		
0				OK Cancel App

2.8.2 作业实例详情

查看作业实例

Studio支持两种方式查看MaxCompute作业实例。

1. 通过Logview URL或本地的logview离线文件以只读方式打开作业详情。

使用Logview来查看一个作业的详细信息是MaxCompute用户熟悉的方式。使用Logview还 有一个便利之处是可以查看其他用户在其他项目空间中提交的任务状态。Studio提供通过输入 一个有效的Logview URL打开任意一个作业详情的功能。

在菜单栏中,打开MaxCompute菜单下的Open Logview,即可自动将粘贴板中有效Logview URL地址复制到弹出窗口中或者选择导出在本地的logview离线文件。



2. 作业浏览信息中,选择某个MaxCompute实例,双击或右键Open,可查看该实例详细信息。

作业详情视图

作业详情页面包括顶部的工具栏,左半部分的基本属性栏以及右半部分详细视图页,其中详细视图 页主要包含四个视图:

- · 执行图: 以DAG图的方式显示作业整体信息,可查看子任务间依赖关系及各个子任务的详细执 行计划。
- · 时序图:显示作业执行timeline,可以从不同粒度查看执行的时序,并提供了多种过滤器。
- · 详情: 以table view方式展示作业详细信息,包括子任务列表,各子任务的worker列 表,worker处理数据量,执行时间及状态信息等。
- ·脚本:显示该作业提交时所对应的SQL语句以及提交作业的参数配置信息。
- · 概要(JSON): 以JSON格式显示作业运行详细信息。
- ·结果视图:显示该作业运行结果。
- ・分析:提供作业执行散点图、长尾柱状图及数据倾斜图。



工具栏

页面左右折叠

<

>

用来收起或完全展开左右侧视图,允许用户最大化某一个视图进行查看。

停止作业



用来中断正在执行的作业,需要具有响应权限才能停掉作业(owner或管理员)。

刷新详情

G

对于运行中的作业,它的基本信息,如状态、quota等会自动刷新,但右侧各个详情视图不会自动刷新,如果想要获取最新详情信息,需要手动刷新。

拷贝logview

ø

复制作业对应的logview到剪切板。

在浏览器打开作业详情

Ę)

生成logview URL,并通过浏览器打开。

作业详情保存到本地

Β

将作业详情信息保存为本地文件。

自动刷新

Enable auto refresh

对于运行中作业,允许自动刷新后,studio会对作业执行全量定时刷新。

基本信息页

展示作业基本信息,包括ID、Owner、状态、起止时间、计算资源用量,输入输出等。运行中作业的基本信息会自动定时刷新。输入,输出项列出了作业的输入表和输出表,双击表名可查看对应表详情。

执行图

执行图作为日常主要使用工具,以可视化的方式展示Fuxi Job, FuxiTask以及Operation的依赖 关系,同时提供一系列辅助工具,如作业回放,进度图,热度图等,绝对是排查问题的好帮手。

执行图可展示三个维度的作业依赖关系:Fuxi Job层,Fuxi Task层,Operation层,可点击面 包屑或向上箭头进行切换,默认会展示Fuxi Task层依赖关系。



非SQL类型作业,仅能展示Fuxi Job和Fuxi Task层作业,不支持Operation层。

sn26pr2 ×201810241031351gxcy66pr2 ×20181207072711675g3tr9qsa ×20181207072533729glodyv3 ×4
秋 12
MaxCompute job > JOB:_SQL_0_1_0 job_0 进度图▼ Succeeded Running Failed Interrupted Waiting 6
myseconaproject2.oas_user_info_a
M1 Instance Count: 0/1/1 I/O Records: 0/0 2018-12-07 15:25:38 2018-12-07 15:25:41 100%
5 my second project 2 o ds_user_into_d no ow MI instance Count: 0/1/11 ino Records: 0/0 2015-12-07 1525-38 2015-12-07 1525-34 100%

说明:

对应序号说明如下:

- 1. 可点击跳转其他层次
- 2. 缩放辅助工具
- 3. 依赖表
- 4. Fuxi Task节点
- 5. 鹰眼
- 6. 展示
- 7. 默认打开FuxiTask层依赖

・Fuxi Job层:

单击MaxCompute Job或在Fuxi Task层中点击向上箭头即可打开Fuxi Job层。Fuxi Job节点 内包含Fuxi Task名称,起止时间等。双击任一Fuxi Job节点即可进入Fuxi Task层。

执行图	时序图 详情 脚本 概要 (JSON) 结果 分析
MaxCompute job	•
	Fuxi Job 图不支持工具栏和回放功能
	Fuxi Job 依赖图
	JOB:_SQL_O_O_O_job_O TaskNodeList: [M1 R2] 2018-03-02 15:10:16 2018-03-02 15:10:18 JOB:_SQL_O_O_1_job_1 TaskNodeList: [M1 M2 J3] 2018-03-02 15:10:22 2018-03-02 15:10:22
	2018-03-02 15:10:26
	J08:_SQL_0_0_0_job_0 TasikNodeList: M1 R2] 2018-03-02 15:10:16 2018-03-02 15:10:18 J08:_SQL_0_0_1_job_1 TasikNodeList: [M1 M2 J3] 2018-03-02 15:10-22 2018-03-02 15:10-22 2018-03-02 15:10-22

・Fuxi Task层:

当有多个Fuxi Job时,默认打开最后一个Fuxi Job的Fuxi Task层。该层可展示Fuxi Task的依赖关系,输入输出表及分区等信息。当作业结束后点击工具栏中下拉框可切换进度图,输入热度 图,输出热度图,Task时间热度图,Instance热度图等。进度图表示节点的完成进度,热度图



通过颜色区分节点热度。双击任一Fuxi Task即可打开Operation层,右键可以展开所有Fuxi Task的Operation层。

Fuxi Task节点内容:

1.Instance Count: a/b/c,指某一时刻正在运行子任务实例个数为a,已结束任务实例 个数b,总任务实例个数c。

2.I/O records: 同理为某一时刻的input records和output records。

3.百分比与橙色进度条:表示该任务运行情况,该比例根据子任务运行实例分析得出。

4.子任务间连线上显示的是输出records数量。箭头表示数据流动方向。

· Operation层

Operation层揭示了Fuxi Task内在的运行方式,单击任一节点即可显示Operaiton完整信息。



作业回放

Studio支持作业回放功能,作业回放就像播放媒体文件一样,可在12s内回顾该Job执行的历史轨 迹。该功能主要用于帮助用户了解MaxCompute实例在不同时刻运行状态,快速判断子任务级运 行顺序及消耗时间,掌握Job执行关键路径,从而针对运行较慢的子任务进行优化。

- ・ 点击 > 按钮即可开始播放,再次点击则暂停。用户也可手动拖动进度条。
- ・ 进度条左边为作业开始时间,中间为播放时间,右边为结束时间。


▋ 说明:

回放功能仅通过时间估算某一个时刻IO数据量,从而确定完成进度,并不能代表该时刻真实IO数 据量。Running 状态作业不支持回放功能。

时序图

以甘特图的方式展示作业分布式执行的详细数据,可以调整展示粒度,将每一个计算节点都在甘特 图中展示。



可以通过甘特图直观的看出作业运行的时间瓶颈、长尾节点等。同时提供多种过滤器,能够直接筛 选出作业执行的关键路径、最大数据节点、最长时间节点等。

详情页

主要针对SQL DML类作业,展示作业在计算集群上的Fuxi Task列表、计算节点列表等

B: SQL01.0 job.0 1 日子名: mysecondproject2_20181207072711675g3tr9qsa_SQL_0_1_0_job_0 莫式: fuxi job CPU用量(core*min): 0.00 巨时(s): 2 内存用量(GB*min): 0.00 Task Input RecoOutput ReInput Bytes Output By Status Progress StartTime M1 0 0 256 INTERCONTINUE 100 2018-12-07 15:27:16 2 2 7 100 2018-12-07 15:27:16 100	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	B: _SQL_0.1.0.job_0 1 #S: SQL_0.1.0.job_0 1 #S: fuxi job CPU用量(core*min): 0.00 题d: fuxi job CPU用量(core*min): 0.00 Exist: fuxi job 1 0 0 0 2 Progress Status Progress 100 2018-12-07 15:27:16 2013 2 0 256 11 × X X 11 × Status Progress 1 Gatebook Gatebook 1 FinishedP.:: 0 1 FinishedP.:: 0 1 FinishedP.:: 0 1 The Status FinishedP.:: 3 1 Status FinishedP.:: 1 The Status FinishedP.:: Status 1 The Status FinishedP.:: Status	第:SQL_0_1_0_job_0 1 第:SQL_0_1_0_job_0 1 第:SQL_0_1_0_job_0 1 第:SQL_0_1_0_job_0 0 1:SQL_0 0	图时序图	详情 脚本 纷	吉果 概要 (文本) 概要 (J	ISON) 分析			
SA: mysecondproject2_20181207072711675g3tr9qsa_SQL_0_1_0_job_0 式: fuxi job CPU用量(core*min): 0.00 聞い; 2 内存用量(GB*min): 0.00 Task Input RecoOutput ReInput Bytes Output By Status Progress StartTime 1 0 0 0 256 THEMMINY 100 2018-12-07 15:27:16 2	econdproject2_20181207072711675g3tr9qsa_SQL_0_1_0.job_0 ob CPU用量(core*min): 0.00 内存用量(GB*min): 0.00 20.0000000000000000000000000000000	Ski: fuxi job CPU用量(core*min): 0.00 Ki(s): 2 内存用量(GB*min): 0.00 Task Input RecoOutput ReInput Bytes Output ByStatus Progress StartTime M1 0 0 0 256 「ビビバババン 100 2018-12-07 15:27:16 2014 Core 10 2018-12-07 15:27:16 Core 10 2018-12-07 Core 1	法: mysecondproject2_20181207072711675g3tr9qsa_SQL_0_1_0_job_0 氏: fuxi job CPU用量(core*min): 0.00 部(6): 2 内存用量(GB*min): 0.00 Task Input RecoOutput ReInput Bytes Output By M1 0 0 0 256 FFEMINA 100 2018-12-07 15:27:16 2019 2 1 × 数: 1 最短时间(s): 0 最长时间(s): 0 所有 ✓ Instance Input RecoOutput ReInput Bytes Output By Status FinishedP StartTime 2 M1#0_0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1	3: SQL 0 1	1 0 job 0 🚺					
EssA: mysecondproject2_2018120/07/2/110/5gstr9qsa_SQL_0_1_0_065_0 表t: fuxi job CPU用量(core*min): 0.00 表t: 100 0.00 Task Input RecoOutput ReInput Bytes Output By Status Progress StartTime M1 0 0 0 256 INEXAMINYAL 100 2018-12-07 15:27:16 2 //// ///// ///// 100 2018-12-07 15:27:16	ob CPU用量(core*min): 0.00 内存用量(GB*min): 0.00 :oOutput ReInput Bytes Output By Status Progress StartTime 0 0 256 INTERVIEW 100 2018-12-07 15:27:16 2013 1 最短时间(s): 0 最长时间(s): 0 所有 ✓	Ess: myseconoproject_2018120/0/2/110/33str9qsa_SQL_0_1_0_00_ 支: fuxi job CPU用量(core*min): 0.00 Ett: fuxi job CPU用量(GB*min): 0.00 Task Input RecoOutput ReInput Bytes/Output By Status Progress StartTime M1 0 0 0 256 Intrinsition 100 2018-12-07 15:27:16 2019 2 100 2018-12-07 15:27:16 2019 2 0 2018-12-07 15:27:16 2019 2 100 2018-12-07 15:27:16 2019 2 100 2018-12-07 15:27:16 2019 2 100 2018-12-07 15:27:16 2019 2 2 <td>1 × ************************************</td> <td>A .</td> <td></td> <td></td> <td>1675-24-0 001-0-1</td> <td></td> <td></td> <td></td>	1 × ************************************	A .			1675-24-0 001-0-1			
Example of on matter of	内存用量(GB*min): 0.00 coOutput ReInput Bytes Output By Status Progress StartTime 0 0 256 TEXTMINA 100 2018-12-07 15:27:16 2013 1 最短时间(s): 0 最长时间(s): 0 所有 ✓	1 × ////////////////////////////////////	1 ×	(중점) [국·	fuxi iob	oroject2_2018120707271	CPU田母(core*m	_0_job_0 in): 0.00		
Task Input RecoOutput Re Input Bytes Status Progress StartTime M1 0 0 0 256 TERMATINA 100 2018-12-07 15:27:16 2	Status Progress StartTime 0 0 256 TERMARY 100 2018-12-07 15:27:16 2013 1 最短时间(s): 0 最长时间(s): 0 原有 ✓	Task Input RecoOutput Re Input Bytes Output By Status Progress StartTime M1 0 0 256 II (INVALINAN) 100 2018-12-07 15:27:16 2018 2 Image: Start Time 100 2018-12-07 15:27:16 2018 1 × X X X X St: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	Task Input RecoOutput ReInput Bytes Output By Status Progress StartTime M1 0 0 0 256 THIMMINY 100 2018-12-07 15:27:16 2019 2 1 × 100 2018-12-07 15:27:16 2019 1 × × 100 2018-12-07 15:27:16 2019 2 × × × × × × 1 × × × × × × ½: 1 最短时间(s): 0 最长时间(s): 0 × 1 1 1 1 1 × × × × 1	。 (时(s):	2		内存用量(GB*min	0.00		
M1 0 0 0 256 <u>TIRRAMINA.</u> 100 2018-12-07 15:27:16 2	0 0 256 IERMAINA 100 2018-12-07 15:27:16 2013 1 最短时间(s): 0 最长时间(s): 0 0 所有 ✓	M1 0 0 0 256 TERM/INA 100 2018-12-07 15:27:16 2013 2	M1 0 0 256 HANAHIMA 100 2018-12-07 15:27:16 2013 2	Task I	_ Input RecoOu	tput Re Input Bytes C	output By Status	Progress	StartTime	
2	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 計时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 ×	M1	0	0 0	256 TERMINA	. 100 2	018-12-07 15:27:16	201
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 V	1 ×	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput ReInput Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3 2	2						
7222	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 ×	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime 3 M1#0_0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3	-						
7222	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime 3 M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × ************************************							
2222	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 V	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3 2							
2002	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 ×							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 計时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput ReInput Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 計时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 时时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	11 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	12 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	1 × 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3							
	1 最短时间(s): 0 最长时间(s): 0 0 平均时间(s): 0 所有 ✓	数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有	加 最短时间(s): 0 最长时间(s): 0 数: 1 最短时间(s): 0 最长时间(s): 0 计时间: 0 平均时间(s): 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3	1 ×						
	1 最短期間(5). 0 最低期間(5). 0 0 平均时间(5): 0 所有 ✓	ACCENTING(S): 0 単式式の10(S): 0 计时间: 0 平均时间(S): 0 所有 Instance Input RecoOutput ReInput Bytes Output By Status Finished P StartTime	Azer I Hackenble(s) O Hackenble(s) O 计时间: 0 平均时间(s) 0 所有 ✓ Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32	×	1	月初时间(4)	. 0	县长时间(4)。		1
	0 半时间(5): 0 所有 ~	Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 3	EX:	1	取起的间(S)	. 0	□ 取大的间(\$):	: 0	
计时间: 0 半均时间(s): 0 所有		Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime	Instance Input Reco Output Re Input Bytes Status FinishedP StartTime M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32	110447	0	半均时间(S)	: 0	所有		~
	RecoOutput Re Input Bytes Output By Status FinishedP StartTime		MI#0_0 0 0 0 256 TERMINA 100 2018-12-0/ 15:2/:18	计时间:						
Instance Input RecoOutput Re Input Bytes Output By Status FinishedP StartTime		M1#0_0 0 0 0 256 TERMINA 100 2018-12-07 15:27:18 32		计时间: Instance	Input Reco	Output Re Input Byte	s Output By Statu	IS FinishedP	StartTime	



对应序号说明如下:

- 1. Fuxi Job Tab
- 2. Fuxi Task列表
- 3. 每个Fuxi Task详细信息及计算节点列表

通常一个作业对应一个或多个Fuxi job,每个Fuxi job拆分成多个Fuxi task(阶段),每个Fuxi task包含多个Fuxi instance(worker)。

在每个Fuxi instance通过右键菜单可以查看作业运行的标准输出、标准错误,及Debug Info。

分析页

展示作业的长尾节点(worker)、数据倾斜节点(worker)。展示节点散点图、及柱状图,辅助作业 执行瓶颈诊断

			执行图 时序图 详情 脚本 概要 (JSON) 结果 分析
结论 散点	图 长尾图 数据倾斜	斜图	
作业な	分析结论		
	. L		
长尾节	点		数据倾斜节点
长尾节点是指* 间的2倍的节点	节点执行时间大于所属Task ā	(平均执行时	数据倾斜节点是指节点输入记录数大于所属Task平均 输入的2倍的节点
发现长尾节点	,点击所在行查看详情		未发现数据倾斜
Task Name	Instances Name	Latency	
M4_3	M4_3#238_0	638	
M1_6	M1_6#281_0	308	
		执行	7图 时序图 详情 脚本 概要 (JSON) 结果 分析
结论 散点图	长尾图 数据倾斜图		
分组		图表	
选择经制输入数	如用和动输出数据		
	ע אג אן מויא-איז דע		
VInput Data(•	•)	650 -	
	a(•)	600 -	
		550 -	
Tasks		500 -	
选择 Fuxi Task	在图标中显示	450 -	
		(s) +000 +	
全选		ecution	
全不选		¥ 300-	
M9		200 -	
M2		150 -	
R6_2		100 -	
R3_2		50 -	
M4_3			
M1_6		0	200000 400000 600000 1000000 1200000 1400000 1600000 2000000 2000000 I/O Data Size (KB)
J5_1_4			
R7_1			

散点图和柱状图支持从图中节点直接调准详情页查看Fuxi instance详情。

结果页

结果页会根据作业类型及提交作业时的参数设置展示不同页面。

- select语句并且设置odps.sql.select.output.format = HumanReadable, select以文本方 式展示
- · select语句并且未设置output format参数,结果以table方式展示
- · 对于数据输出到表的脚本,展示输出表名及到表详情的跳转链接
- ・ 对于异常作业,结果页限制异常详情

2.9 工具集成

2.9.1 与MaxCompute客户端集成

MaxCompute Studio集成了MaxCompute客户端程序,可在Studio中直接打开客户端。

配置客户端安装路径

 Studio中已包含最新版MaxCompute客户端程序,并指定为默认客户端。用户也可自行指定其 他版本客户端程序,单击file > Settings > MaxCompute Studio > SDK & Console,添加客 户端程序和路径。

🖳 Sett	tings	de los homencies tong	netra +	X
Q		MaxCompute Studio > Sl	DK & Console	Reset
► Ap	opearance & Behavior	MaxCompute Console Det	ails	
Ed	litor	Installed Location:		-
Plu	ugins	Version:	No valid Client installed in the specified path. Please check or download	
► Ve	ersion Control		· · · —	
► Bu	uild, Execution, Deployment			
► La	nguages & Frameworks			
► To	ools			
- M	axCompute Studio			
	SDK & Console			
	MaxCompute SQL			
	Accounts			
M	aven Helper			
			OK Cancel Apply H	elp

2. 设置成功后会显示MaxCompute客户端版本信息。

Settings	A DI UNI PLOTE CLASSIFICIA	without an in			23
٩	MaxCompute Studio > SDI	(& Console	•		Reset
Appearance & Behavior Keymap	MaxCompute Console Detail	s			
 Editor 	Installed Location:	D:\dev	elop\odpscmd_public		
Plugins	Version:	0.23.3			
Version Control			1		
Build, Execution, Deployment					
Languages & Frameworks					
► Tools					
MaxCompute Studio					
SDK & Console					
MaxCompute SQL					
Accounts					
Maven Helper					
			Г		
				OK Cancel Apply	Help
			L		

打开MaxCompute客户端

设置MaxCompute客户端安装路径成功后,就可以在Studio中打开客户端程序。

1. 在项目浏览列表中,选中要打开的项目,右键菜单中点击Open in Console。



2. 用户可以打开多个客户端程序,操作步骤同上。



2.10 配置选项

2.10.1 配置 MaxCompute Studio

安装 MaxCompute Studio 插件后,可以在 IntelliJ IDEA 的 Setting 页面的左边栏找到 MaxCompute Studio 的配置项。打开 IntelliJ 配置页面的方法请参考 IntelliJ 的_{文档}。

MaxCompute Studio 配置选项页

MaxCompute Studio 配置选项页提供以下配置项:

1. 本地元数据仓库存储路径

指定在本地存储 MaxCompute 项目空间元数据的路径。Studio 的缺省设置是本地用户目录下的.odps.studio/meta 这个隐含目录。

- 2. 版本更新选项
 - 复选框 Automatically checks for new version 可以控制 Studio 是否自动检查新版本更新
 - · 按钮 Check new versions 用于手动检查新版本。点击检查新版本按钮后,如果有新版本可 以更新,将显示 Install new version 按钮,点击可以安装,安装完成后需要重启 IntelliJ IDEA。
- SDK & Console 配置选项页

SDK & Console 配置选项页面提供以下配置项:

1. MaxCompute 客户端安装路径

指定本地安装的 MaxCompute 客户端的安装路径。 Studio 会检测路径中安装的 MaxCompute 客户端的版本,如果检测失败,会提示错误信息。

📔 说明:

Studio 2.6.1 之后的版本自带了最新的 MaxCompute 客户端,用户不用特别指定。 如果用户 希望使用自己特定版本的 MaxCompute客户端,可以指定路径。

MaxCompute SQL 配置选项页

MaxCompute SQL 配置选项页面提供以下配置项:

1. 启动语法高亮

勾选Enable syntax coloring 选项, 启动语法高亮功能。

2. 启动代码自动补全

勾选Enable code completion 选项,启动代码自动补全功能。

3. 启动代码格式化

勾选Enable code formatting 选项, 启动代码格式化功能。

4. 编译器选项

在这里这是全局缺省的编译器选项。 以下选项还可以在 SQL 编辑器的工具栏上为每个文件单独 设置。

- ·编译器模式(Compiler Mode)
 - 单句模式(Statement Mode): 在该模式下,编译器对 SQL 文件单条语句作为单元进 行编译、提交
 - 脚本模式(Script Mode):在该模式下,编译器对整个 SQL 文件作为单元进行编译、 提交。#注#脚本模式有利于编译器和优化器更大程度地优化执行计划#提高整体执行效 率#目前处于测试阶段#
- ・类型系统
 - 旧有类型系统(Legacy TypeSystem):原有 MaxCompute 的类型系统
 - MaxCompute 类型系统(MaxCompute TypeSystem): MaxCompute 2.0 引入的 新的类型系统
 - Hive 类型系统(Hive Compatible TypeSystem): MaxCompute 2.0 引入的 Hive 兼容模式下的类型系统
- ・编译器版本
 - 默认编译器(Default Version):默认版本的编译器,
 - 实验性编译器(Flighting Version):实验性的版本的编译器,包含正在测试中的编译器的新特性

Account 配置选项页

Account 配置选项页面提供添加和管理用户用于访问 MaxCompute 所用账户,参考 用户认证:

Studio 需要通过用户指定的账号访问 MaxCompute 的项目空间和执行提交作业等操作,目前 Studio 支持的账号类型有:

· 阿里云账号(AccessKey)

添加账户

- 在 Account 配置选项页面,执行以下步骤:
- 1. 点击按钮 + 或者快捷键 Ctrl-N
- 2. 选择账户类型 Aliyun Account by AccessKey

3. 在弹出的 Add Account 窗口中填入:

- · Account Name: 该账户在 Studio 中的标识名称。
- · Using properties file:从配置文件中读取 AccessKey ID 和 AccessKey Secret。
 - 选择一个在用户认证中 conf/odps_config.ini 示例的配置文件。
- · Using properties: 手动填入 AccessKey ID 和 AccessKey Secret。
 - Access Id: 填入用户阿里云账号的 AccessKey ID。
 - Access Key: 填入用户阿里云账号的 AccessKey Secret。

	Add MaxCompute project
	Connection Setting
Properties File:	用odps_config.ini文件来初始化AK/Endpoint等配置项
AK Account: 用AK	账号来初始化AK,可点击右侧+号添加AK Account 📀 🕂
* Access Id:	连接MaxCompute project 时的AK/Endpoit等配
* Access Key: <	置信息,必填。可手工填写,也可通过上述Properties File或AK Account来初始化配置
* Project Name:	
* End Point:	
Tunnel EndPoint:	
?	Cancel OK

4. 点击按钮 OK 完成添加。 添加完成后账号会出现在 Account 配置选项页面的 Account 列表 里。

删除账户

在 Account 配置选项页面,执行以下步骤(该操作仅在 Studio 配置中删除账户配置,对用户账户本身不产生影响):

1. 在 Account 列表中选择要删除的账户名称。

2. 点击按钮 -。

3. 在弹出的的确认对话框中,选择OK。

修改账户 AccessKey

在 Account 配置选项页面,执行以下步骤:

- 1. 在 Account 列表中选择要删除的账户名称。
- 2. 点击"铅笔"图标进行编辑。
- 3. 在弹出的 Edit Account 窗口中编辑 Account 信息,内容与上面的Add Account 窗口类似。

查看MaxCompute Region的开通情况和连接方式及Endpoint的设置,请参见配置Endpoint。

2.11 Studio视频介绍

- ・ Studio 安装介绍
- · 通过Studio管理数据
- · 通过Studio编辑SQL
- Studio SQL Scripting
- · 通过Studio开发UDF
- · 通过Studio remote debug
- · 通过Studio查看所有job

2.12 常见问题(FAQ)

2.12.1 如何通过 Studio 开发 UDF

本文介绍如何通过studio来开发MaxCompute Java UDF。

- 新增一个module: 参考创建MaxCompute java module,我们的UDF代码将放置在 此module中。
- 2. 开发UDF: 参考UDF开发 studio提供了UDF开发模板,你可以依据模板完成UDF开发。
- 3. 测试: studio提供了UDF本地调试的机制,你可以参考UDF test模板编写自己的测试用例。
- 4. 打包:你可以依赖IDE本身提供的打包功能,将自己的UDF源码打包成jar。
- 5. 注册发布: 在打好jar包后,需要先添加资源,随后注册函数。注册成功后就可以 在MaxCompute project explorer窗口的functions节点下查看该UDF,同时也可以 在script editor里使用该UDF。

2.12.2 如何通过 Studio 管理 MaxCompute 元数据

MaxCompute用户在日常使用过程中,需要浏览管理所在project的meta(包括表,函数和资

- 源),我们来看看如何通过Studio来方便的管理操作。
- 如何新增一个project连接: 参考新增连接 通过MaxCompute project explorer新建一
 个MaxCompute project连接,连接连通性测试成功后,就可以看到刚才新增的project节点
 树。
- ·如何查看table列表及schema:参考<u>浏览meta</u>展开project下的tables & views能看 到table和view列表,展开具体的某个table能看到列和类型。
- ·如何查看function代码:展开functions能看到function列表,展开具体的某个function能看 到方法签名,双击该function即可以看到反编译的源码。
- · 如何查看下载表的sample数据: 参考导入导出表数据 双击某个table就能看到详细 的schema,还能预览sample数据,在data preview窗格中右键Export grid data。你也可 以直接在节点树的表名上右键export。
- ·如何更新节点的meta: 譬如某个表新增了一列,你可以在选中该表后右键refresh meta或点 击工具栏上的refresh按钮,即可以看到新增的这一列了。又譬如project下新建了一些表,选 中tables & view后点击工具栏上的refresh按钮即可以看到新建的这些表了。
- ·如何对表做DDL操作:删除是可以直接批量操作的,选中表后,右键Delete table from server即可。新增和编辑还不支持直接通过界面操作,你可以通过在editor里编写相应 的DDL语句来完成。

3 Eclipse开发插件

3.1 安装Eclipse插件

本文向您介绍如何在Eclipse安装MaxCompute插件。

为了方便您使用 *MapReduce* 及*UDF*的Java SDK进行开发工作,MaxCompute(原 ODPS)提 供了Eclipse开发插件。该插件能够模拟MapReduce及UDF的运行过程,为您提供本地调试手 段,并提供了简单的模板生成功能。



- · 目前高版本的Eclipse Neon有可能会导致插件加载失败,请使用Eclipse Luna版本。
- ・请单击 <mark>此处</mark> 下载插件。
- · 与MapReduce提供的本地运行模式不同, Eclipse插件不能够与MaxCompute同步数据。您 使用的数据需要手动拷贝到Eclipse插件的warehouse目录下。

下载Eclipse插件后,将软件包解压,会看到如下Jar内容:

odps-eclipse-plugin-bundle-0.16.0.jar

将插件放置在Eclipse安装目录的plugins子目录下。打开Eclipse,单击右上角的打开透视 图(Open Perspective),如下图。



单击后出现对话框,如下图。

CVS Repository Exploring
🕸 Debug
🔐 Git
Java (default)
SJJava Browsing
Java Type Hierarchy
COPS
Plug-in Development
Resource
E ⁰ Team Synchronizing
Cancel OK

选择ODPS,单击OK 。同样在右上角会出现ODPS图标,表示插件生效。如下图。



3.2 创建MaxCompute工程

本文为您介绍在Eclipse中创建MaxCompute工程的两种方式。

方式一

 在左上角选择文件(File) > 新建(New) > Project > ODPS > ODPS Project, 创建工程(示例 中使用ODPS作为工程名)。

	New	Alt+Shift+N ►	1	ODPS Project	
	Open File		12	Java Project	
	Open Projects from File System			Project	
	Close	Ctrl+W	Ø	MapReduce Drive	r
	Close All	Ctrl+Shift+W	Ø	Mapper	
	Save	Ctrl+S	6	Reducer	
	Save As		UDF	UDF	
G	Save All	Ctrl+Shift+S	LIDTE	UDTF	
	Revert		ŧŶ	Package	
	Move		Ø	Class	
	Rename	F2	O	Interface	
8	Refresh	F5	G	Enum	
	Convert Line Delimiters To	+	e	Annotation	
æ	Print	Ctrl+P	↓ &	Java Working Set	
	11114	curri		Folder	
è	Import		r¢.	File	
4	Export			Untitled Text File	
	Properties	Alt+Enter		E	
	1 DownloadUtils.class [com.aliyun.odp]			Example	
	2 _schema_ [odps/warehouse//wc_in1]			Other	Ctrl+N
	3 UDFExample.java [odps/examples/com/]		- p	roject: example	project
	4 _schema_ [odps/warehouse//wc_in2]		ms	@ Javadoc 🖳 D	eclaration 📮
	Switch Workspace	+	les t	o display at this tin	ne.
	Restart				
	Exit				

 创建MaxCompute工程后会出现如下对话框。输入Project name,选择MaxCompute客户 端的安装路径(客户端需要提前下载),并确认(单击Finish)。

			Ouick Access
Create ODPS project Wzałd Project name: odps Use default location Location: D:\Users\Administrator\WorkSpace\odp Config ODPS console installation path Use default ODPS console installation path Specify ODPS console installation path Version: 0.29.4	Preferences ODPS Settings	Config ODPS console Config ODPS console Development Version: 0.29.4 Run Mode © Local © Remote limit record count of 100 (0~10000) © Retain local job ten	Quick Access installation path
?	?		Restore Defaults Apply Apply and Close Cancel

道 说明:	
关于MaxCompute客户端的介绍请参见客户端。	

3. 创建好工程后,在左侧包资源管理器(Package Explorer)中可以看到目录结构。



方式二

1. 直接单击左上角的新建。



2. 弹出对话框后,选择ODPS Project,单击下一步。

	New			
Select a wizard				
Wizards:				
type filter text				8
 Class Interface Java Project Java Project from Existing Ant Buildfile Plug-in Project General CVS CVS Git Java ODPS MapBeduce Driver ODPS Project Reducer UDAF UDF UDTF Plug-in Development CVS User Assistance 				
?	< Back	Next >	Cancel	Finish

3. 后续操作同方式一。

MaxCompute Eclipse插件安装结束后,您可以使用此插件编写MapReduce或者UDF程序。 关于插件中对MapReduce的功能介绍请参见MapReduce开发插件介绍,UDF的程序编写请参见UDF开发插件介绍。

3.3 MapReduce开发插件介绍

本文向您介绍如何使用Eclipse开发和运行MapReduce程序。

选择ODPS项目中的WordCount示例,如下图。

😫 Package Explorer 🛛	E 😓 😭	~	🕽 Wo	rdCount.java 🛛	-
4 👹 odps			1	<pre>package com.aliyun.odps.examples.mr;</pre>	
⊿ () src ⊕ odps			2 30	<pre>import java.io.IOException;</pre>	
👂 🛋 JRE System Lib	ary [Java]		180	/*	
Referenced Lib	raries		19	* 痰 ₂ √ 潘娟跟她被做MapReduce線披羅病 <u>要</u> 预据供通缉撤滞 */	
4 进 examples			20	public class WordCount {	
b 🌐 com.aliyun.c	dps.examples.graph		22		
a 🔠 com.aliyun.c	dps.examples.mr		23e	<pre>public static class TokenizerMapper extends MapperBase {</pre>	
Note of the second s	.java		25	Record word;	
UserDriv	er.java		26	Record one;	
⊳ 🚺 UserMap	per.java		27	Counter gCnt;	
UserRed	uce.java		290	@Override	
VordCo	unt.java		🔺 30	<pre>public void setup(TaskContext context) throws IOException {</pre>	
b 📠 com.aliyun.c	dps.examples.udf		31	<pre>word = context.createMapOutputKeyRecord(); </pre>	
com.aliyun.c	dps.examples.udf.test		33	one = context.treatemapoutputvalueRecord(); one_set(new Object[] {11});	
🕞 temp	· ·		34	<pre>gCnt = context.getCounter("MyCounters", "global_counts");</pre>	
b b warehouse			35	}	
				4	Þ

右键单击WordCount.java,依次单击Run As > ODPS MapReduce,如下图。

洋 Package Explorer 🛛	E 😫 💱 🔽 🗆	🛿 WordCount.java 🖾
⊿ 💯 odps		<pre>1 package com.aliyun.odps.examples.mr;</pre>
进 src		2 3⊕ import java.io.TOException:□
👂 🛋 JRE System Library [Java]	17
Referenced Libraries	New	18回 /*
a 进 examples	INew	▶ GUCC時度準備支援時度価格信
b # com.aliyun.odps	Open	F3 nt {
4 🔠 com.aliyun.odps	Open With	s TokenizerMapper extends MapperBase {
Resource.jav	Open Type Hierarchy	F4
Noracount.j	Show In	Alt+Shift+W ►
E com.aliyun.odps	Copy	Ctrl+C
🗁 temp	Copy Qualified Name	
b B warehouse	R Paste	Ctrl+V p(TaskContext context) throws IOException {
	Y Delete	.createMapOutputKeyRecord();
		ject[] {1L});
	Remove from Context	Ctrl+Alt+Shift+Down .getCounter("MyCounters", "global_counts");
	Build Path	•
	Source	Alt+Shift+S >
	Refactor	Alt+Shift+T ► Declaration 📮 Console 🖾
	import	Mapreduce Graph] D:\Program Files\Java\jre\bin\javaw.exe (2018年1
	🖾 Export	
	Defense	
	Deslavations	
	Declarations	
	🔗 Refresh	F5
	Assign Working Sets	
	Coverage As	•
	Run As	▶ J Java Application Alt+Shift+X, J
	Debug As	2 ODPS MapreducelGraph
	Validate	
	Restore from Local Histo	ory
	Team	Run Configurations
	Compare With	+ 5
I		

弹出对话框后,选择example_project,单击确认,如下图。

⊖ ODPS MapReduce Graph Run Configuration □ ∞	3
ODPS Mapreduce Graph Run Configuration	
Class com.aliyun.odps.examples.mr.WordCount	
Run Mode	
Select ODPS Project	
MySecondProject2 example_project Edit Remove	
Resources	
•	
Program Arguments	
? <u>Finish</u> Cancel]

运行成功后,会出现结果提示,如下图。

E Console X	- 🗶 🔆 📴 🖉 🖉 🖉 - 📬 - 🗆 -
<terminated>WordCount [ODPS Mapreduce] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java (2015年1月27日下午3:</terminated>	42:38)
Summary: Inputs:	
<pre>example_project.wc_in1,example_project.wc_in2/p1=2/p2=1</pre>	
example_project.wc_out	
M1 example project LOT 0 0 0 job0	
Worker Count: 2	
Input Records:	
Output Records:	
R2_1: 17 (min: 8, max: 9, avg: 8)	
R2_1_example_project_LOT_0_0_0_job0	
Input Records:	
input: 5 (min: 5, max: 5, avg: 5)	
Output Records: R2 1FS 9: 5 (min: 5, max: 5, ava: 5)	
counters: 10	
map-reduce framework: 7	
combine_input_groups=5 combine_output_records=5	
map_input_bytes=87	
map_input_records=7	
<pre>map_output_records=1/ reduce_output_records=1/ reduce_output_records_project.wc_out]_bytes=37</pre>	
reduce_output_[example_project.wc_out]_records=5	
user defined counters: 3	
mycounters alobal_counts=22	
map_outputs=17	
reduce_outputs=5	
OK	
InstanceId: mr_20150127074239_358_27772	

运行自定义MapReduce程序

右键选择src目录,选择新建(New) > Mapper,如下图。



选择Mapper后出现下面的对话框。输入Mapper类的名字,并确认。如下图。

New Mapper	r	
Mapper		
Create a new I	Mapper implementation.	
Source fol <u>d</u> er:	odps/examples	Br <u>o</u> wse
Pac <u>k</u> age:	com.aliyun.odps.examples.mr	Bro <u>w</u> se
Na <u>m</u> e:	UserMapper	
<u>S</u> uperclass:	com.aliyun.odps.mapred.MapperBase	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		Remove
		1
?	<u>F</u> inish	Cancel

会看到在左侧包资源管理器(Package Explorer)中,src目录下生成文件UserMapper.java。该 文件的内容即是一个Mapper类的模板,如下图。

```
package odps;
import java.io.IOException;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.MapperBase;
public class UserMapper extends MapperBase {
    @Override
    public void setup(TaskContext context) throws IOException {
    }
    @Override
    public void map(long recordNum, Record record, TaskContext context
)
        throws IOException {
    }
    @Override
    public void cleanup(TaskContext context) throws IOException {
    }
    }
}
```

}

模板中,将package名称默认配置为"odps",用户可以根据自己的需求进行修改。编写模板内容、如下图。

```
package odps;
import java.io.IOException;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.MapperBase;
public class UserMapper extends MapperBase {
    Record word;
    Record one;
    Counter gCnt;
    @Override
    public void setup(TaskContext context) throws IOException {
          word = context.createMapOutputKeyRecord();
          one = context.createMapOutputValueRecord();
          one.set(new Object[] { 1L });
          gCnt = context.getCounter("MyCounters", "global_counts");
    @Override
    public void map(long recordNum, Record record, TaskContext context
)
            throws IOException {
          for (int i = 0; i < record.getColumnCount(); i++) {</pre>
              String[] words = record.get(i).toString().split("\\s+");
              for (String w : words) {
                word.set(new Object[] { w });
                Counter cnt = context.getCounter("MyCounters", "
map_outputs");
                cnt.increment(1);
                gCnt.increment(1);
                context.write(word, one);
              }
            }
    @Override
    public void cleanup(TaskContext context) throws IOException {
}
```

同理,右键选择src目录,选择新建(New) > Reduce,如下图。

New Reduce	r	
Reducer		
Create a new l	Reducer implementation.	
Source folder:	odps/examples	Browse
Package:	com.aliyun.odps.examples.mr	Browse
Name:	UserReduce	
Superclass:	com.aliyun.odps.mapred.ReducerBase	Browse
Interfaces:		Add
		Remove
?	Finish	Cancel

输入Reduce类的名字(本示例使用UserReduce):

同样在包资源管理器(Package Explorer)中, src目录下生成文件UserReduce.java。该文件的 内容即是一个Reduce类的模板。编辑模板,如下图。

```
package odps;
import java.io.IOException;
import java.util.Iterator;
import com.aliyun.odps.counter.Counter;
import com.aliyun.odps.data.Record;
import com.aliyun.odps.mapred.ReducerBase;
public class UserReduce extends ReducerBase {
    private Record result;
    Counter gCnt;
    @Override
    public void setup(TaskContext context) throws IOException {
          result = context.createOutputRecord();
          gCnt = context.getCounter("MyCounters", "global_counts");
    }
    @Override
    public void reduce(Record key, Iterator<Record> values, TaskContex
t context)
             throws IOException {
          long count = 0;
```

```
while (values.hasNext()) {
    Record val = values.next();
    count += (Long) val.get(0);
    }
    result.set(0, key.get(0));
    result.set(1, count);
    Counter cnt = context.getCounter("MyCounters", "reduce_out
puts");
    cnt.increment(1);
    gCnt.increment(1);
    context.write(result);
    }
    @Override
    public void cleanup(TaskContext context) throws IOException {
    }
}
```

创建main函数: 右键选择src目录,选择新建(New) > MapReduce Driver。

填写Driver Name(示例中是UserDriver), Mapper及Recduce类(示例中

是UserMapper及UserReduce),并确认。同样会在src目录下看到MyDriver.java文件,如下 图。

e New MapRed	duce Driver	
MapReduce Dr	iver	
Create a new N	MapReduce driver.	
Source fol <u>d</u> er:	odps/examples	Br <u>o</u> wse
Pac <u>k</u> age:	com.aliyun.odps.examples.mr	Bro <u>w</u> se
Na <u>m</u> e:	UserDriver	
<u>S</u> uperclass:	java.lang.Object	Brows <u>e</u>
Interfaces:		<u>A</u> dd
		Remove
		·
Ma <u>p</u> per:	UserMapper	<u>B</u> rowse
<u>R</u> educer:	UserReduce	Browse <u>.</u>
(?)	<u> </u>	Cancel

编辑driver内容,如下图。

```
package odps;
import com.aliyun.odps.OdpsException;
import com.aliyun.odps.data.TableInfo;
import com.aliyun.odps.examples.mr.WordCount.SumCombiner;
import com.aliyun.odps.examples.mr.WordCount.SumReducer;
import com.aliyun.odps.examples.mr.WordCount.TokenizerMapper;
import com.aliyun.odps.mapred.JobClient;
import com.aliyun.odps.mapred.RunningJob;
import com.aliyun.odps.mapred.conf.JobConf;
import com.aliyun.odps.mapred.utils.InputUtils;
import com.aliyun.odps.mapred.utils.OutputUtils;
import com.aliyun.odps.mapred.utils.SchemaUtils;
public class UserDriver {
    public static void main(String[] args) throws OdpsException {
        JobConf job = new JobConf();
        job.setMapperClass(TokenizerMapper.class);
        job.setCombinerClass(SumCombiner.class);
        job.setReducerClass(SumReducer.class);
        job.setMapOutputKeySchema(SchemaUtils.fromString("word:string
"));
```

```
job.setMapOutputValueSchema(SchemaUtils.fromString("count:
bigint"));
            InputUtils.addTable(
                 TableInfo.builder().tableName("wc_in1").cols(new String[]
        { "col2", "col3" }).build(), job);
                 InputUtils.addTable(TableInfo.builder().tableName("wc_in2").
        partSpec("p1=2/p2=1").build(), job);
                OutputUtils.addTable(TableInfo.builder().tableName("wc_out").
        build(), job);
                OutputUtils.addTable(TableInfo.builder().tableName("wc_out").
        build(), job);
                RunningJob rj = JobClient.runJob(job);
               rj.waitForCompletion();
        }
}
```

运行MapReduce程序,选中UserDriver.java,右键选择Run As > ODPS MapReduce,点击确认。出现对话框,如下图。

🖨 ODPS MapReduce Graph Run Configuration 🗖 💷 🔀
ODPS Mapreduce Graph Run Configuration
Class com.aliyun.odps.examples.mr.UserDriver
Run Mode Local Remote
Select ODPS Project
MySecondProject2 example_project Add Edit Remove
Resources
Program Arguments
? Finish Cancel

选择ODPS Project为: example_project, 单击Finish按钮开始本地运行MapReduce程序, 如下图。

El Counte M		Sile .		1 🛹 🖃	. = . =	
		-96	-x (18 6- 6		· [_2 ·	
<terminated> UserDriver [ODPS Mapreduce] /Library/Java/JavaVirtualMachines/jdk1.7.0_71.jdk/Contents/Home/bin/java (2015年1月27日下午4:</terminated>	22:42)					
<pre>Summary: Inputs: example_project.wc_in1,example_project.wc_in2/p1=2/p2=1 Outputs: example_project.wc_out M1_example_project_LOT_0_0_0_job0 Worker Count: 2 Input Records: input Records:</pre>						
R2_1_example_project_L01_0_0_0_job0 Worker Count: 1 Input Records: input: 5 (min: 5, max: 5, avg: 5) Output Records: R2_1FS_9: 5 (min: 5, max: 5, avg: 5)						
<pre>counters: 10 map-reduce framework: 7 combine_input_groups=5 combine_output_records=5 map_input_bytes=87 map_input_records=7 map_output_records=17 reduce_output_[example_project.wc_out]_bytes=37 reduce_output_[example_project.wc_out]_records=5 user defined counters: 3 mycounters</pre>						
OK InstanceId: mr_20150127082243_694_27864						
	1					

有如上输出信息,说明本地运行成功。运行的输出结果在warehouse目录下。关于warehouse的 说明请参考本地运行。刷新ODPS工程,如下图。

Package Explorer S Codps Codps A2, 4 A3, 4 Codps Sec A2, 4 A3, 4 Codps Sec A2, 4 A3, 4 Codps Sec A2, 4 A3, 4 Codps Sec A3, 4 Codps Sec A3, 4 Codps Sec Sec <th>Rackage Explorer X Ackage Explorer X Action MordCountjava MordCoun</th> <th>Package Explore: X Package Explore: X Pit odps Pit odps <</th> <th>। 🗄 🕒 🏇 • 🖸 • 🂁 🤁 🧭 • 🔳</th> <th>Ⅲ ∲ • ∲ • ⇔ ⇔ →</th> <th>•</th> <th></th> <th></th> <th>0</th> <th></th> <th>1</th> <th>г 🕹</th> <th>×</th> <th>~</th>	Rackage Explorer X Ackage Explorer X Action MordCountjava MordCoun	Package Explore: X Package Explore: X Pit odps Pit odps <	। 🗄 🕒 🏇 • 🖸 • 🂁 🤁 🧭 • 🔳	Ⅲ ∲ • ∲ • ⇔ ⇔ →	•			0		1	г 🕹	×	~
 ▶ Beferenced Libraries > Wethous and the second second	► A Referenced Libraries Sexamples Fig. Com.aliyun.odps.examples.mr I Resource.java V Groom.aliyun.odps.examples.mr I MosoCount.java V Groom.aliyun.odps.examples.udi.ext I SumReducer I SumReducer I SumReducer I SumReducer I Com.aliyun.odps.examples.udf I com.aliyun.odps.examples.udf.test I com.aliyun.odps.examples.u	► Referenced Libraries ★ Beferenced Libraries ★ Become aliyun.odps.examples.mr ★ Resource.java ★ WordCountiava ★ WordCountiava ★ WordCountiava ★ SumReducer ★ SumReducer ★ TokenizerMapper ★ main(String[]: void ★ contaliyun.odps.examples.udf.test ★ contaliyun.odps.examples.udf.test ★ warshouse ★ example_project ★ able_resource1 ★ contalive_resource2 A file_resource1 ★ costable_resource1 ★ costable_resource1 ★ costable_resource1 ★ costable_resource2 A file_resource1 ★ costable_resource2 ★ file_resource1 ★ costable_resource2 ★ Roomood ★ Roomood ★ Roomood	Package Explorer ಔ	A2,4 A3,4 three1,3 three2,3 three3,3	J UserReduce.java	UserDriver.java	■ R_000000		•				
	<pre>></pre>	<pre>> Console 23</pre>	 ▶ ■ Referenced Libraries ♥ Become.aliyun.odps.examples.mr ▶ Become.aliyun.odps.examples.mr ▶ Become.aliyun.odps.examples.mr ♥ Become.aliyun.odps.examples.und ♥ Become.aliyun.odps.examples.udf 										

wc_out即是输出目录,R_000000即是结果文件。通过本地调试,确定输出结果正确后,可以通过Eclipse导出(Export)功能将MapReduce打包。打包后将jar包上传到ODPS中。在分布式环境下执行MapReduce,详情请参考快速入门。

本地调试通过后,用户可以通过Eclipse的Export功能将代码打成jar包,供后续分布式环境使用。 在本示例中,我们将程序包命名为mr-examples.jar。选择src目录,单击Export,如下图。

▼∰odps		A2,4	
 P src JRE System Library [Java SE 7 [' Referenced Libraries 	New Go Into	AR 4	•
 examples com.aliyun.odps.examples.mi Resource.java WordCount.java WordCount 	Open in New Wind Open Type Hierarc Show In	ow hy ∼≋W	F4 ▶
 SumCombiner SumReducer StokenizerMapper main(String[]) : void com.aliyun.odps.examples.ud 	 Copy Copy Qualified Paste Delete 	Name	жс жv ⊠>
 temp <li< td=""><td>Build Path Source Refactor</td><td>て#S て#T</td><td>* * *</td></li<>	Build Path Source Refactor	て#S て#T	* * *
 table_resource1 table_resource2 file_resource.txt tables rs_out rs_out wc_in1 wc_in2 wc_out schema R_000000 Readme 	🖾 Export		
	🦑 Refresh Assign Working Se	ts	F5
	Debug As Run As Team Compare With Restore from Local % Import from Rep	l History	* * *
	Properties		жı

选择导出模式为Jar File,如下图。

Export	
Select Export resources into a JAR file on the local file system.	ß
Select an export destination:	
type filter text	
 Ceneral Install JAR file Runnable JAR file Plug-in Development Run/Debug Team 	
(?) < Back Next > Finish	Cancel

仅需要导出src目录下package(com.aliyun.odps.mapred.open.example), Jar File名称指定 为mr-examples.jar,如下图。

JAR Export	
JAR File Specification (1) The export destination will be relative to ye	our workspace.
Select the resources to export	
 ▲ ∰ ODPS ▲ ♥ ⊕ src ♥ ⊕ com.aliyun.odps.mapred.o ▶ @ ⊕ examples @ ⊕ mr_local_jobs ▶ @ ⇐ warehouse 	 classpath project
Export generated class files and resource Export all output folders for checked proj Export Java source files and resources	s rects
Export refactorings for checked projects. Select the export destination: JAR file: mr-examples.jar Options: Compress the contents of the IAP file	Select refactorings Browse

确认后,导出成功。

如果用户想在本地模拟新建Project,可以在warehouse下面,创建一个新的子目录(与 example_project平级的目录),目录层次结构,如下图。



schema文件示例,如下图。

```
非分区表:
project=project_name
table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:STRING
分区表:
project=project_name
table=table_name
columns=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:STRING
partitions=col1:BIGINT,col2:DOUBLE,col3:BOOLEAN,col4:DATETIME,col5:
STRING
注: 当前支持5种数据格式:bigint,double,boolean,datetime,string, 对应到java
中的数据类型-long,double,boolean,java.util.Date,java.lang.String。
```

data文件示例,如下图。

1,1.1,true,2015-06-04 11:22:42 896,hello world \N,\N,\N,\N,\N,\N 注:时间格式精确到毫秒级别,所有类型用\N表示null。



- 本地模式运行MapReduce程序,默认情况下先到warehouse下查找相应的数据表或资源,如
 果表或资源不存在会到服务器上下载相应的数据存入warehouse目录下,再以本地模式运行。
- · 运行完MapReduce后,请刷新warehouse目录,才能看到生成的结果。

3.4 UDF开发插件介绍

在本章节我们将介绍如何使用Eclipse插件开发并在本地运行UDF。UDAF、UDTF的编写执行过 程与UDF类似,均可参考UDF的示例介绍完成。ODPS Eclipse插件提供两种运行UDF的方式:菜 单栏和右键单击快速运行方式。

菜单栏运行

1. 从菜单栏选择Run > Run Configurations…弹出如下对话框:

Run Configurations	X
Create, manage, and run cont	figurations
Image: Second	Name: Resource Image: Main ODPS Config M= Arguments Image: Im
?	Run Close

2. 用户可以新建一个Run Configuration,选择运行的UDF类及类型、选择ODPS Project、填 写输入表信息,如:

〕 🗎 🗙 🕒 券▼	Name: udf 2. 选择配置标签页	
type filter text 🛛 🗷	🕒 UDF UDTF UDAF	vironment Common
 Apache Tomcat Eclipse Application 	Project:	3. 选择 Java坝日
	odps_example	<u>B</u> rowse
Eclipse Data Tools	UDF UDTF UDAF class:	4. 选择UDF类
Generic Server	com.aliyun.odps.examples.udf.UDFExample	<u>S</u> earch
 HTTP Preview J2EE Preview 	Class Type 选择UDF类型	
🖭 Java Applet	Select ODPS project	
Java Application	example_project	Add The whole water
Ju JUnit	选择ODPS Project	Fdit OPDC Durationt
Ju JUNIC Plug-In Test		UDPS Project
DDPS Mapreduce		Remove
🕸 wordcount	Input Table 输入表信息	
ODPS UDF UDTF UDAF	Table: wc_in2	
udf	Partitions: p2=1,p1=2	ie: p1=1,p2=1 (default all partitions
OSGi Framework	Columns: colc.cola.colb	ie: c1,c2,c3 (default all columns)
Ju Task Context Test		
× XSL	1. 右键单击>New新建一个Configuration	

上述配置中, Table表示UDF的输入表, Partitions表示读取某个分区下的数据, 分区由逗 号分隔, Columns表示列, 将依次作为UDF函数的参数被传入, 列名由逗号分隔。

3. 单击Run运行,运行结果将显示在控制台中:



右键单击快速运行

1. 选中一个udf.java文件(比如:UDFExample.java)并单击鼠标右键,选择Run As > Run UDF|UDAF|UDTF。

🛱 Package Explorer 🔀	🖻 🔄 🎽 🗖 🚺 UDFExa	ample.java 🔀
▼ ^{¢0P5} odps	pack	<pre>kage com.aliyun.odps.examples.udf;</pre>
/∰ src	impo	ort com alivun odna udf UDE:
JRE System Library [Java SE 7 [1]	.7.0_71]]	ore comartyun.oups.uur.obr,
Hererenced Libraries	publ	lic class UDFExample extends UDF {
Among and a second s		
The com.aliyun.odps.examples.ud	New	<pre>t: example_project</pre>
UDAFExample.java	0	wc_in1
▶ ↓ UDAFResource.java	Open	F3 s: col1,col2
DUDFExample.java	Open With	F
► M UDFResource.java	Open Type Hierarchy	F4 ring evaluate(String a, String b)
UDTEResource java	Show In X#W	V ▶ "ss2s:" + a + "," + b;
Gorn Hesedree, java	Copy	¥C.
i comp		80
▶ 🗁 warehouse		t: example_project
	Paste	帯V wc_in1
	💢 Delete	S: col1,col2,col4,col3
	Build Path	•
	Source 77#S	ring evaluate(String a, String b,
	Befactor VIII	"ssss2s:" + a + "," + b + "," + c
	🔁 Import	
	🖾 Export	t: example_project
		wc_in2
	References	s: colc.colb.cola
	Declarations	
	A Defrech	ring evaluate(String a, String b,
	« Refresh	F5 "sss2s:" + a + "," + b + "," + c
	Assign Working Sets	
	Debug As	•
	Run As	I ODPS UDF UDTF UDAF
	Team	
	Compare With	Run Configurations
	Replace With	•
	Restore from Local History	yproject.wc_in1,example_project.w
2. 配置信息如下:

000	ODPS UDF UDTF UDAF Rur	Configuration
ODPS UDF U	JDTF UDAF Run Configuration	
Class		
com.aliyun	.odps.examples.udf.UDFExample	
Select ODP	PS Project	
example_p	roject	Add Edit Remove
Input Table		
Table:	wc_in2	
Partitions:	p2=1,p1=2	ie: p1=1,p2=1 (default all partitions)
Columns:	colc,colb,cola	ie: c1,c2,c3 (default all columns)
?		Cancel Finish

上述配置中,Table表示UDF的输入表,Partitions表示读取某个分区下的数据,分区由逗 号分隔,Columns表示列,将依次作为UDF函数的参数被传入,列名由逗号分隔。

3. 单击Finish后,运行UDF,获得输出结果。

运行用户自定义UDF程序

右击一个工程并选择New > UDF(或者选择菜单栏File > New > UDF)。

填写UDF类名然后单击Finish。在对应的src目录下生成与UDF类名同名的Java文件,编辑该java文件内容:

```
package odps;
import com.aliyun.odps.udf.UDF;
public class UserUDF extends UDF {
```

```
/**
 * project: example_project
 * table: wc_in1
 * columns: col1,col2
 *
 */
public String evaluate(String a, String b) {
 return "ss2s:" + a + "," + b;
}
```

右击该java文件(如UserUDF.java),选择Run As,再选择ODPS UDF|UDTF|UDAF:

ቹ Package Explorer 🛛 📄 🔄	UDFExample.jav	a UserUDF.java 🛛
▼ ∰ odps ▼ ∰ src ▼ ⊕ odps	package odp import com.	s; aliyun.odps.udf.UDF;
UserUDF.java JBE System Library Java SE 7 [1,7,0,71]	New	DF extends UDF {
 ▶ ■ Referenced Libraries ▼ ⊕ examples ▶ ⊕ com.aliyun.odps.examples.mr ▼ ⊕ com.aliyun.odps.examples.udf ▶ ↓ UDAFExample.java 	Open Open With Open Type Hierarchy Show In て第W	F3 example_project in1 col1,col2
 ▶ ↓ UDAFResource.java ▶ ↓ UDFExample.java ▶ ↓ UDFResource.java ▶ ↓ UDFFExample.java ▶ ↓ UDFFResource.java 	 Copy Copy Qualified Name Paste Delete 	<pre>%C g evaluate(String a, String b) { 2s:" + a + "," + b; %V ☑</pre>
 ► temp ► being warehouse 	Build Path Source て#S Refactor て#T	
	≧ Import ⊿ Export	
	References Declarations	S UDF UDTF UDAF] /Library/Java/JavaVirtual
	Refresh Assign Working Sets	F5
	Debug As	•
	Run As	I ODPS UDF UDTF UDAF
	Team Compare With Replace With Restore from Local History.	Run Configurations
	Properties	81

配置如下对话框:

	ODPS UDF UDTF UDAF	Run Configuration
ODPS UDF U	JDTF UDAF Run Configuration	
Class		
odps.Userl	JDF	
Select ODP	PS Project	
example_p	roject	Add Edit Remove
Table:	wc_in1	
Partitions:		ie: p1=1,p2=1 (default all partitions)
Columns:	col1,col2	ie: c1,c2,c3 (default all columns)
?		Cancel Finish

单击Finish,得出结果:

ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2
ss2s:A1,A2

本示例中仅给出UDF的运行示例,UDTF的运行方式与UDF基本相同,不做特殊说明。

3.5 Graph开发插件介绍

创建ODPS项目后,用户可以编写自己的Graph程序,参照下文步骤操作完成本地调试。

在此示例中,我们选用插件提供的PageRank.java来完成本地调试工作。选中examples下的 PageRank.java文件,如下图。

<pre> WorkSpace - odps/examples/com/aliyun/odps/examples/graph/PageRank.java - Eclipse File Edit Source Refactor Navigate Search Project Run Window Help Package Explorer Package Com.aliyun.odps.examples.graph PageRankjava Package Spipar PageRankjava Package Com.aliyun.odps.examples.graph PackageRankjava Package Com.aliyun.odps.examples.graph PageRankjava Package Com.aliyun.odps.examples.graph PackageRankjava Package Com.aliyun.odps.examples.graph PackageRankjava Package Com.aliyun.odps.examples.graph PackageRankjava Package Com.aliyun.odps.examples.graph PackageRankjava PackageRankjava PackageRankjava PackageRankjava Package PackageRankjava PackageRankjava Package PackageRankjava Package Packa</pre>			
<pre>File Edit Source Refactor Navigate Search Project Run Window Help Package Explorer Package Con.aliyun.odps.examples.graph PageRankjava Package Con.aliyun.odps.examples.graph Pa</pre>	WorkSpace - odps/examples/com/aliyun/odps/examples/	'graph/PageRank.java - Eclipse	
<pre> Y = 0 + 4 + 0 + 4 + 1 + 0 + 4 + 1 + 0 + 4 + 1 + 0 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 4</pre>	<u>File Edit Source Refactor Navigate Search Project</u>	<u>R</u> un <u>W</u> indow <u>H</u> elp	
<pre> Package Explorer Package com.aliyun.odps.examples.graph PageRankjava Package and Package Explorer Package com.aliyun.odps.examples.graph PageRankjava PackageRankjava PackageRankjava</pre>	📑 • 🗐 🕲 • 🔅 • O • 🌯 • 🖶 🎯 • 😕 (∌ 🔗 ▼ : 🖗 🕖 ⋧ 😻 🗉 🗊 : 🐓 ▼ 🧞 ▼ 🟷 🗢 ▼ ⇔ ▼	
<pre> # \$\vec{start}{c} odps * \$\vec{start}{c} odps * \$\vec{start}{c} \$\vec{start}</pre>	😫 Package Explorer 🛛 📄 🗟 🕼 🍸 🗖 🗖	🕐 PageRankjava 🛛	-
<pre>> > warehouse 33 setValue(new DoubleWritable(1.0 / context.getTotalNumVertices())); 34 } else if (context.getSuperstep() >= 1) { 35 double sum = 0; 36 for (DoubleWritable msg : messages) {</pre>	 dos src M.RE System Library [Java] M.Referenced Libraries Examples Econ.aliyun.odps.examples.graph RegeRankjava SSSP.java SSSP.java SSSP.java con.aliyun.odps.examples.udf Econ.aliyun.odps.examples.udf.test temp warehouse 	<pre>1 package com.aliyun.odps.examples.graph; 2 3</pre>	

右键单击,选择Debug As > ODPS MapReduce|Graph,如下图。



单击后出现对话框,作如下配置。

ODPS MapReduce Graph Run Configuration	- • ×
ODPS Mapreduce Graph Run Configuration	
Class com.aliyun.odps.examples.graph.PageRank	
Run Mode Local Remote	
Select ODPS Project	
MySecondProject2 example_project	Add Edit Remove
Resources	
	*
Program Arguments	
pagerank_in pagerank_out	•
? <u>Finish</u>	Cancel

查看作业运行结果,如下图。

🖹 Problems @ Javadoc 😣 Declaration 💷 Console 🛿		×	×.
<terminated> PageRank [ODPS Mapreduce Graph] D:\Program Files\Java\jre\bin\java</terminated>	w.exe (2018年12月17日 上午11:45:54)		
3			
vertex edgs size: 2			
4			
2			
vertex edgs size: 2			
3			
1			
2			
vertex edgs size: 2			
User defined counters: 4			
com.aliyun.odps.graph.local.COUNTER			
TASK_INPUT_BYTE=24			
TASK_INPUT_RECORD=4			
TASK_OUTPUT_BYTE=88			
TASK_OUTPUT_RECORD=4			
[INFO]Reload warehouse table:pagerank_out			
graph task finish			
Job Finished in 0.5 seconds			

可以查看在本地的计算结果,如下图。



调试通过后,用户可以将程序打包,并以Jar资源的形式上传到ODPS,并提交Graph作业。



- ·程序打包过程以及有关本地结果的目录结构介绍,请参见MapReduce开发插件介绍。
- ·关于上传Jar资源的详细介绍,请参见资源操作中创建Jar资源部分。

· 有关提交Graph作业的操作,请参见Graph功能介绍中Jar命令的介绍。

4 相关下载

本文将为您提供在使用MaxCompute过程中,可能用到的相关工具及插件的下载地址。

- · SDK下载信息:如果您使用Maven,可以从Maven库中搜索odps-sdk,获取不同版本的Java SDK。
- 新版客户端:点击此处即可下载新版客户端。
- · Eclipse开发插件:点击此处即可下载 Eclipse开发插件。
- · IntelliJ开发插件: IDEA工具点击此处, Studio插件点击此处即可下载 IntelliJ开发插件。
- · JDBC: MaxCompute提供开源JDBC,您可以在GitHub下载JDBC,也可以在云栖社区查看发 布信息或提问。
- ・ PHP SDK: 您可以在GitHub下载。