



# Day19 打破数据孤岛

## —— 异构数据源联合分析业务创新实践





# 目录

- 异构数据源商业价值
- 案例分享——企业智能化数据经营分析



# 异构数据源商业价值

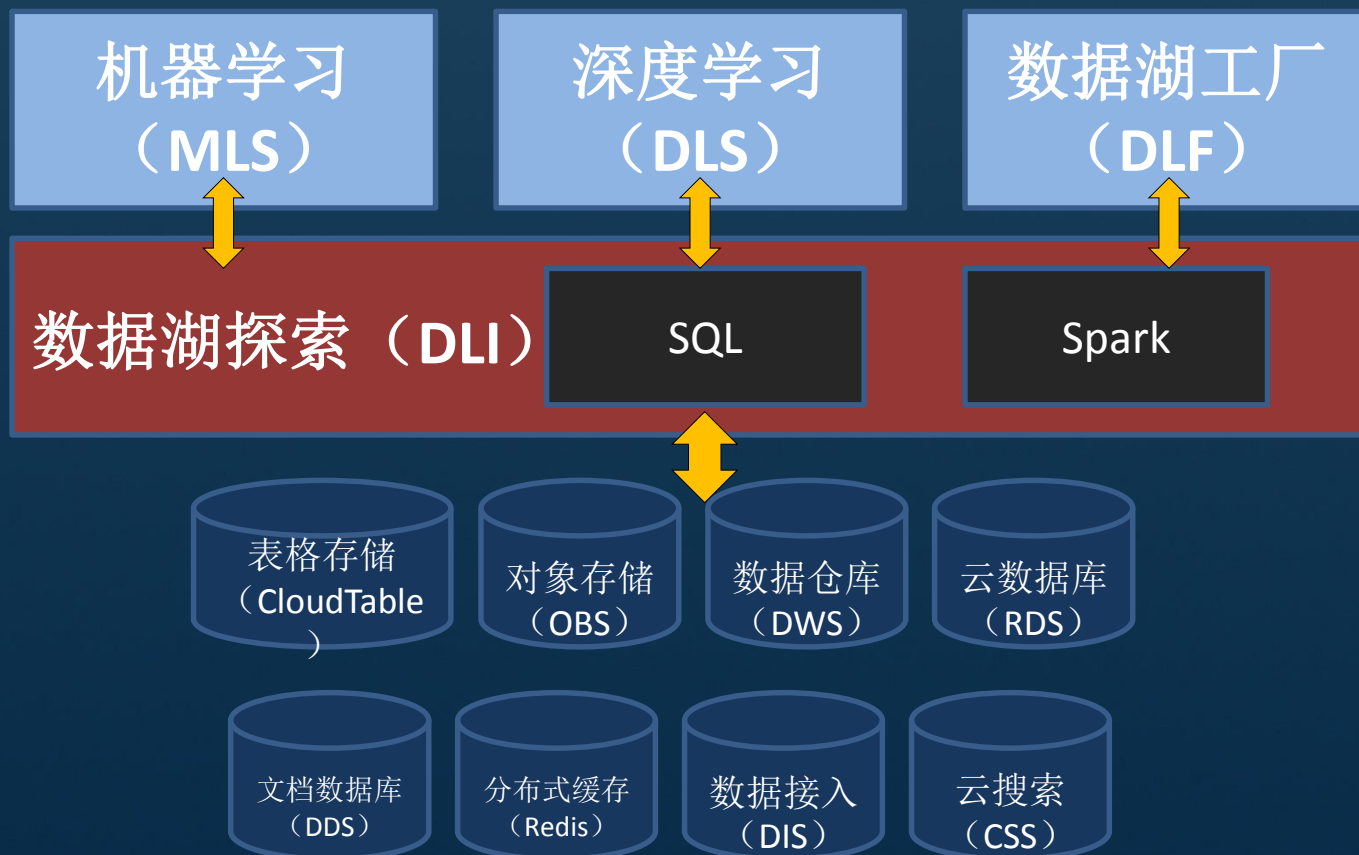
## 普遍的企业多源数据场景

各种场景下的数据，往往需要使用各种不同的大数据产品进行数据的存储与处理。而不同来源的数据，需要联合分析才能发挥其最大价值。



# 数据湖探索（DLI）服务多源数据分析架构

DLI通过Spark的DataSource能力，连接不同数据源，实现不同服务之间的跨源分析计算。同时，DLI提供基于SQL的计算集群与定制化的Spark serverless计算集群。用户可以编写简单的SQL语句，也可以自定义复杂的Spark程序来实现跨源分析。





# 案例分享

## ——企业智能化数据经营分析实战

# 案例介绍

## 背景

企业H需要对其产品、经销商与销售订单进行信息管理。其中，各个数据信息介绍如下：

产品信息包含如下内容：

产品信息表 ( PRODUCT\_INFO ) 的Schema信息

- 产品编号
- 产品名称
- 单价

标签	产品编号	产品名称	单价
字段名	ID	NAME	PRICE

经销商信息包含如下内容：

经销商信息表 ( DEALER\_INFO ) 的Schema信息

- 经销商编号
- 经销商名称

标签	经销商编号	经销商名称
字段名	ID	NAME

# 案例介绍

销售订单信息包含如下内容：

- 经销商编号
- 订单编号
- 订单日期
- 订购商品名称
- 订购商品数量

销售订单信息表 ( BILL ) 的Schema信息

经销商编号	订单编号	日期		商品信息		
RowKey前4位	RowKey后4位	time		arts		
		month	day	商品1	商品2	...

考虑到每份订单订购的商品是不同的，因此企业H采用将订单信息以No-SQL方式（HBase）进行存储。其中：

- RowKey由4位经销商编号+4位订单编号组成
- 商品信息存放至列簇arts中，以每种商品的名称作为列名，值为商品数量
- 因业务需要（后续场景将介绍），考虑将订单日期存放至列簇time中，并以“month”表示月份，“day”表示日期



# 样例数据

## 产品信息数据：

ID	NAME	PRICE
0001	nova	1000
0002	P20	3000
0003	Mate20	5000

## 经销商信息数据：

ID	NAME
0003	B1
0005	X7

# 样例数据

## 订单信息数据：

```
hbase(main):001:0> scan 'BILL'
ROW                                COLUMN+CELL
00030001                          column=arts:Mate20, timestamp=1541678508529, value=\x00\x00\x00\x05
00030001                          column=arts:nova, timestamp=1541679193771, value=\x00\x00\x00\x0A
00030001                          column=time:day, timestamp=1541679193771, value=\x00\x00\x00\x01
00030001                          column=time:month, timestamp=1541679193771, value=Oct
00030002                          column=arts:Mate20, timestamp=1541679171382, value=\x00\x00\x00\x05
00030002                          column=arts:P20, timestamp=1541679208503, value=\x00\x00\x00\x05
00030002                          column=time:day, timestamp=1541679208503, value=\x00\x00\x00\x01
00030002                          column=time:month, timestamp=1541679208503, value=Nov
00030003                          column=arts:Mate20, timestamp=1541679175926, value=\x00\x00\x00\x01
00030003                          column=arts:P20, timestamp=1541679213593, value=\x00\x00\x00\x01
00030003                          column=arts:nova, timestamp=1541679196942, value=\x00\x00\x00\x02
00030003                          column=time:day, timestamp=1541679213593, value=\x00\x00\x00\x02
00030003                          column=time:month, timestamp=1541679213593, value=Nov
00050001                          column=arts:Mate20, timestamp=1541679182273, value=\x00\x00\x00\x05
00050001                          column=arts:P20, timestamp=1541679217519, value=\x00\x00\x00\x05
00050001                          column=arts:nova, timestamp=1541679202374, value=\x00\x00\x00\x14
00050001                          column=time:day, timestamp=1541679217519, value=\x00\x00\x00\x01
00050001                          column=time:month, timestamp=1541679217519, value=Oct
00050002                          column=arts:Mate20, timestamp=1541679186275, value=\x00\x00\x00\x05
00050002                          column=time:day, timestamp=1541679186275, value=\x00\x00\x00\x14
00050002                          column=time:month, timestamp=1541679186275, value=Nov
5 row(s) in 0.1910 seconds
```



rowkey	time		arts		
	month	day	nova	P20	Mate20
00030001	Oct	1	10		5
00030002	Nov	1		5	5
00030003	Nov	2	2	1	1
00050001	Oct	1	20	5	5
00050002	Nov	20			5

## 场景一：计算经销商B1每月Mate20销量与销售额

**解决方案：**每个月的销量数据存放在CT的订单信息中，DLI可以使用分组聚合的方式按月计算销量；而商品信息存放在DWS中，要使用DLI的跨源多表联合能力，计算出销售额（销量 \* 单价）。



# 场景一：计算经销商B1每月Mate20销量与销售额



## 场景一：计算经销商B1每月Mate20销量与销售额

### 数据湖探索 DLI

数据湖探索（Data Lake Insight，简称DLI）是完全托管的大数据处理分析服务。免搬迁，轻松完成对云上异构数据源的探索分析

按需付费，CU单价¥1.4元/小时 [了解详情 →](#)

[进入控制台](#)

[购买套餐包](#)

[体验中心](#)

[快速入门](#) | [最佳实践](#) | [帮助文档](#)

- 推荐搭配[数据湖工厂（DLF）](#)，提供一站式数据开发IDE平台

# 场景一：计算经销商B1每月Mate20销量与销售额



数据湖探索（Data Lake Insight, DLI）是完全托管的大数据处理分析服务。用户不需要管理任何服务器，即开即用。支持标准SQL，兼容SparkSQL，支持多种接入方式，并兼容主流数据格式。数据无需复杂的抽取、转换、加载程序就可以对华为云上CloudTable、RDS、DWS等异构数据进行探索。



## SQL作业

[创建作业](#)

SQL作业为用户提供标准的SQL，通过可视化界面API、JDBC、ODBC、Beeline等多种接入方式对云上异构数据源进行查询分析，兼容CSV、JSON、Parquet、Carbon、ORC等主流数据格式。



## Spark作业

[创建作业](#)

Spark Serverless可为用户提供全托管式的Spark计算服务。用户可通过可视化界面和RESTful API提交作业，支持提交Spark Core、DataSet、Streaming、MLlib、GraphX等Spark全栈作业。

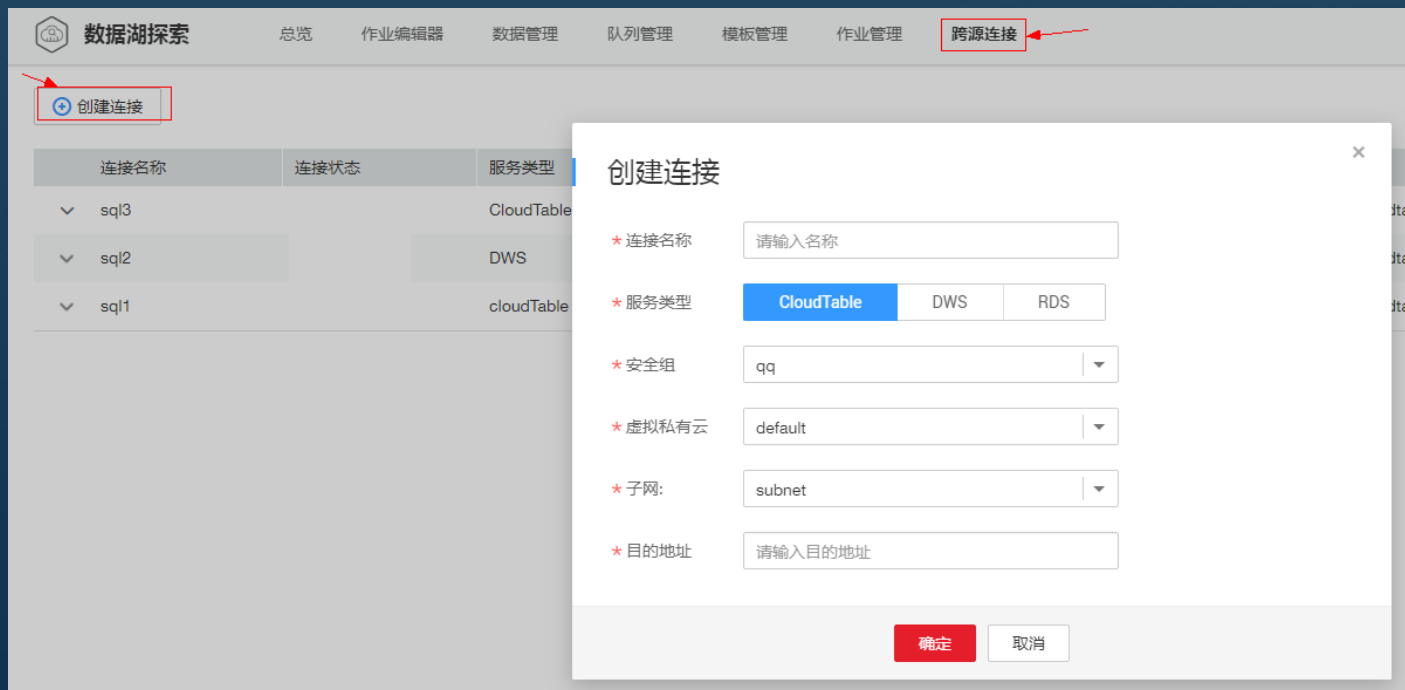


## 基因作业

[创建作业](#)

基因作业可为用户提供基于Spark的基因处理和分析能力。用户通过页面或者原子API提交基因任务，实现GATK流程，从原始测序数据(FASTQ或BAM格式)分析生成基因变异文件(VCF)。

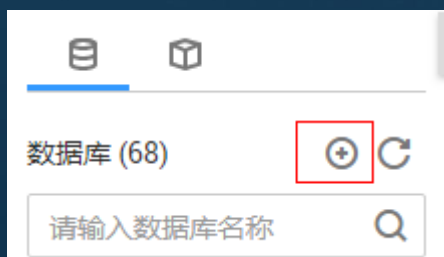
# 场景一：计算经销商B1每月Mate20销量与销售额



连接名称	连接ID	作业状态	连接地址	目的地址	进度	操作
test	0fd6d78a-ce82-4c89-ab9e-6b4a15547f59			dsadasda	10.0%	删除
test1	68233202-63bf-4f27-a99b-fa5f5a2dbae1			cloudtable-8636-zk2-ldQklrNa.mycloudtable.com:21...	75.0%	删除
test1	dc2cfa3b-a973-45db-91ab-89c067024d4b	已成功	to-ct-1174405096-qBugRpMb.datasource.com:2181	cloudtable-8636-zk2-ldQklrNa.mycloudtable.com:21...	100.0%	删除
zhuhs	f89ebc10-39d9-4ef5-8f43-adca1cea9c15			cloudtable-8636-zk2-ldQklrNa.mycloudtable.com:21...	10.0%	删除

# 场景一：计算经销商B1每月Mate20销量与销售额

## 创建数据库



### 创建数据库

提示：您还可以创建132个数据库。

\* 数据库名称

描述

10/100



# 场景一：计算经销商B1每月Mate20销量与销售额

## 创建DLI关联CT表，查询订单详情



The screenshot displays a SQL execution window with the following components:

- SQL Editor:** Contains a `CREATE` statement for a CloudTable (CT) named `CT_Bill`. The statement includes options for `ClusterId`, `TableName`, `RowKey`, `zkhost`, and `Cols`.
- Execution Bar:** Shows the queue as `default` and the database as `bill_manager`. It includes buttons for `执行` (Execute), `格式化` (Format), `设置` (Settings), and `更多` (More).
- Execution Result:** A message box indicating `执行成功` (Execution Successful) for a `DDL` (Data Definition Language) job.

```
1 CREATE table CT_Bill(userId string, billId string, month string, day int, num_nova int, num_P20 int, num_Mate20 int) using CLOUDTABLE OPTIONS (  
2   'ClusterId' = '51bf529e-31ee-4c7f-b639-d494d87a0cfc',  
3   'TableName' = 'BILL',  
4   'RowKey' = 'userId:4, billId:4',  
5   'zkhost' = 'cloudtable-uquery-test002-zk2-nUYFmJxZ.cloudtable.com:2181',  
6   'Cols' = 'month:time.month, day:time.day, num_nova:arts.nova, num_P20:arts.P20, num_Mate20:arts.Mate20'  
7 )
```

行 1, 列 116

Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

执行成功

作业类型: DDL

### 参数说明：

ClusterId: CT的集群ID

TableName: CT中的表名

RowKey: RowKey信息，格式为关联表中的字段名:长度

Zkhost: CT集群中的zk链接地址

Cols: 关联表与CT中实表的字段对应关系，格式为关联表字段:列簇.列名

# 场景一：计算经销商B1每月Mate20销量与销售额

## 创建DLI关联CT表，查询订单详情

SQL

队列 default 数据库 bill\_manager 执行 格式化 设置 更多

1 select \* from bill\_manager.ct\_bill where userId='0003' and billId='0001'

行 1, 列 73

Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

查询耗时4.58s, 已扫描0 KB.

userId	billId	month	day	num_nova	num_P20	num_Mate20
0003	0001	Oct	1	10	null	5

rowkey	time		arts		
	month	day	nova	P20	Mate20
00030001	Oct	1	10		5
00030002	Nov	1		5	5
00030003	Nov	2	2	1	1
00050001	Oct	1	20	5	5
00050002	Nov	20			5

# 场景一：计算经销商B1每月Mate20销量与销售额

## 创建关于订单商品Mate20的关联表

SQL

队列 default 数据库 bill\_manager 执行 格式化 设置 更多

```
1 CREATE table CT_Bill_mate20(userId string, billId string, month string, day int, num int) using CLOUDTABLE OPTIONS (  
2   'ClusterId' = '51bf529e-31ee-4c7f-b639-d494d87a0cfc',  
3   'TableName' = 'BILL',  
4   'RowKey' = 'userId:4, billId:4',  
5   'zkhost' = 'cloudtable-uquery-test002-zk2-nUYFmjxZ.cloudtable.com:2181,cloudtable-uquery-test002-zk1-uEqMaJ0L.cloudtable.com:2181,cloudtable-uquery-test002-zk3-12vpIJLj.cloudtable.com:2181',  
6   'Cols' = 'month:time.month, day:time.day, num:arts.Mate20'  
7 )
```

行 1, 列 11 Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

执行成功

作业类型: DDL

# 场景一：计算经销商B1每月Mate20销量与销售额

## Tips. 新增商品的操作

```
hbase(main):011:0* create 'BILL','time','arts'
2018-11-08 19:03:28,842 INFO [main] client.HBaseAdmin: Created BILL
0 row(s) in 1.2580 seconds

=> Hbase::Table - BILL
```



```
hbase(main):002:0> scan 'BILL'
ROW                                COLUMN+CELL
00030001                          column=arts:Mate20, timestamp=1541678508529, value=\x00\x00\x00\x05
00030001                          column=time:day, timestamp=1541678508529, value=\x00\x00\x00\x01
00030001                          column=time:month, timestamp=1541678508529, value=Oct
1 row(s) in 0.0960 seconds

hbase(main):003:0>
```

# 场景一：计算经销商B1每月Mate20销量与销售额

## 查询经销商B1每月Mate20销量

SQL

队列test\_ct数据库bill\_manager执行格式化设置更多

```
1 select
2   sum(num)
3 from
4   CT_Bill_mate20
5 where
6   userId = '0003'
7 group by
8   month;
```

行 8, 列 9

Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

查询耗时19.46s, 已扫描0 KB。

sum(num)
5
6

# 场景一：计算经销商B1每月Mate20销量与销售额

## 创建DWS关联表，关联商品表

SQL

队列 test\_ct 数据库 bill\_manager 执行 格式化 设置 更多

```
1 create table DWS_Arts(id int, name string, price int) using jdbc options (  
2   'url' = 'jdbc:postgresql://198.19.53.101:18001/postgres',  
3   'dbtable' = 'table_arts',  
4   'user' = 'dbadmin',  
5   'password' = 'your_password'  
6 )
```

行 5, 列 30

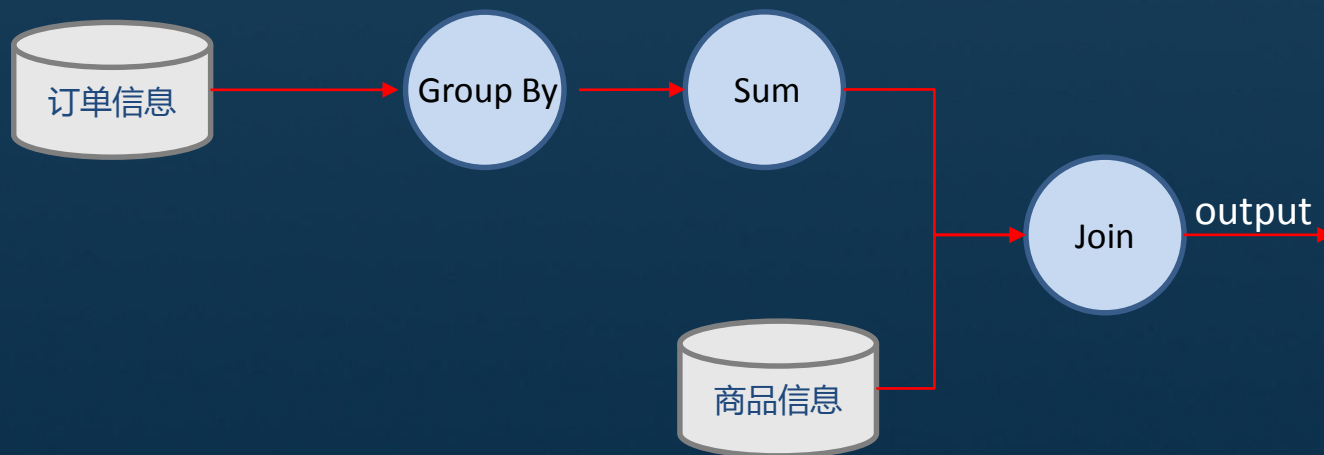
Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

执行成功

作业类型: DDL

## 场景一：计算经销商B1每月Mate20销量与销售额

联合DWS商品表与CT的订单信息表进行计算



# 场景一：计算经销商B1每月Mate20销量与销售额

联合DWS商品表与CT的订单信息表进行计算

SQL

队列test\_ct数据库bill\_manager执行格式化设置更多

```
1 select
2   t1.month,
3   t1.count_mate20,
4   t1.count_mate20 * DWS_Arts.price as total_price
5 from (
6   select 'Mate20' name, CT_Bill_mate20.month month, sum(CT_Bill_mate20.num) count_mate20
7   from CT_Bill_mate20
8   where CT_Bill_mate20.userId = '0003'
9   group by CT_Bill_mate20.month
10  ) t1, DWS_Arts
11 where DWS_Arts.name = t1.name
```

行 11, 列 30Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想

查询耗时4.88s, 已扫描0 KB。

month	count_mate20	total_price
Oct	5	25000
Nov	6	30000



## 场景二：预测经销商B1下月Mate20销量

**解决方案：**MLS是深度集成DLI的机器学习服务，为用户提供简易的工作流操作界面来实现数据的智能分析，如销售预测等。MLS支持通过OBS读取数据并结合创建的工作流，转换成Spark作业提交到DLI后台运行。



## 场景二：预测经销商B1下月Mate20销量

### 创建CSV格式的OBS外表



The screenshot displays a SQL development environment interface. At the top, there's a toolbar with icons for 'SQL', '队列' (Queue), 'default', '数据库' (Database) set to 'bill\_manager', and buttons for '执行' (Execute), '格式化' (Format), '设置' (Settings), and '更多' (More). The main text area contains a SQL statement for creating a table with CSV options:

```
1 create table obs_sale_metric_B1_mate20(  
2   month string,  
3   num int,  
4   total_price int  
5 ) using csv options ('path' = 's3a://bill-manager/sale-metrics')
```

Below the code editor, a status bar indicates '行 5, 列 65' (Line 5, Column 65) and provides keyboard shortcuts: 'Ctrl+R或Ctrl+Enter 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 回退, Tab: 自动联想'.

A message box at the bottom shows '执行成功' (Execution Successful) with a close button. Below this, it specifies '作业类型: DDL' (Job Type: DDL).

## 场景二：预测经销商B1下月Mate20销量

将场景二的结果导出到OBS外表中



The screenshot displays a SQL execution window with the following components:

- SQL Editor:** Contains a SQL query to insert data into the `bill_manager.obs_sale_metric_bi_mate20` table. The query selects data from `CT_Bill_mate20` and `DWS_Arts`, calculating the total price for Mate20 phones.
- Execution Bar:** Includes a "test\_ct" dropdown, a "数据库" (Database) dropdown set to "bill\_manager", and buttons for "执行" (Execute), "格式化" (Format), "设置" (Settings), and "更多" (More).
- Execution Result:** A message box at the bottom indicates "执行成功" (Execution Successful) and "作业类型: INSERT" (Job Type: INSERT).

```
1 insert into bill_manager.obs_sale_metric_bi_mate20 select
2   t1.month,
3   t1.count_mate20,
4   t1.count_mate20 * DWS_Arts.price as total_price
5 from (
6   select 'Mate20' name, CT_Bill_mate20.month month, sum(CT_Bill_mate20.num) count_mate20
7   from CT_Bill_mate20
8   where CT_Bill_mate20.userId = '0003'
9   group by CT_Bill_mate20.month
10  ) t1, DWS_Arts
11 where DWS_Arts.name = t1.name
```

行 11, 列 30

Ctrl+R或Ctrl+Enter: 执行SQL, Ctrl+F: 格式化SQL, Ctrl+Z: 后退, Tab: 自动联想

执行成功

作业类型: INSERT

# 场景二：预测经销商B1下月Mate20销量

## 确认结果

数据湖探索

总览 作业编辑器 数据管理 队列管理 模板管理 作业管理

购买cu时套餐包

作业类型: ALL 状态: 所有 日期:

创建时间	作业类型	状态	执行语句	运行时长	操作
2018/11/09 11:29:49 GMT+08:00	INSERT	已成功	insert into bill_manager.obs_sale_metric_b1_mate20 select t1.month, t1.count_mate20, t1.count_mate20 * DWS_Arts.price ...	3.22s	终止

队列名称: test\_ct 作业ID: 7378dcac-3e91-41c1-9e08-35b77bf977d6

创建时间: 2018/11/09 11:29:49 GMT+08:00 作业类型: INSERT

作业状态: 已成功 执行语句: insert into bill\_manager.obs\_sale\_metric\_b1\_mate20 select t1.month, t1.count\_...

桶列表 > bill-manager > sale-metrics

对象 已删除对象 碎片

对象是数据存储的基本单位，在OBS中文件和文件夹都是对象。您可以上传任何类型（文本、图片、视频等）的文件，并在桶中对这些文件进行管理。了解更多

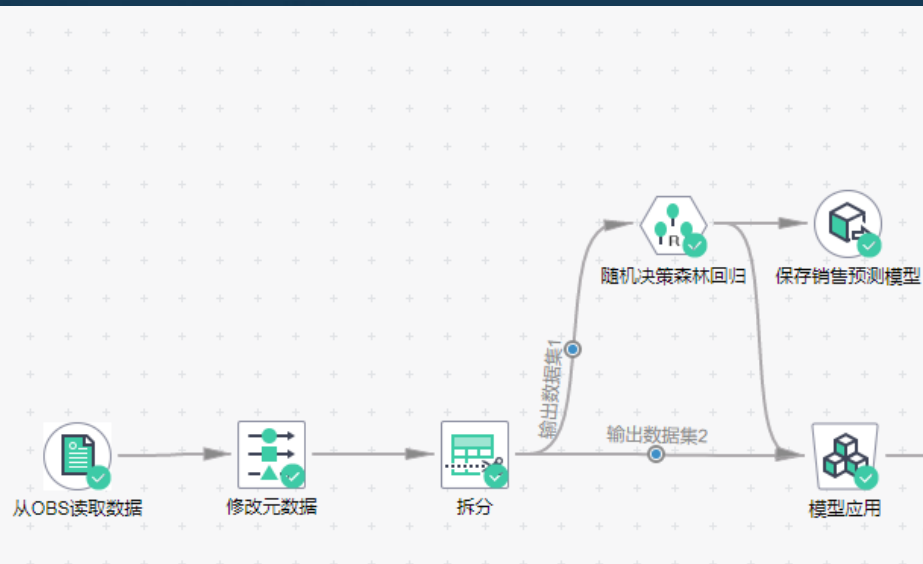
上传文件 新建文件夹 删除 恢复

输入对象名前缀搜索

名称	存储类别	大小	加密状态	恢复状态	最后修改时间	操作
返回上一级						
7378dcac3e9141c19e0835b77bf977d6	-	-	-	-	-	删除
.ignore	标准存储	0 Bytes	未加密	-	2018/11/09 10:05:44 GMT+...	下载 分享 更多

## 场景二：预测经销商B1下月Mate20销量

在机器学习服务（MLS）中创建销售预测实例，并对导出到OBS的数据进行模型训练



### 修改元数据

设置元数据



* 字段:	attr_1	角色:	Input	测量尺度:	Conti...	* 值:	10,11
* 字段:	attr_2	角色:	Target	测量尺度:	Conti...	* 值:	(5,6]

## 场景二：预测经销商B1下月Mate20销量

使用训练好的模型进行预测



## 场景二：预测经销商B1下月Mate20销量

### 查看预测结果

保存预测结果

\* 文件路径:  
/samples/SalesForeca

\* 文件名:  
predict

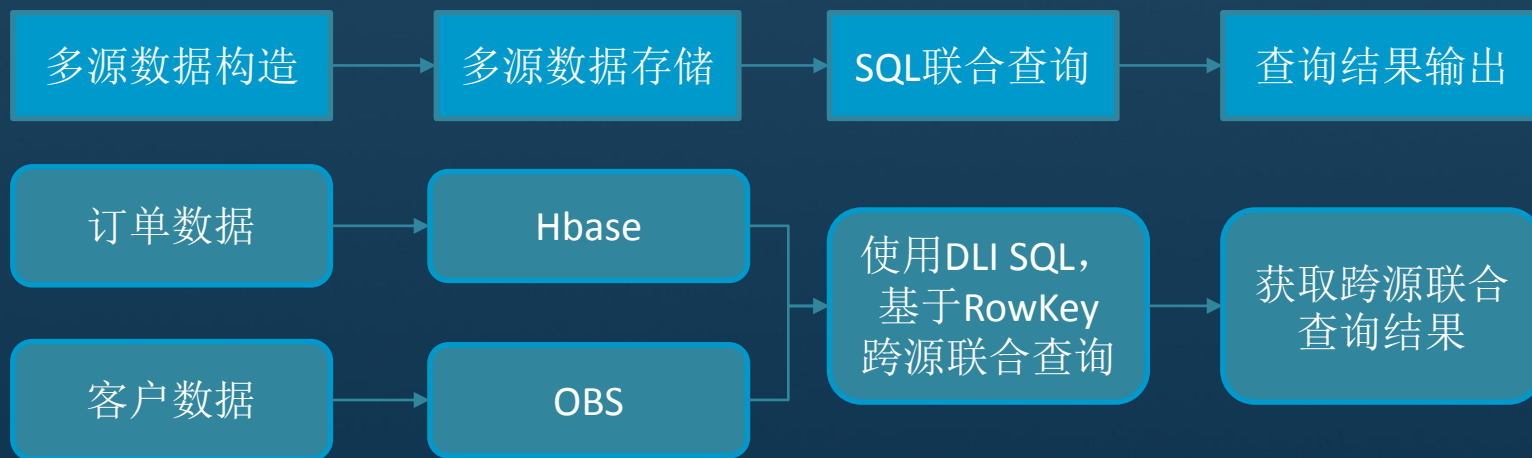
\* 文件格式:

名称	修改时间	大小
predict.csv	2018/11/09 14:12:25 GMT+08:00	1 KB
test.csv	2018/11/09 11:37:52 GMT+08:00	12 MB
train.csv	2018/11/09 11:37:52 GMT+08:00	19 KB

数据预览

7

# 使用数据湖探索（DLI）进行多源数据分析实践



**跨源查询SQL**

```
1 SELECT * FROM CUSTOMER_INFO JOIN ORDER_INFO WHERE
2 ORDER_INFO.CUST_ID='1' AND
3 ORDER_INFO.ORDER_ID='02' AND
4 CUSTOMER_INFO.ID=ORDER_INFO.CUST_ID
```

**查询结果**

查询耗时2.27s，已扫描0.94 KB。

id	name	location	phone	CUST_ID	ORDER_ID	SALE_DATE	ART_INFO
1	华为	杭州市滨江区江虹路4...	0571xxxxxxx	1	02	2018-01-02	YYY





# Thank You.

**Copyright©2018 Huawei Technologies Co., Ltd. All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.