



## 分布式数据库中间件

# 常见问题

文档版本 13

发布日期 2019-09-30

版权所有 © 华为技术有限公司 2019。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <http://e.huawei.com>

# 目录

<b>1 DDM 管理类</b>	<b>1</b>
1.1 如何选择和配置安全组	1
1.2 DDM 提供哪些高可靠保障	2
1.3 DDM 监控能力有哪些	3
1.4 连接 DDM 时密码多次错误被锁定该如何解决	3
<b>2 DDM 使用类</b>	<b>5</b>
2.1 本地环境是否可以连接 DDM 实例	5
2.2 向逻辑表中插入数据时出现报错	5
2.3 DDM 自身是否会存储业务数据	6
2.4 DDM 如何进行分片	6
2.5 DDM 逻辑表支持哪几种类型	6
2.6 如何实现读写分离	7
2.7 MySQL 连接 DDM 时出现乱码如何解决	7
2.8 DDM 实例是否支持跨 VPC 访问	8
2.9 如何解决 JDBC 驱动方式连接 DDM 异常问题	8
2.10 如何选择 JDBC 驱动方式的版本和参数	9
2.11 使用 DDM 事务模型有哪些注意事项	10
2.12 DDM 支持创建逻辑表的方式有哪些	10
2.13 DDM 实例的计费模式是否支持按需和包周期互转	12
2.14 导入数据到 DDM 后出现主键重复	13
<b>3 SQL 语法类</b>	<b>14</b>
3.1 DDM 是否支持 SQL 跨库访问	14
3.2 DDM 是否支持分布式 JOIN	14
3.3 如何进行 SQL 优化	14
3.4 DDM 不支持复杂 SQL 时如何处理	14
3.5 DDM 支持哪些 SQL	15
3.6 DDM 是否支持数据类型强制转换	15
3.7 DDM 是否支持外键	15
<b>4 RDS 相关类</b>	<b>16</b>
4.1 数据库表名是否区分大小写	16
4.2 RDS 实例参数配置 (binlog_format)	16
4.3 如何处理所关联的 RDS 实例异常	16

---

4.4 DDM 对 RDS 采用哪种检测机制.....	16
4.5 DDM 控制台显示的 RDS 实例名称/ID 和 RDS 控制台显示的 RDS 实例名称/ID 为什么不同.....	17
4.6 RDS 哪些高危操作会影响 DDM.....	18
4.7 导入 RDS 失败提示 DDM.0095 是什么原因.....	19
4.8 创建逻辑表时提示 RDS 分片上有残留的同名表怎么办.....	19
4.9 如何处理表中存在主键重复的数据.....	21
<b>A 修订记录.....</b>	<b>23</b>

# 1 DDM 管理类

## 1.1 如何选择和配置安全组

DDM实例采用了VPC和安全组等网络安全保护措施，以下内容帮助您正确配置安全组。

### 通过 VPC 内网访问 DDM 实例

DDM实例的访问和使用，包括客户端所在ECS访问DDM实例，以及DDM实例访问其关联的RDS实例。

除了ECS、DDM实例、RDS实例必须处于相同VPC之外，还需要他们的安全组分别配置了正确的规则，允许网络访问。

1. 建议ECS、DDM、RDS配置相同的安全组。安全组创建后，默认包含同一安全组内网络访问不受限制的规则。
2. 如果配置了不同安全组，可参考如下配置方式：

#### 说明

- 假设ECS、DDM、RDS分别配置了安全组：sg-ECS、sg-DDM、sg-RDS。
- 假设DDM实例服务端口为5066，RDS实例服务端口为3306。
- 以下规则，远端可使用安全组，也可以使用具体的IP地址。

ECS所在安全组需要增加图1-1中的规则，以保证客户端能正常访问DDM实例：

图 1-1 ECS 安全组配置



DDM所在安全组需要增加图1-2和图1-3中的规则，以保证能访问RDS，且被客户端访问。

图 1-2 DDM 安全组入方向配置



图 1-3 DDM 安全组出方向配置



RDS所在安全组需要增加图1-4中的规则，以保证能被DDM访问

图 1-4 RDS 安全组配置



## 1.2 DDM 提供哪些高可靠保障

### 数据完整性

DDM实例故障不会影响数据的完整性。

- 业务数据存储于RDS实例上分片中，DDM不存储业务数据。
- 逻辑库与逻辑表等配置信息存储在DDM数据库中，DDM数据库主备高可用。

## 容错机制

DDM采用的是RDS自身的容错机制，数据是否写入DB由RDS保证。RDS返回sql执行成功，DDM就认为成功。

## 1.3 DDM 监控能力有哪些

DDM当前提供云监控查看指标和DDM管理控制台监控管理的能力，您可以根据监控结果对数据库进行调优，保障性能。

### 云监控

云监控提供DDM实例节点的监控，可查看指标如表1-1所示，您可以配置告警规则。

表 1-1 云监控指标

监控参数名称	监控参数说明
CPU使用率	用于统计数据面core的CPU利用率。
内存使用率	用于统计数据面core的内存使用率。
网络输入吞吐量	用于统计数据面core平均每秒的输入流量。
网络输出吞吐量	用于统计数据面core平均每秒的输出流量。
连接数	用于统计数据面服务core的连接数。
请求数	用于统计数据面服务core的每秒请求数。
慢SQL数	用于统计数据面服务core的慢SQL条数。
读占比	用于统计每个DDM节点的总体的读写比例。

## DDM 管理控制台监控管理

管理控制台提供了对DDM实例的监控管理，包括读占比查看与慢SQL查看，用户可以根据监控反馈结果，对数据库进行调优。

- 步骤1** 登录管理控制台。
  - 步骤2** 在导航上选择“数据库 > 分布式数据库中间件”，进入总览页面。
  - 步骤3** 单击左侧菜单栏的“DDM实例管理”，进入“DDM实例管理”页面。
  - 步骤4** 选择需要操作的实例，单击右边操作栏“更多 > 查看监控”链接，进入实例监控管理页面。
- 结束

## 1.4 连接 DDM 时密码多次错误被锁定该如何解决

连接DDM时，如果多次输入错误密码，系统将会自动锁定20分钟，DDM支持在管理控制台手工解除锁定。

1. 登录管理控制台。
2. 在导航上选择“数据库 > 分布式数据库中间件”，进入总览页面。
3. 单击左侧菜单栏的“DDM实例管理”，进入“DDM实例管理”页面。
4. 单击DDM实例名称，进入实例基本信息页面。
5. 在实例基本信息页面，选择“账号管理”选项卡，查看账号管理页面。
6. 在需要解锁的逻辑库右侧操作栏选择“更多 > 解锁”。

 说明

DDM实例状态如果处于非运行中将无法解锁。



# 2 DDM 使用类

## 2.1 本地环境是否可以连接 DDM 实例

本地环境不能直接连接DDM实例，以下三种场景可以实现本地环境连接。

1. DDM实例公有云采用虚拟私有云（VPC）管理各服务的网络安全，用户创建的DDM实例，只允许被与实例处于相同虚拟私有云的弹性云服务器访问。
2. 用户可以通过网络代理转发的方式，用一台能与DDM实例网络互通的弹性云服务器（ECS）做中转，实现本地环境连接DDM实例，[查看参考](#)。
3. 用户还可以通过本地与华为云专线打通的方式，实现本地环境直接连接DDM实例。用户通过这种方式连接DDM实例需要在安全组出入方向开通5066端口，[查看参考](#)。

## 2.2 向逻辑表中插入数据时出现报错

### 问题描述

连接DDM逻辑库，用source命令执行.sql文件向逻辑表中插入500万数据时，出现“MySQL server has gone away”的错误提示信息如下。

```
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 37
Current database: db_gm
ERROR 2006 (HY000): MySQL server has gone away
mysql>
```

### 原因分析

DDM支持大小不超过16MB的单条SQL语句，本次插入数据大小为50MB，大于最大允许值16MB。

## 2.3 DDM 自身是否会存储业务数据

DDM实例自身不存储客户业务相关数据，客户业务相关数据都存储在RDS实例上分片中。

DDM节点目前硬盘配置为100G，主要用来存储日志和一些临时文件，日志和临时文件会做定期清理，空间足够使用。

## 2.4 DDM 如何进行分片

在分布式数据库中，可以通过分片存储方式，轻松解决大数据量单表容量达到单机数据库存储上限的瓶颈，因此创建逻辑库和逻辑表时，需要根据实际情况确定逻辑表是否进行分片以及按什么规则分。

### 说明

分片存储后，需要尽量避免跨库JOIN操作带来的性能与资源消耗问题。

- 逻辑表是否分片

DDM逻辑表支持全局表、分片表、单表三种类型。用户可以按照数据表的实际使用需求，选择最合适的逻辑表类型创建。

- 单表只在第一个分片创建表以及存储数据。
- 全局表在每一个分片创建表并且存储全量数据。
- 分片表在每一个分片创建表，数据按照拆分规则分散存储在分片中。

- 按什么规则分

逻辑表的拆分键选择非常重要。建议按实际业务场景选择拆分键，不同逻辑表，如果具有E-R关系，建议选择相同字段做拆分键，避免跨库JOIN操作。

在实际使用中，请结合以下建议评估是否进行分片：

- 数据量在1000万以下的表，不建议分片。通过建立合适的索引，采取读写分离策略，单表也可以很好的解决性能问题。
- 数据量在1000万以上的表，建议分片。将数据分片存储后，既能解决单张表容量过大带来的性能瓶颈，同时提高并发支持。注意要选择合适的拆分键，提前做好规划。
- 业务读取尽量少用多表JOIN，同一个事务避免跨分片。
- 查询条件尽量带上拆分键，避免全分片表扫描。

## 2.5 DDM 逻辑表支持哪几种类型

DDM逻辑表支持分片表、全局表、单表和普通表四种类型。

- 分片表

DDM实例在RDS实例上创建多个物理库，每一个物理库称为分片，在分片上建立的表，叫分片表（不包括单表和全局表）。数据按照分片规则被分配到不同的分片上，分片上的数据支持分布式并行计算。

- 全局表

将一些数据量小且更新操作不频繁的数据表，如字典表、配置表等，在所有分片上都存储全量数据，提升JOIN效率，这类表叫全局表。

- 单表  
当表数据量不大，且不需要与分片表进行连接查询时，可以只在DDM的默认分片上建表与存储数据，这类表称为单表。  
单表只能通过MySQL客户端或者应用程序连接逻辑库创建。
- 普通表  
DDM将未配置sharding的表，以单表的形式存储在其中一个默认分片上。

表 2-1 各类型逻辑表创建方法

逻辑库类型	逻辑表类型	建表方法
拆分	分片表	通过管理控制台创建。
拆分	全局表	
拆分	单表	通过sql客户端或者应用程序直接发起创建请求，建表语句兼容MySQL的CREATE TABLE语法。
非拆分	普通表	

## 2.6 如何实现读写分离

读写分离指对RDS实例配置了一个或者多个只读实例后，DDM可以将写操作分发给主实例执行，将读操作分发给只读实例执行。读写分离极大地提高了数据库的并发性能。

读写分离特性一般由DDM内部实现，用户无需在应用代码上做设置。

DDM的读写分离模式为：默认事务内的SQL都会走主实例，非事务内的SQL会根据DDM内置的策略选择只读实例执行。

DDM也支持用户在SQL中自定义读写分离策略，只需要在SQL前加上指定标签：

- `/*balance*/` 标签表示事务内走只读实例，事务外的SQL会走主实例，SQL举例如下：  
`/*balance*/ select a.company_id, a.name from customer a where a.company_id=1;`
- `/*#mycat:db_type=master*/` 标签表示忽略事务状态，强制走主实例，SQL举例如下：  
`/*#mycat:db_type=master*/ select b.recordid, b.destination from travelrecord b;`
- `/*#mycat:db_type=slave*/` 标签表示忽略事务状态，强制走只读实例，SQL举例如下：  
`/*#mycat:db_type=slave*/ select b.recordid, b.destination from travelrecord b;`

## 2.7 MySQL 连接 DDM 时出现乱码如何解决

MySQL连接的编码和实际的编码不一致，可能导致DDM解析时出现乱码。

通过“`default-character-set=utf8`”指定客户端连接的编码即可。

如下所示：

```
mysql -h127.0.0.1 -P5066 -Dbase --default-character-set=utf8 -uddmuser -p
```

## 2.8 DDM 实例是否支持跨 VPC 访问

一般情况下，不同VPC间网络不互通，不在同一VPC下的弹性云服务器无法访问DDM实例。

可以通过创建VPC对等连接，将两个VPC的网络打通，实现跨VPC访问DDM实例。

关于创建和使用VPC对等连接，请参考[VPC对等连接说明](#)。

用户通过VPC对等访问DDM实例时，除了满足VPC对等网跨VPC访问的约束之外，还存在如下约束：

同VPC跨子网互通需求上线之前创建的实例（即2018年5月10日24:00前创建的实例），不支持跨VPC访问。

## 2.9 如何解决 JDBC 驱动方式连接 DDM 异常问题

MySQL驱动（JDBC）通过Loadbalance方式连接DDM，在某些场景下连接切换时会陷入死循环，最终导致栈溢出。

### 问题定位

1. 查看APP日志，定位异常原因。

例如，从以下日志中分析出异常最终原因为栈溢出。

```
Caused by: java.lang.StackOverflowError
    at java.nio.HeapByteBuffer.<init>(HeapByteBuffer.java:57)
    at java.nio.ByteBuffer.allocate(ByteBuffer.java:335)
    at java.nio.charset.CharsetEncoder.encode(CharsetEncoder.java:795)
    at java.nio.charset.Charset.encode(Charset.java:843)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2362)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:2344)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:568)
    at com.mysql.jdbc.StringUtils.getBytes(StringUtils.java:626)
    at com.mysql.jdbc.Buffer.writeStringNotNull(Buffer.java:670)
    at com.mysql.jdbc.MySqlIO.sqlQueryDirect(MySqlIO.java:2636)
```

2. 分析溢出源。

例如，从以下日志可以分析出，溢出原因为驱动内部陷入死循环。

```
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:344)
at
com.mysql.jdbc.LoadBalancedAutoCommitInterceptor.postProcess(LoadBalancedAutoCommitInterceptor.java:104)
at com.mysql.jdbc.MySqlIO.invokeStatementInterceptorsPost(MySqlIO.java:2885)
at com.mysql.jdbc.MySqlIO.sqlQueryDirect(MySqlIO.java:2808)
at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2483)
at com.mysql.jdbc.ConnectionImpl.setReadOnlyInternal(ConnectionImpl.java:4961)
at com.mysql.jdbc.ConnectionImpl.setReadOnly(ConnectionImpl.java:4954)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:381)
at com.mysql.jdbc.MultiHostConnectionProxy.syncSessionState(MultiHostConnectionProxy.java:366)
at
com.mysql.jdbc.LoadBalancedConnectionProxy.pickNewConnection(LoadBalancedConnectionProxy.java:344)
```

3. 查看使用的MySQL版本，为5.1.44。

查看该版本源代码，发现获取连接时，**LoadBalance**会根据负载均衡策略更新连接，并将老连接的配置复制给新连接，在新连接**AutoCommit**为**true**，新连接部分

参数和老连接不一致，`loadBalanceAutoCommitStatementThreshold`参数没有配置的场景下，会陷入死循环，更新连接函数调用同步参数函数，同步参数又调用更新连接，最终导致栈溢出。

## 解决方法

在连接DDM的URL添加

`loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10`参数。

```
//使用负载均衡的连接示例
//jdbc:mysql:loadbalance://ip1:port1,ip2:port2..ipN:portN/{db_name}
String url = "jdbc:mysql:loadbalance://192.168.0.200:5066,192.168.0.201:5066/db_5133?loadBalanceAutoCommitStatementThreshold=5&retriesAllDown=10";
```

- `loadBalanceAutoCommitStatementThreshold`: 表示连接上执行多少个语句后会重新选择连接。

假设`loadBalanceAutoCommitStatementThreshold`设为5，则当执行5个sql后（Queries或者updates等），将会重新选择连接。若为0表示“粘性连接，不重新选择连接”。关闭自动提交时（`autocommit=false`）会等待事务完成再考虑是否重新选择连接。

## 2.10 如何选择 JDBC 驱动方式的版本和参数

DDM暂不支持使用5.1.46版本的JDBC驱动连接DDM，建议您使用以下版本的JDBC驱动：5.1.35-5.1.45。

JDBC驱动下载地址：<https://dev.mysql.com/doc/index-connectors.html>。

JDBC URL中推荐参数如表2-2所示。

表 2-2 参数

参数名称	参数说明	推荐取值
ip:port	连接地址和端口，用于连接DDM。	在DDM管理控制台DDM实例管理中查看连接地址。
db_name	连接逻辑库名称。	在DDM管理控制台，DDM实例管理 > 逻辑库管理下查看逻辑库名称。
loadBalanceAutoCommitStatementThreshold	<p>表示连接上执行多少个语句后会重新选择连接。</p> <ul style="list-style-type: none"> <li>● 若取值为5，则当执行5个sql后（Queries或者updates等），将会重新选择连接。</li> <li>● 若取值为0，则表示“粘性连接，不重新选择连接”。</li> </ul> <p>关闭自动提交时（<code>autocommit=false</code>）会等待事务完成再考虑是否重新选择连接。</p>	5

参数名称	参数说明	推荐取值
loadBalanceHostRemovalGracePeriod	设置主机从负载均衡连接中移除的宽限时间。	15000
loadBalanceBlacklistTimeout	设置服务器在全局黑名单中存留的时间。	60000
loadBalancePingTimeout	使用负载均衡连接时，等待每个负载均衡连接ping响应的毫秒数。	5000
retriesAllDown	当所有的连接地址都无法连接时，轮询重试的最大次数。 重试次数达到阈值仍然无法获取有效连接，将会抛出SQLException。	10
connectTimeout	和数据库服务器建立socket连接时的超时。 单位：毫秒，0表示永不超时，适用于JDK 1.4及更高版本。	10000
socketTimeout	socket操作（读写）超时。 单位：毫秒，0表示永不超时	根据业务实际情况合理配置。

## 2.11 使用 DDM 事务模型有哪些注意事项

1. 事务补偿仅支持对事务中的insert、update和delete语句的补偿。
2. 事务隔离性在最终一致性下，默认极端情况下会有脏读，可以通过对select加**for update**或者**lock in share mode**避免脏读。
3. 强一致性模型下对所有select都会自动加锁，避免脏读，对系统性能可能有较大影响，请根据业务场景进行选择。
4. 对应autocommit=true时，若单条sql（insert、update、delete等）执行跨了多个分片，请在事务里执行。
5. 对DDL如truncate table等，mysql会隐式提交，无法做到事务一致性，建议单独分开。
6. **insert into ...on duplicate key update**语句仅支持对唯一主键方式，对联合主键或其他唯一索引导致的update行为不支持，无法保证一致性。

## 2.12 DDM 支持创建逻辑表的方式有哪些

DDM目前支持单表、全局表、分片表，建表方式支持情况如表2-3所示。

表 2-3 建表方式

表格类型	Console控制台创建	MySQL连接DDM创建
单表	不支持	支持
全局表	支持	支持

表格类型	Console控制台创建	MySQL连接DDM创建
分片表	支持	支持

本文主要介绍使用DDL语句进行建表的语法以及样例。

建表前请确保已经使用MySQL连接DDM。

- 单表：只在第一个分片创建表以及存储数据。
- 全局表：在每一个分片创建表并且存储全量数据。
- 分片表：在每一个分片创建表，数据按照拆分规则分散存储在分片中。

## 单表

建一张单表：

```
CREATE TABLE single_table(
  id int,
  name varchar(30),
  primary key(id)
);
```

## 分片表

- 假设已经建好的逻辑库分片数为8，创建分片表，拆分算法为hash，拆分键为id。

```
CREATE TABLE split_table(
  id int,
  name varchar(30),
  primary key(id)
) dbpartition by hash(id);
```

- 假设已经建好的逻辑库分片数为8，创建分片表，按照日期进行hash。

```
CREATE TABLE table_date(
  id int auto_increment,
  name varchar(30),
  insert_date datetime not null,
  primary key(id)
) dbpartition by hash(day(insert_date));
```

- 当拆分算法不指定任何拆分字段时，系统默认使用主键作为拆分字段。以下示例将介绍如何使用主键。

使用主键当拆分键：

```
CREATE TABLE prmkey_table(
  id int,
  name varchar(30),
  primary key(id)
) dbpartition by hash();
```

- range分片建表，

```
CREATE TABLE t_range_1(
  id int,
  dt datetime,
  primary key(id)
)
dbpartition by range(id)
{
  0-10000000=0,
  10000001-15000000=1,
  15000001-19999999=2,
  20000000=3,
```

```
default=7  
};
```

## 全局表

子句**BROADCAST**用来指定创建全局表。全局表是指将这个表复制到每个分片上，在分片上通过同步机制实现数据一致，有秒级延迟。

优势是可以将 JOIN 操作下推到底层的 RDS (MySQL)，来避免跨库 JOIN。

```
CREATE TABLE global_table(  
  id int,  
  name varchar(30),  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 BROADCAST;
```

## 全局序列

定义：**AUTO\_INCREMENT [by db|time 步长]**，默认为DB。

起始值：**AUTO\_INCREMENT = 1**，默认为1。

```
CREATE TABLE tbl (  
  col1 INT NOT NULL AUTO_INCREMENT [by db|time] 1000,  
  col2 VARCHAR(16),  
  PRIMARY KEY(col1)  
) AUTO_INCREMENT = 1 DBPARTITION BY HASH(col1);
```

## 其他 MySQL 建表属性

您在建表的同时还可以指定其他的MySQL建表属性，例如：

```
CREATE TABLE multi_db_multi_tbl(  
  id int,  
  name varchar(30),  
  primary key(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 dbpartition by hash(id);
```

## 2.13 DDM 实例的计费模式是否支持按需和包周期互转

DDM实例的计费模式支持按需和包周期互转。

### 说明

只有状态为“运行中”的实例才能进行按需和包周期互转操作。

**步骤1** 登录DDM管理控制台。

**步骤2** 在导航上选择“数据库 > 分布式数据库中间件”，进入总览页面。

**步骤3** 单击左侧菜单栏的“DDM实例管理”，进入“DDM实例管理”页面。

**步骤4** 在需要转换计费模式的实例右侧，选择“更多 > 转按需/转包周期”。

**步骤5** 跳转至费用中心界面进行操作，详情请参见[按需变更为包周期](#)和[包周期变更为按需](#)。

----结束



## 2.14 导入数据到 DDM 后出现主键重复

### 问题描述

数据导入到DDM后，出现主键重复的现象。

### 问题分析

DDM全局序列服务实现分布式环境下唯一键、主键等数据的全局唯一性。当向DDM导入数据后，由于记录含有主键ID，全局序列无法变更，新插入数据主键ID值大于当前值，导致主键冲突，从而出现主键重复的现象。

### 解决方案

创表时设置自增起始值，并确保起始值大于导入数据自增键的最大值。

# 3 SQL 语法类

## 3.1 DDM 是否支持 SQL 跨库访问

不支持SQL通过带数据库名称的方式跨逻辑库访问。

DDM会自动去除SQL中的库名，比如select \* from dn1.item,会自动改成select \* from item。

## 3.2 DDM 是否支持分布式 JOIN

DDM支持分布式JOIN。

- 表设计时，增加字段冗余
- 支持跨分片的JOIN，主要实现的方式有三种：全局表，ER分片和ShareJoin。
- DDM目前禁止多个表的跨库update和delete。

## 3.3 如何进行 SQL 优化

- 尽量避免使用LEFT JOIN或RIGHT JOIN，建议使用INNER。
- 在使用LEFT或RIGHT JOIN时，ON会优先执行，WHERE条件在最后执行，所以在使用过程中，条件尽可能在ON语句中判断，减少WHERE的执行。
- 尽量少用子查询，改用JOIN，避免大表全表扫描。

## 3.4 DDM 不支持复杂 SQL 时如何处理

对读写分离等场景，如果某些复杂SQL不支持，可通过注释方式透传到RDS上。

将/\*\* mycat:sql=select 1 \*/放在原SQL最开头处，注意最开头处不能有空格。

### 说明

如果是mysql命令行连接DDM，需要在命令行加上-c，否则mysql客户端会把注释自动去除。

## 3.5 DDM 支持哪些 SQL

DDM对SQL语法的支持情况请参见SQL兼容性和[SQL语法限制](#)。

## 3.6 DDM 是否支持数据类型强制转换

数据类型转换属于高级用法，DDM对SQL的兼容性会逐步完善，如有需要请提工单处理。

## 3.7 DDM 是否支持外键

### 操作场景

对于分布式数据库中间件，目前仅单表支持外键，分片表和全局表不支持外键。

### 操作步骤

**步骤1** 关闭外键校验。

```
SET FOREIGN_KEY_CHECKS=0;
```

**步骤2** 创建表。

```
create table tabletext;
```

**步骤3** 开启外键校验。

```
SET FOREIGN_KEY_CHECKS=1;
```

----结束

### 使用样例

```
SET FOREIGN_KEY_CHECKS=0;

create table t_order_detail (
    detail_id int(10) primary key not null,
    goods_name varchar(13) not null,
    goods_num int not null,
    order_id int not null,
    foreign key(order_id) references t_order(order_id)
);

create table t_order (
    order_id int not null primary key,
    order_type tinyint(1),
    order_user varchar(32) not null
);

SET FOREIGN_KEY_CHECKS=1;
```

# 4 RDS 相关类

## 4.1 数据库表名是否区分大小写

DDM和RDS的MySQL版本默认对数据表名和序列名称不区分大小写，DDM在导入RDS时会将`lower_case_table_names`设置为1。

建议不要修改RDS实例中的参数设置，`lower_case_table_names`取值固定为1，表示不区分大小写，且以小写的形式保存。

## 4.2 RDS 实例参数配置（`binlog_format`）

DDM在导入RDS时会将`binlog_format`设置为`row`，表示将binlog记录成每一行数据被修改的形式，包括修改前和修改后的数据。

建议不要对该参数进行修改，否则可能影响您的正常使用。

## 4.3 如何处理所关联的 RDS 实例异常

DDM针对RDS实例异常的处理机制为：

- DDM采用心跳检测机制保持对RDS实例的状态监控。
- DDM不管理RDS实例。当路由分发到故障实例上，则返回错误，如果路由没有分发到故障实例，则SQL请求正常返回。

因此，建议购买RDS实例类型为主备，以增强RDS实例的高可用性。

## 4.4 DDM 对 RDS 采用哪种检测机制

DDM采用基于sql语句的定时心跳检测机制，属于业务层检测。

- 当RDS只读实例异常时（包括RDS重启或者SQL执行异常），DDM会自动屏蔽该故障节点，确保业务不受影响。
- 当RDS主备实例节点都异常时，DDM会自动重试，待故障恢复后再恢复业务。
- 当RDS主备实例或只读实例故障时，由于DDM不会存储业务数据，所以DDM会返回错误信息给应用程序。

## 4.5 DDM 控制台显示的 RDS 实例名称/ID 和 RDS 控制台显示的 RDS 实例名称/ID 为什么不同

DDM基于RDS的OpenAPI，RDS实例名称/ID为导入RDS时从RDS获取的RDS实例名称/ID，DDM显示的RDS实例名称采用RDS实例名称加后缀方式命名，RDS实例名称/ID可在RDS的“查看监控”中查阅。

例如：

RDS上的实例名称为"rds-infopq"（图4-1），对应DDM上显示的RDS实例名称为“rds-infopq\_node0”（图4-3）。

DDM上显示的RDS实例名称/ID（图4-3），与RDS“查看监控”中的节点名称/ID（图4-2）相对应。

### RDS 实例 ID

在RDS管理控制台，左侧导航单击“实例管理”，图4-1红框所示即为RDS实例ID。

图 4-1 RDS 实例名称/ID

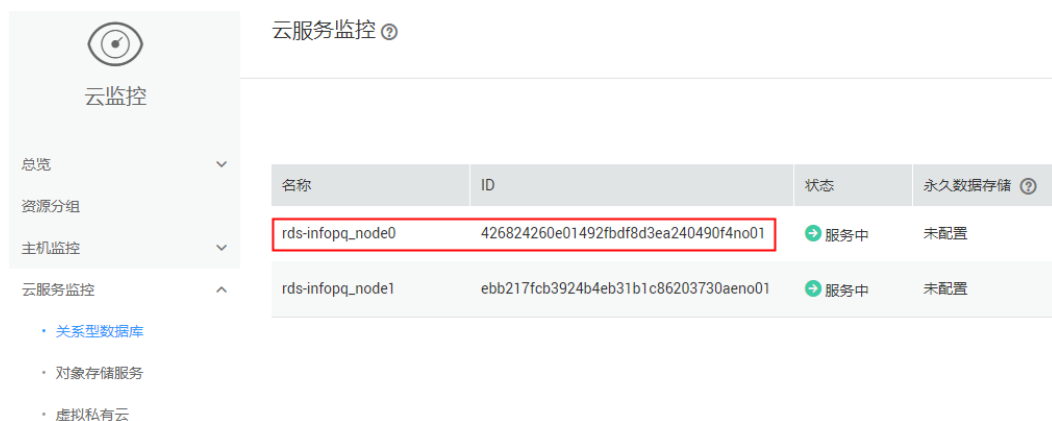


### RDS 节点名称/ID

在RDS管理控制台，在任一实例右操作列选择“更多 > 查看监控”，并在左侧导航选择“关系型数据库”。

图4-2中红框所示即为RDS节点名称/ID。

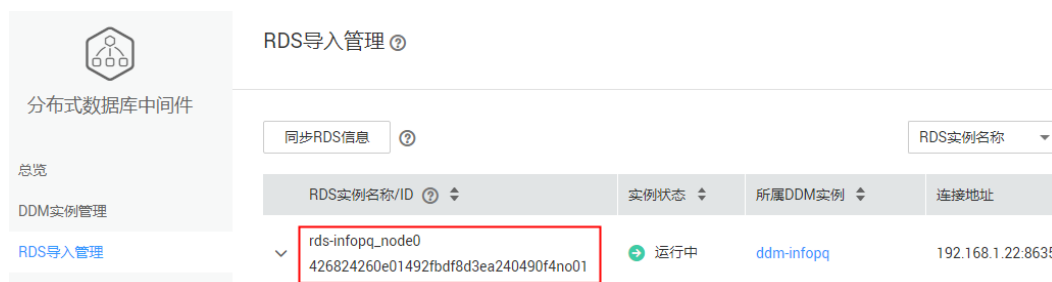
图 4-2 RDS 节点名称/ID



## DDM 实例名称/ID

DDM中的RDS实例名称/ID（图4-3）为实例名称所对应的RDS节点名称/ID（图4-2）。

图 4-3 DDM 实例名称/ID



## 4.6 RDS 哪些高危操作会影响 DDM

RDS相关高危操作如表4-1所示。

表 4-1 RDS 高危操作

操作类别	操作	操作影响
RDS控制台操作类	删除RDS实例	RDS实例删除后，DDM关联该RDS实例的逻辑库、逻辑表都无法使用。
	切换RDS主备实例	切换主备实例可能造成短时间内的RDS服务闪断，并有可能在主备同步时延过大的情况下，导致少量数据丢失。 <ul style="list-style-type: none"> <li>● RDS实例主备切换过程中，DDM将无法进行创建逻辑库、逻辑表等操作。</li> <li>● RDS实例主备切换后，DDM中RDS实例ID不变，固定为首次导入RDS实例时的RDS节点ID。</li> </ul>
	重启实例	重启过程中，RDS实例将不可用，DDM业务将会受影响。
	重置密码	在RDS重置RDS控制台重置密码后，需在DDM同步重置密码，否则将导致无法DDM和RDS无法连接。

操作类别	操作	操作影响
	修改数据库端口	修改RDS数据库端口后，RDS会自动重启RDS实例（如果是主备模式，重启主实例同时也会重启备实例），如果不在DDM同步RDS信息会导致DDM无法连接RDS。
	修改参数组	其中如下参数为固定值，如果修改，将会影响DDM正常运行。 <ul style="list-style-type: none"> <li>● 对数据表名和序列名称不区分大小写，“lower_case_table_names”固定为“1”。</li> <li>● 扩容场景，必须将“local_infile”配置为“ON”。</li> </ul>
	修改安全组	将导致DDM无法连接RDS。
	修改VPC	DDM与RDS不在同一VPC中将导致无法互通。
	恢复	恢复数据可能会破坏数据完整性。
RDS客户端类	删除DDM创建的物理库	删除物理库后，原数据将会丢失，新数据将无法写入。
	删除DDM创建的物理账号	删除物理账号后将无法在DDM上创建逻辑表。
	删除DDM创建的物理表	删除物理表后，将导致DDM数据丢失，DDM后续无法正常使用该逻辑表。
	修改DDM创建的物理表名	将导致DDM无法获取该逻辑表的数据，且后续无法正常使用。
	修改记录	如修改全局表记录，将会影响各分片数据一致性。
	修改白名单	需要确保DDM在RDS的白名单内，否则DDM将无法访问RDS。

## 4.7 导入 RDS 失败提示 DDM.0095 是什么原因

可能是RDS的状态不允许设置参数或者重启RDS，需要先移除RDS，等RDS状态为运行中后再导入。

## 4.8 创建逻辑表时提示 RDS 分片上有残留的同名表怎么办

本章节分别描述在控制台和客户端创建逻辑表时提示RDS分片上有残留的同名表的处理方法。

## 控制台创建表

- 问题现象：  
在DDM管理控制台创建逻辑表时提示，RDS实例分片中已存在同名数据表。错误码：DDM.4208。
- 解决方法：
  - a. 若确认该表并未使用，可先执行删除，同时勾选“删除关联RDS实例的数据”，再重新创建。



- b. 若删除时未勾选“删除关联RDS实例的数据”，可在重建时勾选“覆盖RDS分片上残留的同名表”。



## 客户端创建表

- 问题现象：



通过MySQL客户端连接到DDM创建分片表或全局表，返回表已存在的提示，如下：

```
mysql> CREATE TABLE test_broadcast_all_01 (id int, name varchar(30), primary key(id))  
BROADCAST;  
ERROR 1005 (HY000): Can't create table 'test_broadcast_all_01'. "A data table with the same  
name already exists in the shard of the RDS DB instance."
```

● 解决方法：

- a. 若确认该表并未使用，可使用**DROP TABLE**语句先删除，**DROP TABLE**会删除当前逻辑库所有关联分片上残留的同名表。

```
mysql> drop table test_broadcast_all_01;  
Query OK, 0 rows affected (2.45 sec)
```

- b. 重新创建该表。

```
mysql> CREATE TABLE test_broadcast_all_01 (id int, name varchar(30), primary key(id))  
BROADCAST;  
Query OK, 0 rows affected (3.64 sec)
```

## 4.9 如何处理表中存在主键重复的数据

### 场景一

DDM实例的逻辑表中已存在主键数据类型边界值的记录，如果插入的数据超过主键数据类型的范围，表中会出现主键重复的数据。

### 处理方法

**步骤1** 登录华为云管理控制台。

**步骤2** 在RDS的“实例管理”页面，查找DDM实例对应的RDS实例，单击目标RDS实例名称，进入实例的“基本信息”页面。

**步骤3** 在基本信息页面的左侧导航栏中选择“参数修改”。

**步骤4** 在“参数组”页签搜索“sql\_mode”，单击“值”列中的下拉框，勾选“STRICT\_ALL\_TABLES”或“STRICT\_TRANS\_TABLES”方式，单击“保存”。

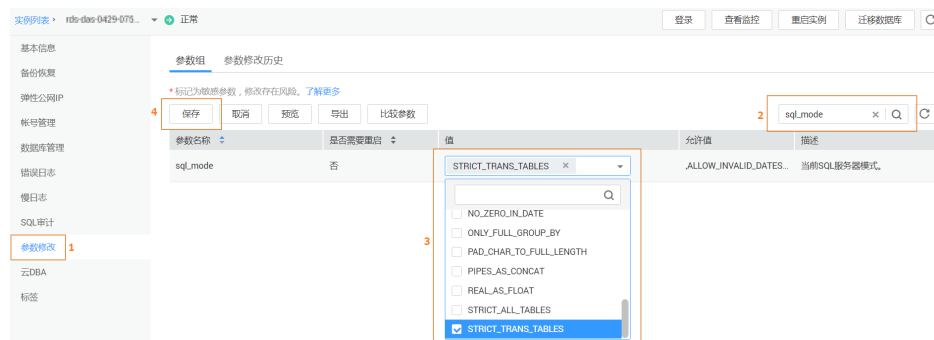
 说明

“STRICT\_ALL\_TABLES”和“STRICT\_TRANS\_TABLES”方式属于严格模式。严格模式控制MySQL如何处理非法或丢失的输入值。

- 非法：数据类型错误或超出范围。
- 丢失：如果某列定义为非空列且没有DEFAULT值，当新插入的行不包含该列时，该行记录丢失。

关于“sql\_mode”更多信息，请参考[Server SQL Modes](#)。

图 4-4 修改实例参数



**步骤5** 在“DDM实例管理”页面，重启DDM实例。

----结束

## 场景二

DDM实例的分片表（hash\range\mod）联合主键，如果拆分键数据中有插入0和1的话，一定会出现重复主键1。

## 处理方法

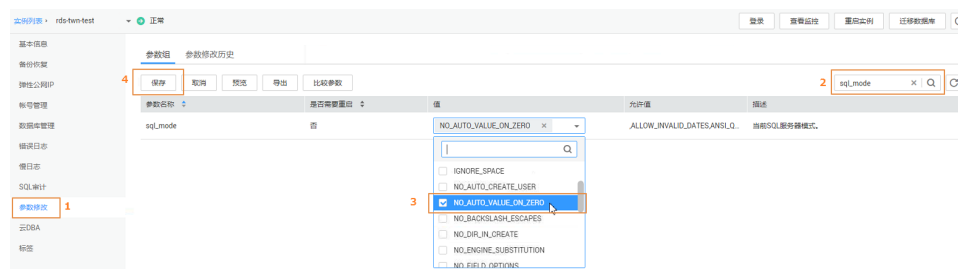
**步骤1** 登录华为云管理控制台。

**步骤2** 在RDS的“实例管理”页面，查找DDM实例对应的RDS实例，单击目标RDS实例名称，进入实例的“基本信息”页面。

**步骤3** 在基本信息页面的左侧导航栏中选择“参数修改”。

**步骤4** 在“参数组”页签搜索“sql\_mode”，单击“值”列中的下拉框，勾选“NO\_AUTO\_VALUE\_ON\_ZERO”方式，单击“保存”。

图 4-5 修改实例参数



**步骤5** 在“DDM实例管理”页面，重启DDM实例。

----结束

# A 修订记录

发布日期	修订记录
2019-09-09	第十三次正式发布，本次修改包括： 优化FAQ： <a href="#">本地环境是否可以连接DDM实例</a> 。 新增FAQ： <a href="#">导入数据到DDM后出现主键重复</a> 。
2019-05-31	第十二次正式发布，本次修改包括： 新增FAQ： <a href="#">DDM是否支持外键</a> 。 新增FAQ： <a href="#">如何处理表中存在主键重复的数据</a> 。
2019-01-30	第十一次正式发布，本次修改包括： 新增FAQ： <a href="#">创建逻辑表时提示RDS分片上有残留的同名表怎么办</a> 。
2018-12-28	第十次正式发布，本次修改包括： ● 新增FAQ： <a href="#">DDM实例的计费模式是否支持按需和包周期互转</a> 。 ● 新增FAQ： <a href="#">导入RDS失败提示DDM.0095是什么原因</a> 。
2018-09-17	第九次正式发布，本次修改包括： 新增FAQ： <a href="#">DDM支持创建逻辑表的方式有哪些</a> 。
2018-08-13	第八次正式发布，本次修改包括： ● 新增FAQ： <a href="#">连接DDM时密码多次错误被锁定该如何解决</a> 。 ● 新增FAQ： <a href="#">DDM支持哪些SQL</a> 。 ● 新增FAQ： <a href="#">DDM是否支持数据类型强制转换</a> 。
2018-07-22	第七次正式发布，本次修改包括： 新增FAQ： <a href="#">使用DDM事务模型有哪些注意事项</a> 。
2018-07-06	第六次正式发布，本次修改包括： ● 新增FAQ： <a href="#">如何解决JDBC驱动方式连接DDM异常问题</a> 。 ● 新增FAQ： <a href="#">如何选择JDBC驱动方式的版本和参数</a> 。

发布日期	修订记录
2018-06-05	第五次正式发布，本次修改包括： <ul style="list-style-type: none"> <li>● 新增FAQ: <a href="#">MySQL连接DDM时出现乱码如何解决</a>。</li> <li>● 新增FAQ: <a href="#">DDM实例是否支持跨VPC访问</a>。</li> <li>● 新增FAQ: <a href="#">数据库表名是否区分大小写</a>。</li> <li>● 新增FAQ: <a href="#">DDM控制台显示的RDS实例名称/ID和RDS控制台显示的RDS实例名称/ID为什么不同</a>。</li> <li>● 新增FAQ: <a href="#">RDS哪些高危操作会影响DDM</a>。</li> </ul>
2018-05-12	第四次正式发布，本次修改包括： 新增FAQ: <a href="#">DDM监控能力有哪些</a> 。
2018-02-28	第三次正式发布。本次修改包括： 新增FAQ: <a href="#">DDM是否支持SQL跨库访问</a>
2017-11-02	第二次正式发布。本次修改包括： <ul style="list-style-type: none"> <li>● 新增FAQ: <a href="#">DDM自身是否会存储业务数据</a>。</li> <li>● 新增FAQ: <a href="#">本地环境是否可以连接DDM实例</a>。</li> <li>● 新增FAQ: <a href="#">DDM提供哪些高可靠保障</a>。</li> <li>● 新增FAQ: <a href="#">如何选择和配置安全组</a>。</li> <li>● 新增FAQ: <a href="#">DDM如何进行分片</a>。</li> <li>● 新增FAQ: <a href="#">DDM逻辑表支持哪几种类型</a>。</li> <li>● 新增FAQ: <a href="#">如何实现读写分离</a>。</li> <li>● 新增FAQ: <a href="#">如何处理所关联的RDS实例异常</a>。</li> <li>● 新增FAQ: <a href="#">DDM对RDS采用哪种检测机制</a>。</li> </ul>
2017-07-29	第一次正式发布。