

分布式数据库中间件

# 最佳实践

文档版本 01  
发布日期 2025-08-01



版权所有 © 华为技术有限公司 2025。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暗示的声明或保证。

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。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

---

# 目录

1 概览.....	1
2 合理制定分片策略.....	2
3 如何选择 DDM 逻辑库分片数.....	6
4 广播表和单表的使用场景.....	8
5 DDM 事务模型.....	10
6 DDM SQL 使用规范.....	12
7 如何将单 RDS 数据整库迁移至 DDM.....	16
8 如何将 Mycat 数据整库迁移至 DDM.....	19
9 通过 JDBC 连接池连接 DDM.....	23
10 通过 Navicat 客户端连接 DDM 实例.....	26
11 通过数据复制服务 (DRS) 将 RDS for MySQL 数据迁移到 DDM.....	27
11.1 方案概述.....	27
11.2 资源和成本规划.....	28
11.3 操作流程.....	31
11.4 源端 RDS for MySQL 准备.....	32
11.4.1 创建源端 VPC 和安全组.....	32
11.4.2 创建 RDS for MySQL 实例.....	33
11.4.3 创建源端 VPN.....	36
11.5 目标端 DDM 准备.....	37
11.5.1 创建目标端 VPC 和安全组.....	37
11.5.2 创建 DDM 实例.....	38
11.5.3 创建 RDS for MySQL 实例.....	39
11.5.4 创建逻辑库并关联 RDS for MySQL 实例.....	41
11.5.5 创建 DDM 账号.....	41
11.5.6 创建目标库表结构.....	42
11.5.7 创建目标端 VPN.....	43
11.6 创建 VPN 连接.....	44
11.7 创建 DRS 迁移任务.....	44
11.8 确认迁移执行结果.....	45

<b>12 通过 DDM 对 RDS for MySQL 实例进行分库分表.....</b>	<b>48</b>
12.1 方案概述.....	48
12.2 RDS for MySQL 准备.....	49
12.3 DDM 准备.....	49
12.3.1 创建 DDM 实例.....	50
12.3.2 创建逻辑库并关联 RDS for MySQL 实例.....	50
12.3.3 创建 DDM 账号.....	51
12.3.4 创建目标库表结构.....	52
12.4 创建 DRS 迁移任务.....	54
12.4.1 创建 DRS 迁移任务.....	54
12.4.2 确认迁移执行结果.....	56
<b>13 如何选择拆分键和拆分算法.....</b>	<b>58</b>
<b>14 跨分片复杂 SQL 调优.....</b>	<b>61</b>

# 1 概览

本手册基于华为云分布式数据库中间件实践所编写，用于指导您完成相关设置，购买更符合业务的分布式数据库中间件实例。

表 1-1 DDM 最佳实践一览表

章节名称	简介
<a href="#">合理制定分片策略</a>	介绍创建拆分表时如何选择拆分键和拆分算法。
<a href="#">如何选择DDM逻辑库分片数</a>	介绍创建拆分库时如何合理选择逻辑库分片。
<a href="#">广播表和单表的使用场景</a>	介绍广播表和单表的常用场景。
<a href="#">DDM事务模型</a>	介绍DDM的事务模型。
<a href="#">DDM SQL使用规范</a>	介绍DDM中的SQL使用规范。
<a href="#">如何将单RDS数据整库迁移至DDM</a>	介绍将单RDS库中的数据整库迁移到DDM实例单库中。
<a href="#">如何将Mycat数据整库迁移至DDM</a>	介绍将Mycat中的数据整库迁移到DDM中。
<a href="#">通过JDBC连接池连接DDM</a>	介绍如何通过JDBC连接池与DDM对接，实现数据操作。
<a href="#">通过Navicat客户端连接DDM实例</a>	介绍如何获取弹性公网的IP地址，并通过Navicat客户端连接DDM实例。
<a href="#">通过数据复制服务（DRS）将RDS for MySQL数据迁移到DDM</a>	介绍如何通过DRS服务将RDS for MySQL数据迁移到DDM实例单库中。
<a href="#">通过DDM对RDS for MySQL实例进行分库分表</a>	介绍如何通过DDM对已有RDS for MySQL实例进行分库分表。
<a href="#">跨分片复杂SQL调优</a>	介绍如何针对跨分片复杂SQL时进行调优操作。

# 2 合理制定分片策略

DDM创建逻辑表时，若逻辑表类型选择为拆分表，则需要制定分片策略，并选择拆分键。

当数据表之间存在E-R关系时，可以制定相同的分片规则，各数据表分别选择有关联关系的字段作为拆分键，这样各表中有关联关系的数据将会存储在一个分片上，避免数据跨分片JOIN操作。如客户表、订单表与订单明细表，在创建拆分表时，建议都选取客户ID作为拆分键。

表 2-1 拆分键与拆分算法使用简介

拆分算法	hash类		range类	
拆分键	表字段	表字段+日期函数	表字段	表字段+日期函数
详细说明	根据指定的表字段将数据平均拆分到各个分片上。	根据指定的表字段+日期函数将数据平均拆分到各个分片上。 表字段必须是日期类型（date、datetime、timestamp）。	将数据表内的记录按照算法元数据定义的规则将数据拆分到指定的分片上。	根据指定的表字段+日期函数将数据按照算法元数据的规则将数据拆分到各个分片上。 表字段必须是日期类型（date、datetime、timestamp）。

<p><b>适用场景</b></p>	<p>适用于需要将数据均匀分布的场景，例如：银行类客户业务应用，业务逻辑主体是客户，可使用客户对应的表字段（例如客户号）作为拆分键，详情参见如下示例。</p>	<p>需要按时间（年、月、日、周及其组合）对数据进行拆分的场景，例如：游戏类的应用，可使用玩家对应的表字段（例如玩家注册时间）作为拆分键，按日、月、年等函数分片，方便统计和查询某日、月玩家的操作数据，帮助游戏厂家做大数据分析。</p>	<p>适合范围类操作较多的场景，例如：电商类应用，如果业务场景是围绕商家做活动进行，业务逻辑主体是活动日期，可使用活动日期对应的表字段（例如活动名称、日期范围）作为拆分键，方便统计某周期内销量等情况。</p>	<p>例如日志分析场景，日志系统中可能包含各类复杂的信息，这时您可以选择时间字段作为拆分键，然后对拆分键使用日期函数拆分。 为了方便日志清理和转储，采用range拆分算法，对时间字段用日期函数转换成年，表示按年存储到各个分片上，详情参见如下示例。</p>
--------------------	---	---	--	---

## 如何选择拆分算法

拆分算法即将逻辑表中数据拆分到多个数据库分片上的算法，DDM支持hash和range两大类拆分算法。

- hash类  
将数据均匀分布在各个分片。  
适用于SQL查询条件使用“=”、“IN”之类运算符相对较多的场景。
- range类  
将数据表内的记录按照算法元数据的取值范围进行分片存储。  
适用于SQL查询条件使用“>”、“<”、“BETWEEN ... AND ...”之类运算符相对较多的场景。

### 注意

如果分库算法是range类+日期函数，且拆分键字段又表示创建时间，在数据入库时可能造成热点问题，无法充分利用多MySQL的优势。

拆分算法的使用，需要结合业务查询场景进行评估，以选择合适的拆分算法，提升DDM效率。

## 如何选择拆分键

拆分键是在水平拆分逻辑表的过程中，用于生成路由结果的表字段，指定表字段后，可以进一步选择日期函数，也可以手动输入“日期函数(字段名)”，数据表字段必须是日期类型（date、datetime、timestamp），日期函数适用于需要按时间（年、月、日、周及其组合）对数据进行拆分的场景。

DDM根据拆分键与拆分算法计算路由结果，对拆分表的数据进行自动水平拆分，分发到数据分片中。

选取拆分算法与拆分键一般遵循以下规则：

- 尽可能使数据均匀分布到各个分片上。
- 该拆分键是最频繁或者最重要的查询条件。
- 优选主键作为拆分键，因为主键作为查询条件时，查询速度最快。

## 有明确主体的业务场景

拆分表的数据量一般都达到千万级，因此选择合适的拆分算法和拆分键非常重要。如果能找到业务主体，并且确定绝大部分的数据库操作都是围绕这个主体的数据进行的，那么可以选择这个主体所对应的表字段作为拆分键，进行水平拆分。

业务逻辑主体与实际应用场景相关，下列场景都有明确的业务逻辑主体。

1. 银行类客户业务应用，业务逻辑主体是客户，可使用客户对应的表字段（例如客户号）作为拆分键。部分业务系统的业务场景为围绕银行卡/账号的，可以选取卡/账号作为拆分键。
2. 电商类应用，如果业务场景是围绕商品进行操作，业务逻辑主体是商品，可使用商品对应的表字段（例如商品编码）作为拆分键。
3. 游戏类的应用，主要围绕玩家数据进行操作，业务逻辑主体是玩家，可使用玩家对应的表字段（例如玩家id）作为拆分键。

以银行类客户业务为例，建表SQL语句如下：

```
CREATE TABLE PERSONALACCOUNT(  
  ACCOUNT VARCHAR(20) NOT NULL PRIMARY KEY,  
  NAME VARCHAR(60) NOT NULL,  
  TYPE VARCHAR(10) NOT NULL,  
  AVAILABLEBALANCE DECIMAL(18, 2) NOT NULL,  
  STATUS CHAR(1) NOT NULL,  
  CARDNO VARCHAR(24) NOT NULL,  
  CUSTOMID VARCHAR(15) NOT NULL  
) ENGINE = INNODB DEFAULT CHARSET = UTF8  
dbpartition by hash(ACCOUNT);
```

## 无明确主体的业务场景

如果业务场景中找不到合适的主体，也可以选择那些数据分布较为均匀的属性所对应的表字段作为拆分键。

例如日志分析场景，日志系统中可能包含各类复杂的信息，这时您可以选择时间字段作为拆分键。

选择时间字段作为拆分键时，支持对拆分键使用日期函数拆分。

为了方便清理和转储，采用range拆分算法，对时间字段用日期函数转换成月，表示按月存储到各个分片上。

建表SQL语句：

```
CREATE TABLE LOG(  
  LOGTIME DATETIME NOT NULL,  
  LOGSOURCESYSTEM VARCHAR(100),  
  LOGDETAIL VARCHAR(10000)  
)  
dbpartition by range(month(LOGTIME)) {  
  1 - 2 = 0,
```

```
3 - 4 = 1,  
5 - 6 = 2,  
7 - 8 = 3,  
9 - 10 = 4,  
11 - 12 = 5,  
default = 0  
};
```

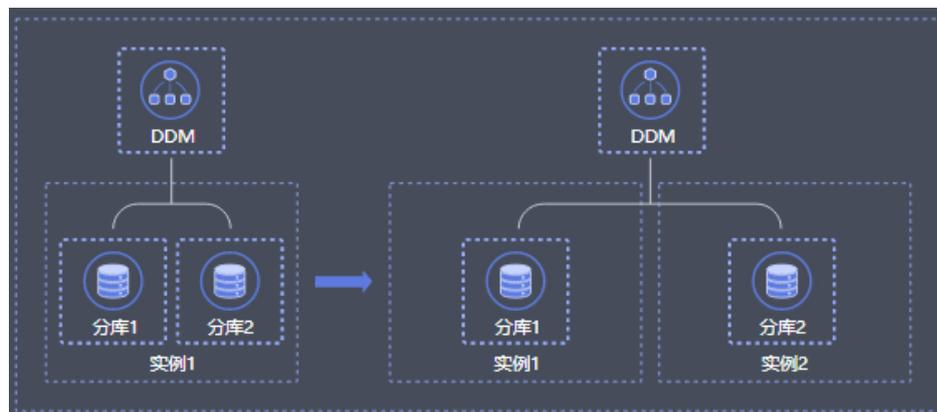
# 3 如何选择 DDM 逻辑库分片数

在DDM实例创建逻辑库时，对需要水平拆分的表，选择合适的MySQL实例数量，通过预估拆分表的数据量，规划出逻辑库分片数量，可以避免各个分片的上单张表数据容量过高。一个分片即为MySQL实例的一个数据库，逻辑库的总分片数等于所关联的MySQL分片数总和。

- 逻辑库选择“非拆分”模式时，一个逻辑库仅对应一个MySQL实例，在该实例上仅创建1个分片。
- 逻辑库选择“拆分”模式时，一个逻辑库可对应多个MySQL实例，在每个实例上，可创建1-64个分片。如果因业务需要分片数超过64，请联系DDM技术人员。

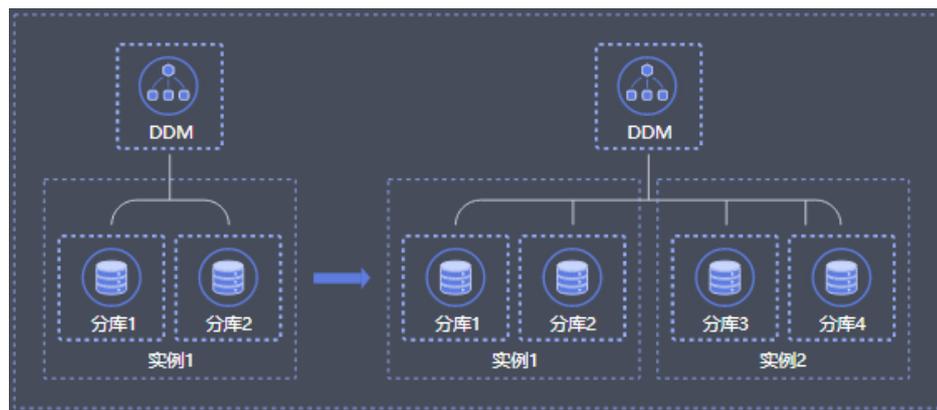
分片平移到新实例：将逻辑库部分分片（分库）平移到新增的实例上，数据不会进行重分布。

图 3-1 分片平移示意图



分片个数变更：逻辑库对应的物理库分片可以根据用户需要自定义添加，单数据节点上的物理分片数不超过64个。如果因业务需要分片数超过64，请联系DDM技术人员。DDM会尽量将分片均匀分配到数据节点上，只要总分片数有变化，数据都会进行重分布。

图 3-2 分片变更示意图



# 4 广播表和单表的使用场景

单表指数据只存储在其中一个默认分片上的表。广播表指在所有分片上都存储全量数据，提升JOIN效率。

## 单表

DDM管理控制台不提供单表创建操作，用户可以通过MySQL客户端或应用程序连接到DDM实例后自行创建。

如果一张表的数据量预估在1000万条以下，且没有与其他拆分表进行关联查询的需求，建议将其设置为单表类型，存储在默认分片中。

**单表创建样例，不做任何拆分：**

```
CREATE TABLE single(  
id int NOT NULL AUTO_INCREMENT COMMENT '主键id',  
name varchar(128),  
PRIMARY KEY(id)  
);
```

## 广播表

在业务数据库中，存在一些数据量不大，更新频率较低，但常常需要用来做关联查询的表。

为了方便这类表与拆分表进行关联操作，DDM实例提供了一种“广播表”，广播表具有以下特点：

- 广播表在各分片中数据一致。插入、更新和删除数据会实时在每一个分片中执行。
- 对广播表的查询，仅在一个分片中执行。
- 任何表都可以与广播表进行关联操作。
- 使用广播HINT的前提是需要确保表的存在性。

例如：

电商企业的订单管理系统，需要查询并统计某个地区的订单数据。这种场景会涉及到地区表与订单流水表进行关联查询，由于订单数据量庞大，订单流水表需要分片存储，因此可以考虑将地区表设计为“广播表”，提高查询效率。

**广播表创建样例：**

```
CREATE TABLE broadcast_tbl (  
  id int NOT NULL AUTO_INCREMENT COMMENT '主键id',  
  name varchar(128),  
  PRIMARY KEY(id)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci  
BROADCAST;
```

**对广播表进行操作时应注意以下两点：**

- 如果执行全表扫描的SQL语句，建议执行频率不要过高，或者选择在业务低峰期执行，否则可能会出现“后端RDS连接数可能不够用”报错。
- 广播表的插入操作，会在每个分片的表中进行插入。如果此表数据量太大（超过100万条数据），不建议使用广播表，以免占用太多数据空间，可以使用拆分表。

# 5 DDM 事务模型

因为DDM中的表一般都做了分库分表，使得这些表中的数据可能分布在很多个RDS实例上的不同数据库分片内。在DDM中一个事务中对各个逻辑表的增删改查很有可能实际发生在多个RDS实例上的不同数据库分片上，每个RDS实例上同一数据库分片内所发生的对分片内数据表的一系列操作相当于一个本地事务，这样DDM层面的事务就相当于由多个RDS实例上的本地事务所组成的分布式事务，这些本地事务要么全部成功，要么全部失败。

## DDM 的分布式事务实现

DDM中的分布式事务模块基于MySQL XA协议实现，XA协议是对2PC（Two Phase Commit）事务模型的一种实现，2PC是一种经典的分布式事务实现方案。基于MySQL XA协议的分布式事物可以实现强一致性写，整体分布式事物的一致性说明可参考MySQL官方文档。

在分布式系统里，每个参与者都可以知晓自己操作的成功或者失败，却无法知道其他节点操作的成功或失败。当一个事务跨多个节点时，为了保持事务的原子性与一致性，需要引入一个协调者（Coordinator）来统一掌控所有参与者（Participant）的操作结果，并指示它们是否要把操作结果进行真正的提交（commit）或者回滚（rollback）。DDM节点在这个分布式事务中扮演协调者的角色，RDS实例则是参与者。

2PC顾名思义分为两个阶段，其实施思路可概括为：

1. 准备阶段（prepare phase）：参与者将操作结果通知协调者，参与者如果给出了已准备的回应，则应保证在协调者做出决定之前需要将本次决议所需资源进行预留。
2. 提交阶段（commit phase）：收到参与者的回应后，协调者再向参与者发出通知，根据反馈情况决定各参与者是否要提交还是回滚。

举例说明：

甲乙丙丁四人要聚餐，需要确定时间，不妨设甲是协调者，乙丙丁是参与者。

准备阶段：

1. 甲发短信给乙丙丁，周二中午十二点聚餐是否有时间。
2. 丁回复有时间。
3. 乙回复有时间。

4. 丙迟迟不回复，此时对于这个活动，甲乙丙丁均处于阻塞状态，算法无法继续进行。
5. 丙回复有时间（或者没有时间）。

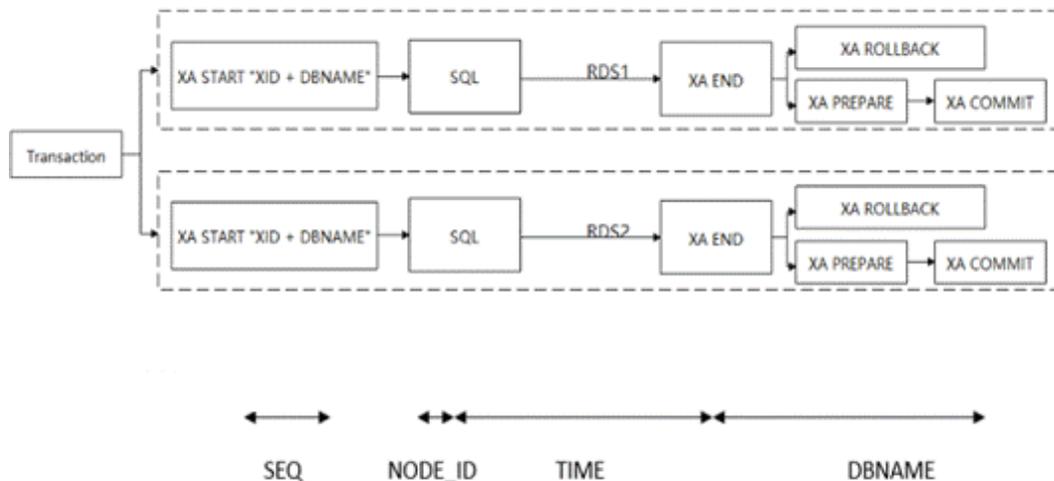
提交阶段：

1. 协调者甲将收集到的结果（周二中午聚餐或者不聚餐）反馈给乙丙丁（什么时候反馈，以及反馈结果如何，在此例中取决于丙的时间与决定）。
2. 乙收到。
3. 丙收到。
4. 丁收到。
5. 如果有人没有收到，则甲后续不断重新进行通知，直到通知成功。

## 2PC 具体实现

DDM对事务的2阶段提交进行了透明化封装。应用侧使用DDM事务的时候不需要关心底层是否为分布式事务，只需像使用普通事务一样执行BEGIN/COMMIT指令进行事务操作。DDM会自动处理分布式事务的两阶段提交逻辑。如果事务只涉及一个数据分片，那么DDM还可以自动将事务降级为一阶段提交的本地事务。一阶段提交的细节本文不详细展开。

图 5-1 分布式事务执行流程



其中XID由节点内事务序号、节点ID和时间戳组成，代表了一个全局维度的分布式事务唯一编号，该编号再加上对应分库名构成实际在物理库上开启分布式事务分支的XA\_ID。

## DDM 事务使用建议

- 虽然DDM支持分布式事务，但如果能够使事务内所有SQL都带上同一个拆分键，DDM可以把事务优化为单分片事务从而获得更优的性能。
- DDM事务模型采取2PC分布式事务模型，RDS自身的死锁检测无法避免跨RDS的分布式事务死锁，如遇到lock wait超时则提示可能存在跨分片死锁，请检查业务模型是否存在这方面问题。

# 6 DDM SQL 使用规范

## insert 操作

- **insert写法规范**

- 不建议逐条insert，推荐使用insert into values (),(.)..();语法。
- MySQL的JDBC连接的url中要加rewriteBatchedStatements参数，并保证5.1.13以上版本的驱动，才能实现批量插入。MySQL JDBC驱动在默认情况下会无视executeBatch()语句，把预计批量执行的一组sql语句拆散，一条一条地发给MySQL数据库，批量插入实际上是单条插入，直接造成较低的性能。只有把rewriteBatchedStatements参数置为true，驱动才会批量执行SQL。另外这个选项对INSERT/UPDATE/DELETE操作都有效。

---

**⚠ 注意**

开启rewriteBatchedStatements后的批量INSERT/UPDATE/DELETE操作要注意设置合理的batch size，batch size过大可能造成性能下降。如无特殊需求，建议batch size不超过1000。

- 拆分字段的值：不建议使用函数、表达式、子查询等，推荐使用常量值。
- 普通字段的值：不建议使用子查询，推荐使用常量、函数、表达式。

- **大批量数据导入**

推荐使用loaddata local infile来实现大批量数据导入。

---

**⚠ 注意**

这个地方只需要开一个session窗口导入即可，DDM内部会自动进行并行导入。

- **数据迁移场景**

建议使用mysqldump导出sql文件，再使用mysql source命令导入。

- **auto\_increment字段**

- DDM使用Sequence来实现auto\_increment语义，并保证全局唯一。
- 赋值：若使用auto\_increment字段，建议不要在values子句中赋值，否则容易造成主键冲突。如果在values中赋了值，建议使用alter sequence语句更改。

- 步长：auto\_increment字段步长建议不要设置成1，会导致性能低下。默认设置成1000。

## update 与 delete 操作

- **普通更新**
  - 进行update/delete操作时，where条件建议带上拆分字段；
  - 无法带上拆分字段的场景，建议控制并发度，控制更新/删除涉及的数据条数。建议先用select查出相应的数据，double check确保数据范围无误后再实行update/delete操作。
- **拆分字段更新**
  - 条数限制：DDM拆分字段更新有数据量限制，一般不能超过10000条数据，数据量越少越好。若超过10000条数据，建议用改重建表的方式来做，或者是拆分成多次update操作来等价实现。
  - 操作的时机：建议选择在业务低谷期做。
- **关联操作**

不建议进行表关联更新、表关联删除操作，即不建议进行多张表同时进行update/delete操作。
- **子查询及limit操作**

不建议update/delete语句中含有子查询。不建议update/delete语句中含有limit或order by limit语句。
- **case when函数**

case when分支数量在不同规格实例下不同，在4U8G中不超过2500个，8U16G中不超过5000个，32U64G中20000个。

## select 操作

- **Order by 及 Limit函数**
  - "order by limit offset, count"场景，禁止给offset赋大数值，即禁止深度翻页。
  - 如遇到临时表超限（Temp table limit exceeded）报错，说明排序中间数据产生了临时表且超限，考虑联系DDM值班进行SQL调优。
- **Group by函数**
  - 不建议select\_list部分含有非group by列。
  - 不支持不可下推的group\_concat聚合函数内含有order by子句。
  - 不建议distinct、group by字段多于3项。
  - 不建议join、或者子查询操作之后含有group by操作。
  - 不建议使用count(distinct),sum(distinct)操作。
  - 如遇到临时表超限（Temp table limit exceeded）报错，说明聚合操作中间数据产生了临时表且超限，考虑联系DDM值班进行SQL调优。
  - 带GROUP BY分组查询的SQL语句，如果所有分组或单分组查询结果超过100万条（默认值），此时会有临时表超限约束（Temp table limit exceeded）。
- **Join函数**
  - select场景，建议join条件是每个表的拆分字段或使用广播表，或者是驱动表是一个小表（inner/left join驱动表是左表，right join驱动表是右表）。

- 不建议两个大表直接进行join操作。
- 不建议join on condition中含有非等值操作。
- 如遇到临时表超限（Temp table limit exceeded）报错，说明JOIN中间数据产生了临时表且超限，考虑联系DDM值班进行SQL调优。
- 不建议5张表以上进行join操作。
- join查询操作建议不要开启事务。
- 不建议在事务中进行join查询，开启事务会影响DDM对join算法的选择，无法使用最高效的算法。

**注意**

这里的大表和小表的意思不是指原始表的规模，指的是经过where条件过滤之后的数据规模。

- **子查询**
  - 不建议子查询包含在OR表达式中，或者是子查询的关联条件包含在OR表达式中
  - 不建议使用含有limit的标量子查询，如 select (select x from t2 where t2.id=t.id limit 1),a,b from t。
  - 如果子查询和主表都路由到同一分片，建议在SQL前加/\*+db=xxx\*/来精准路由。
  - 不建议子查询内部含有join语句。
  - 不支持嵌套子查询。
  - 不建议ROW表达式跟子查询做比较操作，如 select \* from t where (a,b,c)=(select x,y,z from t2 where …)。
  - 不建议select\_list里面含有超过2个以上的子查询。

## DDL

- **DDL执行时机**  
对已有表进行DDL操作时建议放在业务低峰期进行。
- **分片数**  
创建新拆分表时建议结合实际数据量进行合理预估，总分片数满足需求即可。不建议使用超出实际需求的分片数，拆分表分片数并非越多越好
- **高危DDL**  
进行高危DDL时请仔细校验SQL后谨慎操作，如DROP TABLE, TRUNCATE TABLE等操作。
- **DDL失败修复**  
DDL命令如遇报错，可以使用"check table 表名"命令对各个分片表结构进行校验，识别出失败的分片进行针对性修复。如ALTER TABLE命令遭遇失败可以在命令前添加/\*+allow\_alter\_rerun=true/，开启ALTER语句的幂等可重入执行后重试，直到check table 命令提示各个分片表结构达到一致则可认为执行成功。
- **MDL锁导致执行DDL报错**
  - **背景：**为保证DDL的可用性，DDM内部在执行DDL前会检查底层RDS相关表是否存在MDL锁。若存在MDL锁，则DDL会提前报错退出。

metadata lock exists, one of MDL is [%s],DDL operation can not proceed, please use 'show metadata lock' to check current mdl, and use 'kill physical threadId@host:port' to clean it

- **可能出现的问题**: 若系统中存在慢SQL, 执行时间为几分钟不等, 那么可能被MDL锁所阻拦, 无法执行DDL。
- **解决方案1**: 在DDM控制台提高参数 “ddl\_precheck\_mdl\_threshold\_time” 的大小, 如提高到30分钟(1800秒)。

#### 📖 说明

“ddl\_precheck\_mdl\_threshold\_time” 表示DDL允许MDL锁持有的最大时长。持有超过这个时间长度的MDL锁, DDL才会报错, 默认值为120秒。

- **解决方案2**: 执行show metadata lock查看是否因为持有慢事务的MDL锁阻塞了DDL的执行。若存在阻塞, 可以使用kill physical threadId@host:port 来关闭底层慢事务。配合/\*+allow\_alter\_rerun=true\*/的hint, 以及check table 来查看和执行, 直到DDL彻底执行完。

#### 📖 说明

threadId为物理层RDS的线程id, host和port分别为物理层RDS的ip和端口。

- **DDL长时间卡死**

在业务低峰期执行DDL时如果遇到长时间卡死情况, 请另开会话执行xa recover命令查看是否有慢事务存在, 如存在慢事务挂起请及时联系值班人员解决。

# 7 如何将单 RDS 数据整库迁移至 DDM

## 操作场景

本章节主要介绍将单RDS（非拆分，以下简称旧RDS）库中的数据整库迁移到DDM实例非拆分库中，仅使用DDM做读写分离场景。

### 📖 说明

- 迁移过程中可能会出现业务中断情况，中断时长与迁移数据量大小、网络情况相关。
- 数据迁移是一项比较复杂的操作，建议在业务量较低时进行。本实践仅供参考，您需要根据自己业务场景、数据量、停机时间要求等情况，设计合适的迁移方案。
- 对于数据量较大的场景，建议通过工单或售后服务联系DDM技术支持人员进行支撑，在正式数据迁移前进行充分的迁移演练测试。

## 迁移前准备

- 准备可以访问旧RDS实例、目标DDM实例和目标DDM实例关联的RDS实例的ECS。
  - a. 确保旧RDS实例、目标DDM实例和目标DDM实例关联的RDS实例都在同一个VPC下，保证网络互通。
  - b. 旧RDS实例、目标DDM实例和目标DDM实例关联的RDS实例的安全组建议配置相同，如果不同则需要放开对应端口访问。
  - c. ECS已安装MySQL官方客户端，MySQL客户端版本建议为5.6或5.7。
    - Red Hat系列Linux安装命令：**yum install mysql mysql-devel**
    - Debian系列Linux安装命令：**apt install mysql-client-5.7 mysql-client-core-5.7**
  - d. ECS磁盘空间足够存放临时转储文件；ECS内存空间足够，可以用来比较转储文件。
- 准备已关联RDS实例的DDM实例，并配置DDM账号、DDM逻辑库等相关信息。
- 如果目标DDM实例逻辑库为拆分库，则需要在迁移前，在DDM控制台先创建与旧RDS数据表结构相同的逻辑表。

## 约束限制

- 为了保持数据完整性，需要先停止旧RDS业务后再进行数据迁移。

- 该场景不支持通过DDM关联旧RDS实例进行数据关联，需要将旧RDS实例数据导出后再导入到DDM实例完成数据迁移。
- 目标DDM关联的RDS版本与旧RDS的MySQL版本需要保持一致。

## 从旧 RDS 导出数据

**步骤1** 登录ECS。

**步骤2** 执行如下命令导出结构数据，其中斜体参数需根据实际情况配置，详细参数说明如表 7-1所示。

```
mysqldump -h {DB_ADDRESS} -P {DB_PORT} -u {DB_USER} -p --skip-lock-tables --add-locks=false --set-gtid-purged=OFF --no-data {DB_NAME} > {mysql_schema.sql}
```

表 7-1 参数解释

参数	说明	备注
DB_ADDRESS	待导出数据的数据库连接地址。	必填
DB_PORT	数据库侦听端口	必填
DB_USER	数据库用户	必填
DB_NAME	数据库名称	必填
mysql_schema.sql	生成的表结构文件名。	每次导出表结构时文件名不同。建议以“逻辑库名”+“_”+“schema”格式命名，以免数据被覆盖。如mysql_schema.sql。
mysql_data.sql	生成的整库数据文件名。	-

**步骤3** 执行如下命令导出整库数据，其中斜体参数需根据实际情况配置，详细参数说明如表 7-1所示。

```
mysqldump -h {DB_ADDRESS} -P {DB_PORT} -u {DB_USER} -p --hex-blob --complete-insert --skip-lock-tables --skip-tz-utc --skip-add-locks --set-gtid-purged=OFF --no-create-info {DB_NAME} > {mysql_data.sql}
```

**步骤4** 导出完成后，在ECS上执行如下可查看**步骤2**和**步骤3**导出的.sql文件。

```
ls -l
```

----结束

## 将数据导入至 DDM 实例

**步骤1** 在ECS上执行如下命令，将结构文件导入DDM。

```
mysql -f -h {DDM_ADDRESS} -P {DDM_PORT} -u {DDM_USER} -p {DB_NAME} < {mysql_schema.sql}
Enter password: *****
```

表 7-2 参数解释

参数	说明	备注
DDM_ADDRESS	待导入数据的DDM实例连接地址。	可在DDM管理控制台上，实例基本信息页面查看“内网地址”和“端口”。
DDM_PORT	待导入数据的DDM侦听端口。	
DDM_USER	访问DDM的用户。	创建DDM逻辑库时使用的账号，需具备读写权限。
DB_NAME	待导入数据的DDM逻辑库名称。	-
mysql_schema.sql	待导入结构文件的名称。	即导出数据中 <a href="#">步骤2</a> 导出的文件名称。
mysql_data.sql	待导入整库数据文件的名称。	即导出数据中 <a href="#">步骤3</a> 导出的文件名称。

**步骤2** 执行如下命令，将数据文件导入DDM。

```
mysql -f -h {DDM_ADDRESS} -P {DDM_PORT} -u {DDM_USER} -p {DB_NAME} <
{mysql_data.sql}
Enter password: *****
```

----结束

# 8 如何将 Mycat 数据整库迁移至 DDM

## 操作场景

本章节主要介绍将Mycat中的数据整库迁移到DDM中。

### 说明

- 迁移过程中可能会出现业务中断情况，中断时长与迁移数据量大小、网络情况相关。
- 数据迁移是一项比较复杂的操作，建议在业务量较低时进行。本实践仅供参考，您需要根据自己业务场景、数据量、停机时间要求等情况，设计合适的迁移方案。
- 由于DDM仅支持通过弹性云服务器（ECS）访问，因此需要先将数据库导出为文件到ECS，然后从ECS将文件中的数据导入到DDM。

## 迁移前准备

- 准备可以访问Mycat、目标DDM实例和目标DDM实例关联的RDS实例的ECS。
  - a. 确保Mycat、目标DDM实例和目标DDM实例关联的RDS实例都在同一个VPC下，保证网络互通。
  - b. 部署Mycat的ECS、目标DDM实例和目标DDM实例关联的RDS实例的安全组建议配置相同，如果不同则需要放开对应端口访问。
  - c. ECS已安装MySQL官方客户端，MySQL客户端版本建议为5.6或5.7。
    - Red Hat系列Linux安装命令：**yum install mysql mysql-devel**
    - Debian系列Linux安装命令：**apt install mysql-client-5.7 mysql-client-core-5.7**
  - d. ECS磁盘空间足够存放临时转储文件；ECS内存空间足够，可以用来比较转储文件。
- 准备已关联RDS实例的DDM实例，并配置DDM账号、DDM逻辑库等相关信息。
- 本章节以Mycat 1.6版本进行迁移为例。

## 迁移策略

Mycat与DDM数据表类型不同，迁移策略也有所差异，详情如[表8-1](#)所示。

表 8-1 迁移策略

Mycat表类型	DDM表类型	迁移策略
非拆分表	单表	<ol style="list-style-type: none"> <li>1. Mycat导出<b>表结构</b>和<b>表数据</b>。</li> <li>2. 连接目标DDM关联的RDS<b>将数据导入至目标DDM（非拆分表场景）</b>。</li> </ol>
拆分表：分片规则为hash类（含年月日等日期类）	拆分表：拆分算法为hash（含日期函数）	<ol style="list-style-type: none"> <li>1. Mycat导出全部数据表结构。</li> <li>2. 对照导出的表结构，在DDM控制台创建表结构完全相同的表。</li> </ol>
拆分表：分片规则为按范围range类（含年月日等日期类）	拆分表：拆分算法为range（含日期函数）	<ol style="list-style-type: none"> <li>3. <b>从Mycat整库导出数据</b>。</li> </ol>
广播表	广播表	<ol style="list-style-type: none"> <li>4. <b>连接DDM导入整库数据</b>。</li> </ol>

## 约束限制

- 为了保持数据完整性，需要先停止Mycat业务后再进行数据迁移。
- 该场景不支持通过DDM关联Mycat关联的RDS进行数据关联，需要将Mycat数据导出后再导入到DDM完成数据迁移。
- 目标DDM关联的RDS版本与Mycat关联数据库的版本需要保持一致。

## 从 Mycat 导出数据表结构

步骤1 登录ECS。

步骤2 执行如下命令导出Mycat数据中的表结构数据，其中斜体变量参数需根据实际情况配置，详细参数说明如表8-2所示。

```
mysqldump -h {DB_ADDRESS} -P {DB_PORT} -u {DB_USER} -p --skip-lock-tables --add-locks=false --set-gtid-purged=OFF --no-data --order-by-primary {DB_NAME} > {mysql_schema.sql}
Enter password: *****
```

表 8-2 参数解释

参数	说明	备注
DB_ADDRESS	待导出数据的数据库连接地址。	必填
DB_PORT	数据库侦听端口	必填
DB_USER	数据库用户	必填
DB_NAME	数据库名称	必填

参数	说明	备注
mysql_schema.sql	生成的表结构文件名。	每次导出表结构时文件名不同。建议以“逻辑库名”+“_”+“schema”格式命名，以免数据被覆盖。如mysql_schema.sql。
mysql_data.sql	生成的整库数据文件名。	-

----结束

## 从 Mycat 整库导出数据

**步骤1** 登录ECS。

**步骤2** 执行如下命令导出Mycat整库数据，其中斜体变量参数需根据实际情况配置，详细参数说明如表8-2所示。

```
mysqldump -h {DB_ADDRESS} -P {DB_PORT} -u {DB_USER} -p --hex-blob --complete-insert --skip-lock-tables --add-locks=false --set-gtid-purged=OFF --quick --no-create-info --order-by-primary {DB_NAME} > {mysql_data.sql}
Enter password: *****
```

**步骤3** 导出完成后，在ECS上执行如下命令可查看导出的.sql文件。

```
ls -l
```

----结束

## 将数据导入至目标 DDM（非拆分表场景）

在ECS上采用MySQL客户端直连目标DDM关联的RDS，直接执行以下命令导入表结构文本文件和数据文件。

如果是单表或普通表，

```
mysql -f -h {RDS_ADDRESS} -P {RDS_PORT} -u {RDS_USER} -p {DB_NAME} < {mysql_table_schema.sql}
Enter password: *****
mysql -f -h {RDS_ADDRESS} -P {RDS_PORT} -u {RDS_USER} -p {DB_NAME} < {mysql_table_data.sql}
Enter password: *****
```

- RDS\_ADDRESS为待导入数据的RDS的地址。
- RDS\_PORT为RDS实例的端口。
- RDS\_USER为RDS实例的用户名。
- DB\_NAME为RDS数据库名称，如果导入的是单表，DB\_NAME为RDS第一个分片的物理数据库。
- mysql\_table\_schema.sql为待导入的表结构文件名。
- mysql\_table\_data.sql为待导入的表数据文件名。

## 将数据导入至目标 DDM（拆分表、广播表场景）

在ECS上采用MySQL客户端连接DDM执行如下命令，将整库数据文件导入DDM。

```
mysql -f -h {DDM_ADDRESS} -P {DDM_PORT} -u {DDM_USER} -p {DB_NAME} < {mysql_data.sql}
Enter password: *****
```

表 8-3 参数解释

参数	说明	备注
DDM_ADDRESS	待导入数据的DDM实例连接地址。	可在DDM管理控制台上，实例基本信息页面查看“连接地址”和“端口”。
DDM_PORT	待导入数据的DDM侦听端口。	
DDM_USER	访问DDM的用户。	创建DDM逻辑库时使用的账号，需具备读写权限。
DB_NAME	待导入数据的DDM逻辑库名称。	-
mysql_data.sql	待导入整库数据文件的名称。	即导出数据中 <a href="#">步骤2</a> 导出的文件名称。

# 9 通过 JDBC 连接池连接 DDM

## 操作场景

**连接池实现原理：**在系统初始化的时候，将数据库连接作为对象存储在内存中，当用户需要访问数据库时，发出请求，直接从连接池中取出一个已建立的空闲连接对象。使用完毕后，再将连接放回连接池中，供下一个请求访问使用。连接的建立、断开都由连接池自身来管理。同时，还可以通过设置连接池的参数来控制连接池中的初始连接数、连接的上下限数以及每个连接的最大使用次数、最大空闲时间等等。也可以通过系统自身的管理机制来监视数据库连接的数量、使用情况等。

本章节主要介绍了如何通过JDBC连接池与DDM对接，实现数据操作。如果是Java程序，建议您使用[HikariCP](#)。

- Java 8：建议使用3.3.1版本。
- Java 7：建议使用2.4.13版本。
- JDBC连接DDM不支持开启用户游标提取（useCursorFetch）参数。

## 操作步骤

### 步骤1 配置Maven。

- Java 8：

```
<dependency>
  <groupId>com.zaxxer</groupId>
  <artifactId>HikariCP</artifactId>
  <version>3.3.1</version>
</dependency>
```
- Java 7：

```
<dependency>
  <groupId>com.zaxxer</groupId>
  <artifactId>HikariCP-java7</artifactId>
  <version>2.4.13</version>
</dependency>
```

### 步骤2 创建表。

表 9-1 创建表。

表名	字段	类型	是否主键
account	account_number	bigint	是
	account_type	varchar ( 45 )	否
	account_name	varchar ( 50 )	否

### 步骤3 连接DDM实例。

1. 配置连接数：JdbcUrl连接串中的参数和HikariCP参数。
2. 插入数据。

示例：

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.sql.DataSource;
import com.zaxxer.hikari.HikariConfig;
import com.zaxxer.hikari.HikariDataSource;

// 认证用的用户名和密码直接写到代码中有很大的安全风险，建议在配置文件或者环境变量中存放(密码应密文存放，使用时解密)，确保安全。
// 本示例以用户名和密码保存在环境变量中为例，运行本示例前请先在本地环境中设置环境变量(环境变量名称请根据自身情况进行设置)EXAMPLE_USERNAME_ENV和EXAMPLE_PASSWORD_ENV。

public class HikariCPDemo {
    private static DataSource datasource;
    private static DataSource getDataSource() {
        if (datasource == null) {
            HikariConfig config = new HikariConfig();

            // 配置JdbcUrl连接串中的参数
            config.setJdbcUrl("jdbc:mysql:loadbalance://192.168.0.10:5066,192.168.0.11:5066/db_name?loadBalanceAutoCommitStatementThreshold=5&loadBalanceHostRemovalGracePeriod=15000&loadBalanceBlacklistTimeout=60000&loadBalancePingTimeout=5000&retriesAllDown=10&connectTimeout=1000");

            /*
            // 配置JdbcUrl连接串中的参数也可以通过以下方法进行：
            config.addDataSourceProperty("loadBalanceAutoCommitStatementThreshold",5);
            config.addDataSourceProperty("loadBalanceHostRemovalGracePeriod", 15000);
            config.addDataSourceProperty("loadBalanceBlacklistTimeout", 60000);
            config.addDataSourceProperty("loadBalancePingTimeout", 5000);
            config.addDataSourceProperty("retriesAllDown", 10);
            config.addDataSourceProperty("connectTimeout", 10000);
            */
            config.setUsername("EXAMPLE_USERNAME_ENV");
            config.setPassword("EXAMPLE_PASSWORD_ENV");
            config.setMaximumPoolSize(10);
            config.setAutoCommit(true);

            // 配置HikariCP参数
            config.addDataSourceProperty("cachePrepStmts", true);
            config.addDataSourceProperty("prepStmtCacheSize", 250);
            config.addDataSourceProperty("prepStmtCacheSqlLimit", 2048);
            config.addDataSourceProperty("minimumIdle", 5);
            config.addDataSourceProperty("maximumPoolSize", 10);
            config.addDataSourceProperty("idleTimeout", 30000);

            datasource = new HikariDataSource(config);
        }
        return datasource;
    }
}
```

```
public static void main(String[] args) {
    Connection connection = null;
    PreparedStatement pstmt = null;
    ResultSet resultSet = null;
    try {
        DataSource dataSource = getDataSource();
        connection = dataSource.getConnection();
        System.out.println("The Connection Object is of Class: " + connection.getClass());
        // 开启事务
        connection.setAutoCommit(false);

        // 插入测试数据
        String insertSql = "insert into account(account_number, account_type, account_name)
values(?, ?, ?);";
        PreparedStatement insertStmt = connection.prepareStatement(insertSql);
        insertStmt.setLong (1, 1L);
        insertStmt.setString (2, "manager");
        insertStmt.setString (3, "demotest01");
        insertStmt.executeUpdate();
        connection.commit ();

        // 查询数据
        pstmt = connection.prepareStatement("SELECT * FROM account");
        resultSet = pstmt.executeQuery();
        while (resultSet.next()) {
            String accountNumber = resultSet.getString("account_number");
            String accountType = resultSet.getString("account_type");
            String accountName = resultSet.getString("account_name");
            System.out.println(accountNumber + "," + accountType + "," + accountName);
        }
    } catch (Exception e) {
        try {
            if (null != connection) {
                connection.rollback();
            }
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        e.printStackTrace();
    }
}
```

----结束

# 10 通过 Navicat 客户端连接 DDM 实例

## 操作场景

本文将介绍如何获取弹性公网的IP地址，并通过Navicat客户端连接DDM实例。

## Navicat 客户端连接 DDM 实例

- 步骤1** 登录分布式数据库中间件服务，单击需要连接的DDM实例名称，进入实例基本信息页面。
- 步骤2** 在“实例信息”模块的弹性公网IP单击“绑定”。选择任意地址进行绑定。
- 步骤3** 在DDM管理控制台左侧选择虚拟私有云图标。单击“访问控制>安全组”
- 步骤4** 在安全组界面，单击操作列的“配置规则”，进入安全组详情界面。在安全组详情界面，单击“添加规则”，弹出添加规则窗口。根据界面提示配置安全组规则，设置完成后单击“确定”即可。

### 📖 说明

- 绑定弹性公网IP后，建议您在内网安全组中设置严格的出入规则，以加强数据库安全性。

- 步骤5** 打开Navicat客户端，单击“连接”。在新建连接窗口中填写主机IP地址（弹性公网IP地址）、用户名和密码（DDM账号、密码）。
- 步骤6** 单击“连接测试”，若显示连接成功，单击“确定”，等待1-2分钟即可连接成功。连接失败会直接弹出失败原因，请修改后重试。

### ----结束

### 📖 说明

通过其他可视化的MySQL工具（例如 Workbench）连接DDM实例的操作与此章基本一致，不做详细描述。

# 11 通过数据复制服务（DRS）将 RDS for MySQL 数据迁移到 DDM

## 11.1 方案概述

### 场景描述

本实践主要介绍如何通过数据复制服务（DRS）将不同Region的RDS for MySQL的数据迁移到DDM实例单库中，包含了如何在华为云创建RDS for MySQL、DDM实例，如何通过VPN网络进行数据迁移等内容。

### 前提条件

- 拥有华为账号。
- 账户余额大于等于0美元。

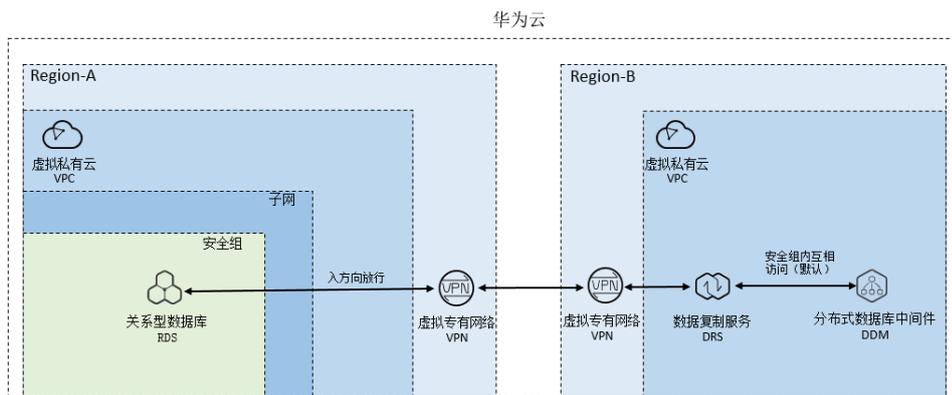
### 服务列表

- 虚拟私有云 VPC
- 虚拟专用网络VPN
- 云数据库 RDS
- 分布式数据库中间件DDM
- 数据复制服务 DRS
- 数据管理服务 DAS

### 部署架构

本示例中，源端为华为云RDS for MySQL，目标端为不同region下的分布式数据库中间件DDM，通过VPN网络，将源库的数据迁移到目标库。部署架构可参考[图11-1](#)。

图 11-1 VPN 场景



## 约束与限制

- 为了保持数据完整性，需要先停止业务后再进行数据迁移。
- 新增RDS for MySQL实例与源RDS for MySQL实例版本需要保持一致。

## 使用说明

- 本实践的资源规划仅作为演示，实际业务场景资源以用户实际需求为准。
- 本实践端到端的数据为测试数据，仅供参考。

## 11.2 资源和成本规划

表 11-1 资源规划

类别	子类	规划	备注
源端VPC	VPC名称	vpc-DRSsrc	自定义，易理解可识别。
	所属Region	亚太-新加坡	选择和自己业务区最近的Region，减少网络时延。
	可用区	可用区二	-
	子网网段	10.0.0.0/24	子网选择时建议预留足够的网络资源。
	子网名称	subnet-drs01	自定义，易理解可识别。
RDS（源库）	RDS实例名	rds-mysql-src	自定义，易理解可识别。
	所属Region	亚太-新加坡	选择和自己业务区最近的Region，减少网络时延。
	数据库版本	MySQL 5.7	-
	实例类型	单机	本示例中为单机。 实际使用时，为提升业务可靠性，推荐选择主备RDS实例。

类别	子类	规划	备注
	存储类型	SSD云盘	-
	可用区	可用区二	本示例中为单机。 实际业务场景推荐选择主备RDS实例，此时建议将两个实例创建在不同的可用区，提升业务可靠性。
	规格	通用型 4 vCPUs   8GB	-
源端VPN	网关名称	vpngw-src	自定义，易理解可识别。
	所属Region	亚太-新加坡	选择和自己业务区最近的Region，减少网络时延。
	虚拟私有云 (VPC)	vpc-DRSsrc	与源端RDS实例VPC保持一致。
	VPN连接名称	vpn-src01	自定义，易理解可识别。
	本端子网	subnet-drs01	与源端RDS实例VPC子网保持一致。
	远端网关	123.60.251.207	即对端VPN的网关，此处为目标端VPN的网关地址，创建目标VPN完成后获取网关信息。
	远端子网	172.16.0.0/24	即对端VPN的子网，此处为目标端VPN的子网，与目标端DDM实例所在VPC的子网保持一致
目标端VPC	VPC名称	vpc-DRStar	自定义，易理解可识别。
	所属Region	中国-香港	选择和自己业务区最近的Region，减少网络时延。
	可用区	可用区一	-
	子网网段	172.16.0.0/24	子网选择时建议预留足够的网络资源。
	子网名称	subnet-drs02	自定义，易理解可识别。
DDM (目标库)	DDM实例名	ddm-drs-tar	自定义，易理解可识别。
	可用区	可用区一	可选择一个或者多个可用区。实际业务场景推荐选择创建在不同的可用区，提升业务可靠性。
	节点规格	通用增强型 8 核   16 GB	-

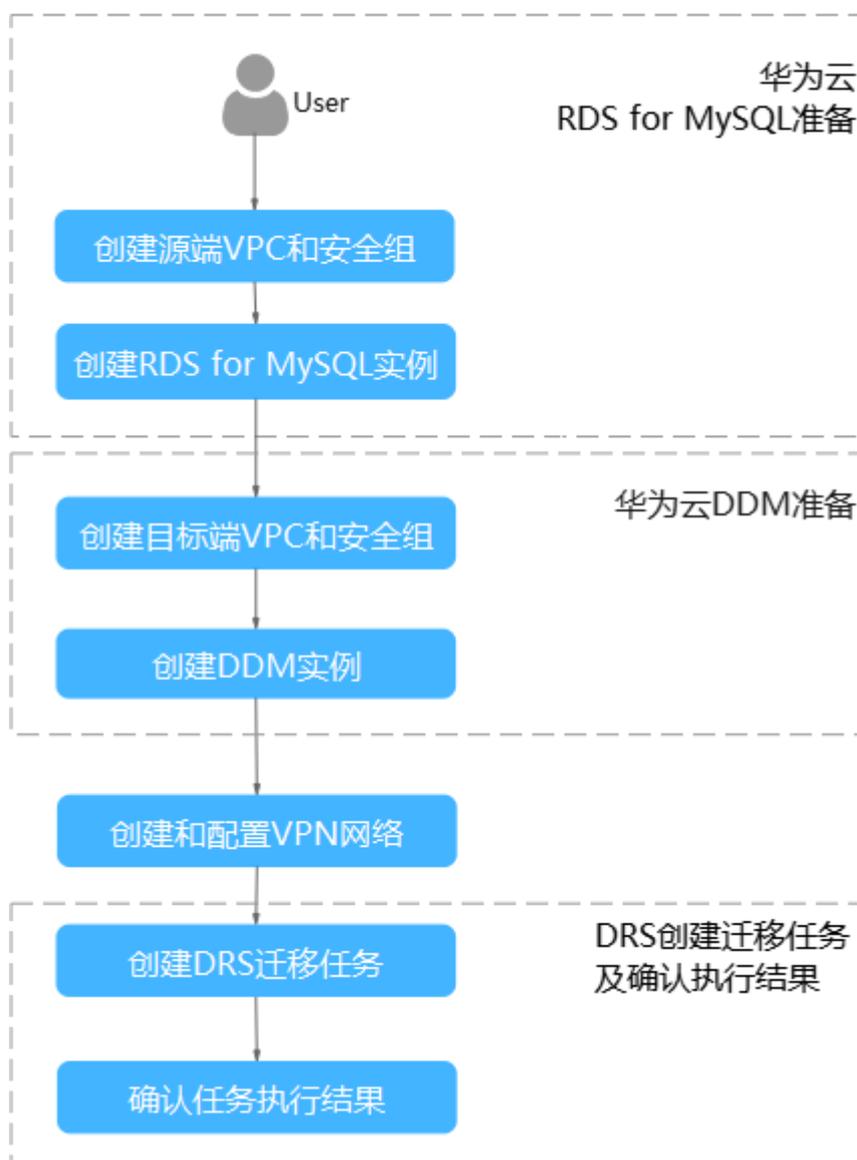
类别	子类	规划	备注
	节点个数	1	单节点存在高可用风险，在实际使用过程中，建议至少创建2节点。
DDM关联RDS实例	RDS实例名	rds-ddm01	自定义，易理解可识别。
	所属Region	中国-香港	选择和自己业务区最近的Region，减少网络时延。
	数据库版本	MySQL 5.7	-
	实例类型	单机	本示例中为单机。 实际使用时，为提升业务可靠性，推荐选择主备RDS实例。
	存储类型	SSD云盘	-
	可用区	可用区一	本示例中为单机。 实际业务场景推荐选择主备RDS实例，此时建议将两个实例创建在不同的可用区，提升业务可靠性。
	规格	通用型 4 vCPUs   8GB	-
目标端VPN	网关名称	vpngw-tar	自定义，易理解可识别。
	所属Region	中国-香港	选择和自己业务区最近的Region，减少网络时延。
	虚拟私有云 ( VPC )	vpc-DRStar	与目标端DDM实例VPC保持一致。
	VPN连接名称	vpn-tar01	自定义，易理解可识别。
	本端子网	subnet-drs02	与目标端DDM实例VPC子网保持一致。
	远端网关	123.60.236.84	即对端VPN的网关，此处为源端VPN的网关地址，创建源端VPN完成后获取网关信息。
	远端子网	10.0.0.0/24	即对端VPN的子网，此处为源端VPN的子网，与源端RDS for MySQL实例所在VPC的子网保持一致。
DRS迁移任务	迁移任务名	DRS-MySQLToDDM	自定义，易理解可识别。
	源数据库引擎	MySQL	本示例中源数据库为MySQL，使用的华为云RDS实例。

类别	子类	规划	备注
	目标数据库引擎	DDM	本示例中目标数据库为DDM。
	网络类型	VPN网络	本示例中采用“VPN网络”。

## 11.3 操作流程

创建RDS for MySQL实例，并且将RDS for MySQL数据同步到DDM的主要任务流程如图11-2所示。

图 11-2 流程图



## 11.4 源端 RDS for MySQL 准备

### 11.4.1 创建源端 VPC 和安全组

创建源端VPC和安全组，为创建RDS for MySQL实例准备好网络资源和安全组。

#### 创建 VPC

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的 ，选择区域“亚太-新加坡”。

**步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟私有云 VPC”。

进入虚拟私有云信息页面。

**步骤4** 单击“创建虚拟私有云”，购买VPC，更多参数的说明请参见[创建虚拟私有云](#)。

**步骤5** 单击“立即创建”。

**步骤6** 返回VPC列表，查看创建VPC是否创建完成。

当VPC列表的VPC状态为“可用”时，表示VPC创建完成。

----结束

#### 创建安全组

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的 ，选择区域“亚太-新加坡”。

**步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟私有云 VPC”。

进入虚拟私有云信息页面。

**步骤4** 在左侧导航栏，选择“访问控制 > 安全组”。

**步骤5** 单击“创建安全组”。

**步骤6** 填写安全组名称等信息。

**创建安全组**

\* 名称: sg-DRS01

\* 企业项目: default [新建企业项目](#) [?](#)

\* 模板: 通用Web服务器

描述: 通用Web服务器, 默认放通22、3389、80、443端口和ICMP协议。适用于需要远程登录、公网ping及用于网站服务的云服务器场景。

查看模板规则 [▼](#)

[确定](#) [取消](#)

**步骤7** 单击“确定”。

**步骤8** 返回安全组列表，单击安全组名称“sg-DRS01”。

**步骤9** 选择“入方向规则”，单击“添加规则”。

**步骤10** 配置入方向规则，放通数据库3306端口。

添加入方向规则 [教程设置](#)

安全组入方向规则为白名单（允许），放通入方向网络流量。

安全组 sg-DRS01

如您要添加多条规则，建议单击导入规则以进行批量导入。

优先级 <a href="#">?</a>	策略	协议端口 <a href="#">?</a>	类型	源地址 <a href="#">?</a>	描述	操作
1-100	允许	TCP 3306	IPv4	IP地址 0.0.0.0/0		<a href="#">复制</a>   <a href="#">删除</a>

[增加1条规则](#)

[确定](#) [取消](#)

----结束

## 11.4.2 创建 RDS for MySQL 实例

本章节介绍创建RDS for MySQL实例并构造测试数据。

### 创建实例

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的 ，选择区域“亚太-新加坡”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 云数据库 RDS”。

**步骤4** 单击“购买数据库实例”。

**步骤5** 配置实例名称和实例基本信息。

图 11-3 计费方式和基本信息

计费模式:  包年/包月  按需计费

区域: 亚太-新加坡

实例名称: rds-53d4

数据库引擎: MySQL PostgreSQL

数据库版本: 8.0 5.7 5.6

实例类型: 主备 单机

存储类型: SSD云盘

主可用区: 可用区1 可用区2 可用区3 可用区5

备可用区: 可用区1 可用区2 可用区3 可用区5

时区: (UTC+08:00) 北京, 重庆, 香港, 乌...

**步骤6** 选择实例规格。

图 11-4 规格与存储

性能规格:  独享型  通用型

vCPUs   内存	建议连接数	TPS   QPS
<input checked="" type="radio"/> 2 vCPUs   4 GB	1,500	580   11,597
<input type="radio"/> 2 vCPUs   8 GB	2,500	590   11,804
<input type="radio"/> 4 vCPUs   8 GB	2,500	1,196   23,914
<input type="radio"/> 4 vCPUs   16 GB	5,000	1,357   27,159
<input type="radio"/> 4 vCPUs   32 GB	10,000	1,435   28,701
<input type="radio"/> 8 vCPUs   16 GB	5,000	2,153   43,058

当前选择实例: rds.mysql.x1.large.2ha | 2 vCPUs | 4 GB (独享型), 建议连接数: 1,500, TPS: 580 | QPS: 11,597

存储空间: 40 GB

磁盘加密:  不加密  加密

**步骤7** 选择实例所属的VPC和安全组、配置数据库端口。

VPC和安全组已在[创建源端VPC和安全组](#)中准备好。

The screenshot shows the configuration interface for a new RDS instance. It includes the following fields and options:

- 虚拟私有云 (VPC):** A dropdown menu set to 'default\_vpc'. To its right is a 'C' icon and a sub-network dropdown set to 'default\_subnet(192.168.0.0/24)'. Further right is another 'C' icon and a text input field for IP address, with a link '查看已使用IP地址 (可用私有IP数量242个)'.
- 数据库端口:** A text input field set to '默认端口3306'. Below it is a note: '创建主实例时只读实例时, 只读实例和主实例数据库端口保持一致。' and a link '查看弹性公网IP'.
- 安全组 (Security Group):** A dropdown menu set to 'default'. To its right is a 'C' icon and a link '查看内网安全组'.

Additional text at the bottom of the form reads: '请确保所选安全组规则允许需要连接实例的服务器能访问3306端口。' and '安全组规则详情 ▲ 设置规则'.

**步骤8** 配置实例密码。

The screenshot shows the password configuration interface. It includes the following elements:

- 设置密码:** Two buttons: '现在设置' (highlighted in blue) and '创建后设置'.
- 管理员账户名:** A text input field containing 'root'.
- 管理员密码:** A text input field with a password icon. To its right is a note: '请妥善保管密码, 系统无法获取您设置的密码内容。'
- 确认密码:** A text input field with a password icon.

**步骤9** 单击“立即购买”。

**步骤10** 返回云数据库实例列表。

当RDS实例运行状态为“正常”时，表示实例创建完成。

----结束

## 构造数据

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的📍，选择区域“亚太-新加坡”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 云数据库 RDS”。

**步骤4** 选择RDS实例，单击实例后的“更多 > 登录”。

**步骤5** 在弹出的对话框中输入实例用户名和密码，单击“测试连接”检查。

**步骤6** 连接成功后单击“登录”，登录RDS实例。

**步骤7** 单击“新建数据库”，创建db\_test测试库。

The screenshot shows a dialog box titled '新建数据库' (New Database). It contains the following fields and buttons:

- 数据库名称 (Database Name):** A text input field containing 'db\_test'. Below it is a note: '只能创建用户数据库'.
- 字符集 (Character Set):** A dropdown menu set to 'utf8mb4'.
- Buttons:** '确定' (Confirm) and '取消' (Cancel).

**步骤8** 在db\_test库中执行如下语句，创建对应的测试表table3\_。

```
CREATE TABLE `db_test`.`table3_` (
  `Column1` INT(11) UNSIGNED NOT NULL,
  `Column2` TIME NULL,
  `Column3` CHAR NULL,
  PRIMARY KEY (`Column1`)
) ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_general_ci;
```



**步骤9** 在测试表table3\_中执行语句，插入三行数据。

```
INSERT INTO `db_test`.`table3_` (`Column1`,`Column2`,`Column3`) VALUES(1,'00:00:11','a');
INSERT INTO `db_test`.`table3_` (`Column1`,`Column2`,`Column3`) VALUES(2,'00:00:22','b');
INSERT INTO `db_test`.`table3_` (`Column1`,`Column2`,`Column3`) VALUES(5,'00:00:55','e');
```



----结束

### 11.4.3 创建源端 VPN

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的📍，选择区域“亚太-新加坡”。

**步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟专用网络 VPN”。

**步骤4** 在左侧导航栏选择“虚拟专用网络 > 企业版-VPN网关”。

**步骤5** 在“VPN网关”界面，单击“创建VPN网关”，更多参数的说明请参见[创建VPN网关](#)。

#### 注意

- VPN网关接入的虚拟私有云（VPC）和源端RDS for MySQL实例的VPC保持一致，即[创建源端VPC和安全组](#)中创建的VPC。
- 本端子网和源端RDS for MySQL实例所在VPC的子网保持一致，即[创建源端VPC和安全组](#)中创建的子网。

**步骤6** 单击“立即购买”，核对VPN信息后单击“提交”完成支付。

**步骤7** VPN创建成功后，即可在列表中查看VPN信息，此时VPN网关的状态为“未连接”。当有VPN连接使用该VPN网关时，VPN网关的状态更新为“正常”。

----结束

## 11.5 目标端 DDM 准备

### 11.5.1 创建目标端 VPC 和安全组

创建目标端VPC和安全组，为创建DDM实例准备好网络资源和安全组。

#### 创建 VPC

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟私有云 VPC”。

进入虚拟私有云信息页面。

**步骤4** 单击“创建虚拟私有云”，购买VPC，更多参数的说明请参见[创建虚拟私有云](#)。

**步骤5** 单击“立即创建”。

**步骤6** 返回VPC列表，查看创建VPC是否创建完成。

当VPC列表的VPC状态为“可用”时，表示VPC创建完成。

----结束

#### 创建安全组

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟私有云 VPC”。

进入虚拟私有云信息页面。

**步骤4** 选择“访问控制 > 安全组”。

**步骤5** 单击“创建安全组”。

**步骤6** 填写安全组名称等信息，单击“确定”。



---结束

## 11.5.2 创建 DDM 实例

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的📍，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。

**步骤4** 在实例管理页面，单击页面右上方的“购买数据库中间件实例”。

**步骤5** 按需设置实例相关信息和规格，相关参数说明请参见[创建DDM实例](#)。

**步骤6** 选择实例所属的VPC和安全组、配置数据库端口。

VPC和安全组已在[创建目标端VPC和安全组](#)中准备好。

**步骤7** 实例信息设置完成后，单击页面下方“立即购买”。

**步骤8** 实例创建成功后，用户可以在“实例管理”页面对其进行查看和管理。

数据库端口默认为5066，实例创建成功后可修改。当实例运行状态为“运行中”时，表示实例创建完成。

----结束

### 11.5.3 创建 RDS for MySQL 实例

本章节介绍创建DDM下关联的RDS for MySQL实例，创建步骤如下。

#### 创建步骤

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的📍，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 云数据库 RDS”。

**步骤4** 单击“购买数据库实例”。

**步骤5** 配置实例名称和实例基本信息。

图 11-5 计费方式和基本信息

The screenshot displays the configuration interface for creating a new RDS instance. The settings are as follows:

- 计费模式 (Billing Mode):** 包年/包月 (Pay-as-you-go) is selected.
- 区域 (Region):** 亚太-新加坡 (Asia-Pacific - Singapore).
- 实例名称 (Instance Name):** rds-53d4.
- 数据库引擎 (Database Engine):** MySQL is selected.
- 数据库版本 (Database Version):** 5.7 is selected.
- 实例类型 (Instance Type):** 主备 (Primary/Standby) is selected.
- 存储类型 (Storage Type):** SSD云盘 (SSD Cloud Disk) is selected.
- 主可用区 (Primary Availability Zone):** 可用区1 (AZ1) is selected.
- 备可用区 (Secondary Availability Zone):** 可用区1 (AZ1) is selected.
- 时区 (Time Zone):** (UTC+08:00) 北京, 重庆, 香港, 乌... (UTC+08:00) Beijing, Chongqing, Hong Kong, U... is selected.

**步骤6** 选择实例规格。

图 11-6 规格与存储

The screenshot shows the 'Performance Specifications' (性能规格) section with two tabs: 'Dedicated Type' (独享型) and 'General Type' (通用型). A table lists various configurations with columns for vCPUs | Memory, Recommended Connections, and TPS | QPS. The selected configuration is '2 vCPUs | 4 GB' with 1,500 connections and 580 TPS | 11,597 QPS. Below the table, a storage configuration section shows a slider for storage space, currently set to 40 GB, with an auto-expand option set to 10%.

vCPUs   内存	建议连接数	TPS   QPS
2 vCPUs   4 GB	1,500	580   11,597
2 vCPUs   8 GB	2,500	590   11,804
4 vCPUs   8 GB	2,500	1,196   23,914
4 vCPUs   16 GB	5,000	1,357   27,159
4 vCPUs   32 GB	10,000	1,435   28,701
8 vCPUs   16 GB	5,000	2,153   43,058

当前选择实例: rds.mysql.x1.large.2.hs | 2 vCPUs | 4 GB (独享型), 建议连接数: 1,500, TPS | QPS: 580 | 11,597

存储空间: 40 GB (slider range: 40 to 4000)

云数据库RDS为您提供相同大小的备份存储空间, 超出部分按照OBS计费规则收取费用。

存储空间自动扩容 可用存储空间率: 10% 存储自动扩容上限: 4,000 GB

自动扩容, 会自动按扩容量扣除存储费用。了解更多

可用存储空间<=10%或者10GB时, 自动扩容当前存储空间的20% (非10倍数向上取整, 账户余额不足, 会导致自动扩容失败)。

磁盘加密:  加密

**步骤7** 选择实例所属的VPC和安全组、配置数据库端口。

VPC和安全组已在[创建目标端VPC和安全组](#)中准备好。

**注意**

RDS for MySQL实例的虚拟私有云 (VPC) 和子网必须和DDM实例保持一致。

The screenshot shows the configuration page for VPC and security group. The 'Virtual Private Cloud' (虚拟私有云) is set to 'default\_vpc' and the 'Subnet' (子网) is set to 'default\_subnet(192.168.0.0/24)'. The 'Database Port' (数据库端口) is set to 'Default 3306'. The 'Security Group' (安全组) is set to 'default'. A note indicates that the VPC and subnet must be consistent with the DDM instance.

虚拟私有云: default\_vpc | 子网: default\_subnet(192.168.0.0/24)

数据库端口: 默认端口 3306

安全组: default

目前RDS实例创建完成后不支持切换虚拟私有云与子网, 请谨慎选择。不同虚拟私有云里的弹性云服务器网络默认不通, 如需创建新的虚拟私有云, 可前往控制台创建。通过公网访问数据库实例需要购买绑定弹性公网IP。查看弹性公网IP

创建主实例加只读实例时, 只读实例和主实例数据库端口保持一致。

请确保所选安全组规则允许需要连接实例的服务器能访问3306端口。安全组规则详情 > 设置规则

**步骤8** 配置实例密码。

The screenshot shows the 'Set Password' (设置密码) section. The 'Admin Username' (管理员账户名) is set to 'root'. The 'Admin Password' (管理员密码) and 'Confirm Password' (确认密码) fields are empty. A note indicates that the password must be complex.

设置密码: 现在设置 | 创建后设置

管理员账户名: root

管理员密码: [input field]

确认密码: [input field]

请妥善保管密码, 系统无法获取您设置的密码内容。

**步骤9** 单击“立即购买”。

**步骤10** 返回云数据库实例列表。

当RDS实例运行状态为“正常”时，表示实例创建完成。

----结束

## 11.5.4 创建逻辑库并关联 RDS for MySQL 实例

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。

**步骤4** 在实例管理页面，单击实例对应操作栏的“创建逻辑库”。

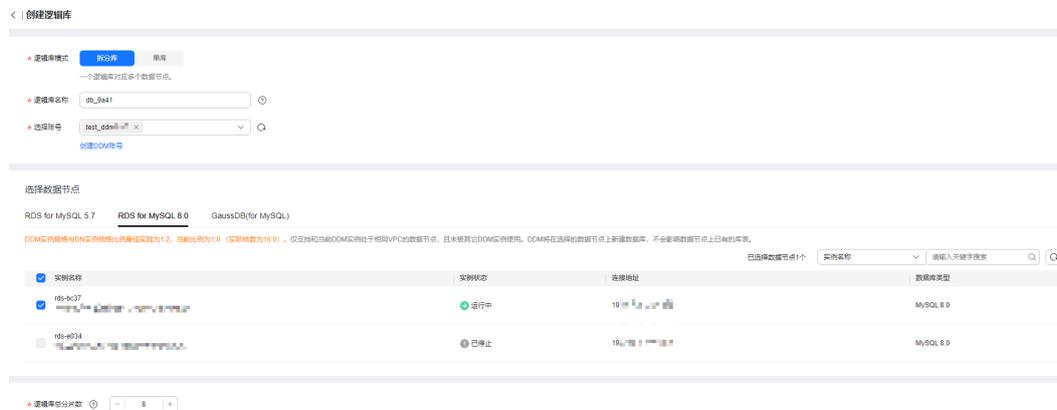
**步骤5** 在创建逻辑库页面，选择“逻辑库模式”、“逻辑库总分片数”，填写“逻辑库名称”，并选择要关联的DDM账号、要关联的实例，单击“下一步”。

本示例中逻辑库模式为单库，逻辑库名称为db\_test。

### 注意

DRS目前仅支持迁移源端MySQL的数据到目标DDM，不支持迁移源库表结构及其他数据库对象，用户需要在目标库根据源端数据库的表结构创建对应的逻辑库。

图 11-7 关联 RDS for MySQL 实例



----结束

## 11.5.5 创建 DDM 账号

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。

**步骤4** 在实例管理页面，单击实例名称，进入实例基本信息页面。

**步骤5** 在左侧导航栏选择“账号管理”，进入账号管理页面。

**步骤6** 在账号管理页面单击“创建DDM账号”，在弹窗中填选账号信息、关联的逻辑库和权限。

DDM目标库账号所需要的权限可参考[DRS使用须知](#)中的数据库权限说明。

**创建DDM账号**

\* 账号名称: dbtest

\* 密码: .....

\* 确认密码: .....

密码有效期 (天): 180

关联逻辑库: db\_7723

\* 账号权限:  全选  
 CREATE  DROP  ALTER  INDEX  INSERT  DELETE  
 UPDATE  SELECT

描述: 请输入描述 (0/256)

确定 取消

**步骤7** 信息填写完成，单击“确定”即可创建成功。

----结束

## 11.5.6 创建目标库表结构

DRS目前仅支持迁移源端MySQL的数据到目标DDM，不支持迁移源库表结构及其他数据库对象。用户需要在目标库根据源端逻辑库的表结构，自行在目标库创建对应的表结构及索引。未在目标库创建的对象，视为用户不选择这个对象进行迁移。更对约束限制可参考[使用须知](#)。

### 操作步骤

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的📍，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。

**步骤4** 在实例管理页面，单击实例对应操作栏的“登录”。

- 步骤5** 在弹出的对话框中输入[创建DDM账号](#)中的DDM账户名和密码，单击“测试连接”检查。
- 步骤6** 连接成功后单击“登录”，登录DDM实例。
- 步骤7** 单击[创建逻辑库并关联RDS for MySQL实例](#)中创建的逻辑库db\_test。
- 步骤8** 在db\_test库中执行如下语句，创建与源端结构一致的同名目标表table3\_。

```
CREATE TABLE `db_test`.`table3_` (  
  `Column1` INT(11) UNSIGNED NOT NULL,  
  `Column2` TIME NULL,  
  `Column3` CHAR NULL,  
  PRIMARY KEY (`Column1`)  
) ENGINE = InnoDB  
  DEFAULT CHARACTER SET = utf8mb4  
  COLLATE = utf8mb4_general_ci;
```



----结束

## 11.5.7 创建目标端 VPN

- 步骤1** 登录[华为云控制台](#)。
- 步骤2** 单击管理控制台左上角的📍，选择区域“中国-香港”。
- 步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟专用网络 VPN”。
- 步骤4** 在左侧导航栏选择“虚拟专用网络 > 企业版-对端网关”。
- 步骤5** 在“VPN网关”界面，单击“创建对端网关”，更多参数的说明请参见[创建对端网关](#)。

### ⚠ 注意

标识中的IP Address和目标端DDM实例所在VPC的子网保持一致，即[创建目标端VPC和安全组](#)中创建的子网。

- 步骤6** 单击“立即创建”。

----结束

## 11.6 创建 VPN 连接

- 步骤1** 登录[华为云控制台](#)。
- 步骤2** 单击管理控制台左上角的 ，选择区域“亚太-新加坡”。
- 步骤3** 单击左侧的服务列表图标，选择“网络 > 虚拟专用网络 VPN”。
- 步骤4** 在左侧导航栏选择“虚拟专用网络 > 企业版-VPN连接”。
- 步骤5** 在“VPN连接”页面，单击“创建VPN连接”，更多参数的说明请参见[创建VPN连接](#)。

### 注意

对端网关和对端子网为目标端VPN的网关名称和子网，参考[创建目标端VPN](#)中创建的VPN信息进行填写。

- 步骤6** 单击“立即购买”，核对VPN连接信息后单击“提交”完成支付。
  - 步骤7** 配置完成后，即可在列表中查看VPN信息，此时VPN网关的状态为“正常”。
- 结束

## 11.7 创建 DRS 迁移任务

本章节介绍如何创建DRS迁移任务，将华为云不同Region的RDS for MySQL上的数据库迁移到华为云DDM。

### 迁移前检查

在创建任务前，需要针对迁移条件进行手工自检，以确保您的同步任务更加顺畅。  
本示例为MySQL到DDM入云迁移，您可以参考[迁移使用须知](#)获取相关信息。

### 创建迁移任务

- 步骤1** 登录[华为云控制台](#)。
  - 步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。
  - 步骤3** 单击左侧的服务列表图标，选择“数据库 > 数据复制服务 DRS”。
  - 步骤4** 单击“创建迁移任务”，进入创建迁移任务页面。
  - 步骤5** 在“迁移实例”页面，填选区域、项目、任务名称、描述、迁移实例信息，单击“开始创建”。
- 更多参数的说明请参见[MySQL迁移至DDM参数说明](#)。
- 步骤6** 在“源库及目标库”页面，待迁移实例创建成功后，填选源库信息和目标库信息，单击源库和目标库处的“测试连接”，分别测试并确定与源库和目标库连通后，勾选协议，单击“下一步”。

**步骤7** 在“迁移设置”页面，设置迁移对象。

迁移对象：表级迁移。

**步骤8** 单击“下一步”，在“预检查”页面，进行迁移任务预校验，校验是否可进行任务迁移。

- 查看检查结果，如有不通过的检查项，需要修复不通过项后，单击“重新校验”按钮重新进行迁移任务预校验。
- 预检查完成后，且预检查通过率为100%时，单击“下一步”。

#### 说明

所有检查项结果均通过时，若存在待确认项，需要阅读并确认详情后才可以继续执行下一步操作。

**步骤9** 在“任务确认”页面，设置迁移任务的启动时间、任务异常通知设置、SMN主题、时延阈值、任务异常自动结束时间，并确认迁移任务信息无误后，单击“启动任务”，提交迁移任务。

**步骤10** 迁移任务提交后，您可在“实时迁移管理”页面，查看并[管理任务](#)。

- 您可查看任务提交后的状态，状态请参见[任务状态说明](#)。
- 在任务列表的右上角，单击  刷新列表，可查看到最新的任务状态。
- 对于未启动、状态为配置中的任务，DRS默认保留3天，超过3天DRS会自动删除后台资源，当前任务状态不变。当用户再次配置时，DRS会重新申请资源。

---结束

## 11.8 确认迁移执行结果

确认迁移结果可参考如下两种方式：

1. DRS会针对迁移对象、数据等维度进行对比，从而给出迁移结果，详情参见在[DRS管理控制台查看迁移结果](#)。
2. 直接登录数据库查看库、表、数据是否迁移完成。手工确认数据迁移情况，详情参见在[DDM管理控制台查看迁移结果](#)。

### 在 DRS 管理控制台查看迁移结果

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的  ，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 数据复制服务 DRS”。

**步骤4** 单击DRS实例名称。

**步骤5** 单击左侧“迁移对比”。

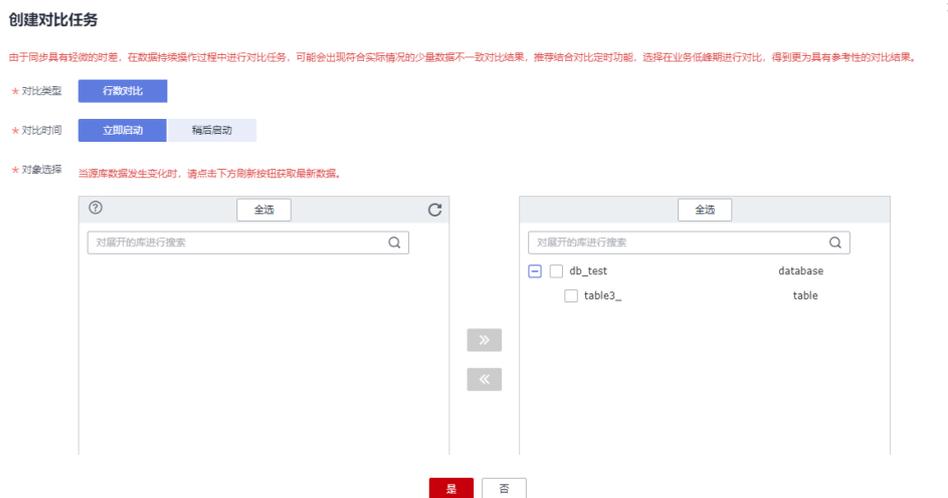
**步骤6** 选择“对象级对比”页签，查看对象是否缺失。

单击“开始对比”，对比完成后，查看对比结果。



**步骤7** 选择“数据级对比”页签，查看迁移对象行数是否一致。

1. 单击“创建对比任务”。
2. 在弹出的对话框中选择对比类型、时间和对象。



3. 等待对比任务结束后，查看数据对比结果。



4. 如需查看对比明细，可单击对比任务后的“查看对比报表”。



---结束

## 在 DDM 管理控制台查看迁移结果

**步骤1** 登录[华为云控制台](#)。

- 步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。
- 步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。
- 步骤4** 选择DDM实例，单击迁移的目标实例的操作列的“登录”。
- 步骤5** 在弹出的对话框中输入密码，单击“测试连接”检查。
- 步骤6** 连接成功后单击“登录”。
- 步骤7** 查看并确认目标库名和表名等，确认相关数据是否迁移完成。

----结束

# 12 通过 DDM 对 RDS for MySQL 实例进行分库分表

---

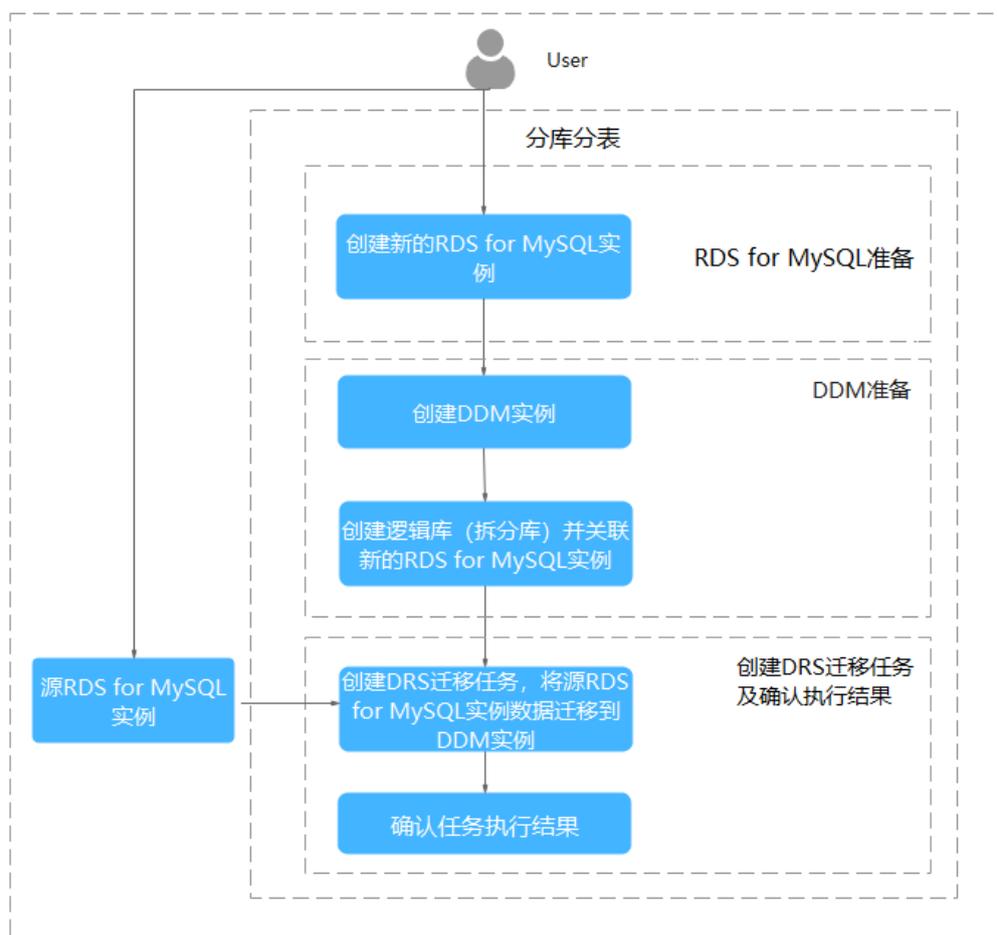
## 12.1 方案概述

### 操作场景

本文将介绍如何通过DDM对已有RDS for MySQL实例进行分库分表。

## 操作流程

图 12-1 流程图



## 前提条件

- 拥有华为账号。
- 新的RDS for MySQL实例已创建成功。
- 源RDS for MySQL实例与目标DDM实例、新RDS for MySQL实例必须处于相同VPC，且安全组配置相同。

## 12.2 RDS for MySQL 准备

此操作默认新的RDS for MySQL实例已创建成功。如需创建实例，请参考[购买RDS for MySQL实例](#)。

## 12.3 DDM 准备

## 12.3.1 创建 DDM 实例

### 操作步骤

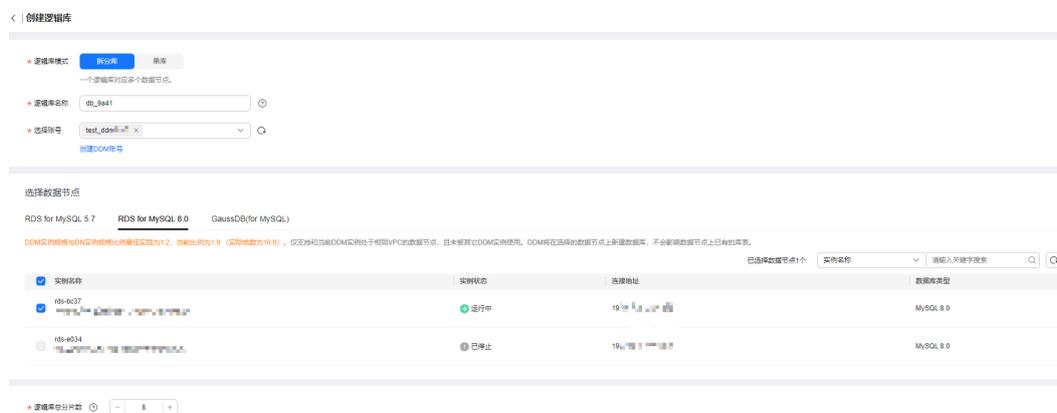
- 步骤1** 登录[华为云控制台](#)。
  - 步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。
  - 步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。
  - 步骤4** 在实例管理页面，单击页面右上方的“购买数据库中间件实例”。
  - 步骤5** 按需设置实例相关信息和规格。参数说明请参考[实例购买](#)。
  - 步骤6** 实例信息设置完成后，单击页面下方“立即购买”。
  - 步骤7** 确认配置信息，根据所选实例的计费模式进行后续操作。
- 结束

## 12.3.2 创建逻辑库并关联 RDS for MySQL 实例

### 操作步骤

- 步骤1** 登录[华为云控制台](#)。
- 步骤2** 单击管理控制台左上角的 ，选择区域“中国-香港”。
- 步骤3** 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。
- 步骤4** 在实例管理页面，单击实例对应操作栏的“创建逻辑库”。
- 步骤5** 逻辑库模式选择拆分库，根据业务需求填写逻辑库总分片数，关联[RDS for MySQL准备](#)中已创建成功的新RDS for MySQL实例，然后单击“下一步”。

图 12-2 关联 RDS for MySQL 实例



- 步骤6** 输入RDS for MySQL实例密码，进行数据节点可用性检测。测试成功后，单击“完成”。

步骤7 逻辑库创建成功。

图 12-3 创建逻辑库成功



----结束

## 12.3.3 创建 DDM 账号

### 操作步骤

- 步骤1 登录[华为云控制台](#)。
- 步骤2 单击管理控制台左上角的📍，选择区域“中国-香港”。
- 步骤3 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。
- 步骤4 在实例管理页面，单击实例名称，进入实例基本信息页面。
- 步骤5 单击左侧导航栏的“账号管理”页签，进入账号管理页面。
- 步骤6 单击“创建DDM账号”按钮进入创建DDM账号页面。
- 步骤7 选择[创建逻辑库并关联RDS for MySQL实例](#)中创建的逻辑库进行关联，设置用户名和密码以及账号权限，单击“确定”按钮进行账号创建。

图 12-4 创建 DDM 账号

创建DDM账号

\* 账号名称: dbtest

\* 密码: .....

\* 确认密码: .....

密码有效期 (天): 180

关联逻辑库: db\_7723

\* 账号权限:  全选  
 CREATE  DROP  ALTER  INDEX  INSERT  DELETE  
 UPDATE  SELECT

描述: 请输入描述

0/256

确定 取消

步骤8 DDM账号创建成功。

----结束

## 12.3.4 创建目标库表结构

### 操作步骤

步骤1 登录[华为云控制台](#)。

步骤2 单击管理控制台左上角的 ，选择区域“中国-香港”。

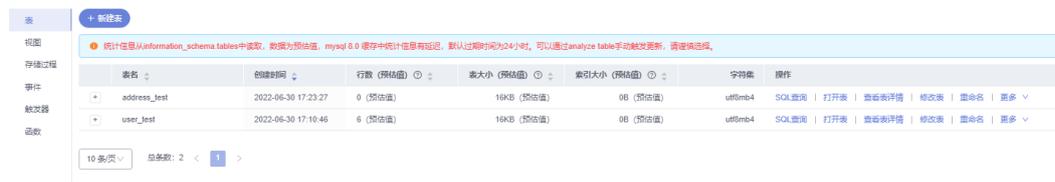
步骤3 单击左侧的服务列表图标，选择“数据库 > 分布式数据库中间件 DDM”，进入DDM管理控制台。

步骤4 在实例管理页面，单击实例对应操作栏的“登录”按钮。

步骤5 在实例登录窗口输入[创建DDM账号](#)中创建的DDM账号和密码，进入数据管理服务进行建表操作。

步骤6 建表操作是将源RDS for MySQL数据库中的表在DDM实例中重新创建，根据表的性质可选择创建为广播表、单表、拆分表。广播表和单表的使用场景请参考[广播表和单表的使用场景](#)。

图 12-5 原数据库表



表名	创建时间	行数 (预估值)	表大小 (预估值)	索引大小 (预估值)	字符集	操作
address_test	2022-06-30 17:23:27	0 (预估值)	16KB (预估值)	0B (预估值)	utf8mb4	SQL查询   打开表   查看表详情   修改表   重命名   更多
user_test	2022-06-30 17:10:46	6 (预估值)	16KB (预估值)	0B (预估值)	utf8mb4	SQL查询   打开表   查看表详情   修改表   重命名   更多

图 12-6 address\_test 表数据



id	name	address
1	wang	xxxxx8802
2	wang	xxxxx8802
3	wang	xxxxx8801

图 12-7 user\_test 表数据



id	age	name
1	82	zha
2	63	qian
3	15	wang

步骤7 执行以下命令，将源数据库中的“user\_test”创建为广播表，将“address\_test”创建为拆分表。建表语句可参考[创建表](#)。

```
create table user_test (id char(3), age int(3), name varchar(255),primary key(id)) broadcast;
```

图 12-8 创建广播表



```
create table address_test (id char(3), name varchar(255), address varchar(255),primary key(id)) dbpartition
by hash(id);
```

图 12-9 创建拆分表



**步骤8** 查看建表结果。

图 12-10 查看建表结果



----结束

## 12.4 创建 DRS 迁移任务

### 12.4.1 创建 DRS 迁移任务

本章节介绍如何创建DRS迁移任务，将源RDS for MySQL实例数据迁移到新DDM实例。

#### 操作步骤

**步骤1** 登录[华为云控制台](#)。

**步骤2** 单击管理控制台左上角的，选择区域“中国-香港”。

**步骤3** 单击左侧的服务列表图标，选择“数据库 > 数据复制服务 DRS”。

**步骤4** 单击“创建迁移服务”按钮，创建一个迁移任务，将源RDS for MySQL实例的数据迁移到DDM实例中。

**步骤5** 填写实例迁移信息，更多参数的说明请参见[MySQL迁移至DDM参数说明](#)。

- 数据流动方向选择“入云”。
- 源数据库引擎选择“MySQL”。
- 目标数据库引擎选择“DDM”。
- 网络类型根据实际情况选择。
- 目标数据库实例选择[创建DDM实例](#)中购买成功的DDM实例。
- 迁移模式可根据您的业务情况选择“全量+增量”或者是“全量”。

**步骤6** 编辑迁移任务，在源库信息栏填入源RDS for MySQL实例的IP地址、端口、用户名密码，单击“测试连接”按钮。目标库信息栏输入DDM账户和密码，单击“测试连接”按钮。两方都连接成功后，单击“下一步”。

**步骤7** 选择迁移对象，然后单击“下一步”。

**步骤8** 在“预检查”页面，进行迁移任务预校验，校验是否可进行迁移。如有不通过的检查项，需要修复不通过项后，单击“重新校验”按钮重新进行迁移任务预校验。所有检查项结果均成功时，单击“下一步”。

图 12-11 预检查



步骤9 启动迁移任务。

图 12-12 启动迁移任务



---结束

## 12.4.2 确认迁移执行结果

### 操作步骤

**步骤1** 待迁移任务结束后，可登录DDM实例查看数据迁移结果。确保DDM实例数据与源数据库数据一致。

图 12-13 查看 user\_test 表数据

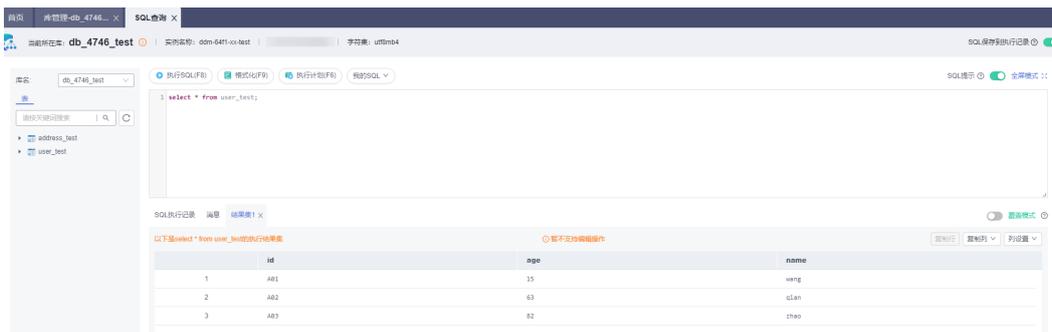
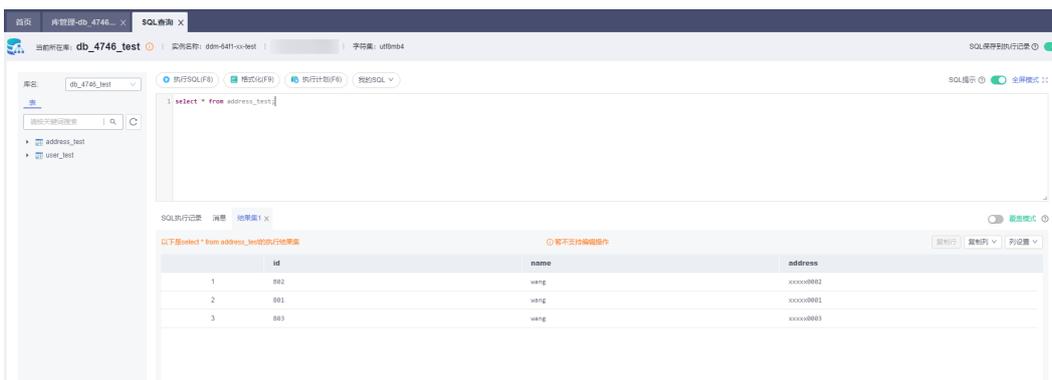


图 12-14 查看 address\_test 表数据



**步骤2** 执行以下命令查看表数据分布情况。广播表各分片数据一致，拆分表数据会按照拆分算法分布在选取的分片中。

```
show topology from <table_name>;
```

图 12-15 广播表 user\_test

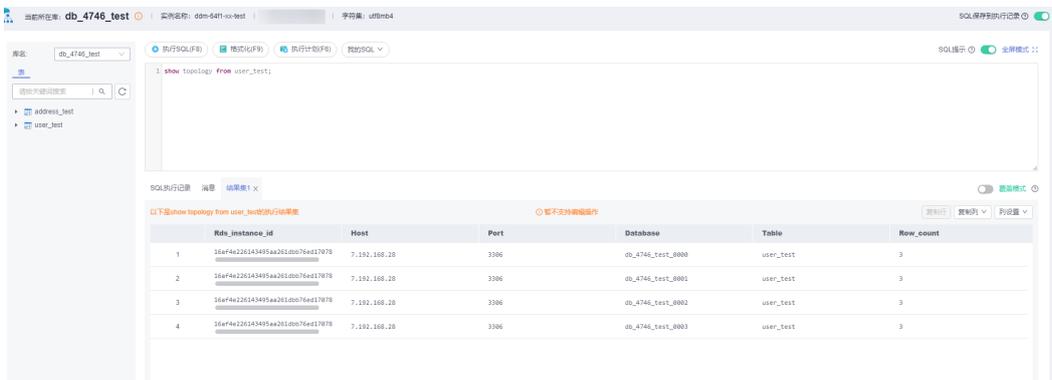
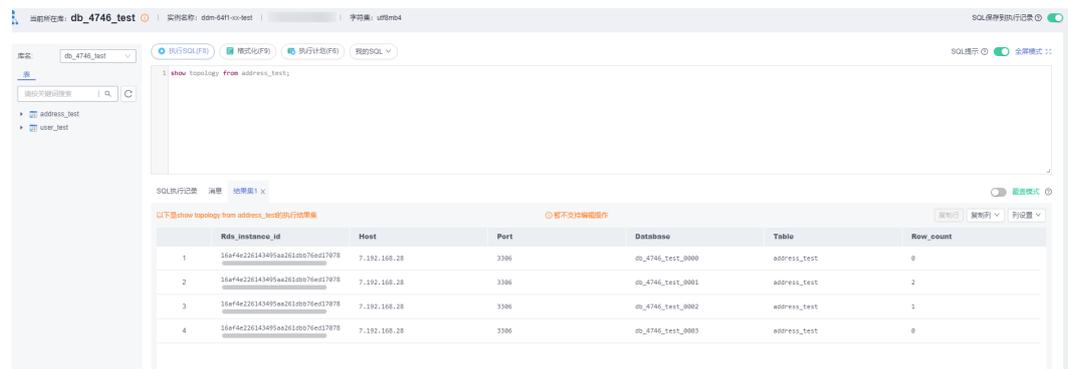


图 12-16 拆分表 address\_test



----结束

# 13 如何选择拆分键和拆分算法

本章节主要介绍拆分键和拆分算法的使用场景，以及如何合理的选择拆分键和拆分算法。

相关介绍请参见[表13-1](#)。

**表 13-1** 拆分键与拆分算法使用简介

拆分算法	hash类		range类	
拆分键	表字段	表字段+日期函数	表字段	表字段+日期函数
详细说明	根据指定的表字段将数据平均拆分到各个分片上。	根据指定的表字段+日期函数将数据平均拆分到各个分片上。 表字段必须是日期类型（date、datetime、timestamp）	将数据表内的记录按照算法元数据定义的规则将数据拆分到指定的分片上。	根据指定的表字段+日期函数将数据按照算法元数据的规则将数据拆分到各个分片上。 表字段必须是日期类型（date、datetime、timestamp）

<p><b>适用场景</b></p>	<p>适用于需要将数据均匀分布的场景，例如：银行类客户业务应用，业务逻辑主体是客户，可使用客户对应的表字段（例如客户号）作为拆分键，详情参见如下示例。</p>	<p>需要按时间（年、月、日、周及其组合）对数据进行拆分的场景，例如：游戏类的应用，可使用玩家对应的表字段（例如玩家注册时间）作为拆分键，按日、月、年等函数分片，方便统计和查询某日、月玩家的操作数据，帮助游戏厂家做大数据分析。</p>	<p>适合范围类操作较多的场景，例如：电商类应用，如果业务场景是围绕商家做活动进行，业务逻辑主体是活动日期，可使用活动日期对应的表字段（例如活动名称、日期范围）作为拆分键，方便统计某周期内销量等情况。</p>	<p>例如日志分析场景，日志系统中可能包含各类复杂的信息，这时您可以选择时间字段作为拆分键，然后对拆分键使用日期函数拆分。</p> <p>为了方便日志清理和转储，采用range拆分算法，对时间字段用日期函数转换成年，表示按年存储到各个分片上，详情参见如下示例。</p>
--------------------	---	---	--	--

## 如何选择拆分算法

拆分算法即将逻辑表中数据拆分到多个数据库分片上的算法，DDM支持hash和range两种拆分算法。

- hash类  
将数据均匀分布在各个分片。  
适用于SQL查询条件使用“=”、“IN”之类运算符相对较多的场景。
- range类  
将数据表内的记录按照算法原数据的取值范围进行分片存储。  
适用于SQL查询条件使用“>”、“<”、“BETWEEN ... AND ...”之类运算符相对较多的场景。

拆分算法的使用，需要结合业务查询场景进行评估，以选择合适的拆分算法，提升DDM效率。

### 📖 说明

在对逻辑库进行扩容时：

- 按hash算法分片的逻辑表，表数据涉及到迁移，会被重新均匀分布所有分片上。
- 按range算法分片的逻辑表，表数据默认不做迁移，如果您修改了拆分规则，重新定义范围分布，请自行直连RDS分片完成数据迁移。

## 如何选择拆分键

拆分键是在水平拆分逻辑表的过程中，用于生成路由结果的表字段，指定表字段后，可以进一步选择日期函数，也可以手动输入“日期函数(字段名)”，数据表字段必须是日期类型（date、datetime、timestamp），日期函数适用于需要按时间（年、月、日、周及其组合）对数据进行拆分的场景。

DDM根据拆分键与拆分算法计算路由结果，对拆分表的数据进行自动水平拆分，分发到数据分片中。

选取拆分算法与拆分键一般遵循以下规则：

- 尽可能使数据均匀分布到各个分片上。
- 该拆分键是最频繁或者最重要的查询条件。
- 优选主键作为拆分键，因为主键作为查询条件时，查询速度最快。

## 有明确主体的业务场景

拆分表的数据量一般都达到千万级，因此选择合适的拆分算法和拆分键非常重要。如果能找到业务主体，并且确定绝大部分的数据库操作都是围绕这个主体的数据进行的，那么可以选择这个主体所对应的表字段作为拆分键，进行水平拆分。

业务逻辑主体与实际应用场景相关，下列场景都有明确的业务逻辑主体。

1. 银行类客户业务应用，业务逻辑主体是客户，可使用客户对应的表字段（例如客户号）作为拆分键。部分业务系统的业务场景为围绕银行卡/账号的，可以选取卡/账号作为拆分键。
2. 电商类应用，如果业务场景是围绕商品进行操作，业务逻辑主体是商品，可使用商品对应的表字段（例如商品编码）作为拆分键。
3. 游戏类的应用，主要围绕玩家数据进行操作，业务逻辑主体是玩家，可使用玩家对应的表字段（例如玩家id）作为拆分键。

以银行类客户业务为例，建表SQL语句如下：

```
CREATE TABLE PERSONALACCOUNT (  
ACCOUNT VARCHAR(20) NOT NULL PRIMARY KEY,  
NAME VARCHAR(60) NOT NULL,  
TYPE VARCHAR(10) NOT NULL,  
AVAILABLEBALANCE DECIMAL(18,2) NOT NULL,  
STATUS CHAR(1) NOT NULL,  
CARDNO VARCHAR(24) NOT NULL,  
CUSTOMID VARCHAR(15) NOT NULL  
) ENGINE=INNODB DEFAULT CHARSET=UTF8;
```

## 无明确主体的业务场景

如果业务场景中找不到合适的主体，也可以选择那些数据分布较为均匀的属性所对应的表字段作为拆分键。

例如日志分析场景，日志系统中可能包含各类复杂的信息，这时您可以选择时间字段作为拆分键。

选择时间字段作为拆分键时，支持对拆分键使用日期函数拆分。

为了方便清理和转储，采用range拆分算法，对时间字段用日期函数转换成年，表示按年存储到各个分片上。

建表SQL语句：

```
CREATE TABLE LOG (  
LOGTIME DATETIME NOT NULL,  
LOGSOURCESYSTEM VARCHAR(100),  
LOGDETAIL VARCHAR(10000)  
);
```

# 14 跨分片复杂 SQL 调优

本章节主要针对跨分片复杂SQL时的调优操作。

## Group by 和 Order by SQL 调优建议

1. 当group by 字段不是表中现有的数据列时，可以使用generated column机制（MySQL5.7版本以上支持，5.6及之前版本可新建普通字段和添加索引），创建新列并添加索引，将新创建字段作为group by条件，避免使用临时表。
2. 保持group by 和order by 字段相同，其中group by字段中包含分片键时性能最优。
3. 尽量让group by和order by过程中使用索引。

## Join 相关 SQL 调优建议

1. 建议进行join操作时，join条件优先使用分片键等值连接。
2. 用结果集小的表作为驱动表，尽量减少join的循环次数（使用left join/join时，左边是驱动表，右边是被驱动表；使用right join时，右边是驱动表，左边是被驱动表）。
3. 对于被驱动表的join字段需添加索引，不建议两张大表直接进行join，尤其是包含不带限制条件的单表，将消耗大量计算资源。
4. 不建议在显示事务中使用join语句，为保证事务内数据可见性，可能无法使用最优join执行计划。

## Union SQL 调优建议

1. 不建议大表间进行union（不包含union all）操作，避免去重时超出临时表限制。
2. 不建议union中使用order by 和group by。

## 临时表调优建议

跨数据分片在执行join、union、group by、order by等复杂SQL时，DDM可能会创建临时表暂时存储数据，对临时表大小限制默认是100万行。超出限制时，会提示“Temp table limit exceeded”错误，说明当前执行SQL需要在DDM节点进行二次计算，并且所需要的临时表大小已超过实例设置。对于以上情况如遇到临时表相关报错，可评估SQL是否存在优化空间，如设置广播表、增加分批查询条件或根据分库分表架构自身特点调整SQL写法。如果无优化空间，同时评估数据量大小总体可控（参

与实际计算的数据不超过300万行)、实例资源相对充裕,可酌情在控制台调整实例级参数temp\_table\_size\_limit来放开对应限制。

## 临时表产生主要场景

1. group by和order by字段不同。
2. 使用了distinct关键字。
3. 执行union查询。
4. 部分跨分片的子查询。
5. 在join查询中:使用了非等值连接条件、两个子查询之间的连接或开启了事务。