

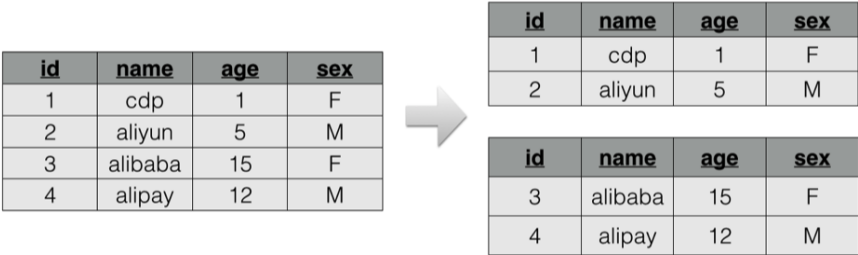
数据集成

进阶与深入

进阶与深入

切分与并行

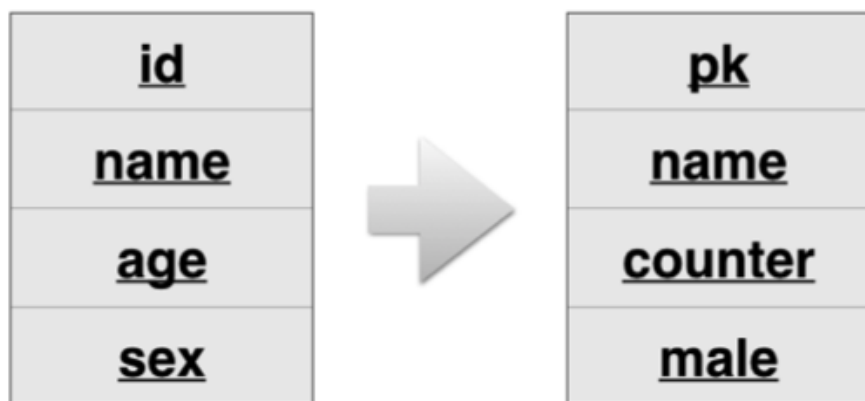
为最大化数据同步的 Job 传输吞吐量，数据集成将对 Job 进行更细粒度的任务切分，并启动多个数据传输单元进行并行数据同步。数据集成的切分规则由数据集成框架定义接口，具体切分规则交由具体传输插件负责实现，例如：对于 ODPS 而言，数据切分根据 ODPS 表的 Record 偏移量计算数据分片。对于Mysql Reader插件而言，其切分算法按照 Mysql 数据表主键范围切分。



以 Mysql 抽取数据为例，MysqlReader 启动后将连接源数据端，计算数据表主键范围，如[Min, Max]，并根据用户带宽需求切分上述[Min, Max]区间范围，每个子区间将作为一个Task提交给后续传输单元进行执行。例如，上述 Mysql 数据表主键范围在[1,4]，MysqlReader根据用户需要的吞吐量计算得出数据切分规则需要两个数据子集，因此将数据区间区分为[1, 2]、[3, 4]两个数据子集，两个数据子集交由不同的同步传输单元完成。

数据字段映射

数据集成数据同步面向是结构化、半结构化的数据同步，数据集成对于源宿两端数据源均会抽象为一张二维表进行传输和映射。数据集成对于字段的映射是按照配置字段顺序进行——映射，而非依赖源宿两端字段名称。



数据集成本身不对源宿两端字段做任何约束，视用户填写字段顺序——传递。由于存在异构源宿两端字段个数可能不一致情况，数据集成做如下约定：

- 源端字段和目标端字段个数一致

数据集成推荐配置源宿两端数据字段个数一致，避免数据字段补空情况。

- 源端字段比目标端字段多

源端字段比目标端字段列数多意味着一定存在数据丢失情况，为避免数据丢失，数据集成对于这类配置直接报错。

- 源端字段比目标端字段少

数据集成对该类情况直接在导入数据末尾添加 null，做补空处理。

类型转换

由于历史遗留问题，数据集成在面对各类异构数据同步过程仍然难以避免源宿两端数据类型不匹配情况，例如用户需要从 Mysql 的 Varchar 类型导入 ODPS 的 BIGINT 类型。对于较为通用的数据类型转换数据集成自身提供了统一的框架级定义，方便用户需要在不同数据类型中做转换。

类型 转换	Long	Double	String	Date	Bool	Byte
Long	Y	Y	Y	[1]	[2]	N
Double	Y	Y	Y	N	N	N
String	[3]	[4]	Y	[5]	[6]	N
Date	[7]	N	[8]	Y	N	N
Bool	[9]	N	Y	N	Y	N
Bytes	N	N	[10]	N	N	Y

数据集成默认的类型转换规则处理了大部分通用的数据转换规则。针对部分数据类型转换，数据集成规则需要用户知晓：

- [1] Long转换为Date，将Long数值视作Java提供Unix时间戳转换为Date，精度为毫秒。
- [2] Long转换为Bool，0视为false，否则为true
- [3] String转换为Long，整数型字符串支持转换为Long，否则报错。不支持科学计数法字符串。
- [4] String转换为Double，浮点型字符串支持转换为Long，否则报错。不支持科学计数法。
- [5] String转换为Date，用户自定义时间格式(例如YYYY-MM-DD)，满足用户时间格式可以转换，否则报错。
- [6] String转换为Bool，“true”转为true，“false”转为false，其他报错。
- [7] Date转为Long，将Date转为Unix时间戳。
- [8] Date转换为String，根据用户自定义时间格式(例如YYYY-MM-DD)转换为字符串。
- [9] Bool转换为Long，true转为1，false转为0。
- [10] Bytes转换为String，根据用户指定的字符集编码转为String。

注意：数据集成传输过程中自动进行编码转换，例如数据集成自动从 Mysql GBK 的字段导入 ODPS UTF-8字段，用户不需要干预。

增量数据同步

增量数据同步是数据同步业务层面的实现，底层数据的同步分为批量Snapshot同步和变更操作日志同步。

数据集成Job通过数据存储数据API接口完成批量数据同步功能，因此数据集成Job本身不支持获取数据存储系统层面的变化过程。例如数据集成无法通过获取解析RDS BinLog日志数据变化量并同步到目的数据源。

若您有增量业务的需求，不仅可通过系统级别增量变化日志解析实现数据增量同步，也可通过以下方案进行解决。

时间戳方式：它是一种基于快照比较的变化数据捕获方式，在源表上增加一个时间戳字段，系统中更新修改表数据的时候，同时修改时间戳字段的值。

当进行数据抽取时，通过比较上次抽取时间与时间戳字段的值来决定抽取哪些数据。有的数据库的时间戳支持自动更新，即表的其它字段的数据发生改变时，自动更新时间戳字段的值。有的数据库不支持时间戳的自动更新，这就要求业务系统在更新业务数据时，手工更新时间戳字段。

在阿里数据库使用实践中，所有的在线业务系统按惯例都默认添加了create_time、modify_time，对于在线业务写入均自动修改上述两个字段，记录修改时间。数据集成在进行数据同步过程中，仅需要在MySQL抽取环节配置where过滤条件做增量抽取。

触发器方式：抽取的表上建立需要的触发器，一般要建立插入、修改、删除三个触发器，每当源表中的数据发生变化，就被相应的触发器将变化的数据写入一个临时表，抽取线程从临时表中抽取数据，临时表中抽取过的数据被标记或删除。对于此类应用方式，数据集成在进行数据同步时仅需指定抽取触发器写入表即可。

表方式：在业务系统中添加系统日志表，当业务数据发生变化时，更新维护日志表内容，当作ETL加载时，通过读日志表数据决定加载那些数据及如何加载。对于此类应用方式，数据集成同样使用Where过滤条件做增量抽取。

日志表方式：在业务系统中添加系统日志表，当业务数据发生变化时，更新维护日志表内容，当作ETL加载时，通过读日志表数据决定加载那些数据及如何加载。对于此类应用方式，数据集成同样使用Where过滤条件做增量抽取。

脏数据和容错

数据同步通常会对接源端和目标端的数据存储，需要根据来源和目标两端数据源的具体信息适配和转换相应的数据内容。在传输过程中，可能存在由于两端元数据不匹配或者本身的业务数据传输转换失败的情况，例如OSS上一个定义为Integer的类型存放了abc字符串，数据集成将自动识别上述异常情况，并提供自动记录和容错机制，最大限度保证数据传输的可靠性和健壮性。

数据集成基于自动识别脏数据功能上，还提供数据传输容错上限。例如：由于历史遗留问题，您本身知晓脏数据影响情况并且对于源端脏数据有一定容忍度。您可以配置单个Job最大脏数据条数阈值，或者数据脏数据阈值。默认情况下，数据集成对脏数据不设限制。