————————————————————————————— 第**5**章 **Sqoop**一些常用命令及参数

**5.1** 常用命令列举

这里给大家列出来了一部分Sqoop操作时的常用参数，以供参考，需要深入学习的可以参 看对应类的源代码。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 命令 |  |  | 类 |  |  | 说明 |  |
|  | 1 |  |  | import |  |  | ImportTool |  |  | 将数据导入到集群 |  |
|  | 2 |  |  | export |  |  | ExportTool |  |  | 将集群数据导出 |  |
|  | 3 |  |  | codegen |  |  | CodeGenTool |  |  | 获取数据库中某张表 |  |
|  |  |  | 数据生成Java并打包 |
|  | | |  | | |  | | | Jar |
|  | 4 |  |  | create-hive-table |  |  | CreateHiveTableTool |  |  | 创建Hive表 |  |
|  | 5 |  |  | eval |  |  | EvalSqlTool |  |  | 查看SQL执行结果 |  |
|  | 6 |  |  | import-all-tables |  |  | ImportAllTablesTool |  |  | 导入某个数据库下所 |  |
|  |  |  | 有表到HDFS中 |
|  | 7 |  |  | job |  |  | JobTool |  |  | 用来生成一个sqoop |  |
|  |  |  | 的任务，生成后，该 |
|  | | |  |  | | | 任务并不执行，除非 |
|  | | | 使用命令执行该任 |
| 务。 |
|  | 8 |  |  | list-databases |  |  | ListDatabasesTool |  |  | 列出所有数据库名 |  |
|  | 9 |  |  | list-tables |  |  | ListTablesTool |  |  | 列出某个数据库下所 |  |
|  |  |  | 有表 |
|  | 10 |  |  | merge |  |  | MergeTool |  |  | 将HDFS中不同目录 |  |
|  |  |  | 下面的数据合在一 |
|  | | |  | | |  | | | 起，并存放在指定的 |
| 目录中 |
|  | 11 |  |  | metastore |  |  | MetastoreTool |  |  | 记录sqoop job的元 |  |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  |  |  |  | | |  | 数据信息，如果不启 |  |
|  | 动metastore实例，则 |
|  | | | 默认的元数据存储目 |
| 录为：~/.sqoop，如果 |
| 要更改存储目录，可 |
| 以在配置文件 |
| sqoop-site.xml中进行 |
| 更改。 |
|  | 12 |  |  | help |  |  | HelpTool |  |  | 打印sqoop帮助信息 |  |
|  | 13 |  |  | version |  |  | VersionTool |  |  | 打印sqoop版本信息 |  |

**5.2** 命令**&**参数详解

刚才列举了一些Sqoop的常用命令，对于不同的命令，有不同的参数，让我们来一一列举 说明。   
首先来我们来介绍一下公用的参数，所谓公用参数，就是大多数命令都支持的参数。   
**5.2.1** 公用参数：数据库连接

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --connect |  |  | 连接关系型数据库的URL |  |
|  | 2 |  |  | --connection-manager |  |  | 指定要使用的连接管理类 |  |
|  | 3 |  |  | --driver |  |  | Hadoop根目录 |  |
|  | 4 |  |  | --help |  |  | 打印帮助信息 |  |
|  | 5 |  |  | --password |  |  | 连接数据库的密码 |  |
|  | 6 |  |  | --username |  |  | 连接数据库的用户名 |  |
|  | 7 |  |  | --verbose |  |  | 在控制台打印出详细信息 |  |

**5.2.2** 公用参数：**import**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --enclosed-by <char> |  |  | 给字段值前加上指定的字符 |  |
|  | 2 |  |  | --escaped-by <char> |  |  | 对字段中的双引号加转义符 |  |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 |  |  | --fields-terminated-by <char> |  |  | 设定每个字段是以什么符号 |  |
|  |  | 作为结束，默认为逗号 |
|  | 4 |  |  | --lines-terminated-by <char> |  |  | 设定每行记录之间的分隔符 | ， |
|  |  | 默认是\n |
|  | 5 |  |  | --mysql-delimiters |  |  | Mysql默认的分隔符设置，字 |  |
|  |  | 段之间以逗号分隔，行之间以 |
|  | | |  | | | \n分隔，默认转义符是\，字 |
| 段值以单引号包裹。 |
|  | 6 |  |  | --optionally-enclosed-by |  |  |  |  |
|  | <char> |  |

**5.2.3** 公用参数：**export**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --input-enclosed-by <char> |  |  | 对字段值前后加上指定字符 |  |
|  | 2 |  |  | --input-escaped-by <char> |  |  | 对含有转移符的字段做转义 |  |
|  |  | 处理 |
|  | 3 |  |  | --input-fields-terminated-by |  |  |  |  |
|  | <char> |  |
|  | 4 |  |  | --input-lines-terminated-by |  |  | 行之间的分隔符 |  |
|  | <char> |  |
|  | 5 |  |  | --input-optionally-enclosed-by |  |  | 给带有双引号或单引号的字 |  |
|  | <char> | 段前后加上指定字符 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 用自定义的字符串替换掉数 |  |
|  |  | 据中的\r\n和\013 \010等字符 |
|  | 2 |  |  | --hive-drop-import-delims |  |  | 在导入数据到hive时，去掉 |  |
|  |  | 数据中的\r\n\013\010这样的 |
|  | | |  | | | 字符 |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 |  |  | --map-column-hive <arg> |  |  | 生成hive表时，可以更改生 |  |
|  |  | 成字段的数据类型 |
|  | 4 |  |  | --hive-partition-key |  |  | 创建分区，后面直接跟分区 |  |
|  |  | 名，分区字段的默认类型为 |
|  | | |  | | | string |
|  | 5 |  |  | --hive-partition-value <v> |  |  | 导入数据时，指定某个分区的 |  |
|  |  | 值 |
|  | 6 |  |  | --hive-home <dir> |  |  | hive的安装目录，可以通过该 |  |
|  |  | 参数覆盖之前默认配置的目 |
|  | | |  | | | 录 |
|  | 7 |  |  | --hive-import |  |  | 将数据从关系数据库中导入 |  |
|  |  | 到hive表中 |
|  | 8 |  |  | --hive-overwrite |  |  | 覆盖掉在hive表中已经存在 |  |
|  |  | 的数据 |
|  | 9 |  |  | --create-hive-table |  |  | 默认是false，即，如果目标 |  |
|  |  | 表已经存在了，那么创建任务 |
|  | | |  | | | 失败。 |
|  | 10 |  |  | --hive-table |  |  | 后面接要创建的hive表,默认 |  |
|  |  | 使用MySQL的表名 |
|  | 11 |  |  | --table |  |  | 指定关系数据库的表名 |  |

公用参数介绍完之后，我们来按照命令介绍命令对应的特有参数。   
**5.2.5** 命令**&**参数：**import**

将关系型数据库中的数据导入到HDFS（包括Hive，HBase）中，如果导入的是Hive，那么 当Hive中没有对应表时，则自动创建。

**1)** 命令：

如：导入数据到hive中

$ bin/sqoop import \

—————————————————————————————

|  |  |  |
| --- | --- | --- |
|  | --connect jdbc:mysql://hadoop102:3306/company \ |  |
| --username root \ |
| --password 000000 \ |
| --table staff \ |
| --hive-import |

如：增量导入数据到hive中，mode=append

|  |  |  |
| --- | --- | --- |
|  | append导入： |  |
| $ bin/sqoop import \ |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff \ |
| --num-mappers 1 \ |
| --fields-terminated-by "\t" \ |
| --target-dir /user/hive/warehouse/staff\_hive \ |
| --check-column id \ |
| --incremental append \ |
| --last-value 3 |

尖叫提示：append不能与--hive-等参数同时使用（Append mode for hive imports is not yet supported. Please remove the parameter --append-mode）

如：增量导入数据到hdfs中，mode=lastmodified

|  |  |  |
| --- | --- | --- |
|  | 先在mysql中建表并插入几条数据： |  |
| mysql> create table company.staff\_timestamp(id int(4), name varchar(255), sex varchar(255), |
| last\_modified timestamp DEFAULT CURRENT\_TIMESTAMP ON UPDATE |
| CURRENT\_TIMESTAMP); |
| mysql> insert into company.staff\_timestamp (id, name, sex) values(1, 'AAA', 'female'); |
| mysql> insert into company.staff\_timestamp (id, name, sex) values(2, 'BBB', 'female'); |

—————————————————————————————

|  |  |  |
| --- | --- | --- |
|  | 先导入一部分数据： |  |
| $ bin/sqoop import \ |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff\_timestamp \ |
| --delete-target-dir \ |
| --m 1 |
| 再增量导入一部分数据： |
| mysql> insert into company.staff\_timestamp (id, name, sex) values(3, 'CCC', 'female'); |
| $ bin/sqoop import \ |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff\_timestamp \ |
| --check-column last\_modified \ |
| --incremental lastmodified \ |
| --last-value "2017-09-28 22:20:38" \ |
| --m 1 \ |
| --append |

尖叫提示：使用lastmodified方式导入数据要指定增量数据是要--append（追加）还是要   
--merge-key（合并）

尖叫提示：last-value指定的值是会包含于增量导入的数据中

**2)** 参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --append |  |  | 将数据追加到HDFS中已经 | 。 |
|  |  | 存在的DataSet中，如果使用 |
|  | | |  | | | 该参数，sqoop会把数据先导 |
| 入到临时文件目录，再合并 |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 |  |  | --as-avrodatafile |  |  | 将数据导入到一个Avro数据 |  |
|  |  | 文件中 |
|  | 3 |  |  | --as-sequencefile |  |  | 将数据导入到一个sequence |  |
|  |  | 文件中 |
|  | 4 |  |  | --as-textfile |  |  | 将数据导入到一个普通文本 |  |
|  |  | 文件中 |
|  | 5 |  |  | --boundary-query <statement> |  |  | 边界查询，导入的数据为该参 |  |
|  |  | 数的值（一条sql语句）所执 |
|  | | |  | | | 行的结果区间内的数据。 |
|  | 6 |  |  | --columns <col1, col2, col3> |  |  | 指定要导入的字段 |  |
|  | 7 |  |  | --direct |  |  | 直接导入模式，使用的是关系 | ， |
|  |  | 数据库自带的导入导出工具 |
|  | | |  | | | 以便加快导入导出过程。 |
|  | 8 |  |  | --direct-split-size |  |  | 在使用上面direct直接导入 |  |
|  |  | 的基础上，对导入的流按字节 |
|  | | |  | | | 分块，即达到该阈值就产生一 |
| 个新的文件 |
|  | 9 |  |  | --inline-lob-limit |  |  | 设定大对象数据类型的最大 |  |
|  |  | 值 |
|  | 10 |  |  | --m或–num-mappers |  |  | 启动N个map来并行导入数 |  |
|  |  | 据，默认4个。 |
|  | 11 |  |  | --query或--e <statement> |  |  | 将查询结果的数据导入，使用 |  |
|  |  | 时必须伴随参--target-dir， |
|  | | |  | | | --hive-table，如果查询中有 |
| where条件，则条件后必须加 |
| 上$CONDITIONS关键字 |
|  | 12 |  |  | --split-by <column-name> |  |  | 按照某一列来切分表的工作 |  |
|  |  | 单元，不能与 |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  | --autoreset-to-one-mapper连 |  |
| 用（请参考官方文档） |
|  | 13 |  |  | --table <table-name> |  |  | 关系数据库的表名 |  |
|  | 14 |  |  | --target-dir <dir> |  |  | 指定HDFS路径 |  |
|  | 15 |  |  | --warehouse-dir <dir> |  |  | 与14参数不能同时使用，导 |  |
|  |  | 入数据到HDFS时指定的目 |
|  | | |  | | | 录 |
|  | 16 |  |  | --where |  |  | 从关系数据库导入数据时的 |  |
|  |  | 查询条件 |
|  | 17 |  |  | --z或--compress |  |  | 允许压缩 |  |
|  | 18 |  |  | --compression-codec |  |  | 指定hadoop压缩编码类，默 |  |
|  |  | 认为gzip(Use Hadoop codec |
|  | | |  | | | default gzip) |
|  | 19 |  |  | --null-string <null-string> |  |  | string类型的列如果null，替 |  |
|  |  | 换为指定字符串 |
|  | 20 |  |  | --null-non-string <null-string> |  |  | 非string类型的列如果null， |  |
|  |  | 替换为指定字符串 |
|  | 21 |  |  | --check-column <col> |  |  | 作为增量导入判断的列名 |  |
|  | 22 |  |  | --incremental <mode> |  |  | mode：append或lastmodified |  |
|  | 23 |  |  | --last-value <value> |  |  | 指定某一个值，用于标记增量 |  |
|  |  | 导入的位置 |

**5.2.6** 命令**&**参数：**export**

从HDFS（包括Hive和HBase）中奖数据导出到关系型数据库中。   
**1)** 命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop export \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |

—————————————————————————————

|  |  |  |
| --- | --- | --- |
|  | --password 000000 \ |  |
| --table staff \ |
| --export-dir /user/company \ |
| --input-fields-terminated-by "\t" \ |
| --num-mappers 1 |

**2)** 参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --direct |  |  | 利用数据库自带的导入导出 |  |
|  |  | 工具，以便于提高效率 |
|  | 2 |  |  | --export-dir <dir> |  |  | 存放数据的HDFS的源目录 |  |
|  | 3 |  |  | -m或--num-mappers <n> |  |  | 启动N个map来并行导入数 |  |
|  |  | 据，默认4个 |
|  | 4 |  |  | --table <table-name> |  |  | 指定导出到哪个RDBMS中 |  |
|  |  | 的表 |
|  | 5 |  |  | --update-key <col-name> |  |  | 对某一列的字段进行更新操 |  |
|  |  | 作 |
|  | 6 |  |  | --update-mode <mode> |  |  | updateonly |  |
|  |  | allowinsert(默认) |
|  | 7 |  |  | --input-null-string |  |  |  |  |
|  | <null-string> |  |
|  | 8 |  |  | --input-null-non-string |  |  | 请参考import该类似参数说 |  |
|  | <null-string> | 明 |
|  |  |  |  |  |  |  | 创建一张临时表，用于存放所 |  |
|  |  | 有事务的结果，然后将所有事 |
|  | | |  | 务结果一次性导入到目标表 |
|  | | | 中，防止错误。 |
|  | 10 |  |  | --clear-staging-table |  |  | 如果第9个参数非空，则可以 |  |

—————————————————————————————

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 在导出操作执行前，清空临时 |  |
| 事务结果表 |

**5.2.7** 命令**&**参数：**codegen**

将关系型数据库中的表映射为一个Java类，在该类中有各列对应的各个字段。 如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop codegen \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff \ |
| --bindir /home/admin/Desktop/staff \ |
| --class-name Staff \ |
| --fields-terminated-by "\t" |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --bindir <dir> |  |  | 指定生成的Java文件、编译 |  |
|  |  | 成的class文件及将生成文件 |
|  | | |  | | | 打包为jar的文件输出路径 |
|  | 2 |  |  | --class-name <name> |  |  | 设定生成的Java文件指定的 |  |
|  |  | 名称 |
|  | 3 |  |  | --outdir <dir> |  |  | 生成Java文件存放的路径 |  |
|  | 4 |  |  | --package-name <name> |  |  | 包名，如com.z，就会生成com |  |
|  |  | 和z两级目录 |
|  | 5 |  |  | --input-null-non-string |  |  |  |  |
|  | <null-str> |  |
|  | | |  |  |
|  | | |  |

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 6 |  |  | --input-null-string <null-str> |  |  | 将null字符串替换成想要替 | ） |
|  |  | 换的值（一般与5同时使用 |
|  | 7 |  |  | --map-column-java <arg> |  |  | 数据库字段在生成的Java文 |  |
|  |  | 件中会映射成各种属性，且默 |
|  | | |  | | | 认的数据类型与数据库类型 |
| 保持对应关系。该参数可以改 |
| 变默认类型，例如： |
| --map-column-java id=long, |
| name=String |
|  | 8 |  |  | --null-non-string <null-str> |  |  | 在生成Java文件时，可以将 |  |
|  |  | 不存在或者null的字符串设 |
|  | | |  | | | 置为其他值 |
|  | 9 |  |  | --null-string <null-str> |  |  | 在生成Java文件时，将null |  |
|  |  | 字符串设置为其他值（一般与 |
|  | | |  | | | 8同时使用） |
|  | 10 |  |  | --table <table-name> |  |  | 对应关系数据库中的表名，生 |  |
|  |  | 成的Java文件中的各个属性 |
|  | | |  | | | 与该表的各个字段一一对应 |

**5.2.8** 命令**&**参数：**create-hive-table**

生成与关系数据库表结构对应的hive表结构。   
命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop create-hive-table \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff \ |
| --hive-table hive\_staff |

————————————————————————————— 参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --hive-home <dir> |  |  | Hive的安装目录，可以通过 |  |
|  |  | 该参数覆盖掉默认的Hive目 |
|  | | |  | | | 录 |
|  | 2 |  |  | --hive-overwrite |  |  | 覆盖掉在Hive表中已经存在 |  |
|  |  | 的数据 |
|  | 3 |  |  | --create-hive-table |  |  | 默认是false，如果目标表已 |  |
|  |  | 经存在了，那么创建任务会失 |
|  | | |  | | | 败 |
|  | 4 |  |  | --hive-table |  |  | 后面接要创建的hive表 |  |
|  | 5 |  |  | --table |  |  | 指定关系数据库的表名 |  |

**5.2.9** 命令**&**参数：**eval**

可以快速的使用SQL语句对关系型数据库进行操作，经常用于在import数据之前，了解一 下SQL语句是否正确，数据是否正常，并可以将结果显示在控制台。

命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop eval \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --query "SELECT \* FROM staff" |

参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --query或--e |  |  | 后跟查询的SQL语句 |  |

**5.2.10** 命令**&**参数：**import-all-tables**

可以将RDBMS中的所有表导入到HDFS中，每一个表都对应一个HDFS目录

————————————————————————————— 命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop import-all-tables \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --warehouse-dir /all\_tables |

参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --as-avrodatafile |  |  | 这些参数的含义均和import |  |
|  | 2 |  |  | --as-sequencefile |  | 对应的含义一致 |
|  | 3 |  |  | --as-textfile |  |  |
|  | 4 |  |  | --direct |  |  | | |
|  | 5 |  |  | --direct-split-size <n> |  |
|  | 6 |  |  | --inline-lob-limit <n> |  |
|  | 7 |  |  | --m或—num-mappers <n> |  |
|  | 8 |  |  | --warehouse-dir <dir> |  |
|  | 9 |  |  | -z或--compress |  |
|  | 10 |  |  | --compression-codec |  |

**5.2.11** 命令**&**参数：**job**

用来生成一个sqoop任务，生成后不会立即执行，需要手动执行。 命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop job \ |  |
| --create myjob -- import-all-tables \ |
| --connect jdbc:mysql://hadoop102:3306/company \ |

—————————————————————————————

|  |  |  |
| --- | --- | --- |
|  | --username root \ |  |
| --password 000000 |
| $ bin/sqoop job \ |
| --list |
| $ bin/sqoop job \ |
| --exec myjob |

尖叫提示：注意import-all-tables和它左边的--之间有一个空格   
尖叫提示：如果需要连接metastore，则--meta-connect jdbc:hsqldb:hsql://linux01:16000/sqoop 参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --create <job-id> |  |  | 创建job参数 |  |
|  | 2 |  |  | --delete <job-id> |  |  | 删除一个job |  |
|  | 3 |  |  | --exec <job-id> |  |  | 执行一个job |  |
|  | 4 |  |  | --help |  |  | 显示job帮助 |  |
|  | 5 |  |  | --list |  |  | 显示job列表 |  |
|  | 6 |  |  | --meta-connect <jdbc-uri> |  |  | 用来连接metastore服务 |  |
|  | 7 |  |  | --show <job-id> |  |  | 显示一个job的信息 |  |
|  | 8 |  |  | --verbose |  |  | 打印命令运行时的详细信息 |  |

尖叫提示：在执行一个job时，如果需要手动输入数据库密码，可以做如下优化

|  |  |  |
| --- | --- | --- |
|  | <property> |  |
| <name>sqoop.metastore.client.record.password</name> |
| <value>true</value> |
| <description>If true, allow saved passwords in the metastore.</description> |
| </property> |

**5.2.12** 命令**&**参数：**list-databases**

命令：  
如：

Sqoop —————————————————————————————

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop list-databases \ |  |
| --connect jdbc:mysql://hadoop102:3306/ \ |
| --username root \ |
| --password 000000 |

参数：与公用参数一样

**5.2.13** 命令**&**参数：**list-tables**

命令：

如：

|  |  |  |
| --- | --- | --- |
|  | $ bin/sqoop list-tables \ |  |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 |

参数：与公用参数一样

**5.2.14** 命令**&**参数：**merge**

将HDFS中不同目录下面的数据合并在一起并放入指定目录中   
数据环境：

|  |  |  |
| --- | --- | --- |
|  | new\_staff |  |
| 1 AAA male |
| 2 BBB male |
| 3 CCC male |
| 4 DDD male |
| old\_staff |
| 1 AAA female |
| 2 CCC female |
| 3 BBB female |
| 6 DDD female |

尖叫提示：上边数据的列之间的分隔符应该为\t，行与行之间的分割符为\n，如果直接复制，

————————————————————————————— 请检查之。

命令：

如：

|  |  |  |
| --- | --- | --- |
|  | 创建JavaBean： |  |
| $ bin/sqoop codegen \ |
| --connect jdbc:mysql://hadoop102:3306/company \ |
| --username root \ |
| --password 000000 \ |
| --table staff \ |
| --bindir /home/admin/Desktop/staff \ |
| --class-name Staff \ |
| --fields-terminated-by "\t" |
|  |
| 开始合并： |
| $ bin/sqoop merge \ |
| --new-data /test/new/ \ |
| --onto /test/old/ \ |
| --target-dir /test/merged \ |
| --jar-file /home/admin/Desktop/staff/Staff.jar \ |
| --class-name Staff \ |
| --merge-key id |
| 结果： |
| 1 AAA MALE |
| 2 BBB MALE |
| 3 CCC MALE |
| 4 DDD MALE |
| 6 DDD FEMALE |

参数：

—————————————————————————————

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --new-data <path> |  |  | HDFS 待合并的数据目录， |  |
|  |  | 合并后在新的数据集中保留 |
|  | 2 |  |  | --onto <path> |  |  | HDFS合并后，重复的部分在 |  |
|  |  | 新的数据集中被覆盖 |
|  | 3 |  |  | --merge-key <col> |  |  | 合并键，一般是主键ID |  |
|  | 4 |  |  | --jar-file <file> |  |  | 合并时引入的jar包，该jar |  |
|  |  | 包是通过Codegen工具生成 |
|  | | |  | | | 的jar包 |
|  | 5 |  |  | --class-name <class> |  |  | 对应的表名或对象名，该 |  |
|  |  | class类是包含在jar包中的 |
|  | 6 |  |  | --target-dir <path> |  |  | 合并后的数据在HDFS里存 |  |
|  |  | 放的目录 |

**5.2.15** 命令**&**参数：**metastore**

记录了Sqoop job的元数据信息，如果不启动该服务，那么默认job元数据的存储目录为 ~/.sqoop，可在sqoop-site.xml中修改。

命令：

如：启动sqoop的metastore服务

$ bin/sqoop metastore

参数：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 序号 |  |  | 参数 |  |  | 说明 |  |
|  | 1 |  |  | --shutdown |  |  | 关闭metastore |  |