第 3 章 Kettle

学习目标

- 了解 Kettle 的含义，能够描述 Kettle 的作用和特点；
- 了解 Kettle 的安装与启动，能够在 Windows 操作系统中安装和启动 Kettle；
- 熟悉 Kettle 的转换和作业，能够描述 Kettle 中转换和作业的作用；
- 掌握 Kettle 的基本操作，能够独立完成转换管理、作业管理和数据库连接的相关操作。

“工欲善其事，必先利其器”这句古语，深刻地阐述了在进行某项工作时，选择和使用合适的工具的重要性。在开源 ETL 工具中，Kettle 是一款常见且广泛使用的工具，掌握它的基本用法对于从事相关工作的人来说是非常必要的。因此，本章将详细讲解 Kettle 这款 ETL 工具的相关知识和基本操作。

3.1 初识 Kettle

3.1.1 Kettle 简介

Kettle 是一款基于 Java 语言开发的开源 ETL 工具，其设计理念是将来自不同数据源的数据视为水流，将水流汇聚在一个象征“水壶”的容器中，然后按照用户预设的形式流出。这形象地说明了 Kettle 的功能是将来自多个数据源的数据整合和转换为用户需要的形式。

Kettle 提供了图形化的用户界面，用户可以通过可视化的方式设计数据处理操作，而不需要关心具体的实现细节。这使得非技术人员也能轻松上手 Kettle，快速搭建复杂的 ETL。Kettle 工具主要由 4 个组件构成，分别是 Spoon、Pan、Kitchen 及 Carte，下面分别介绍它们的功能。

1. Spoon

Spoon 是 Kettle 的图形化开发环境，也是最常用的组件之一，它提供了一个直观的界面，使用户能够可视化地管理转换（transformation）和作业（job），并且还可以监控 ETL 的执行过程。转换和作业是 Kettle 中的基本概念，会在后续进行详细讲解。

2. Pan

Pan 是 Kettle 的命令行工具，用于批量运行转换。用户可以通过 Pan 在命令行界面中指定要运行的转换和相关参数，以便自动化和批量化地运行转换。

3. Kitchen

Kitchen是Kettle的命令行工具,用于批量运行作业。用户可以通过Kitchen在命令行界面中指定要运行的作业和相关参数,以便自动化和批量化地运行作业。

4. Carte

Carte是一个轻量级的Web服务器,允许用户远程执行和监控Kettle中的转换和作业。值得一提的是,Carte还支持集群部署,这意味着用户可以在不同的服务器上运行转换和作业,从而实现ETL在分布式环境中的执行,以提高性能和可伸缩性。

3.1.2 Kettle的特点

Kettle是一款功能强大的ETL工具,致力于简化数据处理过程,它具有以下显著特点。

1. 易于使用

Kettle提供了直观的图形化用户界面,用户可以通过拖曳和连接组件来创建ETL,即使是没有编程经验的用户也能轻松上手。

2. 功能强大

Kettle提供了丰富的功能,包括数据过滤、排序、聚合、连接、拆分、格式化等。用户可以利用这些功能对数据进行转换和清洗,满足各种数据处理需求。

3. 多数据源支持

Kettle支持多种数据源的集成和处理,包括关系数据库(如MySQL、Oracle),文件(如CSV、Excel),非关系数据库(如MongoDB、HBase),大数据平台(如HDFS、Hive)等。用户可以轻松地连接和处理不同类型的数据源。

4. 可扩展性

Kettle提供了插件机制,允许用户根据自己的需求编写和集成插件,使其能够适应各种复杂的数据处理场景。

5. 跨平台支持

Kettle可在多个操作系统上运行,包括Windows、Linux和macOS等,使用户可以在不同的环境中使用Kettle。

6. 调度和监控功能

Kettle提供了调度和监控功能,允许用户自动执行数据处理任务并实时监控任务进展。用户可设置定时任务、依赖关系和警报机制,确保数据处理过程的可靠性和稳定性。

7. 高性能

Kettle支持并行执行数据处理任务,充分利用多核处理器和分布式计算能力,提升处理速度和性能,使用户能更高效地处理大规模数据。

3.2 Kettle的安装与启动

本教材使用的是Kettle 9.2版本,读者可以从Pentaho官方网站下载Kettle的安装包pdi-ce-9.2.0.0-290.zip。接下来,以Windows操作系统为例,讲解如何安装与启动Kettle。

1. 安装Kettle

Kettle是一款免安装的绿色软件,可以直接使用。因此,只需将Kettle的安装包pdi-

ce-9.2.0.0-290.zip 解压缩到计算机上的任意文件夹即可。解压完成后，将会生成一个名为 data-integration 的文件夹，其中包含了 Kettle 的所有必要文件和文件夹，如图 3-1 所示。

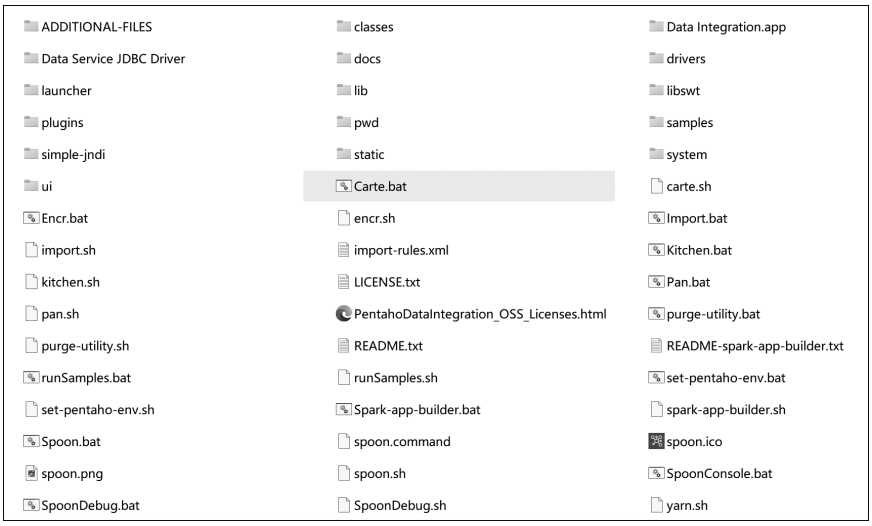

图 3-1　文件夹 data-integration 的内容

下面，针对 Kettle 一些核心的文件和文件夹进行介绍，具体内容如下。

- Kitchen.bat：用于启动 Kitchen 的批处理文件。
- Pan.bat：用于启动 Pan 的批处理文件。
- Spoon.bat：用于启动 Spoon 的批处理文件。
- Carte.bat：用于启动 Carte 的批处理文件。
- logs：用于存放日志文件的文件夹，这些文件记录了 Kettle 的运行日志和错误信息。logs 文件夹会在 Kettle 启动后自动创建。
- samples：用于存放示例文件的文件夹，这些文件提供了一些示例转换和作业，可供学习和参考。
- lib：用于存放 Kettle 所依赖的 JAR 文件的文件夹，这些文件包含了 Kettle 所需的 Java 库。
- plugins：用于存放插件的文件夹，这些插件可以扩展 Kettle 的功能。

2. 启动 Kettle

由于 Kettle 是基于 Java 虚拟机（JVM）运行的，所以在启动 Kettle 之前，需要确保 Windows 操作系统已经安装了 JDK 并正确配置了 Java 环境。本教材使用的 JDK 版本为 8，关于 JDK 的安装和 Java 环境的配置，这里不做详细说明。

由于 Spoon 提供了直观的图形化开发环境，用户能够通过可视化界面便捷地实现 ETL，所以本教材将主要使用 Spoon 讲解 Kettle 的使用。读者可以通过双击批处理文件 Spoon.bat 启动 Spoon，此时会进入 Spoon 的加载界面，如图 3-2 所示。

首次启动 Spoon 时，它需要加载并初始化一系列必要的配置文件和资源，以确保正常运行。这个过程可能会消耗一些时间。

Spoon 启动完成后，将进入 Kettle 图形化界面，该界面可以分为 4 个主要部分，分别是菜单栏、工具栏、浏览窗口和工作区，如图 3-3 所示。

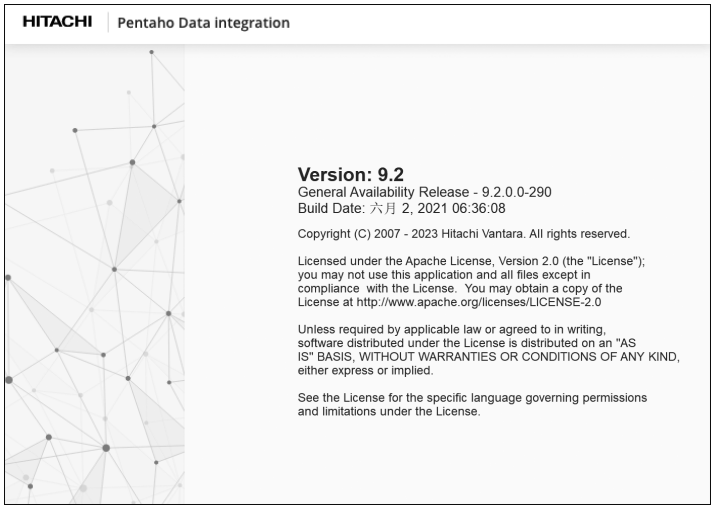

图 3-2 Spoon 的加载界面

图 3-3 Kettle 图形化界面

下面,针对图 3-3 中标注的菜单栏、工具栏、浏览窗口和工作区进行讲解,具体内容如下。

(1) 菜单栏。

菜单栏提供了对所有功能的访问,包括“文件”“编辑”“视图”“执行”“工具”“帮助”6 个选项,每个选项提供了不同类别的功能,例如“文件”选项提供了新建、打开、导出等功能。

(2) 工具栏。

工具栏提供了对常用功能的便捷访问,使用户在操作时更加方便。工具栏由多个按钮组成,每个按钮都有其特定的功能,具体介绍如下。

- “新建文件”按钮用于创建新的作业、转换或数据库连接。
- “打开文件”按钮用于打开已保存的转换或作业。

- “浏览存储器”按钮用于浏览存储库。存储库是一个用于存储和管理转换和作业的中心化数据库。
- “保存”按钮用于将当前转换或作业保存为文件。
- “另存为”按钮用于将当前转换或作业保存到其他文件中。
- “透视”按钮用于切换视图类型。
- Connect 按钮用于创建并连接存储库。

(3) 浏览窗口。

浏览窗口主要由“主对象树”“核心对象”两个选项卡组成,其中“主对象树”选项卡用于查看当前转换或作业的相关信息;“核心对象”选项卡提供了用于构建转换或作业的步骤和作业项。

(4) 工作区。

工作区是用于构建转换或作业的区域,初次启动 Spoon 时,会显示为欢迎界面。当用户新建转换或作业之后,系统将自动跳转到对应的工作区。在工作区中,用户可以自由地添加、配置或删除步骤或作业项。

3.3　Kettle 的转换和作业

在 Kettle 中,可以通过转换和作业来构建复杂的 ETL。具体而言,转换用于定义数据处理流程,包括数据的抽取、转换和加载。作业用于组织和调度转换的执行。作业可以包含一个或多个转换,并规定了这些转换之间的执行顺序和条件。本节将详细介绍 Kettle 中转换和作业。

3.3.1　转换

转换由一个或多个称为步骤(step)的组件构成,每个步骤都具有特定的功能,例如,从 CSV 文件中抽取数据、过滤数据、将数据输出到目标表等。在转换中,用户可以根据特定的顺序和逻辑来组织这些步骤。

转换中的步骤通过跳(hop)相互连接。跳定义了一个单向通道,允许数据从一个步骤流向另一个步骤。通过跳,用户可以定义数据的传递方式,例如,从一个步骤的输出流向另一个步骤的输入。这样,转换中的数据就可以按照用户定义的流程进行处理。

在转换中,跳的类型分为主跳(main hop)和错误跳(error hop),其中主跳表示数据的主要流向,通常用于连接前一个步骤的输出和后一个步骤的输入。主跳是转换中的主要数据路径,决定了数据的流动方向。而错误跳则用于处理错误数据,如不符合要求的数据。通过错误跳,可以将这些错误数据从当前步骤传递到特定的步骤进行处理。

假设有这样一个需求,需要从 CSV 文件中抽取数据,过滤掉不符合要求的数据,对符合要求的数据进行排序,并将排序结果输出到指定的表中。此时,转换的基本结构如图 3-4 所示。

在图 3-4 中,每个带有特定名称的图标都代表了一个步骤,每个步骤都具有特定的功能。例如,名为“CSV 文件输入”的步骤用于从 CSV 文件中抽取数据。步骤之间带有箭头的连接线表示跳,箭头的指向表示数据的流向,其中没有标记或者标记为的跳为主跳,而

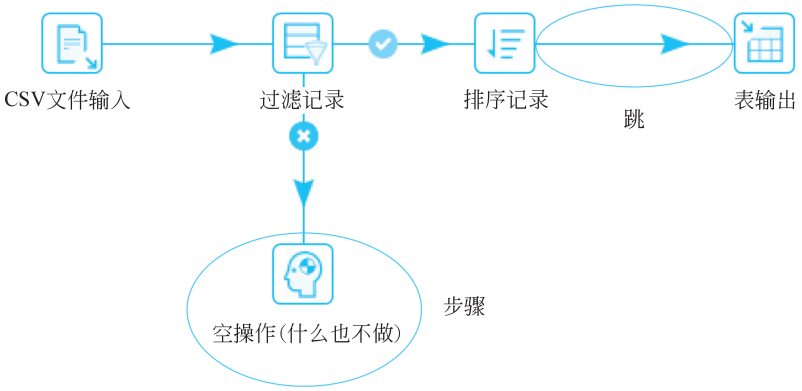

图 3-4 转换的基本结构

标记为✖的跳为错误跳。因此,可以看出名为“过滤记录”的步骤将符合要求的数据输出到名为“排序记录”的步骤,并将不符合要求的数据输出到名为“空操作(什么也不做)”的步骤中。

在转换中,一个步骤可以通过主跳连接至多个步骤,使数据可以同时输出到多个相连的步骤进行处理。转换中的步骤支持两种数据输出方式,即分发和复制。分发是指将数据拆分为多个副本,并将每个副本输出到不同的相连步骤进行独立处理。这意味着每个相连步骤将独立处理其接收到的数据副本。而复制指的是将数据的完整副本发送到每个相连步骤,所以每个步骤接收的都是完全一样的数据。

值得注意的是,在运行转换时,所有的步骤都会同时启动并以并行的方式运行。这意味着不同的步骤将在不同的线程中独立地执行,而它们之间的初始化顺序是不可预测的。这种不可预测性可能会对依赖于其他步骤初始化结果的步骤产生影响。如果一个步骤在初始化时需要依赖其他步骤的输出,而这些步骤尚未完成初始化,那么可能会导致错误或不完整的结果。因此,在设计 Kettle 的转换时,需要考虑步骤之间的依赖关系,并确保在执行步骤之前已经满足了所需的初始化条件,以避免潜在的问题。

多学一招:行集

行集(Row Set)是 Kettle 中用于在步骤之间传递数据的数据结构,它相当于跳在实际传输数据时的载体,用于存储和传输数据。

行集通过缓冲机制来平衡不同步骤之间的处理速度,使数据传递更加稳定和高效。举例来说,当一个步骤尝试向行集写入数据时,如果行集已经达到了其容量上限,那么这个步骤的写入操作就会被阻塞,直到行集中有足够的空间可供使用。同样,当一个步骤试图从行集读取数据时,如果行集为空,那么这个步骤也会被阻塞,直到行集中出现了新的可读取的数据。

行集内部包含多行数据,每行数据包含一个或多个字段,每个字段都有特定的数据类型。关于 Kettle 支持的数据类型如表 3-1 所示。

表 3-1 Kettle 支持的数据类型

数据类型	相关说明
String	以 UTF-8 编码的可变长度文本
Number	双精度浮点数值

续表

数 据 类 型	相 关 说 明
BigNumber	任意精度的数值
Integer	有符号的 64 位长整数
Internet Address	Internet 协议(IP)地址
Date	带有毫秒的日期时间
Boolean	取值为 true 或 false 的布尔值
Binary	包含任何类型二进制数据的字节数组
Timestamp	时间戳

3.3.2　作业

在 Kettle 中，作业由一系列作业项(job entry)组成，旨在实现 ETL 的自动化执行。作业中的每个作业项都具备特定的功能，例如，执行转换、发送邮件、条件判断等。通过组合不同的作业项，使它们可以按照预定义的顺序有序执行，以确保数据处理的准确性和顺序性。

在作业中，每个作业项通过作业跳(job hop)来连接。作业跳定义了作业项之间的依赖关系和执行顺序。作业跳分为无条件(unconditional)、当结果为真时遵循(follow when result is true)和当结果为假时遵循(follow when result is false)3 种类型，具体介绍如下。

1. 无条件

无条件的作业跳表示无论前一个作业项的执行结果如何，相连的两个作业项都会依次执行。换言之，无条件的作业跳没有对前一个作业项的执行结果进行额外的条件限制，它直接连接了两个作业项。

2. 当结果为真时遵循

当结果为真时遵循的作业跳表示只有当前一个作业项的执行结果为真时，相连的后一个作业项才会执行。这种类型的作业跳会根据前一个作业项的执行结果来决定是否执行下一个作业项。

3. 当结果为假时遵循

当结果为假时遵循的作业跳与当结果为真时遵循的作业跳正好相反。它表示只有当前一个作业项的执行结果为假时，相连的后一个作业项才会执行。同样，这种类型的作业跳也是根据前一个作业项的执行结果来决定是否执行下一个作业项。

假设有这样一个需求，要求每 60 秒检查一次指定的 CSV 文件是否存在，如果文件存在，则运行转换。如果在运行转换的过程中出现了错误，那么表示转换执行失败，此时需要向开发人员发送一封邮件来进行通知。如果转换成功了，那么就不会执行任何操作。此时，作业的基本结构如图 3-5 所示。

在图 3-5 中，每个带有特定名称的图标都代表了一个作业项，每个作业项都具有特定的功能。例如，名为 Start 的作业项用于运行作业，并配置作业的执行周期。每个作业项之间带有箭头的连接线表示作业跳，箭头的指向表示作业项的执行顺序，其中标记为  的作业跳为无条件，标记为  的作业跳为当结果为真时遵循，而标记为的作业跳为当结果为假时遵循。通过单击作业跳的标记可以修改其类型。

值得一提的是，在作业中，一个作业项可以有多个输出，这意味着可以通过作业跳将一

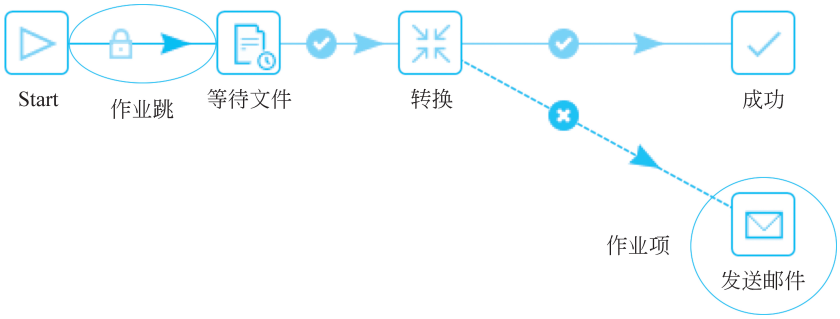

图 3-5 作业的基本结构

个作业项连接到多个其他的作业项。这种情况下,连接到同一作业项的多个作业项会以并行的方式执行,但它们与其他作业项的执行顺序仍然是保持一致的。也就是说,作业项之间的连接和依赖关系仍然会影响整个作业的执行流程,确保数据按照正确的顺序处理。

下面,通过一个简单的例子来介绍作业项具有多个输出的情况,具体如图 3-6 所示。

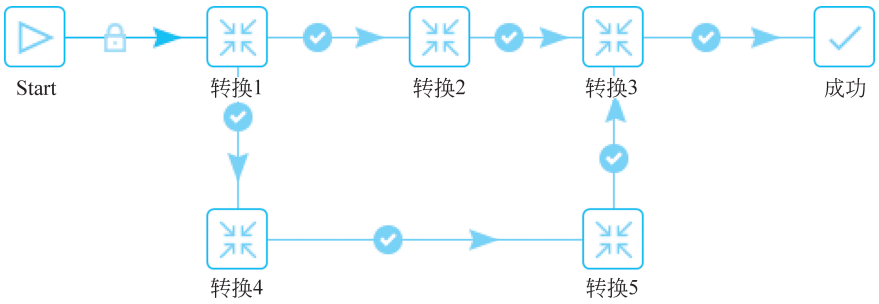

图 3-6 作业项具有多个输出的情况

从图 3-6 可以看出,名为“转换 1”的作业项有两个输出,分别是名为“转换 2”“转换 4”的作业项。当作业执行时,如果名为“转换 1”的作业项的执行结果为真,那么“转换 2”“转换 4”的作业项会并行执行。

多学一招:作业项的执行结果

在 Kettle 中,作业项的执行结果包含以下几部分内容。

- 执行状态:表示作业项的执行状态,如执行中、成功、失败等。
- 错误信息:如果作业项执行失败,执行结果会包含详细的错误信息,这些信息可以帮助我们诊断问题。
- 数据:作业项执行结果可能包含产生的数据,这取决于作业项类型和配置。例如,一个转换的作业项可以生成转换后的数据,供后续的作业项使用。
- 日志记录:作业项的执行结果会被记录到作业的日志中,这些日志信息包括执行时间、详细的执行过程等。这些日志信息有助于跟踪作业的执行过程,并对可能发生的问题进行排查。
- 其他元数据:根据具体的作业项,执行结果可能还会包含其他与作业项执行相关的元数据。这些元数据可能包括输入参数、输出参数、计算结果等。

3.4　Kettle 的基本操作

Kettle 的基本操作主要包括转换管理、作业管理和数据库连接，通过这些操作，用户可以灵活设计和实现 ETL。本节将介绍这些操作的实现方式。

3.4.1　转换管理

转换管理主要涉及创建、编辑、保存和运行转换，具体介绍如下。

1. 创建转换

在 Kettle 的图形化界面中，用户可以通过菜单栏、工具栏或快捷键创建转换，实现方式如下。

（1）通过菜单栏创建转换。首先，单击菜单栏中的“文件”选项。然后，在弹出的菜单中依次选择“新建”“转换”选项，这样就可以创建一个新的转换。

（2）通过工具栏创建转换。首先，单击工具栏中的“新建文件”按钮。然后，在弹出的菜单中选择“转换”选项，这样也可以创建一个新的转换。

（3）通过快捷键创建转换。按 Ctrl＋N 快捷键快速创建一个新的转换。

在 Kettle 的图形化界面中成功创建转换的效果如图 3-7 所示。

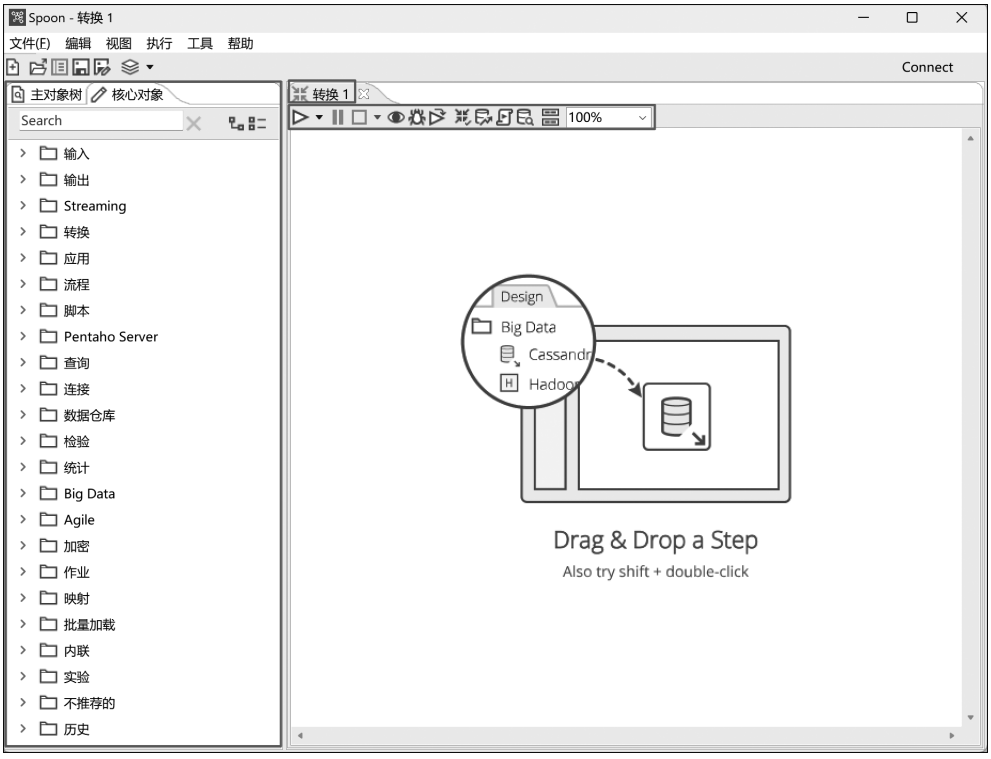

图 3-7　成功创建转换的效果

从图 3-7 中可以看出，转换创建成功后，其默认名称为“转换 1”。在“核心对象”选项卡中显示了所有分类对象，每个分类对象中包含了不同类型的步骤，可以通过单击每个分类对象前面的▶按钮查看其中的具体内容。

在转换的工作区上方存在一个工具栏，它为用户提供了对转换常见功能的快速访问，该

工具栏由多个按钮和一个“缩放比例”下拉框100%组成,其中“缩放比例”下拉框用于调整工作区中内容缩放的百分比。工具栏中不同按钮的功能介绍如下。

- “运行”按钮用于运行转换。在运行转换之前,需要先保存转换。
- “暂停”按钮用于暂停正在运行的转换。
- “停止”按钮用于停止正在运行的转换。
- “预览”按钮用于在预览模式下运行转换。在预览模式下,用户可以查看转换中选定步骤的执行效果,而不需要执行整个转换。
- “调试”按钮用于在调试模式下运行转换。在调试模式下,用户可以逐步运行转换,以便分析可能出现的错误。
- “重播”按钮用于重新运行转换。在重新运行转换之前,需要先保存转换。
- “验证”按钮用于检查转换是否可以正常运行。这样可以提前发现并解决可能导致转换运行失败的问题。
- “分析”按钮用于分析转换是否对数据库有影响。
- “SQL”按钮用于提取转换中涉及的 SQL 语句。
- “浏览数据库”按钮用于查看和管理数据库连接。
- “结果”按钮用于显示或隐藏转换的“执行结果”面板。

2. 编辑转换

编辑转换主要涉及添加步骤、连接步骤、配置步骤和添加注释的相关操作,具体介绍如下。

(1) 添加步骤。

步骤是转换的基本单元,根据实际需求,可以向转换中添加不同类型的步骤。Kettle 支持的步骤都可以在“核心对象”选项卡中找到。这些步骤按照功能进行分类,并归属于不同的分类对象,方便用户查找和使用。在日常生活中,分类可以帮助用户更方便地查找和使用需要的事物。通过分类,可以减少混乱和冗余,提高效率和效果,节省空间和时间,优化管理和组织。

接下来,对 Kettle 中常用的步骤进行介绍,具体如表 3-2 所示。

表 3-2 Kettle 常用的步骤

步 骤	分类对象	相 关 说 明
CSV 文件输入	输入	用于从 CSV 文件中抽取数据
Get data from XML		用于从 XML 文件中抽取数据
JSON input		用于从 JSON 文件中抽取数据
Excel 输入		用于从 Excel 文件中抽取数据
文本文件输入		用于从任意格式的文本文件中抽取数据
生成记录		用于生成特定行数的固定数据
生成随机数		用于生成不同类型的随机数据,包括随机数字、随机整数、随机字符串、UUID 等
自定义常量数据		用于生成一个自定义的数据
获取系统信息		用于获取 Kettle 运行环境的相关信息,包括 IP 地址、主机名、系统日期等
表输入		用于从数据库的表中抽取数据

续表

步　　骤	分类对象	相 关 说 明
JSON output	输出	用于将数据转换为 JSON 格式并加载到文本文件中
Excel 输出		用于将数据加载到 Excel 文件中
SQL 文件输出		用于将数据以一组 SQL 语句的形式加载到文本文件中
XML output		用于将数据加载到 XML 文件中
删除		用于根据指定规则删除数据库中表的数据
插入/更新		用于将数据加载到数据库的表中。适用于向表中插入数据或者更新表的数据
文本文件输出		用于将数据加载到文本文件中
表输出		用于将数据加载到数据库的表中。仅适用于向表中插入数据
Add a checksum	转换	用于计算指定字段值的校验和(checksum)
字符串替换		用于根据指定的规则,匹配指定字段的值,并将其替换为特定的值
值映射		用于将指定字段中的特定值替换为其他值
列拆分为多行		用于根据指定的分隔符,将指定字段的值拆分为多行
列转行		用于将指定字段的值拆分为多个字段
剪切字符串		用于根据索引范围,对指定字段(字符串类型)的值进行裁剪
去除重复记录		用于根据指定字段的值,删除重复的数据。使用该步骤之前,需要对数据进行排序处理
唯一行(哈希值)		用于根据指定字段的值,删除重复的数据
增加序列		用于添加一个自增字段
字段选择		用于选取指定字段
字符串操作		用于对指定字段(字符串类型)的值进行字符串操作
拆分字段		用于根据指定的分隔符,将指定字段的值拆分到多个字段中
排序记录		用于根据指定字段对数据进行排序处理
数值范围		用于判断指定字段(数字类型)的值是否在给定范围内,并根据判断结果为其添加标识
计算器		用于对指定字段的值进行计算
Concat fields		用于将多个字段的值通过指定分隔符进行合并
替换 NULL 值	应用	用于将数据中的 NULL 替换为指定值,或者将指定字段中的 NULL 替换为指定值
设置值为 NULL		用于将指定字段中的特定值替换为 NULL
Switch/case	流程	用于根据指定字段的值,将数据分配至不同的步骤,类似于 Java 语言中的 switch/case 语句
中止		用于根据指定条件停止转换的运行
空操作(什么也不做)		不进行任何操作
过滤记录		用于根据指定条件对数据进行过滤

续表

步　骤	分类对象	相关说明
Java 代码	脚本	用于编写 Java 代码处理数据
JavaScript 代码		用于编写 JavaScript 代码处理数据
公式		用于根据指定公式处理数据
执行 SQL 脚本		用于编写 SQL 语句处理数据库中的数据
正则表达式		用于根据正则表达式处理指定字段的值
数据库查询	查询	用于查询数据库中指定表的数据。基于表和行集中指定字段的值进行比较,实现查询
数据库连接		用于查询数据库中指定表的数据。基于指定 SQL 语句实现查询,可以使用行集中指定字段的值作为查询参数
HTTP client		用于根据指定 URL 从 Web 服务获取数据
HTTP post		用于根据指定 URL 向 Web 服务发送 post 请求并获取响应结果
检查文件是否存在		用于检查指定文件是否存在
检查表是否存在		用于检测数据库中的指定表是否存在
流查询		用于查询指定步骤中的数据
记录集连接	连接	用于连接两个步骤的数据,连接类型包括 INNER(内连接)、LEFT OUTER(左外连接)、RIGHT OUTER(右外连接)和 FULL OUTER(全外连接)
合并记录		用于合并两个步骤的数据。使用该步骤之前,需要对数据进行排序处理
排序合并		用于根据指定字段,对两个步骤的数据进行排序处理,然后将排序后的数据进行合并
记录关联(笛卡儿输出)		用于生成两个步骤中所有数据的组合,即笛卡儿积
数据检验	检验	用于根据指定规则检验数据
分组	统计	用于根据指定字段的值对数据进行分组,并允许对分组后的数据进行聚合运算
单变量统计		用于对指定字段的值执行统计操作,例如求最大值、最小值和中位数等
HBase input	Big Data	用于从 HBase 中的表抽取数据
HBase output		用于将数据加载到 HBase 的表中
Hadoop file input		用于从 HDFS 中的文件抽取数据
Hadoop file output		用于将数据加载到 HDFS 的文件中
MongoDB input		用于从 MongoDB 中的集合抽取数据
MongoDB output		用于将数据加载到 MongoDB 的集合中
设置变量	作业	用于设置变量,该变量无法在当前转换中使用
获取变量		用于获取指定变量

续表

步　　骤	分类对象	相 关 说 明
映射(子转换)	映射	用于在转换中添加子转换
映射输入规范		用于指定输入子转换的数据
映射输出规范		用于输出子转换的处理结果

在转换中添加步骤的方式分为两种，一种是通过拖曳的方式将指定步骤添加到工作区。另一种是通过鼠标双击指定步骤将其添加到工作区。例如，以拖曳的方式，向转换 1 添加“自定义常量数据”“增加序列”“文本文件输出”三个步骤后，转换 1 的工作区如图 3-8 所示。

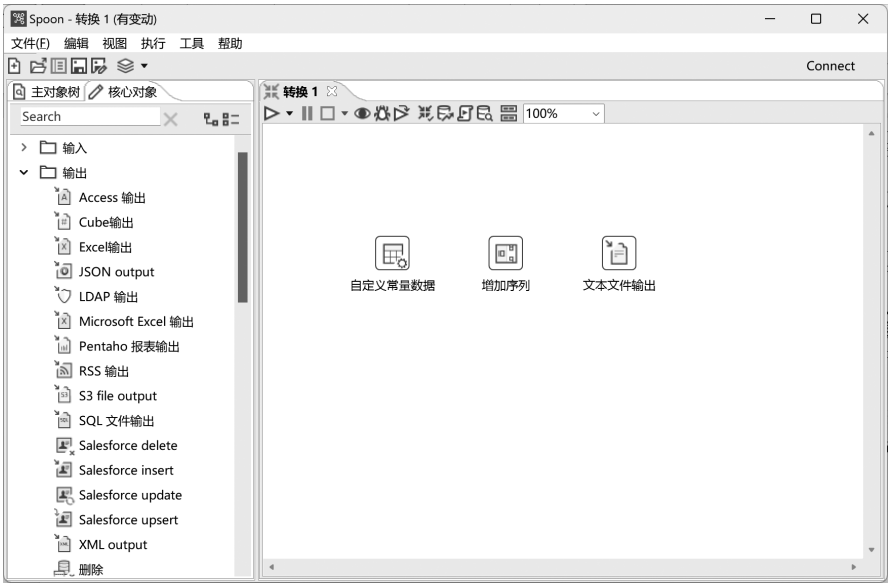

图 3-8　转换 1 的工作区

从图 3-8 中可以看出，转换 1 的工作区中成功添加了“自定义常量数据”“增加序列”“文本文件输出”步骤。

如果想要调整步骤的位置，可以通过拖曳相应的步骤在工作区中进行移动来实现。如果需要删除步骤，可以在选中相应步骤的情况下按下键盘上的 Delete 键即可。

(2) 连接步骤。

在 Kettle 中，将步骤添加到转换的工作区之后，还需要根据实际需求，通过跳来连接这些步骤。连接步骤的常用的方式有两种，具体介绍如下。

- 第一种方式：使用鼠标滚轮键单击输出数据的步骤，当移动鼠标出现带箭头的连接线时，再次使用鼠标滚轮键单击需要连接的步骤。如果在此过程中，想要取消连接步骤的操作，可以使用鼠标滚轮键单击工作区的空白区域。
- 第二种方式：按住键盘的 Shift 键，使用鼠标的左键单击输出数据的步骤，当移动鼠标出现带箭头的连接线时，释放 Shift 键，然后再次使用鼠标的左键单击需要连接的步骤。如果在此过程中，想要取消连接步骤的操作，可以使用鼠标左键单击工作区的空白区域。

接下来，使用跳将转换 1 的工作区中添加的 3 个步骤进行连接，使“自定义常量数据”步

骤的数据输出到“增加序列”步骤,以及使“增加序列”步骤的数据输出到“文本文件输出”步骤,如图 3-9 所示。

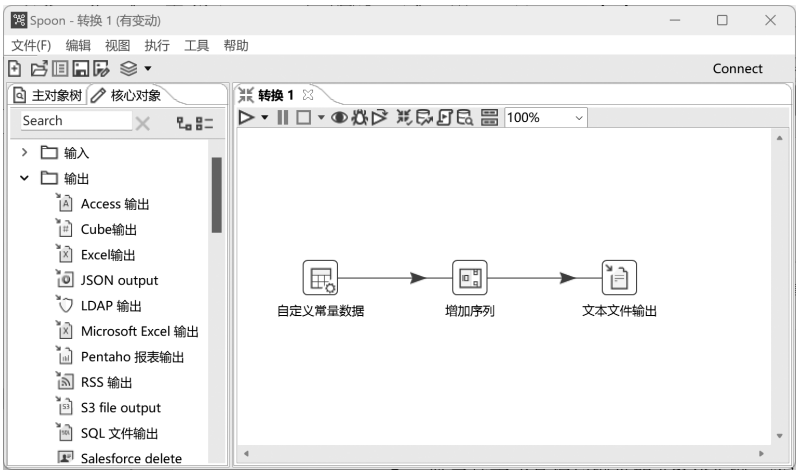

图 3-9 连接步骤

如果想要取消两个步骤之间的连接,那么可以通过鼠标右击相应的跳,然后在弹出的菜单中选择“删除节点连接”选项。

(3) 配置步骤。

在转换的工作区中添加的步骤默认并不具备任何功能,用户需要根据实际需求来配置这些步骤以实现特定的功能。在配置步骤时,用户可以在转换的工作区中双击需要配置的步骤,此时会打开一个窗口,用户可以在这个窗口中对步骤进行详细的配置。

由于每个步骤实现的功能各不相同,所以在配置不同的步骤时,打开的窗口也会有所差异。在这里,以“自定义常量数据”步骤为例,介绍如何配置步骤。至于其他步骤的配置方式,将在后续章节中陆续进行介绍。

在转换 1 的工作区中,双击“自定义常量数据”步骤打开“自定义常量数据”窗口,如图 3-10 所示。

图 3-10 “自定义常量数据”窗口(1)

从图 3-10 中可以看出,“自定义常量数据”窗口主要包含 4 个区域,具体介绍如下。

- 标注①的区域用于设置步骤的名称。同一转换中的每个步骤都需要有唯一的名称。
- 标注②的区域用于配置步骤,其中“元数据”选项卡用于指定自定义数据中字段的信

息，包括名称、类型、格式等；“数据”选项卡用于指定字段的值。

- 标注③的区域包含“确定”“预览”“取消”3 个按钮，其中“确定”按钮用于保存步骤的配置，“预览”按钮用于查看当前步骤中的数据，“取消”按钮用于取消步骤的配置。
- 标注④的区域是一个“帮助”按钮 Help，当用户单击该按钮时，浏览器会自动访问 Pentaho 官方网站中关于当前步骤的说明文档页面，通过该页面用户可以学习当前步骤的使用方式。

在图 3-10 中的“步骤名称”输入框内指定步骤的名称为“用户信息”。在“元数据”选项卡添加两行内容，其中第一行内容中“名称”列和“类型”列的值分别为 name 和 String，表示在自定义数据中添加数据类型为 String 的字段 name，第二行内容中“名称”列和“类型”列的值分别为 age 和 Integer，表示在自定义数据中添加数据类型为 Integer 的字段 age。

“自定义常量数据”窗口配置完成的效果如图 3-11 所示。

图 3-11　“自定义常量数据”窗口(2)

在图 3-11 中单击“数据”选项卡标签，在该选项卡中指定每个字段的数据，如图 3-12 所示。

从图 3-12 中可以看出，字段 name 和 age 都包含 3 行数据。在图 3-12 中单击“确定”按钮保存当前步骤的配置。

值得一提的是，在 Kettle 中，大部分步骤在配置时都可以将打开的窗口划分为图 3-10 所示的 4 部分内容，其中标注①、③和④的区域具有基本相同的功能，而标注②的区域会因步骤的不同而有所差异。

(4) 添加注释。

注释用于在转换中添加额外的描述信息，它可以帮助开发人员更好地了解转换的业务逻辑。在转换的工作区中添加注释时，可以通过鼠标右击工作区的空白区域，在弹出的菜单中选择“新建注释”选项，此时会打开“注释”窗口，如图 3-13 所示。

图 3-12　“数据”选项卡

图 3-13　“注释”窗口(1)

在图 3-13 中,Note 选项卡的 Note 文本框用于输入注释的内容。Font _style 选项卡用于配置注释的样式,包括字体、字号、文字颜色、背景颜色等。在完成注释内容和样式的配置后,通过单击“确定”按钮在转换的工作区中添加注释。

接下来,演示如何在转换 1 的工作区中添加注释,注释的内容为“用户信息包括姓名和年龄”,注释的样式为默认样式,如图 3-14 所示。

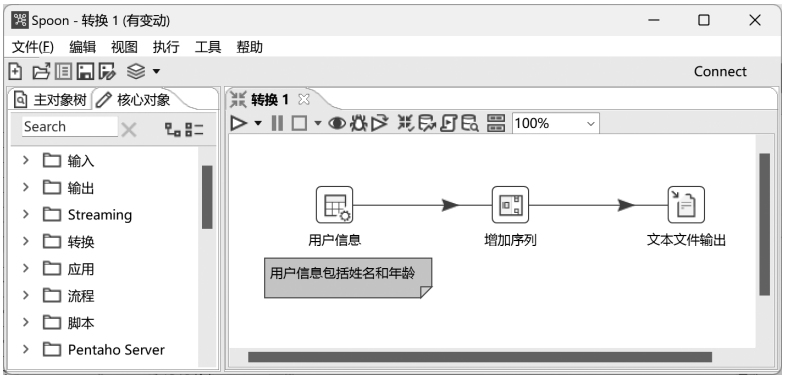

图 3-14 在转换 1 的工作区中添加注释

在图 3-14 中,注释的位置可以通过拖曳的方式进行调整。如果需要对注释的内容或样式进行修改,可以通过鼠标双击相应的注释。

3. 保存转换

编辑好的转换需要在保存后才能运行。在 Kettle 中,转换以文件的形式保存,该文件被称为转换文件,其扩展名为 ktr。在 Kettle 的图形化界面中,用户可以通过菜单栏、工具栏或快捷键来保存转换,实现方式如下。

(1) 通过菜单栏保存转换。首先,单击菜单栏中的“文件”选项。然后,在弹出的菜单中选择“保存”选项,这样就可以保存当前转换。

(2) 通过工具栏保存转换。单击工具栏中的“保存”按钮,这样也可以保存当前转换。

(3) 通过快捷键保存转换。按 Ctrl+S 快捷键快速保存当前转换。

在第一次保存转换时,Kettle 会弹出一个“另存为”对话框,该对话框用于指定转换文件的存储路径,以及定义转换文件的名称。转换文件的名称会直接作为转换的名称。

接下来,演示如何保存转换。在图 3-14 中,单击工具栏中的“保存”按钮,弹出“另存为”对话框,在该对话框内指定转换文件的名称为 transformation_demo,并且指定转换文件的存储路径为 D:\Data\KettleData\Chapter03,如图 3-15 所示。

在图 3-15 中,单击“保存”按钮返回转换 transformation_demo 工作区,如图 3-16 所示。

从图 3-16 中可以看出,转换 1 的名称已经变更为 transformation_demo。

4. 运行转换

在 Kettle 的图形化界面,用户可以在转换的工作区中,单击“工具栏”中的“运行”按钮运行当前转换,也可以直接按键盘的 F9 键运行当前转换。当用户通过这两种方式运行转换时,转换并不会立即启动,而是打开“执行转换”窗口,如图 3-17 所示。

在图 3-17 中,单击“启动”按钮即可运行当前的转换。此外,用户还可以配置当前转换的日志级别、命名参数、变量等内容。但是,通常情况下,这些配置可以保持默认设置。

接下来,演示如何运行转换 transformation_demo。不过在此之前,需要完成转换

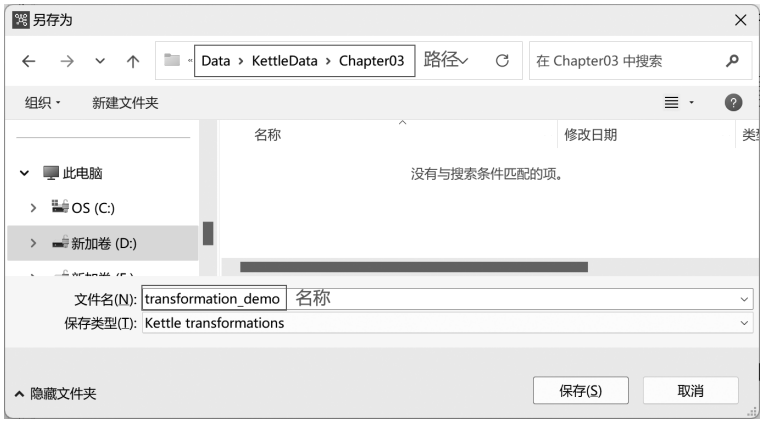

图 3-15　“另存为”对话框(1)

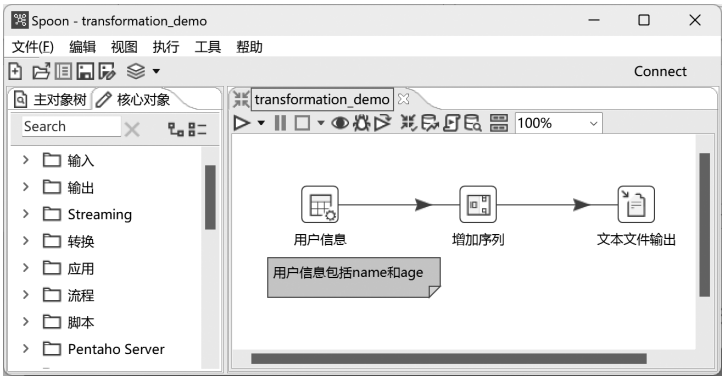

图 3-16　转换 transformation_demo 的工作区(1)

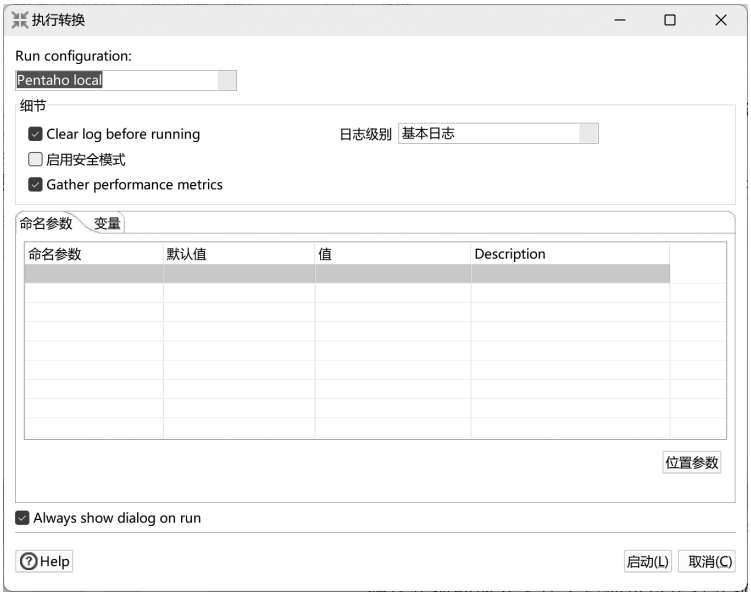

图 3-17　“执行转换”窗口

transformation_demo 中“增加序列”和“文本文件输出”步骤的配置，以实现为自定义的数据增加自增字段，并将数据加载到文本文件的功能，具体操作步骤如下。

(1) 配置“增加序列”步骤。

在图 3-16 中,双击“增加序列”步骤,打开“增加序列”窗口。在该窗口的“步骤名称”输入框中指定步骤的名称为“添加用户编号”。在“值的名称”输入框中指定自增字段的字段名为 id。

“增加序列”窗口配置完成的效果如图 3-18 所示。

图 3-18 “增加序列”窗口

从图 3-18 中可以看出,自增字段 id 的值从 1 开始每次递增 1。在图 3-18 中,单击“确定”按钮保存当前步骤的配置。

(2) 配置“文本文件输出”步骤。

在图 3-16 中,双击“文本文件输出”步骤,打开“文本文件输出”窗口。在该窗口的“步骤名称”输入框中指定步骤名称为“加载用户信息至文件”,如图 3-19 所示。

图 3-19 “文本文件输出”窗口(1)

在图 3-19 中,单击“浏览”按钮打开 Save To 窗口,在该窗口中通过单击 Local 折叠框选择本地文件系统的路径为 D:\Data\KettleData\Chapter03,并分别在 File name 输入框和 File type 下拉框中指定文件的名称和类型为 user 和 *.txt,如图 3-20 所示。

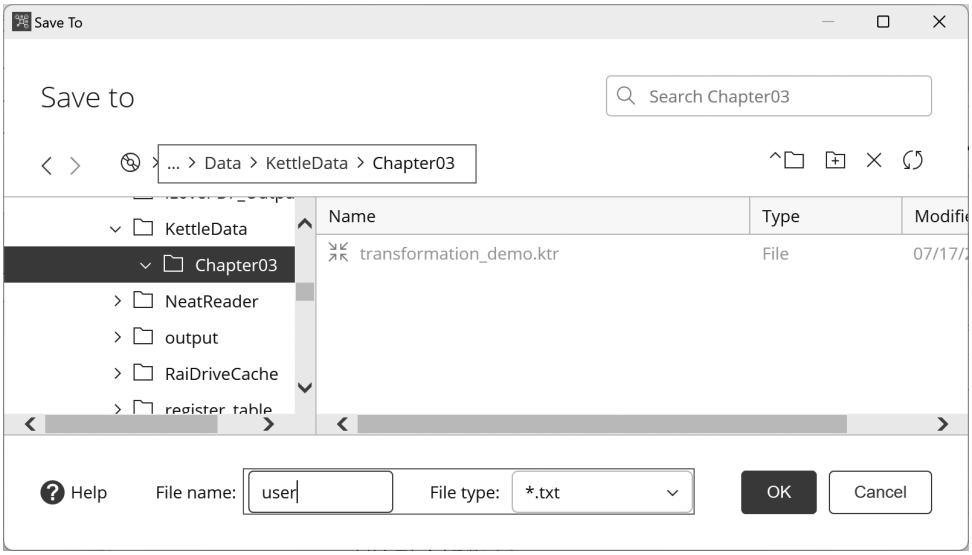

图 3-20　Save To 窗口

图 3-20 中配置的内容表示，将数据加载到本地文件系统中的 D:\Data\KettleData\Chapter03 路径下的 user.txt 文件中。如果需要将数据加载到虚拟文件系统或者 HDFS 文件系统的文件中，那么可以通过在 Save To 窗口中通过单击 VFS Connections 折叠框或 Hadoop Clusters 折叠框来选择路径。

在图 3-20 中，单击 OK 按钮返回至“文本文件输出”窗口，如图 3-21 所示。

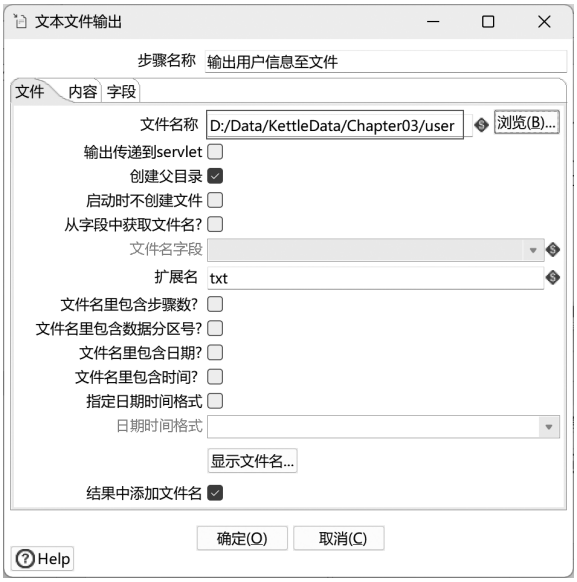

图 3-21　“文本文件输出”窗口(2)

在图 3-21 中，单击“确定”按钮保存当前步骤的配置，并返回至转换 transformation_demo 的工作区，如图 3-22 所示。

(3) 运行转换 transformation_demo。

保存并运行转换 transformation_demo，其运行完成后的效果如图 3-23 所示。

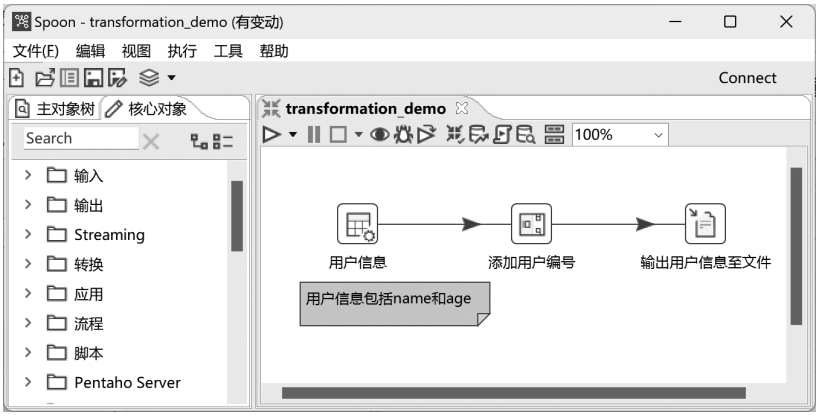

图 3-22 转换 transformation_demo 的工作区(2)

图 3-23 转换 transformation_demo 运行完成后的效果

从图 3-23 可以看出,每个步骤的右上角都新增了一个图标,这表明相应的步骤已经成功运行。因此可以说明,转换 transformation_demo 运行成功。如果在步骤的右上角新增了一个图标,那么表明相应的步骤在运行过程中出现了错误。

此时,可以在本地文件系统的 D:\Data\KettleData\Chapter03 路径下,查看文件 user.txt 的内容,如图 3-24 所示。

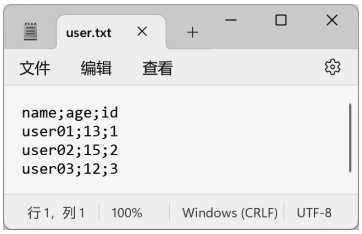

图 3-24 文件 user.txt 的内容

从图 3-24 中可以看出,文件 user.txt 的首行包含各个字段的名称,且每个字段之间以分号分隔。这是因为“文本文件输出”步骤在向文本文件加载数据时,默认使用分号来分隔每个字段,并将字段名称作为首行数据。不过,用户可以通过“文本文件输出”步骤的“内容”选项卡和“字段”选项卡来调整加载数据的格式和内容。关于