# ETL工具功能

## 连接

ETL工具能连接到类型广泛的数据源和数据格式。

1. 连接到普通关系型数据库并获取数据
2. 从有分隔符或固定格式的ASCII文件中获取数据
3. 从xml文件中获取数据
4. 从流行的办公软件中获取数据，如ACCESS数据库和Excel电子表格
5. 使用FTP/SFTP/SSH方式获取数据
6. 能从WEB Service或RSS中获取数据，ERP系统里的数据

## 平台独立

可以在任何平台或不同平台的组合上运行

## 数据规模

并发、分区、集群

能够处理多个数据流，利用多核的硬件架构；使用特定的分区模式，将数据分发到并发的数据流中；能够分配在多台机器上联合完成。

## 设计灵活性

ETL工具分为基于过程和基于映射的。基于映射的工具只在源和目的数据之间提供一组固定的步骤，严重限制了设计工作的自由度。基于过程的工具，根据实际的数据和业务需求，可以创建自定义的步骤和转换。

## 复用性

设计完成的ETL转换可以被复用。一个转换或步骤，从其他地方可以调用这些转换或步骤。

## 扩展性

几乎所有的ETL工具都提供脚本，以编程的方式来解决工具本身不能解决的问题。另外有少数的ETL工具可以通过API或其他方式来为工具增加组件。第三种方法是使用脚本语言写函数，函数可以被其他转换或脚本调用。

## 数据转换

ETL很大一部分工作都是在做数据转换。在输入和输出之间，数据都要经过检验、连接、分割、合并、转置、排序、合并、克隆、重排、过滤、删除、替换、或其他操作。

1. 缓慢变更维度
2. 查询值
3. 行列转换
4. 条件分割
5. 排序、合并、连接
6. 聚集

## 测试和调试

1. ETL过程没有按时完成数据转换的任务怎么办
2. 如果转换过程异常终止怎么办
3. 目标是非空列的数据抽取到数据为空怎么办
4. 转换后的行数和抽取到的数据行数不一致怎么办
5. 转换后计算的数值和另一个系统的数值不一致怎么办

白盒测试和和黑盒测试

## 血统和影响分析

任何ETL工具都有一个重要的功能：读取源数据，提取不同转换构成的数据流的信息。

血统分析：回溯机制，可以查看数据的来源，如“价格”“数量”字段作为输入字段，在转换中根据这两个字段计算出“收入”字段。即使后面的处理流程过滤了“价格”和“数量”字段，血统分析功能也能分析出“收入”字段是基于“价格”和“数量”字段的。

影响分析：该方法可以分析源数据字段对随后的转换以及目标表的影响

## 日志和审计

日志：记录在转换过程中执行了哪些步骤，包括步骤开始和结束的时间戳

审计：可以追溯到对数据做的所有操作，包括读行数、转行行数、写行数。

# KETTLE基本概念

## 避免自定义开发

通过手工写java代码或java脚本来实现一些功能。但增加的每一行代码都给项目增加了复杂度和维护成本。（脚本会将程序中的一些变量固定下来，不利于产品化）

## 字段类型

String:字符类型数据

Number:双精度浮点数

Integer:带符号长整型（64位）

BigNumber:任意精度数值

Date:带毫秒精度的日期时间值

Boolean:取值为TRUE或FALSE的布尔值

Binary:二进制字段可以包括图形、声音、视频及其他类型的二进制数据

## 关系型数据库的力量

关系数据库是一种高级的软件，它在数据的连接、合并、排序等方面有着突出优势。与基于流的数据处理引擎相比，数据库使用的数据都存储在磁盘中。当关系型数据库进行连接或者排序操作时，只要使用这些数据的引用即可，而不用把数据加载到内存里，这就体现出明显的性能优势。缺点：把数据加载到关系型数据库里也会产生性能的瓶颈。

备注：当使用KETTLE的join的时候，需要排序，可以先在数据库中排序之后直接进行join，避免再使用KETTLE自身的排序功能。简化组件，同时提高运行效率

## 连接和事务

数据库连接只在执行作业或转换时使用。在作业里，每个作业项都打开和关闭一个独立的数据库连接。转换也是如此，因为转换中的步骤是并行执行的，每个步骤都打开一个独立的数据库连接并开始一个事务。优点：提高性能。缺点：当不同步骤更新同一个表时，也会带来锁和参照完整性问题。