**ETL**

## 一、基础知识

**1 ETL是将业务系统的数据经过抽取、清洗转换之后加载到数据仓库的过程，目的是将企业中的分散、零乱、标准不统一的数据整合到一起，为企业的决策提供分析依据**。 ETL是BI项目重要的一个环节。 通常情况下，在BI项目中ETL会花掉整个项目至少1/3的时间,ETL设计的好坏直接关接到BI项目的成败。

**2 ETL的设计分三部分：数据抽取、数据的清洗转换、数据的加载。**在设计ETL的时候我们也是从这三部分出发。数据的抽取是从各个不同的数据源抽取到ODS(Operational Data Store，操作型数据存储)中——这个过程也可以做一些数据的清洗和转换)，在抽取的过程中需要挑选不同的抽取方法，尽可能的提高ETL的运行效率。ETL三个部分中，花费时间最长的是“T”(Transform，清洗、转换)的部分，一般情况下这部分工作量是整个ETL的2/3。数据的加载一般在数据清洗完了之后直接写入DW(Data Warehousing，数据仓库)中去。

**3 ETL的实现有多种方法，常用的有三种**。一种是借助ETL工具(如Oracle的OWB、SQL Server 2000的DTS、SQL Server2005的SSIS服务、Informatic等)实现，一种是SQL方式实现，另外一种是ETL工具和SQL相结合。前两种方法各有各的优缺点，借助工具可以快速的建立起ETL工程，屏蔽了复杂的编码任务，提高了速度，降低了难度，但是缺少灵活性。SQL的方法优点是灵活，提高ETL运行效率，但是编码复杂，对技术要求比较高。第三种是综合了前面二种的优点，会极大地提高ETL的开发速度和效率。

## 二、 数据的抽取（Extract）

**1 这一部分需要在调研阶段做大量的工作**，首先要搞清楚数据是从几个业务系统中来,各个业务系统的数据库服务器运行什么DBMS,是否存在手工数据，手工数据量有多大，是否存在非结构化的数据等等，当收集完这些信息之后才可以进行数据抽取的设计。

**2 对于与存放DW的数据库系统相同的数据源处理方法**

这一类数据源在设计上比较容易。一般情况下，DBMS(SQLServer、Oracle)都会提供数据库链接功能，在DW数据库服务器和原业务系统之间建立直接的链接关系就可以写Select 语句直接访问。

**3 对于与DW数据库系统不同的数据源的处理方法**

对于这一类数据源，一般情况下也可以通过ODBC的方式建立数据库链接——如SQL Server和Oracle之间。如果不能建立数据库链接，可以有两种方式完成，一种是通过工具将源数据导出成.txt或者是.xls文件，然后再将这些源系统文件导入到ODS中。另外一种方法是通过程序接口来完成。

**4 对于文件类型数据源(.txt,.xls)**，可以培训业务人员利用数据库工具将这些数据导入到指定的数据库，然后从指定的数据库中抽取。或者还可以借助工具实现**。**

**5 增量更新的问题**

对于数据量大的系统，必须考虑增量抽取。一般情况下，业务系统会记录业务发生的时间，我们可以用来做增量的标志,每次抽取之前首先判断ODS中记录最大的时间，然后根据这个时间去业务系统取大于这个时间所有的记录。利用业务系统的时间戳，一般情况下，业务系统没有或者部分有时间戳。

## 三、数据的清洗转换（Cleaning、Transform）

一般情况下，数据仓库分为ODS、DW两部分。通常的做法是从业务系统到ODS做清洗，将脏数据和不完整数据过滤掉，在从ODS到DW的过程中转换，进行一些业务规则的计算和聚合。

**1 数据清洗**

数据清洗的任务是过滤那些不符合要求的数据，将过滤的结果交给业务主管部门，确认是否过滤掉还是由业务单位修正之后再进行抽取。

不符合要求的数据主要是有不完整的数据、错误的数据、重复的数据三大类。

**(1)不完整的数据：**这一类数据主要是一些应该有的信息缺失，如供应商的名称、分公司的名称、客户的区域信息缺失、业务系统中主表与明细表不能匹配等。对于这一类数据过滤出来，按缺失的内容分别写入不同Excel文件向客户提交，要求在规定的时间内补全。补全后才写入数据仓库。

**(2)错误的数据：**这一类错误产生的原因是业务系统不够健全，在接收输入后没有进行判断直接写入后台数据库造成的，比如数值数据输成全角数字字符、字符串数据后面有一个回车操作、日期格式不正确、日期越界等。这一类数据也要分类，对于类似于全角字符、数据前后有不可见字符的问题，只能通过写SQL语句的方式找出来，然后要求客户在业务系统修正之后抽取。日期格式不正确的或者是日期越界的这一类错误会导致ETL运行失败，这一类错误需要去业务系统数据库用SQL的方式挑出来，交给业务主管部门要求限期修正，修正之后再抽取。

**(3)重复的数据：**对于这一类数据——特别是维表中会出现这种情况——将重复数据记录的所有字段导出来，让客户确认并整理。

数据清洗是一个反复的过程，不可能在几天内完成，只有不断的发现问题，解决问题。对于是否过滤，是否修正一般要求客户确认，对于过滤掉的数据，写入Excel文件或者将过滤数据写入数据表，在ETL开发的初期可以每天向业务单位发送过滤数据的邮件，促使他们尽快地修正错误,同时也可以做为将来验证数据的依据。数据清洗需要注意的是不要将有用的数据过滤掉，对于每个过滤规则认真进行验证，并要用户确认。

**2 数据转换**

数据转换的任务主要进行不一致的数据转换、数据粒度的转换，以及一些商务规则的计算。

**(1)不一致数据转换：**这个过程是一个整合的过程，将不同业务系统的相同类型的数据统一，比如同一个供应商在结算系统的编码是XX0001,而在CRM中编码是YY0001，这样在抽取过来之后统一转换成一个编码。

**(2)数据粒度的转换：**业务系统一般存储非常明细的数据，而数据仓库中数据是用来分析的，不需要非常明细的数据。一般情况下，会将业务系统数据按照数据仓库粒度进行聚合。

**(3)商务规则的计算：**不同的企业有不同的业务规则、不同的数据指标，这些指标有的时候不是简单的加加减减就能完成，这个时候需要在ETL中将这些数据指标计算好了之后存储在数据仓库中，以供分析使用。

## 三、ETL日志、警告发送

**1、 ETL日志**

ETL日志分为三类。

**（1）执行过程日志**，这一部分日志是在ETL执行过程中每执行一步的记录，记录每次运行每一步骤的起始时间，影响了多少行数据，流水账形式。

**（2）错误日志**，当某个模块出错的时候写错误日志，记录每次出错的时间、出错的模块以及出错的信息等。

**（3）日志是总体日志**，只记录ETL开始时间、结束时间是否成功信息。如果使用ETL工具,ETL工具会自动产生一些日志，这一类日志也可以作为ETL日志的一部分。

记录日志的目的是随时可以知道ETL运行情况，如果出错了，可以知道哪里出错。

**2、 警告发送**

如果ETL出错了，不仅要形成ETL出错日志，而且要向系统管理员发送警告。发送警告的方式多种，一般常用的就是给系统管理员发送邮件，并附上出错的信息，方便管理员排查错误。ETL是BI项目的关键部分，也是一个长期的过程，只有不断的发现问题并解决问题，才能使ETL运行效率更高，为BI项目后期开发提供准确与高效的数据。

**3 后记**

做数据仓库系统，ETL是关键的一环。说大了，ETL是数据整合解决方案，说小了，就是倒数据的工具。回忆一下工作这么长时间以来，处理数据迁移、转换的工作倒还真的不少。但是那些工作基本上是一次性工作或者很小数据量。可是在数据仓库系统中，ETL上升到了一定的理论高度，和原来小打小闹的工具使用不同了。究竟什么不同，从名字上就可以看到，人家已经将倒数据的过程分成3个步骤，E、T、L分别代表抽取、转换和装载。其实ETL过程就是数据流动的过程，从不同的数据源流向不同的目标数据。但在数据仓库中，

**4 ETL特点**

**（1）数据同步，它不是一次性倒完数据就拉到**，它是经常性的活动，按照固定周期运行的，甚至现在还有人提出了实时ETL的概念。

**（2）数据量，一般都是巨大的，**值得你将数据流动的过程拆分成E、T和L。

## 四、深入说明

现在有很多成熟的工具提供ETL功能，且不说他们的好坏。从应用角度来说，ETL的过程其实不是非常复杂，这些工具给数据仓库工程带来和很大的便利性，特别是开发的便利和维护的便利。但另一方面，开发人员容易迷失在这些工具中。举个例子，VB是一种非常简单的语言并且也是非常易用的编程工具，上手特别快，但是真正VB的高手有多少？微软设计的产品通常有个原则是“将使用者当作傻瓜”，在这个原则下，微软的东西确实非常好用，但是对于开发者，如果你自己也将自己当作傻瓜，那就真的傻了。ETL工具也是一样，这些工具为我们提供图形化界面，让我们将主要的精力放在规则上，以期提高开发效率。从使用效果来说，确实使用这些工具能够非常快速地构建一个job来处理某个数据，不过从整体来看，并不见得他的整体效率会高多少。问题主要不是出在工具上，而是在设计、开发人员上。他们迷失在工具中，没有去探求ETL的本质。可以说这些工具应用了这么长时间，在这么多项目、环境中应用，它必然有它成功之处，它必定体现了ETL的本质。如果我们不透过表面这些工具的简单使用去看它背后蕴涵的思想，最终我们作出来的东西也就是一个个独立的job，将他们整合起来仍然有巨大的工作量。大家都知道“理论与实践相结合”，如果在一个领域有所超越，必须要在理论水平上达到一定的高度.

## 五、ETL工具比较

**1 Informatica VS SSIS**  
（1）从易用性来说，Informaitca有四个开发管理组件，初学者可能不知道如 何下手，所以关键先要了解PowerCenter的开发流程。SSIS有BIDS和SSMS，主要的开发工作在BIDS上完成，但是BIDS是基于 Visaul Studio的，如果是没有做过.NET开发的人员，VS的复杂的开发环境可能让他们不知所措，但是对于熟悉VS的开发人员来说，通过VS，他们可以使用 更多的资源，比如源管理器，VSTFS等。从界面的人性化上看，SSIS无疑更好，这是Microsoft的一贯特色，PowerCenter有些地方连 右键菜单都没有。  
（2）SSIS的数据流如同PowerCenter的Mapping，SSIS的控制流如同PowerCenter的Task，SSIS的 Package如同PowerCenter的WorkFlow。PowerCenter的Mapping和WorkFlow分开开发，在一个大型开发团队 中，开发人员开发Mapping，而有管理人员将多个Mapping在WorkFlow中统一调度。但如果是只有几个人的开发团队中，这种模式的效率就不 如SSIS了，因为小型团队中每个人都独立工作，开发人员需要开发好Mapping后，再在WorkFlow建立Session，需要再一次为 Session配置连接和参数。相反，SSIS不适合多人开发，因为不像PowerCenter将Mapping和WorkFlow分开开发，所以当一个 人在开发Package时，其他人就插不上手。  
（3）SSIS的运行监控是非常棒的设计，可以看到数据流的执行情况。如果数据流出问题，可以马上知道在那个转换组件上发生了问题。而 PowerCenter的Monitor只能看到源和目标的情况，一旦出状况，就只能看日志了，但在日志大量的文本里，很多时候是找不出问题的所在。除了 执行时监控外，SSIS在开发时可以动态的Validate，而且它的Validate也严格得多。只要发现字段类型对应不上就会报错。 PowerCenter中Port的对应比较自由，比如一个Output的Port可以随意对应到多个Input的Port，而SSIS中一定要用多播组 件才行。所以在数据质量方面，SSIS比PowerCenter更有保障，PowerCenter最郁闷的是，报错后却找不到出错的地方。  
（4）就组件的功能上，SSIS比PowerCenter要丰富，像SSIS控制流的循环组件，数据流中的模糊查找、词频计算、数据挖掘评分等组件都 是Informaitca没有的。特别是有很多针对SQL Server组件。如果使用非SQL Server数据库的话，SSIS功能上要打个七折。  
（5）SSIS最致命的问题是数据库连接的兼容性上，数据流中很多组件只认OLE DB连接，而且只有OLE DB数据源组件才可以加参数，这显然是Microsoft有意为之。加上SSIS的Validate比较严格，所以连接到那些对OLE DB兼容不好的数据库就会报错，像我之前连接Oracle 9i就碰到这样的问题。PowerCenter支持更多的数据源，此外还可以连接SAP, PeopleSoft等商业软件，在大型企业中，这些功能是非常方便的。  
  
  
**2 Informatica VS Kettle**  
（1）Informatica属于商业软件，而Kettle是开源软件；Informatica的收费方式是软件一次性购买，但技术支持按项目收费。具体怎么界定项目不清楚。  
（2）风险和成本方面自然就不用说了，从某些方面来说的话，风险总是与成本成反比的；  
（3）易用性方面，Informatica和Kettle都有GUI图形界面，操作步骤都比较简单易用；主要取决于开发人员的对工具的熟悉情况，没有太大的可比性；  
（4）Kettle是基于Java开发的ETL工具，在使用过程中需要借助JVM，在数据抽取速度以及大数据处理能力方面远远不如Informatica；（Kettle在异构数据库的处理能力比同构数据库的处理能力要弱很多，特别是夸服务器之间就更弱了）  
（5） Informatica将客户端和服务端进行分离，而Kettle则没有但需要基于JVM；因此Kettle的部署比较简易；  
（6）稳定性方面，Informatica也有较大的优势；  
（7）Informatica将元数据及一些配置信息存储于配置的数据库，也就是说Informatica需要借助其他数据库才能运行；  
（8） Kettle的数据转换过程相对比较灵活，可以是手工编写的SQL语句、Java代码、正则表达式等；Informatica则由Data Quality组件来保证数据的质量，当然也支持SQL语句，两者没有明显的差异；  
（9）Informatica和Kettle均有监控日志功能，这也是ETL工具必须具备的，Informatica的监控日志相对要详细一些，但在实际应用中，大多数情况下不需要这么详细；  
（10）数据源方面，二者没有明显区别，Kettle支持广泛的数据库、数据文件等，同时支持扩展；Informatica广泛支持各类结构化、非结构化的数据源；

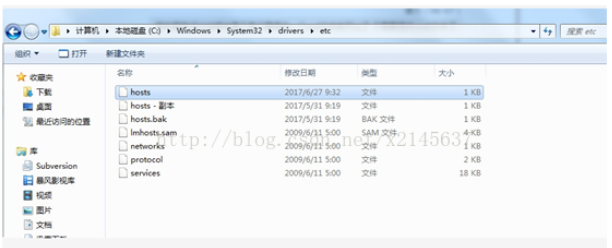
**Informatica 实战**

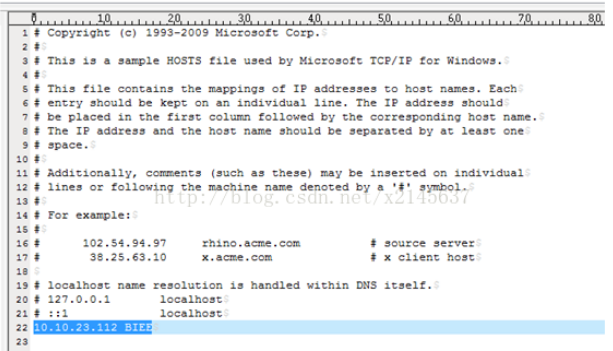
## 一 安装配置

**1 安装好Informatica**

首先，需配置一下修改配置文件路径为

地址：C:\Windows\System32\drivers\etc





**2将“10.10.23.112 BIEE”复制到后面。**

(注：第二个ip地址哪里，要在hots文件加一个配置

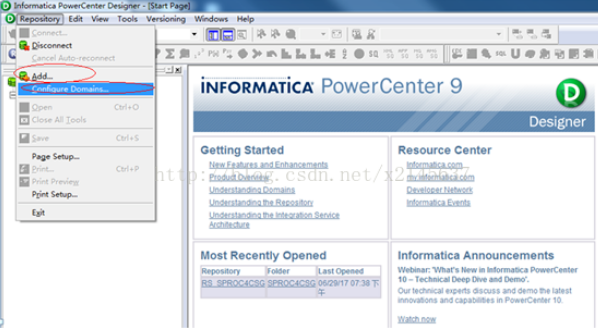
## 二 创建域

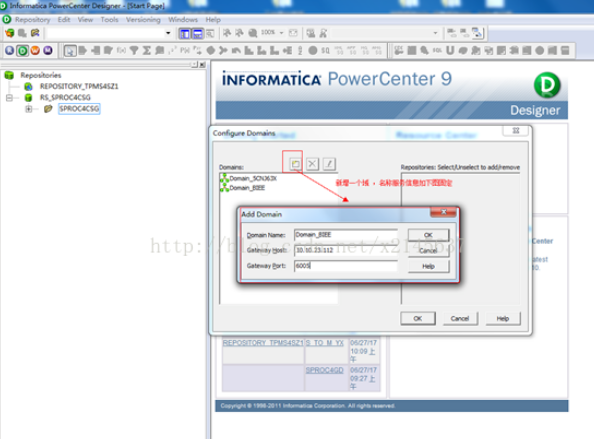
**1 新建一个域**

名称：Domain\_BIEE

地址：10.10.23.112

端口：6005

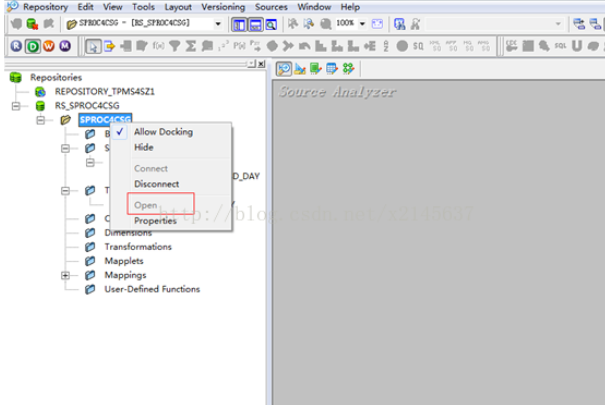




**2 登录的账号密码为SPROC4CSG/SPROC4CSG**

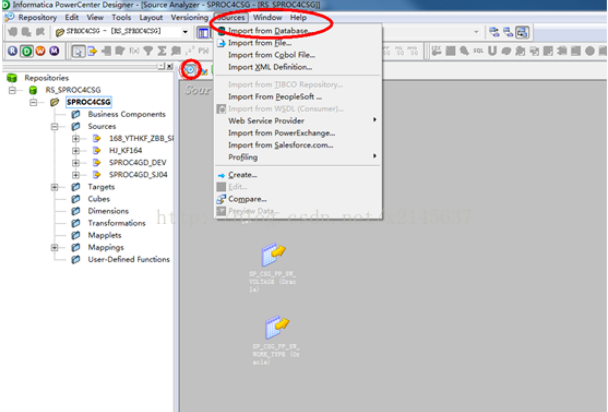
## 三 配置数据源、添加表结构

**1 打开项目**

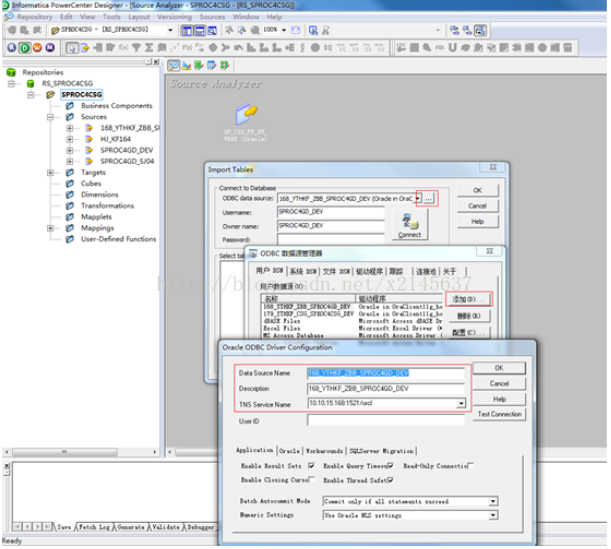


1. **配置源数据驱动及从源数据表、目标数据表中添加源数据表**

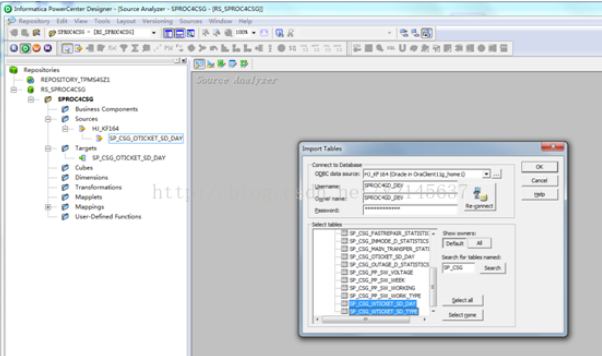
**3 首先选择源视图窗口(SourceAnalyzer)**



**4 配置数据源驱动程序(源和目标数据驱动都配置一个)**



**5 添加源表和目标表两数据源，需要安装32位ORACLE客户端驱动，需要两个相同的驱动然后如下图**

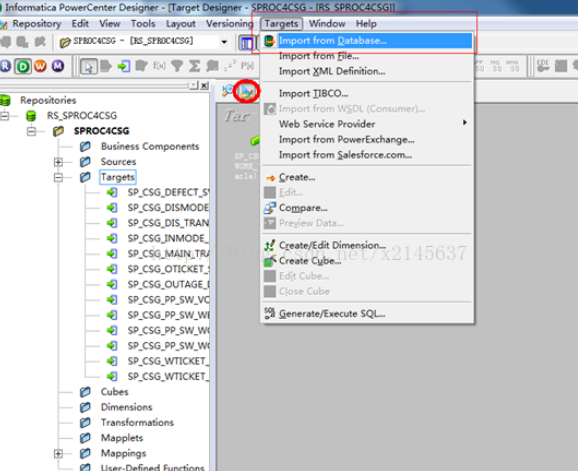


注：可输入表前缀查询自己需要导入的表。例如：SP\_CSG

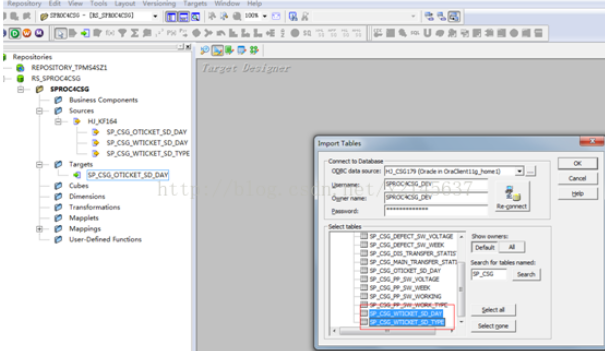
添加原表

**6添加目标数据表结构**

首先选择目标视图窗口(Target Desinger)

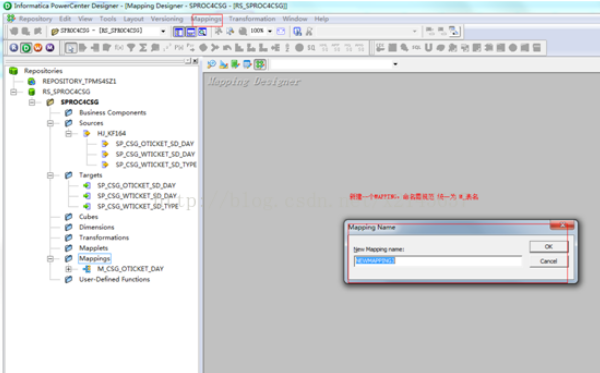


**7 添加目标表结构操作同源表操作类似。注：可通过Search进行过滤相应表**



## 四 创建Mappings

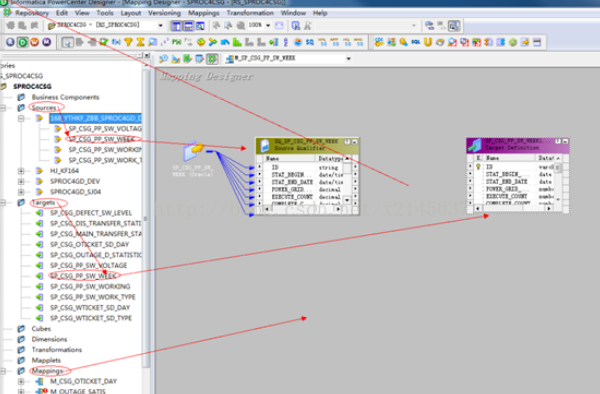
**1 创建一个映射MAPING**



注意切换到Map视图窗口.Mappings的命名：M\_表名

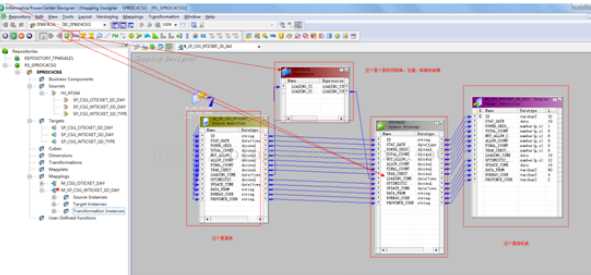
**2 建立源数据与目标数据的关系**

从Sources(源数据)中把需要的源数据表拖进来。从Targets(目标数据)中把需要转换后的数据表拖进来。建立源数据表与目标数据表之间的关联。

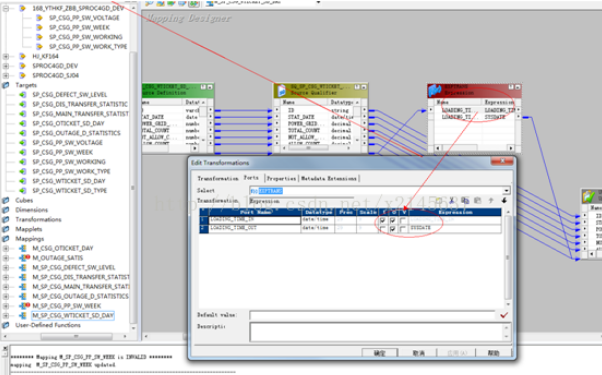


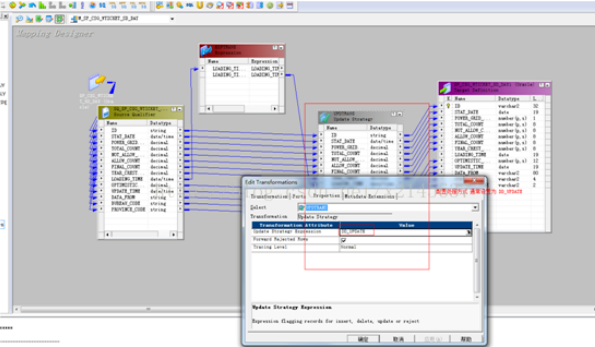
**3 设置统一为更新**

设置loginTime



**4 设置输入输入，名命规则。输入：字段名\_IN 输出：字段名\_OUT**





**5 设置源表抽取数据范围**

