

# 第 6 章 数据清理和数据转换技术

内容提要：数据清理和数据转换是面向数据的计算机审计的必经步骤，也是计算机审计实务中的一个难点，本章循序渐进地介绍了数据清理和数据转换的技术和方法。

## 6.1 数据清理

数据清理是面向数据的计算机审计中的重要一环。检查、控制和分析审计数据的质量，在数据质量问题上发现审计线索，清理有质量问题的数据，为后续的审计数据分析服务，这些是面向数据的计算机审计的一个技术重点。

原始的审计数据是审计人员从被审计信息系统中获得的，这些数据的质量如何？审计人员要首先进行评价。数据质量的主要评价指标是：

- 准确性：数据值与假定正确的值的一致程度。
- 完整性：需要值的属性中无值缺失的程度。
- 一致性：数据对一组约束的满足程度。
- 惟一性：数据记录（及码值）的惟一性。
- 适时性：在所要求的或指定的时间提供一个或多个数据项的程度。
- 有效性：维护的数据足够严格以满足分类准则的接受要求。

原始数据中可能存在的数据质量问题有很多种，包括缺失的或不完整的数据、不准确的数据、不一致的数据、重复的记录等。

例如，在一次对交通部门车辆购置附加费（现改为附加税）的审计中，审计人员采集到的原始数据表“车购费征费情况表”中，存在车辆发票价（GC\_03）和车辆最低征费额（GC\_06）字段值为空（Null）的现象。如图 6-1 所示。在进行数据分析时，原始数据为空的记录值并不等同于“0”，不能参加比较大小说分析，在数据处理过程中会被遗漏掉，必须对这部分空值进行数据清理。

审计人员同时发现征费系统对征费人员录入的如发票价这样的数据并没有值域的约束性限制，极易造成因人员操作失误等原因导致录入的数据中存在错误值的情况。如图 6-2 所示。审计人员通过了解车购费的征费业务知识，对发票价的值进行了错误值的检测。根据车购费征费业务规则可知，一辆国产汽车的购入价一般不会超过 100 万元。可在采集的原始数据中却有发票价为 400 万元和 745.5 万元的数据，这显然是错误的

GC_KEY	GC_03	GC_04	GC_05	GC_06	GC_07	GC_08	GC_09	GC_10	GC_11	GC_01	GC_02
22090	63000	63000	10	5400	6300		<input type="checkbox"/>				
22091	165000	165000	10	14900	16500		<input type="checkbox"/>				
22092	51000	51000	10	4500	5100		<input type="checkbox"/>				
22093	90000	90000	10	9000	9000		<input type="checkbox"/>				
22094	96000	96000	10	7200	9600		<input type="checkbox"/>				
22095	2700	2700	10	270	270		<input type="checkbox"/>				
22096	64000	64000	10	5500	6400		<input type="checkbox"/>				
22098	32900	32900	10	2880	3290		<input type="checkbox"/>				
22099	42000	42000	10	4200	3150		<input type="checkbox"/>				
22101	55000	55000	10	5500	4125		<input type="checkbox"/>				
22102	130000	130000	10	12600	11830		<input type="checkbox"/>				
22103	60000	60000	10	5500	6000		<input type="checkbox"/>				
22104	120000	120000			12000		<input type="checkbox"/>				
22105	110000				2		<input type="checkbox"/>				
22106				650	500		<input type="checkbox"/>				
22107				650	500		<input type="checkbox"/>				
22109					2		<input type="checkbox"/>				
22110				9100	9100		<input type="checkbox"/>				

最低征额为空值  
发票价为空值

图 6-1 含有空值的数据

关键字	发票价	GC_04	GC_05	最低征收额	实际征收额	GC_08	GC_09	GC_10	GC_11	GC_01	GC_02
30040	65000	46500		0	3750		<input type="checkbox"/>				
30041	65000	46500		0	3750		<input type="checkbox"/>				
30042	88000	88000		0	6600		<input type="checkbox"/>				
30043	88000	88000		0	6600		<input type="checkbox"/>				
30044	88000	88000		0	6600		<input type="checkbox"/>				
30045	88000	88000		0	6600		<input type="checkbox"/>				
30046	63500	87300		0	8730		<input type="checkbox"/>				
30047	4000000	317900		0	31790		<input type="checkbox"/>				
30048	7455000	884000		0	88400		<input type="checkbox"/>				
30049	20000	20000		0	2000		<input type="checkbox"/>				
30050	20000	20000		0	2000		<input type="checkbox"/>				
30051	66960	5025		0	5025		<input type="checkbox"/>				
30052	0			14000	15000		<input type="checkbox"/>				
30053	0			0	690		<input type="checkbox"/>				
30055	0			13200	14500		<input type="checkbox"/>				
30057	0			4700	4700		<input type="checkbox"/>				

发票价为400万元

发票价为745.5万元

图 6-2 原始数据中含有大于 100 万元的错误值

需要整理的数据还有另一种情况，即并不是错误，而是不在数据库或 Excel 中进行查询、筛选操作。如图 6-3 中，有时间页码数据和空行。时间页码数据是不规则的数据，如不删除所有这些不规则数据就不能进行查询、筛选、分析等操作，所以必须进行清理删除。

由于被审计数据来源众多，种类繁多，会存在不少的数据质量问题。在面向数据的审计中，审计的对象是数据，不能简单地把有质量问题的数据抛弃掉，因为这些数据中有可能蕴涵审计线索。首先要做的是根据数据质量的要求，对审计数据进行检查，对发现的数据质量问题进行分析，找出造成问题的原因，发现隐含的审计线索。然后对数据进行清理，清理有质量问题的数据。清理的目的是为后续的数据分析做准备，有问题的数据会给数据分析工作带来错误。



010516100780	飞霞化纤纺织公司	0.00
010516100816	某马镇静怡服饰商店	4.58
010516100828	某马镇彩印塑料包装厂	203.15
010516100830	某市马镇兴华整流器厂	15127.58
010516100855	某市江南纺织器材厂	20011.67
010516100867	某市马镇万佳副食店	15896.56
010516100879	某市苏澄机械厂	0.00
010516100881	某市伯达塑业有限公司	85.28
010516100893	某市宝马建筑电塑有限公司	16058.05
010516100917	某市马镇国生服装厂	0.00
010516100929	某市百顺不锈钢管件有限公司	80548.05
010516100931	某市马镇益秀涤纶整理厂	0.00
010516100968	某市长盛船舶清仓有限公司	922.45
010516100982	某市力达化机设备有限公司	0.00
010516101008	某市宏伟粉末冶金有限公司	30022.29
010516101010	某市鸿运合金有限公司	5849.50
010516101123	某市华亭化纤布业有限公司	17.21
010516101135	某市开源压缩机备件有限公司	61567.95
010516101150	某市双喜针织制衣有限公司	400015.00
010516101198	某市久安车业有限公司	156706.59
010517100019	马镇商业社	0.00
010517100033	某市马镇涂塑制品厂	0.00
010517100045	某市马镇镇建筑管理科	0.00
010517100060	某市马镇纺织器材厂	13711.09

图 6-4 经过清理后的数据

## 6.2 数据转换

解决对被审计单位不同类型数据库格式的识别问题，是一个语法层次上的问题；数据转换技术要解决对采集到的原始数据的含义进行识别的问题，同时，它还要将具有相同或相近含义的各种不同形式的数据转换成审计软件处理所需的形式相对统一的数据，这是一个语义层次上的问题。数据转换的前提是数据采集。

### 6.2.1 为什么要进行数据转换

#### 1. 被审计单位信息系统的多样性带来了数据的不一致性

开展计算机审计必然面临各式各样的迥然相异的被审计单位信息系统。被审计单位信息系统的差异，必然给审计工作带来数据的不一致性问题。数据的不一致性大体有以下表现形式：

同一字段在不同的应用中具有不同的数据类型。例如，字段“借贷方标志”在 A 应用中的类型为“字符型”，取值为“Credit/Debit”，在 B 应用中的类型为“数值型”，取值为“0/1”，在 C 应用中类型又为“布尔型”，取值为“True/False”。

同一字段在不同的应用中具有不同的名字。例如，A 应用中的字段“balance”在 B 应用中名称为“bal”，在 C 应用中又变成了“currbal”。

同名字段，不同含义。例如，字段“月折旧额”在 A 应用中表示用直线折旧法提取的月折旧额，在 B 应用中表示用加速折旧法提取的月折旧额，等等。

同一信息，在不同的应用中有不同的格式。例如，字段“日期”在 A 应用中的格式为“YYYY-MM-DD”，在 B 应用中格式为“MM/DD/YY”，在 C 应用中格式又为“DDMMYY”。

同一信息，在不同的应用中有不同的表达方式。例如，对于借贷方发生额的记录，在 A 应用中设计为两个字段：“借方发生额”与“贷方发生额”，在 B 应用中设计为两个字段“借贷方标志”与“借贷方发生额”。

对于这些不一致的数据，必须进行转换后才能供审计软件分析之用。数据的不一致性是多种多样的，对每种情况都必须专门处理，因此，这是一项很繁重的工作。

## 2. 被审计系统的安全性措施给审计工作带来障碍

基于安全性考虑，被审计单位的系统一般都采取一定的加密措施，有系统级的加密措施和数据级的加密措施。特别对具有一定含义的数据库的表与字段的名称，一般都要进行映射或转换。例如，将表命名为 T1, T2, ...；将字段命名为 F1, F2, ...。对于这样的数据，不进行含义的对照与转换就不明白表或字段的经济含义，审计人员是无法处理的。各种各样的加密措施不胜枚举，这些都给计算机审计带来了障碍，也给数据转换带来挑战。

## 3. 审计目的的不同决定了审计数据的范围和要求不同

被审计单位的信息系统规模不一，数据量相差悬殊。审计人员不可能也没有必要将被审计单位的所有数据都采集过来。审计工作有自己的目的和要求，需要处理的往往是某一时段或某一方面发生的业务数据。这样就要求审计人员选取一定范围的、满足一定要求的审计数据。例如，在关税审计业务中，关心的只是与税收征管有关的报关单、加工贸易、减免税、货运单、税单等相关数据库中的数据，而对于操作员授权管理等系统控制数据库，在不对系统进行评价时，可以不采集。同时，审计目的不同，对原始数据的要求也不同。在对银行贷款利率执行情况检查时，需要带有计息标志和利率信息的每一笔贷款的明细数据，如果这样的信息分布在被审计系统的若干张表中，换句话说，如果被审计单位不能提供完全满足审计要求的数据，就有必要对采集到的数据进行转换。

## 4. 数据转换是数据分析、处理的前提

计算机软件设计一般都是基于一定的数据结构的，专用的审计软件更是如此。在输入数据不满足软件处理的需求时，必须对它进行转换。对于一些通用的审计软件，对输入数据的适应性相对强一些，但这并不意味着它可以处理不经转换的任意数据。审计软件中有很多特定的分析方法和专用的工具，这些方法和工具往往要求一定的数据结构。例如，在计算固定资产折旧时，就需要用到“资产原值”、“月折旧率”、“入账日期”等信息。如果这些信息不能完全满足，就无法计算。

### 6.2.2 数据转换的内容和任务

所谓数据转换，从计算机审计的需求来讲，主要包括两个方面的内容：一是将被审计单位的数据有效地装载到审计软件所操纵的数据库中；二是明确地标识出每张表、每

个字段的经济含义及其相互之间的关系。

数据转换的第一步工作，是进行数据的有效性检查。为避免数据冗余和差错，在转换之前，应该对数据进行有效性检查，这是很重要的。如果没有进行数据的有效性检查，就有可能破坏审计软件处理所需的完整性。检查数据的有效性的最好方法是获得被审计单位的有关人员，包括具有技术专业知识和业务专业知识的人员的帮助。

在有效性检查完成后，就要进行数据的清除和转换了。所谓清除，指的是去掉那些与审计目的无关的数据，而仅仅将审计工作所关注的那些数据采集过来。下面定义数据转换的几种基本类型，每一类都有自己的特点和表现形式。

### 1. 简单变换

顾名思义，它是数据转换中最简单的形式，也是所有数据转换的基本构成单元。这些变换一次改变一个字段的属性，而不考虑该属性的背景或与它相关的其他字段的信息。

#### (1) 数据类型转换

最常见的简单变换是转换一个数据元的类型。它是将一种类型的数据转换成另一种类型的数据。数据转换的前提是类型相容。在这里，类型相容指的是一种类型数据的值域可以通过常用的转换函数映射到另一种类型的值域上。这种映射不会丢失数据的精确度。类型相容的转换被认为是合适的转换，如整型（16bit）到文本（字符）型的转换；类型不相容的转换被认为是不合适的转换，如文本（字符）型到整型的转换。以 Microsoft Access 中的某些数据类型为例，下列转换被认为是合适的转换(见表 6-1)。

表 6-1 合适的类型转换

源类型	目标类型	精确度
整型（16bit）	文本（字符）型	
整型（16bit）	单精度型、双精度型	不变
整型（16bit）	长整型（32bit）	不变
长整型（32bit）	文本（字符）型	
日期/时间型（32bit）	文本（字符）型	
单精度型	文本（字符）型	
双精度型	文本（字符）型	

下列转换被认为是不合适的转换(见表 6-2)。

表 6-2 不合适的类型转换

源类型	目标类型	精确度
文本（字符）型	整型	
文本（字符）型	单精度型	
文本（字符）型	双精度型	
文本（字符）型	日期/时间型	
长整型（32bit）	单精度型（16bit）	可能丢失

源类型	目标类型	精确度
长整型 (32bit)	双精度型 (32bit)	可能丢失
整型	日期/时间型	
长整型	日期/时间型	
日期/时间型	整型	
日期/时间型	长整型	
日期/时间型	单精度型	
日期/时间型	双精度型	

有时候，特定的转换任务要求必须进行不合适的转换，这时可以通过编写简单的程序（包括 SQL 语句组）来完成，例如：

在“凭证”表中，“月份”为文本型字段，如果目标表“凭证\_目标”的“月份”为数值型字段，要完成这一类型转换，可以在 SQL 语句中利用 INT（）函数将文本型的月份转换成数值型的月份。用于转换的 SQL 语句如下：

```
SELECT 日期, 凭证号, 科目代码, 摘要, 数量, 金额, 借贷方, INT(凭证表.月份) AS 月份,
对方科目, 单位代码
INTO 凭证_目标
FROM 凭证表
```

### (2) 日期/时间格式的转换

因为大多数系统都采用许多不同的日期和时间格式，所以在审计软件中几乎都要进行日期和时间格式的转换，将它转换成审计软件处理所需的统一格式。这可以通过手工程序编码来完成。它能将一个日期或时间字段拆成几个子部分，然后再将它们拼成想要的格式和字段。然而大多数审计软件中的数据导入和转换工具都提供了日期和时间格式之间转换的设置，采用手工编码的情况就比较少了。

例如，在“入库单”表中，“日期”为文本型字段。假设目标表“入库单\_目标”的“日期”为“日期/时间”型字段，可以用《审计数据采集分析 2.0》中的数据转换工具完成数据转换。如图 6-5 所示。

另外，在数据导入的过程中，审计软件也提供了相应的日期转换功能。例如，在对“贷款利率表”文本文件进行导入的过程中，可以通过设定日期字段的格式来识别日期字段的值并对其进行转换。如图 6-6 所示。

### (3) 代码转换

在业务数据库中建立代码是为了节省数据库存储空间和提高计算机的处理效率。这些代码一般是由系统管理员设置，由应用程序维护的。这给审计软件处理带来了很大的不便。有两种方法可以解决这一问题。如果审计软件中采用了代码设计，而被审计单位的代码能够满足审计软件需要的，可以将被审计单位的代码表转换到审计软件的代码表

上来；如果审计单位的代码不能满足审计软件的需要，就必须根据审计软件的要求对它重新编码。例如，被审计单位提供的汇率表、利率表，有时就可以满足审计软件的需要。如果审计软件不采用代码设计，就必须将被审计单位数据中的代码值转换为经过解码的、易于理解的相应值。



图 6-5 《审计数据采集分析 2.0》中的日期时间转换

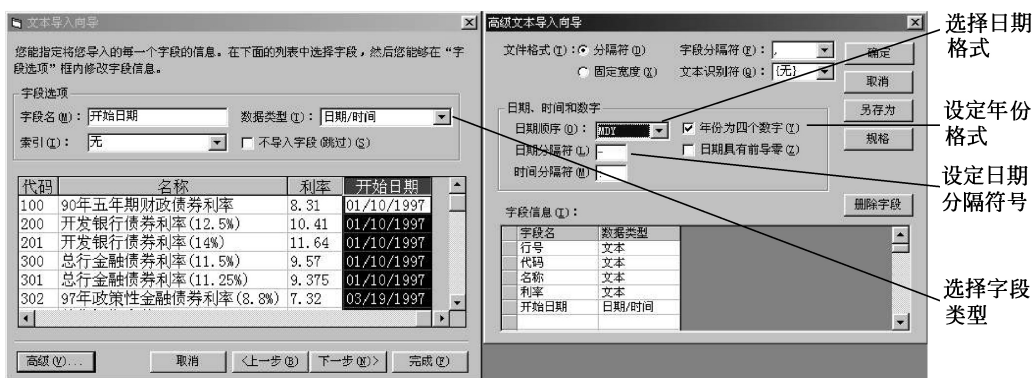


图 6-6 数据导入过程中的日期时间转换

下面的例子将借贷方标志这个代码转换成了相应的值。

在审计过程中，为了便于大多数审计人员对数据进行处理，往往需要对采用编码（外语、拼音、字符等）形式作为名称的表与字段进行名字的转换与标注。这一步工作的前提是通过被审计单位提供的数据字典或审计人员的经验，对被审计系统的数据结构有了



图 6-7 《审计数据采集分析 2.0》中的代码转换

较为清楚的了解。例如，表 6-3 是被审计单位提供的原始数据形式。

为了方便大多数审计人员使用，在识别含义的前提下将其转换为下面的形式（见表 6-4）。

表 6-3 从被审计单位采集到的原始数据的结构（部分）

表名：KMJL	
字段名	字段类型
KMDM	文本型
BMDM	文本型
RQ	日期/时间型
PDH	文本型
BH	文本型
JFJE	数值型
DFJE	数值型
ZY	文本型

表 6-4 经过转换的被审计单位的数据（部分）

表名：科目记录表	
字段名	字段类型
科目代码	文本型
部门代码	文本型
日期	日期/时间型
凭单号	文本型
编号	文本型
借方金额	数值型
贷方金额	数值型
摘要	文本型

#### (4) 值域转换

值域转换是将一个字段的全部或部分取值映射到另一个字段的全部或部分取值上去。

例如，在被审计单位的系统中，某一项数据的取值分为 10 个级别（“0”~“9”），而在审计软件中，该项数据的取值仅仅分为 5 个级别（“A”~“E”），这就需要将具有 10 个级别的值域映射到具有 5 个级别的值域上去。编码转换可以看成是一类特殊的值域转换。

## 2. 清洁和刷洗

清洁和刷洗是两个可互换的术语，指的是比简单变换更复杂的一种数据变换。在这种变换中，要检查的是字段或字段组的实际内容而不仅是存储格式。一种清洁是检查数据字段中的有效值。这可以通过范围检验、枚举清单和相关检验来完成。

### (1) 有效值

范围检验是数据刷洗的最简单形式，它是指检验一个字段中的数据以保证它落在预期范围之内，通常是数值范围或日期范围。

枚举清单也相对容易实现。这种方法是对照数据字段可接受值的清单检验该字段的值。

相关检验稍微复杂一些，因为它要求将一个字段中的值与另一个字段中的值进行对比，看它们是否满足一定的相关关系。当然，数据清洁规则往往是这些不同方法的结合。

### (2) 复杂的重新格式化

数据刷洗的另一种主要类型是重新格式化某些类型的数据，这种方法适用于将许多不同方式存储在不同数据来源中的信息转换成审计软件所要求的统一的表示方式。最需要格式化的信息之一是摘要信息，由于没有一种书写摘要的标准方式，所以同一个内容的摘要可以用许多不同方式表达出来。这就要求将摘要解析成几个组成部分，然后将这些组成部分进行转换并重新排列成一个统一的格式。

#### • 集成

要把从各种不同的数据源中得到的被审计单位的业务数据结合在一起，真正的困难在于将它们集成为审计软件所需的，一个紧密结合的数据模型。这是因为数据必须从多个数据源中提取出来，并结合成为一个新的实体。这些数据源往往遵守的不是同一套业务规则，在生成新数据时，必须考虑到这一差异。

#### • 字段水平的简单映射

字段水平的简单映射在必须执行的数据变换总量中占了一大部分。这种映射的定义是指源数据中的一个字段被转移到目标数据字段中的过程。在这个过程中，该字段可以利用前面讨论过的任何一种简单变换进行变换，它可以被刷洗或重新格式化。

#### • 复杂集成

在审计软件中，数据转换有时会比从源字段到目标字段的简单移动复杂一些。为了将源数据变换为目标数据，这些复杂集成必须做更多的分析。

##### ■ 目标元素的多个来源

当同一个目标数据元有多个来源时，会出现另一个复杂的数据集成问题，即很难保证该元素的各个来源总能保持一致。实际上，这样的数据元存在矛盾值比不同来源中的数据取值相同更为普遍。解决冲突的简单办法是指定某一系统在冲突中占据主导地位。

##### ■ 衍生数据/计算数据

数据转换的最常见形式之一就是计算和生成衍生数据元或计算数据元。它包括平均值、总和或统计计算，还包括复杂的业务计算。衍生数据字段通常是冗余的，

因为计算中涉及的数据也存储在数据库中，然而它却能大大简化查询。保证存入数据库中的这些衍生值的正确性和一致性，就可以在查询中选用它们，而不必在用户需要时都重新计算一次。在这方面，数据采集和转换工具是很有用的，因为这些工具能迅速而轻松地进行各种计算，无需担心编程人员是否正确地编写了计算逻辑。

### 6.2.3 数据转换的一般方法

数据转换是一项复杂的工作，有很多前期的准备工作要完成。在转换被审计单位的数据之前，首先要对被审计单位的信息系统进行一番调查研究。请被审计单位技术人员配合讲解系统设计、功能划分、操作流程等等。同时，获取被审计单位的系统设计文档资料至关重要。被审计单位的数据字典中详细地描述了数据库中的数据项与事务处理逻辑，这些十分有助于审计人员了解被审计单位信息系统的数据情况。

在这一步工作的基础上，才能着手选择需要转换的数据。选择好这些数据之后，就可以考虑采用一定的方法进行转换了。数据转换的方法一般来说有 3 种。

#### 1. 专用工具

专用工具是为数据转换而专门设计的一类辅助工具。对于简单变换和一部分的数据清洁、刷洗和集成，这类工具也可以完成。但对于复杂的转换，有些工具就不能胜任了。根据用途不同，这些工具大致可以分成两类。

##### (1) 数据仓库中的数据采集与转换工具

以 IBM Data Warehouse 为代表的数据库工具就是一个非常出色的数据转换工具。它所支持的数据源类型有：

- DB2 Family
- Oracle
- Sybase
- Microsoft SQL Server
- Informix
- Generic ODBC
- Flat File LAN
- VSAM
- IMS

它是为数据仓库装载数据而设计的一个工具，如果审计软件的数据库平台是 DB2，这种工具可以起到很好的作用。

##### (2) 审计软件中提供的数据转换工具

几乎每一种审计软件都提供了自己的数据转换工具。这些工具可以分为专用的和通用的两大类。专用的转换工具只能转换几种特定系统中的特定数据，如海关审计软件中的数据转换工具；通用的转换具有更大的灵活性。《审计数据采集分析 2.0》中提供的数据转换工具就是一个通用的转换工具，它能够将文件类型数据库（dBase 系列、Fox 系列、Access 系列）中的数据转换到 Access 数据库中。它能完成类型转换、日期/时间格式转换和值域转换。例如，为了将一张“贷款记录表”转换到《金融审计工具包》中，以便进行贷款利率执行情况的检查，可以进行如下操作（见图 6-8、图 6-9）：



图 6-8 利用《审计数据采集分析 2.0》进行数据转换



图 6-9 显示数据转换后的结果

选取源表——选择被转换的原始数据“D:\贷款利率执行检查数据\某行借据数据1.mdb”，这些数据可以是 DBASE 系列、Fox 系列、Access 系列数据库中的表。

选择目标表——指定需要转换到的数据库中的表“C:\Program Files\金融审计工具包\金融审计数据库.mdb”，目前仅支持 Access 类型的数据库。

定义字段之间的对应关系——通过分别选定源字段和目标字段，双击后就可以建立起字段之间的对应关系。在建立对应关系的时候，自动检查源类型和目标类型之间的相容性，提示用户进行日期/时间格式设定、值域转换。本例对“单位名称”、“本金”、“发放日”、“到期日”、“利率”字段建立了对应关系。

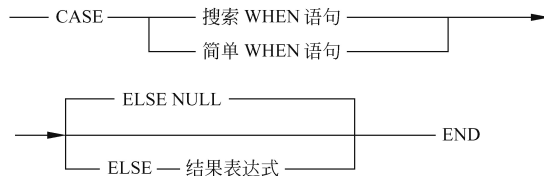
完成转换——单击“显示结果”按钮时，即进行数据转换。

保存转换结果——确信转换结果无误时，将结果永久保存。

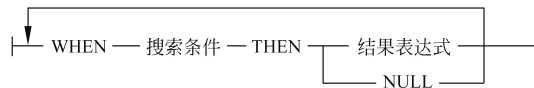
### (3) SQL 语言

SQL 语言是关系数据库的标准语言（详见第 8 章），利用 SQL 语言进行数据转换，对于技术人员来说尤为适合。SQL 语言中的语句可以分为数据定义语句（DDL）、数据操纵语句（DML）和数据控制语句（DCL）。数据转换中用得较多的是数据操纵语句和数据定义语句。一般可以用数据定义语句来定义目标数据库和目标表的结构，用数据操纵语句将源数据检索到目标数据库中并对检索的结果进一步加工。

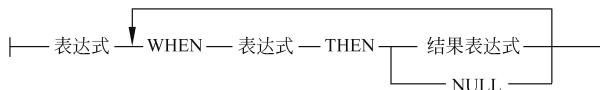
为了提高性能，更好地完成数据查询与操纵，很多数据库都采用扩展的 SQL 语言。这些扩展的 SQL 语言各有特色。例如，DB2 的 SQL 语句中增加了 CASE 表达式，语法如下：



搜索 WHEN 语句：



简单 WHEN 语句：



CASE 语句可以很方便地在检索过程中完成一部分数据转换工作。下面是一些例子。

在“商品销售表”中，类别 = “2” 表示销售记录，类别 = “4” 表示上年结转记录。可以通过下面的语句完成类别代码的转换。

```
SELECT 入账日期, 单位代码, 商品代码, 金额, 数量, 单价, 月份,  
CASE 类别
```

```

        WHEN '2' THEN '销售'
        WHEN '4' THEN '结转'
    END

```

FROM 商品销售表;

在“商品销售表”中的字段“数量”表示销售数量，假定可以根据销售数量来确定市场需求情况，那么可以用 CASE 表达式来将销售数量转换成市场需求程度。

```

SELECT 入账日期, 单位代码, 商品代码, 金额, 单价, 月份,
CASE
    WHEN 数量 > 200 THEN '极受欢迎'
    WHEN 数量 <= 200 AND 数量 > 100 THEN '受欢迎'
    WHEN 数量 <= 100 AND 数量 > 50 THEN '一般'
    WHEN 数量 <= 50 AND 数量 > 10 THEN '不受欢迎'
    ELSE '极不受欢迎'
END

```

FROM 商品销售表

CASE 语句还可用于防止转换过程中的被零除错误。例如，下面的语句用于查找那些固定资产月折旧额大于 2% 的记录。

```

SELECT 设备代码, 单位代码, 资产编号, 规格及型号, 单位, 数量, 使用日期, 入账日期,
    凭证号, 资产原值, 月折旧额/资产总值, 累计折旧, 资产净值
FROM 固定资产明细
WHERE (CASE WHEN 资产总值=0 THEN NULL
        ELSE 月折旧额/资产总值
        END) > 0.02;

```

可见，这些扩展的 SQL 语言具有很强的数据转换功能和灵活性。另外，不同的数据库系统中都提供一整套的函数（包括类型转换函数），这些函数十分有助于完成数据转换。

#### (4) 程序编码

这是一种最基本的转换方法。对于下列情形，可以考虑采用程序编码的方式进行转换。

对复杂数据文件中包含的数据进行转换时

复杂数据文件一般是指非关系型数据库的数据文件。一些变长记录结构的文本文件和一些应用程序用于数据存储、交换的特定格式的二进制文件都属于复杂数据文件。复杂数据文件一般有自己特定的格式，最好通过程序编码的方式进行数据转换。例如，交易流水的二进制文件（Detail.bin）的格式为：

```

XXXXXXXXXX   XXXXXXXX   XXXXXXX   XXXXXXXXXXXXXXXX
交易号（10）  日期（8）   时间（6）  交易金额（共 14 位，4 位小数）
每一笔交易占用 19 个字节，每个字节以 BCD 码形式存放数据

```

在“金融审计数据库”中，“交易流水”表的结构如表 6-5 所示。

表 6-5 “金融审计数据库”中“交易流水”表结构

字段名	字段类型
交易号	文本型(5位字符宽度)
交易时间	日期/时间型
交易金额	双精度浮点型

可以采用下面的代码来将交易记录转换到“金融审计数据库”的“交易流水”表中。

Private Sub TransDetail2MDB()

Dim dbFinancial As Database	标识"金融审计数据库"的数据库变量
Dim rstFinancial As Recordset	标识"交易流水"的记录集变量
Dim intFile As Integer	标识文件句柄的变量
Dim nMaxSize As Long	标识文件大小的变量
Dim nPosition As Long	标识当前读指针的位置的变量
Dim nRecCount As Long	标识记录总数的变量
Dim nCurrRec As Long	标识当前记录位置的变量
Dim tmpStr As String	临时字符串
Dim tmpByte As Byte	临时字节变量
Dim sTransCode As String	保存"交易代码"的变量
Dim sDateTime As String	保存"交易时间"的变量
Dim sTransValue As String	保存"交易金额"的变量
Dim i As Integer	循环变量

打开数据库

Set dbFinancial = OpenDatabase("C:\金融审计数据库.MDB",True,False)

打开记录集

Set rstFinancial = dbFinancial.OpenRecordset("交易流水",dbOpenDynaset)

intFile = FreeFile

分配文件句柄

打开文件

Open "C:\Detail.bin" For Binary Access Read As intFile

nMaxSize = LOF(intFile)

获得文件大小

nRecCount = Int(nMaxSize / 19)

获得记录总数

nPosition = 1

初始化读指针

初始化当前记录指针

nCurrRec = 0

以下部分完成数据转换

While nCurrRec < nRecCount

tmpStr = ""

清空临时字符串

For i = 1 To 5

读交易代码

Seek intFile,nPosition

定位读指针

Get intFile,nPosition,tmpByte

读出一个字节

tmpStr = tmpStr & Chr((tmpByte And &HF0) / 16 + 48)

转成 ASCII 码

tmpStr = tmpStr & Chr((tmpByte And &HF) + 48)

转成 ASCII 码

```

        nPosition = nPosition + 1
    Next i
    sTransCode = tmpStr
    '以下部分读取交易时间
    tmpStr = ""
    For i = 1 To 4
        Seek intFile,nPosition
        Get intFile,nPosition,tmpByte
        tmpStr = tmpStr & Chr((tmpByte And &HF0) / 16 + 48)
        tmpStr = tmpStr & Chr((tmpByte And &HF) + 48)
        nPosition = nPosition + 1
    Next i
    sDateTime = Mid(tmpStr,1,4) & "-" & Mid(tmpStr,5,2) & "-" & Mid(tmpStr,7,2)

    tmpStr = ""
    For i = 1 To 3
        Seek intFile,nPosition
        Get intFile,nPosition,tmpByte
        tmpStr = tmpStr & Chr((tmpByte And &HF0) / 16 + 48)
        tmpStr = tmpStr & Chr((tmpByte And &HF) + 48)
        nPosition = nPosition + 1
    Next i
    SDateTime = sDateTime & " " & Mid(tmpStr,1,2) & ":" & _
        Mid(tmpStr,3,2) & ":" & Mid(tmpStr,5,2)
    '以下部分读取交易金额
    tmpStr = ""
    For i = 1 To 7
        Seek intFile,nPosition
        Get intFile,nPosition,tmpByte
        tmpStr = tmpStr & Chr((tmpByte And &HF0)/ 16 + 48)
        tmpStr = tmpStr & Chr((tmpByte And &HF) + 48)
        nPosition = nPosition + 1
    Next i
    sTransValue = tmpStr
    '进行数据转换并记录到数据库中
    rstFinancial.AddNew
    rstFinancial.Fields("交易号").Value = sTransCode
    rstFinancial.Fields("交易时间").Value = CDate(sDateTime)
    rstFinancial.Fields("交易金额").Value = CDbl(sTransValue) / 10000
    rstFinancial.Update
    '记录指针下移
    nCurrRec = nCurrRec + 1
Wend

```

读指针下移

暂存交易代码

rstFinancial.Close  
dbFinancial.Close  
End Sub

关闭记录集  
关闭数据库

### 对于非关系型数据库中的数据进行转换时

对于非关系型数据库中的数据进行转换，如果不想购买昂贵的数据库网关或其他转换产品，就只能采用程序编码的方式。值得注意的是，这里的程序编码一般要在被审计单位的系统中完成。即在被审计单位技术人员的配合下，在被审计单位的系统上书写一段程序将特定的数据转换成审计人员所需要的格式。

### 对于特别复杂的关系型数据库中的数据进行转换时

有些关系数据库中数据项之间的关系错综复杂，采用 SQL 语言实现转换代价很高。这时可以考虑程序编码方式。利用软件开发工具中的数据访问对象和方法，可以很方便地书写数据转换代码。

### 数据转换的需求固定，使用频繁时

这一类任务最好采用程序编码方式来完成。编写一段相对完整、考究的转换程序可谓一劳永逸。它既避免了多次使用转换工具或 SQL 语言带来的操作误差，又不需要使用者具有很多数据处理方面的专业知识，可谓操作简单，效率可观。

例如，在《海关业务审计软件 2.0》中，需要转换的数据为：年度进出口报关单、年度税单和年度减免税证明表。由于数据转换需求固定，因此可以编写一段代码来实现这一任务，并将这一段代码作为《海关业务审计软件 2.0》的一个子功能。《海关业务审计软件 2.0》中数据转换子系统的部分界面如图 6-10 和图 6-11 所示。

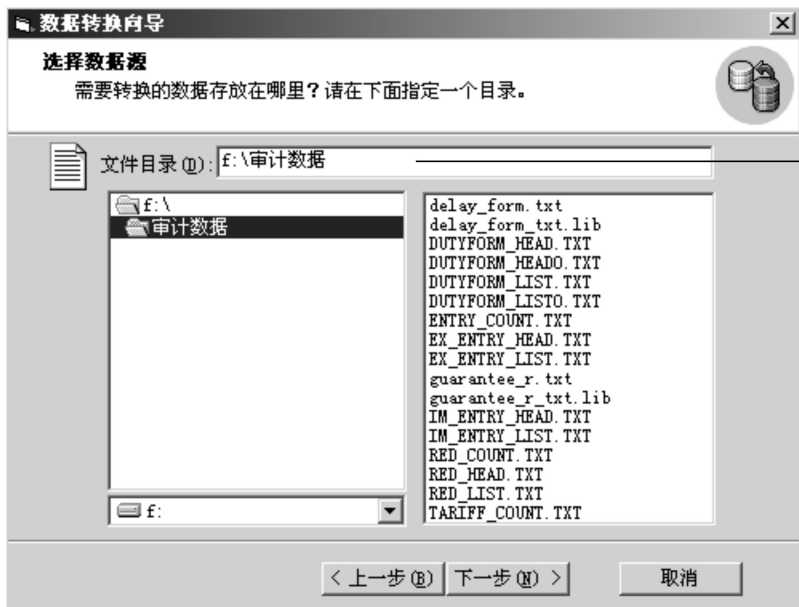


图 6-10 《海关业务审计软件 2.0》中的数据转换模块（之一）