



中华人民共和国国家标准

GB/T 4092.7—1992

程序设计语言 COBOL 排序-合并模块

Programming language COBOL
Sort-merge module

1992-08-04 发布

1993-05-01 实施

国家技术监督局 发布

程序设计语言 COBOL
排序-合并模块

GB/T 4092.7—1992

代替 GB 4092.7—83

Programming language COBOL
Sort-merge module

1 引言

1.1 功能

排序-合并模块,提供这样一种功能:按照一组由用户提出的包含在每个记录中的键,对一个或多个文卷中的记录排序或者合并两个或多个有同样次序的文卷中的记录。此外,用户可以选用输入或输出过程对记录进行特定的处理。这种特定处理可以在记录被 **SORT** 语句排序之前或之后应用或在记录被 **MERGE** 语句合并之后应用。

1.2 语言概念

1.2.1 排序文卷

排序文卷是由 **SORT** 语句进行排序的记录集。排序文卷没有程序员能够控制的标号过程,并且对于分块及内容存储分配的规则仅由 **SORT** 语句所特有。就缓冲区、块或卷而论,**RELEASE** 语句以及 **RETURN** 语句没什么意义。然而,排序文卷可以认为是一个内部文卷,该文卷根据输入文卷建立(**RELEASE** 语句),进行处理(**SORT** 语句),然后使之成为可用的(**RETURN** 语句)输出文卷。

排序文卷通过文卷控制款命名并由排序-合并文卷描述款来描述。排序文卷通过 **RELEASE**、**RETURN** 和 **SORT** 语句引用。

1.2.2 合并文卷

合并文卷是由 **MERGE** 语句进行合并的记录集。合并文卷没有程序员能够控制的标号过程并且对于分块及内部存储分配的规则仅由 **MERGE** 语句实行。就缓冲区、块或卷而论,**RETURN** 语句没什么意义。然而,合并文卷可以认为是一个内部文卷,该文卷根据输入文卷建立并通过对它们的合并(**MERGE** 语句),成为可用的(**RETURN** 语句)输出文卷。

合并文卷由文卷控制款命名并由排序-合并文卷描述款描述。合并文卷通过 **RETURN** 和 **MERGE** 语句引用。

2 排序-合并模块的环境部

2.1 输入-输出节

涉及输入-输出节的信息见顺序 I-O 模块的 2.1。

2.2 FILE-CONTROL 段

涉及 **FILE-CONTROL** 段的信息见顺序 I-O 模块的 2.2。

2.3 文卷控制款

2.3.1 功能

文卷控制款说明排序或合并文卷的有关物理属性。

2.3.2 一般格式

$$\underline{\text{SELECT}} \quad \text{文卷名 1} \quad \underline{\text{ASSIGN}} \quad \text{TO} \left\{ \begin{array}{l} \text{实现名 1} \\ \text{字值 1} \end{array} \right\} \dots$$

2.3.3 语法规则

(1)数据部中描述的每一个排序或合并文卷,在 **FILE-CONTROL** 段中能且只能指称一次。**SELECT** 子句中指出的每一个排序或合并文卷在同一程序的数据部中必须有一个排序-合并文卷描述款。

(2)因为文卷名 1 表示一个排序或合并文卷,所以在 **FILE-CONTROL** 段中只有 **ASSIGN** 子句允许跟在文卷名 1 之后。

2.3.4 一般规则

(1)**ASSIGN** 子句指出文卷名 1 引用的文卷和实现名 1 或字值 1 引用的存储媒体的联系。

2.4 I-O-CONTROL 段

2.4.1 功能

I-O-CONTROL 段指出包括排序或合并文卷的不同文卷共享的存储区。

2.4.2 一般格式

I-O-CONTROL

$$\left[\left[\underline{\text{SAME}} \left\{ \begin{array}{l} \underline{\text{RECORD}} \\ \underline{\text{SORT}} \\ \underline{\text{SORT-MERGE}} \end{array} \right\} \text{AREA FOR 文卷名 1\{文卷名 2\}\dots} \right] \dots \right]$$

2.4.3 语法规则

(1)**SAME** 子句 **RECORD** 选择的可用性依赖于由实现支撑的顺序 **I-O** 模块的级。

2.4.4 一般规则

(1)有关排序-合并模块的 **SAME RECORD/SORT/SORT-MERGE AREA** 子句在下面介绍。

2.5 SAME RECORD/SORT/SORT-MERGE AREA 子句

2.5.1 功能

SAME RECORD/SORT/SORT-MERGE AREA 子句中指出不同文卷(至少有一个是排序或合并文卷)共享的存储区。

2.5.2 一般格式

$$\underline{\text{SAME}} \left\{ \begin{array}{l} \underline{\text{RECORD}} \\ \underline{\text{SORT}} \\ \underline{\text{SORT-MERGE}} \end{array} \right\} \text{AREA FOR 文卷名 1\{文卷名 2\}\dots}$$

2.5.3 语法规则

(1)在 **SAME RECORD/SORT/SORT-MERGE AREA** 中子句指出的每个文卷名必须在同一程序的 **FILE-CONTROL** 段中指出。

(2)文卷名 1 和文卷名 2 不可引用外部文卷连接符。

(3)**SORT** 和 **SORT-MERGE** 是等价的。

(4)除非使用 **SORT**,**SORT-MERGE** 或 **RECORD** 短语,否则,表示排序或合并文卷的名不能出现在 **SAME** 子句中。

(5)在一个程序中可以出现多个 **SAME** 子句,然而:

a. 一个文卷名一定不能出现在两个或两个以上的 **SAME RECORE AREA** 子句中。

b. 表示排序或合并文卷的文卷名不能出现在多于一个 **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句中。

c. 如果在 **SAME** 子句和一个或若干个 **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句中出现了不是表示排序或合并文卷的文卷名,则 **SAME** 子句中命名的所有文卷名都必须在那个

SAME SORT AREA 或 **SAME SORT-MERGE AREA** 子句中命名。

(6)在 **SAME SORT AREA** ,**SAME SORT-MERGE AREA** 或 **SAME RECORD AREA** 子句中引用的文卷名不要求有相同的组织和存取方式。

2.5.4 一般规则

(1)**SAME RECORD AREA** 子句指出文卷名 1、文卷名 2 引用的两个或多个文卷在处理当前逻辑记录中共享同一个存储区。所有这些文卷可以同时处于打开方式。在 **SAME RECORD AREA** 中的逻辑记录被看成是在 **SAME RECORD AREA** 子句中出现的每个被打开的输出文卷的逻辑记录和 **SAME RECORD AREA** 子句中出现的最近读入的输入文卷的逻辑记录。这等价于隐含的存储区重定义,即记录对齐于最左字符位置。

(2)如果使用 **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句,那么这些文卷名中至少有一个是表示排序或合并文卷的。该子句指出存储区按以下规定共享:

a. **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句指出一个存储区,它对于每一个已命名的排序或合并文卷在排序或合并时是可用的。因此为排序或合并文卷的排序或合并分配的存储区,在任一其它排序或合并文卷的排序或合并时也是可以重新使用的。

b. 此外,分配给非排序或合并的文卷的存储区,也可以根据排序或合并的需要分配给在 **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句中命名的排序或合并文卷。这种分配的范围将由实现者指出。

c. 非排序或合并文卷彼此不共享同一个存储区。但是如果用户希望这些文卷彼此共享同一个存储区的话,程序必须包含 **SAME AREA** 或 **SAME RECORD AREA** 子句,指出与这些文卷相关联的文卷名。

d. 在引用该子句中命名的排序或合并文卷的 **SORT** 或 **MERGE** 语句执行期间,在这个子句中命名的任何非排序或合并文卷不能被打开。

3 排序-合并模块的数据部

3.1 文卷节

文卷节位于源程序的数据部。文卷节定义排序和合并文卷的结构。每个排序文卷或合并卷通过一个排序合并文卷描述款以及一个或多个记录描述款来定义。记录描述款紧接在排序合并文卷描述款之后书写。

下面给出排序-合并模块中文卷节的一般格式。

FILE SECTION.

[排序合并文卷描述款

{记录描述款}...]

3.1.1 排序-合并文卷描述款

在 **COBOL** 程序中,排序合并文卷描述款(**SD** 款)表示文卷节中最高层组织。文卷节首后面跟由层指示符(**SD**)、文卷名及一系列独立子句组成的排序合并文卷描述款。排序合并文卷描述款(**SD** 款)的子句规定与排序文卷或合并文卷相关联的数据记录拿按管没有用户可以控制的标号过程,分块规则和内部存储规则仅由 **SORT** 和 **MERGE** 语句所专用。排序合并文卷描述款用句点结束。

3.1.2 记录描述结构

记录描述由描述特定记录之特性的一组数据描述款组成。每个数据描述款由层号跟以数据名或 **FILLER** 子句(若指出的话),在需要时再跟以一系列独立的子句。记录描述可以有层次结构,因此使用某一款的子句可以颇为不同,这依赖于其后是否跟以从属款。数据描述款中特定子句的可用性依赖于由实现支撑的核心模块的级。

3.1.3 初值

在文卷节中数据项的初值是无定义的。

3.2 排序-合并文卷描述款

3.2.1 功能

排序-合并文卷描述款提供关于排序或合并文卷的物理结构和记录名信息。

3.2.2 一般格式

FD 文卷名 1

$$\left[\text{RECORD} \left\{ \begin{array}{l} \text{CONTAINS 整数 1 CHARACTERS} \\ \text{IS VARYING IN SIZE} [[\text{FROM 整数 2}][\text{TO 整数 3}]\text{CHARACTERS}] \\ \text{[DEPENDING ON 数据名 1]} \\ \text{CONTAINS 整数 4 TO 整数 5 CHARACTERS} \end{array} \right\} \right]$$

$$\left[\text{DATA} \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \{ \text{数据名 2} \} \dots \right].$$

3.2.3 语法规则

- (1) 层指示符 **SD** 表明排序-合并文卷描述的开始,并且必须位于文卷名 1 前面。
- (2) 跟在文卷名 1 后面的各子句是任选的,而且出现的次序是无关紧要的。
- (3) 排序或合并文卷描述款后面可以跟一个或多个记录描述款。然而,不能对这种文卷执行输入-输出语句。
- (4) **RECORD** 子句的格式 2 的可用性依赖于实现支撑的顺序 I-O 模块的级。

3.2.4 一般规则

(1) 排序-合并模块的 **DATA RECORDS** 子句与顺序 I-O 模块的 **DATA RECORDS** 子句相同。因此 **DATA RECORDS** 子句的说明见顺序 I-O 的 3.5。在标准 COBOL 的这一版本中视 **DATA RECORDS** 子句是过时成分,因为在标准 COBOL 的以后的修改版中要把它删掉。

(2) 排序-合并模块的 **RECORD** 子句与顺序 I-O 模块的 **RECORD** 子句相同。因此 **RECORD** 子句的说明见顺序 I-O 的 3.8。

4 排序-合并模块的过程部

4.1 MERGE 语句

4.1.1 功能

MERGE 语句按照一组指定的键,把两个或多个有相同顺序的文卷合并在一起。在这一处理期间,**MERGE** 语句按合并次序使记录对输出过程或输出文卷可用。

4.1.2 一般格式

$$\text{MERGE 文卷名 1} \left\{ \text{ON} \left\{ \begin{array}{l} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \text{KEY} \{ \text{数据名 1} \} \dots \right\} \dots$$

$$[\text{COLLATING SEQUENCE IS 字母表名 1}]$$

$$\text{USING 文卷名 2} \{ \text{文卷名 3} \} \dots$$

$$\left\{ \begin{array}{l} \text{OUTPUT PROCEDURE IS 过程名 1} \\ \text{GIVING(文卷名 4)} \dots \end{array} \right\} \left[\left\{ \begin{array}{l} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{过程名 2} \right]$$

4.1.3 语法规则

- (1) **MERGE** 语句可以在过程部中除申述部分以外的任何地方出现。
- (2) 文卷名 1 必须在数据部的排序合并文卷描述款中被描述。
- (3) 若文卷名 1 引用的文卷包含变长记录,则文卷名 2 及文卷名 3 引用的文卷中所含记录的长度不能小于文卷名 1 描述的最小记录也不能大于文卷名 1 描述的最大记录。若文卷名 1 引用的文卷含有定长记录,则文卷名 2 及文卷名 3 引用的文卷中所含记录的长度不能大于文卷名 1 描述的最大记录。

(4) 数据名 1 是键数据名并且遵守下列规则：

- a. 由键数据名标识的数据项必须在文卷名 1 相关联的记录中描述。
- b. 键数据名可以受限。
- c. 键数据名标识的数据项不能是包含变长数据项的组项。
- d. 如果文卷名 1 有多个记录描述,那么要求由键数据名指出的数据项仅能在一个记录描述中描述。把在一个记录描述款中,键数据名引用的相同字符位置作为该文卷所有记录的键。
- e. 由键数据名指出的那些数据项不可以在包含有 **OCCURS** 子句或隶属于包含有 **OCCURS** 子句的款中描述。
- f. 若文卷名 1 引用的文卷含有变长记录,则由键数据名标识的全部数据项必须被包含在记录的开始 **X** 个字符位置中,这里 **X** 等于文卷名 1 引用的文卷所规定的最小记录长度。

(5) 文卷名 2、文卷名 3 和文卷名 4 必须在数据部的文卷描述款,而不在排序合并描述款中描述。

(6) 多文卷带卷中的文卷名最多只能有一个出现在 **MERGE** 语句中。

(7) **MERGE** 语句中同一文卷名不可以重复出现。

(8) 在 **MERGE** 语句中没有一对文卷名可以被规定在同一 **SAME AREA**、**SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句中。**MERGE** 语句中能被规定在同一 **SAME RECORD AREA** 子句中的仅是那些与 **GIVING** 短语相关联的文卷(见 GB/T 4092. 4 中 2. 13 **SAME** 子句和 2. 5 **SAME RECORD/SORT/SORT-MERGE AREA** 子句)。

(9) 字 **THRU** 和 **THROUGH** 是等价的。

(10) 若文卷名 4 引用索引文卷,则数据名 1 的第一个说明必须与 **ASCENDING** 短语有关,并且由数据名 1 引用的数据项在记录中占有的字符位置必须与那个文卷的主记录键相关的数据项相同。

(11) 若指定了 **GIVING** 短语,且文卷名 4 引用的文卷含有变长记录,则文卷名 1 引用的文卷中所含的记录长度不能小于文卷名 4 描述的最小记录,也不能大于其最大记录的长度。若文卷名 4 引用的文卷包含定长记录,则文卷名 1 引用的文卷中所含的记录长度不能大于文卷名 4 描述的最大记录长度。

4. 1. 4 一般规则

(1) **MERGE** 语句将合并所有包含在文卷名 2 和文卷名 3 引用的文卷中的记录。

(2) 若文卷名 1 引用的文卷仅含定长记录,则对文卷名 2 或文卷名 3 引用的文卷中,任何长度小于那个定长的记录,当记录被发送到文卷名 1 引用的文卷上时,在那个记录的最后一个字符之后,从其后右边第一字符位置开始填以空格,空格的数目就是比定长少的那些字符位置数。

(3) 跟在字 **KEY** 后面的数据名,在 **MERGE** 语句中按其重要性的递降次序从左到右地排列而不管它们是如何被划分到各 **KEY** 短语中去的。最左边的数据名是主键,下个数据名是下一个重要的键,依此类推。

a. 当指出 **ASCENDING** 短语时,合并顺序是键数据名指出的数据项内容从最低值到最高值的顺序,这根据关系条件中运算对象比较的规则进行。

b. 当指出 **DESCENDING** 短语时,合并顺序是键数据名指出的数据项内容从最高值到最低值的顺序,这根据关系条件中运算对象的比较规则进行。

(4) 按照关系条件中运算对象的比较规则,当一个数据记录的所有键数据项的内容等于一个或多个其它数据记录的相应键数据项的内容时,这些记录的回送次序是：

a. 按照 **MERGE** 语句中指出的相关输入文卷的次序进行。

b. 与一个输入文卷相关的所有记录在另一个输入文卷的记录回送之前被回送。

(5) 用于非数值键数据项比较的对比序列在 **MERGE** 语句执行开始时,按下列优先次序决定：

a. 首先,如果 **MERGE** 语句中指出 **COLLATING SEQUENCE** 短语的话,就根据 **COLLATING SEQUENCE** 短语建立对比序列。

b. 其次,建立和程序的对比序列一样的对比序列。

(6) 仅当文卷名 2 和文卷名 3 所引用之文卷的记录按 **MERGE** 语句中 **ASCENDING** 或 **DESCENDING KEY** 短语的规定排序时,合并操作的结果才是可预测的。

(7) 文卷名 2 和文卷名 3 引用的文卷中的全部记录被传送到文卷名 1 引用的文卷中。在执行 **MERGE** 语句的开始时,文卷名 2 和文卷名 3 引用的文卷不能处于打开状态。对文卷名 2 和文卷名 3 引用的每一个文卷,**MERGE** 语句的执行引起下述动作:

a. 初始化文卷处理。执行初始化就象执行了具有 **INPUT** 短语的 **OPEN** 语句一样。若输出过程被指出,则在控制转到输出过程之前执行这个初始化。

b. 逻辑记录被获取并被发放给合并操作。获取每个记录就象执行了具有 **NEXT** 和 **AT END** 短语的 **READ** 语句一样。

c. 终止文卷处理。执行终止就象执行了没有任选短语的 **CLOSE** 语句一样。若输出过程被指出,则直到控制通过输出过程的最后一个语句之后才执行这个终止。

执行这些隐式功能,犹如执行相关的 **USE AFTER EXCEPTION** 或 **ERROR** 过程一样。

(8) 输出过程可以由这样一些过程组成,这些过程是从文卷名 1 引用的文卷上选择、修改或复制记录所必需的,**RETURN** 语句依合并次序一次可使一个记录成为可用的。范围包括:在输出过程范围内由执行 **CALL**、**EXIT**、**GO TO** 和 **PERFORM** 语句引起控制转移的全部语句;还有申述过程中的全部语句,这些语句由输出过程范围中语句的执行引起其执行。

输出过程的范围内不能导致任何 **MERGE**、**RELEASE** 或 **SORT** 语句执行。

(9) 如果指出输出过程,则在 **MERGE** 语句执行期间转向它。编译程序在输出过程最后一句的结束处插入返回机制。在控制到达输出过程的最后一个语句时,返回机制终止合并处理然后把控制转移到 **MERGE** 语句后的下一个可执行语句。在进入输出过程之前,合并过程到达这样的一点,在这一点处可以按合并次序选取需要的记录。在输出过程中为了取得下一个记录,**RETURN** 语句是必需的。

(10) 输出过程的执行期间,不能执行对文卷名 2 或文卷名 3 引用的文卷进行管理或访问其有关的记录区的语句。在执行 **MERGE** 语句时隐式引用的任何 **USE AFTER EXCEPTION** 过程的执行期间,不能执行对文卷名 2、文卷名 3 或文卷名 4 引用的文卷进行管理,或访问其有关的记录区的语句。

(11) 若 **GIVING** 短语被指出,则所有已合并的记录书写到文卷名 4 引用的文件上,如同 **MERGE** 语句所隐含的输出过程一样。在执行 **MERGE** 语句的开始处,文卷名 4 引用的文卷不能处于打开状态。对文卷名 4 引用的每一个文卷,**MERGE** 语句的执行导致下述动作:

a. 初始化文卷处理。执行初始化就象执行了具有 **OUTPUT** 短语的 **OPEN** 语句。

b. 已合并的逻辑记录被回送并被写到该文卷。写出每个记录就象执行了不带任何任选短语的 **WRITE** 语句。

对于相对文卷,回送的第一个记录的相对键数据项值为“1”;回送的第二个记录,值为“2”,等等。**MERGE** 语句执行以后,相对键数据项的内容指示回送到该文卷的最后记录。

c. 终止文卷处理。执行终止就象执行了不带任选短语的 **CLOSE** 语句。

执行这些隐含功能,犹如执行相关的 **USE AFTER EXCEPTION** 或 **ERROR** 过程一样;然而,执行这种 **USE** 过程不能引起管理由文卷名 4 引用的文卷或访问与其相关的记录区的语句的执行。在第一次企图往文卷定义的边界之外写记录时,就执行对该文卷规定的任一 **USE AFTER STANDARD EXCEPTION** 或 **ERROR** 过程;当控制从那个 **USE** 过程返回或若没指出这种 **USE** 过程时,如同上述 11C 段一样,终止对文卷的处理。

(12) 若文卷名 4 引用的文卷只含定长记录,则对文卷名 1 引用的文卷中任一长度小于那个定长的记录,当记录被回送到文卷名 4 引用的文卷上时,在那个记录的最后一个字符之后,其右边第一字符位置开始填以空格,空格的数目就是比定长少的那些字符位置数。

(13) 程序分段可以应用于包含 **MERGE** 语句的程序。然而要应用如下限制:

a. 如果 **MERGE** 语句不是在独立程序段中出现,那么 **MERGE** 语句引用的任何输出过程必须:

- 1) 全部出现在非独立的程序段中,或
- 2) 全部包含在一个独立的程序段中。
- b. 如果 **MERGE** 语句出现在独立的程序段中,那么 **MERGE** 语句引用的任何输出过程必须:
 - 1) 全部包含在非独立程序段中,或
 - 2) 全部包含在 **MERGE** 语句所在的同一个独立程序段中。

4.2 **RELEASE** 语句

4.2.1 功能

RELEASE 语句传送记录给 **SORT** 操作的初始阶段。

4.2.2 一般格式

RELEASE 记录名 1 [**FROM** 标识符 1]

4.2.3 语法规则

- (1) 记录名 1 必须是排序合并文卷描述款中的逻辑记录名,并且可以受限。
- (2) **RELEASE** 语句只能在与某一个文卷的 **SORT** 语句相关联的输入过程范围内使用,该文卷的排序合并文卷描述款包含有记录名 1。
- (3) 记录名 1 和标识符 1 不能引用同一个存储区。

4.2.4 一般规则

- (1) **RELEASE** 语句的执行使得记录名 1 指称的记录被发放给排序操作的初始阶段。
- (2) **RELEASE** 语句执行之后,除非与记录名 1 有关的排序合并文卷已在 **SAME RECORD AREA** 子句中指出,否则记录区中的逻辑记录不再可用。和输出文卷有关的在同一个 **SAME RECORD AREA** 子句中引用的其它文卷的记录,以及记录名 1 相关的文卷对程序都还是可用的。
- (3) 具有 **FROM** 短语的 **RELEASE** 语句的执行结果与按指明的顺序执行下述语句的结果等价:
 - a. 按照对 **MOVE** 语句所指明的规则执行:

MOVE 标识符 1 **TO** 记录名 1
 - b. 不带 **FROM** 短语的同一 **RELEASE** 语句。
- (4) **RELEASE** 语句的执行完成以后,与标识符 1 相关联的数据区的内容是可用的。记录名 1 引用的数据区中的信息已不再可用,用 **SAME RECORD AREA** 子句规定时为例外。

4.3 **RETURN** 语句

4.3.1 功能

RETURN 语句从 **SORT** 操作的最后阶段获得排序好的记录或在 **MERGE** 操作期间获得合并好的记录。

4.3.2 一般格式

RETURN 文卷名 1 **RECORD** [**INTO** 标识符 1]

AT END 命令语句 1

[**NOT AT END** 命令语句 2]

[**END-RETURN**]

4.3.3 语法规则

- (1) 跟标识符 1 相关联的存储区和跟文卷名 1 相关联的记录区不能是同一个存储区。
- (2) 文卷名 1 必须在数据部的排序合并文卷描述款中描述。
- (3) **RETURN** 语句只能在和 **SORT** 或 **MERGE** 语句相关联的输出过程范围内对文卷名 1 使用。

4.3.4 一般规则

(1) 当一个文卷的逻辑记录用一个以上的记录描述来描述时,这些记录自动地共享同一个存储区,这等价于隐含的区重定义。任何落在当前数据记录范围之外的数据项的内容,在 **RETURN** 语句执行完成后是无定义的。

(2) **RETURN** 语句的执行使得文卷名 1 的下一个现存记录在与文卷名 1 相关联的记录区里成为可用的。这下一个记录根据 **SORT** 或 **MERGE** 语句中列出的键指出的次序来决定。若文卷名 1 不存在下一个逻辑记录,则出现末端条件并把控制转到 **AT END** 短语的命令语句 1。按照命令语句 1 中指出的各语句的规则继续执行。若执行了过程分支或显式引起控制转移的条件语句,则按照那个语句的规则执行控制转移;否则,在命令语句 1 执行完成后,控制转向 **RETURN** 语句的结尾处并且忽略 **NOT AT END** 短语(若被指出)。当末端条件出现时,**RETURN** 语句的执行是不成功的而且与文卷名 1 相关联的记录区的内容是无定义的。在 **AT END** 短语中的命令语句 1 执行以后,**RETURN** 语句不能再作为当前输出过程中的一部分执行。

(3) 若在 **RETURN** 语句执行期间,未出现末端条件,则记录是可利用的并且根据 **INTO** 短语的存在情况执行隐含的 **MOVE** 操作,控制转向命令语句 2(若指出);否则,控制转向 **RETURN** 语句的结尾处。

(4) **END-RETURN** 短语限定 **RETURN** 语句的作用域(见预备知识 6.6.4.3 语句的作用域)。

(5) **RETURN** 语句中可以规定 **INTO** 短语:

- a. 仅当一个记录描述从属于排序合并文卷描述款时,或
- b. 与文卷名 1 相关联的全部记录名及标识符 1 引用的数据项描述组项或初等字符项时。

(6) 具有 **INTO** 短语的 **RETURN** 语句的执行结果等价于按序应用下述规则:

- a. 不带 **INTO** 短语的同一 **RETURN** 语句的执行。
- b. 按照不带 **CORRESPONDING** 短语的 **MOVE** 语句的规则把当前记录从记录区送到标识符 1 指定地区。根据 **RECORD** 子句指出的规则确定当前记录的长度。若文卷描述款包含 **RECORD IS VARYING** 子句,则隐含的移动是成组移动。当 **RETURN** 语句执行失败时,不出现隐含的 **MOVE** 语句。与标识符 1 相关联的任何下标在记录被读以后和传送到数据项之前立即计算。在记录区和标识符 1 引用的数据项里的数据均是可用的。

4.4 SORT 语句

4.4.1 功能

SORT 语句通过执行输入过程或传送另一个文卷的记录以建立排序文卷,在排序文卷中按照指出的一组键对记录排序。然后,在排序操作的最后阶段,使得排序文卷中的每一个记录按照排序的次序,对某些输出过程或输出文卷可用。

4.4.2 一般格式

$$\begin{aligned} & \text{SORT 文卷名 1} \left\{ \text{ON} \left\{ \begin{array}{c} \text{ASCENDING} \\ \text{DESCENDING} \end{array} \right\} \text{KEY} \{ \text{数据名 1} \} \dots \right\} \dots \\ & \quad [\text{WITH DUPLICATES IN ORDER}] \\ & \quad [\text{COLLATING SEQUENCE IS 字母表名 1}] \\ & \quad \left\{ \begin{array}{l} \text{INPUT PROCEDURE IS 过程名 1} \left[\left\{ \begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{过程名 2} \right] \\ \text{USING} [\text{文卷名 2}] \dots \end{array} \right\} \\ & \quad \left\{ \begin{array}{l} \text{OUTPUT PROCEDURE IS 过程名 3} \left[\left\{ \begin{array}{c} \text{THROUGH} \\ \text{THRU} \end{array} \right\} \text{过程名 4} \right] \\ \text{GIVING} [\text{文卷名 3}] \dots \end{array} \right\} \end{aligned}$$

4.4.3 语法规则:

- (1) **SORT** 语句可以在过程部中除申述部分以外的任何地方出现。
- (2) 文卷名 1 必须在数据部排序合并文卷描述款中描述。
- (3) 若指明 **USING** 短语,而且文卷名 1 引用的文卷包含变长记录,则文卷名 2 引用之文卷所包含记录的长度不能小于文卷名 1 描述的最小记录,也不能大于其最大记录的长度。若文卷名 1 引用的文卷

包含定长记录,则文卷名 2 引用之文卷所含记录的长度不能大于文卷名 1 引用文卷描述的最大记录长度。

(4) 数据名 1 是键数据名,并且要遵守下列规则:

- a. 由键数据名标识的数据项必须在相关联的文卷名 1 的记录中描述。
- b. 键数据名可以受限。
- c. 由键数据名标识的数据项不能是包含变长数据项的组项。
- d. 如果文卷名 1 有多个记录描述,则由键数据名标识的那些数据项只在唯一的一个记录描述中描述。把在一个记录描述中,键数据名引用的相同字符位置作为该文卷所有记录的键。
- e. 由键数据名标识的数据项不能在包含 **OCCURS** 子句的描述款中,或隶属于包含 **OCCURS** 子句的描述款中描述。
- f. 若文卷名 1 引用的文卷包含定长记录,那么由键数据名标识的全部数据项必须被包含在记录的开始 **X** 个字符位置中,这里 **X** 等于文卷名 1 引用的文卷所指明的最小记录长度。

(5) 字 **THRU** 和 **THROUGH** 是等价的。

(6) 文卷名 2 和文卷名 3 必须在数据部的文卷描述款,而不在排序合并文卷描述款中描述。

(7) 文卷名 2 和文卷名 3 引用的文卷可以驻留在同一个多文卷卷上。

(8) 若文卷名 3 引用了索引文卷,则数据名 1 的第一个说明必须与 **ASCENDING** 短语相关,并且由数据名 1 引用的数据项在记录中占有的字符位置必须与那个文件的主记录键相关的数据项一样。

(9) 在同一 **SORT** 语句中,没有一对文卷名可以被规定在同一个 **SAME SORT AREA** 或 **SAME SORT-MERGE AREA** 子句中。与 **GIVING** 短语相关联的文卷名不能在同一 **SAME** 子句中指出(见 GB/T 4092. 4 中 2. 13 **SAME** 子句和 2. 5 **SAME RECORD/SORT/SORT-MERGE AREA** 子句)。

(10) 若指明了 **GIVING** 短语,且文卷名 3 引用的文卷包含变长记录,则文卷名 1 引用之文卷所含记录的长度不能小于文卷名 3 描述的最小记录,也不能大于其最大记录的长度。若文卷名 3 引用的文卷含有定长记录,则文卷名 1 引用文卷所含记录的长度不能大于文卷名 3 引用的文卷的最大记录长度。

4. 4. 4 一般规则

(1) 若文卷名 1 引用的文卷仅含定长记录,则对文卷名 2 引用的文卷中任何长度小于那个定长的记录,当记录被发放到文卷名 1 引用的文卷上时,在那个记录的最后一个字符之后,其右边第一字符位置开始填以空格,空格的数目就是比定长少的那些字符位置数。

(2) 在 **SROT** 语句中,跟在字 **KEY** 后面的数据名,按照重要程序的递减次序从左到右地列出,而不考虑它们如何被划分到各 **KEY** 短语中去。最左边的数据名是主键,下一个数据名是下一个最重要的键,依此类推。

a. 当指出 **ASCENDING** 短语时,排序顺序将是按键数据名指出的数据项内容的最低值到最高值顺序。这依据关系条件中的运算对象比较规则进行。

b. 当指出 **DESCENDING** 短语时,排序顺序将是按键数据名指出的数据项内容的最高值到最低值顺序,这依据关系条件中的运算对象比较规则进行。

(3) 若指出 **DUPLICATES** 短语,并且与一个数据记录相关的所有键数据项的内容等于与一个或多个其它数据记录相关的相应键数据项的内容,则这些记录的回送次序是:

a. 按照 **SORT** 语句中指明的相关输入文卷的次序。在给定的输入文卷中,按对该文卷存取记录的次序。

b. 当输入过程被指出时,按照输入过程发送这些记录的次序。

(4) 若未指出 **DUPLICATES** 短语,而与一个数据记录相关的所有键数据项的内容等于与一个或多个其它数据记录相关的相应键数据项的内容,则这些记录的回送次序是未定义的。

(5) 应用于非数值键数据项比较的对比序列在执行 **SORT** 语句的开始时按下列优先次序决定:

a. 首先,如果在 **SORT** 语句中指出 **COLLATING SEQUENCE** 短语,则根据 **COLLATING**

SEQUENCE 短语建立的对比序列。

b. 其次,建立和程序的对比序列一样的对比序列。

(6) SORT 语句的执行由下面三个不同的阶段组成:

a. 使记录对文卷名 1 引用的文卷成为可用的。为达此目的,或通过执行输入过程中的 **RELEASE** 语句或通过隐含执行对文卷名 2 的 **READ** 语句。在本阶段开始时,文卷名 2 引用的文卷不能处于打开方式。在本阶段终止时,文卷名 2 引用的文卷不是处于打开方式。

b. 文卷名 1 引用的文卷是顺序的。在这阶段期间,对文卷名 2 和文卷名 3 引用的文卷不作处理。

c. 文卷名 1 引用的文卷记录可按排序好的次序应用。排序好的记录或者写到文卷名 3 引用的文卷中,或者通过执行 **RETURN** 语句使记录成为可用的以便输出过程处理。在本阶段开始时,文卷名 3 引用的文卷不能处于打开方式。在本阶段终止时,文卷名 3 引用的文卷不是处于打开方式。

(7) 输入过程可以由这样一些过程组成,这些过程是从文卷名 1 引用的文件上选择、修改或复制记录所必需的,**RELEASE** 语句一次使一个这样的记录成为可用的。范围包括:在输入过程范围内由执行 **CALL**、**EXIT**、**GO TO** 和 **PERFORM** 语句引起控制转移的全部语句;还有申述过程中的全部语句,这些语句由输入过程范围中语句的执行引起其执行。

输入过程的范围不能导致任何 **MERGE**、**RETURN** 或 **SORT** 语句执行。

(8) 如果指出输入过程,则在文卷名 1 由 **SORT** 语句排序之前。控制转到输入过程。编译程序在输入过程的最后语句的末端插入返回机制,当控制达到输入过程中的最后一个语句时,被释放到文卷名 1 上的记录已排序。

(9) 若指出 **USING** 短语,由文卷名 2 引用的诸文卷的全部记录被送到文卷名 1 引用的文卷中,对文卷名 2 引用的每个文卷,执行 **SORT** 语句导致下列动作:

a. 初始化文卷处理。执行初始化就象执行具有 **INPUT** 短语的 **OPEN** 语句一样。

b. 逻辑记录被获取并发送给排序操作。获取每个记录就象执行了具有 **NEXT** 和 **AT END** 短语的 **READ** 语句一样。

对相对文卷,若文卷名 2 在 **GIVING** 短语中被引用,则在 **SORT** 语句执行以后,相对键数据项的内容是无定义的。

c. 终止文卷处理。执行终止就象执行了不带任选短语的 **CLOSE** 语句一样。在文卷名 1 引用的文卷被 **SORT** 语句排序之前执行这个终止。

执行这些隐含功能,犹如执行相关的 **USE AFTER EXCEPTION** 或 **ERROR** 过程一样;然而,执行这种 **USE** 过程不能引起管理由文卷名 2 引用的文卷或访问与其相关的记录区的语句的执行。

(10) 输出过程可以由这样一些过程组成,这些过程是从文卷名 1 引用的文件上选择、修改或复制记录所必需的,**RETURN** 语句一次可使一个记录成为可用的。范围包括:在输出过程范围内由执行 **CALL**、**EXIT**、**GO TO** 和 **PERFORM** 语句引起控制转移的全部语句;还有申述过程中的全部语句,这些语句由输出过程范围中语句的执行引起其执行。

输出过程的范围不能导致任何 **MERGE**、**RELEASE** 或 **SORT** 语句的执行。

(11) 如果指出输出过程,则文卷名 1 引用的文卷被 **SORT** 语句排序以后控制转向它。编译程序在输出过程最后语句的末端插入一个返回机制,当控制转移到输出过程的最后一个语句时,返回机制就结束排序,然后控制转移到 **SORT** 语句后的下一个可执行语句。在进入输出过程之前,排序过程到达这样的一点,在这一点处可以按排序次序选取需要的记录。在输出过程中为了取得下一个记录,**RETURN** 语句是必需的。

(12) 如果指出 **GIVING** 短语,在文卷名 1 上的所有排序过的记录自动地被写到文卷名 3 上去,犹如 **SORT** 语句隐含的输出过程一样。对文卷名 3 引用的每个文卷,**SORT** 语句的执行导致下述动作:

a. 初始化文卷处理。执行初始化就象执行了具有 **OUTPUT** 短语的 **OPEN** 语句。在任一输入过程执行以后,执行这个初始化。

b. 排序好的逻辑记录被回送并写往该文卷。写记录就象执行了不带任何任选短语的 **WRITE** 语句。

对相对文卷,回送的第一个记录的相对键数据项值为“1”;回送的第二个记录,值为“2”,等等。**SORT** 语句执行以后,相对键数据项的内容指出回送该文卷的最后记录。

c. 终止文卷处理。执行终止就象执行了不带任选短语的 **CLOSE** 语句。

执行这些隐含功能,犹如执行了相关的 **USE AFTER EXCEPTION** 或 **ERROR** 过程一样;然而,执行这种 **USE** 过程不能引起管理由文卷名 3 引用的文卷或访问其相关的记录区的语句的执行。在第一次企图往文卷定义的边界之外写记录时,就执行对该文卷规定的任一 **USE AFTER STANDARD EXCEPTION** 或 **ERROR** 过程,当控制从 **USE** 过程返回或若没指出这种 **USE** 过程时,如同上述 12C 段一样,终止文卷处理。

(13) 若文卷名 3 引用的文卷只含定长记录,则对文卷名 1 引用的文卷中任一长度小于那个长度的记录,当记录被回送到文卷名 3 引用的文卷上时,在那个记录的最后一个字符之后,其右边第一字符位置开始填以空格,空格的数目就是比定长少的那些字符位置数。

(14) 程序分段可以应用于含有 **SORT** 语句的程序。然而有下列限制;

a. 如果 **SORT** 语句出现在非独立程序段的节中。那么 **SORT** 语句引用的任何输入过程或输出过程必须:

- 1) 全部出现在非独立程序段中,或
- 2) 全部包含在一个独立程序段中。

b. 如果 **SORT** 语句出现在一独立程序段中,那么由 **SORT** 语句引用的任何输入过程或输出过程必须:

- 1) 全部包含在非独立程序段中,或
- 2) 全部包含在 **SORT** 语句所在的同一个独立程序段中。

附加说明:

本标准由中华人民共和国机械电子工业部提出。

本标准由南京大学负责起草。

本标准主要起草人钱树人、王静英、冯惠、段祥。

本标准由 1983 年 12 月首次发布,1992 年 8 月第一次修订。