1。VSAM 基本概念

虚拟存储技术的发展，为文件管理系统开拓了新的方向，而基于虚拟存储概念而研制发展的虚拟存取方法，在 IBM 系列的机器中也已经普遍地使用。VSAM（Virtual Storage Access Method）是一种虚拟存取方法，它是为了与直接存取存储设备 DASD（Direct Access Storage Device--能够在文件上直接地存取任何记录的设备）一起使用而研制的文件管理系统。

VSAM 是把用户的逻辑数据（应用处理程序）与辅助存储器中的物理数据相连接，从而为程序员在数据管理中提供方便，程序员可根据不同的需要选择不同的数据组织。

VSAM 存取文件记录的方法将不依赖于存放记录的外部设备类型，而是通过这个记录对文件开始点的相对位移（相对位移以字节为单位计算）来访问记录。相对位移值就称之为相对字节地址 RBA（Relative Byte Address）。文件开始点地址定为 0。

VSAM 方法允许使用三种不同的数据组织，每一种数据组织均可采用不同的处理方法。包括了对在 DASD 上文件的自动空间分配、产生替换索引等功能。

VSAM 方法还有一组服务性的程序--------存取方法服务 AMS（Access Method Service）。这组服务性程序可以定义和维护 VSAM 文件，把记录输入到 VSAM 文件中、为文件建立一个或多个替换索引、复制和打印文件、产生文件的副本、恢复失效的数据、把顺序存取方法(SAM）文件和索引顺序存取方法（ISAM）文件转换成 VSAM 格式、并且能对已转换成 VSAM 格式的 ISAM 文件进行处理。除此之外，VSAM 方法还具有如下的特征：

1。自动的数据空间分配

VSAM 的数据空间全是通过独立的目录来管理的。该目录描述了在系统中的全部 VSAM 文件的逻辑属性和物理属性。用于 VSAM 的数据空间是动态的，如需要则可增加更多的空间。

2。数据保护

VSAM 方法利用它本身的设计和存取控制参数，可以对数据进行保护。VSAM 的设计允许用户仅通过指定正确的目录信息来存取数据，目录本身指向数据，在目录中所存储的信息都受到 VSAM 所限制。

3。设备独立性

VSAM 方法允许在不同类型的磁盘设备上处理，而不必重新进行程序设计。事实上，在逻辑上 VSAM 文件被设计成连续的区域，唯一的限制是：VSAM 文件必须全部存储在同样的设备类型的卷中。

4。数据移植性

VSAM 方法提供了在不同操作系统下 VSAM 文件互换的可能性。

2。VSAM 的数据组织

所谓的文件组织形式，是指数据记录在文件中的排列方式。而文件的存取方法是指从文件找到数据记录的方法。VSAM 所使用的数据，均具有一定的组织结构以及存取方法，用户可以选择三种类型的数据组织及其相应的存取方法：

1。键顺序数据组织（Indexed Sequential Organization）

2。进入顺序数据组织（Sequential Organization）

3。相对记录数据组织（Relative Organization）

这三种数据组织所对应的数据集就分别称为：

1。键顺序数据集 KSDS（Key Squenced Data Set）

2。进入顺序数据集 ESDS（Entry Squenced Data Set）

3。相对记录数据集 RRDS（Relative Record Data Set）

2。1。键顺序数据组织

2。1。1。组织形式

在键顺序数据组织中，逻辑记录根据排序序列来存储，起排序序列由记录的主键内容决定，增加新的逻辑记录以及删除原有的逻辑记录时，整个文件根据键的排序序列而限制在序列之中。

键顺序数据组织基本上与 ISAM 文件的组织相似，但 VSAM 并不使用溢出区。

KSDS 文件优于 ISAM 文件是由于：在一定的范围内，文件是自我重新组织的，因而平均探索时间实际上是一个常数。

2。1。2。存取方法

键顺序数据组织允许四种类型的处理：

1。键控直接处理（根据主键对单个逻辑记录的处理）。

2。键控顺序处理（在逻辑序列中根据主键对一系列逻辑记录的处理）。

3。直接访问处理（根据在文件中的位置，对单个逻辑记录的处理）。

4。顺序访问处理（在物理序列中，根据在文件中的位置，对一系列逻辑记录的处理）

2。2。进入顺序数据组织

2。2。1。组织形式

在进入顺序数据组织中，在物理上，逻辑记录是以进入时的同样次序存储，新的逻辑记录存储在文件的末尾。这种数据组织基本上与 SAM 文件相似。

2。2。2。存取方法

进入顺序数据组织只允许两种处理类型：

1。直接访问处理（根据在文件中的位置，对单个逻辑记录的处理）。

2。顺序访问处理（在物理序列中，根据在文件中的位置，对一系列逻辑记录的处理）。

2。3。相对记录数据组织

2。3。1。组织形式

在相对记录数据组织中，逻辑记录是根据记录号、相对于文件的起始位置而存储的。相对记录文件基本上是固定长度的槽（SLOT），每一个槽都有一个相对记录号，从 1 开始。

2。3。2。存取方法

相对记录数据组织允许两种处理类型：

1。键控直接处理（根据主键对单个逻辑记录的处理）。

2。键控顺序处理（在逻辑序列中根据主键对一系列逻辑记录的处理）。

其中相对记录号总是作为键来处理。

3。VSAM 数据集

为了满足用户的需要，程序员可以选择不同的数据结构（数据集 / 文件）。

3。1。KSDS

与 ISAM 文件一样，KSDS 文件根据用户在每个记录中所定义的键字段作为次序，也就是文件中的记录根据在每个记录中的键字段的排序序列而定位，每个记录在键字段有唯一的一个值。

VSAM 使用与每个记录相结合的键，把记录插入到文件中，或者从文件中检索记录，记录的存取次序可以是随机的，也可以是顺序的。

VSAM 文件可以有多个索引。这就是指文件中的记录，既有主键，也有次级键（替换键），但最多能有 253 个次级键，可以是记录中的任何字段，但必须有固定的长度和位置。

替换键与主键一样具有同样的功能，而且，与主键相比，替换键的键值不必是唯一

的值，因此在应用处理中允许用户能充分利用其灵活性。

数据记录　　数据记录　。。。。。　　数据记录

KEY10　　KEY88　　KEY1000

根据数据记录的键顺序而组织的键顺序文件

3。2。ESDS

包含在文件中的记录，是以当时进入的先后顺序而存储在 ESDS 中，而且，这种进入顺序并不关心记录的内容，由于没有用键去标识该记录，因而没有建立主索引。但是，ESDS 可以定义一个或多个替换索引。记录的次序是固定的，不会移动的。因此，将不会通过文件分配自由空间，新记录的插入要放在文件的末端，同时也不能缩短、增长、删除记录，用户要访问这些记录时，必须按其原来写入记录的次序而顺序地访问文件中的记录。

所以，从本质上来说，ESDS 是顺序文件，与 SAM 文件的处理方法类似。

第三个记录

第一个记录　　第二个记录　　。。。。。

增加第三个记录到进入顺序文件中

3。3。RRDS

相对记录文件也没有索引，在其固定长度的槽串中，仅有其相对记录号。相对记录号从 1 到 N，其中 N 是能够存储在文件中最大的记录数。

每一记录占一个槽，并且根据槽的相对记录号而存储或检索记录，而记录的内容与进入的顺序无关。

在相对记录文件中的记录组成的控制区间中，正如它们进入顺序文件或键顺序文件一样

，每个控制区间包含相同数量的槽，每个槽的大小就是记录长度，由用户在文件初始化定义时指定。

(待续)

相对记录 4

相对记录 1　相对纪录 2　　相对记录 6

槽 1　　槽 2　　槽 3　　槽 4　　槽 5　　槽 6

将相对记录 4 插入到相对记录文件中

4。三种数据集的比较

通过上面的描述可知，VSAM 方法所用到的三种数据集（文件），存在着许多的不同之处。因此，在具体的使用中，应该建立哪一种文件更利于处理，就需要视具体的情况

而定。
下面是这三种文件的主要特性的比较：

类型 特征	KSDS	ESDS	RRDS
记录长度	定长或变长	定长或变长	定长
记录地址	可改变记录的 RBA	不可改变记录的 RBA	不可改变槽的相对记录号
记录位置	通过键字段而排序	按进入的物理顺序排序	按相对记录号排序
替换索引	可有一个或多个	可有一个或多个	没有
跨越记录	可有	可有	不可
存取方式	顺序或直接存取。 根据键或 RBA 直接存取。	顺序或直接存取。 除非建立了替换索引，否则只根据 RBA 直接存取。	顺序或直接存取。 根据 RBA（视为键）直接存取。
空间回收	可插入或删除记录。 可回收被删除记录的空间重用。	不可插入或删除记录。 但可用同等长度的记录置换而重用该空间。	可删除记录。 可插入同一相对记录号的新记录重用该空间。
自由空间	可使用文件的自由空间，用以增加记录或者改变记录的长度。	文件末端的空间可用来增加记录，但不能改变记录的长度。	文件中空的槽可用来增加 记录，但不能改变记录的长度。

5。VSAM 的物理结构与逻辑结构

5。1。控制区间（CI）

控制区间 CI（Control Interval）是 DASD 中连续的区域。在该区域内，VSAM 存储数据记录及描述这些数据记录的控制信息。

CI 是 VSAM 方法在虚存（Virtual Storage）和外存（DASD）之间传送数据信息的基本单位。每个 CI 由一个以上的定长或变长的逻辑记录、自由空间、及描述本 CI 数据存放和空间使用情况等控制信息所组成。

不同的文件其 CI 的长度可以不同。但在给定文件的每个 CI，都具有同样的长度，并且这个长度不能改变，对于 CI 长度的优化设计应该视文件性质而定。CI 的长度将取决于：

1。数据记录的最大长度

2。提供给 VSAM 的 I / O 缓冲区的虚存空间的数量

3。用于存放文件的 DASD 设备的类型

CI 的大小必须是 512 字节的倍数，若大于 4096 字节，则必须是 2048 字节的倍。

但不管选择哪一种数量级的倍数，每个 CI 的最大范围只能是 32768 个字节（这是因为最合适的长度应该是一个磁道的长度）。

通常，CI 包含完整的物理记录，VSAM 根据 CI 的大小而选择相应的物理记录的长度。对于一般的 DASD 设备，可接受的物理记录的长度是 512、1024、2048、4096 字节。对于已定的 CI，VSAM 能够使用到它的最大的物理记录。

例如：如果 CI 是 1024 字节，VSAM 可以使用 1024 字节的物理记录；

如果 CI 是 1536 字节，VSAM 只能把 512 字节作为物理记录的长度，这是因为物理长度必须是同一物理长度。

所以，对于任何已给定的文件，每个 CI 的物理记录的长度和数量，都是 VSAM 所决定的。

虽然，每个 CI 里的物理记录数量是固定的，但是逻辑记录的数量却是可以改变的。

而且，在 CI 内，物理记录与逻辑记录在数量上不存在着相互的关系。

选择 CI 的大小虽然要视乎其本身定义所在的 DASD 设备类型而定，却不受这些设备类型的限制。一些 CI 适合于某一种磁盘设备的磁道，但如果要把 VSAM 文件写到另一类的磁盘中，它也可以扩充而跨磁道。

5。2。控制区域（CA）

在 VSAM 文件中，CI 组成更大的结构-----控制区域 CA（Control Area），文件中的每个 CA 都有同样数量与大小的 CI，若干个 CI 构成 CA。CI 的数量由 VSAM 所决定。

CA 是直接存取空间的单位。要把记录加入文件的末端时，VSAM 就要对 CA 进行格式化。扩充文件时，CA 也随之扩充，这种扩充必须是整数量级的扩充。

换言之，在 DASD 里，CA 都是定长的，当 VSAM 要扩充一个文件的空间时，就将得到一个或者若干个 CA。

通常，CA 总是占据整数量级的磁道，实际上在包含文件的设备中，CA 将占有整个柱面（Cylinder），并从其柱面边界开始占据。

VSAM 在分配空间时都要在每一个 CA 里留一定的空白的、自由 CI，以利于文件的扩充。

5。3。存储记录

在 VSAM 文件中，KSDS 和 ESDS 所使用的记录，可以是定长或变长的，而 RRDS 只能用定长记录。VSAM 在处理这三种类型的文件记录时都是采用同样的方法：

把数据记录存放在 CI 的开始位置上，把描述这些记录的控制信息放在 CI 的末端处。因此，虽然数据记录及其本身的控制信息之间的结合，在物理上通常并不是邻接的，但是作为一个完整的信息，就称之为“存储记录”

通常情况下，存储记录不应跨越 CI，因此在定义 VSAM 文件时，为了使 CI 能够存放最大的存储记录，应该指定足够的缓冲空间，这种非跨越记录的最大逻辑记录的长度为 32768 字节。

5。4。跨越记录（Spanned）

键顺序数据记录和进入顺序数据记录的长度如果超出 CI，但又不能把它们分为几部分，或者为了使这些记录适合于 CI 的大小而又不重新格式化，那幺，这些记录可以跨越或扩充至一个或多个 CI 边界，但在跨越之前，程序员应在定义文件时指定选择项“SPANNED”，这一类记录就称之为“跨越记录”。

跨越记录从其 CI 边界开始，并写满 CA 中的一个或多个 CI，包含跨越记录的最后部分的 CI， 可以存到还没有使用的空间，但是这个空间只能用于扩充跨越记录，而不能包含其它记录。

5。5。相对字节地址

文件中的记录由其位移以字节编址，并且编址是从文件的起始位置开始。这个位移就是记录的相对字节地址 RBA（Relative Byte Address）。例如：文件中的第一个记录其 RBA=0，第二个记录所具有的 RBA 就等于第一个记录的长度，如此类推。

RBA 并不取决于在直接存取卷中记录的位置，换言之，就是不取决于设备的柱面或磁道，同时也不取决于记录所存放的存储器的相连域。对于这种相对字节编址，VSAM 把文件中所有的 CI 视为相连的区间。这样，就犹如把文件存放在一个位置上，从其地址 0 开始。

在计算相对字节地址时，VSAM 把在 CI 中的控制信息和自由空间也作为记录本身一样看待，考虑为地址空间的一部分。

5。6。数据空间

使用 VSAM 时，不必为 VSAM 专门设置一个卷（Volum）,系统文件、其它存取方法的文件都可以与 VSAM 文件一起设置在同一个卷中，但是在卷中却要专门为 VSAM 分配一定的数量的空间，这个专门为 VSAM 分配的空间就称之为数据空间（Data Space）。该空间可以通过 DLBL 和 EXTENT 作业控制语句、或是存取方法服务而定义在卷的 VTOC（Volum Table Of Contend）表中。当然，为了某种需要，也可以把整个卷分配给 VSAM 文件使用而不允许其它文件使用。

数据空间最多可有 16~123 个 EXTENT（范围）但它们之间不必是互相邻接的范围。

数据空间可以包含一个或多个文件，相反地，一个文件也可以放在一个或多个数据空间或者直接存取卷中，但这些卷必须是同一类型的。

5。7。目录（Catalog）

在 VSAM 目录中，有两种目录：主目录（Master Catalog）和用户目录（User Catalog）

VSAM 需要一个主目录，却可以有任意多个用户目录。用户目录由主目录指向并且具有和主目录相同的功能与结构。

引入主目录和用户目录的主要目的是提高数据的完整性、卷的可移植性。

每个 VSAM 目录都存放于单个卷上并占有所驻的卷，当然，目录也可以占有几个卷，但是一个卷只能为一个目录所占有。

所有 VSAM 文件，若存在于卷中，则必须编目到目录中。VSAM 目录包含了所有 VSAM 文件的集中的信息、所在卷的有关信息，如 VSAM 文件分配的数据空间等。

由上图可见：VSAM 主目录是用来指引各个用户目录，当 VSAM 要用一个用户目录时首先查找 VSAM 主目录。

而 VSAM 提供多个用户目录，每个用户目录都是独立的，控制了它的数据空间与文件，即这个用户所定义的每个 VSAM 文件与数据空间都要在这个用户目录上有一个进入点。

一个 VSAM 文件在其用户目录进入点有如下信息：文件的位置及其属性（如记录长度、键位置）、文件的一些动态信息（如文件建立后插入的记录数、控制区间分裂的个数等）。

5。8。族（Cluster）与路径（Path）

在 VSAM 方法中，族是由一组有关的部分而组成的结构，如：数据部分及其索引部分所组成的整体结构。VSAM 把所有的文件都当作族来处理。因此，族是 VSAM 实施管理的基本单位。

路径是一个逻辑组织，提供了访问族或者基本族（Basic Cluster）的记录，而这种访问可以直接访问，也可以通过替换索引进行。

5。9。主索引（Prime）与替换索引（Alternate Index）

如何把所有的数据记录存放在键顺序文件中，而在扩充文件时，应该怎样查找及确定记录在文件中的位置？实际上，VSAM 通过建立索引完成上述的运算工作。这个索引称之为主索引（Prime Index）。主索引与每个数据记录的键字段相匹配，以此确定记录在文件中的相对位置。

文件中的每个记录的键字段大小与位置都必须是一样的，这是因为 VSAM 通过键字段值而保持其在磁道中的位置。实际上，用户引用记录只是根据其键字段值（Key Value），而不是根据它在物理设备中的地址。

对于给定的键顺序文件文件或进入顺序文件（没有主索引），VSAM 同样可以建立替换索引（Alternate Index），替换索引提供了能够访问这些数据记录的另一种方法，通过这些记录中不同的键 字段（替换索引），均可访问已给定的数据记录。

由于替换索引同样提供了一种获得同样数据记录的方法，所以，当有几个替换索引

时，就可以有几种不同的方法去访问文件。这样，为使用者带来了方便，因为不必保留信息相同的、但结构不同的、多种的文件复制副本。

5。10。CI 与 CA 的分裂（Split）

以前所讨论的记录的插入、增加和扩充，都是假定在控制区间 （C I）中存在着足够的自由空间足以容纳这些记录。如果要插入的记录并不能全都放在一个 C I 内，就会出现控制区间分裂（CI Split）。这时，VSAM 会将这些数据记录连同它们的控制信息从已写满的控制区间移至同一控制区域 CA 中空的控制区间 CI 中，并以适当的键序插入新的记录。

在指定的 CA 里，当 CI 中的自由空间不能容纳新记录时，就要出现控制区域的分裂（CA Split），VSAM 会在文件的末尾处建立新的 CA。它可通过使用原来已经分配的空间来实现，也可以通过扩充文件而实现。

一般而言，直接插入所引起的分裂，出现在 CI 和 CA 的中点位置，顺序插入引起的分裂，则是出现在 CI 和 CA 的插入位置上。对于有足够自由空间分布的文件，不应经常出现分裂。

5。11。VSAM 数据结构

通常，ISAM 的数据结构是根据磁盘柱面及磁道的物理单元而定的。而 VSAM 的数据结构却是根据 CI 及 CA 的逻辑单元而定。CI 是直接存取存储器的单位，它将数据信息传送给虚拟存储器，或从虚拟存储器中把数据信息传送过来。

VSAM 的数据结构提供了设备的独立性，减少了程序员对数据和索引的物理特性的关注，从而为应用编程带来了方便。

六、AMS 实用程序

1、用于建立和维护 VSAM 数据集和世代数据集。

2、当使用 VSAM 数据集或维护系统目录时，必须使用 AMS 命令。

3、AMS 命令分两类：

功能命令，如定义数据集，列表目录等；

辅助命令，作用是设置条件执行功能。

4、AMS 实用程序的调用模式

使用 AMS 主要有 TSO 环境下通过 AMS 命令及利用 JCL 调用方法。JCL 调用模式：

```
//JOB1  JOB
//JOBCAT  DD
//     DSNAME=DB.DATA,DISP=SHR
//STEP EXEC  PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN DD *
   COMMAND  parameters...
/*
```

注：

JOBCAT DD 语句定义一个目录名；

EXEC 语句指出 AMS 程序名为 IDCAMS；

SYSPRINT DD 语句指定系统的输出信息；

SYSIN DD 语句为 IDCAMS 提供各种 AMS 命令；

5、AMS 命令及功能
ALTER:修改数据集和目录属性；
BLDINDEX:建立辅助索引；
DEFINE ALIAS：为目录或数据集建立别名；
DEFINE ALTERNATEINDEX：定义辅助索引；
DEFINE CLUSTER：为 VSAM 数据集定义簇；
DEFINE GENERATION DATA GROUP:为世代数据集定义编目入口；
DEFINE NONVSAM:为非 VSAM 数据集定义编目入口；
DEFINE PAGE SPACE:定义系统页空间数据集；
DEFINE PATH:定义连接辅助索引与主数据集的路径；
DEFINE USER CATALOG:定义用户目录；
DELETE:删除目录、VSAM 及非 VSAM 数据集;
EXPORT:中断用户目录与主目录的联系；
IMPORT:接通用户目录与主目录的联系；
LISTCAT:列表编目内容；
PRINT:打印 VSAM、非 VSAM 数据集及目录内容；
REPR 拷贝 VSAM、非 VSAM 数据集及目录，分类及综合编目功能
Example : Define a key-sequenced data set

```
DEFINE CLUSTER ( NAME(CUSTOMER.MASTER.FILE)        -
                 OWNER(DLOWE2)                      -
                 INDEXED                           -
                 RECORDSIZE(200 200)                -
                 KEYS(9 12)                        -
                 VOLUMES(MPS800)                    -
                 UNIQUE                            -
                 FREESPACE(20 10)                   -
                 SHAREOPTION(3)                     -
                 SPANNED                           -
                 IMBED)                            -
       DATA     ( NAME(CUSTOMER.MASTER.FILE.DATA)    -
                 CYLINDERS(50 5)                    -
                 CISZ(4096)                        -
       INDEX    ( NAME(CUSTOMER.MASTER.FILE.INDEX) )
```

1--CYLINDERS(primary[secondary])
RECORDS(primary[secondary])
TRACKS(primary[secondary])

用以定义 VSAM 文件的空间，其中 CYLINDER，TRACK，RECORD 为单位，现在我们的磁盘中
1CYLINDER = 849960 BYTE,1TRACK=56664BYTE,1CYLINDER=15TRACK, 而 RECORD 的大小则由 RECORDSIZE
参数决定。另外，primary 为初次分配空间，secondary 为每次 extend 时分配空间，VSAM 会根据两者
中的最小值计算 CA 的大小，但 CA 最大不大于 1CYLINDER。对于本系统，每个 VSAM DATASET 允许 extend 123 次（当使用了 REUSE 参数时只允许 extend 16 次）。另用

RECORDS 定义有利于空间计算，但用此方式　　定义文件会影响 CA 的充分使用，希望不采取此方式定义文件。

2---RECORDSIZE(average maximum)

用以定义每个记录的大小，average 为平均记录长度，maximum 为最大记录长度，应注意的是在不指定 SPANNED 参数时，每个记录的最小值是 1BYTE,最大值是 32761BYTE。(SPANNED 指定允许记录跨 CI）

3---INDEXED|NONINDEXED|NUMBERED

用以定义 VSAM 文件的类型，INDEXED 指定 KSDS，NONINDEXED 指定 ESDS，NUMBERED 指定 RRDS。

4---SHAREOPTIONS(crossregion[crosssystem]|1 3)

用以定义 VSAM 文件的共享类型，对于现役系统，暂时只有第一个数有意义（同一主机下的共享），其中 1 代表允许并发的读请求或单个的更新请求，2 代表允许并发的读请求和单个的更新请求同时发生。

对于 1，系统能确保读写的数据完整性，对于 2，系统确保写的数据完整性，但不确保读的数据完整性。　　另外，由于 3 和 4 VSAM 并不确保写的完整性，暂不在考虑之列，基于本系统的特点，建议此参数选　　SHAREOPTIONS（1，3）。

5---CONTROLINTERVALSIZE(size)

用以定义CI 的大小,若在DEFINE CLUSTER 时指定,则此定义自动影响到DATA 和 INDEX 的定义，或在　　DATA 和 INDEX 中分别指定。应该注意的是在不指定 SPANNED 参数时，此参数应大于等与最大记录长度+7

（见 RECORDSIZE 定义）。一般，对于经常作连续记录处理的文件应选较大的 CI，对于经常作离散记录　　处理的文件应选较小的 CI。由于指定此参数需要有一定的经验和技巧，建议不指定此参数，让系统根据文件的平均记录长度和最大记录长度自动计算适宜的数值。

6--- BUFFERSPACE(size)

定义 BUFFER 的最小值。VSAM 会利用此参数计算 CI 的大小，若不指定此参数，VSAM 默认 BUFFER 为两个 DATA CI 和一个 INDEX CI（KSDS）。

7---REUSE

当指定了 REUSE 参数时，VSAM 文件能在不 delete define 的情况下重新使用，但应注意的是，定义了 REUSE 参数的 VSAM 文件只能 extend 16 次，并不能定义 KEYRANGE，UNIQUE 参数。

8---FREESPACE(CI-percent[CA-percent]|0 0)

CI-percent 定义每个 CI 中预留空间的百分比，CA-percent 定义每个 CA 中预留空间的百分比。此参数一般用于经常有记录插入的 VSAM 文件，以避免经常发生的 CI SPLIT 和 CA SPLIT。建议对于不会发 生记录插入的文件，不定义此参数，对于会发生记录插入的文件，按记录插入的频繁度定义此参数。

9---SPEED

当指定 SPEED 参数时，空 VSAM 文件在首次记录载入时并不预格式化其数据部分，从而加速了首次记录载入的过程。但需注意的是，此参数只在首次记录载入时起作用，以后取而代之的是 RECOVERY 参 数（虽然在 DEFINE 时选定了 SPEED），因此，建议不使用 SPEED（系统默认参数是 RECOVERY）。

10---REPLICATE

当指定了此参数时，每个 INDEX SET 的记录都会重写多次，直到写满一个 TRACE，因此，INDEX 的搜索速度会加快，但也会带来磁盘空间的增加。（见附图）

11---IMBED

当指定此参数时，每个 CA 的 SEQUENCE SET 都会从 INDEX 中分离出来，写入每个 CA 的第一个 TRACK。此参数的可让我们将 INDEX SET 放在高速设备上，而 SEQENCE SET 放在低速设备上。而且，在我们的硬盘上有 CACHE，SEQUENCE SET 一般都已读入 CACHE，设此参数并不能提高性能。所以建议不使用 IMBED。