

Chapter 2

KSDS STRUCTURE

*Get on the
Fast Track!*



TM

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- KSDS file structure.
- VSAM buffering.
- Structure of the index.
- Index record structure.
- Index CI: Control Interval sizes and capacities.
- Index CI capacity.
- How long keys can distort capacity estimates.
- How data is added to and removed from the data component and control interval.
- The role of CI and CA splits.
- Concept of FREESPACE.

1 KSDS File Structure

The KSDS is the most widely used VSAM file structure, replacing ISAM for key-sequence file handling and physical database files. It is composed of an index component and a data component, addressed through VSAM I/O software as a single cluster. Additionally, each component can be addressed separately as an ESDS.

The definition and handling of KSDS clusters is the central focus of VSAM performance and tuning techniques.

In VSAM KSDS, if a search argument of a logical record is its key field, then in all logical records, the key length must be the same. The length of the key must be from one byte to 255 bytes.

In all logical records the key offset in relation to the beginning of the logical record beginning must be the same. The value of the key field cannot be duplicated among the logical records, it must be unique. After it is specified, the value of the key cannot be altered; however the entire record can be deleted.

2 VSAM Buffering

Buffering is central to the I/O performance. A VSAM buffer is a virtual storage area where the CI is transferred during an I/O operation.

In VSAM there are two types of buffers:

- | | |
|-------------------------|--------------------------|
| • buffers for data CIs. | • buffers for index CIs. |
|-------------------------|--------------------------|

A buffer pool is a set of buffers with the same size.

A resource pool is a buffer pool with control blocks which define the pool and the clusters with CIs in the resource pool.

3 Structure of the Index

The index component of a KSDS contains information relating logical keys to data record locations in the data component of the cluster. The logical data key must be in the same place for the same length on all data records

All extents of the index component must be on the same device TYPE.

3.1 Levels and Organization

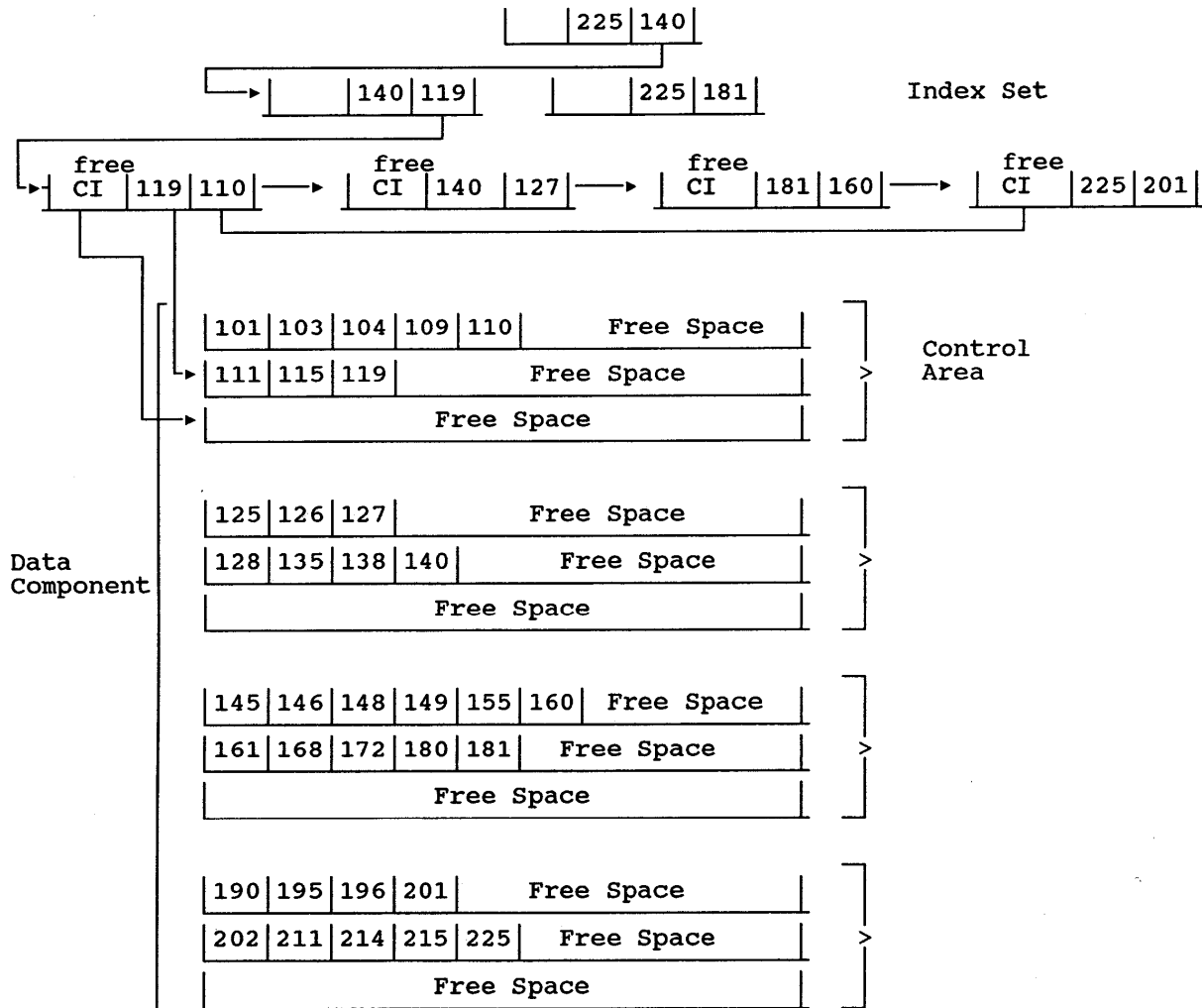
The index consists of one or more levels in a logical tree structure. The top level always has one CI; the lowest level has one CI per data component CA. The lowest level of the index is called the sequence set; higher levels are collectively called the index set. The number of levels is determined by the size of the data component and the size of the index CI.

Index CIs are not grouped into control areas; new index CIs are added to the end of the dataset, regardless of level.

Each entry in an index record contains a pointer to a control interval on the next lower level, along with its high key.

An index entry in the sequence set points to a data component CI, along with its high key.

Key-Sequenced Datasets Index Component



3.2 Index Record Structure

Each index record occupies a control interval by itself.

Index record	RDF : CIDF
--------------	------------

Both the RDF and CIDF contain the record length, CI size - 7.

record length	0
---------------	---

The control interval after the last-used control interval in the index has a CIDF of all zeros.

Each index record consists of:

- A header.
- Pointers to free control intervals in the data component for sequence set records only.
- Unused space.
- Index entries.

	<---free ci pointers --->				<- index entries ->		
header	pointer n	pointer 2	pointer 1	unused space	entry n	entry 2	entry 1

3.3 Index CI Sizes

The same CI sizes are permitted for data component. Size can be specified in the VSAM Define or chosen by VSAM.

The Index must be large enough to hold an entry to each CI in the data CA. Smaller index CI size shortens search time through the index if no extra I/O is created.

3.4 Index CI Capacity

The capacity of an index CI is related to:

- Index CI size.
- Length of the data key.
- Number of CIs in the data CA.
- Effects of key compression routines.

For keys less than 150 bytes long, with average or better compression, these guidelines will be reasonably accurate:

<u># of Data CIs in CA</u>	<u>Index Size</u>
1 < # < 58	512
58 < # < 120	1024
120 < # < 184	1536
184 < # < 248	2048
248 < # < 312	2560
312 < # < 376	3072
376 < # < 440	3584
440 < # < 502	4096
.	.
.	.
.	.

3.5 Long Keys Can Distort Capacity Estimates

For a more precise assessment, use the following formula:

$$\# \text{ Of Keys in CI} = (\text{Index CISZ} - 31) / ((\text{Key Length} / 3) + 3)$$

The estimated key compression ratio is approximately 1/3.

Three bytes of control information are generated for the SSI: Sequence Set Records.

Then compare the answer to the number of CIs in the data control area.

If there are more data CIs than will fit:

- Increase the index CI size and redo the formula.
- or
- Increase the data CI size to decrease the number of CIs in the CA.

VSAM software will force a larger CI size if it detects a specified size which is significantly too small. However, it can not anticipate bad key compression.

If the data portion of a file is significantly larger than expected, check for bad key compression, which uses up too much room in the index and makes data CIs unaddressible. Expand the index sizes accordingly.

4 How Data is Added to and Removed from the Data Component

4.1 How Data is Added to and Removed from the CI

In order to keep a KSDS in sequence, new records are placed in the CI entitled to hold its logical key.

This is determined by finding the CI with a HIGH KEY in the index which is sequentially the first key higher than the new key.

When there is room in the CI for the new record or extension of an old record, the CI is reorganized in order that the new data is placed in the correct physical order.

- Records to the right of the insert location are moved farther right.
- RDF and CIDF values are updated to show the new record values and remaining free space.
- Deletions also maintain the order within the CI.
 - Records to the right of a deleted record are moved left to close the gap.
 - RDF and CIDF values are updated to show the new record values and new free space.
- All insert and delete activity can cause RBAs of data records to change.

4.2 CI and CA Splits

When there is insufficient room in a CI to absorb additional data, a CI split will occur.

- An empty CI is located in the CA.
- A portion of the old CI is moved to the new one.
- The new data is added in its proper place.
- Indexes are adjusted to reflect the new data.
- When no free CIs are in the CA, a CA split occurs.
 - An empty CA is located.
 - A portion of the old CA is moved to the new one.
- The CI split continues as usual.

CI and CA splits occur in two modes:

NIS: Normal Insert Strategy

- Used during direct mode processing.
- Splits occur at the half-way point of the CI or CA to be split, regardless as to the insert location.

SIS: Sequential Insert Strategy

- Used during sequential mode processing or when effective sequential processing is detected by VSAM software.
- Splits occur at the point of logical insertion of the new data.

In the CI:

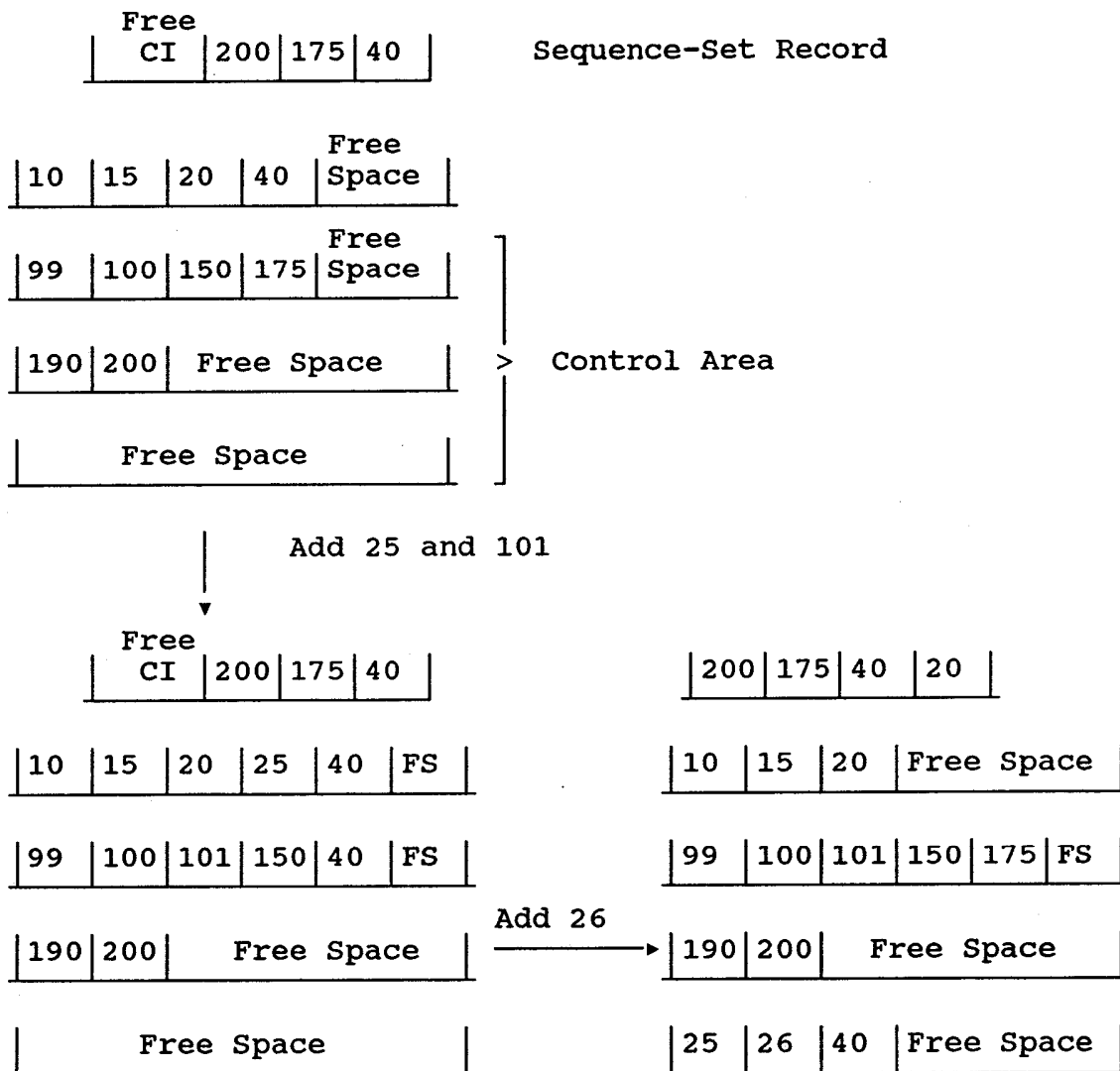
If the key of the new record is higher than every key in the CI, it becomes the first record in the new CI. Otherwise, the new record is placed in the old CI, and all higher records go in the new CI.

In the CA:

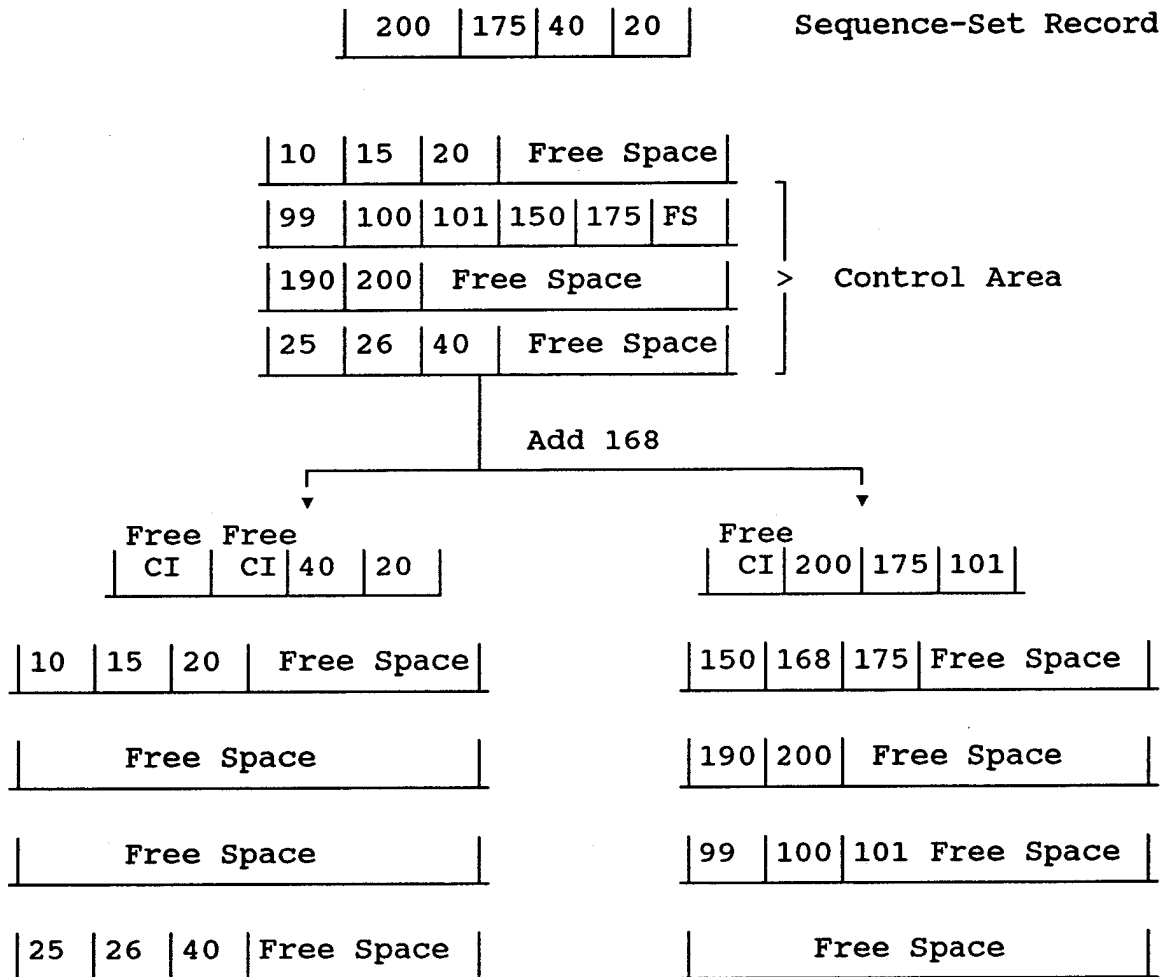
If the CI to be split is the highest key in the CA, it becomes the first CI in the new CA. Otherwise, it stays in the old CA, and all higher CIs are moved to the new CA. CI split logic takes over to choose the split location within the CI.

4.3 CI and CA Splits During Direct Processing

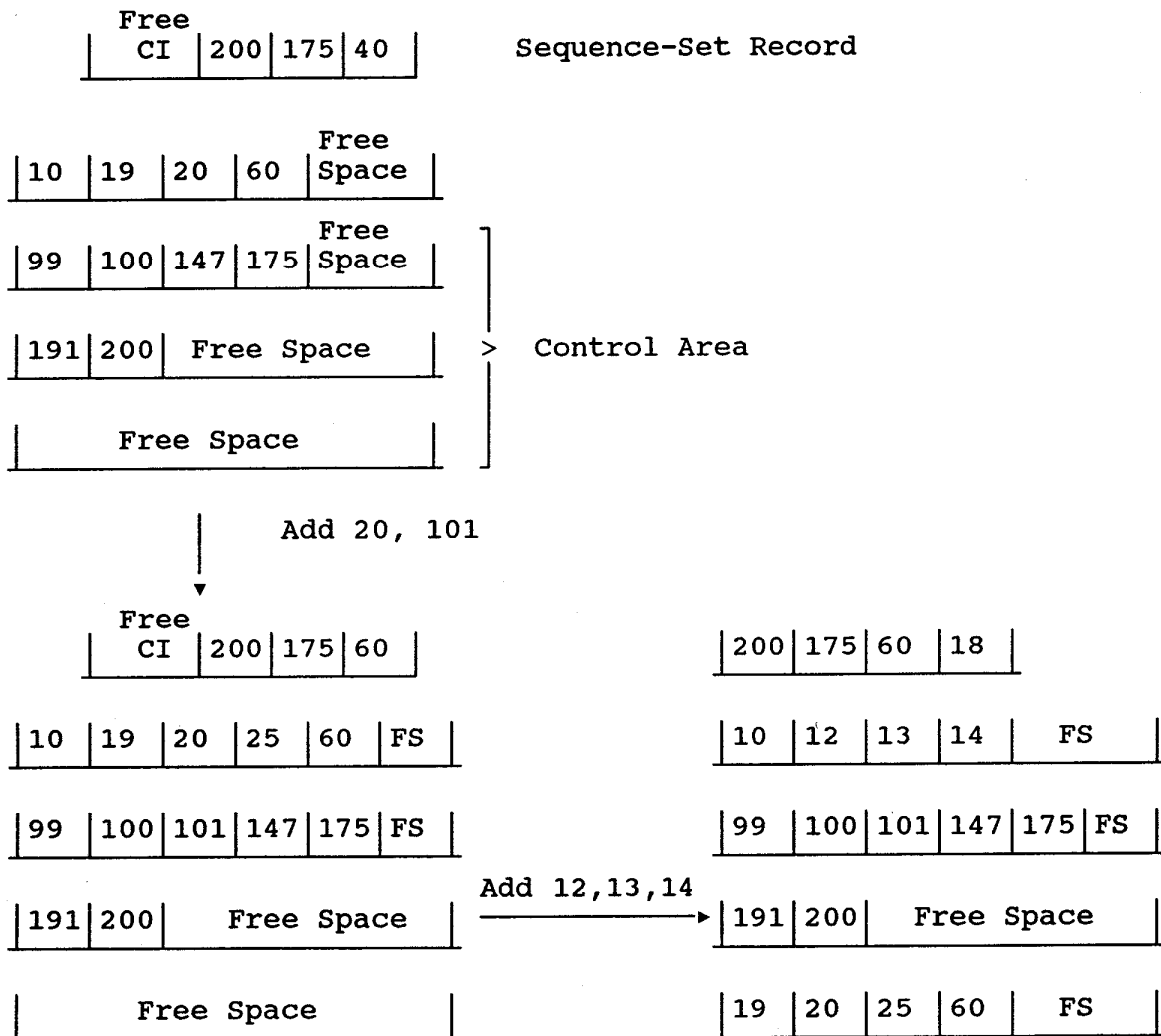
Key-Sequenced Datasets



Key-Sequenced Datasets



Key-Sequenced Datasets



Key-Sequenced Datasets

200	175	60	18
-----	-----	----	----

Sequence-Set Record

10	12	13	14	FS	
99	100	101	147	175	FS
191	200	Free Space			
19	20	25	60	FS	

} Control Area

Add 144

175	146	60	18
-----	-----	----	----

Free	Free	Free	
CI	CI	CI	200

10	12	13	14	FS
----	----	----	----	----

191	200	Free Space		
-----	-----	------------	--	--

99	100	101	144	
----	-----	-----	-----	--

Free Space				
------------	--	--	--	--

147	175	Free Space		
-----	-----	------------	--	--

Free Space				
------------	--	--	--	--

19	20	25	60	
----	----	----	----	--

Free Space				
------------	--	--	--	--

5 Costs of CI and CA Splits

5.1 I/O at Time of Split

Both CI and CA splits incur index I/O as well as extra data I/O, since key ranges for CIs and CAs change as a result of the split activity.

In addition to normal retrieval and update for the CI split:

- A new CI is built and written out.
- The sequence set record is updated to add the new CI high-key, remove it from the free-ci list, and keep all CI high-keys in order in the index record.

For the CA split:

- A new CA is located by interrogating the catalog.
- Half the CIs in the original CA are read and written to the new CA.
- The sequence sets of both the old and the new CAs are rewritten to show the new content of the CAs and reset the horizontal pointers in the CAs.
- The index record above the sequence set level is updated to show the new high-keys in the sequence set cis.
 - If this results in a CI split at the index level above the sequence set, the I/O activity will be propagated upward until all changes are affected.
 - If no space is available and the cluster is eligible for extension, a DADSM: Direct Access Device Space Management space assignment occurs.
 - Both the catalog entry and the VTOC entry may be updated.

5.2 Costs After Split

For the CI split:

- There is no appreciable affect on direct processing.
- Assuming the free space in the resulting CIs is not filled with more records, there will be a slight degradation in sequential processing

For the CA split:

- There is no appreciable effect on direct processing.
- There is no loss of efficiency in sequential processing, since new CAs are placed out of order at the end of extents or in new extents, increasing seek time.
- There will be fragmentation of disk space.

5.3 CA Splits

CI Size	CI in CA
512	785
1,024	462
2,048	315
4,096	180
6,144	120
8,192	90
12,288	60
16,348	45
32,768	22

On a 3390 device, there will be a CA size of 1 cylinder.

5.4 Concept of FREESPACE

FREESPACE is a user-specified amount of space, in percent, to be set aside at load time to provide for future direct access insert activity.

- It is specified at both the CI and CA level at define time.
- It is reestablished in a KSDS when file is reloaded with REPRO or IMPORT.
- It is used during NIS insert mode processing.
- It is set aside during SIS insert mode processing.