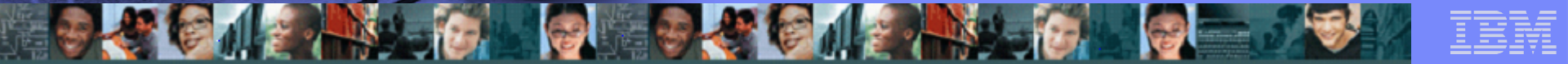


# IBM Academic Initiative



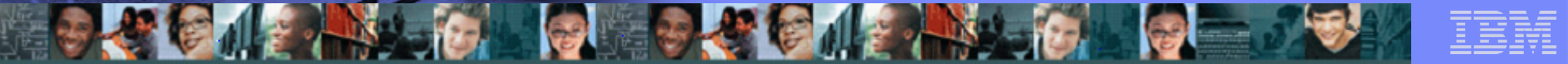
## Modulo 3 Dataset





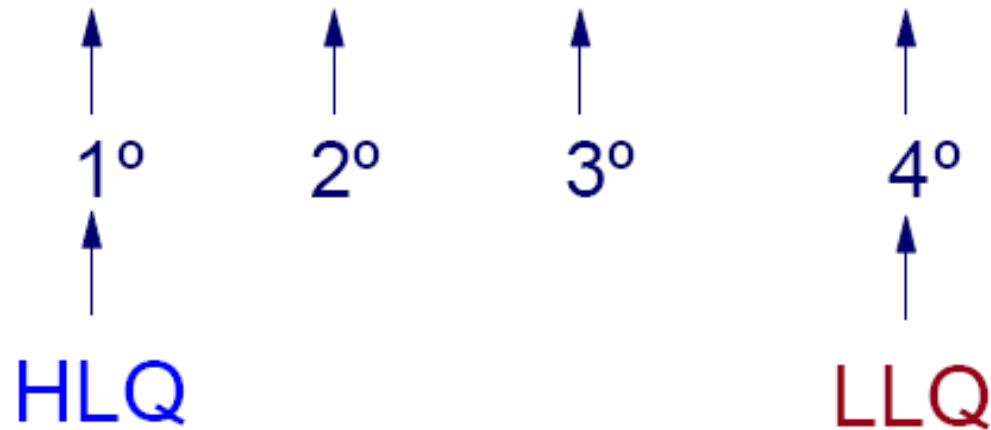
# Definizione di Data set

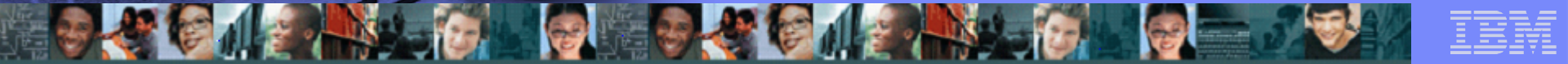
- Un data set e' una collezione di records logicamente correlati memorizzati su uno o più dischi ( o nastri)
- Un data set può essere:
  - programma
  - libreria di macro
  - file (records) usati da un programma.
- Il record (logico) e' l'unità basilare di informazione usata dal sistema operativo z/OS.



## Nomenclatura Dataset

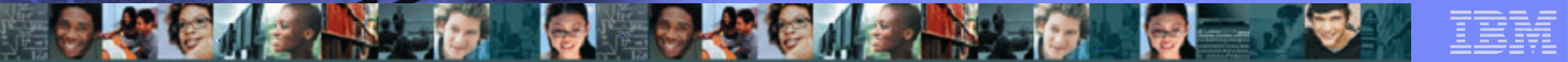
HARRY.FILE.EXAMPLE.DATA





# Come sono memorizzati I data set in z/OS

- I dati sono memorizzati su DASD, nastri magnetici o dischi ottici.
- Si possono memorizzare o reperire records o con accesso diretto o con accesso sequenziale.
- Si possono usare DASD per memorizzare programmi eseguibili , sistema operativo incluso, e per aree di lavoro temporaneo.
- Si può usare un DASD per molti data sets, e riallocare o riusare il volume
- Nastri e dischi ottici sono anche usati in lettura e scrittura.

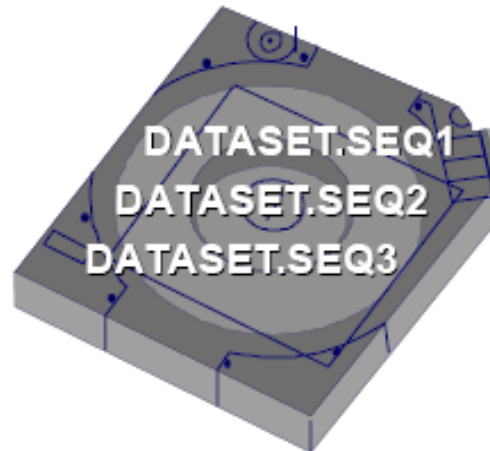


## DASD volume



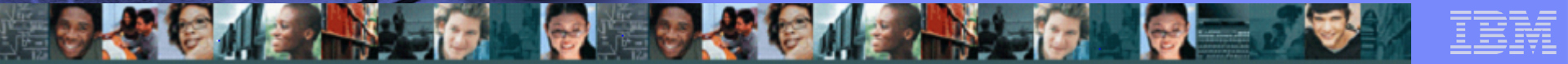
**volser=DASD01**

## tape volume



**volser=SL0001**

*Direct Access Storage Device (DASD)*



# Caratteristiche fisiche dei DASD

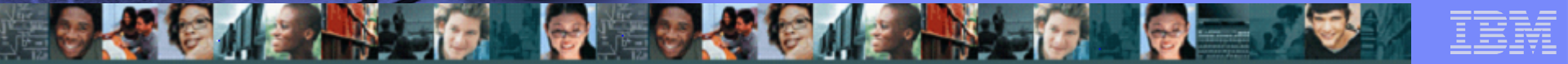
I DASD sono suddivisi in:

- Cilindri
- Cilindri contengono *tracce*
- Tracce contengono *records*.

	Modello 3390
Cilindri	3.339
Tracce per cilindro	15
Bytes per traccia	56.664
Capacità totale	2,8 GB



Dasd asportabile degli anni '70



# Tipi di data set

- Sequential,
- Partitioned,
- VSAM

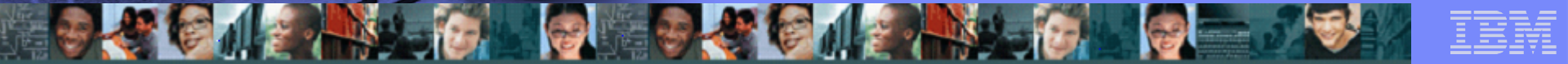
Un data set sequenziale è una collezione di records scritti in ordine sequenziale dall'inizio alla fine.

Un partitioned data set (PDS) è una collezione di data set sequenziali , chiamati membri

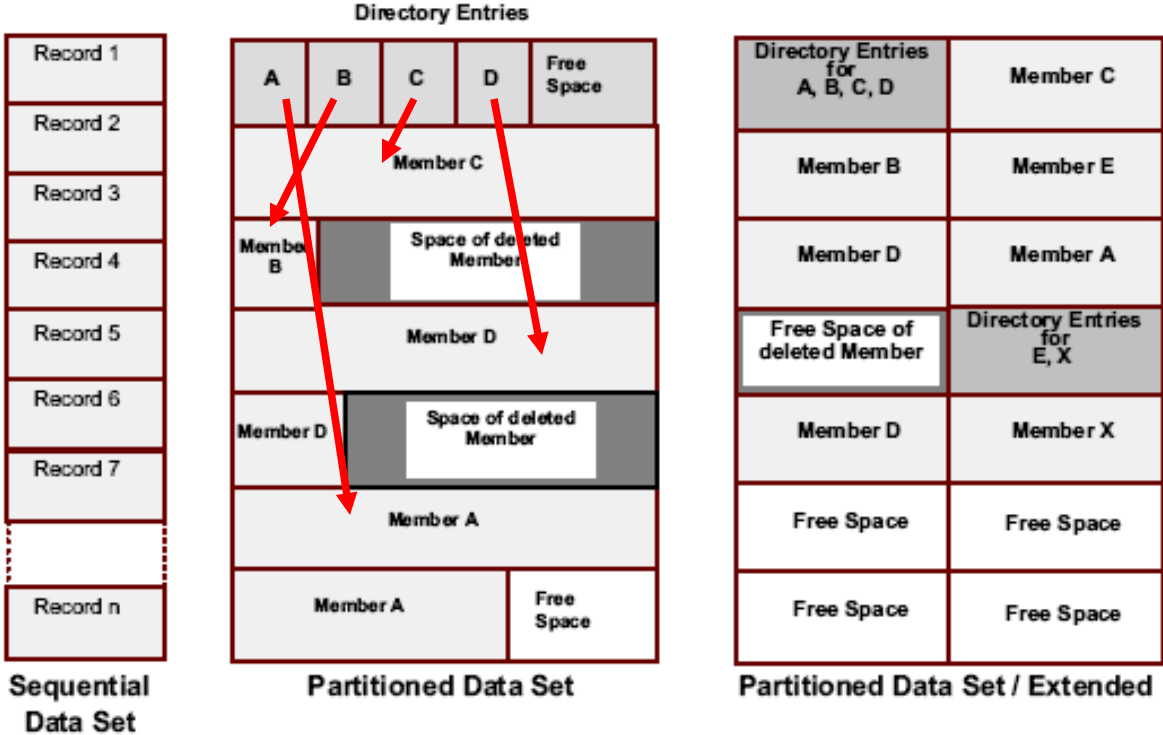
- Consiste di una Directory e di uno o più membri . Esso è anche chiamato libreria (*library*) .

Esiste, inoltre, un tipo di data set partitioned esteso ( PDSE ) con migliori performance.

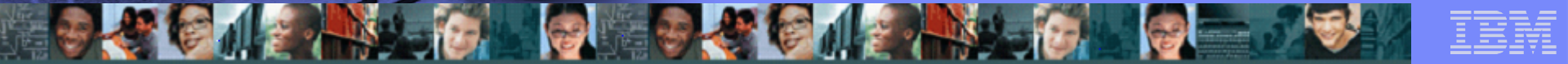




# Tipi di Data set







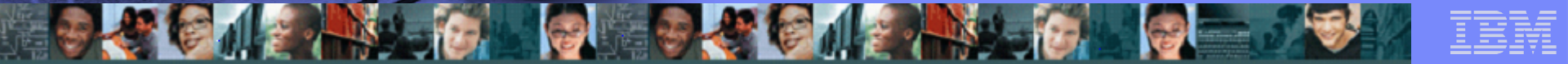
# Record Format

In z/OS data sets sono collezioni di record.

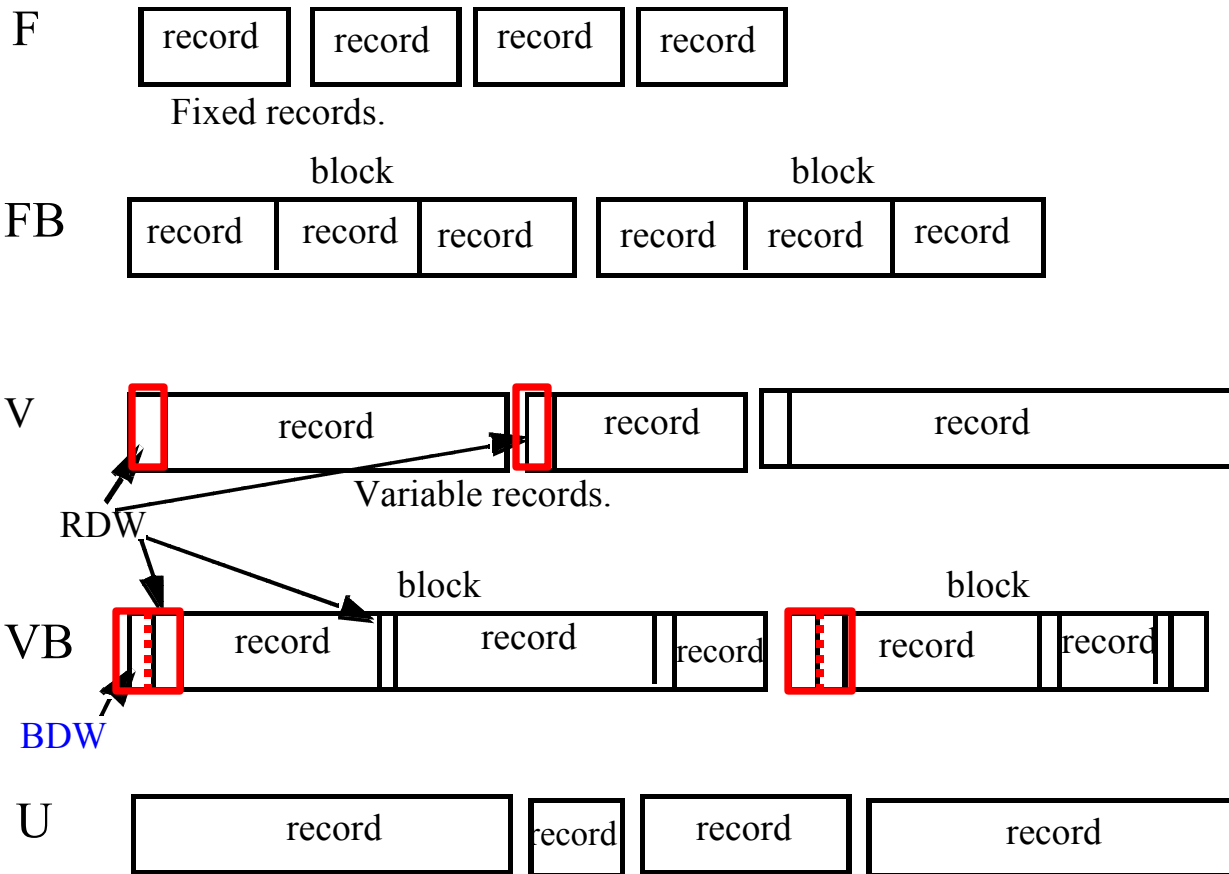
Non esiste il concetto di “byte stream” files come sulle piattaforme Intel (PC) e UNIX, anche se attualmente la funzione “z/OS UNIX System Service” usa dei byte stream files, ma non sono considerati data sets “tradizionali”

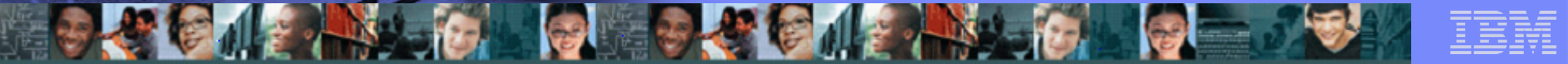
Non esistono i caratteri new line (NL) or carriage return and line feed (CR+LF) per indicare la “end of a record”. Records possono essere di lunghezza fissa “fixed length” o variabile “variable length” in un data set. I data sets hanno i seguenti formati:

- **F - Fixed** Tutti i blocks/records hanno lo stesso size, su disco 1 block è 1 record logico
- **FB - Fixed Blocked** I physical block contengono più logical records. Questo garantisce maggiore efficienza nell'utilizzo dello spazio. Molto utilizzato.
- **V - Variable** Come per i fixed, un logical record è un physical block. L'applicazione deve inserire un Record Descriptor Word (RDW) di 4 bytes all'inizio del record. RDW contiene la lunghezza del record.
- **VB - Variable Blocked** I physical block contengono più logical records, ognuno con il proprio RDW. Inoltre, l'applicazione deve aggiungere un'altra word, la Block Descriptor Word (BDW) che contiene la lunghezza totale del “blocco”.
- **U - Undefined** Questo formato è composto di records/blocks di lunghezza variabile, senza alcuna struttura predefinita. Normalmente usato per i data-set che contengono moduli eseguibili.



# Data set record formats

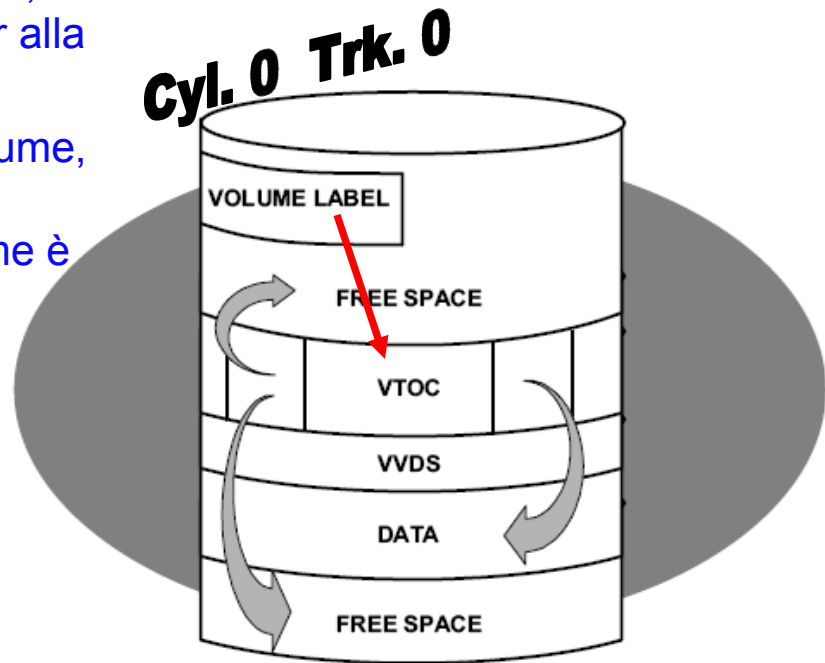


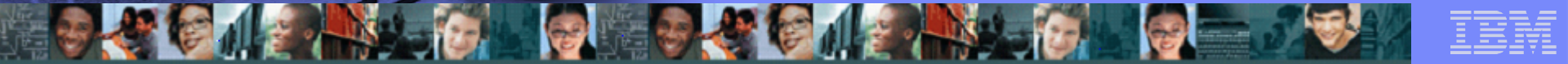


# Organizzazione dei DASD

Ogni DASD contiene una **Volume Label** ed una **Volume Table of Contents (VTOC)**.

- La volume label, posizionata sempre a sul cilindro 0, traccia 0, contiene il Volume Serial Number, **VOLSER**, che identifica in maniera univoca il disco e un pointer alla VTOC.
- La VTOC descrive tutto lo spazio all'interno del volume, incluso lo spazio libero
- La VTOC serve a determinare "dove" su quel volume è localizzato il dataset



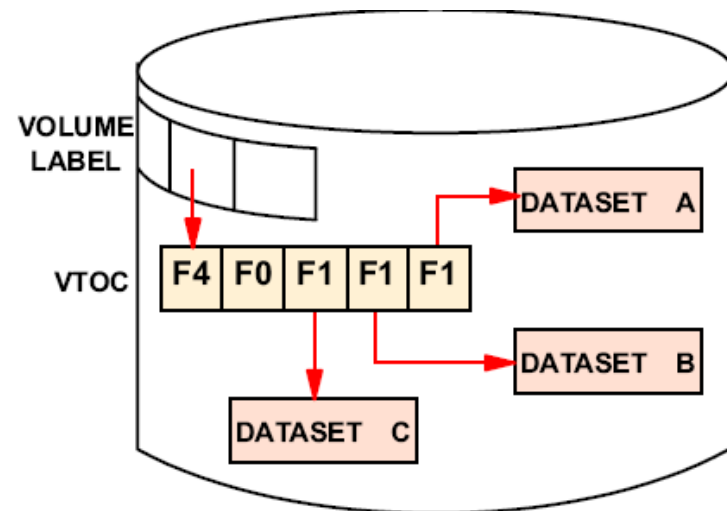


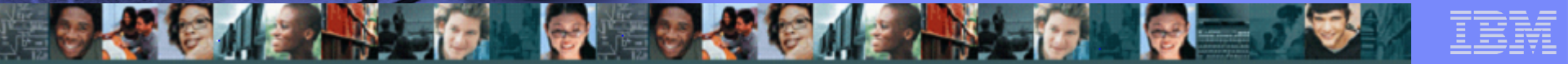
# VTOC

La VTOC è un data-set contenente dei blocchi di controllo detti **Data Set Control Blocks DSCB**, che descrivono il contenuto del volume.

I DSCB sono di formato diverso, a seconda dell'oggetto che rappresentano. I più importanti sono:

- **Format-0** descrive un record non usato nella VTOC
- **Format-1** descrive un dataset
- **Format-4** descrive la VTOC
- **Format-5** descrive lo spazio libero, free space





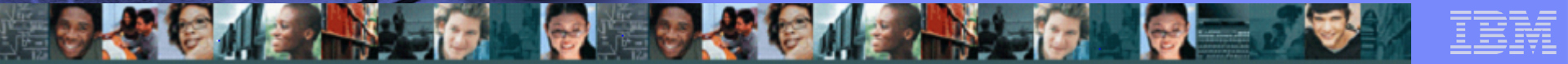
# Uso dei data set

- Per usare un data set occorre prima allocarlo ovvero riservare un'area su disco al programma che lo deve utilizzare.
- Successivamente, il programma potrà accedere al data-set utilizzando modalità d'accesso standard fornite dal sistema.

- Esempio di allocazione:

```
■ //PDCRTSJ2      JOB SIMOTIME,ACCOUNT,CLASS=1,MSGCLASS=(1,1)
■ //IEFBR14       EXEC PGM=IEFBR14
■ //TEMPLIB1     DD DISP=(NEW,CATLG),DSN=TEST.DEMO.LIB1,
■ //              SPACE=(TRK,(45,15,50)),UNIT=SYSDA,
■ //              DCB=(RECFM=FB,LRECL=80,BLKSIZE=800,DSORG=PO)
■ //*
```

E' possibile scegliere di allocare il dataset su un DASD prescelto, specificando il nome del VOLSER col parametro VOL=SER=DASD01.

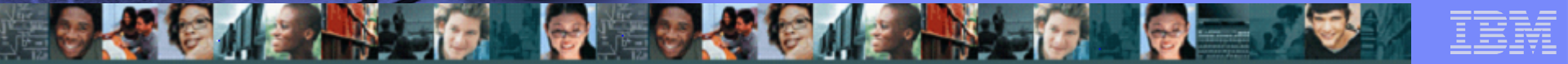


# Unit Names

Il nome Unit identifica l'unità periferica, “**device**” , alla quale si vuole accedere. Si può identificare il device in diverse modalità:

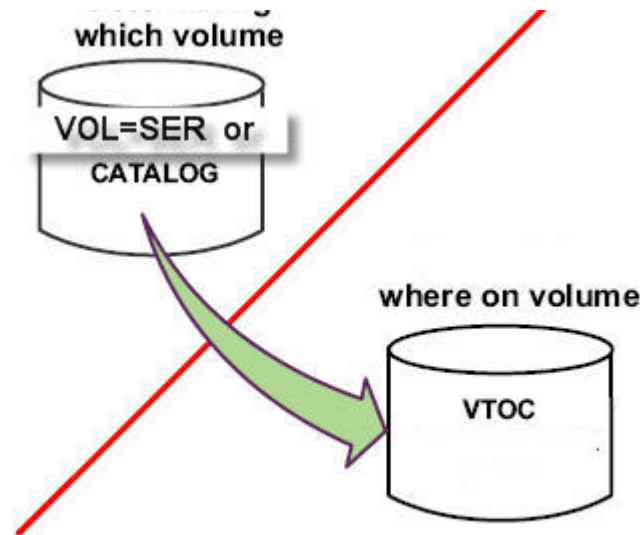
- **Specific**     **UNIT=182** l'utente limita la scelta ad un unico device.
- **Generic**     **UNIT=3390** l'utente identifica un gruppo di devices con caratteristiche omogenee.
- **Esoteric**     **UNIT=SYSDA** l'utente identifica un gruppo di devices pre-determinato a tempo d'installazione. Un gruppo esoterico comprende devices di uno o più “generic” groups.

Generic Device Types	3380				3390			
Esoteric Group Name	SYSDA							
Specific Device Number	180	181	182	183	190	191	192	193



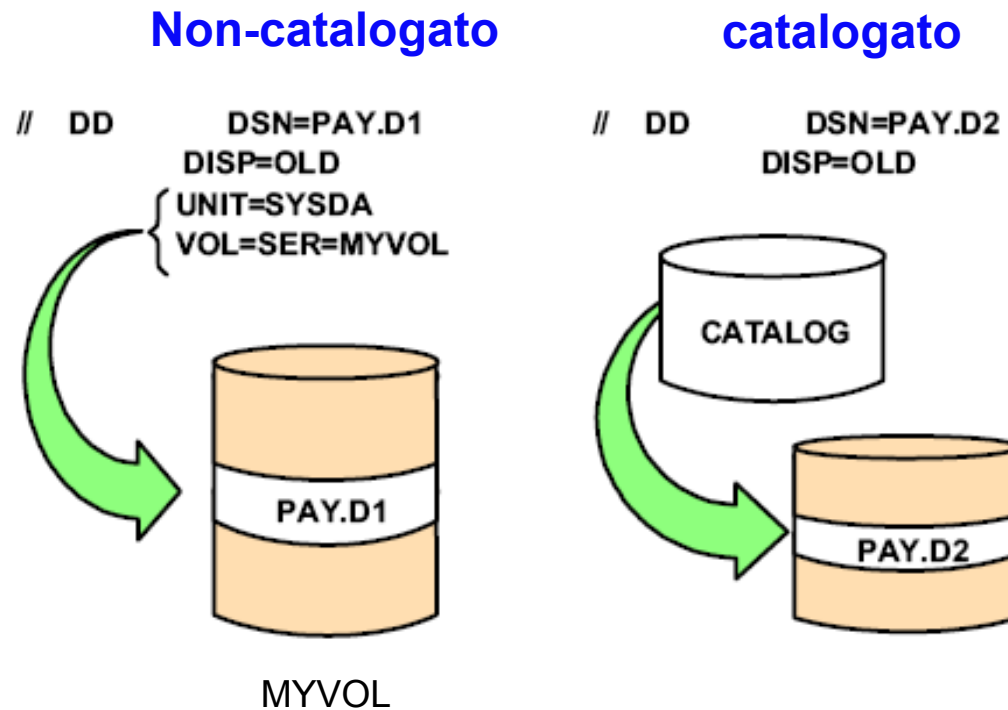
# Localizzazione di un dataset

- Per conoscere la locazione fisica di un dataset, lo z/OS usa metodi e percorsi diversi a seconda delle sue caratteristiche:
- il nome del volume (vol=ser=...) specificato con le JCL, se il dataset non è “catalogato”;
- il catalogo che, sulla base del nome del dataset, indirizza il volume che contiene il dato.
- In entrambi i casi, una volta individuato il volume, verrà usata la VTOC per sapere “dove”, all’interno del volume, è memorizzato il dataset.





## Localizzazione di un dataset (2)

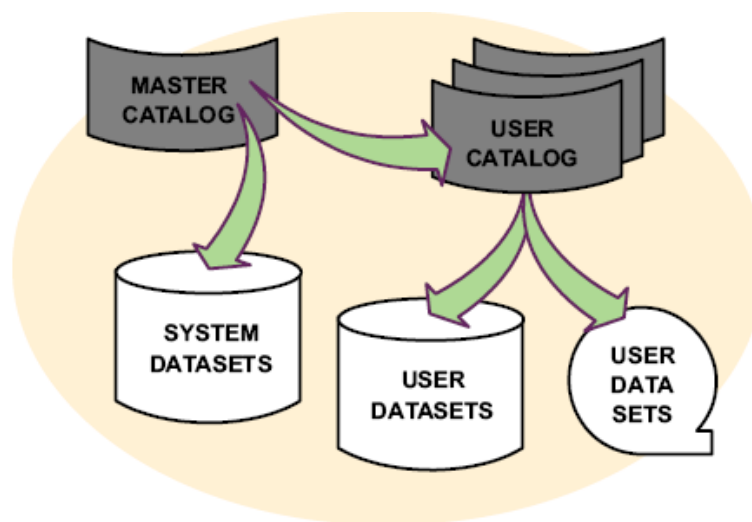


# Catalogo

Il catalogo è lo strumento che permette, sulla base del nome del dataset, una ricerca rapida dello stesso e svincola l'utente dal conoscere l'esatta locazione del dataset da ricercare.

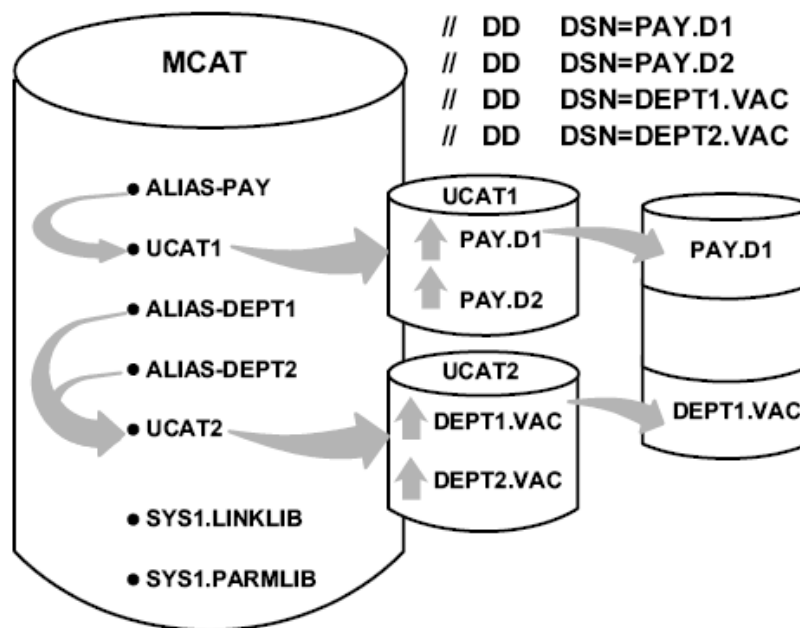
Esistono due tipi di catalogo, **Master** e **User catalog**.

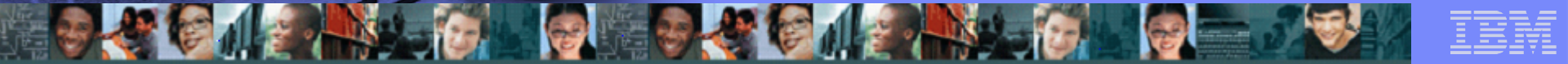
- Il Master contiene il puntamento ai dataset di sistema ("**SYS1**.dsname") e ai cataloghi "user" utente.
- User catalog contiene il puntamento ai dataset utente.



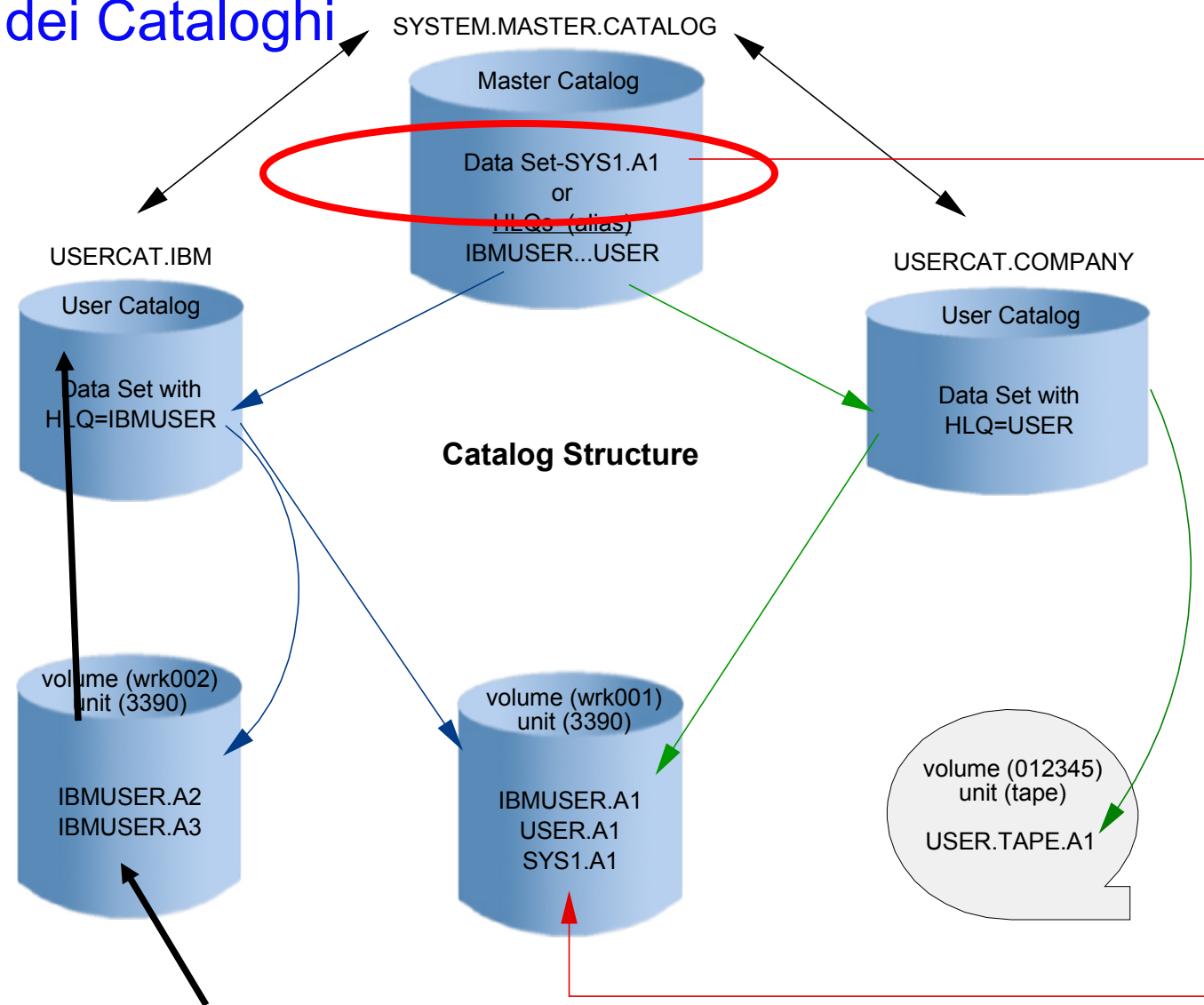
# Aliases

Gli “**aliases**” , definiti all’interno del Master Catalog, servono per associare il primo qualificatore del nome del data set , High Level Qualifier o HLQ, allo User Catalog dove sono catalogati tutti i dataset che hanno quel qualificatore.

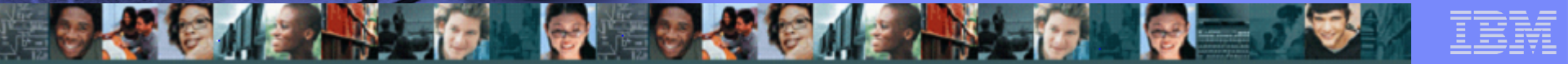




# Struttura dei Cataloghi

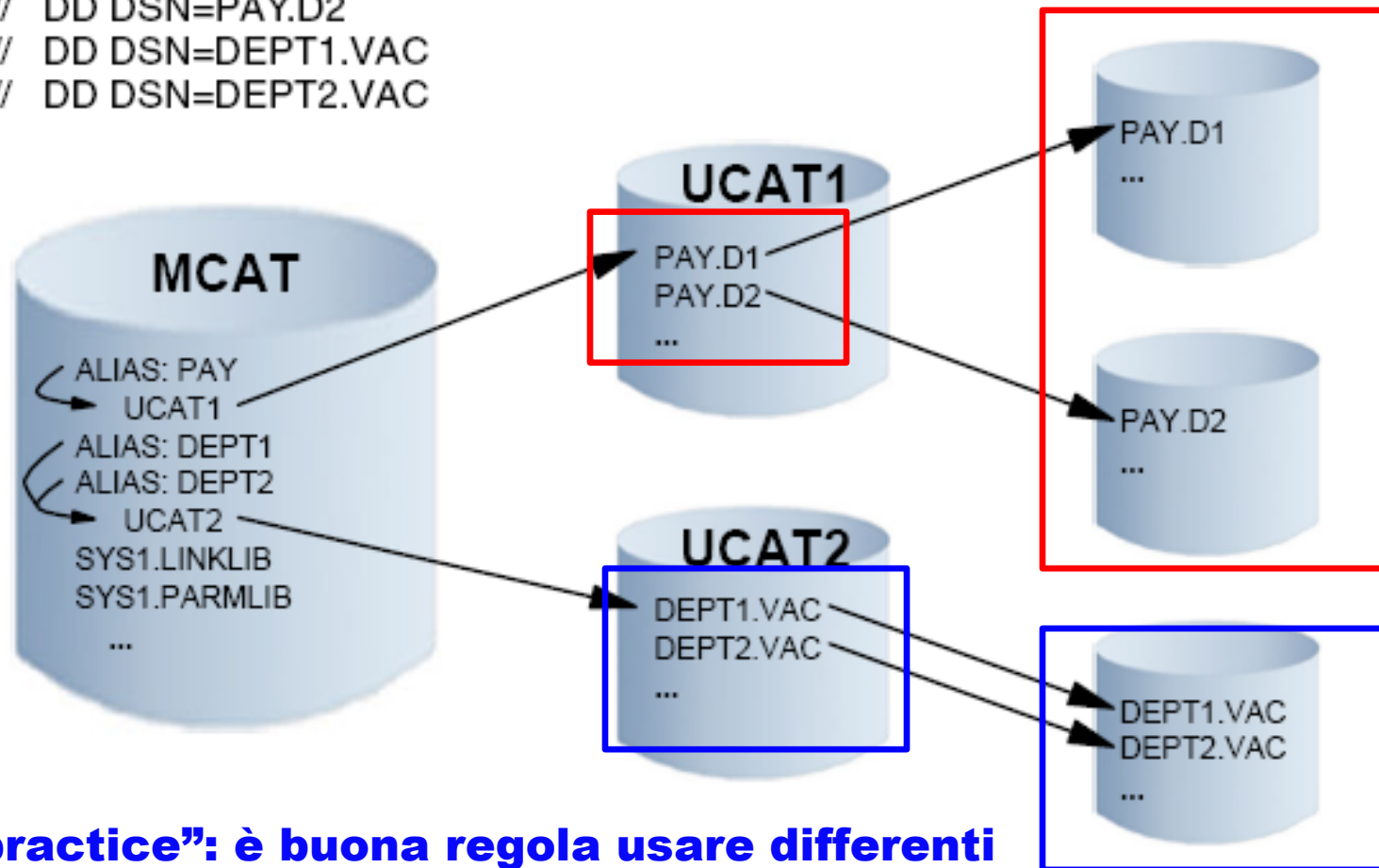


```
DEFINE ALIAS (NAME (IBMUUSER) RELATE (USERCAT.IBM ) )
```

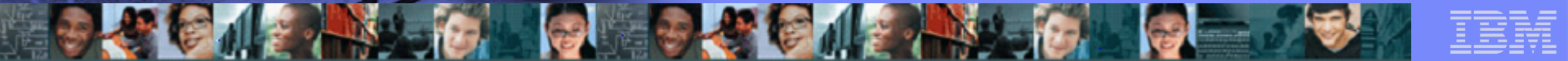


## User Catalog Alias'

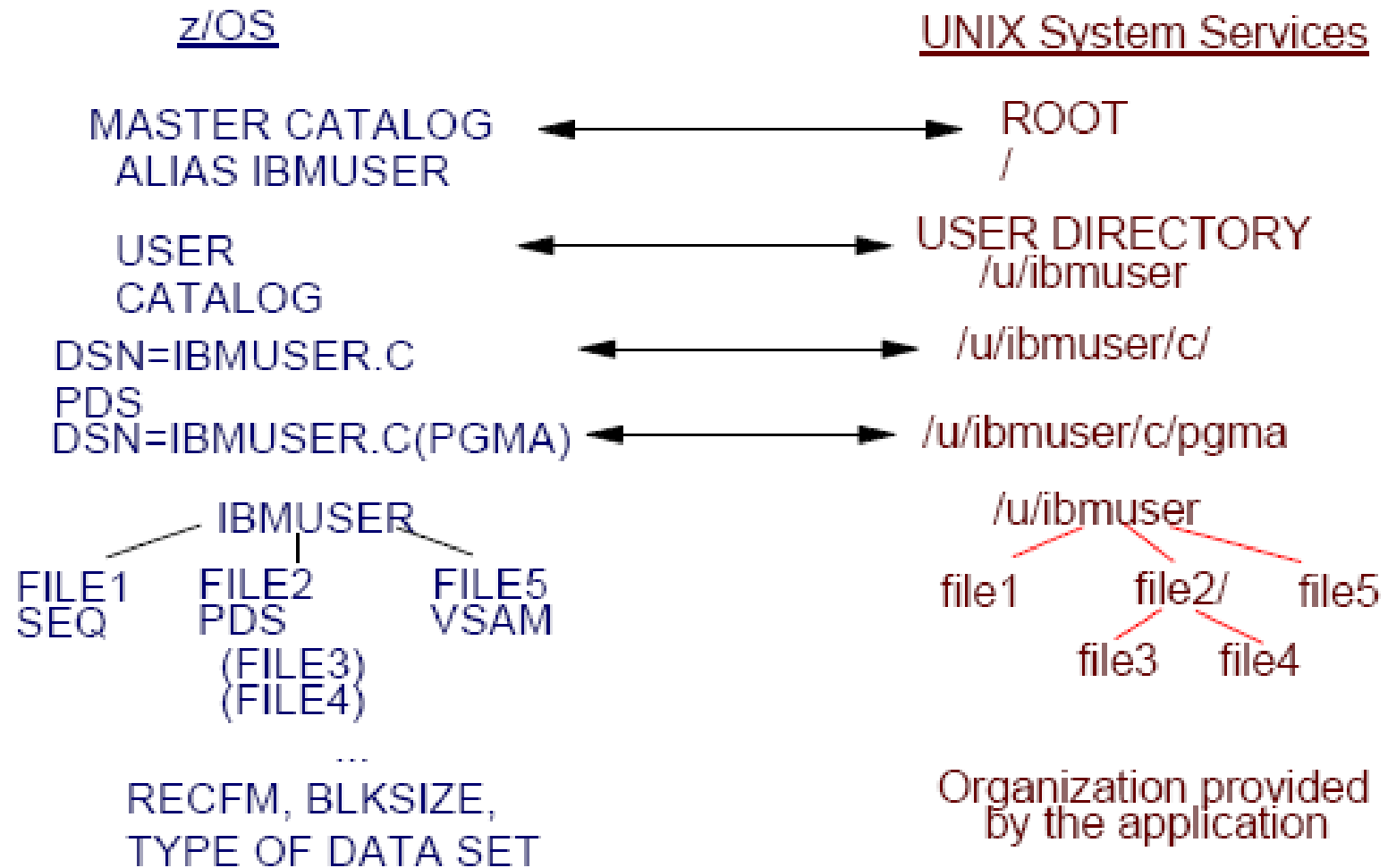
```
// DD DSN=PAY.D1  
// DD DSN=PAY.D2  
// DD DSN=DEPT1.VAC  
// DD DSN=DEPT2.VAC
```



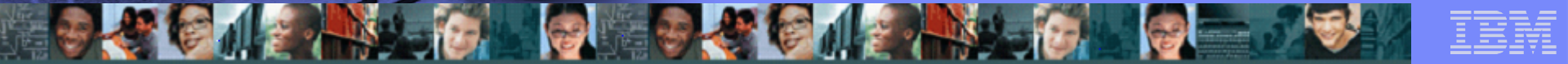
**“Good practice”: è buona regola usare differenti UCATs per differenti applicazioni**



# MVS Datasets and Unix Files



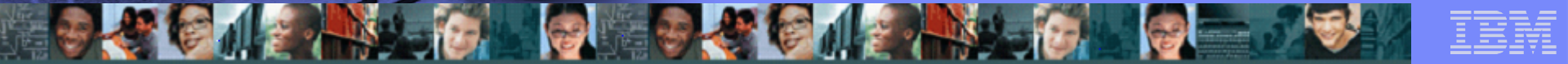




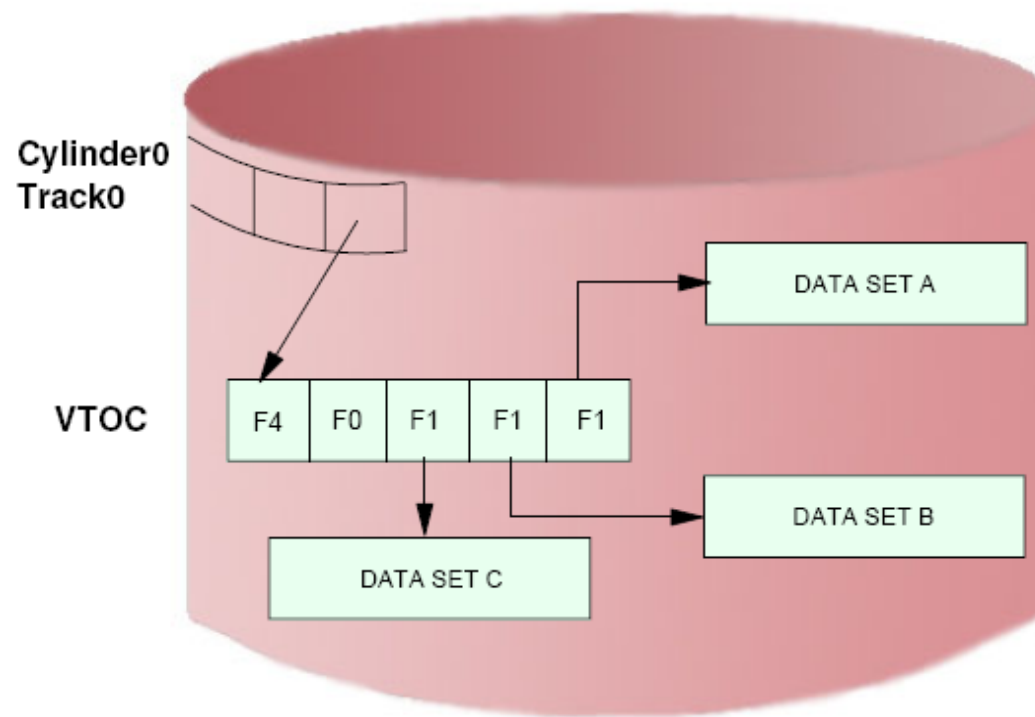
## *Domande ?*

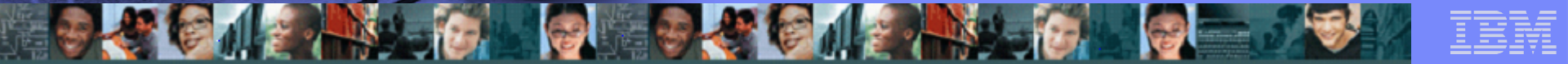




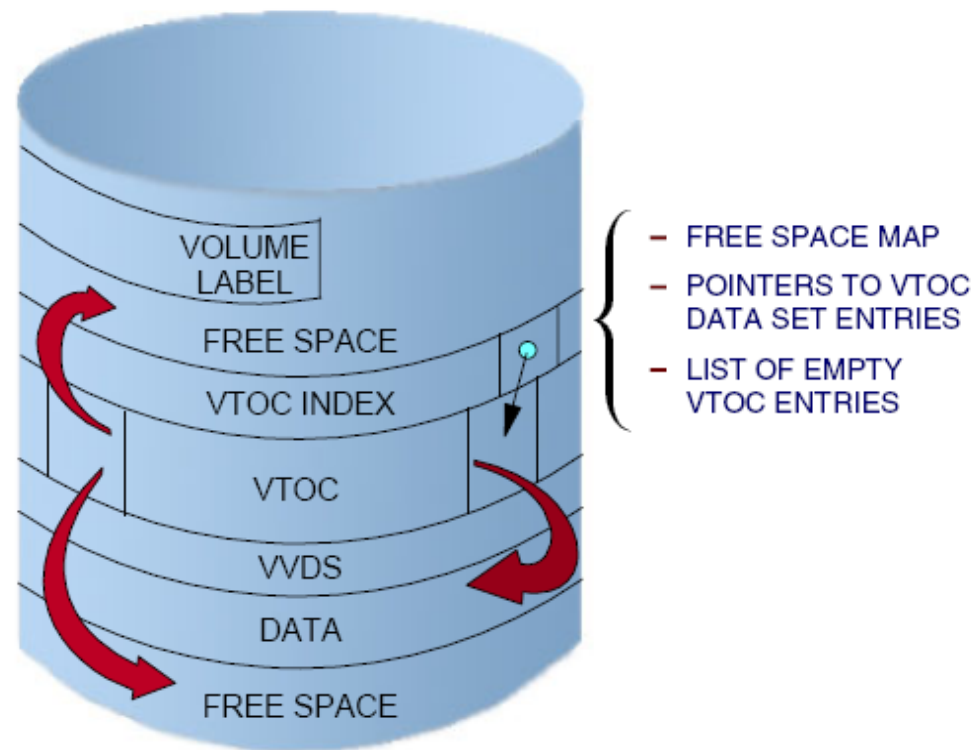


## Dataset Control Blocks (DSCB)

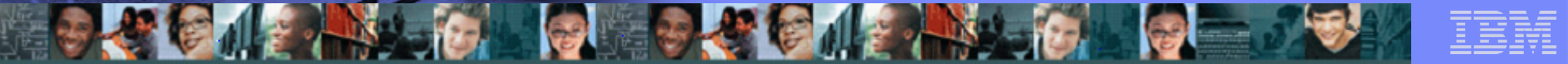




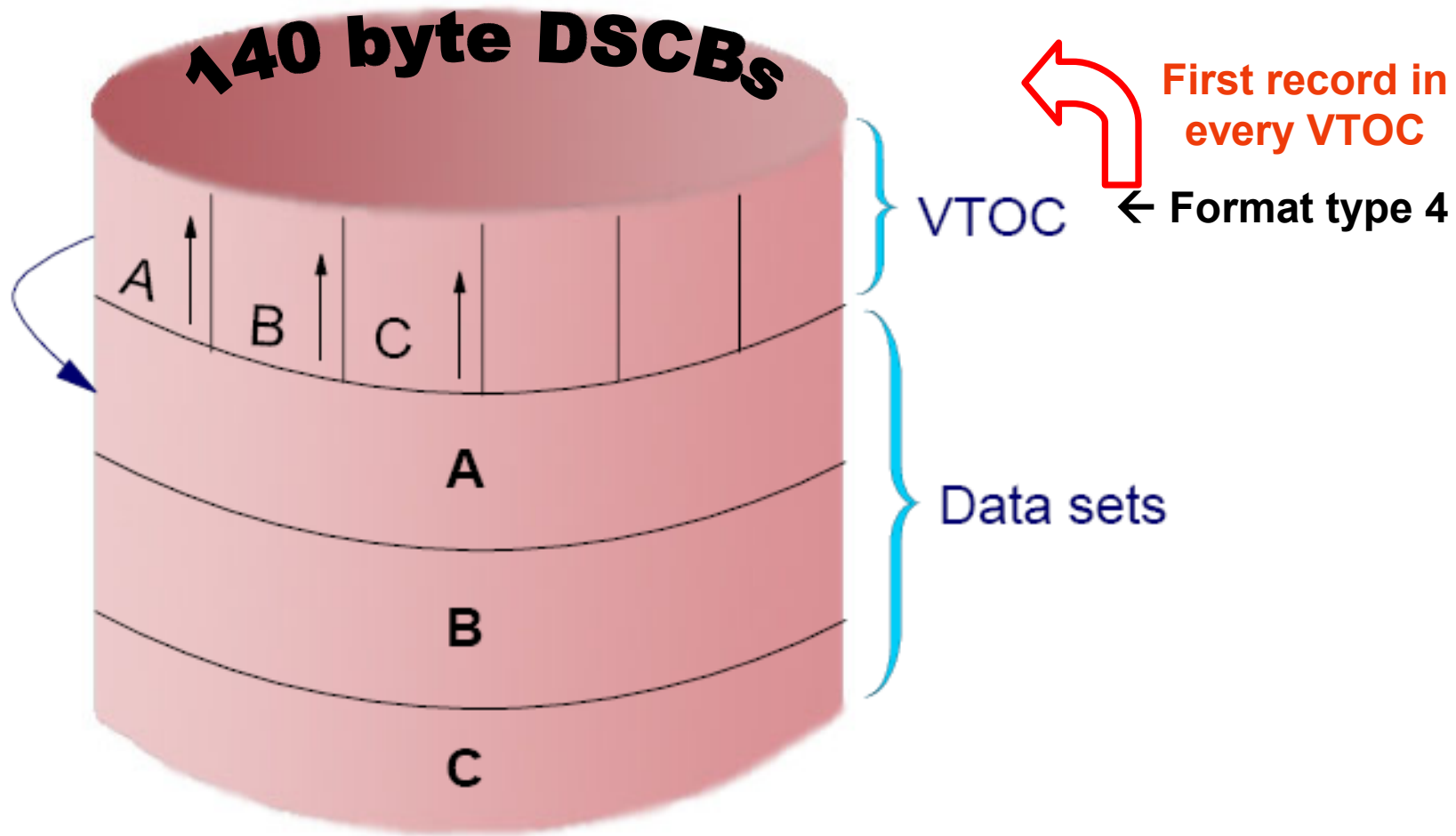
## VTOC Index Structure

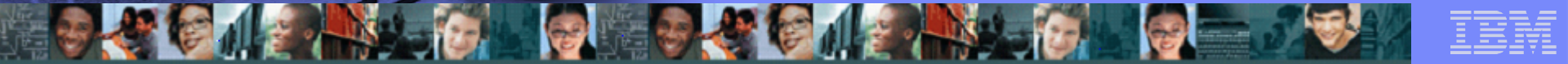


ISPF option 3.4



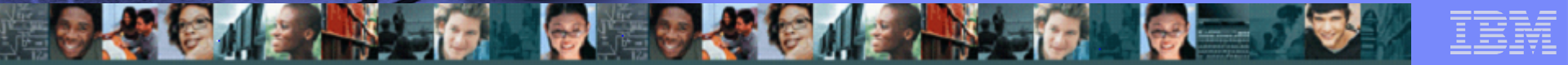
## Volume Table of Contents





## Com'è usato il catalogo

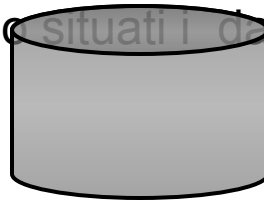
- Un catalogo associa un data set con il volume sul quale il data set is situato.
- Per localizzare il data set occorre :
  - Nome del Data set
  - Nome del Volume
  - Tipo Unità ovvero il tipo di dispositivo (volume device type)
- Il sistema z/OS normalmente include un "master catalog" and numerosis "user catalog".



# Catalog Structure

## Basic Catalog Structure (BCS) – Questo è considerato il Catalog reale

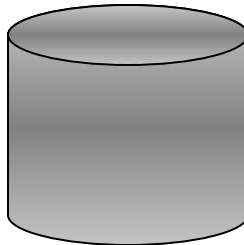
Il BCS è un VSAM KSDS and la sua funzione principale è di puntare ai volumi dove sono situati i dataset

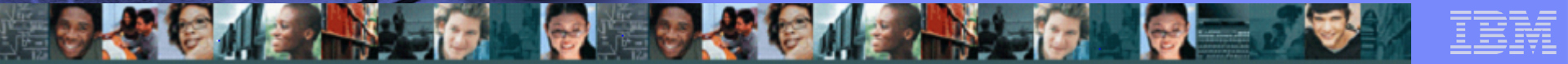


## VSAM Volume Dataset (VVDS) – puo' essere considerato una estensione della VTOC

Il VVDS è un ESDS contenente informazioni per elaborare il data set  
Contenente informazioni correlate al volume

**VTOC**





## VSAM Volume Data Set

### Tre tipi di entrate nel VVDS:

- **Un VVCR (VSAM control Record)**

Contiene informazioni su BCS i quali hanno data set su questo volume

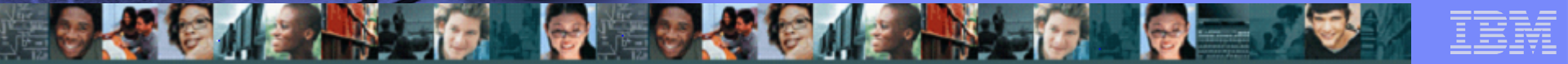
- **Piu' VVR (VSAM Volume Records)**

Contiene informazioni sui Dataset VSAM su Questo volume

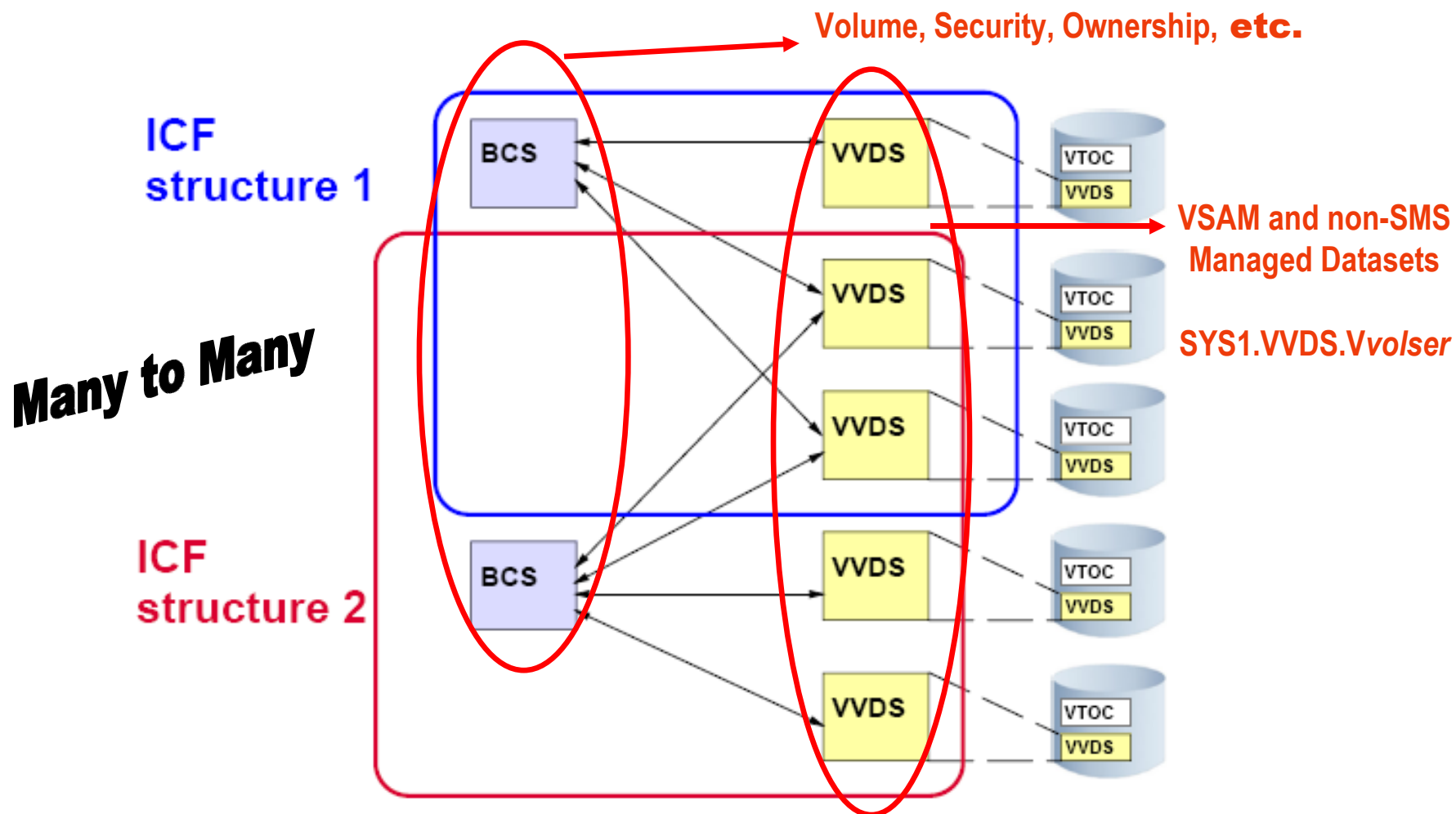
- **Piu' NVR (Non-VSAM volume record)**

Contengono informazioni su non-VSAM data Set su quel volume

VVDS è un VSAM entry-sequenced data set

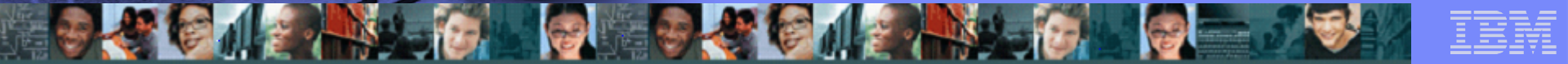


# Integrated Catalog Structure (ICF)



**Basic Catalog Structure (BCS) – Informazioni statiche che raramente cambiano**  
**VSAM Volume DataSet (VVDS) - Informazioni di catalogo addizionali**



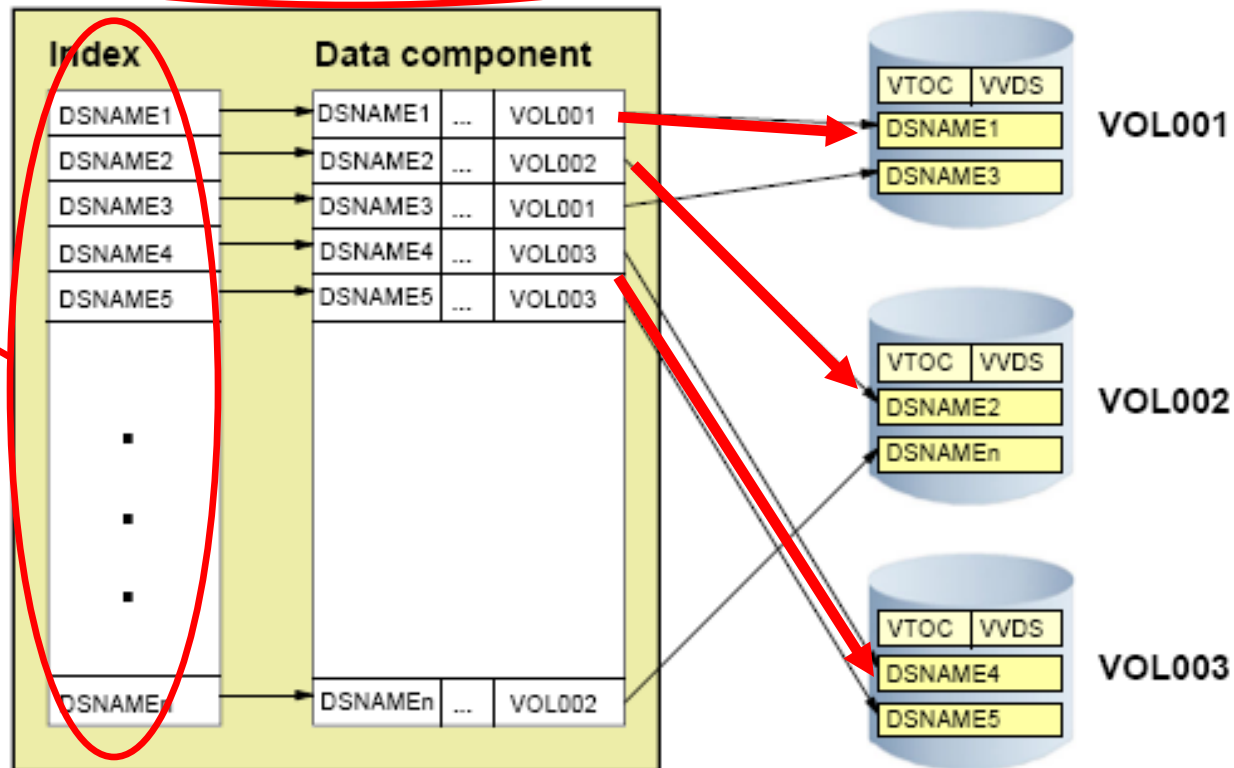


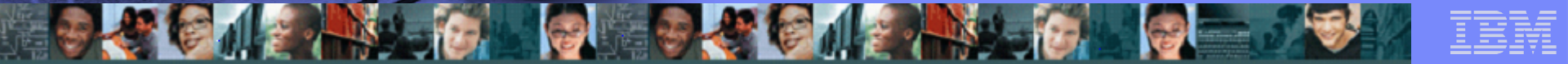
## BCS – itself is a VSAM KSDS dataset

Where DS resides: Tape, Disk,...other

### Basic catalog structure (BCS)

Uses Dataset Names as keys





## Catalog and Uncataloged Datasets

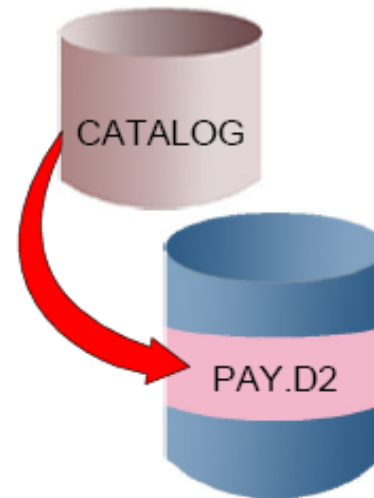
### ❑ Uncataloged reference

```
// DD DSN=PAY.D1  
DISP=OLD  
{ UNIT=3390  
VOL=SER=MYVOL
```

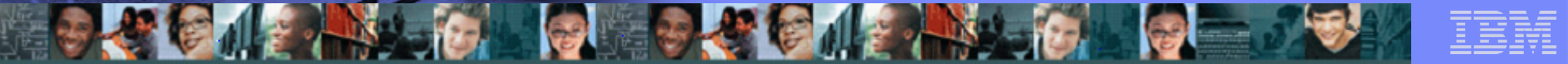


### ❑ Cataloged reference

```
// DD DSN=PAY.D2  
DISP=OLD
```



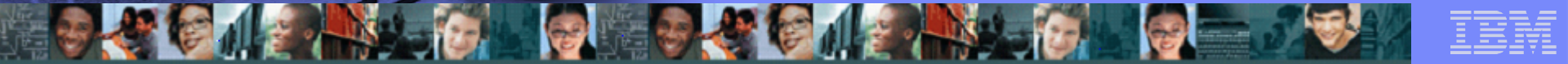
Note the ' // ' and parm statements used for Job Control Language



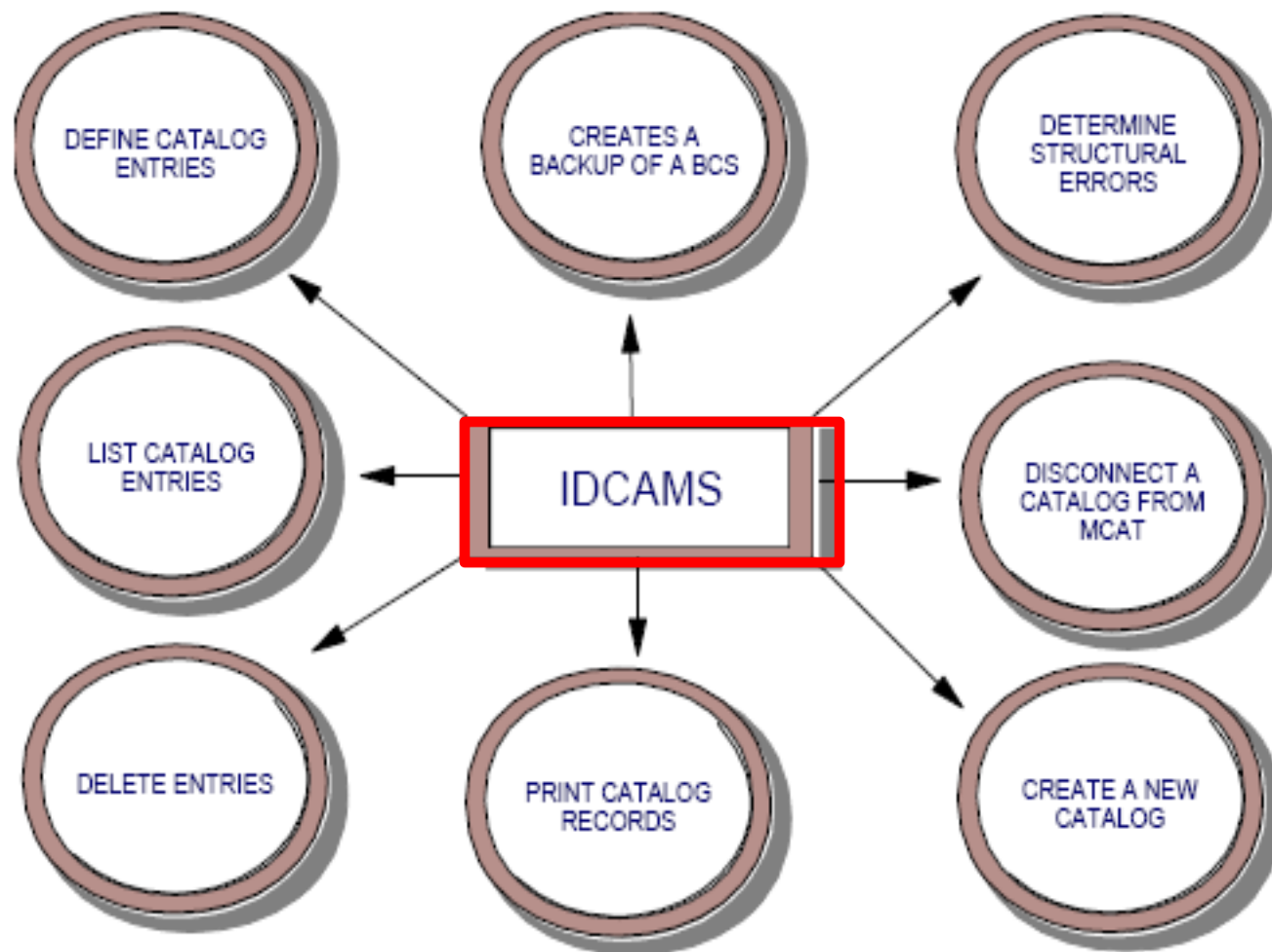
VSAM

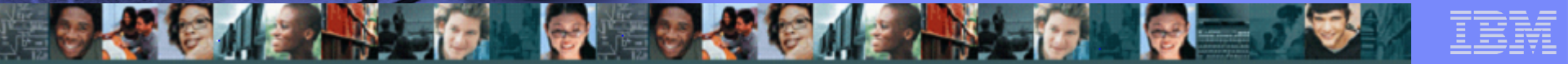
# z/OS' Access Method

- VSAM is *Virtual Storage Access Method*
- VSAM provides more complex functions than other disk access methods
- VSAM record formats:
  - Key Sequence Data Set (KSDS)
  - Entry Sequence Data Set (ESDS)
  - Relative Record Data Set (RRDS)
  - Linear Data Set (LDS)

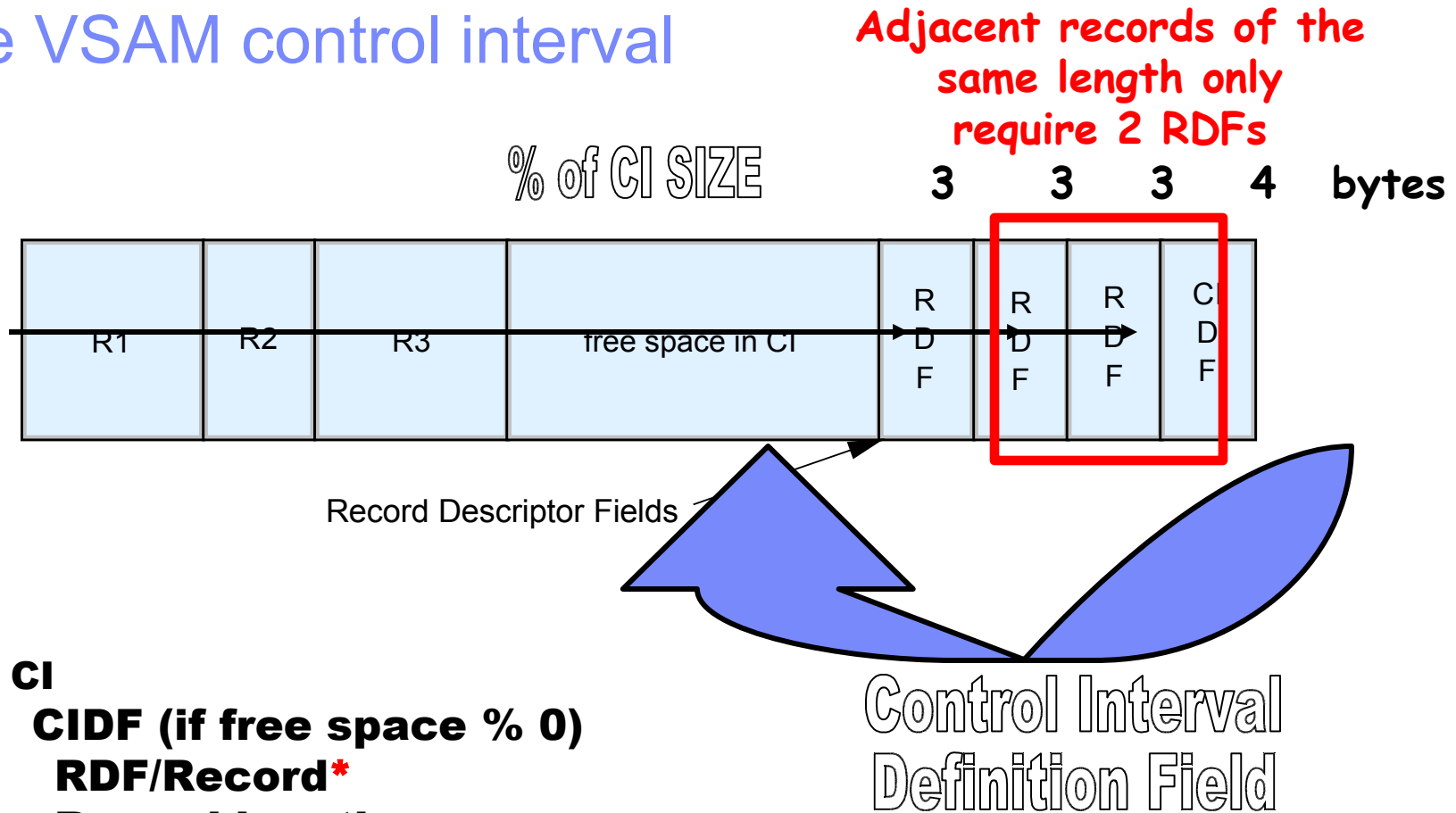


## VSAM Access Method





## Simple VSAM control interval

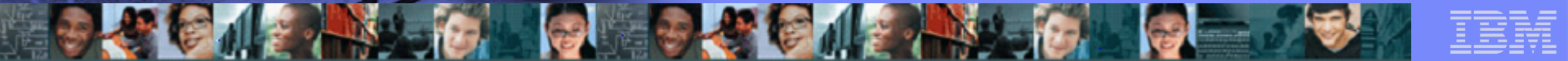


i.e 4096 = CI

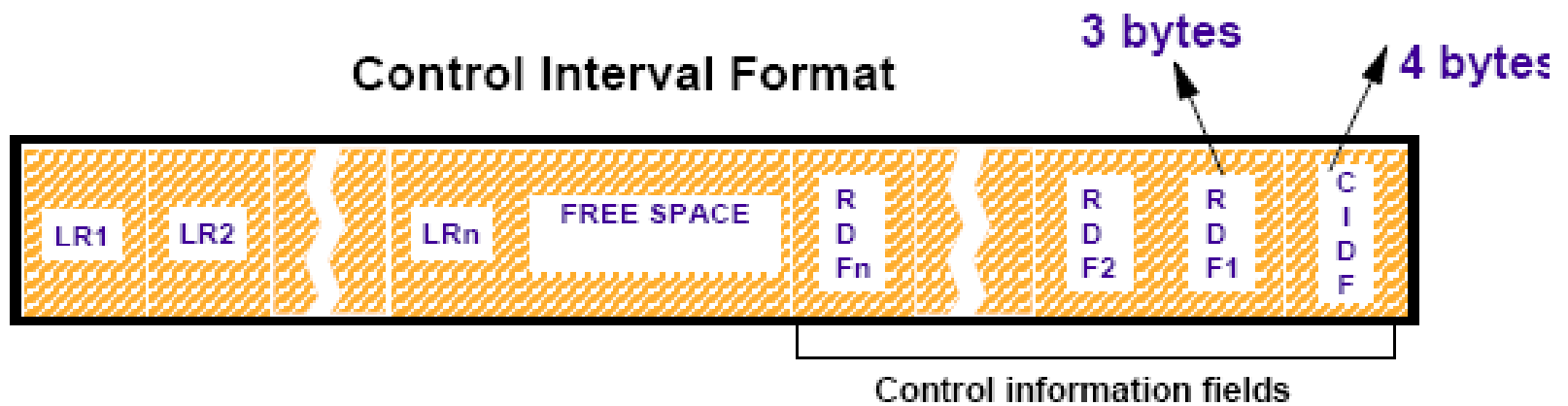
- 4 CIDF (if free space % 0)
- 2 RDF/Record\*
- 80 Record length

-----  
49 records / per CI

But.. 10% freespace = 409 bytes used for inserts (4.9 records)

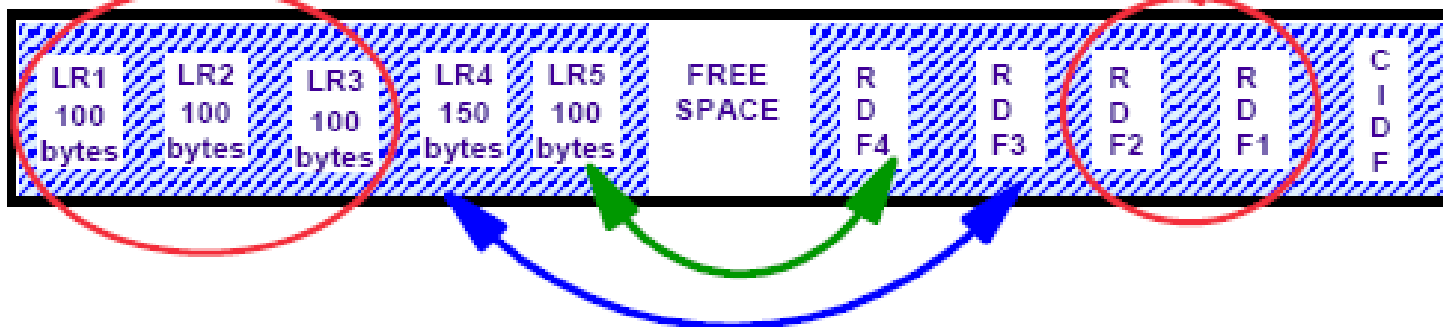


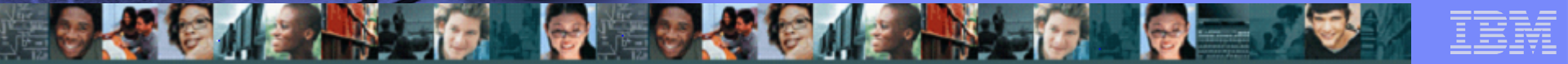
## Control Interval Format



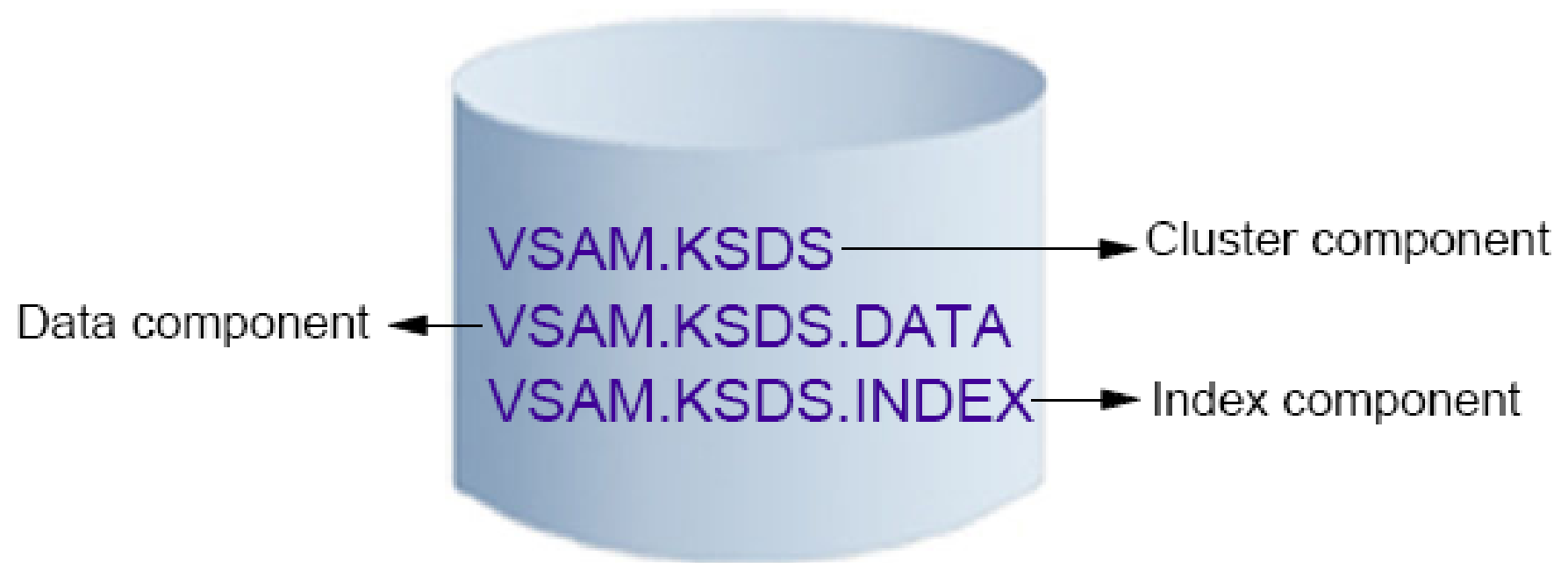
LR = Logical record  
RDF = Record definition field  
CIDF = Control interval definition field

## Contiguous records of the same size

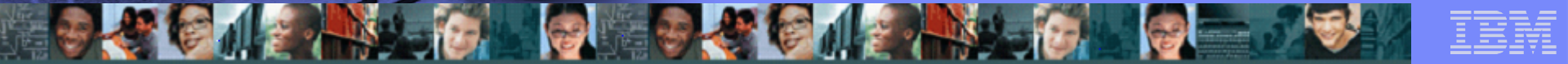




## VSAM KSDS CLUSTER

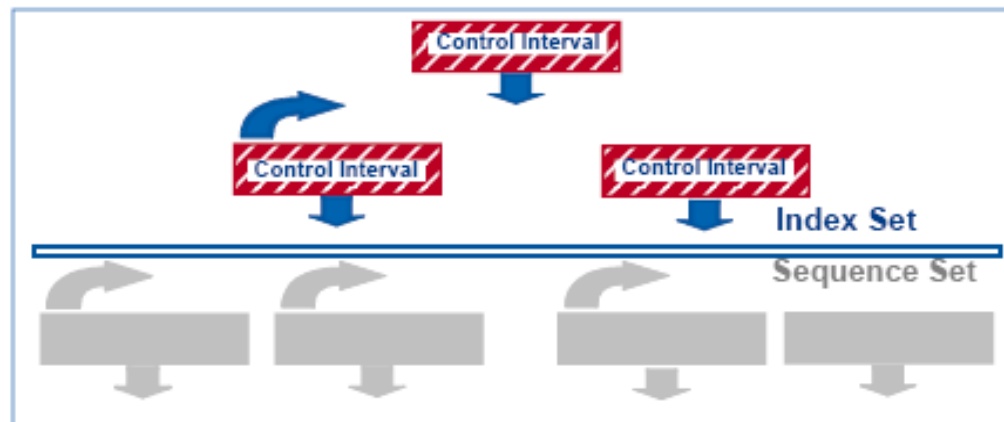




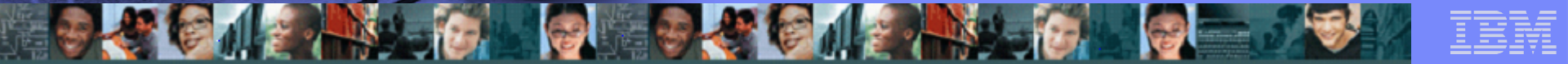


## VSAM Index Structure

### *Index Set*



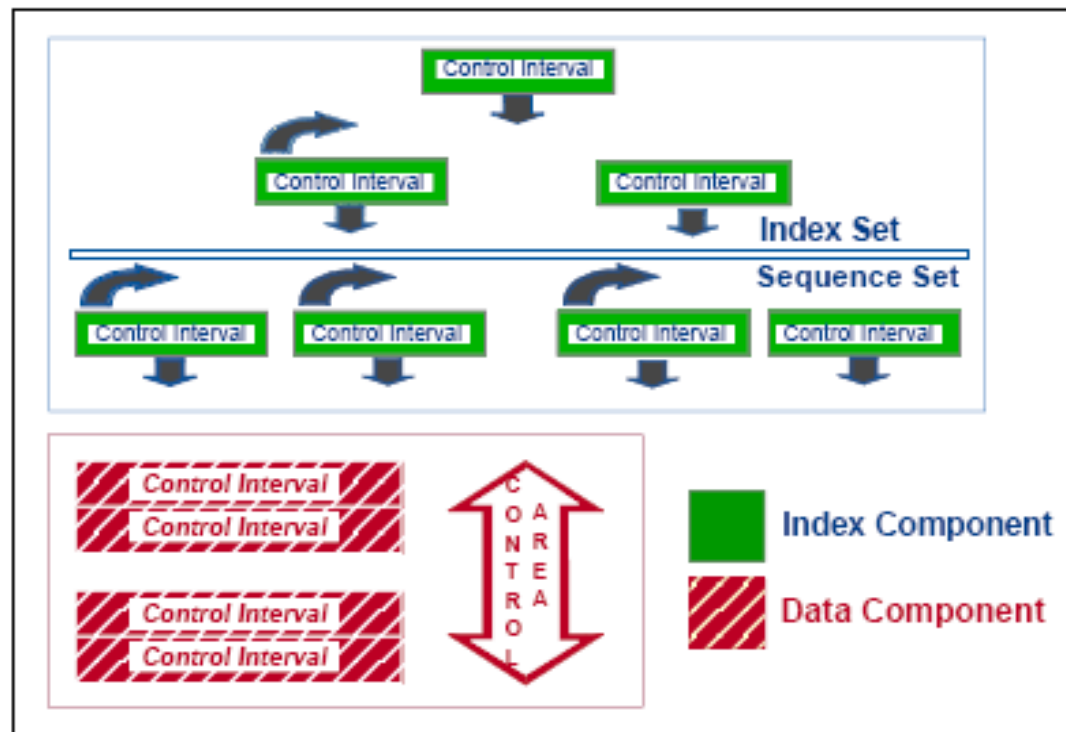
- Forward horizontal pointer at same level
- Vertical pointers to next lower level index records
- Just one CI in the top

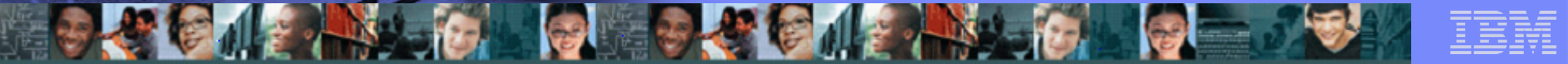


# VSAM Keyed Dataset

## KSDS Structure

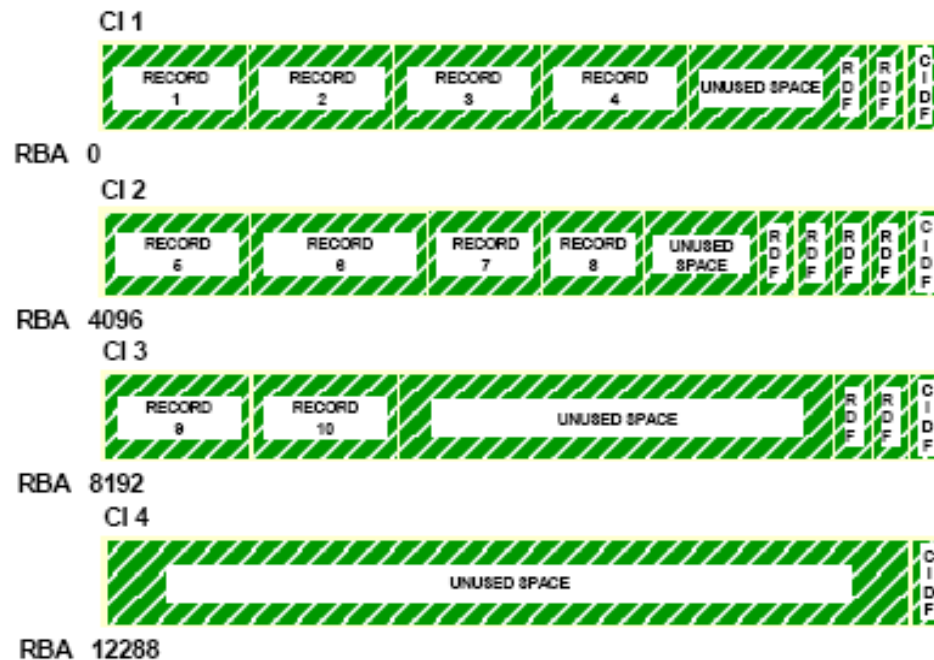
## Cluster

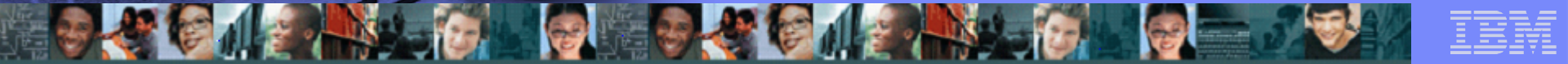




## VSAM Sequential Dataset = ESDS

### Entry Sequenced Data Set

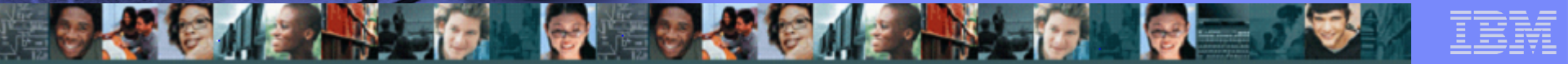




# VSAM - RRDS

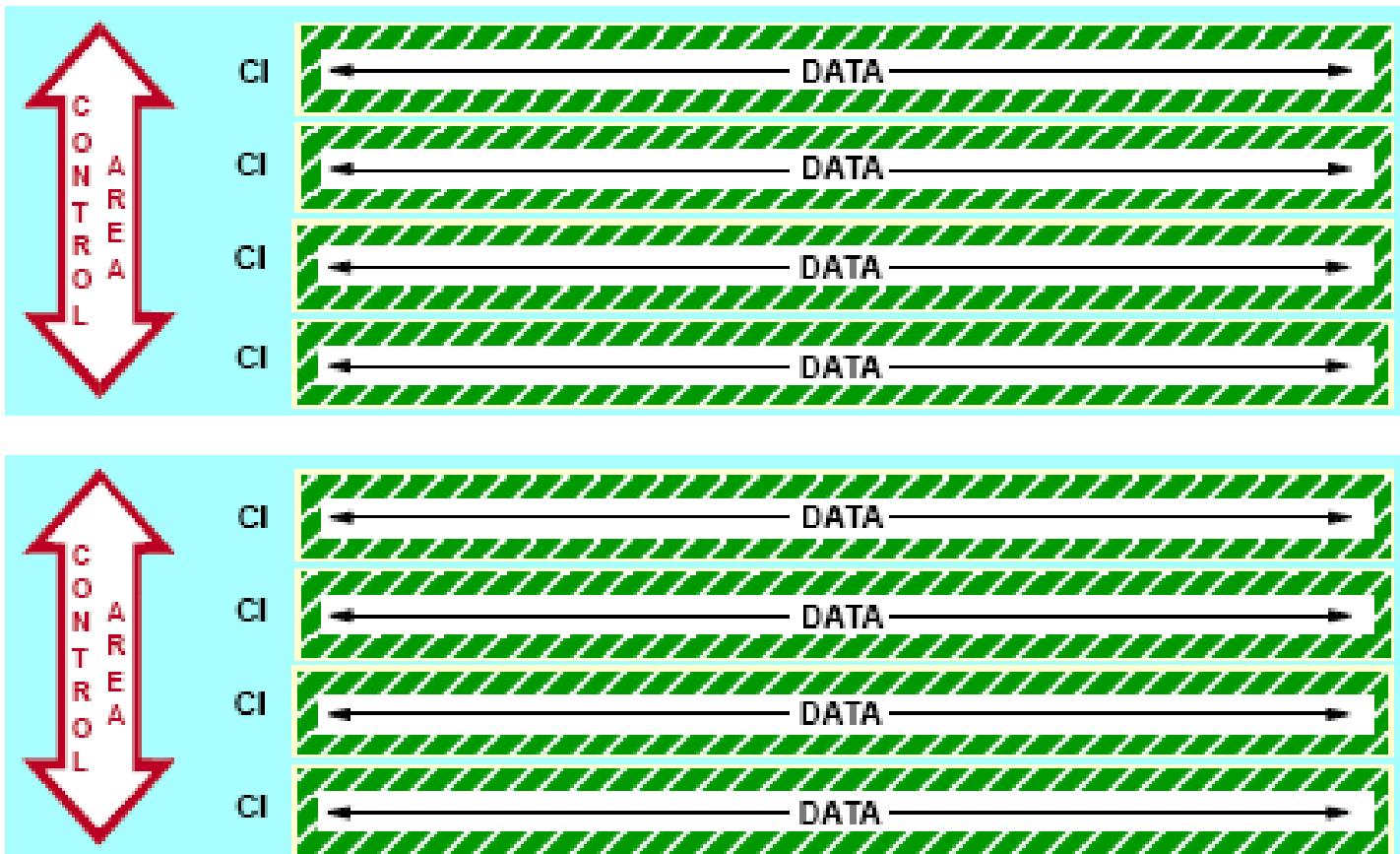
## RELATIVE RECORD DATA SET (RRDS)

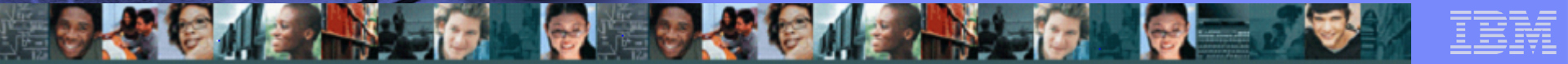




## VSAM LDS

### LINEAR DATA SET (LDS)

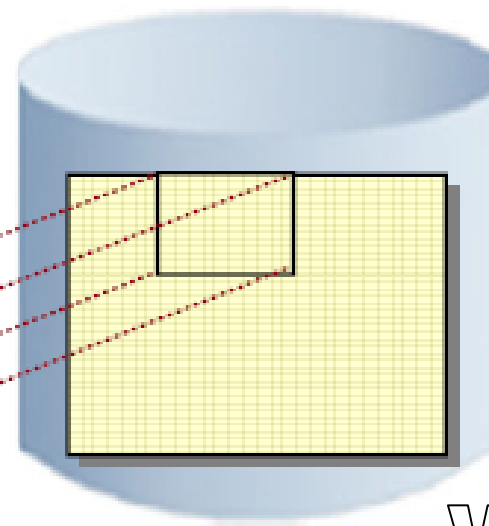
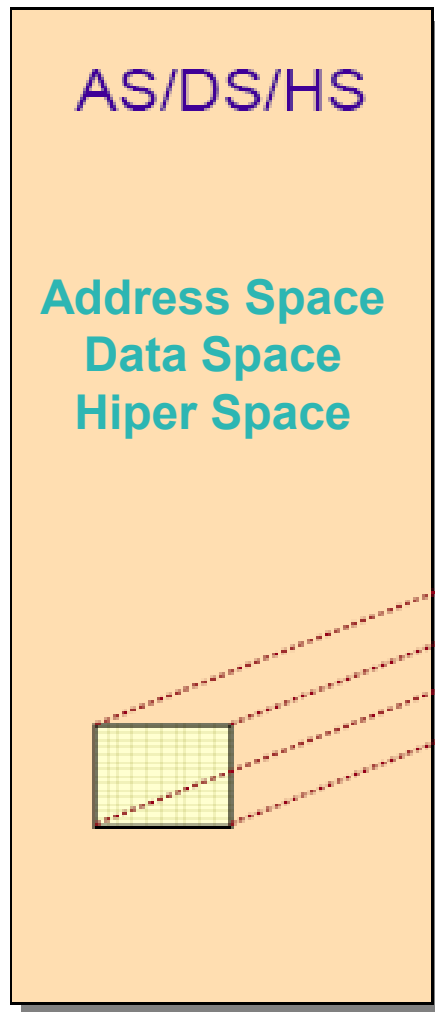




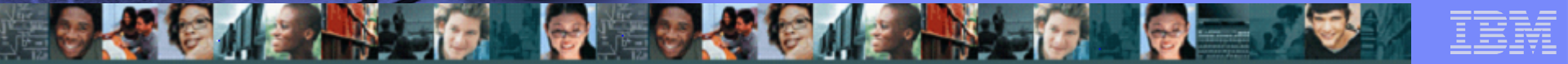
# DATA-IN-VIRTUAL (DIV)

Enable users to:

- ▶ Map data set to virtual storage
- ▶ Access data by exploiting paging algorithms



via RSM  
algorithms



# When to use which dataset type

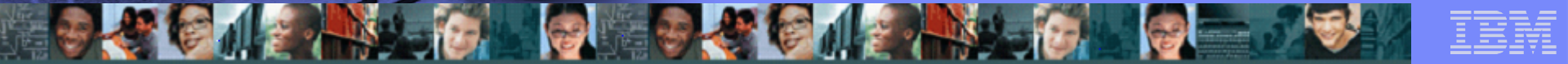
## Use KSDS if:

- The data access is sequential, skip sequential, or direct access by a key field.
- You would prefer easy programming for direct data processing.
- There will be many record insertions, deletions, and logical record length varies.
- You may optionally access records by an alternate index.
- Complex recovery (due to index and data components) is not a problem.
- You want to use data compression

## Use RRDS if:

- The record processing is sequential, skip sequential, or direct processing.
- Easy programming for direct processing is not a requirement.
- The argument for accessing data in direct mode is a relative record number, not the contents of a data field (key). RRDS is suitable for the type of logical records identified by a continuous and dense pattern of numbers (such as 1,2,3,4...).
- All records are fixed length.
- There are a small number of record insertions and deletions, and all the space for insertions must be pre-allocated in advance.
- Performance is an issue. RRDS performance is better than KSDS, but worse than QSAM or BSAM.





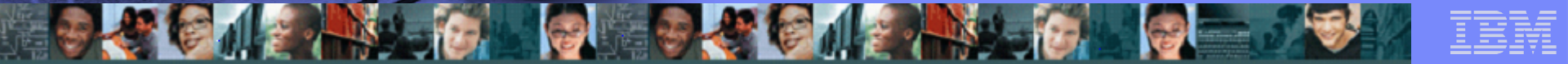
## **When to use which dataset type.....continued**

### **Use ESDS if:**

- You are adding logical records only at the end of the data set and reading them sequentially (in the application control).
- The logical record is variable length
- You seldom need direct record processing by key (using AIX).
- You are using a batch processing application.

### **Use LDS if:**

- You want to exploit DIV.
- Your application manages logical records.
- Performance is an issue.



Extra Slide – to explain Phys. Blocks

