

VSAM/QSAM/DB2原理及 基本概念

2003 / 04



- **VSAM/QSAM基本概念**
- **AMS管理实用程序**
- **DB2子系统简介**
- **DB2基本概念**

VSAM的含义和功能

VSAM (Virtual Storage Access Method)是一种虚拟存储存取方法。

负责处理程序的输入/出请求，实现处理程序所需的数据（逻辑数据）在系统主存储器（虚存）与辅助存储设备（物理数据）之间的传输。

VSAM的主要优点

- 集中管理数据文件的建立、检索、删除
- 数据文件的存取可以按直接方式和顺序方式快速实现
- 通过服务程序提供的命令，进行数据文件的定义、删除等
- 对数据文件的使用加有权限控制
- 数据可移植性
- 具有管理QSAM文件或将QSAM文件转化成VSAM的功能

数据组织和存取方法

数据组织指在辅助存储设备上存放数据记录的技术，它涉及到：选择适应数据处理要求的存储介质、记录在文件的顺序、记录长度、文件的组块因子、文件索引的使用

存取方法是帮助程序员按一种特定的数据组织在主存储器与辅助存储设备之间传输数据的一组程序。

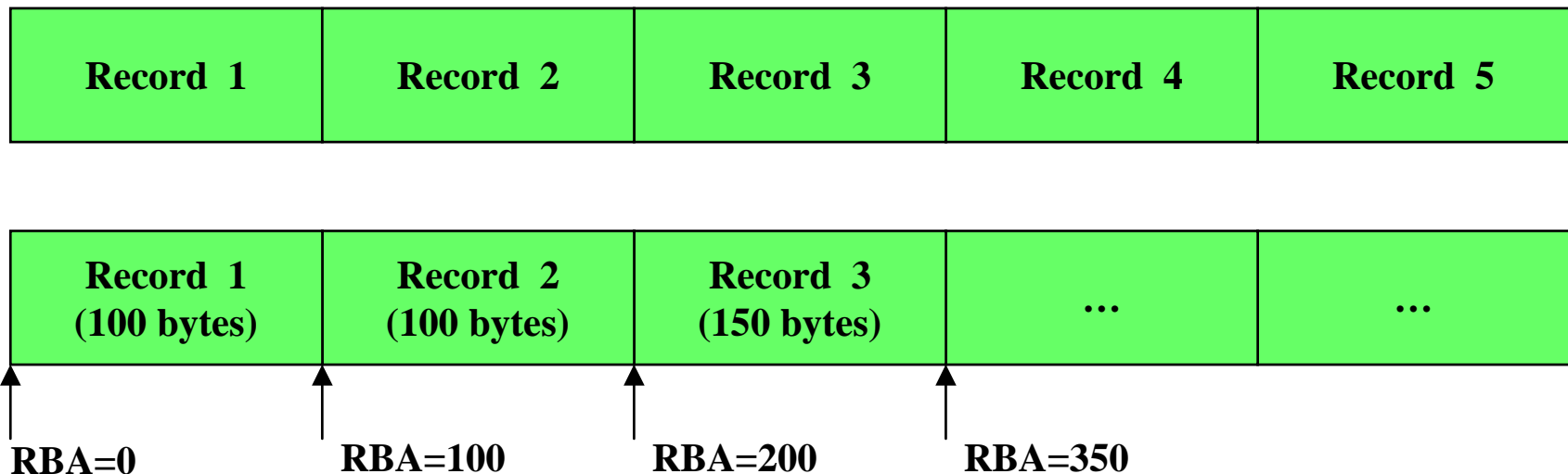
VSAM文件组织

VSAM有三种文件组织:

1. KSDS(Key-Sequenced Data Set) – 按关键字顺序组织的文件，是一种索引型的数据组织，数据记录（定长/变长）按其键值顺序排列

record	record	record	record	record	record
Key 15	Key 30	Key 88	Key 90	Key 100	Key 110

2. ESDS(Entry-Sequenced Data Set) – 按输入顺序组织的文件，是一种顺序型的数据组织，数据记录（定长/变长）按其输入顺序排列。

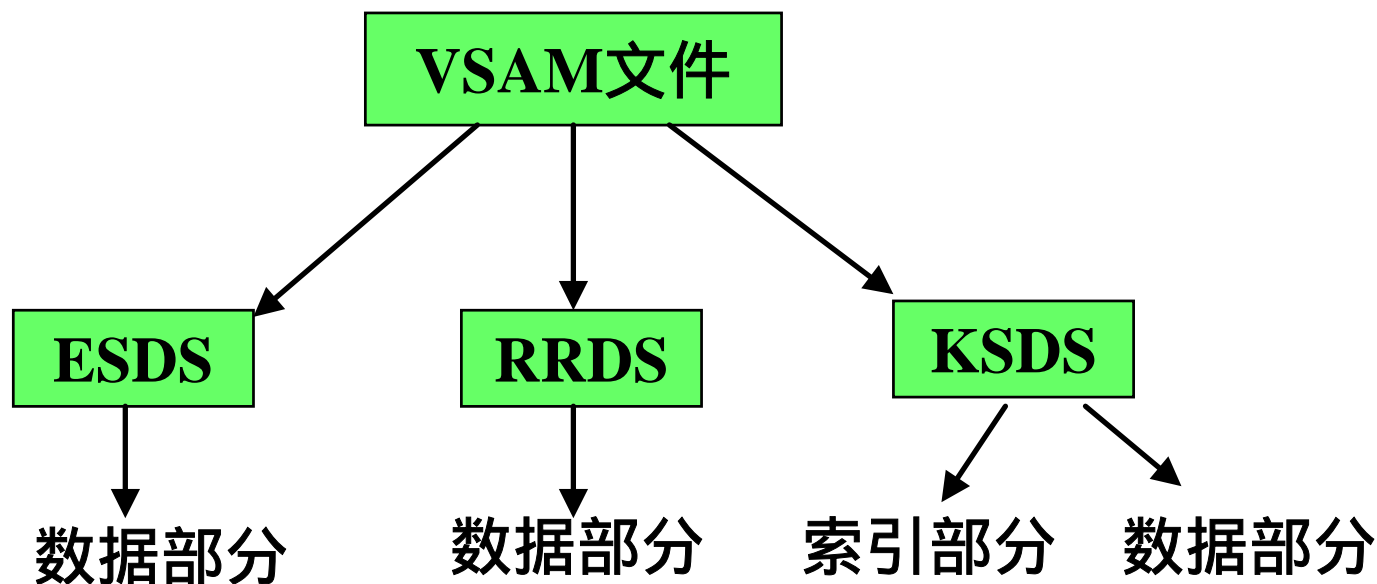


3.RRDS(Relative Record Data Set) – 相对记录文件,是一种直接型的数据组织,数据记录(定长)按其相对记录号RRN存放在对应的槽(slot)中

RR 1		RR 3	RR 4		RR 6
SLOT 1	SLOT 2	SLOT 3	SLOT 4	SLOT 5	SLOT 6

VSAM文件组织

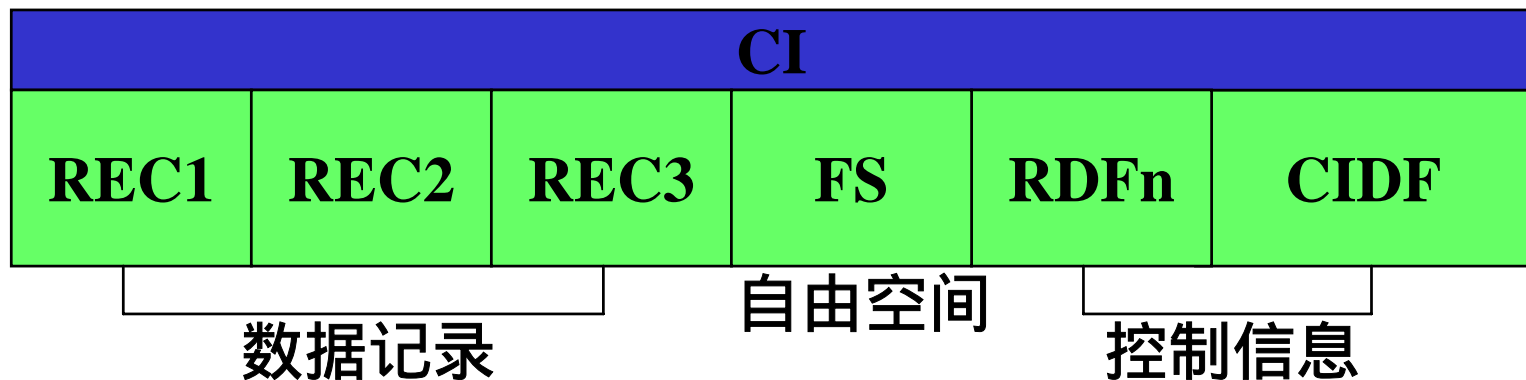
在VSAM中，文件都称为“cluster”。



VSAM文件物理结构和逻辑结构

1.CI (Control Interval-控制区间)

CI是VSAM在虚存和磁盘之间传输数据的单位，数据文件的各个记录都放在CI中。CI中除记录外还有一些VASM对CI进行管理所需的控制信息。

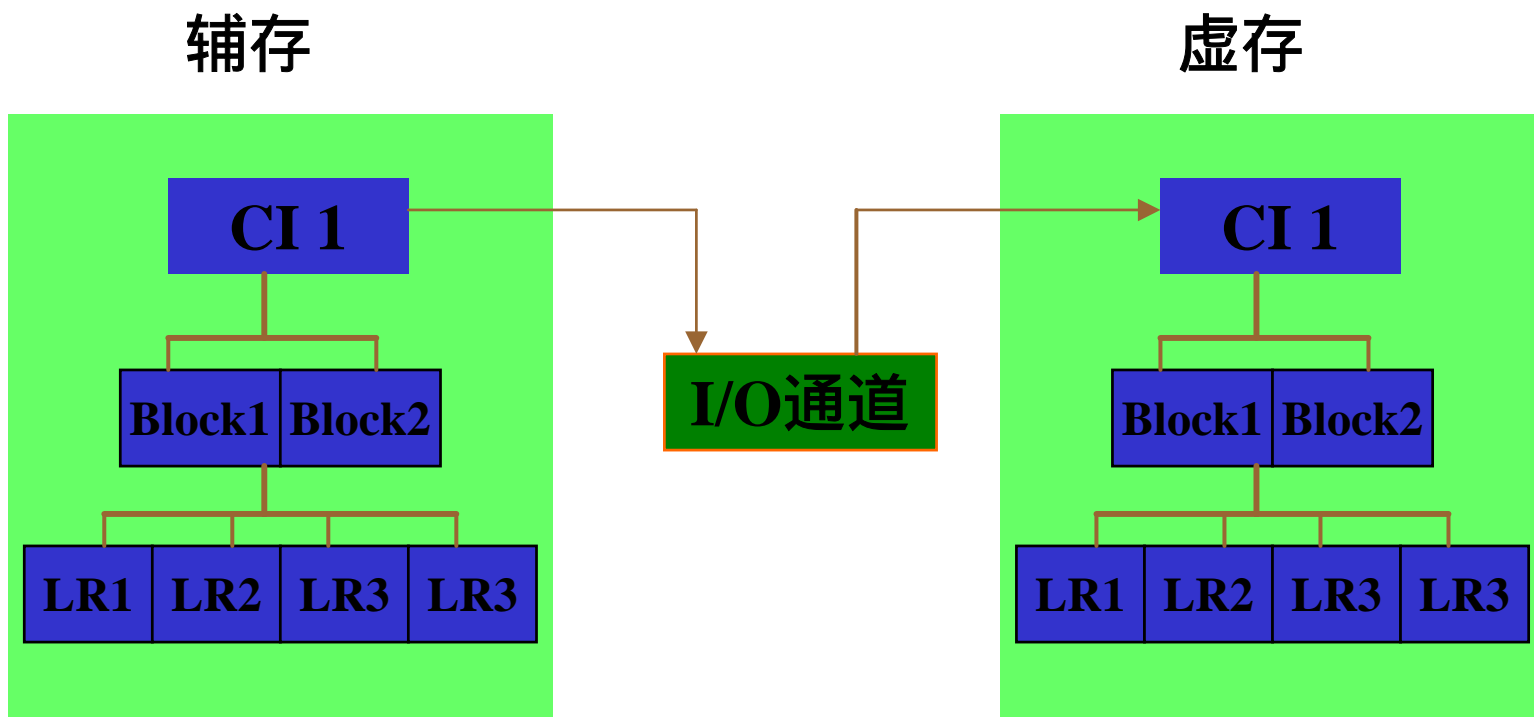


2.CI 特性

- CI大小：** 每个VSAM文件中的CI大小可以不同,但在同一VSAM文件中，CI大小必须相同。
- CI取值范围：** 0.5K-32K,必须是512byte的倍数,如CI大小超过8192byte,则必须是2018byte的倍数。
- CI的物理块：** 一个CI可以包含一个或多个物理块，物理块的长度由VSAM根据CI大小和磁盘的类型来确定。

3.传输单位

对于VSAM，数据在虚存和辅存之间传输的单位是CI。



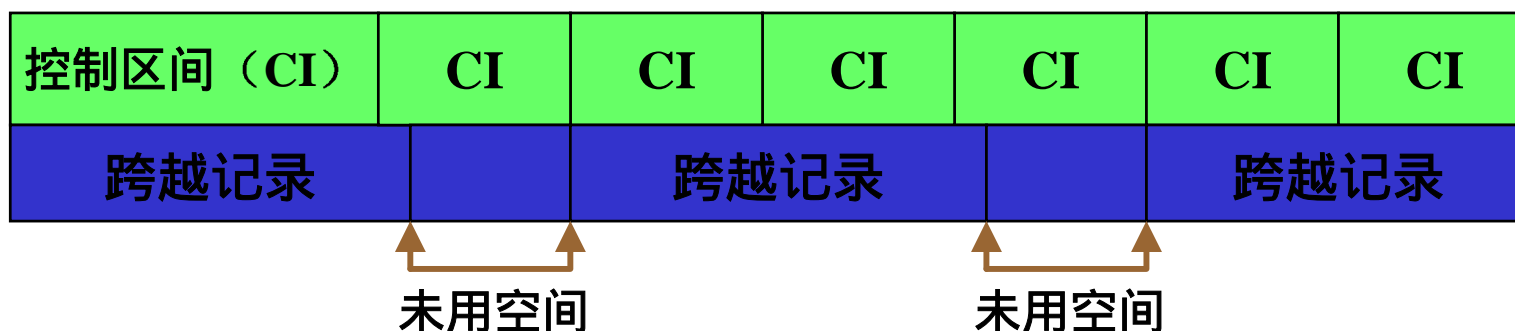
4.CA（Control Area-控制域）

CA是比CI更大的数据组织单位，若干个CI构成一个CA；
在一个VSAM中，每个CA中包含的CI个数是相同的；CA的大小由VSAM确定。

一个VSAM文件可由一个或多个CA组成。

5.跨越记录(Spanned Record)

VSAM允许KSDS和ESDS文件的数据记录长度超出CI，每个跨越记录必须起始于一个新的CI；KSDS文件通过键值访问跨越记录；ESDS文件通过RBA访问跨越记录。



6.VSAM文件物理结构

一个VSAM 由若干CA组成，其中包含有作为自由空间的CA，扩充VSAM以CA为单位;CA由若干CI组成，其中包含作为自由空间的CI;CI由若干逻辑记录、自由空间和控制信息组成。

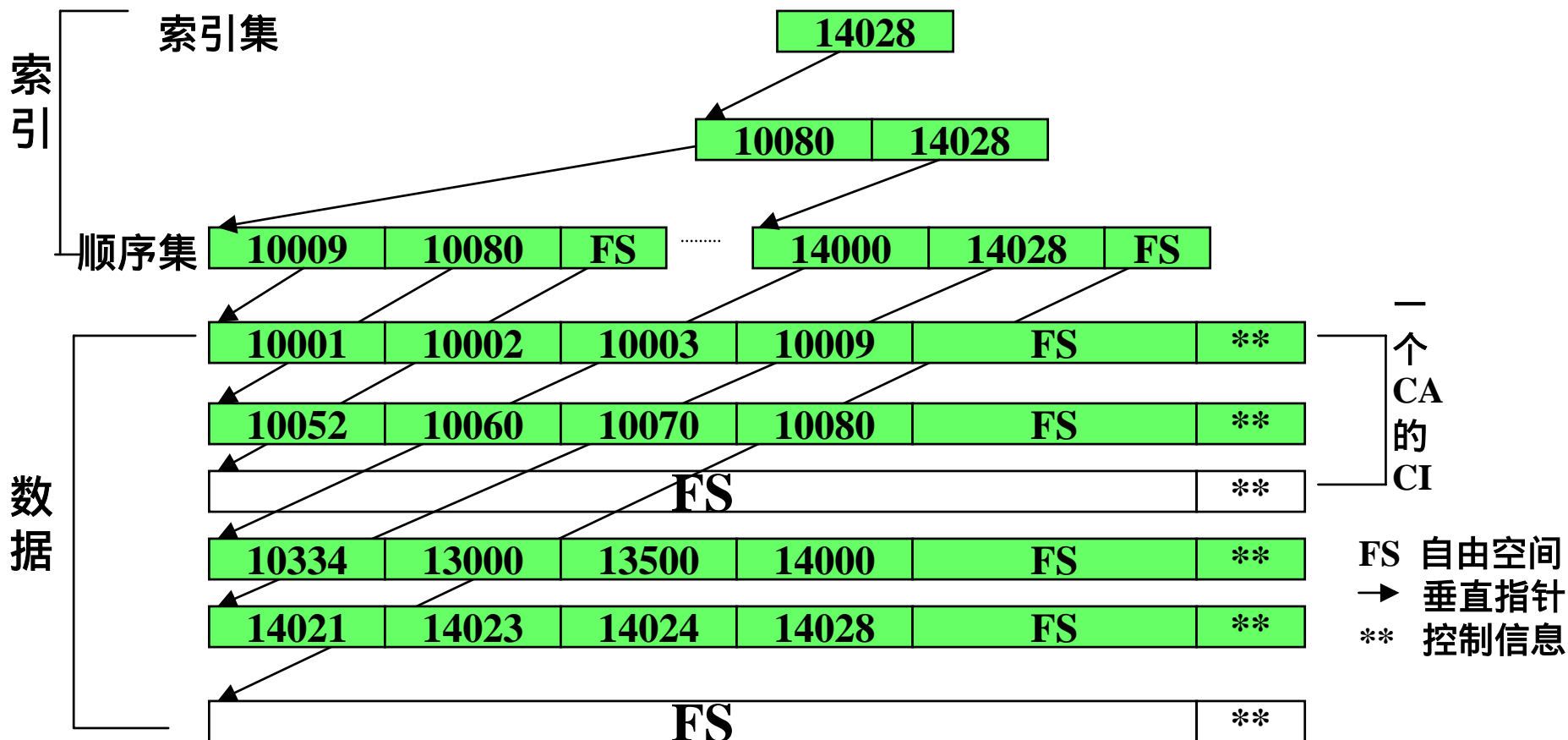
VSAM支持三级自由空间：CI内部为第一级； CA中的某一个CI 为第二级； 整个CA 为第三级

7.KSDS的文件结构

KSDS由两大部分构成，一部分是真正的文件记录所在，称为数据部分（DATA），另一部分是记录指针所在，称为索引部分（INDEX），它又由索引集和顺序集构成。

其逻辑结构如下：

KSDS的VSAM索引逻辑结构



8.ESDS的文件结构

在ESDS中，只有数据部分（DATA），没有索引部分，CI中不预留自由空间。

9.RRDS的文件结构

RRDS的CI被分为若干个定长的槽，每个槽分配一个相对记录号，记录根据其相对记录号存放在对应的槽中。

VSAM文件的访问方式

对于KSDS:

- 顺序访问 (Sequential)
- 直接访问 (Driect)
- 跳跃式顺序检索

对于ESDS:

- 顺序访问 (Sequential)
- 直接访问 (Driect)

对于RRDS:

- 顺序访问 (Sequential)
- 直接访问 (Driect)

CI 和CA的分裂

VSAM向CI中写记录时总是保证CI内各记录的键值按升序排列；若要插入一个新记录或增加现有记录长度时，第一级自由空间不够，则发生CI分裂；若第二级自由空间不够（没有空的CI），则发生CA分裂。

CI和CA分裂对于顺序处理、直接处理采用不同的方式。

CI分裂

10007

插入记录

10009	10080	10333	FS
-------	-------	-------	----

顺序集SS

10003	10009	10080	10333
-------	-------	-------	-------

10001	10002	10003	10006	10009		**
-------	-------	-------	-------	-------	--	----

10052	10060	10070	10080	FS	**
-------	-------	-------	-------	----	----

10222	10250	10300	10333	FS	**
-------	-------	-------	-------	----	----

FS	**
----	----

10001	10002	10003	FS	**
-------	-------	-------	----	----

10006	10007	10009	FS	**
-------	-------	-------	----	----

C
A

替换索引和路径

替换索引（ALTERNATE INDEX）

在一个KSDS或ESDS文件之上，VSAM可建立一个或多个替换索引，从而以另外一种方式对现有文件进行存取，替换索引提供了对同一数据实现不同存取的唯一方法。

替换索引和路径

路径（PATH）

为了通过替换索引存取基文件时，在替换索引与基文件之间必须定义路径。这个路径就是将某一替换索引与基文件建立联系，路径是定义在替换索引之上的；它不占据任何数据空间，仅在VSAM目录中有一条目加以描述。

存取方法服务（AMS）

AMS是一个功能很强的实用性程序，它能够利用VSAM的数据管理技术来建立和维护文件；AMS提供一组命令，VSAM用户正是通过AMS命令来调用AMS的各种功能，从而实现相应的服务。

AMS功能命令

DEFINE

DEFINE MASTERCATALOG	定义VSAM主目录
DEFINE USERCATALOG	定义VSAM用户目录
DEFINE SPACE	定义VSAM数据空间
DEFINE CLUSTER	定义VSAM文件
DEFINE NONVSAM	定义非VSAM文件
DEFINE ALTERNATEINDEX	定义替换索引
DEFINE PATH	定义路径

```
DEFINE CLUSTER ( -  
    NAME (VSAMKSDS) –  
    VOL (SYSWK1) –  
    RECORDS (15000 1000) ) –  
    DATA ( -  
        NAME (DATA1) –  
        KEYS (10,5) –  
        RECORDSIZE (200 200) ) –  
    INDEX ( -  
        NAME (INDEX1) ) –  
    CATALOG (USER01.CAT)
```

DEFINE NONVSAM -

NAME (NONVSAM) –

VOL (SYSWK1) –

DEVICETYPE (3390) –

CATALOG (USER01.CAT)

DEFINE AIX -

NAME (KSDSAIX) –

RELATE (KSDSVSAM) –

KEY (10,0) -

VOL (SYSWK1) –

CYL (3,1) –

NONUNIQUEKEY -

CATALOG (USER01.CAT)

BLDINDEX -

INDATASET (KSDSVSAM) –

OUTDATASET (KSDSAIX) –

CATALOG (USER01.CAT)

DEFINE PATH (-

NAME (PATH1) –

PATHENTRY (KSDSAIX) –

CATALOG (USER01.CAT)

ALTER （修改VSAM各种定义属性）

ALTER DATA1 –

FREESPACE (20 0) –

CATALOG (USER01.CAT)

可修改的属性包括：文件的名字、口令和有关文件保护的参数、CI和CA中自由空间的百分比

DELETE （删除VSAM各种定义）

DELETE (VSAMKSDS VSAMESDS VSAMRRDS) –

CLUSTER –

PURGE –

EARSE –

CATALOG (USER01.CAT)

VERIFY（用来校正目录中有关文件状态的信息，以恢复对文件的存取）

VERIFY DATASET (VSAMKSDS) –

REPRO (用于文件的拷贝和文件的重新组织)

**REPRO INFILE (VSAMKSDS) –
OUTFILE (QSAM)**

LISTCAT: 列出目录内容
PRINT : 打印VSAM或非VSAM文件
IMPORT
EXPORT] 建立文件和用户目录的备份, 以实现系统间的传输
CANCEL: 取消一个作业或当前作业步
BACKUP: 用于VSAM文件建立具有QSAM特性的备份文件
RESTORE: 恢复由BACKUP命令所建立的备份文件
LISTCRA : 用于列出目录恢复区 (CRA) 的内容, 以检验目录和CRA是否同步
EXPORTRA: 从CRA读出目录条目和数据
IMPORTRA: 恢复由EXPORTRA读出的目录条目和数据
RESETCAT: 恢复可恢复目录使CRA信息与目录信息同步



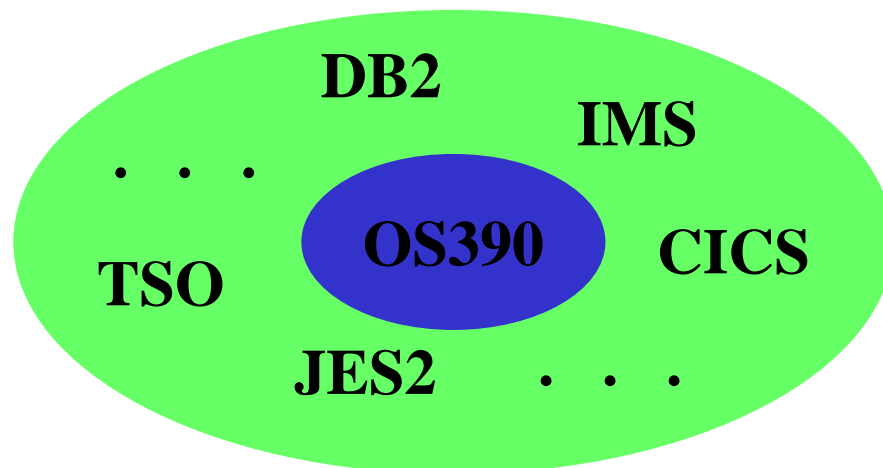
HISUN 高阳科技



谢谢!

DB2与OS/390的关系

DB2是基于IBM主机OS390操作系统的一个子系统，并且是一个关系数据库管理系统，它与CICS、IMS、TSO协同工作，通过CICS、IMS、TSO对DB2进行操作，所以DB2的应用可以分为CICS应用、IMS应用、TSO应用。其中CICS应用、IMS应用必须是联机的，TSO应用则既可以是联机的也可以是批处理的。



关系数据库

在关系数据库中，数据是以表的形式体现，所以说DB2是表的集合，并且DB2能够保证数据的一致性、完整性、安全性。应用系统通过SQL语句对数据库进行操作，其结果是从旧表的数据产生新表的数据。

举例:

S	S#	SNAME	CITY
	S1	smith	london
	S2	jones	paris
	S3	blake	paris
	S4	clark	london

P	P#	PNAME	COLOR	WEIGHT	CITY
	P1	nut	red	12	london
	P2	bolt	green	17	paris
	P3	screw	blue	14	rome
	P4	cam	red	12	london

SP	S#	P#	QTY
	S1	P1	300
	S1	P2	200
	S1	P4	400
	S2	P2	100
	S2	P3	300
	S3	P2	400

DB2子系统

主要有四个地址空间：

dsn1mstr

提供系统服务，完成系统相关的功能

dsn1spas

为用户提供
一个单独运行SQL程序
环境的程序

dsn1dbm1

提供数据库服务，使用
户可进行构建数据库的
操作

irlmproc

提供内部资源锁管理，
用于控制DB2的锁

DB2结构



数据结构



系统结构

DB2的日志

DB2会将所有数据的改变以及系统的一些重要事件都记录在系统日志中，在系统发生故障时用于恢复。db2系统有两种日志，active log(在硬盘上)，achive log(在硬盘上或在磁带上)。

SQL (Structured Query Language)

SQL是定义和操作关系数据库数据的标准语言。
它包括：

SQL DML – Data Manipulation Language

SQL DDL – Data definition Language

SQL DCL – Data Control Language

DB2I PRIMARY OPTION MENU

- | | |
|------------------------------|--|
| 1 SPUI | (Process SQL statements) |
| 2 DCLGEN | (Generate SQL and source language declarations) |
| 3 PROGRAM PREPARATION | (Prepare a DB2 application program to run) |
| 4 PRECOMPILE | (Invoke DB2 precompiler) |
| 5 BIND/REBIND/FREE | (BIND, REBIND, or FREE plans or packages) |
| 6 RUN | (RUN an SQL program) |
| 7 DB2 COMMANDS | (Issue DB2 commands) |
| 8 UTILITIES | (Invoke DB2 utilities) |
| D DB2I DEFAULTS | (Set global parameters) |
| X EXIT | (Leave DB2I) |

SPUFI

SSID: DSN1

Enter the input data set name:

(Can be sequential or partitioned)

1 DATA SET NAME ... ==> 'DEV.XXX.LIB(SELECT)'

2 VOLUME SERIAL ... ==>

(Enter if not cataloged)

3 DATA SET PASSWORD ==>

(Enter if password protected)

Enter the output data set name:

(Must be a sequential data set)

4 DATA SET NAME ... ==> DB2.OUTPUT

Specify processing options:

5 CHANGE DEFAULTS ==> NO

(Y/N - Display SPUFI defaults panel?)

6 EDIT INPUT ==> YES

(Y/N - Enter SQL statements?)

7 EXECUTE ==> YES

(Y/N - Execute SQL statements?)

8 AUTOCOMMIT ==> YES

(Y/N - Commit after successful run?)

9 BROWSE OUTPUT ... ==> YES

(Y/N - Browse output data set?)

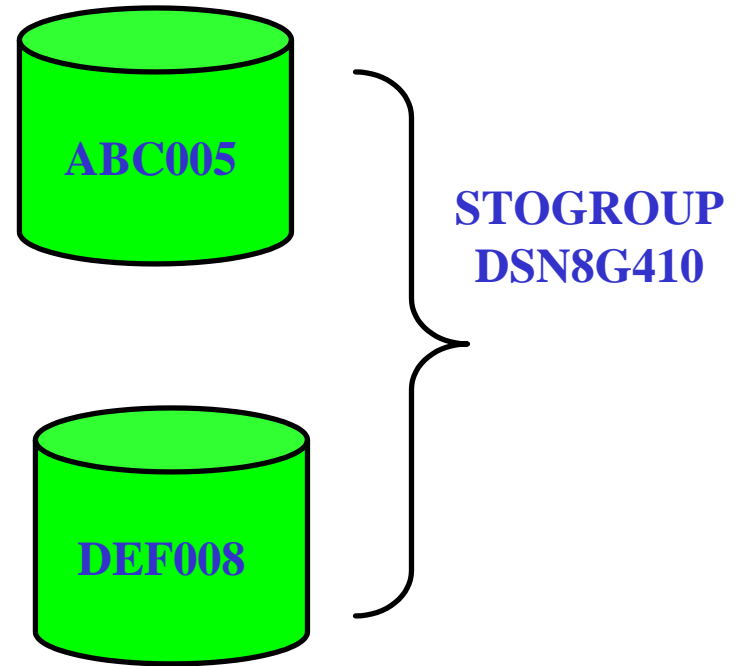
For remote SQL processing:

10 CONNECT LOCATION ==>

STOGROUP

在DB2中，是保存DB2目标的一系列磁盘卷，存放数据的实际空间；数据存储的结构实际上是一系列的VSAM数据集，它以Stogroup的形式关联DB2； Stogroup指明数据库存放数据所需要的空间，提供DB2使用的卷。一个Stogroup可以指向一个或多个卷（最多达133个DASD），一个Stogroup也可以有一个或多个Database。

```
CREATE STOGROUP DSN8G410  
VOLUMES (ABC005,DEF008)  
VCAT DSNCAT  
PASSWORD OSESAME;
```



DATABASE

在DB2中，Database是DB2结构的设置，表空间和索引空间的集合，一个Database可以关联一个应用的或一组相关应用的所有数据，一个Database可以定义一个或多个Tablespace。

CREATE DATABASE DSN8D41P

STOGROUP DSN8G410

BUFFERPOOL BP2;

TABLESPACE

Tablespace实际上就是VSAM数据集，是存放数据的物理空间，一个表空间可以含有一个或多个数据集；一个表空间可以用来存放一个或多个表的数据，即一个Tablespace可以定义一个或多个Tables；按类型有表空间和索引表空间；按数据的存放有简单表空间（Simple tablespace）、分段表空间（Segment tablespace）、分区表空间（Partition tablespace）。

CREATE TABLESPACE DSN8S41D

IN DSN8D41A

USING STOGROUP DSN8G410

PRIQTY 52

SECQTY 20

ERASE NO

LOCKSIZE PAGE

BUFFERPOOL BP1

CLOSE YES

DSETPASS OSESAME;

TABLE

Tables是用来访问DB2的逻辑数据结构，它是rows的集合，这些rows具有相同的结构，每一个row包含多个columns，row与column之间存在着某种关系，每一个row对应的是一个特定的实体（记录）。在建Table之前，必须先建它的Database和Tablespace。

CREATE TABLE DSN8410.DEPT

```
(DEPTNO          CHAR(3)  NOT NULL,  
DEPTNAME        VARCHAR(36) NOT NULL,  
MGRNO           CHAR(6)   ,  
ADMRDEPT        CHAR(3)  NOT NULL,  
LOCATION         CHAR(16)   ,  
PRIMARY KEY(DEPTNO)      )  
IN DSN8D41A.DSN8S41D;
```

DECLARE DSN8410.DEPT TABLE

```
(DEPTNO          CHAR(3)  NOT NULL,  
DEPTNAME        VARCHAR(36) NOT NULL,  
MGRNO           CHAR(6)   ,  
ADMRDEPT        CHAR(3)  NOT NULL,  
LOCATION          CHAR(16)  ) ;
```

INDEX

Index是用来提供一种存取TABLE中数据的指针，每一个Index都是基于Table的一列或多列数据，并且占有自己的索引空间（VSAM linear data set），索引与TABLE是分别定义的；一个表可以有多个索引，Cluster参数决定其索引值是顺序的，但在一个表中只能有一个Cluster Index；索引有Unique indexes 和 Non-unique indexes。


```
CREATE INDEX DSN8410.XEMP2  
ON DSN8410.EMP  
(EMPNO ASC)
```

```
CREATE UNIQUE INDEX DSN8410.XEMP1  
ON DSN8410.DEPT  
(DEPTNO ASC)
```

KEY

一个表可以定义一个或多个KEY，这些KEY，如果能唯一标识本表的行，则称为唯一键（Unique key）；否则，称为非唯一键（Nonunique key）。

一个表的一个或多个列（column），如果它（们）的值，可以唯一标识本表的行（row），那么，就可以将此一个或多个列（column）定义为主键（primary key）；主键必须是唯一键，一个表中，只能定义一个主键。

一个表的一个或多个列（column），可以定义为引用某个parent key的外部键：只有parent key的值存在时，才能加入到外部键(foreign key)。

VIEW

View是Table的视图，可以看作是一个虚拟Table，它不存储数据，只是提供访问Table数据的路径。View的产生可来自一个表或多个表的列的集合。

```
CREATE VIEW XYZ (EMPLOYEE, WHEN_HIRED)  
  
AS SELECT EMPNO, HIREDATE  
  
FROM DSN8410.EMP  
  
WHERE WORKDEPT IN ('A00', 'D11');
```

Data Types vs Host Variables

DB2

smallint
integer
decimal(n,m)
char(n)
varchar(n)
date
time
timestamp
float(53) double precision
float(21) real
graphic(n)
vargraphic(n)

COBOL

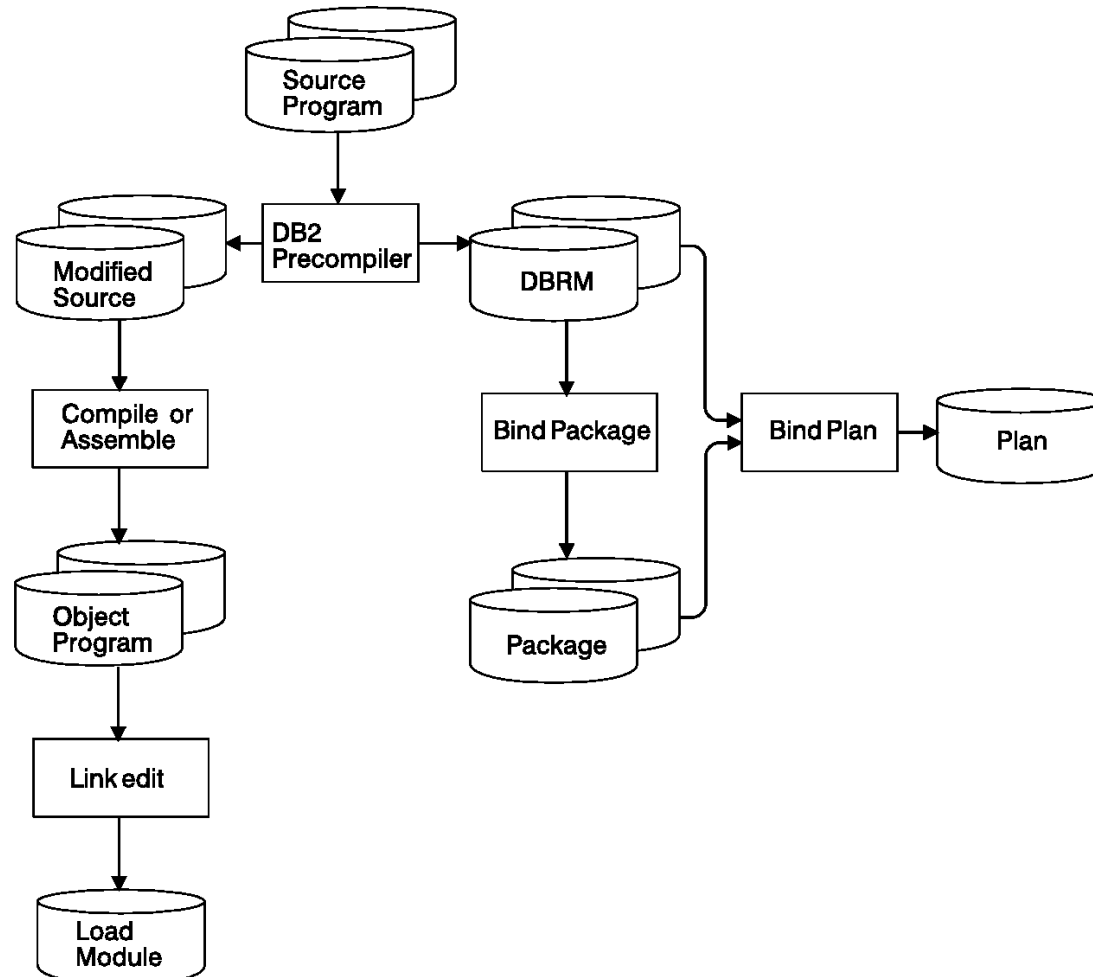
pic s9(4) comp
pic s(9) comp
pic s9(n-m)v9(m) comp-3
pic x(n)
pic s9(4) comp/ pic x(n)
pic x(10)
pic x(8)
pic x(26)
usage comp-2
usage comp-1
pic g(n) disp-1
pic g(n) disp-1/pic s9(4) comp

BIND

Bind是建立应用程序和它将访问相关数据之间的关联，在运行程序之前必须进行Bind。通常DB2允许两个Bind程序的方法：一个是Package，一个是直接的应用Plan。程序预编译处理时，DB2预编译器为每个应用程序产生修正的源码（SQL语句）和数据库要求的模块（DBRM）两个Objects，在应用程序能运行之前，其SQL语句必须进行编译和联接编辑，DBRM必须被Bind到Package或Plan中。

- 验证SQL语句，检查Table、View、Column等是否有效
- 验证用户是否有Bind权限、程序访问数据的权限
- 程序处理数据需要选择访问DB2的路径

Program Prepare



Guidelines for SELECT statement

- Specify only needed columns
- Limit the number of rows
- Specify the FOR UPDATE clause if applicable
- Specify the OPTIMIZED FOR n ROW clause
- Specify the FETCH FIRST n ROW ONLY clause if applicable
- Specify the FOR FETCH ONLY clause if applicable
- Avoid numeric data type conversion
- Avoid DISTINCT or ORDER BY if not required
- Write joins if equivalent to subquery
- Remember UNION ALL



HISUN 高阳科技



谢谢!