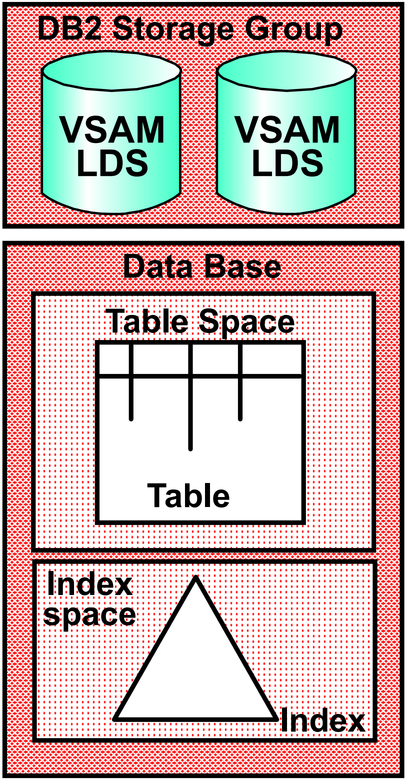
Objectives

- After completing this unit, you should be able to:
  - Describe the DB2 objects that make up a DB2 Database.
  - Select the most appropriate parameters for these objects so that they are implemented with the most appropriate attributes.
  - Create storage groups, databases, tablespaces, tables, views, indexes, synonyms and aliases.
  - Alter the attributes of DB2 Database objects as requirements change over time.
  - Describe how data is stored in a DB2 Database.

DB2 Objects



DB2 Storage Group

VSAM LDS

VSAM LDS

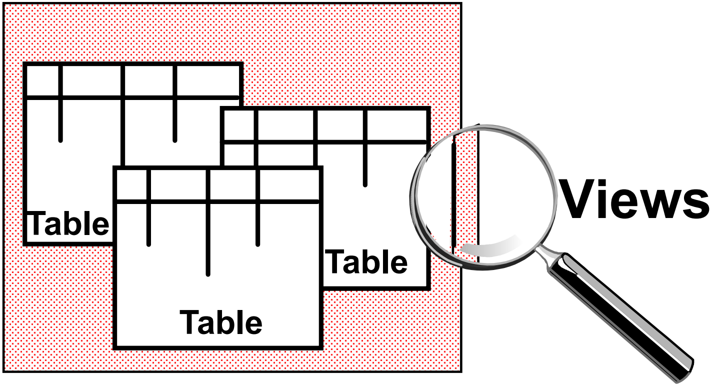
Data Base

Table Space

Table

Index space

Index



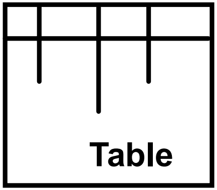
Table

Table

Table

Views

Synonyms and Aliases

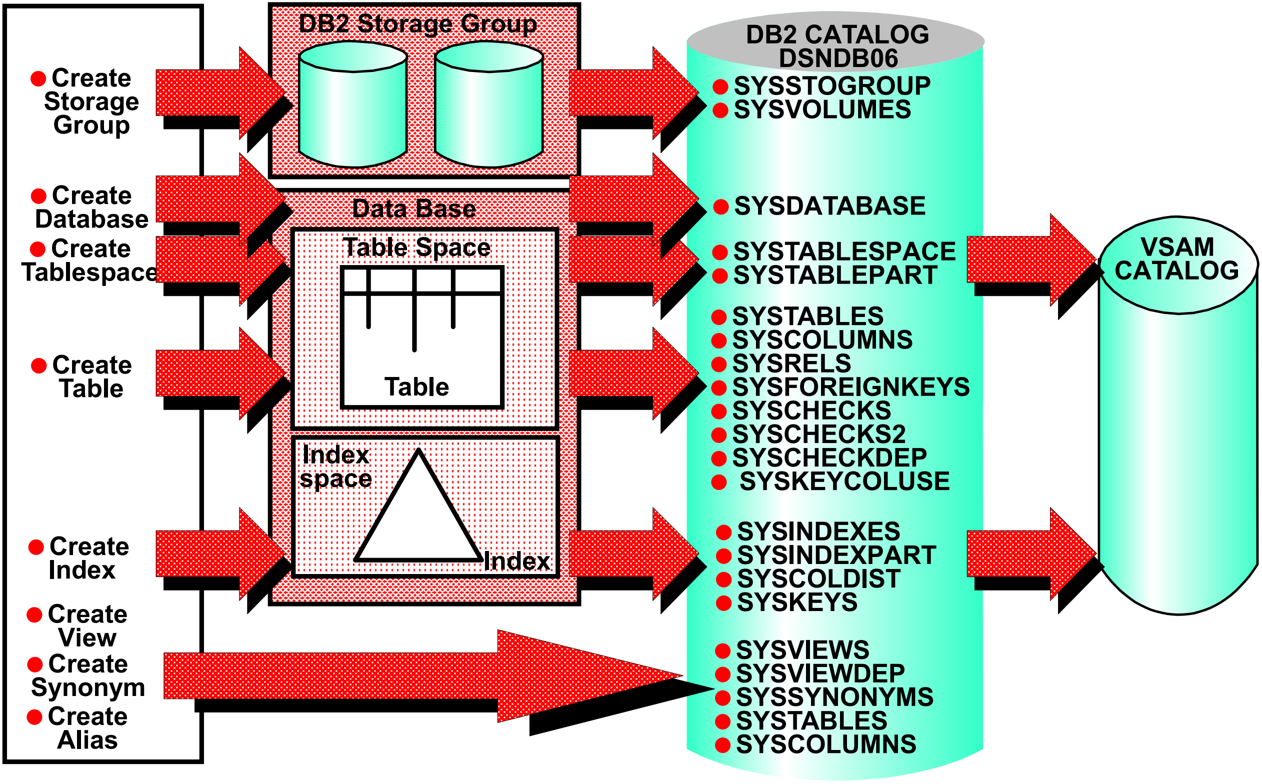


Table

V_LONG_TABLE_NAME

SELECT * FROM T1

Defining DB2 Objects



- Create Storage Group
- Create Database
- Create Tablespace
- Create Table
- Create Index
- Create View
- Create Synonym
- Create Alias

DB2 Storage Group

Data Base

Table Space

Table

Index space

Index

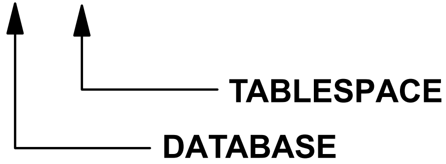
DB2 CATALOG
DSNDB06

- SYSSTOGROUP
- SYSVOLUMES
- SYSDATABASE
- SYSTABLESPACE
- SYSTABLEPART
- SYSTABLES
- SYSCOLUMNS
- SYSRELS
- SYSFOREIGNKEYS
- SYSCHECKS
- SYSCHECKS2
- SYSCHECKDEP
- SYSKEYCOLUSE
- SYSINDEXES
- SYSINDEXPART
- SYSCOLDIST
- SYSKEYS
- SYSVIEWS
- SYSVIEWDEP
- SYSSYNONYMS
- SYSTABLES
- SYSCOLUMNS

VSAM CATALOG

Example of Creating a DB2 Object

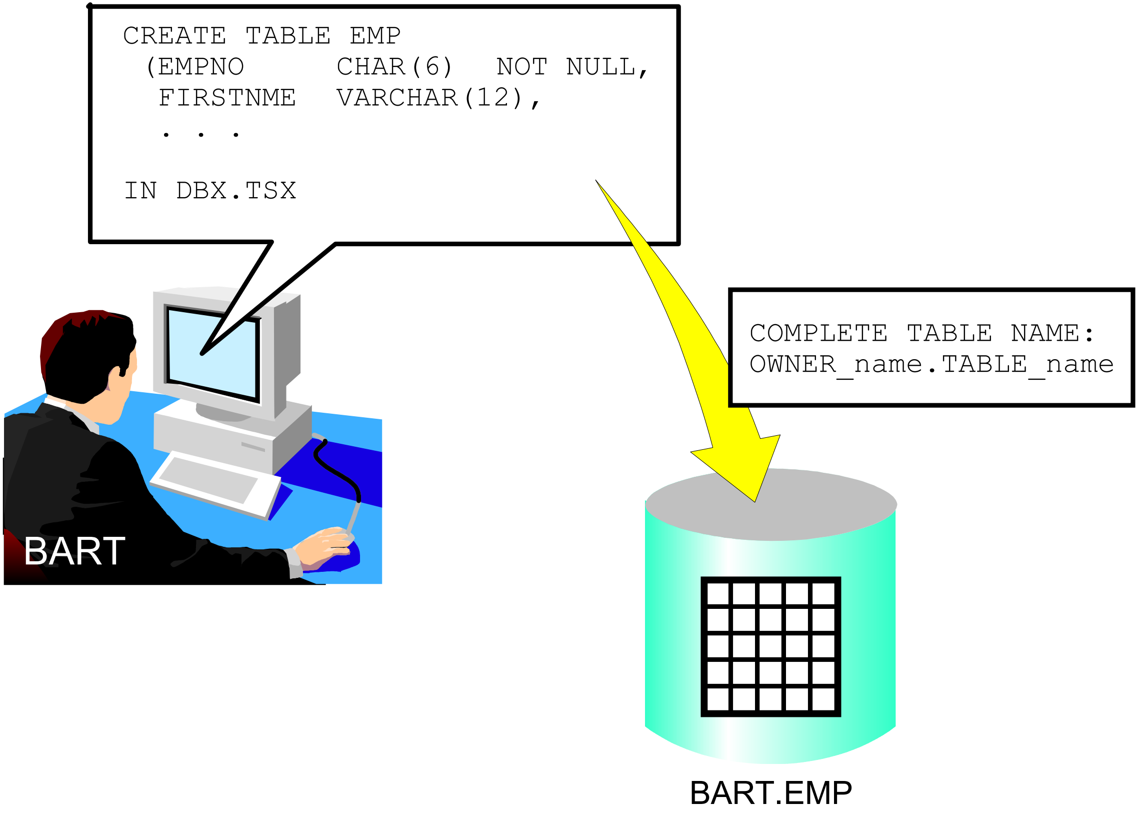
```
CREATE TABLE EMP
 (EMPNO        CHAR(6)        NOT NULL,
  FIRSTNME     VARCHAR(12)    NOT NULL,
  MIDINIT      CHAR(1)        NOT NULL WITH DEFAULT,
  LASTNAME     VARCHAR(15)    NOT NULL,
  WORKDEPT     CHAR(3),
  PHONENO      CHAR(4),
  HIREDATE     DATE,
  JOB          CHAR(8),
  EDLEVEL      SMALLINT,
  SEX          CHAR(1),
  BIRTHDATE DATE,
  SALARY       DECIMAL(9,2),
  BONUS        DECIMAL(9,2),
  COMM         DECIMAL(9,2))

IN DBX.TSX
```



TABLESPACE

DATABASE

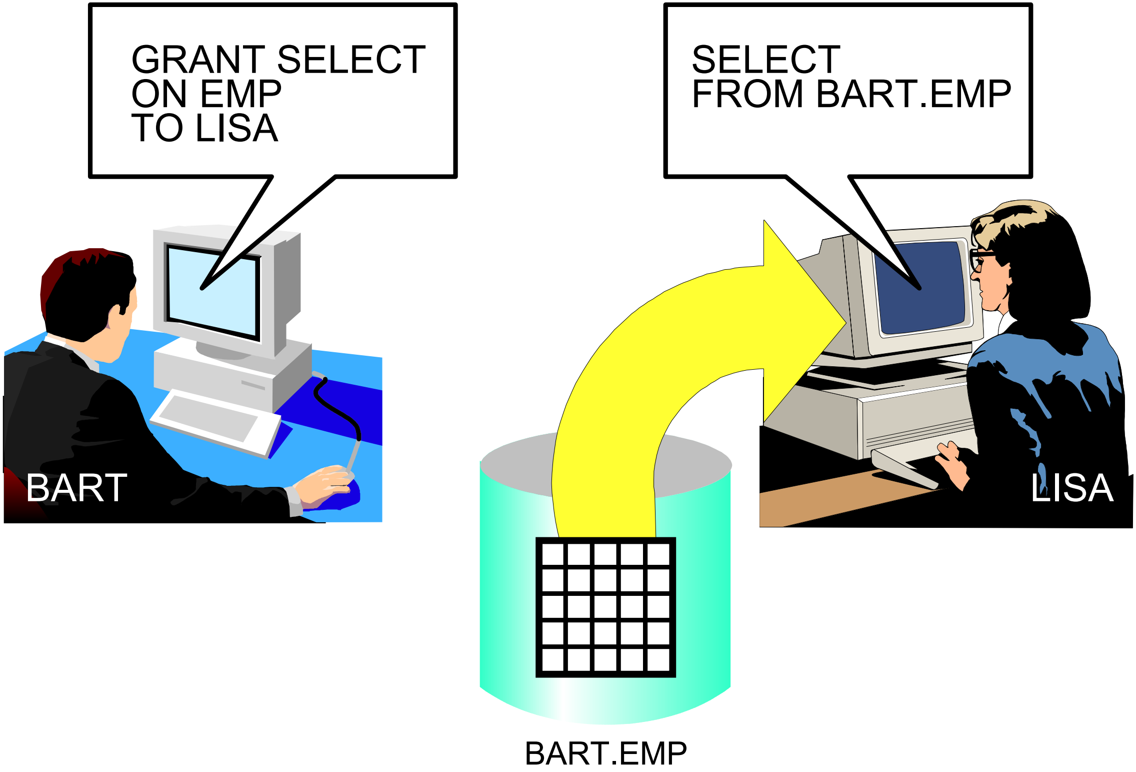
Owner Concept - Example



```
CREATE TABLE EMP
 (EMPNO       CHAR(6)   NOT NULL,
  FIRSTNME    VARCHAR(12),
  .  .  .

IN DBX.TSX
```

COMPLETE TABLE NAME:
OWNER_name.TABLE_name

Owner's Privileges



```
GRANT SELECT
ON EMP
TO LISA
```

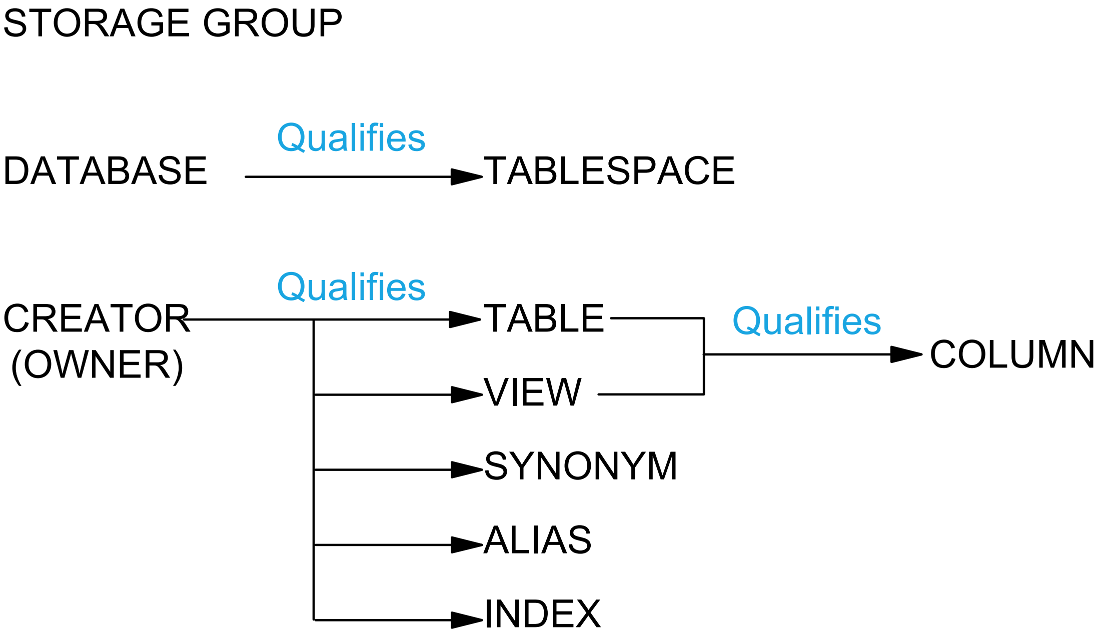
```
SELECT
FROM BART.EMP
```

BART

LISA

BART.EMP

DB2 Naming Rules



STORAGE GROUP

DATABASE

Qualifies

TABLESPACE

CREATOR
(OWNER)

Qualifies

TABLE

VIEW

SYNONYM

ALIAS

INDEX

Qualifies

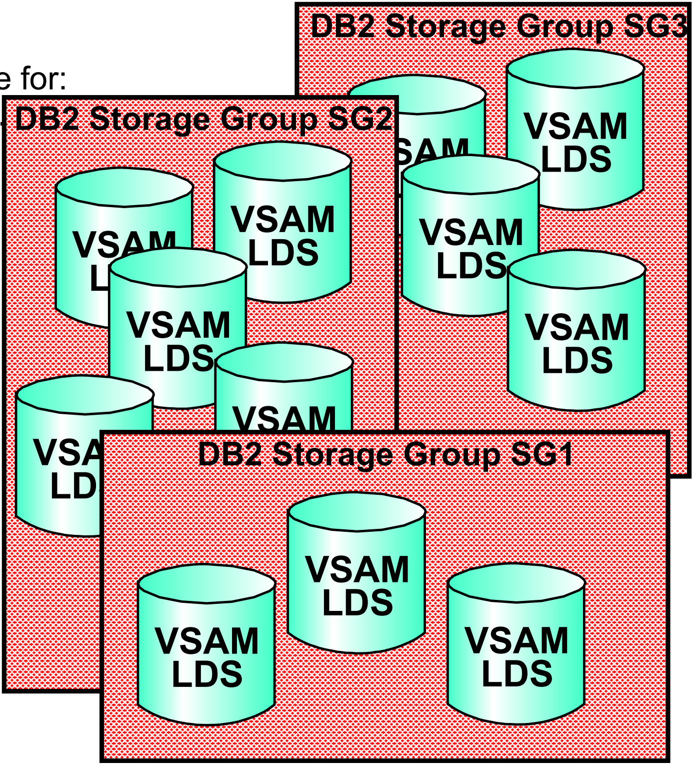
COLUMN

DB2 Storage Group

- DB2 Storage Group
  - Set of DASD volumes
  - From which DB2 can assign space for:
    - Table spaces and index spaces.
- DASD volumes are associated with:
  - An ICF catalog or VCAT.
    - Stores entries for all data sets

```
CREATE STOGROUP SG1
VOLUMES(VOL1, VOL3)
VCAT ALIASICF;

ALTER STOGROUP SG1
ADD VOLUMES(VOL4, VOL9)
REMOVE VOLUMES(VOL1);
```

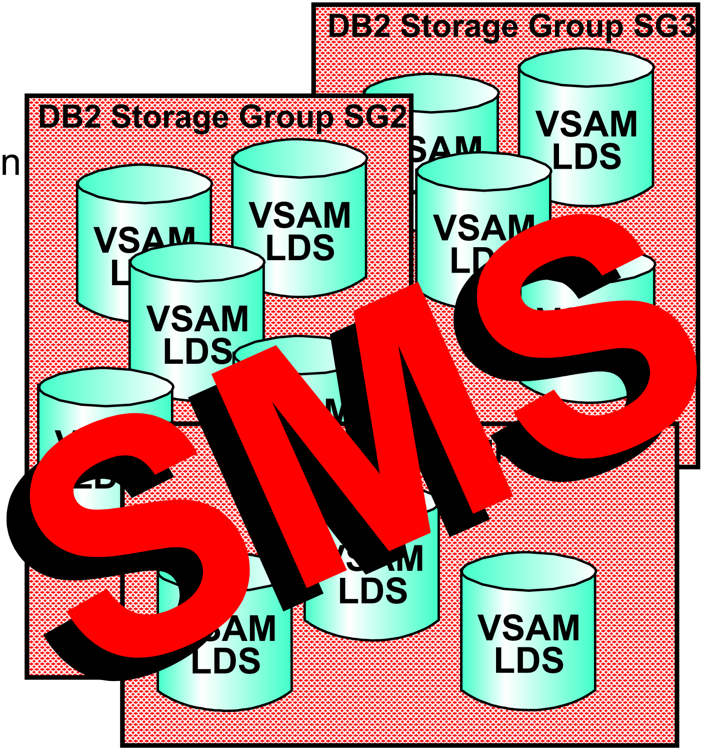
- SYSDEFLT
- Storage Group name
  - Maximum of eight alphanumeric characters
  - Unique within DB2 subsystem
  - No prefix



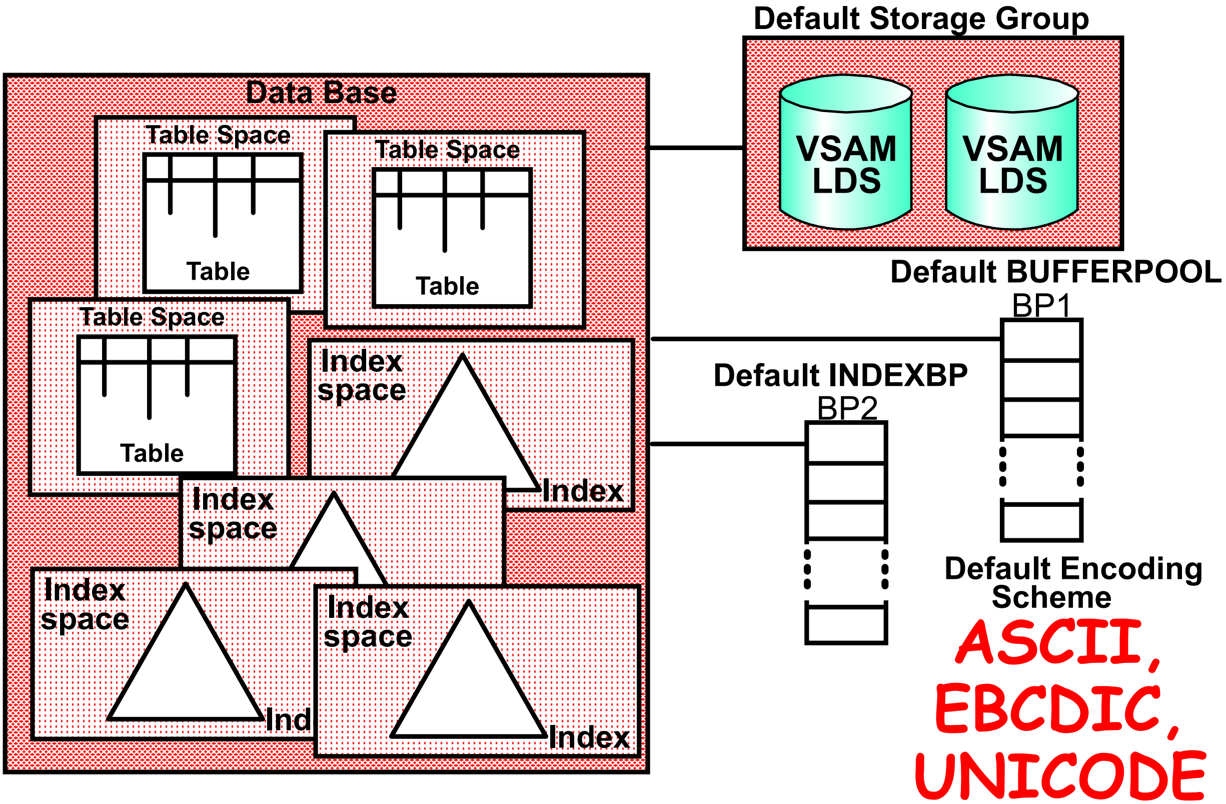
DB2 and SMS

- DFSMS can automatically manage all DB2 data sets
- DFSMS benefits include:
  - Simplified data set allocation
  - Improved allocation control
  - Improved performance management
  - Automated disk space management
  - Improved management of data availability
  - Simplified data movement

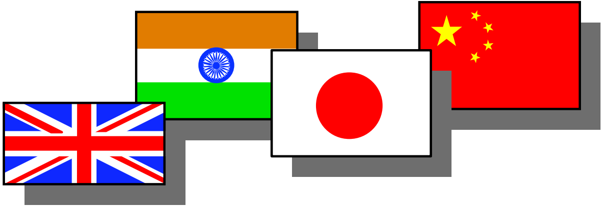
CREATE STOGROUP SG1
VOLUMES ('*')
VCAT ALIASICF



DB2 Data Base

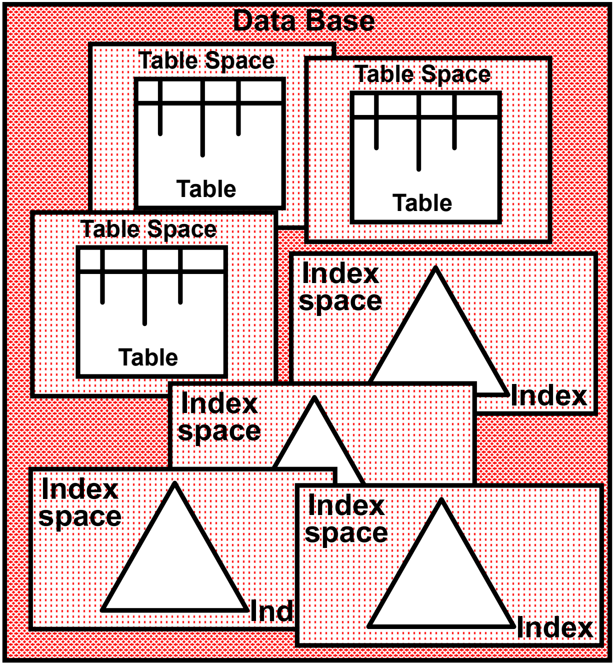


EBCDIC, ASCII and UNICODE



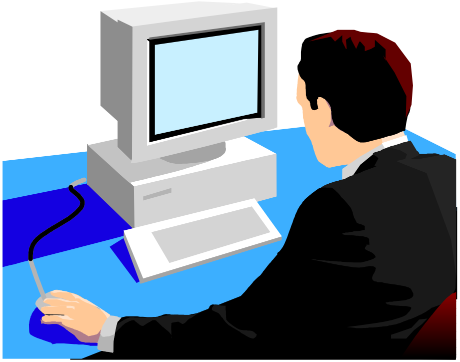
- EBCDIC
  - Mainframe data sets in most cases
- ASCII
  - PCs and workstations store data using ASCII
- UNICODE
  - All countries, languages, platforms, technical characters, punctuation marks, ...
  - Single, unique definition (code point)
    - For every character in the world
  - No character conversion necessary
  - Can specify at table level
    - Need not match other tables in the database

Administration at the Data Base Level



- GRANTs at the Data Base Level
- Commands at the Data Base Level

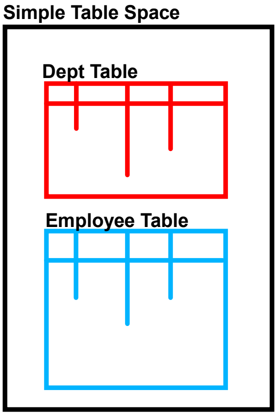
```
-DIS DB(...........) SPACENAM(..............)
-STOP DB(...........) SPACENAM(..............)
-START DB(...........) SPACENAM(..............)
```



DB2 Table Spaces

- What is a table space?
  - DB2 storage structure
  - Contains the data rows for one or more tables
  - Resides in a page set of one or more VSAM Linear Data Sets
    - Page size is 4 (usually), 8, 16 or 32K
  - Created in a data base using SQL
- Three types of table space
  - **Simple** table space
  - **Segmented** table space
  - **Partitioned** table space

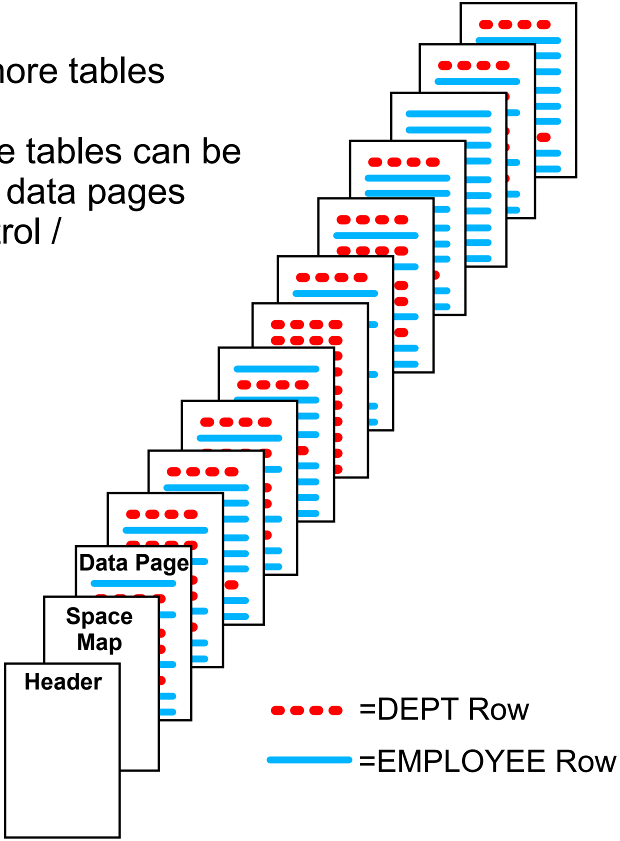
Simple Table Space



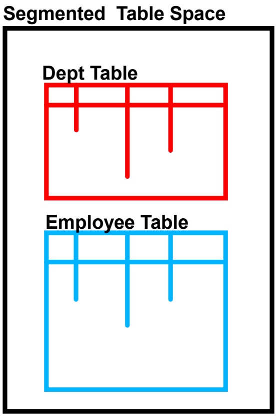
- Can hold one or more tables
  - No limit
- Rows from multiple tables can be interleaved on the data pages
  - Under your control / maintenance

Issues

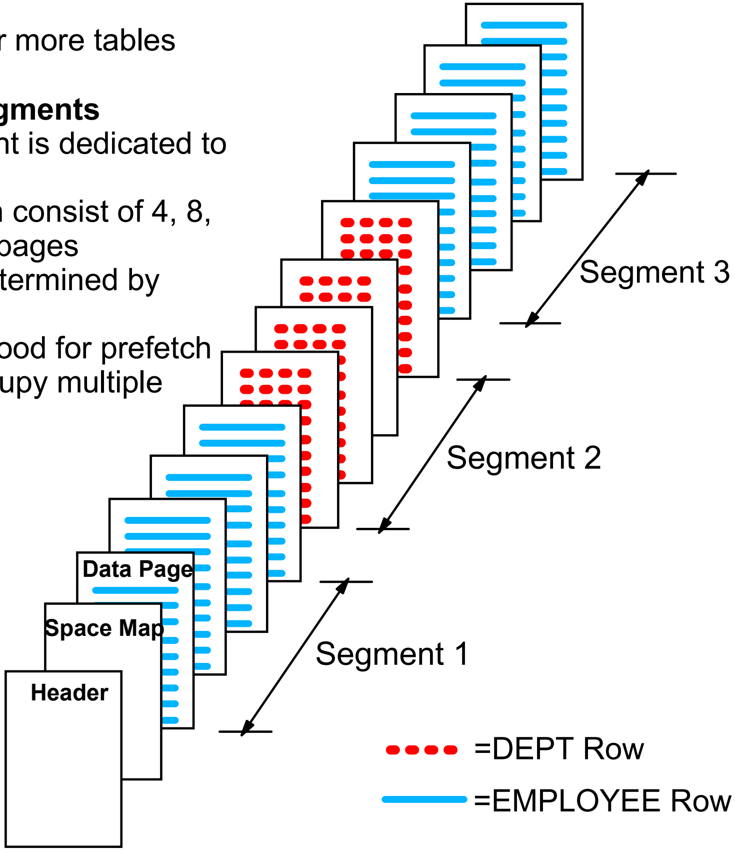
- TS scan
- Locking
- DROP TABLE
- Free space management



Segmented Table Space



- Can hold one or more tables
  - No limit
- Divided into **segments**
  - Each segment is dedicated to a table
  - Segment can consist of 4, 8, 16, 32 or 64 pages
  - SEGSIZE determined by table size
    - 32 or 64 good for prefetch
- A table can occupy multiple segments



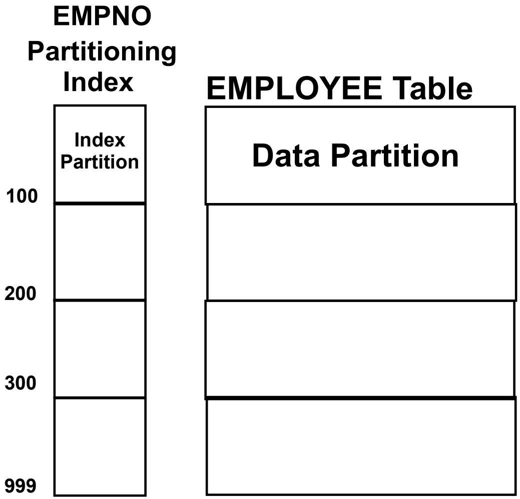
Issues Addressed

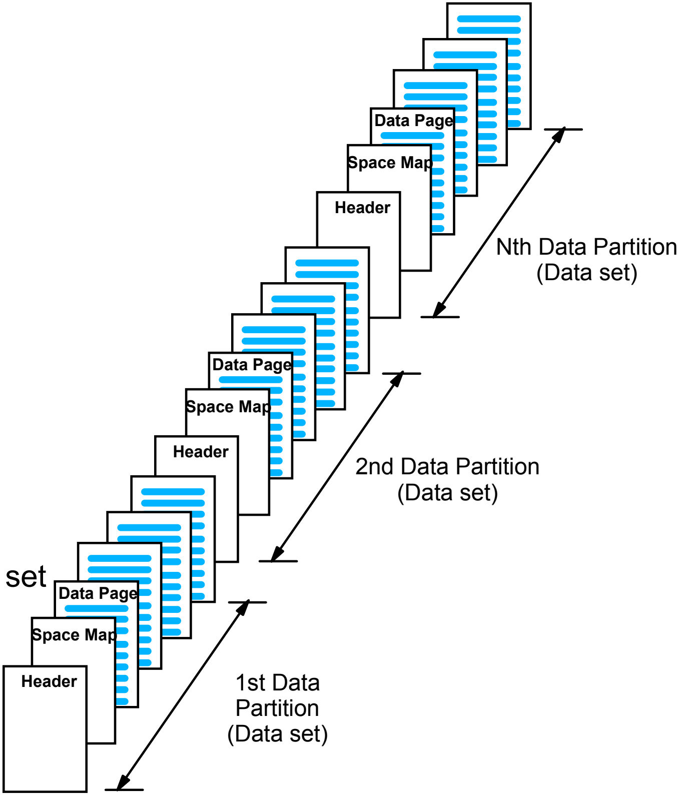
- TS Scan
- Locking
- DROP TABLE
- Free space management

Single and Multiple Table Tablespaces

- **Single**
  - Each table can have different attributes
    - Space allocations
    - Buffer pool assignments
  - Can schedule utilities at table level
  - Pending states (see later) limit availability of only one table
- **Multiple**
  - Need to run fewer utilities
  - Easier to keep related tables in step
    - Especially backup and recovery
  - Good for small reference tables
    - Avoids minimum allocation 2 tracks per table
    - Avoids header / space map for each table

Partitioned Table Space



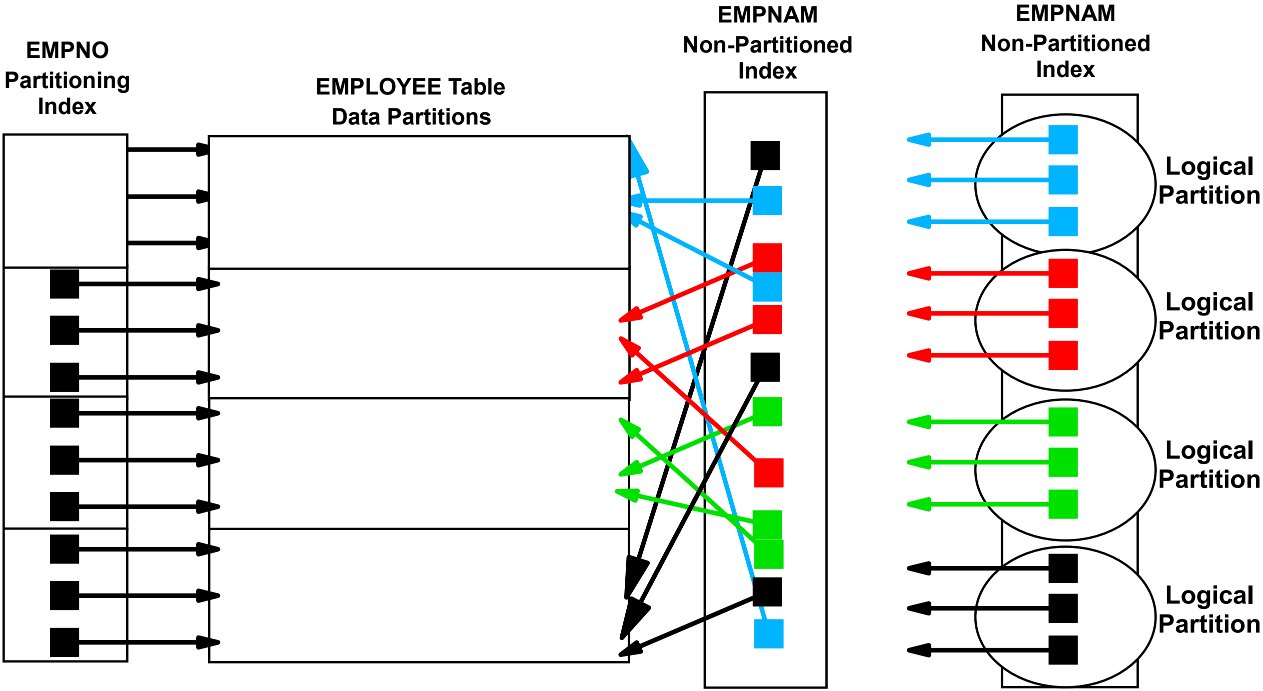


- One table only
- Divided into parts (partitions)
- Each partition is a separate VSAM data set
- Must have partitioning index
  - Designates which rows go into which partition
- Partitioning index is itself partitioned
  - Same number of VSAM data sets as used for table space partitions

Advantages of Partitioning

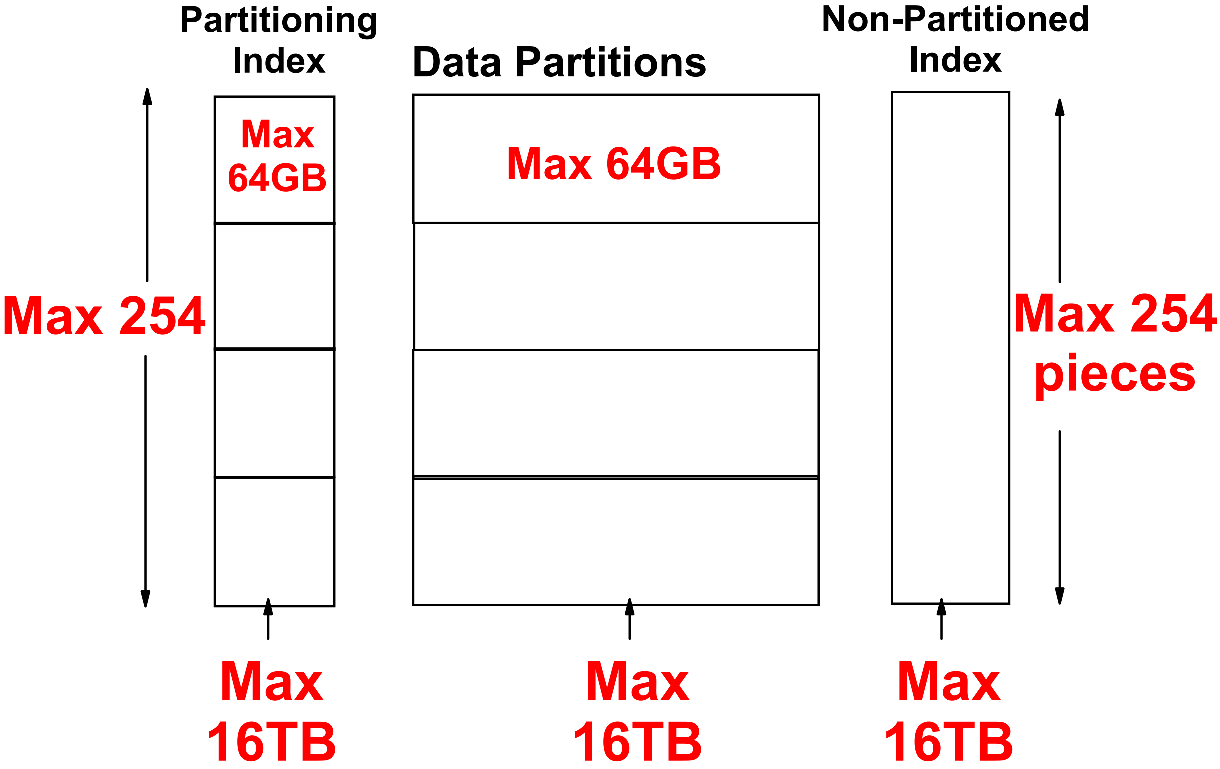
- Good for large volumes of data
        - At least 10,000,000 rows or 0.5GB data
    - Table space is in smaller, more manageable pieces
    - Can run utilities, commands or SQL independently at partition level
- Individual partitions can have different attributes
        - Space allocations
        - Buffer pool assignments
- Can spread partitions across multiple:
        - DASD devices
        - Channels
        - Control units
- Parallelism
- Partition scan

Non-Partitioned Index and Logical Partitions



- Same NPI illustrated in 2 different ways

Number of Partitions



CREATE TABLESPACE

● Simple Table Space

```
---------+---------+---------+---------+---------+--------
CREATE TABLESPACE GBTS1
IN GBDB1
USING STOGROUP GBCF830S
PRIQTY 14400
SECQTY   720
ERASE NO
LOCKSIZE ANY
BUFFERPOOL BP2
FREEPAGE 4
PCTFREE 20
MAXROWS 255
CCSID UNICODE;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

● Segmented Table Space

```
CREATE TABLESPACE GBTS2
IN GBDB1
USING STOGROUP GBCF830S
PRIQTY 14400
SECQTY   720
ERASE NO
SEGSIZE 4
LOCKSIZE ANY
BUFFERPOOL BP2
FREEPAGE 4
PCTFREE 20
MAXROWS 255
CCSID UNICODE;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

Setting Up a Partitioned Table Space

- Step 1: Create Partitioned Table Space

```
CREATE TABLESPACE GBTS3
IN GBDB1
USING STOGROUP GBCF830S
PRIQTY 14400
SECQTY   720
ERASE NO
PCTFREE 20
FREEPAGE 4
NUMPARTS 4
(PART 1 USING STOGROUP GBCF831S
        PRIQTY 7200
        SECQTY  720
        ERASE YES
        PCTFREE 15
        FREEPAGE 8
        COMPRESS YES,
PART 2 USING STOGROUP GBCF832S
        PRIQTY 7200
        SECQTY  720
        ERASE YES
        PCTFREE 25
        FREEPAGE 2
        COMPRESS YES)
BUFFERPOOL BP2;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

Setting Up a Partitioned Table Space

- Step 2: Create Table

```
---------+---------+---------+---------+---------+--------
CREATE TABLE EMPLOYEE
(EMPNO      INTEGER  NOT NULL,
 NAME       CHAR(10) NOT NULL,
 DEPT       INTEGER,
 JOB        CHAR(10) ,
 YEARS      INTEGER,
 SALARY     DECIMAL(10,2) ,
 INVESTP    INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST  INTEGER NOT NULL WITH DEFAULT)
 IN GBDB1.GBTS3;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

Setting Up a Partitioned Table Space

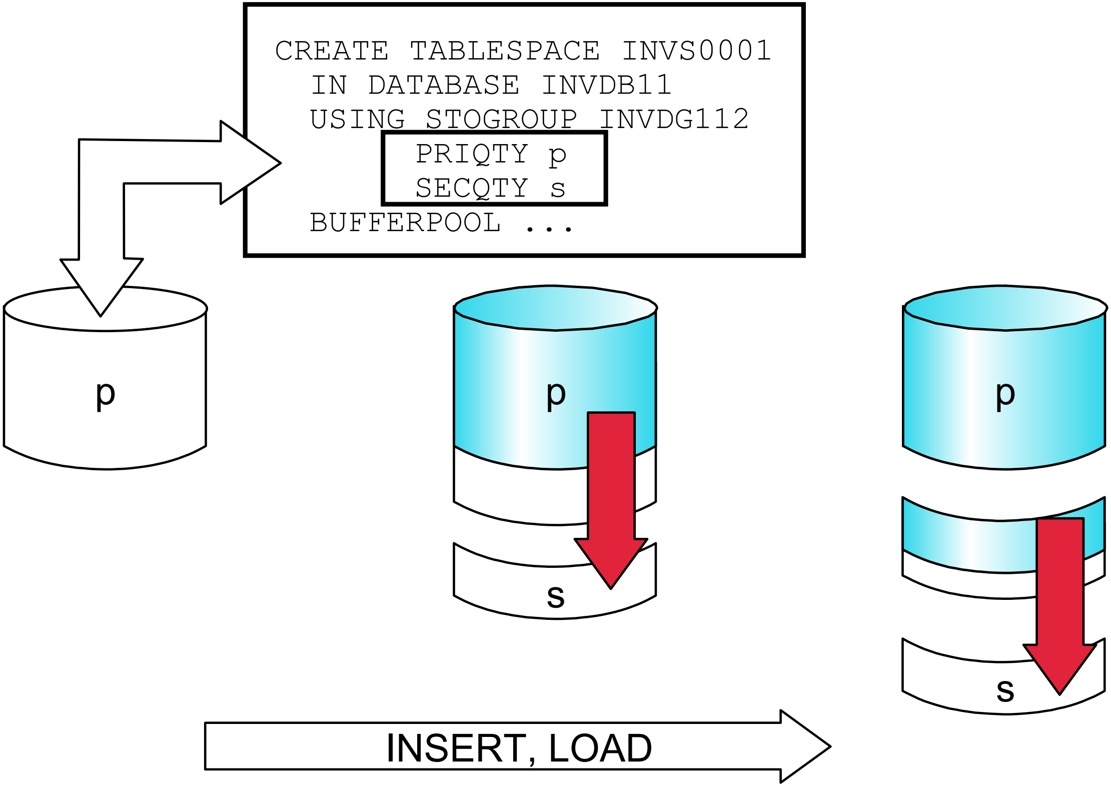
- Step 3: Create Partitioning Index

```
CREATE INDEX GBIXEMPNO
ON EMPLOYEE (EMPNO ASC)
USING STOGROUP GBCF830S
        PRIQTY 14400
        SECQTY  1440
        ERASE NO
        PCTFREE 10
        FREEPAGE 4
        CLUSTER
(PART 1 VALUES(100)
        USING STOGROUP GBCF831S
        PRIQTY 7200
        SECQTY  720
        PCTFREE 15
        FREEPAGE 8,
 PART 2 VALUES(200),
 PART 3 VALUES(300),
 PART 4 VALUES(999))
      BUFFERPOOL BP1
      CLOSE YES
      COPY YES;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

Partitioned Tablespaces - Considerations

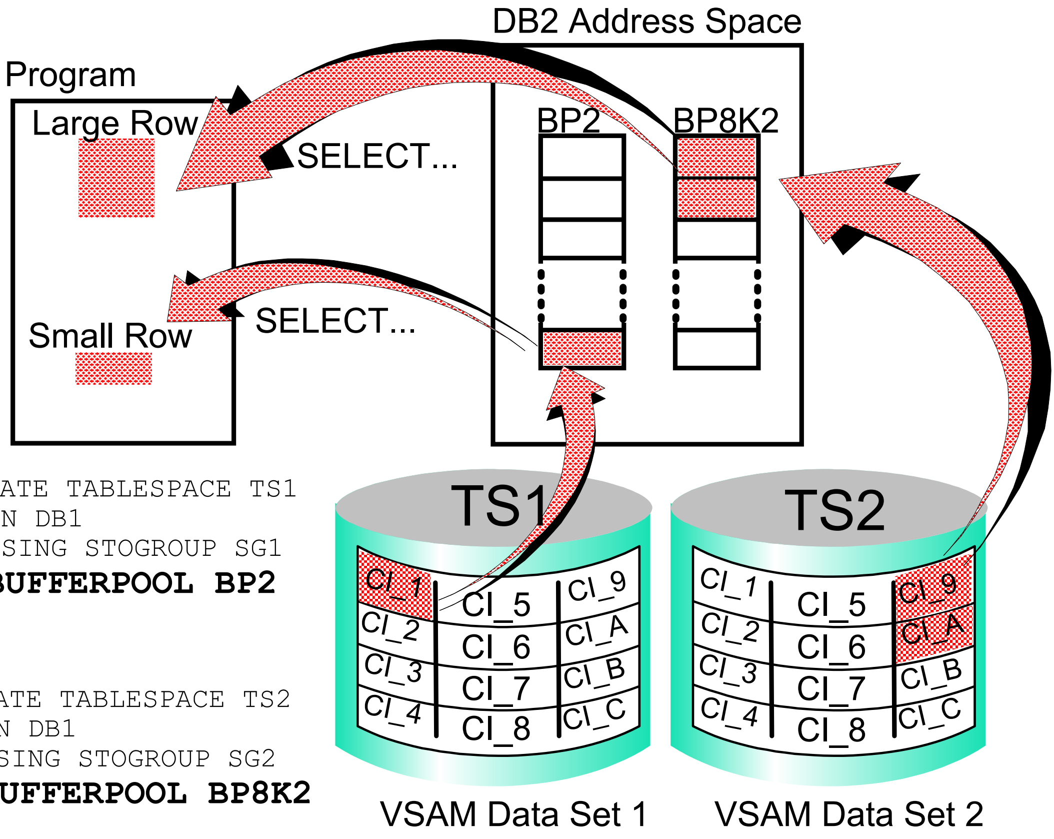
- Maximum 254 partitions
- Key limits can be ALTERed - must be followed by REORG of involved partitions
- Number of partitions cannot be altered - DROP/CREATE needed
- Only first 40 bytes of key (max length 254) used for partitioning
- I/O and CPU parallelism
- Partition independence

Space Allocation



```
CREATE TABLESPACE INVS0001
   IN DATABASE INVDB11
   USING STOGROUP INVDG112
            PRIQTY p
            SECQTY s
   BUFFERPOOL ...
```

DB2 Buffer Pools



```
CREATE TABLESPACE TS1
  IN DB1
  USING STOGROUP SG1
  BUFFERPOOL BP2
```

```
CREATE TABLESPACE TS2
  IN DB1
  USING STOGROUP SG2
  BUFFERPOOL BP8K2
```

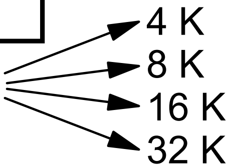
DB2 Buffer Pools

- DB2 supports up to 80 buffer pools:
  - BP0, BP1 BP49 (4 K pages)
  - BP8K0, BP8K1 BP8K9 (8 K pages)
  - BP16K0, BP16K1 BP16K9 (16 K pages)
- Buffer pool association can be dynamically changed (within the same page size):

```
ALTER TABLESPACE INVS0002
  BUFFERPOOL BP3
```

```
ALTER INDEX INVXINV0
  BUFFERPOOL BP8
```

- Implicitly defines page size



4 K
8 K
16 K
32 K

  - 8 K, 16 K and 32 K for table spaces, not indexes
  - BP0 is used by DB2 catalog and directory
  - Specify a default buffer pool for user data and another default for indexes (avoid BP0)

DB2 Naming Conventions for Data Sets

DB2 uses the following naming convention
for table spaces and index spaces

vcat.DSNDBx.db.ts.m0001.Annn

Where:

vcat High Level Qualifier (STOGROUP VCAT)

x C if a cluster, D if a data object

db Data base name (8 chars)

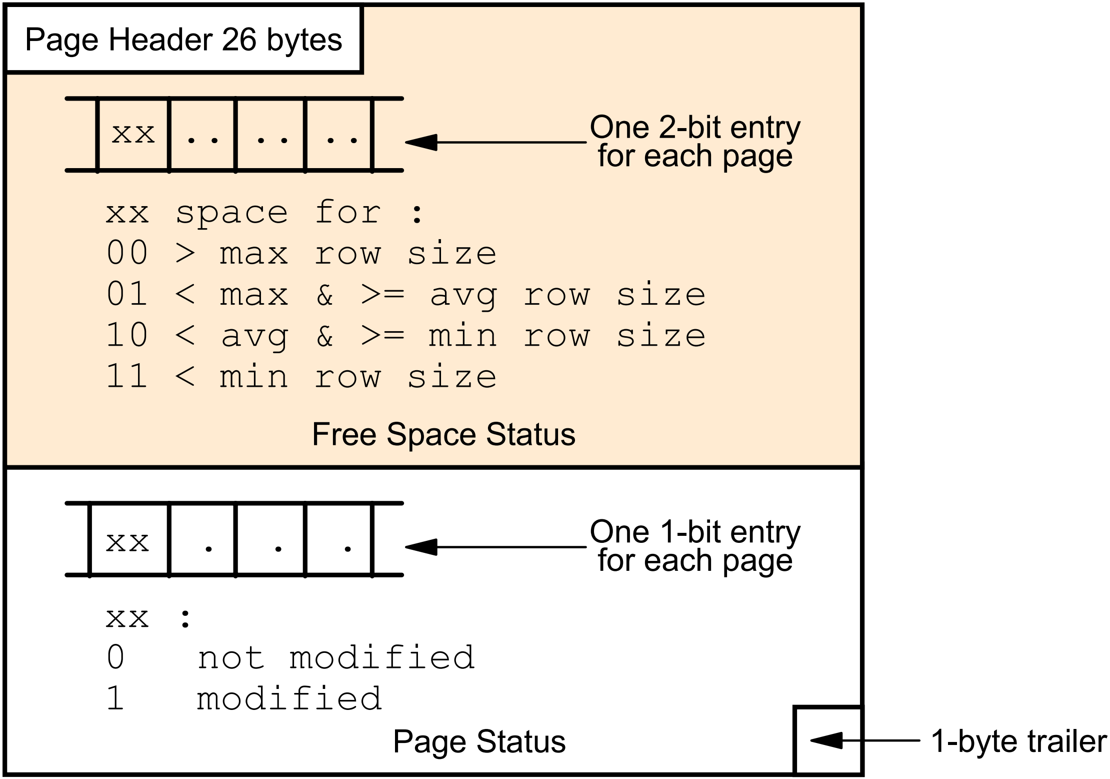
ts Table space name (8 chars) or first 8 chars of indexname, possibly scrambled to ensure uniqueness within DB

m Can be I or J

n Data set or partition number starting 001

Table Space - Space Map Page

- Simple and Partitioned TS



```
xx space for :
00 > max row size
01 < max & >= avg row size
10 < avg & >= min row size
11 < min row size
```

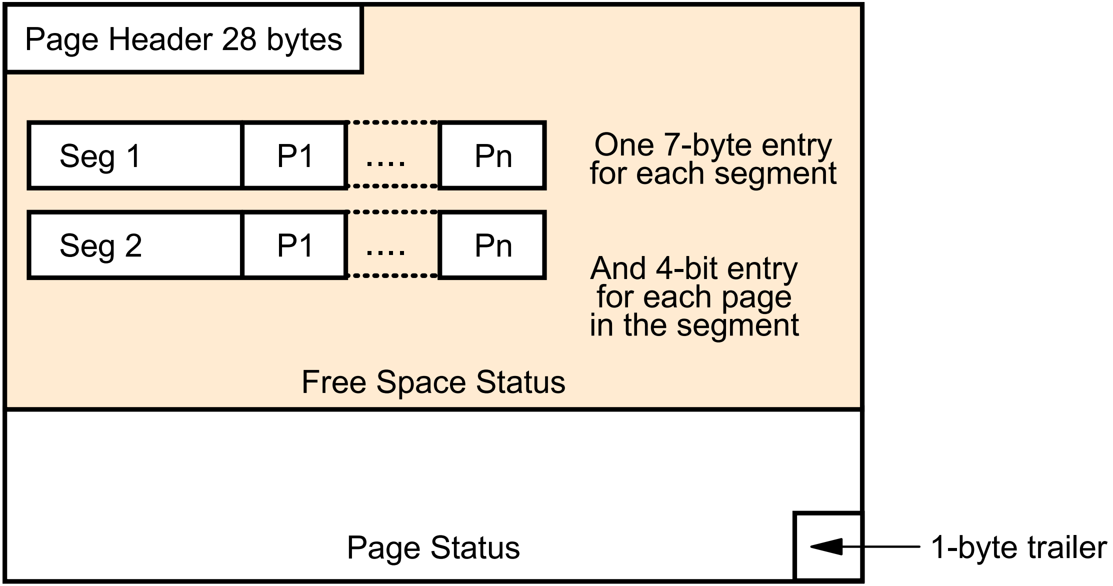
Free Space Status

```
xx :
0    not modified
1    modified
```

Page Status

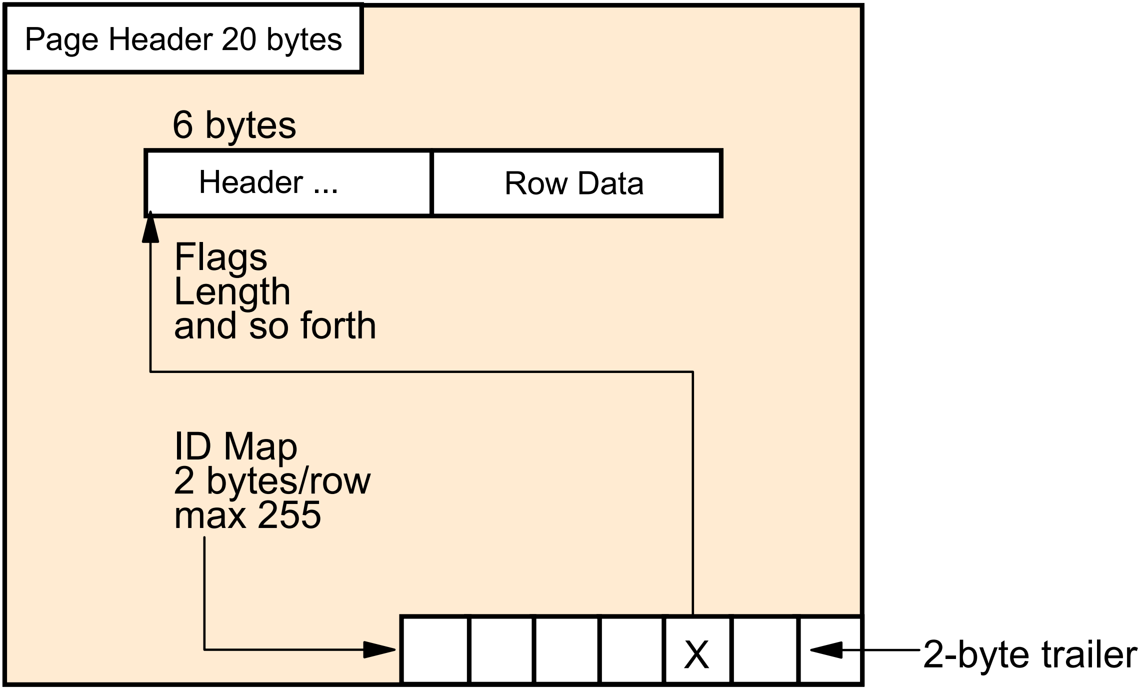
Table Space - Space Map Page

- Segmented TS



- The modified Page Status is the same as in Simple TS

Data Page Format



Data Page Management - Free Space

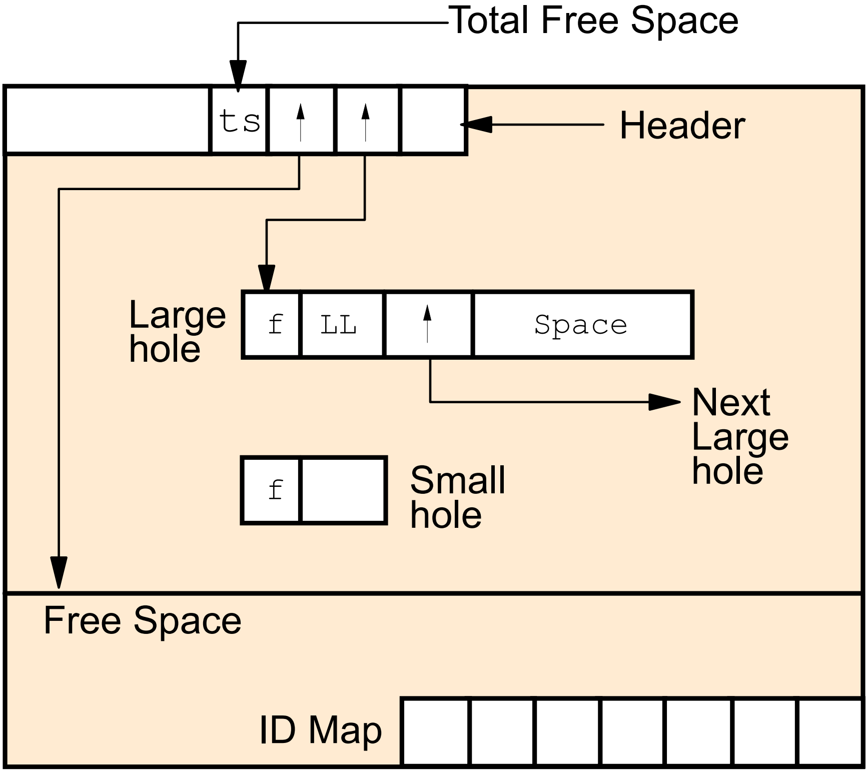


Table Space - Catalog Information

- SYSIBM.SYSTABLESPACE

```
 ---------+---------+---------+---------+---------+---------+
 SELECT   NAME,DBNAME,CREATOR,BPOOL,PARTITIONS,SEGSIZE,
          PGSIZE,STATUS,
          IMPLICIT,NTABLES,CLOSERULE,
          LOCKRULE,LOCKPART
 FROM SYSIBM.SYSTABLESPACE
 WHERE NAME LIKE 'GBTS%'
 ---------+---------+---------+---------+---------+---------+
 NAME      DBNAME    CREATOR   BPOOL     PARTITIONS  SEGSIZE
 ---------+---------+---------+---------+---------+---------+
 GBTS1     GBDB1     KIDDJA    BP2                0        0
 GBTS2     GBDB1     KIDDJA    BP2                0        4
 GBTS3     GBDB1     KIDDJA    BP2                4        0
---------+---------+---------+---------+---------+---------+------
 PGSIZE  STATUS  IMPLICIT  NTABLES  CLOSERULE  LOCKRULE  LOCKPART
---------+---------+---------+---------+---------+---------+------
      4  T       N               0  Y          A
      4  T       N               0  Y          A
      4  A       N               1  Y          A

DSNE610I NUMBER OF ROWS DISPLAYED IS 3
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+-
```

Table Space - Catalog Information

- SYSIBM.SYSTABLEPART

```
---------+---------+---------+---------+---------+---------+
SELECT   TSNAME,DBNAME,PARTITION,IXNAME,
         SUBSTR(LIMITKEY,1,4) AS LIMITKEY,PQTY,SQTY,
         STORNAME,VCATNAME,FREEPAGE,PCTFREE
FROM SYSIBM.SYSTABLEPART
WHERE TSNAME LIKE 'GBTS%'
---------+---------+---------+---------+---------+---------+
TSNAME    DBNAME    PARTITION  IXNAME              LIMITKEY
---------+---------+---------+---------+---------+---------+
GBTS1     GBDB1             0                      0
GBTS2     GBDB1             0                      0
GBTS3     GBDB1             1  GBIXEMPNO           100
GBTS3     GBDB1             2  GBIXEMPNO           200
GBTS3     GBDB1             3  GBIXEMPNO           300
GBTS3     GBDB1             4  GBIXEMPNO           999
--------+---------+---------+---------+---------+---------+
       PQTY     SQTY  STORNAME  VCATNAME  FREEPAGE  PCTFREE
--------+---------+---------+---------+---------+---------+
       3600      180  GBCF830S  GBCF83           4       20
       3600      180  GBCF830S  GBCF83           3       20
       1800      180  GBCF831S  GBCF83           8       15
       1800      180  GBCF832S  GBCF83           2       25
       3600      180  GBCF830S  GBCF83           4       20
       3600      180  GBCF830S  GBCF83           4       20
DSNE610I NUMBER OF ROWS DISPLAYED IS 6
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+--
```

Tables - Catalog Information

- SYSIBM.SYSTABLES

```
---------+---------+---------+---------+---------+---------+---------+
SELECT   SUBSTR(NAME,1,9) AS NAME,CREATOR,TYPE,TSNAME,DBNAME,COLCOUNT,
         RECLENGTH
FROM SYSIBM.SYSTABLES
WHERE NAME = 'EMPLOYEE'
AND CREATOR = 'KIDDJA'
---------+---------+---------+---------+---------+---------+---------+
NAME       CREATOR   TYPE  TSNAME    DBNAME    COLCOUNT  RECLENGTH
---------+---------+---------+---------+---------+---------+---------+
EMPLOYEE   KIDDJA    T     GBTS3     GBDB1            8         58
DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+
```

CREATE TABLE - Column Attributes

Datatype

Null Attribute

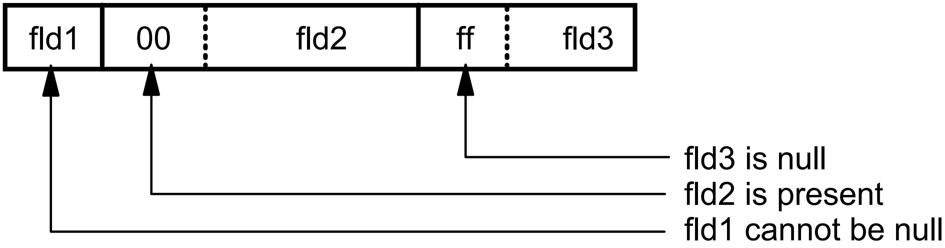
Default Attribute

```
CREATE TABLE EMP
 (EMPNO        CHAR(6)       NOT NULL,
  FIRSTNME     VARCHAR(12)   NOT NULL,
  MIDINIT      CHAR(1)       NOT NULL WITH DEFAULT,
  LASTNAME     VARCHAR(15)   NOT NULL,
  WORKDEPT     CHAR(3),
  PHONENO      CHAR(4)  WITH DEFAULT ‘NONE’,
  HIREDATE     DATE,
  JOB          CHAR(8),
  EDLEVEL      SMALLINT,
  SEX          CHAR(1),
  BIRTHDATE    DATE,
  SALARY       DECIMAL(9,2),
  BONUS        DECIMAL(9,2),
  COMM         DECIMAL(9,2))

IN DBX.TSX
```

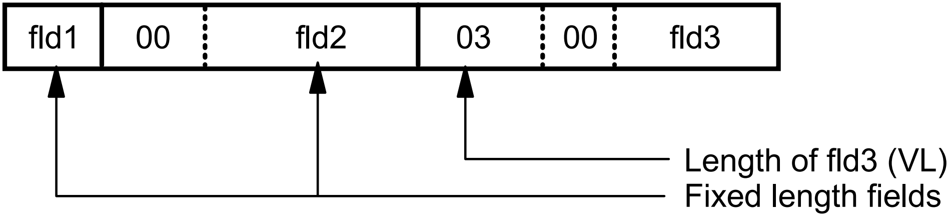
Row Format

- Nullable fields have a flag to indicate whether the field is present (00) or null (ff)

1 Fixed Length Row (no VL fields)



2 Variable Length Row (VL fields)



- ALTER TABLE ... ADD columnname will not update the existing records

➡ Updated when values are inserted for the new column or at REORG time

DEFAULT Attributes

- Default can be either:
  - A constant
  - USER (special register)
  - CURRENT SQLID (special register)
  - NULL
  - System defaults
- Examples:

```
        WITH DEFAULT ‘MY OWN VALUE’
        WITH DEFAULT USER
        WITH DEFAULT CURRENT SQLID
        WITH DEFAULT NULL
        WITH DEFAULT
```

ALTER TABLE

- ALTER TABLE changes certain characteristics of existing tables
- What can you change?
  - Add / remove:
    - Table check constraints or
    - Referential integrity definitions (see later)
  - Add an extra column
  - Change the length of an existing VARCHAR column

```
ALTER TABLE DEPT
  ALTER COLUMN DEPTNAME
  SET DATA TYPE VARCHAR(50);
```

- What can you not change?
  - Remove a column
  - Change a column name, data type, NULL or default attribute
  - Rearrange columns

Table Check Constraints

```
CREATE TABLE EMPLOYEE
(ID           INTEGER  NOT NULL,
 NAME         CHAR(10) NOT NULL,
 DEPT         INTEGER  CHECK (DEPT BETWEEN 1 AND 100),
 JOB          CHAR(10),
 YEARS        INTEGER,
 SALARY       DECIMAL(10,2),
 INVESTP      INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST    INTEGER NOT NULL WITH DEFAULT,
 CHECK   (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0)
                             OR (YEARS >= 5)),
 CHECK   (INVESTP <= MAXINVEST))
 IN DB1.TS1;
---------+---------+---------+---------+---------+---------+---
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---
```

Adding / Removing Check Constraints

```
---------+---------+---------+---------+---------+---------+-
ALTER TABLE EMPLOYEE
      ADD CONSTRAINT SALCHECK
      CHECK(SALARY > 0);
---------+---------+---------+---------+---------+---------+-
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+-
--
ALTER TABLE EMPLOYEE
      DROP CONSTRAINT SALCHECK;
---------+---------+---------+---------+---------+---------+-
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+-
ALTER TABLE EMPLOYEE
      ADD CONSTRAINT SALCHECK
      CHECK(SALARY > 10000);
---------+---------+---------+---------+---------         -+--
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLC
---------+---------+---------+------                      ---
--
ALTER TABLE EMPLOYEE
      DROP CHECK SALCHECK;
---------+---------+-----              ---------+--
DSNE616I STATEMENT EXECUTION              SQLCODI
---------+---------+---------+--             --+-
```

Table Check Constraint Considerations

- If table is empty, the check constraint is added
- If table is populated, the action depends on CURRENT RULES special register (STD|DB2)
  - For STD
    - If no violating rows, then constraint is added
    - If violating rows, then ALTER fails
  - For DB2
    - Check Pending (CHKP) is set

The -DISPLAY DATABASE Command

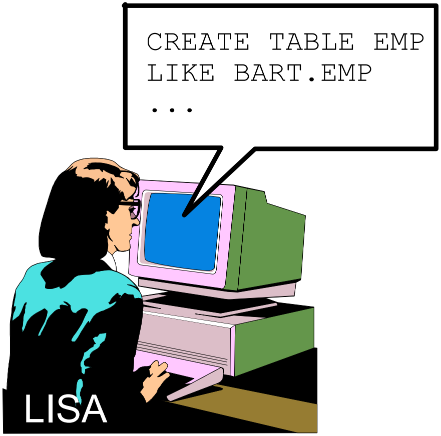
-DIS DB (INVDB11)

```
DSNT360I # *********************************************************
DSNT361I # * DISPLAY DATABASE SUMMARY
           * GLOBAL
DSNT360I # *********************************************************

DSNT362I # DATABASE = INVDB11 STATUS = RW
           DBD LENGTH = 4028

DSNT397I #
NAME     TYPE  PART  STATUS       PHYERRLO PHYERRHI  CATALOG  PIECE
......   ..... ..... ..........   .......  .......   .......  .....
INVSINV  TS          RW
INVSINL  TS          RW,CHKP
INVXINV0 IX          RW
INVXINLO IX          RW
*******  DISPLAY OF DATABASE INVDB11 ENDED *************************
DSN9022I # DSNTDDIS ‘DISPLAY DATABASE’ NORMAL COMPLETION
***
```

AUDIT / RESTRICT ON DROP

```
CREATE TABLE EMPLOYEE
(ID          INTEGER  NOT NULL,
 NAME        CHAR(10) NOT NULL,
 DEPT        INTEGER  CHECK (DEPT BETWEEN 1 AND 100),
 JOB         CHAR(10),
 YEARS       INTEGER,
 SALARY      DECIMAL(10,2),
 INVESTP     INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST   INTEGER NOT NULL WITH DEFAULT,
 CHECK    (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0)
      OR (YEARS >= 5)),
 CHECK    (INVESTP <= MAXINVEST))
 AUDIT ALL
 WITH RESTRICT ON DROP
 IN DB1.TS1;
---------+---------+---------+---------+---------+-------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+----
```

CREATE TABLE - LIKE Another Table



```
CREATE TABLE EMP
LIKE BART.EMP
...
```

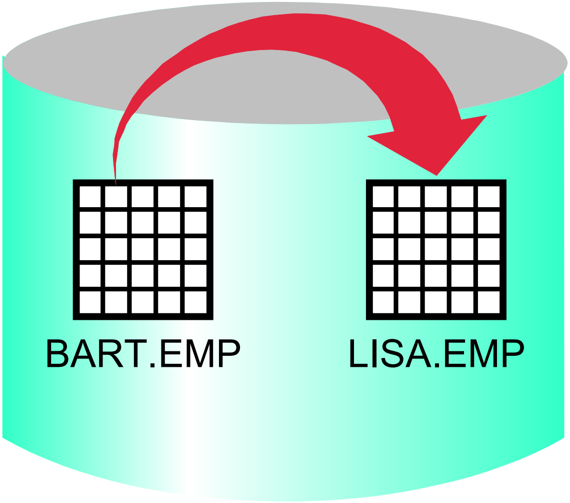
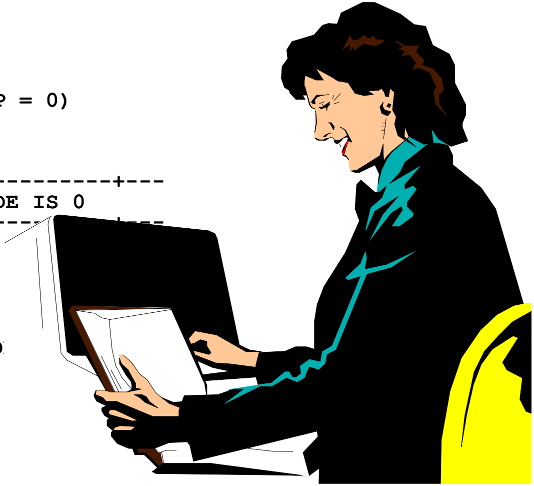


TABLE - Catalog Information

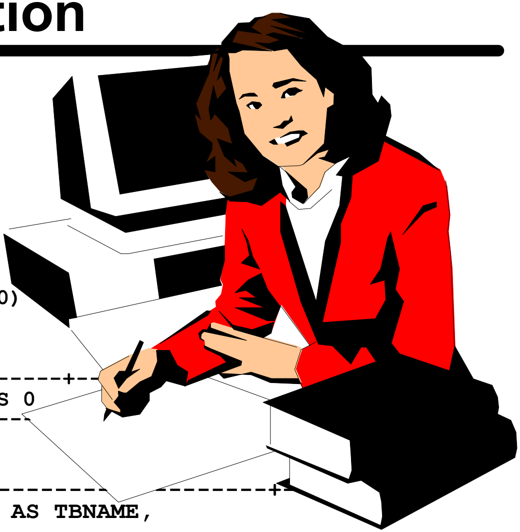
```
CREATE TABLE EMPLOYEE
(ID           INTEGER   NOT NULL,
 NAME         CHAR(10)  NOT NULL,
 DEPT         INTEGER   CHECK (DEPT BETWEEN 1 AND 100),
 JOB          CHAR(10),
 YEARS        INTEGER,
 SALARY       DECIMAL(10, 2),
 INVESTP      INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST    INTEGER NOT NULL WITH DEFAULT,
 CHECK    (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0)
                              OR (YEARS >= 5)),
 CHECK    (INVESTP <= MAXINVEST))
 IN DB1.TS1;
---------+---------+---------+---------+---------+---------+---
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---
```

SYSIBM.SYSTABLES



```
SELECT  SUBSTR(NAME,1,10) AS NAME,CREATOR,TYPE,D
        DBID,OBID,COLCOUNT,CARD,NPAGES,
        PCTPAGES,RECLENGTH,AVGROWLEN,
        CHECKS,ENCODING_SCHEME
FROM    SYSIBM.SYSTABLES
WHERE   NAME     = 'EMPLOYEE'
AND     CREATOR = 'KIDDJA'
---------+---------+---------+---------+---------+---------+---------+--
NAME         CREATOR    TYPE  DBNAME     TSNAME         DBID     OBID  COLCOUNT
---------+---------+---------+---------+---------+---------+---------+--
EMPLOYEE     KIDDJA     T     DB1        TS1             264        8         8

+---------+---------+---------+---------+---------+---------+---------+---------+-
  CARD         NPAGES   PCTPAGES  RECLENGTH    AVGROWLEN   CHECKS  ENCODING_SCHEME
+---------+---------+---------+---------+---------+---------+---------+---------+-
    -1            -1        -1         58           -1        4  E
```

COLUMN - Catalog Information



```
CREATE TABLE EMPLOYEE
(ID          INTEGER  NOT NULL,
 NAME        CHAR(10) NOT NULL,
 DEPT        INTEGER  CHECK (DEPT BETWEEN 1 AND 100),
 JOB         CHAR(10),
 YEARS       INTEGER,
 SALARY      DECIMAL(10, 2),
 INVESTP     INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST   INTEGER NOT NULL WITH DEFAULT,
 CHECK    (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0)
                               OR (YEARS >= 5)),
 CHECK    (INVESTP <= MAXINVEST))
 IN DB1.TS1;
---------+---------+---------+---------+---------+---------+--
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+------
```

SYSIBM.SYSCOLUMNS

```
---------+---------+---------+---------+---------+---------+---------+
SELECT SUBSTR(NAME,1,9) AS NAME,SUBSTR(TBNAME,1,9) AS TBNAME,
       COLNO,COLTYPE,LEN GTH,SCALE,NULLS,DEFAULT
FROM   SYSIBM.SYSCOLUMNS
WHERE  TBNAME    =  'EMPLOYEE'
AND    TBCREATOR = 'KIDDJA'
---------+---------+---------+---------+---------+---------+---------+--
NAME       TBNAME      COLNO  COLTYPE   LENGTH   SCALE  NULLS  DEFAULT
---------+---------+---------+---------+---------+---------+---------+--
ID         EMPLOYEE        1  INTEGER        4       0  N      N
NAME       EMPLOYEE        2  CHAR          10       0  N      N
DEPT       EMPLOYEE        3  INTEGER        4       0  Y      Y
JOB        EMPLOYEE        4  CHAR          10       0  Y      Y
YEARS      EMPLOYEE        5  INTEGER        4       0  Y      Y
SALARY     EMPLOYEE        6  DECIMAL       10       2  Y      Y
INVESTP    EMPLOYEE        7  INTEGER        4       0  N      Y
MAXINVEST  EMPLOYEE        8  INTEGER        4       0  N      Y

DSNE610I NUMBER OF ROWS DISPLAYED IS 8
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+--
```

Check Constraints - Catalog Information

```
CREATE TABLE EMPLOYEE
(ID           INTEGER   NOT NULL,
 NAME         CHAR(10)  NOT NULL,
 DEPT         INTEGER   CHECK (DEPT BETWEEN 1 AND 100),
 JOB          CHAR(10),
 YEARS        INTEGER,
 SALARY       DECIMAL(10, 2),
 INVESTP      INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST    INTEGER NOT NULL WITH DEFAULT,
 CHECK    (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0)
                             OR (YEARS >= 5)),
 CHECK    (INVESTP <= MAXINVEST))
 IN DB1.TS1;
---------+---------+---------+---------+---------+---------+---
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--
```

SYSIBM.SYSCHECKDEP

```
---------+---------+---------+---------+--------
SELECT TBOWNER,
       SUBSTR(TBNAME,1,9) AS TBNAME,
       SUBSTR(CHECKNAME,1,9) AS CHECKNAME,COLNAME
FROM   SYSIBM.SYSCHECKDEP
WHERE  TBOWNER    = 'KIDDJA'
AND    TBNAME = 'EMPLOYEE'
---------+---------+---------+---------+--------
TBOWNER    TBNAME      CHECKNAME   COLNAME
---------+---------+---------+---------+---------+-
KIDDJA     EMPLOYEE    DEPT        DEPT
KIDDJA     EMPLOYEE    INVESTP     INVESTP
KIDDJA     EMPLOYEE    INVESTP     MAXINVEST
KIDDJA     EMPLOYEE    JOB         JOB
KIDDJA     EMPLOYEE    NOINVEST    INVESTP
KIDDJA     EMPLOYEE    NOINVEST    YEARS
DSNE610I NUMBER OF ROWS DISPLAYED IS 6
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+--
```

Check Constraints - Catalog Information

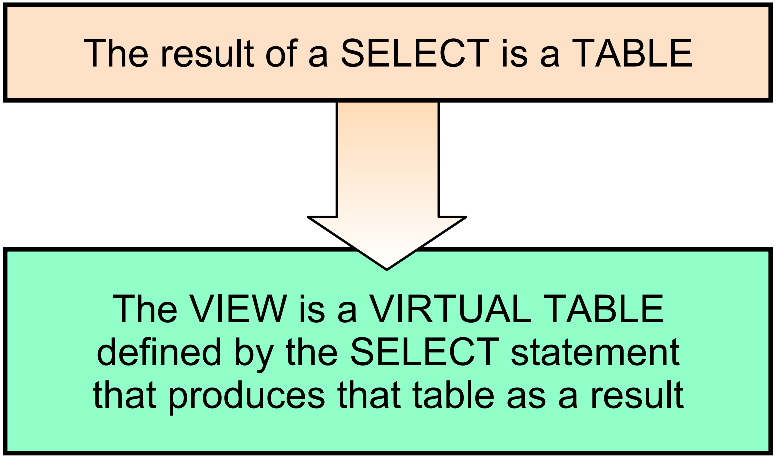
```
CREATE TABLE EMPLOYEE
(ID           INTEGER  NOT NULL,
 NAME         CHAR(10) NOT NULL,
 DEPT         INTEGER  CHECK (DEPT BETWEEN 1 AND 100),
 JOB          CHAR(10),
 YEARS        INTEGER,
 SALARY       DECIMAL(10, 2),
 INVESTP      INTEGER NOT NULL WITH DEFAULT,
 MAXINVEST    INTEGER NOT NULL WITH DEFAULT,
 CHECK    (JOB IN ('MANAGER','SALES','TECHNICAL')),
 CONSTRAINT  NOINVEST CHECK ((YEARS < 5 AND INVESTP = 0
                             OR (YEARS >= 5)),
 CHECK    (INVESTP <= MAXINVEST))
 IN DB1.TS1;
---------+---------+---------+---------+---------+---------+----
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+
```

SYSIBM.SYSCHECKS

```
---------+---------+---------+---------+---------+---------+
SELECT SUBSTR(TBNAME,1,9) AS TBNAME,OBID,TBOWNER,
       SUBSTR(CHECKNAME,1,9) AS CHECKNAME,
       CHECKCONDITION
FROM   SYSIBM.SYSCHECKS
WHERE  TBOWNER      = 'KIDDJA'
AND    TBNAME = 'EMPLOYEE'
---------+---------+---------+---------+---------+---------+---------+---------+--------
TBNAME       OBID  TBOWNER   CHECKNAME   CHECKCONDITION
---------+---------+---------+---------+---------+---------+---------+---------+--------
EMPLOYEE        9  KIDDJA    DEPT        DEPT BETWEEN 1 AND 100
EMPLOYEE       12  KIDDJA    INVESTP     INVESTP <= MAXINVEST
EMPLOYEE       10  KIDDJA    JOB         JOB IN ('MANAGER','SALES','TECHNICAL')
EMPLOYEE       11  KIDDJA    NOINVEST    (YEARS < 5 AND INVESTP = 0)      OR (YEARS >= 5)
DSNE610I NUMBER OF ROWS DISPLAYED IS 4
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---------+---------+--------
```

DB2 VIEWs

Relational Model:



The result of a SELECT is a TABLE

The VIEW is a VIRTUAL TABLE
defined by the SELECT statement
that produces that table as a result

Views can be derived from:

- Row or column subset of a table
- Calculated values (SUM, MAX, ...) from one or more rows
- A join of two or more tables

CREATE VIEW Example

EMPLOYEE

EMPNO	NAME	DEPT	ROOM	TELEPHONE	SALARY
110	LIEBHERR	A10	1018	4388	7500
220	ABELE	E10	1003	4407	4900
290	OBERHAUS	E11	1012	4112	5500
300	SCHMIDT	E11	1034	4234	4300
310	MUELLER	E11	1022	4419	4100

```
CREATE VIEW EMPE11
AS
SELECT EMPNO, NAME, ROOM, TELEPHONE
FROM   EMPLOYEE
WHERE  DEPT = 'E11';
```

EMPE11

EMPNO	NAME	ROOM	TELEPHONE
290	OBERHAUS	1012	4112
300	SCHMIDT	1034	4234
310	MUELLER	1022	4419

```
SELECT ROOM, TELEPHONE
FROM   EMPE11
WHERE  NAME  = 'SCHMIDT';
```

CREATE VIEW - Examples

```
CREATE VIEW MYTABLES
AS
  SELECT *
  FROM   SYSIBM.SYSTABLES
  WHERE  CREATOR = USER
```

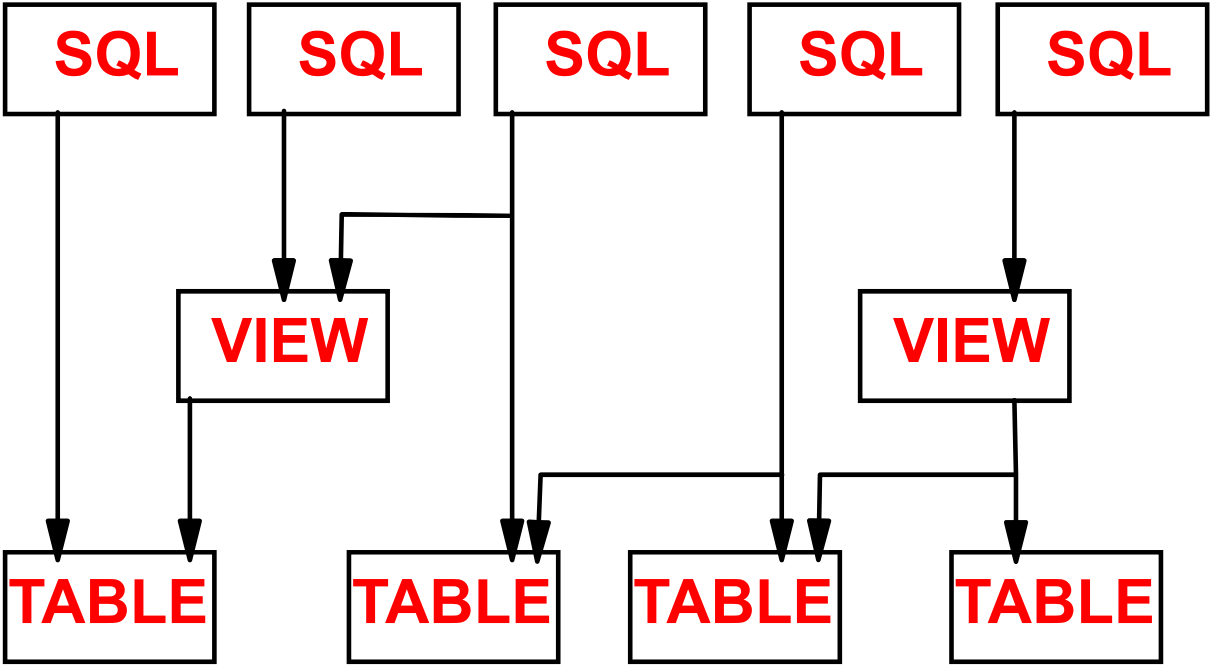
```
CREATE VIEW VSALSTAT
           (AVGSAL,
            SUMSAL,
            MINBIRTH)
AS
  SELECT AVG(SALARY),
         SUM(SALARY),
         MIN(BIRTHDATE)
  FROM   EMP
  WHERE  BIRTHDATE < ‘1980-01-01’
```

OR

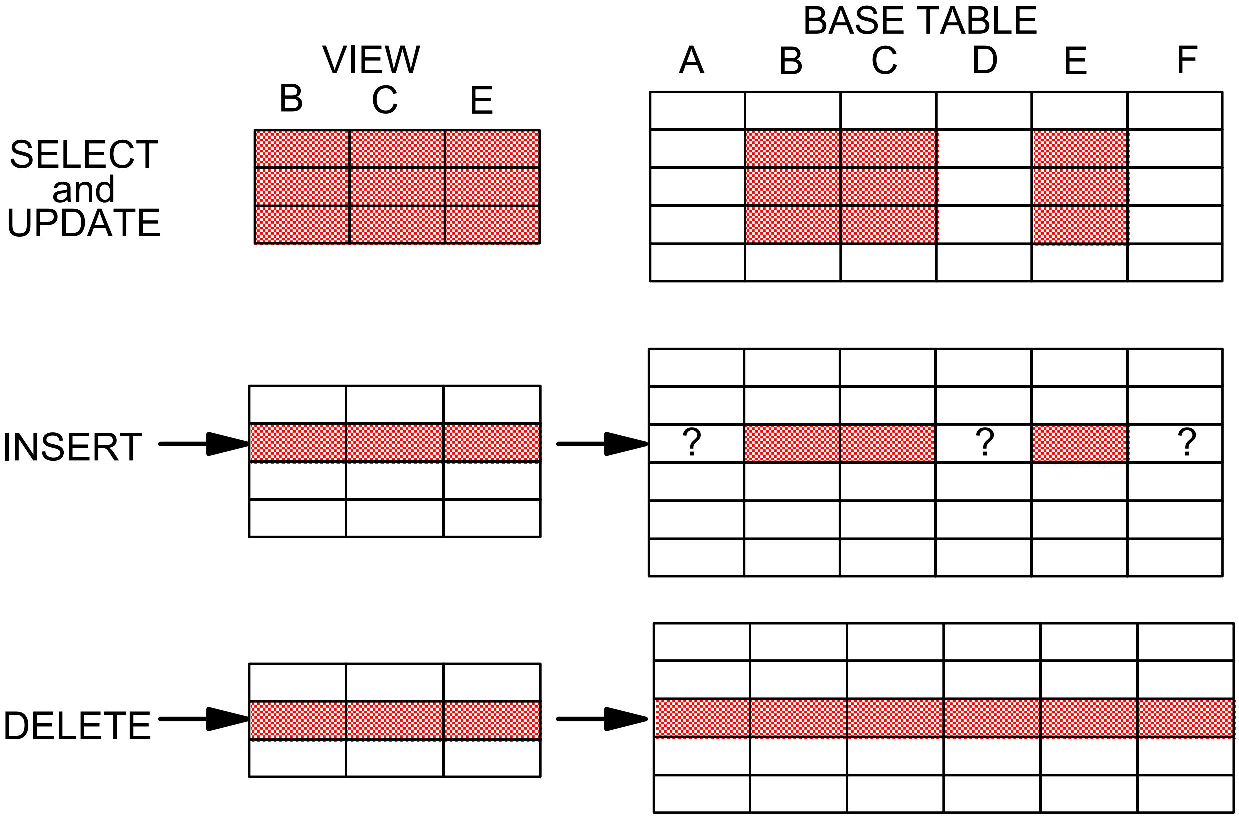
```
CREATE VIEW VSALSTAT
AS
  SELECT AVG(SALARY) AS AVGSAL
         SUM(SALARY)AS SUMSAL
         MIN(BIRTHDATE)AS MINBIRTH
  FROM   EMP
  WHERE  BIRTHDATE < ‘1980-01-01’
```

```
CREATE VIEW VFUTPROJ
AS
  SELECT MGRNO,
         DEPTNAME,
         PROJNO,
         RESPEMP
  FROM   PROJ,DEPT
  WHERE  PROJ.DEPTNO = DEPT.DEPTNO
  AND    PRENDATE + 1 YEAR < ‘1999-01-01’
```

Views



CREATE VIEW



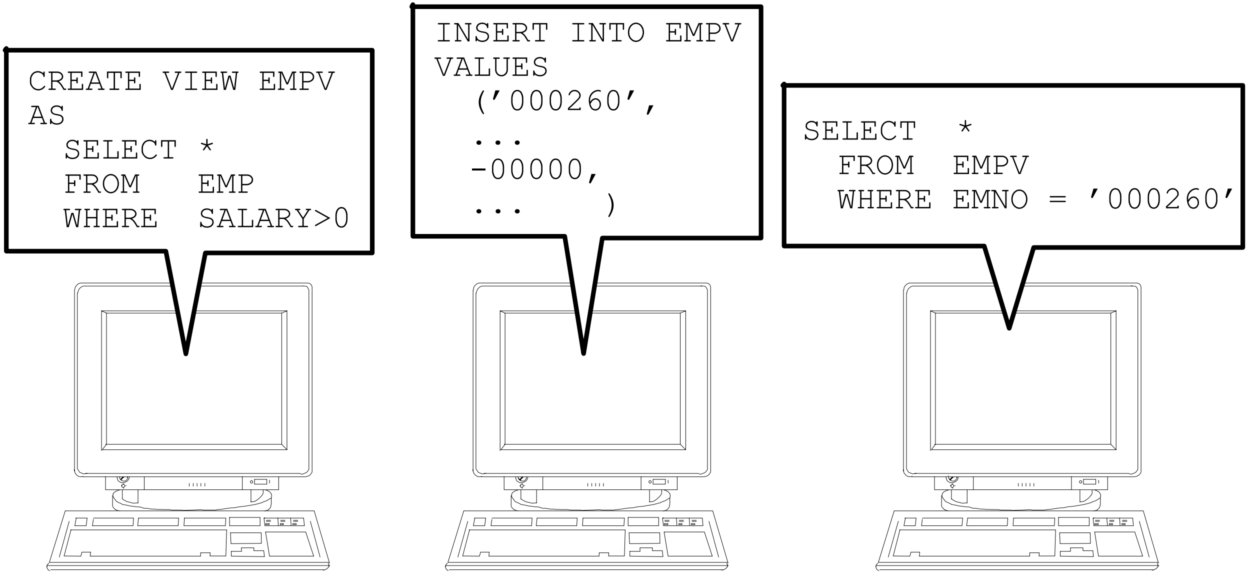
UNIONs in Views

- **Create the view JANUARY2002 that contains:**
  - All account details
  - Across all credit card types
  - For the month January 2002.
  - The columns are to be ACCOUNT, DATE and AMOUNT.

```
CREATE VIEW JANUARY2002 (ACCOUNT, DATE, AMOUNT) AS
SELECT ACCOUNT, DATE, AMOUNT
   FROM PLATINUM
   WHERE DATE BETWEEN '01/01/2002' AND '01/31/2002'
   UNION ALL
   SELECT ACCOUNT, DATE, AMOUNT
   FROM GOLD
   WHERE DATE BETWEEN '01/01/2002' AND '01/31/2002'
   UNION ALL
   SELECT ACCOUNT, DATE, AMOUNT
   FROM BLUE
   WHERE DATE BETWEEN '01/01/2002' AND '01/31/2002';
```

```
DECLARE.....SELECT AVG(AMOUNT), COUNT(*) FROM JANUARY2002;
OPEN.......
FETCH
```

The Disappearing Row



```
CREATE VIEW EMPV
AS
  SELECT *
  FROM   EMP
  WHERE  SALARY>0
```

```
INSERT INTO EMPV
VALUES
   (’000260’,
   ...
   -00000,
   ...      )
```

```
SELECT  *
  FROM  EMPV
  WHERE EMNO = ’000260’
```

CHECK OPTION

EMPLOYEE

EMPNO	NAME	DEPT	ROOM	TELEPHONE	SALARY
110	LIEBHERR	A10	1018	4388	7500
220	ABELE	E10	1003	4407	4900
290	OBERHAUS	E11	1012	4112	5500
300	SCHMIDT	E11	1034	4234	4300
310	MUELLER	E11	1022	4419	4100

1.

```
CREATE VIEW V1
AS
SELECT EMPNO, NAME, SALARY
FROM   EMPLOYEE
WHERE  SALARY < 5000
WITH CHECK OPTION;
```

V1

EMPNO	NAME	SALARY
220	ABELE	4900
300	SCHMIDT	4300
310	MUELLER	4100

2.

```
UPDATE V1
SET SALARY = SALARY + 200;
```

3.

```
CREATE VIEW V2
AS
SELECT EMPNO, NAME, SALARY
FROM   V1
WHERE  SALARY > 4200;
```

V2

EMPNO	NAME	SALARY
220	ABELE	4900
300	SCHMIDT	4300

4.

```
UPDATE V2
SET SALARY = SALARY + 200;
```

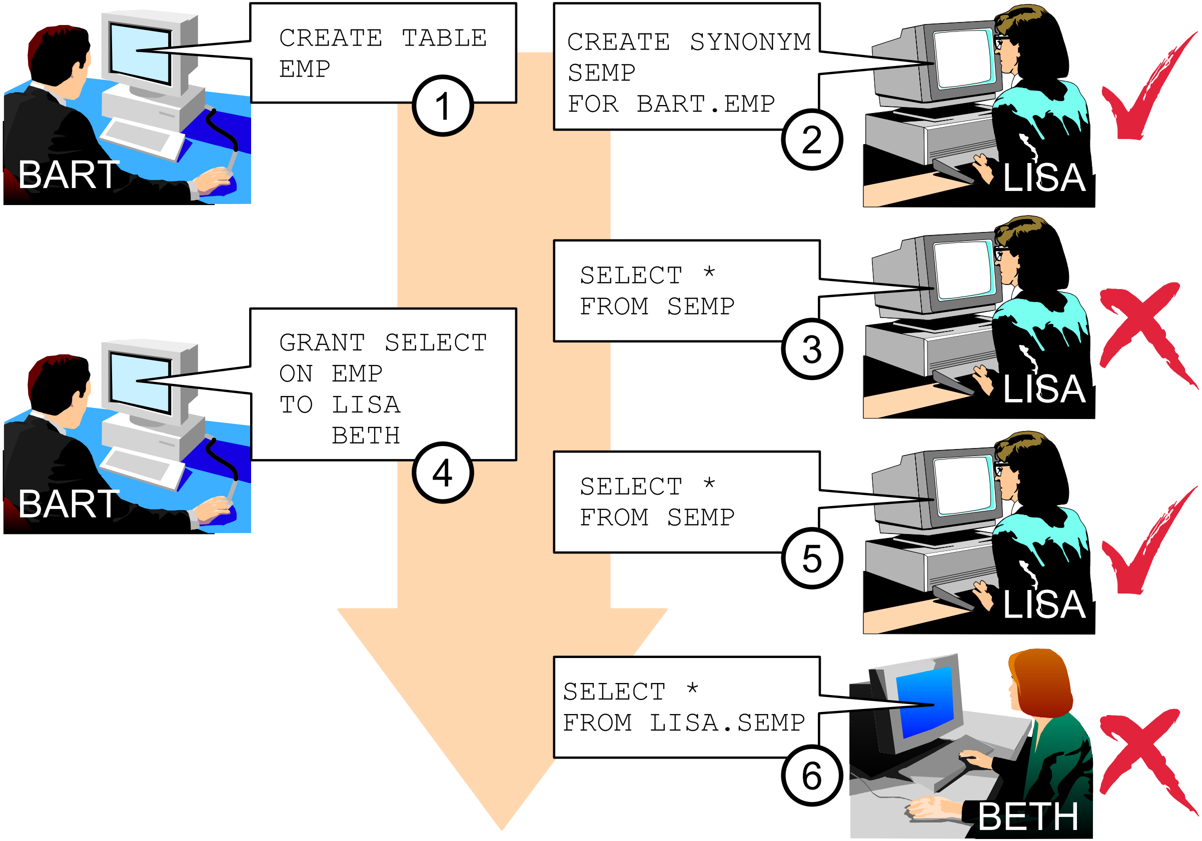
Check Option

- With CHECK OPTION
  - Either LOCAL or CASCADED depending on the release when the view is created
- With LOCAL CHECK OPTION
  - Only verify your own WHERE clause and ignore the definitions of the underlying views
- With CASCADED CHECK OPTION
  - Enforce your WHERE clause and those of any underlying view independent of the definitions of those views

Views - Summary

- Views for simplicity
  - Elimination of unwanted data
  - Elimination of redundant coding
- Views for Security of Data Content
- Views for Data Independence (partial)
- Views to Customize Column Names

SYNONYMs



1
```
CREATE TABLE
EMP
```

2 ✓
```
CREATE SYNONYM
SEMP
FOR BART.EMP
```

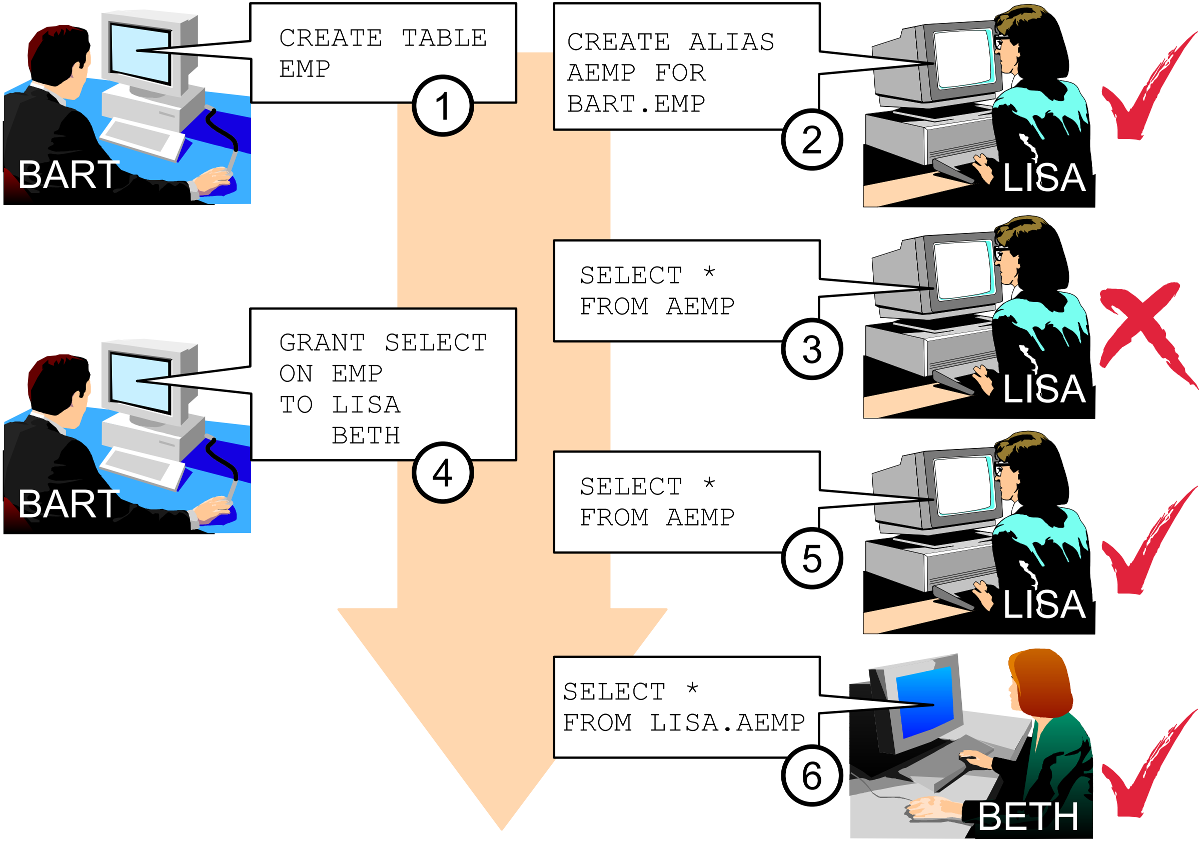
3 ✗
```
SELECT *
FROM SEMP
```

4
```
GRANT SELECT
ON EMP
TO LISA
   BETH
```

5 ✓
```
SELECT *
FROM SEMP
```

6 ✗
```
SELECT *
FROM LISA.SEMP
```

ALIAS



1

```
CREATE TABLE
EMP
```

2 ✓

```
CREATE ALIAS
AEMP FOR
BART.EMP
```

3 ✗

```
SELECT *
FROM AEMP
```

4

```
GRANT SELECT
ON EMP
TO LISA
   BETH
```

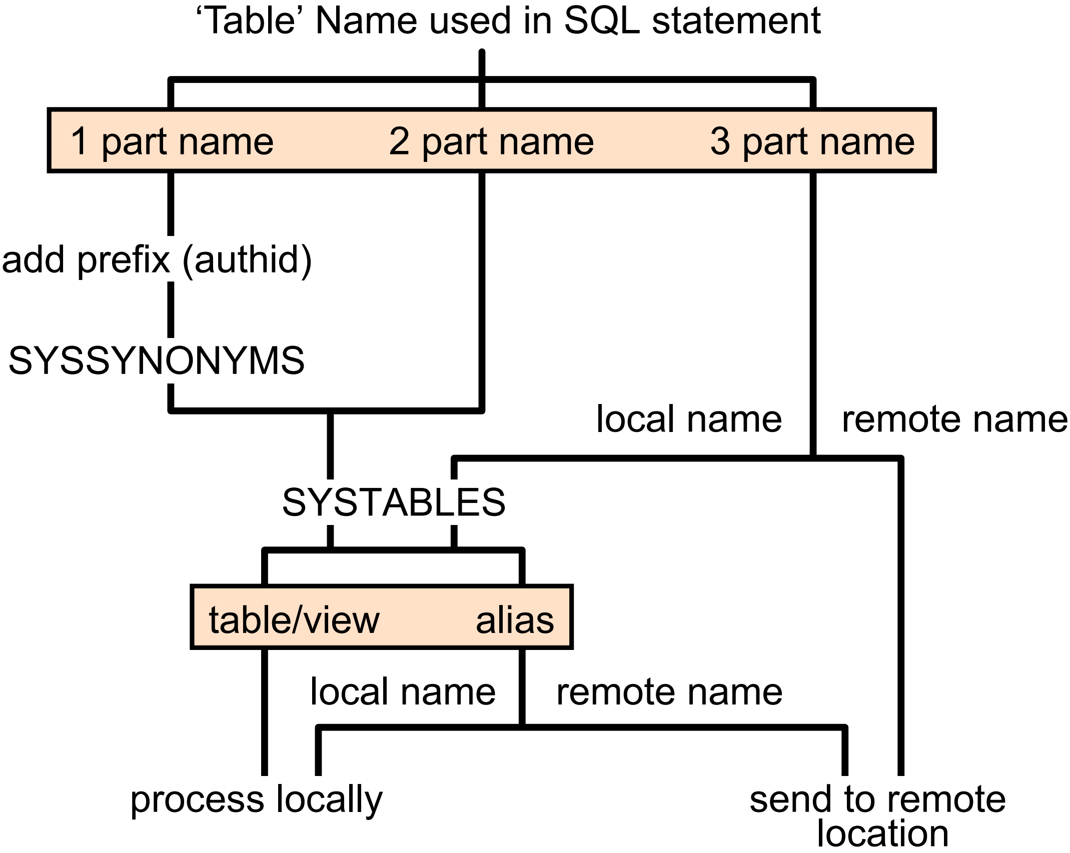
5 ✓

```
SELECT *
FROM AEMP
```

6 ✓

```
SELECT *
FROM LISA.AEMP
```

Object Name Translation



'Table' Name used in SQL statement

1 part name

2 part name

3 part name

add prefix (authid)

SYSSYNONYMS

local name

remote name

SYSTABLES

table/view

alias

local name

remote name

process locally

send to remote location

Views - Catalog Information

```
---------+---------+---------+---------+---------+---------+---
CREATE VIEW EMPLOYEEV AS SELECT EMPNO,DEPT,YEARS FROM EMPLOYEE;
---------+---------+---------+---------+---------+---------+---
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---
```

- SYSIBM.SYSTABLES

```
---------+---------+---------+---------+------
SELECT   NAME,CREATOR,TYPE,COLCOUNT
         FROM SYSIBM.SYSTABLES
WHERE TYPE = 'V'
AND CREATOR = ' KIDDJA';
---------+---------+---------+---------+-----
NAME                CREATOR   TYPE  COLCOUNT
---------+---------+---------+---------+----
EMPLOYEEV           KIDDJA    V            3
```

- SYSIBM.SYSCOLUMNS

```
SELECT NAME,TBNAME,COLNO,COLTYPE,LENGTH,NULLS
FROM SYSIBM.SYSCOLUMNS
WHERE TBNAME = 'EMPLOYEEV';
---------+---------+---------+---------+---------+---------+---------+--
NAME                TBNAME                COLNO  COLTYPE   LENGTH  NULLS
---------+---------+---------+---------+---------+---------+---------+--
EMPNO               EMPLOYEEV                 1  INTEGER        4  N
DEPT                EMPLOYEEV                 2  INTEGER        4  Y
YEARS               EMPLOYEEV                 3  INTEGER        4  Y
```

Views - Catalog Information

```
---------+---------+---------+---------+---------+---------+---
CREATE VIEW EMPLOYEEV AS SELECT EMPNO,DEPT,YEARS FROM EMPLOYEE;
---------+---------+---------+---------+---------+---------+---
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---
```

- SYSIBM.SYSVIEWS

```
SELECT SUBSTR(NAME,1,9) AS NAME,TYPE,TEXT
FROM SYSIBM.SYSVIEWS
WHERE NAME = 'EMPL OYEEV';
---------+---------+---------+---------+---------+---------+---------+---------+
NAME       TYPE  TEXT
---------+---------+---------+---------+---------+---------+---------+---------+
EMPLOYEEV  V     CREATE VIEW EMPLOYEEV AS SELECT EMPNO,DEPT,YEARS FROM EMPLOYEE
```

- SYSIBM.SYSVIEWDEP

```
SELECT DNAME,DTYPE,BNAME,BTYPE
FROM SYSIBM.SYSVIEWDEP
WHERE DNAME = 'EMPLOYEEV';
---------+---------+---------+---------+---------+----
DNAME                  DTYPE  BNAME               BTYPE
---------+---------+---------+---------+---------+----
EMPLOYEEV              V      EMPLOYEE            T
```

SYNONYM - Catalog Information

```
---------+---------+---------+---------+---------+--------
CREATE SYNONYM SEMP1 FOR KIDDJA.EMPLOYEE;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
CREATE SYNONYM SEMP2 FOR KIDDJA.EMPLOYEEV;;
---------+---------+---------+---------+---------+--------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+--------
```

● SYSIBM.SYSSYNONYMS

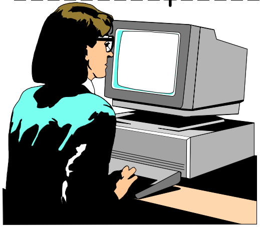
```
---------+---------+---------+---------+---------+---------+---
SELECT NAME,CREATOR,TBNAME,TBCREATOR
FROM SYSIBM.SYSSYNONYMS
WHERE CREATOR = 'KIDDJA';
---------+---------+---------+---------+---------+---------+---
NAME                CREATOR   TBNAME              TBCREATOR
---------+---------+---------+---------+---------+---------+---
SEMP1               KIDDJA    EMPLOYEE            KIDDJA
SEMP2               KIDDJA    EMPLOYEEV           KIDDJA
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---
```

ALIAS - Catalog Information

```
---------+---------+---------+---------+---------+---------+
CREATE ALIAS AEMP1 FOR KIDDJA.EMPLOYEE;
---------+---------+---------+---------+---------+---------+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+
CREATE ALIAS AEMP2 FOR KIDDJA.EMPLOYEEV;
---------+---------+-----
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+
```

- SYSIBM.SYSTABLES

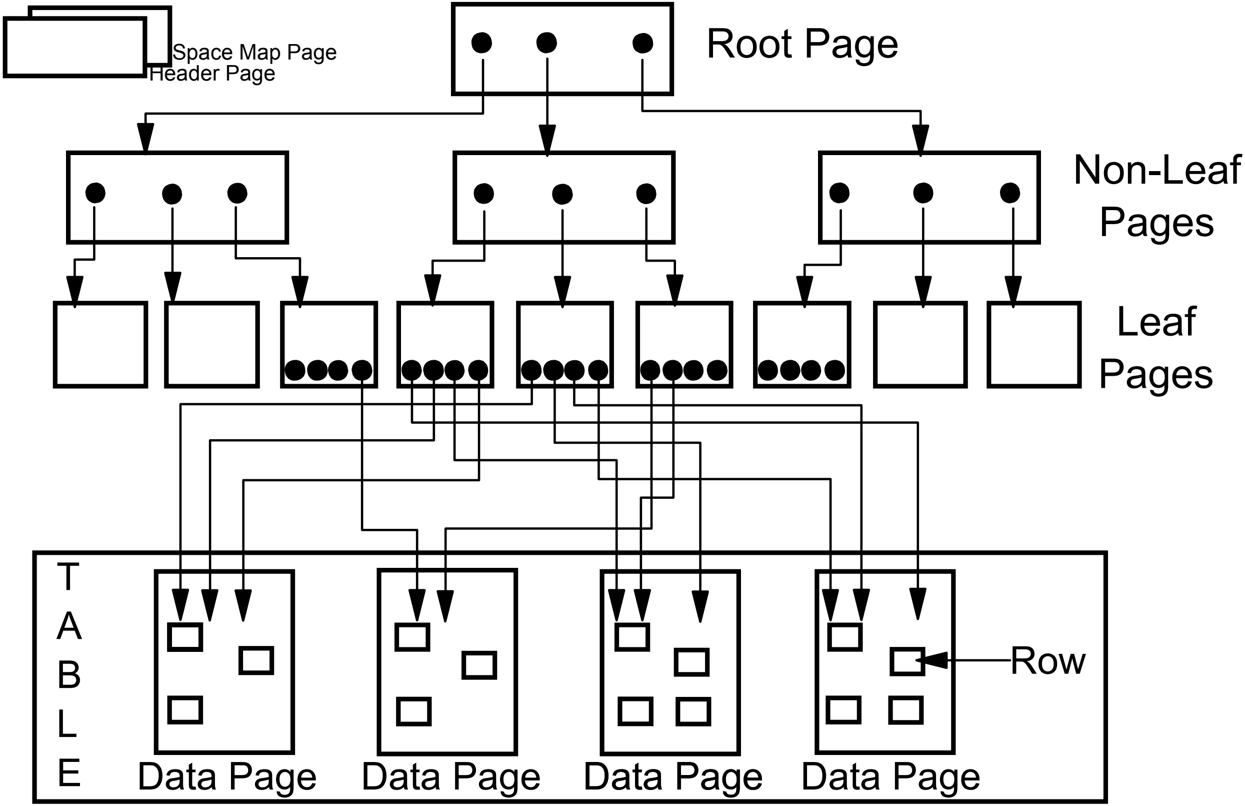
```
---------+---------+---------+---------+---------+---------+---
SELECT   NAME,CREATOR,TYPE,COLCOUNT
         FROM SYSIBM.SYSTABLES
WHERE TYPE = 'A'
AND CREATOR = 'KIDDJA';
---------+---------+---------+---------+---------+---------+---
NAME                CREATOR   TYPE  COLCOUNT
---------+---------+---------+---------+---------+---------+---
AEMP1               KIDDJA    A            0
AEMP2               KIDDJA    A            0
DSNE610I NUMBER OF ROWS DISPLAYED IS 2
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+---
```



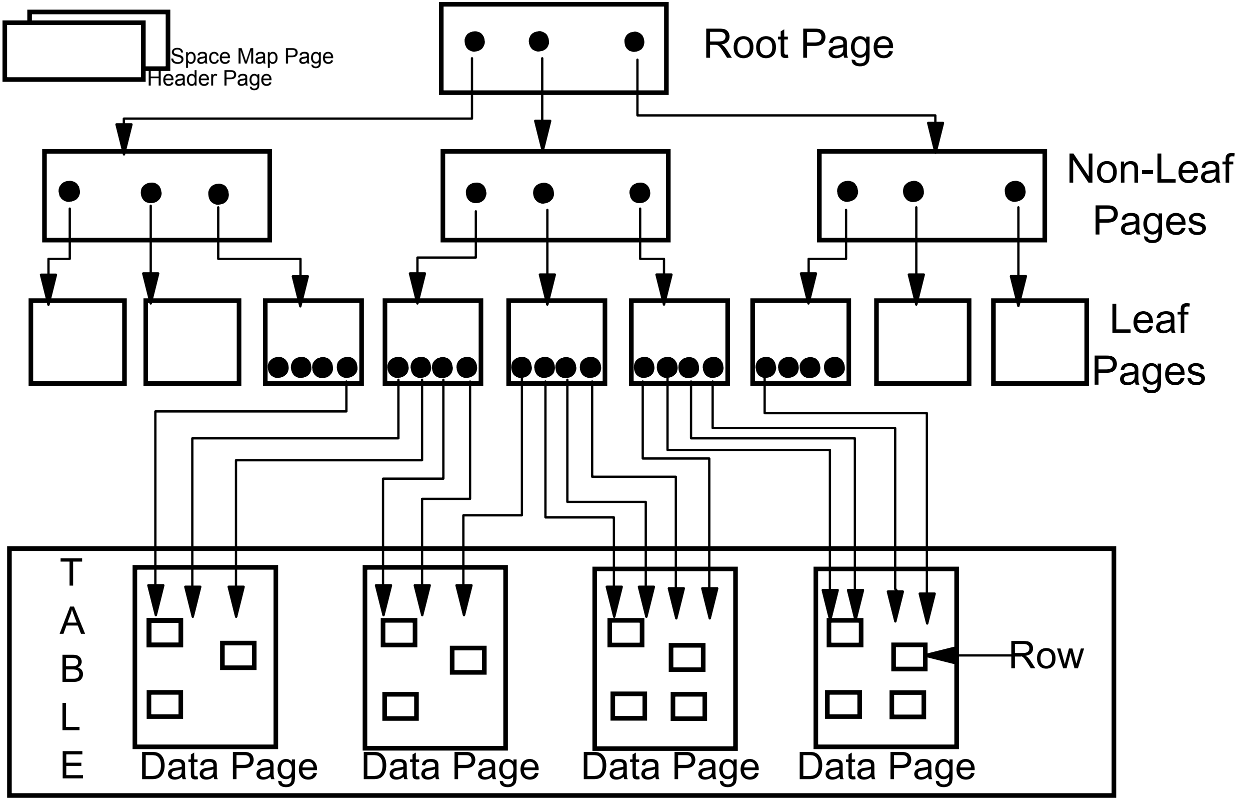
Indexspace Physical Organization

- What?
  - An indexspace is the set of VSAM Linear Datasets that hold the index data
  - An Index (space) is implicitly associated with the database that contains the table the index is defined on
- Index name
  - Maximum 18 alphanumeric bytes (DBCS or SBCS)
  - Prefixed with owner name
  - "owner"."indexname" must be unique within the DB2 subsystem

DB2 Index Tree Structure



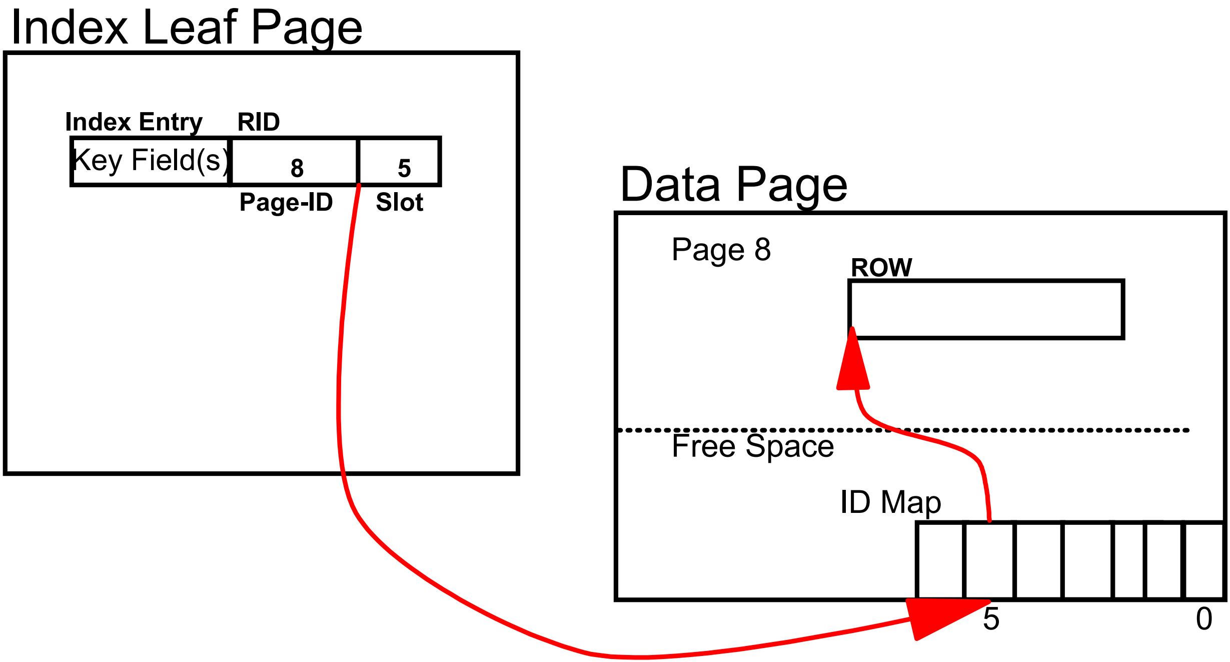
Clustering Index



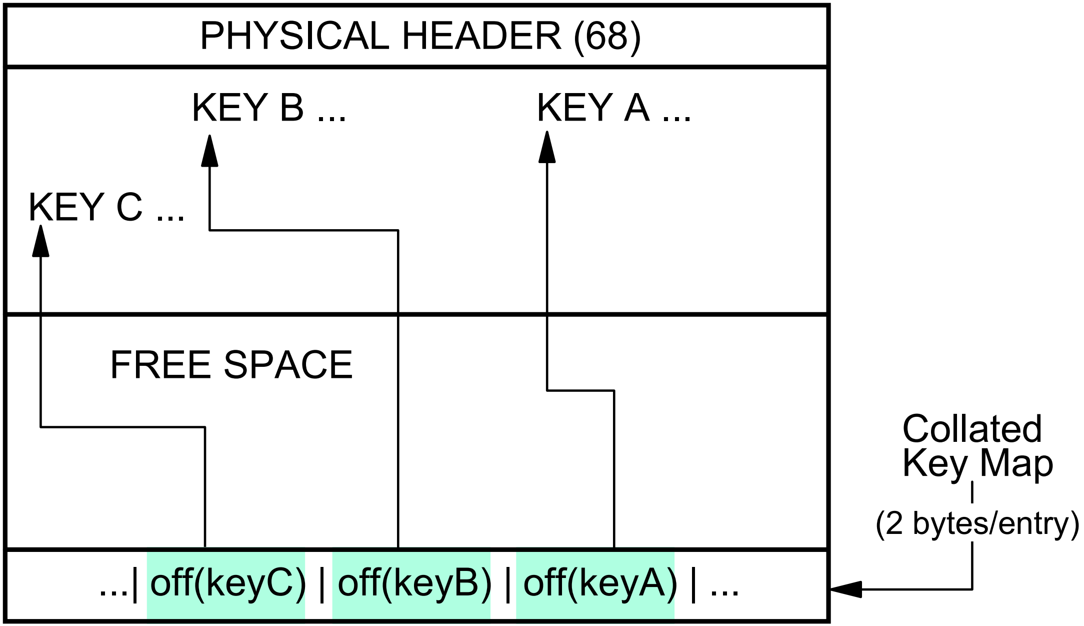
Clustering - Considerations

- DB2 will attempt to INSERT rows in chosen sequence if you create a CLUSTERing index on that key
- Only one CLUSTERing index per table
- Not supported for simple multitable tablespaces
- Required for the partitioning index of a partitioned tablespace
- Consider defining one index on each table as CLUSTERing

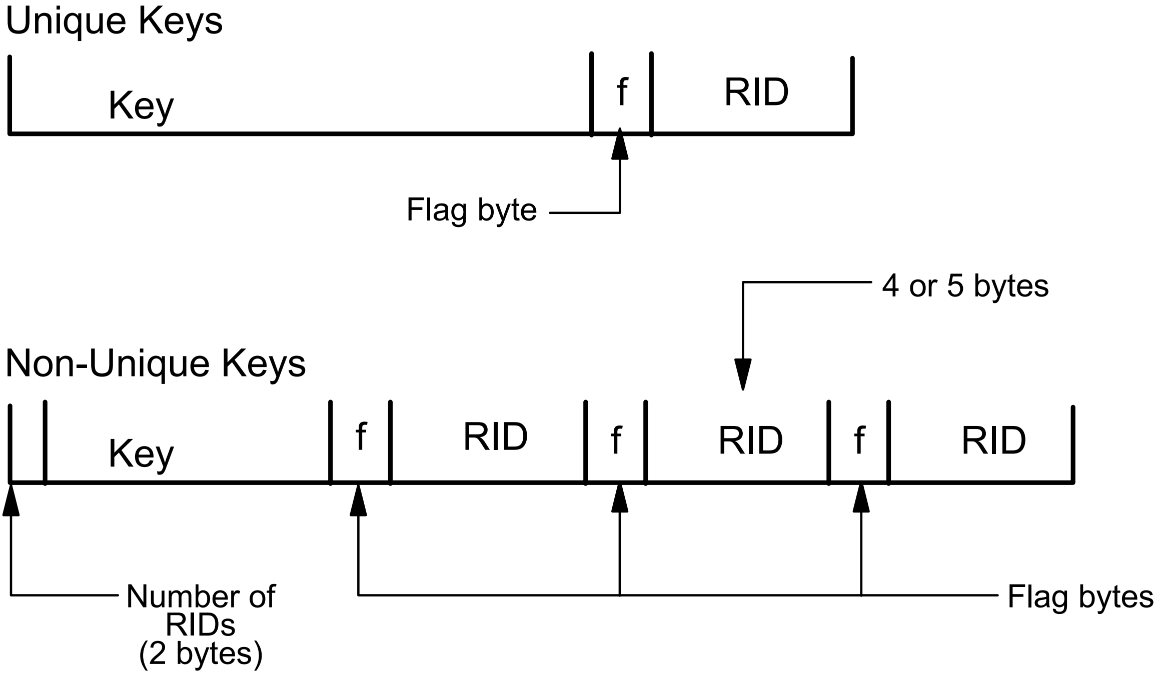
DB2 Addressing Scheme



Leaf Page Format



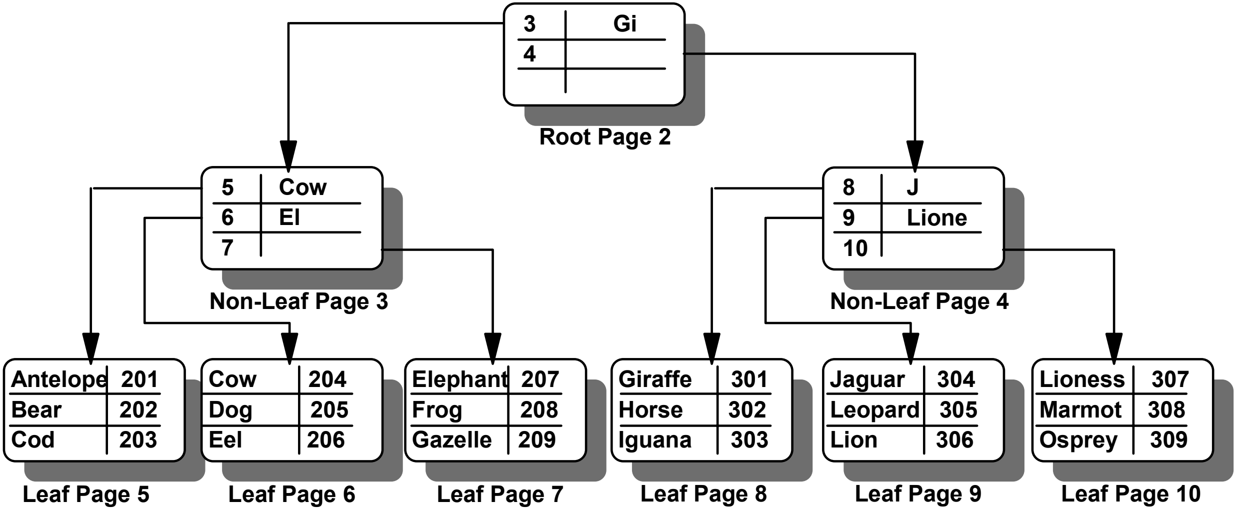
Leaf Page - Index Entry Format



Index Entry Format

- Variable length keys are padded with blanks (to maximum length) to create fixed length keys
- If the fields can be null, one byte null indicator stored with key
- NOT NULL WITH DEFAULT keys are stored as NOT NULL keys, for example, no null indicator bytes
- UNIQUE WHERE NOT NULL indexes are stored as non-unique keys
- Keys are not encoded by user EDITPROC
- Keys are not compressed

Non-Leaf Pages



3	Gi
4	

Root Page 2

5	Cow
6	El
7	

Non-Leaf Page 3

8	J
9	Lione
10	

Non-Leaf Page 4

Antelope	201
Bear	202
Cod	203

Leaf Page 5

Cow	204
Dog	205
Eel	206

Leaf Page 6

Elephant	207
Frog	208
Gazelle	209

Leaf Page 7

Giraffe	301
Horse	302
Iguana	303

Leaf Page 8

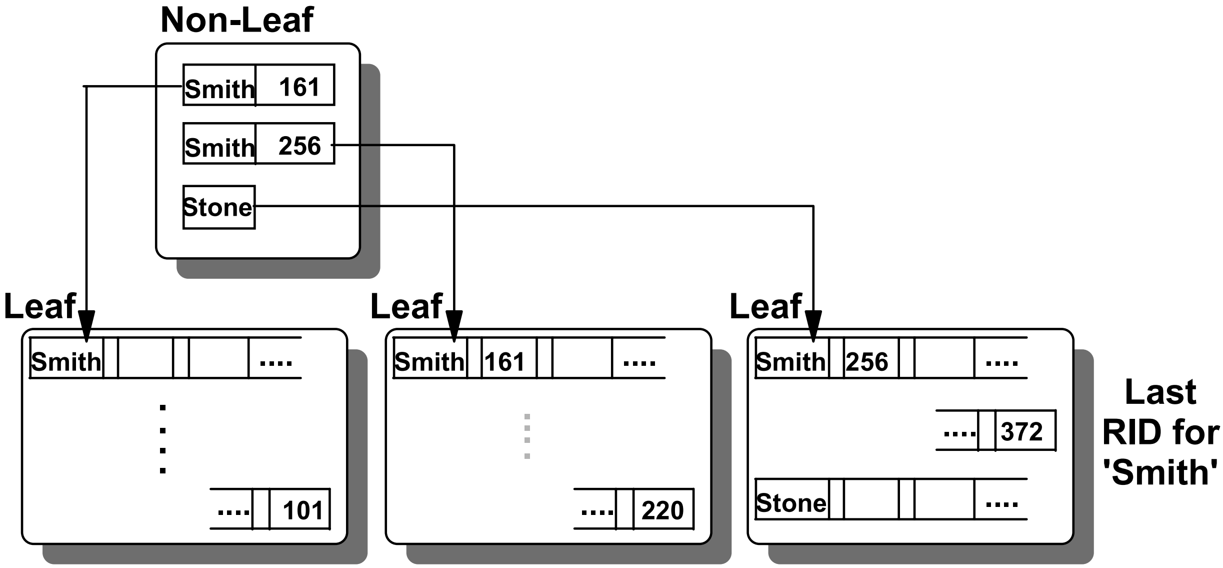
Jaguar	304
Leopard	305
Lion	306

Leaf Page 9

Lioness	307
Marmot	308
Osprey	309

Leaf Page 10

RID Chains


Non-Leaf
Smith 161
Smith 256
Stone
Leaf
Smith
....
101
Leaf
Smith 161
....
220
Leaf
Smith 256
....
....372
Stone
....
Last
RID for
'Smith'

Index Page Management - INSERT

INSERT

- Space search sequence once the correct leaf page is found:

1. If inserting just a new RID:
   - Check for space in the existing RID map.
2. If no space in RID map:
   - Create a RID chain for the new RID.
3. If inserting a key and RID:
   - Allocate space for the new index entry and add a new key map entry, moving other key map entries as needed to maintain collating sequence in the key map.
4. If no space available in the page:
   - Split the leaf page, moving half the keys to the new page. An exception is the case where the new index entry is the highest key in the index, in which case, only the new entry is moved to the new page.

Index Page Management - UPDATE

UPDATE

- Delete and reinsert the key value in the index
- If set to null:
  - Set the null indicator to “FF”
  - Delete the key value
  - Move the record to the “high” key end of the index

Index Page Management - DELETE

PSEUDO DELETE

- Turn on ‘pseudo delete’ in index entry flag byte
  - (Used when page or row lock is held)

PHYSICAL DELETE

- If deleting just a RID
  - Remove it from the RID list or the RID chain
- If deleting a key and RID:
  - Free the space and remove the key from the key map
    - (Used when tablespace or table lock is held)

CREATE INDEX

```
CREATE INDEX DEPTNO_DATA_INDEX
ON TPTOJ
(DEPTNO DESC,PRSTDATE)
..............................
USING STOGROUP GROUP90
PRIQTY 40 SECQTY 4
ERASE NO
BUFFERPOOL BP0
CLOSE YES
FREEPAGE 0
PCTFREE 10
DEFER NO
COPY YES
```

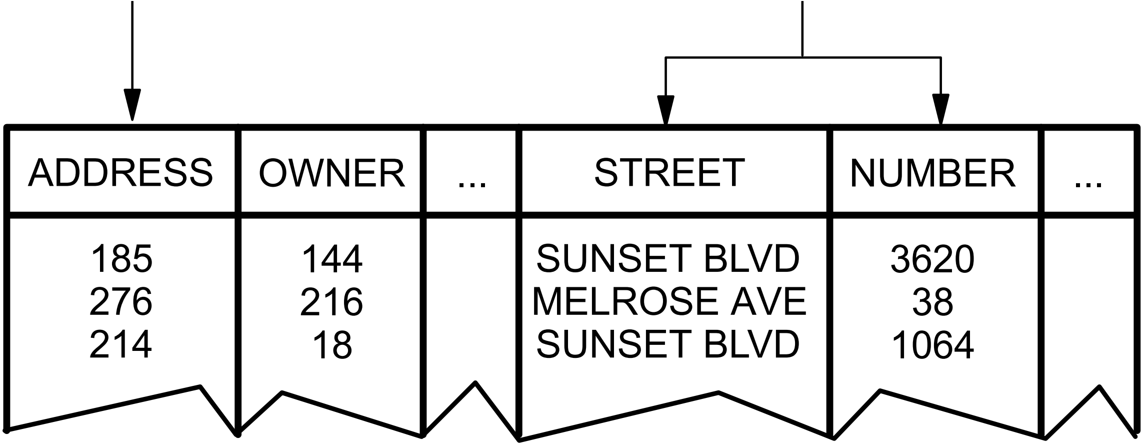
ALTER INDEX

```
ALTER INDEX DEPTNO_DATA_INDEX
  BUFFERPOOL BP2
  CLOSE NO
  FREEPAGE 5
  PCTFREE 15
  USING STOGROUP GROUP83
  PRIQTY 200
  SECQTY 40
  ERASE NO
```

CREATE INDEX - Examples

```
CREATE UNIQUE WHERE NOT NULL
INDEX I_ADDRESS_0
ON ADDRESS(ADDRESS)
```

```
CREATE INDEX I_ADDRESS_1
ON ADDRESS(STREET,NUMBER)
CLUSTER
```



ADDRESS	OWNER	...	STREET	NUMBER	...
185 276 214	144 216 18		SUNSET BLVD MELROSE AVE SUNSET BLVD	3620 38 1064	

Indexes - Catalog Information

- SYSIBM.SYSINDEXES

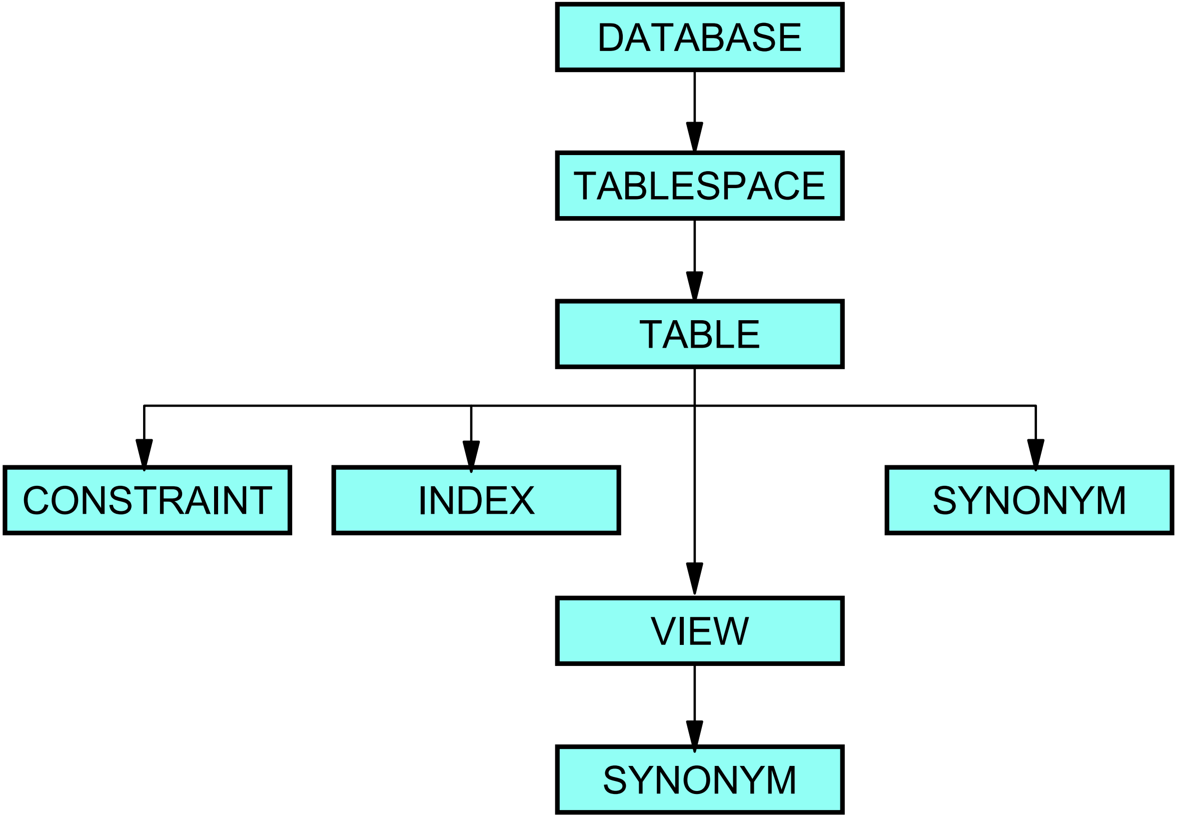
```
---------+---------+---------+---------+---------+---------+--
SELECT   NAME,INDEXSPACE,TBNAME,UNIQUERULE,COLCOUNT,
         CLUSTERING,CLUSTERED,CLUSTERRATIO,
         BPOOL,COPY
FROM SYSIBM.SYSINDEXES
WHERE NAME LIKE 'GB%'
---------+---------+---------+---------+---------+---------+--
NAME                INDEXSPACE  TBNAME              UNIQUERULE
---------+---------+---------+---------+---------+---------+--
GBIXEMPNO           GBIXEMPN    EMPLOYEE            D

-------+---------+---------+---------+---------+---------+------
  COLCOUNT  CLUSTERING  CLUSTERED  CLUSTERRATIO  BPOOL     COPY
-------+---------+---------+---------+---------+---------+------
         1  Y           Y                     0  BP1       Y

DSNE610I NUMBER OF ROWS DISPLAYED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
---------+---------+---------+---------+---------+---------+-
```

Indexes - Catalog Information

- SYSIBM.SYSINDEXPART

```
---------+---------+---------+---------+---------+---------+-
SELECT   IXNAME,PARTITION,
         PQTY,SQTY,STORNAME,VCATNAME,PCTFREE,FREEPAGE,SPACE
FROM SYSIBM.SYSINDEXPART
WHERE IXNAME LIKE 'GB%'
---------+---------+---------+---------+---------+---------+-
IXNAME                   PARTITION         PQTY     SQTY   STORNAME
---------+---------+---------+---------+---------+---------+-
GBIXEMPNO                        1         1800      180   GBCF831S
GBIXEMPNO                        2         3600      360   GBCF830S
GBIXEMPNO                        3         3600      360   GBCF830S
GBIXEMPNO                        4         3600      360   GBCF830S
--------+----------+----------+----------+------
 VCATNAME   PCTFREE   FREEPAGE        SPACE
--------+----------+----------+----------+------
 GBCF83          15          8            0
 GBCF83          10          4            0
 GBCF83          10          4            0
 GBCF83          10          4            0
 DSNE610I NUMBER OF ROWS DISPLAYED IS 4
 DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
 ---------+---------+---------+---------+---------+---------+-
```

Object Dependencies



DATABASE

TABLESPACE

TABLE

CONSTRAINT

INDEX

SYNONYM

VIEW

SYNONYM

Dropping an Object

- Drop with SQL statement DROP
  - Example: DROP TABLE EMP
- Cascade effect on dependent objects and definitions:
  - Indexes
  - Views
  - Synonyms
  - Security definitions
  - RI definitions
  - Check Constraints

Unit Summary

- This unit has covered the following:
  - The DB2 objects that make up a DB2 Database.
  - The most appropriate parameters for these objects so that they can be implemented with the most appropriate attributes.
  - Storage groups, databases, tablespaces, tables, views, indexes, synonyms and aliases.
  - How to alter the attributes of DB2 Database objects as requirements change over time.
  - How data is stored in a DB2 Database.