Disclaimer and Trademarks

Disclaimer
The information in this presentation concerns products and features that are currently under consideration.
Some of the features described in this presentation may change.
References in this publication to IBM products, programs, or services do not imply
that IBM intends to make these available in all countries in which IBM operates.
IBM MAY HAVE PATENTS OR PENDING PATENT APPLICATIONS COVERING
SUBJECT MATTER IN THIS DOCUMENT. THE FURNISHING OF THIS DOCUMENT
DOES NOT IMPLY GIVING LICENSE TO THESE PATENTS.

Trademarks
THE FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF
THE IBM CORPORATION IN THE UNITED STATES AND/ OR OTHER COUNTRIES:
AIX, AS/ 400, DB2, ETHERAND, OPERATING SYSTEM/ 2, OS/ 390, OS/ 400, ES/ 9000,
MVS/ ESA, OS/ 2, PERSONAL SYSTEM/ 2, PRESENTATION MANAGER, PS/ 2, RISC,
RISC SYSTEM/ 6000, SYSTEM/ 390, SQL/ DS, TOKEN RING, VM/ ESA, IBM, APPROACH

THE FOLLOWING TERMS ARE TRADEMARKS OR REGISTERED TRADEMARKS OF
THE MICROSOFT CORPORATION IN THE UNITED STATES AND/ OR OTHER
COUNTRIES: MICROSOFT, WINDOWS, WINDOWS NT, ODBC, WINDOWS 95
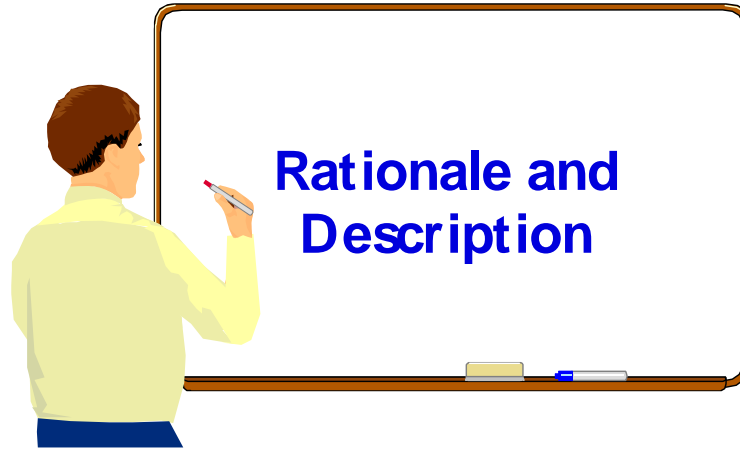
GoFurther

Agenda

- Rationale and description

- DDL statements
  - ALTER TABLE . . . ADD CLONE
  - EXCHANGE DATA BETWEEN . . .

- Authorization

- Catalog and directory changes

- Commands and utilities

GoFurther

Rationale and Description

GoFurther

Rationale

- ". . . many of our processes involve completely emptying the table and then reloading it in its entirety. In order to accomplish this the table space must be offline which does impact the availability of the business application . . . "

- ". . . It is increasingly difficult to do LOAD REPLACEs of data which is required for business as their business becomes more and more 24x7x365 with online transactions 24 hrs a day."

GoFurther

Description

Cloned table support provides the ability to generate a table with the exact same attributes as a table that already exists at the current server. This new table that's being created is referred to as the **clone** table of a **base** table.

Not only is the clone table that's created identical to the base table in every way:

- same number of columns
- same column names
- same data types
- same table check constraints, etc.

It's also created with the same
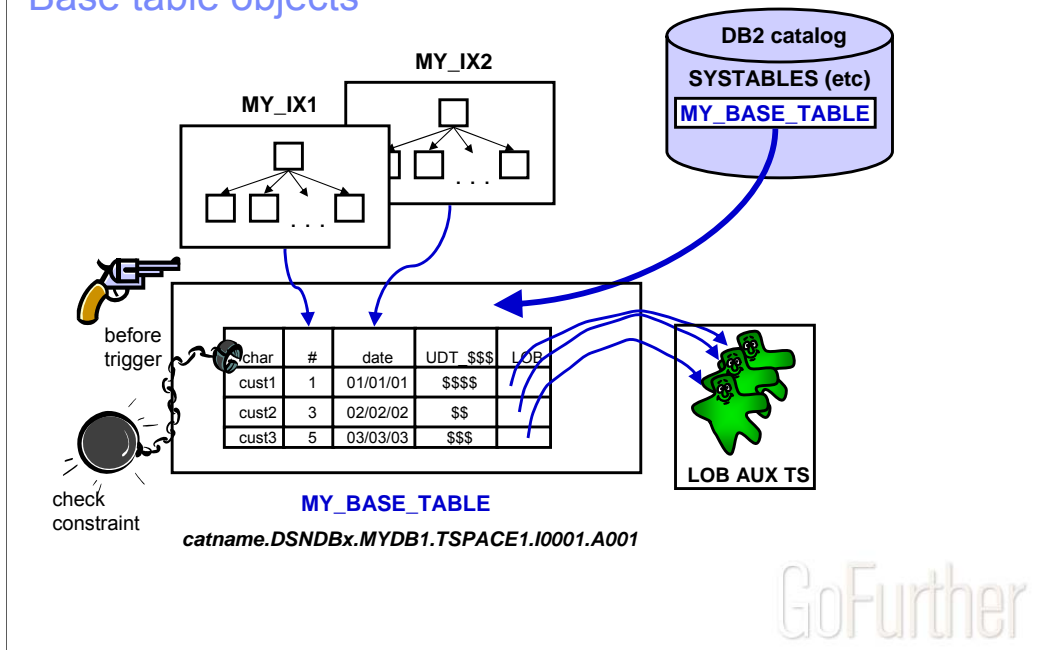
- indexes
- before triggers, etc.

Not cloned?

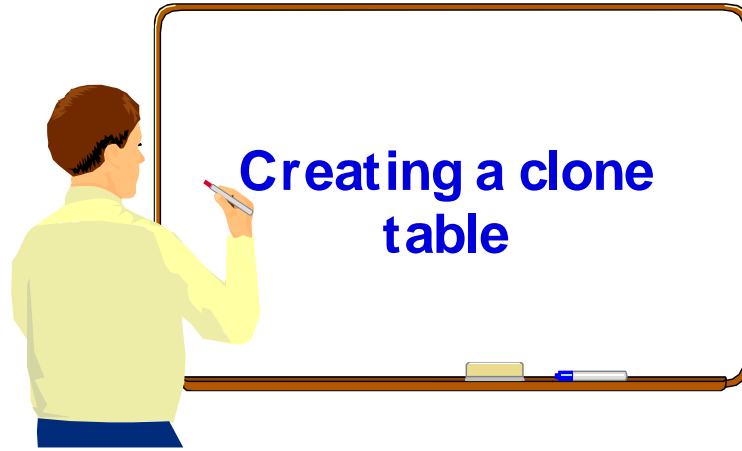- Views and authorization

GoFurther

Views and authorization are not created when the clone table is created. Base and clone tables can have different views and different authorization sets.

Base table objects

IDUG®2007 North America

MY_IX2
MY_IX1

DB2 catalog
SYSTABLES (etc)
MY_BASE_TABLE

before trigger

char	#	date	UDT_$$$	LOB
cust1	1	01/01/01	$$$$	
cust2	3	02/02/02	$$	
cust3	5	03/03/03	$$$	

check constraint

LOB AUX TS

MY_BASE_TABLE
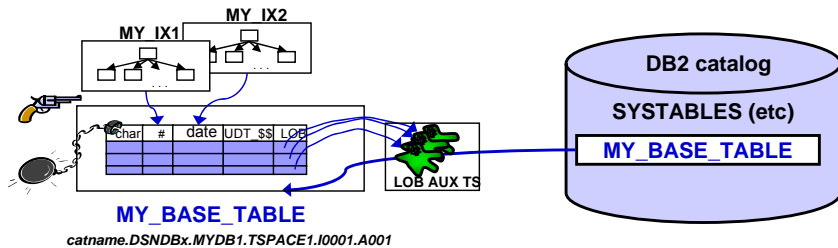catname.DSNDBx.MYDB1.TSPACE1.I0001.A001

GoFurther

Here is an existing base table that has a couple indexes, a before trigger and a table check constraint. There is also a LOB column in the table so you can see there is an auxiliary table as well. This table already has some data in it.

This table is referred to as the 'base' table. This should not be confused with the 'base' terminology which is also used with the description of LOB objects.

Creating a clone table

ALTER TABLE . . . ADD CLONE

MY_IX1   MY_IX2

DB2 catalog

SYSTABLES (etc)

MY_BASE_TABLE

char # date UDT $$ LOB

LOB AUX TS

MY_BASE_TABLE

catname.DSNDBx.MYDB1.TSPACE1.I0001.A001
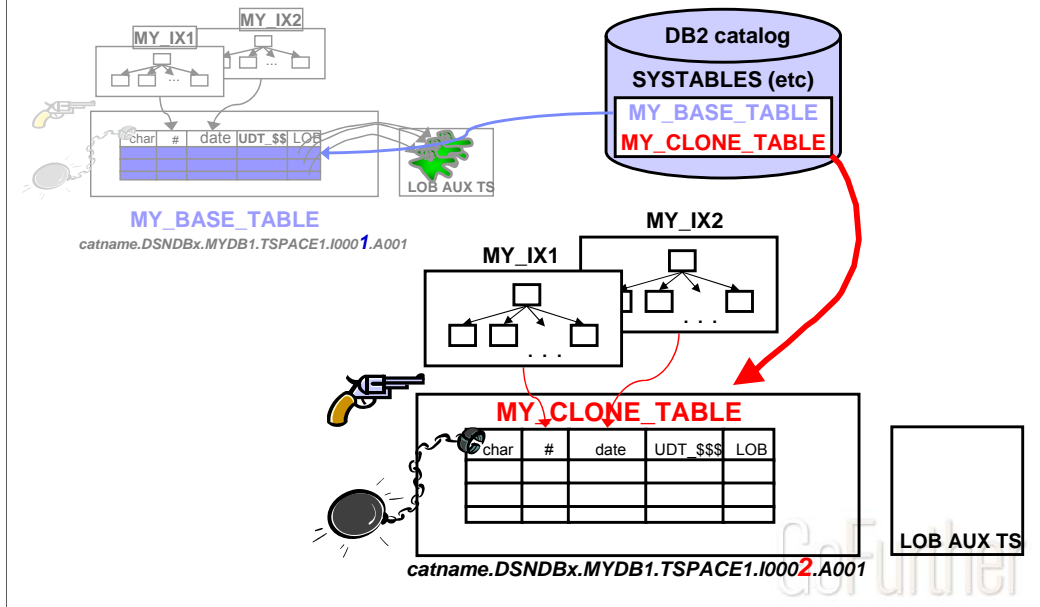
ALTER TABLE **MY_BASE_TABLE**

ADD CLONE **MY_CLONE_TABLE**

GoFurther

The base table MUST be in a Universal Table Space before a clone table can be created. Here we're going to create a clone table of base table MY_BASE_TABLE that already exists in a UTS. Although we'll be creating a clone table we have to use the ALTER TABLE syntax to create the clone table.

Base table and clone

DB2 generates all objects:
- table space (a universal table space (UTS))
- table, with all column names and data types the same
- before triggers
- LOB objects (aux table space, indexes)
- Table check constraints
- indexes

No data is cloned so after the create (via ALTER TABLE) completes successfully the new table is empty.

Note that the clone table is represented by instance number 2 data sets.

Notes about the clone

- Catalog tables cannot be cloned

- Could be created with a different schema (owner) than the base table

- Once created, access must be granted to users -- the clone functions like a "normal" DB2 table

- Like a "normal" DB2 table, after creation, the clone objects (table, indexes, aux objects, etc.) are empty. They can be populated by:

  ‣ INSERT
  ‣ LOAD
  ‣ etc.

GoFurther

A clone table only exists if there was originally a base table. You cannot have a clone table without a base table. But after the clone table is created it is essentially it's own entity. You can grant users access to the clone, you can load it with completely separate data, etc. Conceptually you can think of it as being completely separate from the base table.

Loading and inserting data into the clone
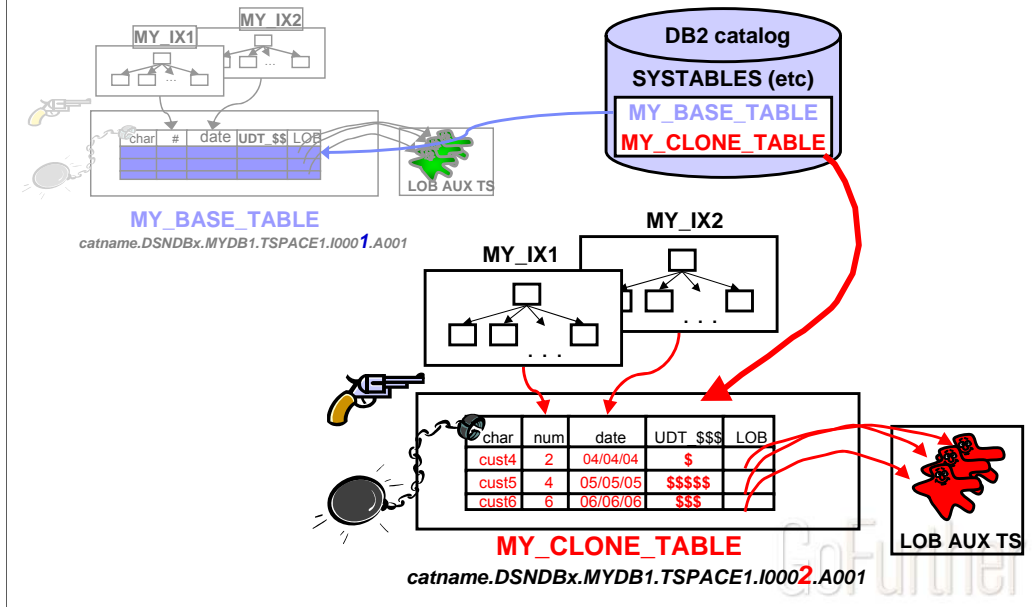
DB2 generates all objects:
•table space
•table, with all column names and data types the same
•before triggers
•LOB objects (aux table space, indexes)
•Table check constraints
•indexes

No data is cloned. But here we show the clone table after it's been LOADed (via LOAD, INSERT, etc) with data.

Exchanging data
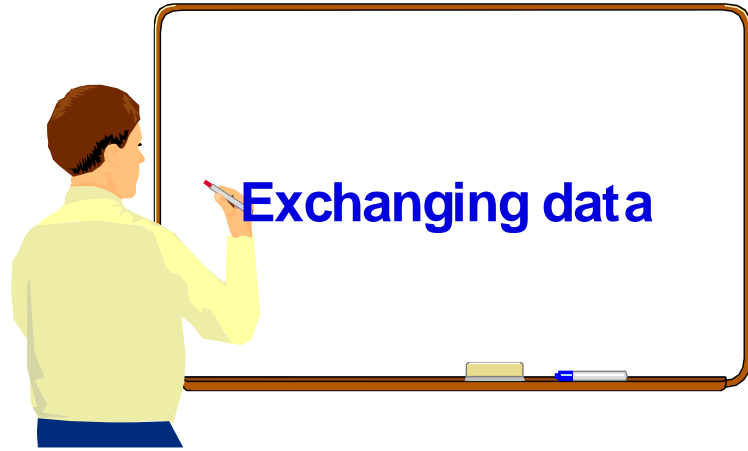
GoFurther

EXCHANGE statement

➤➤── **EXCHANGE DATA BETWEEN** *table-name-1* ─────➤

──────────── **AND** *table-name-2* ─➤◄

GoFurther

table-name-1 and table-name-2 -- base table and clone table names can be specified in any order.

EXCHANGE notes

- EXCHANGE DATA BETWEEN . . .

  - Will switch the data set **instance numbers**. This *effectively* switches the data between the base and clone table objects. ***No data is actually moved!***

    - *Before EXCHANGE:*

      catname.DSNDBx.MYDB1.TSPACE1.I0001.A001   BASE

      catname.DSNDBx.MYDB1.TSPACE1.I0002.A001 CLONE

    - *After EXCHANGE:*

      catname.DSNDBx.MYDB1.TSPACE1.I0002.A001   BASE

      catname.DSNDBx.MYDB1.TSPACE1.I0001.A001 CLONE

  - Very quick operation.

  - Only applicable if cloning in effect.

GoFurther

No data is moved during the EXCHANGE process. We only 'switch' the data set instance numbers. This might make it look like we've moved data back and forth but we don't ever do that. The instance number changes happen for all objects that have underlying VSAM data sets (ie table and index data).
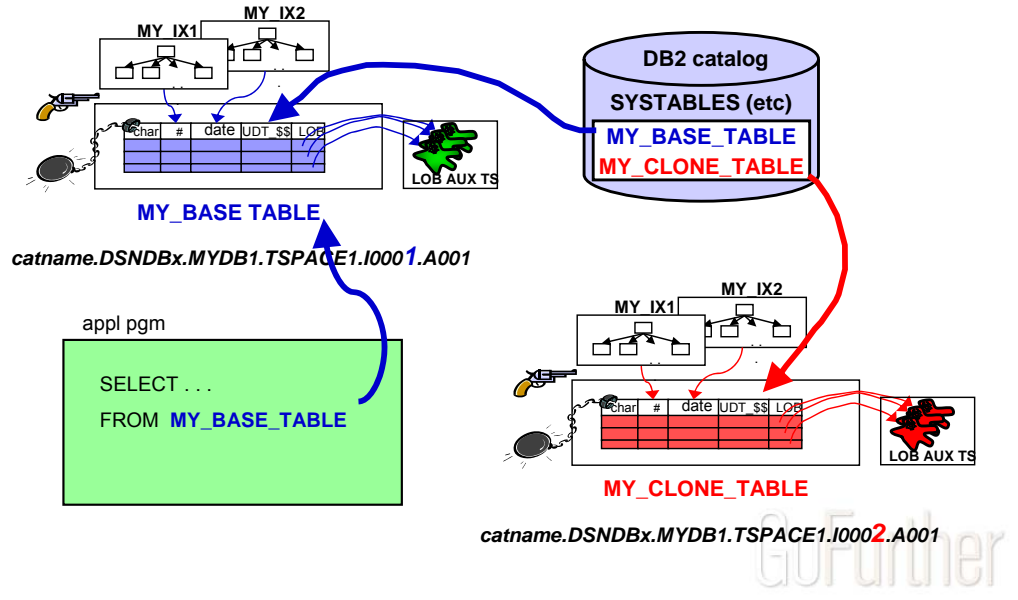
In this slide we're doing a SELECT from the base table which is currently represented by instance number 1 data sets. DB2 knows where to go to get the data (ie instance number 1 data sets) and which indexes to use.

EXCHANGE statement

- Table names in EXCHANGE statement can be specified in any order:

  ‣ EXCHANGE DATA BETWEEN MY_BASE_TABLE AND MY_CLONE_TABLE

  or

  ‣ EXCHANGE DATA BETWEEN MY_CLONE_TABLE AND MY_BASE_TABLE

Applications are bound to the base table.

An authorized person submits the EXCHANGE statement.

Plans and packages are not invalidated.

EXCHANGE (after)
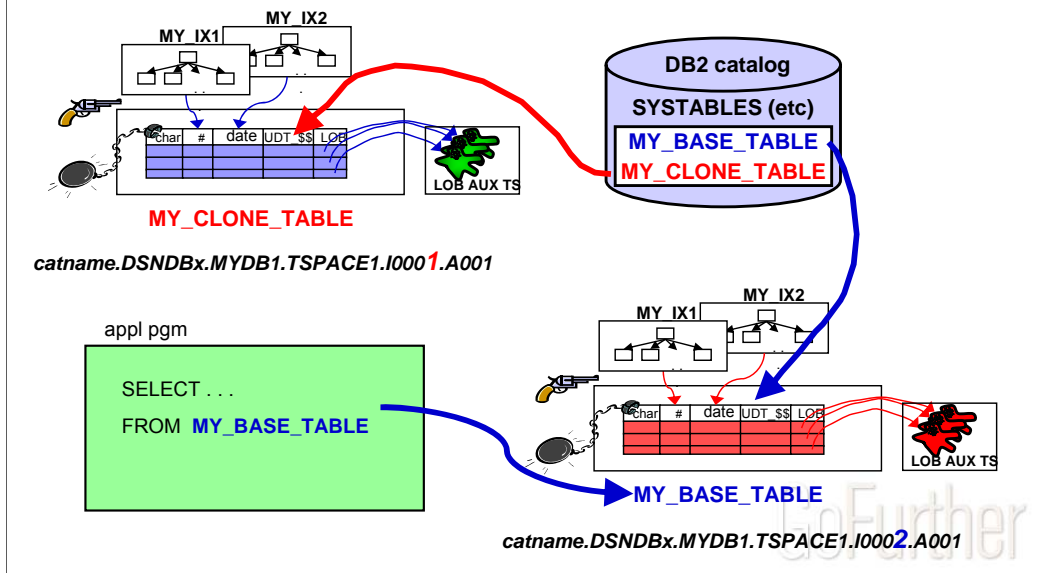
After the EXCHANGE, the base table data is now found in the instance number 2 data sets and the clone table data in the instance number 1 data sets. When you SELECT data from the base table, DB2 knows that is now has to look in the instance number 2 data sets for it's data. If you were to SELECT data from the clone table, DB2 would know that is now has to look in the instance number 1 data sets for it's data.

Notes about cloning

- Can only create a clone if the table space is a universal table space (UTS)

- Can only create a clone table in a DB2 managed table space

- Can only create a clone for a table if it's the only table in a table space
  - A UTS is partitioned by definition

- No FASTSWITCH if there's a clone
  - Can create a clone if base table has a mix of I/J datasets. However, cannot fix/change this as long as the clone exists.

GoFurther

Clone tables can only be created if the base tables are in a universal table space. This means that clone tables (and their associated base table) will be partitioned.

FASTSWITCH is not allowed because the I/J data set information is kept in the DB2 catalog (for tables it's in SYSTABLEPART.IPREFIX and indexes it's kept in SYSINDEXPART.IPREFIX) and this information is 'shared' between the base and clone objects. This means that there will be a single row in SYSTABLEPART for a given partition and this single row is used for BOTH the base and clone table. If we were to allow FASTSWITCH to be used then the catalog would be updated for one object and then we'd be out of synch with the other object. To prevent this from happening we do not allow FASTSWITCH to be used.

Notes about cloning

- No ALTERs (online schema changes for prod/cloned table. For example, no ALTERing column data types, adding columns, etc.)

  ‣ Because most of catalog is 'shared' by the base and clone objects

- Cannot clone a table that has RI or add RI to a table if it is involved in cloning

- After triggers NOT allowed on either table. Only before triggers allowed.

- No clones allowed on MQTs

GoFurther

Most of these restrictions are there because catalog information is shared between the base and clone objects. Some of them (ie the restrictions with referential integrity, after triggers, etc) are there because of time and/or resource restrictions. We may see some of these restrictions eliminated in the future.

Notes about cloning

- **Can** create an index on a base table

  ▸ Creates index on both base and clone tables

  ▸ Clone table indexes are immediately put into RBDP/PSRBD if clone table contains data

- **Can** create a before trigger on the base table

  ▸ Creates before trigger on both base and clone tables

GoFurther

Statistics and the clone

- No RUNSTATS against clone table

- On EXCHANGE, the clone becomes the base and it assumes the base objects' statistics

  ‣ Design premise: base and clone data will be similar enough to where statistics can apply to both base and clone

  ‣ This allows bound static SQL to function without a rebind

GoFurther

Most of the catalog information is shared for base and clone objects. Because of this we will not allow/keep statistics for the clone objects. We believe that base and clone data will be similar enough that the same statistics could be used when accessing data from either set of objects. If the data is not that close then you may need to run RUNSTATS after an EXCHANGE has been done so that the statistics more closely match the actuals for the current base objects.

DDL
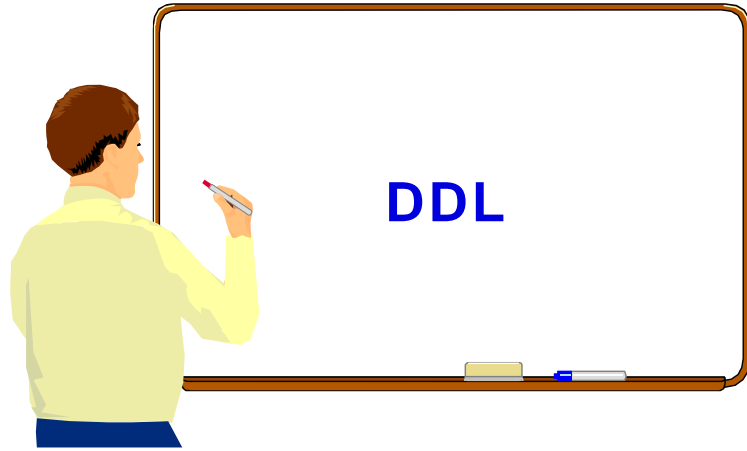
GoFurther

DDL . . .

- ALTER TABLE *base-table-name* DROP CLONE

  ▸ Drops only the clone table objects and their underlying data sets.

- DROP TABLE will drop base table and its clone.

GoFurther

If you only want to DROP the clone objects then you can do so by using the ALTER TABLE DROP CLONE syntax. This will delete only the clone objects. If you want to get rid of both base and clone objects then use the DROP TABLE syntax.

Authorization

GoFurther

Authorization

- DML authorization on base and clone can be different

- EXCHANGE DATA BETWEEN . . . operation requires one of these:
  - ▸ Ownership of both tables
  - ▸ INSERT and DELETE privileges for both tables
  - ▸ DBADM for the database
  - ▸ SYSADM auth

- ALTER TABLE table-name ADD CLONE clone-table-name
  - ▸ Same privilege set as for CREATE TABLE for table being cloned:
    - ▪ CREATETAB for the database
    - ▪ DBADM, DBCTRL, DBMAINT
    - ▪ SYSADM, or SYSCTRL

GoFurther

For the EXCHANGE DATA statement, the privileges held by the authorization ID of the statement must include at least one of the following:

•The INSERT privilege and DELETE privilege for both tables

•Ownership of both tables

•DBADM authority on the database that contains the tables

•SYSADM authority

Catalog implications
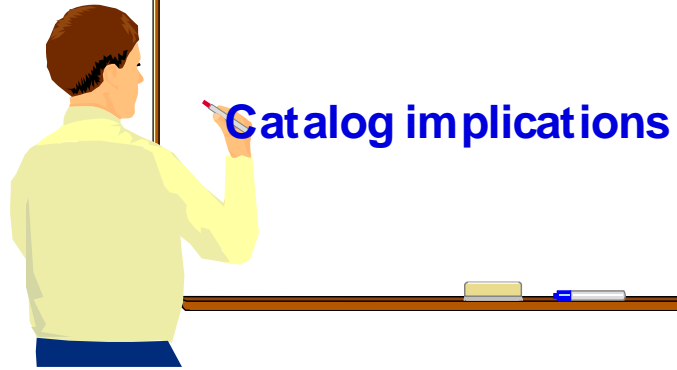
GoFurther

Catalog changes

- SYSTABLESPACE
  - ▸ INSTANCE   SMALLINT NOT NULL WITH DEFAULT 1
  - ▸ CLONE   CHAR(1) NOT NULL WITH DEFAULT 'N'

- SYSCOPY
  - ▸ INSTANCE   SMALLINT NOT NULL WITH DEFAULT 1

  - ▸ The timestamp of a data exchange will be tracked in SYSCOPY

    - ▪ ICTYPE = 'A' and STYPE = 'E' records
    - ▪ SYSCOPY EXCHANGE row for the base objects and also the clone objects

GoFurther

The SYSTABLESPACE.INSTANCE column value will always be a '1' or a '2'. It will always reflect the instance number of the current base objects. The CLONE column indicates if there is there is currently cloning. If the clone objects are dropped when the base objects are represented by instance number 2 data sets then CLONE will be 'N' in this case and SYSTABLESPACE.INSTANCE will be a 2. This is OK. If/when a clone is created again in the future then the clone objects will be created using instance number 1 data sets.

The SYSCOPY catalog table keeps track of the EXCHANGEs that are done. There will be one SYSCOPY row for each base and clone object (ie each index will have a row, the base and clone table space will have a separate row, etc).

Catalog information

- SYSTABLES
  - ▶ There is a separate SYSTABLES row for the base and clone tables.
  - ▶ TYPE = 'C' for clone table
  - ▶ TBCREATOR and TBNAME columns note related/partner table
    - ▪ For base table row, TBCREATOR and TBNAME are for clone table
    - ▪ For clone table row, TBCREATOR and TBNAME are for base table
    - ▪ OBID column value will be same for base and clone table
- SYSCOLUMNS
  - ▶ Contains rows for both base and clone tables
- SYSVIEWS
- SYSTABAUTH
- Rest of catalog
  - ▶ Most information is 'shared' (not duplicated)
    - ▪ SYSINDEXES, SYSINDEXPART, SYSKEYS, SYSTABLEPART

Catalog information

- Object identifiers

  ▸ Page set id

    - High order bit (HOB) indicates instance number

      – '0' indicates instance number 1

      – '1' indicates instance number 2

    - Catalog won't ever reflect a value where HOB = '1'B

EXCHANGE effect on Catalog

- SYSTABLESPACE
  - INSTANCE   reflects the base table/object instance number (1 or 2) and will 'toggle' on an EXCHANGE (1 -> 2, 2 -> 1)
  - CLONE   remains 'Y'

- SYSCOPY
  - INSTANCE   There will be one SYSCOPY EXCHANGE row written (ICTYPE = 'A' and STYPE = 'E' record) for each instance (1 and 2). Does not indicate if it was base or clone object at the time of EXCHANGE.

GoFurther

31

Commands and Utilities

GoFurther

DISPLAY DATABASE with cloning

```
-DISPLAY DATABASE(MYDB*) SPACENAM(*) LIMIT(*)
DSNT360I ( ********************************
DSNT361I ( * DISPLAY DATABASE SUMMARY
. . .
NAME     TYPE     PART     STATUS                    . . .
-------- ----     ----     ------------------   . . .
CHKOUT2  TSB1     001      RW
CHKOUT2  TSB1     002      RW
. . .
XDEPT1   IXB1     001      RECP
XDEPT1   IXB1     002      RW
. . .
CHKOUT2  TSC2     001      RO
CHKOUT2  TSC2     002      RW
. . .
XDEPT1   IXC2     001      ICOPY
XDEPT1   IXC2     002      RO

. . .
CHKOUT3  TS       001      RO
. . .
******* DISPLAY OF DATABASE MYDBX ENDED **********************
```

base objects

clone objects

objects not involved in cloning

A 2-character type suffix is added to the end of these existing types as follows:

•The first suffix character will be either a 'B' to indicate a base object or a 'C' to indicate a clone object.

•The second suffix character will be the associated data set instance number. This will be either a '1' for data sete instance number 1, or a '2' for data set instance number 2.

Here we have some base and clone objects. Note that the objects have different states for different partitions.

EXCHANGE affect on DISPLAY DATABASE

►►— **EXCHANGE DATA BETWEEN** *table-name-1* ——————►

——————————————— **AND** *table-name-2* ——►◄

GoFurther

The base or clone table name can be specified first.

DISPLAY DATABASE after EXCHANGE

```
-DISPLAY DATABASE(MYDB*) SPACENAM(*) LIMIT(*)
DSNT360I ( *********************************
DSNT361I ( * DISPLAY DATABASE SUMMARY
. . .
NAME      TYPE      PART      STATUS                    . . .
--------  ----      ----      -----------------  . . .
CHKOUT2   TSB2      001       RO
CHKOUT2   TSB2      002       RW                        base objects
. . .
XDEPT1    IXB2      001       ICOPY
XDEPT1    IXB2      002       RO
. . .
CHKOUT2   TSC1      001       RW
CHKOUT2   TSC1      002       RW                        clone objects
. . .

XDEPT1    IXC1      001       RECP
XDEPT1    IXC1      002       RW
. . .
CHKOUT3   TS        001       RO                  objects not involved in cloning
. . .
******* DISPLAY OF DATABASE MYDBX ENDED **********************
. . .
```

GoFurther

After an EXCHANGE the instance numbers changed and the base objects are now represented by instance number 2 data sets. ***Note that the states followed the instance numbers!*** This means that if, for example, you have an ICOPY state on all the clone partitions (instance number 2 data sets) and RW on all the base object partitions……and you do an EXCHANGE…. the base objects would now 'inherit' the ICOPY states and the clone objects would then 'inherit' the RW states.

35

DISPLAY DATABASE after DROP of clone

```
-DISPLAY DATABASE(MYDB*) SPACENAM(*) LIMIT(*)
DSNT360I ( ********************************
DSNT361I ( * DISPLAY DATABASE SUMMARY
. . .
NAME      TYPE      PART      STATUS                        . . .
--------  ----      ----      -----------------  . . .
CHKOUT2   TSB2      001       RO
CHKOUT2   TSB2      002       RW
. . .
XDEPT1    IXB2      001       ICOPY
XDEPT1    IXB2      002       RO
. . .
CHKOUT3   TS        001       RO
. . .
******* DISPLAY OF DATABASE MYDBX ENDED *********************
. . .
```

NOTE: If clone is dropped when the base objects are represented by instance number 2
(odd number of EXCHANGES done), DISPLAY will continue to display the 'B2'
type suffix even though there is no clone. When new clone is created it will be
given instance number 1 data sets.

If the clone table is dropped when the base objects are represented by instance number 2 data sets, a DISPLAY DATABASE will still have the 'B2' suffix. This indicates that the base objects are represented by instance number 2 data sets.

Commands

- STOP/START DATABASE gain a CLONE keyword

  ‣ Stop/Start by default processes only base objects, unless CLONE specified then just CLONEd objects.

- DISPLAY DATABASE

  ‣ No CLONE keyword. Displays both base and clone objects by default.

GoFurther

Stand-alone utilities

- DSN1LOGP

  ▶ DSN1LOGP can print the log records for both base and clone table objects. The base and clone objects are differentiated by differences in the OBID/PSID value.

- DSN1COPY

  ▶ DSN1COPY can operate on both base and clone table objects. The base and clone objects are differentiated by differences in the OBID/PSID value.

- DSN1PRNT

  ▶ DSN1PRNT operates on a named (SYSUT1) data set. Use data set name of base or clone object.
    - *catname.DSNDBx.MYDB1.TSPACE1.I0001A001*
    - *catname.DSNDBx.MYDB1.TSPACE1.I0002.A001*

GoFurther

38

Utilities with changes

- CHECK DATA
- CHECK INDEX
- CHECK LOB
- COPY
- COPYTOCOPY
- DIAGNOSE
- LISTDEF
- MERGECOPY
- MODIFY RECOVERY

- QUIESCE
- REBUILD INDEX
- RECOVER
- REORG INDEX
- REORG TABLESPACE
- REPAIR
- UNLOAD

GoFurther

Utilities notes

- Utilities will only ever process base or clone objects, never both.

- In the absence of a CLONE option or keyword the utilities will operate against base objects only by default

- LISTDEF can have INCLUDEs and EXCLUDEs for both base and clone objects but when the final list is used with a utility the utility will only process base or clone objects.

  ‣ Example: If your LISTDEF result has 100 objects in it and only 5 of them have clones then when you use this list with the COPY utility with no CLONE keyword then 100 base objects will be copied. If you use the same list with COPY and specify the CLONE keyword then only the 5 clone objects will be copied.

GoFurther

Utilities 1

- CHECK DATA *table-space-spec* . . . CLONE . . .
  - ▸ Indicates that CHECK DATA is to check the clone table in the specified table space. Because clone tables cannot have referential constraints, the utility checks only table check constraints and for consistencies between the clone table data and the corresponding LOB data. If you do not specify CLONE, CHECK DATA operates on the base table.

- CHECK INDEX . . . CLONE
  - ▸ Indicates that CHECK INDEX is to check only the specified indexes that are on table clones.

- CHECK LOB TABLESPACE *lob-table-space-spec* . . . CLONE
  - ▸ Indicates that CHECK LOB is to check the LOB table space data for only the table clone, note the LOB data for the base table.

GoFurther

Utilities 2

- COPY . . . CLONE . . .
  - ▸ Indicates that COPY is to copy only table clone data in the specified table spaces or indexes on table clones in the specified index spaces. If you use the LIST keyword to specify a list of objects, COPY processes only those objects in the list that contain table clones or indexes on table clones. The other objects in the list are ignored.

- COPYTOCOPY . . . CLONE . . .
  - ▸ Indicates that COPYTOCOPY is to process only image copy data sets that were taken against table clones or indexes on table clones.

- DIAGNOSE . . . CLONE . . .
  - ▸ Indicates that DIAGNOSE is to display information for only the specified objects that are table clones, table spaces that contain table clones, indexes on table clones, or index spaces that contain indexes on table clones.

GoFurther

Utilities 3

- LISTDEF
  - CLONE
    - Indicates that the INCLUDE or EXCLUDE expression is to return only clone tables, table spaces that contain clone tables, indexes on table clones, or index spaces that contain indexes on clone tables. If you specify ALL, LOB objects are also included. The CLONE keyword is ignored if you also specify a table name.
  - ALL
    - Specifies that both BASE and LOB objects are to be included in the list. Auxiliary relationships are to be followed from all objects that result from the initial object lookup, and both BASE and LOB objects are to remain in the final enumerated list. Clone objects are not included.
  - TABLE *creator-id.table-name*
    - Specifies the table that is to be used for the initial search for the object. . . . If you specify a table name with CLONE, the CLONE keyword is ignored.

Utilities 4

- MERGECOPY . . . CLONE . . .
  - ▸ Indicates that MERGECOPY is to process only image copy data sets that were taken against clone objects.

- MODIFY RECOVERY . . . CLONE . . .
  - ▸ Indicates that MODIFY RECOVERY is to delete SYSCOPY records and any related SYSLGRNX records for only clone objects. DBD entries for clone tables are not deleted.

- QUIESCE . . . CLONE . . .
  - ▸ Indicates that QUIESCE is to create a quiesce point for only the specified table spaces that contain clone tables.

GoFurther

44

Utilities 5

- REBUILD INDEX . . . CLONE . . .
  - ▶ ALL
    - Specifies that all indexes in the table space that is referred to by the TABLESPACE keyword are to be rebuilt. If you specify ALL, indexes on table clones are not included; only indexes on the base table are included.
  - ▶ CLONE
    - Indicates that both REBUILD INDEX is to reconstruct only the specified indexes that are on table clones. If you specify CLONE, you cannot specify STATISTICS. Statistics are not collected for table clones.

- RECOVER . . . CLONE . . .

  There are no special recovery considerations for a cloned table space or cloned index. Cannot do PIT recovery to a time that precedes that last exchange.
  - ▶ CLONE
    - Indicates that RECOVER is to recover only table clone data in the specified table spaces or the specified index spaces that contain index on table clones.

45

Utilities 6

- REORG INDEX . . . CLONE . . .
  - ▸ Indicates that REORG INDEX is to reorganize only the specified index spaces that contain indexes on table clones.

- REORG TABLESPACE . . . CLONE . . .
  - ▸ Indicates that REORG TABLESPACE is to reorganize only table clones from the specified table spaces. Base tables in those table spaces are not reorganized. If you specify CLONE, you cannot specify STATISTICS. Statistics are not collected for table clones.

Note: FASTSWITCH cannot be specified for REORG INDEX and REORG TABLESPACE utilities on any object involved with cloning (base table, clone table, indexes)!

GoFurther

Utilities 7

- REPAIR . . . CLONE
  - ▶ Indicates that REPAIR is to process only the specified objects that are table spaces that contain table clones, indexes on table clones, or index spaces that contain indexes on table clones. If you specify CLONE, you cannot specify VERSIONS, because table clones to not have versions. **Clones cannot be created for tables with active versions.**

    If you specify SET with CLONE, the status is changed for only the specified clone tables and their indexes. The CLONE keyword applies to all SET statements and LOCATE statements within the same REPAIR utility control statement.

- UNLOAD . . . CLONE . . .
  - ▶ Indicates that UNLOAD is to unload data from only clone tables in the specified table spaces. If you specify the name of the clone table in the FROM TABLE clause, you do not need to specify the CLONE keyword.

GoFurther

Utilities with clone implications

- MODIFY STATISTICS

- REPORT

- RUNSTATS

GoFurther

Utilities 8

- MODIFY STATISTICS . . . CLONE . . .
  - ▸ The online MODIFY STATISTICS utility deletes unwanted statistics history records from the corresponding catalog tables. You can remove statistics history records that were written before a specific date, or you can remove records of a specific age. You can delete records for an entire table space, index space, or index.

    Run MODIFY STATISTICS regularly to clear outdated information from the statistics history catalog tables. By deleting outdated information from those tables, you can improve performance for processes that access data from those tables.

  Restriction: MODIFY STATISTICS does not process table clones, because statistics are not collected for these tables.

GoFurther

Utilities 9

- REPORT
  - ▶ The report provides information for both base and clone objects. If you specify TABLESPACESET, REPORT also processes related LOBs.

- RUNSTATS
  - ▶ No statistics are kept for a clone table so CLONE cannot be specified.

GoFurther

Summary

- Clone tables gives us an Online Load Replace type of capability.

- Lots of capability

- Online schema restrictions

- Future?

GoFurther

Session: C03

Session Title : Attack of the DB2 for z/OS Clones, Clone
 Tables That Is!

John Lyle

IBM Silicon Valley Laboratory

jlyle@us.ibm.com

GoFurther

Questions?