DB2 Data Management Software

IBM

e business software

Chapter 10: Maintaining Data

IBM DB2 Universal Database V8.1
Database Administration Certification Preparation Course

Maintained by Clara Liu

IBM Software Group

Objectives

- In this section, we will cover:
  - DB2 Export
  - DB2 Import
  - DB2 Load
  - Tools to move data
    - DB2MOVE, DB2LOOK
  - Tools to maintain data
    - REORGCHK, REORG, RUNSTATS, and REBIND
    - INSPECT

DB2 Data Management Software

IBM

e business software

Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

IBM Software Group

Export, Import, Load Utility File Formats

- Types of data
  - Non-Delimited or Fixed Length ASCII (ASC)
  - Delimited ASCII (DEL)
  - PC version of the Integrated Exchange Format (IXF)
    - PC/IXF is a structured description of a database table
  - Worksheet Format (WSF)
    - For Export and Import only
  - Cursor
    - A cursor declared against an SQL query
    - For LOAD only

Export Utility

- Exports data from database table(s) to file using an SQL SELECT statement
- Format of exported data can be IXF, WSF, or DEL
- If exporting to IXF or WSF files, new column names can be specified
- Include the MESSAGES option to record error, warning, and informational messages received from the export
- Only PC/IXF file format supported by DB2 Connect
- Must have SYSADM or DBADM authority, or CONTROL or SELECT privilege on table(s) being accessed in the export command
- Example:
  - CONNECT TO sample ;
  - EXPORT TO myfile
    OF IXF MESSAGES msg
    SELECT staff.name, staff.dept, org.location
    FROM org, staff
    WHERE org.deptnum = staff.dept ;

Export Utility

- Additional options to customize the export operation
  - EXPORT TO *filename* OF *filetype*
    - MODIFIED BY *filetype-mod*
    - MESSAGES *message-file*
    - *select-statement*
- Example of file type modifiers
  - chardelx - specify x, a single character string delimiter, default is a double quotation mark (")
  - coldelx - specify x, a single character column delimiter, default is a comma (,)
  - codepage=x - specify x, an ASCII character string, the code page of the data in the output file
- Example:
  - EXPORT TO myfile.del OF del
    MODIFIED BY chardel" coldel!
    SELECT * FROM staff

Exporting Large Objects

- By default, the first 32 KB of LOB data is exported and place in the same file as the rest of the column data
  - If LOB greater than 32 KB, it is truncated
- To store multiple LOBs in a single file, use the LOBSINFILE file modifier
- A LOB Location Specifier (LLS) is a string indicating where LOB data can be found within a file
- The format of the LLS is filename.ext.nnn.mmm/
  - filename.ext is the name of the file that contains the LOB
  - nnn is the offset of the LOB within the file (measured in bytes)
  - mmm is the length of the LOB (in bytes)
  - For example, an LLS of db2exp.001.123.456/ indicates:
    - LOB is located in file db2exp.001
    - Begins at an offset of 123 bytes of the file
    - 456 bytes long
  - If the indicated size in the LLS is 0, the LOB is considered to have a length of 0
  - If the length is -1, the LOB is considered to be NULL and the offset and file name are ignored

Exporting Large Objects - Example

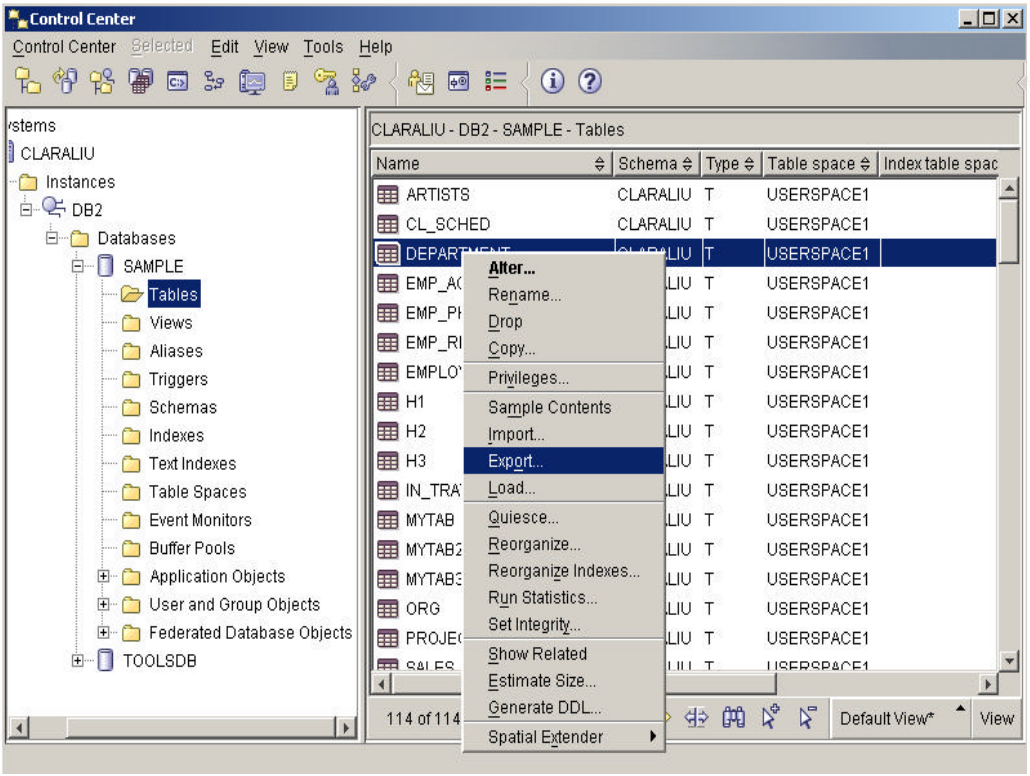
- LOBSINFILE has to be specified before other parameters related to exporting LOBs are considered
- Example:
  - CONNECT TO sample
  - EXPORT TO empresume.del of DEL
    LOBS TO d:\lob1\
    LOBFILE resume MODIFIED BY LOBSINFILE
    SELECT * FROM emp_resume ;
- Result:

File: empresume.del

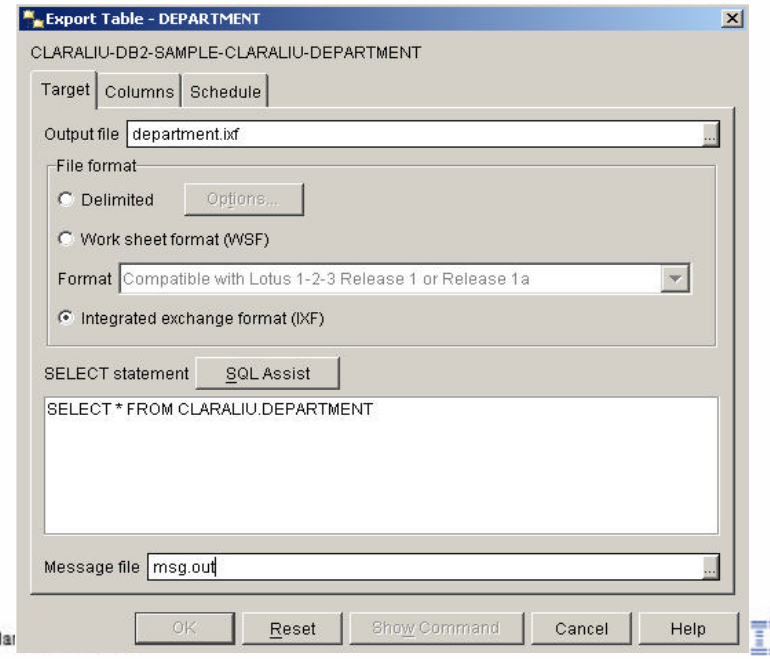
```
"000130","ascii","res.001.0.1313/"
"000130","script","res.001.1313.1817/"
"000140","ascii","res.001.3130.1316/"
"000140","script","res.001.4446.1878/"
"000150","ascii","res.001.6324.1363/"
"000150","script","res.001.7687.1923/"
"000190","ascii","res.001.9610.1292/"
"000190","script","res.001.10902.1852/"
```

Directory: d:\lob1

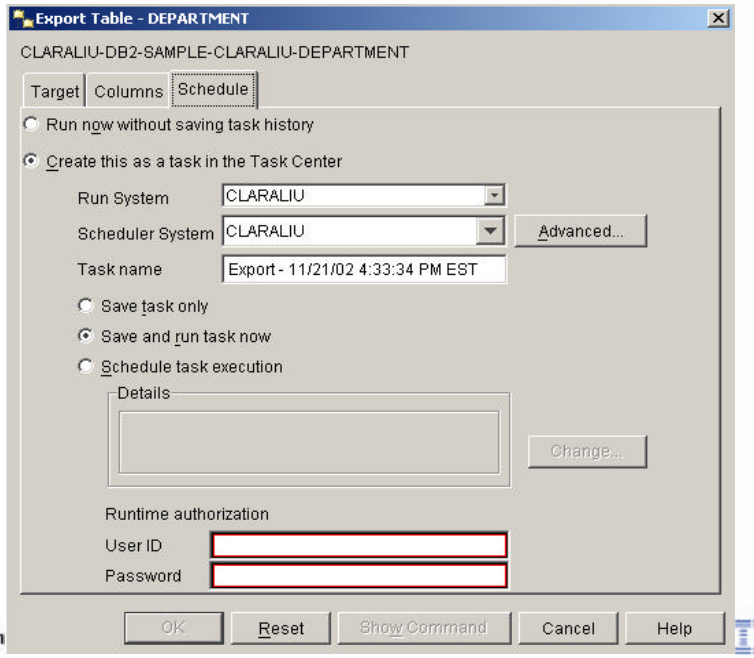
```
res.001
   - contains 8 LOB data
```

Export Using the Control Center



Export Using the Control Center



DB2 Data Man

IBM

DB2 Data Management Software

IBM

e business software

Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

IBM Software Group

Import Utility

- Imports data from a file to a database table
- The format of the input file can be IXF, WSF, DEL, or ASC
- Only PC/IXF file format supported by DB2 Connect
- Include the MESSAGES option to record error, warning, and informational messages received from the export
- By default, import only commit once at the end of the operation
- Target can be a table, a table with DATALINK columns, a typed table, or a view
- Target cannot be a system table, a declared temporary table or a summary table
- Must have SYSADM or DBADM authority, or underlying privileges (SELECT, INSERT, CONTROL, or CREATETAB) on the target table
- Example:
  - CONNECT TO sample ;
  - IMPORT FROM myfile.ixf OF IXF
    - MESSAGES msg.out **create** / **insert** / **insert_update** / **replace** / **replace_create** NTO newtab ;

Import Utility

- INSERT option
  - Adds the imported data to the table without changing the existing table data
  - Target table must exist
- INSERT_UPDATE option
  - Adds rows of imported data to the target table, or updates existing rows of the target table with matching primary keys
  - Target table must exist
- REPLACE options
  - Deletes all existing data from the table and inserts the imported data
  - Target table must exist, table and index definitions are not changed
- REPLACE_CREATE option
  - If the target table exists, deletes all existing data from the table by truncating the data object, and inserts the imported data without changing the table definition or the index definitions
  - If the table does not exist, creates the table and index definitions
  - Can only be used with IXF files
  - Cannot be used if the table has a primary key reference by another foreign key
- CREATE INTO option
  - The target table and indexes will be created
  - Can be used with PC/IXF input file format only
  - Can also specify the tablespace clause

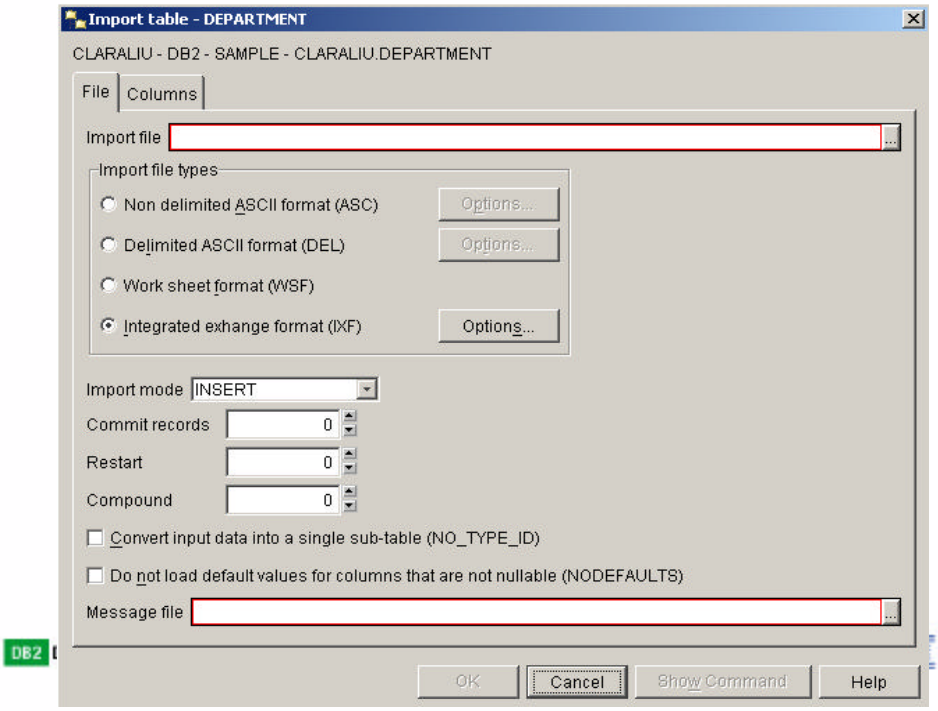
Import Utility

- Import is logged
  - Need enough primary and secondary logs and log space
- COMMITCOUNT option
  - Forces intermediate commits
- RESTARTCOUNT option
  - Permits restart of import after failure
- Indexes updated during import
- Referential integrity checked during import
- Use the MODIFIED BY option to customize the import operation
  - compound=x
    - x is a number between 1 and 100 inclusive
    - Uses nonatomic compound SQL to insert the data, and x statements will be attempted each time
  - column names
    - Specify into which columns data should go
- Example:
  - IMPORT FROM myfile OF ixf
    MODIFIED BY compound=5 COMMITCOUNT 100 MESSAGES msg.out
    INSERT INTO newtab ;

Importing LOBs From Files

- Example:
  - IMPORT FROM ascfile1 OF ASC
    LOBS FROM /u/db2load/lob1, /u/db2load/lob2
    MODIFIED BY lobsinfile
    INSERT INTO table1
- lobsinfile modifier specifies the path of the files that contain LOB data
- The files are located in the directories specified in the LOBS FROM clause

Import Using the Control Center



DB2 Data Management Software

IBM

e business software

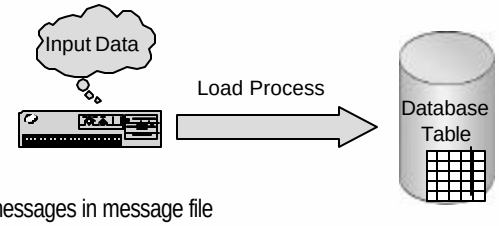
Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

IBM Software Group

Load Utility

- The load process has four phases
  - Load
    - Loads data into tables
    - Collects index keys and table statistics
    - Records consistency points
    - Places invalid data rows in dump file and messages in message file
  - Build
    - Create indexes based on the keys collected during the load phase
  - Delete
    - Delete rows that caused unique key violation and place them in the exception table
    - Record message in the message file
    - Deletion is logged; if there is a large number of unique key violated records, db2 log files could fill up
  - Index Copy
    - Index data is copied from a system temporary table space to the original table space
    - This will only occur if a system temporary table space was specified for index creation during load with ALLOW READ ACCESS specified



Load Utility

- Load data into a target table
- The input can be a file, named pipe, or a device
- The format of the input source can be DEL, ASC, PC/IXF, or CURSOR
- Example of CURSOR as the input source:
  - DECLARE mycurs CURSOR FOR SELECT col1, col2, col3 FROM tab1 ;
  - LOAD FROM mycurs OF CURSOR INSERT INTO tab2 ;
- Include the MESSAGES option to record error, warning, and informational messages received from the export
- Target must exist prior to load
- Target can be a table, a table with DATALINK columns, a typed table, or an alias
- Target cannot be a system table or a declared temporary table
- Authority and privilege required:
  - Must have SYSADM, DBADM, or LOAD authority on the database
  - INSERT privilege on the table when the load utility is invoked in INSERT mode
  - INSERT and DELETE privilege on the table when the load utility is invoked in REPLACE mode
  - INSERT privilege on the exception table if such a table is used as part of the load

Load Utility - Example

- Example:
  - LOAD FROM emp.ixf OF IXF
    MODIFIED BY dumpfile = /u/db2/load/rejectrow.out
    ROWCOUNT 10000 SAVECOUNT 1000 WARNINGCOUNT 100
    MESSAGES msg.out
    TEMPFILES PATH /tmp/db2load/
    INSERT / **REPLACE** / **RESTART** / **TERMINATE** INTO employee
    FOR EXCEPTION except_table
    ALLOW READ ACCESS USE TABLESPACE systemptbsp
    CHECK PENDING CASCADE DEFERRED
    LOCK WITH FORCE ;

Invoke the Load Utility in Different Mode

- INSERT mode
  - Adds the loaded data to the table without changing the existing table data
  - Possible to specifies the table column into which the data is to be inserted
- REPLACE mode
  - Deletes all existing data from the table, and inserts the loaded data
  - Table and index definitions are not changed
  - If an error occurs, the original data in the table is lost
  - This option is not supported for tables with DATALINK columns
- RESTART mode
  - Restarts a previously interrupted load
  - It will automatically continue from the last consistency point in the load, build, or delete phase
  - Specify the same parameters as in the previous load invocation, so that the utility can find the necessary temporary files
- TERMINATE mode
  - Terminates a load operation and rolls back the operation to the point in time at which it started, even if consistency points were passed
  - Table space state returns to normal, and all table objects are made consistent
  - If the load operation being terminated is a load REPLACE, the table will be truncated
  - If the load operation being terminated is a load INSERT, original recores are retained
  - This option is not supported for tables with DATALINK columns

Load Dump File and Load Temporary Files

- Load Dump File:
  - Rows that do not comply with the definition of the table are rejected
  - Use the **dumpfile** modifier to specify the name and location of a file to which rejected rows are written

filetype modifier

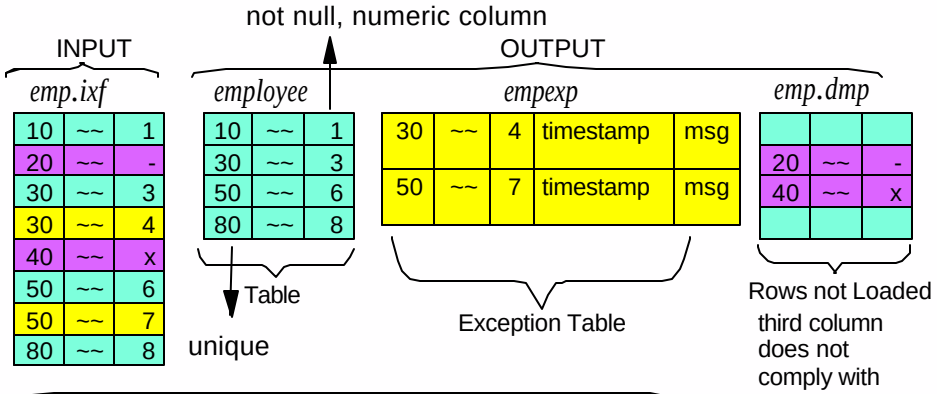
- Load Temporary Files:
  - DB2 creates temporary binary files during load processing
  - They are used for load crash recovery, load terminate operations, warning and error messages, and runtime control data
  - They are removed when the load operation completes without error
  - Use the **TEMPFILES PATH** option to specify the directory to store the temporary files

Load Exception Table

- Load exception table is specified by the FOR EXCEPTION clause
- It is a user-defined table used to store copies of rows that violate unique index rules
- Load does not check for constraints or foreign key violations other than violations of uniqueness
- It has to have the same definition of the table being loaded
  - If at least one of the columns is not present in exception table, offending row is discarded
- Two optional additional columns can be added to the end of the table
  - Timestemp column is a timestamp recording when row inserted
  - Message column is a CLOB(32K) or larger recording the associated error message
- No additional column is allowed in the exception table
- Must be free of any constraints and triggers

- Note: If an exception table is not specified and duplicate records are found, duplicate records are deleted and warning messages are generated

Load Utility - Example




```
LOAD FROM emp.ixf OF IXF MESSAGES emp.msg
    MODIFIED BY DUMPFILE=c:\emp.dmp
    TEMPFILES PATH d:\tmp
    INSERT INTO employee
    FOR EXCEPTION empexp
```

- Examine the message file emp.msg and the exception table empexp

ROWCOUNT, SAVECOUNT, WARNINGCOUNT

- ROWCOUNT n
  - Specifies the number of n physical records in the file to be loaded
  - Allows a user to load only the first n rows in a file
- SAVECOUNT n
  - Establish consistency points after every n rows
  - Messages are generated indicating how many input rows were successfully loaded at the time of the save point
  - n should be set to a sufficiently high value so that performance will not be impacted because a message is issued at each consistency point
  - Not supported with the CURSOR filetype
- WARNINGCOUNT n
  - Stops the load operation after n warnings
  - If load is stopped because n warnings are encountered, load can be started again with the RESTART or REPLACE mode

Indexing Mode

- Specifies whether the load utility is to rebuild indexes or to extend them incrementally
- AUTOSELECT
  - Load will automatically decide between REBUILD or INCREMENTAL mode
- REBUILD
  - All indexes will be rebuilt
- INCREMENTAL
  - Indexes will be extended with new data
- DEFERRED
  - Load will not attempt index creation
  - Indexes will be marked as needing a refresh
  - When the database is restarted or first access to such indexes may cause the index to be rebuilt

No Table Access During Load

- ALLOW NO ACCESS (default behavior)
  - Target table is locked for exclusive access
  - Table state is set to LOAD IN PROGRESS during the load
  - It is the only valid option for LOAD REPLACE
  - If there are constraints on the table, the table state is also set to CHECK PENDING
    - Use the SET INTEGRITY command to validate the data against the constraints

With Table Access During Load

- ALLOW READ ACCESS
  - Target table is locked in share mode
  - This option is not supported with REPLACE mode because existing data is truncated at the beginning the load
  - Readers may access the non-delta portion of the data while the table is being loaded
  - Data that is being loaded is not available until the load is complete
  - If load fails, the data that existed in the table prior to the load operation will continue to be available in read only mode after the failure
  - If there are constraints on the table, the table state is also set to CHECK PENDING
    - Use the SET INTEGRITY command to validate the new portion of the data (if possible) against the constraints
    - Once the SET INTEGRITY command is issued data that is being validated is not accessible, until the command is completed
  - Specify the USE TABLESPACE <tbsp-name> option
    - If a full index is being rebuilt, a shadow copy of the index is built in the specified table space and copied over to the original table space during the INDEX COPY PHASE
    - The target table is taken offline when the new indexes are copied into the target table space
    - Only system temporary table spaces can be used
    - If system temporary table space not specified, then the shadow index will be created in the same table space as the index object

LOAD Option - CHECK PENDING CASCADE

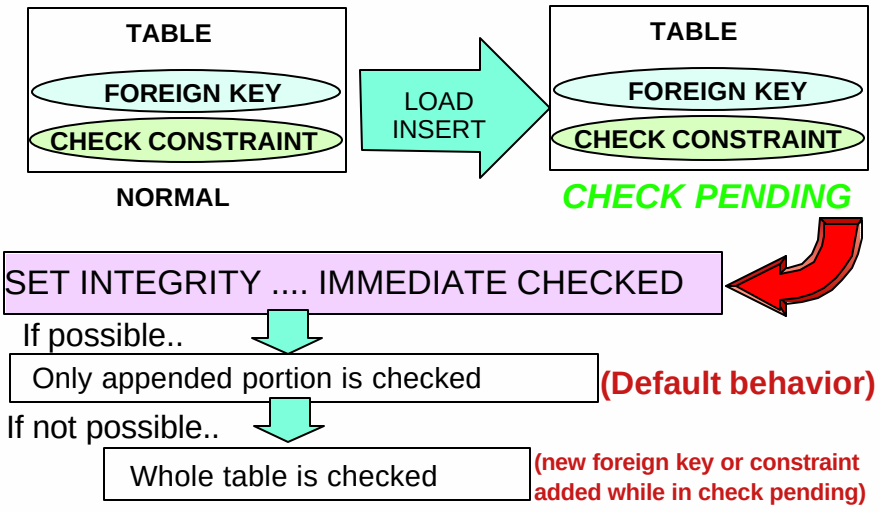
- If LOAD puts the table into a check pending state, this option allows the user to specify whether or not the check pending state of the loaded table is immediately cascaded to all descendents, for example descendent foreign key tables
- CHECK PENDING CASCADE IMMEDIATE
  - Indicates that the check pending state for foreign key constraints is immediately extended to all descendent foreign key tables
- CHECK PENDING CASCADE DEFERRED
  - Indicates that only the loaded table will be placed in the check pending state

SET INTEGRITY Command

- It is used to turn off integrity checking for one or more tables
- Integrity checking options: Foreign Key, Check, Materialzed Query, DATALINK, Generated Column, Staging
- SET INTEGRITY command options:
  - INCREMENTAL
    - Force incremental constraint checking on the appended portion of data or summary table refresh processing
    - Even if option not specified, incremental processing is default
    - Error message returned if full processing needed
  - IMMEDIATE CHECKED
    - Integrity checking turned on and carry out deferred integrity checking
  - IMMEDIATE UNCHECKED
    - Integrity checking turned on and deferred integrity checking is not performed
    - Can also check one type of integrity and leave in check pending state
  - ACCESS MODE
  - CASCADE MODE

SET INTEGRITY Command

- Maintain integrity incrementally



Check Pending Status

- CONST_CHECKED columns of SYSCAT.TABLES indicates status of each constraint defined in the table

Value	Description
Y	Checked by SYSTEM
N	Not checked (In CHECK PENDING)
U	Checked by USER (As a result of SET INTEGRITY ...IMMEDIATE UNCHECKED)
W	Previously checked by USER and some data needs to be verified by SYSTEM (In CHECK PENDING)
F	Byte 5 - materialized query table cannot be refreshed incrementally Byte 7 - content of staging table is incomplete and cannot be used for incremental refresh of the associated materialized query table

- Byte 1 - foreign key constraints
- Byte 2 - check constraints
- Byte 5 - materialized query table
- Byte 6 - generated columns
- Byte 7 - staging table

```
db2 "select tabname,status,CONST_CHECKED from SYSCAT.TABLES"
TABNAME     STATUS CONST_CHECKED
TABLE1      N      YYYYYYYYYYYYYYYYYYYYYYY
```


SET INTEGRITY - Examples

- Example 1:
  - Set tables T1 and T2 to check pending no access state, and immediately cascade the check pending state to their descendants.
  - SET INTEGRITY FOR T1, T2 OFF NO ACCESS CASCADE IMMEDIATE
- Example 2:
  - Set parent table T1 to check pending read state without immediately cascading the check pending state to its child table T2.
  - SET INTEGRITY FOR T1 OFF READ ACCESS CASCADE DEFERRED
- Example 3:
  - Check integrity for T1, and get the first violation only.
  - SET INTEGRITY FOR T1 IMMEDIATE CHECKED
- Example 4:
  - Check integrity for T1 and T2, and put the violating rows into exception tables E1 and E2.
  - SET INTEGRITY FOR T1, T2 IMMEDIATE CHECKED
    - FOR EXCEPTION IN T1 USE E1, IN T2 USE E2

LOAD Option - LOCK WITH FORCE

- Various locks (e.g. table locks) are required for loading
- LOCK WITH FORCE allows load to force off other applications that hold conflicting locks
- Authority requires is the same as to perform the FORCE APPLICATIONS command - SYSADM or SYSCTRL

MODIFIED BY Clause

- dumpfile
  - Specify file to store rejected rows
- fastparse
  - Reduce syntax checking on loaded data to enhance performance
- generatedignore, generatedmissing, generatedoverride
  - Ignore or override generated value, or indicate generated values are missing
- identityignore, identitymissing, identityoverride
  - Ignore or override identity column data, or indicate identity column data is missing
- indexfreespace, pagefreespace, totalfreespace
  - Leave specified amount of free space in index pages and data pages
- norowwarnings
  - Suppress row warnings
- lobsinfile
  - Specifies the path to the files containing LOB data

Loading LOBs From Files

- Example:
  - LOAD FROM ascfile1 OF ASC
    LOBS FROM /u/db2load/lob1, /u/db2load/lob2
    MODIFIED BY lobsinfile
    INSERT INTO table1
- lobsinfile modifier tells the loader that all LOB data is to be loaded from files
- The files are located in the directories specified in the LOBS FROM clause

Gathering Statistics During Load

- Specifies whether to gather statistics during the load process with STATISTICS YES/NO
- STATISTICS YES is supported in REPLACE mode only
  - If data is appended to a table, statistics are not collected, need to use RUNSTATS following completion of the load
- If STATISTICS YES, can also collect:
  - Distribution Statistics - option WITH DISTRIBUTION
  - Index Statistics - option INDEXES ALL

LOAD Option - COPY YES/NO

- Load almost completely eliminates the logging associated with the loading of data
- To allow database or table space to be recoverable after load in case of failure:
  - Take a backup of table space after the completion of the load
  - Explicitly request a copy of the loaded portion of table
- For database with forward log recovery enabled:
  - Table space in which the loading table resides is in LOAD IN PROGRESS state during the load
  - Table space is left in BACKUP PENDING state at the completion of the load regardless if it is successful or not
  - This is the behavior of COPY NO (default)
  - If COPY YES option is used, a copy of the loaded data will be saved in the specified location
- For database with forward log recovery disabled:
  - Table space in which the loading table resides is in LOAD IN PROGRESS state during the load
  - Table space will not be in BACKUP PENDING state
  - To ensure that the table space can be restored for any reason, the table space should be backed up before and after the load
  - Cannot use COPY YES option, it is not supported if forward log recovery is disabled

LOAD Option - NONRECOVERABLE

- Specifies that the load transaction is to be marked as non-recoverable
- Table spaces are not put in backup pending state following the load
- If a rollforward is performed, the rollforward utility will skip the transaction, and will mark the table into which data was being loaded as "invalid"
- If load fails, table must be dropped and recreated
- "All or nothing" - load works or start from scratch
- Good for initial load to empty table or read-only table

LOAD QUERY - Table States

- Checks the status of a load operation during processing and returns the table state
- If a load is not processing, then the table state alone is returned
- Example:
  - LOAD QUERY TABLE staff TO /u/mydir/staff.tempmsg
- Table states returned by the LOAD QUERY command are:
  - Normal
  - Check Pending
    - The table has constraints and they have yet to be verified.
  - Load in Progress
  - Load Pending
    - A load has been aborted. Issue a load terminate, a load restart or a load replace to bring the table out of the Load Pending state.
  - Read Access Only
    - The table data is available for read access queries.
  - Unavailable
  - Not Load Restartable
    - The table is in a partially loaded state that will not allow a load restart.
  - Unknown

Remote Load Support

- The CLIENT option specifies that the data to be loaded resides on a remotely connected client
- It is ignored if the load operation is not being invoked from a remote client
- Example:
  - LOAD CLIENT FROM /u/user/data.del OF DEL
    MODIFIED BY CODEPAGE=850
    INSERT INTO mytable

IMPORT vs LOAD

IMPORT	LOAD
Slower on large amounts of data	Faster on large loads - writes formatted pages
Creation of tables & indexes with IXF format	Tables and indexes must exist
WSF supported	WSF not supported
Import into tables and views (Aliases supported)	Load tables only (Aliases supported)
No support for importing into materialized query tables	Support for loading into materialized query tables
Table space(s) On-line during import	Table space(s) Off-line during load
All rows logged	Minimal logging performed
Triggers will be fired	Triggers not supported

IMPORT vs LOAD

IMPORT	LOAD
Temporary space used within the database. Largest index plus 10% (approx)	Temporary space used outside the database. Sum of all indexes (approx)
Constraints validated during import	All Unique key is verified during load Other constraints are validated with the SET INTEGRITY command
If interrupted table is usable with data up to the last commit point.	If interrupted the table is held in LOAD PENDING state. Either restart or restore tables effected.
Run RUNSTATS after import for Statistics	Statistics gathered during Load
Import into mainframe database via DB2 Connect	Cannot load into mainframe database
Files must reside on the same node as the import	Files/Pipes must reside on the database node
No back-up image required	Backup can be created during load

DB2 Data Management Software

IBM

e business software

Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

IBM Software Group

Database Movement Tool Command - db2move

- db2move can be used to move large numbers of tables between DB2 databases
- It queries the system catalog tables, compiles a list of all user tables, and exports these tables in PC/IXF format
- The PC/IXF files can be imported or loaded to another DB2 database on the same and different platform
- db2move <dbname> <action> <options>
  - Action can be EXPORT, IMPORT, or LOAD
- Examples to illustrate some of the options:
  - **db2move sample export -tc userid1,us*rid2 -tn tbname1,*tbname2**
    - Export all tables created by "userid1" or user IDs LIKE "us%rid2", and with the name "tbname1" or table names LIKE "%tbname2"
  - **db2move sample import -l D:\LOBPATH1,C:\LOBPATH2**
    - Import all tables in the SAMPLE database
    - LOB paths "D:\LOBPATH1" and "C:\LOBPATH2" are to be searched for LOB files.
  - **db2move sample import -io replace -u userid -p password**
    - Import all tables in the SAMPLE database in REPLACE mode
    - The specified user ID and password will be used

DB2 Statistics and DDL Extraction Tool Command - db2look

- Tool can:
  - Extract DDL statements to reproduce database objects
  - Generate UPDATE statements to replicate statistics on the objects
  - Generate UDDATE statements to update database configuration database manager configuration parameters
  - Generate db2set statements so that the registry variables and configuration parameter settings
  - Extract statistics or report on statistics
- Handy tool to match catalog statistics of a test database to a production database
- Utility can be executed from command prompt or Control Center
- Example:
  - **db2look -d department -u walid -e -o db2look.sql**
  - Generate DDL statements for objects created by user walid in database DEPARTMENT
  - The db2look output is sent to file db2look.sql

DB2 Data Management Software

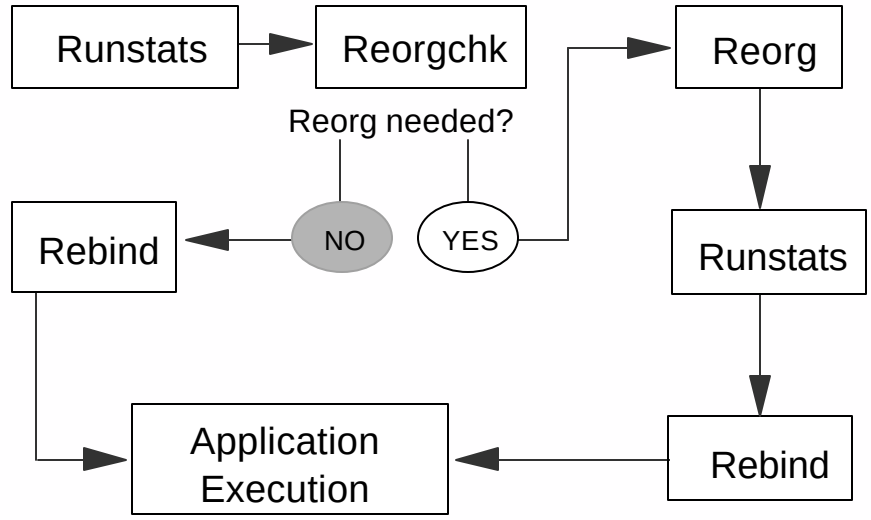
IBM

e business software

Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

Data Maintenance Process



RUNSTATS Utility

- Updates statistics about the physical characteristics of a table and the associated indexes
- Characteristics include number of records, number of pages, and average record length
- Examples:
  - **RUNSTATS ON TABLE db2user.employee ALLOW WRITE ACCESS**
    - Collect statistics on the table only, on all columns without distribution statistics
    - Other users can read from and write to the table while statistics are calculated
  - **RUNSTATS ON TABLE db2user.employee WITH DISTRIBUTION ON COLUMNS (empid, empname) ALLOW READ ACCESS**
    - Collect statistics on the table only, on columns empid and empname with distribution statistics
    - Other users can have read-only access to the table while statistics are calculated
  - **RUNSTATS ON TABLE db2user.employee AND SAMPLED DETAILED INDEXES ALL**
    - Collect basic statistics on the table and all indexes using sampling for the detailed index statistics collection
- Statistics of an index can also be collected at index creation time
  - **CREATE INDEX item ON stock (itemno) COLLECT DETAILED STATISTICS**

FLUSH PACKAGE CACHE Statement

- Dynamic SQL statements are cached in the package cache
- When should this cache refreshed?
  - Most normal activities that affect the validity of cached dynamic SQL statements are already handled by DB2 by invalidating the affected cached entries, certain activities such as online update of DBM and DB configuration parameters
  - After RUNSTATS, the dynamic SQL statements should be prepared again so that the latest database statistics is used
- Use the FLUSH PACKAGE CACHE statement to remove all cached dynamic SQL statements currently in the package cache
- Cached dynamic SQL statements that are not in use are invalidated
- Any cached dynamic SQL statement currently in use will continue be used until it is no longer needed by the its current user; the next new user of the same statement will force an implicit prepare of the statement by DB2, and the new user will execute the new version of the cached dynamic SQL statement
- Example:
  - FLUSH PACKAGE CACHE

REORGCHK Utility

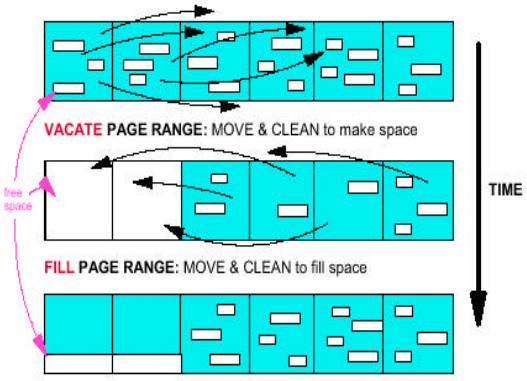
- Calculates statistics on the database to determine if tables or indexes, or both, need to be reorganized or cleaned up
- Examples:
  - **REORGCHK CURRENT STATISTICS ON TABLE USER**
    - Checks the tables that are owned by the run time authorization ID
  - **REORGCHK UPDATE STATISTICS ON SCHEMA smith**
    - Checks all the tables created under the specified schema
  - **REORGCHK UPDATE STATISTICS ON TABLE SYSTEM**
    - Checks all system tables
  - **REORGCHK UPDATE STATISTICS ON TABLE ALL**
    - Checks all users and system tables

REORGCHK Sample Output

```
Table statistics:

F1: 100 * OVERFLOW / CARD < 5
F2: 100 * (Effective Space Utilization of Data Pages) > 68
F3: 100 * (Required Pages / Total Pages) > 80

SCHEMA    NAME                  CARD    OV    NP    FP    TSIZE  F1  F2  F3 REORG
------------------------------------------------------------------------------
SYSIBM    SYSATTRIBUTES            -     -     -     -        -   -   -   - ---
SYSIBM    SYSBUFFERPOOLNODES       -     -     -     -        -   -   -   - ---
SYSIBM    SYSBUFFERPOOLS           1     0     1     1       52   0   - 100 ---
SYSIBM    SYSCHECKS                -     -     -     -        -   -   -   - ---
SYSIBM    SYSCODEPROPERTIES        -     -     -     -        -   -   -   - ---
SYSIBM    SYSCOLAUTH               -     -     -     -        -   -   -   - ---


Index statistics:

F4: CLUSTERRATIO or normalized CLUSTERFACTOR > 80
F5: 100*(KEYS*(ISIZE+9)+(CARD-KEYS)*5) / ((NLEAF-NUM_EMPTY_LEAFS)*INDEXPAGESIZE) > 50
F6: (100-PCTFREE)*((INDEXPAGESIZE-96)/(ISIZE+12))**(NLEVELS-2)*(INDEXPAGESIZE-96)/
(KEYS*(ISIZE+9)+(CARD-KEYS)*5) < 100
F7: 100 * (NUMRIDS DELETED / (NUMRIDS DELETED + CARD)) < 20
F8: 100 * (NUM EMPTY LEAFS / NLEAF) < 20

SCHEMA   NAME                CARD  LEAF ELEAF LVLS ISIZE NDEL KEYS  F4 F5 F6 F7 F8 REORG
-------------------------------------------------------------------------------------
Table: SYSIBM.SYSATTRIBUTES
SYSIBM   IBM83                  -     -     -    -     -    -    -   -  -  -  -  - -----
SYSIBM   IBM84                  -     -     -    -     -    -    -   -  -  -  -  - -----
SYSIBM   IBM85                  -     -     -    -     -    -    -   -  -  -  -  - -----
Table: SYSIBM.SYSBUFFERPOOLNODES
SYSIBM   IBM69                  -     -     -    -     -    -    -   -  -  -  -  - -----
Table: SYSIBM.SYSBUFFERPOOLS
SYSIBM   IBM67                  1     1     0    1    22    0    1 100  -  -  0  0 -----
SYSIBM   IBM68                  1     1     0    1    10    0    1 100  -  -  0  0 -----
Table: SYSIBM.SYSCHECKS
SYSIBM   IBM37                  -     -     -    -     -    -    -   -  -  -  -  - -----
```


REORG Utility

- Online and Offline table reorganization are supported in DB2 V8
- Offline REORG
  - The table is by default available for read only (ALLOW READ ACCESS option)
  - Can also specify no access allowed during table reorganization (ALLOW NO ACCESS option)
  - Long and LOB data stored in large table space is no longer reorganized by default
    - Specify LONGLOBDATA to reorganize this data
- Online (or In-place) REORG
  - Takes longer time to complete
    - In-place reorganization
      defers to concurrent applications
    - Long running statements
      can slow the REORG progress
  - Requires more log space to recover
    unexpected failure
  - Can be paused and resumed later



REORG Utility

```
>>-REORG-----------------------------------------------------------><

>--+-TABLE--table-name--| Table Clause |-----------------+------><
   '-INDEXES ALL FOR TABLE--table-name--| Index Clause |-'


Table Clause

|--+-------------------+---------------------------------------><
   '-INDEX--index-name-'

     .-ALLOW READ ACCESS-.
>--+-+-------------------+--+-------------+--+-----------+--+-------------+-+--|
   | '-ALLOW NO ACCESS---'  '-USE--tbspace-'  '-INDEXSCAN-'  '-LONGLOBDATA-' |
   |           .-ALLOW WRITE ACCESS-.                          .-START--.    |
   '-INPLACE--+-+-------------------+--+------------------+--+--------+-+---'
              | '-ALLOW READ ACCESS--'  '-NOTRUNCATE TABLE-'  '-RESUME-' |
              '-+-STOP--+-------------------------------------------------'
                '-PAUSE-'

Index Clause

   .-ALLOW READ ACCESS--------.
|--+--------------------------+--+-------------------------+----|
   '-ALLOW--+-NO----+--ACCESS-'  |               .-ALL---. |
            '-WRITE-'            +-CLEANUP ONLY--+-------+-+
                                 |               '-PAGES-' |
                                 '-CONVERT-----------------'
```


REORG Utility

- Table Options
  - INDEX index-name
  - ALLOW READ / WRITE / NO ACCESS
  - USE tablespace-name
    - Specifies the system temporary table space temporary copy of the table being reorganized
  - INDEXSCAN
    - For a clustering REORG an index scan will be used to re-order table records
  - LONGLOBDATA
    - Long field and LOB data are to be reorganized
  - INPLACE
    - Reorganize the table while permitting user access
    - Allowed only on tables with type-2 indexes
  - NOTRUNCATE TABLE
    - Do not truncate the table after inplace reorganization
  - START
    - Start the inplace REORG processing (default)
  - STOP
    - Stop the inplace REORG processing at its current point
  - PAUSE
    - Suspend or pause inplace REORG for the time being
  - RESUME
    - Continue or resume a previously paused inplace table reorganization

REORG Utility

- Index Options
  - ALLOW READ / WRITE / NO ACCESS
    - Use the ALLOW READ ACCESS or ALLOW WRITE ACCESS option to allow other transactions either read-only or read-write access to the table while the indexes are being reorganized
    - During the period in which the reorganized copies of the indexes are made available, no access to the table is allowed
  - CLEANUP ONLY
    - A cleanup rather than a full reorganization will be done
    - The indexes will not be rebuilt and any pages freed up will be available for reuse by indexes defined on this table only
  - CONVERT
    - Convert indexes to type-2 indexes
    - If the index is type 1, this option will convert it into type 2
    - If the index is already type 2, this option has no effect

REBIND Utility

- Allows the user to recreate a package stored in the database without the need for a bind file
  - Must use qualified package name or it will assume the current authorization ID
  - Does not automatically commit unless auto-commit is enabled
  - Provides a quick way to recreate a package
- Example:
  - REBIND PACKAGE package-name

DB2 Data Management Software

IBM

e business software

Chapter 10: Maintaining Data

DB2 Export
DB2 Import
DB2 Load
DB2MOVE, DB2LOOK
REORGCHK, REORG, RUNSTATS, and REBIND
INSPECT Command

IBM Software Group

INSPECT Command

- Inspect database for architectural integrity
- Check pages of the database for page consistency
- Check the structures of table objects and structures of table spaces are valid
- INSPECT is online and will not lock objects
- Objects can be specified with INSPECT are databases, table spaces, and tables
- Access database objects using uncommitted read isolation level
- INSPECT check processing will write unformatted data results to a specified file
- Use the DB2INSPT utility to format the inspection result data
- One of the following authority can use the INSPECT command:
  - SYSADM
  - SYSCTRL
  - SYSMAINT
  - DBADM
  - CONTROL privilege if a single table

INSPECT Command

```
>>-INSPECT--CHECK-------------------------------------------->

>----+-+-DATABASE--+-------------------------------------+------------+-+---->
     | |           '-BEGIN TBSPACEID--n--+-------------+-'            | |
     | |                                 '-OBJECTID--n-'              | |
     | '-TABLESPACE--+-NAME--tablespace-name-+--+-------------------+-' |
     |               '-TBSPACEID--n----------'  '-BEGIN OBJECTID--n-'   |
     '-TABLE--+-NAME--table-name--+--------------------+-+------------'
              |                   '-SCHEMA--schema-name-' |
              '-TBSPACEID--n--OBJECTID--n----------------'

                             .-LIMIT ERROR TO DEFAULT--.
>--+---------------------+--+-------------------------+--------->
   '-FOR ERROR STATE ALL-'  '-LIMIT ERROR TO--+-n---+-'
                                              '-ALL-'

>--+------------------+--RESULTS--+------+--filename------------>
   '-| Level Clause |-'           '-KEEP-'

Level Clause:

   .-EXTENTMAP NORMAL----.  .-DATA NORMAL----.
|--+---------------------+--+---------------+-----------------><
   '-EXTENTMAP--+-NONE-+-'  '-DATA--+-NONE-+-'
                '-LOW--'            '-LOW--'

   .-BLOCKMAP NORMAL----.  .-INDEX NORMAL----.
>--+--------------------+--+----------------+-----------------><
   '-BLOCKMAP--+-NONE-+-'  '-INDEX--+-NONE-+-'
               '-LOW--'             '-LOW--'

   .-LONG NORMAL----.  .-LOB NORMAL----.
>--+----------------+--+---------------+------------------------|
   '-LONG--+-NONE-+-'  '-LOB--+-NONE-+-'
           '-LOW--'            '-LOW--'
```