

Teradata Bacis

Lesson 06: Teradata Utilities
Multiupload

Module Object

- Introduction about Teradata Utility
- About Multi load
- Supporting Environment
- Multi Load Tasks
- Multi Load Tasks-IMPORT
- Phases of Import Task
- Example of Import Task
- Multi Load Tasks-Delete
- Example of Delete Task
- DELETE Task Differences from IMPORT Task
- Restarting Multiload
- MultiLoad Commands



Introduction about Teradata Utility

- What is the need of Teradata utilities in Data ware house
 - Quick access to data for more timely decision making.
 - Solutions for the entire spectrum of load requirements from batch to near real time.
 - Unmatched scalability for large volume loads.
 - Fail-proof loads with checkpoint restart capabilities.
 - Proven technology from the data warehouse technology leader.
 - Integration with industry-leading ETL and ELT tools.
- Teradata Utilities
 - BTEQ: Help for Report formatting, Ad hoc query tool, Database administration, Best for small data volumes
 - Multi Load :High-performance data unload in client format. Fast Load: High-performance initial table load.
 - Multi Load: High-performance maintenance operations applies updates to multiple tables in single pass.
 - Apart from these teradata having other utilities like Teradata Parallel Transporter, T pump e.t.c.

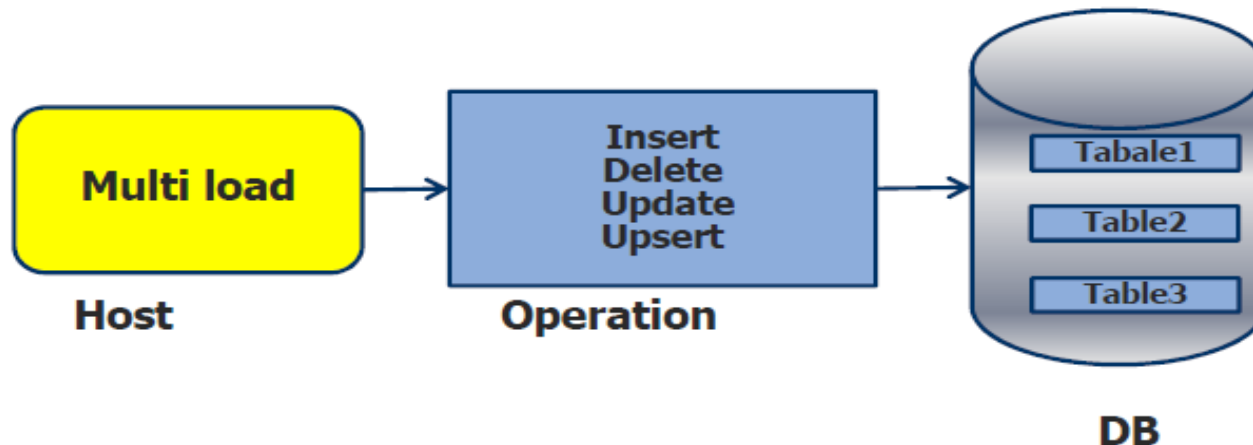
About Multi load

- Multi load is a command driven parallel load utilities , used to load high volume of data to populated or empty
 - teradata tables and views. Teradata MultiLoad executes a series of MultiLoad commands and Teradata SQL statements written in a batch mode job script or interactively entered. The MultiLoad commands provide the session control and data handling specifications for the data transfer operations.
- Features:--
 - Batch mode utility that runs on a server or host system. Supports up to five populated or empty tables.
 - Supports INSERTs, UPDATEs, DELETEs and UPSERTs; typically with batch inputs from a host file. Allows Duplicate rows.
 - Host and LAN support. Full Restart capability.
 - Error reporting via error tables.
 - Support for INMODs MULTILOAD HOST Teradata.

About Multi load

■ Features:--

- Batch mode utility that runs on a server or host system. Supports up to five populated or empty tables.
- Supports INSERTs, UPDATEs, DELETEs and UPSERTs; typically with batch inputs from a host file. Allows Duplicate rows.
- Host and LAN support. Full Restart capability.
- Error reporting via error tables.
- Support for INMODs MULTILOAD HOST Teradata.



About Multi Load

■ Advantages:-

- Minimizes the use of the PEs.
- Gets input data to the AMPs as quickly as possible.
- Uses multiple-AMP sessions.
- Uses the parallelism of the AMPs to apply changes.
- Keeps BYNET activity low with AMP-local processing.
- Avoids Transient Journaling overhead.
- Allows Checkpoint/Restartability even with down AMPs.
- Prevents lengthy rollbacks of aborted jobs.
- Allows for maximum access to table during processing.
- Posts errors to special error tables.
- Provides extensive processing statistics

Supporting Environment

- The Multi Load utility is supported either on either the mainframe or on network attached system(LAN).
- The LAN environment supports the following Operating Systems:
 - UNIX MP-RAS
 - •Windows 2000
 - •Windows 95
 - •Windows NT
 - •UNIX HP-UX
 - •AIX
 - •Solaris SPARC
 - •Solaris Intel

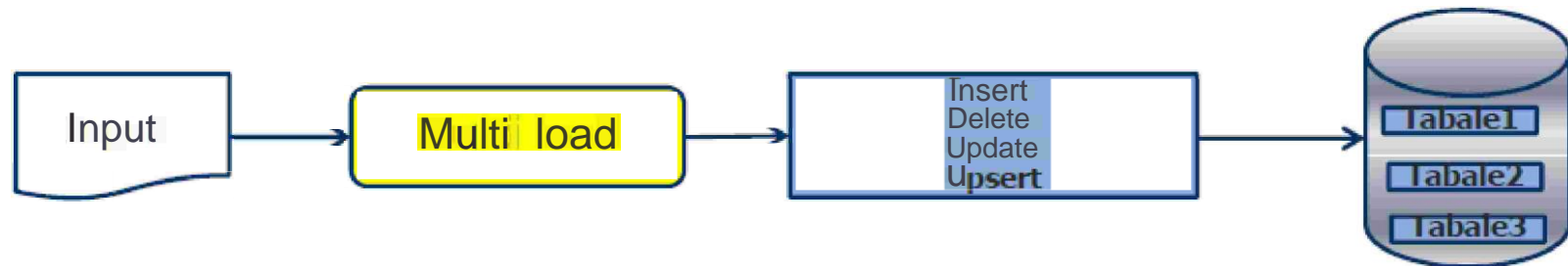
Supporting Environment

- The Mainframe (Channel Attached) environment supports the following Operating Systems:
 - MVS
 - VM
- CAUTION: The Teradata RDBMS will only support a maximum of 15 simultaneous Fast Load, Multi Load, or Fast Export utility job

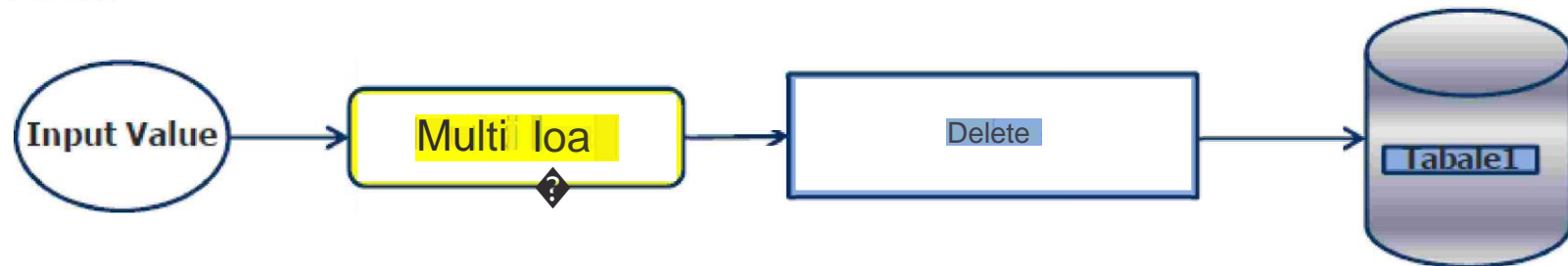
Multi Load Tasks

- Basically Multiload use the source data and do the operation like Insert, Update, Delete and Up sert and keep in Teradata table.
- There are two types of tasks that Multiload can perform.

1) Import Task



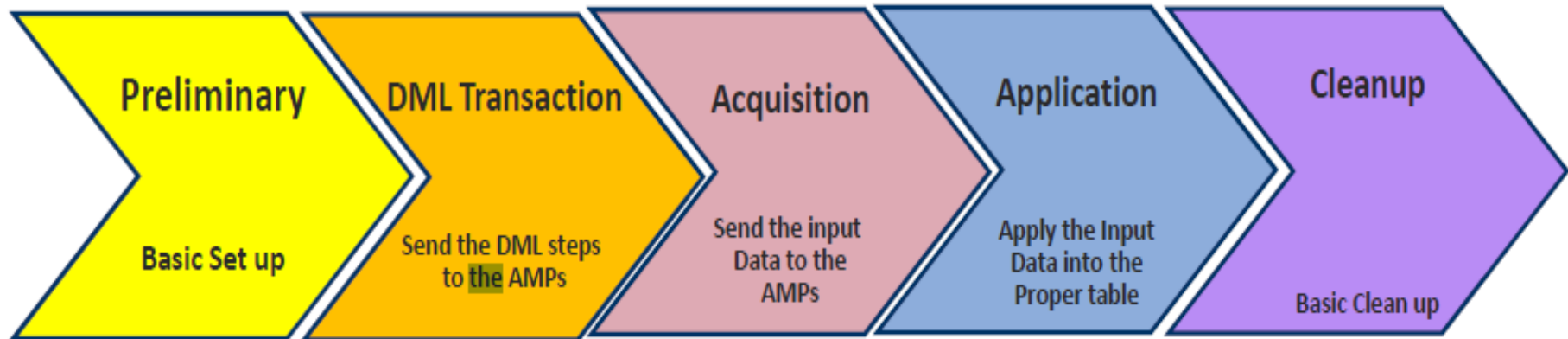
2) Delete Task



Multi Load Tasks-IMPORT

- INSERTs, DELETEs, UPDATEs and UPSERTs are allowed.
- Up to a maximum of five tables:–Empty or populated.–NUSIs permitted.
- MultiLoad Import task operations are always primary index operations - however, you are not allowed
- to change the value of a table's primary index.
- Change the value of a column based on its current value.
- Permits non-exclusive access to target tables from other users except during Application Phase.
- Input error limits may be specified as a number or percentage.
- Allows restart and checkpoint during each operating phase.
- IMPORT tasks cannot be done on tables with USI's, Referential Integrity, Join Indexes, Hash
- Indexes, or Triggers.–IMPORT tasks can be done on tables defined with "Soft Referential Integrity".

Phases of Import Tasks



Multiload: Preliminary Phase

- Preliminary Phase:-
- Validate the Multi load and SQL statements.
 - The first task is to be sure that the SQL syntax and Multi Load commands are valid. Multi Load ensures that it is much better to identify any syntax errors, right up front. All the preliminary steps are automated, so no user intervention is required in this phase.
- Establish all Sessions.
 - Second, all Multi Load sessions with Teradata need to be established. The default is the number of available AMPs.

Multiload: Preliminary Phase

■ Preliminary Phase:-

- Create Work Table Multiload create a work table for its support work. Work Table hold two things: the DML tasks requested and the input data that is ready to APPLY to the AMPs. Default is in user's default database and the work table is named WT_TableName
- Alternative may be specified as DataBaseName.WorkTableName

Example:

```
BEGIN [IMPORT] MLOAD  
  
TABLES           Employee, Paycheck  
  
WORKTABLES      util_db.WT_Emp ,  
                  util_db.WT_Pay  
  
...
```

- Create Error Table Multi Load requires two error tables per target table. The first error table contains constraint violations, while the second error table stores Unique Primary Index violations

Multiload: Preliminary Phase

- Preliminary Phase:-

- Error table 1 (ET)
 - Default is the user's database and the table is named ET_Tablename.
 - Contains any errors that occur in the Acquisition Phase.
 - Contains primary index overflow errors that occur in the Application phase
- Error table 2 (UV)
 - Default is the user's database and the table is named UV_Tablename.
 - Contains Application Phase errors.
 - Uniqueness violations
 - Constraint errors
 - Overflow errors on columns other than primary index

Multiload: Preliminary Phase

- Preliminary Phase:-
- Example

```
BEGIN [IMPORT] MLOAD
TABLES Employee, PayCheck
WORKTABLES util_db.WT_Emp, util_db.WT_Pay
ERRORTABLES util_db.ET_Emp util_db.UV_Emp,
util_db.ET_Pay util_db.UV_Pay
...;
```

- Apply locks to target tables:
 - The final task of the Preliminary Phase is to apply utility locks to the target tables. Initially, access locks are placed on all target tables, allowing other users to read or write to the table for the time being. However, this lock does prevent the opportunity for a user to request an exclusive lock. Although, these locks will still allow the MultiLoad user to drop the table, no one else may DROP or ALTER a target table while it is locked for loading. This leads us to Phase 2.

Multiload: DML Transaction Phase

- DML Transaction phase:-

- All of the SQL Data Manipulation Language (DML) statements are sent ahead to Teradata.
- •Teradata's Parsing Engine (PE) parses the DML and generates a step-by-step plan to
- •Execute the request. This execution plan is then communicated to each AMP and stored in the appropriate worktable for each target table.
- Add a USING modifier to the request and host to be filled in from input file.

Multiload: Acquisition Phase

Acquisition Phase:-

1 Get the data from host and apply it to appropriate AMP worktables.

Re block and store in worktable off target table

Set up transition to the Application phase

- The lock applied during this phase is:

○ Acquisition lock

✓ DML -allows all

./ DDL -allows DROP only

Note: A. Errors that occur in this phase go into the Acquisition Error Table (default name is ET_table_name).
B. There is no acquisition phase activity for a DELETE Task

./Add e..Match Tag" information to record, v"Make blocks and send "quick path" to AMPs. v"Deblock and resend record to "correct" A.MP

vSort 1:the reblocked records in the work tables, vSort by hash value and sequence to be applied

vUpgrade locks on target tables to Write.
vSet table headers for Application phase.

Multiload: Acquisition Phase

■ Application Phase:-

- The purpose of this phase is to write, or APPLY, the specified changes to both the target tables and NUSI sub tables. To accomplish the substitution of data into SQL, when sending the data, the host has already attached some sequence information and five (5) match tags (describe in the next slide) to each data row. Those match tags are used to join the data with the proper SQL statement based on the SQL statement. In addition to associating each row with the correct DML statement, match tags also guarantee that no row will be updated more than once, even when a RESTART occurs.
- Execute MLOAD for each target table as a single multi-statement request.
 - End of host interaction until end of phase.
 - AMPs independently apply changes to target tables.
 - Executed as a single transaction without rollback.
 - Restart able based on last checkpoint.
 - No transient journal needed.

Multiload: Acquisition Phase

- The lock applied during this phase is:
 - Application lock
 - DML —allows SELECT with ACCESS only
 - DDL —allows DROP only
- Note: Errors that occur in this phase go into the Application Error Table (default name is UV_tablename).

Multiload: Cleanup Phase

- Cleanup Phase:-
- If the last error code is zero (0), all of the job steps have ended successfully. If this being the case, all error tables made emptied, worktables and the log table are dropped. All locks, both Teradata and MultiLoad, are released. The statistics for the job are generated for output (SYSPRINT) and the system count variables are set. After this, each MultiLoad session is logged off. Below are the summary of the activities:

Multiload: Cleanup Phase

- All locks are released.
 - Table headers are restored across all AMPs.
 - Dictionary cache of Target Tables is spoiled.
 - Statistics are reported.
 - Final Error Code is reported.
 - Target tables are made available to other users.
 - Work Tables are dropped.
 - Empty Error Tables are dropped.
 - Log Table is dropped (if Error Code = 0).

Example of Import Task

```
.LOGTABLE logtableOO'I_mllid;
• LOGON tdp3/user:2,,tyler;
.BEGIN MLOAD IABLESE m1plo.yee,
Employee_History
TEINACITY2
SLEEP 10;
.LAYOUT Record;L;yo;u;t!,l[i,l,l1SJ:o1Mle'9:'!t'!':-----71
.FILLIER in_Tranocode CHAR(3.);
.FIELD in_EmpNo, IN10*
SMALLINT;
.FIELD in_DeptNo* SMALLINT;
.FIELD in_Salary * DECIMAL
.DML IABLES Payroll DO INSERT FOR IMISSING UPDATE
ROWS;
UPDATE Employee SET Salary =
:in_Salary
WHERE EmpNo = :in_EmpNo;
INSERT INTO Employee (EmpNo,
Salary)
VALUES (:in_EmpNo,
:in_Salary);
.DML IABLES Terminate;
DELETE FROM Employee WHERE EmpNo =
:in_EmpNo; INSERT INTO Employee_History
(EmpNo, DeptNo, Date, Salary)
VALUES (:in_EmpNo, :in_DeptNo, :in_Date, :in_Salary);
.IMPORT INFILE
inrfilel
LAYOUT Record;L;yo;u;t!,l[i,l,l1SJ:o1Mle'9:'!t'!':-----71
APPLY Payroll WHERE in_Tranocode = 'DEL'
APPLY Terminate WHERE in_Tranocode = 'DEL'

.ENID
MLOAD;
.LOGOFF;
```

Begin Loading

How many hours (2) to try
Establishing session when initial
Effort to do so is rebuffed

Tells multi load how frequently
in minute (10) to try login into the
stem

Definition of input layout

Definition of an
UPSERT.

file name to import from

End loading.

Multi Load Delete Task

- Not part of Multiload Import, which consider the sql “Delete” statement.
- Not follow the Transient Journal like Import delete.
- No Rollback applies, when job failed.
- Follow the Restartibility and check points
- Not consider the primary index.

Example of Delete Task

```
..LOGTABLE ILogtbl1e003_mid  
..LOGON tdlp3/user2,tyler  
..BEGIN DELETE MLOAD TABLES Employee  
..LA.VOUT R1remove;  
..FIELD inTermdate* ININTEGER;  
  DELETE FROM Employee WHERE le rm_date >   
:in_J?ermdate;  
..IMPORT INIFILEinfUe2LAYOUT Remove;  
..END MIJOA.D;  
..LOGOFF;
```

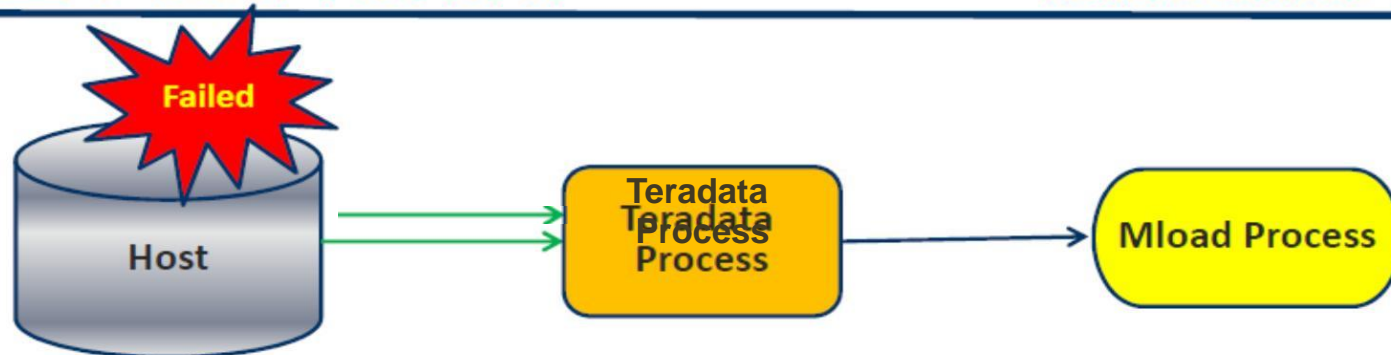
Begin stmt for
delete records

Delete stmt

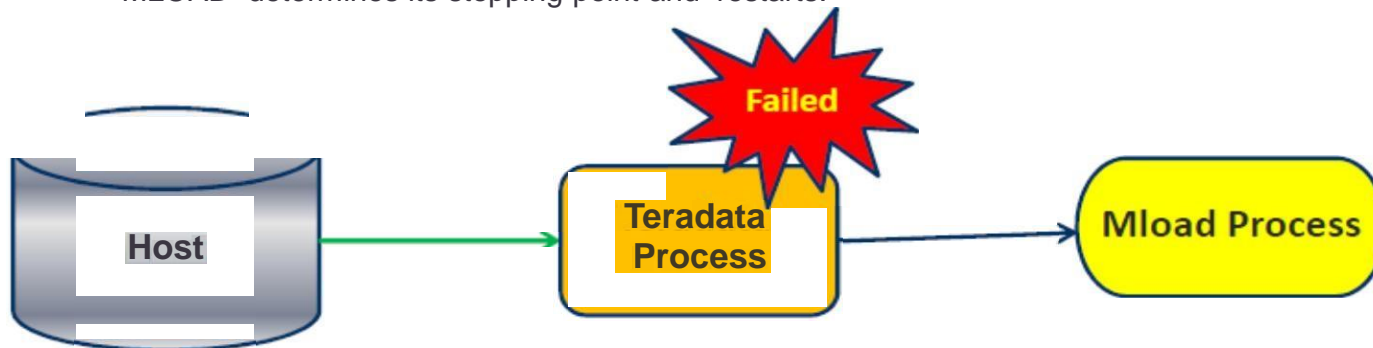
End Mload

DELETE Task Differences from IMPORT Task

- DELETE tasks operate very similarly to IMPORT tasks with some differences:
 - Deleting based on a equality UPI Value is not permitted.
 - A DML DELETE statement is send to each AMP with match tag parcel.
 - No Acquisition phase.
 - Application Phase reads all target blocks and deletes qualifying rows.
- Multiload Delete vs SQL Delete.
 - Multi load Delete is Faster as compared to SQL Delete.
 - Multi load Delete is restartable.

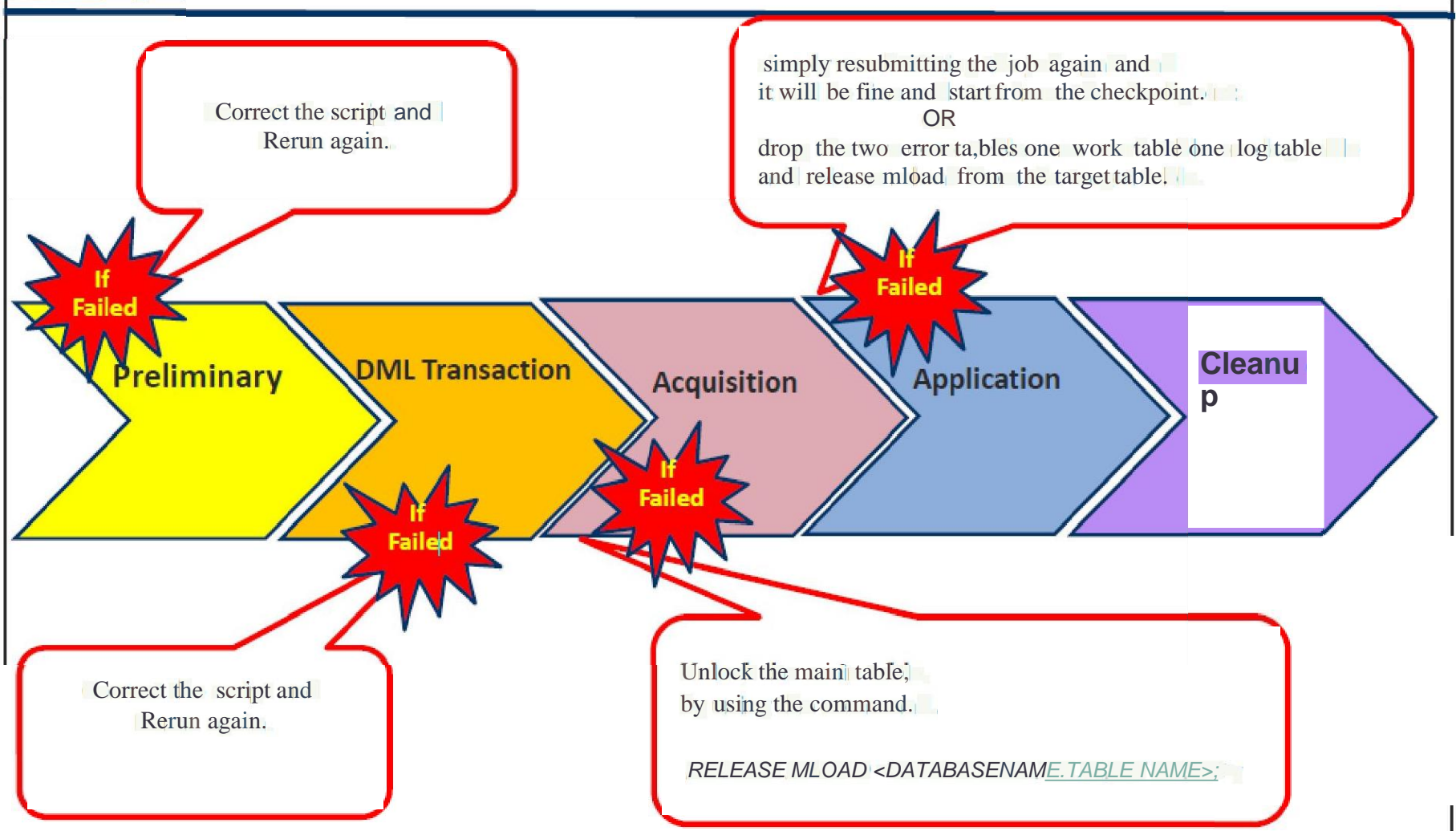


Resubmit the original script.
MLOAD determines its stopping point and restarts.



MLOAD reinitiated automatically after Teradata recovery.
Continue from checkpoint without user interaction.

Restarting Multiload



MULTILOAD COMMAND

.BEGIN =IMPORT]
MLOAD .BEGIN
DELETE MLOAD

This command communicates directly with Teradata to specify if the Multiload mode is going to be IMPORT or DELETE. Note that the word IMPORT is optional in the syntax because it is the "DEFAULT". DELETE is required. *We recommend using the word IMPORT to coding consistent and easier for others to read.* Any parameters load, such as error limits or checkpoints will be included under the .BEGIN command, too. It is important to know which commands or parameters are optional since, if you do not include them, Multiload may supply defaults that may impact your load.

but
make time
for the

.DML LABEL

The DML LABEL defines treatment options and labels for the application {APPLY) of data for the INSERT, UPDATE, UPSERT and DELETE operations. A LABEL is simply a name for a requested SQL activity. The LABEL is defined first, and then referenced later in the APPLY clause. This is the Multiload to finish the APPLY operation. *With the changes to the designated databases and tables.*

.FIELD

This defines a column of the data source record that will be sent to the Teradata database via SQL. *When writing the script, you must include a FIELD for each data field you need in SQL. This command is used with the LAYOUT command.*

.FILLER

Do not assume that Multiload has somehow uncovered records of what you, used in your term paper at the university! FILLER defines a field that is accounted for as part of the data source's row format, but is *not* sent to the Teradata DBS. It is used with the LAYOUT command:

.LAYOUT

LAYOUT defines the format of the INPUT DATA record so Teradata knows what to expect. If one record is not large enough, you can concatenate multiple by using the LAYOUT parameter CONTINUEIF to tell which value to perform for the concatenation. Another option is INDICATORS, which is used to represent nulls by using the bit map (1 bit per field) at the front of the data record.

MULTILOAD COMMAND

<code>.LOGON</code>	This specifies the username or LOGON string that will establish sessions for Multiload with Teradata.
<code>LOGTABLE</code>	This support command names the name of the Restart Log that will be used for storing CHECKPOINT data pertaining to a load. The LOGTABLE is then used to tell Multiload where to RESTART, should that be necessary. It is recommended that this command be placed before the .LOGON command.
<code>.LOGOFF</code>	This command terminates any sessions established by the LOGON command.
<code>.IMPORT</code>	This command defines the INPUT DATA FILE, the LAYOUT to use and here to APPLY the data to SQL.
<code>..SET</code>	Optionally, you can SET utility variables. An example would be { <code>.SET DBName TO CD.ILLTest</code> }.
<code>.SYSTEM</code>	This interrupts the operation of Multiload in order to issue commands to the local operating system.
<code>.TABLE</code>	This is a command that may be used with the .LAYOUT command. It identifies a table whose columns (names, their order and data types) are to be used as the field names and data descriptions of the data source records.

MULTILOAD Parameters

Parameter

Description

SESSIONS

This refers to the number of SESSIONS that should be established with Teradata. For Multiload, the optimal number of sessions is the number of AMPs in the system, plus two more. (One for Control functions and other is for the backup purpose). You can also use MAX or MIN, which automatically use the maximum or minimum number of sessions to complete the job. If you specify nothing, it will default to MAX.

SLEEP
P

Tells MultiLoad how frequently, in minutes, to try logging on to the system.

TENACITY
Y

Tells MultiLoad how many hours to try logging on when its initial effort to do so is rebuffed.

Limitation of MultiLoad

- Though MultiLoad is a very powerful utility; it has following limitations:
 - 1. MultiLoad Utility doesn't support SELECT statement.
 - 2. Concatenation of multiple input data files is not allowed.
 - 3. MultiLoad doesn't support Arithmetic Functions i.e. ABS, LOG etc. in Mload Script.
 - 4. MultiLoad doesn't support Exponentiation and Aggregator Operators i.e. AVG, SUM etc. in Mload Script.
 - MultiLoad doesn't support USIs (Unique Secondary Indexes), Referential Integrity, Join Indexes, Hash Indexes and Triggers.
 - Import task require use of PI (Primary Index).

MultiLoad Review Questions

- Answer True or False.

- 1. With MultiLoad, you can import data from the host into populated tables.
- 2. MultiLoad cannot process tables with USIs or Referential Integrity defined.
- 3. MultiLoad allows changes to the value of a table's primary index.
- 4. MultiLoad allows you to change the value of a column based on its current value.
- 5. MultiLoad permits non-exclusive access to target tables from other users except during Application Phase



MultiLoad Review Questions

- Match the MultiLoad Phase in the first column to its corresponding task in the second column.

1. _____ Preliminary	A. Acquires or creates Restart Log Table.
2. _____ DML Transaction	B. Locks are released.
3. _____ Acquisition	C. Applies (loads) data to the work tables.
4. _____ Application	D. Execute mload for each target table as a single multi-statement request.
5. _____ Cleanup	Stores DML steps in work tables.



Answers of MultiLoad Review Questions

- Match the MultiLoad Phase in the first column to its corresponding task in the second column.

1. ___A___ Preliminary	A. Acquires or creates Restart Log Table.
2. ___E___ DML Transaction	B. Locks are released.
3. ___C___ Acquisition	C. Applies (loads) data to the work tables.
4. ___D___ Application	D. Execute mload for each target table as a single multi-statement request.
5. ___B___ Cleanup	Stores DML steps in work tables.

ANSWERS OF MULTILOAD REVIEW QUESTION

- With MultiLoad, you can import data from the host into populated tables.
 - True
- MultiLoad cannot process tables with USIs or Referential Integrity defined.
 - True
- MultiLoad allows changes to the value of a table's primary index
 - False
- MultiLoad allows you to change the value of a column based on its current value.
 - True
- MultiLoad permits non-exclusive access to target tables from other users except during Application Phase.
 - True

MultiLoad LAB Exercise

- Purpose:

- In this lab, you will use MultiLoad utility to insert the data rows in OLAP_EXAMPLE table. We use the Linux/Unix environment to execute the script.

- What you need:

- The required table and data file.

- Tasks:

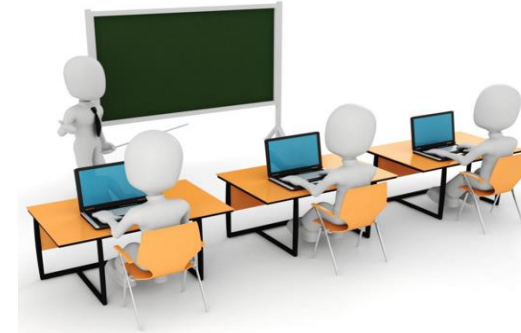
- 1.Prepare a MultiLoad script which inserts rows into the table using the redefinition feature of MultiLoad.



MultiLoad LAB Exercise

■ Tasks:

- 1. Prepare a MultiLoad script which inserts rows into the table using the redefinition feature of MultiLoad.
- 2. Ensure that the table has been created in the database properly.
- 3. Run the script.



- Note: The Trans_Date is exported with an ANSI Date Format of 'YYYY-MM-DD' and a data type of CHAR(10).