

# IBM Db2

Db2's External Tables a deep dive

Mike Springgay  
springga@ca.ibm.com

# Please note :

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

# Notices and disclaimers

•© 2020 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.

•**U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

•Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.**

IBM products and services are warranted per the terms and conditions of the agreements under which they are provided.

•IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

•**Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

•Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

•References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

•Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

•It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

# Agenda

- What are External Tables
- Syntax and Usage
- Diagnostics and Security
- Options
- Summary

# What are External Tables?

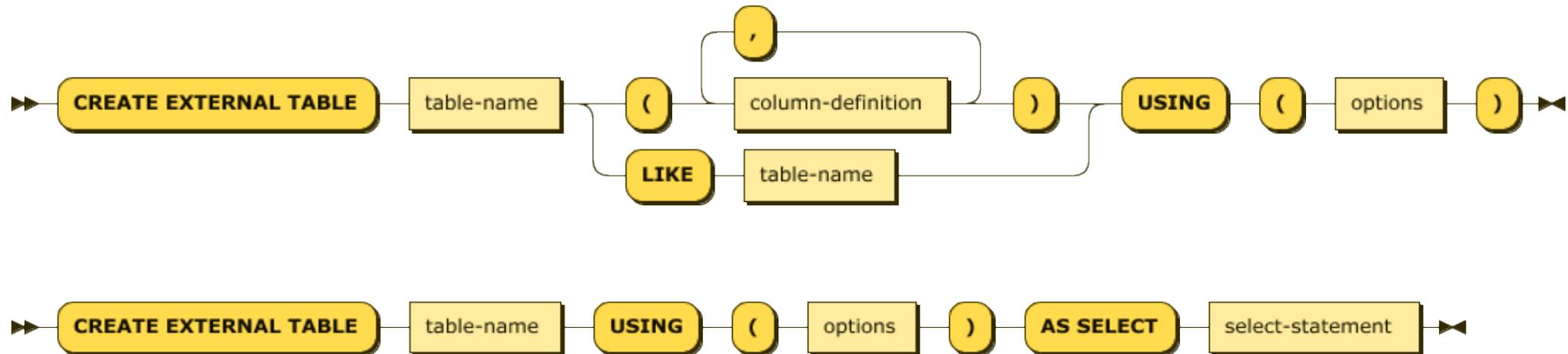
- External Table is a table which points to a file outside the database
- SQL interface to file
- External Table can be used for both read and write operations
- File Formats:
  - TEXT (CSV), FIXED (ASC), INTERNAL (Netezza Binary), BINARY (Db2 Binary)
- Many formatter options: Load, import, export compatible

# External Tables

- Efficient interface to transfer data between systems
- Provide Ad-hoc Query access to files maintained outside database
- Power of SQL during ELT
  - Perform transformations on the data read from the file while INSERTing into target table

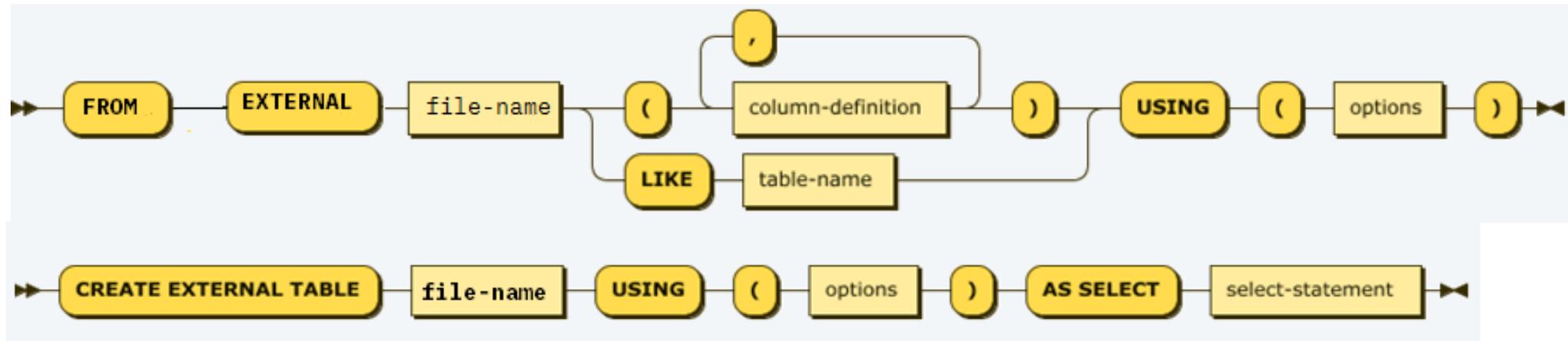
# External Table Types (1 | 2)

- Named
  - The external table has a name and catalog entry like a normal table



# External Table Types (2 | 2)

- Transient
  - The external table has a system-generated name of the form SYSTET<number> and does not have a catalog entry
  - The lifetime of such a table is the duration of the query



# External Table Types – Parameters

- table-name
  - The name of the external table
- file-name
  - The fully-qualified name of the file (or any medium that can be treated as a file) that is to contain the external table to be created
- column-definition
  - Defines the attributes of a column
- using-clause (options)
  - Options control the formatting of data within an external-table file

# External Table Sources

- Local External Tables
  - File or pipe accessible from the server
    - mounted file path
    - Object Storage
- Remote External Tables
  - File or pipe accessible from the client
  - File contents streamed over network connection
  - Network stream can be compressed using GZIP or LZ4

# External Table Operations

- **SELECT**

- External Tables are query-able like a normal table

- **INSERT**

- Is a replace operation – every insert statement truncates file first
- Use bulk operations (INSERT FROM SELECT)

# Restrictions

- No support for UPDATE, DELETE or MERGE targeting an external table
- No DDL, TRUNCATE, or utility operations like REORG, etc.
- Only 1 remote external table per statement
- No support for remote external tables in nested blocks (procedures, functions , anonymous blocks, etc.)
- No ability to APPEND on INSERT (unload)
- Data Type Restrictions:
  - XML type, Large Objects (LOBs) need to be < 64K

# External Tables Usage (1 | 3)

- Creating External Table:
  - `CREATE EXTERNAL TABLE EXTERNAL_TABLE(column-defn) using (DATAOBJECT 'flat-file' DELIMITER '|')`
- Loading into target table from External Table:
  - `INSERT into TARGET_TABLE SELECT * FROM EXTERNAL_TABLE`
- Changing the source data before loading into target table:
  - `INSERT into TARGET_TABLE SELECT (salary*2) as double_salary from EXTERNAL_TABLE`

# External Tables Usage (2 | 3)

- Loading selective rows into target table:
  - `INSERT into TARGET_TABLE SELECT (salary*2) as double_salary from EXTERNAL_TABLE where experience > 5`
- Unloading from a base table to an External Table:
  - `INSERT into EXTERNAL_TABLE SELECT * FROM BASE_TABLE`
- Query an external table directly
  - `SELECT * FROM EXTERNAL_TABLE`
  - `SELECT name, salary FROM EXTERNAL_TABLE`

# External Tables Usage (3 | 3)

- Query a transient external table directly
  - `SELECT * FROM EXTERNAL '/foo/foo.txt' (C1 INT, C2 CHAR(10)) USING(CCSID 1208, DELIMITER '|', FORMAT TEXT)`
- Transient load
  - `INSERT INTO BASE_TABLE AS SELECT * FROM EXTERNAL '/foo/foo.txt' USING(CCSID 1208, DELIMITER '|', FORMAT TEXT)`
- Transient unload
  - `CREATE EXTERNAL TABLE '/tmp/external_table.txt' AS SELECT * FROM BASE_TABLE`

# External Table – File Formats

- TEXT
  - The data to be loaded or unloaded is in ASCII delimited format
- FIXED
  - The data is in fixed-length format (or non-delimited ASCII)
- BINARY
  - The data is in an internal format to Db2
- INTERNAL
  - The data is in an internal format used by Netezza Platform Software (NPS)

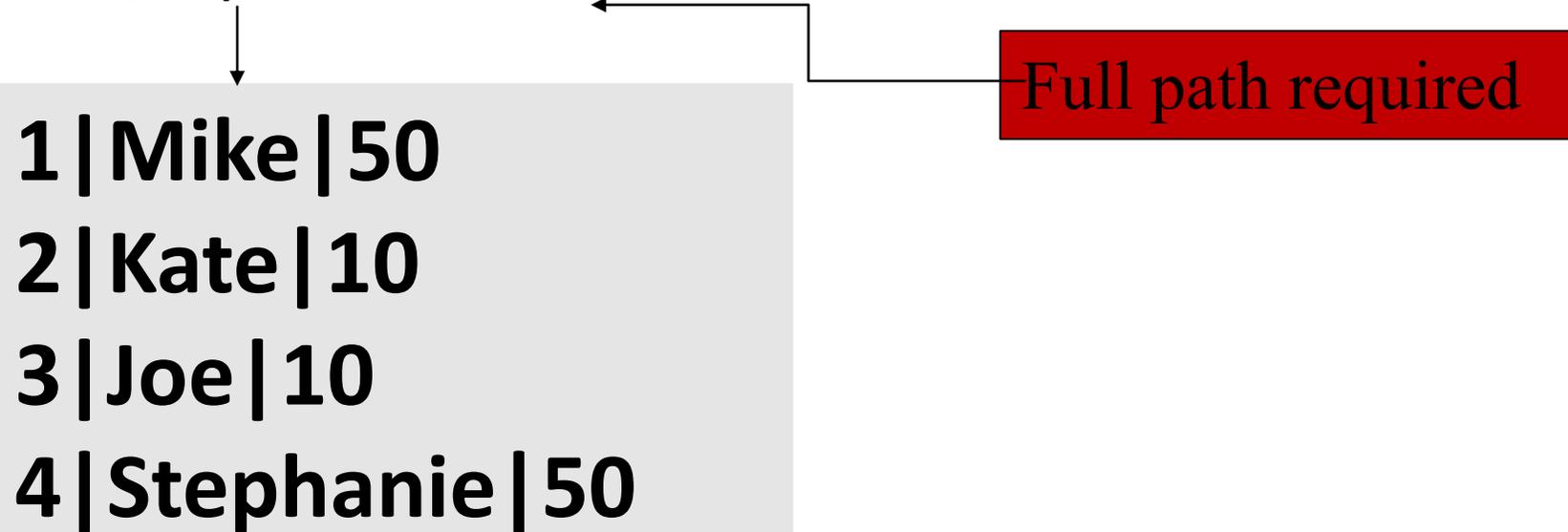
# External Table – Character Encoding specifics

- CCSID
  - Preferred Option for describing character encoding
  - should be specified if file was not extracted with Netezza External Tables
  - should be specified if not unloading to move back to Netezza
- ENCODING option of INTERNAL, LATIN9, UTF8
  - Netezza compatible option – use with files originating from Netezza
  - Appropriate codepage conversion done within database
  - INTERNAL:
    - OCTETS columns encoded in LATIN9, CODEUNIT32 columns encoded in UTF-8
    - Only supported in Unicode database

**\*Default is ENCODING INTERNAL for Netezza compatibility\***

# Example of TEXT (delimited) Format (1 | 2)

- CREATE EXTERNAL TABLE textfile(ID int, Name char(50), DeptCode int) USING (DATAOBJECT '/myfiles/textfile.txt' FORMAT TEXT CCSID 1208 DELIMITER '|')



The diagram illustrates the connection between the SQL statement and the data format. An arrow points from the file path in the SQL statement to a grey box containing the data. Another arrow points from the same file path to a red callout box.

```
1 | Mike | 50  
2 | Kate | 10  
3 | Joe | 10  
4 | Stephanie | 50
```

Full path required

# Example of TEXT (delimited) Format (2 | 2)

- `SELECT * FROM EXTERNAL '/myfiles/textfile.txt'`  
(ID int, Name char(50), DeptCode int)  
`USING (FORMAT TEXT CCSID 1208 DELIMITER '|')`



```
1 | Mike | 50  
2 | Kate | 10  
3 | Joe | 10  
4 | Stephanie | 50
```

# External Table – FIXED FORMAT

- The following parameters apply to FIXED format files
- LAYOUT - A layout is an ordered collection of zone or field definitions
  - USE TYPE - Indicates if the zone is normal data, reference, or filler zone
  - NAME - The name of the zone
  - TYPE - Defines the type of the zone
  - STYLE - Defines the zone representation
  - LENGTH - Specified as bytes or characters followed by the number or the internal reference to the reference zone
  - NULLIF - Definition of the zone nullness attribute
  - RECORDLENGTH - Specifies the length of the entire record

# Examples of Fixed Format (1 | 2)

- CREATE EXTERNAL TABLE LINEITEM\_RECORDLENGTH  
(Col1 char(1), col2 int, col3 Char(20))  
USING (DATAOBJECT '/myfiles/lineitem\_recordlength.fixed'      FORMAT FIXED  
LAYOUT(REF BYTES 1,col1 BYTES \@1, col2 int  
BYTES 1, col3 char(20) BYTES 4) RECORDLENGTH \@1+6)

```
111abcd  
122efgh  
133ijk  
144lmn
```

col1	col2	col3
1	1	abcd
2	2	efgh
3	3	ijk
4	4	lmn

# Examples of Fixed Format (2 | 2)

```
CREATE EXTERNAL TABLE LINEITEM_NULLIF (col1 int, col2 int, col3 int, col4 int)
  USING (DATAOBJECT '/myfiles/fixed_nullif.txt' FORMAT FIXED
  LAYOUT(col1 bytes 5, ref bytes 5, int4 bytes 5 nullif \@2 = 22, ref bytes 5, col3
  bytes 5 nullif &4 = ' 44', ref bytes 5, col4 int bytes 5 nullif &&-1 = '66'))
```

1	2	3	4	5	6	7
11	22	33	44	55	66	77
111	222	333	444	555	666	777

col1	col2	col3	col4
1	3	5	7
11	-	-	-
111	333	555	777

# External Table – Object Storage Access

- External Tables can read and write to Object Storage directly
  - Supports S3 (AWS and IBM Cloud) as well as AZURE
  - Option takes: Endpoint, Credentials and Bucket
- IBM Cloud access using S3 compatible interface
  - Must use HMAC credentials: creating service credentials, specify {"HMAC":true}
- LOG file and BAD file written back to Object Store bucket
  - If LOGDIR specified assumed to be a sub path of “bucket”
- READs are streamed from the object store
- WRITES limited to 5GB – no multi-part upload support at present

# External Table – Object Storage Access Examples

- INSERT INTO BASE\_TABLE AS SELECT \* FROM EXTERNAL '/foo/foo.txt' (C1 INT, C2 CHAR(10)) USING(CCSID 1208, DELIMITER '|', FORMAT TEXT S3('s3.amazonaws.com', 'authkey1', 'authkey2', 'mybucket'))
- CREATE EXTERNAL TABLE '/tmp/external\_table.txt' AS SELECT \* FROM BASE\_TABLE USING(S3('s3-api.us-geo.objectstorage.softlayer.net', 'authkey1', 'authkey2', 'mybucket'))

# External Table – Data Partitioning

- Can be partitioned in DPF database
- Leverage the PARTITION [ALL | (N TO N) | (n,n,...)] option
  - N must be a valid partition number within existing database
- File naming must conform to <filename>.NNN
  - were NNN is 3 digit number corresponding to nodes specified in partition clause
- Leverage the parallelism of DPF cluster
  - Unload PARTITION ALL unloads with no inter-node communication
  - Load multiple files in parallel within single statement
    - No partitioning key – scatter partition assumed

# External Tables Bad and Log File

- A log file is generated for every external table read
  - <database>.<schema>.  
<external-table-name>.  
<file-name>.  
<application-handle>.<id>.log
- Unless NOLOG option specified
- Bad file if any rejected rows

```
=====
Load started at: 09-Apr-20 20:26:11 EDT

Database:      EXTERDB
Schema:        SPRINGGA
DefinerID:     SPRINGGA
Tablename:     SYSTET32839
Datafile:      /home/springga/t5.out
Host:          hotellnx118

Load Options

Record Format:      TEXT
Field delimiter:   '|'
File Buffer Size (MB): 8
Encoding:          N/A
Skip records:      0
FillRecord:        No
Escape Char:       None
Allow CR in string: No
Quoted data:       NO

NULL value:      NULL
Load Replay Region (MB): 0
Max errors:      1
Max rows:        0
Truncate String: No
Accept Control Chars: No
Ignore Zero:     No
Require Quotes:  No

BoolStyle:        1_0
Disable NFC:      No
Date Style:       YMD
Y2Base:           2000
Time Style:       24HOUR
Time extra zeros: No
Time Format:       hh24:mi:ss
NOLOG:            No
DecPlusBlank:    NONE
Quoted Null:      Yes

Decimal Delimiter: '.'
Date Delim:       '-'
Time Delim:       ':'
Date Format:       yyyy-mm-dd
TimeStamp Format:  yyyy-mm-dd hh24:mi:ss.FF6
Meridian Delim:  ''
DateTime Delim:  ''

CCSID :           1208
Record Delim:

Found bad records

bad #: input row #(byte offset to last char examined) [field #, declaration] diagnostic, "text consumed"[last char examined]
-----
1: 1(52)[2, CHAR(50)] text field too long for column, "abc" [ ]

Statistics

number of records read: 1
number of bytes read:  212
number of records skipped: 0
number of bad records: 1
-----
number of records loaded: 0

Elapsed Time (sec): 0.0

=====
Load completed at: 09-Apr-20 20:26:11 EDT
=====
```

# Monitoring

EXT_TABLE_RECV_WAIT_TIME	BIGINT	<u>Total agent wait time for external table readers monitor element</u>
EXT_TABLE_RECVS_TOTAL	BIGINT	<u>Total row batches received from external table readers monitor element</u>
EXT_TABLE_RECV_VOLUME	BIGINT	<u>Total data received from external table readers monitor element</u>
EXT_TABLE_READ_VOLUME	BIGINT	<u>Total data read by external table readers monitor element</u>
EXT_TABLE_SEND_WAIT_TIME	BIGINT	<u>Total agent wait time for external table writers monitor element</u>
EXT_TABLE_SENDS_TOTAL	BIGINT	<u>Total row batches sent to external table writers monitor element</u>
EXT_TABLE_SEND_VOLUME	BIGINT	<u>Total data sent to external table writers monitor element</u>
EXT_TABLE_WRITE_VOLUME	BIGINT	<u>Total data written by external table writers monitor element</u>

# Security

- **EXTBL\_LOCATION**
  - Provides the allowed list of paths external tables may access
- **STRICT\_IO**
  - When enabled enforces “home” directory structure; file access is always `<EXTBL_LOCATION>/authid/<filename>`
- Files are access through the fenced mode process running as definer
  - Definer must have read access on the file (query)
  - Definer must have write access on LOGDIR (query)
  - Definer must have write access on the file (insert)

# Configuration

- EXTBL\_LOCATION
  - discussed in security but important configuration step
- DB2\_FMP\_COMM\_HEAPSZ
  - External tables share the fenced mode process comm heap
  - May need increasing if many concurrent external table operations occurring.
  - Applications are encountering SQL5119N rc = 2

# CLI LOAD Extension

- Enhanced to leverage external tables
  - Specify `SQL_USE_LOAD_WITH_ET` to use External tables instead of LOAD
- Supports “LOAD INSERT” only
- Implicitly used for “LOAD INSERT” within Data Server Driver
  - i.e when no load API available
- Leverages Db2’s binary format

# Many Formatting Options!

Option	Default	Option	Default
BOOLSTYLE or BOOLEAN_STYLE	1_0	DECIMALDELIM or DECIMAL_CHARACTER	'.'
CARDINALITY	(no default)	DELIMITER	' '
CCSID	(no default)	ENCODING	INTERNAL
COMPRESS	NO	ESCAPECHAR or ESCAPE_CHARACTER	(no default)
CRINSTRING	FALSE	FILLRECORD	FALSE
CTRLCHARS	FALSE	FORMAT or FILE_FORMAT	TEXT
DATAOBJECT or FILE_NAME	(no default)	IGNOREZERO or TRIM_NULLS	FALSE
DATEDELIM	'-'	INCLUDEHEADER or COLUMN_NAMES	FALSE
DATETIMEDELIM	A space (' ')	INCLUDEZEROSECONDS	FALSE
DATESTYLE	YMD	LOGDIR or ERROR_LOG	target directory of external-table file
DATE_FORMAT	YYYY-MM-DD	MAXERRORS or MAX_ERRORS	1

# Many Formatting Options!

Option	Default	Option	Default
MAXROWS or MAX_ROWS	0	SOCKETBUFSIZE	8 MB
MERIDIANDELIM	A space (' ')	STRICTNUMERIC	FALSE
NOLOG	FALSE	AZURE	(no default)
NULLVALUE or NULL_VALUE	'NULL'	S3	(no default)
PARTITION	(no default)	TIMDELIM	':'
QUOTEDNULL	TRUE	TIMEROUNDNANOS or TIMEEXTRAZEROS	FALSE
QUOTEDVALUE	NO	TIMESTAMP_FORMAT	'YYYY-MM-DD HH.MI.SS'
RECORDDELIM or RECORD_DELIMITER	'\n'	TIMESTYLE	24HOUR
REMOTESOURCE	LOCAL	TIME_FORMAT	HH.MI.SS
REQUIREQUOTES	FALSE	TRIMBLANKS	NONE
SKIPROWS or SKIP_ROWS	0	TRUNCSTRING or TRUNCATE_STRING	FALSE
		Y2BASE	2000

# External Table – Date Parameters

- **DATEDELIM**
  - The delimiter character that separates the components of a date, according to the format specified by the DATESTYLE option
- **DATESTYLE**
  - How to interpret the date format
- **DATETIMEDELIM**
  - A single-byte character that separates the date component and time component of the timestamp data type
- **DATE\_FORMAT**
  - The format of the date field in the data file
- **Y2BASE**
  - The year that is the beginning of the 100-year range

# External Table – Time Parameters (1 | 2)

- **INCLUDEZEROSECONDS**

- For an unload operation, whether to specify 00 as the value for seconds when no value for seconds is available

- **MERIDIANDELIM**

- A single-byte character that separates the seconds component from the AM token or PM token in the 12-hour delimited and undelimited formats of a time value

- **TIME\_FORMAT**

- The format of the time field in the data file

- **TIMEDELIM**

- The single-byte character that is to separate time components (hours, minutes, and seconds)

# External Table – Time Parameters (2 | 2)

- **TIMEROUNDNANOS or TIMEEXTRAZEROS**
  - Specifies whether records that contain time values whose non-zero precision exceeds six decimal places are to be accepted (and rounded to the nearest microsecond) or rejected
- **TIMESTYLE**
  - The time format that is to be used in the data file
- **TIMESTAMP\_FORMAT**
  - The format of the timestamp field in the data file

# External Table – Decimal and Boolean Parameters

- **BOOLSTYLE** or **BOOLEAN\_STYLE**
  - During a load operation, all Boolean values must use the same style
- **DECIMALDELIM** or **DECIMAL\_CHARACTER**
  - The decimal delimiter for the data types FLOAT, DOUBLE, TIME, and TIMESTAMP
- **DECPLUSBLANK**
  - Specifies how the positive decimal value is represented during the unload operation
- **STRICTNUMERIC**
  - For a load operation, how to treat a value that is to be inserted into a DECIMAL field when its scale exceeds that defined for the field

# External Table – String Parameters

- **IGNOREZERO** or **TRIM\_NULLS**
  - Specifies whether the binary value zero in CHAR fields and VARCHAR fields is to be discarded
- **QUOTEDNULL**
  - For a load operation, how to interpret a value that is enclosed in single or double quotation marks and that matches the null value specified by the NULLVALUE or NULL\_VALUE
- **QUOTEDVALUE** or **STRING\_DELIMITER**
  - Whether data values are enclosed in quotation marks
- **TRIMBLANKS**
  - How an external table is to treat leading or trailing blanks (that is, leading or trailing space characters) in a string
- **TRUNCSTRING** or **TRUNCATE\_STRING**
  - How the system processes a CHAR or VARCHAR string that exceeds its declared storage size
- **REQUIREQUOTES**
  - Whether quotation marks are mandatory

# External Table – Data Format Parameters (1 | 2)

- **CCSID**

- The coded character set identifier (CCSID) of the input data file

- **COMPRESS**

- Specifies whether the source data file data is compressed. Supports GZIP and LZ4
- On read GZIP is implicitly supported

- **CRINSTRING**

- How to interpret an unescaped carriage-return (CR) or carriage-return line-feed (CRLF)

- **CTRLCHARS**

- Whether to allow an ASCII value 1 - 31 in a CHAR or VARCHAR field

- **DELIMITER** or **COLUMN\_DELIMITER**

- The character that is used to delimit the fields of an input or output record

# External Table – Data Format Parameters (2 | 2)

- **ENCODING**

- The type of data in the file
- *Mutually Exclusive with CCSID – Use CCSID if not originating from Netezza*

- **ESCAPECHAR or ESCAPE\_CHARACTER**

- Which character is to be regarded as an escape character

- **FILLRECORD**

- This option specifies whether an input record can contain fewer fields than there are columns defined for the target table

- **NULLVALUE or NULL\_VALUE**

- The UTF-8 string of at most 4 bytes that is to be used to indicate a null value

# External Table – Processing Options

- **INCLUDEHEADER** or **COLUMN\_NAMES**
  - For an unload operation, whether the table column names are to be included as headers in the external-table file
- **LOGDIR** or **ERROR\_LOG**
  - The directory to which the following files are written
- **MAXERRORS** or **MAX\_ERRORS**
  - For a load operation, the threshold for the number of rejected records at which the system stops processing and immediately rolls back the load
- **DATAOBJECT** or **FILE\_NAME**
  - The fully-qualified name of the file (or any medium that can be treated as a file) that is to contain the external table to be created
- **CARDINALITY**
  - Non-zero positive integer value to override the estimation of the expected number of returned rows

# External Table – Processing Options

- **MAXROWS** or **MAX\_ROWS**
  - If set to a positive integer, this specifies the maximum number of records (rows) in the external table that are to be processed
  - If set to 0 (the default), there is no limit and all rows are processed
- **NOLOG**
  - Specifies whether the .log file for the external table is created.
- **PARTITION**
  - If the Database Partitioning Feature (DPF) is enabled for the database, an external table can be partitioned into several files
  - For a partitioned external table, the PARTITION option specifies to which partition or partitions the statement applies
- **RECORDELIM** or **RECORD\_DELIMITER**
  - The string literal that is to be interpreted as a row (record) delimiter

# External Table – Processing Options

- **REMOTESOURCE**

- Where the external-table file resides and, if it resides on a remote system, whether the file data is to be compressed for network transportation

- **SKIPROWS** or **SKIP\_ROWS**

- For a load operation, the number of rows to skip before beginning to load the data

- **SOCKETBUFSIZE**

- The size, in bytes, of the chunks of data that are read from the source file

- **S3**

- Specifies that the source data file is located in an S3 compatible object store

- **AZURE**

- Specifies that the source data file is located in Microsoft Azure object store

# External Table Best Practices (1 | 2)

- Loading
  - Specify `CCSID <codepage>` to specify the codepage encoding of file
    - Use default `ENCODING` only if files originate from Netezza
  - GZIP files can be read implicitly
  - LOCAL & DPF and multiple files consider using `PARTITION [ALL | 0-N]` to load files in parallel using single statement
    - Files do not need to be pre-partitioned to use. Scatter partitioning assumed
  - REMOTE use `LZ4` compression for most efficient network exchange

# External Table Best Practices (2 | 2)

- Unload

- Specify CCSID <db codepage> to keep character data in database encoding
  - Default ENCODING(INTERNAL) only if moving to Netezza
- Use **BINARY** format if moving data homogenously
- Use COMPRESS(**LZ4**) to produced a LZ4 compressed file
  - Optionally COMPRESS(GZIP) to produce a GZIP compressed file but slower
- LOCAL & DPF: Consider **PARTITION ALL** to increase performance by unloading each MLN into its own file.
- REMOTE use **LZ4** compression for most efficient network exchange

# External Table vs Export and Import

- Export

- External tables an improved and faster replacement
- Use when external table restrictions (eg. LOBs) can not be over come

- Import

- External tables an improved and faster replacement
  - Import uses single row INSERT statements
- Use when external table restrictions (eg. LOBs) can not be over come
- Can inject “commit points”

# External Table compared to INGEST and LOAD

- INGEST
  - Continuous ingestion tool, formatting off loaded to INGEST tool location
  - Can inject “commit points”
  - Leverages “array insert” better than single row
    - but still not as efficient as INSERT FROM SELECT
- Load
  - Requires exclusive access to table
  - Reduce logging requirements (assuming full logging)
    - although COPY option still requires space
  - Less flexibly to transform input data during load operation
    - staging often required

# Other things to keep in mind

- PIPE need to be consumed or feed
  - ET may appear to hang when pipes are used if no consumer or producer present
  - By default no timeout – will be unblocked by interrupt or force
    - client side must be an interrupt
- Unload
  - NO data NO file
  - PARTITION ALL - Data File created only on partitions containing data
- No Commit Interval – need to self batch or use Not Logged Initially (Unless Db2 Warehouse with reduced logging enabled)

# External Tables Summary

- Advantages of using External Tables
  - SQL based, works from any client or application
  - Complex expressions/Joins/Filters on data being loaded – ETL capabilities
  - Does not require a Z-lock on target table
  - Load remote data files without any staging space – remote streaming
  - Ability to load compressed files directly and from heterogenous data sources
  - Easier to integrate with external application because of SQL interface
  - Enhanced file security
  - Logged insert operations into target table
  - Constraint validation on target table
- Exploited in DataStage 11.7.1.1 and Informatica 10.4

# Acknowledgements and Disclaimers

**Availability.** References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© **Copyright IBM Corporation 2020. All rights reserved.**

— ***U.S. Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.***

IBM, the IBM logo, [ibm.com](http://ibm.com), and DB2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at

•“Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

•Oracle is a trademark of Oracle Corporation

•Other company, product, or service names may be trademarks or service marks of others.

# Db2 Resources

## Information Resources:

- Db2 Roadmap - <http://ibm.biz/AnalyticsRoadmaps>
- Db2 RFE (Idea) Portal - <http://ibm.biz/submitdb2idea>
- Db2 Recorded Educational Webinars- <http://ibm.biz/db2webinar>
- Subscribe to Db2 technical newsletter - <http://ibm.biz/db2nlsignup>
- Connect with the Db2 online community - <http://ibm.biz/db2tribe>

## Developer Resources:

- Db2 Developer Page to get started - <http://ibm.biz/db2developer>
- For Experienced Db2 developers, get your fav Db2 code sample on github - <http://ibm.biz/db2github>
- Want to try Machine Learning with Db2, check out - <http://ibm.biz/learndb2>
- Want details on Db2 Python Driver - <http://ibm.biz/db2-drivers-python>
- Want Details on Db2 PHP Driver - <http://ibm.biz/db2-drivers-php>
- Want Details on Db2 Node.js Driver - <http://ibm.biz/db2-drivers-node>
- Download the free Db2 python e-book - <http://ibm.biz/db2pythonbook>

IBM