Db2 V11.5 Technical Update

George Baklarz
Db2 Digital Technical Engagement

Legal Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. 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. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MY NEVER BE DELIVERED BY IBM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. 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 ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

IBM Db2 - the AI database

Analytics Performance powered by cost optimizer, MPP
OLTP performance with Scale up (and/or out) high volume low latency transactions with SMP, pureScale
Enterprise Grade Resiliency & Scalability with pureScale, HADR
Comprehensive Security with RCAC, LBAC, Encryption, SSL

Powered by AI

Confidence-based query results
leveraging ML-SQL

Up to 10x better query performance
powered by an ML-Optimizer

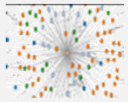
No data movement & a single view on all data
delivered by Data Virtualization

Auto resource optimization
delivered by Adaptive Workload Management

Built for AI

Faster data exploration
by using NLQ in Augmented Data Explorer

Build AI based applications
with Python, GO , JSON and Jupyter notebooks

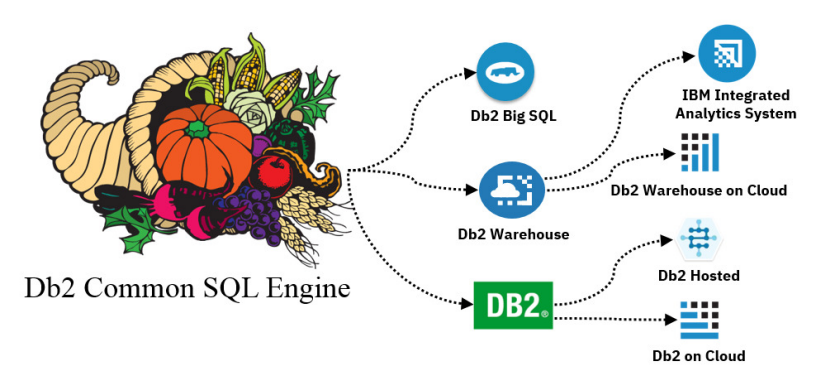


Model Complex Relationships
by using Db2 Graph and SQL

Blockchain Ready
using Db2 Blockchain Connector

Db2 11.x GA – Key Objective

- With the introduction of Db2 11.x in 2019, all Db2 products will be based on the same level of the Db2 common SQL engine



- Benefits for Db2 customers:
  - Existing enhancements in Db2 Warehouse branch will become available to them
  - Reduced latency between feature appearance in different Db2 products as development will all be on same level
    - Any enhancement that is “ready to go” for Db2 environments will appear in next Db2 update

Db2 - Graph Capability

Real time visualizations and modelling

Support graph queries and SQL on the same data

- A graph layer between graph applications and Db2
- No change to Db2 runtime for relational data
- Support open-source graph with Gremlin language & Tinkerpop framework
- Store graph data in tables, so that SQL engine can query them
- ACID transactions update the graph real-time without disturbing (huge number of) existing relational applications
  - Ability to view the graph real-time (as and when transactions happen)
- Also provide a graph view of data with Gremlin API
- Run SQL analytics on graph data, as SQL is fully operational on graph data

Db2 - Natural Language Query (NLQ) Support

New tool add on for data analysis

Eliminate the need to know ANSI SQL syntax

- Conversational analytics is the future…
  - Alexa, Google Home, etc. prove trend
  - Natural extension of conversation to data
  - It is fast, accommodative, and can work with any schema / table
  - Covers most use cases for business users
- Free tool available to download from Developer Works
  - Currently in Beta
  - Seamlessly plugs and plays with Db2

Db2 - Augmented Data Exploration Support

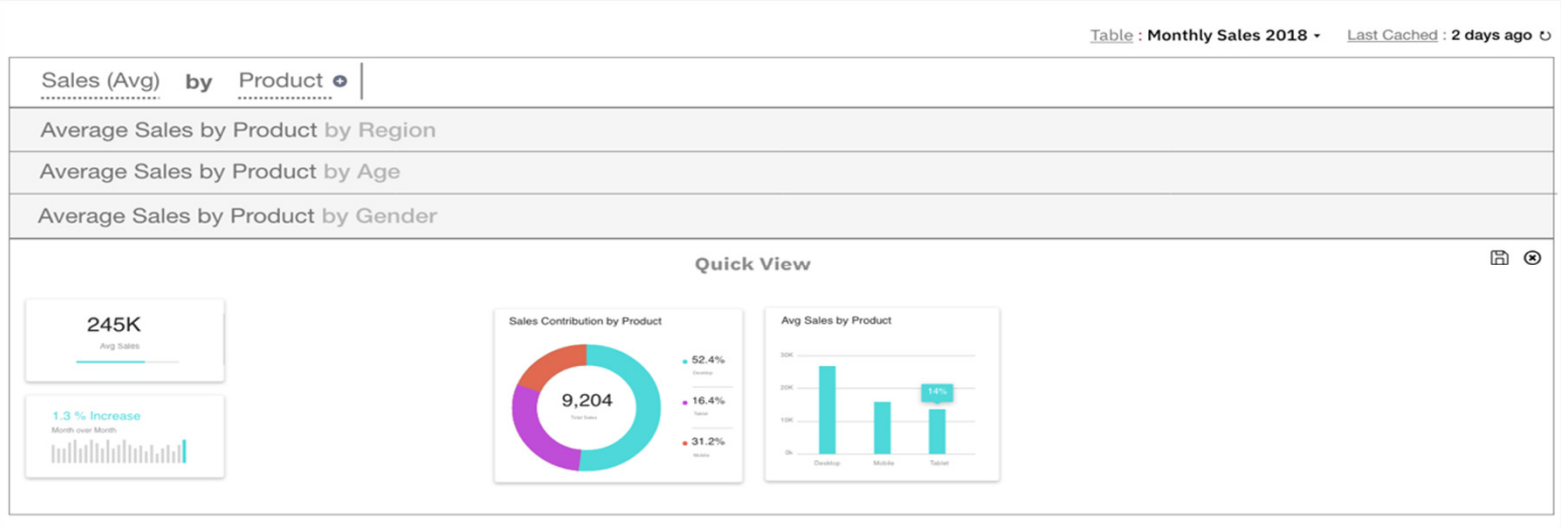
New tool for rich and deeper insights

Get to the insights faster – explore unfamiliar data sets

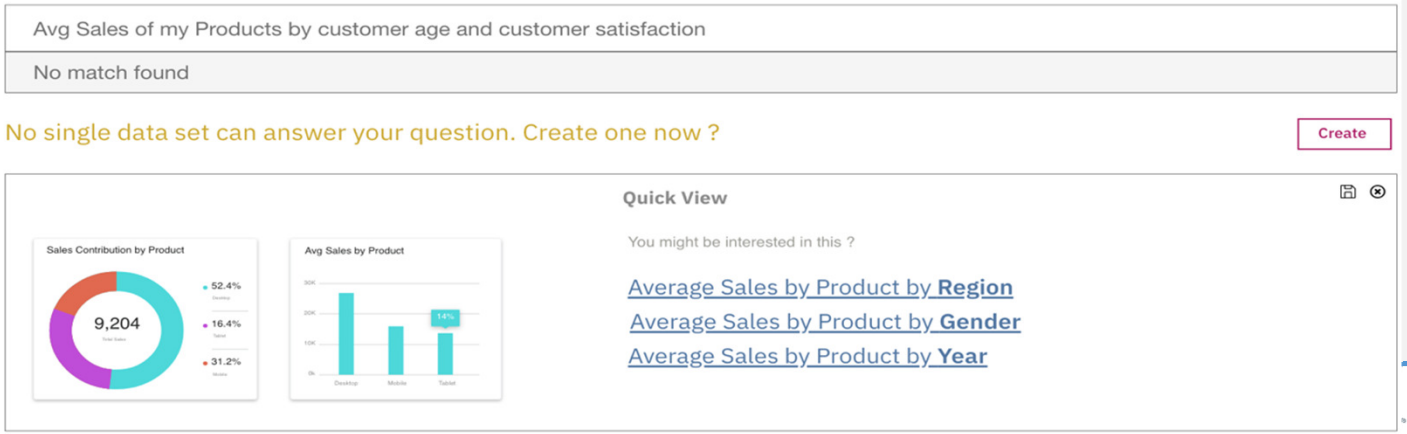
- Data scientists often face the challenge of exploring data sets that are not familiar to them
  - Time consuming
  - Not sure what to look for
  - Need some knowledge transfer or understanding of what is in the data set
- Need an intuitive data exploration tool
  - Provide multiple views of the data with minimal touch
- Free tool available to download from Developer Works
  - Currently in Beta
  - Seamlessly plugs and plays with Db2

Beta – Try out the new Db2 Augmented Data Explorer

https://www.ibm.com/us-en/marketplace/db2-augmented-data-explorer



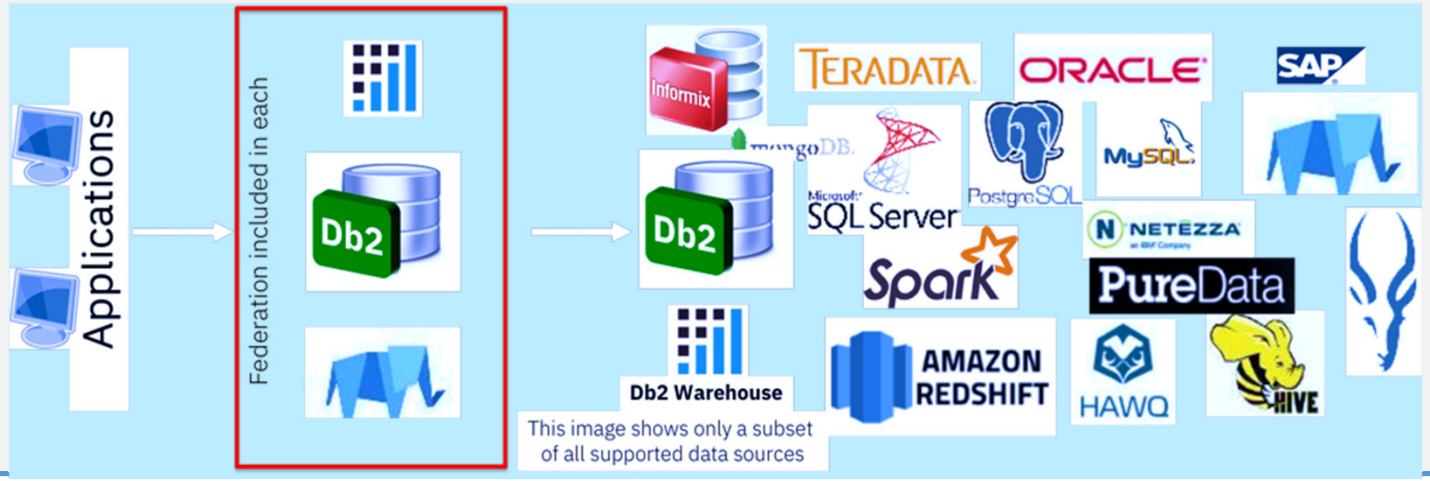
Answer known and unknown questions



Db2 Data Virtualization is the next generation of federation

Data Virtualization will offer sophisticated capabilities to work with data distributed across multiple data stores

- Continuing to invest in federation capabilities and performance
- Extending federation to more and more data sources
- More drivers directly integrated into the Db2 installer
- Expansion to both SQL- and NoSQL-based data stores
- Expansion to both on-premises and cloud data sources
- Expansion to open source databases like PostgreSQL and MySQL



Db2 – ML Optimizer

Tech Preview

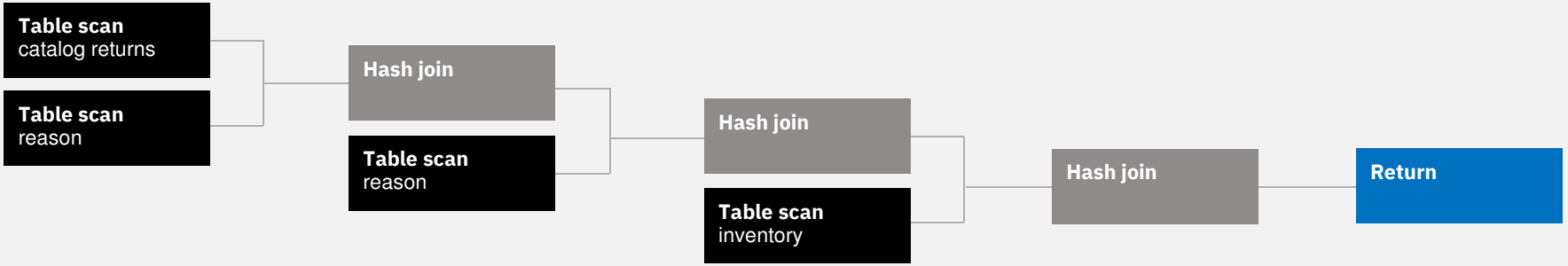
Machine Learning Optimizer that improves from experience to make query optimization simple, reliable and stable

- Number of rows flowing through the various operations impacts performance for most common issues and is calculated via cardinality estimates
- Basic tuning to improve cardinality estimation is high impact
- **Phase 1 – Cardinality Estimation**
  - Initial phase – Support equality & range local operators with no expressions
  - Future phases – Cardinality support for expressions, predicates & BLU, support for join enumerations,

Without Machine Learning

Tech Preview

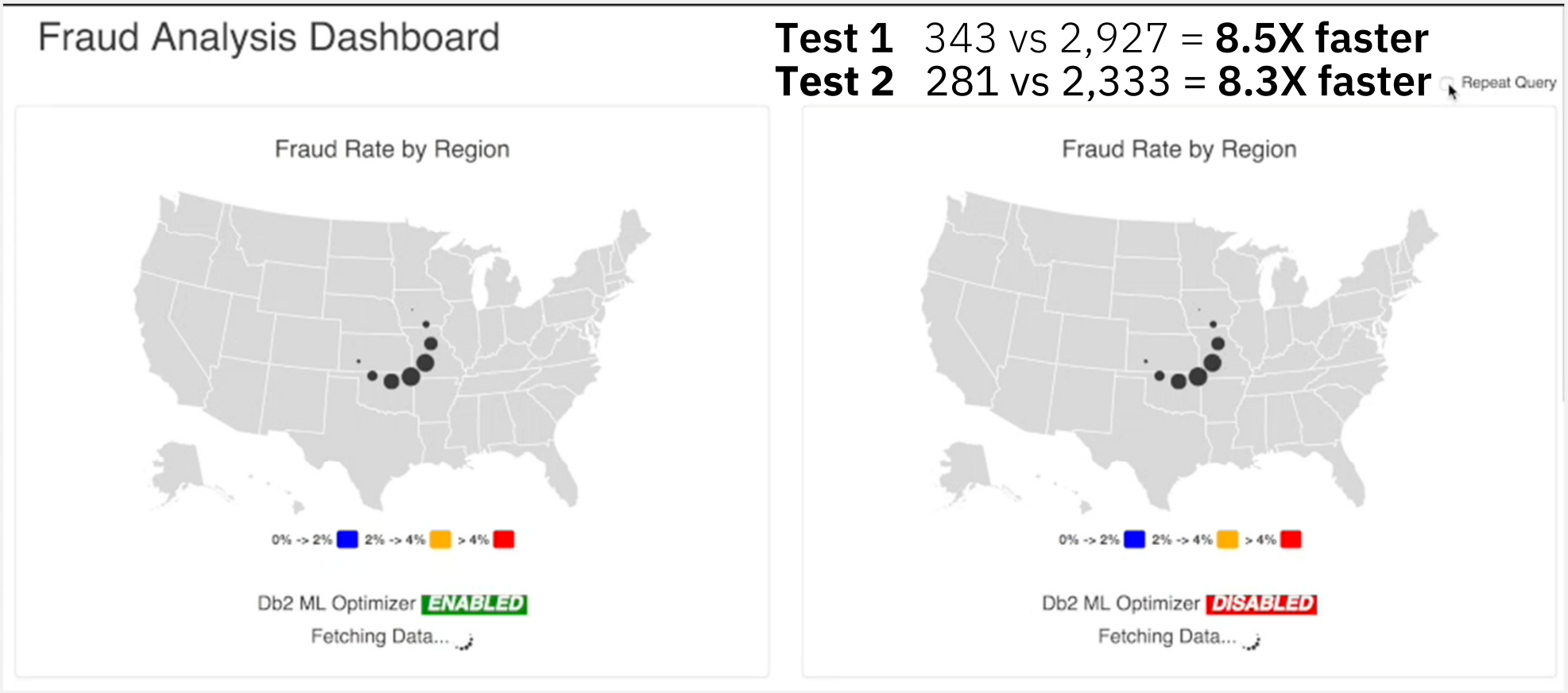
With Machine Learning



ML Optimizer Performance Test

Tech Preview

Test 1 343 vs 2,927 = **8.5X faster**
Test 2 281 vs 2,333 = **8.3X faster**



Db2 11.x Highlights

- Performance enhancements
- Compression and Space Reclamation Enhancements
- SQL compatibility enhancements
- Db2 pureScale Feature enhancements
- Workload management enhancements
- Monitoring enhancements
- IBM data server clients and drivers enhancements

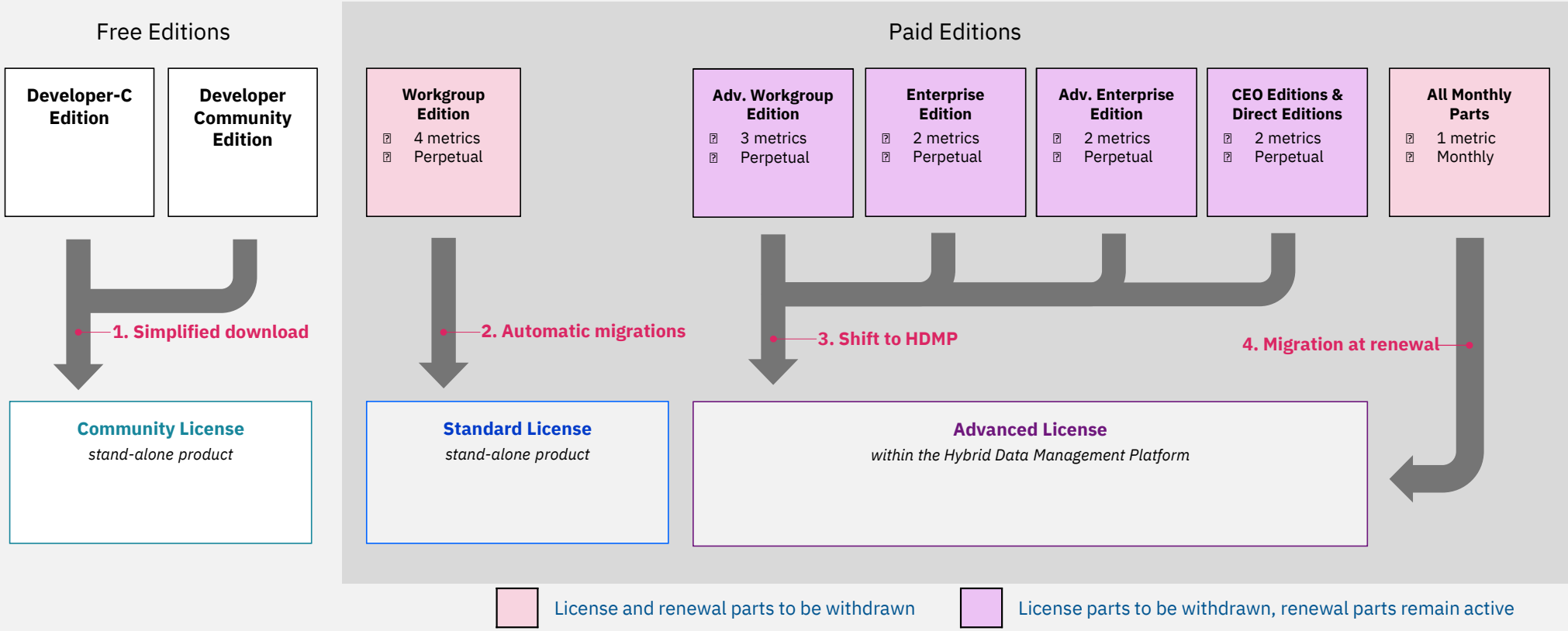
Db2 Edition Rationalization

One Db2 download, three Db2 licenses

On 6/4/2019, Db2 will be available as one common download from www.IBM.com, Passport Advantage, DockerHub, and Fixpack Central. All downloads will begin as the free Db2 Community License, but upon purchase of a Standard or Advanced license, customers can thereafter apply the respective .LIC file to upgrade from the Community License. This will result in one, common Db2 download with a common experience shared across all three license flavors. Each license is described below:

	Community License	Standard License	Advanced License
Price	Free, unsupported $50.00 USD / core per month, supported	$1,850.00 License + S&S (perpetual) *Subscription TBD*	$61,460.00 USD License + S&S (perpetual) $7,800.00 USD / core (subscription)
License	Development / Production	Development / Production	Development / Production
Metric	Not applicable	Virtual Processor Core (VPC)	Virtual Processor Core (VPC)
Target	Developer community	Departmental	Enterprise
Caps	4 cores, 16 GB RAM, 100 GB Storage	16 cores, 128 GB RAM	Unrestricted
Features	All features	All features	All features

How will IBM consolidate the portfolio?



Performance Enhancements

Performance Enhancements

- SQL insert and update statement optimizations
  - SQL based insert and update statements for columnar tables are optimized to support large data sets efficiently, and encoding dictionary creation for such operations are also optimized
  - Four areas of internal improvements
    - Vectorized insert
    - Efficient bulk insert
    - Vectorized ADC (Automatic Dictionary Creation)
    - Synchronous ADC
- Encoding dictionary support for SQL insert and update
  - Encoding dictionary support for SQL based insert and update statements for columnar tables are enhanced to provide encoding in some cases where encoding was not previously possible including:
    - Declared Global Temporary Tables
    - Not Logged Initially tables
    - Tables created using CREATE TABLE AS Select statements.

Compression and Space Reclamation Enhancements

- New compression daemon
  - EDU db2cmpd executes compression related tasks in the background
- Automatic Recompress feature for tables
  - If an insert operation is used to populate a table, automatic creation of the compression dictionaries begin once a threshold count of rows are inserted into the table
    - This design ensures that a large enough sample of rows exists to build dictionaries that yield an adequate compression ratio
    - Rows that are inserted before the compression dictionaries are created are not initially compressed
  - The Automatic Recompress feature for tables uses the new compression daemon to asynchronously check for tables with uncompressed rows at the front of the table
    - It then recompresses those rows in place which may leave empty extents in the table
    - These empty extents can be reclaimed later with the dictionary

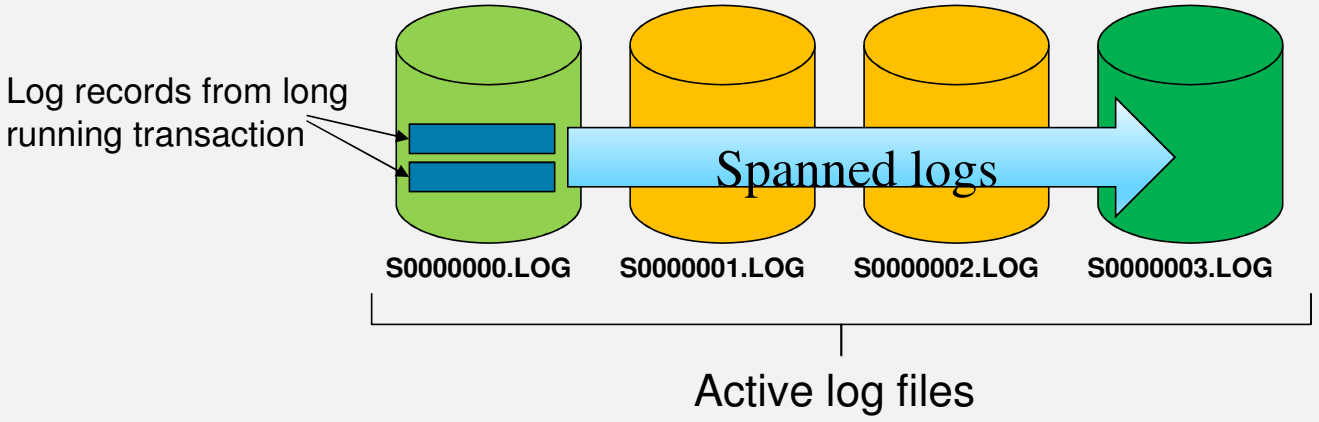
Compression and Space Reclamation Enhancements

- The Automatic Recompress feature requires no user intervention, and uses the REORG TABLE command with a new RECOMPRESS option to recompress the rows
  - The user may notice the REORG TABLE ... RECOMPRESS command while monitoring the table
  - This command is run automatically by the system and is not available for use
- REORG TABLE... RECLAIM EXTENTS
  - The REORG TABLE... RECOMPRESS encodes and compresses the data in place and may leave empty extents in table
  - REORG TABLE... RECLAIM EXTENTS has been enhanced to find these empty extents and reclaim them so that other tables in the same table space may reuse them

Simplified Log Management

Tech Preview

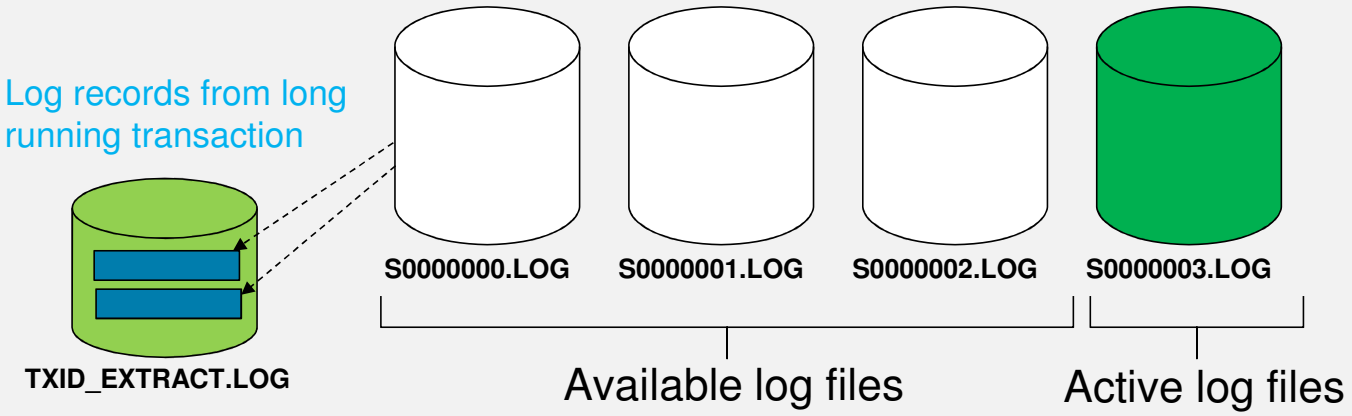
- As part of a focus on simplifying log management, the first delivery will result from a focus on avoiding “log full” scenarios caused by long running, low volume transactions hold up log space
  - E.g. transactions that span multiple log files with little content in the intermediate files



Simplified Log Management Continued

Tech Preview

- Solution is to extract log records for long running active transactions to a separate file and allow intermediate log files to be closed, archived, and reused



SQL Enhancements

SQL Compatibility Enhancements

- New Options for CREATING/DROPPING Existing Tables
- ISO JSON Functions and additional functions for generating identifiers and values
- Oracle Compatibility Library functions
- External Table Support
- Improved NZSQLPL Support

Table Creation - IF NOT EXISTS

- IF NOT EXISTS Clause
  - Specifies that no error message is shown when the table cannot be created because a table with the specified name already exists in the current database and schema
  - The existing table and the specified table in the command are not compared
  - The application must ensure that the target table and rows are as expected

```
CREATE TABLE IF NOT EXISTS CUSTOMER …
```

- Use this option for scripted applications that are running SQL commands
  - When you suppress the Table not found error message, the scripted application is not impacted or halted
  - You cannot use the IF NOT EXISTS option with the AS SELECT clause
- Unless other errors prevent the creation of the table, a CREATE TABLE message is returned although no table is created
  - Failure is ignored if a table with the specified name already exists.

Table Deletion - IF EXISTS

- IF EXISTS Clause
  - Specifies that no error message is shown when the table is not dropped because it does not currently exist in the database

```
DROP TABLE CUSTOMER IF EXISTS
```

- Use this option for scripted applications that are running SQL commands
  - When you suppress the Table not found error message, the scripted application is not impacted or halted
- The DROP TABLE message is returned although no table was dropped
  - Failure is ignored if a table with the specified name does not exist

Table Creation – Columnar BLOB and CLOB

- The columns of a column-organized table can now include the following large object types:
  - CLOB
  - BLOB
  - DBCLOB
  - NCLOB

```
CREATE TABLE LARGEOBJ
  (
   ID    INT,
   IMAGE BLOB(20000) INLINE LENGTH 2000
  )
  ORGANIZE BY COLUMN
```

- Large Objects can be inlined similar to what is available for ROW-organized tables
- Supported datatypes

SMALLINT	INTEGER	BIGINT	DECIMAL	REAL	DOUBLE
DECFLOAT	CHAR	VARCHAR	BINARY	VARBINARY	GRAPHIC
VARGRAPHIC	DATE	TIME	TIMESTAMP(*n*)	BOOLEAN	CLOB
BLOB	DBCLOB	NCLOB			

- XML columns are not supported in a BLU environment

QUOTE_INDENT and QUOTE_LITERAL

- The **QUOTE_IDENT (*string*)** function returns a string that can be used as an identifier in an SQL statement

Statement	Result
`quote_ident('HELLO WORLD')`	`"HELLO WORLD"`
`quote_ident('HELLOWORLD')`	`HELLOWORLD`
`quote_ident('HELLO_WORLD')`	`HELLO_WORLD`
`quote_ident('hello world')`	`"hello world"`
`quote_ident('hello"world')`	`"hello""world"`
`quote_ident('hello''world')`	`"hello'world"`

- The **QUOTE_LITERAL(*string*)** function returns a string that can be used as a string constant in an SQL statement

Statement	Result
`quote_literal(42.5)`	`'42.5'`
`quote_literal('You''re here!')`	`'You''re here!'`

Oracle Compatible Libraries

- **DBMS_APPLICATION_INFO**
  - Includes procedures that set custom client info exported through some of the table functions in the database
  - Helps identify the targeted sessions upon executing the procedure
- **DBMS_LOCK** `Future Release`
  - Provides lock management functionality that allows SQL PL developers to control concurrent access to critical resources in their applications
- **UTL_RAW**
  - Provides a set of routines for manipulating binary data of the data type VARBINARY
  - The routines perform various functions including data conversion, casting, comparison, concatenation, substring, xrange and translation

- The schema for all procedures and functions in these modules is **SYSIBMADM**

DBMS_APPLICATION_INFO Module

- **READ_CLIENT_INFO (** *client_info* **)**
  - Reads and returns the value of the client information field from the current session
- **READ_MODULE (** *module_name, action_name* **)**
  - Reads and returns the value of the module and actions fields from the current session
- **SET_CLIENT_INFO (** *client_info* **)**
  - Sets and registers the value of the client information field from the current session
- **SET_MODULE (** *module_name, action_name* **)**
  - Sets and registers the module name that is currently being executed
- **SET_ACTION (** *action_name* **)**
  - Sets and registers the action name within the current module
- **SET_SESSION_LONGOPS**
  (*rindex, slno, op_name, target, context, sofar, totalwork, target_desc, units* **)**
  - Sets and registers a row in the SYSTOOLS.SESSION_LONGOPS table, to store progress information for long operations

UTL_RAW Module

```
011010110
100010101
111010111
000111010
```

- **BIT_AND (*x,y*)** – bitwise logical AND operation against x, y
- **BIT_OR (*x,y*)** – bitwise logical OR operation against x, y
- **BIT_XOR (*x,y*)** – bitwise logical EXCLUSIVE OR operation against x,y
- **BIT_COMPLEMENT (*x*)** – bitwise logical COMPLEMENT operation against x
- **COMPARE (*x, y, pad*)** – compares to values with optional padding
- **CONCAT (*x1, x2, ..., x12*)** – concatenates up to twelve (12) VARBINARY values into a single value
- **COPIES (*x, n*)** – returns the concatenated results of the VARBINARY value a specified number times
- **LENGTH (*x*)** – returns the length of a VARBINARY value
- **REVERSE (*x*)** – reverses the order of digits of a VARBINARY value
- **SUBSTR (*x, start, length*)** – returns a specified portion of a VARBINARY value
- **XRANGE(*start_byte, end_byte*)** – returns all valid one-byte values that exists in a specified interval, concatenated as a single VARBINARY value

UTL_RAW Module – CAST Operations

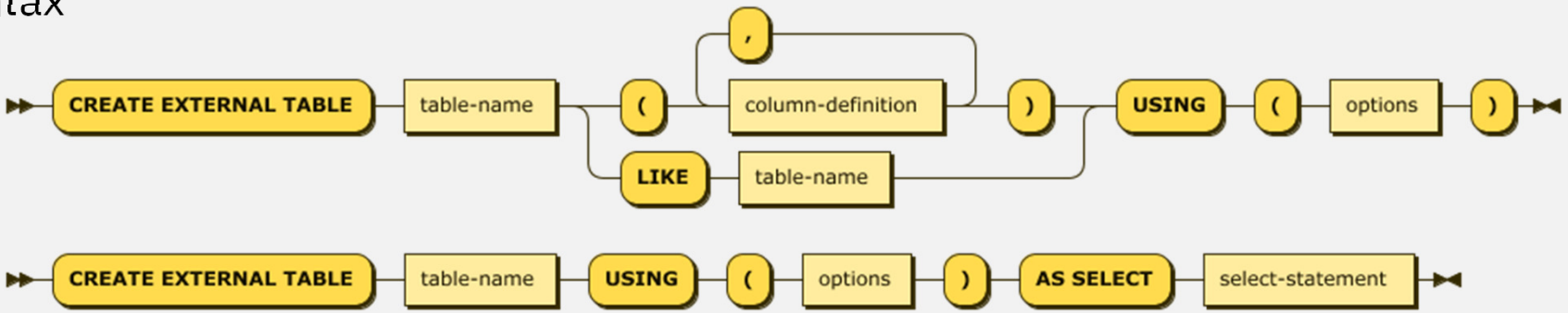
- **CAST_TO_RAW** – casts a VARCHAR value to a VARBINARY value
- **CAST_TO_VARCHAR2** – casts a VARBINARY value to a VARCHAR2 value
- **CAST_FROM_NUMBER** – casts a DECFLOAT value to a VARBINARY value
- **CAST_TO_NUMBER** – casts a VARBINARY value to a DECFLOAT value
- **CAST_FROM_BINARY_DOUBLE** – casts a DOUBLE value to a VARBINARY value
- **CAST_FROM_BINARY_FLOAT** – casts a FLOAT value to a VARBINARY value
- **CAST_FROM_BINARY_INTEGER** – casts an INTEGER value to a VARBINARY value
- **CAST_TO_BINARY_DOUBLE** – casts a VARBINARY value to a DOUBLE value
- **CAST_FROM_BINARY_FLOAT** – casts a VARBINARY value to a FLOAT value
- **CAST_FROM_BINARY_INTEGER** – casts a VARBINARY value to an INTEGER value

External Tables

- External tables resides in a text-based, delimited file or in a fixed-length-format file outside of a database
- Use an external table to:
  - Load data from an external file into a table in the database
  - Unload data from the database into an external file
  - Store data outside the database while retaining the ability to query that data
- External Table Usage
  - Use a FROM clause in a SELECT SQL statement as you would for any other table
  - Specify the external table as the target table in one of the following SQL statements:
    - INSERT SQL
    - SELECT INTO SQL
    - CREATE EXTERNAL TABLE AS SELECT SQL
  - You can perform operations such as casts, joins, and dropping columns to manipulate data during loading

External Table Types

- Named
  - The external table has a name and catalog entry similar to a normal table
- Transient
  - The external table has a system-generated name of the form SYSTET<number> and does not have a catalog entry
  - For example, the system might create a transient external table to hold the result of a query
  - The lifetime of such a table is the duration of the query
- Syntax



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
- ***options***
  - Options control the loading of data to or retrieval of data from an external-table file
- ***select-statement***
  - Specifies that, for each column in the derived result table of the fullselect, a corresponding column is to be defined for the table and populated with the results of the query

External Table – Filename

- The filename is the fully-qualified name of the file that is to contain the external table to be created
- If this parameter is specified, the DATAOBJECT or FILE_NAME option cannot be specified
- When both the REMOTESOURCE option is set to LOCAL (this is its default value) and the extbl_strict_io configuration parameter is set to NO, the path to the external table file is an absolute path and must be one of the paths specified by the extbl_location configuration parameter
  - Otherwise, the path to the external table file is relative to the path that is specified by the extbl_location configuration parameter followed by the authorization ID of the table definer
  - For example, if extbl_location is set to /home/xyz and the authorization ID of the table definer is user1, the path to the external table file is relative to /home/xyz/user1/
- For a load operation, this file must already exist and you must have read permission for it
- For an unload operation, you must have read and write permission for the parent directory of this file, and the file is overwritten if it exists

External Table – Column Definition

- Defines the attributes of a column
- Supported datatypes

SMALLINT	INTEGER	BIGINT	DECIMAL	REAL
DOUBLE	DECFLOAT	CHAR	VARCHAR	BINARY
VARBINARY	GRAPHIC	VARGRAPHIC	DATE	TIME
TIMESTAMP(*n*)	BOOLEAN	CLOB	BLOB	DBCLOB
NCLOB				

- No XML Support

External Table – Like Table

- Specifies that the columns of the table have the same name and description as the columns of the specified table, view, or nickname
- The specified table, view, or nickname must either exist in the catalog or must be a declared temporary table
- The use of LIKE is an implicit definition of n columns, where n is the number of columns in the identified table (including implicitly hidden columns), view, or nickname
  - The implicit definition includes the column name, data type, hidden attribute, and nullability characteristic of each of the columns of that table
  - A column of the new table that corresponds to an implicitly hidden column in the existing table will also be defined as implicitly hidden
  - If EXCLUDING COLUMN DEFAULTS is not specified, then the column default is also included
  - The implicit definition does not include any other attributes of the specified table, view, or nickname
    - The new table does not have any primary key, unique constraints, foreign key constraints, referential integrity constraints, triggers, indexes, ORGANIZE BY specification, or PARTITIONING KEY specification

External Table – Options

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

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

- **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)
- **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

External Table – Time Parameters (2)

- **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)

- **CCSID**
  - The coded character set identifier (CCSID) of the input data file
- **COMPRESS**
  - For an unload operation, whether the source data file data is compressed
- **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
- **ENCODING**
  - The type of data in the file

External Table – Data Format Parameters (2)

- **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
- **RECORDDELIM** 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
- **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
- **SWIFT**
  - Specifies that the source data file is located in a Swift object store
- **S3**
  - Specifies that the source data file is located in an S3 compatible object store

External Table – FORMAT/FILE_FORMAT Options

- **TEXT** - The data to be loaded or unloaded is in text-delimited format
- **INTERNAL** - The data is in an internal format used by Netezza Platform Software (NPS)
- **FIXED** - The data is in fixed-length format
  - The following parameters apply when the FORMAT option of the external table is set to FIXED
  - **LAYOUT** - A layout is an ordered collection of zone or field definitions
    - USE TYPE - Indicates whether a zone is a normal data zone, a reference zone, or a 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 NULLESS attribute
  - **RECORDLENGTH** - Specifies the length of the entire record

External Table – Cloud File Systems

- **SWIFT Storage**

```
CREATE EXTERNAL TABLE exttab1(a int)
  USING (dataobject 'datafile1.dat'
  swift('https://dal05.objectstorage.softlayer.net/auth/v1.0/',
  'XXX…456', 'b207…df16', 'my_dev' ) )
```

- **S3 Storage (AWS)**

```
CREATE EXTERNAL TABLE exttab2(a int)
  USIN (dataobject 'datafile2.dat'
  s3('s3.amazonaws.com',
  'XXX…456', 'bs07…df16', 'my_dev' ) )
```

- **IBM Cloud Cloud Object Store**

```
CREATE EXTERNAL TABLE exttab2(a int)
  USING (dataobject 'datafile2.dat'
  s3('s3-api.us-geo.objectstorage.softlayer.net',
  '1a2b…gYY9' 'my_dev' ) )
```

External Table – Cloud File Systems

- **Unloading data**

```
CREATE EXTERNAL TABLE 'order.tbl' USING (DELIMITER '|')
       AS SELECT * from orders;
CREATE EXTERNAL TABLE 'export.csv' USING (DELIMITER ',')
       AS SELECT foo.x, bar.y, bar.dt FROM foo, bar WHERE foo.x = bar.x;
```

- **Load data from an external table:**

```
INSERT INTO target
       SELECT * FROM EXTERNAL 'data.txt' USING (DELIMITER '|');
INSERT INTO orders
       SELECT * FROM EXTERNAL 'order.tbl'( order_num INT, order_dt TIMESTAMP)
       USING (DELIMITER '|');
```

- **Select data from an external table:**

```
SELECT * FROM EXTERNAL 'order.tbl' (order_num INT, order_dt TIMESTAMP)
       USING (DELIMITER '|');
SELECT * FROM EXTERNAL 'test.txt' LIKE test_table USING (DELIMITER ',');
SELECT x, y FROM EXTERNAL 'test.txt' ( x integer, y decimal(18,4) )
       USING (DELIMITER ',');
```

pureScale Enhancements

Db2 pureScale Feature Enhancements

- Improved re-use of free pages across Db2 members
- Simplified Geographically dispersed Db2 pureScale cluster (GDPC) configuration
- Improved CF recovery time
- Cross Invalidation Connections in RDMA Networks
- IBM Spectrum Scale changes to Db2 pureScale instances
- New registry variable supports enhanced security
- Improved castout performance
- Support for Currently Committed Locking

Db2 pureScale Currently Committed

- In pureScale environments, the Currently Committed isolation method is supported when:
  - The application performing the row read, and the application performing the row UPDATE or DELETE, reside on the same member
  - The application performing the row read, and the application performing row INSERTs, reside on any member
- In current pureScale environments, the Currently Committed isolation method has a significant restriction:
  - A row-reader application cannot benefit from currently committed semantics when the row is being updated or deleted (locked) by an application on a different member
- This feature brings completeness to Currently Committed behaviour in pureSacle environments, by relieving this restriction and allowing a row-reader application in CS isolation to retrieve the currently committed version of a row, even when that row is locked by an application on a different member

Monitoring and Workload Management

Simplifying Threshold Declaration

- The ACTIVITIES and ENFORCEMENT clauses on the CREATE THRESHOLD statement are now optional, leading to simpler declaration of THRESHOLD objects
- For example, you can now define and ACTIVITYTOTALTIME threshold using a statement like:

```
CREATE THRESHOLD TH1 FOR DATABASE WHEN ACTIVITYTOTALTIME > 1 MINUTE STOP EXECUTION
```

- Rather than:

```
CREATE THRESHOLD TH1 FOR DATABASE ACTIVITIES ENFORCEMENT DATABASE
       WHEN ACTIVITYTOTALTIME > 1 MINUTE STOP EXECUTION
```

Dropping Service Classes

- Dropping a disabled service class with the DROP statement now drops all dependent service subclases, thresholds, and work action sets.
- For example, to drop a service superclass A with three subclasses B, C, and D, and a work action set WAS1, you can simply run the following statements:

```
ALTER SERVICE CLASS A DISABLE DROP SERVICE CLASS A
```

- In previous releases of Db2, you would have to run the following statement to get the same result:

```
ALTER SERVICE CLASS B UNDER A DISABLE
DROP SERVICE CLASS B UNDER A
ALTER SERVICE CLASS C UNDER A DISABLE
DROP SERVICE CLASS C UNDER A
ALTER SERVICE CLASS D UNDER A DISABLE
DROP SERVICE CLASS D UNDER A
ALTER WORK ACTION SET WAS1 DISABLE
DROP WORK ACTION SET WAS1
ALTER SERVICE CLASS A DISABLE
DROP SERVICE CLASS A
```

Monitoring Enhancements

- New monitoring metrics for determining failure rate of SQL statements
- New monitor interfaces for workloads at the service superclass level
- Improved table functions provide access to information without having to perform joins
- Improved table functions provide direct access to WLM statistics
- New monitor element for concurrently executing activities
- New monitor elements report thread and memory usage for workload and service class objects

Monitoring – SQL Failure Rate

- It is now possible to monitor the failure rate of an SQL statement using the MON_GET_PKG_CACHE_STMT table function.
- The following columns have been added to MON_GET_PKG_CACHE_STMT
  - LAST_EXEC_ERROR: The Error SQLCODE for the last error of this statement
  - LAST_EXEC_ERROR_SQLERRMC: The SQLCODE tokens from SQLERRMC for the last error of this statement
  - LAST_EXEC_ERROR_TIMESTAMP: Time when the last error for this statement occurred
  - LAST_EXEC_WARNING: The Warning SQLCODE for the last warning of this statement
  - LAST_EXEC_WARNING_SQLERRMC: The SQLCODE tokens from SQLERRMC for the last warning of this statement
  - LAST_EXEC_WARNING_TIMESTAMP: Time when the last warning for this statement occurred
  - NUM_EXEC_WITH_ERROR: The number of executions of this statement that resulted in an error
  - NUM_EXEC_WITH_WARNING: The number of executions of this statement that resulted in an warning

Monitoring – Aggregation Across all Service Subclasses

- It is now possible to examine monitor data without having to manually aggregate data across all service subclasses within a service superclass.
- The following new table functions and logical data groups in the statistics event monitor allow you to monitor your workload at the service superclass level
- Table Functions:
  - MON_GET_SERVICE_SUPERCLASS
  - MON_GET_SERVICE_SUPERCLASS_DETAILS
  - MON_GET_SERVICE_SUPERCLASS_STATS
- Statistics event monitor logical data groups:
  - SUPERCLASSSTATS
  - SUPERCLASSMETRICS

Monitoring – Simpler Monitoring

- The addition of the APPLICATION_NAME monitor element to MON_GET_ACTIVITY, MON_GET_ACTIVITY_DETAILS, MON_GET_UNIT_OF_WORK and MON_GET_UNIT_OF_WORK_DETAILS allows you to determine the application name for a connection or statement without the need to join your monitor query with MON_GET_CONNECTION
- Similarly the addition of WORKLOAD_ID to MON_GET_ACTIVITY and MON_GET_ACTIVITY_DETAILS allows you to view the WORKLOAD for the connection running an activity without having to join your monitor query with other interfaces.
- Both of these changes promote simpler use of monitoring by removing the need for joins in some monitor queries.

Monitoring – Easier Access to Statistics

- The addition of several monitor elements to the MON_GET_SERVICE_SUBCLASS_STATS and MON_GET_WORKLOAD_STATS table functions allows you to use the table functions to access all WLM statistics through a table function interface
- The enhanced table functions now report information that was previously available only in the statistics event monitor tables
  - CPU_ENTITLEMENT, CPU_SHARE_TYPE, ACT_CPU_TIME_TOP
  - ACT_ROWS_READ_TOP
  - COORD_ACT_EST_COST_AVG, COORD_ACT_EST_COST_STDDEV
  - COORD_ACT_INTERARRIVAL_TIME_AVG
  - COORD_ACT_INTERARRIVAL_TIME_STDDEV
  - COST_ESTIMATE_TOP, ROWS_RETURNED_TOP
  - TEMP_TABLESPACE_TOP
  - AGG_TEMP_TABLESPACE_TOP
  - CONCURRENT_CONNECTION_TOP

Monitoring – Easier Access to Statistics

- The addition of several monitor elements to the MON_GET_SERVICE_SUBCLASS_STATS and MON_GET_WORKLOAD_STATS table functions allows you to use the table functions to access all WLM statistics through a table function interface
- The enhanced table functions now report information that was previously available only in the statistics event monitor tables
  - CPU_ENTITLEMENT, CPU_SHARE_TYPE, ACT_CPU_TIME_TOP
  - ACT_ROWS_READ_TOP
  - COORD_ACT_EST_COST_AVG, COORD_ACT_EST_COST_STDDEV
  - COORD_ACT_INTERARRIVAL_TIME_AVG
  - COORD_ACT_INTERARRIVAL_TIME_STDDEV
  - COST_ESTIMATE_TOP, ROWS_RETURNED_TOP
  - TEMP_TABLESPACE_TOP
  - AGG_TEMP_TABLESPACE_TOP
  - CONCURRENT_CONNECTION_TOP

Workload Management Enhancements

- New monitor elements are introduced to report thread (parallelism) and memory usage
- New monitor elements for MON_GET_WORKLOAD, MON_GET_SERVICE_SUPERCLASS and MON_GET_SERVICE_SUBCLASS table functions
  - EFF_PARALLELISM – total effective query degree for all currently executing activities
  - ACTUAL_PARALLELISM – total actual query degree for all currently executing activities
  - SORT_SHRHEAP_UTILIZATION – current shared sort memory usage expressed as a percentage of *sheapthres_shr*

Workload Management Enhancements

- New monitor elements for MON_GET_WORKLOAD_STATS, MON_GET_SERVICE_SUPERCLASS_STATS and MON_GET_SERVICE_SUBCLASS_STATS
  - EFF_PARALLELISM_TOP – peak effective parallelism for all currently executing activities
  - EFF_PARALLELISM_AVG – average effective parallelism for all currently executing activities
  - ACTUAL_PARALLELISM_TOP – actual query degree for all currently executing activities
  - ACTUAL_PARALLELISM_AVG – average query degree for all currently executing activities
  - SORT_SHRHEAP_UTILIZATION_TOP – peak shared sort memory usage expressed as a percentage of *sheapthres_shr*
  - SORT_SHRHEAP_UTILIZATION_AVG – average sort memory usage expressed as a percentage of *sheapthres_shr*

Client Enhancements

IBM Data Server Clients and Drivers Enhancements

- CLI (Call-level Interface) Update
- IBM Data Server Provider for .NET enhancements
- IBM Data Server Driver for JDBC and SQLJ enhancements

Call-Level Interface Driver Enhancements

- Data movement using cliload with an External Table
  - cliload now has the capability to trigger load through an External Tables interface
- Generate unique utility id for zload
  - Enhance the Utility ID generation for zLoad
- Db2dsdriver.cfg support in MS ODBC Administrator
  - Enable reading and writing in Db2dsdriver.cfg from MS ODBC Administrator
- Enhancement to the WLB/ACR algorithm
  - The WLB algorithm is enhanced for improved workload distribution
  - Route to the group IP address in case of failover
- Server side enablement of WLB/ACR for Db2 pureScale server
  - WLB/ACR can be triggered by setting the Member subset attribute in the server

.NET Enhancements

- Db2 .Net EF Core
  - Code First support for EF Core
- Db2 .Net EF Core
  - Support for Microsoft Entity Framework 2.1
- IBM Data Server Driver for ADO.Net
  - Support for Code First on Microsoft Azure using EntityFramework 6.0
- Visual Studio Code Extension for Db2
  - Support for Visual Studio Code Extension for Db2

JDBC and Java Enhancements

- Server side enablement of WLB/ACR for Db2 pureScale server
  - WLB/ACR can be triggered by setting the Member subset attribute in the server
- Various enhancements are added to the driver to simplify the usage of clientApplcompat
  - Command line input option in db2connectactivate tool for clientApplcompat
  - Read clientApplcompat and currentPackageSet value through db2dsdriver.cfg file in CLPPlus
  - Support clientApplcompat and currentPackageSet value through the db2dsdriver.cfg file
- The default behaviour of the timerLevelForQueryTimeOut property changed to QUERYTIMEOUT_CONNECTION_LEVEL(2)
  - The default timer level for query timeout changed from Statement to Connection
  - This will improve performance
- Support for named parameter markers with (var1=>?,var2=>?,... syntax added in java.sql.CallableStatement for Db2 for z/OS
  - java.sql.CallableStatement supports (var1=>?,var2=>?,...) syntax for Db2 for z/OS
  - It is already supported for Db2 LUW

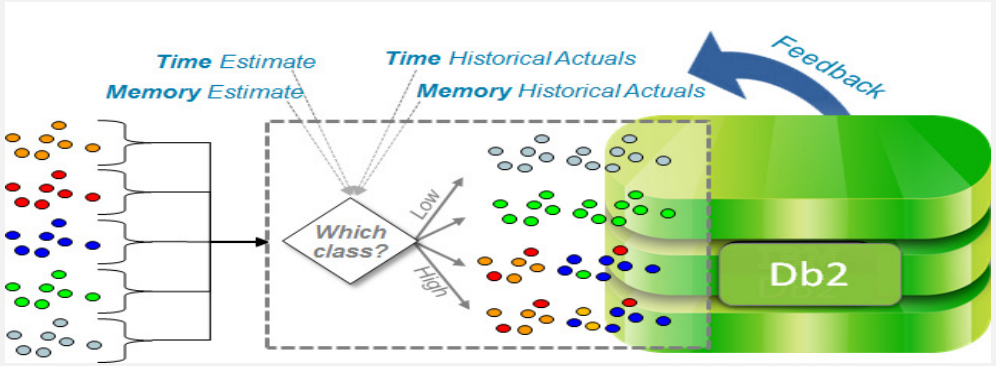
Additional Features Under Consideration

Db2 Adaptive Workload Management

Future Release

Ensures system stability and responsiveness with zero tuning

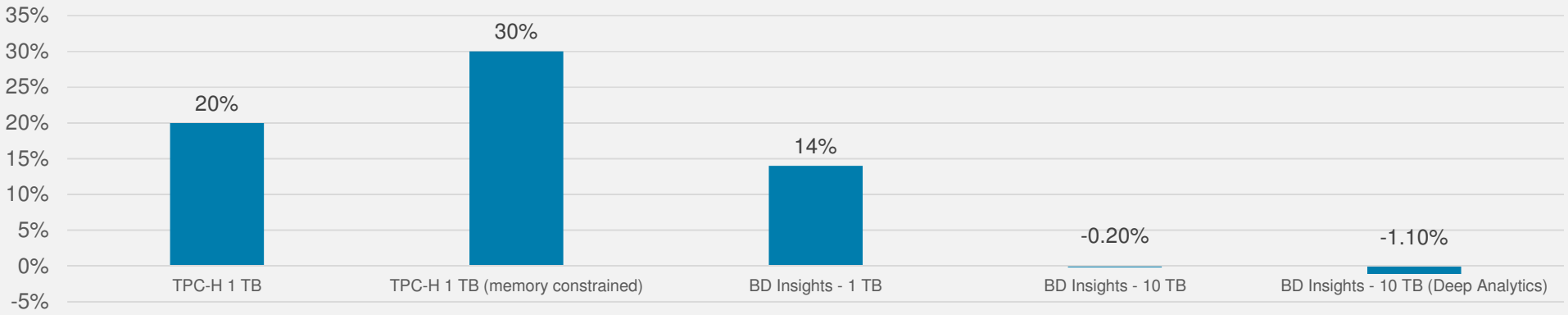
- **Adaptive Workload Management** is the “next generation” of Db2 workload management (WLM)
- Delivers automatic workload management within Db2 that ensures system stability and responsiveness with zero tuning
  - Make sure that the system is well-utilized but don’t let it become overcommitted
  - Schedule jobs appropriately to ensure fairness and appropriate responsiveness



Adaptive WLM Performance Numbers

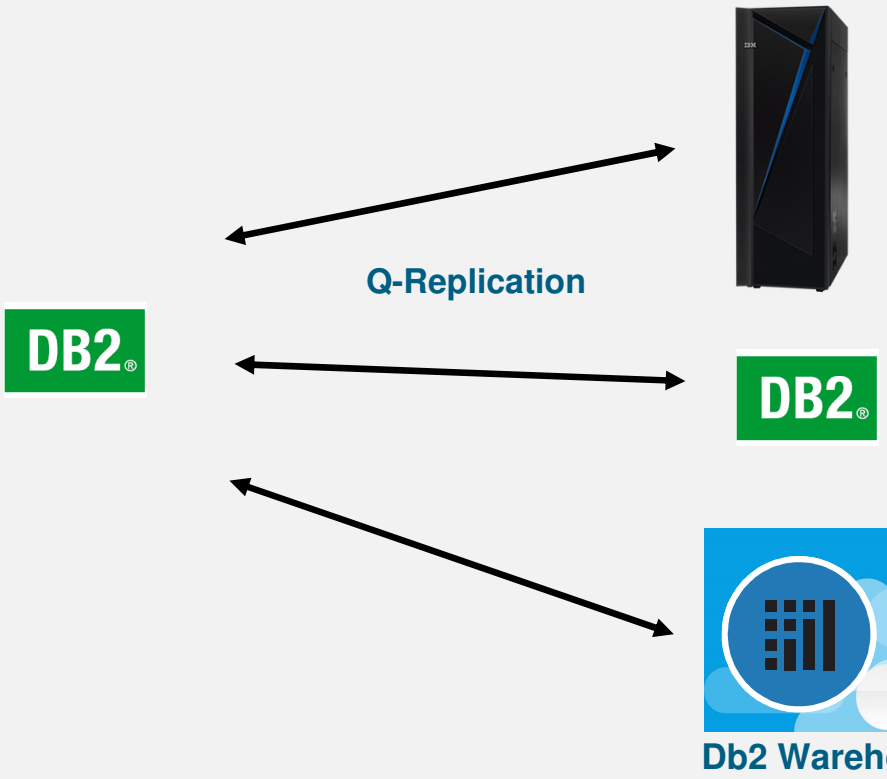
Future Release

- Analytical workloads (BLU) performed at par or better with Adaptive WLM compared to default WLM configuration
  - Since work is admitted into the system based on matching of CPU and memory needs versus current usage, processing of work is generally more efficient due to smoother, steadier consumption patterns
  - System stability is maintained throughout



Disaster Recovery and Replication Solution

Future Release

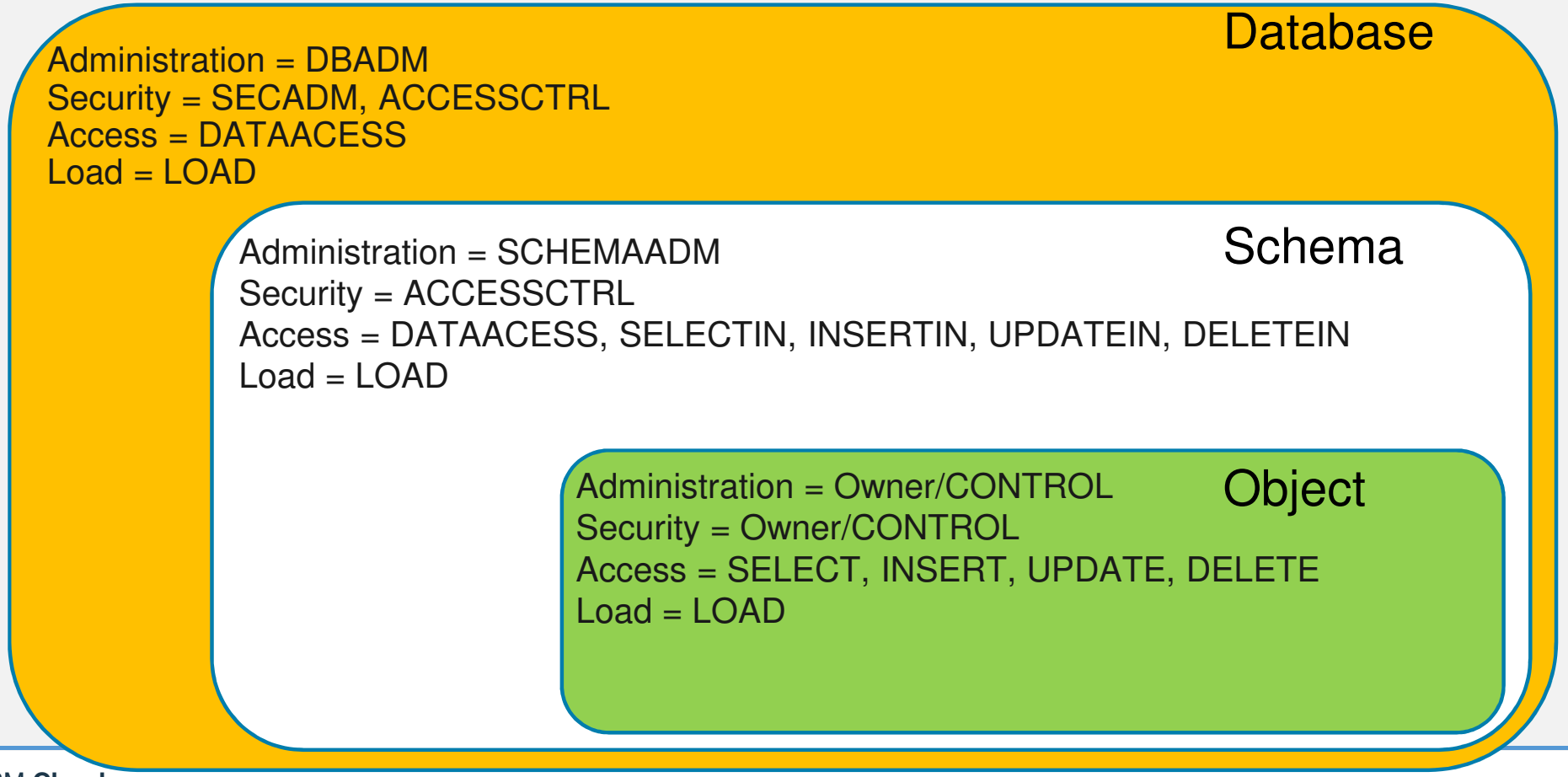


- ✓ Multiple stand-bys
- ✓ R/W Active-Active
- ✓ Bi-directional R/W Active-Active
- ✓ Time Delay
- ✓ Sub-setting
- ✓ Automatic Client Reroute

- ✓ Support for columnar
- ✓ Reduced logging is automatically handled

Schema Level Authorization

Future Release



Other Features

Future Release

- Adaptation and Integration of new technologies into the Db2 common SQL engine:
  - Leveraging Machine Learning (ML) as part of Db2 optimization
  - Exploring Natural Language Query (NLQ) within Db2

- Focus on making life better for application developers work with Db2
  - Introduction of support for new languages
  - Introduction of server side REST API support
  - Better integration with common developer environments

- Completion of JSON ISO SQL function enhancements
- Support for triggers on columnar tables
- Support for system maintained temporal columnar tables

Reduced Logging for Columnar Tables

Future Release

- An enhanced solution to the same problem addressed by “Not Logged Initially” (NLI) tables
  - Inserting large amounts of data into a table without swamping the transaction logs
  - Table contents are preserved during rollback, crash recovery, and roll-forward recovery to END of online backup
- Unlike NLI, reduced logging is not implemented as a table property
  - It is associated with any operation driving a large amount of insert work against a table and kicks in automatically after a certain amount of insert work has already occurred
  - Relevant operations include: INSERT, INGEST, MERGE, and UPDATE statements as well as CREATE TABLE AS DDL, etc.
- With this behaviour active, very large INSERTs are possible without running out of log space
  - Up to 95% less data logged!

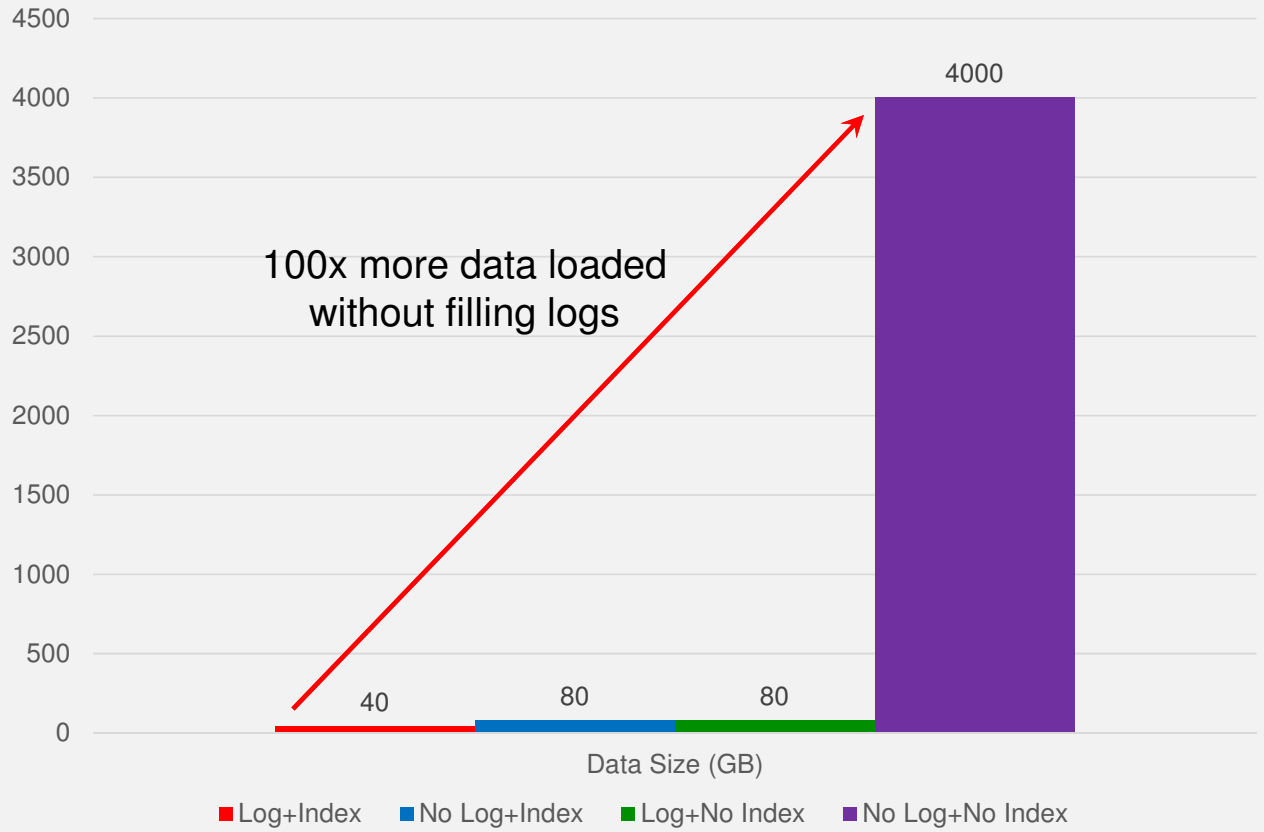
Reduced Logging Fine Print

Future Release

- Reduced Logging Restrictions
  - No reduced logging for ROW ORGANIZED tables
  - No reduced logging for indexes
  - Enforced primary key/unique key constraints reduce effectiveness
- Test Environment
  - The results show how much data can be inserted in a single transaction before hitting a log full condition
  - These numbers are just an average, since each case depends on actual customer data and compression rates
- **Without Reduced Logging** & **With Enforced Constraint**: 40Gb
- **With Reduced Logging** & **With Enforced Constraint**: 80Gb (reduced logging benefit)
- **Without Reduced Logging** & **Without Enforced Constraint:** 80Gb (no index logging)
- **With Reduced Logging** & **Without Enforced Constraint**: 4TB of raw data inserted!

Data Load Size

Future Release



DBMS_LOCK Module

Future Release

- **ALLOCATE_UNIQUE (** *lockname, lockhandle, expiration_secs* **)** – allocates a unique lock handle for a specified lock name
- **SLEEP (** *seconds* **)** – suspends execution of the current session for a specified amount of time
- **REQUEST (** *id, lockhandle, lockmode, timeout, release_on_commit***)** – requests a lock
- **CONVERT (** *id, lockhandle, lockmode, timeout* **)** – converts the current lock mode to a new one
- **RELEASE (** *id* **)** – releases a lock that was previously acquired

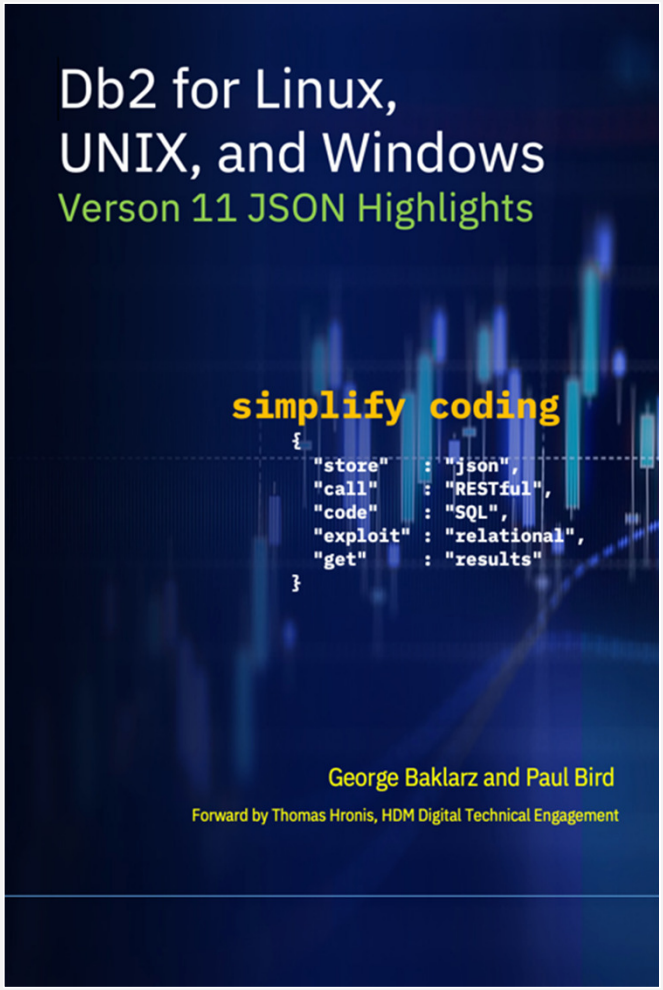
Summary

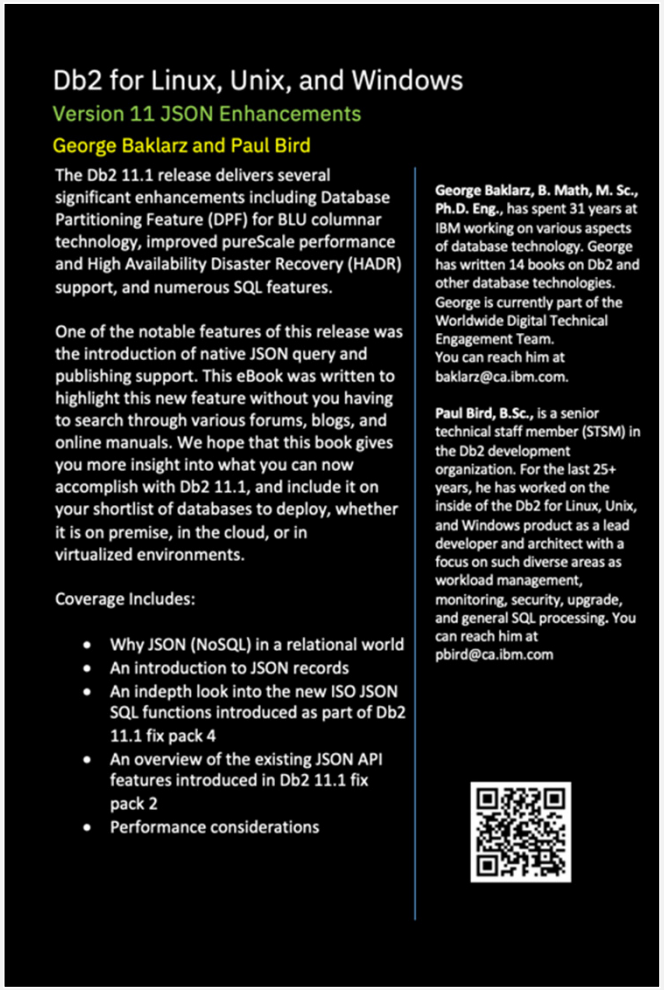
Db2 11.x Highlights

- Performance enhancements
- Compression and Space Reclamation Enhancements
- SQL compatibility enhancements
- Db2 pureScale Feature enhancements
- Workload management enhancements
- Monitoring enhancements
- IBM data server clients and drivers enhancements

Additional Resources

- Read the new Db2 JSON Book
  - ibm.biz/db2json
- Visit the Digital Technical Engagement Site
  - The Digital Technical Engagement group (DTE) provides videos, product tours, and product labs for you to try out technology at your leisure
  - The product labs are fully functional servers that are provisioned for you
  - These servers contain the base products (Db2) along with self-paced examples
  - The Db2 product lab contains Jupyter notebooks which demonstrate new SQL features
  - https://www.ibm.com/cloud/garage/dte/tutorial/modern-application-development-db2
- GitHub Db2-Samples
  - There are a number of Db2 sample programs available on GitHub
  - If you have a Docker environment available, or are using Jupyter notebooks, then the following repository may be of interest
  - https://github.com/DB2-Samples/db2jupyter


Db2 for Linux,
UNIX, and Windows
Verson 11 JSON Highlights
simplify coding
{
"store" : "json",
"call" : "RESTful",
"code" : "SQL",
"exploit" : "relational",
"get" : "results"
}
George Baklarz and Paul Bird
Forward by Thomas Hronis, HDM Digital Technical Engagement


Db2 for Linux, Unix, and Windows
Version 11 JSON Enhancements
George Baklarz and Paul Bird
The Db2 11.1 release delivers several significant enhancements including Database Partitioning Feature (DPF) for BLU columnar technology, improved pureScale performance and High Availability Disaster Recovery (HADR) support, and numerous SQL features.
One of the notable features of this release was the introduction of native JSON query and publishing support. This eBook was written to highlight this new feature without you having to search through various forums, blogs, and online manuals. We hope that this book gives you more insight into what you can now accomplish with Db2 11.1, and include it on your shortlist of databases to deploy, whether it is on premise, in the cloud, or in virtualized environments.
Coverage Includes:
Why JSON (NoSQL) in a relational world
An introduction to JSON records
An indepth look into the new ISO JSON SQL functions introduced as part of Db2 11.1 fix pack 4
An overview of the existing JSON API features introduced in Db2 11.1 fix pack 2
Performance considerations
George Baklarz, B. Math, M. Sc., Ph.D. Eng., has spent 31 years at IBM working on various aspects of database technology. George has written 14 books on Db2 and other database technologies. George is currently part of the Worldwide Digital Technical Engagement Team.
You can reach him at baklarz@ca.ibm.com.
Paul Bird, B.Sc., is a senior technical staff member (STSM) in the Db2 development organization. For the last 25+ years, he has worked on the inside of the Db2 for Linux, Unix, and Windows product as a lead developer and architect with a focus on such diverse areas as workload management, monitoring, security, upgrade, and general SQL processing. You can reach him at pbird@ca.ibm.com