

IBM Db2

A Technology Review and Update

George Baklarz

May 8th, 2018

simplify coding

```
{  
  "store"    : "json",  
  "call"     : "RESTful",  
  "code"     : "OData",  
  "exploit"  : "relational",  
  "get"      : "results"  
}
```

Disclaimer

© IBM Corporation 2018. All Rights Reserved.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication 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.

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.

Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. 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.

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.

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.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.

[illegible]

Agenda

- Introduction
- Developer Community Edition
- Fixpak 2 and 3 Highlights
- JSON Technology Update
- OData
- Eventstore
- DTE Assets

V11.1
24x7 for
Everyone, MPP
Scale for BLU,
and more !

June 2016

**V11.1.1.1 &
V11.1.2.2**
Advances in
HTAP, Analytics,
Performance
and more !

*Dec 2016
June 2017*

**V11.1.3.3 and
Beyond !**
HTAP,
Availability, and
more !

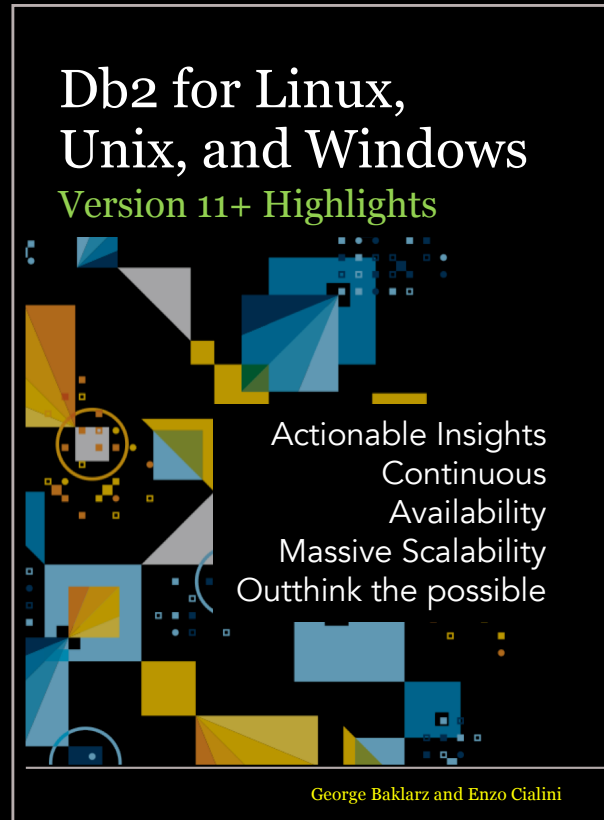
March/2018

**V11.1.4.4 and
Beyond !**

Next generation
advances in
HTAP, JSON,
Availability, and
more !

*2H/2018 and
Beyond*

Db2 V11+ eBook



ibm.box.com/v/DB2v11eBook

IBM Db2 Developer Community Edition

▪ New Db2 Developer-C Edition

- Freely downloadable just like Express-C
- **Fully functional Db2 database server**
 - pureScale and DPF deployments
 - Compression and BLU Acceleration

▪ Use Limitations

- For development and **production**
- **Unsupported edition (non-warranted)**
- Environment limited usage:
 - 4 cores, 16GB of memory, 100GB of data in user tablespaces

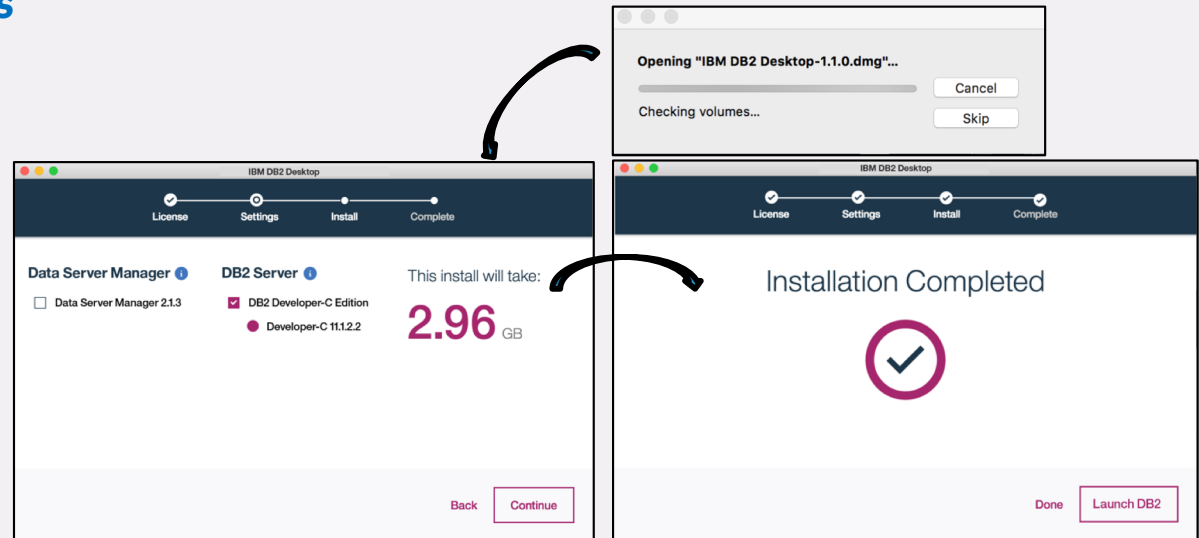
▪ Functional Differences from Db2 Developer Edition

- Excludes some “Supporting Programs”
 - Cognos, IDA, WebSphere AS / MQ, DSM Enterprise

A Simple “Download and Go” experience!

Up, Running, and Productive in Minutes

- **Marketplace or Discovery Portal – single point of download**
- **Download platform-specific executable (Mac, Windows, or Linux)**
- **Small download, less than 15 minutes from “I want” to “I got”**
- **3 clicks and userID/pwd input only**



Unifying Database Systems for Data Access

Using SQL to Unify Access to a Variety of Data Sources

IBM Db2

Common Analytics Engine

Federation Database SQL queries

BigSQL queries

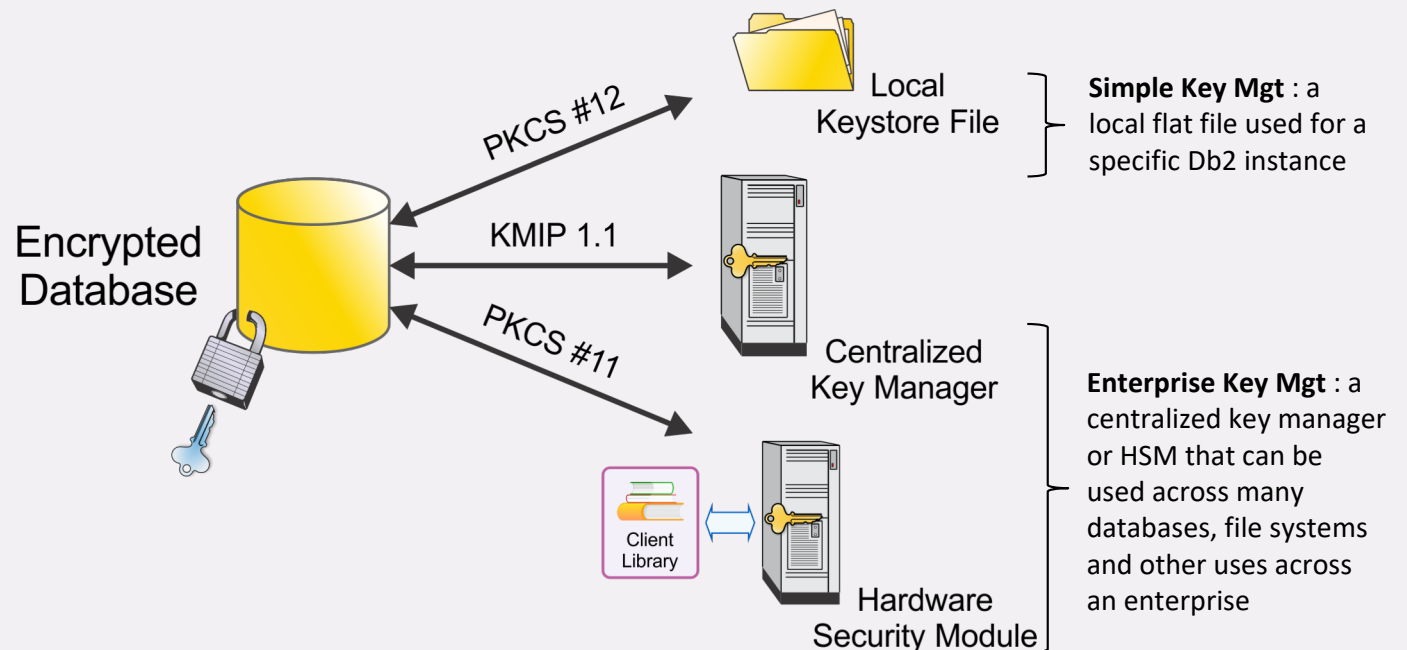


Encryption Included to Protect Your Data

Ensuring your Data stays Safe

- **All user data in a database is encrypted**

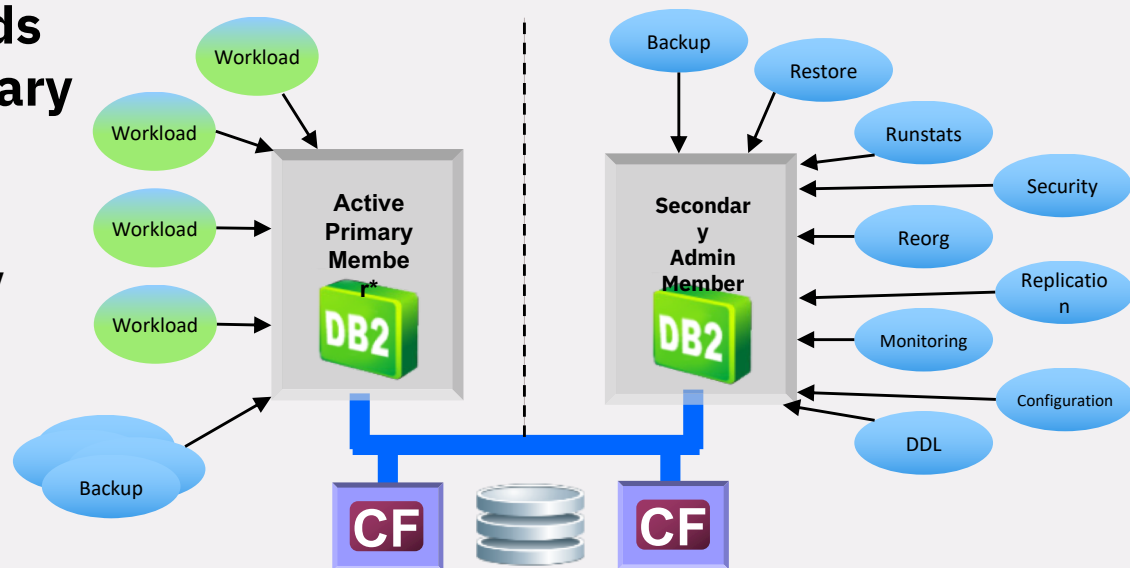
- Tables
- All data types
- Transaction and Archive logs
- LOAD COPY data
- Staging files
- Diagnostic data
- Backup images



High Availability and Disaster Recovery

Keeping your Data Available at all Times

- **A Variety of HA and DR options**
 - Cold, Warm, Hot
- **All application workloads are directed to the primary active member**
- **Utilities and admin tasks allowed on the secondary member**

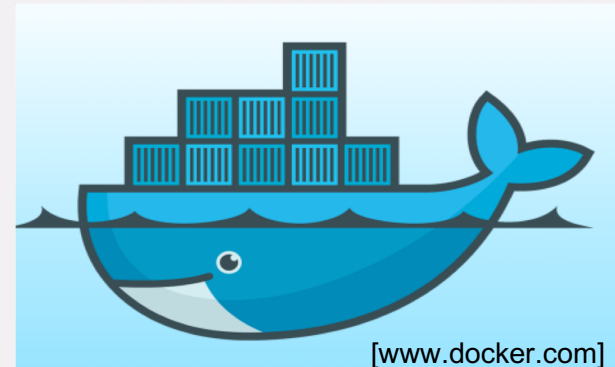


IBM Db2 Developer Community Edition - Requirements

- **8GB minimum memory**
- **30GB minimum space**
 - C:\ on Windows
 - /Users/<username> on Mac
 - ~/.config on Linux
- **OS Required:**
 - Windows 7 Professional, Windows 10 Professional/Enterprise/Home
 - AMD/VT-X must to be enabled in the BIOS and it will try to enable Hyper-V in Windows 10
 - Will require a reboot if the above features need to be enabled
 - Mac OS X (10.12+)
 - Linux
 - Distros with kernel level less than 4.11 should work (Fedora 25, CentOS 1611, SLES12, Ubuntu 16.04+)
- **For OS other than the ones tested, we display a warning that we have not tested it so continue on your own risk**
 - Pressing continue allows to proceed but who knows what happens next

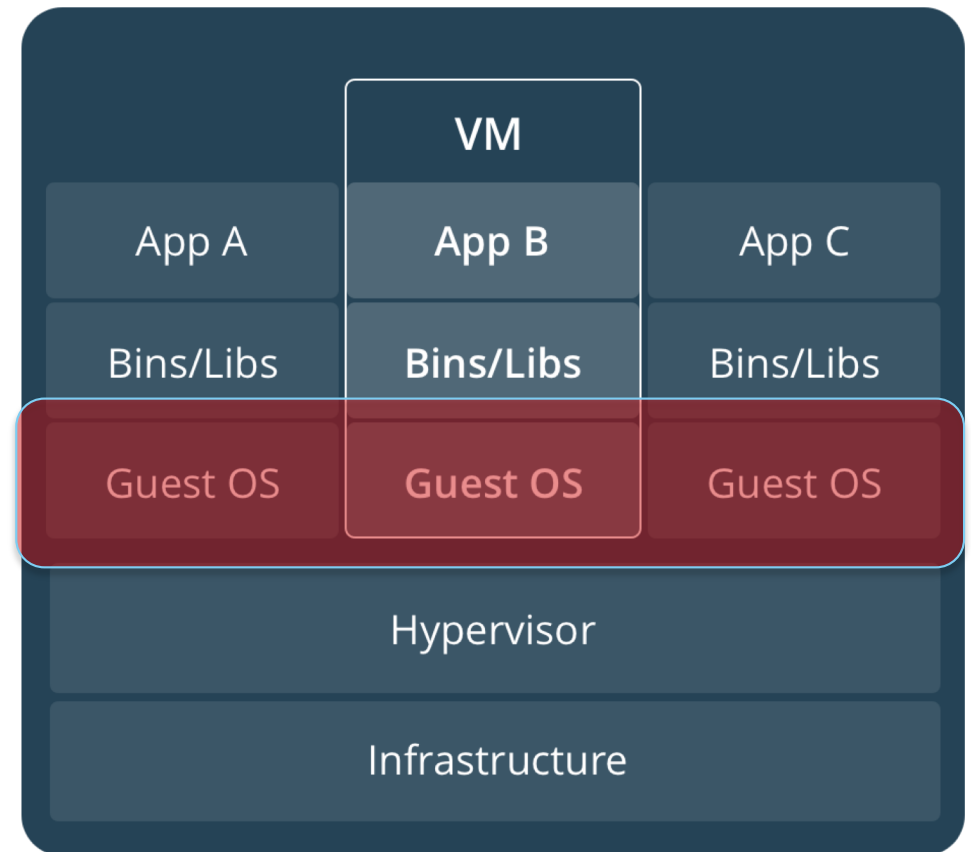
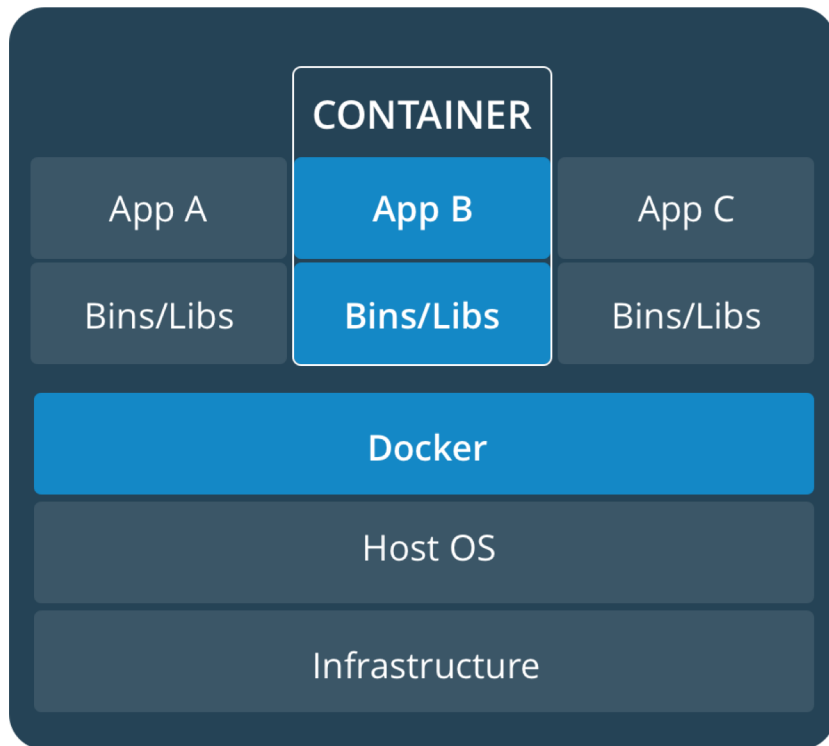
Docker Highlights

- **Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system–level virtualization on Linux**
- **Docker enables you to separate your applications from your infrastructure so you can deliver software quickly**



Comparing Docker to a Hypervisor (VMWare, Virtual Box)

IBM Cloud



Docker Components

■ Docker Objects

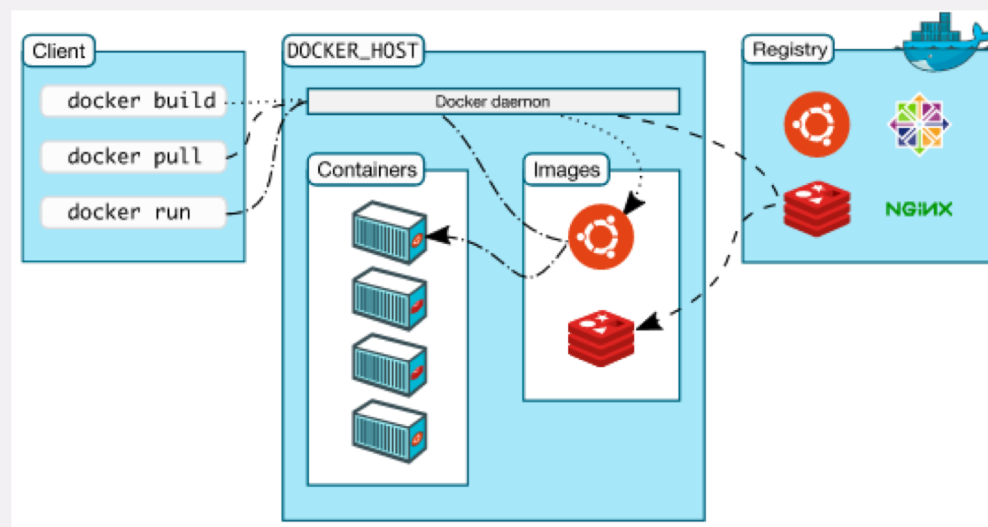
- Image: the packaging of functionality. Can be created from other images.
- Container: an instance of an image (May contain multiple processes)
- Service: An application divided in multiple containers that can be deployed on a cluster (Docker swarm)
- Network: Network definition
- Volume: storage on the host

■ Docker architecture

- Daemon
- Client
- Registries

■ Docker Hub

- Repository of available Docker Images



Evolutionary Enhancement to Db2

Db2 Version 11.1 for Linux, UNIX, and Windows English Manuals


 Be the first to ask a question  Ask the IBM Support Agent Tool
















Product documentation

Abstract

This page contains a list of the Db2® Version 11 English documentation in Portable Document Format (PDF).

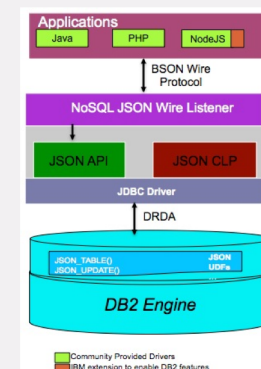
Content

To download a PDF, right-click the PDF icon () in the following table and specify where to save the file; or save the file while using a PDF viewer. The latest version of the Db2 documentation is in [Knowledge Center](#), as such, the PDFs on this page may not contain the most recent changes that are available.

English PDF Guide	Link	Date Posted
Db2 Fundamentals		
System Monitor Guide and Reference		December 2017
Database Security Guide		December 2017
Performance Tuning		December 2017
Administration		
Data Movement Utilities Guide and Reference		April 2018
Partition and Clustering Guide		April 2018
Text Search Guide		April 2018
Data Recovery and High Availability Guide and Reference		April 2018
Application Development		
Developing Embedded SQL and XQuery Database Applications		December 2017
SQL Applications		December 2017
SQL Procedural Languages Support		December 2017
Developing Perl, PHP, Python, and Ruby on Rails Applications		April 2018
Reference		
API Reference		December 2017
Built-In Modules Reference		December 2017
Built-In Routines and Views		December 2017
Command Reference		December 2017
Compatibility Features for Oracle		December 2017
Db2 CI Application Development		December 2017
Developing Routines		December 2017
SQL Reference		December 2017
Troubleshooting and Support		February 2018
Registry and Environment Variables		April 2018
Configuration Parameters		April 2018
CLPPlus Reference		April 2018
Installation		
Upgrading to Db2 Version 11.1		April 2018
Back to top		

Db2 JSON Support

- **Initial JSON support in Db2 10.5 FP1 consisted of:**
 - IBM NoSQL Wire Listener, JSON Java APIs, JSON command line
- **Db2 11.1.2.2 added full support for direct SQL access to underlying JSON interfaces used by externals above**
 - Integrated installation/configuration during database creation or migration update so no need to catalog JSON functions manually
 - High-level documentation on their existence similar to what Db2 for z/OS already provides
 - JSON2BSON, BSON2JSON are catalogued as part of SYSIBM schema
 - JSON_VAL is function built into the SQL syntax
 - All other JSON functions are part of the SYSTOOLS schema
 - Overview of all functions and how to use them can be found in Db2 11 ebook at:
ibm.box.com/v/DB2v11eBook
 - Manually catalog tables with prior releases using the DB2-V11-JSON-Examples.sql script in the eBook directory



JSON SQL

Combine your SQL with JSON records

Name	Comments
BSON2JSON	Convert BSON formatted document into JSON strings
BSON_VALIDATE	Checks to make sure that a BSON field in a BLOB object is in a correct format
JSON2BSON	Convert JSON strings into a BSON document format
JSON_GET_POS_ARR_INDEX	Find a value within an array
JSON_LEN	Returns the count of elements in an array type inside a document
JSON_TABLE	Returns a table of values for an array field
JSON_TYPE	Returns the data type of a specific field within a JSON document
JSON_UPDATE	Update a field or document using set syntax
JSON_VAL	Extracts data from a JSON document into SQL data types

```
{
  "empno": "000070",
  "firstnme": "EVA",
  "midinit": "D",
  "lastname": "PULASKI",
  "workdept": "D21",
  "phoneno": [7831, 1422, 4567],
  "hiredate": "09/30/2005",
  "job": "MANAGER",
  "edlevel": 16,
  "sex": "F",
  "birthdate": "05/26/2003",
  "pay": {
    "salary": 96170.00,
    "bonus": 700.00,
    "comm": 2893.00
  }
}
```

pureScale Multi-host Maintenance Mode

- **In 11.1.2.2 (and earlier) you have 2 options:**

- Single host in maintenance mode
- ALL hosts (cluster outage) in maintenance mode
- If you have an issue resulting in a member in maintenance mode for a long time (e.g. hardware issue), it will prevent any other host going into maintenance mode, including for a Rolling Update.

- **As of 11.1.3.3 you can now:**

- Enable multiple hosts to be in maintenance mode concurrently
- Enter maintenance on hosts one at a time until there is only one host (with at least a CF, and preferably a member - even if 2 hosts) left in the cluster

Add / Drop CF

- **Db2 already supports the ability to ADD a member online**

- **As of 11.1.3.3 you can now:**

- Add a CF online
- Drop a CF online

Note: you cannot online drop the primary if the secondary is not in PEER state

- **Expected use cases:**

- Move CF to a new host
- Remove a failed or maintenance-mode CF
- Improve HA of a single-CF cluster by adding a second CF

pureScale Online CREATE INDEX (concurrent write access)

▪ Online CREATE INDEX with concurrent write access now in pureScale

```
db2set DB2_CREATE_INDEX_ALLOW_WRITE=ON  
db2 connect to MYDB  
db2 create index ....
```

- Default is **ON** for non-pS, **OFF** for pS. Only affects a recoverable database (logarchmeth1 or logarchmeth2 database configuration parameters set).
- Relies on log merge and replay to ensure no index changes are missed!
- Concurrent write access not supported for expression-based indexes in pS

Additional pureScale Diagnostics

- Chrony automatically setup in favor of NTP if it exists when Db2 install is run

- Ability to list hosts in maintenance mode

```
db2cluster -verify -req -rdma_ping
```

https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0056712.html

- Single command for pS perf data collection

```
As root: db2cluster -cm -verify -maintenance -all
```

```
db2cluster -verify -perf -db2mon -collect -db < db name > -interval <value in seconds>
```

Technote: <http://www-01.ibm.com/support/docview.wss?uid=swg22014557>"

Availability enhancements

▪ HADR Tablespace Recovery

- If the standby database has a tablespace in an invalid or error state, a TAKEOVER by the Standby will not be fully available to applications
- Prior to Db2 11.1.2.2, the entire Standby database needed to be recovered
- In 11.1.2.2, the tablespace can be recovered online without a full database refresh
- Tablespace recovery been retrofitted into Db2 10.5fp9+
- Details: <http://www-01.ibm.com/support/docview.wss?uid=swg21993389>

▪ Restore REBUILD in pureScale

- Prior to 11.1.2.2, pureScale required a database backup in order to perform a database recovery
- Functionality available today in non-pureScale deployments
- In 11.1.2.2, pureScale can now recover a database from a set of tablespace backups

Monitoring HADR Tablespace Status

▪ Tablespace Error State

- When a tablespace is in an invalid or error state on the Standby database, the **HADR_FLAGS** field will display the value **STANDBY_TABLESPACE_ERROR**

▪ Monitoring with db2pd

- The **HADR_FLAGS** field can be monitored by using the **db2pd -hadr** command on the Primary or Standby database

```
db2pd -db HADRDB1 -hadr
... HADR_FLAGS = STANDBY_TABLESPACE_ERROR TCP_PROTOCOL ...
```

▪ Monitoring with Table Function

- The **MON_GET_HADR()** table function will display the current status on either the Primary database or Standby database with Reads on Standby enabled

```
SELECT STANDBY_ID, HADR_FLAGS FROM
       TABLE (MON_GET_HADR(NULL) )

STANDBY_ID
-----
1 STANDBY_TABLESPACE_ERROR TCP_PROTOCOL
```

Export of TSAMP config to XML File

▪ db2haicu has a new "-o" option

- Used to facilitate rapid HA configuration backup for fast redeployment.
- Supported for HADR, DPF, and single-partitioned (non-DPF) environment.

- Run db2haicu to create XML file,

```
Original_system> db2haicu -o primary.xml
Welcome to the DB2 High Availability Instance Configuration Utility (db2haicu). db2haicu determined the current DB2
database manager instance is 'db2inst1'. The cluster configuration that applies to this instance will be exported. All
cluster configurations have been exported successfully. db2haicu exiting.
```

- edit XML file for correct local/remote Host

If the TSAMP configuration exported from the primary host is used to import the configuration on both primary and standby hosts, the following fields for the HADRDBSet element will depend on the host where import is being performed. The localhost must refer to the host on which the import is being run and remote who it connects to.

```
<HADRDBSet><HADRDB databaseName="SAMPLE" localInstance="amahadev" remoteInstance="amahadev" localHost="ceha04" remoteHost="ceha03"
monitorDataMounts="false"/> <VirtualIPAddress baseAddress="9.26.54.119" subnetMask="255.255.255.0" networkName="db2_public_network_0"/>
</HADRDBSet>
```

- run db2haicu on new system(s), edit as needed.

```
new_system> db2haicu -f primary.xml
```

Availability & Security Enhancements

- **HADR Add resilience for network transmission data corruptions. On by default!**
- **Add Timeout to Log-Archive/Retrieve Requests from Vendor API**

`http://www-01.ibm.com/support/docview.wss?uid=swg22008073`

- **Improve identification of HADR ROS 'replay-only-windows'. Default=1**

For logarchopt1/2 you can set: `--vendor_archive_timeout=<seconds>`
`http://www-01.ibm.com/support/docview.wss?uid=swg21670158`

- **SSL for HADR for all platforms (previously Linux/x86 only)**

Additional diagnostic messages report the operation that causes the replay-only window
When the LOG_DDL_STMTS database configuration parameter is set to YES, the DDL statement text is included

BLU Index Support

- **BLU acceleration provides superior performance for analytic workloads**
- **Db2 extends BLU capabilities for hybrid use cases with indexes**
- **Improves performance for a range of applications:**
 - Data warehouse ETL with OLTP characteristics
 - Operational analytics
 - “Light” OLTP
 - Analytics
 - Only used in more limited situations
 - Analytic queries with transactional characteristics

BLU Index Support

- Provides the ability to create indexes on BLU tables and leverage them in queries

- `CREATE TABLE TEST.T1 (C1 INTEGER NOT NULL, C2 INTEGER, C3 INTEGER)
ORGANIZE BY COLUMN
CREATE INDEX IDX1 ON TEST.T1 (C1)`

- The following **CREATE INDEX** options are supported

- Unique and non-unique indexes
- Nullable and non-nullable columns
- COLLECT STATISTICS clause
- ALLOW/DISALLOW REVERSE SCANS
- MINPCTUSED clause
- PCTFREE clause
- PAGE SPLIT clause
- LEVEL2 PCTFREE
- COMPRESS
- WITH SPECIFICATION ONLY clause (applies to nicknames)
- Optimization guidelines (aka “hints”)

- **ALTER INDEX** is also supported

```
Rows
RETURN
(1)
Cost
I/O
|
1
FETCH
(2)
17.0787
1
/-----+-----\
1          98168
IXSCAN CO-TABLE: VICCHANG
(3)      /BIC/SZCCUST
6.91242  Q1
1
+
98168
INDEX: VICCHANG
/BIC/SZCCUST~0
Q1
```

The Explain plan for column-organized FETCH is similar to that of row-organized FETCH

Multi-Core Parallelism – BLU INSERT

- **Db2 has exceptional multi-core scalability for queries:**

- Combined MPP and SMP parallelism
- Includes columnar and row-based tables
- Applies to sub-select of INSERT

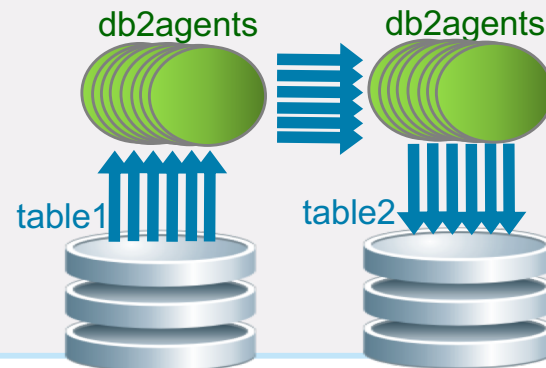
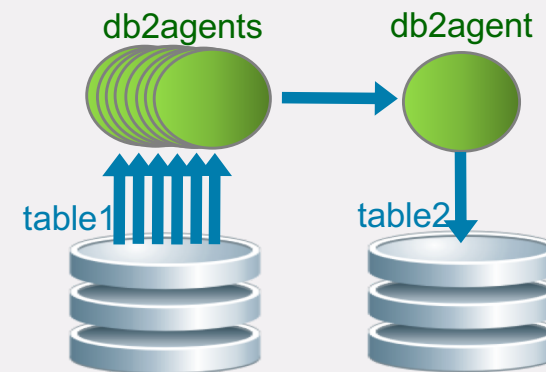
- **New Multi-Core Parallelism for INSERT**

- Includes a variety of data population methods (i.e., INSERT, INGEST, etc.)
- Extends to index maintenance
- Also applicable to MPP deployments

- **Very significant reduction in time for:**

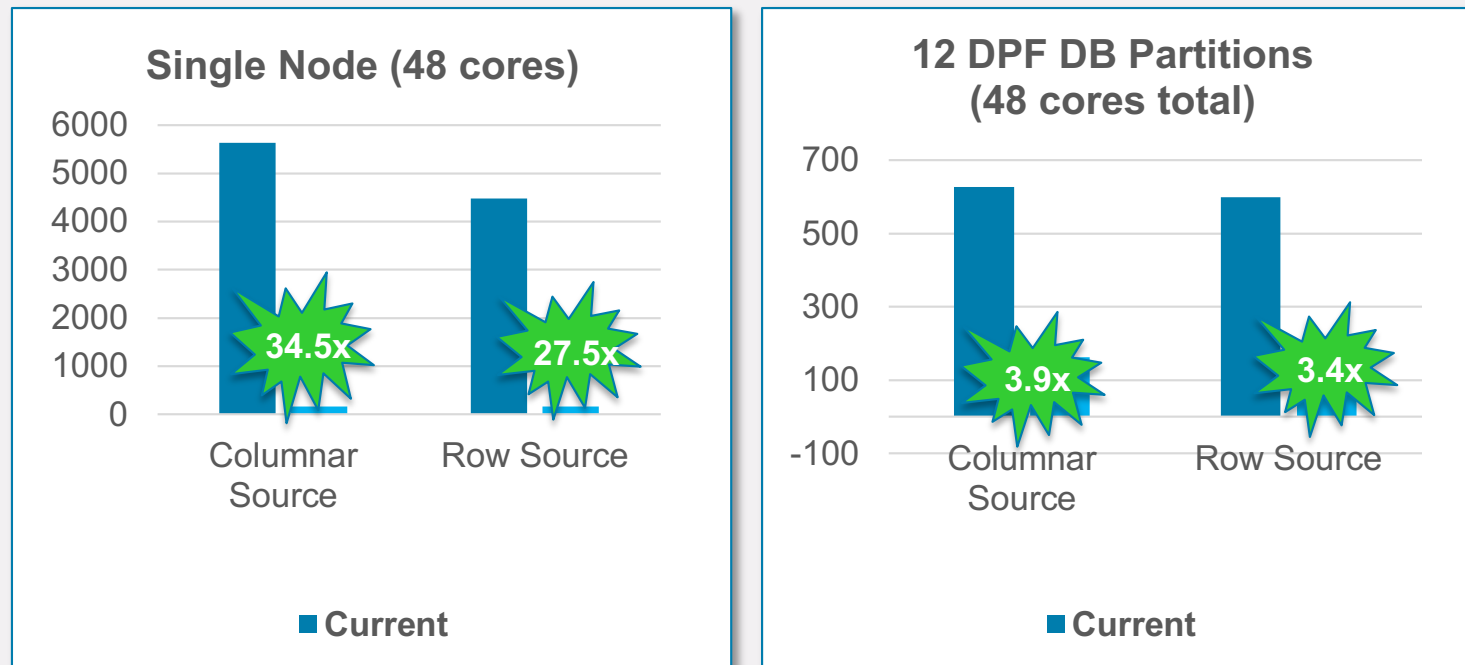
- ETL/ELT batch jobs
- Data population/ingest jobs
- Move/copy data from one table into another

```
INSERT INTO table2  
SELECT * FROM table1
```



Multi-Core INSERT Parallelism – Example Benefits

Elapsed time (seconds) for: `INSERT INTO table2 SELECT * FROM table1`



- 100 Million Rows
- 47GB
- DPF tests used 12 logical DB partitions, & co-located tables
- Same 48 core server used for single-node and DPF tests

User Maintained Temporal Tables for BLU

- Create column-organized user-maintained system-period temporal tables (STT)
- Create column-organized user-maintained application-period temporal tables (ATT)
- Create column-organized and user-maintained bi-temporal table (BTT)
- Execute read-only temporal queries on column-organized temporal tables.
- UPDATE and DELETE statements which use the FOR PORTION OF BUSINESS_TIME clause and row-splitting feature of temporal tables **are not supported**

```
UPDATE policy_info FOR PORTION OF BUSINESS_TIME FROM '2008-06-01' TO  
'2008-08-01' SET coverage = 14000 WHERE policy_id = 'A123';
```

The query above will result in row splitting and hence is not supported

Resolution: The user will have to delete the existing row and manually insert new rows with the updated values of coverage between '2008-06-01' and '2008-08-01', avoiding business time overlaps.

- **Technote to be published soon!**

Example IUD and Select for STT:

IUD:

```
INSERT INTO policy_info VALUES('C123',18000,'2009-03-01','2017-11-08');  
UPDATE policy_info SET coverage=20000 WHERE policy_id='B456';  
DELETE policy_info WHERE policy_id='B456';
```

Note: no history records are automatically generated for STT/BTT. But the USER (app) can maintain the history file themselves (insert to history as part of delete/update).

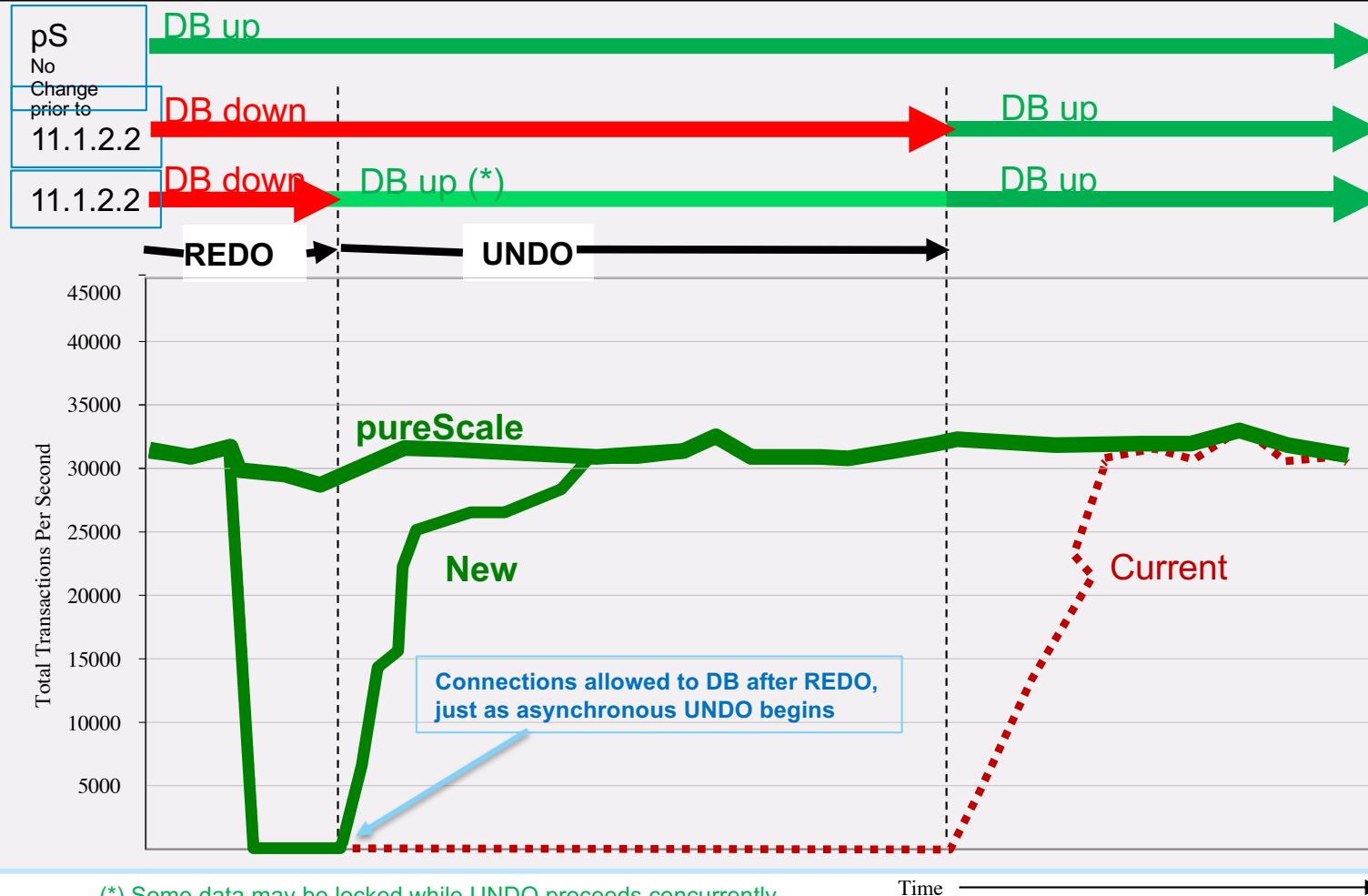
Users can issue the following typical temporal queries against column-organized user-maintained STT

```
SELECT * FROM policy_info FOR system_time AS OF '2009-12-01';  
SELECT * FROM policy_info FOR system_time BETWEEN '2009-05-01' AND '2010-02-01';  
SELECT * FROM policy_info FOR system_time FROM '2009-05-01' TO '2010-02-01';
```

Online Crash Recovery (Async UNDO)

- **Prior to 11.1.2.2, during Crash Recovery (or HADR Takeover by Force), connections are not allowed to the database until all recovery is complete**
 - A REDO phase to redo all transaction activity to the end of log
 - An UNDO phase to rollback transactions that had not committed
- **In Db2 11.1.2.2, we **allow connections and activity into the database** after the REDO phase and while the UNDO phase is executing**
 - This was a technical preview in Db2 11.1.1.1

Online Crash Recovery (Async UNDO)



(*) Some data may be locked while UNDO proceeds concurrently

Recoverability Enhancements

▪ Improved Transaction Rollback Performance

- Additional Db2 process now dedicated to maintaining recovery starting point
- Internal test scenario of 4 concurrent applications inserting & rolling back showed **5-6X faster rollback performance**

▪ Faster pureScale Member Crash Recovery

- Improved hashing algorithm and increased parallelism in global lock manager recovery design
 - Reduces the recovery window
- Benefits all pureScale deployments
 - Particularly beneficial for large clusters configured with huge lock list, global lock manager, and global buffer pool

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.

Performance Enhancements (1 of 2)

- **Adaptive Compression**

- Decompression efficiency improved with lab tests showing ~30% cpu savings

- **DECFLOAT**

- Hardware acceleration of DECFLOAT operators on supported platforms

- **COUNT/BIG_COUNT/SUM**

- Optimized operators in row and column based systems

- **TRUNC, MOD, ROUND**

- Constants in an SQL statement are evaluated at compile time rather than executed continuously at run time

```
SELECT * FROM EMPLOYEE WHERE SALARY > TRUNC(50000.8) ...
```

- The `TRUNC(50000.8)` would be evaluated before running the query

- **RENAME TABLESPACE**

- Catalog cache maintenance takes time and impacts concurrency
- Changes made internally to optimize memory usage in catalog cache
- Command runs much more efficiently

Performance Enhancements (2 of 2)

- **Query performance: BLU synopsis exploitation for LIKE predicates**
 - Data skipping capabilities leveraged by queries using LIKE predicates
- **Improved text search performance for queries containing OR predicates used with the text search functions**
 - Queries are converted (query rewrite) to use UNION DISTINCT
 - Example: books table contains 1000 rows

```
SELECT title FROM books WHERE CONTAINS(title, 'abc')=1
      OR CONTAINS(title, 'def')=1
      OR CONTAINS(title, 'ghi')=1
```
 - Previous design calls TS server *for each row for each value* – 3000 times
 - New design calls TS server *once for each value*– 3 times

SAP Performance Enhancements

- **Redundant Function Elimination**

- Eliminate functions in SQL statements that do not effect the outcome

SELECT * FROM EMPLOYEE WHERE RTRIM(RTRIM(NAME)) = 'HAAS'

- **Reduced Overhead in Unfenced C-UDFs Calls**

- **Complex CASE Statement Support (Increased Size)**

- **SAP BW Use of Parallel Insert (Up to 11x faster ETL processing)**

- **SAP DBA Cockpit exploitation of DB2_GET_INSTANCE_INFO for Windows**

- **Extend CHANGE HISTORY event monitor to include the following:**

- exclude / include logs, number of sessions, parallelism, size of the buffers
 - number of buffers, UTIL_IMPACT_PRIORITY, usage of deduplication
 - usage of compression with which compression library and which options
 - usage of encryption with which encryption library and which options
 - snapshot backup with which vendor library and which options
 - scripted snapshot with which script

Package Cache Efficiency

- **Current statement concentrator has two options:**
 - OFF: Statement Concentrator is off
 - LITERAL: Literals are removed from the SQL and compiled with parameter markers for more efficient package cache use
- **Using LITERAL does not work well when the SQL statement includes optimization information or comments**
 - SQL statements could be the same but the comments make them look different in the package cache
- **Two new options for the STMT_CONC database configuration parameter**
 - **COMMENTS**: concentrate statements based on comments. All comments in the SQL statements would be removed.
 - **COMM_LIT**: concentrate statements based on comments and literals in the SQL statement text.
- **Syntax:**
db2 update db cfg using STMT_CONC COMMENTS

Package Cache Efficiency

▪ Example 1:

SELECT EMPNO FROM EMPLOYEE --OPTLEVEL(5) --QUERY_DEGREE(1)

SELECT EMPNO FROM EMPLOYEE --OPTLEVEL(6) --QUERY_DEGREE(1)

- LITERAL: Each statement would be considered different in the package cache
- COMMENTS: Each statement is considered the same and we can reuse the package cache

▪ Example 2

SELECT EMPNO FROM EMPLOYEE WHERE EMPNO = 5 --OPTLEVEL(5)

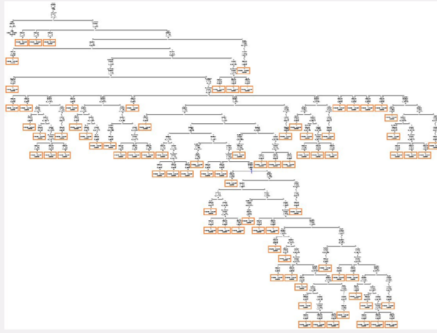
SELECT EMPNO FROM EMPLOYEE WHERE EMPNO = 6 --OPTLEVEL(6)

- COMMENTS: Each statement would be considered different in the package cache
- COMM_LIT: Each statement is considered the same because the literals are replaced with parameter markers and we can reuse the package cache

- **MON_GET_PKG_CACHE_STMT** now includes **STMT_COMMENTS** to view comment information in original SQL statement

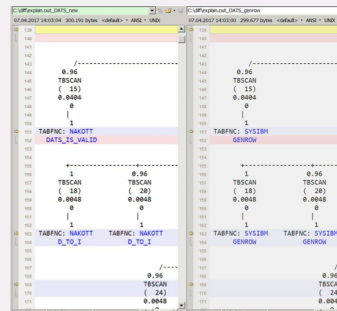
Updated Explain

- User-defined Table Functions in Explain Plans are listed as GENROW



Every Yellow box represents a table function – But which one?

- Explain Information Updated to include Function Name and Fragment Details
 - Easier to determine which function was involved in a branch of the explain plan



Security Enhancements

- **Support for local Certificate Authority within KeySecure**

- `ALLOW_NONCRITICAL_BASIC_CONSTRAINT` in `ekeystore.cfg`
 - Optional. If you set the parameter to `TRUE`, this allows Db2 to use local Certificate Authority within KMIP server that does not have a 'critical' keyword set and avoids '414' error returned by `GSKit[*]`. Default value: `FALSE`

- New check box in GUI labeled "Enable SSL security" is added in the security tab of the Microsoft ODBC Administrator window

- This element allows users to set the security configuration parameter value to SSL in Db2 Instance-based client

- **New option to create an encrypted sample database**

```
>>-db2sampl-----+-----+-----+-----+----->
      '-dbpath--path-name-'      '-name--database-name-'

>-----+-----+-----+-----+-----+-----+----->
      '-encrypt-----+-----+-----+-----+-----+-----+----->
      '-encrypts--options-'      '-force-'      '-quiet-----'
```

Other Enhancements

- **ADMIN_MOVE_TABLE updates**

- Transfer original ROW CHANGE TIMESTAMP content under ALLOW_READ_ACCESS when using COPY_USE_LOAD
- Optimize statistics during AMT

- **DB2_AVOID_LOCK_ESCALATION registry variable**

- Provides ability to prevent lock escalation when either MAXLOCKS or LOCKLIST thresholds are hit
- Application will return an SQL0912N error

- **COL_TABLE_SERIALIZE lock details externalized**

- Internal lock for serializing IUD activity on column organized tables
- Now externalized in all lock monitoring (monitor, snapshot, db2pd, etc)

- **Improved logging for db2chkupgrade to pinpoint invalid objects**

- Continuing to improve our messages for better self-help
- Identify names of the invalid object to avoid customers to search for them

- **New sub-options to db2trc for –cli : -clitracelevel and -pattern**

- https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0002027.html

XML Enhancements

▪ Implicit Names for XML Query

- Users can refer to a document without assigning a name. The XPATH expression won't need a document name at the beginning of the XPATH
- Compatible with syntax supported by Db2 for z/OS

Required syntax prior to 11.1.2.2

1. SELECT XMLQUERY('\$i/root/customerinfo[@Cid=1003]' passing INFO **as "i"**) FROM customer;
2. SELECT T.* from xs.ORDERS, XMLTABLE('\$i/order[orderedby=\$name]' passing xmlcol **as "i"**, 'John Doe' as "name" columns name VARCHAR(30) path 'orderedby', item INT path 'items/item/@id', quantity INT path 'items/item/quantity') AS T;

New and simpler syntax supported

1. SELECT XMLQUERY('/root/customerinfo[@Cid=1003]' passing INFO) FROM customer
2. SELECT T.* from xs.ORDERS, XMLTABLE('/order[orderedby=\$name]' passing xmlcol, 'John Doe' as "name" columns name VARCHAR(30) path 'orderedby', item INT path 'items/item/@id', quantity INT path 'items/item/quantity') AS T;

Federation Enhancements

- **Support for PostgreSQL and MySQL**

- No longer need to use the generic ODBC data source and wrapper
- Direct knowledge of these newly supported data sources
- Two simple DDL statements to query open source databases

```
create server POST type postgresql version 6.0 options (host '9.112.250.82', dbname  
'testdb', password 'Y', pushdown 'Y');
```

```
create user mapping for user server POST options(REMOTE_AUTHID 'iidev6',  
REMOTE_PASSWORD 'password');
```

- **Increased security with the ability to configure Secure Socket Layer (SSL) to connect to Db2 family data sources**

- https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.sec.doc/doc/t0070301.html

- **Integrated setup/configuration for ODBC drivers on PPC64LE**

- Previously available on all platforms except PPC64LE

Db2 Roadmap

JSON: Where we are Going

- **We will begin to implement an official (i.e. documented) set of built-in JSON SQL interfaces to conform with the recently released industry technical report from ISO (11.1.2.2)**
 - ISO/IEC TR 19075-6:2016, Information technology - Database languages - SQL - Part 6: SQL support for JavaScript Object Notation (JSON)
- **We will supplement this as needed with unique, built-in DB2 extensions to support the DB2 family storage model**
 - These will be compatible across the DB2 family
- **This will align us with both industry and DB2 family support for native SQL access of JSON**

Official SQL functions

Schema	Name	Comments
SYSIBM	BSON2JSON	Convert BSON formatted document into JSON strings
SYSIBM	BSON_VALIDATE	Checks to make sure that a BSON field in a BLOB object is in a correct format
SYSIBM	JSON2BSON	Convert JSON strings into a BSON document format
SYSIBM	JSON_EXISTS	Determine whether a JSON object contains the desired JSON value
SYSIBM	JSON_ARRAY	Creates JSON array from input key value pairs
SYSIBM	JSON_ARRAYAGG	Creates aggregate JSON array from input key value pairs
SYSIBM	JSON_OBJECT	Creates JSON object from input key value pairs
SYSIBM	JSON_OBJECTAGG	Creates aggregate JSON object from input key value pairs
SYSIBM	JSON_TABLE	Creates relational output from a JSON object
SYSIBM	JSON_QUERY	Extract a JSON object from a JSON object
SYSIBM	JSON_VALUE	Extract an SQL scalar value from a JSON object
(predicate)	IS JSON	Validates that input value is a valid JSON object

JSON Support for Existing Functions

- **DB2 will continue to store JSON as BSON in BLOB columns**
 - UTF8 database required
 - User will need to remember to ***explicitly*** convert to/from BSON when accessing JSON data from applications
 - Other JSON SQL functions will accept BSON format as input (where appropriate) to minimize conversion
- **Official JSON functions will exist in SYSIBM schema and become built-ins (11.1.2.2)**
 - Will no longer require privilege to execute
 - Cannot be dropped or modified by customer
- **Will maintain same level of BSON format to ensure compatibility for SYSTOOL version of functions**
- **Any new functions will be added to new and existing databases by default**

JSON Storage

- **You choose the format: JSON or BSON**
- **You choose the table organization: row or column**
- **You choose the column data type:**
 - By default, Db2 will assume character data types are JSON and binary ones are BSON
- **Examples:**
 - JSON
 - `CREATE TABLE T1 (C1 CHAR(300))`
 - `CREATE TABLE T1 (C1 VARCHAR(100))`
 - `CREATE TABLE T1 (C1 CLOB(1M))`
 - BSON
 - `CREATE TABLE T1 (C1 CHAR(300) FOR BIT DATA)`
 - `CREATE TABLE T1 (C1 VARCHAR(100) FOR BIT DATA)`
 - `CREATE TABLE T1 (C1 BLOB(1M))`

JSON Storage Examples

- **Normal SQL mechanisms are used to load JSON (or BSON) data into tables (e.g. INSERT, Load, etc.)**
 - **INSERT INTO T1 VALUES ('{{ "id": "0001", "type": "donut", "name": "Cake", "ppu": 0.55, "batters": { "batter": [{ "id": "1001", "type": "Regular" }, { "id": "1002", "type": "Chocolate" }, { "id": "1003", "type": "Blueberry" }, { "id": "1004", "type": "Devil's Food" }] }, "topping": [{ "id": "5001", "type": "None" }, { "id": "5002", "type": "Glazed" }, { "id": "5005", "type": "Sugar" }, { "id": "5007", "type": "Powdered Sugar" }, { "id": "5006", "type": "Chocolate with Sprinkles" }, { "id": "5003", "type": "Chocolate" }, { "id": "5004", "type": "Maple" }] }}');**
- **Complimentary (but optional) conversion functions are provided to move between JSON and BSON if so desired although you can also use other products to do this**
 - SYSIBM.BSON_TO_JSON
 - SYSIBM.JSON_TO_BSON

Parsing JSON with SQL

- Normal SQL mechanisms are used to retrieve stored JSON or BSON documents
 - `SELECT C1 FROM T1`
- New JSON scalar functions can be used to extract objects and values from JSON
 - `JSON_QUERY` returns a JSON object or array value
 - `JSON_VALUE` returns an SQL scalar value
- Examples:
 - Return the JSON object associated with the name key from a JSON text.
`VALUES (JSON_QUERY('{ "id": "701", "name": { "first": "John", "last": "Doe" } }', '$.name'));`
`Answer={ "first": "John", "last": "Doe" }`
 - Return a value from a JSON text as an integer.
`VALUES (JSON_VALUE('{ "id": "987" }', '$.id' RETURNING INTEGER));`
`Answer=987`

Integrating JSON with the Relational World

- **New JSON_TABLE function outputs the contents of JSON as a relational result set**

- Similar to what XML_TABLE does for XML

- **Example:**

- List the employee id, first name, last name, and first phone type and number:

```
SELECT U."id", U."first name",U."last name",U."phone type",U."phone number"
FROM EMPLOYEE_TABLE E,
     JSON_TABLE(E.jsondoc,'lax $',
        COLUMNS( "id" INTEGER,
                  "first name" VARCHAR(20) PATH 'lax $.name.first',
                  "last name" VARCHAR(20) PATH 'lax $.name.last',
                  "phone type" VARCHAR(20) PATH 'lax $.phones[0].type',
                  "phone number" VARCHAR(20) PATH 'lax $.phones[0].number')) AS U
```

id	first name	last name	phone type	phone number
901	John	Doe	home	555-3762
887	Mary	Brown	home	555-2732
891	Qi	Cheng	work	555-8377

Constructing JSON with SQL

- New JSON SQL functions can be used to construct JSON from relational data:
 - **JSON_ARRAY** creates a JSON array from input key value pairs
 - **JSON_ARRAYAGG** creates aggregate JSON array from input key value pairs
 - **JSON_OBJECT** creates JSON object from input key value pairs
 - **JSON_OBJECTAGG** creates aggregate JSON object from input key value pairs
- **Example: Generate a JSON object containing the last name, date hired, and salary for the employee with an employee number of '000020'**

```
SELECT JSON_OBJECT('Last name' : LASTNAME, 'Hire date' : HIREDATE,  
                  'Salary' : SALARY)  
FROM EMPLOYEE WHERE EMPNO = '000020'
```

```
{"Last name":"THOMPSON","Hire date":"1973-10-10","Salary":41250.00}
```


Miscellaneous JSON SQL functions

- **JSON_EXISTS** is a predicate that determines if a JSON object contains the specified JSON value
- **Format validation:**
 - **IS JSON** validates that the input value is a valid JSON object
 - **BSON_VALIDATE** checks that the input value is a valid BSON object
- **Examples:**
 - Return rows for employees who do not have an emergency contact in their JSON_DATA

```
SELECT empno, lastname FROM employee WHERE
      NOT JSON_EXISTS(COALESCE(json_data, ''),
        'strict $.emergency' FALSE ON ERROR);
```
 - Create a trigger to ensure only a valid JSON object can be inserted into a JSON_DATA column.

```
CREATE TRIGGER VALIDATE_JSON BEFORE INSERT ON T
REFERENCING NEW AS N
FOR EACH ROW
  IF N.JSON_DATA IS NOT JSON OBJECT THEN
    SIGNAL SQLSTATE '75007' SET MESSAGE_TEXT = 'Input is not valid JSON';
END IF
```

Indexing JSON documents

- Leveraging Db2's index on expression capability allows us to create indexes on JSON documents to allow faster access

- **Examples:**

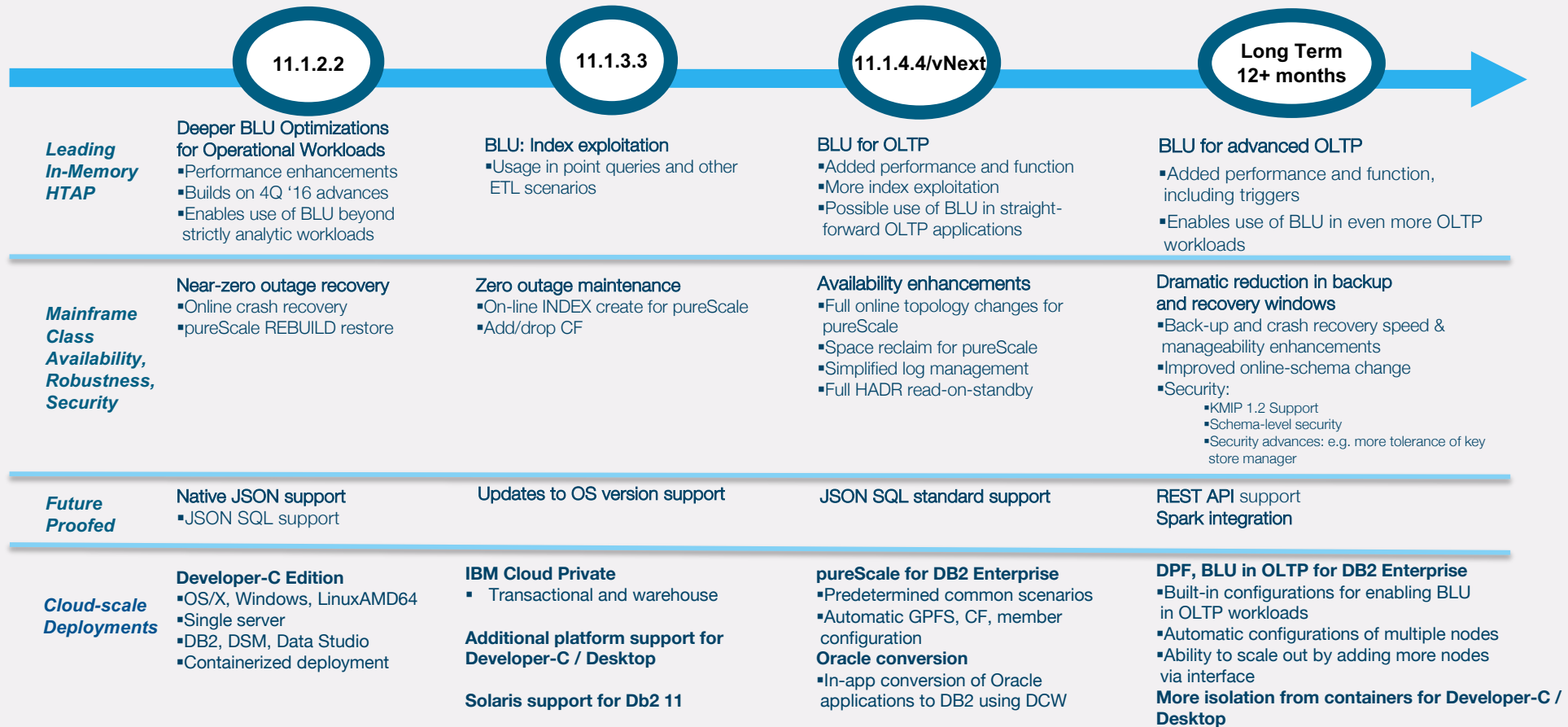
- Searching for an employee number will result in a scan against the table if no indexes are defined:

```
SELECT JSON_VALUE(EMP_DATA, 'lastname' RETURNING CHAR(20)) AS LASTNAME  
FROM JSON_EMP  
WHERE JSON_VALUE(EMP_DATA, 'empno' RETURNING CHAR(6)) = '000010';
```

- Creating the following index will greatly improve performance of this query

```
CREATE INDEX IX_JSON  
ON JSON_EMP (JSON_VALUE(EMP_DATA, 'empno' RETURNING CHAR(6)));
```

Db2 Roadmap



Summary and Close

DTE Assets

- **Sample Assets**
 - **How-to Videos**
 - **Self-guided Demos**
 - **Proof of Technology**
- **Available at:** <https://ibm.biz/dte-live>
- **Questions?** dte@us.ibm.com

Making Magic with Jupyter

- **Db2 Magic Commands**

- Secure and persistent database connections
- One command from SQL to DataFrame
- SQL Formatting in Cell
- Graphing

Db2 Jupyter Notebook Extensions Tutorial

The SQL code tutorials for Db2 rely on a Jupyter notebook extension, commonly refer to as a "magic" command. The beginning of all of the notebooks begin with the following command which will load the extension and allow the remainder of the notebook to use the %sql magic command.

```
%run db2.ipynb
```

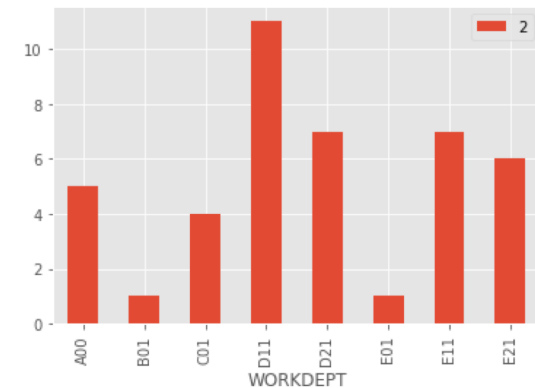
The cell below will load the Db2 extension. Note that it will take a few seconds for the extension to load, so you should generally wait until the "Db2 Extensions Loaded" message is displayed in your notebook.

```
In [1]: %run db2.ipynb
```

DB2 Extensions Loaded.

```
In [5]: %%sql -pb
SELECT WORKDEPT, COUNT(*)
FROM EMPLOYEE
GROUP BY WORKDEPT
```

<matplotlib.figure.Figure at 0x7f41aee06ef0>



github.com/DB2-Samples/db2jupyter

Why use Jupyter Notebooks with Db2?

- Combine SQL with Documentation - Markdown
- Combine SQL with the power of advanced analytics
- One line pivot
- Simple sort and group
- Powerful graphing
- Document management processes
- Built to share
- Up and running in minutes with DSX

JSON2BSON: Inserting a JSON Document

Inserting into a column requires the use of the JSON2BSON function. The JSON2BSON function (and BSON2JSON) are used to transfer data in and out of a traditional Db2 BLOB column. There is no native JSON data type in Db2. Input to the JSON2BSON function must be a properly formatted JSON document. In the event that the document does not follow proper JSON rules, you will get an error code from the function.

```
In [8]: %%sql
INSERT INTO TESTJSON VALUES ( JSON2BSON('{Name:"George"}') )

Command completed.
```

Db2 Jupyter Notebooks

JSON2BSON: Inserting a JSON Document

Inserting into a column requires the use of the JSON2BSON function. The JSON2BSON function (and BSON2JSON) are used to transfer data in and out of a traditional Db2 BLOB column. There is no native JSON data type in Db2. Input to the JSON2BSON function must be a properly formatted JSON document. In the event that the document does not follow proper JSON rules, you will get an error code from the function.

```
In [8]: %%sql
INSERT INTO TESTJSON VALUES ( JSON2BSON('{"Name":"George"}') )

Command completed.
```

```
{
  "empno": "000070",
  "firstnme": "EVA",
  "midinit": "D",
  "lastname": "PULASKI",
  "workdept": "D21",
  "phoneno": [7831, 1422, 4567],
  "hiredate": "09/30/2005",
  "job": "MANAGER",
  "edlevel": 16,
  "sex": "F",
  "birthdate": "05/26/2003",
  "pay": {
    "salary": 96170.00,
    "bonus": 700.00,
    "comm": 2893.00
  }
}
```

github.com/DB2-Samples/db2jupyter

Data Without Borders

- Customers want to access Data from non traditional platforms like Mobile platforms, from mobile apps (Android, Windows, iOS, etc.)
 - And in some cases want to publish access to data as well to 3rd parties
- Mobile and Cloud developers expect direct HTTP and JSON access to data without requiring Database Drivers
- OData is standards based way for our customers to leverage Db2 modern Cloud and Mobile Application Models

*OData (Open Data Protocol) is an **ISO/IEC approved, OASIS standard** that defines a set of best practices for building and consuming **RESTful APIs**.*

*OData helps you **focus on your business logic** while building **RESTful APIs** **without having to worry about the various approaches to define request and response headers, status codes, HTTP methods, URL conventions, media types, payload formats, query options, etc.***

OData also provides guidance for tracking changes, defining functions/actions for reusable procedures, and sending asynchronous/batch requests.

<http://www.odata.org/>

RESTful Support

Simplifying Access to Data Sources from Mobile Devices

- **The IBM OData Gateway provides a RESTful service that provides for the following operations:**
 - GET (SELECT)
 - POST (INSERT)
 - PUT (UPDATE a full row)
 - PATCH (UPDATE selected columns in a row)
 - DELETE (Delete a row)

```
url = http://localhost:9080/ODataOne/ODataService/SAMPLE-db2ab6eeb40d4a75a525333ee610fb50
```

```
header = {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json'  
}
```

```
OData = "/EMPLOYEES?$select=LASTNAME,SALARY&$filter=SALARY gt 50000"
```

```
RESTful.get(url + OData, headers=header)
```

V11.1 Summary of pureScale Enhancements

New Support

- SYNC and Near-SYNC HADR modes
- Power Linux LE
- Text Search
- Failover Priority for member subsets
- GDPC multi-switch
- SLES 12, RHEL 6.8 on x86
- RESTORE REBUILD
- Online CREATE INDEX (allow write)
- Online ADD/DROP CF
- Multiple hosts in maintenance mode
- AIX and Linux OS level updates

Improvements

- Install process and steps
- GPFS replication setup
- TRUNCATE TABLE
- Unified workload balancing
- Online management
- Member crash recovery (MCR)
- MCR improvements by default (no regvar)
- TCP/IP socket performance improvements
- Chrony auto-setup
- Single db2cluster commands to list hosts in maintenance mode and collecting perf data

■ Db2 v11.1 GA ■ Db2 v11.1.1.1 ■ Db2 v11.1.2.2 ■ Db2 v11.1.3.3

V11.1 Security, Availability, Usability and Serviceability

New Support

- Encryption centralized key managers / HSMs
- Partition level REORG for partitioned tables
- Roll-forward thru upgrade (avoid offline backup)
- Upgrade directly from 9.7, 10.1, or 10.5
- **Admin Move Table new FORCE_ALL option**
- **SSL encryption for HADR (Linux/x86)**
- **SSL encryption for Db2 family data sources**
- **Asynchronous UNDO**
- **HADR table space recovery**
- **DSM: Kerberos and 2 factor authentication**
- **db2haicu option to export TSAMP config to XML**
- **SSL encryption for HADR – all platforms**
- **Timeout for Vendor log archive API (UNIX only)**
- **PDF manuals**
- **DB2_GET_INSTANCE_INFO for Windows**
- **db2cli and tooling to capture/replay SAP BW EML**

Improvements

- Rollback and recovery (including HADR takeover)
- HADR upgrade – no reinitialization of standby
- Federation install, integration, and simplification
- Serviceability and problem determination (self help)
- **DSM: Simplification of federation setup, accelerated PD**
- **HADR resiliency for network data corruptions**
- **Improve identification of HADR replay only windows**
- **Reduced impact of buffer pool flush during BACKUP**
- **Large transaction rollback performance (optimize i/o)**
- **HADR replay window monitoring**
- **Db2pd enhancements for recovery and barstats**
- **Show LEFT OUTER TO ONE join in Explain**

■ Db2 v11.1 GA ■ Db2 v11.1.1.1 ■ Db2 v11.1.2.2 ■ Db2 v11.1.3.3

V11.1 Summary of BLU Acceleration Enhancements

New Support

- MPP (DPF)
- Nested Loop Join
- ROWID
- Parallel BLU Sort
- Wide rows
- Logical characters (CODEUNITS32)
- IDENTITY/EXPRESSION generated columns
- European Language (codepage 819)
- NOT LOGGED INITIALLY (NLI)
- Declared Global Temporary Tables (DGTT)
- Row and Column Access Control (RCAC)
- **INSERT from sub-select parallelism**
- **ALTER VARCHAR/VARGRAPHIC length**
- **Synopsis table LIKE predicates**
- **Indexes**
- **User maintained temporal tables**

Improvements

- SQL Merge
- Memory management
- Query rewrite
- OLAP function pushdown
- Sort utilization / elimination
- **Synopsis table - small row inserts**
- **SIMD exploitation**
- **GROUP BY and Aggregations**
- **Automatic dictionary (uncommitted data)**
- **Even better memory utilization**
- **CASE clause and UDF performance**
- **Compression performance**
- **AVX-512 exploitation for improved SIMD processing**

■ Db2 v11.1 GA ■ Db2 v11.1.1.1 ■ Db2 v11.1.2.2 ■ Db2 v11.1.3.3

V11.1 Additional General Enhancements

New Support

- SQL for JSON
- BOOLEAN, BINARY, and other data types (compatibility)
- Ubuntu 16.04 and Text Search on z Systems
- Windows Server 2016
- Federation to PostgreSQL and MySQL
- Drivers: .Net core and External Tables
- Solaris SPARC 11.3
- Federation to MariaDB, iSeries 7.2/7.3, Teradata 16
- Federation of JSON over RESTful service (MongoDB)
- DEC_ARITHMETIC DB parameter for Db2/z compatibility
- FFnRO now a select-statement clause for Db2/z compatibility
- OFFSET clause for Db2/z compatibility (OFFSET n ROW[S])

Improvements

- Bufferpool latch algorithm – scalable performance
- WLM: CPU shares and limits at DB level
- WLM: 2 new thresholds to include WLM queued time
- P7+ and P8 BACKUP and log archive compression
- INSERT sub-select with IDENTITY column
- Text Search with OR predicates
- SQL Compatibility: Oracle, Netezza, PostgreSQL, Db2/z
- Maximum log file size increased (16x) to 64GB
- DECIMAL, DECFLOAT, COUNT, SUM processing
- Adaptive compression “decompression”
- Package cache and Explain efficiency and management
- RENAME TABLESPACE (performance)
- Federation boolean and (var)binary data type mapping
- Federation pushdown for Hadoop and function mapping
- Apply early out to LEFT OUTER TO ONE joins
- TRIM and VARCHAR_BIT_FORMAT performance
- Query rewrite for complex joins