DB2 LUW V11.1 CERTIFICATION TRAINING PART #1

Kent Collins
Mohankumar Saraswatipura

Agenda

Why Certification?

DB2 V11.1 Certification Roadmap

C2090-600: IBM DB2 11.1 DBA for LUW Test Preparation

Sample Questions

Why Certification?

Expand your Knowledge

Create you Network

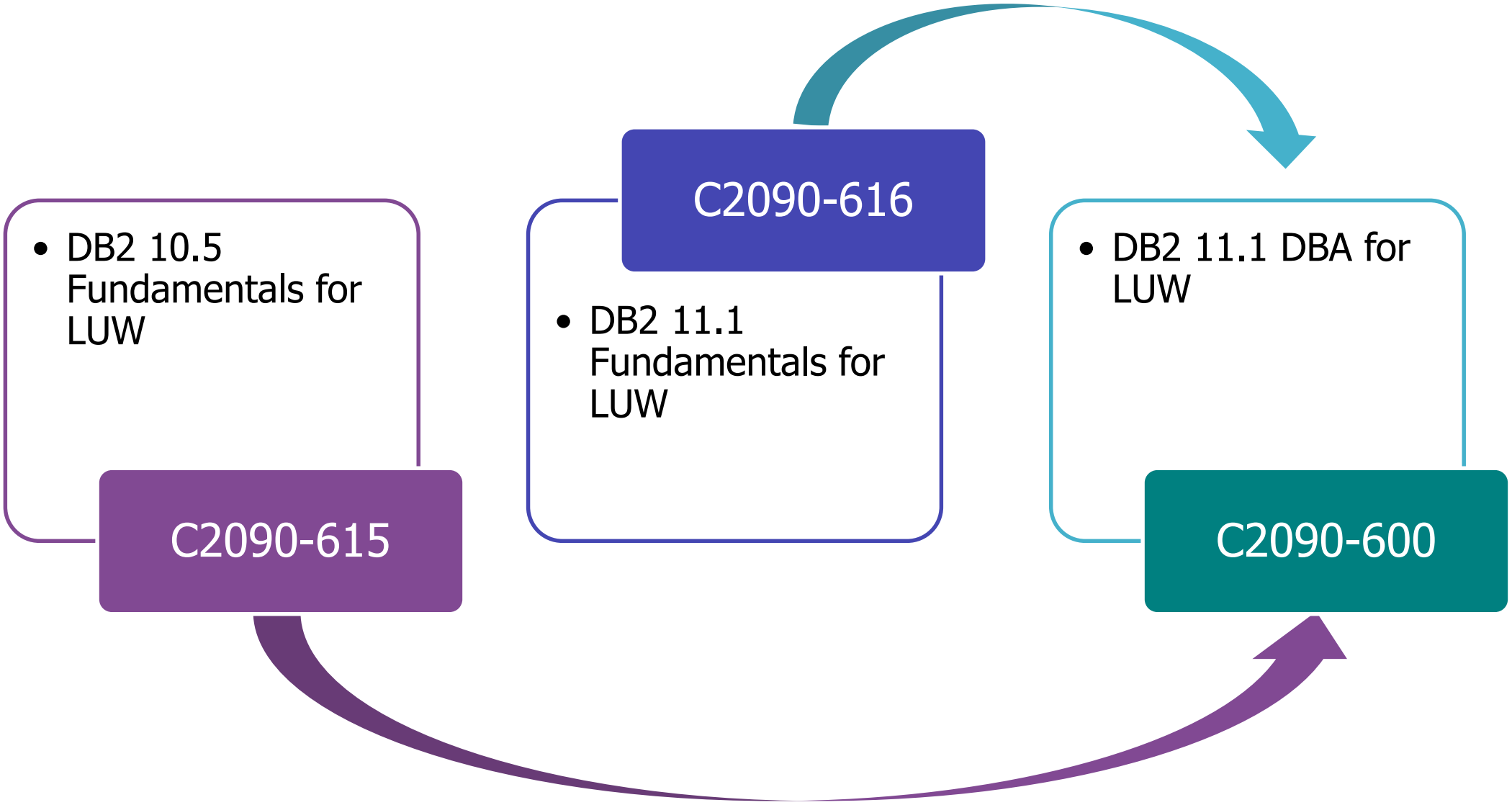
Incredibly Valuable in Consulting Services

Earn Better Salary

Fresher! don’t worry – certification will enable you learn faster

Experience Matters, however Certifications Help to Upgrade!

DB2 v11.1 Certification Roadmap



C2090-600: IBM DB2 11.1 DBA for LUW

Unit	Topic	Percentage Coverage
1	Server Management	15
2	Physical Design	22
3	Business Rules Implementation	10
4	Monitoring DB2 Activity	12
5	Utilities	13
6	High Availability	13
7	Security	15

DB2 BLU Acceleration

DB2 with BLU Acceleration integrates innovative new techniques for defining and processing column-organized table that speed reporting queries by a factor of 10x to 50x with a compression factor of up to 10x, compared to traditional row-organized tables, without having to create indexes or materialized tables/views.

You can configure a BLU environment by setting a few configuration parameters. If you set the instance aggregated registry variable **DB2_WORKLOAD=ANALYTICS**, all databases created with this setting in place will have the following parameters:

```
Degree of parallelism (DFT_DEGREE)                    = ANY
Default tablespace extentsize (pages)  (DFT_EXTENT_SZ)= 4
Default table organization             (DFT_TABLE_ORG)= COLUMN
Database page size                                    = 32768
```

DB2 BLU Acceleration …Cont.

The three critical database configuration settings for BLU workloads are

- SHEAPTHRES_SHR
- SORTHEAP
- DATABASE_MEMORY

Of the three DATABASE_MEMORY is set to AUTOMATIC. The DB2 database configuration manager will set the initial values for SHEAPTHRES_SHR and SORTHEAP based on server resources at the time of database creation.

On a 64 GB RAM server, set the database manager configuration, as follows:

```
SHEAPTHRES_SHR                  = 1266237
SORTHEAP                        = 63311
DATABASE_MEMORY                 = AUTOMATIC (9729422)
```

DB2 BLU Implementation Techniques

1. Create a DB2 instance
2. Set the registry variables
   DB2COMM=TCPIP
   DB2_WORKLOAD=ANALYTICS
   This aggregated registry variable implicitly sets
   DB2_WORKLOAD=ANALYTICS
   DB2_USE_ALTERNATE_PAGE_CLEANING=ON [DB2_WORKLOAD]
   DB2_ANTIJOIN=EXTEND [DB2_WORKLOAD]
3. Stop and start DB2 instance

DB2 BLU Implementation Techniques … Cont.

4. Set INSTANCE_MEMORY

Set INSTANCE_MEMORY to <PAGES> where <PAGES> is 90% of the server memory if the server dedicated to only one columnar database instance and has 128GB RAM or more.
If the server is multiple instances of DB2, select an appropriate percentage based on the workload. What is an appropriate number? Select a percentage that will use the available RAM without causing system paging and monitor the system paging on the server.

DB2 BLU Implementation Techniques … Cont.

4. Set Utility Heap to a Large Number

Set UTIL_HEAP_SZ to a large value with AUTOMATIC

5. Before you start building databases, consider these points. BLU and pureScale require databases with ASM enabled. Both also require Unicode code sets and IDENTITY or IDENTITY_16BIT collation. For column-organized tables, you must define table spaces with automatic space reclaim enabled.

```
CREATE DATABASE DB_COL ON '/db/ts01','/db/ts02','/db/ts03','/db/ts04'
DBPATH ON '/db/home' USING CODESET UTF-8 TERRITORY en_US COLLATE USING
IDENTITY
DB20000I The CREATE DATABASE command completed successfully.
```

DB2 BLU Implementation Techniques … Cont.

6. Set DATABASE_MEMORY to 80% of INSTANCE_MEMORY

Set DATABASE_MEMORY in both the row database and in the column database to “COMPUTE”. Review the computed amount of each to make sure the setting sum of both is less than INSTANCE_MEMORY * 80%.

DB2 BLU – Statistics Collection

For column-organized tables, both compression and statistics are managed automatically and internally. It is not possible to turn them on or off. This is why when the compression dictionary or statistics are being initially seeded, it is important to process the entire data set.

In some cases, the statistics associated with RUNSTATS are collected, and in other cases they are not. Using table functions, it is possible to determine whether statistics collections are executing or are waiting to run.

Real-time statistics collection (RTS) output is stored in the statistics cache and can be used by database agents for subsequent statements. The cached statistics are later written to the database catalog by a daemon process in servicing a WRITE_STATS request.

DB2 BLU – Statistics Collection… Cont.

A COLLECT_STATS request can be issued by a db2agent process during a statement compilation and the status can be queried using the below listed SQL statement

```
SELECT QUEUE_POSITION,
REQUEST_STATUS,
REQUEST_TYPE,
OBJECT_TYPE,
VARCHAR (OBJECT_SCHEMA, 10) AS SCHEMA,
VARCHAR (OBJECT_NAME, 10) AS NAME
FROM TABLE (MON_GET_RTS_RQST()) AS T
ORDER BY QUEUE_POSITION, SCHEMA, NAME;
```

There are three possible statuses for REQUEST_STATUS: EXECUTING, QUEUED, or PENDING. At most, you can have one table with EXECUTING status. RTS checks for PENDING requests every five minutes and places the requests on the run queue.

DB2 BLU – Space Management

DB2 has for a long time performed what is called logical deletion of rows. This is different from pseudo deletion, because in logical deletion, the space occupied by the deleted row on the data page can be overwritten with an inserted row, while pseudo deleted rows are not available until additional cleanup operations are performed.

For column-organized tables, data is pseudo deleted; for row organized tables, it is logically deleted. Thus, for column-organized tables, space reclaims are performed at the extent level, and the extent space is returned to the table space for use with any defined table.

Column-organized tables have no clustering index, so traditional REORG operations are unnecessary except when used for recovering pseudo deleted rows. You can execute the command manually, but it is automatically performed with AUTO_REORG. Extents recovered can be used by any column-organized tables in the table space.

To reclaim extents, execute a REORG command similar to the following:

```
REORG TABLE <TABLENAME> RECLAIM EXTENTS ;
```

Row and Column Organized Table Coexistence

Row and Colum-organized tables can coexist in the same database, in the same schema, table space and can be accessed by a single SQL statement.

To build a mix of row- and column-configured databases, begin with the instance configured for row. Build the row-based databases you need, set the **DB2_WORKLOAD=ANALYTICS** registry setting, and then restart the instance and create the database configured for column.

When running a DB2 instance with row- and column-organized databases, configure the database manager and database configuration settings for column workloads. Row workloads (OLTP) will run fine in a column-configured DB2 instance, but not vice versa.

DB2 BLU – Extended to MPP

- BLU DPF extends BLU Acceleration into a true MPP column store
- Data exchange during distributed joins and aggregation processing occurs entirely within the BLU runtime in native columnar format
- BLU MPP exploits a common compression encoding across data slices
- Automatic Global Dictionary Creation
- Compressed Communications

DB2 BLU – Extended to MPP …Cont.

- Load and Go Simplicity
- It auto detect and adapt to available memory, cores and cache
- Optimizes FCM configuration automatically
  - FCM_BUFFER_SIZE
  - FCM_PARALLELISM
- Just like a regular row-organized MPP, Data is distributed across database partitions according to a distribution key (that is used to determine the database partition in which a particular row of data is stored)
- Each table has its own distribution key defined
- A distribution key can be a single column or group of columns
- The performance of queries that join tables will typically be increased if the join is collocated

DB2 BLU – Extended to MPP …Cont.

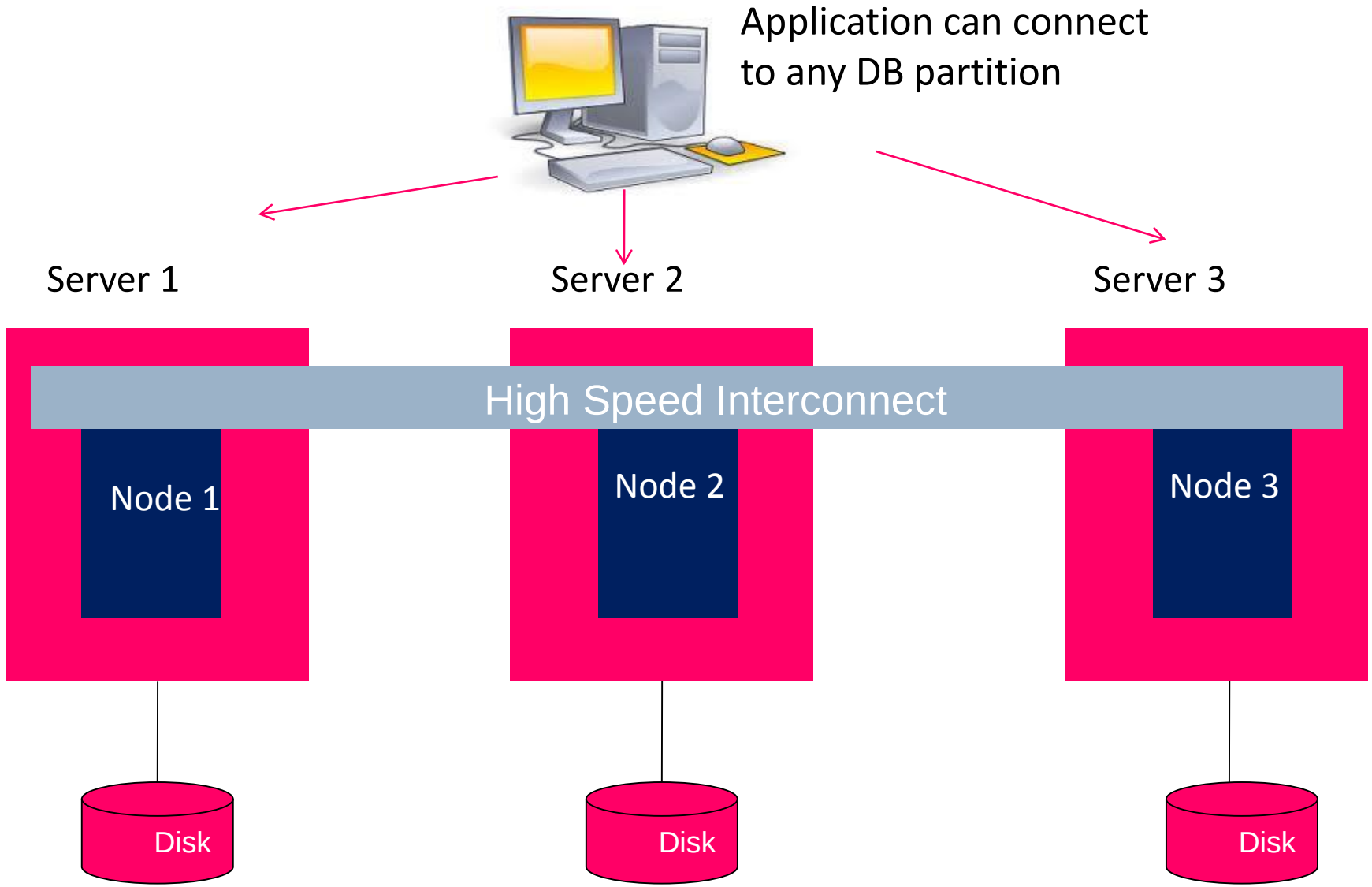
RANDOM DISTRIBUTE BY Clause

```
CREATE TABLE sample(c1 INTEGER NOT NULL, c2 VARCHAR NOT NULL, c3 CHAR(10))
IN TBSP1
ORGANIZE BY COLUMN
DISTRIBUTE BY RANDOM [~Generally HASH]
```

Choose RANDOM when

- Collocated joins are not possible
- Collocated joins are not necessary due to smaller table size
- Distribution by hash result in significant data skew across the data partitions

DB2 BLU – Extended to MPP… Cont.



DB2 BLU – Points to Remember

- Cannot create column-organized table database in pureScale environment
- Code set should always be UNICODE or code page 819 and collation should be IDENTITY or IDENTITY_16BIT
- Do not support RS and RR isolation levels
- Cannot be used with automatic tuning of SORT memory
- Table spaces must always be automatic storage enabled that supports reclaimable storage
- Schemas cannot be transported
- Indexes cannot be explicitly created
- Triggers cannot be created
- Column-organized tables cannot the source for CDC (change data capture) or for data replication
- LBAC cannot be used on column-organized tables
- Column-organized table cannot be a Range Partition, MDC, Temporal and Typed table
- Columns with the BLOB, CLOB, DBCLOB, NCLOB or XML data types cannot be included in a column-organized table

DB2 pureScale HADR Implementation

One of the significant benefits of PureScale design is much of what a DBA knows running ESE transfers to running PureScale and HADR is no exception.

Basic Steps

- Create Standby DB
- Configure HADR on Primary and Standby
- Start HADR

HADR Restrictions

What are some HADR restrictions?

- Same base OS and same DB2 base software except for a short time during upgrade.
- No DPF
- Same Bit level (32 or 64)
- Direct Access log files are not supported (RAW)
- HADR does not support infinite logging. (Not even sure I know what this means)

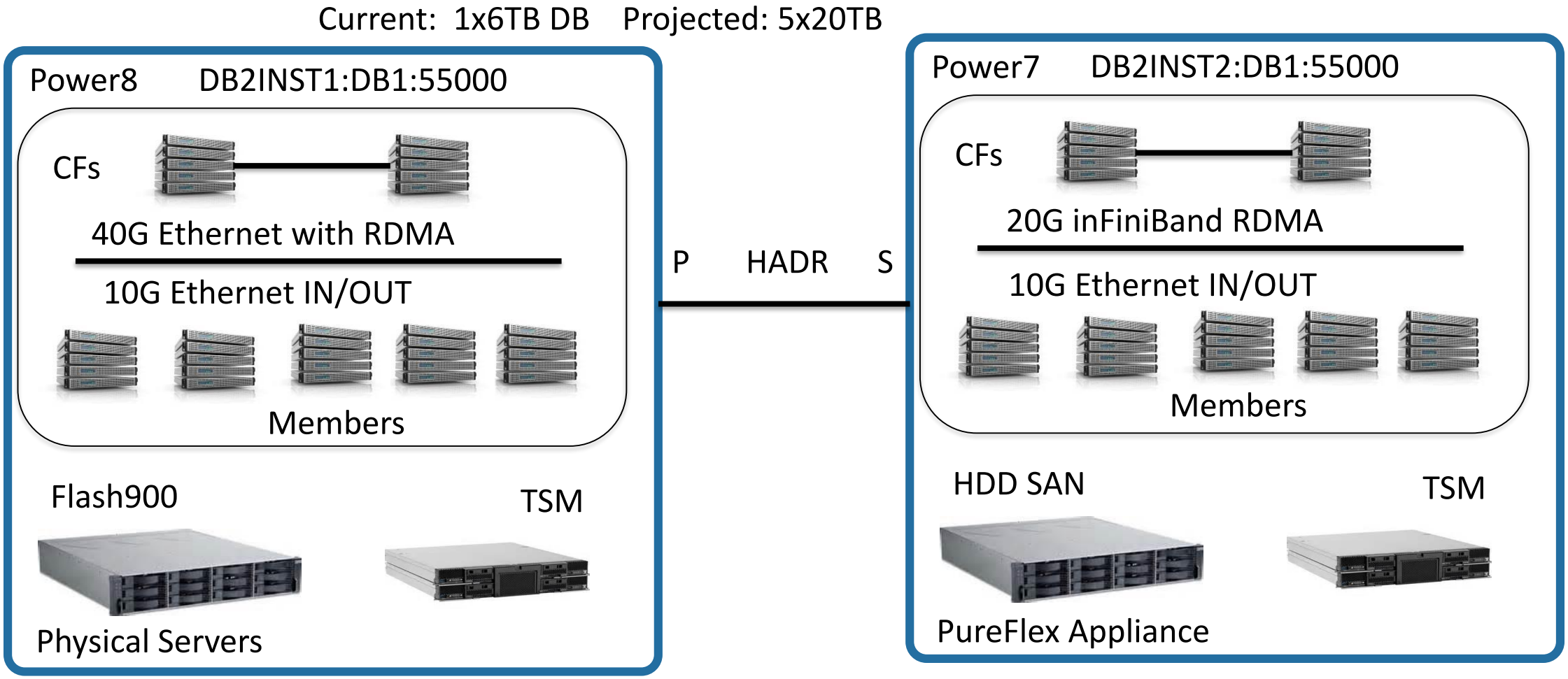
What about PS?

- Hadr_peer_window is not supported (Set to 0)
  HADR peer window duration (seconds) (HADR_PEER_WINDOW) = 0
- Multiple Standby databases is not supported
- Primary and Standby topology must be synchronized
- “ROS” is not supported
- DB2/TSA is not supported
- NAT between Primary and Standby is not supported

DB2 pureScale HADR Implementation…Cont.

Setting up HADR in a PureScale configuration:

We assume you have two PureScale clusters running and you want to setup HADR on one of the databases.



DB2 pureScale HADR Implementation…Cont.

Servers	PRIMARY	
DB2P031	db2inst1	Member 0
DB2P032		
DB2P033		
DB2P034		
DB2P035		
CCFP011		
CCFP012		

Servers	STANDBY		
DB2P051	db2inst2	REPLAY MEMBER	Member 0
DB2P052			
DB2P053			
DB2P054			
DB2P055			
CCFP051			
CCFP052			

DB2 pureScale HADR Implementation…Cont.

- Take an offline or online with logs full backup of Primary and restore it on Standby
- Set DB2 HADR configuration parameters on primary

```
HADR local host name                         (HADR_LOCAL_HOST)              = db2p035
HADR local service name                      (HADR_LOCAL_SVC)    = 56002
HADR remote host name                        (HADR_REMOTE_HOST)             =
{db2p051:56002|db2p052:56002|db2p053:56002|db2p054:56002|db2p055:56002}
HADR remote service name                     (HADR_REMOTE_SVC)              =
HADR instance name of remote server          (HADR_REMOTE_INST)             = db2inst2
HADR timeout value                           (HADR_TIMEOUT)      = 120
HADR target list                             (HADR_TARGET_LIST) =
{db2p051:56002|db2p052:56002|db2p053:56002|db2p054:56002|db2p055:56002}
HADR log write synchronization mode          (HADR_SYNCMODE)    = ASYNC
HADR spool log data limit (4KB)              (HADR_SPOOL_LIMIT)             = AUTOMATIC(28180480)
HADR log replay delay (seconds)              (HADR_REPLAY_DELAY)            = 0
HADR peer window duration (seconds)          (HADR_PEER_WINDOW) = 0
```

DB2 pureScale HADR Implementation…Cont.

- Make /etc/services entries on primary server

```
db2c_db2inst2           55000/tcp
db2c_hadr_db1           56002/tcp   # Reserve communication and interrupt ports 56002/56003
db2inst2_monhadr        29000/tcp   # db2inst2-HADR F5 listener
DB2_db2inst2            60002/tcp
DB2_db2inst2_1          60003/tcp
DB2_db2inst2_2          60004/tcp
DB2_db2inst2_3          60005/tcp
DB2_db2inst2_4          60006/tcp
DB2_db2inst2_END        60007/tcp
DB2CF_db2inst2          56003/tcp
DB2CF_db2inst2_MGMT     56004/tcp
```

DB2 pureScale HADR Implementation…Cont.

- Set DB2 HADR configuration parameters on standby

```
HADR local host name                    (HADR_LOCAL_HOST)     = db2p055
HADR local service name                 (HADR_LOCAL_SVC)      = 56002
HADR remote host name                   (HADR_REMOTE_HOST)    =
{db2p031:56002|db2p032:56002|db2p033:56002|db2p034:56002|db2p035:56002}
HADR remote service name                (HADR_REMOTE_SVC)     =
HADR instance name of remote server     (HADR_REMOTE_INST)    = db2inst1
HADR timeout value                      (HADR_TIMEOUT)        = 120
HADR target list                        (HADR_TARGET_LIST)    =
{db2p031:56002|db2p032:56002|db2p033:56002|db2p034:56002|db2p035:56002}
HADR log write synchronization mode     (HADR_SYNCMODE)       = ASYNC
HADR spool log data limit (4KB)         (HADR_SPOOL_LIMIT)    = AUTOMATIC(28180480)
HADR log replay delay (seconds)         (HADR_REPLAY_DELAY)   = 0
HADR peer window duration (seconds)     (HADR_PEER_WINDOW)    = 0
```

DB2 pureScale HADR Implementation…Cont.

- Make /etc/services entries on standby server

```
db2c_db2inst1           55000/tcp
db2c_hadr_db1           56002/tcp   # Reserve communication and interrupt ports 56002/56003
db2inst1_monhadr        29000/tcp   # db2inst1-HADR F5 listener
DB2_db2inst1            60002/tcp
DB2_db2inst1_1          60003/tcp
DB2_db2inst1_2          60004/tcp
DB2_db2inst1_3          60005/tcp
DB2_db2inst1_4          60006/tcp
DB2_db2inst1_END        60007/tcp
DB2CF_db2inst1          56003/tcp
DB2CF_db2inst1_MGMT     56004/tcp
```

DB2 pureScale HADR Implementation…Cont.

- In V10.5 HADR_SYNCMODE can be ASYNC or SUPERASYNC and DB2 V11.1 supports SYNC and NEARSYNC

- Auto Configuration Option in pureScale will set HADR_REMOTE_HOST, HADR_REMOTE_SVC and HADR_REMOTE_INST values when HADR START is executed and the current settings are "NULL". Stopping and starting the Standby will not propagate Primary changes to the Standby. You must completely shutdown HADR in order for changes to be propagated. Also note that HADR_REMOTE_HOST is set from REMOTE_TARGET_LIST but HADR_REMOTE_SVC is always NULL.

  Recommendation: Leave HADR_REMOTE_HOST, HADR_REMOTE_SVC, HADR_REMOTE_INST as "NULL"

DB2 pureScale HADR Implementation…Cont.

- Activate HADR database DB1 on both clusters
- Start HADR on Replay Member both Primary and Standby
  - On DB2P051 as db2inst2 run db2 "START HADR ON DB db1 AS STANDBY"
  - On DB2P031 as db2inst1 run db2 "START HADR ON DB db1 AS PRIMARY"
- You can monitor HADR using the db2pd command
  - On Primary Cluster from any Member run "db2pd -d db1 -hadr –allmembers”
- On the Standby Cluster the db2pd command must be run from the REPLAY member

DB2 pureScale HADR Implementation…Cont.

- Takeover (Failing the DB and Active Connections to the Standby and Back)

The DB2 takeover command can be run from any Member of the Standby Cluster. It can be run in FORCED(Failover) and NON-FORCED(Role Switch or Normal) mode requires a active primary DB, but make sure the database is in PEER state before running the command in NON-FORCE mode. The "FAILOVER" or FORCED method is normally run with the PRIMARY DB is down.

db2 " TAKEOVER HADR ON DB db1 "

Please note we can't automate automatic HADR failover using TSA at this moment.

ADMIN_MOVE_TABLE ()

This procedure moves data stored in an existing table to a new table that has the same name but may have been defined in a different table space.

You can also use this procedure to change the

- MDC (ORGANIZE BY DIMENSION) specification for a table,
- alter the ITC (ORGANIZE BY INSERT TIME) specification for a table,
- modify a table's partitioning keys (DISTRIBUTE BY HASH),
- change the range partitioning (PARTITION BY RANGE) specification for a table,
- add or remove columns from a table, alter a column's data type (provided the new data type is compatible with the data type being used),
- create a new compression dictionary for a table that has deep compression enabled

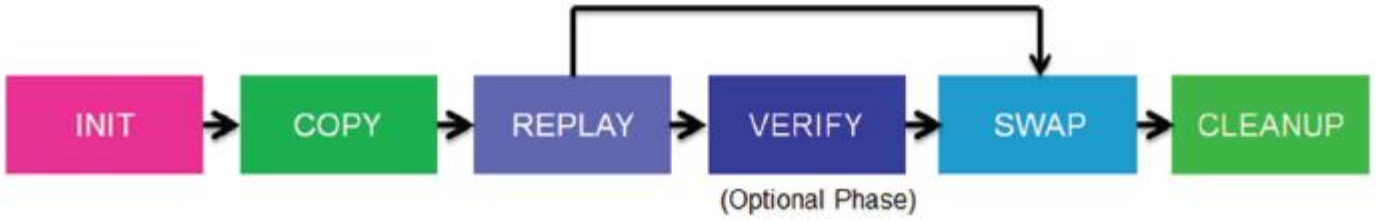
ADMIN_MOVE_TABLE () … Cont.

When you invoke the ADMIN_MOVE_TABLE () procedure, a shadow copy of the table to move is created in the table space specified (you can also create the target table manually beforehand).

This procedure also creates a staging table and set of triggers to capture data changes made on the source table during the move operation. Data is then copied from the source to the target by using either INSERT FROM CURSOR or LOAD FROM CURSOR.

Once the data has been copied, changes captured in the staging table are replayed against the target table to bring it up to date. During this phase, the source table is briefly taken offline to rename it. By default, the source table is then dropped; however, it can be kept and renamed by using the KEEP option.

ADMIN_MOVE_TABLE () Phases



INIT. This phase initializes all the objects required for the operation, including the staging table that is necessary for capturing all the data changes during the move.

COPY. This phase creates a copy of the source table according to the current definition and copies the data into the target table.

REPLAY. This phase replays all the changes captured in the staging table into the target table just before swapping the source and target tables.

VERIFY. This is an optional phase that checks the table contents between source and target to make sure they are identical before the swap.

SWAP. This phase performs a swapping of source and target tables. The source table will be taken offline briefly to complete the REPLAY.

ADMIN_MOVE_TABLE () Phases… Cont.

CLEANUP. This phase drops all the intermediate tables created during the online move such as the staging table, any non-unique indexes, and triggers.

In DB2 v11.1, we have 2 new options available

- TERM: Terminates a running or killed table move
- REPORT: Calculates a set of values to monitor the progress of a single or multiple table moves. Focus is the COPY and REPLAY phase of a running table move

Sample Questions: #1

The DBA of company ABC is managing a HADR multiple standby environment having one primary, one principal standby, and one auxiliary standby. The DBA uses the MON_GET_ HADR table function to monitor the HADR status. What will be the HADR_STATE for the auxiliary standby database?

A. Local Catchup state
B. Remote Catchup state
C. Remote Catchup Pending state
D. PEER state

Answer:

The correct answer is B. The HADR auxiliary standby database will always be in the REMOTE_CATCHUP state irrespective of the HADR_LOG_GAP between the primary and the auxiliary standby. In the following example, standby member 10.112.0.1 is the auxiliary standby server.

```
db2 "SELECT HADR_ROLE, STANDBY_ID, HADR_STATE, PRIMARY_MEMBER_HOST,
STANDBY_MEMBER_HOST FROM TABLE (MON_GET_HADR (NULL))“

HADR_ROLE STANDBY_ID HADR_STATE PRIMARY_MEMBER_HOST STANDBY_MEMBER_HOST
--------- ---------- ---------- -------------------- ----------------
-
PRIMARY 1 PEER 10.110.20.5 10.110.7.6
PRIMARY 2 REMOTE_CATCHUP 10.110.20.5 10.112.0.1
2 record(s) selected.
```

Sample Questions: #2

Which isolation level is supported on the HADR active standby read-only database?

A. Read Stability
B. Repeatable Read
C. Uncommitted Read
D. Cursor Stability
E. Currently Committed (CUR_COMMIT)

Answer:

The correct answer is C. The only isolation level that is supported on the Read on Standby HADR database is Uncommitted Read (UR). Any application requests other than UR will receive an error SQL1773N reason code 1. DBA can enforce the UR isolation level on the standby by setting the DB2 registry variable DB2_STANDBY_ISO=UR.

When you set the CUR_COMMIT database configuration parameter to ON, all the queries will return the data's committed value when the query is submitted.

Sample Questions: #3

Company ABC has an HADR production environment and wants to isolate most of the expensive read-only SQL operations on the standby by using the read on standby (ROS) feature. How do you enable the ROS feature in this case?

A. Set the DB2 registry variable DB2_HADR_ROS to ON
B. Set the DB2 registry variable DB2_HADR_SOSNDBUF to ON
C. Set the DB2 registry variable DB2_HADR_PEER_WAIT_LIMIT to ON
D. Set the DB2 registry variable DB2_HADR_NO_IP_CHECK to ON

Answer:

The correct answer is A.

You can enable the ROS on the HADR standby database by using the DB2 instance level registry variable DB2_HADR_ROS. The steps involved are:

Step 1: Set the registry variable.
db2set DB2_HADR_ROS=ON
Step 2: Deactivate the standby database.
DEACTIVATE DB HADRDB
Step 3: Stop HADR on the standby database.
STOP HADR ON DATABASE HADRDB
Step 4: Stop and start the standby DB2 instance.
db2stop; db2start
Step 5: Start HADR on the standby database.
START HADR ON DATABASE HADRDB AS STANDBY

Sample Questions: #4

Which statement is true with respect to aggregated registry variable functionality?

A. A registry variable that is explicitly set by an application can only be overwritten by an aggregated registry setting
B. An aggregated registry variable that is explicitly set by an application cannot be overwritten
C. A registry variable that is implicitly configured through an aggregated registry variable can also be explicitly configured
D. A registry variable that is implicitly configured through an aggregated registry variable takes precedence over an explicitly configured value

Answer:

The correct answer is C. An aggregate registry variable is a group of several registry variables as a configuration that is identified by one registry variable name. Each registry variable that is part of the group has a predefined setting. The purpose of an aggregate registry variable is to ease registry configuration for broad operational objectives.

You can use an aggregate registry variable to explicitly define any registry variable that is implicitly configured, which in a way overrides the aggregated registry variable implicit value.

Option A is incorrect—an aggregated registry variable cannot override the registry variable.

Option B is also incorrect—an aggregated registry variable can easily be overwritten by explicitly setting the value for a registry.

Option D is incorrect as well—the explicit registry setting takes precedence over the implicit aggregated registry setting.

Sample Questions: #5

Which command shows the location of the global registry file?

A. db2licm
B. db2ls
C. db2greg
D. db2 show install locations

Answer:

The correct answer is C. The command to display the location of the global registry file is db2greg –g. Option A is incorrect due to the fact that db2licm command is used to work on licenses and option B db2ls to list the installed DB2 copies. Option D is invalid – no such command in DB2.

Sample Questions: #6

Which of the following statements about BLU MPP is TRUE?

A. DBA will have to manually configure FCM_BUFFER_SIZE and FCM_PARALLELISM
B. Row and Column-organized tables data cannot be collocated
C. Each table has multiple compression dictionary based on the number of partitions
D. When data is shipped across partitions, there is no need to decode and encode the data

Answer:

The correct answer is D. When we load data into BLU MPP, each member creates a local histogram which is then sent to coordinator/build node to create a common global dictionary. The build node will distribute the common dictionary to all the nodes to exploit a common compression encoding across the data partitions.

Option A is incorrect, FCM parameters are automatically configured by DB2. Option B is incorrect due to the fact that both row and column-organized tables data can form a collocation join. Option C. contradicts option D.

Sample Questions: #7

Tables can be converted from row to column organization by using db2convert utility command. Identify the characteristics that will not stop conversion.

A. Trigger
B. Foreign Key
C. MQT
D. XML/LOB

Answer:

The correct answer is B. Secondary indexes are dropped and not defined on a column table.
Option A is incorrect, if a trigger is defined, drop it and then convert the table. Option C and D are not supported in column-organized tables.

Sample Questions: #8

Which phase of ADMIN_MOVE_TABLE () is essential to minimize the table offline window?

A. REPLAY
B. COPY
C. VERIFY
D. TERM

Answer:

The correct answer is A. REPLAY phase replays all the changes captured in the staging table into the target table just before swapping the source and target tables. This is very essential to run this phase multiple times to copy the staging data at frequent intervals to minimize the change data volume.

Sample Questions: #9

Which statement is false regarding HADR auxiliary standby functionality in a multiple standby environment?

A. Supports the ROS feature
B. Supports a maximum of two auxiliary standbys
C. Supports only a manual HADR failover
D. Supports all four HADR synchronization modes

Answer:

The correct answer is D. The only supported synchronization mode for auxiliary standby is SUPERASYNC. You can see the supported modes below:

Principal Standby	Auxiliary Standby
Supports ROS feature	Supports ROS feature
Only one standby database can act as the principal standby	Maximum of two standby databases can act as auxiliary standbys
Synchronization is through a TCP/IP direct connection	Synchronization is through a TCP/IP direct connection
Time delayed log shipping is supported	Time delayed log shipping is supported
TSA MP automated failover is supported	Only a manual failover is supported
All four synchronization modes are supported	Only SUPERASYNC synchronization mode is supported

Sample Questions: #10

Which data movement utility is suitable for moving and processing large amounts of real-time data into the data warehouse without affecting availability?

A. LOAD WITH NO ACCESS
B. SQL or Q replication
C. INGEST
D. IMPORT WITH COMMITCOUNT AUTOMATIC

Answer:

The correct answer is C. The Ingest utility is a high-speed, client-side, highly configurable, multithreaded DB2 utility that streams data from files and pipes into DB2 target tables by using SQL-like commands. Because the Ingest utility can move large amounts of real-time data without locking the target table, you do not need to choose between the data currency and availability.

LOAD WITH NO ACCESS does not allow users to access the table until it finishes, and SQL and Q replication can replicate one or more tables between the source and target systems to capture the data changes; however, these are not designed to move large amount of data at one go because they do call INSERT/UPDATE/DELETE/MEGRE internally in sequence.

IMPORT WITH COMMITCOUNT AUTOMATIC, even though this option allows other users to access the target table, it takes very long to process the data load due to sequential, multiple INSERTs threads.

Sample Questions: #11

When a database is created with DB2_WORKLOAD=ANALYTICS setting, what is the default DB CFG value for the below listed parameters?

Database Page Size
DFT_DEGREE
DFT_EXTENT_SZ
DFT_TABLE_ORG

A. 32768, ANY, 4 and COLUMN
B. 4096, ANY, 4 and COLUMN
C. 32768, 1, 4 and COLUMN
D. 32768, ANY, 32 and COLUMN

Answer:

The correct answer is A.

You can configure a BLU environment by setting a few configuration parameters. If you set the instance aggregated registry variable **DB2_WORKLOAD=ANALYTICS**, all databases created with this setting in place will have the following parameters:

```
Degree of parallelism (DFT_DEGREE)                    = ANY
Default tablespace extentsize (pages)  (DFT_EXTENT_SZ)= 4
Default table organization            (DFT_TABLE_ORG)= COLUMN
Database page size                                   = 32768
```

Thank you!

Robert (Kent) Collins
@rcollins963

Kent, founder of Shiloh Consulting, Inc., is currently a Data Solutions Architect with BNSF Railway. He is an IBM Champion (2010–2017) and a frequent speaker at the DB2Night Show and IDUG and IOD (IBM Insight) conferences. Kent has worked continually with DB2 from its introduction to the market in 1984, amassing a wealth of knowledge and experience. He graduated from the University of Texas at Dallas with majors in mathematics and computer science. He is an IBM Certified Solutions Expert and also holds certifications in DB2, AIX, Linux, .NET, Oracle, SQL Server, Windows, and z/OS. Kent is proficient in many programming languages and, as a Java Architect, specializes in Enterprise HA/Performance systems. He lives in Dallas with his wife, Vicki; together, they have three children and one grandchild.

Mohankumar Saraswatipura
@mohankumarsp

Mohan works as a Database Solutions Architect focusing on IBM DB2, Linux, UNIX, and Windows solutions. Prior to his current position, he worked as a Database Solutions Architect at Reckitt Benckiser Group, plc (UK) focusing on IBM Smart Analytics System 5600, Siebel, SQL Server, and SAP HANA solutions. He is an IBM Champion (2010–2017) and a *DB2's Got Talent* 2013 winner. Mohan has written dozens of technical papers for IBM developerWorks and *IBM Data* Magazine. He is an IBM Certified DB2 Advanced Database Administrator, DB2 Application Developer, and DB2 Problem Determination Master. Mohan holds a Master's of Technology (M Tech) degree in computer science and an Executive MBA (IT).

Please contact us if you have any questions @ www.EnterpriseDB2.com