Maintaining Resilience: Optimizing Your Db2 Subsystems

Samantha Buhler, Product Manager
Steen Rasmussen, Technical Consultant

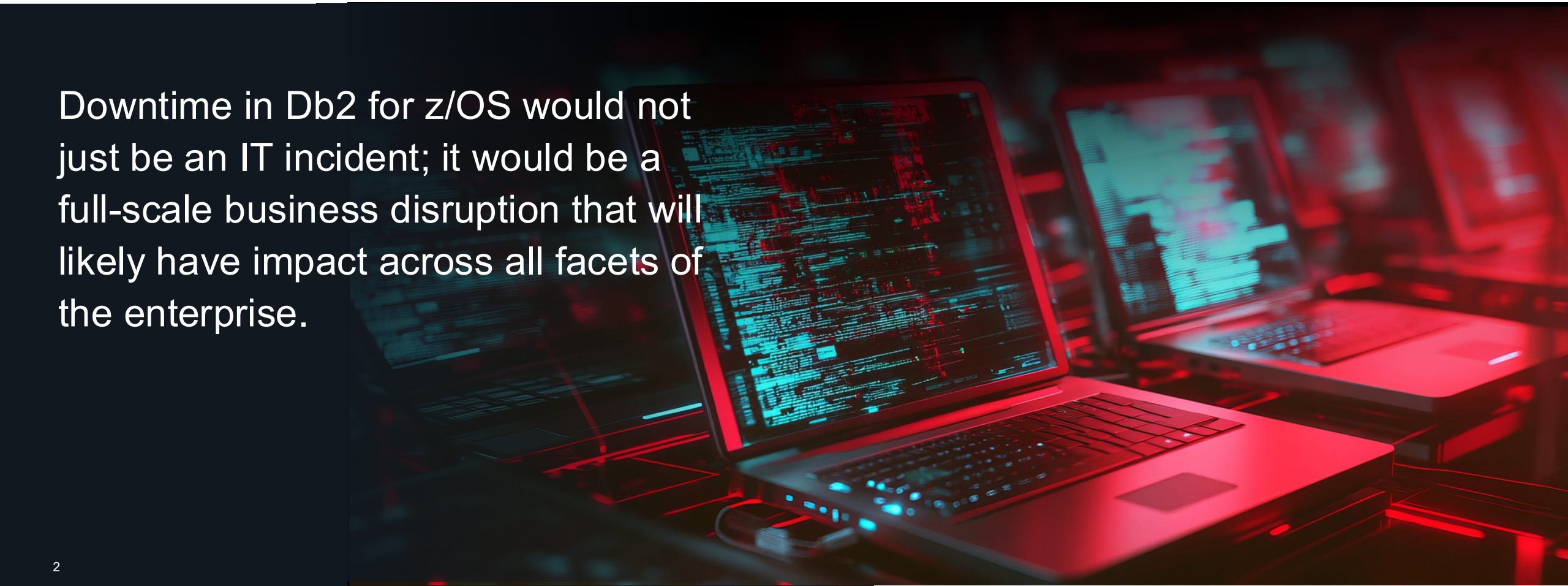
February 2026

Db2 is the Heart of the Enterprise

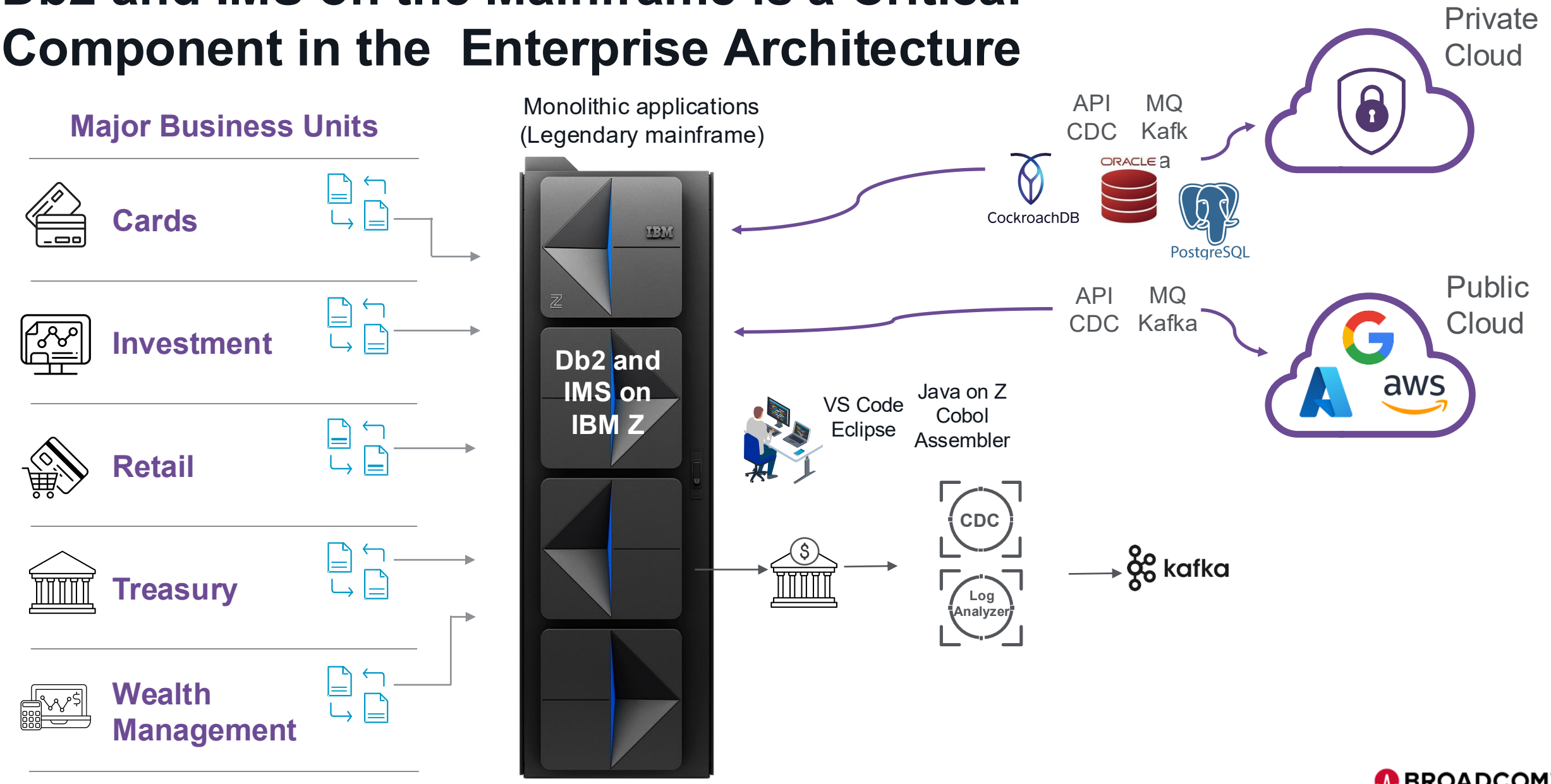
System of record and single source of truth

All revenue generating transactions go through Db2 on the mainframe

Downtime in Db2 for z/OS would not just be an IT incident; it would be a full-scale business disruption that will likely have impact across all facets of the enterprise.



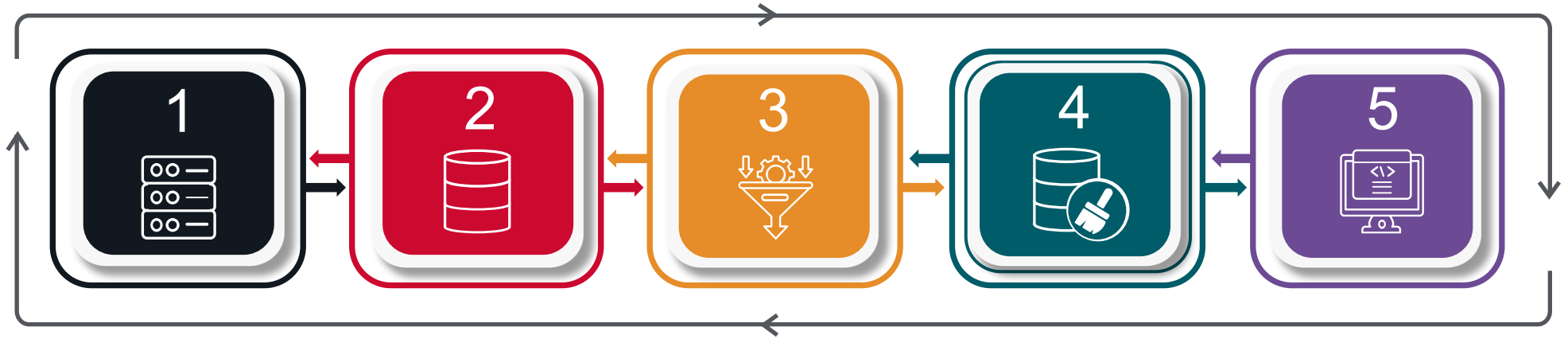
Db2 and IMS on the Mainframe is a Critical Component in the Enterprise Architecture



BROADCOM MAINFRAME SOFTWARE

Good Db2 Hygiene Requires Proactive Support: Key Areas to Review and Act

Steps to ensure Db2 for z/OS Resiliency



1	2	3	4	5
Regular review of storage reports	Memory tuning to ensure sufficient bufferpool resources	Review SMF reports to identify and resolve potential CPU bottlenecks & anomalies	Regular object housekeeping to maintain clean, accurate and optimized Db2 catalog	Maintain optimal DDL and object parameters

BROADCOM MAINFRAME SOFTWARE

1. Mitigate potential storage issues with regular review

Monitor Data Set Size to prevent out-of-space conditions

- Maintain high availability using Database Analyzer
- Provide performance stability and scalability

Ensure index availability and integrity

- Prevent index outages
- Monitor index piece growth for improved performance and recoverability

Optimize storage utilization with compression while reducing recovery time objective:

- Recovery ESTIMATION and SIMULATION are HOT topics

Ensure there's enough space to prevent performance and scalability issues for data sets that continue to grow on Db2.

To do so, the DBA must monitor index size as well as ensure compression is applied where appropriate.

1. Sample SQL Statement for Storage Report from Db2 Catalog

```
-- Query for PBR storage report
select a.dbname AS QUALIFIER, a.tsname AS NAME , a.partition, DEC(Float((B.SPACE)) / 1048576,7,2)
as Size_GB, DEC(((FLOAT(b.space)))/C.DSSIZE * 100,  6,3)  as Pct_Space_Used
    from sysibm.systablepart a,sysibm.systablespacestats b,SYSIBM.SYSTABLESPACE C
where a.dbname = b.dbname
    and a.tsname = b.name
    and a.partition = b.partition
    AND A.DBNAME = C.DBNAME
    AND A.TSNAME = C.NAME
    and a.partition > 0
    AND C.TYPE = 'R'
AND A.TSNAME LIKE ('XYZ%') order by 5 desc

-- Query for PBG storage report
select a.dbname AS QUALIFIER, a.tsname AS NAME , MAX(c.maxpartitions) as Maxpart,
DEC(Float(SUM(B.SPACE)) / 1048576,7,2) as Size_GB, DEC(((FLOAT(SUM(b.space))))/
(MAX(C.DSSIZE) * max(c.maxpartitions)) * 100,  6,3)  as Pct_Space_Used
    from sysibm.systablepart a,sysibm.systablespacestats b,SYSIBM.SYSTABLESPACE C
where a.dbname = b.dbname
    and a.tsname = b.name
    and a.partition = b.partition
    AND A.DBNAME = C.DBNAME
    AND A.TSNAME = C.NAME
    and a.partition > 0
    AND C.TYPE = 'G'
AND A.TSNAME LIKE ('XYZ%')
group by a.dbname, a.tsname
order by 5 desc
```

SQL Statement to use to generate report for Percent Space Used for PBR or PBG tablespaces.

1. Sample Storage Report Output from Db2 Catalog

Example of Storage PBR Report

QUALIFIER	NAME	PARTITION	SIZE_GB	PCT_SPACE_USED ↓
[redacted]	[redacted]	35	47.58	74.349
[redacted]	[redacted]	1	1.71	42.985
[redacted]	[redacted]	39	0.64	16.239
[redacted]	[redacted]	7	0.64	16.239
[redacted]	[redacted]	5	0.64	16.239
[redacted]	[redacted]	39	0.62	15.501

Example of Storage PBG Report

QUALIFIER	NAME	MAXPART	SIZE_GB	PCT_SPACE_USED
[redacted]	[redacted]	25	22.06	22.060
[redacted]	[redacted]	25	21.33	21.339
[redacted]	[redacted]	25	19.99	19.998
[redacted]	[redacted]	25	17.56	17.563
[redacted]	[redacted]	25	17.56	17.563
[redacted]	[redacted]	25	16.29	16.297

SQL Statement to use to generate report for Percent Space Used for PBR or PBG tablespaces.

BROADCOM MAINFRAME SOFTWARE

Another Storage Report Output from the Db2 Catalog - Tablespaces

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
TABLE_NAME	TABLESP/	PARTITIO	MAXPART	DSSIZE_MB	MAX_SIZE_MB	USED_MB	USED_PERCENTAGE	TOTALROWS	COMPRES	R	REORGIN	REORGDE	REORGUI	REORGUI	REORGUIPCT
PTI.ALOGFILE	PTDB.ALC	0	0	2048	65536	1	0	349		2	626	277	0	37	5
PTI.ALOGRANGE_1700	PTDB.ALC	0	0	2048	65536	1	0	836		2	3132	2296	0	189	6
PTI.BPLOG_0203	PTDB.PTI	0	0	2048	65536	3	0	18246		2	38156	19910	60914	6715	17
PTI.PDT_DYNAMREQ_160	PTDB.PDT	1	4096	4096	4096	0	0	5	Y	2	5	0	0	0	0
PTI.PDT_DYNAMREQ_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	100	100	0	0	0
PTI.PDT_DYNAMTXT_160	PTDB.PDT	1	4096	4096	4096	0	0	1	Y	2	1	0	0	0	0
PTI.PDT_DYNAMTXT_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	54	54	0	0	0
PTI.PDT_ERRORTXT_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	84	84	0	0	0
PTI.PDT_OBJECT_160	PTDB.PDT	1	4096	4096	4096	0	0	28	Y	2	28	0	0	0	0
PTI.PDT_SQLERROR_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	352	352	0	0	0
PTI.PDT_STANDARD_160	PTDB.PDT	1	4096	4096	4096	0	0	79	Y	2	79	0	0	0	0
PTI.PDT_STANDARD_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	701	701	0	0	0
PTI.PDT_STANTEXT_160	PTDB.PDT	1	4096	4096	4096	0	0	42	Y	2	42	0	0	0	0
PTI.PDT_STANTEXT_180	PTDB.PDT	0	0	2048	65536	0	0	0	Y	2	89	89	0	0	0
PTI.PTALT_ACM_0160	PTDB.PTA	0	0	2048	65536	0	0	7		2	14	7	0	0	0
PTI.PTALT_SYSTBL_0160	PTDB.PTI	0	0	2048	65536	0	0	481		2	606	125	0	0	0
PTI.PTAN_PRFM_0201	PTDB.PTI	0	0	2048	65536	22	0	67		2	67	0	0	0	0
PTI.PTAN_SQL_0201	PTDB.PTI	0	0	2048	65536	56	0	151763		2	1002896	851133	0	39827	3
PTI.PTAN_SQL_1200	PTDB.PTI	0	0	2048	65536	45	0	56310		2	61165	4855	58468	54789	89
PTI.PTAN_STMT_0201	PTDB.PTI	0	0	2048	65536	26	0	64466		2	415364	350898	50	8030	1
PTI.PTAN_STMT_1200	PTDB.PTI	0	0	2048	65536	24	0	55928		2	60785	4857	58118	53245	87
PTI.PTGL500_HISTORY	PTDB.PTC	0	0	2048	65536	2	0	502		2	502	0	0	0	0
PTI.PTGL600_RESTART	PTDB.PTC	0	0	2048	65536	0	0	26		2	77700	77674	43000	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	1	0	4096	4096	2	0	199		2	169179	168990	699591	3	0
PTI.PTGL600_RESTART2	PTDB.PTC	2	0	4096	4096	2	0	0		2	153	153	309	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	3	0	4096	4096	2	0	13		2	4141	4128	2358	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	4	0	4096	4096	2	0	0		2	229	229	51	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	5	0	4096	4096	2	0	52		2	100211	100159	113024	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	8	0	4096	4096	2	0	26		2	2552	2526	0	0	0
PTI.PTGL600_RESTART2	PTDB.PTC	9	0	4096	4096	2	0	0		2	6	6	0	0	0
PTI.PTLOGLABEL_1800	PTDB.PTI	0	0	2048	65536	0	0	2		2	2	0	5	0	0
PTI.PTLOGLABEL_VERSION_18	PTDB.PTI	0	0	2048	65536	1	0	5		2	5	0	0	0	0
PTI.PTLOG_CTSTATS_1105	PTDB.PTF	0	0	2048	65536	2	0	788	Y	2	798	10	0	0	0
PTI.PTLOG_DASTATS_1105	PTDB.PTF	0	0	2048	65536	2	0	788	Y	2	798	10	0	0	0
PTI.PTLOG_MAIN_1500	PTDB.PTI	0	0	2048	65536	1320	2	3996809		2	4004759	7950	0	170	0
PTI.PTLOG_RDAMSG_1105	PTDB.PTF	0	0	2048	65536	2	0	788	Y	2	798	10	0	0	0

Another Storage Report Output from the Db2 Catalog - Indexes

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
TBNAME	INDEX	PARTITIO	UNIQUE	CLUSTER	CLRATIO	PCTFREE	FREEPAG	NLEVELS	PGSIZE	PIECESIZ	USED_MB	USED_PIE	IX_COLUM	REORGLA	LASTUSEI	TOTALEN	REORGIN	REORGDI	REORGAF	REORGLE	REORGLEAF
ADMIN_TA	SYSIBM.A	0	U	Y	0	10	0	2	4	2097152	0	1	TASKID	2013-05-1	NULL	0	0	0	0	0	0
ADMIN_TA	SYSIBM.A	0	U	Y	0	10	0	2	4	2097152	0	1	TASKID,	2013-05-1	NULL	0	0	0	0	0	0
DBDR	SYSIBM.D	0	U	N	0	0	0	NULL	4	2097152	0	1	DBID, SE	NULL	41752	11789	0	0	0	7	16
DSNPROC	SYSIBM.D	0	U	N	0	10	0	2	4	2097152	0	1	PROGNAM	2013-07-1	NULL	2	2	0	2	0	0
DSNRLST	SYSIBM.D	0	U	Y	0	10	0	2	4	2097152	0	1	RLFFUNC,	2013-05-1	NULL	0	0	0	0	0	0
DSN_DET	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_FILT	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_FUN	SYSIBM.F	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_PGR	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_PGR	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_PRE	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	41655	0	0	0	0	0	0
DSN_PRE	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_PRE	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_PRO	SYSIBM.D	0	U	N	100	10	0	2	4	2097152	0	1	PROFILEI	2013-05-1	41750	79	325	246	11	0	1
DSN_PRO	SYSIBM.D	0	D	N	100	10	0	2	4	2097152	0	1	PROFILE_	2013-05-1	41709	23	163	140	15	0	0
DSN_PRO	SYSIBM.D	0	P	N	100	10	0	2	4	2097152	0	1	PROFILEI	2013-05-1	41750	23	41	18	11	0	0
DSN_PTA	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_QUE	SYSIBM.D	0	U	N	-2	10	0	2	4	4194304	0	1	AUXID, A	2013-05-1	NULL	0	0	0	0	0	0
DSN_QUE	SYSIBM.D	0	U	N	-2	10	0	2	4	4194304	0	1	AUXID, A	2013-05-1	NULL	0	0	0	0	0	0
DSN_QUE	SYSIBM.D	0	U	N	-2	10	0	2	4	4194304	0	1	AUXID, A	2013-05-1	NULL	0	0	0	0	0	0
DSN_QUE	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_QUE	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_SOR	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0
DSN_SOR	SYSIBM.D	0	D	N	0	10	0	2	4	2097152	0	1	QUERYNO	2013-05-1	NULL	0	0	0	0	0	0

BROADCOM
MAINFRAME SOFTWARE

1. Take steps to right size tablespace, index, and ensure compression

- Monitor tablespace size against its DSSIZE attribute
  - Identify tablespaces approaching maximum capacity
  - If more than 5 parts for PBG or more than 80% for PBR
    - Alter DSSIZE to adjust then Reorg
- Index Piecesize
  - PIECESIZE can run out of space (e.g., large non-partitioned indexes over partitioned tables) leading to failures for transactions that require index updates
  - Identify indexes that are experiencing rapid expansion and allow for proactive REBUILD INDEX operations to larger PIECESIZE
- Compression
  - Tablespace
  - LOB
  - Indexspace

Key actions include performing a reorg on tablespaces that reach a certain size, ensuring indexes do not run out of space, and freeing up storage with compression that helps with bufferpool density.

2. Memory tuning drives performance and resilience

- Overall aim to ensure Db2 has sufficient memory resources to minimize I/O, reduce CPU overhead, and efficiently process transactions
- Buffer pools need to be tuned to reduce I/O bottlenecks, improve response times, and offers performance stability
- Insufficient memory can cause:
  - Db2 Sort Work to failover to disk
  - RID Pool overflows
  - Excessive EDM Pool reloads

Collaboration needed due to roles.

Define alerts – RID failures expensive

Without sufficient memory in Db2, applications will not be able to respond to meet SLAs.

A big part of memory tuning involves tuning the buffer pools and the objects aligned to each one.

2. Take steps to right-size parameters of critical memory components

Buffer pool Tuning

- I/O Reports: use Db2 performance monitoring tool – e.g., Detector, Subsystem Analyzer and Sysview/Db2
- Simulations: helps predict optimal bufferpool sizes
- BP Parms: understand thresholds and settings
- BP I/O Reporting

Db2 Sort Work: Tablespace Allocations

RID Pool: Failures

EDM Pool: Hit Ratio

Dynamic Statement Cache: MAXKEEPD

Low BP-HITRATIO not necessarily bad (isolate random).

Use SIMULATION for verification.

Michal Bialecki IDUG presentation tuning Dynamic SQL.

Bufferpool tuning is the key piece to ensure application performance and Db2 resilience.

The different parameters give a sense for whether the bufferpool is large enough to provide the expected performance level and to help eliminate I/O activity.

2. Sample SQL Statements to Query Db2 Catalog for Bufferpool Allocations

```
-- Query to give you TS objects per bufferpool
 SELECT  A.NAME as TSNAME, B.NAME as TABLE_NAME,
        B.DBNAME , A.BPOOL, B.CARD as ROWS , B.NPAGES
   FROM SYSIBM.SYSTABLESPACE A , SYSIBM.SYSTABLES B
  WHERE ( A.DBID = B.DBID
    AND A.NAME = B.TSNAME
    AND B.TYPE IN ( 'T' , 'G' , 'X' , 'M' , 'P' , 'C' , 'H' , 'R' , 'D' ) )
  ORDER BY A.BPOOL, B.DBNAME , B.NAME , B.CREATOR WITH CS
```

```
-- Query to give you IX objects per bufferpool
  SELECT  A.DBNAME , A.TBNAME as TABLE_NAME, A.NAME as INDEXNAME,
  A.INDEXSPACE , A.BPOOL , A.NLEVELS  FROM SYSIBM.SYSINDEXES A
  ORDER BY A.BPOOL, A.NAME , A.CREATOR WITH CS;
```

```
-- Query to report TS count per Bufferpool
SELECT  BPOOL, COUNT(*)
   FROM SYSIBM.SYSTABLESPACE
  GROUP BY BPOOL
ORDER BY BPOOL
  WITH UR
```

```
-- Query to report TS count per Bufferpool
SELECT  BPOOL, COUNT(*)
   FROM SYSIBM.SYSINDEXES
  GROUP BY BPOOL
ORDER BY BPOOL
  WITH UR
```

Sample queries from Db2 Catalog that will help with organizing BP Object data for alignment between objects and bufferpools

3. Monitor CPU to remediate performance bottlenecks

Proactive review and tuning of CPU helps to identify and resolve bottlenecks or high consuming workloads before they impact critical workloads or escalate into major performance issues

Overall process

- Establish a baseline
- Monitor and report at a regular cadence
- Anomaly detection
- Drill-down analysis
- Formulate hypothesis
- Devise and implement tuning solution
- Verify and validate solution
- Document findings, actions, and outcomes

Need to know “normal behavior” and not chase ghosts.

Not only CPU – GETP, ASYNC IO, SYNC IO changes vital to track.

Look at AVERAGE too – not only TOTALS.

Being able to detect anomalies in CPU usage can help the DBA identify performance bottlenecks or workload anomalies before the application end user starts to experience a slow down.

3. Where do I begin to look for issues that can impact CPU?

Start with SMF Reports

- Monitoring tools like Detector can process and present SMF data in a much more digestible and actionable format

Identify

- Top 10 Consumers
- Deadlocks / Unavailable Resources
- High Getpages (Synchronous and Asynchronous I/O)

Index Design

Review Access Paths *(RC/Query QECOMP)*

Bufferpool Object Placement

Latch Contentions

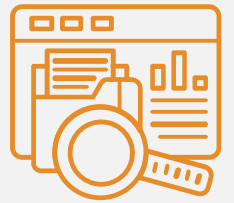
Reports for Stats Coverage / Stats Feedback

- Invalidate Statement Cache

TOP-10 probably not sufficient – dynamic workload.

Offload to reporting tables.

AP review & Virtual IX. IX INCLUDE challenges.



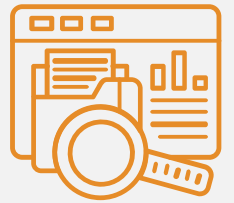
Db2 produces a lot of data that tells the DBA how the engine is doing; however, those SMF records are not easy to interpret and that's where a tool like Detector comes in. Detector captures, stores, and analyzes Db2 performance data.

3. Sample Db2 Catalog Queries to find Index Optimization

```
-- Query to ensure all columns of primary index
SELECT CHAR(CREATOR,8), CHAR(NAME,20) FROM
SYSIBM.SYSINDEXES
EXCEPT (SELECT CHAR(A.IXCREATOR,8),
CHAR(A.IXNAME,20) FROM
SYSIBM.SYSKEYS A, SYSIBM.SYSINDEXES B
WHERE A.IXCREATOR = B.CREATOR
AND A.IXNAME = B.NAME
AND (B.UNIQUERULE <> 'D'
OR (A.COLNO, A.COLSEQ) NOT IN (SELECT C.COLNO,
C.COLSEQ FROM
SYSIBM.SYSKEYS C, SYSIBM.SYSINDEXES D
WHERE C.IXCREATOR = D.CREATOR
AND C.IXNAME = D.NAME
AND B.TBCREATOR = D.TBCREATOR
AND B.TBNAME = D.TBNAME
AND D.UNIQUERULE = 'P')));
```

```
--INDEXES THAT HAVE NOT BEEN USED FOR AT LEAST 12 MONTHS
SELECT CHAR(A.CREATOR, 8) AS CREATOR, CHAR(A.NAME, 30) AS
NAME, MAX(A.LASTUSED)
 FROM SYSIBM.SYSINDEXSPACESTATS A, SYSIBM.SYSINDEXES B
 WHERE A.CREATOR = B.CREATOR AND A.NAME = B.NAME
 AND B.UNIQUERULE = 'D' AND A.DBNAME LIKE 'XYZ%'
GROUP BY A.CREATOR, A.NAME HAVING MAX(A.LASTUSED) < CURRENT
DATE - 12 MONTHS
ORDER BY 3, 1, 2;
```

```
-- Query to find index with not enough columns
SELECT DISTINCT CHAR(IXCREATOR,8), CHAR(IXNAME,30)
FROM SYSIBM.SYSKEYS A, SYSIBM.SYSINDEXES B
WHERE A.IXCREATOR = B.CREATOR
AND A.IXNAME = B.NAME
AND B.UNIQUERULE = 'D'
AND EXISTS (SELECT 1 FROM SYSIBM.SYSINDEXES I
WHERE B.NAME = I.NAME
AND B.CREATOR = I.CREATOR
AND I.UNIQUERULE = 'P')
AND NOT EXISTS (SELECT 1 FROM
SYSIBM.SYSKEYS C, SYSIBM.SYSINDEXES D
WHERE C.IXCREATOR = D.CREATOR
AND C.IXNAME = D.NAME
AND B.TBNAME = D.TBNAME
AND B.TBCREATOR = D.TBCREATOR
AND D.UNIQUERULE = 'P'
AND (C.COLNO, C.COLSEQ) NOT IN (SELECT COLNO, COLSEQ
FROM SYSIBM.SYSKEYS E
WHERE E.IXCREATOR = B.CREATOR
AND E.IXNAME = B.NAME));
```



Sample Db2 Queries to help optimize Index configurations and performance of indexes, including usage for obsolesce reporting

4. Maintain clean, accurate, and optimized Db2 catalog

- Db2 catalog is the heart of your Db2 subsystem containing:
  - Metadata about Db2 objects
  - Relationships across objects
  - Statistics about those objects
- RC/Compare helps ensure consistency as we migrate objects across environments *(BI-directional)*
- A catalog that is overgrown, fragmented, cluttered by obsolete entries, or inaccurate can impact performance and stability of Db2
  - IBM recommendation: Reorg catalog and directory 1-2x per yr

As the “heart” of a Db2 subsystem, maintaining a clean, accurate, and optimized catalog is foundational to ensuring Db2 resiliency – i.e., its ability to withstand and recover from various failures and disruptions while maintaining acceptable service levels.

4. How are catalog clean-ups evaluated and executed?

Object Clean-ups

- Modify Recovery – clean up recovery information from Db2 Catalog *(be careful with DELETE-DATASETS)*
- Modify Statistics – clean up statistics history from Db2 Catalog

RTS Reports

- SYSTABLESPACESTATS: Real-time tablespace information
  - Last Reorg Time, Total Rows, Last Data Change
- SYSINDEXSPACESTATS: Real-time index information
  - Reorg Num Levels, Last Used Index

Package Clean Up

- How many version to keep? Plan Management
- Still Using OptHints?
- LastUsed in SYSPACKAGE for clean up
  - ✓ Free Package commands: FREE PACKAGE (*.*.*.(*)) PLANMGMTSCOPE(INACTIVE) INVALIDONLY(YES)

DBAs need to evaluate “stale” objects that are no longer actively used. They take them through a review process before taking action to remove them. Removing those objects help to keep a more efficient catalog that operates more efficiently.

4. Sample Db2 Queries RTS Tables

```
 --Query to Review Utility Activity on Tablespace
 SELECT DBNAME, NAME, PARTITION, TOTALROWS, LASTDATACHANGE,
REORGSCANACCESS, LOADRLASTTIME, REORGLASTTIME, COPYLASTTIME, SPACE
 FROM SYSIBM.SYSTABLESPACESTATS
 WHERE DBNAME LIKE 'XYZ%'
 ORDER BY 2
```

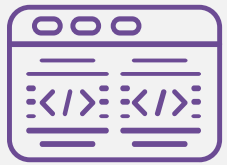
```
-- Query for INDEX compress candidates
SELECT SUBSTR(CREATOR,1,8) AS CREATOR, SUBSTR(NAME,1,20) AS NAME,
SUM(REORGINSERTS) / MAX(DAYS(CURRENT TIMESTAMP) - DAYS(MAX(
COALESCE(MAX(REORGLASTTIME),'0001-01-01-00.00.00.000000'),
COALESCE(MAX(REBUILDLASTTIME),'0001-01-01-00.00.00.000000'),
COALESCE(MAX(LOADRLASTTIME),'0001-01-01-00.00.00.000000'))),1)
AS INSERTS_PER_DAY,
SUM(REORGDELETES) / MAX(DAYS(CURRENT TIMESTAMP) - DAYS(MAX(
COALESCE(MAX(REORGLASTTIME),'0001-01-01-00.00.00.000000'),
COALESCE(MAX(REBUILDLASTTIME),'0001-01-01-00.00.00.000000'),
COALESCE(MAX(LOADRLASTTIME),'0001-01-01-00.00.00.000000'))),1)
AS DELETES_PER_DAY
,DEC(FLOAT(SUM(SPACE))/1048576,7,3) AS SPACE_GB
FROM SYSIBM.SYSINDEXSPACESTATS
WHERE DBNAME LIKE 'XYZ%'
GROUP BY CREATOR, NAME
ORDER BY 5 DESC;
```

Sample Db2 Queries from RTS tables to provide valuable information on tablespace utility activity and index activity from real time data

BROADCOM MAINFRAME SOFTWARE

5. Proactive support of DDL and Utility Parms are fundamental to a resilient Db2 environment

- DDL (Data Definition Language) defines structure of Db2 objects
  - Design Db2 objects with appropriate DDL parameters
  - Review Default Parms *(site standard COMPARE RULES & Global Changes)*
- Utility Parameters govern execution and behavior of Db2 utilities
  - Regularly execute maintenance utilities with well-chosen parameters *(automation is key)*
  - Review Zparms where applicable



Db2 objects designed with the appropriate DDL along with well-chosen utility parameters for object maintenance ensures a resilient Db2 environment.

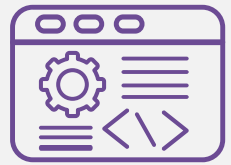
5. Key characteristics to define for object and utility parameters

Object Parms

- TS Parms
  - Trackmod No – not taking incrementals?
  - Compress Yes: LOB
  - Pagenum: Absolute (PBR) or Relative (RPN)
  - PQTY -1 SQTY -1 = Db2 Managed
  - PCTFREE / FREEPAGE for UPDATE -1
- IX Parms
  - PQTY -1 SQTY -1 = Db2 Managed
  - PCTFREE / FREEPAGE / PCTFREE for UPDATE
  - Piecesize
  - Compress Yes? Depends: SQL Activity

Utility Parms

- Statistics: Loads / Reorg / Rebuild IX and Invalidate Statement Cache



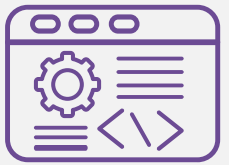
Setting the appropriate parameters here help to ensure underlying data sets have room to grow. Managing actions through DDL with the appropriate object parameters (TS Parms and IX Parms) is a big part of proactive support.

5. Sample Db2 Catalog Queries to Generate Control Statements

```
-- SQL to generate for FREEPAGE 10, PCTFREE 15, UPDATE -1 alter statements
SELECT 'ALTER TABLESPACE '||STRIP(DBNAME)||'.'||STRIP(NAME)||' FREEPAGE 10; '
 FROM SYSIBM.SYSTABLESPACE
WHERE (FREEPAGE > 10 OR FREEPAGE = 0)
AND DBNAME LIKE 'XYZ%'
UNION
SELECT 'ALTER TABLESPACE '||STRIP(DBNAME)||'.'||STRIP(NAME)||' PCTFREE '||MAX(PCTFREE,15)||' FOR UPDATE -1;'
FROM SYSIBM.SYSTABLESPACE
WHERE (DBNAME LIKE 'XYZ%')

-- SQL to generate for Primary / Secondary QTY for Tablespaces
SELECT 'ALTER TABLESPACE '
  CONCAT RTRIM(DBNAME)
  CONCAT '.'
  CONCAT RTRIM(NAME)
  CONCAT ' PRIQTY -1 SECQTY -1; COMMIT;'
FROM SYSIBM.SYSTABLESPACE
WHERE DBNAME LIKE 'XYZ%'

-- SQL to generate for Repair statements for tablespaces
SELECT SUBSTR('  REPAIR SET TABLESPACE 'CONCAT STRIP(DBNAME) CONCAT
       '.' CONCAT STRIP(NAME) CONCAT ' NOCOPYPEND'
        CONCAT '                                        ',1,72)
   FROM SYSIBM.SYSTABLESPACE
  WHERE DBNAME LIKE 'XYZ%'
  UNION
SELECT SUBSTR('  REPAIR SET TABLESPACE 'CONCAT STRIP(DBNAME) CONCAT
       '.' CONCAT STRIP(NAME) CONCAT ' NOCHECKPEND'
        CONCAT '                                        ',1,72)
   FROM SYSIBM.SYSTABLESPACE
  WHERE DBNAME LIKE 'XYZ%'
```



Sample Db2 Catalog queries to help with generating Alter statement for tablespace and index spaces

BROADCOM MAINFRAME SOFTWARE

Summary and Takeaways

Maintaining a resilient Db2 environment is non-negotiable

Regularly review storage and compression

Tune memory

Review and tune for CPU consumption

Maintain a clean and accurate Db2 catalog

Define DDL and Utility Parms

A healthy Db2 keeps the enterprise humming. Perform these actions to take care of Db2 (or the results could be catastrophic).

We would like your feedback

Db2 13 priorities at your organization

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation

25

Thank You

Q&A