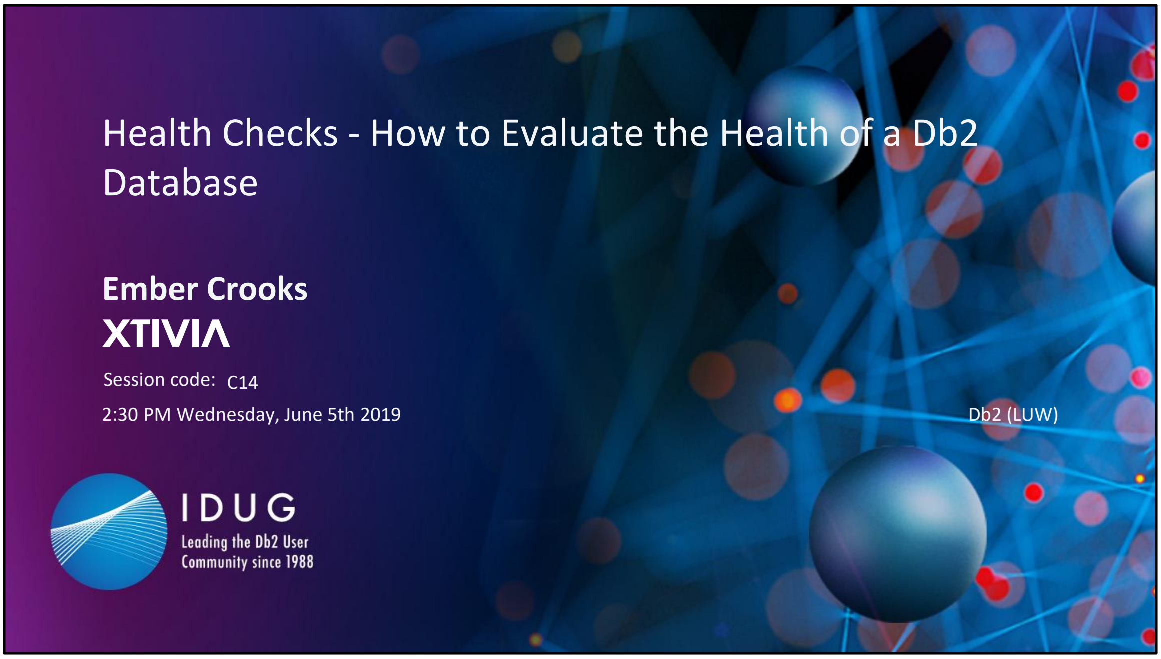


Health Checks - How to Evaluate the Health of a Db2 Database

Ember Crooks
XTIVIA

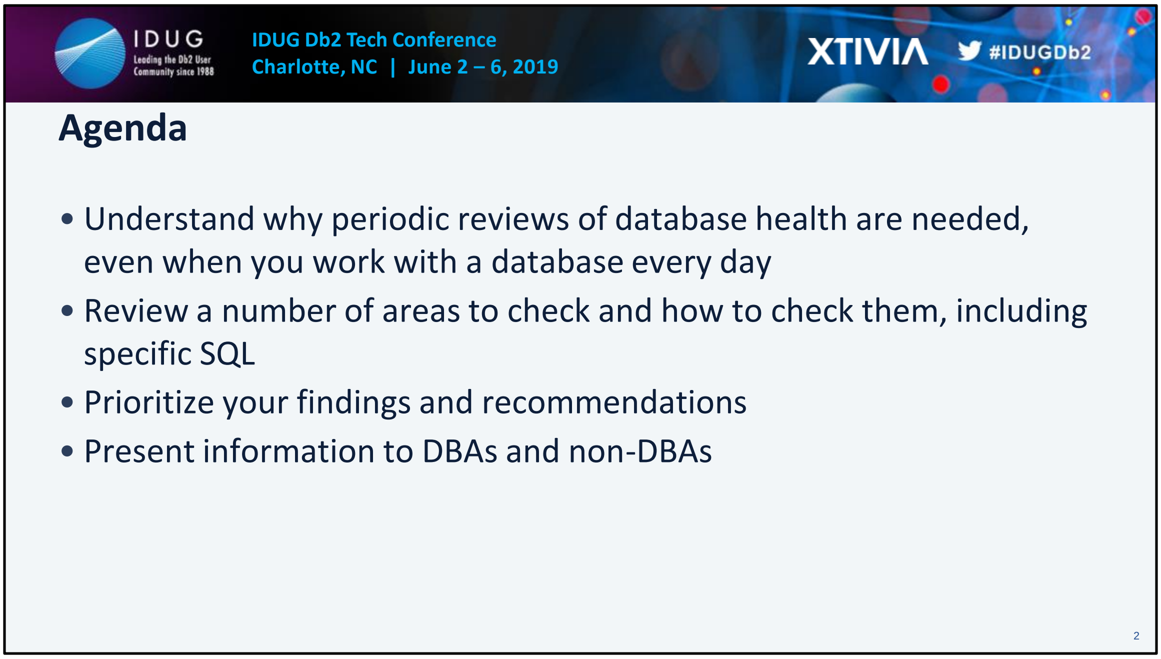
Session code: C14

2:30 PM Wednesday, June 5th 2019

Db2 (LUW)

IDUG
Leading the Db2 User Community since 1988

Every database can be improved upon. Sometimes a second set of eyes finds problems that we don't see when we work with a database every day. What should you check when evaluating the health of your database or a co-worker's database? How do you get all of the information you need? Join Ember to see what she looks at when doing a health check and why.



IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

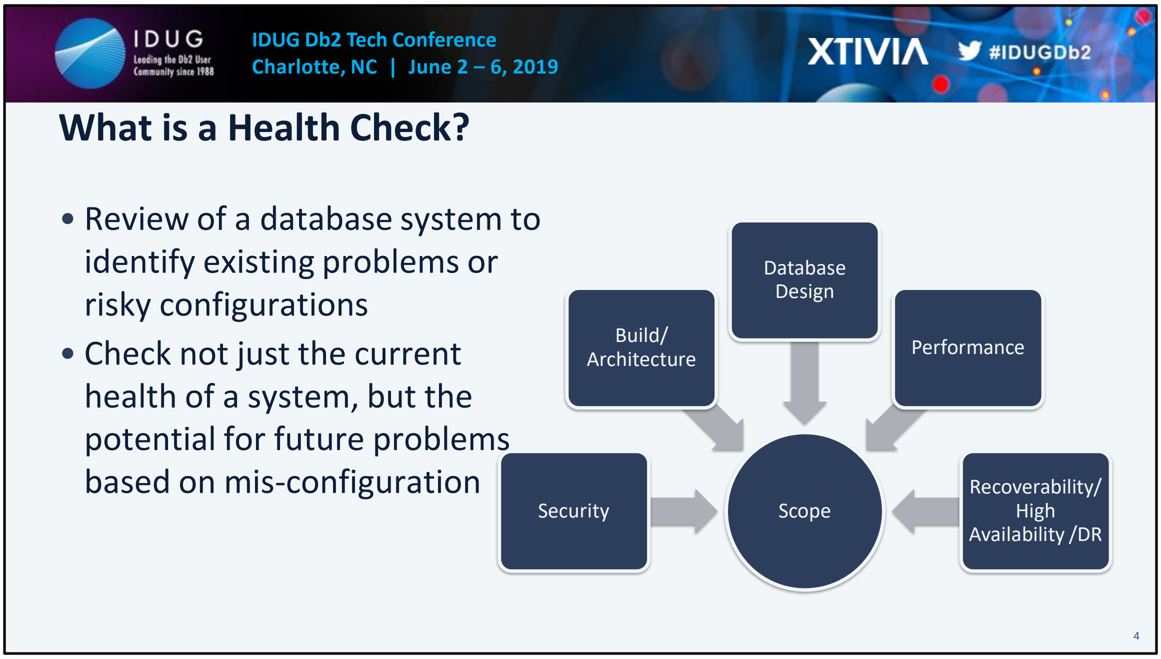
Agenda

- Understand why periodic reviews of database health are needed, even when you work with a database every day
- Review a number of areas to check and how to check them, including specific SQL
- Prioritize your findings and recommendations
- Present information to DBAs and non-DBAs

2

The amount of information to be covered here is enormous, so I'll take a high level look at most items I check for and offer in-depth details in only a few areas.

Health Checks: Who, What, When, and Why?



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

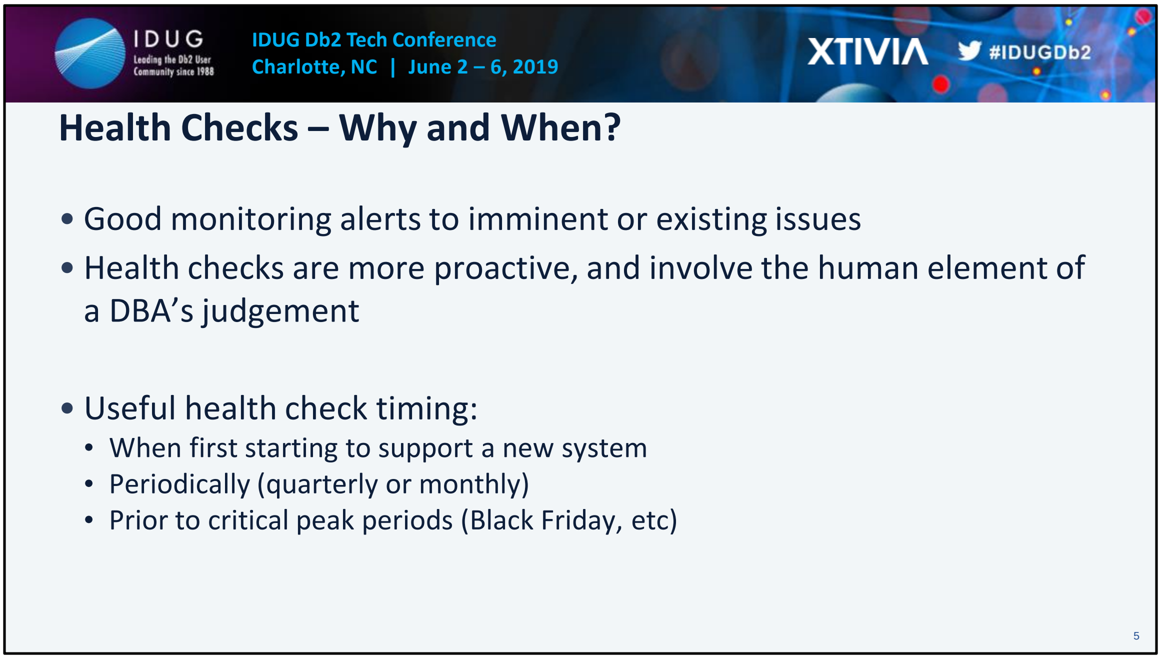
XTIVIA #IDUGDb2

What is a Health Check?

- Review of a database system to identify existing problems or risky configurations
- Check not just the current health of a system, but the potential for future problems based on mis-configuration

4

Part of the problem in categorizing health checks is that the things to check are an interlacing matrix where some items apply to multiple categories. For example, I’ll look at memory configuration both from a Build/Architecture perspective and from a Performance perspective. I’ll review tablespace details as part of Database Design, Performance, and Build/Architecture.



Health Checks – Why and When?

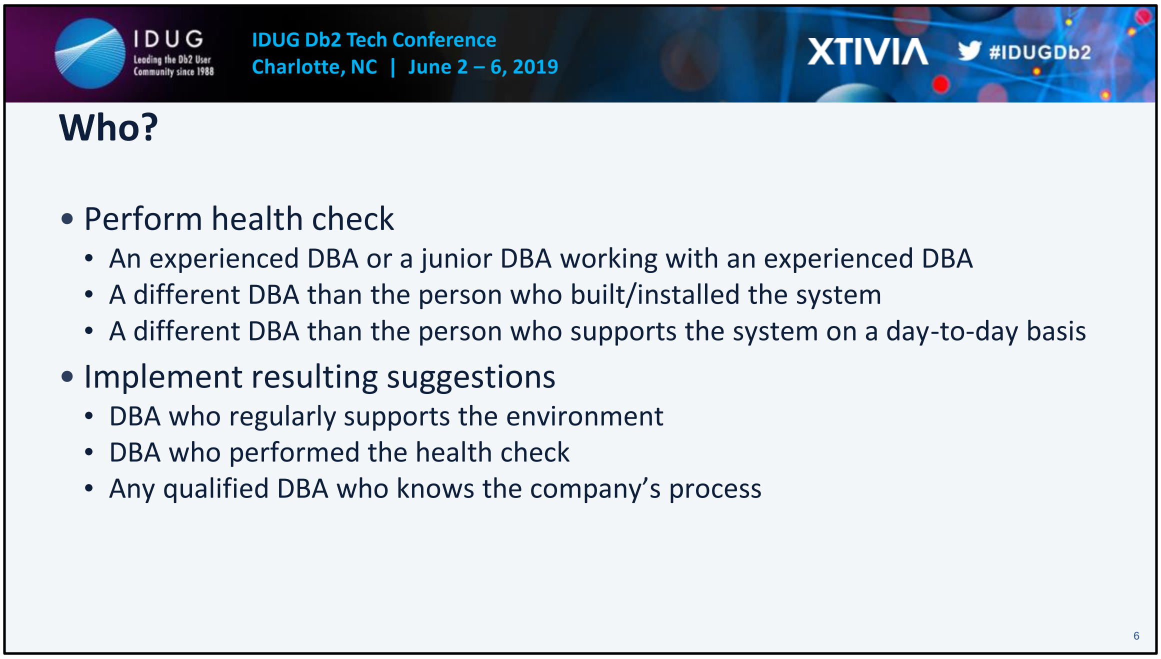
- Good monitoring alerts to imminent or existing issues
- Health checks are more proactive, and involve the human element of a DBA's judgement

- Useful health check timing:
  - When first starting to support a new system
  - Periodically (quarterly or monthly)
  - Prior to critical peak periods (Black Friday, etc)

Good monitoring should alert us of any emergent database issues. Health checks allow us to proactively assess the health of the database.

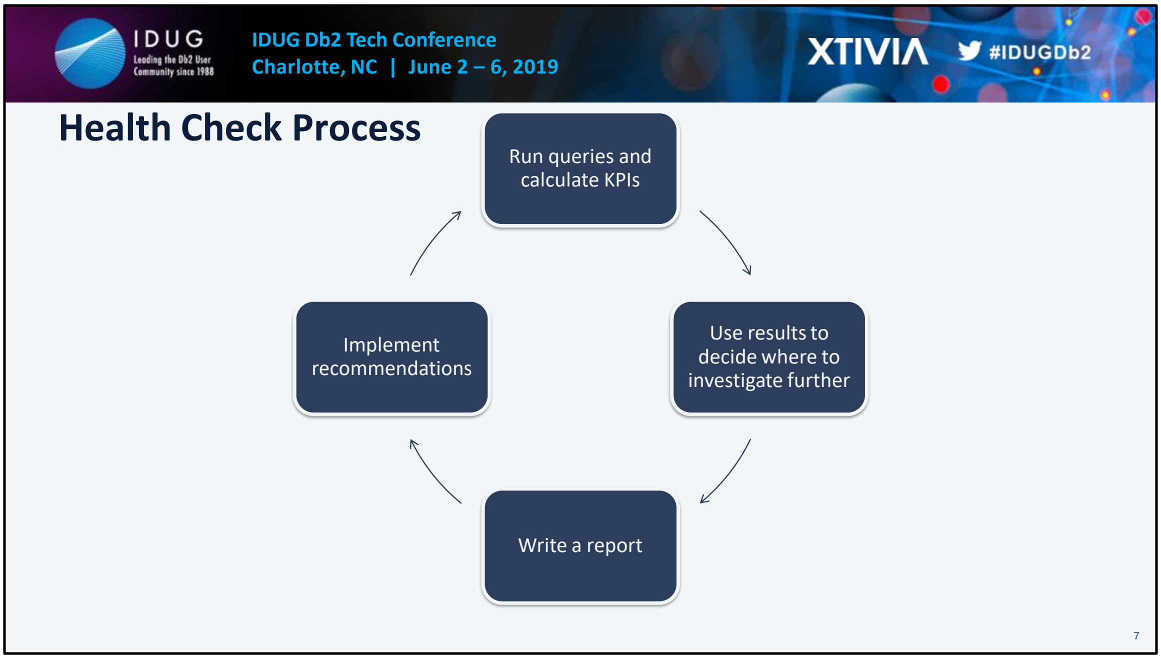
Health checks may be useful:
- When first starting to support a new system
- Periodically (quarterly or monthly)
- Prior to critical peak periods (Black Friday, etc)

Who should do health checks:
Ideally a different DBA than the DBA who does day-to-day support
Ideally a different DBA than the DBA who built the system



Who?

- Perform health check
  - An experienced DBA or a junior DBA working with an experienced DBA
  - A different DBA than the person who built/installed the system
  - A different DBA than the person who supports the system on a day-to-day basis
- Implement resulting suggestions
  - DBA who regularly supports the environment
  - DBA who performed the health check
  - Any qualified DBA who knows the company’s process



IDUG
Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA

#IDUGDb2

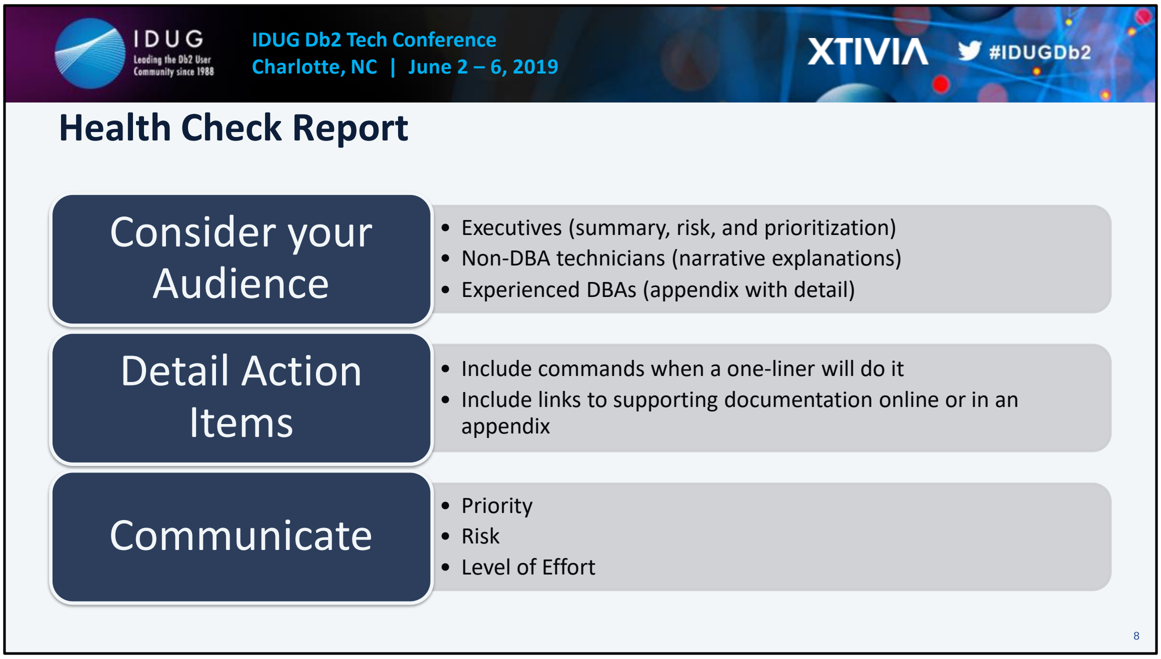
Health Check Process

Run queries and calculate KPIs

Use results to decide where to investigate further

Write a report

Implement recommendations



Health Check Report

Consider your Audience

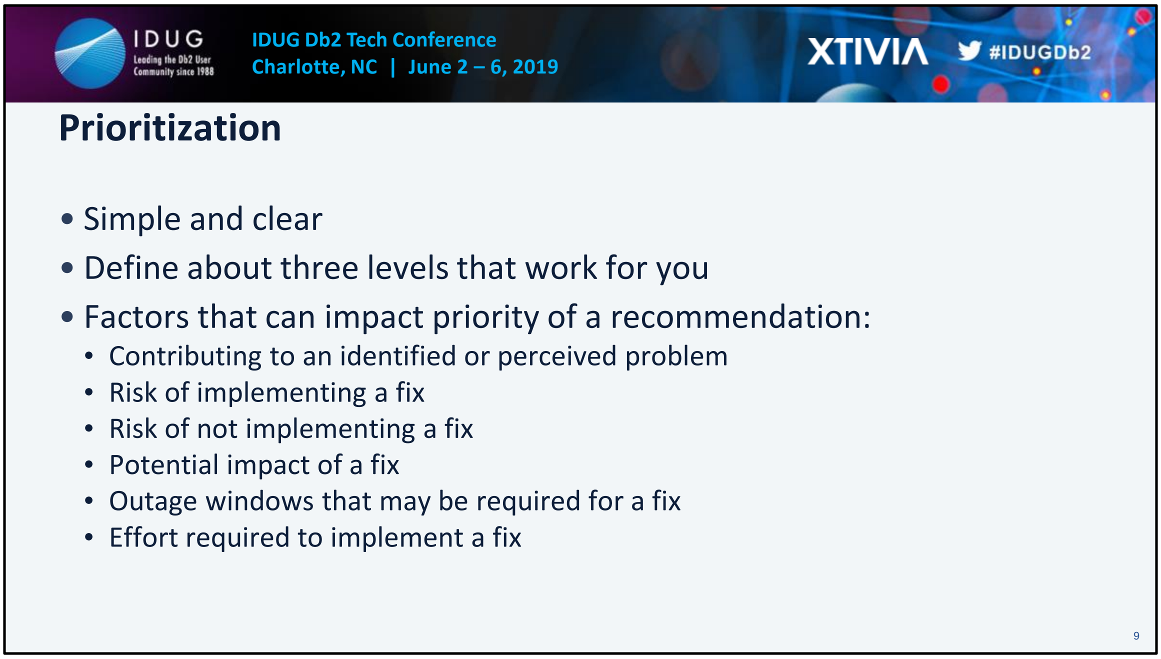
- Executives (summary, risk, and prioritization)
- Non-DBA technicians (narrative explanations)
- Experienced DBAs (appendix with detail)

Detail Action Items

- Include commands when a one-liner will do it
- Include links to supporting documentation online or in an appendix

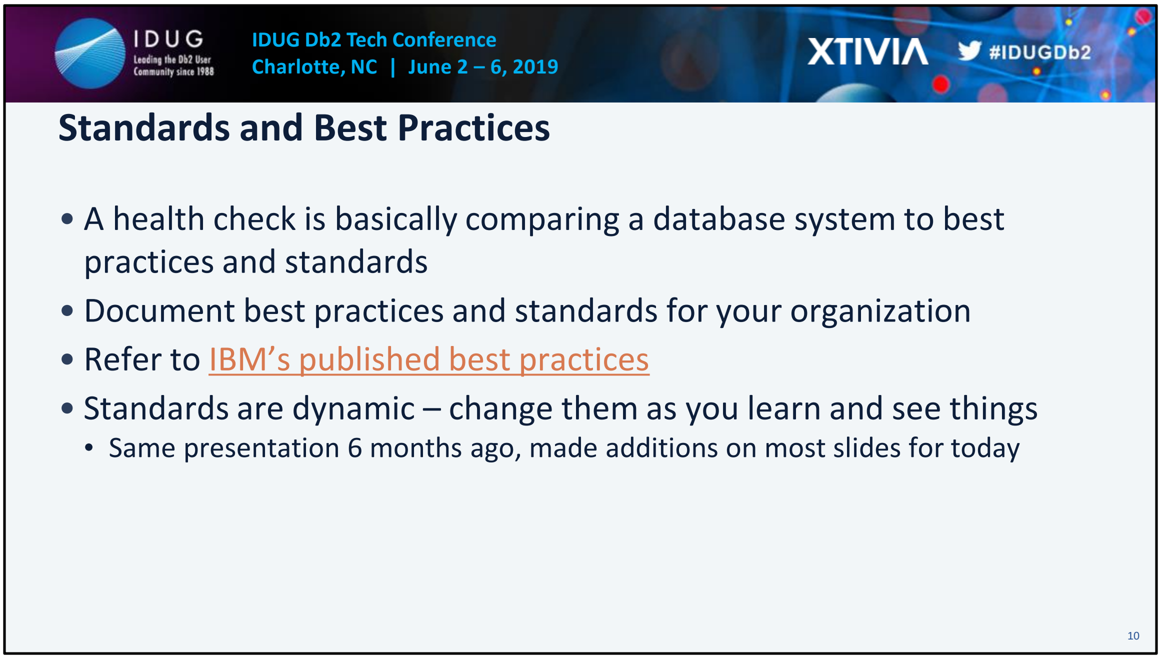
Communicate

- Priority
- Risk
- Level of Effort



Prioritization

- Simple and clear
- Define about three levels that work for you
- Factors that can impact priority of a recommendation:
  - Contributing to an identified or perceived problem
  - Risk of implementing a fix
  - Risk of not implementing a fix
  - Potential impact of a fix
  - Outage windows that may be required for a fix
  - Effort required to implement a fix



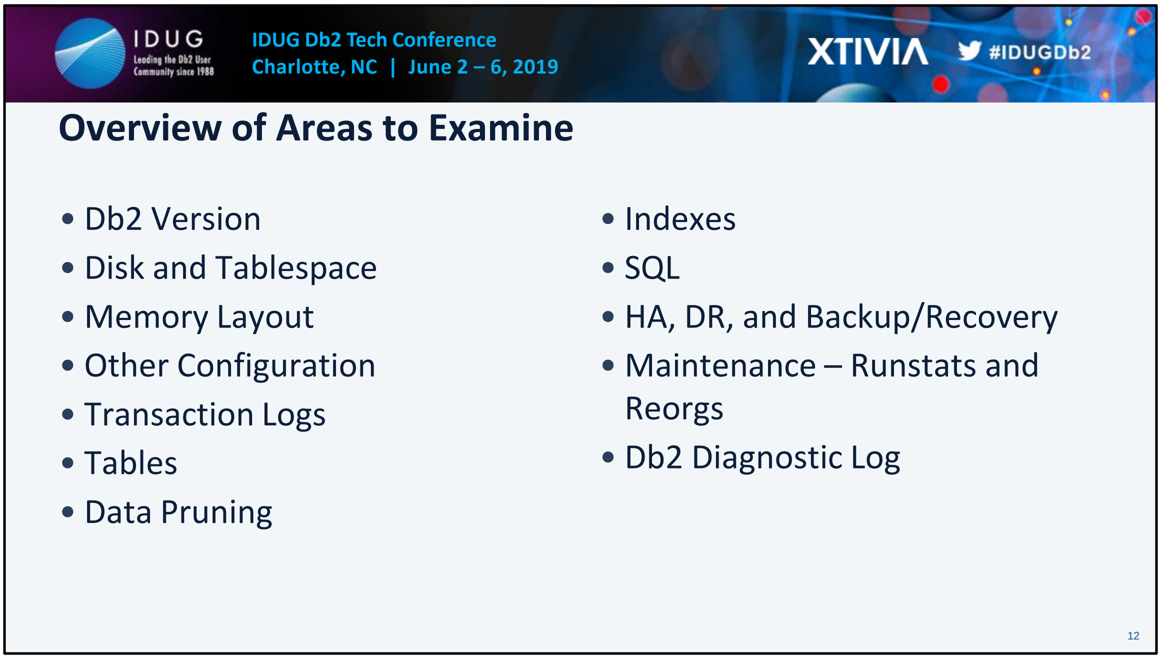
Standards and Best Practices

- A health check is basically comparing a database system to best practices and standards
- Document best practices and standards for your organization
- Refer to IBM’s published best practices
- Standards are dynamic – change them as you learn and see things
  - Same presentation 6 months ago, made additions on most slides for today

Defining standards and best practices for your organization gives you something to compare against when performing a health check.

Keep in mind the reasoning behind the standards and best practices when deciding whether a violation is worth addressing and the priority.

Specific Checks

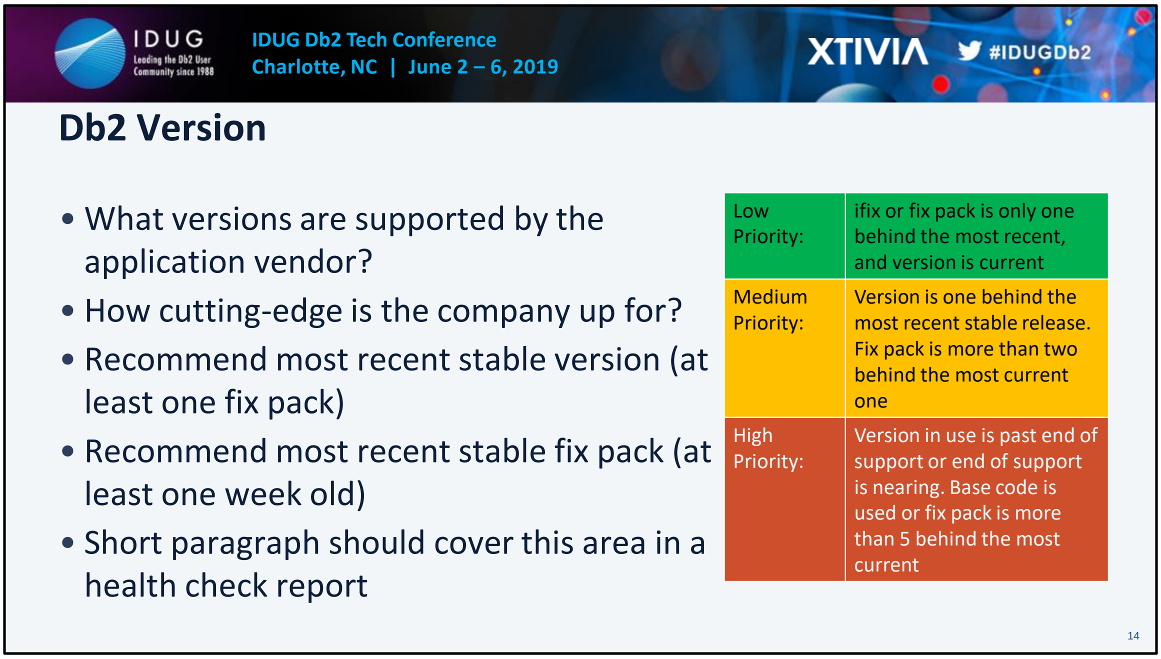


Overview of Areas to Examine

- Db2 Version
- Disk and Tablespace
- Memory Layout
- Other Configuration
- Transaction Logs
- Tables
- Data Pruning
- Indexes
- SQL
- HA, DR, and Backup/Recovery
- Maintenance – Runstats and Reorgs
- Db2 Diagnostic Log

Db2 Version and Patching

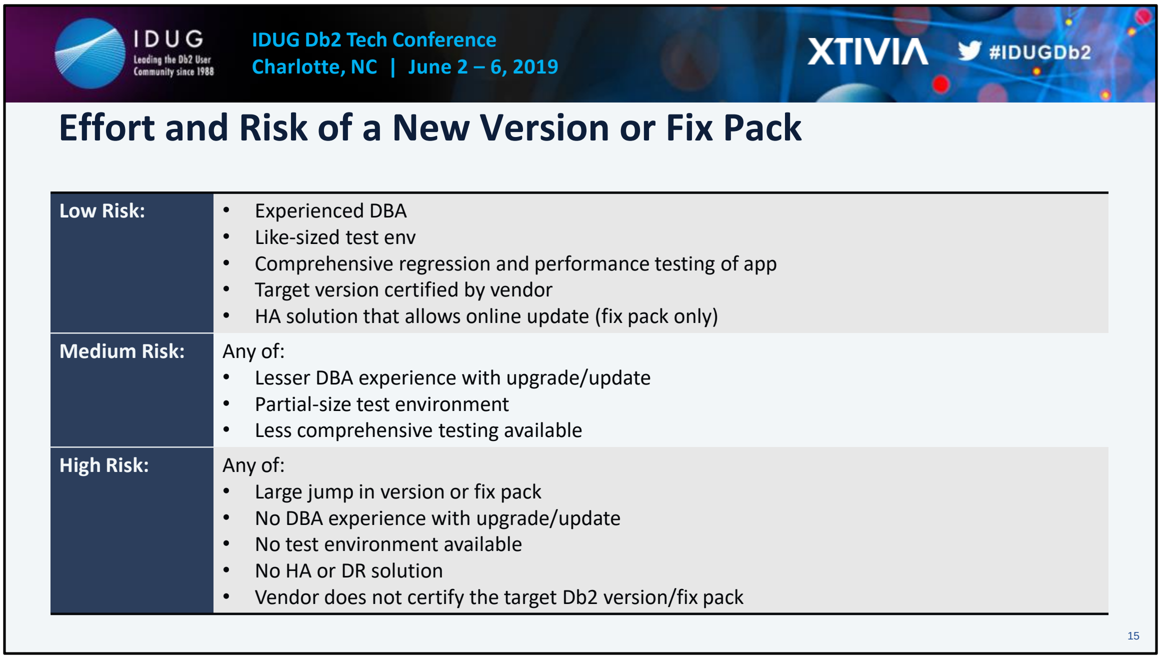
Technical Areas of Health Check



Db2 Version

- What versions are supported by the application vendor?
- How cutting-edge is the company up for?
- Recommend most recent stable version (at least one fix pack)
- Recommend most recent stable fix pack (at least one week old)
- Short paragraph should cover this area in a health check report

Low Priority:	ifix or fix pack is only one behind the most recent, and version is current
Medium Priority:	Version is one behind the most recent stable release. Fix pack is more than two behind the most current one
High Priority:	Version in use is past end of support or end of support is nearing. Base code is used or fix pack is more than 5 behind the most current



Effort and Risk of a New Version or Fix Pack

Low Risk:	• Experienced DBA • Like-sized test env • Comprehensive regression and performance testing of app • Target version certified by vendor • HA solution that allows online update (fix pack only)
Medium Risk:	Any of: • Lesser DBA experience with upgrade/update • Partial-size test environment • Less comprehensive testing available
High Risk:	Any of: • Large jump in version or fix pack • No DBA experience with upgrade/update • No test environment available • No HA or DR solution • Vendor does not certify the target Db2 version/fix pack

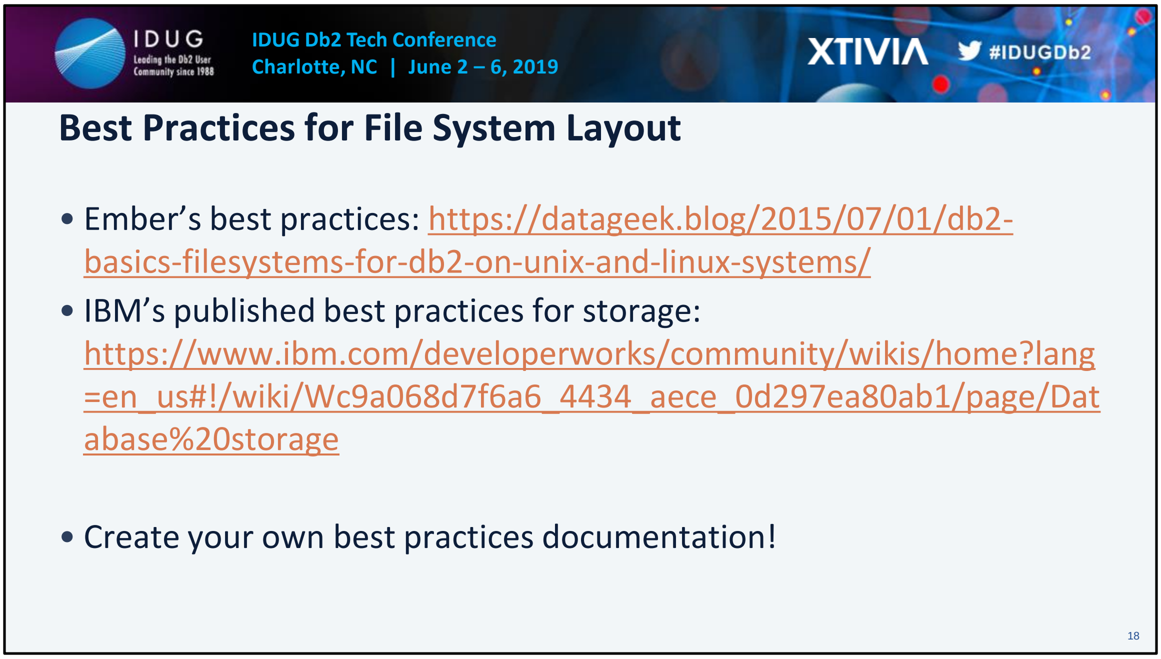
Disk and Tablespace

Technical Areas of Health Check

Disk

- File system layout as compared to best practices
- Where is the fastest disk used?
- What high-speed disk options are available to the server?
- Is there only one fully separate I/O path or multiple ones?
- Look for I/O wait to see if it is a problem for this database

- Compare the disk layout to your standard and identify differences
- Compare the disk layout to best practices and identify differences



Best Practices for File System Layout

- Ember’s best practices: https://datageek.blog/2015/07/01/db2-basics-filesystems-for-db2-on-unix-and-linux-systems/
- IBM’s published best practices for storage: https://www.ibm.com/developerworks/community/wikis/home?lang=en_us#!/wiki/Wc9a068d7f6a6_4434_aece_0d297ea80ab1/page/Database%20storage

- Create your own best practices documentation!

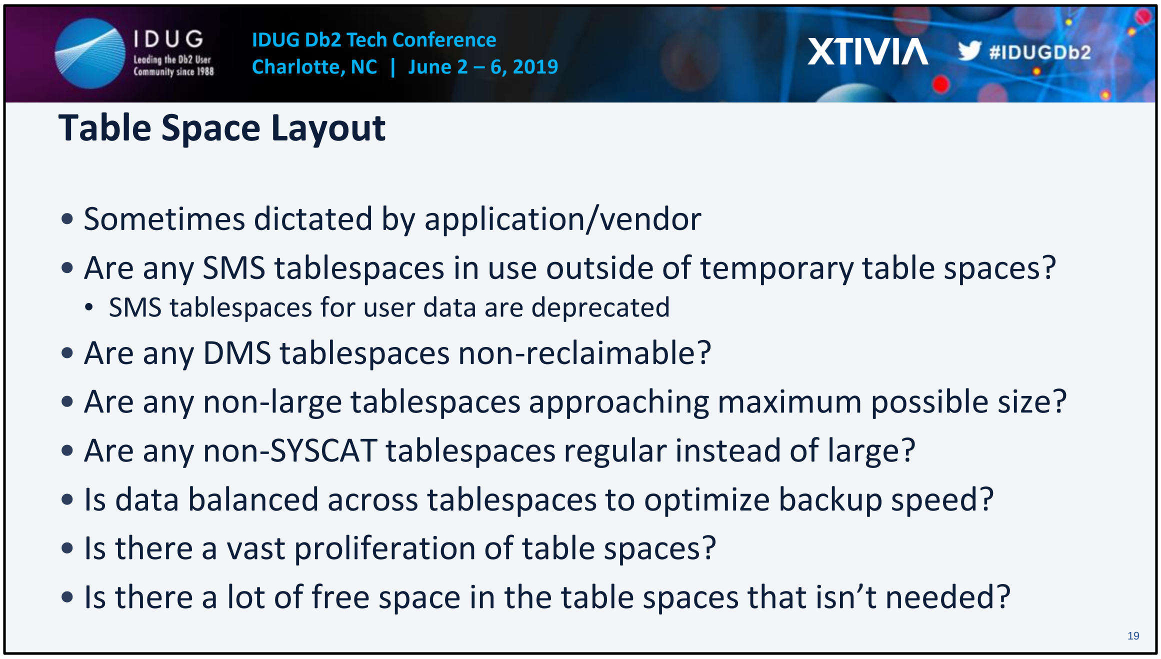
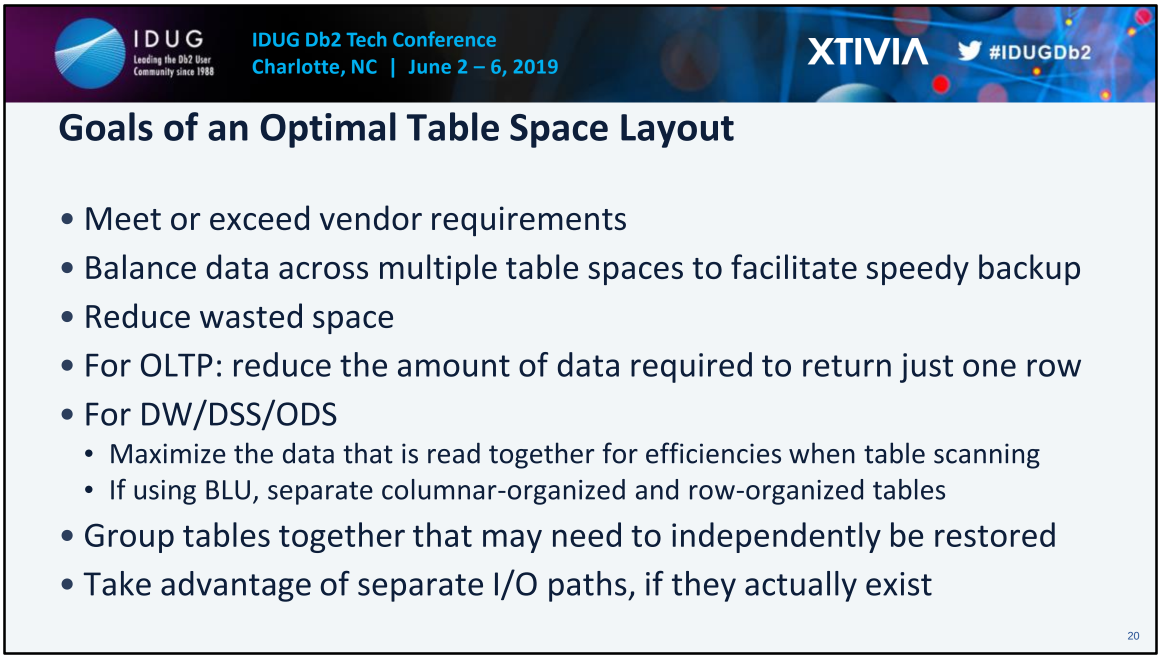


Table Space Layout

- Sometimes dictated by application/vendor
- Are any SMS tablespaces in use outside of temporary table spaces?
  - SMS tablespaces for user data are deprecated
- Are any DMS tablespaces non-reclaimable?
- Are any non-large tablespaces approaching maximum possible size?
- Are any non-SYSCAT tablespaces regular instead of large?
- Is data balanced across tablespaces to optimize backup speed?
- Is there a vast proliferation of table spaces?
- Is there a lot of free space in the table spaces that isn’t needed?



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

Goals of an Optimal Table Space Layout

- Meet or exceed vendor requirements
- Balance data across multiple table spaces to facilitate speedy backup
- Reduce wasted space
- For OLTP: reduce the amount of data required to return just one row
- For DW/DSS/ODS
  - Maximize the data that is read together for efficiencies when table scanning
  - If using BLU, separate columnar-organized and row-organized tables
- Group tables together that may need to independently be restored
- Take advantage of separate I/O paths, if they actually exist

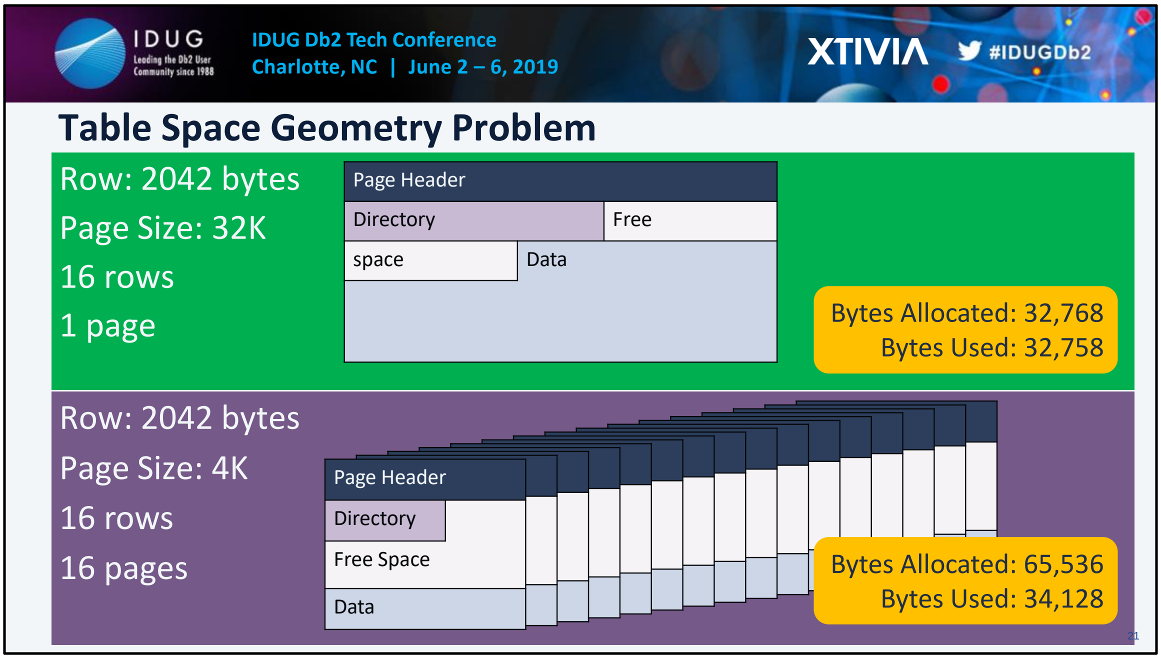


Table Space Geometry Problem

Row: 2042 bytes
Page Size: 32K
16 rows
1 page

Bytes Allocated: 32,768
Bytes Used: 32,758

Row: 2042 bytes
Page Size: 4K
16 rows
16 pages

Bytes Allocated: 65,536
Bytes Used: 34,128

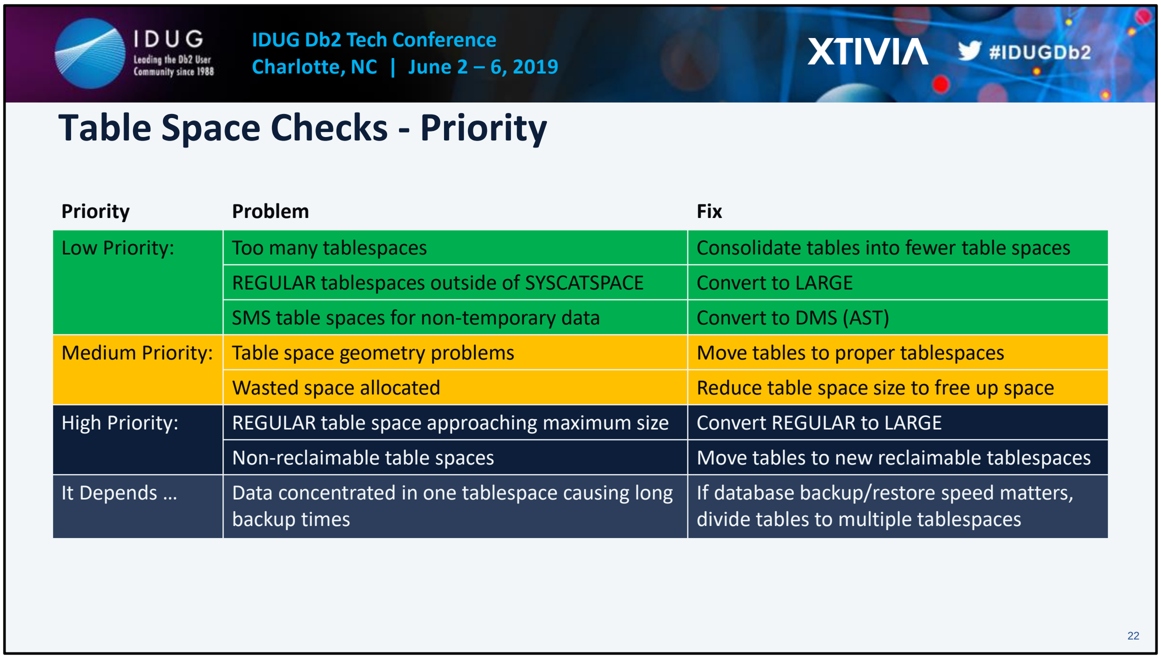
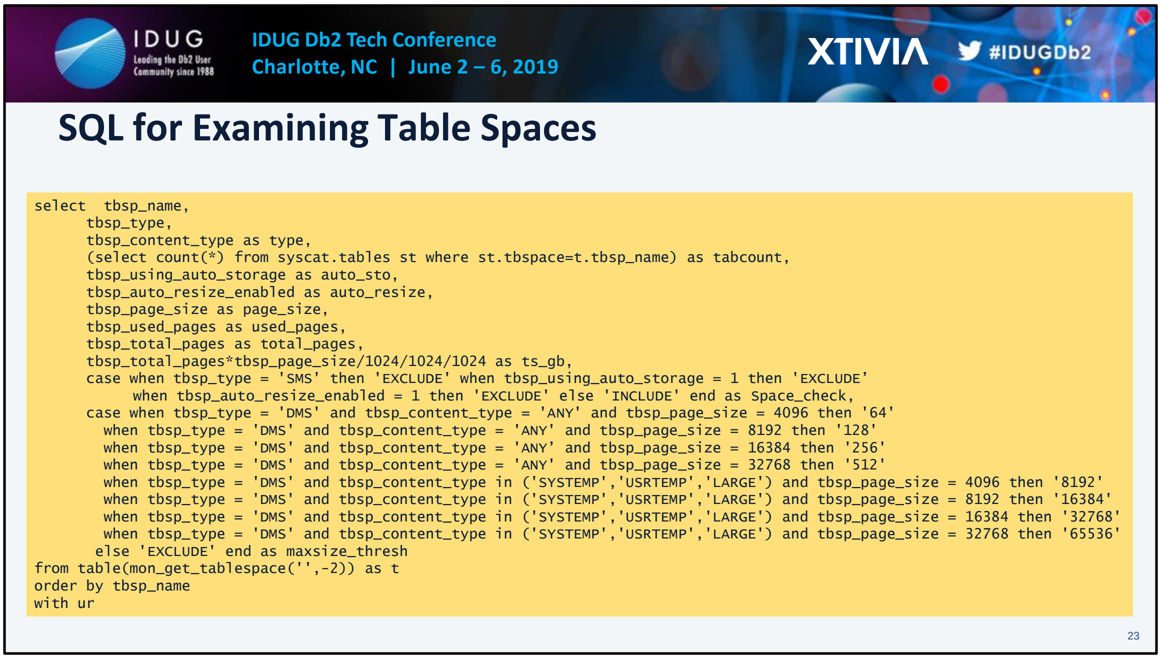


Table Space Checks - Priority

Priority	Problem	Fix
Low Priority:	Too many tablespaces	Consolidate tables into fewer table spaces
	REGULAR tablespaces outside of SYSCATSPACE	Convert to LARGE
	SMS table spaces for non-temporary data	Convert to DMS (AST)
Medium Priority:	Table space geometry problems	Move tables to proper tablespaces
	Wasted space allocated	Reduce table space size to free up space
High Priority:	REGULAR table space approaching maximum size	Convert REGULAR to LARGE
	Non-reclaimable table spaces	Move tables to new reclaimable tablespaces
It Depends …	Data concentrated in one tablespace causing long backup times	If database backup/restore speed matters, divide tables to multiple tablespaces



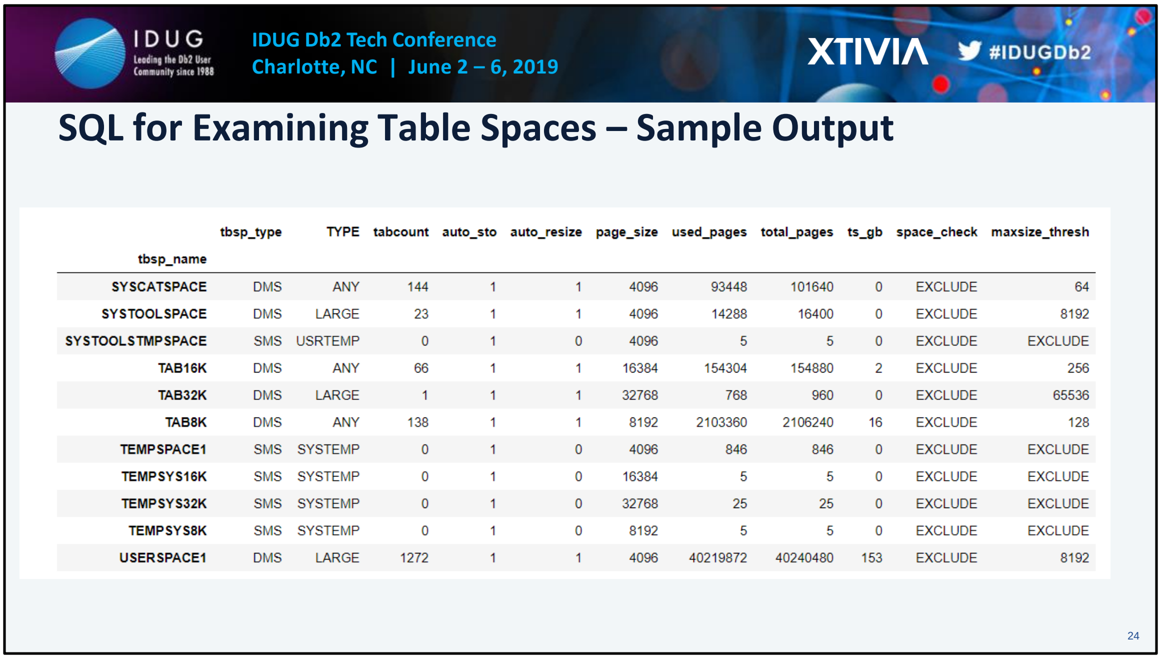
IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

SQL for Examining Table Spaces

```
select  tbsp_name,
      tbsp_type,
      tbsp_content_type as type,
      (select count(*) from syscat.tables st where st.tbspace=t.tbsp_name) as tabcount,
      tbsp_using_auto_storage as auto_sto,
      tbsp_auto_resize_enabled as auto_resize,
      tbsp_page_size as page_size,
      tbsp_used_pages as used_pages,
      tbsp_total_pages as total_pages,
      tbsp_total_pages*tbsp_page_size/1024/1024/1024 as ts_gb,
      case when tbsp_type = 'SMS' then 'EXCLUDE' when tbsp_using_auto_storage = 1 then 'EXCLUDE'
           when tbsp_auto_resize_enabled = 1 then 'EXCLUDE' else 'INCLUDE' end as Space_check,
      case when tbsp_type = 'DMS' and tbsp_content_type = 'ANY' and tbsp_page_size = 4096 then '64'
        when tbsp_type = 'DMS' and tbsp_content_type = 'ANY' and tbsp_page_size = 8192 then '128'
        when tbsp_type = 'DMS' and tbsp_content_type = 'ANY' and tbsp_page_size = 16384 then '256'
        when tbsp_type = 'DMS' and tbsp_content_type = 'ANY' and tbsp_page_size = 32768 then '512'
        when tbsp_type = 'DMS' and tbsp_content_type in ('SYSTEMP','USRTEMP','LARGE') and tbsp_page_size = 4096 then '8192'
        when tbsp_type = 'DMS' and tbsp_content_type in ('SYSTEMP','USRTEMP','LARGE') and tbsp_page_size = 8192 then '16384'
        when tbsp_type = 'DMS' and tbsp_content_type in ('SYSTEMP','USRTEMP','LARGE') and tbsp_page_size = 16384 then '32768'
        when tbsp_type = 'DMS' and tbsp_content_type in ('SYSTEMP','USRTEMP','LARGE') and tbsp_page_size = 32768 then '65536'
       else 'EXCLUDE' end as maxsize_thresh
from table(mon_get_tablespace('',-2)) as t
order by tbsp_name
with ur
```

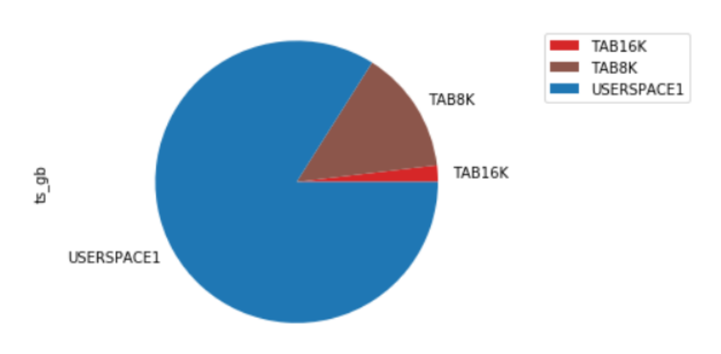
23



SQL for Examining Table Spaces – Sample Output

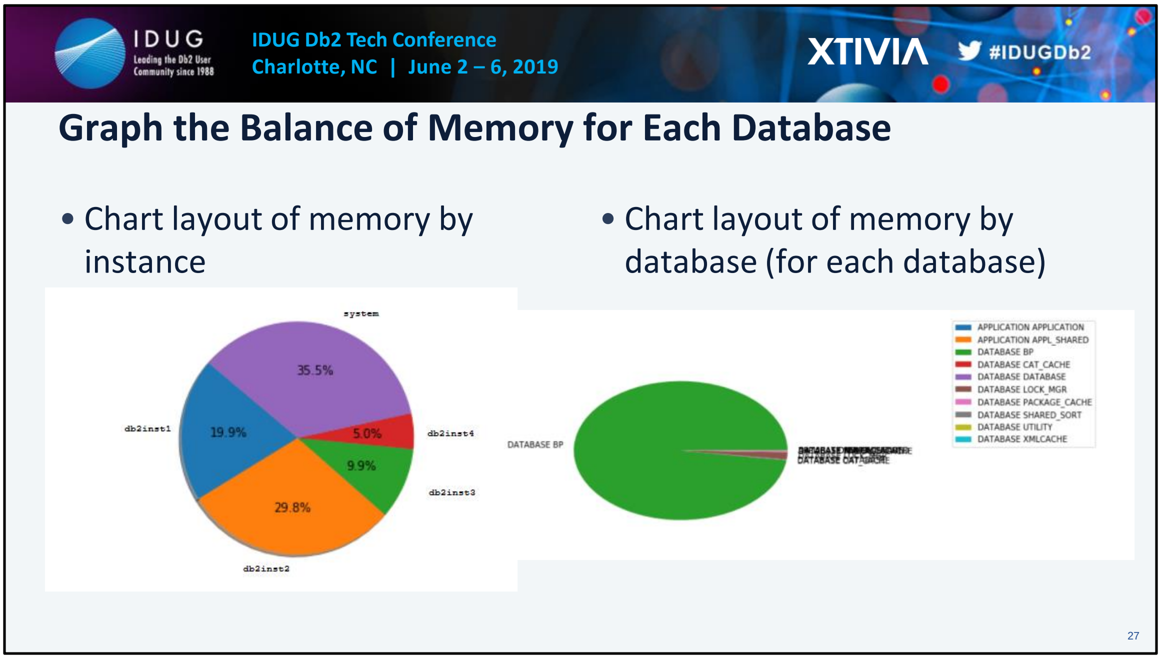
	tbsp_type	TYPE	tabcount	auto_sto	auto_resize	page_size	used_pages	total_pages	ts_gb	space_check	maxsize_thresh
tbsp_name											
SYSCATSPACE	DMS	ANY	144	1	1	4096	93448	101640	0	EXCLUDE	64
SYSTOOLSPACE	DMS	LARGE	23	1	1	4096	14288	16400	0	EXCLUDE	8192
SYSTOOLSTMPSPACE	SMS	USRTEMP	0	1	0	4096	5	5	0	EXCLUDE	EXCLUDE
TAB16K	DMS	ANY	66	1	1	16384	154304	154880	2	EXCLUDE	256
TAB32K	DMS	LARGE	1	1	1	32768	768	960	0	EXCLUDE	65536
TAB8K	DMS	ANY	138	1	1	8192	2103360	2106240	16	EXCLUDE	128
TEMPSPACE1	SMS	SYSTEMP	0	1	0	4096	846	846	0	EXCLUDE	EXCLUDE
TEMPSYS16K	SMS	SYSTEMP	0	1	0	16384	5	5	0	EXCLUDE	EXCLUDE
TEMPSYS32K	SMS	SYSTEMP	0	1	0	32768	25	25	0	EXCLUDE	EXCLUDE
TEMPSYS8K	SMS	SYSTEMP	0	1	0	8192	5	5	0	EXCLUDE	EXCLUDE
USERSPACE1	DMS	LARGE	1272	1	1	4096	40219872	40240480	153	EXCLUDE	8192

Graphical Representation Matters



Memory Layout and Balance

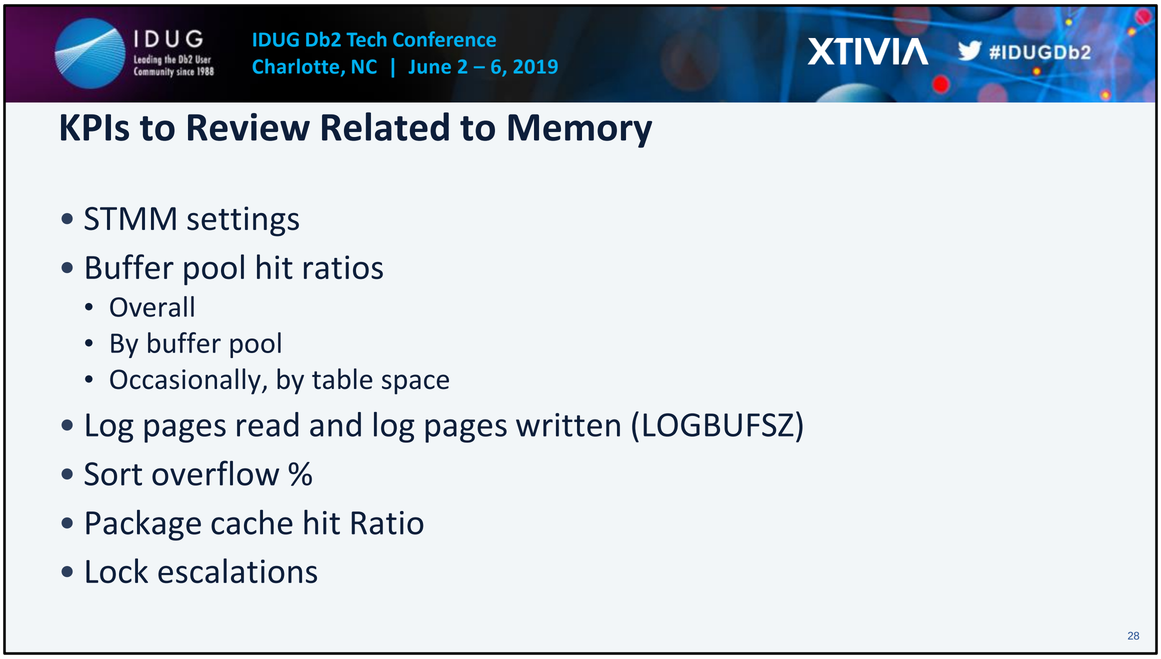
Technical Areas of Health Check



IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

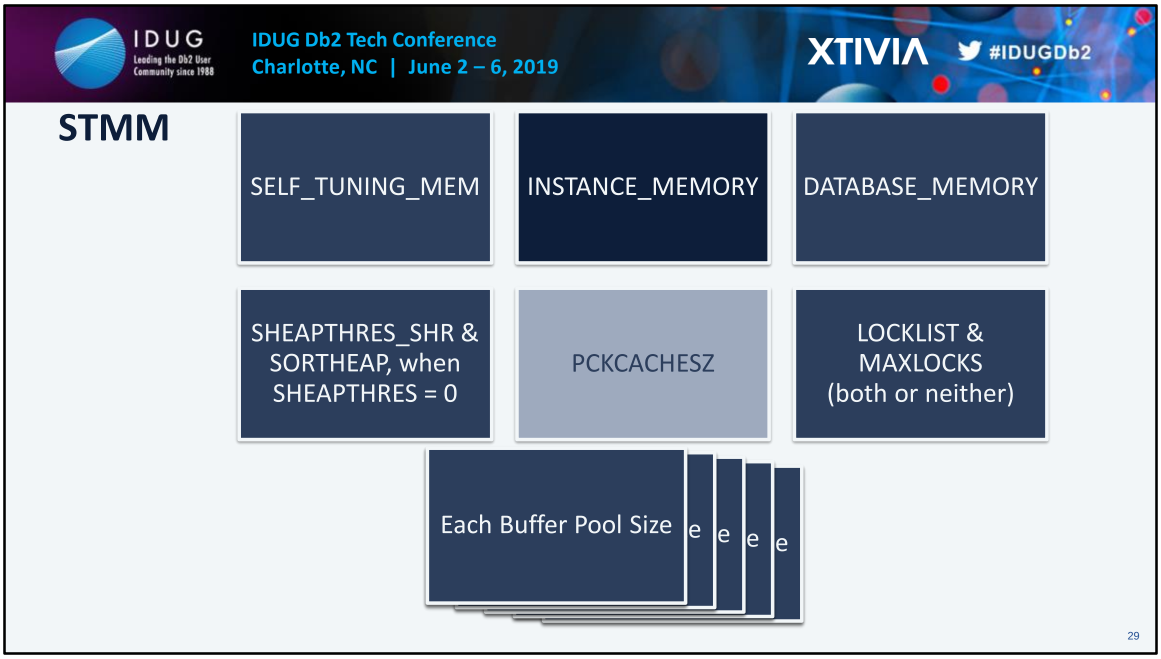
Graph the Balance of Memory for Each Database

- Chart layout of memory by instance
- Chart layout of memory by database (for each database)



KPIs to Review Related to Memory

- STMM settings
- Buffer pool hit ratios
  - Overall
  - By buffer pool
  - Occasionally, by table space
- Log pages read and log pages written (LOGBUFSZ)
- Sort overflow %
- Package cache hit Ratio
- Lock escalations



STMM

SELF_TUNING_MEM

INSTANCE_MEMORY

DATABASE_MEMORY

SHEAPTHRES_SHR & SORTHEAP, when SHEAPTHRES = 0

PCKCACHESZ

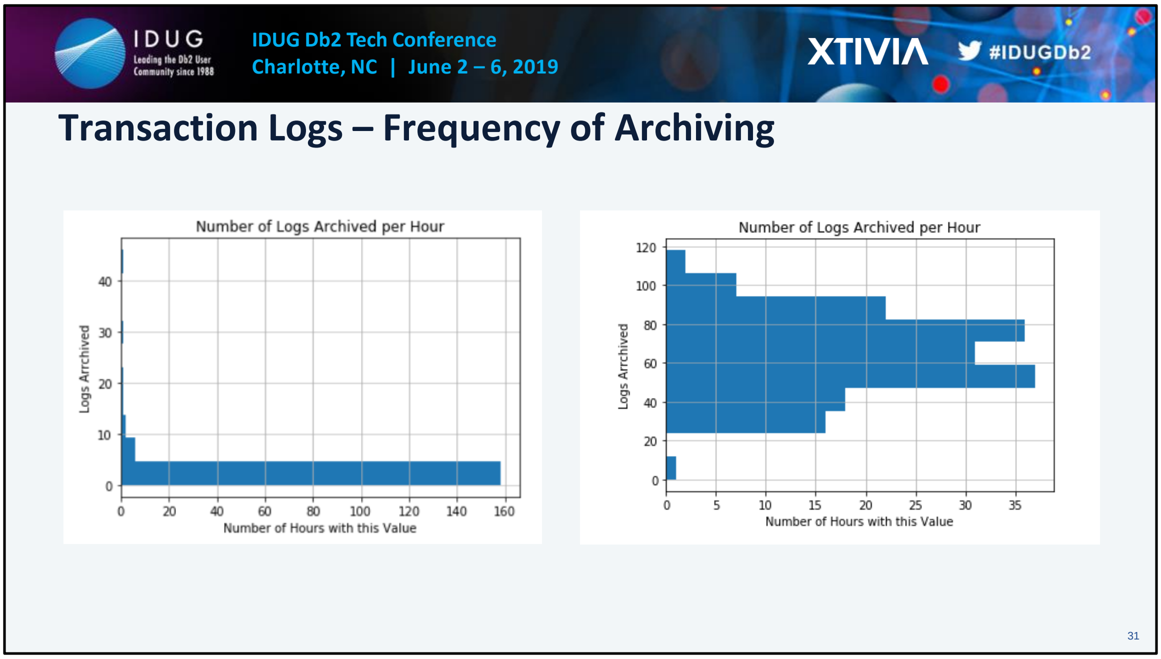
LOCKLIST & MAXLOCKS (both or neither)

Each Buffer Pool Size

e e e e

Transaction Logs

Technical Areas of Health Check



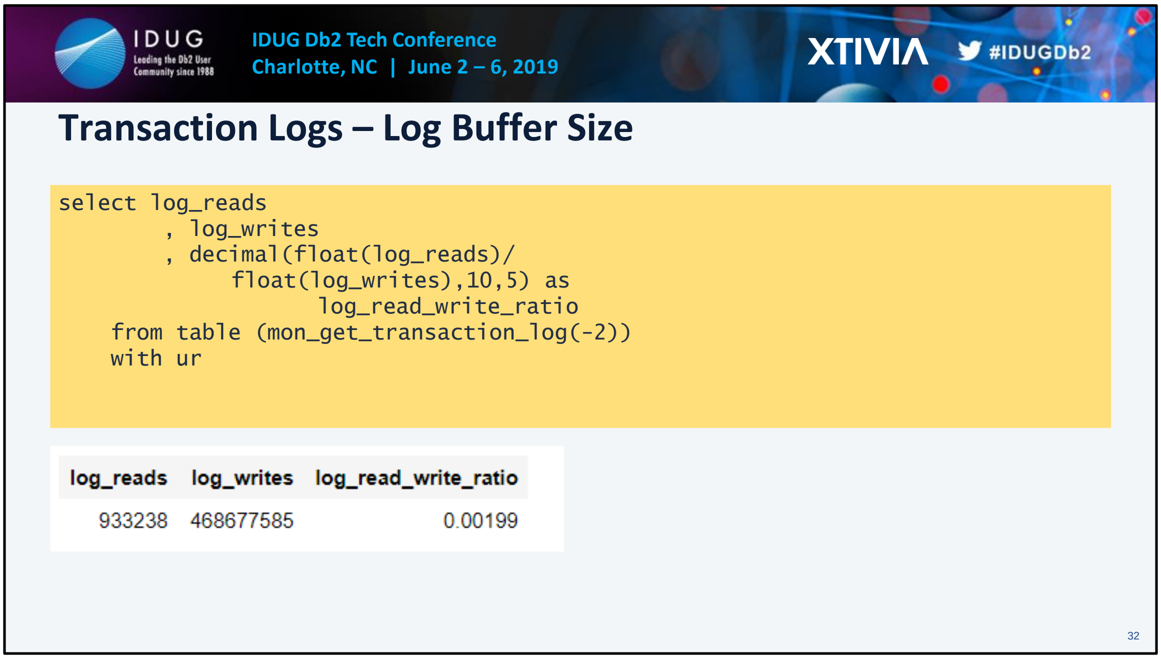
IDUG
Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

Transaction Logs – Frequency of Archiving

SQL for log file histogram: https://datageek.blog/2015/05/19/generating-log-archive-activity-histograms/



Transaction Logs – Log Buffer Size

```
select log_reads
        , log_writes
        , decimal(float(log_reads)/
             float(log_writes),10,5) as
                   log_read_write_ratio
    from table (mon_get_transaction_log(-2))
    with ur
```

log_reads	log_writes	log_read_write_ratio
933238	468677585	0.00199

SQL for log file histogram: https://datageek.blog/2015/05/19/generating-log-archive-activity-histograms/

Tables

Technical Areas of Health Check

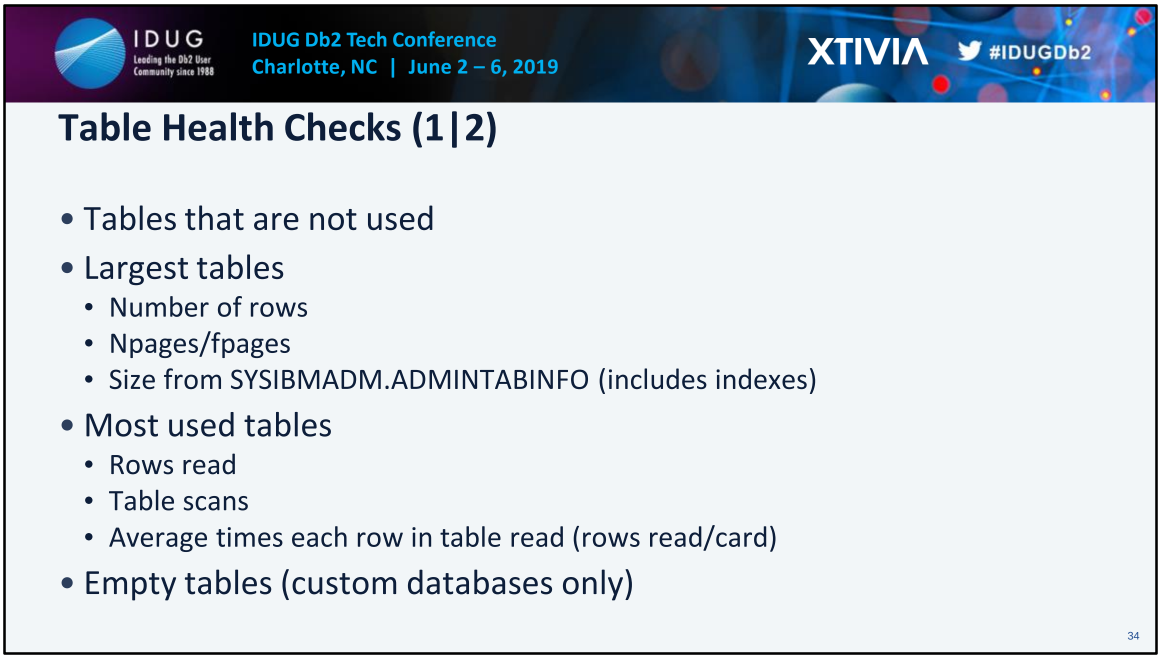


Table Health Checks (1|2)

- Tables that are not used
- Largest tables
  - Number of rows
  - Npages/fpages
  - Size from SYSIBMADM.ADMINTABINFO (includes indexes)
- Most used tables
  - Rows read
  - Table scans
  - Average times each row in table read (rows read/card)
- Empty tables (custom databases only)

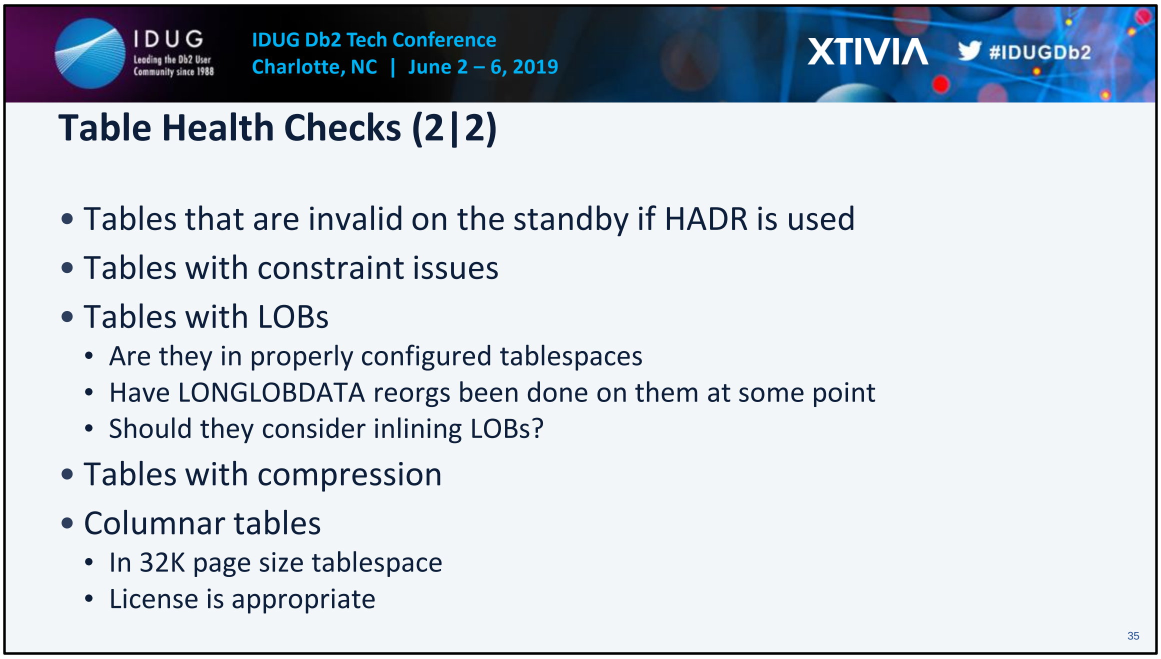
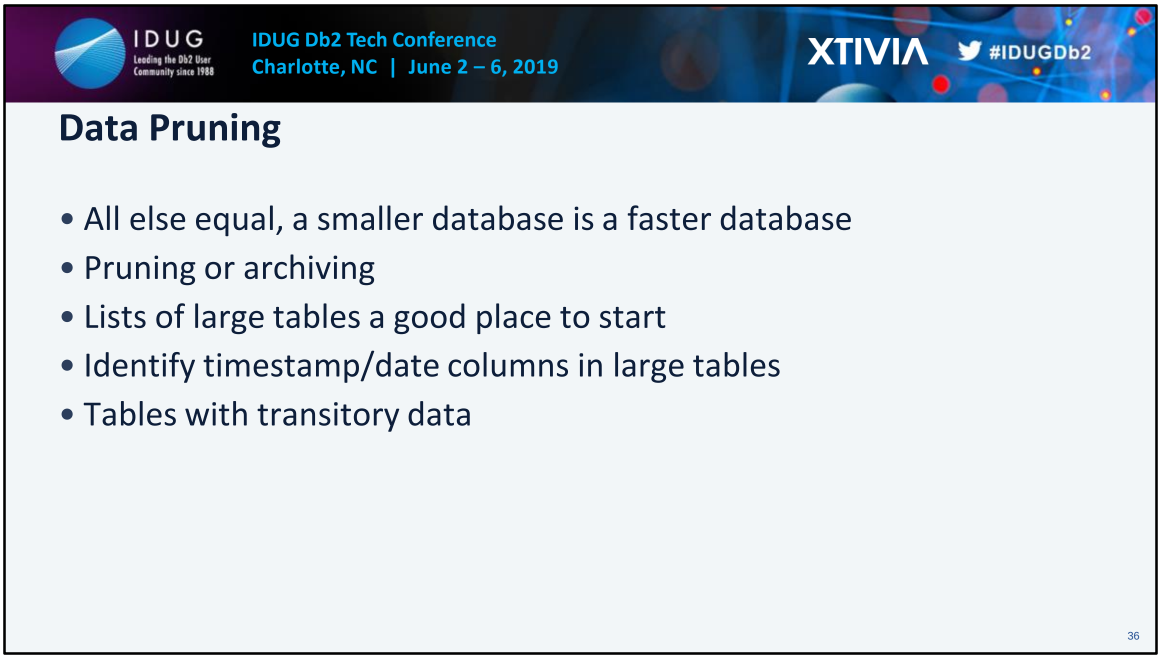


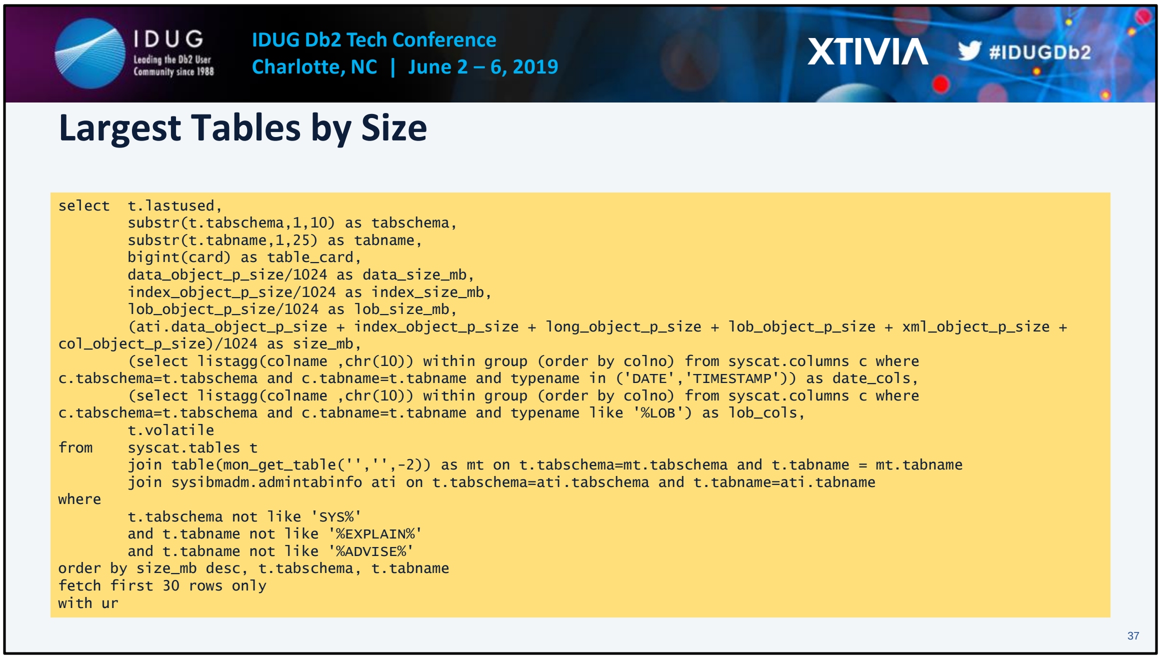
Table Health Checks (2|2)

- Tables that are invalid on the standby if HADR is used
- Tables with constraint issues
- Tables with LOBs
  - Are they in properly configured tablespaces
  - Have LONGLOBDATA reorgs been done on them at some point
  - Should they consider inlining LOBs?
- Tables with compression
- Columnar tables
  - In 32K page size tablespace
  - License is appropriate



Data Pruning

- All else equal, a smaller database is a faster database
- Pruning or archiving
- Lists of large tables a good place to start
- Identify timestamp/date columns in large tables
- Tables with transitory data



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

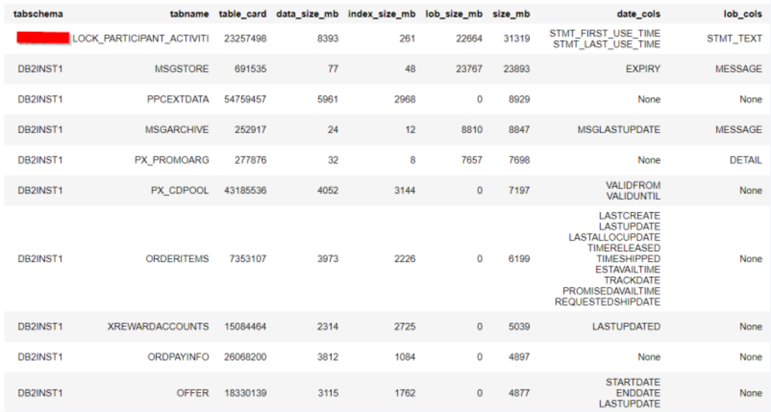
XTIVIA #IDUGDb2

Largest Tables by Size

```
select  t.lastused,
        substr(t.tabschema,1,10) as tabschema,
        substr(t.tabname,1,25) as tabname,
        bigint(card) as table_card,
        data_object_p_size/1024 as data_size_mb,
        index_object_p_size/1024 as index_size_mb,
        lob_object_p_size/1024 as lob_size_mb,
        (ati.data_object_p_size + index_object_p_size + long_object_p_size + lob_object_p_size + xml_object_p_size +
col_object_p_size)/1024 as size_mb,
        (select listagg(colname ,chr(10)) within group (order by colno) from syscat.columns c where
c.tabschema=t.tabschema and c.tabname=t.tabname and typename in ('DATE','TIMESTAMP')) as date_cols,
        (select listagg(colname ,chr(10)) within group (order by colno) from syscat.columns c where
c.tabschema=t.tabschema and c.tabname=t.tabname and typename like '%LOB') as lob_cols,
        t.volatile
from    syscat.tables t
        join table(mon_get_table('','',-2)) as mt on t.tabschema=mt.tabschema and t.tabname = mt.tabname
        join sysibmadm.admintabinfo ati on t.tabschema=ati.tabschema and t.tabname=ati.tabname
where
        t.tabschema not like 'SYS%'
        and t.tabname not like '%EXPLAIN%'
        and t.tabname not like '%ADVISE%'
order by size_mb desc, t.tabschema, t.tabname
fetch first 30 rows only
with ur
```

37

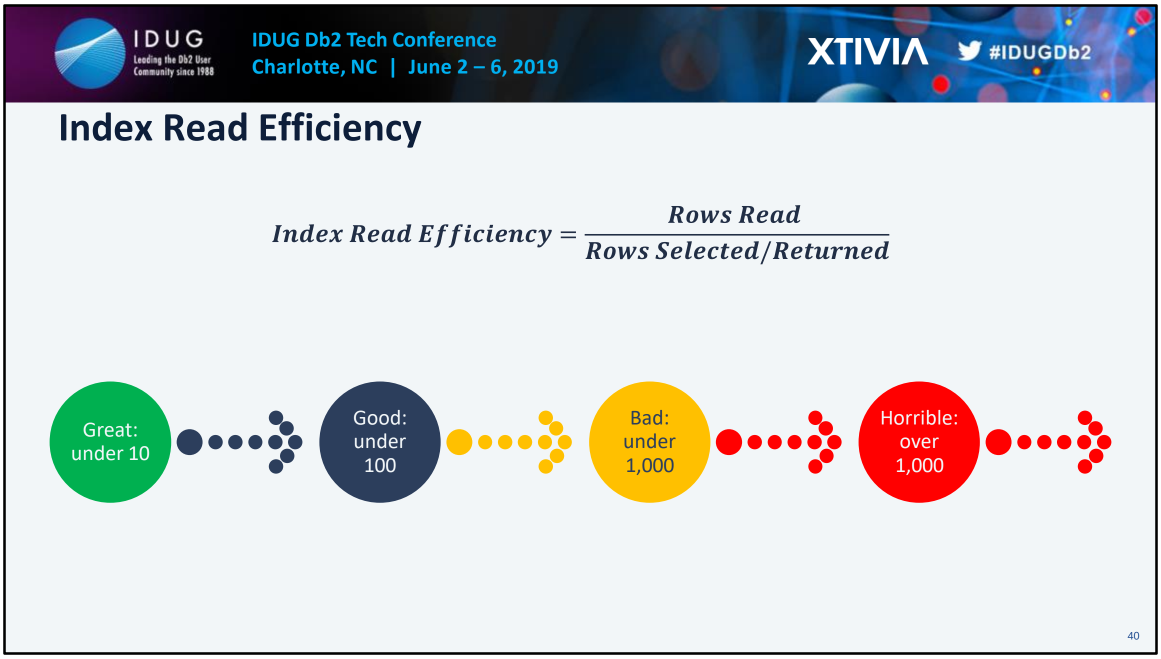
Largest Tables by Size – Partial Sample output



tabschema	tabname	table_card	data_size_mb	index_size_mb	lob_size_mb	size_mb	date_cols	lob_cols
[redacted]	LOCK_PARTICIPANT_ACTIVITI	23257498	8393	261	22664	31319	STMT_FIRST_USE_TIME STMT_LAST_USE_TIME	STMT_TEXT
DB2INST1	MSGSTORE	691535	77	48	23767	23893	EXPIRY	MESSAGE
DB2INST1	PPCEXTDATA	54759457	5961	2968	0	8929	None	None
DB2INST1	MSGARCHIVE	252917	24	12	8810	8847	MSGLASTUPDATE	MESSAGE
DB2INST1	PX_PROMOARG	277876	32	8	7657	7698	None	DETAIL
DB2INST1	PX_CDPOOL	43185536	4052	3144	0	7197	VALIDFROM VALIDUNTIL	None
DB2INST1	ORDERITEMS	7353107	3973	2226	0	6199	LASTCREATE LASTUPDATE LASTALLOCUPDATE TIMERELEASED TIMESHIPPED ESTAVAILTIME TRACKDATE PROMISEDAVAILTIME REQUESTEDSHIPDATE	None
DB2INST1	XREWARDACCOUNTS	15084464	2314	2725	0	5039	LASTUPDATED	None
DB2INST1	ORDPAYINFO	26068200	3812	1084	0	4897	None	None
DB2INST1	OFFER	18330139	3115	1762	0	4877	STARTDATE ENDDATE LASTUPDATE	None

Indexes

Technical Areas of Health Check



Index Read Efficiency

$$Index\ Read\ Efficiency = \frac{Rows\ Read}{Rows\ Selected/Returned}$$

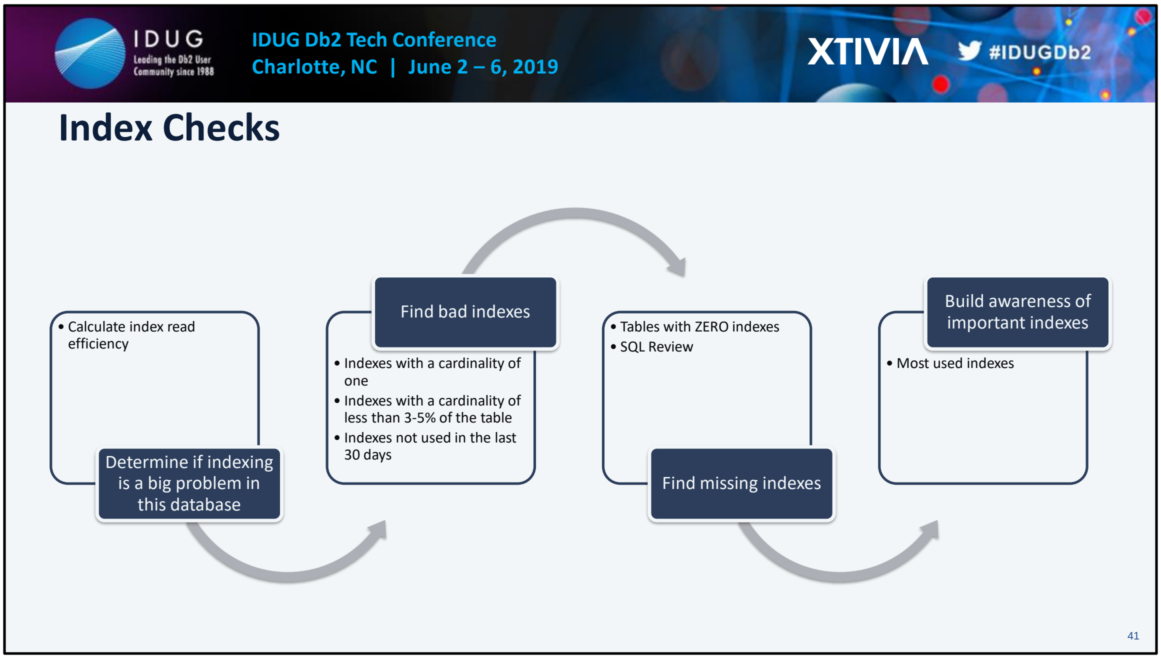
Great: under 10

Good: under 100

Bad: under 1,000

Horrible: over 1,000

For one row returned as a part of a query result set, how many rows did Db2 have to scan/consider?
Standards here are for OLTP databases.



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

Index Checks

Determine if indexing is a big problem in this database

- Calculate index read efficiency

Find bad indexes

- Indexes with a cardinality of one
- Indexes with a cardinality of less than 3-5% of the table
- Indexes not used in the last 30 days

Find missing indexes

- Tables with ZERO indexes
- SQL Review

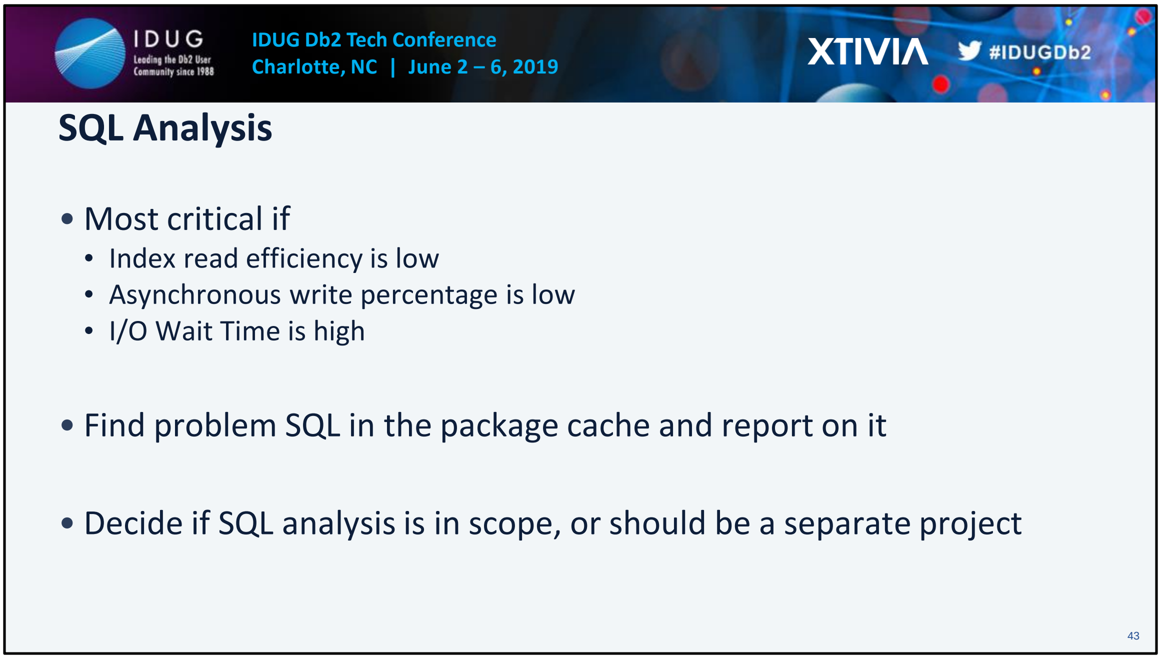
Build awareness of important indexes

- Most used indexes

41

SQL

Technical Areas of Health Check



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

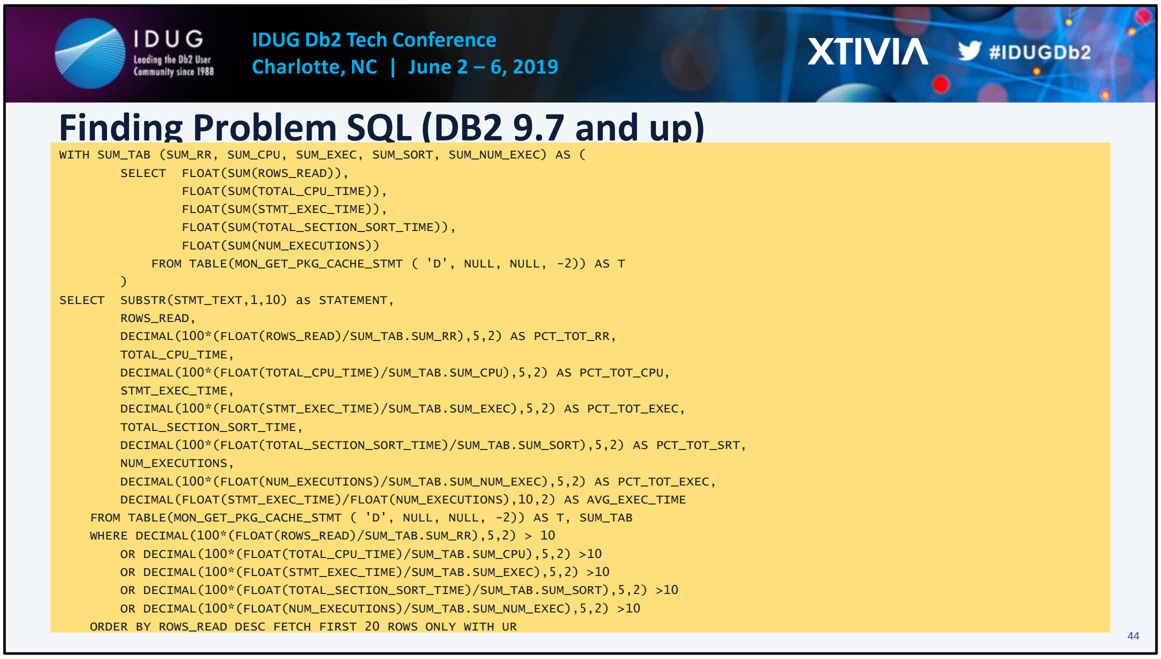
XTIVIA #IDUGDb2

SQL Analysis

- Most critical if
  - Index read efficiency is low
  - Asynchronous write percentage is low
  - I/O Wait Time is high
- Find problem SQL in the package cache and report on it
- Decide if SQL analysis is in scope, or should be a separate project

43

Finding problem SQL: https://datageek.blog/2015/02/12/db2-administrative-sql-cookbook-finding-problem-sql-in-the-package-cache/

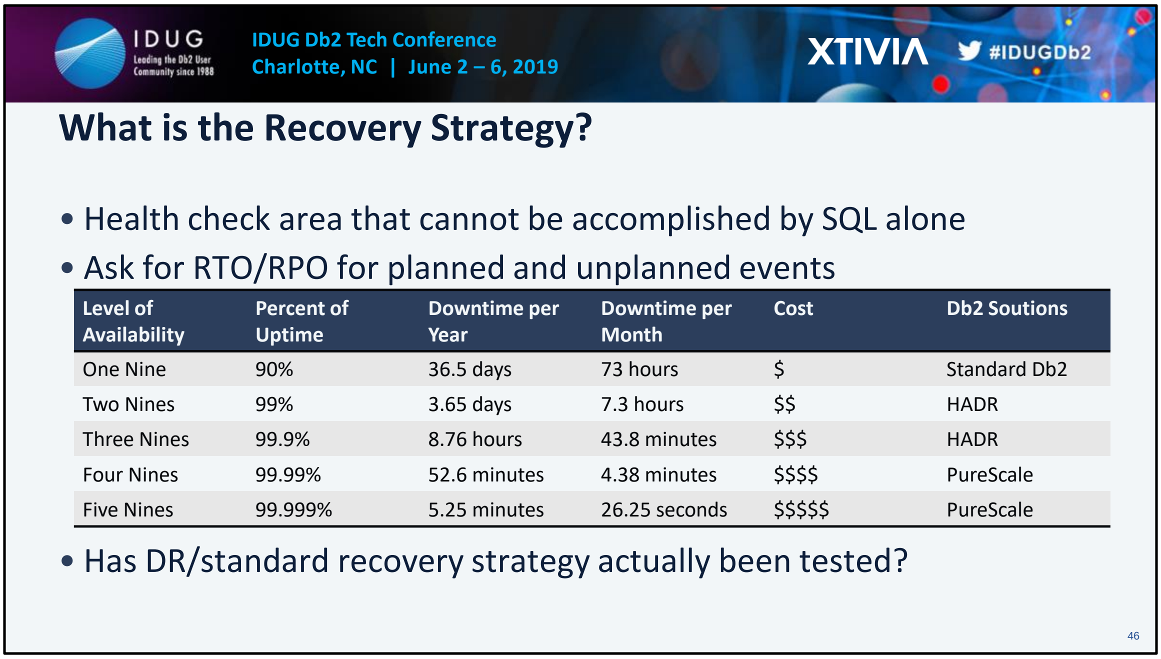


Finding Problem SQL (DB2 9.7 and up)

```
WITH SUM_TAB (SUM_RR, SUM_CPU, SUM_EXEC, SUM_SORT, SUM_NUM_EXEC) AS (
        SELECT  FLOAT(SUM(ROWS_READ)),
                FLOAT(SUM(TOTAL_CPU_TIME)),
                FLOAT(SUM(STMT_EXEC_TIME)),
                FLOAT(SUM(TOTAL_SECTION_SORT_TIME)),
                FLOAT(SUM(NUM_EXECUTIONS))
            FROM TABLE(MON_GET_PKG_CACHE_STMT ( 'D', NULL, NULL, -2)) AS T
        )
SELECT  SUBSTR(STMT_TEXT,1,10) as STATEMENT,
        ROWS_READ,
        DECIMAL(100*(FLOAT(ROWS_READ)/SUM_TAB.SUM_RR),5,2) AS PCT_TOT_RR,
        TOTAL_CPU_TIME,
        DECIMAL(100*(FLOAT(TOTAL_CPU_TIME)/SUM_TAB.SUM_CPU),5,2) AS PCT_TOT_CPU,
        STMT_EXEC_TIME,
        DECIMAL(100*(FLOAT(STMT_EXEC_TIME)/SUM_TAB.SUM_EXEC),5,2) AS PCT_TOT_EXEC,
        TOTAL_SECTION_SORT_TIME,
        DECIMAL(100*(FLOAT(TOTAL_SECTION_SORT_TIME)/SUM_TAB.SUM_SORT),5,2) AS PCT_TOT_SRT,
        NUM_EXECUTIONS,
        DECIMAL(100*(FLOAT(NUM_EXECUTIONS)/SUM_TAB.SUM_NUM_EXEC),5,2) AS PCT_TOT_EXEC,
        DECIMAL(FLOAT(STMT_EXEC_TIME)/FLOAT(NUM_EXECUTIONS),10,2) AS AVG_EXEC_TIME
    FROM TABLE(MON_GET_PKG_CACHE_STMT ( 'D', NULL, NULL, -2)) AS T, SUM_TAB
    WHERE DECIMAL(100*(FLOAT(ROWS_READ)/SUM_TAB.SUM_RR),5,2) > 10
        OR DECIMAL(100*(FLOAT(TOTAL_CPU_TIME)/SUM_TAB.SUM_CPU),5,2) >10
        OR DECIMAL(100*(FLOAT(STMT_EXEC_TIME)/SUM_TAB.SUM_EXEC),5,2) >10
        OR DECIMAL(100*(FLOAT(TOTAL_SECTION_SORT_TIME)/SUM_TAB.SUM_SORT),5,2) >10
        OR DECIMAL(100*(FLOAT(NUM_EXECUTIONS)/SUM_TAB.SUM_NUM_EXEC),5,2) >10
    ORDER BY ROWS_READ DESC FETCH FIRST 20 ROWS ONLY WITH UR
```

HA, DR, and Backup/Recovery

Technical Areas of Health Check

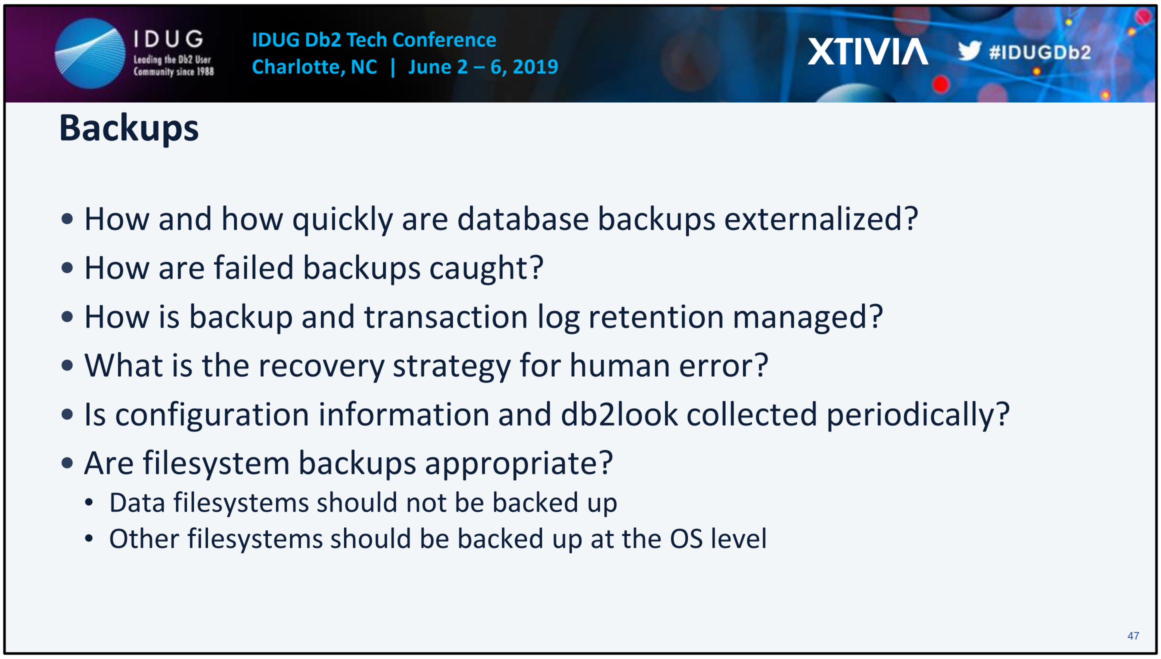


What is the Recovery Strategy?

- Health check area that cannot be accomplished by SQL alone
- Ask for RTO/RPO for planned and unplanned events

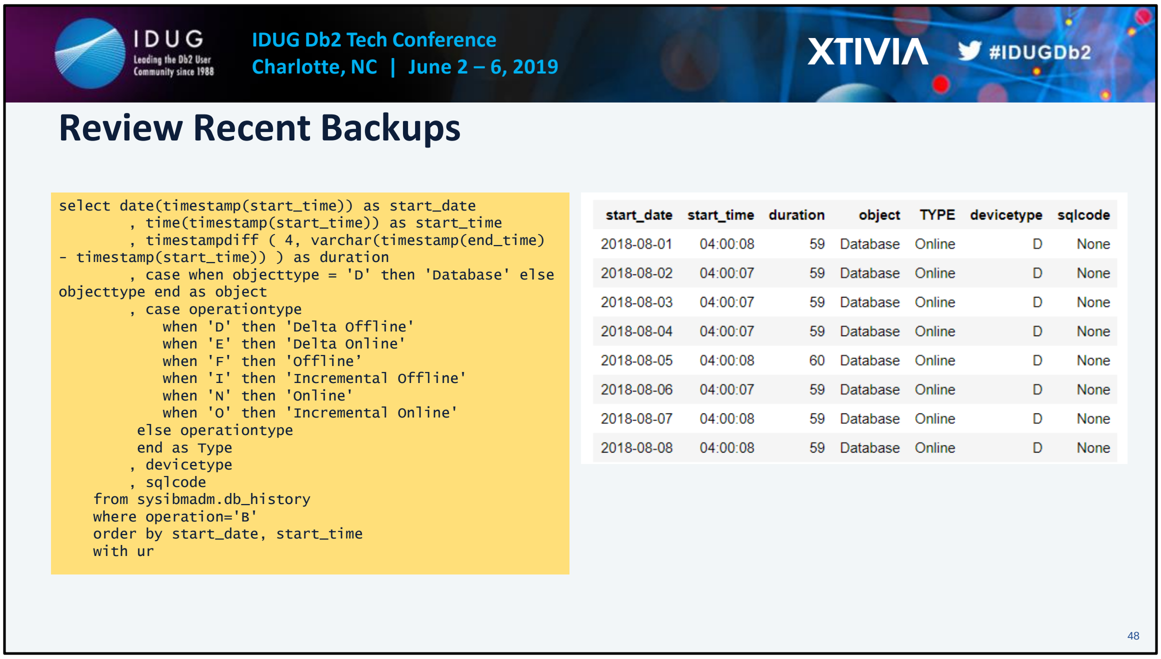
Level of Availability	Percent of Uptime	Downtime per Year	Downtime per Month	Cost	Db2 Soutions
One Nine	90%	36.5 days	73 hours	$	Standard Db2
Two Nines	99%	3.65 days	7.3 hours	$$	HADR
Three Nines	99.9%	8.76 hours	43.8 minutes	$$$	HADR
Four Nines	99.99%	52.6 minutes	4.38 minutes	$$$$	PureScale
Five Nines	99.999%	5.25 minutes	26.25 seconds	$$$$$	PureScale

- Has DR/standard recovery strategy actually been tested?



Backups

- How and how quickly are database backups externalized?
- How are failed backups caught?
- How is backup and transaction log retention managed?
- What is the recovery strategy for human error?
- Is configuration information and db2look collected periodically?
- Are filesystem backups appropriate?
  - Data filesystems should not be backed up
  - Other filesystems should be backed up at the OS level

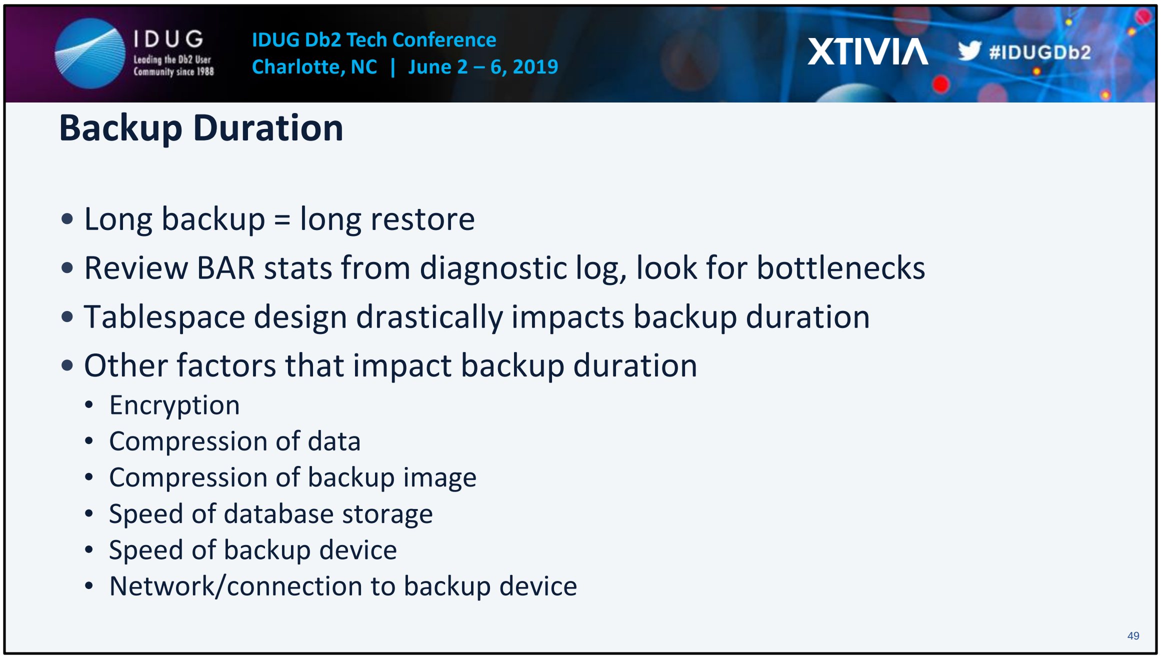


Review Recent Backups

```
select date(timestamp(start_time)) as start_date
        , time(timestamp(start_time)) as start_time
        , timestampdiff ( 4, varchar(timestamp(end_time)
- timestamp(start_time)) ) as duration
        , case when objecttype = 'D' then 'Database' else
objecttype end as object
        , case operationtype
            when 'D' then 'Delta Offline'
            when 'E' then 'Delta Online'
            when 'F' then 'Offline'
            when 'I' then 'Incremental Offline'
            when 'N' then 'Online'
            when 'O' then 'Incremental Online'
         else operationtype
         end as Type
        , devicetype
        , sqlcode
    from sysibmadm.db_history
    where operation='B'
    order by start_date, start_time
    with ur
```

start_date	start_time	duration	object	TYPE	devicetype	sqlcode
2018-08-01	04:00:08	59	Database	Online	D	None
2018-08-02	04:00:07	59	Database	Online	D	None
2018-08-03	04:00:07	59	Database	Online	D	None
2018-08-04	04:00:07	59	Database	Online	D	None
2018-08-05	04:00:08	60	Database	Online	D	None
2018-08-06	04:00:07	59	Database	Online	D	None
2018-08-07	04:00:08	59	Database	Online	D	None
2018-08-08	04:00:08	59	Database	Online	D	None

Sometimes backups are silently failing
Verify actual backup schedule meets stated backup schedule



IDUG
Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

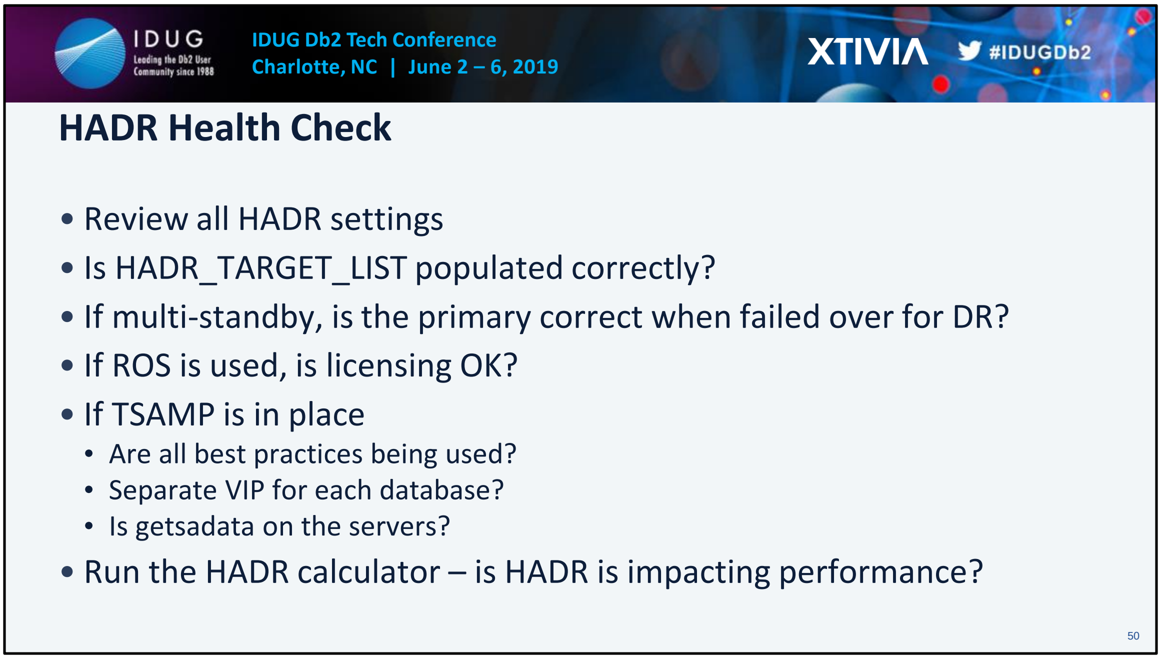
XTIVIA

#IDUGDb2

Backup Duration

- Long backup = long restore
- Review BAR stats from diagnostic log, look for bottlenecks
- Tablespace design drastically impacts backup duration
- Other factors that impact backup duration
  - Encryption
  - Compression of data
  - Compression of backup image
  - Speed of database storage
  - Speed of backup device
  - Network/connection to backup device

BAR stats: https://datageek.blog/2017/10/03/backup-performance-investigation/

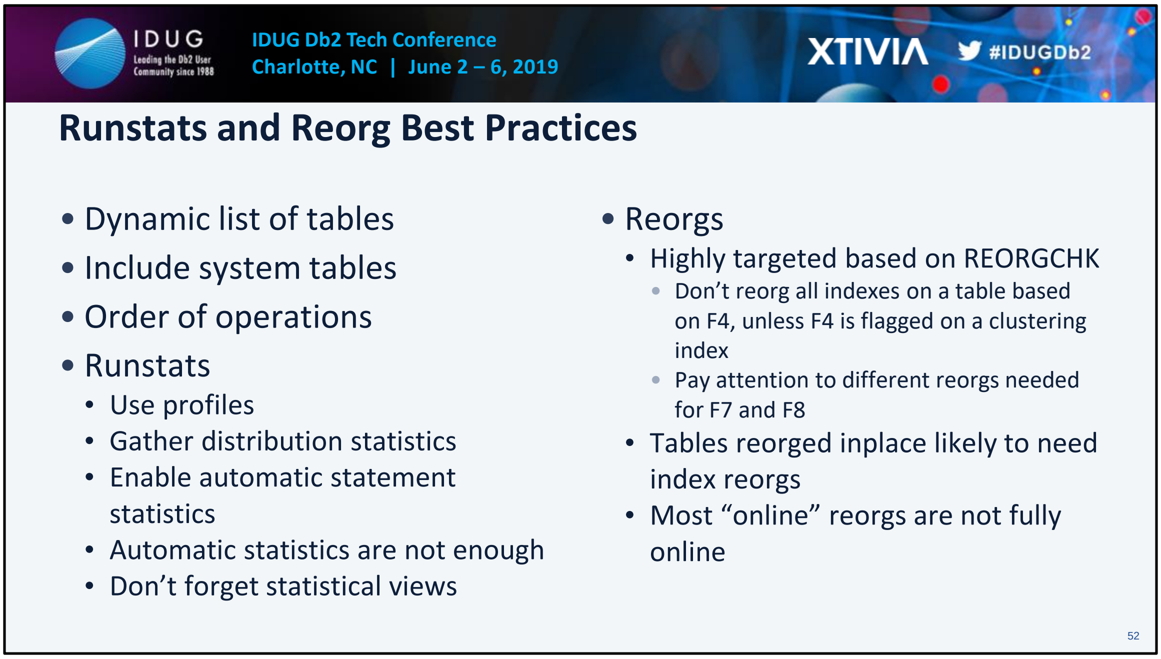


HADR Health Check

- Review all HADR settings
- Is HADR_TARGET_LIST populated correctly?
- If multi-standby, is the primary correct when failed over for DR?
- If ROS is used, is licensing OK?
- If TSAMP is in place
  - Are all best practices being used?
  - Separate VIP for each database?
  - Is getsadata on the servers?
- Run the HADR calculator – is HADR is impacting performance?

Maintenance – Runstats and Reorgs

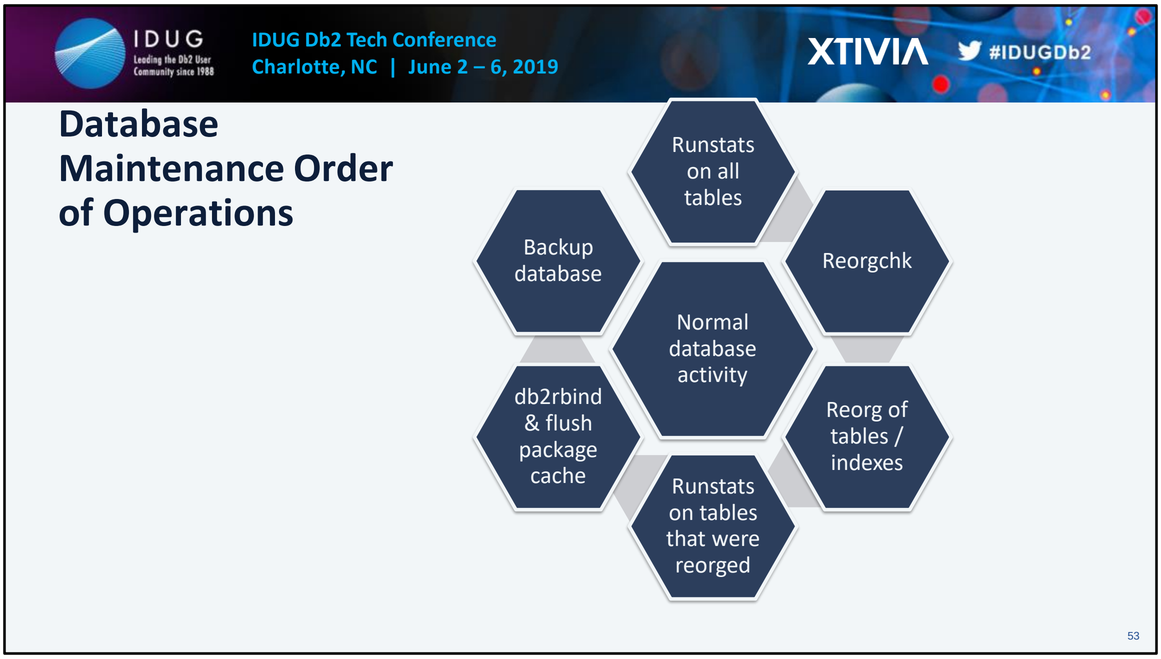
Technical Areas of Health Check



Runstats and Reorg Best Practices

- Dynamic list of tables
- Include system tables
- Order of operations
- Runstats
  - Use profiles
  - Gather distribution statistics
  - Enable automatic statement statistics
  - Automatic statistics are not enough
  - Don’t forget statistical views
- Reorgs
  - Highly targeted based on REORGCHK
    - Don’t reorg all indexes on a table based on F4, unless F4 is flagged on a clustering index
    - Pay attention to different reorgs needed for F7 and F8
  - Tables reorged inplace likely to need index reorgs
  - Most “online” reorgs are not fully online

https://datageek.blog/2016/09/27/embers-best-practices-for-runstats-and-reorgs/



IDUG Leading the Db2 User Community since 1988

IDUG Db2 Tech Conference
Charlotte, NC | June 2 – 6, 2019

XTIVIA #IDUGDb2

Database Maintenance Order of Operations

Runstats on all tables

Reorgchk

Reorg of tables / indexes

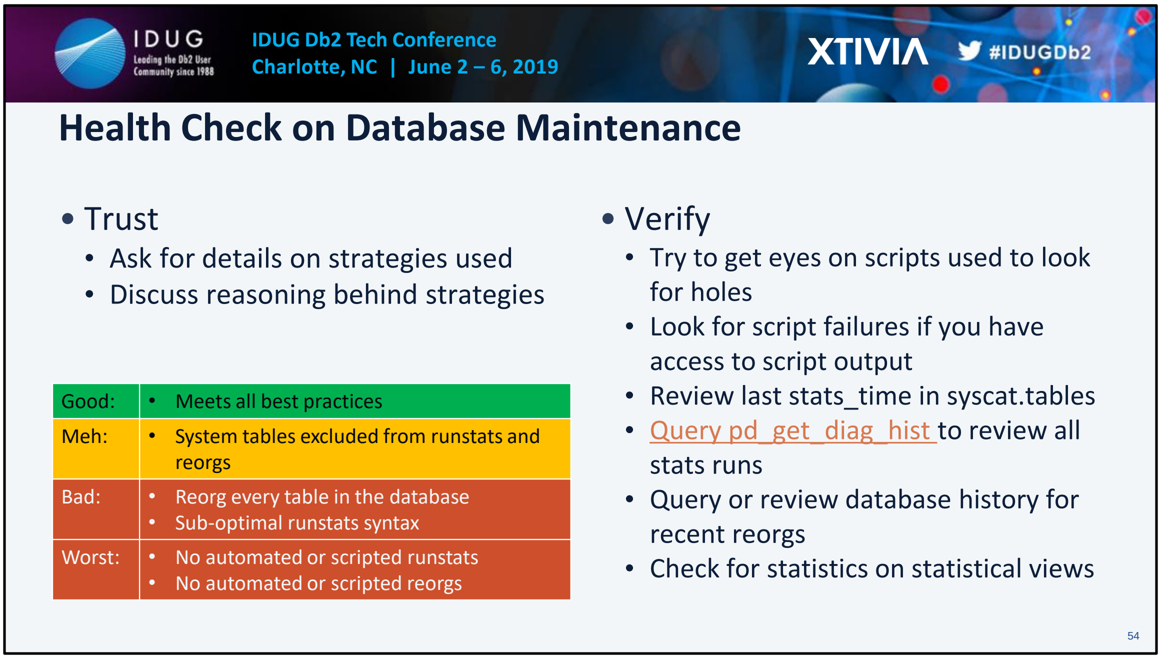
Runstats on tables that were reorged

db2rbind & flush package cache

Backup database

Normal database activity

53



Health Check on Database Maintenance

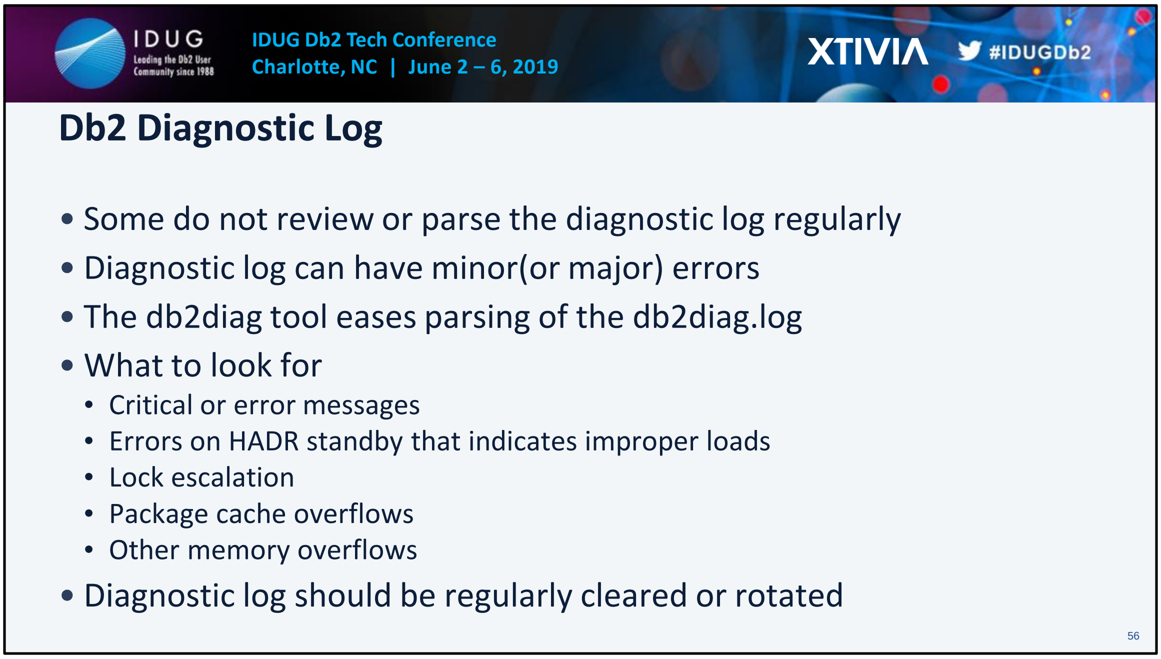
- Trust
  - Ask for details on strategies used
  - Discuss reasoning behind strategies

Good:	• Meets all best practices
Meh:	• System tables excluded from runstats and reorgs
Bad:	• Reorg every table in the database • Sub-optimal runstats syntax
Worst:	• No automated or scripted runstats • No automated or scripted reorgs

- Verify
  - Try to get eyes on scripts used to look for holes
  - Look for script failures if you have access to script output
  - Review last stats_time in syscat.tables
  - Query pd_get_diag_hist to review all stats runs
  - Query or review database history for recent reorgs
  - Check for statistics on statistical views

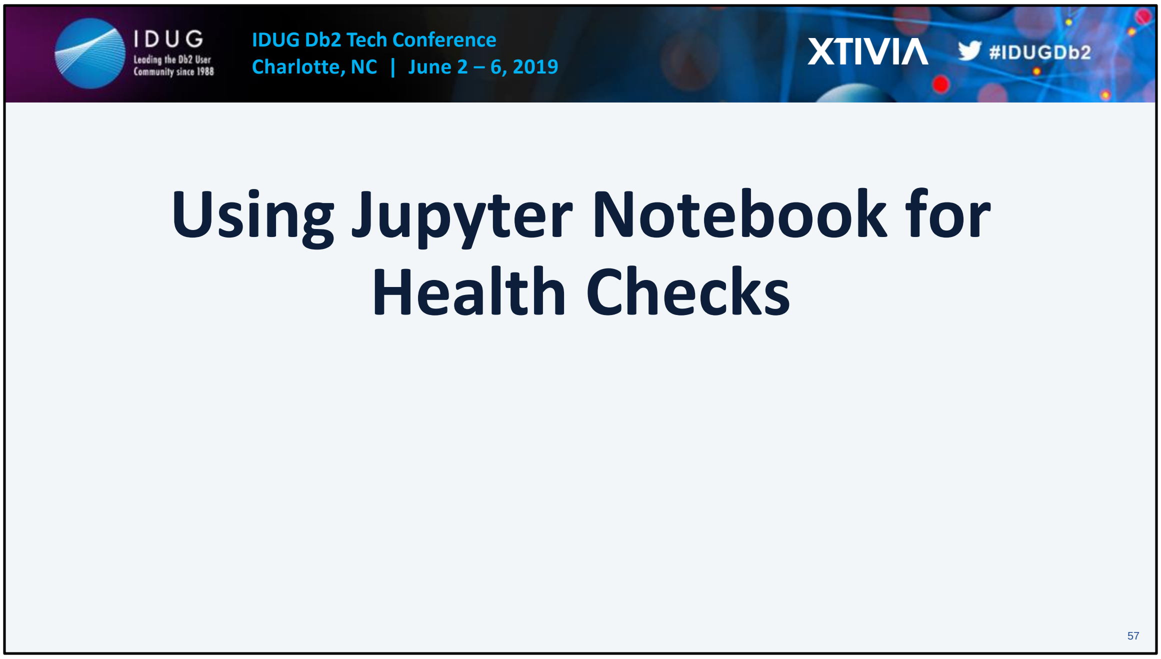
Db2 Diagnostic Log

Technical Areas of Health Check

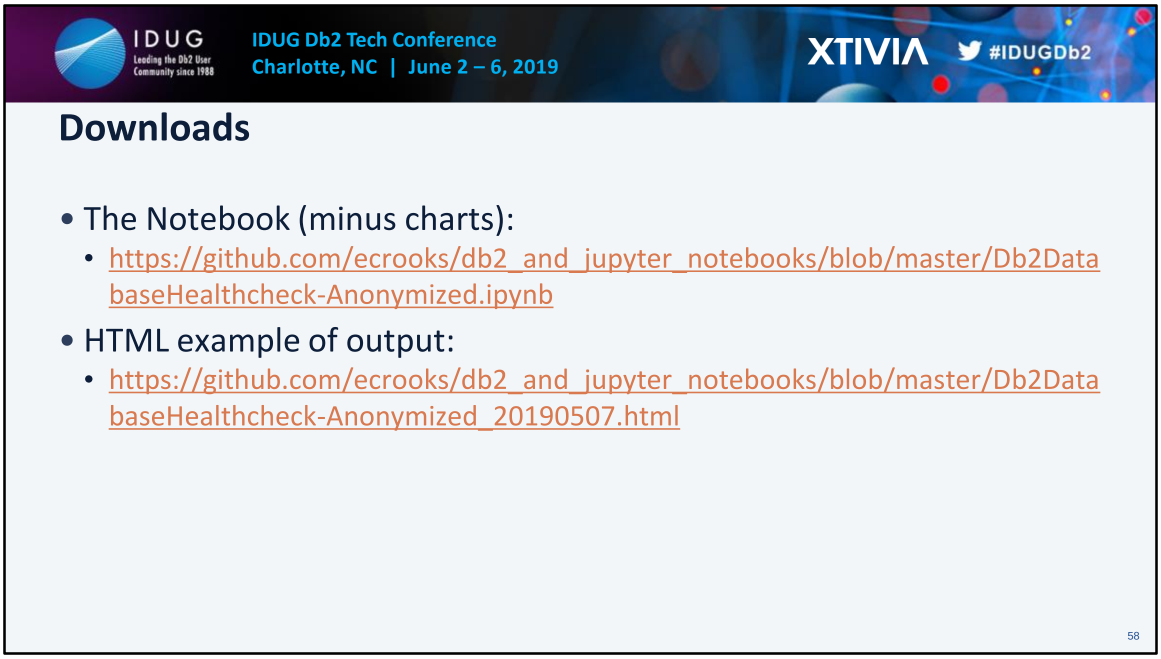


Db2 Diagnostic Log

- Some do not review or parse the diagnostic log regularly
- Diagnostic log can have minor(or major) errors
- The db2diag tool eases parsing of the db2diag.log
- What to look for
  - Critical or error messages
  - Errors on HADR standby that indicates improper loads
  - Lock escalation
  - Package cache overflows
  - Other memory overflows
- Diagnostic log should be regularly cleared or rotated



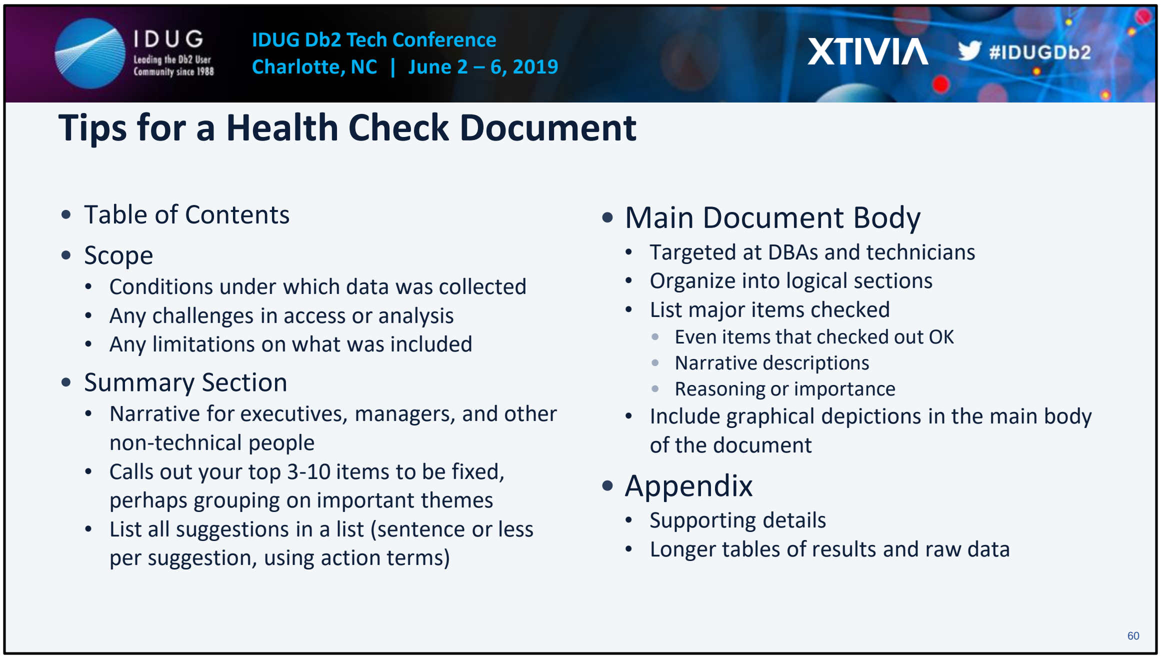
Using Jupyter Notebook for Health Checks



Downloads

- The Notebook (minus charts):
  - https://github.com/ecrooks/db2_and_jupyter_notebooks/blob/master/Db2DatabaseHealthcheck-Anonymized.ipynb
- HTML example of output:
  - https://github.com/ecrooks/db2_and_jupyter_notebooks/blob/master/Db2DatabaseHealthcheck-Anonymized_20190507.html

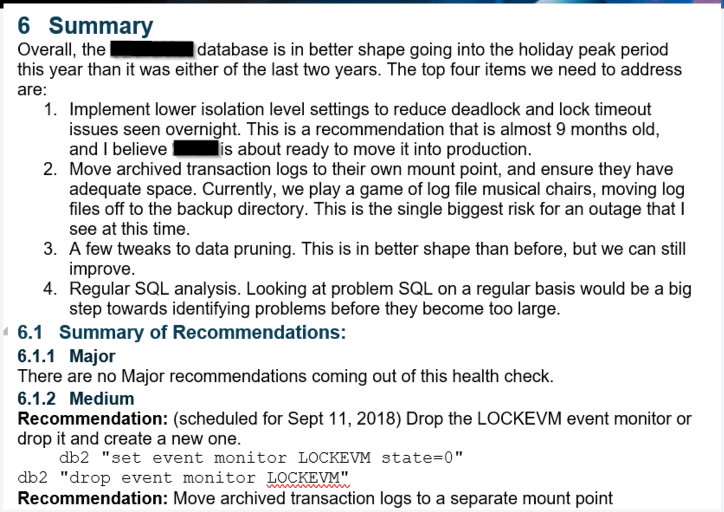
The Health Check Document



Tips for a Health Check Document

- Table of Contents
- Scope
  - Conditions under which data was collected
  - Any challenges in access or analysis
  - Any limitations on what was included
- Summary Section
  - Narrative for executives, managers, and other non-technical people
  - Calls out your top 3-10 items to be fixed, perhaps grouping on important themes
  - List all suggestions in a list (sentence or less per suggestion, using action terms)
- Main Document Body
  - Targeted at DBAs and technicians
  - Organize into logical sections
  - List major items checked
    - Even items that checked out OK
    - Narrative descriptions
    - Reasoning or importance
  - Include graphical depictions in the main body of the document
- Appendix
  - Supporting details
  - Longer tables of results and raw data

Sample Summary Section



6 Summary

Overall, the [redacted] database is in better shape going into the holiday peak period this year than it was either of the last two years. The top four items we need to address are:

1. Implement lower isolation level settings to reduce deadlock and lock timeout issues seen overnight. This is a recommendation that is almost 9 months old, and I believe [redacted] is about ready to move it into production.
2. Move archived transaction logs to their own mount point, and ensure they have adequate space. Currently, we play a game of log file musical chairs, moving log files off to the backup directory. This is the single biggest risk for an outage that I see at this time.
3. A few tweaks to data pruning. This is in better shape than before, but we can still improve.
4. Regular SQL analysis. Looking at problem SQL on a regular basis would be a big step towards identifying problems before they become too large.

6.1 Summary of Recommendations:

6.1.1 Major

There are no Major recommendations coming out of this health check.

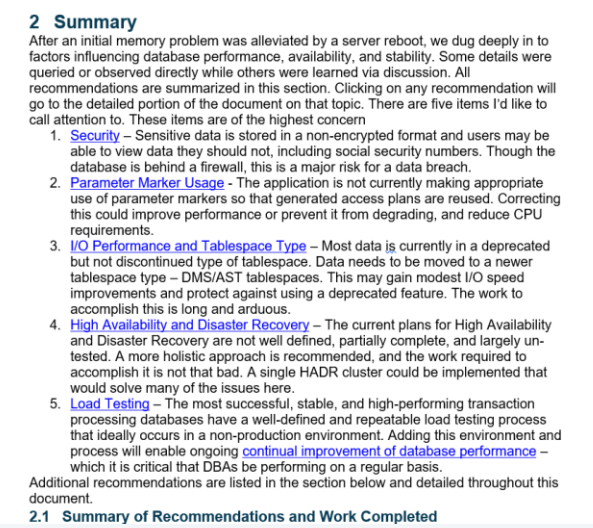
6.1.2 Medium

Recommendation: (scheduled for Sept 11, 2018) Drop the LOCKEVM event monitor or drop it and create a new one.

```
    db2 "set event monitor LOCKEVM state=0"
db2 "drop event monitor LOCKEVM"
```

Recommendation: Move archived transaction logs to a separate mount point

Sample Summary Section



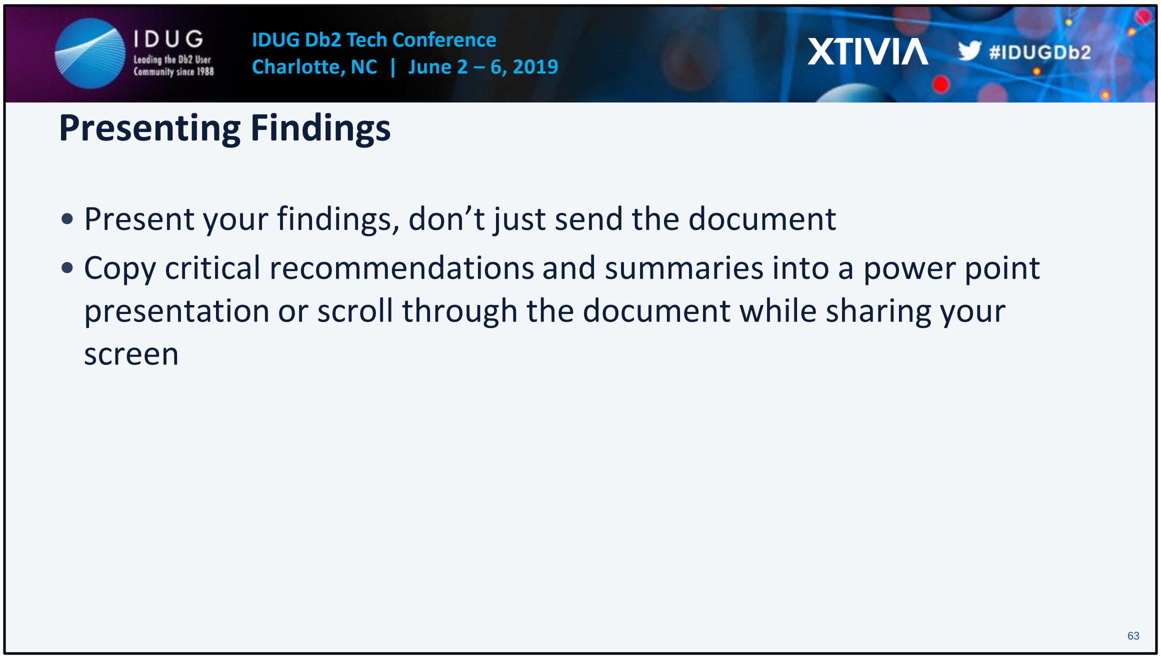
2 Summary

After an initial memory problem was alleviated by a server reboot, we dug deeply in to factors influencing database performance, availability, and stability. Some details were queried or observed directly while others were learned via discussion. All recommendations are summarized in this section. Clicking on any recommendation will go to the detailed portion of the document on that topic. There are five items I'd like to call attention to. These items are of the highest concern

1. Security – Sensitive data is stored in a non-encrypted format and users may be able to view data they should not, including social security numbers. Though the database is behind a firewall, this is a major risk for a data breach.
2. Parameter Marker Usage - The application is not currently making appropriate use of parameter markers so that generated access plans are reused. Correcting this could improve performance or prevent it from degrading, and reduce CPU requirements.
3. I/O Performance and Tablespace Type – Most data is currently in a deprecated but not discontinued type of tablespace. Data needs to be moved to a newer tablespace type – DMS/AST tablespaces. This may gain modest I/O speed improvements and protect against using a deprecated feature. The work to accomplish this is long and arduous.
4. High Availability and Disaster Recovery – The current plans for High Availability and Disaster Recovery are not well defined, partially complete, and largely un-tested. A more holistic approach is recommended, and the work required to accomplish it is not that bad. A single HADR cluster could be implemented that would solve many of the issues here.
5. Load Testing – The most successful, stable, and high-performing transaction processing databases have a well-defined and repeatable load testing process that ideally occurs in a non-production environment. Adding this environment and process will enable ongoing continual improvement of database performance – which it is critical that DBAs be performing on a regular basis.

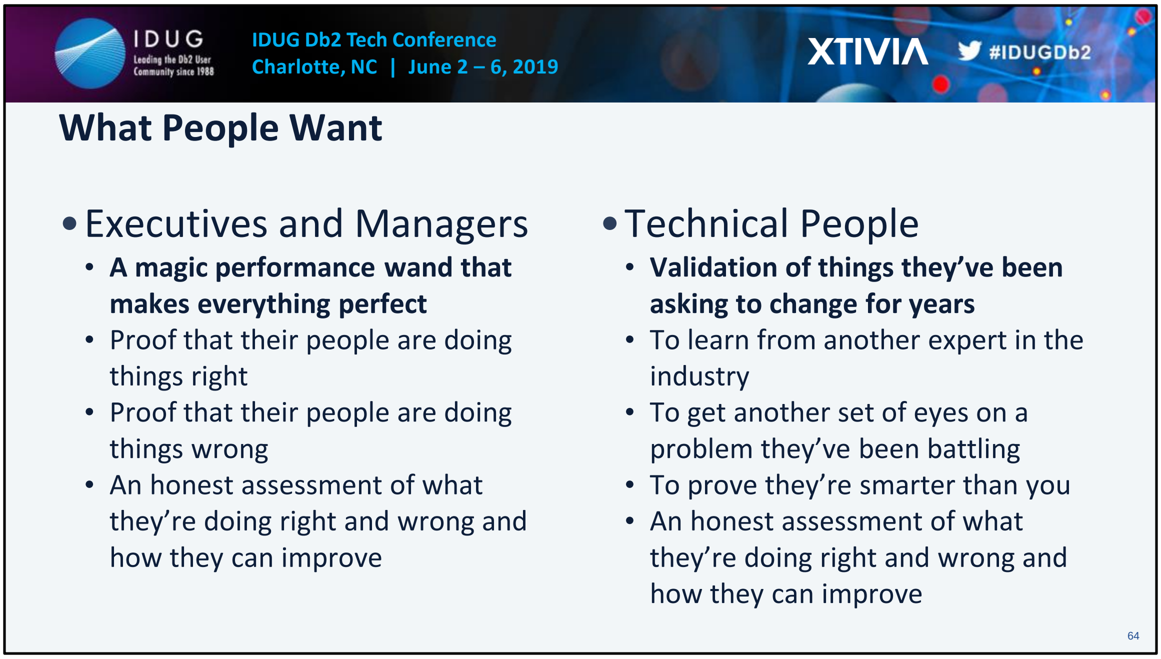
Additional recommendations are listed in the section below and detailed throughout this document.

2.1 Summary of Recommendations and Work Completed



Presenting Findings

- Present your findings, don’t just send the document
- Copy critical recommendations and summaries into a power point presentation or scroll through the document while sharing your screen



What People Want

- Executives and Managers
  - **A magic performance wand that makes everything perfect**
  - Proof that their people are doing things right
  - Proof that their people are doing things wrong
  - An honest assessment of what they're doing right and wrong and how they can improve
- Technical People
  - **Validation of things they've been asking to change for years**
  - To learn from another expert in the industry
  - To get another set of eyes on a problem they've been battling
  - To prove they're smarter than you
  - An honest assessment of what they're doing right and wrong and how they can improve

Ember Crooks
XTIVIA
ecrooks@xtivia.com

XTIVIA

Session code: D16

IDUG
Leading the Db2 User Community since 1988

Please fill out your session evaluation before leaving!

Ember is a Db2 Lead DBA and Delivery Manager at XTIVIA. She is responsible for the design, build, and management of a wide range of Db2 databases for multiple clients. Ember has 17 years of experience with Db2 on Linux, Unix, and Windows platforms. She is the founder and principal author of the popular datageek.blog technical blog where she educates herself and others through example and case study. Ember is an IBM Gold Consultant and IBM Champion in Information Management.