Driving Down Database Development Dollars

TIPS FOR DBAS AND APPLICATION PROGRAMMERS TO HELP REDUCE COSTS

Marcus Davage
Lead Product Developer, BMC Software

Marcus Davage

Lead Product Developer, BMC Software

Agenda

01. The challenge

02. The characters

- The Dev (Application Developers)
- The DBA (Database Administrators)
- The Ops (Operations)
- The DevOps Consultant

03. The collaboration

04. The conclusion

The Challenge

bmc

The Challenge

The characters

- The Dev (Application Developers)
- The DBA (Database Administrators)
- The Ops (Operations)
- The DevOps Consultant

© Copyright 2024 BMC Software, Inc.

The Dev

© Copyright 2024 BMC Software, Inc.

The challenges

- Skills Gap
- Security and Compliance
- Integration Challenges
- Modernization Demands
- Cost Management
- Agility and DevOps
- Compatibility and Legacy Dependencies

© Copyright 2024 BMC Software, Inc.
© intersystems and Cicso


Pandemic Problems
<innovate>
<currency>
<more quickly>
<more correctly>
<tighter deadlines>
<more code>
<shorter cycles>
<changing requirements>
<heavier workloads>
<REDUCE COSTS!>

Pandemic problems

59%

of developers say project deadlines are the most stressful part of the job

49%

find balancing workloads challenging

47%

say it's "changing project requirements"

55%

are reluctant to tell their manager if they feel overworked

89%

report "feeling under immense pressure at work"

© intersystems and Cicso

© Copyright 2024 BMC Software, Inc.

Post-pandemic problems

15%

have found it has been difficult to try and manage expectations

31%

have had trouble communicating with relevant stakeholders or colleagues

11%

say the biggest challenge has been processes taking longer

46%

of developers think stress could be lessened by flexible working times

43%

would welcome extra resource and tools to automate processes as well as access to online communities

© intersystems

© Copyright 2024 BMC Software, Inc.

Careless coding costs!

Cartesian joins

Table T1

ID COL1
1 'Baby Monitor'
2 'iTouch 1'
3 'iPad 2'
4 'iPhone 6'

6 'iPhone 3'
7 'Samsung Galaxy S22'
8 'IBM z16 Model A02'
9 'Raspberry Pi 4'
10 'BBC Micro Model B'

Table T2

ID COL1
1 'Dave'
2 'Jenny'
3 'Jess'
4 'Terry'
5 'Pascal'
6 'Helmut'
7 'Roberto'
8 'Sasha'
9 'Alexa'
10 'Marcus'

© Copyright 2024 BMC Software, Inc.

Careless coding costs!

Cartesian joins

SELECT T1.*, T2.* FROM T1, T2 ;

```
"1","Baby Monitor","1","Dave"
"1","Baby Monitor","2","Jenny"
"1","Baby Monitor","3","Jess"
"1","Baby Monitor","4","Terry"
"1","Baby Monitor","5","Pascal"
"1","Baby Monitor","6","Helmut"
"1","Baby Monitor","7","Roberto"
"1","Baby Monitor","8","Sasha"
"1","Baby Monitor","9","Alexa"
"1","Baby Monitor","10","Marcus"
"2","iTouch 1","1","Dave"
"2","iTouch 1","2","Jenny"
"2","iTouch 1","3","Jess"
"2","iTouch 1","4","Terry"
"2","iTouch 1","5","Pascal"
"2","iTouch 1","6","Helmut"
"2","iTouch 1","7","Roberto"
"2","iTouch 1","8","Sasha"
"2","iTouch 1","9","Alexa"
"2","iTouch 1","10","Marcus"
"3","iPad 2","1","Dave"
"3","iPad 2","2","Jenny"
"3","iPad 2","3","Jess"
"3","iPad 2","4","Terry"
"3","iPad 2","5","Pascal"
"3","iPad 2","6","Helmut"
"3","iPad 2","7","Roberto"
"3","iPad 2","8","Sasha"
"3","iPad 2","9","Alexa"
"3","iPad 2","10","Marcus"
"4","iPhone 6","1","Dave"
"4","iPhone 6","2","Jenny"
"4","iPhone 6","3","Jess"
"4","iPhone 6","4","Terry"
"4","iPhone 6","5","Pascal"
"4","iPhone 6","6","Helmut"
"4","iPhone 6","7","Roberto"
"4","iPhone 6","8","Sasha"
"4","iPhone 6","9","Alexa"
"4","iPhone 6","10","Marcus"
"7","Samsung Galaxy S22","1","Dave"
"7","Samsung Galaxy S22","2","Jenny"
"7","Samsung Galaxy S22","3","Jess"
"7","Samsung Galaxy S22","4","Terry"
"7","Samsung Galaxy S22","5","Pascal"
"7","Samsung Galaxy S22","6","Helmut"
"7","Samsung Galaxy S22","7","Roberto"
"7","Samsung Galaxy S22","8","Sasha"
"7","Samsung Galaxy S22","9","Alexa"
"7","Samsung Galaxy S22","10","Marcus"
"8","IBM z16 Model A02","1","Dave"
"8","IBM z16 Model A02","2","Jenny"
"8","IBM z16 Model A02","3","Jess"
"8","IBM z16 Model A02","4","Terry"
"8","IBM z16 Model A02","5","Pascal"
"8","IBM z16 Model A02","6","Helmut"
"8","IBM z16 Model A02","7","Roberto"
"8","IBM z16 Model A02","8","Sasha"
"8","IBM z16 Model A02","9","Alexa"
"8","IBM z16 Model A02","10","Marcus"
"9","Raspberry Pi 4","1","Dave"
"9","Raspberry Pi 4","2","Jenny"
"9","Raspberry Pi 4","3","Jess"
"9","Raspberry Pi 4","4","Terry"
"9","Raspberry Pi 4","5","Pascal"
"9","Raspberry Pi 4","6","Helmut"
"9","Raspberry Pi 4","7","Roberto"
"9","Raspberry Pi 4","8","Sasha"
"9","Raspberry Pi 4","9","Alexa"
"9","Raspberry Pi 4","10","Marcus"
"10","BBC Micro Model B","1","Dave"
"10","BBC Micro Model B","2","Jenny"
"10","BBC Micro Model B","3","Jess"
"10","BBC Micro Model B","4","Terry"
"10","BBC Micro Model B","5","Pascal"
"10","BBC Micro Model B","6","Helmut"
"10","BBC Micro Model B","7","Roberto"
"10","BBC Micro Model B","8","Sasha"
"10","BBC Micro Model B","9","Alexa"
"10","BBC Micro Model B","10","Marcus"
```

© Copyright 2024 BMC Software, Inc.

Careless coding costs!

Cartesian joins

- Current date

```
SELECT CURRENT DATE
  INTO :CURRDATE
  FROM 70_MILLION_ROW_ONLINE_CUSTOMER_TABLE;
```

- No qualifiers
- No WHERE clause
- No predicates
- No really!
- Per transaction!

- The first answer:

```
SELECT CURRENT DATE
  INTO :CURRDATE
  FROM SYSIBM.SYSDUMMY1 ;
```

- The next answer:

```
EXEC SQL
  SET :CURRDATE = CURRENT DATE;
```


© Copyright 2024 BMC Software, Inc.

Don’t even bother!

COBOL

```
01  CURRDATE        Pic 9(8).
...
   Move Function Current-Date(1:8) to CURRDATE
```

Assembler

```
      STCK     TODCLOCK     CURRENT TIME
      STCKCONV STCKVAL=TODCLOCK,
                       CONVVAL=CURRDATE,
                       TIMETYPE=BIN,
                       DATETYPE=YYYYMMDD
TODCLOCK DS      XL8        TOD CLOCK VALUE
CURRDATE DS      CL16        CONVERTED VALUE
```

Java

```
public class ZUtil
  extends java.lang.Object

public static void getTodClock
  (byte[] buffer)
```

© Copyright 2024 BMC Software, Inc.

Careless coding costs!

- Current application compatibility

```
SELECT CURRENT APPLICATION COMPATIBILITY
  INTO :CURRAPPL
  FROM SYSIBM.SYSDUMMY1 ;
```

- Missing index keys in WHERE clause
- When the colcard of first key column is 1

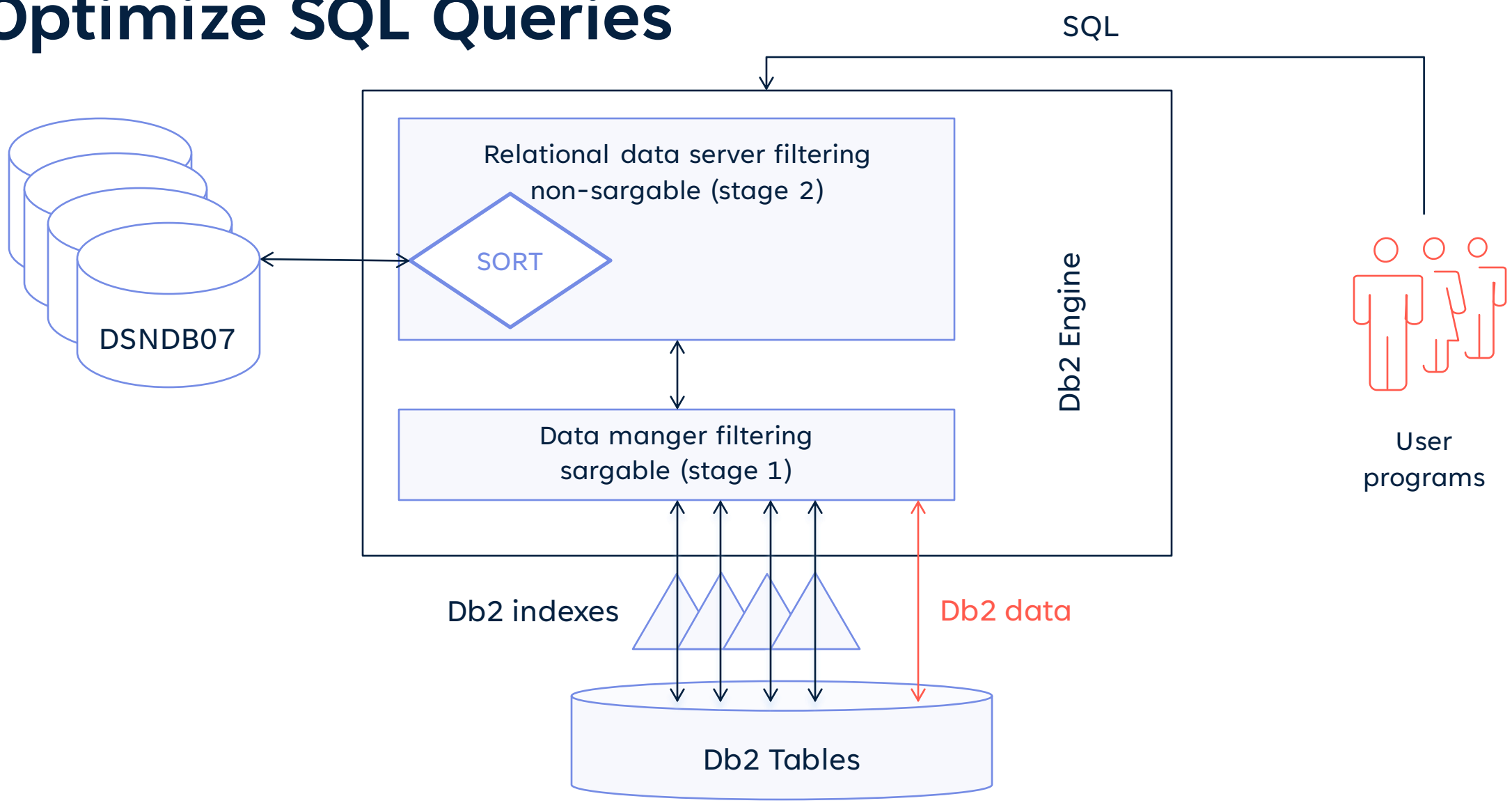
© Copyright 2024 BMC Software, Inc.

SQL tips

- Optimize SQL Queries
- Reuse Prepared Statements
- Data Compression
- Indexing Strategy
- Resource Constraints
- VS Code Db2 Extensions
- Reduce Data Transfer
- Connection Pooling
- Use Efficient Data Types
- Collaborate with DBAs
- Stay Informed
- EXPLAIN!

bmc © Copyright 2024 BMC Software, Inc.

Optimize SQL Queries



© Copyright 2024 BMC Software, Inc.

Ready player (1|4)

```
SELECT KEYCOL2, DATACOL1, DATACOL2, DATACOL3
FROM TABLE1
WHERE KEYCOL1=1;
-- Returns 2 rows
```

KEYCOL1	KEYCOL2	DATACOL1	DATACOL2	DATACOL3
1	2023-01-01	iPhone	14	Yellow
1	2023-10-02	Cake	Colin	Caterpillar
2	2023-06-27	Samsung	S22	Black
3	2023-12-25	Cake	Christmas	Icing

© Copyright 2024 BMC Software, Inc.

Ready player (2|4)

```
SELECT KEYCOL2, DATACOL1, DATACOL2, DATACOL3
FROM TABLE1
WHERE KEYCOL1=1 AND KEYCOL2= '2023-10-02';
-- Returns 1 row
```

KEYCOL1	KEYCOL2	DATACOL1	DATACOL2	DATACOL3
1	2023-01-01	iPhone	14	Yellow
1	2023-10-02	Cake	Colin	Caterpillar
2	2023-06-27	Samsung	S22	Black
3	2023-12-25	Cake	Christmas	Icing

© Copyright 2024 BMC Software, Inc.

Ready player (3|4)

```
SELECT KEYCOL2, DATACOL1, DATACOL2, DATACOL3
FROM TABLE1
WHERE KEYCOL1=1 AND DATACOL1 = 'Cake';
-- Returns 1 row
```

KEYCOL1	KEYCOL2	DATACOL1	DATACOL2	DATACOL3
1	2023-01-01	iPhone	14	Yellow
1	2023-10-02	Cake	Colin	Caterpillar
2	2023-06-27	Samsung	S22	Black
3	2023-12-25	Cake	Christmas	Icing

© Copyright 2024 BMC Software, Inc.

Ready player (4|4)

```
SELECT KEYCOL2, DATACOL1, DATACOL2, DATACOL3
FROM TABLE1
WHERE DATACOL1 = 'Cake'
ORDER BY DATACOL3;
-- Returns 2 rows
```

KEYCOL1	KEYCOL2	DATACOL1	DATACOL2	DATACOL3
1	2023-01-01	iPhone	14	Yellow
1	2023-10-02	Cake	Colin	Caterpillar
2	2023-06-27	Samsung	S22	Black
3	2023-12-25	Cake	Christmas	Icing

© Copyright 2024 BMC Software, Inc.

DIY query rewrite (1|3)

```
SELECT EMPNO, LASTNAME
FROM EMP
WHERE YEAR(HIREDATE) = ‘2009’ ;
-- Instead, do
SELECT EMPNO, LASTNAME
FROM EMP
WHERE HIREDATE BETWEEN ‘2009-01-01’ AND ‘2009-12-31’ ;
```

© Copyright 2024 BMC Software, Inc.

DIY query rewrite (2|3)

```
-- Usually, you find
WHERE COLUMN1 BETWEEN :HV1 AND :HV2
-- Don’t do this:
WHERE :HV BETWEEN COLUMN1 AND COLUMN2
-- Instead, do this
WHERE :HV >= COLUMN1 AND :HV <= COLUMN2
```

© Copyright 2024 BMC Software, Inc.

DIY query rewrite (3|3)

Code Most Restrictive Predicates First

- Indexes being used
- Stage 1 or Stage 2
- Predicate type
  - =, >, <, BETWEEN, etc.
- In order of appearance

Know your data

- Use higher cardinality columns first

© Copyright 2024 BMC Software, Inc.

Reduce data transfer (1|2)

- IBM are adding more Stage 1 predicates with each version
- Fetch only the columns and rows needed
- Don’t do SELECT *
- Use indexed predicates
  - Stage 1 / 2
  - Join on Indexed columns
- SORTs
  - Use CLUSTERING index instead of ORDER BY
  - Join on Clustered columns
  - Avoid if you can

© Copyright 2024 BMC Software, Inc.

Reduce data transfer (2|2)

Good boy...FETCH!

- FIRST n ROWS or FETCH FIRST n ROWS ONLY to limit result sets.
- Scrollable cursors
- FETCH NEXT n ROWS
- OPTIMIZE FOR n ROWS

Native Stored Procedures

- Free! zIIP
- Reduces network traffic
- Reusable

© Copyright 2024 BMC Software, Inc.

Indexing strategy

- Index on Primary keys (obviously)
- Index on Foreign keys
- Clustering indexes
  - JOIN
  - RI
- +1 +1 +1 +1 Oops! -1

© Copyright 2024 BMC Software, Inc.

Visual Studio Code (1|7)

```
220                 CURRENT-MINUTE ':' CURRENT-SECOND.
221         * TODO this is a task tag
222             PERFORM 700-OPEN-FILES .
223             PERFORM 800-INIT-REPORT .
224
225             PERFORM 730-READ-CUSTOMER-FILE .
226             PERFORM 100-PROCESS-TRANSACTIONS
227                     UNTIL WS-TRAN-EOF = 'Y' .
228
229
230             PERFORM 850-REPORT-TRAN-STATS .
231             PERFORM 790-CLOSE-FILES .
232
233             ACCEPT CURRENT-DATE FROM DATE.
234             ACCEPT CURRENT-TIME FROM TIME.
235             DISPLAY 'SAM1 STOPPED DATE = ' CURRENT-MONTH '/'
236                   CURRENT-DAY '/' CURRENT-YEAR   '  (mm/dd/yy)'.
237             DISPLAY '              TIME = ' CURRENT-HOUR ':'
```

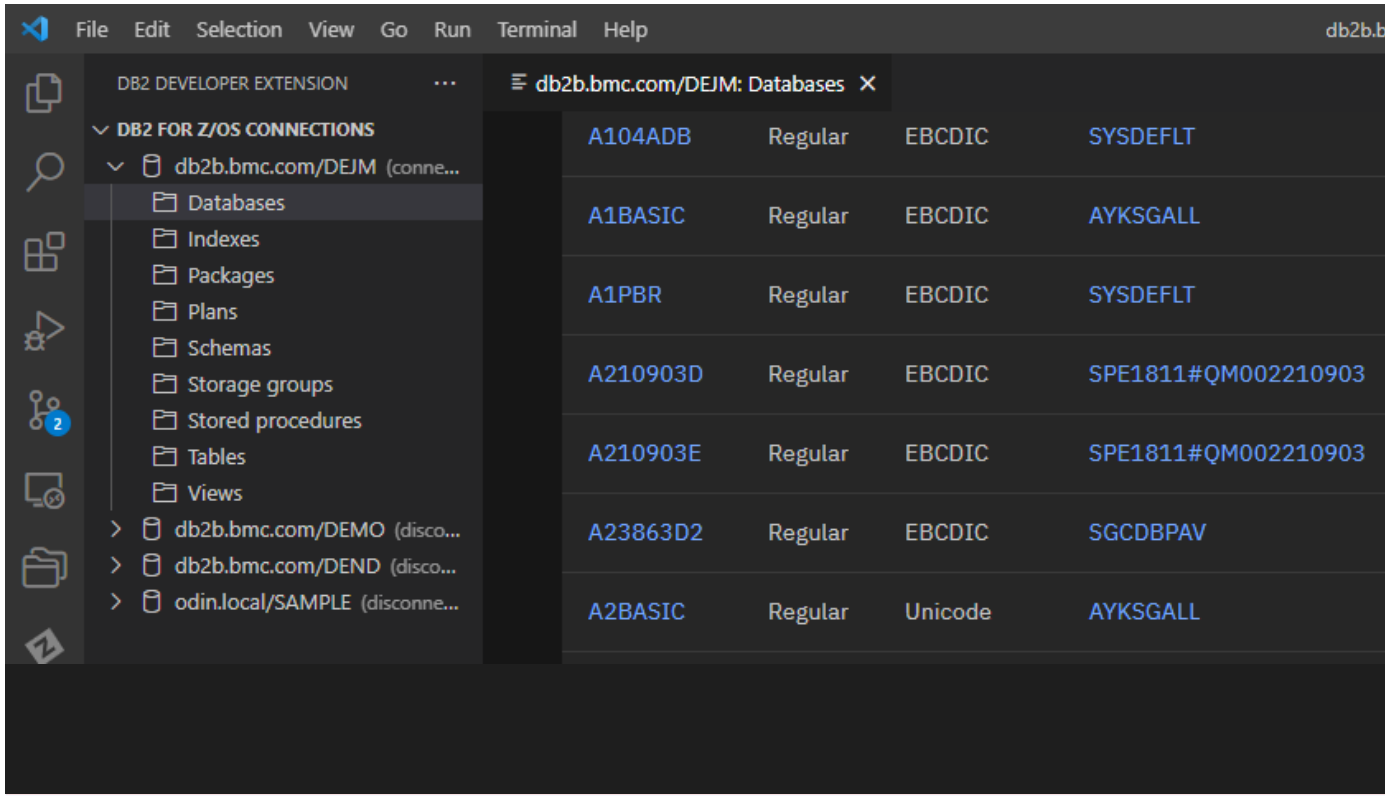
© Copyright 2024 BMC Software, Inc.

Visual Studio Code (2|7)

```
STARTIT  DS     0H                                                  00150000
         SPACE 2                                                    00160000
         LA     R0,72+1000          get savearea plus workarea      00170000
         GETMAIN R,LV=(0)                                           00180000
         MVI    0(R1),X'00'         MOVE X'00' TO FIRST BYTE        00190002
         LR     R2,R1               SAVE POINTER IN EVEN REG        00200002
         LA     R4,1(R1)            SET RECEIVING POINTER           00210000
         LR     R5,R0               SET RECEIVING LENGTH            00220000
         BCTR   R5,R0               DECREMENT LENGTH                00230000
         LA     R5,0(R5)            CLEAR HIGH ORDER BYTE           00240000
         LA     R3,1                SET SENDING LENGTH              00250000
         MVCL   R4,R2               INSTRUCTION PADS WITH X'00'     00260002
         ST     R13,4(R1)           SAVE BACK CHAIN                 00270000
         ST     R1,8(R13)           SET FORWARD CHAIN               00280000
         LR     R2,R1               SAVE NEW SAVEAREA ADDRESS       00290002
         L      R15,16(R13)         RESTORE REG 15                  00300000
         ST     R0,16(R13)          SAVE SAVEAREA LENGTH            00310000
         LM     R0,R1,20(R13)       RESTORE REGS USED IN GETMAIN    00320000
         LR     R13,R2              SET SAVEAREA POINTER            00330002
         USING WORK,R13             USING ON WORKAREA VIA R13       00340000
         EJECT                                                      00350000
```

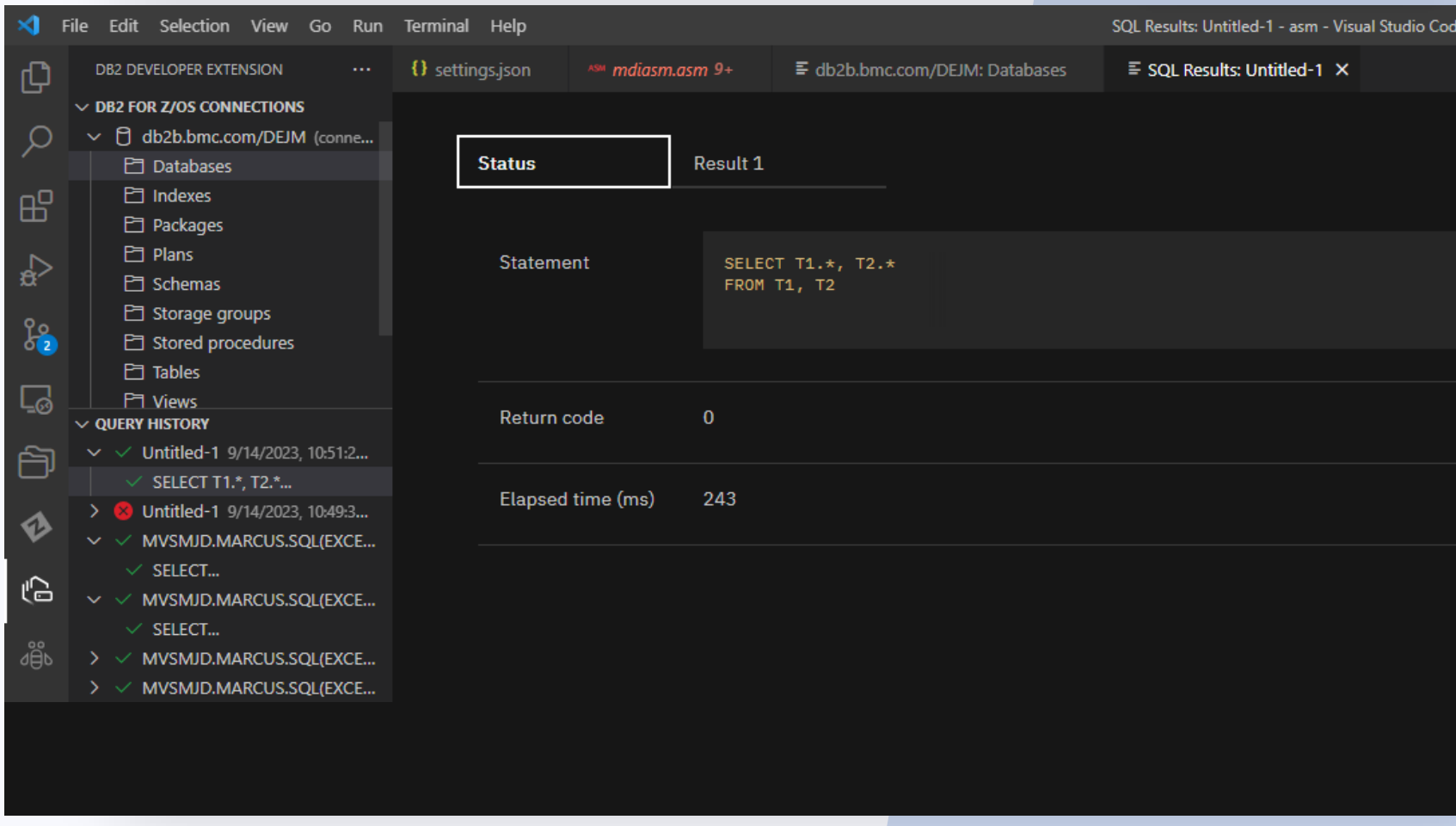
© Copyright 2024 BMC Software, Inc.

Visual Studio Code (3|7)



© Copyright 2024 BMC Software, Inc.

Visual Studio Code(4|7)



© Copyright 2024 BMC Software, Inc.

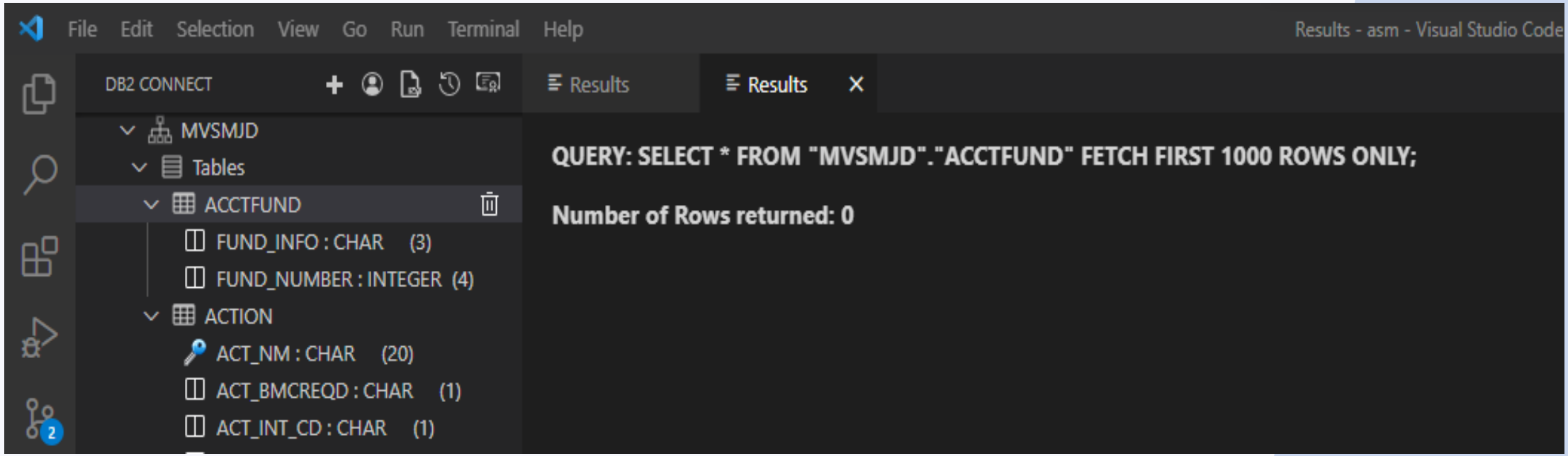
Visual Studio Code (5|7)

Status | **Result 1**

ID	COL1	ID	COL1
1	Baby Monitor	1	Dave
1	Baby Monitor	2	Jenny
1	Baby Monitor	3	Jess
1	Baby Monitor	4	Terry
1	Baby Monitor	5	Pascal
1	Baby Monitor	6	Helmut
1	Baby Monitor	7	Roberto
1	Baby Monitor	8	Sasha
1	Baby Monitor	9	Alexa

© Copyright 2024 BMC Software, Inc.

Visual Studio Code (6|7)



© Copyright 2024 BMC Software, Inc.

Visual Studio Code (7|7)

File Edit Selection View Go Run Terminal Help • MVSMJD.MARCUS.SQL(#0DEMO) -

ZOWE EXPLORER

DATA SETS

- MVSMJD.MARCUS.OLDJCL
- MVSMJD.MARCUS.OUTPUT
- MVSMJD.MARCUS.PDF
- MVSMJD.MARCUS.PDFINFO
- MVSMJD.MARCUS.PLIB
- MVSMJD.MARCUS.PROCLIB
- MVSMJD.MARCUS.SLIB
- MVSMJD.MARCUS.SQL
  - #0DEMO
  - #1SQL
  - #ALIAS
  - #ALIASES

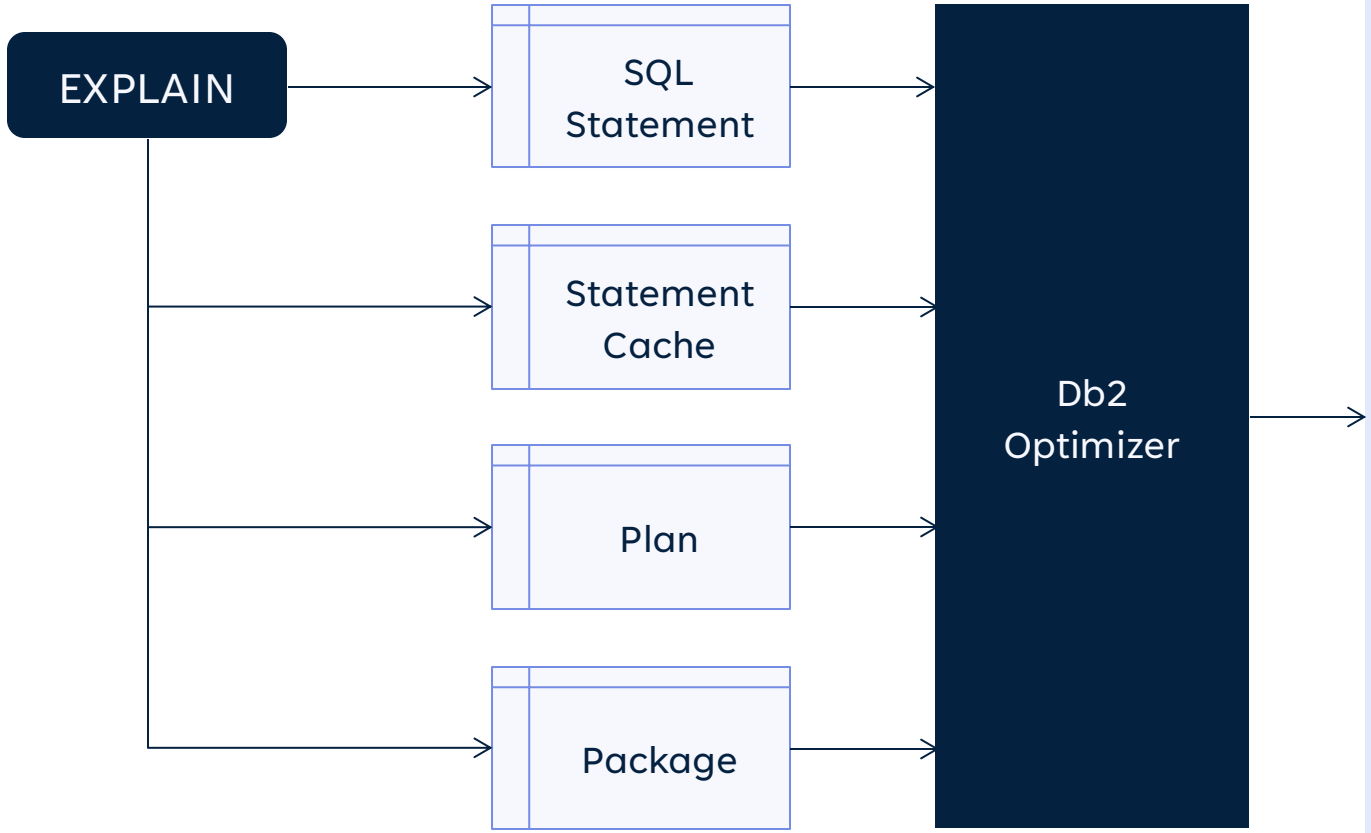
MVSMJD.MARCUS.SQL(#0DEMO) 7

C: > Users > mdavage > .vscode > extensions > zowe.vscode-extension-for-zowe-2.11.0 > resources > temp

```
-- ######################################################################
-- # Basic MERGE statement
-- # See https://www.ibm.com/docs/en/db2-for-zos/13?topic=statements-merge
-- ######################################################################
MERGE INTO table-name
AS correlation-name (column-name)
USING table-reference
ON search-condition
WHEN MATCHED THEN
    UPDATE SET
        column-name = expression
WHEN NOT MATCHED THEN
    INSERT (column-name)
    VALUES (expression)
```


© Copyright 2024 BMC Software, Inc.

Explain!

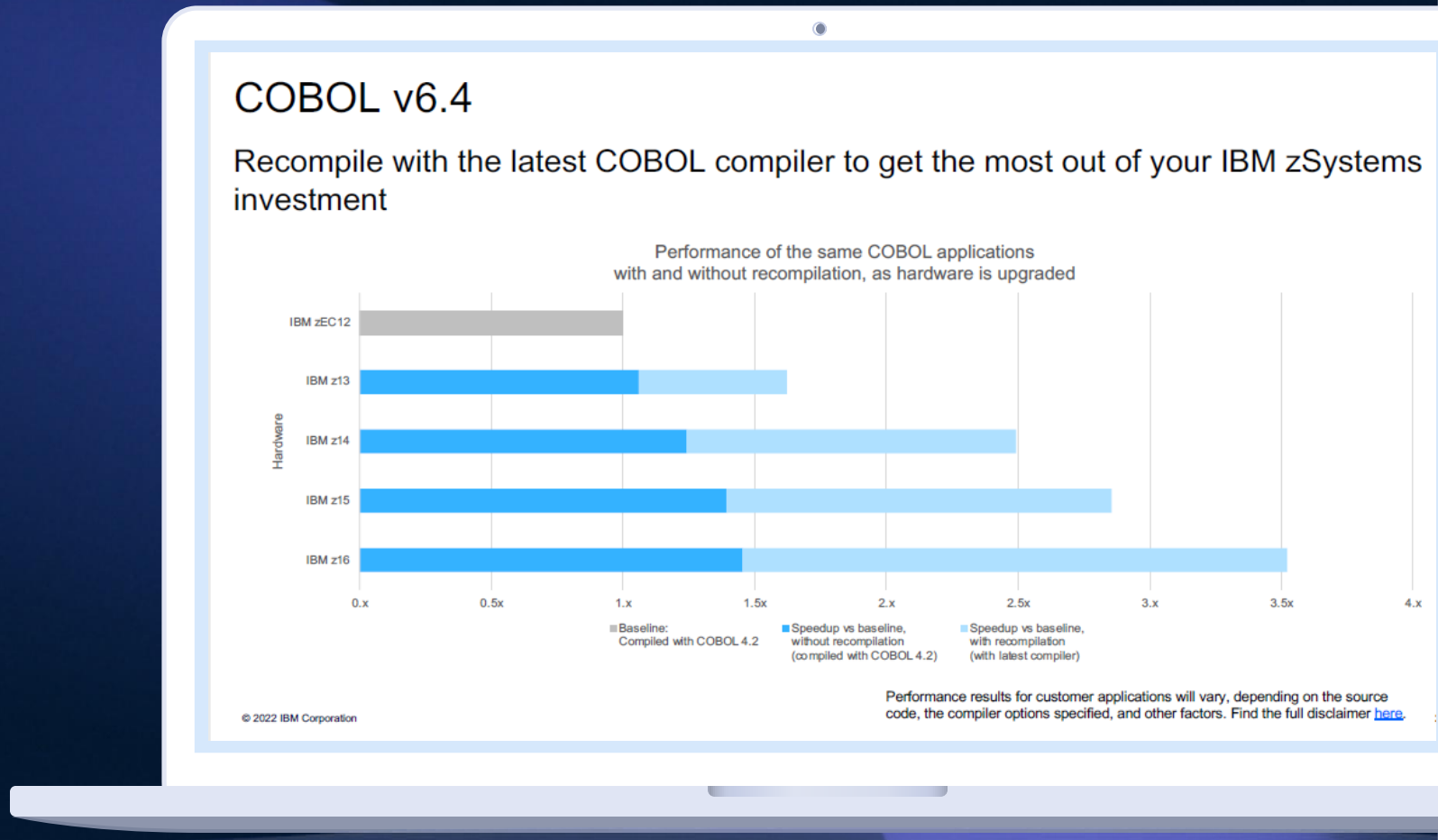


PLAN_TABLE	DSN_VIRTUAL_KEYTARGETS
DSN_VIRTUAL_INDEXES	DSN_PGRANGE_TABLE
DSN_PTASK_TABLE	DSN_KEYTGTDIST_TABLE
DSN_QUERYINFO_TABLE	DSN_STAT_FEEDBACK
DSN_STATEMNT_TABLE	DSN_STATEMENT_CACHE_TABLE
DSN_FUNCTION_TABLE	DSN_QUERY_TABLE
DSN_SORTKEY_TABLE	DSN_DETCOST_TABLE
DSN_SORT_TABLE	DSN_PGROUP_TABLE
DSN_PREDICATE_SELECTIVITY	DSN_FILTER_TABLE
DSN_PREDICAT_TABLE	DSN_STRUCT_TABLE
DSN_VIEWREF_TABLE	DSN_COLDIST_TABLE
DSN_USERQUERY_TABLE	intentionally left blank

bmc © Copyright 2024 BMC Software, Inc.

Compiler tips (1|3)

Keep your compiler up-to-date –
Save up to 20% CPU



© Copyright 2024 BMC Software, Inc.

Compiler tips (2|3)

- Review Compiler/Assembler/Link Options
- ALWAYS use the highest ARCH that your machine supports

ARCH(14) – z16	20 new instructions
ARCH(13) – z15	39 new instructions
ARCH(12) – z14	48 new instructions
ARCH(11) – z13	152 new instructions
ARCH(10) – zEC12	19 new instructions
ARCH(9) – z196	132 new instructions
ARCH(8) – z10	78 new instructions
ARCH(7) – z9	757 instructions

Never, ever, ever use an ARCH Level above the lowest level hardware in your environment, **Including** your DR Sites.

If your DR site is z13's then use ARCH(11).

© Copyright 2024 BMC Software, Inc.

Compiler tips (3|3)

- Use Rexx? Compile it!
- Compile off-mainframe?
  - Vendor Solutions
  - C/C++ or HLASM
  - Compile on PC using z-compatible compilers/assemblers
  - FTP object and DBRM to mainframe
  - Link and Bind on mainframe

© Copyright 2024 BMC Software, Inc.

Coding Tips (1|3)

- Horses for courses
- Clean Code
  - “Uncle” Bob Martin
- Keep short code paths
  - E.g. Use ADD instead of COMPUTE

- Cheat!
  - Stackoverflow – not very mainframe-aware
  - ChatGPT – surprisingly good!
  - TabNine / GitHub Copilot

© Copyright 2024 BMC Software, Inc.

Coding Tips (2|3)

Visual Studio Code

- COBOL/Java/C/C++/HLASM/PLI/Rexx Language Extensions

Performance Tools

- BMC AMI Strobe
- BMC AMI Apptune
- BMC AMI DevOps for Db2 – SQL Assurance

SonarQube

- Code smells
- Rules-based

© Copyright 2024 BMC Software, Inc.

Coding Tips (3|3)

- Automated Testing Tools
  - BMC AMI DevX
- REST APIs
- Java
- No, really. Java

© Copyright 2024 BMC Software, Inc.

Java on Z

- Java compiles to Byte Code. Always uses the latest ARCH
- Java can be run on zIIP – reducing software costs
  - Warning: if you overflow zIIP capacity the system may run the workload on a GP
- Java may run faster than COBOL on Sub Capacity mainframes as zIIPs always run at full speed
- Most mainframe performance monitors now support Java
- Better performance and garbage collection on z14 and up
- https://www.blackhillsoftware.com/news/2021/08/10/java-vs-c-drag-racing-on-z-os/

© Copyright 2024 BMC Software, Inc.

The DBA

© Copyright 2024 BMC Software, Inc.

The Challenges

- Data Growth
- Security and Compliance
- Cost Management
- Integration with New Technologies
- Vendor Support and Updates
- Data Privacy and Ethics
- Performance Optimization
- Disaster Recovery and High Availability
- Complexity of Database Environments
- Automation and DevOps
- Skills Shortages
- Migration and Modernization

© Copyright 2024 BMC Software, Inc.

DBA Cost-saving Tips

- Optimize Query Performance
- Data Compression
- Buffer Pool Tuning
- Monitoring and Alerting
- Workload Management
- Automation and Scripting
- Housekeeping for Performance
- Index Management
- Archiving and Purging
- Capacity Planning
- Optimize Backup and Recovery
- Educate and Train Staff

bmc © Copyright 2024 BMC Software, Inc.

The Ops

© Copyright 2024 BMC Software, Inc.

The Challenges

- Legacy System Maintenance
- Resource Management
- High Availability
- Integration with Modern Technologies
- Legacy Application Modernization
- Energy Efficiency
- Automation and DevOps
- Security and Compliance
- Skilled Workforce
- Cost Management
- Capacity Planning
- Vendor Support and Updates
- Performance Optimization

© Copyright 2024 BMC Software, Inc.

Ops Tips

- Consolidation
- Automation
- Legacy Application Modernization
- Disaster Recovery Efficiency
- Monitoring and Performance Tuning
- DevOps and Automation Tools
- Application Performance Management
- Collaboration

© Copyright 2024 BMC Software, Inc.

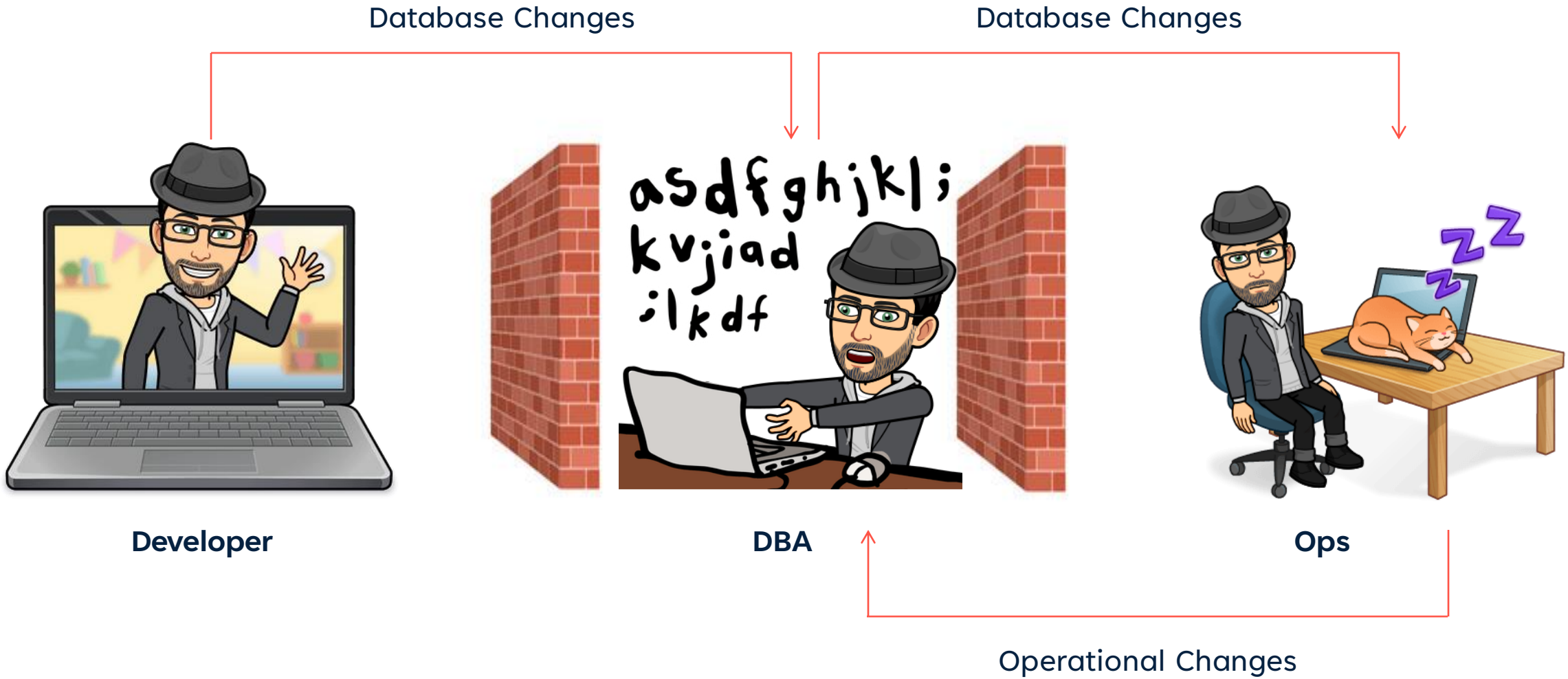
The DevOps Consultant

© Copyright 2024 BMC Software, Inc.

The Collaboration

bmc

The Collaboration



© Copyright 2024 BMC Software, Inc.

The Collaboration

© Copyright 2024 BMC Software, Inc.

The Collaboration

© Copyright 2024 BMC Software, Inc.

The Collaboration

Bad SQL is determined by the Environment that it runs in

Be sure to have accurate RUNSTATS for your environment

SQL performance is a joint effort between Developers and DBAs

App Developer

Database Admin

Developer	Unit Test	System Test	Accp. Test	Prod
Development	Smoke testing	**Quality Tests**	Manual UAT	RUN
Unit tests	Unit integration testing	Regression testing	Performance testing	Smoke testing
Code metrics	Functional testing - regression	Functional testing	Training	Monitoring
		Integration testing		

Visual Explain **Visual Explain** **Visual Explain**

Prevent Bad SQL from reaching Prod

Db2 Run Stats Info

© Copyright 2024 BMC Software, Inc.

The Conclusion

The future is bright...

The future is mainframe!

bmc

“

I predict that the last mainframe will be unplugged on March 15, 1996.”

- Stewart Alsop, March 1991

It’s clear that corporate customers still like to have centrally controlled, very predictable, reliable computing systems – exactly the kind of systems that IBM Specializes in.”

- Stewart Alsop, February 2002

© Copyright 2024 BMC Software, Inc.

The Conclusion

- 20B+ Monthly Db2 Transactions
- 10M+ Monthly Batch Jobs
- 99.99999% Uptime (Seven 9s)
- 80M+ Lines of Production Code
- 1,200+ Production Program Changes Monthly
- 40M Db2 Utility Jobs Annually
- DR: Transition 100% of Tier 1 Applications in 4 hours
- DR: Recover Tier 1 production Db2 tables in 4 hours our less

© Copyright 2023 BMC Software, Inc.

Thoroughly Modern Mainframe

- Quantum Safe Crypto
- Hybrid Cloud Development
- AI Enabled Processor
- Process up to One Trillion Web Transactions a Day
- Up to 300 billion AI inference operations a Day
- New Telum Processors

IBM z16
Announced April 5, 2022

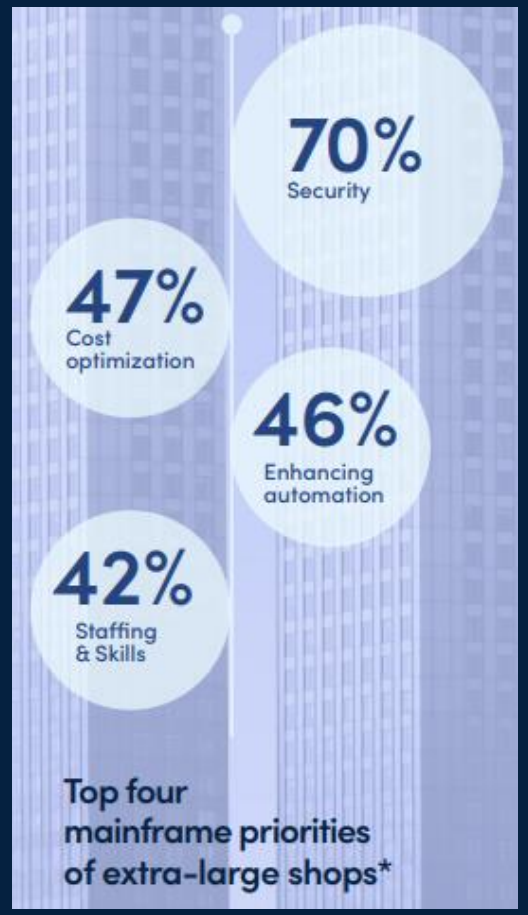
92%
of respondents see the mainframe long-term platform for growth

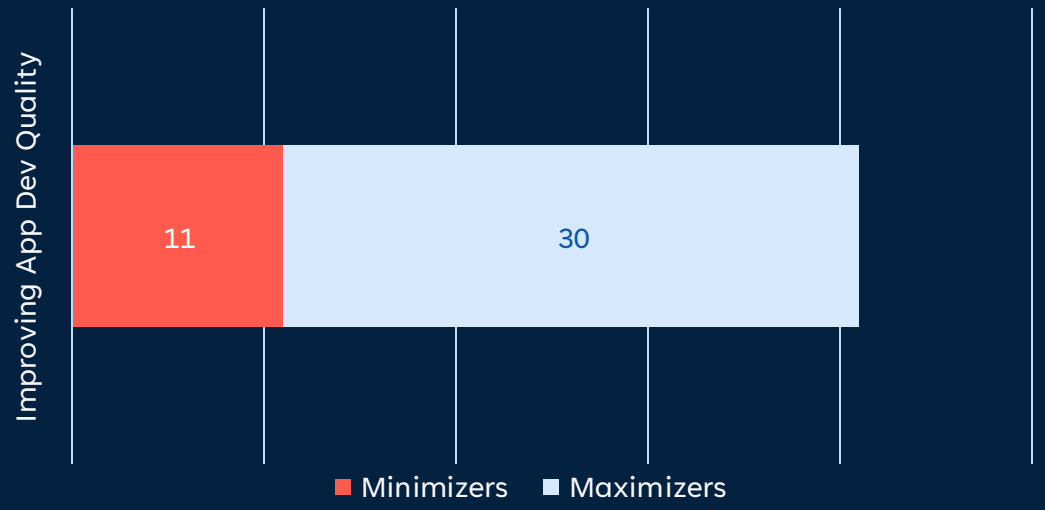
72%
72% of extra-large shops have more than 50% of their data on the mainframe

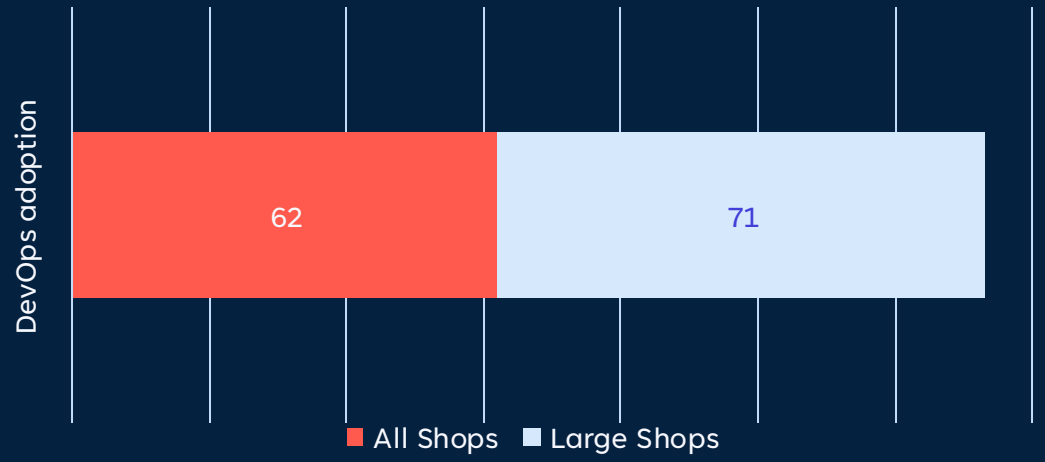
Source: BMC 2021 Mainframe Survey

bmc © Copyright 2024 BMC Software, Inc.

The Conclusion

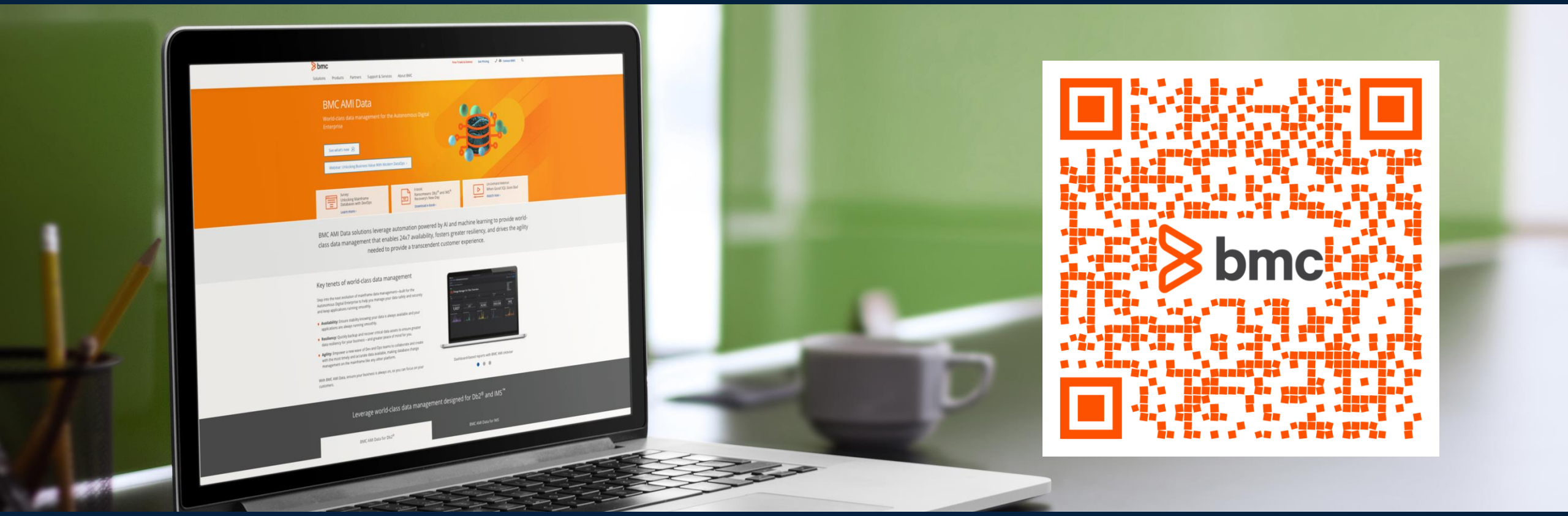






2023 BMC Mainframe Survey

© Copyright 2024 BMC Software, Inc.



Learn more at bmc.com/AMIdata

BMC AMI Data

© Copyright 2024 BMC Software, Inc.

BMC works with 86% of the Forbes Global 50 and customers and partners around the world to create their future. With our history of innovation, industry-leading automation, operations, and service management solutions, combined with unmatched flexibility, we help organizations free up time and space to become an Autonomous Digital Enterprise that conquers the opportunities ahead.

BMC—Run and Reinvent

www.bmc.com

BMC, the BMC logo, and BMC's other product names are the exclusive properties of BMC Software, Inc. or its affiliates, are registered or pending registration with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other trademarks or registered trademarks are the property of their respective owners. © Copyright 2023 BMC Software, Inc.