

1. Record Nr.	UNINA9910819163303321
Titolo	DB2 UDB for z/OS version 8 : everything you ever wanted to know-- and more // Bart Steegmans ... [et al.]
Pubbl/distr/stampa	White Plains, NY, : IBM, International Technical Support Organization, c2004
Edizione	[1st ed.]
Descrizione fisica	xxiv, 1066 p. : ill
Collana	IBM redbooks
Altri autori (Persone)	SteegmansBart
Disciplina	005.75/85
Soggetti	Database management
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	"SG24-6079-00." "May 2004."
Nota di bibliografia	Includes bibliographical references (p. 1039-1040) and index.
Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Summary of changes -- July 2004 -- April 2005 -- Chapter 1. DB2 UDB for z/OS Version 8: At a glance -- 1.1 DB2 UDB for z/OS Version 8 - overview -- 1.2 Architecture -- 1.2.1 Unicode support -- 1.2.2 DB2 Connect and DRDA -- 1.2.3 Universal Driver for SQLJ and JDBC -- 1.2.4 Schema evolution -- 1.2.5 64-bit virtual storage -- 1.3 Usability, availability, and scalability -- 1.3.1 Partitioning -- 1.3.2 Create index dynamic statement invalidation -- 1.3.3 Minimize impact of creating deferred indexes -- 1.3.4 Column data type change -- 1.3.5 LOB ROWID transparency -- 1.3.6 Longer table and column names -- 1.3.7 SQL statements 2 MB long -- 1.3.8 Multiple DISTINCT clauses in SQL statements -- 1.3.9 More open data sets -- 1.3.10 More log data sets -- 1.3.11 CI size larger than 4 KB -- 1.4 Data warehouse -- 1.4.1 More tables in joins -- 1.4.2 Sparse index and in-memory workfiles for star join -- 1.4.3 Common table expression and recursive SQL -- 1.4.4 Materialized query tables -- 1.5 Performance -- 1.5.1 Locking improvements -- 1.5.2 Multi-row INSERT and FETCH -- 1.5.3 RUNSTATS improvements -- 1.5.4 Host variables impact on access paths -- 1.5.5 Index only access for VARCHAR -- 1.5.6 Backward index scan -- 1.5.7 Local SQL cache issues and short prepare -- 1.5.8 Multiple IN values -- 1.5.9 DDF performance --

1.5.10 Automatic space management -- 1.5.11 Dynamic statement cache statement ID in EXPLAIN -- 1.5.12 Instrumentation enhancements -- 1.5.13 Migration changes -- 1.6 System level point-in-time backup and recovery -- Chapter 2. Scalability -- 2.1 Where are we today? (The problem) -- 2.2 DBM1 virtual storage constraint relief needed -- 2.3 DBM1 - Major VSTOR consumers -- 2.4 How to solve this problem?.

2.5 64-bit processor support -- 2.6 64-bit memory architecture (The solution) -- 2.7 Central storage on zSeries (z990/z900/z800/z890) -- 2.8 REAL storage support on zSeries - provided -- 2.9 REAL storage support on zSeries - migration -- 2.10 Large REAL memory support -- 2.11 DB2 (pre-V8) z/Architecture exploitation -- 2.12 Increasing processor speeds -- 2.13 Larger buffer pools -- 2.14 Hiperpools -- 2.15 BPs in data spaces - benefits -- 2.16 BPs in data spaces - problems -- 2.17 Other considerations -- 2.18 64-bit virtual address space memory map -- 2.19 DB2 V8 exploitation of 64-bit virtual storage -- 2.20 Why implement 64-bit VSTOR support? -- 2.21 What is moving above the 2 GB Bar? -- 2.22 DB2 benefits of 64-bit VSTOR support -- 2.23 64-bit virtual buffer pool support -- 2.23.1 Configuration changes -- 2.23.2 Migration sizing of buffer pools -- 2.23.3 Buffer pool sizing -- 2.23.4 Additional buffer pool information -- 2.23.5 New error messages -- 2.24 RIDPOOL - moving above the bar -- 2.25 Compression dictionaries - moving above the bar -- 2.26 Sort pool - moving above the bar -- 2.27 EDM pool - DBDs/OBDs/DSC - moving above the bar -- 2.27.1 Dynamic statement cache -- 2.27.2 Storing the DBDs -- 2.27.3 Storage for plans and packages -- 2.28 Other virtual storage related enhancements -- 2.28.1 LOB data -- 2.28.2 IPCS IRLM and DB2 dump formatting enhancements -- 2.28.3 Up to 65 000 open data sets -- 2.29 IRLM V2.2 64-bit IRLM for DB2 V8 -- 2.30 Immediate benefits -- 2.31 DB2 code considerations -- 2.32 Summarizing 64-bit changes in DB2 -- 2.33 Support for 4096 partitions -- 2.33.1 Requirements for 4096 partitions -- 2.33.2 Maximum number of partitions -- 2.33.3 Table space size and number of partitions -- 2.33.4 New data set naming convention -- 2.33.5 Database commands support enhancement -- 2.33.6 Display database command support.

2.33.7 Database command support - OVERVIEW keyword -- 2.33.8 Some considerations -- 2.34 More active log data sets -- 2.35 Increased maximum number of archive log data sets -- 2.36 Complex joins - up to 225 tables -- 2.36.1 Complex joins - up to 225 tables - 2 -- 2.36.2 Considerations for optimization thresholds -- 2.36.3 Affected interfaces -- 2.37 Longer SQL statements -- 2.37.1 Using 2 MB statements with dynamic SQL -- 2.37.2 Long SQL statement considerations -- 2.38 Long predicates -- 2.39 Long index keys -- Chapter 3. Availability -- 3.1 Availability - overview -- 3.2 Availability - 2 -- 3.3 Availability - 3 -- 3.4 Why partitioned table spaces -- 3.4.1 VLDB - DB2 objects keep getting bigger! -- 3.5 V7 Partitioned tables -- 3.5.1 Creating a partitioned table in Version 7 and prior -- 3.5.2 Creating the partitioning index -- 3.5.3 V7 Partitioned table space - index-controlled partitioning -- 3.5.4 Logical and physical partitions -- 3.5.5 Challenges when using NPIs prior to DB2 V8 -- 3.5.6 V7 and prior - contention on the NPI -- 3.6 V8 Partitioned tables -- 3.6.1 V8 partitioned tables, table-controlled partitioning -- 3.6.2 V8 - Creating partitioned tables -- 3.6.3 Converting to table-controlled partitioning -- 3.6.4 Catalog support for table-controlled partitioning -- 3.7 DB2 V8 classification of indexes -- 3.7.1 Index classification -- 3.7.2 Partitioning indexes -- 3.7.3 Secondary indexes -- 3.7.4 Partitioned and non-partitioned indexes -- 3.7.5 Partitioned and non-partitioned

partitioning indexes -- 3.7.6 Partitioned and non-partitioned secondary indexes -- 3.8 Data-partitioned secondary indexes -- 3.8.1 Creating a data-partitioned secondary index -- 3.8.2 DPSIs and utility operations -- 3.8.3 Data sharing overhead reduction -- 3.8.4 DPSI query performance -- 3.8.5 Design considerations - initial thoughts -- 3.8.6 Utility operations - DPSIs.

3.8.7 System planning and administration of DPSIs -- 3.9 Displaying indexes in DB V8 -- 3.9.1 Displaying a partitioned partitioning index -- 3.9.2 Displaying non-partitioned indexes -- 3.9.3 Displaying data-partitioned secondary indexes -- 3.10 Clustering indexes -- 3.10.1 A clustering index can be a secondary index -- 3.10.2 Clustering NPSI -- 3.10.3 Clustering DPSI -- 3.11 Online schema changes -- 3.11.1 The availability story -- 3.11.2 Performing schema changes today (V7) -- 3.11.3 What is new in V8 -- 3.11.4 Key benefits -- 3.11.5 Altering tables -- 3.11.6 Altering column data types -- 3.11.7 Supported alter data types -- 3.11.8 Alter data type syntax -- 3.11.9 What happens to the table? -- 3.11.10 What happens to the data? -- 3.11.11 What happens to dependent indexes? -- 3.11.12 How about referencing views and check constraints? -- 3.11.13 Considerations when changing data types -- 3.11.14 Restrictions -- 3.11.15 Operational impact -- 3.11.16 Altering index attributes -- 3.11.17 Altering index add column syntax -- 3.11.18 Alter index add column -- 3.11.19 Restrictions -- 3.11.20 RBDP considerations -- 3.12 Versioning -- 3.12.1 Catalog support for versioning -- 3.12.2 Minimize the number of active versions -- 3.12.3 Reclaiming versions -- 3.12.4 Segmented table space example -- 3.12.5 A word on system pages -- 3.13 Partition management -- 3.13.1 Adding a partition to a partitioned table space - current situation -- 3.13.2 ALTER TABLE ADD PARTITION syntax -- 3.13.3 Add a partition to a partitioned table space -- 3.13.4 Adding a partition - considerations -- 3.13.5 Rotate partition - overview -- 3.13.6 Alter table rotate partition syntax -- 3.13.7 Rotate partition example -- 3.13.8 Rotate partition effect -- 3.13.9 Considerations when rotating partitions -- 3.13.10 Display command output -- 3.13.11 Alter partition boundary.

3.13.12 Alter partition boundary example -- 3.13.13 Rebalance partitions -- 3.14 Other index enhancements -- 3.14.1 Altering index padding attribute -- 3.14.2 Alter index not padded/padded -- 3.14.3 Alter the clustering attribute of indexes -- 3.14.4 RBDP Index avoidance -- 3.14.5 Index creation enhancements -- 3.15 Reviewing DBET states used by online schema evolution -- 3.15.1 Review of new DBET states -- 3.16 System level point-in-time recovery -- 3.16.1 Prerequisites for this feature -- 3.16.2 BACKUP SYSTEM -- 3.16.3 BACKUP SYSTEM operation -- 3.16.4 RESTORE SYSTEM operation -- 3.17 More online ZPARAMs -- 3.18 Other availability enhancements -- 3.18.1 Control interval larger than 4KB -- 3.18.2 Monitoring system checkpoints and log offload activity -- 3.18.3 Log Monitoring long running UR backout -- 3.18.4 Detecting long readers -- 3.18.5 Lock holder can inherit WLM priority from lock waiter -- 3.18.6 Lock escalation IFCID -- 3.18.7 Partitioning key update enhancements -- 3.18.8 Improved LPL recovery -- 3.18.9 SMART DB2 extent sizes for DB2 managed objects -- 3.18.10 Logging manageability enhancements -- Chapter 4. SQL enhancements -- 4.1 Breaking SQL limitations -- 4.2 Static scrollable cursors - V7 review -- 4.2.1 Sensitive and insensitive cursors - V7 review -- 4.3 Dynamic scrollable cursors - V8 new function -- 4.3.1 DECLARE CURSOR syntax -- 4.3.2 Declare cursor - new attributes -- 4.3.3 FETCH syntax using dynamic scrollable cursors -- 4.3.4 Implications on FETCH -- 4.3.5 Locking with dynamic scrollable cursors -- 4.3.6 UPDATE using dynamic scrollable cursors --

4.3.7 Dynamic scrollable cursors considerations -- 4.3.8 DRDA considerations -- 4.3.9 Dynamic scrollable cursor example - 1 -- 4.3.10 Dynamic scrollable cursor example - 2 -- 4.3.11 Cursor type comparison -- 4.4 Multi-row FETCH and INSERT -- 4.5 Multi-row FETCH.  
4.5.1 DECLARE CURSOR syntax.

---