DB2 LUW Administration for SAP

DB2@SAP 9 Deep Compression

JongHwan Kim
johkim@kr.ibm.com
FTSS, IM Team
SWG, IBM Korea
Disclaimer & Trademarks

© Copyright 2007 IBM Corporation. All Rights Reserved.

Neither this documentation nor any part of it may be copied or reproduced in any form or by any means or translated into another language, without the prior consent of all of the above mentioned copyright owners.

IBM makes no warranties or representations with respect to the content hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. IBM assumes no responsibility for any errors that may appear in this document. The information contained in this document is subject to change without any notice. IBM reserves the right to make any such changes without obligation to notify any person of such revision or changes. IBM makes no commitment to keep the information contained herein up to date.

The information in this document concerning non-IBM products was obtained from the supplier(s) of those products. IBM has not tested such products and cannot confirm the accuracy of the performance, compatibility or any other claims related to non-IBM products. Questions about the capabilities of non-IBM products should be addressed to the supplier(s) of those.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

Trademarks

DB2, DB2 Universal Database, AIX, the ebusiness logo, IBM, the IBM logo, xSeries, and the eServer logo are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

A full list of U. S. trademarks owned by IBM may be found at http://iplswww.nas.ibm.com/wpts/trademarks/trademar.htm.

Linux is a registered trademark of Linus Torvalds.

SAP, R/3 and all other SAP product and service names mentioned herein are trademarks or registered trademarks of SAP AG in Germany and several other countries.

Oracle is a registered trademark of Oracle Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Other company, product or service names may be trademarks or service marks of others.
DB2 9 Deep Compression

- **Unit Objectives**

  After completing this unit, you will be able to:
  
  - Understand DB2 deep compression technology and benefit
  
  - Understand how DB2 deep compression feature can be used in SAP system
1.1 SAP Databases are growing…

- Continuously growing (Number of tables, Size)
  - SAP R/3 4.6D: ~10,000 - 15,000 tables
  - mySAP ERP 2005: > 50,000 tables

- SAP NetWeaver BI
  - 10+ TB databases
  - TB-size tables
  - Customers planning for 50, 100, … TB

- Large SAP customers have hundreds of SAP systems installed
1.2 DB2 Storage Optimization Solution

- Prior to DB2 9
  - V8 GA - NULL and Default Value Compression
    • No disk storage consumed for NULL column values, zero length data in variable length columns and system default values
  - V8 GA - Multidimensional Clustering (MDC)
    • Significant index compression can be achieved through block indexes
      • One key per thousands of records (vs one key per record with traditional indexes)
  - V8 FP4 - Database Backup Compression
    • Smaller backup images; compress index and If/lob tablespaces
- DB2 9 - Deep Compression
1.3 DB2 9 Deep Compression example – Table GLPCA

### Table GLPCA: EC-PCA - 102 columns

<table>
<thead>
<tr>
<th>RCLNT</th>
<th>GL_SIRID</th>
<th>RLDNR</th>
<th>RRCTY</th>
<th>RVERS</th>
<th>RYEAR</th>
<th>RTCUR</th>
<th>RUNIT</th>
<th>LRCRK</th>
<th>POPER</th>
<th>DOCCT</th>
<th>DOCNR</th>
<th>DOCLN</th>
<th>RBUKRS</th>
<th>RPRCTR</th>
<th>RHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>4751</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>1</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4752</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>2</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4753</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>3</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4754</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td>1</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4755</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td>2</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Compression dictionary

<table>
<thead>
<tr>
<th>x'01C</th>
<th>8A,0,0,1995,DEM,,S,1,A</th>
</tr>
</thead>
<tbody>
<tr>
<td>F'67t</td>
<td>1000,1402</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

#### Uncompressed table

<table>
<thead>
<tr>
<th>RCLNT</th>
<th>GL_SIRID</th>
<th>RLDNR</th>
<th>RRCTY</th>
<th>RVERS</th>
<th>RYEAR</th>
<th>RTCUR</th>
<th>RUNIT</th>
<th>LRCRK</th>
<th>POPER</th>
<th>DOCCT</th>
<th>DOCNR</th>
<th>DOCLN</th>
<th>RBUKRS</th>
<th>RPRCTR</th>
<th>RHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>4751</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>1</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4752</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>2</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4753</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>3</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4754</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td>1</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4755</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td>2</td>
<td>1000</td>
<td>1402</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Compressed table

<table>
<thead>
<tr>
<th>RCLNT</th>
<th>GL_SIRID</th>
<th>RLDNR</th>
<th>RRCTY</th>
<th>RVERS</th>
<th>RYEAR</th>
<th>RTCUR</th>
<th>RUNIT</th>
<th>LRCRK</th>
<th>POPER</th>
<th>DOCCT</th>
<th>DOCNR</th>
<th>DOCLN</th>
<th>RBUKRS</th>
<th>RPRCTR</th>
<th>RHR</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>4751</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>F'67t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4752</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>F'67t</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4753</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>21</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4754</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>800</td>
<td>4755</td>
<td>8A</td>
<td>0</td>
<td>0</td>
<td>1995</td>
<td>DEM</td>
<td>S</td>
<td>A</td>
<td>22</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Common sequences of consecutive bytes in row replaced with symbol**

```
1.4 DB2 Deep Compression Concepts

- Dictionary based - symbol table for compressing/decompressing data records
  - Lempel-Ziv (LZ) based algorithm (static dictionary) utilizing 12bit symbols
  - Dictionary per table stored within the permanent table object (~75KB in size; disk+memory)

- Data resides compressed on pages (both on-disk and in bufferpool)
  - Significant I/O bandwidth savings
  - Significant memory (bufferpool) savings
  - CPU costs
    - Rows must be decompressed before being processed for evaluation
1.5 Compression Dictionaries

- Stored as part of table data object
- Not accessible externally
- Size max. 150 KB (typical size ~75 KB)
- Compression Symbols are 12 bits in size
  - Each dictionary can have up to 4096 compression symbols
  - Column values and multi column values can be replaced by a single symbol
- DPF: created and stored on each database partition
- Data partitioning: created and stored on each data partition
- Compression dictionaries are created by offline table REORG or by the DB2 INSPECT command
1.6 DB2 Deep Compression Benefits and Considerations

**Benefits**
- Saves disk space (on tablespaces, log files and backup images) ➔ reduced TCO
- Minimizes I/O ➔ improved performance on I/O bound system
- Increases bufferpool hit ratio ➔ improved performance

**Considerations**
- Increased CPU workload for compression / decompression, especially for operations that access the table itself (e.g. table scans, non-sampled RUNSTATS, REORG)
- Database heap - Memory for compression dictionaries is allocated from database heap (on average ~75 KB per dictionary)
- Utility heap - Building compression dictionaries requires a temporary in-memory buffer of 10 MB that is allocated from the utilities heap
1.7 Limitations of Data Row Compression

- Row compression is only supported for tables supported by table REORG. The following tables cannot be compressed:
  - RCT tables (range-clustered tables)
  - Catalog tables
  - Declared global temporary tables
  - System temporary tables

- Row compression is only applicable to rows in data objects table space!! (the key to this is if it is stored with tables, or somewhere else)
  - Pure XML in separate object – XML data is not compressed (can be compressed in V9.5)
  - LOBs are stored in separate object – LOB data is not compressed
  - LONGs are stored in separate object - LONG data is not compressed
  - Indexes are not compressed

- Compression is not practical for a table smaller than 100KB, since the space savings may not offset the storage requirements for the dictionary.

- Row compression support and table data replication support will not be compatible.
  - DATA CAPTURE CHANGES option is not compatible with the COMPRESS YES option. If the compress row attribute is NO for the table but the table has a dictionary, the DATA CAPTURE CHANGES option is also not allowed since there may be rows that are compressed in the table.
1.8 How Do I Compress a Table?

In V9.1, it takes two steps to compress data in a table:

1) The table COMPRESS attribute must be set to YES
   - CREATE TABLE <table name> COMPRESS YES
   - ALTER TABLE <table name> COMPRESS YES

2) A Compression Dictionary must exist for the table object
   - REORG--<table name> … KEEPDICTIONARY|RESETDICTIONARY
   - INSPECT ROWCOMPESTIMATE-TABLE--+-NAME--<table-name>

Once both step 1) and step 2) are satisfied, all data subsequently populated into the table is subject to being compressed.
1.8.1 Enabling Compression with V9.1

1) CREATE ...
   COMPRESS YES

2) LOAD

3) OFFLINE REORG

Or, IMPORT, INSERT,...

Or, ALTER <existing table> ...
   COMPRESS YES
1.8.2 Automatic (Compression) Dictionary Creation (ADC) – V9.5

- **Row Compression supports now Automatic Compression Dictionary Creation (ADC)**
  - Table compress attribute must be set
  - Synchronous dictionary creation when table data reaches a certain threshold (appr. 1 MB system default)
  - Once the dictionary is created new data will be inserted in compressed format
  - Works regardless of growth mechanism
    - INSERT
    - IMPORT
    - LOAD
    - REDISTRIBUTE
  - No need to ‘worry’ about creating compression dictionary – occurs automatically when data threshold is reached
  - No lengthy offline reorg operation required to enable compression

- **Tradeoffs**
  - Compression ratio depends on the first one MB of data
  - Depending how representative the first MB of data is, the compression ratio less than optimal
  - Slight performance ‘hiccup’ incurred when threshold is crossed. The transaction which needs to create the dictionary must pay the cost of creating the dictionary.
1.8.3 Automatic (Compression) Dictionary Creation (ADC) – V9.5

1) CREATE
   ... COMPRRESS YES

2) LOAD

Once threshold is reached, dictionary built automatically, and subsequent rows compressed.

Internal threshold designed as best compromise b/w compression ratio, and dictionary build speed. Default ~1 MB.

Or, IMPORT, INSERT, ...
1.9 SAP Support for Deep Compression Overview

- Fully Integrated into DBACockpit (NW 7.0 SP12)

- Separate Compression Tool for
  - Monitoring table sizes
  - Estimating compression disk space savings
  - Enabling deep compression
  - Scheduling INSPECT or offline REORG

- SAP Note 980067 – Using DB2 9 Row Compression
  - Description and set of SQL scripts and tools for finding deep compression candidate tables and enabling compression

- SAP Note 905614 – R3load support for data row compression
1.10 NetWeaver 2007 Support – Compression Check
1.11 NetWeaver 2007 Support - Compression Enabling
1.12 Compression Tool in SAP Note 980067 (1)

- For use with 4.6C – NW2004s
- Tool attached to SAP Note – rowcompression.zip
- Install tool as an SAP transport
- Call compression tool program /ISIS/ZCOMP from transaction SE38
1.13 Compression Tool in SAP Note 980067 (2)

- The first step is to get a list of tables that are candidates for compression. The candidates are selected using the following criteria:
  - 20 largest tables (excluding LOBs) in the system which are:
    - Not compressed
    - Not volatile
    - Do not belong to a SAP cluster or pool table, or DB2 index only table
    - The row size is larger than the minimum row size for the tablespace page size
- Schedule the job
1.14 Compression Tool in SAP Note 980067 (3)

- Candidates identified and listed under the “Candidates” tab
1.15 Compression Tool in SAP Note 980067 (4)

- After tables have been compressed, they are listed under the “Compressed” tab
1.16 Deep Compression Support in SAP BW

- **Special support for SAP BW**
  - SAP Note 906765 - DB2 9 data row compression for SAP BW 3.x
    - SAP BW 3.0 SP 33, SAP BW 3.1 SP 27, SAP BW 3.5 (SAP NetWeaver ’04) SP 19
  - SAP Note 926919 – DB2 9 data row compression for SAP NetWeaver BI 2004s
    - SAP NetWeaver BI 2004s Support Package (SP) 9

- **Configuration parameter DB6_ROW_COMPRESSION**
  - When set to **YES**, PSA, DataStore and Fact tables are created with deep compression enabled. Use report SAP_RSADMIN_MAINTAIN to set the parameter
  - After one or more InfoPackages have been loaded into the tables an offline REORG or INSPECT is required to create the compression dictionary

- **Transaction RSRV offers 2 new tests for checking whether a compression dictionary exists for SAP NetWeaver BI tables that have deep compression enabled**
  - Check Deep Compression for InfoProvider tables in DB2
  - Check Deep Compression for PSA tables in DB2
1.17 BW Tables Deep Compression Check in Transaction RSRV

- Call transaction RSRV

- Select test ‘Check deep compression of database tables of InfoProviders in DB2/UDB’ or ‘Check deep compression of PSA database tables in DB2/UDB’
1.18 To Enable Deep Compression for BW Tables

Select offline REORG or INSPECT for tables chosen for compression
1.19 Customer 1 Experience

- **Compression test on a real customer BW database**
  a) Total db size (over all partitions): 3.356 TB
  b) Total used space (over all partitions): 2.873 TB
  c) Total free space (over all partitions): 480 GB
  d) Data Tablespaces with tables, which will be compressed:
      - Total size: 2.398 TB
      - Used: 2.029 TB
      - Free: 0.367 TB
  e) Index Tablespaces, which will not be compressed:
      - Total size: 0.781 TB
      - Used: 0.689 TB
      - Free: 0.091 TB
  g) Data and Index Tablespaces, which will be not compressed:
      - Total size: 0.177 TB
      - Used: 0.155 TB
      - Free: 0.022 TB

<table>
<thead>
<tr>
<th>Table name</th>
<th>% of original table</th>
<th>Number of rows</th>
<th>Sample size (GB)</th>
<th>Sample Compressed size (GB)</th>
<th>Compression ratio (%)</th>
<th>Total Estimated Disk Savings (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/BI0/F0PUR_C01</td>
<td>100</td>
<td>159259747</td>
<td>71.49</td>
<td>13.23</td>
<td>81.5%</td>
<td>58</td>
</tr>
<tr>
<td>/BIC/FZBSDOR02</td>
<td>10</td>
<td>77390826</td>
<td>21.48</td>
<td>5.23</td>
<td>75.7%</td>
<td>162</td>
</tr>
<tr>
<td>/BIC/AZOSDOR0200</td>
<td>100</td>
<td>191362309</td>
<td>141.46</td>
<td>36.18</td>
<td>74.4%</td>
<td>105</td>
</tr>
<tr>
<td>/BIC/AZOSDDBL0100</td>
<td>20</td>
<td>19474261</td>
<td>11.80</td>
<td>3.05</td>
<td>74.2%</td>
<td>43</td>
</tr>
<tr>
<td>/BIC/B0000326000</td>
<td>10</td>
<td>23370458</td>
<td>25.78</td>
<td>8.54</td>
<td>66.9%</td>
<td>172</td>
</tr>
<tr>
<td>/BIC/B0000285000</td>
<td>10</td>
<td>39573969</td>
<td>23.24</td>
<td>6.32</td>
<td>72.8%</td>
<td>169</td>
</tr>
</tbody>
</table>

- **Database size reduced from 2.873TB to 1.453TB (50%)**
1.20 Customer 2 Experience – INTER Versicherung (Germany)

- SAP R/3 4.7 with IS Insurance.

- DB size:
  - Overall DB size (used) uncompressed: 266 GB
  - Overall DB size (used) compressed: 151 GB (43 % savings)

  - Compressed size includes larger index sizes due to large RID conversion (for all the tables that were compressed and therefore reorganized)
  - SAP note #980067 used to identify tables for compression - 310 tables, shrinking from 132,5 GB to 39,7 GB (70 %)

- Performance behaviour:
  - During two weeks of production
    - Dialog response time for their 30 most important transactions went down from 285 ms average to 198 ms average (30% improvement).
    - Dialog response time for all transactions went down from 247 ms average to 183 ms average (26% improvement).
    - There was also an improvement for batch jobs.
  - They pay a slight CPU penalty. CPU user time went from 18,3 % average to 19,6 % average (+ 5%) ... they for sure never had a CPU bottleneck
  - No performance problems before, overall they ran a well balanced system
1.21 Customer 2 Experience – INTER Versicherung (Germany)

- Week 46 – database uncompressed
- Week 47 – database compressed

Through monitoring the SAP system performance for these two weeks, we can conclude:

- The overall response time had improved by over 23%, for both Dialog and Batch transaction. The main contributor was the reduced database response time.
- For Dialog transactions, bigger throughput was achieved, and with less CPU.
- For Batch transactions, CPU usage was up by 22.3%.

<table>
<thead>
<tr>
<th>Task Type</th>
<th>Calendar Week (CW)</th>
<th># steps</th>
<th>Avg. resp. [ms]</th>
<th>Avg. CPU [ms]</th>
<th>Avg. DB [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dialog</td>
<td>CW 46</td>
<td>361,036</td>
<td>248,8</td>
<td>99,8</td>
<td>79,4</td>
</tr>
<tr>
<td></td>
<td>CW 47</td>
<td>483,750</td>
<td>183,2</td>
<td>83,3</td>
<td>48,6</td>
</tr>
<tr>
<td>Batch</td>
<td>Diff.</td>
<td></td>
<td>63,6</td>
<td>16,3</td>
<td>29,8</td>
</tr>
<tr>
<td></td>
<td>% Diff.</td>
<td></td>
<td>(25,8%)</td>
<td>(16,4%)</td>
<td>(38,0%)</td>
</tr>
<tr>
<td></td>
<td>CW 46</td>
<td>44,850</td>
<td>10,135,9</td>
<td>2,722,8</td>
<td>3,076,9</td>
</tr>
<tr>
<td></td>
<td>CW 47</td>
<td>43,106</td>
<td>7,708,9</td>
<td>3,331,0</td>
<td>2,374,3</td>
</tr>
<tr>
<td></td>
<td>Diff.</td>
<td></td>
<td>2,427,0</td>
<td>-608,2</td>
<td>702,6</td>
</tr>
<tr>
<td></td>
<td>% Diff.</td>
<td></td>
<td>(23,9%)</td>
<td>(-22,3%)</td>
<td>(22,8%)</td>
</tr>
</tbody>
</table>

Table 3 - Overall Response Time Comparison (ST03N)
1.22 Customer 3 Experience - SAP BI PoC Unnamed customer

- PoC done on a customer’s 20TB database and workload
  - p595 and DS8300 storage
- Total DB reduced from 16TB to 8TB (used pages)
- Average Compression Rate
  - ODS 79% (26 tables)
  - PSA 64% (517 tables)
  - Fact 86% (100 tables)
- Average Query Times
  - Transactions/sec 1.39 to 1.44 (3.5%)
  - Response (sec) 10.6 to 7.12 (23%)
1.23 Backup Compression and Data Row Compression

- Test backup compression in addition to tables with row compression
- Backup compression can be expensive and may not provide much added value in additional savings to backup image size
  - Time/size/value depends on the percentage of table space content with row compression.
  - E.g. Are all tables compressed? Are indexes or long data stored in the same table space?

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total User Time (seconds)</th>
<th>Pages Used</th>
<th>Table Space Size (GB)</th>
<th>Backup Image Size (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No compression</td>
<td>468</td>
<td>1510400</td>
<td>11.57</td>
<td>12</td>
</tr>
<tr>
<td>Backup Compression Only</td>
<td>1028</td>
<td>1510400</td>
<td>11.57</td>
<td>4.2</td>
</tr>
<tr>
<td>Data Row Compression Only</td>
<td>198</td>
<td>610816</td>
<td>4.68</td>
<td>4.7</td>
</tr>
<tr>
<td>Data Row and Backup Compression</td>
<td>662</td>
<td>610816</td>
<td>4.68</td>
<td>4.2</td>
</tr>
</tbody>
</table>
THANK YOU FOR YOUR ATTENTION!