# Djupkomprimering på Folksam

Lennart Henäng, Folksam

Folksam®

1

# Agenda

- What is Deep Compression?
- What tools are there?
- Experiences with tools run on real data

Folksam®

## What is Deep Compression?

- Remember DB2 Version 3 1993?
- DB2 9 ”Venom” technology
  - dictionary based algorithm
    - dictionary is typically 75 KB, max 150 KB
  - assigns short numeric keys to repetitive entries
    - based on entire rows, not just particular fields
    - based on entire table(partition), not just on each page
- Row compression is not compatible with replication
- Row compression statistics gathered by RUNSTATS
- Decompression included in optimizer costing model

Folksam®

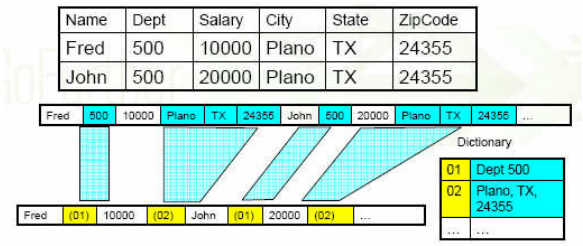
3

**Venom** or **zootoxin** (literally, poison of animal origin) is any of a variety of toxins used by several groups of animal species, for the purpose of defense and hunting prey. Though subjective, the definition of a venom differs from a poison (as used in most zoology and medicine texts). Generally, a venom is defined as a biologic toxin that is injected to cause its effect while a poison is a biologic toxin which is absorbed through epithelial linings (either of the gut or through the skin).

The animals most widely known to use venom are snakes, some species of which inject venom into their prey through hollow fangs; spiders and centipedes, which also inject venom through fangs; scorpions and stinging insects, which inject venom with a sting (which is a modified egg-laying device - the ovipositor). It is important to note the difference between organisms that are "venomous" and "poisonous", two commonly confused terms with regards to plant and animal life. Venomous, as stated above, refers to animals that inject venom into their prey or as a self-defence mechanism. Poisonous, on the other hand, describes plants or animals that are harmful when consumed or touched.

*From Wikipedia, the free encyclopedia*

## Deep Compression Example




Deep compression:

Finds repeating values using complete row

Single dictionary per table/DPF-partition

Folksam®

4

Compression only happens if all of the following are true

1.Table has COMPRESS YES

2.A dictonary exists for the table

3.Space will be saved compared to an uncompressed row

There are two ways to insert a dictonary into a table.

1. To run a classic (offline) REORG when COMPRESS YES
2. To run INSPECT ROWCOMPESTIMATE

When COMPRESS NO, data is still compressed but no rows will be compressed hereafter neither at INSERTs or UPDATEs

There are two ways to delete a dictonary from a table after setting COMPRESS NO

1.Run a classic (offline) REORG RESETDICTIONARY

2.Table reinitialization or truncation (such as, for example, a replace operation)

## What tools are there?

- Estimation tool included in DB2 9
  - db2 inspect rowcompestimate
  - db2inspf
- DB2 Compression Estimate
  - can be used on DB2 V8 as well as DB2 9
- CREATE/ALTER TABLE x COMPRESS YES/NO
- REORG TABLE x (classic/offline reorg)
  - KEEPDICTIONARY (default)
  - RESETDICTIONARY

Folksam®

5

The dictionary is logged in its entirety for recovery purposes and for future support with data capture changes (i.e. replication).

**KEEPDICTIONARY**

If the COMPRESS attribute for the table is YES and the table has a compression dictionary then no new dictionary is built. All the rows processed during reorganization are subject to compression using the existing dictionary. If the COMPRESS attribute for the table is NO and the table has a compression dictionary then reorg processing will preserve the dictionary and all the rows in the newly reorganized table will be in non-compressed format. It is not possible to compress long, LOB, index, or XML objects.

**RESETDICTIONARY**

If the COMPRESS attribute for the table is YES then a new row compression dictionary is built. All the rows processed during reorganization are subject to compression using this new dictionary. This dictionary replaces any previous dictionary. If the COMPRESS attribute for the table is NO and the table does have an existing compression dictionary then reorg processing will remove the dictionary and all rows in the newly reorganized table will be in non-compressed format. It is not possible to compress long, LOB, index, or XML objects.

## Row Compression Estimate

INSPECT ROWCOMPESTIMATE TABLE NAME x SCHEMA y RESULTS KEEP file

- Creates a dictionary in all or specified partitions and produces a result file into the db2dump directory

db2inspf data-file out-file

- Formats the result file into human readable format
  - Percentage of bytes saved from compression
  - Percentage of pages saved from compression
  - Percentage rows ineligible for compression due to small data size
  - Compression dictionary size
  - Expansion dictionary size

Folksam®

## Real Data Example - DKU

- Table QDPG70.QTG70DKU (“Customer Dimension")
  - more than 11 million rows
  - 115,984 32K pages
  - about 3,62 GB of data
- Many character columns with fixed length

Folksam®

## Real Data Example - DKU

- Moved 387,403 rows to a DB2 9 database
  - 32,300 pages
  - rowsize 330 bytes

```
>db2 inspect rowcompestimate table name
QTG70DKU schema QDPG70 results keep
qtg70dku.comp.txt

>db2inspf qtg70dku.compr.txt qtg70.compr.out

>vi qtg70.compr.out
```

Folksam®

# Real Data Example - DKU

```
DATABASE: COMPTEST
VERSION : SQL09010
2006-11-06-14.20.42.154000


Action: ROWCOMPESTIMATE TABLE
Schema name: QDPG70
Table name: QTG70DKU
Tablespace ID: 2  Object ID: 4
Result file name: qtg70dku.compr.txt

    Table phase start (ID Signed: 4, Unsigned: 4; Tablespace ID: 2) :
QDPG70.QTG70DKU

      Data phase start. Object: 4  Tablespace: 2
      Row compression estimate results:
      Percentage of pages saved from compression: 77
      Percentage of bytes saved from compression: 77
      Percentage of rows ineligible for compression due to small row size: 0
      Compression dictionary size: 46592 bytes.
      Expansion dictionary size: 32768 bytes.
      Data phase end.
    Table phase end.
Processing has completed. 2006-11-06-14.21.34.420000
```

77%

Σ 79360 bytes

Folksam®

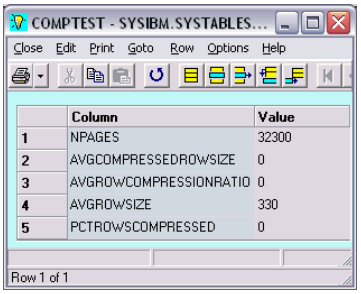
# Establish Compression

```
>db2 ALTER TABLE QDPG70.QTG70DKU COMPRESS YES

>db2 REORG TABLE QDPG70.QTG70DKU RESETDICTIONARY

>db2 select dictionary_size from
table(sysproc.admin_get_tab_info('QDPG70','QTG70DKU'))
as t

79360

>db2 call admin_cmd('runstats on table QDPG70.QTG70DKU')
```

Folksam®

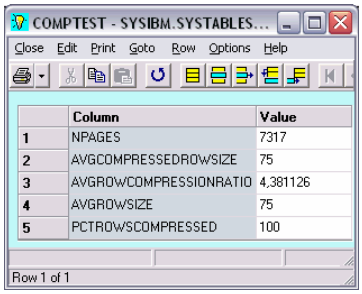
## Compare Reality Against Estimate

Before




| | Column | Value |
|---|---|---|
| 1 | NPAGES | 32300 |
| 2 | AVGCOMPRESSEDROWSIZE | 0 |
| 3 | AVGROWCOMPRESSIONRATIO | 0 |
| 4 | AVGROWSIZE | 330 |
| 5 | PCTROWSCOMPRESSED | 0 |

After




| | Column | Value |
|---|---|---|
| 1 | NPAGES | 7317 |
| 2 | AVGCOMPRESSEDROWSIZE | 75 |
| 3 | AVGROWCOMPRESSIONRATIO | 4,381126 |
| 4 | AVGROWSIZE | 75 |
| 5 | PCTROWSCOMPRESSED | 100 |

Page level: 7317/32300 = 0.2265 => 77,35%

Row level: 1/4.381126 = 0.2283 => 77,17%

Row level: 75/330 = 0.2272 => 77,28%

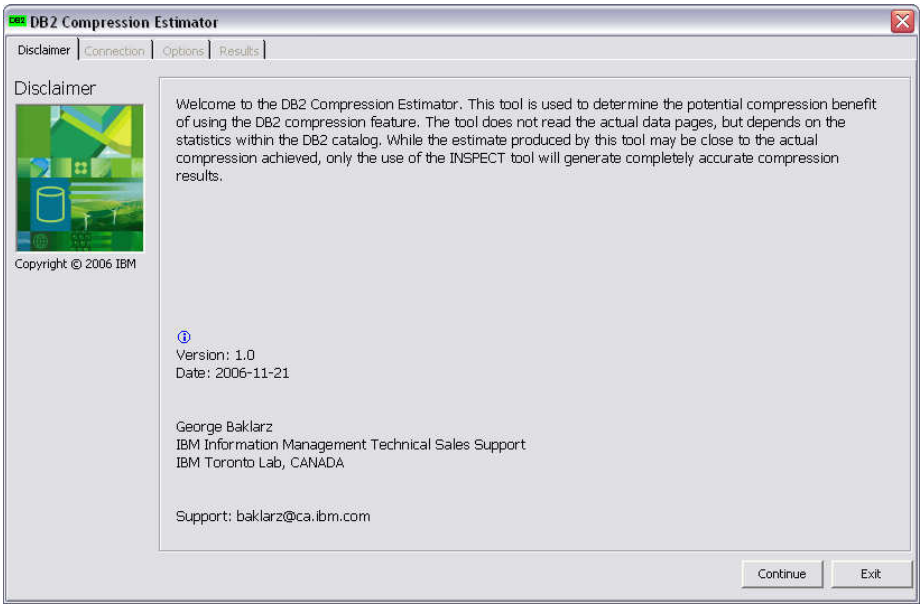
23% of 3,62 GB = 0,83 GB

Folksam®

11

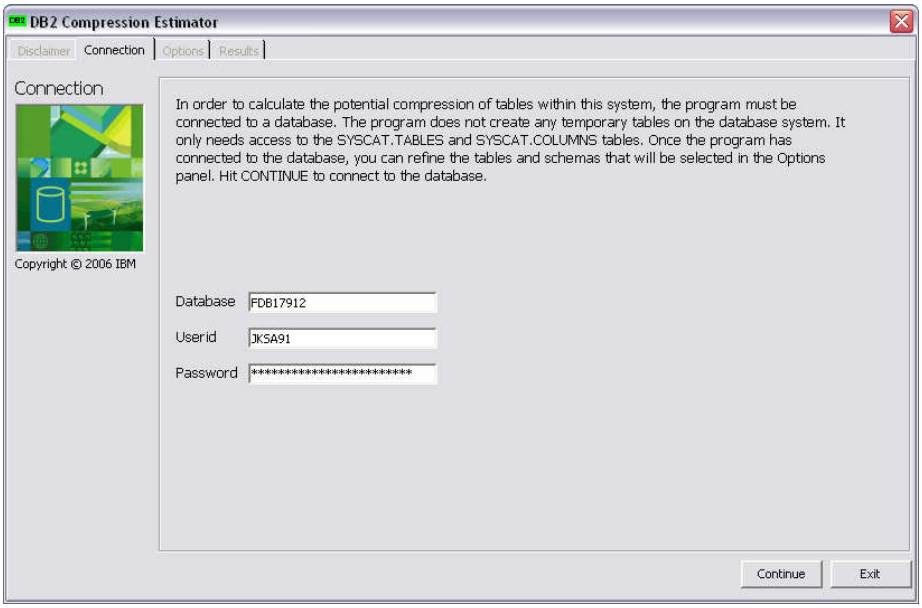
```
select npages,
        avgcompressedrowsize,
        avgrowcompressionratio,
        avgrowsize,
        pctrowscompressed
from   sysibm.systables
        where name='QTG70DKU'
```

# DB2 Compression Estimator



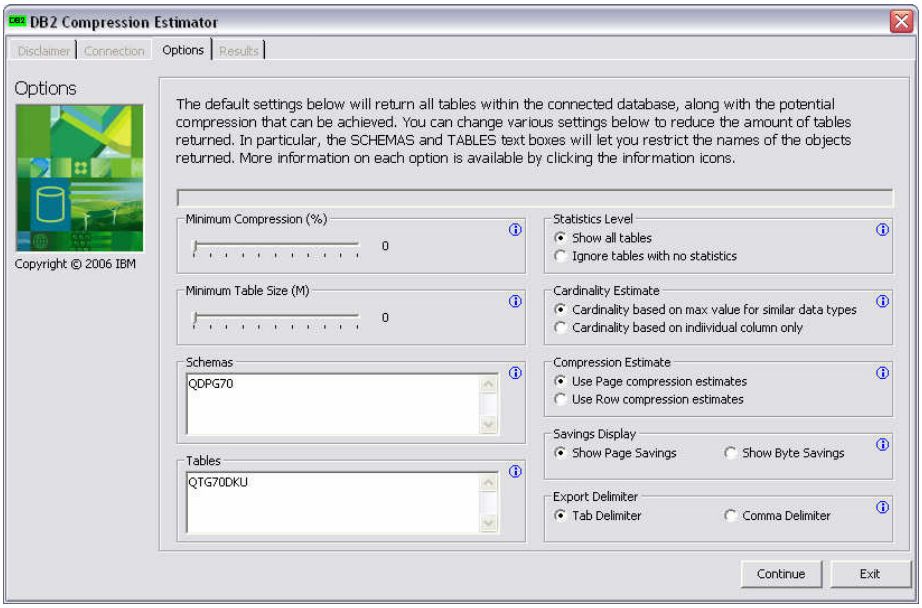

Folksam®

# Connection



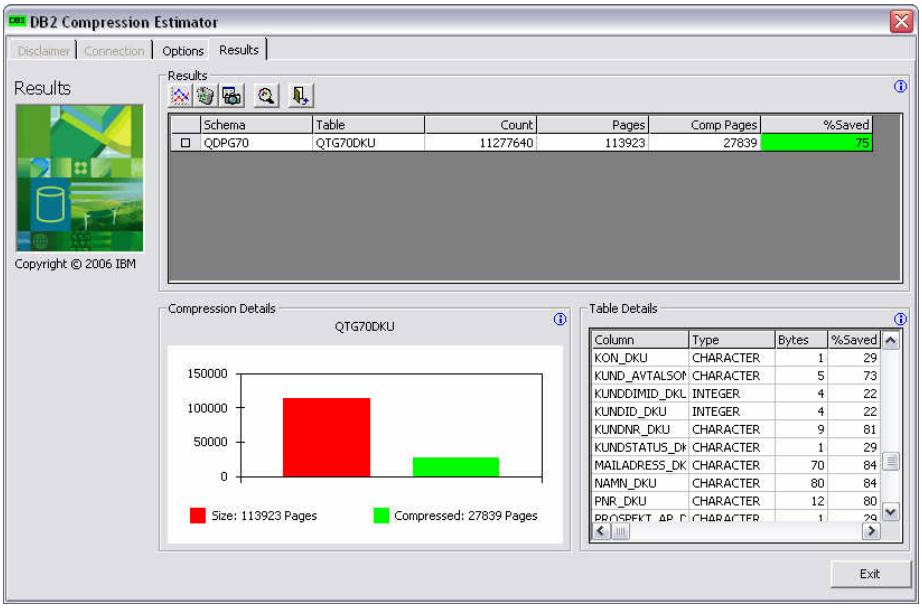

Folksam®

# Options




Folksam®

# Results



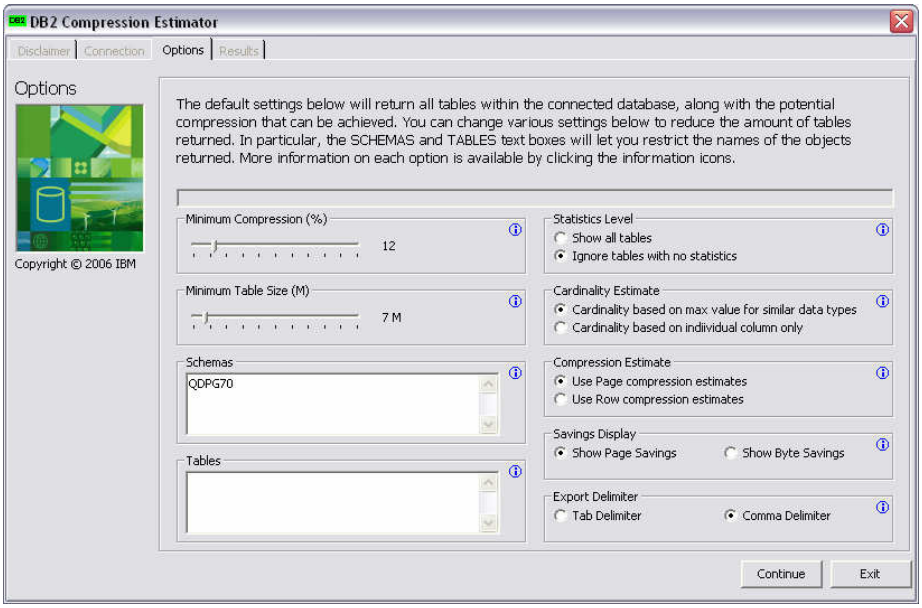

Folksam®

# Comparison

| Compression Estimation Tool | INSPECT ROWCOMPESTIMATE | Real Life |
|---|---|---|
| 75-76% | 77% | 77,35% |

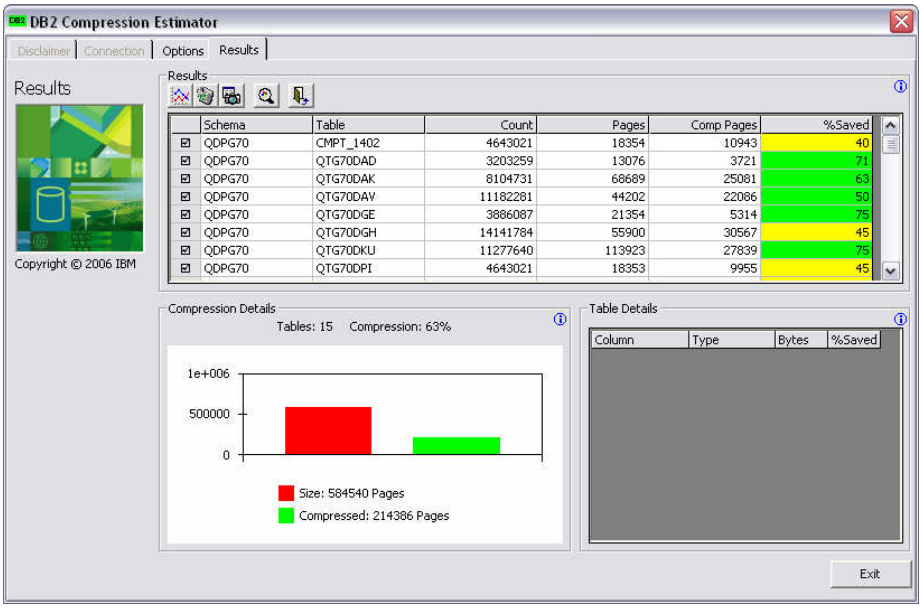
Folksam®

# Strength of Compression E Tool




Folksam®

# Results




| | Schema | Table | Count | Pages | Comp Pages | %Saved |
|---|---|---|---|---|---|---|
| ☑ | QDPG70 | CMPT_1402 | 4643021 | 18354 | 10943 | 40 |
| ☑ | QDPG70 | QTG70DAD | 3203259 | 13076 | 3721 | 71 |
| ☑ | QDPG70 | QTG70DAK | 8104731 | 68689 | 25081 | 63 |
| ☑ | QDPG70 | QTG70DAV | 11182281 | 44202 | 22086 | 50 |
| ☑ | QDPG70 | QTG70DGE | 3886087 | 21354 | 5314 | 75 |
| ☑ | QDPG70 | QTG70DGH | 14141784 | 55900 | 30567 | 45 |
| ☑ | QDPG70 | QTG70DKU | 11277640 | 113923 | 27839 | 75 |
| ☑ | QDPG70 | QTG70DPI | 4643021 | 18353 | 9955 | 45 |

Folksam®

# Exported Data (Tab Delimited)

| Schema | Table | Count | Pages | Comp Pages | %Saved |
|---|---|---|---|---|---|
| QDPG70 | CMPT_1402 | 4643021 | 18354 | 10943 | 40 |
| QDPG70 | QTG70DAD | 3203259 | 13076 | 3721 | 71 |
| QDPG70 | QTG70DAK | 8104731 | 68689 | 25081 | 63 |
| QDPG70 | QTG70DAV | 11182281 | 44202 | 22086 | 50 |
| QDPG70 | QTG70DGE | 3886087 | 21354 | 5314 | 75 |
| QDPG70 | QTG70DGH | 14141784 | 55900 | 30567 | 45 |
| QDPG70 | QTG70DKU | 11277640 | 113923 | 27839 | 75 |
| QDPG70 | QTG70DPI | 4643021 | 18353 | 9955 | 45 |
| QDPG70 | QTG70DPI_H | 3411670 | 13486 | 7342 | 45 |
| QDPG70 | QTG70DPN | 4684504 | 65068 | 20660 | 68 |
| QDPG70 | QTG70FAM | 8233667 | 32547 | 16876 | 48 |
| QDPG70 | QTG70FCL | 302284 | 3699 | 820 | 77 |
| QDPG70 | QTG70FRA | 2840732 | 14874 | 7850 | 47 |
| QDPG70 | QTG70FRE | 12178173 | 99016 | 24827 | 74 |
| QDPG70 | QTG70PCL | 490163 | 1999 | 505 | 74 |

Folksam®

## ROI Calculator - output

- Excel spreadsheet to calculate economical benefit
- Calculates storage cost saved
  - Conservative - based on raw data
  - Aggressive - based on total database (4 x raw)
- Calculates additional on going cost savings
  - Less time for restore (shorter outage)
  - Less DBA time for backup/reorg/runstats
  - Less disk space for backups

Folksam®

## ROI Calculator - input

- Data from CET imported to spreadsheet
- Customer specific cost factors as
  - Investments (Software License & Maintenance)
  - Yearly Storage Cost per Terabyte
  - Y2Y Data Growth %
  - Y2Y Storage Cost Increase %
  - Cost of Backup Media
  - Cost of DBA Time

Folksam®

# ROI Calculator

Microsoft Excel - Folksam_DW_DB2V9_Storage_ROI_calculator.xls

Arkiv Redigera Visa Infoga Format Verktyg Data Fönster Hjälp

B7 = =13500*(B14/0,25)

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | DB2 Storage Optimization / Return on Investment Calculator | | | | | | | |
| 2 | ROI Summary: | | | | | | | |
| 3 | | | | | | | | |
| 4 | Input Cell | | | | | | | |
| 5 | Input Cell | | | | | | | |
| 6 | | Year 1 | Year 2 | Year 3 | Total | Three Year Net Savings | | |
| 7 | Investments(Software License & Maintenance) | 108000 | 21600 | 21600 | 151200 | | | Assume 250GB per CPU |
| 8 | Cost avoided/saved - Conservativ | 832395 | 864126 | 908880 | 2605201,2 | 2 454 001,23 | | |
| 9 | Cost avoided/saved - Aggressive | 1089575 | 1214102 | 1389917 | 3693593,8 | 3 542 393,76 | | |
| 10 | | | | | | | | |
| 11 | Storage Cost Avoidance Details: | | | | | | | |
| 12 | | Year 1 | Year 2 | Year 3 | | Y2Y Data Growth % | | |
| 13 | Raw data size in TB's: | | | | | | | |
| 14 | - Production | 2 | 2,6 | 3,38 | | 30 | | |
| 15 | - Development | 0,5 | 0,5 | 0,5 | | | | Assume Dev and Test do not grow at |
| 16 | - Test/QA | 0,5 | 0,5 | 0,5 | | | | the same rate as Prod and DR |
| 17 | - DR | 2 | 2,6 | 3,38 | | | | |
| 18 | Total Disk for Raw Data | 5 | 6,2 | 7,76 | | | | |
| 19 | Total Disk for entire database(s) | 20 | 24,8 | 31,04 | | Y2Y storage cost increase % | | |
| 20 | Today storage costs per TB (including Disk, controllers, power, cooling, SAN Administrator, Floor Space, etc.) | 33600 | 36360 | 40656 | | 10 | | |
| 21 | Other Storage Costs | | | | | | | |
| 22 | | | | | | | | |
| 23 | Raw Data Storage Costs | 168000 | 223152 | 315430,56 | | | | |
| 24 | Total Database Storage Costs | 672000 | 316608 | 1261962,2 | | | | |
| 25 | | | | | | | | |
| 26 | Compression | 51% | 51% | 51% | | | | |
| 27 | Storage Cost Saved Conservative | 85526,7 | 116658 | 160612 | | | | |
| 28 | Storage Cost Saved Agressive | 342107 | 466634 | 642449 | | | | |
| 29 | | | | | | | | |
| 30 | Additional On Going Cost Savings: | | | | | | | |
| 31 | | | | | | | | |
| 32 | | | | Yearly Total Aditional Savings | | | 747468 | |

DB2 Storage Opt. - ROI / Import [

Klar

Folksam®

## Summary

- Accurate tools for compression estimation
  - single table
  - multiple tables
- Compression can be significant
  - up to 77% for a specific table
  - up to 64% for a specific schema (QDPG70 – 15 tables)
- ROI tool needed since compression is a feature
  - i.e, it costs a lot of money

Folksam®

23

QDPG70 – tables over 5 MB