

Introducing DB-123
a new approach to
Database- and IT-migrations

Thomas Schneider (www.db-123.com)

Rexx LA meeting

IBM Laboratories, Boeblingen

May 2004

IT-migrations performed between 1999 and 2004

- 1999 – 2000 Year 2000 Conversion (sliding Window technique, COBOL Preprocessor DO*IT 1999)
- 2001-2002 EURO Conversion (the EURO*Transformer Tool-set)
- 2004 IBM VSE COBOL → Micro-Focus Cobol with concurrent migration from IBM DL/I to IBM DB2

DB-123 Design Goals

- Upwards compatible database management
- Minimize costs for migrations
- Maximize security for users
- Do not loose already existing meta- in formation (hierarchies, repetitions)

the Donauland data-base: a practical example

29	DL/I Data Bases
75	DL/I Segments
10	permanent VSAM KSDS Files
6	permanent VSAM ESDS Files
10	SQL-Tables

which implies:

55	Download-Programs necessary
95	DB2-LOAD-Files !!!

DL/I --> DB2 Migration

which implies:

*75 new DB2-Tables necessary only for DL/I →
DB2 Migration (excluding VSAM-Files)*

*all 75 NEW DB2-Tables have to be downloaded
synchronized from the HOST and uploaded to
the new system at once!*

DB-123: the DL/I --> DB2 Migration-Tool

- performs the IBM DB2-database DESIGN automatically (from IBM DL/I)
- manual adaption of DESIGN possible
- automatic addition of new fields possible (e.g. USER-ID and UPDATE-Date) in all new tables

DB-123: the DL/I --> DB2 Migration-Tool(2)

- generates DOWNLOAD Programs (COBOL SOURCE and VSE JCL!) for unloading the DL/I databases
- with rigorous checks of all values
 - check NUMERIC fields for valid entries.
 - Check all DATE-fields for valid Dates
 - 000000 --> SQL NULL
 - 999999 --> SQL NULL
 - Character-Fields: no imbedded quotes allowed

DB-123: the DL/I --> DB2 Migration-Tool(3)

- generates COBOL HOST-Variables for SQL-interface
- automatic Update of the ,old‘ DL/I record COPY-books (add parent keys, USER-ID, LAST-UPDATE)

DB-123: the DLI --> DB2 Migration-Tool (4)

- generates COBOL procedures (inline code) for the various access methods: .
 - xxxGET - for keyed access (unique key given)
 - xxxNEXT - for SQL Cursor access (alle seq.)
 - xxxNWP - ,next within *logical* parent‘
 - xxxISRT - for INSERTS
 - xxxDELE - for DELETES (with given key)
 - xxxUPDT - for Updates

DB-123 SQLACC Generator: the advantages

- for 95 DB2-Tables in the Donauland example
- We did need $95 * 6 = 570$ subroutines/procedures for SQL-Access
- Using DB-123, those access procedures need *NOT* to be developed manually and tested, but are generated automatically.

DB-123 DESIGN: The DB2- Data-Base Design

- for each *DL/I Segment* and each *VSAM KSDS-File* a *DB2-Table* has to be created
- following the rules of the ,third normal form‘
- These rules of the third normal form are very simple:

DB-123 DESIGN: The first normal form

- All parent *DL/I* primary keys must be added to the child records (in front of the existing fields)
- All unique *DL/I primary keys* become primary keys of the DB2-table as well.

DB-123 DESIGN: the second normal form

- If the key is still not unique, a sequence field (trigger) is added
- *or a so called **TIMESTAMP***
- Each DB2 table has now a unique key anyway!

Is the DESIGN now complete?

- . NO, as the DL/I databases may contain OCCURS-items in the COBOL record structures!
 - *OCCURS-items are NOT supported by DB2!*

Is the DESIGN now complete?

- DB123 resolution:
- either new, higher level TABLES (DB2DETAIL) are added,
- or NUMBERING of the respective DB2 columns
- both Alternatives are performed automatically, driven by proper clauses in the Data Dictionary

Is the DB2-DESIGN now complete?

- Yes !!
- New Data Dictionary used to GENERATE ACCESS functions and needed COPY-books.

DB-123: The third normal-form (no repeated items)

- All Tables are flat tables (including logical parent keys)
- Hierarchic relationship retained thru ,Logical Parent' relationships (LPARENT-clause) in data Dictionary.
- Repeated COBOL structures resolved thru DBDETAIL clause in Data Dictionary.

DB-123 DESIGN: NEXT Necessary Steps

- New Data Dictionary used to GENERATE
- SQL CREATE TABLE definition files
- Proper SQL \Leftrightarrow COBOL name&type mappings
- COBOL HOST Variable COPY books
- ACCESS functions and needed COPY-books.

DB-123 DESIGN: NEXT Necessary Steps

- Done automatically again
- By DB123 CREATE TABLE <dictionary>
- And DB123 HOSTVAR <dictionary>
- And DB123 SQLACC <dictionary>
- Generates ALL the necessary COPY-books for declaring the necessary auxiliary variables and COBOL DB2 Access-Procedures