MIGRATION OF DATA FOR IKNOW APPLICATION AT EURM – A CASE STUDY

Marko Vuckovik, Toni Stojanovski

Introduction

- Applications for employing business rules and managing the data
  - With specifically designed database
- Application replacement with newer application
  - Many reasons for this requirement
- EURM is now replacing the existing custom-made application with the iKnow application designed for the iKnow project
- The data from the old application needs to be migrated to the new application database
- Pilot phase
- Purpose of the paper

Presentation outline

- Slide 4 – Background
- Slide 5 – ETL Process
- Slide 7 to slide 11 – Encountered problems and solutions
- Slide 12 – Increasing the scalability and performance
- Slide 13 – Conclusion

Background

- There are issues related to the EURM application
- The iKnow application is expected to resolve these issues
- The EURM application uses three databases for every available study cycle
- Both application databases are created in MSSQL server
- The data migration from the old EURM application database to the new iKnow application database is a typical example of ETL process

Extract, Transform and Load process

- The ETL process consists of these three steps
  - Extract data from legacy database
  - Transform the data to fit the data schema of the new database
  - Load the data into the new target database
- Extraction parses the extracted data, checks if the data meets an expected pattern or structure and converts the data into a single format appropriate for the transform stage
- At transform stage prepares the data for loading into the target database using series of rules or functions
- The load phase loads the data into the target database

Our ETL process

- Understanding the two database designs
- Identification of related tables and fields
- Preserving the relations between the records from different tables
- Occurred problems and their solutions are described on the following slides

Problem 1

- *Problem*: The foreign key constrain, which enforces referential integrity, does not allow us to insert a value for a foreign key field if the foreign key table does not have a record with the corresponding primary key.
- *Solution*: Source tables acquire priority number. The tables with the highest priorities are the first to be populated. The priorities are determined with this rule: Tables that do not have foreign key fields have the highest priority. If table has one or more foreign key fields, then the priority of that table is lower than the priority of its foreign key tables.

Problem 2

- *Problem*: Several tables from the iKnow database do not have corresponding tables in the old EURM databases.
- *Solution*: The information for these tables can be found in other table fields. Distinct data contained in free-text fields are extracted (using "*SELECT DISTINCT*") from the three EURM databases. The target table is populated with this record set. Every record now gets its own ID number which is used as foreign key.

Problem 3

- *Problem*: There are situations when there is a need for keeping the existing IDs for specific table records. Since there are three SAA databases with exactly the same schema, existing IDs could have been kept only if the three source tables from the three EURM databases contained the same records.
- *Solution*: Example is the *Faculty* table. The stored faculties are replicated for every EURM database. We can keep the existing IDs by temporary turning of the auto increment operation on the ID field for the target table using the "*SET IDENTITY_INSERT <table_name> ON/ OFF"* operation.

Problem 4

- *Problem*: How to distinguish records that are coming from the same table, but from different EURM database? How to divide these records into different study cycles?
- *Solution*: iKnow database has table called *StudyCycles* that is used to distinguish the different study cycles, unlike the EURM application that uses different databases per study cycle.

Problem 5

- *Problem*: Certain tables (*Students, Programmes, Courses*) exist in the three EURM databases but contain different records. Records with the same ID exist in the same table in different EURM database, but contain data for different entity e.g. student from different study cycle
- *Solution*: We use temporary mapping tables for storing the mapping between ID fields in related tables from the old and the new database. These tables have three integer fields (*NewKeys, OldKeys, DatabaseID).* The *NewKeys* field stores the new ID generated in the target table, and the *OldKeys* field stores the old ID in correlation with the *DatabaseID* value (the source database number)

Increasing the scalability and performance

- iKnow application uses ASP.NET Membership Provider
- In previous studies we identified several improvements that can be applied to the membership provider.
- Our research analyses different SQL server-based paging technique, and identifies a design for the SQL stored procedures that can improve the response time, scalability and the performance
- iKnow application uses the default SQL stored procedures from the ASP.NET Membership Provider
- Our improved procedures can increase the iKnow application scalability and performance

Conclusion

- Data migration is essential
- In the data migration we applied solutions typical for the ETL process
- The main purpose of the data migration is ensuring overall quality
- The solutions for the problems described in this paper can provide guidelines for other data migrations
- This paper can help other educational institutions that are partners of the Tempus project to migrate their own data to the iKnow application

Thank you for your attention