DATABASE REVERSE ENGINEERING

DBTech_EXT Workshop in Thessaloniki
2009-09-10

Kari Silpiö
HAAGA-HELIA University of Applied Sciences
Database Reverse Engineering

OUTLINE

✓ What is Database Reverse Engineering?
✓ Why / where Database Reverse Engineering is needed?
✓ What are the main steps in the Database Reverse Engineering process?
✓ What are the main Reverse Engineering features in Database Design tools?
✓ How Reverse Engineering can be utilized in a process where a database structure is ported from one DBMS to another?

Note: The focus of this presentation and lab is on relational databases only. Multidimensional databases and other types of databases (ODBMS etc.) are not covered.
Why / Where?

E.g., in the industry Database Reverse Engineering is being used for

- **Creating documentation** on existing database structures
  - To create a representation of the database structure at a higher level of abstraction (visualizing the existing DB structure, re-drawing the DB structure after changes, ...)
  - To reduce manual work end errors, increase productivity
  - To generate very detailed documentation of the DB structure
  - Database built in-house, but lacking up-to-date documentation
  - Database in a system that is purchased from elsewhere with no documentation for further application development

- **Porting** existing database structures between different DBMS environments (e.g., DB2 → SQL Server)

- **Integration:** Reverse Engineering from Repository (general data dictionary)
Database Reverse Engineering Process

Bottom-up Modeling

=> Build a database design based on either one of the following:
   • By importing metadata directly from an existing database
   • By importing a DDL script that reflects an existing database implementation

1. Reverse Engineer from a database or DDL script
   => The resulting database is represented as a Relational Schema and definitions for Physical & Relational Schema objects

2. Reverse Engineer from the Relational Schema to a higher-level schema
   => The resulting schema is represented as an ER Diagram (or Class Diagram) and definitions for ER model objects
"Conceptual Schema"
- Created in the Conceptual Data Modeling stage
- Typically consists of a Data Dictionary and an ER diagram (or UML Class Diagram) with entities, attributes, relationships, etc. Different diagramming conventions exist in DB Design tools.
- Create manually in the tool

"Relational Schema" (in Database Design Tools)
- Tables, columns, constraints, ...
- Generate (1st version of) the Relational Schema in either of the following ways:
  - Manually in the tool
  - **Forward engineering** from the "Conceptual Schema"
  - **Reverse Engineering**
    - By importing metadata directly from an existing database
    - By importing a DDL script
    - By importing metadata from another modeling tool / repository
Modeling Levels in Common DB Design Tools

"Physical Schema" (in Database Design Tools)

- Add target DBMS-specific definitions for storage structures, data security, triggers etc.
- Relational objects will be generated based on the Relational Schema and definitions for relational objects.
- A typical tool allows several DBMS-specific Physical Models/Schemas to be defined for a single relational schema.
Some Limitations

- Methodology and notation limitations
- Latest DBMS versions may not be supported
- Level of support for different DBMS's may vary a lot in the tool
- Latest SQL development may be missing
- ...

IBM Rational Rose Data Modeler 7.0.0
Oracle SQL Developer Data Modeler 1.5.1
IBM InfoSphere Data Architect 7.5.1
Terminology Issues

- The language for data modeling?
  - Academic vs. Practitioners
  - Vendors, DBMS-specific?
  - Methodologies vs. Data modeling tools

- Stages & models in database design?
  - Number and names of stages and models
  - Definition of stages (activities per stage) and models

Example: Terminology used in some data modeling tools

<table>
<thead>
<tr>
<th>Level</th>
<th>Oracle Data Modeler</th>
<th>Rational Rose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;Logical Model&quot;</td>
<td>&quot;Analysis&quot;</td>
</tr>
<tr>
<td></td>
<td>User’s guide: &quot;ER Model&quot;</td>
<td>(&quot;logical&quot;)</td>
</tr>
<tr>
<td>2</td>
<td>&quot;Relational Model&quot;</td>
<td>&quot;Design&quot;</td>
</tr>
<tr>
<td>3</td>
<td>&quot;Physical Model&quot;</td>
<td>&quot;Physical Data Model&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>&quot;Logical Data Model&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Physical Data Model&quot;</td>
</tr>
</tbody>
</table>
Oracle SQL Developer Data Modeler

Version 1.5.1 (free) released in 2009. Current version: 2.0 (priced product)

- Standalone single-user product with data and database modeling tools
  - Modeling for Entity-Relationship Diagrams
    - Barker (crow's feet) or Bachman notation, Supertypes & subtypes
  - Relational (database design)
  - RDBMS-specific "physical models"

- Forward and Reverse Engineering and DDL code generation
  - E.g., imports from and exports to DB2, Oracle, and SQL Server
  - Promises to create, compare and synchronize changes
  - Import from Oracle Designer repository

- Other
  - Data Type modeling, Multi-dimensional modeling, Data Flow diagrams, ...
  - Saves model definitions locally as XML files

For more details, see the product home page...
USING ORACLE DATA MODELER

1. **Reverse Engineer**
   - Import database structure directly from an existing DB2 database
   - Generate diagrams for visualizing the database structure
   - Create a subview for viewing a part of the DB structure at one time

2. Modify the design

3. **Forward engineer**
   - Create the SQL Server specific DDL script

4. Realize the design
   - Create the database structure in SQL Server by running the DDL script

![Diagram showing the process from DB2 to SQL Server with Oracle Data Modeler in the middle.]
## LAB ENVIRONMENT

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Data Modeling Tool</td>
<td>1.5.1</td>
</tr>
<tr>
<td>IBM DB2</td>
<td>Express-C 9.5</td>
</tr>
<tr>
<td>IBM DB2 Control Center</td>
<td></td>
</tr>
<tr>
<td>SQL Server</td>
<td>Express 2008</td>
</tr>
<tr>
<td>SQL Server Management Studio</td>
<td></td>
</tr>
<tr>
<td>WMVare Player</td>
<td>2.0.5</td>
</tr>
</tbody>
</table>
References

Oracle SQL Developer Data Modeler
www.oracle.com/technology/products/database/datamodeler

- Getting started with Oracle Data Modeler
download.oracle.com/docs/cd/E15276_01/doc.20/e13677/data_modeling.htm

Some other products
- ER/Studio
  www.embarcadero.com/products/er-studio
- ERwin
- Power Designer
  www.sybase.com/products/modelingdevelopment/powerdesigner
- Toad
  www.quest.com/toad-data-modeler

A study on data modeling concepts and terminology