Overview of Data Modeling Using IBM Graphical Tools

Dan Cruikshank – dcrank@us.ibm.com
Agenda

- Data Modeling Concepts
- Types of Graphical Data Models
- Creating a Data Model
- Reverse Engineering
What Is Data Modeling?

- Data modeling is a **method** used to define and analyze data requirements needed to support the business processes of an organization.

- Data modeling is used to **communicate** the business rules and processes.

- Data modeling is the **process** of creating a blueprint to visually represent data, its organization and the relationships between structures.

More information:
Data Models in Practice

- **Relational Model**
  - Used in on-line transaction processing (OLTP)
  - Data is highly normalized to avoid redundancy
  - Supports “transactional” business processes

- **Dimensional Model**
  - Used in on-line analytical processing (OLAP)
  - Specialized structures and relationships
    - Star schema, Snowflake schema
    - Fact tables
    - Dimension tables
    - Lightly summarized, highly normalized
    - Highly summarized, lightly normalized
  - Supports “business intelligence” activities
    - Multidimensional queries
Data Model Versus Data Flow

Data Flow
- Shows how data is handled
  - Who is processing the data
  - Where the data is stored
  - What happens to the data

Data Model
- Depicts relationships between the data
  - Independent of data flow

*This session is focused here*
Types of Data Models

- **Conceptual Data Model**
  - Typically developed first

- **Logical Data Model**
  - Typically developed second

- **Physical Data Model**
  - Typically developed third

- **Data Dictionary Models**
  - **Domain Model**
    - Collection of named data types for consistency and reuse
  - **Glossary Model**
    - Collection of column names and abbreviations
  - The above models are created in conjunction with the first three
Conceptual Data Model

- Describes data from a high level
- Includes entities and their relationships
  - Referred to as Entity Relationship Diagram (ERD)
- An entity identifier is the only attribute that needs to be included
- Many to many relationships are OK
Logical Data Model (LDM)

- Describes business information
- Defines business rules
- Provides attributes
- Tables are normalized (Discussed later)
- Database independent
Physical Data Model (PDM)

- Business Rules are constraints
- Describes the implementation in a targeted database
- LDM generic types are replaced with DB2 for i specifics
- LDM Surrogate key is PDM identity column
- View de-normalizes data
- Attributes are columns
- Normalization
- Is the blueprint for the generated DDL
- Hides complexity
- Minimizes coding

- Constraints
- Columns
- Data Type
- Length
- Scale
- Not Null
- Generated
- Default Value/Generate...
Data Dictionary Models

- **Domain Model**
  - Typically used for
    - standardization of data types
    - Impact analysis

- **Glossary Model**
  - Typically used for standardization of data names and abbreviations
Creating a Data Model

- IBM Data Modeling Tools
- Launching the Workbench
- Setting preferences
- Understanding Perspectives and Views
- The Properties View
IBM Data Modeling Tools

- Infosphere Data Architect (IDA) and IBM Data Studio (IDS)
  - Provide tools for developing database applications.
  - Can be used with multiple vendor solutions, not just DB2 for i
  - IDS is integrated with IDA

- Both products are part of an Integrated Development Environment (IDE) providing:
  - Design and deployment of database objects
    • tables, views, indexes, etc.
  - Development and deployment of SQL queries and routines
    • SQL Functions, Procedures, and triggers

- Can be installed with other IBM software products so that they share a common workbench.

- They are not a replacement for System i Navigator
  - Use System i Navigator for admin, query analysis
  - Use IDA/IDS for application development and enterprise database management
Available Tools

- **IBM Data Studio (IDS)**
  - Reverse Engineering
  - Physical Data Modeling
  - SQL scripting and routine development

- **InfoSphere Data Architect (IDA)**
  - Same as IDS plus
  - Logical Data Modeling
  - Data Dictionary

- **Rational Developer for IBM i (Rdi)**
  - Same as IDS without modeling
  - In addition:
    - DDS, RPG and/or COBOL

- **System i Navigator (iNav)**
  - DB2 for i Only
    - Not eclipse based
  - Reverse Engineering
  - SQL scripting and routine development
  - Visual Explain
Installing IBM Data Modeling Tools

Standalone
- Own package group
- Or add to compatible Rdi package group

With Rdi
- Install together into same package group
Validating IDS/Rdi Installation

- If you intend to use IDA/IDS and Rdi then it is important that they are installed in the same package group.

View Installed Packages

### Package Group
- IDS 4.1.2
- Rdi 9.5.2
Launch IDA/IDS

- IDA/IDS can be launched in 2 ways:
  - Windows Program List
  - Desktop icon (Shortcut)

### Windows Program List
- Click on IBM Data Studio in the program list

### Shortcut
- Double click on the Data Studio icon located on your desktop
The Workspace Launcher

- A workspace is a folder that contains the tools and supporting artifacts for a given session
  - Think of it as a Workbench
- When starting IDA/IDS you must choose an existing workspace - or create a new workspace
- There is no limit to the number of workspaces you can have
Creating A New Workspace – Step 1

**Workspace Launcher**
- Choose Browse from the Workspace Launcher

**Workspace Directory**
- Locate the folder to use in the Select Workspace Directory
- You can also create a new folder
Creating A New Folder – Step 2

Workspace Directory
- Click on Make New Folder
- Rename the new folder (e.g. DCP Workspace)
- Press OK

The Workbench
- Default Perspective
- Become familiar with the Task Launcher
- Can be opened anytime from Help
Understanding Perspectives

Workbench
- Made up of 1 or more perspectives
  - Data, IBM i, etc.
- A perspective defines the initial set and layout of views in the Workbench window.
  - The same set of editors are shared by each perspective within the window.
- Each perspective provides a set of tools aimed at accomplishing specific functions depending on the type of work.
Opening the Data Perspective

- The Data Perspective contains the views and tools for Data Centric development
  - SQL statement development and testing
  - SQL routine (procedures, triggers, functions, etc.) development and testing
  - Database design, DDL generation and deployment
- Use the Menu Bar Window view, or tool bar, to open the Data Perspective

Menu Bar->Window
- Open Perspective->Other
- Choose Data from the list of perspectives
Understanding Views

- Views support editors
- Views provide alternative presentations and ways to navigate the Workbench.
- You can change the layout of a perspective by:
  - Opening and closing views
  - Docking views in different positions
  - Combining views from different perspectives
- For example, the i Project Navigator and Remote Systems views can be added to the Data perspective.
View Menus

- Views have their own menus.
  - To open the menu for a view, click the icon at the left end of the view's title bar.

- Some views have their own toolbars.
  - The actions represented by buttons on view toolbars only affect the items within that view.

- A view might appear by itself, or be stacked with other views in a tabbed notebook.
The Properties View

- This view displays property names and basic properties of a selected resource.
- The types of properties vary by resource.
- It is the “go to” view for many resource actions – e.g. adding attributes, changing information.
The Physical Data Model (PDM)

- The PDM is where you add, modify and deploy data objects (i.e. schemas, tables, etc.)
- PDMs are created within a Data Design Project
  - Multiple models can be created in a single project
  - Defined from scratch, transformed from an IDA LDM or via reverse engineering from IBM i
- The ultimate goal of using the PDM is to eventually eliminate IBM i source files for DDL
  - PDMs can be managed via Eclipse based SCM products
  - DDS source would still be stored in IBM i DDS source files
Creating a Physical Data Model (PDM)

- To create a PDM follow these steps:

  Expand the Data Design Project
  - Right click on Data Models
  - Choose New->Physical Data Model

  New Physical Data Model wizard
  - Enter a name for the PDM
  - Select your current release level
  - Choose Create from Template then press Finish

  Expand the new Physical Data Model
  - Right Click on Schema
  - Choose Refactor->Rename to change the schema name
Creating a Diagram

- A diagram is used to visualize the tables, views and relationships
- The diagram consists of 3 main parts
  - Editor
  - Palette
  - Outline

To create a diagram:
- Expand the Data Model
- Right click on Diagrams
- Click New Blank Diagram
Generating DDL from the Data Design Project

- DDL can be generated from any of the following:
  - Folders such as Database and Schema
  - Database Objects (tables, views, functions, etc.)
  - Diagrams

- The Generate DDL wizard can be accessed by:
  - Right clicking on Folders and Database Objects
  - Choosing Data from the Menu bar
    - DDL will be generated from the currently open folder, object or diagram

- The Generate DDL wizard has a series of prompts allowing you to customize the generation process:
  - Options, Objects, Save and Open
Generating DDL from the Data Design Project

- DDL can be generated from any of the following:
  - Folders such as Database and Schema
  - Database Objects (tables, views, functions, etc.)
  - Diagrams

- The Generate DDL wizard can be accessed by:
  - Right clicking on Folders and Database Objects
  - Choosing Data from the Menu bar
    - DDL will be generated from the currently open folder, object or diagram

- The Generate DDL wizard has a series of prompts allowing you to customize the generation process
  - Options, Objects, Save and Open
Modifying the Physical Data Model

- Change is inevitable
- At this point all future changes to the database should be made within the data model
- Create a new diagram containing the objects to be changed
- Once changes are compete use the Compare and Merge tool to generate delta DDL
Comparing and Merging Modifications

**Filter**
- Based on changes
- Select to fit needs
- Only the table was changed

**Compare**
- Left side is PDM
- Right side is DB2 for i
- Copy changes left to right
- Generate Delta DDL

**DDL**
- `ALTER TABLE ADD COLUMN`
- Homework
  - Read up on `ALTER`
Demo 1 Creating a Data Model

- In this exercise you will perform the following tasks:
  1. Create a Data Design Project
  2. Create a new data model
  3. Add a table to a blank diagram
  4. Add columns to a table in a diagram
  5. Create a relationship between two tables
  6. Copy columns from one table to another
  7. Generate and review DDL from a diagram
Reverse Engineering Using IDS/Rdi

- Follows basic flow on right
- Multiple methods
  - Using template individual objects can be reversed engineered from Data Source Explorer
  - Entire Schemas can be reversed engineered from Data Design Project

1. Create a Data Design Project
   - Current State
2. Create a Physical Data Model
   - From template or from RE
3. Choose Method
   - Entire schema or selected objects
4. Create and review Overview Diagram
Reverse Engineering From a Schema

- All existing database objects contained in schema are eligible
- Only select Tables and supporting objects
  - No views, synonyms, UDTs or routines
- Avoid re-engineering the entire schema
  - It is not necessary and can be very time consuming
Reverse Engineering From IBM i (IDS/Rdi) Part 1

- Recommended approach
- Ability to select individual table objects from a schema

Create a new Physical Data Model
- Choose DB2 for i
- Choose the current release of IBM i
- Select Create From Template
- Press Finish

Expand the new Data Model
- Expand Database
- Locate Schema

From the Data Source Explorer
- Locate the Schema containing the files
- Select the files
- Drag and drop onto Schema in Data Project Explorer
Create an Overview Diagram
- Expand Schema
- Right click on Diagrams and choose Overview
- Name the new diagram Current State
- Select Schema

Review the Diagram
- Rearrange tables as desired
- Look for implicit relationships
- Add CFI
Considerations When Using IDA/IDS

- Latest tool release may not be current with DB2 for i release
  - May be release tolerant only
- Eclipse based tool
  - Supports most, if not all, relational databases
- Able to manage and modify data models offline
- IBM Data Studio is no charge (free)
Demo 2 Reverse Engineering

- In this exercise you will perform the following tasks:
  1. Reverse engineer an existing schema into a PDM.
  2. Create a new data model from an existing data model.
  3. Modify existing tables by adding core fundamental items.
  4. Deploy the new data model.
  5. Migrate the data.
Data Migration

- Export/Import Auto-generated values using Triggers
- Initial bulk load – No triggers
- Synchronization – Virtual Layer not in place
- Transaction processing – Live, virtual layer in place
Export/Import Auto-generated values using Triggers

- Insert into parent with auto-generated columns (e.g. IDENTITY)
  - AFTER Trigger sets (exports) global variable with auto-generated value

- Insert into child with hidden columns (e.g. foreign key)
  - BEFORE Trigger sets hidden column in child using global variable value (import)

Session global variables
Initial bulk load – No triggers

- Disable triggers

- Insert with sub-select into parent (Alias)
  - Sub-select references surrogate file or view.
  - Alias does not contain hidden columns

- Insert with sub-select into child (all columns)
  - Sub-select uses join to parent to retrieve auto-generated values.

After parent load each child load can run in separate session

Enable Triggers
Synchronization – Virtual Layer not in Place

Journal Event

Insert, update or delete parent.

- All values come from journal image.
- Primary key used for update or delete
- If RI enabled, child deletes can be ignored

Insert or update of child

- Insert values come from journal image.
- Primary key for update comes from image or parent table using sub-select

Commit
Transaction processing – Live, virtual layer in place

- Insert, update or delete event against view

- Parent
  - All values come from transaction.
  - Combination of application key and row change timestamp used for update or delete
  - If RI enabled, cascade delete for identified dependents

- Child
  - Insert/update values come from transaction, hidden values come from global variable.
  - Primary key for update comes from parent table using sub-select

- Commit
SQL Index as Surrogate LF

- In exceptional cases an SQL index can be a surrogate LF
  - Consolidation of files/members where new column is required
  - Overcoming limits to growth

- The RCDFMT is required to match existing file format
  - Can be referenced via FORMAT keyword
  - WHERE clause can reference column not in format
    - DDS S/O fields must exist in format

- Index can span multiple partitions (default behavior)

- The following table compares a DDS surrogate file to a DDL surrogate index:
  - Programs continue to work without modification, compilation is optional

<table>
<thead>
<tr>
<th></th>
<th>Select/Omit LF</th>
<th>Sparse SQL Index</th>
<th>Spanning Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record format contains tenant id</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Record format does not contain tenant id</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Table exceeds non partitioned size limit</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
What Are We Building?

- **Application Layer**
  - Host programs
    - RLA, DDS Logical files
    - External (Browser, Java, ODBC, etc.)
    - Open Access as bridge to SQL

- **Data Access Layer**
  - SQL Procedures for all new applications
    - External Procedures using embedded SQL

- **Virtual or Data Abstract Layer**
  - Aliases/Views for SQL access
    - Instead of Triggers
    - Could be multiple schemas

- **Physical Data Model**
  - DB2 for i
  - Normalized
  - Indexes, triggers, constraints
  - Typically a single schema
IBM Data Modeling Tools Summary

- Data Modeling is:
  - a method used to define and analyze data requirements
  - used to communicate the business rules and processes
  - the process of creating a blueprint to visually represent data

- There are four types of data models:
  - Conceptual, logical, physical and data dictionary

- IBM provides two data modeling tools for use with IBM i
  - IBM Data Studio (IDS) – Free download
    - Reverse Engineering, Physical Data Modeling,
    - SQL scripting and routine development
  - InfoSphere Data Architect (IDA) – Requires License
    - Same as IDS plus
    - Logical Data Modeling and Data Dictionary (Domain and Glossary)
    - Data Model Analysis (naming standards and normalization compliance)

- “A picture is worth a thousand words”
More Help Getting Started

Search the web for Getting started with Infosphere Data Architect

Search the web for Getting started with IBM Data Studio
Where to Find More Details on DB2 for i

- **developerWorks**
  - DB2 for i enhancements: [https://www.ibm.com/developerworks/ibmi/techupdates/db2](https://www.ibm.com/developerworks/ibmi/techupdates/db2)

- **Blog**
  - Mike Cain: [http://db2fori.blogspot.com](http://db2fori.blogspot.com)

- **Modernization Redbook**
Questions?