Appendix D

Use of CASE Tools

A very large array of CASE (computer-aided software engineering) tools is available to support every type of computing effort in today's environment. Over the years, the CASE tools market has matured with some leading vendors producing sophisticated tools. In today's industry, no aspect of computing seems to be beyond the scope of CASE tools. In order to give you a flavor of the wide variety, here is a brief sample of the numerous categories of CASE tools:

Analysis
Animation of mission-critical systems
Application generation for mainframe systems
Architecture modeling
Automated documentation
Batch code analysis
Business modeling
Business process re-engineering
Change management
Charting and graphing
Client/Server development
Code generation
Code visualization
Component modeling
Component relationship design
Cost estimation
Cross-referencing
Data collection
Data conversion
Data exchange
Data management
Data modeling
Data sharing
Database design
Database publishing
DBMS testing
Design
Diagramming and flowcharting
Flow diagram editor
Forward engineering
Full life cycle

Database Design and Development: An Essential Guide for IT Professionals by Paulraj Ponniah
ISBN 0-471-21877-4 Copyright © 2003 by John Wiley and Sons, Inc.

Function modeling
GUI development
Impact analysis
Information engineering
Java development
Metamodeling
Object modeling
Object-oriented analysis and design
Object-oriented modeling
Performance monitoring
Performance simulation
Problem tracing
Process management
Process modeling
Program profiling
Project management
Prototyping
Rapid application development
Report generation
Requirements engineering
Reverse engineering
Runtime error checking
Simulation
SQL code generation
Structured analysis and design
Test case generator
Test management
Version management

In the study of database design and development, however, we are more interested in the tools that aid in the design, development, and implementation of database systems. This appendix highlights the major features of CASE tools applicable to the database development life cycle (DDLC).

LOGICAL DATA MODELING

- Defining and naming entities and attributes
- Selecting primary keys
- Designating alternate key attributes
- Defining one-to-one and one-to-many relationships
- Resolving many-to-many relationships
- Specifying special relationship types (*n*-ary, recursive, subtype)
- Defining foreign keys and specifying identifying and nonidentifying relationships
- Establishing referential integrity constraints
- Completing the entity-relationship diagram (ERD)

PHYSICAL DATA MODELING (for the relational model)

- Transforming entities into tables
- Converting attributes into columns
- Assigning primary and foreign keys
- Defining data validation constraints
- Defining triggers and stored procedures for business rules
- Including triggers for INSERT, UPDATE, and DELETE to preserve referential integrity
- Set data types based on target DBMS

DIMENSIONAL DATA MODELING

- Defining fact tables
- Defining dimension tables
- Designing the STAR schema
- Designing outrigger tables (snowflake schema)
- Accounting for slowly changing dimensions
- Defining and attaching data warehouse rules
- Defining data warehouse sources
- Importing from data warehouse sources
- Attaching sources to columns

CALCULATING PHYSICAL STORAGE SPACE

- Estimating database table sizes
- Establishing volumes
- Setting parameters for space calculations

DOMAIN DICTIONARY

- Establishing standards
- Setting domain inheritances and overrides
- Creating domains
- Defining domain properties
- Changing domain properties

FORWARD ENGINEERING

- Selecting schema generation options by category
- Setting schema generation options: referential integrity, trigger, schema, table, view, column, index, and special features relating to target DBMS
- Reviewing summary information before schema generation
- Generating SQL data definition code for target server
- Executing SQL code to generate appropriate schema definitions

BACKWARD OR REVERSE ENGINEERING

- Selecting data dictionary entries of the relational database
- Selecting the file of the SQL data definition statement
- Creating the physical data model
- Creating the logical data model
- Reviewing generated data models