

Introduction

Many hours of programming time are dedicated to tasks that are repeated on every project. With regard to data and databases, several areas usually require effort to design and code:

- Data history (audit trails, "undo", archiving, etc...)
- Surrogate keys (very useful, but not intuitive to users)
- Simple domain integrity (ex. "yes" or "no" without foreign key tables)
- Multi-Tier deployment (data caching and integrity at the mid-tier)
- Hierarchical data and paths (ex. manager to employee relationships)
- Data maintenance forms (basic data query and modification)
- Background logging facility (including ease of instrumentation)

One way to avoid hours of programming time is to use a software generator. Many software generators are available. Oracle's own Data Modeler can generate data dictionary language (DDL) for the Oracle database. However, the real strength of a generator comes from an ability to modify the generator for individual project or company needs. DTGen was created as a starting point for customized Oracle database software generation.

Data History

- Tracking of what data was available when
- Complicated by data entry errors
- "Undo" is a modern expectation of data systems
- Auditing implies tracking who changed the data
- Auditing "Undo" is even more complicated
- Reporting becomes very difficult
- LOG tables do simple "when it happened" tracking
- EFF tables allow historical entry of "when it happened"
- OMNI views allow a complete view of data, history, and audit

Surrogate Keys

- Natural keys are easier for the user to understand
- Enable natural key changes without losing original record reference
- A single number is simpler than multi-column natural keys
- Can follow records from one database to another
- More difficult for user to work with than natural keys
- Transactional views allow natural key data manipulation of foreign surrogate keys

Simple Domain Integrity

- Any small, discrete data set that rarely changes
- Examples are (Yes, No), (Male, Female), (Flag), Types of Things
- Avoid using foreign key tables
- Avoid overloading a single table of domain values
- Automatically built into application as check constraints

Multi-Tier Deployment

- Need to move data integrity checks out of database
- ex. Transaction Processing Performance Council's TPC-C
- Caching of slow-moving data outside of database
- Distributed table packages allow integrity checks at mid-tier
- Materialized views allow data caching at mid-tier

Hierarchical Data and Paths

- A table that is a foreign key to itself
- ex. A manager oversees employees and is also an employee
- Hierarchical organizations imply an organizational path
- Transactional views allow hierarchical path data manipulation

Data Maintenance Forms

- Basic data query user interface as soon as schema is generated
- Data maintenance with application specific integrity checking

Background Logging Facility

- Easily capture error and debug information for administrators
- Autonomous transaction for independently committed log data
- DBMS packages for self-identifying location data

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates.