

Using Thick Database Principles to Leverage Oracle SQL and PL/SQL Part II:

Design, Create, and Maintain a Business Rules Repository

Peter Koletzke

Technical Director & Principal Instructor



ORACLE
ACE Director

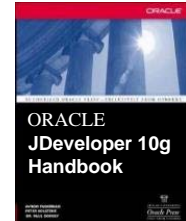


ORACLE
Developer
Community



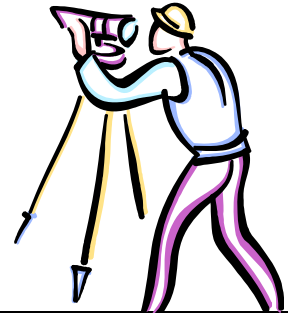
Me

- 34 yrs. database industry
- 30 yrs. consulting in Oracle arena
 - Since Oracle 5.1C, SQL*Forms 2.3
- 38 yrs. as trainer/presenter
- User groups
 - 350+ presentations, 12 awards
 - 7+ yrs. total on boards of directors
 - IOUG(-A), NYOUG, UTOUG
- Oracle ACE Director
 - Since program inception in Aug. 2005
- Oracle Certified Master
 - Since program inception in Dec. 2001
- 8 Oracle Press books coauthored
 - 6262 pages total



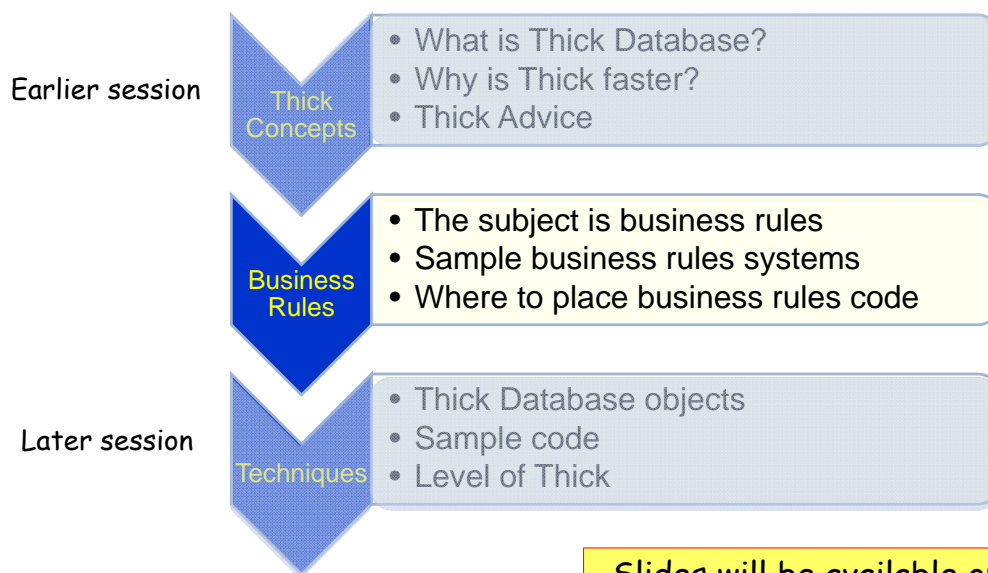
You

- Job responsibilities?
 - DBA, developer
- Development tools?
 - Oracle Developer Forms/Reports
 - APEX
 - ADF, MAF
 - MAX, VBCS
 - Other JavaScript tools
 - .NET
 - PL/SQL
 - Other



3

Series Overview

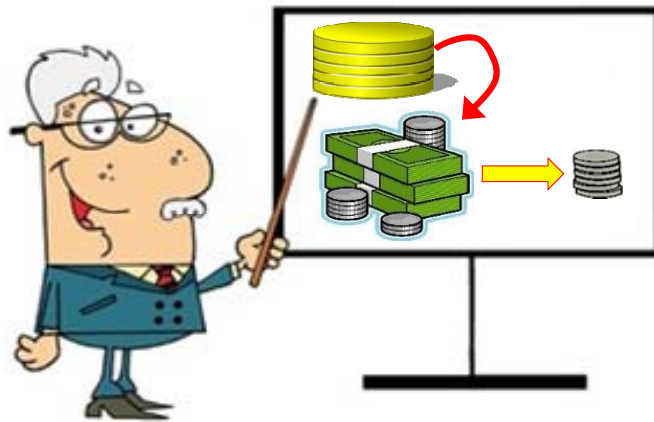


Slides will be available on the NoCOUG website.



4

A Brief Review



Review

About Thick Database

Review

- A code development strategy
 - Maximize use of database code to simplify the user interface
 - The user's device (client) runs minimal code
- Name plays off the term "thin client"
 - A "Year of the Internet" term
 - Means most processing occurs on a server
 - Slightly outmoded now
- Thick database means "thin client"

A.k.a.

- Thick Database Approach
- Thick Database Paradigm
- Smart Database
- SmartDB
- Fat Database

Oracle prefers

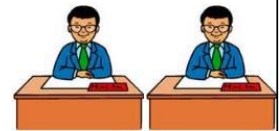


Review

Guiding Principles for Code

Review

- Database code that implements business rules in PL/SQL
 - Using database features to enforce data integrity defined by business rules seems obvious
- Database views to represent complex business objects (SQL)
 - Each view has an accompanying application programming interface (API)
 - Written in PL/SQL
 - Interaction is with view and API



7

Review

Benefits

Review

- Application accuracy
 - Business rules match application code
 - Test plans can be generated from business rules system
- Productivity
 - Can greatly simplify user interface code
- Code reusability
 - Ease of application maintenance
- Faster performance
 - Code is close to data storage – fewer messages, easy access
 - Views also reduce the number of round trips needed
- Proper use of staff
 - User interface developers can concentrate on UI code
 - Database code developers can concentrate on database code to support the UI



8

Review

More Benefits

Review

- Save cloud database processing time
 - Application's use of database is more efficient
 - Proof in next section
 - Less database processing time == lower cloud costs
 - Save application processing time, too
- Simplify user interface development
 - Database views and PL/SQL form API to the database
 - Application code is reduced
 - UI code technology can change without total application rewrite



9

Review

Two Related Strategies

Review

1. System for tracking business rules
 - Store definitions in database tables
 - Code implementations linked to rules
 - Linked to, or serves as, requirements documentation
2. Database features implement the business rules
 - Constraints
 - Database API: PL/SQL code, updatable views



Oracle is considering support for SQL Assertions:
<https://community.oracle.com/ideas/13028>



10

Agenda

Business Rules

- The subject is business rules
- Sample business rules systems
- Where to place business rules code



11

Oracle Does Not Support Plowing

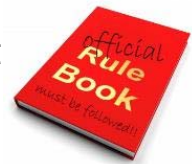
Too thick to drink,
too thin to plow.

—Common saying about
Nebraska's Platte River

12

What is a Business Rule?

- *A statement of a behavior, definition, or constraint that allows an organization to achieve its goals.*
- Systems analysis is all about determining business rules
 - Often *business requirements* are equated with business rules
 - Business rules used to communicate business with business users
- A full definition of business rules can identify all aspects of an application
 - Possible exceptions: technical details like development software, server specifics



13

Sample Business Rules

- *An employee is active in one and only one department at a time.*
- *The employee's job end date must be on or after the job start date.*
- *The value of the "State" portion of an address in the United States must be from the list of US states (including the District of Columbia).*
- *Saving a change to an employee record archives the old version of the record in a history table.*
- *Only managers can view salaries other than their own for staff in their department.*

14

Business Rules Categories

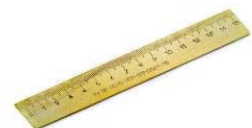
- Business definition
 - A statement that explains a fact relevant to the business, for example:
 - *An employee is active in one and only one department at a time.*
- Data validation
 - A statement that describes how data is verified, for example:
 - *The employee's job start date must be on or after the job start date.*



15

More Business Rule Categories

- Allowed values
 - Related to data validation
 - Defines field values from a fixed list (hard coded or in a table) or range, for example,
 - *The value of the "State" portion of an address in the United States must be from the list of US states (including the District of Columbia).*
- System Behavior
 - A statement that guides the internal actions in the system, for example:
 - *Saving a change to an employee record archives the old version of the record in a history table.*



16

Another Business Rule Category

- Data privileges
 - Selective to users or (better) roles
 - Defines access to view or modify certain data, for example:
 - *Only directors can update salaries for staff in their division.*
 - *Only managers can view salaries other than their own for staff in their department.*
 - *Personal data for clients, such as credit card numbers and Medicaid IDs, are only visible to staff who have been cleared to view it.*
 - *Staff may only view profile information for clients in the department's territory.*



17

Test Plans and Business Rules

- Business rules statements can be used as or linked with test plans
- Generation of test plans is then just a report
- Each business requirement will be properly tested



18

Agenda

Business Rules

- The subject is business rules
- Sample business rules systems
- Where to place business rules code



19

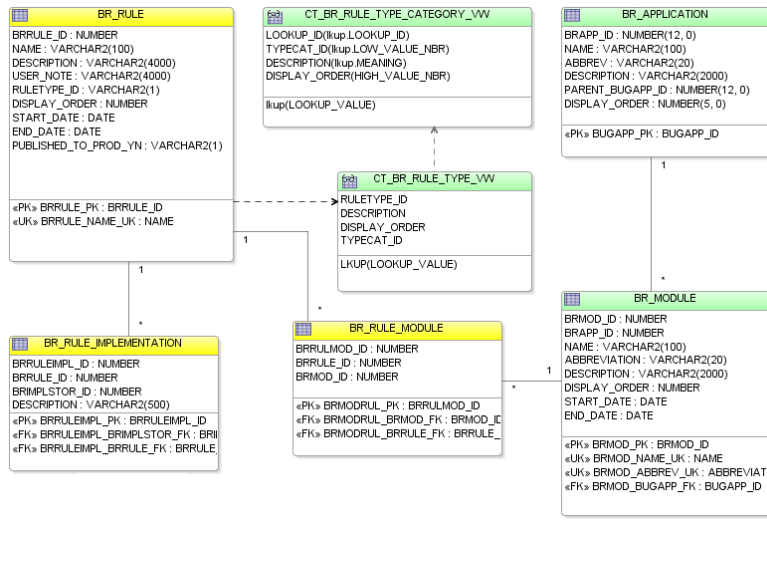
Primary Business Rules Tables

- BR_RULE
 - Business rules descriptions and dates.
- BR_RULE_IMPLEMENTATION
 - Code and column implementations that enforce the business rule.
- BR_IMPLEMENTATION_STORE
 - A code or other unit that enforces business rules.
- BR_APP_MESSAGE_IMPLEMENTATION
 - Code units that use the application message. Maintained by triggers.
- BR_RULE_MODULE
 - The instance of a business rule for a specific module.
- BR_MODULE
 - Module descriptions and dates to which business rules will attach.



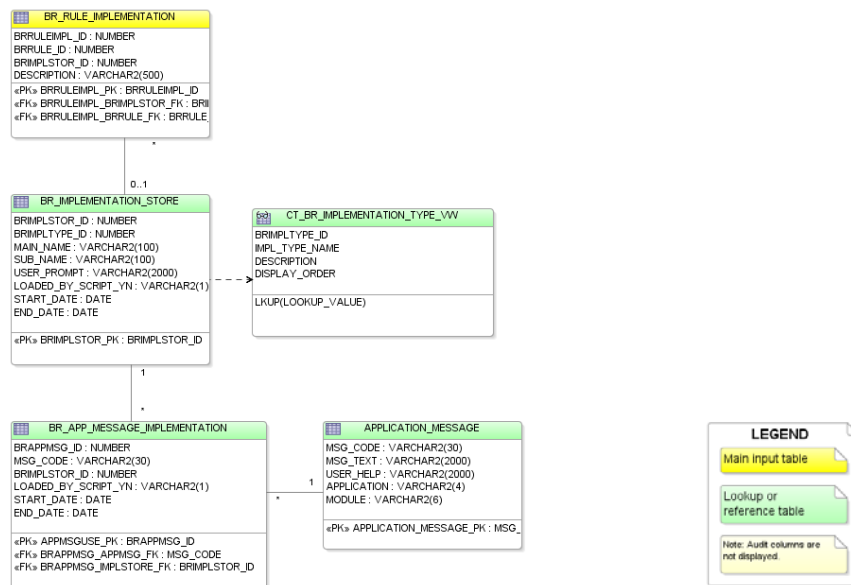
20

Sample BR Repository – Main Tables



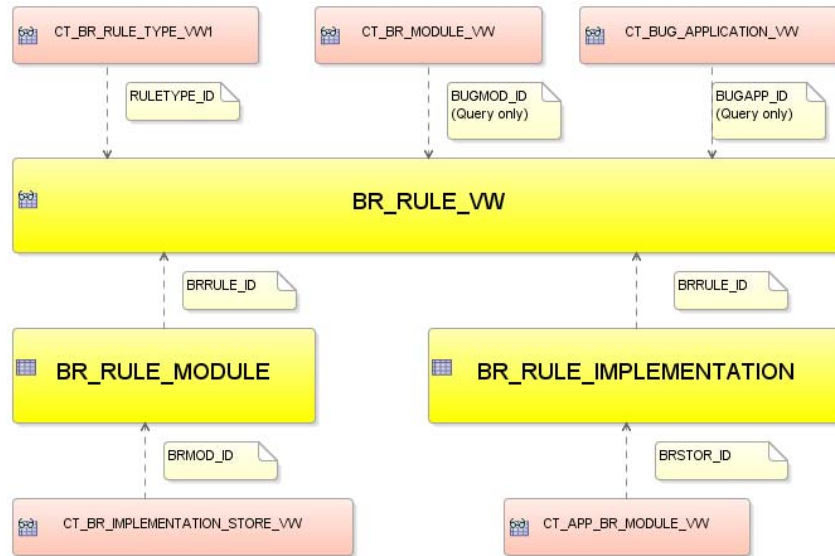
21

Sample BR Repository – Other Tables



22

UI Data Model



23

Search Page

Business Rule Search

[New Business Rule](#)

☒ Result matches all conditions
☐ Result matches any condition
 Case Sensitive? ☐

Rule ID:
 Application: Module:
 Rule Name:
 Description:
 Rule Type: Published to Prod?: MIS Internal?:

[Find](#) [Clear](#)

[Details](#) [Previous](#) 10 21-24 of 24 [Next](#)

Select	Details	Rule ID	Application - Module	Rule Name	Pub?	MIS Int?
<input checked="" type="radio"/>	Show	80	Assessment	Birth Date vs Transaction Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="radio"/>	Show	80	Screening	Birth Date vs Transaction Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="radio"/>	Show	80	Treatment	Birth Date vs Transaction Date	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[New Business Rule](#)

24

Details Page

Business Rule | Message Implementation | Application Message | Implementation Store

Business Rule Search >

Edit Business Rule 80

Save | Migration Script

[1 / 18] > >>

Rule ID : 80

* Rule Name : Birth Date vs Transaction Date

Description : Birth date must be earlier than all transaction dates in the client history.

User Note : Birth date must be earlier than all transaction dates in the client history.

* Rule Type : Cross-row validation Published to Prod? : ☒ MIS Internal? : ☐

* Start Date : 01/01/1985 End Date : End Date : Display Order : 0

(mm/dd/yyyy) (mm/dd/yyyy)

New Business Rule

25

Details Page (continued)

* Rule Type : Cross-row validation Published to Prod? : ☒ MIS Internal? : ☐

* Start Date : 01/01/1985 End Date : End Date : Display Order : 0

(mm/dd/yyyy) (mm/dd/yyyy)

New Business Rule

Application - Modules

Available




Sales Rep Profile
Sales Order Entry
Sales Reporting
Marketing Personnel

Selected

Marketing Reporting
Marketing Leads

Business Rule Implementation

Add Row Delete

Details	Main Name	Sub Name	Impl Type Name	Delete?
Show	CDS_COMMON_PKG	ADMISSION_VS_BIRTH_DATE	PLSQL CODE UNIT	 <input type="checkbox"/>
Show				 <input type="checkbox"/>
Show				 <input type="checkbox"/>

26

Implementation Store Page

The screenshot shows the 'Implementation Store Search' interface. At the top, there are tabs for 'Business Rule', 'Message Implementation', 'Application Message', and 'Implementation Store'. The 'Implementation Store' tab is selected. Below the tabs, there are search criteria: 'Result matches all conditions' (selected) and 'Result matches any condition'. There is a 'Case Sensitive?' checkbox. Search fields include 'Main Name', 'Subname', 'Implementation Type' (dropdown), 'User Prompt', 'Loaded by Script?' (dropdown), 'Start Date From', 'Start Date To', 'End Date From', and 'End Date To'. There are 'Find' and 'Clear' buttons. Below the search fields is a table with columns 'Main Name', 'Subname', 'Impl Type', and 'Delete?'. The table has a 'Hide' button and a 'Show' button. Below the table, there are fields for 'User Prompt', 'Loaded by Script?' (checked), 'Start Date' (mm/dd/yyyy), and 'End Date' (mm/dd/yyyy).

27

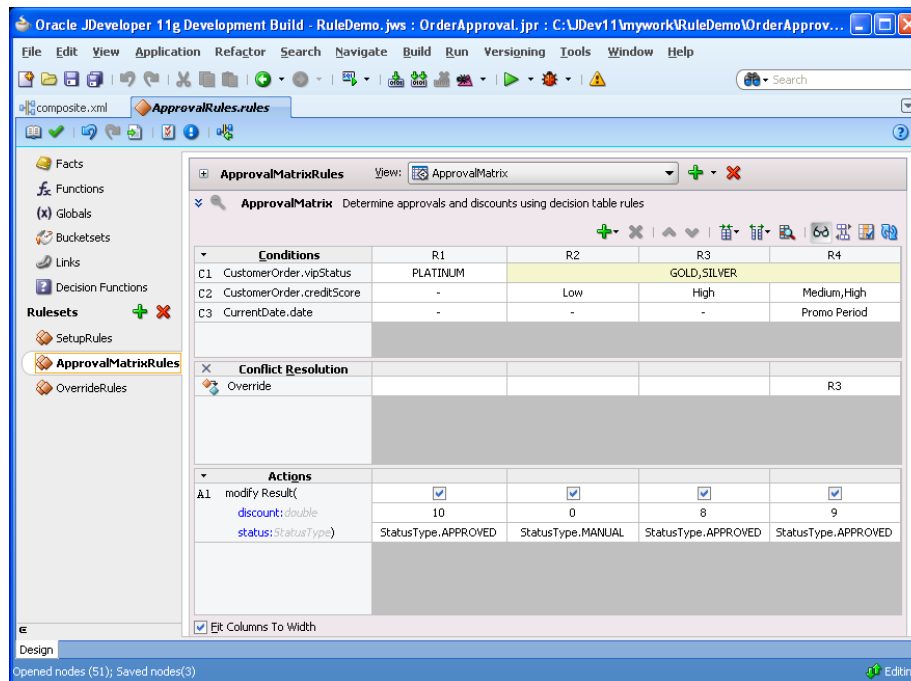
About Oracle Business Rules

- Product alternative to Thick Database
- Part of SOA Suite and BPM Suite
- Define rules in English
 - Definitions stored in XML
 - Deployed as a Java application for use as a service
 - Your application calls the service to perform business rules work
- Based on Sandia Labs' Jess
 - Highly reliable, thousands of users
 - Industry standard Rete algorithm

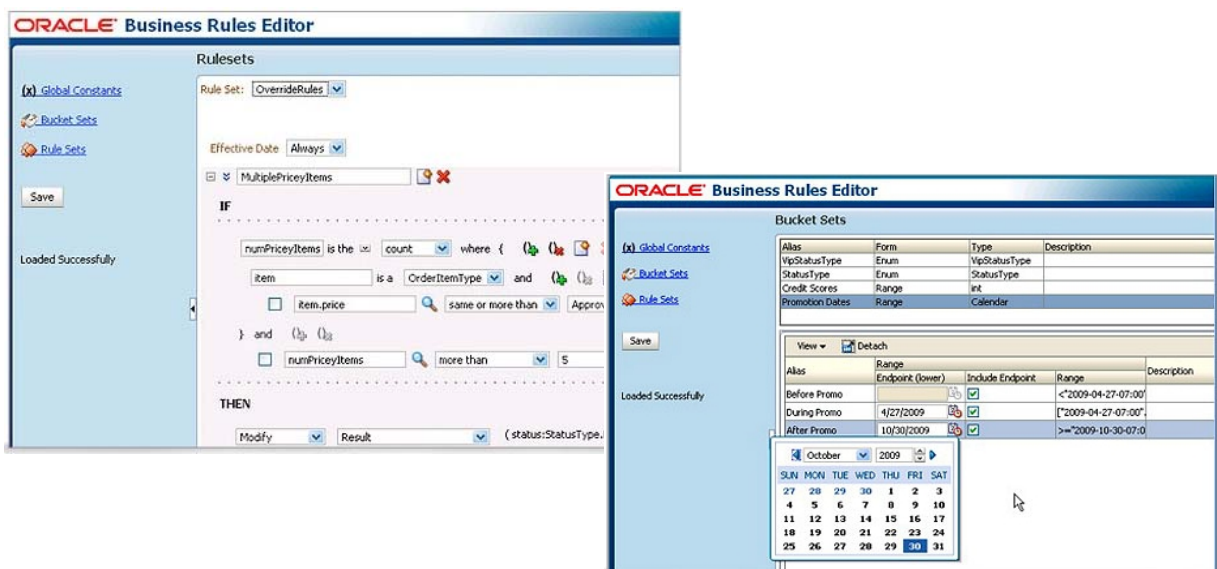
ORACLE

28

Oracle Business Rules (in JDev)



Oracle Business Rules Editor



Agenda

Business Rules

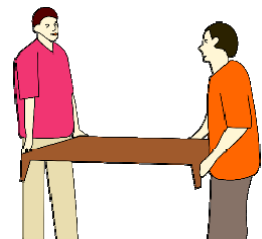
- The subject is business rules
- Sample business rules systems
- Where to place business rules code



31

Where to Place the Business Rules Code?

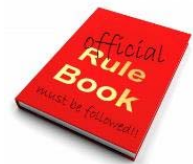
- Environments these days are multi-tier
 - Database tier
 - Middle tier
 - Application server/web server (SOA, web services, ESB, etc.)
 - Client tier
 - Web browser
 - Mobile device
- Code can be located on one or more tiers



32

Primary Assumption

- Standard relational database constraints are ALWAYS used to protect data integrity
 - Primary key
 - Foreign key
 - Unique key
 - Check constraints
 - NOT NULL
 - Value- or function-base (optional)
- This is true regardless of the database vendor



33

Code on the Client Tier

- Web application consideration:
 - Since HTML is *not* a programming language, you need JavaScript for this
- Benefits
 - Fast feedback to user: very friendly
 - No processing at all on database or middle tiers
- Drawbacks
 - Difficult to maintain business rules documentation
 - Some browsers handle JavaScript differently
 - Possible need to repeat code for each app
 - Potential for omission in a single app
 - Not centralized



34

Code on the Middle Tier

- **Business rules code is in the middle tier**
 - That is, if there is a middle tier
 - For example, APEX has none
 - ADF
 - Java and XML files for the application
 - Declarative validation rules, EO, VO, App module code
- **Benefits**
 - Saves database server CPU time
 - Returns messages to user faster and friendlier
- **Drawbacks**
 - Each app needs to repeat the code for a particular table
 - Requires database roundtrip messages
 - Documenting or checking business rules requires visiting many files unless you use a Rules Engine or other repository



35

Code on the Database Tier

- **Thick Database approach**
 - Views
 - Table API code
 - Triggers and procedures (and policies) that enforce rules
- **Benefits**
 - Data integrity is enforced for all applications
 - Business rules code can be generated from metadata or, at least, documented from one source
 - Maintenance requires only database changes
 - Application modification may not be needed
 - Primary language is PL/SQL
- **Drawbacks**
 - Handling return messages from the database in a friendly way is not a default
 - Places complete burden of validation of data on the database server – possibly more CPU time taken



36

So, Which is Best?

- Depends on the application
- Database tier (Thick Database) ensures data integrity
 - Any application
- Middle tier saves database round trips
 - If processing only on middle tier
- Client tier provides best interactivity
 - Immediate feedback to user
 - Also saves database round trips



37

Feature Comparison

1 (no support) to 10 (the best support)

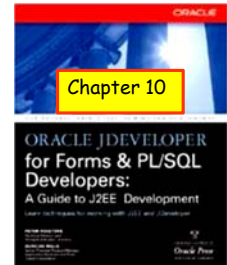
Feature	Location of Business Rules Code	Client Computer	Application Server	Database Server
User interactivity		10	7	5
Saves client computer resource usage **		2	10	10
Saves roundtrip message to client computer			10	10
Saves application server resource usage **		10	2	10
Saves roundtrip message to application server		10		10
Saves database server resource usage **		10	10	2
Saves roundtrip message to database server		10	10	
Ease of maintenance (dependency analysis, adding, updating, reporting) *		2	5	10
Reuse of code		2	5	10
Assurance that business rules are applied to all applications		5	5	10
Total		61	64	77

* Assumes that the business rules repository is not used at runtime or to generate code.

** This feature reflects use of that tier for business rules purposes

Suggestion

- *Modified Database-centric Approach*
 - Always code rules in the database
 - Selectively duplicate business rules in the middle tier and client tier
 - Carefully consider each rule
 - Know and document that you are duplicating rules
 - Can even turn off database rule for a transaction if it has been run on the client side
 - Consider using a BR repository tool
 - Home grown or Oracle Business Rules
- Guiding principles
 - Use database code when possible
 - It is the closest to the data == most efficient
 - Save database round trips when possible
 - Client side can check data type, for example



39

Some Challenges

- Identifying business rules
- Stating them accurately
- Representing the business rules in system programmatic code
- Defining and maintaining business rules statements
- Communicating rules to users
- Synchronizing programmatic code and the business rules repository



40

Summary

- Enforcing business rules is the goal of ThickDB
- Business rules coordinate with requirements and test plans
- Business rules repositories can save maintenance time
- Different levels of business rules repository use
- Requires management buy-in



41



- Please fill out the evaluations
- 7 of 8 books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills
- Slides will be on the NoCOUG website

42