OrientDB Design & Development

Two-Day Training Syllabus

Day One (Design & Modeling)

* Class Structure and Goals
	+ Training goals
	+ Intended audience
	+ Course agenda
	+ About certification
* Use Case Overview
	+ Extension of core OrientDB demonstration database
	+ Key objects
	+ Exercise and quiz summary
* What is a Graph Database?
	+ Index-free adjacency
	+ Vertices
	+ Edges
	+ Graph navigation
* OrientDB Multi Model Architecture
	+ Graph
	+ Document
* OrientDB Connection and Storage Architecture
	+ Embedded
	+ Client/server
	+ Storage models
	+ Disk persistence
	+ In-memory processing
	+ Clustering
	+ Relationships
	+ Indexing
	+ Caching
	+ Transactions
* Classes
	+ Schema-full
	+ Schema-less
	+ Hybrid
	+ Base classes
	+ Attributes
	+ Properties
	+ Inheritance
* Relationships
	+ One-to-one
	+ One-to-many
	+ Many-to-many
	+ Embedded
	+ Referenced
* Documents
	+ Terminology across RDBMS, document databases, and OrientDB
	+ Document collections
	+ Identifying a document
	+ Documents and schema
	+ Relationships
* Graphs
	+ Terminology across RDBMS, graph databases, and OrientDB
	+ Base classes
	+ Vertices
	+ Edges
	+ Lightweight vs. standard edges
	+ Graphs and schema
	+ Relationships
* A Workflow for Modeling
	+ User stories
	+ Defining major artifacts
	+ Testing model with sample queries
	+ Adjusting the model
* Evolving the Model
	+ Dealing with application changes
	+ Planning for data fluctuations
	+ Maintaining information and code integrity

Day Two (Interaction & Development)

* APIs
	+ OrientDB
	+ TinkerPop/Blueprints
	+ Choosing the optimal API
* OrientDB Object Model
* Establishing a Connection
	+ Connection options
	+ Browsing classes
	+ Querying metadata
	+ Error handling
* OrientDB SQL
	+ Standards
	+ Enhancements
	+ Major data types
* Creating New Records
	+ Record ids
	+ Defining classes
	+ Extending base classes
	+ Linking documents
	+ Linking vertices
	+ Working with binary data
* Querying the Database
	+ Browsing classes
	+ Querying metadata
	+ Working across the graph
	+ SELECT, MATCH, and TRAVERSE
	+ MATCH shorthand
	+ Sorting and aggregation
	+ Expanding results
	+ Making the most of fetch plans
* Indexing
	+ Defining effective indexes
	+ Filtering using indexes
	+ Unique indexes
	+ Composite indexes
	+ Querying against indexes
* Updating Existing Records
	+ Single values
	+ Multiple values
	+ Version control
	+ Schema-less updates
* Removing Data
	+ Removing properties from a class
	+ Deleting single records
	+ Maintaining data integrity
	+ Dropping classes
* Multi user considerations
	+ Concurrency
	+ Transactions
* Designing and Using Functions and Methods
	+ Comparison with RDBMS stored procedures
	+ Examples
	+ Interacting with the database API
	+ Accessing Java packages
	+ Developing functions and methods
* Creating and Utilizing Hooks
	+ Comparison with RDBMS triggers
	+ Custom cascading deletes
	+ Implementing algorithms
	+ Enforcing constraints
* Best practices