【A3】DatabaseGearテクニカルセッション

EMBARCADERO
TECHNOLOGIES®

CodeGear開発者のためのDatabaseGearツール入門

Embarcadero Technologies
Philip Rathle

「開発者の1日」

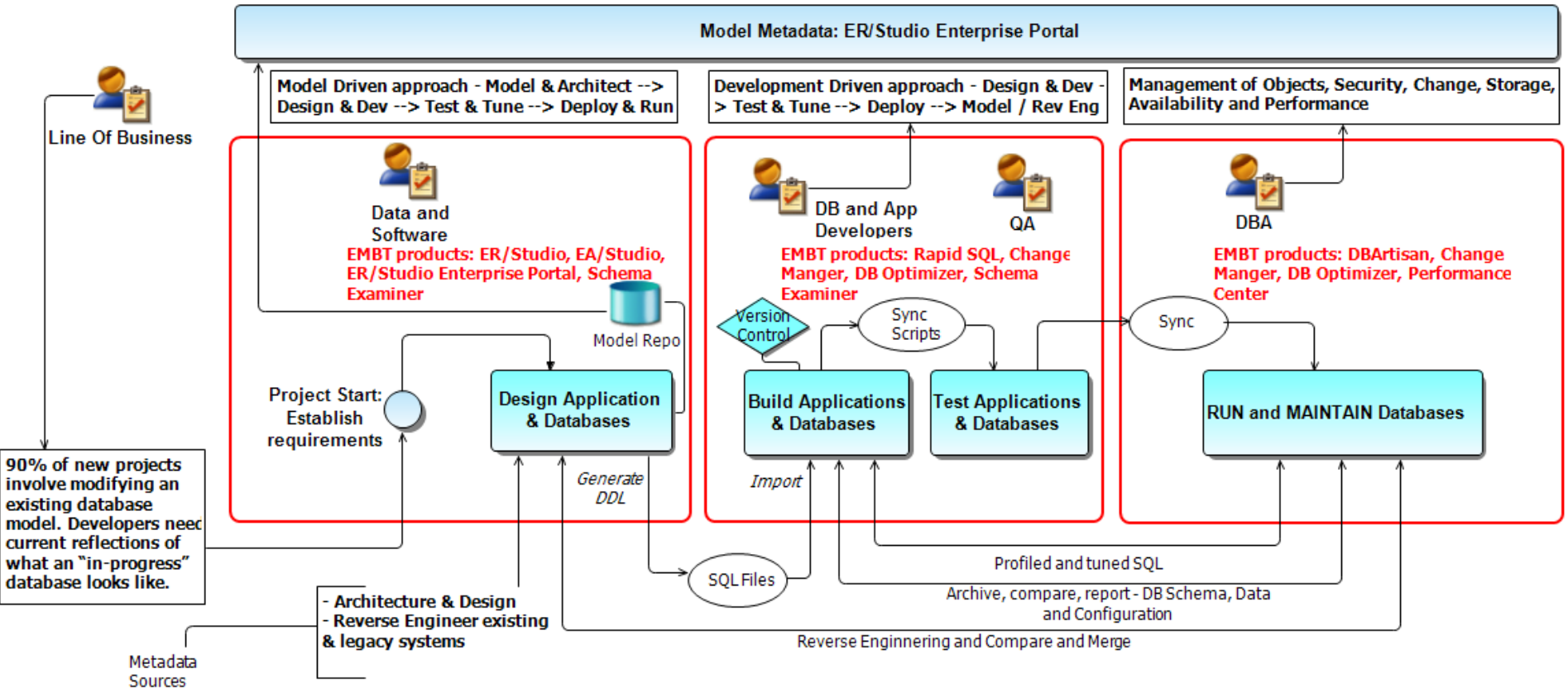
EMBARCADERO TECHNOLOGIES

- アーキテクチャ＆設計
  - **アプリケーションコード:** UML、ソフトウェアアーキオロジー／リバースエンジニアリング、可視化
  - **データベースコード:** データモデリング、リバースエンジニアリング、ドキュメント・モデルメタデータの分析
- コーディング＆デバッグ
  - **アプリケーションコード:** Java、Windows RAD、.NET、Ruby on Rails、PHPなど
  - **データベースコード:** ストアドプロシージャ・SQL
- パフォーマンスチューニング＆プロファイリング
  - **アプリケーションコード:** ボトルネックの検出・ 除去
  - **クエリーチューニング＆データベースチューニング:** SQL・データベース設計の最適化
- 変更管理
  - **アプリケーションコード:** ソースコードコントロール
  - **データベーススキーマオブジェクト:** スキーマの比較・同期
  - **データベースデータ:** データの比較・同期、データの移動・移行

エンバカデロのツールと プロジェクト ライフサイクル

EMBARCADERO TECHNOLOGIES

Software Development Life Cycle for Database Applications



データベース開発の解剖学

EMBARCADERO TECHNOLOGIES

データベースには以下が含まれます:

- DBMSソフトウェア
- 論理・物理ストレージと…
- データ

以下のような活動に対する要求に対応していかなければなりません:

- データベース設計
- データベース開発
  - SQL開発、クエリーチューニング、クエリー・オブジェクトに対する変更の管理
- データベース管理
  - バックアック・リカバリープランニング、パフォーマンスチューニング、キャパシティプランニング、データベース変更管理、セキュリティ／アクセスコントロールなど
- そしてSQL言語: DML, DDL, and DCL

データベースオブジェクト

EMBARCADERO TECHNOLOGIES

DB2 for LUW	Oracle	SQL Server	Sybase
Server Objects Buffer Pools Event Monitors Groups Node Groups Tablespaces Users **Database Objects** Aliases Check Constraints Foreign Keys Functions Indexes Materialized Query Tables Primary Keys Procedures Sequences Structured Types Tables Triggers Unique Keys User Defined Types Views	**Server Objects** Directories Groups Profiles Redo Log Groups Roles and Users Rollback Segments Tablespaces **Database Objects** Check Constraints Clusters Database Links Foreign Keys Functions Indexes Java Sources Libraries Materialized Views and Logs Outlines Packages and Bodies Primary Keys Procedures Sequences Synonyms Tables Triggers Types and Bodies Unique Keys Views XML Schemas	**Server Objects** Databases Dump (Backup) Devices Linked Servers Logins Remote Servers Server Triggers Synonyms User Messages **Database Objects** Asymmetric Keys Certificates Check Constraints Defaults Extended Procedures Foreign Keys Functions Indexes Primary Keys Procedures Roles Rules Server Triggers Symmetric Keys Tables Triggers Unique Keys Users User Defined Datatypes Views	**Server Objects** Data Caches Database Devices Databases Dump Devices Logins Remote Servers Roles **Database Objects** Aliases Check Constraints Defaults Extended Procedures Foreign Keys Indexes Logical Foreign Keys Logical Primary Keys Primary Keys Procedures Rules Segments Tables Triggers Unique Keys Users User Messages User Defined Datatypes Views

サンプルデモ

EMBARCADERO TECHNOLOGIES

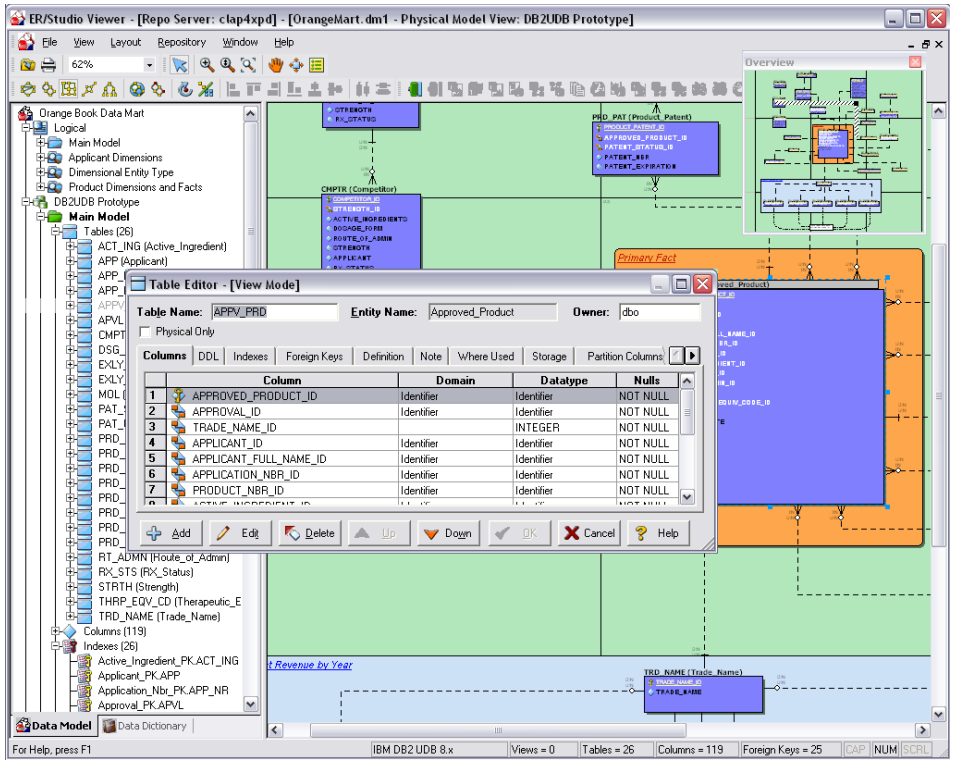
- ビジネスプロセスのドキュメント化
- テーブル変更のモデリングと実装
  - データベースのリバースエンジニアリング
  - モデルを使った変更
  - DDLの生成と実行
- データベースコード変更の実装
- SQLのチューニング
- マイグレーションと変更の収集

デモ

まとめ

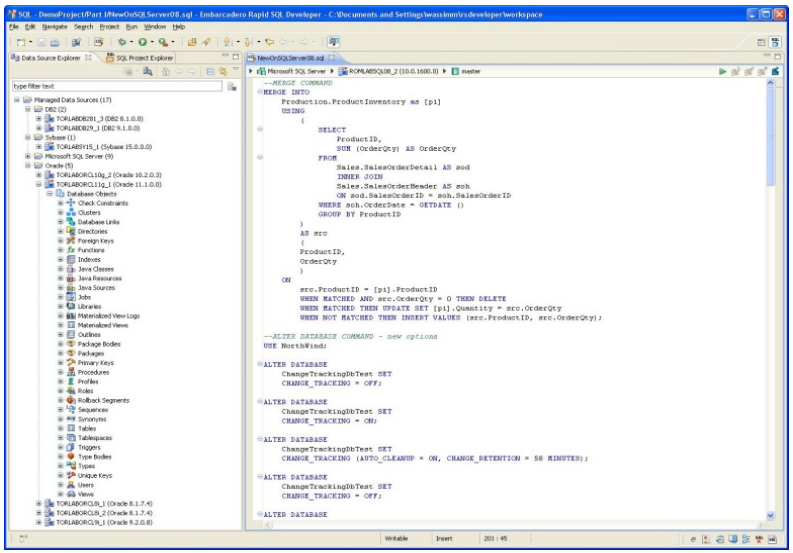
データモデリング

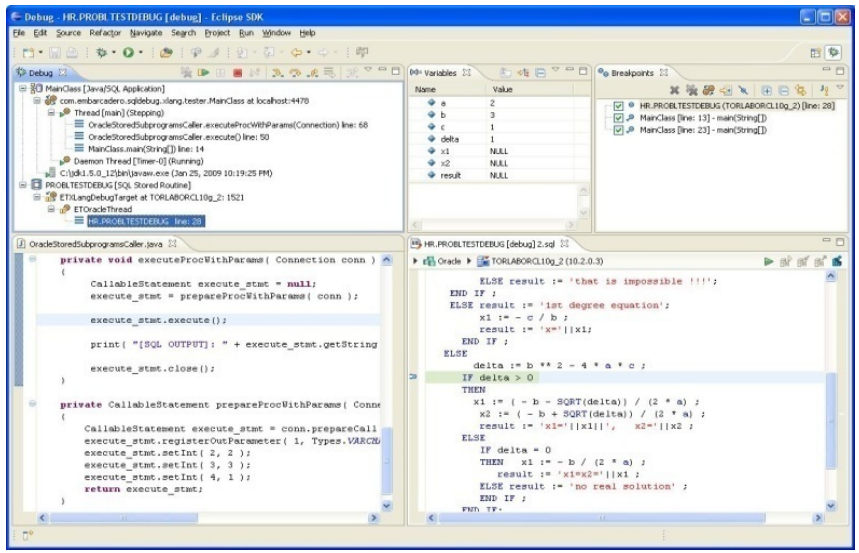
- リバースエンジニアリング・SQLの生成
- モデルによるビジュアル化
- コラボレーションモデル開発
- メタデータタグ・レポート
- レポート生成
- 論理／物理モデリング
  - DDL・XMLスキーマ生成
- ビジュアルデータリネッジ



データベース開発

- データベースコードの記述と実行（SQL、ストアドプロシージャ）
  - 実行計画、SQLの書式化、スタイル規約違反、ビジュアルクエリー構築、コード補完
- データベースコードのデバッグ
- データの編集と操作
- ソースコードコントロールとの統合
- オブジェクトの作成、DDLの生成

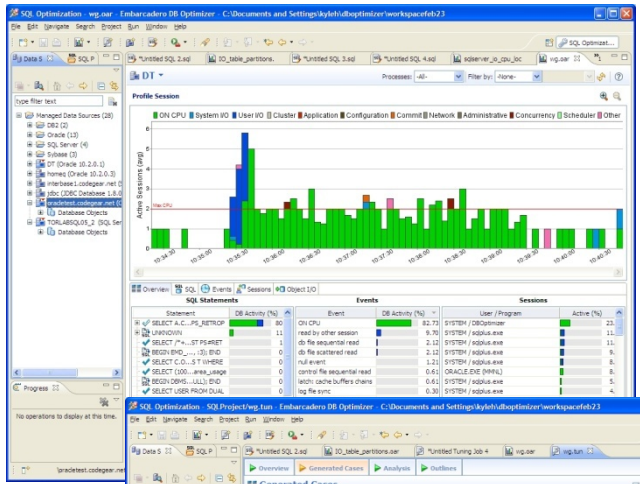




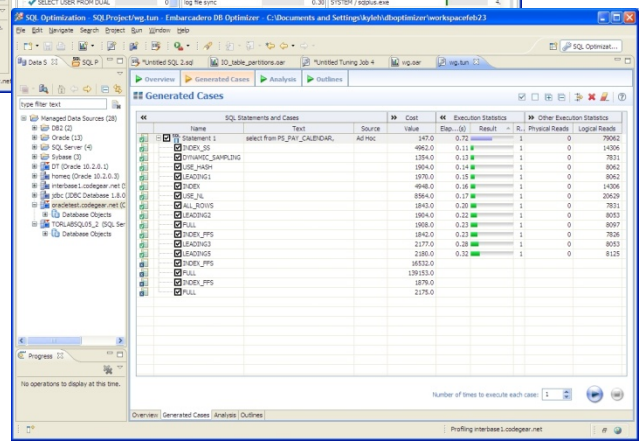
SQL の最適化

- **SQLプロファイリング**
  - 待ち時間分析をグラフィカル表示
  - 継続的なプロファイリング
  - 「右クリック」一発でSQL文のチューニング
  - 実行計画
- **SQLチューニング**
  - DML文のバッチチューニング
  - コスト生成
  - ケース生成
  - ヒントインジェクションとSQLの書き直し
- **SQLの編集**
  - リアルタイムクイック修正、コードアシスト、SQLの検査機能などを搭載したSQL IDE

SQL プロファイリング



SQL チューニング



データベース変更管理

EMBARCADERO TECHNOLOGIES

- 選択したスキーマオブジェクトのアーカイブスナップショットのキャプチャー
  - 時間軸での変更を理解
  - プロジェクトの進捗をドキュメント化するレポートの生成
- スキーマの比較と変更
  - 異なる環境間での比較、異なる時間での比較
  - 差分実装のためのDDLの生成
- ソースコードコントロールシステムとの統合
- データの管理
  - データ参照のための変更の管理と移行
  - テストデータの移行と比較
- 扱い注意データのマスク

まとめ: 開発者のための エンバカデロのデータベースツール

EMBARCADERO TECHNOLOGIES

- アーキテクチャ＆設計
  - **アプリケーションコード:** Delphi, C++ Builder, JBuilder, Delphi for PHP, 3rd Rail
  - **ビジネス・アプリケーションプロセス**: EA/Studio
  - **データベースコード:** ER/Studio, Rapid SQL, ER/Studio Enterprise Portal, Schema Examiner
- コーディング＆デバッグ
  - **アプリケーションコード:** Delphi, C++ Builder, JBuilder, Delphi for PHP, 3rd Rail
  - **データベースコード:** Rapid SQL
- パフォーマンスチューニング＆プロファイリング
  - **アプリケーションコード:** J Optimizer
  - **クエリーチューニング＆データベースチューニング:** DB Optimizer
- 変更管理
  - **データベーススキーマオブジェクト:** Embarcadero Change Manager
  - **データベースデータ:** Embarcadero Change Manager

Thank you.

Any questions, please contact Hitoshi Fujii: Hitoshi.Fujii@Embarcadero.com

Or me: Philip.Rathle@Embarcadero.com

Supplemental Slides

Database Change Management ~ Some Challenges

EMBARCADERO TECHNOLOGIES

- Preserve data when making a structural change
- Maintain separate storage settings for objects across environments
- Maintain security differences between environments
- Shared ownership responsibility for certain object types
- Keep physical and logical models in sync with the database
- Manage and report on database differences over time
- Manage reference data across many environments
- Validate the synchronicity in a replicated environment
- Support for parallel development efforts
- Accommodate both planned and unplanned changes
- Manage reference data across many environments

Typical Change Management Process

1. Identify the need for change
2. Log the change
3. Analyze the impact
4. Design the change
5. Codify the change
6. Test the change
7. Schedule the change
8. Roll out the change
9. Communicate the change
10. Monitor the change
11. Close the change ticket