embarcadero
Developer Camp

【T2】データベースアプリケーション開発

「アプリケーション開発者のための、DBべからず集からパフォーマンスチューニングまで」

株式会社ドリームハイブ
代表取締役　山本　悟
株式会社日本情報システム
筑木　真志
アナハイムテクノロジー株式会社
代表取締役　はやし　つとむ

自己紹介

- 名前：山本 悟 (やまもと　さとる)
  - 代表取締役 & 不動産投資専門ITコンサルタント
- 会社：株式会社　ドリームハイブ
  - 会社URL　　：http://www.dreamhive.co.jp/
  - お得なコンテンツ配信サイト　　：http://dhive.jp/
  - 山本のブログ　：http://dhive.jp/blog/yama/
- 略歴：
  - 17歳からこのIT業界へ。
  - Delphi は1.0からの親友。
  - 経営に加え、システム開発とITコンサルティング、スピーカーなどが主な仕事ですが、テレビ埼玉に出たりしたこともあります。

自己紹介

- 名前：筑木　真志　(ちくぎ　しんじ)
  - 多分、同姓同名はいません
- 会社：株式会社　日本情報システム
  - URL　：http://www.nihon-jyoho.com
  - E-MAIL　：nis@nihon-jyoho.com
- 会社概要：
  - 土木・建築・測量系の開発が多い
  - GISとかDBを活用したシステムなんて大好きです
  - でも、会計とか集計処理などOSや言語問わずいろいろやっています
  - GIS関連スクールの講師とかもやることがあります
  - あと、レガシーシステムのお守りとか移行サポートとか
  - ぶっちゃけ、IT系の「大工さん」というか「何でも屋さん」
  - だから、道具にこだわる故に自分はC++Builderが大好き!!

自己紹介

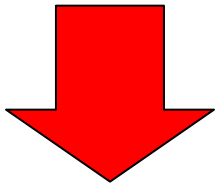
- 名前：はやし　つとむ　(林　務)
  - 代表取締役　＆　Firebird日本ユーザー会理事
- 会社：アナハイムテクノロジー株式会社
  - URL　　：http://www.anaheim-tech.com/
  - メール　　：tsutomu.hayashi@anaheim-tech.com
  - Twitter　　：tomneko_p
- 略歴：
  - 建設・不動産の会社、株式会社アペックスの専務も兼務。
  - 2003年にFirebird日本ユーザー会を立ち上げた張本人。
  - 各地のOSC等で盛り上げ役をやってます。
  - 名前は漢字でググると全く出てこないので、ひらがな表記です。（笑

embarcadero
Developer Camp

移行でのケース
その1

古いDelphiから新しいDelphiにマイグレーション

- 導入先のハードウェアのリプレースに伴い
  OSのバージョンが変わった。
- システムを利用するユーザー数が増加した
- とりあえず、今の構成のまま単純にマイグレーション



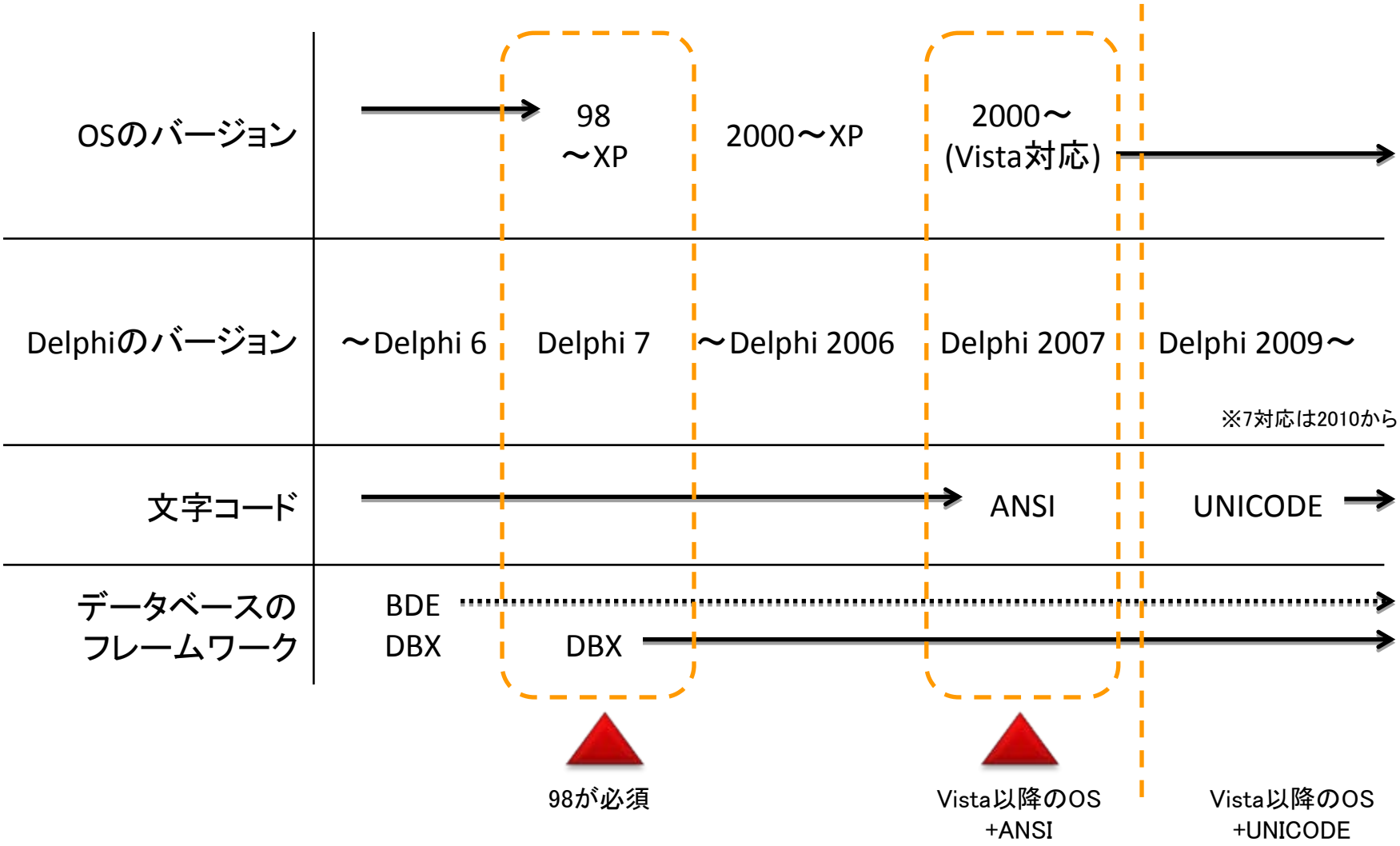
- BDEのエラーが出た？
- 複数人で使おうとしたらエラーが出た？

いきなり新しいDelphiに移行しようとしてません？

- 移行はステップバイステップで！
  - 様々な要因でステップが変わります
    - OSのバージョン
    - Delphiのバージョン
    - 文字コードの種類
    - データベースのフレームワーク
  - 基本はDelphi 2007への移行を挟むこと
- 複数の動作環境を準備しておきましょう！

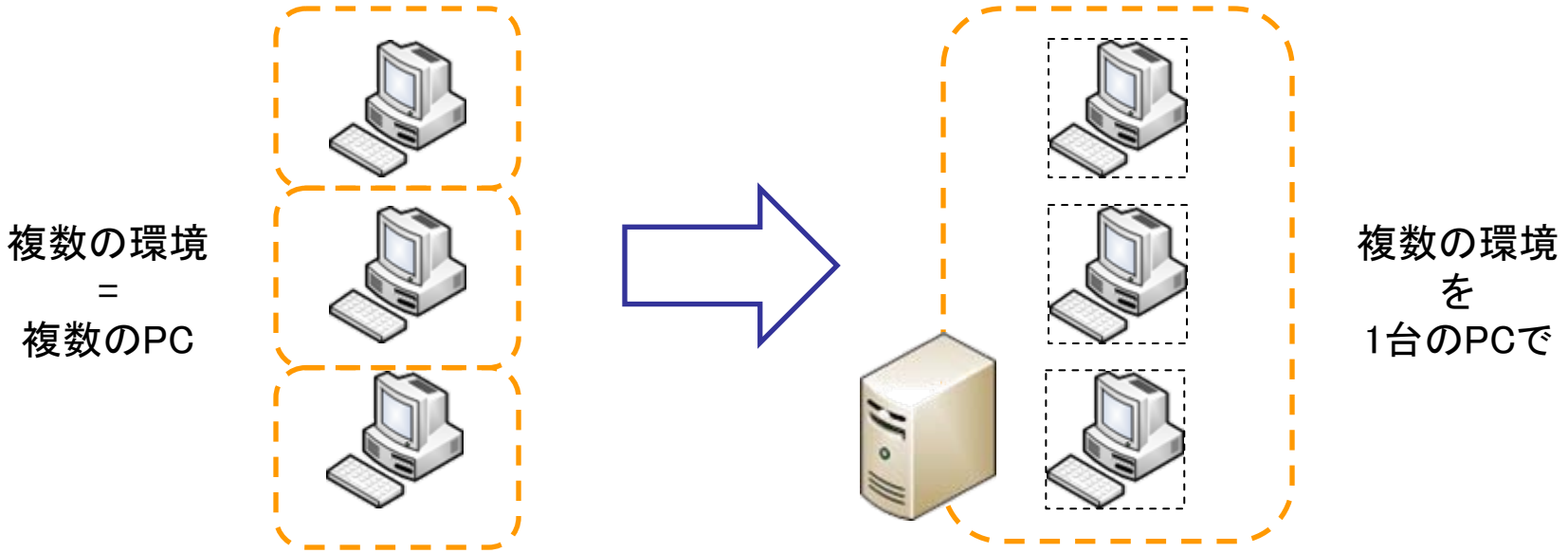
いきなり移行するのはやめよう！
→多くの場合で工数が増えます

移行の基準


OSのバージョン
98
～XP
2000～XP
2000～
(Vista対応)
Delphiのバージョン
～Delphi 6
Delphi 7
～Delphi 2006
Delphi 2007
Delphi 2009～
※7対応は2010から
文字コード
ANSI
UNICODE
データベースの
フレームワーク
BDE
DBX
DBX
98が必須
Vista以降のOS
+ANSI
Vista以降のOS
+UNICODE

開発環境はどうするのが良い？

- 仮想環境を構築するのがオススメ
  - 仮想化：1台のコンピュータを、あたかも複数台のコンピュータであるかのように論理的に分割し、それぞれに別のOSを動作させることで、複数の環境を少ないリソースで比較的安全に構築することができる
    - Hyper-V　　：マイクロソフト
    - VMWare　　：ヴイエムウェア
    - Xen　　：シトリックス・システムズ・ジャパン
    - VirtulBox　　：Oracleなど



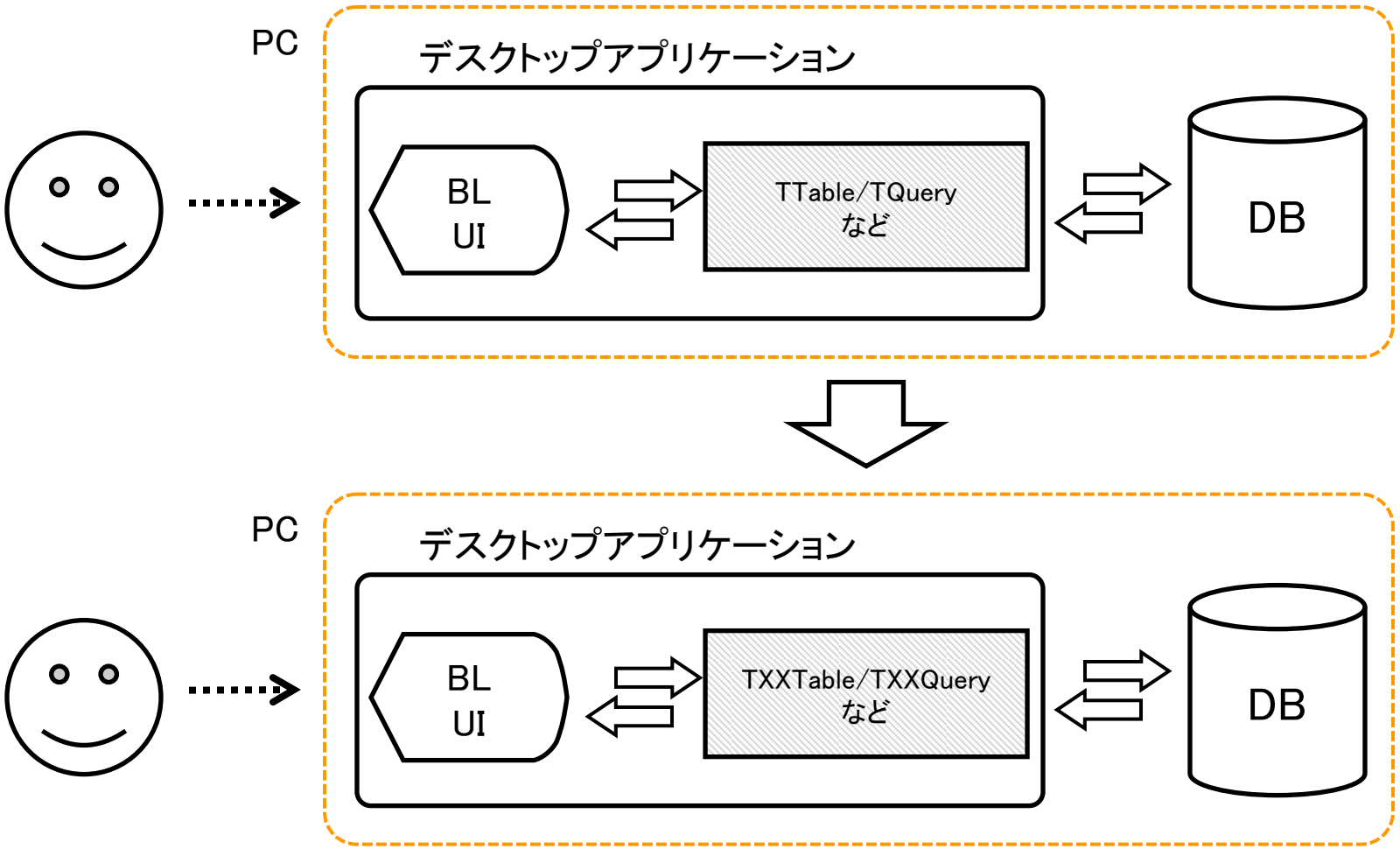
BDEをそのまま使おうとしていませんか？

- BDEはもうダメ！いやマジでマジで
  - Delphi 7附属のBDE5.2で開発およびメンテナンスが終了してるので、正規サポートは望めない
  - XEに付属しているBDEもDelphi 7 Professionalレベルのものであり、Enterpriseエディションの SQLLinkは含まれていない)
  - あくまでも、過去の資産(dBase/Paradox,ODBC接続)の保守用
- せめて3rdパーティの互換コンポーネントを利用しましょう

BDEを残すのはやめよう
→何かあったら開発者の責任！

BDEアプリは互換コンポーネントで置き換える

BL:ビジネスロジック
UI:ユーザーインターフェイス



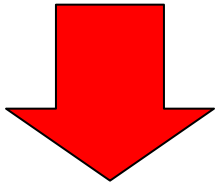
※移行はとても簡単だけど、
基本的にはユーザーが単数の時だけにしましょう

embarcadero
Developer Camp

移行でのケース
その2

古いDelphiから Webに対応する

- より広いネットワーク対応を求められた
  - ディスクトップからイントラネット
  - イントラネットからインターネット



- Professional版でリモート接続ってOK？
- パフォーマンスが超低下

コネクション数を考えてます？

- ローカル or リモート
  - そもそもPro版はリモート接続できないので、ライセンス違反です
- コネクション数が増えれば、
  - パフォーマンスが悪くなりがちです
  - 整合性が取りづらくなるはずです
    - Accessとかのクラサバだと最悪です
- 自分でどうにかするよりEnterprise版買う方が、速くて楽で安くて安心です

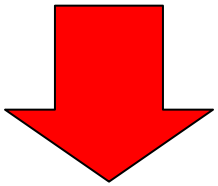
コネクション数を考えないのは
止めよう！
→規模が大きくなるにつれ、
問題の切り分けがしづらくなります

embarcadero
Developer Camp

移行でのケース
その3

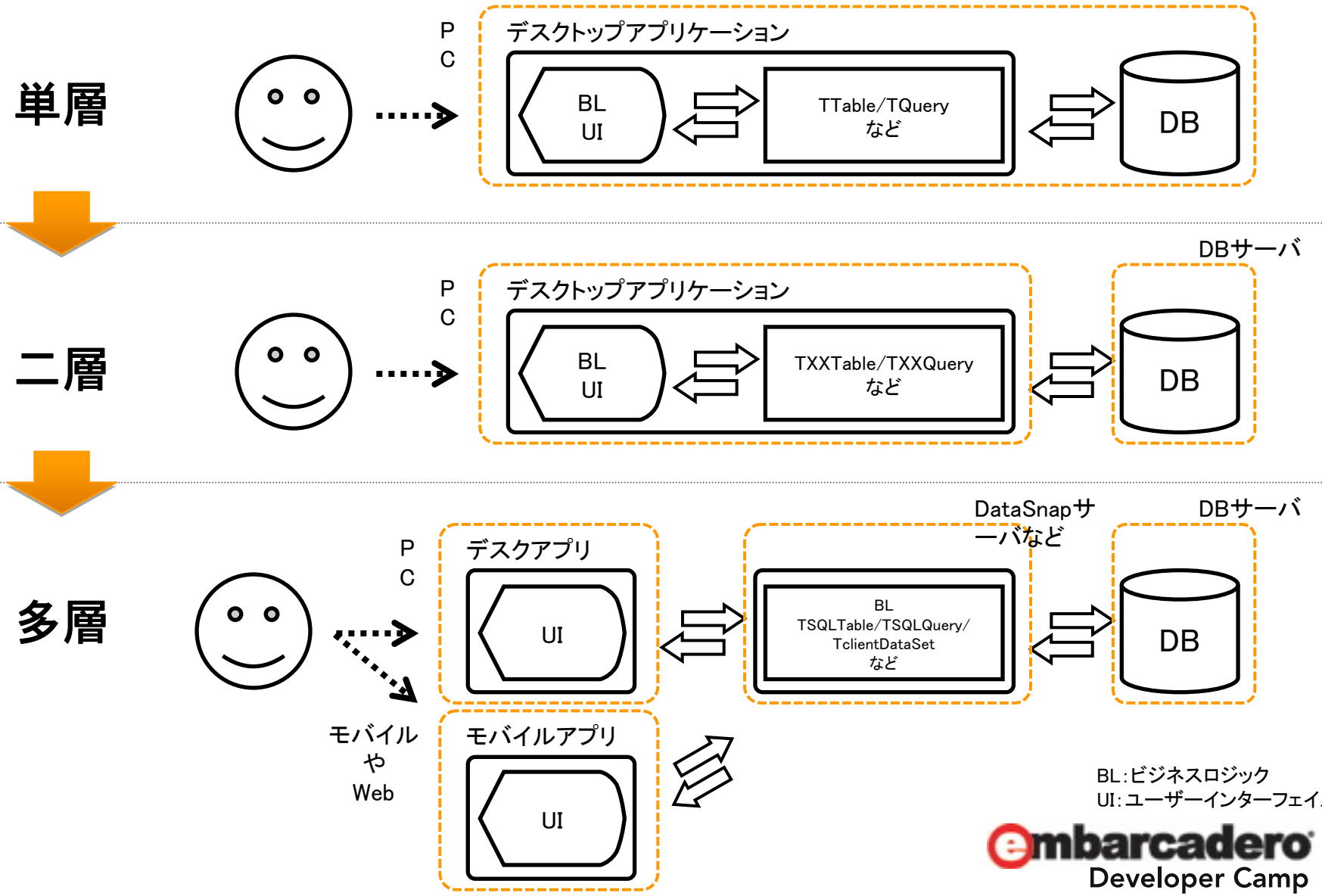
古いDelphiから多層アプリケーション化

- 業務拡大によりクライアントを増加
- 単純にクライアントアプリケーションを配布



- パフォーマンス的にどうなの？
- 低減するはずの工数が何故か増加

多層化がイントラ/インターネット対応のカギ



単層のみで作っちゃってないですか？

- クライアントの台数は？
  - クライアント1台だけにしか対応しない、いわゆるデスクトップアプリケーションのときのみ単層構造でも良いでしょう
- でも、
  - dbExpress(DBX)を使って多層構造に対応しておきましょう
  - DataSnapなどを使えば後の拡張が容易です

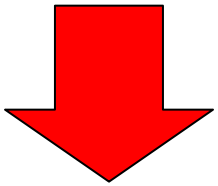
単層構造のアプリで居続けるのはやめよう！
→後がもっと大変になります

embarcadero
Developer Camp

新規開発でのケース その1

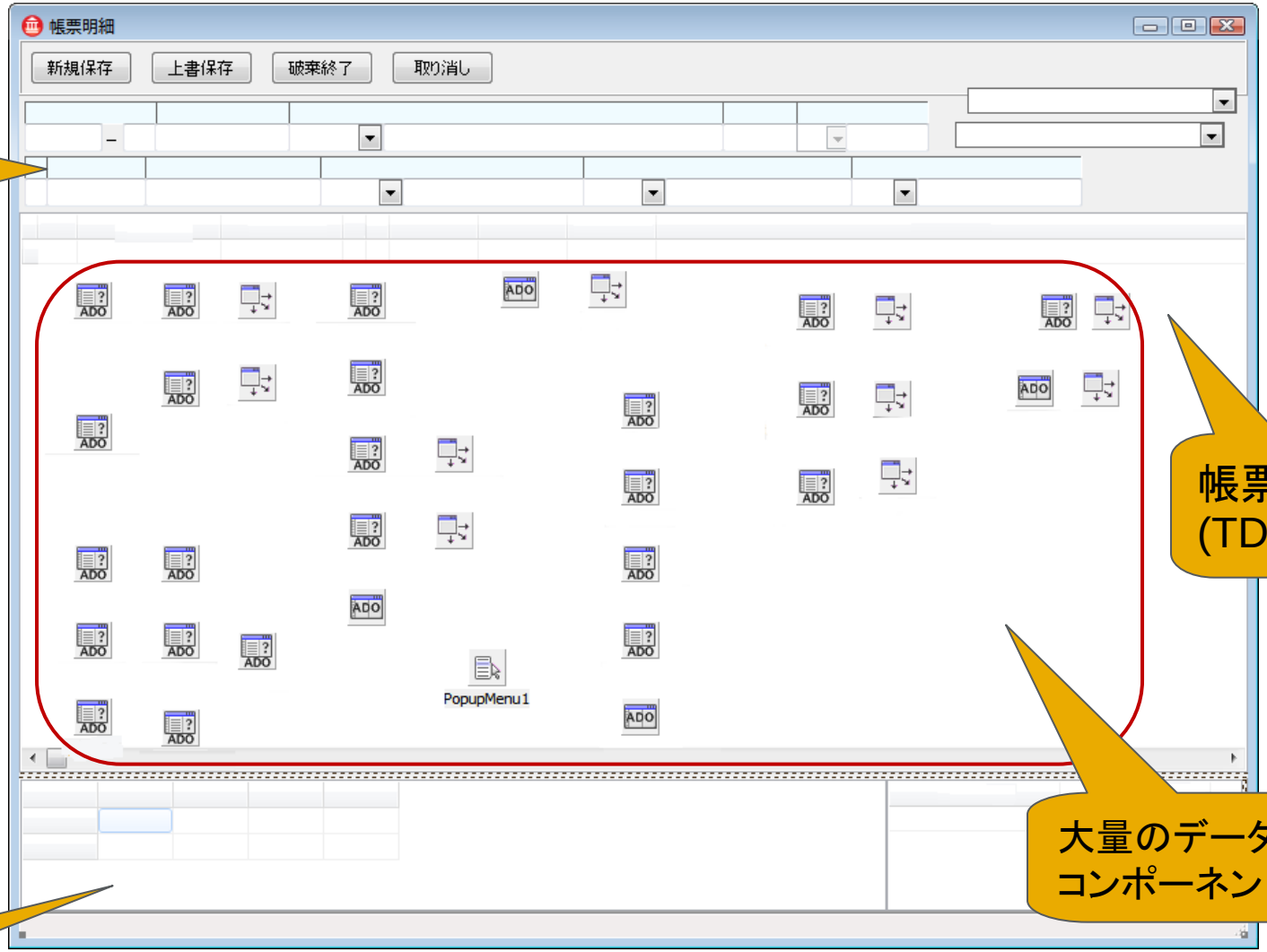
リッチすぎたUI

- 顧客からの要求
  - 帳票のヘッダーと明細は同一画面で表示/編集
  - Accessのように項目番号と項目名を一緒に表示
  - 項目番号が判れば、項目番号を直接入力したい
  - 項目名の内容を直接入力して絞り込みもしたい
  - Excelのように表形式で編集
  - 編集中の数値はリアルタイムに集計、集計値のチェックをしながら入力したい



- アプリケーションのパフォーマンスが超悪い(その正体は次のページで…)

結局、こんなことをやってしまいました



帳票ヘッダ

帳票明細
(TDBGrid)

大量のデータベース
コンポーネント

集計結果

データベースコンポーネントの貼りすぎ

- レコードの取得(SELECT)と更新(UPDATE/INSERT)が別のデータベースコンポーネント
- 違うフォームでも同じテーブルを参照するデータベースコンポーネントを貼っている
  →テーブルの仕様が変わった場合は？

- データベースコンポーネントは、データモジュールに分離して一元的に管理しよう！

データコントロールコンポーネントの貼りすぎ

- 一つのフォームにそれだけの情報が必要なの？
- ステータスコードなどの各種マスターの参照にTJvDBLookupComboEditを使用している
  →常にクエリが発生する

- ヘッダと明細はフォームを分けるなど、ひとつのフォーム上に必要な情報は何かを見極めよう！

データの絞り込み

- 明細情報をTADOTableで参照し、必要なレコードをFilterプロパティで絞り込んでいる
  →テーブルコンポーネントは
  全てのレコードをフェッチする。

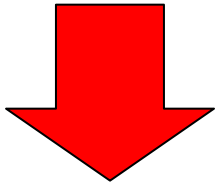
- Queryコンポーネントを使用して、取得するデータをまず限定！

embarcadero
Developer Camp

新規開発でのケース その2

そんなデータベースで大丈夫か？

- 大丈夫だ、問題ない
  - ということで、お手軽で使いやすいAccessを選択
  - 統計データを処理
    （データベース的な意味で）たった、数万レコード



- CPUパワーは問題ないのに、フェッチ＆集計で数時間

データベースの選択はさ…

- 本当にそのデータベースでよかったの？
- リモート接続する？しない？

- データベースの規模を把握して規模にあったデータベースを選ぼう！
- データベースの設計も大事！

embarcadero
Developer Camp

DB設計 基本の「キ」
「主キー」

主キーの無いテーブルなんて、●●の無いコーヒーのような・・

- 商用の業務系システムでも、データベースに主キーがないケースが見受けられる

主キーがないテーブルなんて
あり得ない！

主キーに使うデータ型

- smallintの主キーで一ヶ月で止まったシステムがあった（僕じゃないよ）
- 現状では64bit整数などを使う Bigint、int64
- クラスタ化するときは必要なカラムの複合インデックスなどで構成すべき

主キーをSMALLINTにしない

主キーをクラスタ化する？

- Oracle、MS SQL、DB2、Sybase、MySQL、PostgreSQLではクラスタ化インデックスを作成出来る
- クラスタ化すると、そのインデックスでのアクセスは高速化されるけど、他のインデックスでのアクセスは遅くなる
- テーブルの更新も遅くなる可能性がある

なんでもクラスタ化しない

主キーをコードに使ってる？

- 主キーは一意であればいいので、seaquenceなどを利用
- コードは人間が見て分かる意味のある数字や文字にする
- インデックスのサイズを小さくするとインデックススキャンが速くなるので、文字型の主キーはやめよう

主キーをコードに流用するのはやめよう

embarcadero
Developer Camp

DB設計 基本の「ホ」
「インデックス」

インデックス有りますか？

- RDBMSはインデックスの固まり
- なんか遅いと思ったら、インデックスが無い／使われていないことがほとんど
- DBで一番遅いのはディスクI/Oなので、これを減らすためにインデックスを使う

インデックスがないDBなんて意味がない

そのSQL、PLAN確認しました？

- PLAN＝実行計画、どのテーブルにどうアクセスしてデータを取るか表示される
- PLANは各DBのバージョンによっても変わる
- 考えてるのはオプティマイザ、ルールベース／コストベースがあり、ヒントも指定出来る

PLANを確認しないでSQLを発行しない

そのインデックス使われてますか？

- テーブルの特定のカラムを検索条件にして、そのカラムにインデックスがあっても使われないことがある
- SELECT * FROM TBL WHERE SEQ_NO=10 というような単純なクエリーでも選択ミスが起きる可能性
- ヒントを与えて、インデックスの使用を促すことも必要

インデックスを作成すれば速くなるわけではない

テーブルコンポーネント、使ってませんか？

- ちゃんとクエリーを書いて、検索条件を指定して、データセットを小さくしよう
- 適切にインデックスを張って使用して、小さいデータセットを使う事

Delphi的にテーブルコンポーネントだと、丸ごとアクセスなので、RDBMSの意味がない

データベースのメンテしてますか？

- InterBase/Firebird,PostgreSQL等のバージョニングデータベースでは、テーブルデータの定期的なメンテナンスをしないとインデックススキャンでも無駄が発生します。
- InterBase/FirebirdではSweepやバックアップ＋リストア、PostgreSQLではVacuumを定期的に実行しましょう

バージョニングを採用したDBでは定期メンテしないと遅くなるかも

embarcadero
Developer Camp

DB設計 基本の「ン」
「ジョイン」

テーブルのジョインは出来るだけ結果セットを絞る

- 基本的にクロスジョインしない
- SELECT T1.A, T2.B FROM T1, T2 とかすると、1000行と1000行でも百万行になる

クロスジョインはやめよう

OUTER JOIN ではNULLが発生する？

- LEFT/RIGHTのOUTER JOINでは、結合先のテーブルにない値がNULLになる
- 主キーをSELECTするようなケースでもNULLになることがあるので注意

OUTER JOIN後にはNULLが無いと思い込まない

そのカラム、外部キー設定されてる？

- インデックスがあればジョインは速くなる可能性がある
- 外部キーがあるとテストしずらい場合もあるので、その場合は二次インデックスでもよい

ジョインで使用するカラムにインデックスが無い！

暗黙の型変換、信用してる？

- ジョインするキーで文字型と数値型の比較はしないと思うが、そういうケースもある
- 明示的にキャストして、暗黙の型変換を信用しないこと

暗黙の型変換を信用しないこと

沢山のテーブルをジョインするとわけがわからない？

- その為にビューを使ったりもするが、ビューではインデックスが使われないケースがある
- マテリアライズドビューを使ったりすると、こんどはディスクとメモリを食う
- 手続き型で記述できるストアドの利用も検討した方がよい

SELECT文だけで解決しようと思わないこと

embarcadero
Developer Camp

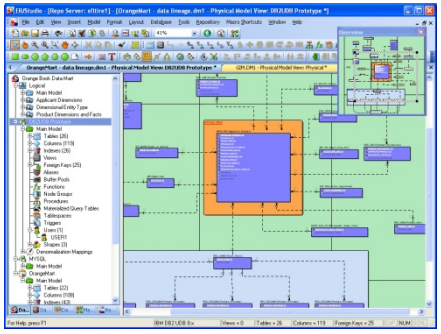
データベース問題の解決

エンバカデロのデータベースツール

DB設計に関わる問題の解決には

ER/Studio

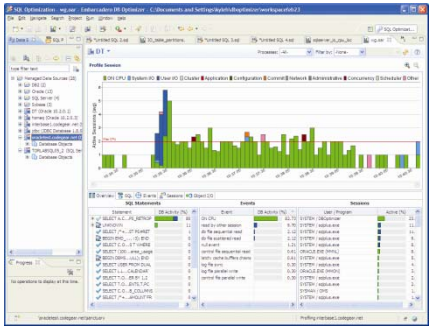
確実なデータ設計を実現。正規化や異なるDBへのモデルの移行なども支援



SQLパフォーマンスの問題解決には

DB Optimizer

SQLパフォーマンスのボトルネックを検出し修正。経験やカン頼みの作業から脱却



データベースの管理やデータ移行については

DBArtisan

DBAの日々DB管理業務をサポート。異なるDBへのデータの移行にも対応

embarcadero
Developer Camp

べからず Know-how 集

移行でも新規開発でもDB注意点は同じ

- DBのエンジンを確認するべし
  - Microsoft Access / SQLite / InterBase / Firebird / MySQL / PostgreSQL / Oracle / Microsoft SQL Server / DB2 / Sybase ..etc
  - Accessを共有しての似非クラサバは絶対禁止
  - 条件つきだけど Express版はとても便利
  - アプリケーションの「規模」に応じたエンジンを
- DBへのアクセスを確認するべし
  - BDE / dbExpress / dbGo (ADO) /ODBC / API直叩き / 埋め込みSQL ..etc

Professional版でのdbExpessについての注意

- EULAの追加条項によってリモート接続はできない
  - MySQLやInterBaseのドライバは同梱されているけど ローカル接続に限定される
- 解決法 その1:
  - 素直にEnterprise版以上にする
    - OracleやMSSQL、DB2のドライバもあるよ！
- 解決法 その2:
  - Professional版ではdbGoが事実上唯一の選択肢
    - BDE？なに、それ？

Access便利ですよね…

- とにかく手軽で使いやすい！（SQLビューを除く)
  - VB / Delphi / C++BuilderのローカルDBとしても便利
- でも、統計データを処理すると半端無いです…
  - カラム数：200カラム以上
  - レコード数：数万行
  - *.accdbのサイズ：350MB(!)
- フェッチ＆集計で数時間かかりました。(本当)
  - CPU:C2Q Q9550 メモリ：8GB OS:Vista x64
  - 同じデータをSQL Server Expressに処理させたら10数分

Express版ってなに？

- SQL ServerとDB2にはExpress版があります。
  - サポートがないけど無償(!)
  - エンジンは製品版（ほぼ）と同じ
  - 当然、dbGoやdbExpressで接続できる
- Microsoft SQL Server 2008 Express Edition
  - http://www.microsoft.com/express/Database/
- DB2 Express-C
  - http://www-06.ibm.com/software/jp/data/db2/v9/express-c/

Express版を選択する場合の注意点

- 使用できるリソースに制限
  - SQL Server 2008 Express
    - CPU1コア、メモリ1GB、サイズ10GB
  - DB2 Express-C 9.7
    - CPU2コア、メモリ2GB、サイズ無制限
- サポート、保守一切無し
- 画像ファイルを扱うと、すぐにリソースを使い切る
- あくまでも「簡易版」

移行での注意点

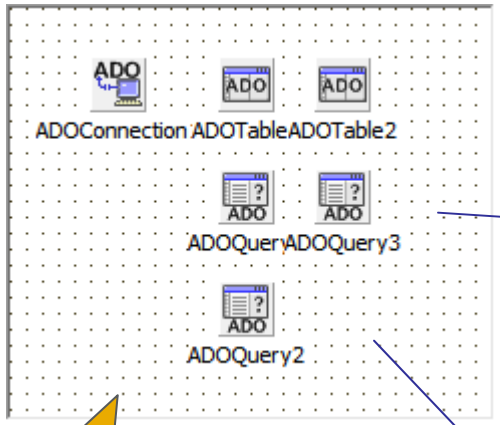
- 移行の範囲を明確にするべし
- 早めにリファクタリングするべし
- 複数バージョンのテストができる環境を用意するべし
- 時は金なり。時間は金で買え。
- やっぱ Enterprise版でしょう

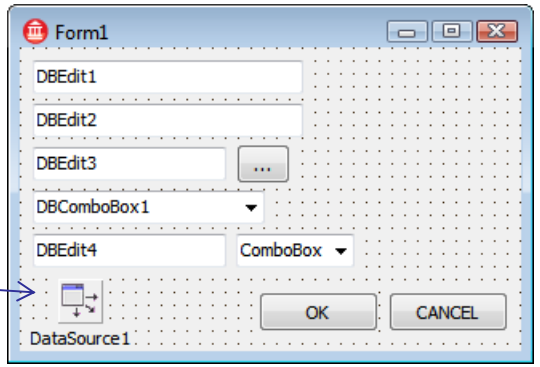
新規開発時の注意点

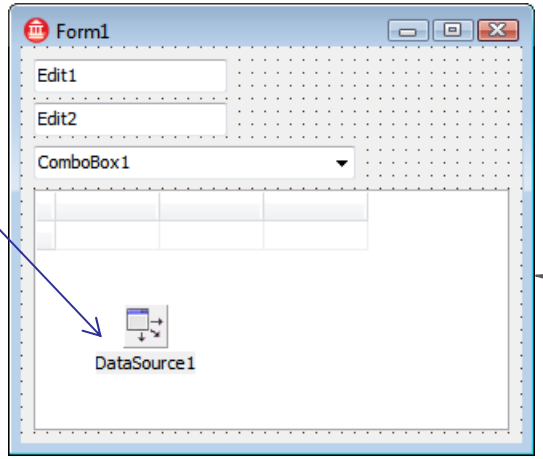
- アプリケーションの設計はしっかりと
- 時は金なり。時間は金で買え。
- やっぱ Enterprise版でしょう
- データベースの設計も大事！

パフォーマンスを上げるには…

- データベースコンポーネントは、データモジュールに分離して、一元的に管理する
- 「分割して統治せよ」の原則





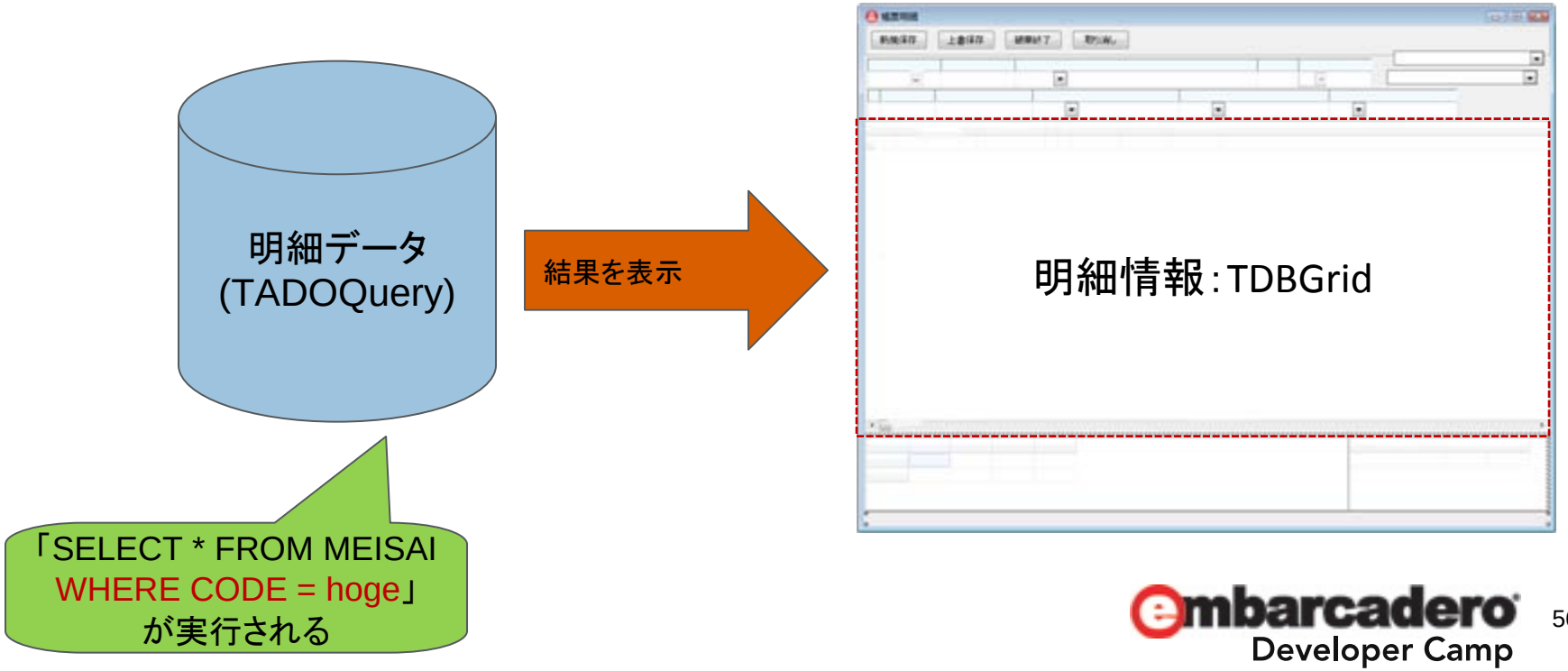


データベースコンポーネントは
TDataModuleで一元管理

フォームは
TDataModuleを参照する

パフォーマンスを上げるには…

- クエリコンポーネントを使用して、取得するデータを限定
  - 取得するデータセットが小さくなる
  - テーブルコンポーネントは全レコードを取得する
  - クエリコンポーネントでもレコードの追加、削除が可能



パフォーマンスを上げるには…

- データコントロールコンポーネントの使用を可能な限り減らす
  - ヘッダと明細とでフォームを分ける
  - ステータスコードのようにあまり更新しないデータは、OnCreateイベント内で先読みしてキャッシュする

embarcadero
Developer Camp

Q & A