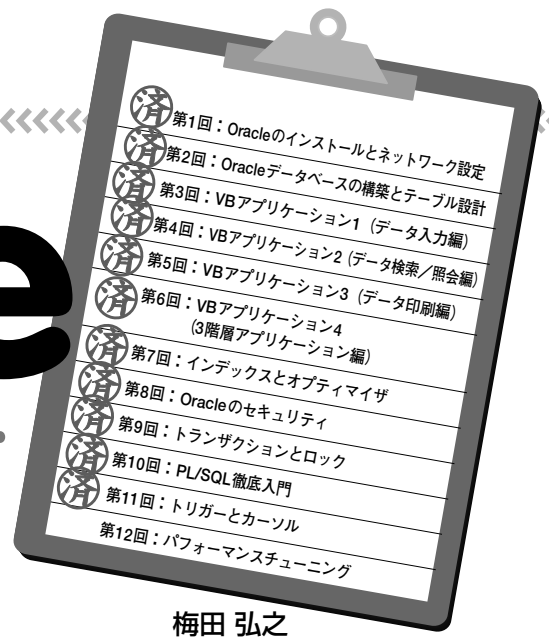


# Oracle

## プログラミング教習所

### 最終回 パフォーマンスチューニング



|      |                            |
|------|----------------------------|
| 第1回  | Oracleのインストールとネットワーク設定     |
| 第2回  | Oracleデータベースの構築とテーブル設計     |
| 第3回  | VBアプリケーション1 (データ入力編)       |
| 第4回  | VBアプリケーション2 (データ検索/照会編)    |
| 第5回  | VBアプリケーション3 (データ印刷編)       |
| 第6回  | VBアプリケーション4 (3階層アプリケーション編) |
| 第7回  | インデックスとオプティマイザ             |
| 第8回  | Oracleのセキュリティ              |
| 第9回  | トランザクションとロック               |
| 第10回 | PL/SQL 徹底入門                |
| 第11回 | トリガーとカーソル                  |
| 第12回 | パフォーマンスチューニング              |

梅田 弘之  
UMEDA, Hiroyuki

みなさん、「Oracleプログラミング教習所」へようこそ。1年間に渡って学んだ「Visual Basic 6.0 + Oracle 講習」も今回が最後です。

最終回は、Oracleの“パフォーマンスチューニング”について、基礎と実践を学びます。これまでに学んだことを復習しながらの総まとめとします。パフォーマンスの良いアプリケーション構築技術をしっかりと身に付けて、本教習所から飛び出してハイウェイをぶっ飛ばしましょう。



#### 学科：Oracleのチューニングステップを理解する

みなさんは、プログラム開発時点において大量テストデータを用意してパフォーマンステストを行なっていますか？ 「本当はやらなければいけないと思うのだけど余裕がなくて……」「パフォーマンステストを行なう環境がないので、本番マシンが入ってから直接現地で……」「本番データを移行した後で確認しようかと……」などと“やらない理由”も聞こえてきそうですね。

従来のアプリケーション開発では、このような“やらない風習”がまかり通っていました。しかし、現代におい

ては、開発時点でのパフォーマンステストは必須項目です。とくにOracleなどのRDBMSを使ったシステムでは、データ量が大きくなりがちです。最近では、広域ネットワークにおける利用もごく普通に行なわれているので、パフォーマンス問題が生ずる可能性が非常に高くなります。また、不特定多数のユーザーが特定時間にアクセスする可能性のあるWeb系のアプリケーションでは、パフォーマンスを十分考慮した仕組みにしておかないと重大なトラブルを引き起こしかねません。

そこで最終回は“パフォーマンスチューニング”をテーマに卒業教習を行ないます。Oracleのパフォーマンスチューニングというと、内部パラメータの調整やSQL文の書き方を主体とした狭義の技術説明が多いようです。しかし、実はパフォーマンスはシステム全体で設計段階から考えなければ大きな効果が得られません。その方針を基に今回は、Oracleを使ったアプリケーション作成時に気を付けなければならないポイントを、システム構築の流れに

#### 本稿で前提となるもの

|      |   |                               |
|------|---|-------------------------------|
| O    | S | Windows NT4.0 (SP6a)          |
|      |   | Windows 2000 (SP3)            |
|      |   | Windows XP Professional (SP2) |
| 開発環境 |   | Visual Basic 6.0 (SP5)        |
|      |   | Oracle8/9i                    |



沿って説明することになります。

## Oracleのチューニングステップ

「まず、プログラムを作成して、パフォーマンスは最後にチューニングする」というアプローチが正しいと思う人は手を上げてください。あれ、誰もいないですね。では、「自分はそういうアプローチはしていない」と自信をもって言える人は何人いますか？ おや、数人しかいませんね。つまり“わかっているけど、実践できていない”ということなのです。

でも、本当に“わかっている”のでしょうか。普段やっていないことは、実際にやろうとしてもなかなか難しいものです。そこで、どんなアプローチでアプリケーション開発を行えば、パフォーマンスの良いシステムが構築できるかを一緒に考えてみましょう。

図1は、Oracleのマニュアルで推奨しているパフォーマンスチューニング（以下チューニング）ステップを少シア

レンジしたものです。Oracleシステムを構築する際に必要となるチューニング作業は、大きく分けると「設計／開発段階」と「運用段階」になります。ここでの設計／開発段階と運用段階の区別は、開発マシンから本番マシンに切り替えた時点を想定しているのです。本番環境におけるテスト作業から運用段階に含めています。以下、各ステップの具体的な内容について説明しましょう。



## 学科：設計／開発段階におけるチューニング



### 業務規則／フローのチューニング

システムのパフォーマンスを考える前に、業務規則／フローにおけるパフォーマンスを考えてみましょう。ここでのパフォーマンスは、更新処理の“速さ”ではなく、業務の“効率化”ということがモノサシとなります。プログラミング段階より前の要求分析／基

本設計フェーズで行なうことになりませんが、ちょっとした配慮が大きな成果につながるポイントなので押さえておきましょう。

### ①業務の流れをスリム化する

新しいシステムを構築するときには、既存の業務運用ルールをヒアリングしながら業務フローを作成します。このとき大切なことは、既存の業務の流れをそのまま転記するのではなく、常に「こうすれば業務が効率化できる」という改善意識を強くもつということです。ユーザーは、長いこと既存の業務の流れに沿ったスタイルで業務を行なってきたので、それが当たり前と感じている場合があります。しかし、実はその作業の中に“無駄な作業”や“省略できる作業”はたくさんあるものです。「省ける処理は省く」という当たり前のことを、きちんと提案しながらユーザーに喜んでもらえるシステムを作りましょう。

具体的な例として、「交通費精算」のシステム化を考えてみましょう。ある会社では、交通費精算用紙に担当者が記入し、上司の承認印をもらってから事務担当者が端末に入力するという仕組みになっていました（図2-①）。しかし、この方法だとユーザー全員が毎回計算を行わなければならないので、計算ミスも多かったため、Excelの得意な社員が「交通費精算書」をExcelシートで作ってくれました。事務担当者が精算書を会計システムに入力する手間は変わらないのですが、交通費入力時の計算と事務担当者のチェックが非常に簡単になりました（図2-②）。

図1：Oracleのチューニングステップ

