

## 目次

<b>第1章</b>	<b>データベースを制する者はシステムを制す</b>	<b>1</b>
1-1	システムとデータベース	3
	データ処理としてのシステム	3
	データと情報	4
1-2	データベースあれこれ	6
	データベースの代表的なモデル	6
	DBMSの違いは設計に影響するか?	9
1-3	システム開発の工程と設計	11
	システム開発の設計工程	11
	設計工程と開発モデル	12
1-4	設計工程とデータベース	15
	DOAとPOA	15
	3層スキーマ	18
	概念スキーマとデータ独立性	20
	演習問題	25
<b>第2章</b>	<b>論理設計と物理設計</b>	<b>27</b>
2-1	概念スキーマと論理設計	29
	論理設計のステップ	30
	エンティティの抽出	30
	エンティティの定義	31
	正規化	32
	ER図の作成	32
2-2	内部スキーマと物理設計	34
	物理設計のステップ	34
	テーブル定義	35
	インデックス定義	35
	ハードウェアのサイジング	35
	ストレージの冗長構成	41
	ファイルの物理配置	48
2-3	バックアップ設計	53

バックアップの基本分類	53
完全/差分/増分	54
フルバックアップ	54
差分バックアップ	56
増分バックアップ	58
バックアップ方式にもトレードオフがある	59
どんなバックアップ方式を採用すべきか?	61
<b>2-4 リカバリ設計</b>	63
リカバリとリストア	63
リストアとロールフォワード	65
演習問題	66

**第3章 論理設計と正規化**  
 ~なぜテーブルは分割する必要があるのか? 67

<b>3-1 テーブルとは何か?</b>	69
二次元表≠テーブル	69
<b>3-2 テーブルの構成要素</b>	72
行と列	72
キー	72
制約	78
テーブルと列の名前	80
<b>3-3 正規化とは何か?</b>	84
正規形の定義	84
<b>3-4 第1正規形</b>	86
第1正規形の定義~スカラ値の原則	86
第1正規形を作ろう	87
なぜ一つのセルに複数の値を入れてはダメなのか? ~関数従属性	90
<b>3-5 第2正規形~部分関数従属</b>	93
第2正規化を行なう	94
第2正規形でないと何が悪いのか?	95
無損失分解と情報の保存	96
<b>3-6 第3正規形~推移的関数従属</b>	99
推移的関数従属	99
第3正規化を行なう	100
<b>3-7 ボイスコード正規形</b>	102
3次と4次の狭間	102
ボイスコード正規化を行なう	106

<b>3-8 第4正規形</b>	109
多値従属性~キーと集合の対応	110
第4正規化を行なう	112
第4正規形の意義	112
<b>3-9 第5正規形</b>	114
第5正規化を行なう	115
<b>3-10 正規化についてのまとめ</b>	116
正規化の三つのポイント	116
正規化は常にすべきか?	117
演習問題	122

**第4章 ER図~複数のテーブルの関係を表現する** 123

<b>4-1 テーブルが多すぎる!</b>	125
<b>4-2 テーブル同士の関連を見抜く</b>	126
1対1、1対多、多対多	127
<b>4-3 ER図の描き方</b>	129
テーブル(エンティティ)の表記方法	129
IE表記法でER図を描く	130
IDEF1XでER図を描く	131
<b>4-4 「多対多」と関連実体</b>	135
演習問題	137

**第5章 論理設計とパフォーマンス**  
 ~正規化の欠点と非正規化 139

<b>5-1 正規化の功罪</b>	141
正規化とSQL(検索)	141
正規化とSQL(更新)	146
正規化と非正規化、どちらが正解なのか?	147
<b>5-2 非正規化とパフォーマンス</b>	149
サマリデータの冗長性とパフォーマンス	149
選択条件の冗長性とパフォーマンス	153
<b>5-3 冗長性とパフォーマンスのトレードオフ</b>	157

更新時のパフォーマンス	157
データのリアルタイム性	157
改修コストの大きさ	158
🔍 演習問題	159

## 第6章 データベースとパフォーマンス 161

6-1 データベースのパフォーマンスを決める要因	163
インデックス	163
統計情報	164
6-2 インデックス設計	166
まずはB-treeインデックスから	167
B-treeインデックスの長所	167
B-treeインデックスの構造	169
6-3 B-treeインデックスの設計方針	173
B-treeインデックスはどの列に作れば良いか?	173
B-treeインデックスとテーブルの規模	173
B-treeインデックスとカーディナリティ	174
B-treeインデックスとSQL	176
B-treeインデックスに関するその他の注意事項	178
6-4 統計情報	180
オブティマイザと実行計画	180
統計情報の設計指針	182
🔍 演習問題	186

## 第7章 論理設計のバッドノウハウ 187

7-1 論理設計の「やってはいけない」	189
7-2 非スカラ値 (第1正規形未満)	191
配列型による非スカラ値	191
スカラ値の基準は何か?	194
7-3 ダブルミーニング	196
この列の意味は何でしょう?	196
テーブルの列は「変数」ではない	197
7-4 単一参照テーブル	199

多すぎるテーブルをまとめたい?	199
単一参照テーブルの功罪	201

7-5 テーブル分割	203
テーブル分割の種類	203
水平分割	204
垂直分割	206
集約	208
7-6 不適切なキー	213
キーは永遠に不変です!	214
同じデータを意味するキーは同じデータ型にすべし	214
7-7 ダブルマスタ	216
ダブルマスタはSQLを複雑にし、パフォーマンスを悪化させる	216
ダブルマスタはなぜ生じるのか	218
🔍 演習問題	222

## 第8章 論理設計のグレーノウハウ 223

8-1 違法すれすれの「ライン上」に位置する設計	225
8-2 代理キー～主キーが役に立たないとき	226
主キーが決められない、または主キーとして不十分なケース	226
代理キーによる解決	229
自然キーによる解決	230
インターバル	232
オートナンバリングの是非	234
8-3 列持ちテーブル	241
配列型は使えない、でも配列を表現したい	241
列持ちテーブルの利点と欠点	242
行持ちテーブル	243
8-4 アドホックな集計キー	246
8-5 多段ビュー	249
ビューへのアクセスは「2段階」で行なわれる	249
多段ビューの危険性	250
8-6 データクレンジングの重要性	253
データクレンジングは設計に先立って行なう	253
代表的なデータクレンジングの内容	254
🔍 演習問題	259

第9章	<b>一步進んだ論理設計 ～SQLで木構造を扱う</b>	261
9-1	リレーショナルデータベースのアキレス腱	263
	木構造とは?	263
9-2	伝統的な解法～隣接リストモデル	265
9-3	新しい解法～入れ子集合モデル	267
	入れ子集合モデルを使った検索	268
	入れ子集合モデルを使った更新	271
9-4	もしも無限の資源があったなら ～入れ子区間モデル	275
	使っても使っても尽きない資源	275
	入れ子区間モデルを使った更新	276
9-5	ノードをフォルダだと思え～経路列挙モデル	279
	ファイルシステムとしての階層	279
	経路列挙モデルによる検索	282
	経路列挙モデルを使った更新	285
9-6	各モデルのまとめ	290
	演習問題	294

## 付 録 **演習問題の解答** 295

第1章	解答	296
	演習 1-1 DBMSの情報確認	296
	演習 1-2 アプリケーション改修のタイプとコスト	298
第2章	解答	301
	演習 2-1 データベースサーバーのクラスタリング構成	301
	演習 2-2 ハードウェアリソースの情報取得	304
	演習 2-3 サーバー CPUの机上サイジング	304
第3章	解答	307
	演習 3-1 正規形の次数	307
	演習 3-2 関数従属性	307
	演習 3-3 正規化	307
第4章	解答	310

	演習 4-1 ER図	310
	演習 4-2 関連エンティティ	312
	演習 4-3 多対多の関連	312
第5章	解答	313
	演習 5-1 正規化されたテーブルに対するSQL	313
	演習 5-2 非正規化によるSQLチューニング	316
第6章	解答	320
	演習 6-1 ビットマップインデックスとハッシュインデックス	320
	演習 6-2 インデックスの再編成	322
第7章	解答	324
	演習 7-1 パーティションの特徴	324
	演習 7-2 マテリアライズドビューの機能	325
第8章	解答	328
	演習 8-1 ビジネスロジックの実装方法の検討	328
	演習 8-2 一時テーブル	333
第9章	解答	336
	演習 9-1 木構造を扱うモデルの正規形	336
	演習 9-2 実数のデータ型	338
	索引	344

<b>COLUMN</b>	Web3層モデル	22
	クラウドとスケラビリティ	40
	RAID6	47
	関係（リレーショナル）とは何か?	81
	正規化を学ぶことに対する外的な批判について	118
	損失分解	120
	shardingとカラムベース	211
	パッドノウハウのどこが悪いのか?	219
	主キーはなぜ必要か?	239
	バックアップとレプリケーション	260
	入れ子集合とフラクタル	291