

徹底攻略

# データベース スペシャリスト 教科書

株式会社わくわくスタディワールド 瀬戸美月 著

平成 **29** 年度  
(2017年度)

電子版無料  
ダウンロード!

2大特典

スマホで学べる  
単語帳アプリ  
「でる語句200」付き

さらに

過去問題  
解説

4回分\*

動画  
講義

も付いて  
くる!

インプレス

\*紙面 1 回分 + PDF 提供 3 回分

徹底攻略

# データベーススペシャリスト 教科書

株式会社わくわくスタディワールド 瀬戸美月 著

平成29年度

インプレス

## インプレス情報処理シリーズ購入者限定特典!!

### ●電子版の無料ダウンロード

本書の全文の電子版（PDFファイル）を無料でダウンロードいただけます。  
加えて、本書に掲載していない過去問題＆解説もダウンロードできます。

#### ▼本書でダウンロード提供している過去問題解説

- ・平成25年度春期試験（平成26年度版に収録した過去問題＆解説）
- ・平成26年度春期試験（平成27年度版に収録した過去問題＆解説）
- ・平成27年度春期試験（平成28年度版に収録した過去問題＆解説）

電子版は、以下のURL からダウンロードいただけます。

ダウンロードURL： <http://book.impress.co.jp/books/1116101018>

※画面の指示に従って操作してください。

※ダウンロードには、無料の読者会員システム「CLUB Impress」への登録が必要となります。

### ●スマホで学べる単語帳アプリ「でる語句200」のダウンロード

データベーススペシャリスト試験で出題頻度の高い200の語句をいつでもどこでも暗記できる単語帳アプリ「でる語句200」を無料でダウンロードいただけます。  
手順については、638ページを参照ください。

※本特典のご利用は、書籍をご購入いただいた方に限ります。

※ダウンロード期間は、いずれも本書発売より1年間です。

## インプレスの書籍ホームページ

書籍の新刊や正誤表など最新情報を随時更新しております。

<http://book.impress.co.jp/>

- ・本書は、データベーススペシャリスト試験対策用の教材です。著者、株式会社インプレスは、本書の使用による合格を保証するものではありません。
- ・本書の内容については正確な記述につとめました。著者、株式会社インプレスは本書の内容に基づくいかなる試験の結果にも一切責任を負いかねますので、あらかじめご了承ください。
- ・本書の試験問題は、独立行政法人 情報処理推進機構の情報処理技術者試験センターが公開している情報に基づいて作成しています。
- ・本文中の製品名およびサービス名は、一般に各開発メーカーおよびサービス提供元の商標または登録商標です。なお、本文中には©および®、™は明記していません。

## はじめに

データベーススペシャリスト試験は、以前は“10年経っても傾向が変わらない試験”でした。データベースの理論を中心とした出題で、正規化やE-R図を中心に、変わらない基本を問う試験だったからです。

しかしここ数年、データベーススペシャリスト試験は少しずつ変わりつつあります。データをクラウド上に格納することが多くなり、またビッグデータの取扱いも増えているため、データベースの使用方法が変わってきています。これは試験内容にも少しずつ反映されています。データベース理論の基礎は大切ですが、それ以外にも情報セキュリティやNoSQLをはじめ、知っておくべきことが増えてきているのです。

この本は、データベーススペシャリスト試験に必要な知識を、基本から順に説明し、理論を中心にすべて学習することを意図して作成しました。しっかりと基礎を固めることで、付け焼き刃ではない本物の実力を身に付けることができます。試験対策本なので、問題の解き方などのテクニックも載せていますが、正攻法で合格するための情報を中心にまとめました。2016年5月30日に発表されたシラバスの改訂 (Ver 3.0) にも対応しています。さらに深く理解できるように動画による解説も用意していますので、あわせてご活用ください。本書をお供にしながら、データベーススペシャリスト試験の合格に向かって進んでいきましょう。

本書の発刊にあたり、企画・編集等様々な分野で多大なるご尽力をいただきました、インプレスの皆様、ソキウス・ジャパンの皆様、私の恩師である水岡祥二様、わくわくスタディワールドの齋藤健一様に感謝いたします。そして、本書を執筆するにあたっていろいろな示唆を与えてくださった「わく☆すたセミナー」や、企業研修、勉強会など、様々な場所で関わってくださった受講生の皆様にも感謝いたします。皆様のおかげで、この本を完成させることができました。本当に、ありがとうございます。

平成28年7月

わくわくスタディワールド 瀬戸 美月

## CONTENTS

# 目次

はじめに.....	3
合格のポイント .....	10

## 第1章 データベースとは

<b>1-1 データベースの基本 .....</b>	<b>20</b>
1-1-1 データベースとは .....	20
1-1-2 データモデル .....	22
1-1-3 関係モデル .....	26
1-1-4 データベースの三値論理 .....	30
<b>1-2 システム開発とデータベース .....</b>	<b>32</b>
1-2-1 システム開発のアプローチ .....	32
1-2-2 データ中心アプローチ .....	36

## 第2章 データベース基礎理論

<b>2-1 正規化理論 .....</b>	<b>42</b>
2-1-1 正規化とは .....	42
2-1-2 第1正規形 .....	50
2-1-3 第2正規形 .....	56
2-1-4 第3正規形 .....	62
2-1-5 高次の正規形 .....	68
<b>2-2 関係演算 .....</b>	<b>76</b>
2-2-1 関係演算の種類 .....	76
2-2-2 結合演算 .....	83
<b>2-3 問題演習 .....</b>	<b>88</b>
2-3-1 午前問題 .....	88
2-3-2 午後問題 .....	93
2-3-3 基礎理論 午後 I 問題の傾向 .....	105

## 第3章 データベース設計

### 3-1 データベース設計 ..... 108

3-1-1 データベース設計の二つのアプローチ .....	108
3-1-2 トップダウンアプローチ .....	112
3-1-3 ボトムアップアプローチ .....	116
3-1-4 正規化による不都合と非正規化 .....	120
3-1-5 データベースの制約 .....	123
3-1-6 データベースシステム設計 .....	125

### 3-2 E-R図 ..... 129

3-2-1 E-R図 .....	129
3-2-2 スーパータイプ/サブタイプ .....	137
3-2-3 E-R図での表現 .....	140

### 3-3 問題演習 ..... 143

3-3-1 午前問題 .....	143
3-3-2 午後問題 .....	149
3-3-3 データベース設計 午後 I 問題の傾向 .....	164

## 第4章 SQL

### 4-1 SQL ..... 168

4-1-1 SQLとは .....	168
4-1-2 SQL-DDL .....	170
4-1-3 SQL-DCL .....	178
4-1-4 SQL-DML .....	182

### 4-2 SQLのポイント ..... 191

4-2-1 グループ化 .....	191
4-2-2 結合 .....	195
4-2-3 副問合せ .....	199

### 4-3 SQLでできること ..... 206

4-3-1 ビュー .....	206
4-3-2 カーソル .....	211
4-3-3 ストアドプロシージャ .....	214

<b>4-4</b>	<b>問題演習 .....</b>	<b>216</b>
4-4-1	午前問題 .....	216
4-4-2	午後問題 .....	228
4-4-3	SQL問題の傾向.....	243

## 第5章 DBMS

---

<b>5-1</b>	<b>DBMSとは.....</b>	<b>246</b>
5-1-1	DBMSの三大機能 .....	246
5-1-2	インデックス .....	249
5-1-3	性能設計 .....	255
5-1-4	テーブルの結合 .....	258

<b>5-2</b>	<b>トランザクション管理 .....</b>	<b>260</b>
5-2-1	トランザクションとは.....	260
5-2-2	ロック .....	264
5-2-3	トランザクション分離レベル .....	272

<b>5-3</b>	<b>障害回復処理 .....</b>	<b>276</b>
5-3-1	ログ .....	276
5-3-2	障害回復処理.....	283

<b>5-4</b>	<b>分散データベース .....</b>	<b>285</b>
5-4-1	分散データベース .....	285

<b>5-5</b>	<b>問題演習 .....</b>	<b>290</b>
5-5-1	午前問題 .....	290
5-5-2	午後問題 .....	302
5-5-3	DBMS 午後問題の傾向 .....	314

## 第6章 概念設計

<b>6-1</b>	<b>概念設計 .....</b>	<b>318</b>
6-1-1	概念データモデルの作成 .....	318
6-1-2	概念データモデル作成上の注意点 .....	323
<b>6-2</b>	<b>概念設計 午後Ⅱ問題 .....</b>	<b>326</b>
6-2-1	概念設計 午後Ⅱ問題の解き方 .....	326
6-2-2	概念設計 午後Ⅱ問題の傾向 .....	355
<b>6-3</b>	<b>問題演習 .....</b>	<b>357</b>
6-3-1	午後問題 .....	357

## 第7章 論理設計・物理設計

<b>7-1</b>	<b>論理設計 .....</b>	<b>386</b>
7-1-1	論理設計とは .....	386
7-1-2	移行設計 .....	388
7-1-3	論理設計 午後Ⅱ問題の傾向 .....	390
<b>7-2</b>	<b>物理設計 .....</b>	<b>392</b>
7-2-1	性能設計 .....	392
7-2-2	運用・保守設計 .....	394
7-2-3	物理設計 午後Ⅱ問題の傾向 .....	396
<b>7-3</b>	<b>問題演習 .....</b>	<b>397</b>
7-3-1	午後問題 .....	397



## 第8章 セキュリティ

### 8-1 セキュリティ ..... 426

8-1-1 情報セキュリティ .....	426
8-1-2 情報セキュリティ管理 .....	438
8-1-3 セキュリティ技術評価 .....	445
8-1-4 情報セキュリティ対策 .....	449
8-1-5 セキュリティ実装技術 .....	455

### 8-2 データベースセキュリティ ..... 461

8-2-1 データベースセキュリティ .....	461
--------------------------	-----

### 8-3 問題演習 ..... 464

8-3-1 午前問題 .....	464
------------------	-----

## 第9章 最新データベース技術

### 9-1 様々なデータベース ..... 470

9-1-1 NoSQL .....	470
9-1-2 データウェアハウス .....	476
9-1-3 オブジェクト指向データベース .....	481
9-1-4 XMLDB .....	483

### 9-2 データベース周辺技術 ..... 484

9-2-1 仮想化とストレージ .....	484
9-2-2 データベースの評価 .....	489
9-2-3 ビッグデータとAI .....	492

### 9-3 問題演習 ..... 494

9-3-1 午前問題 .....	494
9-3-2 午後問題 .....	497

## 付録 平成28年度春 データベーススペシャリスト試験

午前Ⅰ 問題.....	515
午前Ⅰ 解答と解説.....	524
午前Ⅱ 問題.....	529
午前Ⅱ 解答と解説.....	538
午後Ⅰ 問題.....	543
午後Ⅰ 解答と解説.....	562
午後Ⅱ 問題.....	581
午後Ⅱ 解答と解説.....	601
索引.....	627
「でる語句200」の使い方.....	638



新しくなりつつあるデータベーススペシャリスト試験 .....	29
データベーススペシャリストは、主にデータベース設計を行う人.....	35
第何正規形まで出題される? .....	75
データベーススペシャリスト試験に必要な国語力 .....	106
SQL ができなくても受かる? .....	244
計算問題の解き方 .....	315
「分かる」を「できる」にしよう .....	325
情報セキュリティの最新情報をチェックする.....	463
新しい技術への対応 .....	491



# 合格のポイント

## データベーススペシャリスト試験の概要

情報処理技術者試験におけるデータベーススペシャリスト試験の位置付けは、データベースの専門家を対象とした高度区分試験の一つです。試験要綱によると、データベーススペシャリスト試験の対象者像は、「高度IT人材として確立した専門分野をもち、データベースに関係する固有技術を活用し、最適な情報システム基盤の企画・要件定義・開発・運用・保守において中心的な役割を果たすとともに、固有技術の専門家として、情報システムの企画・要件定義・開発・運用・保守への技術支援を行う者」です。データベースに関係する技術の専門家で、システム開発を中心に様々な分野でデータベースに関する技術支援を行うことが想定されています。単にデータベースが使えるというだけでなく、**データベースに関する技術を活用し、情報システム基盤の構築を行ったり、情報システムへの技術支援を行ったりする人**がデータベーススペシャリストです。

一般的には、データベース関連の技術者は大きく次の二つのタイプに分けられます。

### ①データベース設計を行う技術者

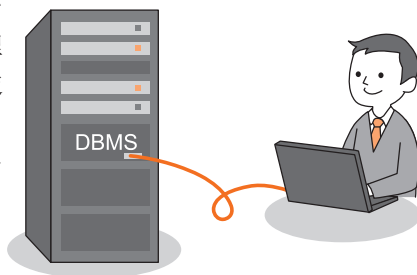
情報システム開発において、データに関する要件定義やデータベース設計を主に行う技術者です。会社全体の業務を分析し、会社特有の状況を考えて、データベース理論に裏打ちされた、業務に最適なデータベースを設計します。



### ②データベース管理者

DBMSの設定や運用管理、データベースに障害が起こったときのトラブルシューティングなどを行う技術者です。日々のデータベース管理をメインで行う技術者で、実際にDBMSを設定し、構築作業や運用作業を行います。

それぞれのDBMSについての知識や、パフォーマンスチューニングや運用管理の方法など、様々なスキルが必要です。



データベーススペシャリスト試験の対象は、主に①の技術者です。データベース設計を確実に行うことができる技術者かどうかを、午後Ⅰや午後Ⅱの問題で詳細に問われます。②のDBMSの管理者に関しては、この試験ではあまり対象としていません。こちらはOracle MasterやMCPなどのベンダー資格が数多くあり、OSS-DBなどオープンソースの資格もあります。データベーススペシャリスト試験はこれらの試験とは少し毛色が違い、あくまでも要件定義や概要設計などのシステム開発の場面でデータベース設計を行う技術者を主な対象としています。ただし、午前Ⅱや午後の一部ではDBMSに関すること問われるので、ひとつおりの知識を身に付けておく必要はあります。

データベーススペシャリスト試験は午前Ⅰ試験、午前Ⅱ試験、午後Ⅰ試験、午後Ⅱ試験の4区分に分かれていて、それぞれ異なる方法で異なる力が試されます。まとめると、以下のようになります。

#### データベーススペシャリスト試験の構成

	試験時間	出題形式	出題数・解答数	合格ライン
午前Ⅰ	9:30～10:20(50分)	多肢選択式(四肢択一)	30問・30問	60点／100点満点(18問正解)
午前Ⅱ	10:50～11:30(40分)	多肢選択式(四肢択一)	25問・25問	60点／100点満点(15問正解)
午後Ⅰ	12:30～14:00(90分)	記述式	3問・2問	60点／100点満点
午後Ⅱ	14:30～16:30(120分)	記述式	2問・1問	60点／100点満点

また、過去3回の各時間区分及び全体の合格率は次のとおりです。

#### 各時間区分の突破率と合格率

	平成26年春	平成27年春	平成28年春
午前Ⅰ突破率	71.7%	68.1%	73.9%
午前Ⅱ突破率	75.6%	83.2%	81.3%
午後Ⅰ突破率	54.5%	51.3%	51.5%
午後Ⅱ突破率	48.4%	49.2%	48.7%
全体の合格率	16.7%	17.6%	17.5%

※情報処理技術者試験センター公表の統計情報を基に算出

午前Ⅰは高度区分共通であり、免除制度を利用する受験者が多いのですが、突破率は6割前後なので受験される方は注意が必要です。また、それぞれの時間区分で基準点(60点)を突破できないと次の時間区分は採点されないため、それぞれの突破率は、その前までの時間区分の試験をすべて突破した受験者に対する突破率です。合格率は15%前後と難易度の高い試験ではありますが、それぞれの時間区分を見ると半分程度の受験者が突破しています。満点を取らなくてもいいので、それぞれの時間区分に対応した試験対策を行い、確実に60点以上取れるように対策を行う必要があります。

基本的に難易度は午前より午後の方が難しく、特にデータベーススペシャリストは午後Ⅱが合否の一番のポイントになるので、勉強するときには午後対策に重点を置く必要があります。ただし、午前Ⅰが免除されない場合は、データベース分野以外の勉強も幅広く行わなければならないので、そのための準備も必要になります。

## 午前試験のポイント

データベーススペシャリスト試験の午前試験は、午前Ⅰが30問、午前Ⅱが25問です。午前Ⅰは高度区分共通であり、ITに関連する全分野から出題されます。過去3回の出題数は、以下のとおりです。

### 午前Ⅰの分野別出題数

出題分野		平成26年春	平成27年春	平成28年春
テクノロジー系	基礎理論	2	3	3
	コンピュータシステム	5	4	4
	技術要素	8	8	8
	開発技術	2	2	2
マネジメント系	プロジェクトマネジメント	2	2	1
	サービスマネジメント	3	3	4
ストラテジ系	システム戦略	3	3	3
	経営戦略	3	3	3
	企業と法務	2	2	2

最も重視されるのがテクノロジー系の分野で、出題数は毎回17問程度です。その他は、マネジメント系が5問、ストラテジ系が8問程度の配分です。全体的にまんべんなく出題され、合格ラインは60%なので、30問中18問以上に正解する必要があります。したがって、応用情報技術者試験用の参考書や問題集などを活用し、**すべての分野についてひととおりは学習**することが午前Ⅰ攻略のポイントになります。午前Ⅰを一度突破すると、以後2年間は免除対象者となるので、まずは午前Ⅰを確実に突破して合格に近づきましょう。また、午前Ⅰの勉強をしっかりと行っておくと、専門的な勉強も理解しやすくなります。勉強は積み重ねですし、データベースは数学を利用することが多いので、特に基礎理論分野を理解していると、専門的な学習を進めていく上でとても助けになります。

午前Ⅱでは、データベースに関連する分野が出題されます。といっても、データベースだけではなく、コンピュータ構成要素、システム構成要素、データベース、セキュリティ、システム開発技術、ソフトウェア開発管理技術の6分野から出題されることになっています。このうちデータベースのみが重点分野で、難易度はレベル4(最高難度)とされています。

過去3回の出題数は、およそ次のとおりです。

#### 午前Ⅱの分野別出題数

出題分野		平成26年春	平成27年春	平成28年春
コンピュータシステム	コンピュータ構成要素	1	1	1
	システム構成要素	1	1	1
技術要素	データベース	19	19	19
	セキュリティ	2	2	2
開発技術	システム開発技術	1	1	1
	ソフトウェア開発管理技術	1	1	1

データベース以外の分野は、セキュリティが2問、そのほかは1問ずつのみの出題です。過去問題の再出題が半分以上を占めるので、過去問題をしっかり演習しておくことが大切です。特にデータベース分野は、過去のテクニカルエンジニア（データベース）時代も含めると、かなりの割合で過去問題が再出題されています。また、データベースに関する新たな知識は午前Ⅱで最初に出題されることが多いので、過去問題を演習しつつ最新の動向にも目を向けておきましょう。

## 午後試験のポイント

データベーススペシャリスト試験の午後試験は、午後Ⅰで3問、午後Ⅱで2問が出題され、そのうち午後Ⅰでは2問、午後Ⅱでは1問選択して解答します。

まず午後Ⅰについて、過去3回で出題されたテーマは次のとおりです。

#### 午後Ⅰの分野別出題テーマ

問	平成26年春	平成27年春	平成28年春
問1	データベースの設計	データベースの設計	データベースの設計
問2	データベースアクセスの同時実行制御	データベースの設計	データベースの運用設計
問3	テーブルの設計及びSQLの設計	バッチ処理の性能設計	RDBMSのセキュリティ

午後Ⅰでは各問で、データベースの基礎理論や設計など、特定のテーマについて出題されます。定番で出題されるのは、データベースの設計です。また、SQLの設計やデータベースの性能改善、同時実行制御などのDBMSに関することも出題されます。どの問題も単なる知識を問うことはなく、理論をもとに実例に当てはめて考えていくかたちになっています。そのため、表面的な知識ではなく、理論をしっかりと理解して使いこなせるようになることが重要です。

次に午後Ⅱについて、過去3回で出題されたテーマは次のとおりです。

### 午後Ⅱの分野別出題テーマ

問	平成 26 年 春	平成 27 年 春	平成 28 年 春
問1	データベースの物理設計	データベースの物理設計	データベースの物理設計とデータ移行
問2	ホテルの宿泊管理システムのデータベース設計	部品在庫の倉庫管理	アフタサービス業務の概念データモデリング

午後Ⅱでは、基本的に実際の事例を基にデータベース設計を行っていきます。問1は、物理設計を中心に出题される傾向があり、問2は概念設計を中心にした問題が出题される傾向があります。

問題文が長く、登場する企業の状況に沿った解答が求められる設問が多いので、午後Ⅰよりも問題文をしっかり読んで状況を理解することが大切です。

データベーススペシャリスト試験の午後の問題では、定番の内容が多く出題されます。そのため、データベーススペシャリストの午後試験の対策としては、過去問題の演習を行って1問1問をしっかり理解しておくことが大切です。

一つ一つの理論や用語を丁寧に深く理解しておくことと、過去問題の演習をしっかり行うことが合格のポイントとなります。

## 合格に必要なスキル・能力

データベーススペシャリスト試験に合格するために一番必要な能力は、**勉強を続ける**力です。データベース分野の勉強では、じっくり地道に演習を積み重ねていくことが一番大切です。才能がないと受からないという試験ではないので、きちんと勉強を続けていくことができさえすれば、合格できます。

また、午後Ⅱは特に問題文が長いので、**問題文を正確に読み取る読解力**や、**文章で表現するための文章力**も必要です。データベーススペシャリスト試験の午後問題は、午後Ⅰが1問6～10ページ、午後Ⅱが1問12～20ページ程度で、他の試験区分に比べてかなり長い文章になっています。読むのが遅いと時間が足りなくなりますし、読んで勘違いをすると適切な答えが書けなくなります。データベーススペシャリスト試験の場合は、問題文に専門用語が多く、最初は読みこなすのに時間がかかる傾向があるので、意識して読解力を身に付ける必要があります。

スキルとしてあると有利なのは、**DBMS構築経験**と、データベースに関するシステム開発に関わった**システム開発経験**です。情報処理技術者試験は、あくまで仕事をする上

での基本を問う試験なので、実務経験がある分野は容易に解けるはずです。また、実務ではなくても、DBMSやデータベースを使ったサービスを構築してみたといった経験があれば、もちろん役に立ちます。遊びがてら、自分のPCにMySQLなどフリーのDBMSを入れて実験してみるのも楽しいでしょう。

このように、データベーススペシャリスト試験に合格するには、様々なスキルや能力が必要になってきます。自分にある能力、ない能力を見極めて、必要なスキルや能力を身に付けていきましょう。

## 合格のための勉強法

試験合格のために勉強計画を立てる上で最も大切なのは、**自分の現状を知ること**です。データベーススペシャリスト試験ではデータベースに関連する問題が出題されますが、ただ漠然と勉強していけばいいわけではありません。実務経験がある分野はあまり勉強しなくてもいいですし、苦手分野は重点的に学習する必要があります。基本的に、データベーススペシャリスト試験の受験者は**応用情報技術者合格レベルが前提**なので、そこからのステップアップを考えます。応用情報技術者試験での基礎の上に専門知識を積み重ねていくので、応用情報技術者試験の勉強をされていない方は、まずはそこからスタートすることをおすすめします。受験しなくても、応用情報技術者試験の勉強を行ってからデータベーススペシャリスト試験の内容に入ると、無理なく学習を進めることができます。午前Ⅰを突破していない場合は、午前Ⅰの勉強を兼ねて応用情報技術者試験レベルの基礎知識を身に付けることをおすすめします。

データベーススペシャリスト試験の勉強法として最も一般的なのは、**参考書を1冊マスターし、過去問題の演習を行う**ことです。知識を身に付けてから過去問題で演習を行うことで、これまで学習したことを定着させることができます。

データベーススペシャリスト試験の場合、過去問題の質が高く、問題演習を行うことで実力をアップさせることができるので、最低でも過去3回分程度の問題の演習をすることをおすすめします。本書では、各章に午前・午後の演習問題を掲載し、さらに巻末に最新の過去問題とその解答解説を掲載しています。ぜひ活用してください。また、データベース技術は内容が古くなりにくいので、不十分だと感じたら、過去の問題にどんどんさかのぼって演習を行うこともできます。



## 本書の使い方

本書では、これまでに出题された問題を分析し、試験によく出てくる分野を中心にまとめています。ですから、本書をすべて読んで頭に入れていただければ、試験に合格するための知識は十分に身に付きます。ただ、この試験は知識を問う試験ではないので、理解を深めるためにも、途中にある演習問題はぜひ、考えながら読み進めてください。

また、文章を読むのが苦手な人、特に参考書を読み続けるのがつらいという人は、無理に最初から全部読む必要はありません。過去問題などで問題演習を行いながら、辞書として必要なことを調べるといった用途に使っていただいてもかまいません。特に午後Ⅰや午後Ⅱの問題は、解き方を読みながら演習を行うと、効率良く勉強していただけると思います。

巻末に平成28年春試験の問題と解答解説を掲載しました。また、平成25年、26年、27年の春試験の解答解説は、本書の特典としてダウンロード可能です。学習してきたことの力試しに、そして問題演習に、ぜひお役立てください。

## 本書のフォローアップ

本書の訂正情報につきましては、インプレスのサイトをご参照ください。内容に関するご質問は、「お問い合わせフォーム」よりお問い合わせください。

### ●お問い合わせと訂正ページ

<http://book.impress.co.jp/books/1116101018>

上記のページで「お問い合わせフォーム」ボタンをクリックしますとフォーム画面に進みます。

また、書籍以外の手段でも学べるように、データベース理論などを動画で解説した内容を公開しています。本書との関連は以下のWebページにまとめてありますので、よろしければご活用ください。

### 徹底攻略データベーススペシャリスト教科書 書籍関連情報

<http://www.wakuwakustudyworld.co.jp/blog/dbinfo/>

それでは、試験合格に向けて、楽しく勉強を進めていきましょう。

## 本書の構成

本書は、解説を読みながら問題を解くことで、知識が定着するように構成されています。また、側注には、理解を助けるヒントを豊富に盛り込んでいますので、ぜひ活用してください。

アイコンで種別された側注で、知識を補足します。

過去問題の分析に基づき、頻出の分野を中心に構成されているので、試験に必要な知識が確実に身につきます。

随所に設けられた問題を解くことで、知識が定着します。

**5-1 DBMSとは**

DBMSは、メタデータ管理、異種処理、トランザクション管理という三大機能を実現するためのシステムです。DBMSでは、インテリクスの活用によって高速化を実現します。

**5-1-1 DBMSの三大機能**

DBMSが実現すべき機能には、メタデータ管理、異種処理、トランザクション管理の三つがあり、これをDBMSの三大機能といえます。

**DBMSの三大機能**

DBMS (Data Base Management System) はデータベース管理システム (データベース) のことを指します。

① **メタデータ管理**

メタデータ管理とは、データベースに蓄積するデータを管理することです。メタデータとは、データベースにどのようなデータが蓄積されているか、そのデータにどのような処理が行われているか、といったデータに関する情報を指します。これらは、データベースの外部スキーマ、概念スキーマ、内部スキーマに関する情報となります。

メタデータは、データベースの管理者がデータベースに蓄積されるデータの種類、構造、内容、アクセス方法、セキュリティに関する情報を管理するために必要となります。

② **異種処理**

異種処理とは、ユーザーがデータベースに実行する異種 (問合せ) に対応する処理です。異種の内容は複雑なSQL、複雑な計算にわたるため、データベースに蓄積されたデータを読み取り、必要な処理を行います。

異種を解釈する過程は処理方法によって大きく変わってきます。

5-2 トランザクション管理

**5-2-1 トランザクション管理**

トランザクションとは、データベースがデータを読み書きする一連の処理の単位です。トランザクションを管理することで、複数のトランザクションの同時実行を制御することができます。また、トランザクション単位で障害時の回復処理を行うことも可能です。

それでは、次の問題を考えてみましょう。

**問題**

トランザクションのACID特性のうち、一貫性 (consistency) の説明はどれか。

ア 整合性の概念がデータベースに於いて、トランザクション実行前後の整合性が保たれている性質である。

イ 同時実行される複数のトランザクションは互いに干渉しないという性質である。

ウ トランザクションは、発生から実行完了するまで、全く同じ状態でデータベースの更新、読み取りが行われる。

エ 発生から実行完了するまで、データベースの整合性が保たれていない状態である。

正解はアである。

トランザクションのACID特性のうち、一貫性 (consistency) は、データベースの整合性を保つことを指します。これは、データベースの整合性が、トランザクションの実行前後で保たれることを意味します。

アは整合性の概念がデータベースに於いて、トランザクション実行前後の整合性が保たれている性質である。これは、データベースの整合性を保つことを指します。

イは同時実行される複数のトランザクションは互いに干渉しないという性質である。これは、データベースの整合性を保つことを指します。

ウはトランザクションは、発生から実行完了するまで、全く同じ状態でデータベースの更新、読み取りが行われる。これは、データベースの整合性を保つことを指します。

エは発生から実行完了するまで、データベースの整合性が保たれていない状態である。これは、データベースの整合性を保つことを指します。

5-3 トランザクション管理

**5-3-1 トランザクション管理**

トランザクションとは、データベースがデータを読み書きする一連の処理の単位です。トランザクションを管理することで、複数のトランザクションの同時実行を制御することができます。また、トランザクション単位で障害時の回復処理を行うことも可能です。

それでは、次の問題を考えてみましょう。

**問題**

トランザクションのACID特性のうち、一貫性 (consistency) の説明はどれか。

ア 整合性の概念がデータベースに於いて、トランザクション実行前後の整合性が保たれている性質である。

イ 同時実行される複数のトランザクションは互いに干渉しないという性質である。

ウ トランザクションは、発生から実行完了するまで、全く同じ状態でデータベースの更新、読み取りが行われる。

エ 発生から実行完了するまで、データベースの整合性が保たれていない状態である。

正解はアである。

トランザクションのACID特性のうち、一貫性 (consistency) は、データベースの整合性を保つことを指します。これは、データベースの整合性が、トランザクションの実行前後で保たれることを意味します。

アは整合性の概念がデータベースに於いて、トランザクション実行前後の整合性が保たれている性質である。これは、データベースの整合性を保つことを指します。

イは同時実行される複数のトランザクションは互いに干渉しないという性質である。これは、データベースの整合性を保つことを指します。







ウはトランザクションは、発生から実行完了するまで、全く同じ状態でデータベースの更新、読み取りが行われる。これは、データベースの整合性を保つことを指します。

エは発生から実行完了するまで、データベースの整合性が保たれていない状態である。これは、データベースの整合性を保つことを指します。

重要用語は色文字で表記されているので、直前対策にも役立ちます。

項の最後に重要ポイントを押さえます。

## 本書で使用している側注のマーク

 <p><b>勉強のコツ</b></p> <p>学習を進めるうえでの準備や、勉強方法などを紹介</p>	 <p><b>用語</b></p> <p>本文に登場した用語を詳しく解説</p>	 <p><b>関連</b></p> <p>本書における関連項目や、参照URLなどを記載</p>
 <p><b>発展</b></p> <p>上のレベルの学習につなげるために知っておくと有意義な知識を解説</p>	 <p><b>過去問題をチェック</b></p> <p>同様の問題が出題された年度と問題番号を紹介</p>	 <p><b>動画</b></p> <p>本書の内容の補足として著者が公開している動画学習サイトの案内</p>



# 第 1 章

## データベースとは

データベーススペシャリスト試験の勉強を行う前に、データベースの基本を理解することはとても大切です。

この章ではまず、データベースとは何かについて学びます。様々なデータモデルと、現在最も使われている関係モデルについて解説します。

次に、システム開発とデータベースの関連について、その概要を学びます。データベースは単独で使われるものではなく、システムから利用されるものです。システム開発の流れと、その中でのデータベースの位置付けについて、開発手法の一つであるデータ中心アプローチを中心に学習していきます。

### 1-1 データベースの基本

- ◆ 1-1-1 データベースとは
- ◆ 1-1-2 データモデル
- ◆ 1-1-3 関係モデル
- ◆ 1-1-4 データベースの三値論理

### 1-2 システム開発とデータベース

- ◆ 1-2-1 システム開発のアプローチ
- ◆ 1-2-2 データ中心アプローチ

## 1-1 データベースの基本

データベースとは、いろいろなシステムで使われるデータを1か所に集めて、データ活用の利便性を高めたものです。ここではまず、データベースの成り立ちとデータの格納方法について基本を押さえておきます。

### 1-1-1 データベースとは



#### 勉強のコツ

データベースにはいろいろな製品がありますが、ほとんどが関係モデルを基にした関係データベースです。様々なデータモデルについて学習しつつ、なぜ関係モデルが普及したのか知っておくと、その利点がよく分かります。



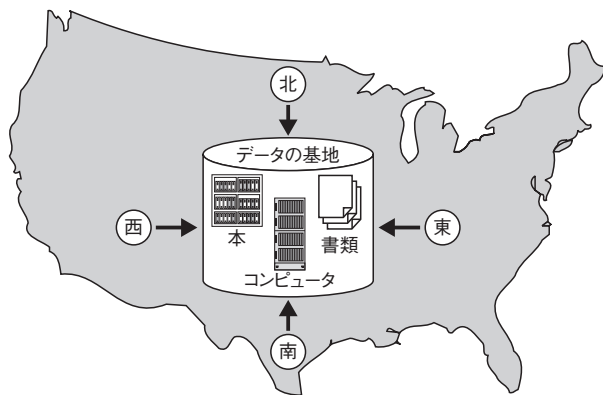
#### 発展

ネットワーク分野でよく知られているTCP/IPプロトコル群も、別名をDoDモデルといい、DoDが作成したモデルです。データベースやネットワークなど現在のITの基本になる技術は、ほとんどがアメリカの軍事産業での研究に基づくものです。

#### 動画

この「データベースとは」などを学ぶ動画を公開しています。以下にまとめてありますので、参考にしてください。  
<http://www.wakuwaku-studyworld.co.jp/blog/dbinfo/>

データベースの語源は、1950年頃のDoD（Department of Defense：米国国防総省）において、各所に分散していた軍事情報などを1か所に集め、そこに行きさえすればすべての情報を見られるようにした「データの基地（Data Base）」からとられたという説が一般的です。



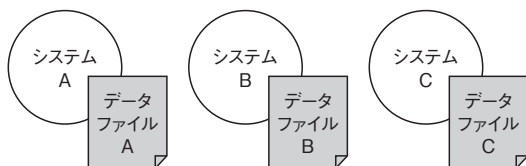
データの基地 (Data Base)

### データを一か所で集中管理

データベースでは、データを1か所に集めることで、情報を集中的に管理でき、そのデータを活用しやすくしています。統合することで検索しやすくなり、また同じデータがいくつも重複することを避けられるので、業務を効率化することができます。

## ■ データ独立

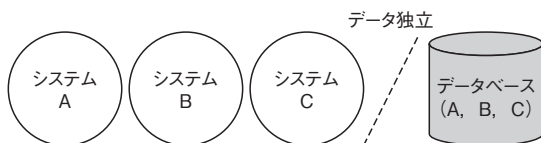
昔のシステム開発では、システムごとにデータを保存してファイルなどに格納していました。次のようなイメージで、それぞれのシステムが別々にデータを管理していたのです。



それぞれのシステムでデータを別々に管理

この方法では同じようなデータを重複してもつことになり、また、データの受け渡しの効率も悪くなります。

そこで、システムからデータを独立させ、そのデータをまとめてデータベースに入れるという方法が考え出されました。次のように、システムとは別にデータベースを用意します。



システムからデータを独立させ、データベースにデータを格納

システムからデータを独立させて別に管理することを**データ独立**といいます。データベースは、このようにシステムから独立したデータを扱うために必要な仕組みです。



昔のシステムには、データがプログラムと結び付いているものが多くありました。このようなシステムだと、プログラムを変更するとデータも変更しなければならず、システムの修正が大変でした。そこで、プログラムの変更がデータに影響を与えないように、データを独立させる手法が考えられました。この手法により、変更に強いシステムが実現できるようになりました。



データ独立の観点から考案されたデータの格納方法が、関係モデルです。次項以降で詳しく解説します。

### ▶▶ 覚えよう！

- ☐ データを1か所に集めることで、業務を効率化できる
- ☐ システムからデータを独立させるデータ独立が大切

## 1-1-2 データモデル

データベースを構築するときには、現実世界にあるデータをデータベースに適切に格納できるように変換する必要があります。データを変換するときのモデルを**データモデル**といいます。



データモデルには、大きな分類として、論理データモデル、概念データモデル、物理データモデルの3種類があります。ここで紹介している階層型、ネットワーク、関係、オブジェクト指向の四つのデータモデルは、論理データモデルにあたります。論理データモデルとは、データベースで実際に実装可能なモデルのことです。

概念データモデル、物理データモデルなど、その他のデータモデルについては、「1-2-2 データ中心アプローチ」で取り上げます。また、概念データモデルについては、第6章で詳しく学習します。

### 主な論理データモデル

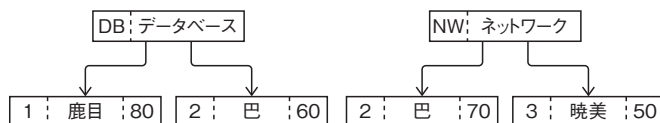
論理データモデルとしては、次の四つがよく知られています。

- ・ 階層型 (ハイアラキカル) モデル
- ・ ネットワークモデル
- ・ 関係 (リレーショナル) モデル
- ・ オブジェクト指向モデル

それぞれのデータモデルの特徴は、次のとおりです。

### 階層型モデル

データを階層型の親子関係で表現する、最も古くからある方法です。データ同士の関係は次のようにポインタ (矢印[→]での関連付け) で表します。

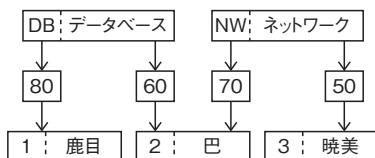


階層型モデル

階層型モデルでは、階層構造の親子関係は1対1または1対多で表します。そのため、子が複数の親と関係をもつことはありません。上の図の場合、巴さんのデータが二つありますが、これらは、「データベース→巴」「ネットワーク→巴」のかたちで別々に管理され、交わることはありません。

## ■ ネットワークモデル

階層型で表現できない、子が複数の親をもつ場合などを網状（ネットワーク）で表現するデータモデルです。データ同士の関係は次のようにポインタで表します。



ネットワークモデル

先ほどの階層型モデルとは異なり、子は複数の親をもつことが可能になります。巴さんのデータは一つで、複数の関係「データベース→60→巴」「ネットワーク→70→巴」をもつことができます。

## ■ 関係モデル

テーブル（表）とテーブル間の関連でデータを表現する方法です。数学の理論（関係理論）を基に考え出されたデータモデルなので、階層型モデルやネットワークモデルとは、考え方がまったく異なります。

データをシステムから切り離す**データ独立**という概念を考えた計算機科学者のエドガー・F・コッドが、数学的、形式的な観点から徹底的に練り上げて作成したのが関係モデルです。

関係モデルでは、データベースを**関係**（リレーション）の集まりだと考えます。その関係を二次元の表にすることで、データとその関係を表現します。

先ほどの階層型モデル、ネットワークモデルの図を関係モデルで表現すると、次のような形式になります。



階層型モデルやネットワークモデルは、システム開発の過程で経験的にでき上がってきたモデルです。そのため、データ独立は考えられておらず、データとシステムは同時に変更する必要があります。関係モデルは、データ独立のために数学的に考えられているモデルなので、データとシステムを独立させることが可能になります。



科目		生徒		受験		
科目	科目名	番号	氏名	科目	番号	点数
DB	データベース	1	鹿目	DB	1	80
NW	ネットワーク	2	巴	DB	2	60
		3	暁美	NW	2	70
				NW	3	50

関係モデル

このように表の形式で表現することで、関係を分かりやすく示すことが可能になります。データ独立が実現すると、システムの変更が容易になり、開発効率が大幅に上がるので、関係モデルは急速に普及していきました。

## ■ オブジェクト指向モデル

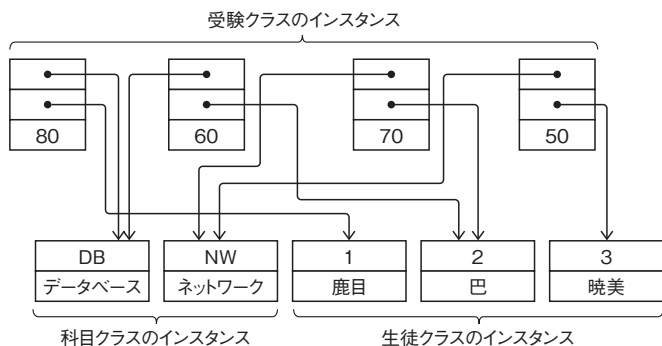
オブジェクト指向とは、データと操作を一体化して扱う考え方です。これに対応したデータモデルが、オブジェクト指向モデルです。

オブジェクト指向ではデータの型であるクラスを作成し、そのクラスからデータの数だけインスタンスを生成します。オブジェクト指向モデルでは、それぞれのクラスのインスタンスを関連付けて、関係を表します。

関係モデルの表と同じデータをオブジェクト指向モデルで表現すると、次の図のようになります。



**インスタンス (Instance)** とは、オブジェクトの実体です。オブジェクト指向では、クラスは実体ではなくデータ型を定義するものです。そこから実際のデータであるインスタンスを生成します。この例では、科目(属性は科目、科目名)がクラスです。インスタンスはその実体(実データ)である(DB, データベース)や(NW, ネットワーク)にあたります。



オブジェクト指向モデル

オブジェクト指向モデルに合わせたオブジェクト指向データベース (OODB : Object Oriented DataBase) も開発されていますが、関係モデルのデータベースにデータを格納することも可能です。その場合には、オブジェクトと関係 (リレーション) を対応付ける **O-R マッピング** が行われます。

**関連**

オブジェクト指向データベースについては、「9-1-3 オブジェクト指向データベース」で改めて取り上げます。

ここまで四つのデータモデルを見てきましたが、その中で関係モデルは、理論的に他のデータベースよりも完成度が高く、一般的によく利用されています。そこで、これ以降では関係モデルを中心に学習していきます。

**覚 え よ う !**

- ☐ 論理データモデルは、階層型、ネットワーク、関係、オブジェクト指向の4種類
- ☐ 関係モデルが最も普及しており、様々なデータベースで利用されている



関係モデルでは、プログラムとしてどのように実装するかをまったく考えていません。そのため、開発された当初は、ネットワークモデルなどの方が高速で、関係モデルは速度が遅いという欠点がありました。しかし現在の関係データベースでは、質問処理の最適化技術が格段に進歩し、処理効率も改善しています。そのため、「とりあえず普通の用途なら関係モデル」という認識が一般的になりました。

## 1-1-3 関係モデル

関係モデルは、数学の集合論を基に、論理的に考え抜かれたモデルです。

### 関係モデルの考え方

関係モデルでは、データもデータ間の関連も、すべて**リレーション**(関係)と**タプル**(行, 組)で表現することを基本にしています。

### リレーション(関係)

ここで、リレーションについて正確に理解しておきましょう。まず、**ドメイン**(定義域)という概念があります。ドメインとは集合のことで、人名の集合、年齢の集合など、様々な集合がドメインです。

例えば、先ほどの関係データベースのうち、関係“科目”について考えてみます。関係“科目”でドメインを表すと、次の二つのドメイン、“科目”と“科目名”が定義されます。

科目 {DB, NW}

科目名 {データベース, ネットワーク}

次にドメインの**直積**を考えます。直積とは、それぞれのドメインを単純にかけ合わせ、すべての組合せを表したものです。科目×科目名の直積は、次のようになります。

(DB, データベース)

(DB, ネットワーク)

(NW, データベース)

(NW, ネットワーク)

これらの各要素、1行1行のことを**タプル**(行, 組)といいます。直積で表されたタプルはすべて必要なわけではなく、実際に存在する組合せのみを取り出します。

このとき、必要なタプルのみを選んだ**直積の部分集合**のことを**リレーション**(関係)といいます。先ほどの直積から、実際に



関係モデルの用語は、同じ意味を複数の言葉で表現しているので、ややこしく感じられることも多いと思います。次のように整理して、一度覚えておきましょう。

表=関係, リレーション  
行=組, タプル  
列=属性, カラム

何度も出てくる言葉なので、忘れたらこのページに戻って見直すことを繰り返していると、そのうち頭に自然に入ってきます。

ある組合せを部分集合として取り出すと、次のようになります。

(DB, データベース)

(NW, ネットワーク)

このリレーションは、表(テーブル)として表すことができ、次のように表現されます。

“科目”表

科目	科目名
DB	データベース
NW	ネットワーク

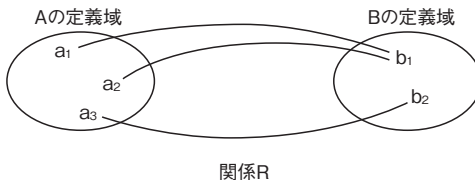
このとき大切なのは各タプルのドメインの組合せであり、その並び順に意味はありません。

また、テーブルの縦の列のことをカラム(列, 属性)と呼びます。

それでは、次の問題で確認してみましょう。

### 問題

図のような関係 $R(A, B)$ において、属性 $A$ の定義域の要素は $\{a_1, a_2, a_3\}$ 、属性 $B$ の定義域の要素は $\{b_1, b_2\}$ である。 $a_1$ と $b_1$ を結ぶ線は、 $(a_1, b_1)$ のように、関係 $R$ の要素を表している。この関係 $R$ の要素を表す語として、適切なものはどれか。



ア 組          イ 属性          ウ ドメイン          エ 列



### 過去問題をチェック

リレーション(関係)やタブル(組)などの用語に関する問題は、データベーススペシャリスト試験の午前Ⅱでよく出題されます。この問のほかには以下で出題されています。

【関係 $R$ での最大のタブル数について】

・平成23年特別 午前Ⅱ 問10

午後で改めて設問として問われることはありませんが、用語は当たり前のように出題文中に出てきます。きちんと理解して、読んだらすぐにイメージが浮かぶくらいにしておきましょう。

## 解説

Aの定義域  $\{a_1, a_2, a_3\}$  と, Bの定義域  $\{b_1, b_2\}$  の直積  $A \times B$  は,  $(a_1, b_1)$ ,  $(a_1, b_2)$ ,  $(a_2, b_1)$ ,  $(a_2, b_2)$ ,  $(a_3, b_1)$ ,  $(a_3, b_2)$  の六つです。

このうち, 関係(リレーション) Rとして  $(a_1, b_1)$ ,  $(a_2, b_1)$ ,  $(a_3, b_2)$  の三つの要素を取り出します。この一つ一つの要素のことをタプル(組, 行)といいます。選択肢の中ではAの組が当てはまるので, Aが正解です。

イの属性とエの列にはAとBが, ウのドメインには定義域が対応します。

《解答》ア

## 覚 え よ う !

- ☐ ドメイン(定義域)の直積の部分集合がリレーション(関係, 表)
- ☐ リレーション(関係, 表)には, タプル(行, 組)とカラム(列, 属性)がある



## 新しくなりつつある データベーススペシャリスト試験

データベーススペシャリスト試験は、平成7年(1995年)にスタートして以来、15年ぐらいはほとんど出題傾向が変わらない試験でした。データベース基礎理論、特に関係モデルについては数学理論がベースになっており、時代の流れで変化することがなかったからです。そのため、他の高度区分に比べて、特に午後試験の傾向がまったく変わらず、基礎理論やデータベース設計が中心に出題されていました。

しかし、平成23年頃から少しずつ傾向が変わってきています。基礎理論やデータベース設計について出題されることは同じなのですが、その比率が下がりました。その分、情報セキュリティや大量のデータ、分散したデータの扱い方など、実際的な内容がいろいろ出題されるようになりました。

これは、近年のクラウドシステムなどを中心としたシステム環境の変化により、データベースを構築するためには専門分野以外のことも知る必要が出てきたからだと考えられます。データベーススペシャリストだからといって、データベースについてだけ知っていればよいという時代ではなくなってきたのです。

データベースの中には個人情報が含まれることもありますので、当然、情報セキュリティを意識する必要が出てきます。そういった業務の幅広さをカバーするためにも、データベーススペシャリスト試験で様々なことが問われるようになったと考えられます。

といっても、関係モデルの考え方は普遍的な知識ですので身に付ける必要があることは変わりません。その上で、現状に対応して幅広く知識を積み重ねていきましょう。



### 勉強のコツ

データベーススペシャリスト試験の傾向は、5年前ぐらいいから少しずつ変化してきました。

そのため、過去問演習を行うときには、なるべく直近の問題を使用することが効果的です。

過去3年分をしっかりと演習すれば確実に実力は付きますので、あまり昔の過去問にこだわらずに最近の問題を学習するようにしましょう。

## 1-1-4 ■ データベースの三値論理

データベースでは、三値論理という、通常とは異なる論理を使用します。真、偽のほかに第3の値としてNULLを利用することで、いろいろな状況を表現できます。



第2章で詳しく解説する主キーには、NULLを使用できません。しかし、候補キーにはNULLが含まれていてもかまいません。このように、用語の定義を学習するとき、NULLが使用可能かどうかを押さえておくに役立ちます。

### ■ 三値論理

データベースを扱う上での大切な考え方に、**三値論理**があります。通常、論理は「真(true)」か「偽(false)」の二値で表されるもので、この考え方を二値論理といいます。

三値論理では、真、偽のほかに第三の論理値をもちます。この第三の値として、データベースでは**NULL**を使用します。このNULLは特別な値として様々な場面で利用されます。

### ■ NULLの利用方法

NULLの代表的な利用方法には、次の二つがあります。

- ①**不明・未知** (Unknown)
- ②**非存在・適用不能** (Inapplicable)

①の**不明・未知**とは、「値は存在するはずだが、どのような値か分からない」という場合を指します。例えば、顧客の表に氏名という属性があり、その値が分からないときにはNULLを設定します。

②の**非存在・適用不能**は、「値が存在しないので、そのことを示す」場合に使われます。例えば、図書館の書籍貸出表で、返却年月日という属性があったとします。書籍を貸し出す際には返却年月日にNULLを設定し、返却されたときに返却年月日にその日の日付を設定します。このときのNULLは、「返却年月日が存在しない(=返却していない)」という意味を示すことになります。

①と②のどちらの場合でNULLを使用するかは、データの種類によって変わってきます。例えば、メールアドレスの欄がNULLになっている場合には、メールアドレスをもっていないことを示すのかもしれませんが、メールアドレスが分からないことを示すのかもしれませんが。こういったデータ特有の意味は、NULLを使用するときにあらかじめ定義しておく必要があります。

## ■ NULLを利用する際の注意点

NULLを扱う上での一番の注意点は、NULLは値ではないということです。そのため、NULLを含むデータを取り扱うときには注意が必要です。

具体的には、あるカラム(列)のデータ数を数えるときに、NULLの行はカウントされません。また、以上、以下などの条件指定で行を抽出するときに、NULLとの演算では条件を満たすことはありません。さらに、NULLを含む演算をAND、ORなどで結び付けると、全体がNULLになり、結果が不明になることがあります。

そのため、NULLを含む演算では、意図した結果が得られないというトラブルがよく生じます。NULLは第三の値であり、通常の値とは異なる扱いになるので、その性質を知って有効活用していきましょう。



データベース内部では、NULLは値とは別に、「NULLかどうか」についての情報を保管するメモリ領域で管理されることが多いです。そのため、SQLでNULLを含む値を検索するときには、「列名=NULL」ではなく「列名 **IS** NULL」というように、「NULLかどうか」を判定する必要があるのです。

### ▶▶ 覚えよう！

- ☐ データベースの三値論理では、真、偽に加えてNULLを使用する
- ☐ NULLは値ではなく、不明・未知や、非存在・適用不能を示すために利用する



## 1-2 システム開発とデータベース

データベースは、単独でデータを保存するわけではなく、何らかのシステムから利用されることがほとんどです。そのため、システム開発とデータベースは切っても切り離せない関係にあります。ここでは、データベースを活用するシステムの開発手法を取り上げます。



### 勉強のコツ

データベーススペシャリスト試験で出てくる内容は、ほとんどが「システム開発におけるデータベース設計」です。

そのため、システム開発の基本についてはひととおり知っておく必要があります。システム開発の知識や経験がまったくない方は、基本的な開発手法などについて学習しておきましょう。応用情報技術者試験で出題されるレベルの知識があれば大丈夫です。

### 1-2-1 システム開発のアプローチ

システム開発において、目的のシステムを作っていく方法は複数あります。この方法のことをシステム開発のアプローチといいます。

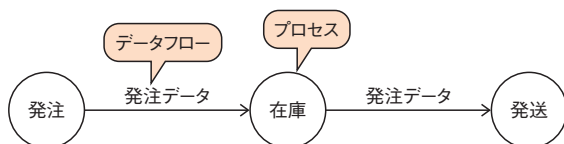
#### システム開発の三つのアプローチ

システム開発のアプローチには、大きく分けて次の3種類があります。

##### ①プロセス中心アプローチ

プロセス中心アプローチ (POA : Process Oriented Approach) とは、ソフトウェアの機能 (プロセス) を中心としたアプローチです。機能を一つのプロセスと考え、そのプロセスを段階的に詳細化していき、最終的には最小機能の単位であるモジュールに分割していきます。

図法としては、データの流れ (データフロー) を表現する **DFD** (Data Flow Diagram) や、プロセスの状態遷移を表現する **状態遷移図** などがよく用いられます。DFDでは、データフローを矢印で表し、プロセス間で受け渡されるデータを記述します。

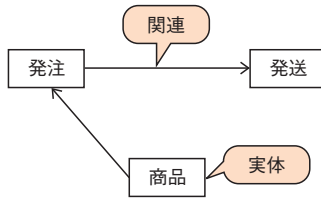


プロセス中心アプローチでのDFDの例

## ②データ中心アプローチ

データ中心アプローチ (DOA: Data Oriented Approach) とは、業務で扱うデータに着目したアプローチです。まず、業務で扱うデータ全体をモデル化し、データベースを設計します。個々のシステムはこのデータベースを中心に設計することによって、データの整合性・一貫性が保たれ、システム間のやり取りが容易になります。プログラミングとデータベースを分離するデータ独立の考え方が基になっています。

図法としては主に、実体(データ)と実体間の関連を表現する **E-R図** (Entity-Relationship Diagram) が用いられます。

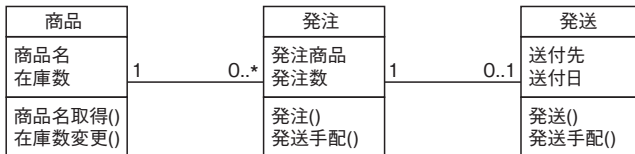


データ中心アプローチでのE-R図の例

## ③オブジェクト指向アプローチ

オブジェクト指向アプローチ (OOA: Object Oriented Approach) とは、プログラムやデータをオブジェクトとしてとらえ、それを組み合わせてシステムを構築するアプローチです。

図法としては、クラス図やシーケンス図などの **UML** (Unified Modeling Language) が用いられます。クラス図のクラスでは、データ(属性)とメソッド(操作)を一体化してカプセル化します。データを操作するためには、対応するクラスのメソッドを利用する必要があります。



オブジェクト指向アプローチでのクラス図の例



情報処理技術者試験のうちデータベーススペシャリスト試験では、主にデータ中心アプローチでのシステム開発について出題されます。その他のプロセス中心アプローチやオブジェクト指向アプローチでのシステム開発については、主にシステムアーキテクト試験で出題されます。そのため、DFDやUMLはシステムアーキテクト試験の定番であり、E-R図はデータベーススペシャリスト試験の定番です。システム開発を極めたい方は、両方とも学習してみることをおすすめします。



**CRUD 分析**とは、機能とテーブルを Create（作成）、Read（参照）、Update（更新）、Delete（削除）の四つの操作で関連付ける分析です。関連付けることで、「作成」がないのに「参照」があるといったデータのライフサイクルの矛盾がないことを確認します。

詳しくは、「3-1-6 データベースシステム設計」で解説します。

## データベーススペシャリスト試験での出題

データベーススペシャリストの試験では、3種類のアプローチのうち、主にデータ中心アプローチでのデータベース設計を取り扱います。そこで、データ中心アプローチについては次項で詳しく学習します。

また、実際のシステム開発は、一つのアプローチだけでなく複数のアプローチによって行い、融合することが多いため、試験でも、プロセス中心アプローチで作成されたプロセス（機能）と、データ中心アプローチで作成されたエンティティ（テーブル）を対応させる CRUD 分析など、開発アプローチを融合させる問題がときどき出題されます。

### 覚 え よ う !

- ☐ システム開発のアプローチは、プロセス中心アプローチ、データ中心アプローチ、オブジェクト指向アプローチの三つ
- ☐ データベーススペシャリスト試験では、データ中心アプローチで設計を行う



## データベーススペシャリストは、主にデータベース設計を行う人

データベーススペシャリストという試験は、その名称から、「データベースを直接さわって、設定を行う人」が対象というイメージをもたれがちですが、実際には、「システム開発において主にデータベース設計を行う人」という位置づけです。

データベースに関する専門職には、次の二つがあります。

- ・ DA (Data Administrator) : データ管理者
- ・ DBA (DataBase Administrator) : データベース管理者

DAは、データそのものの管理を行う人です。データベースの中にどのようなデータをどれだけ格納するのかを決めます。

DBAは、データベースの管理を行う人です。データベースを構築し、運用保守も行います。データの内容によっては暗号化などが必要になる場合もあるため、個人情報保護などの情報セキュリティについても意識する必要があります。

システム開発の工程では、データベース設計などの上流工程をDAが行い、物理設計や運用保守などの下流工程をDBAが行うという役割分担です。

データベーススペシャリストは、この二つのうち、主にDAに該当する人のための試験です。そのため、データ中心アプローチでのデータベース設計などについて主に出題されます。

DBAに関する問題もある程度出題されるので勉強する必要がありますが、メインではありません。DBAが行うことはDBMS(データベース管理システム)に大きく依存するので、国家試験であるデータベーススペシャリスト試験ではDBMSに関する内容は、多くのDBMSに共通する部分を中心に出題されます。代わりにベンダ資格が充実していて、OracleならOracle Master, SQL ServerならMCSA : SQL Server, PostgreSQLなどのオープンソースの場合はOSS-DBなど、DBMSごとに資格試験が用意されています。

自分の状況に合わせて、必要な勉強をしていきましょう。



### 勉強のコツ

データベーススペシャリスト試験はDAの試験なので、DAの勉強をメインで行うことがとても大切になります。具体的には、データベース設計をしっかりと学習して、正規化ができてER図が書けるようになることが肝心です。

実務でデータベースに関わる場合には、DAとDBAに関する両方のスキルが必要になることが多いです。

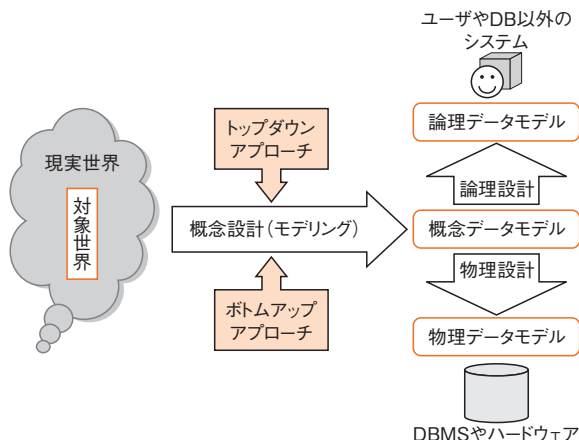
データベーススペシャリストの資格を取得したら、他のベンダ試験に挑戦すると、データベースのスキルをさらにアップさせることができます。また、データベーススペシャリストの勉強と並行してベンダ試験の勉強を行うのも、相乗効果があるのでおすすめです。

## 1-2-2 データ中心アプローチ

データベーススペシャリスト試験では、データ中心アプローチでのシステム開発が取り上げられます。データ中心アプローチでは、データベース設計として、概念設計、論理設計、物理設計を行います。

### データ中心アプローチの流れ

データ中心アプローチでのシステム開発は、以下の図のような流れで行います。



データ中心アプローチでのシステム開発の全体像



関連

トップダウンアプローチやボトムアップアプローチなど、概念設計の具体的な手法については、第3章「データベース設計」で学びます。論理設計、物理設計については、第7章「論理設計・物理設計」で学びます。ここでは全体像を理解して、各章に入ったときに全体のどの部分のことを説明しているのかイメージできるようにしておきましょう。

システムを作成するときには、開発する対象を絞り込むため、まず現実世界から、開発を行う対象世界を決定します。

その対象世界に対して、**概念設計**（モデリング、またはデータモデリング）を行います。概念設計には、**トップダウンアプローチ**と**ボトムアップアプローチ**の二つの手法があり、両方を組み合わせて設計を行います。概念設計の結果、でき上がるのが、**概念データモデル**です。

次に、でき上がった概念データモデルに対して**論理設計**を行います。論理設計は、データベースとユーザやデータベース以外のシステムとを結び付けるための設計です。論理設計ででき

上がるデータモデルを**論理データモデル**といいます。

さらに、概念データモデルに対して**物理設計**を行います。物理設計は、データベースとDBMS（データベース管理システム）やハードウェアを結び付けるための設計です。物理設計でできるデータモデルを**物理データモデル**といいます。

## ■ 論理データ独立と物理データ独立

データモデルを3層に分ける理由は、**データ独立**を行って、変更に強いシステムにするためです。

概念データモデルとは別に論理データモデルを作成することによって、互いの独立性が保たれます。例えば、データベース以外のシステムの変更があった場合には、論理データモデルを変更するだけでいいので、概念データモデルには影響を及ぼしません。逆に、データベースの構造など、概念データモデルに変更があった場合でも論理データモデルは変更されないで、システムに影響が及ぶことはありません。このような、概念データモデルと論理データモデルの間のデータ独立のことを**論理データ独立**といいます。

同様に、概念データモデルとは別に物理データモデルを作成することによって、互いの独立性が保たれます。例えば、DBMSやハードウェアに変更があった場合でも、物理データモデルを変更すればいいので、概念データモデルに影響を与えません。このような、概念データモデルと物理データモデルの間のデータ独立のことを**物理データ独立**といいます。

概念データモデルは、DBMSやシステムにはまったく依存しない机上のデータモデルです。そのモデルを、データ独立を確保しつつ、論理データモデルや物理データモデルで実装することによって、システム変更による影響箇所が少ない、変更に強いシステムを構築することが可能になります。

## ■ 3層スキーマアーキテクチャ

データベースを3層に分ける方式は、前述した3層のデータモデルのほかにもあります。ANSI（American National Standards Institute：アメリカ規格協会）で標準化されたANSI/X3/SPARCの**3層スキーマアーキテクチャ**です。



ANSI/X3/SPARCのSPARCは、Standards Planning And Requirements Committee（標準化計画委員会）の略です。3層スキーマアーキテクチャは、この委員会のスタンディグループが制定したことから付けられた名称です。

スキーマとは、データベースの構造であり、DBMSでサポートされる言語(SQLなど)で定義される具体的なものです。ANSI/X3/SPARCの3層スキーマアーキテクチャでは、次の三つのスキーマが定義されています。



データベーススペシャリストの午後試験では、**関係スキーマ**を記述する問題が多く出てきます。関係スキーマは、3層スキーマアーキテクチャでは概念スキーマに当たり、関係データベースでの関係(つまり、表(テーブル))を記述するのに使われます。

### ①外部スキーマ

データベースに関係ないプログラムやユーザが使用する、データベースの記述です。代表的なものに**ビュー**があります。

### ②概念スキーマ

エンティティやテーブル、テーブル間の関連などの記述です。内部スキーマと外部スキーマの間に位置します。

### ③内部スキーマ

DBMSで使用する、データベースを物理的にどのように配置するのかなど、具体的に実装するための記述です。代表的なものに**インデックス**があります。

それでは、次の問題で確認してみましょう。

## 問題

データベースの3層スキーマアーキテクチャに関する記述として、適切なものはどれか。

- ア 概念スキーマは、内部スキーマと外部スキーマの間に位置し、エンティティやデータ項目相互の関係に関する情報をもつ。
- イ 外部スキーマは、概念スキーマをコンピュータ上に具体的に実現させるための記述であり、データベースに対して、ただ一つ存在する。
- ウ サブスキーマは、複数のデータベースを結合した内部スキーマの一部を表す。
- エ 内部スキーマは、個々のプログラム又はユーザの立場から見たデータベースの記述である。

(平成27年春 データベーススペシャリスト試験 午前Ⅱ 問1)

## 解説

データベースの3層スキーマでは、概念スキーマは、内部スキーマと外部スキーマの間に位置します。そして、エンティティやデータの関連に関する情報をもつので、アが正解です。

イは内部スキーマ、エは外部スキーマを指します。

ウのサブスキーマは、CODASYL (Conference on Data Systems Languages) という、COBOLを開発した団体が定義する3層スキーマでの用語です。サブスキーマは、外部スキーマの一部に対応します。

## 《解答》ア

### ▶▶ 覚えよう！

- ☐ 3層スキーマでは、外部スキーマ、概念スキーマ、内部スキーマがある
- ☐ 外部スキーマではビュー、内部スキーマではインデックスを定義する



データモデルやスキーマには、いろいろな分け方があります。3層スキーマや3層データモデルにも多くの種類があり、分け方が2層になっているものもあります。しかし、これらの違いを細かく覚える必要はありません。大切なのは、階層化してデータ独立をさせるという概念です。データベーススペシャリスト試験で問われるのは、基本的にANSI/X3/SPARCの3層スキーマアーキテクチャだけなので、それだけしっかりと押さえておきましょう。





# 第2章

## データベース基礎理論

この章では、関係データベースの理論的な基本となるデータベース基礎理論について学びます。

関係データベースにおける最も重要な理論に、正規化理論があります。更新に強いデータベースにするために、第1正規形、第2正規形、第3正規形、及び高次の正規形へと正規化を行う手法について、順番に学んでいきます。

また、関係データベースでの演算である関数演算についても学びます。さらに、データベーススペシャリスト試験の午後Iでは、データベースの基礎理論に関する問題が定番で出題されますので、その問題の解き方、考え方について、実際の問題を例に学んでいきます。

### 2-1 正規化理論

- ◆ 2-1-1 正規化とは
- ◆ 2-1-2 第1正規形
- ◆ 2-1-3 第2正規形
- ◆ 2-1-4 第3正規形
- ◆ 2-1-5 高次の正規形

### 2-2 関係演算

- ◆ 2-2-1 関係演算の種類
- ◆ 2-2-2 結合演算

### 2-3 問題演習

- ◆ 2-3-1 午前問題
- ◆ 2-3-2 午後問題
- ◆ 2-3-3 基礎理論 午後I問題の傾向

## 2-1 正規化理論

正規化理論とは、関係データベースでデータを適切な関係(リレーション)に分解するための理論です。第1, 第2, 第3, ボイスコッド, 第4, 第5正規形の六つの正規形があり、順に正規化を行っていきます。



### 勉強のコツ

データベーススペシャリスト試験では、正規化は合格への一番のポイントです。正規化のルールを知っているだけでなく、素早く正確に正規化を行うことが求められます。

単なる知識で終わらず、「体で覚える」レベルまで、正規化の演習を行っていきましょう。



### 用語

更新時異状(update anomaly)の異状(anomaly)とは、「通常とは違った状態」のことです。

データベースのデータが、更新によって本来の姿とは違った状態になることが更新時異状です。

異常(abnormal)とは違うので、漢字を間違えないように押さえておきましょう。

### 動画

正規化の目的について解説した動画を公開しています。  
<http://www.wakuwakuacademy.net/itcommon/3/>  
 リストから「正規化の目的」を選択してください。

## 2-1-1 正規化とは

正規化(Normalization)は一般的に、一定のルールに従って変形を行うことです。関係データベースの正規化では、正規化理論というルールに従って、関係を分解するという変形を行っていきます。

### 正規化の目的

関係データベースでの正規化の目的は、タプルの更新時に起こる異常である**更新時異状**を排除することです(詳細は後述します)。関係データベースでは、正規化を行わないと、更新時に様々な異状が発生します。このときの更新とは、単にデータを修正することだけではなく、タプルの**挿入**、タプルの**更新(修正)**、タプルの**削除**の三つを指します。

また、正規化を行うことによって、一つのデータは1か所のみが存在する「**1か所1事実(1 fact in 1 place)**」が実現できます。データの重複をなくすことによって、データの更新時に複数か所を更新する必要がなくなり、整合性のあるデータベースが作成できます。

### 正規化を行わない場合

正規化を行うことで、更新時異状は排除できます。しかし、逆に、**更新しない場合には正規化は行う必要がない**ともいえます。

正規化を行うことは、実際の作業としてはテーブルを分解することです。そうすると、データを扱うときには表の結合操作が必要になるので、処理が遅くなることが多いのです。正規化によって性能が下がることが多いため、あえて正規化を行わないこともよくあります。

一般に、正規化を行わないデータベースには、次の3種類があります。

### ①データの更新を行わないもの

データウェアハウスやアクセスログなど、データを追記するだけで更新を行わないものは、正規化の必要がありません。

### ②データの履歴を残すもの

社員異動履歴や単価改変履歴など、古いデータの履歴を残すものは更新されては困るため、正規化は行いません。

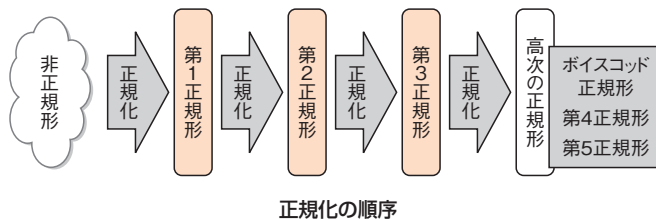
### ③高速化が特別に必要なもの

データの更新時異状よりも高速化が優先される場合、あえて正規化を行わないことがあります。この場合には、更新時異状への対策を別途考える必要があります。

これら以外の場合には、データベースに格納されるデータを不具合なく整合性のとれたものにするために、正規化を行うことが必要になります。

## ■ 正規形の種類

正規化を行うことでできる関係(リレーション)を**正規形**といいます。正規形には、第1正規形、第2正規形、第3正規形、ボイスコード正規形、第4正規形、第5正規形の6種類があります。基本的には、次のように順番に正規化を行っていきます。



ほとんどの場合は**第3正規形までの正規化**を行います。ボイスコード正規形、第4正規形、第5正規形については、必要となる場面が少ないので、場合に応じて考慮します。



関連

データウェアハウスは、分析のためのデータを他のデータベースから取得します。このとき、データの更新は基本的に行わず、新しいデータを追加するのみです。そのため、正規化は行わず、分析するデータはファクトテーブルと呼ばれる一つのテーブルにまとめられます。データウェアハウスについては、「9-1-3 データウェアハウス」で詳しく取り上げます。



### 過去問題をチェック

更新時異状については、データベーススペシャリスト試験の午後問題で、「タプル挿入時」「タプル更新時」「タプル削除時」を区別して出題されます。例えば、以下の出題があります。

**【タプル挿入で起こる問題について】**

- ・平成21年春 午後Ⅰ 問1 設問2 (4)
- ・平成20年春 午後Ⅰ 問1 設問2 (3) (テクニカルエンジニア(データベース)試験)

## 更新時異状

更新時異状は、正規化を行っていないときに起こるタプルの更新時の異状です。それぞれの正規形で、正規化が行われていない場合に、次の3種類の更新時異状が起こることが考えられます。

- ①タプル挿入時異状
- ②タプル更新(修正)時異状
- ③タプル削除時異状

それでは、次の問題を例に上記の更新時異状を考えてみましょう。

### 問題

教員の担当科目と給与を管理する“科目－教員”表を更新するときに発生する問題はどれか。ここで、科目番号を主キーとし、基本給は科目によらず教員ごとに決まっているものとする。

科目－教員

科目番号	科目名	教員番号	担当教員	単位	基本給
2761	一般システム理論	8823	田中亮	2	180
2762	問題形成と問題解決	6673	佐藤永吉	2	250
2763	情報システム開発の経済性	6654	小林正路	2	400
2864	一般システム理論	7890	大野俊郎	2	230
2865	情報システムの都市計画法	4664	斉藤秀夫	4	320
3966	UML モデリング	8823	田中亮	4	180

- ア ある教員が唯一担当していた科目の行を削除すると、その教員の基本給データだけが残ってしまう。
- イ ある教員の基本給を変更するには、該当する行を1件ずつコミットしないとデータの不整合が生じる。
- ウ 担当科目のない教員の基本給を登録するときは、一つ以上の科目を削除しなければならない。
- エ 複数の科目を担当する教員の基本給を変更するときは、担当するすべての科目について変更しないとデータの不整合が生じる。

(平成22年春 データベーススペシャリスト試験 午前Ⅱ 問7)

## 解説

“科目－教員”表の場合、次の3種類の更新時異状があります。

### ①タプル挿入時異状

新しい教員を登録するときに、担当科目が決まっていない場合には登録できません。“科目－教員”表では主キーが科目番号で、主キーはデータが必須となるので、科目番号のないタプルは挿入できないからです。

### ②タプル更新(修正)時異状

複数の科目を担当する教員の情報を更新するときに、該当する複数の箇所をすべて一度に更新しないと、データの不整合が生じます。具体的には、科目番号2761のタプルで教員番号8823の田中亮の基本給180を200に更新し、同じ田中亮が担当する科目番号3966のタプルの基本給を更新しなかった場合には、データが矛盾してしまいます。

### ③タプル削除時異状

一つの科目だけを担当する教員の情報は、その科目の情報が削除されるとなくなります。具体的には、科目番号2762のタプルを削除すると、教員番号6673の佐藤永吉の情報もすべて消えてしまいます。

以上から、タプル更新時異状の説明であるエが正解です。

- ア 行(タプル)を削除すると起こるのは、その教員のデータがすべて削除されてしまうことです。
- イ 1行ずつコミットを行うとデータの不整合が発生するので、すべての該当する行を更新したあとにまとめてコミットする必要があります。
- ウ 担当科目のない教員を登録する場合には、一つ以上の科目について登録を行う必要があります。



発展

更新時異状については、どの正規形でも、正規化を行わないと、「タプル挿入」「タプル更新」「タプル削除」時の異状が起こります。

この間の“科目－教員”表は第2正規形であり、第3正規形ではありません。そのため、第3正規形になっていないことによる更新時異状が発生します。

更新時異状の内容は、データの内容によらず正規形によって機械的に決まってきます。そこで、各正規形でどのような更新時異状が起こるのがあらかじめ理解しておく、問題を解くときの助けになります。

2



用語

コミット(COMMIT)とは、トランザクションで変更を確定させることです。トランザクションについての詳細は、「5-2-1 トランザクション管理」で詳しく取り扱います。

《解答》エ



関数従属性の「関数」とは、数学で学ぶ関数  $Y=f(X)$  のように、ある値  $X$  が決まると、その値に対応して  $Y$  が一意に決まるという関係です。関数従属性をより正確に言うと、「関係(リレーション)の属性集合間で、一方の属性集合の値の集合が、他方の属性の値の集合を関数的に決定すること」です。

このとき、すべての属性の値で、この関係が成り立っている必要があります。右の例では、すべての科目名に、対応する科目番号があることが必要です。

## 関数従属性

正規化を理解するための大切な考え方の一つに、**関数従属性**があります。関数従属性とは、ある属性  $X$  の値が決まれば、別の属性  $Y$  の値が一意(一つ)に決まる、といった性質です。これを、 $X \rightarrow Y$  と表します。このとき、 $X$  を**決定項**、 $Y$  を**従属項**といいます。

例えば、先ほどの問題にある“科目-教員”表では、科目番号の値が“2761”に決まれば科目名は“一般システム理論”に、科目番号が“2762”なら科目名は“問題形成と問題解決”に一意に決まります。これが関数従属性であり、「科目番号→科目名」と表すことができます。

データベーススペシャリストの午後試験では、次のように関数従属性を図に表すことがあります。



関数従属性の表記法の例

それでは、次の問題を考えてみましょう。

## 問題

関数従属に関する記述のうち、適切なものはどれか。ここで、 $A$ 、 $B$ 、 $C$ はある関係の属性の集合とする。

- ア  $B$  が  $A$  に関数従属し、 $C$  が  $A$  に関数従属すれば、 $C$  は  $B$  に関数従属する。
- イ  $B$  が  $A$  の部分集合であり、 $C$  が  $A$  に関数従属すれば、 $C$  は  $B$  に関数従属する。
- ウ  $B$  が  $A$  の部分集合であれば、 $A$  は  $B$  に関数従属する。
- エ  $B$  と  $C$  の和集合が  $A$  に関数従属すれば、 $B$  と  $C$  はそれぞれが  $A$  に関数従属する。



関数従属性の種類には、部分関数従属性、推移的関数従属性などがあります。これらを排除することは、第2、第3正規形の条件となります。

## 解説

BとCの和集合  $\{B, C\}$  がAに関数従属する  $A \rightarrow \{B, C\}$  が成り立つときには、 $A \rightarrow B$ 、 $A \rightarrow C$ が成り立ちます。具体例としては、A {科目番号}、B {科目名}、C {単位} という属性集合を考え、 $\{B, C\}$  (科目名, 単位) の属性の集合がA (科目番号) に関数従属するなら、 $A \rightarrow B$ 、 $A \rightarrow C$ は単独でも成り立ちます。したがって、エが正解になります。

アでは、 $A \rightarrow B$ 、 $A \rightarrow C$ のときに  $B \rightarrow C$  が成り立つかどうか問われています。こちらは、BとCには特に関係はないので成立しません。

イでは、BがAの部分集合で、 $A \rightarrow C$ の場合に  $B \rightarrow C$  が成り立つかを問われています。具体例として、前掲の問題を例に、A {科目番号, 教員番号}、B {教員番号}、C {科目名} を考えてみます。BはAの部分集合ですが、Bの教員番号が決まっても、Cの科目名は一意には決まりません。

ウも、イと同様の具体例で考えてみます。BはAの部分集合ですが、 $B \rightarrow A$ は成り立ちません。教員番号が決まっても、科目番号は一意に決まらないからです。

## 《解答》エ



### 勉強のコツ

この間では、前掲の問題のテーブル“科目-教員”表を具体例として、属性集合を表してみました。抽象的な  $A \rightarrow B$  といった話だと分かりにくいので、このような具体例を考えてみると、理解しやすくなります。

2

## ■ 導出属性の排除

データベースの属性の中には、他の属性から演算を行うことによって導くことができる**導出属性**があります。例えば、伝票などでよくある、単価×数量＝金額という計算で求められる金額は、導出属性です。

導出属性は正規化の過程で取り除かれることが多いのですが、これは、導出属性が残っているとデータの変更のたびに導出属性を計算し直さなければならず、更新時異状が起こる可能性があるためです。しかし、前述した正規化を行わない場合と同様に、更新しない場合や履歴を残す場合、性能を上げる必要がある場合には、あえて導出属性を残すこともあります。

導出属性についても、正規化と同様、**臨機応変に考えていく**ことが大切です。



### 過去問題をチェック

導出属性は削除するのが基本ですが、データベーススペシャリスト試験の問題でもあえて残していることがよくあります。以下の例を見てみましょう。

#### 【導出属性を残す例】

・平成23年特別 午後Ⅱ 問2  
表4中の“補正数量”は、棚卸数量と倉庫在庫数量の差で求められる導出属性です。しかし、設問(4)空欄zの解答例ではこの属性は記載されており、削除されていません。更新しない値なので、毎回検索するよりも残しておいた方が効率がいいからだと考えられます。



## 情報無損失分解

正規化を行うときに、ただ更新時異状を避けるために適当に分解すると、もともとあった情報が失われることがあります。関係を分解するときには、その関係を結合することで元の関係を復元できる必要があります。



自然結合については、「2-2-2 結合演算」で詳しく学習します。

ここでは、共通の属性を使って二つの関係を結合できることを理解しておきましょう。

分解した関係を自然結合したときに元に戻せる分解の仕方を**情報無損失分解**といいます。情報無損失分解を行うために正規化の規則があり、第1正規形から順に正規化していくことによって、情報を保持することができます。

それでは、次の問題を考えてみましょう。

### 問題

次の表を情報無損失分解したものはどれか。ここで、下線部は主キーを表す。

発注伝票 (注文番号, 商品番号, 商品名, 商品単価, 注文数量)

ア 発注 (注文番号, 注文数量)  
商品 (商品番号, 商品名, 商品単価)

イ 発注 (注文番号, 注文数量)  
商品 (注文番号, 商品番号, 商品名, 商品単価)

ウ 発注 (注文番号, 商品番号, 注文数量)  
商品 (商品番号, 商品名, 商品単価)

エ 発注 (注文番号, 商品番号, 注文数量)  
商品 (商品番号, 商品名, 商品単価, 注文数量)

(平成22年春 データベーススペシャリスト試験 午前Ⅱ 問6)

## 解説

情報無損失分解では、自然結合すると元の関係を戻すことができます。そのため、各選択肢で関係“発注伝票”が再現できるかどうかを確認していきます。

- ア 関係“発注”と“商品”に共通の属性がなく自然結合できないので、情報が損失しています。
- イ 関係“発注”で、注文番号のみから注文数量が特定できる関係になっており、商品ごとに決まる注文数量の情報が保持できないので、情報が損失しています。
- ウ 商品番号で結合すると関係“発注伝票”が復元できるので、情報無損失分解です。
- エ 注文数量の列が複数あり、特に関係“商品”の注文数量は、注文番号ごとに異なる注文数量の情報を保持できないので、情報が損失しています。

したがって、情報無損失分解はウのみとなり、これが正解です。この分解は、第2正規形への正規化を行うと実現できます。

### 《解答》ウ

## ■ メタデータ

データには、データそのもののほかに、データについてのデータであるメタデータがあります。あるデータが付随してもつ、データ自身に関する抽象度の高いデータです。例えば、関係スキーマ **商品 (商品番号, 商品名)** があったときに、商品番号、商品名は属性名で、商品は関係名です。この商品番号、商品名に対する“属性名”、商品に対する“関係名”がメタデータになります。

データモデルの定義では、このメタデータを**メタ属性**として扱い、属性の一つとすることがあります。



### 過去問題をチェック

メタ属性については、データベーススペシャリスト試験の午後問題の設問の一つでときどき取り扱われます。

#### 【メタ属性について】

・平成22年春 午後Ⅰ 問1 設問3

この設問では、データモデルを拡張し、メタ概念を導入してメタ属性を定義しています。

### ▶▶ 覚えよう！

- ☐ 正規化は、更新時異状を排除するために、通常は第3正規形まで行う
- ☐ 更新時異状には、ダブル挿入時異状、ダブル更新時異状、ダブル削除時異状の三つがある

**勉強のコツ**

第1正規形の定義は正確に理解しておきましょう。第1正規形の場合は、属性が単一値になればいいので、表(関係)を分解する必要はありません。試験問題の演習で迷ったら、定義に立ち返って考えてみましょう。

**動画**

第1正規形について解説した動画を公開しています。  
<http://www.wakuwakuacademy.net/itcommon/3/>  
 リストから「第1正規形」を選択してください。

## 2-1-2 第1正規形

第1正規形とは、シンプルなドメイン(定義域)上で定義された関係です。第1正規形では、シンプルなドメインにするために、属性を単一値にし、巾集合や直積集合を排除します。また、候補キーや主キーも決定する必要があります。

### 第1正規形の定義

第1正規形の定義は、シンプルなドメイン上で定義された関係です。シンプルなドメインとは、**属性がすべて単一値をとること**です。具体的には、ドメインに直積集合や巾集合<sup>べき</sup>がある場合には、それを排除していきます。

例えば、次のような関係“伝票”を考えます。

関係“伝票”

伝票番号	顧客名(顧客番号)	商品番号	商品名	数量
1001	ねこ商事(11)	2001	チョコレート	5
		2002	キャラメル	10
1002	うさぎ開発(13)	2001	チョコレート	7
		2003	ドーナツ	20
		2004	シュークリーム	8
1003	くま工業(29)	2005	プロテイン	30
1004	くま工業(29)	2005	プロテイン	50

この関係は、第1正規形に正規化されていないので**非正規形**です。これを第1正規形にするために、ドメインをシンプルにしていきます。

まず、直積集合の排除です。直積集合(direct product)とは、集合の集まりで、各集合を組にしたものです。上の関係では、“顧客名(顧客番号)”に当たります。一つのドメイン(マス目)の中に(ねこ商事, 11)のように複数の値が入っており単一値ではないため、これを排除します。具体的には、顧客名と顧客番号を別の属性とし、次のようなかたちにします。

直積集合を排除した関係“伝票”

伝票番号	顧客番号	顧客名	商品番号	商品名	数量
1001	11	ねこ商事	2001	チョコレート	5
			2002	キャラメル	10
1002	13	うさぎ開発	2001	チョコレート	7
			2003	ドーナツ	20
			2004	シュークリーム	8
1003	29	くま工業	2005	プロテイン	30
1004	29	くま工業	2005	プロテイン	50

次に、<sup>べき</sup>巾集合を排除します。巾集合 (power set) とは、与えられた集合から、その部分集合を元として含む集合のことです。具体的には、伝票番号“1001”に対する商品番号 {2001, 2002} のように、一つの値に対して複数の値 (集合) が対応するような関係です。関係“伝票”の場合には、伝票番号などに対して、商品番号、商品名、数量は巾集合となっています。

これを排除するためには、商品番号などの一つの値に対して、それぞれ伝票番号などを対応させます。伝票番号などのデータは重複することになりますが、巾集合は排除できます。具体的には、次のようなかたちにします。

巾集合、直積集合を排除した関係“伝票”

伝票番号	顧客番号	顧客名	商品番号	商品名	数量
1001	11	ねこ商事	2001	チョコレート	5
1001	11	ねこ商事	2002	キャラメル	10
1002	13	うさぎ開発	2001	チョコレート	7
1002	13	うさぎ開発	2003	ドーナツ	20
1002	13	うさぎ開発	2004	シュークリーム	8
1003	29	くま工業	2005	プロテイン	30
1004	29	くま工業	2005	プロテイン	50

このように、第1正規形の関係は、シンプルな基盤の目のような表の形式となり、それぞれのドメインは単一値となります。



<sup>べき</sup>巾集合は、正確には冪集合と書きますが、略字で巾と書くのが一般的です。冪は冪乗または累乗を意味し、ある値同士を掛け合わせることを指します。具体的には、 $2^5$ のように上付き数字で書く数が冪乗です。



## 勉強のコツ

候補キーの定義はこれだけで、単純なものですが、感覚的には理解しづらいと思います。問題演習の際に分からなくなったら、しっかり頭に入るまで、このページに戻ってきて改めて確認しましょう。このページに付箋を貼っておいてもいいくらいです。

## 候補キー

候補キー (candidate key) の定義は、次のとおりです。

関係 (リレーション) のタプルを一意に識別できる属性または属性の組のうち極小のもの

この条件に当てはまる属性または属性の組を、候補キーといいます。

例えば、先ほどの関係“伝票”を基に候補キーを考えます。

第1正規形にした関係“伝票”

伝票番号	顧客番号	顧客名	商品番号	商品名	数量
1001	11	ねこ商事	2001	チョコレート	5
1001	11	ねこ商事	2002	キャラメル	10
1002	13	うさぎ開発	2001	チョコレート	7
1002	13	うさぎ開発	2003	ドーナツ	20
1002	13	うさぎ開発	2004	シュークリーム	8
1003	29	くま工業	2005	プロテイン	30
1004	29	くま工業	2005	プロテイン	50

さらに、次のような条件を加えます。

- ・顧客は複数回注文するので、伝票は複数ある
- ・顧客番号に対応する顧客名の重複はない
- ・商品番号に対応する商品名の重複はない

ここでまず、候補キーの条件のうち、「関係のタプルを一意に識別」を考えてみます。

上の関係では、{伝票番号, 顧客番号, 顧客名, 商品番号, 商品名, 数量} のすべての属性で全部値が同じタプル (行) はないので、すべての属性があれば、関係のタプルを一意に識別することは可能です。ちなみに、このように「関係のタプルを一意に識別」できるという条件のみを満たす属性または属性の組のことをスーパーキー (super key) といいます。

しかし、すべての属性がなくても、タプルを一意に特定することはできそうです。ここで、候補キーのもう一つの条件、「極小」について考えていきます。極小とは、「もう一つ属性が欠落すると、条件を満たさなくなる」ぎりぎりの属性または属性の組です。

関係“伝票”では、{伝票番号, 商品番号}の両方が同じ属性の組であるタプルはありません。さらにここから、伝票番号だけ、商品番号だけを取り出すと、両方に同じ値の属性が存在するので、これ以上削ることができません。したがって、{伝票番号, 商品番号}は候補キーの一つになります。

さらに、候補キーは一つとは限りません。この関係の場合、「商品番号に対する商品名の重複はない」という条件があるので、商品名でも、商品番号の代わりに商品を一意に識別することが可能です。つまり、{伝票番号, 商品名}でもタプルが一意に識別できるので、これも候補キーの一つになります。

したがって、関係“伝票”の候補キーは、{伝票番号, 商品番号}および{伝票番号, 商品名}の二つになります。



候補キーを識別するときの条件は、あくまで「タプル(行)が一意(1行)に決まるかどうかです。」「一番重要な属性」でもありませんし、「キーになりそうな～番号を全部挙げる」わけでもありません。

候補キーはあくまで数学の理論から導き出されるので、重要なものに目が行きがちな人間の感覚とは違っています。

なんとなく解くのではなく、一つ一つ理由を考えて候補キーを決定していきます。

それでは、次の問題を解いてみましょう。

## 問題

関係モデルの候補キーの説明のうち、適切なものはどれか。

- ア 関係Rの候補キーは関係Rの属性の中から選ばない。
- イ 候補キーの値はタプルごとに異なる。
- ウ 候補キーは主キーの中から選ぶ。
- エ 一つの関係に候補キーが複数あってはならない。

(平成28年春 データベーススペシャリスト試験 午前Ⅱ 問7)

## 解説

候補キーとはタプルを一意に識別するキーなので、タプルごとに候補キーの値は異なります。したがって、イが正解です。

- ア 候補キーは、関係Rの属性の中から選びます。
- ウ 主キーは、候補キーの中から一つ選ぶものです。
- エ 候補キーは、一つの関係に複数存在することもよくあります。

《解答》イ



候補キーの中から主キーを選ぶ方法は、NULLが含まれる候補キー以外では任意です。

ただ、通常は番号やコードなどが選ばれます。これは、データのサイズが小さく固定長であり、DBMSで扱いやすいからです。



主キー制約については、[3-1-5 データベースの制約]でも取り上げます。



主キーについて解説した動画を公開しています。

<http://www.wakuwakuacademy.net/itcommon/3/>リストから「主キー」を選択してください。



#### 過去問題をチェック

候補キーを答えさせる問題は、データベーススペシャリスト試験 午後 I 問 1 の定番中の定番で、毎回出題されています。

#### 【候補キーを答える問題】

- ・平成22年春 午後 I 問 1 設問 1 (3)
- ・平成23年特別 午後 I 問 1 設問 1 (3)、設問 2 (1)
- ・平成24年春 午後 I 問 1 設問 1 (2)、設問 2 (1)
- ・平成25年春 午後 I 問 1 設問 1 (1)、設問 2 (1)
- ・平成26年春 午後 I 問 1 設問 1 (1)
- ・平成27年春 午後 I 問 1 設問 1 (1)
- ・平成28年春 午後 I 問 1 設問 1 (1)

## 主キー

主キー (primary key) は、候補キーの中から**一つ**を選んだものです。主キーをどれにするかは、データベースの目的によって、データベースを設計する人が決めることができます。

ただし、主キーには候補キーの条件に加えて、もう一つの制約があります。それは、**主キーを構成する属性の値は空値 (NULL) ととらない**ことです。これを主キー制約といいます。

例えば、次のような関係を考えてみます。

関係「顧客」

顧客番号	顧客名	メールアドレス
11	ねこ商事	nyan@nekoneko.utau
13	うさぎ開発	pyon@usagi.drops
29	くま工業	—

ここで、顧客名が同じ顧客はおらず、メールアドレスの重複もない場合には、候補キーは次の三つになります。

{顧客番号}

{顧客名}

{メールアドレス}

顧客番号29のくま工業はメールアドレスをもっていないので、空値(—, NULLの意味)となっています。このような、空値をとる可能性がある候補キー {メールアドレス} は、主キーとして選ぶことができません。残った {顧客番号} と {顧客名} は、空値をとらないのであれば、どちらを選んでも理論的には問題ありません。通常は、番号の方がデータサイズが小さいので、{顧客番号} が選ばれます。

候補キーは、空値があってもかまいません。空値ではない値がタプルごとに一意であることが確認できれば、NULLが入っているタプルがあっても候補キーになります。また、主キーに選ばれなかった候補キーは**代理キー** (alternate key) となります。

この後、第2正規形、第3正規形への正規化は、**主キーではなく候補キーを正規化の定義で用います**。このとき、すべての候補キーが対象となるので、候補キーは一つだけでなくすべて挙

げておくことが大切です。

## ■ 関係スキーマの表記ルール

データベーススペシャリスト試験で用いられるデータベースの表記法として、関係スキーマがあります。関係スキーマの表記ルールでは、関係を、関係名とその右側のカッコでくくった属性名の並びで表します。具体的には、次のようなかたちです。

関係名 (属性名1, 属性名2, 属性名3, ..., 属性名n)

主キーを表す場合は、属性名1, 属性名2のように、主キーを構成する属性または属性の組に**実線の下線**を付けます。

外部キーを表す場合は、属性名3のように、外部キーを構成する属性または属性の組に**破線の下線**を付けます。ただし、主キーを構成する属性の組の一部が外部キーを構成する場合は、破線の下線を付けません。

関係スキーマのこの表記ルールは、「問題文中で共通に使用される表記ルール」として、午後Ⅰ、午後Ⅱの問題の冒頭に毎回書かれています。

例として、先ほどの関係“顧客”を、関係スキーマの表記ルールで表記してみます。

顧客 (顧客番号, 顧客名, メールアドレス)

候補キーの中から選んだ主キー“顧客番号”に対して、実線の下線を付けます。

関係スキーマを表記するときには、主キーや外部キーを記述する必要があることが多いので、主キーや外部キーを正確に選べるようにしておくことが重要です。



関連

外部キーについては、正規化後に出てくるキーなので「2-1-4 第3正規形」で詳しく取り扱います。



発展

主キー、外部キーが二つ以上の属性の集合で成り立っている場合を**複合キー**といいます。

複合キーは属性の数が多くなると扱いにくいので、連番など一つの属性で一意となるキーで代替することもあります。

このようなキーを**代替キー**といいます。

### ▶▶ 覚えよう！

- ☐ 第1正規形では、属性がすべて単一値をとる
- ☐ 候補キーは、関係のタプルを一意に特定できる属性または属性の組のうち極小のもの
- ☐ 主キーは、候補キーのうちの一つで、NULLを許さない





## 勉強のコツ

正規化するときには第3正規形まで行うことが多いので、第2正規形は中途半端に感じると思います。

しかし、途中段階の第2正規形を理解しておくことで確実に正規化を行うことができ、また、非正規化にも役立ちます。

午後1では「第2正規形になっていない」ことを答えさせる問題は定番なので、確実にできるようにしておきましょう。

## 2-1-3 第2正規形

第2正規形は、すべての非キー属性が各候補キーに完全関数従属している関係です。第2正規形にするには、各候補キーの一部に従属する部分関数従属性を排除していきます。

### 第1正規形の問題点

第1正規形のままの関係で第2正規形になっていない場合には、それが原因で更新時異状が起こります。

例として、前項で挙げた関係“伝票”の第1正規形を見えます。ここで、主キーは「伝票番号、商品番号」とします。

第1正規形にした関係“伝票”

伝票番号	顧客番号	顧客名	商品番号	商品名	数量
1001	11	ねこ商事	2001	チョコレート	5
1001	11	ねこ商事	2002	キャラメル	10
1002	13	うさぎ開発	2001	チョコレート	7
1002	13	うさぎ開発	2003	ドーナツ	20
1002	13	うさぎ開発	2004	シュークリーム	8
1003	29	くま工業	2005	プロテイン	30
1004	29	くま工業	2005	プロテイン	50

この表では、次の3種類の更新時異状が起こります。

#### ①タプル挿入時異状

伝票番号1005の伝票を、購入する商品(商品番号)が決まる前に、(1005, 15, かね道場, -, -, -)(-はNULL)と登録しようとする、主キーの商品番号がNULL不可のため、登録できません。

#### ②タプル更新(修正)時異状

伝票番号1001の顧客番号を15に、顧客名をかね道場に変更しようとする場合、伝票番号1001の列は2行あるため、両方一度に修正する必要があります。1行だけ更新すると、データに矛盾が生じてしまいます。



## 発展

ここで挙げたような関係“伝票”の場合は、実際には、伝票番号だけ登録して商品番号を登録しないということはないかもしれません。ただ、第2正規形でないために第1正規形で起こる更新時異状は普遍的なもので、「こういったところで矛盾が出る」ということを理解しておいてください。



## 動画

第2正規形について解説した動画を公開しています。  
<http://www.wakuwakuacademy.net/itcommon/3/>  
 リストから「第2正規形」を選択してください。

### ③タプル削除時異状

伝票番号1003の商品番号2005のタプルを削除すると、伝票番号1003のその他の情報も消えてしまいます。伝票内容に関する情報が保持できなくなります。

このような更新時異状は、第2正規形に正規化することで解消できます。

## ■ 第2正規形の定義

第2正規形の定義は、次のとおりです。

関係Rが第2正規形であるとは、次の二つの条件を満たすときをいう：

- (1) Rは第1正規形である
- (2) Rのすべての非キー属性はRの各候補キーに完全関数従属している

つまり、第1正規形であるという要件を満たし、候補キーをすべて洗い出しておくのが前提です。

その上で、どの候補キーにも当てはまらない属性である非キー属性について、各候補キーに完全関数従属しているかどうかを確認します。

## ■ 完全関数従属と部分関数従属

完全関数従属とは、関数従属性  $X \rightarrow Y$  において、Xのすべての真部分集合  $X'$  について、 $X' \rightarrow Y$  が成立しないことを指します。

例えば、先ほどの関係“伝票”の第1正規形を考えてみます。関係スキーマで表記すると、関係“伝票”は次のようになります。主キーには、{伝票番号, 商品番号} を選びます。

伝票 (伝票番号, 顧客番号, 顧客名, 商品番号, 商品名, 数量)

商品番号に対応する商品名の重複はないとすると、この関係の候補キーは、{伝票番号, 商品番号} と {伝票番号, 商品名} の二つとなります。



関連

更新時異状は、それぞれの正規形で、正規化されていないことで起こります。

ここで取り上げたのは、部分関数従属性があることによる更新時異状で、これを排除したものが第2正規形です。

なお、第3正規形になっていない場合は、推移的関数従属性があることで更新時異状が起こります。この更新時異状については、「2-1-4 第3正規形」で改めて取り上げます。



用語

真部分集合とは、部分集合のうちの全体集合を除いた集合です。

例えば、集合 {A, B, C} があった場合、部分集合としては、{A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C} と  $\emptyset$  (空集合) の8種類が考えられます。

ここから、全体集合である {A, B, C} を除いたものが、真部分集合です。

そうすると非キー属性は，“顧客番号”，“顧客名”，“数量”の三つになります。これらの非キー属性について，それぞれが完全関数従属しているかどうかを考えていきます。



#### 勉強のコツ

関数従属性を考えるときには，単純に，**一意に決まるかどうか**だけに着目しましょう。

データとして関係があるかどうか，意味があるかどうかは，正規化を行うときには関係ありません。

大事なのは数学的に一意に特定できるかどうかです。その属性または属性の組が決まったときに，他の属性または属性の組が一意に特定できれば，関数従属があるということになります。

関数従属性  $X \rightarrow Y$  において，候補キーの一つである {伝票番号，商品番号} を  $X$  とします。ここで， $Y$  に {数量} を選んで，

{伝票番号，商品番号}  $\rightarrow$  {数量}

という関数従属性を考えます。この関数従属性は，候補キーはすべての属性を一意に特定するので，問題なく成り立ちます。

ここで，{伝票番号，商品番号} の真部分集合  $X'$  は，{伝票番号} のみ，{商品番号} のみの二つです。この真部分集合  $X'$  のそれぞれについて，

{伝票番号}  $\rightarrow$  {数量}

{商品番号}  $\rightarrow$  {数量}

が成り立つかどうかを，第1正規形にした関係“伝票”のデータを基に考えていきます。

伝票番号1001に対応する数量は5，10の2種類，伝票番号1002に対応する数量は7，20，8の3種類なので，伝票番号が決まっても数量は一意に決まりません。

また，商品番号2001に対応する数量は5，7の2種類なので，商品番号が決まっても数量は一意に決まりません。

つまり，すべての真部分集合で関数従属性が成り立たないので，非キー属性“数量”は，候補キー {伝票番号，商品番号} に完全関数従属しているといえます。

同様に，非キー属性“顧客番号”，“顧客名”についても考えていきます。{伝票番号，商品番号} を  $X$ ，{顧客番号，顧客名} を  $Y$  とすると，

{伝票番号，商品番号}  $\rightarrow$  {顧客番号，顧客名}

となり，これは問題なく成り立ちます。ここで，真部分集合である {伝票番号} {商品番号} について，それぞれ，

{伝票番号}  $\rightarrow$  {顧客番号，顧客名}

{商品番号}  $\rightarrow$  {顧客番号，顧客名}

が成立するかどうかを考えます。

伝票番号1001に対して顧客番号は11，顧客名はねこ商事と，

一意に対応します。また、伝票番号1002に対しても、顧客番号は13、顧客名はうさぎ開発と、一意に対応します。伝票番号1003、1004も含め、すべての伝票番号について、伝票番号が決まると顧客番号と顧客名が一意に決まります。

ということは、候補キー {伝票番号, 商品番号} の一部である {伝票番号} に対して、

{伝票番号} → {顧客番号, 顧客名}

という関数従属性が成り立ちます。この関数従属性のことを**部分関数従属性**といいます。

つまり、部分関数従属性があると完全関数従属ではなく、第2正規形の条件が成り立たないことになります。そのため、第2正規形にするときには、この部分関数従属性を排除していきます。

## ■ 部分関数従属性の排除

第1正規形を第2正規形にするためには、部分関数従属性を排除します。

それには、候補キーに対して部分関数従属となっている関数従属性を別の関係として分けていきます。

先ほどの例では、{伝票番号} → {顧客番号, 顧客名} という部分関数従属性があるので、これを取り出して別の関係とします。具体的には、部分関数従属する候補キーの一部を**新たな関係の候補キー**（主キー）とし、**非キー属性をその関係に移動**させます。

第1正規形の関係“伝票”の関係スキーマは、次のようになっています。

伝票 (伝票番号, 顧客番号, 顧客名, 商品番号, 商品名, 数量)

ここから、部分関数従属性を排除して新しい関係を作ると、次の二つになります。

伝票明細 (伝票番号, 商品番号, 商品名, 数量)

伝票 (伝票番号, 顧客番号, 顧客名)



関連

商品番号と顧客番号、顧客名はまったく関係ないので、一意には特定できません。

実際、商品番号2001に対し、顧客番号、顧客名は、11のねこ商事、13のうさぎ開発の二つのデータが存在します。

そのため、{商品番号} → {顧客番号, 顧客名} の関数従属性はありません。



関連

ここでの関係における顧客番号と顧客名、商品番号と商品名の関係は、第2正規形のルールでは分解されません。

この後の正規化(第3正規形、ボイスコード正規形)でまた取り上げますので、ここではとりあえず、第2正規形の概念をつかんでください。

関係には適宜、名前を付ける必要があります。ここは、新しく作った関係の方が“伝票”という名前にふさわしいため“伝票”に、元の関係は明細なので“伝票明細”として名前の変更を行いました。データを含めて表形式で書くと、次のようになります。

第2正規形にした関係“伝票明細”

伝票番号	商品番号	商品名	数量
1001	2001	チョコレート	5
1001	2002	キャラメル	10
1002	2001	チョコレート	7
1002	2003	ドーナツ	20
1002	2004	シュークリーム	8
1003	2005	プロテイン	30
1004	2005	プロテイン	50

第2正規形にした関係“伝票”

伝票番号	顧客番号	顧客名
1001	11	ねこ商事
1002	13	うさぎ開発
1003	29	くま工業
1004	29	くま工業

それでは、次の問題を考えてみましょう。



発展

この問は、解答を考えるだけならすぐに「第1正規形である」と「第2正規形ではない」とが見抜けるので、簡単だと思います。しかし、候補キーの選び方など、正確に正規化を行うのは意外と難しいので、一度しっかり正規化しながら考えることをおすすめします。

## 問題

受注入力システムによって作成される次の表に関する説明のうち、適切なものはどれか。受注番号は受注ごとに新たに発行される番号であり、項番は1回の受注で商品コード別に連番で発行される番号である。

なお、単価は商品コードによって一意に定まる。

受注日	受注番号	得意先コード	項番	商品コード	数量	単価
2015-03-05	995867	0256	1	20121	20	20,000
2015-03-05	995867	0256	2	24005	10	15,000
2015-03-05	995867	0256	3	28007	5	5,000

ア 正規化は行われていない。

イ 第1正規形まで正規化されている。

- ウ 第2正規形まで正規化されている。  
エ 第3正規形まで正規化されている。

(平成27年春 データベーススペシャリスト試験 午前Ⅱ 問6)

## 解説

問題文の表は、巾集合も直積集合もなく、属性は単一値をとっているため、第1正規形ではあります。この表が第2正規形の条件に当てはまるかどうかを考えていきます。

受注番号に対して項番が付けられており、受注番号に加えて項番があれば、その他の属性は一意に特定できるので、{受注番号, 項番}は候補キーの一つになります。

また、項番は「1回の受注で複数の商品の注文があった場合に、商品別に連番で発行される番号」とあります。つまり、同じ受注の中で商品コードごとに連番が対応するので、受注番号に加えて商品コードがあれば、連番も含めて他の属性を一意に特定できます。したがって、{受注番号, 商品コード}も候補キーの一つになります。

候補キー {受注番号, 項番}, {受注番号, 商品コード} とすると、非キー属性は、“受注日”、“得意先コード”、“数量”、“単価”の四つです。これらの属性について、候補キーの真部分集合に関数従属している関係がないかを考えると、次の二つが出てきます。

{受注番号} → {受注日, 得意先コード}

{商品コード} → {単価}

この二つの部分関数従属性が存在するため、問題文の表は第2正規形ではありません。したがって第1正規形となるので、イが正解です。

## 《解答》イ



### 過去問題をチェック

第2正規形、特に部分関数従属性について表を埋めて正規形を答えさせる問題は、データベーススペシャリスト試験 午後Ⅰの定番です。

#### 【部分関数従属性について(穴埋め問題)】

- ・平成26年春 午後Ⅰ 問1 設問1 (1), (2)
- ・平成27年春 午後Ⅰ 問1 設問1 (1), (2)
- ・平成28年春 午後Ⅰ 問1 設問1 (1), (2)



### 発展

関係と候補キーを表記するときには、カッコを使い分ける必要があります。

関係を表現するときには、関係(属性1, 属性2)というように( )を使います。これに対し、候補キーの組を示すときには、{属性1, 属性2}といったかたちで{ }を使います。

このことは問題文の前や問題文中に表記法として記載されているので、それに倣って書くクセをつけておきましょう。

## ▶▶ 覚えよう！

- ☐ 第2正規形では、すべての非キー属性が各候補キーに完全関数従属している
- ☐ 第2正規形にするために、部分関数従属性を排除する

**勉強のコツ**

第3正規形は、なんとなくテーブルを分けた場合でもだいたいできていることが多い正規形です。ただし、自己流でやっている、微妙に間違えてしまうことも多々あります。正確に正規化できないと更新時異状の問題が起きますし、また、誤ったデータベース設計は不具合の影響が大きいのです。定義をしっかりと理解して、第3正規形への分解を正確にできるようにしましょう。

## 2-1-4 第3正規形

第3正規形は、すべての非キー属性が候補キーに推移的に関数従属しない関係です。第3正規形にするには、候補キー以外の属性または属性の組に従属する推移的関数従属性を排除していきます。

### 第2正規形の問題点

第1正規形だけでなく、第2正規形の関係でも更新時異状が起きます。

例えば、前項で取り上げた関係“伝票”の第2正規形は次のとおりです。

第2正規形にした関係“伝票”

伝票番号	顧客番号	顧客名
1001	11	ねこ商事
1002	13	うさぎ開発
1003	29	くま工業
1004	29	くま工業

この表では、次の3種類の更新時異状が起きます。

**動画**

第3正規形について解説した動画を公開しています。  
<http://www.wakuwakuacademy.net/itcommon/3/>  
 リストから「第3正規形」を選択してください。

#### ① タプル挿入時異状

顧客番号が15、顧客名がかめ道場のデータを、伝票が発生する前に、(ー, 15, かめ道場)(ーはNULL)と登録しようとする、主キーの伝票番号がNULL不可のため、登録できません。

#### ② タプル更新(修正)時異状

顧客番号29のくま工業が社名変更したので顧客名をくまAKに修正しようとした場合、顧客番号29の列は2行あるため、両方を一度に修正する必要があります。1行だけ更新すると、データに矛盾が生じてしまいます。

#### ③ タプル削除時異状

伝票番号1001のタプルを削除すると、顧客番号11のねこ商事の情報が消えてしまいます。そのため、顧客に関する情報が保

持できなくなります。

このような更新時異常は、第3正規形に正規化することで解消できます。

## 第3正規形の定義

第3正規形の定義は、次のとおりです。

関係Rが第3正規形であるとは、次の二つの条件を満たすときをいう：

- (1) Rは第2正規形である
- (2) Rのすべての非キー属性はRのいかなる候補キーにも推移的に関数従属しない

つまり、第2正規形であるという要件を満たすことは前提であり、そこから推移的関数従属性を排除していきます。

## 推移的関数従属性

**推移的関数従属性**とは、関係Rの異なる属性または属性の集合であるX, Y, Zについて、 $X \rightarrow Y$ ,  $Y \rightarrow X$ ,  $Y \rightarrow Z$ の三つの制約が成立している関数従属性です。図にすると次のようなかたちになります。



推移的関数従属性

この関係は、「Xが決まるとYが決まる」「Yが決まると、Xに  
関係なくZが決まる」「Yが決まってもXは決まらない」という関  
係です。

例えば、先ほどの第2正規形にした関係“伝票”の場合、Xを  
伝票番号、Yを顧客番号、Zを顧客名として考えると、

{伝票番号} → {顧客番号}

{顧客番号} → {顧客名}

の関数従属性は成立します。また、 $Y \rightarrow X$ である

{顧客番号} → {伝票番号}



推移的関数従属性は、 $Y \rightarrow X$ が成立しないことがポイントです。 $X \rightarrow Y$ ,  $Y \rightarrow X$ と戻ってしまうときには、二つが等価というだけなので、分けても意味がないからです。

例えば候補キーが二つあった場合には、互いの候補キーは、互いに対して一意に決定することができます。このような関係は推移的関数従属性ではないので、排除する必要はありません。



推移的関数従属性について、データベーススペシャリスト試験では次のような出題があります。

【推移的関数従属性の有無について】

・平成25年春 午後1 問1  
URI → リソースID と、リ  
ソースID → URI の両方が  
成り立つ関数従属性の問題  
が出題されています。  
この場合には、これらの  
属性間には推移的関数従  
属性は成立しません。