

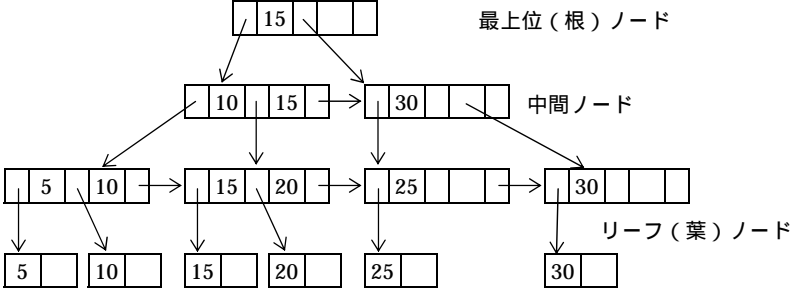
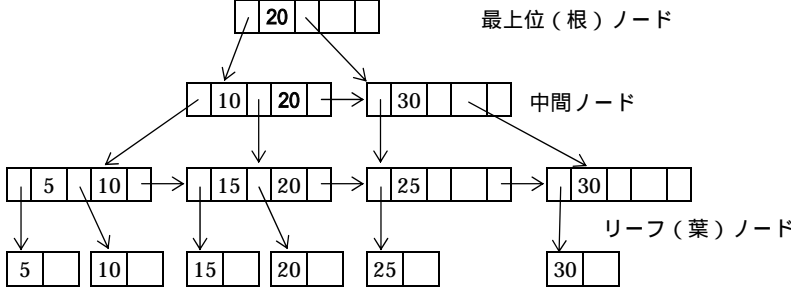
正 誤 表
-------

下記の部分に誤りがありましたので訂正させていただきます。

ご迷惑をおかけして大変申し訳ございません。

### 2009 データベーススペシャリスト「専門知識 + 午後問題」の重点対策（第1版 第2刷）

No.	訂正箇所	誤	正
1	P.43 図 2-2 下から 2 行目	属性が三つ : { 社員番号, 氏名, 電話番号 }, { 社員番号, 氏名, 所属部課コード }, { 氏名, 電話番号, 所属部課コード }	属性が三つ : { 社員番号, 氏名, 電話番号 }, { 社員番号, 氏名, 所属部課コード }, { 氏名, 電話番号, 所属部課コード }, <b>{ 社員番号, 電話番号, 所属部課コード }</b>
2	P.83 下から 2 行目	上位のキー ( 学生番号 )	上位のキー ( <b>注文番号</b> )
3	P.86 上から 17 行目	図表 1-5 の学生を分解しますと	図表 <b>3-11</b> の学生を分解しますと
4	P.117 上から 9 行目	図表 3-2 の商品表と	図表 <b>4-2</b> の商品表と
5	P.133 上から 8 行目	PRIMARY KEY(部門番号))	PRIMARY KEY( <b>商品番号</b> )
6	P.138 解説 下から 8 行目 ~ 3 行目	(イ)の更新権限は、「GRANT UPDATE (列名, ...) ON 表名 TO 権限受領者」と指定します。UPDATE (列名, ...) によって、ON 句の表の列単位に更新権限を定義でき、また (列名, ...) を省略した場合には、ON 句の表単位に更新権限を定義できます。したがって、(イ)が正解です。このほかの選択肢の説明は次のとおりです。 ア：一意性制約のある列には、NULL は入れることはできません。	<b>一意性制約のある列には、NULL を入れることはできません。したがって、(ア)が正解です。</b> このほかの選択肢の説明は次のとおりです。 <b>イ：更新権限に関することであり、スキーマ定義とは無関係です。</b>
7	P.139 解答	解答 イ	解答 <b>ア</b>
8	P.140 解説 上から 14 行目	FROM 商品	FROM <b>注文</b>

No.	訂正箇所	誤	正
9	P.161 解説 下から 14 行目～6 行目 下から 3 行目～1 行目	…… いずれも、トランザクションの実行によって、データの整合性が崩れることを防止するか、トランザクションの実行結果が正しくなるように保証することを目的としています。READ COMMITTED では、あるデータを更新したトランザクションが COMMIT を発行する前に、他のトランザクションが同じデータを READ することができません。必ず COMMIT の後に の READ が行われるので、 が先に終了することはありません。したがって、(イ)が誤りとなります。正解は(イ)です。 ア：二つの SQL 文は、列 (A, B, C) を同時にアクセスしているので、デッドロック発生の原因となります。READ COMMITTED は、データの整合性を維持する目的で使われ、デッドロックを防ぐ機能はありません。 ウ：READ COMMITTED では、必ず COMMIT の後に READ を行います。 エ：READ COMMITTED では、Non-repeatable read や Phantoms が発生します。したがって、記述のような現象が発生します。 解答 イ	…… いずれも、トランザクションの実行によって、データの整合性が崩れることを防止するか、トランザクションの実行結果が正しくなるように保証することを目的としています。二つの SQL 文は、一つの表の列(A, B, C)に同時にアクセスしています。 は READ COMMITTED ですから、SELECT ではページ単位で共有ロックを掛けますが、複数のページ(複数の資源)を同時にロックしようとはしません。したがって、デッドロックは発生しないことになり、(ア)が誤っています。 イ：READ COMMITTED では、SELECT 文では共有ロックを SELECT 文実行中だけ掛けます。 の実行中に を実行すると、 の SQL 文終了後に、まだ COMMIT 文が発行されない時点で、 の SQL 文のロックが掛かっていないので終了する場合があります。 ウ：READ COMMITTED では、必ず COMMIT の後に READ を行います。 エ： を連続して実行していても、COMMIT 文は直後にあるので、 が の終了した更新結果を読むのは当然です。 解答 ア
10	P.175 図表 5 - 7 (d)の B+木	 <p style="text-align: center;">(d) (5, データ), (10, データ), (15, データ), (25, データ), (30, データ)</p>	 <p style="text-align: center;">(d) (5, データ), (10, データ), (15, データ), (25, データ), (30, データ)</p>
11	P.235 [ 解答 ] 上から 9 行目	根拠：非キー属性は候補キー { 電話番号, 年月月 }	根拠：非キー属性は候補キー { 電話番号, 年月度 }

No.	訂正箇所	誤	正
12	P.251 上から 8 行目～12 行目	<p>図 2 の関数従属性図を見れば、成約番号 顧客番号 {氏名, 住所, 年収, 家族構成}, 顧客番号 成約番号 (顧客番号 成約番号という関数従属が成立しない) という推移的関数従属性があるのは明白です。{氏名, 住所, 年収, 家族構成} のうちの部分集合の属性でもよいでしょう。また, (1) で示した候補キーを用いて, {物件番号, 棟番号, 住居番号} 顧客番号 {氏名, 住所, 年収, 家族構成} \ 顧客番号 {物件番号, 棟番号, 住居番号} でもよいでしょう。</p>	<p>図 2 の関数従属性図を見れば、成約番号 顧客番号 {氏名, 住所, 年収, 家族構成}, 顧客番号 \ 成約番号 (顧客番号 成約番号という関数従属が成立しない) という推移的関数従属性があるのは明白です。{氏名, 住所, 年収, 家族構成} のうちの部分集合の属性でもよいでしょう。また, (1) で示した候補キーを用いて, {物件番号, 棟番号, 住居番号} 顧客番号 {氏名, 住所, 年収, 家族構成}, 顧客番号 \ {物件番号, 棟番号, 住居番号} でもよいでしょう。</p>
13	P.261 [ 解答 ] 上から 5 行目	注文 ( <u>注文番号</u> , 会員番号, 注文日, 注文受付日, <u>商品番号</u> , 注文数量, 取消しフラグ )	注文 ( <u>注文番号</u> , <del>会員番号</del> , 注文日, 注文受付日, <u>商品番号</u> , 注文数量, 取消しフラグ )

(株) アイテック 商品開発本部 教材制作グループ