Chapter９

データベース

# 1. データベース方式

## 1. データベース

学習のポイント

✅ 関係データベースを中心に用語を覚えよう！

複数のファイルを集め、更新などを容易にできるようにしたデータの集合体をデータベースといいます。データベースは、大量のデータを扱うのに適しており、顧客管理、在庫管理、販売管理などの多くの業務で利用されています。

ファイルを利用する問題点として次のようなものがあります。

|  |  |
| --- | --- |
| ファイルを利用する問題点 | 理由 |
| データが重複する | ファイルの内容はプログラムの中で定義されているため、異なるファイルに同じ内容のデータが複数発生することがある。 |
| 保守性・整合性が低い | 更新漏れなどが発生しやすくなり、ファイルの内容を常に正確に保てない。また、ファイル変更時はプログラム修正も伴う。 |
| 保守費用が増大する | ファイルの内容を変更するとプログラムの修正も必要になる。 |
| 安全性が低い | ファイルが分散しているため、セキュリティに問題が生じる。また、セキュリティ対策の機能も不十分である。 |

ファイルのさまざまな問題点を解消するためにデータベースが登場しました。データベースはファイルを整理統合・一元化し、より効率的、多目的に使用することができます。

売上ファイル

在庫ファイル

商品ファイル

統合化

在庫管理プログラム

データ

商品ファイル

在庫ファイル

販売管理プログラム

データ

商品ファイル

在庫ファイル

売上ファイル

プログラム

データベース化

### １）データベースの種類と特徴

代表的なデータベースの種類は次のとおりです。

#### ①関係データベース

関係データベース（Relational Database）は、現在最も広く用いられているデータベースで、データを２次元の表によって表現します。

#### ②構造型データベース

構造型データベースは、データを親子関係でもつデータベースで、親子間にどのような関係があるかで、階層型データベースと網型データベースの２つに分けられます。

**ⅰ）階層型データベース**

階層型データベース（Hierarchical Database）は、親は複数の子をもつことができるが、子は親を１つしかもてない構造のデータベースです。

**ⅱ）網型データベース**

網型データベース（Network Database）は、親は複数の子を、子も複数の親をもつことができる構造のデータベースです。

#### ③オブジェクト指向データベース

オブジェクト指向データベース（Object Oriented Database）は、オブジェクト指向設計に基づいたデータベースです。

|  |
| --- |
| 例題  関係データベースのデータ構造の説明として，適切なものはどれか。  ア　親レコードと子レコードをポインタで結合する。  イ　タグを用いてデータの構造と意味を表す。  ウ　データと手続を一体化（カプセル化）してもつ。  エ　データを２次元の表によって表現する。  ア　網型データベースに関する記述です。  イ　マークアップ言語に関する記述です。  ウ　オブジェクト指向データベースに関する記述です。  基本情報　平成25年度秋　問29　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-478

### ２）データベースのモデル

🏋プラスアルファ

データベースについて考える場合、最も問題となるのは、現実世界のデータをどのようにコンピュータ内部で表現するかという、データモデルの作成に関することです。データのモデル化については、1978年にANSI/X3/SPARCが３階層のデータモデルを提唱し、現在最も広く受け入れられています。

利用者の立場から定義

データの論理構造を定義

データの物理構造を定義

論理データモデル

（外部モデル）

概念データモデル

物理データモデル

（内部モデル）

３階層のデータモデル

**①概念データモデル**

概念データモデルは、現実世界全体のデータをコンピュータやプログラムは意識せずに、データ同士の相互関係などの論理的な構造を定義したものです。１つの概念データモデルが１つのデータベースに対応します。概念データモデルの図式的表現方法（データモデリング技法）に、E-RモデルやUMLがあります。

**②論理データモデル**

論理データモデル（外部モデル）は、利用者（プログラム）の立場から必要なデータをあげて、データベースを利用者（プログラム）ごとに定義したものです。論理データモデルには、関係モデル、階層モデル、ネットワークモデルがあります。

**ⅰ）関係モデル**

関係モデルは、関連するデータを２次元の表形式で表現します。関係モデルに従って集められたデータの集合を、関係データベース（RDBMS）と呼びます。

1

男

2

3

4

85001

85012

85025

85053

鈴木

青木

高橋

山田

男

女

男

75

85

81

90

関係モデルのデータ構造

**ⅱ）階層モデル**

階層モデルは、データを関連の強い幾つかのグループにまとめ、このグループを上下関係で表現します。上位の階層を親といい、下位の階層を子といいます。親は複数の子をもつことができますが、子は１つの親しかもつことができません。

階層モデルに従って集められたデータの集合を、階層型データベースと呼びます。

課長Ｄ

部長Ｂ

社長Ａ

課長Ｅ

部長Ｃ

課長Ｆ

社員Ｇ

社員Ｈ

階層モデルのデータ構造

**ⅲ）ネットワークモデル**

ネットワークモデルは、階層モデルと同じように、親子関係でデータ構造を表現しますが、階層モデルとは異なり、子が複数の親をもつことができます。

ネットワークモデルに従って集められたデータの集合を、網型データベースと呼びます。

顧客Ｄ

商品Ｂ

商店Ａ

顧客Ｅ

商品Ｃ

顧客Ｆ

ネットワークモデルのデータ構造

|  |
| --- |
| 例題　🏋プラスアルファ  データモデルにおいて，データの関係を木構造で表すものはどれか。  ア　E-Rモデル イ　階層モデル  ウ　関係モデル エ　ネットワークモデル  ア　E-Rモデルは、データ項目をエンティティ（実体）とエンティティ間の関連（リレーションシップ）で表すデータモデルです。  ウ　関係モデルは、データを行と列からなる２次元の表形式で表すデータモデルです。  エ　ネットワークモデルは、網型モデルともいい、階層構造と同じように、親子関係でデータ構造を表現します。階層構造とは異なり、子が複数の親をもつことができます。  基本情報　平成17年度秋　問58　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-479

🏋プラスアルファ

**③物理データモデル**

物理データモデル（内部モデル）は、概念データモデルで定義されたデータベースを、コンピュータ上でどのように実現するか（物理的な格納方法やアクセス方式など）を定義したものです。

３階層のデータモデルを採ることで、物理的なデータの独立性と論理的なデータの独立性が保証され、アプリケーションにデータ操作を組み込む必要がないため、開発・保守作業を大幅に軽減できます。また、データベースの設計変更や拡張を、アプリケーションを変更せずに実施できるなどの利点が生じます。

対象世界

E-Rモデル

階層モデル

ネットワークモデル

関係モデル

階層型データベース

ネットワークデータベース

関係データベース

データベース

概念データモデル

論理データモデル（外部モデル）

物理データモデル（内部モデル）

ANSI/SPARCの３階層データモデル

|  |
| --- |
| 例題　🏋プラスアルファ  概念データモデルの説明として，最も適切なものはどれか。  ア　階層モデル，ネットワークモデル，関係モデルがある。  イ　業務プロセスを抽象化して表現したものである。  ウ　集中型DBMSを導入するか，分散型DBMSを導入するかによって内容が変わる。  エ　対象世界の情報構造を抽象化して表現したものである。  ア　論理データモデルに関する記述です。  イ　業務プロセスではなく、業務で使うデータを対象とします。  ウ　物理データモデルに関する記述です。  データベース　平成26年度春Ⅱ　問1　[出題頻度：★☆☆]  解答－エ |

データモデルでは、現実のデータのモデル化を行いますが、このデータモデルの具体的な定義を**スキーマ**といいます。３層スキーマは、データベースの論理構造（利用者から見たデータ構造やデータの指示方法などを表したもの）、格納構造（データ項目や論理構造をDBMSとは独立して定義したもの）、物理構造（補助記憶装置上にどのように格納するかを表したもの）について記述したもので、概念モデルを記述した**概念スキーマ**、外部モデルを記述した**外部スキーマ**、内部モデルを記述した**内部スキーマ**に分類されます。

なお、CODASYL委員会では、概念スキーマをスキーマ、外部スキーマをサブスキーマ、内部スキーマを記憶スキーマと呼んでいます。

外部スキーマ

プログラム

外部スキーマ

プログラム

プログラム

外部スキーマ

概念スキーマ

内部スキーマ

データベース

データベース仕様を定義するスキーマ

|  |
| --- |
| 例題  RDBMSにおけるスキーマの説明として，適切なものはどれか。  ア　実表ではない，利用者の視点による仮想的な表である。  イ　データの性質，形式，他のデータとの関連などのデータ定義の集合である。  ウ　データの挿入，更新，削除，検索などのデータベース操作の総称である。  エ　データベースの一貫性を保持するための各種制約条件の総称である。  ア　ビューに関する記述です。  ウ　DML（Data Manipulation Language）に関する記述です。  エ　整合性制約に関する記述です。  基本情報　平成26年度秋　問26　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-480～483

### ３）関係データモデル

データを２次元の表形式で表現した関係データモデルでは、表は、行と列で構成され、各行には１件のデータを格納します。また、表を関係（リレーション）、行を**タプル**（組）、列を**属性**（項目、フィールド）、実際のデータを実現値、属性に入力できる値の集合を**定義域**（**ドメイン**）と呼びます。例えば、下記の例では、性別の定義域は「男、女」になります。なお、表内の列の並び順は入れ替えても処理に影響はありません。また、同じ表内の複数の列に、同じ名前を付けたり、名前をもたない列を定義することはできません。

列

行

列名

表名

従業員表

１つの行の１つの列のデータ要素 … １つの値

従業員名

小森泰治

江沢宏美

山田耕一

渡辺真美

加藤　弘

職位

66

61

60

58

55

生年月日

830814

980202

910511

750915

950707

給与

316500

247500

229500

241000

193500

所属

A00

B01

D01

E01

E01

性別

男

女

男

女

男

従業員番号

00510

02020

95130

92150

01060

表

データベース

|  |
| --- |
| 例題  関係データモデルにおいて，属性が取り得る値の集合を意味する用語はどれか。  ア　関係（リレーション） イ　実現値 ウ　タプル（組） エ　定義域  関係データモデルでは、属性が取り得る値の集合を「定義域」と呼びます。  基本情報　平成18年度秋　問59　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-484

🏋プラスアルファ

**●データベース管理システム**（**DBMS**：DataBase Management System）

データベース管理システムは、利用者とデータベースの中間に位置し、データベースを管理するソフトウェアです。データベース管理システムに求められる役割は、次のとおりです。

・データベースの定義

・データの効率的な利用

・データベースの共用

・データベースの障害に対する対策

・データベースの機密性に対する対策

・データベースにアクセスできる言語の提供

|  |
| --- |
| 例題　🏋プラスアルファ  データベース管理システムを利用する目的はどれか。  ア　OSがなくてもデータを利用可能にする。  イ　ディスク障害に備えたバックアップを不要にする。  ウ　ネットワークで送受信するデータを暗号化する。  エ　複数の利用者がデータの一貫性を確保しながら情報を共有する。  データベース管理システムは、利用者とデータベースの中間に位置し、データベースを管理するソフトウェアです。  ITパスポート　平成23年度秋　問53　[出題頻度：★☆☆]  解答－エ |

# 2. データベース設計

## 1. データベースの設計

学習のポイント

✅ 参照制約を中心に、例題と演習ドリルの問題をしっかり解いて解答力を身につけよう！

関係データベースは、記録するデータに重複や矛盾がないようにするため、あらかじめ一定のルールに基づいてデータの記録方法を最適化します。

|  |  |
| --- | --- |
| 1．E-R図の作成 | 設計対象のデータ同士の関連を表現する。 |
| 2．キーや制約条件の定義 | 主キー、外部キー、表に矛盾が生じないように制約を定める。 |
| 3．正規化 | 表を最適化する。 |

### １）データベースの概念設計

DBMSに依存しないデータの関連を表現する方法に、E-R図があります。

**E-R図**（E-Rモデル）は、企業活動で取り扱うデータの構造を、人や物、場所、金、サービスなどをさすエンティティ（実体）という概念と、エンティティ間の関連性である関連（リレーションシップ）という概念の２つでモデル化する技法です。

例１：担任と学生の関連をE-R図で表す。

①＜担任から見て＞

担任１に対して学生は多。

担任

１　　　　　　　　　　多

学生

②＜学生から見て＞

学生１に対して担任は１。

担任

１　　　　　　　　　　１

学生

担任と学生の関連は

【１対多】

担任

１　　　　　　　　　　多

学生

例２：先生と学生の関連をE-R図で表す。

①＜先生から見て＞

先生１に対して学生は多。

先生

１　　　　　　　　　　多

学生

②＜学生から見て＞

学生１に対して先生は多。

先生

多　　　　　　　　　　１

学生

先生と学生の関連は

【多対多】

先生

多　　　　　　　　　　多

学生

|  |
| --- |
| 例題  データベースの概念設計に用いられ，対象世界を，実体と実体間の関連という二つの概念で表現するデータモデルはどれか。  ア　E-Rモデル イ　階層モデル  ウ　関係モデル エ　ネットワークモデル  イ　階層モデルは、データを関連の強い幾つかのグループにまとめ、このグループを上下関係で表現します。上位の階層を親といい、下位の階層を子といいます。親は複数の子をもつことができますが、子は複数の親をもつことができません。  ウ　関係モデルは、データを行と列からなる２次元の表形式で表現します。  エ　ネットワークモデルは、網型モデルともいい、階層モデルと同じように、親子関係でデータ構造を表現します。階層モデルとは異なり、子が複数の親をもつことができます。  基本情報　平成20年度秋　問56　[出題頻度：★☆☆]  解答－ア |

### ２）データベースの論理設計

関係データベースにおいて、表（テーブル）の行を一意に識別するための属性の集まりを**候補キー**と呼びます。多くの場合、１つのテーブルには、候補キーが複数存在します。例えば、社員表では、社員番号、メールアドレス、社員名などが考えられますが、同姓同名の社員がいる場合、氏名は候補キーにはなりません。候補キーの中からキーを１つ選択し**主キー**（primary key）とします。主キーとして選択されなかったキーを代理キーといいます。また、複数の属性を組み合わせて主キーとしたものを複合キー、行を一意に識別するためだけに追加されたキーを代用キーと呼びます。なお、主キーはNULL値（空値）であってはいけません。

一方、表の候補キーを参照する属性の集まりを**外部キー**（foreign key）と呼びます。外部キーは主キーを兼ねることもありますが、外部キーとしてのみ使用する場合は、NULL値を入れることができます。

表を設計する場合、データの重複や矛盾が発生しないように注意する必要があり、次に示す制約条件を守らなければなりません。

#### ①一意性制約

一意性制約は、指定された項目、またはその組合せがとるべき値の集合中に、値の重複がないということです。

#### ②参照制約

参照制約は、外部キーをもつ表（参照元）から他の表（参照先）を参照する場合、その参照先に必ず参照元の外部キーの値が存在する必要があるということです。

#### ③非ナル制約

非ナル制約は、NOT NULLが指定した項目が、NULL（空）値であることを禁止することです。

#### ④存在制約

存在制約は、あるデータが存在するために、ほかのデータが存在する必要があるということです。例えば、網型データベースにおいて、親が存在しない場合、子は追加できないなどが該当します。

#### ⑤更新制約

更新制約は、ある項目を更新するとき、指定された条件以外の更新を禁止することです。

#### ⑥形式制約（ドメイン制約）

形式制約は、文字、数字、桁数など、データ型に関する制約です。

#### ⑦検査制約

検査制約は、データを入力するときに、指定された条件以外の入力を禁止することです。

#### ⑧主キー制約

主キー制約は、主キーに付けられる制約で、一意性制約と非ナル制約をあわせもつ制約です。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  　関係データベースの“注文”表の“顧客番号”は，“顧客”表の主キー“顧客番号”を参照する外部キーである。このとき，参照の整合性を損なうデータ操作はどれか。ここで，ア～エの記述におけるデータの並びは，それぞれの表の列の並びと同順とする。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 注文 | |  | 顧客 | | | 伝票番号 | 顧客番号 |  | 顧客番号 | 顧客名 | | 0001 | C005 |  | C005 | 福島 | | 0002 | K001 |  | D010 | 千葉 | | 0003 | C005 |  | K001 | 長崎 | | 0004 | D010 |  | L035 | 宮崎 |  |  |  |  |  |  | | --- | --- | --- | --- | --- | | ア | “顧客”表の行 | L035 | 宮崎 | を削除する。 | |  |  |  |  |  | | イ | “注文”表に行 | 0005 | D010 | を追加する。 | |  |  |  |  |  | | ウ | “注文”表に行 | 0006 | F020 | を追加する。 | |  |  |  |  |  | | エ | “注文”表の行 | 0002 | K001 | を削除する。 |   参照の整合性とは、参照制約（外部キー制約）のことで、追加、更新及び削除時に、関連する表の外部キーと対応する列との間で不一致が発生しないようにすることをいいます。  顧客番号F020は、“顧客”表に存在しないため、“注文”表に行　　　　　　　　　　　　を追加することはできません。  L035  宮崎  0006  F020  ア　“顧客”表の行　　　　　　　　　　　　　を削除しても、“注文”表に参照している行がないので問題はありません。  0005  D010  イ　“注文”表に行　　　　　　　　　　　　　を追加しても、顧客番号D010は“顧客”表に存在するので問題ありません。  0002  K001  エ　“注文”表の行　　　　　　　　　　　　　を削除しても、参照の整合性には関係なく、問題はありません。  基本情報　平成24年度秋　問31　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-485～493

## 2. データの正規化

学習のポイント

✅ 正規化の目的を覚えよう！

✅ 第１正規化では、繰返し項目を独立させることを覚えよう！

✅ 第2正規化では、部分関数従属を解消するために表を分割することを覚えよう！

✅ 第３正規化では、推移的関数従属を解消するために表を分割することを覚えよう！

✅ 正規化が完了していない表を、第3正規形にできるようになろう！

正規化の流れは次のとおりです。

|  |  |
| --- | --- |
| 1. 非正規形 | 繰返し項目をもつ表 |
| 1. 第１正規化 | 繰返し項目を独立させる（主キーを決定する） |
| 1. 第２正規化 | 主キーの一部分に従属する関係（部分関数従属）を排除する |
| 1. 第３正規化 | 主キー以外の項目に従属する関係（推移的関数従属）を排除する |

関係データベースを作る場合には、まず表の構成を定義する必要があります。ただし必要な項目を単純に集めただけでは、同じデータが重複した効率の悪いデータベースになってしまいます。正規化は、このような問題を防ぐことを目的に、ある一定の手順に従って表を分割し、簡潔にする作業をいいます。正規化を行うことでデータ間の関連が明確になり、表の更新も容易になります。一般的には第３正規化までが行われます。

|  |
| --- |
| 例題  データベースの正規化の目的のうち，適切なものはどれか。  ア　アクセスパスを固定して，データベースのアクセス速度を上げる。  イ　属性間の従属関係を単純化して，更新時の物理的なI/O回数を最小にする。  ウ　データの重複を排除して，重複更新を避け，矛盾の発生を防ぐ。  エ　テーブルの大きさを平準化して，データの参照速度を上げる。  データベースの正規化は、データの重複を防ぐために行われる操作で、ある一定の手順に従ってテーブル（表）を分割し、簡潔にすることです。正規化を行うことでデータ間の関連が明確になり、表の更新も容易になります。  ア　参照する項目により、アクセスパスが異なります。  イ　更新時の物理的なI/O回数が最小になるとはいえません。  エ　正規化により、分割されたテーブルごとに項目数が異なるため、大きさは一定にはなりません。  ソフトウェア開発　平成19年度春　問62　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-494

正規化は、第１正規化から始まり、一般的に次に示す手順で第３正規化までが行われます。

### １）第１正規化

第１正規化は、レコード中の繰返し項目を独立させてそれぞれ１行とする作業です。

非正規形から第１正規形への正規化の例を次に示します。

**非正規形**

〔受講表〕

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生  番号 | 氏名 | 講座  番号 | 講座名 | 講師  番号 | 講師名 | 受講  開始日 | 講座  番号 | 講座名 | 講師  番号 | 講師名 | 受講  開始日 |
| 0117 | 大原一郎 | S01 | ITパスポート | 10 | 田中 | 02.05.01 | J02 | 基本情報 | 25 | 岡田 | 02.11.01 |
| 0125 | 鈴木友子 | S01 | ITパスポート | 10 | 田中 | 01.11.01 | B03 | 簿記 | 07 | 山本 | 02.05.01 |
| 0133 | 高橋真理 | J02 | 基本情報 | 25 | 岡田 | 01.11.01 |

繰返し項目

|  |
| --- |
| ＜非正規形の問題事例＞  ・受講科目の数により、レコード長が異なっている  ・内容を修正する場合、複数レコードの修正が必要となり修正漏れが生じる可能性がある  ・２行目を削除すると、データベース上から簿記の講座に関する情報がなくなる  ・新しい講座を開講する場合、その講座を受講する人が存在しないと、データベース上に登録できない |

|  |
| --- |
| ＜第１正規化＞  ・繰り返し項目を独立させる  ・主キーを設定する |

**第１正規形**

〔受講表〕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生番号 | 氏名 | 講座番号 | 講座名 | 講師番号 | 講師名 | 受講開始日 |
| 0117 | 大原一郎 | S01 | ITパスポート | 10 | 田中 | 02.05.01 |
| 0117 | 大原一郎 | J02 | 基本情報 | 25 | 岡田 | 02.11.01 |
| 0125 | 鈴木友子 | S01 | ITパスポート | 10 | 田中 | 01.11.01 |
| 0125 | 鈴木友子 | B03 | 簿記 | 07 | 山本 | 02.05.01 |
| 0133 | 高橋真理 | J02 | 基本情報 | 25 | 岡田 | 01.11.01 |
| 0133 | 高橋真理 | J01 | 応用情報 | 25 | 岡田 | 02.11.02 |

※　　　は主キーを表す

|  |
| --- |
| ＜第１正規形の問題事例＞  ・内容を修正する場合、修正漏れが生じる可能性がある  ・４行目を削除すると、データベース上から簿記の講座に関する情報がなくなる  ・新しい講座を開講する場合、その講座を受講する人が存在しないと、データベース上に登録できない |

この表では、複数ある行から１行のみを特定するために「学生番号」と「講座番号」を指定する必要があります。したがって、「学生番号」と「講座番号」の組合せが主キーということになります。なお、ある項目が決まれば他の項目も一意に決まることを**関数従属**といいます。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  次の表において，属性Ａに対して関数従属性を満たしている属性はどれか。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Ａ | Ｂ | Ｃ | Ｄ | Ｅ | | 100 | 3100 | 10 | 東京都 | 2006年 5月 | | 100 | 1200 | 60 | 東京都 | 2006年11月 | | 100 | 1200 | 20 | 東京都 | 2007年 1月 | | 200 | 1100 | 10 | 大阪府 | 2006年 6月 | | 200 | 2200 | 20 | 大阪府 | 2006年10月 | | 300 | 3200 | 10 | 北海道 | 2006年 7月 | | 300 | 1200 | 30 | 北海道 | 2006年 9月 | | 400 | 4030 | 40 | 東京都 | 2006年 8月 | | 400 | 2200 | 40 | 東京都 | 2006年 9月 | | 400 | 1200 | 20 | 東京都 | 2006年12月 |   ア　Ｂ イ　Ｃ ウ　Ｄ エ　Ｅ  例えば、属性Ａが決まると属性Ｂが一意に定まるとき、「ＢはＡに関数従属する」といいます。本問において、属性Ａを決めたとき一意に定まる、すなわちＡに関数従属する属性はＤのみで、その他の属性は属性Ａを決めても一意には定まりません。  基本情報　平成18年度秋　問60　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-495～497

### ２）第２正規化

第２正規化では、主キー全体によって内容が決定する**完全関数従属**している項目と、主キーの一部分で内容が決定する**部分関数従属**している項目に分割して、各々別の表として独立させます。

主キー（「学生番号」と「講座番号」の両方）が決まって初めて確定する「受講開始日」は主キーに完全関数従属する項目です。それに対し、主キーの一部である「学生番号」が決まると確定する「氏名」は主キーに部分関数従属する項目です。同様に、「講座番号」が決まると確定する「講座名」及び「講師番号」、「講師名」も主キーに部分関数従属する項目です。

|  |
| --- |
| ＜第２正規化＞  ・部分関数従属（主キーの一部分に従属する関係）を独立した表に分割する |

〔受講表〕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生番号 | 氏名 | 講座番号 | 講座名 | 講師番号 | 講師名 | 受講開始日 |
| 0117 | 大原一郎 | S01 | ITパスポート | 10 | 田中 | 02.05.01 |
| 0117 | 大原一郎 | J02 | 基本情報 | 25 | 岡田 | 02.11.01 |
| 0125 | 鈴木友子 | S01 | ITパスポート | 10 | 田中 | 01.11.01 |
| 0125 | 鈴木友子 | B03 | 簿記 | 07 | 山本 | 02.05.01 |
| 0133 | 高橋真理 | J02 | 基本情報 | 25 | 岡田 | 01.11.01 |
| 0133 | 高橋真理 | J01 | 応用情報 | 25 | 岡田 | 02.11.02 |

※「学生番号」が分かれば、氏名が分かる

※「講座番号」が分かれば、「講座名」「講師番号」「講師名」が分かる

※「受講開始日」は「学生番号」と「講座番号」の両方が分からなければ確定できない（完全関数従属）ので、そのまま残す。

**第２正規形**

〔学生表〕 〔講座表〕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生番号 | 氏名 |  | 講座番号 | 講座名 | 講師番号 | 講師名 |
| 0117 | 大原一郎 |  | S01 | ITパスポート | 10 | 田中 |
| 0125 | 鈴木友子 |  | J02 | 基本情報 | 25 | 岡田 |
| 0133 | 高橋真理 |  | J01 | 応用情報 | 25 | 岡田 |
|  |  |  | B03 | 簿記 | 07 | 山本 |

〔受講表〕

〔受講表〕の「学生番号」は〔学生表〕の内容を参照する際の項目であり、〔受講表〕の「講座番号」は〔講座表〕の内容を参照する際の項目です。

したがって、「学生番号」や「講座番号」は外部キーということになります。

|  |  |  |
| --- | --- | --- |
| 学生番号 | 講座番号 | 受講開始日 |
| 0117 | S01 | 02.05.01 |
| 0117 | J02 | 02.11.01 |
| 0125 | S01 | 01.11.01 |
| 0125 | B03 | 02.05.01 |
| 0133 | J02 | 01.11.01 |
| 0133 | J01 | 02.11.02 |

外部キー

※　　　は主キーを表す

|  |
| --- |
| ＜第２正規形の問題事例＞  ・内容を修正する場合、修正漏れが生じる可能性がある  ・講座表の１行目や４行目を削除すると、データベース上から田中講師や山本講師の情報がなくなる  ・講座を担当していない講師の登録ができない |

第２正規形

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  次の表はどこまで正規化されたものか。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 従業員番号 | 氏名 | 入社年 | 職位 | 職位手当 | | 12345 | 情報　太郎 | 1981 | 部長 | 90,000 | | 12346 | 処理　次郎 | 1995 | 課長 | 50,000 | | 12347 | 技術　三郎 | 1997 | 課長 | 50,000 |   ア　第２正規形 イ　第３正規形 ウ　第４正規形 エ　非正規形  主キーは従業員番号のみであることから部分関数従属が存在する可能性はなく、少なくとも第２正規形以降となります。  このとき、「主キー以外の項目に従属（推移的関数従属）」する項目が含まれていれば第２正規形の状態です。  したがって、主キー（従業員番号）以外の項目に従属する項目があるかを探します。そのためには、項目間の関連性に着目することになります。  結果として、職位と職位手当の間に関連性があり、「職位が分かれば職位手当が分かる」（職位→職位手当）という推移的関数従属の関係が存在しているので、第２正規形ということが分かります。  応用情報　平成26年度春　問27　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-498

### ３）第３正規化

第３正規化では、主キー以外の項目に従属（**推移的関数従属**）する項目があれば、別の表として独立させます。

主キー以外の項目である「講師番号」が決まると確定する「講師名」は推移的関数従属する項目です。

|  |
| --- |
| ＜第３正規化＞  ・推移的関数従属（主キー以外の項目に従属する関係）を独立した表に分割する |

〔講座表〕

|  |  |  |  |
| --- | --- | --- | --- |
| 講座番号 | 講座名 | 講師番号 | 講師名 |
| S01 | ITパスポート | 10 | 田中 |
| J02 | 基本情報 | 25 | 岡田 |
| J01 | 応用情報 | 25 | 岡田 |
| B03 | 簿記 | 07 | 山本 |

※「講師番号」が分かれば、「講師名」が分かる

**第３正規形** 外部キー

〔学生表〕 〔講座表〕

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学生番号 | 氏名 |  | 講座番号 | 講座名 | 講師番号 |  |
| 0117 | 大原一郎 |  | S01 | ITパスポート | 10 |  |
| 0125 | 鈴木友子 |  | J02 | 基本情報 | 25 |  |
| 0133 | 高橋真理 |  | J01 | 応用情報 | 25 |  |
|  |  |  | B03 | 簿記 | 07 |  |

〔受講表〕 〔講師表〕

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生番号 | 講座番号 | 受講開始日 |  | 講師番号 | 講師名 |
| 0117 | S01 | 02.05.01 |  | 07 | 山本 |
| 0117 | J02 | 02.11.01 |  | 10 | 田中 |
| 0125 | S01 | 01.11.01 |  | 25 | 岡田 |
| 0125 | B03 | 02.05.01 |  |  |  |
| 0133 | J02 | 01.11.01 |  |  |  |
| 0133 | J01 | 02.11.02 |  |  |  |

〔講座表〕の「講師番号」は〔講師表〕の内容を参照する際の項目で  
外部キーとなります。

外部キー

※　　　は主キーを表す

第３正規形

|  |
| --- |
| 例題  項目a～fからなるレコードがある。このレコードの主キーは，項目aとbを組み合わせたものである。また，項目fは項目bによって特定できる。このレコードを第３正規形にしたものはどれか。  a  b  c  d  e  f  ア  a  b  c  d  e  b  f  a  b  c  d  e  b  f  c  d  e  a  b  f  b  f  a  c  d  e  b  c  d  e  b  f  イ  ウ  エ  問題のデータ構造より、主キーは項目aとbであり、項目c、d、e、fが関数従属しています。また、項目fは項目bに部分的に従属しているため、元の表から切り離して項目bと項目fで構成される表を新たに作成する必要があります。  このとき、a、bからｃ、ｄ、ｅへの従属関係は同一表内に残しておく必要がありますので、選択肢ア、ウ、エは不適切です。  なお、推移的関数従属が存在しないため、この段階で第３正規形となります。  基本情報　平成18年度春　問58　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-499～508

# 3. データの操作

## 1. データベースの操作

学習のポイント

✅ 関係演算（選択、射影、結合）を中心に用語を覚えよう！

関係データベースでは、行、列、表をそれぞれ１つの集合として扱い、データの演算を行い、その結果について抽出処理を行います。

### １）関係演算

関係演算には、選択（selection）、射影（projection）、結合（join）の３つがあります。

**選択**は、表から特定の行だけを取り出す操作です。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 〔受講表〕 |  |  | 選択 | 〔受講表〕 |  |  |
| 学生番号 | 講座番号 | 受講開始日 | 学生番号 | 講座番号 | 受講開始日 |
| 0117 | S01 | 02.05.01 | 0125 | S01 | 01.11.01 |
| 0117 | J02 | 02.11.01 | 0125 | B03 | 02.05.01 |
| 0125 | S01 | 01.11.01 | ※「受講表の学生番号が”0125”を選択する」 | | |
| 0125 | B03 | 02.05.01 |
| 0133 | J02 | 01.11.01 |
| 0133 | J01 | 02.11.02 |

**射影**は、表から特定の列だけを取り出す操作です。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 〔講座表〕 |  |  | 射影 | 〔講座表〕 |  |
| 講座番号 | 講座名 | 講師番号 | 講座番号 | 講座名 |
| S01 | ITパスポート | 10 | S01 | ITパスポート |
| J02 | 基本情報 | 25 | J02 | 基本情報 |
| J01 | 応用情報 | 25 | J01 | 応用情報 |
| B03 | 簿記 | 07 | B03 | 簿記 |

※「講座表の講座番号と講座名を射影する」

**結合**は、複数の表を関連のある列の値をもとに１つにまとめ、新たな表を作成する操作です。どの列をもとに結合するかを決定する条件を結合条件と呼びます。なお、同じ名前の列をもとに結合することを自然結合（natural join）と呼びます。また、列の値が等しい行のみが取り出される結合を内部結合（innner join）と呼び、値が等しくない行も残す結合を外部結合（outer join）と呼びます。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 〔講座表〕 |  |  |  | 〔講師表〕 |  |
| 講座番号 | 講座名 | 講師番号 |  | 講師番号 | 講師名 |
| S01 | ITパスポート | 10 |  | 07 | 山本 |
| J02 | 基本情報 | 25 | 10 | 田中 |
| J01 | 応用情報 | 25 | 25 | 岡田 |
| B03 | 簿記 | 07 |

結合

※「講座表と講師表を結合する」

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 講座番号 | 講座名 | 講師番号 | 講師番号 | 講師名 |
| S01 | ITパスポート | 10 | 10 | 田中 |
| J02 | 基本情報 | 25 | 25 | 岡田 |
| J01 | 応用情報 | 25 | 25 | 岡田 |
| B03 | 簿記 | 07 | 07 | 山本 |

|  |
| --- |
| 例題  関係データベースの操作の説明のうち，適切なものはどれか。  ア　結合は，二つ以上の表を連結して，一つの表を生成することをいう。  イ　射影は，表の中から条件に合致した行を取り出すことをいう。  ウ　選択は，表の中から特定の列を取り出すことをいう。  エ　挿入は，表に対して特定の列を挿入することをいう。  イ　射影は、表から特定の列だけを取り出すことをいいます。  ウ　選択は、表から特定の行だけを取り出すことをいいます。  エ　挿入は、表に対して特定の行を挿入することをいいます。  基本情報　平成23年度秋　問35　[出題頻度：★★★]  解答－ア |

別冊演習ドリル 》 1-509～511

### ２）集合演算

集合演算には、和（union）、積（intersection（共通））、差（difference）、直積（direct product）、商（divide）の５つがあります。

**和**は、２つの表がもつ全ての行を集め、重複を取り除く操作です。

**積**は、２つの表がもつ行のうち、項目に共通の値をもつ行を集める操作です。

**差**は、２つの表（表Ａと表Ｂ）について、表Ａから表Ｂと共通する項目の値をもつ行を取り除く操作です。

🏋プラスアルファ

**●直積と商**

**直積**は、２つの表（表Ｂと表Ｃ）について、行同士の全ての組合せで結合する操作です。

**商**は、２つの表（表Ａと表Ｄ）について、表Ａから表Ｄとまったく同じデータがある行を選択し、表Ｄの列を取り除く操作です。

単価

製品NO.

得意先NO.

受注数

製品NO.

15000

C345A

34201

690

C345A

30000

M2300

45098

860

M2300

30000

C345A

34201

690

M2300

15000

M2300

45098

860

C345A

直積

製品NO.

C345A

表Ｄ

単価

15000

製品NO.

C345A

30000

M2300

表Ｃ

得意先NO.

受注数

34201

690

商

製品NO.

C345A

得意先NO.

受注数

34201

690

S245C

45098

2156

S702Z

19804

3400

表Ａ

製品NO.

C345A

得意先NO.

受注数

34201

690

M2300

45098

860

表Ｂ

積

和

差

製品NO.

得意先NO.

受注数

S245C

45098

2156

S702Z

19804

3400

Ａ　－　Ｂ

製品NO.

C345A

得意先NO.

受注数

34201

690

Ａ　AND　Ｂ

製品NO.

C345A

得意先NO.

受注数

34201

690

S245C

45098

2156

S702Z

19804

3400

M2300

45098

860

Ａ　OR　Ｂ

集合演算

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  同じ属性から成る関係ＲとＳがある。ＲとＳの属性値の一部が一致する場合，関係演算Ｒ－（Ｒ－Ｓ）と同じ結果が得られるものはどれか。ここで，－は差集合，∩は共通集合，∪は和集合，×は直積，÷は商の演算を表す。  ア　Ｒ∩Ｓ イ　Ｒ∪Ｓ ウ　Ｒ×Ｓ エ　Ｒ÷Ｓ  関係演算とは、関係データベースの表に対する処理の一つです。例えば、表である関係ＲとＳが次のようであったとします。   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 関係Ｒ |  |  | 関係Ｓ |  | | 商品コード | 商品名 |  | 商品コード | 商品名 | | A100 | ボールペンＡ |  | A100 | ボールペンＡ | | A101 | ボールペンＢ |  | A102 | 消しゴムＡ | | A102 | 消しゴムＡ |  | A104 | 定規Ａ | | A103 | 消しゴムＢ |  | A105 | 定規Ｂ | | A104 | 定規Ａ |  |  |  |   この例に基づいて、Ｒ－(Ｒ－Ｓ)の演算を考えます。  差集合（－）は、左側の表から、右側の表と共通する行を差し引く処理です。  したがって、この例では、まず、Ｒ－Ｓを行うと、「ボールペンＢ」「消しゴムＢ」が残ります。これを更にＲから差し引きますので、Ｒ－(Ｒ－Ｓ)の結果は、「ボールペンＡ」「消しゴムＡ」「定規Ａ」となります。これは、結果的に関係Ｒと関係Ｓの共通する行が抜き出された共通集合になります。  したがって、同じ結果が得られるものは、関係ＲとＳの共通集合を求める「Ｒ∩Ｓ」になります。  基本情報　平成23年度秋　問32　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-512

## 2. データベース言語

学習のポイント

✅ 書式がたくさん出てくるので、しっかり習得しよう！

### １）データベース言語の種類

**SQL**（Structured Query Language）は、関係データベースを操作して、データの検索や変更、削除といった処理や、ユーザの立場から見た場合の論理的なデータ構造を記述するための言語です。データを定義する言語（DDL：Data Definition Language）とデータを操作する言語（DML：Data Manipulation Language）があります。

### ２）データ定義言語

磁気ディスク上に実際に作られる表を実表と呼びます。

#### ①実表の定義

実表を定義する場合、次のように記述します。

書式

CREATE TABLE 表名

（列名１ データ型,列名２ データ型,…）

（例）学生番号、氏名、国語、数学、クラスで構成される「成績」表を定義する（学生番号を主キーとする）。

CREATE TABLE 成績

（学生番号 CHAR(5) UNIQUE NOT NULL,

列名 ５文字の文字列 重複不可 空欄不可

氏名 NCHAR(5) NOT NULL,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 成績 |  |  |  |  |
| 学生番号 | 氏名 | 国語 | 数学 | クラス |

５文字の漢字列

国語 DECIMAL(3),

３桁の数値

数学 DECIMAL(3),

クラス CHAR(1) NOT NULL,

PRIMARY KEY（学生番号）,

主キーの設定

FOREIGN KEY（クラス） REFERENCES 担任（クラス））

「担任」表を参照するときの外部キーはクラス

実表の定義

#### ②列の追加

既存の表に新しく列を追加する場合、次のように記述します。

書式

ALTER TABLE 表名 ADD 列名 データ型

（例）「担任」表に「担当科目」を追加する。

ALTER TABLE 担任 ADD 担当科目 CHAR(4)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 担任 |  |  |  |  |  |
| クラス | 担任名 | ⇒ | クラス | 担任名 | 担当科目 |
| 1 | 木下恵子 | 1 | 木下恵子 | 英語 |
| 2 | 横山大介 | 2 | 横山大介 | 数学 |
| 3 | 佐藤直美 | 3 | 佐藤直美 | 商業 |
| 4 | 多和正敏 | 4 | 多和正敏 | 情報 |

　　　　　　　　　　　　　　　　　　　※担当科目の値は列追加後の入力例です。

列の追加

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  BUSHO表とSHAIN表があり，SHAIN表は次のSQL文で定義されている。  CREATE TABLE SHAIN  (S\_CODE CHAR(3) PRIMARY KEY,  S\_NAME NCHAR(3),  BU\_CODE CHAR(3),  S\_AGE DECIMAL(2),  FOREIGN KEY(BU\_CODE) REFERENCES BUSHO,  CHECK (S\_AGE BETWEEN 18 AND 60))  BUSHO表とSHAIN表に次のデータが格納されている状況で，SHAIN表に追加可能なデータはどれか。   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | BUSHO | |  | SHAIN | | | | | BU\_CODE | BU\_NAME |  | S\_CODE | S\_NAME | BU\_CODE | S\_AGE | | B01 | 人事部 |  | 111 | 山田 | B02 | 60 | | B02 | 総務部 |  | 122 | 川上 | B03 | 55 | | B03 | 経理部 |  | 233 | 田中 | B01 | 35 | |  |  |  | 259 | 岡本 | B02 | 34 |  |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | S\_CODE | S\_NAME | BU\_CODE | S\_AGE | | ア | 012 | 山田 | B03 | 60 | | イ | 111 | 山田 | B02 | 55 | | ウ | 320 | 山本 | B04 | 34 | | エ | 920 | 山下 | B03 | 17 |   表の定義は、CREATE文で表名や属性（データの型）、主キー、外部キーなどを指定します。問題文にあるCREATE文の構成は、次のとおりです。  CREATE TABLE SHAIN  表名  (S\_CODE CHAR(3) PRIMARY KEY,  列名　３文字の文字列　　　　主キー  S\_NAME NCHAR(3),  列名　　３文字の漢字列  BU\_CODE CHAR(3),  列名　　３文字の文字列  S\_AGE DECIMAL(2),  列名　　２桁の数値  FOREIGN KEY(BU\_CODE) REFERENCES BUSHO,  「BUSHO」表を参照するときの外部キーは「BU\_CODE」  CHECK (S\_AGE BETWEEN 18 AND 60))  年齢が18歳以上60歳以下という制約を付ける  ※列を定義する際に“CHECK”を利用して、独自の制約を付けることができる。  イ　SHAIN表のS\_CODEで「111」という値はすでに使用しています。主キーの重複は許されないので追加できません。  ウ　BUSHO表のBU\_CODEに「B04」という値は存在しません。BU\_CODEは外部キーとして使用しており、参照制約に反するので追加できません。  エ　CHECK句によりS\_AGEの値は18以上60以下という条件を付けているので、S\_AGEが「17」という値は追加できません。  ソフトウェア開発　平成17年度秋　問67　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-513,514

### ３）データ操作言語（その１）

#### ①データの挿入

データを実表に追加、挿入する場合、次のように記述します。

書式

INSERT INTO 表名(列名,…)

VALUES （値,…）

（例）「担任」表に、クラス「5」、担任名「渡辺圭一」というデータを挿入する。

INSERT INTO 担任(クラス,担任名)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 担任 |  |  | 担任 |  |
| クラス | 担任名 | ⇒ | クラス | 担任名 |
| 1 | 木下恵子 | 1 | 木下恵子 |
| 2 | 横山大介 | 2 | 横山大介 |
| 3 | 佐藤直美 | 3 | 佐藤直美 |
| 4 | 多和正敏 | 4 | 多和正敏 |
|  |  | 5 | 渡辺圭一 |

VALUES (‘5’,'渡辺圭一')

データの挿入

#### ②データの削除

データを実表から削除する場合、次のように記述します。

書式

DELETE FROM 表名 WHERE 条件

（例）「担任」表から、クラス番号が「5」のデータを削除する。

DELETE FROM 担任 WHERE クラス = 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 担任 |  |  | 担任 |  |
| クラス | 担任名 | ⇒ | クラス | 担任名 |
| 1 | 木下恵子 | 1 | 木下恵子 |
| 2 | 横山大介 | 2 | 横山大介 |
| 3 | 佐藤直美 | 3 | 佐藤直美 |
| 4 | 多和正敏 | 4 | 多和正敏 |
| 5 | 渡辺圭一 |  |  |

データの削除

#### ③データの更新

実表のデータを書き換える場合、次のように記述します。

書式

UPDATE 表名 SET 列名 = 変更データ

WHERE 条件

（例）「担任」表の、クラス番号が「3」の担任名を「小林直美」に変更する。

UPDATE 担任 SET 担任名 = '小林直美'

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 担任 |  |  | 担任 |  |
| クラス | 担任名 | ⇒ | クラス | 担任名 |
| 1 | 木下恵子 | 1 | 木下恵子 |
| 2 | 横山大介 | 2 | 横山大介 |
| 3 | 佐藤直美 | 3 | 小林直美 |
| 4 | 多和正敏 | 4 | 多和正敏 |

WHERE クラス = 3

データの更新

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  “商品”表に対してデータの更新処理が正しく実行できるUPDATE文はどれか。ここで，“商品”表は次のCREATE文で定義されている。  CREATE TABLE 商品  (商品番号 CHAR(4),商品名 CHAR(20),仕入先番号 CHAR(6),単価 INT,PRIMARY KEY(商品番号))   |  |  |  |  | | --- | --- | --- | --- | | 商品 |  |  |  | | 商品番号 | 商品名 | 仕入先番号 | 単価 | | S001 | A | XX0001 | 18000 | | S002 | A | YY0002 | 20000 | | S003 | B | YY0002 | 35000 | | S004 | C | ZZ0003 | 40000 | | S005 | C | XX0001 | 38000 |   ア　UPDATE 商品 SET 商品番号＝'S001' WHERE 商品番号 ＝'S002'  イ　UPDATE 商品 SET 商品番号＝'S006' WHERE 商品名 ＝ 'C'  ウ　UPDATE 商品 SET 商品番号＝NULL WHERE 商品番号 ＝'S002'  エ　UPDATE 商品 SET 商品名＝'D' WHERE 商品番号 ＝ 'S003'  表の更新を行う際には、表を作成した時点での定義情報に注目する必要があります。本問は、問題文中のCREATE文により、半角英数字４けた以内の商品番号、20けた以内の商品名、６けた以内の仕入先番号、整数値の単価、から構成されており、主キーは商品番号であることが分かります。  また、UPDATE文は、更新したい内容をSET句に記述し、対象となる行を指定するための条件をWHERE句に記述します。  ア　「商品番号が‘S002’である行の、商品番号を‘S001’に更新する。」という内容ですが、商品番号は主キーであり、既に商品表には商品番号‘S001’の行があるため、更新することができません（主キーは重複を許さない）。  イ　「商品名が‘C’である行の、商品番号を‘S006’に更新する。」という内容ですが、商品名が‘C’の行は２行あり、これらを更新すると、主キーである商品番号が重複してしまうため、更新することができません。  ウ　「商品番号が‘S002’である行の、商品番号をNULL（空白）に更新する。」という内容ですが、商品番号は主キーであり、NULLにすることはできないため、更新することができません。  基本情報　平成22年度秋　問31　[出題頻度：★☆☆]  解答－エ |

### ４）データ操作言語（その２）

射影、選択、結合などのデータ操作にはSELECT文を用います。SELECT文には単純な抽出から、複数の表を用いた複雑な抽出まで、さまざまな機能が備わっています。

ここからは、次の「成績」表および｢担任｣表を使い、データの操作方法を確認します。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成績 |  |  |  |  |  | 担任 |  |
| 学生番号 | 氏名 | 国語 | 数学 | クラス |  | クラス | 担任名 |
| 86001 | 青木光二 | 90 | 100 | 1 |  | 1 | 木下恵子 |
| 86002 | 太田治 | 75 | 70 | 2 |  | 2 | 横山大介 |
| 86003 | 小島浩 | 70 | 60 | 3 |  | 3 | 小林直美 |
| 87001 | 鈴木正明 | 100 | 50 | 2 |  | 4 | 多和正敏 |
| 87002 | 内藤俊彦 | 90 | 100 | 3 |  |  |  |
| 87003 | 山本弘一 | 95 | 80 | 1 |  |  |  |
| 87004 | 吉田和夫 | 85 | 90 | 1 |  |  |  |
| 87005 | 青山隆 | 50 | 60 | 2 |  |  |  |

#### ①単純な抽出

表の中から特定の列を抽出する（画面に表示する）場合、次のように記述します。

書式

SELECT 列名,列名,… FROM 表名

↑

取り出す列を記述する

（複数あれば、カンマ(,)で区切る）

↑

取り出す列が記録されている表の名前を記述する

（例）「成績」表から数学の点数を抽出する。

|  |
| --- |
| 数学 |
| 100 |
| 70 |
| 60 |
| 50 |
| 100 |
| 80 |
| 90 |
| 60 |

SELECT 数学 FROM 成績

単純な抽出１

「成績」表に同じ点数が複数行ある場合、同じ得点は重複して抽出されます。重複する行（同じ値をもつ行）を除いて抽出するには、SELECTの直後に「DISTINCT」を記述します。

（例）「成績」表から、数学の点数を重複なく抽出する。

|  |
| --- |
| 数学 |
| 100 |
| 70 |
| 60 |
| 50 |
| 80 |
| 90 |

SELECT DISTINCT 数学 FROM 成績

↑

抽出する際に重複する行を除く

単純な抽出２

なお、列名に「＊」を指定すると、表に含まれる全ての列を抽出することができます。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  関係Ｒ（ID，Ａ，Ｂ，Ｃ）のＡ，Ｃへの射影の結果とSQL文で求めた結果が同じになるように，ａに入れるべき字句はどれか。ここで，関係Ｒを表Ｔで実現し，表Ｔに各行を格納したものを次に示す。   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Ｔ | |  |  |  | 〔SQL文〕 | | ID | Ａ | | Ｂ | Ｃ | SELECT 　　ａ　　 A,C FROM T | | 001 | a1 | | b1 | c1 | | 002 | a1 | | b1 | c2 |  | | 003 | a1 | | b2 | c1 |  | | 004 | a2 | | b1 | c2 |  | | 005 | a2 | | b2 | c2 |  |   ア　ALL イ　DISTINCT ウ　ORDER BY エ　REFERENCES  射影は、実際には集合論に基づいています。集合論では、一つの集合内で重複を許しません。射影は表から特定の列だけを取り出す操作ですが、厳密には重複も許しません。したがって、「SELECT」の直後に「DISTINCT」と記述することで重複する行（同じ値を持つ行）を除いて抽出します。  ア　「SELECT」の直後に「ALL」と記述すると、条件に一致する全ての行を抽出します。「SELECT」の直後に「DISTINCT」等を記述しない場合は、「ALL」が省略されているとして処理されます。通常は省略します。  ウ　「SELECT」の直後に「ORDER BY」と記述することはできません。なお、「ORDER BY」は出力順を指定する字句です。  エ　「SELECT」の直後に「REFERENCES」と記述することはできません。なお、「REFERNCES」は参照する表を指定する字句です。  応用情報　平成29年度秋　問28　[出題頻度：★☆☆]  解答－イ |

#### ②抽出条件の指定

表の中から特定の条件を満たす行を抽出する場合、関係演算子を用いて条件を指定します。

|  |  |  |
| --- | --- | --- |
| 記号 | 意味 | |
| = | A = B | ：ＡとＢが等しい |
| <> | A <> B | ：ＡとＢが等しくない |
| > | A > B | ：ＡはＢより大きい（Ｂを含まない） |
| < | A < B | ：ＡはＢより小さい（Ｂを含まない） |
| >= | A >= B | ：ＡはＢ以上（Ｂを含む） |
| <= | A <= B | ：ＡはＢ以下（Ｂを含む） |

関係演算子

書式

SELECT 列名,列名,… FROM 表名 WHERE 条件

（例）「成績」表から、国語が90点以上の学生の学生番号を抽出する。

|  |
| --- |
| 学生番号 |
| 86001 |
| 87001 |
| 87002 |
| 87003 |

SELECT 学生番号 FROM 成績 WHERE 国語 >= 90

↑

条件を記述する場合には、関係演算子が用いられる

抽出条件の指定

表から特定の行を抽出するための条件は、論理演算子を用いることで一つだけでなく複数指定できます。論理演算子には次のものがあります。

|  |  |  |
| --- | --- | --- |
| 論理演算子 | 意味 | |
| OR | Ａ　OR　Ｂ | ：ＡまたはＢ |
| AND | Ａ AND Ｂ | ：ＡかつＢ |
| NOT | NOT A | ：Ａでない |

論理演算子

書式

SELECT 列名,… FROM 表名

WHERE 条件 論理演算子 条件

↑

複数の条件を指定する場合、各条件を論理演算子で結ぶ

（例１）「成績」表から、国語が70点以上かつ、数学が70点以上の学生の学生番号を抽出する。

|  |
| --- |
| 学生番号 |
| 86001 |
| 86002 |
| 87002 |
| 87003 |
| 87004 |

SELECT 学生番号 FROM 成績

WHERE 国語 >= 70 AND 数学 >= 70

（例２）「成績」表から、国語が70点以上または、数学が70点以上の学生の学生番号を抽出する。

|  |
| --- |
| 学生番号 |
| 86001 |
| 86002 |
| 86003 |
| 87001 |
| 87002 |
| 87003 |
| 87004 |

SELECT 学生番号 FROM 成績

WHERE 国語 >= 70 OR 数学 >= 70

複数条件の指定

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  次のSQL文によって表（学生一覧）から抽出されないデータはどれか。   |  |  |  |  | | --- | --- | --- | --- | | 表　学生一覧 | | |  | | 氏名 | 専攻 | 年齢 | SELECT 氏名 FROM 学生一覧  WHERE 専攻 = '物理' OR 年齢 < 20 | | 佐藤恒一 | 物理 | 22 | | 山田健次 | 化学 | 20 | | 鈴木有三 | 生物 | 18 | | 田中真司 | 物理 | 19 | | 斉藤五郎 | 数学 | 19 |   ア　佐藤恒一 イ　佐藤恒一，田中真司  ウ　鈴木有三，田中真司，斉藤五郎 エ　山田健次  特定の条件を満たす行の抽出は、SELECT文で列名、表名および抽出条件を指定します。問題文にあるSELECT文の構成は、次のとおりです。  SELECT 氏名 FROM 学生一覧  列名（氏名） 表名（学生一覧）  WHERE 専攻 = '物理' OR 年齢 < 20  抽出条件（「専攻が物理」または「年齢が20未満」）  したがって、「専攻が物理」または「年齢が20歳未満」の学生の氏名（佐藤恒一、田中真司、鈴木有三、斉藤五郎）が抽出されるため、抽出されないデータは山田健次となります。  初級システムアドミニストレータ　平成9年度秋　問47　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-515～517

#### ③抽出条件の値の範囲指定

抽出条件として値の範囲を指定でき、次のように記述します。

書式

SELECT 列名,… FROM 表名

WHERE 列名 BETWEEN 値 AND 値

（例）「成績」表から、国語が70点から80点までの学生の氏名を抽出する。

|  |
| --- |
| 氏名 |
| 太田治 |
| 小島浩 |

SELECT 氏名 FROM 成績

WHERE 国語 BETWEEN 70 AND 80

抽出条件の値の範囲指定

#### ④抽出条件の値の指定

抽出条件として値を指定でき、次のように記述します。値は「,」で区切ることで複数指定することができます。

書式

SELECT 列名,… FROM 表名

WHERE 列名 IN(値,値,…)

（例）「成績」表から、国語が70点または、90点の学生の氏名を抽出する。

|  |
| --- |
| 氏名 |
| 青木光二 |
| 小島浩 |
| 内藤俊彦 |

SELECT 氏名 FROM 成績

WHERE 国語 IN (70,90)

抽出条件の値の指定

#### ⑤あいまい検索の利用

部分的に一致するデータや、一定のパターンに合致するデータを抽出する場合、次のように記述します。

書式

SELECT 列名,… FROM 表名

WHERE 列名 LIKE パターン

↑

文字列のパターンを記述する

「％」は０文字以上の任意の文字列

「\_」は任意の１文字

（例１）「成績」表から、学生番号が“３”で終わる学生の学生番号と氏名を抽出する。

SELECT 学生番号,氏名 FROM 成績

|  |  |
| --- | --- |
| 学生番号 | 氏名 |
| 86003 | 小島浩 |
| 87003 | 山本弘一 |

WHERE 学生番号 LIKE '％３'

（例２）「成績」表から、氏名が“青”で始まる４文字の学生の学生番号と氏名を抽出する。

SELECT 学生番号,氏名 FROM 成績

|  |  |
| --- | --- |
| 学生番号 | 氏名 |
| 86001 | 青木光二 |

WHERE 氏名 LIKE '青\_ \_ \_'

あいまい検索

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  氏名に“三”の文字をもつ社員を“社員”表から検索して，次のような“社員リスト”表を作成するSQL文中の　a　に入れるべき適切な字句はどれか。  社員リスト   |  |  |  |  | | --- | --- | --- | --- | | 社員番号 | 氏名 | 課コード | 内線電話 | | S02 | 田中　三郎 | K001 | 1001 | | S05 | 佐藤　幸三 | K003 | 1003 | | S15 | 森　三恵子 | K022 | 3022 | | S20 | 鈴木　俊三郎 | K105 | 1105 |   SELECT 社員番号, 氏名, 課コード, 内線電話 FROM 社員 WHERE 　　a  ア　氏名 = '三' イ　氏名 = '%三%'  ウ　氏名 LIKE '%三%' エ　氏名 NOT LIKE '%三%'  WHERE句のLIKE述語は、  列名　LIKE　'××××'　→　「××××のような文字列」  という意味を持ちます。  検索された社員リストを見ると、氏名の列において、氏名の途中及び最後に“三”という文字があります。'％'は任意の文字列を表す記号であり、“三”の前後の文字の有無にかかわらず、検索対象とするためには、“三”の前後に'％'を記述する必要があります。  基本情報　平成14年度春　問68　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-518

#### ⑥NULL値の列の抽出

列値が未入力（NULL値）の行を抽出する場合、次のように記述します。

書式

SELECT 列名,… FROM 表名

WHERE 列名 IS NULL

（例）「成績」表から、数学が未受験の学生の学生番号を抽出する。

SELECT 学生番号FROM 成績

WHERE 数学 IS NULL

|  |
| --- |
| 学生番号 |
| 86003 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 成績 |  |  |  |  |
| 学生番号 | 氏名 | 国語 | 数学 | クラス |
| 86001 | 青木光二 | 90 | 100 | 1 |
| 86002 | 太田治 | 75 | 70 | 2 |
| 86003 | 小島浩 | 70 | NULL | 3 |

NULL値の列の抽出

なお、「列名　IS NOT NULL」と記述することにより、指定の列に値が入力されている行だけを抽出することができます。

#### ⑦集約関数の利用

表に記録されている列の合計や平均を抽出する場合、次のように記述します。

書式

SELECT 集約関数(列名) FROM 表名

↑

合計や平均を求める関数を集約関数と呼び、次の種類がある。

|  |  |
| --- | --- |
| 関数名 | 意味 |
| SUM | ある列の合計値を求める |
| AVG | ある列の平均値を求める |
| MAX | ある列の最大値を求める |
| MIN | ある列の最小値を求める |
| COUNT | 行の数を求める |

集約関数

（例１）「成績」表から、全学生の数学の平均点を抽出する。

|  |
| --- |
| AVG(数学) |
| 76.25 |

SELECT AVG(数学) FROM 成績

集約関数１

（例２）「成績」表から、数学が100点の学生の人数を抽出する。

|  |
| --- |
| COUNT(\*) |
| 2 |

SELECT COUNT(\*) FROM 成績

WHERE 数学 = 100

集約関数２

抽出した表の列に対して、「列名　AS　新たな列名」と記述することで新たに列名を付けることもできます。

（例３）「成績」表から、国語の最高点を「国語最高点」という列名で抽出する。

|  |
| --- |
| 国語最高点 |
| 100 |

SELECT MAX(国語) AS 国語最高点 FROM 成績

集約関数３

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  “出庫記録”表に対するSQL文のうち，最も大きな値が得られるものはどれか。   |  |  |  | | --- | --- | --- | | 出庫記録 |  |  | | 商品番号 | 日付 | 数量 | | NP200 | 2015-10-10 | ３ | | FP233 | 2015-10-10 | ２ | | NP200 | 2015-10-11 | １ | | FP233 | 2015-10-11 | ２ |   ア　SELECT AVG（数量） FROM 出庫記録 WHERE 商品番号＝'NP200'  イ　SELECT COUNT（\*） FROM 出庫記録  ウ　SELECT MAX（数量） FROM 出庫記録  エ　SELECT SUM（数量） FROM 出庫記録 WHERE 日付＝'2015-10-11'  各選択肢のSQL文の抽出結果は次のとおりです。  ア　「出庫記録」表から商品番号が‘NP200’の平均数量を抽出します。  （３＋１）÷２＝２  イ　「出庫記録」表の行数を抽出します。  ４  ウ　「出庫記録」表から最大数量を抽出します。  ３  エ　「出庫記録」表から日付が‘2015-10-11’の合計数量を抽出します。  １＋２＝３  基本情報　平成27年度秋　問28　[出題頻度：★☆☆]  解答－イ |

#### ⑧グループごとの抽出

同一の値をもつデータを１つのグループとしてまとめ、グループごとに抽出する場合、次のように記述します。

書式

SELECT 列名または集約関数（列名），…

FROM 表名 GROUP BY 列名

↑

グループ化する列を記述する

※GROUP BYには、SELECTで指定した列のうち集約関数以外の全てを記述する

（例１）「成績」表をもとに、クラスごとの国語の平均点を抽出する。

SELECT クラス,AVG(国語) FROM 成績

|  |  |
| --- | --- |
| クラス | AVG（国語） |
| 1 | 90 |
| 2 | 75 |
| 3 | 80 |

GROUP BY クラス

↑

GROUP BYには、SELECTで指定した「クラス」と「AVG(国語)」のうち集約関数である「AVG(国語)」を除く全てを記述するため、「クラス」を記述する

となる

（例２）「成績」表をもとに、学生番号が87001以上の学生の、クラスごとの国語の平均点を抽出する。

|  |  |
| --- | --- |
| クラス | AVG（国語） |
| 1 | 90 |
| 2 | 75 |
| 3 | 90 |

SELECT クラス,AVG(国語) FROM 成績

WHERE 学生番号 >= '87001'

GROUP BY クラス

グループごとの抽出１

「GROUP BY」句でグループ化した列に続けて、「HAVING」句で更に条件を指定し抽出できます。

（例３）「成績」表から、クラスごとの国語の平均点が80点以上のクラスとその平均点を抽出する。

|  |  |
| --- | --- |
| クラス | AVG（国語） |
| 1 | 90 |
| 3 | 80 |

SELECT クラス,AVG(国語) FROM 成績

GROUP BY クラス

HAVING AVG(国語) >= 80

グループごとの抽出２

#### ⑨出力順の指定

データを抽出する場合に並べ替えの指定ができ、次のように記述します。

書式

SELECT 列名または集約関数(列名),… FROM 表名

ORDER BY 第１キーとする列名　並び順，

第２キーとする列名　並び順，…

↑

並び順は、昇順（小さい順）ならば「ASC」、

降順（大きい順）ならば「DESC」と指定する

ただし、省略すると昇順になる

（例1）「成績」表から、クラスと国語の点数をクラスの小さい順に、更に同じクラス内では点数の大きい順に抽出する。

|  |  |
| --- | --- |
| クラス | 国語 |
| １ | 95 |
| １ | 90 |
| １ | 85 |
| ２ | 100 |
| ２ | 75 |
| ２ | 50 |
| ３ | 90 |
| ３ | 70 |

SELECT クラス,国語 FROM 成績

ORDER BY クラス ASC,国語 DESC

↑

省略可能

出力順の指定１

なお、「ORDER BY」句の後ろに並び替える列名を記述しましたが、列名の代わりに「SELECT」句の後ろに記述した列名を、左から順に１、２、３…として、その数字を記述することもできます。

書式

SELECT 列名１,列名２ FROM 表名

↑

１番目

↑

２番目

ORDER BY 1 ASC, 2 DESC

↑

列名１

↑

列名２

（例２）「成績」表から、クラスごとの数学の平均点を点数の大きい順に抽出する。

|  |  |
| --- | --- |
| クラス | AVG（数学） |
| １ | 90 |
| ３ | 80 |
| ２ | 60 |

SELECT クラス,AVG(数学) FROM 成績

GROUP BY クラス

ORDER BY 2 DESC

出力順の指定２

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  表Ａから実行結果Ｂを得るためのSQL文はどれか。   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Ａ |  |  |  |  | 実行結果Ｂ |  |  | | 社員コード | 名前 | 部署コード | 給料 |  | 部署コード | 社員コード | 名前 | | 10010 | 伊藤幸子 | 101 | 200,000 |  | 101 | 10010 | 伊藤幸子 | | 10020 | 斉藤栄一 | 201 | 300,000 |  | 101 | 10030 | 鈴木裕一 | | 10030 | 鈴木裕一 | 101 | 250,000 |  | 102 | 10040 | 本田一弘 | | 10040 | 本田一弘 | 102 | 350,000 |  | 102 | 10050 | 山田五郎 | | 10050 | 山田五郎 | 102 | 300,000 |  | 201 | 10020 | 斉藤栄一 | | 10060 | 若山まり | 201 | 250,000 |  | 201 | 10060 | 若山まり |   ア　SELECT 部署コード,社員コード,名前 FROM Ａ GROUP BY 社員コード  イ　SELECT 部署コード,社員コード,名前 FROM Ａ GROUP BY 部署コード  ウ　SELECT 部署コード,社員コード,名前 FROM Ａ ORDER BY 社員コード,部署コード  エ　SELECT 部署コード,社員コード,名前 FROM Ａ ORDER BY 部署コード,社員コード  GROUP BY句はグループ化の際に用いられますが、GROUP BY句で選択した項目以外をSELECT文で指定することはできません。従って選択肢アとイは文法上誤りとなります。  また、実行結果Bは第１キー「部署コード」、第２キー「社員コード」の昇順に並んでいます。従って、ORDER BY句で「部署コード　社員コード」を指定することになります。  応用情報　平成22年度春　問33　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-519～521

#### ⑩表の結合（その１）

複数の表を結合してデータを抽出する場合、次のように記述します。

書式

←　結合する表をカンマで区切って全て並べる

SELECT 列名,… FROM 表名,表名,…

WHERE 表を結合する条件式

↑

関連付ける列を記述する

（例1）「成績」表と「担任」表から、学生の氏名とそれぞれの担任名を抽出する。

SELECT 氏名,担任名 FROM 成績,担任

WHERE 成績.クラス = 担任.クラス

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成績 |  |  |  |  |  | 担任 |  |
| 学生番号 | 氏名 | 国語 | 数学 | クラス |  | クラス | 担任名 |
| 86001 | 青木光二 | 90 | 100 | 1 |  | 1 | 木下恵子 |
| 86002 | 太田治 | 75 | 70 | 2 |  | 2 | 横山大介 |
| 86003 | 小島浩 | 70 | 60 | 3 |  | 3 | 小林直美 |
| 87001 | 鈴木正明 | 100 | 50 | 2 |  | 4 | 多和正敏 |
| 87002 | 内藤俊彦 | 90 | 100 | 3 |  |  |  |
| ： | ： | ： | ： | ： |  |  |  |

|  |  |
| --- | --- |
| 氏名 | 担任名 |
| 青木光二 | 木下恵子 |
| 太田治 | 横山大介 |
| 小島浩 | 小林直美 |
| 鈴木正明 | 横山大介 |
| 内藤俊彦 | 小林直美 |
| 山本弘一 | 木下恵子 |
| 吉田和夫 | 木下恵子 |
| 青山隆 | 横山大介 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 成績 |  |  |  |  | 担任 |  |
| 学生番号 | 氏名 | 国語 | 数学 | クラス | クラス | 担任名 |
| 86001 | 青木光二 | 90 | 100 | 1 | 1 | 木下恵子 |
| 86002 | 太田治 | 75 | 70 | 2 | 2 | 横山大介 |
| 86003 | 小島浩 | 70 | 60 | 3 | 3 | 小林直美 |
| 87001 | 鈴木正明 | 100 | 50 | 2 | 2 | 横山大介 |
| 87002 | 内藤俊彦 | 90 | 100 | 3 | 3 | 小林直美 |
| ： | ： | ： | ： | ： | ： | ： |

表の結合（その１）１

「FROM」句で表名のあとに別名を指定することで、条件式内で別名を利用できます。

（例２）「成績」表と「担任」表から、学生番号が87001以上の学生の氏名とそれぞれの担任名を抽出する。

|  |  |
| --- | --- |
| 氏名 | 担任名 |
| 鈴木正明 | 横山大介 |
| 内藤俊彦 | 小林直美 |
| 山本弘一 | 木下恵子 |
| 吉田和夫 | 木下恵子 |
| 横山隆 | 横山大介 |

SELECT 氏名,担任名 FROM 成績 X,担任 Y

↑

「担任」表の別名

↑

「成績」表の別名

WHERE X.クラス = Y.クラス

↑

「担任」表の別名

↑

「成績」表の別名

AND 学生番号 >= '87001'

表の結合（その１）２

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  “注文”表と“製品”表に対して，次のSQL文を実行したときに得られる結果はどれか。  SELECT 製品名,数量 FROM 注文,製品  WHERE 注文.製品コード ＝ 製品.製品コード   |  |  |  | | --- | --- | --- | | 注文 | | | | 日付 | 製品コード | 数量 | | 2011-04-10 | P2 | 120 | | 2011-04-15 | P1 | 100 | | 2011-04-22 | P4 | 50 | | 2011-04-30 | P8 | 80 | | 2011-05-06 | P1 | 100 | | 2011-05-08 | P3 | 70 |  |  |  | | --- | --- | | 製品 | | | 製品コード | 製品名 | | P1 | PC | | P2 | テレビ | | P3 | 掃除機 | | P4 | 冷蔵庫 | | P5 | エアコン | | P6 | 電話機 | | P7 | 時計 |  |  |  |  | | --- | --- | --- | | ア | 製品名 | 数量 | | テレビ | 120 | | PC | 100 | | 冷蔵庫 | 50 | | 掃除機 | 70 | |  | | |  |  |  |  | | --- | --- | --- | | イ | 製品名 | 数量 | | テレビ | 120 | | PC | 200 | | 冷蔵庫 | 50 | | 掃除機 | 70 | |  | | |  |  |  |  | | --- | --- | --- | | ウ | 製品名 | 数量 | | テレビ | 120 | | PC | 100 | | 冷蔵庫 | 50 | | PC | 100 | | 掃除機 | 70 | |  | | |  |  |  |  | | --- | --- | --- | | エ | 製品名 | 数量 | | テレビ | 120 | | PC | 100 | | 冷蔵庫 | 50 | | NULL | 80 | | PC | 100 | | 掃除機 | 70 | |  | | |   問題のSQL文は、表を結合し、製品名と数量を抽出します。結合条件では、２つの表の製品コードをもとに結合することになっているので、注文表の製品コードと製品表の製品コードが同じものの製品名とその数量を抽出します。  SELECT 製品名,数量 FROM 注文,製品  WHERE 注文.製品コード ＝ 製品.製品コード  結合条件  注文表にある製品コード“P8”については、製品表にはないため抽出されません。  基本情報　平成24年度春　問30　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-522～526

#### ⑪表の結合（その２）

複数の表を結合してデータを抽出する場合、⑩で示したようにWHERE句で結合条件を指定する以外に、FROM句でINNER JOINを使って結合することもできます。

書式

SELECT 列名,… FROM

表名１ INNER JOIN 表名２

↑

結合する表を記述する

ON 表名１.列名 = 表名２.列名

↑

関連付ける列を記述する

（結合条件）

（例）「成績」表と「担任」表から、学生の氏名とそれぞれの担任名を抽出する。

|  |  |
| --- | --- |
| 氏名 | 担任名 |
| 青木光二 | 木下恵子 |
| 太田治 | 横山大介 |
| 小島浩 | 小林直美 |
| 鈴木正明 | 横山大介 |
| 内藤俊彦 | 小林直美 |
| 山本弘一 | 木下恵子 |
| 吉田和夫 | 木下恵子 |
| 青山隆 | 横山大介 |

SELECT 氏名,担任名 FROM

成績 INNER JOIN 担任

ON 成績.クラス = 担任.クラス

なお、ON句で結合条件を指定する代わりにUSING句を使用して結合列を記述することができます。

書式

SELECT 列名,… FROM

表名１ INNER JOIN 表名２

USING (列名,…)

また、３つ以上の表を結合する場合には、( )でくくり、( )内のINNER JOINを１つの表とみなして結合します。

書式

SELECT 項目名,… FROM

表名１ INNER JOIN

(表名２ INNER JOIN 表名３

ON 表名２.列名 = 表名３.列名)

ON 表名１.列名 = 表名３.列名

表の結合（その２）

#### ⑫外部結合

WHERE句で表の結合条件を指定する場合、指定された表の両方に存在する行だけが抽出されます。これに対して、どちらかの表を優先し、優先された表の行を全て抽出するようにして結合する方法があります。このような結合方法を外部結合と呼びます。

書式

SELECT 列名,… FROM 表名１

LEFT OUTER JOIN 表名２

ON 表名１.列名 = 表名２.列名

※ OUTERは省略可。

（例）「成績」表と「担任」表を外部結合によって、「担任」表の全ての行を抽出する。

SELECT \* FROM 担任

LEFT OUTER JOIN 成績

ON 担任.クラス = 成績.クラス

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 担任名 | 担任.クラス | 成績.クラス | 学生番号 | 氏名 | 国語 | 数学 |
| 木下恵子 | 1 | 1 | 86001 | 青木光二 | 90 | 100 |
| 木下恵子 | 1 | 1 | 87003 | 山本弘一 | 95 | 80 |
| 木下恵子 | 1 | 1 | 87004 | 吉田和夫 | 85 | 90 |
| 横山大介 | 2 | 2 | 86002 | 太田治 | 75 | 70 |
| 横山大介 | 2 | 2 | 87001 | 鈴木正明 | 100 | 50 |
| 横山大介 | 2 | 2 | 87005 | 青山隆 | 50 | 60 |
| 小林直美 | 3 | 3 | 86003 | 小島浩 | 70 | 60 |
| 小林直美 | 3 | 3 | 87002 | 内藤俊彦 | 90 | 100 |
| 多和正敏 | 4 |  |  |  |  |  |

「成績」表に該当するデータがない場合には、空白「NULL」となります。

なお、「LEFT OUTER JOIN」の代わりに、「RIGHT OUTER JOIN」とすると、右側の表名２の全ての行が表示されます。

書式

SELECT 列名,… FROM 表名１

RIGHT OUTER JOIN 表名２

ON 表名１.列名 = 表名２.列名

※ OUTERは省略可。

外部結合

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  “商品”表と“売上明細”表に対して，次のSQL文を実行した結果の表として，正しいものはどれか。ここで，結果の表中の“－”は，値がナルであることを示す。  SELECT X.商品番号, 商品名, 数量  FROM 商品 X LEFT OUTER JOIN 売上明細 Y  ON X.商品番号 = Y.商品番号   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | 商品 |  |  | 売上明細 |  |  |  |  | | 商品番号 | 商品名 |  | 売上番号 | 売上日 | 商品番号 | 数量 | 売上金額 | | S101 | A |  | U001 | 2006-02-10 | S101 | 5 | 7,500 | | S102 | B |  | U002 | 2006-02-26 | S104 | 2 | 4,000 | | S103 | C |  | U002 | 2006-02-26 | S101 | 10 | 15,000 | | S104 | D |  | U003 | 2006-03-05 | S103 | 5 | 5,000 | |  |  |  | U003 | 2006-03-05 | S104 | 8 | 16,000 |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ア | 商品番号 | 商品名 | 数量 |  | イ | 商品番号 | 商品名 | 数量 | |  | S101 | A | 5 |  |  | S101 | A | 5 | |  | S101 | A | 10 |  |  | S101 | A | 10 | |  | S102 | B | ― |  |  | S103 | C | 5 | |  | S103 | C | 5 |  |  | S104 | D | 2 | |  | S104 | D | 2 |  |  | S104 | D | 8 | |  | S104 | D | 8 |  |  |  |  |  | |  |  |  |  |  |  |  |  |  | | ウ | 商品番号 | 商品名 | 数量 |  | エ | 商品番号 | 商品名 | 数量 | |  | S101 | A | 15 |  |  | S101 | A | 15 | |  | S102 | B | ― |  |  | S103 | C | 5 | |  | S103 | C | 5 |  |  | S104 | D | 10 | |  | S104 | D | 10 |  |  |  |  |  |   商品 X LEFT OUTER JOIN 売上明細 Y　とあるので、「商品」表が優先され、商品番号をもとに「商品」表と「売上明細」表を結合します。「売上明細」表にデータがない商品番号S102については、「売上明細」表の項目である数量欄の値はナル「NULL（空値）」となります。  データベース　平成18年度春　問32　[出題頻度：★☆☆]  解答－ア |

#### ⑬副問合せ

抽出結果に基づいて、別の表の抽出を行うことがあります。このような方法を副問合せと呼び、次のように記述します。

書式

SELECT 列名,… FROM 表名 WHERE 列名

＝（SELECT 列名 FROM 表名 WHERE 条件）

↑

先に実行する括弧の中のSELECT文により抽出される値は一つ

（例１）「成績」表と「担任」表をもとに、“横山大介”が担任をしている学生の氏名を抽出する。

|  |
| --- |
| 氏名 |
| 太田治 |
| 鈴木正明 |
| 青山隆 |

SELECT 氏名 FROM 成績 WHERE クラス

= (SELECT クラス FROM 担任

WHERE 担任名 ＝ '横山大介')

（　）内に記述された「SELECT」文により複数の値が抽出されるときは、「＝」の代わりに「IN」を記述します。

書式

SELECT 列名,… FROM 表名 WHERE 列名

IN（SELECT 列名 FROM 表名 WHERE 条件）

↑

先に実行する括弧の中のSELECT文により抽出される値は複数

（例２）「成績」表と「担任」表から、クラスごとの数学の平均点が80点以上のクラスの担任名を抽出する。

|  |
| --- |
| 担任名 |
| 木下恵子 |
| 小林直美 |

SELECT 担任名 FROM 担任 WHERE クラス

IN(SELECT クラス FROM 成績

GROUP BY クラス HAVING AVG(数学) >= 80)

なお、「IN」の直前に「NOT」を記述すると、カッコ内の「SELECT」文で取り出された値を除いた値が、抽出の対象になります。

書式

SELECT 列名,… FROM 表名 WHERE 列名

NOT IN（SELECT 列名 FROM 表名 WHERE 条件）

副問合せ

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  二つの表“納品”，“顧客”に対する次のSQL文と同じ結果が得られるSQL文はどれか。  SELECT 顧客番号, 顧客名 FROM 顧客  WHERE 顧客番号 IN  (SELECT 顧客番号 FROM 納品  WHERE 商品番号 = 'G1')   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 納品 |  |  |  | 顧客 |  | | 商品番号 | 顧客番号 | 納品数量 |  | 顧客番号 | 顧客名 |   ア　SELECT 顧客番号, 顧客名 FROM 顧客  WHERE 'G1' IN (SELECT 商品番号 FROM 納品)  イ　SELECT 顧客番号, 顧客名 FROM 顧客  WHERE 商品番号 IN  (SELECT 商品番号 FROM 納品  WHERE 商品番号 = 'G1')  ウ　SELECT 顧客番号, 顧客名 FROM 納品, 顧客  WHERE 商品番号 = 'G1'  エ　SELECT 顧客番号, 顧客名 FROM 納品, 顧客  WHERE 納品.顧客番号 = 顧客.顧客番号 AND 商品番号 = 'G1'  まず、副問合せ文(SELECT 顧客番号 FROM 納品 WHERE 商品番号 = 'G1')が実行され、“納品”表から、商品番号が‘G1’の商品を仕入れた顧客の顧客番号が抽出されます。  続いて、SELECT 顧客番号, 顧客名 FROM 顧客 WHERE 顧客番号 INが実行され、副問合せ文で抽出された顧客番号と同じ顧客番号と顧客名が“顧客”表から抽出されます。  ア、イ　商品番号は、“顧客”表の検索条件にはできません。  ウ　“納品”表と“顧客”表の直積から、商品番号が‘G1’のものが選択されます。  データベース　平成17年度春　問37　[出題頻度：★☆☆]  解答－エ |

別冊演習ドリル 》 1-527,528

#### ⑭相関副問合せ

主問合せの表の１行ごとに副問合せを実行し、主問合せで指定されたデータの存在をチェック、その結果に基づいてデータを抽出します。

書式

SELECT 列名,… FROM 表名

WHERE EXISTS

(SELECT 列名 FROM 表名 WHERE 条件)

（例１）「成績」表と「担任」表から、「成績」表にデータが存在するクラスと担任名を抽出する。

SELECT クラス,担任名 FROM 担任

|  |  |
| --- | --- |
| クラス | 担任名 |
| 1 | 木下恵子 |
| 2 | 横山大介 |
| 3 | 小林直美 |

WHERE EXISTS

(SELECT \* FROM 成績

WHERE 成績.クラス = 担任.クラス)

なお、「EXISTS」の直前に「NOT」を記述すると、真と偽の判定が逆となり、データが存在しない場合に抽出されます。

書式

SELECT 列名,… FROM 表名

WHERE NOT EXISTS

（SELECT 列名 FROM 表名 WHERE 条件）

（例２）「成績」表と「担任」表をもとに、「成績」表にデータが存在しないクラスと担任名を抽出する。

|  |  |
| --- | --- |
| クラス | 担任名 |
| 4 | 多和正敏 |

SELECT クラス,担任名 FROM 担任

WHERE NOT EXISTS

(SELECT \* FROM 成績

WHERE 成績.クラス = 担任.クラス)

相関副問合せ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  “商品”表，“在庫”表に対する次のSQL文の結果と同じ結果が得られるSQL文はどれか。ここで，下線部は主キーを表す。  SELECT 商品番号 FROM 商品  WHERE 商品番号 NOT IN (SELECT 商品番号 FROM 在庫)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 商品 |  |  |  | 在庫 |  |  | | 商品番号 | 商品名 | 単価 |  | 倉庫番号 | 商品番号 | 在庫数 |   ア　SELECT 商品番号 FROM 在庫  WHERE EXISTS (SELECT 商品番号 FROM 商品)  イ　SELECT 商品番号 FROM 在庫  WHERE NOT EXISTS (SELECT 商品番号 FROM 商品)  ウ　SELECT 商品番号 FROM 商品  WHERE EXISTS (SELECT 商品番号 FROM 在庫  WHERE 商品.商品番号 = 在庫.商品番号)  エ　SELECT 商品番号 FROM 商品  WHERE NOT EXISTS (SELECT 商品番号 FROM 在庫  WHERE 商品.商品番号 = 在庫.商品番号)  EXISTS句は、条件にあてはまるレコードが存在している場合には真を返し、存在していない場合には偽を返します。またNOT EXISTSであるときには逆に条件に当てはまるレコードが存在している場合は偽を返し、存在していない場合には真を返します。  問題ではNOT INを使用して、在庫表にない商品番号を商品表から抽出しているので、選択肢ア、イではありません。ウではEXISTSのみなので在庫表にある商品番号を抽出します。したがって正解はNOT EXISTSを使用しているエとなります。  基本情報　平成26年度春　問28　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-529,530

🏋プラスアルファ

**●SELECT文の統合**

別々の表として集計した結果を１つの表として参照したい場合など、SELECT文の結果を統合して、１つの結果とすることができます。

書式

SELECT 列名,… FROM 表名

WHERE 条件

UNION

SELECT 列名,… FROM 表名

WHERE 条件

統合は、複数表の要素を加え、重複を取り除きます。

（例）クラス別に抽出した３つの「成績」表を統合する。

SELECT \* FROM 成績

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生番号 | 氏名 | 国語 | 数学 | クラス |
| 86001 | 青木光二 | 90 | 100 | 1 |
| 87003 | 山本弘一 | 95 | 80 | 1 |
| 87004 | 吉田和夫 | 85 | 90 | 1 |
| 86002 | 太田治 | 75 | 70 | 2 |
| 87001 | 鈴木正明 | 100 | 50 | 2 |
| 87005 | 青山隆 | 50 | 60 | 2 |
| 86003 | 小島浩 | 70 | 60 | 3 |
| 87002 | 内藤俊彦 | 90 | 100 | 3 |

WHERE クラス = 1

UNION

SELECT \* FROM 成績

WHERE クラス = 2

UNION

SELECT \* FROM 成績

WHERE クラス = 3

なお、「UNION ALL」と記述すると、重複は取り除かれずに、全ての行が抽出されます。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題　プラスアルファ  “科目”表と“実習”表に対して，次のSQL文を実行した結果，導出される表はどれか。  SELECT 科目.科目番号  FROM 科目,実習  WHERE 科目.科目番号 = 実習.科目番号  UNION  SELECT 科目.科目番号  FROM 科目  WHERE 単位数 >= 5   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 科目 |  |  |  | 実習 |  |  | | 科目番号 | 科目名 | 単位数 |  | 実習番号 | 実習名 | 科目番号 | | 1 | 国文学 | 5 |  | A1 | 重力実験 | 2 | | 2 | 物理学 | 6 |  | A2 | 発光反応 | 5 | | 3 | 数学 | 6 |  |  |  |  | | 4 | 英文学 | 4 |  |  |  |  | | 5 | 化学 | 3 |  |  |  |  | | 6 | 世界史 | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | ア | 科目番号 |  | イ | 科目番号 |  | ウ | 科目番号 |  | エ | 科目番号 | |  | 1 |  |  | 1 |  |  | 2 |  |  | 5 | |  | 2 |  |  | 2 |  |  |  |  |  |  | |  | 2 |  |  | 3 |  |  |  |  |  |  | |  | 3 |  |  | 5 |  |  |  |  |  |  | |  | 5 |  |  |  |  |  |  |  |  |  |   UNIONは、２つの問合せ（SELECT文）の出力結果を統合します。統合は、複数表の要素を加え、その後、重複を取り除くため、次のようになります。  前半のSELECT文  後半のSELECT文  和演算  ＝   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 科目番号 |  | 科目番号 |  | 科目番号 | | 2 |  | 1 |  | 1 | | 5 |  | 2 |  | 2 | |  |  | 3 |  | 3 | |  |  |  |  | 5 |   ソフトウェア開発　平成18年度秋　問66　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-531

### ５）データ定義言語（その２）

CREATE TABLEで作成された実表を元に作成される仮想的な表がビュー表です。実表から必要なデータを抜き出すことができるため、操作が簡単になります。また、データベース管理者が、利用者の実表アクセスを禁止する代わりに、ビュー表によって必要なデータだけを提供することで、データの機密保護が可能になります。

#### ①ビュー表の定義

ビュー表を定義する場合、次のように記述します。

書式

CREATE VIEW 表名

（列名１,列名２,…）

AS SELECT …

（例）「成績」表から、学生番号、氏名、国語の成績を抜き出したビュー表「国語成績」表を定義する。

|  |  |  |
| --- | --- | --- |
| 国語成績 | | |
| 学生番号 | 氏名 | 国語 |
| 86001 | 青木光二 | 90 |
| 87003 | 山本弘一 | 95 |
| 87004 | 吉田和夫 | 85 |
| 86002 | 太田治 | 75 |
| 87001 | 鈴木正明 | 100 |
| 87005 | 青山隆 | 50 |
| 86003 | 小島浩 | 70 |
| 87002 | 内藤俊彦 | 90 |

CREATE VIEW 国語成績

（学生番号,氏名,国語）

AS SELECT 学生番号,氏名,国語

FROM 成績

ビュー表の定義

ビュー表に対する更新は、更新するビュー表の行が、実表上のどの行かを特定できる場合は可能です。ビュー表の定義で、DISTINCTや集約関数、演算式、GROUP BY句、HAVING句、副問合せ文などを使用して、実表のデータを加工して表示するなど、複数行をまとめるような処理を行っている場合は、実表上の更新箇所を特定できないため、更新できません。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 例題  四つの表“注文”，“顧客”，“商品”，“注文明細”がある。これらの表から，次のビュー”注文一覧”を作成するSQL文はどれか。ここで，下線の項目は主キーを表す。  注文（注文番号,注文日,顧客番号）  顧客（顧客番号,顧客名）  商品（商品番号,商品名）  注文明細（注文番号,商品番号,数量,単価）   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 注文一覧 |  |  |  |  |  | | 注文番号 | 注文日 | 顧客名 | 商品名 | 数量 | 単価 | | 001 | 2006-01-10 | 佐藤 | AAAA | 5 | 5,000 | | 001 | 2006-01-10 | 佐藤 | BBBB | 3 | 4,000 | | 002 | 2006-01-15 | 田中 | BBBB | 6 | 4,000 | | 003 | 2006-01-20 | 高橋 | AAAA | 3 | 5,000 | | 003 | 2006-01-20 | 高橋 | CCCC | 10 | 1,000 |   ア　CREATE VIEW 注文一覧  AS SELECT ＊ FROM 注文,顧客,商品,注文明細  WHERE 注文.注文番号 = 注文明細.注文番号 AND  注文.顧客番号 = 顧客.顧客番号 AND  商品.商品番号 = 注文明細.商品番号  イ　CREATE VIEW 注文一覧  AS SELECT 注文.注文番号,注文日,顧客名,商品名,数量,単価  FROM 注文,顧客,商品,注文明細  WHERE 注文.注文番号 = 注文明細.注文番号 AND  注文.顧客番号 = 顧客.顧客番号 AND  商品.商品番号 = 注文明細.商品番号  ウ　CREATE VIEW 注文一覧  AS SELECT 注文.注文番号,注文日,顧客名,商品名,数量,単価  FROM 注文,顧客,商品,注文明細  WHERE 注文.注文番号 = 注文明細.注文番号 OR  注文.顧客番号 = 顧客.顧客番号 OR  商品.商品番号 = 注文明細.商品番号  エ　CREATE VIEW 注文一覧  AS SELECT 注文.注文番号,注文日,商品名,数量,単価  FROM 注文,商品,注文明細  WHERE 注文.注文番号 = 注文明細.注文番号 AND  商品.商品番号 = 注文明細.商品番号  ア　全項目を抽出する命令になっているため、“注文一覧”にはない顧客番号や商品番号も抽出されてしまいます。  ウ　各表の結合条件をORでつないでいるため、１つの結合条件を満たしているだけで抽出されてしまいます。  エ　“注文”表と“顧客”表を結合する条件が含まれていません。  ソフトウェア開発　平成18年度春　問66　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-532～536

🏋プラスアルファ

**●権限の定義**

表に対するユーザのアクセス権限を定義することができます。アクセス権限には、SELECT権限、INSERT権限、DELETE権限、UPDATE権限などがあります。権限を定義することで、データの保護が可能になります。

特定のユーザに特定の作業を行う権限を与える場合、次のように記述します。

書式

GRANT 権限 ON 表名 TO 対象者

（例）「成績」表の国語の成績を更新する権限を、国語の担当者「小林直美」に付与する。

GRANT UPDATE 国語 ON 成績 TO 小林直美

なお、全ての権限を付与する場合には以下のように記述します。

GRANT ALL PRIVILEGES ON 表名 TO 対象者

特定のユーザに与えた権限を剥奪する場合、次のように記述します。

REVOKE 権限 ON 表名 FROM 対象者

（例）「成績」表に学生データを挿入する権限を、「渡辺圭一」から剥奪する。

REVOKE INSERT ON 成績 FROM 渡辺圭一

|  |
| --- |
| 例題　プラスアルファ  “商品”表へのアクセス権限のうち，行挿入権限とデータ更新権限だけをUSER01とUSER02の２利用者に与えるSQL文はどれか。  ア　GRANT ALL PRIVILEGES ON 商品 TO USER01,USER02  イ　GRANT INSERT, SELECT ON 商品 TO USER01,USER02  ウ　GRANT UPDATE, INSERT ON 商品 TO USER01,USER02  エ　REVOKE UPDATE, INSERT ON 商品 TO USER01,USER02  ア　「GRANT ALL PRIVILEGES」から、全ての権限を与えていることが分かります。  イ　「GRANT INSERT、 SELECT」から、行の挿入とデータの検索の権限を与えていることが分かります。  エ　「REVOKE UPDATE、 INSERT」から、以前与えたデータの更新と行の挿入の権限を剥奪していることが分かります。  データベース　平成14年度春　問27　[出題頻度：★☆☆]  解答－ウ |

別冊演習ドリル 》 1-537

🏋プラスアルファ

**●埋込型SQL**

埋込型SQLは、ＣやJava、COBOLなどのアプリケーションプログラム中に記述するSQL文です。

アプリケーションプログラムに埋め込まれたSQLを用いてデータベースを操作する場合、SQL文の実行結果を取り出すために用いられるのがカーソルです。**カーソル**は、プログラムがデータベースを操作する際の場所を指し示す変数にあたります。

一般に、SELECT文によって抽出されたデータは複数行から成り立っています。これは一括処理には適していますが、アプリケーションプログラムでは抽出されたデータを１行ずつ処理することがあります。カーソルはこのような場合に用いられ、抽出結果を一時的な作業領域に蓄えておき、そこから１行ずつ取り出してアプリケーションプログラムとやり取りします。

**①カーソルの宣言**

まず、最初に対象となるSELECT文を定義するため、次のように記述します。

書式

DECLARE カーソル名 CURSOR FOR SELECT文

**②カーソルのオープン**

次に、カーソルを利用するため、次のように記述します。その結果、定義されたSELECT文が実行され、抽出結果にアクセスでき  
　　るようになります。

書式

OPEN カーソル名

**③データの取り出し**

続いて開いたカーソルから、１行ごとにデータを取り出すため、次のように記述します。

書式

FETCH カーソル名 INTO 作業領域

***プラスアルファ***

**＋α**

**④データの更新**

カーソルが現在位置している行のデータを更新する場合には、次のように記述します。

書式

UPDATE 表名 SET 列名 = 変更データ

WHERE CURRENT OF カーソル名

**⑤カーソルのクローズ**

最後に、目的とするデータ処理を全て終えた時点で確保していた作業領域を開放するため、次のように記述します。

書式

CLOSE カーソル名

DEALLOCATE カーソル名

|  |
| --- |
| 例題　プラスアルファ  次の埋込みSQLを用いたプログラムの一部において，Xは何を表す名前か。  EXEC SQL OPEN X;  EXEC SQL FETCH X INTO :NAME, :DEPT, :SALARY;  EXEC SQL UPDATE 従業員  SET 給与 = 給与 \* 1.1  WHERE CURRENT OF X;  EXEC SQL CLOSE X;  ア　カーソル イ　スキーマ ウ　テーブル エ　ビュー  イ　スキーマは、データの性質、形式、ほかのデータとの関連などのデータ定義の集合です。  ウ　テーブルは、データ要素の集合を２次元の表で構成したものです。  エ　ビューは、抽出結果をテーブルのように扱うための仮想表です。  基本情報　平成30年度春　問28　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-538

# 4. トランザクション処理

　トランザクションとは、システムで処理される、複数の処理を１つにまとめ、分離できない処理の単位です。

Ａさんの銀行口座からＢさんの銀行口座に１万円振り込む

トランザクション

処理１

Ａさんの口座から10,000円差し引いた

残高で更新する

処理２

Ｂさんの口座に10,000円加えた

残高で更新する

※処理１と処理２の順番を変えることはできますが、どちらかだけが処理されるということはありません。

トランザクションの例

## 1. 排他制御

学習のポイント

✅ 登場する用語の意味を覚えよう！

同一データに対して複数の利用者が同時に更新（データの書換え）処理を行うことを認めると、処理結果に矛盾が生じる場合があります。排他制御は、複数のユーザが同時に同じデータをアクセスした場合、ある利用者の利用中は、他の利用者が使えないようにする機能です。

### １）排他制御を用いない場合

排他制御を用いない場合、次のように内容に論理的矛盾が生じることがあります。

排他制御を用いない場合のデータベースの状態

①　プログラムＡがデータＸを読む。

②　プログラムＢがデータＸを読む。

③　プログラムＡでＸに50を加え、その結果をデータベースに書く。

④　プログラムＢでＸから50を引き、その結果をデータベースに書く。

①～④の処理が、この順番で行われるとデータは100になってしまう。

Ｘを読む

Ｘに50を加える

Ｘの更新処理

Ｘは150

Ｘは200

プログラムＡ

データベース

Ｘ

③

①

②

④

Ｘを読む

Ｘから50を引く

Ｘは150

Ｘは100

プログラムＢ

Ｘの更新処理

150

↓

200

↓

100

排他制御を用いない場合の更新（例）

### ２）排他制御を用いた場合

排他制御は、ある利用者がデータを使用している間は、当該データをロックすることで他の利用者の使用を禁止する方法です。

ロックには、データ参照中に行う共有ロックと、データ更新中に行う専有（占有）ロックがあります。**共有ロック**は、他の利用者はデータを参照できますが更新はできません。また、ある利用者が共有ロックした資源を他の利用者も共有ロックできますが、専有ロックはできません。これに対して**専有ロック**は、他の利用者はデータの参照も更新もできません。また、専有ロック中の資源は共有ロックも専有ロックもできません。そのためロックが解除されるまで資源待ちが発生します。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 他の利用者の操作 | | | |
|  |  | 参照処理 | 更新処理 | 共有ロック | 専有ロック |
| ロックの種類 | 共有ロック | **○** | × | **○** | × |
| 専有ロック | × | × | × | × |

なお、ロックするデータの範囲を**ロックの粒度**（ロックレベル）と呼びます。ロックの粒度を大きくすれば、資源待ちの確率が高くなり、全体の処理の効率が低下します。逆にロックの粒度を小さくすれば、同時更新の確率は下がりますが、頻繁にロックをかけたり解除する処理が必要となり、CPUの負荷が増大します。そのため、業務の特徴を考慮して、最適な粒度にすることが求められます。

排他制御を用いて同様の処理を行うと、次のようになります。

排他制御を用いた場合のデータベースの状態

①　プログラムＡがデータＸを読むと同時にロックをかける。ロックは他の利用者のアクセスを禁止し、処理の終了まで待たせる機能である。

②　プログラムＢはプログラムＡの終了後、データＸを読み、処理を行う。

③　両プログラムの実行後の結果は、150となる。

Ｘを読む

Ｘに50を加える

Ｘの更新処理

Ｘは150

Ｘは200

プログラムＡ

Ｘを読む

Ｘから50を引く

Ｘの更新処理

プログラムＢ

データベース

150

↓

200

Ｘ

アクセスできない

ロック

プログラムＡの処理終了後

処理終了

プログラムＡ

Ｘを読む

Ｘから50を引く

Ｘの更新処理

プログラムＢ

Ｘは200

Ｘは150

データベース

200

↓

150

Ｘ

ロックの解除

排他制御を用いた場合の更新（例）

ただし、排他制御を用いた場合、デッドロックの発生に注意する必要があります。

次の①～④の順で処理を行う場合

①　プログラムＡがデータＸを読むと同時にロックをかける。

②　プログラムＢがデータＹを読むと同時にロックをかける。

③　プログラムＡがデータＹを読む。

④　プログラムＢがデータＸを読む。

③

①

データベース

データＸ

データＹ

Ｘを読む

Ｙを読む

プログラムＡ

※Ｙは利用中なので

待ち状態となる

ロック

②

④

Ｙを読む

プログラムＢ

Ｘを読む

※Ｘは利用中なので

待ち状態となる

ロック

デッドロック（例）

**デッドロック**とは、お互いの処理が待ち状態となり、処理が進まなくなってしまう状態です。データベース管理システムは、デッドロックが発生しないように待ち状態のプログラムの監視などを行っています。

|  |
| --- |
| 例題  DBMSの排他制御に関する記述のうち，適切なものはどれか。  ア　アクセス頻度の高いデータの処理速度を上げるためには，排他制御が必要である。  イ　処理速度を上げるため，排他制御を行うデータの範囲は極力大きくすべきである。  ウ　データアクセス時のデッドロックを防止するために，排他制御が必要である。  エ　複数の人が同時に更新する可能性のあるデータには，排他制御が必要である。  排他制御は、ある利用者がデータを使用している間は、当該データを他の利用者が使用することを禁止する制御です。なお、データを利用禁止にするためにロックするデータの範囲をロックの粒度（ロックレベル）と呼びます。ロックの粒度を大きくすれば、他の利用者の資源待ちの確率が高くなり、全体の処理の効率が低下します。排他制御では、複数の利用者が他者が必要とする資源を排他的に使用してしまうことで、お互いに待ち状態となり処理が停止する、デッドロックの発生に注意する必要があります。  基本情報　平成24年度春　問33　[出題頻度：★★★]  解答－エ |

別冊演習ドリル 》 1-539～547

### ３）コミットメント制御

データベースでは、UPDATE（データの更新）やDELETE（データの削除）、INSERT（データの挿入）などの命令が発行されても、物理的には更新されません。トランザクション処理が正常に終了し、データベースの更新が確定するのはCOMMIT（コミット：更新確定）命令が発行されたときです。しかし、トランザクション処理が異常終了した場合には、トランザクション処理の開始前の状態に戻すことになります。その場合には、ROLLBACK（ロールバック：更新取消）命令を発行し、最後にCOMMIT命令が発行されてからの全ての命令を取り消します。

●正常に終了する場合（コミット）

DB

Ａさんの残高を  
１万円減らす

Ｂさんの残高を  
１万円増やす

完了

（コミット）

…

…

更新前ログ

Ａ残高10万円

Ｂ残高 ５万円

更新後ログ

Ａ残高９万円

Ｂ残高６万円

●トランザクション障害が発生した場合（ロールバック）

取り消し

（ロールバック）

DB

Ａさんの残高を  
１万円減らす

Ｂさんの残高を  
１万円増やす

更新前ログ

Ａ残高10万円

Ｂ残高 ５万円

※更新前ログを用いて、トランザクションの開始前に戻す。

例）Ａさんの銀行口座からＢさんの銀行口座に１万円振り込む

|  |
| --- |
| 例題  SQLによるトランザクション管理に関する記述として，適切なものはどれか。  ア　トランザクションがCOMMIT文を実行した後にROLLBACK文を実行した場合，そのトランザクション内でのすべての更新は，他のトランザクションには認識されない。  イ　トランザクションがROLLBACK文を実行した後にデータベースを更新し，さらにCOMMIT文を実行した場合，ROLLBACK文の実行後のデータベース更新だけが，他のトランザクションに認識される。  ウ　トランザクションによって実行されたデータベースの更新は，そのトランザクションがCOMMIT文を実行しても，トランザクションが終了するまで，他のトランザクションには認識されない。  エ　トランザクションの開始と終了の間では，同一カーソルのOPENとCLOSEを繰り返しても，検索される行は常に同じである。  ROLLBACK文を実行すると、以前に実行されたデータベースの更新処理が無効になります。その後、COMMIT文を実行すると、ROLLBACK文の実行後に行われたデータベースの更新だけが確定します。  ア　COMMIT文を実行したことでデータベースの更新は確定しているので、その後にROLLBACK文を実行しても、そのトランザクション内での全ての更新は、他のトランザクションに認識されます。  ウ　COMMIT文を実行した時点でデータベースの更新が確定するので、トランザクションによって実行されたデータベースの更新は他のトランザクションにも認識されます。  エ　カーソルを移動させれば対象となる行は異なります。  基本情報　平成20年度１月　問34　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-548,549

## 2. 障害回復

学習のポイント

✅ 出題パターンが多い項目！例題や演習ドリルの問題を解いて、解答力を身につけよう！

データベースは、障害に備えて定期的にバックアップを取っておく必要があります。仮に、データベースに障害が発生した場合は、バックアップファイルと更新データを書き出している**ログファイル**（**ジャーナルファイル**）を用いてデータベースを復旧します。

復旧方法には、次の表に示すように**ロールバック**と**ロールフォワード**の２つがあります。

|  |  |  |
| --- | --- | --- |
| 復旧方法 | 障害の種類 | 説　明 |
| ロールバック処理 | トランザクション障害 | プログラム障害などの論理的な誤作動が発生したときに行う。障害発生時のデータベースの内容とログファイルの更新前のイメージを用いて、更新前の状態に戻す。 |
| ロールフォワード処理 | 媒体障害 | 磁気ディスクの故障などの物理的な障害が発生したときに行う。バックアップファイルの内容を用いて処理前の状態に戻し、ログファイルの更新後イメージにより順次更新し、障害発生直前の状態に戻す。 |

障害復旧の方法

●障害発生までの状況

DB

トランザクション１

トランザクション２

DB

更新前ログ①

更新後ログ①

更新前ログ②

更新後ログ②

DB

バックアップファイル

故障

●ロールフォワード

バックアップファイルに更新後ログの情報を反映する

DB

DB

更新後ログ②

障害発生直前の

状態に復旧

更新後ログ①

ロールフォワード処理

なお、データベースの更新頻度が多い場合には、**チェックポイント**を設定してデータの回復に備える場合があります。チェックポイントは一定の間隔で設定され、このタイミングでデータベースを更新します。障害発生時には、チェックポイント時の状態にデータベースを戻し、復旧作業を行います。

終了

チェックポイント

障害発生

トランザクションＡ

トランザクションＢ

トランザクションＣ

開始

終了

開始

開始

①　チェックポイント時より前に処理が終了している。（トランザクションＡ）

チェックポイント時の状態に復旧することで、トランザクション処理が正常に終了した状態になる。

②　チェックポイント時には処理中、障害発生時より前に処理が終了している。（トランザクションＢ）

チェックポイント時の状態から、ロールフォワード処理により、トランザクションが正常に終了した状態に復旧する。

③　チェックポイント時には処理中、障害発生時に処理が終了していない。（トランザクションＣ）

チェックポイント時の状態から、ロールバック処理により、トランザクション開始前の状態に復旧し、トランザクション処理を最初からやり直す。

チェックポイントによる復旧

なお、電源を切ることなくコンピュータを再スタートさせることを**ウォームスタート**と呼びます。通常は、ソフトウェアをリセットすることで再起動します。

ウォームスタートが行われると、更新前ログによってチェックポイントまで戻り、更新後ログを用いてエラーとなった処理（トランザクション）の障害を回復します。すなわちウォームスタートは、ロールフォワードとロールバックを組み合わせてデータベースの障害回復を行い、システムを再起動させる方式です。

これに対して、データベースの障害回復を行うことなくシステムを再起動する方法を、**コールドスタート**と呼びます。

|  |
| --- |
| 例題  データベースの障害回復に関する記述のうち，適切なものはどれか。  ア　データベースの信頼性を高めるために，同一のコピーをもつよりは，常に一世代前の内容を保持することがシステムダウン対策として効果的である。  イ　ログファイルやジャーナルファイルには，データベース更新以前の情報よりも，むしろ記録の意味で更新後の情報を格納する。  ウ　ロールバックとは，OLTPなどの実行中に障害が発生したとき，トランザクション開始直前の状態にデータベースを復旧する処理をいう。  エ　ロールフォワードでは，定期的に取得したデータベースのダンプを書き戻すことでデータベースを復旧するので，障害発生時に更新データの一部が反映されないものの，高速な復旧が期待できる。  ア　システム復旧の際、迅速な対応を行うためには、できる限り最新のコピー（バックアップデータ）を用意することが望ましいです。  イ　ログファイル（ジャーナルファイル）には、更新前と更新後の両方のデータを格納する必要があります。  エ　ロールフォワードでは、バックアップデータを用いて一定時間前の状態にデータベースを復旧したあと、ログファイル（ジャーナルファイル）の更新後情報を用いて障害発生前までの更新を反映します。  基本情報　平成17年度春　問47　[出題頻度：★★★]  解答－ウ |

別冊演習ドリル 》 1-550～559

## 3. トランザクション管理

学習のポイント

✅ それぞれ名称と意味を覚えよう！

### １）ACID（アシッド）特性

データベースの処理は、トランザクション単位に管理されます。トランザクションのACID特性は、データの状態を矛盾なく、一貫した状態に保つためにデータベースが備えていなければならない特性です。

|  |  |  |
| --- | --- | --- |
| 名　称 | | 説　明 |
| 原子性 | Atomicity | 全ての処理が完了または全く行われていない状態で処理が完了し、中間的な状態はないという性質である。 |
| 一貫性 | Consistency | トランザクションの終了状態に関係なく、データベースの内容に矛盾があってはならないという性質である。 |
| 独立性 | Isolation | 複数のトランザクションを同時に実行した場合と、順番に実行した場合とで、処理結果に違いがあってはならないという性質である。隔離性、直列可能性ともいう。 |
| 耐久性 | Durability | トランザクションが終了すれば、障害などによって結果が損なわれることはないという性質である。 |

ACID特性

|  |
| --- |
| 例題  トランザクションのACID特性に関する記述のうち，適切なものはどれか。  ア　コミット後にシステム障害が発生した場合，その内容は変更前の状態に戻される。  イ　トランザクションが同時に実行されても，互いに干渉しない。  ウ　トランザクションの実行の結果，データベースの整合性が崩れることも許容する。  エ　トランザクションの途中でシステム障害が発生しても，障害発生時までの変更内容は保存される。  選択肢イは、独立性に関する記述です。  ア　耐久性から、コミット後に内容が変更前の状態に戻ることはありません。  ウ　一貫性から、整合性が崩れることはありません。  エ　原子性から、トランザクションの途中で障害が発生した場合は、内容は変更前の状態に戻されます。  基本情報　平成18年度春　問60　[出題頻度：★★☆]  解答－イ |

別冊演習ドリル 》 1-560～564

## 4. データベースの性能向上

学習のポイント

✅ 例題を解くことができるように用語を覚えておこう！

データベースの検索を行う場合、アクセス効率を上げるために**インデックス**を付けるのが一般的です。インデックスの設定に際しては、検索条件だけでなく、テーブルのレコード数についても考慮する必要があります。ただし、更新の対象のデータにはインデックスも含まれるため、更新頻度が高い列に対してインデックスを付けると逆にアクセス効率が下がることになります。また、狭い範囲のデータにインデックスを付けた場合もアクセス効率は下がります。

特定のデータにアクセスが集中して入出力に時間がかかる場合には、アクセス効率を上げるためにデータベースを分割します。

なお、データベースに対してデータの追加や削除が繰り返されると、管理情報やデータの格納場所が断片化することで応答時間が長くなるなどの性能の劣化が起こります。そのため、一定期間ごとに**再編成**を行う必要があります。

|  |
| --- |
| 例題  関係データベースの表の列に利用者がインデックスを設定する目的はどれか。  ア　外部キーの列の値を別の表の主キーの値に一致させる。  イ　データの格納位置への効率的なアクセスが可能となり，検索速度の向上が期待できる。  ウ　一つの大きなテーブルを複数のディスクに分散格納する場合，ディスク容量が節約できる。  エ　列内に重複する値がないようにする。  インデックスは検索などを高速に行うために使用します。ただし、更新の際にはインデックスについても更新対象となるため、更新頻度が高い列に設けるとアクセス効率が悪くなってしまいます。  基本情報　平成24年度秋　問27　[出題頻度：★★☆]  解答－イ |

別冊演習ドリル 》 1-565～567

# 5. データベース応用

## 1. データベース応用

学習のポイント

✅ 登場する用語の意味を覚えよう！

データベースの目的は、当初は大量のデータを効率的に保管することでしたが、現在では、コンピュータ技術の進歩と外部状況（社会環境）の変化により、より幅広い目的に利用されています。

### １）データウェアハウス

データウェアハウスは、意思決定支援システムのための全社規模のデータベースです。オンライントランザクション処理（OLTP：OnLine Transaction Processing）などにより更新された基幹系データベースのデータを目的別に時系列で蓄積します。なお、データウェアハウスを構築するために、業務システムごとに異なっているデータ属性やコード体系を統一する処理を**データクレンジング**と呼びます。

#### ①オンライン解析処理

データウェアハウスに蓄積された大量のデータを、属性（項目）を切替えて検索できる多次元データベースに格納し、これをさまざまな角度から検索、集計して問題点や解決策を発見する処理をオンライン解析処理（**OLAP**：OnLine Analytical Processing）と呼びます。情報技術部門ではなく、エンドユーザが直接システムを操作して解析を行う点が従来の解析システムとは異なります。

代表的なオンライン解析処理には、スライシング、ダイス、ドリリングなどがあります。**スライシング**は、多次元データベースをある軸（dimension）で切って２次元の表にすることです。**ダイス**は、集計項目の軸を入れ替えて集計軸の異なるさまざまな表を表示することです。また、**ドリリング**は、取り扱うデータの集計レベルを変更する操作のことで、詳細化する操作をドリルダウン、逆の操作をロールアップと呼びます。そして、集計データの元になる明細データを分析することをドリルスルーと呼びます。

#### ②データマート

データウェアハウスに蓄積されたデータの中から、OLAPにより、目的別にデータを抽出し、作成されたデータベースをデータマートと呼びます。データマートを利用することで、データの検索時間が短縮されるという利点がありますが、データを絞り込んでしまうので、予想外の解析結果を見落としてしまう場合があります。

日常業務で利用されている

データベースや基幹データベース

データウェア

ハウス

整理・統合・集約

データ

マート

利用者の目的に応じて抽出

データ

マート

データ

マート

データウェアハウスのイメージ

|  |
| --- |
| 例題  企業の様々な活動を介して得られた大量のデータを整理・統合して蓄積しておき，意思決定支援などに利用するものはどれか。  ア　データアドミニストレーション イ　データウェアハウス  ウ　データディクショナリ エ　データマッピング  ア　データアドミニストレーションとは、データを管理することです。  ウ　データディクショナリとは、データベースに保存されているデータの属性や項目を管理する表のことです。  エ　データマッピングとは、異なるデータ間の関連付けを行うことです。  基本情報　平成22年度春　問33　[出題頻度：★★★]  解答－イ |

別冊演習ドリル 》 1-568～570

### ２）ビッグデータ

SNSや電子掲示板などのソーシャルメディアの多種多様な大量の書込み、センサの情報やサーバのログなど、従来のデータベースソフトウェアでは管理することが不可能な膨大な量のデータなどをビッグデータと呼びます。

ビッグデータは、意思決定に役立つ知見を得ることなどが期待されるデータで、大量（Volume）、テキスト、画像、音声といった多種多様なデータ形式（Variety）、リアルタイム性（Velocity）という特長があり、そのままの形式や構造で**データレイク**と呼ばれる貯蔵場所に格納されます。

コンピュータの処理能力が向上し、従来は分析不可能だったビッグデータは、パターン処理などをすることで詳細な分析が可能となりました。

また、データ項目を整理・統一しないと扱えない関係データベースでは、扱うデータの形式が多種多様なビッグデータは扱いにくいことから、従来の形式に捉われないデータベース形式に注目が集まりました。

このような、従来の関係データベース以外のデータベース形式は、関係データベースが主にSQLで操作されることから、**NoSQL**（“Not only SQL”の略）と呼ばれます。代表的なNoSQLとして、データを識別できる値（Key）とそれに関連付けられたデータ（Value）を組みとして保存する、**キーバリューストア**（Key-Value Store:KVS）があります。キーバリューストアは、構造が単純なため、応答速度に優れ、検索用のキー値と関連付けるだけでさまざまな形式のデータを格納することができるため、ビッグデータに向いているとされています。

|  |
| --- |
| 例題  ビッグデータの処理で使われるキーバリューストアの説明として，適切なものはどれか。  ア　“ノード”，“リレーションシップ”，“プロパティ”の３要素によってノード間の関係性を表現する。  イ　１件分のデータを“ドキュメント”と呼び，個々のドキュメントのデータ構造は自由であって，データを追加する都度変えることができる。  ウ　集合論に基づいて，行と列から成る2次元の表で表現する。  エ　任意の保存したいデータと，そのデータを一意に識別できる値を組みとして保存する。  ア　NoSQLデータベースの一種である、グラフデータベースの説明です。  イ　NoSQLデータベースの一種である、ドキュメント指向データベースの説明です。  ウ　関係データベース（リレーショナルデータベース）の説明です。  基本情報　平成31年度春　問30　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-571～572

### ３）データマイニング

データマイニングとは、さまざまな角度で分析した結果の中に潜む法則や因果関係などの意味情報を発掘することです。データマイニングでは、価値ある法則を見つけるために、膨大な角度からデータを分析し、さまざまな仮説を立てて検証をします。そのため、人工知能や統計学などの高度な数学的手法が用いられます。また、これらを行うため、大量のデータを高速に処理する機能と、条件に合わせたルールを発見するアルゴリズムを備えたソフトウェアが開発されています。

|  |
| --- |
| 例題  “缶ビールを購入する顧客は，スナック菓子を同時に買い求める傾向にある”というような，データベースに蓄積された大量のデータを分析して，新たな情報を得る技術はどれか。  ア　データウェアハウス イ　データエンティティ  ウ　データマート エ　データマイニング  ア　データウェアハウスは、企業戦略の立案や意思決定などに役立てるため、組織内の各種データを１ヵ所に集約する仕組みです。  イ　データエンティティは、データの管理対象となる実体を指します。E-Rモデルでは、長方形で実体(人や物、サービスなど)を表します。  ウ　データマートは、データウェアハウスに蓄積されたデータの中から、目的別にデータを抽出し、作成されたデータベースです。  システム監査　平成18年度春　問5　[出題頻度：★★☆]  解答－エ |

別冊演習ドリル 》 1-573,574

## 2. 分散データベース

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

分散データベースは、ネットワークに接続した複数のコンピュータがもっているデータベースを単一のデータベースとして利用する技術です。

### １）出現の背景 🏋プラスアルファ

データベースは、データを集中管理することで、システムの開発、保守、運用管理などのコストを削減することができますが、巨大化すると、費用や要員が増大するなどといった問題点が生じます。こうした問題点を解決するために、データベースを分散し、ネットワークで結ぶことで、複数のデータベースをあたかも一つのデータベースとして利用する分散データベースが生まれました。

### ２）透過性 🏋プラスアルファ

分散データベースが備える要件として、利用者が分散データベースであることを意識することなく利用できる環境（透過性）が必要です。そのため、ネットワークで結ばれた複数のデータベース間で、更新情報のやりとりを行うことで、データの整合性を保つ**レプリケーション**（複製）と呼ぶ作業が必要となります。

なお、透過性には次の６つがあります。

#### ①位置に対する透過性（location transparency）

データの物理的な配置を意識することなくデータを利用できる性質

#### ②移動に対する透過性（migration transparency）

データを移動しても、業務プログラムや操作手順を変更することなく利用できる性質

#### ③分割に対する透過性（fragmentation transparency）

１つの表のデータが、複数の場所に分割して格納されていても、意識することなく利用できる性質

#### ④複製に対する透過性（replication transparency）

１つの表が複数の場所に存在しても意識することなく利用できる性質

#### ⑤障害に対する透過性（failure transparency）

特定のデータベースに障害が発生しても、代替データベースが機能を代行することで、継続してサービスが利用できる性質

#### ⑥データモデルに対する透過性（data model transparency）

階層モデル、ネットワークモデル、リレーショナルモデルなど、データベースごとにデータモデルが異なっても意識することなく利用できる性質

|  |
| --- |
| 例題　プラスアルファ  分散データベースの透過性の説明として，適切なものはどれか。  ア　クライアントのアプリケーションプログラムは，複数のサーバ上のデータベースをアクセスする。アプリケーションプログラムは，データベースがあたかも一つのサーバ上で稼働しているかのようにアクセスできる。  イ　クライアントのアプリケーションプログラムは，複数のサーバ上のデータベースをアクセスする。アプリケーションプログラムはどのサーバ上のデータベースをアクセスするのかを知っている必要がある。  ウ　複数のクライアントのアプリケーションプログラムが，一つのサーバ上のデータベースを共有してアクセスする。  エ　複数のクライアントのアプリケーションプログラムは，一つのサーバ上のデータベースを，サーバ上のアプリケーションを介してアクセスする。  分散データベースは、遠隔地などに分散したデータベースをネットワークで接続することで、１つのデータベースシステムを利用しているように見せる仕組みです。  データベースが分散されていることをユーザが意識せずに利用できることを透過性と呼び、分散データベースの要件の１つです。  基本情報　平成21年度秋　問35　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-575～577

### ３）２相コミットメント

コミット処理は、メモリ上の更新情報を全てデータベースに反映させることです。分散データベースでは、分散した更新対象のデータベースに対して、完全に更新同期をとらなければ、全体としての一貫性を保つことができないので２相コミットメントがよく使われています。

２相コミットメントは、１回のコミットでは更新を確定せず、コミットの前に更新確定（コミット）も、更新無効（ロールバック）も、どちらとも可能な中間状態（セキュア状態）を設定して、一貫性を図る方式です。

次の図（１相コミットメント）のようにクライアントからのトランザクション処理をした場合、通常は処理を確定するためにコミット要求が発行され、これに基づきサイトＡとサイトＢに対してコミット処理が行われます。このとき、サイトＡでは正常終了し、サイトＢでは異常終了した場合、更新処理の整合性が失われ、データベースの内容が矛盾してしまいます。

コミット

更新

サイトＡ

更新

サイトＢ

(120)

＋20

100

サイトＡ

100

サイトＢ

障害

100

ロールバック

更新無効

(120)

＋20

更新確定

120

※（　）は更新確定前の状態を表す。

矛盾

１相コミットメント（例）

そこで、分散データベースではトランザクション処理の結果をすぐに確定するのではなく、コミットもロールバックもできる中間状態（セキュア状態）を設定し、その後にコミット処理を行うといった２段階の処理を行います。

100

サイトＡ

100

サイトＢ

障害

更新無効

更新無効

100

100

ロールバック

セキュア状態

(120)

？

OK

100

？

NG

矛盾

なし

サイトＡ

更新

サイトＢ

更新

(120)

＋20

＋20

(120)

※（　）は更新確定前の状態を表す。

２相コミットメント（例）

|  |
| --- |
| 例題  分散データベースシステムにおいて，一連のトランザクション処理を行う複数サイトに更新処理が確定可能かどうかを問い合わせ，全てのサイトが確定可能である場合，更新処理を確定する方式はどれか。  ア　２相コミット イ　排他制御  ウ　ロールバック エ　ロールフォワード  イ　排他制御とは、複数のプログラムが同一のデータベースを同時に更新しようとしても、データベース内の情報に矛盾が生じないようにするための制御機能のことです。  ウ　ロールバックとは、プログラム障害などの論理的な誤作動が発生したときに、障害発生時のデータベースの内容とログファイルの更新前のイメージを用いて、更新前の状態に戻す動作のことです。  エ　ロールフォワードとは、磁気ディスクの故障などの物理的な障害が発生したときに、バックアップファイルの内容を用いて処理前の状態に戻し、ログファイルの更新後イメージにより順次更新し、障害発生直前の状態に戻す動作のことです。  基本情報　平成29年度春　問28　[出題頻度：★☆☆]  解答－ア |

別冊演習ドリル 》 1-578,579

## 3. データ資源の管理

学習のポイント

✅ 出題頻度を参考に効率よく学習しよう！

情報資源の有効活用の観点から、現在、ユーザ各自が個々に利用しているデータを、組織として統括し、選別・管理する必要があります。データベースもデータ資源管理の対象となります。

🏋プラスアルファ

**●メタデータ**

メタデータは、データについて記述したデータです。データベース内の全てのスキーマの定義情報、スキーマに割り当てられている領域と現在使用されている領域の容量、作成日や更新日、作者、タイトル、要約、記述されている言語、データ構造などの各種情報がメタデータになります。なお、メタデータを登録したものをデータディクショナリと呼びます。

|  |
| --- |
| 例題　プラスアルファ  DBMSが管理するデータ，利用者，プログラムに関する情報，及びそれらの間の関係を保持するデータの集合体はどれか。  ア　データウェアハウス イ　データディクショナリ  ウ　データマート エ　データマイニング  ア　データウェアハウスは、情報分析と意思決定を支援する目的で、基幹業務システムからデータを抽出、再構成して構築されるデータベースです。  ウ　データマートは、利用目的に合わせてデータウェアハウスから作成されたデータベースです。  エ　データマイニングは、大量のデータを統計的、数学的手法で分析し、法則や因果関係を見つけ出す技術です。  基本情報　平成23年度春　問35　[出題頻度：★☆☆]  解答－イ |

別冊演習ドリル 》 1-580

🏋プラスアルファ

**●データディクショナリ／データディレクトリ**

メタデータを登録したものをデータディクショナリと呼ぶのに対して、テーブル名、データ型、列名、主キー、外部キーなどのデータベースの定義情報を登録したものをデータディレクトリと呼びます。データディクショナリ／ディレクトリを用いれば、分散データベースにおいて、データベースの位置を、利用者が意識する必要がなくなります（位置に対する透過性）。

🏋プラスアルファ

**●リポジトリ**

リポジトリとは、元々は倉庫、金庫などの意味で、一般的には小規模なデータベースという意味で使われますが、具体的に何を表現するかは場合によって異なります。

例えば、アプリケーション開発で用いられる場合には、開発にかかわる定義情報、設計情報、プログラム、テスト結果などの設計情報を格納するデータベースを意味します。また、データウェアハウスでは、メタデータを格納するデータベースをリポジトリと呼びます。

|  |
| --- |
| 例題　プラスアルファ  リポジトリの説明として，適切なものはどれか。  ア　データの更新履歴を保持し，CPU停止などのシステム障害時にデータベース回復のために利用される。  イ　データの定義情報（メタデータ）を保持し，データベースのデータディクショナリ又はデータディレクトリとして利用される。  ウ　データベースの物理的なコピーを保持し，ディスク障害時のデータベース回復に利用される。  エ　ユーザ識別情報やアクセス制限情報を保持し，データベースのセキュリティ管理に利用される。  ア　ジャーナル（ログ）ファイルに関する記述です。  ウ　バックアップファイルに関する記述です。  エ　ACL（Access Control List）に関する記述です。  初級システムアドミニストレータ　平成17年度秋　問20　[出題頻度：★☆☆]  解答－イ |