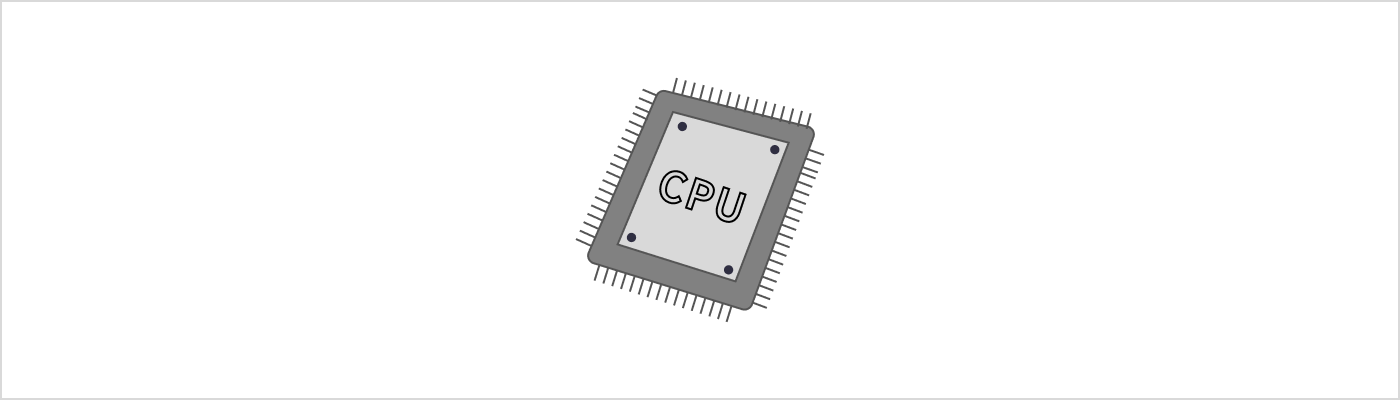
**コンピュータの種類と大きさ**

技術の進歩に伴い、コンピュータを区別する基準が曖昧になっている、そのためコンピュータの種類を区別する基準は**大きさ**と**処理能力**しかない

ITパスポート試験に出題されやすい種類はブレードサーバ

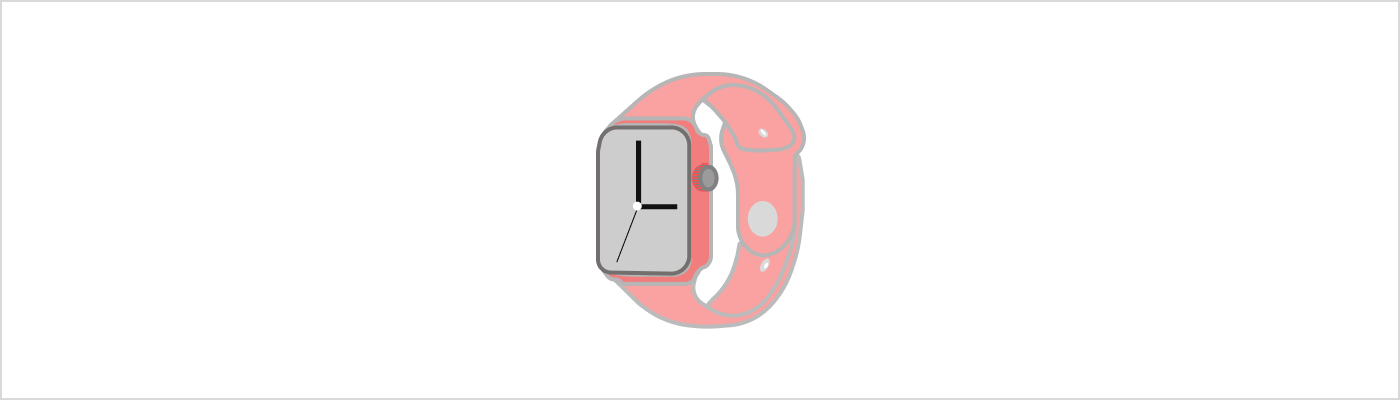
**マイクロコンピュータ**

CPUや主記憶装置を半導体チップに格納した超小型のコンピュータ



**ウェアラブル端末**

身に着けて使うコンピュータ、例えば時計型やメガネ型などの形状がある



**携帯情報端末**

手のひらに収まり、簡単に持ち運びができるコンピュータ

例えばスマホとか



**タブレット端末**

タッチパネルのディスプレイを持つ携帯情報端末



**PC**

持ち運びが可能な個人向けにコンピュータ

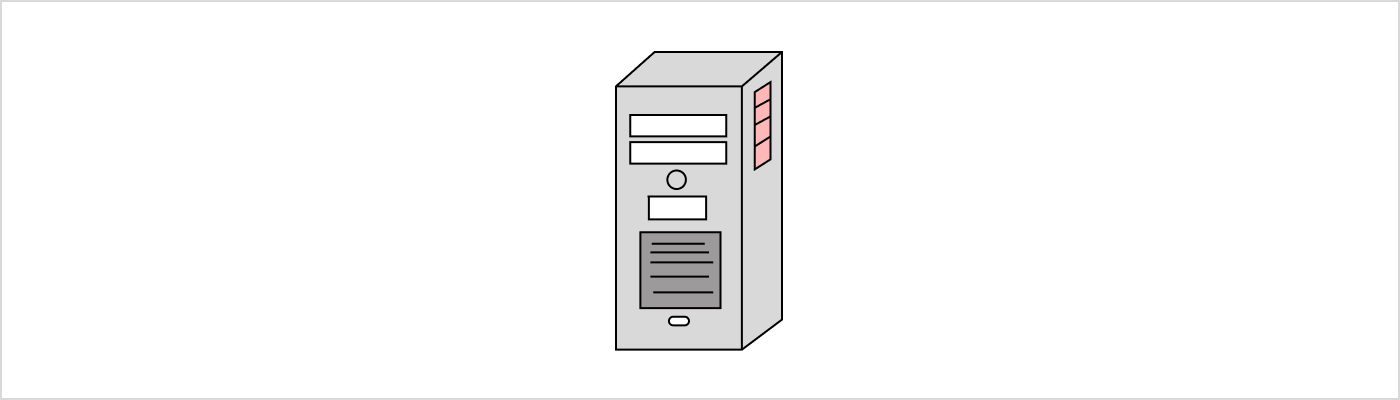


**サーバ**

ユーザからの要求に応じて処理を行うコンピュータ

例えば、メールサーバ、ウェブサーバなど

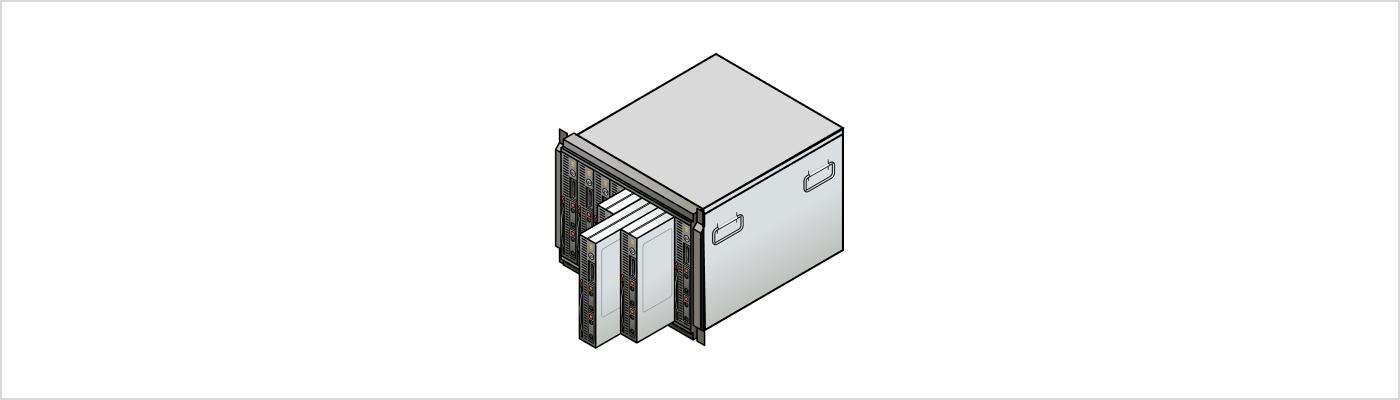
大人が一人でやっと持ち上げられるくらいの大きさと重さ



**ブレードサーバ**

CPU、主記憶装置、ハードディスクなどの１枚の基盤（ブレード）に組み込み

複数の基盤をまとめて専用のラックに収納したコンピュータ、高密度や省スペースの実現が可能



**汎用コンピュータ**

基幹業務を処理するために利用されるコンピュータ、基幹業務とは

企業の根幹を担う業務のこと、例えば空港の管制塔システムに使われる

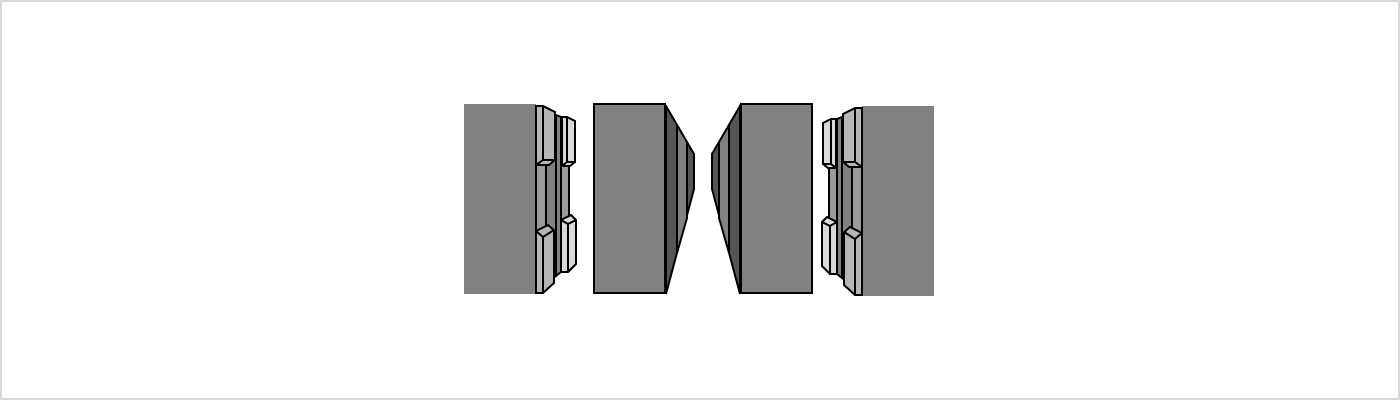
「メインフレーム」とも呼ばれる汎用コンピュータは大人でも持ち上げることができない



**スーパーコンピュータ**

大量の計算を高速に行うコンピュータ

例えば地球規模の気象予測に使われる、日本では「京（けい）」「富岳（ふがく）」が有名



**コンピュータの５つの役割**

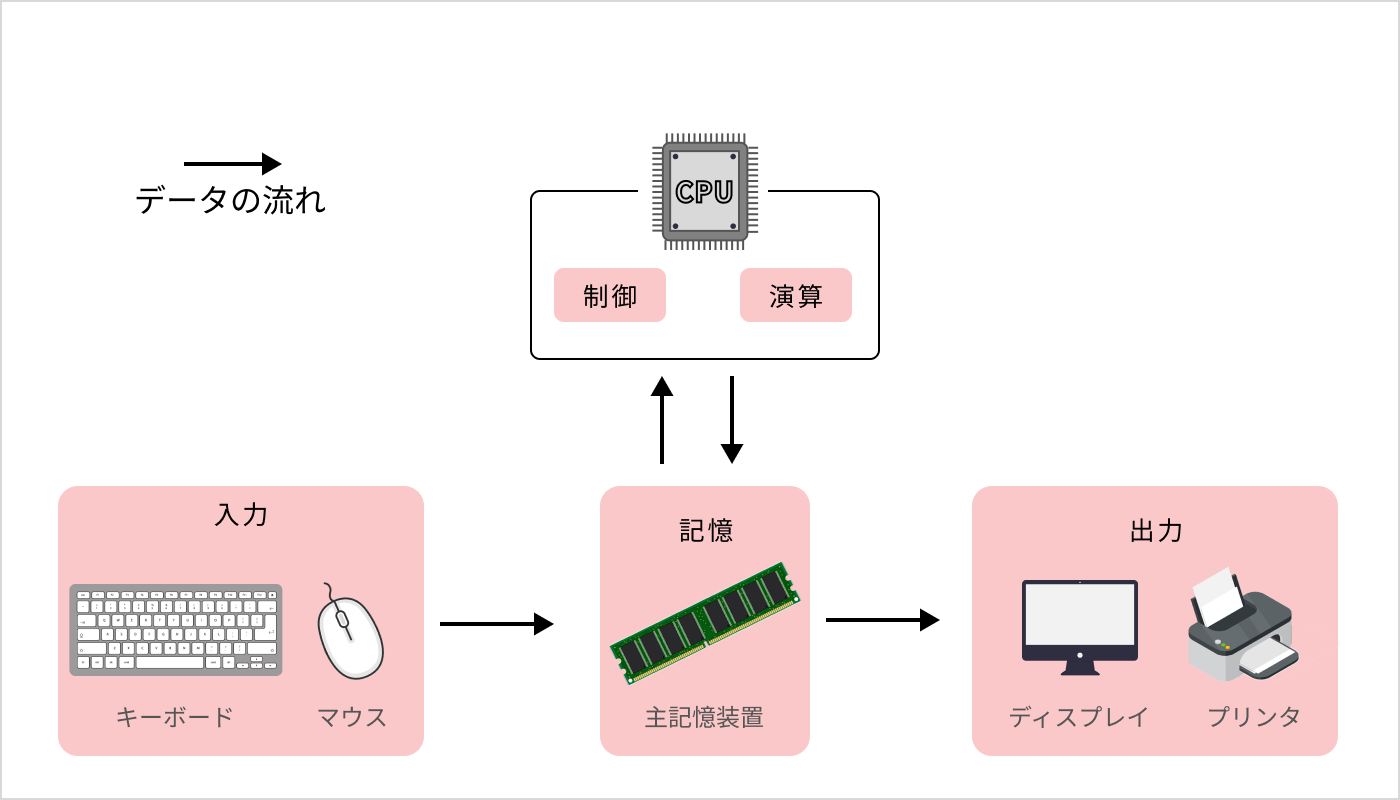
「人間が情報を処理する手順」と「コンピュータが情報を処理する手順」は似ている

・人間は、目や耳から情報を入手し、脳で考え、考えた結果を話す

・コンピュータは、キーボードで情報を入力、CPUで演算、ディスプレイに出力

**コンピュータの部品や機器の役割**

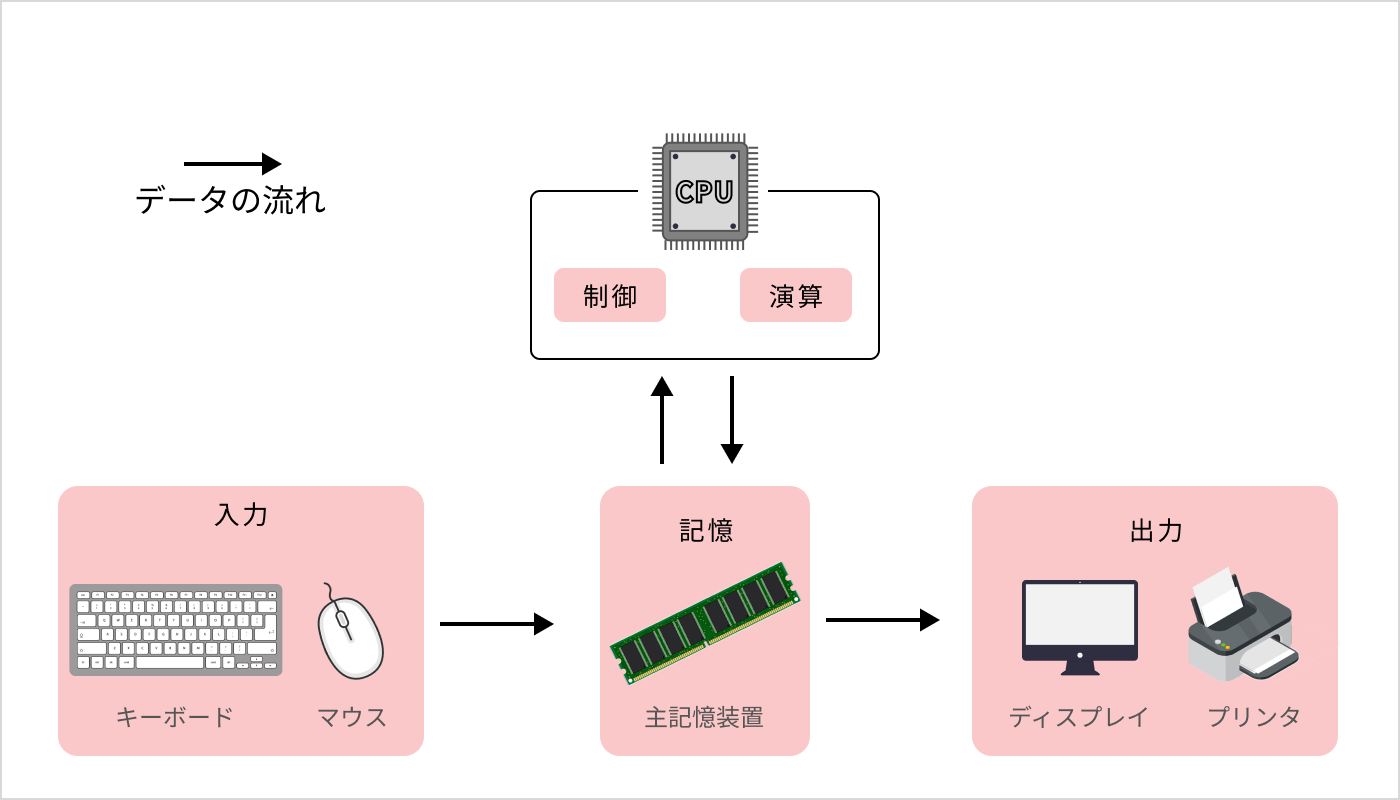
コンピュータには様々な部品や機器があり、それぞれに役割がある、その役割は「演算」「制御」「記憶」「入力」「出力」の５つに分けられる



| **演算** | **データの計算をする**。コンピュータが最も基本的な役割。 コンピュータでは「**CPU（中央処理装置）**」が担っている 人間に例えると「脳」 |
| --- | --- |
| **制御** | 他の役割（演算、記憶、入力、出力）を担う装置を**コントロール（制御）する**。司令塔のような役割。 コンピュータでは「**CPU**」が担っている 人間に例えると「脳」 |
| **記憶** | **データを覚えておくこと。** コンピュータでは「主記憶装置」「SSD」「HDD」などが担っている 人間に例えると「脳」 |
| **入力** | **データを取り入れること** コンピュータでは「キーボード」や「マウス」「マイク」などが担っている 人間に例えると「目」「耳」「鼻」「口」「皮膚」の五感 |
| **出力** | **データを表示したり送信したりこと** コンピュータでは「プリンタ」や「ディスプレイ」などが担っている 人間に例えると「口」 |

**演算と制御**

「演算」と「制御」は「CPU」という装置が担っている



**演算と制御とCPU**

CPUはコンピュータの中心となる機器、プロセッサともいう、人間に例えると「脳」になる

ＣＰＵの性能を表す指標は、次の３つ

・クロック周波数

・コア

・レジスタのデータサイズ

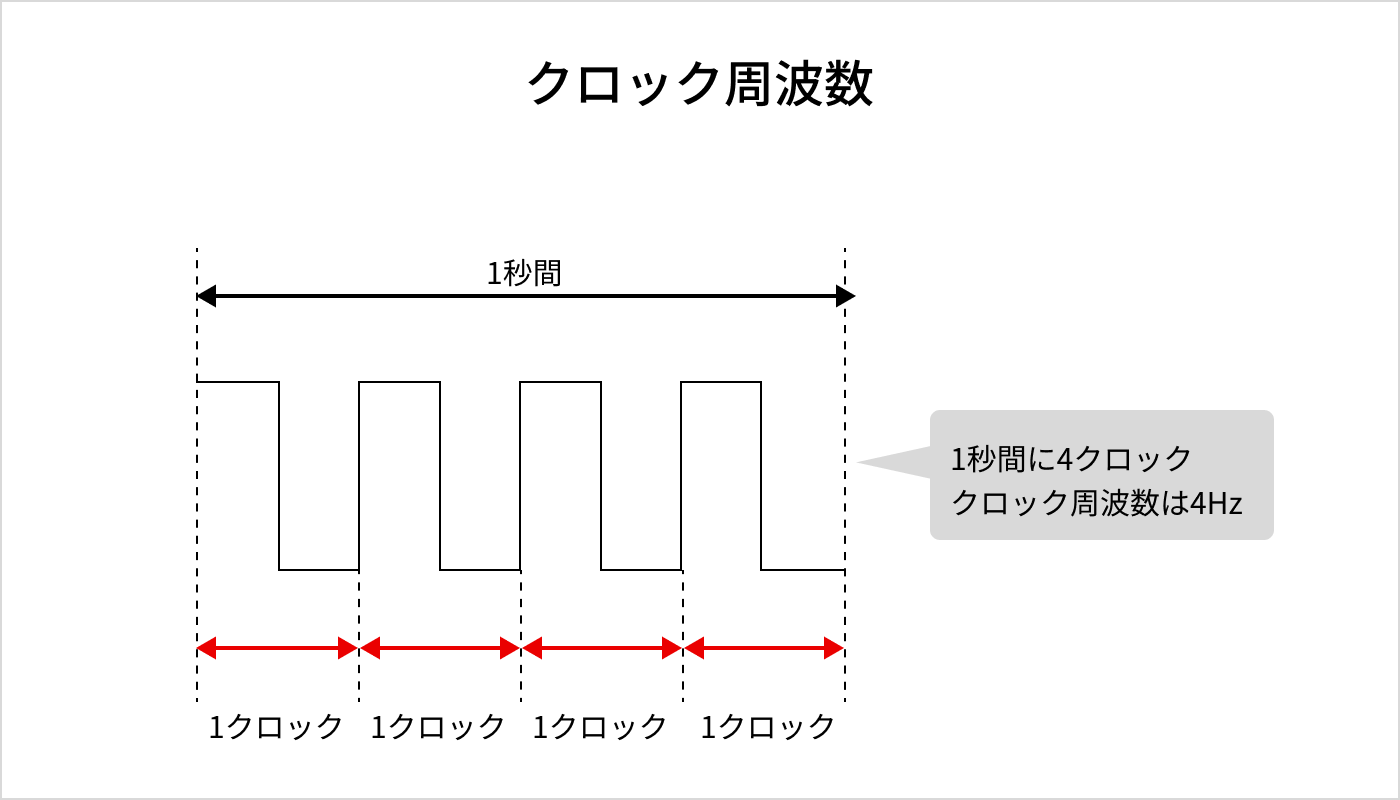
**クロック周波数**

コンピュータで行われるあらゆる処理は、ＣＰＵによって行われるため、ＣＰＵの速度が重要になる

クロック周波数とは、ＣＰＵの動作速度を表す指標であり、クロックが１秒間に発生する回数のこと

クロックとは、ＣＰＵや周辺機器などの動作を同期させるために、定期的に繰り返される信号のこと

クロック周波数の単位「Ｈｚ　ヘルツ」で表現され、１秒間に１回のクロックを発生させること「１Ｈｚ」と表す



例えば、クロック周波数が４HzのCPUがあった場合、１秒間に発生するクロック数は「４クロック」になる

クロック周波数が高いほどCPUの動作速度が速くなる、しかしクロック周波数が高いほど消費電力が増えるというデメリットも存在する

**コア**

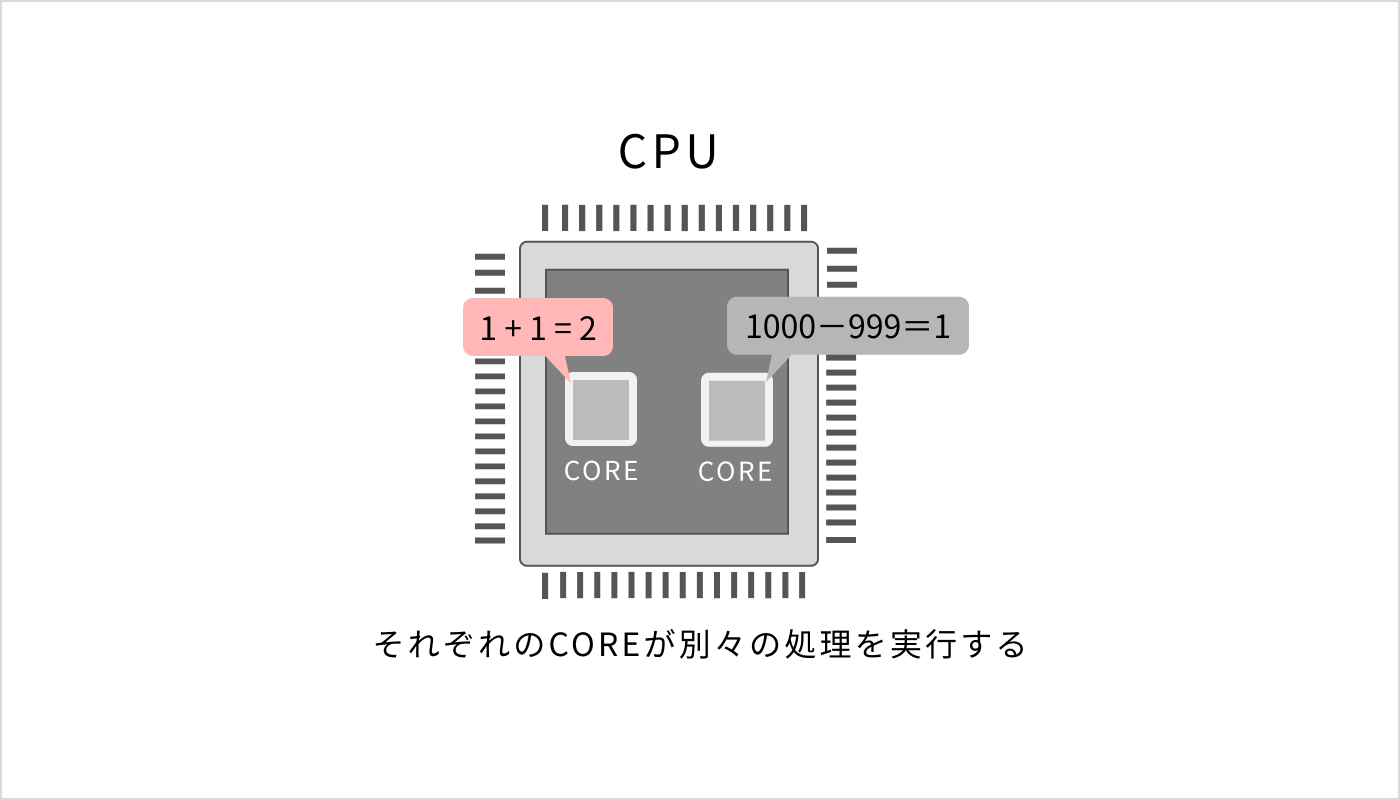
CPU内にある「演算処理を行う装置」、現在は１つのCPUに複数のコアが搭載されているコンピュータが主流

複数のコアが入っているCPUのことを「マルチコアプロセッサ」という

・２つのコアを持つCPUは「デュアルコアプロセッサ」

・４つのコアを持つCPUは「クアッドコアプロセッサ」

マルチコアプロセッサは、複数のコアそれぞれが別々の処理を実行できるため、多くのコアが入っているCPUの方が性能が高いといえます



**レジスタのデータサイズ**

レジスタとはCPU内にある「データを一時的に記憶する装置」

コンピュータは基本的に手記記憶装置にデータを保存するが、CPU内部にもデータを記憶するレジスタがある

レジスタのサイズには「32ビット」と「６４ビット」の２種類がある

・32ビットのレジスタを持つCPUを「**32ビットCPU**」  
・64ビットのレジスタを持つCPUを「**64ビットCPU**」  
64ビットCPUのほうが、一度に記憶できるデータの量が多いので、より高速に処理できる。

まとめ

・コンピュータの種類は大きさで分類する  
・コンピュータの役割には「演算」「制御」「記憶」「入力」「出力」の５つがある

**ブレードサーバ**  
CPU、主記憶装置、ハードディスクなどを１枚の基盤（ブレード）に組み込み、複数の基盤をまとめて専用のラックに収納したコンピュータ  
高密度や省スペースの実現が可能  
  
**演算**  
データの計算をする、コンピュータが最も基本的な役割、コンピュータでは「CPU（中央処理装置）」が担っている人間に例えると「脳」  
  
**制御**  
他の役割（演算、記憶、入力、出力）を担う装置をコントロールする司令塔のような役割、コンピュータではCPUが担っている

**記憶**  
データを覚えておくこと、コンピュータでは「主記憶装置」「ハードディスク」「SSD」「HDD」などが担っている

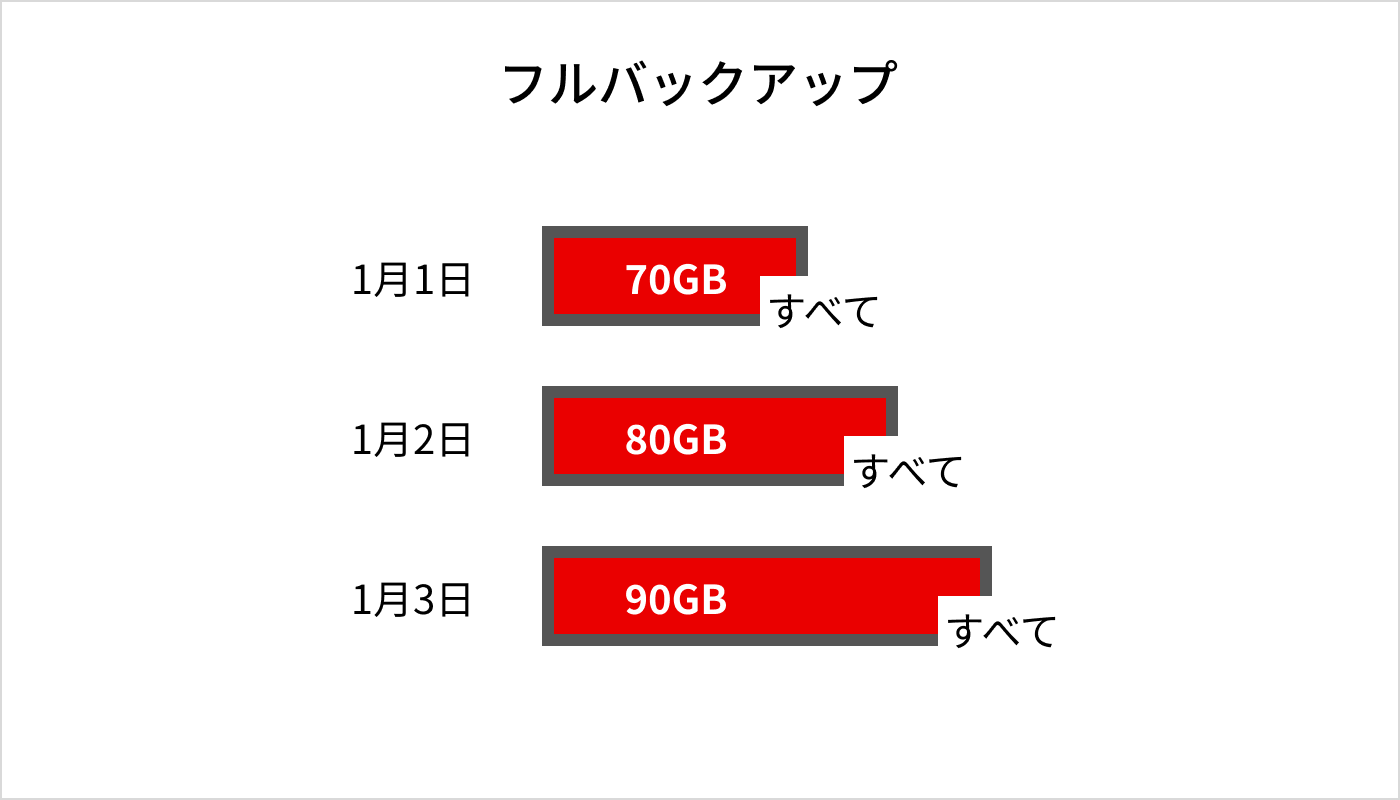
**入力**  
データを取り入れていること、コンピュータでは「キーボード」「マウス」「マイク」などが担っている

**出力**  
データを表示したり送信したりすること、コンピュータでは「プリンタ」「ディスプレイ」などが担っている

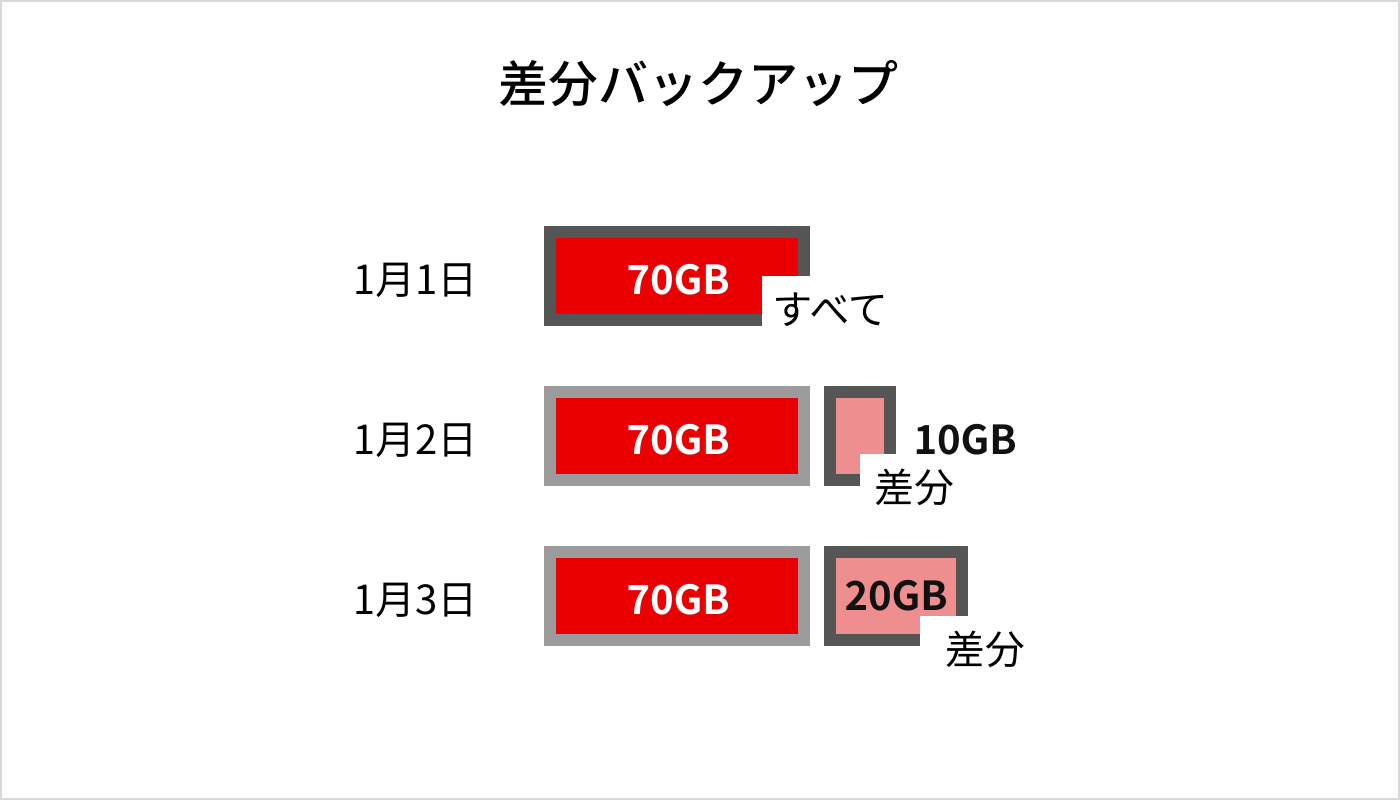
**クロック周波数**  
ＣＰＵが動作速度を示す指標

**コア**  
演算処理を行う装置

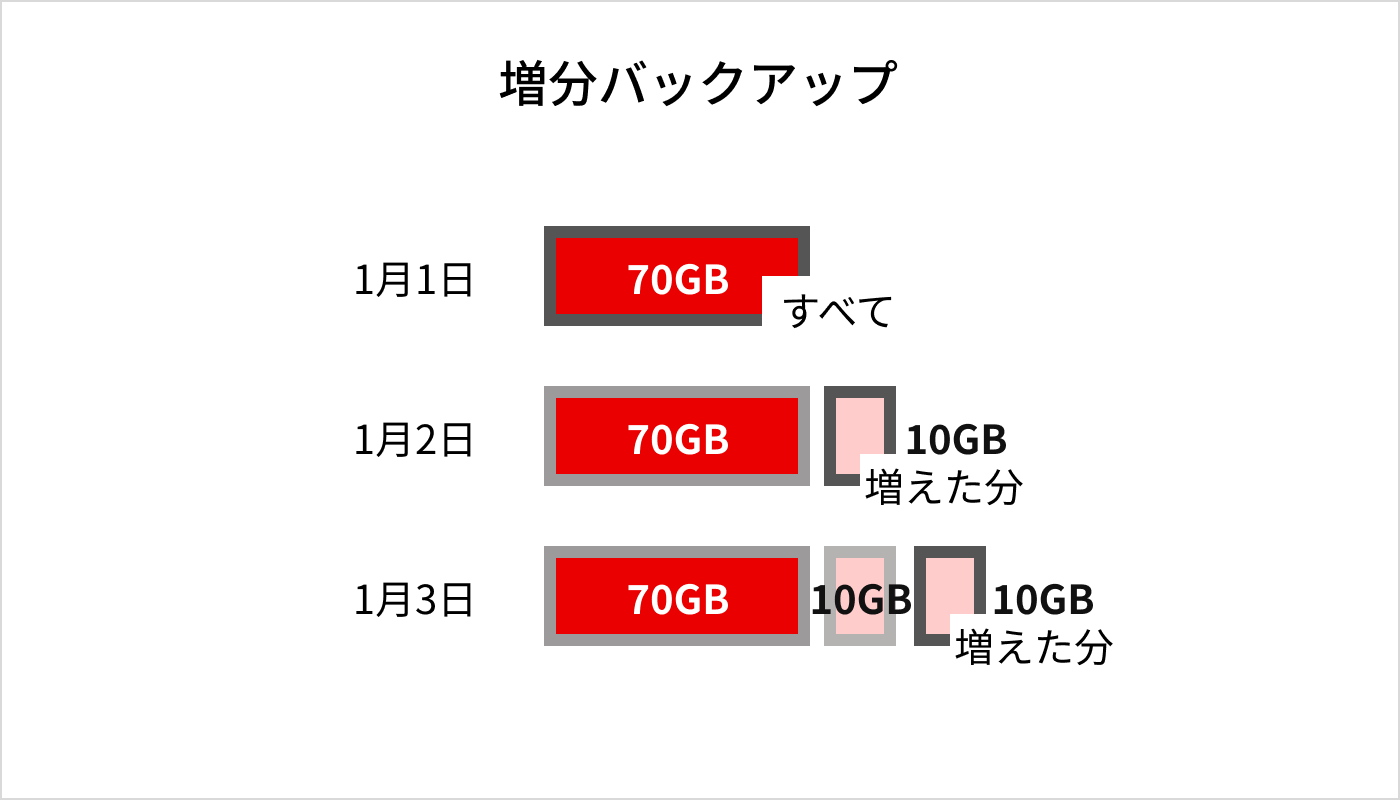
**レジスタ**  
ＣＰＵ内にある「データを一時的に記憶する装置」

バックアップ  
普段利用している補助記憶装置とは別の補助記憶装置にデータを複製すること、補助記憶装置とはハードディスク、USBなど  
  
フルバックアップ  
すべてのデータをバックアップすること、特徴は復旧作業が簡単な点  
しかし、毎回すべてのデータをバックアップする必要があるため、データが多くなるとバックアップファイルの作成に時間がかかる  
システム障害などが起きた場合には、最新のバックアップデータを使うことでシステムを元の状態に戻せる  


差分バックアップ  
フルバック後に、追加されたデータのみをバックアップすること  
差分バックアップで対象とするデータは、フルバックアップからの追加分、変更分のデータだけなので、バックアップ処理にかかる時間はフルバックアップより速い  
しかし「フルバックアップファイル」と「差分バックアップファイル」の２つのファイルが必要となるため、復旧作業が複雑になる  
例えば1/3のデータを復元する場合は、1/1のフルバックアップのデータと1/3の差分バックアップデータが必要



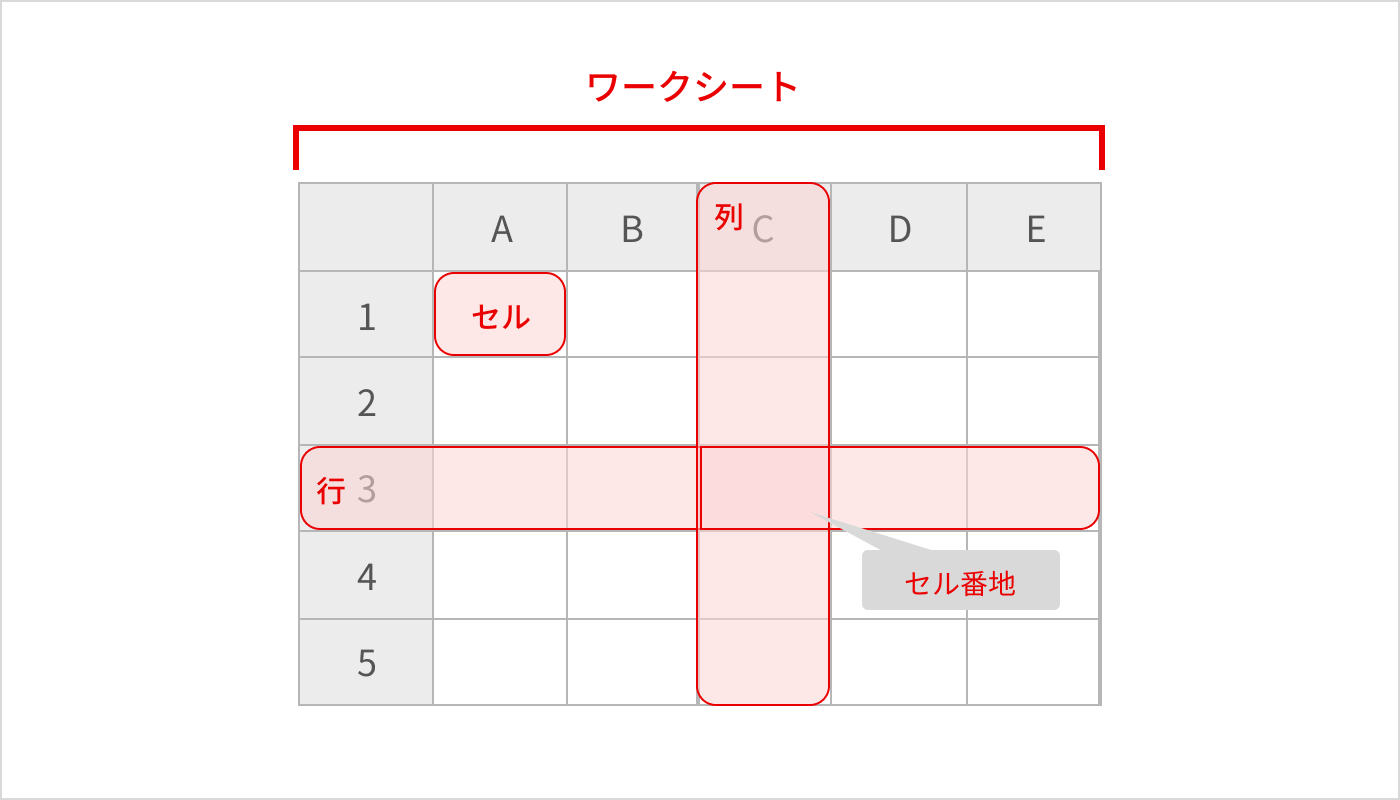
増分バックアップ  
前回のバックアップ後に、追加されたデータのみをバックアップすること  
差分バックアップよりも、バックアップ処理にかかる時間が短いのが特徴  
ただし、バックアップファイルが複数になるため、復旧作業が差分バックアップよりもさらに複雑になってしまう  
例えば、1/3のデータを復元する場合は「1/1のフルバックアップデータ」と「1/2の増えた分のデータ」と「1/3の増えた分のデータ」が必要になる



フルバックアップ  
すべてのデータをバックアップすること

差分バックアップ  
フルバックアップ後に追加されたデータのみをバックアップすること

増分バックアップ  
前回のバックアップ後に、追加されたデータのみをバックアップすること

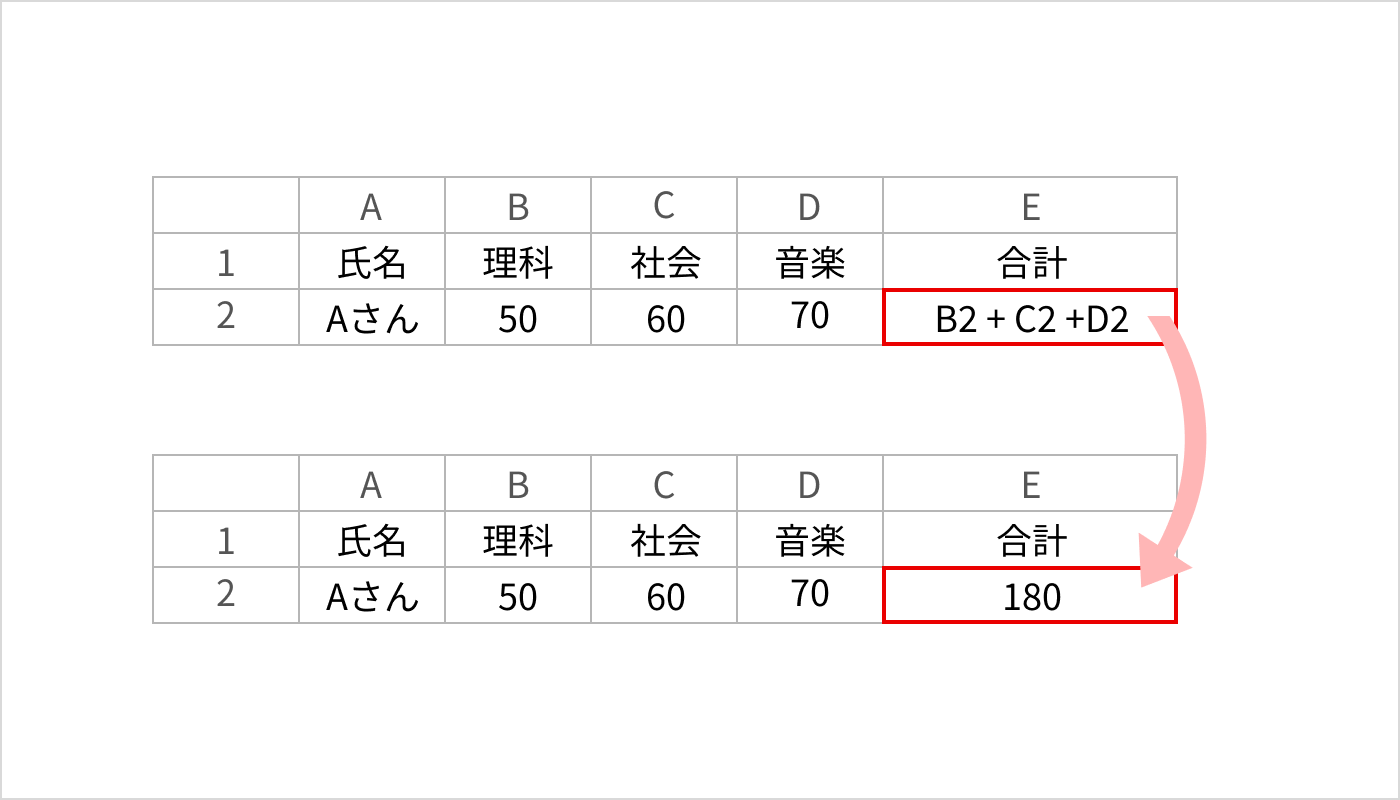
アプリケーションソフトウェア  
利用者に用途や目的など特定の機能を提供するソフトウェアのこと  
例えば、文章作成ソフトや表計算ソフトなどになる  
ＯＳ上で稼働し、目的に応じた作業を行うことが可能  
  
表計算ソフト  
表に入力された数値の計算を行うソフトウェア、数値以外にもテキストを入力できる  
例えば、ExcelやGoogleスプレッドシートなど  


表計算ソフトに関する重要な用語  
ワークシート　作業スペース全体のこと  
セル　１つ１つのマス目のこと、ワークシートの大きさはセルの増減を変更できる  
列　縦方向のセルに並び、各列にアルファベットが割り振られる  
行　横方向のセルの並び、番号が割り振られている  
セル番地　セルの位置を列と行で表した例えば、上図のセル番地は「Ｃ３」と書く

セルには「**180**」のように固定の数値を入力することもできます。  
また、「**B2+C2+D2** 」とセル番地を使った**計算式**の入力もできます。

人が計算して入力するより、計算式を入力するほうが、より速く、より正確な値を計算可能です。

「**表**」を使って「**計算**」するのが表計算ソフトです。



PCのキーボードには、「×」や「÷」の記号がありません。表計算ソフトで四則演算を行うときは、次の記号を使います。

| **四則演算の記号** | **PCのキーボードの記号** |
| --- | --- |
| +（たし算） | + |
| -（ひき算） | - |
| ×（かけ算） | \* |
| ÷（わり算） | / |

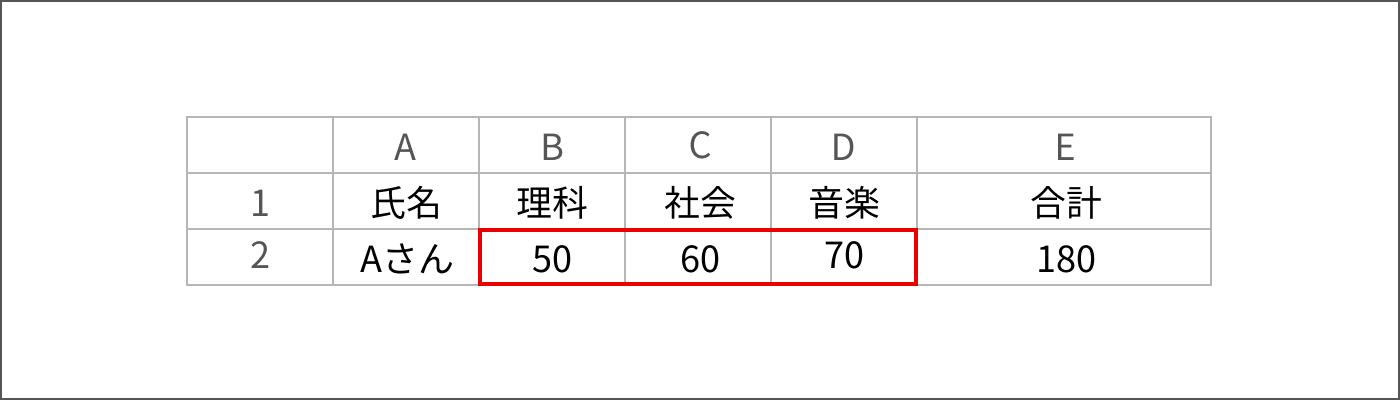
関数とは、**特定の計算処理を行い結果を表す機能**です。

関数の例は、以下のとおりです。

* 指定された範囲の値の合計値を求める「**合計関数**」
* 平均値を求める「**平均関数**」
* 最大値を求める「**最大値関数**」

関数を使うことで、複雑な計算式を短く、シンプルな式で表現できます。

例として、B2、C2、D2の合計を求めてみます。国語、数学、英語の合計値を求める計算式は「B2 + C2 + D2」となります。



3科目の点数の合計は、3つの点数の合計となります。

この場合は、まだ入力の数が少ないですが、多い場合には1つずつ足していくため計算式が長くなります。

計算式が長くなるほど、**関数**は便利です。関数を使って、ある範囲の値の合計値を求める場合は次のように入力します。

* **合計(セル範囲)**

セル範囲とは、**長方形の枠に含まれたセルの集まり**です。範囲の始まりと終わりは「:」（コロン）で指定します。

上記のB列からD列までの合計値をで求めると、次のようになります。

**合計(B2:D2)**

上記のようにセル範囲を指定すると、上図の赤枠内の合計値を求めることが可能です。

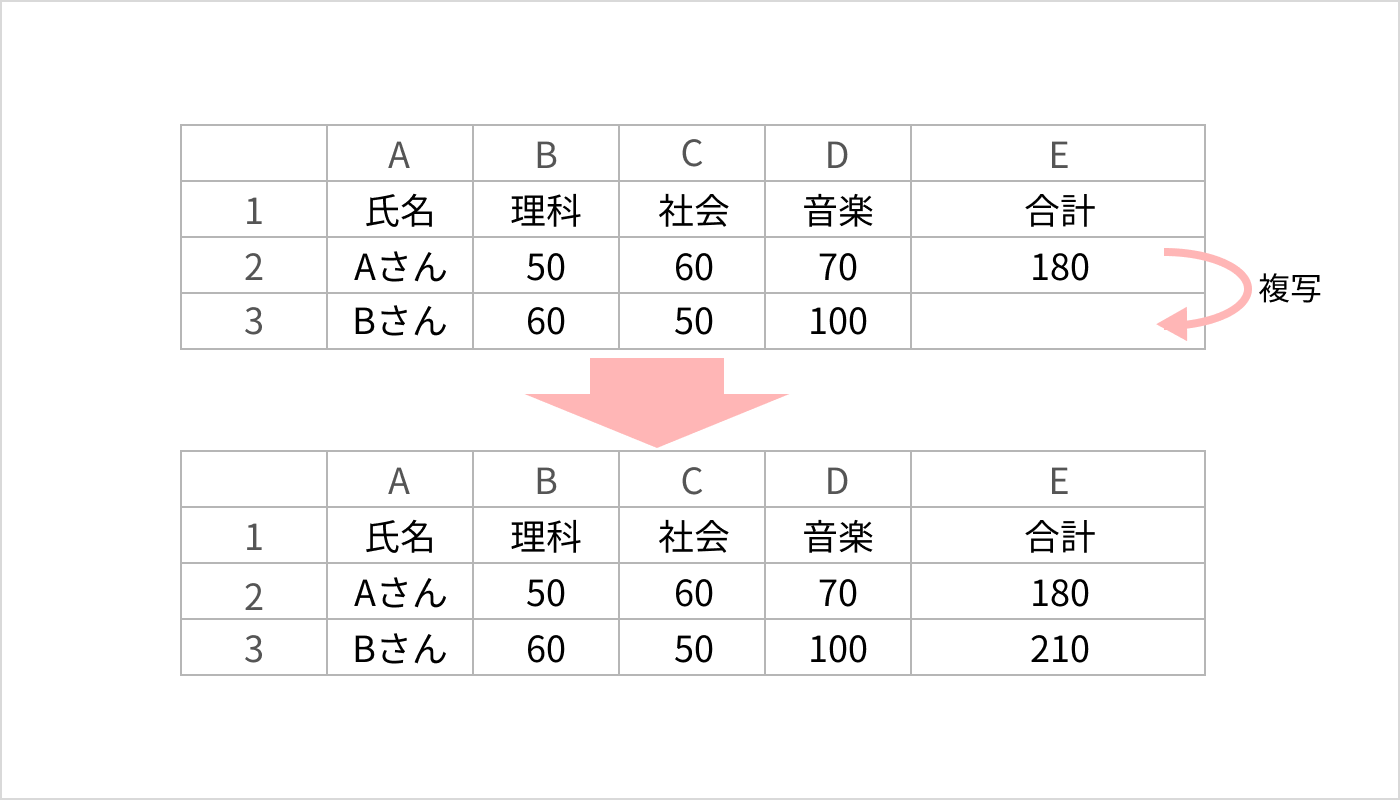
### **複写**

複写とは、セルの値をコピーすることです。

下の図でセルE2をセルE3へ複写すると、セルE3の値は「180」になります。

通常の複写の場合は「180」と期待されますが、表計算ソフトでは自動的に花子さんの合計値を求める計算式に変換されています。

つまり、花子さんの合計値を求める計算式は「合計(B3:D3)」です。



表計算ソフトでは、人の手間を省くように、自動的にセル番地を変更しています。

#### **絶対参照**

上記のように、自動的にセル番地を変更する機能はとても便利ですが、**セル番地を固定しておきたい場合**があります。

この場合は、セル番地の前に「**$**」（ドルマーク）をつけます。「$」は列名、行番号のどちらにもつけられます。

**列名の前につけると列名が固定でき、行番号の前につけると行番号が固定できます**。

「$」を使用し、セル番地を固定して参照する方法を「**絶対参照**」といいます。反対に、「$」を使わない参照方法を「**相対参照**」といいます。

それぞれの指定方法は、以下のとおりです。

| **指定方法** | **記載方法** |
| --- | --- |
| 列を固定する絶対参照 | $B2 |
| 行を固定する絶対参照 | B$2 |
| 列と行の両方を固定 | $B$2 |

| **バックアップ種類** | **説明** |
| --- | --- |
| フルバックアップ | **すべて**のデータをバックアップすること |
| 差分バックアップ | **フルバックアップ後**に、追加されたデータのみをバックアップすること |
| 増分バックアップ | **前回のバックアップ後**に、追加されたデータのみをバックアップすること |

| **表計算ソフトに関数る重要用語** | **説明** |
| --- | --- |
| ワークシート | 作業スペース全体のこと 例えば、Excelなどの1つのシートなど |
| セル | 1つひとつのマス目のこと ワークシートの大きさはセルの増減で自由に変更できる |
| 列 | 縦方向のセルの並び A、B、Cのように各列にアルファベットが割り振られている |
| 行 | 横方向のセルの並び 1、2、3のように各行に番号が割り振られている |
| セル番地 | セルの位置を列と行で表したもの 例えば、上図のセル番地は「C3」と書く |

| **「複写」に関する用語** | **説明** |
| --- | --- |
| 複写 | セルの値をコピーすること |
| 「$」（ドルマーク） | 列名、行番号の前に入力すると複写時に値を固定できる |
| 絶対参照 | 「$」を使用し、セル番地を固定して参照する方法を |

**情報デザイン**

**情報を視覚的に表現し、受け手にわかりやすいデザイン**のことを「情報デザイン」といいます。

私たちが普段使うものは、利用者がわかりやすい工夫がされています。

信号機を例に挙げます。私たちは「青は進む」「赤は止まる」という先入観があります。  
信号機の色は、この先入観に沿ったデザインになっているため、「進むか止まるか」で迷うことはありません。

情報デザインでは、「**ユニバーサルデザイン**」「**アクセシビリティ**」「**ユーザビリティ**」の3つが重要な用語

### **ユニバーサルデザイン**

ユニバーサルデザイン（Universal Design）とは、**すべての人が利用しやすい製品やサービスを設計するための理念や手法**のことです。

年齢、文化、言語、障害の有無など、能力の違いやさまざまなニーズを持つ人にとって、利用しやすい製品やサービスを提供することを目的としています。

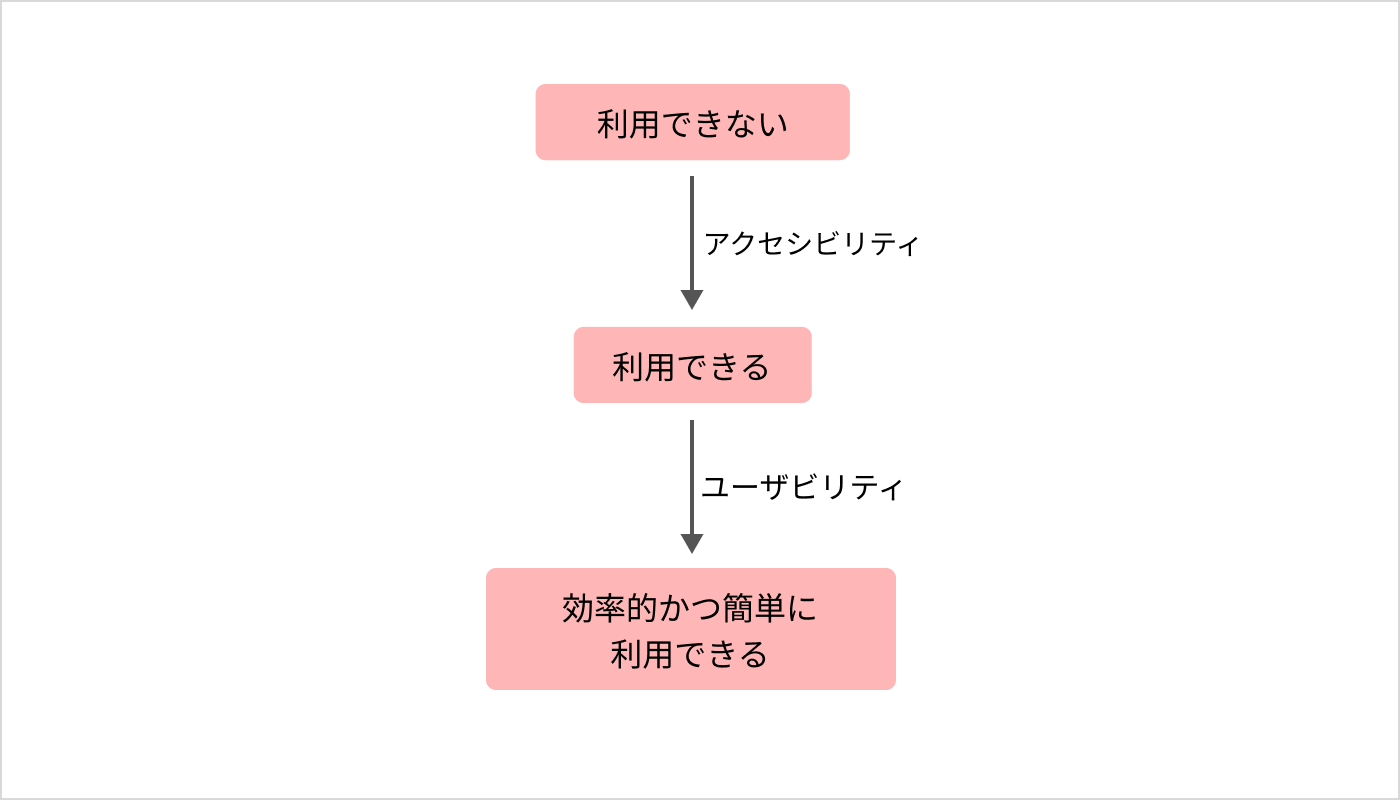
### **アクセシビリティとユーザビリティ**

アクセシビリティ（Accessibility）とユーザビリティ（usability）の意味と違いは、以下のとおりです。

| **用語** | **説明** | **例** |
| --- | --- | --- |
| アクセシビリティ | 身体的・知的・感覚的な障害を持つ人や高齢者を含む、**すべての人**が製品やサービスを利用できるようにすること | 視覚障害者が音声読み上げソフトを利用して、Webサイトを閲覧できるようにする |
| ユーザビリティ | 製品やサービスを利用する人が、**効率的かつ簡単に**利用できるようにすること | スマートフォンの画面が小さくて操作しづらい場合、画面の操作性を改善する |

アクセシビリティは、「**すべての人が製品やサービスを利用できるか**」がポイントです。

ユーザビリティは、「**効率的かつ簡単に利用できるか**」がポイントとなります。



**ソフトウェアの権利とオープンソフトウェア**

プログラムは著作権法で保護されています。

そのため、プログラムで作成されている**ソフトウェアも著作権法の保護対象**になります。また、通常のソフトウェアは改変、配布などについて制限があります。

**オープンソフトウェア**オープンソースソフトウェア（Open Source Software：OSS）とは、**ソースコードが公開されており、誰でも改変や再配布ができるソフトウェア**のことです。OSSと呼ばれます。

改変とは、ソースコードを自由にカスタマイズすることを指します。再配布とは、無料で入手したOSSを他の人に再び配布することです。

OSSのポイントは、4つあります。

1. 具体例
2. 改変
3. 販売
4. 著作権

#### **具体例**

| **種類** | **OSSの名称** |
| --- | --- |
| OS | Linux（リナックス）、Android（アンドロイド） |
| Webサーバ | Apache（アパッチ）、Nginx（エンジンエックス） |
| メールソフト | Thunderbird（サンダーバード） |
| Webブラウザ | Firefox（ファイヤーフォックス）、chromium（クロミウム） |
| データベース管理システム（DBMS） | MySQL（マイエスキューエル）、SQlite（エスキューライト） |

#### **改変**

ポイント2つ目は、OSSの「改変可能」な点です。

繰り返しになりますが、OSSは**ソースコードが公開されており、誰でも改変できます**。

通常のソフトウェアは、ソースコードが公開されていません。そのため、ソフトウェアを改変することも不可能です。この点が、ソフトウェアとOSSの違いになります。

#### **販売**

ポイント3つ目は、OSSの「販売可能」な点です。

OSSは、無料で入手できます。入手したOSSに、新しい機能やサービスを追加し、有料で**販売**することが可能です。

#### **著作権**

ポイント4つ目は、OSSの「著作権あり」な点です。

OSSは、「ソースコードも公開してるためフリーのソフトウェアではないか」と勘違いされやすいです。しかし、**OSSにも著作権はあります**。

ソフトウェアとOSSの違いは、以下のとおりです。

| **項目** | **OSS** | **ソフトウェア** |
| --- | --- | --- |
| 費用 | 無料 | 有料 |
| 改変 | 可 | 不可 |
| 再配布 | 可 | 不可 |
| 転売 | 可 | 不可 |
| 著作権 | 製作者 | 製作者 |

**データベース**

コンピュータシステムでは、膨大なデータを扱うことが珍しくない、例えば学校に所属している生徒の名前や住所などのデータ

これらの情報を整理した集まりが「データベース」という

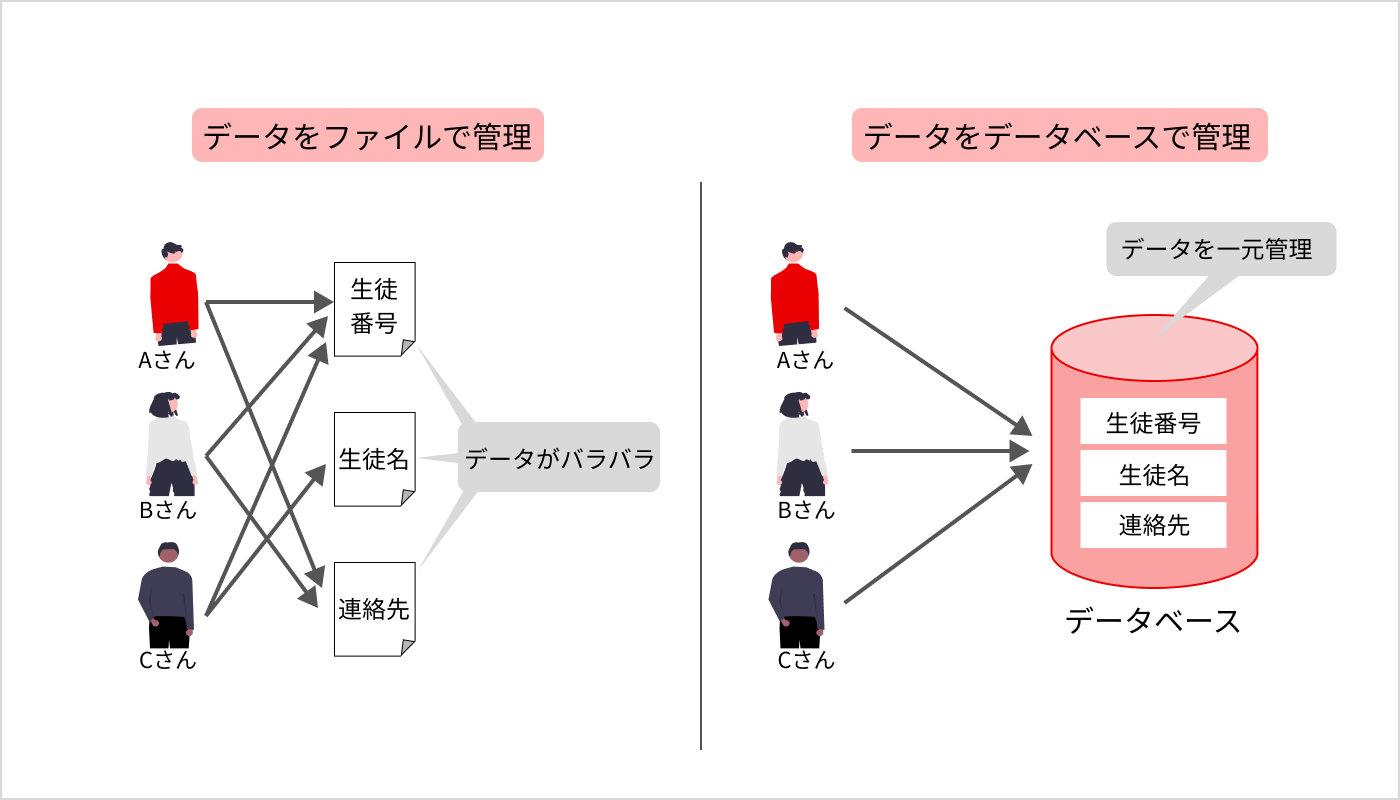
**データベースの基本**

DBとは整理された情報の集まり

データをDBに集めるｋとおで情報を効率的に保存、管理できる

データはファイルで管理することも可能、ただし、データの量が膨大になってくると、必要な情報を取り出すだけでも労力と時間がかかる

データをＤＢで管理すれば、必要データの検索や変更などが容易にできる



**関係データベース**

関係データベース（Relational Database）とは、データを表形式で管理するデータベース

関係データベースの表の例は、以下のとおり



表は「行（レコード）」「列（属性）」で構成されている



上図は「３つの行」と「３つの列」がある表、行はデータ数、列はデータの属性を表したもの

データの属性は「生徒番号」「生徒名」「連絡先」にあたる、データの属性には名称があるのが特徴

また、データ数を追加すると１行増え、データを削除すると１行削除される

**フィールド**

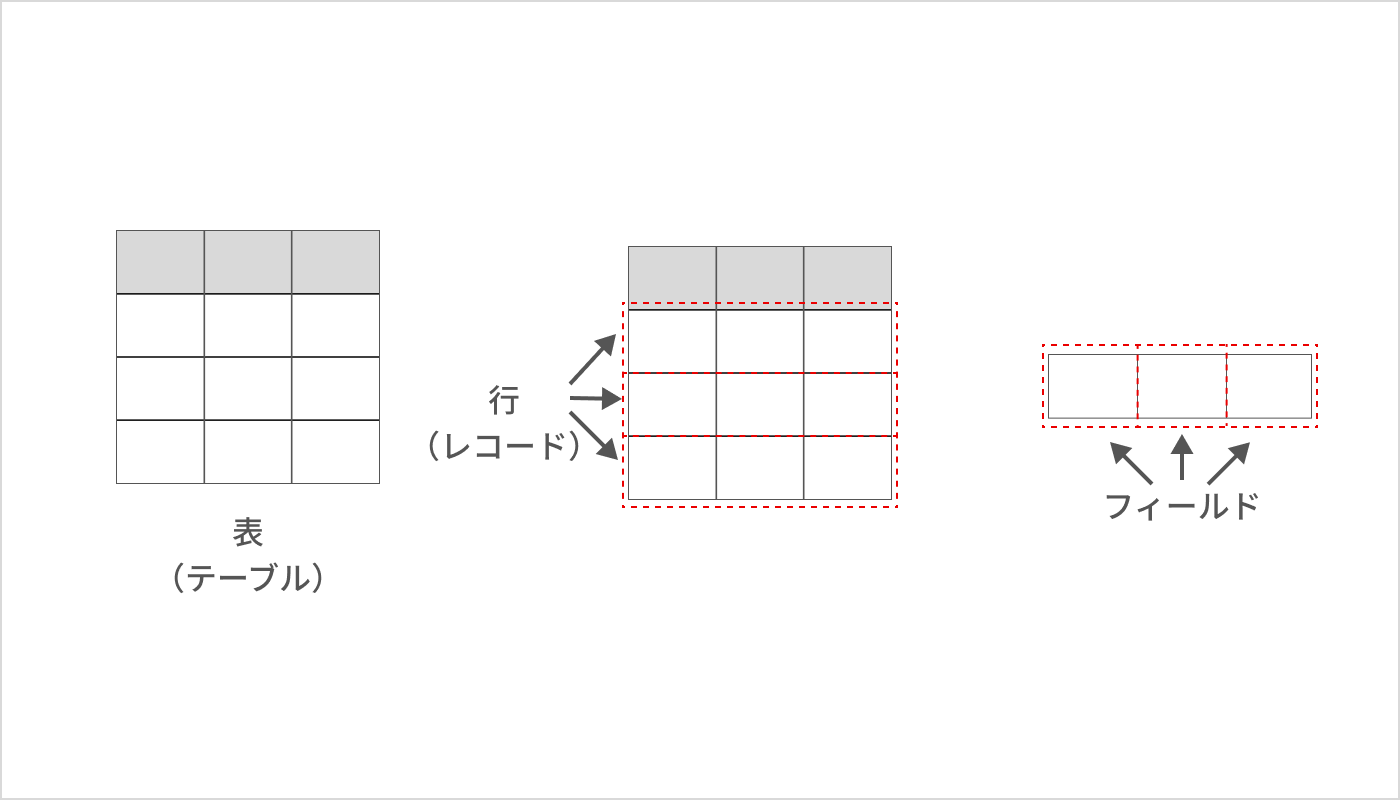
フィールドとは、行（レコード）を構成する１つの要素のこと

表（テーブル）行（レコード）、フィールドの関係は以下のとおり

・データベースは「表（テーブル）」で構成

・表は複数の「行（レコード）」で構成

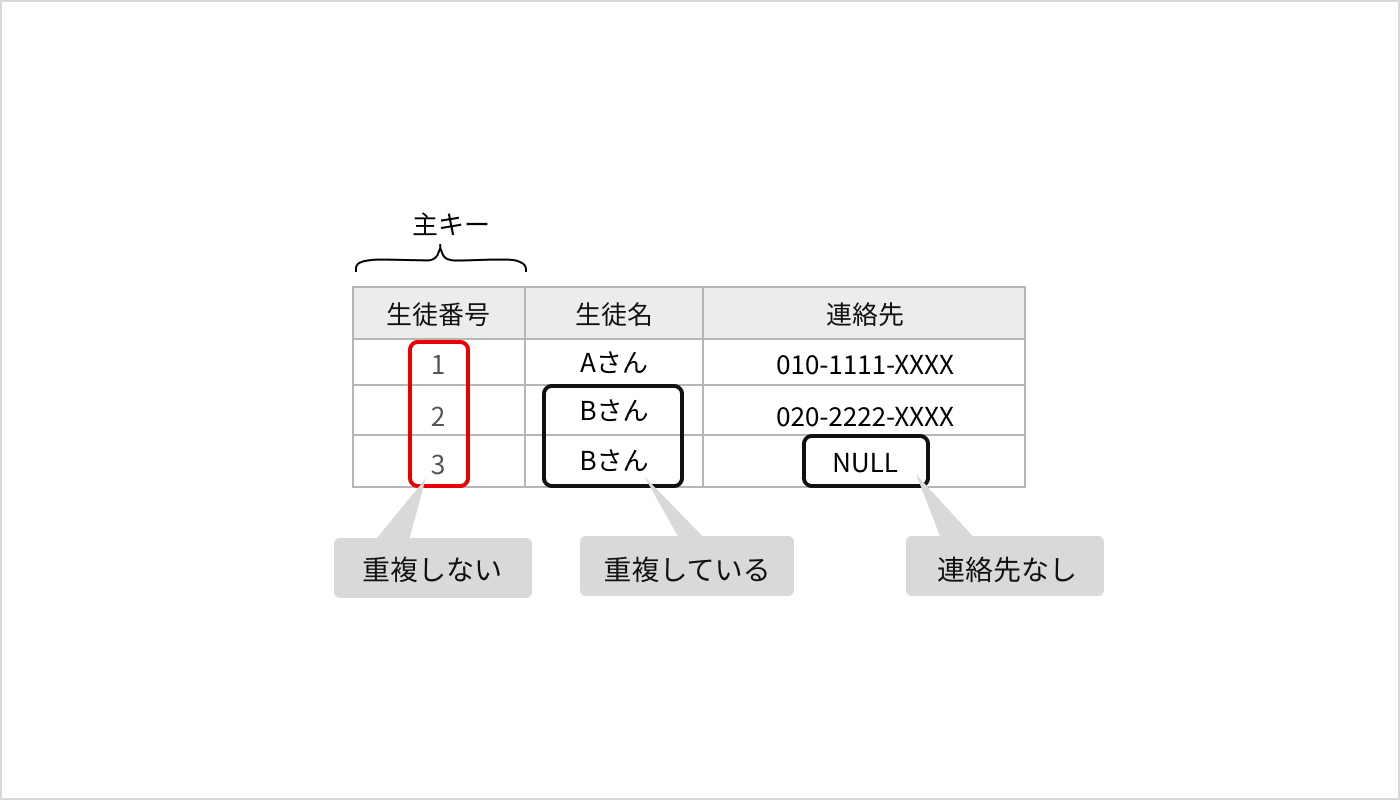
・行は複数の「フィールド」で構成



**主キー**

表の中から１つの行を特定するための列、簡単に言うと、他の行と内容が重ならない、固有の値となっている「列」を指す

主キーは、関係データベース設計において非常に重要な役割を果たす、主キーを使用することで、データの一意性を確保し、データの整合性を維持できる



上図のように、主キーの特徴は以下のとおり

・主キーの値は重複しないこと

・主キーNULL（データなし）ではないこと

NULLとはデータがないこと

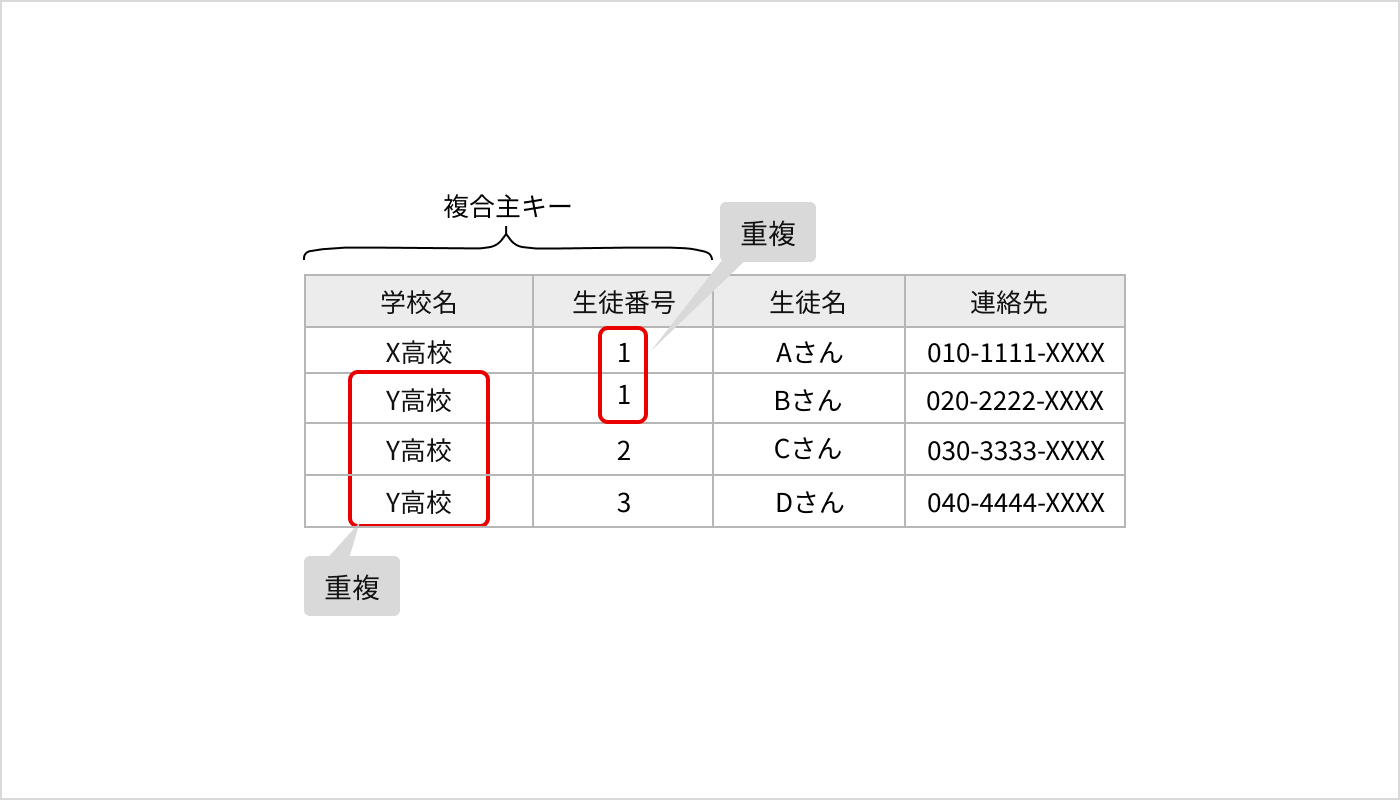
特徴に一致するのが「生徒番号」であるため、主キーは「生徒番号」になる、また、主キーは表（テーブル）の最初の列に配置されることが多い

また、主キーは「１つの列」に限らず、「２つの列」で主キーとなる場合もある

**複合主キー**

複数の列を組み合わせた主キーのこと、１つの列で主キーの条件を満たす列がない場合、複合主キーを使用する

下図は１つの列で主キーの条件を満たす列がないため、「学校名」と「生徒番号」を組み合わせて複合主キーにしたデータベース



**外部キー**

関連した表同士をひもづけるために設定する列のこと

下図のように関連した「生徒表」と「教師表」がある場合、生徒表の「教師番号」が外部キーになる



上図のように、外部キーは関連した別の表の主キーを参照する

このように、関係データベースはデータの種類ごとに表を分け、データを管理する、これにより、データの追加や変更が簡単にできる

**データベース設計**

データベースでは、膨大なデータを管理する

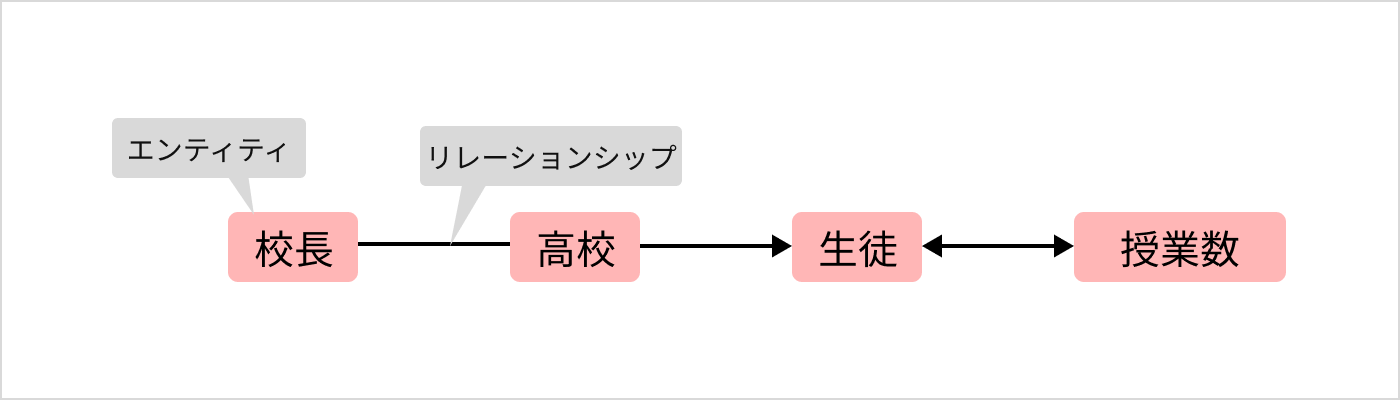
データの構造が複雑になったり重複していたりすると、データの追加や編集に時間がかかったり、更新漏れが発生したりする可能性がある

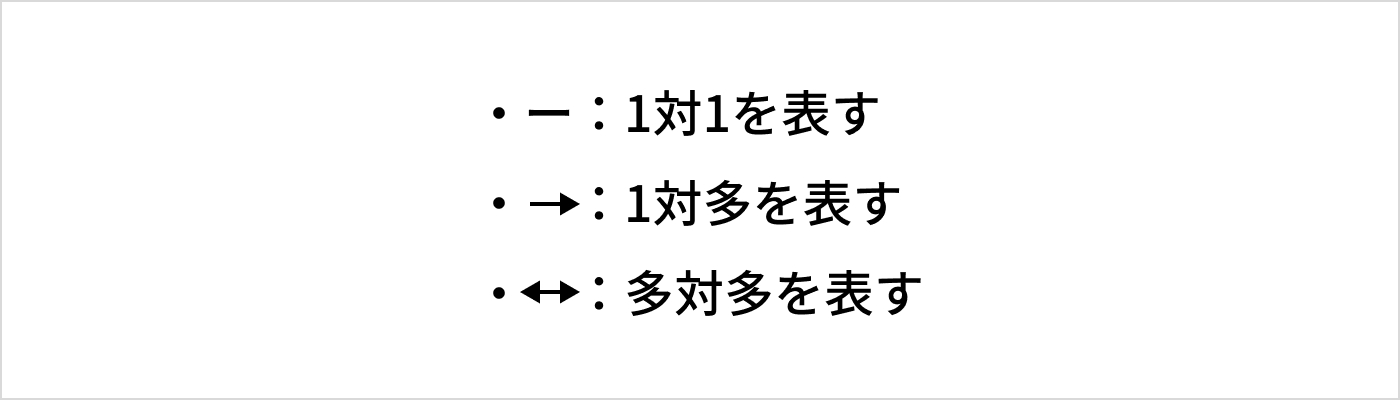
データベースで効率よくデータを管理するためには、データベースの設計が大切になる

**E-R図**

「エンティティ（実体）」と「リレーションシップ（関連）」でデータの関係性を表現する図

簡単にいうと、箱（エンティティ）と線（リレーションシップ）でデータの関係性を表現する





上図の場合は以下のような関係になる

・「校長と高校」は「１対１」の関係

・「高校と生徒」は「１対多」

・「生徒と授業数」は「多対多」

E-R図で表現することで、データベースの設計において、実体間の関係性を視覚的に表現できる、これにより、データベースの正確性や一貫性を確保できる

**正規化**

データベースの設計において、データが重複しないようにデータを整理し、データの整合性を高めるための手法

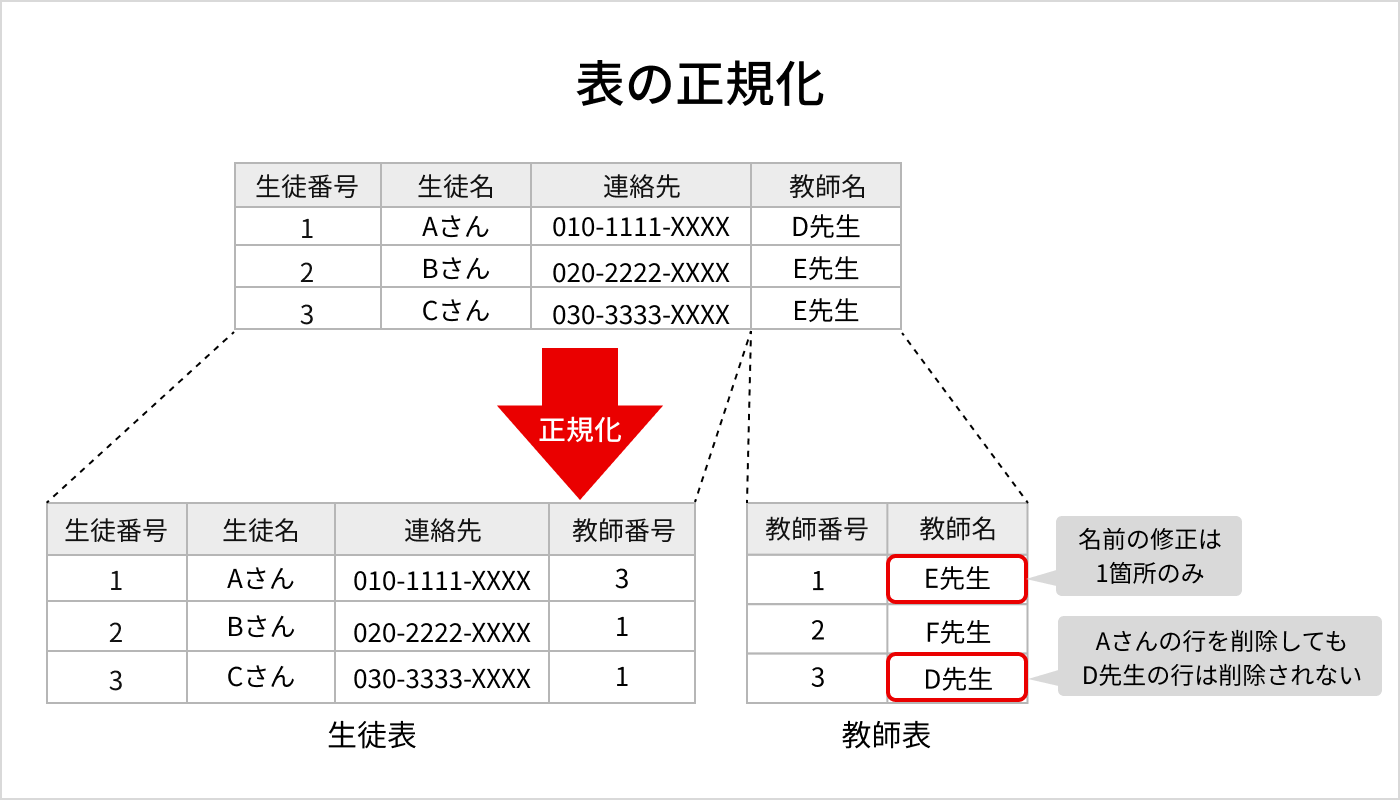
例えば、「生徒表」の中に「教師名の列」が存在している場合を考える



このような形でデータを格納していると、もし「E先生」の名字が変わった場合は、２か所の修正が必要

「Aさん」が転校することになり、Aさんの行を削除した場合、教師の「D先生」の行も削除してしまう

「生徒表」と「教師表」に分けることで、この問題を解決する



「生徒表」と「教師表」を分ければ、教師の「E先生」の苗字を更新する場合でも、１カ所の修正で済む、また「Aさん」が転校しても「D先生」の名前は削除されない

このように、データが重複しないようにデータを整理し、データの整合性を高めることを「正規化」という

ITパスポートに出題されやすいもの、まとめ

・**データベース**とは、整理された情報の集まり

・**関係データベース**とは、データを表形式で管理するデータベース

**主キー**　表の中から１つの行を特定するための列

**複合主キー**　複数の列を組み合わせた主キー

**外部キー**　関連した表同士を紐づけるために設定する列、別の表の主キーを参照する

**E-R図**　「エンティティ」と「リレーションシップ」でデータの関係性を表現する図、実体間の関係性を視覚的に表現する

**正規化**　データが重複しないようにデータを整理し、データの整合性を高めるための手法

データベース管理システムとSQL

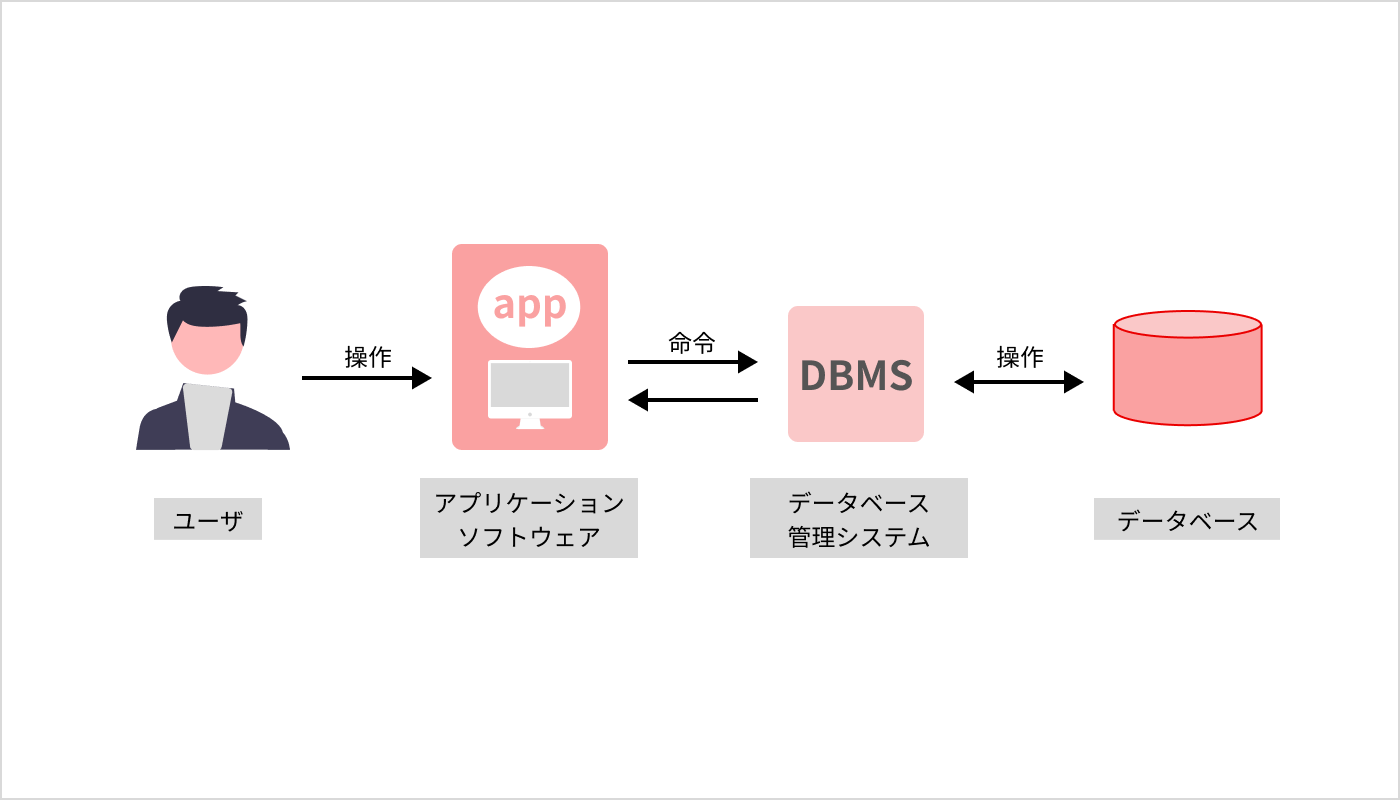
DBを操作するためには「データベース管理システム」が必要

DBを操作するときは「SQL」と呼ばれるDB用の言語を使用する

データベース管理システムとは

データベース管理システム（Database Management System:DBMS）とはDBを適切に管理するためのソフトウェア

アプリケーションソフトウェアの命令によってDBを操作する



アプリケーションソフトウェアは直接データベースを操作することはない、必ずＤＢＭＳを仲介して操作する

ＤＢＭＳがあることで、複数のアプリの同時アクセスなどを可能にするため

データベース管理システムは「ＤＢＭＳ」という略語で表現される場合がある

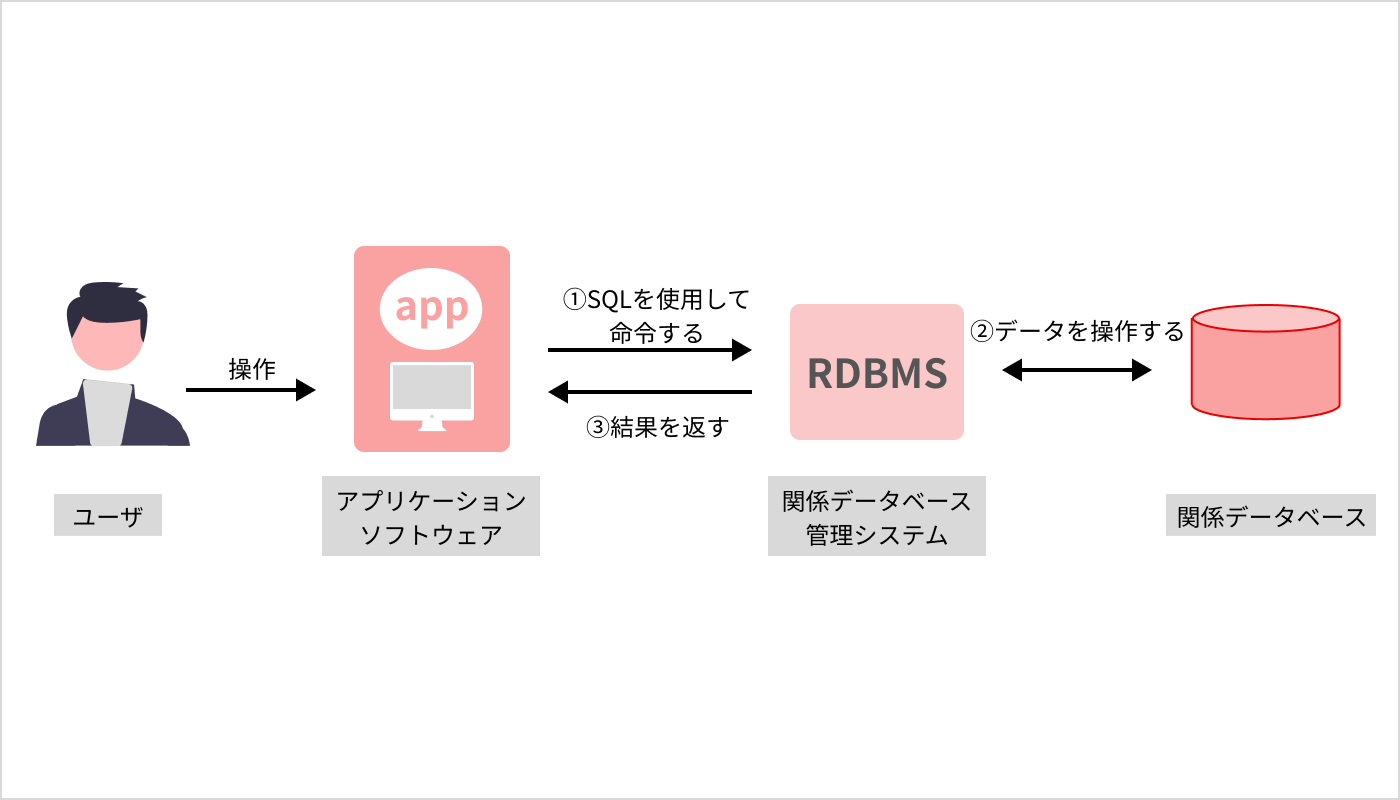
ＤＢＭＳのうち「関係データベース」を管理するデータベース管理システムのことを「ＲＤＢＭＳ（Relational Database Managenment System）」という

また「ＲＤＢＭＳ以外」のデータベース管理システムを「NoSQL(Not Only SQL)」という

SQL

SQL（Structured Query Language）とは関係データベースを操作するための言語

SQLを使用して関係データベースを操作する流れは以下のとおり



1. SQLを使用してアプリケーションソフトウェアからRDBMSに命令を出す
2. RDBMSは関係データベースのデータを操作する
3. RDBMSはアプリケーションソフトウェアに結果を返す

DBMSの３つの機能

1. データ操作
2. トランザクション管理
3. 排他制御

データ操作

データベースに保存されているデータを操作すること

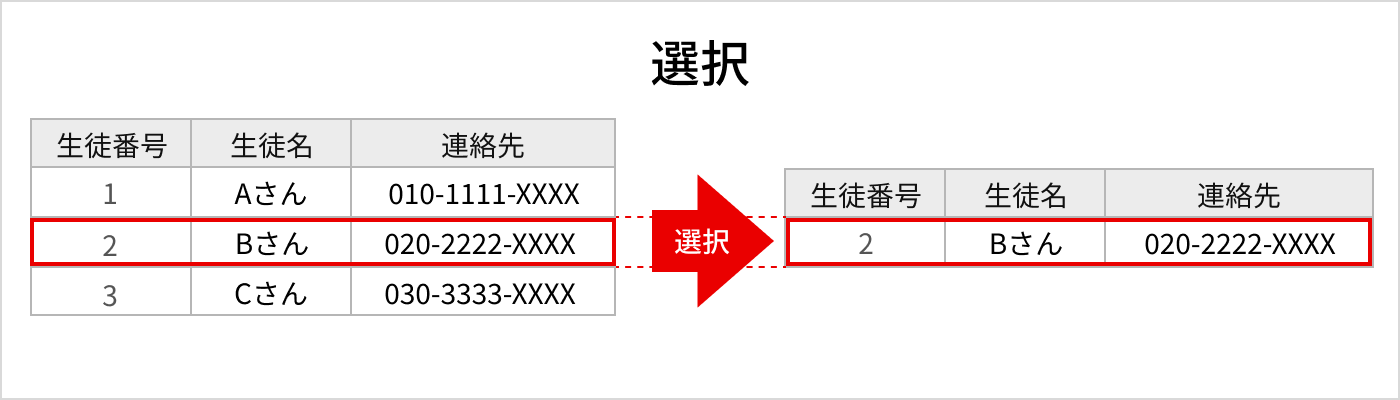
データ操作の方法は、大きく分けて６種類ある、これらの操作を組み合わせたり、繰り返し行ったりすることで、アプリからの要求に応答することが可能

| **データ操作の方法** | **説明** |
| --- | --- |
| **選択** | 表の中からある特定の行のみを抽出する操作 |
| 挿入 | 表に行を追加する操作 |
| 更新 | 行のデータを変更する操作 |
| 削除 | 表から行を削除する操作 |
| **射影（しゃえい）** | 表の中からある特定の列のみを抽出する操作 |
| **統合** | 複数の表をまとめて新しい表を作成する操作 |

#### **選択**

選択とは、**表の中からある特定の行のみを抽出する操作**です。

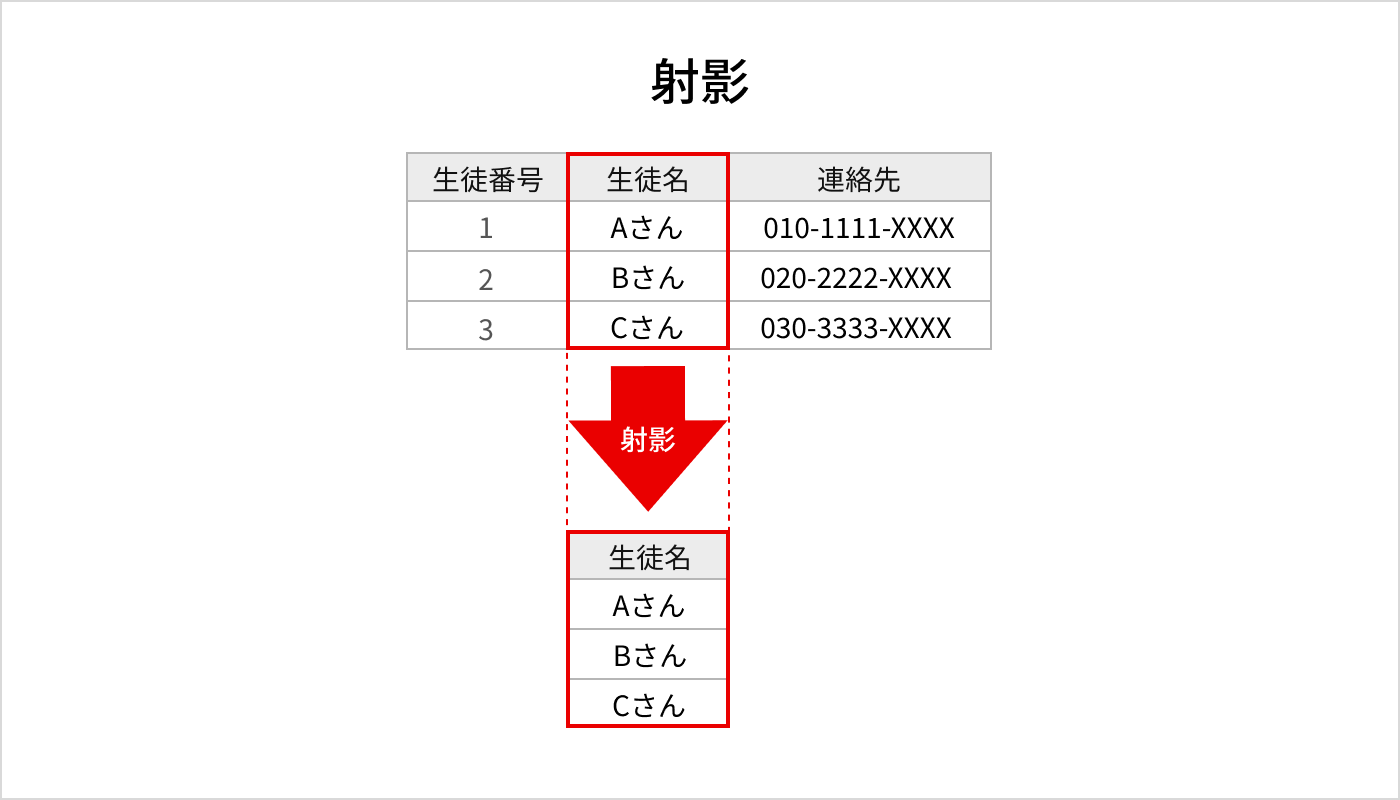
例えば、名前が「Aさん」という生徒情報だけを抽出する場合に使用します。



#### **射影**

射影とは、**表の中からある特定の列のみを抽出する操作**です。

例えば、「生徒名」という列だけを取り出す場合に使用します

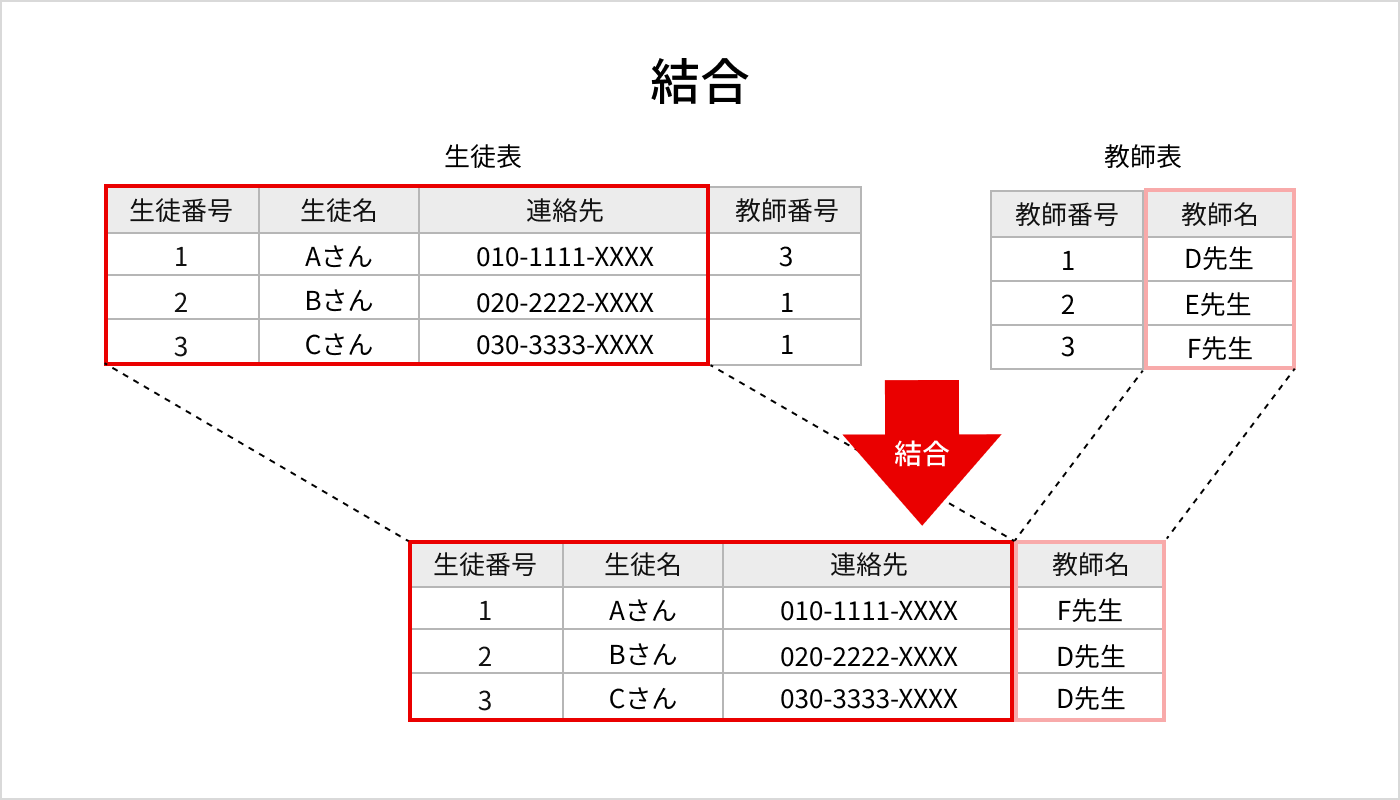


#### **結合**

結合とは、**複数の表をまとめて新しい表を作成する操作**です。

表をまとめる場合、それぞれの表に「共通の値を持つ列」が存在する必要があります。結合は「共通の値を持つ列」をキーとして、表をまとめる操作です。

下図は「教師番号」を「共通の値を持つ列」として、表をまとめた例です。

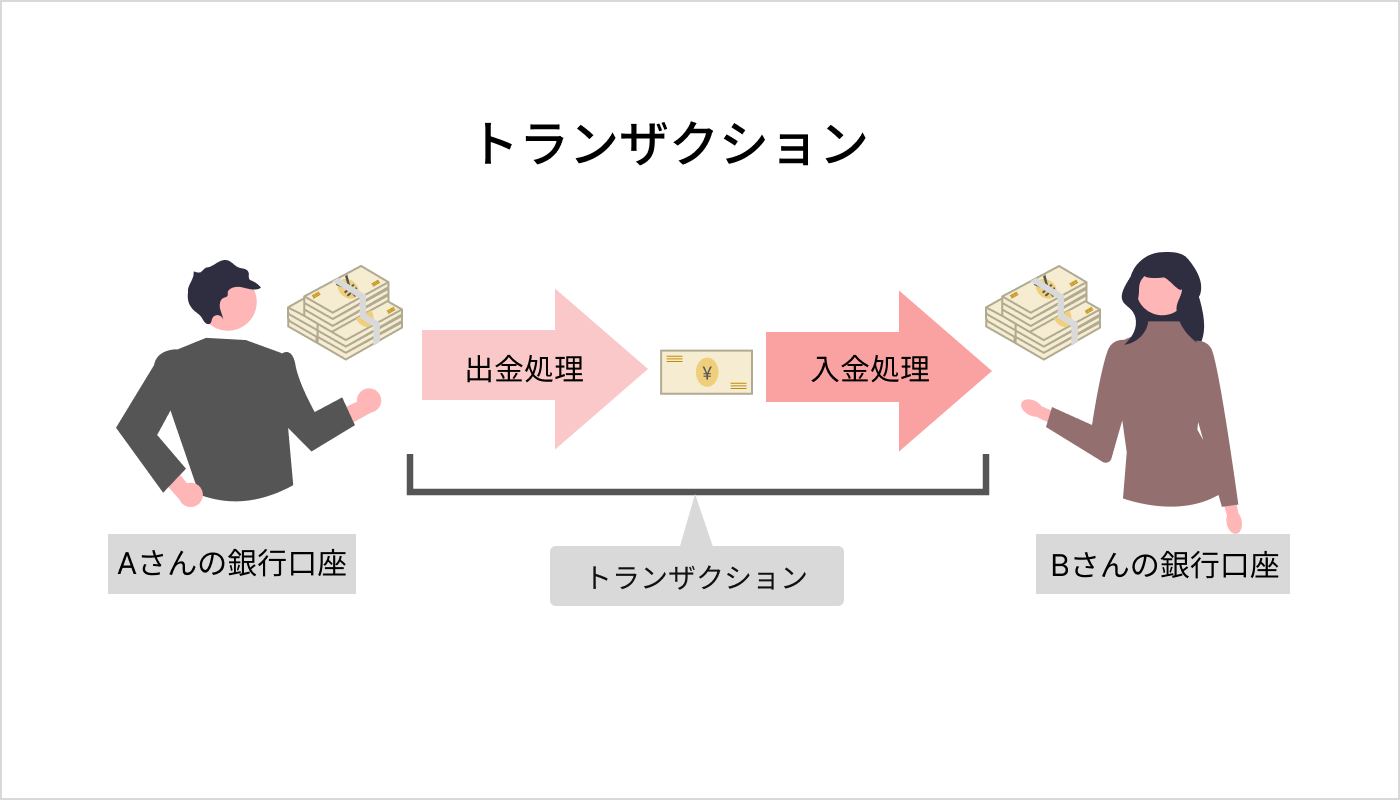
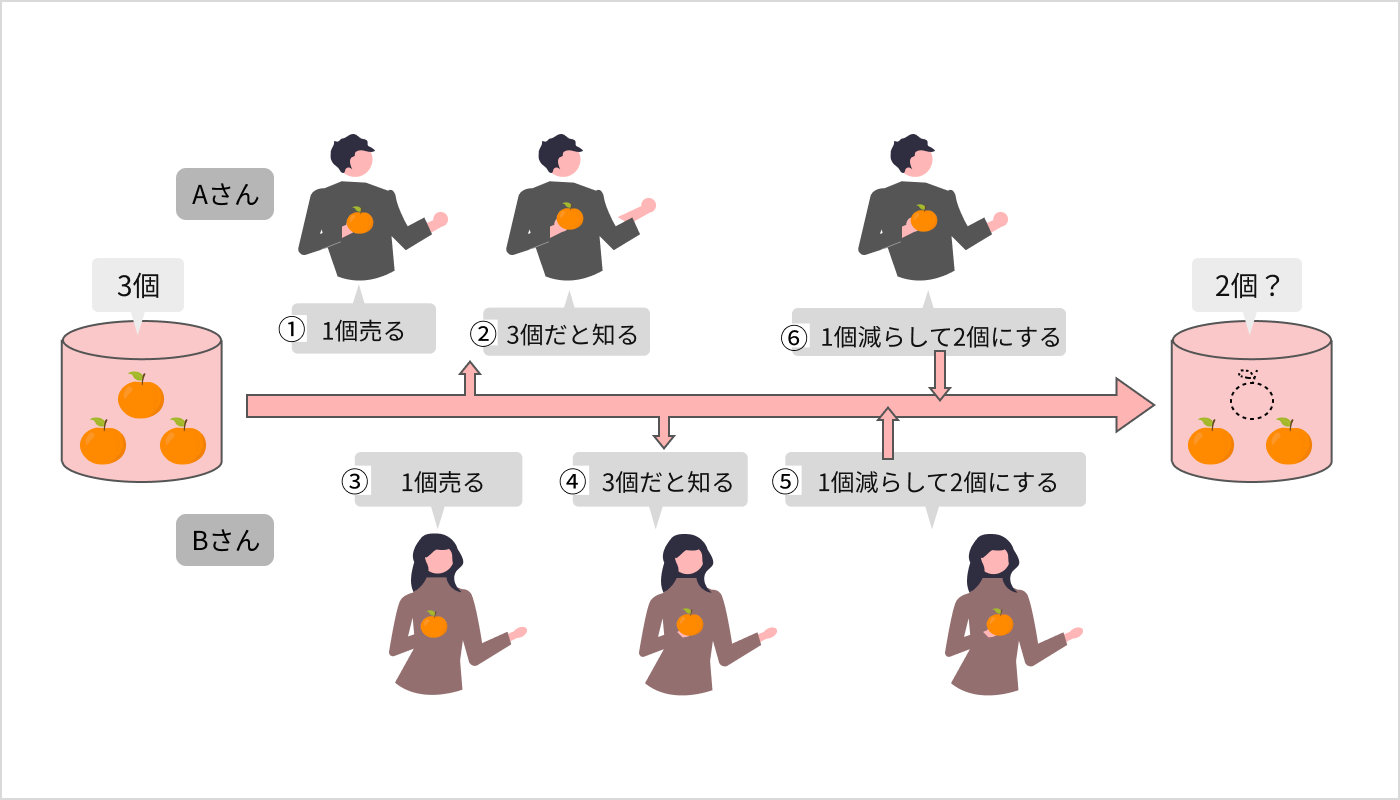


データを参照するとき、関連情報を1つにまとめるほうが分かりやすい場合があります。このようなときに行うのが「**結合**」です。

トランザクション管理  
DBにおける複数の処理をまとめて１つの処理として扱う機能

例えばAさんの銀行口座にある「１９万円」をBさんの銀行口座に送金する場合を考えてみる、ここで必要な処理は次の２つになる

* Aさんの銀行口座から10万円を出金する（出金処理）
* Bさんの銀行口座へ10万円を入金する（入金処理）

  
利用者の視点では「１９万円」という１つの処理に見えるが、システムとして「１９万円の出金処理」と「１９万円の入金処理」の２つを行う  
このシステムが行う一連の処理を１つにまとめたのが、トランザクション管理  
トランザクションが正常に処理された場合、その結果をDBに反映することを「コミット」という  
  
排他制御  
複数の人が同時にDBにアクセスし、データ更新しようとした場合にデータの一貫性を保つ機能、排他制御はDBMSの最も重要な機能といえる  
以下は排他制御を行っていない場合の例  


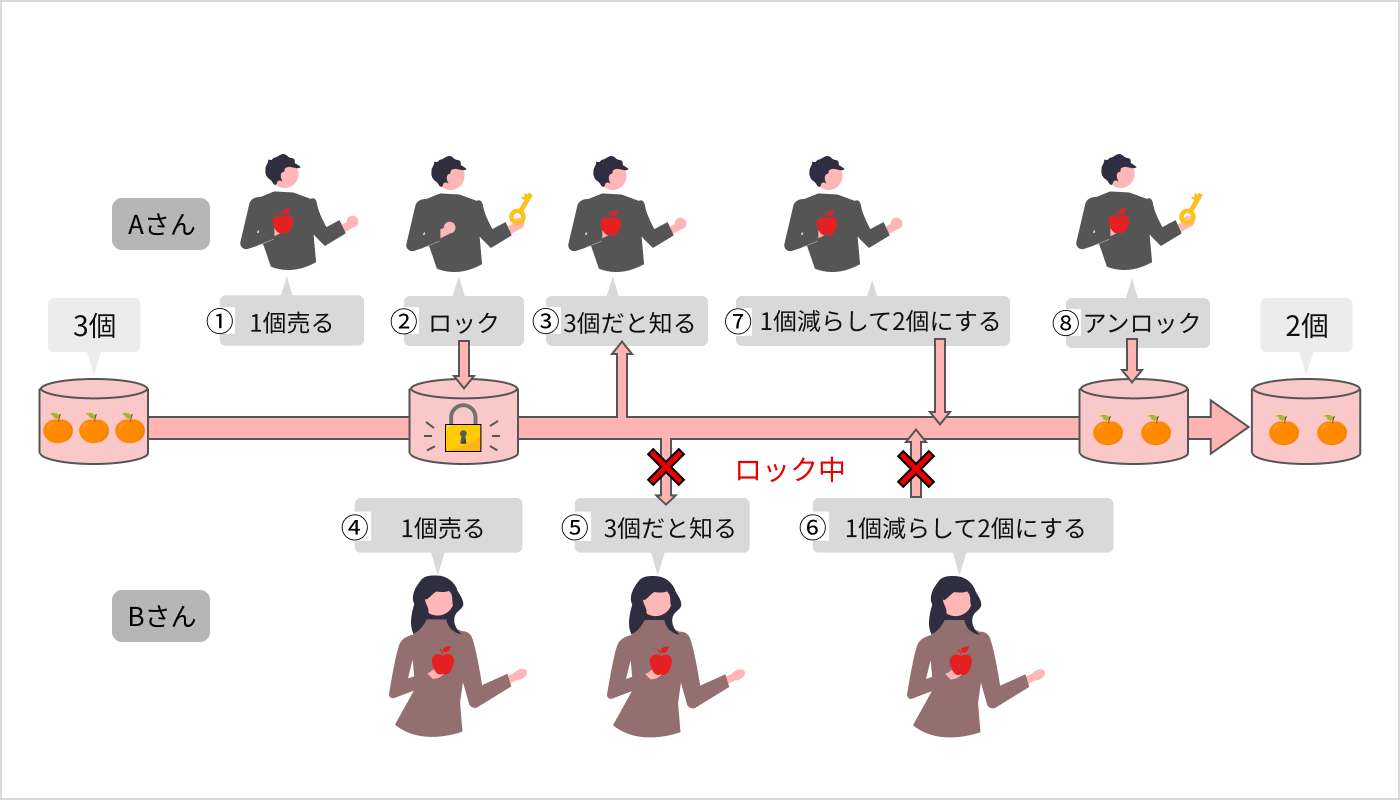
ミカンが「3個」ある状態から、AさんとBさんがそれぞれ1個ずつ売りました。

その結果、残りのミカンの数は「1個」になるはずです。しかし、同時に在庫数を更新したことで在庫数は「2個」となり、矛盾が生じてしまいました。

このような矛盾を防ぐのが「排他制御」です。

次は、排他制御で使用する「**ロック**」についての説明です。

ロック  
DBにおけるデータの更新を一時的に制限する機能、ロック機能を使用すれば、矛盾は生じない  
先ほどの例では、Aさんが先に「みかんの在庫数」を読みだした、このときに「ロック」をかけることでBさんの読み出しを制限して矛盾を防止する

  
このように「ロック」を使うことで、複数の人が同時にDBにアクセスしても矛盾が生じないようにすることが可能、また「ロック」を外すことを「アンロック」という

ITパスポートに出題されやすい用語、まとめ

* データベース管理システムとはデータベースを適切に管理するためのソフトウェア

| **用語** | **説明** |
| --- | --- |
| データ操作 | データベースに保存されているデータを操作すること |
| トランザクション管理 | データベースにおける複数の処理をまとめて一つの処理として扱う機能 |
| 排他制御 | 複数の人が同時にデータベースにアクセスして更新や削除しようとした場合に、データの一貫性を保つ機能 |
| データ操作「選択」 | 表の中からある特定の行のみを抽出する操作 |
| データ操作「射影」 | 表の中からある特定の列のみを抽出する操作 |
| データ操作「結合」 | 複数の表をまとめて新しい表を作成する操作 |