

0001 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：基礎

関係モデルにおける「属性」に相当するものとして、最も適切なものはどれか。

- ア：関係を構成する列の意味を表す項目
- イ：関係に含まれる1件分の行
- ウ：関係に格納されている行数
- エ：複数の関係を結合する処理そのもの

答え・解説 正答：ア

ア：属性は、関係を構成する列に対応し、各列が表す意味や値の領域を持つ。

イ：1件分の行はタプルに相当する。

ウ：行数はカーディナリティに相当する。

エ：関係を結合する処理は関係演算であり、属性そのものではない。

総合解説：属性は、関係を構成する列に対応し、各列が表す意味や値の領域を持つ。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0002 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

ある関係Rが属性 A, B, C, D の4属性から成り、現在25個のタプルを含む。この関係の次数とカーディナリティの組合せとして、最も適切なものはどれか。

- ア：次数25、カーディナリティ4
- イ：次数4、カーディナリティ25
- ウ：次数4、カーディナリティ4
- エ：次数25、カーディナリティ25

答え・解説 正答：イ

ア：属性数とタプル数を逆にしている。

イ：関係の次数は属性数、カーディナリティはタプル数なので、4と25である。

ウ：カーディナリティはタプル数なので25である。

エ：次数は属性数なので4である。

総合解説：関係の次数は属性数、カーディナリティはタプル数なので、4と25である。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0003 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

関係Rから、条件「部署='営業部'」を満たすタプルだけを取り出す関係代数演算はどれか。

ア：射影（projection）

イ：直積（Cartesian product）

ウ：選択（selection）

エ：和（union）

答え・解説 正答：ウ

ア：射影は必要な属性（列）を取り出す演算である。

イ：直積は二つの関係のタプルを全組合せで結合する。

ウ：選択は、条件を満たすタプル（行）を抽出する演算である。

エ：和は和両立な二関係のタプルをまとめる集合演算である。

総合解説：選択は、条件を満たすタプル（行）を抽出する演算である。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせて問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0004 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：基礎

関係R(社員番号, 氏名, 部署番号)から、部署番号だけを取り出した関係を得るときに用いる基本的な関係代数演算はどれか。

ア：選択（selection）

イ：差（difference）

ウ：直積（Cartesian product）

エ：射影（projection）

答え・解説 正答：エ

ア：選択はタプルを条件で絞り込む演算である。

イ：差は一方の関係にだけ存在するタプルを求める集合演算である。

ウ：直積は二関係のタプルの全組合せを作る。

エ：射影は指定した属性を取り出す演算であり、この場合は部署番号列を取り出す。

総合解説：射影は指定した属性を取り出す演算であり、この場合は部署番号列を取り出す。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせて問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0005 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

関係R(A,B)とS(B,C)の自然結合について、最も適切な説明はどれか。
ア：共通属性Bの値が等しいタブルを組み合わせ、共通属性を重複させずに結果を作る。
イ：RとSの全てのタブルの組合せを条件なしで作る。
ウ：Rに存在してSに存在しないタブルだけを求める。
エ：RとSから同じ位置にある属性だけを無条件に残す。

答え・解説 正答：ア

ア：自然結合は同名の共通属性の等値条件で結合し、共通属性を一つにまとめる。
イ：これは直積の説明である。
ウ：これは差の説明に近い。
エ：自然結合は属性の位置ではなく、共通属性とその値の一致を用いる。
総合解説：自然結合は同名の共通属性の等値条件で結合し、共通属性を一つにまとめる。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0006 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

関係R(A,B)とS(A,B)に和演算を適用できるための前提として、最も適切なものはどれか。
ア：両関係のタブル数が同じであること
イ：対応する属性の数と値の領域が互換で、和両立であること
ウ：両関係に主キーが定義されていないこと
エ：両関係が同じ物理ファイルに格納されていること

答え・解説 正答：イ

ア：タブル数は同じである必要はない。
イ：和・積・差などの集合演算では、対応する属性が比較可能な和両立性が必要である。
ウ：主キーの有無は和演算の前提ではない。
エ：物理格納場所は関係代数上の前提ではない。
総合解説：和・積・差などの集合演算では、対応する属性が比較可能な和両立性が必要である。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0007 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：基礎

関係RとSが和両立であるとき、 $R - S$ の結果として最も適切なものはどれか。

- ア：RとSの両方に存在するタブルの集合
- イ：RまたはSの少なくとも一方に存在するタブルの集合
- ウ：Rに存在し、Sには存在しないタブルの集合
- エ：RとSの全タブルの全組合せ

答え・解説 正答：ウ

ア：両方に存在するものは積（intersection）で求める。

イ：少なくとも一方に存在するものは和で求める。

ウ：差演算 $R - S$ は、Rに属するがSには属さないタブルを求める。

エ：全組合せは直積で求める。

総合解説：差演算 $R - S$ は、Rに属するがSには属さないタブルを求める。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0008 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

関係Rにm個、関係Sにn個のタブルがあり、重複や条件を考慮せず直積 $R \times S$ を求める。結果のタブル数として正しいものはどれか。

- ア：和集合と同じように件数を足し合わせ、最大で $m+n$ 件になる。
- イ：片方の関係の行数だけが結果件数となり、もう片方の行数は影響しない。
- ウ：条件に合う行だけを残す選択演算と同様なので、mとnだけから件数は決められない。
- エ：Rの各行とSの各行を全て組み合わせるため、 $m \times n$ 件になる。

答え・解説 正答：エ

ア：直積は行数を単純加算する演算ではない。和集合とも異なり、各行同士の全組合せを作る。

イ：直積では両方の関係の行数が結果件数に影響するため、片方だけの行数にはならない。

ウ：直積は条件で行を絞る選択演算ではなく、RとSの全ての行の組合せを生成する。

エ：直積ではRのm行それぞれにSのn行を組み合わせるので、結果件数は $m \times n$ 件である。

総合解説：直積はRの各タブルとSの各タブルを全て組み合わせるため、 $m \times n$ 個になる。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0009 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：応用

関係Enroll(学生, 科目)とRequired(科目)があり、「Requiredにある全ての科目を履修している学生」を求めたい。この要求を表現する関係代数の考え方として最も適切なものはどれか。

ア：除算 (division) を用いて、全ての指定科目との対応を満たす学生を求める。

イ：選択だけで、任意の1科目を履修する学生を求めればよい。

ウ：直積だけで、学生と科目の全組合せを作ればよい。

エ：差だけで、RequiredからEnrollをそのまま引けばよい。

答え・解説 正答：ア

ア：除算は「Sの全ての値と関係をもつR側の値」を求める全称条件の表現に適する。

イ：1科目だけでは「全て」を保証できない。

ウ：直積を作るだけでは履修実績との照合にならない。

エ：属性構造が異なる関係をそのまま差し引く考え方では要求を表せない。

総合解説：除算は「Sの全ての値と関係をもつR側の値」を求める全称条件の表現に適する。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせることで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0010 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：基礎

関係R(A,B)にタプル (1,'X'), (1,'Y'), (2,'X') がある。Rを属性Aだけに射影した関係のタプル数として最も適切なものはどれか。関係は集合として扱う。

ア：1個

イ：2個

ウ：3個

エ：射影では必ず元と同じ3個になる

答え・解説 正答：イ

ア：Aには1と2の二種類がある。

イ：射影後のAの値は1,1,2だが、関係は集合なので重複する1は一つになり、{1,2}の2タプルとなる。

ウ：重複を残す多重集合の扱いではなく、関係モデルの集合として考える。

エ：射影では属性を減らすことで重複タプルが生じ、集合として重複が除かれることがある。

総合解説：射影後のAの値は1,1,2だが、関係は集合なので重複する1は一つになり、{1,2}の2タプルとなる。学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせることで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0011 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：標準

関係代数の閉包性について、最も適切な説明はどれか。

- ア：関係演算の結果は必ず数値一つになる。
- イ：一度関係演算を行うと、その結果には別の関係演算を適用できない。
- ウ：関係に関係演算を適用した結果も関係として扱える。
- エ：関係演算は物理ファイルの並び順を保存しなければならない。

答え・解説 正答：ウ

ア：結果は関係であり、スカラ値に限定されない。
イ：閉包性により演算を組み合わせられる。
ウ：関係代数では演算の入力も出力も関係であり、演算結果にさらに演算を適用できる。
エ：物理的な行順は関係モデルの本質的性質ではない。
総合解説：関係代数では演算の入力も出力も関係であり、演算結果にさらに演算を適用できる。 学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0012 2-1 関係データモデル > 関係モデル・関係代数 | 難易度：応用

社員(社員番号, 氏名, 部署番号)から「部署番号10の社員の氏名」だけを得たい。中間結果を不必要に大きくしない基本方針として、最も適切な演算順序はどれか。

- ア：まず氏名だけに射影し、その後で部署番号=10を選択する。
- イ：社員関係を自分自身と直積してから部署番号=10を選択する。
- ウ：部署番号10とは無関係な別関係との和を先に行う。
- エ：まず部署番号=10で選択し、その結果を氏名に射影する。

答え・解説 正答：エ

ア：氏名だけに射影すると部署番号が失われ、後から部署番号条件を評価できない。
イ：不要な直積は中間結果を大きくする。
ウ：無関係な和は要求結果に不要である。
エ：条件に必要な部署番号がある段階で選択して行数を減らし、その後必要列だけに射影すると、意味を保ちながら中間結果を抑えやすい。
総合解説：条件に必要な部署番号がある段階で選択して行数を減らし、その後必要列だけに射影すると、意味を保ちながら中間結果を抑えやすい。 学習ポイント：関係代数では、選択は行、射影は列、結合は複数関係の対応付けを扱う。演算結果も関係になるため、複数演算を組み合わせで問合せを表現できる。

対象：2026年度 / 基準日 2026-09-02

0013 2-1 関係データモデル > キー・関数従属性 | 難易度：基礎

関係における候補キーの説明として、最も適切なものはどれか。

ア：タブルを一意に識別でき、かつ不要な属性を取り除けない最小のスーパーキー

イ：タブルを一意に識別できれば、どのような余分な属性を含んでもよい属性集合

ウ：外部キーとして他の関係を参照する属性だけからなる集合

エ：NULLを含むことが許される任意の非キー属性集合

答え・解説 正答：ア

ア：候補キーは一意識別能力を持つスーパーキーのうち、包含関係で最小なものをいう。

イ：これはスーパーキーの説明であり、候補キーの最小性がない。

ウ：候補キーは参照先を示す外部キーとは別概念である。

エ：候補キーはタブル識別のための属性集合であり、任意の非キー属性集合ではない。

総合解説：候補キーは一意識別能力を持つスーパーキーのうち、包含関係で最小なものをいう。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0014 2-1 関係データモデル > キー・関数従属性 | 難易度：標準

関係 $R(A,B,C)$ で、 $A \rightarrow B$ 、 $B \rightarrow C$ が成立する。このとき A から C への関係として最も適切なものはどれか。

ア： $C \rightarrow A$ が必ず成立する。

イ： $A \rightarrow C$ が推移的に成立する。

ウ： A と C は互いに独立なので関数従属性は成立しない。

エ： $A \rightarrow C$ は B の値がNULLのときだけ成立する。

答え・解説 正答：イ

ア：逆向きの関数従属性は一般には導けない。

イ：関数従属性の推移律により、 $A \rightarrow B$ かつ $B \rightarrow C$ なら $A \rightarrow C$ が導かれる。

ウ：与えられた従属性から $A \rightarrow C$ が導かれる。

エ：NULLの有無を条件にした定義ではない。

総合解説：関数従属性の推移律により、 $A \rightarrow B$ かつ $B \rightarrow C$ なら $A \rightarrow C$ が導かれる。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0015 2-1 関係データモデル > キー・関数従属性 | 難易度：標準

関数従属性 $X \rightarrow Y$ の意味として、最も適切なものはどれか。

ア：同じY値をもつ二つのタプルは、X値も必ず同じである。

イ：XとYの値は全タプルで互いに異なっていなければならない。

ウ：同じX値をもつ二つのタプルは、Y値も同じでなければならない。

エ：Xの値が変わるとYの値も必ず異なる値に変わる。

答え・解説 正答：ウ

ア：これは逆の $Y \rightarrow X$ を意味しており、一般には成立しない。

イ：関数従属性は全値が相異なることを要求しない。

ウ： $X \rightarrow Y$ は、Xの値が決まればYの値が一意に定まるという制約を表す。

エ：異なるX値が同じY値をもつことはあり得る。

総合解説： $X \rightarrow Y$ は、Xの値が決まればYの値が一意に定まるという制約を表す。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0016 2-1 関係データモデル > キー・関数従属性 | 難易度：基礎

関係R(受注番号, 商品番号, 数量, 商品名)で候補キーが(受注番号, 商品番号)であり、商品番号→商品名が成立する。商品名の候補キーへの依存について最も適切なものはどれか。

ア：候補キー全体にだけ依存する完全関数従属である。

イ：商品名→商品番号が必ず成立するため相互関数従属である。

ウ：関数従属性は一切存在しない。

エ：候補キーの一部である商品番号だけに依存する部分関数従属である。

答え・解説 正答：エ

ア：候補キー全体を必要とせず、商品番号だけで決まる。

イ：逆向きの従属性は与えられていない。

ウ：商品番号→商品名が明示されている。

エ：複合候補キーの真部分集合である商品番号だけで商品名が決まるので、部分関数従属である。

総合解説：複合候補キーの真部分集合である商品番号だけで商品名が決まるので、部分関数従属である。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0017 2-1 関係データモデル > キー・関数従属性 | 難易度：標準

一つの関係に候補キーK1とK2がある。主キーをK1に選んだ場合、K2の呼び方として最も適切なものはどれか。

- ア：代替キー
- イ：外部キー
- ウ：部分キー
- エ：非キー属性

答え・解説 正答：ア

ア：複数の候補キーのうち主キーに選ばなかった候補キーは代替キーとして扱われる。

イ：外部キーは他の関係などの候補キーを参照する属性である。

ウ：部分キーは一般的なこの文脈でK2を表す呼称ではない。

エ：K2は候補キーなので非キー属性ではない。

総合解説：複数の候補キーのうち主キーに選ばなかった候補キーは代替キーとして扱われる。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0018 2-1 関係データモデル > キー・関数従属性 | 難易度：標準

関係R(A,B,C,D)に関数従属性 $A \rightarrow B$ 、 $B \rightarrow C$ 、 $A \rightarrow D$ がある。Aの属性閉包A+として最も適切なものはどれか。

- ア：{A,B,D}
- イ：{A,B,C,D}
- ウ：{A,C,D}
- エ：{A,B,C}

答え・解説 正答：イ

ア：BからCも導出できるのでCが不足している。

イ：AからBとDが導かれ、さらに $B \rightarrow C$ からCも導かれるため、A+は全属性を含む。

ウ： $A \rightarrow B$ によりBも含まれる。

エ： $A \rightarrow D$ によりDも含まれる。

総合解説：AからBとDが導かれ、さらに $B \rightarrow C$ からCも導かれるため、A+は全属性を含む。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0019 2-1 関係データモデル > キー・関数従属性 | 難易度：基礎

関係 $R(A,B,C)$ について、関数従属性 $AB \rightarrow C$ が成立し、 $A \rightarrow C$ も $B \rightarrow C$ も成立しない。このときCのABへの依存として最も適切なものはどれか。

- ア：CはAに部分関数従属している。
- イ：CはBに部分関数従属している。
- ウ：CはABに完全関数従属している。
- エ：CはABから独立している。

答え・解説 正答：ウ

ア： $A \rightarrow C$ は成立しないとされている。
イ： $B \rightarrow C$ は成立しないとされている。
ウ：AB全体でCが決まり、その真部分集合AまたはBだけでは決まらないので完全関数従属である。
エ： $AB \rightarrow C$ が成立しているため独立ではない。
総合解説：AB全体でCが決まり、その真部分集合AまたはBだけでは決まらないので完全関数従属である。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0020 2-1 関係データモデル > キー・関数従属性 | 難易度：標準

関係 $R(A,B,C,D)$ で、関数従属性 $A \rightarrow B$ 、 $A \rightarrow C$ 、 $C \rightarrow D$ が成立し、Aが候補キーである。非キー属性DのAへの依存について最も適切なものはどれか。

- ア：Dは候補キーAの一部にだけ依存する部分関数従属である。
- イ： $D \rightarrow A$ が必ず成立する。
- ウ：AとDの間には関数従属性は成立しない。
- エ： $A \rightarrow C \rightarrow D$ という経路による推移的関数従属がある。

答え・解説 正答：エ

ア：Aは単一属性の候補キーなので「候補キーの一部」という部分依存ではない。
イ：逆向きの $D \rightarrow A$ は導出できない。
ウ：推移律により $A \rightarrow D$ が成立する。
エ：AからCが決まり、CからDが決まるので $A \rightarrow D$ が推移的に成立する。
総合解説：AからCが決まり、CからDが決まるので $A \rightarrow D$ が推移的に成立する。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0021 2-1 関係データモデル > キー・関数従属性 | 難易度：応用

関係 $R(A,B,C,D)$ に $A \rightarrow B$ 、 $B \rightarrow A$ 、 $BC \rightarrow D$ が成立する。候補キーとして最も適切なものはどれか。なお、他の関数従属性はこれらから導かれるものだけとする。

ア：{A,C} と {B,C}

イ：{A}だけ

ウ：{B}だけ

エ：{C,D}だけ

答え・解説 正答：ア

ア：ACから $A \rightarrow B$ によりBを得て、 $BC \rightarrow D$ によりDも得られるので全属性を決定する。同様にBCも全属性を決定する。どちらもAまたはB、Cを一つ除くと全属性を決定できない。

イ：AだけではCとDを決定できない。

ウ：BだけではCとDを決定できない。

エ：C,DからAまたはBを導く従属性はない。

総合解説：ACから $A \rightarrow B$ によりBを得て、 $BC \rightarrow D$ によりDも得られるので全属性を決定する。同様にBCも全属性を決定する。どちらもAまたはB、Cを一つ除くと全属性を決定できない。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0022 2-1 関係データモデル > キー・関数従属性 | 難易度：応用

関係 $R(A,B,C)$ に関数従属性 $AB \rightarrow C$ 、 $C \rightarrow B$ が成立する。スーパーキーでない決定項が存在するかという観点で最も適切な説明はどれか。

ア： $AB \rightarrow C$ の決定項ABが候補キーなので、必ずBCNFを満たす。

イ： $C \rightarrow B$ の決定項Cはスーパーキーでないため、BCNFの条件を満たさない。

ウ： $C \rightarrow B$ があるとCは自動的に候補キーになるのでBCNFを満たす。

エ：BCNFでは関数従属性を一切許さないため、どの関係もBCNFにならない。

答え・解説 正答：イ

ア：一つの従属性の決定項がキーでも、他に非キー決定項があればBCNF違反となる。

イ：BCNFでは、自明でない関数従属性 $X \rightarrow Y$ の決定項Xがスーパーキーであることを要求する。C+はBまででAを含まないためCはスーパーキーでない。

ウ：CからAを導けないのでCは候補キーではない。

エ：BCNFは関数従属性そのものを禁止するのではなく、決定項に条件を課す。

総合解説：BCNFでは、自明でない関数従属性 $X \rightarrow Y$ の決定項Xがスーパーキーであることを要求する。C+はBまででAを含まないためCはスーパーキーでない。学習ポイント：キーと関数従属性は正規化の基礎である。属性閉包で決定可能な属性を確認し、候補キーの最小性、部分依存・推移依存、決定項の性質を区別する。

対象：2026年度 / 基準日 2026-09-02

0023 2-1 関係データモデル > 整合性・制約 | 難易度：基礎

関係データベースの実体整合性 (entity integrity) として、最も適切な考え方はどれか。

ア：外部キーは必ず参照先に存在しない値だけを保持する。

イ：全ての非キー属性は同じ値でなければならない。

ウ：主キーを構成する属性の値によって各タプルを一意に識別でき、主キー値を欠落させない。

エ：全ての列に同一のデータ型を使用する。

答え・解説 正答：ウ

ア：これは参照整合性と逆の内容である。

イ：非キー属性が同一である必要はない。

ウ：実体整合性は、各実体を主キーによって一意に識別できることを確保する考え方である。

エ：列ごとに適切なドメインを持てばよく、全列同型である必要はない。

総合解説：実体整合性は、各実体を主キーによって一意に識別できることを確保する考え方である。

学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0024 2-1 関係データモデル > 整合性・制約 | 難易度：標準

注文(注文番号, 顧客番号, ...)の顧客番号が顧客(顧客番号, ...)を参照している。参照整合性を直接保つための制約として最も適切なものはどれか。

ア：注文.注文番号を常に0にするCHECK制約を設定する。

イ：顧客表の全列をNOT NULLにするだけでよい。

ウ：注文表と顧客表の物理格納先を同じディスクにする。

エ：注文.顧客番号を外部キーとして顧客.顧客番号を参照させる。

答え・解説 正答：エ

ア：注文番号を0に固定しても顧客参照の妥当性は保証できない。

イ：NOT NULLだけでは参照先に該当顧客が存在することを保証しない。

ウ：物理格納先は論理的な参照整合性とは無関係である。

エ：外部キー制約は、参照元の値と参照先キーの対応をDBMSに検証させ、参照整合性を保つ基本手段である。

総合解説：外部キー制約は、参照元の値と参照先キーの対応をDBMSに検証させ、参照整合性を保つ基本手段である。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0025 2-1 関係データモデル > 整合性・制約 | 難易度：標準

列「数量」に1以上999以下の整数だけを許可したい。値域をデータベース側で検証する目的に最も適した制約はどれか。

- ア：CHECK制約
- イ：外部キー制約
- ウ：ビューだけを作成すること
- エ：インデックスだけを作成すること

答え・解説 正答：ア

ア：CHECK制約は、行の値が指定した条件を満たすことを検証するのに適する。
イ：外部キーは他のキー値との参照関係を検証する。
ウ：ビューだけでは基表への不正値登録を一般に防げない。
エ：インデックスは検索効率や一意性に関係し得るが、範囲条件そのものを表す基本手段ではない。
総合解説：CHECK制約は、行の値が指定した条件を満たすことを検証するのに適する。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0026 2-1 関係データモデル > 整合性・制約 | 難易度：基礎

社員表のメールアドレス列について、NULLは許可するが、値が登録される場合は同じメールアドレスを重複させたくない。設計意図として最も近い制約はどれか。

- ア：メールアドレスを外部キーにする。
- イ：メールアドレスに一意性制約を設ける。
- ウ：メールアドレスを全社員で同じ固定値にする。
- エ：メールアドレス列を削除する。

答え・解説 正答：イ

ア：外部キーは別の候補キーなどを参照するための制約であり、単独で重複禁止を表さない。
イ：登録された値の重複を禁止する目的には一意性制約が適する。NULLの扱いの細部はDBMS仕様を確認する必要がある。
ウ：固定値にすると一意性の意図に反する。
エ：列削除では業務要件を満たさない。
総合解説：登録された値の重複を禁止する目的には一意性制約が適する。NULLの扱いの細部はDBMS仕様を確認する必要がある。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0027 2-1 関係データモデル > 整合性・制約 | 難易度：標準

参照先の親行を削除しようとしたとき、参照中の子行が存在するなら削除を拒否したい。最も適切な方針はどれか。

ア：親行を削除した後も、子行の外部キーに存在しない親キー値を必ず残す。

イ：外部キー制約を無効にしてから毎回削除する。

ウ：参照整合性制約により、参照中は親行の削除を拒否する。

エ：主キーを削除して全ての行を区別できなくする。

答え・解説 正答：ウ

ア：孤児参照を残すため参照整合性に反する。

イ：制約を無効化することは整合性を守る方針ではない。

ウ：参照中の親行削除を禁止する動作は、参照整合性を維持する代表的な方針である。

エ：主キー削除は実体識別を損なう。

総合解説：参照中の親行削除を禁止する動作は、参照整合性を維持する代表的な方針である。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0028 2-1 関係データモデル > 整合性・制約 | 難易度：標準

業務規則「終了日は開始日以降でなければならない」を同一行の二列で保証したい。最も適切な方法はどれか。

ア：両列を外部キーにする。

イ：終了日にだけ一意性制約を付ける。

ウ：主キーを開始日だけに変更する。

エ：開始日と終了日の大小関係を検証するCHECK制約を定義する。

答え・解説 正答：エ

ア：外部キーは参照先との対応を保証するもので、日付の大小を直接保証しない。

イ：一意性は重複を防ぐが大小関係を保証しない。

ウ：主キー変更は日付順序の検証にはならない。

エ：同一行内の値同士の条件を検証する業務規則は、DBMSが対応する範囲でCHECK制約として表現できる。

総合解説：同一行内の値同士の条件を検証する業務規則は、DBMSが対応する範囲でCHECK制約として表現できる。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0029 2-1 関係データモデル > 整合性・制約 | 難易度：応用

複合主キー(A,B)をもつ関係で、Aだけが同じ複数タプルを許容する設計について最も適切な説明はどれか。

- ア：A単独では一意でなくても、(A,B)の組が一意なら複合主キーの一意性を満たせる。
- イ：主キーの各構成属性はそれぞれ単独で必ず一意でなければならない。
- ウ：Aが重複すると必ず参照整合性違反になる。
- エ：複合主キーではBの値は常にAと同じでなければならない。

答え・解説 正答：ア

- ア：複合キーの一意性は構成属性の組合せ全体で判定する。
- イ：各属性単独の一意性までは要求されない。
- ウ：Aの重複自体は参照整合性違反を意味しない。
- エ：AとBが同値である必要はない。

総合解説：複合キーの一意性は構成属性の組合せ全体で判定する。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0030 2-1 関係データモデル > 整合性・制約 | 難易度：応用

顧客区分コードを顧客表に保存する。許可される区分コードは区分マスタで一元管理され、区分追加も運用で行われる。この要件に最も適した整合性設計はどれか。

- ア：許可コードを全てアプリケーションの画面文言だけに埋め込み、DBでは検証しない。
- イ：顧客区分コードを区分マスタのキーへ外部キー参照させる。
- ウ：顧客表に区分名を自由入力し、区分マスタとは関連付けない。
- エ：区分コード列を主キーにして顧客を1区分1件に制限する。

答え・解説 正答：イ

- ア：DB側の整合性を保証できず、複数アプリからの更新で不整合が生じ得る。
 - イ：変更され得る許可値集合をマスタで管理する場合、外部キーで参照させると登録値をマスタに存在するコードへ制限できる。
 - ウ：表記揺れや存在しない区分が登録される可能性がある。
 - エ：顧客の識別子と区分コードは役割が異なり、区分ごとに1顧客しか登録できなくなる。
- 総合解説：変更され得る許可値集合をマスタで管理する場合、外部キーで参照させると登録値をマスタに存在するコードへ制限できる。学習ポイント：整合性制約は、主キーによる実体識別、外部キーによる参照関係、CHECKや一意性などによる値・業務ルールをデータベース側で保証するために使う。

対象：2026年度 / 基準日 2026-09-02

0031 2-2 概念データモデリング > エンティティ・属性 | 難易度：基礎

概念データモデリングにおけるエンティティの説明として、最も適切なものはどれか。

ア：データベースの物理ファイルそのもの

イ：SQL文の実行順序だけを表すもの

ウ：業務上、独立して識別・管理する必要がある対象を抽象化したもの

エ：必ず一つの数値属性だけで構成されるもの

答え・解説 正答：ウ

ア：物理ファイルは実装上の構造であり概念エンティティとは異なる。

イ：SQL実行順序ではない。

ウ：エンティティは、人・物・事象など業務上識別して管理したい対象を概念化したものである。

エ：エンティティは複数属性を持ち得る。

総合解説：エンティティは、人・物・事象など業務上識別して管理したい対象を概念化したものである。

学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0032 2-2 概念データモデリング > エンティティ・属性 | 難易度：標準

「社員」というエンティティタイプと「社員番号E001の田中さん」の関係として、最も適切なものはどれか。

ア：前者が属性、後者がリレーションシップである。

イ：前者が外部キー、後者が主キーである。

ウ：両者は常に同じ概念で区別しない。

エ：前者がエンティティタイプ、後者がそのインスタンスである。

答え・解説 正答：エ

ア：社員は属性ではなく業務対象の種類である。

イ：型と個体の関係をキー概念で置き換えることはできない。

ウ：概念モデリングでは型と個々の実体を区別する。

エ：エンティティタイプは同種の実体の集合を表し、個々の具体的な対象がインスタンスである。

総合解説：エンティティタイプは同種の実体の集合を表し、個々の具体的な対象がインスタンスである。

学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0033 2-2 概念データモデリング > エンティティ・属性 | 難易度：標準

社員エンティティの属性として「氏名」を「姓」と「名」に分解して扱う必要がある。この「氏名」の捉え方として最も適切なものはどれか。

- ア：複合属性
- イ：導出属性
- ウ：多値属性
- エ：識別リレーションシップ

答え・解説 正答：ア

ア：より細かな意味単位である姓・名へ分解できる属性は、概念上は複合属性として捉えられる。
イ：導出属性は他の値から計算・導出される属性である。
ウ：多値属性は一つの実体が複数值を持つ属性である。
エ：リレーションシップは実体間の関連であり属性ではない。
総合解説：より細かな意味単位である姓・名へ分解できる属性は、概念上は複合属性として捉えられる。
学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0034 2-2 概念データモデリング > エンティティ・属性 | 難易度：基礎

生年月日から現在年齢を計算できる場合、概念モデル上の「年齢」の扱いとして最も適切なものはどれか。

- ア：必ず主キーにする。
- イ：生年月日などから求められる導出属性として扱うことを検討する。
- ウ：必ず多値属性にする。
- エ：生年月日とは無関係な独立エンティティにする。

答え・解説 正答：イ

ア：年齢は通常、実体識別には適さない。
イ：年齢は基準日によって変化し、生年月日から導出できるため、導出値として扱う設計が考えられる。
ウ：一人が同時に複数の現在年齢を持つという意味ではない。
エ：独立した業務対象ではなく社員等の性質である。
総合解説：年齢は基準日によって変化し、生年月日から導出できるため、導出値として扱う設計が考えられる。
学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0035 2-2 概念データモデリング > エンティティ・属性 | 難易度：標準

一人の顧客が複数の電話番号を持てる。概念モデルで電話番号を単一属性にカンマ区切りで詰め込む案と比較して、最も適切な考え方はどれか。

ア：常にカンマ区切り文字列にすれば、検索・制約・追加件数の変更にも最適である。

イ：電話番号は顧客の主キーに含めなければならない。

ウ：複数値を持つという業務構造を明示し、必要なら電話番号を別エンティティ等としてモデル化する。

エ：複数電話番号を許すなら顧客エンティティ自体を作れない。

答え・解説 正答：ウ

ア：区切り文字列は値の構造を隠し、個別番号への制約や検索を難しくする。

イ：電話番号が顧客識別に必須とは限らない。

ウ：複数値を一つ相当へ詰め込むより、繰返し構造をモデル上で明示した方が意味・制約・検索を扱いやすい。

エ：多値を持つ属性があっても適切にモデル化できる。

総合解説：複数値を一つ相当へ詰め込むより、繰返し構造をモデル上で明示した方が意味・制約・検索を扱いやすい。学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0036 2-2 概念データモデリング > エンティティ・属性 | 難易度：標準

受注明細のように、親である受注の識別子と明細番号の組合せで初めて一意に識別できる対象を概念モデルで捉えるとき、最も適切なものはどれか。

ア：親と無関係な単一の導出属性として扱う。

イ：必ず親エンティティの属性一つに全明細を埋め込む。

ウ：リレーションシップを一切持たない独立エンティティとする。

エ：親への識別関係を伴う弱エンティティとして捉えることができる。

答え・解説 正答：エ

ア：明細は導出属性ではなく識別・管理される対象となり得る。

イ：複数明細を単一属性に埋め込むと構造を適切に表現しにくい。

ウ：親との関係が識別に関わるため完全独立とはいえない。

エ：弱エンティティは、自身の部分的な識別子だけではなく、所有者側の識別子と組み合わせて識別される。

総合解説：弱エンティティは、自身の部分的な識別子だけではなく、所有者側の識別子と組み合わせて識別される。学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0037 2-2 概念データモデリング > エンティティ・属性 | 難易度：基礎

顧客エンティティに「顧客番号」と「氏名」があり、顧客番号は業務上重複せず変更もしない。概念モデルで顧客を識別する属性として最も適切なのはどれか。

- ア：顧客番号
- イ：氏名
- ウ：登録日時の月だけ
- エ：顧客の注文金額合計

答え・解説 正答：ア

ア：業務上安定して一意な顧客番号は、顧客インスタンスを識別する識別子として適する。

イ：氏名は同姓同名があり得る。

ウ：月だけでは多数顧客が重複する。

エ：注文金額合計は変動し、一意性も保証されない。

総合解説：業務上安定して一意な顧客番号は、顧客インスタンスを識別する識別子として適する。学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0038 2-2 概念データモデリング > エンティティ・属性 | 難易度：標準

概念モデル作成時に、属性「住所」を一つの文字列とするか、郵便番号・都道府県・市区町村などに分けるかを決める際の判断として最も適切なものはどれか。

- ア：物理ディスクの空き容量だけで決める。
- イ：検索・検証・業務ルールなどで各構成要素を独立して扱う必要性を確認する。
- ウ：常に最も細かい文字単位まで分解する。
- エ：常に一つの文字列にまとめ、業務要件は考慮しない。

答え・解説 正答：イ

ア：概念設計の粒度は物理容量だけで決まらない。

イ：属性の粒度は、業務上の意味と利用・制約の要件に合わせて決めるべきである。

ウ：過剰な分解は意味のない複雑化を招く。

エ：構成要素別の検索や検証が必要なら単一文字列では扱いにくい。

総合解説：属性の粒度は、業務上の意味と利用・制約の要件に合わせて決めるべきである。学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0039 2-2 概念データモデリング > エンティティ・属性 | 難易度：応用

業務ヒアリングで「商品」は商品コードで識別できる一方、「商品の現在価格」は価格改定で変わることが分かった。概念モデル上の判断として最も適切なものはどれか。

ア：現在価格が変わるたびに商品の識別子も必ず変更する。

イ：現在価格が同じ商品は同一商品として統合する。

ウ：商品コードは識別子候補だが、現在価格は通常、商品の可変な属性であり識別子には向かない。

エ：商品コードは価格が変わると一意性を失う。

答え・解説 正答：ウ

ア：価格改定で対象そのものの同一性が変わるとは限らない。

イ：同価格でも別商品はあり得る。

ウ：識別子は対象の同一性を安定して表す必要があり、変動する価格は通常その役割に適さない。

エ：商品コードの一意性は価格変化とは別である。

総合解説：識別子は対象の同一性を安定して表す必要があり、変動する価格は通常その役割に適さない。
学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0040 2-2 概念データモデリング > エンティティ・属性 | 難易度：応用

顧客の「希望連絡方法」をメール・電話など複数選択可能とし、各方法に優先順位も持たせる。概念モデルとして最も表現力が高い設計はどれか。

ア：顧客表相当の一つの自由記述属性に全て詰め込む。

イ：優先順位を顧客の氏名属性に含める。

ウ：連絡方法ごとに顧客エンティティを複製する。

エ：顧客と連絡方法の間に、優先順位などの属性を持つ関連対象を設ける。

答え・解説 正答：エ

ア：自由記述では個々の方法や順位への制約・検索が難しい。

イ：氏名と優先順位は意味が異なる。

ウ：顧客を複製すると同一顧客の一貫性維持が難しくなる。

エ：顧客と連絡方法の対応そのものに優先順位という性質があるため、関連をエンティティ化する設計が適する。

総合解説：顧客と連絡方法の対応そのものに優先順位という性質があるため、関連をエンティティ化する設計が適する。
学習ポイント：概念データモデルでは、業務対象をエンティティ、その性質を属性として表す。識別子の安定性、多値・導出・複合属性、弱エンティティなどを業務意味から判断する。

対象：2026年度 / 基準日 2026-09-02

0041 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：基礎

部門と社員の関係が「一つの部門に複数社員が所属でき、各社員は一つの部門に所属する」である。多重度として最も適切なものはどれか。

ア：部門1対社員多の1:N

イ：1:1

ウ：N:M

エ：部門多対社員1という意味で社員一人が複数部門を同時に必ず持つ

答え・解説 正答：ア

ア：一部門に複数社員、社員側からは一部門なので、部門から社員への1:Nである。

イ：一部門に複数社員を許せない。

ウ：社員が複数部門を持つ要件ではないのでN:Mではない。

エ：方向の説明が要件と逆になっている。

総合解説：一部門に複数社員、社員側からは一部門なので、部門から社員への1:Nである。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0042 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

学生と科目の関係が「学生は複数科目を履修でき、各科目は複数学生に履修される」である。最も適切な多重度はどれか。

ア：1:1

イ：N:M

ウ：1:Nのみ

エ：0:1のみ

答え・解説 正答：イ

ア：双方が一つに限定されない。

イ：両側とも複数の相手と対応できるため多対多である。

ウ：片側だけが多ではない。

エ：任意参加の有無とN:Mという最大多重度は別の論点である。

総合解説：両側とも複数の相手と対応できるため多対多である。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0043 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

社員には駐車場区画が割り当てられる場合も割り当てられない場合もあり、一つの区画は高々一人にしか割り当てない。この関係として最も適切なものはどれか。

ア：社員から区画は必ず1、区画から社員は必ず多数

イ：社員から区画は必ず多数、区画から社員は必ず1

ウ：社員から駐車場区画は0または1、区画から社員も0または1の任意1:1

エ：双方必ず1で、未割当を許さない1:1

答え・解説 正答：ウ

ア：多数割当の要件ではない。

イ：社員が複数区画を持つ要件ではない。

ウ：最大多重度は双方1だが、割当なしを許すため最小多重度は0になり得る。

エ：未割当を許す条件を表せない。

総合解説：最大多重度は双方1だが、割当なしを許すため最小多重度は0になり得る。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0044 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：基礎

「全ての受注明細は必ず一つの受注に属するが、受注はまだ明細を持たない状態でも作成できる」という要件を最も適切に表すものはどれか。

ア：両側とも必須参加にする。

イ：両側とも任意参加にする。

ウ：受注明細から受注への関係を削除する。

エ：受注明細から受注への参加は必須、受注から受注明細への参加は任意とする。

答え・解説 正答：エ

ア：受注明細0件の受注を許せなくなる。

イ：親なしの明細を許してしまう。

ウ：明細の所属先を表現できない。

エ：明細は親受注なしに存在できないが、受注は明細0件でも存在できるという最小多重度の違いを表す。

総合解説：明細は親受注なしに存在できないが、受注は明細0件でも存在できるという最小多重度の違いを表す。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0045 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

社員が別の社員を上司として持つ「上司 - 部下」の関係を概念モデルで表す。最も適切なものはどれか。

- ア：社員エンティティが自分自身と結ぶ再帰的リレーションシップ
- イ：社員と無関係な商品エンティティとの1:1
- ウ：社員属性を全て削除したモデル
- エ：上司だけを物理ファイル名として表すモデル

答え・解説 正答：ア

ア：同じエンティティタイプのインスタンス同士の関係なので再帰的リレーションシップで表せる。
イ：業務対象が異なる。
ウ：関係表現のために属性を削除する必要はない。
エ：物理ファイル名では概念上の上司部下関係を表せない。
総合解説：同じエンティティタイプのインスタンス同士の関係なので再帰的リレーションシップで表せる。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0046 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

「医師が患者に対して薬剤を処方する」という事実について、処方量が医師だけでも患者だけでも薬剤だけでも決まらず、三者の組合せに依存する。最も適切な概念モデルはどれか。

- ア：医師と患者の1:1だけで全情報を表す。
- イ：医師・患者・薬剤を結ぶ三項リレーションシップ、またはそれを表す処方エンティティを用いる。
- ウ：薬剤を患者の氏名属性に埋め込む。
- エ：三者間の関係を無視して三つの独立エンティティだけにする。

答え・解説 正答：イ

ア：薬剤ごとの差異や処方量を表せない。
イ：三者の組合せ自体に意味と属性があるため、三項関係または関連エンティティとして表すのが自然である。
ウ：意味の異なる情報を氏名に埋め込むのは不適切である。
エ：処方という業務事実を失う。
総合解説：三者の組合せ自体に意味と属性があるため、三項関係または関連エンティティとして表すのが自然である。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0047 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：基礎

学生と科目のN:M関係「履修」に、履修年度と成績という属性がある。最も適切なモデル化はどれか。

ア：成績を学生氏名の一部にする。

イ：履修年度を科目名に埋め込む。

ウ：履修を関連エンティティとして表し、学生・科目との関係と履修年度・成績を持たせる。

エ：N:M関係には属性を持ってないので情報を捨てる。

答え・解説 正答：ウ

ア：氏名とは独立した情報である。

イ：科目そのものの名称と履修年度は別概念である。

ウ：履修年度や成績は学生単体・科目単体ではなく、その組合せである履修事実に属する。

エ：関係に固有の属性は関連エンティティとして表現できる。

総合解説：履修年度や成績は学生単体・科目単体ではなく、その組合せである履修事実に属する。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0048 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

一つの注文には1件以上の注文明細が必要で、各注文明細は必ず一つの注文に属する。最小・最大多重度の表現として最も適切なものはどれか。

ア：注文から明細は0..1、明細から注文は0..*

イ：注文から明細は1..1、明細から注文は1..*

ウ：注文から明細は0..*、明細から注文は0..1

エ：注文から明細は1..*、明細から注文は1..1

答え・解説 正答：エ

ア：注文の明細数と明細の所属条件の両方が要件と異なる。

イ：注文を1明細に制限してしまう。

ウ：注文の最低1件と明細の必須所属を表せない。

エ：注文は最低1明細・上限複数、明細は必ず一つの注文に属するので1..*と1..1である。

総合解説：注文は最低1明細・上限複数、明細は必ず一つの注文に属するので1..*と1..1である。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0049 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：応用

部門 - 社員が1:Nであり、「社員が部門を異動しても過去の所属履歴を残したい」という要件が追加された。現在所属だけを表す単純な1:Nモデルに対する改善として最も適切なものはどれか。

ア：所属履歴を表す関連エンティティを設け、開始日・終了日などを持たせる。

イ：社員エンティティを異動のたびに別人として複製する。

ウ：過去所属は全て削除し、現在部門だけを上書きする。

エ：部門名を社員氏名の末尾に付加する。

答え・解説 正答：ア

ア：期間を伴う所属という業務事実を独立してモデル化すると、同一社員の時系列の所属を保持できる。

イ：同一人物の同一性が失われる。

ウ：履歴保存要件を満たさない。

エ：氏名に所属情報を混在させるのは不適切である。

総合解説：期間を伴う所属という業務事実を独立してモデル化すると、同一社員の時系列の所属を保持できる。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0050 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：基礎

顧客と担当者の関係について「各顧客には必ず1人の主担当者がいる。担当者は0人以上の顧客を担当できる」とする。顧客側から担当者を見た多重度として最も適切なものはどれか。

ア：0..1

イ：1..1

ウ：0..*

エ：1..*

答え・解説 正答：イ

ア：担当者不在を許してしまう。

イ：顧客ごとに主担当者が必ず一人なので、最小1・最大1である。

ウ：複数担当者を許してしまう。

エ：一人より多い担当者を許してしまう。

総合解説：顧客ごとに主担当者が必ず一人なので、最小1・最大1である。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0051 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：標準

商品と倉庫の間に在庫数量がある。商品は複数倉庫に置かれ、倉庫は複数商品を保管する。最も適切なモデル化はどれか。

ア：在庫数量を商品だけの属性にして倉庫差を無視する。

イ：在庫数量を倉庫だけの属性にして商品差を無視する。

ウ：商品と倉庫のN:M関係を在庫エンティティで表し、在庫数量をその属性にする。

エ：商品と倉庫を1:1に制限する。

答え・解説 正答：ウ

ア：倉庫ごとの数量を表せない。

イ：商品ごとの数量を表せない。

ウ：在庫数量は「どの商品がどの倉庫にどれだけあるか」という組合せに依存するため、関連エンティティが適する。

エ：実際の多対多要件を満たさない。

総合解説：在庫数量は「どの商品がどの倉庫にどれだけあるか」という組合せに依存するため、関連エンティティが適する。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0052 2-2 概念データモデリング > リレーションシップ・多重度 | 難易度：応用

顧客 - 契約 - サービスの関係で、契約は顧客とサービスの組合せごとに開始日・契約番号を持ち、同じ顧客が同じサービスを期間を変えて再契約できる。識別設計として最も適切な考え方はどれか。

ア：顧客とサービスの組合せだけを永久に一意として、再契約を禁止する。

イ：契約番号をサービス名に埋め込む。

ウ：開始日を顧客の氏名に追加して顧客を別人として扱う。

エ：契約自体を独立して識別し、顧客・サービスへの関係と契約期間を保持する。

答え・解説 正答：エ

ア：再契約という要件を満たさない。

イ：異なる意味の情報を混在させる。

ウ：顧客の同一性を壊す。

エ：同一顧客・同一サービスで複数回の契約事実が存在するため、契約を独立した業務事実として識別する必要がある。

総合解説：同一顧客・同一サービスで複数回の契約事実が存在するため、契約を独立した業務事実として識別する必要がある。学習ポイント：リレーションシップでは最大多重度だけでなく最小多重度、必須・任意参加、関連自体の属性を読み取る。N:Mや三項関係は業務事実を失わない形で表す。

対象：2026年度 / 基準日 2026-09-02

0053 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：基礎

顧客を「個人顧客」と「法人顧客」に分類し、共通属性は顧客に、個人固有属性と法人固有属性をそれぞれ持たせたい。最も適切な概念はどれか。

ア：スーパータイプ/サブタイプ

イ：直積

ウ：射影

エ：物理パーティションだけ

答え・解説 正答：ア

ア：共通性を上位のスーパータイプに、固有性をサブタイプに表すのが一般的である。

イ：直積は関係代数の演算である。

ウ：射影も関係代数の演算である。

エ：物理パーティションは概念分類そのものではない。

総合解説：共通性を上位のスーパータイプに、固有性をサブタイプに表すのが一般的である。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0054 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：標準

「従業員は正社員または契約社員のどちらか一方にだけ属する」というサブタイプ制約として最も適切なものはどれか。

ア：重複可能（overlapping）なサブタイプ

イ：排他的（disjoint）なサブタイプ

ウ：サブタイプを一切持たない制約

エ：直積制約

答え・解説 正答：イ

ア：重複可能では両方に属することを許す。

イ：一つのスーパータイプ・インスタンスが同時に複数サブタイプへ属せない条件は排他的である。

ウ：サブタイプ分類要件が存在する。

エ：直積はこの分類制約を表す用語ではない。

総合解説：一つのスーパータイプ・インスタンスが同時に複数サブタイプへ属せない条件は排他的である。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0055 2-2 概念データモデリング> サブタイプ・イベント・履歴 | 難易度：標準

「全ての支払は、現金支払・カード支払・振込支払のいずれかに必ず分類される」という制約として最も適切なものはどれか。

- ア：部分（partial）な特殊化
- イ：分類なし
- ウ：完全（total）な特殊化
- エ：任意の直積

答え・解説 正答：ウ

ア：部分特殊化ではどのサブタイプにも属さないインスタンスを許す。

イ：分類要件を表せない。

ウ：スーパータイプの全インスタンスが少なくとも一つのサブタイプに属する条件は完全特殊化である。

エ：直積は特殊化の完全性と無関係である。

総合解説：スーパータイプの全インスタンスが少なくとも一つのサブタイプに属する条件は完全特殊化である。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0056 2-2 概念データモデリング> サブタイプ・イベント・履歴 | 難易度：基礎

受注の「受付」「出荷」「取消」など、業務上発生した事実を時刻とともに残し、後から発生順に追跡したい。最も適切なモデル化はどれか。

- ア：受注の現在状態だけを上書きし、過去事実は保存しない。
- イ：全イベントを受注番号の文字列末尾に追記する。
- ウ：発生時刻を商品マスタに保存する。
- エ：受注イベントを表すエンティティを設け、イベント種別と発生時刻を記録する。

答え・解説 正答：エ

ア：履歴追跡要件を満たさない。

イ：識別子に履歴を埋め込むのは不適切である。

ウ：商品マスタは受注イベントの格納先ではない。

エ：発生事実をイベントとして独立記録すれば、時系列・監査・状態遷移の分析が可能になる。

総合解説：発生事実をイベントとして独立記録すれば、時系列・監査・状態遷移の分析が可能になる。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0057 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：標準

契約料金について、将来の料金改定を事前登録し、任意の日付時点で適用される料金を判定したい。最も適切な履歴設計はどれか。

ア：料金レコードに有効開始日・有効終了日などの有効期間を持たせる。

イ：常に料金を一行だけ持ち、改定時に旧料金を上書きする。

ウ：料金履歴を契約者氏名の一部として保存する。

エ：料金改定のたびに契約者を別人として登録する。

答え・解説 正答：ア

ア：有効期間を持つ履歴にすると、過去・現在・将来の任意時点で有効な料金を判定できる。

イ：過去や予約済み将来料金を失う。

ウ：氏名と料金履歴は別の意味である。

エ：契約者の同一性を壊す。

総合解説：有効期間を持つ履歴にすると、過去・現在・将来の任意時点で有効な料金を判定できる。
学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0058 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：標準

注文の状態履歴を保存する際、「いつ業務上その状態になったか」と「いつDBに記録されたか」の二種類の時刻を区別したい。最も適切な考え方はどれか。

ア：二つの時刻は必ず同一なので一つだけにする。

イ：業務上の有効時間とシステムへの記録時間を別の時間軸として保持する。

ウ：どちらの時刻も主キーの代わりに顧客名へ埋め込む。

エ：時間情報を一切持たず行順だけで履歴順を判断する。

答え・解説 正答：イ

ア：遅延登録や訂正があれば一致しない。

イ：業務上の事実発生時刻とDB登録・訂正時刻は異なり得るため、要件次第で別軸として管理する。

ウ：時刻を氏名へ埋め込むのは意味的に不適切である。

エ：関係の物理行順は業務履歴順を保証しない。

総合解説：業務上の事実発生時刻とDB登録・訂正時刻は異なり得るため、要件次第で別軸として管理する。
学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0059 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：応用

資産の所在地変更について、監査目的では「変更前の値・変更後の値・変更者・変更日時」を残し、業務目的では「各期間に有効だった所在地」を参照したい。この二要件への対応として最も適切なものはどれか。

ア：監査ログさえあれば、常に効率よく任意時点の業務状態を表現できるので他の履歴設計は不要である。

イ：現在所在地だけを上書きし、監査・履歴の両方を満たしたことにする。

ウ：監査ログと有効期間付き業務履歴は目的が異なるため、必要情報を明確に分けて設計する。

エ：変更者だけを残し、変更内容と期間は捨てる。

答え・解説 正答：ウ

ア：監査ログの粒度や形式が時点照会に最適とは限らない。

イ：過去状態と変更証跡を失う。

ウ：変更証跡と時点状態参照は似ていても目的・検索単位が異なるため、要件に応じて履歴構造を設計する。

エ：何がいつ変わったかを復元できない。

総合解説：変更証跡と時点状態参照は似ていても目的・検索単位が異なるため、要件に応じて履歴構造を設計する。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0060 2-2 概念データモデリング > サブタイプ・イベント・履歴 | 難易度：応用

会員に「一般」「学生」「法人」の種別があるが、学生であり法人担当者でもあるなど複数特性を同時に持てる要件へ将来拡張する可能性が高い。排他的サブタイプ固定案に対する最も適切な評価はどれか。

ア：将来重複しても排他的サブタイプのままで必ず表現できる。

イ：重複特性は全て会員番号の桁に埋め込めば概念モデルは不要である。

ウ：複数特性を持つ会員はデータベースに登録しない。

エ：特性が重複し得るなら、排他的分類を前提にせず、役割や資格を別の関連として表す案も検討する。

答え・解説 正答：エ

ア：同一インスタンスが複数分類へ属せず要件違反になる。

イ：識別子へ意味を過剰に埋め込むと変更に弱い。

ウ：業務対象を除外するのは要件解決にならない。

エ：分類が相互排他的でないなら、単一の排他的サブタイプ階層は業務意味と不一致になることがある。

総合解説：分類が相互排他的でないなら、単一の排他的サブタイプ階層は業務意味と不一致になることがある。学習ポイント：サブタイプは共通性と固有性を整理する手段であり、排他・重複、完全・部分の制約を業務規則に合わせる。履歴では現在値だけでなく、イベントや有効期間の要件を区別する。

対象：2026年度 / 基準日 2026-09-02

0061 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：基礎

強エンティティ「部門」から関係スキーマを作る基本的な変換として、最も適切なものはどれか。

- ア：部門の識別子を主キーとし、単純属性を列として部門表を作る。
- イ：部門を必ず二つの無関係な表に分割する。
- ウ：識別子を捨て、氏名だけを主キーにする。
- エ：部門をSQL実行計画だけで表す。

答え・解説 正答：ア

ア：強エンティティは通常、その識別子を主キーとする一つの関係へ写像し、属性を列にする。
イ：分割が必須ではない。
ウ：識別子を捨てる理由はない。
エ：実行計画はデータモデルではない。
総合解説：強エンティティは通常、その識別子を主キーとする一つの関係へ写像し、属性を列にする。
学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0062 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：標準

部門1対社員Nの関係を、関係表として実装する基本方針として最も適切なものはどれか。関係自体に属性はない。

- ア：1側の部門表に全社員番号を一つの列へカンマ区切りで持たせる。
- イ：N側の社員表に部門の主キーを外部キーとして持たせる。
- ウ：社員と部門を無関係な表のままにする。
- エ：社員表の主キーを全て部門番号と同じ値にする。

答え・解説 正答：イ

ア：複数値を一列に埋め込む設計になる。
イ：1:Nでは、N側に1側のキーを外部キーとして配置するのが基本である。
ウ：関係を保持できない。
エ：社員識別子と部門識別子は別役割である。
総合解説：1:Nでは、N側に1側のキーを外部キーとして配置するのが基本である。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0063 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：標準

学生と科目のN:M関係「履修」を関係スキーマに変換する。最も適切なものはどれか。

ア：学生表の一行に全科目コードを連結して保存する。

イ：科目表の主キーを学生番号に置き換える。

ウ：学生キーと科目キーを外部キーとして持つ履修表を設ける。

エ：N:M関係は関係データベースでは表現できない。

答え・解説 正答：ウ

ア：繰返し値を一行に押し込むことになる。

イ：科目の識別を失う。

ウ：N:Mは中間の関係を作り、双方のキーを外部キーとして持たせることで表現できる。

エ：中間表で表現可能である。

総合解説：N:Mは中間の関係を作り、双方のキーを外部キーとして持たせることで表現できる。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0064 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：基礎

社員とロッカーが任意1:1で、社員だけ存在してロッカー未割当のケースが多い。一方、ロッカーは割り当てられるなら一人の社員だけに対応する。外部キー配置案として最も適切なものはどれか。

ア：社員表の一行に複数ロッカーキーを保存する。

イ：ロッカー一台を複数社員へ同時に割り当てる前提でN:M表を必ず作る。

ウ：1:1なので外部キーや一意性制約は一切不要である。

エ：割当を表す側にNULL可能なロッカーキーを持たせ、一意性制約などで1:1を保証する案を検討する。

答え・解説 正答：エ

ア：複数ロッカーは要件に反する。

イ：N:Mではない。

ウ：1:1を関係スキーマで保証する仕組みは必要である。

エ：任意1:1は、NULL発生やアクセス方向を考慮して外部キーの側を選び、一意性で最大1を保証する設計がある。

総合解説：任意1:1は、NULL発生やアクセス方向を考慮して外部キーの側を選び、一意性で最大1を保証する設計がある。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0065 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：標準

顧客が複数の電話番号を持てる多値属性を関係スキーマへ変換する場合、最も適切なものはどれか。

ア：顧客キーと電話番号を持つ顧客電話表を別に作る。

イ：電話番号1、電話番号2、電話番号3の固定列を十分多く作る。

ウ：全電話番号を一つの列に区切り文字で保存する。

エ：顧客キーを削除して電話番号だけで顧客を識別する。

答え・解説 正答：ア

ア：可変個数の多値属性は別関係に分離し、所有者のキーと値を持たせると関係モデルで扱いやすい。

イ：上限変更に弱く、繰返し列になる。

ウ：値を個別に検索・制約しにくい。

エ：電話番号が顧客の安定した識別子とは限らない。

総合解説：可変個数の多値属性は別関係に分離し、所有者のキーと値を持たせると関係モデルで扱いやすい。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0066 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：標準

弱エンティティ「注文明細」が、受注番号と明細番号で識別される。関係スキーマとして最も適切な主キーはどれか。

ア：明細番号だけ

イ：(受注番号, 明細番号)の複合主キー

ウ：商品名だけ

エ：受注日だけ

答え・解説 正答：イ

ア：別受注で同じ明細番号があり得る。

イ：明細番号が受注内でのみ一意なら、所有者の受注番号と部分キーである明細番号を組み合わせで識別する。

ウ：商品名は複数明細で重複し得る。

エ：同日に複数明細が存在する。

総合解説：明細番号が受注内でのみ一意なら、所有者の受注番号と部分キーである明細番号を組み合わせで識別する。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0067 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：基礎

スーパータイプ「顧客」とサブタイプ「個人顧客」「法人顧客」を関係スキーマへ写像する案の一つとして適切なものはどれか。

ア：各サブタイプ表に顧客キーを持たせず、氏名だけで無関係に管理する。

イ：サブタイプ情報を全て顧客キーの文字列形式だけに埋め込む。

ウ：顧客表を作り、個人顧客表・法人顧客表は顧客キーを主キー兼外部キーとして持つ。

エ：スーパータイプとサブタイプは関係モデルへ変換できない。

答え・解説 正答：ウ

ア：上位個体との対応が保証できない。

イ：識別子の表現に業務分類を過剰に埋め込む。

ウ：上位の共通属性を顧客表に置き、下位表のキーを上位表への外部キーにもする方式は代表的な写像方法である。

エ：複数の写像戦略があり変換可能である。

総合解説：上位の共通属性を顧客表に置き、下位表のキーを上位表への外部キーにもする方式は代表的な写像方法である。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0068 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：標準

医師・患者・薬剤の三項関係「処方」に処方量がある。この概念モデルを関係スキーマへ変換する基本案として最も適切なものはどれか。

ア：三項関係を無視し、医師表に患者名と薬剤名を一つの文字列で保存する。

イ：薬剤表だけに全処方量を一つ保存する。

ウ：三項関係は必ず三つの1:1に分解すれば意味が完全に同じになる。

エ：処方表を設け、医師・患者・薬剤の各キーを外部キーとして持ち、処方量も保持する。

答え・解説 正答：エ

ア：関係の粒度が失われる。

イ：患者・医師ごとの差を表せない。

ウ：三項関係は安易に二項関係へ分解すると元の意味や制約を失うことがある。

エ：三者の組合せ自体が業務事実なので、それを表す関係を作り、各参加実体への外部キーと関係属性を持たせる。

総合解説：三者の組合せ自体が業務事実なので、それを表す関係を作り、各参加実体への外部キーと関係属性を持たせる。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0069 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：応用

履修(学生番号, 科目コード, 履修年度, 成績)で、同じ学生が同じ科目を年度を変えて再履修できる。この要件を反映する主キーとして最も適切なものはどれか。

ア：(学生番号, 科目コード, 履修年度)

イ：(学生番号, 科目コード)だけ

ウ：成績だけ

エ：履修年度だけ

答え・解説 正答：ア

ア：再履修により学生番号と科目コードの組だけでは複数行になり得るため、履修年度も識別に必要である。

イ：同じ組合せで年度違いの複数履修を表せない。

ウ：成績は重複し変更もあり得る。

エ：同年度に多数の学生・科目がある。

総合解説：再履修により学生番号と科目コードの組だけでは複数行になり得るため、履修年度も識別に必要である。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0070 2-3 論理設計・正規化 > ERモデルから関係スキーマへの変換 | 難易度：応用

1:Nの部門 - 社員関係で、「社員は所属部門を必ず持つ」という概念モデル上の必須参加を関係スキーマで強く表現したい。最も適切な設計はどれか。

ア：社員表の部門番号を自由入力文字列にし、参照制約を設けない。

イ：社員表の部門番号外部キーをNOT NULLとし、部門表を参照させる。

ウ：部門表の全社員番号を一つの列に連結する。

エ：社員表から部門番号を削除する。

答え・解説 正答：イ

ア：存在しない部門や未入力を防げない。

イ：外部キーで参照先存在を保証し、NOT NULLで所属なしを禁止すれば、必須参加を関係スキーマへ反映できる。

ウ：1:Nを繰返し値として格納する設計になる。

エ：所属関係を表現できない。

総合解説：外部キーで参照先存在を保証し、NOT NULLで所属なしを禁止すれば、必須参加を関係スキーマへ反映できる。学習ポイント：ERモデルから関係スキーマへ変換するときは、識別子をキーへ、1:NをN側の外部キーへ、N:Mや多値属性を別関係へ写像し、参加制約もNULL可否や参照制約へ反映する。

対象：2026年度 / 基準日 2026-09-02

0071 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：基礎

第1正規形（1NF）の基本的な考え方として、最も適切なものはどれか。

ア：全ての関係を必ず一列だけにする。

イ：全ての非キー属性を削除する。

ウ：各属性位置に、関係として扱う上で一つの値を保持し、繰返し群を列内や反復列で表さない。

エ：全ての表を必ず二つ以上に分解する。

答え・解説 正答：ウ

ア：一列だけにする要件ではない。

イ：非キー属性を削除する正規形ではない。

ウ：1NFでは、各属性値を単一値として扱える形にし、繰返し構造を関係として表現する。

エ：分解数は要件によって異なる。

総合解説：1NFでは、各属性値を単一値として扱える形にし、繰返し構造を関係として表現する。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0072 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

受注明細(受注番号, 商品番号, 商品名, 数量)の候補キーが(受注番号, 商品番号)で、商品番号→商品名が成立する。第2正規形の観点から最も問題となるものはどれか。

ア：数量が受注番号と商品番号の組合せで決まること

イ：候補キーが二属性であること自体

ウ：商品番号が候補キーの一部であること自体

エ：商品名が複合候補キーの一部である商品番号に部分関数従属していること

答え・解説 正答：エ

ア：組合せ全体への完全従属なら2NF上の問題ではない。

イ：複合キー自体は禁止されない。

ウ：キーの構成属性であることだけでは違反にならない。

エ：2NFでは非キー属性の候補キーへの部分関数従属を排除する。商品名は商品番号だけで決まるため問題となる。

総合解説：2NFでは非キー属性の候補キーへの部分関数従属を排除する。商品名は商品番号だけで決まるため問題となる。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0073 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

社員(社員番号, 部門番号, 部門名)で、社員番号→部門番号、部門番号→部門名が成立し、社員番号が候補キーである。第3正規形の観点で最も問題となるものはどれか。

ア：部門名が社員番号に対して部門番号を介して推移的に従属していること

イ：社員番号が単一属性の候補キーであること

ウ：部門番号が非キー属性であることだけ

エ：部門名が文字列型であること

答え・解説 正答：ア

ア：3NFでは候補キーから非キー属性への不適切な推移的依存を排除する。ここでは社員番号→部門番号→部門名がある。

イ：単一属性キー自体は問題ではない。

ウ：非キー属性が存在すること自体ではなく、従属性が問題である。

エ：データ型は3NF判定の本質ではない。

総合解説：3NFでは候補キーから非キー属性への不適切な推移的依存を排除する。ここでは社員番号→部門番号→部門名がある。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0074 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：基礎

BCNFの条件として最も適切なものはどれか。

ア：全ての属性が外部キーである。

イ：自明でない関数従属性 $X \rightarrow Y$ が成立するなら、 X はスーパーキーである。

ウ：全ての関係に関数従属性が一つもない。

エ：全ての候補キーが単一属性である。

答え・解説 正答：イ

ア：外部キーの有無はBCNFの定義ではない。

イ：BCNFは、自明でない関数従属性の決定項がスーパーキーであることを要求する。

ウ：キーから非キーへの従属性など、適切な関数従属性は存在し得る。

エ：複合候補キーも許される。

総合解説：BCNFは、自明でない関数従属性の決定項がスーパーキーであることを要求する。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0075 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

関係 $R(A,B,C)$ で候補キーが (A,B) 、関数従属性 $A \rightarrow C$ が成立する。 R を2NFへ近づける分解として最も適切なものはどれか。

ア： $R1(A)$ と $R2(B)$ だけに C を捨てる。

イ： $R1(A,B,C)$ をそのまま二つ複製する。

ウ： $R1(A,C)$ と $R2(A,B)$ に分解する。

エ： $R1(B,C)$ だけに A を捨てる。

答え・解説 正答：ウ

ア：属性 C を失い情報保存できない。

イ：依存関係と冗長性が残ったままである。

ウ： $A \rightarrow C$ という部分依存を A,C の関係へ分離し、元のキー成分 A,B の対応を別関係に保持する。

エ： A を失うため元の対応を復元できない。

総合解説： $A \rightarrow C$ という部分依存を A,C の関係へ分離し、元のキー成分 A,B の対応を別関係に保持する。
学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0076 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

社員(社員番号, 部門番号, 部門名)で社員番号 $\rightarrow$ 部門番号、部門番号 $\rightarrow$ 部門名が成立する。3NF化する分解として最も適切なものはどれか。

ア：社員(社員番号, 部門名)だけ

イ：社員(社員番号, 部門番号, 部門名)をそのまま保持するだけ

ウ：部門(部門名)だけ

エ：社員(社員番号, 部門番号)と部門(部門番号, 部門名)

答え・解説 正答：エ

ア：部門番号という業務上の対応を失う。

イ：推移的依存が残る。

ウ：社員と部門の対応を復元できない。

エ：部門番号 $\rightarrow$ 部門名を部門関係へ分離し、社員から部門番号を参照することで推移的依存と重複を減らせる。

総合解説：部門番号 $\rightarrow$ 部門名を部門関係へ分離し、社員から部門番号を参照することで推移的依存と重複を減らせる。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0077 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：基礎

関係 $R(A,B,C)$ に関数従属性 $A \rightarrow B$ 、 $B \rightarrow C$ があり、 A が候補キーである。 R の正規形について最も適切なものはどれか。各属性は単一値とする。

ア：2NFは満たし得るが、 $B \rightarrow C$ による推移的依存があるため3NFにはならない。

イ：1NFにもならない。

ウ：3NFを満たし、必ずBCNFも満たす。

エ：複合候補キーでないと2NFを満たせない。

答え・解説 正答：ア

ア： A は単一属性キーなので部分依存はないが、非キー B が非キー C を決め、 $A \rightarrow B \rightarrow C$ の推移的依存がある。

イ：単一値という前提から1NF上の問題は示されていない。

ウ： B はスーパーキーではなく、3NF上も問題となる。

エ：2NFは複合キーを必須としない。

総合解説： A は単一属性キーなので部分依存はないが、非キー B が非キー C を決め、 $A \rightarrow B \rightarrow C$ の推移的依存がある。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0078 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

関係 $R(A,B,C)$ の候補キーが AB と AC で、関数従属性 $B \rightarrow C$ と $C \rightarrow B$ が成立する。3NFとBCNFの関係について最も適切なものはどれか。

ア：3NFを満たす関係は必ずBCNFも満たす。

イ：3NFを満たしても、 B や C がスーパーキーでないためBCNFを満たさない場合がある。

ウ：BCNFは3NFより弱い条件なので、BCNFだけ満たして3NFを満たさないことがある。

エ：候補キーが複数ある関係は全て1NFにもならない。

答え・解説 正答：イ

ア：BCNFは3NFより強い正規形である。

イ： $B \rightarrow C$ では右辺 C が候補キーの構成属性（素属性）なので3NF条件を満たし得る一方、 B はスーパーキーでないためBCNFには違反し得る。

ウ：BCNFを満たせば3NFも満たす。

エ：複数候補キーの存在は1NF違反ではない。

総合解説： $B \rightarrow C$ では右辺 C が候補キーの構成属性（素属性）なので3NF条件を満たし得る一方、 B はスーパーキーでないためBCNFには違反し得る。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0079 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：応用

分解 $R \rightarrow R_1, R_2$ が無損失結合（lossless join）であることの意味として、最も適切なものはどれか。

ア：分解後は元の関係を絶対に再構成できない。

イ：分解すると必ずタプル数が半分になる。

ウ： R を R_1 と R_2 へ分解しても、適切に自然結合すれば元の情報を余計な組合せなく再構成できる。

エ：分解後の二関係は共通属性を一切持ってはならない。

答え・解説 正答：ウ

ア：これは損失分解の説明である。

イ：タプル数の単純な半減を意味しない。

ウ：無損失結合分解は、分解・再結合によって元の関係情報を正確に復元できる性質をいう。

エ：共通属性が結合の手掛かりになることが多い。

総合解説：無損失結合分解は、分解・再結合によって元の関係情報を正確に復元できる性質をいう。
学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0080 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：基礎

正規化で「関数従属性を保存する分解」を重視する主な理由として、最も適切なものはどれか。

ア：全ての問合せを必ず1表だけで実行できるようにするため

イ：外部キーを一切使わなくて済むようにするため

ウ：必ず物理I/Oが最小になるため

エ：元の主要な関数従属性を、分解後の各関係上の制約として確認しやすくするため

答え・解説 正答：エ

ア：正規化により表数が増えることもあり、全問合せが1表になるわけではない。

イ：外部キー等の制約が不要になるわけではない。

ウ：論理設計上の性質であり物理I/O最小を保証しない。

エ：依存性保存は、元の従属性を結合せず各分解関係の制約から検査・導出しやすいことに価値がある。

総合解説：依存性保存は、元の従属性を結合せず各分解関係の制約から検査・導出しやすいことに価値がある。
学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0081 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：標準

関係 $R(A,B,C)$ に $A \rightarrow B$ 、 $B \rightarrow C$ があり、 A が候補キーである。 $R1(A,B)$ と $R2(B,C)$ へ分解したときの評価として最も適切なものはどれか。

ア：共通属性 B が $R2$ のキーとなるので無損失に再結合でき、 $A \rightarrow B$ と $B \rightarrow C$ も各関係で保持できる。

イ：共通属性がある分解は必ず損失分解になる。

ウ： $B \rightarrow C$ は分解後に表現できなくなる。

エ： $A \rightarrow B$ は $R1$ に A と B があるにもかかわらず表現できない。

答え・解説 正答：ア

ア： $R1 \cap R2 = \{B\}$ で、 $B \rightarrow BC$ が成立するため二分解の無損失条件を満たす。また各従属性は対応する関係内に残る。

イ：共通属性は無損失性を支える場合がある。

ウ： $R2(B,C)$ で $B \rightarrow C$ を保持できる。

エ： $R1(A,B)$ で $A \rightarrow B$ を保持できる。

総合解説： $R1 \cap R2 = \{B\}$ で、 $B \rightarrow BC$ が成立するため二分解の無損失条件を満たす。また各従属性は対応する関係内に残る。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0082 2-3 論理設計・正規化 > 第1～第3正規形・BCNF | 難易度：応用

正規化を進めた論理モデルを性能上の理由で非正規化する場合の原則として、最も適切なものはどれか。

ア：正規化は常に性能を悪化させるので、設計開始時から全て一表にまとめる。

イ：冗長性による更新不整合リスクを認識し、測定された性能要件と整合性維持策を根拠に限定的に行う。

ウ：非正規化すれば整合性制約は不要になる。

エ：検索性能を上げるには主キーを削除するのが最も一般的である。

答え・解説 正答：イ

ア：正規化と性能の関係はワークロードによって異なる。

イ：非正規化はトレードオフであり、論理的な冗長性と更新コストを増やす可能性があるため、要件と計測に基づく判断が必要である。

ウ：冗長データほど整合性維持策が重要になる。

エ：主キー削除は識別性を損なう。

総合解説：非正規化はトレードオフであり、論理的な冗長性と更新コストを増やす可能性があるため、要件と計測に基づく判断が必要である。学習ポイント：正規化は関数従属性に基づいて冗長性と更新異状を減らす。1NF、2NF、3NF、BCNFの違いに加え、分解では無損失結合性と依存性保存も確認する。

対象：2026年度 / 基準日 2026-09-02

0083 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：基礎

同じ部門名が社員ごとの行に重複保存されているため、部門名変更時に複数行を更新しなければならない。この問題として最も適切なものはどれか。

- ア：挿入時異状
- イ：削除時異状
- ウ：更新時異状
- エ：参照整合性が常に自動回復する現象

答え・解説 正答：ウ

- ア：新規事実を単独で登録できない問題ではない。
- イ：削除で別事実まで失う問題ではない。
- ウ：同一事実が重複し、一部だけ更新すると不整合になる問題は更新時異状である。
- エ：自動回復ではなく冗長性に起因する異状である。

総合解説：同一事実が重複し、一部だけ更新すると不整合になる問題は更新時異状である。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0084 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：標準

社員と部門情報を一表にまとめ、社員が一人もない新設部門は社員番号がないため登録できない。この問題として最も適切なものはどれか。

- ア：更新時異状
- イ：削除時異状
- ウ：無損失結合
- エ：挿入時異状

答え・解説 正答：エ

- ア：既存値の重複更新問題ではない。
- イ：削除による情報消失ではない。
- ウ：無損失結合は分解の性質である。
- エ：ある事実（部門）を別の事実（社員）がないため登録できないのは挿入時異状である。

総合解説：ある事実（部門）を別の事実（社員）がないため登録できないのは挿入時異状である。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0085 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：標準

社員と部門情報を一表に保存し、ある部門の最後の社員行を削除したら、その部門名や所在地まで失われた。この問題として最も適切なものはどれか。

- ア：削除時異状
- イ：挿入時異状
- ウ：更新時異状
- エ：直積

答え・解説 正答：ア

ア：一つの事実を削除した結果、保持すべき別の事実まで失うのが削除時異状である。

イ：新規登録の問題ではない。

ウ：複数箇所更新の不整合問題ではない。

エ：直積は関係代数演算である。

総合解説：一つの事実を削除した結果、保持すべき別の事実まで失うのが削除時異状である。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0086 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：基礎

正規化による分解を行うとき、無損失結合性を確認する主な目的はどれか。

- ア：分解後の表を必ず物理的に同じディスクへ置くため
- イ：分解した関係を結合したとき、元になかった余計なタプルや情報損失を避けて元の関係を再現できるようにする。
- ウ：全ての外部キーを削除するため
- エ：関係の属性名を全て一文字にするため

答え・解説 正答：イ

ア：物理配置の条件ではない。

イ：無損失性は、論理分解によって元の情報内容を損なわないための重要な条件である。

ウ：外部キー削除が目的ではない。

エ：命名規則とは無関係である。

総合解説：無損失性は、論理分解によって元の情報内容を損なわないための重要な条件である。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0087 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：標準

関係 $R(A,B,C)$ を $R1(A,B)$ と $R2(A,C)$ に分解する。 A が R の候補キーで $A \rightarrow B$ 、 $A \rightarrow C$ が成立する場合、この分解の評価として最も適切なものはどれか。

ア：共通属性 A があるので必ず損失分解になる。

イ： A が候補キーであることは結合結果と無関係である。

ウ： A で結合すれば元の対応を復元できるため、無損失分解となる。

エ： $R1$ と $R2$ を結合すると必ず全ての B と C の任意組合せが発生する。

答え・解説 正答：ウ

ア：共通属性がキーであることは無損失性を支える。

イ： A の一意決定性が余計な組合せを防ぐ。

ウ： A が各 A 値に対して B と C を一意に決めるため、 $R1$ と $R2$ を A で結合しても不要な組合せが生じない。

エ： A ごとに B,C が一意なので任意組合せにはならない。

総合解説： A が各 A 値に対して B と C を一意に決めるため、 $R1$ と $R2$ を A で結合しても不要な組合せが生じない。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0088 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：標準

売上明細に商品名を冗長保存して検索時の結合を減らす非正規化を採用する場合、最も重要な追加設計はどれか。

ア：商品マスタの主キーを削除する。

イ：更新処理は考慮せず検索だけを設計する。

ウ：冗長値は絶対に不整合にならないと仮定する。

エ：商品名変更時に明細側の冗長値をどう同期・整合させるかを明確にする。

答え・解説 正答：エ

ア：識別性を失うため適切でない。

イ：更新時異状のリスクを無視している。

ウ：冗長データは更新経路によって不整合になり得る。

エ：非正規化では同一事実を複数箇所に持つため、更新整合性の維持方法が重要になる。

総合解説：非正規化では同一事実を複数箇所に持つため、更新整合性の維持方法が重要になる。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0089 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：応用

受注明細に「明細金額=数量×単価」を保存する案について、正規化・整合性の観点から最も適切な評価はどれか。

ア：導出可能な値を保存すると冗長性が生じるため、性能や履歴要件があるなら更新整合性を保証する仕組みと合わせて判断する。

イ：計算可能な値は必ず保存しなければならない。

ウ：導出値を保存すれば数量や単価は不要になる。

エ：保存した明細金額は数量・単価変更後も更新しなくてよい。

答え・解説 正答：ア

ア：導出値の保存は読取り性能などに利点があり得る一方、基礎値との不整合リスクを生む。

イ：必須ではなくトレードオフである。

ウ：数量や単価自体が業務上必要な事実であり得る。

エ：基礎値変更後に再計算しなければ不整合になる。

総合解説：導出値の保存は読取り性能などに利点があり得る一方、基礎値との不整合リスクを生む。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0090 2-3 論理設計・正規化 > 更新時異状・分解・非正規化 | 難易度：応用

高頻度の参照処理で、正規化された5表の結合がボトルネックだと測定された。非正規化を検討する際の最も適切な進め方はどれか。

ア：測定せず最初から5表を一つに統合する。

イ：まず索引・SQL・統計情報・物理設計などを確認し、それでも要件を満たせない場合に、冗長性と更新コストを定量評価して非正規化する。

ウ：参照性能向上のため全ての制約を削除する。

エ：更新処理の頻度や不整合リスクは評価しない。

答え・解説 正答：イ

ア：根拠のない統合は更新異状や保守性低下を招く。

イ：非正規化は論理設計の整合性を複雑化させるため、他の性能改善策と比較し、実測とトレードオフに基づいて行うべきである。

ウ：制約削除は整合性を損なう。

エ：読み書き双方のコストを評価する必要がある。

総合解説：非正規化は論理設計の整合性を複雑化させるため、他の性能改善策と比較し、実測とトレードオフに基づいて行うべきである。学習ポイント：正規化不足では挿入・更新・削除の各異状が起こり得る。分解は情報を失わないことが重要で、非正規化は性能上の根拠と整合性維持策を伴う例外的な判断として扱う。

対象：2026年度 / 基準日 2026-09-02

0091 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：基礎

社員コードを設計する。将来、所属部門が変わる可能性が高い場合のコード設計として、最も適切なものはどれか。

- ア：社員コードの先頭2桁を必ず現在の部門コードにして、異動時には社員コードも変更する。
- イ：社員名の先頭4文字を社員コードとして利用する。
- ウ：入社年と現在の役職を連結し、役職変更時にはコードを再発行する。
- エ：社員コード自体には所属部門を埋め込まず、所属は別属性・別関係として管理する。

答え・解説 正答：エ

- ア：部門をコードに埋め込むと、部門変更が識別子変更に波及しやすい。
 - イ：氏名は重複・変更があり得るため、一意で安定した識別子として不適切である。
 - ウ：役職も変更される属性であり、コードの安定性を損なう。
 - エ：所属のように変更される属性を識別子に埋め込むと、異動のたびに識別子変更や参照先更新が必要になりやすい。
- 総合解説：所属のように変更される属性を識別子に埋め込むと、異動のたびに識別子変更や参照先更新が必要になりやすい。コード設計では、一意性だけでなく、意味の変化に対する安定性、桁数、採番・廃止・再利用方針も検討する。

対象：2026年度 / 基準日 2026-09-02

0092 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：基礎

データモデリングにおける「ドメイン」の説明として、最も適切なものはどれか。

- ア：テーブルを格納する物理ディスク領域だけを表すもの。
- イ：ある属性が取り得る値の集合を、型だけでなく意味、範囲、形式、単位などの制約とともに定義したもの。
- ウ：一つのエンティティに存在する属性数を表すもの。
- エ：複数テーブルを結合するSQL構文の種類を表すもの。

答え・解説 正答：イ

- ア：これは表領域など物理記憶域の説明であり、ドメインではない。
 - イ：ドメインは、単なる物理データ型より広く、業務上の意味と許容値の規則を含む概念である。
 - ウ：属性数は関係の次数に関する概念であり、ドメインではない。
 - エ：SQL構文の分類はドメインの定義ではない。
- 総合解説：ドメインは、単なる物理データ型より広く、業務上の意味と許容値の規則を含む概念である。

対象：2026年度 / 基準日 2026-09-02

0093 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：基礎

全社データ標準で「顧客番号」「得意先コード」「取引先ID」がシステムごとに同じ概念を指している。最も優先すべき対応はどれか。

ア：概念の定義を確認して標準名称を一つ定め、別名は同義語としてリポジトリに対応付ける。

イ：各システムの名称をそのまま全社標準として併記し、意味の定義は行わない。

ウ：名称が違うので、必ず別のデータ項目として扱う。

エ：各システムで別々の略称へ変更し、標準名称や同義語対応は管理しない。

答え・解説 正答：ア

ア：同一概念は意味を統一した上で標準名称を定め、既存の別名との対応をメタデータとして管理するのが適切である。

イ：表記だけ併記して意味を定義しなければ、同義かどうかを機械的・組織的に判断できない。

ウ：名称差だけでは概念差とは限らず、業務定義を確認する必要がある。

エ：略称を増やすだけでは表記ゆれを解消できず、標準化と追跡可能性が悪化する。

総合解説：同一概念に複数名称がある状態を放置すると、連携・分析・変更時の誤解を招く。標準名称と同義語管理が有効である。

対象：2026年度 / 基準日 2026-09-02

0094 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：標準

商品分類コード「07」を廃止した5年後、別の新分類に同じ「07」を再利用する案が出た。過去データを長期保存し分析する要件がある。最も適切な対応はどれか。

ア：原則として同じコード値を別意味で再利用せず、廃止状態と有効期間を管理して新分類には新しいコードを付与する。

イ：コード桁数を節約するため、廃止後は必ず即時再利用する。

ウ：過去データも一括して新分類の意味へ読み替える。

エ：表示名だけ変えればコードの意味は常に一意なので、再利用しても問題ない。

答え・解説 正答：ア

ア：同じ値を異なる意味で再利用すると、過去データの「07」が旧分類か新分類かをコードだけで判定できなくなり、履歴解釈を誤るおそれがある。

イ：桁数節約よりも意味の一意性と履歴整合性が重要である。

ウ：過去の事実を後から別意味へ書き換えると、監査・分析結果を損なう。

エ：コード値と意味の対応が時期によって変われば、表示名変更だけでは曖昧さを解消できない。

総合解説：同じ値を異なる意味で再利用すると、過去データの「07」が旧分類か新分類かをコードだけで判定できなくなり、履歴解釈を誤るおそれがある。

対象：2026年度 / 基準日 2026-09-02

0095 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：標準

複数システムで「重量」というデータ項目を共有する。一方はkg、他方はgで記録していた。全社標準化で最も適切な対応はどれか。

ア：両方とも数値型なので、単位は定義せず利用者の判断に任せる。

イ：値をすべて文字列型にすれば単位差は解消する。

ウ：小数点以下の桁数だけを同じにすれば意味も統一される。

エ：標準の単位と変換規則をデータ定義に明記し、必要なら単位そのものもメタデータとして管理する。

答え・解説 正答：エ

ア：型だけでは業務上の意味の差を表せず、誤集計の原因になる。

イ：文字列化してもkgとgの意味差は残る。

ウ：桁数は表現上の性質であり、単位の違いを解消しない。

エ：数値の型が同じでも単位が異なれば同じ値の意味は一致しない。ドメイン定義には単位・尺度・変換規則を含めるべきである。

総合解説：数値の型が同じでも単位が異なれば同じ値の意味は一致しない。ドメイン定義には単位・尺度・変換規則を含めるべきである。

対象：2026年度 / 基準日 2026-09-02

0096 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：標準

外部機関が提供する業種コードを社内の顧客マスタで利用する。外部コードは将来改定される可能性がある。設計として最も適切なものはどれか。

ア：外部業種コードを顧客の主キーにし、外部改定時には全参照を更新する。

イ：社内の安定した識別子と外部業種コードを分け、外部コードの版・有効期間を含めて対応関係を管理する。

ウ：外部コードは改定されるため、データベースには一切保存しない。

エ：現在のコードだけを上書きし、旧コードとの対応関係は残さない。

答え・解説 正答：イ

ア：分類コードは顧客そのものの識別子ではなく、改定時の影響範囲が大きくなる。

イ：外部コード体系の改定を内部主キーに直接波及させないように、内部識別子と外部分類コードを分離し、マッピングを管理するのが堅牢である。

ウ：外部分類を業務で利用するなら、保存・対応付けが必要になる。

エ：旧版との対応を失うと過去データの解釈や移行確認が困難になる。

総合解説：外部コード体系の改定を内部主キーに直接波及させないように、内部識別子と外部分類コードを分離し、マッピングを管理するのが堅牢である。

対象：2026年度 / 基準日 2026-09-02

0097 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：標準

「都道府県コード」を20表で利用しており、各表が個別に値チェックを実装している。標準化の観点から最も適切な改善はどれか。

ア：各表で異なる値体系を持たせ、入力時に利用者が変換する。

イ：値チェックをすべて削除して自由入力にする。

ウ：都道府県コードの意味・形式・許容値を共通定義し、各表・アプリケーションが同じ標準を参照するようにする。

エ：都道府県名を主キーにすれば、標準定義は不要になる。

答え・解説 正答：ウ

ア：同一概念の値体系が異なると、連携時に変換と不整合が増える。

イ：自由入力は表記揺れや不正値を増やす。

ウ：同一概念の制約を共通化すると、システム間の不一致を減らし、変更時の影響把握や保守を容易にできる。

エ：名称は変更・表記差があり得るため、標準化が不要になるわけではない。

総合解説：同一概念の制約を共通化すると、システム間の不一致を減らし、変更時の影響把握や保守を容易にできる。

対象：2026年度 / 基準日 2026-09-02

0098 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：標準

既存DBでは物理列名が「CUST_NO」、新しい全社標準の論理名が「顧客番号」と定義された。段階的移行を行う場合の考え方として最も適切なものはどれか。

ア：論理名を定めた時点で、全ての物理列名を無条件に即時変更する。

イ：物理名が英字なら論理名や定義は不要である。

ウ：論理名と物理名の対応は開発者が覚えればよく、管理対象にしない。

エ：論理名・定義・物理名の対応をリポジトリで管理し、物理名変更の必要性は影響範囲を評価して決める。

答え・解説 正答：エ

ア：物理名変更はSQL、プログラム、ETL等に影響し、段階移行では評価が必要である。

イ：物理名だけでは業務意味や同義語関係を十分に表せない。

ウ：属人的な記憶では変更・引継ぎに耐えず、メタデータとして管理すべきである。

エ：名称標準化の目的は概念の意味を統一することであり、既存物理名を即時改名すること自体が目的ではない。対応関係を管理し、変更影響を評価する。

総合解説：名称標準化の目的は概念の意味を統一することであり、既存物理名を即時改名すること自体が目的ではない。対応関係を管理し、変更影響を評価する。

対象：2026年度 / 基準日 2026-09-02

0099 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：応用

取引先コードが「地域2桁＋業種2桁＋連番4桁」で構成されている。地域変更や業種変更が頻繁に起き、コード変更のたびに多数の外部キー更新が必要である。最も適切な再設計方針はどれか。

ア：地域と業種の桁数を増やして、より多くの意味をコードへ埋め込む。

イ：取引先の識別子には変更されにくい内部IDを用い、地域・業種は独立した属性または関係として管理する。

ウ：外部キー制約を削除すればコード変更の問題は解決する。

エ：変更時には旧取引先を削除し、新しい取引先として登録し直す。

答え・解説 正答：イ

ア：意味を増やすほど変更時の影響が大きくなる。

イ：識別子に変化しやすい分類情報を埋め込むと、分類変更が参照整合性や連携先へ波及する。識別と分類を分離するのが基本である。

ウ：制約削除は不整合を許すだけで、識別子設計の問題を解決しない。

エ：同一取引先の履歴連続性が失われ、取引履歴の追跡に支障が出る。

総合解説：識別子に変化しやすい分類情報を埋め込むと、分類変更が参照整合性や連携先へ波及する。識別と分類を分離するのが基本である。

対象：2026年度 / 基準日 2026-09-02

0100 2-4 データ定義・標準化 > コード・ドメイン・命名規則 | 難易度：応用

3事業部が独自の商品分類コードを持ち、全社分析では共通分類が必要だが、各事業部のローカル分類も業務上残す必要がある。最も適切な設計はどれか。

ア：全社分析を優先し、各事業部コードを直ちに削除する。

イ：全社共通分類を定義し、各事業部コードとの多対一・一対多を含み得る対応関係と有効期間を別管理する。

ウ：各事業部コードを単純に文字列連結し、それを全社分類とする。

エ：共通分類を設けず、集計のたびに担当者が手作業で読み替える。

答え・解説 正答：イ

ア：ローカル業務要件を失う可能性があり、移行影響も大きい。

イ：ローカル分類を無理に一つへ置換せず、共通分類とのマッピングを管理すれば、業務継続と全社分析の双方を支えられる。対応が時期で変わるなら有効期間も必要になる。

ウ：連結は分類概念の統合ではなく、意味の対応関係を表さない。

エ：手作業は再現性・統制・保守性に乏しい。

総合解説：ローカル分類を無理に一つへ置換せず、共通分類とのマッピングを管理すれば、業務継続と全社分析の双方を支えられる。対応が時期で変わるなら有効期間も必要になる。

対象：2026年度 / 基準日 2026-09-02

0101 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：基礎

「顧客:生年月日」のデータ型、桁数、意味、更新責任部門、参照元テーブルなどを記録した情報は、何に該当するか。

ア：トランザクションデータ

イ：ログバックアップ

ウ：インデックスページ

エ：メタデータ

答え・解説 正答：エ

ア：顧客の取引そのものを表すデータではない。

イ：バックアップは障害回復用の複製であり、項目定義そのものではない。

ウ：インデックスページは物理アクセス構造であり、データ定義情報とは異なる。

エ：メタデータは「データについてのデータ」であり、データ項目の意味・構造・由来・管理情報などを記述する。

総合解説：メタデータは「データについてのデータ」であり、データ項目の意味・構造・由来・管理情報などを記述する。

対象：2026年度 / 基準日 2026-09-02

0102 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：基礎

データ定義標準化におけるリポジトリの利用目的として、最も適切なものはどれか。

ア：業務データの全件を必ずリポジトリに格納して本番DBを廃止する。

イ：CPUキャッシュを拡張してSQL性能だけを向上させる。

ウ：データ項目、コード、モデル、関係、定義などのメタデータを集約し、共有・変更管理に利用する。

エ：障害時のデータ復旧だけを目的にログを保管する。

答え・解説 正答：ウ

ア：リポジトリは通常、本番業務データそのものの代替ではない。

イ：これはハードウェア/性能の話であり、リポジトリの役割ではない。

ウ：リポジトリは、設計成果物やデータ定義に関するメタデータを一元的に管理し、標準化・影響分析・再利用を支援する。

エ：障害回復ログとは目的が異なる。

総合解説：リポジトリは、設計成果物やデータ定義に関するメタデータを一元的に管理し、標準化・影響分析・再利用を支援する。

対象：2026年度 / 基準日 2026-09-02

0103 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度: 基礎

次のうち、一般にマスターデータとして扱うのが最も適切なものはどれか。

ア: 1件ごとの受注明細

イ: 毎日のWebアクセスログ

ウ: 一時的なSQL実行結果

エ: 商品コード、商品名、標準分類など、複数取引から参照される商品基本情報

答え・解説 正答: エ

ア: 受注明細は業務イベントごとに発生するトランザクションデータである。

イ: アクセスログはイベント記録であり、一般的な商品マスタとは性質が異なる。

ウ: 一時結果は永続的な業務基礎情報ではない。

エ: マスターデータは、取引から繰り返し参照される比較的安定した業務上の基礎情報を表す。

総合解説: マスターデータは、取引から繰り返し参照される比較的安定した業務上の基礎情報を表す。

対象: 2026年度 / 基準日 2026-09-02

0104 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度: 標準

CRMと販売システムで同一顧客が別ID・表記揺れで重複登録されている。マスターデータ統合で最も適切な進め方はどれか。

ア: 照合ルールで同一実体候補を特定し、統合後の代表レコードと各システムIDの対応関係を管理する。

イ: IDが異なるレコードは必ず別顧客として扱う。

ウ: 全レコードを文字列連結し、一つの巨大な顧客レコードにする。

エ: CRM側を無条件に正として、販売側データは確認せず削除する。

答え・解説 正答: ア

ア: マスタ統合では、単純なID一致だけでなく、重複照合・統合ルール・元システムとの対応を管理する必要がある。

イ: 異なるシステムでは同一実体に別IDが付くため、ID不一致だけで別人とは判断できない。

ウ: 連結では重複解消や属性競合の解決にならない。

エ: 正本候補の選定には品質・更新責任・最新性などのルールが必要である。

総合解説: マスタ統合では、単純なID一致だけでなく、重複照合・統合ルール・元システムとの対応を管理する必要がある。

対象: 2026年度 / 基準日 2026-09-02

0105 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：標準

基幹DBの「売上金額」の定義変更が、DWHやBIレポートへどこまで影響するか確認したい。最も役立つメタデータはどれか。

ア：売上金額列の現在のデータ型だけを記録したメタデータ

イ：DWHの現在行数だけを記録した容量メタデータ

ウ：データの生成元、変換処理、転送先を追跡できるデータリネージュ情報

エ：BIレポートの画面上の表示順だけを記録したメタデータ

答え・解説 正答：ウ

ア：データ型は影響分析の一要素だが、どの処理・データセット・レポートへ伝播するかは追跡できない。

イ：行数は容量把握には有用だが、データの流れや依存関係を示さない。

ウ：データリネージュは生成元から変換・転送・利用先までの依存関係を追跡でき、定義変更の影響範囲把握に適する。

エ：表示順は画面設計情報であり、上流から下流へのデータ依存関係を示さない。

総合解説：リネージュは、データがどこから来てどの処理を経てどこへ流れるかを示すため、変更影響分析や原因追跡に有効である。

対象：2026年度 / 基準日 2026-09-02

0106 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：標準

「顧客区分」のコード桁数を1桁から2桁へ拡張する。変更前にリポジトリで確認すべき内容として最も適切なものはどれか。

ア：その項目を作成した担当者の勤続年数だけ

イ：その項目を参照するテーブル、ビュー、ETL、API、帳票などの依存関係

ウ：DBサーバの設置フロアだけ

エ：過去1時間のCPU使用率だけ

答え・解説 正答：イ

ア：担当者属性だけでは技術的依存関係を把握できない。

イ：データ定義変更はDB内部だけでなく、連携・変換・出力にも影響する。リポジトリに依存関係を持たせると影響範囲を系統的に確認できる。

ウ：設置場所は項目定義変更の依存先を示さない。

エ：CPU使用率は性能情報であり、項目依存関係の代替にはならない。

総合解説：データ定義変更はDB内部だけでなく、連携・変換・出力にも影響する。リポジトリに依存関係を持たせると影響範囲を系統的に確認できる。

対象：2026年度 / 基準日 2026-09-02

0107 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：標準

国コードや通貨コードのように、業務データから繰り返し参照される小規模な値集合を管理する場合、最も適切な考え方はどれか。

ア：値の意味はシステムごとに自由に変更してよい。

イ：一覧が小さいので、ソースコード内にだけ埋め込み定義書は不要とする。

ウ：古い値は履歴を残さず、常に同じコードへ別の意味を上書きする。

エ：値、意味、出典、版、有効期間を明確にし、利用システム間で一貫して参照できるよう管理する。

答え・解説 正答：エ

ア：意味がシステムごとに異なると連携結果が一致しない。

イ：埋め込みだけでは変更管理・共有・監査が困難になる。

ウ：コード意味の上書きは過去データの解釈を壊す。

エ：参照データは値集合そのものが業務判断・連携の基準になるため、コードと意味、版、有効期間、出典などの統制が重要である。

総合解説：参照データは値集合そのものが業務判断・連携の基準になるため、コードと意味、版、有効期間、出典などの統制が重要である。

対象：2026年度 / 基準日 2026-09-02

0108 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：標準

顧客住所を5システムがそれぞれ更新できるため、どの住所が最新か分からなくなる問題が起きている。最も適切な改善方針はどれか。

ア：属性ごとの更新責任と正本を定め、更新競合・配信タイミング・同期失敗時の処理を設計する。

イ：5システムすべてを常時自由更新のままにし、最新判定は利用者に任せる。

ウ：更新日時だけを削除し、競合の存在を見えなくする。

エ：住所は毎回手入力すればよいのでマスタ管理を廃止する。

答え・解説 正答：ア

ア：マスタの一貫性には、単にコピーするだけでなく、どこが正本か、誰が更新できるか、どのように配信・再同期するかの統制が必要である。

イ：複数の無統制更新元がある限り競合は解消しない。

ウ：情報を削除しても競合原因は残る。

エ：重複入力と表記揺れが増え、むしろ整合性が低下する。

総合解説：マスタの一貫性には、単にコピーするだけでなく、どこが正本か、誰が更新できるか、どのように配信・再同期するかの統制が必要である。

対象：2026年度 / 基準日 2026-09-02

0109 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：応用

データ項目定義をリポジトリで一元管理しているが、担当者が定義を上書きでき、変更理由や旧定義が残らない。改善として最も適切なものはどれか。

ア：最新定義だけあればよいので、旧版は自動削除する。

イ：定義変更を口頭連絡だけにし、リポジトリ更新を停止する。

ウ：定義の版、変更者、変更日時、承認状態、変更理由を記録し、旧版を追跡できるようにする。

エ：担当者名だけをファイル名に入れれば、変更内容の履歴は不要である。

答え・解説 正答：ウ

ア：過去データがどの定義で作られたか追えなくなる。

イ：口頭だけでは共有・監査・再現性を確保できない。

ウ：メタデータも変更管理の対象である。版と承認履歴を残すことで、過去時点の定義確認、監査、障害原因分析が可能になる。

エ：変更者だけでは何をなぜ変えたかを説明できない。

総合解説：メタデータも変更管理の対象である。版と承認履歴を残すことで、過去時点の定義確認、監査、障害原因分析が可能になる。

対象：2026年度 / 基準日 2026-09-02

0110 2-4 データ定義・標準化 > メタデータ・リポジトリ・マスターデータ | 難易度：応用

顧客マスタ統合で、氏名は本人確認済みCRM、請求先住所は請求システム、メールアドレスは会員サイトが最も信頼できる。統合レコードの設計として最も適切なものはどれか。

ア：最初に登録されたシステムの全属性を無条件に採用する。

イ：属性ごとに信頼できるソースと優先順位・競合解決ルールを定義し、統合値と出典を保持する。

ウ：各システムの値をすべて同じ列へ連結して保存する。

エ：競合が発生した属性はランダムに一つ選ぶ。

答え・解説 正答：イ

ア：属性ごとの信頼性差を無視している。

イ：「一つのシステムを全部正」とする必要はなく、属性単位で品質・責任・最新性を評価して正本値を決めることができる。出典を残すと追跡性も高まる。

ウ：連結は一意な業務値を決定せず、利用側に曖昧さを残す。

エ：再現性と業務根拠がなく、品質統制にならない。

総合解説：「一つのシステムを全部正」とする必要はなく、属性単位で品質・責任・最新性を評価して正本値を決めることができる。出典を残すと追跡性も高まる。

対象：2026年度 / 基準日 2026-09-02

0111 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：基礎

「注文数量は1以上でなければならない」というビジネスルールをDBで表現する方法として、最も直接的なものはどれか。

- ア：注文数量に対するCHECK制約を定義する。
- イ：注文数量列にインデックスだけを作成する。
- ウ：注文テーブルを毎日バックアップする。
- エ：注文数量列の名前をQTYに変更する。

答え・解説 正答：ア

ア：単一行の属性値に対する範囲条件は、RDBMSが対応していればCHECK制約で直接表現できる。
イ：インデックスは主にアクセス効率のための構造であり、値域制約そのものではない。
ウ：バックアップは障害回復策であり、入力値の妥当性制約ではない。
エ：列名変更はルールを強制しない。
総合解説：単一行の属性値に対する範囲条件は、RDBMSが対応していればCHECK制約で直接表現できる。

対象：2026年度 / 基準日 2026-09-02

0112 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：基礎

業務フローでは「返品受付」で返品理由を必ず登録するが、データモデルには返品理由を保持する項目・関連が存在しない。最も適切な評価はどれか。

- ア：業務プロセスが必要とする情報をモデルが表現できておらず、モデルの完全性に問題がある。
- イ：業務フローとデータモデルは無関係なので問題ない。
- ウ：返品理由は画面に表示できればDBに保持しなくてよい。
- エ：項目が不足しているので、返品エンティティ自体を削除する。

答え・解説 正答：ア

ア：概念・論理モデルは業務プロセスで必要な情報を保持できる必要がある。業務フローとの突合せは欠落の発見に有効である。
イ：シラバスでもビジネスプロセスと概念モデルの整合性確認が求められる。
ウ：履歴・分析・監査要件があるなら画面表示だけでは保持できない。
エ：不足に対してエンティティ削除は逆方向の対応である。
総合解説：概念・論理モデルは業務プロセスで必要な情報を保持できる必要がある。業務フローとの突合せは欠落の発見に有効である。

対象：2026年度 / 基準日 2026-09-02

0113 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：基礎

業務ルールが「会員は0件以上の注文を持てる。各注文は必ず1人の会員に属する」である。モデルの検証観点として最も適切なものはどれか。

ア：会員→注文を必ず1、注文→会員を0..*にする。

イ：両方向とも必ず1にする。

ウ：会員→注文は0..*、注文→会員は1となっているか確認する。

エ：多重度は性能だけに関係するため確認しない。

答え・解説 正答：ウ

ア：ルールの方向を逆にしている。

イ：会員が注文0件を許す要件と矛盾する。

ウ：多重度と必須性はビジネスルールをモデルへ写したものであり、ルールとモデル表現を対応付けて検証する。

エ：多重度は業務上可能な関連数を表す重要な意味制約である。

総合解説：多重度と必須性はビジネスルールをモデルへ写したものであり、ルールとモデル表現を対応付けて検証する。

対象：2026年度 / 基準日 2026-09-02

0114 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：標準

通常ケースでは正しく動く受注モデルを検証する。業務には「一部返品」「注文後の配送先変更」「同一商品を複数配送先へ分割」という例外がある。最も適切な検証方法はどれか。

ア：通常ケースが通れば例外はモデル検証対象から除外する。

イ：通常ケースに加え、境界・例外シナリオをモデル上でたどり、必要な事実と関係を保持できるか確認する。

ウ：例外はすべて自由記述の備考列に格納する前提にする。

エ：例外がある場合は正規化を行わない。

答え・解説 正答：イ

ア：例外が実業務要件ならモデルで扱える必要がある。

イ：代表例だけではモデルの欠落を見逃しやすい。例外・変更・分割などを具体的な業務シナリオでウォークスルーすると妥当性を検証しやすい。

ウ：備考への押込みは構造化・制約・検索性を失う。

エ：例外の存在は正規化を不要にする理由ではない。

総合解説：代表例だけではモデルの欠落を見逃しやすい。例外・変更・分割などを具体的な業務シナリオでウォークスルーすると妥当性を検証しやすい。

対象：2026年度 / 基準日 2026-09-02

0115 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：標準

「請求残高」は請求額と入金額から算出できる。データモデルで請求残高を保存する案を検証する際、最も重要な観点はどれか。

ア：算出規則、更新タイミング、再計算方法を明確にし、元データとの不整合を防げるか確認する。

イ：導出値は保存すれば必ず正確になるので、更新規則は不要である。

ウ：請求額と入金額を削除して残高だけ残す。

エ：導出値を保存する場合は主キーを不要にできる。

答え・解説 正答：ア

ア：導出値を保存する場合、元データ更新と同期しないと整合性が崩れる。性能上の利点がある場合でも、更新ルールを明確にすべきである。

イ：保存だけでは正確性は保証されず、同期ルールが必要である。

ウ：元の事実を失うと取引内訳や再計算ができない。

エ：導出値の有無と主キーの必要性は別問題である。

総合解説：導出値を保存する場合、元データ更新と同期しないと整合性が崩れる。性能上の利点がある場合でも、更新ルールを明確にすべきである。

対象：2026年度 / 基準日 2026-09-02

0116 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：標準

申請の状態は「下書き→申請中→承認済み」の順で進み、「承認済み→下書き」へは戻れないというルールがある。モデル検証として最も適切な対応はどれか。

ア：状態列を文字列にすればどの遷移でも正しい。

イ：許可される状態と遷移を明文化し、状態履歴や更新処理が禁止遷移を許さないか検証する。

ウ：承認済みになったら申請データを削除する。

エ：状態遷移は画面デザインだけの問題なのでデータモデルでは扱わない。

答え・解説 正答：イ

ア：データ型だけでは遷移規則を強制できない。

イ：属性の値集合だけでなく、値の変更順序もビジネスルールである。状態遷移表などで許可・禁止遷移を定義すると検証しやすい。

ウ：削除すると履歴・監査情報を失う。

エ：状態は業務データの意味に関わるため、モデル・更新規則の検証対象である。

総合解説：属性の値集合だけでなく、値の変更順序もビジネスルールである。状態遷移表などで許可・禁止遷移を定義すると検証しやすい。

対象：2026年度 / 基準日 2026-09-02

0117 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：標準

「与信残高 + 今回受注額が、外部与信サービスから取得した最新限度額を超える場合は受注を確定できない」というルールがある。最も適切な設計判断はどれか。

ア：列のNOT NULL制約だけでこのルールを完全に実現する。

イ：ルールを業務仕様として明示し、トランザクション整合性と外部情報の取得条件を含め、適切なアプリケーション/DB処理で一貫して検証する。

ウ：外部限度額は毎回無視し、受注額だけ保存する。

エ：ルールは実装せず、担当者の記憶だけに任せる。

答え・解説 正答：イ

ア：NOT NULLは欠損禁止であり、集計・外部限度額との比較を表さない。

イ：すべてのビジネスルールを単純な列CHECK制約で表せるとは限らない。複数行集計や外部情報を伴う場合は、実装層と一貫性条件を設計する必要がある。

ウ：外部条件を無視すると業務ルールを満たさない。

エ：属人的運用では一貫性・監査性を確保できない。

総合解説：すべてのビジネスルールを単純な列CHECK制約で表せるとは限らない。複数行集計や外部情報を伴う場合は、実装層と一貫性条件を設計する必要がある。

対象：2026年度 / 基準日 2026-09-02

0118 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：応用

営業部では「顧客」を契約先法人と定義するが、サポート部では問い合わせを行う個人担当者も「顧客」と呼んでいる。概念モデル検証で最も適切な対応はどれか。

ア：両部門と用語の意味・識別単位・関係を確認し、「法人」と「担当者」など必要な概念を分け、定義を合意する。

イ：営業部の定義だけを採用し、サポート部には説明しない。

ウ：同じ言葉なので必ず一つのエンティティとして統合する。

エ：用語差はデータモデルに影響しないため確認しない。

答え・解説 正答：ア

ア：同名異義の業務用語をそのまま一つのエンティティに押し込むと、属性・多重度・識別子が矛盾しやすい。利用者を交えた定義合意が必要である。

イ：利用者間の意味差を放置するとモデルの妥当性が損なわれる。

ウ：名称一致だけでは概念一致を意味しない。

エ：概念の粒度・識別対象が異なればモデル構造に直接影響する。

総合解説：同名異義の業務用語をそのまま一つのエンティティに押し込むと、属性・多重度・識別子が矛盾しやすい。利用者を交えた定義合意が必要である。

対象：2026年度 / 基準日 2026-09-02

0119 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：応用

販売部門は「一つの契約に複数請求先を登録できる」と要求し、経理部門は「契約ごとに請求先は必ず一つ」と要求している。モデル担当者の最も適切な対応はどれか。

ア：最初に聞いた販売部門の要求だけを採用する。

イ：両方を同時に満たすため、多重度を定義しない。

ウ：経理部門の要求をDB制約にし、販売部門には知らせない。

エ：両要求の背景・適用条件・例外を確認し、ルールを合意したうえで多重度や履歴をモデルへ反映する。

答え・解説 正答：エ

ア：要求の優先順位や背景を確認していない。

イ：多重度を曖昧にするとルール不整合を残すだけである。

ウ：隠れた制約は業務障害を招く。関係者の合意が必要である。

エ：対立要求を技術者が勝手に一方へ決めるのではなく、業務上の条件を明確化して合意し、その結果をモデルに反映する必要がある。

総合解説：対立要求を技術者が勝手に一方へ決めるのではなく、業務上の条件を明確化して合意し、その結果をモデルに反映する必要がある。

対象：2026年度 / 基準日 2026-09-02

0120 2-4 データ定義・標準化 > ビジネスルール・モデル検証 | 難易度：標準

顧客には個人と法人があり、「顧客種別が法人の場合だけ法人番号を必須とする」というルールがある。モデル・制約の検証として最も適切なものはどれか。

ア：顧客種別と法人番号の組合せ条件を明文化し、法人では法人番号が必須、個人では不要となることを検証する。

イ：法人番号列を全顧客でNOT NULLにし、個人には架空の番号を入れる。

ウ：法人番号列を常にNULL可にし、法人でも未入力を許す。

エ：顧客種別を削除し、法人番号の有無だけを利用者が目視判断する。

答え・解説 正答：ア

ア：必須性が全行一律ではなく条件付きの場合は、単純なNOT NULLだけでは表現できない。種別と属性の整合条件を業務ルールとして定義し、適切な制約や処理で保証する。

イ：個人に存在しない値を強制すると意味品質を損なう。

ウ：法人に必須というルールを保証できない。

エ：業務上の種別情報と条件を曖昧にし、一貫した検証が困難になる。

総合解説：必須性が全行一律ではなく条件付きの場合は、単純なNOT NULLだけでは表現できない。種別と属性の整合条件を業務ルールとして定義し、適切な制約や処理で保証する。

対象：2026年度 / 基準日 2026-09-02

0121 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：基礎

業務要件文から概念モデルを作るときの基本姿勢として、最も適切なものはどれか。

ア：要件文に現れる名詞はすべて独立エンティティにする。

イ：動詞だけをエンティティにし、名詞はすべて属性にする。

ウ：名詞候補を洗い出した後、独立して識別・管理する必要がある実体か、属性や値にすべきかを業務ルールで検討する。

エ：名詞候補を一つの巨大なエンティティへまとめ、関係や識別子の検討は後回しにする。

答え・解説 正答：ウ

ア：名詞抽出は候補発見に使えるが、値・属性・帳票名なども含まれるため、全てをエンティティにするのは不適切である。

イ：動詞は関係やイベント候補になることがあるが、名詞を一律に属性とすることはできない。

ウ：概念モデルでは業務上独立して識別・管理すべき対象かどうかを業務ルールと照合して判断する。

エ：異なる実体を一つに混在させると、識別子・関係・制約が不明確になりモデルの意味が失われる。

総合解説：名詞抽出は候補発見には有効だが、全名詞がエンティティになるわけではない。識別単位、ライフサイクル、関係、業務ルールを確認する。

対象：2026年度 / 基準日 2026-09-02

0122 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：基礎

「学生は複数の講座を受講でき、講座には複数の学生が参加する。受講ごとに申込日と成績を持つ」という要件を表す最も適切なモデルはどれか。

ア：学生に講座名1個だけを属性として持たせる。

イ：学生と講座の間に「受講」を関連エンティティとして置き、申込日と成績を受講の属性にする。

ウ：講座に学生名をカンマ区切り文字列で保持する。

エ：学生と講座を無関係な二つのエンティティとして扱う。

答え・解説 正答：イ

ア：複数受講を表現できない。

イ：学生と講座は多対多であり、関係自体に属性があるため、受講を関連エンティティとして表すのが自然である。

ウ：複数值を一行へ詰め込むと構造化・制約・検索性が低下する。

エ：受講という業務事実を表せない。

総合解説：学生と講座は多対多であり、関係自体に属性があるため、受講を関連エンティティとして表すのが自然である。

対象：2026年度 / 基準日 2026-09-02

0123 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：基礎

取引先は名称変更があり、同名企業も存在する。取引先エンティティの識別子として最も適切な考え方はどれか。

- ア：名称とは別に、一意で安定した取引先IDを定義する。
- イ：取引先名だけを主識別子にする。
- ウ：最新の電話番号だけを主識別子にする。
- エ：担当営業者名を主識別子にする。

答え・解説 正答：ア

ア：識別子は可能な限り一意・安定している必要がある。表示名は変更・重複があり得るため、識別子と分離するのが適切である。

イ：名称は重複・変更があり得る。

ウ：電話番号は変更・共有があり得る。

エ：担当者は異動・変更があるため、取引先そのものの識別には不安定である。

総合解説：識別子は可能な限り一意・安定している必要がある。表示名は変更・重複があり得るため、識別子と分離するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0124 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：基礎

在庫管理で「入庫」「出庫」が発生するたびに数量、日時、理由、担当者を記録し、後から残高の根拠を追跡したい。最も適切なモデルはどれか。

- ア：商品マスタの在庫数だけを上書きし、入出庫履歴は保存しない。
- イ：理由と担当者を商品名に連結して保存する。
- ウ：入庫と出庫をDBには保存せず、口頭で管理する。
- エ：入出庫を在庫移動のイベントとして記録し、商品・倉庫などと関連付ける。

答え・解説 正答：エ

ア：現在値だけでは、いつ・なぜ変化したかを追跡できない。

イ：商品名に別概念を埋め込むのは不適切である。

ウ：履歴・監査・残高根拠の要件を満たさない。

エ：日時や理由を伴う業務イベントは、単に現在残高へ上書きするだけでなく、独立した事実として保持すると追跡性を確保できる。

総合解説：日時や理由を伴う業務イベントは、単に現在残高へ上書きするだけでなく、独立した事実として保持すると追跡性を確保できる。

対象：2026年度 / 基準日 2026-09-02

0125 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

要件は「注文には注文日・顧客・配送先があり、1注文に1件以上の商品明細がある。明細には商品・数量・受注単価がある」である。最も適切なモデルはどれか。

ア：注文テーブルに商品1～商品100まで固定列を作る。

イ：商品名・数量・単価をカンマ区切りで一行に保存する。

ウ：注文明細だけを作り、顧客や注文日を明細ごとに無制御で重複保存する。

エ：注文と注文明細を分け、注文1件に対して注文明細1件以上の関係を持たせる。

答え・解説 正答：エ

ア：最大件数が固定され、未使用列や拡張性の問題が生じる。

イ：複数値を一行に詰めると構造化された操作・制約が困難になる。

ウ：注文共通情報の重複と更新不整合を招きやすい。

エ：注文全体に共通する属性と商品ごとの繰返し属性を分けることで、1対多の業務構造を適切に表せる。

総合解説：注文全体に共通する属性と商品ごとの繰返し属性を分けることで、1対多の業務構造を適切に表せる。

対象：2026年度 / 基準日 2026-09-02

0126 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

組織エンティティについて「各組織は最上位を除き、一つの上位組織を持つ。上位組織は複数の下位組織を持てる」とする。最も自然なモデルはどれか。

ア：組織が組織自身を参照する親子の再帰リレーションシップを定義する。

イ：階層レベルごとに「部」「課」「係」という固定テーブルを必ず別々に作る。

ウ：上位組織名を自由記述文字列としてだけ保存する。

エ：階層情報はデータではないので一切保持しない。

答え・解説 正答：ア

ア：同一種類の実体間に階層関係があるため、組織エンティティの自己参照関係として表せる。

イ：階層レベルが増減する要件では固定テーブルが硬直化しやすい。

ウ：名称だけでは参照整合性や組織変更を追跡しにくい。

エ：組織階層を利用する要件があるならデータモデルで表現する必要がある。

総合解説：同一種類の実体間に階層関係があるため、組織エンティティの自己参照関係として表せる。

対象：2026年度 / 基準日 2026-09-02

0127 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

契約には「営業担当者」と「承認担当者」がそれぞれ1人ずつ存在し、どちらも社員エンティティを参照する。モデルとして最も適切なものはどれか。

ア：社員を営業社員と承認社員という完全に別の実体へ必ず分割する。

イ：契約に社員IDを一つだけ持たせ、役割は記録しない。

ウ：契約から社員への二つの関係を定義し、それぞれ営業担当・承認担当という役割名を付ける。

エ：担当者名を一行に二人分連結して保存する。

答え・解説 正答：ウ

ア：同一社員が両役割を担う可能性もあり、実体を不必要に分割する必要はない。

イ：二つの役割を区別して保持できない。

ウ：同じエンティティ間でも業務上の役割が異なる関係は、別のリレーションシップとして意味を明示する。

エ：構造化・参照整合性・役割識別ができない。

総合解説：同じエンティティ間でも業務上の役割が異なる関係は、別のリレーションシップとして意味を明示する。

対象：2026年度 / 基準日 2026-09-02

0128 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

商品マスタの標準単価は将来変更されるが、過去注文では受注時の単価で請求額を再現する必要がある。最も適切なモデルはどれか。

ア：商品マスタの現在単価とは別に、注文明細へ受注時に確定した単価を保持する。

イ：過去注文も常に商品マスタの最新単価で再計算する。

ウ：単価は商品名の一部として文字列保存する。

エ：単価変更時に過去の注文明細をすべて最新単価へ更新する。

答え・解説 正答：ア

ア：取引時点で確定した値は、その時点の業務事実である。現在のマスタ値だけを参照すると、後日の変更で過去取引を再現できなくなる。

イ：契約・請求の過去事実が変わってしまう。

ウ：値の意味と型を損ない、計算に不向きである。

エ：過去時点の確定値を失い、再現性を損なう。

総合解説：取引時点で確定した値は、その時点の業務事実である。現在のマスタ値だけを参照すると、後日の変更で過去取引を再現できなくなる。

対象：2026年度 / 基準日 2026-09-02

0129 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

「一人の医師は0件以上の診療を担当できる。各診療は必ず1人の担当医師を持つ」という要件をモデルへ反映する場合、最も適切なものはどれか。

ア：医師1人に対して診療1、診療1件に対して医師0..*とする。

イ：両方向を0..*とする。

ウ：医師1人に対して診療0..*、診療1件に対して医師1とする。

エ：両方向を必ず1とする。

答え・解説 正答：ウ

ア：方向と条件が逆である。

イ：診療が医師なしでもよいことになり、要件に反する。

ウ：「0件以上」と「必ず1人」という自然言語上の条件を、関係の多重度と必須性へ正確に対応付ける。

エ：医師が診療0件でもよい要件を表せない。

総合解説：「0件以上」と「必ず1人」という自然言語上の条件を、関係の多重度と必須性へ正確に対応付ける。

対象：2026年度 / 基準日 2026-09-02

0130 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：標準

注文番号は全社で一意で、各注文内の明細番号は1から採番される。「注文明細」を業務上識別するキーとして最も自然な候補はどれか。

ア：注文番号と明細番号の組合せ

イ：明細番号だけ

ウ：商品番号だけ

エ：注文日だけ

答え・解説 正答：ア

ア：明細番号は注文内でのみ一意なので、注文番号と組み合わせることで明細を一意に識別できる。実装では別の代理キーを追加する選択肢もあるが、業務上の一意性はこの組合せで表せる。

イ：明細番号1は複数注文に存在し得る。

ウ：同一商品が複数注文・複数明細に現れ得る。

エ：同日に複数注文・複数明細があるため一意にならない。

総合解説：明細番号は注文内でのみ一意なので、注文番号と組み合わせることで明細を一意に識別できる。実装では別の代理キーを追加する選択肢もあるが、業務上の一意性はこの組合せで表せる。

対象：2026年度 / 基準日 2026-09-02

0131 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：応用

要件では「顧客」は法人契約先を指す。一方、問い合わせ履歴は法人内の複数担当者ごとに電話番号・メールアドレスを管理する必要がある。顧客エンティティに担当者情報を1組だけ持たせる案の評価として最も適切なものはどれか。

ア：担当者は法人の属性なので、担当者1～担当者50の固定列を作る。

イ：担当者情報を全て自由記述の備考欄へ格納する。

ウ：問い合わせ履歴を削除すれば顧客エンティティだけで十分になる。

エ：法人と担当者は識別単位・多重度・ライフサイクルが異なるため、担当者を別エンティティとして法人に関連付けるのが適切である。

答え・解説 正答：エ

ア：繰返し列となり拡張性・正規化に問題がある。

イ：構造・識別・参照ができず業務要件を満たしにくい。

ウ：必要な業務履歴を削除してモデルに合わせるのは本末転倒である。

エ：同一法人に複数担当者が存在し、担当者ごとに連絡先や履歴を持つなら、法人の単一属性としては粒度が合わない。

総合解説：同一法人に複数担当者が存在し、担当者ごとに連絡先や履歴を持つなら、法人の単一属性としては粒度が合わない。

対象：2026年度 / 基準日 2026-09-02

0132 2-5 モデリング実践 > 業務要件からのモデル作成 | 難易度：応用

会議室予約について、(1)予約は会議室1室と申請者1人を持つ、(2)同一会議室では時間帯が重複する予約を許さない、(3)予約変更履歴を追跡する、という要件がある。最も適切な設計方針はどれか。

ア：予約を独立エンティティとして会議室・申請者に関連付け、開始/終了時刻を保持し、重複禁止ルールと変更履歴の保持方式を別途設計する。

イ：会議室エンティティに「現在の予約者」だけを1項目保存する。

ウ：予約時刻を会議室名に連結して保存する。

エ：重複予約は利用者が気を付ければよいので、ルールとして定義しない。

答え・解説 正答：ア

ア：識別対象、関連、多重度、時間制約、履歴という複数の業務ルールをモデルと制約へ分解して表現する必要がある。

イ：複数予約や将来予約・履歴を表現できない。

ウ：名称と予約事実を混在させ、構造化された時間条件を扱えない。

エ：重要な業務制約を属人的運用に任せると一貫性を確保できない。

総合解説：識別対象、関連、多重度、時間制約、履歴という複数の業務ルールをモデルと制約へ分解して表現する必要がある。

対象：2026年度 / 基準日 2026-09-02

0133 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：基礎

顧客の会員ランクについて「いつからいつまでゴールドだったか」を問い合わせたい。履歴設計として最も直接的な方法はどれか。

ア：顧客マスタの会員ランクを常に上書きし、過去値は捨てる。

イ：ランク名に日付を文字列連結する。

ウ：会員ランク履歴に適用開始日と適用終了日を持たせる。

エ：ランク変更のたびに顧客IDを変更する。

答え・解説 正答：ウ

ア：上書きだけでは過去時点の状態を再現できない。

イ：検索・制約・期間判定が困難になる。

ウ：値が有効であった期間を表すには、値とともに有効期間を保持する方法が分かりやすい。

エ：識別子変更は同一顧客の連続性を壊す。

総合解説：値が有効であった期間を表すには、値とともに有効期間を保持する方法が分かりやすい。

対象：2026年度 / 基準日 2026-09-02

0134 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：基礎

「住所変更は4月1日から有効だったが、システムへの登録は4月3日に行われた」という事実を二つの時間軸で管理する場合、4月1日は何を表すか。

ア：DBへ記録された時刻だけ

イ：バックアップ取得時刻

ウ：業務上その値が有効である時刻（有効時刻）

エ：SQL実行計画作成時刻

答え・解説 正答：ウ

ア：登録が4月3日なら、記録時刻とは一致しない。

イ：バックアップ時刻とは無関係である。

ウ：業務世界で値が成立する期間と、DBに記録された期間は別の時間軸として扱える。4月1日は住所が業務上有効になった時点である。

エ：実行計画の生成時刻とは無関係である。

総合解説：業務世界で値が成立する期間と、DBに記録された期間は別の時間軸として扱える。4月1日は住所が業務上有効になった時点である。

対象：2026年度 / 基準日 2026-09-02

0135 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：基礎

顧客住所の「現在住所」と「過去住所」を両方参照する要件がある。最も不適切な設計はどれか。

ア：住所履歴を別表で保持し、有効期間を管理する。

イ：住所レコードに有効開始・終了を持たせ、現在行を判定できるようにする。

ウ：変更イベントを履歴として保持し、必要に応じて時点状態を再構成できるようにする。

エ：顧客マスタの住所を上書きするだけで、変更前住所を一切保存しない。

答え・解説 正答：エ

ア：履歴要件を満たす代表的な方法である。

イ：期間付き履歴として過去・現在を表現できる。

ウ：イベント履歴から時点状態を再構成する設計も要件次第で成立する。

エ：過去住所の参照要件があるのに上書きだけでは履歴を失うため不適切である。

総合解説：過去住所の参照要件があるのに上書きだけでは履歴を失うため不適切である。

対象：2026年度 / 基準日 2026-09-02

0136 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：標準

社員の所属履歴で「同一社員は同一時点に一つの主所属だけを持つ」というルールがある。最も重要な整合性チェックはどれか。

ア：所属名が50文字以内であることだけ。

イ：全社員の入社日が同一であること。

ウ：同一社員について主所属の有効期間が互いに重複しないこと。

エ：所属履歴件数が必ず1件であること。

答え・解説 正答：ウ

ア：文字長は同時点一所属というルールを保証しない。

イ：入社日の一致は無関係である。

ウ：同一時点に主所属が一つというルールは、同一社員の有効期間の重複禁止として表現できる。

エ：異動がある社員は履歴が複数件になるため要件と矛盾する。

総合解説：同一時点に主所属が一つというルールは、同一社員の有効期間の重複禁止として表現できる。

対象：2026年度 / 基準日 2026-09-02

0137 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：標準

有効期間を開始日時と終了日時で管理する際、境界の重複・空白を避けるために最も重要な設計事項はどれか。

ア：開始日時と終了日時のデータ型をシステムごとに自由にする。

イ：終了日時を含むか含まないかなど、期間境界の規則を全体で統一して定義する。

ウ：終了日時の意味は利用者ごとに解釈してもらう。

エ：期間判定は行わず、最新行をランダムに選ぶ。

答え・解説 正答：イ

ア：型や精度が異なると比較が難しくなる。

イ：期間を「開始日時以上かつ終了日時未満」として統一するなど、境界規則を明示すると隣接期間の重複や空白を判定しやすい。

ウ：境界解釈が異なると同じ時点の判定が不一致になる。

エ：再現性と正確性を満たさない。

総合解説：期間を「開始日時以上かつ終了日時未満」として統一するなど、境界規則を明示すると隣接期間の重複や空白を判定しやすい。

対象：2026年度 / 基準日 2026-09-02

0138 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：標準

消費税率マスタに適用開始日を持たせ、売上日の税率を決定する。10月1日に税率変更がある場合の設計として最も適切なものはどれか。

ア：売上日が含まれる有効期間の税率を選択できるよう、税率の履歴を保持する。

イ：税率マスタは常に1行だけにして、変更日に過去値を上書きする。

ウ：売上日とは無関係に、検索時点の最新税率を過去売上にも適用する。

エ：税率変更時に過去売上の税額をすべて再計算して上書きする。

答え・解説 正答：ア

ア：時点依存するマスタ値は、現在値だけでなく適用期間を持たせることで過去・将来の取引に対応できる。

イ：過去取引の再現に必要な旧税率を失う。

ウ：取引時点の法的・業務的条件を再現できない。

エ：過去確定取引を現在ルールで書き換えることになり得る。

総合解説：時点依存するマスタ値は、現在値だけでなく適用期間を持たせることで過去・将来の取引に対応できる。

対象：2026年度 / 基準日 2026-09-02

0139 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：標準

契約の開始日を「4月1日」と登録したが、4月10日に入力誤りが判明し「3月25日」が正しかったと訂正した。業務上の有効期間とシステム上の訂正履歴の両方を追跡したい。最も適切な考え方はどれか。

ア：開始日を3月25日に上書きし、4月1日だった記録は完全に削除する。

イ：開始日は変更せず、備考欄に「誤り」とだけ書く。

ウ：訂正のたびに契約IDを新規発行し、元契約との関係は残さない。

エ：業務上いつ有効だったかと、DBにいつその情報が記録されていたかを別の時間軸で保持する。

答え・解説 正答：エ

ア：監査上、いつどの値が記録されていたかを失う。

イ：正しい業務値を構造化して扱えず、時点問い合わせが困難である。

ウ：同一契約の継続性と訂正関係が失われる。

エ：後日訂正を扱うには、有効時刻だけでなく記録・訂正の時間軸を分けると、「当時システムが何を知っていたか」も追跡できる。

総合解説：後日訂正を扱うには、有効時刻だけでなく記録・訂正の時間軸を分けると、「当時システムが何を知っていたか」も追跡できる。

対象：2026年度 / 基準日 2026-09-02

0140 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：標準

商品の価格は1日に複数回変更される可能性があり、「注文時刻時点の価格」を再現する必要がある。価格履歴の有効期間を日付だけで管理する案の評価として最も適切なものはどれか。

ア：日付だけでも1日に何回変更しても必ず一意に判定できる。

イ：日内に複数変更がある要件を表せないため、必要な精度の日時で有効期間を管理すべきである。

ウ：時刻を使うと正規化違反になるので禁止すべきである。

エ：価格変更が多い場合は履歴自体を保存しない方がよい。

答え・解説 正答：イ

ア：日付だけでは同日内の順序や有効区間を識別できない。

イ：履歴の時間粒度は業務イベントの粒度以上に細くなければ、同一日の複数状態を区別できない。

ウ：日時利用と正規化違反は直接関係しない。

エ：再現要件がある以上、履歴削除は要件違反になる。

総合解説：履歴の時間粒度は業務イベントの粒度以上に細くなければ、同一日の複数状態を区別できない。

対象：2026年度 / 基準日 2026-09-02

0141 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：応用

契約履歴は契約ごとに有効期間を持ち、請求は請求日時点で有効な契約版を参照する必要がある。設計上の最も重要な性質はどれか。

ア：契約履歴の全行が同じ開始日時を持つこと。

イ：任意の請求時点に対して対象契約の有効版が一意に決まり、期間の重複や意図しない空白がないこと。

ウ：請求日時を保存しないこと。

エ：契約版を毎回ランダムに選べること。

答え・解説 正答：イ

ア：同一開始では期間が競合しやすく、要件に合わない。

イ：時点参照では、単なる契約ID一致だけでなく、その時点に該当する版が一意に存在する期間整合性が必要である。

ウ：時点判定に必要な情報を失う。

エ：一意に決まらない設計は再現性がない。

総合解説：時点参照では、単なる契約ID一致だけでなく、その時点に該当する版が一意に存在する期間整合性が必要である。

対象：2026年度 / 基準日 2026-09-02

0142 2-5 モデリング実践 > 履歴・時系列データ設計 | 難易度：応用

顧客の現在ステータスは非常に高頻度で参照される一方、全変更履歴を監査用に永久保存する必要がある。設計判断として最も適切なものはどれか。

ア：必ず現在表だけにして履歴は削除する。

イ：必ず履歴表だけにして現在状態を求める索引・設計は考えない。

ウ：現在値の参照経路と履歴保存を分ける案を含め、更新原子性・重複データの整合性・監査要件を評価して方式を決める。

エ：現在値と履歴値が不一致でも、参照先が違うので問題ない。

答え・解説 正答：ウ

ア：監査要件を満たさない。

イ：高頻度参照要件を無視している。履歴表だけでも設計次第だが性能評価が必要である。

ウ：現在表＋履歴表は性能面で有効な場合があるが、二重管理の整合性が必要である。単一期間表にも利点があり、要件と更新方式に基づくトレードオフ判断が必要である。

エ：二重管理する場合は整合性が最重要であり、不一致は業務障害になる。

総合解説：現在表＋履歴表は性能面で有効な場合があるが、二重管理の整合性が必要である。単一期間表にも利点があり、要件と更新方式に基づくトレードオフ判断が必要である。

対象：2026年度 / 基準日 2026-09-02

0143 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：基礎

販売システムの「得意先」と請求システムの「請求顧客」が、調査の結果、同じ法人単位を指すと判明した。統合モデルで最も適切な対応はどれか。

ア：名称が違うので必ず別エンティティとして残す。

イ：両方の名称を連結した長いエンティティ名だけを作り、定義は確認しない。

ウ：定義と識別条件を合わせた上で共通の法人概念へ統合し、各システム名称との対応を管理する。

エ：一方のシステムを削除するまで統合は不可能である。

答え・解説 正答：ウ

ア：名称差だけでは概念差とは判断できない。

イ：名称連結は意味の統合ではない。

ウ：名称が異なっても同じ実体を表すなら、統合モデルでは共通概念に整理し、元名称をメタデータとして対応付けるのが適切である。

エ：システム統合とデータ概念統合は別であり、対応付けは可能である。

総合解説：名称が異なっても同じ実体を表すなら、統合モデルでは共通概念に整理し、元名称をメタデータとして対応付けるのが適切である。

対象：2026年度 / 基準日 2026-09-02

0144 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：基礎

二つのシステムに「契約」というエンティティがある。一方は顧客との販売契約、他方は仕入先との購買契約を表す。最も適切な統合判断はどれか。

ア：名前が同じなので無条件に一つの契約エンティティへ統合する。

イ：名称が同じなら属性差は無視する。

ウ：両方の主キー値が偶然一致する行だけを同じ契約とみなす。

エ：名称一致だけで統合せず、意味・識別子・属性・関係・ライフサイクルを比較して別概念か判断する。

答え・解説 正答：エ

ア：意味の異なる概念を混在させる危険がある。

イ：属性差は概念粒度や業務ルールの差を示す可能性がある。

ウ：別システムのID一致は同一実体を保証しない。

エ：同じ名称でも業務意味が異なる場合がある。統合では定義と構造を確認し、同名異義語による誤統合を避ける。

総合解説：同じ名称でも業務意味が異なる場合がある。統合では定義と構造を確認し、同名異義語による誤統合を避ける。

対象：2026年度 / 基準日 2026-09-02

0145 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：標準

システムAは売上を「伝票単位」、システムBは「明細単位」で管理している。統合DWH向けモデルを設計する際の最も適切な進め方はどれか。

ア：列名が同じものをそのままUNIONすればよい。

イ：明細を無条件に削除して全て伝票単位にする。

ウ：粒度差はデータ型が同じなら無視できる。

エ：各モデルの粒度を明示し、必要な分析粒度を決めた上で、集約・分解可能性とキーの対応規則を設計する。

答え・解説 正答：エ

ア：列名一致は行の意味・粒度一致を保証しない。

イ：必要な明細分析能力を失う可能性がある。

ウ：粒度は型ではなく、行が表す業務事実の単位である。

エ：粒度が異なるデータをそのまま結合すると二重計上や欠落を生みやすい。まず1行が何を表すかという粒度を揃える必要がある。

総合解説：粒度が異なるデータをそのまま結合すると二重計上や欠落を生みやすい。まず1行が何を表すかという粒度を揃える必要がある。

対象：2026年度 / 基準日 2026-09-02

0146 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：標準

顧客を、システムAは顧客番号、システムBは会員番号で識別し、番号体系に共通性がない。統合モデルで最も適切な方法はどれか。

ア：数値が偶然同じ番号は同一顧客とみなす。

イ：Aの顧客番号をBへ上書きし、旧会員番号は捨てる。

ウ：統合顧客IDを設け、各システムのローカルIDとの対応表を管理する。

エ：氏名だけで全顧客を一意に識別する。

答え・解説 正答：ウ

ア：別体系の同値は同一実体を意味しない。

イ：追跡性や既存参照を失う。

ウ：ローカルキーを保持したまま共通識別子へマッピングすると、元システムとの追跡性を保ちつつ統合できる。

エ：同姓同名や名称変更があり、氏名だけでは不十分である。

総合解説：ローカルキーを保持したまま共通識別子へマッピングすると、元システムとの追跡性を保ちつつ統合できる。

対象：2026年度 / 基準日 2026-09-02

0147 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：標準

「商品コード」を8桁から12桁へ変更する。DB列だけ変更すればよいとする判断が不十分な理由として、最も適切なものはどれか。

ア：列長変更は物理DBにしか影響しないことが規格で保証されているから。

イ：ビュー、索引、外部キー、ETL、API、ファイル連携、帳票など、同項目を利用する依存先にも影響し得るから。

ウ：コードが長くなると必ず正規化違反になるから。

エ：桁数変更時には全データを削除しなければならないから。

答え・解説 正答：イ

ア：実際には多くの依存コンポーネントへ波及し得る。

イ：データモデル変更は実装・連携・運用へ波及する。リポジトリやリネージュを使って依存関係を確認し、変更計画を立てる必要がある。

ウ：桁数変更そのものは正規化の問題ではない。

エ：データ削除は通常不要で、移行・互換性を設計する。

総合解説：データモデル変更は実装・連携・運用へ波及する。リポジトリやリネージュを使って依存関係を確認し、変更計画を立てる必要がある。

対象：2026年度 / 基準日 2026-09-02

0148 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：標準

顧客テーブルに法人属性と担当者属性が混在していたため、「法人」と「担当者」に分割する。変更計画として最も適切なものはどれか。

ア：新テーブルを作れば既存データは自動的に正しく分割される。

イ：旧テーブルを先に削除してから移行方法を考える。

ウ：新しい識別子・関係を定義し、既存行をどう法人と担当者へ分解するか、参照元・API・帳票を含めて移行と互換性を設計する。

エ：法人と担当者と同じ主キーを無条件に割り当てれば関係設計は不要になる。

答え・解説 正答：ウ

ア：業務的な分解規則がなければ自動的に正しくは分けられない。

イ：先に削除するとロールバック・検証が困難になる。

ウ：論理モデルの分割は既存データの変換だけでなく、参照関係や利用インタフェースにも影響するため、移行規則と依存先を一体で検討する。

エ：識別子共有だけでは多重度や所属関係を表せない。

総合解説：論理モデルの分割は既存データの変換だけでなく、参照関係や利用インタフェースにも影響するため、移行規則と依存先を一体で検討する。

対象：2026年度 / 基準日 2026-09-02

0149 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：応用

新ルールとして「契約には必ず責任者が1人必要」を追加し、責任者IDをNOT NULLにした。しかし既存契約の20%は責任者未登録である。最も適切な変更手順はどれか。

ア：既存データを確認せず直ちにNOT NULLを追加する。

イ：既存データの補完・例外方針と移行時点を決め、アプリ側の入力変更と整合を取った後に制約を有効化する。

ウ：責任者IDがない行は説明なく全て削除する。

エ：新ルールは文書にだけ書き、DB・アプリには反映しない。

答え・解説 正答：イ

ア：既存不整合によりDDLが失敗するか、運用停止を招く可能性がある。

イ：新制約が既存データに適合するかを事前確認し、データ移行とアプリ変更を同期しなければ、制約追加で更新・移行が失敗する。

ウ：業務データ消失につながり、正当な移行策ではない。

エ：強制されないルールは一貫した品質を保証しにくい。

総合解説：新制約が既存データに適合するかを事前確認し、データ移行とアプリ変更を同期しなければ、制約追加で更新・移行が失敗する。

対象：2026年度 / 基準日 2026-09-02

0150 2-5 モデリング実践 > モデル統合・変更影響 | 難易度：応用

グループ3社で顧客モデルを統合する。氏名・住所・連絡先は共通だが、1社だけに「会員ランク」が存在する。統合モデルの方針として最も適切なものはどれか。

ア：1社だけの属性なので会員ランク要件を削除する。

イ：全社で会員ランクを必須属性にし、他2社には架空の値を入れる。

ウ：会社ごとに顧客モデルを完全分離し、共通定義を一切持たない。

エ：共通概念・共通属性を標準化し、会員ランクは適用範囲を明示したローカル拡張または関連モデルとして管理する。

答え・解説 正答：エ

ア：正当な業務要件を失う。

イ：存在しない概念へ架空値を入れると意味品質を損なう。

ウ：共通項目まで重複定義となり、統合・標準化の効果を失う。

エ：統合モデルは全社共通部分と固有要件を区別することで、無理な共通化や大量の無意味なNULLを避けつつ標準化できる。

総合解説：統合モデルは全社共通部分と固有要件を区別することで、無理な共通化や大量の無意味なNULLを避けつつ標準化できる。

対象：2026年度 / 基準日 2026-09-02