

0001 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：基礎

受注明細には「注文数量は1以上でなければならない」と記載されている。データモデルへ反映するビジネスルールとして最も適切なものはどれか。

ア：受注明細の注文数量に、1以上という値域制約を設ける。

イ：受注明細を注文ごとに別テーブルへ分割する。

ウ：注文数量を商品マスタへ移す。

エ：注文数量を文字列型にして入力時に自由記述させる。

答え・解説 正答：ア

ア：業務要件が直接示しているのは注文数量の許容範囲なので、属性の値域制約として表現するのが適切である。

イ：表分割は値域制約そのものを表現しない。

ウ：注文数量は受注明細ごとの事実であり商品固有属性ではない。

エ：自由記述化すると1以上という制約をDB側で保証しにくくなる。

総合解説：業務要件が直接示しているのは注文数量の許容範囲なので、属性の値域制約として表現するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0002 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：基礎

会議室予約システムで「同じ会議室を同じ時間帯に二重予約してはならない」という要件がある。モデル上で最も重視すべき制約はどれか。

ア：予約者IDを必ず一意にする。

イ：会議室と予約時間帯の重複を禁止する制約を設ける。

ウ：会議室名を予約ごとに変更する。

エ：予約終了時刻だけを主キーにする。

答え・解説 正答：イ

ア：同一利用者が複数予約することはあり得るため要件と一致しない。

イ：禁止したいのは同一資源の時間帯重複なので、会議室と時間区間に対する排他・重複禁止ルールが必要である。

ウ：名称変更は二重予約の防止にならない。

エ：終了時刻単独では同じ会議室・時間帯の重複を識別できない。

総合解説：禁止したいのは同一資源の時間帯重複なので、会議室と時間区間に対する排他・重複禁止ルールが必要である。

対象：2026年度 / 基準日 2026-09-02

0003 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：基礎

人事システムで「在籍中の社員は必ず一つの所属部門を持つ。部門には社員が一人もいない場合がある」とする。最も適切な多重度の解釈はどれか。

ア：社員から部門は0件以上、部門から社員は必ず1件である。

イ：社員と部門は常に一対一である。

ウ：社員から部門は必須の多対一、部門から社員は0件以上である。

エ：社員と部門の関係は不要で、部門名を社員名に埋め込む。

答え・解説 正答：ウ

ア：社員側の所属必須条件と逆である。

イ：一つの部門には複数社員が所属できる。

ウ：各社員はちょうど1部門に所属し、部門側は0人以上の社員を持てるため、その多重度が要件に合う。

エ：所属は独立した業務関係であり名称への埋込みでは表現できない。

総合解説：各社員はちょうど1部門に所属し、部門側は0人以上の社員を持てるため、その多重度が要件に合う。

対象：2026年度 / 基準日 2026-09-02

0004 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：標準

請求システムで、支払期日は「請求日 + 取引先ごとの支払サイト」で決まる。支払サイト変更後も過去請求の当時の支払期日を再現する必要がある。設計として最も適切なものはどれか。

ア：支払期日は常に現在の取引先マスタから再計算する。

イ：支払期日を保持せず請求日だけ保存する。

ウ：支払サイトを全取引先で固定値にする。

エ：請求時点で決定した支払期日を請求データに保持し、算定に用いた条件の履歴も追跡できるようにする。

答え・解説 正答：エ

ア：現在値で再計算すると、マスタ変更後に過去の支払期日が変わってしまう。

イ：請求日だけでは取引先別条件を再現できない。

ウ：業務要件にない固定化は不適切である。

エ：過去時点の確定結果を再現するには、当時の支払期日またはその算定条件を履歴として保持する必要がある。

総合解説：過去時点の確定結果を再現するには、当時の支払期日またはその算定条件を履歴として保持する必要がある。

対象：2026年度 / 基準日 2026-09-02

0005 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：標準

会員資格テーブルに会員ID・有効開始日・有効終了日を保持する。同一会員について有効期間が重複する資格レコードを登録してはならない。最も適切なルールはどれか。

ア：同一会員の既存期間と新規期間が交差しないことを検証する。

イ：会員IDだけを一意制約にする。

ウ：有効開始日だけを全会員で一意にする。

エ：有効終了日をNULL禁止にするだけでよい。

答え・解説 正答：ア

ア：要求は同一会員内の期間重複禁止なので、期間交差の有無を検証する必要がある。

イ：会員ごとに複数期間を履歴管理できなくなる。

ウ：異なる会員が同日に開始できなくなるため過剰制約である。

エ：NULL可否だけでは期間重複を防げない。

総合解説：要求は同一会員内の期間重複禁止なので、期間交差の有無を検証する必要がある。

対象：2026年度 / 基準日 2026-09-02

0006 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：標準

倉庫業務で「出荷引当数量は、現在庫数量から既引当数量を差し引いた利用可能数量を超えてはならない」とする。最も適切なモデル上の扱いはどれか。

ア：商品コードの桁数制約だけで実現する。

イ：引当登録時に利用可能数量との整合を確認する業務制約として扱う。

ウ：出荷先住所を必須にすれば実現できる。

エ：現在庫数量を常に1に固定する。

答え・解説 正答：イ

ア：コード桁数は在庫数量の整合と無関係である。

イ：複数の数量から成立する整合条件であり、引当処理時に業務制約として検証するのが適切である。

ウ：住所必須性は数量制約を保証しない。

エ：在庫を固定するのは業務要件を損なう。

総合解説：複数の数量から成立する整合条件であり、引当処理時に業務制約として検証するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0007 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：標準

顧客は複数の配送先住所を登録できるが、配送先が1件以上ある場合は「既定配送先」を高々1件だけ指定できる。最も適切なルールはどれか。

ア：配送先住所を顧客テーブルの固定3列に持つ。

イ：全顧客で既定配送先は1件だけにする。

ウ：顧客ごとに既定フラグが真の配送先を0件または1件に制限する。

エ：配送先IDを全て同じ値にする。

答え・解説 正答：ウ

ア：固定列では件数可変の配送先を適切に扱えない。

イ：異なる顧客はそれぞれ既定配送先を持てる。

ウ：制約の単位は顧客ごとであり、既定指定を高々1件に制限する必要がある。

エ：配送先を識別できなくなり要件を満たさない。

総合解説：制約の単位は顧客ごとであり、既定指定を高々1件に制限する必要がある。

対象：2026年度 / 基準日 2026-09-02

0008 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：標準

商品価格は将来日付で改定予約でき、受注時には受注日時点で有効な価格を適用する。過去受注の価格根拠も追跡したい。最も適切なデータ要件はどれか。

ア：商品マスタの価格を上書きし、過去価格は保持しない。

イ：受注時の価格は毎回現在価格で再計算する。

ウ：価格改定ごとに商品コード自体を変更する。

エ：商品価格を有効期間付き履歴として管理し、受注には適用価格または参照した価格履歴を保持する。

答え・解説 正答：エ

ア：上書きだけでは過去の価格根拠を失う。

イ：再計算では現在価格変更後に過去受注金額が変わり得る。

ウ：商品識別子と価格版を混同すると商品同一性の管理が不自然になる。

エ：時点に応じた価格選択と過去根拠の再現には、有効期間付き価格履歴と受注側の確定情報が必要である。

総合解説：時点に応じた価格選択と過去根拠の再現には、有効期間付き価格履歴と受注側の確定情報が必要である。

対象：2026年度 / 基準日 2026-09-02

0009 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：応用

契約書は改訂版を複数保存するが、「一つの契約について現在有効な版は一つだけ」とする。将来版は事前登録できる。設計上のルールとして最も適切なものはどれか。

ア：契約版に有効期間を持たせ、同一契約で同一時点に複数版が有効にならないよう制約する。

イ：契約版番号を全契約共通で一意にするだけでよい。

ウ：最新版以外の版は削除する。

エ：契約IDを改訂ごとに新しい顧客IDへ置き換える。

答え・解説 正答：ア

ア：将来版・過去版を保持しつつ現在有効版を一つにするには、同一契約内で有効期間が重ならないことが重要である。

イ：版番号の一意性だけでは有効期間の重複を防げない。

ウ：履歴要件に反する。

エ：契約と顧客の識別を混同している。

総合解説：将来版・過去版を保持しつつ現在有効版を一つにするには、同一契約内で有効期間が重ならないことが重要である。

対象：2026年度 / 基準日 2026-09-02

0010 6-1 データモデリング事例 > 業務要件・ビジネスルール抽出 | 難易度：応用

退会者の個人データは「法令・業務上必要な保持期間を過ぎたら匿名化する。ただし取引履歴の集計値は残す」とする。データ要件として最も適切なものはどれか。

ア：退会時に全取引履歴を即時削除する。

イ：個人識別情報と取引事実を分離し、保持期限到来時に識別情報を匿名化できる構造と処理ルールを定義する。

ウ：個人識別情報を全テーブルへ重複コピーする。

エ：保持期限を考慮せず永久保存する。

答え・解説 正答：イ

ア：保持が必要な取引履歴まで失う可能性がある。

イ：識別情報と取引事実を分けることで、必要な集計・履歴を残しつつ個人識別性を下げる処理が可能になる。

ウ：匿名化対象が広がり整合管理も難しくなる。

エ：保持期限の業務要件に反する。

総合解説：識別情報と取引事実を分けることで、必要な集計・履歴を残しつつ個人識別性を下げる処理が可能になる。

対象：2026年度 / 基準日 2026-09-02

0011 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：基礎

「一人の顧客は0件以上の注文を行い、一つの注文は必ず一人の顧客に属する」。最も適切なER関係はどれか。

ア：顧客1：注文1の一对一関係。

イ：顧客0..N：注文0..Nの多対多関係。

ウ：顧客1：注文0..Nの一对多関係。

エ：顧客0..1：注文1の関係。

答え・解説 正答：ウ

ア：顧客は複数注文できる。

イ：一つの注文が複数顧客に属する要件ではない。

ウ：顧客側1件に対して注文が複数件対応し、各注文の顧客は一意なので一对多となる。

エ：顧客は注文がなくても存在できるが、注文側から顧客は必須である。

総合解説：顧客側1件に対して注文が複数件対応し、各注文の顧客は一意なので一对多となる。

対象：2026年度 / 基準日 2026-09-02

0012 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：基礎

一つの注文には複数商品が含まれ、一つの商品は複数注文に現れる。さらに注文ごと・商品ごとに数量と販売単価を保持する。最も適切なモデルはどれか。

ア：数量と販売単価を商品マスタだけに持たせる。

イ：注文と商品を一对一にする。

ウ：注文テーブルに商品1～商品20の列を固定で用意する。

エ：注文と商品の間に受注明細エンティティを置き、数量と販売単価を受注明細へ持たせる。

答え・解説 正答：エ

ア：販売単価は注文時点や条件で変わり得るため商品固有属性とは限らない。

イ：実際の多対多関係を表現できない。

ウ：商品数の上限を固定し、繰返し項目を生む。

エ：多対多関係の属性である数量・販売単価は、関連エンティティである受注明細に置くのが適切である。

総合解説：多対多関係の属性である数量・販売単価は、関連エンティティである受注明細に置くのが適切である。

対象：2026年度 / 基準日 2026-09-02

0013 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：基礎

部門は必要に応じて上位部門を一つ持ち、上位部門は複数の下位部門を持てる。最も適切なER表現はどれか。

- ア：部門エンティティ自身への再帰的な一対多関係を設ける。
- イ：上位部門と下位部門を別種類の社員エンティティで表す。
- ウ：部門階層を自由記述の文字列だけで持つ。
- エ：全ての部門を一つの行に列として並べる。

答え・解説 正答：ア

- ア：同一種類の部門間に親子関係があるため、部門から部門への再帰関係が自然である。
 - イ：部門と社員は異なる概念である。
 - ウ：構造的な親子関係をDB制約や参照で扱えない。
 - エ：階層の深さ変化に弱く正規化上も不自然である。
- 総合解説：同一種類の部門間に親子関係があるため、部門から部門への再帰関係が自然である。

対象：2026年度 / 基準日 2026-09-02

0014 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：基礎

請求書番号は全体で一意であり、請求明細番号は同一請求書内で1から採番する。請求明細の識別として最も適切なものはどれか。

- ア：明細番号だけで全請求明細を識別する。
- イ：請求書番号と明細番号の組合せで請求明細を識別する。
- ウ：請求日だけで識別する。
- エ：商品名だけで識別する。

答え・解説 正答：イ

- ア：異なる請求書で同じ明細番号が使われる。
 - イ：明細番号は親請求書の範囲内でのみ一意なので、親キーと部分キーの組合せが必要である。
 - ウ：同一日に複数明細が存在する。
 - エ：同一商品が複数請求に現れ得る。
- 総合解説：明細番号は親請求書の範囲内でのみ一意なので、親キーと部分キーの組合せが必要である。

対象：2026年度 / 基準日 2026-09-02

0015 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

社員は社用車を割り当てられない場合があり、社用車は運用中なら高々一人の社員に割り当てて。未割当車も存在する。最も適切な多重度はどれか。

ア：社員側1、社用車側1の必須一対一。

イ：社員側0..N、社用車側0..Nの多対多。

ウ：社員側0..1、社用車側0..1の任意な一対一関係。

エ：社員側1..N、社用車側1の一対多。

答え・解説 正答：ウ

ア：未割当を表現できない。

イ：一台を複数社員へ同時割当する要件ではない。

ウ：双方とも未割当があり得て、割当時も一対一なので0..1対0..1が適切である。

エ：社員が複数台必須という要件ではない。

総合解説：双方とも未割当があり得て、割当時も一対一なので0..1対0..1が適切である。

対象：2026年度 / 基準日 2026-09-02

0016 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

顧客には法人と個人があり、共通して顧客番号・名称・住所を持つ。法人だけ法人番号、個人だけ生年月日を持つ。最も適切な概念モデルはどれか。

ア：法人番号と生年月日を全顧客で必須にする。

イ：法人と個人を全く無関係なエンティティにし共通属性を重複定義するだけにする。

ウ：顧客種別を廃止し全属性を自由記述にする。

エ：顧客をスーパータイプとし、法人顧客・個人顧客をサブタイプとして固有属性を持たせる。

答え・解説 正答：エ

ア：法人には生年月日がなく個人には法人番号がない。

イ：重複定義が増え、同一概念としての顧客管理が難しくなる。

ウ：型・制約を失いデータ品質が下がる。

エ：共通属性と区分固有属性が明確なのでスーパータイプ/サブタイプが自然である。

総合解説：共通属性と区分固有属性が明確なのでスーパータイプ/サブタイプが自然である。

対象：2026年度 / 基準日 2026-09-02

0017 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

学生と科目は多対多で、履修年度・成績は学生と科目の組合せごとに決まる。属性配置として最も適切なものはどれか。

ア：履修という関連エンティティを設け、履修年度・成績を持たせる。

イ：成績を学生だけに持たせる。

ウ：履修年度を科目だけに持たせる。

エ：学生と科目を一对一に変更する。

答え・解説 正答：ア

ア：履修年度と成績は学生単独・科目単独ではなく関係の組合せに依存する。

イ：一人の学生は複数科目で異なる成績を持つ。

ウ：同一科目でも学生ごと・年度ごとに異なる。

エ：実際の多対多要件を失う。

総合解説：履修年度と成績は学生単独・科目単独ではなく関係の組合せに依存する。

対象：2026年度 / 基準日 2026-09-02

0018 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

仕入先・商品・倉庫の組合せごとに納入単価が決まる。仕入先と商品の組合せだけでは単価が一意に決まらず、倉庫によっても異なる。最も適切な概念モデルはどれか。

ア：仕入先だけに納入単価を持たせる。

イ：仕入先・商品・倉庫の三者を関連付ける供給条件エンティティを設け、納入単価を持たせる。

ウ：商品だけに納入単価を持たせる。

エ：倉庫だけに納入単価を持たせる。

答え・解説 正答：イ

ア：商品や倉庫による差を表現できない。

イ：単価の決定要因が三者の組合せなので、その組合せを識別できる関係として表現する必要がある。

ウ：仕入先や倉庫による差を表現できない。

エ：仕入先や商品による差を表現できない。

総合解説：単価の決定要因が三者の組合せなので、その組合せを識別できる関係として表現する必要がある。

対象：2026年度 / 基準日 2026-09-02

0019 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

請求の支払者は「個人顧客」または「法人顧客」のどちらか一方であり、同時に両方を支払者にはできない。最も適切な表現はどれか。

ア：個人顧客IDと法人顧客IDを両方必須にする。

イ：支払者を文字列だけで自由入力する。

ウ：顧客スーパータイプとの関係に統一するか、排他的な関係制約を明示して一方だけを参照させる。

エ：請求ごとに新しい顧客エンティティ種別を作る。

答え・解説 正答：ウ

ア：両方必須は要件と逆である。

イ：参照整合性や支払者の同一性を保証しにくい。

ウ：支払者という共通概念へ統一するか、排他制約を持たせることで「どちらか一方」を構造的に保証できる。

エ：種別が増殖しモデルの意味が不明確になる。

総合解説：支払者という共通概念へ統一するか、排他制約を持たせることで「どちらか一方」を構造的に保証できる。

対象：2026年度 / 基準日 2026-09-02

0020 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：標準

顧客の担当営業は時期によって変わり、過去の各期間に誰が担当していたかを参照したい。最も適切なモデルはどれか。

ア：顧客テーブルの担当者IDを上書きするだけにする。

イ：営業担当者名を顧客名へ連結する。

ウ：担当変更のたび顧客IDを変える。

エ：顧客と営業担当者との関係を担当履歴として独立させ、有効開始日・有効終了日を持たせる。

答え・解説 正答：エ

ア：過去の担当者を失う。

イ：構造化された参照関係にならない。

ウ：顧客の同一性を不必要に変えてしまう。

エ：関係自体が時系列で変化するので、関係の履歴エンティティに期間を持たせるのが適切である。

総合解説：関係自体が時系列で変化するので、関係の履歴エンティティに期間を持たせるのが適切である。

対象：2026年度 / 基準日 2026-09-02

0021 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：応用

顧客は電話番号を0件以上登録でき、各電話番号には「自宅・携帯・勤務先」などの種別を付ける。最も適切なモデルはどれか。

ア：顧客電話エンティティを設け、顧客ID・電話番号・種別を管理する。

イ：顧客テーブルに電話1、電話2、電話3の固定列を置く。

ウ：電話番号をカンマ区切り一文字列で持つ。

エ：電話種別を顧客全体で一つだけ持つ。

答え・解説 正答：ア

ア：件数可変の多値属性に付随属性もあるため、独立エンティティとして管理するのが適切である。

イ：件数上限が固定され、空き列や拡張問題が生じる。

ウ：検索・制約・個別更新が難しくなる。

エ：電話番号ごとに種別が異なる要件を表現できない。

総合解説：件数可変の多値属性に付随属性もあるため、独立エンティティとして管理するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0022 6-1 データモデリング事例 > ER図・リレーションシップ完成 | 難易度：応用

在庫数量の変化理由を追跡するため、入庫・出庫・棚卸調整を全て記録し、現在庫はそれらから整合確認できるようにしたい。最も適切な概念はどれか。

ア：現在庫数量だけを上書きし変動履歴を残さない。

イ：在庫変動イベントを独立して記録し、商品・倉庫・数量・発生時刻・種別を関連付ける。

ウ：入庫だけ記録し出庫は無視する。

エ：在庫変動理由を商品名に埋め込む。

答え・解説 正答：イ

ア：変化理由を再現できない。

イ：変化の原因を追跡する要件には、状態値だけでなく変動イベントを記録するモデルが有効である。

ウ：現在庫と履歴が一致しない。

エ：商品識別とイベント理由を混同している。

総合解説：変化の原因を追跡する要件には、状態値だけでなく変動イベントを記録するモデルが有効である。

対象：2026年度 / 基準日 2026-09-02

0023 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：基礎

従来は商品ごとに在庫数量を一つだけ持っていたが、今後は複数倉庫別に在庫を管理する。最も適切なモデル変更はどれか。

ア：商品テーブルに倉庫A在庫、倉庫B在庫を固定列で追加し続ける。

イ：倉庫を商品名の一部として表す。

ウ：商品と倉庫の組合せごとに在庫を表すエンティティを設ける。

エ：在庫数量を倉庫テーブルだけに一つ持つ。

答え・解説 正答：ウ

ア：倉庫追加のたび列追加が必要になる。

イ：商品同一性と保管場所を混同する。

ウ：在庫は商品×倉庫の組合せに依存するため、その交差を独立して管理するのが拡張性に優れる。

エ：倉庫には複数商品があるため一つの在庫数量では表せない。

総合解説：在庫は商品×倉庫の組合せに依存するため、その交差を独立して管理するのが拡張性に優れる。

対象：2026年度 / 基準日 2026-09-02

0024 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：基礎

商品に色・サイズ別のSKUを導入し、在庫と価格はSKU単位で管理する一方、商品説明は商品共通とする。最も適切な変更はどれか。

ア：商品説明をSKUごとに完全重複保存するだけにする。

イ：色とサイズを在庫テーブル名として埋め込む。

ウ：SKUを導入せず商品コードだけで全ての色・サイズを同一在庫にする。

エ：商品とSKUを分離し、共通情報を商品、色・サイズ・在庫管理単位をSKUへ置く。

答え・解説 正答：エ

ア：共通情報の更新時に複数行へ反映が必要になる。

イ：データ値をスキーマ名へ埋め込むのは拡張性が低い。

ウ：SKU単位管理という要件を満たさない。

エ：共通属性とバリエーション固有属性の依存先を分けることで、重複と更新不整合を抑えられる。

総合解説：共通属性とバリエーション固有属性の依存先を分けることで、重複と更新不整合を抑えられる。

対象：2026年度 / 基準日 2026-09-02

0025 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：標準

従来は一契約につき一利用者だったが、法人契約では複数利用者が一つの契約を共有できるようになった。最も適切なモデル変更はどれか。

ア：契約と利用者を分離し、契約会員などの関連エンティティで多対多または一対多の所属関係を表す。

イ：契約テーブルへ利用者1～利用者10列を追加する。

ウ：利用者IDを全員で同じ値にする。

エ：契約IDを利用者名へ変更する。

答え・解説 正答：ア

ア：契約と利用者の独立性を保ち、共有関係を別に表現するのが適切である。

イ：人数上限が固定され拡張性が低い。

ウ：個々の利用者を識別できない。

エ：識別子と表示名を混同している。

総合解説：契約と利用者の独立性を保ち、共有関係を別に表現するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0026 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：標準

一つの受注を複数回に分けて出荷できるようになった。各受注明細も複数出荷に分割され得る。最も適切な変更はどれか。

ア：受注ヘッダに出荷日をつつだけ追加する。

イ：出荷と出荷明細を追加し、受注明細と出荷明細の対応で出荷数量を管理する。

ウ：受注明細の数量を出荷のたび上書きする。

エ：受注ごとに出荷テーブルを個別作成する。

答え・解説 正答：イ

ア：複数出荷日を表現できない。

イ：分割出荷では受注と出荷が一対多となり、明細レベルで何個出荷したかを管理する必要がある。

ウ：受注数量の元情報を失う。

エ：スキーマがデータ件数に応じて増殖する。

総合解説：分割出荷では受注と出荷が一対多となり、明細レベルで何個出荷したかを管理する必要がある。

対象：2026年度 / 基準日 2026-09-02

0027 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：標準

単発売切り型の商品販売に加え、毎月自動更新されるサブスクリプションを扱う。契約期間・更新条件・請求周期を管理したい。最も適切な対応はどれか。

ア：毎月の契約情報を商品名へ追記する。

イ：契約期間を注文日の一列だけで表す。

ウ：単発注文とは別に契約エンティティを設け、期間・更新条件・請求周期を管理し、必要に応じ請求イベントへ関連付ける。

エ：更新条件を担当者のメモだけに保存する。

答え・解説 正答：ウ

ア：構造化されず検索・制約が困難になる。

イ：開始・終了・更新条件を表現できない。

ウ：継続契約は単発取引とは異なるライフサイクルと属性を持つため、独立した契約概念が必要である。

エ：業務ルールをDBモデルで管理できない。

総合解説：継続契約は単発取引とは異なるライフサイクルと属性を持つため、独立した契約概念が必要である。

対象：2026年度 / 基準日 2026-09-02

0028 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：標準

顧客を物理削除していたが、監査のため退会後も取引との関連を保持し、一定期間内の再入会時には同一人物として扱いたい。最も適切な変更はどれか。

ア：退会時に顧客と取引を全て削除する。

イ：再入会ごとに過去取引の顧客IDを書き換える。

ウ：顧客IDをメールアドレス文字列そのものに限定する。

エ：顧客の状態・退会日等を保持し、取引参照を維持したままライフサイクルを管理する。

答え・解説 正答：エ

ア：監査対象の取引関連を失う。

イ：過去事実の改変となる。

ウ：メールアドレス変更時の識別が不安定になる。

エ：監査・再入会要件には、顧客の同一性を保ちつつ状態遷移を履歴化する方法が適切である。

総合解説：監査・再入会要件には、顧客の同一性を保ちつつ状態遷移を履歴化する方法が適切である。

対象：2026年度 / 基準日 2026-09-02

0029 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：応用

従来は税率が全地域共通だったが、今後は地域と商品区分によって税率が異なり、施行日も異なる。過去請求の根拠も必要である。最も適切な変更はどれか。

ア：地域・商品区分・有効期間をキー要素とする税率履歴を設け、請求時に適用結果または参照版を保持する。

イ：税率を一つのシステム定数のままにする。

ウ：請求金額だけ保存し税率根拠は保存しない。

エ：地域ごとに別DB製品へ分けることだけで対応する。

答え・解説 正答：ア

ア：税率の決定要因と時点をデータとして管理し、過去根拠を再現できる設計が必要である。

イ：複数税率を表現できない。

ウ：監査や再計算で根拠を確認できない。

エ：DB分割は税率ロジックのモデル化そのものではない。

総合解説：税率の決定要因と時点をデータとして管理し、過去根拠を再現できる設計が必要である。

対象：2026年度 / 基準日 2026-09-02

0030 6-1 データモデリング事例 > 要件変更・モデル修正 | 難易度：応用

売上金額を円だけで保持していたが、取引通貨と決済時の換算レートを記録し、後日同じ円換算額を再現したい。最も適切な変更はどれか。

ア：現在の為替レートだけを参照し過去取引も毎回再計算する。

イ：取引金額・通貨コード・適用換算レート（またはレート版）・換算後金額を取引事実として保持する。

ウ：通貨コードを商品コードに埋め込む。

エ：円換算額だけを残し元通貨を捨てる。

答え・解説 正答：イ

ア：レート変動後に過去金額が変わる。

イ：過去の換算結果を再現するには、取引時に用いた通貨とレートの根拠を保持する必要がある。

ウ：商品識別と通貨を混同する。

エ：元取引の通貨情報を失う。

総合解説：過去の換算結果を再現するには、取引時に用いた通貨とレートの根拠を保持する必要がある。

対象：2026年度 / 基準日 2026-09-02

0031 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：基礎

関係 $R(A,B,C)$ で $A \rightarrow B$ 、 $B \rightarrow C$ が成り立ち、 A の値は行を一意に識別できる。候補キーとして最も適切なものはどれか。

- ア：{A,B}
- イ：{A,B,C}
- ウ：{A}
- エ：{B}

答え・解説 正答：ウ

ア：スーパーキーではあるがAだけで十分なので最小ではない。
イ：全属性を含むため一意にはできても最小ではない。
ウ：AからB、さらにCを導出でき、A単独で全属性を決定できるうえ最小なので候補キーである。
エ：BからCは決まるがAを決定できるとは限らない。
総合解説：AからB、さらにCを導出でき、A単独で全属性を決定できるうえ最小なので候補キーである。

対象：2026年度 / 基準日 2026-09-02

0032 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：基礎

受講(学生ID, 科目ID, 年度, 成績)で、一人の学生は同じ科目を年度が違えば再受講でき、各年度では一度だけ受講できる。候補キーとして最も適切なものはどれか。

- ア：{学生ID, 科目ID}
- イ：{科目ID, 年度}
- ウ：{学生ID, 成績}
- エ：{学生ID, 科目ID, 年度}

答え・解説 正答：エ

ア：再受講があるため年度なしでは一意にならない。
イ：同じ科目・年度に複数学生がいる。
ウ：成績は識別子ではなく重複し得る。
エ：受講事実を一意にするには学生・科目・年度の組合せが必要である。
総合解説：受講事実を一意にするには学生・科目・年度の組合せが必要である。

対象：2026年度 / 基準日 2026-09-02

0033 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：基礎

注文明細(注文番号, 商品番号, 商品名, 数量)の候補キーが(注文番号, 商品番号)で、商品番号→商品名が成り立つ。商品名の従属性はどれか。

ア：候補キーの一部である商品番号への部分関数従属。

イ：候補キー全体への完全関数従属だけが成り立つ。

ウ：商品名→商品番号の多値従属だけが成り立つ。

エ：注文番号への推移的従属である。

答え・解説 正答：ア

ア：複合キーの一部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。

イ：商品名はキー全体を必要としない。

ウ：問題文は商品番号→商品名という関数従属を示している。

エ：注文番号から商品名が決まるとは示されていない。

総合解説：複合キーの一部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。

対象：2026年度 / 基準日 2026-09-02

0034 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 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 が成り立つ。社員番号から部門名への関係として最も適切なものはどれか。

ア：部門名が社員番号の候補キーである。

イ：部門番号を介した推移的関数従属。

ウ：部門番号が社員番号へ必ず従属するため対称である。

エ：多値従属であり関数従属ではない。

答え・解説 正答：イ

ア：部門名は複数社員で同じになり得る。

イ：社員番号→部門番号→部門名の連鎖により、部門名は社員番号へ推移的に従属する。

ウ：関数従属は一般に対称ではない。

エ：各社員に部門名が一つ決まるので関数従属として扱える。

総合解説：社員番号→部門番号→部門名の連鎖により、部門名は社員番号へ推移的に従属する。

対象：2026年度 / 基準日 2026-09-02

0035 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：標準

$R(A,B,C,D)$ で $A \rightarrow B$ 、 $B \rightarrow C$ 、 $AC \rightarrow D$ が成り立つ。 A の属性閉包 A^+ として最も適切なものはどれか。

ア：{A,B}

イ：{A,B,C}

ウ：{A,B,C,D}

エ：{A,D}

答え・解説 正答：ウ

ア：BからCをさらに導出できる。

イ： $AC \rightarrow D$ を適用できるためDも含まれる。ウ： $A \rightarrow B$ 、 $B \rightarrow C$ からAでCまで得られ、AとCが得られたので $AC \rightarrow D$ も使え、全属性を導出できる。エ： $A \rightarrow B$ によりBが必要であり、その後Cも導出される。総合解説： $A \rightarrow B$ 、 $B \rightarrow C$ からAでCまで得られ、AとCが得られたので $AC \rightarrow D$ も使え、全属性を導出できる。

対象：2026年度 / 基準日 2026-09-02

0036 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：標準

関数従属性集合 $\{AB \rightarrow C, A \rightarrow C, C \rightarrow D\}$ がある。 $AB \rightarrow C$ について最も適切な説明はどれか。

ア：Aは必ず余分属性である。

イ： $AB \rightarrow C$ は他の従属性から決して導出できない。ウ： $C \rightarrow D$ があるためCは $AB \rightarrow C$ の左辺から削除できる。エ： $A \rightarrow C$ が既に成り立つため、 $AB \rightarrow C$ のBは左辺の余分属性である。

答え・解説 正答：エ

ア：B単独でCを決定できる情報はない。

イ： $A \rightarrow C$ から $AB \rightarrow C$ は増加律で導出できる。ウ：Cは $AB \rightarrow C$ の右辺であり左辺属性ではない。エ：A単独でCを決定できるため、 $AB \rightarrow C$ のBを除いても同じ結論が得られる。総合解説：A単独でCを決定できるため、 $AB \rightarrow C$ のBを除いても同じ結論が得られる。

対象：2026年度 / 基準日 2026-09-02

0037 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：標準

候補キーが{社員番号}である社員関係について、{社員番号, 氏名}の位置付けとして最も適切なものはどれか。

ア：行を一意に識別できるスーパーキーだが、最小ではないので候補キーではない。

イ：候補キーでもスーパーキーでもない。

ウ：主キーに選ばれていなければ一意性はない。

エ：氏名が含まれるので必ず候補キーである。

答え・解説 正答：ア

ア：候補キーを含む属性集合はスーパーキーになるが、余分な氏名を含むため最小性を満たさない。

イ：社員番号を含むので一意識別は可能である。

ウ：主キー選択と候補キー・スーパーキーの論理的性質は別である。

エ：属性数が多いことは候補キーの条件ではなく、最小性が必要である。

総合解説：候補キーを含む属性集合はスーパーキーになるが、余分な氏名を含むため最小性を満たさない。

対象：2026年度 / 基準日 2026-09-02

0038 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：標準

顧客テーブルに代理キーcustomer_idを追加した。業務上、会員番号も重複を許さない。最も適切な設計はどれか。

ア：代理キーがあるので会員番号の重複を無条件に許す。

イ：customer_idを主キーにしても、会員番号には一意制約を設け業務上の候補キー性を保つ。

ウ：会員番号を全てNULLにする。

エ：主キーを二つ定義する。

答え・解説 正答：イ

ア：同一会員番号が複数顧客へ割り当たると業務ルールに反する。

イ：代理キー導入は業務上の一意性要件を消さないため、自然キー側にも一意制約が必要である。

ウ：必要な業務識別子を失う。

エ：関係には主キーは一つだが、候補キーは複数存在し得る。

総合解説：代理キー導入は業務上の一意性要件を消さないため、自然キー側にも一意制約が必要である。

対象：2026年度 / 基準日 2026-09-02

0039 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：応用

$R(A,B,C)$ で $A \rightarrow B$ 、 $B \rightarrow A$ 、 $A \rightarrow C$ が成り立つ。候補キーの組として最も適切なものはどれか。

ア：Aだけが候補キーでBはなり得ない。

イ：Cだけが候補キーである。

ウ：AとBの両方がそれぞれ候補キーになり得る。

エ：ABだけが候補キーである。

答え・解説 正答：ウ

ア： $B \rightarrow A \rightarrow C$ があるのでBも全属性を決定できる。

イ：CからAやBを決める従属性はない。

ウ：AはBとCを決定でき、BはAを決定してから $A \rightarrow C$ でCも決定できるため、 $A \cdot B$ はいずれも最小のキーとなる。

エ：AまたはB単独で十分なのでABは最小ではない。

総合解説：AはBとCを決定でき、BはAを決定してから $A \rightarrow C$ でCも決定できるため、 $A \cdot B$ はいずれも最小のキーとなる。

対象：2026年度 / 基準日 2026-09-02

0040 6-2 関係スキーマ事例 > キー・関数従属性の特定 | 難易度：応用

「郵便番号が決まれば都道府県・市区町村までの住所情報が一意に決まる」という前提で、郵便番号→住所が成り立つ。このとき郵便番号の呼び方として最も適切なものはどれか。

ア：住所から必ず決まる被決定項。

イ：外部キーでなければ関数従属性とは呼べない。

ウ：候補キーでなければ決定項にはなれない。

エ：住所を決定する決定項。

答え・解説 正答：エ

ア：被決定項は右辺側の住所である。

イ：関数従属性は外部キーの有無とは独立した概念である。

ウ：決定項は候補キーに限られない。

エ： $X \rightarrow Y$ でXはYを決定する側、すなわち決定項である。

総合解説： $X \rightarrow Y$ でXはYを決定する側、すなわち決定項である。

対象：2026年度 / 基準日 2026-09-02

0041 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：基礎

顧客(顧客ID, 氏名, 電話番号一覧)で電話番号一覧を「090...,03...」のような複数値を一つの属性に格納している。第一正規形の観点から最も適切な改善はどれか。

ア：電話番号を顧客電話などの別関係へ分離し、1セルに一つの値を持たせる。

イ：電話番号をさらに長い文字列型へ変更する。

ウ：区切り文字をカンマからスラッシュへ変える。

エ：顧客IDを削除する。

答え・解説 正答：ア

ア：第一正規形では属性値を原子的に扱うため、件数可変の電話番号は別関係にするのが適切である。

イ：型の長さを変えても複数値格納は解消しない。

ウ：区切り文字変更では構造は変わらない。

エ：識別子削除は正規化の解決にならない。

総合解説：第一正規形では属性値を原子的に扱うため、件数可変の電話番号は別関係にするのが適切である。

対象：2026年度 / 基準日 2026-09-02

0042 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：基礎

注文明細(注文番号, 商品番号, 商品名, 数量)でキーは(注文番号,商品番号)、商品番号→商品名が成り立つ。第二正規形にする分解として最も適切なものはどれか。

ア：注文明細から商品番号を削除する。

イ：注文明細(注文番号,商品番号,数量)と商品(商品番号,商品名)に分ける。

ウ：注文番号と商品名だけを残す。

エ：商品名を注文ヘッダへ移す。

答え・解説 正答：イ

ア：明細がどの商品か分からなくなる。

イ：商品名の部分関数従属を商品関係へ分離し、明細にはキー全体に依存する数量を残す。

ウ：数量や商品識別が欠落する。

エ：商品名は注文単位ではなく商品番号に依存する。

総合解説：商品名の部分関数従属を商品関係へ分離し、明細にはキー全体に依存する数量を残す。

対象：2026年度 / 基準日 2026-09-02

0043 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：基礎

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→部門名が成り立つ。第三正規形に近づける分解として最も適切なものはどれか。

ア：社員番号と部門名だけを社員に残す。

イ：部門番号を削除し部門名を主キーにする。

ウ：社員(社員番号,部門番号)と部門(部門番号,部門名)に分ける。

エ：社員ごとに部門名を複製し続ける。

答え・解説 正答：ウ

ア：部門の識別・参照が不安定になる。

イ：部門名変更など自然キー変化の影響が大きい。

ウ：非キー属性部門名の推移的従属を部門関係へ分離することで更新不整合を抑えられる。

エ：同一部門名の重複と更新異常が残る。

総合解説：非キー属性部門名の推移的従属を部門関係へ分離することで更新不整合を抑えられる。

対象：2026年度 / 基準日 2026-09-02

0044 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：標準

関係R(学生,科目,教員)で、(学生,科目)→教員、教員→科目が成り立つ。教員は一つの科目だけを担当するが、同じ科目を複数教員が担当できる。この関係が3NFを満たしていてもBCNFに違反し得る主な理由はどれか。

ア：(学生,科目)が候補キーだから。

イ：科目が文字列型だから。

ウ：学生が外部キーだから。

エ：教員→科目の決定項「教員」がスーパーキーではないから。

答え・解説 正答：エ

ア：候補キーであること自体は違反理由ではない。

イ：データ型はBCNF条件ではない。

ウ：外部キーかどうかはBCNF条件と直接関係しない。

エ：BCNFでは全ての非自明な関数従属性の決定項がスーパーキーである必要がある。

総合解説：BCNFでは全ての非自明な関数従属性の決定項がスーパーキーである必要がある。

対象：2026年度 / 基準日 2026-09-02

0045 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：標準

R(A,B,C)をR1(A,B)とR2(B,C)に分解する。B→Aが成り立つとき、無損失分解を判断する上で最も重要な点はどれか。

ア：共通属性Bが少なくとも一方の分解関係の全属性を関数的に決定するので、自然結合で余分な組合せが生じにくい。

イ：分解後の表名が短いこと。

ウ：各関係の列数が同じこと。

エ：Bのデータ型が文字列であること。

答え・解説 正答：ア

ア：二分解の無損失性は共通属性が一方を関数的に決定する条件で判定できる。

イ：名称は論理的な無損失性に関係しない。

ウ：列数の一致は条件ではない。

エ：データ型の種類だけでは無損失性は決まらない。

総合解説：二分解の無損失性は共通属性が一方を関数的に決定する条件で判定できる。

対象：2026年度 / 基準日 2026-09-02

0046 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：標準

正規化で関係を分解するとき、従属性保存を重視する主な理由として最も適切なものはどれか。

ア：必ず全ての問合せを単一表で処理するため。

イ：元の関数従属性を、分解後の各関係上の制約を用いて結合なしに検査しやすくするため。

ウ：主キーを不要にするため。

エ：外部キーを全て削除するため。

答え・解説 正答：イ

ア：正規化後は複数関係を結合する問合せも存在する。

イ：従属性保存されていれば、元の制約を分解後の局所的な制約として管理しやすい。

ウ：主キーは依然必要になり得る。

エ：参照関係を表す外部キーはむしろ必要になることが多い。

総合解説：従属性保存されていれば、元の制約を分解後の局所的な制約として管理しやすい。

対象：2026年度 / 基準日 2026-09-02

0047 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：標準

商品分類名を各商品行に繰返し保存しているため、分類名変更時に一部の商品行だけ旧名称が残った。この問題に最も対応する設計はどれか。

ア：分類名列をさらに二重に持つ。

イ：分類名を毎回手作業で同時更新する運用だけに頼る。

ウ：分類を独立関係にし、商品から分類IDを参照する。

エ：商品IDを削除する。

答え・解説 正答：ウ

ア：重複が増え問題が悪化する。

イ：人手運用だけでは整合性保証が弱い。

ウ：同一事実の重複保存を減らし、分類名を一箇所で管理すれば更新異常を防ぎやすい。

エ：商品識別を失っても分類名重複は解決しない。

総合解説：同一事実の重複保存を減らし、分類名を一箇所で管理すれば更新異常を防ぎやすい。

対象：2026年度 / 基準日 2026-09-02

0048 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：標準

科目と受講者を一つの表だけで管理しており、受講者がまだいない新設科目を登録できない。これは主にどの異常か。

ア：削除異常。

イ：デッドロック。

ウ：ファントムリード。

エ：挿入異常。

答え・解説 正答：エ

ア：削除に伴って別の事実まで失う問題ではない。

イ：並行制御上の待合いではない。

ウ：トランザクション分離レベルの読取り現象ではない。

エ：他の事実が存在しないと新しい独立事実を登録できない問題は挿入異常である。

総合解説：他の事実が存在しないと新しい独立事実を登録できない問題は挿入異常である。

対象：2026年度 / 基準日 2026-09-02

0049 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：応用

支店と担当者を同一表で管理しており、ある支店の最後の担当者行を削除すると支店住所まで失われる。この問題として最も適切なものはどれか。

- ア：削除異常。
- イ：挿入異常。
- ウ：ロストアップデート。
- エ：参照循環。

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は削除異常である。

イ：新規登録ができない問題ではない。

ウ：並行更新による上書き問題ではない。

エ：外部キー循環の問題ではない。

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は削除異常である。

対象：2026年度 / 基準日 2026-09-02

0050 6-2 関係スキーマ事例 > 正規化・関係スキーマ完成 | 難易度：応用

十分に正規化されたOLTPモデルで、実測の結果、特定集計の結合コストがSLAを満たさない。非正規化を検討するとき最も適切な方針はどれか。

- ア：全テーブルを一つに統合する。
- イ：性能効果を測定し、冗長データの更新整合性を維持する仕組みを設計した上で限定的に適用する。
- ウ：正規化の理由を確認せず重複列を無制限に追加する。
- エ：整合性を犠牲にすれば必ず性能が上がるとみなす。

答え・解説 正答：イ

ア：不要な冗長性と更新異常を大幅に増やす。

イ：非正規化は性能改善の手段になり得るが、整合性コストとのトレードオフを測定・管理する必要がある。

ウ：効果と副作用を評価しない設計は危険である。

エ：性能向上はアクセスパターン次第で保証されない。

総合解説：非正規化は性能改善の手段になり得るが、整合性コストとのトレードオフを測定・管理する必要がある。

対象：2026年度 / 基準日 2026-09-02

0051 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：基礎

注文には必ず既存顧客を対応付ける。DBでこの整合性を最も直接的に保証する仕組みはどれか。

ア：注文日をUNIQUEにする。

イ：注文番号を文字列型にする。

ウ：注文の顧客IDを顧客主キーへ参照する外部キーにする。

エ：顧客名を注文へ自由入力する。

答え・解説 正答：ウ

ア：注文日は複数注文で同じになり得る。

イ：データ型だけでは参照先存在を保証しない。

ウ：外部キーは参照先行の存在を保証し、孤立した顧客参照を防ぐ。

エ：表記揺れや存在しない顧客を登録できる。

総合解説：外部キーは参照先行の存在を保証し、孤立した顧客参照を防ぐ。

対象：2026年度 / 基準日 2026-09-02

0052 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：基礎

売上明細は法的保存期間中に削除してはならず、参照する商品マスタも過去売上から識別できる必要がある。商品削除時の方針として最も適切なものはどれか。

ア：商品削除時に過去売上明細も連鎖削除する。

イ：売上明細の商品IDをNULLへ自動変更する。

ウ：商品IDをランダム値へ書き換える。

エ：売上で参照中の商品は物理削除を制限し、廃止状態などで運用する。

答え・解説 正答：エ

ア：保存要件に反する。

イ：過去に何を売ったか識別できなくなる。

ウ：参照整合性と監査証跡を壊す。

エ：過去売上の参照可能性を保つには、参照中マスタの削除を防ぎ状態管理するのが適切である。

総合解説：過去売上の参照可能性を保つには、参照中マスタの削除を防ぎ状態管理するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0053 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：基礎

社員の所属履歴に社員ID・部門ID・開始日・終了日を持つ。同一社員が同じ時点に二つの部門へ所属しない前提である。最も重要な整合性はどれか。

- ア：同一社員の所属期間が重ならないこと。
- イ：全社員の開始日が一意であること。
- ウ：部門IDが全履歴で一意であること。
- エ：終了日は開始日より必ず前であること。

答え・解説 正答：ア

ア：要件は社員ごとの時点重複禁止なので、期間の非重複制約が中心になる。
イ：異なる社員は同日に異動できる。
ウ：同一部門へ複数社員が所属できる。
エ：通常は終了日は開始日以後である。
総合解説：要件は社員ごとの時点重複禁止なので、期間の非重複制約が中心になる。

対象：2026年度 / 基準日 2026-09-02

0054 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：標準

商品コードの先頭2桁にカテゴリを埋め込んでいたが、カテゴリ再編が頻繁になりコード変更の影響が大きい。改善策として最も適切なものはどれか。

- ア：カテゴリ変更のたび商品コードを全履歴で書き換える。
- イ：商品識別子とカテゴリコードを分離し、カテゴリは属性・参照関係として管理する。
- ウ：カテゴリ名をコード桁へさらに追加する。
- エ：商品コードを廃止して商品名だけをキーにする。

答え・解説 正答：イ

ア：過去参照や外部連携への影響が大きい。
イ：変化しやすい業務分類を識別子へ埋め込まないことで、再編の影響を局所化できる。
ウ：意味をさらに埋め込むほど変更耐性が下がる。
エ：商品名は変更・重複し得るため識別子として不安定である。
総合解説：変化しやすい業務分類を識別子へ埋め込まないことで、再編の影響を局所化できる。

対象：2026年度 / 基準日 2026-09-02

0055 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：標準

注文に代理キーorder_idを使う。業務上は「店舗番号 + 店舗内注文番号」も重複禁止である。最も適切な制約はどれか。

ア：order_idがあれば業務キー重複は許可する。

イ：店舗番号だけを一意にする。

ウ：order_idを主キーとし、店舗番号 + 店舗内注文番号には別途一意制約を設ける。

エ：店舗内注文番号だけを全店舗で一意にする。

答え・解説 正答：ウ

ア：同一店舗で注文番号重複が生じ要件違反になる。

イ：一店舗に複数注文を登録できなくなる。

ウ：物理識別と業務一意性は別に保証でき、代理キー導入後も業務キー制約は必要である。

エ：店舗ごと採番という要件より過剰な制約になる。

総合解説：物理識別と業務一意性は別に保証でき、代理キー導入後も業務キー制約は必要である。

対象：2026年度 / 基準日 2026-09-02

0056 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：標準

監査対象の契約条件に誤登録が見つかった。過去に何が登録され、いつ誰が訂正したかも残したい。最も適切な方式はどれか。

ア：誤登録行を物理削除し正しい値だけ残す。

イ：更新者情報を担当者の記憶に任せる。

ウ：契約番号自体を毎回別顧客番号へ変える。

エ：元レコードを証跡なく上書きせず、版・有効期間・訂正イベントなどで変更履歴を残す。

答え・解説 正答：エ

ア：変更前の事実を確認できない。

イ：システム上の証跡にならない。

ウ：契約と顧客の識別概念を混同する。

エ：監査では変更前後・時点・主体を追跡できることが重要で、履歴として変更を残す設計が適切である。

総合解説：監査では変更前後・時点・主体を追跡できることが重要で、履歴として変更を残す設計が適切である。

対象：2026年度 / 基準日 2026-09-02

0057 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：標準

住所区分コードの一部を廃止するが、過去取引では旧コードを表示できる必要がある。最も適切な管理はどれか。

- ア：コードマスタに有効期間・廃止状態を持たせ、過去参照を維持しつつ新規利用を制限する。
- イ：廃止日にコード行を物理削除する。
- ウ：旧コードを全過去取引から新コードへ一括置換する。
- エ：コード値をNULLにする。

答え・解説 正答：ア

- ア：過去の意味を保持しながら新規利用だけ止めるには、コードのライフサイクルを管理するのが適切である。
- イ：過去データの参照整合性を失う。
- ウ：過去時点の事実を改変する。
- エ：区分の意味を失う。
- 総合解説：過去の意味を保持しながら新規利用だけ止めるには、コードのライフサイクルを管理するのが適切である。

対象：2026年度 / 基準日 2026-09-02

0058 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：標準

予約テーブルで「終了日時は開始日時より後」とするルールをDB制約で保証したい。最も直接的な方法はどれか。

- ア：終了日時にUNIQUEだけを設定する。
- イ：同一行の開始日時と終了日時を比較するCHECK相当の制約を設ける。
- ウ：開始日時を外部キーにする。
- エ：予約者名をNOT NULLにするだけでよい。

答え・解説 正答：イ

- ア：終了日時の重複可否とは別のルールである。
- イ：同一行内の属性間条件なので、行レベルの検査制約が直接的である。
- ウ：参照整合性の問題ではない。
- エ：予約者必須性は日時順序を保証しない。
- 総合解説：同一行内の属性間条件なので、行レベルの検査制約が直接的である。

対象：2026年度 / 基準日 2026-09-02

0059 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：応用

商品マスタの価格履歴を、同一商品について複数版保存する。商品IDだけを主キーにすると登録できない。最も適切な識別方法はどれか。

ア：商品IDの一意制約をそのまま維持する。

イ：価格を主キーにする。

ウ：商品IDと版番号または有効開始日時の組合せなど、履歴版を一意に識別できるキーを用いる。

エ：商品名だけを主キーにする。

答え・解説 正答：ウ

ア：同一商品で複数履歴を持ってない。

イ：同価格が再登場し得る。

ウ：履歴は同一商品に複数行が存在するため、商品識別子に版を区別する要素を加える必要がある。

エ：商品名は変更・重複の可能性がある。

総合解説：履歴は同一商品に複数行が存在するため、商品識別子に版を区別する要素を加える必要がある。

対象：2026年度 / 基準日 2026-09-02

0060 6-2 関係スキーマ事例 > 整合性・履歴・コード設計 | 難易度：応用

受注には受注日時点で有効だった顧客ランクを適用し、後に顧客ランクが変更されても過去受注の割引根拠は変えない。最も適切な設計はどれか。

ア：顧客の現在ランクだけを受注表示時に参照する。

イ：過去受注を現在ランクに合わせて更新する。

ウ：ランク名を顧客氏名へ埋め込む。

エ：顧客ランクを履歴管理し、受注に適用ランクID/版または確定割引率を保持して時点根拠を固定する。

答え・解説 正答：エ

ア：現在値変更で過去の根拠が変化する。

イ：過去事実を改変する。

ウ：属性の意味が混在し参照整合性も失われる。

エ：時点依存のルールを再現するには、過去に適用した版や結果を固定する必要がある。

総合解説：時点依存のルールを再現するには、過去に適用した版や結果を固定する必要がある。

対象：2026年度 / 基準日 2026-09-02

0061 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：基礎

顧客customer(cust_id)と注文orders(order_id,cust_id)がある。「注文を一度もしていない顧客」を抽出する条件として最も適切なものはどれか。

ア：WHERE NOT EXISTS (SELECT 1 FROM orders o WHERE o.cust_id = c.cust_id)

イ：WHERE EXISTS (SELECT 1 FROM orders o WHERE o.cust_id = c.cust_id)

ウ：INNER JOIN orders o ON o.cust_id=c.cust_id

エ：WHERE c.cust_id IN (SELECT cust_id FROM orders)

答え・解説 正答：ア

ア：相関したNOT EXISTSは、その顧客に対応する注文行が存在しない場合だけ真になる。

イ：注文が存在する顧客を選ぶ。

ウ：注文が存在する顧客だけが結合結果に出る。

エ：注文済み顧客を選ぶ条件である。

総合解説：相関したNOT EXISTSは、その顧客に対応する注文行が存在しない場合だけ真になる。

対象：2026年度 / 基準日 2026-09-02

0062 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：基礎

顧客を全件残しつつ、注文がない顧客だけを抽出するSQLとして一般的に適切な形はどれか。

ア：customerとordersをINNER JOINする。

イ：customerを左側にLEFT JOIN ordersし、結合先の主キーがNULLの行を選ぶ。

ウ：ordersを全件削除してからSELECTする。

エ：customer.cust_idをNULLにする。

答え・解説 正答：イ

ア：内部結合では注文なし顧客が消える。

イ：左外部結合では注文がない顧客も残り、結合先列がNULLになるため未対応行を識別できる。

ウ：データ破壊は抽出方法ではない。

エ：顧客識別子を変更する必要はない。

総合解説：左外部結合では注文がない顧客も残り、結合先列がNULLになるため未対応行を識別できる。

対象：2026年度 / 基準日 2026-09-02

0063 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：基礎

注文を顧客別に集計し、「注文件数が5件以上の顧客だけ」を返したい。件数条件を置く位置として最も適切なものはどれか。

ア：WHERE COUNT(*) >= 5 とする。

イ：ORDER BY COUNT(*) >= 5 とする。

ウ：GROUP BY cust_id の後、HAVING COUNT(*) >= 5 とする。

エ：SELECT句からcust_idを削除する。

答え・解説 正答：ウ

ア：WHEREは集約前の行条件であり集約関数を直接条件にする用途ではない。

イ：ORDER BYは並べ替えであり絞込みではない。

ウ：集約結果に対する条件はHAVINGで指定するのが基本である。

エ：顧客別集計には顧客識別が必要である。

総合解説：集約結果に対する条件はHAVINGで指定するのが基本である。

対象：2026年度 / 基準日 2026-09-02

0064 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：標準

売上(salesperson_id, amount)を営業担当者別に合計し、合計額の高い順に順位も付けたい。集計行を失わず順位を付ける考え方として最も適切なものはどれか。

ア：全行にROW_NUMBER()を付けてから集計をやめる。

イ：WHERE句だけで順位を計算する。

ウ：主キー値の大小を売上順位とみなす。

エ：担当者別集計結果に対しRANK() OVER(ORDER BY 合計額 DESC)などのウィンドウ関数を適用する。

答え・解説 正答：エ

ア：元明細の番号では担当者別合計順位にならない。

イ：WHEREは順位算出機能ではない。

ウ：主キー順と売上額順に意味的な関係はない。

エ：ウィンドウ関数は結果行を保持したまま順序に基づく順位を付与できる。

総合解説：ウィンドウ関数は結果行を保持したまま順序に基づく順位を付与できる。

対象：2026年度 / 基準日 2026-09-02

0065 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：標準

employees(emp_id, dept_id, salary)について、各社員の給与が「同じ部門の平均給与より高い」社員を求めたい。最も適切な条件はどれか。

ア：WHERE e.salary > (SELECT AVG(x.salary) FROM employees x WHERE x.dept_id = e.dept_id)

イ：WHERE e.salary > (SELECT AVG(x.salary) FROM employees x)

ウ：WHERE e.salary = e.dept_id

エ：WHERE e.dept_id > (SELECT MAX(salary) FROM employees)

答え・解説 正答：ア

ア：外側社員のdept_idで関連させることで、社員ごとに同一部門の平均と比較できる。

イ：全社平均との比較になり要件と異なる。

ウ：給与と部門IDを比較するのは意味が異なる。

エ：部門IDと給与最大値を比較しており要件に合わない。

総合解説：外側社員のdept_idで関連させることで、社員ごとに同一部門の平均と比較できる。

対象：2026年度 / 基準日 2026-09-02

0066 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：標準

A集合に存在しB集合には存在しないキーを求めたい。集合演算の考え方として最も適切なものはどれか。

ア：AとBをUNIONして重複を除くだけにする。

イ：Aの結果からBの結果を差し引く集合差（EXCEPT相当）を使う。

ウ：AとBをCROSS JOINする。

エ：AとBの全件を同じ値に更新する。

答え・解説 正答：イ

ア：和集合になりBだけの要素も含み得る。

イ：要求はA - Bなので集合差が直接対応する。

ウ：直積は組合せを増やす。

エ：検索ではなく更新であり要件と無関係である。

総合解説：要求はA - Bなので集合差が直接対応する。

対象：2026年度 / 基準日 2026-09-02

0067 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：標準

副問合せ結果にNULLが含まれる可能性があるとき、「存在しないこと」を判定するSQLでNOT INよりNOT EXISTSが安全な場合がある主な理由はどれか。

ア：NOT EXISTSは必ず全表を削除するから。

イ：NOT INは集約関数でしか使えないから。

ウ：NOT INはNULLを含む比較でUNKNOWNが生じ、意図した行が返らない場合があるから。

エ：NOT EXISTSは主キーを自動生成するから。

答え・解説 正答：ウ

ア：NOT EXISTSは検索条件であり削除処理ではない。

イ：NOT INは一般の比較条件として使える。

ウ：SQLの三値論理ではNULLとの比較がUNKNOWNになり、NOT INの結果へ影響することがある。

エ：キー生成機能ではない。

総合解説：SQLの三値論理ではNULLとの比較がUNKNOWNになり、NOT INの結果へ影響することがある。

対象：2026年度 / 基準日 2026-09-02

0068 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：標準

二つの問合せ結果を結合し、重複行を除きたい。最も適切な集合演算はどれか。

ア：UNION ALL。

イ：CROSS JOIN。

ウ：UPDATE。

エ：UNION。

答え・解説 正答：エ

ア：UNION ALLは重複を保持する。

イ：直積を生成する。

ウ：データ更新文である。

エ：UNIONは通常、和集合として重複行を除去する。

総合解説：UNIONは通常、和集合として重複行を除去する。

対象：2026年度 / 基準日 2026-09-02

0069 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：応用

注文表から顧客別に「完了注文件数」と「取消注文件数」を同じ行へ集計したい。最も適切な考え方はどれか。

ア：SUM(CASE WHEN status=... THEN 1 ELSE 0 END)のような条件付き集計を状態ごとに行う。

イ：statusをGROUP BYから必ず除きCOUNT(*)だけを一つ返す。

ウ：各状態を文字列連結して件数とみなす。

エ：注文IDを全て同じ値に更新する。

答え・解説 正答：ア

ア：条件付き集計なら顧客単位の一行に状態別件数を列として算出できる。

イ：COUNT(*)一つでは状態別件数を分けられない。

ウ：文字列連結は件数集計ではない。

エ：識別子更新は不要でデータを破壊する。

総合解説：条件付き集計なら顧客単位の一行に状態別件数を列として算出できる。

対象：2026年度 / 基準日 2026-09-02

0070 6-3 SQL・トランザクション事例 > SQL読解・SQL完成 | 難易度：応用

注文(order_id,store_id)と店舗内注文属性(store_id,order_id,...)を結合したところ、意図せず行が増えた。注文番号order_idは店舗内でのみ一意である。原因として最も可能性が高いものはどれか。

ア：SELECT句に列を多く書いた。

イ：結合条件にstore_idを含めずorder_idだけで結合した。

ウ：ORDER BYを指定した。

エ：WHEREで日付条件を追加した。

答え・解説 正答：イ

ア：列数は結合一致件数を増やす直接原因ではない。

イ：order_idが全体一意でないなら、複合識別子の一部だけで結合すると別店舗の行まで一致して重複が生じる。

ウ：並べ替えは件数を変えない。

エ：絞込みは通常件数を減らす方向である。

総合解説：order_idが全体一意でないなら、複合識別子の一部だけで結合すると別店舗の行まで一致して重複が生じる。

対象：2026年度 / 基準日 2026-09-02

0071 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：基礎

二つのトランザクションが同じ在庫数100を読み、それぞれ10減算と20減算を計算して110ではなく最終的に80または90の片方だけが反映された。発生した問題として最も適切なものはどれか。

- ア：ダーティリード。
- イ：ファントムリード。
- ウ：ロストアップデート。
- エ：参照整合性違反。

答え・解説 正答：ウ

- ア：未コミット値を読む現象ではない。
 - イ：同一検索条件で行集合が増減する現象ではない。
 - ウ：双方が古い値を基に更新し、一方の更新結果が他方に上書きされて失われる現象である。
 - エ：外部キー参照の問題ではない。
- 総合解説：双方が古い値を基に更新し、一方の更新結果が他方に上書きされて失われる現象である。

対象：2026年度 / 基準日 2026-09-02

0072 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：基礎

T1が未コミットで残高を更新し、T2がその値を読み取った後、T1がROLLBACKした。T2が読んだ現象として最も適切なものはどれか。

- ア：非再現読取り。
- イ：デッドロック。
- ウ：ロストアップデート。
- エ：ダーティリード。

答え・解説 正答：エ

- ア：同一行を二度読んだ結果の変化を指す現象ではない。
 - イ：相互待ちではない。
 - ウ：更新の上書き損失ではない。
 - エ：コミットされていない変更を別トランザクションが読んだためダーティリードである。
- 総合解説：コミットされていない変更を別トランザクションが読んだためダーティリードである。

対象：2026年度 / 基準日 2026-09-02

0073 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：基礎

T1が顧客Aのランクを読み、その後T2が同じ顧客Aのランクを更新してCOMMITし、T1が再度読むと値が変わっていた。最も適切な現象はどれか。

- ア：非再現読取り。
- イ：ファントムリード。
- ウ：ダーティライト。
- エ：外部キー循環。

答え・解説 正答：ア

ア：同一トランザクション内で同じ行を再読した結果が変わる現象である。
イ：検索条件に該当する行の集合自体が増減する現象とは異なる。
ウ：未コミット同士の更新競合を指す表現であり事例と異なる。
エ：スキーマ参照関係の問題ではない。
総合解説：同一トランザクション内で同じ行を再読した結果が変わる現象である。

対象：2026年度 / 基準日 2026-09-02

0074 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：標準

T1が「金額10万円以上の注文件数」を検索した後、T2が条件を満たす新規注文を挿入してCOMMITし、T1が同じ検索をすると件数が増えた。最も適切な現象はどれか。

- ア：ロストアップデート。
- イ：ファントムリード。
- ウ：ダーティリード。
- エ：チェック制約違反。

答え・解説 正答：イ

ア：更新結果の上書き損失ではない。
イ：範囲条件に該当する行集合が他トランザクションの挿入などで変化する現象である。
ウ：未コミット値を読んだとは限らない。
エ：値域制約の問題ではない。
総合解説：範囲条件に該当する行集合が他トランザクションの挿入などで変化する現象である。

対象：2026年度 / 基準日 2026-09-02

0075 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：標準

T1が行Aをロック後に行Bを待ち、T2が行Bをロック後に行Aを待っている。最も適切な説明はどれか。

- ア：両者は必ず直列化可能なので待ちは発生しない。
- イ：ロックが一つずつなのでデッドロックではない。
- ウ：循環待ちが形成され、デッドロック状態である。
- エ：これはファントムリードである。

答え・解説 正答：ウ

- ア：循環待ちがあるため処理は進まない。
 - イ：各トランザクションが複数資源を跨いで相互待ちしている。
 - ウ：T1→B→T2→A→T1という待ちの循環があり、典型的なデッドロックである。
 - エ：読取り行集合の増減ではない。
- 総合解説：T1→B→T2→A→T1という待ちの循環があり、典型的なデッドロックである。

対象：2026年度 / 基準日 2026-09-02

0076 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：標準

複数口座間振替で、口座XとYを更新するトランザクション同士のデッドロックを減らしたい。最も一般的で有効な設計はどれか。

- ア：各トランザクションが任意の順でロックする。
- イ：ロック待ち時間を無限にする。
- ウ：全ての更新を未コミットのまま放置する。
- エ：更新対象口座を口座ID順など一貫した順序でロック取得する。

答え・解説 正答：エ

- ア：逆順取得が混在すると循環待ちが起きやすい。
 - イ：待ち時間延長は循環待ち自体を除去しない。
 - ウ：ロック保持が長くなり状況を悪化させる。
 - エ：資源取得順序を統一すると循環待ちを形成しにくくなり、デッドロック予防に有効である。
- 総合解説：資源取得順序を統一すると循環待ちを形成しにくくなり、デッドロック予防に有効である。

対象：2026年度 / 基準日 2026-09-02

0077 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：標準

二相ロック(2PL)の基本的な考え方として最も適切なものはどれか。

- ア：ロックを獲得する成長相と、解放だけを行う縮小相を分け、縮小後に新たなロックを取得しない。
- イ：ロックを取得した直後に必ず全て解放する。
- ウ：読取り処理では一切ロック概念を考えない。
- エ：トランザクションごとにDBを物理分割する。

答え・解説 正答：ア

- ア：2PLはロック獲得と解放の順序を二つの相に分け、競合スケジュールの直列化可能性を支える代表的方式である。
 - イ：必要な保護期間を確保できず、2PLの定義にも反する。
 - ウ：方式や実装により読取り保護も重要である。
 - エ：物理分割は2PLの定義ではない。
- 総合解説：2PLはロック獲得と解放の順序を二つの相に分け、競合スケジュールの直列化可能性を支える代表的方式である。

対象：2026年度 / 基準日 2026-09-02

0078 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：標準

業務が強い一貫性を求めるため直列化可能な実行に近づける分離レベルを採用する。一般的なトレードオフとして最も適切なものはどれか。

- ア：分離性を上げるほど必ず性能も無条件に向上する。
- イ：不整合な並行現象を抑えやすい一方、競合時の待ち・再実行などで並行性やスループットが低下する場合がある。
- ウ：分離レベルは結果整合性と無関係で、ログ容量だけに影響する。
- エ：分離性を上げると主キーが不要になる。

答え・解説 正答：イ

- ア：競合が多い場合は性能低下もあり得る。
 - イ：強い分離は整合性を高める反面、競合制御コストが増える可能性がある。
 - ウ：分離レベルは並行実行時の可視性・整合性に直接関係する。
 - エ：キー制約とは別概念である。
- 総合解説：強い分離は整合性を高める反面、競合制御コストが増える可能性がある。

対象：2026年度 / 基準日 2026-09-02

0079 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：応用

更新競合が少ないWeb業務で、読み取り時には長時間ロックせず、更新時に「読み取った版から変更されていないか」を確認して競合時だけやり直したい。最も近い方式はどれか。

ア：悲観的に全テーブルを長時間排他ロックする方式。

イ：バックアップ・リストア。

ウ：楽観的同時実行制御。

エ：ハッシュパーティション。

答え・解説 正答：ウ

ア：競合を前提に先にロックする方式で要件と逆である。

イ：障害回復方式であり並行制御ではない。

ウ：版番号や更新時刻を比較し、競合が判明したとき再実行する考え方は楽観的制御の典型である。

エ：データ配置方式である。

総合解説：版番号や更新時刻を比較し、競合が判明したとき再実行する考え方は楽観的制御の典型である。

対象：2026年度 / 基準日 2026-09-02

0080 6-3 SQL・トランザクション事例 > 排他制御・分離レベル・デッドロック | 難易度：応用

行ロックと表ロックを比較した一般的な説明として最も適切なものはどれか。

ア：行ロックは常に表ロックより管理コストが小さい。

イ：表ロックは同時更新可能性を必ず最大化する。

ウ：ロック粒度は並行性と無関係である。

エ：行ロックは競合範囲を小さくできる一方、多数ロックの管理コストが増える場合がある。

答え・解説 正答：エ

ア：多数行を扱うとロック管理量が増え得る。

イ：表全体を占有すると競合範囲は広くなる。

ウ：粒度は競合範囲と並行性へ影響する。

エ：細粒度ロックは並行性を高めやすいが、ロック数増加による管理コストとのトレードオフがある。

総合解説：細粒度ロックは並行性を高めやすいが、ロック数増加による管理コストとのトレードオフがある。

対象：2026年度 / 基準日 2026-09-02

0081 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：基礎

二つの独立DBをまたぐ一つの業務更新を「両方コミットするか両方ロールバックするか」にしたい。2相コミットを用いる主な目的はどれか。

ア：複数の参加資源にまたがるトランザクションの原子性を協調して確保する。

イ：全参加DBの読取り性能を必ず向上させる。

ウ：ネットワーク障害を物理的に防止する。

エ：各DBの主キーを不要にする。

答え・解説 正答：ア

ア：2PCは複数資源に対するグローバルトランザクションのコミット判断を協調し、原子性を実現するための代表的方式である。

イ：性能向上が主目的ではなく、むしろ調整コストがある。

ウ：通信障害そのものを無くす方式ではない。

エ：キー制約とは無関係である。

総合解説：2PCは複数資源に対するグローバルトランザクションのコミット判断を協調し、原子性を実現するための代表的方式である。

対象：2026年度 / 基準日 2026-09-02

0082 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：基礎

2相コミットの第1相でコーディネータが参加者へ確認する内容として最も適切なものはどれか。

ア：今後一切障害が起きないことを保証できるか。

イ：そのトランザクションをコミット可能な状態まで準備できたか。

ウ：全データを他参加者へ複製済みか。

エ：全SQLの実行計画が同一か。

答え・解説 正答：イ

ア：将来の障害ゼロは保証できない。

イ：第1相は各参加者がコミット可能な準備状態に入れるかを投票・確認する段階である。

ウ：全データ複製は2PCの必須要件ではない。

エ：実行計画一致はコミット判定条件ではない。

総合解説：第1相は各参加者がコミット可能な準備状態に入れるかを投票・確認する段階である。

対象：2026年度 / 基準日 2026-09-02

0083 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：基礎

参加者がPREPARED相当の状態に入った直後、最終決定を受け取る前にコーディネータと通信できなくなった。2PCの一般的な問題として最も適切なものはどれか。

ア：参加者は必ず独自判断でコミットしてよい。

イ：自動的に全参加者が即座にロールバックすることが保証される。

ウ：参加者が最終決定を確定できず、資源を保持したまま待機するブロッキングが起こり得る。

エ：この状況ではデッドロック検出だけで必ず解決する。

答え・解説 正答：ウ

ア：独自コミットするとグローバル原子性を壊す可能性がある。

イ：既にcommit決定がなされていた可能性もあり一律rollbackはできない。

ウ：prepare後は独断で決定を変えられず、コーディネータの決定が不明な間は待機が必要になる場合がある。

エ：通信・コミット決定の不確実性はローカルデッドロックとは別問題である。

総合解説：prepare後は独断で決定を変えられず、コーディネータの決定が不明な間は待機が必要になる場合がある。

対象：2026年度 / 基準日 2026-09-02

0084 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：標準

注文確定イベントがネットワーク再送によって同じevent_idで複数回配送される可能性がある。二重請求を防ぐ設計として最も適切なものはどれか。

ア：同じイベントを受け取るたび必ず請求を追加する。

イ：event_idを無視し本文だけ比較する。

ウ：重複が起きないと仮定して制御を入れない。

エ：受信側でevent_idの処理済み記録を持ち、同一イベントを重複適用しない冪等な処理にする。

答え・解説 正答：エ

ア：再送のたび副作用が重複する。

イ：本文差や順序で誤判定し得る。

ウ：ネットワーク再送の現実的な故障モデルに弱い。

エ：少なくとも一回配送では重複が起こり得るため、イベント識別子を使って再適用を防ぐ設計が有効である。

総合解説：少なくとも一回配送では重複が起こり得るため、イベント識別子を使って再適用を防ぐ設計が有効である。

対象：2026年度 / 基準日 2026-09-02

0085 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：標準

非同期レプリケーションされた複数拠点で、更新直後は値が異なる可能性があるが、更新が止まれば最終的に同じ状態へ収束する性質として最も適切なものはどれか。

ア：結果整合性（eventual consistency）。

イ：厳密な直列化可能性。

ウ：参照整合性。

エ：完全同期バックアップ。

答え・解説 正答：ア

ア：一時的不一致を許容しつつ最終的な収束を目指す性質を結果整合性という。

イ：全操作が単一の直列順序と同等であることを求める強い性質である。

ウ：外部キーなど参照関係の整合性を指す。

エ：バックアップ方式の名称ではない。

総合解説：一時的不一致を許容しつつ最終的な収束を目指す性質を結果整合性という。

対象：2026年度 / 基準日 2026-09-02

0086 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：標準

長時間業務を複数サービスのローカルトランザクションへ分け、途中失敗時には既に完了した処理を業務的に取り消す方式を採る。最も重要な考え方はどれか。

ア：全サービスを一つの共有テーブルへ統合することだけを行う。

イ：各ステップに対応する補償処理を設計し、失敗時に必要な逆操作を実行する。

ウ：途中失敗を無視して後続処理を続ける。

エ：全ステップのログを削除する。

答え・解説 正答：イ

ア：共有表化は補償ロジックそのものではない。

イ：Sagaでは各ローカルコミットを後から物理ROLLBACKできないため、業務上の補償操作で整合を回復する。

ウ：部分完了状態が残る整合性を損ねる。

エ：監査と復旧情報を失う。

総合解説：Sagaでは各ローカルコミットを後から物理ROLLBACKできないため、業務上の補償操作で整合を回復する。

対象：2026年度 / 基準日 2026-09-02

0087 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：標準

注文DBへのCOMMITには成功したが、その直後にイベントブローカーへの送信前にアプリが停止し、下流へ注文確定が伝わらない問題を減らしたい。最も適切な設計はどれか。

ア：DB更新後にメモリ上のフラグだけを立てる。

イ：イベント送信を先に行い、その成否に関係なくDB更新を省略する。

ウ：注文更新と同一ローカルトランザクションでOutbox行も記録し、別処理がOutboxから確実にイベント送信する。

エ：障害時は注文行を手作業で推測して再作成する。

答え・解説 正答：ウ

ア：プロセス停止でフラグを失う。

イ：イベントだけ送られてDB更新されない逆の不整合が起こり得る。

ウ：Outboxを同一DBトランザクションへ含めると、業務更新と送信予定記録の原子性を確保しやすい。

エ：再現性・監査性が低く自動復旧できない。

総合解説：Outboxを同一DBトランザクションへ含めると、業務更新と送信予定記録の原子性を確保しやすい。

対象：2026年度 / 基準日 2026-09-02

0088 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：標準

複数拠点がネットワーク分断中でも注文を受け付け、後で統合する。注文IDの衝突を避けたい。最も適切な方針はどれか。

ア：各拠点で1から同じ連番だけを発行する。

イ：注文日時 of 「日」だけをIDにする。

ウ：顧客名を注文IDにする。

エ：拠点識別子を含むIDや十分な一意性を持つ分散ID生成方式を採用する。

答え・解説 正答：エ

ア：統合時に同じ番号が衝突する。

イ：同日に多数注文があるため一意にならない。

ウ：同姓同名や名称変更があり一意・安定でない。

エ：中央採番に常時依存できない状況では、拠点を跨いで衝突しにくい識別方式が必要である。

総合解説：中央採番に常時依存できない状況では、拠点を跨いで衝突しにくい識別方式が必要である。

対象：2026年度 / 基準日 2026-09-02

0089 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：応用

利用者がプロフィールを更新した直後の確認画面では、必ず自分の最新値を見せたい。一方、通常参照は遅延する非同期レプリカでもよい。最も適切な設計はどれか。

ア：更新直後の確認は書き込み先または最新性を保証できる経路から読み、通常参照だけレプリカへ分散する。

イ：確認画面も必ず遅延レプリカだけから読む。

ウ：更新後は数分待つことだけを仕様にする。

エ：プロフィールをDBに保存せず画面だけ更新する。

答え・解説 正答：ア

ア：read-after-write要件がある箇所だけ強い最新性を確保し、他はレプリカ活用することで整合性と性能を両立しやすい。

イ：反映遅延で古い値を表示する可能性がある。

ウ：利用者要件をシステムで保証していない。

エ：永続データと画面表示が不一致になる。

総合解説：read-after-write要件がある箇所だけ強い最新性を確保し、他はレプリカ活用することで整合性と性能を両立しやすい。

対象：2026年度 / 基準日 2026-09-02

0090 6-3 SQL・トランザクション事例 > 分散トランザクション・整合性 | 難易度：応用

決済API呼出しがタイムアウトした。実際には決済先で処理が成功している可能性もある。単純再試行で二重決済を避けるため最も適切な設計はどれか。

ア：タイムアウトしたら必ず未処理と決めつけ新規IDで再送する。

イ：要求ごとの冪等性キーを用い、再試行時に同一業務要求として決済先が重複実行を抑止できるようにする。

ウ：結果確認をせず別金額で再送する。

エ：通信エラー時は顧客IDを変更する。

答え・解説 正答：イ

ア：既に成功済みなら二重決済になる。

イ：タイムアウトは相手側成否が不確実なので、同一要求を識別して副作用を一度に抑える冪等設計が重要である。

ウ：業務意味が変わり整合性を壊す。

エ：顧客識別変更は重複実行対策にならない。

総合解説：タイムアウトは相手側成否が不確実なので、同一要求を識別して副作用を一度に抑える冪等設計が重要である。

対象：2026年度 / 基準日 2026-09-02

0091 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：基礎

受注表は1億行あり、日次処理では「顧客IDを等価条件、受注日を範囲条件」として検索する。ほかの検索条件はほとんどない。候補の中で最も目的に合う索引はどれか。

ア：(顧客ID, 受注日)の複合索引

イ：(受注日, 顧客ID)の複合索引

ウ：受注明細件数だけの索引

エ：索引を作らず毎回全表走査する

答え・解説 正答：ア

ア：顧客IDの等価条件で絞り込み、その後受注日の範囲を利用しやすい列順であり、この検索パターンに適する。

イ：受注日が広い範囲条件になると、その後の顧客IDを効率よく利用できない場合があり、主検索パターンには劣る。

ウ：検索条件に使わない列なのでアクセス対象の絞り込みに寄与しない。

エ：1億行から少数行を取得する定型検索では、適切な索引を検討するのが合理的である。

総合解説：顧客IDの等価条件で絞り込み、その後受注日の範囲を利用しやすい列順であり、この検索パターンに適する。

対象：2026年度 / 基準日 2026-09-02

0092 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：基礎

会員表1,000万行のうち約95%が status='ACTIVE' である。SELECT * FROM 会員 WHERE status='ACTIVE' を高速化したい。B-tree索引をstatusだけに作成しても全表走査が選ばれた。最も妥当な説明はどれか。

ア：B-tree索引は等価検索では利用できない。

イ：取得対象が表の大部分なので、索引経由のランダムアクセスより全表走査の方が低コストと見積もられ得る。

ウ：全表走査が選ばれるとSQLの結果件数が変わる。

エ：status列にNULLがないと索引は利用できない。

答え・解説 正答：イ

ア：B-treeは一般に等価検索にも利用できるため説明にならない。

イ：低選択性条件では大量行の表アクセスが必要となり、索引が必ず有利とは限らない。

ウ：アクセス経路は通常、結果の意味ではなく性能に影響する。

エ：NULLの有無はこの現象の本質ではない。

総合解説：低選択性条件では大量行の表アクセスが必要となり、索引が必ず有利とは限らない。

対象：2026年度 / 基準日 2026-09-02

0093 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：基礎

検索で必要な列が検索条件列と出力列の少数に限られ、表本体へのアクセスを減らしたい。物理設計上の考え方として最も適切なものはどれか。

ア：全列を毎回SELECTしてキャッシュに載せる。

イ：検索条件列の型を毎回実行時に変更する。

ウ：必要な列を索引だけで賄えるように設計し、索引のみで完結できるアクセス経路を検討する。

エ：索引を削除して必ずソート処理を発生させる。

答え・解説 正答：ウ

ア：不要列まで取得するとI/Oや転送量が増え、目的と逆行する。

イ：型変換は索引利用を妨げることがあり、表アクセス削減の方法ではない。

ウ：表本体へのアクセスを減らすカバリング索引 / index-only型の考え方に合う。

エ：索引削除や不要なソートは表本体アクセス削減にはつながらない。

総合解説：表本体へのアクセスを減らすカバリング索引 / index-only型の考え方に合う。

対象：2026年度 / 基準日 2026-09-02

0094 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：標準

実行計画ではある条件の推定行数が100行だが、実測では200万行だった。索引の定義自体は妥当で、データ分布が先月大きく変化している。最初に確認すべき対応はどれか。

ア：主キー制約を削除する。

イ：全テーブルを1列だけにする。

ウ：結果件数を固定値100件に書き換える。

エ：統計情報を更新し、カーディナリティ推定が改善するか再確認する。

答え・解説 正答：エ

ア：主キー削除は推定誤差への直接的な対応ではなく整合性も損なう。

イ：データモデルを壊す極端な変更であり、原因分析にならない。

ウ：要求する結果自体を変えるため正しい性能対策ではない。

エ：大きな推定誤差と分布変化があるため、まず統計情報の鮮度・精度を確認するのが筋である。

総合解説：大きな推定誤差と分布変化があるため、まず統計情報の鮮度・精度を確認するのが筋である。

対象：2026年度 / 基準日 2026-09-02

0095 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：標準

取引履歴を「口座ID=指定値」で絞り、取引日時の降順で最新20件だけ取得する処理が非常に多い。最も有力な索引設計はどれか。

ア：口座IDと取引日時を検索・並び順に合わせた複合索引にする。

イ：金額だけに索引を作る。

ウ：取引日時だけに索引を作り、全口座を最新順に読む。

エ：索引を作らず、毎回全件を並べ替える。

答え・解説 正答：ア

ア：等価条件で口座を絞り、取引日時順に読み出せるため、余分なソートや広い走査を減らせる。

イ：主要な絞込み条件にも並び順にも一致しない。

ウ：最新順には寄与しても口座IDでの絞込み効率が弱い。

エ：大量履歴から最新20件を頻繁に取る用途では不利になりやすい。

総合解説：等価条件で口座を絞り、取引日時順に読み出せるため、余分なソートや広い走査を減らせる。

対象：2026年度 / 基準日 2026-09-02

0096 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：標準

注文日時に通常の索引がある。WHERE DATE(注文日時)=DATE '2026-09-01' の検索で索引が有効利用されず遅い。一般的な改善方針として最も適切なものはどれか。

ア：注文日時を毎回文字列へ変換して比較する。

イ：注文日時 >= 日付開始 AND 注文日時 < 翌日開始 の範囲条件へ書き換える。

ウ：全行の注文日時をNULLにする。

エ：ORDER BY句を必ず追加する。

答え・解説 正答：イ

ア：さらに関数・型変換が加わり、索引利用を阻害し得る。

イ：索引列そのものに対する範囲条件にすれば、通常の索引を利用しやすくなる。

ウ：検索要件を破壊するだけである。

エ：問題は絞込み条件の形であり、並べ替え追加は解決にならない。

総合解説：索引列そのものに対する範囲条件にすれば、通常の索引を利用しやすくなる。

対象：2026年度 / 基準日 2026-09-02

0097 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：標準

照会性能を上げるため、更新頻度が非常に高いテーブルに10本の索引を追加する案が出た。設計レビューで最も重要な指摘はどれか。

ア：索引を増やすほど更新処理も必ず速くなる。

イ：索引の本数はトランザクション性能に一切影響しない。

ウ：索引は検索を助ける一方、INSERT・UPDATE・DELETE時の索引保守コストと領域消費を増やすため、必要性をワークロードで評価すべきである。

エ：索引があればバックアップは不要になる。

答え・解説 正答：ウ

ア：索引保守が増えるため一般には逆の影響もあり得る。

イ：更新I/Oやロック等に影響し得る。

ウ：索引は読み取り性能だけでなく更新コストとのトレードオフで判断する必要がある。

エ：索引とバックアップは目的が異なる。

総合解説：索引は読み取り性能だけでなく更新コストとのトレードオフで判断する必要がある。

対象：2026年度 / 基準日 2026-09-02

0098 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：標準

10年分の監査ログを月単位でパーティション分割している。通常の照会は直近1か月を対象にユーザーIDで検索する。最も合理的な設計方針はどれか。

ア：全期間を毎回1つの巨大ソートに掛ける。

イ：ユーザーIDを削除して日付だけ保存する。

ウ：全パーティションを毎日結合して単一表へ戻す。

エ：日付条件で対象パーティションを絞るようにし、その中でユーザーID検索に適した索引を用意する。

答え・解説 正答：エ

ア：直近1か月しか不要なのに全期間処理するのは非効率である。

イ：要件であるユーザー検索ができなくなる。

ウ：分割の利点を失い、保守負荷も増える。

エ：パーティションブルーニングと索引の双方を活用し、読むデータ範囲を段階的に減らせる。

総合解説：パーティションブルーニングと索引の双方を活用し、読むデータ範囲を段階的に減らせる。

対象：2026年度 / 基準日 2026-09-02

0099 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：応用

実行計画で、外側10行に対して内側の索引検索を10回行うNested Loopが選ばれている。別案は両表を大量走査してHash Joinする。外側の実測件数も10行で安定している。この条件だけから見た判断として最も適切なものはどれか。

ア：Nested Loopは外側がごく少数で内側を索引検索できるため合理的であり、Hash Joinへ固定変更する根拠は弱い。

イ：Hash Joinはどんなデータ量でも必ず最速である。

ウ：Nested Loopでは索引を利用できない。

エ：結合方式は性能に影響しないので確認不要である。

答え・解説 正答：ア

ア：結合方式は入力件数・アクセス方法で評価し、少数外側＋索引内側はNested Loopが適する典型例である。

イ：結合方式に万能なものはなく、入力規模やメモリ等で有利不利が変わる。

ウ：内側アクセスに索引を利用するNested Loopは一般的である。

エ：大量データでは結合方式が性能を大きく左右し得る。

総合解説：結合方式は入力件数・アクセス方法で評価し、少数外側＋索引内側はNested Loopが適する典型例である。

対象：2026年度 / 基準日 2026-09-02

0100 6-4 物理設計・運用事例 > インデックス・アクセス経路 | 難易度：応用

RDBMS更新後、同じSQLの応答時間が1秒から40秒へ悪化した。旧版では索引走査、新版では全表走査となり、推定行数が実測より大幅に多い。最も適切な調査順序はどれか。

ア：直ちに全SQLへ索引利用を強制し、原因調査は行わない。

イ：統計情報・推定行数と実行計画の差を確認し、索引定義やデータ分布、プランナ変更の影響を切り分ける。

ウ：データを全件削除して応答時間だけ測る。

エ：旧版に戻せない前提で監視を停止する。

答え・解説 正答：イ

ア：強制は一時対策になり得ても、根本原因や他SQLへの悪影響を把握できない。

イ：観測された計画変化と推定誤差に直接対応し、原因を段階的に切り分ける手順である。

ウ：本番相当のデータ分布を失い、原因分析として不適切である。

エ：監視停止は原因特定を難しくし、性能問題を解消しない。

総合解説：観測された計画変化と推定誤差に直接対応し、原因を段階的に切り分ける手順である。

対象：2026年度 / 基準日 2026-09-02

0101 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：基礎

ピーク時にDBサーバのCPU使用率は35%だが、ディスク待ち時間が急増し、読み取りレイテンシも悪化している。SQL待機の多くもI/O待ちである。最初に疑うべきボトルネックはどれか。

ア：CPU演算能力

イ：DNS解決

ウ：ストレージI/O

エ：画面の文字サイズ

答え・解説 正答：ウ

ア：CPU使用率35%という観測だけではCPU飽和を示さない。

イ：DB内部のI/O待ち増大を直接説明しにくい。

ウ：CPUには余力があり、待機・レイテンシがI/Oに集中しているため最有力である。

エ：サーバのI/Oレイテンシとは無関係である。

総合解説：CPUには余力があり、待機・レイテンシがI/Oに集中しているため最有力である。

対象：2026年度 / 基準日 2026-09-02

0102 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：基礎

オンライン照会の要件は「99%の要求を2秒以内」と定められている。性能試験結果を評価するとき、平均応答時間だけでなく特に確認すべき指標はどれか。

ア：平均応答時間。平均が2秒未満でも遅い1%を隠す可能性がある。

イ：中央値（50パーセンタイル）。典型値は示すが99%要件を直接判定できない。

ウ：最大応答時間。外れ値把握には使えるが「99%が2秒以内」の直接判定指標ではない。

エ：99パーセンタイル付近の応答時間。

答え・解説 正答：エ

ア：平均だけでは裾の遅延を把握できず、99%要件の可否を直接判定できない。

イ：中央値は半数点を示すため、上位1%付近の遅延を評価できない。

ウ：最大値は参考になるが、1件の極端値に左右され、99%達成条件そのものではない。

エ：99パーセンタイルが2秒以内かを確認すれば「99%の要求を2秒以内」という要件を直接評価できる。

総合解説：要件が99%という分位点で定義されているため、平均値ではなく分布の上側を確認する必要がある。

対象：2026年度 / 基準日 2026-09-02

0103 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：基礎

データ領域は現在7.2TB使用、上限9.0TBで、増加量は平均300GB/月で安定している。安全余裕を考えない単純計算では、上限まで何か月程度か。

ア：約6か月（ $(9.0-7.2)\text{TB} \div 0.3\text{TB/月}$ ）

イ：約2か月（残容量を0.9TBと取り違えた値）

ウ：約12か月（増加量を150GB/月と取り違えた値）

エ：約30か月（上限9.0TB全体を300GB/月で割った値）

答え・解説 正答：ア

ア：残容量は $1.8\text{TB}=1800\text{GB}$ で、 $1800/300=6$ か月である。

イ：残容量は0.9TBではなく1.8TBである。

ウ：与えられた増加量は300GB/月である。

エ：既使用7.2TBを差し引かず上限全体を割るのは誤りである。

総合解説：残り $1.8\text{TB}=約1,800\text{GB}$ を300GB/月で割ると約6か月である。

対象：2026年度 / 基準日 2026-09-02

0104 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：標準

接続プール上限を100から2,000へ増やしたところ、TPSはほぼ増えず、コンテキスト切替とロック待ちが増えて応答時間が悪化した。最も妥当な判断はどれか。

ア：接続数は多いほど必ずTPSが比例して増えるため、さらに増やす。

イ：同時実行数を増やし過ぎてDB内部競合が増えた可能性があり、適正なプール上限を負荷試験で求める。

ウ：ロック待ちは接続数と無関係なので無視する。

エ：全利用者へDB管理者権限を与える。

答え・解説 正答：イ

ア：観測結果に反し、飽和後は競合を悪化させ得る。

イ：並行性は増やせば無限に性能向上するわけではなく、CPU・ロック・メモリ競合で逆効果になり得る。

ウ：同時トランザクション増加に伴いロック競合が増えることはある。

エ：性能改善とは無関係でセキュリティを悪化させる。

総合解説：並行性は増やせば無限に性能向上するわけではなく、CPU・ロック・メモリ競合で逆効果になり得る。

対象：2026年度 / 基準日 2026-09-02

0105 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：標準

DBのバッファキャッシュヒット率が99.5%である。性能担当者の解釈として最も適切なものはどれか。

ア：99%以上ならすべてのSQLが最適であることを証明する。

イ：ヒット率が高いほど必ずディスク故障率が上がる。

ウ：多くの読み取りがキャッシュで満たされていることは示すが、それだけで性能問題がないとは断定できない。

エ：ヒット率が高ければバックアップは不要である。

答え・解説 正答：ウ

ア：ヒット率はSQL計画やロック競合の最適性を証明しない。

イ：キャッシュヒット率と物理故障率は別概念である。

ウ：CPU、ロック、書き込み、SQL計画など別のボトルネックが残るため単一指標で断定しない。

エ：可用性・復旧要件とは無関係である。

総合解説：CPU、ロック、書き込み、SQL計画など別のボトルネックが残るため単一指標で断定しない。

対象：2026年度 / 基準日 2026-09-02

0106 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：標準

毎日10:00～10:10だけ処理が遅い。原因を切り分けるための監視設計として最も有効なのはどれか。

ア：月平均CPU使用率だけを確認する。

イ：障害時間帯以外の画面デザインを確認する。

ウ：遅延発生中はログを止める。

エ：同時刻のSQL実行時間、待機イベント、CPU、I/O、ロック、ジョブ実行状況を時系列で突き合わせる。

答え・解説 正答：エ

ア：10分間の局所的な異常が平均で埋もれる可能性が高い。

イ：DB処理遅延の原因分析に結びつかない。

ウ：観測情報を失い、切り分けを困難にする。

エ：複数層のメトリクスを同じ時間軸で関連させることで原因候補を絞れる。

総合解説：複数層のメトリクスを同じ時間軸で関連させることで原因候補を絞れる。

対象：2026年度 / 基準日 2026-09-02

0107 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：標準

実測ピークは800TPSで、今後ピークが25%増える見込みである。さらに設計上20%の余裕を「予測ピークに対して」確保したい。最低限の処理能力目標として最も近いものはどれか。

ア：1,200TPS ($800 \times 1.25 \times 1.20$)

イ：960TPS (800×1.20 で将来25%増を反映していない)

ウ：1,000TPS (800×1.25 で設計余裕20%を反映していない)

エ：1,500TPS (予測ピークに50%の余裕を加えた値)

答え・解説 正答：ア

ア：予測ピークは $800 \times 1.25 = 1000$ TPS、その20%余裕を含め $1000 \times 1.20 = 1200$ TPSである。

イ：現在ピークへの20%余裕だけで、将来増加を含めていない。

ウ：将来25%増だけを反映し、追加余裕20%を含めていない。

エ：指定された余裕率は20%であり、この値は過大である。

総合解説： $800 \times 1.25 = 1,000$ TPS、そこへ20%余裕を加えると1,200TPSである。

対象：2026年度 / 基準日 2026-09-02

0108 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：標準

大規模集計の実行計画を確認すると、ソート処理がメモリ内で完結せず一時ファイルへ退避しており、その時間帯に一時領域I/Oが急増している。最も直接的な改善検討はどれか。

ア：主キーをすべて削除する。

イ：対象SQLのデータ量・ソート方式を見直し、必要に応じて安全な範囲で作業メモリや索引・クエリ設計を調整する。

ウ：バックアップ保存期間を0日にする。

エ：ネットワークDNSを変更する。

答え・解説 正答：イ

ア：一時ソートの原因と直接関係せず整合性を損なう。

イ：一時ファイルへのspillが観測されているため、ソート量と利用可能メモリ、索引設計を合わせて評価する。

ウ：一時I/Oの性能問題を解決しない。

エ：DB内部のソートspillとは直接関係しない。

総合解説：一時ファイルへのspillが観測されているため、ソート量と利用可能メモリ、索引設計を合わせて評価する。

対象：2026年度 / 基準日 2026-09-02

0109 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：応用

負荷試験で同時利用者100人では1,000TPS・平均0.3秒、200人では1,450TPS・0.6秒、400人では1,470TPS・2.8秒、800人では1,430TPS・8.5秒だった。最も妥当な評価はどれか。

ア：800人が最も応答時間が長いので処理能力も最大である。

イ：利用者を増やせばTPSは無限に比例増加する傾向である。

ウ：200～400人の間でスループットが頭打ちになり、以降は待ち時間だけが增える飽和状態に入っている可能性が高い。

エ：このデータから性能限界を検討する必要はない。

答え・解説 正答：ウ

ア：応答時間増加と処理能力最大は同義ではなく、TPSはむしろ低下している。

イ：400人以降の結果が比例増加を否定している。

ウ：TPSがほぼ増えず応答時間が急増しているため、資源または競合の飽和を示す典型的な形である。

エ：明確なスループット頭打ちと遅延悪化が観測されている。

総合解説：TPSがほぼ増えず応答時間が急増しているため、資源または競合の飽和を示す典型的な形である。

対象：2026年度 / 基準日 2026-09-02

0110 6-4 物理設計・運用事例 > 性能・ボトルネック・キャパシティ | 難易度：応用

あるAPIの到着率はほぼ一定なのに、平均DB応答時間が0.2秒から2秒へ10倍になり、同時実行中の要求数も約10倍になった。新規要求数の急増はない。この観測から最も妥当な説明はどれか。

ア：同時実行数が増えたので必ず入力トラフィックも10倍になっている。

イ：応答時間が長いほどDB性能は良い。

ウ：要求滞留と待ち時間には関係がない。

エ：処理時間増大によりシステム内に滞留する要求が増えており、原因となる待ち・資源競合を調べるべきである。

答え・解説 正答：エ

ア：到着率一定という条件と矛盾する。

イ：応答時間悪化は性能劣化を示す。

ウ：待ち行列系では処理時間と滞留量は密接に関係する。

エ：到着率が同程度なら応答時間増大に伴って平均滞留数も増えるため、待機時間の原因分析が必要である。

総合解説：到着率が同程度なら応答時間増大に伴って平均滞留数も増えるため、待機時間の原因分析が必要である。

対象：2026年度 / 基準日 2026-09-02

0111 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：基礎

業務要件は「データ損失を最大5分以内に抑える」である。毎日深夜に1回だけフルバックアップを取る方式について最も適切な評価はどれか。

ア：単独ではRPO 5分を満たせないため、ログ継続保全などより短い間隔の復旧手段を追加検討する必要がある。

イ：フルバックアップがあれば取得間隔に関係なくRPOは0になる。

ウ：RPOは復旧作業に要する時間だけを表す。

エ：RPO要件はバックアップ設計と無関係である。

答え・解説 正答：ア

ア：日次バックアップだけでは最大約24時間分を失う可能性があり、5分要件には不足する。

イ：バックアップ間の更新を失う可能性があるため誤り。

ウ：復旧時間の目標はRTOであり、RPOは許容データ損失量を時間で表す。

エ：バックアップ・ログ保全方式を決める主要要件の一つである。

総合解説：日次バックアップだけでは最大約24時間分を失う可能性があり、5分要件には不足する。

対象：2026年度 / 基準日 2026-09-02

0112 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：基礎

「障害発生から30分以内にサービスを再開する」という要件は、主にどの指標を示すか。

ア：RPO

イ：RTO

ウ：MTBF

エ：選択度

答え・解説 正答：イ

ア：RPOは許容できるデータ損失の時間幅を示す。

イ：RTOは障害後に業務・サービスを復旧させるまでの目標時間を示す。

ウ：MTBFは平均故障間隔であり、復旧時間目標ではない。

エ：検索条件の絞込み度合いに関する概念である。

総合解説：RTOは障害後に業務・サービスを復旧させるまでの目標時間を示す。

対象：2026年度 / 基準日 2026-09-02

0113 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：基礎

旧DBから新DBへ1,000万件を移行し、移行前後の件数は一致した。移行検証として最も適切な次の確認はどれか。

ア：件数が一致した時点で内容検証は不要である。

イ：新DBのバックアップを削除する。

ウ：キー別件数、金額合計、ハッシュ・サンプル照合、制約違反など内容面の整合も確認する。

エ：移行ログを残さない。

答え・解説 正答：ウ

ア：同件数でも値の誤変換や行の入替えは起こり得る。

イ：検証を弱めるだけである。

ウ：件数一致だけでは値変換誤りや重複・欠損を十分検出できない。

エ：追跡・再実行・原因分析が困難になる。

総合解説：件数一致だけでは値変換誤りや重複・欠損を十分検出できない。

対象：2026年度 / 基準日 2026-09-02

0114 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：標準

14:03に誤った一括UPDATEが実行された。13:00のベースバックアップと継続保存された更新ログがあり、14:02:59時点まで戻したい。最も適切な考え方はどれか。

ア：14:03以降のログだけを空のDBへ適用する。

イ：誤更新後のDBをそのまま正として利用する。

ウ：索引だけを復元すればデータ値も元に戻る。

エ：ベースバックアップを復元し、保存ログを目標時刻直前まで再適用する。

答え・解説 正答：エ

ア：基礎となるデータ状態がなければ完全な復旧にならない。

イ：要件である誤更新直前への復旧を満たさない。

ウ：索引はデータ本体の誤更新を巻き戻すものではない。

エ：PITRは基点となるバックアップにログを再生して指定時点まで進める。

総合解説：PITRは基点となるバックアップにログを再生して指定時点まで進める。

対象：2026年度 / 基準日 2026-09-02

0115 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：標準

遠隔拠点への同期レプリケーション導入で書き込み応答時間が大きく増えた。RPOをほぼ0にしたい要件は強い。最も適切な説明はどれか。

ア：遠隔側の確認を待つ同期方式はデータ損失リスクを抑えやすい一方、ネットワーク遅延が書き込み応答へ反映されるトレードオフがある。

イ：同期レプリケーションならネットワーク遅延は応答時間に影響しない。

ウ：非同期方式なら常にRPO=0である。

エ：レプリケーションはバックアップや復旧設計を完全に不要にする。

答え・解説 正答：ア

ア：同期複製の耐障害性とレイテンシの関係を正しく捉えている。

イ：確認待ちがある構成では遅延の影響を受け得る。

ウ：未転送・未適用データを失う可能性がある。

エ：論理誤操作や同時破損などに備え、別のバックアップは依然重要である。

総合解説：同期複製の耐障害性とレイテンシの関係を正しく捉えている。

対象：2026年度 / 基準日 2026-09-02

0116 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：標準

ネットワーク分断時に旧プライマリと新プライマリの双方が更新を受け付け、データが分岐する事故を防ぎたい。最も重要な設計観点はどれか。

ア：両ノードへ常に自由に書き込ませ、後で人手で直す。

イ：クォーラムやフェンシング等で、同時に複数ノードがプライマリとして更新受付しない仕組みを設ける。

ウ：監視を停止して分断を検知しない。

エ：バックアップファイル名を同じにする。

答え・解説 正答：イ

ア：分岐と競合を増やし、整合性リスクが高い。

イ：スプリットブレイン防止には「誰が書き込み権限を持つか」を一意に決める仕組みが必要である。

ウ：障害認識を遅らせるだけである。

エ：プライマリ選出・書き込み権限の制御とは無関係である。

総合解説：スプリットブレイン防止には「誰が書き込み権限を持つか」を一意に決める仕組みが必要である。

対象：2026年度 / 基準日 2026-09-02

0117 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：標準

移行対象8TB、実効転送・変換速度が200MB/sで、停止可能時間は6時間である。単純転送だけでも約11時間かかる見込みである。最も適切な対応はどれか。

ア：6時間を超えても計画どおり一括移行を始め、終了時刻は考えない。

イ：データ件数を検証せず一部を捨てる。

ウ：事前同期や差分移行、並列化などで停止時間中に移すデータ量を減らし、リハーサルで実効時間を再見積もる。

エ：転送速度は見積りに使わず、経験だけで判断する。

答え・解説 正答：ウ

ア：業務停止要件を満たせない。

イ：データ完全性を損なう。

ウ：現行方式は停止窓を超えるため、移行方式自体を見直し、実測で成立性を確認する必要がある。

エ：容量・回線能力・変換性能は移行時間の主要要因である。

総合解説：現行方式は停止窓を超えるため、移行方式自体を見直し、実測で成立性を確認する必要がある。

対象：2026年度 / 基準日 2026-09-02

0118 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：標準

毎日バックアップジョブは「成功」と記録されているが、過去1年間復元テストをしていない。監査で最も適切な指摘はどれか。

ア：ジョブ成功ログがあれば復元テストは不要である。

イ：復元試験は本番障害時に初めて行う。

ウ：バックアップを暗号化すれば復元手順確認は不要になる。

エ：取得成功だけでは復旧可能性を保証できないため、定期的な復元・リカバリ試験でRTO/RPOを含めて確認する。

答え・解説 正答：エ

ア：媒体破損、手順不備、権限不足などは取得成功だけで検出できない。

イ：手順未検証のまま本番障害を迎えるリスクが高い。

ウ：暗号化と復元可能性の検証は別問題である。

エ：バックアップは実際に復元でき、所要時間が要件内であることまで確認して初めて復旧設計として信頼できる。

総合解説：バックアップは実際に復元でき、所要時間が要件内であることまで確認して初めて復旧設計として信頼できる。

対象：2026年度 / 基準日 2026-09-02

0119 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：応用

新DBへ継続同期しながら旧DBを本番利用し、短時間停止で新DBへ切り替える計画である。切替判定として最も適切なものはどれか。

ア：同期遅延が許容範囲内で、差分適用完了、整合性照合、性能確認、ロールバック条件・手順まで満たしたことを確認して切り替える。

イ：新DBが起動した時点で即時切替し、整合性確認は後日行う。

ウ：ロールバック手順は障害が起きてから決める。

エ：同期遅延は切替判断に関係しない。

答え・解説 正答：ア

ア：切替はデータ整合・性能・復旧可能性を複合的に確認して判断すべきである。

イ：データ欠損や不整合を本番化する危険がある。

ウ：切替前に失敗時の戻し方と判断条件を決める必要がある。

エ：未反映データが残る可能性があるため重要な指標である。

総合解説：切替はデータ整合・性能・復旧可能性を複合的に確認して判断すべきである。

対象：2026年度 / 基準日 2026-09-02

0120 6-4 物理設計・運用事例 > 障害回復・可用性・データ移行 | 難易度：応用

主拠点障害でスタンバイへ切替可能だが、スタンバイの最終受信ログが30秒前である。業務要件はRPO 1分、RTO 10分。主拠点は安全確認に数時間掛かる。最も妥当な判断はどれか。

ア：RPO 1分なので30秒の遅延でも必ず切替禁止である。

イ：30秒の遅延が事実として確認でき、整合性確認と切替手順が満たされるなら、RPO/RTOの両要件内でスタンバイ切替を進める判断が合理的である。

ウ：主拠点が戻るまで数時間待つのがRTO 10分を満たす。

エ：RTOとRPOは災害時の判断に使わない。

答え・解説 正答：イ

ア：30秒は許容損失1分以内である。

イ：30秒の潜在損失はRPO 1分内で、主拠点復旧待ちはRTO 10分を超えるためスタンバイ利用が要件に合う。

ウ：明らかにRTOを超える。

エ：復旧方式・切替判断の主要基準である。

総合解説：30秒の潜在損失はRPO 1分内で、主拠点復旧待ちはRTO 10分を超えるためスタンバイ利用が要件に合う。

対象：2026年度 / 基準日 2026-09-02

0121 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：基礎

性能試験表に「目標: $p95 \leq 1.0$ 秒、試験結果: $p95 = 1.4$ 秒、平均=0.5秒」とある。最も適切な判定はどれか。

ア：平均0.5秒なので合格である。

イ：p95は平均より大きいので測定ミスである。

ウ：平均は良好でもp95要件を満たしていないため不合格である。

エ：性能要件は数値化されていても判定に使わない。

答え・解説 正答：ウ

ア：平均値は今回の合否基準ではない。

イ：分布の上位点であるp95が平均より大きいのは自然である。

ウ：判定基準はp95であり、平均値が小さくても目標違反は解消されない。

エ：仕様に明記された数値は受入判定に用いるべきである。

総合解説：判定基準はp95であり、平均値が小さくても目標違反は解消されない。

対象：2026年度 / 基準日 2026-09-02

0122 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：基礎

ログは 10:00:01「DB接続タイムアウト増加」、10:00:03「接続プール使用率100%」、10:00:20「アプリ再試行急増」と記録されている。少なくともこのログから直接言えることはどれか。

ア：再試行が必ず根本原因である。

イ：DBサーバのディスク故障が確定した。

ウ：接続プール使用率は常に0%だった。

エ：接続タイムアウトとプール飽和が、再試行急増より先に観測されている。

答え・解説 正答：エ

ア：時系列だけでは因果関係までは確定できない。

イ：ディスク障害を示すログは提示されていない。

ウ：提示されたログと反する。

エ：ログ時刻の順序から直接読み取れる事実である。

総合解説：ログ時刻の順序から直接読み取れる事実である。

対象：2026年度 / 基準日 2026-09-02

0123 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：基礎

仕様書には「参照は数秒の遅延を許容するが、注文確定時の在庫引当は二重引当を許容しない」とある。データ整合性設計で最も重要な読み取りはどれか。

ア：処理ごとに整合性要求が異なり、注文確定系では強い排他・整合性保証が必要である。

イ：すべての処理で数秒の不整合を許容できる。

ウ：在庫引当には排他制御が不要である。

エ：参照処理をすべて停止すべきである。

答え・解説 正答：ア

ア：仕様が参照系と更新系で異なる一貫性要求を明示している。

イ：在庫引当では許容しないと明記されている。

ウ：二重引当禁止という要件と矛盾する。

エ：仕様からその必要性は導けない。

総合解説：仕様が参照系と更新系で異なる一貫性要求を明示している。

対象：2026年度 / 基準日 2026-09-02

0124 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：標準

実行計画の一部が「Index Scan estimated rows=50, actual rows=500000, loops=1」となっている。最も重要な読み取りはどれか。

ア：estimated rowsとactual rowsは常に一致しなければ実行不能である。

イ：索引走査自体よりも、まず推定行数が実測と大きく乖離している点を性能原因候補として調査する。

ウ：actual rowsが多いほど索引走査は必ず最適である。

エ：loops=1なので性能問題は存在しない。

答え・解説 正答：イ

ア：推定には誤差があり、実行自体は可能である。

イ：大幅なカーディナリティ推定誤差は不適切な計画選択につながり得る。

ウ：大量行取得では別経路が有利な場合もある。

エ：1回の走査でも50万行処理していれば重い可能性がある。

総合解説：大幅なカーディナリティ推定誤差は不適切な計画選択につながり得る。

対象：2026年度 / 基準日 2026-09-02

0125 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：標準

バックアップ一覧には「日曜00:00フル、月～土00:00差分、更新ログは常時保管」とある。水曜15:00時点へ復旧する際の基本的な考え方として最も適切なものはどれか。

ア：水曜15:00のバックアップファイルがないので絶対に復旧不能である。

イ：更新ログだけでフルバックアップは不要である。

ウ：復旧方式の仕様に従い、必要な基点バックアップと差分を復元し、更新ログを水曜15:00まで適用する。

エ：差分バックアップは取得後すぐ削除すべきである。

答え・解説 正答：ウ

ア：更新ログを用いた時点復旧が可能な設計なら15:00へ進められる。

イ：一般にログ再生の基点となるデータ状態が必要である。

ウ：時点復旧には基点状態とその後の変更ログを組み合わせるのが基本である。

エ：復旧チェーンに必要なバックアップを削除すると復旧不能になり得る。

総合解説：時点復旧には基点状態とその後の変更ログを組み合わせるのが基本である。

対象：2026年度 / 基準日 2026-09-02

0126 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：標準

業務仕様は「注文には1件以上の注文明細が必須」である。一方、ER図では注文側から注文明細側の多重度が0..*と記載されている。最も適切な指摘はどれか。

ア：0..*は必ず1件以上を意味するので矛盾しない。

イ：ER図では多重度を表せない。

ウ：注文と明細の関係はデータモデルでは扱わない。

エ：業務仕様とER図が矛盾しており、注文に対する明細の最小多重度を1として表現できるか見直す。

答え・解説 正答：エ

ア：0..*は0件も許容する。

イ：ER表現では関係の多重度を表すことができる。

ウ：中心的なモデリング対象である。

エ：0件を許す図は「1件以上必須」という業務ルールと一致しない。

総合解説：0件を許す図は「1件以上必須」という業務ルールと一致しない。

対象：2026年度 / 基準日 2026-09-02

0127 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：標準

同じ5分間に、DB CPU 40→45%、I/O待ち 5→48%、平均読取レイテンシ 2ms→38ms、ロック待ち件数は変化なしとなった。最も妥当な一次評価はどれか。

ア：CPUやロックより、ストレージI/O遅延が応答悪化と関連している可能性を優先して調べる。

イ：CPUが45%なのでCPU飽和が確定した。

ウ：ロック待ちが変化していないためDBに問題はない。

エ：メトリクス間の時刻を合わせる必要はない。

答え・解説 正答：ア

ア：I/O待ちと読取レイテンシだけが大きく悪化しており、観測事実に沿った仮説である。

イ：45%だけではCPU飽和とは言えず、I/O指標の変化が大きい。

ウ：I/O起因のDB性能問題はあり得る。

エ：相関分析には同じ時間軸で比較することが重要である。

総合解説：I/O待ちと読取レイテンシだけが大きく悪化しており、観測事実に沿った仮説である。

対象：2026年度 / 基準日 2026-09-02

0128 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：標準

移行仕様書で旧列「顧客区分」はA/B/C、新列は1/2/3/9であるが、A→1、B→2しか定義されていない。最も適切な指摘はどれか。

ア：AとBが定義されているのでCは自動的に3になる。

イ：Cの変換先と、想定外値・NULLの扱いが未定義なので、移行前にルールを確定する必要がある。

ウ：新列に9があるので全件9へ変換する。

エ：移行時はデータ型・値域の確認は不要である。

答え・解説 正答：イ

ア：仕様がない対応を推測してはならない。

イ：変換マッピングの欠落は移行エラーや不正値の原因になる。

ウ：業務意味を無視した変換である。

エ：データ品質と変換ルール確認は重要である。

総合解説：変換マッピングの欠落は移行エラーや不正値の原因になる。

対象：2026年度 / 基準日 2026-09-02

0129 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：応用

障害ログでは、09:59:58 ストレージレイテンシ急増、10:00:01 WAL書込み待ち増加、10:00:04 コミット遅延、10:00:12 アプリのタイムアウト増加となっている。現時点の最も妥当な説明はどれか。

ア：アプリタイムアウトが最初の原因だと確定する。

イ：WAL待ちはストレージI/Oと無関係である。

ウ：ストレージ遅延が先行し、その後WAL待ち・コミット遅延・アプリタイムアウトが続いているため、ストレージ側を有力原因候補として検証する。

エ：このログからストレージ故障を100%断定できる。

答え・解説 正答：ウ

ア：タイムアウトは最後に観測されており、少なくとも時系列とは合わない。

イ：WAL書込みはストレージ性能の影響を受け得る。

ウ：時系列は因果を確定しないが、上流候補を絞る根拠として妥当である。

エ：時系列は有力仮説を示すが、故障確定には追加証拠が必要である。

総合解説：時系列は因果を確定しないが、上流候補を絞る根拠として妥当である。

対象：2026年度 / 基準日 2026-09-02

0130 6-5 長文総合判断 > 表・図・ログ・仕様書の読解 | 難易度：応用

受入基準は「ピーク1,000TPS以上、p95 $\leq$ 1.5秒、エラー率 $<0.1\%$ 」。試験結果は1,080TPS、p95=1.7秒、エラー率0.03%だった。最も適切な判定はどれか。

ア：TPSが目標超過なので他条件に関係なく合格である。

イ：エラー率が低いので応答時間条件は無視してよい。

ウ：三つの数値を平均して基準値と比較する。

エ：TPSとエラー率は満たすがp95を満たさないため、総合的には受入基準未達である。

答え・解説 正答：エ

ア：p95条件が未達である。

イ：仕様に明記された応答時間条件も満たす必要がある。

ウ：単位・意味の異なる指標を平均しても受入判定にならない。

エ：複数の必須条件のうち一つでも未達なら合格とはできない。

総合解説：複数の必須条件のうち一つでも未達なら合格とはできない。

対象：2026年度 / 基準日 2026-09-02

0131 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：基礎

案Aは索引2本で照会平均200ms、更新50ms。案Bは索引10本で照会80ms、更新180ms。業務は更新が全処理の80%で、更新SLOは100ms以内である。最も妥当な選択はどれか。

ア：案Aを基本とし、必要な照会だけ追加最適化を検討する。

イ：照会が速いので無条件に案Bにする。

ウ：索引数が多い案を必ず選ぶ。

エ：どちらも性能要件を確認せずランダムに選ぶ。

答え・解説 正答：ア

ア：案Bは主要処理である更新のSLOを違反しており、業務比率も更新中心である。

イ：更新SLO違反と業務比率を無視している。

ウ：索引は更新コストとのトレードオフがある。

エ：要件と測定値を基に判断すべきである。

総合解説：案Bは主要処理である更新のSLOを違反しており、業務比率も更新中心である。

対象：2026年度 / 基準日 2026-09-02

0132 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：基礎

案Aは遠隔同期複製でRPOほぼ0だが書込みが40ms増える。案Bは非同期で書込み影響5msだが最大30秒の遅延がある。要件はRPO 10秒以内である。最も適切な判断はどれか。

ア：5msしか増えないので案Bを無条件採用する。

イ：案BはRPO要件を満たさない可能性があるため、そのままでは採用不可で、案Aまたは遅延を10秒以内に抑える別案を検討する。

ウ：同期方式は必ずRPOが30秒以上になる。

エ：RPOと書込み遅延は比較時に考慮しない。

答え・解説 正答：イ

ア：RPO 10秒要件を満たさない可能性を無視している。

イ：性能だけでなくRPOという必須条件でふるいに掛ける必要がある。

ウ：一般に同期方式はデータ損失をより小さくする目的で使われる。

エ：可用性方式の主要なトレードオフである。

総合解説：性能だけでなくRPOという必須条件でふるいに掛ける必要がある。

対象：2026年度 / 基準日 2026-09-02

0133 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：基礎

新DB基盤について、案Aはオンプレミス、案BはマネージドDBである。最も適切な比較方法はどれか。

ア：月額料金が安い方だけで決める。

イ：新しい技術の方を必ず採用する。

ウ：性能、可用性、運用負荷、拡張性、セキュリティ、移行性、総コストなどを要件に重み付けして比較する。

エ：比較表を作らず担当者の好みで決める。

答え・解説 正答：ウ

ア：運用・可用性・性能等の重要条件を無視する。

イ：新旧ではなく要件適合性が基準である。

ウ：単一指標ではなく業務・非機能要件との適合度で総合評価すべきである。

エ：説明可能な評価基準がなく、設計判断として弱い。

総合解説：単一指標ではなく業務・非機能要件との適合度で総合評価すべきである。

対象：2026年度 / 基準日 2026-09-02

0134 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：標準

売上集計で5表結合がボトルネックになっている。案Aは正規化構造を維持して索引・SQLを改善、案Bは集計用の冗長列を持たせる。最も適切な進め方はどれか。

ア：性能問題があれば必ず正規化を全面廃止する。

イ：冗長列は更新整合性を考えなくてよい。

ウ：索引やSQLの実行計画は比較に不要である。

エ：まず案Aで改善余地を測定し、なお要件未達なら更新整合性維持方法を明確にしたうえで案Bを検討する。

答え・解説 正答：エ

ア：整合性や保守性の悪化を招き、過剰対応である。

イ：複数箇所に同じ事実を持つため同期方法が必要である。

ウ：既存構造での改善余地を評価する重要材料である。

エ：非正規化は読み取りを速める可能性があるが整合性・更新コストを増やすため段階的に判断する。

総合解説：非正規化は読み取りを速める可能性があるが整合性・更新コストを増やすため段階的に判断する。

対象：2026年度 / 基準日 2026-09-02

0135 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：標準

履歴表は日付条件で古いデータを一括削除し、照会も期間指定が中心である。案Aは月単位の範囲パーティション、案Bは列を二群に分ける垂直分割。より直接的に要件へ合うのはどれか。

- ア：案Aの月単位範囲パーティション
- イ：案Bの垂直分割
- ウ：どちらも同じ効果なので比較不要
- エ：主キーを削除する案

答え・解説 正答：ア

ア：日付での絞込みと古い期間単位の保守に直接適合する。
イ：列アクセスの偏りには有効だが、期間単位の削除・検索という要件への直接性は低い。
ウ：分割軸が異なり、効果も異なる。
エ：期間管理の改善にはならず整合性を損なう。
総合解説：日付での絞込みと古い期間単位の保守に直接適合する。

対象：2026年度 / 基準日 2026-09-02

0136 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：標準

商品情報参照の負荷を下げるためキャッシュ導入を検討している。一方、価格変更は即時反映が必要である。最も適切な設計はどれか。

- ア：TTLを1日固定にして価格変更の即時反映要件を無視する。
- イ：キャッシュ無効化・更新戦略と許容鮮度を明確化し、価格変更時に古い値を返さない仕組みを設計する。
- ウ：キャッシュを使えばDB整合性は自動的に保証される。
- エ：価格情報をDBから削除する。

答え・解説 正答：イ

ア：最大1日古い価格を返す可能性があり要件違反である。
イ：キャッシュ性能とデータ鮮度のトレードオフを要件に合わせて制御する必要がある。
ウ：キャッシュとDBの同期・無効化は別途設計が必要である。
エ：正本データを失う。
総合解説：キャッシュ性能とデータ鮮度のトレードオフを要件に合わせて制御する必要がある。

対象：2026年度 / 基準日 2026-09-02

0137 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：標準

24時間稼働システムで停止可能時間は30分、データ量は20TB、日次変更量は100GBである。案Aは停止中に20TB一括コピー、案Bは事前に全量同期し、切替時は差分のみ適用する。最も妥当な案はどれか。

- ア：案A
- イ：どちらもデータ量を測らず選ぶ
- ウ：案B
- エ：本番停止後に初めて移行手順を考える

答え・解説 正答：ウ

ア：20TBを停止30分内で移す成立性が示されておらず高リスクである。
イ：容量と実効速度は移行方式決定の基本情報である。
ウ：停止時間中の処理量を差分へ縮小できるため、30分制約へ適合する可能性が高い。
エ：リハーサルと手順確立が必要である。
総合解説：停止時間中の処理量を差分へ縮小できるため、30分制約へ適合する可能性が高い。

対象：2026年度 / 基準日 2026-09-02

0138 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：標準

案Aは同一拠点の冗長化でRTO 5分・低コストだが拠点災害に弱い。案Bは遠隔待機系でRTO 30分・高コストだが拠点災害に対応する。業務要件は「拠点災害でも1時間以内に再開」である。最も適切な判断はどれか。

- ア：RTOが5分なので案Aを無条件採用する。
- イ：高コストな案は要件に関係なく必ず却下する。
- ウ：RTOは災害対策案比較に使わない。
- エ：案Bを中心に検討し、費用対効果を含めて遠隔復旧要件を満たす構成を選ぶ。

答え・解説 正答：エ

ア：拠点災害時の再開要件を満たせない。
イ：必須の事業継続要件とのバランスで判断すべきである。
ウ：復旧方式選定の中心的な指標である。
エ：必須要件が拠点災害対応なので、同一拠点だけの案Aでは要件を満たせない。
総合解説：必須要件が拠点災害対応なので、同一拠点だけの案Aでは要件を満たせない。

対象：2026年度 / 基準日 2026-09-02

0139 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：応用

分析照会を本番更新系から分離する案として非同期読み取りレプリカを使う。一般レポートは数分遅延可だが、「注文直後の確認画面」は最新状態必須である。最も適切な構成はどれか。

ア：一般レポートはレプリカへ逃がし、最新性必須の確認画面はプライマリ等の強い整合性を確保できる経路を使う。

イ：すべての読み取りを非同期レプリカへ固定する。

ウ：すべてをプライマリへ戻し、分析負荷分散の目的を捨てるしかない。

エ：非同期レプリカは常にプライマリと完全同期である。

答え・解説 正答：ア

ア：読み取り用途ごとの鮮度要件に応じて経路を分けるのが合理的である。

イ：注文直後に未反映の可能性があるため最新性要件に反する。

ウ：用途別ルーティングで両立できる可能性がある。

エ：遅延が生じる可能性があるため誤り。

総合解説：読み取り用途ごとの鮮度要件に応じて経路を分けるのが合理的である。

対象：2026年度 / 基準日 2026-09-02

0140 6-5 長文総合判断 > 代替案比較・トレードオフ判断 | 難易度：応用

コスト削減案として、DB、バックアップ保存先、監視サーバを同一物理ホストに集約する提案がある。通常時コストは下がる。最も重要な評価はどれか。

ア：同じホストならデータは絶対に失われない。

イ：同一ホスト障害で本体・バックアップ・監視を同時に失う共通障害点が生じるため、可用性・復旧要件とのトレードオフを再評価する。

ウ：監視サーバは障害時に不要なので同居リスクはない。

エ：バックアップは本体と同一障害ドメインに置くほど安全である。

答え・解説 正答：イ

ア：共通障害で同時喪失する可能性がある。

イ：コスト削減が障害ドメインの集中を招き、復旧可能性を損なう可能性がある。

ウ：障害検知・診断に必要であり同時喪失は問題となる。

エ：独立した障害から守るため分離が基本である。

総合解説：コスト削減が障害ドメインの集中を招き、復旧可能性を損なう可能性がある。

対象：2026年度 / 基準日 2026-09-02

0141 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：基礎

設問は「なぜ索引が利用されなかったか、実行計画の情報を根拠に40字程度で答えよ」。実行計画では対象行が表の90%と推定され全表走査が選ばれている。最も適切な答案はどれか。

ア：索引は便利なので使うべきだった。

イ：DBは難しいので遅かったため。

ウ：対象行が表の大部分と推定され、索引経由より全表走査が低コストと判断されたため。

エ：表名が長かったため。

答え・解説 正答：ウ

ア：設問の「なぜ」に答えず、計画上の根拠もない。

イ：技術的根拠がなく曖昧である。

ウ：実行計画の根拠と理由を直接結び付けている。

エ：提示情報から導けず、性能理由として不適切である。

総合解説：実行計画の根拠と理由を直接結び付けている。

対象：2026年度 / 基準日 2026-09-02

0142 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：基礎

設問は「日次バックアップだけでは不十分な理由をRPO 10分の観点で述べよ」。最も適切な答案はどれか。

ア：バックアップは容量を使うため。

イ：毎日実行すると管理者が忙しいため。

ウ：RPOは応答時間の指標だから。

エ：日次取得では最長約24時間分を失い得て、許容損失10分を超えるため。

答え・解説 正答：エ

ア：RPOとの関係を説明していない。

イ：技術要件への回答になっていない。

ウ：RPOの定義が誤っている。

エ：RPOの意味と日次間隔を直接比較している。

総合解説：RPOの意味と日次間隔を直接比較している。

対象：2026年度 / 基準日 2026-09-02

0143 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：基礎

設問は「CPUではなくI/Oを優先調査する理由を述べよ」。CPU45%、I/O待ち50%、読取遅延40msという表がある。最も適切な答えはどれか。

ア：CPUに余力がある一方、I/O待ちと読取遅延が大きく増えているため。

イ：I/Oという言葉が難しそうだから。

ウ：CPUは常に性能問題の原因にならないため。

エ：ディスクは必ず故障しているため。

答え・解説 正答：ア

ア：表の具体的な観測値を根拠として優先順位を説明している。

イ：技術的根拠になっていない。

ウ：CPUが原因となる場合もあり断定が誤り。

エ：遅延増大だけで故障を断定できない。

総合解説：表の具体的な観測値を根拠として優先順位を説明している。

対象：2026年度 / 基準日 2026-09-02

0144 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：標準

ログではストレージ遅延→WAL待ち→コミット遅延の順に発生した。設問は「原因候補を述べよ」であり、追加調査前である。最も適切な答えはどれか。

ア：ストレージ故障が100%確定した。

イ：ストレージ遅延が先行しているため、WAL書き込み遅延の原因候補としてストレージを優先調査する。

ウ：コミット遅延が最初の原因である。

エ：ログからは何も読み取れない。

答え・解説 正答：イ

ア：追加調査前に故障を断定し過ぎている。

イ：時系列を根拠にしつつ「候補」として表現し、過剰断定を避けている。

ウ：提示された時系列に反する。

エ：少なくとも発生順序と有力候補は読み取れる。

総合解説：時系列を根拠にしつつ「候補」として表現し、過剰断定を避けている。

対象：2026年度 / 基準日 2026-09-02

0145 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：標準

受入条件は「1,000TPS以上かつp95 1.5秒以内」。結果は1,100TPS、p95 1.8秒。設問は「受入可否と理由」を答えるものとする。最も適切な答案はどれか。

ア：受入可。TPSが高いため。

イ：受入不可。なんとなく遅いため。

ウ：受入不可。TPSは満たすがp95が1.5秒を超え、必須条件の一つを満たさないため。

エ：受入可否は平均値がないので判断不能。

答え・解説 正答：ウ

ア：p95違反を無視している。

イ：数値根拠がなく設問への説明として弱い。

ウ：可否と、その根拠となる満足・未満足条件を簡潔に示している。

エ：受入基準に平均値は指定されていない。

総合解説：可否と、その根拠となる満足・未満足条件を簡潔に示している。

対象：2026年度 / 基準日 2026-09-02

0146 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：標準

移行前後で総件数は一致するが、顧客別売上合計の一部が異なる。設問は「次に確認すべき点」を述べるものとする。最も適切な答案はどれか。

ア：件数が一致しているので差異を無視する。

イ：全顧客の売上を0へ更新する。

ウ：索引名を変更する。

エ：変換・集計対象列のマッピングとクレンジング規則を確認し、差異行をキー単位で追跡する。

答え・解説 正答：エ

ア：値の誤りを見逃す。

イ：原因調査にならずデータを破壊する。

ウ：売上値の不一致を直接解決しない。

エ：件数一致では検出できない値変換誤りに焦点を当て、具体的な追跡方法を示す。

総合解説：件数一致では検出できない値変換誤りに焦点を当て、具体的な追跡方法を示す。

対象：2026年度 / 基準日 2026-09-02

0147 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：標準

設問は「遠隔同期レプリケーションの長所と短所を述べよ」。最も適切な答えはどれか。

ア：データ損失を小さくしやすい一方、遠隔確認待ちにより書き込み遅延が増える可能性がある。

イ：必ず速く、欠点はない。

ウ：遅いのでデータ保護には何の効果もない。

エ：バックアップと完全に同じ機能である。

答え・解説 正答：ア

ア：可用性上の利点と性能上の代償を一文で対比している。

イ：同期方式にもレイテンシ等のトレードオフがある。

ウ：データ保護上の利点を無視している。

エ：複製とバックアップは障害・誤操作への役割が異なる。

総合解説：可用性上の利点と性能上の代償を一文で対比している。

対象：2026年度 / 基準日 2026-09-02

0148 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：標準

業務要件が「1人の担当者が複数部署を兼務可能」に変更された。旧モデルは担当者表に部署IDを1列保持している。設問は「モデル修正の方向性と理由」を答えるものとする。最も適切な答えはどれか。

ア：部署ID列に「A,B,C」のような文字列を格納する。

イ：担当者と部署の多対多関係を表す中間関係を設け、1担当者に複数部署を対応付けられるようにする。

ウ：担当者表を削除する。

エ：部署を一つしか選べないままにする。

答え・解説 正答：イ

ア：多値属性を1列に詰め込み、検索・整合性管理を難しくする。

イ：要件変更を多重度の変更として捉え、関係モデルへ自然に反映している。

ウ：要件を表現できなくなる。

エ：変更後要件を満たさない。

総合解説：要件変更を多重度の変更として捉え、関係モデルへ自然に反映している。

対象：2026年度 / 基準日 2026-09-02

0149 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：応用

障害時の観測は、CPU 30%、I/O待ち55%、ストレージ遅延45ms、ロック待ち平常、実行計画は通常時と同じである。設問は「最優先調査箇所と根拠」を50字程度で答えるものとする。最も適切な答えはどれか。

ア：CPUを交換する。CPUが30%だから。

イ：SQLを書き換える。理由はない。

ウ：ストレージI/Oを優先調査する。CPU・ロック・実行計画に大きな変化がなく、I/O待ちと遅延だけが急増しているため。

エ：すべて同時に変更する。

答え・解説 正答：ウ

ア：30%は飽和を示さず、他の観測とも合わない。

イ：実行計画が通常時と同じという情報を無視し、根拠もない。

ウ：複数の反証材料と正の証拠を併せて、優先順位を説明している。

エ：原因切り分けができず、変更リスクが高い。

総合解説：複数の反証材料と正の証拠を併せて、優先順位を説明している。

対象：2026年度 / 基準日 2026-09-02

0150 6-5 長文総合判断 > 根拠抽出・記述答案判断 | 難易度：応用

要件は「拠点災害対応、RPO 1分、RTO 15分、通常時書込み遅延の増加は20ms以内」。候補Aは遠隔同期でRPOほぼ0・遅延+60ms、候補Bは非同期でRPO最大30秒・遅延+5ms・自動切替10分である。設問は「採用候補と根拠」を答えるものとする。最も適切な答えはどれか。

ア：候補A。同期という名前の方が強そうだから。

イ：候補B。安そうだから。

ウ：どちらもRPOを満たさない。

エ：候補B。RPO30秒、RTO10分、遅延+5msで三つの数値要件を満たすため。

答え・解説 正答：エ

ア：遅延+60msが20ms以内という要件に違反する。

イ：コスト情報は提示されておらず根拠にならない。

ウ：候補Bの最大30秒はRPO 1分以内である。

エ：提示された要件を一つずつ照合し、候補Bがすべて満たすことを具体的に示している。

総合解説：提示された要件を一つずつ照合し、候補Bがすべて満たすことを具体的に示している。

対象：2026年度 / 基準日 2026-09-02