

0001 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：基礎

複数部門をまたぐデータベース要件定義で、同じ「顧客」という用語が部門ごとに異なる意味で使われていることが分かった。最初に行う対応として最も適切なものはどれか。

ア：各部門の定義・識別条件・利用目的を整理し、用語とデータの意味を合意する。

イ：物理テーブル名だけ先に決め、意味の違いは実装後に調整する。

ウ：すべての顧客データを一つの文字列列に格納する。

エ：一つの部門の定義を確認せず全社標準として採用する。

答え・解説 正答：ア

ア：同じ用語でも業務意味が異なればモデルやキー設計が食い違うため、意味・範囲・識別条件を明確にする必要がある。

イ：論理的な意味の不一致を物理名では解消できない。

ウ：構造化や識別ルールの問題を解決せず、整合性を悪化させる。

エ：他部門の業務要件を取りこぼす可能性が高い。

総合解説：同じ用語でも業務意味が異なればモデルやキー設計が食い違うため、意味・範囲・識別条件を明確にする必要がある。

対象：2026年度 / 基準日 2026-09-02

0002 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：基礎

現行業務で「どの業務が、どのデータを作成・参照・更新・削除しているか」を整理する代表的な観点は何ですか。

ア：全データを一律に削除対象とする。

イ：CRUDの対応関係を整理する。

ウ：CPU命令セットだけを比較する。

エ：SQLの予約語一覧だけを作る。

答え・解説 正答：イ

ア：利用実態の分析ではなく、要件を破壊する。

イ：Create/Read/Update/Deleteの観点で業務プロセスとデータの利用関係を可視化すると、要件漏れや責任範囲を把握しやすい。

ウ：業務とデータ利用の把握には直接つながらない。

エ：構文知識だけでは現状業務のデータ利用実態を把握できない。

総合解説：Create/Read/Update/Deleteの観点で業務プロセスとデータの利用関係を可視化すると、要件漏れや責任範囲を把握しやすい。

対象：2026年度 / 基準日 2026-09-02

0003 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：基礎

要件定義書に記載する内容として、データ要件に最も近いものはどれか。

ア：索引は必ずB-treeで3本作成する。

イ：ストレージ装置は特定メーカーの型番Xだけを使う。

ウ：受注データは受注番号で一意に識別でき、受注日・顧客・合計金額を保持する必要がある。

エ：全テーブルを一つの表領域に置く。

答え・解説 正答：ウ

ア：具体的な物理設計・実装方式に踏み込んでいる。

イ：製品指定であり、データ要件ではない。

ウ：何を保持し、どのような整合性が必要かはデータ要件であり、具体的製品や物理配置より上位の要求である。

エ：物理配置の決定であり、要件定義より後段の設計事項である。

総合解説：何を保持し、どのような整合性が必要かはデータ要件であり、具体的製品や物理配置より上位の要求である。

対象：2026年度 / 基準日 2026-09-02

0004 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：標準

旧システムからの移行を前提に顧客データの現状を調査する。欠損・重複・値域外・コード体系のばらつきを把握する方法として最も適切なものはどれか。

ア：テーブル名だけを一覧化し、データ値は調べない。

イ：移行後にエラーが出たデータだけ個別に修正する前提とする。

ウ：代表1件だけを目視し、全件も同じ品質だとみなす。

エ：列ごとのNULL率、重複率、値分布、最小最大、コード別件数などをプロファイリングする。

答え・解説 正答：エ

ア：構造は把握できても品質問題を把握できない。

イ：事前の品質把握とクレンジング設計を省略しており危険である。

ウ：分布や例外を把握できず、移行リスクを過小評価する。

エ：データプロファイリングにより品質問題を定量化し、クレンジングや変換要件の根拠を得られる。

総合解説：データプロファイリングにより品質問題を定量化し、クレンジングや変換要件の根拠を得られる。

対象：2026年度 / 基準日 2026-09-02

0005 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：標準

同じ売上金額が複数システムに存在し、値が一致しない。データ要件定義で優先して確認すべきことはどれか。

- ア：各値の発生源、変換処理、連携経路、更新責任を追跡し、正本となるデータを特定する。
- イ：一番古いシステムを無条件に正本とする。
- ウ：値が大きい方を常に正しい値とする。
- エ：すべて平均して新しい値を作る。

答え・解説 正答：ア

- ア：データの由来と流通を明らかにしなければ、不一致の原因や正しい更新元を判断できない。
 - イ：システムの古さだけでは正当性を判断できない。
 - ウ：業務根拠がなく、誤った正本を選ぶ可能性がある。
 - エ：業務上の意味を失い、整合性を保証できない。
- 総合解説：データの由来と流通を明らかにしなければ、不一致の原因や正しい更新元を判断できない。

対象：2026年度 / 基準日 2026-09-02

0006 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：標準

受注確定時に「誰が、いつ、どの商品を、何個、どの価格で確定したか」を後から監査できる必要がある。この要件から導くデータ設計上の対応として最も適切なものはどれか。

- ア：現在の商品マスタ価格だけ保存し、受注時価格は保持しない。
- イ：受注確定という業務イベントと、その時点で必要な属性・識別子・時刻を保持する要件を定義する。
- ウ：確定時刻をアプリケーション画面にだけ表示し、保存しない。
- エ：受注明細を毎日上書きして最新状態だけにする。

答え・解説 正答：イ

- ア：後日価格が変更された場合、受注時点の事実を再現できない。
 - イ：監査要件は業務イベント時点の事実を再現できるデータ要件へ落とし込む必要がある。
 - ウ：監査で後から確認できない。
 - エ：履歴が失われ、確定時点の事実を追跡できない。
- 総合解説：監査要件は業務イベント時点の事実を再現できるデータ要件へ落とし込む必要がある。

対象：2026年度 / 基準日 2026-09-02

0007 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：標準

営業部は顧客住所を常に最新1件だけ保持したいが、法務部は契約締結時の住所を10年間確認できる必要がある。要件定義として最も適切な対応はどれか。

ア：法務部の要求だけ採用し、常に全履歴を画面へ表示する。

イ：住所項目を削除して対立をなくす。

ウ：両部門の利用目的と保存期間を整理し、最新住所と契約時点情報を両立できるデータ要件を合意する。

エ：営業部の要求だけ採用し、法務部には説明しない。

答え・解説 正答：ウ

ア：営業部の利用性要件を無視している。

イ：必要な業務情報を失うだけで解決にならない。

ウ：対立する要求は背景と利用目的を明確化し、履歴保持などの設計要件として両立可能性を検討する。

エ：監査・法務要件を満たせない。

総合解説：対立する要求は背景と利用目的を明確化し、履歴保持などの設計要件として両立可能性を検討する。

対象：2026年度 / 基準日 2026-09-02

0008 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：標準

データベース要件定義書のレビュー参加者として最も適切な組合せはどれか。

ア：経理部門だけで全業務の要件を決める。

イ：データベース担当者1名だけ。

ウ：製品ベンダーだけで業務利用者は参加させない。

エ：業務利用者、アプリケーション開発者、データベース担当者など、要求と実装影響を確認できる関係者。

答え・解説 正答：エ

ア：関係部門の要件を網羅できない。

イ：業務要求やアプリケーション側の影響を十分に確認できない。

ウ：業務要求の妥当性や優先順位を確認できない。

エ：IPAシラバスでも利用者・アプリケーション開発者を交えたレビューが求められており、多面的な妥当性確認が重要である。

総合解説：IPAシラバスでも利用者・アプリケーション開発者を交えたレビューが求められており、多面的な妥当性確認が重要である。

対象：2026年度 / 基準日 2026-09-02

0009 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：応用

利用者から「顧客データは絶対に正しくしてほしい」という要求が出た。要件定義として最も適切な進め方はどれか。

ア：正しさの意味を、必須項目、重複許容、参照整合性、更新責任、許容誤差など検証可能な条件へ分解する。

イ：「絶対に正しい」とだけ要件書に記載して終了する。

ウ：すべての列を文字列型にして入力自由度を上げる。

エ：データ量をゼロにすれば誤りがなくなると判断する。

答え・解説 正答：ア

ア：抽象的要求は測定・検証可能な条件へ翻訳して初めて設計・テストにつなげられる。

イ：検証基準がなく、設計や受入れ判定に使えない。

ウ：整合性確保とは逆方向になり得る。

エ：業務目的を満たさない。

総合解説：抽象的要求は測定・検証可能な条件へ翻訳して初めて設計・テストにつなげられる。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0010 4-1 要件定義・方式設計 > 現状調査・データ要件 | 難易度：応用

新システムでは顧客を法人単位だけでなく事業所単位でも識別する必要があるが、現行データには事業所識別子がない。最も適切な要件定義上の対応はどれか。

ア：法人IDを事業所IDとしてそのまま複製する。

イ：必要な識別粒度を明確化し、事業所識別子の採番・取得元・既存データへの補完方法を要件として整理する。

ウ：事業所を備考欄の自由記述だけで管理する。

エ：現行にない項目は新システムでも不要とみなす。

答え・解説 正答：イ

ア：一法人に複数事業所がある要件を表現できない。

イ：To-Beで必要なデータがAs-Isに存在しない場合、取得・移行・責任分担まで含めたギャップ対応が必要である。

ウ：一意識別や参照整合性を確保しにくい。

エ：将来業務要件を満たせない。

総合解説：To-Beで必要なデータがAs-Isに存在しない場合、取得・移行・責任分担まで含めたギャップ対応が必要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0011 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：基礎

オンライン検索の性能要件として最も検証しやすい記述はどれか。

ア：平均応答時間は良い感じにする。

イ：検索はできるだけ速くする。

ウ：通常負荷時、検索要求の95パーセンタイル応答時間を2秒以内とする。

エ：DBサーバを高性能にする。

答え・解説 正答：ウ

ア：測定値も閾値も定義されていない。

イ：基準が曖昧で合否判定できない。

ウ：測定条件・指標・閾値が具体的であり、テストで合否を判定できる。

エ：手段の記述であり、性能要件そのものではない。

総合解説：測定条件・指標・閾値が具体的であり、テストで合否を判定できる。

対象：2026年度 / 基準日 2026-09-02

0012 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：基礎

RPO（Recovery Point Objective）の説明として最も適切なものはどれか。

ア：障害発生からサービス再開までの許容時間。

イ：平常時の平均SQL応答時間。

ウ：バックアップファイルの圧縮率。

エ：障害発生時に、どの時点までのデータ損失を許容できるかを表す目標。

答え・解説 正答：エ

ア：これはRTOの説明である。

イ：性能指標でありRPOではない。

ウ：復旧時点目標とは別概念である。

エ：RPOは許容可能なデータ損失の時間幅・復旧時点に関する目標である。

総合解説：RPOは許容可能なデータ損失の時間幅・復旧時点に関する目標である。

対象：2026年度 / 基準日 2026-09-02

0013 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：基礎

データベースのアクセス制御要件として最小権限の原則に最も合うものはどれか。

ア：利用者・アプリケーションごとに業務上必要な操作だけを許可する。

イ：認証済みなら全テーブルの更新を許可する。

ウ：権限管理をアプリケーション画面だけに任せ、DB側では制御しない。

エ：全利用者にDB管理者権限を付与する。

答え・解説 正答：ア

ア：必要最小限の権限に限定することで、誤操作や侵害時の影響範囲を縮小できる。

イ：認証と認可を区別しておらず、最小権限に反する。

ウ：多層防御や直接接続時の保護が弱くなる。

エ：過大権限となりリスクが高い。

総合解説：必要最小限の権限に限定することで、誤操作や侵害時の影響範囲を縮小できる。

対象：2026年度 / 基準日 2026-09-02

0014 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：標準

Web APIのDB応答時間について、平均値だけでなく95/99パーセンタイルも要件に含める主な理由はどれか。

ア：パーセンタイルを使えば負荷試験が不要になるから。

イ：一部の遅い要求が平均値に埋もれるのを防ぎ、利用者が経験する遅延の裾を評価できるから。

ウ：パーセンタイルを使うと必ずCPU使用率が下がるから。

エ：平均値はSQLでは計算できないから。

答え・解説 正答：イ

ア：性能要件の検証には負荷試験が必要である。

イ：平均値だけでは少数の極端に遅い処理を見落とす可能性がある。

ウ：指標の選択自体が資源使用率を下げるわけではない。

エ：平均値は計算可能である。

総合解説：平均値だけでは少数の極端に遅い処理を見落とす可能性がある。

対象：2026年度 / 基準日 2026-09-02

0015 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：標準

「1秒間に500件処理できること」という要件をDB性能評価に使う際、追加で明確にすべき情報として最も重要なものはどれか。

ア：CPUクロック周波数だけを固定し、同時接続数やデータ量は定めない。

イ：処理件数だけを示し、1件当たりの処理内容や読書き比率は定めない。

ウ：処理内容、同時接続数、データ量、読書き比率など負荷条件。

エ：データ量だけを固定し、トランザクション構成や同時実行条件は定めない。

答え・解説 正答：ウ

ア：性能はCPUだけで決まらず、負荷モデルが不明では再現可能な評価ができない。

イ：同じ500件/秒でも処理内容が異なれば必要資源が大きく変わるため条件不足である。

ウ：性能要件は処理内容・同時性・データ量・読書き構成などの負荷条件と組にして評価可能にする必要がある。

エ：データ量は重要だが、それだけでは処理負荷を特定できない。

総合解説：同じTPSでも処理の重さや並行度によって必要資源が大きく異なるため、負荷条件を定義する必要がある。

対象：2026年度 / 基準日 2026-09-02

0016 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：標準

年間稼働率99.99%を目標とする。1年を365日として、計画外停止を含む許容停止時間の概算として最も近いものはどれか。

ア：約8.8時間（99.9%程度の停止時間と取り違えた値）

イ：約3.65日（1%停止と取り違えた値）

ウ：約5.3分（99.999%程度の停止時間と取り違えた値）

エ：約53分（365日×24時間×0.01%）

答え・解説 正答：エ

ア：99.99%の停止率は0.01%であり、約8.8時間では長すぎる。

イ：3.65日は年間の約1%に相当し、99.99%要件とは合わない。

ウ：約5.3分は停止率0.001%程度で、1桁厳しい値である。

エ：年間525600分×0.0001=約52.56分なので約53分である。

総合解説：1年は525,600分で、0.01%は約52.56分である。

対象：2026年度 / 基準日 2026-09-02

0017 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：標準

業務要件が「RTO 30分、RPO 5分」である。夜間に1日1回だけフルバックアップを取得し、ログを保存しない方式の評価として最も適切なものはどれか。

ア：最大24時間近いデータ損失が起こり得るため、RPO 5分を満たせない可能性が高い。

イ：バックアップを圧縮すれば必ずRPO 5分になる。

ウ：RTOが30分なのでRPOも自動的に30分になる。

エ：フルバックアップなら取得頻度に関係なくRPOは0になる。

答え・解説 正答：ア

ア：RPOを満たすには、バックアップ間隔だけでなくログ・レプリケーション等で復旧点を細かくする必要がある。

イ：圧縮率は復旧点の間隔を決めない。

ウ：RTOとRPOは別の目標である。

エ：取得後の更新は失われ得る。

総合解説：RPOを満たすには、バックアップ間隔だけでなくログ・レプリケーション等で復旧点を細かくする必要がある。

対象：2026年度 / 基準日 2026-09-02

0018 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：標準

機密データについて「保存媒体の盗難」と「ネットワーク盗聴」の両方を想定する。要件として最も適切なものはどれか。

ア：保存時暗号化だけで通信盗聴も必ず防げる。

イ：保存時暗号化と通信経路の暗号化を別々に要件化し、鍵管理も含めて設計する。

ウ：通信暗号化だけで盗まれた未暗号化ディスクも保護できる。

エ：暗号化すればアクセス制御は不要になる。

答え・解説 正答：イ

ア：通信中の平文を別途保護する必要がある。

イ：保存時と通信時では脅威面が異なり、両方に対応する暗号化と鍵管理が必要になる。

ウ：保存媒体上のデータ保護とは別である。

エ：暗号化と認可は補完関係にあり、代替できない。

総合解説：保存時と通信時では脅威面が異なり、両方に対応する暗号化と鍵管理が必要になる。

対象：2026年度 / 基準日 2026-09-02

0019 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：応用

監査部門は全更新の詳細ログを長期保存したい一方、業務部門は更新遅延を最小化したい。最も適切な設計判断はどれか。

ア：監査を優先して性能試験を行わない。

イ：ログを画面表示だけにして保存しない。

ウ：監査上必要なログ粒度・保存期間を明確にし、非同期化や保管層分離なども含め性能とのトレードオフを検証する。

エ：性能を優先して監査ログを全廃する。

答え・解説 正答：ウ

ア：性能要件の達成可能性を確認できない。

イ：長期監査要件を満たさない。

ウ：要求をどちらか一方に寄せるのではなく、必要性和性能影響を定量評価して両立方式を設計する。

エ：監査要件を満たせない。

総合解説：要求をどちらか一方に寄せるのではなく、必要性和性能影響を定量評価して両立方式を設計する。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0020 4-1 要件定義・方式設計 > 性能・可用性・セキュリティ要件 | 難易度：応用

DBクラスタ自体は99.999%の稼働率を持つが、アプリケーションからDBへ接続する単一のネットワーク装置に冗長性がない。可用性評価として最も適切なものはどれか。

ア：DBを暗号化すればネットワーク装置の冗長化は不要になる。

イ：DBの稼働率が高いのでネットワーク障害は無視できる。

ウ：単一ネットワーク装置の停止はRPOにだけ影響し、可用性には影響しない。

エ：サービス全体の可用性は依存経路の単一障害点にも制約されるため、DBだけでなく接続経路も評価する。

答え・解説 正答：エ

ア：暗号化は可用性の冗長化を代替しない。

イ：利用者がDBへ到達できなければサービスとして利用できない。

ウ：接続不能はサービス停止を引き起こし得る。

エ：エンドツーエンドの可用性は各構成要素の依存関係で決まるため、DB単体の数値だけでは保証できない。

総合解説：エンドツーエンドの可用性は各構成要素の依存関係で決まるため、DB単体の数値だけでは保証できない。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0021 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：基礎

複数テーブル間の参照整合性と、複数更新をまとめたトランザクション整合性が重要な基幹業務に対して、まず有力候補となるデータ管理方式はどれか。

ア：RDBMS

イ：単純なファイル共有だけ。

ウ：DNSだけ。

エ：画像キャッシュだけ。

答え・解説 正答：ア

ア：関係モデル、制約、トランザクション機能を備えるRDBMSは、厳密な整合性を要求する業務の有力候補となる。

イ：参照整合性やトランザクション制御を自前実装する負担が大きい。

ウ：名前解決サービスであり業務データ管理方式ではない。

エ：永続的な基幹データ管理の主要方式ではない。

総合解説：関係モデル、制約、トランザクション機能を備えるRDBMSは、厳密な整合性を要求する業務の有力候補となる。

対象：2026年度 / 基準日 2026-09-02

0022 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：基礎

クラウドコンピューティングの一般的な特性として、需要に応じて資源を迅速に増減しやすいことを表すものはどれか。

ア：第三正規形

イ：Rapid elasticity（迅速な弾力性）

ウ：二相ロック

エ：参照整合性

答え・解説 正答：イ

ア：論理データモデルの正規化概念である。

イ：NISTのクラウド定義ではrapid elasticityが主要特性の一つである。

ウ：並行制御方式でありクラウドの基本特性ではない。

エ：DBの制約概念であり、クラウドの基本特性そのものではない。

総合解説：NISTのクラウド定義ではrapid elasticityが主要特性の一つである。

対象：2026年度 / 基準日 2026-09-02

0023 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：基礎

RDBMS製品を選定する際の評価として最も適切なものはどれか。

ア：ベンチマーク結果だけで決め、機能要件は無視する。

イ：知名度だけで決める。

ウ：機能、性能、可用性、運用性、セキュリティ、既存環境との適合性、コストを要件に照らして比較する。

エ：初期価格だけで決め、運用費は考えない。

答え・解説 正答：ウ

ア：性能以外の要件を満たさない可能性がある。

イ：要件適合性を評価できない。

ウ：IPAシラバスではコスト・機能・性能・利用性などのトレードオフを踏まえた選定能力が求められる。

エ：TCOや運用負荷を見落とす。

総合解説：IPAシラバスではコスト・機能・性能・利用性などのトレードオフを踏まえた選定能力が求められる。

対象：2026年度 / 基準日 2026-09-02

0024 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：標準

マネージドDBサービスを採用した場合の責任分界として最も適切な考え方はどれか。

ア：サービス事業者がいるので利用者側の権限設計は不要である。

イ：クラウドならバックアップ要件を定義する必要がない。

ウ：マネージドDBならアプリケーションのSQL品質は性能に影響しない。

エ：基盤保守の一部を事業者へ委ねられても、利用者側のデータ分類、権限設定、アプリ設計、バックアップ方針確認などの責任は残る。

答え・解説 正答：エ

ア：過大権限や設定不備のリスクが残る。

イ：RPO/RTOや保持期間に応じた確認が必要である。

ウ：非効率なSQLは依然として性能へ影響する。

エ：クラウド利用で管理作業の一部は移転しても、利用者のセキュリティ・データガバナンス責任が消えるわけではない。

総合解説：クラウド利用で管理作業の一部は移転しても、利用者のセキュリティ・データガバナンス責任が消えるわけではない。

対象：2026年度 / 基準日 2026-09-02

0025 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：標準

長期利用するクラウドDBを選定する。将来の他サービス移行可能性を確保する観点で、優先して確認すべき事項はどれか。

- ア：データエクスポート形式、標準SQLとの互換性、独自機能への依存、移行ツールや移出コスト。
- イ：初期利用料金の安さだけを比較し、データの取り出し方法は評価しない。
- ウ：独自機能を最大限利用することを優先し、代替サービスでの実現可能性は評価しない。
- エ：可用性SLAだけを比較し、スキーマ・SQL・データ形式の可搬性は評価しない。

答え・解説 正答：ア

- ア：可搬性にはデータ取り出し、SQL・スキーマの互換性、独自機能依存、移行手段とコストが影響する。
 - イ：価格は選定要素だが、将来の移行可能性を直接評価していない。
 - ウ：独自機能への強い依存はロックインを高める可能性があり、移行可能性確保とは逆方向になり得る。
 - エ：SLAは可用性評価には重要だが、サービス間のデータ・機能移行性を示さない。
- 総合解説：可搬性にはデータ取り出しとスキーマ・SQL・運用機能の互換性が影響する。

対象：2026年度 / 基準日 2026-09-02

0026 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：標準

国内利用者向けの低遅延と、データを国内に保存する社内規程の両方がある。クラウドDBの配置選定として最も適切なものはどれか。

- ア：リージョンは性能や規程に影響しないと考える。
- イ：利用者とのネットワーク遅延、利用可能な国内リージョン、データ所在地、冗長化方式を要件に照らして評価する。
- ウ：最も遠いリージョンを無条件に選ぶ。
- エ：データ所在地要件を無視し、価格だけで選ぶ。

答え・解説 正答：イ

- ア：ネットワーク遅延やデータ所在地に影響する。
 - イ：性能・コンプライアンス・可用性を同時に満たす配置を検討する必要がある。
 - ウ：遅延要件を悪化させる可能性がある。
 - エ：社内規程を満たさない。
- 総合解説：性能・コンプライアンス・可用性を同時に満たす配置を検討する必要がある。

対象：2026年度 / 基準日 2026-09-02

0027 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：標準

更新は少なく参照が非常に多い業務で、多少の反映遅延は許容できる。DB方式設計で有効な候補はどれか。

ア：全参照を更新ノード1台へ集中させるだけ。

イ：参照のたびに全DBをバックアップする。

ウ：更新系から非同期レプリカへ複製し、参照をレプリカへ分散する。

エ：索引をすべて削除すれば参照性能が必ず上がる。

答え・解説 正答：ウ

ア：負荷分散にならない。

イ：参照性能改善の方式ではない。

ウ：参照負荷を分散できるが、非同期複製では最新性の遅延を要件として許容できるか確認が必要である。

エ：一般に検索性能を悪化させる可能性が高い。

総合解説：参照負荷を分散できるが、非同期複製では最新性の遅延を要件として許容できるか確認が必要である。

対象：2026年度 / 基準日 2026-09-02

0028 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：標準

オンプレミスDBとクラウドDBを比較するとき、クラウド側のコスト評価として最も適切なものはどれか。

ア：クラウドは必ずオンプレミスより安いと前提する。

イ：バックアップや転送費は常に0円と仮定する。

ウ：月額料の最小インスタンス価格だけで比較する。

エ：計算資源だけでなく、ストレージ、IO、バックアップ、データ転送、冗長化、運用工数なども含めてTCOを比較する。

答え・解説 正答：エ

ア：負荷や利用期間によって結果は異なる。

イ：サービス条件を確認せず無視できない。

ウ：実運用の容量・IO・転送・冗長化費用を見落とす。

エ：従量課金の対象は複数あり、単一単価だけでは総費用を評価できない。

総合解説：従量課金の対象は複数あり、単一単価だけでは総費用を評価できない。

対象：2026年度 / 基準日 2026-09-02

0029 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：応用

巨大なセンサイイベントをキー単位で高速蓄積し、スキーマ変更が頻繁で、複雑な多表結合はほぼない。一方で一部の基幹マスタは厳密な参照整合性が必要である。最も適切な方式設計はどれか。

ア：データ特性ごとに適性を評価し、イベント系と基幹マスタで異なるデータ管理方式を組み合わせることも検討する。

イ：センサイイベントを紙に出力して保存する。

ウ：全データを必ず一つの方式に統一し、要件差は無視する。

エ：基幹マスタも整合性制約を一切持たない方式だけに移す。

答え・解説 正答：ア

ア：一律に一製品へ寄せるのではなく、整合性・問い合わせ特性・スケール・運用性を要件ごとに評価する。

イ：検索・蓄積・処理要件に適さない。

ウ：異なるワークロード特性を満たせない可能性がある。

エ：厳密な参照整合性要件を満たせない。

総合解説：一律に一製品へ寄せるのではなく、整合性・問い合わせ特性・スケール・運用性を要件ごとに評価する。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0030 4-1 要件定義・方式設計 > RDBMS・クラウド・システム方式選定 | 難易度：応用

同一物理ホスト上の仮想マシン2台にDBを冗長配置し、「ホスト障害にも耐える」としている。評価として最も適切なものはどれか。

ア：VMが2台なら物理ホスト障害も必ず耐えられる。

イ：同一ホスト障害で2台とも停止する可能性があるため、障害ドメインを分離した配置を検討すべきである。

ウ：暗号化を有効にすればホスト障害に耐えられる。

エ：バックアップを圧縮すれば同一ホスト配置でも無停止になる。

答え・解説 正答：イ

ア：同一ホストに共存していれば共通障害点が残る。

イ：冗長化では論理的な台数だけでなく、電源・ホスト・ラック・AZなど共通障害点の分離が重要である。

ウ：暗号化は可用性の障害ドメイン分離を代替しない。

エ：バックアップはリアルタイム冗長化ではない。

総合解説：冗長化では論理的な台数だけでなく、電源・ホスト・ラック・AZなど共通障害点の分離が重要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0031 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：基礎

DBストレージ容量を見積もる際、表データ以外に考慮すべき代表的な要素はどれか。

ア：現在の表データ量だけをそのまま必要容量とし、付随領域や増加分は見込まない。

イ：CPUコア数をストレージ容量へ換算して見積もる。

ウ：索引、ログ、一時領域、バックアップ、空き領域、将来増加分。

エ：現在の空き容量だけを確認し、データ増加率やバックアップ保持量は考慮しない。

答え・解説 正答：ウ

ア：表データ以外にもDB運用に必要な領域があり、現在値だけでは将来容量を不足させる可能性がある。

イ：CPUコア数はストレージ容量を直接決める指標ではない。

ウ：DB容量設計では表に加え、索引、ログ、一時領域、バックアップ、運用余裕、増加見込みを考慮する。

エ：現時点の空き容量だけでは将来の需要と保持要件を評価できない。

総合解説：実運用では表本体以外にも複数の領域が必要で、成長や保守作業の余裕も考慮する。

対象：2026年度 / 基準日 2026-09-02

0032 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：基礎

小さなランダムIOを大量に処理するOLTPで、特に重要になりやすいストレージ指標はどれか。

ア：大容量の連続読取りだけを想定したシーケンシャル帯域幅

イ：CPUのクロック周波数だけ

ウ：DBサーバ外部とのネットワーク帯域だけ

エ：IOPSとIOレイテンシ

答え・解説 正答：エ

ア：帯域幅も重要だが、小さなランダムIO中心のOLTPではI/O回数と応答遅延の方が直接的な評価軸になりやすい。

イ：CPU性能だけではストレージの小粒度ランダムアクセス性能を評価できない。

ウ：ネットワーク性能だけではストレージ装置内部のI/O処理能力を評価できない。

エ：小さなランダムアクセスでは単位時間あたりI/O回数と1回当たりの遅延が性能へ強く影響する。

総合解説：小さなランダムアクセスでは単位時間あたりI/O回数と応答遅延が性能へ強く影響する。

対象：2026年度 / 基準日 2026-09-02

0033 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：基礎

DBMSの表領域（tablespace）等の仕組みを利用する主な目的として最も適切なものはどれか。

- ア：テーブルや索引などのDBオブジェクトを特定のストレージ領域へ配置する管理単位として使う。
- イ：SQLの意味を自動的に正規化する。
- ウ：主キーの候補を自動で決める。
- エ：利用者のパスワードを表示する。

答え・解説 正答：ア

ア：表領域はオブジェクトの物理配置やストレージ特性の使い分けに利用できる。

イ：表領域はSQL意味解析の機能ではない。

ウ：論理設計のキー選定機能ではない。

エ：認証情報表示機能ではない。

総合解説：表領域はオブジェクトの物理配置やストレージ特性の使い分けに利用できる。

対象：2026年度 / 基準日 2026-09-02

0034 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：標準

現在の表データが500GBで、今後3年間は毎年100GBずつ増える見込みである。3年後の表データに対し索引等の追加領域を25%見込む。単純計算で必要容量はおよそいくらか。

- ア：800GB（3年後の表データだけで、25%追加領域を加えていない）
- イ：1,000GB（ $(500+100 \times 3) \times 1.25$ ）
- ウ：1,500GB（増加分と25%を重複して過大計上した値）
- エ：625GB（現在500GBに25%だけ加え、3年間の増加を無視した値）

答え・解説 正答：イ

ア：3年後の表データは800GBだが、追加領域25%をまだ加えていない。

イ： $500+300=800\text{GB}$ 、 $800 \times 1.25=1000\text{GB}$ である。

ウ：条件からはこの大きさにはならない。

エ：将来3年間の増加300GBを無視している。

総合解説：3年後の表データは $500+100 \times 3=800\text{GB}$ 。追加25%を見込むと $800 \times 1.25=1,000\text{GB}$ である。

対象：2026年度 / 基準日 2026-09-02

0035 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：標準

大規模なソートやハッシュ結合を夜間バッチで実行する。通常時は空き容量が十分でも、バッチ時だけ一時領域不足が発生する。改善策として最も適切なものはどれか。

ア：SQLのコメントを短くして容量を確保する。

イ：表データ件数だけ見て一時領域は無視する。

ウ：ピーク処理の一時領域使用量を測定・見積りし、専用領域や余裕容量を含めて設計する。

エ：エラー時に不要データを手作業で削除する運用だけにする。

答え・解説 正答：ウ

ア：一時処理の主要容量とは無関係である。

イ：ピーク時不足が再発する。

ウ：容量計画は定常データだけでなく、ピーク時に必要となる一時領域も対象にする必要がある。

エ：予防的な容量設計になっていない。

総合解説：容量計画は定常データだけでなく、ピーク時に必要となる一時領域も対象にする必要がある。

対象：2026年度 / 基準日 2026-09-02

0036 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：標準

直近3か月の注文は頻繁に検索されるが、5年以上前の注文は法定保存のため残すだけで参照頻度が非常に低い。物理配置として有力な考え方はどれか。

ア：全履歴を最も高価な高速層に固定し、コストは評価しない。

イ：古いデータは要件確認なしに削除する。

ウ：直近データを最も遅い層へ移す。

エ：アクセス頻度と性能要件に応じ、ホットデータを高速層、低頻度データを低コスト層へ配置する。

答え・解説 正答：エ

ア：低頻度データまで高性能層に置く必要性を検討すべきである。

イ：保存義務を満たせない。

ウ：頻繁な検索性能を悪化させる可能性がある。

エ：データライフサイクルとアクセス特性に応じた階層化で、性能とコストの両立を図れる。

総合解説：データライフサイクルとアクセス特性に応じた階層化で、性能とコストの両立を図れる。

対象：2026年度 / 基準日 2026-09-02

0037 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：標準

データ圧縮を物理設計に採用する評価として最も適切なものはどれか。

- ア：容量とIO削減の効果だけでなく、圧縮・展開によるCPU負荷や更新特性も測定して判断する。
- イ：圧縮するとデータ整合性制約が不要になる。
- ウ：圧縮率だけで製品を選び、復旧や更新性能は見ない。
- エ：圧縮すれば必ず全SQLが高速化する。

答え・解説 正答：ア

- ア：圧縮はストレージ・IOを削減できる一方、CPUや更新コストなどのトレードオフがある。
 - イ：圧縮と整合性制約は別の機能である。
 - ウ：運用全体の要件評価が不足している。
 - エ：処理内容やCPUボトルネックによっては逆効果もあり得る。
- 総合解説：圧縮はストレージ・IOを削減できる一方、CPUや更新コストなどのトレードオフがある。

対象：2026年度 / 基準日 2026-09-02

0038 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：標準

独立した物理ストレージ装置を複数利用できる環境で、データファイルのランダムIOとトランザクションログの連続書込みが競合している。改善案として有力なものはどれか。

- ア：すべての索引を削除するだけでログIOがなくなる。
- イ：IO特性を測定した上で、データとログを異なる独立デバイスへ配置して競合を減らす。
- ウ：論理ドライブ名だけ変え、同じ物理装置に置いたまま必ず高速化すると考える。
- エ：ログを削除して書込みをなくす。

答え・解説 正答：イ

- ア：更新ログそのものは必要であり、索引削除だけでは解決しない。
 - イ：物理デバイスが実際に独立しているなら、異なるIO特性を分離することで競合を緩和できる場合がある。
 - ウ：同一物理資源なら競合が残る可能性がある。
 - エ：障害回復能力を失う。
- 総合解説：物理デバイスが実際に独立しているなら、異なるIO特性を分離することで競合を緩和できる場合がある。

対象：2026年度 / 基準日 2026-09-02

0039 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：応用

現在使用量がストレージの70%で、月5%ずつデータが増えている。追加容量の調達に2か月かかる。最も適切な運用設計はどれか。

ア：100%になるまで何もしない。

イ：空き容量を監視せず、利用者から障害連絡が来たら対応する。

ウ：増加率と調達リードタイムを基に閾値を設定し、枯渇前に増設を開始できるよう継続監視する。

エ：増加率を無視して毎月1GBだけ固定で増設する。

答え・解説 正答：ウ

ア：増設が間に合わず停止リスクが高い。

イ：予防的管理になっていない。

ウ：キャパシティ管理では現在値だけでなく成長率と調達時間を考慮した予測が必要である。

エ：実際の成長に追従できない可能性が高い。

総合解説：キャパシティ管理では現在値だけでなく成長率と調達時間を考慮した予測が必要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0040 4-2 物理データベース設計 > データ量・容量・ストレージ設計 | 難易度：応用

仮想ストレージ上でDBデータとログを別ボリュームに分けたが、両方が同じ物理ディスク群へ割り当てられている。性能評価として最も適切なものはどれか。

ア：仮想ストレージではIO性能測定は不要である。

イ：ログをメモリだけに置けば永続性は必ず保証される。

ウ：ボリューム名が違えば物理競合は必ずなくなる。

エ：論理分離だけでは物理IO競合が解消しない可能性があるため、下位の物理配置と実測IOを確認する。

答え・解説 正答：エ

ア：実測しなければボトルネックを把握できない。

イ：障害時に失われるため永続性要件を満たさない。

ウ：同じ物理資源を共有していれば競合し得る。

エ：仮想化層のボリューム分離と物理資源分離は同義ではない。

総合解説：仮想化層のボリューム分離と物理資源分離は同義ではない。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0041 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：基礎

注文表を注文日で月単位に分割する方式として最も直接的なものはどれか。

ア：RANGEパーティショニング

イ：ビューだけを作れば物理分割になる。

ウ：LISTパーティショニングだけが日付範囲を表せる。

エ：HASHパーティショニングで月境界を直接定義する。

答え・解説 正答：ア

ア：日付の連続範囲ごとに分割する用途にはRANGEが適する。

イ：通常のビューは物理データ分割ではない。

ウ：LISTは列挙値による分割が中心である。

エ：HASHはハッシュ値で分散し、月範囲を直接表現しない。

総合解説：日付の連続範囲ごとに分割する用途にはRANGEが適する。

対象：2026年度 / 基準日 2026-09-02

0042 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：基礎

地域コードが「東日本」「西日本」「海外」のような離散カテゴリで、カテゴリごとにデータを分けたい。代表的な方式はどれか。

ア：RANGEで必ず連続数値に変換しなければならない。

イ：LISTパーティショニング

ウ：パーティションを使うと主キーが不要になる。

エ：全行を同じパーティションに置く。

答え・解説 正答：イ

ア：離散カテゴリならLISTを直接検討できる。

イ：離散的な値集合ごとに分割する用途にLISTが適する。

ウ：論理的な識別要件とは別問題である。

エ：分割の目的を達成しない。

総合解説：離散的な値集合ごとに分割する用途にLISTが適する。

対象：2026年度 / 基準日 2026-09-02

0043 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：基礎

連続範囲による偏りを避け、キーに基づいて複数パーティションへ概ね均等分散したい。候補となる方式はどれか。

ア：全件を同じ表に集約し分割しない。

イ：ORDER BYだけを設定する。

ウ：HASHパーティショニング

エ：RANGEを1パーティションだけ作る。

答え・解説 正答：ウ

ア：分散の目的を達成しない。

イ：結果順序であり物理分割ではない。

ウ：ハッシュ値に基づく分割は、キー分布が適切なら複数領域へ負荷を分散しやすい。

エ：均等分散にならない。

総合解説：ハッシュ値に基づく分割は、キー分布が適切なら複数領域へ負荷を分散しやすい。

対象：2026年度 / 基準日 2026-09-02

0044 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：標準

注文日で月次RANGEパーティションを構成している。2026年8月分だけ検索するSQLで期待できる効果として最も適切なものはどれか。

ア：パーティション化すると索引は一切利用できなくなる。

イ：全パーティションを必ず完全走査する。

ウ：パーティション化するとSQL条件は不要になる。

エ：注文日の条件から不要な月のパーティションを除外できれば、走査範囲を減らせる。

答え・解説 正答：エ

ア：パーティションごとの索引等を利用できるDBMSがある。

イ：ブルーニングが有効なら不要パーティションを除外できる。

ウ：必要データを指定する条件は引き続き必要である。

エ：パーティションキーに適合する条件によりパーティションブルーニングが働くと、不要領域のアクセスを避けられる。

総合解説：パーティションキーに適合する条件によりパーティションブルーニングが働くと、不要領域のアクセスを避けられる。

対象：2026年度 / 基準日 2026-09-02

0045 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：標準

月次パーティションで10年分の履歴を管理している。保持期限を過ぎた1か月分を迅速に削除したい。パーティション化の利点として期待できるものはどれか。

ア：対象月のパーティション単位で切離し・削除でき、行単位大量DELETEより保守しやすい場合がある。

イ：保持期限を過ぎても削除してはいけない。

ウ：全10年分を毎回DELETEし直す必要がある。

エ：パーティション化するとバックアップが禁止される。

答え・解説 正答：ア

ア：時間範囲で分割していれば、データライフサイクル管理をパーティション単位で行いやすい。

イ：業務要件に従って削除判断するのであり、パーティション方式が禁止するわけではない。

ウ：対象パーティション単位の操作が可能なDBMSでは不要である。

エ：バックアップとは両立する。

総合解説：時間範囲で分割していれば、データライフサイクル管理をパーティション単位で行いやすい。

対象：2026年度 / 基準日 2026-09-02

0046 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：標準

頻繁な集計を高速化するため、注文表に顧客名を重複保持する非正規化を検討する。評価として最も適切なものはどれか。

ア：非正規化すれば外部キーは必ず不要になる。

イ：参照時の結合削減効果と、顧客名変更時の重複更新・不整合リスクを比較して判断する。

ウ：正規化されたDBでは性能改善が一切できない。

エ：非正規化すれば更新時異状は必ず減る。

答え・解説 正答：イ

ア：参照整合性要件は別途存在する。

イ：非正規化は読取性能向上と引換えに冗長性や更新時異状を増やす可能性がある。

ウ：索引やSQL調整など多くの改善手段がある。

エ：冗長データの同期が必要になり、むしろ増え得る。

総合解説：非正規化は読取性能向上と引換えに冗長性や更新時異状を増やす可能性がある。

対象：2026年度 / 基準日 2026-09-02

0047 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：標準

全更新が単一の最新パーティションへ集中し、そのストレージだけIOが飽和している。設計見直しとして最も適切な考え方はどれか。

ア：全履歴を毎更新時に読み直す。

イ：最新パーティションをさらに同じ単一領域へ集約する。

ウ：更新キーと分割方式を再評価し、必要ならハッシュ等を組み合わせて書き込みを複数領域へ分散する。

エ：IO飽和を無視してCPUだけ増設する。

答え・解説 正答：ウ

ア：負荷を増大させる。

イ：ホットスポットを悪化させる。

ウ：時間範囲だけの分割では最新範囲に書き込みが集中する場合があるため、負荷特性に合わせて分割戦略を見直す。

エ：ボトルネックがストレージIOなら効果が限定的である。

総合解説：時間範囲だけの分割では最新範囲に書き込みが集中する場合があるため、負荷特性に合わせて分割戦略を見直す。

対象：2026年度 / 基準日 2026-09-02

0048 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：標準

分散DBで注文と注文明細をほぼ常に注文IDで結合する。ノード間通信を減らすための有力な配置はどれか。

ア：注文と注文明細を完全に無関係な乱数で別々に配置する。

イ：毎回全ノードから全データを中央へ集める。

ウ：分散DBではJOINを禁止する。

エ：注文IDに基づく同じ分割規則で、関連する注文と注文明細を同じノードへ配置する。

答え・解説 正答：エ

ア：結合時のネットワーク移動が増える可能性がある。

イ：通信量が大きくなる。

ウ：製品や要件によりJOINは可能であり、配置最適化を検討すべきである。

エ：関連データをコロケーションできれば、結合時のノード間データ移動を減らせる。

総合解説：関連データをコロケーションできれば、結合時のノード間データ移動を減らせる。

対象：2026年度 / 基準日 2026-09-02

0049 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：応用

ログデータは月単位で削除したい一方、月内の書き込み集中も避けたい。設計として有力な考え方はどれか。

- ア：月でRANGE分割し、各月内をHASH等でサブパーティション化する。
- イ：全期間を一つのパーティションへ戻す。
- ウ：パーティションキーを毎行ランダムに変更する。
- エ：月ごとに1パーティションだけとし、書き込み集中は考慮しない。

答え・解説 正答：ア

- ア：ライフサイクル管理と書き込み分散という二つの目的を、多段分割で両立できる場合がある。
 - イ：月単位削除の利点を失う。
 - ウ：同じ行の配置が不安定になり、管理しにくい。
 - エ：書き込みホットスポット要件を満たせない可能性がある。
- 総合解説：ライフサイクル管理と書き込み分散という二つの目的を、多段分割で両立できる場合がある。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0050 4-2 物理データベース設計 > パーティション・非正規化・データ配置 | 難易度：応用

テーブルは顧客ID検索が80%、月単位のアーカイブ処理が20%である。顧客IDの分布には偏りがある。パーティションキー選定として最も適切な進め方はどれか。

- ア：データ分布を調べず、列名の短い方を選ぶ。
- イ：検索条件、データ分布、更新集中、アーカイブ単位を実測し、単一方式または複合方式のトレードオフを評価する。
- ウ：検索比率80%だけ見て顧客IDを無条件に採用する。
- エ：アーカイブ20%だけ見て月だけを無条件に採用する。

答え・解説 正答：イ

- ア：負荷特性と無関係である。
 - イ：分割キーは一つの処理だけでなく、アクセス頻度・偏り・保守単位を総合評価して決める。
 - ウ：偏りによるホットスポットや月次保守を無視している。
 - エ：主要検索の効率を無視している。
- 総合解説：分割キーは一つの処理だけでなく、アクセス頻度・偏り・保守単位を総合評価して決める。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0051 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：基礎

RAID1の基本的な特徴として最も適切なものはどれか。

- ア：パリティなしで全ディスクを単純ストライピングし冗長性を持たない。
- イ：必ず3台以上で分散パリティを持つ。
- ウ：同じデータを複数ディスクへミラーリングして冗長性を持たせる。
- エ：データベースの論理正規化を行う。

答え・解説 正答：ウ

- ア：これはRAID0の特徴である。
 - イ：これはRAID5の一般的特徴に近い。
 - ウ：RAID1はミラーリングによってディスク障害への耐性を高める。
 - エ：RAIDはストレージ冗長化技術であり論理設計ではない。
- 総合解説：RAID1はミラーリングによってディスク障害への耐性を高める。

対象：2026年度 / 基準日 2026-09-02

0052 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：基礎

RAID0をDBの唯一の保存領域として使う場合の重要な注意点はどれか。

- ア：パリティで必ず2台同時故障に耐える。
- イ：バックアップが不要になる。
- ウ：1台故障してもミラーから必ず復旧できる。
- エ：ストライピングで性能向上を期待できるが冗長性がなく、1台のディスク障害で全体を失う可能性がある。

答え・解説 正答：エ

- ア：RAID0にはパリティがない。
 - イ：RAID種別にかかわらずバックアップは別途必要である。
 - ウ：RAID0にはミラーがない。
 - エ：RAID0は冗長性を提供しないため、可用性を必要とするDBの唯一の保存先には慎重な評価が必要である。
- 総合解説：RAID0は冗長性を提供しないため、可用性を必要とするDBの唯一の保存先には慎重な評価が必要である。

対象：2026年度 / 基準日 2026-09-02

0053 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：基礎

多数列を持つ巨大表から数列だけを読み、全件に近い範囲を集計する分析処理に向きやすい格納方式はどれか。

ア：列指向格納

イ：パーティションを作ると必ず列指向になる。

ウ：行指向だけが集計処理を実行できる。

エ：RAID0を使えば自動的に列指向になる。

答え・解説 正答：ア

ア：必要列だけを連続的に読みやすく、圧縮やベクトル化と相性がよいため分析ワークロードに向く場合が多い。

イ：パーティション方式と格納形式は独立した設計軸である。

ウ：行指向でも集計は可能だが、列指向が有利な場合がある。

エ：RAIDと行 / 列指向は別概念である。

総合解説：必要列だけを連続的に読みやすく、圧縮やベクトル化と相性がよいため分析ワークロードに向く場合が多い。

対象：2026年度 / 基準日 2026-09-02

0054 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：標準

小さなランダム更新が多いためにDBでRAID5を使うとき、RAID10と比べて注意すべき代表的な点はどれか。

ア：RAID5では読取りが一切できない。

イ：パリティ更新に伴う追加IOが発生し、書き込み性能が不利になる場合がある。

ウ：RAID5は必ずRAID10より高速である。

エ：RAID5には容量効率の利点がない。

答え・解説 正答：イ

ア：通常は読取り可能である。

イ：RAID5の小規模書き込みでは旧データ・旧パリティの読出しと新パリティ書き込み等のオーバーヘッドが生じ得る。

ウ：ワークロードによって異なり、書き込みでは不利な場合がある。

エ：ミラー方式より容量効率が高い場合がある。

総合解説：RAID5の小規模書き込みでは旧データ・旧パリティの読出しと新パリティ書き込み等のオーバーヘッドが生じ得る。

対象：2026年度 / 基準日 2026-09-02

0055 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：標準

4台構成の典型的なRAID10について、障害耐性の説明として最も適切なものはどれか。

ア：1台故障しただけで必ず全体停止する。

イ：パリティだけで構成され、ミラーは使わない。

ウ：ミラーペアごとに1台ずつ故障する組合せなら2台故障でも継続できる場合があるが、同一ミラーペアの2台故障には耐えられない。

エ：どの2台が同時故障しても必ず継続できる。

答え・解説 正答：ウ

ア：ミラーによる冗長性がある。

イ：RAID10はミラーとストライピングを組み合わせる。

ウ：RAID10の耐障害性はどのディスクが故障したかに依存する。

エ：同一ミラーペアの両方が故障するとデータを失う。

総合解説：RAID10の耐障害性はどのディスクが故障したかに依存する。

対象：2026年度 / 基準日 2026-09-02

0056 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：標準

RAID1でディスクを二重化しているDBについて、「バックアップは不要」とする評価が誤っている主な理由はどれか。

ア：RAID1はネットワーク暗号化機能だから。

イ：RAID1ではデータを保存できないから。

ウ：バックアップを取るとRAID1が無効になるから。

エ：誤削除、論理破損、マルウェア、複数障害などはミラーへ同時反映され得るため、過去時点へ戻すバックアップが別に必要だから。

答え・解説 正答：エ

ア：ストレージのミラーリング方式である。

イ：RAID1はデータ保存に利用できる。

ウ：両者は併用できる。

エ：RAIDは主にハードウェア障害への可用性対策であり、論理障害や履歴復旧の代替ではない。

総合解説：RAIDは主にハードウェア障害への可用性対策であり、論理障害や履歴復旧の代替ではない。

対象：2026年度 / 基準日 2026-09-02

0057 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：標準

主キーで1件を検索し、その行の多数列を参照・更新する処理が中心のOLTPで、行指向格納が向きやすい主な理由はどれか。

ア：1行の列が近接して格納されるため、行単位アクセスや更新に適しやすい。

イ：全列を別々の場所に格納するから。

ウ：行指向では索引が不要になるから。

エ：行指向ならトランザクション制御が不要だから。

答え・解説 正答：ア

ア：点検索・行単位更新では行指向のアクセス局所性が有利になりやすい。

イ：それは列指向の特徴に近い。

ウ：索引要否は検索特性で決まる。

エ：並行制御や整合性は依然必要である。

総合解説：点検索・行単位更新では行指向のアクセス局所性が有利になりやすい。

対象：2026年度 / 基準日 2026-09-02

0058 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：標準

列指向ストレージで高い圧縮率を得やすい理由として最も適切なものはどれか。

ア：すべての値をNULLへ変換するから。

イ：同じ列には似た型・値分布のデータが連続しやすく、辞書圧縮やランレングス等が効きやすいから。

ウ：RAIDのパリティを使って圧縮するから。

エ：列指向ではデータを保存しないから。

答え・解説 正答：イ

ア：データ意味を失うため誤りである。

イ：列単位で同種データが集まるため、値の規則性を利用した圧縮に向く。

ウ：RAIDのパリティと列圧縮は別概念である。

エ：実際にデータを格納する。

総合解説：列単位で同種データが集まるため、値の規則性を利用した圧縮に向く。

対象：2026年度 / 基準日 2026-09-02

0059 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：応用

DBのレスポンスが遅い。CPU使用率は30%だが、ストレージ待ち時間が長く、ランダムIOPSが装置上限に達している。改善の優先順位として最も適切なものはどれか。

ア：ログを無効化してIOを減らす。

イ：CPUが30%なのでCPUを10倍にすれば必ず解決する。

ウ：IOがボトルネックと判断し、アクセス経路、キャッシュ、索引、データ配置、より高IOPSのストレージなどを検討する。

エ：待ち時間を測定せず全テーブルを非正規化する。

答え・解説 正答：ウ

ア：障害回復・永続性を損なう。

イ：主要ボトルネックがIOならCPU増強だけでは効果が限定的である。

ウ：性能改善では実測ボトルネックに対して資源・SQL・配置を調整する。

エ：根拠のない設計変更で副作用が大きい。

総合解説：性能改善では実測ボトルネックに対して資源・SQL・配置を調整する。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0060 4-2 物理データベース設計 > RAID・行指向 / 列指向・物理構成 | 難易度：応用

基幹DBでは小さな更新トランザクションが多い一方、同じデータを使う全件集計が日中に多数実行され、OLTP性能を圧迫している。設計案として最も適切なものはどれか。

ア：分析を速くするため基幹DBの全索引を削除する。

イ：更新中も毎回全件スキャンを基幹DBで実行し続ける。

ウ：分析結果を手作業で紙に転記する。

エ：OLTP向け行指向系と分析向け列指向レプリカ/DWHなどを分離し、鮮度要件を踏まえて連携する案を評価する。

答え・解説 正答：エ

ア：OLTP検索を悪化させる可能性が高い。

イ：ワークロード競合を解消しない。

ウ：データ鮮度・自動化・品質の要件に適さない。

エ：異なるワークロードを同一物理構成へ無理に載せるより、用途別に最適化し連携することで干渉を減らせる場合がある。

総合解説：異なるワークロードを同一物理構成へ無理に載せるより、用途別に最適化し連携することで干渉を減らせる場合がある。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0061 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：基礎

IPAシラバスに示されるDB実装の基本的な定義順序に最も近いものはどれか。

ア：データベース定義→テーブル定義→インデックス定義→ビュー定義→アクセス権設定。

イ：ビュー定義→DB削除→テーブル定義。

ウ：索引定義だけ行い、テーブルは定義しない。

エ：アクセス権設定→バックアップ削除→データベース定義。

答え・解説 正答：ア

ア：物理・論理設計をRDBMS上の具体的オブジェクトへ順に実装し、最後に権限を設定する流れが示されている。

イ：ビューは参照する表等が存在する必要がある。

ウ：索引の対象となる表が必要である。

エ：基本的な依存関係の順序と合わない。

総合解説：物理・論理設計をRDBMS上の具体的オブジェクトへ順に実装し、最後に権限を設定する流れが示されている。

対象：2026年度 / 基準日 2026-09-02

0062 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：基礎

通常のビュー（materialized viewではない）の説明として最も適切なものはどれか。

ア：ビューを作ると主キーが自動削除される。

イ：問い合わせ定義を名前付きで提供し、基表のデータを論理的に見せる。

ウ：常に独立した複製データを物理保存する。

エ：基表がなくても必ず更新可能である。

答え・解説 正答：イ

ア：主キー制約とは無関係である。

イ：通常ビューはクエリ定義を通じた論理的な表現を提供し、基表の変更を参照時に反映する。

ウ：これはマテリアライズドビュー等の性質に近い。

エ：ビューは基表等に依存し、更新可能性にも条件がある。

総合解説：通常ビューはクエリ定義を通じた論理的な表現を提供し、基表の変更を参照時に反映する。

対象：2026年度 / 基準日 2026-09-02

0063 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：基礎

索引の役割として最も適切なものはどれか。

ア：索引があれば表データは不要になる。

イ：索引を作ると全更新が無料になる。

ウ：検索条件や結合条件などに対するアクセス経路を提供し、対象行の探索を高速化する。

エ：索引は権限管理だけのために使う。

答え・解説 正答：ウ

ア：索引は通常、基表データを完全に代替するものではない。

イ：索引の維持には更新コストがかかる。

ウ：索引はデータ行へのアクセス経路であり、検索性能を改善するために利用される。

エ：主目的はアクセス経路の提供である。

総合解説：索引はデータ行へのアクセス経路であり、検索性能を改善するために利用される。

対象：2026年度 / 基準日 2026-09-02

0064 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：標準

注文明細(order_item)が注文(order)を参照する外部キーを実装する。事前に確認すべき事項として最も重要なものはどれか。

ア：参照先キーの型は参照元と無関係でよい。

イ：外部キーを付ける前に親表を削除する。

ウ：外部キーを追加すれば孤児データが自動的に正しい注文へ割り当てられる。

エ：既存の注文明細に参照先が存在しない孤児データがないか、キー型や値が整合しているか確認する。

答え・解説 正答：エ

ア：互換性のある型と意味が必要である。

イ：参照関係を実装できなくなる。

ウ：DBMSが業務上正しい親を推測できるわけではない。

エ：既存データが制約を満たさなければ外部キー追加に失敗するか、データ修正が必要になる。

総合解説：既存データが制約を満たさなければ外部キー追加に失敗するか、データ修正が必要になる。

対象：2026年度 / 基準日 2026-09-02

0065 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：標準

検索性能を上げるため、多数の索引を追加しようとしている。最も適切な評価はどれか。

ア：検索改善効果と、INSERT/UPDATE/DELETE時の索引維持コスト・容量増加を実測して必要な索引だけ採用する。

イ：索引を追加すると自動的にデータ圧縮率が100%になる。

ウ：索引は多いほど常に性能が上がる。

エ：索引は更新処理には一切影響しない。

答え・解説 正答：ア

ア：索引は読取を高速化できる一方、更新と容量にコストを追加する。

イ：そのような効果はない。

ウ：過剰索引は更新負荷と容量を増やす。

エ：更新時に索引エントリの維持が必要である。

総合解説：索引は読取を高速化できる一方、更新と容量にコストを追加する。

対象：2026年度 / 基準日 2026-09-02

0066 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：標準

人事テーブルには給与や個人番号など機密列がある。一般部門には氏名と部署だけ参照させたい。実装案として有力なものはどれか。

ア：基表に全員のUPDATE権限を付与する。

イ：必要列だけを公開するビューを作成し、そのビューへのSELECT権限だけを付与する。

ウ：機密列を列名変更するだけで保護する。

エ：全列をCSVで全員へ配布する。

答え・解説 正答：イ

ア：過大権限で機密列の改変も可能になる。

イ：ビューと権限制御を組み合わせることで、利用者に必要な列だけを論理的に公開できる。

ウ：列名の難読化はアクセス制御ではない。

エ：機密情報の漏えいにつながる。

総合解説：ビューと権限制御を組み合わせることで、利用者に必要な列だけを論理的に公開できる。

対象：2026年度 / 基準日 2026-09-02

0067 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：標準

巨大テーブルへ列・索引を追加する本番変更を計画する。最も適切な進め方はどれか。

ア：本番ピーク時間に無検証で実行する。

イ：DDL実行中はバックアップや復旧計画を考えなくてよい。

ウ：対象DBMSで必要なロック・書換え・所要時間を事前検証し、オンライン機能や段階的の変更を含めて停止影響を評価する。

エ：DDLは必ず瞬時なので影響調査は不要である。

答え・解説 正答：ウ

ア：業務影響が大きい。

イ：変更失敗に備えた復旧手段が必要である。

ウ：DDLは製品・操作によって長時間ロックや表書換えを伴うことがあるため、実測と変更計画が必要である。

エ：大規模表では長時間処理やロックが発生し得る。

総合解説：DDLは製品・操作によって長時間ロックや表書換えを伴うことがあるため、実測と変更計画が必要である。

対象：2026年度 / 基準日 2026-09-02

0068 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：標準

集計に30秒かかる分析SQLを頻繁に参照するが、結果は5分程度古くてもよい。候補として適切なものはどれか。

ア：毎参照時に全バックアップを復元する。

イ：通常ビューを作れば必ず集計結果が物理保存される。

ウ：基表を削除して集計結果だけ残す。

エ：集計結果を物理保持するマテリアライズドビュー等を用い、必要な更新間隔でリフレッシュする。

答え・解説 正答：エ

ア：参照性能改善には適さない。

イ：通常ビューは通常、参照時に基礎クエリを評価する。

ウ：元データや他用途を失う。

エ：多少の鮮度遅延を許容できるなら、事前計算で参照性能を改善できる。

総合解説：多少の鮮度遅延を許容できるなら、事前計算で参照性能を改善できる。

対象：2026年度 / 基準日 2026-09-02

0069 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：応用

問い合わせの大半が「customer_id = ? AND order_date BETWEEN ? AND ?」で、customer_idは高選択度、order_dateは範囲条件である。一般的なB-tree複合索引の候補としてまず検討しやすいものはどれか。

ア：(customer_id, order_date) の順の複合索引。

イ：注文金額だけの索引。

ウ：(order_date, customer_id)だけが常に唯一の正解である。

エ：索引を一切作らず毎回全表走査する。

答え・解説 正答：ア

ア：等価条件を先頭、範囲条件を後段に置くことで、顧客を絞った後に日付範囲を効率よく走査できる候補となる。

イ：主要な検索条件を直接支援しない。

ウ：負荷やDBMS次第だが、提示条件ではcustomer_id先頭が有力であり「常に唯一」とは言えない。

エ：大規模表で主要検索を支援できない可能性が高い。

総合解説：等価条件を先頭、範囲条件を後段に置くことで、顧客を絞った後に日付範囲を効率よく走査できる候補となる。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0070 4-3 実装・セキュリティ > DB・テーブル・索引・ビュー実装 | 難易度：応用

テーブル・索引・ビュー・権限を実装した後の確認として最も適切なものはどれか。

ア：テストデータは不要で、本番データだけで初めて確認する。

イ：テストデータをロードし、代表SQLの応答、制約違反時の動作、権限拒否、障害復旧などを要件に照らして検証する。

ウ：権限は付与できることだけ確認し、拒否されるべき操作は試さない。

エ：DDLがエラーなく実行できれば全要件を満たしたとみなす。

答え・解説 正答：イ

ア：安全な事前検証ができない。

イ：IPAシラバスではテストデータのロード、応答シミュレーション、復旧シミュレーションが実装工程に含まれる。

ウ：最小権限が正しく実装されたか確認できない。

エ：性能・セキュリティ・整合性・復旧は別途検証が必要である。

総合解説：IPAシラバスではテストデータのロード、応答シミュレーション、復旧シミュレーションが実装工程に含まれる。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0071 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：基礎

多数の利用者に同じ職務権限を付与・変更する場合、管理しやすい方式はどれか。

ア：パスワードを全員共通にする。

イ：全利用者へ個別にDBA権限を付与する。

ウ：職務ごとのロールに権限を付与し、利用者をロールへ所属させる。

エ：権限表を紙だけで管理しDBには反映しない。

答え・解説 正答：ウ

ア：認証・追跡性を弱める。

イ：過大権限で管理も困難になる。

ウ：ロール単位の権限管理により、個人ごとの重複設定を減らし変更を一元化できる。

エ：実際のアクセス制御にならない。

総合解説：ロール単位の権限管理により、個人ごとの重複設定を減らし変更を一元化できる。

対象：2026年度 / 基準日 2026-09-02

0072 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：基礎

DBオブジェクトへのSELECT権限を利用者から取り消す代表的な操作はどれか。

ア：COMMIT

イ：CHECKPOINT

ウ：ANALYZE

エ：REVOKE

答え・解説 正答：エ

ア：トランザクション確定であり権限取消しではない。

イ：障害回復のための処理であり権限管理ではない。

ウ：統計情報更新等に用いる操作で権限取消しではない。

エ：GRANTは権限付与、REVOKEは付与済み権限の取消しに用いる。

総合解説：GRANTは権限付与、REVOKEは付与済み権限の取消しに用いる。

対象：2026年度 / 基準日 2026-09-02

0073 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：基礎

DBの保存データを暗号化する場合、鍵管理で最も重要な考え方はどれか。

ア：暗号鍵を保護し、生成・保管・配布・ローテーション・失効・復旧をライフサイクルとして管理する。

イ：同じ鍵を無期限に使い続け、失効手順を持たない。

ウ：鍵を利用者全員に共有して管理を簡単にする。

エ：暗号鍵を暗号化対象データと同じ場所に平文で保存する。

答え・解説 正答：ア

ア：暗号化の安全性は鍵の保護と運用に強く依存するため、鍵ライフサイクル管理が不可欠である。

イ：鍵漏えい時の影響を抑えにくい。

ウ：不正利用や追跡困難のリスクが高い。

エ：媒体流出時に鍵も同時に漏れるリスクが高い。

総合解説：暗号化の安全性は鍵の保護と運用に強く依存するため、鍵ライフサイクル管理が不可欠である。

対象：2026年度 / 基準日 2026-09-02

0074 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：標準

顧客テーブルの「氏名・住所」は参照可だが「個人番号」は一部担当者だけに限定したい。最も適切な実装方針はどれか。

ア：一般利用者にもUPDATE権限を追加する。

イ：列権限や限定ビューを用い、一般ロールには個人番号へアクセスできない権限を設定する。

ウ：全員へ表全体のSELECT権限を付与し、個人番号を見ないように注意喚起だけする。

エ：個人番号の列名をx1に変えるだけ。

答え・解説 正答：イ

ア：必要以上の権限を広げる。

イ：必要な列だけにアクセス権を限定することで最小権限を実現できる。

ウ：技術的なアクセス制御にならない。

エ：難読化は権限管理ではない。

総合解説：必要な列だけにアクセス権を限定することで最小権限を実現できる。

対象：2026年度 / 基準日 2026-09-02

0075 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：標準

同じ営業テーブルを使いながら、営業担当者には自分の担当顧客行だけ参照させたい。DB側で実装する候補として最も適切なものはどれか。

ア：テーブル名を営業担当者ごとに手で変更する。

イ：全員に全行を見せ、業務ルールだけで禁止する。

ウ：利用者・ロールに応じた行条件を適用する行レベルセキュリティポリシー。

エ：索引を削除して読みにくくする。

答え・解説 正答：ウ

ア：管理が煩雑で、アクセス制御の仕組みにならない。

イ：技術的にアクセスを防止できない。

ウ：行レベル制御により、同一表でも利用者ごとに参照可能な行を制限できる。

エ：性能を落とすだけでセキュリティにはならない。

総合解説：行レベル制御により、同一表でも利用者ごとに参照可能な行を制限できる。

対象：2026年度 / 基準日 2026-09-02

0076 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：標準

DBファイルを強力的に暗号化したので「DB管理者には全表のSELECT権限を付与しても問題ない」とする判断の評価として最も適切なものはどれか。

ア：正しい。暗号化すると監査ログも不要になる。

イ：誤りだが、理由は暗号化するとDBが一切起動できないから。

ウ：正しい。暗号化されていればSELECTしても平文は絶対に見えない。

エ：誤り。DBが復号して処理できる権限を持つ主体には平文が見えるため、暗号化とは別に最小権限が必要である。

答え・解説 正答：エ

ア：監査と暗号化は別の統制である。

イ：正しく設定すればDBは暗号化データを扱える。

ウ：DB処理時には適切な鍵で復号されるため、権限があれば閲覧できる。

エ：保存時暗号化は媒体流出等を保護するが、正規のDBアクセス経路での認可を代替しない。

総合解説：保存時暗号化は媒体流出等を保護するが、正規のDBアクセス経路での認可を代替しない。

対象：2026年度 / 基準日 2026-09-02

0077 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：標準

DBバックアップをクラウドストレージへ保存し、暗号化も行う。鍵保護としてより適切な方式はどれか。

ア：バックアップとは独立した鍵管理サービスやHSM等で鍵を保護し、アクセス権を分離する。

イ：鍵を全利用者にメール配布する。

ウ：暗号化バックアップと復号鍵を同じファイルへ平文で入れる。

エ：復号鍵をソースコードへ固定文字列として公開する。

答え・解説 正答：ア

ア：暗号文と復号鍵の保護境界を分けると、一つの媒体侵害で両方が漏れるリスクを下げられる。

イ：アクセス制御と追跡性が弱くなる。

ウ：媒体が漏れいすると鍵も同時に漏れる。

エ：鍵の漏えい・ローテーション困難につながる。

総合解説：暗号文と復号鍵の保護境界を分けると、一つの媒体侵害で両方が漏れるリスクを下げられる。

対象：2026年度 / 基準日 2026-09-02

0078 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：標準

本番DBの機密データに対する不正操作リスクを下げるため、職務分離として最も適切なものはどれか。

ア：一人の共通アカウントへ全権限を集約する。

イ：DB運用、セキュリティ監査、アプリ開発などの権限を必要に応じて分離し、特権操作を記録・レビューする。

ウ：監査ログを特権利用者が自由に削除できるようにする。

エ：開発者全員へ本番DBA権限を付与する。

答え・解説 正答：イ

ア：誰が操作したか追跡しにくく、牽制が働かない。

イ：一人に全権限を集中させず、相互牽制と監査証跡を持たせることで不正・誤操作リスクを下げられる。

ウ：証跡の信頼性を損なう。

エ：最小権限と職務分離に反する。

総合解説：一人に全権限を集中させず、相互牽制と監査証跡を持たせることで不正・誤操作リスクを下げられる。

対象：2026年度 / 基準日 2026-09-02

0079 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：応用

脅威として「ストレージ媒体の盗難」「通信盗聴」「DB管理者による特定列の閲覧」を想定する。最も適切な考え方はどれか。

ア：TLSだけで保存媒体盗難も防げる。

イ：列名を難読化すれば暗号化は不要である。

ウ：保存時暗号化、通信暗号化、列単位暗号化やアクセス制御などを脅威ごとに組み合わせ、鍵・権限を分離する。

エ：ディスク暗号化だけで三つの脅威すべてを必ず防げる。

答え・解説 正答：ウ

ア：TLSは主に通信経路を保護する。

イ：難読化は暗号化ではない。

ウ：一つの暗号化方式ですべての脅威を防げるとは限らず、脅威モデルに応じた多層防御が必要である。

エ：正規DBアクセス権を持つ管理者の閲覧までは防げない場合がある。

総合解説：一つの暗号化方式ですべての脅威を防げるとは限らず、脅威モデルに応じた多層防御が必要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0080 4-3 実装・セキュリティ > 権限・暗号化・アクセス制御 | 難易度：応用

利用者Aから表TへのSELECT権限をREVOKEしたのに、AはまだTを参照できる。調査として最も適切なものはどれか。

ア：テーブル名を変更すれば権限問題は必ず解決する。

イ：REVOKEは必ずDBを再起動するまで無効なので待つ。

ウ：索引を削除すればSELECTできなくなる。

エ：Aが所属する他ロールやPUBLIC等を通じて権限を継承していないか、実効権限を確認する。

答え・解説 正答：エ

ア：権限設計を解決していない。

イ：権限継承を確認すべきであり、再起動が本質ではない。

ウ：索引はアクセス権を制御しない。

エ：権限は直接付与だけでなくロールメンバーシップやPUBLIC等から得ている場合がある。

総合解説：権限は直接付与だけでなくロールメンバーシップやPUBLIC等から得ている場合がある。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0081 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：基礎

大量データをDBへ取り込む際にRDBMSのバルクロードユーティリティを使う主な目的はどれか。

- ア：大量行を効率よく取り込み、通常の1行ずつの処理より高いスループットを得る。
- イ：データ型や制約の概念を完全になくす。
- ウ：バックアップを自動的に不要にする。
- エ：すべてのデータを暗号鍵へ変換する。

答え・解説 正答：ア

- ア：専用ロード機能は大量データ投入向けに最適化されていることが多い。
 - イ：ロード後もスキーマや整合性は必要である。
 - ウ：ロード方式とバックアップ要件は別である。
 - エ：データロードの目的ではない。
- 総合解説：専用ロード機能は大量データ投入向けに最適化されていることが多い。

対象：2026年度 / 基準日 2026-09-02

0082 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：基礎

外部CSVを本番テーブルへ取り込む前にステージング表へロードする主な利点はどれか。

- ア：本番表より必ず小さくしなければならない。
- イ：形式・値域・重複などを検証し、変換・クレンジングしてから本番へ反映できる。
- ウ：ステージング表は主キーを必ず持てない。
- エ：ステージング表ではデータ品質確認ができない。

答え・解説 正答：イ

- ア：容量要件次第であり、本質ではない。
 - イ：本番制約や業務データを直接汚さず、検証と変換の中間層を持てる。
 - ウ：必要に応じて制約や索引を設けられる。
 - エ：むしろ品質確認に利用できる。
- 総合解説：本番制約や業務データを直接汚さず、検証と変換の中間層を持てる。

対象：2026年度 / 基準日 2026-09-02

0083 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：基礎

本番の個人情報をテスト環境で利用する際にデータマスキングを行う主な目的はどれか。

ア：データを暗号化せず平文コピーすることをマスキングと呼ぶ。

イ：本番の個人情報をより読みやすくする。

ウ：テストに必要な形式や分布をある程度保ちつつ、実在個人を識別できる情報を露出しないようにする。

エ：テスト環境の全員に本番DBA権限を与える。

答え・解説 正答：ウ

ア：マスキングは識別情報を置換・匿名化する処理である。

イ：目的と逆である。

ウ：機密データをそのまま非本番へ持ち出すリスクを下げるためにマスキングを用いる。

エ：機密性を下げる。

総合解説：機密データをそのまま非本番へ持ち出すリスクを下げるためにマスキングを用いる。

対象：2026年度 / 基準日 2026-09-02

0084 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：標準

1,000万件のマスタデータを一括ロードした。完了判定として最も適切なものはどれか。

ア：コマンドが0秒で終わったので成功とする。

イ：先頭1件が見えれば全件成功とみなす。

ウ：エラーログを削除して問題がなかったことにする。

エ：入力件数、成功件数、拒否件数、主要制約、集計値やチェックサムなどを照合する。

答え・解説 正答：エ

ア：処理結果を検証していない。

イ：大量データの欠落や拒否を検出できない。

ウ：障害原因と再処理対象を失う。

エ：ユーティリティが終了しただけでは取り込み漏れや変換誤りを検出できないため、複数観点で照合する。

総合解説：ユーティリティが終了しただけでは取り込み漏れや変換誤りを検出できないため、複数観点で照合する。

対象：2026年度 / 基準日 2026-09-02

0085 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：標準

バルクロードで一部行だけ形式エラーになる。可用性と追跡性を両立する設計として適切なものはどれか。

- ア：正常行とエラー行を識別し、エラー理由と元データを保持して修正後に再処理できるようにする。
- イ：1件エラーが出たら毎回全データを永久に破棄する。
- ウ：エラー値をすべて0へ自動変換し記録しない。
- エ：エラー行を理由記録なしで捨てる。

答え・解説 正答：ア

- ア：大量ロードでは不正行の扱いを明確にし、追跡可能な再処理経路を設計することが重要である。
 - イ：再処理性がなく非効率である。
 - ウ：意味の異なる誤データを作る危険がある。
 - エ：データ欠損の原因を追跡できない。
- 総合解説：大量ロードでは不正行の扱いを明確にし、追跡可能な再処理経路を設計することが重要である。

対象：2026年度 / 基準日 2026-09-02

0086 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：標準

空の巨大表へ数億件を一括ロードし、その後に複数索引を必要とする。性能面で検討価値がある方針はどれか。

- ア：全索引を永久に削除し検索要件を無視する。
- イ：製品特性を確認し、ロード中の索引維持と、ロード後の一括索引作成のどちらが速いかベンチマークする。
- ウ：索引が多いほどロードは必ず高速になる。
- エ：索引はデータ投入と無関係なので検討不要である。

答え・解説 正答：イ

- ア：ロード後の利用性能を満たせない。
 - イ：大量投入では行ごとの索引更新より後から一括構築が有利な場合があるが、DBMSや可用性要件に依存する。
 - ウ：索引維持により書込みコストが増える。
 - エ：ロード性能へ影響する。
- 総合解説：大量投入では行ごとの索引更新より後から一括構築が有利な場合があるが、DBMSや可用性要件に依存する。

対象：2026年度 / 基準日 2026-09-02

0087 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：標準

高速化のため外部キー制約を一時的に無効化してデータをロードする案が出た。最も適切な条件はどれか。

ア：無効化したら再有効化しなくてよい。

イ：エラーがあっても検証せず本番公開する。

ウ：ロード後に参照整合性を完全検証し、制約を確実に再有効化できる手順とロールバックを用意する。

エ：制約を無効化すればデータ型も自動的に正しくなる。

答え・解説 正答：ウ

ア：恒常的に参照整合性を失う。

イ：不整合データを利用者へ提供する危険がある。

ウ：制約無効化は性能効果がある場合でも、不整合を混入させるリスクがあるため検証手順が不可欠である。

エ：型や値の問題は別途検証が必要である。

総合解説：制約無効化は性能効果がある場合でも、不整合を混入させるリスクがあるため検証手順が不可欠である。

対象：2026年度 / 基準日 2026-09-02

0088 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：標準

顧客IDをマスキングしたテストデータで、注文表のcustomer_idとの結合を維持したい。最も適切な方法はどれか。

ア：顧客表だけマスキングし、注文表は本番IDのままにする。

イ：顧客ID列をすべてNULLにする。

ウ：各出現箇所を独立した乱数に置換する。

エ：同じ元顧客IDには同じマスク値を一貫して割り当て、参照側にも同じ変換を適用する。

答え・解説 正答：エ

ア：参照整合性が壊れる。

イ：結合・テスト要件を満たせない。

ウ：同じ顧客でも表ごとに値が変わり、結合できなくなる。

エ：参照関係を保つには、関連テーブル間で決定的・一貫した変換が必要である。

総合解説：参照関係を保つには、関連テーブル間で決定的・一貫した変換が必要である。

対象：2026年度 / 基準日 2026-09-02

0089 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：応用

一括ロードを高速化するため、DBMSの「最小ログ」相当のモードを使う案がある。採用判断として最も適切なものはどれか。

ア：復旧方式、レプリケーション、バックアップ、障害時の再ロード可否への影響を確認し、RPO/RTTOを満たす場合だけ採用する。

イ：本番唯一データでもバックアップなしで実行する。

ウ：ログが少ないほど障害回復能力も必ず高くなる。

エ：ロード性能だけ測定し、障害時の挙動は確認しない。

答え・解説 正答：ア

ア：ログ削減は性能を改善できる場合があるが、回復性や複製方式に制約を生じることがあるため要件との整合確認が必要である。

イ：失敗時の復旧手段が不足する。

ウ：回復に必要な情報が減る可能性がある。

エ：可用性・復旧要件を評価できない。

総合解説：ログ削減は性能を改善できる場合があるが、回復性や複製方式に制約を生じることがあるため要件との整合確認が必要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0090 4-3 実装・セキュリティ > ユーティリティ・ロード・データマスキング | 難易度：応用

テストでは郵便番号形式・年齢分布・顧客間の世帯関係が重要だが、实在個人を特定できてはいけない。最も適切なマスキング設計はどれか。

ア：氏名だけ伏字にして住所・電話番号・個人番号はそのまま残す。

イ：必要な統計分布・形式・関係性を保つ変換や合成データを設計し、再識別リスクを評価して本番識別子を除去する。

ウ：全項目を同じ固定値にする。

エ：本番データを暗号化ZIPで配布し、全テスターへ復号パスワードを共有する。

答え・解説 正答：イ

ア：他属性から容易に再識別され得る。

イ：テスト有効性を維持しつつ個人識別性を下げるには、単純な一律置換ではなく要件に応じたマスキング・合成が必要である。

ウ：分布や関係性が失われ、テストとして無効になる可能性が高い。

エ：实在個人データの露出リスクが残る。

総合解説：テスト有効性を維持しつつ個人識別性を下げるには、単純な一律置換ではなく要件に応じたマスキング・合成が必要である。要件・物理設計・実装を分離し、性能・可用性・セキュリティ・運用性のトレードオフを根拠付きで判断することが重要である。

対象：2026年度 / 基準日 2026-09-02

0091 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：基礎

データベースアクセステストで、利用者要求への適合性を確認する対象の組合せとして最も適切なものはどれか。

- ア：性能、セキュリティ、インテグリティ、バックアップ・リカバリを要求条件に照らして確認する。
- イ：機能結果だけを確認し、性能や障害回復の試験は別要件として扱わない。
- ウ：ピーク性能だけを確認し、権限・整合性・障害回復は試験対象外とする。
- エ：バックアップ取得の成功だけを確認し、リストアや業務整合性は確認しない。

答え・解説 正答：ア

ア：DBアクセステストでは利用者要求に関わる性能、セキュリティ、整合性、障害回復などを総合的に検証する。

イ：機能が正しくても非機能要件を満たさない可能性がある。

ウ：性能だけでは機密性・完全性・回復可能性を確認できない。

エ：取得できても復元できなければ回復要件を満たしたとはいえない。

総合解説：DBアクセステストでは、単なるSQL構文確認だけでなく、性能・セキュリティ・整合性・回復性を含めて要件への適合を評価する。

対象：2026年度 / 基準日 2026-09-02

0092 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：基礎

二つのDB構成をベンチマークで比較する。比較結果の妥当性を高めるために最も重要な対応はどれか。

- ア：速い方だけ測定時間を短くする。
- イ：同じ代表ワークロード、データ量、測定条件を用いて比較する。
- ウ：一方だけキャッシュを事前に十分温める。
- エ：各構成で異なるSQLを使い、最良値だけを採用する。

答え・解説 正答：イ

ア：測定条件が異なるため、公平な比較にならない。

イ：比較対象以外の条件をそろえることで、性能差が構成差によるものかを判断しやすくなる。

ウ：キャッシュ状態の差が結果に混入し、構成そのものの比較が難しくなる。

エ：ワークロードが異なれば同一目的のベンチマークとして比較できない。

総合解説：比較対象以外の条件をそろえることで、性能差が構成差によるものかを判断しやすくなる。

対象：2026年度 / 基準日 2026-09-02

0093 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：基礎

本番データを複製して性能テスト環境で利用する必要がある。個人情報を含む列への対応として最も適切なものはどれか。

ア：本番と同じ個人情報を無条件に全テスト担当者へ公開する。

イ：個人情報列をすべて空文字にし、データ分布も件数も無視する。

ウ：テスト目的を満たす分布や形式を保ちつつ、識別可能性を下げるようデータをマスキングする。

エ：本番環境で直接負荷試験を行い、テスト環境を使わない。

答え・解説 正答：ウ

ア：必要最小限のアクセスや保護の原則に反し、漏えいリスクを高める。

イ：性能や制約を検証するために必要なデータ特性まで失う可能性がある。

ウ：テストで必要なデータ特性を保ちながら、不要な個人識別を防ぐことが重要である。

エ：本番サービスへの影響が大きく、通常は安全な検証方法とはいえない。

総合解説：テストで必要なデータ特性を保ちながら、不要な個人識別を防ぐことが重要である。

対象：2026年度 / 基準日 2026-09-02

0094 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：標準

オンライン処理の性能要件が「95%以上の要求を2秒以内に応答」である。テスト結果の判定方法として最も適切なものはどれか。

ア：平均応答時間が2秒未満なら、遅い要求が多数あっても合格とする。

イ：最速1件の応答時間だけを測る。

ウ：CPU使用率だけを測り、応答時間は測らない。

エ：応答時間の分布や95パーセンタイルを確認し、要件を満たす要求割合を評価する。

答え・解説 正答：エ

ア：平均値は裾の遅延を隠すため、95%要件を直接検証できない。

イ：代表性がなく、利用者全体の応答性能を評価できない。

ウ：CPU使用率は原因分析には有用だが、要求された応答時間SLOの判定指標ではない。

エ：割合で定義されたSLOは平均値だけでなく分布・パーセンタイルで評価する必要がある。

総合解説：割合で定義されたSLOは平均値だけでなく分布・パーセンタイルで評価する必要がある。

対象：2026年度 / 基準日 2026-09-02

0095 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：標準

想定ピーク負荷を超えて負荷を増加させ、性能劣化や限界点、回復挙動を確認する試験として最も適切なものはどれか。

- ア：ストレステスト
- イ：単体テスト
- ウ：静的レビュー
- エ：受入れ文書の校正

答え・解説 正答：ア

ア：想定範囲を超える負荷を与えて限界や故障モードを確認するのがストレステストである。
イ：個別モジュールの正しさを確認する試験であり、システムの負荷限界確認が主目的ではない。
ウ：成果物を実行せず確認する方法であり、負荷による挙動は測定できない。
エ：文書品質の確認であり、DBの限界性能や回復挙動を評価しない。
総合解説：想定範囲を超える負荷を与えて限界や故障モードを確認するのがストレステストである。

対象：2026年度 / 基準日 2026-09-02

0096 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：標準

バックアップジョブは毎日成功しているが、これまで復元を試したことがない。可用性要件の確認として最も適切なものはどれか。

- ア：バックアップファイル名が存在すれば復旧可能とみなす。
- イ：隔離した検証環境で実際にリストア・リカバリし、復旧時間とデータ整合性を確認する。
- ウ：復旧試験をせず、障害発生時に初めて手順を作る。
- エ：本番DBを削除してから復旧試験を始める。

答え・解説 正答：イ

ア：ファイル存在だけでは破損や手順不備、必要ログ欠落を検出できない。
イ：バックアップの成功記録だけでは復元可能性を保証できず、実際の復旧試験が必要である。
ウ：復旧時間や手順の実効性を事前に保証できない。
エ：業務影響が大きく、安全な試験手順ではない。
総合解説：バックアップの成功記録だけでは復元可能性を保証できず、実際の復旧試験が必要である。

対象：2026年度 / 基準日 2026-09-02

0097 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：標準

注文表の顧客IDに外部キー制約を設定した。テストとして最も適切なものはどれか。

ア：存在する顧客IDだけで登録できることしか確認しない。

イ：外部キー制約を無効化してから試験する。

ウ：存在しない顧客IDを持つ注文の登録を試み、制約違反として拒否されることを確認する。

エ：注文表の行数だけを確認し、顧客表との関係は確認しない。

答え・解説 正答：ウ

ア：正常系は必要だが、制約が実際に不正データを防ぐかは確認できない。

イ：検証対象の制約自体を無効化しており、目的を満たさない。

ウ：正常系だけでなく制約違反データが拒否される異常系を確認すると、インテグリティ要件を検証できる。

エ：参照整合性の検証にならない。

総合解説：正常系だけでなく制約違反データが拒否される異常系を確認すると、インテグリティ要件を検証できる。

対象：2026年度 / 基準日 2026-09-02

0098 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：標準

同じSQLを繰り返すベンチマークで、初回だけ極端に遅く2回目以降が速い。原因分析として最も適切なものはどれか。

ア：初回値は必ず測定誤差なので無条件に削除する。

イ：2回目以降が速ければストレージ性能は無関係と断定する。

ウ：SQLの意味が実行ごとに自動的に変わったと判断する。

エ：バッファキャッシュや実行環境のウォームアップの影響を疑い、測定手順で初回と定常状態を区別する。

答え・解説 正答：エ

ア：初回アクセスが実運用で重要な場合もあり、要件に応じて評価すべきである。

イ：キャッシュミス時にはストレージが影響し得るため断定できない。

ウ：同一SQLの意味が自動変化することを前提にするのは不適切である。

エ：キャッシュやJIT・接続確立などの初期コストが初回値に影響するため、測定条件を明確にする必要がある。

総合解説：キャッシュやJIT・接続確立などの初期コストが初回値に影響するため、測定条件を明確にする必要がある。

対象：2026年度 / 基準日 2026-09-02

0099 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：応用

DB製品AとBを比較するベンチマークで、測定の再現性を高める組合せとして最も適切なものはどれか。

ア：データ生成手順、同時実行数、キャッシュ状態、測定時間、統計の取り方を記録し、複数回測定する。

イ：各回で同時実行数を変え、最も良い値だけ公表する。

ウ：AはSSD、Bは低速HDDを使うなど、製品ごとに有利な構成を選ぶ。

エ：一度だけ短時間測定し、結果のばらつきは考慮しない。

答え・解説 正答：ア

ア：再現可能な条件を記録し、複数回の測定でばらつきを確認することで、偶然や条件差の影響を減らせる。

イ：条件を変えた最良値だけでは比較の再現性・公平性がない。

ウ：比較対象以外の差が混入し、製品差を評価できない。

エ：一時的な負荷やキャッシュ状態の影響を受けやすく、再現性が低い。

総合解説：再現可能な条件を記録し、複数回の測定でばらつきを確認することで、偶然や条件差の影響を減らせる。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0100 4-4 テスト・移行 > DBテスト・ベンチマーク | 難易度：応用

新DBの切替え前に運用管理マニュアルを作成した。運用面のリスクを最も減らす次の対応はどれか。

ア：マニュアルを作成した時点で運用確認は完了とする。

イ：マニュアルに基づいて監視、バックアップ、障害復旧などの運用訓練を行い、手順の不足を修正する。

ウ：障害復旧手順は機密なので運用担当者にも共有しない。

エ：運用訓練は本番障害が起きた後だけ行う。

答え・解説 正答：イ

ア：記述が正しくても実行不能・不足が残る可能性がある。

イ：文書を作るだけでなく、実際の訓練で手順の実効性や担当者の理解を検証することが重要である。

ウ：実行主体が手順を知らなければ障害時の対応が遅れる。

エ：事前に問題を発見する機会を失う。

総合解説：文書を作るだけでなく、実際の訓練で手順の実効性や担当者の理解を検証することが重要である。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0101 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：基礎

データ移行計画の初期段階で最初に明確化すべき事項として最も適切なものはどれか。

ア：本番切替え後に対象テーブルを決める。

イ：対象を決めず全データを無条件でコピーする。

ウ：どのテーブル・データ期間・履歴を移行対象にするかを定める。

エ：移行先の索引名だけ先に決める。

答え・解説 正答：ウ

ア：移行設計とリハーサルができず、切替えリスクが高くなる。

イ：不要データや品質問題まで持ち込む可能性がある。

ウ：移行範囲が決まらなければ容量、変換、所要時間、検証方法を具体化できない。

エ：移行対象そのものを定義する作業より優先度は低い。

総合解説：移行範囲が決まらなければ容量、変換、所要時間、検証方法を具体化できない。

対象：2026年度 / 基準日 2026-09-02

0102 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：基礎

旧システムの「性別コード」が M/F、新システムが 1/2 を使用する。移行設計に必要な対応はどれか。

ア：値を変換せずそのまま数値列へ投入する。

イ：MとFを両方1へ変換する。

ウ：変換ルールを担当者の記憶だけに任せる。

エ：新旧項目の対応表に変換ルール M→1、F→2 を定義し、例外値の扱いも決める。

答え・解説 正答：エ

ア：型やドメイン制約に違反し、意味も失われる。

イ：元の意味を区別できなくなり、データ品質を損なう。

ウ：再現性・監査性が低く、誤変換の原因になる。

エ：コード体系が異なる場合は明示的な変換規則と例外処理が必要である。

総合解説：コード体系が異なる場合は明示的な変換規則と例外処理が必要である。

対象：2026年度 / 基準日 2026-09-02

0103 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：基礎

旧顧客データに、同一人物と思われる重複レコードと形式不正の郵便番号が多数ある。移行前の対応として最も適切なものはどれか。

ア：重複判定基準と補正ルールを定義し、クレンジング結果を検証してから移行する。

イ：すべてそのまま移行し、新システム利用者に手作業で直してもらう。

ウ：重複候補を根拠なく一律削除する。

エ：郵便番号列を削除して問題を隠す。

答え・解説 正答：ア

ア：品質問題の種類と判断基準を定め、再現可能なクレンジングを行う必要がある。

イ：品質問題を本番へ持ち込み、業務影響を拡大する。

ウ：正しい顧客まで失う可能性がある。

エ：必要な業務データを欠損させるため適切ではない。

総合解説：品質問題の種類と判断基準を定め、再現可能なクレンジングを行う必要がある。

対象：2026年度 / 基準日 2026-09-02

0104 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：基礎

外部キー制約を有効にしたまま顧客表と注文表を新DBへロードする。通常、適切な順序はどれか。

ア：注文表だけを先にロードし、顧客表は後でよい。

イ：参照される顧客表を先にロードし、その後に注文表をロードする。

ウ：順序は常に無関係である。

エ：両表の主キーを削除してからロードする。

答え・解説 正答：イ

ア：外部キーが即時検査される場合、参照先不在でエラーになる。

イ：子表の外部キーが参照する親行が先に存在すれば、参照整合性違反を避けやすい。

ウ：制約方式によってはロード順序が重要である。

エ：識別性を失わせるため、一般的な解決策ではない。

総合解説：子表の外部キーが参照する親行が先に存在すれば、参照整合性違反を避けやすい。

対象：2026年度 / 基準日 2026-09-02

0105 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

24時間稼働の旧DBから新DBへ、停止時間を短くして移行したい。初期コピー中も旧DBへの更新が続く場合の方式として最も適切なものはどれか。

ア：初期コピーだけ行い、その後の更新は無視する。

イ：旧DBへの更新を数週間止めてから移行する。

ウ：初期スナップショットを取得し、その後の変更をCDC等で捕捉して追隨し、切替え直前に差分を収束させる。

エ：差分の順序を考慮せずランダムに適用する。

答え・解説 正答：ウ

ア：コピー後の更新が欠落し、切替え時点のデータと一致しない。

イ：停止時間短縮という要件に反する。

ウ：初期ロードと継続差分を組み合わせることで、長時間停止を避けつつ最終同期を行える。

エ：更新順序や依存関係が崩れ、整合性を損なう可能性がある。

総合解説：初期ロードと継続差分を組み合わせることで、長時間停止を避けつつ最終同期を行える。

対象：2026年度 / 基準日 2026-09-02

0106 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

移行後の検証として、単純な行数比較だけでは不十分な理由はどれか。

ア：行数はデータベースでは取得できないから。

イ：行数が一致すれば必ず全列の値も一致するから。

ウ：行数比較は性能を必ず低下させるため禁止されているから。

エ：行数が同じでも、値の誤変換、重複相殺、欠損と余分な行の組合せが存在し得るから。

答え・解説 正答：エ

ア：多くのDBで行数や件数集計は可能である。

イ：同じ件数でも内容が異なるケースは容易に存在する。

ウ：検証方法として禁止される一般原則はない。

エ：件数一致は必要条件になり得るが、内容の一致までは保証しないため、キー単位・集計値・ハッシュ等の追加照合が必要である。

総合解説：件数一致は必要条件になり得るが、内容の一致までは保証しないため、キー単位・集計値・ハッシュ等の追加照合が必要である。

対象：2026年度 / 基準日 2026-09-02

0107 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

1億件の移行ETLが8千万件処理後に一時障害で停止した。再実行時の重複や全件やり直しを避けたい。設計として最も適切なものはどれか。

ア：処理済み範囲をチェックポイント等で記録し、再実行しても同じ最終状態になるよう投入処理を設計する。

イ：障害時は必ず最初からINSERTし、重複は後で目視削除する。

ウ：処理状態を一切記録しない。

エ：エラーを無視して成功扱いにする。

答え・解説 正答：ア

ア：長時間ETLでは再開点と重複防止を設計すると、障害復旧時間と誤投入リスクを減らせる。

イ：大量データでは非効率で、重複削除ミスも起きやすい。

ウ：どこまで正しく処理されたか判断できず、再開が困難になる。

エ：欠損データを残す可能性がある。

総合解説：長時間ETLでは再開点と重複防止を設計すると、障害復旧時間と誤投入リスクを減らせる。

対象：2026年度 / 基準日 2026-09-02

0108 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

大量データ移行の所要時間を見積もるとき、最も適切な考え方はどれか。

ア：データ量を無視し、担当者人数だけで決める。

イ：データ量だけでなく、実効転送帯域、変換処理能力、並列度、切替え時間窓などを含めて見積もる。

ウ：NICの公称最大速度だけを使えば実行時間は必ず正確になる。

エ：移行リハーサルは不要で、計算だけで十分である。

答え・解説 正答：イ

ア：処理量と基盤能力を反映しないため妥当な見積りにならない。

イ：理論帯域だけではなく、I/O・CPU・ネットワーク・変換処理などのボトルネックを含む実測値が重要である。

ウ：プロトコル、ストレージ、変換などのオーバーヘッドで実効速度は低下する。

エ：ターゲット環境での実測により未知の制約を把握することが重要である。

総合解説：理論帯域だけではなく、I/O・CPU・ネットワーク・変換処理などのボトルネックを含む実測値が重要である。

対象：2026年度 / 基準日 2026-09-02

0109 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

ETL中に、日付形式が壊れた少数レコードが見つかった。大量移行全体の品質と追跡性を両立する対応として最も適切なものはどれか。

ア：不正値をすべて現在日付に置換し、記録しない。

イ：エラー行を無条件に削除して件数だけ合わせる。

ウ：不正レコードを理由付きで隔離・記録し、ルールに基づき補正または再処理する。

エ：一件でも不正なら永続的に移行を中止し、原因調査もしない。

答え・解説 正答：ウ

ア：データの意味を勝手に変更し、後から検証できない。

イ：必要データの欠損を隠すことになる。

ウ：不正値を黙って捨てず、再処理可能な形で分離して原因と件数を管理することが重要である。

エ：品質問題への対応計画がなく、復旧可能性を損なう。

総合解説：不正値を黙って捨てず、再処理可能な形で分離して原因と件数を管理することが重要である。

対象：2026年度 / 基準日 2026-09-02

0110 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：標準

本番切替え直後に重大な業務不具合が発覚した場合に備え、移行計画に含める内容として最も適切なものはどれか。

ア：問題が起きたら担当者がその場で自由に判断する。

イ：新DBへ切り替えた瞬間に旧DBを削除する。

ウ：戻し手順は本番障害後に初めて検討する。

エ：戻し条件、戻し手順、切替え後に発生した更新データの扱い、判断責任者を事前に定義する。

答え・解説 正答：エ

ア：判断基準と責任が曖昧で、復旧遅延やデータ不整合につながる。

イ：ロールバック手段を失い、重大障害時の復旧が困難になる。

ウ：切替え前のリスク対策として不十分である。

エ：単に旧環境を残すだけでなく、どの時点で誰が戻すか、切替え後データをどう整合させるかまで決める必要がある。

総合解説：単に旧環境を残すだけでなく、どの時点で誰が戻すか、切替え後データをどう整合させるかまで決める必要がある。

対象：2026年度 / 基準日 2026-09-02

0111 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：応用

複数の関連テーブルを長時間かけて抽出する間も旧DBでは更新が続く。抽出データ内部の整合性を確保する方法として最も適切なものはどれか。

- ア：DBの一貫性スナップショット等を利用し、関連データを論理的に同一時点の状態として抽出する。
イ：各テーブルを任意の時刻に読み、時点差は無視する。
ウ：更新トランザクションの途中状態も含めて読み取る。
エ：主キー列だけ同時に抽出し、他列は数日後に取得する。

答え・解説 正答：ア

- ア：テーブルごとに異なる時点の状態を混在させると、親子関係や残高などの整合が崩れる可能性がある。
イ：更新が続く環境では、テーブル間の整合が崩れる可能性がある。
ウ：未コミット状態を含めれば一貫した移行元データにならない。
エ：同じ行でも時点差が生じ、属性間の整合を保証できない。
総合解説：テーブルごとに異なる時点の状態を混在させると、親子関係や残高などの整合が崩れる可能性がある。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0112 4-4 テスト・移行 > データ移行・ETL・クレンジング | 難易度：応用

複雑なデータクレンジングを伴う移行で、元データを保持しつつ変換結果を追跡可能にした。最も適切な設計はどれか。

- ア：抽出直後に元データを破棄し、変換後だけ残す。
イ：抽出データをステージング領域に保持し、変換ルール・エラー・投入結果を追跡できるようにする。
ウ：変換ルールをコードに埋め込み、版管理や記録をしない。
エ：移行結果の検証は利用者から苦情が来た場合だけ行う。

答え・解説 正答：イ

- ア：誤変換の調査や再処理が困難になる。
イ：ステージングを用いると元データと変換後データを分離し、再処理や監査を行いやすい。
ウ：再現性と監査性が低下する。
エ：事前の品質確認がなく、本番障害を招きやすい。
総合解説：ステージングを用いると元データと変換後データを分離し、再処理や監査を行いやすい。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0113 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：基礎

RDBMSのメジャーバージョンアップ前に最優先で確認すべき事項として最も適切なものはどれか。

ア：バックアップさえ取得すれば、非互換や廃止機能の事前調査は不要とする。

イ：同一RDBMS製品ならメジャーバージョン間に非互換はないと仮定する。

ウ：利用中の機能、SQL、ドライバ、拡張機能などに非互換・廃止・仕様変更がないかを調査する。

エ：本番へ直接適用し、問題が出た機能だけ後から調べる。

答え・解説 正答：ウ

ア：バックアップは復旧策であり、互換性問題を事前に除去する代わりにはならない。

イ：メジャーバージョンアップでは仕様変更や廃止機能があり得るため事前確認が必要である。

ウ：依存機能と互換性を洗い出し、テスト・改修・ロールバック計画へ反映するのが基本である。

エ：本番で初めて互換性問題を発見すると停止時間やデータ不整合のリスクが高い。

総合解説：メジャーアップデートでは互換性変更が業務影響の主要因になり得るため、利用箇所と対応策を洗い出す必要がある。

対象：2026年度 / 基準日 2026-09-02

0114 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：基礎

RDBMSを更新するとSQLの構文は同じでも性能が変わる可能性がある。主な確認対象はどれか。

ア：SQL構文が同じなら実行計画も必ず同じと判断し、計画比較を省略する。

イ：新バージョンでは統計情報が使われないと仮定し、統計収集を停止する。

ウ：論理結果が同じなら性能退行は起こらないと判断し、負荷試験を省略する。

エ：オプティマイザや統計処理の変更によってアクセス経路・実行計画が変化していないか。

答え・解説 正答：エ

ア：オプティマイザの実装やコスト推定が変われば、同じSQLでも計画が変わり得る。

イ：多くのRDBMSで統計情報は実行計画選択に重要であり、停止する根拠にならない。

ウ：結果の正しさと性能は別特性であり、性能退行は別途確認する必要がある。

エ：バージョン変更に伴うオプティマイザ・統計処理の差はアクセス経路や実行計画を変え得る。

総合解説：シラバスでもバージョンアップ時にアクセス経路変更の有無を調べるのが求められている。

対象：2026年度 / 基準日 2026-09-02

0115 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：標準

バージョンアップに伴い照合順序の仕様が変わる可能性がある。確認事項として最も適切なものはどれか。

ア：文字列の比較・並び順・一意制約・文字列索引に影響がないか代表データで検証する。

イ：数値列だけ確認すれば十分である。

ウ：照合順序は表示色だけに影響するのでDB動作には関係しない。

エ：索引を全て削除すれば互換性確認は不要になる。

答え・解説 正答：ア

ア：照合順序はORDER BYだけでなく、文字列比較や一意性、索引動作に影響し得る。

イ：文字列処理の影響を見落とす。

ウ：比較・ソート・一意性などに影響する。

エ：性能と制約の問題を新たに生み、根本的な確認にならない。

総合解説：照合順序はORDER BYだけでなく、文字列比較や一意性、索引動作に影響し得る。

対象：2026年度 / 基準日 2026-09-02

0116 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：標準

DBサーバだけを新バージョンへ更新する計画である。アプリケーション側で確認すべき事項として最も適切なものはどれか。

ア：アプリの画面配色だけ確認する。

イ：使用中のDBドライバ、接続プロトコル、認証方式、型マッピングの対応状況を確認する。

ウ：クライアント側はサーバ更新の影響を絶対に受けないと仮定する。

エ：接続試験をせず、文書だけで互換性を保証する。

答え・解説 正答：イ

ア：DB接続互換性とは関係がない。

イ：サーバが動作しても、クライアントライブラリや認証・型変換の非互換でアプリが接続・処理できない場合がある。

ウ：プロトコルやドライバ互換性の変更があり得る。

エ：実環境での統合テストが必要である。

総合解説：サーバが動作しても、クライアントライブラリや認証・型変換の非互換でアプリが接続・処理できない場合がある。

対象：2026年度 / 基準日 2026-09-02

0117 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：標準

DBにベンダ固有の拡張機能とストアードプロシージャを多数利用している。アップグレード時の対応として最も適切なものはどれか。

ア：拡張機能はDB本体と無関係なので確認不要とする。

イ：全てのストアード処理を検証せず自動的に正しいとみなす。

ウ：依存機能を棚卸しし、新版での提供状況・互換性・代替手段を確認して回帰テストする。

エ：本番で失敗した拡張だけ削除する。

答え・解説 正答：ウ

ア：DBバージョン依存やAPI変更があり得る。

イ：構文・権限・実行計画などの変更影響を見逃す。

ウ：コアDBだけでなく依存コンポーネントの互換性がシステム全体の成否を左右する。

エ：業務機能停止を招く事後対応である。

総合解説：コアDBだけでなく依存コンポーネントの互換性がシステム全体の成否を左右する。

対象：2026年度 / 基準日 2026-09-02

0118 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：標準

メジャーバージョンアップの本番計画として最も適切なものはどれか。

ア：更新開始後に初めてバックアップを検討する。

イ：新バージョン起動確認だけで旧環境を直ちに破棄する。

ウ：切戻し判断は担当者個人の感覚だけに任せる。

エ：事前バックアップと復旧手順を検証し、切替え判定基準と切戻し条件を定義しておく。

答え・解説 正答：エ

ア：問題発生前の復旧点が確保できない可能性がある。

イ：業務回帰が後から判明した際に切戻せない。

ウ：明確な判定基準がなく、判断遅延や混乱を招く。

エ：更新失敗や互換性問題に備え、戻せることを事前に確認する必要がある。

総合解説：更新失敗や互換性問題に備え、戻せることを事前に確認する必要がある。

対象：2026年度 / 基準日 2026-09-02

0119 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：応用

数千種類のSQLを使う基幹DBを更新する。性能回帰を効率よく検出する方法として最も適切なものはどれか。

ア：実運用ログ等から代表・高負荷SQLを抽出し、旧新版で実行時間・実行計画・資源使用量を比較する。

イ：最も短いSQL一つだけを比較する。

ウ：SQL性能はDBバージョンに左右されないと仮定する。

エ：新版の最速値と旧版の最遅値だけを比較する。

答え・解説 正答：ア

ア：全SQLを同じ優先度で見ると、実運用上重要なSQLをベースライン化して比較すると回帰を検出しやすい。

イ：業務負荷を代表せず、高負荷SQLの回帰を見落とす。

ウ：オプティマイザや統計仕様変更で実行計画が変わり得る。

エ：公平な条件ではなく、回帰の判定に使えない。

総合解説：全SQLを同じ優先度で見ると、実運用上重要なSQLをベースライン化して比較すると回帰を検出しやすい。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0120 4-4 テスト・移行 > バージョンアップ・変更影響 | 難易度：応用

アップグレードで新旧DBを一定期間並行稼働させ、両方に更新を許可する案が出た。最も重要な設計上の懸念はどれか。

ア：読取り専用で並行稼働する場合と同じと考え、更新競合の設計は不要とする。

イ：双方向更新の競合、同一データの更新順序、切替え時の正本判定を明確にしないと整合性が崩れること。

ウ：新旧DBの容量差だけを調整すれば、双方向更新の整合性は自動的に保証されること。

エ：バックアップ取得時刻を揃えるだけで、同時更新競合も解消できること。

答え・解説 正答：イ

ア：両系で更新を許す場合は読取り専用並行稼働と異なり、競合と同期を設計する必要がある。

イ：二重書き込みでは競合解決、順序、正本、再送などを決めなければ不整合が発生し得る。

ウ：容量調整は更新競合や順序保証を解決しない。

エ：バックアップ時刻はオンラインの更新競合制御を代替しない。

総合解説：並行書き込みは単純な複製より難しく、競合解決と権威あるデータ源の定義が必要である。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0121 4-5 分散データベース > レプリケーション・データ連携 | 難易度：基礎

論理レプリケーションの特徴として最も適切なものはどれか。

ア：常にディスクブロックをバイト単位で完全同一に複製する。

イ：データ変更を一切転送しない。

ウ：変更をテーブルや行など論理的な単位として扱い、対象を選択して複製できる方式がある。

エ：必ず異なるDB製品間でのみ利用できる。

答え・解説 正答：ウ

ア：これは物理レプリケーション側の特徴に近い。

イ：レプリケーションの目的と矛盾する。

ウ：論理レプリケーションは物理ブロックの完全複製とは異なり、論理変更を基に選択的な複製を行える。

エ：論理レプリケーションは同一製品間でも利用される。

総合解説：論理レプリケーションは物理ブロックの完全複製とは異なり、論理変更を基に選択的な複製を行える。

対象：2026年度 / 基準日 2026-09-02

0122 4-5 分散データベース > レプリケーション・データ連携 | 難易度：基礎

同期レプリケーションを非同期レプリケーションと比較した一般的な特徴として最も適切なものはどれか。

ア：必ず応答時間が短くなり、可用性も無条件に上がる。

イ：非同期方式より常に古いデータしか保持できない。

ウ：レプリケーション先に障害があっても一切影響しない。

エ：遠隔側への反映確認をコミット条件に含める構成では、データ損失リスクを下げやすい一方、遅延や可用性への影響が増え得る。

答え・解説 正答：エ

ア：同期待ちにより応答遅延や障害影響が増える場合がある。

イ：一般にはむしろ反映の確実性を高める目的で利用される。

ウ：同期方式では確認待ちのため影響を受け得る。

エ：同期確認を待つことでRPOを小さくできる反面、ネットワーク遅延や遠隔側障害の影響を受けやすい。

総合解説：同期確認を待つことでRPOを小さくできる反面、ネットワーク遅延や遠隔側障害の影響を受けやすい。

対象：2026年度 / 基準日 2026-09-02

0123 4-5 分散データベース > レプリケーション・データ連携 | 難易度：基礎

非同期レプリケーションにおける「レプリケーションラグ」とは何か。

ア：送信元で確定した変更が複製先へ適用されるまでの遅れを表す指標である。

イ：送信元と複製先のネットワーク往復時間そのものだけを表す指標である。

ウ：最新バックアップの取得時刻からの経過時間だけを表す指標である。

エ：送信元DBと複製先DBの物理ファイルサイズ差だけを表す指標である。

答え・解説 正答：ア

ア：非同期複製では確定済み変更が複製先へ反映されるまで時間差が生じ、この遅れを監視する。

イ：ネットワーク遅延は要因の一つだが、適用待ちや処理遅延も含むレプリケーションラグそのものではない。

ウ：バックアップ鮮度とレプリケーション反映遅延は別指標である。

エ：ファイルサイズ差だけでは変更反映の時間差を測れない。

総合解説：ラグが大きいほど複製先が送信元の最新状態から遅れている可能性が高い。

対象：2026年度 / 基準日 2026-09-02

0124 4-5 分散データベース > レプリケーション・データ連携 | 難易度：標準

基幹DBの更新を分析基盤へほぼリアルタイムに連携したい。全表を毎回全件再読込せずに済む方法として最も適切なものはどれか。

ア：毎秒すべてのテーブルをDROPして再作成する。

イ：トランザクションログ等からINSERT・UPDATE・DELETEの変更をCDCで捕捉して送る。

ウ：更新を検出せず、月末に手作業で入力する。

エ：SELECT権限をなくして差分を見えなくする。

答え・解説 正答：イ

ア：停止と負荷が大きく、継続連携方式として不適切である。

イ：変更分だけを抽出するCDCは、全件再ロードより効率よく継続連携できる。

ウ：ほぼリアルタイムという要件を満たさない。

エ：データ連携を実現できない。

総合解説：変更分だけを抽出するCDCは、全件再ロードより効率よく継続連携できる。

対象：2026年度 / 基準日 2026-09-02

0125 4-5 分散データベース > レプリケーション・データ連携 | 難易度：標準

本番DBの誤DELETEが即座に同期レプリカへも反映された。これが示す教訓として最も適切なものはどれか。

ア：同期レプリカがあればバックアップは不要である。

イ：レプリケーションはDELETEを絶対に複製しない。

ウ：レプリケーションは誤操作も複製するため、世代管理されたバックアップやPITRを別途用意すべきである。

エ：誤操作対策として全てのDB更新を禁止する。

答え・解説 正答：ウ

ア：誤削除や破損が複製されるため、別の復旧点が必要である。

イ：通常、変更として削除も複製される。

ウ：レプリカは可用性や読取り分散に有効だが、論理誤操作からの復旧手段を代替しない。

エ：業務を成立させない極端な対応である。

総合解説：レプリカは可用性や読取り分散に有効だが、論理誤操作からの復旧手段を代替しない。

対象：2026年度 / 基準日 2026-09-02

0126 4-5 分散データベース > レプリケーション・データ連携 | 難易度：標準

新しいレプリカを作成する際、既存データが大量にあり、初期コピー中も更新が続く。一般的な進め方として最も適切なものはどれか。

ア：初期コピー中に発生した更新は全て無視する。

イ：既存データはコピーせず差分だけ適用する。

ウ：更新順序をランダム化して適用する。

エ：一貫した初期スナップショットをコピーし、その後の変更ログを適用して追いつかせる。

答え・解説 正答：エ

ア：レプリカが永続的に欠落を持つ。

イ：過去の基礎データが存在しないため完全なレプリカにならない。

ウ：依存関係やトランザクション整合性を損なう可能性がある。

エ：初期状態とその後の差分を連続的に扱うことで、レプリカを整合した最新状態へ収束させられる。

総合解説：初期状態とその後の差分を連続的に扱うことで、レプリカを整合した最新状態へ収束させられる。

対象：2026年度 / 基準日 2026-09-02

0127 4-5 分散データベース > レプリケーション・データ連携 | 難易度：標準

二つの拠点が同一顧客レコードを同時に更新できるマルチライタ構成を採用する。設計上最も重要な追加事項はどれか。

- ア：同時更新競合を検出し、優先規則・マージ・拒否などの競合解決方針を定める。
- イ：競合は分散環境では絶対に発生しないと仮定する。
- ウ：主キーを削除して同じ顧客を別々の行として保存する。
- エ：両拠点の更新を常に無条件で上書きし、履歴も残さない。

答え・解説 正答：ア

- ア：複数ノードが同じデータを書ける場合、更新競合をどう解決するかが整合性に直結する。
 - イ：同時更新が許される以上、競合は現実的に発生し得る。
 - ウ：識別・整合性を悪化させる。
 - エ：正しい変更を失う可能性があり、監査もできない。
- 総合解説：複数ノードが同じデータを書ける場合、更新競合をどう解決するかが整合性に直結する。

対象：2026年度 / 基準日 2026-09-02

0128 4-5 分散データベース > レプリケーション・データ連携 | 難易度：標準

論理レプリケーション中のテーブルに列追加などのスキーマ変更を行う。最も適切な考え方はどれか。

- ア：DDL変更はデータ連携に一切影響しない。
- イ：送信元・受信先のスキーマ互換性と適用順序を確認し、変更データが解釈できる状態を維持する。
- ウ：受信先だけ先に任意の列を削除し、送信元は変更しない。
- エ：スキーマ不一致があってもエラーを無視すれば整合する。

答え・解説 正答：イ

- ア：論理レプリケーションではスキーマ整合が重要である。
 - イ：論理変更を適用する受信先で列定義が不整合だと、適用失敗やデータ欠落の原因になる。
 - ウ：変更適用の互換性を壊す可能性が高い。
 - エ：エラーを無視しても正しいデータにはならない。
- 総合解説：論理変更を適用する受信先で列定義が不整合だと、適用失敗やデータ欠落の原因になる。

対象：2026年度 / 基準日 2026-09-02

0129 4-5 分散データベース > レプリケーション・データ連携 | 難易度：応用

遠隔地へDBを複製する。業務要件は「災害時の許容データ損失は数秒以内」だが、拠点間ネットワーク遅延が大きい。設計判断として最も適切なものはどれか。

ア：RPOを確認せず最も安い方式を選ぶ。

イ：遠隔地なら必ず同期方式にすれば、遅延や障害影響は考慮不要である。

ウ：要求RPOとネットワーク遅延・可用性を数値で評価し、同期または十分短い遅延の非同期方式と障害時の切替え手順を選ぶ。

エ：レプリケーションを使えば切替え手順は不要である。

答え・解説 正答：ウ

ア：許容データ損失を満たせない可能性がある。

イ：同期は遅延・可用性に影響するため無条件に最適ではない。

ウ：RPOだけでなく応答時間・拠点障害の影響を含めたトレードオフ評価が必要である。

エ：障害検知・昇格・接続先変更などの運用設計が必要である。

総合解説：RPOだけでなく応答時間・拠点障害の影響を含めたトレードオフ評価が必要である。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0130 4-5 分散データベース > レプリケーション・データ連携 | 難易度：応用

複数DB間で双方向の論理レプリケーションを構成する場合、同じ変更が循環して再適用され続けるリスクを減らす考え方として最も適切なものはどれか。

ア：全変更を受信するたびに必ず全ノードへ再送する。

イ：主キーを毎回ランダムに変更する。

ウ：重複を検出せず無限に適用する。

エ：変更の起点や適用済み識別子を追跡し、受信した変更を元の経路へ再送しないよう制御する。

答え・解説 正答：エ

ア：循環や重複適用を起こしやすい。

イ：データ同一性を壊し、ループ防止の本質的解決にならない。

ウ：データ膨張や反復更新を引き起こす。

エ：双方向連携では変更の由来を識別し、ループ防止ルールを設ける必要がある。

総合解説：双方向連携では変更の由来を識別し、ループ防止ルールを設ける必要がある。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0131 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：基礎

2相コミット（2PC）におけるコーディネータの主な役割として最も適切なものはどれか。

ア：参加者に準備可否を問い合わせ、投票結果に基づいて全体のコミットまたはロールバックを決定・通知する。

イ：各参加者のデータを勝手に別値へ書き換える。

ウ：一つの参加者だけ成功すれば全体を成功扱いにする。

エ：参加者の状態を確認せずランダムにコミットする。

答え・解説 正答：ア

ア：2PCでは調停者が参加者の準備状態を集約し、全体の最終決定を伝える。

イ：業務データの内容を決める役割ではない。

ウ：分散トランザクションの原子性を損なう。

エ：2PCの準備・合意手順に反する。

総合解説：2PCでは調停者が参加者の準備状態を集約し、全体の最終決定を伝える。

対象：2026年度 / 基準日 2026-09-02

0132 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：基礎

2相コミットの第1フェーズ（prepare）で参加者が「YES」を返す意味として最も適切なものはどれか。

ア：すでに全参加者のコミットが完了したことを示す。

イ：必要な変更を永続化でき、後でコミット命令を受けたら確実にコミットできる準備が整ったことを示す。

ウ：その参加者だけは自由にロールバックしてよいことを示す。

エ：トランザクションを読み取り専用へ変換したことを示す。

答え・解説 正答：イ

ア：prepareは最終コミット前の準備段階である。

イ：YES投票後は最終決定に従えるようログや資源を保持する必要がある。

ウ：YES後はコーディネータの最終決定に従う必要がある。

エ：prepareの一般的意味ではない。

総合解説：YES投票後は最終決定に従えるようログや資源を保持する必要がある。

対象：2026年度 / 基準日 2026-09-02

0133 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：基礎

2相コミットで、一つの参加者がprepareに失敗してNOを返した。通常の全体結果として最も適切なものはどれか。

ア：NOを返した参加者だけ無視して他をコミットする。

イ：NOをYESとして扱う。

ウ：コーディネータは全体をロールバックする決定を通知する。

エ：結果を永久に未決定のままにするのが必須である。

答え・解説 正答：ウ

ア：全体原子性が崩れる。

イ：参加者がコミット不能である可能性を無視することになる。

ウ：全参加者がコミット可能であることが原子性維持の前提であり、一つでもNOなら全体を中止する。

エ：通常はNOを受けてロールバック決定が可能である。

総合解説：全参加者がコミット可能であることが原子性維持の前提であり、一つでもNOなら全体を中止する。

対象：2026年度 / 基準日 2026-09-02

0134 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：標準

全参加者がprepare済みになった直後、最終決定を通知する前にコーディネータが長時間停止した。2PCの代表的な問題はどれか。

ア：全参加者が自動的に別々の結果を選ぶので必ず整合する。

イ：prepare済みならロックは必ず即時解放される。

ウ：2PCでは障害が起きるとログを使えない。

エ：参加者が最終決定を待つ間、準備済みトランザクションのロックや資源を保持し続ける可能性がある。

答え・解説 正答：エ

ア：独立判断は原子性を壊す可能性がある。

イ：未決定の変更を保護するため資源保持が続く場合がある。

ウ：むしろ回復のために永続ログが重要である。

エ：prepared参加者は勝手に結果を変えられず、調停者の決定が分かるまでブロックすることがある。

総合解説：prepared参加者は勝手に結果を変えられず、調停者の決定が分かるまでブロックすることがある。

対象：2026年度 / 基準日 2026-09-02

0135 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：標準

分散トランザクション管理で、prepare済み状態や最終決定を永続ログへ記録する主な目的はどれか。

ア：障害再起動後にもトランザクションの状態と決定を復元し、参加者間で一貫した完了処理を行うため。

イ：ログがあるとネットワーク遅延が必ずゼロになるため。

ウ：ログを使えば参加者のデータを読み取らなくてよくなるため。

エ：ログは監査表示だけで、回復処理には使わないため。

答え・解説 正答：ア

ア：メモリ情報だけではクラッシュで状態が失われるため、回復可能なログが必要である。

イ：ログは通信遅延を消すものではない。

ウ：回復状態管理が主目的である。

エ：2PCでは状態回復に重要な役割を持つ。

総合解説：メモリ情報だけではクラッシュで状態が失われるため、回復可能なログが必要である。

対象：2026年度 / 基準日 2026-09-02

0136 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：標準

一つのRDBMS内の単一トランザクションだけで処理が完結する。外部資源も別DBも利用しない。設計判断として最も適切なものはどれか。

ア：どのトランザクションでも必ず2PCを使わなければならない。

イ：通常はRDBMS自身のローカルトランザクション機能で原子性を確保し、不要な2PCを導入しない。

ウ：単一DBではCOMMITやROLLBACKは不要である。

エ：2PCを使わないとSELECTできない。

答え・解説 正答：イ

ア：単一資源ではローカルトランザクションで十分な場合が多い。

イ：2PCは複数の独立資源を原子的に調停する際に必要となるため、単一資源では過剰になることが多い。

ウ：ローカルトランザクションでも原子性制御は必要である。

エ：SELECT可否とは直接関係しない。

総合解説：2PCは複数の独立資源を原子的に調停する際に必要となるため、単一資源では過剰になることが多い。

対象：2026年度 / 基準日 2026-09-02

0137 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：標準

クライアントがコミット要求を送った直後に通信が切れ、実際にコミットされたか分からない。再試行で二重処理を避ける設計として最も適切なものはどれか。

ア：毎回新しい取引IDを採番して無条件に再実行する。

イ：応答がない場合は必ず未コミットだったと仮定する。

ウ：取引IDなどの一意な幕等キーを持たせ、同じ業務要求の再実行を検出して重複効果を防ぐ。

エ：重複が発生しても後で利用者が気付くまで放置する。

答え・解説 正答：ウ

ア：同じ業務処理が複数回確定する可能性がある。

イ：サーバ側ではコミット済みの可能性がある。

ウ：通信障害では結果不明が起こり得るため、業務レベルで再試行安全性を持たせると二重登録を防げる。

エ：整合性・金銭処理などで重大な障害になる。

総合解説：通信障害では結果不明が起こり得るため、業務レベルで再試行安全性を持たせると二重登録を防げる。

対象：2026年度 / 基準日 2026-09-02

0138 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：標準

2PCのprepared状態を長時間放置しない方がよい主な理由はどれか。

ア：preparedになるとデータベース全体が必ず読み取り専用になるため。

イ：prepared状態ではログが一切保存されないため。

ウ：prepared中はネットワークパケットが存在できないため。

エ：ロックや資源を保持し、他トランザクションの進行や保守処理を妨げる可能性があるため。

答え・解説 正答：エ

ア：必ず全体が読み取り専用になるわけではない。

イ：むしろクラッシュ回復できるよう状態が保持される。

ウ：一般的な2PCの性質ではない。

エ：preparedは未完了のトランザクションであり、長時間化すると競合・資源占有が問題になる。

総合解説：preparedは未完了のトランザクションであり、長時間化すると競合・資源占有が問題になる。

対象：2026年度 / 基準日 2026-09-02

0139 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：応用

複数拠点のDBを2PCで一つの業務トランザクションに参加させている。拠点間ネットワークが頻繁に分断する環境で最も妥当な評価はどれか。

ア：強い原子性を得られる一方、参加者やコーディネータに到達できないと完了待ちが増え、可用性・待ち時間に影響し得る。

イ：2PCを使えばネットワーク分断の影響は完全になくなる。

ウ：2PCは原子性を弱める代わりに常に可用性を最大化する方式である。

エ：分散トランザクションでは参加者障害を考慮する必要がない。

答え・解説 正答：ア

ア：分散合意には通信が必要であり、ネットワーク不安定性は完了可能性や遅延に影響する。

イ：通信不能時の調停はできない。

ウ：目的は複数資源の原子性であり、待ちが増える場合がある。

エ：参加者障害は主要な設計課題である。

総合解説：分散合意には通信が必要であり、ネットワーク不安定性は完了可能性や遅延に影響する。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0140 4-5 分散データベース > 分散トランザクション・2相コミット | 難易度：応用

数時間続く業務フローで複数サービスをまたぎ、途中で外部承認も待つ。全時間にわたりDBロックを保持する2PCを避けたい。設計として最も適切なものはどれか。

ア：外部承認が終わるまで全DBの排他ロックを保持し続ける。

イ：各サービスでは短いローカルトランザクションを使い、失敗時は補償処理などで整合させる方式を検討する。

ウ：途中の失敗を無視し、完了した処理だけ残す。

エ：トランザクションという概念を全て削除する。

答え・解説 正答：イ

ア：長時間の資源拘束で可用性・スループットを悪化させる。

イ：長時間の分散業務では2PCで資源を拘束し続けるより、Saga等の補償型設計が適する場合がある。

ウ：業務整合性を保つ仕組みがない。

エ：各サービス内での原子性は依然として必要である。

総合解説：長時間の分散業務では2PCで資源を拘束し続けるより、Saga等の補償型設計が適する場合がある。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0141 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：基礎

BASEという考え方に最も近い説明はどれか。

ア：全処理で常に単一DBの直列実行だけを要求する考え方。

イ：永続データを一切保存しない設計原則。

ウ：分散システムで可用性や柔軟な状態遷移を重視し、結果整合性によって時間経過とともに整合へ収束させる考え方である。

エ：バックアップを作らないことで性能を上げる方式。

答え・解説 正答：ウ

ア：BASEの説明ではない。

イ：Soft stateは永続化禁止を意味しない。

ウ：BASEはBasically Available、Soft state、Eventual consistencyを中心とする設計思想として説明される。

エ：バックアップの有無とは別の概念である。

総合解説：BASEはBasically Available、Soft state、Eventual consistencyを中心とする設計思想として説明される。

対象：2026年度 / 基準日 2026-09-02

0142 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：基礎

結果整合性（eventual consistency）の説明として最も適切なものはどれか。

ア：更新直後から全ノードが必ず同じ値を返すことを保証する。

イ：ノード間の値が永久に異なっていてよい。

ウ：データを失っても整合性と呼べる。

エ：更新直後には複製間で値が異なる場合があるが、新たな更新がなければ最終的に同じ状態へ収束することを目指す。

答え・解説 正答：エ

ア：これは強い整合性に近い。

イ：収束するという要件に反する。

ウ：結果整合性は無制限なデータ損失を許容する概念ではない。

エ：結果整合性は一時的な不一致を許容しつつ、時間経過で収束するモデルである。

総合解説：結果整合性は一時的な不一致を許容しつつ、時間経過で収束するモデルである。

対象：2026年度 / 基準日 2026-09-02

0143 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：基礎

メッセージコンシューマを冪等に設計する主な目的はどれか。

- ア：同じメッセージが複数回届いても、業務上の最終効果を一回処理した場合と同じにするため。
- イ：同じメッセージを必ず毎回別取引として処理するため。
- ウ：メッセージの順序を完全にランダム化するため。
- エ：ネットワーク障害を物理的に発生しなくするため。

答え・解説 正答：ア

- ア：再配送やタイムアウト再試行がある分散環境では、重複処理による二重更新を防ぐことが重要である。
 - イ：冪等性とは逆の動作である。
 - ウ：順序制御とは別の概念である。
 - エ：冪等性は障害を防止するのではなく、再試行を安全にする。
- 総合解説：再配送やタイムアウト再試行がある分散環境では、重複処理による二重更新を防ぐことが重要である。

対象：2026年度 / 基準日 2026-09-02

0144 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：標準

at-least-once配信のメッセージ基盤を利用する。コンシューマ側の設計として最も適切なものはどれか。

- ア：一度送信したメッセージは絶対に再配送されないと仮定する。
- イ：同じメッセージが再配送される前提で、一意IDによる処理済み判定などを実装する。
- ウ：メッセージIDを毎回削除して識別不能にする。
- エ：重複時は同じ請求を何度でも登録する。

答え・解説 正答：イ

- ア：at-least-onceの性質に反する。
 - イ：少なくとも一回の配送では重複が起き得るため、重複処理への耐性が必要である。
 - ウ：重複検出が難しくなる。
 - エ：業務的な二重処理を発生させる。
- 総合解説：少なくとも一回の配送では重複が起き得るため、重複処理への耐性が必要である。

対象：2026年度 / 基準日 2026-09-02

0145 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：標準

注文DBへの更新は成功したが、その直後のメッセージ発行前にアプリが停止するとイベントが失われる問題を防ぎたい。適切なパターンはどれか。

ア：DB更新後にメモリ上だけイベントを置く。

イ：メッセージ送信だけ先に行い、DB更新成否は無視する。

ウ：業務データ更新と同じローカルトランザクションでOutbox表へイベントを記録し、別処理がOutboxから確実に配信する。

エ：障害時はイベントを永久に捨てる。

答え・解説 正答：ウ

ア：プロセス停止でイベントを失う可能性がある。

イ：イベントと業務状態が不整合になる可能性がある。

ウ：Transactional OutboxはDB更新とイベント発行要求の記録を一つのローカルトランザクションにまとめ、二重書き込み問題を避ける。

エ：下流システムとの整合性を保てない。

総合解説：Transactional OutboxはDB更新とイベント発行要求の記録を一つのローカルトランザクションにまとめ、二重書き込み問題を避ける。

対象：2026年度 / 基準日 2026-09-02

0146 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：標準

旅行予約で「航空券確保→ホテル確保→決済」を別サービスのローカルトランザクションで実行し、ホテル確保に失敗した。Saga型設計で適切な対応はどれか。

ア：航空券確保をそのまま残し、失敗を記録しない。

イ：全サービスのDBを数時間ロックし続けることだけがSagaである。

ウ：失敗したホテル予約を成功したことに書き換える。

エ：既に完了した航空券確保を取り消す補償処理を実行し、業務全体を整合した状態へ戻す。

答え・解説 正答：エ

ア：業務として中途半端な状態が残る。

イ：Sagaは長時間2PCを避ける目的で使われることが多い。

ウ：実際の資源確保と矛盾する。

エ：Sagaでは各ステップのローカル確定後に後続が失敗した場合、逆操作となる補償で業務整合性を回復する。

総合解説：Sagaでは各ステップのローカル確定後に後続が失敗した場合、逆操作となる補償で業務整合性を回復する。

対象：2026年度 / 基準日 2026-09-02

0147 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：標準

メッセージ基盤が「同じ注文IDを同じパーティションへ送れば、そのパーティション内では順序を保つ」とする。設計として最も適切なものはどれか。

- ア：同一注文のイベントを同じキーでルーティングし、注文単位の順序依存をその範囲で扱う。
- イ：全世界の全注文について単一の完全順序が自動保証されると仮定する。
- ウ：同じ注文のイベントを毎回ランダムなパーティションへ送る。
- エ：順序が重要でもイベント種別や注文IDを記録しない。

答え・解説 正答：ア

- ア：順序保証がパーティションやキーの範囲なら、業務上の順序単位をそのキーに合わせる必要がある。
 - イ：保証範囲を超えた前提である。
 - ウ：順序依存処理が逆転する可能性が高くなる。
 - エ：整合性検証や並べ替えが困難になる。
- 総合解説：順序保証がパーティションやキーの範囲なら、業務上の順序単位をそのキーに合わせる必要がある。

対象：2026年度 / 基準日 2026-09-02

0148 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：標準

形式不正で何度再試行しても処理できないメッセージが、正常メッセージの処理を妨げている。適切な対応はどれか。

- ア：無限に即時再試行し続ける。
- イ：一定回数失敗後にデッドレターキューへ隔離し、原因調査・修正・再投入を可能にする。
- ウ：エラー内容を記録せず完全に削除する。
- エ：正常メッセージも全て同時に破棄する。

答え・解説 正答：イ

- ア：CPUやキュー資源を浪費し、後続処理を妨げる。
 - イ：永続的に失敗するメッセージを本流から分離すると、全体停滞を避けつつ問題を追跡できる。
 - ウ：原因追跡や業務データ復旧ができない。
 - エ：影響範囲を不必要に広げる。
- 総合解説：永続的に失敗するメッセージを本流から分離すると、全体停滞を避けつつ問題を追跡できる。

対象：2026年度 / 基準日 2026-09-02

0149 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：応用

利用者がプロフィールを更新した直後、別レプリカから古い値を読み「更新に失敗した」と誤認する問題がある。結果整合性を維持しつつ対策する案として最も適切なものはどれか。

ア：更新直後ほど必ず最も遅延の大きいレプリカを読む。

イ：古い値を返しても常に更新失敗と表示する。

ウ：更新した利用者について一定期間は書き込み先または更新反映済みレプリカへ読み取りを誘導し、read-your-writes相当を提供する。

エ：レプリケーション遅延を測らず、問題は存在しないとする。

答え・解説 正答：ウ

ア：古い値を返す可能性を高める。

イ：実際の状態と利用者表示が矛盾する。

ウ：全体を強整合にしなくても、セッション単位の読み取り経路制御で利用者自身の更新を見せる設計が可能である。

エ：観測可能な不整合を放置する。

総合解説：全体を強整合にしなくても、セッション単位の読み取り経路制御で利用者自身の更新を見せる設計が可能である。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02

0150 4-5 分散データベース > BASE・結果整合性・メッセージ連携 | 難易度：応用

メッセージ基盤がexactly-once配信機能を提供している。業務DBへの二重書き込みを防ぐ設計について最も適切な説明はどれか。

ア：exactly-onceという名称なら、どの外部システムへの副作用も自動的に一回だけになる。

イ：一意IDを全て削除するとexactly-onceが強くなる。

ウ：確認応答が失敗しても絶対に再配送は起きない。

エ：配信側の保証範囲を確認し、業務処理側でも一意IDやトランザクション境界を設計してエンドツーエンドの重複効果を防ぐ。

答え・解説 正答：エ

ア：保証はサービスやリージョン、APIの範囲を持ち、外部副作用まで自動保証するとは限らない。

イ：重複検出や追跡が難しくなる。

ウ：配信機構や条件によって再配送・再試行があり得るため、契約を確認する必要がある。

エ：配信保証があっても、発行側の重複や外部副作用、確認応答とDB更新の境界など別要因があるため、業務処理の冪等性は依然重要である。

総合解説：配信保証があっても、発行側の重複や外部副作用、確認応答とDB更新の境界など別要因があるため、業務処理の冪等性は依然重要である。この論点では、個別技術の暗記だけでなく、要件・障害時の挙動・運用上のトレードオフまで含めて判断する。

対象：2026年度 / 基準日 2026-09-02