

0001

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：基礎

「目的起点」の演習レビューで、誤答理由として「製品名だけでは業務目的や必要要件を定義できない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：システム企画では、技術を先に決めるのではなく、解決したい業務課題、利用者、必要なデータ、期待する成果を起点に要件を整理する。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ア

ア：正しい。システム企画では、技術を先に決めるのではなく、解決したい業務課題、利用者、必要なデータ、期待する成果を起点に要件を整理する。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：システム企画では、技術を先に決めるのではなく、解決したい業務課題、利用者、必要なデータ、期待する成果を起点に要件を整理する。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0002

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：基礎

「システム企画・システム設計」の理解確認で、レビュー担当者は「表示機能の要件であり機能要件に当たる。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：機能要件は「何ができるか」、非機能要件は性能・可用性・拡張性・保守性など「どの品質で実現するか」を表す。

答え・解説

正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。機能要件は「何ができるか」、非機能要件は性能・可用性・拡張性・保守性など「どの品質で実現するか」を表す。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：機能要件は「何ができるか」、非機能要件は性能・可用性・拡張性・保守性など「どの品質で実現するか」を表す。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0003 5-1 システム・環境構築 > システム企画・システム設計 | 難易度：基礎

「パッチ対リアルタイム」について誤答分析を行ったところ、「前日までの値を毎朝確認する要件を満たさない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。
ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
ウ：処理方式は「最新であるほど良い」ではなく、必要な更新頻度、許容遅延、コストを基に選ぶ。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。処理方式は「最新であるほど良い」ではなく、必要な更新頻度、許容遅延、コストを基に選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：処理方式は「最新であるほど良い」ではなく、必要な更新頻度、許容遅延、コストを基に選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0004 5-1 システム・環境構築 > システム企画・システム設計 | 難易度：標準

「システム企画・システム設計」の解答を見直している。誤答の理由は「継続的な増加と保存期間を無視している。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。
ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
イ：容量設計では現在量だけでなく、保存期間、増加率、レプリカ、バックアップ、一時領域などを含めて将来容量を見積もる。
ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。容量設計では現在量だけでなく、保存期間、増加率、レプリカ、バックアップ、一時領域などを含めて将来容量を見積もる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：容量設計では現在量だけでなく、保存期間、増加率、レプリカ、バックアップ、一時領域などを含めて将来容量を見積もる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0005

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：標準

「可用性要件」の演習レビューで、誤答理由として「可用性は非機能要件であり設計段階から考慮する。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：可用性は曖昧な「止めない」ではなく、業務影響と復旧目標を明確化し、それに見合う冗長化・監視・復旧手段へ落とし込む。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ア

ア：正しい。可用性は曖昧な「止めない」ではなく、業務影響と復旧目標を明確化し、それに見合う冗長化・監視・復旧手段へ落とし込む。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：可用性は曖昧な「止めない」ではなく、業務影響と復旧目標を明確化し、それに見合う冗長化・監視・復旧手段へ落とし込む。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0006

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：標準

「システム企画・システム設計」の理解確認で、レビュー担当者は「負荷条件の変化を考慮する必要がある。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：PoCは技術的実現性を確認する段階であり、本番化では可用性、性能、監視、セキュリティ、運用などの要件を追加して設計する。

答え・解説

正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。PoCは技術的実現性を確認する段階であり、本番化では可用性、性能、監視、セキュリティ、運用などの要件を追加して設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：PoCは技術的実現性を確認する段階であり、本番化では可用性、性能、監視、セキュリティ、運用などの要件を追加して設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0007

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：標準

「監視設計」について誤答分析を行ったところ、「パイプラインの健全性とは直接関係しない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：監視は「動いているか」だけでなく、処理量、処理時間、遅延、エラーなど、期待したサービス水準を満たしているかを観測できるよう設計する。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：ウ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。ウ：正しい。監視は「動いているか」だけでなく、処理量、処理時間、遅延、エラーなど、期待したサービス水準を満たしているかを観測できるよう設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。総合解説：監視は「動いているか」だけでなく、処理量、処理時間、遅延、エラーなど、期待したサービス水準を満たしているかを観測できるよう設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0008

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：標準

「システム企画・システム設計」の解答を見直している。誤答の理由は「後段の一貫性が損なわれる。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：変更が想定される部分を分離し、明確なインターフェースでつなぐ疎結合な設計は、変更影響を限定しやすい。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。イ：正しい。変更が想定される部分を分離し、明確なインターフェースでつなぐ疎結合な設計は、変更影響を限定しやすい。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：変更が想定される部分を分離し、明確なインターフェースでつなぐ疎結合な設計は、変更影響を限定しやすい。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0009

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：応用

「障害復旧設計」の演習レビューで、誤答理由として「業務要件を超えた過剰設計になる場合がある。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：障害復旧では、業務が許容する復旧時間とデータ損失量を明確にし、それを満たすバックアップ・冗長化・復旧手順を設計する。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ア

ア：正しい。障害復旧では、業務が許容する復旧時間とデータ損失量を明確にし、それを満たすバックアップ・冗長化・復旧手順を設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：障害復旧では、業務が許容する復旧時間とデータ損失量を明確にし、それを満たすバックアップ・冗長化・復旧手順を設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0010

5-1 システム・環境構築 > システム企画・システム設計 | 難易度：応用

「システム企画・システム設計」の理解確認で、レビュー担当者は「要件以上の資源を常時確保すると過剰コストになり得る。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：システム設計では性能・可用性・コストを同時に満たす必要があり、利用頻度や負荷特性に合わせて資源構成を選ぶ。

答え・解説

正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。システム設計では性能・可用性・コストを同時に満たす必要があり、利用頻度や負荷特性に合わせて資源構成を選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：システム設計では性能・可用性・コストを同時に満たす必要があり、利用頻度や負荷特性に合わせて資源構成を選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0011

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：基礎

「責務分離」について誤答分析を行ったところ、「構成を分けても品質管理は必要である。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：アーキテクチャでは責務を明確に分離し、変更・障害・拡張の影響範囲を管理しやすくなることが重要である。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

0012

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：基礎

「アーキテクチャ設計」の解答を見直している。誤答の理由は「データマートは特定部門向けの分析データ領域であり、障害点の概念ではない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：高可用性設計では、1箇所の故障で全体が停止する単一障害点を特定し、必要に応じて冗長化する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。アーキテクチャでは責務を明確に分離し、変更・障害・拡張の影響範囲を管理しやすくなることが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：アーキテクチャでは責務を明確に分離し、変更・障害・拡張の影響範囲を管理しやすくなることが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。高可用性設計では、1箇所の故障で全体が停止する単一障害点を特定し、必要に応じて冗長化する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：高可用性設計では、1箇所の故障で全体が停止する単一障害点を特定し、必要に応じて冗長化する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0013

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：基礎

「負荷分散」の演習レビューで、誤答理由として「これはデータ変換処理である。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：複数の処理ノードを活用するには、要求を適切に振り分ける仕組みが必要であり、ロードバランサは代表的な負荷分散手段である。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ア

ア：正しい。複数の処理ノードを活用するには、要求を適切に振り分ける仕組みが必要であり、ロードバランサは代表的な負荷分散手段である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：複数の処理ノードを活用するには、要求を適切に振り分ける仕組みが必要であり、ロードバランサは代表的な負荷分散手段である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0014

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：標準

「アーキテクチャ設計」の理解確認で、レビュー担当者は「複数台を活用できない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：ステートレスな処理層は任意のインスタンスで要求を処理しやすく、水平スケーリングや障害時の切替に適している。

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。ステートレスな処理層は任意のインスタンスで要求を処理しやすく、水平スケーリングや障害時の切替に適している。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：ステートレスな処理層は任意のインスタンスで要求を処理しやすく、水平スケーリングや障害時の切替に適している。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0015

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：標準

「非同期化」について誤答分析を行ったところ、「結合が強まり、独立した拡張が難しくなる。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：非同期メッセージングは、送信側と受信側の処理速度差を吸収し、独立したスケーリングや障害分離をしやすいとする。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

0016

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：標準

「アーキテクチャ設計」の解答を見直している。誤答の理由は「元データとキャッシュの役割は異なる。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：キャッシュは性能改善に有効だが、更新タイミングや有効期限など、元データとの整合性を保つ仕組みが必要である。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。非同期メッセージングは、送信側と受信側の処理速度差を吸収し、独立したスケーリングや障害分離をしやすいとする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：非同期メッセージングは、送信側と受信側の処理速度差を吸収し、独立したスケーリングや障害分離をしやすいとする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。キャッシュは性能改善に有効だが、更新タイミングや有効期限など、元データとの整合性を保つ仕組みが必要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：キャッシュは性能改善に有効だが、更新タイミングや有効期限など、元データとの整合性を保つ仕組みが必要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0017

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：標準

「API変更」の演習レビューで、誤答理由として「利用者が多いほど明確な契約と変更管理が重要である。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。
ア：共有APIはシステム間の契約であるため、破壊的変更を避け、バージョン管理や移行期間を設けて変更影響を制御する。
イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ア

ア：正しい。共有APIはシステム間の契約であるため、破壊的変更を避け、バージョン管理や移行期間を設けて変更影響を制御する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：共有APIはシステム間の契約であるため、破壊的変更を避け、バージョン管理や移行期間を設けて変更影響を制御する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0018

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：標準

「アーキテクチャ設計」の理解確認で、レビュー担当者は「分布の偏りを無視するとホットスポットが生じ得る。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。
ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
エ：分散設計では「分ける」だけでなく、実際のデータ分布とアクセスパターンを見て偏りを抑えることが重要である。

答え・解説

正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。分散設計では「分ける」だけでなく、実際のデータ分布とアクセスパターンを見て偏りを抑えることが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：分散設計では「分ける」だけでなく、実際のデータ分布とアクセスパターンを見て偏りを抑えることが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0019

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：応用

「障害分離」について誤答分析を行ったところ、「障害の影響範囲が拡大しやすい。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：重要度や負荷特性の異なる処理を分離し、障害を局所化できるようにすることは、信頼性の高いアーキテクチャの基本である。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。重要度や負荷特性の異なる処理を分離し、障害を局所化できるようにすることは、信頼性の高いアーキテクチャの基本である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：重要度や負荷特性の異なる処理を分離し、障害を局所化できるようにすることは、信頼性の高いアーキテクチャの基本である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0020

5-1 システム・環境構築 > アーキテクチャ設計 | 難易度：応用

「アーキテクチャ設計」の解答を見直している。誤答の理由は「要件との適合や運用可能性を評価する必要がある。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：アーキテクチャはビジネス・データ・非機能要件と制約を起点に比較し、トレードオフを明示して選ぶ。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。アーキテクチャはビジネス・データ・非機能要件と制約を起点に比較し、トレードオフを明示して選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：アーキテクチャはビジネス・データ・非機能要件と制約を起点に比較し、トレードオフを明示して選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0021 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：基礎

「クラウド特性」の演習レビューで、誤答理由として「NISTはMeasured Serviceも基本特性としている。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。
ア：クラウドは、共有資源をネットワーク経由で必要時に迅速に利用・解放できるモデルとして定義される。
イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ア

ア：正しい。クラウドは、共有資源をネットワーク経由で必要時に迅速に利用・解放できるモデルとして定義される。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：クラウドは、共有資源をネットワーク経由で必要時に迅速に利用・解放できるモデルとして定義される。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0022 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：基礎

「クラウド・分散処理環境」の理解確認で、レビュー担当者は「SaaSは提供者のアプリケーションを利用する形で、基盤管理は通常利用者の役割ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。
ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
エ：IaaS・PaaS・SaaSは、利用者がどこまで管理するかという責任範囲の違いを理解して区別する。

答え・解説 正答：エ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。IaaS・PaaS・SaaSは、利用者がどこまで管理するかという責任範囲の違いを理解して区別する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：IaaS・PaaS・SaaSは、利用者がどこまで管理するかという責任範囲の違いを理解して区別する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0023 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：基礎

「スケーリング」について誤答分析を行ったところ、「正規化はデータ表現やDB設計などで使われる概念である。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：スケールアウトはノード数を増やす方式、スケールアップは1ノードの資源を増強する方式である。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。スケールアウトはノード数を増やす方式、スケールアップは1ノードの資源を増強する方式である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：スケールアウトはノード数を増やす方式、スケールアップは1ノードの資源を増強する方式である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0024 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：標準

「クラウド・分散処理環境」の解答を見直している。誤答の理由は「クラウドの弾力的な資源調整と反する。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：負荷変動が大きい処理では、必要時に増やし不要時に減らせる弾力性が、性能とコストの両立に役立つ。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。負荷変動が大きい処理では、必要時に増やし不要時に減らせる弾力性が、性能とコストの両立に役立つ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：負荷変動が大きい処理では、必要時に増やし不要時に減らせる弾力性が、性能とコストの両立に役立つ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0025 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：標準

「データ分割」の演習レビューで、誤答理由として「単一ノードの制約を受け、並列性を活かせない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。
ア：分散処理は、データやタスクを複数ノードへ分割し、独立に処理できる部分を並列化することで大規模処理へ対応する。
イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：ア

ア：正しい。分散処理は、データやタスクを複数ノードへ分割し、独立に処理できる部分を並列化することで大規模処理へ対応する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：分散処理は、データやタスクを複数ノードへ分割し、独立に処理できる部分を並列化することで大規模処理へ対応する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0026 5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：標準

「クラウド・分散処理環境」の理解確認で、レビュー担当者は「不要なデータ転送が増える。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。
ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
エ：分散処理では、データ移動はコストが大きいいため、データ局所性を意識して計算を配置する考え方が重要である。

答え・解説 正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。分散処理では、データ移動はコストが大きいいため、データ局所性を意識して計算を配置する考え方が重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：分散処理では、データ移動はコストが大きいいため、データ局所性を意識して計算を配置する考え方が重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0027

5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：標準

「レプリケーション」について誤答分析を行ったところ、「SQLの複雑さを直接減らすものではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：分散ストレージではハードウェア故障を前提とし、複製により一部ノード障害時もデータを利用できるようにする。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説

正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。分散ストレージではハードウェア故障を前提とし、複製により一部ノード障害時もデータを利用できるようにする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分散ストレージではハードウェア故障を前提とし、複製により一部ノード障害時もデータを利用できるようにする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0028

5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：標準

「クラウド・分散処理環境」の解答を見直している。誤答の理由は「分散集計のデータ移動性能とは関係しない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：分散処理では計算量だけでなく、ノード間のデータ移動やI/Oがボトルネックになるため、シャッフル量を意識する必要がある。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。分散処理では計算量だけでなく、ノード間のデータ移動やI/Oがボトルネックになるため、シャッフル量を意識する必要がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分散処理では計算量だけでなく、ノード間のデータ移動やI/Oがボトルネックになるため、シャッフル量を意識する必要がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0029

5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：応用

「データスキュー」の演習レビューで、誤答理由として「型の一致だけではこの偏りを説明できない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：並列処理の速度は最も遅いタスクに引っ張られるため、パーティションのデータ量やキー頻度の偏りを確認する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ア

ア：正しい。並列処理の速度は最も遅いタスクに引っ張られるため、パーティションのデータ量やキー頻度の偏りを確認する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：並列処理の速度は最も遅いタスクに引っ張られるため、パーティションのデータ量やキー頻度の偏りを確認する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0030

5-1 システム・環境構築 > クラウド・分散処理環境 | 難易度：応用

「クラウド・分散処理環境」の理解確認で、レビュー担当者は「並列度がさらに下がる可能性が高い。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：スケールアウトの効果は、並列化可能性、通信量、共有資源、データ偏りなどに制約される。実測してボトルネックを特定することが重要である。

答え・解説

正答：エ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。スケールアウトの効果は、並列化可能性、通信量、共有資源、データ偏りなどに制約される。実測してボトルネックを特定することが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：スケールアウトの効果は、並列化可能性、通信量、共有資源、データ偏りなどに制約される。実測してボトルネックを特定することが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0031 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：基礎

「API/SDK識別」について誤答分析を行ったところ、「APIは物理機器に限定されない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：データ収集では、外部サービスが提供するAPIやSDKの役割を理解し、認証・形式・制限などの仕様に沿って利用する。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。データ収集では、外部サービスが提供するAPIやSDKの役割を理解し、認証・形式・制限などの仕様に沿って利用する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ収集では、外部サービスが提供するAPIやSDKの役割を理解し、認証・形式・制限などの仕様に沿って利用する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0032 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：基礎

「クライアント・通信技術」の解答を見直している。誤答の理由は「PATCHは部分更新に使われ、safe methodではない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：HTTPメソッドには意味があり、GETなどのsafe methodは読み取りを意図する。API利用時はメソッドの意味を踏まえる。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：イ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。HTTPメソッドには意味があり、GETなどのsafe methodは読み取りを意図する。API利用時はメソッドの意味を踏まえる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：HTTPメソッドには意味があり、GETなどのsafe methodは読み取りを意図する。API利用時はメソッドの意味を踏まえる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0033

5-2 データ収集・構造 > クライアント・通信技術 | 難易度：基礎

「データ交換形式」の演習レビューで、誤答理由として「SQLの実行計画を定義するものではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：通信では、プロトコルとデータ表現を区別する。HTTP上でJSONを送受信する構成は代表的な組合せである。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ア

ア：正しい。通信では、プロトコルとデータ表現を区別する。HTTP上でJSONを送受信する構成は代表的な組合せである。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。 この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：通信では、プロトコルとデータ表現を区別する。HTTP上でJSONを送受信する構成は代表的な組合せである。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0034

5-2 データ収集・構造 > クライアント・通信技術 | 難易度：標準

「クライアント・通信技術」の理解確認で、レビュー担当者は「冪等性は遅延の大小を表す概念ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：再試行では、通信失敗前に処理が実行済みか不明な場合がある。冪等性や重複防止キーを考慮して二重処理を防ぐ。

答え・解説

正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。再試行では、通信失敗前に処理が実行済みか不明な場合がある。冪等性や重複防止キーを考慮して二重処理を防ぐ。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：再試行では、通信失敗前に処理が実行済みか不明な場合がある。冪等性や重複防止キーを考慮して二重処理を防ぐ。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0035 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：標準

「重複送信」について誤答分析を行ったところ、「重複作成の問題とは関係しない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：作成系APIの再試行では、処理済み確認やidempotency keyなど、重複適用を防ぐ設計が重要である。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：ウ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。作成系APIの再試行では、処理済み確認やidempotency keyなど、重複適用を防ぐ設計が重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：作成系APIの再試行では、処理済み確認やidempotency keyなど、重複適用を防ぐ設計が重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0036 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：標準

「クライアント・通信技術」の解答を見直している。誤答の理由は「即時性要件を満たしにくい。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：更新通知が利用できる場合はイベント駆動で受け取り、ポーリング回数を抑える選択肢がある。要件と相手の仕様で使い分ける。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。更新通知が利用できる場合はイベント駆動で受け取り、ポーリング回数を抑える選択肢がある。要件と相手の仕様で使い分ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：更新通知が利用できる場合はイベント駆動で受け取り、ポーリング回数を抑える選択肢がある。要件と相手の仕様で使い分ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0037 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：標準

「レート制限」の演習レビューで、誤答理由として「制限違反を悪化させる可能性がある。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：外部APIはレート制限、ページング、認証、再試行条件などの仕様を守り、安定的に収集できるよう制御する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：ア

ア：正しい。外部APIはレート制限、ページング、認証、再試行条件などの仕様を守り、安定的に収集できるよう制御する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部APIはレート制限、ページング、認証、再試行条件などの仕様を守り、安定的に収集できるよう制御する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0038 5-2 データ収集・構造 > クライアント・通信技術 | 難易度：標準

「クライアント・通信技術」の理解確認で、レビュー担当者は「提供APIの仕様外である。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：API収集では、レスポンス件数上限とページング方式を確認し、取りこぼしや重複がないように全ページを処理する。

答え・解説 正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。API収集では、レスポンス件数上限とページング方式を確認し、取りこぼしや重複がないように全ページを処理する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：API収集では、レスポンス件数上限とページング方式を確認し、取りこぼしや重複がないように全ページを処理する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0039

5-2 データ収集・構造 > クライアント・通信技術 | 難易度：応用

「通信性能」について誤答分析を行ったところ、「高いCPUクロックが直列通信待ちを引き起こすわけではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：通信性能では、帯域だけでなくレイテンシ、要求回数、並列度、ペイロードサイズを分けて考える。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。通信性能では、帯域だけでなくレイテンシ、要求回数、並列度、ペイロードサイズを分けて考える。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：通信性能では、帯域だけでなくレイテンシ、要求回数、並列度、ペイロードサイズを分けて考える。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0040

5-2 データ収集・構造 > クライアント・通信技術 | 難易度：応用

「クライアント・通信技術」の解答を見直している。誤答の理由は「データ欠損を見逃す。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：外部通信は失敗を前提に、タイムアウト、上限付き再試行、バックオフ、重複防止、失敗記録を組み合わせで設計する。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：イ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。外部通信は失敗を前提に、タイムアウト、上限付き再試行、バックオフ、重複防止、失敗記録を組み合わせで設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部通信は失敗を前提に、タイムアウト、上限付き再試行、バックオフ、重複防止、失敗記録を組み合わせで設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0041

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：基礎

「収集方式」の演習レビューで、誤答理由として「データを類似グループに分ける分析手法である。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：データ収集方式は、許容遅延やデータ量に応じてバッチ、ストリーミングなどを使い分ける。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

0042

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：基礎

「データ抽出・データ収集」の理解確認で、レビュー担当者は「実測データに基づく異常検知にならない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：リアルタイム性が必要なデータでは、イベントを継続的に取り込む方式を検討する。

答え・解説

正答：ア

ア：正しい。データ収集方式は、許容遅延やデータ量に応じてバッチ、ストリーミングなどを使い分ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ収集方式は、許容遅延やデータ量に応じてバッチ、ストリーミングなどを使い分ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：エ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。リアルタイム性が必要なデータでは、イベントを継続的に取り込む方式を検討する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：リアルタイム性が必要なデータでは、イベントを継続的に取り込む方式を検討する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0043

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：基礎

「ETL識別」について誤答分析を行ったところ、「ETLの抽出処理ではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

- ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
- イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
- ウ：ETLはExtract（抽出）、Transform（変換）、Load（ロード）の流れでデータを統合・準備する。
- エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。ETLはExtract（抽出）、Transform（変換）、Load（ロード）の流れでデータを統合・準備する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：ETLはExtract（抽出）、Transform（変換）、Load（ロード）の流れでデータを統合・準備する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0044

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：標準

「データ抽出・データ収集」の解答を見直している。誤答の理由は「実データとの不整合が生じる。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

- ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
- イ：大量データの定期連携では、変更分を識別できるなら増分抽出を使うことで負荷を抑えやすい。
- ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
- エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。大量データの定期連携では、変更分を識別できるなら増分抽出を使うことで負荷を抑えやすい。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：大量データの定期連携では、変更分を識別できるなら増分抽出を使うことで負荷を抑えやすい。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0045 5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：標準

「変更データ取得」の演習レビューで、誤答理由として「カテゴリ変数の数値表現方法である。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：CDCは全件再取得を避け、変更イベントを継続的または増分で連携する際に利用される。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ア

ア：正しい。CDCは全件再取得を避け、変更イベントを継続的または増分で連携する際に利用される。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：CDCは全件再取得を避け、変更イベントを継続的または増分で連携する際に利用される。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻るかを確認する。

対象：2026年度 / 基準日 2026-09-09

0046 5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：標準

「データ抽出・データ収集」の理解確認で、レビュー担当者は「業務負荷をさらに悪化させる可能性がある。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：データ収集は取得側だけでなくソースシステムへの影響も考え、時間帯・差分取得・負荷分離を設計する。

答え・解説 正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。データ収集は取得側だけでなくソースシステムへの影響も考え、時間帯・差分取得・負荷分離を設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：データ収集は取得側だけでなくソースシステムへの影響も考え、時間帯・差分取得・負荷分離を設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻るかを確認する。

対象：2026年度 / 基準日 2026-09-09

0047

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：標準

「スキーマ変更」について誤答分析を行ったところ、「データ欠損や誤解釈を見逃す。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：収集パイプラインでは、型・列名・必須項目などのスキーマを検証し、変更時の互換性やエラー処理を決めておく。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ウ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。収集パイプラインでは、型・列名・必須項目などのスキーマを検証し、変更時の互換性やエラー処理を決めておく。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：収集パイプラインでは、型・列名・必須項目などのスキーマを検証し、変更時の互換性やエラー処理を決めておく。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0048

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：標準

「データ抽出・データ収集」の解答を見直している。誤答の理由は「通信障害や再試行を考慮していない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：収集では欠落だけでなく重複も品質問題になる。識別子や処理履歴を用いて冪等な取り込みを設計する。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説

正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。収集では欠落だけでなく重複も品質問題になる。識別子や処理履歴を用いて冪等な取り込みを設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：収集では欠落だけでなく重複も品質問題になる。識別子や処理履歴を用いて冪等な取り込みを設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0049

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：応用

「時刻整合」の演習レビューで、誤答理由として「時系列分析や順序確認ができなくなる。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：分散したデータ源では、イベント時刻・受信時刻・タイムゾーンを区別し、比較可能な時刻表現へ統一する。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

0050

5-2 データ収集・構造 > データ抽出・データ収集 | 難易度：応用

「データ抽出・データ収集」の理解確認で、レビュー担当者は「追跡性が低下する。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：データ収集では、値だけでなく取得元・時刻・バージョンなどのメタデータを残し、データの来歴を追えるようにすることが品質管理に有効である。

答え・解説

正答：ア

ア：正しい。分散したデータ源では、イベント時刻・受信時刻・タイムゾーンを区別し、比較可能な時刻表現へ統一する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分散したデータ源では、イベント時刻・受信時刻・タイムゾーンを区別し、比較可能な時刻表現へ統一する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

答え・解説

正答：エ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。データ収集では、値だけでなく取得元・時刻・バージョンなどのメタデータを残し、データの来歴を追えるようにすることが品質管理に有効である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：データ収集では、値だけでなく取得元・時刻・バージョンなどのメタデータを残し、データの来歴を追えるようにすることが品質管理に有効である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0051

5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：基礎

「表構造」について誤答分析を行ったところ、「列ごとに適切な型を持てる。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
ウ：テーブル設計では、1行が何を表すかという粒度と、各列の意味・型を明確にする。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。テーブル設計では、1行が何を表すかという粒度と、各列の意味・型を明確にする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：テーブル設計では、1行が何を表すかという粒度と、各列の意味・型を明確にする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0052

5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：基礎

「データ構造・テーブル定義」の解答を見直している。誤答の理由は「一意識別に反する。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
イ：主キーはレコードを一意に特定するための基本的なキーであり、他テーブルから参照される基準にもなる。
ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：イ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。主キーはレコードを一意に特定するための基本的なキーであり、他テーブルから参照される基準にもなる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：主キーはレコードを一意に特定するための基本的なキーであり、他テーブルから参照される基準にもなる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0053 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：基礎

「外部キー」の演習レビューで、誤答理由として「高速参照のための一時保存でありキー制約ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：外部キーはテーブル間の参照関係を定義し、存在しない親レコードへの参照などを防ぐために利用できる。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ア

ア：正しい。外部キーはテーブル間の参照関係を定義し、存在しない親レコードへの参照などを防ぐために利用できる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部キーはテーブル間の参照関係を定義し、存在しない親レコードへの参照などを防ぐために利用できる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0054 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：標準

「データ構造・テーブル定義」の理解確認で、レビュー担当者は「型を変えるだけでは欠損・重複を禁止できない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：テーブル制約はデータ品質をDBレベルで守る手段であり、欠損禁止と一意性は別の制約として考える。

答え・解説 正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。テーブル制約はデータ品質をDBレベルで守る手段であり、欠損禁止と一意性は別の制約として考える。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：テーブル制約はデータ品質をDBレベルで守る手段であり、欠損禁止と一意性は別の制約として考える。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0055 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：標準

「関係設計」について誤答分析を行ったところ、「注文数の上限が固定され、拡張性や扱いやすさが低い。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：繰り返し発生する明細は別テーブルにし、外部キーで親レコードへ関連付けるのが基本的な関係設計である。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：ウ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。繰り返し発生する明細は別テーブルにし、外部キーで親レコードへ関連付けるのが基本的な関係設計である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：繰り返し発生する明細は別テーブルにし、外部キーで親レコードへ関連付けるのが基本的な関係設計である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0056 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：標準

「データ構造・テーブル定義」の解答を見直している。誤答の理由は「顧客と注文を結び付けられなくなる。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：正規化はデータの重複や更新時の不整合を減らし、事実を適切な単位で管理するための基本的な設計手法である。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。正規化はデータの重複や更新時の不整合を減らし、事実を適切な単位で管理するための基本的な設計手法である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：正規化はデータの重複や更新時の不整合を減らし、事実を適切な単位で管理するための基本的な設計手法である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0057 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：標準

「粒度定義」の演習レビューで、誤答理由として「データの観測単位を表す概念である。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：分析テーブルでは「1行=何」を明示し、主キーや集計単位、JOIN時の多重化を一貫して設計する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ア

ア：正しい。分析テーブルでは「1行=何」を明示し、主キーや集計単位、JOIN時の多重化を一貫して設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分析テーブルでは「1行=何」を明示し、主キーや集計単位、JOIN時の多重化を一貫して設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0058 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：標準

「データ構造・テーブル定義」の理解確認で、レビュー担当者は「表記揺れが増える。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：データ型は保存形式だけでなく、妥当性・演算・比較の一貫性に関係するため、意味に合う型を選ぶ。

答え・解説 正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。データ型は保存形式だけでなく、妥当性・演算・比較の一貫性に関係するため、意味に合う型を選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：データ型は保存形式だけでなく、妥当性・演算・比較の一貫性に関係するため、意味に合う型を選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0059 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：応用

「複合キー」について誤答分析を行ったところ、「同じ商品が複数注文に現れるため通常一意ではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：一意性の単位が複数属性の組合せで決まる場合、複合主キーでその業務上の一意性を表現できる。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。一意性の単位が複数属性の組合せで決まる場合、複合主キーでその業務上の一意性を表現できる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：一意性の単位が複数属性の組合せで決まる場合、複合主キーでその業務上の一意性を表現できる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0060 5-2 データ収集・構造 > データ構造・テーブル定義 | 難易度：応用

「データ構造・テーブル定義」の解答を見直している。誤答の理由は「キー重複の解消にはならない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：JOIN前には、結合キーの一意性と各テーブルの粒度を確認する。想定外の多対多結合は件数や集計値を膨らませる。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：イ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。JOIN前には、結合キーの一意性と各テーブルの粒度を確認する。想定外の多対多結合は件数や集計値を膨らませる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：JOIN前には、結合キーの一意性と各テーブルの粒度を確認する。想定外の多対多結合は件数や集計値を膨らませる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0061 5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：基礎

「DWH定義」の演習レビューで、誤答理由として「DWHの主目的ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

- ア：DWHは分析・レポーティングのために、複数ソースのデータを統合して蓄積する基盤である。
- イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
- ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
- エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：ア

ア：正しい。DWHは分析・レポーティングのために、複数ソースのデータを統合して蓄積する基盤である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：DWHは分析・レポーティングのために、複数ソースのデータを統合して蓄積する基盤である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0062 5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：基礎

「DWH・データ蓄積技術」の理解確認で、レビュー担当者は「注文DBは業務トランザクションを扱う。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

- ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。
- イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
- ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
- エ：分析処理を業務トランザクションDBから分離すると、双方の用途に合った性能・構造を設計しやすい。

答え・解説 正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。分析処理を業務トランザクションDBから分離すると、双方の用途に合った性能・構造を設計しやすい。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：分析処理を業務トランザクションDBから分離すると、双方の用途に合った性能・構造を設計しやすい。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0063

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：基礎

「データマート」について誤答分析を行ったところ、「テーブル行を一意に識別するキーである。」という問題点
が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
ウ：データマートは、全社的なDWHに対して特定部門・業務テーマへ焦点を絞った分析用データ集合として使われ
る。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。データマートは、全社的なDWHに対して特定部門・業務テーマへ焦点を絞った分析用データ集合として使われる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データマートは、全社的なDWHに対して特定部門・業務テーマへ焦点を絞った分析用データ集合として使われる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0064

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：標準

「DWH・データ蓄積技術」の解答を見直している。誤答の理由は「5年間の比較ができなくなる。」であった。再
発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
イ：DWHでは履歴分析が重要な用途の一つであり、時間軸や変更履歴を失わない設計が求められる。
ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。DWHでは履歴分析が重要な用途の一つであり、時間軸や変更履歴を失わない設計が求められる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：DWHでは履歴分析が重要な用途の一つであり、時間軸や変更履歴を失わない設計が求められる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0065

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：標準

「スタースキーマ」の演習レビューで、誤答理由として「売上金額や数量などの測定値はファクトとして扱う。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：分析モデルでは、測定対象の事実と、商品・顧客・日付などの分析軸を分けると集計しやすい。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ア

ア：正しい。分析モデルでは、測定対象の事実と、商品・顧客・日付などの分析軸を分けると集計しやすい。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分析モデルでは、測定対象の事実と、商品・顧客・日付などの分析軸を分けると集計しやすい。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0066

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：標準

「DWH・データ蓄積技術」の理解確認で、レビュー担当者は「用途やデータ構造、処理方法が異なる。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：DWHとデータレイクは対立概念ではなく、用途に応じて組み合わせる。DWHは分析用に整理された表データ、レイクは多様な原データの保持に強みがある。

答え・解説

正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。DWHとデータレイクは対立概念ではなく、用途に応じて組み合わせる。DWHは分析用に整理された表データ、レイクは多様な原データの保持に強みがある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：DWHとデータレイクは対立概念ではなく、用途に応じて組み合わせる。DWHは分析用に整理された表データ、レイクは多様な原データの保持に強みがある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0067

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：標準

「ETL/ELT」について誤答分析を行ったところ、「EはExtractを意味する。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
ウ：ETL/ELTは処理順序と変換場所の設計パターンであり、対象基盤の処理能力やデータ量、ガバナンスに応じて選ぶ。
エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

0068

5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：標準

「DWH・データ蓄積技術」の解答を見直している。誤答の理由は「業務要件に反する可能性が高い。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
イ：DWHは分析処理を業務DBから分離する役割も持ち、異なるワークロードをそれぞれに適した基盤で処理できる。
ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ウ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。ETL/ELTは処理順序と変換場所の設計パターンであり、対象基盤の処理能力やデータ量、ガバナンスに応じて選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：ETL/ELTは処理順序と変換場所の設計パターンであり、対象基盤の処理能力やデータ量、ガバナンスに応じて選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。DWHは分析処理を業務DBから分離する役割も持ち、異なるワークロードをそれぞれに適した基盤で処理できる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：DWHは分析処理を業務DBから分離する役割も持ち、異なるワークロードをそれぞれに適した基盤で処理できる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0069 5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：応用

「統合品質」の演習レビューで、誤答理由として「識別不整合は解消しない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：DWH統合では、項目名だけでなくコード体系、単位、粒度、意味をそろえ、同じ事実を一貫して分析できるようにする。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：ア

ア：正しい。DWH統合では、項目名だけでなくコード体系、単位、粒度、意味をそろえ、同じ事実を一貫して分析できるようにする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：DWH統合では、項目名だけでなくコード体系、単位、粒度、意味をそろえ、同じ事実を一貫して分析できるようにする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0070 5-3 データ蓄積・加工 > DWH・データ蓄積技術 | 難易度：応用

「DWH・データ蓄積技術」の理解確認で、レビュー担当者は「要求を満たさない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：DWHの更新方式は利用者が必要とする鮮度、ソース負荷、処理コストを踏まえて設計する。

答え・解説 正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。DWHの更新方式は利用者が必要とする鮮度、ソース負荷、処理コストを踏まえて設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：DWHの更新方式は利用者が必要とする鮮度、ソース負荷、処理コストを踏まえて設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0071 5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：基礎

「HDFS構成」について誤答分析を行ったところ、「HDFSのファイルシステムメタデータ管理役ではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。
ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。
イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
ウ：HDFSではNameNodeがメタデータ、DataNodeが実データブロックを主に担当するという役割分担を理解する。
エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。HDFSではNameNodeがメタデータ、DataNodeが実データブロックを主に担当するという役割分担を理解する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：HDFSではNameNodeがメタデータ、DataNodeが実データブロックを主に担当するという役割分担を理解する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0072 5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：基礎

「分散技術・データ基盤」の解答を見直している。誤答の理由は「分析手法ではなくストレージ構成要素である。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。
ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：HDFSはファイルをブロックへ分割して複数DataNodeへ保存し、NameNodeがメタデータを管理する。
ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。HDFSはファイルをブロックへ分割して複数DataNodeへ保存し、NameNodeがメタデータを管理する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：HDFSはファイルをブロックへ分割して複数DataNodeへ保存し、NameNodeがメタデータを管理する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0073

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：基礎

「RDD基礎」の演習レビューで、誤答理由として「分散データ抽象ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

- ア：SparkではRDDなどの分散データをパーティション単位で複数ノードへ割り当て、並列計算を行う。
- イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
- ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
- エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ア

ア：正しい。SparkではRDDなどの分散データをパーティション単位で複数ノードへ割り当て、並列計算を行う。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：SparkではRDDなどの分散データをパーティション単位で複数ノードへ割り当て、並列計算を行う。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0074

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：標準

「分散技術・データ基盤」の理解確認で、レビュー担当者は「HDFSには自動再複製の仕組みがある。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

- ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
- イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
- ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
- エ：分散ストレージは故障を前提に複製と障害検知を組み込み、必要な冗長度へ回復させる。

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。分散ストレージは故障を前提に複製と障害検知を組み込み、必要な冗長度へ回復させる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：分散ストレージは故障を前提に複製と障害検知を組み込み、必要な冗長度へ回復させる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0075

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：標準

「Spark操作」について誤答分析を行ったところ、「Sparkのtransformationsは原則lazyである。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：Sparkではtransformationが新しい分散データの処理系を定義し、actionが結果を要求して実計算を起動する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説

正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。Sparkではtransformationが新しい分散データの処理系を定義し、actionが結果を要求して実計算を起動する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：Sparkではtransformationが新しい分散データの処理系を定義し、actionが結果を要求して実計算を起動する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0076

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：標準

「分散技術・データ基盤」の解答を見直している。誤答の理由は「むしろ即時計算ではないことを表す。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：遅延評価により、Sparkは必要な処理をまとめて実行計画へ反映できる。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。遅延評価により、Sparkは必要な処理をまとめて実行計画へ反映できる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：遅延評価により、Sparkは必要な処理をまとめて実行計画へ反映できる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0077

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：標準

「driver負荷」の演習レビューで、誤答理由として「型変換処理ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：分散処理では、大規模データを不用意に1台へ集約しない。最終結果を小さく集約してからdriverへ返す設計が重要である。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ア

ア：正しい。分散処理では、大規模データを不用意に1台へ集約しない。最終結果を小さく集約してからdriverへ返す設計が重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分散処理では、大規模データを不用意に1台へ集約しない。最終結果を小さく集約してからdriverへ返す設計が重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0078

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：標準

「分散技術・データ基盤」の理解確認で、レビュー担当者は「計算再利用の方法ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：Sparkのpersist/cachelは、複数回利用する中間結果を保持して再計算を減らすために使える。

答え・解説

正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。Sparkのpersist/cachelは、複数回利用する中間結果を保持して再計算を減らすために使える。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：Sparkのpersist/cachelは、複数回利用する中間結果を保持して再計算を減らすために使える。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0079

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：応用

「シャッフル最適化」について誤答分析を行ったところ、「大規模データではdriverメモリ問題を起こし得る。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：分散処理のボトルネックがシャッフルなら、JOIN・grouping・partitioningの設計を見直してデータ移動を減らす。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。分散処理のボトルネックがシャッフルなら、JOIN・grouping・partitioningの設計を見直してデータ移動を減らす。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分散処理のボトルネックがシャッフルなら、JOIN・grouping・partitioningの設計を見直してデータ移動を減らす。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0080

5-3 データ蓄積・加工 > 分散技術・データ基盤 | 難易度：応用

「分散技術・データ基盤」の解答を見直している。誤答の理由は「原因解決にならずデータ損失を招く。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：並列処理では平均だけでなくタスクごとの処理量・時間を確認し、データスキューや特定パーティションの偏りを診断する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。並列処理では平均だけでなくタスクごとの処理量・時間を確認し、データスキューや特定パーティションの偏りを診断する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：並列処理では平均だけでなくタスクごとの処理量・時間を確認し、データスキューや特定パーティションの偏りを診断する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0081

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：基礎

「SQLフィルタ」の演習レビューで、誤答理由として「結果の並び順を指定する。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：WHEREは行単位のフィルタリングに使い、集計前に対象データを絞る。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ア

ア：正しい。WHEREは行単位のフィルタリングに使い、集計前に対象データを絞る。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：WHEREは行単位のフィルタリングに使い、集計前に対象データを絞る。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0082

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：基礎

「フィルタ・結合・変換・演算処理」の理解確認で、レビュー担当者は「行を条件で除外する句であり、グループ化そのものではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：GROUP BYは分析で頻出する基本操作で、カテゴリ単位などの集計粒度を定義する。

答え・解説

正答：エ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。GROUP BYは分析で頻出する基本操作で、カテゴリ単位などの集計粒度を定義する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：GROUP BYは分析で頻出する基本操作で、カテゴリ単位などの集計粒度を定義する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0083

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：基礎

「JOIN基礎」について誤答分析を行ったところ、「INNER JOINは一致しない行を基本的に結果へ残さない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。
ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。
イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
ウ：JOINでは、どのテーブルの未対応行を残すかによりINNER/LEFTなどを使い分ける。
エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。JOINでは、どのテーブルの未対応行を残すかによりINNER/LEFTなどを使い分ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：JOINでは、どのテーブルの未対応行を残すかによりINNER/LEFTなどを使い分ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0084

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：標準

「フィルタ・結合・変換・演算処理」の解答を見直している。誤答の理由は「店舗別集計にはグループ化が必要である。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。
ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
イ：SQLの処理では、行条件と集計後条件を区別し、WHEREとHAVINGを適切に使い分ける。
ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説

正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。SQLの処理では、行条件と集計後条件を区別し、WHEREとHAVINGを適切に使い分ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：SQLの処理では、行条件と集計後条件を区別し、WHEREとHAVINGを適切に使い分ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0085

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：標準

「LEFT JOIN」の演習レビューで、誤答理由として「注文がない顧客は一致行がなく結果から落ちる。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：未対応行も分析対象へ残す必要がある場合、LEFT JOINなど外部結合を使う。どちらを基準集合にするかが重要である。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：ア

ア：正しい。未対応行も分析対象へ残す必要がある場合、LEFT JOINなど外部結合を使う。どちらを基準集合にするかが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：未対応行も分析対象へ残す必要がある場合、LEFT JOINなど外部結合を使う。どちらを基準集合にするかが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0086

5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：標準

「フィルタ・結合・変換・演算処理」の理解確認で、レビュー担当者は「重複を大幅に増やす。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：DISTINCTは重複値を除いた一覧を得る基本操作だが、なぜ重複があるかを理解せず乱用しないことも重要である。

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。DISTINCTは重複値を除いた一覧を得る基本操作だが、なぜ重複があるかを理解せず乱用しないことも重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：DISTINCTは重複値を除いた一覧を得る基本操作だが、なぜ重複があるかを理解せず乱用しないことも重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0087 5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：標準

「集約関数」について誤答分析を行ったところ、「JOINとLIMITは集計値を計算する関数ではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：集計では、求めたい指標の意味に応じてCOUNT、SUM、AVG、MIN、MAXなどを使い分ける。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。集計では、求めたい指標の意味に応じてCOUNT、SUM、AVG、MIN、MAXなどを使い分ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：集計では、求めたい指標の意味に応じてCOUNT、SUM、AVG、MIN、MAXなどを使い分ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0088 5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：標準

「フィルタ・結合・変換・演算処理」の解答を見直している。誤答の理由は「文字列連結や解釈困難を招く。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：データ加工では、列の意味に合う型へそろえ、カテゴリコードなどは分析者が解釈できる形へ変換する。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。データ加工では、列の意味に合う型へそろえ、カテゴリコードなどは分析者が解釈できる形へ変換する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ加工では、列の意味に合う型へそろえ、カテゴリコードなどは分析者が解釈できる形へ変換する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0089 5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：応用

「JOIN多重化」の演習レビューで、誤答理由として「事例の増加はJOINのキー多重性で説明できる。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：JOIN前に各表の粒度とキー一意性を確認し、意図しない多対多結合による件数・金額の膨張を防ぐ。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ア

ア：正しい。JOIN前に各表の粒度とキー一意性を確認し、意図しない多対多結合による件数・金額の膨張を防ぐ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：JOIN前に各表の粒度とキー一意性を確認し、意図しない多対多結合による件数・金額の膨張を防ぐ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0090 5-3 データ蓄積・加工 > フィルタ・結合・変換・演算処理 | 難易度：応用

「フィルタ・結合・変換・演算処理」の理解確認で、レビュー担当者は「LIMITは返却件数制限であり集計ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQL加工は、どの段階で行を絞り、どの単位でグループ化し、どの集計結果を残すかを順序立てて設計する。

答え・解説 正答：エ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。SQL加工は、どの段階で行を絞り、どの単位でグループ化し、どの集計結果を残すかを順序立てて設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：SQL加工は、どの段階で行を絞り、どの単位でグループ化し、どの集計結果を残すかを順序立てて設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0091

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：基礎

「状態更新」について誤答分析を行ったところ、「実行順をランダム化する仕組みではない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：変数は値を保持し、後続処理から名前で参照するための基本要素である。型やスコープも処理の正しさに関係する。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。変数は値を保持し、後続処理から名前で参照するための基本要素である。型やスコープも処理の正しさに関係する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：変数は値を保持し、後続処理から名前で参照するための基本要素である。型やスコープも処理の正しさに関係する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0092

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：基礎

「基礎プログラミング・アルゴリズム」の解答を見直している。誤答の理由は「再帰は関数が自身を呼ぶ構造であり、この単純な条件判定には不要である。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：条件分岐は、真偽条件に応じて異なる処理を実行するための基本的な制御構造である。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：イ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。条件分岐は、真偽条件に応じて異なる処理を実行するための基本的な制御構造である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：条件分岐は、真偽条件に応じて異なる処理を実行するための基本的な制御構造である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0093 5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：基礎

「ループ」の演習レビューで、誤答理由として「反復処理の代替にはならない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：反復処理はコレクション内の各要素に同じ処理を適用する代表的な方法で、重複コードを減らせる。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ア

ア：正しい。反復処理はコレクション内の各要素に同じ処理を適用する代表的な方法で、重複コードを減らせる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：反復処理はコレクション内の各要素に同じ処理を適用する代表的な方法で、重複コードを減らせる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0094 5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：標準

「基礎プログラミング・アルゴリズム」の理解確認で、レビュー担当者は「修正漏れや不整合が起きやすい。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：再利用される一貫した処理は、責務を明確にした関数としてまとめると、テスト・再利用・変更がしやすくなる。

答え・解説 正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。再利用される一貫した処理は、責務を明確にした関数としてまとめると、テスト・再利用・変更がしやすくなる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：再利用される一貫した処理は、責務を明確にした関数としてまとめると、テスト・再利用・変更がしやすくなる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0095 5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：標準

「型変換」について誤答分析を行ったところ、「列名変更だけでは値の型や内容は変わらない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：外部入力は想定どおりの型とは限らないため、変換可能性を検証し、欠損・不正値の扱いを明示してから演算する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：ウ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。外部入力は想定どおりの型とは限らないため、変換可能性を検証し、欠損・不正値の扱いを明示してから演算する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部入力は想定どおりの型とは限らないため、変換可能性を検証し、欠損・不正値の扱いを明示してから演算する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0096 5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：標準

「基礎プログラミング・アルゴリズム」の解答を見直している。誤答の理由は「入力検証と例外処理は役割が異なる。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：例外処理では、想定される異常を識別し、ログ、再試行、フォールバック、停止など業務要件に合う挙動を設計する。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。例外処理では、想定される異常を識別し、ログ、再試行、フォールバック、停止など業務要件に合う挙動を設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：例外処理では、想定される異常を識別し、ログ、再試行、フォールバック、停止など業務要件に合う挙動を設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0097

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：標準

「探索アルゴリズム」の演習レビューで、誤答理由として「暗号化は二分探索の必要条件ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：二分探索は整列済みデータを前提に探索範囲を段階的に半減させる。未整列データにはそのまま適用できない。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：ア

ア：正しい。二分探索は整列済みデータを前提に探索範囲を段階的に半減させる。未整列データにはそのまま適用できない。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：二分探索は整列済みデータを前提に探索範囲を段階的に半減させる。未整列データにはそのまま適用できない。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0098

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：標準

「基礎プログラミング・アルゴリズム」の理解確認で、レビュー担当者は「単純に1回ずつ調べるなら通常は線形に増える。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：アルゴリズム選択では、入力サイズが増えたときに処理量がどの程度増えるかという計算量の見方が重要である。

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。アルゴリズム選択では、入力サイズが増えたときに処理量がどの程度増えるかという計算量の見方が重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：アルゴリズム選択では、入力サイズが増えたときに処理量がどの程度増えるかという計算量の見方が重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0099

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：応用

「アルゴリズム改善」について誤答分析を行ったところ、「自動処理の代替にならず現実的でない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：キーによる検索・結合が可能な問題で全組合せ比較を行うのは非効率である。適切なデータ構造やDBのJOINを利用する。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。キーによる検索・結合が可能な問題で全組合せ比較を行うのは非効率である。適切なデータ構造やDBのJOINを利用する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：キーによる検索・結合が可能な問題で全組合せ比較を行うのは非効率である。適切なデータ構造やDBのJOINを利用する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0100

5-4 プログラミング・SQL > 基礎プログラミング・アルゴリズム | 難易度：応用

「基礎プログラミング・アルゴリズム」の解答を見直している。誤答の理由は「恣意的な選択となり、評価の信頼性を損なう。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：分析コードは入力データ、ライブラリ、パラメータ、乱数条件などを記録し、同じ処理を再実行できるようにすることが重要である。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説

正答：イ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。分析コードは入力データ、ライブラリ、パラメータ、乱数条件などを記録し、同じ処理を再実行できるようにすることが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分析コードは入力データ、ライブラリ、パラメータ、乱数条件などを記録し、同じ処理を再実行できるようにすることが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0101 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：基礎

「射影」の演習レビューで、誤答理由として「HAVINGは集約後のグループを絞り込むために使う。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：SQLのSELECT句は、結果として返す列や計算式を指定する基本要素である。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ア

ア：正しい。SQLのSELECT句は、結果として返す列や計算式を指定する基本要素である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：SQLのSELECT句は、結果として返す列や計算式を指定する基本要素である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0102 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：基礎

「分析プログラム・SQL」の理解確認で、レビュー担当者は「そのような自動補完はNULLの定義ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：NULLは通常の値とは異なるため、比較や集計での扱いを明示的に理解する必要がある。

答え・解説 正答：エ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。NULLは通常の値とは異なるため、比較や集計での扱いを明示的に理解する必要がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：NULLは通常の値とは異なるため、比較や集計での扱いを明示的に理解する必要がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0103

5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：基礎

「集約」について誤答分析を行ったところ、「並べ替えだけでは集計されない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：グループ単位の集計ではGROUP BYで単位を指定し、COUNT・SUM・AVGなどの集約関数を使う。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

0104

5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：標準

「分析プログラム・SQL」の解答を見直している。誤答の理由は「INNER JOINの挙動と逆である。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：INNER JOINは結合条件に一致する組だけを返す。片側の未一致行も残したい場合は外部結合を検討する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説

正答：ウ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。グループ単位の集計ではGROUP BYで単位を指定し、COUNT・SUM・AVGなどの集約関数を使う。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：グループ単位の集計ではGROUP BYで単位を指定し、COUNT・SUM・AVGなどの集約関数を使う。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。INNER JOINは結合条件に一致する組だけを返す。片側の未一致行も残したい場合は外部結合を検討する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：INNER JOINは結合条件に一致する組だけを返す。片側の未一致行も残したい場合は外部結合を検討する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0105 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：標準

「集約前後」の演習レビューで、誤答理由として「HAVINGは削除処理ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：SQLでは処理段階を意識し、行レベルの条件と集約後のグループ条件を適切に分ける。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ア

ア：正しい。SQLでは処理段階を意識し、行レベルの条件と集約後のグループ条件を適切に分ける。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：SQLでは処理段階を意識し、行レベルの条件と集約後のグループ条件を適切に分ける。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0106 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：標準

「分析プログラム・SQL」の理解確認で、レビュー担当者は「各注文行を残す要件を満たさない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：行粒度を保持しながらグループ内順位・累積値・平均との差などを計算する場合、ウィンドウ関数が有効である。

答え・解説 正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。 この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。 この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。行粒度を保持しながらグループ内順位・累積値・平均との差などを計算する場合、ウィンドウ関数が有効である。 誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：行粒度を保持しながらグループ内順位・累積値・平均との差などを計算する場合、ウィンドウ関数が有効である。 本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0107 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：標準

「SQL構造化」について誤答分析を行ったところ、「重複が増え、修正漏れの原因になる。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：分析SQLは正しく動くだけでなく、レビュー・テスト・変更ができるよう段階と責務を明確に構造化する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：ウ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。分析SQLは正しく動くだけでなく、レビュー・テスト・変更ができるよう段階と責務を明確に構造化する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分析SQLは正しく動くだけでなく、レビュー・テスト・変更ができるよう段階と責務を明確に構造化する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0108 5-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：標準

「分析プログラム・SQL」の解答を見直している。誤答の理由は「不要な計算量を増やす。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：分析プログラムでは、ライブラリが提供するベクトル化・一括処理を活用すると、可読性や性能を改善できる場合がある。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。分析プログラムでは、ライブラリが提供するベクトル化・一括処理を活用すると、可読性や性能を改善できる場合がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：分析プログラムでは、ライブラリが提供するベクトル化・一括処理を活用すると、可読性や性能を改善できる場合がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01095-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：応用

「JOIN診断」の演習レビューで、誤答理由として「SUMにそのような性質はない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：JOIN前に主キー・外部キーと多重度を確認し、必要な1行を特定する条件を設計しないと集計値が水増しされることがある。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説正答：ア

ア：正しい。JOIN前に主キー・外部キーと多重度を確認し、必要な1行を特定する条件を設計しないと集計値が水増しされることがある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：JOIN前に主キー・外部キーと多重度を確認し、必要な1行を特定する条件を設計しないと集計値が水増しされることがある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01105-4 プログラミング・SQL > 分析プログラム・SQL | 難易度：応用

「分析プログラム・SQL」の理解確認で、レビュー担当者は「再現性や検証可能性を下げる。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：分析プログラムでは入力データの版、処理手順、パラメータ、依存関係を管理し、再実行可能な状態を保つことが重要である。

答え・解説正答：エ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。分析プログラムでは入力データの版、処理手順、パラメータ、依存関係を管理し、再実行可能な状態を保つことが重要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：分析プログラムでは入力データの版、処理手順、パラメータ、依存関係を管理し、再実行可能な状態を保つことが重要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0111 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：基礎

「API利用」について誤答分析を行ったところ、「失敗状態を適切に扱う必要がある。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：外部AIサービスを利用する際は、APIの認証、入力形式、出力形式、制限、エラー処理を理解して組み込む。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。外部AIサービスを利用する際は、APIの認証、入力形式、出力形式、制限、エラー処理を理解して組み込む。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部AIサービスを利用する際は、APIの認証、入力形式、出力形式、制限、エラー処理を理解して組み込む。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0112 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：基礎

「AIサービス活用・生成AIによるコーディング支援」の解答を見直している。誤答の理由は「生成AIは誤ったコードや存在しないAPIを提示することがある。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：生成AIは開発補助として有用だが、出力を自動的に正解とみなさず、要件適合、テスト、セキュリティを人間が確認する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：イ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。生成AIは開発補助として有用だが、出力を自動的に正解とみなさず、要件適合、テスト、セキュリティを人間が確認する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：生成AIは開発補助として有用だが、出力を自動的に正解とみなさず、要件適合、テスト、セキュリティを人間が確認する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0113 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：基礎

「プロンプト設計」の演習レビューで、誤答理由として「処理仕様を正しく推定できない可能性が高い。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：コーディング支援では、目的、入出力、制約、エラー条件、利用環境などを具体化し、生成結果を評価できる状態にする。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ア

ア：正しい。コーディング支援では、目的、入出力、制約、エラー条件、利用環境などを具体化し、生成結果を評価できる状態にする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：コーディング支援では、目的、入出力、制約、エラー条件、利用環境などを具体化し、生成結果を評価できる状態にする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0114 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：標準

「AIサービス活用・生成AIによるコーディング支援」の理解確認で、レビュー担当者は「自己評価だけでは独立した検証にならない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：生成コードは通常のソフトウェアと同様に、仕様に基づいてユニットテストや境界値テストを行う必要がある。

答え・解説 正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。生成コードは通常のソフトウェアと同様に、仕様に基づいてユニットテストや境界値テストを行う必要がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：生成コードは通常のソフトウェアと同様に、仕様に基づいてユニットテストや境界値テストを行う必要がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0115 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：標準

「秘密情報保護」について誤答分析を行ったところ、「秘密情報を伏せる適切な方法になり得る。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：生成AIへ入力するデータは外部送信となる場合があるため、秘密情報・個人情報・機密コードの扱いを組織ルールに従って管理する。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：ウ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。生成AIへ入力するデータは外部送信となる場合があるため、秘密情報・個人情報・機密コードの扱いを組織ルールに従って管理する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：生成AIへ入力するデータは外部送信となる場合があるため、秘密情報・個人情報・機密コードの扱いを組織ルールに従って管理する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0116 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：標準

「AIサービス活用・生成AIによるコーディング支援」の解答を見直している。誤答の理由は「セキュリティや動作上のリスクが高い。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIの技術的提案は、公式ドキュメント・パッケージ情報・対象バージョンなどで実在性と互換性を確認する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。生成AIの技術的提案は、公式ドキュメント・パッケージ情報・対象バージョンなどで実在性と互換性を確認する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：生成AIの技術的提案は、公式ドキュメント・パッケージ情報・対象バージョンなどで実在性と互換性を確認する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0117 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：標準

「レビュー」の演習レビューで、誤答理由として「量は品質指標ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：AIコーディング支援を使っても、コード所有責任は利用組織側にあり、通常の品質保証プロセスを省略しない。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ア

ア：正しい。AIコーディング支援を使っても、コード所有責任は利用組織側にあり、通常の品質保証プロセスを省略しない。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：AIコーディング支援を使っても、コード所有責任は利用組織側にあり、通常の品質保証プロセスを省略しない。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0118 5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：標準

「AIサービス活用・生成AIによるコーディング支援」の理解確認で、レビュー担当者は「意味上の誤りは実行エラーにならないことがある。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

エ：AIが生成した分析コードは、構文エラーの有無だけでなく、業務定義・集計粒度・既知ケースでの期待結果まで検証する。

答え・解説 正答：エ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。AIが生成した分析コードは、構文エラーの有無だけでなく、業務定義・集計粒度・既知ケースでの期待結果まで検証する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：AIが生成した分析コードは、構文エラーの有無だけでなく、業務定義・集計粒度・既知ケースでの期待結果まで検証する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0119

5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：応用

「権限設計」について誤答分析を行ったところ、「追跡や事故分析ができなくなる。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

ウ：AIサービスに外部ツール操作権限を与える場合、最小権限、承認、監査、入力検証を組み合わせることで影響範囲を制御する。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。AIサービスに外部ツール操作権限を与える場合、最小権限、承認、監査、入力検証を組み合わせることで影響範囲を制御する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：AIサービスに外部ツール操作権限を与える場合、最小権限、承認、監査、入力検証を組み合わせることで影響範囲を制御する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0120

5-4 プログラミング・SQL > AIサービス活用・生成AIによるコーディング支援 | 難易度：応用

「AIサービス活用・生成AIによるコーディング支援」の解答を見直している。誤答の理由は「利用回数は成果そのものではない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：生成AIによる開発支援は、速度、品質、セキュリティ、保守性、レビュー負荷などを含む全体プロセスで効果を測る。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：イ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。生成AIによる開発支援は、速度、品質、セキュリティ、保守性、レビュー負荷などを含む全体プロセスで効果を測る。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：生成AIによる開発支援は、速度、品質、セキュリティ、保守性、レビュー負荷などを含む全体プロセスで効果を測る。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0121 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：基礎

「CSV」の演習レビューで、誤答理由として「JPEGは主に画像形式であり表形式データ交換には適さない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：CSVは単純な表形式データの受け渡しに広く使われるが、文字コード、区切り、引用符、型情報の扱いを共有する必要がある。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説 正答：ア

ア：正しい。CSVは単純な表形式データの受け渡しに広く使われるが、文字コード、区切り、引用符、型情報の扱いを共有する必要がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：CSVは単純な表形式データの受け渡しに広く使われるが、文字コード、区切り、引用符、型情報の扱いを共有する必要がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0122 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：基礎

「データ出力・共有・連携」の理解確認で、レビュー担当者は「WAVは音声データ形式である。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：JSONは言語非依存の軽量なテキスト形式で、API等の構造化データ交換に広く用いられる。

答え・解説 正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。JSONは言語非依存の軽量なテキスト形式で、API等の構造化データ交換に広く用いられる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：JSONは言語非依存の軽量なテキスト形式で、API等の構造化データ交換に広く用いられる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01235-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：基礎

「メタデータ」について誤答分析を行ったところ、「メタデータはハードウェア性能を変えない。」という問題点
が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
ウ：データ共有では値だけでなく、定義、単位、粒度、更新時刻、品質、利用制約などを合わせて伝えると誤用を
防ぎやすい。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説正答：ウ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。データ共有では値だけでなく、定義、単位、粒度、更新時刻、品質、利用制約などを合わせて伝えると誤用を防ぎやすい。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ共有では値だけでなく、定義、単位、粒度、更新時刻、品質、利用制約などを合わせて伝えると誤用を防ぎやすい。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01245-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：標準

「データ出力・共有・連携」の解答を見直している。誤答の理由は「自動対応は保証されない。」であった。再発
を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
イ：システム間連携では、APIやデータ契約の変更が利用者へ与える影響を管理し、バージョンングや移行期間を
設ける。
ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。システム間連携では、APIやデータ契約の変更が利用者へ与える影響を管理し、バージョンングや移行期間を設ける。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：システム間連携では、APIやデータ契約の変更が利用者へ与える影響を管理し、バージョンングや移行期間を設ける。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0125 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：標準

「連携方式」の演習レビューで、誤答理由として「自動連携要件に適さない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：連携方式は必要な鮮度・遅延・量・再送要件に応じて、バッチ、API、イベント/ストリームなどを選ぶ。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：ア

ア：正しい。連携方式は必要な鮮度・遅延・量・再送要件に応じて、バッチ、API、イベント/ストリームなどを選ぶ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：連携方式は必要な鮮度・遅延・量・再送要件に応じて、バッチ、API、イベント/ストリームなどを選ぶ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0126 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：標準

「データ出力・共有・連携」の理解確認で、レビュー担当者は「構造化データとして再利用しにくくなる。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：ファイル共有では、形式名だけでなくエンコーディング、区切り、引用規則、日付形式、NULL表現等まで契約として定める。

答え・解説 正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。ファイル共有では、形式名だけでなくエンコーディング、区切り、引用規則、日付形式、NULL表現等まで契約として定める。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：ファイル共有では、形式名だけでなくエンコーディング、区切り、引用規則、日付形式、NULL表現等まで契約として定める。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0127 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：標準

「データ契約」について誤答分析を行ったところ、「誤データを処理し続ける危険がある。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

ウ：データ共有はシステム間の契約であり、スキーマ・意味・品質・変更管理を明確にすることが安定連携につながる。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：ウ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。データ共有はシステム間の契約であり、スキーマ・意味・品質・変更管理を明確にすることが安定連携につながる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ共有はシステム間の契約であり、スキーマ・意味・品質・変更管理を明確にすることが安定連携につながる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0128 5-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：標準

「データ出力・共有・連携」の解答を見直している。誤答の理由は「秘匿性やアクセス制御の代替にならない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：データ共有では利便性だけでなく、誰が何の目的でどの範囲へアクセスできるかを制御し、監査可能にする。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：イ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。データ共有では利便性だけでなく、誰が何の目的でどの範囲へアクセスできるかを制御し、監査可能にする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：データ共有では利便性だけでなく、誰が何の目的でどの範囲へアクセスできるかを制御し、監査可能にする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01295-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：応用

「冪等性」の演習レビューで、誤答理由として「システム上の整合性対策にならない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：ネットワーク連携では再送・重複を前提に、イベントID、一意制約、冪等な更新などで同一処理の多重適用を防ぐ。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説正答：ア

ア：正しい。ネットワーク連携では再送・重複を前提に、イベントID、一意制約、冪等な更新などで同一処理の多重適用を防ぐ。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：ネットワーク連携では再送・重複を前提に、イベントID、一意制約、冪等な更新などで同一処理の多重適用を防ぐ。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01305-5 データ共有・AI運用 > データ出力・共有・連携 | 難易度：応用

「データ出力・共有・連携」の理解確認で、レビュー担当者は「広範な障害につながる可能性がある。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

エ：共有スキーマの破壊的変更は、バージョンング、並行期間、利用者通知、互換性テストを使って段階的に進める。

答え・解説正答：エ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。共有スキーマの破壊的変更は、バージョンング、並行期間、利用者通知、互換性テストを使って段階的に進める。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：共有スキーマの破壊的変更は、バージョンング、並行期間、利用者通知、互換性テストを使って段階的に進める。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01315-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：基礎

「AutoML概念」について誤答分析を行ったところ、「入力データの品質確認は別途必要である。」という問題点
が確認された。次に示す判断原則のうち最も適切なものはどれか。
ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。
イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
ウ：AutoMLはモデル開発の探索や学習を自動化して効率化する一方、目的・データ・評価・運用の責任は残る。
エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。AutoMLはモデル開発の探索や学習を自動化して効率化する一方、目的・データ・評価・運用の責任は残る。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：AutoMLはモデル開発の探索や学習を自動化して効率化する一方、目的・データ・評価・運用の責任は残る。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

01325-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：基礎

「AutoML・MLOps・AIOps」の解答を見直している。誤答の理由は「MLライフサイクル管理の説明ではない。」
であった。再発を防ぐフィードバックとして最も適切なものはどれか。
ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
イ：MLOpsではデータ・コード・モデル・設定・評価・デプロイ・監視を一連の運用可能なプロセスとして管理する。
ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。MLOpsではデータ・コード・モデル・設定・評価・デプロイ・監視を一連の運用可能なプロセスとして管理する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：MLOpsではデータ・コード・モデル・設定・評価・デプロイ・監視を一連の運用可能なプロセスとして管理する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0133

5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：基礎

「AIOps概念」の演習レビューで、誤答理由として「AIOpsの目的ではない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：AIOpsは運用監視データを分析して、異常検知、イベント相関、原因候補提示、自動化などを支援する考え方である。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

0134

5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：標準

「AutoML・MLOps・AIOps」の理解確認で、レビュー担当者は「探索数だけでは性能を保証しない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：AutoMLは探索を自動化しても、データ分割、評価指標、業務制約、最終検証は人間が設計・確認する必要がある。

答え・解説

正答：ア

ア：正しい。AIOpsは運用監視データを分析して、異常検知、イベント相関、原因候補提示、自動化などを支援する考え方である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：AIOpsは運用監視データを分析して、異常検知、イベント相関、原因候補提示、自動化などを支援する考え方である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。AutoMLは探索を自動化しても、データ分割、評価指標、業務制約、最終検証は人間が設計・確認する必要がある。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：AutoMLは探索を自動化しても、データ分割、評価指標、業務制約、最終検証は人間が設計・確認する必要がある。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0135 5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：標準

「モデル管理」について誤答分析を行ったところ、「組織として再現できない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：MLOpsではモデル成果物と、その生成に使ったデータ・コード・設定・評価をバージョン付きで追跡可能にする。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。MLOpsではモデル成果物と、その生成に使ったデータ・コード・設定・評価をバージョン付きで追跡可能にする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：MLOpsではモデル成果物と、その生成に使ったデータ・コード・設定・評価をバージョン付きで追跡可能にする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0136 5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：標準

「AutoML・MLOps・AIOps」の解答を見直している。誤答の理由は「モデル品質との関連がない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：MLOpsの監視はシステム稼働だけでなく、データドリフト、予測分布、遅延、実測性能などモデル固有の指標も含む。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説 正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。MLOpsの監視はシステム稼働だけでなく、データドリフト、予測分布、遅延、実測性能などモデル固有の指標も含む。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：MLOpsの監視はシステム稼働だけでなく、データドリフト、予測分布、遅延、実測性能などモデル固有の指標も含む。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0137 5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：標準

「継続学習」の演習レビューで、誤答理由として「件数だけでは品質や性能を保証しない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

- ア：継続学習では自動化と同時に、データ検証、モデル検証、承認、ロールバックを組み込み、品質を守る。
- イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
- ウ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
- エ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

答え・解説 正答：ア

ア：正しい。継続学習では自動化と同時に、データ検証、モデル検証、承認、ロールバックを組み込み、品質を守る。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：継続学習では自動化と同時に、データ検証、モデル検証、承認、ロールバックを組み込み、品質を守る。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0138 5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：標準

「AutoML・MLOps・AIOps」の理解確認で、レビュー担当者は「業務影響を考慮した判断が必要である。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

- ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
- イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
- ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
- エ：AIOpsは運用判断を支援するもので、変更履歴や業務スケジュールと組み合わせ、誤検知・見逃しを継続評価する。

答え・解説 正答：エ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。AIOpsは運用判断を支援するもので、変更履歴や業務スケジュールと組み合わせ、誤検知・見逃しを継続評価する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：AIOpsは運用判断を支援するもので、変更履歴や業務スケジュールと組み合わせ、誤検知・見逃しを継続評価する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0139

5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：応用

「ロールバック」について誤答分析を行ったところ、「切り戻しができない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。

イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

ウ：モデルのデプロイは、段階的リリース、監視、版管理、ロールバックを含めて設計すると本番リスクを抑えられる。

エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

0140

5-5 データ共有・AI運用 > AutoML・MLOps・AIOps | 難易度：応用

「AutoML・MLOps・AIOps」の解答を見直している。誤答の理由は「再現性を失う。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

イ：MLOpsでは学習・評価・本番推論の特徴量処理を一貫させ、training-serving skewを防ぐためのテストや共通化を行う。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説

正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。モデルのデプロイは、段階的リリース、監視、版管理、ロールバックを含めて設計すると本番リスクを抑えられる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：モデルのデプロイは、段階的リリース、監視、版管理、ロールバックを含めて設計すると本番リスクを抑えられる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

答え・解説

正答：イ

ア：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。MLOpsでは学習・評価・本番推論の特徴量処理を一貫させ、training-serving skewを防ぐためのテストや共通化を行う。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：MLOpsでは学習・評価・本番推論の特徴量処理を一貫させ、training-serving skewを防ぐためのテストや共通化を行う。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0141 5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：基礎

「CIA」の演習レビューで、誤答理由として「これは主に完全性に関係する。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

- ア：情報セキュリティの基本要素として、機密性・完全性・可用性を区別して理解する。
- イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。
- ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
- エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

答え・解説 正答：ア

ア：正しい。情報セキュリティの基本要素として、機密性・完全性・可用性を区別して理解する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：情報セキュリティの基本要素として、機密性・完全性・可用性を区別して理解する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0142 5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：基礎

「ITセキュリティ・暗号化・認証」の理解確認で、レビュー担当者は「暗号化の目的ではない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

- ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
- イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。
- ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。
- エ：暗号化はデータの機密性を守る代表的手段であり、鍵管理と組み合わせて利用する。

答え・解説 正答：エ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：正しい。暗号化はデータの機密性を守る代表的手段であり、鍵管理と組み合わせて利用する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
総合解説：暗号化はデータの機密性を守る代表的手段であり、鍵管理と組み合わせて利用する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0143 5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：基礎

「認証認可」について誤答分析を行ったところ、「集約はデータをまとめる処理である。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。
ア：本番運用後の監視や再現性を考えず、一度動けば十分とする。
イ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。
ウ：認証は「誰か」を確認し、認可は「何をしてよいか」を制御する。両者を分けて設計する。
エ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

答え・解説 正答：ウ

ア：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。
ウ：正しい。認証は「誰か」を確認し、認可は「何をしてよいか」を制御する。両者を分けて設計する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
エ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：認証は「誰か」を確認し、認可は「何をしてよいか」を制御する。両者を分けて設計する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0144 5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：標準

「ITセキュリティ・暗号化・認証」の解答を見直している。誤答の理由は「暗号方式の説明ではなく危険な運用である。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。
ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。
イ：AESは代表的な対称鍵暗号であり、データ暗号化に利用される。アルゴリズムだけでなく鍵の安全管理が必要である。
ウ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。
エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説 正答：イ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。
イ：正しい。AESは代表的な対称鍵暗号であり、データ暗号化に利用される。アルゴリズムだけでなく鍵の安全管理が必要である。誤答理由から、定義・前提・目的に戻って修正する説明になっている。
ウ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。
エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。
総合解説：AESは代表的な対称鍵暗号であり、データ暗号化に利用される。アルゴリズムだけでなく鍵の安全管理が必要である。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0145

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：標準

「TLS」の演習レビューで、誤答理由として「通信内容の保護にはならない。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：機密データをネットワークで送る際は、TLS等を用いて転送中のデータを保護し、相手先の認証も適切に行う。

イ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：ア

ア：正しい。機密データをネットワークで送る際は、TLS等を用いて転送中のデータを保護し、相手先の認証も適切に行う。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：機密データをネットワークで送る際は、TLS等を用いて転送中のデータを保護し、相手先の認証も適切に行う。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0146

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：標準

「ITセキュリティ・暗号化・認証」の理解確認で、レビュー担当者は「秘密情報漏洩の原因になる。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：認証ではパスワードだけに依存せず、必要な保証レベルに応じて多要素認証等を組み合わせる。

答え・解説

正答：エ

ア：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：正しい。認証ではパスワードだけに依存せず、必要な保証レベルに応じて多要素認証等を組み合わせる。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

総合解説：認証ではパスワードだけに依存せず、必要な保証レベルに応じて多要素認証等を組み合わせる。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

対象：2026年度 / 基準日 2026-09-09

0147

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：標準

「最小権限」について誤答分析を行ったところ、「隠蔽は権限管理の代替にならない。」という問題点が確認された。次に示す判断原則のうち最も適切なものはどれか。

ア：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

イ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

ウ：ゼロトラストや最小権限の考え方では、主体に必要な資源・操作だけを許可し、過剰な権限を持たせない。

エ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

答え・解説

正答：ウ

ア：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：正しい。ゼロトラストや最小権限の考え方では、主体に必要な資源・操作だけを許可し、過剰な権限を持たせない。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

エ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：ゼロトラストや最小権限の考え方では、主体に必要な資源・操作だけを許可し、過剰な権限を持たせない。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0148

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：標準

「ITセキュリティ・暗号化・認証」の解答を見直している。誤答の理由は「リポジトリには残るため秘密にならない。」であった。再発を防ぐフィードバックとして最も適切なものはどれか。

ア：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

イ：認証情報はソースコードへ埋め込まず、アクセス制御されたシークレット管理を使い、漏洩時は失効・ローテーションする。

ウ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

エ：SQLのJOINでは多重度を確認せず、行数が増えてもそのまま集計する。

答え・解説

正答：イ

ア：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。認証情報はソースコードへ埋め込まず、アクセス制御されたシークレット管理を使い、漏洩時は失効・ローテーションする。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。JOINによる行の多重化は集計値の水増しにつながる。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：認証情報はソースコードへ埋め込まず、アクセス制御されたシークレット管理を使い、漏洩時は失効・ローテーションする。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

0149

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：応用

「インジェクション」の演習レビューで、誤答理由として「侵害時の影響を大きくする。」が挙げられた。この指摘を踏まえた修正方針として最も適切なものはどれか。

ア：外部入力をSQLへ直接連結せず、パラメータ化や適切な入力処理を用いる。DB権限も最小化し多層防御を行う。

イ：生成AIが書いたコードは、テストや依存関係確認をせず本番へ反映する。

ウ：要件や障害時の影響を確認せず、構成要素を増やすほど良いと考える。

エ：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

0150

5-5 データ共有・AI運用 > ITセキュリティ・暗号化・認証 | 難易度：応用

「ITセキュリティ・暗号化・認証」の理解確認で、レビュー担当者は「鍵管理とアクセス制御が不十分なら安全とはいえない。」と指摘した。正しい理解へ戻すための説明として最も適切なものはどれか。

ア：スキーマ・型・キー・通信条件を確認せず、受け取ったデータをそのまま結合する。

イ：暗号化ではアルゴリズム選定だけでなく、鍵の生成・保管・アクセス・ローテーション・失効・監査までライフサイクルで管理する。

ウ：認証・権限・暗号化を省略し、内部利用なら安全だとみなす。

エ：本番運用後の監視や再現性を考えず、一度動けば十分とする。

答え・解説

正答：ア

ア：正しい。外部入力をSQLへ直接連結せず、パラメータ化や適切な入力処理を用いる。DB権限も最小化し多層防御を行う。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

イ：不適切。生成コードもレビュー・テスト・セキュリティ確認が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

ウ：不適切。システム設計は要件・コスト・可用性・運用性のトレードオフで決める。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：外部入力をSQLへ直接連結せず、パラメータ化や適切な入力処理を用いる。DB権限も最小化し多層防御を行う。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。

答え・解説

正答：イ

ア：不適切。型・キー・粒度の不整合は欠損や多重化を生む。この対応では、提示された誤答理由を正しい原則へ修正できない。

イ：正しい。暗号化ではアルゴリズム選定だけでなく、鍵の生成・保管・アクセス・ローテーション・失効・監査までライフサイクルで管理する。誤答理由から、定義・前提・目的に戻って修正する説明になっている。

ウ：不適切。内部環境でも認証・認可・暗号化などの統制が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

エ：不適切。継続運用では監視・再現性・変更管理が必要である。この対応では、提示された誤答理由を正しい原則へ修正できない。

総合解説：暗号化ではアルゴリズム選定だけでなく、鍵の生成・保管・アクセス・ローテーション・失効・監査までライフサイクルで管理する。本問では、直接の解答ではなく、誤答理由から正しい原則へ戻れるかを確認する。