

令和5年度秋期 応用情報技術者試験 午後問6「企業向け電子書籍サービス」 徹底復習・分析レポート

1. 本問題の全体像と出題趣旨の構造的分析

個人向け電子書籍の販売および閲覧サービスを提供するH社が、近年のテレワーク普及に伴う企業ニーズに応える形で「企業向け電子書籍サービス」を追加設計・実装するという実務に即した業務要件が本問題の題材です。既存の個人向けシステム基盤やID体系をそのまま維持・継承しつつ、法人展開（一括購入、企業補助、割引購入、書籍閲覧）という新たなビジネスモデルに合わせて関係データベース（RDB）の構造を不整合なく拡張することは、実務におけるデータベース設計およびシステム開発においてきわめて戦略的重要性の高いテーマです。独立行政法人情報処理推進機構（IPA）が公表している出題趣旨によると、本問は企業向け電子書籍サービスを題材として、E-R図の解釈やSQL文の正確な記述、トランザクション処理における同時実行制御の理解、さらには複雑な業務要件をデータモデルへ正しく落とし込む実装能力を問う構成となっています。本問は単なる試験対策にとどまらず、実務で頻出する「既存スキーマの拡張設計」「データ競合（レースコンディション）を防ぐトランザクション制御」「性能と利便性を両立するSQLクエリ構築」といった高度なデータアーキテクチャ思考を身につける上で、非常に実用的価値が高い一問といえます。それでは、本問題の核となるデータベース構造の解析と最適化の手法について、設問1のデータモデル（E-R図）から順を追って確認していきましょう。

2. 設問1：データベース構造（E-R図）の最適化と整合性分析

概念データモデルにおける正確なリレーションシップの定義および必要な属性の漏れのない抽出は、データベース全体のデータ整合性を決定づける根幹であり、システム構築における基礎にして最重要の工程です。

空欄a, b, cの導出根拠と解答解説

- **空欄a（書籍分類と企業補助のリレーション）：** → （1対多）
- **導出根拠：** 表1のNo.2「企業補助」の業務要件において、「企業は負担する上限金額を書籍分類ごとに設定する」と規定されています。「書籍分類」エンティティは主キー「書籍分類ID」によって分類を一意に識別します。一方、「企業補助」エンティティは企業ごと・書籍分類ごとの補助条件を管理するため、複合外部キーとして「書籍分類ID」と「企業ID」を保持します。1つの書籍分類に対して複数の企業による補助設定（複数レコード）が対応する関係となるため、cardinality（多重度）の表記規則に従い、書籍分類（1）から企業補助（多）へ向かう矢印 → が導出されます。
- **空欄b（企業エンティティの追加属性）：** 割引率
- **導出根拠：** 表1のNo.3「割引購入」の業務要件に、「それぞれの企業がH社と契約した一定の割引率を適用した価格で購入できる」と明記されています。既存の「企業」エンティティには主キー「企業ID」や「企業名」が定義されていますが、契約に基づく値引きの計算根拠となる属性が存在しません。そのため、企業ごとの割引

購入機能を正しく計算・管理するための必須属性として「割引率」を追加定義する必要があります。

- **空欄c（社員書籍購入エンティティの追加属性）：** 書籍ID
- **導出根拠：** 表1のNo.2「企業補助」の要件において、社員が自学研鑽のために特定の分類の電子書籍を購入した際、企業負担額と社員負担額を記録する「社員書籍購入」エンティティが使用されます。しかし、初期の属性定義では購入日時や金額のみで、「どの電子書籍を購入したか」という参照情報が欠落しています。そのため、「書籍」エンティティの主キーである「書籍ID」を外部キーとして定義しなければなりません。
- **E-R図の表記ルールの留意点：** 試験問題の凡例にある通り、「実線の下線＝主キー」「破線の下線＝外部キー」を表します。したがって、解答として「書籍ID」を導出するだけでなく、E-R図上では 書籍ID に破線下線を付与して参照制約（FK）を表現することが求められます。

データモデルの定義漏れが引き起こす実務的リスク分析

属性やリレーションシップの定義漏れは、データベース層でのデータ不整合を引き起こし、最終的にアプリケーションの致命的な機能不全を招きます。例えば、「社員書籍購入」エンティティにおいて外部キー「書籍ID」の定義が欠落した場合、社員が企業補助を用いて書籍を購入したという会計上の決済履歴（購入日時や負担金額）は残るものの、「どの書籍の閲覧権限をその社員に付与すべきか」をシステムが判別不能となります。結果として、購入したはずの電子書籍がリーダ上で表示・閲覧できなくなり、サービスとして重大な欠陥を引き起こすリスクが存在します。データモデルの構造的健全性を確認したところで、続いてデータ更新時の整合性を担保する「一括購入割当処理とトランザクション制御」の設計について解説します。

3. 設問2：一括購入割当処理のロジックと同時実行制御の危機管理

複数のユーザーが同時にアクセスし更新を行うオンライン処理システムにおいて、トランザクションの適切な隔離性（Isolation）の確保とデータの整合性維持は、システムの信頼性を左右するきわめて重要な要素です。

表2の処理フローおよびSQL空欄補完（d, e, f）の解説

空欄, IPA公式解答例, 処理ロジックと構文の解説

d, COUNT(*), 手順2の「社員が割当依頼した一括購入IDのうち、現在割り当てられている数量を取得する」処理です。「一括購入割当」表には割当済みの社員レコードが蓄積されており、数量列を持たないため、行数を集計する COUNT(*) を用いて割り当て済み数量を取得します。

e, 大きい, 手順3における割当上限判定ロジックです。手順1で取得した「一括購入数量（上限枠）」が、手順2で取得した「割り当て済み数量」より「大きい」場合（購入数量 > 割当済み数量）、まだ割り当て可能な残枠が存在すると判定し、手順4の割当登録に進みます。

f, "VALUES (:一括購入 ID, :社員 ID, :企業 ID)", 手順4の「割当依頼した社員に一括購入IDを割り当てる」挿入処理です。INSERT INTO 一括購入割当 (一括購入ID, 社員ID, 企業

ID) の列指定順序に合わせて、埋込み変数を記述します。IPAの公式表記規則に基づき、コロンと変数名間のスペース配置 (:一括購入 ID, :社員 ID, :企業 ID) を正確に揃える必要があります。”

排他制御欠如による競合メカニズムとビジネスインパクト

下線部①の「専有ロック（排他ロック）」を使用せずに複数トランザクションが同時実行された場合、以下の問題が発生します。

- **IPA公式解答（字数制限28文字）：** 一括購入数量より多い数量の書籍を割り当ててしまう問題
- **試験対策アドバイス：** 記述問題で「どのような問題が発生するか（30文字以内）」と問われた場合、現象の原因ではなく「ビジネス・業務上の結果（購入数量を超えて割り当てられてしまうこと）」を簡潔かつ具体的に要約して凝縮記述するアプローチが不可欠です。

競合メカニズムの実行タイムライン分析

専有ロック（FOR UPDATE 等）を行わない場合、複数の社員が同一の一括購入IDに対して同時に割当依頼を行うと、以下のようなタイムラインでダーティリードやノンリピータブルリードに類する割り当て過多が発生します。

【時刻】		【トランザクションA (社員1)】	【トランザクションB (社員2)】
		└─ t1 ─> SELECT COUNT(*) (割当数9を取得)	
		(上限10に対して「残あり」と判定)	
			└─ t2 ─> SELECT COUNT(*) (割当数9を取得)
			(上限10に対して「残あり」と判定)
		└─ t3 ─> INSERT INTO 一括購入割当 ...	
		(割当数が 9 → 10 に増加)	
			└─ t4 ─> INSERT INTO 一括購入割当 ...
			(割当数が 10 → 11 に増加！)
	▼		
		【結果】 契約数量（10冊）を超過して 11 冊目の割り当てが発生（不整合）	

このように、両トランザクションが手順2で同一の「割当済み数量（9）」を平行して読み取ってしまうことで、どちらも手順3の判定を通過して手順4の INSERT を実行します。その結果、企業があらかじめ一括購入した契約ライセンス数を超える書籍が社員に割り当てられてしまい、企業の未払い損失や権利者に対するライセンス違反といった深刻なトラブルへ直結します。この競合を防ぐためには、手順1または手順2の段階で参照するレコードに対して SELECT ... FOR UPDATE による専有ロックを確保し、他トランザクションによる割り込み読み取り・書き込みを排他制御（Thread Interleavingの遮断）する必要があります。トランザクション制御によるデータ整合性の確保を理解した後は、複数の購入ルートから閲覧可能な書籍を抽出する「重複を排除したSQLデータ抽出」の設計に進めます。

4. 設問3：効率的なデータ抽出クエリ（SQL）の構築と集合操作

社員が電子書籍を閲覧する際、「個人として企業補助等を利用して購入した書籍（社員書籍購入）」と「企業の一括購入によって社員に割り当てられた書籍（一括購入割当）」という複数のルートから閲覧権限を得た書籍を、漏れなくかつ重複なく抽出するクエリ設計は、正確なUI表示とシステム性能の観点から決定的に重要です。

図2のSQL文における空欄gおよびhの解説

空欄g: *UNION*

図2のSQL文は、前半の「社員書籍購入（sk）」から購入した書籍IDを取得するクエリと、後半の「一括購入（ik）」および「一括購入割当（iw）」から割り当てられた書籍IDを取得するクエリの2つを統合する構造となっています。

- **UNION と UNION ALL の機能・内部動作の違い：**
- **UNION ALL：** 2つのクエリ結果を単純に結合して返します。内部的な重複排除処理を行わないため、CPUやメモリのオーバーヘッドが非常に小さく高速です。
- **UNION：** 2つのクエリ結果を統合した後、データベースエンジン内部でソート（Sort Distinct）やハッシュテーブルの構築（Hash Aggregate）を行い、重複する行を排除して一意な結果セットを生成します。
- **選定理由：** 本要件では「閲覧可能な重複を含まない書籍の一覧を取得する」ことが求められています。社員が「個人購入」と「一括購入割当」の両方で同じ書籍の閲覧権限を得ている場合、UNION ALL では画面上に同一書籍が2重表示されてしまいます。ソート/ハッシュ演算の負荷というパフォーマンス上のトレードオフを考慮した上でも、UI表示の整合性を保証するために重複を排除する UNION を選択することが不可欠です。

空欄h: *ON ik. 一括購入 ID = iw. 一括購入 ID*

一括購入表（ik）と一括購入割当表（iw）を内部結合（INNER JOIN）するための結合条件式です。

- **導出プロセス：** 一括購入表（ik）には購入された書籍の「書籍ID」が保持されており、一括購入割当表（iw）にはどの社員に割り当てられたか（社員ID・企業ID）が保持されています。ログインした社員に割り当てられている書籍を特定するには、両テーブルの共通属性でありリレーシヨンの鍵である「一括購入ID」を結びつける必要があります。設問の指示である「表の列名には必ずその表の関連名を付けて答えよ」に従い、*ON ik. 一括購入 ID = iw. 一括購入 ID* が導出されます。効率的なクエリ構築の手法を確認したところで、次にシステムの利便性を向上させる「UI/UX改善とデータモデルの拡張」について考察します。

5. 設問4：ユーザー体験（UX）向上に伴うデータモデルの拡張と画面設計

システム稼働後に発覚する「ログイン切り替えの手間」といったUI/UX上の課題に対し、既存のデータベース構造や認証基盤を破壊せずに適切なエンティティと画面を追加して解決を図る設計アプローチは、実務における機能拡張の標準的パターンです。

設問解答と構造的拡張の分析

(1) 追加エンティティの属性名

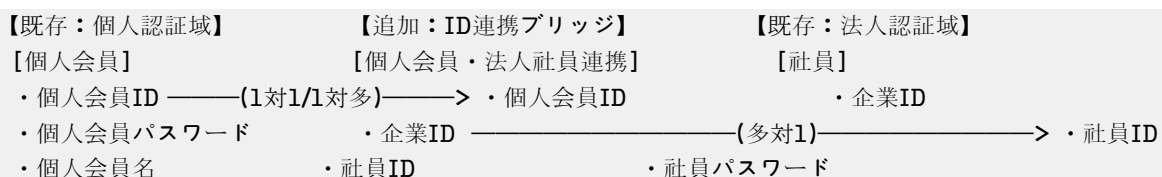
- **IPA公式解答：** 個人会員ID, 企業ID, 社員ID
- **構造的分析：** 既存サービスを個人で利用している社員が、個人会員IDでログインした状態のまま、法人社員として閲覧権限を持つ書籍も一括して参照できるようにするためのブリッジエンティティ（ID連携テーブル）の定義です。個人会員を特定する「個人会員ID」と、法人側の社員を特定する主キー（複合主キー）である「企業ID」および「社員ID」の3属性を定義することで、個人アカウントと法人アカウントの1対1（または1対多）の紐付け関係を安全にモデル化できます。

(2) 追加画面の機能と目的

- **IPA公式解答（字数制限21文字）：** 個人会員が企業IDと社員IDを登録する画面
- **解説と凝縮のテクニック：** (1)で追加したブリッジエンティティにデータを登録するための入力インターフェースが必要です。25文字以内という厳格な文字数制限の中で、「誰が（個人会員が）」「何を（企業IDと社員IDを）」「どうする画面か（登録する画面）」という要素を過不足なく盛り込むことで、21文字の簡潔な解答が完成します。

既存スキーマを保護するデータモデル拡張とUX向上の両立

既存の個人向け「個人会員」テーブルと、法人向け「社員」テーブルのID体系を強引に統合・改変するのではなく、両者を紐付ける独立したブリッジエンティティ（ID連携テーブル）を追加するアプローチは、アーキテクチャ設計としてきわめて優れています。



この設計パターンを採用することで、既存の認証ロジックやセキュリティ境界、データベースのテーブル構造に一切の非互換変更（Breaking Change）を与えることなく安全に機能拡張を達成できます。同時に、ユーザーに対しては「ログインID切り替えの煩雑さ解消」という直接的な顧客価値を提供でき、顧客エンゲージメントの維持とシステム運用セキュリティの厳格化を高度に両立させることが可能となります。以上の詳細な解説を踏まえ、最後に本問を通じて得られる応用情報技術者試験の攻略ノウハウを総括します。

6. 総括：応用情報技術者試験におけるデータベース分野の攻略ノウハウ

本問題の徹底分析から得られた知見は、応用情報技術者試験（午後問6）での高得点獲得のみならず、実際のシステム開発現場におけるデータベース設計・運用の指針としても直接活用できる実践的な内容です。

応用情報技術者試験（午後問6）攻略のための解法タクティクス

試験本番において、データベース問題を制限時間内に精度高く解き明かすための実践的タクティクスを伝授します。

- **業務要件テキストと E-R 図の高速対比**
- 問題文の「新サービスの機能概要（表1など）」を読み進める際、文章中の名詞（エンティティ候補）と数値・制約条件（「1つの～に対して」「上限を～ごとに設定」等）にマーカーを引き、即座に E-R 図の矢印（1対1、1対多、多対多）や属性の有無（欠落している主キー・外部キー）と照合します。
- **擬似コード・SQL処理フローからの競合リスク発見**
- 処理フロー（表2など）が提示された場合、「SELECTで状態読み取り → 条件判定 → INSERT/UPDATEで更新」という非原子的なステップを見つけたら、即座に排他制御（専有ロック）の欠如による競合リスク（レースコンディション）を疑ってください。
- **文字数制限のある短文記述の解法**
- 「どのような問題が発生するか」「どのような画面か」という設問に対しては、主語・目的語・結びの動作を明確にし、余分な修飾語を削ぎ落としてIPA解答例のような体裁（～する問題、～する画面）に凝縮させる練習を重ねてください。

データベース分野 攻略チェックリスト

- **1. E-R図の読み取りとモデル化**
- 業務要件文からエンティティ間のカーディナリティ（1対1、1対多、多対多）を正しく導出できるか。
- 参照関係（FK）を結ぶ際、親テーブルの主キーを外部キーとして子テーブルに含め、E-R図上の表記ルール（破線下線など）に従って正しく図示できるか。
- **2. トランザクションと同時実行制御**
- 複数処理の同時実行において、読み取りから更新までの間に割り込み（Interleaving）が発生した場合の不整合メカニズムを視覚的にイメージできるか。
- 割当超過や重複登録を防ぐため、適切なタイミングで専有ロック（FOR UPDATE 等）を適用するロジックを設計できるか。
- **3. SQL構文と集合操作**
- 複数のSELECT結果を統合する際、重複排除が必要なケース（UNION）と不要なケース（UNION ALL）の性能・機能トレードオフを理解しているか。
- INNER JOIN等における結合条件（ON 句）で、指示された相関名（テーブル別名）を正しく付与して記述できるか。
- **4. 業務要件の拡張とデータモデル変化**
- 既存の認証体系やテーブル構造を変更せず、ブリッジエンティティ（中間テーブル）を追加することでID連携やUX改善を実現するデータモデル設計パターンを適用できるか。