

【復習レポート】令和4年度秋期 応用情報技術者試験 問6「スマートデバイス管理システムのデータベース設計」徹底解説

1. システム刷新のビジネス背景とデータベース要件の構造化

自動車メーカーJ社は、グループ連結で約3万人の従業員を抱える大企業です。従来、J社では事業継続性（BCP）や災害時対応の一環として、本社の部長職以上の特定少数にのみスマートデバイス（情報端末）を貸与していました。そのため、単一のワークシートによる表計算ソフト（スマートデバイス管理台帳）で運用・管理を行うことができていました。しかし、昨今の働き方改革の推進に伴い、「全従業員がいつでもどこでも作業できる環境」の実現を目指し、グループ企業を含む全従業員約3万人へ配布対象を拡大する方針が決定されました。

管理対象が数十人規模から3万人超へと跳ね上がるにあたり、表計算ソフトによる一元管理は限界を迎えます。データ整合性の破綻、同時並行での更新に伴う競合問題、アクセス権限の不備による情報セキュリティリスクなど、致命的な運用上の障害が発生することは明白です。このビジネス変革を支える強固なIT基盤として、関係データベース（RDB）による「スマートデバイス管理システム」の構築が不可欠となりました。

現場ヒアリング（表2）で判明した4つの課題と新システムの要件、およびそれがデータベース（DB）設計（エンティティ抽出・制約設定・アクセス制御）にどのように直接反映されるのか、その設計上の意図を構造化したマッピングは以下の通りです。

課題と新システム要件・DB設計への反映マッピング

項番	現状の課題	新システムの要件	データベース（DB）設計上の反映・意図
1	利用者が個別に通信事業者や料金プランを選択できるため、高額なプランが契約されるケースがある。	通信事業者を原則としてL社に統一し、低価格な料金プランで契約できるようにする。	【初期値設定・リレーションシップ】 料金プラン表の「通信事業者コード」に初期値としてL社のコード（'1234'）を設定。通信事業者・料金プラン・契約間のリレーションシップを正確にモデル化。
2	費用を本社総務部が一括負担しており、コスト意識が低く高額請求が発生	従業員の異動情報に基づき、請求を年月ごと、部署ごとに管理できるようにする。	【エンティティ・属性追加と追跡チェーン】 「請求」エンティティに「年月」属性を追加。「部署」「従業員」「利用」「情報端末」「契約」「請求」の各エンティティがリレーションシップで連鎖（チェ

生ずる。

ーン) 接続されることで、**「誰が・どの端末で・いつ・どの部署にいくら請求されるか」の全追跡(トレーサビリティ) **を可能にする。

アプリ追加や機能 業務上必要な機能やアプリ制限解除の依頼が りについて、従業員と情報端末の組合せ(利用) 3 あっても、管理が 報端末の組合せ(利用) 煩雑になるため認 ごとに個別許可できる仕組みにしない。 組みにする。

【交差エンティティとカーディナリティ】

従業員と情報端末の紐づけを表す「利用」エンティティを定義し、1つの「利用」に対して複数の「アプリ追加」や「機能制限解除」を許容する「1対多」のリレーションシップを設定。

運用管理担当者が 全契約の暗証番号 4 を参照できるため、不正な契約変更リスクが残る。

暗証番号は運用管理担当者の上長(ADMIN)しか参照できないようアクセス制御を行う。

【DCLによる物理的アクセス制御】

データ制御言語(GRANT文)を用い、契約表の「契約ID」および「暗証番号」の参照(SELECT)権限をADMINロール(上長アカウント)のみに限定付与。人為的運用に頼らず、DBMSレベルで物理的に不正リスクを遮断する。

このように、現場の業務ルールやビジネス課題は、DB設計における「エンティティの定義」「外部キーによるリレーションシップ・チェーンの構築」「属性の保持」「権限管理(アクセス制御)」へとダイレクトに変換されます。次セクションでは、これらの要件がE-R図(Entity-Relationship Diagram)上でどのように視覚的・論理的に表現されるのかを徹底解剖します。

2. 設問1 徹底解説：E-R図の解説とリレーションシップ・属性の特定

E-R図は、現実世界の業務対象(実体=エンティティ)と、それらの間に存在する関係性(リレーションシップ)を可視化する論理設計の根幹です。システム全体でどのようなデータが保持され、業務ルールがどうデータ構造として固定化されているかをブレなく把握するために、DB設計において最も重要な設計書となります。

E-R図の空欄[a]～[f]について、問題文やヒアリング結果から解を導き出す思考プロセスをステップバイステップで解説します。

【図1 新システムのE-R図(抜粋)における該当箇所】

- ・ 請求エンティティ内の空欄[a]
- ・ 契約と情報端末間のリレーションシップ[b]
- ・ 利用エンティティ内の外部キー属性[c],[d]

- ・利用とアプリ追加間のリレーションシップ [e]
- ・部署エンティティの自己参照リレーションシップ [f]

各空欄の論理的導出プロセス

空欄 [a] (請求エンティティの属性)

- ・ **論理的思考ステップ:** 表2の項番2の要件には「請求を年月ごと、部署ごとに管理できるようにする」と明記されています。図1の「請求」エンティティを確認すると、主キーとして「請求番号」「契約ID」、属性として「請求先部署ID」が定義されていますが、「年月」に関する属性が存在しません。したがって、請求を年月単位で特定・管理するために不可欠な属性は**「年月」** (または「請求年月」) となります。

空欄 [b] (「契約」と「情報端末」のリレーションシップ)

- ・ **論理的思考ステップ:** 表1の「交換予定日」の説明において、「同一の回線番号のままで2年ごとに旧情報端末から新情報端末への交換を行っており」と記載されています。
- ・ **【マスター・インストラクターのワンポイントアドバイス：初学者がハマる思考トラップ】** 初学者から非常によく出る質問が「今現在、従業員が手に持っている端末は1台なのだから、契約と端末は1対1ではないのですか？」という疑問です。ここが最大の罠です。「現在の一時点の断面」だけを見るのではなく、「2年ごとの端末交換という時間の経過（履歴）」に着目することが1対多を見抜く鍵です。回線番号(=1つの契約)は変わらないまま、2年ごとに端末が「旧端末→新端末→さらに次の端末…」と更新されていきます。つまり、時間の経過に伴い、**1つの「契約」に対して複数の「情報端末」の履歴が紐づく(1対多)** **関係となります。
- ・ **カーディナリティ(矢印)の決定:** 「契約」側が「1」、「情報端末」側が「多」となるため、矢印は「契約」から「情報端末」に向かう上向きの矢印**「↑」**となります。

空欄 [c], [d] (「利用」エンティティの属性)

- ・ **論理的思考ステップ:** 表2の項番3の要件に「従業員と情報端末の組合せごとに個別に許可できる仕組みにする」とあります。「利用」エンティティは、特定の「従業員」が特定の「情報端末」を一定期間利用するという関連(交差エンティティ)を表します。
- ・ 1つの「利用」を成立させるためには、親エンティティである「従業員」の主キーと「情報端末」の主キーをそれぞれ外部キーとして保持する必要があります。したがって、空欄には**「従業員ID」と「情報端末ID」**が入ります(順不同)。
- ・ **【実践的アドバイス：表記の厳密性について】** IPAの公式解答および問題文図1の属性リストでは、「従業員」と「ID」、「情報端末」と「ID」の間に半角スペースが含まれた 従業員ID / 情報端末ID と表記されています。試験本番ではスペースを詰めて書いても正解とされるケー

が多いですが、**「問題文の凡例や属性リストの表記と 100%完全に一致させて記述する」**
のが最も確実で安全な解答法です。

空欄 [e] (「利用」と「アプリ追加」のリレーションシップ)

- **論理的思考ステップ:** 表 2 の項番 3 の要件「従業員と情報端末の組合せごとに個別に許可できる仕組みにする」および「複数のアプリが追加できる」という記述に着目します。「利用 (特定の従業員と端末のペア)」1 件に対して、複数の「アプリ追加」申請が紐づくため、**「利用」が「1」、「アプリ追加」が「多」**のリレーションシップが成り立ちます。
- **カーディナリティ (矢印) の決定:** 「利用」から「アプリ追加」に向かう下向きの矢印**「↓」**が入ります。

空欄 [f] (「部署」の自己参照リレーションシップ)

- **論理的思考ステップ:** 問題文に「J 社内の部署の階層構造は、自己参照の関連を用いて表現する」と指示されています。「部署」エンティティには主キー「部署 ID」のほかに、親組織を指す外部キー「上位部署 ID」が定義されています。1 つの上位部署に対して複数の下位部署が存在するため、「部署」エンティティが自分自身を参照する 1 対多の自己参照モデルとなります。
- **【解答用紙への正しい描画方法】** 図 1 の凡例および公式解答に従い、**「『部署』エンティティの枠線の右側 (または下部) から出て、自身に戻る直角の折り返し線 (矢印付きの『コ』の字型記号)」**として描く必要があります (解答欄の記号としては コ またはループ矢印の形状となります)。

設問 1 解答まとめ一覧表

空欄	解答例	解答導出の要点と実戦アプローチ
a	年月 (または 請求年月)	ヒアリング結果「請求を年月ごと、部署ごとに管理できる」より導出。
b	↑	2 年ごとの端末交換という「時間の経過」により、1 契約(1)に対しいくつもの情報端末(多)が存在。
c	従業員 ID (d と順不同)	従業員と情報端末の組合せ (利用) を特定するための外部キー。スペース表記に配慮。
d	情報端末 ID (c と順不同)	同上。情報端末エンティティの主キーを参照する外部キー。

e	↓	1つの利用(1)に対して複数のアプリ追加(多)を個別に許可する1対多関係。
f	コ(直角折り返しの自己参照記号)	部署の階層構造(上位部署と下位部署)を表現する自己参照リレーションシップ。

3. 設問2 徹底解説：物理設計とデータベース制約（PK・UK・非 NULL）の適用

データベースの物理設計（表定義）において、不整合なデータ（重複データや実体のない空データ）の混入を防ぎ、データの高度な完全性（整合性）を担保するために設定されるのが「データベース制約」です。代表的な3つの制約の意味と役割を以下に整理します。

- **主キー制約（PK: Primary Key）**：表内のレコード（行）を一意に識別するための最重要制約。重複値および空値（NULL）を一切許容しない。
- **UNIQUE 制約（UK: Unique Key）**：主キー以外の列において、含まれる値が一意（重複がない）であることを保証する制約。
- **非 NULL 制約（NOT NULL / 非 NULL）**：列に NULL（値が存在しない状態）が入ることを禁止し、データ入力を必須化する制約。

問題文の注記ルールと論理的判定プロセス

設問2では、表3「契約表の表定義」および表4「料金プラン表の表定義」の1行目における空欄〔g〕（PK）、〔h〕（UK）、〔i〕（非 NULL）に入る記号の組み合わせを判定します。

【表3および表4の1行目（対象列：契約 ID / 料金プランコード）】

列名：契約 ID / 料金プランコード

データ型：CHAR(8)

PK：〔g〕 | UK：〔h〕 | 非 NULL：〔i〕

判定にあたっては、問題文直前に与えられた**注記ルール（指示事項）**を厳格に適用します。

1. PK（主キー制約）の判定〔g〕:

- 「契約 ID」および「料金プランコード」は、それぞれの表における主キーです。したがって、PK 欄は**「Y」**となります。

2. UK（UNIQUE 制約）の判定〔h〕:

- 問題文の注記に「主キーに対しては UNIQUE 制約を指定せず」と明記されています。

したがって、UK 欄は**「N」**となります。

3. 非 NULL 制約の判定 [i]:

- 。 問題文の注記に「非 NULL 制約は指定するものとする」と明記されています。したがって、非 NULL 欄は**「Y」**となります。

【マスター・インストラクターの専門的背景解説：なぜ主キーに UK を指定しないのか？】

実務や学術的なデータベース理論において、**「主キー（PK）は一意性（UNIQUE）と非 NULL（NOT NULL）の性質をあらかじめ内包している」**という大原則が存在します。標準 SQL や多くの DBMS 実装において、列を PRIMARY KEY に指定した時点で内部的に一意性と非 NULL の制約が自動付与されます。

そのため、DBMS によっては PK に対してさらに UK 制約を重ねて指定することは「冗長（重複指定）」とみなされます。本試験の問記ルールは、この RDB 理論上の原則に厳格に沿った仕様となっているのです。単にルールを暗記するだけでなく、こうした標準 SQL の理論的背景を知っておくと、実務や上位試験でも迷いがなくなります。

解答群との照合

導き出された指定の組み合わせは **g = Y, h = N, i = Y**（順に Y, N, Y）となります。

解答群（アイウエオカ）と照合します。

記号 g h i 判定

ア N N N ×

イ N N Y ×

ウ N Y N ×

エ N Y Y ×

オ Y N Y ○（正解）

カ Y Y Y ×

設問 2 正解：オ

4. 設問 3 徹底解説：SQL によるデータ制御言語（DCL）とデータ定義言語（DDL）の実装

論理設計および物理設計で定義されたテーブル構造やセキュリティ要件は、SQL (Structured Query Language) 文を用いて実際の DBMS へ構築・実装されます。ここでは、アクセス権限を制御するデータ制御言語 (DCL: Data Control Language) と、テーブルの作成・構造定義を行うデータ定義言語 (DDL: Data Definition Language) の文法・記述力が問われています。

各空欄の文法解剖と解答導出

空欄 [j] (GRANT 文によるアクセス制御)

- **要件とビジネスの意味:** 表 2 の項番 4「暗証番号は運用管理担当者の上長 (ADMIN) しか参照できないようにアクセスを制御する」および表 3 のアクセス制御欄に基づきます。運用管理担当者から直接参照権限を剥奪し、上長アカウント (ADMIN) のみに限定参照を許可することで、**人為的なコンプライアンス運用に頼らず、DBMS レベルで不正な契約変更リスク (課題 4) を物理的に遮断**しています。
- **DCL 構文解剖:** GRANT 文を用いて特定表の「特定の列のみ」の参照権限を与える構文は以下の通りです。 $\text{\$}\text{\$}\text{\$}\text{GRANT SELECT (列名 1, 列名 2, \cdots) ON 表名 TO ユーザー名}\text{\$}\text{\$}\text{\$}$
- **記述の導出:** 図 2 の構文 GRANT [j] ON 契約 TO ADMIN において、「契約」表の「契約 ID」列と「暗証番号」列の参照権限を与えるため、[j] には **SELECT (契約 ID, 暗証番号)** が入ります (列リストをカッコ () で括り、カンマで区切る点に注意してください)。

空欄 [k] (CREATE TABLE 文の列定義・初期値・制約)

- **要件の確認:** 表 4「料金プラン表の表定義」の「通信事業者コード」列を確認すると、データ型は CHAR(4)、非 NULL 制約は Y (NOT NULL)、初期値は 1234 と指定されています。また、「その他の指定内容」に「行挿入時に、初期値として L 社の通信事業者コード '1234' を設定する」と記載されています。
- **DDL 構文解剖:** 列定義の中で初期値を設定する DEFAULT 句 と、NULL を禁止する NOT NULL 句 を組み合わせます。文字型 (CHAR) の初期値指定では、値をシングルクォーテーション '1234' で囲むことが必須です。
- **記述の導出:** IPA 公式解答の指定は **CHAR(4) DEFAULT '1234' NOT NULL** です。 ※なお、SQL の文法上、*CHAR(4) NOT NULL DEFAULT '1234'* のように *NOT NULL* と *DEFAULT* の記述順序が前後しても正当な SQL として機能します。本試験解説テキスト等でも双方の順序が提示されていますが、公式解答の順序で書くのが最も確実です。

空欄 [l] (主キー制約キーワード)

- **DDL 構文解剖:** CREATE TABLE 文の文末で表制約として主キーを指定する定型構文です。 $\text{\$}\text{\$}\text{\$}\text{PRIMARY KEY (列名)}\text{\$}\text{\$}\text{\$}$

- **記述の導出:** 図3の [1] (料金プランコード) より、[1] には **PRIMARY KEY** が入ります。

空欄 [m] (外部キー制約キーワード)

- **DDL 構文解剖:** 他の表を参照して親子の参照整合性を保持する外部キー制約を指定する定型構文です。 `$$$text{FOREIGN KEY (列名) REFERENCES 参照先表名(参照先列名)}$$`
- **記述の導出:** 図3の [m] (通信事業者コード) REFERENCES 通信事業者(通信事業者コード) より、[m] には **FOREIGN KEY** が入ります。

設問3 解答まとめ一覧表

空欄	解答例	構文文法と設計上のポイント
j	SELECT (契約 ID, 暗証番号)	カッコ内に参照を許可する複数列名をカンマ区切りで指定。上長のみに開示し物理セキュリティを確保。
k	CHAR(4) DEFAULT '1234' NOT NULL (または <i>CHAR(4) NOT NULL DEFAULT '1234'</i>)	データ型、初期値 (シングルクォーテーション必須)、非 NULL 制約を指定。記述順序の自由度も配慮。
l	PRIMARY KEY	表の主キー制約を宣言する標準 DDL キーワード。
m	FOREIGN KEY	他表 (通信事業者表) との参照整合性を維持する外部キー制約キーワード。

5. 総括：初心者向け・DB 設計問題を攻略するための思考プロセス

応用情報技術者試験のデータベース分野は、一見すると専門用語や SQL 構文が並び、難解な印象を受けるかもしれませんが、しかし、本質を解き明かすと**「問題文に書かれた業務ルールに従って、データを論理的・物理的にきれいに整理整頓するパズル」**に過ぎません。

ルール通りにステップを踏めば、必ず1つの正解に収束する「最も安定して高得点が狙える得点源分野」なのです。

過去問演習や本番試験では、以下の**3ステップの思考プロセス**と**実戦チェックリスト**を頭に叩き込んで臨みましょう。

DB 設計問題を攻略するための思考プロセス 3 ステップ

[ステップ 1: 業務要件の抽出（ビジネス層）]

ヒアリング結果・問題文から「実体（何）」と「業務ルール（制約・条件）」を抽出し、課題との対応を整理する。

|



[ステップ 2: 論理構造化（概念設計層）]

E-R 図上でエンティティ間の関連（1 対 1、1 対多、多対多）を見抜く。

★「一時点の断面」ではなく「時間の経過（履修・更新履歴）」を考慮してカーディナリティを決める。

|



[ステップ 3: 物理実装（SQL・制約層）]

問題文の「注記ルール」を厳格に適用し、PK/UK/非 NULL 制約の設定、DCL/DDL 文の定型構文へ正確に落とし込む。

試験本番でケアレスミスゼロにする実戦チェックリスト

- ☐ 問題文の「注記」を 1 行残らず確認したか？
 - 設問 2 のように「主キーに対して UNIQUE 制約を指定するかどうか」などの注記指示は、一般的な RDB の慣習よりも**問題文の指示が 100%優先**されます。
- ☐ 時間の経過（履歴）に伴う「1 対多」を見落としていないか？
 - 「2 年ごとの端末交換」「異動に伴う部署変更」など、時間の経過によって 1 つの契約・従業員に対して複数のデータが発生しないかを必ず疑いましょう。
- ☐ 属性名や列名の「スペース表記」を問題文と一致させたか？
 - 従業員 ID や 契約 ID など、問題文や凡例に半角スペースが含まれている場合は、そのまま完全に真似て記述するのが最も安全です。
- ☐ SQL 文のシングルクォーテーションやカッコを忘れていないか？
 - DEFAULT '1234' のような文字列初期値の囲み忘れや、GRANT SELECT (列名 1, 列名 2) のカッコ括り漏れがないか、記述後に必ずセルフチェックしましょう。

