

令和6年度 秋期  
**データベーススペシャリスト試験**  
**午後Ⅰ 問題**

試験時間

12:30 ~ 14:00 (1 時間 30 分)

**注意事項**

- 試験開始及び終了は、監督員の時計が基準です。監督員の指示に従ってください。
- 試験開始の合図があるまで、問題冊子を開いて中を見てはいけません。
- 答案用紙への受験番号などの記入は、試験開始の合図があつてから始めてください。
- 問題は、次の表に従って解答してください。

問題番号	問1～問3
選択方法	2問選択

- 答案用紙の記入に当たっては、次の指示に従ってください。
  - B又はHBの黒鉛筆又はシャープペンシルを使用してください。
  - 受験番号欄に受験番号を、生年月日欄に受験票の生年月日を記入してください。  
 正しく記入されていない場合は、採点されないことがあります。生年月日欄については、受験票の生年月日を訂正した場合でも、訂正前の生年月日を記入してください。
  - 選択した問題については、次の例に従って、選択欄の問題番号を○印で囲んでください。○印がない場合は、採点されません。3問とも○印で囲んだ場合は、はじめの2問について採点します。
  - 解答は、問題番号ごとに指定された枠内に記入してください。
  - 解答は、丁寧な字ではっきりと書いてください。読みにくい場合は、減点の対象になります。

[問1、問3を選択した場合の例]

選択欄	
2	問1
問選択	問2
	問3

注意事項は問題冊子の裏表紙に続きます。  
 こちら側から裏返して、必ず読んでください。



## 問題文中で共通に使用される表記ルール

概念データモデル、関係スキーマ、関係データベースのテーブル（表）構造の表記ルールを次に示す。各問題文中に注記がない限り、この表記ルールが適用されているものとする。

### 1. 概念データモデルの表記ルール

(1) エンティティタイプとリレーションシップの表記ルールを、図1に示す。

- ① エンティティタイプは、長方形で表し、長方形の中にエンティティタイプ名を記入する。
- ② リレーションシップは、エンティティタイプ間に引かれた線で表す。  
“1対1”のリレーションシップを表す線は、矢を付けない。  
“1対多”のリレーションシップを表す線は、“多”側の端に矢を付ける。  
“多対多”のリレーションシップを表す線は、両端に矢を付ける。

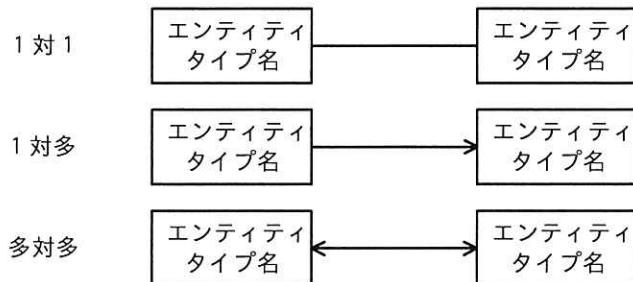


図1 エンティティタイプとリレーションシップの表記ルール

(2) リレーションシップを表す線で結ばれたエンティティタイプ間において、対応関係にゼロを含むか否かを区別して表現する場合の表記ルールを、図2に示す。

- ① 一方のエンティティタイプのインスタンスから見て、他方のエンティティタイプに対応するインスタンスが存在しないことがある場合は、リレーションシップを表す線の対応先側に“○”を付ける。
- ② 一方のエンティティタイプのインスタンスから見て、他方のエンティティタイプに対応するインスタンスが必ず存在する場合は、リレーションシップを表す線の対応先側に“●”を付ける。

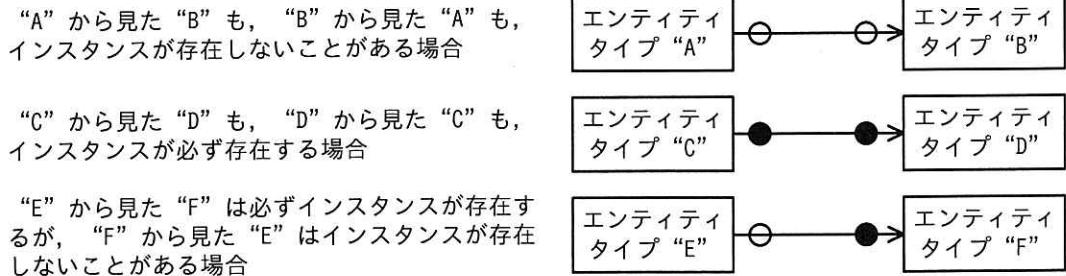
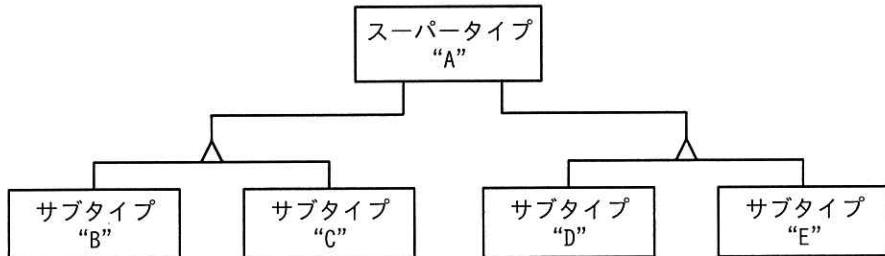


図2 対応関係にゼロを含むか否かを区別して表現する場合の表記ルール

(3) スーパータイプとサブタイプの間のリレーションシップの表記ルールを、図3に示す。

- ① サブタイプの切り口の単位に“△”を記入し、スーパータイプから“△”に1本の線を引く。
- ② 一つのスーパータイプにサブタイプの切り口が複数ある場合は、切り口の単位ごとに“△”を記入し、スーパータイプからそれぞれの“△”に別の線を引く。
- ③ 切り口を表す“△”から、その切り口で分類されるサブタイプのそれぞれに線を引く。



スーパー タイプ “A”に二つの切り口があり、それぞれの切り口にサブ タイプ “B”と“C”及び“D”と“E”がある例

図3 スーパータイプとサブタイプの間のリレーションシップの表記ルール

(4) エンティティタイプの属性の表記ルールを、図4に示す。

- ① エンティティタイプの長方形内を上下2段に分割し、上段にエンティティタイプ名、下段に属性名の並びを記入する。<sup>1)</sup>
- ② 主キーを表す場合は、主キーを構成する属性名又は属性名の組に実線の下線を付ける。
- ③ 外部キーを表す場合は、外部キーを構成する属性名又は属性名の組に破線の下線を付ける。ただし、主キーを構成する属性の組の一部が外部キーを構成する場合は、

破線の下線を付けない。

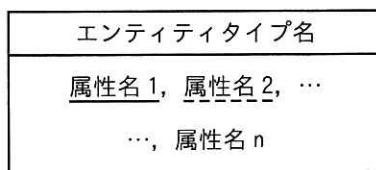


図 4 エンティティタイプの属性の表記ルール

## 2. 関係スキーマの表記ルール及び関係データベースのテーブル（表）構造の表記ルール

(1) 関係スキーマの表記ルールを、図 5 に示す。

関係名 (属性名 1, 属性名 2, 属性名 3, …, 属性名 n)

図 5 関係スキーマの表記ルール

- ① 関係を、関係名とその右側の括弧でくくった属性名の並びで表す。<sup>1)</sup> これを関係スキーマと呼ぶ。
- ② 主キーを表す場合は、主キーを構成する属性名又は属性名の組に実線の下線を付ける。
- ③ 外部キーを表す場合は、外部キーを構成する属性名又は属性名の組に破線の下線を付ける。ただし、主キーを構成する属性の組の一部が外部キーを構成する場合は、破線の下線を付けない。

(2) 関係データベースのテーブル（表）構造の表記ルールを、図 6 に示す。

テーブル名 (列名 1, 列名 2, 列名 3, …, 列名 n)

図 6 関係データベースのテーブル（表）構造の表記ルール

関係データベースのテーブル（表）構造の表記ルールは、(1) の ① ~ ③ で “関係名” を “テーブル名” に、 “属性名” を “列名” に置き換えたものである。

---

注<sup>1)</sup> 属性名と属性名の間は “,” で区切る。

[ × 用 紙 ]

問1 オンライン学習プラットフォームの概念データモデリングに関する次の記述を読んで、設問に答えよ。

教育事業者のA社は、オンライン学習プラットフォーム（以下、PFという）を運営している。このたび、新規要件の追加に伴い、データベースを再設計した。

[現行業務]

1. アカウント

- (1) PFを利用する個人をアカウントと呼び、アカウント番号（以下、番号を#で示す）で識別する。
- (2) PF上で学習教材を受講するアカウントを受講生と呼ぶ。受講生としてアカウントを登録する際、学習教材を購入するための支払情報を登録する。後述する講師としてのアカウントを既に保持している場合、再度アカウントを登録する必要はなく、支払情報を追加で登録する。
- (3) PF上で受講生に提供する学習教材を制作するアカウントを講師と呼ぶ。講師としてアカウントを登録する際、自身のプロフィールを登録する。受講生としてのアカウントを既に保持している場合、再度アカウントを登録する必要はなく、自身のプロフィールを追加で登録する。

2. コース

- (1) コースは、講師がPF上で受講生に提供する一連の学習教材を指し、コース#で識別する。各コースの制作は1名の講師が行い、制作した講師が分かるようになっている。全て有料であり、標準価格を設定している。
- (2) コースは、複数のセクションから成る。セクションは、コースごとにセクション#で識別する。
- (3) セクションは、一つ又は複数の講義動画と、講義動画を視聴した後の理解度を確認する高々一つのテストから成る。講義動画とテストとを併せてコンテンツと呼び、セクションごとにコンテンツ#で識別する。
  - ・テストは、複数の設問から成る。設問は、テストごとに設問#で識別する。
  - ・設問は、全て単一選択式であり、設問ごとに設問文と配点を設定する。
  - ・設問の各選択肢は、設問ごとに選択肢#で識別し、選択文と正解フラグを設

定する。

### 3. コース購入

- (1) 受講生は、コースを検索して、受講したいコースを購入する。コースの購入は、コース購入#で識別する。購入情報（購入年月日、購入価格及び受講生アカウント#）は、PF 上に記録される。
- (2) 受講生は、PF 上で利用できるクーポンを使用すると、標準価格より安くコースを購入できる。クーポンを使用する際は、クーポン#を指定して、購入の手続を行う。クーポンの併用はできない。
  - ・クーポンは、講師が自身のコースの販売促進を目的に発行している。
  - ・クーポンは、クーポン#で識別し、割引率や値引額、クーポンの適用期間及び同一クーポンの使用可能人数をもつ。使用可能人数は最大 1,000 名であり、クーポン使用者数が使用可能人数の上限に達すると、適用期間にかかわらず、クーポンの使用は不可となる。
- (3) 受講生は、お薦めしたいコースがある場合、アカウントをもっている相手にコースをプレゼントすることができる。受講生がコースをプレゼントするには、相手のアカウントと対象のコースをそれぞれ一つ指定して購入の手続を行う。

### 4. コース受講

- (1) 受講生は、購入したコース及びプレゼントされたコースを、いつでも受講することができる。
  - ① コンテンツの受講の完了は、講義動画については視聴を終えた時点、テストについては全設問に解答した時点である。セクション内の全コンテンツの受講が完了するとセクションの受講が完了となり、全セクションの受講が完了するとコースの受講が完了となる。
  - ② 自己研鑽<sup>さん</sup>が目的なので、どのコンテンツから受講してもよい。テストに合格基準点ではなく、どのコンテンツも繰り返し受講することができる。
  - ③ コースの受講中又は完了後、5段階評価と感想を登録することができる。
  - ④ 受講生は、自分が最後に実施したテストで選択した各設問の選択肢と解答日時を確認することができる。また、自分が過去に実施したテストについては、最高獲得点数と、全設問の合計点数の履歴を確認できる。
- (2) 受講生は、コンテンツの内容に疑問があれば、PF 上で講師に質問することができる。

できる。各質問に対する講師からのコメントは1回だけできる。

## 5. 購入実績及び受講の確認

講師は PF 上で、自分が制作したコースを購入した受講生の購入情報、クーポンを使用した場合のクーポン#, 受講生による 5 段階評価及び感想を確認することができる。

### [企業向けサービスの追加]

PF に、次のような企業向けのサービスを追加することにした。

#### 1. 受講生の登録

企業は、研修対象の従業員を受講生としてアカウントを登録する。これを法人受講生と呼び、現行業務の受講生とは区別する。既に個人でアカウントを保持していても、法人用のアカウント#を新たに付与する。

#### 2. 研修カリキュラムの登録

(1) 企業は、受講させる一つ以上のコースをまとめた研修カリキュラムを作成する。研修カリキュラムは、企業ごとに研修カリキュラム#で識別し、研修カリキュラム名をもつ。研修カリキュラムに含まれるコースには、受講順を設定する。

(2) 企業は、法人受講生に対して一つ以上の研修カリキュラムを受講させる。

(3) 研修カリキュラムへの法人受講生の割当てでは、あらかじめ複数の法人受講生を束ねたグループを作成し、グループ単位で割当てを行う。グループは、企業ごとにグループ#で識別し、グループ名をもつ。また、同じ研修カリキュラムでも、割り当てられるグループによって、受講開始可能年月日及び受講完了期限年月日が異なる場合がある。

(4) テストを含むコースについては、研修カリキュラムごとに各テストの合格基準点を設定できる。

#### 3. 法人受講生によるコース受講

(1) 法人受講生は、指定された研修カリキュラムを期限までに受講する。受講を開始すると、研修カリキュラム受講開始日時が記録される。全てのコースの受講が完了すると、研修カリキュラムが完了となり、研修カリキュラム受講完了日時が記録される。

(2) 現行業務のコース受講と同じサービスを提供するが、企業が合格基準点を定

めているテストについては、合格基準点に達しないとテストの受講は完了とならない。

#### [受講生からの要望]

受講生から、学習に対する理解を深めるために、次の要望があった。

- ・コースを提供する講師だけではなく、他の講師及び受講生も質問に対して回数に制限なくコメントできるようにしてほしい。
- ・その際、誰がコメントしたかを特定できるようにしてほしい。
- ・質問だけではなく、既出のコメントを指定してコメントできるようにしてほしい。

#### [概念データモデルと関係スキーマの設計]

##### 1. [現行業務]に基づく設計

現行業務の概念データモデルを図1に、関係スキーマを図2に示す。

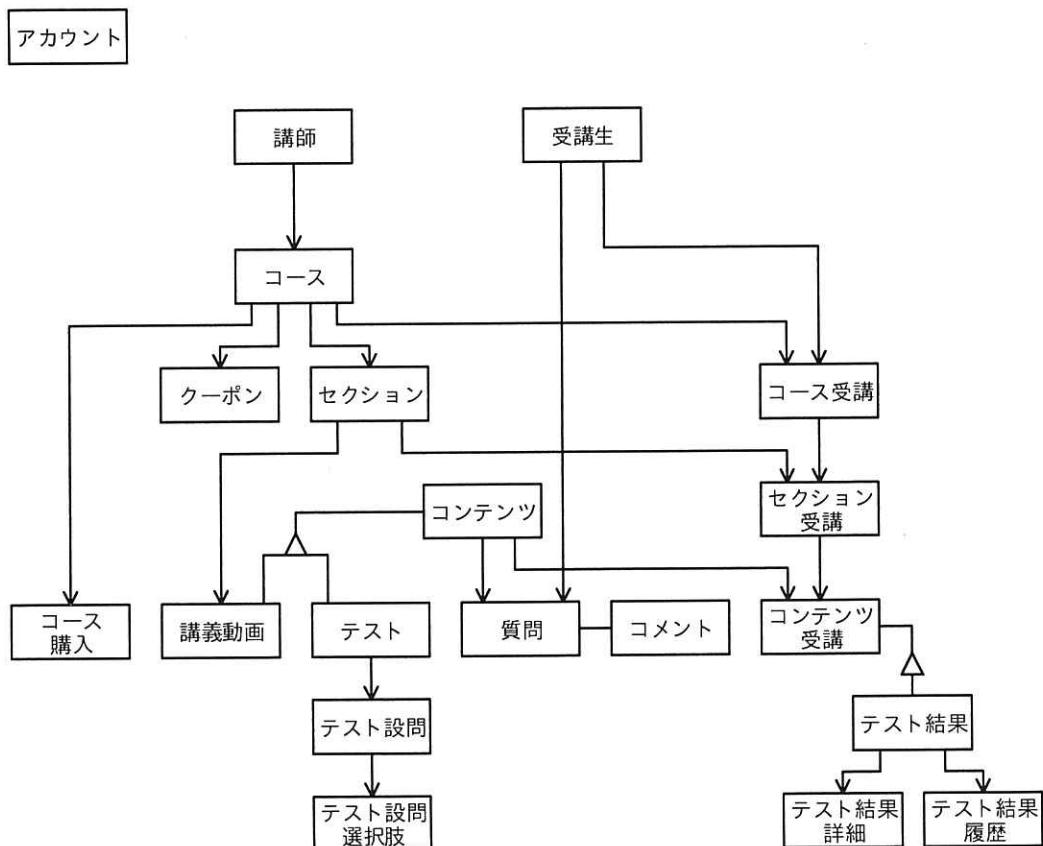


図1 現行業務の概念データモデル（未完成）

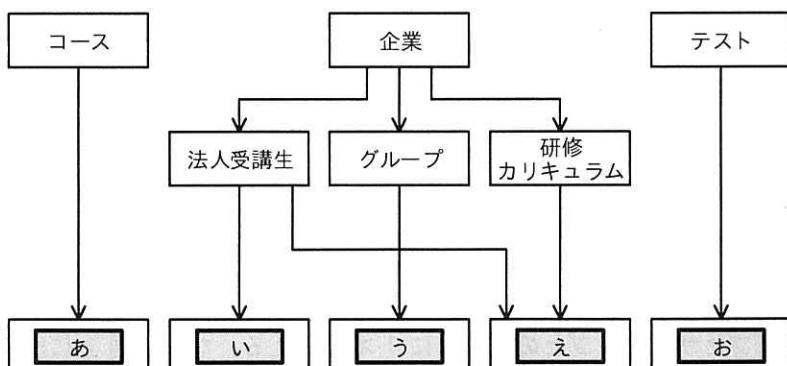
アカウント（アカウント#, アカウント名, [ ] ）  
 講師（アカウント#, 講師プロフィール）  
 受講生（アカウント#, 支払情報）  
 コース（コース#, アカウント#, コース名, 標準価格）  
 セクション（コース#, セクション#, セクション名）  
 コンテンツ（コース#, セクション#, コンテンツ#, コンテンツ名, コンテンツ区分）  
 講義動画（コース#, セクション#, コンテンツ#, 講義時間）  
 テスト（コース#, セクション#, コンテンツ#, 解答制限時間）  
 テスト設問（コース#, セクション#, コンテンツ#, 設問#, 設問文, 配点）  
 テスト設問選択肢（コース#, セクション#, コンテンツ#, 設問#, 選択肢#, 選択文, 正解フラグ）  
 クーポン（クーポン#, コース#, 割引率, 値引額, 適用開始年月日, 適用終了年月日, 使用可能人数, クーポン発行年月日）  
 コース購入（コース購入#, コース#, 購入年月日, 購入価格, [ a ] ）  
 コース受講（アカウント#, コース#, コース受講開始日時, コース受講完了日時, 5段階評価, 感想）  
 セクション受講（アカウント#, コース#, セクション#, セクション受講開始日時, セクション受講完了日時）  
 コンテンツ受講（アカウント#, コース#, セクション#, コンテンツ#, コンテンツ受講開始日時, コンテンツ受講完了日時）  
 テスト結果（アカウント#, コース#, セクション#, コンテンツ#, 最高獲得点数）  
 テスト結果詳細（アカウント#, コース#, セクション#, コンテンツ#, [ b ] , 解答日時）  
 テスト結果履歴（アカウント#, コース#, セクション#, コンテンツ#, テスト実施日時, 合計点数）  
 質問（質問#, コース#, セクション#, コンテンツ#, アカウント#, 質問タイトル, 質問文, 質問日時）  
 コメント（質問#, コメント文, コメント日時）

注記 設問の都合上、網掛け部分は表示していない。

図2 現行業務の関係スキーマ（未完成）

## 2. [企業向けサービスの追加]に基づく設計

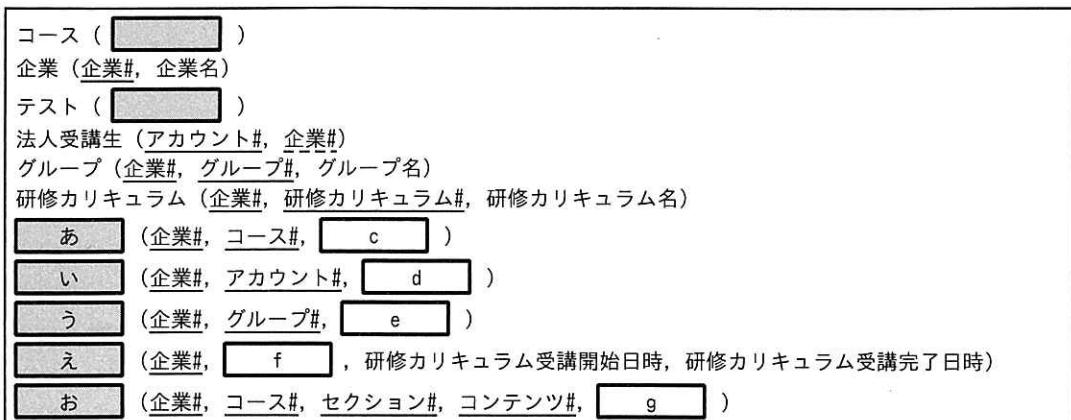
企業向けサービスの追加部分の概念データモデルを図3に、関係スキーマを図4に示す。



注記1 コースとテストのエンティティタイプは、図1中のものと同じである。

注記2 設問の都合上、網掛け部分は表示していない。

図3 企業向けサービスの追加部分の概念データモデル（未完成）



注記1 コースとテストの網掛け部分は、図2と同じ属性である。

注記2 図中の [ ] ～ [ ] には、図3の [ ] ～ [ ] と同じ字句が入る。

図4 企業向けサービスの追加部分の関係スキーマ（未完成）

解答に当たっては、巻頭の表記ルールに従うこと。ただし、エンティティタイプ間の対応関係にゼロを含むか否かの表記は必要ない。エンティティタイプ間のリレーションシップとして“多対多”的リレーションシップを用いないこと。サブタイプが存在する場合、他のエンティティタイプとのリレーションシップは、スーパー・タイプ又はいずれかのサブタイプの適切な方との間に設定すること。同一のエンティティタイプ間に異なる役割をもつ複数のリレーションシップが存在する場合、役割の数だけリレーションシップを表す線を記述すること。属性名は、意味を識別できる適切な名称とすること。関係スキーマに入れる属性を答える場合、主キーを表す下線、外部キーを表す破線の下線についても答えること。

設問1 【現行業務】の概念データモデル及び関係スキーマについて答えよ。

- (1) 図1の概念データモデルは未完成である。欠落しているリレーションシップを7本補って図を完成させよ。スーパー・タイプとサブ・タイプとの間のリレーションシップの本数は、サブ・タイプの切り口の数に関係なく、スーパー・タイプとサブ・タイプとの組ごとに1本と数える。なお、図1に表示されていないエンティティタイプは考慮しなくてよい。
- (2) 図2中の [ ] a と [ ] b に入れる一つ又は複数の適切な属性名を補って関係スキーマを完成させよ。

設問 2 [企業向けサービスの追加] の概念データモデル及び関係スキーマについて  
答えよ。

- (1) 図 3 の概念データモデルは未完成である。図 3 中の欠落しているリレーションシップを 4 本補って図を完成させよ。なお、図 3 に表示されていないエンティティタイプは考慮しなくてよい。
- (2) 図 4 中の  ~  に入れる一つ又は複数の適切な属性名を補って関係スキーマを完成させよ。

設問 3 [受講生からの要望] に対応するために、図 2 の関係スキーマの関係 “コメント” の主キーとして “コメント#” を新たに設け、“質問#” は外部キーへ変更した上で、次の変更を行う。

- (1) 図 1 の概念データモデル上のリレーションシップを一つ変更する。どのようにリレーションシップを変更すべきか。具体的なエンティティタイプ名を挙げ、40 字以内で答えよ。
- (2) 図 1 の概念データモデルに新しくリレーションシップを二つ追加する。どのようにリレーションシップを追加すべきか。具体的なエンティティタイプ名を挙げ、それぞれ 40 字以内で答えよ。

問2 総合商社の労務管理システムのデータベース実装に関する次の記述を読んで、設問に答えよ。

総合商社のY社は、労務管理にRDBMSを用いている。既存の勤怠管理機能に、新たに所在情報管理機能と監査機能を追加することになった。

#### [RDBMSの排他制御]

1. ロックは行単位で掛ける。共有ロックを掛けている間、他のトランザクションから対象行への共有ロックは可能であり、専有ロックは共有ロックの解放待ちとなる。専有ロックを掛けている間、他のトランザクションから対象行への共有ロック及び専有ロックは、専有ロックの解放待ちとなる。
2. 索引を使わずに表探索で全ての行に順次アクセスする場合、検索条件に合致するか否かにかかわらず全行をロックの対象とする。索引探索の場合、索引から読み込んだ行だけをロックの対象とする。

#### [所在情報管理機能の追加]

##### 1. 機能の概要

オフィス内のフリーアドレス化を推進するため、従業員の就業エリア（以下、エリアという）への入退室のログを記録し、従業員の所在情報をエリア情報表示端末で社内に公開する。エリア情報表示端末では、エリアの混雑状況なども表示する。

エリアには、建屋、フロア、執務室があり、従業員は各エリアへの入退室時にICカード社員証で認証を行う。従業員は入退室時に必ず認証を行うように定められており、他の従業員に続いて認証せずに入退室することは禁止されている。従業員は、建屋内にいる限りいずれかのエリアに所在しており、建屋から出ることで退出として扱われる。

所在情報管理機能では、認証時に入退室ログを記録する。また、記録した入退室ログを基に、従業員の現在位置を示す所在情報及びエリアの混雑状況を定期的に更新する。

## 2. テーブル構造

主なテーブルのテーブル構造は、図1のとおりである。主キーには主索引が定義されている。

従業員（従業員番号，氏名，生年月日，入社年月日，性別，…）
エリア状況（エリアコード，エリア名，座席数，在席者数，混雑判定閾値，…）
所在情報（従業員番号，最終入室エリアコード，…）
入退室ログ（ログID，従業員番号，エリアコード，入退室区分，入退室年月日，入退室時刻，…）

図1 主なテーブルのテーブル構造（一部省略）

## 3. 入退室ログ登録処理

従業員が入退室で認証する都度，“入退室ログ” テーブルに行を追加する。ログIDは時系列に昇順で採番し、入退室区分には入室（‘I’）又は退室（‘0’）を設定する。必要に応じて、連携する勤怠管理機能及びエリア情報表示端末に情報を送信する。

処理の概要を図2に示す。トランザクションの ISOLATION レベルは REPEATABLE READ とする。

- ① 入退室ログを挿入する。
- ② 入退室区分が入室の場合、当該従業員の所在情報を参照し、最終入室エリアコードが退出の状態であれば、勤怠管理機能に入室情報を送信する。入退室区分が退室の場合、何もしない。
- ③ 当該エリアの在席者数と混雑判定閾値を参照し、今回の入退室によって混雑判定閾値を上回る又は下回ることになるかを確認し、上回る又は下回る場合はエリア情報表示端末に混雑情報を送信する。
- ④ コミットする。

図2 入退室ログ登録処理の概要

## 4. 所在情報更新処理

5分ごとに実行するバッチ処理で、定期的にエリア状況及び所在情報を更新する。“入退室ログ” テーブルに登録されている入退室ログを読み込み、“エリア状況” テーブルのエリアごとの在席者数を更新，“所在情報” テーブルの従業員ごとの最終入室エリアコードを更新する（退出している場合は、最終入室エリアコードには ‘9999’ を設定する）。全ての更新が終わったら、読み込み終わった入退室ログ

の位置を記録しておき、次回実行時はその続きから読み込む。

所在情報更新処理の概要を図3、所在情報更新処理に用いるSQL文を表1に示す。

トランザクションのISOLATIONレベルはREPEATABLE READとする。

- ① SQL1を実行して最新の入退室ログのログIDを取得し、ホスト変数HNEWIDに設定する。  
なお、ホスト変数HLASTIDには、前回実行時のホスト変数HNEWIDの値が設定されている。
- ② SQL2を実行してエリアごとの在席者増減数（入室した従業員と退室した従業員の差）を取得する。SQL2の結果行ごとに、ホスト変数HAREACODEにエリアコードを、ホスト変数HDIFFに在席者増減数を設定し、SQL3を実行してエリア状況の在席者数を更新する。
- ③ SQL4を実行して従業員ごとの最新の入退室ログのログIDを取得する。SQL4の結果行ごとに、ホスト変数HEMPLOYEEIDに従業員番号を、ホスト変数HCURRENTIDにログIDを設定し、SQL5を実行して最終入室エリアのエリアコードを取得する。さらに、そのエリアコードをホスト変数HAREACODEに設定し、SQL6を実行して所在情報の最終入室エリアコードを更新する。
- ④ ホスト変数HLASTIDに、ホスト変数HNEWIDの値を設定する。
- ⑤ コミットする。

図3 所在情報更新処理の概要

表1 所在情報更新処理に用いるSQL文（未完成）

SQL	SQL文の構文
SQL1	SELECT MAX(ログID) FROM 入退室ログ
SQL2	SELECT エリアコード, [ a ] (CASE WHEN [ b ] THEN 1 ELSE 0 END) - [ a ] (CASE WHEN [ c ] THEN 1 ELSE 0 END) AS 在席者増減数 FROM 入退室ログ WHERE ログID > :HLASTID AND ログID <= :HNEWID GROUP BY エリアコード
SQL3	UPDATE エリア状況 SET 在席者数 = 在席者数 + :HDIFF WHERE エリアコード = :HAREACODE
SQL4	SELECT 従業員番号, [ d ] FROM 入退室ログ WHERE ログID > :HLASTID AND ログID <= :HNEWID GROUP BY 従業員番号
SQL5	SELECT CASE WHEN [ b ] THEN エリアコード ELSE '9999' END AS エリアコード FROM 入退室ログ WHERE ログID = :HCURRENTID
SQL6	UPDATE 所在情報 SET 最終入室エリアコード = :HAREACODE WHERE 従業員番号 = :HEMPLOYEEID

#### 〔監査機能の追加〕

従業員の残業時間、休暇、エリアを、入退室ログと突き合わせてチェックする監

査機能を、要件に基づいて設計した。

### 1. 監査機能の要件

- (1) 監査対象月の 1 日から月末日までの勤怠をチェックする。監査対象月の月末の営業日から翌月の 5 営業日までの間、毎日実行する。
- (2) 上長承認済み、かつ、監査未チェックの勤怠を対象にチェックする。なお、一度監査が終了した行でも、勤怠情報を従業員が更新した場合は、上長承認と監査結果をリセットして、再度、上長承認及び監査を行う。
- (3) 監査機能のジョブは多重実行する。全てのジョブに一意なジョブ ID を割り当て、それぞれのジョブに重複していない従業員番号の範囲を指定して実行する。また、処理の途中で失敗した場合、その原因を取り除いて、同じジョブ ID で再実行することで、処理を再開する。なお、ジョブ ID は再利用する。

### 2. 監査機能のテーブル設計

主なテーブルのテーブル構造は図 4 のとおりである。主キーには主索引が定義されている。

勤怠 ( <u>従業員番号</u> , <u>年月日</u> , <u>年休フラグ</u> , <u>出勤時刻</u> , <u>退勤時刻</u> , <u>残業時間</u> , <u>エリアコード</u> , <u>上長承認</u> , <u>監査結果</u> )
再開位置 ( <u>ジョブ ID</u> , <u>再開従業員番号</u> )

注記 エリアコードには、就業した建屋のエリアコードが設定される。自社建屋に立ち寄らない直行直帰の出張をした場合は、NULL が設定される。

図 4 監査機能に関するテーブルのテーブル構造

“勤怠” テーブルには、従業員ごと年月日ごとの勤怠情報を記録する。上長承認には、承認済み（‘Y’）又は未承認（‘N’）を設定する。監査結果には、監査未チェックの行は NULL を設定し、チェックしたら適正（‘Y’）又は不適正（‘N’）を設定する。

監査機能のジョブが途中で失敗した場合に、ジョブを再実行するために“再開位置” テーブルを使用する。再開従業員番号には処理中の従業員番号を記録し、再実行時には記録していた従業員番号から処理を再開する。

### 3. 監査機能の処理設計

処理の流れと SQL 文を図 5 に示す。トランザクションの ISOLATION レベルは READ COMMITTED とする。

- ① 当該ジョブの処理対象の従業員番号の範囲を、HBEGINID と HENDID に設定する。  
当該ジョブの処理対象の年月日の範囲を、HBEGINDAY と HENDDAY に設定する。
  - ② 当該ジョブのジョブ ID を、HJOBID に設定し、かつ、“再開位置” テーブルからジョブ ID が HJOBID の行を検索する。
    - ②-1 結果行がない場合、①のHBEGINIDをHTARGETIDに設定し、  
“再開位置” テーブルに、HJOBID と HTARGETID の行を挿入する。
    - ②-2 結果行がある場合、その再開従業員番号を、HTARGETID に設定する。
  - ③ “勤怠” テーブルから行を読むために次のカーソルを開く。
- ```

DECLARE CSR CURSOR WITH HOLD FOR
  SELECT * FROM 勤怠
    WHERE 従業員番号 BETWEEN [e] AND [f]
      AND 年月日 BETWEEN :HBEGINDAY AND :HENDDAY
      AND 上長承認 = 'Y' AND 監査結果 [g]
    ORDER BY [h], 年月日
  
```
- ④ カーソルで読む行がある限り⑤～⑩の処理を繰り返し実行し、行がなくなれば⑪に進む。  
なお、処理行数をカウントし、⑩の処理は N 行おきにだけ実行する。
  - ⑤ カーソルから 1 行読む。
  - ⑥ 出勤時刻と退勤時刻が、入退室ログの入退室時刻と矛盾しないかチェックする。
  - ⑦ 年休を取得している場合、不適正な出勤でないか入退室ログをチェックする。
  - ⑧ エリアコードが NULL でない場合、入退室ログのエリアと矛盾がないかチェックする。
  - ⑨ ⑤で読んだ行の監査結果を、⑥～⑧のチェック結果で更新する。
  - ⑩ “再開位置” テーブルの当該ジョブの再開従業員番号を、⑤で読んだ行の従業員番号で  
更新し、コミットする。
  - ⑪ カーソルを閉じる。
  - ⑫ [ア]
  - ⑬ コミットし、ジョブを終了する。

注記 HBEGINID, HENDID, HBEGINDAY, HENDDAY, HJOBID 及び HTARGETID は、それぞれ開始従業員番号、終了従業員番号、開始年月日、終了年月日、処理中ジョブ ID、処理中従業員番号を示すホスト変数である。

図 5 監査機能の処理の流れと SQL 文（未完成）

監査機能の実行時には他トランザクションも実行されており、排他ロック解放待ちタイムアウトの考慮も必要である。排他ロック解放待ちタイムアウト時間を T 秒とすると、他トランザクションで排他ロック解放待ちタイムアウトさせないための図 5 中の N 行を見積もる計算式は、次の式となる。

$$N < (T - [A] \text{ に必要な時間}) \div [B] \text{ に必要な時間}$$

設問1 〔所在情報管理機能の追加〕について答えよ。

- (1) 表1中の [a] ~ [d] に入る適切な字句を答えよ。
- (2) 所在情報更新処理を実行中に、入退室ログ登録処理が発生するとデッドロックとなる可能性がある。デッドロックを引き起こすロックの状況について、時系列に、対象となるテーブルのテーブル名、ロックを掛けるトランザクション、ロック種別、ロック状態を答えて、次に示す表2を完成させよ。なお、ロックを掛けるトランザクション、ロック種別、ロック状態については、表中の該当する方を○で囲んで示せ。

表2 ロック状況（未完成）

| 時系列 | テーブル名 | ロックを掛けるトランザクション   | ロック種別          | ロック状態            |
|-----|-------|-------------------|----------------|------------------|
| 1   | エリア状況 | 所在情報更新<br>入退室ログ登録 | 共有ロック<br>専有ロック | ロック済み<br>ロック解放待ち |
| 2   |       | 所在情報更新<br>入退室ログ登録 | 共有ロック<br>専有ロック | ロック済み<br>ロック解放待ち |
| 3   |       | 所在情報更新<br>入退室ログ登録 | 共有ロック<br>専有ロック | ロック済み<br>ロック解放待ち |
| 4   |       | 所在情報更新<br>入退室ログ登録 | 共有ロック<br>専有ロック | ロック済み<br>ロック解放待ち |

- (3) (2)のデッドロックを回避するために、図2の入退室ログ登録処理の処理順序を変更する。変更内容を、図2中の①～③を用いて20字以内で答えよ。

設問2 〔監査機能の追加〕について答えよ。

- (1) 図5中の [e] ~ [h] に入る適切な字句を答えよ。
- (2) 図5中の [ア] で行うべき処理を40字以内で答えよ。
- (3) [A] と [B] の対象となる処理に該当するものを、図5中の④～⑯から選べ。なお、該当する処理が複数ある場合は全て選ぶこと。

問3 情報システム会社のプロジェクト稼働管理システムのデータベース物理設計・SQL設計・性能に関する次の記述を読んで、設問に答えよ。

情報システム会社のE社は、自社のプロジェクト稼働管理システム（以下、PJシステムという）を、RDBMSを用いて更改することになり、Fさんが実装を任せられた。

[RDBMSの主な仕様]

1. DMLのアクセス経路は、RDBMSによって索引探索又は表探索が選択される。
2. 索引は、クラスタ性という性質によって、高クラスタな索引と低クラスタな索引に分けられる。
  - ・高クラスタな索引は、キー値の順番と、キーが指す行の物理的な並び順が一致しているか、完全に一致していなくても、隣接するキーが指す行が同じページに格納されている割合が高い。
  - ・低クラスタな索引は、キー値の順番と、キーが指す行の物理的な並び順が一致している割合が低く、行へのアクセスがランダムになる。

[業務の概要]

1. 組織、従業員、役職、ランク、時間単価
  - (1) E社には、複数の組織がある。組織は階層構造であり、最上位の組織以外はいずれか一つの上位組織に属する。
  - (2) 従業員は、従業員コードで識別し、いずれか一つの組織に属する。
  - (3) 役職には、SE、シニアSE、マネージャなどがある。役職は役職コードで識別する。従業員はいずれか一つの役職をもつ。
  - (4) ランクは、労務費の時間単価を区別するもので、ランクコードで識別する。役職はいずれか一つのランクに対応する。
  - (5) 時間単価は、ランク別組織別年月日別に決めている。組織の変更、従業員の異動などによって、月初に時間単価を見直すことがある。
2. プロジェクト、稼働計画、稼働実績
  - (1) プロジェクト（以下、PJという）は、従業員の稼働状況を管理する単位であ

る。従業員は複数の PJ に参加することがあり、PJ に参加していない従業員も一部いる。

- (2) PJ に必要な人員を要員という。
- (3) PJ 開始前に稼働計画を立案するとき、要員ごと参加年月ごとに計画時間を見積もる。参加する従業員が確定したとき、稼働計画の要員に対して従業員を割り当てる。PJ 開始後、必要に応じて計画を修正する。
- (4) PJ に参加している各従業員は、稼働実績として月内の日別 PJ 別の稼働時間を入力する。従業員は同じ日に複数 PJ の稼働時間を入力できる。

#### [PJ システムのテーブル]

##### 1. テーブル構造、列の意味・制約、統計情報・索引定義

主なテーブルのテーブル構造を図 1 に、主な列の意味・制約を表 1 に示す。また、“従業員” テーブルの主な統計情報・索引定義を表 2 に、“稼働実績” テーブルの主な統計情報・索引定義を表 3 に示す。

|                                              |
|----------------------------------------------|
| 組織（組織コード、組織名、発足年月日、廃止年月日、組織長従業員コード、上位組織コード）  |
| 従業員（従業員コード、従業員氏名、組織コード、役職コード、退職年月日、更新年月日）    |
| 従業員履歴（従業員コード、更新年月日、従業員氏名、組織コード、役職コード）        |
| 役職（役職コード、役職名、ランクコード）                         |
| 時間単価（ランクコード、組織コード、適用開始年月日、時間単価）              |
| ランク（ランクコード、ランク名）                             |
| PJ（PJ コード、PJ 名、開始年月日、終了年月日、PM 従業員コード）        |
| 稼働計画（PJ コード、要員番号、計画年、計画月、ランクコード、従業員コード、計画時間） |
| 稼働実績（PJ コード、稼働年月日、従業員コード、稼働時間）               |

図 1 主なテーブルのテーブル構造

表 1 主な列の意味・制約

| 列名        | 意味・制約                                                                      |
|-----------|----------------------------------------------------------------------------|
| 上位組織コード   | 組織が属する上位組織の組織コード。最上位組織の上位組織コードには NULL を設定する。                               |
| 更新年月日     | “従業員” テーブルの列値を更新した日付                                                       |
| PM 従業員コード | PJ を管理する従業員の従業員コード                                                         |
| 要員番号      | 稼働計画において PJ 内で要員を一意に識別する通し番号。要員に従業員が割り当てられていない場合、当該要員の従業員コードには NULL を設定する。 |
| 計画時間      | PJ ごと要員ごと計画年ごと計画月ごとに見積もった作業計画。単位は時間。                                       |
| 稼働時間      | PJ ごと稼働年月日ごと従業員ごとの作業実績。単位は時間。                                              |

表2 “従業員” テーブルの主な統計情報・索引定義（一部省略）

| 上段：行数、下段：ページ当たり行数   | 列名     | 列値個数  | 主索引<br>(列の定義順) | 副次索引1<br>(列の定義順) |
|---------------------|--------|-------|----------------|------------------|
| 8,800 行<br>20 行／ページ | 従業員コード | 8,800 | 1              |                  |
|                     | 従業員氏名  | 8,750 |                |                  |
|                     | 組織コード  | 400   |                | 1                |
|                     | :      | :     |                |                  |

表3 “稼働実績” テーブルの主な統計情報・索引定義

| 上段：行数、下段：ページ当たり行数        | 列名     | 列値個数  | 主索引<br>(列の定義順) | 副次索引1<br>(列の定義順) | 副次索引2<br>(列の定義順) |
|--------------------------|--------|-------|----------------|------------------|------------------|
| 9,600,000 行<br>100 行／ページ | PJ コード | 5,000 | 1              |                  |                  |
|                          | 稼働年月日  | 1,000 | 2              |                  | 1                |
|                          | 従業員コード | 8,000 | 3              | 1                |                  |
|                          | 稼働時間   |       |                |                  |                  |

注記 網掛け部分は表示していない。

## 2. “従業員” テーブルの行更新における更新履歴処理

“従業員” テーブルの組織コード、役職コードを更新するとき、当該従業員の更新前の行を更新の履歴として“従業員履歴” テーブルに挿入する。

## 3. “組織” テーブルの行削除処理

E 社では、組織の改廃がある。PJ 管理に不要になった組織コードを削除する場合、次のような手順で行う。

- ① 廃止済みの組織であり、かつ、PJ が終了済みなど PJ 管理に不要と判断できる組織コードを、SELECT 文を用いて調べる。
- ② ①で調べた組織コードの行を、“組織” テーブルから DELETE 文を用いて削除する。

## [テーブルの定義と実装]

### 1. テーブルの定義

- (1) Fさんは、図1中の各テーブルを定義するCREATE TABLE文を設計した。ここで、各CREATE TABLE文には外部キー制約を実装することとした。そのうち、“組織” テーブルを定義するCREATE TABLE文を、図2に示す。

```

CREATE TABLE 組織 (
    組織コード      CHAR(8)      NOT NULL PRIMARY KEY,
    組織名          VARCHAR(50)   NOT NULL,
    発足年月日      DATE        NOT NULL,
    廃止年月日      DATE,
    組織長従業員コード CHAR(8),
    上位組織コード  CHAR(8),
    FOREIGN KEY (組織長従業員コード) REFERENCES 従業員 (従業員コード) ON DELETE RESTRICT,
    FOREIGN KEY (上位組織コード)      REFERENCES 組織 (組織コード)      ON DELETE RESTRICT)

```

図 2 “組織” テーブルを定義する CREATE TABLE 文

(2) Fさんは、他の表が未定義の状態で“組織” テーブルを定義する図 2 の CREATE TABLE 文を実行したところ失敗した。そこで、Fさんは、図 2 の CREATE TABLE 文を見直し、次の①～③の順番で定義を実行したところ、全ての実行が成功した。

- ① “組織” テーブルを定義する図 2 中から、[a] を外部キーとする指定を削除した CREATE TABLE 文を実行する。
- ② “従業員”, “役職”, “時間単価”, “ランク” の各テーブルを定義する CREATE TABLE 文を “[b]”, “[c]”, “役職”, “[d]” の順番で実行する。
- ③ “組織” テーブルに対する [e] 文を用いて、[a] を外部キーとする指定の定義を追加する。

## 2. テーブルへの行登録

次に、Fさんは、“組織” テーブルに INSERT 文を用いて行を挿入した。次いで“従業員”, “役職”, “時間単価”, “ランク” の各テーブルに対しても INSERT 文を用いて行を挿入した。その後、UPDATE 文で適宜列値を更新した。

### [稼働計画の立案・稼働実績の確認]

Fさんは、稼働計画の立案及び稼働実績の確認を支援するための SQL 文を設計した。設計した SQL 文の例を、表 4 に示す。

表4 稼働計画の立案及び稼働実績の確認を支援するSQL文の例

| SQL  | SQL文の構文（上段：目的、下段：構文）                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SQL1 | <p>要員が埋まらないPJがあるので、2024年11月の稼働計画において役職コードがSEの従業員を対象に、稼働計画に登録されていない従業員も含めて要員を探す。従業員に対して、組織ごとに計画時間の少ない順に順序付けし、順序に沿って従業員数が均等となるように1~3の番号を付与した“時間階級”，及び従業員全体で計画時間の少ない順に順位付けした“時間ランク”を算出する。結果行は組織コード順、時間ランクが低い順、従業員コード順に並べる。ここで、稼働計画にない従業員の計画時間はゼロで表示する。</p> <pre> SELECT 組織コード, A.従業員コード, COALESCE(SUM(計画時間), 0) AS 計画時間合計,        NTILE(3) OVER (PARTITION BY [ア] ORDER BY [イ]) AS 時間階級,        RANK() OVER (ORDER BY [イ]) AS 時間ランク   FROM [ウ] A LEFT OUTER JOIN [エ] B     ON A.従業員コード = B.従業員コード AND 計画年 = '2024' AND 計画月 = '11'    WHERE 役職コード = 'SE'   GROUP BY 組織コード, A.従業員コード ORDER BY 組織コード, 時間ランク, A.従業員コード </pre> |
| SQL2 | <p>指定した組織に所属する従業員の指定した月の1か月分の稼働時間の合計を調べる。</p> <pre> SELECT SUM(稼働時間) AS 稼働時間合計   FROM 稼働実績 K JOIN 従業員 S ON K.従業員コード = S.従業員コード   WHERE 組織コード = :hv1     AND 稼働年月日 BETWEEN :hv2 AND :hv3 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                         |

注記 ホスト変数のhv1には組織コードを、hv2には開始年月日（例：2024-06-01）を、hv3には終了年月日（例：2024-06-30）を設定する。

#### 〔問合せの性能改善〕

Fさんは、“稼働実績”テーブルへの問合せに利用される表4中のSQL2について、性能の改善を依頼された。Fさんが調べたところ、稼働実績を一括入力する従業員が多く、1か月単位で見たとき、行の登録順が従業員、稼働年月日、PJコード順であり、従業員当たり1か月分の行が高々2ページに格納されることが分かった。そこで、索引のクラスタ性と次の三つの前提を踏まえて、(1)～(5)の手順で性能改善を試みた。

- ・それぞれの列値は均等に分布していると仮定する。
- ・PJに参加しない従業員だけで構成される組織はないと仮定する。
- ・全従業員が同じ曜日で働いていると仮定し、1か月は20日として計算する。

- (1) SQL2のアクセス経路として、“従業員”テーブルを外表、“稼働実績”テーブルを内表とする入れ子ループ結合を想定する。
- (2) このアクセス経路では、まず外表から指定した組織コードに対して、外表の副次索引1を用いて平均22行を読み込む。外表の副次索引1は低クラスタな索引なので、最大で [f] ページを読み込む。
- (3) 外表から読み込んだ従業員コード1件ごとに、内表の副次索引1を用いて①従

業員 1 人当たりの稼働実績である 1,200 行を読み込み、行データの稼働年月日に  
対して BETWEEN 述語を評価する。表 3 の稼働年月日列の列値個数は 1,000 (50 か  
月分) なので、内表の集計対象の行は、1 人当たり [g] 行である。稼働実  
績を計上している従業員は組織当たり [h] 人なので、集計対象の行は組  
織当たり [i] 行となる。内表の副次索引 1 は高クラスタなので、読み込みペ  
ージ数は組織当たり最大 [j] ページである。

- (4) 次に、読み込みページ数を削減するために、{従業員コード、稼働年月日}をキーとする副次索引 3 を追加した場合の性能を検討した。この索引を使用した場合、副次索引 1 と比較すると、1 か月分を索引で絞り込めるので、表からの読み込み行数及び読み込みページ数は [k] 分の 1 に削減される。
- (5) 副次索引 3 の利用によって、表からの読み込み行数及び読み込みページ数を削減できるので、副次索引 3 を実装することにした。

設問 1 〔テーブルの定義と実装〕について答えよ。

- (1) “1. テーブルの定義”について、本文中の [a] ~ [e] に  
入れる適切な字句を答えよ。
- (2) “2. テーブルへの行登録”において、“組織” テーブルへ行を挿入する場合、外部キーである組織長従業員コード及び上位組織コードのそれについて、考慮すべき事項を一つずつ、25 字以内で答えよ。
- (3) F さんは、図 1 中のテーブルのうち、“時間単価” テーブルの定義では、外部キーである“組織コード” の DELETE オプションを CASCADE に指定した。SET NULL 又は RESTRICT を指定した場合、“時間単価” テーブルの定義時又は“組織” テーブルの行削除時に制約違反で失敗するおそれがあると考えたからである。なぜ制約に違反するのか、理由をそれぞれ 45 字以内で具体的に答えよ。

設問 2 〔稼働計画の立案・稼働実績の確認〕について、表 4 中の SQL1 の [ア] ~ [エ] に入れる適切な字句を答えよ。

設問 3 〔問合せの性能改善〕について答えよ。

- (1) 統計情報について、表 3 の“稼働実績” テーブルの従業員コードの列値個数が表 2 の“従業員” テーブルの従業員コードの列値個数より少ないのでな

ぜか。本文中の用語を用いて、30字以内で答えよ。

- (2) 本文中の  に入れる適切な数値を答えよ。
- (3) 下線①について、稼働年月日列の列値個数が 1,000 であるにもかかわらず、従業員 1 人当たりの稼働実績の行数が 1,000 よりも多いのはなぜか。本文中の用語を用いて、30字以内で答えよ。
- (4) 本文中の  ~  に入れる適切な数値を答えよ。

[ メモ用紙 ]

6. 退室可能時間中に退室する場合は、手を挙げて監督員に合図し、答案用紙が回収されてから静かに退室してください。

|        |               |
|--------|---------------|
| 退室可能時間 | 13:10 ~ 13:50 |
|--------|---------------|

7. 問題に関する質問にはお答えできません。文意どおり解釈してください。
8. 問題冊子の余白などは、適宜利用して構いません。ただし、問題冊子を切り離して利用することはできません。
9. 試験時間中、机上に置けるものは、次のものに限ります。  
なお、会場での貸出しありません。
- 受験票、黒鉛筆及びシャープペンシル（B 又は HB）、鉛筆削り、消しゴム、定規、時計（時計型ウェアラブル端末は除く。アラームなど時計以外の機能は使用不可）、ハンカチ、ポケットティッシュ、目薬  
これら以外は机上に置けません。使用もできません。
10. 試験終了後、この問題冊子は持ち帰ることができます。
11. 答案用紙は、いかなる場合でも提出してください。回収時に提出しない場合は、採点されません。
12. 試験時間中にトイレへ行きたくなったり、気分が悪くなったりした場合は、手を挙げて監督員に合図してください。
13. 午後Ⅱの試験開始は 14:30 ですので、14:10 までに着席してください。

試験問題に記載されている会社名又は製品名は、それぞれ各社又は各組織の商標又は登録商標です。

なお、試験問題では、<sup>TM</sup> 及び <sup>®</sup> を明記していません。