

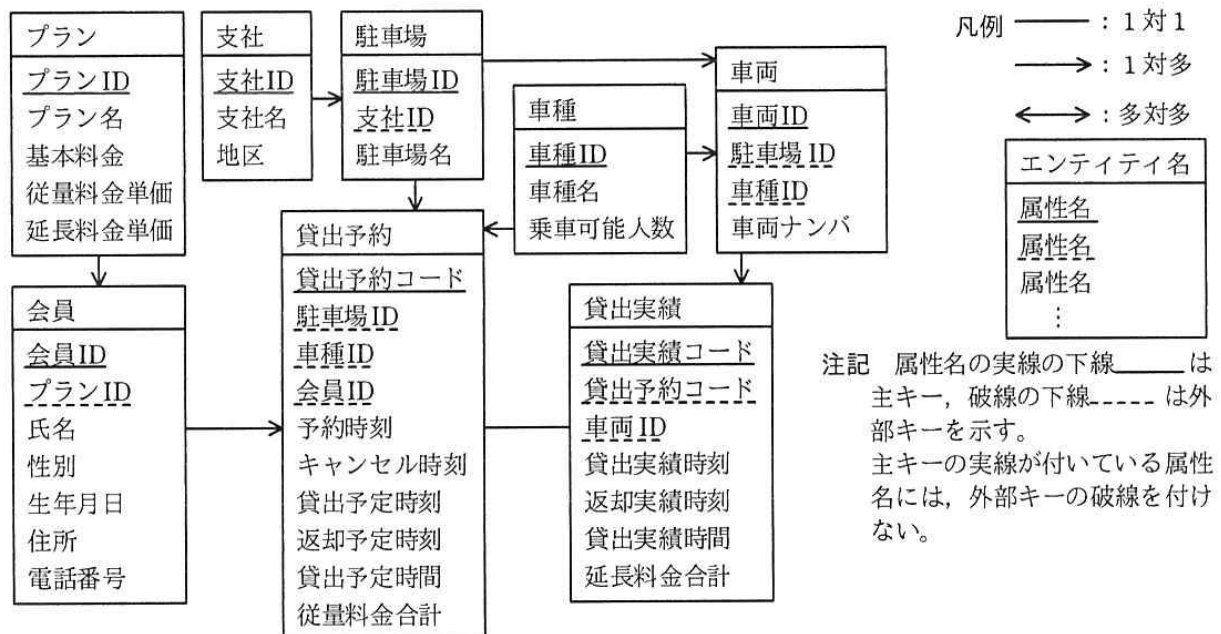
問6 経営分析システムのためのデータベース設計に関する次の記述を読んで、設問 1～4 に答えよ。

P 社は、個人向けのカーシェアリングサービスを運営する MaaS（Mobility as a Service）事業者である。シェアリングのニーズが高い大都市の地区を中心に、500 駐車場で約 2,000 台の自動車（以下、車両という）を貸し出している。P 社には本社のほかに、各地区でのサービス運営を担当する支社が 10 社ある。本社はサービス全体を統括しており、新サービスの企画やマーケティングなどを行っている。支社は貸出管理システムを用いて現場で車両の貸出管理業務を行っている。

本社では、サービス運営状況を多角的な観点でタイムリーに把握して、適切な意思決定を行うために、貸出管理システムのデータをソースとする経営分析システムを構築することになった。本社の情報システム部の Q さんはデータエンジニアに任命され、データサイエンティストである R さんとプロジェクトを推進することになった。

#### 〔データソースの調査〕

貸出管理システムには、貸出予約及び貸出実績のデータが過去 5 年間分蓄積されている。貸出管理システムのデータモデルの抜粋を図 1 に示す。



利用希望者はあらかじめ P 社の会員になり、いずれかのプランに加入しておく必要がある。プランごとに基本料金（月額）、従量料金及び延長料金（いずれも 10 分単位）の単価が決まっている。会員が車両を借りたいときは、P 社のホームページで借りたい日時や駐車場、車種などを選択し、貸出を予約する。貸出や返却の実績時刻が予約時の内容と異なる場合であっても、貸出予約の情報は修正しない。従量料金合計は予約時に指定された貸出予定時間を基に算出する。予約時に指定した返却予定時刻より早い時刻に返却しても、従量料金合計は減算しない。予約時に指定した返却予定時刻より遅い時刻に返却した場合は遅延返却として扱う。遅延返却は後の時間帯に予約している別の会員の迷惑となるので、超過した時間については従量料金よりも高い延長料金によって延長料金合計を算出する。これによって、遅延返却の発生件数（以下、遅延返却発生件数という）の低減を図っている。毎月末に当月の基本料金、従量料金合計及び延長料金合計を合算して、翌月に会員に請求する。

貸出管理システムのデータベースでは、データモデルのエンティティ名を表名にし、属性名を列名にして、適切なデータ型で表定義した関係データベースによって、データを管理している。時刻は TIMESTAMP 型、年月日は DATE 型で定義されている。

また、P 社では KPI の一つとして車両稼働率を重視している。車両稼働率とは、各車両における 1 日当たりの貸出実績時間の割合である。平均車両稼働率の目標データは、表計算ソフトのデータとして、年月日別・駐車場別・車種別に過去 3 年間分が蓄積されており、それ以前のデータは破棄されている。

#### 〔業務要件の把握〕

P 社の経営企画部では、車両の追加整備計画の立案を検討している。Rさんは経営企画部にヒアリングを行い、経営分析システムの業務要件を把握した。業務要件の抜粋を図 2 に示す。

Qさんは、データソースの調査結果を踏まえて、図 2 の業務要件の実現可能性を評価した。その結果、①業務要件の一部は経営分析システムの運用開始直後には実現できないことが判明した。対応方針を経営企画部と協議した結果、業務要件は変更せず、運用開始直後の分析は、実現可能な範囲で行うことで合意した。



〔データ加工処理の開発〕

貸出管理システムのデータベースから経営分析システムのデータベースへのデータ連携時に、一部のデータを加工する必要がある。Qさんは、データ加工処理用のデータベースを用意し、データ加工を行うバッチ処理プログラムを開発した。図4のSQL文は、そこで用いられている図3の貸出表の遅延返却発生件数データを作成するためのものである。ここで、TIMESTAMP\_TO\_DATE関数は、指定されたTIMESTAMP型の時刻をDATE型の年月日に変換するユーザ定義関数である。

バッチ処理プログラムでは、図4のSQL文で作成したデータを貸出表に挿入する際、遅延返却発生件数が0件のレコードに対する処理も別途行うようになっている。

```
SELECT R.貸出予定年月日, R.駐車場ID, R.車種ID, R.会員ID, COUNT(*) AS 遅延返却発生件数
FROM (SELECT Y.貸出予約コード, Y.駐車場ID, Y.車種ID, Y.会員ID,
TIMESTAMP_TO_DATE(Y.貸出予定時刻) AS 貸出予定年月日, Y.返却予定時刻 FROM 貸出予約 Y) R
WHERE R.返却予定時刻 < J.返却実績時刻
```

図4 遅延返却発生件数データを作成するSQL文

〔分析のレスポンス性能の改善〕

性能検証を実施したところ、分析対象期間を過去複数年間、時間軸を月別として人気車種及び遅延返却発生件数を分析する場合、種々の分析に時間が掛かり過ぎるので改善してほしいという要望が経営企画部から挙がった。経営分析システムのデータベースのインデックスは既に適切に作成している。分析のレスポンス性能を改善するために、Qさんは②データマートとして集計表を追加した。

設問1 本文中の下線①について、実現できない業務要件を40字以内で具体的に答えよ。

設問2 〔経営分析システムのデータモデル設計〕について、(1), (2)に答えよ。

(1) 本文中の a に入れる適切な字句を解答群の中から選び、記号で答えよ。

解答群

ア 3層スキーマ

イ オブジェクト指向

ウ スタースキーマ

エ スノーフレイクスキーマ

オ 第3正規形

カ 非正規形

- (2) 図3中の  ～  に入れる適切なエンティティ間の関連及び属性名を答えよ。なお、エンティティ間の関連及び属性名の表記は図1の凡例及び注記に倣うこと。

設問3 「データ加工処理の開発」について、(1)，(2)に答えよ。

- (1) 図4中の ,  に入れる適切な字句を答えよ。なお、表の列名には必ずその表の相関名を付けて答えよ。

- (2) 図4のSQL文を実行すべき頻度を2字以内で答えよ。

設問4 本文中の下線②について、追加した集計表の主キーを答えよ。

問6 企業向け電子書籍サービスの追加設計と実装に関する次の記述を読んで、設問 1～4 に答えよ。

H 社は、個人会員向けに電子書籍の販売及び閲覧サービス（以下、既存サービスという）を提供する中堅企業である。近年、テレワークの普及に伴い、企業での電子書籍の需要が高まってきた。そこで、既存サービスに加え、企業向け電子書籍サービス（以下、新サービスという）を開発することになった。

新サービスの開始に向けて、企業向け書籍購入サイトを新たに作成し、既存サービスで提供している電子書籍リーダを改修する。新サービスの機能概要を表 1 に、検討したデータベースの E-R 図の抜粋を図 1 に示す。

このデータベースでは、E-R 図のエンティティ名を表名にし、属性名を列名にして、適切なデータ型で表定義した関係データベースによって、データを管理する。

表 1 新サービスの機能概要

No.	機能名	概要
1	一括購入	企業の一括購入担当者が、電子書籍を一括購入する。購入した電子書籍を企業の社員に割り当てる方法には、次の二つがある。 (1) 一括購入担当者が、配布対象の社員にあらかじめ割り当てておく方法 (2) 社員が、未割当の一括購入された電子書籍を割当依頼する方法
2	企業補助	社員が、自己啓発に役立つビジネスや技術など特定の分類の電子書籍を購入する。その際、企業が購入額の一部を負担する。ただし、企業は負担する上限金額を書籍分類ごとに設定する。
3	割引購入	社員が、個人として読みたい本や雑誌などの電子書籍を購入する。その際、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。
4	書籍閲覧	社員が、電子書籍リーダに、H 社が付与した企業 ID、社員 ID 及び社員パスワードを用いてログインし、No. 1～3 で購入した電子書籍を閲覧する。電子書籍リーダにログインすると、一括購入で割り当てられた電子書籍や、社員が購入した電子書籍が一覧表示され、各電子書籍を選択して閲覧できる。

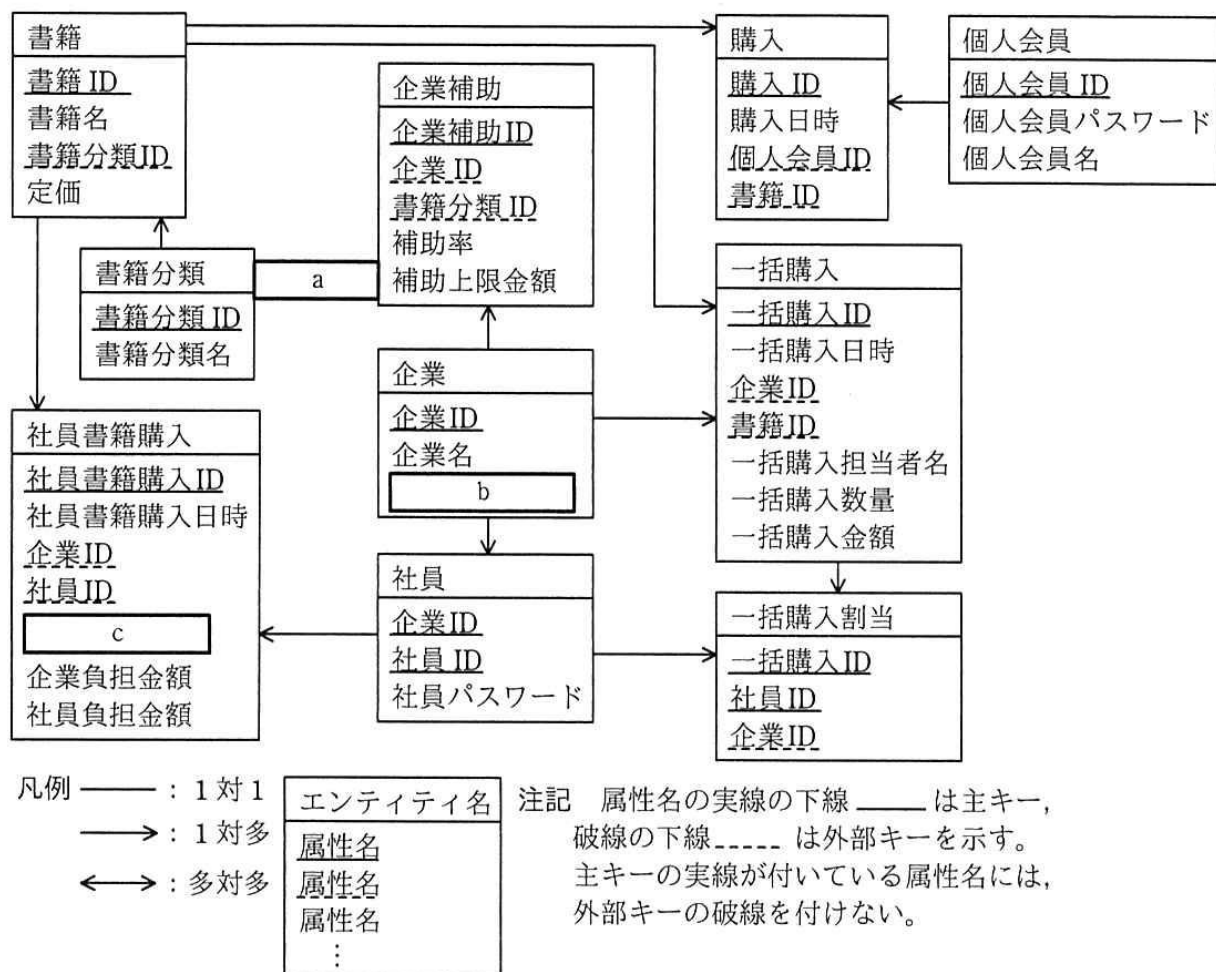


図1 検討したデータベースのE-R図(抜粋)

#### [一括購入機能の社員割当処理の作成]

表1中の一括購入機能の概要(2)にある, 社員が割当依頼した電子書籍を割り当てる処理を考える。検討した処理の流れを表2に示す。ここで, “:一括購入ID”は割当依頼された一括購入IDを, “:企業ID”及び“:社員ID”は割当依頼した社員の企業IDと社員IDを格納する埋込み変数である。



表 2 検討した処理の流れ

手順	処理概要	使用するSQL文
1	社員が割当依頼した一括購入 ID から、一括購入数量を取得する。	SELECT 一括購入数量 FROM 一括購入 WHERE 一括購入ID = :一括購入ID
2	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。	SELECT <span style="border: 1px solid black; padding: 2px;">d</span> FROM 一括購入割当 WHERE 一括購入ID = :一括購入ID
3	手順 1 で取得した数量が、手順 2 で取得した数量より <span style="border: 1px solid black; padding: 2px;">e</span> 場合、手順 4 に進む。そうでない場合、処理を終了する。	なし
4	割当依頼した社員に一括購入 ID を割り当てる。	INSERT INTO 一括購入割当 (一括購入ID, 社員ID, 企業ID) <span style="border: 1px solid black; padding: 2px;">f</span>

表 2 のレビューを実施したところ、処理の流れや SQL 文に問題はないが、①トランザクションの同時実行制御には専有ロックを用いるように、とのアドバイスを受けた。

#### 〔書籍閲覧機能の作成〕

電子書籍リーダに、社員がログインした際、閲覧可能な重複を含まない書籍の一覧を取得する SQL 文を図 2 に示す。ここで、“:企業 ID” 及び “:社員 ID” は、ログインした社員の企業 ID と社員 ID を格納する埋込み変数である。また、図 2 の c には、図 1 の c と同じ字句が入る。

SELECT sk. <span style="border: 1px solid black; padding: 2px;">c</span> FROM 社員書籍購入 sk WHERE sk.企業ID = :企業ID AND sk.社員ID = :社員ID <span style="border: 1px solid black; padding: 2px;">g</span> SELECT ik. <span style="border: 1px solid black; padding: 2px;">c</span> FROM 一括購入 ik INNER JOIN 一括購入割当 iw <span style="border: 1px solid black; padding: 2px;">h</span> WHERE ik.企業ID = :企業ID AND iw.社員ID = :社員ID
--

図 2 閲覧可能な重複を含まない書籍の一覧を取得する SQL 文



〔書籍閲覧機能の改善〕

書籍閲覧機能のレビューを実施したところ、既存サービスを個人で利用している社員は、電子書籍リーダーのログイン ID を個人会員 ID から企業 ID と社員 ID に切り替えて利用しなければならず煩雑である、との指摘を受けた。

そこで、電子書籍リーダーに個人会員 ID を用いてログインした際、社員として閲覧できる書籍も一覧に追加して閲覧できるように、E-R 図に新たに②一つエンティティを追加し、電子書籍リーダーに③一つ画面を追加した上で書籍閲覧機能に改修を施した。

設問 1 図 1 中の a ～ c に入れる適切なエンティティ間の関連及び属性名を答え、E-R 図を完成させよ。

なお、エンティティ間の関連及び属性名の表記は、図 1 の凡例に倣うこと。

設問 2 〔一括購入機能の社員割当処理の作成〕について、(1)、(2)に答えよ。

(1) 表 2 中の d ～ f に入れる適切な字句を答えよ。

(2) 本文中の下線①の専有ロックを用いなかった場合、どのような問題が発生するか。30 字以内で述べよ。

設問 3 図 2 中の g , h に入れる適切な字句又は式を答えよ。

なお、表の列名には必ずその表の相関名を付けて答えよ。

設問 4 〔書籍閲覧機能の改善〕について、(1)、(2)に答えよ。

(1) 本文中の下線②で追加したエンティティの属性名を全て列挙せよ。

なお、エンティティの属性名に主キーや外部キーを示す下線は付けなくてよい。

(2) 本文中の下線③とは、どのような画面か。25 字以内で述べよ。

問6 クーポン発行サービスに関する次の記述を読んで、設問1～4に答えよ。

K社は、インターネットでホテル、旅館及びレストラン（以下、施設という）の予約を取り扱う施設予約サービスを運営している。各施設は幾つかの利用プランを提供していて、利用者はその中から好みのプランを選んで予約する。会員向けサービスの拡充施策として、現在稼働している施設予約サービスに加え、クーポン発行サービスを開始することにした。

発行するクーポンには割引金額が設定されていて、施設予約の際に料金の割引に利用することができる。K社は、施設、又は都道府県、若しくは市区町村を提携スポンサとして、提携スポンサと合意した割引金額、枚数のクーポンを発行する。

クーポン発行に関しては、提携スポンサによって各種制限が設けられているので、クーポンの獲得、及びクーポンを利用した予約の際に、制限が満たされていることをチェックする仕組みを用意する。

提携スポンサによって任意に設定可能なチェック仕様の一部を表1に、クーポン発行サービスの概要を表2に示す。

表1 提携スポンサによって任意に設定可能なチェック仕様（一部）

提携スポンサ	クーポンの獲得制限	クーポンを利用した予約制限
施設	・ 同一会員による同一クーポンの獲得可能枚数を、1枚に制限する（以下、“同一会員1枚限りの獲得制限”という）。	・ 設定した施設だけを予約可能にする。 ・ 利用金額が設定金額以上の予約だけを可能にする。
都道府県、市区町村	・ 設定地区に居住する会員だけが獲得可能にする。	・ 設定地区にある施設だけを予約可能にする。

表2 クーポン発行サービスの概要

利用局面	概要
クーポンの照会	・ 発行予定及び発行中クーポンの情報は、会員向けのメール配信によって会員に周知され、施設予約サービスにおいて検索、照会ができる。
クーポンの獲得	・ 発行中のクーポンを利用するためには、会員がクーポン獲得を行う必要がある。 ・ クーポン獲得を行える期間は定められている。 ・ クーポンの発行枚数が上限に達すると、以降の獲得はできない。
クーポンの利用	・ 獲得したクーポンは、施設予約サービスにおいて料金の割引に利用できる。 ・ 1枚のクーポンは一つの予約だけに利用できる。 ・ クーポンを利用した予約をキャンセルすると、そのクーポンを別の予約に利用できる。 ・ クーポンの利用期間は定められていて、期限を過ぎたクーポンは無効となる。

クーポン発行サービスと施設予約サービスで使用するデータベース（以下、予約サイトデータベースという）の E-R 図（抜粋）を図 1 に示す。予約サイトデータベースでは、E-R 図のエンティティ名をテーブル名に、属性名を列名にして、適切なデータ型で表定義した関係データベースによってデータを管理する。

```

    erDiagram
        会員 ||--}| 予約 : "予約コードと会員コード"
        予約 ||--}| クーポン管理 : "クーポン発行連番"
        予約 ||--}| クーポン管理 : "獲得会員コード"
        施設 ||--}| クーポン管理 : "獲得会員コード"

        会員 {
            string 会員コード PK
            string 氏名
            string 会員住所都道府県コード
            string 会員住所市区町村コード
        }

        予約 {
            string 予約コード PK
            string 会員コード FK
            int a
            int b
            int 利用人数
            date 利用開始日時
            string クーポンコード FK
            string クーポン発行連番
        }

        クーポン管理 {
            string クーボンコード PK
            int 発行上限枚数
            float 割引金額
            date 獲得可能開始日時
            date 獲得可能終了日時
            date 利用期間開始日
            date 利用期間終了日
            int 獲得制限_1枚限り
            string 獲得制限_会員住所都道府県コード
            string 獲得制限_会員住所市区町村コード
            string 予約制限_施設コード
            float 予約制限_最低利用金額
            string 予約制限_施設住所都道府県コード
            string 予約制限_施設住所市区町村コード
        }

        施設 {
            string 施設コード PK
            string 施設名
            string 施設住所都道府県コード
            string 施設住所市区町村コード
        }
  
```

このER図は、予約管理システムのデータベース構造を示しています。

- 会員**: 会員コード (主キー)、氏名、会員住所都道府県コード、会員住所市区町村コード。
- 予約**: 予約コード (主キー)、会員コード (外キー)、属性a、属性b、利用人数、利用開始日時、クーポンコード (外キー)、クーポン発行連番。
- クーポン管理**: クーボンコード (主キー)、発行上限枚数、割引金額、獲得可能開始日時、獲得可能終了日時、利用期間開始日、利用期間終了日、獲得制限\_1枚限り、獲得制限\_会員住所都道府県コード、獲得制限\_会員住所市区町村コード、予約制限\_施設コード、予約制限\_最低利用金額、予約制限\_施設住所都道府県コード、予約制限\_施設住所市区町村コード。
- 施設**: 施設コード (主キー)、施設名、施設住所都道府県コード、施設住所市区町村コード。

**関係:**

- 予約** と **会員**: 予約コードと会員コードによる1対多関係 (属性a, b)。
- 予約** と **クーポン管理**: 予約コードとクーポン発行連番による1対多関係。
- 予約** と **クーポン管理**: 予約コードと獲得会員コードによる1対多関係。
- 施設** と **クーポン管理**: 施設コードと獲得会員コードによる1対多関係。

**凡例:**

- : 1対1
- > : 1対多
- <———— : 多対多

図1 予約サイトデータベースのE-R図(抜粋)

– 31 –

UNIQUE 制約を付けた。なお、予約サイトデータベースにおいては、UNIQUE 制約を構成する複数の列で一つの列でも NULL の場合は、UNIQUE 制約違反とならない。

d	クーポン明細 ADD CONSTRAINT クーポン明細_IX1 UNIQUE(クーポンコード, 獲得会員コード, 獲得制限_1枚限り)
---	---

図2 “同一会員1枚限りの獲得制限”を制約とするためのSQL文

L 主任は、①予約テーブルの“クーポンコード”，“クーポン発行連番”に対しても、UNIQUE 制約を付けた。

予約サイトデータベースでは、更新目的の参照処理と更新処理においてレコード単位にロックを掛け、多重処理を行う設定としている。ロックが掛かるとトランザクションが終了するまでの間、他のトランザクションによる同一レコードに対する処理はロック解放待ちとなる。

[クーポン獲得処理の連番管理方式]

クーポン発行サービスと施設予約サービスの CRUD 図（抜粋）を図3に示す。

クーポン新規登録処理では、1種類のクーポンにつき1レコードをクーポン管理テーブルに追加する。クーポン獲得チェック処理では、獲得可能期間、会員住所による獲得制限、発行上限枚数に関するチェックを行う。チェックの結果、エラーがない場合に表示される同意ボタンを押すことによって、クーポン獲得処理を行う。

処理名		テーブル名			
		会員	予約	クーポン管理	クーポン明細
クーポン発行サービス	クーポン新規登録	—	—	C	—
	クーポン獲得チェック	R	—	R	R
	クーポン獲得	R	—	②R	③CR
施設予約サービス	施設予約前チェック	R	R	R	R
	施設予約実行	R	C	—	R
	施設予約キャンセル	R	RD	—	—

注記 C：追加，R：参照，U：更新，D：削除

図3 クーポン発行サービスと施設予約サービスの CRUD 図（抜粋）

クーポン発行サービスでは、上限の定められた発行枚数分のクーポンを抜けや重複なく連番管理する方式が必要になる。特に、提携スポンサが都道府県，市区町村であ

るクーポンは割引金額が大きく、クーポンの発行直後にトラフィックが集中することが予想される。発行上限枚数到達後にクーポン獲得処理が動作する場合の考慮も必要である。L 主任は、トラフィック集中時のリソース競合によるレスポンス悪化を懸念して、ロック解放待ちを発生させない連番管理方式（以下、ロックなし方式という）の SQL 文（図 4）を考案した。この SQL 文では、ロックを掛けずに参照し、主キー制約によってクーポン発行連番の重複レコード作成を防止する。

ここで、関数 COALESCE(A, B)は、A が NULL でないときは A を、A が NULL のときは B を返す。また、“:クーポンコード”、“:会員コード” は、該当の値を格納する埋込み変数である。

```
INSERT INTO クーポン明細 (クーポンコード, クーポン発行連番, 獲得会員コード, 獲得制限_1 枚限り)
WITH 発行済枚数取得 AS (SELECT COALESCE(MAX( e ), 0) AS 発行済枚数
FROM クーポン明細 WHERE クーポンコード = :クーポンコード)
SELECT :クーポンコード,
       (SELECT 発行済枚数 + 1 FROM 発行済枚数取得 WHERE
        (SELECT 発行済枚数 FROM 発行済枚数取得) < 発行上限枚数),
       :会員コード, 獲得制限_1 枚限り
FROM クーポン管理 WHERE クーポンコード = :クーポンコード
```

図 4 ロックなし方式の SQL 文

〔クーポン獲得処理の連番管理方式の見直し〕

ロックなし方式をレビューした M 課長は、トラフィック集中時に主キー制約違反が発生することによって、会員による再オペレーションが頻発するデメリットを指摘し、ロック解放待ちを発生させることによって更新が順次行われる連番管理方式（以下、ロックあり方式という）の検討と方式の比較、高負荷試験の実施を指示した。

L 主任は、クーポン管理テーブルに対して初期値が 0 の“発行済枚数”という列を追加し、このデータ項目のカウントアップによって連番管理をするロックあり方式の SQL 文（図 5）を考案した。

```
UPDATE クーポン管理 f
WHERE クーポンコード = :クーポンコード AND 発行済枚数 < g ;
INSERT INTO クーポン明細 (クーポンコード, クーポン発行連番, 獲得会員コード, 獲得制限_1 枚限り)
SELECT :クーポンコード, 発行済枚数, :会員コード, 獲得制限_1 枚限り
FROM クーポン管理 WHERE クーポンコード = :クーポンコード;
```

図 5 ロックあり方式の SQL 文

④ロックあり方式では、図 3 の CRUD 図の一部に変更が発生する。

L 主任は、ロックなし方式とロックあり方式の比較を表 3 にまとめ、高負荷試験を実施した。

表 3 ロックなし方式とロックあり方式の比較

方式	ロック 解放待ち	主キー制約違反による 再オペレーション	発行上限枚数に到達後の動作
ロック なし	発生しない	発生する	副問合せで取得する発行済枚数+1 の値が NULL になり、クーポン明細テーブルのクーポン発行連番が NULL のレコードを追加しようとして、主キー制約違反となる。
ロック あり	発生する	発生しない	更新が行われず、クーポン明細テーブルのクーポン発行連番が <span style="border: 1px solid black; padding: 0 5px;">g</span> のレコードを追加しようとして、主キー制約違反となる。

注記 表 3 中の g には、図 5 中の g と同じ字句が入る。

高負荷試験実施の結果、どちらの方式でも最大トラフィック発生時のレスポンス、スループットが規定値以内に収まることが確認できた。そこで、会員による再オペレーションの発生しないロックあり方式を採用することにした。

設問 1 「クーポン発行サービスと施設予約サービスの E-R 図」について、(1)～(3)に答えよ。

(1) 図 1 中の a ～ c に入れる適切なエンティティ間の関連及び属性名を答え、E-R 図を完成させよ。

なお、エンティティ間の関連及び属性名の表記は、図 1 の凡例及び注記に倣うこと。

(2) 図 2 中の d に入れる適切な字句を答えよ。

(3) 本文中の下線①は、どのような業務要件を実現するために行ったものか。  
30 字以内で述べよ。

設問 2 図 4 中の e に入れる適切な字句を答えよ。

設問 3 図 5 中の f , g に入れる適切な字句を答えよ。

設問 4 本文中の下線④について、図 3 中の下線②、下線③の変更後のレコード操作内容を、注記に従いそれぞれ答えよ。