

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。

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 コーポン明細 ADD CONSTRAINT コーポン明細_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タ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 <input type="text" value="g"/> 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 には、図5中の 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 ~ 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 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 に入れる適切な字句を答えよ。

設問3 図5中の , 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。