

問3 データベースの保守・運用に関する次の記述を読んで、設問1～3に答えよ。

E社は、登録している会員向けに健康食品を通信販売する会社である。情報システム部のFさんは、請求情報照会処理と請求情報更新処理から構成される請求情報処理で使用するテーブルの保守とバックアップ計画の作成を任されている。

[テーブルのバックアップ、回復及びロード]

請求情報処理で使用するテーブルのバックアップ、回復及びロードは、関係データベース管理システム(RDBMS)が備える機能を使用して行われている。その概要は、次のとおりである。

#### 1. テーブルのバックアップ

バックアップコマンドによって、テーブルごとにバックアップファイルを作成する。そのファイルを、イメージコピー(以下、ICという)と呼ぶ。

① ICは、取得するページの範囲によって、全体IC、増分IC及び差分ICの3種類に分けられる。

- ・全体ICには、テーブルの全ページが含まれる。
- ・増分ICには、前回の全体IC取得後に変更されたすべてのページが含まれる。
- ・差分ICには、前回の全体IC取得後に変更されたページが含まれる。ただし、前回の全体IC取得以降に差分ICを取得していた場合は、前回の差分IC取得後に変更されたページだけが含まれる。

② ICは、取得する時期によって、オフラインICとオンラインICの2種類に分けられる。

- ・オフラインICの取得中は、ほかの利用者からのアクセスは参照だけが可能である。
- ・オンラインICの取得中は、ほかの利用者からのアクセスはすべての操作(追加、参照、更新、削除)が可能であり、ICには未コミット状態のデータが含まれることがある。

#### 2. テーブルの回復

ICとRDBMSが回復用に取得するログを入力すれば、回復コマンドによって、テーブルを直近の状態まで回復可能である。可能な入力の組合せは、次のとおりである。

- ・全体ICと全体IC取得後のすべてのログ
- ・全体ICとすべての差分IC、最後の差分IC取得後のすべてのログ

- ・全体 IC と最後の増分 IC, その増分 IC 取得後のすべてのログ

### 3. テーブルのロード

ロードコマンドによって、大量のデータを INSERT 文よりも効率よくテーブルに追加できる。ログを出力しないオプションを使った場合、ロード後のテーブルへのアクセスは、参照だけが可能であり、全体 IC を取得すれば、すべての操作が可能になる。

#### [テーブルの構造]

請求情報処理で使用するテーブルの構造を、図 1 に示す。

会員 (会員番号, 住所, 電話番号, 最新請求日, 最新入金日) 請求 (請求番号, 会員番号, 請求日, 明細数, 請求取消日, 入金日) 請求明細 (請求番号, 請求明細番号, 商品コード, 請求額, 返品フラグ) 商品別請求 (商品コード, 請求額合計, 最新請求日)
---

図 1 請求情報処理で使用するテーブルの構造

#### [請求情報照会処理の概要]

請求情報照会処理では、図 2 の SQL 文を使用する。照会 1 と照会 2 のアクセス経路は主索引 (主キーに定義された索引) である。トランザクションの ISOLATION レベルは REPEATABLE READ であり、排他は行単位に行われる。

<pre> 【照会 1】 SELECT X.明細数, Y.請求明細番号, Y.商品コード FROM 請求 X, 請求明細 Y               WHERE X.請求番号=:請求番号 AND X.請求番号=Y.請求番号 【照会 2】 SELECT SUM(請求額) AS 請求額合計 FROM 請求明細               WHERE 請求番号=:請求番号 AND 返品フラグ&lt;&gt;'Y'           </pre>
--

図 2 請求情報照会処理で使用する SQL 文

#### [請求情報更新処理の概要]

請求情報更新処理は、請求登録処理、入金登録処理、請求一括削除処理及び請求取消処理から構成される。

1. 請求登録処理は、会員を四つのグループに均等に分け、グループ単位に日を分けて、それぞれ毎月 1 回、バッチ処理で行う。当処理は、“請求” テーブルと“請求明細” テーブルに行を追加し、請求番号ごとに同期点を取る。“請求” テーブルの請求日と“会員” テーブルの最新請求日には処理日を設定し、“商品別請求” テーブルの請求

額合計に請求明細行の請求額を加算する。請求登録処理の後、1 か月以内に入金される。

2. 入金登録処理は、請求登録処理日に合わせて、バッチ処理で行い、“請求”テーブルの入金日と“会員”テーブルの最新入金日に実際の入金日を設定する。入金登録処理も、毎回、全会員数の 25%を対象にするが、同じ日の請求登録処理対象の会員とは限らないので、二つの処理を合わせて、最大で全会員数の 50%が対象になる。
3. 請求一括削除処理はバッチ処理で行い、月末日に入金日を調べて、請求日が前月の行のうち、入金済の行を削除する。
4. 請求取消処理は、納品後に返品されたとき、“請求”テーブルの請求取消日に返品日を設定し、“請求明細”テーブルの返品フラグを‘N’から‘Y’に更新する。また、“商品別請求”テーブルの請求額合計から返品対象の請求額を減算する。

テーブルのデータに関する主な特徴は、次のとおりである。

1. 年間を通じて扱う全体の商品の種類の数は変わらない。売れる商品は、月平均で全種類の 25%である。
2. 請求番号には、一意な連番を付与する。請求明細番号には、請求番号ごとに 1 から 999 の連番を付与する。
3. “請求明細”テーブルは、一つの請求番号に対して請求明細が平均 2 行存在する。
4. “請求”テーブル及び“請求明細”テーブルは、最大 2 か月間保存する。両テーブルの行数は、最も多い月で平均的な月の 1.2 倍となる。また、今後、年率 20%以上の増加が見込まれる。
5. “商品別請求”テーブルの請求額合計には、年度決算後に 0 を設定する。

請求一括削除処理は、図 3 に示す 2 個の DELETE 文を使って一括削除をする。図 3 中の削除 1 と削除 2 の間で同期点を取るものとする。ページ中の削除された行の領域は、新たな行の追加のときに再利用可能である。

```
【削除 1】 DELETE FROM 請求明細 WHERE 請求番号 IN (SELECT 請求番号 FROM 請求
              WHERE 請求日<=:前月末日 AND 入金日 IS NOT NULL)
【削除 2】 DELETE FROM 請求 WHERE 請求日<=:前月末日 AND 入金日 IS NOT NULL
```

図 3 請求一括削除処理で使用する SQL 文

〔問題の発生・検討〕

Fさんが請求情報更新処理のテストを行ったところ、次の問題A～Cが発生した。

問題A 障害テストを行うために、請求登録処理を意図的に異常終了させた後、そのまま最初から再実行したところ、請求番号を除き、同じ内容の請求データ行が追加された。

問題B 請求一括削除処理では、図3中の削除1を実行したとき、データ量によって失敗することがあった。

問題C 請求一括削除処理中は、図2のSQL文を発行するオンライントランザクションが長時間にわたって排他待ちになることがあった。

Fさんは、発生した問題A～Cについて、それぞれ次の検討A～Cを行った。

検討A 問題Aの原因は、RDBMSが請求番号に常に一意な番号を自動採番しているからであった。Fさんは、“請求”テーブルに制約を追加することによって解決しようとした。

検討B 問題Bの原因は、回復用に取得するログの量が1作業単位の許容上限値を超えたからであった。Fさんは、対策として次の二つの方法を比較し、方法2を選択した。

方法1：1作業単位の許容上限値を20%増加させる。

方法2：一定行数を削除するたびに同期点を取るように、削除1を変更する。

検討C 問題Cに対する対策として、請求情報照会処理のトランザクションのISOLATIONレベルをREPEATABLE READからREAD UNCOMMITTEDに変更したところ、長時間の排他待ちは解消したが、照会2では、実行のタイミングによって、請求額合計が少ないという不整合が発生した。そこで、“請求”テーブルの明細数と“請求明細”テーブルの明細行の数が不一致の場合は、照会結果に含まれないように、図2中の照会2を図4のように修正した。

```
SELECT SUM(Y.請求額) AS 請求額合計 FROM 請求 X, 請求明細 Y
WHERE X.請求番号=:請求番号 AND X.請求番号= Y.請求番号 AND Y.返品フラグ<>'Y'
AND X.明細数=(SELECT  FROM  Z WHERE  )
```

図4 照会2を修正したSQL文（一部未完成）

〔請求一括削除処理の内容・手順の変更〕

Fさんは、検討Cで述べた対策とは別に、問題Cに対する対策として、請求一括削除処理の手順を表1のように変更した。このとき、ある手順からある手順の間、オンライントランザクションからの“請求”テーブルと“請求明細”テーブルへのアクセスを停止することにした。また、手順5の終了後、次の請求一括削除処理を行うまで、差分IC又は増分ICを定期的を取得することにした。

表1 変更後の請求一括削除処理の手順

手順番号	“請求”テーブルと“請求明細”テーブルに対する処理内容
1	全体オンラインICを取得する。
2	残すべき行を選択して抽出したファイルを作成する（注：作成したファイルは、1か月間保存する）。
3	RDBMSの機能を利用して、テーブルと索引を格納した全ページを空にする（注：削除された行と索引のログは出力されず、空になったページは再利用される）。
4	ログを出力しないオプションによって手順2のファイルをテーブルにロードする。
5	全体オフラインICを取得する。

〔テーブルの定期バックアップ計画〕

Fさんはまず、請求情報処理の各処理とテーブルのCRUD及び各処理の稼働情報（処理形態、稼働時間帯及び処理頻度）を調査し、表2のように整理した。

表2 請求情報処理の各処理とテーブルのCRUD及び各処理の稼働情報（一部未完成）

テーブル名 処理	請求	請求明細	会員	商品別 請求	処理 形態	稼働時間帯	処理頻度
請求情報照会	R	R			オン	0時～24時	随時
請求登録	d	e	f	U	バッチ	1時～2時	4回/月
入金登録	U		U		バッチ	1時～2時	4回/月
請求一括削除	D	D			バッチ	23時～24時	1回/月
請求取消	g	U		U	オン	0時～24時	随時

注 CRUD C：追加，R：参照，U：更新，D：削除

処理形態 バッチ：バッチ処理， オン：オンライン処理

次に、月末日の請求一括削除処理の終了後に全体 IC を取得し、請求登録処理と入金登録処理の終了後に、差分 IC 又は増分 IC を取得することにした。バックアップ時間は、その時点の IC の容量にほぼ比例することが分かったので、各テーブルの 1 か月分の全体 IC の容量に対する差分 IC 及び増分 IC の容量の割合を見積もった。その結果を表 3 に示す。ここで、請求取消処理による影響は無視するものとする。

表 3 1 か月分の全体 IC に対する差分 IC 及び増分 IC の容量の割合 (一部未完成)

IC の取得時期と種類	単位 %				
	テーブル名	請求	請求明細	会員	商品別請求
1 か月分の全体 IC		100	100	100	100
1 回目の差分 IC		h	25	50	25
2 回目の差分 IC			25	k	m
3 回目の差分 IC			25		
4 回目の差分 IC			25		
1 回目の増分 IC			25	50	25
2 回目の増分 IC		i	50	100	n
3 回目の増分 IC			75		
4 回目の増分 IC		j	100	l	o

注  は表示していない。

設問1 [問題の発生・検討] 中の検討A～Cについて、(1)～(3)に答えよ。

- (1) 検討Aにおいて、“請求”テーブルに追加すべき制約の適切な列名又は列名の組合せを答えよ。また、その制約の追加目的を、15字以内で述べよ。
- (2) 検討Bにおいて、Fさんは方法2を選択した。方法2が方法1よりも優れている理由を、20字以内で述べよ。
- (3) 検討Cに関する、図4中の  ～  に入れる適切な字句を答えよ。

設問2 表1の変更後の請求一括削除処理の手順について、(1)～(3)に答えよ。

- (1) 表1の手順1の全体オンラインICは、どのような障害の発生に備えて必要か、例を一つ答えよ。
- (2) オンライントランザクションからのアクセス(参照だけの場合とすべての操作の場合)を停止すべき対象範囲(手順)に関する次の記述中の  ～  に入れる適切な手順番号を答えよ。  
参照だけの場合：手順  の開始から手順  の終了まで  
すべての操作の場合：手順  の開始から手順  の終了まで
- (3) 手順5の処理終了後に障害が発生し、手順5のICを利用した回復が必要になったにもかかわらず、そのICが利用不可能だった。しかし、できるだけ障害直前の整合性のある状態に戻したい。回復可能な整合性のある状態のうち、障害直前に最も近い状態は、どの手順番号の後の状態か、その手順番号を答えよ。また、その回復方法を表1や本文中の用語を用いて、25字以内で述べよ。

なお、整合性のある状態とは、未コミット状態のデータが含まれず、かつ、オンライントランザクションを開始できる状態であるものとする。また、回復方法にはバックアップファイルを作成する手順を含めなくてよい。

設問3 [テーブルの定期バックアップ計画] について、(1)、(2)に答えよ。

- (1) 表2中の  ～  に入れる CRUD の中で、最も適切で欠かせないものを一つ答えよ。
- (2) 表3中の  ～  に入れる最も近い値を 25, 50, 75, 100, 125, 150, 175, 200 の中から選んで、表を完成させよ。