

問 3 バッチ処理の性能設計に関する次の記述を読んで、設問 1, 2 に答えよ。

クレジットカード会社として国内外で事業を展開している E 社は、カード利用の増加に伴い、売上登録処理を見直すことにした。F さんがその見直しを担当した。

〔カード業務システムの概要〕

1. カード利用者による商品の購入から入金までの概要

- (1) E 社に登録している加盟店で商品を購入したカード利用者（以下、利用者という）は、商品の代金をカードで支払い、カード売上票の控えを受け取る。
- (2) 加盟店は、カード売上票を売上票保管センタに送る。当該センタは、毎月、クレジットカード会社ごとに、カード売上票を基に売上明細ファイルを作成し、各クレジットカード会社に送る。
- (3) E 社は、毎月末の売上登録処理によって、売上明細ファイルを基に各カードの利用残高を更新する。さらに、利用明細書作成処理によって、カード番号、利用年月日、利用店名、利用金額、支払区分、支払金額、支払金額合計、当月獲得ポイント数などを記載したカード利用明細書を作成し、各利用者に送る。
- (4) 利用者は、代金を銀行口座振替にすることも、ATM から入金することもできる。入金した時点で、即時に入金トランザクションが利用残高を更新する。

2. 売上登録処理の概要

- (1) 売上登録処理で使用する主なテーブル構造を図 1 に、主な列の説明を表 1 に、主なテーブル・列の統計情報及び索引定義情報を表 2 に示す。ここで、ページ及び行は追加された順に並び、同じページに異なるテーブルの行が格納されることはない。また、ページは、ページ単位で順次又はランダムに磁気ディスク装置からバッファに読み込まれる。

加盟店 (加盟店コード, 加盟店名, 加盟店住所, …)
売上明細 (売上明細連番, 登録年月, 加盟店コード, 利用年月日, 利用時刻, 利用店名, カード番号, 利用金額, 支払区分, 支払金額, 端末番号, 伝票番号, …)
利用残高 (カード番号, 利用可能枠, 利用残高, 当月獲得ポイント数, …)
利用明細 (利用明細連番, カード番号, 加盟店コード, 利用年月日, 利用店名, 利用金額, 支払区分, 支払金額, …)

図 1 主なテーブル構造 (一部省略)

表 1 主な列の説明

列名	説明
加盟店コード	E 社に登録している全ての加盟店を一意に識別するコード
売上明細連番	テーブルに行が追加されたとき、自動的に一意かつ昇順で付与される連番
登録年月	テーブルに行が追加されたときの年月
カード番号	全てのカードを一意に識別する番号
利用金額	利用者が加盟店で商品を購入し、その代金のうちカードで支払った金額
利用年月日	利用者が加盟店でカードを利用した年月日
利用可能枠	カードで最大限利用可能な金額
利用残高	現時点で支払が完了していない利用金額の合計。利用金額は利用残高に加算され、入金額は利用残高から減算される。
利用明細連番	テーブルに行が追加されたとき、自動的に一意かつ昇順で付与される連番

表 2 主なテーブル・列の統計情報及び索引定義情報

テーブル名	ページ当たりの行数	列名	列値個数	主索引	副次索引
売上明細	20	売上明細連番	200,000,000	1A	
		登録年月	10		1A
		カード番号	2,000,000		
利用残高	10	カード番号	5,000,000	1A	

注記 主索引、副次索引の数字は索引キーに定義される列の順番を示し、A は昇順を示す。

(2) 売上登録処理は、次の三つのジョブから構成される。現行のジョブ 2 で実行するプログラムの処理の流れと SQL 文を、図 2 に示す。

なお、全てのジョブには、一意なジョブ ID を割り当てる。また、ISOLATION レベルは READ COMMITTED であり、排他制御は行単位で行う。

ジョブ 1 売上明細ファイルの各レコードに登録年月を付加し、加盟店コード、利用年月日順にソートした後、“売上明細”テーブルに追加する。

ジョブ 2 “売上明細”テーブルから 1 行読み込むごとに、カード番号をキーにして“利用残高”テーブルから 1 行読み込み、利用残高に利用金額を加算する（図 2 の②）。カード利用明細書の作成に必要な情報を付加した行を、“利用明細”テーブルに追加する（図 2 の③）。

ジョブ 3 当月を含む直近 10 か月以外の行を、“売上明細”テーブルから削除する。表 2 の統計情報は、ジョブ 3 が終了した直後の値である。

- ① 当年月をホスト変数の登録年月に設定し、当ジョブで処理すべき売上明細連番の範囲をホスト変数の開始連番と終了連番に設定した後、“売上明細”テーブルから行を読むためにカーソルを開く。

```
DECLARE CSR CURSOR WITH HOLD FOR
  SELECT 売上明細連番, カード番号, 利用金額 FROM 売上明細
  WHERE 登録年月 = :登録年月 AND 売上明細連番 BETWEEN :開始連番 AND :終了連番
OPEN CSR
```

- ② カーソルから1行読むごとに、“利用残高”テーブルを更新する。

```
FETCH CSR INTO :売上明細連番, :カード番号, :利用金額
UPDATE 利用残高 SET 利用残高 = 利用残高 + :利用金額
WHERE カード番号 = :カード番号
```

- ③ カード利用明細書の作成に必要な情報を付加した行を、“利用明細”テーブルに追加する。(SQL文は省略)
- ④ カーソルの行がなくなれば⑤に進み、それ以外は②に戻る。
- ⑤ カーソルを閉じる。

```
CLOSE CSR
```

- ⑥ 同期を取り、ジョブを終了する。

図2 現行のジョブ2で実行するプログラムの処理の流れとSQL文（一部省略）

〔売上登録処理の現状調査及び見直し〕

1. Fさんは売上登録処理の現状を調査し、その結果を次のようにまとめた。

- (1) 10 か月間に利用されたカードは、 枚であった。毎月のカード利用明細書には、カード1枚当たり平均 件の利用明細が印字されていた。
- (2) ジョブ2は、毎月平均 行の売上明細を処理するために、“売上明細”テーブルから副次索引を用いて ページを順次に読み込み、行がカード番号順に格納されている“利用残高”テーブルから主索引を用いて延べ最大 ページをランダムに読み込んでいた。
- (3) ジョブ2は、重複していない売上明細連番の範囲を指定することで、多重処理できる設計になっている。しかし、多重処理すると“”テーブルの行でデッドロックが発生するので、これまでは1多重で実行していた。
- (4) ジョブ2は、最後に1回だけ同期を取っていた。これは、ジョブ2が異常終了（以下、失敗という）した場合、失敗の原因を取り除いた後、ジョブ2を最初から単純に再実行できるからであった。
- (5) 処理量が増えるにつれて、メモリリークなどによってジョブ2が失敗したと

き、ジョブ 2 の終了直前に RDBMS によって行われる キ 処理の時間が長くなるおそれがある。

- (6) 処理量が増えるにつれて、“利用残高”テーブルを更新する入金トランザクションの排他ロック待ち時間が長くなるおそれがある。
2. F さんの上司である G 部長は、ジョブ 2 の見直しについて次のように指示した。
- 指示① 多重処理でもジョブ 2 でデッドロックが発生しないようにすること
- 指示② ジョブ 2 の途中から処理を再開できるように、チェックポイントリスタート機能（以下、C/R 機能という）を追加すること
3. F さんは、G 部長の指示に対し、売上登録処理を変更した。その変更内容を表 3 に示す。

表 3 売上登録処理の変更内容

変更箇所	変更前（現行）	変更後
ジョブ 1：売上明細ファイルのソート順序	加盟店コード，利用年月日	カード番号，利用年月日
ジョブ 2：同期を取る契機	最後に 1 回だけ	一定の N 行を処理すること

[チェックポイントリスタート機能の概要]

F さんは、G 部長の指示②に対し、次のようにジョブ 2 が実行するプログラムに C/R 機能を追加した。C/R 機能を追加した後のプログラムの処理の流れと SQL 文を、図 3 に示す。図 3 中の破線で囲んだ部分は、F さんが C/R 機能を追加するために変更した箇所である。

1. “チェックポイント” テーブル（以下，“CKPT” テーブルという）を定義する。
2. ジョブ 2 は、“売上明細” テーブルの N 行を処理するごとに、そのときの売上明細連番をチェックポイントとして“CKPT” テーブルに記録し、かつ、COMMIT 文を発行することで同期を取る。
3. K を 0 以上の整数とするとき、2. の処理の (K+1) 回目の繰返しの途中で処理が失敗した場合、その原因を取り除いて、同じジョブ ID のジョブを再実行する。そのとき、“売上明細” テーブルのジョブ 2 で処理すべき行全体の中の (N×K+1) 行目から処理を再開する。

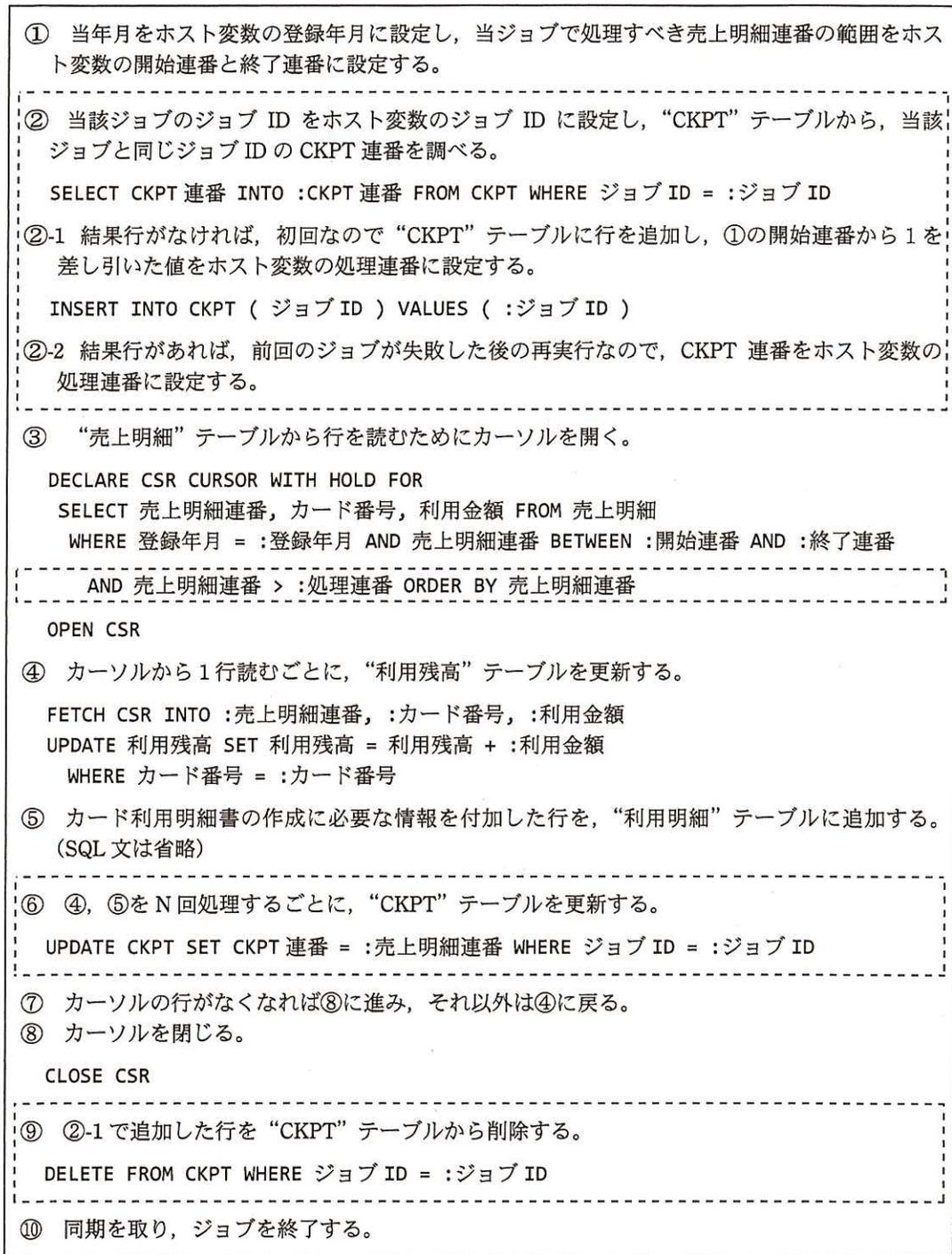


図 3 C/R 機能を追加した後のプログラムの処理の流れと SQL 文（一部省略）

設問1 [売上登録処理の現状調査及び見直し] について、(1)～(3)に答えよ。

- (1) ～ に入れる適切な字句を答えよ。

なお、月ごとに利用されるカード番号に偏りはしないものとする。また、索引のバッファヒット率は 100%であり、ページ中の行をアクセスするとき、次にアクセスするページはバッファにないものとする。

- (2) 変更前（現行）のジョブ2を多重処理すると“”テーブルの行でデッドロックが発生する理由を、関係する列名を含めて 30 字以内で述べよ。
- (3) 売上登録処理を変更した後にジョブ2を1多重で実行したところ、“利用残高”テーブルから読み込まれるページ数が減り、処理時間が短縮した。ジョブ1で売上明細ファイルのソート順を変えたことによって、なぜジョブ2で“利用残高”テーブルから読み込まれるページ数が減ったのか。その読み込まれるページ数とページ数が減った理由を、30 字以内で述べよ。

設問2 [チェックポイントリスタート機能の概要] について、(1)～(4)に答えよ。

- (1) “CKPT” テーブルのテーブル構造を示せ。主キーには下線を引くこと。
- (2) 図3では同期を取る処理が漏れている。ジョブが失敗した後に同じジョブを再実行したとき、 $(N \times K + 1)$ 行目から処理を再開するためには、図3中に同期を取る処理が必要である。図3中の④～⑧のうち、どの番号の後に同期を取る処理を挿入すべきか。適切な番号を一つ答えよ。
- (3) 図3中の③のSELECT文においてWHERE句に必ずしも必要でない述語がある。その述語とは何かを答え、必要でない理由を、30字以内で述べよ。

なお、ホスト変数には適切な値を設定して、多重処理を行うものとする。

- (4) 排他ロック待ちタイムアウト時間が T 秒のとき、図3中の④、⑤を合わせた処理に平均 A 秒、⑥の処理に平均 B 秒、同期を取る処理に平均 C 秒の時間を要すると仮定する。この場合、他のトランザクションに排他ロック待ちタイムアウトをさせないためには、少なくとも N 行更新するごとに同期を取らなければならない。N を見積もる次の計算式の , に入れる適切な式を T, A, B 及び C を用いて答えよ。

$$N < \frac{\text{a}}{\text{b}}$$