問2　データベースの実装に関する次の記述を読んで，設問1～3に答えよ。

クレジットカード会社のC社では，キャッシュレス決済の普及に伴いカード決済システムのオンライントランザクションの処理量が増えている。情報システム部のFさんは，将来の処理量から懸念される性能低下の対策を検討することになった。

〔RDBMSの主な仕様〕

1. アクセス経路，区分化，再編成

(1) アクセス経路は，RDBMSによって表探索又は索引探索に決められる。表探索では，索引を使わずに先頭ページから順に全行を探索する。索引探索では，WHERE句中の述語に適した索引によって絞り込んでから表の行を読み込む。

(2) テーブルごとに一つ又は複数の列を区分キーとし，区分キーの値に基づいて物理的な表領域に分割することを区分化という。

(3) 区分方法にはハッシュとレンジの二つがある。どちらも，テーブルを検索するSQL文のWHERE句の述語に区分キー列を指定すると，区分キー列で特定した区分だけを探索する。

① ハッシュは，区分キー値を基にRDBMSが生成するハッシュ値によって一定数の区分に行を分配する方法である。

② レンジは，区分キー値の範囲によって区分に行を分配する方法である。

(4) テーブル又は区分を再編成することによって，行を主キー順に物理的に並び替えることができる。また，各ページ中に指定した空き領域を予約することができる。

(5) INSERT文で行を挿入するとき，RDBMSは，主キー値の並びの中で，挿入行の主キー値に近い行が格納されているページに空き領域があればそのページに，なければ表領域の最後のページに格納する。最後のページに空き領域がなければ，新しいページを表領域の最後に追加する。

2. データ入出力とログ出力

(1) データとログはそれぞれ別のディスクに格納される。同じディスクに対し同時に入出力は行われないものとする。

(2) データ入出力とログ出力は4,000バイトのページ単位に行われる。

(3) データバッファはテーブルごとに確保される。

(4) ページをランダムに入出力する場合，SQL 処理中の CPU 処理と入出力処理は並行して行われない。これを同期データ入出力処理と呼び，SQL 処理時間は次の式で近似できる。

SQL 処理時間 ＝ CPU 時間 ＋ 同期データ入出力処理時間

(5) ページを順次に入出力する場合，SQL 処理中の CPU 処理と入出力処理は並行して行われる。これを非同期データ入出力処理と呼び，SQL 処理時間は次の式で近似できる。ここで関数 MAX は引数のうち最も大きい値を返す。

SQL 処理時間 ＝ MAX（CPU 時間，非同期データ入出力処理時間）

(6) 行を挿入，更新，削除した場合，変更内容がログとして RDBMS に一つ存在するログバッファに書き込まれる。ログバッファが一杯の場合，トランザクションの INSERT 文，UPDATE 文，DELETE 文の処理は待たされる。

(7) ログは，データより先にログバッファからディスクに出力される。これをログ出力処理と呼ぶ。このとき，トランザクションのコミットはログ出力処理の完了まで待たされる。ログ出力処理は，次のいずれかの事象を契機に行われる。

① ログバッファが一杯になった。

② トランザクションがコミット又はロールバックを行った。

③ あるテーブルのデータバッファが変更ページによって一杯になった。

〔カード決済システムの概要〕

1. テーブル

主なテーブルのテーブル構造を図 1，将来の容量見積りを表 1 に示す。各テーブルの主キーには索引が定義されており，索引キーを構成する列の順はテーブルの列の順と同じである。

加盟店（加盟店番号，加盟店名，住所，電話番号，…） オーソリ履歴（カード番号，利用日，オーソリ連番，加盟店番号，利用金額，審査結果，請求済フラグ，請求日，…）

図 1　主なテーブルのテーブル構造（一部省略）

表1　主なテーブルの将来の容量見積り

テーブル名	行長（バイト）	見積行数	1ページ当たりの行数	ページ数	容量（Gバイト）
加盟店	1,000	100万	4	25万	1
オーソリ履歴	200	480億	20	24億	9,600

2. オーソリ処理（オンライン処理）

オーソリ処理は，会員がカードで支払う際にカード有効期限，与信限度額を超過していないかなどを判定する処理である。判定した結果，可ならば審査結果を‘Y’に，否ならば‘N’に設定した行を“オーソリ履歴”テーブルに挿入する。オーソリ処理は最大100多重で処理される。“オーソリ履歴”テーブルには直近5年分を保持する。

3. 利用明細抽出処理（バッチ処理）

請求書作成に必要な1か月分の利用明細の記録を“オーソリ履歴”テーブルから抽出しファイルに出力する。

〔参照処理の性能見積り〕

将来の処理時間が懸念される利用明細抽出処理のSQL文を，図2に示す。

```
SELECT A.カード番号，A.利用日，A.オーソリ連番，A.利用金額，B.加盟店名
  FROM オーソリ履歴 A，加盟店 B
 WHERE A.審査結果 = 'Y' AND A.利用日 BETWEEN :hv1 AND :hv2
   AND A.加盟店番号 = B.加盟店番号
```

注記　ホスト変数hv1，hv2は，請求対象月の初日と末日をそれぞれ表す。

図2　利用明細抽出処理のSQL文

Fさんは，利用明細抽出処理の処理時間を，次のように見積もった。

1. このSQL文での表の結合方法を調べたところ，“オーソリ履歴”テーブルを外側，“加盟店”テーブルを内側とする入れ子ループ法だった。“オーソリ履歴”テーブルのアクセス経路は表探索だったので，［　a　］ページを非同期に読み込む。
2. ディスク転送速度を100Mバイト／秒と仮定すれば，［　a　］ページを非同期に読み込むデータ入出力処理時間は，［　b　］秒である。

3. カード数を 1,000 万枚，カード・月当たり平均オーソリ回数を 80 回，審査結果が全て可であると仮定すると，“オーソリ履歴”テーブルの結果行数は，[c] 行である。これに掛かる CPU 時間は，96,000 秒である。

4. この結合では，外側の表の結果行ごとに“加盟店”テーブルの主キー索引を索引探索し，“加盟店”テーブルを 1 行，ランダムに合計 [d] 回読み込む。

5. 索引はバッファヒット率 100%，テーブルはバッファヒット率 0%と仮定すれば，“加盟店”テーブルを合計で [e] ページを同期的に読み込むことになる。同期読込みにページ当たり 1 ミリ秒掛かると仮定すれば，同期データ入出力処理時間は [f] 秒である。

6. 内側の表の索引探索と結合に掛かる CPU 時間は，1 結果行当たり 0.01 ミリ秒掛かると仮定すれば，[g] 秒である。

7. 外側の表の CPU 時間は 96,000 秒，内側の表の CPU 時間は [g] 秒，内側の表の同期データ入出力処理時間は [f] 秒なので，SQL 文の処理時間を [h] 秒と見積もった。

処理時間が長くなることが分かったので時間短縮のため，次の 2 案を検討した。

案 1　“加盟店”テーブルのデータバッファを増やしバッファヒット率 100%にする。

案 2　“オーソリ履歴”テーブルの利用日列をキーとする副次索引を追加する。

〔“オーソリ履歴”テーブルの区分化〕

F さんは，上司である G 氏から，次の課題の解決策の検討を依頼された。

課題 1　月末近くに起きるオーソリ処理の INSERT 文の性能低下を改善すること

課題 2　将来懸念される利用明細抽出処理の処理時間を短縮すること

課題 3　月初に行う“オーソリ履歴”テーブル再編成の処理時間を短縮すること

F さんは，課題を解決するために，“オーソリ履歴”テーブルを区分化することにし，区分キーについて表 2 に示す 3 案を評価した。いずれの案も 60 区分に行を均等に分配する前提であり，図 2 の SQL 文を基に区分化に対応した SQL 文を作成した。作成した SQL 文の WHERE 句を図 3 に示す。利用明細抽出処理及び再編成について，アクセスする総ページ数が最小になるようにジョブを設計した。このときジョブは，

必要に応じて並列実行させる。

表 2　課題ごとに各案を評価した結果（未完成）

	案 A	案 B	案 C
区分方法	ハッシュ	レンジ	レンジ
区分キー	カード番号	カード番号	利用日（1 か月を 1 区分）
課題 1	評価：○	評価：○	評価：×
課題 2	評価：× ジョブ当たり 60 区分，24 億ページを探索	評価：○ ジョブ当たり [　　] 区分，[　　] ページを探索	評価：○ ジョブ当たり [イ] 区分，[ロ] ページを探索
課題 3	評価：○	評価：○	評価：○

注記 1　○：課題を解決する。×：課題を解決しない。

注記 2　網掛け部分は表示していない。

```
WHERE A.審査結果 = 'Y' AND A.利用日 BETWEEN :hv1 AND :hv2
  AND A.カード番号 BETWEEN :hv3 AND :hv4 AND A.加盟店番号 = B.加盟店番号
```

注記 1　ホスト変数 hv1，hv2 は，請求対象月の初日と末日をそれぞれ表す。

注記 2　ホスト変数 hv3，hv4 は，カード番号の範囲（レンジ）の始まりと終わりをそれぞれ表す。

図 3　区分化に対応した SQL 文の WHERE 句

〔更新処理の多重化〕

“オーソリ履歴”テーブルの請求済フラグと請求日を更新する処理も同様に，将来の処理時間が懸念された。更新処理の SQL 文を図 4 に示す。更新処理はバッチ処理であり，カーソルを使用して 1,000 行を更新するごとにコミットする。

```
UPDATE オーソリ履歴 SET 請求済フラグ = 'Y', 請求日 = :hv1
 WHERE 利用日 BETWEEN :hv2 AND :hv3 AND  カード番号 BETWEEN :hv4 AND :hv5
```

注記 1　ホスト変数 hv1 は，請求日を表す。

注記 2　ホスト変数 hv2，hv3 は，請求対象月の初日と末日をそれぞれ表す。

注記 3　ホスト変数 hv4，hv5 は，カード番号の範囲（レンジ）の始まりと終わりをそれぞれ表す。

図 4　更新処理の SQL 文

F さんは，次のように，区分化と併せて，更新処理を多重化することにした。

1.　“オーソリ履歴”テーブルについて，カード番号，利用日の順の組で区分キーとし，レンジによって区分化する。

2. 区分ごとのジョブで更新処理を多重化する。

3. 更新処理を多重化しても競合しないように，各区分を異なるディスクに配置し，データバッファを十分に確保する。

設問1 〔参照処理の性能見積り〕について，(1)～(3)に答えよ。

(1) 本文中の［ a ］～［ h ］に入れる適切な数値を答えよ。

(2) 案1について，“加盟店”テーブルのデータバッファを増やすのはなぜか。また，“オーソリ履歴”テーブルはデータバッファを増やさないのはなぜか。アクセス経路に着目し，それぞれ理由を25字以内で述べよ。

(3) 案2を適用した場合，オーソリ処理の処理時間が長くなると考えられる。その理由を25字以内で述べよ。

設問2 〔“オーソリ履歴”テーブルの区分化〕について，(1)～(4)に答えよ。

(1) 課題1について，案Aと案Bは案Cに比べてオーソリ処理のINSERT文の性能が良いと考えられる。その理由を25字以内で具体的に述べよ。

(2) 課題2について，区分限定の表探索を行う場合，1ジョブが探索する区分数及びページ数の最小値はそれぞれ幾らか。表2中の［ イ ］，［ ロ ］に入れる適切な数値を答えよ。

(3) 課題2について，案Aではカード番号にBETWEEN述語を追加しても改善効果を得られないと考えられる。その理由を30字以内で具体的に述べよ。

(4) 課題3について，特に案Cは，区分キーの特徴から，案Aと案Bに比べて再編成の効率が良いと考えられる。その理由を20字以内で具体的に述べよ。

設問3 〔更新処理の多重化〕について，(1)，(2)に答えよ。

(1) ジョブの多重度を幾ら増やしても，それ以上は更新処理全体の処理時間を短くできない限界がある。このときボトルネックになるのはログである。その理由をRDBMSの仕様に基づいて30字以内で述べよ。

(2) 更新処理では1,000行更新するごとにコミットしているが，仮に1行更新するごとにコミットすると，更新処理の処理時間のうち何がどのように変わるか。本文中の用語を用いて25字以内で述べよ。