問３　データベースの実装と性能に関する次の記述を読んで，設問に答えよ。

事務用品を関東地方で販売するＣ社は，販売管理システム（以下，システムという）にRDBMSを用いている。

〔RDBMSの仕様〕

1.　表領域

(1) テーブル及び索引のストレージ上の物理的な格納場所を，表領域という。

(2) RDBMS とストレージとの間の入出力単位を，ページという。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。

2.　再編成，行の挿入

(1) テーブルを再編成することで，行を主キー順に物理的に並び替えることができる。また，再編成するとき，テーブルに空き領域の割合（既定値は 30%）を指定した場合，各ページ中に空き領域を予約することができる。

(2) INSERT 文で行を挿入するとき，RDBMS は，主キー値の並びの中で，挿入行のキー値に近い行が格納されているページを探し，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し，格納する。

〔業務の概要〕

1.　顧客，商品，倉庫

(1) 顧客は，C 社の代理店，量販店などで，顧客コードで識別する。顧客には C 社から商品を届ける複数の発送先があり，顧客コードと発送先番号で識別する。

(2) 商品は，商品コードで識別する。

(3) 倉庫は，1 か所である。倉庫には複数の棚があり，一連の棚番号で識別する。商品の容積及び売行きによって，一つの棚に複数種類の商品を保管することも，同じ商品を複数の棚に保管することもある。

2.　注文の入力，注文登録，在庫引当，出庫指示，出庫の業務の流れ

(1) 顧客は，C 社が用意した画面から注文を希望納品日，発送先ごとに入力し，C 社の EDI システムに蓄える。注文は，単調に増加する注文番号で識別する。注文

する商品の入力順は自由で，入力後に商品の削除も同じ商品の追加もできる。

(2) C 社は，毎日定刻（9 時と 14 時）に注文を締める。EDI システムに蓄えた注文をバッチ処理でシステムに登録後，在庫を引き当てる。

(3) 出庫指示書は，当日が希望納品日である注文ごとに作成し，倉庫の出庫担当者（以下，ピッカーという）を決めて，作業開始の予定時刻までにピッカーの携帯端末に送信する。携帯端末は，棚及び商品のバーコードをスキャンする都度，システム中のオンラインプログラムに電文を送信する。

(4) 出庫は，ピッカーが出庫指示書の指示に基づいて 1 件の注文ごとに行う。

① 棚の通路の入口で，携帯端末から出庫開始時刻を伝える電文を送信する。

② 棚番号の順に進みながら，指示された棚から指示された商品を出庫する。

③ 商品を出庫する都度，携帯端末で棚及び商品のバーコードをスキャンし，商品を台車に積む。ただし，一つの棚から商品を同時に出庫できるのは 1 人だけである。また，順路は 1 方向であるが，通路は追い越しができる。

④ 台車に積んだ全ての商品を，指定された段ボール箱に入れて梱（こん）包する。

⑤ 別の携帯端末で印刷したラベルを箱に貼り，ラベルのバーコードをスキャンした後，梱包した箱を出荷担当者に渡すことで 1 件の注文の出庫が完了する。

〔システムの主なテーブル〕

システムの主なテーブルのテーブル構造を図 1 に，主な列の意味・制約を表 1 に示す。主キーにはテーブル構造に記載した列の並び順で主索引が定義されている。

顧客（顧客コード，顧客名，…）
顧客発送先（顧客コード，発送先番号，発送先名，発送先住所，…）
商品（商品コード，商品名，販売単価，注文単位，商品容積，…）
在庫（商品コード，実在庫数，引当済数，引当可能数，基準在庫数，…）
棚（棚番号，倉庫内位置，棚容積，…）
棚別在庫（棚番号，商品コード，棚別実在庫数，出庫指示済数，出庫指示可能数，…）
ピッカー（ピッカーID，ピッカー氏名，…）
注文（注文番号，顧客コード，注文日，締め時刻，希望納品日，発送先番号，…）
注文明細（注文番号，注文明細番号，商品コード，注文数，注文額，注文状態，…）
出庫（出庫番号，注文番号，ピッカーID，出庫日，出庫開始時刻，…）
出庫指示（出庫番号，棚番号，商品コード，注文番号，注文明細番号，出庫数，出庫時刻，…）

図 1　テーブル構造（一部省略）

表1　主な列の意味・制約（一部省略）

列名	意味・制約
棚番号	1以上の整数：棚の並び順を表す一連の番号
注文状態	0：未引当，1：引当済，2：出庫指示済，3：出庫済，4：梱包済，5：出荷済，…
出庫時刻	棚から商品を取り出し，商品のバーコードをスキャンしたときの時刻

〔システムの注文に関する主な処理〕

注文登録，在庫引当，出庫指示の各処理をバッチジョブで順に実行する。出庫実績処理は，携帯端末から電文を受信するオンラインプログラムで実行する。バッチ及びオンラインの処理のプログラムの主な内容を，表2に示す。

表2　処理のプログラムの主な内容

処理		プログラムの内容
バッチ	注文登録	・顧客が入力したとおりに注文及び商品を，それぞれ“注文”及び“注文明細”に登録し，注文ごとにコミットする。
	在庫引当	・注文状態が未引当の“注文明細”を主キー順に読み込み，その順で“在庫”を更新し，“注文明細”の注文状態を引当済に更新して注文ごとにコミットする。
	出庫指示	・当日が希望納品日である注文の出庫に，当日に出勤したピッカーを割り当てる。 ・注文状態が引当済の“注文明細”を主キー順に読み込む。 ・ピッカーの順路が1方向となる出庫指示を“出庫指示”に登録する。 ・“出庫指示”を主キー順に読み込み，その順で“棚別在庫”を更新し，“注文明細”の注文状態を出庫指示済に更新する。 ・注文ごとにコミットし，出庫指示書をピッカーの携帯端末に送信する。
オンライン	出庫実績	・出庫開始を伝える電文を携帯端末から受信すると，当該注文について，“出庫”の出庫開始時刻を出庫を開始した時刻に更新する。 ・棚及び商品のバーコードの電文を携帯端末から受信すると，当該商品について，“棚別在庫”，“在庫”を更新し，また“出庫指示”の出庫時刻を棚から出庫した時刻に，“注文明細”の注文状態を出庫済に更新してコミットする。 ・商品を梱包した箱のラベルのバーコードの電文を携帯端末から受信すると，“注文明細”の注文状態を梱包済に更新し，コミットする。

注記1　二重引用符で囲んだ名前は，テーブル名を表す。
注記2　いずれの処理も，ISOLATION レベルは READ COMMITTED で実行する。

〔ピーク日の状況と対策会議〕

注文量が特に増えたピーク日に，朝のバッチ処理が遅延し，出庫作業も遅延する事態が発生した。そこで，関係者が緊急に招集されて会議を開き，次のように情報を収集し，対策を検討した。

1. システム資源の性能に関する基本情報

次の情報から特定のシステム資源に致命的なボトルネックはないと判断した。

(1) ページングは起きておらず，CPU 使用率は 25%程度であった。

(2) バッファヒット率は 95%以上で高く，ストレージの入出力処理能力（IOPS，帯域幅）には十分に余裕があった。

(3) ロック待ちによる大きな遅延は起きていなかった。

2. 再編成の要否

アクセスが多かったのは“注文明細”テーブルであった。この 1 年ほど行の削除は行われず，再編成も行っていないことから，時間が掛かる行の削除を行わず，直ちに再編成だけを行うことが提案されたが，この提案を採用しなかった。なぜならば，当該テーブルへの行の挿入では予約された空き領域が使われないこと，かつ空き領域の割合が既定値だったことで，割り当てたストレージが満杯になるリスクがあると考えられたからである。

3. バッチ処理のジョブの多重化

バッチ処理のスループット向上のために，ジョブを注文番号の範囲で分割し，多重で実行することが提案されたが，デッドロックが起きるリスクがあると考えられた。そこで，どの処理とどの処理との間で，どのテーブルでデッドロックが起きるリスクがあるか，表 3 のように整理し，対策を検討した。

表 3　デッドロックが起きるリスク（未完成）

ケース	処理名	処理名	テーブル名	リスクの有無	リスクの有無の判断理由
1	在庫引当	在庫引当	在庫	ある	a
2	出庫指示	出庫指示	棚別在庫	ない	b
3	在庫引当	出庫指示	注文明細	ない	c

注記　ケース 3 は，ジョブの進み具合によって異なる処理のジョブが同時に実行される場合を表す。

4. 出庫作業の遅延原因の分析

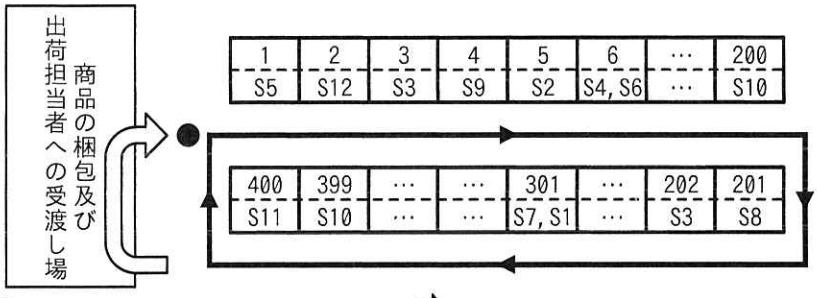
出庫作業の現場の声を聞いたところ，特定の棚にピッカーが集中し，棚の前で待ちが発生したらしいことが分かった。そこで，棚の前での待ち時間と棚から商品を取り出す時間の和である出庫間隔時間を分析した。出庫間隔時間は，ピッカーが出庫指示書の 1 番目の商品を出庫する場合では当該注文の出庫開始時刻からの

時間，2 番目以降の商品の出庫の場合では一つ前の商品の出庫時刻からの時間である。出庫間隔時間が長かった棚と商品が何かを調べた SQL 文の例を表 4 に，このときの棚と商品の配置，及びピッカーの順路を図 2 に示す。

表 4　SQL 文の例（未完成）

SQL 文（上段：目的，下段：構文）
ホスト変数 h に指定した出庫日について，出庫間隔時間の合計が長かった棚番号と商品コードの組合せを，出庫間隔時間の合計が長い順に調べる。
WITH TEMP(出庫番号，ピッカーID，棚番号，商品コード，出庫時刻，出庫間隔時間) AS (SELECT A.出庫番号，A.ピッカーID，B.棚番号，B.商品コード，B.出庫時刻，B.出庫時刻 - COALESCE(LAG(B.出庫時刻) OVER (PARTITION BY [x] ORDER BY B.出庫時刻), A.出庫開始時刻) AS 出庫間隔時間 FROM 出庫 A JOIN 出庫指示 B ON A.出庫番号 = B.出庫番号 AND 出庫日 = CAST(:h AS DATE)) SELECT 棚番号，商品コード，SUM(出庫間隔時間) AS 出庫間隔時間合計 FROM TEMP GROUP BY 棚番号，商品コード ORDER BY 出庫間隔時間合計 DESC

注記　ここでの LAG 関数は，ウィンドウ区画内で出庫時刻順に順序付けられた各行に対して，現在行の 1 行前の出庫時刻を返し，1 行前の行がないならば，NULL を返す。



凡例　●通路入口　→出庫作業の順路　商品の梱包及び受渡し場を通る順路

注記　太枠は一つの棚を表し，枠内の上段は棚番号，下段はその棚に保管した商品の商品コードを表す。

図 2　棚と商品の配置，及びピッカーの順路（一部省略）

表 4 中の [x] に，B.出庫番号，A.ピッカーID，B.棚番号のいずれか一つを指定することが考えられた。分析の目的が，特定の棚の前で長い待ちが発生していたことを実証することだった場合，[x] に [あ] を指定すると，棚の前での待ち時間を含むが，商品の梱包及び出荷担当者への受渡しに掛かった時間が含まれてしまう。[い] を指定すると，棚の前での待ち時間が含まれないので，分析の目的を達成できない。

分析の結果，棚 3 番の売行きの良い商品 S3（商品コード）の出庫で長い待ちが発生したことが分かった。そこで，出庫作業の順路の方向を変えない条件で，多くのピッカーが同じ棚（ここでは，棚 3 番）に集中しないように出庫指示を作成する対策が提案された。しかし，この対策を適用すると，表 3 中のケース 2 でデッドロックが起きるリスクがあると予想した。

例えば，あるピッカーに，1 番目に棚 3 番の商品 S3 を出庫し，2 番目に棚 6 番の商品 S6 を出庫する指示を作成するとき，別のピッカーには，1 番目に棚 [う] の商品 [え] を出庫し，2 番目に棚 [お] の商品 [か] を出庫する指示を同時に作成する場合である。

設問 1　“2.　再編成の要否” について答えよ。

(1) 注文登録処理が “注文明細” テーブルに行を挿入するとき，再編成で予約した空き領域が使われないのはなぜか。行の挿入順に着目し，理由を RDBMS の仕様に基づいて，40 字以内で答えよ。

(2) 行の削除を行わず，直ちに再編成だけを行うと，ストレージが満杯になるリスクがあるのはなぜか。前回の再編成の時期及び空き領域の割合に着目し，理由を RDBMS の仕様に基づいて，40 字以内で答えよ。

設問 2　“3.　バッチ処理のジョブの多重化” について答えよ。

(1) 表 3 中の [a] ～ [c] に入れる適切な理由を，それぞれ 30 字以内で答えよ。ここで，在庫は適正に管理され，欠品はないものとする。

(2) 表 3 中のケース 1 のリスクを回避するために，注文登録処理又は在庫引当処理のいずれかのプログラムを変更したい。どちらかの処理を選び，選んだ処理の処理名を答え，プログラムの変更内容を具体的に 30 字以内で答えよ。ただし，コミット単位と ISOLATION レベルを変更しないこと。

設問 3　“4.　出庫作業の遅延原因の分析” について答えよ。

(1) 本文中の [あ] ～ [か] に入れる適切な字句を答えよ。

(2) 下線の対策を適用した場合，表 3 中のケース 2 で起きると予想したデッドロックを回避するために，出庫指示処理のプログラムをどのように変更すべきか。具体的に 40 字以内で答えよ。ただし，コミット単位と ISOLATION レベルを変更しないこと。