

問3 オフィスじゅう器メーカーの在庫管理システムのデータベース実装に関する次の記述を読んで、設問に答えよ。

オフィスじゅう器メーカーのY社は、RDBMSを用いた在庫管理システムを稼働させてから10年が経ち、システム更改を予定している。そこで、Kさんが改善点の検討を任された。

[RDBMSの主な仕様]

1. アクセス経路

- (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読み込む。索引探索では、索引によって絞り込んでから表の行を読み込む。
- (2) 索引探索を選択するためには、WHERE句又はON句のANDだけで結ばれた一つ以上の等値比較の述語の対象列が、索引キーの全体又は先頭から連続した一つ以上の列に一致していなければならない。
- (3) 索引探索のとき、SELECT文で取得対象とする列と、探索条件で絞り込みに利用する列が、索引キーに全て含まれている場合は、表の行を読み込まずに、索引へのアクセスだけで結果を返す。

2. 排他制御

- (1) 排他制御では行単位で共有ロック又は専有ロックを取得する。共有ロックを取得している間、他のトランザクションから対象行への共有ロックの取得は可能であり、専有ロックの取得は共有ロックの解放待ちとなる。専有ロックを取得している間、他のトランザクションから対象行への共有ロック及び専有ロックの取得は、専有ロックの解放待ちとなる。
- (2) アクセス経路に従って、探索した行をロックの対象とする。
- (3) SELECT文にFOR UPDATE句を指定すると、ISOLATIONレベルにかかわらず、対象行の専有ロックを取得し、トランザクション終了時に解放する。

3. 時刻印型の列

- (1) テーブル定義で、時刻印型の列（以下、TS列という）のDEFAULT句にCURRENT_TIMESTAMPを指定することで、行が挿入されたとき、TS列に現在時刻

印を自動的に設定することができる。

- (2) 行が更新されたときに TS 列を CURRENT_TIMESTAMP で更新するトリガーを定義することで、TS 列を現在時刻印で自動的に更新することができる。

〔在庫管理システムの概要〕

1. テーブル構造

主なテーブルのテーブル構造を図 1 に示す。主キーには主索引が定義されている。

倉庫	(倉庫コード, 倉庫名)
部品	(部品番号, 部品名, 部品単価)
在庫	(倉庫コード, 部品番号, 在庫数量, 引当済在庫数量, 最終更新 TS)
出庫	(出庫番号, 出庫年月日, 出庫倉庫コード, 部品番号, 出庫数量, 処理状況, 最終更新 TS)
入庫	(入庫番号, 入庫年月日, 入庫倉庫コード, 部品番号, 入庫数量, 最終更新 TS)

注記 1 年月日を示す列のデータ型は、日付型とする。

注記 2 最終更新 TS 列は、挿入・更新時の現在時刻印で自動的に更新される TS 列とする。

図 1 主なテーブルのテーブル構造

2. 業務の概要

- (1) 各地の生産拠点には、組立工場と、これに隣接する倉庫がある。
- (2) 倉庫からの部品の出庫には、倉庫から隣接する組立工場に出庫する場合と、倉庫から他の生産拠点の倉庫に出庫する場合がある。
- (3) 倉庫は、倉庫コードで識別する。部品は、部品番号で識別する。
- (4) 部品の在庫は、倉庫と部品の組合せで管理する。倉庫内に存在する在庫の数量を在庫数量と呼ぶ。このうち、出庫対象として引き当てた数量を、引当済在庫数量と呼ぶ。また、在庫数量から引当済在庫数量を引いた数量を、出庫可能在庫数量と呼ぶ。
- (5) 部品の出庫は、出庫要求、在庫引当、出庫、在庫反映の流れで、それぞれのアプリケーションプログラム（以下、AP という）を用いて処理し、“出庫” テーブルに処理状況を記録する。
 - ・ 出庫要求とは、部品の出庫要求を受け付けて、“出庫” テーブルに行を追加する処理である。処理状況を‘要求発生’にする。
 - ・ 在庫引当とは、出庫要求に応じて倉庫内の在庫を引き当てる処理である。指

定された倉庫コード，部品番号，出庫数量の出庫が可能かどうかチェックし，出庫が可能であれば“在庫”テーブルの引当済在庫数量を更新し，処理状況を‘引当実施’にする。

- ・ 出庫とは，倉庫から在庫の引当ができた部品を出す処理である。処理状況を‘出庫済’にする。

- ・ 在庫反映とは，処理状況が‘出庫済’の全ての出庫を対象に，“在庫”テーブルの在庫数量及び引当済在庫数量から引き落として，処理状況を‘在庫反映済’にする一括処理である。“在庫反映”AP は，毎日の業務終了時に実行する。

(6) 入庫とは，部品メーカーから納品された，又は他の生産拠点の倉庫から出庫された部品を倉庫に入れることである。運ばれてきた部品を倉庫内の集積所に蓄積し，当日の入庫予定分がそろったら順に入庫を処理し，在庫を更新する。入庫による在庫の更新は，在庫反映と同時に実施しない。

〔在庫履歴分析手段の見直し〕

日別の在庫数量を集計する AP を用いて，在庫数量の日別の推移状況を分析している。この AP では，“在庫”，“出庫”及び“入庫”テーブルを参照して，在庫数量を集計しており，データ量の増加に伴って処理時間が長くなっていた。この対策として，図 2 に示す“在庫履歴”テーブルを作成して，在庫数量の推移を蓄積していくことにした。

在庫履歴（倉庫コード，部品番号，履歴年月日，在庫数量）

注記 年月日を示す列のデータ型は，日付型とする。

図 2 “在庫履歴”テーブルのテーブル構造

K さんは，“在庫”，“出庫”及び“入庫”テーブルからこれまでの在庫履歴を作成する方法を検討した。

全ての倉庫の全ての部品について，在庫履歴を作成する初日時点の在庫数量の行をあらかじめ作成しておき，在庫履歴を蓄積する年月日分繰り返し実行する SQL1 を設計した。

また，区間数 7 の中央移動平均のグラフを作成したいとの要望があった。区間数 7

の中央移動平均は、ある年月日とその前後 3 日間の計 7 日間の平均値である。計 7 日間のデータがそろわない期間は対象外である。K さんは、作成した在庫履歴のデータから区間数 7 の中央移動平均用データを加工する SQL2 を設計した。

設計した SQL 文を、表 1 に示す。

表 1 在庫履歴用の SQL 文（未完成）

SQL	SQL 文の構文（上段：目的，下段：構文）
SQL1	全ての倉庫の全ての部品について、指定した年月日の業務終了時点の在庫数量を算出する。 <pre> INSERT INTO 在庫履歴(倉庫コード, 部品番号, 履歴年月日, 在庫数量) WITH 前日在庫 AS (SELECT 倉庫コード, 部品番号, 履歴年月日, 在庫数量 FROM 在庫履歴 WHERE 履歴年月日 = PREVIOUS_DAY(:hv1)), 出庫状況 AS (SELECT 出庫倉庫コード, 部品番号, 出庫年月日, [a] AS 当日出庫数量 FROM 出庫 WHERE 出庫年月日 = CAST(:hv1 AS DATE) AND 処理状況 = '在庫反映済' GROUP BY 出庫倉庫コード, 部品番号, 出庫年月日), 入庫状況 AS (SELECT 入庫倉庫コード, 部品番号, 入庫年月日, [b] AS 当日入庫数量 FROM 入庫 WHERE 入庫年月日 = CAST(:hv1 AS DATE) GROUP BY 入庫倉庫コード, 部品番号, 入庫年月日) SELECT A.倉庫コード, A.部品番号, CAST(:hv1 AS DATE), A.在庫数量 - COALESCE([c], 0) + COALESCE([d], 0) AS 在庫数量 FROM 前日在庫 A LEFT OUTER JOIN 出庫状況 B ON A.倉庫コード = B.出庫倉庫コード AND A.部品番号 = B.部品番号 LEFT OUTER JOIN 入庫状況 C ON A.倉庫コード = C.入庫倉庫コード AND A.部品番号 = C.部品番号 </pre>
SQL2	在庫履歴の倉庫コードごと部品番号ごとに、在庫数量の区間数 7 の中央移動平均を算出する。 <pre> SELECT 倉庫コード, 部品番号, 履歴年月日, ⑦ CASE WHEN (COUNT(*) OVER 移動平均区間) = 7 THEN CAST(([e] OVER 移動平均区間) AS INTEGER) ELSE NULL END AS 中央移動平均 FROM 在庫履歴 WINDOW 移動平均区間 AS (PARTITION BY [f] ORDER BY [g] ROWS BETWEEN [h] PRECEDING AND [i] FOLLOWING) </pre>

注記 1 ホスト変数 hv1 には、年月日を設定する。

注記 2 PREVIOUS_DAY(引数)は、引数（文字型）に指定された日付の前日（日付型）を返すユーザー定義関数である。

注記 3 WINDOW 句は、ウィンドウ関数のウィンドウを定義して名前を付ける。OVER 句の後に名前を指定して参照する。

〔トランザクションの排他制御の見直し〕

既存システムの実行ログを分析すると、複数の“在庫引当”APが、“在庫”テーブルの同じ行に対して同時実行されたときに、デッドロックによってAPのリトライが発生していることが判明した。“在庫引当”APの処理の流れとSQL文を図3に示す。ISOLATIONレベルはREPEATABLE READで実行している。

- ① SELECT 在庫数量, 引当済在庫数量 INTO :hv5, :hv6 FROM 在庫
WHERE 倉庫コード = :hv2 AND 部品番号 = :hv3
出庫可能在庫数量 (hv5 - hv6) と hv4 を比較し、出庫が可能な場合だけで以降を実行する。

② UPDATE 在庫 SET 引当済在庫数量 = 引当済在庫数量 + :hv4
WHERE 倉庫コード = :hv2 AND 部品番号 = :hv3

③ UPDATE 出庫 SET 処理状況 = '引当実施' WHERE 出庫番号 = :hv1

④ COMMIT

注記 ホスト変数 hv1, hv2, hv3, hv4 には、それぞれ出庫番号、出庫倉庫コード、部品番号、要求する出庫数量を設定する。ホスト変数 hv5 と hv6 には、それぞれ在庫数量、引当済在庫数量の検索結果を返す。

図3 “在庫引当”APの処理の流れとSQL文

Kさんはデッドロックの原因を調査し、上司であるL氏と対応について議論した。

L氏：デッドロックの原因は何でしょうか。

Kさん：ある“在庫引当”APが j で共有ロックを取得した後、k で専有ロックを取得し直す前に、別の“在庫引当”APが j で共有ロックを取得してしまい、互いに専有ロックを取得しようとしてロックの解放待ちになっていました。

L氏：どのような改善案がありますか。

Kさん：“在庫”テーブルの同じ行に対して、“在庫引当”APが同時実行されないように最終更新TS列で制御する案があります。“在庫引当”APを次のように変更します。①のSELECT文で l するように変更し、①の後にCOMMIT文を追加します。②のUPDATE文のWHERE句に、m ことを条件として追加します。②の後に、②のUPDATE文で該当する行がなかったら、n に戻る制御を追加します。

L氏：その案ではAPの大幅な変更が必要になります。変更を最小限にとどめる方

法はありますか。

K さん：ISOLATION レベルを READ COMMITTED に変更して、 の共有ロックがすぐに解放されるようにする案もあります。

L 氏：単純に READ COMMITTED に変更すると、引当済在庫数量が不正になってしまうおそれがあります。同じ倉庫の同じ部品に対して、“在庫引当” AP が複数同時に実行されるとき、それぞれの要求する出庫数量の合計が、 していると、引当済在庫数量が不正になってしまいます。ISOLATION レベルを変更しない方法はありませんか。

K さん：①の SELECT 文に、 する案があります。

〔索引の見直し〕

毎日の業務終了時に実行される“在庫反映” AP の処理時間の短縮要望を受け、K さんが対策を検討した。その結果、出庫倉庫コードと部品番号の組合せの値の範囲で分割し、分割した値の範囲それぞれを各プロセスに配分して並列実行するように変更することにした。変更後の“在庫反映” AP の処理の流れを図 4 に示す。ISOLATION レベルは REPEATABLE READ で実行する。

1. “出庫”テーブルに登録されている、指定された範囲内の、出庫倉庫コードと部品番号の組合せの値の昇順に、行ごとに次の 2 の処理を行う。
2. “出庫”テーブルを参照して、処理可否を判定する。処理状況が‘出庫済’の場合だけ次の 2-1～2-3 を実行する。
 - 2-1. “在庫”テーブルの在庫数量を更新（出庫数量分を減算）する。
 - 2-2. “在庫”テーブルの引当済在庫数量を更新（出庫数量分を減算）する。
 - 2-3. “出庫”テーブルの処理状況を‘在庫反映済’に更新する。
3. “出庫”テーブルの指定された範囲内の全ての行の処理が完了したら、COMMIT する。

図 4 変更後の“在庫反映” AP の処理の流れ

これを確認した L 氏は、索引の見直しを指示した。

L 氏：“”テーブルの列と列の組合せをこの順で索引キーとする複数列索引を定義しておく必要があります。この索引を定義していないと、自プロセスの対象行かどうかを判定するための参照が

表探索となります。その場合、他 AP が [t] した行を参照しようとして [u] が生じた結果、処理時間が長くなるおそれがあります。デッドロックが発生しないかの検証も必要です。

K さん：処理要否の判定も、表の行の読み込みを不要にして性能を向上させるために、“ [q] ” テーブルの [v] 列もこの複数列索引の索引キーに加えてもよいでしょうか。

L 氏：そのメリットと、そうすることで “ [q] ” テーブルを更新する AP が、①索引の更新頻度の増加によって性能が低下するデメリットとのトレードオフを検証してください。

設問 1 「在庫履歴分析手段の見直し」について答えよ。

- (1) 表 1 中の [a] ～ [i] に入れる適切な字句又は数値を答えよ。
- (2) 表 1 中の SQL2 の下線㉗について、CASE 式の ELSE 句で中央移動平均を NULL にする理由を 25 字以内で答えよ。

設問 2 「トランザクションの排他制御の見直し」について答えよ。

- (1) 本文中の [j] , [k] に入れる適切な数字を、図 3 中の①～④の中から選んで答えよ。
- (2) 本文中の [l] , [m] に入れる適切な字句を答えよ。また、[n] に入れる適切な数字を、図 3 中の①～④の中から選んで答えよ。
- (3) 本文中の [o] に入れる引当済在庫数量の不正を引き起こす条件を答えよ。
- (4) 本文中の [p] に入れる適切な変更内容を答えよ。

設問 3 「索引の見直し」について答えよ。

- (1) 本文中の [q] ～ [v] に入れる適切な字句を答えよ。
- (2) 本文中の下線①について、索引キーに [v] 列を加えることで索引の更新頻度が増加するのはなぜか。 [v] 列の特徴に着目して、30 字以内で具体的に答えよ。