

問2 トランザクションの排他制御に関する次の記述を読んで、設問1～3に答えよ。

Y社は、オフィスじゅう器メーカーである。現在、在庫管理システムのアプリケーションプログラム（以下、APという）の改修を実施している。

〔在庫管理システムのテーブル〕

在庫管理システムの主なテーブル構造を、図1に示す。各テーブルには主索引が定義されている。

倉庫（ <u>倉庫コード</u> ，倉庫名）
組立工場（ <u>工場コード</u> ，工場名，所在地， <u>隣接倉庫コード</u> ）
定期便（ <u>便番号</u> ， <u>発送年月日</u> ， <u>発送元倉庫コード</u> ， <u>発送先倉庫コード</u> ）
部品（ <u>部品番号</u> ，部品名，部品単価）
在庫（ <u>倉庫コード</u> ， <u>部品番号</u> ， <u>倉庫内在庫数量</u> ， <u>出庫対象在庫数量</u> ）
出庫（ <u>出庫番号</u> ， <u>出庫年月日</u> ， <u>出庫元倉庫コード</u> ， <u>出庫先倉庫コード</u> ， <u>出庫先工場コード</u> ， <u>部品番号</u> ， <u>出庫数量</u> ， <u>出庫便番号</u> ， <u>処理状況</u> ）

図1 主なテーブル構造

〔在庫管理業務の概要〕

- (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それぞれ一つ配置されている。
- (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、倉庫コードで一意に識別され、組立工場は、工場コードで一意に識別される。生産拠点を識別するコードは存在しない。
- (4) 定期便は、倉庫間で部品を配送する便であり、便番号で一意に識別される。
- (5) 部品は、部品番号で一意に識別される。
- (6) 部品の在庫は、倉庫と部品の組合せで、その数量をもつ。倉庫内に存在する在庫を、倉庫内在庫と呼ぶ。このうち、隣接する組立工場又は他の生産拠点の倉庫に向けて出庫対象となったものを、出庫対象在庫と呼ぶ。
- (7) 出庫要求とは、倉庫に対して部品の出庫を要求することである。“出庫”テーブルに出庫要求の内容が登録され、処理状況に‘要求発生’が記録される。出庫番号

- は、出庫要求の発生順の一意的な連番である。組立工場が出庫要求する場合、出庫先倉庫コード及び出庫便番号の値は NULL となり、出庫先工場コードが記録される。他の生産拠点の倉庫が出庫要求する場合、出庫先工場コードは NULL となり、出庫先倉庫コードが記録され、出庫便番号には該当する定期便の便番号が記録される。
- (8) 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ことである。在庫引当 AP は、毎日の業務中に定期的に行われ、その時点で登録されている出庫要求を処理する。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、出庫可能であれば出庫対象在庫数量を更新する。在庫引当が完了したら、処理状況は‘引当実施’に更新される。
- (9) 出庫とは、出庫要求に従って、倉庫から部品を出すことである。出庫は多頻度で行われるので、出庫ごとに在庫は更新されず、出庫確定 AP でまとめて更新される。
- (10) 出庫確定 AP は、毎日の業務終了時に実行される。“出庫”テーブルの処理状況が‘引当実施’のものを対象に、倉庫内在庫数量及び出庫対象在庫数量が更新され、処理が完了したら、処理状況は‘出庫実施’に更新される。
- (11) 入庫とは、他の生産拠点の倉庫で出庫された部品を倉庫に入れることである。入庫による在庫の更新は、在庫引当 AP 及び出庫確定 AP と同時に実行されることはない。

[RDBMS の排他制御]

- (1) 在庫管理システムの RDBMS で選択できるトランザクションの ISOLATION レベルとその排他制御は、表 1 のとおりである。
- ロックは行単位で掛ける。共有ロックが掛かっている間、他のトランザクションからの対象行の参照は可能であり、更新は共有ロックの解放待ちとなる。専有ロックが掛かっている間、他のトランザクションからの対象行の参照、更新は専有ロックの解放待ちとなる。

表1 トランザクションの ISOLATION レベルとその排他制御

ISOLATION レベル	排他制御
READ COMMITTED	データ参照時に共有ロックを掛け、参照終了時に解放する。 データ更新時に専有ロックを掛け、トランザクション終了時に解放する。
REPEATABLE READ	データ参照時に共有ロックを掛け、トランザクション終了時に解放する。 データ更新時に専有ロックを掛け、トランザクション終了時に解放する。

(2) 索引を使わずに、テーブルスキャンで全ての行に順次アクセスする場合、検索条件に合致するか否かにかかわらず全行がロック対象となる。索引スキャンの場合、索引から読み込んだ行だけがロック対象となる。

[分析機能の追加]

適切な生産計画を立てるために、部品ごとに在庫数量、出庫数量の日別の推移状況を見たいという要望があり、そのための集計 AP を追加した。集計 AP で実行する SQL の一部を、表 2 に示す。SQL1 は各部品の出庫年月日ごとの出庫数量を集計する。また、SQL1 では、出庫が全くない部品も集計対象とする。SQL2 は、各部品の倉庫間の出庫について、出庫年月日、出庫元倉庫、出庫先倉庫ごとに在庫数量を集計する。

表2 集計 AP で実行する SQL の構文 (未完成)

SQL	SQL の構文
SQL1	SELECT B. 部品番号, S. 出庫年月日, <input type="text" value="a"/> FROM 在庫 AS B <input type="text" value="b"/> 出庫 AS S ON B. 倉庫コード = S. 出庫元倉庫コード AND B. 部品番号 = S. 部品番号 AND S. 出庫年月日 >= '2016-04-01' AND S. 出庫年月日 < '2017-04-01' <input type="text" value="c"/>
SQL2	SELECT 部品番号, 出庫年月日, 出庫元倉庫コード, 出庫先倉庫コード, SUM(出庫数量) FROM 出庫 WHERE <input type="text" value="d"/> IS <input type="text" value="e"/> AND 出庫年月日 >= '2016-04-01' AND 出庫年月日 < '2017-04-01' GROUP BY 部品番号, 出庫年月日, 出庫元倉庫コード, 出庫先倉庫コード

[在庫引当 AP の改修]

在庫管理システムでは、トランザクションの ISOLATION レベルを REPEATABLE READ として設計、運用していた。システムの改修に当たり、在庫引当 AP のトラン

ザクシヨンの ISOLATION レベルを READ COMMITTED に変更することにした。
ISOLATION レベルの変更で問題が発生しないように在庫引当 AP を改修した。

改修前の在庫引当 AP は図 2, 改修後の在庫引当 AP は図 3 のとおりである。これらの AP の実行に先立って, “出庫” テーブルの処理状況が ‘要求発生’ の行を抽出し, 出庫先倉庫ごとに分割したファイルを作成する。それぞれのファイルのレコードは, 出庫番号順に記録されている。作成したファイルを入力として, 在庫引当 AP を並列実行している。在庫引当 AP は, 入力ファイルのレコードごとに繰り返し実行される。

なお, 図 2, 3 中のホスト変数 hv0 は出庫番号, hv1 は出庫元倉庫コード, hv2 は部品番号, hv3 は出庫数量を表す。hv4 と hv5 は, 検索結果を返す出力ホスト変数を表す。

入力ファイルから hv0, hv1, hv2, hv3 の値を設定する。

```
① SELECT 倉庫内在庫数量, 出庫対象在庫数量 INTO :hv4, :hv5 FROM 在庫  
   WHERE 倉庫コード = :hv1 AND 部品番号 = :hv2
```

hv4 - hv5 と hv3 を比較し, 出庫が可能な場合だけ以降を実行する。

```
② UPDATE 在庫 SET 出庫対象在庫数量 = 出庫対象在庫数量 + :hv3  
   WHERE 倉庫コード = :hv1 AND 部品番号 = :hv2
```

```
③ UPDATE 出庫 SET 処理状況 = '引当実施' WHERE 出庫番号 = :hv0
```

```
④ COMMIT
```

図 2 REPEATABLE READ で実行していた改修前の在庫引当 AP

入力ファイルから hv0, hv1, hv2, hv3 の値を設定する。

```
DECLARE 在庫カーソル CURSOR
```

```
  FOR SELECT 倉庫内在庫数量, 出庫対象在庫数量
```

```
  FROM 在庫 WHERE 倉庫コード = :hv1 AND 部品番号 = :hv2  FOR UPDATE
```

```
OPEN 在庫カーソル
```

```
FETCH 在庫カーソル INTO :hv4, :hv5
```

hv4 - hv5 と hv3 を比較し, 出庫が可能な場合だけ以降を実行する。

```
UPDATE 在庫 SET 出庫対象在庫数量 = 出庫対象在庫数量 + :hv3
```

```
  WHERE f OF 在庫カーソル
```

```
CLOSE 在庫カーソル
```

```
UPDATE 出庫 SET 処理状況 = '引当実施' WHERE 出庫番号 = :hv0
```

```
COMMIT
```

注記 在庫カーソルに FOR UPDATE を指定した場合, FETCH された行に専有ロックが掛かる。

図 3 READ COMMITTED で実行する改修後の在庫引当 AP (未完成)

〔出庫確定 AP の改修〕

出庫確定 AP の処理に掛かる時間を短縮するために、出庫確定 AP を並列に多重プロセスで実行するように変更することにした。“出庫”テーブルの出庫番号の値の範囲指定で各プロセスに均等に配分して、REPEATABLE READ で並列実行する。出庫確定 AP の概要は、図 4 のとおりである。

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番号の昇順に、各行ごとに次の 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‘引当実施’の場合だけ次の 2-1～2-3 を実行する。
 - 2-1. “在庫”テーブルの倉庫内在庫数量を更新（出庫数量分を減算）する。
 - 2-2. “在庫”テーブルの出庫対象在庫数量を更新（出庫数量分を減算）する。
 - 2-3. “出庫”テーブルの処理状況を‘出庫実施’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 出庫確定 AP の概要

設問 1 〔分析機能の追加〕について、表 2 中の ～ に入れる適切な字句を答えよ。

設問 2 〔在庫引当 AP の改修〕について、(1)～(3)に答えよ。

- (1) 図 2 の改修前の在庫引当 AP が、REPEATABLE READ で複数同時に実行されるとデッドロックが発生するおそれがある。どのような場合にデッドロックが発生するか、AP 間の SQL の実行状況を、図 2 中の丸数字を用いて 60 字以内で述べよ。
- (2) 図 2 の改修前の在庫引当 AP が、READ COMMITTED で同じ倉庫の同じ部品に対して複数同時に実行されると、在庫数量が不正になるおそれがある。在庫数量が不正になる AP の実行状況を図 5 に示す。不正になるのは、AP2 の①～④の各 SQL が、t2, t4, t6, t8 のどの時間帯で実行された場合か、該当する時間帯に①～④を記入せよ。ここで、一つの時間帯に複数の SQL を実行できる。

また、この状況が発生した場合の、在庫数量が不正とは具体的にどのような状態か、30 字以内で述べよ。

	t1	t2	t3	t4	t5	t6	t7	t8
AP1	①		②		③		④	
AP2								

注記 AP1：先行している AP

AP2：問題を引き起こす AP（未記入）

t1～t8：時間帯を示す記号。t1 から t8 の向きに時間が流れる。

①～④：図 2 中の SQL を示す丸数字

図 5 在庫数量が不正になる AP の実行状況

(3) 図 3 中の に入れる適切な字句を答えよ。

設問 3 [出庫確定 AP の改修] について、(1), (2)に答えよ。

(1) 並列に実行するように変更したが、スループットはさほど向上しなかった。

ボトルネックはどこにあるかを説明する次の記述について、 ～
 に入れる適切な字句を答えよ。

複数の出庫を並列に処理することになるが、同じ と
に対する出庫が複数存在するので、“” テーブルの更新で
 が発生する。

(2) (1)のボトルネックを解消するためには出庫確定 AP をどのように変更する
必要があるかを説明する次の記述について、 ～ に入
れる適切な字句を答えよ。

出庫番号ではなく、 と の組合せの値の範囲指定で各
プロセスに配分するように変更する。また、図 4 の 1 で、出庫番号の昇順で
はなく、 と の昇順に処理を行うように変更する。
“” テーブルの“” 列と“” 列に、複数
列索引を定義しておく。

なお、この索引を定義していない場合、自プロセスの対象行かを判定する
ための参照が となるので、他プロセスが した行を参
照しようとしてロックの解放待ちとなり、別のボトルネックが生じる。