問2　データベースアクセスの同時実行制御に関する次の記述を読んで，設問 1～3 に答えよ。

ソフトウェア開発会社である K 社は，イントラネットに会議室予約システムを構築し，運用している。

〔会議室予約システムの概要〕

会議室予約システムは，各社員の自席の PC 及び会議室に設置されているタブレット端末で利用する。

会議室予約システムには社員番号でログインし，会議室番号，予約日，予約開始時刻，予約終了時刻を指定して会議室予約を行う。予約開始時刻，予約終了時刻には 30 分単位の時刻を入力する。また，会議室番号を指定して予約状況を確認したり，人数，日時を指定して空き会議室を検索したりすることができる。複数の社員が，同じ会議室に対して重複する日時を指定して予約した場合は，最も早く実行された予約を予約成功とし，その他は予約失敗とする。

〔会議室予約システムのテーブル〕

会議室予約システムの主要なテーブルのテーブル構造，概要は，図 1，表 1 のとおりである。

社員（社員番号，社員氏名，…）
会議室（会議室番号，収容可能人数，階数，プロジェクタ設置有無，TV 会議設備設置有無，…）
会議室予約（会議室番号，予約日，予約開始時刻，予約終了時刻，社員番号）

図 1　主要なテーブルのテーブル構造（一部省略）

表 1　主要なテーブルの概要

テーブル名	概要
社員	・社員の情報を管理する。社員番号で社員を一意に識別する。
会議室	・会議室の情報を管理する。会議室番号で会議室を一意に識別する。 ・会議室ごとに収容可能な人数，階数，設備の設置有無が設定されている。
会議室予約	・会議室の予約状況を管理する。会議室番号，予約日，予約開始時刻で会議室予約を一意に識別する。 ・予約開始時刻及び予約終了時刻の分の指定は 00 分又は 30 分とする。

〔RDBMS のトランザクション制御〕

会議室予約システムで使用する RDBMS のトランザクションの ISOLATION レベルは READ COMMITTED であり，行単位でロックをかける。データ参照時には共有ロックをかけ，参照終了時に解放する。データ更新時には専有ロックをかけ，トランザクション終了時に解放する。専有ロックがかかっている間，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。

〔会議室予約システムでの検索〕

会議室予約システムで空き会議室の検索結果一覧を表示する際に必要な情報を得るために実行する SQL 文の例を図 2 に示す。

なお，図 2 中のホスト変数の hv1 は予約希望日，hv2 は予約希望開始時刻，hv3 は予約希望終了時刻を表す。

```
SELECT * FROM 会議室 X
 WHERE [   a   ]
(SELECT * FROM 会議室予約 Y
  WHERE X.会議室番号 = Y.会議室番号 AND Y.予約日 = :hv1
  AND Y.予約開始時刻 [   b   ] :hv3 AND Y.予約終了時刻 [   c   ] :hv2)
```

図 2　検索で実行する SQL 文の例

〔会議室予約システムでの予約処理〕

会議室予約システムの予約処理内容は，図 3 のとおりである。

なお，図 3 中のホスト変数の hv1 は指定予約日，hv2 は指定予約開始時刻，hv3 は指定予約終了時刻，hv4 は指定会議室番号，hv5 は予約者の社員番号を表す（以降の図 5，図 7 でも同様とする）。

① 指定された条件に重なる予約が入っているかを SELECT 文で確認する。結果行がある場合，予約失敗として③に進む。結果行がない場合，②に進む。

```
SELECT * FROM 会議室予約 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
 AND 予約開始時刻 [ b ] :hv3 AND 予約終了時刻 [ c ] :hv2
```

② 指定された条件の予約を INSERT 文で登録する。行が挿入できた場合，予約成功としてコミットし，③に進む。行が挿入できなかった場合，予約失敗として③に進む。

```
INSERT INTO 会議室予約(会議室番号，予約日，予約開始時刻，予約終了時刻，社員番号)
 VALUES(:hv4, :hv1, :hv2, :hv3, :hv5)
```

③ 予約の成否を通知する。

図 3　予約処理内容

〔会議室予約システムでのダブルブッキングの検証〕

会議室予約システムにおいてダブルブッキングが発生した。K 社ではその原因を突き止めるために当日の状況を基に，次の例を用いて図 3 の予約処理内容の検証を行った。

（例）

・会議室 123 に対する 2014 年 4 月 1 日の予約を行う。その日の予約は終日入っていなかった。

・A さん，B さん，C さん，D さん，E さんの 5 人が，それぞれ次の時間帯を指定した。

A さん：11 時 00 分～12 時 00 分

B さん：11 時 00 分～13 時 00 分

C さん：11 時 30 分～13 時 00 分

D さん： 9 時 00 分～11 時 00 分

E さん： 8 時 00 分～18 時 00 分

（検証）

5 人の予約処理の実行が重ならない場合と，重なった場合について，それぞれ(1)，(2)で検証した。

(1) A さん，B さん，C さん，D さん，E さんの順番で，予約処理の実行が重ならない場合の結果を検証して，表 2 を作成した。

(2) A さん，B さん，C さん，D さん，E さんのうち，2 人ずつの全ての組合せに対して，先行，後続を入れ替えて，予約処理の実行が重なった場合の結果を検証して，表 3 を作成した。表 3 では，先行の予約処理が図 3①の処理を実

行した後に，後続の予約処理が図3①の処理を実行，その後に先行の予約処理が図3②の処理を実行することを想定している。

なお，後続の予約処理は，先行の予約処理を追い抜くことはないものとする。

表2　予約処理の実行が重ならない場合の結果

	予約成否	失敗検知箇所
Aさん	○	
Bさん	×	①
Cさん	d	e
Dさん	f	g
Eさん	h	i

注記　予約成否：○（予約成功），△（ダブルブッキング発生），×（予約失敗）
失敗検知箇所：予約失敗を検知した箇所を図3の番号で表す。

表3　2人の予約処理の実行が重なった場合の後続の結果

		先行の予約処理				
		Aさん	Bさん	Cさん	Dさん	Eさん
後続の予約処理	Aさん		予約成否 j 失敗検知箇所 k	予約成否 ℓ 失敗検知箇所 m	予約成否 n 失敗検知箇所 o	予約成否 p 失敗検知箇所 q
	Bさん	省略		同上	同上	同上
	Cさん	省略	省略		同上	同上
	Dさん	省略	省略	省略		同上
	Eさん	省略	省略	省略	省略	

注記　予約成否：○（予約成功），△（ダブルブッキング発生），×（予約失敗）
失敗検知箇所：予約失敗を検知した箇所を図3の番号で表す。

(1)，(2)の検証から，ダブルブッキングとなる理由を次のように結論付けた。

同じ会議室に対して，予約処理の実行が重なった時に，次の二つの条件が成立する場合にダブルブッキングが発生する。

- r が重なる。
- s が異なる。

〔会議室予約システムの改良案〕

ダブルブッキングを防ぐために図4，表4の“日別予約管理”テーブルを追加し，予約処理内容を図5のように改良することを検討した。

日別予約管理（会議室番号，予約日，予約処理中フラグ）

図4　追加する“日別予約管理”テーブルのテーブル構造

表4　追加する“日別予約管理”テーブルの概要

テーブル名	概要
日別予約管理	・“会議室予約”テーブルへの登録の可否を判断するために用いる。 ・予約処理中フラグには‘Y’と‘N’のいずれかが設定される。‘Y’は予約処理中を表し，‘N’は予約処理中でないことを表す。 ・予約受付対象の全ての会議室番号，予約日の組合せは，予約処理中フラグの初期値を‘N’として，あらかじめ登録されている。

① 予約処理中フラグをUPDATE文で‘Y’に更新する。更新できたか否かを記憶する。

```
UPDATE 日別予約管理 SET 予約処理中フラグ = 'Y'
 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
 AND 予約処理中フラグ = 'N'
```

② コミットする。

③ 指定された条件に重なる予約が入っているかをSELECT文で確認する。

```
SELECT * FROM 会議室予約
 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
 AND 予約開始時刻 [ b ] :hv3 AND 予約終了時刻 [ c ] :hv2
```

④ ①で予約処理中フラグを更新できており，③で結果行がない場合は，⑤に進む。
①で予約処理中フラグを更新できており，③で結果行がある場合は，予約失敗として⑥に進む。
①で予約処理中フラグを更新できておらず，③で結果行がない場合は，①に戻る。
①で予約処理中フラグを更新できておらず，③で結果行がある場合は，予約失敗として⑧に進む。

⑤ 指定された条件の予約をINSERT文で登録して，予約成功とする。

```
INSERT INTO 会議室予約(会議室番号，予約日，予約開始時刻，予約終了時刻，社員番号)
 VALUES(:hv4, :hv1, :hv2, :hv3, :hv5)
```

⑥ 予約処理中フラグをUPDATE文で‘N’に更新する。

```
UPDATE 日別予約管理 SET 予約処理中フラグ = 'N'
 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
```

⑦ コミットする。

⑧ 予約の成否を通知する。

図5　予約処理内容改良案

〔会議室予約システムの改良結果〕

図 5 の⑤～⑦の実行時に，ディスク容量不足などのエラーが発生して，予約処理のトランザクションが中断してしまうと問題が生じることが分かったので，〔会議室予約システムの改良案〕は不採用とし，“会議室予約”テーブルを図 6 のように変更した。

会議室予約（会議室番号，予約日，予約開始時刻，予約終了時刻，社員番号，予約済フラグ）

図 6　変更した“会議室予約”テーブルのテーブル構造

会議室予約を行う予約単位をコマと呼び，1 コマは 30 分とする。図 6 の“会議室予約”テーブルでは，各コマを 0 時 00 分から 30 分間隔で設定する。予約受付対象の全てのコマはあらかじめ登録しておく。該当するコマが予約済みか否かを予約済フラグで識別し，予約済みであれば‘Y’，予約済みでなければ‘N’とする。例えば，10 時 00 分～11 時 30 分を予約済みとする場合，予約開始時刻が 10 時 00 分，10 時 30 分，11 時 00 分の 3 コマの予約済フラグを‘Y’に更新する。

変更した“会議室予約”テーブルを使用して，予約処理内容を図 7 のように変更した。

なお，図 7 中のホスト変数の cnt はコマ数を表す。また，図 7 中のユーザ定義関数について次に示す。

- PERIODSTART 関数は，コマの終了時刻を与えて，コマの開始時刻を求めるユーザ定義関数とする。例えば，11 時 30 分を指定した場合，11 時 00 分が返却される。
- PERIODCOUNT 関数は，開始時刻と終了時刻を与えて，含まれるコマ数を求めるユーザ定義関数とする。例えば，10 時 00 分と 11 時 30 分を指定した場合，3 が返却される。
- PERIODNEXT 関数は，指定の時刻とコマ数を与えて，指定の時刻からコマ数だけ後にずらしたコマの開始時刻を求めるユーザ定義関数とする。例えば，10 時 00 分とコマ数 2 を指定した場合，11 時 00 分が返却される。

① 指定された条件に重なる予約が入っていないことを SELECT 文で確認する。結果行がない場合，予約失敗として③に進む。結果行がある場合，②に進む。

```
SELECT 会議室番号 FROM 会議室予約
 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
 AND 予約開始時刻 BETWEEN :hv2 AND PERIODSTART(:hv3) AND 予約済フラグ = 'N'
 GROUP BY 会議室番号 HAVING [ t ] = PERIODCOUNT(:hv2, :hv3)
```

② 指定された条件の予約のため UPDATE 文で更新処理を該当コマ数分繰り返す（cnt は 0 から該当コマ数－1 まで）。全て正常に更新できた場合，予約成功としてコミットし，③に進む。繰り返した中で，1 回でも更新行がなかった場合，予約失敗としてロールバックし，③に進む。

```
UPDATE 会議室予約 SET 予約済フラグ = 'Y', 社員番号 = :hv5
 WHERE 会議室番号 = :hv4 AND 予約日 = :hv1
 AND 予約開始時刻 = PERIODNEXT(:hv2, :cnt) AND 予約済フラグ = 'N'
```

③ 予約の成否を通知する。

図 7　予約処理内容の改良結果

設問 1　会議室予約システムについて，(1)～(4)に答えよ。

(1) 図 2 中の SQL 文の [a] ～ [c] に入れる適切な字句を答えよ。

(2) 図 3 中の②は，行が挿入できないことでダブルブッキングとならないように制御している。その制御で行が挿入できない理由を 30 字以内で述べよ。

(3) 表 2 中の [d] ～ [i] 及び表 3 中の [j] ～ [q] に入れる予約成否，失敗検知箇所を答えよ。

なお，予約成否は，表 2 中の注記の記号で答えること。失敗検知箇所は，予約成否が×の場合に予約失敗を検知した箇所を図 3 中の番号で答えること。×でない場合は空欄のままとすること。

(4) ダブルブッキングとなる条件の [r], [s] に入れる適切な字句を答えよ。

設問 2　〔会議室予約システムの改良案〕について，(1)～(3)に答えよ。

(1) 同じ日に，同じ会議室に対して予約が集中する状況を想定すると，図 5 中の②でコミットを行わない場合，スループットが低下する。その原因となる処理を図 5 中の番号で答えよ。また，原因を 25 字以内で述べよ。

(2) 図 5 中の④において，⑧に進む処理は，速やかに予約失敗を検知するために行っている。この処理はどのような状況を想定して行っているか。20 字以

内で述べよ。

(3) 〔会議室予約システムの改良結果〕で述べている〔会議室予約システムの改良案〕で生じる問題について，どのテーブルがどのような状態になるかを40字以内で述べよ。また，それによって引き起こされる問題を30字以内で述べよ。

設問3 〔会議室予約システムの改良結果〕について，(1)，(2)に答えよ。

(1) 図7中の [t] に入れる適切な字句を答えよ。

(2) 図7中の②において，更新行がなくて予約失敗となるのはどのような状況か。40字以内で述べよ。