

問6 宿泊施設の予約を行うシステムに関する次の記述を読んで、設問1～3に答えよ。

U社は、旅館や民宿などの宿泊施設の宿泊予約を行うWebシステム（以下、予約システムという）を開発している。予約システムの主要要件を図1に示す。

- ・利用者が予約システムを最初に利用する際には、氏名、住所、電話番号を入力し、利用者登録を行う。
- ・利用者は空き部屋照会のための条件入力の画面上で、希望する施設に対し、チェックインとチェックアウトの日付、予約したい部屋の種別及び部屋数を指定して空き状況を照会する。
- ・予約は部屋の種別ごとに行う。種別の違う部屋を予約したい場合は、部屋の種別ごとに分けて予約を行う。
- ・空き状況の照会を行った時点で、希望した種別の部屋に、希望した部屋数の空きがなかった場合は、部屋が空いていない旨を画面に表示する。
- ・空き状況の照会を行った時点で希望した部屋数の空きがあった場合は、予約手続の画面に遷移する。利用者は、宿泊人数を入力し、部屋の予約を確定する。
- ・部屋の予約を確定するまでの間に他の利用者が予約を入れてしまい、必要な部屋数を確保できなくなってしまう場合には、その旨を画面に表示して予約の処理を中断する。

図1 予約システムの主要要件

[データベースの設計]

予約システムを開発するに当たり、データベースの設計を行った。データベースのE-R図を図2に示す。

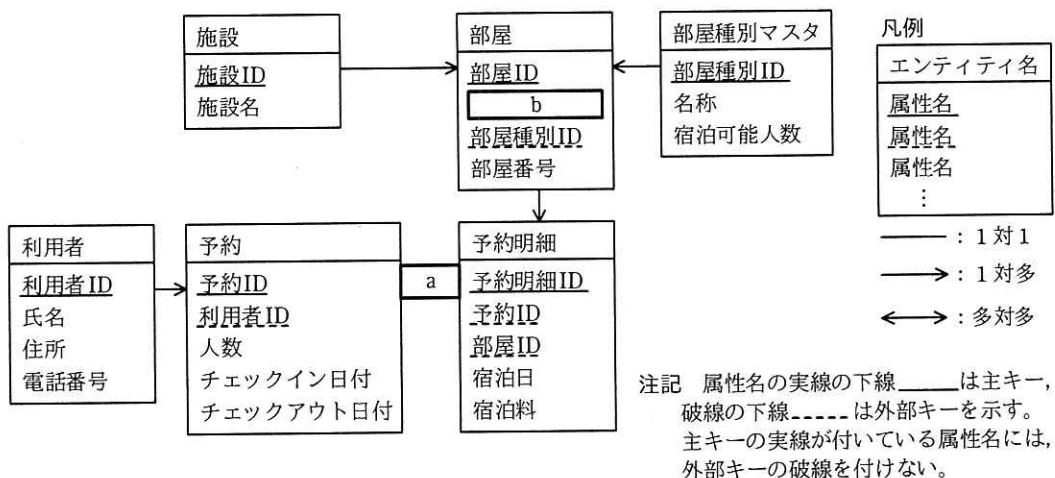


図2 データベースのE-R図（一部）

このデータベースでは、E-R 図のエンティティ名を表名にし、属性名を列名にして、適切なデータ型で表定義した関係データベースによって、データを管理する。部屋 ID は、全施設を通して一意な値である。また、予約 ID、予約明細 ID は、レコードを挿入した順に値が大きくなる。

[部屋の予約の流れ]

部屋の予約は、部屋の空き状況の確認と、予約確定の二つの処理から成る。部屋を予約する際には、希望した施設、部屋の種別、チェックイン日付、チェックアウト日付、部屋数について、空き状況の照会を行う。照会の結果、部屋に空きがあった場合は、予約手続の画面を表示する。部屋に空きがなかった場合は、部屋が空いていない旨を画面に表示し、空き部屋照会のための条件入力の画面に戻って条件を変更するよう促す。

部屋の空き状況の確認を行うための SQL 文を図 3 に示す。予約する部屋の施設 ID、部屋種別 ID、チェックイン日付、チェックアウト日付及び部屋数は、埋込み変数“:施設 ID”、“:部屋種別 ID”、“:チェックイン日付”、“:チェックアウト日付”及び“:部屋数”に設定されている。

```
SELECT 施設 ID, 部屋種別 ID, COUNT(*) FROM 部屋
WHERE c (
  SELECT * FROM 予約明細 WHERE 予約明細.部屋 ID = 部屋.部屋 ID
  AND 予約明細.宿泊日 >= :チェックイン日付 AND 予約明細.宿泊日 < :チェックアウト日付
)
AND 施設 ID = :施設 ID AND 部屋種別 ID = :部屋種別 ID
GROUP BY 施設 ID, 部屋種別 ID
d >= :部屋数
```

図 3 部屋の空き状況の確認を行うための SQL 文

[部屋の空き状況の確認の処理]

予約システムは、図 3 の SQL 文の検索結果として、レコードが返された場合に予約可能であると判定し、予約手続の画面を表示する。レコードが返されなかった場合は、部屋が空いていない旨を画面に表示する。空き状況確認の処理の流れを図 4 に示す。

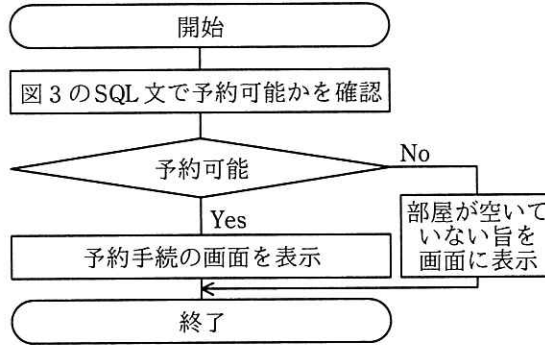


図4 空き状況確認の処理の流れ

[予約確定の処理]

予約手続の画面が表示された後、利用者は予約の確定の操作を行うことで部屋の予約を確定させる。予約の確定の処理では、予約のレコードを挿入した後、各宿泊日について、予約明細に必要な部屋数分のレコードを挿入する。

予約手続の画面が表示されてから、利用者が予約の確定の操作を行うまでの間に、他の利用者が先に予約を確定してしまうこともある。そこで、予約確定の処理では、レコードの挿入の前に図3のSQL文を再度実行し、まだ予約可能な状態であるかを確認してから挿入を行う。予約確定の処理の流れを図5に示す。

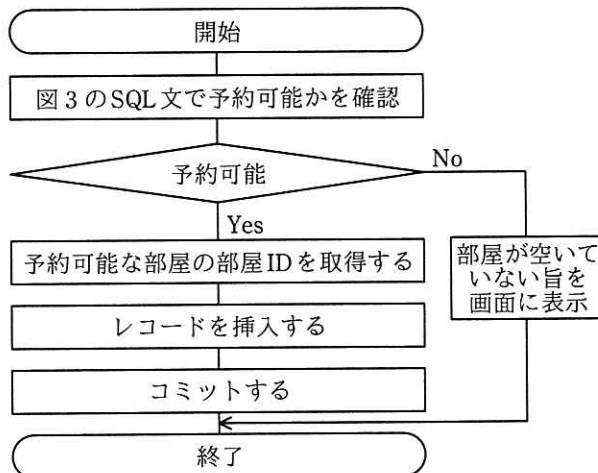


図5 予約確定の処理の流れ

[不具合の報告と対応]

予約システムのテスト中に、同じ宿泊日の同じ部屋について、予約明細のレコードが重複して挿入されてしまう不具合が報告された。報告された事象について確認すると、別々の利用者が同じ時刻に予約確定の操作を行った際に発生していた。

そこで、今後同じ宿泊日の同じ部屋の予約が重複して入らないようにするために、予約明細テーブルの 列と 列の複合キーに対して制約を追加することにした。このような制約のことを、 という。

を追加するためには、既に重複して挿入されてしまったレコードを削除する必要がある。削除に当たっては、同じ宿泊日の同じ部屋の予約が重複した予約明細のレコードについて、最初に挿入された予約のレコードと、それに紐づく予約明細のレコードを残し、それ以外の予約明細、予約のレコードを削除することにした。

予約明細について、削除するレコードを抽出する SQL 文を図 6 に示す。図 6 で得られた該当の予約明細のレコードを削除するとともに、それらに紐づく予約のレコードを削除してから、テストの作業を再開することにした。

予約明細テーブルへの制約の追加後、当該の不具合について再度テストを行ったところ、追加した制約によって、重複が発生しなくなったことが確認できた。

```
SELECT t1.予約 ID, t1.予約明細 ID, t1.部屋 ID, t1.宿泊日 FROM 予約明細 t1
WHERE t1.予約 ID > ( SELECT  FROM 予約明細 t2
                     WHERE  AND  )
```

図 6 削除するレコードを抽出する SQL 文

設問 1 図 2 中の , に入れる適切なエンティティ間の関連及び属性名を答え、E-R 図を完成させよ。

なお、エンティティ間の関連及び属性名の表記は、図 2 の凡例及び注記に倣うこと。

設問 2 図 3 中の , に入れる適切な字句を答えよ。

設問 3 [不具合の報告と対応] について、(1)~(3)に答えよ。

- (1) 本文中の , に入れる適切な列名を答えよ。
- (2) 本文中の に入れる適切な字句を答えよ。
- (3) 図 6 中の ~ に入れる適切な字句を答えよ。