DB

平成26年度　春期
データベーススペシャリスト試験
午後Ⅰ　問題

試験時間	12:30 ～ 14:00（1時間30分）

注意事項

1. 試験開始及び終了は，監督員の時計が基準です。監督員の指示に従ってください。
2. 試験開始の合図があるまで，問題冊子を開いて中を見てはいけません。
3. **答案用紙への受験番号などの記入は，試験開始の合図があってから始めてください。**
4. 問題は，次の表に従って解答してください。

問題番号	問1 ～ 問3
選択方法	2問選択

5. 答案用紙の記入に当たっては，次の指示に従ってください。
   (1) B又はHBの黒鉛筆又はシャープペンシルを使用してください。
   (2) **受験番号欄**に**受験番号**を，**生年月日欄**に**受験票の生年月日**を記入してください。正しく記入されていない場合は，採点されないことがあります。生年月日欄については，受験票の生年月日を訂正した場合でも，訂正前の生年月日を記入してください。
   (3) **選択した問題**については，次の例に従って，**選択欄**の**問題番号**を○印で囲んでください。○印がない場合は，採点されません。3問とも○印で囲んだ場合は，はじめの2問について採点します。
   (4) 解答は，問題番号ごとに指定された枠内に記入してください。
   (5) 解答は，丁寧な字ではっきりと書いてください。読みにくい場合は，減点の対象になります。

〔問1，問3を選択した場合の例〕

注意事項は問題冊子の裏表紙に続きます。
こちら側から裏返して，必ず読んでください。

問題文中で共通に使用される表記ルール

概念データモデル，関係スキーマ，関係データベースのテーブル（表）構造の表記ルールを次に示す。各問題文中に注記がない限り，この表記ルールが適用されているものとする。

1. 概念データモデルの表記ルール

(1) エンティティタイプとリレーションシップの表記ルールを，図1に示す。

① エンティティタイプは，長方形で表し，長方形の中にエンティティタイプ名を記入する。

② リレーションシップは，エンティティタイプ間に引かれた線で表す。

“1対1”のリレーションシップを表す線は，矢を付けない。

“1対多”のリレーションシップを表す線は，“多”側の端に矢を付ける。

“多対多”のリレーションシップを表す線は，両端に矢を付ける。

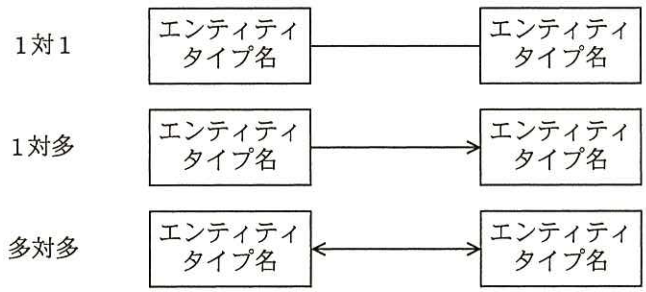


図1　エンティティタイプとリレーションシップの表記ルール

(2) リレーションシップを表す線で結ばれたエンティティタイプ間において，対応関係にゼロを含むか否かを区別して表現する場合の表記ルールを，図2に示す。

① 一方のエンティティタイプのインスタンスから見て，他方のエンティティタイプに対応するインスタンスが存在しないことがある場合は，リレーションシップを表す線の対応先側に“○”を付ける。

② 一方のエンティティタイプのインスタンスから見て，他方のエンティティタイプに対応するインスタンスが必ず存在する場合は，リレーションシップを表す線の対応先側に“●”を付ける。

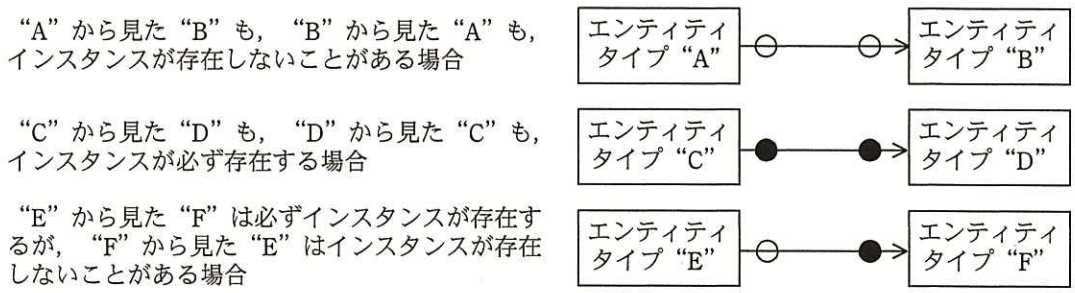


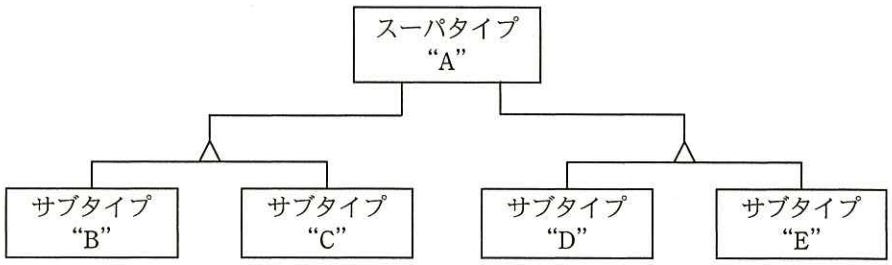
図2　対応関係にゼロを含むか否かを区別して表現する場合の表記ルール

(3)　スーパタイプとサブタイプの間のリレーションシップの表記ルールを，図3に示す。

①　サブタイプの切り口の単位に“△”を記入し，スーパタイプから“△”に1本の線を引く。

②　一つのスーパタイプにサブタイプの切り口が複数ある場合は，切り口の単位ごとに“△”を記入し，スーパタイプからそれぞれの“△”に別の線を引く。

③　切り口を表す“△”から，その切り口で分類されるサブタイプのそれぞれに線を引く。



スーパタイプ“A”に二つの切り口があり，それぞれの切り口にサブタイプ“B”と“C”及び“D”と“E”がある例

図3　スーパタイプとサブタイプの間のリレーションシップの表記ルール

(4)　エンティティタイプの属性の表記ルールを，図4に示す。

①　エンティティタイプの長方形内を上下2段に分割し，上段にエンティティタイプ名，下段に属性名の並びを記入する。[1)]

②　主キーを表す場合は，主キーを構成する属性名又は属性名の組に実線の下線を付ける。

③　外部キーを表す場合は，外部キーを構成する属性名又は属性名の組に破線の下線を付ける。ただし，主キーを構成する属性の組の一部が外部キーを構成する場合は，

破線の下線を付けない。

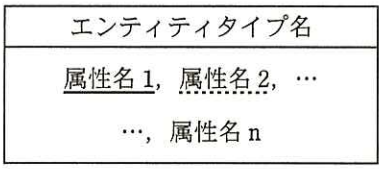


図4　エンティティタイプの属性の表記ルール

2. 関係スキーマの表記ルール及び関係データベースのテーブル（表）構造の表記ルール

(1) 関係スキーマの表記ルールを，図5に示す。

関係名（属性名1，属性名2，属性名3，…，属性名n）

図5　関係スキーマの表記ルール

① 関係を，関係名とその右側の括弧でくくった属性名の並びで表す。[1] これを関係スキーマと呼ぶ。

② 主キーを表す場合は，主キーを構成する属性名又は属性名の組に実線の下線を付ける。

③ 外部キーを表す場合は，外部キーを構成する属性名又は属性名の組に破線の下線を付ける。ただし，主キーを構成する属性の組の一部が外部キーを構成する場合は，破線の下線を付けない。

(2) 関係データベースのテーブル（表）構造の表記ルールを，図6に示す。

テーブル名（列名1，列名2，列名3，…，列名n）

図6　関係データベースのテーブル（表）構造の表記ルール

関係データベースのテーブル（表）構造の表記ルールは，(1) の ① ～ ③ で“関係名”を“テーブル名”に，“属性名”を“列名”に置き換えたものである。

注 [1] 属性名と属性名の間は“,”で区切る。

問1　データベースの設計に関する次の記述を読んで，設問1～3に答えよ。

A社は，ソフトウェアパッケージの開発及び販売を主力事業としている会社である。A社ではこれまで，ソフトウェアの開発中に発生したバグの管理に表計算ソフトを用いてきたが，大規模なBソフトウェアパッケージ開発プロジェクト（以下，Bプロジェクトという）の立上げを機に，新たにバグ管理システムを構築することになった。バグ管理システムの設計担当には，C君が任命された。

〔Bプロジェクトの概要〕

Bプロジェクトの概要は，次のとおりである。

(1) 組織は，階層構造の複数のチーム編成である。

(2) チームは，チームIDで一意に識別され，チーム名，リーダを任されたメンバ，上位階層のチームが定められている。

(3) メンバは，メンバIDで一意に識別され，所属するチームが定められている。メンバは，主担当として必ず一つのチームに所属するほか，他の一つ又は複数のチームを兼任する場合もある。

(4) 開発モデルは，ウォータフォールモデルを採用している。開発工程は，工程IDで一意に識別され，工程名，工程の順序番号が定められている。

(5) 各開発工程では，設計書，ソースコードなどの様々な成果物が作成される。成果物は，成果物IDで一意に識別される。成果物には，成果物名，成果物の作成工程，作成担当チームが記される。

〔バグ管理の概要〕

Bプロジェクトにおけるバグ管理の概要は，次のとおりである。

(1) ソフトウェアのテストを実施し，期待するテスト結果と実際のテスト結果にかい離があり，何らかの対応が必要と考えられる現象をバグと呼ぶ。

(2) バグ種別とは，バグの原因を分類するための区分であり，バグ種別名及び成果物の修正有無が定められている。

(3) バグが発見されたら，表1のプロセスに従って解決する。

(4) ソフトウェアの品質分析を行うメンバは，登録されたバグの集計及び分析を行

う。品質分析の対象とするバグは，成果物の修正が必要なバグ種別が設定されたバグである。

表１　バグ解決プロセス

プロセス	内容
バグの登録	・バグを発見したメンバは，バグの発見日，発見した工程，発見したメンバ，発見内容を登録する。登録時のバグのステータスは‘未着手’である。
バグの発見内容の確認	・リーダは，ステータスが‘未着手’のバグを対象に，バグの発見内容を確認し，スケジュール影響度及びソフトウェア影響度を決定し，ステータスを‘調査中’に更新する。
バグへの対応	・バグの原因調査，バグの修正，バグの修正内容の確認の各作業を総称して対応と呼び，その対応がどの作業であるかを，対応区分として記録する。 ・対応ごとにメンバを１人割り当てて登録する。 ・各対応を開始したら開始日時を，終了したら終了日時を，それぞれ更新する。
バグの原因調査	・バグの原因調査担当に割り当てられたメンバは，バグの原因調査を行い，調査内容，再現方法を記録する。 ・既知のバグと同一原因のバグの場合は，その既知のバグを記録し，ステータスを‘解決済’に更新する。 ・既知のバグと同一原因のバグでなかった場合は，バグ種別を記録し，成果物の修正が必要なバグ種別の場合，ステータスを‘修正中’に，不要な場合，ステータスを‘解決済’に更新する。
バグの修正	・バグの修正担当に割り当てられたメンバは，バグの原因調査の結果に基づいて，一つ又は複数の成果物の修正を行う。 ・修正後に，修正内容と，修正した成果物を記録し，ステータスを‘確認中’に更新する。
バグの修正内容の確認	・バグの修正内容の確認担当に割り当てられたメンバは，バグが解決したかどうかの確認テストを実施し，確認結果を記録する。 ・確認テストの結果，バグが解決していた場合は，ステータスを‘解決済’に更新する。 ・確認テストの結果，バグが解決していない場合は，ステータスを‘調査中’に更新し，バグの原因調査を新たな対応として実施する。
バグのクローズ	・リーダは，ステータスが‘解決済’のバグを対象に，成果物の修正が必要なバグ種別のバグの対応内容を確認し，バグの原因を作り込んだと考えられる工程，バグを発見すべきと考えられる工程を記録する。 ・ステータスが‘解決済’のバグについて完了日を記録し，ステータスを‘クローズ済’に更新し，一連のバグ解決プロセスを完了する。

〔データモデルの設計〕

C 君は，バグ管理システムの構築に当たり，具体例を用いて，概念データモデル（図 1）及び関係スキーマ（図 2）の設計を行った。

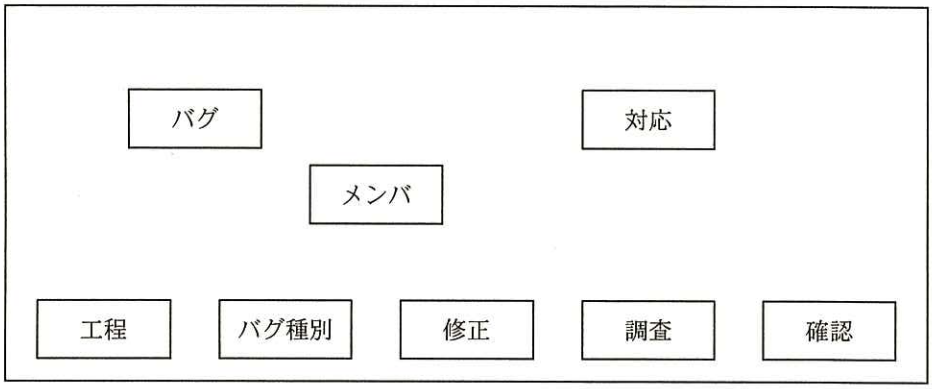


図 1　C 君が設計した概念データモデル（未完成）

工程（工程 ID，工程名，順序番号）
バグ種別（バグ種別 ID，バグ種別名，修正有無）
優先度（緊急度コード，スケジュール影響度，重大度コード，ソフトウェア影響度，
　　優先度コード，リソース投入度）
チーム（チーム ID，チーム名，リーダメンバ ID）
メンバ（メンバ ID，氏名，主担当チーム ID，兼任チーム ID）
バグ（バグ ID，発見日，発見工程 ID，発見メンバ ID，発見内容，緊急度コード，
　　重大度コード，同一原因バグ ID，バグ種別 ID，作り込み工程 ID，
　　発見すべき工程 ID，［ a ］，［ b ］）
対応（バグ ID，対応連番，［ c ］，［ d ］，開始日時，終了日時）
調査（バグ ID，対応連番，調査内容，再現方法）
修正（バグ ID，対応連番，修正内容）
確認（バグ ID，対応連番，確認結果）

図 2　C 君が設計した関係スキーマ（未完成）

図 2 の関係スキーマの主な属性とその意味・制約を表 2 に示す。図 4 は，図 3 の関数従属性の表記法に従って，関係“優先度”の属性間の関数従属性を示したものである。表 3～5 は，関係“バグ”，“工程”，“バグ種別”の具体例である。

表２　属性とその意味・制約（一部省略）

属性名	意味・制約
緊急度コード	スケジュール影響度を一意に識別するコード
スケジュール影響度	バグが開発スケジュールに与える影響の大きさを表す尺度。緊急度コードが異なるが，スケジュール影響度が同一となる場合がある。
重大度コード	ソフトウェア影響度を一意に識別するコード
ソフトウェア影響度	バグがソフトウェアの機能に与える影響の大きさを表す尺度。重大度コードが異なるが，ソフトウェア影響度が同一となる場合がある。
優先度コード	リソース投入度を一意に識別するコード。緊急度コードと重大度コードの組合せによって一意に定まる。
リソース投入度	バグを解決するために，メンバや時間などのリソースをどの程度投入するかを表す尺度。優先度コードが異なるが，リソース投入度が同一となる場合がある。
バグ ID	バグを一意に識別する ID
発見工程 ID	バグが発見された工程を示す工程 ID
作り込み工程 ID	バグの原因を作り込んだと考えられる工程を示す工程 ID
発見すべき工程 ID	本来，バグを発見すべきと考えられる工程を示す工程 ID
同一原因バグ ID	発見したバグの原因が既知のバグの原因と同一であった場合の既知のバグを示すバグ ID
工程 ID	プロジェクトの開発工程を一意に識別する ID
工程名	プロジェクトの開発工程の名称
順序番号	プロジェクトの開発工程の順序を表す番号
バグ種別 ID	バグの種別を一意に識別する ID
バグ種別名	バグの種別の名称
修正有無	成果物の修正の有無を識別するフラグ。成果物の修正が必要なものには‘あり’，不要なものには‘なし’が設定される。
対応連番	バグ ID との組合せで対応を一意に識別する番号
対応区分	対応が，バグの原因調査，バグの修正，バグの修正内容の確認のいずれに該当するかを識別する区分

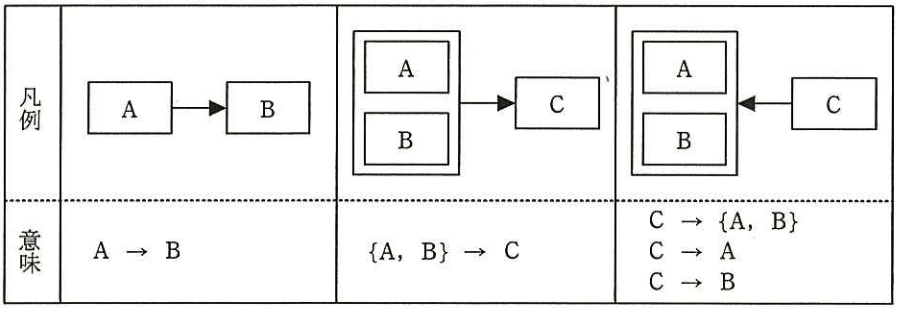


図３　関数従属性の表記法

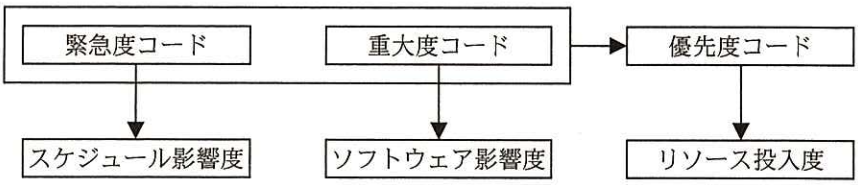


図４　関係“優先度”の属性間の関数従属性

表３　関係“バグ”の具体例（一部属性は省略）

バグ ID	発見日	発見工程 ID	同一原因バグ ID	バグ種別 ID	作り込み工程 ID	発見すべき工程 ID	…
B1	2013-07-19	K5	NULL	S2	K2	K5	…
B2	2013-07-19	K5	B1	NULL	NULL	NULL	…
B3	2013-08-22	K6	NULL	S3	NULL	NULL	…
B4	2013-08-25	K6	NULL	S4	K3	K6	…
B5	2013-09-02	K7	NULL	S1	K1	K2	…

表４　関係“工程”の具体例

工程 ID	工程名	順序番号
K1	基本設計	1
K2	詳細設計	2
K3	プログラム設計	3
K4	実装	4
K5	単体テスト	5
K6	結合テスト	6
K7	総合テスト	7

表５　関係“バグ種別”の具体例

バグ種別 ID	バグ種別名	修正有無
S1	インタフェースの誤り	あり
S2	業務ロジックの誤り	あり
S3	仕様どおり	なし
S4	エラーチェックの誤り	あり
S5	テスト実施手順の誤り	なし
S6	機能の欠如	あり
S7	データの誤り	なし

〔D 部長の指摘事項〕

C 君の上司の D 部長は C 君が設計した内容をレビューし，次の指摘をした。

指摘事項①　図 1 は，リレーションシップが記入されていない。また，図 2 の関係スキーマの一部も未記入である。

指摘事項②　関係“チーム”，“メンバ”には，プロジェクトの組織構造の一部を管理できない不具合がある。

指摘事項③　成果物と，バグの修正を行ったときに修正した成果物の情報を管理す

る関係スキーマが設計されていない。

〔バグの集計及び分析〕

C 君は，バグの集計及び分析を行う際に使用する，関係“バグ”，“工程”，“バグ種別”に対する検索内容と関係代数演算について検討した。表 6 は，関係代数演算の表記法を示したものであり，表 7 は，表 3～5 の具体例を用いて検討した検索内容及び関係代数演算である。

表 6　関係代数演算の表記法

演算	式	備考
射影	R[A1，A2，…]	A1，A2 は，関係 R の属性を表す。同じ内容のタプルは重複が排除される。
選択	R[X　比較演算子　Y]	X，Y は，関係 R の属性を表す。X，Y のいずれか一方は，定数でもよい。
結合	R[RA　比較演算子　SA]S	RA は関係 R の属性，SA は関係 S の属性を表す。

注記　比較演算子は，＜，＝，＞のいずれかである。演算結果を使って入れ子になる演算を行う場合には，演算の過程で生じる中間の関係は，括弧でくくる。例えば，選択演算の結果の関係に対して，結合演算を行う場合は，次のような形式になる。

（R[X　比較演算子　Y]）[RA　比較演算子　SA]S

表 7　検討した検索内容及び関係代数演算

項番	検索内容	関係代数演算式	検索結果
①	2013 年 7 月 19 日に発見されたバグについて，バグ ID の値を求める。	（バグ[発見日 ＝ ‘2013-07-19’]）[バグ ID]	B1，B2
②	バグが発見された工程と，そのバグを本来発見すべきと考えられる工程が同一のバグについて，そのバグ ID の値を求める。	（　e　）[バグ ID]	ア
③	ソフトウェアの品質分析を行う際に分析対象とするバグについて，バグ ID の値を求める。	（（　f　[　g　＝　g　] （　h　[　i　＝　j　] ））[バグ ID]	イ
④	k　で 　ℓ　と考えられるバグについて，バグ ID の値を求める。	（バグ[作り込み工程 ID ＝ 工程 ID] （工程[順序番号 ＜ 順序番号] （工程[工程 ID ＝ ‘K3’]））[バグ ID]	ウ

解答に当たっては，巻頭の表記ルールに従うこと。関係スキーマの解答に当たっては，主キー及び外部キーを明記せよ。

設問1　図2及び図4の関係“優先度”について，(1)，(2)に答えよ。

(1) 関係“優先度”の候補キーを全て答えよ。また，部分関数従属性，推移的関数従属性の有無を，“あり”又は“なし”で答えよ。“あり”の場合は，その関数従属性の具体例を，図3中の意味の欄に示した表記法に従って示せ。

(2) 関係“優先度”は，第1正規形，第2正規形，第3正規形のうち，どこまで正規化されているかを答えよ。また，第3正規形でない場合は，第3正規形に分解した関係スキーマを示せ。

設問2　図1，図2及び〔D部長の指摘事項〕について，(1)～(4)に答えよ。

(1) 指摘事項①について，図1のエンティティタイプ間のリレーションシップを全て記入せよ。同一のエンティティタイプ間に異なる役割をもつ複数のリレーションシップが存在する場合，役割の数のリレーションシップを表す線を記入すること。

なお，図に表示されていないエンティティタイプは考慮しなくてよい。また，エンティティタイプ間の対応関係にゼロを含むか否かの表記は不要である。

(2) 指摘事項①について，図2中の［ a ］～［ d ］に入れる属性名を答えよ。

(3) 指摘事項②の不具合を二つ挙げ，それぞれ25字以内で述べよ。また，不具合を解消した関係スキーマを示せ。

(4) 指摘事項③で設計されていないとしている関係スキーマを設計せよ。

設問3　表3～7及び〔バグの集計及び分析〕について，(1)～(3)に答えよ。

(1) 表7中の項番②，③の検索を行うためには，どのような関係代数演算を行えばよいか。表7中の項番①の例に倣って，［ e ］～［ j ］に入れる適切な字句を答えよ。

なお，関係代数演算の表記法は，表6に従うこと。

(2) 表7中の項番④の関係代数演算式は，どのようなバグを検索するために行

うものか。［ k ］に入れる適切な字句を，工程名を含めて 20 字以内で，［ ℓ ］に入れる適切な字句を 15 字以内で，それぞれ具体的に述べよ。

(3) 表 3～5 の具体例について，表 7 中の項番②～④の検索を行った場合の，［ ア ］～［ ウ ］に入れる検索結果を，表 7 中の項番①の例に倣って答えよ。

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。

問3　テーブルの設計及びSQLの設計に関する次の記述を読んで，設問1～3に答えよ。

健康食品をインターネット販売しているE社は，受注管理システムを開発することになり，Fさんがデータベースの設計を任された。

〔受注管理システムの要求仕様〕

1. 商品

(1) 商品は，単品商品と詰合せセット商品（以下，セット商品という）に区分する。商品には，一意な商品番号を付与する。

(2) セット商品には，一つの化粧箱に複数個の単品商品を詰め合わせたものと，複数種類の単品商品を詰め合わせたものがある。単品商品は，複数種類のセット商品に含まれる。セット商品を構成する単品商品ごとの数量（構成数）は，決まっている。

2. 注文

(1) 顧客は，1回の注文（以下，注文単位という）で，一つ以上の単品商品と一つ以上のセット商品を組み合わせて注文できる。注文単位には，注文全体で一意な注文番号を付与する。

(2) 顧客は，インターネットから商品一覧照会処理を呼び出し，商品番号，商品名，商品説明，写真，販売単価を商品一覧画面に表示させる。表示される順番は，商品全体で重複がないように，商品企画担当者が決めた表示順に基づく。

(3) 顧客は，表示画面から全ての購入希望の商品を検索して商品ごとの注文数を入力した後，注文処理を呼び出す。

(4) 注文処理は，顧客が注文した商品を在庫から引き当て，注文番号，注文日，商品番号，商品名，販売単価，注文数，注文額合計及びお届け予定日の日付（注文日の3日後）を確認画面に表示する。セット商品が不足した場合，そのセット商品に必要な数の単品商品を在庫から引き当てる。確認画面のお届け予定日には，通常のお届け予定日に単品商品を化粧箱に詰め合わせるのに必要な日数を加える。単品商品が不足することはない。

(5) 顧客は，注文内容を確認し，商品の送付先住所，顧客名，連絡先電話番号及び支払に必要な情報を入力し，注文を確定する。

〔テーブルの設計〕

Fさんが設計した関係“商品”及び“在庫”の関係スキーマを，図1に示す。

```
商品（商品番号，商品名，商品説明，写真，販売単価，表示順）
  単品商品（商品番号，社内原価）
  セット商品（商品番号，化粧箱番号，詰合せ日数）
在庫（商品番号，引当可能数）
  単品商品在庫（商品番号，不足セット商品用引当済数）
  セット商品在庫（商品番号，不足セット商品数）
```

図1　関係“商品”及び“在庫”の関係スキーマ

Fさんは，関係“商品”のテーブルの設計に当たり，次の二つの案を考えた。

案1　サブタイプをスーパタイプに統合し，一つの“商品”テーブルとする。

案2　サブタイプ別に“単品商品”テーブル及び“セット商品”テーブルとする。

案1の“商品”テーブルの構造を図2に，案2の“単品商品”テーブル及び“セット商品”テーブルの構造を図3に示す。

```
商品（商品番号，商品名，商品説明，写真，販売単価，表示順，単品区分，
      社内原価，化粧箱番号，詰合せ日数）
```

図2　案1の“商品”テーブルの構造

```
単品商品（商品番号，商品名，商品説明，写真，販売単価，表示順，社内原価）
セット商品（商品番号，商品名，商品説明，写真，販売単価，表示順，
            化粧箱番号，詰合せ日数）
```

図3　案2の“単品商品”テーブル及び“セット商品”テーブルの構造

Fさんは，関係“在庫”については，一つの“在庫”テーブルを設計した。“在庫”テーブルと，その他の主なテーブルの構造を，図4に示す。図4のテーブルは，全て案1，2に共通とする。また，主な列の意味を表1に示す。

セット商品構成（セット商品番号，単品商品番号，構成数）
在庫（商品番号，引当可能数，不足セット商品数，不足セット商品用引当済数）
注文（注文番号，注文日，お届け予定日，住所，顧客名，電話番号，支払情報）
注文明細（注文番号，注文明細番号，商品番号，販売単価，注文数）

図４　両案に共通の主なテーブルの構造

表１　主な列の意味

列名	意味
単品区分	単品商品とセット商品を識別する区分値。区分値は，単品商品では‘Y’，セット商品では‘N’が設定される。
社内原価	単品商品の社内原価。セット商品の販売単価を決める際に必ず使用される。
化粧箱番号	セット商品に使用される化粧箱を一意に識別する番号。セット商品には必ず一つの化粧箱が使われ，同じ化粧箱番号が複数のセット商品で使用される。
詰合せ日数	単品商品をセット商品として化粧箱に詰め合わせるのに要する日数。未定の場合，NULL が設定される。
引当可能数	注文を受け付けたときに引き当て可能な数
不足セット商品数	当該行がセット商品の場合，注文を受け付けたときに引き当てられなかったセット商品の数
不足セット商品用引当済数	当該行が単品商品の場合，注文を受け付けたときに引き当てられなかったセット商品の詰合せのために引き当てた単品商品の数

Ｆさんが案１の“商品”テーブルに定義した制約を表２に，案２の“単品商品”テーブル及び“セット商品”テーブルに定義した制約を表３に示す。

受注管理システムに採用する予定の RDBMS の UNIQUE 制約は，ユニーク索引を用いて実現される。ユニーク索引は，一つのテーブル内でキー列の一意性を保証するものであり，ユニーク索引を複数のテーブルにまたがって作成することはできない。

表2　案1の“商品”テーブルに定義した制約（未完成）

テーブル名	制約の種類	制約を定義した列名
商品	PRIMARY KEY	商品番号
	UNIQUE	a
	NOT NULL	商品名，商品説明，写真，販売単価，表示順，単品区分

表3　案2の“単品商品”テーブル及び“セット商品”テーブルに定義した制約（未完成）

テーブル名	制約の種類	制約を定義した列名
単品商品	PRIMARY KEY	商品番号
	UNIQUE	a
	NOT NULL	商品名，商品説明，写真，販売単価，表示順，社内原価
セット商品	PRIMARY KEY	商品番号
	UNIQUE	a
	NOT NULL	商品名，商品説明，写真，販売単価，表示順，化粧箱番号

表2に示した案1でのNOT NULL制約は不十分なので，Fさんは，図5に示すように案1の“商品”テーブルに検査制約を追加した。検査制約は，次の①～⑥のいずれかの述語を組み合わせて指定する。

① 社内原価 IS NOT NULL

② 社内原価 IS NULL

③ 化粧箱番号 IS NOT NULL

④ 化粧箱番号 IS NULL

⑤ 詰合せ日数 IS NOT NULL

⑥ 詰合せ日数 IS NULL

```
CHECK (   ( 単品区分 = 'Y'  AND   ①   AND  [ b ]  AND  [ c ]  )
        OR ( 単品区分 = 'N'  AND  [ d ]  AND  [ e ]            ) )
```

図5　案1の“商品”テーブルに追加した検査制約（未完成）

〔SQL 文の設計〕

F さんが，案 1 と案 2 のそれぞれについて設計した主な SQL 文を表 4 に示す。

表 4　案 1 と案 2 について設計した主な SQL 文（未完成）

案	SQL	SQL 文
案 1	SQL1	SELECT　M.商品番号，　P.社内原価，　P.化粧箱番号 FROM　注文明細 M，　商品 P WHERE　M.商品番号 = P.商品番号　AND　M.注文番号= :hv
案 2	SQL2	SELECT　M.商品番号，　T.社内原価，　S.化粧箱番号 FROM　注文明細　M [f]　JOIN　単品商品 T　ON　M.商品番号 = T.商品番号 [f]　JOIN　セット商品 S　ON　M.商品番号 = S.商品番号 WHERE　M.注文番号 = :hv
案 2	SQL3	SELECT　M.商品番号，　T.社内原価，　CAST(NULL AS INT) 化粧箱番号 FROM　注文明細 M　[g]　JOIN 単品商品 T ON　M.商品番号 = T.商品番号　WHERE　M.注文番号 = :hv UNION ALL SELECT　M.商品番号，　CAST(NULL AS INT) 社内原価，　S.化粧箱番号 FROM　注文明細 M　[g]　JOIN セット商品 S ON　M.商品番号 = S.商品番号　WHERE　M.注文番号 = :hv
案 1	SQL4	SELECT　K.単品商品番号，　SUM(　[h]　) FROM　注文明細 M，　商品 P，　セット商品構成 K WHERE　M.注文番号 = :hv　AND　M.商品番号 = P.商品番号 AND P.商品番号 = K.セット商品番号 GROUP BY K.単品商品番号
案 2	SQL5	SELECT　K.単品商品番号，　SUM(　[h]　) FROM　注文明細 M，　セット商品 S，　セット商品構成 K WHERE　M.注文番号 = :hv　AND　M.商品番号 = S.商品番号 AND S.商品番号 = K.セット商品番号 GROUP BY K.単品商品番号

注記　hv は，ホスト変数を表す。

〔注文トランザクションの設計〕

F さんは，注文トランザクションについて，次のように設計した。

(1)　注文単位を一つのトランザクションで処理し，最後に COMMIT 文を発行する。

(2)　注文に基づいて，“注文”テーブル及び“注文明細”テーブルに行を挿入する。

(3)　商品については，商品一覧画面に表示された順番に“在庫”テーブルの引当可能数を調べ，引当可能ならば注文数を減算した値で引当可能数を更新する。

(4) セット商品が在庫不足のとき，“在庫”テーブルの不足セット商品数に不足数を加算する。“セット商品構成”テーブルから，主キー順に当該セット商品を構成する単品商品の構成数を調べ，必要数を計算する。単品商品については，“在庫”テーブルの引当可能数には必要数を減算した値で，不足セット商品用引当済数には必要数を加算した値で更新する。

(5) トランザクションの ISOLATION レベルは，READ COMMITTED とする。

設問 1 〔テーブルの設計〕について，(1)～(4)に答えよ。

(1) 表 2 中の［ a ］に入れる適切な字句を答えよ。また，UNIQUE 制約を定義する目的を，要求仕様に関する本文中の字句を用いて 30 字以内で述べよ。

(2) 表 3 中のテーブルについては，UNIQUE 制約を定義し，ユニーク索引を作成しただけでは，［ a ］に関する要求仕様を満たせない。その理由を，本文中の字句を用いて 40 字以内で述べよ。

(3) 次の表に示すように外部キーを定義したとすれば，“セット商品構成”テーブルについては案 1 の場合に，“在庫”テーブルについては案 2 の場合に，不都合が起きるおそれがある。項番 2，3 の［ ア ］，［ イ ］に入れる不都合の内容を，項番 1 に倣って，35 字以内で述べよ。

項番	テーブル名	外部キー	不都合
1	セット商品構成	セット商品番号	案 1 の場合，セット商品番号列に単品商品番号を設定できてしまう。
2	セット商品構成	単品商品番号	案 1 の場合，［ ア ］
3	在庫	商品番号	案 2 の場合，［ イ ］

(4) 図 5 中の［ b ］～［ e ］に入れる適切な述語を，①～⑥から一つずつ選んで答えよ。

設問 2 〔SQL 文の設計〕について，(1)～(3)に答えよ。

(1) SQL2 及び SQL3 の実行結果を SQL1 と同じにしたい。［ f ］，［ g ］に入れる適切な字句を，1 語又は 2 語で答えよ。結果行の並び順は異なってよい。

(2) SQL4 及び SQL5 は，指定した注文について，注文されたセット商品を構成する単品商品の合計数を求める SQL 文である。SQL4 及び SQL5 の実行結果が同じになるように，［ h ］に入れる適切な字句を答えよ。

(3) SQL4 及び SQL5 のアクセスパスは，次に示すネストループ結合である。

① “注文明細”テーブルの主索引を用いて指定された注文番号の行を取り出し，商品番号を調べる。

② ①で調べた商品番号ごとに，案 1 では“商品”テーブル，案 2 では“セット商品”テーブルの主索引を用いてアクセスする。

③ “セット商品構成”テーブルの主索引を用いてアクセスする。

これらのアクセスパスでは，SQL4 の方が，“セット商品構成”テーブルの主索引をアクセスする頻度が多かった。そのアクセス頻度を減らすために，SQL4 の WHERE 句に AND で追加すべき述語を一つ答えよ。

設問 3 〔注文トランザクションの設計〕について，(1)，(2)に答えよ。

(1) 次の TR1～TR4 のうち，いずれか二つの組合せのトランザクションを同時に実行したとき，デッドロックが起きるおそれがある。次の表中の［ ウ ］～［ カ ］に，デッドロックが起きない組合せには○を，起きるおそれがある組合せには×を記入せよ。

TR1：単品商品 2 個を注文する。

TR2：単品商品 1 個とセット商品 1 個を注文する。

TR3：セット商品 1 個を注文する。

TR4：セット商品 2 個を注文する。

	TR1	TR2	TR3	TR4
TR1	○	［ ウ ］	［ エ ］	×
TR2		×	×	×
TR3			［ オ ］	［ カ ］
TR4				×

(2) (1)で起きるおそれがあるとしたデッドロックを防ぐためには，一つのトランザクションの中で“在庫”テーブルの行をどの列の順番で更新すればよいか。列名を答えよ。

6. 退室可能時間に途中で退室する場合には，手を挙げて監督員に合図し，答案用紙が回収されてから静かに退室してください。

退室可能時間	13:10 ～ 13:50

7. **問題に関する質問にはお答えできません。**文意どおり解釈してください。
8. 問題冊子の余白などは，適宜利用して構いません。
9. 試験時間中，机上に置けるものは，次のものに限ります。
   なお，会場での貸出しは行っていません。
   受験票，黒鉛筆及びシャープペンシル（B 又は HB），鉛筆削り，消しゴム，定規，時計（アラームなど時計以外の機能は使用不可），ハンカチ，ポケットティッシュ，目薬
   これら以外は机上に置けません。使用もできません。
10. 試験終了後，この問題冊子は持ち帰ることができます。
11. 答案用紙は，いかなる場合でも提出してください。回収時に提出しない場合は，採点されません。
12. 試験時間中にトイレへ行きたくなったり，気分が悪くなったりした場合は，手を挙げて監督員に合図してください。
13. 午後Ⅱの試験開始は 14:30 ですので，14:10 までに着席してください。