

① データベース言語SQL

① データベース言語

データベース言語には表の作成など、データを定義するための機能とデータの取得、追加、変更、削除などデータを操作するための機能がある。データを定義するための言語がデータ定義言語(SQL-DDL)であり、データを操作するための言語がデータ操作言語(SQL-DML)である。

② SQLの機能の分類

- ㊦ データの定義、作成
- ㊧ データの取得、追加、変更、削除
- ㊨ データへのアクセスの制御
- ㊩ データの一貫性と整合性の保証

② ホスト言語方式

① ホスト言語方式とは

ホスト言語方式はCOBOLやCなどの既存の高水準言語のプログラムの中に、データベースを操作するためのデータ操作言語を混在させる方法で、SQL文やNDL文を組み込んで、データベースをアクセスする。モジュール言語方式と埋め込み方式がある。

② モジュール言語方式

SQLやNDLなどのモジュール言語で書かれたデータベース手続きをCALL文で呼び出して、データベースにアクセスする方式である。SQLの場合、いくつかのパラメータ定義とSQL文からなるモジュールをSQL言語を使用して定義する。ホスト言語側の機能として、モジュール呼び出し機能が必要である。アプリケーションとモジュールとの値の引渡は、パラメータと戻り値を利用する。

③ 埋め込み方式

COBOLプログラムなどに、直接SQL文を書く方法でデータベースをアクセスする方式である。埋め込み方式でコーディングしたプログラムは、SQL文をそのまま解釈できないので、親言語プリコンパイラに通して、SQL文を親言語のCALL文に変換し、プリコンパイラを通してできたソースプログラムを親言語のコンパイラに通すと、CALL命令とモジュール

ルがつながる仕組みになっている。この方法を静的SQLという。

④ 動的SQL

ホスト言語の実行時にSQL文を生成する方法で、パフォーマンスでは静的SQLに比して劣るが、柔軟性の高いアプリケーションを開発できる利点がある。プログラムで実行するSQL文や検索条件がプログラム作成時に決まらない場合に使用できる機能である。プログラムの中に変数を定義しておき、その変数に対して実行時にSQL文を文字列として与えると、それに応じてDBMSが実行される形式である。

③ 利用者言語方式

① 利用者言語方式とは

既存のプログラム言語とは独立して、端末装置から直接にコマンドを入力したり、フォームやメニューを作って、データベースとやりとりする方式である。

② 利用者言語方式の特徴

- ㊦ ユーザが直接データベースにアクセスすることが容易である。
- ㊧ 対話方式では、すぐに結果が出てくるので、非定型業務に向いている。
- ㊨ SQL文を解析しながら実行するので、処理効率が悪い。

④ SQL文とその機能

① 定義言語と操作言語

データ定義言語のDDLに対して、データ操作言語としてDMLがある。データ操作の基本はカーソルの操作、データの検索、挿入、更新、削除などがある。

② SQL文の機能と構文

㊦ SELECT文

機能：特定行を取り出す。特定列を取り出す。出力順序を指定する。

構文：SELECT 列名 FROM 表名 WHERE 検索条件

GROUP BY 列名 HAVING 検索条件 ORDER BY 式

㊧ INSERT文

機能：データの変更に使う。一つの表の中に行を追加する。

構文：INSERT INTO 表名 (列名 1、列名 2、…)

SELECT 列名 1、列名 2、… FROM 表名 WHERE 検索条件

㉞ DELETE文

機能：データの変更に使う。一つの表の中の行を削除する。

構文：DELETE FROM 表名 WHERE 検索条件

㉟ DELETE CURRENT(位置づけ削除)

機能：カーソルを用い、その位置づけられた組を削除する。

構文：DELETE FROM 表名

WHERE CURRENT OF カーソル名

㊱ UPDATE文

機能：データの変更に使う。一つの表の中の列を更新する。

構文：UPDATE 表名 SET 列名 1 = 値 1、列名 2 = 値 2、… WHERE 検索条件

㊲ UPDATE CURRENT(位置づけ更新)

機能：カーソルを用い、その位置づけられた組を更新する。

構文：UPDATE 表名 SET 更新式

WHERE CURRENT OF カーソル名

㊳ DECLARE CURSOR文 (埋込方式で使用)

機能：処理対象とする表を設定し、カーソルを割り当てる。

構文：DECLARE カーソル名 CURSOR FOR SELECT 文

㊴ OPEN文 (埋込方式で使用)

機能：カーソル処理を開始し、カーソルを一行目に位置づける。

構文：OPEN カーソル名

㊵ FETCH文 (埋込方式で使用)

機能：カーソル位置にある行の値を変数に取り出し、カーソル位置を次の行に進める。

構文：FETCH カーソル名 INTO 変数リスト

㊶ CLOSE文 (埋込方式で使用)

機能：カーソル処理を終了する。

構文：CLOSE カーソル名

㊷ COMMIT文 (埋込方式で使用)

機能：データベースに対して行った更新処理を確定し、トランザクションを終了する。

構文 : COMMIT WORK

㉞ **ROLLBACK文（埋込方式で使用）**

機能 : データベースに対して行った更新処理を取り消す。

構文 : ROLLBACK WORK

㉟ **SET TRANSACTION文**

機能 : トランザクションを開始する。

構文 : SET TRANSACTION

㊱ **SAVEPOINT文**

機能 : セーブポイントを作成する。

構文 : SAVEPOINT セーブポイント名

㊲ **CREATE DATABASE文**

機能 : データベースを作成する。

構文 : CREATE DATABASE データベース名

㊳ **CREATE TABLE文**

機能 : 表を作成する。

構文 : CREATE TABLE 表名 (列名 1 データ型、列名 2 データ型、…)

㊴ **CREATE VIEW文**

機能 : ビューを作成する。

構文 : CREATE VIEW ビュー名 (列名、列名、…) AS SELECT文

㊵ **ALTER TABLE文**

機能 : 表の構造を変更する。

構文 : 追加の場合 ALTER TABLE 表名 ADD 列名 データ型 …

修正の場合 ALTER TABLE 表名 ALTER(MODIFY) 列名 データ型 …

削除の場合 ALTER TABLE 表名 DROP 列名 …

㊶ **GRANT文**

機能 : ユーザに権限を与える。

構文 : GRANT 権限 ON 表名 TO ユーザ名

⑤ データ定義SQLの機能

① データベースの作成

データベースの設計、データベースの定義、表の定義、表間の関係の定義、索引の定義などが必要である。データベースの設計に基づいて、データベースの定義を行う。関係データベースに関するいろいろな情報を定義し、スキーマ定義、実表定義、ビュー定義、権限定義などを行い、関係データベースシステムに登録する。

② スキーマ定義と実表定義

データベース全体とそれぞれの表について、その枠組みを定義することができる。これらの情報はDBMSの機能により、データ辞書(リポジトリ)に自動的に登録される。

③ データ辞書システム(リポジトリ)

データベース内にどのようにデータが格納されているか、どこに格納されているのかといったデータに関する情報の集まりである。データベースの構築や保守のための生産性向上の支援を行ったり、データの冗長性の認識、排除を行ったり、標準化や文書化の推進の目的に使用される。

⑥ スキーマの定義

① 機能

実表、ビュー表、権限などのスキーマを定義する。スキーマを定義する際には、誰がスキーマを定義するかを明確にし、その人を識別するスキーマ認可識別子を用いる。認可識別子をもつ利用者は、スキーマ内で作成された表やビューに対して処理権限をもつ。認可識別子を持たない人はデータベースにアクセスすることもできないため、認可識別子はデータベースの保護機能ももつことになる。

② 構文

```
CREATE SCHEMA AUTHORIZATION <スキーマ認可識別子> <スキーマ要素>
```

スキーマ認可識別子は作成者や使用者を識別するもので、ユーザIDを用いる。スキーマ識別子はスキーマごとの所有者を識別するための一意性をもった名前である。スキーマ要素には、実表定義、ビュー定義、権限定義が含まれる。

⑦ 実表定義

① 機能

スキーマ中に新しい実表を定義する。実表は記憶媒体上に実在する表である。定義された段階では、表中にデータは存在せず枠組みだけが作成される。

② 構文

```
CREATE TABLE 表名  
    属性定義、属性定義、…  
    PRIMARY KEY 列名  
    FOREIGN KEY 列名 REFERENCES 表名
```

表名は、新しく作成する実表に対して付けた名前であり、同一のスキーマ中のどの表とも異なる一意性をもった名前である。表名の次は、属性定義を記述する。1つの表に対して1つ以上の属性定義がなければならない。属性名、データ型、…、などを指定する。属性名は、表中の属性を識別するための一意性をもった名前である。同じ表の中のどの属性名とも異ならない。表が異なると同じ名前を用いてもよい。データ型は属性が持つデータのタイプや桁数を指定する。DEFAULT 句は既定値を与えるもので、NOT NULLは非ナル値制約を定義する。PRIMARY KEY 列名によって主キーを一意性制約で定義し、FOREIGN KEY 列名で外部キーを指定し、REFERENCE 表名によって、参照する表を参照制約で定義する。

⑧ ビュー定義

① 機能

スキーマ中に新しいビュー表を定義する。ビュー表は記憶媒体上に実在しない仮想の表であり、実表から導出する。ビュー表を用いると、データの機密保護や表の別の見方が提示できる。

② 構文

```
CREATE VIEW ビュー表名  
    属性名、属性名、…、  
    AS SELECT 属性名、属性名、…、  
    FROM 表名  
    WHERE 導出条件
```

ビュー表名は、新たに作成するビューに対してつけた名前である。同一スキーマの中の実表名、ビュー表名は一意性をもった名前である。属性名は、ビュー表に対する属性定義を記述する。属性名を省略すると、SELECT文で指定した属性名とデータ型を用いる。属性名を指定する場合、ビュー表を構成するすべての属性に対して指定する。ビュー表で指定した属性名に対して実表の属性のデータ型が継承される。属性名は、実表の属性名と同一でもよいし、異なった

名前にしてもよい。属性名はビュー表中の属性を識別するための一意性をもった名前とする。FROM句の表名は、実表名、ビュー表名のどちらでも指定できる。

⑨ SQLの整合性制約

① SQLの整合性制約とは

関係データベースにおける整合性制約を定義する。表中の行と列の定義や表と表の間での制約の定義に関係する。SQLの整合性制約には、一意性制約、参照制約、非ナル制約、検査条件、表明などがある。

② 整合性制約の種類

㊦ 一意性制約

指定した列に同じ値を持たないようにする制約である。行を一意に識別するための1つ以上の列からなる主キーを定義する。表の作成時に主キーを定義することによって、レコードを追加するときに主キーの重複がないかを常に検査できる。

① 参照制約

ある表のある列の値が別の表を参照するとき、参照される表に列の値として存在しなければいけないという制約である。関係データベースでは外部キーの定義によって、他の表の主キーとの参照の整合性を保つ。

㊵ 非ナル制約

指定した列の値にナル値を許さないという制約である。表の定義時に列ごとにナル値を許すかどうかを決定する。主キーの列は非ナル制約を指定しなければならない。

㊶ 検査条件

関係データベースでは列の値が特定の条件を満たすかどうかを検査することができる。表の定義時にある列に対して検査条件を定義する。

⑩ 発注書作成のSQL具体例

① 構文

```
CREATE TABLE 発注書 (  
    発注番号 NUMERIC(4) NOT NULL,  
    発注日 DATE NOT NULL,
```

商品コード CHAR(5) NOT NULL,
数量 NUMERIC(5) NOT NULL,
PRIMARY KEY (発注番号),
FOREIGN KEY (商品コード) REFERENCES 商品表)

⑥ 内容

発注書の定義が行われる。発注書には4つの列名が定義されている。発注番号、発注日、商品コード、数量の4項目である。各列名のデータ型が定義される。各列名ともNOT NULLであるから空白の挿入ができない。PRIMARY KEY として発注番号が設定される。FOREIGN KEY として商品コードが設定され、参照する表名は商品表である。

⑪ SQLのデータの型

① 文字型

人名、会社名、商品名、住所など、様々な文字列を格納する。

列を定義するときに、バイト単位で長さを指定することができる。

固定長文字列型：CHAR(長さ)、CHARACTER(長さ)

可変長文字列：VARCHAR(長さ)、CHAR VARYING(長さ)、CHARACTE VARYING(長さ)

固定長各国語文字列：NCHAR(長さ)、NATIONAL CHAR(長さ)、NATIONAL CHARACTE(長さ)、
NATIONA CHARACTE VARYING(長さ)

② ビット型

固定長ビット列：BIT(長さ)

可変長ビット列：BIT VARYING(長さ)

③ 数値型

整数は、数量、番号、年齢など、整数値を格納する。社員番号、顧客番号、注文番号など、様々な識別番号を格納する。小数は、パーセンテージなど、小数部をもつ数値を格納する。列を定義するときに、精度とスケールを指定する。精度には、整数部と小数部を合わせた桁数を指定する。スケールには、小数部の桁数を指定する。

整数：INT、INTEGER

低精度整数：SMALLINT

小数：NUMERIC(精度、スケール)、DECIMAL(精度、スケール)、DEC(精度、スケール)

浮動小数点数：FLOAT(精度)

倍精度浮動小数点数：REAL

高精度浮動小数点数：DOUBLRE PRECISION

④ 日時型

日付 : DATE

時刻 : TIME (精度)

日付と時刻 : TIMESTAMP (精度)

⑤ 時間隔型

時間間隔 : INTERVAL

⑥ NULL

列になんの値も格納されていない状態である。数値型の列に0の値が格納されている場合や文字型の列に空白の値が格納されている場合はNULLではない。どの行に対してもその列がNULLであってはならない場合は、その列にNOT NULLを指定する。

⑫ SQLの関数

① 集計関数

表から取り出したデータを集計する機能を持っている関数で、次の関数がある。

㊦ COUNT関数

クエリーの結果の行数を求める。WHERE句を指定すると、検索条件を満たす 行数を求める。構文は、COUNT (列名)、COUNT (*)となる。

COUNT (DISTINCT, 列名)は重複を除いた行数をカウントする

㊧ SUM関数

指定された列の合計値を計算する。WHERE句を指定すると、検索条件を満たす合計値を求める。構文はSUM (列名)となる。

㊨ AVG関数

指定された列の平均値を計算する。WHERE句を指定すると、検索条件を満たす平均値を求める。構文はAVG (列名)となる。

㊩ MAX関数

指定された列の最大値を求める。WHERE句を指定すると、検索条件を満たす 最大値を求める。構文はMAX (列名)となる。

㊪ MIN関数

指定された列の最小値を求める。WHERE句を指定すると、検索条件を満たす 最小値を求

める。構文はMIN(列名)となる。

⑥ 文字列関数

文字列を操作する関数で、次の関数がある。

㊦ UPPER関数、LOWER関数

UPPER関数は英文字を大文字に変換し、LOWER関数は英文字を小文字に変換する。
構文はUPPER(文字列)、LOWER(文字列)となる。

㊧ LTRIM関数、RTRIM関数

LTRIM関数は左側の空白を取り除き、RTRIM関数は右側の空白を取り除く。
構文はLTRIM(文字列)、RTRIM(文字列)となる。

㊨ SUBSTR関数

文字列の一部を取り出す関数である。構文はSUBSTR(文字列、開始位置、文字数)となる。

⑦ 算術関数

表から取り出したデータに算術的な操作を加えるもので、次のものがある。

㊦ ABS関数

指定された数値の絶対値を求める。構文はABS(数値)となる。

㊧ CEIL関数、FLOOR関数

CEIL関数は指定された数値以上の最小の整数を求める。FLOOR関数は指定された数値以下の最大の整数を求める。構文はCELL(数値)、FLOOR(数値)となる。

㊨ SIGN関数

指定された値の正負の符号を求める。構文はSIGN(数値)となる。

㊩ SQRT関数

指定された数値の平方根を求める。構文はSQRT(数値)となる。

⑧ 日付計算用の関数

日付計算を行うための関数で、次の関数がある。

㊦ YEAR(列名)

年だけを取り出す

- ① MONTH(列名)
月だけを取り出す
- ② DAY(列名)
日だけを取り出す
- ③ CHAR(列名)
日付列、時刻列を文字列として取り出す

⑬ カーソルの処理

① カーソル処理の機能

データベースのように複数の行からなるデータの集合の中から、1行ずつデータを取り出す機能がカーソルの処理である。アプリケーションプログラムでデータベースを利用する場合、まず、カーソルを作成し、そのカーソルを利用することによって、特定のデータを効率よく使用することができる。

② カーソルの利用手順

① カーソルを作成する。

カーソルを作成する構文は次の通りである。
DECLARE カーソル名 CURSOR FOR SELECT文

② カーソルを開く。

カーソルを使用するにはカーソルを開く必要がある。構文は次の通りである。
OPEN カーソル名

③ データベースの最後のレコードに達するまで1レコードずつ取り出す。

データベースの中から1レコードを取り出す構文は次のようになる。
FETCH カーソル名 INTO 変数リスト

④ 作業が終了したら、カーソルを閉じる。

カーソルを閉じた後も、そのカーソルはデータベースの中に存続する。再利用する場合にはもう一度開く必要がある。閉じるための構文は次のようになる。
CLOSE カーソル名

⑤ カーソルの割当を解除し、カーソルを破棄する。

カーソルの割当を解除し、カーソルを破棄すると、カーソルに割り当てられていたメモリ領域が解除される。カーソルを破棄する構文は次のようになる。

DEALLOCATE カーソル名

例題演習

関係データベースの“製品”表と“売上”表から、売上報告のビュー表を定義するSQL文中のaに入るものはどれか。

```
CREATE VIEW 売上報告 (製品番号, 製品名, 納品数, 売上年月日, 売上金額)
AS [ a ] 製品. 製品番号, 製品. 製品名, 売上. 納品数, 売上. 売上年月日,
売上. 納品数 * 製品. 単価
FROM 製品, 売上
WHERE 製品. 製品番号 = 売上. 製品番号
```

表名	列名
製品	製品番号, 製品名, 単価
売上	製品番号, 納品数, 売上年月日

ア GRANT イ INSERT ウ SCHEMA エ SELECT

解答解説

ビューに関する問題である。

ビュー定義の構文は次のようになる。

```
CREATE VIEW ビュー表名 属性名、属性名、…、
AS SELECT 属性名、属性名、…、FROM 表名 WHERE 導出条件
a の内容はSELECTとなり、求める答えはエとなる。
```

例題演習

関係データベースシステムにおいて、カーソル定義を行った後、カーソルの示す行を取り出すのに用いるSQL文はどれか。

ア DECLARE文 イ FETCH文 ウ OPEN文 エ SELECT文

解答解説

関係データベースシステムのカーソル操作に関する問題である。

アのDECLARE はカーソルの宣言を行う。DECLARE カーソル名 CURSOR

イのFETCH はカーソルの位置にある行の値を取り出し、カーソルの位置を次の行に進める。求める答えはイとなる。

ウのOPENは宣言したファイル、カーソルをオープンする。

エのSELECTは、特定の行または特定の列を取り出し表示する。

例題演習

カーソル操作に関する記述のうち、適切なものはどれか。

- ア INSERT文を実行する前に、OPEN文を実行してカーソルを開いておかなければならない。
- イ PREPARE文は、与えられたSQL文を実行し、その結果を自分のプログラム中に記録する。
- ウ カーソル操作によって複数行を一括更新することが可能となる。
- エ プログラム言語は通常レコード単位の処理を行うのに対し、SQLでは集合単位の処理を行うので、その間をカーソルによって橋渡しする。

解答解説

カーソル操作に関する問題である。

アのINSERTはデータの変更に使い、一つの表の中に行を追加する。OPEN文は、埋め込み方式で使用する場合に、カーソル処理を開始し、カーソルを一行目に位置づける。

イのPREPARE文は動的SQLの準備に使用され、指定された問い合わせを構文解析し、書き換え、その後EXECUTE文が発行されると、準備された文の実行だけが行われる。

ウの複数行の一括更新はできない。

エのカーソルは集合単位の処理を行い、レコード間の橋渡しにカーソルを使用するは適切な記述である。求める答えはエとなる。

例題演習

埋込型SQL文を使用して、プログラムから関係データベースの表をアクセスするときの導出表を定義するためのSQL文はどれか。

- ア CLOSE イ DECLARE CURSOR ウ FETCH エ OPEN

解答解説

SQL言語に関する問題である。

データベースのように複数の行からなるデータの集合の中から1行ずつデータを取り出す機能がカーソルの処理である。アプリケーションプログラムでデータベースを利用する場合、まず、カーソルを作成し、そのカーソルを利用手順に従って活用することによって、特定のデータを効率よく取り出すことができる。

アのCLOSEは、カーソル処理を終了する。

イのDECLARE CURSORは、処理対象とする表を設定し、カーソルを割り当てる。表をアクセスするときの導出表を定義することである。求める答えはイとなる。

ウのFETCHは、カーソルの位置にある行の値を外部の変数に取り出し、カーソルの位置を次の行に進める。

エのOPENは、カーソル処理を開始し、カーソルを一行目に位置づける。

例題演習

6行だけから成る“配送”表において成立している関数従属はどれか。ここで、 $X \rightarrow Y$ は、 X は Y を関数的に決定することを表す。

配送

配送日	部署ID	部署名	配送先	部品ID	数量
2016-08-21	300	第二生産部	秋田事業所	1342	300
2016-08-21	300	第二生産部	秋田事業所	1342	300
2016-08-25	400	第一生産部	名古屋工場	2346	300
2016-08-25	400	第一生産部	名古屋工場	2346	1,000
2016-08-30	500	研究開発部	名古屋工場	2346	30
2016-08-30	500	研究開発部	川崎事業所	1342	30

ア 配送先→部品ID

イ 配送日→部品ID

ウ 部署ID→部品ID

エ 部署名→配送先

解答解説

関数従属に関する問題である。

関数従属は、関係データベースにおける表の2つの属性集合間で、一方の属性集合の値がもう一方の属性集合の値を関数的に決定するという制約である。

アの配送先→部品IDは配送先が決まれば部品IDは一意に決まる。求める答えはアとなる。

イの配送日→部品IDは、配送日2016-08-30に対して部品IDは、2346と1342の2通りあり、一意に定まらない。

ウの部署ID→部品IDは、部署ID500に対して部品IDは、2346と1342の2通りあり、一意に定まらない。

エの部署名→配送先は、部署名研究開発に対して、配送先は名古屋工場と川崎事業所の2通りあり、一意に定まらない。

例題演習

“出庫記録”表に対するSQL文のうち、最も大きな値が得られるものはどれか。

出庫記録

商品番号	日付	数量
NP200	2006-10-10	3
FP233	2006-10-10	2
NP200	2006-10-11	1
FP233	2006-10-11	2

ア SELECT AVG(数量) FROM 出庫記録 WHERE 商品番号 = 'NP200'

イ SELECT COUNT(*) FROM 出庫記録

ウ SELECT MAX(数量) FROM 出庫記録

エ SELECT SUM(数量) FROM 出庫記録 WHERE 日付 = '2006-10-11'

解答解説

SQL文に関する問題である。

アは商品番号NP200の数量の平均値で、2となる。

イは在庫記録の件数で、4となる。

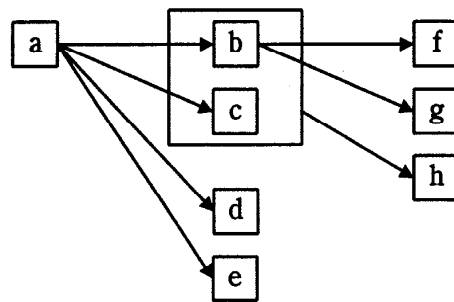
ウは数量の最大値で、3となる。

エは日付が2006-10-11の合計数量で、3となる。

従って、最大値はレコード件数の4であり、求める答えはイとなる。

例題演習

属性 x の値によって属性 y の値が一意に定まることを、 $x \rightarrow y$ で表す。図に示される関係を、第3正規形の表として正しく定義しているものはどれか。ここで、 x の四角内に複数の属性が入っているものは、それら複数の属性すべての値によって、属性 y の値が一意に定まることを示す。



ア 表1 {a}

表2 {b, c, d, e}

表3 {f, g, h}

イ 表1 {a, b, c, d, e}

表2 {a, c}

表3 {b, e, f, g, h}

ウ 表1 {a, b, c, d, e}

表2 {b, c, f, g, h}

表3 {b, c, h}

エ 表1 {a, b, c, d, e}

表2 {b, f, g}

表3 (b, c, h)

解答解説

関係データベースに関する問題である。

属性 a の値によって属性 b 、 c 、 d 、 e が一意に定まる表は、表1 {a, b, c, d, e} で表すことができる。

属性 b の値によって属性 f 、 g が一意に定まる表は、表2 {b, f, g} で表すことができる。

属性 b 、 c の値によって属性 h が一意に定まる表は、表3 {b, c, h} で表すことができる。

図に示された第3正規形の表として正しく定義されているのはエの3つの表である。求める答えはエとなる。