

選択した問題は、選択欄の(選)をマークしてください。マークがない場合は、採点されません。

問3 コンサートチケット販売サイトの関係データベースの設計及び運用に関する次の記述を読んで、設問1～4に答えよ。

D社は、Web上で会員制のコンサートチケット販売サイトを運営している。販売サイトのシステムは販売サブシステムと席予約サブシステムから構成され、販売サブシステムで購入申込み及び決済を処理し、席予約サブシステムで座席指定を処理する。本問では、販売サブシステムだけを取り扱う。

販売サブシステムで利用しているデータベースの表構成とデータの格納例を図1に示す。下線付きの項目は主キーである。

会員表

<u>会員ID</u>	氏名	電子メールアドレス
K00001	情報太郎	taro@example.com
⋮	⋮	⋮

商品表

<u>コンサートID</u>	コンサート情報	開催日時
C00001	クリスマスコンサート 2018 in 東京 出演: Xバンド…	2018-12-24 18:00:00
⋮	⋮	⋮

商品詳細表

<u>コンサートID</u>	<u>席種</u>	価格	発売席数
C00001	S	6000	500
⋮	⋮	⋮	⋮

決済表

<u>販売ID</u>	決済日	決済額
H000001	2018-09-29	12000
⋮	⋮	⋮

販売表

<u>販売ID</u>	会員ID	コンサートID	席種	席数	販売日	販売額	決済期限日
H000001	K00001	C00001	S	2	2018-09-02	12000	2018-10-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

図1 販売サブシステムで利用しているデータベースの表構成とデータの格納例

[コンサートの席の説明]

- (1) コンサートの席種には、S、A 及び B がある。
- (2) 各席種の価格（常に有料）及び発売席数は、コンサートごとに異なる。

[販売サブシステムの説明]

- (1) 販売サブシステムは取り扱うコンサートの席種ごとの販売可能な席数を管理する。
- (2) 会員が購入申込みを行うと、販売サブシステムは一意的な販売 ID を生成して販売表にレコードを追加する。
- (3) 会員が支払手続を行うと、決済処理として販売サブシステムは販売 ID を主キーとするレコードを決済表に追加する。ここで、決済日はレコードを追加した日とする。
- (4) 販売サブシステムは決済期限日の翌日に、決済期限日を過ぎた販売表中のレコードと販売 ID が同じレコードが決済表にない場合、その購入申込みは取り消されたものとして、バッチ処理によって決済表に当該販売 ID を主キーとするレコードを追加する。このレコードの決済日は NULL で、決済額は -1 とする。
- (5) バッチ処理は、毎夜 0~4 時の販売サイトのシステムのメンテナンス時間帯を行う。
- (6) 会員が購入を申し込んだ席数が、その時点で販売可能な席数を上回る場合には、販売サブシステムは“販売終了”と表示し、この購入申込みを受け付けない。

設問 1 データベースのデータの整合性を保つために DDL で制約をつけている。図 1 の表構成において、列名とその列に指定する制約の正しい組合せを、解答群の中から選べ。

解答群

	表名.列名	制約
ア	決済表.決済額	検査制約
イ	決済表.決済日	非 NULL 制約
ウ	商品詳細表.席種	参照制約
エ	販売表.会員 ID	一意性制約

設問2 “販売終了”の表示判定を行うために、販売できない席数を求める必要がある。販売できない席数を出力する SQL 文の  に入れる正しい答えを、解答群の中から選べ。ここで、コンサート ID は C00001、席種は S である。a1 と a2 に入れる答えは、a に関する解答群の中から組合せとして正しいものを選ぶものとする。

```
SELECT SUM(販売表.席数)
FROM 販売表  a1 決済表 ON 販売表.販売 ID = 決済表.販売 ID
WHERE 販売表.コンサート ID = 'C00001'
      AND 販売表.席種 = 'S'
      AND  a2
```

a に関する解答群

	a1	a2
ア	INNER JOIN	決済表.決済額 = -1
イ	INNER JOIN	決済表.決済額 >= 0
ウ	LEFT OUTER JOIN	(決済表.決済額 IS NULL OR 決済表.決済額 = -1)
エ	LEFT OUTER JOIN	(決済表.決済額 IS NULL OR 決済表.決済額 >= 0)
オ	RIGHT OUTER JOIN	(決済表.決済額 IS NULL OR 決済表.決済額 = -1)
カ	RIGHT OUTER JOIN	(決済表.決済額 IS NULL OR 決済表.決済額 >= 0)

設問3 決済期限日まで残り3日となっても支払手続が行われていない購入申込みがある会員に、支払手続を促す電子メールを送る。この会員の氏名、電子メールアドレス及び販売 ID を出力する SQL 文の  に入れる正しい答えを、解答群の中から選べ。NOW は SQL を実行した日の日付を返すユーザ定義関数であり、DATEDIFF はともに日付である二つの引数を受け取って第1引数から第2引数を引いた日数を整数値で返すユーザ定義関数である。

```
SELECT 会員表.氏名, 会員表.電子メールアドレス, 販売表.販売 ID
       b
```

b)に関する解答群

- ア FROM 会員表, 販売表  
WHERE DATEDIFF(販売表. 決済期限日, NOW()) = 3  
AND 販売表. 会員 ID = 会員表. 会員 ID  
AND 販売表. 販売 ID NOT IN (SELECT 販売 ID FROM 決済表)
- イ FROM 会員表, 販売表  
WHERE DATEDIFF(販売表. 決済期限日, NOW()) = 3  
AND 販売表. 会員 ID = 会員表. 会員 ID  
AND 販売表. 販売 ID IN  
(SELECT 販売 ID FROM 決済表 WHERE 決済額 >= 0)
- ウ FROM 会員表, 販売表, 決済表  
WHERE DATEDIFF(販売表. 決済期限日, NOW()) = 3  
AND 販売表. 会員 ID = 会員表. 会員 ID  
AND 販売表. 販売 ID = 決済表. 販売 ID
- エ FROM 会員表, 販売表, 決済表  
WHERE DATEDIFF(販売表. 決済期限日, NOW()) = 3  
AND 販売表. 会員 ID = 会員表. 会員 ID  
AND 販売表. 販売 ID = 決済表. 販売 ID  
AND 決済表. 決済額 <> -1

設問4 会員への優待サービスのために、ポイント制度を導入する。そのために修正した会員表、決済表及び販売表の表構成を図2に示す。ポイント制度を導入するとき追加した列は0で初期化する。

会員表

会員ID	氏名	電子メールアドレス	ポイント残高
------	----	-----------	--------

決済表

販売ID	決済日	決済額	付与ポイント
------	-----	-----	--------

販売表

販売ID	会員ID	コンサートID	席種	席数	販売日	販売額	決済期限日	使用ポイント
------	------	---------	----	----	-----	-----	-------	--------

図2 修正した会員表、決済表及び販売表の表構成

会員は購入申込み時に、1ポイント1円としてポイント残高の範囲で、販売額に充当するポイント数を指定する。販売サブシステムは、指定したポイント

数を使用ポイントに格納し、ポイント残高から減じる。会員は、販売額から使用ポイントを差し引いた金額を決済額として支払う。販売額の全額にポイントを充当した場合は、販売サブシステムは購入申込み時に支払手続が行われたものとし、決済処理として、決済表にレコードを追加する。

ポイント制度の導入時に追加したバッチ処理によって、前日に決済処理された販売 ID ごとに、その決済額が 20,000 円以上、10,000 円以上 20,000 円未満、10,000 円未満の場合に、それぞれ 3%、2%、1%のポイントを付与する。付与したポイント数は、付与ポイントに格納し、ポイント残高に加える。

決済表の付与ポイントを更新する正しい SQL 文を、解答群の中から選べ。NOW、DATEDIFF は設問 3 で使用したユーザ定義関数と同じであり、FLOOR は引数の値以下で最大の整数値を返す関数である。

#### 解答群

- ア INSERT INTO 決済表( 付与ポイント )  
SELECT IF 決済額 >= 20000 THEN FLOOR(決済額 \* 0.03)  
ELSEIF 決済額 >= 10000 THEN FLOOR(決済額 \* 0.02)  
ELSE FLOOR(決済額 \* 0.01) END  
WHERE DATEDIFF(NOW(), 決済日) = 1
- イ UPDATE 決済表 SET 付与ポイント = (  
CASE 決済額 >= 20000 THEN FLOOR(決済額 \* 0.03)  
WHEN 決済額 >= 10000 THEN FLOOR(決済額 \* 0.02)  
ELSE FLOOR(決済額 \* 0.01) END )  
WHERE DATEDIFF(NOW(), 決済日) = 1
- ウ UPDATE 決済表 SET 付与ポイント = (  
CASE WHEN 決済額 >= 20000 THEN FLOOR(決済額 \* 0.03)  
WHEN 決済額 >= 10000 THEN FLOOR(決済額 \* 0.02)  
ELSE FLOOR(決済額 \* 0.01) END )  
WHERE DATEDIFF(NOW(), 決済日) = 1
- エ UPDATE 決済表 SET 付与ポイント = (  
IF 決済額 >= 20000 THEN FLOOR(決済額 \* 0.03)  
ELSEIF 決済額 >= 10000 THEN FLOOR(決済額 \* 0.02)  
ELSE FLOOR(決済額 \* 0.01) END )  
WHERE DATEDIFF(NOW(), 決済日) = 1