

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(データベース) (H30秋FE 午後問3)

【解答】

【設問1】 ア
【設問2】 a-エ
【設問3】 b-ア
【設問4】 ウ

【解説】

コンサートチケット販売サイトのチケット販売を管理する関係データベースに関する問題である。基本情報技術者試験のデータベース問題では、SELECT文を主体としたSQL文の基本的な問題が多く出題されている。本問題でも設問2、3がSELECT文に関する問題となっており、基礎知識があれば得点できる問題である。また、設問4ではCASE式に関する問題が出題されている。CASE式は、標準SQLのSQL-92から提供された機能であり、IF文がないSQLでは便利な機能であるが、SQLの基本的な範囲を若干超えているかもしれない。しかし、書式を覚えていれば解答できる問題であるため、少し幅広く関数やキーワードの書式を覚えておくとよいだろう。

【設問1】

データベースのデータの整合性を保つために、CREATE TABLE文等のDDL(Data Definition Language)文を使用する際に制約を付けることができる。

本問題では、問題文で示されている条件を基に、表名、列名と制約の正しい組合せを選ぶ。制約には次のものがあるので、選択肢を確認しながら考えていく。

- 一意性制約：データの重複を排除する。
- 非NULL制約：データのNULL保存を排除する。
- 検査制約：データの値の範囲等をチェックする。
- 参照制約：テーブル間の参照整合性を保証する。

ア：[コンサートの席の説明] (2)に「各席種の価格(常に有料)」とあり、決済額は必ず0円より多くなる(設問4でポイント制度が導入されると決済額=0円の場合もあるが、ここでは考えないものとする)。また、[販売サブシステムの説明] (4)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」と記述されているので、決済額は-1になることがある。これらのことから、決済額>0円や決済額=-1円となり、0円や-1円より小さい決済額の登録を禁止するためには、決済額に検査制約を設定する必要がある。したがって、「決済表.決済額」と「検査制約」の組合せは正しい。

イ：(ア)と同様に[販売サブシステムの説明] (4)の記述を確認する。「このレコードの決済日はNULLで、決済額は-1とする」と記述されているので、決済表.決済日はNULLになることがある。したがって、「決済表.決済日」と「非NULL制約」の組合せは誤りである。

ウ：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」には席種を管理する表はなく、商品詳細表.席種は他のテーブルのデータを参照することはない。したがって、「商品詳細表.席種」と「参照制約」の組合せは誤りである。

エ：[販売サブシステムの説明] (2)に「会員が購入申込みを行うと、販売サブシステムは一意な販売IDを生成して販売表にレコードを追加する」とあり、販売表のレコードは会員がチケットを購入するたびに作成されることが分かる。会員は、複数のコンサートチケットを購入することができるため、販売表の中では複数レコードで同じ会員IDが保存される。したがって、「販売表.会員ID」と「一意性制約」の組合せは誤りである。

正解は(ア)である。

【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SQL文の空欄に入れる適切な答えを選ぶ。対象はコンサートIDがC00001、席種がSである。

購入申込み時点での販売席数は、販売表の席数を集計すれば求められる。しかし、[販売サブシステムの説明] (4)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」とあり、実際の販売席数を出力するには販売表の席数の集計から、取り消された席数を差し引く必要がある。そのため、設問にあるSQL文では販売表と決済表を結合して席数を集計している。

・空欄a1：まず、空欄a1について考える。SQL文のFROM句内の空欄であり、解答群から結合演算に関するキーワードが入ることが分かる。

[販売サブシステムの説明] (3)に「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」とあり、支払手続が行われるまで決済表にはレコードが追加されないため、販売表のレコードには決済表と結合できないレコードがある。しかし、まだ支払手続が行われていない販売も販売済の席数として集計する必要があるため、決済表と結合できない販売表のレコードも抽出しなければならない。そのようなレコードを抽出するためには、外(外部)結合を使用する。このSQL文では空欄の左側に販売表、右側に決済表があり、販売表(左側)のレコードを全て抽出したいため、左外結合(LEFT OUTER JOIN)を使用する。したがって、空欄a1には「LEFT OUTER JOIN」が入る。

なお、INNER JOIN、LEFT OUTER JOIN、RIGHT OUTER JOINの違いは次のとおりである(OUTERは省略可能)。

<JOIN句>

INNER JOIN：結合条件と一致するレコードだけを抽出
LEFT OUTER JOIN：左側の表のレコードを全て抽出
RIGHT OUTER JOIN：右側の表のレコードを全て抽出

・空欄a2：次に、空欄a2について考える。WHERE句内の抽出条件であり、解答群から決済表の決済額に関する条件であることが分かる。

このSQL文は、既に販売済の席数を集計する。販売済と扱うのは、販売済か

つ支払手続が行われた(決済表の決済額が0円以上)レコードと、販売済かつ販売期限前でまだ支払手続が行われていない(決済表の決済額がNULL)レコードである。したがって、空欄a2には「決済表.決済額 IS NULL OR 決済表.決済額 >= 0」が入る。

正しい組合せは(エ)である。

【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文の空欄に入れる適切な答えを選ぶ。

会員の氏名、電子メールアドレスは会員表で、販売ID、決済期限日は販売表でそれぞれ管理されているため、これらの表を使用する必要がある。また、支払手続が行われているかどうかは決済表で管理されているため、会員表と販売表に加えて、決済表も使用する必要がある。

解答群を見ると、選択肢の違いはFROM句、WHERE句内の販売表.販売IDに関する条件、決済表などに関する条件である。そこで、選択肢の違いから考えていく。

ア：WHERE句内の副問合せで決済表にない販売IDの販売を抽出している。支払手続が行われていない販売は、決済表にレコードのない販売であるため、この条件は電子メールの送信条件に適合している。正しい。

イ：WHERE句内の副問合せで決済表から決済額が0円以上のレコードの販売IDを抽出し、販売表と結合している。しかし、この条件で抽出されるのは支払手続が行われた販売であり、電子メールの送信条件とは異なる。誤りである。

ウ：WHERE句で会員表、販売表、決済表を結合しているが、この条件で抽出されるのは決済済み、又は取消し済みの販売であり、電子メールの送信条件とは異なる。誤りである。

エ：WHERE句で会員表、販売表、決済表を結合し、決済表.決済額が-1でないレコードを抽出している。しかし、この条件で抽出されるのは支払手続が行われた販売であり、電子メールの送信条件とは異なる。誤りである。

したがって、正解は(ア)である。

【設問4】

ポイント制度の導入に当たり、決済表の付与ポイントを更新する正しいSQL文を選ぶ。

設問文では、ポイントを更新するSQL文の条件として「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」とあり、決済額によって付与するポイントが異なることが分かる。

SQL文において、条件によってアクションを変更したい場合は、CASE式を使用する。

<書式>

(単純CASE式)

```
CASE 項目名
WHEN 値1 THEN 項目名 = 値1の場合のアクション
WHEN 値2 THEN 項目名 = 値2の場合のアクション
...
WHEN 値X THEN 項目名 = 値Xの場合のアクション
ELSE 値1〜Xに該当しない場合のアクション
END
```

(検索CASE式)

```
CASE
WHEN 条件1 THEN 条件1に該当する場合のアクション
WHEN 条件2 THEN 条件2に該当する場合のアクション
...
WHEN 条件X THEN 条件Xに該当する場合のアクション
ELSE 条件1〜Xに該当しない場合のアクション
END
```

単純CASE式は、評価対象の項目と値が同じかどうかを評価し、アクションを選択する。一方、検索CASE式は、条件を自由に設定できるため、単純CASE式に比べて条件分岐の応用の幅が広い。

解答群から書式にあったものを選べばよく、(ウ)が検索CASE式である。

```
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
```

なお、このUPDATE文で付与ポイントを更新すると、取り消された申込み(決済額=-1円)のレコードも更新され、付与ポイントに-1が登録される。しかし、ポイント残高に加える処理の中で配慮すればよいので、ここでは問題ないと考えられる。ア、エ：SQLにIF文はなく、誤りである。

イ：検索CASE式の書式であるが、一つ目の条件の前にWHEN句がないため、誤りである。