

問3 遊園地の入園者情報を管理する関係データベース (データベース) (H28 春・FE 午後問3)

【解答】

- [設問1] aーオ, bーエ
[設問2] cーイ
[設問3] dーウ
[設問4] エ

【解説】

遊園地の入園者情報を管理する関係データベースを題材にした問題である。設問1は、再設計（入園者表に退園時刻、利用表の追加など）したことによって、新たに可能となった事項についての記述の空欄穴埋め、設問2はアトラクションごとの延べ利用者数を表示するSQL文のSELECT句の空欄穴埋め、設問3は入園券でアトラクションを利用した入園者のうち、退園時の精算において2,000円以上を支払った入園者について、精算額が多い入園者から降順に、入園者番号と精算額を表示するSQL文のWHERE句の空欄穴埋め、設問4は会員情報を格納する会員表を作成し、入園者表に会員番号を格納する項目を追加した後に、設問のSQL文で抽出できる会員について正しい答えを選ぶ。

設問1は新たに追加された項目と利用表などを見ればすぐに分かるだろう。設問2、3はSQLのグループ化と集合関数の初歩的知識があれば易しい。設問4は丁寧に選択肢を見ていく必要はあるが難しくはない。

[設問1]

再設計後のデータ構造に関する設問文の記述中の空欄a, bに入れる答えを選ぶ。
・空欄a：図2によると入園者表に退園時刻の項目を追加したのであるから、最初からある入園時刻との差をとれば滞在時間が分かるので、(オ)の「入園者の滞在時間」が入る。

- ・空欄b：図2の利用表を見ると入園者番号が「00025987」、「00025992」と二つあり、入園者がどのアトラクションを利用したか、その利用時刻とともに分かるようになっている。したがって、(エ)の「入園者のアトラクション利用状況」が入る。
ア：図1, 2の表には「アトラクションの待ち時間」に関する情報はない。
イ：図1, 2の表には「休園日」に関する情報はない。
ウ：入園者数は、図1の入園者表の行数をカウントすれば分かるので、該当しない。

[設問2]

アトラクションごとの延べ利用者数を表示するSQL文のSELECT句の空欄cに入れる答えを選ぶ。
ア：AVG(利用表.アトラクション番号)……アトラクション番号ごとにグループ化されたレコード間で、アトラクション番号の（算術）平均をとってもアトラクション番号が分かるだけであり、該当しない。
イ：COUNT(*)……利用表をアトラクション番号ごとにグループ化し、グループごとの件数をカウントすることによって各アトラクションの利用回数、言い換えると延べ利用者数を求めることができるので、正解である。
ウ：MAX(利用表.アトラクション番号)……アトラクション番号ごとにグループ化されたレコード間で、アトラクション番号の最大値をとってもアトラクション番号が分かるだけであり、該当しない。
エ：SUM(利用表.アトラクション番号)……アトラクション番号ごとにグループ化されたレコード間で、アトラクション番号の合計をとってもここでは意味がなく、該当しない。

```
SELECT 利用表.アトラクション番号, アトラクション表.アトラクション名,  
       (イ):COUNT(*)  
FROM 利用表, アトラクション表  
WHERE 利用表.アトラクション番号 = アトラクション表.アトラクション番号  
GROUP BY 利用表.アトラクション番号, アトラクション表.アトラクション名
```

[設問3]

入園券でアトラクションを利用した入園者のうち、退園時の精算において2,000円以上を支払った入園者について、精算額が多い入園者から降順に、入園者番号と精算額を表示するSQL文のWHERE句の空欄dに入れる答えを選ぶ。

ア： 入園者表.券種 = '01' AND
アトラクション表.料金 >= 2000 …… ①
GROUP BY 利用表.アトラクション番号 …… ②
① アトラクション表のアトラクションごとの料金となり、退園時の精算料金とはならない。
② 利用表のアトラクション番号でグループ化しても意味はない。
2,000円以上のアトラクションを利用した入園者をアトラクションごとに表示しようとするが、入園者表の入園者番号が集計項目ではなくアトラクションごとには出力できずにSQLエラーとなる。

イ： 入園者表.券種 = '01'
GROUP BY 入園者表.入園者番号 …… ①
HAVING MAX(アトラクション表.料金) >= 2000 …… ②
① 入園者番号ごとにグループ化するのは正しい。
② 入園者が利用したアトラクション表の最大利用料金となり、退園時の精算料金とはならない。
2,000円に満たないアトラクションの利用を重ねて精算額が2,000円を超える入園者が表示されない。

ウ： 入園者表.券種 = '01'
GROUP BY 入園者表.入園者番号 …… ①
HAVING SUM(アトラクション表.料金) >= 2000 …… ②
① 入園者番号ごとにグループ化するのは正しい。
② 入園者が利用したアトラクション表の合計料金が2,000円以上となり、退園時の精算料金となる。
正解である。

エ： 入園者表.券種 = '01'
GROUP BY 利用表.アトラクション番号 …… ①
HAVING SUM(アトラクション表.料金) >= 2000 …… ②
① 利用表のアトラクション番号でグループ化しても意味はない。
② 利用表のアトラクション番号でグループ化しており、グループ化の条件で合計に意味はない。

合計利用料金が2,000円以上のアトラクションを利用した入園者をアトラクションごとに表示しようとするが、入園者表の入園者番号が集計項目ではなくアトラクションごとには出力できずにSQLエラーとなる。

```
SELECT 入園者表.入園者番号, SUM(アトラクション表.料金) AS 支払金額  
FROM 入園者表, 利用表, アトラクション表  
WHERE 入園者表.入園者番号 = 利用表.入園者番号 AND  
       利用表.アトラクション番号 = アトラクション表.アトラクション番号 AND
```

(ウ) 入園者表.券種 = '01'
GROUP BY 入園者表.入園者番号
HAVING SUM(アトラクション表.料金) >= 2000

ORDER BY 支払金額 DESC

[設問4]

会員情報を格納する会員表を作成し、入園者表に会員番号を格納する項目を追加した後に、設問のSQL文で抽出できる会員について正しい答えを選ぶ。

```
SELECT 会員表.会員番号, 会員表.氏名 …… ①  
FROM 会員表, 入園者表 …… ②  
WHERE 会員表.会員番号 = 入園者表.会員番号 …… ③  
       GROUP BY 会員表.会員番号, 会員表.氏名 …… ④  
       HAVING MAX(入園者表.入園日) < '20160101' …… ⑤
```

- ① 会員表の会員番号と氏名を表示
② 会員表と入園者表を結合対象とする。
③ 会員番号で結合
④ 会員表の会員番号でグループ化しているが、その上のWHERE句に「会員表.会員番号 = 入園者表.会員番号」の会員表と入園者表の結合条件があるので、GROUP BY 句の指定は「入園者表.会員番号, 会員表.氏名」としても等価である。なお、GROUP BY 句の会員表.氏名は、SELECT 句に会員表.氏名が指定されているので必要であるが、情報処理技術者試験が準拠する標準SQL (SQL2003から) では必須ではなく、GROUP BY 句には会員表.会員番号 (又は入園者表.会員番号) だけでもよい。新しいGROUP BY 句の仕様に準拠した問題はまだ出題されていないので、考慮する必要はないだろう。
⑤ 入園者表を会員番号でグループ化して、その入園者（会員）の最大の入園日 (最近の入園日) が2016年の1月1日より小さいという条件は「2016年より前に入園しただけ」、逆に言うと「2016年以降は1回も入園していない」という条件である。したがって、(エ)が正解である。次表の例では、会員番号＝000110の会員が2016年以降に1回も入園していないので、抽出されることになる。

入園者表 (一部省略)			
入園者番号	会員番号		入園日
00028707	000108		20160218
00028706	000108		20160201
00028705	000109		20160115
00028704	000109		20151225
00028703	000110		20151203
00028702	000110		20151103
00028701	—		20151010
MAX(入園者表.入園日)			

GROUP BY 会員番号

結合とグループ化の対象にならない

参考までに、設問のSQL文（結合とグループ化）と同じ結果が得られる別のSQL文（副問合せ）とSQL文（相関副問合せ）の例を示す。結合を使うSQL文は多くの場合、副問合せ、又は相関副問合せで記述できる。

・副問合せの例
SELECT 会員表.会員番号, 会員表.氏名
FROM 会員表
WHERE 会員表.会員番号 NOT IN(SELECT 入園者表.会員番号
FROM 入園者表
WHERE 入園者表.入園日 >= '20160101')

・相関副問合せの例
SELECT 会員表.会員番号, 会員表.氏名
FROM 会員表
WHERE NOT EXISTS (SELECT *
FROM 入園者表
WHERE 入園者表.会員番号 = 会員表.会員番号
AND 入園者表.入園日 >= '20160101')