

問020038解説

◆解答

- 設問1 a オ b エ
設問2 c イ
設問3 d ウ
設問4 エ

◆解説

遊園地の入園者情報を管理する問題である。

入園者情報

入園者表

入園者番号	券種	入園日	入園時刻	退園時刻
00025987	02	20150115	084517	150851
⋮	⋮	⋮	⋮	⋮

券種表

券種	券種名	料金
01	入園券	500
02	小学生パスポート	2000
⋮	⋮	⋮

利用表

入園者番号	利用時刻	アトラクション番号
00025987	091806	04
00025987	102611	05
00025992	102757	05
⋮	⋮	⋮

アトラクション表

アトラクション番号	アトラクション名	料金
01	ジェットコースター	500
⋮	⋮	⋮

システム導入後の表の構成とデータの格納例

当初の入園者表には退園時刻の項目がなかった。この項目を付加することによって入園者の滞在時間が把握できるようになった。更に、利用表とアトラクション表が付け加えられた。この2つの情報を加えることによって、アトラクションの利用状況が分かるようになった。2つの表によって、アトラクション別の延べ利用者数が求められるようになった。

GROUP BY を使用したSQLの基本文

商品番号毎の受注数量の合計を表示する。

```
SELECT 商品番号, SUM(数量) FROM 受注明細表 GROUP BY 商品番号
```

GROUP BY 句はグループ化を伴う操作であり、「どの表から」、「グループ化して」、「式の結果を取り出す」の構文である。特定の列の値に従って行をグループ化する。特定の列の値が同じ行をまとめて、表をいくつかのグループに分ける場合に使用する。GROUP BYに指定できるのは、列名でSELECT文節に表れているものである。SELECT文節にグループ化する列名と集計関数がある場合、GROUP BYの指定がないとエラーになる。また、GROUP BYを指定すると、SELECT文節で指定できるものは、GROUP BYで指定した列名またはその列名による集計関数しか記述することができない。GROUP BYによってグループ化する場合、グループの中で値が一意に定まらない列名を、SELECT文節で指定することができない。ただし、GROUP BYで指定した列名をSELECT文節で必ずしも指定する必要はない。

複数の列を対象にグループ化する場合は、GROUP BY 列名, ..., 列名の形式を用いる。集計関数を複数個使用する場合、次の構文になる。

```
SELECT 列名, 集計関数(列名), ..., 集計関数(列名) FROM 表名 GROUP BY 列名
```

受注明細表は、(伝票番号、商品番号、顧客番号、数量)のレコード様式である。

GROUP BY を使用したSQL文の検索例

受注表から2件以上の注文を受けている顧客の最新の受注月日を求める。

```
SELECT 顧客番号, MAX(受注日) FROM 受注表  
GROUP BY 顧客番号 HAVING COUNT(*) >= 2
```

最新の受注日を求めるためにMAX関数を利用する。HAVING文節は、GROUP BYによって、グループ化されたデータに検索条件を設定し、データを絞り込むことができる。

WHEREとGROUP BY を指定すると、検索条件が与えられた後にグループ化が行われる。GROUP BYとHAVINGを使用すると、グループ化が行われた後に検索条件が与えられる。GROUP BYと集関数を使用して値を求め、その結果を条件に従って評価する場合、条件式にHAVINGを使用する。WHEREを使用するとエラーになる。

COUNT(*)は、行数(レコード数)のカウントである。重複を許す場合である。COUNT(DISTINCT, 列名)の場合は重複を除いた行数である。

アトラクション毎の延べ利用者数を調査するSQL文

```
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, 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  
GROUPBY 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
```

グループ化されたレコードの件数を調べるのはCOUNT(*)を用いる。SQL文は上に示す内容となる。

WHERE GROUP BY HAVING を含むSQL文の検索

受注表から2件以上の注文を受けている顧客の最新の受注月日を求める。

```
SELECT 顧客番号, MAX(受注日) FROM 受注表
      GROUP BY 顧客番号 HAVING COUNT(*) >= 2
```

最新の受注日を求めるためにMAX関数を利用する。HAVING文節は、GROUP BYによって、グループ化されたデータに検索条件を設定し、データを絞り込むことができる。

WHEREとGROUP BYを指定すると、検索条件が与えられた後にグループ化が行われる。GROUP BYとHAVINGを使用すると、グループ化が行われた後に検索条件が与えられる。GROUP BYと集合関数を使用して値を求め、その結果を条件に従って評価する場合、条件式にHAVINGを使用する。WHEREを使用するとエラーになる。

COUNT(*)は、行数(レコード数)のカウントである。重複を許す場合である。COUNT(DISTINCT, 列名)の場合は重複を除いた行数である。

アトラクション利用者の退園時の利用金額の計算と表示のSQL文

- ① 結果の表示内容 入園者番号、アトラクション料金の合計
- ② 対象者は、アトラクション料金2000円以上、券種が“01”の人
- ③ 表示方法は精算金額の多い順に降順で表示する。
- ④ 使用する表は、入園者表、利用表、アトラクション表
- ⑤ 検索条件は、入園者表の入園者番号と利用表の入園者番号、利用表のアトラクション番号とアトラクション表のアトラクション番号、入園者表の券種が01、精算金額が2000円以上の入園者

会員表

会員番号	氏名	住所	生年月日	入会日
000108	情報太郎	東京都文京区〇〇〇	19920901	20160114
⋮	⋮	⋮	⋮	⋮

入園者表

入園者番号	会員番号	券種	入園日	入園時刻	退園時刻
00028707	000108	04	20160218	100555	182021
⋮	⋮	⋮	⋮	⋮	⋮

会員情報を追加した表の構成とデータの格納例

```
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
      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
ORDERBY 支払金額 DESC

設問 1

a は、当初の入園者表には退園時刻の項目がなかったが、再設計した表には退園時刻の項目を付加されたため入園者の滞在時間が把握できるようになり、a の求める答えはオとなる。

b は、利用表とアトラクション表が付け加えられたことによって、入園者のアトラクションの利用状況が分かるようになった。求める答えはエとなる。

設問 2

アトラクション毎の延べ利用者数を表示する問題である。利用表とアトラクション表を使用して、利用表のアトラクション番号とアトラクション表のアトラクション番号が一致するレコードを対象にアトラクション番号、アトラクション名でグループ化し、グループ内のレコード件数をカウントすればよい。条件に一致するレコードのカウントにはCOUNT(*)を使用する。SQL文は次のようになり、下線の部分が答えになる。

アトラクション毎の延べ利用者数を調査するSQL文は次のようになる。

```
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, 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
GROUPBY 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
下線の部分が答えで、求める答えはイとなる。
```

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、精算額が2000円以上の人を精算額の多い順に降順に表示する内容である。SQL文は次のようになる。

```
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
      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
ORDERBY 支払金額 DESC
```

下線の部分が答えで、求める答えはウとなる。

設問4

最新の入園日を求めるためにMAX関数を利用する。HAVING文節は、GROUP BYによって、グループ化されたデータに検索条件を設定し、データを絞り込むことができる。次のSELECT文は分析すると、次のようになる。

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

- ① 会員表と入園者表を使用して、WHERE 以降の条件に一致する会員番号と氏名を抽出する。
- ② 会員表の会員番号と入園者表の会員番号の一致するレコードを対象とする。
- ③ 会員表の会員番号と会員表の氏名を使用してグループ化する。
- ④ グループ化したレコードを対象に最新の入園日が2016年1月1日より前の人を対象とする。

答えは2016年以降に1回も入園していない会員が抽出されることになる。求める答えはエとなる。