

- 【解答】
- 【設問1】 aーイ, bーエ
- 【設問2】 ヲ
- 【設問3】 cーウ
- 【設問4】 dーエ

【解説】

自治会員の情報を管理する関係データベースの設計及び運用問題であり、設問1がデータベースの設計 (正規化)、設問2～4がSQLの問題である。設問1は、午前レベルの表の正規化の知識があれば、解答できる問題である。SQLの問題は、紛らわしい選択肢が含まれるものもあり、やや難しい感があるが、一つずつ SQL 文の実行結果を吟味していけば、解答できる問題である。

【設問1】

図1の帳票から図2の表構成を完成するまでの、テーブル設計に関する記述の空欄に入れる内容が問われている。図1の帳票は非正規形であるため、関係データベースにするには、表の正規化が必要である。図2の表には、一部省略されている項目があるが、ここでは、正規化の理解を深めるため、一部省略されている項目をも含めて、非正規形から第3正規形まで順を追って正規化を行う。

世帯主	氏名	住所	電話番号	登録日	班
	情報太郎	桜ヶ丘304ー8	999-999-9999	2005年4月7日	
世帯情報	氏名	続柄	性別	生年月日	備考
	情報太郎	本人	男	1969年1月23日	
	情報花子	配偶者	女	1972年8月24日	
	情報一郎	子	男	2004年6月13日	
	情報二郎	子	男	2007年1月5日	

図 A (図1と同じ)

図Aの表は、世帯の情報と、世帯ごとの住人の情報が、組み合わせられて一つの表になっている。表の行によって、項目数が変わる帳票のようなレアウトとなっており、そのままで서는リレーショナルDBに格納することができない。そのため、二つの表に分割する必要がある。

■第1正規形への変形

世帯主の情報と世帯情報に関連する属性を別の表に分離する。

世帯主	氏名	住所	電話番号	登録日	班
	情報太郎	桜ヶ丘304ー8	999-999-9999	2005年4月7日	2

世帯情報	氏名	続柄	性別	生年月日	備考
	情報太郎	本人	男	1969年1月23日	
	情報花子	配偶者	女	1972年8月24日	
	情報一郎	子	男	2004年6月13日	
	情報二郎	子	男	2007年1月5日	

図 B 図 A の表を分割した表

このままでは、各表ともにキーが氏名 (文字) となり同姓同名などの場合、重複データが発生し、行を一意に識別できないことが想定される。そこで、世帯を識別する識別子と、会員を識別するための識別子が必要である。

そのため、図2及び「表の説明」から、世帯表の主キーとして世帯番号が、会員表の主キーとして会員番号が追加されていることが分かる。

また、図2にならい、続柄とその名称 (続柄名称) と性別と名称 (性別名称) にコードを付けると、表Cのようになる。

世帯番号	世帯主番号	住所	電話番号	登録日	班
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02
⋮	⋮	⋮	⋮	⋮	⋮

世帯表	世帯番号	氏名	続柄	続柄名称	性別	性別名称	生年月日	備考
会員番号	0412	0181	情報太郎	01	本人	01	1969年1月23日	
	0413	0181	情報花子	02	配偶者	02	1972年8月24日	
	0414	0181	情報一郎	03	子	01	2004年6月13日	
	0415	0181	情報二郎	03	子	01	2007年1月5日	
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

図 C 第1正規形の表

これで、世帯情報の繰返しが排除されているので、第一正規形を満たしている。よって空欄aは、「第1正規化に基づいて」の (イ) が正解となる。また、図Cの会員表は、主キーが会員番号だけなので、第2正規形を満たしている。第2正規形は、部分関数従属を排除することである。部分関数従属の排除とは、部分に従属する項目を排除することである。例えば、主キーが {A, B} の2項目の組合せの場合、{A, B} の部分は、AとBとなり、Aだけに従属する項目とBだけに属する項目は、部分関数従属の項目である。この表の場合、主キーが単独の主キーであるので、部分関数従属の項目は存在しない。したがって、既に、第2正規形を満たしている。

■続柄表と性別表の分離

図Cは、第2正規形を満たしているが、会員表の会員番号、続柄、続柄名称に注目すると、会員番号が決まれば続柄が決まり、続柄が決まれば続柄名称が決まるので、これは推移的関数従属が存在することになる。また、会員番号、性別、性別名称に注目すると、会員番号が決まれば性別が決まり、性別が決まれば性別名称が決まるので、ここにも推移的関数従属が存在することになる。第3正規形を満たすには、推移的関数従属を排除する必要がある。そこで、図2のように、続柄表と性別表を分割すると、推移的関数従属を排除できるので、第3正規形を満たすことができる。したがって、空欄bは、「第3正規化に基づいて」の (エ) が正解となる。第3正規形において、続柄表と性別表とに分割することによって、“男”と、“男性”のような、入力の違いを防止することができる。

【設問2】

1940年よりも前に生まれた会員が含まれる世帯の世帯番号を抽出するSQL文が問われている。なお、同じ世帯番号は一つだけ抽出することに注意する必要がある。

1個ずつSQLの実行結果をイメージしながら確認していく必要がある。次に、解答群を順番に見ていく。

ア：SELECT DISTINCT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19400101

会員表の生年月日が19400101以上、すなわち1940年1月1日以降に生まれた会員を含む世帯番号を重複なしで抽出するので、条件式が不適切である。なお、SELECT文において重複なしの場合DISTINCTを使用するので、(ア)を選択しがちであるが、注意する必要がある。

イ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19391231

会員表の生年月日が19391231以上、すなわち1939年12月31日以降に生まれた会員を含む世帯番号を抽出するので、これも条件式が不適切である。

ウ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 < 19400101

GROUP BY 世帯番号

会員表の生年月日が19400101未満、すなわち1940年1月1日より前に生まれた会員を含む世帯番号を、世帯番号でグループ化して抽出する。GROUP BYで世帯番号ごとにグループ化するので、重複しないので、これは正しいSQLである。

エ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 <= 19391231

GROUP BY 世帯番号, 会員番号

会員表の生年月日が19391231以下、すなわち1939年12月31日までに生まれた会員を会員番号と世帯番号でグループ化している。そのSQLは、GROUP BYで世帯番号と会員番号の二つの項目があるために、最初に世帯番号でグループ化した後に、二つ目の項目として、世帯番号に属する会員番号の組合せを抽出する。SELECT文には世帯番号だけであるため世帯番号だけが表示されるが、DBの内部的には、世帯番号と会員番号の組合せで、抽出している点に注意したい。例えば、世帯番号100番の世帯にWHERE条件を満たしている会員Aと会員Bが存在する場合を考えると、SQLの結果は内部的には 100 A と 100 B の2行が抽出され、SELECTで世帯番号だけが表示されるため 100 が2行表示される。そのため、世帯番号が重複するので、不適切である。

したがって、(ウ) が正解である。

【設問3】

班ごとの世帯数と会員数を集計するSQLが問われている。

```
SELECT 世帯表.班,
FROM 世帯表, 会員表
WHERE 世帯表.世帯番号 = 会員表.世帯番号
GROUP BY 世帯表.班
```

GROUP BYで世帯表.班を指定しているので、DBの内部では図Dのように、班ごと世帯番号でひも付けされた世帯表と会員表の項目をまとめることになる。

世帯番号	世帯主番号	住所	電話番号	登録日	班	会員番号	世帯番号	氏名	続柄	...
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02	0412	0181	情報太郎	01	
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02	0413	0181	情報花子	02	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	

図 D DB内部にある表のイメージ

次に、解答群を順番に見ていく。

ア：COUNT(\*), COUNT(会員表.会員番号)

COUNT(\*)とすると、班ごとの世帯数ではなく、会員数がカウントされる。更にCOUNT(会員表.会員番号)としても、班ごとの会員数が集計されるので、不適切である。

イ：COUNT(\*), MAX(会員表.会員番号)

MAX(会員表.会員番号)とすると、班ごとの会員のうち、会員番号が一番大きい値を選択するので、不適切である。

ウ：COUNT(DISTINCT 世帯表.世帯番号), COUNT(\*)

COUNT(DISTINCT 世帯表.世帯番号)で、班ごとの世帯番号を重複なしでカウントしている。これは、班ごとの世帯数を集計しており、その後のCOUNT(\*)で、会員番号をカウントしている。これは、班ごとの会員数を集計している。したがって、適切である。

エ：COUNT(世帯表.世帯番号), MAX(会員表.会員番号)

これは、DISTINCTがないので、世帯番号が重複してカウントされてしまう。また、MAX(会員表.会員番号)も、全会員の中での会員番号最大値を選択するので、不適切である。

したがって、(ウ) が正解である。

- 【解答】
- 【設問1】 aーイ, bーエ
- 【設問2】 ヲ
- 【設問3】 cーウ
- 【設問4】 dーエ

【解説】

自治会員の情報を管理する関係データベースの設計及び運用問題であり、設問1がデータベースの設計 (正規化)、設問2～4がSQLの問題である。設問1は、午前レベルの表の正規化の知識があれば、解答できる問題である。SQLの問題は、紛らわしい選択肢が含まれるものもあり、やや難しい感があるが、一つずつ SQL 文の実行結果を吟味していけば、解答できる問題である。

【設問1】

図1の帳票から図2の表構成を完成するまでの、テーブル設計に関する記述の空欄に入れる内容が問われている。図1の帳票は非正規形であるため、関係データベースにするには、表の正規化が必要である。図2の表には、一部省略されている項目があるが、ここでは、正規化の理解を深めるため、一部省略されている項目をも含めて、非正規形から第3正規形まで順を追って正規化を行う。

世帯主	氏名	住所	電話番号	登録日	班
	情報太郎	桜ヶ丘304ー8	999-999-9999	2005年4月7日	2
世帯情報	氏名	続柄	性別	生年月日	備考
	情報太郎	本人	男	1969年1月23日	
	情報花子	配偶者	女	1972年8月24日	
	情報一郎	子	男	2004年6月13日	
	情報二郎	子	男	2007年1月5日	

図 A (図1と同じ)

図Aの表は、世帯の情報と、世帯ごとの住人の情報が、組み合わせられて一つの表になっている。表の行によって、項目数が変わる帳票のようなレイアウトとなっており、そのままで서는リレーショナルDBに格納することができない。そのため、二つの表に分割する必要がある。

■第1正規形への変形

世帯主の情報と世帯情報に関連する属性を別の表に分離する。

世帯主	氏名	住所	電話番号	登録日	班
	情報太郎	桜ヶ丘304ー8	999-999-9999	2005年4月7日	2

世帯情報	氏名	続柄	性別	生年月日	備考
	情報太郎	本人	男	1969年1月23日	
	情報花子	配偶者	女	1972年8月24日	
	情報一郎	子	男	2004年6月13日	
	情報二郎	子	男	2007年1月5日	

図 B 図 A の表を分割した表

このままでは、各表ともにキーが氏名 (文字) となり同姓同名などの場合、重複データが発生し、行を一意に識別できないことが想定される。そこで、世帯を識別する識別子と、会員を識別するための識別子が必要である。

そのため、図2及び [表の説明] から、世帯表の主キーとして世帯番号が、会員表の主キーとして会員番号が追加されていることが分かる。

また、図2にならい、続柄とその名称 (続柄名称) と性別と名称 (性別名称) にコードを付けると、表Cのようになる。

世帯番号	世帯主番号	住所	電話番号	登録日	班
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02
⋮	⋮	⋮	⋮	⋮	⋮

会員表

会員番号	世帯番号	氏名	続柄	続柄名称	性別	性別名称	生年月日	備考
0412	0181	情報太郎	01	本人	01	男性	1969年1月23日	
0413	0181	情報花子	02	配偶者	02	女性	1972年8月24日	
0414	0181	情報一郎	03	子	01	男性	2004年6月13日	
0415	0181	情報二郎	03	子	01	男性	2007年1月5日	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

図 C 第1正規形の表

これで、世帯情報の繰返しが排除されているので、第一正規形を満たしている。よって空欄aは、「第1正規化に基づいて」の (イ) が正解となる。また、図Cの会員表は、主キーが会員番号だけなので、第2正規形を満たしている。第2正規形は、部分関数従属を排除することである。部分関数従属の排除とは、部分に従属する項目を排除することである。例えば、主キーが {A, B} の2項目の組合せの場合、{A, B} の部分は、AとBとなり、Aだけに従属する項目とBだけに属する項目は、部分関数従属の項目である。この表の場合、主キーが単独の主キーであるので、部分関数従属の項目は存在しない。したがって、既に、第2正規形を満たしている。

■続柄表と性別表の分離

図Cは、第2正規形を満たしているが、会員表の会員番号、続柄、続柄名称に注目すると、会員番号が決まれば続柄が決まり、続柄が決まれば続柄名称が決まるので、これは推移的関数従属が存在することになる。また、会員番号、性別、性別名称に注目すると、会員番号が決まれば性別が決まり、性別が決まれば性別名称が決まるので、ここにも推移的関数従属が存在することになる。第3正規形を満たすには、推移的関数従属を排除する必要がある。そこで、図2のように、続柄表と性別表を分割すると、推移的関数従属を排除できるので、第3正規形を満たすことができる。したがって、空欄bは、「第3正規化に基づいて」の (エ) が正解となる。第3正規形において、続柄表と性別表とに分割することによって、“男”と、“男性”のような、入力の違いを防止することができる。

【設問2】

1940年よりも前に生まれた会員が含まれる世帯の世帯番号を抽出するSQL文が問われている。なお、同じ世帯番号は一つだけ抽出することに注意する必要がある。1個ずつSQLの実行結果をイメージしながら確認していく必要がある。次に、解答群を順番に見ていく。

ア：SELECT DISTINCT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19400101

会員表の生年月日が19400101以上、すなわち1940年1月1日以降に生まれた会員を含む世帯番号を重複なしで抽出するので、条件式が不適切である。なお、SELECT文において重複なしの場合DISTINCTを使用するので、(ア)を選択しがちであるが、注意する必要がある。

イ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 >= 19391231

会員表の生年月日が19391231以上、すなわち1939年12月31日以降に生まれた会員を含む世帯番号を抽出するので、これも条件式が不適切である。

ウ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 < 19400101

GROUP BY 世帯番号

会員表の生年月日が19400101未満、すなわち1940年1月1日より前に生まれた会員を含む世帯番号を、世帯番号でグループ化して抽出する。GROUP BYで世帯番号ごとにグループ化するので、重複しないので、これは正しいSQLである。

エ：SELECT 世帯番号 FROM 会員表 WHERE 会員表.生年月日 <= 19391231

GROUP BY 世帯番号, 会員番号

会員表の生年月日が19391231以下、すなわち1939年12月31日までに生まれた会員を会員番号と世帯番号でグループ化している。そのSQLは、GROUP BYで世帯番号と会員番号の二つの項目があるために、最初に世帯番号でグループ化した後に、二つ目の項目として、世帯番号に属する会員番号の組合せを抽出する。SELECT文には世帯番号だけであるため世帯番号だけが表示されるが、DBの内部的には、世帯番号と会員番号の組合せで、抽出している点に注意したい。例えば、世帯番号100番の世帯にWHERE条件を満たしている会員Aと会員Bが存在する場合を考えると、SQLの結果は内部的には 100 A と 100 B の2行が抽出され、SELECTで世帯番号だけが表示されるため 100 が2行表示される。そのため、世帯番号が重複するので、不適切である。

したがって、(ウ) が正解である。

【設問3】

班ごとの世帯数と会員数を集計するSQLが問われている。

```
SELECT 世帯表.班,  
FROM 世帯表, 会員表
WHERE 世帯表.世帯番号 = 会員表.世帯番号
GROUP BY 世帯表.班
```

GROUP BYで世帯表.班を指定しているので、DBの内部では図Dのように、班ごと世帯番号でひも付けされた世帯表と会員表の項目をまとめることになる。

世帯番号	世帯主番号	住所	電話番号	登録日	班	会員番号	世帯番号	氏名	続柄	...
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02	0412	0181	情報太郎	01	
0181	0412	桜ヶ丘304ー8	999-999-9999	20050407	02	0413	0181	情報花子	02	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	
Xxxx	xxxx	xxxx	xxxx	xxxx	01	xxxx	xxxx	Xxxx	xx	

図 D DB内部にある表のイメージ

次に、解答群を順番に見ていく。

ア：COUNT(\*), COUNT(会員表.会員番号)

COUNT(\*)とすると、班ごとの世帯数ではなく、会員数がカウントされる。更にCOUNT(会員表.会員番号)としても、班ごとの会員数が集計されるので、不適切である。

イ：COUNT(\*), MAX(会員表.会員番号)

MAX(会員表.会員番号)とすると、班ごとの会員のうち、会員番号が一番大きい値を選択するので、不適切である。

ウ：COUNT(DISTINCT 世帯表.世帯番号), COUNT(\*)

COUNT(DISTINCT 世帯表.世帯番号)で、班ごとの世帯番号を重複なしでカウントしている。これは、班ごとの世帯数を集計しており、その後のCOUNT(\*)で、会員番号をカウントしている。これは、班ごとの会員数を集計している。したがって、適切である。

エ：COUNT(世帯表.世帯番号), MAX(会員表.会員番号)

これは、DISTINCTがないので、世帯番号が重複してカウントされてしまう。また、MAX(会員表.会員番号)も、全会員の中での会員番号最大値を選択するので、不適切である。

したがって、(ウ) が正解である。