

第1回 解答・解説

問1

解答

設問1 aーエ bーア cーイ

設問2 dーア eーカ

解説

処理の概要

■プログラム1

プログラム1は、 $\text{Src}[\text{Sidx}] \neq \text{EOS}$ という継続条件をもつループで、Srcの文字列を先頭から1文字ずつ順に処理し、Dstに格納している。このループの中で、

$\text{Src}[\text{Sidx}] = \text{TAB}$

という条件を判定し、処理を分岐している。この条件は、

対象となる文字型配列中にタブ文字が出現した

ことを表すので、真の場合はタブ文字の展開を行うことになる。一方、偽の場合は、タブ文字ではない文字が出現したことになるので、タブ文字以外の文字は、そのままDstに格納する処理が行われる。

タブ文字の展開では、

①次の文字が最も近い右のタブ位置に格納され、タブ文字を一つ以上の間隔文字に置換できるようにタブ位置(TabPos)を、計算によってタブ間隔TabGapなどから求める。

②内側のループで、「タブ位置-1」文字目($\text{Dst}[\text{TabPos}-1]$)まで間隔文字(SPC)を順に格納する

という処理を行う。

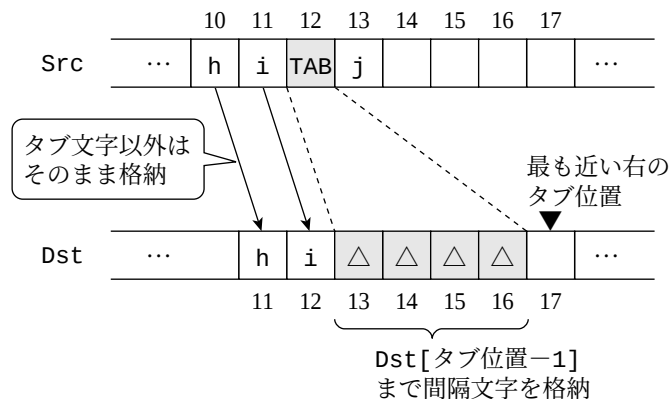


図1 各文字の処理

Srcのすべての文字を処理し終わったら、外側のループを抜けて「Dst[Didx] ← EOS」という処理を行い、Dst[Didx]に文字列の終わりを表すシステム定数EOSを格納する。

■プログラム2

プログラム2では、タブ文字が出現すると、配列TabSetを前回のタブ位置(最初は先頭要素)から線形探索することによって、タブ位置を求めるように処理内容が変更されている。

最初にTabPosをDidx+1に、Loopをtrueに初期化してから、ループにより次の格納位置Didxと各タブ位置を順に比較し、

Didx < TabSet[Tidx]

となる要素が見つかるまで探索処理を繰り返す。

見つかった「Didxよりも大きな値のタブ位置」とは、最も近い右のタブ位置のことであるから、TabPosに見つかったタブ位置TabSet[Tidx]を、Loopにfalseを設定し、ループを抜ける。

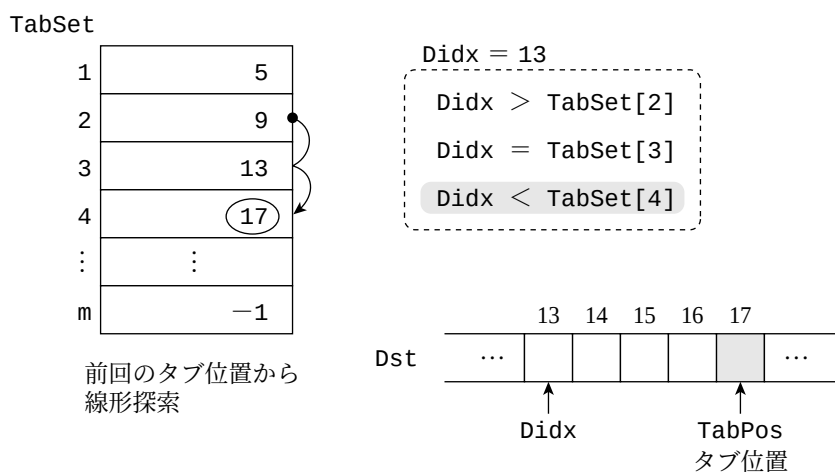


図2 タブ位置探索の例

一方、Didxよりも大きなTabSet[Tidx]が見つからなかった場合、最終的にTabSet[Tidx]は番兵の-1となってループを抜ける。この場合は、TabPosは更新されず初期値のDidx+1のままである。これにより、最後のタブ位置以降にタブ文字が出現した場合は、一つの間隔文字に置換することができる。

また、その後にタブ文字が何個出現しても、Tidxは変化しないため、常にTabSet[Tidx]は-1となり、タブ位置を決定するループは実行されず、TabPosはDidx+1となり、一つの間隔文字に置換することができる。

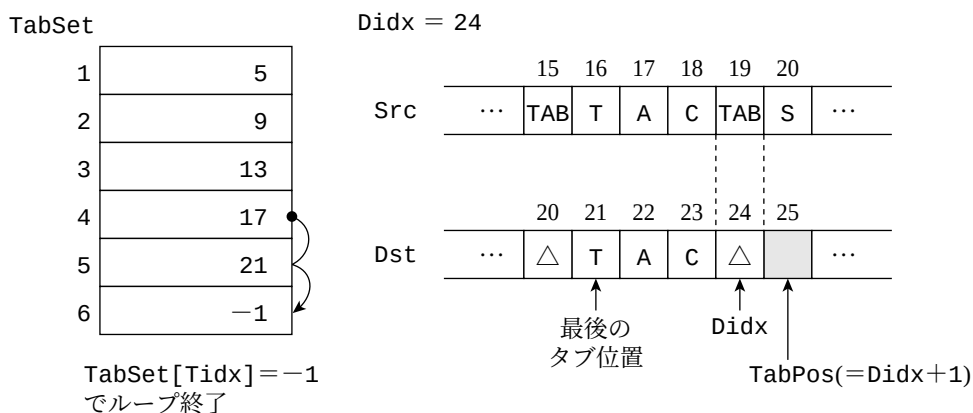


図3 最後のタブ位置以降に出現したタブ文字の場合

タブ位置を求めた後の処理は、プログラム1と同様である。

解答の導き方

設問1

プログラム1では、「Src[Sidx] ≠ EOS」という継続条件のループが最も外側にあり、このループに入る前にSidxが1で初期化され、ループの末尾でSidxを一つ増やしている。よって、このループは「Srcを先頭から1文字ずつ順に、最後の文字まで調べる」ループであることがわかる。このループ中で、タブ文字の展開およびDstへの格納が行われるはずである。

ループ中では、「Src[Sidx] = TAB」という条件を判定し、処理を分岐している。この条件は、「Src中にタブ文字が出現した」ことを表すので、真の場合はタブ文字の展開を行うことになる。

aについて

空欄aは、Src中にタブ文字が出現した場合の処理である。この場合は、

次の文字が最も近い右のタブ位置に格納されるように、

タブ文字を一つ以上の間隔文字に置換してDstに格納する

という処理を行えばよい。

空欄aを含む処理で最初にNの値を求めてから、

$$\bullet \text{TabPos} \leftarrow \text{TabGap} \times N + 1$$

という処理を行っている。問題文には、

$$\text{タブ位置} = \text{タブ間隔} \times n + 1$$

という同様の式が提示されているので、ここではタブ位置を求める処理が行われており、

TabPos：見つかったタブ位置

であることが分かる。

それでは、タブ間隔(TabGap)が4の場合、タブ位置がどのようになるかが図1に提示されているので、これに従って考えてみよう。ここで重要なのはDidxである。Didxは、処理結果を格納する配列Dstの添字(文字位置)として使用されており、格納前に1で初期化されていることから、Dstにおける「次の格納位置」を示していることがわかる。タブ文字が出現したときに

は、次の文字はこのDst[Didx]ではなく「最も近い右のタブ位置」に格納することから、求めるべき「最も近い右のタブ位置」とは、このDidxの示す格納位置を基準として考えることになる。

図1より、Srcにタブ文字が出現した時点でのDidx(図1では間隔文字の先頭になる)、そのとき得られるべきタブ位置TabPos、得られるべきNを、表1に示す。

表1 図1のDidxとタブ位置、Nの対応

タブ文字出現位置	Didx	TabPos	N
Src[3]	3	5	1
Src[7]	8	9	2
Src[12]	13	17	4

Didxの値とタブ間隔(TabGap)からいかにしてこのNを求めればよいかは、TabGapが4の場合について、表1(図1)の値を使って各選択肢について具体的に計算してみればよい。ここでは、〔プログラム1の説明〕(4)のタブ位置17の計算から順に検討してみる。なお、除算は整数除算を行うので、小数点以下が切り捨てられることに注意しよう。

〔Didx=13の場合〕

ア $(13+1) \div 4 = 3$

イ $(13-1) \div 4 = 3$

ウ $(13+4+1) \div 4 = 4$

エ $(13+4-1) \div 4 = 4$

→ 正しいNを求められるのは、ウとエ

〔Didx=8の場合〕

ウ $(8+4+1) \div 4 = 3$

エ $(8+4-1) \div 4 = 2$

→ 正しいNを求められるのは、エのみ

したがって、Nを求める計算式は、

$$(Didx + TabGap - 1) \div TabGap$$

となる。

なお、表2に、タブ間隔が4の場合のDidxの値と、そのときのタブ位置、Nの一般的な関係を示す。上式は、表2のどのDidxからも正しいNを計算できる。

表2 タブ間隔が4のときのDidxとタブ位置、Nの対応

Didx	タブ位置(TabPos)	N
1～4	5	1
5～8	9	2
9～12	13	3
13～16	17	4

bについて

空欄bを含むループ中では、結果を格納する配列の要素Dst[Didx]に間隔文字SPCを格納しているの、ここでは「タブ文字を一つ以上の間隔文字(SPC)に置換してDstに格納する」処理が行われていると判断できる。この処理は、「次の文字が最も近い右のタブ位置に格納される

ように」行う。ループの継続条件は、

$\text{Didx} < \text{TabPos}$

であるから、 $\text{Dst}[\text{TabPos}-1]$ まで間隔文字が格納されることになる。 TabPos には、空欄aで述べたように最も近い右のタブ位置が格納されているので、このループでは、

・ $\text{Dst}[\text{ループ開始時のDidx}] \sim \text{Dst}[\text{タブ位置}-1]$ まで間隔文字を格納するという処理が行われている(タブ位置には次の文字が入る)。

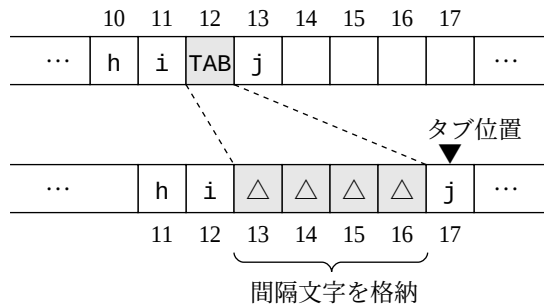


図4 タブ文字の置換処理

「文字型配列の各要素には、文字を1文字ずつ順に格納する」のであるから、 $\text{Dst}[\text{Didx}]$ に間隔文字を格納したならば、 Didx の値を一つ増やし次の格納位置を指すように更新すればよい。したがって、空欄bには、

$\text{Didx} \leftarrow \text{Didx} + 1$

が入る。

cについて

空欄cは、「 $\text{Src}[\text{Sidx}] = \text{TAB}$ 」の条件が偽の場合に行われる。この場合は、タブ文字ではない文字が出現したことになるので、問題文に従って「タブ文字以外の文字は、そのまま Dst に格納する」という処理を行えばよい。

処理対象の文字は $\text{Src}[\text{Sidx}]$ で、文字を格納する位置は $\text{Dst}[\text{Didx}]$ で表されるので、空欄cには、

$\text{Dst}[\text{Didx}] \leftarrow \text{Src}[\text{Sidx}]$

が入る。

設問2

d, eについて

設問2で用いられるプログラム2は、「タブ位置を任意の文字位置に指定できる」点がプログラム1と異なる。タブ位置は整数型配列 TabSet に格納されているので、プログラム1の

・ $N \leftarrow (\text{Didx} + \text{TabGap} - 1) \div \text{TabGap}$

・ $\text{TabPos} \leftarrow \text{TabGap} \times N + 1$

の部分が、「 TabSet を用いて TabPos を求める(探す)」という処理に改変されたことになる。

プログラム1と比較すると、ループの前に Tidx を1で初期化する処理が追加されており、タブ位置を求める処理がループ処理に置き換えられていることがわかるだろう。このループの継続条件に着目すると、

(TabSet[Tidx] ≠ -1) and (Loop = true)

となっているので、「TabSet[Tidx] が -1になる」か、「Loop が trueでない(falseである)」場合にループを抜けることになる。

続いて、このループでどのようにタブ位置を求めるのかを考える。問題文中に「Src中にタブ文字が出現した場合、次の文字が最も近い右のタブ位置に格納されるように」という記述があるので、ここでは空欄aでも述べたように格納位置Didxを基準として、

Didxよりも大きく(右にあり)、最小の(最も近い)タブ位置

を探せばよい。TabSetにはタブ位置が昇順で格納されており、TabSetの添字として用いられるTidxが1で初期化されていることから、

TabSet[1]

TabSet[2]

TabSet[3]

...

といった具合に先頭から順にDidxと比較(線形探索)すればよい。Didxより大きなタブ位置を格納した要素が出現(Didx < TabSet[Tidx])すると、その要素はDidxよりも大きな最小のタブ位置となっているはずである。この場合は、見つかったタブ位置をTabPosに設定してからLoopにfalseを設定し、ループを抜ければよい。したがって、空欄dには、

TabPos ← TabSet[Tidx]

が入る。

また、条件「Didx < TabSet[Tidx]」が偽の場合には、線形探索を続行するために、添字Tidxの値を一つ増やし、次の要素を指すように更新すればよい。よって、空欄eには、

Tidx ← Tidx + 1

が入る。

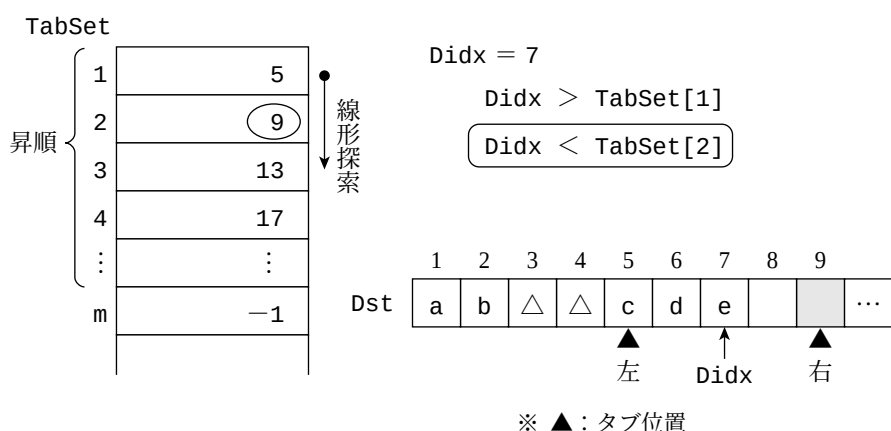


図5 タブ位置を線形探索

なお、ループ中にTidxの初期化処理は存在しないので、二つ目以降のタブ文字は前回のタブ位置から線形探索を開始することで、今回のタブ位置を見つけることになる。

問2

解答

設問1 aーイ

設問2 bーウ cーカ dーエ

解説

本問で取り上げられている変換の考え方は、音楽CDで用いられている、EFM(Eight-Fourteen Modulation：8/14変換)方式に近いものである。

EFMでは8ビットの情報を，“1”と“1”の間に必ず2～10個の“0”が挟まれるように14ビットに変換する。これによって同じビットが長く連続するということがなくなり，読取りの精度を高めることができる。

解答の導き方

設問1

aについて

ビット列は連続しているので，どこからどこまでが一つのチャンネルビットなのかを正しく認識しなければならない。ここで図2中に，「8個連続している」ビットがあることに注目する。

一つのチャンネルビット中で“0”が8個連続することはないので，ビット／ランドが8個連続するということは，

- ・あるチャンネルビットの末尾に“0”がm個連続している
- ・直後のチャンネルビットの先頭に“0”がn個連続している
- ・ $m+n$ が8となる

ことを意味している。表1の変換表においてこのような条件を満たすのは，

「あるチャンネルビットが“1001000000”であり，

直後のチャンネルビットが“001～”となる」

場合だけである。この組合せ以外では，“0”は最大でも7個しか連続しない。よって，区切り位置は8個の“0”を6+2に分割する位置，すなわちaであることがわかる。

あとは区切り位置以降について，

ビット／ランドの長さを列挙

→ 間に“1”を挟んだビット列に変換

→ 表1を用いて元のデータを復元

という手順で値を求めればよい。具体的な結果は図3のようになる。

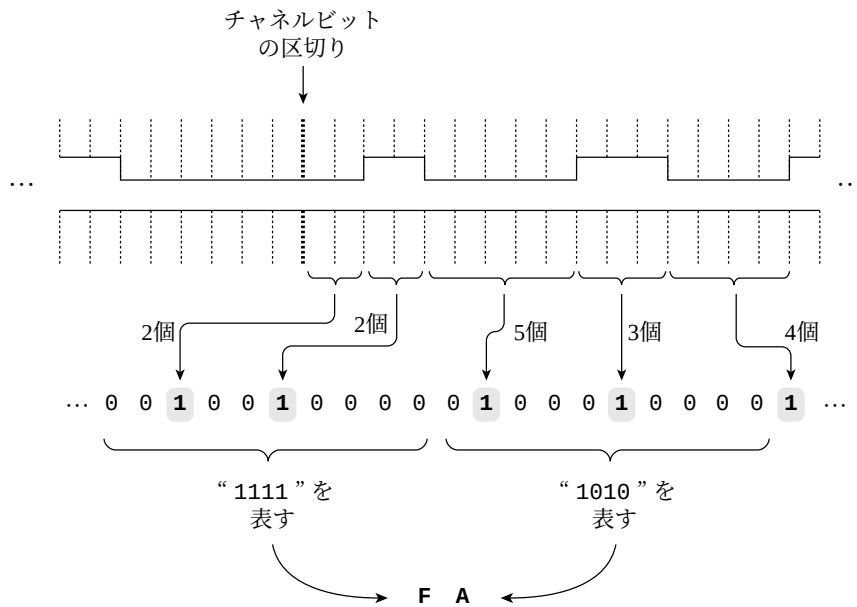


図3 結果

設問2

bについて

元のビット列が“0000”である場合、これを変換したチャネルビットは

1000000100

である。よって、「チャネルビット中の“0”の個数」は8個であり、これを記録するのに必要なビット／ランド長は、8ブロック分となる。

cについて

表1を見ると、チャネルビットには、“1”が2回登場するものと、3回登場するものがあることがわかる。これは

(a) “1”が2回登場するチャネルビット → 記録に $10 - 2 = 8$ ブロック分を要する

(b) “1”が3回登場するチャネルビット → 記録に $10 - 3 = 7$ ブロック分を要する

と言い換えることができる。

“0000”～“1111”までの16種類のチャネルビットを調べると、(a)が12個、(b)が4個ある。各種類の出現確率が等しい場合、4ビットの元データを記録するのに必要なビット／ランド長の平均は

$$\begin{aligned}
 &= \frac{(a)の個数 \times 8 + (b)の個数 \times 7}{16} \\
 &= \frac{12 \times 8 + 4 \times 7}{16} \\
 &= \frac{124}{16} = \frac{31}{4} \\
 &= 7.75 \text{ [ブロック分]}
 \end{aligned}$$

となる。問題文では「1バイト分」となっているので、答えはこれを2倍した
15.5 [ブロック分]
となる。

dについて

1バイトの記録に必要なトラックの長さは、
(1バイト分の記録に必要なブロック数)×(1ブロックの長さ)
 $= 15.5 \times 5 \times 10^{-7}$
 $= 77.5 \times 10^{-7}$ [メートル]

となる。よって、光ディスク1枚に記録できるデータ量は、

$$\begin{aligned} &= \frac{\text{トラックの総延長}}{\text{1バイトの記録に必要なトラック長}} \\ &= \frac{6.2 \times 10^3}{77.5 \times 10^{-7}} \\ &= \frac{6.2 \times 10^{10}}{77.5} \\ &= 0.08 \times 10^{10} \\ &= 800 \times 10^6 \text{ [バイト]} \\ &= 800 \text{ [Mバイト]} \end{aligned}$$

となる。

問3

解答

aーカ bーイ cーイ dーエ eーア fーア gーイ

解説

重点整理

■サブネットマスク

サブネットマスクは、IPアドレスのネットワーク部がすべて1、ホスト部がすべて0に設定された32ビットのビットパターンである。通常は、8ビットずつ区切って表現する。

まず、送信元では、あて先IPアドレスとサブネットマスクのビットごとのAND演算を行い、送信元IPアドレスとサブネットマスクのビットごとのAND演算を行った結果と比較をする。両者が等しい場合は、あて先は同一サブネット内なので直接あて先へ送信し、異なる場合はサブネット外なのでデフォルトゲートウェイへ送信する。

たとえば、サブネットマスクが255.255.255.0で、送信元IPアドレスが192.168.2.1、あて先IPアドレスが192.168.1.1の場合、

- ・送信元IPアドレスとサブネットマスクのビットごとのAND

```
11000000 10101000 00000010 00000001
AND 11111111 11111111 11111111 00000000
-----
11000000 10101000 00000010 00000000
```

- ・あて先IPアドレスとサブネットマスクのビットごとのAND

```
11000000 10101000 00000001 00000001
AND 11111111 11111111 11111111 00000000
-----
11000000 10101000 00000001 00000000
```

となり、AND演算の結果が異なる。この場合は、あて先IPアドレスはサブネット外であると判断できる。逆に、両者が等しい場合は、両IPアドレスのネットワーク部が一致していることを表し、あて先は同一サブネット内にあると判断できるので、あて先に直接パケットを送信すればよい。

■NAPT(IPマスカレード)

NAPT(IPマスカレード)機能をもつルータは、パケットの
{送信元IPアドレス(プライベート), 送信元ポート番号}
を
{送信元IPアドレス(グローバル), 新たに生成した送信元ポート番号}
に書き換え、これらの情報を記憶しておく。

これにより、複数のプライベートIPアドレスから一つのグローバルIPアドレスへの変換を行うと共に、あて先からの返信パケットを本来の送信元に正しく届けることができる。

解答の導き方

a, bについて

ここでは、あて先が送信元と同一のサブネット内にあるのか否かを判断する。あて先がサブネット内か外かにより、以下のように処理を分ける。この判断のために用いられるのが、サブネットマスクである。

サブネット内：あて先に直接パケットを送信

サブネット外：デフォルトゲートウェイにパケットを送信

PC1がパケットを送信するさいは、

送信元であるIP1とサブネットマスク、

あて先である“IPX (空欄a)”とサブネットマスク

のそれぞれについて、ビットごとのAND演算を行う。インターネット上のWebサーバXはサブネット外なので、このAND演算の結果は、

異なる (空欄b)

ものになる。

cについて

スイッチングハブは、あて先MACアドレスの情報をもとに、どのポート(接続口)へパケットを送信するかを決定する。ここで、PC1から受け取ったパケットのあて先MACアドレスは、ルータ1の「PC1側の接点」のアドレス、すなわち、

M2

である。

d, eについて

ルータ1からルータ2へ送信するパケットであるから、あて先MACアドレスは、ルータ2の「ルータ1側の接点」のアドレス、すなわち、

M4 (空欄d)

である。また、送信元IPアドレスは、ルータ1を経由しても変化しないので、

IP1 (空欄e)

のままである。

fについて

図2と図3の比較からもわかるように、各ルータはパケットの、
送信元MACアドレス 及び あて先MACアドレス
を書き換えて中継を行う。

gについて

NAPTは、LAN側のプライベートIPアドレスを、インターネット上で用いることのできるグローバルIPアドレスに変換する機能である。このとき、LAN側の、
{送信元IPアドレス(プライベート), 送信元TCP/UDPポート番号}
と、
{送信元IPアドレス(グローバル), 新たに生成した送信元TCP/UDPポート番号}
の間で変換を行い、その対応を組にして覚えておくことにより、あて先であるWebサーバXからルータ2へ届いた返信パケットを、どのプライベートIPアドレスのPCへ転送したらよいか決定することができる。

問4

解答

設問1 ア, ウ, オ

設問2 a-エ b-ウ c-オ d-エ

解説

設問1

SELECT文のWHERE句に記述する条件は、

- ・各表の対応する行どうしを結合するための条件
- ・目的に応じて抽出する行を絞り込むための条件

に大別できる。以下、それぞれについて考察する。

・結合条件について

FROM句には

FROM 受講表, 研修実施表, 社員表

と、三つの表が指定されている。これらはそれぞれ

受講表 … 社員ごと, 研修ごとの受講履歴

研修実施表 … 研修ごとの情報

社員表 … 社員ごとの情報

となっているので、

受講表の各行に対して

- (i) 研修実施表から、対応する研修の行を結合する
- (ii) 社員表から、対応する社員の行を結合する

ような条件を記述すればよい。

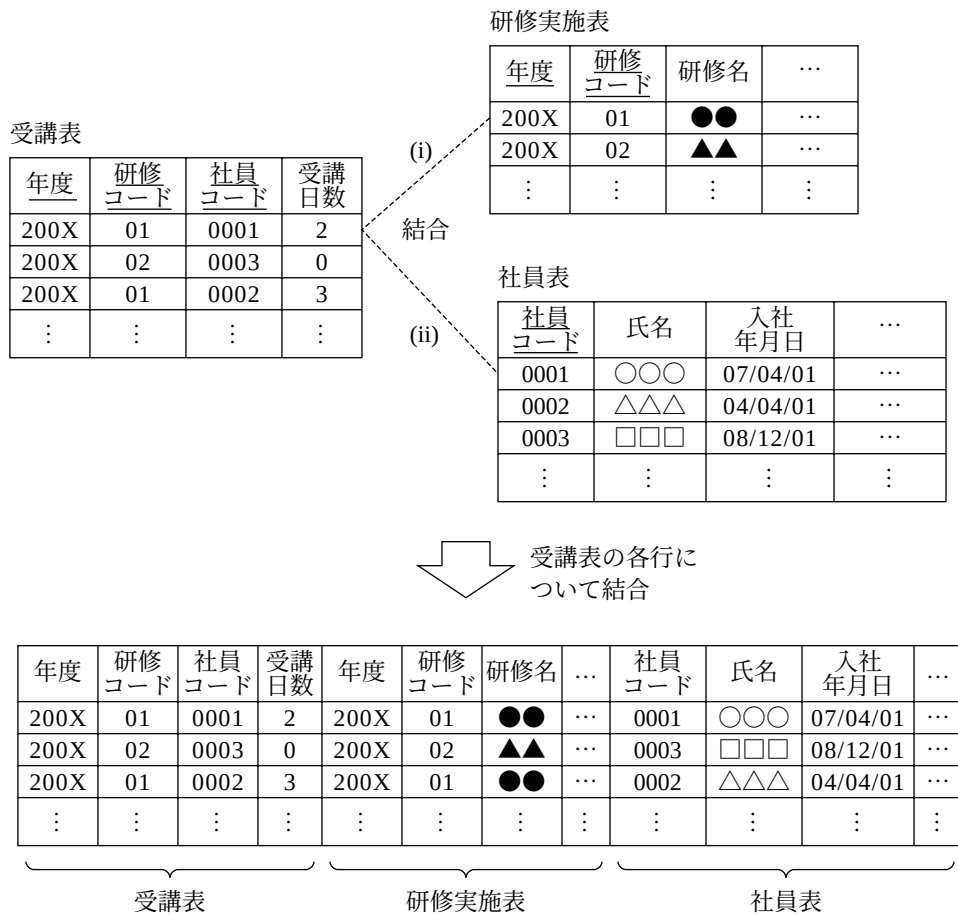


図1 結合のイメージ

(ii)に関する結合条件は、すでに

受講表.社員コード = 社員表.社員コード

として記されているので、あとは(i)に関する結合条件が必要である。受講表は、研修実施表の主キーである

年度, 研修コード

をもっているので、この組合せの内容が等しい行どうしを結合するため、

研修実施表.年度 = 受講表.年度

と

研修実施表.研修コード = 受講表.研修コード

を記述すればよい。

・絞込み条件について

“入社2年目の社員が、入社以来1日でも受講した研修”について知りたいのだから、
社員について…入社した年度が2007年度、かつ在職中
受講日数について…受講日数が0より大きい(1以上)
という条件が必要である。社員については、すでに
退職年月日 IS NULL AND
入社年月日 BETWEEN '2007-04-01' AND '2008-03-31'
として記されているので、あとは受講日数に関する条件として、
受講日数 > 0
を記述すればよい。

設問2

aについて

“部長”という文字列を含むすべての役職を検索するためには、LIKE述語を用いたパターンマッチングが必要である。

LIKE述語は、

列名 LIKE 'パターン'

という形式で指定し、値がパターンに合致する行が抽出される。このとき、パターン中に次のワイルドカード(特殊文字)を指定できる。

% …0文字以上の任意の文字列

_ (アンダバー) …任意の1文字

ここでは“部長”という文字がどの位置にあっても合致するようにしたいのだから、“部長”の前後に '%' を付け、

%部長%

をパターンとして指定すればよい。SELECT文全体としては、

SELECT 役職コード FROM 役職表 WHERE 役職名 LIKE '%部長%'

となる。

ア この指定では、役職名が“部長”の行しか抽出されない。

イ、ウ この指定では、役職名が“部長”、“副部長”、“部長代理”のいずれかである行しか抽出されない。設問文では、(“部長”、“副部長”、“部長代理”など)としており、「など」という表現で、他にも“部長”を含む役職名(“部長補佐”など)が存在することを示唆しているので、十分な指定とはいえない。

オ この指定では、役職名の先頭が“部長”である行しか抽出されず、“副部長”などが抽出対象から外れることになってしまう。

b～dについて

問題文の流れに沿い、各表の意義と「どの項目を含み、どの項目を含んでいないか」を考えていけば、自ずと解答が得られる。

設問1の考察でも述べたように、受講表は社員ごと、研修ごとの受講履歴を表すものであり、社員を識別するための項目として

社員コード (空欄b)

をもっている。しかし、社員に関する情報はこれだけであり、各社員についての他の情報(氏名など)は含まれていない。もちろん、各社員の所属部署を表す項目である

部署コード (空欄c)

も含まれていない。

部署コードの情報を得たいのであれば、社員コードを用いて

社員表 (空欄d)

を結合し、対応する行の中の部署コードを参照する必要がある。