

翔泳社 「情報処理教科書 データベーススペシャリスト 2025 年度」 対応版

データベーススペシャリスト試験

午前問題 **全 242 問** 完全版

一問題・解答・解説一

令和 6 年 10 月試験まで収録



2025.4.25

Ver8.00

Copyright by MSNET

memo

もくじ

第1章 SQL (76問)	005
第2章 概念データモデル (26問)	139
第3章 関係スキーマ (27問)	187
第4章 重要キーワード (113問)	229

本資料使用時の注意事項

- ※1. 本資料は、原則、翔泳社「情報処理教科書データベーススペシャリスト 2025 年版」の読者特典になります。読者の方、もしくは読者でなくとも個人が自身の試験対策でご利用される場合に限りご利用いただいても構いません。但し、解説部分の著作権は放棄していませんので、改変や複製して配布すること（研修や勉強会で使用するなど）は禁止です。あくまでも個人利用の範囲でご利用ください。
- ※2. 本資料は、(株)翔泳社の許可を得て編集したものですが、個人的に作成したものです。どのような責任も負いませんので、その点ご理解の上ご利用ください。
- ※3. 本資料は、同読者特典として翔泳社のサイトからダウンロードできる過去問題の解説資料をパソコン画面に表示させてハードコピーを取って編集したものです。したがって読みにくい部分、見にくい部分があります。その場合、必要に応じて翔泳社のサイトから、読者特典の過去の午前問題の解説をダウンロードして内容を確認してください。
- ※4. 個々の解説は、書籍化された時のものなので統一感がありませんし、参照先も当時の書籍になるので、2025 年版を指しているわけではありません。そのまま掲載しているのでご注意ください。
- ※5. 年度の古い解説は不十分なものが多く、分かりにくい部分は順次改訂していきますが、質問等には回答いたしませんので、それを十分理解した上でご利用ください。誤植等に関しても同様の扱いですが、誤植に関しては指摘してもらえれば嬉しいです。
- ※6. 何の断りもなく、誤植の改善や分かりにくい部分の改善を行いますので、定期的に版を確認ください。改訂した時には改訂履歴に記載するとともに、表紙の版を上げていきます。
- ※7. 解説上部に記載している「過去の出題回数」は、IPA から入手可能な平成 16 年以後の問題のみでカウントしています。そのため平成 15 年以前に出題されている問題もありますが、それは考慮していません。

【改訂履歴】

初版	V1.00	2017.10.30
第 2 版	V2.00	2019.01.08
第 3 版	V3.00	2019.10.10
第 4 版	V4.00 令和 2 年 10 月試験分まで収録	2021.08.25
第 5 版	V5.00 令和 3 年 10 月試験分まで収録	2022.04.06
第 6 版	V6.00 令和 4 年 10 月試験分まで収録	2023.04.23
第 7 版	V7.00 令和 5 年 10 月試験分まで収録	2024.04.01
第 8 版	V8.00 令和 6 年 10 月試験分まで収録	2025.04.24

1 SQL

第1章

この章では、DBMS を操作する SQL について説明する。SQL は試験に必ず出題されるため、十分に理解することが合格への絶対条件になりつつある。**毎回 SQL に関する問題は必ず出題されている。**しっかりと押さえておきたいテーマのひとつだと言えるだろう。しかし、SQL は“言語”である。そのすべてを短期間で習得することは困難であり、実務で SQL を利用していない人にとっては脅威でもある。そこで、実務経験者でない人でも効率よく学習できるように、過去に出題された問題を優先するとともにポイントだけを抜粋して構成した。最低限の範囲なので、十分に習得してもらいたい。

1.1 SELECT

令和3年度・DB・午前Ⅱ

問9 属性が n 個ある関係の異なる射影は幾つあるか。ここで、射影の個数には、元の関係と同じ結果となる射影、及び属性を全く含まない射影を含めるものとする。

- ア $\log_2 n$ イ n ウ $2n$ エ 2^n

平成25年度・DB・午前Ⅱ

問6 SQLのSELECT文の選択項目リストに関する記述として、適切なものはどれか。

- ア 指定できるのは表の列だけである。
イ 集約関数で指定する列は、GROUP BY句で指定した列でなければならない。
ウ 同一の列を異なる選択項目に指定できる。
エ 表の全ての列を指定するには、全ての列名をコンマで区切って指定しなければならない。

問9：正解（エ）

関係演算に関する問題。射影とは、関係 R を構成する属性のうち、指定した属性だけを取り出して新しい関係 R' を作成することである。たとえば、関係 R の属性を (a,b) の 2 個と仮定すると、「元の関係と同じ結果になる射影、及び属性を全く含まない射影を含める」という条件があるので、取り出せる射影の個数は、(), (a), (b), (a,b) の 4 つとなる。同様に (a,b,c) の 3 個と仮定すると、(), (a), (b), (c), (a,b), (a,c), (b,c), (a,b,c) の 8 つとなる。要するに属性が 1 増えるたびに 2 倍の個数になるので、正解はエになる。

問6：正解（ウ）

過去問題（平成20年・DB午前問25）
参照 「3.4.1 SELECT文の基本的な使い方」

SQL の SELECT 文の基本的な使い方にに関する問題。選択項目リストとは、SELECT と FROM の間に記述する項目のこと。それを踏まえて、選択肢を順番に見ていく。

ア：選択項目リストには、FROM 句で指定した表の列だけではなく、文字列定数や計算式、集約関数（集合関数）なども指定できる。よって誤りである。

イ：集約関数（集合関数）で指定する列は、GROUP BY 句で指定した列でなくとも良い。逆に数量や金額など、GROUP BY 句で指定しない列を指定するのが一般的。よって誤りである。

ウ：正解である。

エ：表のすべての列を指定する場合、すべての列をコンマで区切って指定する方法以外に、“*”も使える。よって誤りである。

問7 表Rと表Sに対し、SQL文を実行して結果を得るとき、aに入れる字句はどれか。

ここで、結果のNULLは値が存在しないことを表す。

R	
ID	名称1
1	AAA
2	BBB
3	CCC

S	
ID	名称2
2	bbb
3	ccc
4	ddd

(結果)

ID	名称1	名称2
1	AAA	NULL
2	BBB	bbb
3	CCC	ccc
4	NULL	ddd

[SQL文]

```
SELECT [ ] (R.ID, S.ID) AS ID, 名称1, 名称2
FROM R FULL JOIN S ON R.ID = S.ID
ORDER BY ID
```

ア COALESCE

イ DISTINCT

ウ NULLIF

エ UNIQUE

問7：正解（ア）

SQLに関する問題。COALESCE句について問われている。解答を得る考え方は次の通り。

SQL文で，“R”と“S”的結合に“FULL JOIN”を使っているので，“R”か“S”的いずれかに存在するすべての‘ID’が対象になる。この点は、問題文の〔結果〕の‘ID’列を見れば確認できる。

また、SELECT文の選択リストでは、“R”にしか存在しない‘ID’=1は、名称2がNULLになっており、同様に“S”にしか存在しない‘ID’=4も、名称1がNULLになっている。これを可能にするのは、可変長の引数を持ち、NULLでない最初の引数を返す関数COALESCEだということがわかる。したがって、正解はアになる。

問 8 “社員” 表から、部署コードごとの主任の人数と一般社員の人数を求める SQL 文
とするために、a に入る字句はどれか。ここで、実線の下線は主キーを表す。

社員（社員コード、部署コード、社員名、役職）

[SQL 文]

```
SELECT 部署コード,
    COUNT(CASE WHEN 役職 = '主任' a END) AS 主任の人数,
    COUNT(CASE WHEN 役職 = '一般社員' a END) AS 一般社員の人数
FROM 社員 GROUP BY 部署コード
```

[結果の例]

部署コード	主任の人数	一般社員の人数
AA01	2	5
AA02	1	3
BB01	0	1

ア THEN 1 ELSE -1

イ THEN 1 ELSE 0

ウ THEN 1 ELSE NULL

エ THEN NULL ELSE 1

問 9 SQL が提供する 3 値論理において、A に 5、B に 4、C に NULL を代入したとき、
次の論理式の評価結果はどれか。

(A > C) or (B > A) or (C = A)

ア ϕ (空)

イ false (偽)

ウ true (真)

エ unknown (不定)

問8：正解（ウ）

SQLの選択項目リスト内の CASEに関する問題。CASEをSELECT文の選択項目リスト内で使用すると、ある条件のものだけをカウントしたり加算したりできる。問題文のように“COUNT”を使用すると、条件式に合致している件数をカウントすることになる。その場合の構文は次のようになる。

```
COUNT (CASE WHEN 条件式 THEN 処理1 ELSE 処理2 END) AS 名称
```

(問題文にも書いているが) 問題文の“AS”の後を見ればどういうことをしたいのかがわかる。今回は“主任の人数”と“一般社員の人数”になる。

また“COUNT”を使っているので、条件式(役職 = ‘主任’、もしくは役職 = ‘一般社員’)に合致している場合だけの“件数”を合算するため、“THEN” (=条件に合致する場合) は“1”を設定してカウント対象とし、“ELSE” (=条件に合致しない場合) はカウント対象から外すために“NULL”を設定する。したがって正解はウになる。

選択肢エは真逆なので論外である。また、選択肢アや選択肢イのようにしてしまうと、カウントの対象になってしまふため注意が必要である。

問9：正解（エ）

3値論理に関する問題。一般的なプログラム言語の真理値が“真(true)”と“偽(false)”の二つの論理値であるのに対し、SQLが“不定(unknown)”という三つ目の論理値を持っているため、これを3値論理という。この不明というものが“NULL”である。

この3値論理では、計算式の値にNULLが入っていると、その結果は“不定(unknown)”になる。したがって正解はエになる。

unknown AND true → unknown

unknown AND false → false

unknown AND unknown → unknown

unknown OR true → true

unknown OR false → unknown

unknown OR unknown → unknown

$(A > C) \text{ or } (B > A) \text{ or } (C = A)$

$(5 > \text{NULL}) \text{ or } (4 > 5) \text{ or } (\text{NULL} = 5)$

unknown or false or unknown → unknown

問9 “成績” 表から、クラスごとに得点の高い順に個人を順位付けした結果を求める SQL 文の、a に入る字句はどれか。

成績

氏名	クラス	得点
情報太郎	A	80
情報次郎	A	63
情報花子	B	70
情報桜子	B	92
情報三郎	A	78

〔結果〕

氏名	クラス	得点	順位
情報太郎	A	80	1
情報三郎	A	78	2
情報次郎	A	63	3
情報桜子	B	92	1
情報花子	B	70	2

[SQL 文]

```
SELECT 氏名, クラス, 得点,
    a () OVER (PARTITION BY クラス ORDER BY 得点 DESC) 順位
FROM 成績
```

ア CUME_DIST

イ MAX

ウ PERCENT_RANK

エ RANK

問9：正解（工）

SQLに関する問題。ウインドウ関数について問われている。空欄aの後に「() OVER PARTITION BY クラス…」と続いている点と、問題文の「得点の高い順に個人を順位付けした結果を求める」としている点、選択肢などからウインドウ関数に関する問題だとわかるだろう。「得点の高い順」ということなので、空欄aには単純に順位付けを行う「RANK」関数が入る。したがって、正解は工になる。

なお、問題文の「クラスごと」は「PARTITION BY クラス」に、「得点の高い順に」は「ORDER BY 得点 DESC」が対応している。

問9 “成績”表から、**クラスごとに得点の高い順に個人を順位付けした結果を求める** SQL文の、aに入れる字句はどれか。

成績

氏名	クラス	得点
情報太郎	A	80
情報次郎	A	63
情報花子	B	70
情報桜子	B	92
情報三郎	A	78

氏名	クラス	得点	[結果]
			順位
情報太郎	A	80	1
情報三郎	A	78	2
情報次郎	A	63	3
情報桜子	B	92	1
情報花子	B	70	2

[SQL文]

```
SELECT 氏名, クラス, 得点,
    a () OVER (PARTITION BY クラス ORDER BY 得点 DESC) 順位
FROM 成績
```

ア：CUME_DISTは、累積分布値を求める関数になる。

イ：MAXは、クラスの中で最大値を求めるときに使う関数になる。

ウ：PERCENT_RANKは、RANKのように単純に順位付けをするのではなく、当該順位の相対値を求める時に使う。

問42 “学生”表に対し次の SELECT 文を実行した結果、導出される表はどれか。ここで、表中の“-”は、値がNULLであることを示す。

```
SELECT 学生番号, 氏名 FROM 学生
WHERE 住所 = '東京都' AND 自宅電話 IS NOT NULL
AND クラブ <> 'テニス'
```

学生

学生番号	氏名	生年月日	性別	自宅電話	住所	クラブ
S001	佐藤一郎	1986-05-15	男性	03-1111-1111	東京都	-
S002	鈴木花子	1988-01-10	女性	044-222-2222	神奈川県	テニス
S003	田中太郎	1986-11-05	男性	03-3333-3333	東京都	野球
S004	高橋次郎	1988-08-26	男性	-	千葉県	テニス
S005	渡辺一代	1986-09-14	女性	045-444-4444	神奈川県	-
S006	高橋恵子	1985-03-02	女性	-	東京都	水泳

ア

学生番号	氏名
S001	佐藤一郎
S003	田中太郎
S006	高橋恵子

イ

学生番号	氏名
S001	佐藤一郎
S003	田中太郎

ウ

学生番号	氏名
S003	田中太郎

エ

学生番号	氏名
S003	田中太郎
S006	高橋恵子

問 42：正解ウ

SQLのSELECT文に関する問題。学生表の中から、①住所が“東京都”で、②“自宅電話”がNULLではない（この表の場合“-”ではない）、③クラブが“テニス”以外（ただし、比較演算子の一方または両方がNULLの場合は、対象外になる）のものを抽出する。

参照 ➤ 「3.4 SELECT」

条件①に該当：S001, S003, S006

条件②に該当：S001, S002, S003, S005

条件③に該当：S003, S006

すべての条件を満たしているのは、S003のみである。よって正解はウになる。なお、条件③の“クラブ<>テニス”に、S001, S005を含まないよう注意が必要である。

問38 A 社では、社員教育の一環として全社員を対象に英会話研修を行っていたが、本年度（2005 年度）からは、4 月時点で入社 3 年を経過しているにもかかわらず初級システムアドミニストレータ（初級シスアド）試験に合格していない技術職種の社員に対して、英会話の代わりに初級シスアド研修を受講させることにした。本年度の英会話研修を受講させる社員の一覧を出力するための SQL 文はどれか。

なお、A 社では、社員はすべて 4 月 1 日入社であり、事業年度の始まりは 4 月 1 日である。また、ここで使用するデータベースには、2005 年 4 月 1 日時点でのデータが格納されているものとする。

- ア `SELECT 社員 FROM 社員テーブル
WHERE (入社年度 <= (2005 - 3) AND 職種 = '技術')
AND 初級シスアド合格 = 'No'`
- イ `SELECT 社員 FROM 社員テーブル
WHERE (入社年度 <= (2005 - 3) AND 職種 = '技術')
OR 初級シスアド合格 = 'Yes'`
- ウ `SELECT 社員 FROM 社員テーブル
WHERE NOT (入社年度 <= (2005 - 3) AND 職種 = '技術')
AND 初級シスアド合格 = 'No'`
- エ `SELECT 社員 FROM 社員テーブル
WHERE NOT (入社年度 <= (2005 - 3) AND 職種 = '技術')
OR 初級シスアド合格 = 'Yes'`

問38：正解（工）

問題文の説明がわかりにくいので整理すると、2005年度からは次のような。

＜従来＞ 全社員を対象に英語研修を実施

＜2005年度～＞

初級シニアド研修の受講対象者 = 下記の①～③をすべて満たす社員

① 4月時点で入社3年を経過している（4年目以後）

② 技術職

③ 初級シニアドに合格していない

英会話の受講対象者 = 初級シニアド研修の受講対象者以外

ア：これだと（①AND②）AND③になるので、初級シニアド受講対象者になる。誤り。

イ：これだと③に合致する社員は英会話の受講対象者になるが、それに加えて、①AND②の社員も対象に含まれるので、初級シニアドを受講しなければならない受講生も英会話の受講対象に含まれてしまう。誤り。

ウ：（①AND②）以外の社員（すなわち、技術職以外か、入社3年目以下の社員）で、かつ③（=初級シニアドを持っていない社員）なので、例えば技術職ではない社員や入社3年目以下の社員でも初級シニアド試験に合格している社員がいたら、（本来は英会話研修の受講対象者）でないといけないので、対象から外れてしまう。誤り。

エ：（①AND②）以外の社員（すなわち、技術職以外か、入社3年目以下の社員）か、初級シニアド試験に合格しているか、どちらかの条件を満たしている社員が受講対象者になる。これは、問題文の英会話研修の受講対象要件に合致する。これが正解。

問 9 “社員”表と“人事異動”表から社員ごとの勤務成績の平均を求める適切な SQL 文はどれか。ここで、求める項目は、社員コード、社員名、勤務成績（平均）の 3 項目とする。

社員

社員コード	社員名	性別	生年月日	入社年月日
O1553	太田 由美	女	1970-03-10	1990-04-01
S3781	佐藤 義男	男	1943-11-20	1975-06-11
O8665	太田 由美	女	1978-10-13	1999-04-01

人事異動

社員コード	配属部門	配属年月日	担当勤務内容	勤務成績
O1553	総務部	1990-04-01	広報（社内報）	69.0
O1553	営業部	1998-07-01	顧客管理	72.0
S3781	資材部	1975-06-11	仕入在庫管理	70.0
S3781	経理部	1984-07-01	資金計画	81.0
S3781	企画部	1993-07-01	会社組織、分掌	95.0
O8665	秘書室	1999-04-01	受付	70.0

ア SELECT 社員.社員コード, 社員名, AVG(勤務成績) AS "勤務成績(平均)"

FROM 社員, 人事異動

WHERE 社員.社員コード = 人事異動.社員コード

GROUP BY 勤務成績

イ SELECT 社員.社員コード, 社員名, AVG(勤務成績) AS "勤務成績(平均)"

FROM 社員, 人事異動

WHERE 社員.社員コード = 人事異動.社員コード

GROUP BY 社員.社員コード, 社員.社員名

ウ SELECT 社員.社員コード, 社員名, AVG(勤務成績)/COUNT(勤務成績)

AS "勤務成績(平均)"

FROM 社員, 人事異動

WHERE 社員.社員コード = 人事異動.社員コード

GROUP BY 社員.社員コード, 社員.社員名

エ SELECT 社員.社員コード, 社員名, MAX(勤務成績)/COUNT(*)

AS "勤務成績(平均)"

FROM 社員, 人事異動

WHERE 社員.社員コード = 人事異動.社員コード

GROUP BY 社員.社員コード, 社員.社員名

問9：正解イ

SQL文の解釈に関する問題。問われているのは、“GROUP BY (参照 ➤ 3.4.4 GROUP BY句と集合関数)”と複数表の問い合わせ、すなわち“結合 (参照 ➤ 3.4.6 結合)”である。題意に沿ってSQL文を書いてみて、その結果と選択肢をつき合わせてチェックする。

- (1) 問題文の“求める項目”と選択肢のSELECT文以下の記述でチェック
 - ・選択肢アとイは正しい。
 - ・選択肢ウは、'AVG ()' の使い方を間違っている。割り算をしなくても、'AVG (属性名)' で平均値を求めることができる。よって誤りである。
 - ・選択肢エの'MAX ()' は、最大値を求める関数。よって誤りである。
- (2) FROM句とWHERE句
 - ・すべて正しい
- (3) GROUP BY句
 - ・選択肢アは誤り。列名に‘社員コード’や‘社員名’を指定する場合、GROUP BY以降でもそれを指定しなければならない。
 - ・選択肢イ～エはいずれも正しい

以上より、イが正しいSQL文になる。

問10 ある電子商取引サイトでは、会員の属性を柔軟に変更できるように、“会員項目”表で管理することにした。“会員項目”表に対し、次の条件でSQL文を実行して結果を得る場合、SQL文のaに入れる字句はどれか。ここで、実線の下線は主キーを、NULLは値がないことを表す。

〔条件〕

- (1) 同一“会員番号”をもつ複数の行によって、1人の会員の属性を表す。
- (2) 新規に追加する行の行番号は、最後に追加された行の行番号に1を加えた値とする。
- (3) 同一“会員番号”で同一“項目名”的行が複数ある場合、より大きい行番号の項目値を採用する。

会員項目

行番号	会員番号	項目名	項目値
1	0111	会員名	情報太郎
2	0111	最終購入年月日	2021-02-05
3	0112	会員名	情報花子
4	0112	最終購入年月日	2021-01-30
5	0112	最終購入年月日	2021-02-01
6	0113	会員名	情報次郎

〔結果〕

会員番号	会員名	最終購入年月日
0111	情報太郎	2021-02-05
0112	情報花子	2021-02-01
0113	情報次郎	NULL

〔SQL文〕

```

SELECT 会員番号,
    [a] (CASE WHEN 項目名='会員名' THEN 項目値 END) AS 会員名,
    [a] (CASE WHEN 項目名='最終購入年月日' THEN 項目値 END)
        AS 最終購入年月日
FROM ( SELECT 会員番号, 項目名, 項目値 FROM 会員項目
        WHERE 行番号 IN ( SELECT [a] (行番号) FROM 会員項目
                            GROUP BY 会員番号, 項目名 )
    ) T
        GROUP BY 会員番号
        ORDER BY 会員番号

```

ア COUNT イ DISTINCT ウ MAX エ MIN

問 10：正解（ウ）

SQL の SELECT 文に関する問題。集約関数について問われている。

まず、FROM 以後の（ ）内の SELECT 文から解析する。そこには、その SELECT 文の副問合せの IN（ ）内にも SELECT 文があり、その SELECT 文に空欄 a がある。その空欄 a は“行番号”に対するものなので、ここが問題文の「より大きい行番号の項目値を採用する。」という箇所になると判断できる。したがってここに MAX が入る。正解はウ。

ちなみに、FROM 以後の（ ）内の SELECT 文を実行すると、問題文の「より大きい行番号の項目値を採用する。」という点はクリアできる。問題文に例示されている“会員項目”テーブルだと、4 行目だけが無くなったテーブルになる。そのため一見すると、残りの 2 つの空欄 a に MAX は不要のように思われるが、これは GROUP BY を指定しているからである。GROUP BY を使う場合、SELECT 文の後に記述する選択項目リストには、GROUP BY の後に指定したものか集約関数、定数などに限定されるため、集約関数を使わないと選択項目リストに含めることができないからである。値がない場合に NULL を表示させるためにもここで MAX が必要になる。

なお、解答に当たっては、選択肢を一つずつ代入してみて判断してみてもいいし、問題文の「より大きい行番号の項目値を採用する。」という記述から“MAX”を代入してみて問題ないかで判断してもいい。

問7 “商品”表と“商品別売上実績”表に対して、SQL文を実行して得られる売上平均金額はどれか。

商品

商品コード	商品名	商品ランク
S001	PPP	A
S002	QQQ	A
S003	RRR	A
S004	SSS	B
S005	TTT	C
S006	UUU	C

商品別売上実績

商品コード	売上合計金額
S001	50
S003	250
S004	350
S006	450

[SQL文]

```
SELECT AVG(売上合計金額) AS 売上平均金額
FROM 商品 LEFT OUTER JOIN 商品別売上実績
ON 商品.商品コード = 商品別売上実績.商品コード
WHERE 商品ランク = 'A'
```

ア 100

イ 150

ウ 225

エ 275

問7：正解（イ）

SQLに関する問題。AVG関数を使って平均を求める場合にNULL値をどう扱うかという点が問われている。問題文のSQL文では，“商品”表と“商品別売上実績”表を左外部結合し、‘商品ランク’が‘A’のものだけを対象に平均値を求めている。ちょうど下図の赤枠の部分になる。

商品.商品コード	商品名	商品ランク	商品別売上実績.商品コード	売上合計金額	} 平均を 求める
S001	PPP	A	S001	50	
S002	QQQ	A	NULL	NULL	
S003	RRR	A	S003	250	
S004	SSS	B	S004	350	
S005	TTT	C	NULL	NULL	
S006	UUU	C	S006	450	

この時、‘商品コード’が‘S002’の商品は“商品別売上実績”表には値がないので、左外部結合をすると‘売上合計金額’がNULLになる。AVG関数で平均値を求める時、対象がNULLの場合は次のように無視して計算する（数に含めない）。

$$(50 + 250) / 2 = 150$$

したがって正解はイになる。

問 6 次の SQL 文によって “会員” 表から新たに得られる表はどれか。

[SQL 文]

```
SELECT AVG(年齢)
  FROM 会員
 GROUP BY グループ
 HAVING COUNT(*) > 1
```

会員

会員番号	年齢	グループ
001	20	B
002	30	C
003	60	A
004	40	C
005	40	B
006	50	C

ア

AVG (年齢)
36

イ

AVG (年齢)
40

ウ

AVG (年齢)
30
40

エ

AVG (年齢)
60
30
40

問6：正解ウ

過去問題（平成14年・DB午前問28など）

参照▶「3.4.4 GROUP BY句と集合関数」

SQL（第3章）－GROUP BY句及び集合関数に関する問題。特に、HAVINGに関する知識が問われている。HAVINGをGROUP BY句の中で使用すると、グループ化の対象とする条件を指定することができる。今回のように“HAVING COUNT(*) > 1”的場合、「グループ内の件数が1より多いもの（2以上のもの）」だけを対象に平均値を求めるという意味になる。

グループの件数は、A（1件）、B（2件）、C（3件）なので、Aは対象外となって平均を求めない。したがって、結果はBの平均値、Cの平均値の2件になり、そこからだけでも正解は選択肢ウだとわかる（SELECT AVG(年齢) FROM…なので、抽出対象は平均値オンリーになる）。

念のため、それぞれを計算すると次のようになる。

A…1件しかないので「GROUP BY グループ HAVING COUNT(*) > 1」より、対象外

B… (001番の“20” + 005番の“40”) ÷ 2 = 30

C… (002番の“30” + 004番の“40” + 006番の“50”) ÷ 3 = 40

問35 “部品” 表に対し次の SELECT 文を実行したときの結果として、正しいものはど
か。

```
SELECT 部品区分, COUNT(*) AS 部品数, MAX(単価) AS 単価
FROM 部品 GROUP BY 部品区分 HAVING SUM(在庫量) > 200
```

部品

部品番号	部品区分	単価	在庫量
001	P1	1,500	90
002	P2	900	30
003	P2	950	90
004	P3	2,000	50
005	P1	2,000	100
006	P3	2,500	60
007	P1	1,500	50
008	P2	900	80
009	P3	1,000	40
010	P4	900	80
011	P3	1,500	70
012	P4	950	100

ア

部品区分	部品数	単価
P1	3	2,000
P2	3	1,000

イ

部品区分	部品数	単価
P1	3	2,000
P3	4	2,500

ウ

部品区分	部品数	単価
P2	3	1,000
P4	2	950

エ

部品区分	部品数	単価
P1	3	2,000
P2	3	1,000
P3	4	2,500

問 35：正解イ

問のSQL文では、部品区分でグループ化した部品の在庫量の合計が200を超える部品区分を部品数、単価とともに検索している。したがって、正解はイとなる。[\[参照\]「3.3 SELECT」](#)

問 5 “社員”表から、役割名がプログラマである社員が 3 人以上所属している部門の部門名を取得する SQL 文はどれか。ここで、実線の下線は主キーを表す。

社員 (社員番号, 部門名, 社員名, 役割名)

ア SELECT 部門名 FROM 社員
GROUP BY 部門名
HAVING COUNT(*) >= 3
WHERE 役割名 = 'プログラマ'

イ SELECT 部門名 FROM 社員
WHERE COUNT(*) >= 3 AND 役割名 = 'プログラマ'
GROUP BY 部門名

ウ SELECT 部門名 FROM 社員
WHERE COUNT (*) >= 3
GROUP BY 部門名
HAVING 役割名 = 'プログラマ'

エ SELECT 部門名 FROM 社員
WHERE 役割名 = 'プログラマ'
GROUP BY 部門名
HAVING COUNT(*) >= 3

問5：正解（エ）

NEW (但し、類似問題多数)

参照 「3.4.4 GROUP BY句と集合関数」

SQL – SELECT文 – GROUP BYに関する問題。HAVINGの正しい使い方が問われている。

解答に当たっては、問題文に書かれている通りにSQL文を作成していくべきだが、そうしなくとも、選択肢にある“差”に着眼して解いても構わない。

選択肢はすべて“GROUP BY”を使用している。これは問題文の「3人…部門の部門名を取得する」という記述に対するもので、“GROUP BY 部門名”で部門名単位にグルーピングしなければならないからだ。選択肢にはすべて“GROUP BY 部門名”が入っているので、そこは問題ない。後は、問題文の「3人以上所属している部門の部門名を取得する」を HAVINGを使って表現すればいい。HAVINGの使い方は次のようになる。これで「部門ごとにグルーピングして、その結果、合計数が“3”以上の部門名だけ」という抽出が完成する。

GROUP BY 部門名

HAVING COUNT(*) >= 3

但し今回は、「プログラマである社員が3人以上」なので、全社員のうちプログラマだけが対象になる。部門に何人いようが、対象はプログラマだけだ。それはグルーピングとは無関係で、全社員からプログラマだけに絞り込むので、次のような表記になる。

SELECT …

WHERE 役割名 = ‘プログラマ’

GROUP BY 部門名

HAVING COUNT(*) >= 3

以上より、選択肢（エ）が正解になる。

問 7 過去 3 年分の記録を保存している“試験結果”表から、2014 年度の平均点数が 600 点以上となったクラスのクラス名と平均点数の一覧を取得する SQL 文はどれか。ここで、実線の下線は主キーを表す。

試験結果 (学生番号, 受験年月日, 点数, クラス名)

- ア SELECT クラス名, AVG(点数) FROM 試験結果
GROUP BY クラス名 HAVING AVG(点数) >= 600
- イ SELECT クラス名, AVG(点数) FROM 試験結果
WHERE 受験年月日 BETWEEN '2014-04-01' AND '2015-03-31'
GROUP BY クラス名 HAVING AVG(点数) >= 600
- ウ SELECT クラス名, AVG(点数) FROM 試験結果
WHERE 受験年月日 BETWEEN '2014-04-01' AND '2015-03-31'
GROUP BY クラス名 HAVING 点数 >= 600
- エ SELECT クラス名, AVG(点数) FROM 試験結果
WHERE 点数 >= 600
GROUP BY クラス名
HAVING (MAX(受験年月日)
BETWEEN '2014-04-01' AND '2015-03-31')

問7：正解（イ）

SQLに関する問題（参照▶第3章）。GROUP BY句及び集合関数、HAVINGについて問われている。HAVINGをGROUP BY句の中で使用すると、グループごとに抽出対象とする条件を指定することができる。

今回の場合、問題文には「平均点数が600点以上となったクラスの一覧を取得する」という条件が付いているので、HAVINGを使って {HAVING AVG(点数) >= 600}と指定する。これは選択肢ア、イになる。

選択肢アとイの違いは、{BETWEEN '2014-04-01' AND '2015-03-31'}があるかないかの差。問題文には「2014年度の」と指定があるため、正解はイになる。

問9 “商品”表と“納品”表を商品番号で等結合した結果はどれか。

商品

商品番号	商品名	価格
S01	ボールペン	150
S02	消しゴム	80
S03	クリップ	200

納品

商品番号	顧客番号	納品数
S01	C01	10
S01	C02	30
S02	C02	20
S02	C03	40
S03	C03	60

ア

商品番号	商品名	価格	顧客番号	納品数
S01	ボールペン	150	C01	10
S02	消しゴム	80	C02	20
S03	クリップ	200	C03	60

イ

商品番号	商品名	価格	商品番号	顧客番号	納品数
S01	ボールペン	150	S01	C01	10
S02	消しゴム	80	S02	C02	20
S03	クリップ	200	S03	C03	60

ウ

商品番号	商品名	価格	顧客番号	納品数
S01	ボールペン	150	C01	10
S01	ボールペン	150	C02	30
S02	消しゴム	80	C02	20
S02	消しゴム	80	C03	40
S03	クリップ	200	C03	60

エ

商品番号	商品名	価格	商品番号	顧客番号	納品数
S01	ボールペン	150	S01	C01	10
S01	ボールペン	150	S01	C02	30
S02	消しゴム	80	S02	C02	20
S02	消しゴム	80	S02	C03	40
S03	クリップ	200	S03	C03	60

問 9：正解（エ）

関係代数の演算に関する問題。等結合について問われている。商品番号で等結合した場合、① 2つの表に共通する商品番号のデータは全て保持され、②結合に使った“商品番号”も（両表のものが）重複したまま表示される。この問題のケースでは、“納品”的 5つのデータは全て残され、（特に指定がないので）列（属性）も全て表示され、さらに、商品番号は 2つ存在する結果表（行数 = 5、列数 = 6）になる。それを表している表は（エ）。これが正解。

問13 関係“履修”と関係“担当”を自然結合した結果はどれか。

履修

学生	科目
山田太郎	情報処理
山田太郎	代数
加藤花子	情報処理

担当

科目	教官
情報処理	鈴木一郎
代数	斎藤正樹

ア

学生	科目	教官
山田太郎	情報処理	鈴木一郎
山田太郎	代数	斎藤正樹
加藤花子	情報処理	鈴木一郎

イ

履修.学生	履修.科目	担当.科目	担当.教官
山田太郎	情報処理	情報処理	鈴木一郎
山田太郎	代数	代数	斎藤正樹
加藤花子	情報処理	情報処理	鈴木一郎

ウ

履修.学生	履修.科目	担当.科目	担当.教官
山田太郎	情報処理	代数	斎藤正樹
山田太郎	代数	情報処理	鈴木一郎
加藤花子	情報処理	代数	斎藤正樹

工

履修.学生	履修.科目	担当.科目	担当.教官
山田太郎	情報処理	情報処理	鈴木一郎
山田太郎	情報処理	代数	斎藤正樹
山田太郎	代数	情報処理	鈴木一郎
山田太郎	代数	代数	斎藤正樹
加藤花子	情報処理	情報処理	鈴木一郎
加藤花子	情報処理	代数	斎藤正樹

問 13：正解ア

関係代数の演算に関する問題（参照 第4章-4 関係代数）。自然結合とは、二つの表の等結合から重複した列を取り除いた表のことである。直積と選択（等結合）に射影をかけて実現する。この問題の表“履修”と“担当”には、共通の属性“科目”があるので、それで結合し、重複しないようにどちらか一方の“科目”だけ残す。その条件に合致する表は、選択肢アしかない。

イ：これは等結合の結果である。誤り。

ウ：直積から等結合を排除した結果。誤り。

エ：直積の結果。誤り。

問30 化粧品の製造を行っている A 社では、販売代理店を通じて商品販売を行っている。

今後の販売戦略に活用するために、次の三つの表からなるデータベースの作成を計画している。これらのデータを用いるだけでは得ることのできない情報はどれか。

顧客データ

顧客	氏名	性別	生年月日
----	----	----	------

販売代理店の日別販売データ

販売代理店	日付	商品	販売数量
-------	----	----	------

顧客の商品購入データ

顧客	販売代理店	商品	販売数量
----	-------	----	------

- ア 商品ごとの販売数量の日別差異
- イ 性別ごとの売れ筋商品
- ウ 販売代理店ごとの購入者数の日別差異
- エ 販売代理店ごとの購入者の年齢分布

問 30：正解ウ

3つの表から得られない情報について問われている。解答に当たっては、選択肢の情報が得られるかどうかを順番に見ていく。

- ア：「販売代理店の日別販売データ」を使って、商品別に集計すれば得られる
- イ：「顧客の商品購入データ」と「顧客データ」を“顧客”属性で結合し、“性別”, “商品”, 属性ごとの“販売数量”を求め、販売数量で降順にソートすれば得られる
- ウ：「販売代理店の日別販売データ」では顧客単位に集計していないので、販売代理店ごとの購入者数の日別差異は得られない。また、「顧客の商品購入データ」には日付を持っていないのでこちらも不可能である。最後に、この2つの表の間には関連性もないため、この情報は得ることができない。これが正解
- エ：「顧客の商品購入データ」と「顧客データ」を“顧客”属性で結合し、“販売代理店”, “顧客”, “生年月日”属性を使って分布を求めるべき

問8 “部品”表から、部品名に ‘N11’ が含まれる部品情報（部品番号、部品名）を検索する SQL 文がある。この SQL 文は、検索対象の部品情報のほか、対象部品に親部品番号が設定されている場合は親部品情報を返し、設定されていない場合は NULL を返す。a に入る字句はどれか。ここで、実線の下線は主キーを表す。

部品（部品番号、部品名、親部品番号）

[SQL文]

```
SELECT B1.部品番号, B1.部品名,  
       B2.部品番号 AS 親部品番号, B2.部品名 AS 親部品名  
  FROM 部品 a  
 WHERE B1.親部品番号 = B2.部品番号  
   AND B1.部品名 LIKE '%N11%'
```

- ア B1 JOIN 部品 B2
- イ B1 LEFT OUTER JOIN 部品 B2
- ウ B1 RIGHT OUTER JOIN 部品 B2
- エ B2 LEFT OUTER JOIN 部品 B1

問8：正解（イ）

SQL文に関する問題。JOINを使った結合に関して問われている。問題文に書かれている条件を整理すると、次のようになる。

- ①“部品”表から、部品名に‘N11’が含まれる部品情報（部品番号、部品名）を検索する。
- ②対象部品に親部品番号が設定されている場合は親部品情報を返す。
- ③対象部品に親部品番号が設定されていない場合はNULLを返す。

上記の②の親部品情報とは、問題文のSQL文を確認すると、親部品として自己参照されている方の“部品”表（B2）の部品番号と部品名になる。

また、上記の②と③より、①の検索結果は全て抽出対象になることがわかる。それは“部品”表（B1）になるので、B1が全て対象となる外部結合方式を選択肢より探せばいい。結果、選択肢イ（B1の左外部結合）が正解になる。選択肢アは内部結合、選択肢ウ、エはB2を全て対象とする外部結合になるため、いずれも不適切である。

問32 “商品”表と“売上明細”表に対して、次の SQL 文を実行した結果の表として、正しいものはどれか。ここで、結果の表中の“－”は、値がナルであることを示す。

```
SELECT X.商品番号, 商品名, 数量
  FROM 商品 X LEFT OUTER JOIN 売上明細 Y
    ON X.商品番号 = Y.商品番号
```

商品	
商品番号	商品名
S101	A
S102	B
S103	C
S104	D

売上明細				
売上番号	売上日	商品番号	数量	売上金額
U001	2006-02-10	S101	5	7,500
U002	2006-02-26	S104	2	4,000
U002	2006-02-26	S101	10	15,000
U003	2006-03-05	S103	5	5,000
U003	2006-03-05	S104	8	16,000

ア

商品番号	商品名	数量
S101	A	5
S101	A	10
S102	B	－
S103	C	5
S104	D	2
S104	D	8

イ

商品番号	商品名	数量
S101	A	5
S101	A	10
S103	C	5
S104	D	2
S104	D	8

ウ

商品番号	商品名	数量
S101	A	15
S102	B	－
S103	C	5
S104	D	10

エ

商品番号	商品名	数量
S101	A	15
S103	C	5
S104	D	10

問32：正解ア（平成16年度の問32とほぼ同じ問題）

このSELECT文は左外部結合（LEFT OUTER JOIN）なので、左側に記述された“商品テーブル（別名X）”に存在するものは、右側に記述された“売上明細テーブル（別名Y）”に存在しなくてもすべて作られる。その場合、表示項目のうち、売上明細テーブルからの属性（今回は“数值”）は空値(NULL)になる。また、売上明細テーブルに複数存在する場合はその数だけ作られる。よって、正解はアとなる。

参照 ➔ 「3.3.6 結合」

問8 “社員取得資格”表に対し、SQL文を実行して結果を得た。SQL文のaに入れる字句はどれか。

社員取得資格	
社員コード	資格
S001	FE
S001	AP
S001	DB
S002	FE
S002	SM
S003	FE
S004	AP
S005	NULL

〔結果〕		
社員コード	資格1	資格2
S001	FE	AP
S002	FE	NULL
S003	FE	NULL

[SQL文]

```
SELECT C1.社員コード, C1.資格 AS 資格1, C2.資格 AS 資格2
FROM 社員取得資格 C1 LEFT OUTER JOIN 社員取得資格 C2
```

a

ア ON C1.社員コード = C2.社員コード
 AND C1.資格 = 'FE' AND C2.資格 = 'AP'
 WHERE C1.資格 = 'FE'

イ ON C1.社員コード = C2.社員コード
 AND C1.資格 = 'FE' AND C2.資格 = 'AP'
 WHERE C1.資格 IS NOT NULL

ウ ON C1.社員コード = C2.社員コード
 AND C1.資格 = 'FE' AND C2.資格 = 'AP'
 WHERE C2.資格 = 'AP'

エ ON C1.社員コード = C2.社員コード
 WHERE C1.資格 = 'FE' AND C2.資格 = 'AP'

問8：正解（ア）

SQLに関する問題。外結合、自己結合について問われている。この手の問題は、まず元の表と【結果】を見て、何がしたいのか？を推測する。

社員取得資格	
社員コード	資格
S001	FE
S001	AP
S001	DB
S002	FE
S002	SM
S003	FE
S004	AP
S005	NULL

〔結果〕		
社員コード	資格 1	資格 2
S001	FE	AP
S002	FE	NULL
S003	FE	NULL

資格1が、全てFEになっていることから、「FEの資格を持っている人」がひとつの抽出条件になっていると考えられる。社員コード'S004'の人が結果ないので、ほぼ間違いない。

資格2は、「二つの資格を表示させる」とも考えられるが、それだと社員コード'S002'の人のSMが表示されていないといけないので違う。「S002」の人がNULLになっていることから、資格2は「APの資格保有者は、ここにAPと表示する」と考えるのが妥当だろう。後は、SQL文を見ながら考える。

次に、これらの仮説を元にSQL文をチェックする。

問題文のSQL文に、全ての選択肢で共通の“ON C1. 社員コード = C2. 社員コード”まで考えてみる。ここまで結果なら、図のようになる。

```
SELECT C1. 社員コード, C1. 資格 AS 資格1, C2. 資格 AS 資格2
FROM 社員取得資格 C1 LEFT OUTER JOIN 社員取得資格 C2
ON C1. 社員コード = C2. 社員コード
```

The diagram illustrates the result of the SQL query. It shows two input tables, '社員取得資格 C1' and '社員取得資格 C2', and their join result. A dashed arrow connects the input tables to the result table. An arrow labeled 'ここまでなら' points to the result table.

社員コード	資格
S001	FE
S001	AP
S001	DB
S002	FE
S002	SM
S003	FE
S004	AP
S005	NULL

社員コード	資格
S001	FE
S001	AP
S001	DB
S002	FE
S002	SM
S003	FE
S004	AP
S005	NULL

社員コード	資格 1	資格 2
S001	FE	FE
S001	FE	AP
S001	FE	DB
S001	AP	FE
S001	AP	AP
S001	AP	DB
S001	DB	FE
S001	DB	AP
S002	FE	FE
S002	FE	SM
S002	SM	FE
S003	FE	FE
S004	AP	AP
S005	NULL	NULL

これをベースに、難しければ選択肢の違いを見ながら考察を進めていけばいいだろう。

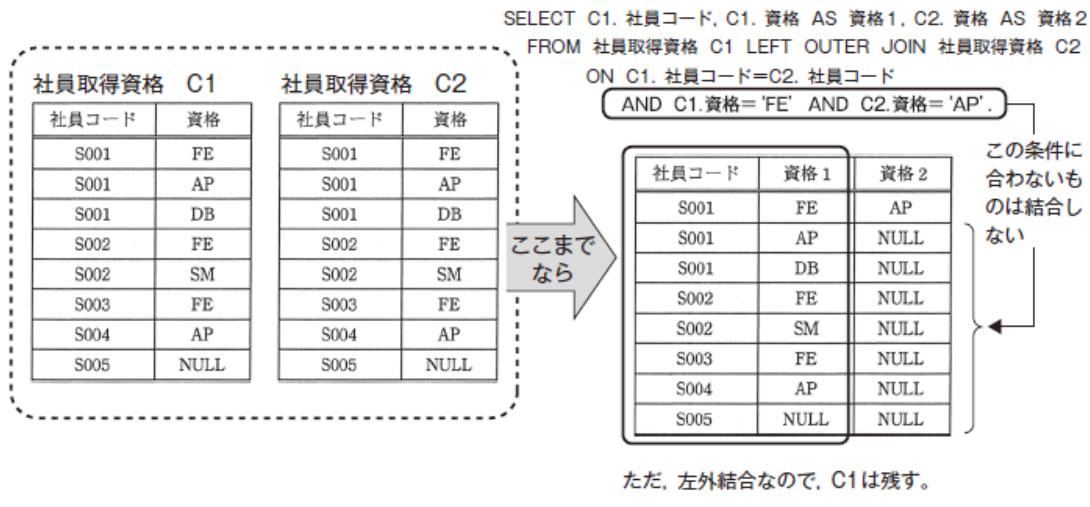
- (1) ON 以下の結合条件に “AND C1. 資格 = ‘FE’ AND C2. 資格 = ‘AP’” という制限条件が必要かどうかを考える。

選択肢エのよう、この結合時の制限条件がなく、SELECT 文の WHERE 句に “WHERE C1. 資格 = ‘FE’ AND C2. 資格 = ‘AP’” だけにすると、次のような結果になる。したがって、これは誤り。

社員コード	資格 1	資格 2
S001	FE	AP

- (2) ON 以下の結合条件に “AND C1. 資格 = ‘FE’ AND C2. 資格 = ‘AP’” という制限条件を付けた場合にどうなるのか。そこまで考えてみる。

ON 以下の結合条件に、結合目的以外の制限条件を指定しても、結局は、それらの全部の条件を評価して、その結果が “真” なら結合し、“偽” なら結合しないというだけ。したがって、“AND C1. 資格 = ‘FE’ AND C2. 資格 = ‘AP’” の条件を満たす場合にだけ結合対象になる。今回の場合だと、下図の 1 行目だけだ。しかし、左外結合なので左側のテーブルはそのまま残すことになる（下図参照）。



- (3) 最後に、結合後のテーブルに対する SELECT 文の制限条件を評価する。

上記の図のうち、問題文の【結果】を得るには、結合後に “C1. 資格 = ‘FE’” で絞り込まないといけない。それを実施しているのは選択肢アになる。

正解は (ア) になる。

memo

問10 表 A と表 B から、どちらか一方にだけ含まれる ID を得る SQL 文の a に入る字句はどれか。

A	B
ID	ID
100	200
200	400
300	600
400	800

[SQL 文]

```
SELECT COALESCE(A.ID, B.ID)
  FROM A [ ] B ON A.ID = B.ID
 WHERE A.ID IS NULL OR B.ID IS NULL
```

ア FULL OUTER JOIN

イ INNER JOIN

ウ LEFT OUTER JOIN

エ RIGHT OUTER JOIN

問 10：正解（ア）

SQLに関する問題。結合について問われている。問題文の「どちらか一方にだけ含まれる ID」とは、この問題の場合だと①Aにだけ含まれるか、②Bにだけ含まれるかのものを指し、AやBに存在していても両方にあるものは対象外になるIDをいう。具体的には、Aにだけしかない {100, 300} と Bにだけしかない {600, 800} の4つだけを抽出するSQLになる。

SQL文の「COALESCE (A.ID, B.ID)」は、A.IDに値がある場合にはA.IDを、A.IDにはなくB.IDにある場合はB.IDを示すので、後は、両方にあるものを除外して結合させさせればいいことがわかる。「WHERE」以下の条件式で、A.IDかB.IDのいずれか一方がNULLの場合に限定していることを考慮して、どの結合なのかを選択肢から選べばいい。COALESCEを除き、結合とWHERE句の条件だけで考えれば次のようになる（選択肢は抽出結果が無いので割愛している）。

ア

A.ID	B.ID
100	NULL
300	NULL
NULL	600
NULL	800

ウ

A.ID	B.ID
100	NULL
300	NULL

エ

A.ID	B.ID
NULL	600
NULL	800

以上より、正解はアになる。

問6 “文書”表, “社員”表から結果を得るSQL文のaに入れる字句はどれか。

文書

文書 ID	作成者 ID	承認者 ID
1	100	200
2	100	300
3	200	400
4	500	400

社員

社員 ID	氏名
100	山田太郎
200	山本花子
300	川上一郎
400	渡辺良子

[結果]

文書 ID	作成者 ID	作成者氏名	承認者 ID	承認者氏名
1	100	山田太郎	200	山本花子
2	100	山田太郎	300	川上一郎
3	200	山本花子	400	渡辺良子
4	500	NULL	400	渡辺良子

[SQL文]

```
SELECT 文書 ID, 作成者 ID, A. 氏名 AS 作成者氏名,
       承認者 ID, B. 氏名 AS 承認者氏名 FROM a
```

ア 文書 LEFT OUTER JOIN 社員 A ON 文書.作成者 ID = A. 社員 ID
 LEFT OUTER JOIN 社員 B ON 文書.承認者 ID = B. 社員 ID

イ 文書 RIGHT OUTER JOIN 社員 A ON 文書.作成者 ID = A. 社員 ID
 RIGHT OUTER JOIN 社員 B ON 文書.承認者 ID = B. 社員 ID

ウ 文書, 社員 A, 社員 B

LEFT OUTER JOIN 社員 A ON 文書.作成者 ID = A. 社員 ID

LEFT OUTER JOIN 社員 B ON 文書.承認者 ID = B. 社員 ID

エ 文書, 社員 A, 社員 B

WHERE 文書.作成者 ID = A. 社員 ID AND 文書.承認者 ID = B. 社員 ID

問6：正解（ア）

SQLに関する問題。3表以上の外部結合について問われている。問題文の〔結果〕を得るには、次のようにしなければならない。

- ① “文書”表と“社員”表を結合する（結合キーは、‘作成者ID’と‘社員ID’）
- ② “文書”表と“社員”表を結合する（結合キーは、‘承認者ID’と‘社員ID’）

そして、〔結果〕の4行目の‘作成者氏名’がNULLになっていることから、“文書”表の行をすべて残すようにLEFT OUTER JOINを使わなければならないと判断する。この時（LEFT OUTER JOINを使う時）の構文が次のようになることを知っていれば、選択肢アが正解だとわかるだろう。

<2つの表を左外部結合するときの記述>

表1 LEFT OUTER JOIN 表2 ON 結合条件

<3つの表を左外部結合するときの記述>

表1 LEFT OUTER JOIN 表2 ON 結合条件 LEFT OUTER JOIN 表3 ON 結合条件

問34 “会員”表に対し次の SQL 文を実行した結果として、正しいものはどれか。

```
SELECT X.会員名  
FROM 会員 X, 会員 Y  
WHERE X.リーダ会員番号 = Y.会員番号  
AND X.生年月日 < Y.生年月日
```

会員

会員番号	会員名	生年月日	リーダ会員番号
001	田中	1960-03-25	002
002	鈴木	1970-02-15	002
003	佐藤	1975-05-27	002
004	福田	1960-10-25	004
005	渡辺	1945-09-01	004

ア

会員名

(該当者なし)

イ

会員名
佐藤

ウ

会員名
鈴木
福田

エ

会員名
田中
渡辺

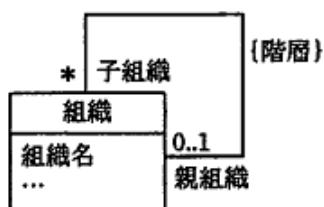
問 34：正解工

問のSQL文では、会員テーブルに別名XとYを指定し、会員テーブル同士をリーダの生年月日が得られるように結合している。結合した結果のテーブルに対して、会員とその会員のリーダの生年月日を比較し、生年月日がリーダより小さい会員名を検索している。したがって、正解はエとなる。[参照 ➤ 「3.3 SELECT」](#)

問 6 複数の事業部、部、課及び係のような組織階層の概念データモデルを、第 3 正規形の表、

組織（組織 ID, 組織名, …）

として実装した。組織の親子関係を表示する SQL 文中の a に入れるべき適切な字句はどれか。ここで、“組織”表記述中の下線部は、主キーを表し、追加の属性を想定する必要がある。また、モデルの記法として UML を用いる。{階層} は組織の親子関係が循環しないことを指示する制約記述である。



```
SELECT 組織1.組織名 AS 親組織, 組織2.組織名 AS 子組織
FROM 組織 AS 組織1, 組織 AS 組織2
WHERE [ ] a
```

- ア 組織1.親組織ID = 組織2.子組織ID
- イ 組織1.親組織ID = 組織2.組織ID
- ウ 組織1.組織ID = 組織2.親組織ID
- エ 組織1.組織ID = 組織2.子組織ID

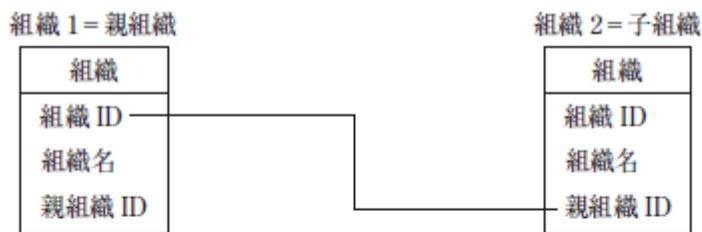
問6：正解ウ

自己結合時のSQL文に関する問題。

この図のパターンの組織階層のスキーマは、以下のようになると考えられる。

組織 (組織ID, 組織名, 親組織ID ...)

これを図示すると、このようになる。



自己結合の問題では、このように問題文のとおりに図を描いてみるとよくわかる。組織1と組織2の結合条件は、組織1の組織IDと、組織2の親組織IDになる。したがって、選択肢ウが正解になる。

問15 関係 A と B に対して和集合演算が成立するための必要十分条件はどれか。

- ア 同じ属性名でドメインが等しい属性が含まれている。
- イ 次数が同じで、対応する属性のドメインが等しい。
- ウ 主キー属性のドメインが等しい。
- エ 濃度（タブル数）が同じで、ドメインが等しい属性が少なくとも一つ存在する。

問7 地域別に分かれている同じ構造の三つの商品表，“東京商品”，“名古屋商品”，“大阪商品”がある。次の SQL 文と同等の結果が得られる関係代数式はどれか。ここで、三つの商品表の主キーは“商品番号”である。また、 $X - Y$ は X から Y の要素を除いた差集合を表す。

```
SELECT * FROM 大阪商品
  WHERE 商品番号 NOT IN (SELECT 商品番号 FROM 東京商品)
UNION
SELECT * FROM 名古屋商品
  WHERE 商品番号 NOT IN (SELECT 商品番号 FROM 東京商品)
```

- ア $(\text{大阪商品} \cap \text{名古屋商品}) - \text{東京商品}$
- イ $(\text{大阪商品} \cup \text{名古屋商品}) - \text{東京商品}$
- ウ $\text{東京商品} - (\text{大阪商品} \cap \text{名古屋商品})$
- エ $\text{東京商品} - (\text{大阪商品} \cup \text{名古屋商品})$

問 15：正解（イ）

関係代数の和集合（UNION）に関する問題（[参照 第4章](#)）。和（演算）は、“R”と“S”のOR演算を意味する。和両立の場合のみ成立する演算で、SQL文では、UNIONを用いて表現する。和両立とは、二つのリレーションの構造がすべて一致することを指す。具体的には、①属性の数（次数）が同じで、②各属性の並びとタイプが同じ（対応する属性のドメインが等しい）こと。これは選択肢イになる。

※上記の“R”と“S”は、問題文の“A”と“B”に当たるものです。読み替えてください。

問 7：正解イ

過去問題（平成19年・DB午前問26など）
[参照 「3.4.2 SELECT文で使える便利なテクニック」](#)

SQL（第3章）－集合演算子に関する問題。UNIONは、複数のSELECT文の和集合を返す。集合演算子を使っている場合、その集合演算子の左右のSELECT文を個別に解釈して、その結果の集合を求める。今回の問題文のSQL文も集合演算子のUNIONを軸に「A UNION B」のように分けて個別に解釈すると、A及びBのSELECT文の意味は、それぞれ次のようになる。

A：“東京商品”表には無く、“大阪商品”表にだけ存在する“商品番号”

B：“東京商品”表には無く、“名古屋商品”表にだけ存在する“商品番号”

この和集合（＝UNION）なので、“大阪商品”表と“名古屋商品”表にだけ存在する“商品番号”が抽出される。それを念頭に置きながら、選択肢の中から同じ意味のものを選べば良い。

アとウ：「大阪商品UN名古屋商品」は、“大阪商品”表と“名古屋商品”表の両方に存在する“商品番号”。その時点で誤り。

イ：「大阪商品UN名古屋商品」は、“大阪商品”表と“名古屋商品”表の少なくともどちらか一方に存在する“商品番号”。そこから“東京商品”表に存在する“商品番号”を除外するので、問題文のSQLの結果と同じになる。これが正解。

エ：“東京商品”表から「大阪商品UN名古屋商品」の結果を除外してしまうと、“東京商品”表にだけしか存在しない“商品番号”が抽出されるので誤り。

問11 関係 R, S に次の演算を行うとき、 R と S が和両立である必要のないものはどれか。

ア 共通集合

イ 差集合

ウ 直積

エ 和集合

問 11：正解（ウ）

集合演算に関する問題。和両立とは、二つのリレーションの構造がすべて一致する関係性のこと。具体的には、①属性の数が同じ（次数が同じ）で、②各属性の並びとタイプが同じ（対応する属性のドメインが等しい）になる構造のことである。

選択肢の演算のうち、共通集合、和集合、差集合に関しては和両立が必須条件になる。直積だけは和両立である必要はない。したがってウが正解になる。これは覚えておこう。

問 9 SQL 文 1 と SQL 文 2 を実行した結果が同一になるために、表 R が満たすべき必要十分な条件はどれか。

[SQL 文 1]

```
SELECT * FROM R UNION SELECT * FROM R
```

[SQL 文 2]

```
SELECT * FROM R
```

- ア 値に NULL をもつ行は存在しない。
- イ 行数が 0 である。
- ウ 重複する行は存在しない。
- エ 列数が 1 である。

問9：正解（ウ）

SQLに関する問題。[SQL文1]と[SQL文2]の実行結果が同一になるために、満たすべき条件について問われている。[SQL文2]では、R表から単純に全ての行と列を抽出している。一方、[SQL文1]では、UNIONを使ってR表どうしの和を求めている。選択肢の逆の条件でも[SQL文1]と[SQL文2]の実行結果が同一になったら、その条件は必要十分な条件ではないことが証明できる。その観点で選択肢を順番に見ていく。

ア：R表の値にNULLを持つ行があっても[SQL文2]ではそのまま抽出する。一方[SQL文1]でUNIONを使って和を求めても、そのまま[SQL文2]の実行結果と同じになる。したがって、「NULLをもつ行は存在しない」ことが同一になるための必要十分な条件にはならない。

イ：R表の行数が0でない場合は、普通に[SQL文1]と[SQL文2]の実行結果が同一になるケースがある。必要十分な条件ではない。

ウ：R表に重複する行が存在する場合、[SQL文1]のようにUNIONでR表の和を求めると、重複行は一つにまとめられてしまう。その結果、[SQL文2]と同一の実行結果にはならない。したがって、R表に重複する行が存在しないことが必要十分な条件になる。これが正解。

R	
顧客	月間売上
Aさん	200
Aさん	200
Bさん	150
Cさん	140
Dさん	140

UNION

R	
顧客	月間売上
Aさん	200
Aさん	200
Bさん	150
Cさん	140
Dさん	140

[SQL文1]の実行結果

顧客	月間売上
Aさん	200
Bさん	150
Cさん	140
Dさん	140

[SQL文2]の実行結果

顧客	月間売上
Aさん	200
Aさん	200
Bさん	150
Cさん	140
Dさん	140

エ：列数が1ではない場合でも、普通に[SQL文1]と[SQL文2]の実行結果が同一になるケースがある。必要十分な条件ではない。

問6 “商品”表と“当月商品仕入合計”表に対して、SQL文を実行した結果はどれか。

商品

商品コード	仕入先コード
S001	K01
S002	K01
S003	K02
S004	K02
S005	K03
S006	K04

当月商品仕入合計

仕入先コード	仕入合計金額
K01	150,000
K03	100,000
K05	250,000

[SQL文]

```
(SELECT 仕入先コード FROM 商品)
EXCEPT
(SELECT 仕入先コード FROM 当月商品仕入合計)
```

ア	イ	ウ	エ
仕入先コード	仕入先コード	仕入先コード	仕入先コード
K01	K01	K02	K02
K01	K03	K02	K04
K03		K04	

問6：正解（工）

SQLに関する問題。EXCEPTについて問われている。EXCEPTは、その前にあるSELECT文等の結果行から、その後ろにあるSELECT文等の結果行を差し引いて、差分を求める命令になる。問題文のようなSQL文だと、図のように右側の表にある‘仕入先コード’が‘K01’, ‘K03’, ‘K05’を、左側の表の‘仕入先コード’から差し引くイメージだ。

(SELECT 仕入先コード FROM 商品) (SELECT 仕入先コード FROM 当月商品仕入合計)

商品	仕入先コード
S001	K01
S002	K01
S003	K02
S004	K02
S005	K03
S006	K04

当月商品仕入合計	仕入合計金額
K01	150,000
K03	100,000
K05	250,000

右の表にあるものは差し引く

残ったのは‘K02’, ‘K02’, ‘K04’になるが、EXCEPTだけだと（EXCEPT ALLではない場合）重複行を一つにまとめることになるので、‘K02’, ‘K04’だけになる。したがって、正解は“工”になる。

問 5 “社員番号”と“氏名”を列としてもつ R 表と S 表に対して、差 (R-S) を求める SQL 文はどれか。ここで、R 表と S 表の主キーは“社員番号”であり、“氏名”は“社員番号”に閲数従属する。

- ア SELECT R.社員番号, S.氏名 FROM R, S
WHERE R.社員番号 <> S.社員番号
- イ SELECT 社員番号, 氏名 FROM R
UNION SELECT 社員番号, 氏名 FROM S
- ウ SELECT 社員番号, 氏名 FROM R
WHERE NOT EXISTS (SELECT 社員番号 FROM S
WHERE R.社員番号 = S.社員番号)
- エ SELECT 社員番号, 氏名 FROM S
WHERE S.社員番号 NOT IN (SELECT 社員番号 FROM R
WHERE R.社員番号 = S.社員番号)

問5：正解ウ

NEW

参照 「3.4.8 相関副問合せ」

SQL(第3章) - 副問合せに関する問題。同じ属性を持つ二つの社員表“R”表と“S”表があり、その差($R - S$)を求める。この二つの表は‘社員番号’を主キーに‘氏名’だけの属性を持つシンプルなものである。そして、 $(R - S)$ をわかりやすく言い換えると、「R表にだけ存在する社員を抽出せよ」という指示をしていることだとわかるだろう。それを表しているSQL文を探し出せば良い。選択肢のSQL文を順次解釈しながら見ていく。

ア：“R”表と“S”表で異なる‘社員番号’の社員、すなわち、どちらか一方にしか存在しない社員を抽出するSQLなので、誤り。

イ：“R”表と“S”表の和集合なので、どちらか一方にいる社員すべてになる。誤り。

ウ：“R”表の社員をチェックして(主問合せ)、“S”表に同じ社員番号が存在しないとき(相関副問合せ)に抽出する。すなわち、“R”表にだけ存在する社員なので、これが正解。

エ：“S”表の社員をチェックして(主問合せ)、“R”表に同じ社員番号が存在しないとき(相関副問合せ)に抽出する。すなわち、“S”表にだけ存在する社員なので誤り。

上記アの解説を下記のように修正するとともに詳細にしました (2024.5.27)

- 結合条件に等号(=)を使う結合を等結合という(等価結合、等値結合ともいふこともあるが、情報処理技術者試験では令和4年の午前Ⅱの問題で“等結合”としている)。一方、選択肢アのSQLのように、結合条件に等号を使わない(不等号や否定、範囲指定など等号以外を使う)結合を非等価結合や非等値結合といふ。
- FROM句に複数表を指定すると直積(各表の全ての列を属性とし、かつすべての行の組み合わせ)になる。JOINを使う場合の“CROSS JOIN”と同等で「交差結合」という。そのため、SELECT文の選択項目リストで抽出する列や、WHERE句で抽出する条件指定をして必要な部分だけを抽出することになる。

R

社員番号	氏名
1	miyoshi
2	yamada
4	aoyma
5	miyuki

S

社員番号	氏名
1	miyoshi
3	take
6	tozaki

「SELECT * FROM R, S」の実行結果

社員番号	氏名	社員番号	氏名
1	miyoshi	1	miyoshi
1	miyoshi	3	take
1	miyoshi	6	tozaki
2	yamada	1	miyoshi
2	yamada	3	take
2	yamada	6	tozaki
4	aoyma	1	miyoshi
4	aoyma	3	take
4	aoyma	6	tozaki
5	miyuki	1	miyoshi
5	miyuki	3	take
5	miyuki	6	tozaki

ここで WHERE 句に等結合を使うと、下図のように両方の表に登録されている者だけが抽出対象になるので、これが結合条件と同じ役割を果たし INNER JOIN と同じ結果になる。SELECT 以下に選択項目リストを指定すると、よく見かける使い方になる。

「SELECT * FROM R, S WHERE R.社員番号 = S.社員番号」の実行結果

社員番号	氏名	社員番号	氏名
1	miyoshi	1	miyoshi
1	miyoshi	3	take
1	miyoshi	6	tozaki
2	yamada	1	miyoshi
2	yamada	3	take
2	yamada	6	tozaki
4	aoyma	1	miyoshi
4	aoyma	3	take
4	aoyma	6	tozaki
5	miyuki	1	miyoshi
5	miyuki	3	take
5	miyuki	6	tozaki

※この行だけを抽出

一方、非等値結合の「R.社員番号 <> S.社員番号」のように指定すると、(上記の図のように) 両方にある社員の組み合わせを除いた部分が抽出対象になる (下図参照)。

「SELECT * FROM R, S WHERE R.社員番号 <> S.社員番号」の実行結果

社員番号	氏名	社員番号	氏名
1	miyoshi	1	miyoshi
1	miyoshi	3	take
1	miyoshi	6	tozaki
2	yamada	1	miyoshi
2	yamada	3	take
2	yamada	6	tozaki
4	aoyma	1	miyoshi
4	aoyma	3	take
4	aoyma	6	tozaki
5	miyuki	1	miyoshi
5	miyuki	3	take
5	miyuki	6	tozaki

これらが対象

選択肢アでは、このうち R.社員番号と S.氏名を抽出しているので、実行結果は次のようなになる。各 SQL 製品の仕様によって実行結果は微妙に異なるが、少なくとも “R – S” にはならない。

「SELECT R.社員番号, S.氏名
FROM R, S WHERE R.社員番号 <> S.社員番号」の実行結果

社員番号	氏名	社員番号	氏名
1	miyoshi	1	miyoshi
1	miyoshi	3	take
1	miyoshi	6	tozaki
2	yamada	1	miyoshi
2	yamada	3	take
2	yamada	6	tozaki
4	aoyma	1	miyoshi
4	aoyma	3	take
4	aoyma	6	tozaki
5	miyuki	1	miyoshi
5	miyuki	3	take
5	miyuki	6	tozaki

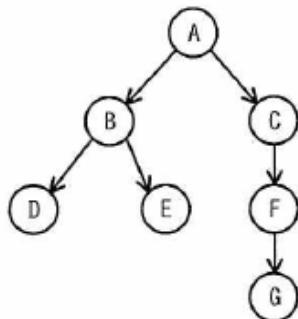
これらが対象

ここが対象

ここが対象

memo

問8 図のツリー構造に対するデータを格納した“隣接リスト”表から、リーフノードを取得するSQL文のaに入れる字句はどれか。ここで、図の丸はノードを表し、矢印は親ノードから子ノードへの関係を表す。



隣接リスト

親ノード	子ノード
NULL	A
A	B
A	C
B	D
B	E
C	F
F	G

[SQL文]

SELECT 子ノード FROM 隣接リスト

SELECT 親ノード FROM 隣接リスト

ア EXCEPT

イ INTERSECT

ウ UNION

エ UNION ALL

問8：正解（ア）

SQLに関する問題。集合演算について問われている。ここで問われているリーフノードとは、ツリー構造の末端、つまり子を持たないノードのことを言う。この図では、D・E・Gのことを指す。そして、そのリーフノードを取得するSQL文を完成させるために使う集合演算を、選択肢の中から一つ選ぶ。解答に当たっては、空欄aに選択肢ア～エを入れた結果から考えればいいだろう。ちなみに、上段のSELECT文と下段のSELECT文の実行結果は次のようになる。

```
SELECT 子ノード FROM 隣接リスト … {A,B,C,D,E,F,G}  
SELECT 親ノード FROM 隣接リスト … {NULL,A,A,B,B,C,F}
```

ア：EXCEPTは差集合である。上段-下段になるので、下段にはない{D,E,G}が取得される。これはリーフノードと同じなので正解になる。

イ：INTERSECTは共通集合である。上段と下段の両方にある{A,B,C,F}が取得される。

ウ：UNIONは和集合である。但し、重複無しなので{NULL,A,B,C,D,E,F,G}が取得される。

エ：UNION ALLも和集合だが重複をそのまま取得する。{A,B,C,D,E,F,G,NULL,A,A,B,B,C,F}が取得される。

なお、SELECT文単体やUNIONではNULLはそのまま出力されるが、EXCEPT(差集合)やINTERSECT(共通集合)ではNULLは比較対象にならないため、結果に影響しない。

問10 和両立である関係 R と S がある。 $R \cap S$ と等しいものはどれか。ここで、 $-$ は差演算、 \cap は共通集合演算を表す。

ア $(R-S)-(S-R)$

ウ $R-(S-R)$

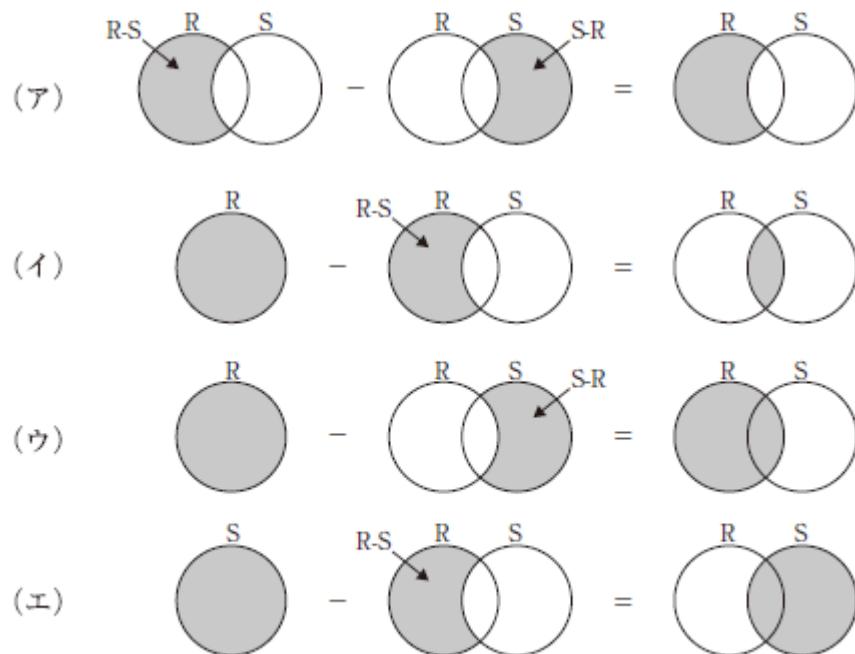
イ $R-(R-S)$

エ $S-(R-S)$

問 10：正解（イ）

関係代数の演算に関する問題。“和両立である関係”とは、二つのリレーションの構造がすべて一致している関係（①属性の数が同じ、②各属性の並びとタイプが同じ関係）のことである。また、“ $R \cap S$ ”は、設問にもあるように共通集合演算のことで、AND 演算を意味する。すなわち、 R と S のどちらにも全く同じ形で存在するものだけが対象になる。

その関係を念頭に、選択肢を順番に見ていく。このとき、以下のようにベン図を描いて整理していくけば、間違いないだろう。



令和2年度・DB・午前Ⅱ

問9 関係代数における直積に関する記述として、適切なものはどれか。

- ア ある属性の値に付加した条件を満たす全てのタブルの集合である。
- イ ある一つの関係の指定された属性だけを残して、他の属性を取り去って得られる属性の集合である。
- ウ 二つの関係における、あらかじめ指定されている二つの属性の 2 項関係を満たす全てのタブルの組合せの集合である。
- エ 二つの関係における、それぞれのタブルの全ての組合せの集合である。

令和4年度・DB・午前Ⅱ

問11 関係 R, S の等結合演算は、どの演算によって表すことができるか。

- | | |
|-----------|---------|
| ア 共通 | イ 差 |
| ウ 直積と射影と差 | エ 直積と選択 |

問 9：正解（エ）

関係代数の演算に関する問題。問題文で問われている直積は、二つの関係におけるすべてのタプルの組合せのことなので、正解はエになる。

ア：選択の説明である。誤り

イ：射影の説明である。誤り

ウ：結合の説明である。誤り

問 11：正解（エ）

関係代数の演算に関する問題。等結合とは、二つのテーブルを共通の属性で結合し、結合に使用した属性を重複したまま残すのが特徴である。等結合は、関係代数の直積と選択で表すことができる。よって正解は（エ）になる。

① 直積にて、すべての組み合わせを、すべての属性とともに表示する

② 結合に使用した属性が同じ値のものだけ選択する

問11 関係 R と関係 S において、 $R \div S$ の関係演算結果として、適切なものはどれか。ここで、 \div は商演算を表す。

R

店	商品
A	a
A	b
B	a
B	b
B	c
C	c
D	c
D	d
E	d
E	e

S

商品
a
b
c

ア

店
A
A
B
B
B
C
D

イ

店
A
B
C
D

ウ

店
B

エ

店
E

問 11：正解（ウ）

関係代数の演算に関する問題。関係代数の商は、四則演算の割り算にあたる。関係 R において、関係 S の商品 (a,b,c) すべてをもつ店は B のみである。したがって、正解は（ウ）となる。

R	
店	商品
A	a
A	b
B	a
B	b
B	c
C	c
D	c
D	d
E	d
E	e

↓

S	
商品	
a	
b	
c	

商品 (a,b,c) の組合せを持っているのは "B" 店だけになる。したがって、正解は(ウ)になる。

問 9 関係 R と関係 S から、関係代数演算 $R \div S$ で得られるものはどれか。ここで、 \div は商の演算を表す。

R

X	Y	Z
a	1	甲
b	2	甲
a	1	乙
b	2	丙

S

X	Y
a	1
b	2

ア

X	Y	Z
a	1	甲
b	2	甲

イ

Z
乙
丙

ウ

Z
甲
乙
丙

エ

Z
甲

問9：正解工

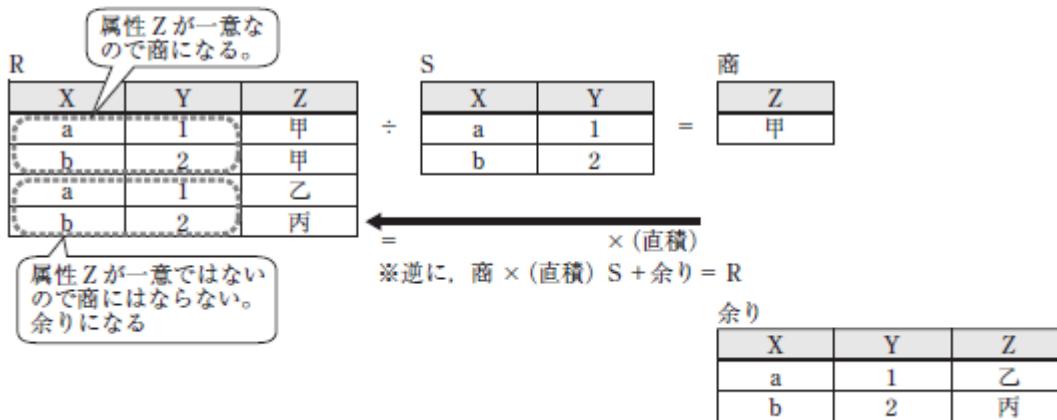
過去問題（平成19年・DB午前問28など）

参照 第4章 重要キーワード「4 関係代数」

関係代数の演算に関する問題。商の計算においては、イメージ的に説明すると、①割る数“S”のパターンが、割られる数“R”の中に出てきていて、②その残りの属性が行単位で同じものということになる（図参照）。これは、逆算して考えるとイメージしやすいかもしれない（直積になるので）。（図の表現を使っての説明になるが）次のような関係が成立する。

$$\begin{aligned} "R" \div "S" &= \text{"商"} \cdots \text{"余り"} \\ \rightarrow "S" \times (\text{直積}) \text{"商"} + \text{"余り"} &= "R" \end{aligned}$$

今回の問題では、上記①②を満足させる“商”は選択肢Eになる。



問10 次の関係 R, S, T, U において、関係代数表現 $R \times S \div T - U$ の演算結果はどれか。

ここで、 \times は直積、 \div は商、 $-$ は差の演算を表す。

関係 R	A	B
1	a	
2	b	
3	a	
3	b	
4	a	

関係 S	C
x	
y	

関係 T	A
1	
3	

関係 U	B	C
a	x	
c	z	

ア	B	C
a	y	

イ	B	C
b	x	

ウ	B	C
a	$y-x$	
b	x	
b	y	

エ	B	C
a	$y-x$	
-c	-z	

問 10：正解（ア）

過去問題（平成22年・DB午前 問12など）

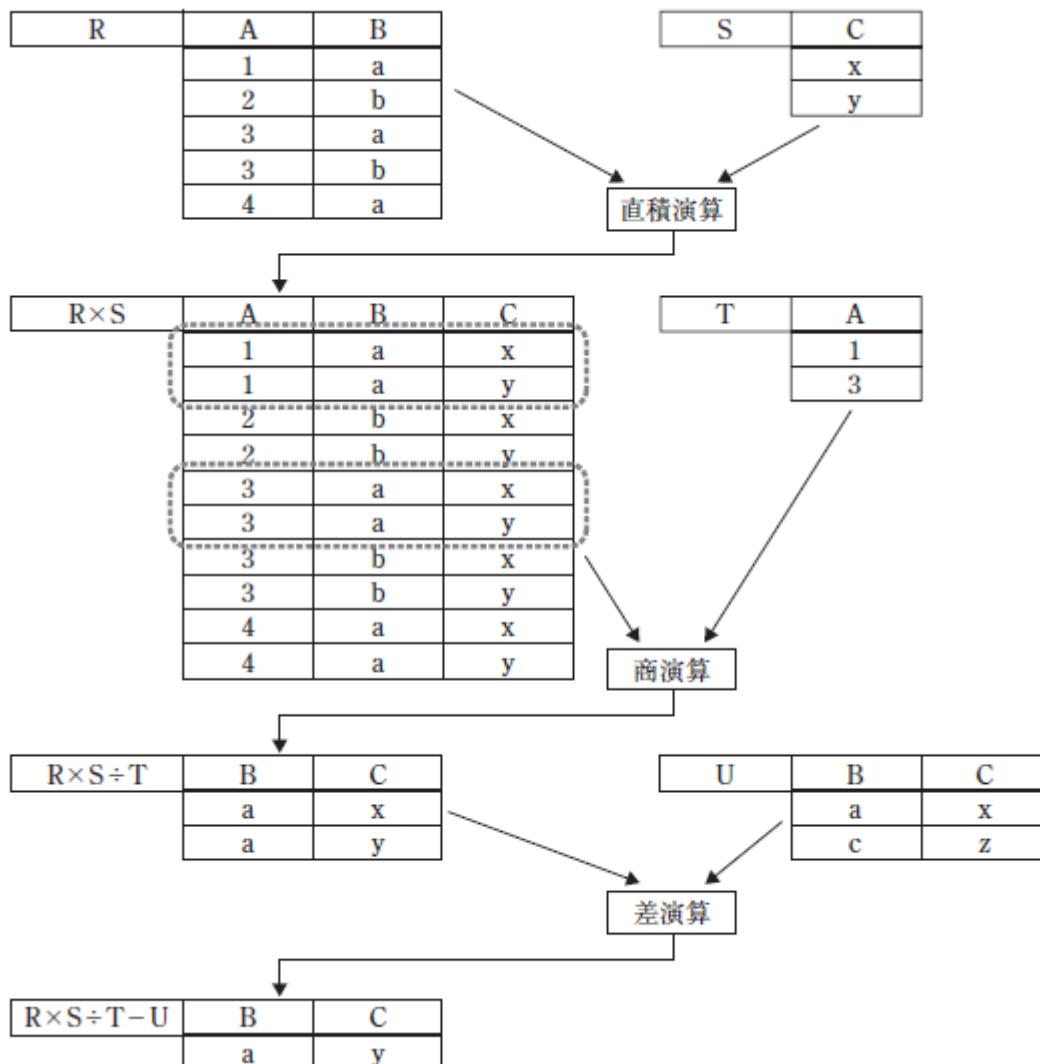
参照▶ 第4章 重要キーワード「4 関係代数」

関係代数の演算に関する問題。演算の優先順位は次のとおりなので、順番に確認する。この結果、正解は（ア）になる。

$$\textcircled{1} \quad R \times S$$

$$\textcircled{2} \quad \textcircled{1} \div T$$

$$\textcircled{3} \quad \textcircled{2} - U$$



問11 “社員”表と“プロジェクト”表に対して、次の SQL 文を実行した結果はどれか。

```
SELECT プロジェクト番号, 社員番号 FROM プロジェクト
WHERE 社員番号 IN
(SELECT 社員番号 FROM 社員 WHERE 部門 <= '2000')
```

社員

社員番号	部門	社員名
11111	1000	佐藤一郎
22222	2000	田中太郎
33333	3000	鈴木次郎
44444	3000	高橋美子
55555	4000	渡辺三郎

プロジェクト

プロジェクト番号	社員番号
P001	11111
P001	22222
P002	33333
P002	44444
P003	55555

ア

プロジェクト番号	社員番号
P001	11111
P001	22222

イ

プロジェクト番号	社員番号
P001	22222
P002	33333

ウ

プロジェクト番号	社員番号
P002	33333
P002	44444

エ

プロジェクト番号	社員番号
P002	44444
P003	55555

問 11：正解（ア）

過去問題（平成19年・DB午前問35など）
参考▶「3.4.8 相関副問合せ」

SQL – SELECT文 – 副問合せの問題。SQL文を解釈していく。

最初に、下記の部分を実行する。その結果、下図の①が抽出される（抽出されるのは、社員番号なので‘11111’と‘22222’が対象になる）。

```
SELECT 社員番号 FROM 社員 WHERE 部門 <= '2000'
```

次に、下記の部分を実行する。その結果、下図の②が抽出される。これは選択肢（ア）になる。

```
SELECT プロジェクト番号, 社員番号 FROM プロジェクト  
WHERE 社員番号 (IN '11111', '22222')
```

社員			プロジェクト		
社員番号	部門	社員名	プロジェクト番号	社員番号	
11111	1000	佐藤一郎	P001	11111	②
22222	2000	田中太郎	P001	22222	
33333	3000	鈴木次郎	P002	33333	
44444	3000	高橋美子	P002	44444	
55555	4000	渡辺三郎	P003	55555	

問37 二つの表“納品”, “顧客”に対する次の SQL 文と同じ結果が得られる SQL 文はどれか。

```
SELECT 顧客番号, 顧客名 FROM 顧客
  WHERE 顧客番号 IN
    (SELECT 顧客番号 FROM 納品
      WHERE 商品番号 = 'G1')
```

納品			顧客	
商品番号	顧客番号	納品数量	顧客番号	顧客名

- ア SELECT 顧客番号, 顧客名 FROM 顧客
WHERE 'G1' IN (SELECT 商品番号 FROM 納品)
- イ SELECT 顧客番号, 顧客名 FROM 顧客
WHERE 商品番号 IN
(SELECT 商品番号 FROM 納品
WHERE 商品番号 = 'G1')
- ウ SELECT 顧客番号, 顧客名 FROM 納品, 顧客
WHERE 商品番号 = 'G1'
- エ SELECT 顧客番号, 顧客名 FROM 納品, 顧客
WHERE 納品.顧客番号 = 顧客.顧客番号 AND 商品番号 = 'G1'

問37：正解工

SQLの副問合せに関する知識を問う問題。問のSQL文は、商品番号‘G1’を納品した顧客の顧客番号を顧客テーブルから検索している。[参照](#) 「3.3.7 副問合せ」

- ア：顧客テーブル商品番号が存在しないため‘G1’を指定するのはおかしい。
- イ：顧客テーブルには、商品番号が存在しない。
- ウ：納品テーブルと顧客テーブルの直積から、商品番号が‘G1’であるものを検索している。

問34 T1 表と T2 表が、次のように定義されているとき、次の SELECT 文と同じ検索結果が得られる SELECT 文はどれか。

(T1 表の定義)

```
CREATE TABLE T1 (SNO CHAR(6) PRIMARY KEY, SNAME CHAR(20))
```

(T2 表の定義)

```
CREATE TABLE T2 (CODE CHAR(4), SNO CHAR(6), SURYO INT)
```

(SELECT 文)

```
SELECT DISTINCT T1.SNAME  
FROM T1, T2  
WHERE T1.SNO = T2.SNO  
ORDER BY T1.SNAME
```

ア SELECT DISTINCT SNAME
FROM T1
WHERE SNO IN (SELECT SNO FROM T2)
ORDER BY SNAME

イ SELECT DISTINCT SNAME
FROM T1
WHERE T1.SNO IN (SELECT SNO FROM T1)
ORDER BY SNAME

ウ SELECT SNAME
FROM T1
WHERE SNO NOT IN (SELECT SNO FROM T2)
ORDER BY SNAME

エ SELECT T2.SNAME
FROM T1, T2
WHERE T1.SNO = T2.SNO
ORDER BY T2.SNAME

問 34：正解ア

SQL – SELECT 文 – 内部結合の問題。問題文の SELECT 文を解釈すると、「① T1 と T2 を SNO で内部結合し、②重複行を 1 行にまとめ（DISTINCT 句）、T1.SNAME の昇順に表示する」という意味になる。これを念頭に選択肢を順番に見ていく。

ア：問題文と同じ。よってこれが正解である。なお、内部結合はこのように、副問合せを使っても表現できる

イ：T2 を利用していない。よって①に反するので同じ結果は得られない

ウ：DISTINCT 句を使っていないので②に反する。同じ結果は得られない

エ：DISTINCT 句を使っていないので②に反する。同じ結果は得られない

問14 “製品”表と“在庫”表に対し、次の SQL 文を実行した結果として得られる表の行数は幾つか。

```
SELECT DISTINCT 製品番号 FROM 製品
WHERE NOT EXISTS (SELECT 製品番号 FROM 在庫
WHERE 在庫数 > 30 AND 製品.製品番号 = 在庫.製品番号)
```

製品

製品番号	製品名	単価
AB1805	CD-ROM ドライブ	15,000
CC5001	ディジタルカメラ	65,000
MZ1000	プリンタ A	54,000
XZ3000	プリンタ B	78,000
ZZ9900	イメージスキャナ	98,000

在庫

倉庫コード	製品番号	在庫数
WH100	AB1805	20
WH100	CC5001	200
WH100	ZZ9900	130
WH101	AB1805	150
WH101	XZ3000	30
WH102	XZ3000	20
WH102	ZZ9900	10
WH103	CC5001	40

ア 1

イ 2

ウ 3

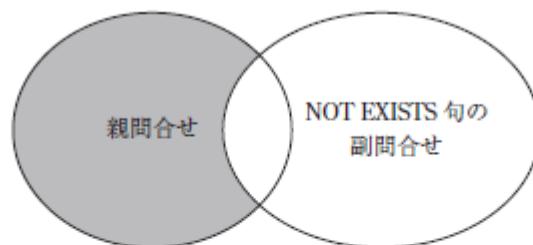
エ 4

問 14：正解イ

SQL文に関する問題。相関副問合せ（参照 ➔ 3.4.8 相関副問合せ）について問われている。

このSQL文は、「在庫数が30を超えているデータが在庫表に存在しない（を除く）製品番号を抽出する」という意味である。在庫表には、同一の製品番号が複数ある点を加味すると、「在庫数が30を超えて在庫を保有している倉庫が存在しない製品番号」であり、言い換えると「すべての倉庫の在庫数が30以下の製品」である。この手の問題では、境界値の“30”が入るかどうかの解釈が成否を分けるポイントになるので、そこを確実にチェックしておくようにしよう。ちなみに、この相関副問合せの結果セットは下図の網掛け部分になる。

表から、「すべての倉庫の在庫数が30以下の製品番号」をピックアップすると、“XZ3000”と“MZ1000”となる。したがって、正解はイである。



親問合せ	SELECT 製品番号 FROM 製品
NOT EXISTS 句の副問合せ	SELECT 製品番号 FROM 在庫 WHERE 在庫数 > 30 AND 製品. 製品番号 = 在庫. 製品番号

問8 “社員”表に対して、SQL文を実行して得られる結果はどれか。ここで、実線の下線は主キーを表し、表中のNULLは値が存在しないことを表す。

社員

社員コード	上司	社員名
S001	NULL	A
S002	S001	B
S003	S001	C
S004	S003	D
S005	NULL	E
S006	S005	F
S007	S006	G

[SQL文]

```
SELECT 社員コード FROM 社員 X
WHERE NOT EXISTS
    (SELECT * FROM 社員 Y WHERE X.社員コード = Y.上司)
```

ア	社員コード	イ	社員コード	ウ	社員コード	エ	社員コード
	S001		S001		S002		S003
	S003		S005		S004		S006
	S005				S007		
	S006						

問8：正解（ウ）

SQLに関する問題。NOT EXISTSを使った相関副問合せについて問われている。自己参照しているので、下図のように図示してみるとわかりやすくなる。

社員 X		
社員コード	上司	社員名
S001	NULL	A
S002	S001	B
S003	S001	C
S004	S003	D
S005	NULL	E
S006	S005	F
S007	S006	G

社員 Y		
社員コード	上司	社員名
S001	NULL	A
S002	S001	B
S003	S001	C
S004	S003	D
S005	NULL	E
S006	S005	F
S007	S006	G

上図の左側の“社員 X”が外側のSELECT文になる。そこから1件ずつ取り出し、右側の“社員 Y”的‘上司’と結合する。そして‘NOT EXISTS’なので、右側の上司の中に結合しようとしている社員コードが存在しない‘社員コード’を抽出する（上記の枠内の‘S002’であれば、社員 Yには存在しないので抽出対象になる）。つまり「誰の上司にもなっていない社員を抽出する」ことになる。その結果‘S002’, ‘S004’, ‘S007’が抽出されるので、正解は選択肢ウになる。

問10 “従業員”表から、男女それぞれの最年長従業員を除く全ての従業員を取り出すSQL文とするために、aに入れる字句はどれか。ここで、“従業員”表の構造は次のとおりであり、実線の下線は主キーを表す。

従業員 (従業員番号, 従業員名, 性別, 生年月日)

[SQL文]

```
SELECT 従業員番号, 従業員名 FROM 従業員 AS S1  
      WHERE 生年月日 > ( a )
```

- ア SELECT MIN(生年月日) FROM 従業員 AS S2
 GROUP BY S2.性別
- イ SELECT MIN(生年月日) FROM 従業員 AS S2
 WHERE S1.生年月日 > S2.生年月日
 OR S1.性別 = S2.性別
- ウ SELECT MIN(生年月日) FROM 従業員 AS S2
 WHERE S1.性別 = S2.性別
- エ SELECT MIN(生年月日) FROM 従業員
 GROUP BY S2.性別

問 10：正解（ウ）

SQLに関する問題。SELECT文の副問合せに関して問われている。要求されている内容（男女それぞれの最年長従業員を除くすべての従業員を取り出す SQL文）と選択肢の個々の SQL文を比較する。

<問題文より>

- ① この SELECT 文では、(a) よりも大きい生年月日（すなわち、年下）の従業員の従業員番号と従業員名を選択している
- ② 対象は“従業員”表だけなので、“従業員”表どうしで比較する
- ③ 「FROM 従業員 AS S1」より、「従業員 AS S2」が空欄 a の中に存在する（相関）副問合せを使っていると推測できる

<選択肢より>

- ④ 「SELECT MIN（生年月日） FROM 従業員」までは全部同じ
- ⑤ 一つの“従業員”表を二つの用途で使っているので、別名を付けなければならない。
したがって選択肢エは誤り

以上をざっと整理した上で、解答するためのポイントを押さえていこう。

まず、副問合せの比較条件（空欄 a を含む（）の前）が“>”である点に注意する。等号、不等号で条件比較する場合、副問合せで複数の値が返ってくるとエラーになる。そのため、選択肢アとエのように、性別の GROUP BY を使うと、男女 2 件の複数値が返ってくるため誤りである。選択肢イとウは、その点はクリアしている。

次に選択肢イとウを比較すると、選択肢イは“OR”で不要な生年月日の比較が入っているので、適当ではない。よって正解はウになる。

ちなみに、相関副問合せそのものの動きがわかりにくい場合は、選択肢ウの動きをトレースしてみるといい。図を例に見ていく。

相関副問合せは、主問合せ（S1）の1行を処理するごとに、副問合せ（S2）の処理を実行する。このとき、主問合せ（S1）の性別が“男”的な場合は、副問合せの最小値は常にAさんになる。また、“女”的な場合は常にBさんになる。よって、主問合せ側（S1）の各データは、最年長であるAさんもしくはBさんと比較して、それよりも年下（生年月日は大きくなる）のときだけ表示することになる。ただし、相関副問合せを使うと、このように主問合せを1件処理するたびに、毎回副問合せを行うから、処理効率は悪い。

S1

従業員番号	従業員名	性別	生年月日
0001	Aさん	男	S20
0002	Bさん	女	S25
0003	Cさん	男	S40
0004	Dさん	男	S32
0005	Eさん	女	S35

S2

従業員番号	従業員名	性別	生年月日
0001	Aさん	男	S20
0002	Bさん	女	S25
0003	Cさん	男	S40
0004	Dさん	男	S32
0005	Eさん	女	S35

※生年月日は便宜上、年だけにした

(1件ずつ処理される主問合せの結果)

1件目処理 0001 Aさん 男 S20

>

0001	Aさん	男	S20
0003	Cさん	男	S40
0004	Dさん	男	S32

(結果)

表示なし

2件目処理 0002 Bさん 女 S25

>

0002	Bさん	女	S25
0005	Eさん	女	S35

表示なし

3件目処理 0003 Cさん 男 S40

>

0001	Aさん	男	S20
0003	Cさん	男	S40
0004	Dさん	男	S32

0003

4件目処理 0004 Dさん 男 S32

>

0001	Aさん	男	S20
0003	Cさん	男	S40
0004	Dさん	男	S32

0004

5件目処理 0005 Eさん 女 S35

>

0002	Bさん	女	S25
0005	Eさん	女	S35

0005

memo

問16 “商品月間販売実績” 表に対して、SQL 文を実行して得られる結果はどれか。

商品月間販売実績

商品コード	総販売数	総販売金額
S001	150	45,000
S002	250	50,000
S003	150	15,000
S004	400	120,000
S005	400	80,000
S006	500	25,000
S007	50	60,000

(SQL 文)

```
SELECT A.商品コード AS 商品コード, A.総販売数 AS 総販売数
FROM 商品月間販売実績 A
WHERE 3 > (SELECT COUNT(*) FROM 商品月間販売実績 B
              WHERE A.総販売数 < B.総販売数)
```

ア

商品コード	総販売数
S001	150
S003	150
S006	500

イ

商品コード	総販売数
S001	150
S003	150
S007	50

ウ

商品コード	総販売数
S004	400
S005	400
S006	500

エ

商品コード	総販売数
S004	400
S005	400
S007	50

問 16：正解（ウ）

SQL – SELECT 文 – 相関副問合せに関する問題。この SQL 文だと、外側の SQL で 1 件ずつ取り出して順次処理していく。つまり、“商品月間販売実績”表を 1 行ずつ順次処理していく。取り出したその 1 件ごとに、その総販売数を超える総販売数のものだけを抽出して件数を数える。その合計数が 3 件を超えないものだけをピックアップする。

(例 1) や (例 2) のように、順次計算していく、合計が 0 ~ 2 件のものだけを抽出する。

〔SQL 文〕

```
SELECT A.商品コード AS 商品コード, A.総販売数 AS 総販売数
FROM 商品月間販売実績 A
WHERE 3 > (SELECT COUNT(*) FROM 商品月間販売実績 B
              WHERE A.総販売数 < B.総販売数)
```

A. 総販売数よりも、B. 総販売数の方が大きい行の件数をカウントする。

(例 1) “S001”的場合、総販売数 150 を超えるのは、“S002”, “S004”, “S005”, “S006”的 4 件になる

(例 2) “S002”的場合、総販売数 250 を超えるのは、“S004”, “S005”, “S006”的 3 件になる

“商品月間販売実績” = A の、1 件目 “S001”的総販売数は 150

“商品月間販売実績” = B の、1 件目 “S001”的総販売数は 150…カウントしない

“商品月間販売実績” = B の、2 件目 “S002”的総販売数は 250…カウントする

“商品月間販売実績” = B の、3 件目 “S003”的総販売数は 150…カウントしない

“商品月間販売実績” = B の、4 件目 “S004”的総販売数は 400…カウントする

“商品月間販売実績” = B の、5 件目 “S005”的総販売数は 400…カウントする

“商品月間販売実績” = B の、6 件目 “S006”的総販売数は 500…カウントする

“商品月間販売実績” = B の、7 件目 “S007”的総販売数は 50…カウントしない

この場合は、COUNT(*) は 4 件。3 件より小さくないので抽出（選択）の対象外。

これを、“商品月間販売実績” = A の、2 件目 “S002”以後繰り返していくと、“S004”, “S005”, “S006”が抽出対象になるので、正解はウになる。

問11 庭に訪れた野鳥の数を記録する“観測”表がある。観測のたびに通番を振り、鳥名と観測数を記録している。AVG 関数を用いて鳥名別に野鳥の観測数の平均値を得るために、一度でも訪れた野鳥については、観測されなかったときの観測数を 0 とするデータを明示的に挿入する。SQL 文の a に入る字句はどれか。ここで、通番は初回を 1 として、観測のタイミングごとにカウントアップされる。

```
CREATE TABLE 観測 (
    通番    INTEGER,
    鳥名    CHAR(20),
    観測数  INTEGER,
    PRIMARY KEY (通番, 鳥名))

INSERT INTO 観測
    SELECT DISTINCT obs1.通番, obs2.鳥名, 0
        FROM 観測 AS obs1, 観測 AS obs2
        WHERE NOT EXISTS (
            SELECT * FROM 観測 AS obs3
            WHERE [ ] a
                AND obs2.鳥名= obs3.鳥名)
```

- ア obs1.通番 = obs1.通番
- イ obs1.通番 = obs2.通番
- ウ obs1.通番 = obs3.通番
- エ obs2.通番 = obs3.通番

問 11：正解（ウ）

SQLに関する問題（参照▶第3章）。CREATE文とINSERT文を使用しているが、問われているのはEXISTSを使った相関副問合せ。まずは問題文を読んで、何をしたいのかを把握しよう。問題文を一読しても少々わかりにくいかもしれないが、この問題は次のようなことをしたいと書いている。

- 1) 通番は初回の観測を1として、観測のたびにカウントアップされる。
→ 1回目の観測は‘通番’ = 1, 2回目の観測は‘通番’ = 2
- 2) 観測ごとに（通番ごとに）、観測できた鳥は、‘鳥名’と‘観測数’を記録する
→ 観測ごとに、観測できる鳥と、観測できない鳥が発生する。登録するのは観測できた鳥だけ。
- 3) 2)に対して、（過去の観測データ全てを探索して）1度でも訪れた野鳥（すなわち、1件でもデータのある鳥名）について抽出し、その観測回に、観測されなかった鳥については観測数=0のデータを登録する。
→ INSERT文はこの処理を実施する。

上記の3)がINSERT文で実施したいことなので、下記の図のように頭の中イメージできれば、後は比較的容易に解けるだろう。解答中に、どうしてもイメージが湧かなければ、下図のような図を作成して整理してみるといいだろう。

通番	鳥名	観測数
1	ヒバリ	10
1	メジロ	15
1	キジ	1
2	ヒバリ	3
2	ウグイス	4
3	メジロ	3
3	ウグイス	2
3	セキセイインコ	3

1回目の観測 →

通番	鳥名	観測数
1	ヒバリ	10
1	メジロ	15
1	キジ	1
1	ウグイス	0
1	セキセイインコ	0
2	ヒバリ	3
2	メジロ	0
2	キジ	0
2	ウグイス	4
2	セキセイインコ	0
3	ヒバリ	0
3	メジロ	2
3	キジ	0
3	ウグイス	3
3	セキセイインコ	3

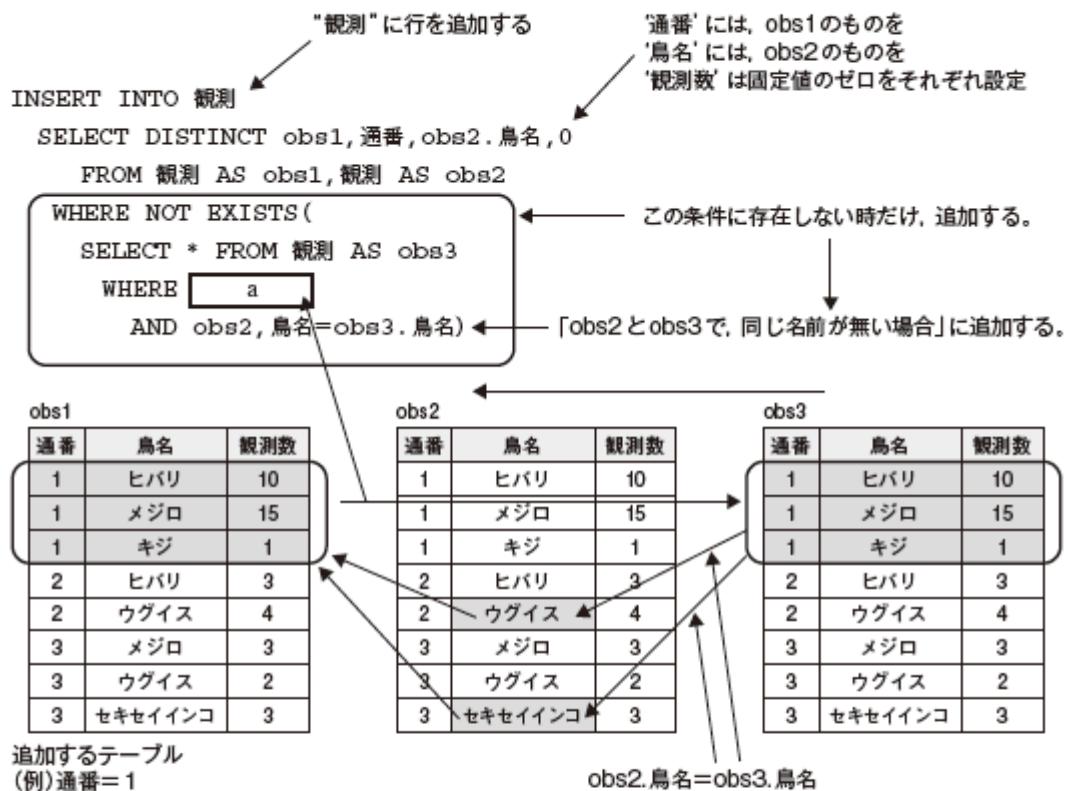
2回目の観測 →

通番	鳥名	観測数
1	ヒバリ	10
1	メジロ	15
1	キジ	1
1	ウグイス	0
1	セキセイインコ	0
2	ヒバリ	3
2	メジロ	0
2	キジ	0
2	ウグイス	4
2	セキセイインコ	0
3	ヒバリ	0
3	メジロ	2
3	キジ	0
3	ウグイス	3
3	セキセイインコ	3

3回目の観測 →

95

この問題で実施したいことは、図にあるような結果を求めることがある。例えば、1回目の観測では、2回目、3回目に観測したことのある“ウグイス”と“セキセイインコ”が観測できなかった。そこで、観測数を“0”で、1回目の観測に追加で登録したというわけだ。これが理解できれば、続いて、問題文の INSERT 文と比較しながら見ていく。



この INSERT 文では、“観測” テーブルに 3 つの役割を持たせて自己結合している。この 3 つの “観測” テーブルの役割を明確にする。

① obs1

SELECT 文の列リストを見ると、「通番」を obs1 からとっている。そのため、obs1 は、INSERT するテーブルだと考えればいいだろう。

後は、次の 2 つの条件に合致するものを obs1 に追加する。

a) obs1 と同じ ‘通番’ で、obs1 にはない ‘鳥名’

(obs1 にある鳥名は追加しないため)

b) obs1 以外の ‘通番’ で、obs2 にある ‘鳥名’

② obs2, obs3

SELECT文の列リストを見ると、‘鳥名’をobs2からとっかけていている。

ということは、obs2には存在する‘鳥名’で、かつ、これがobs1には無い‘鳥名’でなければならぬ。

問題文のSQLでは、NOT EXISTSの中の条件として、“obs2.鳥名 = obs3.鳥名”があるので、obs1に鳥名とゼロを追加する場合には、図のように、obs2とobs3で鳥名が同じでないものを追加することになる。

ということは、obs1とobs3で通番を合せないといけないことがわかるだろう。

したがって、正解は（ウ）の“obs1.通番 = obs3.通番”になる。

問9 次の SQL 文と同じ検索結果が得られる SQL 文はどれか。

```
SELECT DISTINCT TBL1.COL1 FROM TBL1  
    WHERE COL1 IN (SELECT COL1 FROM TBL2)
```

- ア SELECT DISTINCT TBL1.COL1 FROM TBL1
 UNION SELECT TBL2.COL1 FROM TBL2
- イ SELECT DISTINCT TBL1.COL1 FROM TBL1
 WHERE EXISTS
 (SELECT * FROM TBL2 WHERE TBL1.COL1 = TBL2.COL1)
- ウ SELECT DISTINCT TBL1.COL1 FROM TBL1, TBL2
 WHERE TBL1.COL1 = TBL2.COL1
 AND TBL1.COL2 = TBL2.COL2
- エ SELECT DISTINCT TBL1.COL1 FROM TBL1 LEFT OUTER JOIN TBL2
 ON TBL1.COL1 = TBL2.COL1

問9：正解（イ）

SQL文に関する問題（参照 第3章）。SELECT文で、INを使った副問合せと DISTINCT の使い方について問われている。

解答に当たっては、最初に問題文のSQL文を理解してから、選択肢を順番に見ていく。

＜問題文のSQL文の解釈＞

表 ‘TBL2’ の属性 ‘COL1’ と同じ属性 ‘COL1’ を持つタプルを、表 ‘TBL1’ から抜き出して属性 ‘COL1’ を重複なく抽出する。

ア：“UNION”を使うと和集合になる。したがって、UNIONで結合した二つのSELECT文のどちらか一方にあるものは抽出されてしまうため、誤り。

イ：同じ結果になる。INを使った副問合せと EXISTSを使った相関副問合せは、記述の仕方にもよるが同じ結果になる。これが正解。

ウ：‘COL1’と‘COL2’の2つの属性で結合しているため、同じ結果にはならない。誤り。

エ：左外部結合なので、‘TBL1’（左側の表）に存在しているものは‘TBL2’に存在しなくても抽出してしまう。誤り。

1.3 CREATE

平成 22 年度・D B・午前Ⅱ

問20 関係データベース管理システム（RDBMS）のデータディクショナリに格納されるものはどれか。

- ア OS が管理するファイルの定義情報
- イ スキーマの定義情報
- ウ 表の列データの組
- エ 表の列に付けられたインデックスの内容

平成 30 年度・D B・午前Ⅱ

問1 SQL における BLOB データ型の説明として、適切なものはどれか。

- ア 全ての比較演算子を使用できる。
- イ 大量のバイナリデータを格納できる。
- ウ 列値でソートできる。
- エ 列値内を文字列検索できる。

問 20：正解イ

データディクショナリに関する問題。データディクショナリとは、データベースを管理するための様々な情報、すなわちスキーマの定義情報を格納した特別な表のこと、具体的には次のような情報を格納している。したがって、正解はイになる。

- ・すべてのオブジェクトに関する情報（オブジェクト名称、オーナ、作成日付等）
- ・すべてのユーザに関する情報（ユーザ名、作成日付等）
- ・スキーマ・オブジェクトに割り当てられている領域と、現在使用されている領域の容量
- ・整合性制約に関する情報

問 1：正解（イ）

データ型に関する問題。BLOB (Binary Large Object) データ型とは、略字をフルスペルに戻すとわかりやすいが、大きな (= Large) バイナリ (= Binary) データを格納することができるデータ型になる。したがって正解は選択肢イとなる。

ア：BLOB データ型は、プログラムや動画・音声など任意のバイナリデータを格納するものなので、等号での比較はできるものの、その列値の大小が判断できるものではないので、大小比較などの比較演算子は使用できない。

ウ：列値のバイナリデータに“順番”は無いので意味はないし、通常はソートはできない。

エ：バイナリデータは文字ではないので文字列検索という概念がない。

問3 表 R に、(A, B) の 2 列でユニークにする制約 (UNIQUE 制約) が定義されているとき、表 R に対する SQL 文でこの制約の違反となるものはどれか。ここで、表 R には主キーの定義がなく、また、すべての列は値が決まっていない場合 (NULL) もあるものとする。

R

A	B	C	D
AA01	BB01	CC01	DD01
AA01	BB02	CC02	NULL
AA02	BB01	NULL	DD03
AA02	BB03	NULL	NULL

- ア DELETE FROM R WHERE A = 'AA01' AND B = 'BB02'
- イ INSERT INTO R VALUES ('AA01', NULL, 'DD01', 'EE01')
- ウ INSERT INTO R VALUES (NULL, NULL, 'AA01', 'BB02')
- エ UPDATE R SET A = 'AA02' WHERE A = 'AA01'

問45 DBMS の表において、指定した列に NULL 値の入力は許すが、既に入力されている値の入力は禁止する SQL の制約はどれか。

- | | |
|--------------|---------------|
| ア CHECK | イ PRIMARY KEY |
| ウ REFERENCES | エ UNIQUE |

問3：正解工

DBMSの整合性制約に関する問題。UNIQUE制約（参照 ➔ 3.2.1 CREATE TABLE）について問われている。問題文の条件を念頭において、選択肢を順番に見ていく。

- ア：表Rの2行目が削除されるだけで問題はない。
- イ：表Rに、追加しようとしているタプルと同じ(A, B)の組合せ('AA01', NULL)は存在しないので追加できる。問題なし。
- ウ：表Rに、追加しようとしているタプルと同じ(A, B)の組合せ(NULL, NULL)は存在しないので追加できる。問題なし。
- エ：A = 'AA01' のタプルは2件存在する。この2件の属性Aを 'AA02' に更新すると、'AA02', 'BB01' がすでに存在するので、UNIQUE制約に反する。したがって、これが正解になる。

問45：正解工

- ア：CHECK制約。列に入力できる条件を設定する。
- イ：主キー制約。
- ウ：外部キー制約。

問 7 商品情報に価格、サイズなどの管理項目を追加する場合でもスキーマ変更を不要とするために、“管理項目”表を次の SQL 文で定義した。“管理項目”表の“ID”は商品ごとに付与する。このとき、同じ ID の商品に対して、異なる商品名を定義できないようにしたい。a に入れる字句はどれか。

管理項目

ID	項目名	データ型	値
1	商品名	文字列	ライト 01
1	商品番号	文字列	L001
1	価格	数値	400
2	商品名	文字列	ノート 02
2	:	:	:

〔商品情報〕

ID	商品名	商品番号	価格	サイズ
1	ライト 01	L001	400	
2	ノート 02	N001	120	A4
			:	

〔SQL 文〕

```
CREATE TABLE 管理項目 (
    ID          INTEGER NOT NULL,
    項目名      VARCHAR(20) NOT NULL,
    データ型    VARCHAR(10) NOT NULL,
    値          VARCHAR(100) NOT NULL,
    a
)
```

ア UNIQUE (ID)

イ UNIQUE (ID, 項目名)

ウ UNIQUE (ID, 項目名, 値)

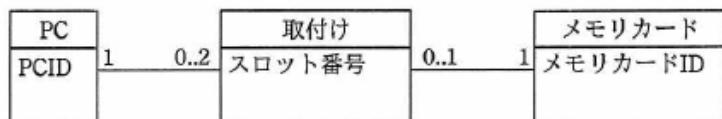
エ UNIQUE (項目名, 値)

問7：正解（イ）

DBMSの整合性制約に関する問題。UNIQUE制約について問われている。問題文の条件を念頭において、選択肢を順番に見ていく。

- ア：表“管理項目”を見れば明らかだが、属性‘ID’は UNIQUE ではない。不適切。
- イ：(ID, 項目名) に UNIQUE 制約を付与すると、例えば (1, 商品名) が 2 件登録できなくなるため、問題文にある「同じ ID の商品に対し、異なる商品名を定義できないようにしたい。」という要求を実現できる。したがってこれが正解になる。
- ウ：(ID, 項目名, 値) に UNIQUE 制約を付与すると、この 3 つの属性の組合せで一意になればいいわけだから、例えば (1, 商品名, ライト 01) と (1, 商品名, ノート 02) が登録できるので、「異なる商品名が定義」できてしまう。不適切。
- エ：(項目名, 値) に UNIQUE 制約を付与するだけでは、例えば (商品名, ライト 01) と (商品名, ノート 02) などは普通に登録できる。これで同じ属性‘ID’を両方に付与すると、2 件「異なる商品名が定義」できてしまう。不適切。

問11 PC へのメモリカードの取付け状態を管理するデータモデルを作成した。1 台の PC は、スロット番号によって識別されるメモリカードスロットを二つ備える。“取付け”表を定義する SQL 文の a に入る適切な制約はどれか。ここで、モデルの表記には UML を用いる。



[SQL 文]

```

CREATE TABLE 取付け (
    PCID INTEGER NOT NULL FOREIGN KEY REFERENCES PC(PCID),
    スロット番号 INTEGER NOT NULL,
    メモリカード ID INTEGER NOT NULL
        FOREIGN KEY REFERENCES メモリカード(メモリカード ID),
    a
    CHECK(スロット番号 IN (1, 2))
)
  
```

- ア PRIMARY KEY(PCID, スロット番号),
- イ PRIMARY KEY(PCID, スロット番号, メモリカード ID),
- ウ PRIMARY KEY(PCID, スロット番号),
UNIQUE(メモリカード ID),
- エ PRIMARY KEY(スロット番号, メモリカード ID),
UNIQUE(PCID),

問 11：正解（ウ）

(問題文に図示されている) “取付け” の CREATE 文を完成させる問題。“取付け” には “PC” に対する外部キーと, “メモリカード” に対する外部キーの二つの属性を持っている。

また, 問題文には「スロット番号によって識別されるメモリカードスロットを二つ備える。」という記述があるものの, “PC” からみた “取付け” は, 0 ~ 2 の範囲なので, メモリカードが装着されていないスロットに関しては, “取付け” のインスタンスは存在しないと考えられる。これは “取付け” インスタンスには必ず “メモリカード” が一つ存在することからも確認できるし, “取付け” の “PC” に対する外部キーも “メモリカード” に対する外部キーも “NOT NULL” 制約が付いていることからも間違いないと判断できるだろう。

選択肢を見れば, 主キーをどうするのかが問われていることがわかる。

スロット番号は, 検査制約 (CHECK (スロット番号 IN (1, 2))) より, “1” もしくは “2” なので ‘スロット番号’ だけを主キーにはできない。また, このスロット番号の “1” と “2” は, 問題文より PC 単位のものなので, 主キーは [PCID, スロット番号] になる。以上より, 選択肢アか選択肢ウが正解になる。

後は, 選択肢アと選択肢ウの違い, メモリカード ID に UNIQUE 制約が必要か否かだ。常識的に考えれば “同じメモリカード” がいろいろな PC に取付けられているようにも思えるが, この E - R 図だと, “メモリカード” と “取付け” が 1 対多ではなく, 1 対 1 の関係になっている。そのため, ひとつの ‘メモリカード ID’ に対して, “取付け” は (ゼロもしくは) “1” になり, 同じ ‘メモリカード ID’ が “取付け” の中に存在してはいけないことになる。したがって UNIQUE 制約が必要になるので正解はウになる。

問45 DBMS の整合性制約のうち、データの追加、更新及び削除を行うとき、関連するデータ間で不一致が発生しないようにする制約はどれか。

ア 形式制約

イ 更新制約

ウ 参照制約

エ 存在制約

問45 “社員”表、“受注”表からなるデータベースの参照制約について記述したものはどれか。

ア “社員”表の列である社員番号は、ドメインをもつ。

イ “社員”表の列である社員番号は、“社員”表の主キーである。

ウ “社員”表の列である社員名は、入力必須である。

エ “受注”表の列である受注担当社員番号は、外部キーである。

問45：正解ウ

DBMSの整合性制約に関する頻出問題。それぞれの整合性制約の意味をしっかりと理解しておこう。

参照 ➔ 「3.2.2 CREATE TABLE」

- ア：項目のデータ型や範囲に対する制約である
- イ：データの更新時に発生する制約~~こそ~~である
- ウ：関連するテーブル間の整合性を保つようにする制約が参照制約である。参照制約を指定していると、一方のテーブルでデータの追加、更新、削除を行ったときに関連するデータとの間で整合性を保つように働く。よってこれが正解
- エ：ある項目のデータが存在するためには、別の項目のデータが存在しなければならないという制約である

問45：正解エ

参照制約では、ある表の外部キーと、別の表の主キーが連携する。そこから選択肢はイかエに絞り込まれ、イは単に主キーを説明しているだけなのでエが正解になる。その他、ア～ウまではいずれも社員表のみの説明であり、複数表の関連性について言及しているものはエだけという点からも、解答を導くことができる。

参照 ➔ 「3.2.1 CREATE TABLE」

問17 SQLにおいて、A表の主キーがB表の外部キーによって参照されている場合、行を追加・削除する操作の制限について、正しく整理した図はどれか。ここで、△印は操作が拒否される場合があることを表し、○印は制限なしに操作できることを表す。

ア		追加	削除
A表	○	△	
B表	△	○	

イ		追加	削除
A表	○	△	
B表	○	△	

ウ		追加	削除
A表	△	○	
B表	○	△	

工		追加	削除
A表	△	○	
B表	△	○	

問43 関係データベースの“注文”表と“注文明細”表が、次のように定義されている。

“注文”表の行を削除すると、対応する“注文明細”表の行が、自動的に削除されるようにしたい。この場合、SQL 文に指定する語句として、適切なものはどれか。ここで、表定義中の実線の下線は主キーを、破線の下線は外部キーを表す。

注文		
注文番号	注文日	顧客番号

注文明細		
注文番号	商品番号	数量

ア CASCADE

イ INTERSECT

ウ RESTRICT

工 SET NULL

問 17：正解ア

過去問題（平成13年・DB午前問45など）

参考▶「3.2.1 CREATE TABLE」

SQL等の参照制約に関する問題。問題文に記述されている関連性を整理してみる。ここでも次のようなシンプルな例を使って具体的にイメージしてトレースするとわかりやすいかもしれません。

- A表 顧客（顧客番号, 顧客名）
 B表 注文（注文番号, 日付, 顧客番号, …）

- ① この場合、顧客（A表）に関しては、
 追加…制限はない。B表に関係なく可能（○）
 削除…B表に存在する顧客番号は削除できないようになることが可能（△）
 ② また、注文（B表）に関しては、
 追加…A表に存在しない顧客番号は追加できないようになることが可能（△）
 削除…制限はない。A表に関係なく可能（○）

以上より、正解はアになる。

問 43：正解ア

SQLで参照制約を定義する場合、外部キー（注文明細テーブルの注文番号）と、その参照先（注文テーブル）の関係で、参照先の行（注文テーブルのある注文番号）を削除した場合、外部キーを指定している従属テーブルのレコード（注文明細テーブルの同一注文番号のデータ）を、（連動して）自動的に削除するように設定する場合、CASCADEを指定する。

イ：積集合を表す演算子。

ウ：従属テーブルに行が存在しないときのみ削除を行う。

エ：従属テーブルに行が存在する場合、外部キーを指定している列にNULL値を設定する。

問44 事業本部制をとっている A 社で、社員の所属を管理するデータベースを作成することになった。データベースは表 a, b, c で構成されている。新しいデータを追加するときに、ほかの表でキーになっている列の値が、その表に存在しないとエラーとなる。このデータベースに、各表ごとにデータを入れる場合の順序として、適切なものはどれか。ここで、下線は各表のキーを示す。

表 a	<u>社員番号</u>	氏名	事業本部コード	部門コード
-----	-------------	----	---------	-------

表 b	<u>事業本部コード</u>	事業本部名
-----	----------------	-------

表 c	<u>事業本部コード</u>	<u>部門コード</u>	部門名
-----	----------------	--------------	-----

ア 表 a → 表 b → 表 c

イ 表 a → 表 c → 表 b

ウ 表 b → 表 a → 表 c

エ 表 b → 表 c → 表 a

問 7 SQL におけるドメインに関する記述のうち、適切なものはどれか。

ア 基底表を定義するには、ドメインの定義が必須である。

イ ドメインの定義には CREATE 文、削除には DROP 文を用いる。

ウ ドメインの定義は、それを参照する基底表内に複製される。

エ ドメイン名は、データベースの中で一意である必要はない。

問 44：正解工

表 a, b, c から外部キーを明確にして、表間の依存関係を把握できれば解答が得られる。表間の依存関係は下図のようになる。外部キーのないのが表 b なのでこれが最初になる。表 b が作成されると、表 c の作成が可能になる。最後が表 a。よって正解はエになる。

参照 ➔ 「3.2.2 CREATE TABLE」



問 7：正解(イ)

完全 NEW

参照 ➔ 「3.2.1 CREATE TABLE」

SQLにおけるドメインに関する問題。データベース試験で“ドメイン”という言葉はよく出てくるが、SQLにおけるドメインというと、CREATE DOMAINとして使用されるものになる。そして、そのCREATE DOMAINは、例えば“男だけ”とか、“取引先限定”とか、ある属性の値の集合体や範囲を定義するときに使う。その名の通り“(取りうる値の)領域”というわけだ。それを前提知識として、選択肢を順番に見ていく。

ア：基底表とは、その名の通り“ベースになる表”的ことで、他の表から導出も派生もない表ということになる。その定義は、CREATE TABLEで行うが、その中で、必ずドメインの定義を使わないといけないというわけじゃない。誤り。

イ：ドメインの定義にはCREATE DOMAINを使い、削除するときはDROP DOMAINを使用する。正解。ちなみに、ドメインの定義を変更する場合はALTER DOMAINを使用する。

ウ：ドメインを定義したら、そのまま“ドメイン”として管理される。基底表内に複製されない。誤り。

エ：ドメイン名は、データベース（スキーマ）の中で、一意でなければならない。誤り。

問1 SQL のシーケンスに関する説明のうち、適切なものはどれか。

- ア 現在基底値が $0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow \dots$ のように、周期的に繰り返すシーケンスを定義することはできない。
- イ シーケンスの現在基底値を、最小値と最大値との間の任意の整数値に変更することはできない。
- ウ シーケンスの増分を、負の整数値にすることはできない。
- エ トランザクション開始後のシーケンスの値の取得による現在基底値の変更は、トランザクションのロールバックでその変更を取り消すことができない。

問22 関係データベースの利用において、仮想の表（ビュー）を作る目的として、適切なものはどれか。

- ア 記憶容量を節約するため
- イ 処理速度を向上させるため
- ウ セキュリティを向上させるためや表操作を容易にするため
- エ デッドロックの発生を減少させるため

問 1：正解（工）

SQLに関する問題。シーケンス（SEQUENCE）について問われている。SQLのシーケンスとは、主にID等の自動採番に使われる連番生成オブジェクトである。“CREATE SEQUENCE”文でシーケンスを作成する。この時、開始値や増分（負の整数値を指定すれば減少させることも可能）、上限値や下限値などを設定できる。CYCLEオプションを指定すれば循環させることも可能である。そして、シーケンスを作成した後、CREATE TABLE文でテーブルを作成する際に、特定の属性のデフォルト値としてデータベースごとのシーケンス取得関数（例：標準SQLではNEXT VALUE FOR シーケンス名、PostgreSQLではNEXTVAL（' シーケンス名 '）、Oracleではシーケンス名.NEXTVALなど）を指定することで、シーケンスを利用することができる。

- ア：CYCLEオプションを指定すれば、上限値（MAXVALUE）に達したときに最小値（MINVALUE）に戻るように設定できる。したがって誤り。
- イ：ALTER SEQUENCEを使えば、現在のシーケンスの値を任意の整数値に変更できる。したがって誤り。
- ウ：CREATE SEQUENCEのINCREMENT BYに、負の整数値を指定すれば減少させることも可能。したがって誤り。
- エ：正しい。シーケンスで管理される連番はトランザクションの影響を受けない。障害が発生してROLLBACKしても、取得されたシーケンスの値は元に戻らず、次回のNEXTVALはロールバックされた値の続きの番号を返す。ROLLBACKによってシーケンスの値を元に戻すことはできない。

問 22：正解ウ

ビューを使った場合、結果的に補助記憶装置の記憶容量を節約できたり、処理速度が実表を使ったときよりも向上したりするかもしれないが、それらが主目的ではない。最大の目的は、表操作や表のメンテナンスを含む管理を容易にしたり、セキュリティを向上させるために利用される。よって正解はウになる。デッドロックの発生を減少させることとは無関係。

参照 ➔ 「3.2.3 CREATE VIEW」

問31 四つの表“注文”，“顧客”，“商品”，“注文明細”がある。これらの表から、次のビュー“注文一覧”を作成する SQL 文はどれか。ここで、下線の項目は主キーを表す。

注文（注文番号，注文日，顧客番号）

顧客（顧客番号，顧客名）

商品（商品番号，商品名）

注文明細（注文番号，商品番号，数量，単価）

注文一覧

注文番号	注文日	顧客名	商品名	数量	単価
001	2006-01-10	佐藤	AAAA	5	5,000
001	2006-01-10	佐藤	BBBB	3	4,000
002	2006-01-15	田中	BBBB	6	4,000
003	2006-01-20	高橋	AAAA	3	5,000
003	2006-01-20	高橋	CCCC	10	1,000

ア CREATE VIEW 注文一覧

```
AS SELECT * FROM 注文, 顧客, 商品, 注文明細
      WHERE 注文.注文番号 = 注文明細.注文番号 AND
            注文.顧客番号 = 顧客.顧客番号 AND
            商品.商品番号 = 注文明細.商品番号
```

イ CREATE VIEW 注文一覧

```
AS SELECT 注文.注文番号, 注文日, 顧客名, 商品名, 数量, 単価
      FROM 注文, 顧客, 商品, 注文明細
      WHERE 注文.注文番号 = 注文明細.注文番号 AND
            注文.顧客番号 = 顧客.顧客番号 AND
            商品.商品番号 = 注文明細.商品番号
```

ウ CREATE VIEW 注文一覧

```
AS SELECT 注文.注文番号, 注文日, 顧客名, 商品名, 数量, 単価
      FROM 注文, 顧客, 商品, 注文明細
      WHERE 注文.注文番号 = 注文明細.注文番号 OR
            注文.顧客番号 = 顧客.顧客番号 OR
            商品.商品番号 = 注文明細.商品番号
```

エ CREATE VIEW 注文一覧

```
AS SELECT 注文.注文番号, 注文日, 商品名, 数量, 単価
      FROM 注文, 商品, 注文明細
      WHERE 注文.注文番号 = 注文明細.注文番号 AND
            商品.商品番号 = 注文明細.商品番号
```

問31：正解イ

ビューの SELECT 文に関する問題。ビュー “注文一覧” から次の順に選択肢を絞り込んでいく。

参照 ➔ 「3.2.3 CREATE VIEW」

ア : SELECT 文の列に “*” を使っているため誤り

イ : 正しい

ウ : WHERE に指定している四つの表の結合条件に “OR” を使っているため誤り

エ : 顧客名がないため、誤り

問10 ある月の“月末商品在庫”表と“当月商品出荷実績”表を使って、ビュー“商品別出荷実績”を定義した。このビューに SQL 文を実行した結果の値はどれか。

月末商品在庫

商品コード	商品名	在庫数
S001	A	100
S002	B	250
S003	C	300
S004	D	450
S005	E	200

当月商品出荷実績

商品コード	商品出荷日	出荷数
S001	2017-03-01	50
S003	2017-03-05	150
S001	2017-03-10	100
S005	2017-03-15	100
S005	2017-03-20	250
S003	2017-03-25	150

[ビュー“商品別出荷実績”の定義]

```
CREATE VIEW 商品別出荷実績 (商品コード, 出荷実績数, 月末在庫数)
AS SELECT 月末商品在庫.商品コード, SUM(出荷数), 在庫数
FROM 月末商品在庫 LEFT OUTER JOIN 当月商品出荷実績
ON 月末商品在庫.商品コード = 当月商品出荷実績.商品コード
GROUP BY 月末商品在庫.商品コード, 在庫数
```

[SQL 文]

```
SELECT SUM(月末在庫数) AS 出荷商品在庫合計
FROM 商品別出荷実績 WHERE 出荷実績数 <= 300
```

ア 400

イ 500

ウ 600

エ 700

問 10：正解（ア）

SQL – CREATE VIEW 文、及び SELECT 文 – の問題。二つの SQL 文を順番に解釈していくべきいい。

〔ビュー“商品別出荷実績”の定義〕を実行すると、下図のようになる。“LEFT OUTER JOIN”で左外結合しているので、“LEFT OUTER JOIN”的左側にある“月末商品在庫”表の5行はすべて保持される。また、出荷実績数は“当月商品出荷実績”表の商品コードごとの出荷数を合計したものになる。

次に〔SQL文〕を実行する。抽出条件が‘出荷実績数<=300’なので、下図の網掛部分、すなわち、商品コードが‘S001’と‘S003’だけが条件に合致する。この2つのタブルの‘月末在庫数’を合計すると400になる。したがって、正解は（ア）になる。なお、‘S002’と‘S004’は、外結合の段階で‘出荷実績数’にはNULLが設定されるので対象外になる。その点に注意しよう。

商品別出荷実績

商品コード	出荷実績数	月末在庫数
月末商品在庫. 商品コード	SUM(出荷数)	在庫数
S001	150	100
S002	NULL	250
S003	300	300
S004	NULL	450
S005	350	200

問10 更新可能なビューの定義はどれか。ここで、ビュー定義の中で参照する基底表は全て更新可能とする。

- ア CREATE VIEW ビュー1(取引先番号, 製品番号)
AS SELECT DISTINCT 納入.取引先番号, 納入.製品番号
FROM 納入
- イ CREATE VIEW ビュー2(取引先番号, 製品番号)
AS SELECT 納入.取引先番号, 納入.製品番号
FROM 納入
GROUP BY 納入.取引先番号, 納入.製品番号
- ウ CREATE VIEW ビュー3(取引先番号, ランク, 住所)
AS SELECT 取引先.取引先番号, 取引先.ランク, 取引先.住所
FROM 取引先
WHERE 取引先.ランク > 15
- エ CREATE VIEW ビュー4(取引先住所, ランク, 製品倉庫)
AS SELECT 取引先.住所, 取引先.ランク, 製品.倉庫
FROM 取引先, 製品
HAVING 取引先.ランク > 15

問10：正解（ウ）

SQL—更新可能なビューに関する問題（参照▶3.2.2 CREATE VIEW）。標準SQLでは、次の条件に合致する場合に更新可能になる。下記の③については問題文で“可能”となっているので、下記の①と②を中心に選択肢を順番にみていいだろ。

- ①基底表（元の実表）そのものが特定できること
- ②基底表（元の実表）の“行”が特定できること
 - 1. 集約関数（AVG, MAX 等）を含む場合はできない
 - 2. GROUP BY 句, HAVING 句を用いている場合もできない
 - 3. 重複値を排除する DISTINCT 句を用いている場合もできない
- ③~~そもそも~~基底表（元の実表）が更新可能のこと

ア：DISTINCT 句を用いているため、更新はできない。

イ：GROUP BY 句を用いているため、更新はできない。

ウ：更新可能

エ：HAVING 句を用いているため、更新はできない。

上記エの補足説明（2024.5.27）

- ・標準SQLでも「<GROUP BY 句>または<HAVING 句>を単純に含まない」としているので、選択肢エのように HAVING 句を用いている場合、更新可能なビューとは言えない。正解を得るだけなら、この判断基準で問題は無い。
- ・上記以前に、選択肢エのVIEW自体が成立しない可能性がある。VIEW内のSELECT文がGROUP BYを使わずに HAVING 句を使っている。これはNGではないものの、この場合、表全体を一つのグループとみなすことになる。そのため、SELECT文の選択項目リストに特定の列を指定することはできないと考えられる。実際に多くの製品でエラーになる。この場合は HAVING 句ではなく WHERE 句を使う必要がある。
- ・また、同ビューのSELECT文のFROM句には複数表が指定されている。WHEREで結合条件に該当する抽出条件を指定していないため直積になる（交差結合）。仮に HAVING を WHERE に変えたとしても、直積の場合ビューから対象行を特定できないので更新可能ではない。

問11 三つの表“取引先”，“商品”，“注文”を基底表とするビュー“注文 123”を操作する SQL 文のうち、実行できるものはどれか。ここで、各表の列のうち下線のあるものを主キーとする。

取引先

取引先 ID	名称	住所
111	中央貿易	東京都中央区
222	上野商会	東京都台東区
333	目白商店	東京都豊島区

商品

商品番号	商品名	価格
111	スパナ	1,000
123	レンチ	1,300
313	ドライバ	800

注文

注文番号	注文日	取引先 ID	商品番号	数量
1	2013-04-17	111	111	3
2	2013-04-18	222	123	4
3	2013-04-19	111	313	3
4	2013-04-20	333	123	2

[ビュー“注文 123”的定義]

```
CREATE VIEW 注文 123 AS
SELECT 注文番号, 取引先.名称 AS 取引先名, 数量
FROM 注文, 取引先, 商品
WHERE 注文.商品番号 = '123'
AND 注文.取引先 ID = 取引先.取引先 ID
AND 注文.商品番号 = 商品.商品番号
```

- ア DELETE FROM 注文 123 WHERE 取引先 ID = '111'
- イ INSERT INTO 注文 123 VALUES (8, '目白商店', 'レンチ', 3)
- ウ SELECT 取引先.名称 FROM 注文 123
- エ UPDATE 注文 123 SET 数量 = 3 WHERE 取引先名 = '目白商店'

問 11：正解（工）

過去問題（平成23年・DB午前Ⅱ問8）

参照 「3.2.2 CREATE VIEW」

SQL（第3章）－CREATE VIEWに関する問題。更新可能なビューに関する出題である。まずはSELECT文の実現可否をチェックしてみて、それが実行できそうになれば、他の選択肢をチェックすると良い。チェックする場合、今回のように問題文に実データがあれば、そのデータでVIEWを実行してみて、選択肢を一つずつチェックしていくけば効率が良いだろう。

ビュー“注文123”を実行すると、次のようになる。

注文番号	取引先名	数量
2	上野商会	4
4	目白商店	2

ア：取引先IDが“111”的ものをデリートしようとしているが、取引先IDという属性はビュー“注文123”にはない。したがって実行できない。

イ：まず、ビューの属性の数と追加したい属性の数が違う。それにおそらく数量のところは数値だろうから、そこに‘レンチ’という日本語を入れようとしている。そのようないろいろな点で、実行できないことがわかるだろう。

ウ：SELECT文なので最も実現可能性が高いが、今回は、残念ながら実行できないと考えられる。というのも、このSELECT文では“取引先.名称”を抽出しようとしているが、ビューでは、それをASを使って別名に変えている。その場合、別名の方を指定しなければならない。今回なら「SELECT 取引先名 FROM 注文123」とすべきだ。誤り。

エ：上記のビューの実行結果に対して、2件目の数量を“2”から“3”に更新している。このとき、実際に変更されるのは“注文”表の注文番号4（問題文の4件目）の数量が変わるだけなので実行可能。これが正解になる。

問11 ビューの SELECT 権限に関する記述のうち、適切なものはどれか。

- ア ビューに対して問合せをするには、ビューに対する SELECT 権限だけではなく、元の表に対する SELECT 権限も必要である。
- イ ビューに対して問合せをするには、ビューに対する SELECT 権限又は元の表に対する SELECT 権限のいずれかがあればよい。
- ウ ビューに対する SELECT 権限にかかわらず、元の表に対する SELECT 権限があれば、そのビューに対して問合せをすることができる。
- エ 元の表に対する SELECT 権限にかかわらず、ビューに対する SELECT 権限があれば、そのビューに対して問合せをすることができる。

問7 体現ビュー（Materialized view）に関する記述のうち、適切なものはどれか。

- ア 同じデータが実表と体現ビューとに重複して格納されることはない。
- イ 更新可能であると DBMS が判断したビューのことである。
- ウ 実表のようにデータベースに格納されるビューのことである。
- エ 問合せや更新要求のたびにビュー定義を SQL 文に組み込んで処理する。

問 11：正解工

SQL - ビューの権限に関する問題（参照 ➔ 3.2.2 CREATE VIEW）。ビューを使用するときの SELECT 権限は、ビューに対する権限だけがあればいい。したがって、正解はエになる。

問 7：正解（ウ）

(完全 NEW)

体現ビュー(Materialized view)に関する問題。体現ビューとは、実テーブルと同じように(ビューの) 内容を保持するビューのことである。通常のビューが、利用する都度実行されるのに対し、体現ビューは、作成時やリフレッシュ時（元表が変わった場合に、どのタイミングで変更するかは別途決めるが、この動作をリフレッシュという）におけるビューの内容を保持する（レプリカを作成する）。これにより、ビューの高速化が可能になる。以上を前提に、選択肢を順番にみていく。

ア：同じデータがレプリカとして重複して格納される。誤り。

イ：この説明は、単に「更新可能なビュー」のことである。誤り。

ウ：これが体現ビューの説明である。正解。

エ：通常のビューの処理。誤り。

問12 導出表に関する記述として、適切なものはどれか。

- ア 算術演算によって得られた属性の組である。
- イ 実表を冗長にして利用しやすくする。
- ウ 導出表は名前をもつことができない。
- エ ビューは導出表の一つの形態である。

問 12：正解（工）

導出表に関する問題。導出表とは、実表から関係データベースの操作によって“導出”される仮想表のこと。その代表的なものに“ビュー”がある。したがって選択肢エが正解になる。

- ア：他の属性から算術演算によって得られる項目を導出属性や導出項目という。導出表とはそうした導出属性ばかりを集めた表のことではない。誤り。
- イ：実表を分散データベース上に持たせたりすることで、冗長で利用しやすくすることはできるが、それを導出表とは言わない。誤り。
- ウ：導出表に新たに名前を付けることも、その名前で利用することもできる。誤り。

1.4 権限

平成 22 年度・D B・午前Ⅱ

問 2 表の所有者が、SQL 文の GRANT を用いて設定するアクセス権限の説明として、適切なものはどれか。

- ア パスワードを設定してデータベースの接続を制限する。
- イ ビューによって、データベースへのアクセス処理を隠ぺいし、表を直接アクセスできないようにする。
- ウ 表のデータを暗号化して、第三者がアクセスしてもデータの内容が分からないようにする。
- エ 表の利用者に対し、表への問合せ、更新、追加、削除などの操作を許可する。

平成 21 年度・D B・午前Ⅱ

問 7 次の SQL 文の実行結果の説明として、適切なものはどれか。

```
CREATE VIEW 東京取引先 AS
    SELECT * FROM 取引先
    WHERE 取引先.所在地 = '東京'
GRANT SELECT
    ON 東京取引先 TO "8823"
```

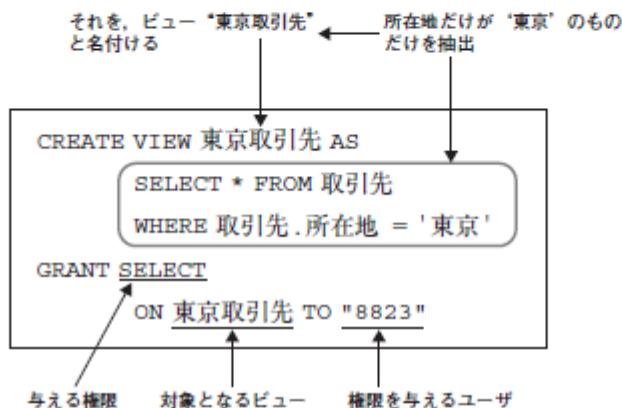
- ア 8823 のユーザは、所在地が“東京”の行を参照できるようになる。
- イ このビューの作成者は、このビューに対する SELECT 権限をもたない。
- ウ 実表“取引先”が削除されても、このビューに対するユーザの権限は残る。
- エ 導出表“東京取引先”には、8823 行までを記録できる。

問2：正解工

表に対するアクセス権限に関する問題。GRANT（参照 ➤ 3.3.1 GRANT）の利用シーンについて問われている。GRANT文を使用するのは、「表の所有者が、表の利用者に権限を与えるとき」である。したがって、正解はエになる。

問7：正解ア

SQL文のビューの構文（参照 ➤ 3.2.2 CREATE VIEW）とGRANT文に関する問題。まずは、以下のようにSQL文を解析する。その上で、選択肢を順番に見ていく。



ア：GRANTの構文（参照 ➤ 3.3.1 GRANT）をそのまま解釈するとこの意味になる。正解である。

イ：ビューの作成者（所有者）は、そのビューに対するSELECT権限を持つ。よって誤りである。

ウ：今回のようにビューの権限を作成者から付与されたユーザは、元の表の権限に関係なくビューに対する権限の有無だけで判断できるようになるが、それも、実表が存在していることが前提である。よって誤りである。

エ：GRANTの構文の意味、使い方が全く違う。よって誤りである。

問30 SQL におけるオブジェクトの処理権限に関する記述のうち、適切なものはどれか。

- ア 権限の種類は INSERT, DELETE, UPDATE の三つである。
- イ 権限は実表だけに適用でき、ビューには適用できない。
- ウ 権限を取り上げるには REVOKE 文を用いる。
- エ 権限を付与するには COMMIT 文を用いる。

問 30：正解ウ

ア：SELECTなどもあるため、この3つだけではない。

イ：権限の適用はビューにもできる。

エ：権限の付与はGRANTで行う。

1.5 プログラム言語における SQL 文

平成 25 年度・D B・午前Ⅱ

問8 SQL で用いるカーソルの説明のうち、適切なものはどれか。

- ア COBOL, C などの親言語内では使用できない。
- イ 埋込み型 SQL において使用し、会話型 SQL では使用できない。
- ウ カーソルは検索用にだけ使用可能で、更新用には使用できない。
- エ 検索処理の結果集合が單一行となる場合の機能で、複数行の結果集合は処理できない。

平成 16 年度・D B・午前

問34 埋込み SQL に関する記述として、適切なものはどれか。

- ア INSERT を実行する前に、カーソルを OPEN しておかなければならない。
- イ PREPARE は与えられた SQL 文を実行し、その結果を自分のプログラム中に記録する。
- ウ SQL では一度に 0 行以上の集合を扱うのに対し、親言語では通常一度に 1 行のコードしか扱えないので、その間をカーソルによって橋渡しする。
- エ データベースとアプリケーションプログラムが異なるコンピュータ上にあるときは、カーソルによる 1 行ごとの伝送が効率的である。

問8：正解（イ）

過去問題（平成20年・口述 午前 問35）

参照▶「3.6.1 カーソルの使用方法」

SQLのカーソルに関する問題。カーソルとは、現在処理しているデータ（行）の箇所をさすマークのこと。COBOLやCなどの親言語内でSQL（特に、埋込み型SQLという）を使用するときに使う。逆に、会話型SQLでは使えない。以上より、正解は（イ）になる。

ア：上記の理由から誤りである。

ウ：カーソルを使って検索も更新も可能なので、誤りである。

エ：検索処理の結果集合をビューとした場合、それが單一行でも、複数行でもカーソル処理することは可能。誤りである。

問34：正解ウ

埋め込みSQLとは、他の言語にSQLを組み込むことで、カーソルを使用する。SQLでは通常複数行を返すことができるが、SQLを組み込む親言語では、1行ずつしか処理できないことが多い。そのためカーソルを使用する。したがって正解はウとなる。

問33 次の SQL 文は、 COBOL プログラムでテーブル A のレコードを読み込むためにカーソル宣言をしている。a に入れるべき適切な語句はどれか。

a
SELECT * FROM A
ORDER BY 1, 2
END-EXEC

- ア EXEC SQL DECLARE C1 CURSOR FOR
- イ EXEC SQL DECLARE CURSOR FOR C1
- ウ EXEC SQL OPEN CURSOR C1 FOR
- エ EXEC SQL OPEN CURSOR DECLARE C1 FOR

問6 次の SQL 文は、 A 表に対するカーソル B のデータ操作である。a に入れる字句はどれか。

UPDATE A
SET A2 = 1, A3 = 2
WHERE a

ここで、 A 表の構造は次のとおりであり、 実線の下線は主キーを表す。

A (A1, A2, A3)

- | | |
|-----------------|------------------|
| ア CURRENT OF A1 | イ CURRENT OF B |
| ウ CURSOR B OF A | エ CURSOR B OF A1 |

問 33：正解ア

カーソルを宣言するための構文は以下のようになる。

DECLARE カーソル名 CURSOR FOR

したがって、正解はアである。 [参照 ➔ 「3.5.1 カーソルの使用方法」](#)

問 6：正解（イ）

SQL のカーソルに関する問題。カーソルを用いた位置設定による UPDATE 文の構文は下記のようになる。ここで、カーソル名は “B” なので、正解はイになる。

```
UPDATE 表名  
SET 代入コマンドリスト  
WHERE CURRENT OF カーソル名
```

問12 SQL トランザクション内で変更を部分的に取り消すために設定するものはどれか。

ア コミットポイント

イ セーブポイント

ウ 制約モード

エ チェックポイント

問 12：正解（イ）

トランザクションに関する問題。SQL トランザクション内で変更を部分的に取り消すために設定するのはセーブポイント（SAVEPOINT）である。したがって正解はイになる。

通常、ひとつのトランザクション処理は、その中の処理すべてを実行するか、すべてを取り消すかのいずれかである。そもそも、トランザクション処理がそのためのものだからだ。一連のトランザクション処理の実行に成功するとコミットを発行して確定させ、失敗すればロールバック処理で全ての処理を取り消す。

しかし、時に、一連のトランザクション処理に多くの命令が含まれていたり、複雑なケースだったりして全ての処理を取り消したくない場合（ゆえに一部だけを取り消したい場合）がある。そういう時に使うのがセーブポイントである。トランザクション処理の中にセーブポイントを設定しておけば、その後のロールバック処理で、そのセーブポイント以後の処理だけを取り消すことができる。

ア：コミットは、それまでの処理を確定させるものである。

ウ：トランザクションにおける制約モードは、制約検査のタイミングを設定するモード（同時に検査するか、コミットのタイミングで検査するかを設定する）である。SET CONSTRAINTS で指定する。

エ：コミット処理では、必ずしもストレージのデータファイルそのものを更新しているわけではない。データベースバッファやログファイルだけを更新している場合もある。そういうケースで、ストレージのデータファイルを更新しているポイントをチェックポイントという。DBMS が管理している。

memo

2

第2章

概念データモデル

最初に、概念データモデル（下図）について説明する。概念データモデルとは、対象世界の情報構造を抽象化して表現したものである。データベースの種類にも、特定のDBMS製品にも依存せず、情報化しない範囲まで対象範囲とするのが特徴。情報処理技術者試験では、午後Ⅱ事例解析試験で必ず登場しており、E-R図で表現されている。午後Ⅱ対策は、ここからスタートしよう。

2.1 情報処理試験の中の概念データモデル

令和5年度・D B・午前Ⅱ

問3 概念データモデルの説明として、最も適切なものはどれか。

- ア 階層モデル、ネットワークモデル、関係モデルがある。
- イ 業務プロセスを抽象化して表現したものである。
- ウ 集中型DBMSを導入するか、分散型DBMSを導入するかによって内容が変わる。
- エ 対象世界の情報構造を抽象化して表現したものである。

問3：正解（工）

概念データモデルに関する問題。選択肢のうち、概念データモデルの説明になるのは、選択肢エである。

ア：これら（階層モデル、ネットワークモデル、関係モデル）は、論理データモデルになる。

イ：業務プロセスを抽象化したものは論理モデルという。

ウ：物理データモデルの説明である。

2.2 E-R 図（拡張 E-R 図）詳細解説

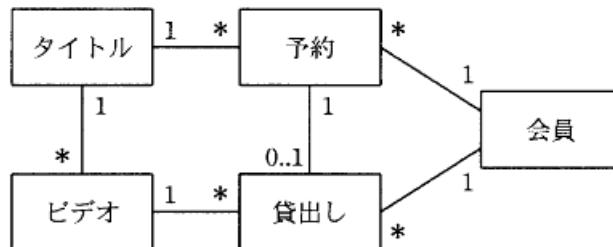
平成 18 年度・D B・午前

問17 UML を DFD 又は E-R 図と対比した記述のうち、適切なものはどれか。

- ア UML ではデータの関係を記述できないので、E-R 図を併用する必要がある。
- イ UML ではデータの流れを記述できないので、DFD を併用する必要がある。
- ウ UML におけるコラボレーション図（協調図）やコンポーネント図が DFD に相当する。
- エ UML における静的な構造を示すクラス図が、E-R 図に相当する。

平成 17 年度・D B・午前

問32 ビデオ貸出業務の概念データモデルにおける多重度の解釈として、適切なものはどれか。ここで、長方形はエンティティセットを表し、その中にその名称を記す。長方形間の線は関連を表す。関連の多重度を関連先の近傍に付す。多重度の下限値及び上限値を特に指定する場合は、“下限..上限” の形式とし、 “*” は 0 以上を示す。



- ア タイトルのないビデオも存在する。
- イ ビデオが存在していなくても予約ができる。
- ウ 予約したビデオは必ず貸し出される。
- エ 予約をしなくてもビデオは借りられる。

問 17：正解工

UML（[参照](#)用語集）、DFD（[参照](#)用語集）、E-R図（[参照](#)用語集）のそれぞれの特徴に関しての問題である。選択肢を順番に見ていく。

- ア：クラス図（[参照](#)用語集）を使うとデータの関係を記述できるので誤り
- イ：シーケンス図（[参照](#)用語集）やコラボレーション図（[参照](#)用語集）を使うとデータの流れ（メッセージ交換）を記述できるため誤り
- ウ：コンポーネント図（[参照](#)用語集）ではなく、シーケンス図の誤り
- エ：正しい

問 32：正解イ

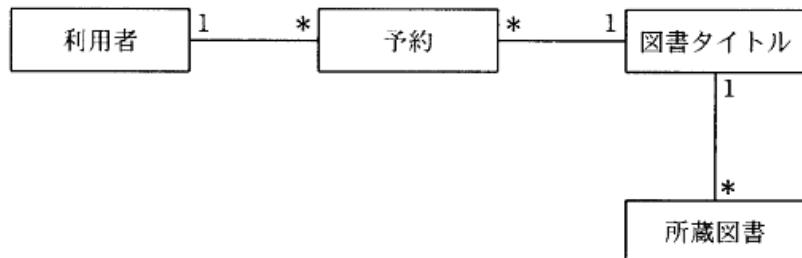
概念データモデルにおけるエンティティ間のカーディナリティの知識を問う問題。エンティティ間の多度を表すカーディナリティには、1対1、1対多、多対多の3種類がある。

[参照](#)「1.6.5 カーディナリティ」

- ア：タイトルとビデオは、1対0以上の関係があるため、ビデオのないタイトルは存在しても、タイトルのないビデオは存在しない。
- ウ：予約とビデオとの間に直接関連ではなく、貸出しを通じて予約とビデオの関連がつけられている。予約と貸出しの関連は、1対(0.1)のため、予約したビデオが必ず貸し出されるわけではない。
- エ：ウと同様の理由で、予約がなければビデオは借りられない。

問26 図書館の予約システムの一部について、次のようなデータモデルを作成した。この説明として適切なものはどれか。ここで、1*は、1 対多のカーディナリティを表し、表定義中の下線の付いた属性は、主キーを表す。

[予約システムのデータモデル]



利用者（利用者 ID, 利用者名, 住所）

予約（利用者 ID, 図書タイトル ID, 予約日）

図書タイトル（図書タイトル ID, 分類コード, 書名, 著者）

所蔵図書（所蔵図書 ID, 図書タイトル ID, 購入日, 累計貸出回数）

- ア 図書タイトルエンティティと所蔵図書エンティティの間のカーディナリティは逆である。
- イ 図書タイトルエンティティと所蔵図書エンティティを分けるのは冗長である。
- ウ 図書タイトルエンティティは、物理的な実体を伴わない抽象的なエンティティである。
- エ 予約時に貸し出す所蔵図書が決定できるようになっている。

問 26：正解ウ

ア：所蔵図書と図書タイトル間のカーディナリティは正しい。

イ：冗長ではない。

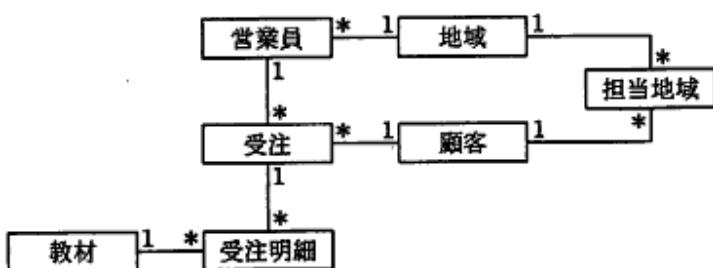
エ：特に決定できるようにはなっていない。

問4 業務ルールが次のように与えられている。これに基づき E-R 図を作成した。適切な E-R 図はどれか。ここで、1*は1対多の関連を表す。

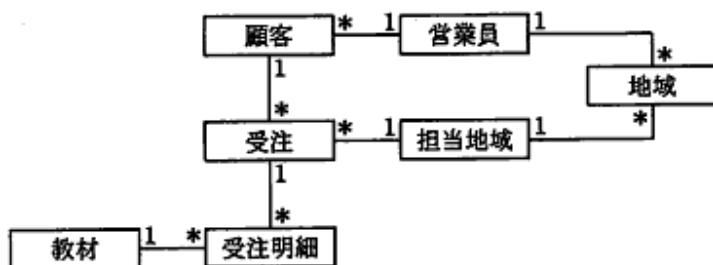
〔業務ルール〕

この会社は語学教材を販売している。営業員は一つ以上の担当地域が定められており、担当地域の 1 人以上の顧客から受注を得る。一つの地域を 1 人以上の営業員が担当する。

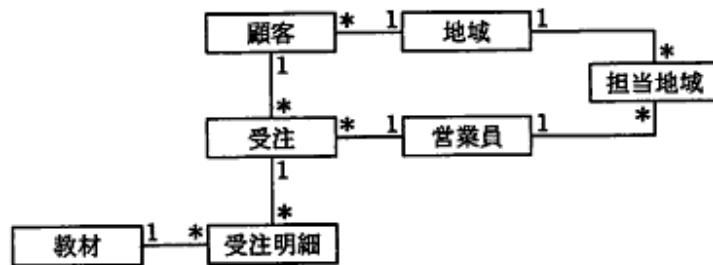
ア



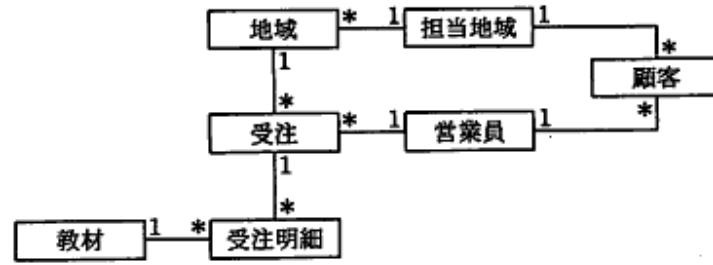
イ



ウ



エ

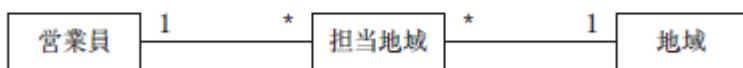


問4：正解ウ

E-R図に関する問題。選択肢を順番に見ていけば良い。選択肢の異なる点に着目すれば明白だが、選択肢の中にある関連の多重度はすべて同じである。また、“受注”，“受注明細”，“教材”的位置もすべて同じである。そこで，“顧客”，“地域”，“営業員”的関連だけに着目すれば良い。問題文の条件を読むと，“営業員”と“地域”は多対多の関係にあることがわかる。この関係をE-R図で図示すると、以下のようになる。



多対多の関係は、連関エンティティを用いて以下のように正規化していく。問題文には、それを“担当地域”と言っているため、その言葉を使う。

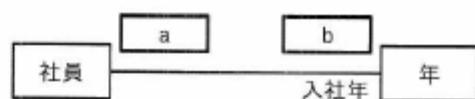


この関係を示しているのは選択肢ウだけになるので、正解はウになる。

問2 社員と年との対応関係をUMLのクラス図で記述する。二つのクラス間の関連が次の条件を満たす場合、a, bに入る多重度の適切な組合せはどれか。ここで、“年”クラスのインスタンスは毎年存在する。

[条件]

- (1) 全ての社員は入社年を特定できる。
- (2) 年によっては社員が入社しないこともある。



	a	b
ア	0..*	0..1
イ	0..*	1
ウ	1..*	0..1
エ	1..*	1

問 2：正解（イ）

多重度に関する問題。“社員”のインスタンスを一つ想定し、それに対する“年”インスタンスの数（の可能性）を考え、その後、同様に“年”的一つのインスタンスを想定し、それに対する“社員”インスタンスの数（の可能性）を考える。このとき、“0”, “1”, “2 以上”的 3 点を考えるのがポイントになる。

1人の社員に対して入社年がいくつあるのか、すなわち（b）の正解を考える。

条件（1）（2）に従えば、例えば、次のように考えられる。

- ・社員 A = 入社年 2001 年
- ・入社年度は特定できるので、2001 年と 2002 年など複数（2 以上）はない。
- ・「すべての社員が…特定できる」という表現は、「特定できない」ケースではなく、すなわち NULL。もしくは相手が“0”ということはない。

以上より、（b）の正解は「1」。

→選択肢イまたはエ

1つの入社年に対して社員が何人いるのか、すなわち（a）の正解を考える。

・「年クラスのインスタンスは毎年存在する」

（例）1999 年、2000 年、2001 年、…

・「年によっては社員が入社しないこともある」

例えば、1999 年は 0 人など、“年”的あるインスタンスに対する“社員”が 0 の場合あり

・選択肢は、すべて“.. *”

以上より、（a）の正解は「0.. *」→選択肢アまたはイ

以上より（a）と（b）の正解のどちらにもある選択肢イが正解になる。

問3 データモデリングの過程で、二つのエンティティ間に多対多の対応が生じた。これを関係データベース上に実装する場合、二つのエンティティを表として定義した上で、多対多の対応を表す方策はどれか。

- ア それぞれの表の主キーに対して明示的にインデックスを定義する。
- イ それぞれの表の主キーへの部分関数従属性を排除するように表を分解する。
- ウ それぞれの表の属性間の推移的な関数従属性を排除するように表を分解する。
- エ それぞれの表を参照する外部キーの組合せを主キーの一部とする表を新たに定義する。

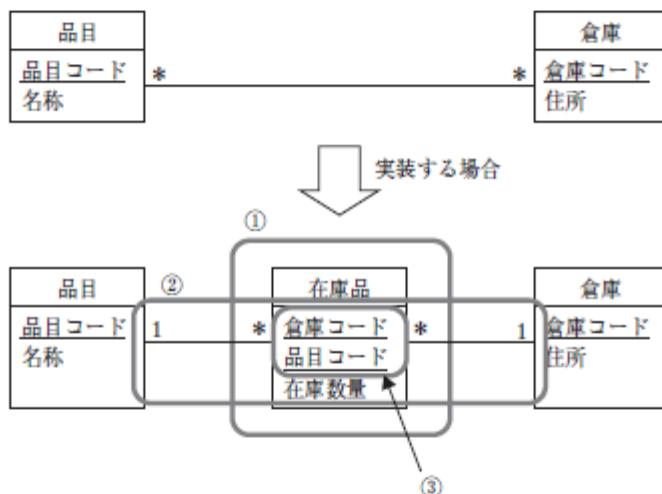
問3：正解（エ）

NEW
参照▶「1.2.2 多重度」

連関エンティティに関する基本的な問題。問題文のように、二つのエンティティ間に多対多の対応が生じたとき、関係データベース上に二つのエンティティだけで実装しようとすると少々都合が悪くなる。下図の例で説明すると、“品目”の中に複数の倉庫コードを属性として定義しないといけないが、その場合、倉庫コードは有限でなければならない。逆に“倉庫”の中に品目コードを属性として保持する場合も同じである。

【多対多の例】（下線部が主キー）

品目の中には、複数の倉庫に保管されているものがある。
一つの倉庫には、（0以上）複数の品目が保管されている。



そこで、二つのエンティティ間に多対多の対応が生じたとき、それを関係データベース上に実装する場合は、次のような対応になる。

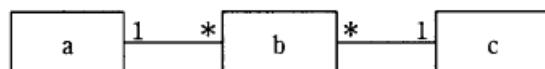
- ① 二つのエンティティ間に入る“新たな表”を一つ作成する（連関エンティティ）。
- ② その連関エンティティと既存の二表のリレーションシップは、いずれも“多対1”になる。
- ③ その連関エンティティの主キーは、二表の主キーを連結したものになり、それぞれが既存の二表への外部キーになる。

したがって、この対応を表現している選択肢（エ）が正解になる。

問31 部品在庫管理台帳における、部品、仕入先、在庫の三つのエンティティの関係を E-R 図として記述した。エンティティ a ~ c の組合せとして、適切なものはどれか。ここで、1 * は1対多のカーディナリティを表す。

部品在庫管理台帳

部品コード	部品名	仕入先コード	仕入先名	仕入日付	仕入価格	在庫数
001	R 部品	Z010	P 商会	9月1日	1,500	1,000
001	R 部品	Z010	P 商会	10月15日	1,400	1,500
002	S 部品	Z010	P 商会	9月20日	800	500
003	T 部品	Z015	Q 商店	10月8日	1,600	1,450



E-R図

	a	b	c
ア	在庫	仕入先	部品
イ	在庫	部品	仕入先
ウ	仕入先	部品	在庫
エ	部品	在庫	仕入先

問 31：正解工

まず、正規化してエンティティを抽出する。部品在庫管理台帳の主キーは、{部品コード, 仕入先コード, 仕入日付}である。主キーに仕入日付があるのは、同じ部品コードでも仕入日付によって、仕入価格が異なるためである。仕入先コード→仕入先名、(部品コード, 仕入日付) → {仕入価格, 在庫数}という部分関数従属性があるため、これを分解する。解答の選択肢から、エンティティは、“在庫”, “仕入先”, “部品”となるため、以下のようになる。

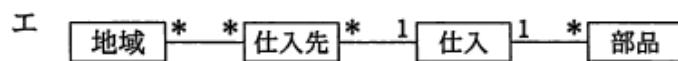
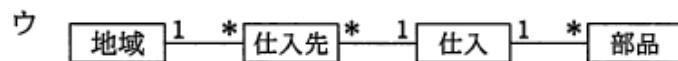
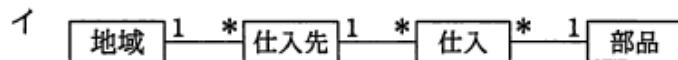
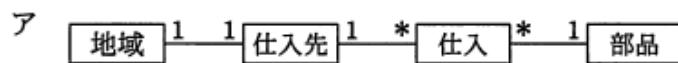
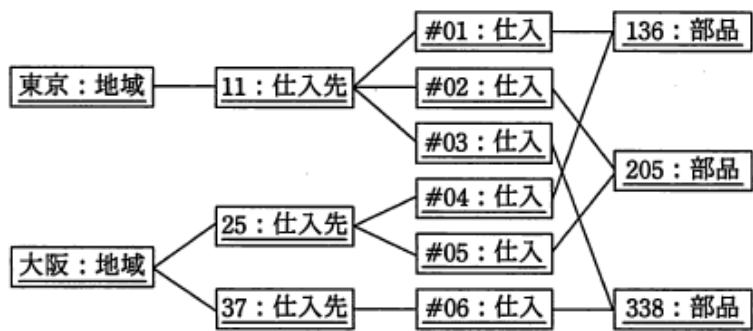
部品 (部品コード, 部品名)

仕入先 (仕入先コード, 仕入先名)

在庫 (部品コード, 仕入日付, 仕入価格, 在庫数)

これらの関連を考えると、部品と在庫の間に1対多、仕入先と在庫の間に1対多のカーディナリティがあることがわかる。したがって、正解はエとなる。 **参照** ➔ 「1.6.5 カーディナリティ」

問4 次のオブジェクト図（インスタンスを表す図）に対応する概念データモデルはどれか。ここで、オブジェクト図及び概念データモデルの表記には UML を用いる。



問4：正解イ

NEW

参照 「1.2 E-R図」

概念データモデルの表記方法に関する問題。モデルの表記にはUMLを用いるという制約がある点に注意。図を解釈しながら、選択肢を順番にチェックしていく。消去法が良いだろう。

<図より>

- ① “地域”と“仕入先”的関係
一つの“地域”に、“仕入先”は複数ある。選択肢アは誤り。
- ② “仕入先”と“仕入”的関係
一つの“仕入先”に、“仕入”は複数ある。選択肢ウ、エは誤り。
以上より、正解はイになるが、念のため、最後までチェックする。
- ③ “仕入”と“部品”的関係
一つの“部品”に、“仕入”は複数ある。選択肢イは正しい。

問4 四つの表の関係を表す E-R 図として、適切なものはどれか。ここで、実線の下線
は主キーを、破線の下線は外部キーを表す。

医師

<u>医師番号</u>	医師名	<u>診療科コード</u>
-------------	-----	---------------

診療科

<u>診療科コード</u>	診療科名称
---------------	-------

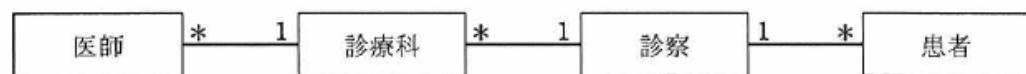
診察

<u>診療科コード</u>	<u>患者番号</u>	<u>診察日時</u>
---------------	-------------	-------------

患者

<u>患者番号</u>	患者名
-------------	-----

ア



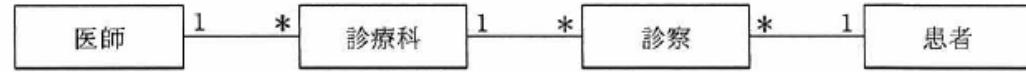
イ



ウ

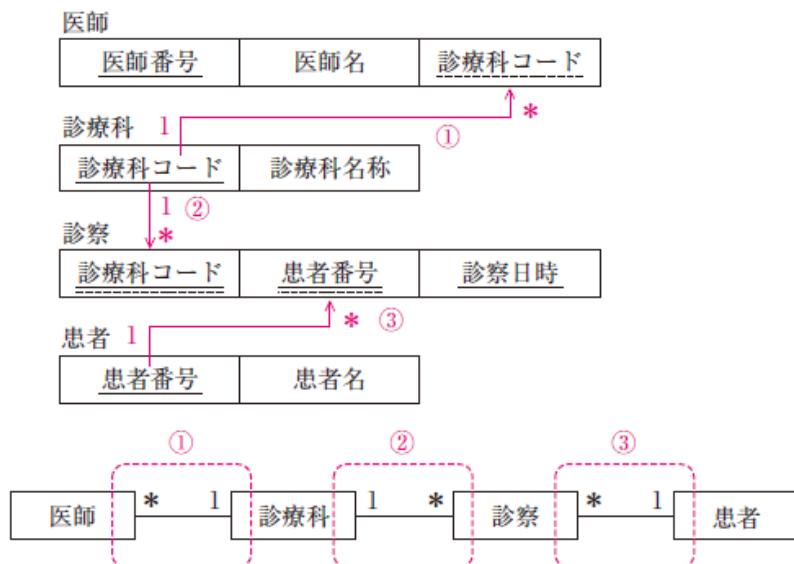


エ



問4：正解（イ）

E-R図に関する頻出問題。こういった問題は、図から関係を見出すと容易に解ける。まず、図から「主キーと外部キーの関係」をピックアップする。このとき「主キー側が“1”に、外部キー側が“*”になるので、それを図に加える（図参照）。これが、すなわちエンティティ間の関係になるので整理する。その結果、この関係にあるのはイになるため、イが正解になる。



問5 E-R モデルにおいて、実体 A のインスタンス a が他の実体 B のインスタンス b と関連しており、インスタンス a が存在しなくなれば、インスタンス b も存在しなくなる場合、このような実体 B を何と呼ぶか。

- ア 仮想実体 イ 強実体 ウ 弱実体 エ 正実体

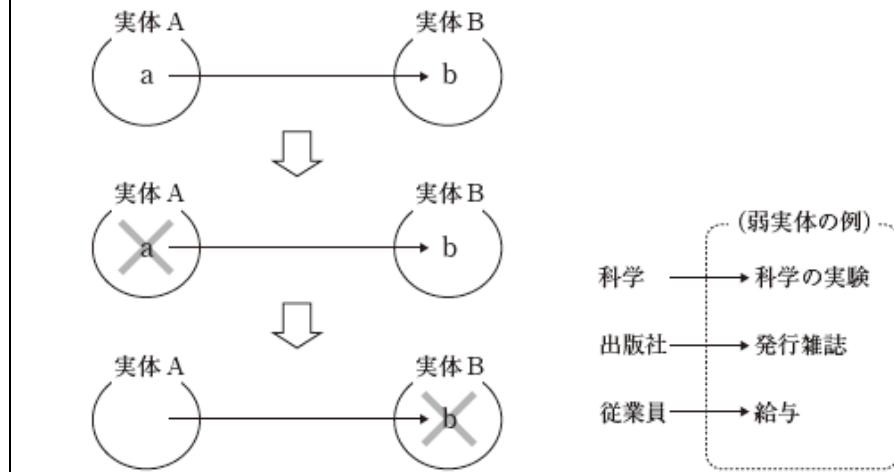
問16 次の E-R 図の解釈として、適切なものはどれか。ここで、多重度の*印は 0 以上を表すものとする。また、自己参照は除くものとする。



- ア 子組織の数より親組織の数が多い可能性がある。
イ 組織は 2 段階の階層構造である。
ウ 組織は必ず子組織をもつ。
エ 組織はネットワーク構造になっていない。

問5：正解（ウ）

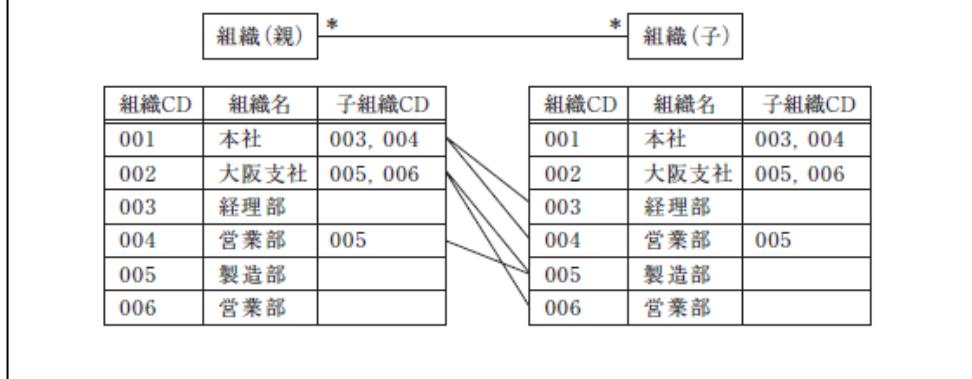
概念データモデルに関する問題。問題文「実体Aのインスタンスaが他の実体Bのインスタンスbと関連しており、aが存在しなくなれば、bも存在しなくなる。」とは、ちょうど次の図のような関係である（たとえば、科学の授業が存在するから、その授業の中で実験ができる。このとき、科学の授業がなくなれば、実験もなくなる）。このような実体Bは弱実体と呼ばれる。よって（ウ）が正解になる。ちなみに、選択肢イの強実体は実体Aのことを指している。



問16：正解ア

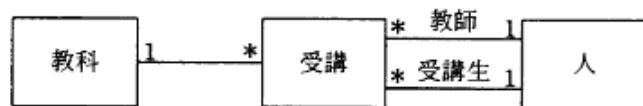
①多密度が0以上、②自己参照を除く、この関係を例示すると次図のようになる。この図をもとに選択肢を順番に見ていく。なお、自己参照をこの図を使って説明すると、組織CDと子組織CDが同じケース（003、経理部、003）のことである。自己参照を認めると、分類階層がループしてしまう。

- ア：子組織CDに何も設定されていなければ、親組織CDの数が多くなる。正しい記述
- イ：図のように、001（本社）→004（営業部）→005（製造部）のケースのように3段階以上になるケースもある。誤り
- ウ：多密度が0以上なので、003、006のように子組織を持たない組織も存在する
- エ：このように多対多の関係を保持できるのはネットワーク構造だけである（図では便宜上、表形式で示しているだけ）



問31 次のデータモデルにおいて，“受講” エンティティの属性として適切なものはどれか。

ここで、長方形の間の線は関連を表し、1 * は 1 対多の多重度を表す。また、関連に付した“教師”と“受講生”はロールを表す。



ア 氏名

イ 成績

ウ 単位数

エ 入学年

問31：正解イ

E-R図に関する問題。図を見て、受講エンティティの属性として適当なものを選択肢から選ぶ問題である。まず、図を次のように整理して考える（以下、“教科”エンティティのエンティティは省略する）。

参照 ➔ 「1.1 E-R図（拡張E-R図）詳細解説」

<“教科”と“受講”的関係>

- ① ひとつの“教科”は複数開催される（複数の“受講”がある）。
- ② ひとつの“受講”に対して，“教科”はひとつ。つまり，“受講”には，“教科”的主キー（教科コードなど）が外部キーとして存在する。

<“受講”と“人（教師）”の関係>

- ① ひとりの“人（教師）”は、複数の講座を担当している（複数の“受講”がある）。
- ② ひとつの“受講”に対して，“人（教師）”はひとつ。つまり，“受講”には，“人（教師）”の主キー（人（教師）コードなど）が外部キーとして存在する。

<“受講”と“人（受講生）”の関係>

- ① ひとりの“人（受講生）”は、複数の講座を受講できる（複数の“受講”がある）。
- ② ひとつの“受講”に対して，“人（受講生）”はひとつ。つまり，“受講”には，“人（受講生）”の主キー（人（受講生）コードなど）が外部キーとして存在する。

以上をまとめると，“受講”に必須の属性は、次のようなになる。

- (a) “教科”的主キー
- (b) “人”的主キー（教師、受講生）
- (c) (a)と(b)を複合キーとし、それに従属する属性

これらの点を踏まえて、選択肢を順番に見ていいく。

ア：上記の（b）に該当するとも考えられるが、同姓同名の存在を考えれば、氏名を“人”

の主キーにするよりも、人（教師・受講生）コード→氏名にする方がよい。よって“人”エンティティの属性が適当。誤りである

イ：上記の（a）（b）ではないが、成績は受講ごとに発生するものなので、[教科コード、教師コード、受講生コード] → 成績になる。そう考えると（c）に該当するので“受講”エンティティの属性が適当。正解である

ウ：上記の（a）（b）でもなく、教科→単位数なので、“教科”エンティティの属性が適当。（c）にもならないので誤りである

エ：上記の（a）（b）でもなく、受講生→入学年なので、“人”エンティティの属性が適当。（c）にもならないので誤りである

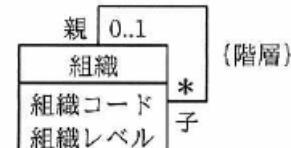
2.3 様々なビジネスモデル

令和3年度・D B・午前Ⅱ

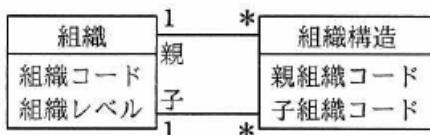
問2 部、課、係の階層関係から成る組織のデータモデルとして、モデル A～C の三つの案が提出された。これらに対する解釈として、適切なものはどれか。組織階層における組織の位置を組織レベルと呼ぶ。組織間の階層関係は、親子として記述している。親と子は循環しないものとする。ここで、モデルの表記には UML を用い、{階層}は組織の親と子の関連が循環しないことを指定する制約記述である。



モデルA



モデルB



モデルC

- ア 新しい組織レベルを設ける場合、どのモデルも変更する必要はない。
- イ どのモデルも、一つの子組織が複数の親組織から管轄される状況を記述できない。
- ウ モデル B を関係データベース上に実装する場合、親は子の組織コードを外部キーとする。
- エ モデル C では、組織の親子関係が循環しないように制約を課す必要がある。

問2：正解（工）

リレーションシップに関する問題。個々のモデルは次のようになる。

モデルA：部を最上位、係を最下位とする階層構造（親子関係）を持ったリレーションシップ。

モデルB：自己参照によって階層構造を示したモデル。

モデルC：“組織構造”を連関エンティティとして、ひとつの“組織”エンティティで階層構造を示したモデル。

これらの相違を念頭に、選択肢を順番に見ていく。

ア：組織レベルとは「組織階層における組織の位置」と言っているので“部”や“課”を指している。これを追加する場合、モデルAでは階層がひとつ増えるわけなのでエンティティそのものを増やすといけない。したがって変更が必要になる。他の二つのモデルは、組織レベルを属性として持っているので、データの追加で対応できる。

イ：モデルAとモデルBは“親”的方の上限が“1”になっているので、複数の親組織から管轄される状況を記述できないが（複数の親を持つ子は存在しないが）、モデルCの場合は、親組織コードと子組織コードの組合せを“組織構造”に持たせるだけなので、その関係はn対nになる。したがって、複数の親組織から管轄される状況でも記述できる。

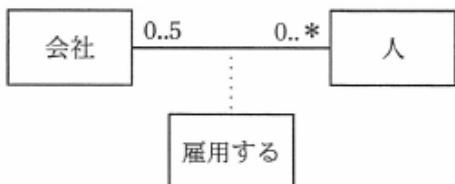
ウ：逆。子は親の組織コードを外部キーとする。

エ：正解。モデルCは、n対nで親子関係を自由に設定できてしまうので、循環しないように制約を課さなければならない。

親組織コード	子組織コード
0001	0010
0010	0001
0002	0003
0003	0004
0004	0002

図：モデルCで循環する例

問4 次の概念データモデルを関係データベース上に実装することとし、実装用のデータモデルを作成した。適切な多重度が指定されているものはどれか。ここで、モデルの表記には UML を用いる。



問4：正解（エ）

過去問題（平成20年・DB午前 問34など）

参照▶「1.2.2 多重度」

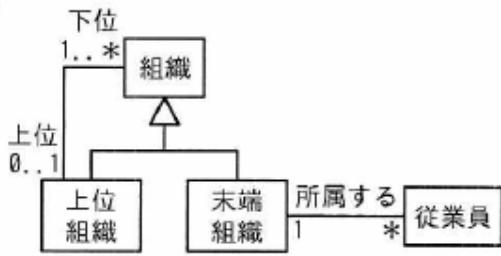
UMLのクラスに関する問題。多対多の関係は非正規形モデルになるため、1対多の関係に分けなければならない。その場合、選択肢のように、①新たなテーブルを作成し、②両方の主キーからなる複合キーを主キーとする。ただし、多重度に関しては次のように考える。

- ① ひとつの会社は、0人から*（無制限）の人を雇用できる（問題文の図より）
- ② 一人の人は、0社から5社までの会社に雇用される（問題文の図より）
<これを“雇用”エンティティとの関係で考えると……>
- ③ ひとつの会社は、0人から*（無制限）の人を雇用に登録できる
- ④ 一人の人は、0社から5社まで雇用に登録できる
- ⑤ 雇用1件には、会社も1社、人も一人（いずれも外部キーになる）

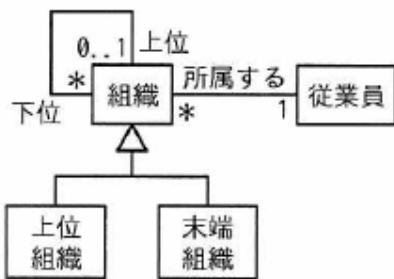
以上より、正解は（エ）になる。

問6 階層構造をもつ組織と、従業員の組織への所属を表す UML のクラス図のうち、“従業員は組織階層中のどの組織にも所属できるが、兼務はしない”とするものはどれか。

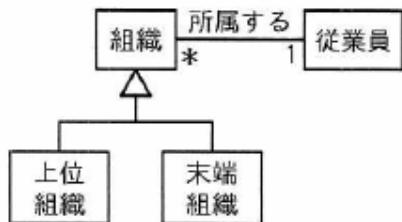
ア



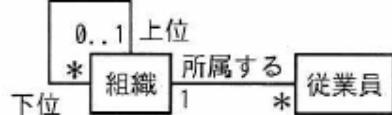
イ



ウ



エ



UML表記のクラス図に関する問題。階層構造をもつ組織で「従業員は組織階層中のどの組織にも所属できるが、兼務はしない（＝複数の組織には所属しない）」を表現しているものを答える。選択肢は、よく似ているので共通部分を次のように整理した上で順番に見ていく。

① 「△」は継承関係（Generalization）を示す（選択肢ア・イ・ウ）

- ・“組織”がスーパークラス（親クラス）であり、“上位組織”と“末端組織”がサブクラス（子クラス）になる。
- ・この関係は「種類の分類」を意味し、組織の管理構造（親子関係）ではない。
- ・“上位組織”と“末端組織”は同じ親を持つ兄弟クラスになる。互いに継承関係もなければ、そこに階層構造も存在しない。

② “組織”的自己参照（選択肢イ・エ）

自己参照を使うことで「組織」は階層構造を持つことができる

③ “従業員”と“組織”的間にあるリレーションシップ（選択肢イ・ウ・エ）

- ・このリレーションシップがあれば、従業員はどの組織にも所属できる。
- ・“従業員”と“組織”が多対1の関係…“従業員”から見て“組織”は一つしかない。つまり兼務はしないことを示している。選択肢エ。
- ・“従業員”と“組織”が1対多の関係…“従業員”から見て“組織”は複数ある。つまり兼務は可能だということを示している。選択肢イ・ウ。

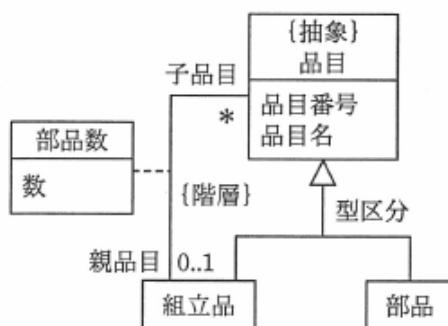
ア：上記の①に加えて、“組織”と“上位組織”的間に多対1のリレーションシップがある。これは一つの“上位組織”が複数の“組織”を持つことを意味する。そういう意味では1階層だけになるが階層化はされている。しかし、“末端組織”はその階層化とは無関係に存在することになる。また、“従業員”と“末端組織”的間にしかリレーションシップが無いため、従業員は末端組織にしか所属できないことになる。その点で要件を満たすことができない。誤り。

イ：“組織”には自己参照するリレーションシップがあるため、階層構造を持つことになる（上記の②）。ただし、その階層構造と“上位組織”及び“末端組織”とは無関係である（上記の①）。また、上記の③より、1人の従業員は複数の組織に所属できる。いずれも問題の要件を満たさないため誤り。

ウ：上記の①より、組織の管理構造（親子関係）としての階層化にはなっていない。また、上記の③より、1人の従業員は複数の組織に所属できる。いずれも問題の要件を満たさないため誤り。

エ：正しい。自己参照を使うことで組織は階層構造を持つことができるし（上記の②）、加えて上記の③より、従業員はどの組織にも所属できるし、従業員は兼務はしないという要件も満たす。

問3 次の概念データモデルを関係データベース上に実装するとき、適切な関係スキーマ定義はどれか。ここで、モデルの表記には UML を用いる。関係スキーマ定義中の実線の下線は主キーを、破線の下線は外部キーを表す。



- ア 組立品（親品目番号, 品目名）
- 部品（子品目番号, 品目名）
- 部品数（親品目番号, 子品目番号, 数）
- イ 品目（親品目番号, 品目名, 型区分）
- 部品数（子品目番号, 数）
- ウ 品目（品目番号, 品目名, 型区分）
- 組立品（組立品番号, 品目番号, 品目名）
- 部品（部品番号, 品目番号, 品目名）
- 部品数（部品番号, 組立品番号, 数）
- エ 品目（品目番号, 品目名, 型区分）
- 部品数（親品目番号, 子品目番号, 数）

問3：正解（エ）

(NEW)

参照 [1.2 E-R図]

概念データモデルと関係スキーマの表記方法に関する問題。モデルの表記にはUMLを用いるという制約がある点に注意。問題文の図の理解の仕方を、下図に示す。

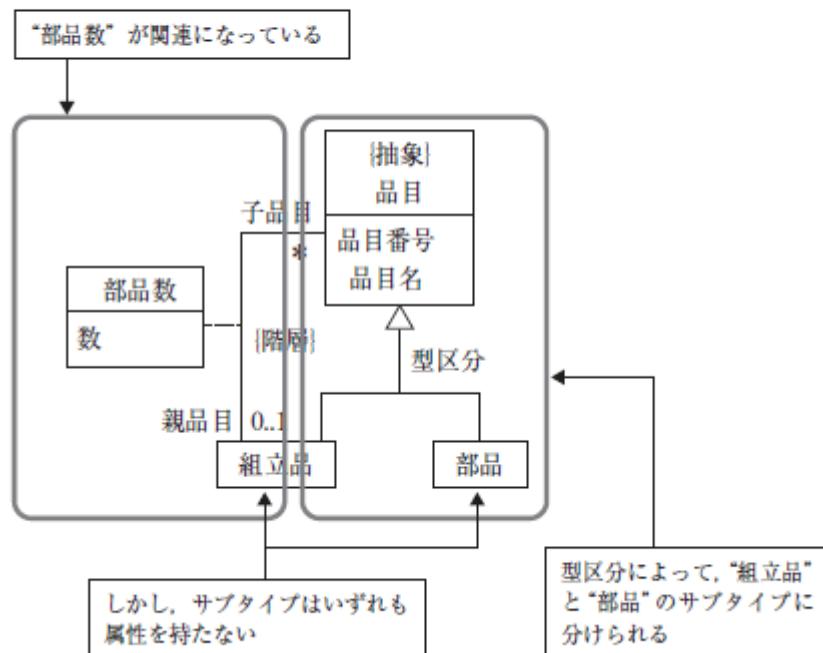
まず、下図の右側のスーパー型とサブタイプの関係より、これを、次のように関係スキーマに実装する。サブタイプは属性を持たないので、実装はこれで良い。

品目（品目番号、品目名、型区分）

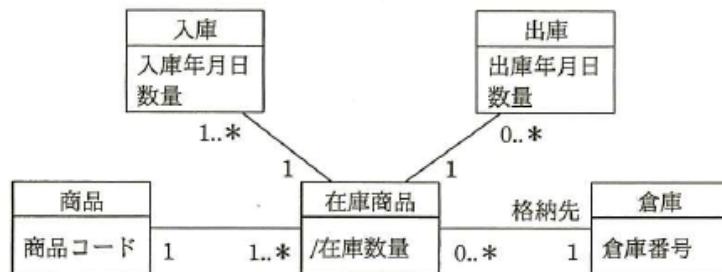
次に、親品目と子品目の関係を考える。下図の左側を見ると、“部品数”という関連の存在を見つけることができる。関連なので、“部品数”の主キーは、親品目番号と子品目番号の連結キーになる。そして属性に数を持つ。

部品数（親品目番号、子品目番号、数）

以上より、正解は（エ）になる。



問 1 UML を用いて表した商品と倉庫のデータモデルに関する記述のうち、適切なもの
はどれか。ここで、商品の倉庫間の移動はないものとする。



- ア 1種類の商品を二つの倉庫に初めて入庫すると、“在庫商品”データが 2 件追加される。
- イ 2種類の商品を一つの倉庫に入庫すると、“入庫”データが 1 件追加される。
- ウ 格納先となる倉庫が確定していない商品が存在する。
- エ 出庫の実績がない在庫商品は存在しない。

問1：正解（ア）

UML表記に関する問題。選択肢を順番に見ていく。わかりにくければ、実際に何かしら具体的なデータを使って考えてみればいい。

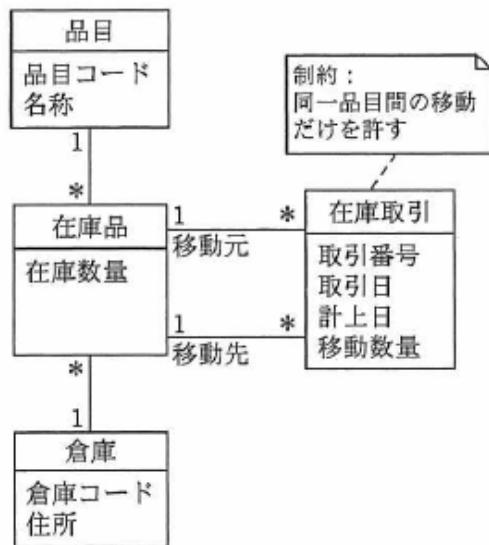
ア：商品“001”を、倉庫“A001”と“B001”に初めて入庫する場合に、“在庫商品”データは“001,A001”と“001,B001”的2件が追加される（“在庫商品”は、“商品”と“倉庫”的連関エンティティになるので、その主キーは、{商品コード、倉庫番号}になる）。したがって、この記述は正しい。

イ：2種類の商品“001”, “002”を、一つの倉庫“A001”に入庫すると、“在庫商品”データは“001,A001”と“002,A001”的2件が追加・更新される。“入庫”は、“在庫商品”と多対1の関係にあるので、主キーは {商品コード、倉庫番号、入庫年月日} だと考えられる。だとすれば2件追加されることになるので、この記述は誤り。

ウ：“商品”に対して“在庫商品”が1以上なので、その主キー {商品コード、倉庫番号}を考えると、必ず倉庫は決まっていないといけない。誤り。

エ：常識的に考えても誤り。入荷だけで、まだ出庫していない在庫商品は普通に存在する。“在庫商品”に対して、“出庫”は“0”的場合があることからも誤りであることが確認できる。

問 1 UML を用いて表した図のデータモデルの解釈として、適切なものはどれか。

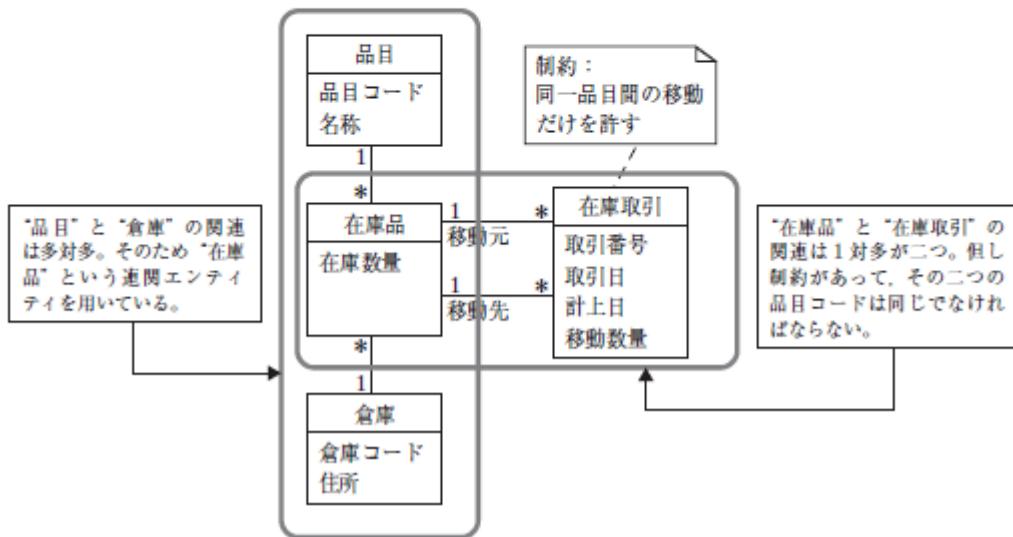


- ア 1 件の“在庫取引”データを記録する際、2 件の“在庫品”データも更新する。
- イ “在庫品”データは、現在の在庫数量だけでなく、過去の在庫数量も保持する。
- ウ 倉庫別、品目別に入出庫の状況を把握することはできない。
- エ 品目の異なる“在庫品”データ間で“在庫取引”データを記録できる。

問1：正解（ア）

NEW

データモデルの理解に関する問題。モデルの表記にはUMLを用いるという制約がある点に注意。問題文の図の理解の仕方を、下図に示す。



この図の解釈をもとに、選択肢を順番に見ていく。

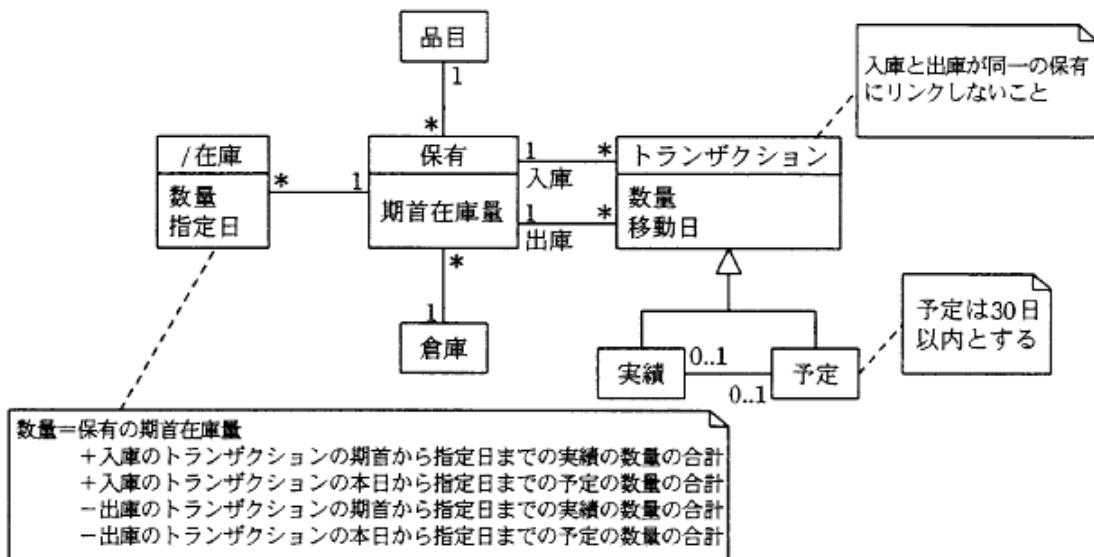
ア：正しい。“在庫取引”から“在庫品”に関して、リレーションシップが2本出ている。これは、1件の“在庫取引”データを記録するときに、2件の“在庫品データを更新することを意味している。

イ：“在庫品”データは、品目別倉庫別に1件作成される。したがって、在庫数量は品目別倉庫別に1種類しか保持できない。したがって、現在在庫数量と過去の在庫数量を同時に管理することはできない（通常は現在在庫数として利用）。したがって誤り。

ウ：“在庫取引”データは、ある品目に関して倉庫の移動元と移動先を保持しているので、それを利用すれば、入出庫の状況を把握することは可能。したがって誤り。

エ：“在庫取引”には「同一品目間の移動だけを許す」という制約があるので、品目の異なる“在庫品”データ間で“在庫取引”データは記録できない。誤り。

問31 次のデータモデルに関する記述のうち、適切なものはどれか。ここで、モデルの表記に UML を用いる。クラス名の先頭の “/” は、それが導出クラスであることを示す。



- ア “/在庫” クラスは、“トランザクション” クラスにインスタンスが追加されるた
びに更新されなければならない。
 - イ “トランザクション” クラスの移動日は、“/在庫” クラスの数量を日ごとに集計
するために利用される。
 - ウ “トランザクション” クラスの一つのインスタンスは、入庫と出庫のどちらか一
方の移動を記録する。
 - エ “保有” クラスのインスタンスは品目コードごとに作られる。

問 31：正解イ

データモデルの理解に関する問題。選択肢を順番に見ていく。ちなみに、この図から次のようなことが推測できる。

- ・“保有” クラスは、倉庫別品目別の期首在庫量を持っている
- ・“トランザクション” クラスは、ある日のある品目の移動データである

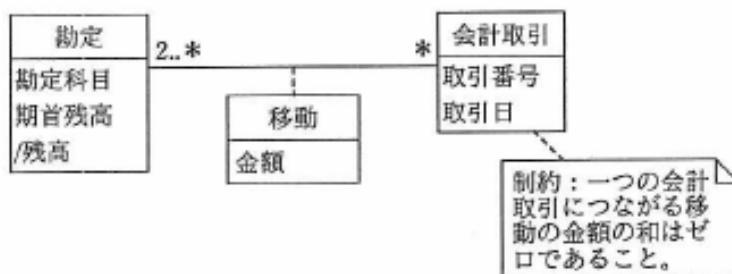
ア：特にその記述はない

イ：正解である

ウ：“トランザクション” クラスと“保有” クラスの間に入庫と出庫の2本の線が出ているので、ひとつのインスタンスに入庫と出庫がある。よって誤りである

エ：“保有” クラスのインスタンスは、品目別倉庫別に作成される。よって誤りである

問 2 図のデータモデルは会計取引の仕訳を表現している。“移動”がリンクする“勘定”の残高を増やす場合は金額の符号を正に、減らす場合は負にすることで、貸借平均の原理を表現する。このモデルに基づき、“勘定”表，“会計取引”表，“移動”表を定義した。勘定科目“現金”の 2017 年 4 月 30 日における残高を導出するための SQL 文はどれか。ここで、モデルの表記には UML を用い、表中の実線の下線は主キーを表す。また、“会計取引”表には今期分のデータだけが保持される。



勘定（勘定科目, 期首残高）

会計取引（取引番号, 取引日）

移動（勘定科目, 取引番号, 金額）

ア SELECT SUM(金額) AS 残高 FROM 勘定, 移動, 会計取引
 WHERE 勘定.勘定科目 = 移動.勘定科目 AND
 会計取引.取引番号 = 移動.取引番号 AND
 勘定.勘定科目 = '現金' AND
 取引日 <= '2017-04-30'

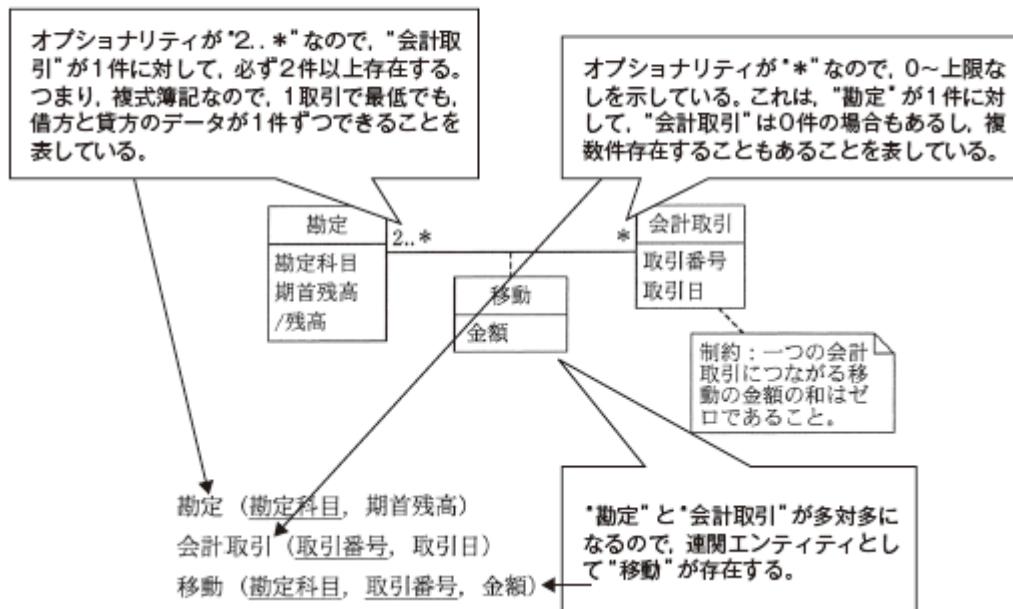
イ SELECT 期首残高 + SUM(金額) AS 残高 FROM 勘定, 移動, 会計取引
 WHERE 勘定.勘定科目 = 移動.勘定科目 AND
 会計取引.取引番号 = 移動.取引番号 AND
 勘定.勘定科目 = '現金' AND
 取引日 <= '2017-04-30'
 GROUP BY 勘定.勘定科目, 期首残高

ウ SELECT 残高 FROM 勘定, 移動, 会計取引
 WHERE 勘定.勘定科目 = '現金' AND
 取引日 <= '2017-04-30'

エ SELECT 残高 FROM 勘定, 移動, 会計取引
 WHERE 勘定.勘定科目 = 移動.勘定科目 AND
 勘定.勘定科目 = '現金' AND
 取引日 <= '2017-04-30'

問2：正解（イ）

SQLに関する問題（参照▶第1章）。解答に当たっては、問題文のUML表記と簿記の仕訳に関する知識が必要になる。まずは、問題文の記述とUML表記、関係スキーマから具体的なデータを想像しながら適切なSQL文を探していく。問題文の図の読解ポイントは次の通り。



以上をまとめると、次のようなことが確認できる。

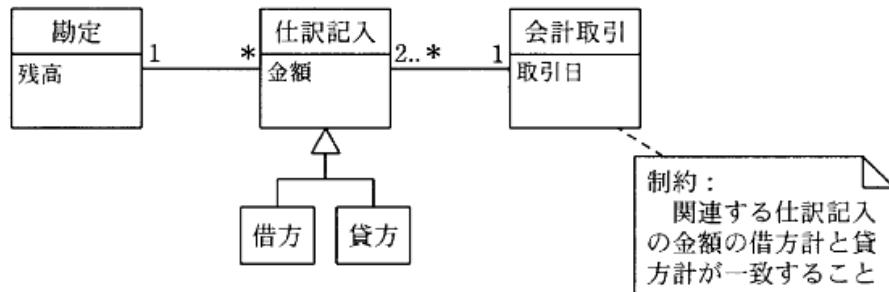
- あらかじめ“勘定”は登録されている。
- “会計取引”1件に対して、必ず2件以上の“勘定”が登録される。複式簿記。
- “移動”は連関エンティティ。

次に、選択肢アからエまでのSQLを比較して、どういう違いがあるのかを確認する。

- 使用テーブルが3つというのは全部同じ (FROM以下)。
- 勘定.勘定科目='現金'という条件式が含まれているのも全部同じ。
- 取引日<='2017-04-30'という条件式が含まれているのも全部同じ。
- 3つの表の結合が正しいかどうかをチェック。ウとエは結合できない。誤り。
- GROUP BYがあるものとないもの。
- 残高が、SUM(金額)と期首残高+SUM(金額)。

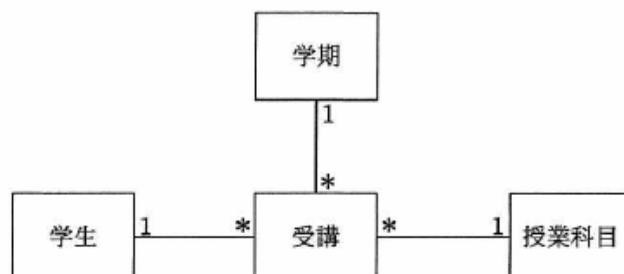
求められているのは「勘定科目“現金”の2017年4月30日における残高」なので、2017年4月30日の取引金額だけではなく、期首残高を加えなくてはいけない。条件式で勘定科目を現金だけに絞り込んでいるが、期首残高をリストに含めているので、GROUP BYで指定している。選択肢アは、期首残高を残高に加えていないので誤り。正解はイになる。

問28 概念レベルのデータモデル中のエンティティ “会計取引” に対する制約の意味について、適切な説明はどれか。ここで、モデルの表記には UML を用いる。



- ア 会計取引に対応する仕訳記入の金額が今後も変化しないことを保証する。
- イ 勘定間を移動する金額が、その会計取引において一致することを保証する。
- ウ 勘定の残高が、その会計取引を記録するごとに計算されることを保証する。
- エ 同一の勘定同士で会計取引が行われないことを保証する。

問4 UML を用いて記述したデータモデルにおいて、“受講” クラスの属性として、適切なものはどれか。



ア 学期名

イ 科目名

ウ 氏名

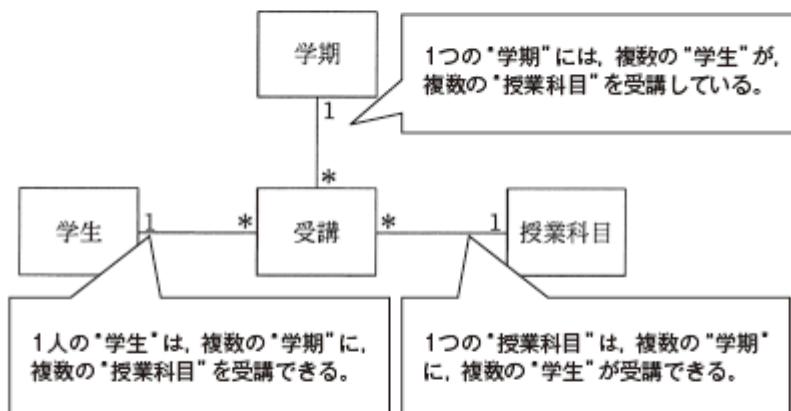
エ 成績

問 28：正解イ

複式簿記の基礎知識があれば簡単に答えることができる問題。複式簿記では、必ず借方と貸方の金額は一致しなければならない。そうなるように制約をかけているのである。これは、様々な勘定科目に分類かつ登録されたとしても、最終的に借方合計と貸方合計が一致することを保証するもので、正解はイになる。

問 4：正解（工）

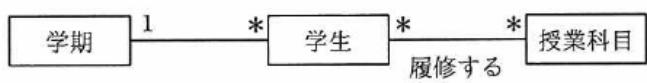
データモデルの表記方法に関する問題。午後試験とは異なり、UML表記について問われている。最初に、問題文の図から、下記のような関係性を読み取る。その上で、選択肢の属性がどこに属されるのが最適かを考えていく。



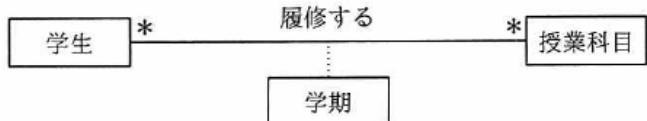
- ア：‘学期名’は、‘学期’単位に確定するものなので、‘学期’の属性とするのが妥当である。
イ：‘科目名’は、‘授業科目’単位に確定するものなので、‘授業科目’の属性とするのが妥当である。
ウ：‘氏名’は、‘学生’単位に確定するものなので、‘学生’の属性とするのが妥当である。
エ：‘成績’は、1人の‘学生’に対して‘学期’ごと、‘授業科目’ごとに存在するものなので‘受講’の属性とするのが妥当である。これが正解。

問6 “学生は、学期が異なれば同じ授業科目を何度も履修できる”を適切に表現しているデータモデルはどれか。ここで、モデルの表記には UML を用いる。

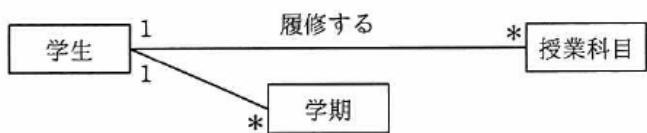
ア



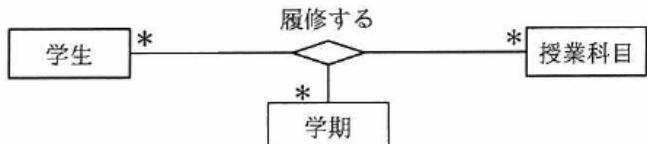
イ



ウ



エ



問6：正解（エ）

データモデルの表記方法に関する問題。UML表記について問われている。最初に、問題文の記述を整理して個々のデータ関係を明確にしていく。

- ①“学生”と“授業科目”は多対多の関係（ただし、学期が異なることが条件）
- ②“学生”と“学期”は、多対多
- ③“学期”と“授業科目”は、多対多

つまり、三つ以上のクラスに関連がある。その場合、UML2.0では、エンティティ間を“ひし形”で3項をつなげる記述になるので、正解は（エ）になる。なお、念のために他の選択肢もチェックしておく。

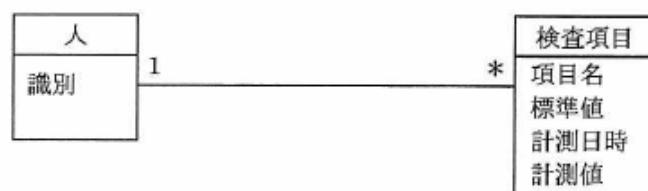
ア：1人の“学生”に対して“学期”は一つじゃない。誤り。

イ：UMLの表記方法ではないので、誤り。

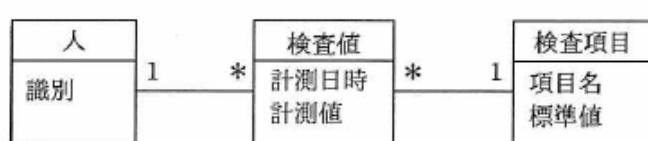
ウ：一つの“学期”に対して、“学生”が1人しか存在しないわけがない。また、一つの“授業科目”に“学生”が1人しかいないこともない。したがって誤り。

問3 人の健康状態の検査では、検査項目が人によって異なるだけでなく、あらかじめ決まっていないことが多い。このような場合のデータモデルとして、最も適切なものはどれか。ここで、検査項目の標準値は、検査項目ごとに最新の値だけを保持し、計測値は計測日時とともに保持する。また、モデルの表記には UML を用いる。

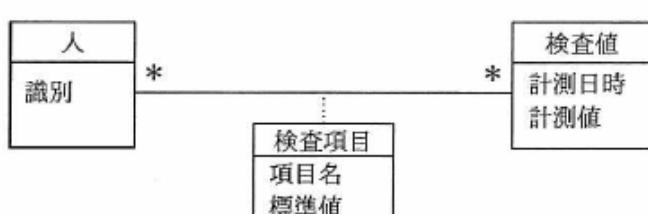
ア



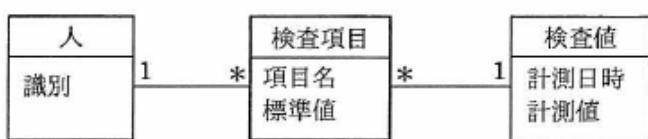
イ



ウ



エ



問 3：正解（イ）

概念データモデルの表記方法に関する問題。モデルの表記には UML を用いるという制約がある点に注意。“人”と“検査項目”，及び“検査値”に着目して問題文を整理すると次のようなになる。なお，イメージがわからない場合は，次のように具体的なインスタンスを使って考えてみるとわかりやすい。

① 「検査項目が人によって異なるだけでなく，あらかじめ決まっていないことが多い。」

A さん…肺検査，胃の検査

B さん…頭の検査

C さん…未定 (NULL)

② 「検査項目の標準値は… 計測値は計測日時とともに保持する。」 = その履歴を持つ

肺検査…25, (見直し) 27, (見直し) 29

胃の検査…以下略

→ 選択肢を確認。“標準値”はすべて“検査項目”に従属しているので，“検査項目”には履歴を持つ必要はない。その時々の計測値は“検査値”で保持しているので問題ない。

※ ひとりの人が複数の検査をするのかどうかは，問題文には書かれていないが，選択肢を見る限り，すべて，“人”から見た“検査項目”あるいは“検査値”は“多 (*)”になる。

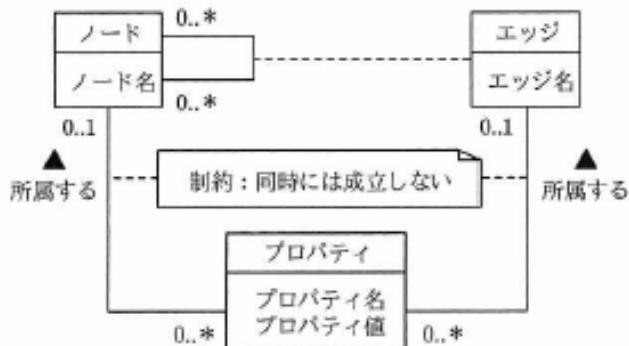
以上より，次のことが言える。

③ “人”と“検査項目”的関係が「多対多」

④ 上記を，連関エンティティ“検査値”を用いて「1対多」に分ける

したがって正解はイになる。

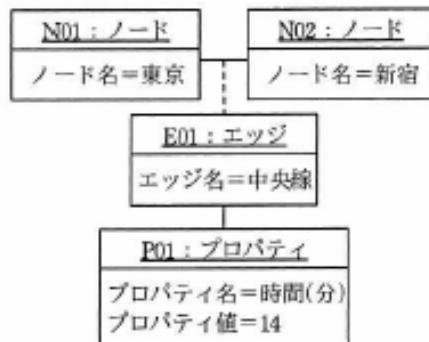
問4 プロパティグラフを表した図のデータモデルを適切に解釈したオブジェクト図は
どれか。ここで、モデルの表記には UML を用いる。



ア



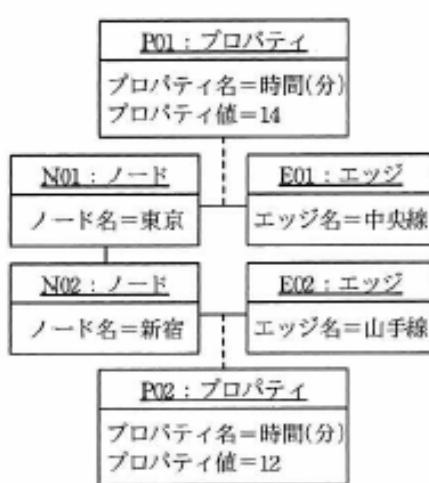
イ



ウ



エ



問4：正解（イ）

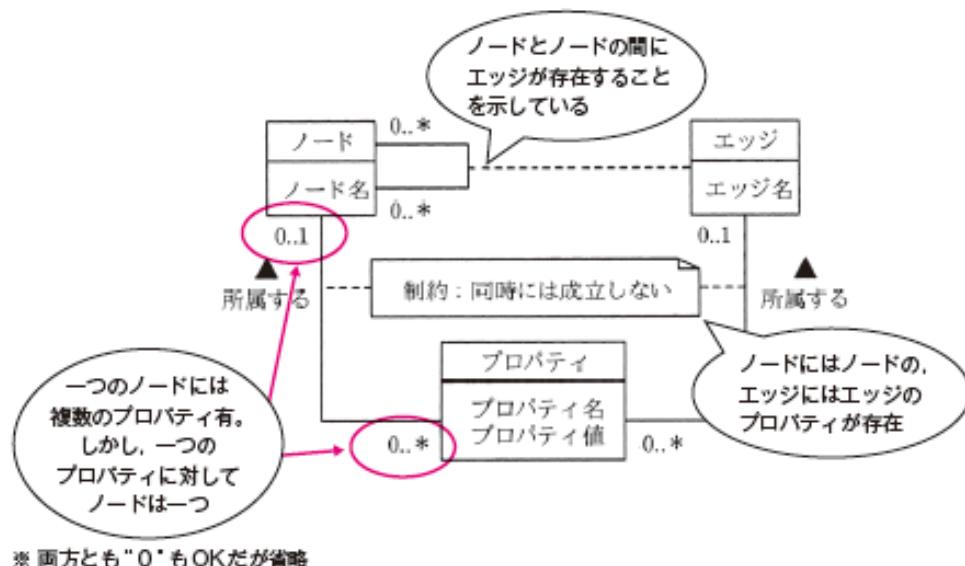
プロパティグラフに関する問題。プロパティグラフは、グラフ理論に基づくグラフ構造を備えたグラフデータベースの要素で、ビッグデータ環境で使用される。ノードとエッジはグラフ理論の名称からきている。具体的には、この問題の選択肢の例にあるように、駅をノードとして、駅と駅の二地点間の接続経路をエッジとしている。そしてプロパティは、ノードもしくはエッジの属性情報になる。

ノード：エンティティを表す

エッジ：ノード間の経路や関係性を表す

プロパティ：ノード、エッジの属性情報

その上で、問題文のUML表記より、次の規則性を読み取る。



以上の知識を前提にして選択肢を順番にチェックしていく。

ア：エッジ間のノードになっている。エッジとノードの関係性が逆。誤り。

イ：全ての要件をクリア。正解。

ウ：一つのプロパティに複数のノードがある点で誤り。

エ：ノードとエッジが1対1で結ばれている。しかもノードとエッジの“関連”に対して各プロパティがある。誤り。

なお、グラフデータベースに関する知識がなくても、UML 表記の知識だけで消去法で正解を導くことはできる。

ア：この図はエッジ同士が関連性をもっている（実線でつながっている）が、問題の図にはエッジ同士の関連性に関する記載はない。誤り。

イ：ノード間同士の関連付けは、問題の図に記載されている。“ノード間の関連”と“エッジ”の間には、どちらも点線がある（関連がある）。また、エッジかノードのいずれかにプロパティが関連性をもっている（但し、同時には成立しない）が、選択肢イの図もそうなっている。したがっておかしなところはないので、（あるいは消去法で）正解になる。

ウ：2つのノードとエッジの関連性などは設問イで説明した通り問題はないが、1つのプロパティが2つのノードに関連付けられているため誤り。

エ：ノードとエッジが直接関連付けられている。問題では“ノードとノードの関連”と“エッジ”が関連付けられているため誤り。

他にも、選択肢の例の意味の合理性から正解を探ってもいいだろう。

3

第3章

関係スキーマ

ここでは、関係スキーマについて説明する。関係スキーマとは、関係を関係名とそれを構成する属性名で表したもの。情報処理技術者試験では、午後Ⅰ・午後Ⅱの両試験で必ず登場している第2章の概念データモデルに並ぶ最重要キーワードの一つといえるだろう。

3.2 関数従属性

平成 28 年度・D B・午前Ⅱ

問 3 関係 $R(A, B, C)$ において、関数従属 $A \rightarrow B, B \rightarrow C$ が成立するとき、導けない
関数従属はどれか。

- ア $\{A, B, C\} \rightarrow \{A, B\}$ イ $\{A, C\} \rightarrow \{A, B\}$
ウ $\{A, C\} \rightarrow \{A, B, C\}$ エ $\{B, C\} \rightarrow \{A, C\}$

平成 25 年度・D B・午前Ⅱ

問 2 関数従属に関する記述のうち、適切なものはどれか。ここで、 A, B, C はある関係
の属性の集合とする。

- ア B が A に関数従属し、 C が A に関数従属すれば、 C は B に関数従属する。
イ B が A の部分集合であり、 C が A に関数従属すれば、 C は B に関数従属する。
ウ B が A の部分集合であれば、 A は B に関数従属する。
エ B と C の和集合が A に関数従属すれば、 B と C はそれぞれが A に関数従属する。

問 3：正解（工）

関数従属に関する問題（参照 ➔ 第 2 章）。選択肢を順番に見ていく。

- ア : A, B, C が決定しているのだから、当然 A, B の組合せも決定している。成立する。
- イ : A → B なので、A が決定すれば |A, B| も決定する。成立する。
- ウ : A → B なので、A が決定すれば |A, B| も決定する。これに、C との組合せも決定するので |A, C| → |A, B, C| も成立する。
- エ : B → C より |A, C| のうち C は決定するが、A は決定しない。したがって、これは成立しない。

問 2：正解（工）

過去問題（平成 20 年・DB 午前 問 22）
参照 ➔ 「2.2 関数従属性」

関数従属に関する問題。選択肢を順番に見ていく。

- ア : A → B, A → C から、B → C を導き出すことはできない。誤り。
- イ : アと同様に、A → B (B は A の部分集合であるため関数従属が成り立つ), A → C から、B → C を導き出すことはできない。
- ウ : B が A の部分集合であれば、A が B に関数従属するのではなく、その逆 (B は A に関数従属する)。したがって誤り。
- エ : 正解である。

問4 関係 $R(A, B, C, D, E)$ に対し、関数従属の集合 $W = \{A \rightarrow \{B, C\}, \{A, D\} \rightarrow E, \{A, C, D\} \rightarrow E, B \rightarrow C, C \rightarrow B\}$ がある。関数従属の集合 X, Y, Z のうち、 W から冗長な関数従属をなくしたもののはどれか。

$$X = \{A \rightarrow B, B \rightarrow C, C \rightarrow B, \{A, D\} \rightarrow E\}$$

$$Y = \{A \rightarrow C, B \rightarrow C, C \rightarrow B, \{A, D\} \rightarrow E\}$$

$$Z = \{A \rightarrow B, C \rightarrow B, \{A, C, D\} \rightarrow E\}$$

ア X だけ

イ X と Y

ウ Y と Z

エ Z だけ

問4：正解（イ）

関数従属に関する問題。関数従属の集合 W から冗長な関数従属をなくしたものを見る。

$$W = \{A \rightarrow \{B, C\}, \{A, D\} \rightarrow E, \{A, C, D\} \rightarrow E, B \rightarrow C, C \rightarrow B\}$$

まず、 $\{A, D\} \rightarrow E$ と $\{A, C, D\} \rightarrow E$ に着目する。A と D が決まれば E が決まるということなので、A と D に C を加えても E は一意に決まる。したがって、 $\{A, D\} \rightarrow E$ と $\{A, C, D\} \rightarrow E$ は冗長なので $\{A, C, D\} \rightarrow E$ を排除する。

次に、 $A \rightarrow \{B, C\}$ と $B \rightarrow C, C \rightarrow B$ に着目する。 $A \rightarrow \{B, C\}$ は、 $A \rightarrow B$ と $A \rightarrow C$ に分解できる（分解律）。そして $A \rightarrow B, A \rightarrow C, B \rightarrow C, C \rightarrow B$ で考えると、 $A \rightarrow B$ と $B \rightarrow C$ ならば $A \rightarrow C$ が成立する（推移律）。 $A \rightarrow B$ と $B \rightarrow C$ の組み合わせと $A \rightarrow C$ は冗長になる。それゆえ $A \rightarrow C$ も排除する。以上より、残った $\{A, D\} \rightarrow E, A \rightarrow B, B \rightarrow C, C \rightarrow B$ が W から冗長な関数従属をなくしたものになる。この集合は X と同じなので、X は W から冗長な関数従属をなくしたものになる。

また、 $A \rightarrow B, A \rightarrow C, B \rightarrow C, C \rightarrow B$ のうち、 $A \rightarrow C$ と $C \rightarrow B$ ならば $A \rightarrow B$ が成立する（推移律）。 $A \rightarrow C$ と $C \rightarrow B$ の組み合わせと $A \rightarrow B$ は冗長になる。それゆえ、この場合は $A \rightarrow B$ を排除する。以上より、残った $\{A, D\} \rightarrow E, A \rightarrow C, B \rightarrow C, C \rightarrow B$ が W から冗長な関数従属をなくしたものになる。この集合は Y と同じなので、Y も W から冗長な関数従属をなくしたものになる。

以上より、X と Y が対象になるので、正解はイになる。

問3 関係 $R(A, B, C, D, E)$ において、関数従属性 $A \rightarrow B$, $A \rightarrow C$, $\{C, D\} \rightarrow E$ が成立する。最初に属性集合 $\{A, B\}$ に対して、これらの関数従属性によって関数的に決定される属性をこの属性集合に加える。この操作を繰り返して得られる属性集合（属性集合の閉包）はどれか。

ア $\{A, B, C\}$

イ $\{A, B, C, D\}$

ウ $\{A, B, C, D, E\}$

エ $\{A, B, E\}$

問3：正解（ア）

集合論に関する問題。閉包について問われている。但し、この問題は閉包のことを知らないても、問題文に記載されている通りに考えていけば正解は求められる。具体的には、最初に与えられた属性集合 $|A, B|$ から、下記の関数従属性に関して一つずつ、導出される属性を加えていけばいい。

- ① $A \rightarrow B$, ② $A \rightarrow C$, ③ $|C, D| \rightarrow E$

① $A \rightarrow B$ A も B も集合に含まれるので、属性の増加はない。 $|A, B|$

② $A \rightarrow C$ 部分集合 A から C が導出できるので、属性 C を追加する。 $|A, B, C|$

- ③ $|C, D| \rightarrow E$

D が存在しないので、 E を導出することはできない。

よって、属性の追加はない。

以上より、 $|A, B, C|$ が正解になる。ちなみに、閉包とは、集合論における重要な概念の一つで、ある集合に対して、その集合を構成する要素に関連する全ての要素を含む最小の集合のことである。

3.3 キー

平成 17 年度・D B・午前

問23 次の関係“注文”的属性に①～⑦の関数従属性があるとき、主キー属性の組として正しいものはどれか。ここで、(A, B)という記述は、属性 A と B の組を表す。また、 $A \rightarrow C$ という記述は、C が A に関数従属していることを表す。

関係“注文”

(注文番号, 注文日, 顧客番号, 顧客名, 商品番号, 商品名, 数量, 金額)

関数従属性

- | | |
|---------------------|---------------------|
| ① 注文番号 → 注文日 | ② 注文番号 → 顧客番号 |
| ③ 注文番号 → 顧客名 | ④ 顧客番号 → 顧客名 |
| ⑤ (注文番号, 商品番号) → 数量 | ⑥ (注文番号, 商品番号) → 金額 |
| ⑦ 商品番号 → 商品名 | |

ア (注文番号)

イ (注文番号, 顧客番号)

ウ (注文番号, 顧客番号, 商品番号)

エ (注文番号, 商品番号)

問 23：正解工

主キーとは、候補キーの一つで、タブル（行）を一意に識別できる属性の組のことである。したがって、主キー属性は、すべての非キー項目に対して関数従属性を持つ必要がある。これを満たす属性の組合せは、エ（注文番号、商品番号）である。“注文番号”、“商品番号”を含むよりもタブルを一意に識別できるが、極小ではないため主キーとなりえない。

参照 ➤ 「1.7.2 キー」、「2.2.1 関数従属性」

問 2 データベースの関係モデルの候補キーの説明として、最も適切なものはどれか。ここで、極小の組とは、その組から属性が一つでも欠落すると唯一識別性を失ってしまう組をいう。

- ア 関係のタプルを一意に識別できる属性、又は属性の組
- イ 関係のタプルを一意に識別できる属性、又は属性の組で極小のもの
- ウ 関係のタプルを一意に識別できる属性、又は属性の組で極小のものから選んだ一つ
- エ 関係のタプルを一意に識別できる属性、又は属性の組で極小のものから一つ選んだものの残り

問 3 関係 R は属性 A, B, C, D, E から成り、関数従属 $A \rightarrow \{B, C\}$, $\{C, D\} \rightarrow E$ が成立するとき、R の候補キーはどれか。

- ア $\{A, C\}$
- イ $\{A, C, D\}$
- ウ $\{A, D\}$
- エ $\{C, D\}$

問2：正解イ

NEW

参照 「2.3 キー：候補キー」

候補キーに関する問題。候補キーの意味が問われている。適切なものを一つ選ぶ問題なので、選択肢を順番に見ていくべきだ。

ア：“極小のもの”ではないので誤り。“極小のもの”ではない場合は、スーパーキーになる。

イ：これが候補キーの説明。正解。

ウ：“極小のものから選んだ一つは、主キーになる。誤り。

エ：“極小のものから選んだものの残り”は、代理キーになる。誤り。

問3：正解（ウ）

候補キーをすべて列挙させる問題。通常は午後Iで出題されるが、このように午前問題でも出題されることもある。それはさておき、候補キーをすべて列挙するには、次の手順に従う。

問題文にある二つの関数従属を① $A \rightarrow [B, C]$, ② $[C, D] \rightarrow E$ とする。

(手順1) すべての「 \rightarrow 」の視点をピックアップする。① A, ② {C, D}

(手順2) ①②がすべての属性を一意に決定できるかを確認

①…D, E を一意に決定することはできないので、A は候補キーではない

②…A, B を一意に決定することができないので、CD も候補キーではない

結果的に候補キーが見つからなかった場合。

②の {C, D} の中の “C” には、①で、A から \rightarrow が伸びている (① $A \rightarrow [B, C]$)。

そこで、{C, D} の “C” を “A” と置き換える。すると {A, D} になる。この {A, D} からすべての属性を一意に決定できるかを確認する。その結果、{A, D} はすべての属性を一意に決定する。そして、これが極小なので、候補キーになる。

以上より、正解は（ウ）になる。

但し、午前問題の場合は4択なので、選択肢アからエまでの属性が全ての属性を一意に決めることができるかどうかという点と、それが極小かどうかという点でひとつずつチェックしていくても解答できる。

ア：D, E が決まらない。不正解。

イ：すべてを一意に決定できるが、選択肢ウと比べると、極小ではないことがわかる

ウ：正解

エ：A, B が決まらない。不正解。

問4 関係 R (A, B, C, D, E) において、

関数従属 $\{A, B\} \rightarrow C$, $\{B, C\} \rightarrow D$, $D \rightarrow \{A, E\}$

が成立する。これらから決定できる R の候補キーを全て挙げたものはどれか。

ア {A, B, C}

イ {A, B}, {B, C}

ウ {A, B}, {B, C}, {B, D}

エ {B, C}, {C, D}

問4：正解（ウ）

午前には珍しい“候補キーを全て挙げる”問題（午後Iでは定番）。本書の序章に書いてある解答テクニックに準じて解いていこう。

手順① 全ての「 \rightarrow 」の始点をピックアップする。

$\{A, B\}$, $\{B, C\}$, D

手順② 上記のそれぞれがすべての属性を決定できるかどうかを確認

$\{A, B\} \cdots \{A, B\} \rightarrow C$ なので, $\{B, C\}$ も一意になる。

$\{B, C\} \rightarrow D$ なので, D も一意になるため E も決まる

したがって, $\{A, B\}$ が決まれば, C, D, E の全てが決まる

$\{B, C\} \cdots \{B, C\} \rightarrow D$ なので, $\{A, E\}$ も一意になる。

したがって, $\{B, C\}$ が決まれば, A, D, E の全てが決まる

D \cdots D, A, E は決まるが, B, C が決まらない。

手順③ $\{A, B\}$ もしくは A, もしくは B を被決定項とする属性があるかどうかを探す。すると

$D \rightarrow \{A, E\}$ があり, D が決まれば A が決まるので, $\{A, B\}$ の A を D に置き換えた $\{B, D\}$ も候補キーになる。

$\{B, C\}$ もしくは B, もしくは C を被決定項とする属性があるかどうかを探す。すると $\{A, B\} \rightarrow C$ が存在する。したがって $\{B, C\}$ の C を $\{A, B\}$ に置き換えてみると $\{A, B\}$ になり, これは既に候補キーになっている。

以上より, $\{A, B\}$, $\{B, C\}$, $\{B, D\}$ が候補キーになるので, 正解はウになる。念のため選択肢を順番に見ていく。

ア : $\{A, B\}$ が候補キーなので, $\{A, B, C\}$ は極小ではないため候補キーではない。

イ : $\{B, D\}$ も候補キーなので, この二つがすべてではない。

エ : $\{C, D\}$ では, A, E は一意に決定できるが B は一意に決定できない。加えて $\{A, B\}$ が足りない。

問 7 関係モデルの候補キーの説明のうち、適切なものはどれか。

- ア 関係 R の候補キーは関係 R の属性の中から選ばない。
- イ 候補キーの値はタブルごとに異なる。
- ウ 候補キーは主キーの中から選ぶ。
- エ 一つの関係に候補キーが複数あってはならない。

問 5 あるエンティティを関係データベース上に実装しようとしたとき、その主キーが多くの属性から構成される複合キーとなることが分かった。主キーとして扱う属性を少なくして扱いやすくしたい。この対応として、適切なものはどれか。

- ア 複合キーを構成する属性のうち、エンティティの性格を最もよく表している一つの属性を主キーとし、残りの属性を外部キー (foreign key) にする。
- イ 複合キーを構成する属性のうち、エンティティの性格を最もよく表している一つの属性を主キーとし、残りの属性を代替キー (alternate key) とする。
- ウ 連番などを値としてとる列を新たに設けて主キーとし、複合キーの代理キー (surrogate key) とする。
- エ 連番などを値としてとる列を新たに設けて主キーとし、複合キーを外部キー (foreign key) にする。

問7：正解（イ）

候補キーに関する問題（参照 第2章）。選択肢を順番に見ていく。

ア：関係Rの候補キーは、関係Rの属性の中にしかない。誤り。

イ：正しい。これが候補キーの定義である。

ウ：候補キーと主キーが逆。正しくは「主キーは候補キーの中から選ぶ」。誤り。

エ：候補キーは、一つの関係に複数存在する場合がある。午後1の設問でも「すべての候補キーを列挙せよ」というものがある。誤り。

問5：正解（ウ）

キーに関する問題。この問題では次の図のように、主キーが複合キーであり、複合キーの属性が多いので少なくする方法について問われている。この図をイメージしながら、選択肢を順番にチェックしていくけば容易に解答できる。

売上明細（地域コード, 店コード, 年月日, 商品コード, 数量, 単価, 金額）



主キー



売上明細（連番, 地域コード, 店コード, 年月日, 商品コード, 数量, 単価, 金額）

代理キー（surrogate key）

主キーを構成する属性のうち、どれか一つだけを主キーにすることはできない（そもそも、一つにすることしかできないから複合キーを用いて一意にしている）。そのため、選択肢ア、イは誤りである。残りの選択肢ウ、エはいずれも「連番などを値としてとる列を新たに設けて主キーとし」というが、その点については適切である。選択肢ウとエの違いは、元の主キー（複合キー）を何というかだが、これは代理キー（surrogate key）という。したがって、正解はウになる。

問6 関係データベースの表を設計する過程で、A表とB表が抽出された。主キーはそれぞれ列aと列bである。この二つの表の対応関係を実装する表の設計に関する記述のうち、適切なものはどれか。

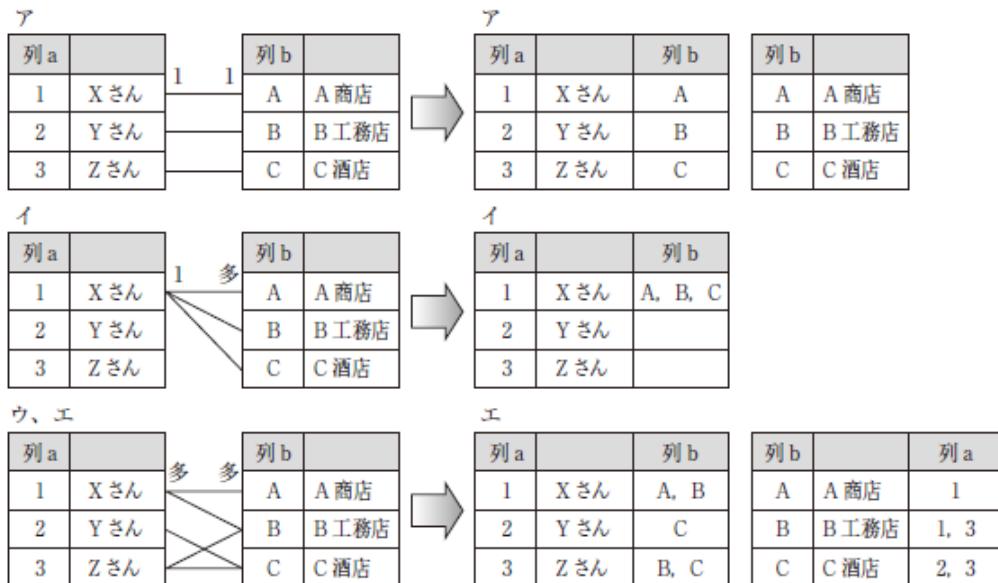
A	
a	

B	
b	

- ア A表とB表の対応関係が1対1の場合、列aをB表に追加して外部キーとしてもよいし、列bをA表に追加して外部キーとしてもよい。
- イ A表とB表の対応関係が1対多の場合、列bをA表に追加して外部キーとする。
- ウ A表とB表の対応関係が多対多の場合、新しい表を作成し、その表に列aか列bのどちらかを外部キーとして設定する。
- エ A表とB表の対応関係が多対多の場合、列aをB表に、列bをA表にそれぞれ追加して外部キーとする。

問 6：正解（ア）

データ構造設計に関する問題。2つの表 A, B を結合させるときの具体的な方法について問われている。選択肢の条件を図のようにイメージすれば、どの方法が適切か容易にわかる。



ア：1対1の関係の場合、列 b を表 A に組み入れて外部キーにしても（上図の“ア”的ところはその例）、列 a を表 B に組み入れて外部キーにしても問題ない。よって正解である。

イ：1対多の関係において、列 b を表 A に組み入れると図のように繰り返し列ができるてしまう（上図の“イ”的ところはその例）。よって適切ではない。

ウ：多対多の関係において、新しい表を作って、表 A と表 B を結合した場合、列 a と列 b の両方が外部キーとして必要になる。よって適切ではない。

エ：多対多の関係において、列 b を表 A に、列 a を表 B にそれぞれ組み入れると、上図の“エ”的例のようにどちらにも繰り返し項目ができるてしまう。よって適切ではない。

問 5 従業員番号と氏名と使用できるプログラム言語を管理するために，“従業員”表及び“プログラム言語”表を設計する。“プログラム言語を 2 種類以上使用できる従業員がいる。プログラム言語を全く使用できない従業員もいる。”という状況を管理する“プログラム言語”表の設計として、適切なものはどれか。ここで、実線の下線は主キーを表す。

〔従業員表〕

従業員（従業員番号, 氏名）

- ア プログラム言語（氏名, プログラム言語）
- イ プログラム言語（従業員番号, プログラム言語）
- ウ プログラム言語（従業員番号, プログラム言語）
- エ プログラム言語（従業員番号, プログラム言語）

問5：正解（工）

表の設計の問題。問題文の要件に合う“プログラム言語”表を設計する。問題文に記載されている要件は次の通り。

- ① プログラム言語を2種類以上使用できる従業員がいる。
- ② プログラム言語を全く使用できない従業員もいる。

要するに、従業員から見た使用可能プログラム言語が複数あり、使用可能プログラム言語から見た従業員も複数あるので、従業員とプログラム言語は多対多の関係になる。そのため、従業員とプログラム言語の連関エンティティを用いて、組み合わせとして保持する必要がある。その関係にあるのは選択肢エになる。

ア：上記要件の①を満たさない。

イ：一つのプログラム言語に1人の従業員しか登録できない。

ウ：上記要件の①を満たさない。

3.4 正規化

平成 19 年度・D B・午前

問23 データベースの正規化の目的のうち、適切なものはどれか。

- ア アクセスパスを固定して、データベースのアクセス速度を上げる。
- イ 属性間の従属関係を単純化して、更新時の物理的な I/O 回数を最小にする。
- ウ データの重複を排除して、重複更新を避け、矛盾の発生を防ぐ。
- エ テーブルの大きさを標準化して、データの参照速度を上げる。

平成 19 年度・D B・午前

問24 第 1, 第 2, 第 3 正規形とそれらの特徴 a ~ c の組合せとして、適切なものはどれか。

- a : どの非キー属性も、主キーの真部分集合に対して関数従属しない。
- b : どの非キー属性も、推移的に関数従属しない。
- c : 繰返し属性が存在しない。

	第 1 正規形	第 2 正規形	第 3 正規形
ア	a	b	c
イ	a	c	b
ウ	c	a	b
エ	c	b	a

問 23：正解ウ

正規化に関する問題。正規化の目的は明らかにウである。よって正解はウになる。

参照 ➔ 「2.1 正規化理論における重要な概念」

ア：正規化とは無関係。アクセス速度は、物理 I/O が増えるのでどちらかといえば逆に遅くなる

イ：単純化が目的ではないし、更新時の物理的な I/O 回数はどちらかといえば逆に増える

ウ：正解である

エ：平準化が目的ではないし、参照速度はどちらかといえば逆に下がる

問 24：正解ウ

正規化に関する標準的な問題。問題文の a, b, c をひとつずつ見ていく。

a：「真部分集合に対して関数従属しない」とは、すなわち、部分関数従属性を持たない、つまり、主キーに対して完全従属であるということ。これは第2正規形である

b：「推移的に関数従属しない」のは第3正規形の特徴である

c：「繰返し属性が存在しない」のは第1正規形の特徴である

以上より、正解はウになる。

参照 ➔ 「2.2.1 非正規形から第3正規形への変換」

問 5 第 1 正規形から第 5 正規形までの正規化に関する記述のうち、適切なものはどれか。

- ア 正規形にする分解は全て関数従属性が保存される。
- イ 正規形にする分解は全て情報無損失の分解である。
- ウ 第 3 正規形への分解では、情報無損失かつ関数従属性が保存される。
- エ 第 4 正規形から第 5 正規形への分解は自明な多値従属性が保存される分解である。

問 6 関係モデルにおいて、情報無損失分解ができ、かつ、関数従属性保存が成り立つ変換が必ず存在するものはどれか。ここで、情報無損失分解とは自然結合によって元の関係が復元できる分解をいう。

- ア 第 2 正規形から第 3 正規形への変換
- イ 第 3 正規形からボイス・コッド正規形への変換
- ウ 非正規形から第 1 正規形への変換
- エ ボイス・コッド正規形から第 4 正規形への変換

問5：正解（ウ）

正規化に関する問題。選択肢を順番に見ていく。一般的に、第3正規化までの情報無損失分解は、関数従属性保存分解である。関数従属性保存分解とは、テーブルを分解する際に、もともとあった関数従属性が分解先のテーブルのどちらかに保存されていることを言う。第3正規形からボイス・コッド正規形にする段階で、関数従属性を実装するテーブルが失われることがあるが、第3正規形ではデータ整合性を侵すような事態は起こらない。そこで、ボイス・コッド正規形まで正規度を上げずに、第3正規形にとどめることが多い。したがって、正解はウになる。

問6：正解（ア）

情報無損失分解に関する問題。一般的に、第3正規化までの情報無損失分解は、関数従属性保存分解である。関数従属性保存分解とは、テーブルを分解する際に、もともとあった関数従属性が分解先のテーブルのどちらかに保存されていることを言う。第3正規形からボイス・コッド正規形にする段階で、関数従属性が失われることがあるが、第3正規形ではデータ整合性を侵すような事態は起こらない。そこで、ボイス・コッド正規形まで正規度を上げずに、第3正規形にとどめることが多い。したがって、正解はアになる。

問 6 次の表を情報無損失分解したものはどれか。ここで、下線部は主キーを表す。

発注伝票（注文番号, 商品番号, 商品名, 商品単価, 注文数量）

- ア 発注（注文番号, 注文数量）
商品（商品番号, 商品名, 商品単価）
- イ 発注（注文番号, 注文数量）
商品（注文番号, 商品番号, 商品名, 商品単価）
- ウ 発注（注文番号, 商品番号, 注文数量）
商品（商品番号, 商品名, 商品単価）
- エ 発注（注文番号, 商品番号, 注文数量）
商品（商品番号, 商品名, 商品単価, 注文数量）

問6：正解ウ

情報無損失分解に関する問題。情報無損失分解（参照 ➤ 2.4 正規化）とは、分解した後の関係を自然結合すると、分解前の関係を完全に復元できるものをいう。そのため、選択肢ア～エをそれぞれ自然結合してみて、問題文のような元の形に戻るものを探せば良い。ただし、発注伝票の属性間の関係は一切説明されていないため、そこは常識で考えるしかない。

ア：結合条件になる属性がないので、自然結合できない。誤り。

イ：自然結合で元の状態に戻るよう見えるが、常識的に、商品名と商品単価は商品番号によって関数決定されるし、数量は注文番号と商品番号に関数従属しなければならない。したがって、選択肢ウの方が正しく、こちらは誤り。

ウ：自然結合で元の状態に戻る。これが正解。

エ：自然結合すると、注文数量が冗長になる。したがって誤り。

問 7 便名に対して、客室乗務員名の集合及び搭乗者名の集合が決まる関係“フライト”がある。関係“フライト”に関する説明のうち、適切なものはどれか。ここで、便名、客室乗務員名、搭乗者名の組が主キーになっているものとする。

フライト

便名	客室乗務員名	搭乗者名
BD501	東京建一	大阪一郎
BD501	東京建一	京都花子
BD501	横浜涼子	大阪一郎
BD501	横浜涼子	京都花子
BD702	東京建一	大阪一郎
BD702	東京建一	神戸順子
BD702	千葉建二	大阪一郎
BD702	千葉建二	神戸順子

- ア 関係“フライト”は、更新時異状が発生することはない。
- イ 関係“フライト”は、自明でない関数従属が存在する。
- ウ 関係“フライト”は、情報無損失分解が可能である。
- エ 関係“フライト”は、ボイス・コッド正規形の条件は満たしていない。

問7：正解（ウ）

正規化に関する問題。更新時異状、自明でない関数従属、情報無損失分解、ボイス・コッド正規形などに関する知識について問われている。選択肢を順番に見ていく。

ア：問題文の例を見て考えればいい。例えば、この例で「BD501 の客室乗務員が東京健一さんから千葉建二さんに変更したい」とき、1行目と2行目を同じように更新しないと更新時異状が発生してしまう。便名と客室乗務員名の組み合わせが複数行になる可能性があるため、更新時異状は発生する可能性がある。誤り。

イ：「自明ではない関数従属」とは、ビジネスルールや要件によって関数従属性が成立するものになる。つまり、ビジネスルール次第なので「便名が決まつたら客室乗務員が決まる」というような関係だ。ビジネスルール次第では決まらないこともあるからだ。逆に「自明な関数従属」とは、ビジネスルールに関係なく、今更言わなくてもそう決まっているよねというやうなもの。例えば「便名と客室乗務員名が決まつたら、客室乗務員名も決まるよね」というやうなもの。当たり前すぎるくらい当たり前のやうなもの。今回の場合、主キーの属性だけになるので、主キーが決まればすべての属性が決まる。つまり、この三つの属性の中に含まれるすべての関数従属は自明なものになる。「便名、客室乗務員名、搭乗者名」が決まれば、便名も、客室乗務員名も、それらの組み合わせもすべてが自明な関数従属になる。したがって自明でない関数従属は存在しない。

ウ：情報無損失分解とは、分解後の関係を自然結合したら、分解前の関係を復元できる分解のことをいう。関係“フライト”は主キーの三つの属性だけの関係なので、この関係“フライト”がある限り、どのように分解しても情報無損失分解が可能になる。したがって、これが正解になる。

エ：関係“フライト”は第3正規形である。また、主キーが一つというだけではなく候補キーも一つになる。第3正規形で候補キーが一つの場合は、ボイス・コッド正規形でもあるので、関係“フライト”はボイス・コッド正規形でもある（条件を満たしている）。誤り。

問 6 受注入力システムによって作成される次の表に関する記述のうち、適切なものはど
れか。受注番号は受注ごとに新たに発行される番号であり、項番は 1 回の受注で商品
コード別に連番で発行される番号である。

なお、単価は商品コードによって一意に定まる。

受注日	受注番号	得意先コード	項番	商品コード	数量	単価
2015-03-05	995867	0256	1	20121	20	20,000
2015-03-05	995867	0256	2	24005	10	15,000
2015-03-05	995867	0256	3	28007	5	5,000

- ア 正規化は行われていない。
- イ 第 1 正規形まで正規化されている。
- ウ 第 2 正規形まで正規化されている。
- エ 第 3 正規形まで正規化されている。

問6：正解（イ）

正規化に関する問題（参照 第2章）。問題の説明と表の情報から正規化がどこまで行われているか判断する。問題文の記述を整理すると、次のような。

候補キー						
受注日	受注番号	得意先コード	項目	商品コード	数量	単価
2015-03-05	995867	0256	1	20121	20	20,000
2015-03-05	995867	0256	2	24005	10	15,000
2015-03-05	995867	0256	3	28007	5	5,000

候補キー

- ・ 受注番号 → {受注日, 得意先コード}
- ・ {受注番号, 項番} → {商品コード, 数量, 単価}
- ・ {受注番号, 商品コード} → {項番, 数量, 単価}
- ・ 商品コード → 単価

商品コード

以上より、候補キーは {受注番号, 項番}, {受注番号, ~~商品番号~~} になる（この問題文の記述だと、1つの受注番号において、複数の商品コードは存在しないと考えられるので、同一受注番号内だと、項番と商品番号は1対1の関係にあることになる）。

商品コード

- ①繰り返し項目がないため第1正規形までは正規化されている。
- ②「受注番号は受注毎に新たに発行される番号」とあるので、受注日と得意先コードは、候補キーの一部である‘受注番号’によって一意に決まると考えられる。したがって第2正規形までは正規化されていない。

以上の2点より、第1正規形まで正規化されている。正解はイになる。

令和 2 年度・D B・午前 II

問 5 第 2 正規形であるが第 3 正規形でない表はどれか。ここで、講義名に対して担当教員は一意に決まり、所属コードに対して勤務地は一意に決まるものとする。また、() は繰返し項目を表し、実線の下線は主キーを表す。

ア

<u>学生番号</u>	講義名	担当教員	成績
2122	経済学	山田教授	優

イ

<u>社員番号</u>	氏名	入社年月日	電話番号
71235	山田 太郎	2001-04-01	03-1234-5678

ウ

<u>社員番号</u>	氏名	所属コード	勤務地
15547	小林 明	75T	東京

エ

<u>社員番号</u>	身長	体重	趣味
71234	170	62	{テニス, ゴルフ}

令和 4 年度・D B・午前 II

問 5 第 2 正規形である関係 R が、第 3 正規形でもあるための条件として、適切なものは
どれか。

- ア いかなる部分従属性も成立しない。
- イ 推移的関数従属性が存在しない。
- ウ 属性の定義域が原子定義域である。
- エ 任意の関数従属 $A \rightarrow B$ に関して、B は非キー属性である。

問 5：正解（ウ）

正規化に関する問題。第2正規形であるが第3正規形ではない表は、“非キー属性” → “非キー属性”がある（推移的関数従属が存在する）表である。問題文より、「所属コードに対して勤務地は一意に決まる」という記述があるので、選択肢ウが該当する。

ア：主キーの一部（講義名）に従属する項目（担当教員）があるため、第1正規形であるが第2正規形ではない。

イ：第3正規形である。

エ：繰返し項目（‘趣味’）があるので、非正規形である。

問 5：正解（イ）

正規化に関する問題。第3正規形の定義について問われている。

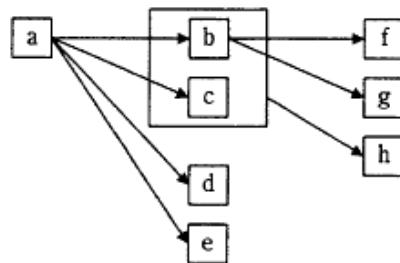
ア：第2正規形の定義である。誤り。

イ：正しい。

ウ：第1正規形の定義である。誤り。

エ：正規形とは無関係。誤り。

問24 属性 x の値によって属性 y の値が一意に定まることを、 $\boxed{x} \rightarrow \boxed{y}$ で表す。図に示される関係を、第 3 正規形の表として正しく定義しているものはどれか。ここで、 \boxed{x} の四角内に複数の属性が入っているものは、それら複数の属性すべての値によって、属性 y の値が一意に定まることを示す。



ア 表1 {a}
表2 {b, c, d, e}
表3 {f, g, h}

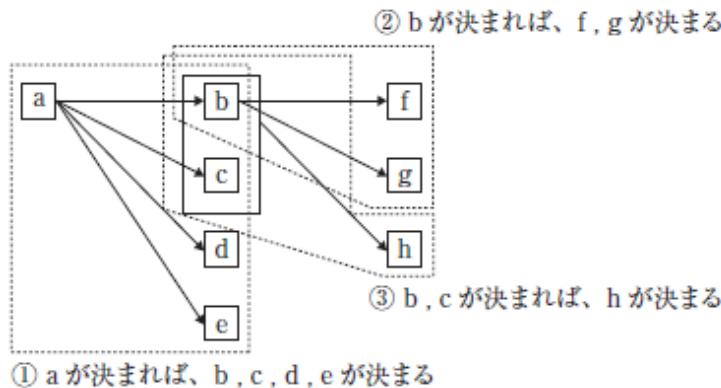
イ 表1 {a, b, c, d, e}
表2 {a, c}
表3 {b, e, f, g, h}

ウ 表1 {a, b, c, d, e}
表2 {b, c, f, g, h}
表3 {b, c, h}

エ 表1 {a, b, c, d, e}
表2 {b, f, g}
表3 {b, c, h}

問 24：正解工

関数従属性に関する問題。図の中から、下記のように関係性を抜粋する。



そして、①②③を表に展開する。ただし、わかりやすいように主キーには下線を引いている。
これに合致する選択肢はエになる。

① {a, b, c, d, e}

② {b, f, g}

③ {b, c, h}

参照 ➤ 「2.1.1 関数従属性」

問 5 属性間の従属関係を次のように表記するとき、属性 a～e で構成される関係を第 3 正規形にしたものはどれか。

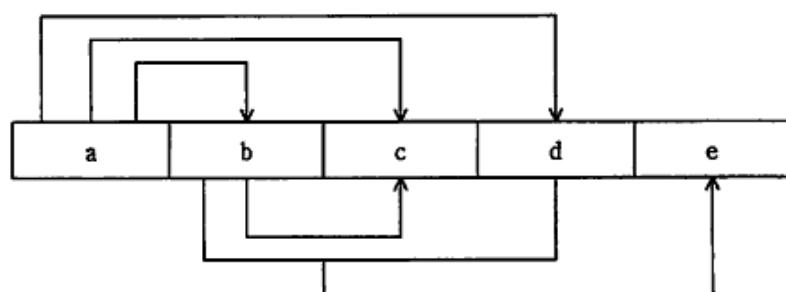
[属性間の従属関係]

(1) 属性 X の値が与えられると、
属性 Y の値を一意に決めること
ができる。

(2) 属性 X と属性 Y の二つの値が
与えられると、属性 Z の値を一
意に決めることができる。



[正規化する関係]



- ア

a	b	c	d
---	---	---	---

b	d	e
---	---	---
- イ

a	b	c	d
---	---	---	---

b	d	e
---	---	---

b	c
---	---
- ウ

a	b	d
---	---	---

b	d	c	e
---	---	---	---
- エ

a	b	d
---	---	---

b	c
---	---

b	d	e
---	---	---

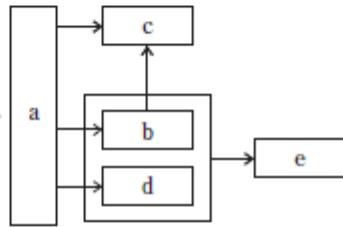
問5：正解工

正規化（参照▶第2章 正規化理論）に関する問題。従属関係を順次ピックアップしていく。

①矢印は全部で5本。

$a \rightarrow b$
 $a \rightarrow c$
 $a \rightarrow d$
 $b \rightarrow c$
 $b, d \rightarrow e$

これを図示したものが



この図からも明らかなように、主キーは a になる。繰り返し項目も、部分キーもないで第2正規形までは完了している。後は、推移関数従属をなくしていけば良い。

②推移関数従属 $(b, d) \rightarrow e$ を分割していく。

(a, b, c, d)
 (b, d, e)

③推移関数従属 $b \rightarrow c$ を分割していく。

(a, b, d)
 (b, d, e)
 (b, c)

したがってエが正解になる。

問8 次の表を、第3正規形まで正規化を行った場合、少なくとも幾つの表に分割されるか。ここで、顧客の1回の注文に対して1枚の受注伝票が作られ、顧客は1回の注文で一つ以上の商品を注文できるものとする。

受注番号	顧客コード	顧客名	受注日	商品コード	商品名	単価	受注数	受注金額
1055	A7053	鈴木電気	2023-07-01	T035	テレビA	85,000	10	850,000
1055	A7053	鈴木電気	2023-07-01	K083	無線スピーカーA	23,000	5	115,000
1055	A7053	鈴木電気	2023-07-01	S172	Blu-rayプレイヤーB	78,000	3	234,000
2030	B7060	中村商会	2023-07-03	T050	テレビB	90,000	3	270,000
2030	B7060	中村商会	2023-07-03	S172	Blu-rayプレイヤーB	78,000	10	780,000
3025	C9025	佐藤電気	2023-07-03	T035	テレビA	85,000	3	255,000
3025	C9025	佐藤電気	2023-07-03	K085	無線スピーカーB	25,000	2	50,000
3025	C9025	佐藤電気	2023-07-03	S171	Blu-rayプレイヤーA	50,000	8	400,000
3090	B7060	中村商会	2023-07-04	T050	テレビB	90,000	1	90,000
3090	B7060	中村商会	2023-07-04	T035	テレビA	85,000	2	170,000

ア 2

イ 3

ウ 4

エ 5

問9：正解ウ

正規化に関する標準的な問題（参照 ➤ 2.4 正規化）。まず、この問題の主キーを明確にしなければならない。問題文の記述と表の例ですべてを一意にする最小のもの、すなわち主キーは、{受注番号、商品コード} だとわかる。それを明確にできたら正規化を進める。

<第1正規形 → 第2正規形> (部分関数従属の分解)

- 受注番号 → {顧客コード、顧客名、受注日}, 商品コード → {商品名、単価} より,
- ・受注 (受注番号, 顧客コード, 顧客名, 受注日)
 - ・受注明細 (受注番号, 商品コード, 受注数, 受注金額)
 - ・商品 (商品コード, 商品名, 単価)

<第2正規形 → 第3正規形> (推移関数従属の分解)

さらに、“受注”には、受注番号 → 顧客コード → 顧客名という推移的関数従属性があるため、これも分解する。

- ・受注 (受注番号, 顧客コード, 受注日)
- ・受注明細 (受注番号, 商品コード, 受注数, 受注金額)
- ・商品 (商品コード, 商品名, 単価)
- ・顧客 (顧客コード, 顧客名)

以上より、四つの表に分解されるので、正解はウになる。

問 8 第 3 正規形において存在する可能性のある関数従属はどれか。

- ア 候補キーから繰返し属性への関数従属
- イ 候補キーの真部分集合から他の候補キーの真部分集合への関数従属
- ウ 候補キーの真部分集合から非キー属性への関数従属
- エ 非キー属性から他の非キー属性への関数従属

問 3 関係 R (A, B, C) の候補キーが {A, B} と {A, C} であり, $\{A, B\} \rightarrow C$ 及び $C \rightarrow B$ の関数従属性があるとき, 関係 R はどこまでの正規形の条件を満足しているか。

- | | |
|-----------|--------------|
| ア 第 1 正規形 | イ 第 2 正規形 |
| ウ 第 3 正規形 | エ ボイス・コッド正規形 |

問 8：正解（イ）

正規化に関する問題。第 3 正規形に関して存在する可能性のある関数従属性について問われている。選択肢を順番に見ていく。

ア：候補キーから繰り返し属性への関数従属性は、第 1 正規形にした段階で取り除かれているため、存在しない。

イ：候補キーの真部分集合から非キー属性への関数従属性は、第 2 正規形にする段階で除去されているが、他の候補キーの真部分集合への関数従属性は取り除かれていな。したがって、これは第 3 正規形になても存在する可能性がある。これが正解。

ウ：候補キーの真部分集合から非キー属性への関数従属性は、第 2 正規形にする段階で除去されている。存在しない。

エ：非キー属性から非キー属性への関数従属性、いわゆる推移的関数従属性は、第 3 正規形にする段階で取り除かれる。存在しない。

問 3：正解（ウ）

正規化に関する問題。関係 R には、非単純属性（繰り返し項目）、部分従属性のいずれもない。加えて、 $\{A, B\} \rightarrow C, C \rightarrow B$ という関数従属性については、B が $\{A, B\}$ の部分集合なので推移的関数従属性には該当しないので、第 3 正規形の条件は満たしている。

しかし、今回のケースでは①候補キーが複数あり、②その中に、候補キーの一部である C が決定項でもあるので（すなわち、全ての決定項が候補キーではないので）、ボイス・コッド正規形ではないことになる。

以上より、関係 R は第 3 正規形になるため“ウ”が正解になる。

問 7 教員の担当科目と給与を管理する“科目一教員”表を更新するときに発生する問題はどれか。ここで、科目番号を主キーとし、基本給は科目によらず教員ごとに決まっているものとする。

科目一教員

科目番号	科目名	教員番号	担当教員	単位	基本給
2761	一般システム理論	8823	田中亮	2	180
2762	問題形成と問題解決	6673	佐藤永吉	2	250
2763	情報システム開発の経済性	6654	小林正路	2	400
2864	一般システム理論	7890	大野俊郎	2	230
2865	情報システムの都市計画法	4664	斎藤秀夫	4	320
3966	UML モデリング	8823	田中亮	4	180

- ア ある教員が唯一担当していた科目的行を削除すると、その教員の基本給データだけが残ってしまう。
- イ ある教員の基本給を変更するには、該当する行を 1 件ずつコミットしないとデータの不整合が生じる。
- ウ 担当科目のない教員の基本給を登録するときは、一つ以上の科目を削除しなければならない。
- エ 複数の科目を担当する教員の基本給を変更するときは、担当するすべての科目について変更しないとデータの不整合が生じる。

問7：正解工

更新時異状（参照 ➔ 2.4.8 更新時異状）に関する問題。問題文に書かれている条件と、表の例を見て属性間の関係を整理する。すると、この表の主キーは科目番号なので、推移関数従属（科目番号→教員番号→{担当教員、基本給}）を含んでいることがわかる。その場合に“発生する問題”を想定しながら、選択肢を順番に見ていく。

ア：ある教員が唯一担当していた科目的行を削除すると、その教員の持つ属性“担当教員”と“基本給”的データが失われてしまう。正規化していないケースの典型的問題点だ。したがって「基本給データだけが残ってしまう」というのは正反対で誤り。

イ：問題文で紹介されている例を使って説明すると、教員番号“8823”的データは2件存在する。つまり教員と基本給の関係が複数行存在することもある。その場合、基本給を変更するには、該当する行すべてに変更をかけなければ不整合が発生してしまう。だから、コミットは1件ずつではなく、すべての変更が完了した後に1度だけ行わなければならない。したがって誤り。

ウ：担当科目的ない教員の基本給を登録することはできない（主キーはNULL値不可なので）。したがって誤り。

エ：選択肢イの逆。正しい。

memo

4

第4章

データベースの実装

ここでは、データベースの実装に必要な知識の中から、特に頻出する重要な項目をまとめた。これらは、午前Ⅱ・午後Ⅰ・午後Ⅱのいずれにおいても求められる、最も重要なキーワードである。

1. 表領域
2. トランザクション管理機能
3. 障害回復機能
4. 索引（インデックス）
5. 分散データベース
6. テーブルの物理分割
7. クラスタ構成

データベーススペシャリストの仕事

平成 19 年度・D B・午前

問20 DA (Data Administrator) と DBA (Database Administrator) を別々に任命した場合の DA の役割として、適切なものはどれか。

- ア 業務データ量の増加傾向を把握し、ディスク装置の増設などを計画・実施する。
- イ システム開発の設計工程では、主に論理データベース設計を行い、データ項目を管理し標準化する。
- ウ システム開発のテスト工程では、主にパフォーマンスチューニングを担当する。
- エ システム障害が発生した場合には、データの復旧や整合性のチェックなどを行う。

問20：正解イ

DAとDBAの役割の違いを考える。DA（データ管理者）は、情報システム全体のデータ資源を管理する役割を持ち、システム開発工程においては分析・論理設計といった上流フェーズを担当する。一方、DBA（データベース管理者）は、データの器となるデータベースの構築と維持を行う役割を持ち、システム開発工程では、実装・運用・保守といった中流以降のフェーズを担当する。その基準で選択肢を順番に見ていく。

参照 「6.1 データベーススペシャリストの業務」

ア：ハードウェアの増強計画は保守の話なので、DBAもしくはハードウェア専任者の役割である

イ：論理データベースの設計はDAの役割になる。これが正解

ウ：パフォーマンスチューニングは物理データベース設計なので実装時の話。したがってDBAの仕事である

エ：障害時の復旧や整合性チェックは保守の話。したがってDBAの仕事である

ANSI/SPARC 3層スキーマアーキテクチャ

平成 29 年度・D B・午前Ⅱ

問 1 データベースの 3 層スキーマアーキテクチャに関する記述として、適切なものはどれか。

- ア 概念スキーマは、内部スキーマと外部スキーマの間に位置し、エンティティやデータ項目相互の関係に関する情報をもつ。
- イ 外部スキーマは、概念スキーマをコンピュータ上に具体的に実現させるための記述であり、データベースに対して、ただ一つ存在する。
- ウ サブスキーマは、複数のデータベースを結合した内部スキーマの一部を表す。
- エ 内部スキーマは、個々のプログラム又はユーザの立場から見たデータベースの記述である。

平成 21 年度・D B・午前Ⅱ

問 1 ANSI/SPARC 3 層スキーマに関する記述として、適切でないものはどれか。

- ア ANSI/SPARC 3 層スキーマの意義は、物理的データ独立性及び論理的データ独立性を確保することである。
- イ 外部スキーマは、概念スキーマが変化しても応用プログラムができるだけ影響を受けないようにするための考え方である。
- ウ 関係データベースのビューやネットワークデータベースのサブスキーマは、概念スキーマに相当する。
- エ 内部スキーマは、概念スキーマをコンピュータ上に実装するための記述である。

問 1：正解（ア）

3層スキーマに関する問題（参照▶第4章）。3層スキーマは、利用者に近い層から順に、外部スキーマ、概念スキーマ、内部スキーマに分けられる。外部スキーマは「ユーザがアクセスするスキーマ」、概念スキーマは「現実世界のデータ全体」、内部スキーマは「コンピュータ上に実装するためのスキーマ」を、それぞれ表している。選択肢を順番に見ていく。

ア：正しい。

イ：「コンピュータ上に具体的に実現させるための記述」は、外部スキーマではなく内部スキーマである。誤り。

ウ：「サブスキーマ」は、外部スキーマの別の言い方である。誤り。

エ：「個々のプログラム又はユーザの立場から見たデータベースの記述」は、内部スキーマではなく外部スキーマである。誤り。

問 1：正解ウ

ANSI/SPARC3層スキーマ（参照▶用語集）に関する知識を問う問題。関係データベースのビューやネットワークデータベースのサブスキーマは外部スキーマに相当するため、ウの記述は誤っている。適切でないものはどれかという問であるため、ウが正解である。

問43 DBMS の記憶管理に関する記述のうち、最も適切なものはどれか。

- ア 関係データベースの参照制約を実現する処理の高速化に連結リストを用いることが多い。
- イ 関係データベースの一つの表は、ページと呼ばれるデータベースの格納単位内に収まるよう管理される。
- ウ クラスタリングとは、磁気ディスク装置へのアクセス効率向上を目的としたデータ格納手法である。
- エ バッファ管理では、通常 FIFO (First In First Out) と呼ばれる手法によって、主記憶上のデータ領域を管理する。

問 43：正解ウ（平成 14 年度の問 42 とほぼ同じ問題）

DBMS の記憶管理に関する問題。選択肢を順番に見ていく。

ア：この処理の高速化には、連結リストではなく、別の方針（外部キーにインデックスを作成する方法）が用いられる。よって誤り

イ：「ページ」ではなく、表領域（商用 DBMS により呼称はまちまちである）である。よって誤り

ウ：クラスタリングとは、一緒に用いることの多いデータの集まり（クラスタ）を、前もって同じブロックと一緒に格納しておく手法である。こうすることで、磁気ディスクへのアクセス効率が向上する。よって正解である

エ：FIFO ではなく、LRU（Least Recently Used）である。よって誤り

4.2 トランザクション管理機能

平成 29 年度・D B・午前Ⅱ

問16 トランザクションの ACID 特性の説明として、適切なものはどれか。

- ア トランザクションでは、実行すべき処理が全て行われるか、何も処理が行われないかという状態の他に、処理の一部だけが行われるという状態も発生する。
- イ トランザクションの実行完了後でも障害の発生によって実行結果が失われることがある。
- ウ トランザクションの実行の結果が矛盾した状態になることはない。
- エ トランザクションは相互に関連しており、同時に実行される他のトランザクションの影響を受ける。

令和 4 年度・D B・午前Ⅱ

問15 トランザクションの ACID 特性のうち、原子性 (atomicity) の記述として、適切なものはどれか。

- ア データベースの内容が矛盾のない状態であること
- イ トランザクションが正常終了すると、障害が発生しても更新結果はデータベースから消失しないこと
- ウ トランザクションの処理が全て実行されるか、全く実行されないかのいずれかで終了すること
- エ 複数のトランザクションを同時に実行した場合と、順番に実行した場合の処理結果が一致すること

問 16：正解（ウ）

ACID 特性（参照 ➤ 第 4 章）。に関する問題。選択肢を順番に見ていくう。

- ア：「処理の一部だけが行われる状態が発生する」と原子性に反するので誤り。
- イ：「障害の発生によって実行結果が失われてしまう」と耐久性に反するので誤り。
- ウ：一貫性の説明。正解。
- エ：「他のトランザクションの影響を受ける」と独立性に反するので誤り。

問 15：正解（ウ）

トランザクションの ACID 特性に関する問題。原始性（atomicity）について問われている。選択肢を順番に見ていくう。

- ア：一貫性（Consistency）の説明。
- イ：耐久性（Durability）の説明。
- ウ：これが原始性（Atomicity）の説明。正解。
- エ：直列化可能性の説明。ACID 特性の中だと、他のトランザクションからの影響を受けないという点では独立性（Isolation）になる。

問14 トランザクションの ACID 特性のうち、一貫性 (consistency) の説明はどれか。

- ア 整合性の取れたデータベースに対して、トランザクション実行後も整合性が取れている性質である。
- イ 同時実行される複数のトランザクションは互いに干渉しないという性質である。
- ウ トランザクションは、完全に実行が完了するか、全く実行されなかつたかの状態しかとらない性質である。
- エ ひとたびコミットすれば、その後どのような障害が起こっても状態の変更が保たれるという性質である。

問41 トランザクションの ACID 特性のうち、耐久性 (durability) に関する記述として、適切なものはどれか。

- ア 正常に終了したトランザクションの更新結果は、障害が発生してもデータベースから消失しないこと
- イ データベースの内容が矛盾のない状態であること
- ウ トランザクションの処理がすべて実行されるか、全く実行されないかのいずれかで終了すること
- エ 複数のトランザクションを同時に実行した場合と、順番に実行した場合の処理結果が一致すること

問 14：正解ア

NEW

参照 第4章 重要キーワード「6 トランザクション管理機能」

トランザクションの ACID 特性に関する問題。今回は、一貫性について問われている。一貫性が問われるるのは過去の DB 午前問題では平成 14 年まで遡っても出題されていないが、ACID 特性に関しては、正誤問題（H15）、耐久性（H20）、原子性（H16）、独立性（H14）など出題頻度は高い。ちなみに一貫性とは、トランザクション実行後も実行前と一貫して整合性が取れている性質のことを言う。したがって正解はアになる。

イ：独立性の説明。誤り。

ウ：原子性の説明。誤り。

エ：耐久性の説明。誤り。

問 41：正解ア

ACID 特性に関する問題。耐久性について問われている。ちなみに、耐久性とは、障害が発生したとしても失われないような性質のことをいう。それを踏まえて選択肢を順番に見ていく。

参照 「4.1.1 トランザクションと ACID 特性」

ア：耐久性の説明。正解である

イ：一貫性の説明。適切でない

ウ：原始性の説明。適切でない

エ：独立性の説明。適切でない

問13 DBMS の排他制御機能に関する記述のうち、適切なものはどれか。

- ア 排他制御機能によって、同時実行処理でのデータの整合性を保つことができる。
- イ 排他制御機能の使用によって、デッドロックを防止できる。
- ウ 排他制御は DBMS が自動的に行い、アプリケーションプログラムからロック、アンロックの指示はできない。
- エ パッチによる更新処理では排他制御を行う必要はない。

問 13：正解ア

データベースの排他制御に関する問題（参照▶ 4.1.3 排他制御（同時実行制御））。選択肢を順番に見ていく。

ア：正解である。

イ：排他制御機能とは、複数のトランザクションを同時に実行させても、データの整合性がとれるようにする機能のこと。ロックを使用する。そのため、“排他制御を行うからこそ” デッドロックは発生する。よって誤りである。

ウ：排他制御は、必要に応じてアプリケーションプログラムから指示するものである。よって誤りである。

エ：バッチによる更新処理でも、複数トランザクションが実行されたり、オンライン処理と同時に実行されたりするような場合、排他制御が必要になる。よって誤りである。

問15 トランザクション T_1 と T_2 の並行実行における不整合検索異状 (inconsistent retrieval anomaly) の説明はどれか。

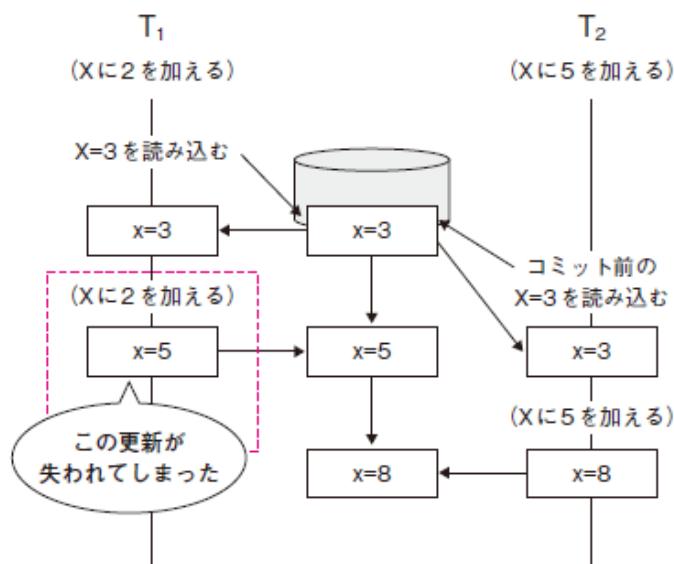
- ア T_1 が x を更新した後コミットする前に、 T_2 も x を更新すること
- イ T_1 と T_2 が同時にデータ x を読み、 T_1 は x を更新する。 T_2 は更新前の x の値に基づいてデータ y を更新することによって、 y の値が x の値に基づかなくなること
- ウ 先行する T_1 が更新したデータ x を、後続の T_2 が読んで処理を行うとき、 T_1 がロールバックすると、 T_2 もロールバックしなければならなくなること
- エ 先行する T_1 が更新したデータ x を、後続の T_2 が読んで処理を行った後に、 T_1 がロールバックすることによって、 T_2 は結果的に誤った処理になること

問 15：正解（イ）

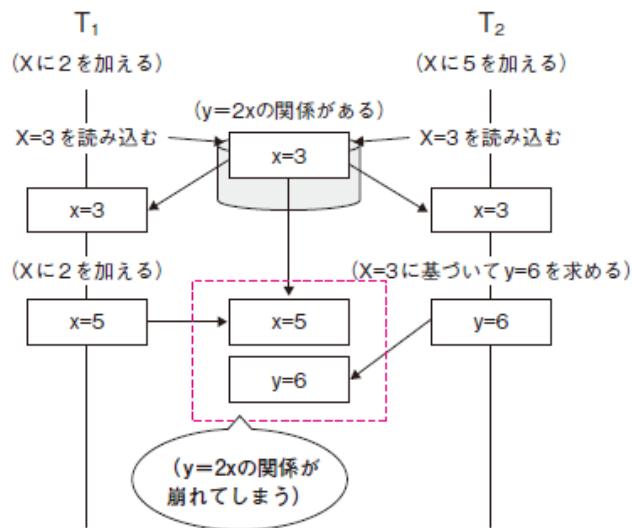
データベースの同時実行制御に関する問題。不整合検索異状 (inconsistent retrieval anomaly)について問われている。不整合検索異状とは、データ項目間に意味的な関係（依存関係）があるとき ($y = 2x$ など) に、並行実行中に読み取った値が一貫しておらず、当初の意味的な関係（依存関係）を満たさなくなる異常である。READ COMMITTED 以下の隔離レベルではこのような事象が発生する可能性がある。REPEATABLE READ や SERIALIZABLE であれば、 T_2 は古い x を見た時点でその値が変化しないことが保証される（または排他制御される）ので発生しない。

こうした性質を念頭に置きながら、選択肢を順番に見ていく。わかりにくければ、各選択肢で例を挙げて考えてみたらいいだろう。

ア：二つのトランザクション (T_1, T_2) がある。例えば、 T_1 では X に + 2 をする、 T_2 では X に + 5 をする処理だとしよう。そして T_1 と T_2 が同時に $x = 3$ を読み込む。そして T_1 が $x = 3 + 2 = 5$ に更新する。このコミット前に T_2 は「 $x = 3$ 」に対して + 5 をして 8 とする。本来であれば、 T_1 で + 2 を行い、その後 T_2 で + 5 をすると $x = 10$ にならなければいけないところ、8 にしかならず、 T_1 で + 2 を行った処理が失われた形になる。これはロストアップデートという。誤り。



イ : $y = 2x$ という関係があるとする。 T_1 と T_2 が同時にデータ $x = 3$ を読みこんだとしよう。そして T_1 が $X = 5$ に更新したとする。「 T_2 は更新前の x の値に基づいてデータ y を更新する」ということなので、読み込んだ $x = 3$ （すなわち T_1 による $x = 5$ に更新する前の値）に基づいてデータ y を更新すると、 $y = 6$ になる。その結果、 $y = 2x$ という関係なのに、 $x = 5 \cdot y = 6$ になってしまう。本来なら $x = 3$ と $y = 6$ か、 $x = 5$ と $y = 10$ でなければならない。これが不整合検索異状である。下図のように図示してみれば、ロストアップデートとの違いがはっきりするだろう。



ウとエ：これは「先行する T_1 が更新したデータ x 」を基に「後続の T_2 がデータ x を読んで処理をする」ということなので、この二つのトランザクション T_1 と T_2 には依存関係（前後関係・連続性）がある。そのため、 T_1 をロールバックした場合、当然 T_2 もロールバックをしないといけない。この選択肢の解説になっているかどうかは微妙だが、このように 1 つのトランザクションのロールバックが他のトランザクションに波及する現象をカスケードロールバックもしくは連鎖的アボートという。選択肢ウのような連続性を必要とするトランザクションにおいては、連鎖的アボートをしなければ選択肢エのような異常（非回復スケジュール）になってしまう。なお、選択肢ウもエも、コミット前のデータを読み込んだことで異常が発生する点だけでいうとダーティリードと言える。

memo

問17 二つのトランザクション T1 と T2 を並列に実行した結果が、T1 の完了後に T2 を実行した結果、又は T2 の完了後に T1 を実行した結果と等しい場合、このトランザクションスケジュールの性質を何と呼ぶか。

ア 一貫性

イ 原子性

ウ 耐久性

エ 直列化可能性

問 17：正解（工）

ACID 特性を担保するためには、たとえ複数のトランザクションを同時実行したとしてもそれを個別に実行したときと同様の結果を返さなければならない。この性質をトランザクションスケジュールにおける直列可能性という。

今回の問題においても、「二つのトランザクション T1 と T2 を並列に実行した結果が」個別に実行したときの結果と等しいとあるので、トランザクションスケジュールの直列可能性の性質についての言及である。よって正解はエ。ほかの選択肢は ACID 特性の一つである。

問11 トランザクションの直列化可能性 (serializability) の説明はどれか。

- ア 2 相コミットが可能であり、複数のトランザクションを同時実行できる。
- イ 隔離性水準が低い状態であり、トランザクション間の干渉が起こり得る。
- ウ 複数のトランザクションが、一つずつ順にスケジュールされて実行される。
- エ 複数のトランザクションが同時実行された結果と、逐次実行された結果とが同じになる。

問19 トランザクション管理の直列化可能スケジュールを説明したものはどれか。

- ア デッドロックの発生を最小限に抑えるために、可能な限りトランザクションを直列に実行するスケジュール
- イ トランザクションの前後関係を考慮しながら、処理時間が最短になるようにトランザクションを直列に実行するスケジュール
- ウ トランザクションを順番に実行した場合と同じ結果をもつように、並行実行するスケジュール
- エ 読取り専用トランザクションは並行実行するが、更新を行うトランザクションは直列に実行するスケジュール

問 11：正解（エ）

トランザクションに関する問題。トランザクションの直列化可能性とは、複数のトランザクションを直列に（順番に）実行したときと同じ結果（=逐次実行された結果）を持つように並行して（同時）実行できる特性のことを使う。したがって、正解は（エ）になる。

ア：2相コミットは分散データベースの機能。誤り。

イ：直列化可能性は、隔離性水準が高い SERIALIZABLE で保証される。隔離性水準が低い場合には保証されない。誤り。

ウ：逐次処理の説明。誤り。

問 19：正解（ウ）

NEW（但し、類似問題多数）

[参照 第4章 重要キーワード「5.トランザクション管理機能」](#)

トランザクション管理の直列化可能スケジュールとは、トランザクションを直列に（順番に）実行したときと同じ結果を持つように並行して実行させるスケジュールのことを使う。本当に直列に実行するわけではない。したがって、正解は（ウ）になる。

ア：「トランザクションを直列に実行する」と書いてるので、単なる「直列したスケジュール」、すなわち順番に配置されたスケジュールのこと。誤り。

イ：ここも「トランザクションを直列に実行するスケジュール」と書いている。誤り。

ウ：正解。“直列化可能”とは、直列に（順番に）実施した場合と同じ結果になる、並行処理のことを指す。

エ：読み取りや更新で分けない。誤り。

問19 二つのトランザクション T1, T2 が、データ a, b に並行してアクセスする。T1, T2 の組合せのうち、直列可能性を保証できるものはどれか。ここで、トランザクションの各操作の意味は次のとおりとする。

- LOCK x : データ x をロックする
- READ x : データ x を読み込む
- WRITE x : データ x を書き出す
- UNLOCK x : データ x をアンロックする

ア

T1	T2
READ a	READ a
LOCK a	LOCK a
LOCK b	LOCK b
$a = a + 3$	$a = a + 3$
WRITE a	WRITE a
READ b	READ b
$b = b + 5$	$b = b + 5$
WRITE b	WRITE b
UNLOCK a	UNLOCK a
UNLOCK b	UNLOCK b

イ

T1	T2
LOCK a	LOCK a
READ a	READ a
$a = a + 3$	$a = a + 3$
WRITE a	WRITE a
UNLOCK a	UNLOCK a
LOCK b	LOCK b
READ b	READ b
$b = b + 5$	$b = b + 5$
WRITE b	WRITE b
UNLOCK b	UNLOCK b

ウ

T1	T2
LOCK a	LOCK a
READ a	READ a
$a = a + 3$	LOCK b
WRITE a	READ b
UNLOCK a	UNLOCK a
LOCK b	UNLOCK b
READ b	UNLOCK b
$b = b + 5$	
WRITE b	
UNLOCK b	

工

T1	T2
LOCK a	LOCK a
READ a	READ a
$a = a + 3$	LOCK b
WRITE a	READ b
LOCK b	UNLOCK b
READ b	UNLOCK a
$b = b + 5$	
WRITE b	
UNLOCK b	
UNLOCK a	

問 19：正解（工）

過去問題（平成19年・DB午前 問42など）

参照 第4章 重要キーワード「5 トランザクション管理機能」

トランザクションT1, T2で直列可能性を保証するには、処理実行前に必要な資源にロックをかけ、処理実行後に（ロックをかけた資源を）アンロックする2相ロックプロトコルを用いる。

ア：T1では、aのREAD後にaをLOCKしているため、その間にT2が実行される可能性がある。よって直列可能性は保証できない。

イ：T1の“UNLOCK a”と“LOCK b”的間にT2がすべて実行される可能性がある。よって直列可能性は保証できない。

ウ：T1の“UNLOCK a”と“LOCK b”的間、すなわちT1の途中にT2が実行される（T2の値がT1途中の値になる）可能性がある。よって直列可能性は保証できない。

エ：スタートはどちらも“LOCK a”である。T1が先に“LOCK a”を実行した場合、T2は、“UNLOCK a”でT1の実行終了まで待たなければならない。一方、T2が先に“LOCK a”を実行した場合も、T1は、T2の最終行“UNLOCK a”を待たなければならないため、どちらのケースでも直列可能性は保証できる。

問14 RDBMS のロックに関する記述のうち、適切なものはどれか。ここで、X、Y はトランザクションとする。

- ア X が A 表内の特定行 a に対して共有ロックを獲得しているときは、Y は A 表内の別の特定行 b に対して専有ロックを獲得することができない。
- イ X が A 表内の特定行 a に対して共有ロックを獲得しているときは、Y は A 表に対して専有ロックを獲得することができない。
- ウ X が A 表に対して共有ロックを獲得しているときでも、Y は A 表に対して専有ロックを獲得することができる。
- エ X が A 表に対して専有ロックを獲得しているときでも、Y は A 表内の特定行 a に対して専有ロックを獲得することができる。

問15 2相ロック方式に従うトランザクションに関する記述のうち、適切なものはどれか。

- ア デッドロックが発生することはない。
- イ 同一トランザクション内であれば、アンロック後にロックを行うことができる。
- ウ トランザクションが利用するロックは、専有ロックに限られる。
- エ トランザクションの競合直列可能性が保証される。

問 14：正解（イ）

データベースの同時実行制御に関する問題（参照▶第4章：WEB掲載）。専有ロックと共有ロックについて問われている。選択肢を順番に見ていく。

ア：行レベルのロックができているので、行が異なれば共有ロックでも専有ロックでも可能なので誤り。

イ：行レベルの共有ロックをかけているので、その行を含む A 表全体に専有ロックをかけようとしても、特定行 a には専有ロックをかけられないので、A 表に対して専有ロックを獲得することができない。正しい。

ウ：共有ロックをかけている表に、専有ロックを後からかけることはできない。誤り。

エ：ある表に専有ロックをかけた場合、後から同表の特定行に専有ロックをかけることはできない。誤り。

問 15：正解（エ）

2相ロック方式（参照▶第4章）に関する問題。選択肢を順番に見ていく。

ア：デッドロックが発生する可能性はある。誤り。

イ：2相ロック方式では、同一トランザクション内で、一度アンロックした後、再度ロックをかけることは許されない。

ウ：ロックには共有ロックと専有ロックがある。

エ：正解

問34 トランザクションの同時実行制御である 2 相ロックングプロトコルに関する記述として、適切なものはどれか。

- ア 共有ロック、占有ロックの概念はない。
- イ 異なるテーブルであれば、すべてのロックが完了する前にアンロックを行ってもよい。
- ウ デッドロックが発生することがある。
- エ 読込みを行うトランザクションは、ロックする必要がない。

令和 5 年度・D B・午前 II (※R3-13 同じ問題)

問12 2 相ロック方式を用いたトランザクションの同時実行制御に関する記述のうち、適切なものはどれか。

- ア 全てのトランザクションが直列に制御され、デッドロックが発生することはない。
- イ トランザクションのコミット順序は、トランザクション開始の時刻順となるよう制御される。
- ウ トランザクションは、自身が獲得したロックを全て解除した後にだけ、コミット操作を実行できる。
- エ トランザクションは、必要な全てのロックを獲得した後にだけ、ロックを解除できる。

問 34：正解ウ（平成 15 年度の問 36 とほぼ同じ問題）

2相ロックングプロトコル（[参照](#)▶用語集）に関する頻出問題。選択肢を順番に見ていく。

[参照](#)▶「4.3.3 分散トランザクション制御」

ア：2相ロックングプロトコルでは、共有ロック、占有ロックの概念を用いている

イ：2相ロックングプロトコルでは、アンロックした後、再度ロックをかけることは許されない

ウ：正解

エ：読み込みを行うトランザクションでも、読み込んでいる間に値を変化させたくないケースなど、ロックをかける場合がある

問 12：正解（エ）

トランザクションの同時実行制御に関する問題。2相ロックについて問われている。2相ロックは、トランザクションの直列実効性を保障するための仕組みであり、トランザクションは必要なロック獲得命令を全て実行した後にだけ、ロック解除命令を実行できる。したがって、正解は（エ）になる。

ア：ロック獲得の順序およびタイミングによっては、デッドロックが発生する可能性がある。

イ：実行開始の時刻順になるとは限らない。

ウ：ロック解除命令を実行するのは、コミットまたはロールバックした後である。

問18 DBMS の多版同時実行制御 (MVCC) に関する記述として、適切なものはどれか。

- ア 同時実行される二つのトランザクションのうち、先発のトランザクションがデータを更新し、コミットする前に、後発のトランザクションが同じデータを参照すると、更新前の値を返す。
- イ トランザクションがデータを更新する前に専有ロックを、参照する前に共有ロックを掛け、コミットかロールバック後に全てアンロックする。
- ウ トランザクションがデータを更新する前に専有ロックを、参照する前に共有ロックを掛け、専有ロックはコミットかロールバック後までアンロックしないが、共有ロックは必要にならなければアンロックする。
- エ トランザクションがデータを更新する前にロックを掛けず、コミット直前に他のトランザクションがそのデータを更新したかどうか確認し、更新していないときだけコミットする。

問16 多版同時実行制御 (MVCC) の特徴のうち、適切なものはどれか。

- ア アプリケーションプログラムからデータに対する明示的なロックをかけることができない。
- イ データアクセスの対象となる版をアプリケーションプログラムが指定する必要がある。
- ウ データ書き込みに対して新しい版を生成し、同時にデータ読み取りが実行されるときの排他制御による待ちを回避する。
- エ デッドロックは発生しない。

問 18：正解（ア）

データベースの同時実行制御に関する問題。多版同時実行制御（MVCC：MultiVersion Concurrency Control）について問われている。MVCCとは、ロック方式とは異なる方法で行う同時実行制御で、現在のデータベースの状態ではなく、過去のある時点のスナップショットを参照する方式である。先発のトランザクションがデータを更新し、コミットする前に、後発のトランザクションが同じデータを参照すると、更新前の値（これが過去のある時点のスナップショット）を返す。したがって選択肢アが正解になる。他の選択肢は全てロックを使う同時実行制御になる。

問 16：正解（ウ）

データベースの同時実行制御に関する問題。多版同時実行制御（MVCC：MultiVersion Concurrency Control）について問われている。MVCCとは、ロック方式とは異なる方法で行う同時実行制御で、現在のデータベースの状態ではなく、過去のある時点のスナップショットを参照する方式である。先発のトランザクションがデータを更新し、コミットする前に、後発のトランザクションが同じデータを参照すると、更新前の値（これが過去のある時点のスナップショット）を返す。

要するに、ロック方式ではデータの読み込み時のロック獲得と書き込み時のロック獲得は競合するが、MVCCでは競合しないということになる。したがって、選択肢ウが正解になる。

ア：MVCCを採用しているDBMSでも、アプリケーションプログラムから明示的にロックをすることが可能としているのが一般的。

イ：特に必要ない。DBMSが自動的に行う。

ウ：正しい。

エ：書き込みが重なると、タイミングによってはデッドロックになる可能性はある。

問14 トランザクション P が資源 X の値を 4 から 5 に更新した後にトランザクション Q が開始し、P がコミットする前に Q が資源 X を参照しようとした。Q の挙動の a, b に入れる字句の組みはどれか。ただし、隔離性水準は READ COMMITTED とする。

〔Q の挙動〕

同時実行制御が単版 2 相ロックングプロトコルの場合 Q は , 多版同時実行制御 (MVCC) の場合 Q は 。

	a	b
ア	P がコミット完了するまで待機した後、X の値 5 を得る	P のコミット完了を待機することなく、X の値 4 を得る
イ	P がコミット完了するまで待機した後、X の値 5 を得る	P のコミット完了を待機することなく、X の値 5 を得る
ウ	P のコミット完了を待機することなく、X の値 4 を得る	P のコミット完了を待機することなく、X の値 5 を得る
エ	P のコミット完了を待機することなく、X の値 5 を得る	P のコミット完了を待機することなく、X の値 4 を得る

問 14：正解（ア）

データベースの同時実行制御に関する問題。単版2相ロックングプロトコルと多版同時実行制御（MVCC）について問われている。また、解答に当たっては隔離性水準に関する知識も必要になる。

2相ロック方式（[参照▶本書第4章](#)）とは、トランザクションがデータの読み取りまたは書き込みを行う際にロックを取得する段階（第1相：拡張相）と、ロックを解除していく段階（第2相：縮退相）の2つのフェーズからなるロック方式の総称である。この方式では、第1相（拡張相）が終了した後に第2相（縮退相）が始まるというルールがあるため、同一トランザクション内で一度ロックを解除した後に、再びロックを取得することはできない。2相ロック方式では、共有ロック（Sロック）および専有ロック（Xロック）の両方を使用可能であるが、デッドロックが発生する可能性がある。なお、この問題で「単版」としているのは、2相ロック方式が一般的に单一のデータバージョンを扱うためであり、多版同時実行制御（MVCC）と対比するための表現である。

一方、多版同時実行制御（MVCC：Multi-Version Concurrency Control）とは、ロックを使用せず、データの複数のバージョンを管理することで実現される同時実行制御方式である。MVCCでは、現在のデータベースの状態ではなく、過去のある時点のスナップショットを参照することで、読み取り処理（SELECT）と書き込み処理（INSERT, UPDATE, DELETE）を非ブロッキングで並行実行できるようとする。具体的には、先行するトランザクションがデータを更新し、コミットする前に、後続のトランザクションが同じデータを参照すると、その時点の確定済みデータ（=過去のある時点のスナップショット）が返される。これにより、読み取り処理はロックを必要とせず、一貫したデータを取得できる。要するに、ロック方式ではデータの読み込み時のロック獲得と書き込み時のロック獲得は競合するが、MVCCでは競合しないということになる。

また、隔離性水準については第4章を参照してほしい。

まず、単版2相ロックングプロトコルの場合、隔離性水準がREAD COMMITTEDにおいては、次のような処理手順になる。

- ①トランザクションPが資源Xの値を4から5に変更
- ②Qが資源Xを参照しようとする
(但し、上記①のコミット前なので、コミット完了まで待たれる)
- ③トランザクションPがコミットして完了
- ④トランザクションQが参照する

一方、MVCCの場合は次のようになる。

- ①トランザクションPが資源Xの値を4から5に変更
- ②Qが資源Xを参照しようとする。コミット完了前だが、多版の場合コミットの完了を待たずにコミット前の「Xの値=4」を得る
- ③トランザクションPがコミットして完了

上記の組み合わせの解答は、選択肢Aになる。

問13 デッドロックが発生する可能性のある排他制御の方式はどれか。

- ア 2相ロックプロトコルに従ってロックする方式
- イ 時刻印を用いて、トランザクションの優先順位を決める方式
- ウ 全てのトランザクションにおいて、ロック対象のデータは一定の順序でロックを行い、全てのロックが完了するまで、アンロックを行わない方式
- エ トランザクション開始時に一括してロックする方式

問13：正解（ア）

過去問題（平成20年・DB午前 問43など）

参照 第4章 重要キーワード「5 トランザクション管理機能」

デッドロックに関する問題。選択肢を順番に見ていく。

ア：2相ロックプロトコルは、トランザクションの実行前にロックをかけ、終了時にロックを解除する方式で、デッドロックは発生する。よって正解である。

イ：時刻印アルゴリズムでは、二つのトランザクションで競合が発生した場合、先に開始したトランザクションから順番に実行するようにスケジューリングする。後から実行したトランザクションはアボートされるので、デッドロックは発生しない。

ウ：ロック対象のデータの獲得順序を一定にするのは、デッドロック回避方法の常套手段。デッドロックは発生しない。

エ：各トランザクションが、開始時に（すべての共有資源に）一括してロックしてしまえば、すべての処理時間は並列処理できなくなるのでデッドロックは発生しない。

問38 三つのトランザクション T1, T2, T3 が、①～⑪の順序でデータ a, b に対する処理を行った場合、デッドロックとなるのはどの時点か。ここで、DBMS は READ の直前に共有ロック、UPDATE の直前に占有ロックをかけ、ROLLBACK 又は COMMIT ですべてのロックを解除する。

T1
① READ a
③ READ b
⑦ UPDATE b
⑨ UPDATE a
⑩ COMMIT

T2
② UPDATE b
④ ROLLBACK

T3
⑤ READ b
⑥ UPDATE a
⑧ UPDATE b
⑪ COMMIT

ア ⑥

イ ⑦

ウ ⑧

エ ⑨

問 38：正解イ

デッドロックに関する問題。共有ロックと占有ロックの関係を表1に、トランザクションの処理①～⑪を表2にまとめた。ここからわかるように、⑦でデッドロックが成立する。よって正解はイになる。

参照 ➔ 「4.1.3 排他制御（同時実行制御）」

表1 共有ロックと占有ロックの関係

かけようとするロック	すでにかかっているロック	
	共有ロック	占有ロック
共有ロック	○	×
占有ロック	×	×

表2 トランザクションの処理

		共有資源に対するロック指示		トランザクションの状態
		a	b	
①	T1 が READ a	T1により共有ロック(成功)		
②	T2 が UPDATE b		T2により占有ロック(成功)	
③	T1 が READ b		T1により共有ロック(失敗)	ここでT1は、“b”が解放されるまで待つ
④	T2 が ROLLBACK		T2の占有ロックが解除される。T1により共有ロック(成功)	上記T1の待ちは解消
⑤	T3 が READ b		T3により共有ロック(成功)	
⑥	T3 が UPDATE a	T3により占有ロック(失敗)		ここでT3は、“a”が解放されるまで待つ
⑦	T1 が UPDATE b		T1により占有ロック(失敗)	ここでT1は、“b”が解放されるまで待つ
⑧	T3 が UPDATE b			T3は待ち状態・・・
⑨	T1 が UPDATE a			T1は待ち状態・・・
⑩	T1 が COMMIT			T1は待ち状態・・・
⑪	T3 が COMMIT			T3は待ち状態・・・

問40 分散データベースにおけるトランザクションの同時実行制御のうち、デッドロックが発生しない方式だけから成る組合せはどれか。

- | | |
|---------------|---------------------|
| ア 時刻印方式、楽観的方式 | イ 時刻印方式、楽観的方式、ロック方式 |
| ウ 時刻印方式、ロック方式 | エ 楽観的方式、ロック方式 |

問13 複数のバッチ処理を並行して動かすとき、デッドロックの発生をできるだけ回避したい。バッチ処理の設計ガイドラインのうち、適切なものはどれか。

- | |
|---|
| ア 参照するレコードにも、専有ロックをかけるように設計する。 |
| イ 大量データに同じ処理を行うバッチ処理は、まとめて一つのトランザクションとして処理するように設計する。 |
| ウ トランザクション開始直後に、必要なレコード全てに専有ロックをかける。ロックに失敗したレコードには、しばらく待って再度ロックをかけるように設計する。 |
| エ 複数レコードを更新するときにロックをかける順番を決めておき、全てのバッチ処理がこれに従って処理するように設計する。 |

問 40：正解ア

デッドロックが発生しない方式は時刻印方式と楽観的方式である。ロック方式は、使用中の資源にロックをかける方式なので、デッドロックが発生する可能性はある。したがって正解はアになる。

問 13：正解（エ）

複数のバッチ処理を並行して動かす場合のデッドロック回避方法としては、トランザクションの実行順序を、全トランザクションで合わせる必要がある。そのことに言及している選択肢は（エ）になる。他の選択肢は、いずれも、複数のバッチ処理を並行して動かすことはできないため、誤り。

問35 並行プロセス環境において、デッドロックが発生する原因とならないものはどれか。

- ア 一度割り当てられた資源は、プロセスがその使用を終了するまで強制的には取り上げられない。
- イ すべてのプロセスには、決められた順序に従って資源が割り当てられる。
- ウ プロセスがほかのプロセスを待ち合うという関係が、環状になっている。
- エ プロセスは、一つの資源を占有しながらほかの資源へ要求を出す。

問16 DBMSにおいて、デッドロックを検出するために使われるデータ構造はどれか。

- | | |
|-------------------|-----------|
| ア 資源割当表 | イ 時刻印順管理表 |
| ウ トランザクションの優先順管理表 | エ 待ちグラフ |

問 35：正解イ

- ア：プロセスがその使用を終了するまで、割り当てられた資源を開放しないから、資源獲得のタイミングによってはデッドロックが発生する。
- ウ：環状になっている場合、デッドロックが発生する可能性がある。例えば、3つのプロセス A、B、C が、A が B を、B が C を、C が A をそれぞれ待ち合うようなケースである。
- エ：「ほかの資源」へ要求を出したときに、他のプロセス(A)が利用しているため、待ちがかかるれば、「一つの(占有している)資源」は占有したままである。このときプロセス A が、「一つの(占有している)資源」を獲得に行くとデッドロックになる。

問 16：正解（エ）

トランザクションの同時実行制御に関する問題。デッドロックの検出方法が問われている。

デッドロックを検出する方法には、時間監視による方法や待ちグラフを用いる方法などがある。待ちグラフは、トランザクションによるロック獲得をグラフで管理し、グラフに閉路が生じた場合にデッドロック（すくみ状態）と判定する方法である。他の選択肢に列挙された方法では、デッドロックを検出することはできない。

ア：資源を獲得する順番を管理する必要があるので、資源の割当てを管理するだけでは不十分である。

イ：時刻印順アルゴリズムはロックをかけずに同時実行制御を実現する方式である。ロックをかけないのでデッドロックは発生しない。

ウ：デッドロックを検出するためには、互いが獲得済みの資源を待っている状態を検出する必要があるので、優先順位は関係がない。

問13 DBMSにおいて、トランザクション間でデッドロックが発生していることを検出するために使用するものはどれか。

ア 2相ロック

イ 時刻印アルゴリズム

ウ チェックポイント

エ 待ちグラフ

問13：正解（エ）

デッドロックを検出する方法が問われている。デッドロックを検出する方法としては、時間監視による方法と、待ちグラフを用いる方法がある。待ちグラフは、トランザクションによるロック獲得をグラフで管理し、グラフに閉路が生じた場合にデッドロック（すくみ状態）と判定する方法である。他の選択肢に列挙された方法では、デッドロックを検出することはできない。したがって正解はエになる。

ア：2相ロック（2相ロッキングプロトコル）は、排他制御の方式の一つ。各トランザクションは、実行前にすべての必要資源にロックをかけ（第1相）、操作終了後にロックを解除する（第2相）という方法。デッドロックが発生する。

イ：時刻印アルゴリズムとは、ロックを掛けずに同時実行制御を実現する方式である。ロックをかけないのでデッドロックは発生しない。

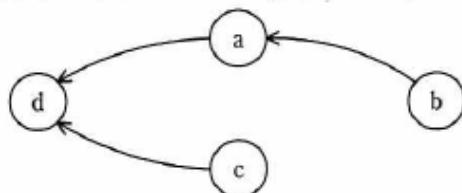
ウ：チェックポイントとは、障害回復に必要な機能の一つである。トランザクション処理におけるデータの更新は、ディスク内のデータを操作せずに、メモリ上にバッファの内容を更新する。このメモリ上のバッファの内容と、2次記憶上のデータベースの内容を一致させるタイミングのことをチェックポイントという。チェックポイントはトランザクションの途中でも発生する。

問10 $t_1 \sim t_{10}$ の時刻でスケジュールされたトランザクション $T_1 \sim T_4$ がある。時刻 t_{10} で T_1 が commit を発行する直前の、トランザクションの待ちグラフを作成した。a に当てはまるトランザクションはどれか。ここで、select (X) は共有ロックを掛けて資源 X を参照することを表し、update (X) は専有ロックを掛けた資源 X を更新することを表す。これらのロックは、commit されるまでアンロックされないものとする。また、トランザクションの待ちグラフの矢印は、 $T_i \rightarrow T_j$ としたとき、 T_j がロックしている資源のアンロックを、 T_i が待つことを表す。

[トランザクションのスケジュール]

時刻	トランザクション			
	T_1	T_2	T_3	T_4
t_1	select (A)	—	—	—
t_2	—	select (B)	—	—
t_3	—	—	select (A)	—
t_4	—	—	—	select (B)
t_5	—	—	—	update (B)
t_6	select (C)	—	—	—
t_7	—	select (C)	—	—
t_8	—	update (C)	—	—
t_9	—	—	update (A)	—
t_{10}	commit	—	—	—

[トランザクションの待ちグラフ]

ア T_1 イ T_2 ウ T_3 エ T_4

問10：正解（イ）

トランザクションの排他制御（同時実効制御）に関する問題。ロックの種類（共有ロックと占有ロック）の違いを明確にしながら、スケジュール通りに t_{10} まで確認していくべき。問題は、「 t_0 で T_1 が commit を発行する直前」の状態だという点に注意しよう。

ロックの種類は表1で確認。その後、表2のように、資源A, B, Cに関して、順番にチェックしていくべき。

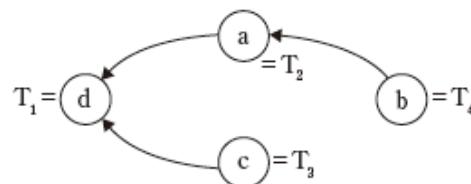
表1：共有ロックと占有ロックの競合

		既にかかっているロック	
		共有ロック	占有ロック
かけようとする ロック	共有ロック	○	×
	占有ロック	×	×

表2：トランザクションのスケジュールと資源の関係

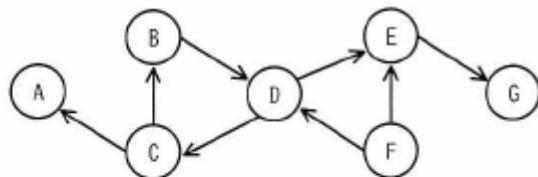
時刻	T_1	T_2	T_3	T_4	A	B	C
t_1	select(A)				T_1 が共有ロック		
t_2		select(B)				T_2 が共有ロック	
t_3			select(A)		T_3 が共有ロック		
t_4				select(B)		T_4 が共有ロック	
t_5				update(B)		T_4 が占有ロック待ち (T_2 のロック解除待ち)	
t_6	select(C)						T_1 が共有ロック
t_7		select(C)					T_2 が共有ロック
t_8		update(C)					T_2 が占有ロック待ち (T_1 のロック解除待ち)
t_9			update(A)		T_3 が占有ロック待ち (T_1 のロック解除待ち)		
t_{10}	commit				・ T_1 が共有ロック解除 ・ T_3 が占有ロック ・ T_3 がAを更新	・ T_1 が共有ロック解除 ・ T_2 が占有ロック ・ T_2 がCを更新	

以上を踏まえて、トランザクションの待ちグラフを確認する。ポイントは、dに対して、二つ資源待ちが発生している点だ。表2を見れば、資源Aと資源Cが、いずれも T_1 のロック解除待ち (= commit待ち) になる。したがって「d = T_1 」「aもしくはcが、 T_2 と T_3 」になる。残りのbは T_4 なので、 T_4 は T_2 のロック解除待ちになっていることから、次のように決定できる。空欄aは、 T_2 になるので正解はイになる。



問13 トランザクション A～G の待ちグラフにおいて、永久待ちの状態になっているトランザクション全てを列挙したものはどれか。ここで、待ちグラフの $X \rightarrow Y$ は、トランザクション X はトランザクション Y がロックしている資源のアンロックを待っていることを表す。

[トランザクション A～G の待ちグラフ]



ア A, B, C, D

ウ B, C, D, F

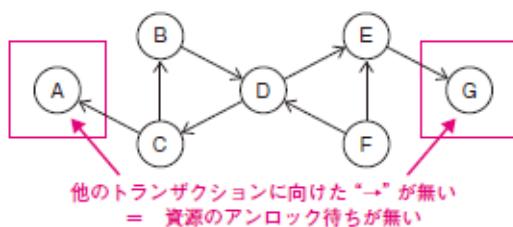
イ B, C, D

エ C, D, E, F, G

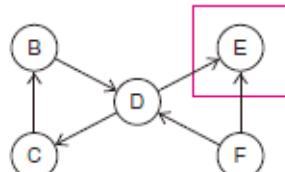
問 13：正解（ウ）

待ちグラフに関する問題。待ちグラフとは、デッドロックの検出に用いられる有向グラフである。問題文にも書いている通り「待ちグラフの $X \rightarrow Y$ は、トランザクション X はトランザクション Y がロックしている資源のアンロックを待っていること」を指している。そこで、“ \rightarrow ”の先のトランザクションのうち、他のトランザクションに “ \rightarrow ” が伸びていないところから「処理が完了していく」と考えていいべきだろう。

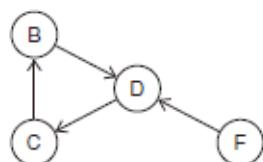
- ①トランザクション A と G は、他のトランザクションがロックしている資源のアンロックを待っていないので、処理は進行していずれ終了する（獲得していた資源をアンロックする）。



- ②トランザクション G が資源をアンロックすると、トランザクション E の処理が進行していずれ終了する（獲得していた資源をアンロックする）。



- ③残ったトランザクション（B, C, D, F）はすべて、他のトランザクションがロックしている資源のアンロックを待っている状態なので、これ以上は進行しない。永久待ちの状態になっているトランザクションは（B, C, D, F）になる。したがって、選択肢ウが正解になる。



問14 同時実行制御の手法の一種である楽観的制御法に関する記述として、適切なものはどうか。

- ア データに対して順序付けを行い、その順序に従ってロックを掛ける。
- イ データに対してのロックは行わずに、更新対象のデータが他のトランザクションと競合がなかったことを確認してからコミットを行う。
- ウ 分散データベースシステムにおいて、コミット可否の問合せをした上で、コミット指示を行う。
- エ ロックが増加していく成長フェーズである第 1 相と、ロックが減少していく縮退フェーズである第 2 相の 2 相制御を行う。

問16 トランザクションの隔離性水準を高めたとき、不整合なデータを読み込むトランザクション数と、単位時間に処理できるトランザクション数の傾向として、適切な組合せはどれか。

	不整合なデータを読み込むトランザクション数	単位時間に処理できるトランザクション数
ア	増える	増える
イ	増える	減る
ウ	減る	増える
エ	減る	減る

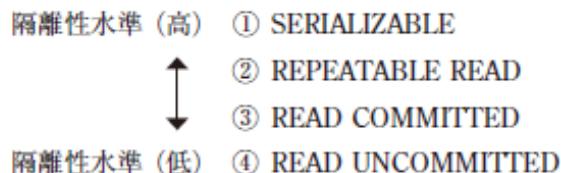
問14：正解（イ）

同時実行制御に関する問題。楽観的制御法について問われている。楽観的制御法は、データに対してロックは行わず、更新対象のデータが他のトランザクションと競合しなかったことを確認してからコミットを行う。したがって正解はイになる。ロックをかけないので、当然、デッドロックは発生しない。全体的に競合が少ない場合に効果的な方式である。

- ア：ロック方式の説明
- ウ：2相コミット方式の説明
- エ：2相ロック方式の説明

問16：正解（工）

隔離性水準に関する問題。問題文の「隔離性水準を高める」とは、SERIALIZABLE の方向に向かうということになり、隔離性水準が高いものから順番に並べると次のようになる。



要するに、隔離性水準を高めていくと排他制御を確実に行う SERIALIZABLE に至るので、「不整合なデータを読み込むトランザクション数」は“減り”、「単位時間に処理できるトランザクション数」も“減る”。したがって、選択肢エが正解になる。

問 7 次の(1), (2)に該当するトランザクションの隔離性水準はどれか。

- (1) 対象の表のダーティリードは回避できる。
(2) 一つのトランザクション中で、対象の表のある行を 2 回以上参照する場合、1 回目の読み込みの列値と 2 回目以降の読み込みの列値が同じであることが保証されない。

ア READ COMMITTED
ウ REPEATABLE READ

イ READ UNCOMMITTED
エ SERIALIZABLE

問 18 DBMS のトランザクション管理に関する記述のうち、適切なものはどれか。

- ア 2 相ロック方式は、分散型データベースのための制御方式であり、集中型データベースでは使用されない。
イ 資源をロックする時間の長さのことをロックの粒度といい、この粒度が細かいほど、トランザクションのスループットは向上する。
ウ ダーティリードを許すなど、隔離性水準を下げると、トランザクションのスループットは低下する。
エ 同時実行制御の目的は、データベースの一貫性を保ちながら複数のトランザクションを並行に処理することである。

問 7：正解（ア）

トランザクション処理における隔離性水準（ISOLATION LEVEL）に関する問題。（1）と（2）の条件にしたがって絞り込んでいく。対象は、選択肢の四つの水準になる。

（1）で“ダーティリード”が回避できると書いているので、ダーティリードが発生してしまう READ UNCOMMITTED が対象から外れる。これで残りの3つに対象が絞り込むことができる。

次に（2）。ここに書いている「2回以上参照する場合、1回の読み込みの列値と2回目以降の読み込みの列値が同じであることが保障されない。」というは、繰り返し読むことができない（読み込むとまずい）ので、ノンリピータブルリードの説明だとわかるだろう。それが発生するのは、READ UNCOMMITTED と READ COMMITTED になる。

上記の（1）（2）の結果から、正解は“READ COMMITTED”的選択肢（ア）になる。

問 18：正解工

トランザクション管理に関する問題（[参照 第4章-5 トランザクション管理機能](#)）。選択肢を順番に見ていく。

ア：2相ロックではなく、2相コミットメントに関する記述である。2相ロックは集中型のデータベースでも使用される。誤り。

イ：ロックの粒度とは、ロックする時間の長さではなく、ロック対象の大きさ（テーブル、行など）である。誤り。

ウ：ダーティリードを許すなど、隔離性水準を下げると、トランザクション処理のスループットは高くなる。誤り。

エ：正しい記述である。正解。

問15 SQL でトランザクションの隔離性水準を READ COMMITTED に指定したときに発生する状態はどれか。

- ア ダーティリードとアンリピータブルリードとファントムリードが発生する。
- イ ダーティリードとアンリピータブルリードは発生しないが、ファントムリードが発生する。
- ウ ダーティリードは発生しないが、アンリピータブルリードとファントムリードが発生する。
- エ ダーティリードもアンリピータブルリードもファントムリードも発生しない。

問35 ページ単位で排他制御を行う DBMS において、T 表に対する処理①と②をトランザクションモード READ COMMITTED で並行処理した場合の事象に関して、誤っているものはどれか。

ここで、T 表には三つの列 (A, B, C) があり、列 A が主キーである。

また、①②ともに SQL 文の直後に COMMIT 文が付属しているものとする。

- ①の SQL 文 `SELECT SUM(B) , SUM(C) INTO :HSB , :HSC FROM T`
- ②の SQL 文 `UPDATE T SET B=B+:HB , C=C+:HC WHERE A=:HA`

- ア ①と②の間でデッドロックが発生する場合がある。
- イ ①の実行中に②を実行すると、②が先に終了する場合がある。
- ウ ②の COMMIT 実行前の結果が①に反映されることはない。
- エ ②を連続して実行しているときに①を実行すると、①より前に終了した②の結果が①に反映される場合がある。

問 15：正解（ウ）

(NEW)

参照 ➤ 第4章 重要キーワード「5 トランザクション管理機能」

トランザクションの隔離性水準に関する問題。READ COMMITTED で発生する状態について問われている。READ COMMITTED では、ダーティリードは発生しないが、アンリピータブルリード、ファンтомリードが発生する。したがって、選択肢（ウ）が正解になる。

ア：全てが発生するのは、READ UNCOMMITTED

イ：ファンтомリードだけが発生するのは、REPEATABLE READ

ウ：これが、READ COMMITTED

エ：いずれも発生しないのは、SERIALIZABLE

※いわゆる「反復不能な読み取り」や「非再現リード」のことを、本書では「ノンリピータブルリード」と説明しているが、情報処理技術者試験では「アンリピータブルリード」と言っている。他に「ファジーリード」ということもあり、これらはすべて同じ現象のことを言っている。

問 35：正解ア

“READ COMMITTED”とは、コミットされたデータのみを読み込むことである。設問を順番に見ていく。

参照 ➤ 「4.1.3 排他制御（同時実行制御）」

ア：複数の資源を順番にロックをしているわけではないので、デッドロックは発生しない。

誤っている内容のため、これが正解になる

イ：ページ単位の排他制御で、“READ COMMITTED”的ため、①実行中に②の処理が先に終了することはある。正しい

ウ：COMMITされたものしか処理しないため、正しい

エ：②→①→②の順序で処理されると、②の結果は①に反映されているので正しい

問14 トランザクション T_1 がある行 X を読んだ後、別のトランザクション T_2 が行 X の値を更新してコミットし、再び T_1 が行 X を読むと、以前読んだ値と異なる値が得られた。この現象を回避する SQL の隔離性水準のうち、最も水準の低いものはどれか。

- | | |
|-------------------|--------------------|
| ア READ COMMITTED | イ READ UNCOMMITTED |
| ウ REPEATABLE READ | エ SERIALIZABLE |

問14 厳格な 2 相ロックングプロトコルと表ロックを適用して、同時実行している他のトランザクションの処理結果に影響を与えないようにする。実現されるトランザクションの隔離性水準はどれか。

- | | |
|--------------------|------------------|
| ア READ UNCOMMITTED | イ READ COMMITTED |
| ウ REPEATABLE READ | エ SERIALIZABLE |

問 14：正解（ウ）

トランザクションの隔離性水準の問題。問題文のように「トランザクション T_1 がある行 X を読んだ後、別のトランザクション T_2 が行 X の値を更新してコミットし、再び T_1 が行 X を読むと、以前読んだ値と異なる値が得られた。」という現象をノンリピータブルリードという。このノンリピータブルリードを回避する隔離性水準には、REPEATABLE READ と SERIALIZABLE がある。このうち最も水準が低いのは REPEATABLE READ なので、正解はウになる。

問 14：正解（エ）

隔離性水準に関する問題。厳格な2相ロッキングプロトコルと表ロックを適用して、同時に実行している他のトランザクションの処理結果に影響を与えないようにする隔離性水準は SERIALIZABLE（選択肢エ）になる。あるトランザクションが処理中の場合には、他のトランザクションは処理待ち（排他待ち）になるので、他のトランザクションの処理結果に影響は与えない。ダーティリードもノンリピータブルリード、ファンтомリードのいずれも発生しない。しかし、全体のスループットは他の隔離性水準に比べて悪い。

問12 二つのトランザクションが、同じデータに対して、更新、参照を行うときに発生し得るダーティリードの事象を記述したものはどれか。

- ア トランザクション A がある検索条件を満たす、ある表の行の集合を参照した。次に、トランザクション B がトランザクション A と同じ検索条件を満たす新しい行を挿入しコミットした。その後、トランザクション A が同じ検索条件で再度参照すると、以前には存在しなかった行が出現した。
- イ トランザクション A がある表の行の列を参照した。次に、トランザクション B がその列の値を更新しコミットした。その後、トランザクション A がその列を再度参照すると、以前の値と異なった。
- ウ 二つのトランザクションがそれぞれ 2 相ロックをかけ、デッドロックを起こした。
- エ まだコミットしていないトランザクション A の更新後データをトランザクション B が参照した。その後、更新後データはロールバックされた。

問14 データベースのトランザクション T2 の振る舞いのうち、ダーティリード (dirty read) に関する記述はどれか。

- ア トランザクション T1 が行を検索し、トランザクション T2 がその行を更新する。その後 T1 は先に読んだ行を更新する。その後に T2 が同じ行を読んでも、先の T2 による更新が反映されない値を得ることになる。
- イ トランザクション T1 が行を更新し、トランザクション T2 がその行を検索する。その後 T1 がロールバックされると、T2 はその行に存在しない値を読んだことになる。
- ウ トランザクション T2 がある条件を満たす行を検索しているときに、トランザクション T1 が T2 の検索条件を満たす行を挿入する。その後 T2 が同じ条件でもう一度検索を実行すると、前回は存在しなかった行を読むことになる。
- エ トランザクション T2 が行を検索し、トランザクション T1 がその行を更新しコミットする。その後 T2 が同じ行を検索した場合、同じ行を読んだにもかかわらず、異なる値を得ることになる。

問 12：正解（工）

トランザクション管理に関する問題。独立阻害要因のひとつダーティリードについて問われている。ダーティリードとは、一言で言うと「他のトランザクションのコミット前のレコードをリードし、その後、（他のトランザクションが）当該レコードを取り消したために、不整合が発生してしまうこと」である。ISOLATION LEVEL が READ UNCOMMITTED の時に発生する。それを念頭に、選択肢を順番にチェックしていく。

- ア：ファンタムリードの説明。
- イ：ノンリピータブルリードの説明。
- ウ：デッドロックの説明。
- エ：これがダーティリードの事象になる。正しい。

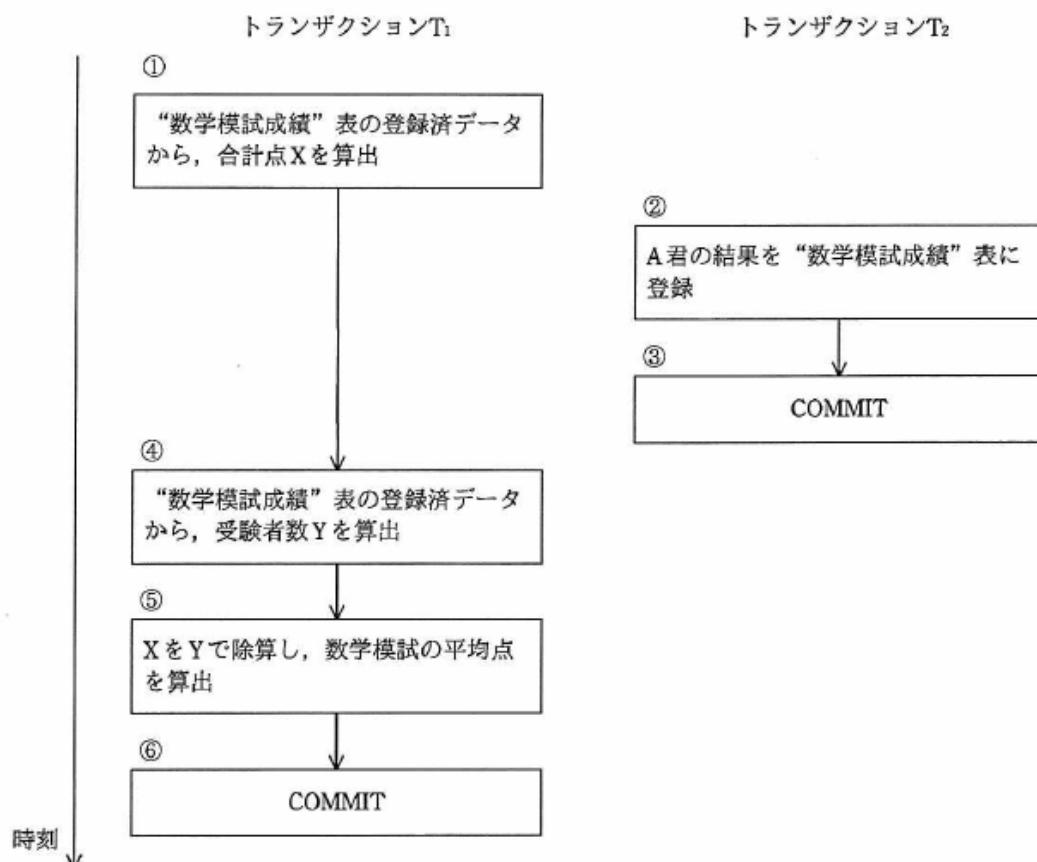
問 14：正解（イ）

過去問題（平成23年度・DB午前Ⅱ 問15）
参照▶ 「4.5 トランザクション管理機能」

トランザクション管理に関する問題。独立阻害要因の一つダーティリードについて問われている。ダーティリードとは、一言で言うと「他のトランザクションのコミット前のレコードを取り込んでしまうようなリード」である。それを念頭に、選択肢を順番にチェックしていく。

- ア：ロストアップデートの説明。
- イ：正しい。
- ウ：ファンタムリードの説明。
- エ：ノンリピータブルリードの説明。

問17 図は、ある探索条件を使って数学模試の平均点を算出している間、当該探索条件に合致する A 君の結果を“数学模試成績”表に登録したときの様子を示している。平均点を求めるトランザクション T_1 と、登録作業のトランザクション T_2 が①～⑥の順序で処理された結果、合計点算出時の受験者数と平均点算出時の受験者数が異なり、正しい平均点を得ることができなかつた。このとき発生した事象はどれか。ここで、トランザクションの隔離性水準は READ UNCOMMITTED であったとする。



ア アンリピータブルリード

イ シーケンシャルリード

ウ ダーティリード

エ ファントムリード

問17：正解（工）

独立性阻害要因に関する問題（参照 第4章）。READ UNCOMMITTEDなので、選択肢のダーティリード、アンリピータブルリード、ファントムリードのいずれも発生する。

- ① “数学模試成績”表のA君の結果を除いた合計点Xを算出している。
- ④ “数学模試成績”表のA君を含む受験者数Yを算出している。

これが「正しい平均点を得ることができなかった。」理由である。つまり、①と④の間に追加されたデータを読み込んでしまっているからだ。これをファントムリードという。したがって正解はエになる。

ア：アンリピータブルリードは、同じデータを2回読み込んだ時に値が違う事象。

イ：シーケンシャルリードは、独立性阻害要因ではなく順番にリードすること。

ウ：ダーティリードは、他のトランザクションで更新された“コミット前”的データを読み込んでしまうこと。ロールバックした時に不整合が発生する。

4.3 障害回復機能

令和2年度・D B・午前Ⅱ

問4 DBMS が取得するログに関する記述として、適切なものはどれか。

- ア トランザクションの取消しに備えて、データベースの更新されたページに対する更新後情報を取得する。
- イ 媒体障害からの復旧に備えて、データベースの更新されたページに対する更新前情報を取得する。
- ウ ロールバック後のトランザクション再実行に備えて、データベースの更新されたページに対する更新後情報を取得する。
- エ ロールフォワードに備えて、データベースの更新されたページに対する更新後情報を取得する。

平成28年度・D B・午前Ⅱ

問16 WAL (Write Ahead Log) プロトコルの目的に関する説明のうち、適切なものはどれか。

- ア 実行中のトランザクションを一時停止させることなく、チェックポイント処理を可能にする。
- イ デッドロック状態になっているトランザクションの検出を可能にする。
- ウ 何らかの理由で DBMS が停止しても、コミット済みであるがデータベースに書き込まれていない更新データの回復を可能にする。
- エ ログを格納する記録媒体に障害が発生しても、データベースのデータ更新を可能にする。

問4：正解（エ）

障害回復機能に関する問題。選択肢を順番に見ていく。

- ア：トランザクションを取り消す場合、それが実行される前の状態まで戻さなければならぬいため、利用するのは“更新前情報”になる。誤り。
- イ：媒体障害については、媒体交換後に、障害発生直前までのログ（更新されたページの更新後情報）を利用してロールフォワード（前進復帰）させる。したがって、“更新前情報”ではないため誤り。
- ウ：ロールバックし再実行するのだから、更新後情報は不要である。誤り。
- エ：正しい。これが正解。

問16：正解（ウ）

WAL (Write Ahead Log) は、“ログ先書き”というように、次のような順番でコミットを実行する。この順番を念頭に選択肢を順番に見ていく。

- ①更新前イメージのログファイルへの書出し
- ②更新後イメージのログファイルへの書出し
- ③データベースの更新

ア：チェックポイント処理の説明。誤り。

イ：デッドロックの検出方法ではない。誤り。

ウ：正解。通常、コミット済みの場合は、データベースの更新も終わっているはずだが、コミット中の障害等何らかの理由でデータベースに更新されていないデータが存在しても、更新後イメージのログファイルを先に書出しているので、それを使用して更新データの回復が可能になる。

エ：ログを格納する記憶媒体に障害が発生してしまったらデータベースの更新はできなくなる。誤り。

問17 更新前レコードと更新後レコードをログとして利用する DBMSにおいて、ログを先に書き出す WAL (Write Ahead Log) プロトコルに従うとして、処理①～⑥を正しい順番に並べたものはどれか。

- ① begin transaction レコードを書き出す。
- ② データベースを更新する。
- ③ ログに更新前レコードを書き出す。
- ④ ログに更新後レコードを書き出す。
- ⑤ commit レコードを書き出す。
- ⑥ end transaction レコードを書き出す。

ア ①→②→③→④→⑤→⑥

イ ①→③→②→④→⑥→⑤

ウ ①→③→②→⑤→④→⑥

エ ①→③→④→②→⑤→⑥

問 17：正解（エ）

過去問題（平成21年・DB午前Ⅱ 問11など）

参照 第4章 重要キーワード「5 トランザクション管理機能」

ログ書き出し順序に関する問題。ログ先書き出し方式では、次の順序に従って処理される。したがって正解は（エ）になる。

- (1) トランザクション開始時 ((①ログファイルへ begin transaction レコードを書き出す))
- (2) トランザクション更新時 ((③ログファイルへ更新前レコードを書き出す))
- (3) トランザクション更新時 ((④ログファイルへ更新後レコードを書き出す))
- (4) トランザクション更新時 ((②データベースへの書き出し))
- (5) トランザクション終了時 ((⑤commit を書き出す))
- (6) トランザクション終了時 ((⑥end transaction レコードを書き出す))

ちなみに、「②データベースへの書き出し」は、チェックポイントが発生した時やバッファがフラッシュした時にまとめて行われる。

問2 次のチェックポイントの仕様に従ってトランザクション処理を行う DBMS において、チェックポイントの発生頻度は 1 時間当たり何回か。ここで、トランザクションは毎秒 20 件発生し、1 トランザクションごとに消費されるデータベースバッファ領域のデータ量及びログファイルに書き出すログ長はどちらも 10 k バイトとする。データベースバッファ領域の容量は 480 M バイトとし、一つのログファイルのサイズは 240 M バイトとする。 $1\text{M バイト} = 10^3\text{k バイト}$ とする。開始時点では、データベースバッファ領域、ログファイルとともに初期化状態であるとする。DBMS は、ログファイルを二つもち、一方を使い切ったら他方に切り替え、使い切った一方をアーカイブして初期化する。ログファイルへの書き込み処理は WAL プロトコルに従う。

[チェックポイントの仕様]

1. チェックポイントが発生する条件

- A. データベースバッファ領域に空きがなくなったとき、又は
- B. ログファイルが切り替わるとき

2. チェックポイント終了時のデータベースバッファの状態

データベースバッファ領域は、データベースファイルへの反映後、初期化される。

ア 1.5

イ 2

ウ 3

エ 6

問2：正解（ウ）

チェックポイントに関する問題。チェックポイントに加えて、WALプロトコルに関しても問われている。

1時間に発生するトランザクションは720Mバイトになる（1トランザクションが10Kバイトで毎秒20件発生することから、1秒間に200Kバイト、1分間に12Mバイト発生する）。

まず、ひとつのログファイルの大きさが240Mなので、240Mバイトのトランザクションが発生すればもうひとつのログファイルに切り替わる。その時にチェックポイントが発生する（チェックポイントの発生状況のB）。

$$720 \text{ (M)} \div 240 \text{ (M)} = 3 \cdots \text{ チェックポイントは3回発生する}$$

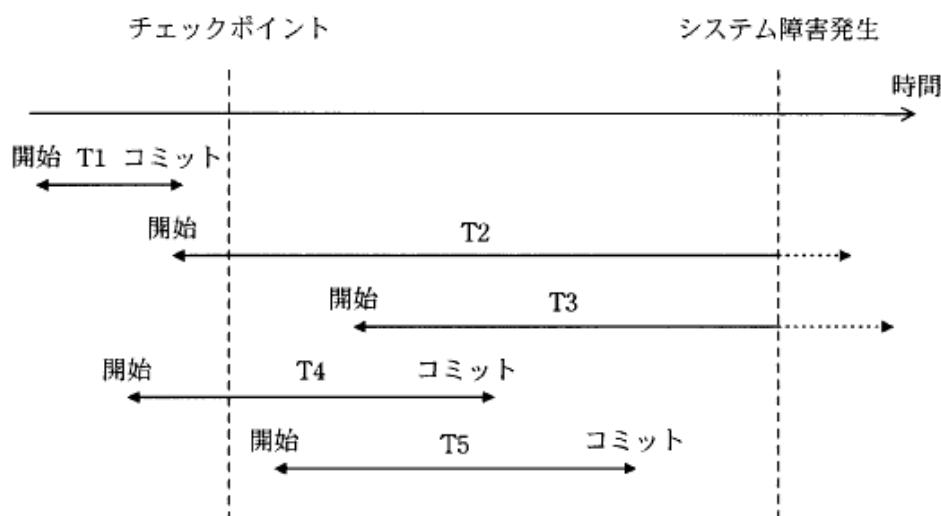
ちょうど1時間経過時に3回目になるので、ログの書き込みがトランザクションの発生との関係性がどうなるのかによって2回になるのか3回になるのか微妙かもしれないが、問題文にはWALプロトコルを使うと書いてるので、トランザクション発生直後にログファイルに書き込む点と、データベースバッファ領域やログファイルに書き出すための時間について言及されていないので無視できるものだと判断すれば、チェックポイントは3回になる。

なお、[チェックポイントの仕様]の「2. チェックポイント終了時のデータベースバッファの状態」で、「データベースバッファ領域は、データベースファイルへの反映後、初期化される。」という記述から、チェックポイントが発生するとデータベースファイルへの反映が行われるので、ログファイルが切り替わる20分（240Mバイト）ごとに初期化されるため、データベースバッファに空き領域が発生することはない。以上より正解はウになる。

問13 システム障害発生時には、データベースの整合性を保ち、かつ、最新のデータベース状態に復旧する必要がある。このために、DBMS がトランザクションのコミット処理完了とみなすタイミングとして、適切なものはどれか。

- ア アプリケーションの更新命令完了時点
- イ チェックポイント処理完了時点
- ウ ログバッファへのコミット情報書込み完了時点
- エ ログファイルへのコミット情報書込み完了時点

問40 チェックポイントを取得する DBMS において、図のような時間経過でシステム障害が発生した。前進復帰（ロールフォワード）によって障害回復できるすべてのトランザクションはどれか。



- ア T1
- イ T2 と T3
- ウ T4 と T5
- エ T5

問 13：正解（エ）

過去問題（平成24年度・DB午前Ⅱ 問14など）

参照 ➤ 「4.5 トランザクション管理機能」

トランザクションの同時実行制御に関する問題。DBMSでは、障害対策のため、コミットとログファイルへの書き出しは同期しており、ログファイルへの書きしが完了した時点でコミット処理完了とみなしている。したがって、正解は（エ）になる。

ちなみに、ログファイルへの書き出しはシーケンシャルライトであるので、ディスクアクセスに伴うシークオーバヘッドがなく、性能が劣化しない仕組みとなっている（ただし、データファイルとログファイルは別々のディスクに配置する必要がある）。

※ログファイルへの書き出しは“コミット情報”を書き出した時点だということ。それとデータファイルとログファイルは同一のディスクに配置することが不可能だという意味ではありません。媒体障害への備えや速度面を考えて“望ましい”ということ。

問 40：正解ウ

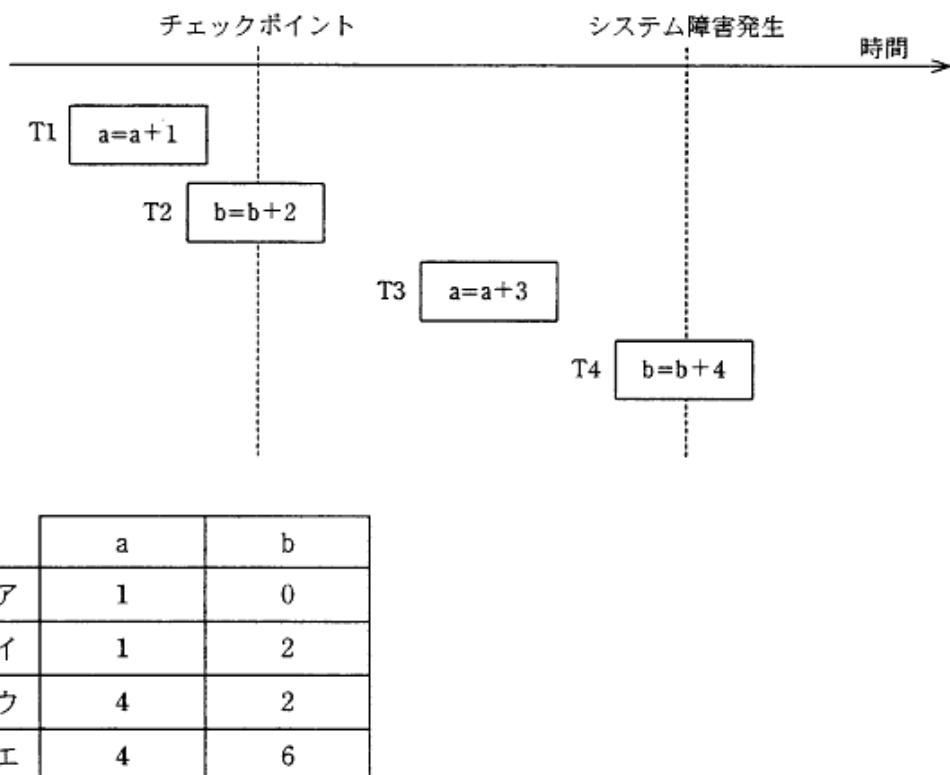
DBMSの障害回復処理に関する知識を問う問題。チェックポイント時に実行中か実行前で、システム障害発生時までに完了しているトランザクションは、前進処理によって障害回復することができる。したがって、正解はウである。 参照 ➤ 「4.1.4 障害回復機能」

ア：T1は、チェックポイントまでに実行が完了している。

イ：T2は、障害発生時に処理が完了していないため、ロールバックされる。T3は、チェックポイント後に開始されているため、トランザクションの再実行が必要である。

エ：T4も前進処理によって障害回復できるため誤り。

問39 チェックポイントを取得する DBMSにおいて、図のような時間経過でシステム障害が発生し、前進復帰によって障害回復を行った。前進復帰後の a, b の値は幾つか。ここで、 Tn [] は長方形の左右両端がトランザクションの開始と終了を表し、長方形内の記述は処理内容を表す。T1 開始前の a, b の初期値は 0 とする。



問16 DBMS がチェックポイント時点で、チェックポイントトレコードをログに書き出す他に行っている処理はどれか。

- ア 完了していないトランザクションをロールバックする。
- イ 更新したデータの更新前後情報ログをログファイルに書き出す。
- ウ データベースの内容をバックアップファイルに書き出す。
- エ バッファに残っている更新後のデータをデータベースに書き出す。

問 39：正解ウ

障害回復の頻出問題である。問題文では、チェックポイントとシステム障害発生までの間のトランザクションを前進復帰させたとなっている。この場合、それぞれのトランザクションは次のようなになる。

- T1：チェックポイントより前（○）
- T2：チェックポイント中には終了していない。前進復帰（○）
- T3：前進復帰（○）
- T4：障害発生時に完了していないので復帰は不可（×）

よって、前進復帰後の a の値は 4, b の値は 2 になる。正解はウ。

参照 「4.1.4 障害回復機能」

問 16：正解（エ）

NEW (但し、類似問題多数)

参照 第4章 重要キーワード「5 トランザクション管理機能」

チェックポイントでは、バッファに残っている更新後のデータをデータベースに書き出す。したがって正解は（エ）になる。それを前提に、他の選択肢をみていこう。

- ア：完了していないトランザクションをロールバックするのは、障害発生時である。誤り。
- イ：更新したデータの更新前後情報ログをログファイルに書き出すのは、コミットの前後になる。誤り。
- ウ：データベースの内容をバックアップファイルに書き出すのは、任意のタイミングになる。誤り。

問15 a～c それぞれの障害に対して、DBMSはロールフォワード又はロールバックを行い回復を図る。適切な回復手法の組合せはどれか。

- a デッドロックによるトランザクション障害
- b ハードウェアの誤動作によるシステム障害
- c データベースの記録媒体が使用不可能となる媒体障害

	a	b	c
ア	ロールバック	ロールフォワード又は ロールバック	ロールバック
イ	ロールバック	ロールフォワード又は ロールバック	ロールフォワード
ウ	ロールフォワード	ロールバック	ロールフォワード又は ロールバック
エ	ロールフォワード又は ロールバック	ロールフォワード	ロールバック

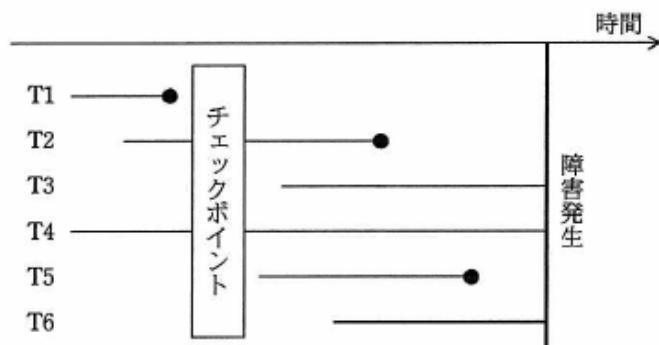
問15：正解（イ）

DBMSの持つ障害回復機能に関する問題。発生した障害と処理の関係は、それぞれ次のようになる。適切な組合せはイになる。

- a : トランザクション障害は、ロールバック（後退復帰）してトランザクション開始前の状態に戻し、原因を取り除いた後に再実行する。
- b : DBMS以外のシステム障害に関しては、障害発生時の状況に応じて、ロールフォワード（前進復帰）、ロールバック（後退復帰）を組み合わせて処理する。
- c : 媒体障害については、媒体交換後に、障害発生直前までのログを利用してロールフォワード（前進復帰）させる。

問14 DBMS をシステム障害発生後に再立ち上げするとき、ロールフォワードすべきトランザクションとロールバックすべきトランザクションの組合せとして、適切なものはどれか。ここで、トランザクションの中で実行される処理内容は次のとおりとする。

トランザクション	データベースに対する Read 回数 と Write 回数
T1, T2	Read 10, Write 20
T3, T4	Read 100
T5, T6	Read 20, Write 10



——— はコミットされていないトランザクションを示す。
● はコミットされたトランザクションを示す。

	ロールフォワード	ロールバック
ア	T2, T5	T6
イ	T2, T5	T3, T6
ウ	T1, T2, T5	T6
エ	T1, T2, T5	T3, T6

問 14：正解（ア）

障害回復機能に関する問題（参照 第4章）。複数トランザクションが実行されている環境で、システム全体に影響がある障害が発生した場合、更新前ログを使った後退復帰（ロールバック）と、更新後ログを使った前進復帰（ロールフォワード）を使い分けながら復旧する。今回の場合は、DBMSを再起動したときに行われる暗黙の復旧で、次のようなルールで処理される。

- ①チェックポイントまでのデータは、ディスクに反映されている（確定している）→ T1は何もしなくてもよい
- ②障害発生時点で、コミットされているトランザクションは、更新後ログを使って前進復帰（ロールフォワード）させる→ T2, T5
- ③障害発生時点で、コミットされていないトランザクションは、更新前ログを使って後退復帰（ロールバック）させる→ T3, T4, T6（ただし、T3, T4はRead処理のみのため戻す必要はない）

以上より、T2, T5は前進復帰、T6は後退復帰になるので、正解はアになる。

問14 データベースの障害回復処理に関する記述のうち、適切なものはどれか。

- ア 異なるトランザクション処理プログラムが、同一データベースを同時更新することによって生じる論理的な矛盾を防ぐために、データのブロック化が必要となる。
- イ システムが媒体障害以外の原因によって停止した場合、チェックポイントの取得以前に終了したトランザクションについての回復作業は不要である。
- ウ データベースの媒体障害に対して、バックアップファイルをリストアした後、ログファイルの更新前情報を使用してデータの回復処理を行う。
- エ トランザクション処理プログラムがデータベースの更新中に異常終了した場合には、ログファイルの更新後情報を使用してデータの回復処理を行う。

問20 ディザスタリカバリを計画する際の検討項目の一つである RPO (Recovery Point Objective) はどれか。

- ア 業務の継続性を維持するために必要な人員計画と交代要員の要求スキルを示す指標
- イ 業務を代替する遠隔地のシステム環境と、通常稼働しているシステム環境との設備投資の比率を示す指標
- ウ 災害発生時からシステムを再稼働するまでの時間示す指標
- エ システムが再稼働したときに、データが災害発生前のどの時点の状態まで復旧されなければならないかを示す指標

問 14：正解（イ）

チェックポイントとは、バッファキャッシュの内容とデータファイルの内容の同期を取ることである。従って、チェックポイントの取得以前に終了したトランザクションはデータファイルに反映済のため、回復作業は不要になる。したがって正解は（イ）である。

ア：「ブロック化」ではなく、「ロック」が正しい。誤り。

ウ：媒体障害に対しての回復処理は、バックファイルをリストアした後、ログファイルの更新後情報を用いてロールバックする。誤り。

エ：障害発生前にトランザクションがコミットしていた場合には、ログファイルの更新後情報を用いてロールフォワードする。まだコミットしていなかった場合には、ログファイルの更新前情報を用いてロールバックする。今回は後者なので更新前情報を使用しないといけない。誤り。

※誤植。選択肢ウの解説：ロールバックではなく、ロールフォワードの誤り。

※選択肢エの解説。「今回は後者」というのは、トランザクションが終了していない=コミットしていないということ。いわゆるトランザクション障害のこと。

問 20：正解（エ）

SLA の指標に関する問題。RPO (Recovery Point Objective) が問われている。RPO とは、システムが再稼働したときに、データが災害や障害で停止するまでのどの時点まで復旧させなければならないかを示す指標になる。したがって、正解はエになる。ちなみに、選択肢のウは RTO である。

問14 データベースの REDO のべき等 (idempotent) の説明として、適切なものはどれか。

- ア REDO による障害回復の時間を短縮するために、あるルールに従って整合性の取られたデータを記録媒体に適宜反映すること
- イ REDO を繰返し実行しても、正常終了するときには 1 回実行したときと同じデータの状態になること
- ウ 事前に取得していたバックアップデータを記録媒体に復旧し、そのデータに対して REDO を実行すること
- エ トランザクションをコミットする前に REDO に必要な情報を書き出し、データの更新はその後で行うこと

問 14：正解（イ）

データベースの障害回復関連用語に関する問題。REDO のべき等 (idempotent) について問われている。REDO (リドゥ) は「やり直し」という意味で、データベースの復旧時に使う REDO ログファイルの名称などに使われる用語である。また、「べき等」とは、ある操作を何回実行しても実行結果が同じになる性質のことという。つまり、REDO を繰り返し実行しても、正常終了する時には 1 回実行した時と同じデータの状態になることなので、選択肢イが正解になる。

- ア：チェックポイントの説明。
- ウ：リカバリ処理の説明。
- エ：WAL (Write Ahead Log) の説明。

4.4 索引（インデックス）

平成 21 年度・D B・午前Ⅱ

問10 関係データベースの表において、検索速度を向上させるために、列 Z にインデックスを付与する。ア～エは、列 Z の値が等しい行の数を示したものである。インデックスを付与することによって、1 行当たりの平均検索速度が最も向上するものはどれか。ここで、各行は等頻度で検索されるものとする。

ア

データ値	行の数
p	600
q	600
r	0
s	0
t	0
u	0

イ

データ値	行の数
p	1000
q	200
r	0
s	0
t	0
u	0

ウ

データ値	行の数
p	20
q	40
r	80
s	160
t	300
u	600

エ

データ値	行の数
p	200
q	200
r	200
s	200
t	200
u	200

問 10：正解工

B木インデックスに関する問題（参照▶ 4.4.3 B木）。B木インデックスを用いる場合、1行当たりの平均検索効率が良くなる条件は、データ値の種類が多いことである。よって、ウ、エの方がア、イよりも優れている。次にウとエを比較する。このように、種類の数が等しい場合、各々のデータ値の行数を比較してその偏りの小さい方が、平均検索効率が良い。よってエの方がウよりも優れている。

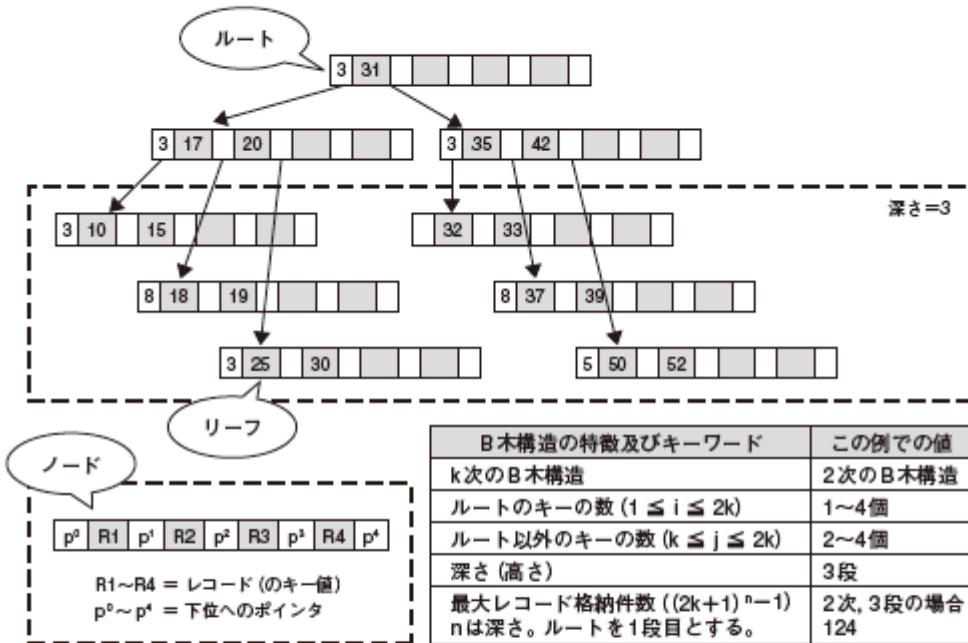
問2 k 次のB木構造において、ルートノードは i 個 ($1 \leq i \leq 2k$) のレコードをもち、ルート以外のノードは j 個 ($k \leq j \leq 2k$) のレコードをもつものとする。ルートノードを1段目とした場合、B木は1段目から n 段目までに最大何レコードを格納することができるか。ここで、 k, n は自然数とし、 $n \geq 2$ とする。

ア $(2k+1)^{n-1}-1$
ウ $2(k+1)^{n-1}-1$

イ $(2k+1)^n-1$
エ $2(k+1)^n-1$

問2：正解（イ）

B木インデックスに関する問題（参照 第4章）。全体像は次のようになる。



問題文にある「k次のB木構造」の“k次”とは“位数”とよばれている。ルートノード以外のノードは、常にその“位数”以上の数のキー値を持つように分割等をされるようになっている（“ $k \leq j \leq 2k$ ”の最小値がkだということ）。すべてのノードは、“位数”的2倍を上限とする数のキー値しか取れないので、ルートノード以外のノードの格納率は常に50%以上になる。

最大レコード数は、次の計算式で求められる。

上記の図を見ながら、各段数のノード数がどのように増えていくのかを下表のように整理してみるとわかりやすい。

深さ (段数)	各段数のノード数		合計ノード数	
	k 次	次数=2 の例	k 次	次数=2 の例
1	1	1	1	1
2	$2k+1$	5	$1 + 2k+1$	6
3	$(2k+1)^2$	25	$1 + 2k+1 + (2k+1)^2$	31
4	$(2k+1)^3$	125	省略	156
n	$(2k+1)^{n-1}$	—	$\Sigma (2k+1)^{n-1}$	—

各段数は、上位の一つのノードから $(2k + 1)$ 分ずつ増えていく。例えば2次のB木構造だと一つのノードのレコード数(キー値)は4つ($2k$)で、ポインタは5つ($2k + 1$)があるので、 $2k + 1$ ずつ(すなわち、 $1 \rightarrow 5 \rightarrow 25 \rightarrow 125$ と)増えていくことになる。これは、深さをnとすると、n段目のノード数は $(2k+1)^{n-1}$ であることを意味する。

全ノード数は、それらの合計数になるので下記のような等比数列になる。

$$\Sigma (2k+1)^{n-1} = (2k+1)^0 + (2k+1)^1 + (2k+1)^2 + \cdots + (2k+1)^{n-1}$$

これを等比数列の展開公式に当てはめると

$$\frac{(2k + 1)^n - 1}{2k + 1 - 1} = \frac{(2k + 1)^n - 1}{2k}$$

これに各ノードのレコード数が“ $2k$ ”なので、 $2k$ をかけければレコード数が求められる。

$$\frac{(2k + 1)^n - 1}{2k} \times 2k = (2k + 1)^n - 1$$

以上より選択肢イが正解になる。

但し、これは等比数列の公式を覚えていないと求められないので、この答えを覚えてしまうか、あるいは情報処理技術者試験の午前問題のように選択肢がある場合に限定されるが、次のように例えば $k = 2$ を選択肢の計算式に順次当てはめながら計算しても正解は得られる。現実的には、下記で正解を得るのが妥当かもしれない。

(例) $k = 2$ とした場合、2段目($n = 2$)の最大レコード数は24になる。

ア : $(2k + 1)^{n-1} - 1 = 5^1 - 1 = 4$ 誤り

イ : $(2k + 1)^n - 1 = 5^2 - 1 = 24$

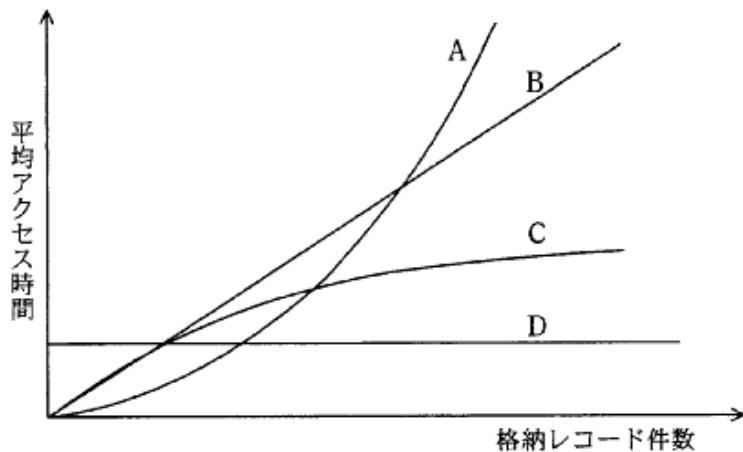
ウ : $2 (k + 1)^{n-1} - 1 = 5$ 誤り

エ : $2 (k + 1)^n - 1 = 17$ 誤り

※結果が24になるのは、選択肢イしかない。

memo

問40 次のグラフのうち、 B^+ 木インデックスを使用した検索を行った場合の、格納レコード件数と平均アクセス時間の関係を表すものはどれか。



ア A

イ B

ウ C

エ D

問4 B^+ 木インデックスが定義されている候補キーを利用して、1 件のデータを検索するとき、データ総件数 X に対する B^+ 木インデックスを格納するノードへのアクセス回数のオーダーはどれか。

ア \sqrt{X}

イ $\log X$

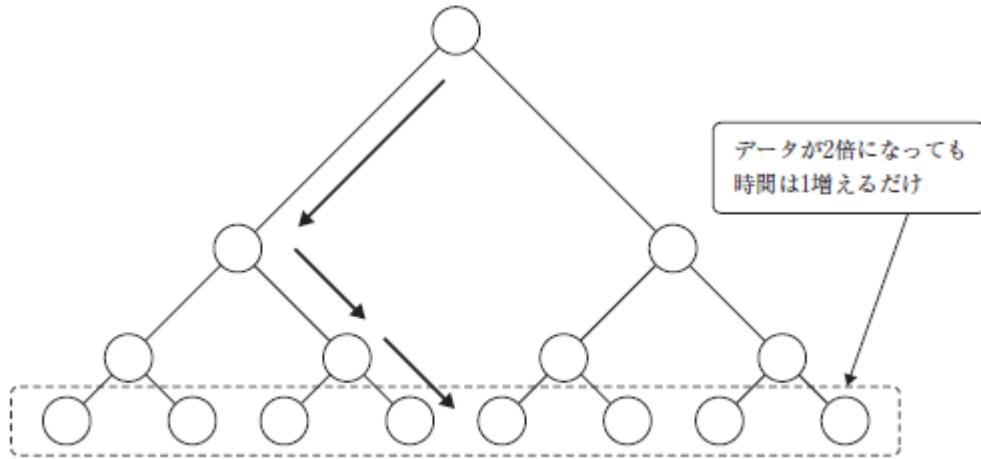
ウ X

エ $X!$

問 40：正解ウ

B+木インデックスの特徴は、下図のようにルート（根）からリーフ（葉）へと検索していくため、データが2倍になったときでも検索時間（コスト）は1増えるだけである。よって、データの増加に対して平均アクセス時間は緩やかにしか増加しないため、Cのグラフになる。正解はウ。

参照▶「4.4.4 B+木」



問 4：正解（イ）

B+木インデックスに関する問題。1件のデータを検索するときのノードへのアクセス回数のオーダーが問われている。B木やB+木の特徴をもとに考えれば解ける問題。B+木インデックスは多分木であり、深さが一定だという性質を持つ。したがって、実際に綺麗なバランス木をイメージしながら考えればいいだろう。例えば、ノードの数が4つで深さが3階層の場合、ノードへのアクセス回数は深さの回数なので3回になる。この時、データの総件数は4の3乗の64個になる。この時の「深さ」は、データの総件数の対数になるため、ノードへのアクセス回数のオーダーはイになる。

問13 “部品”表のメーカーコード列に対し, B^+ 木インデックスを作成した。これによって, “部品”表の検索の性能改善が最も期待できる操作はどれか。ここで, 部品及びメーカーのデータ件数は十分に多く, “部品”表に存在するメーカーコード列の値の種類は十分な数があり, かつ, 均一に分散しているものとする。また, “部品”表のごく少数の行には, メーカーコード列に NULL が設定されている。実線の下線は主キーを, 破線の下線は外部キーを表す。

部品 (部品コード, 部品名, メーカーコード)

メーカー (メーカーコード, メーカー名, 住所)

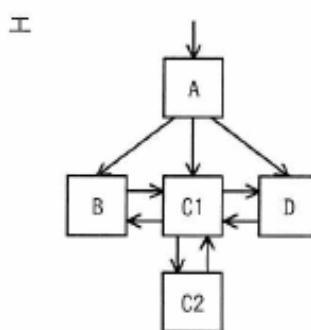
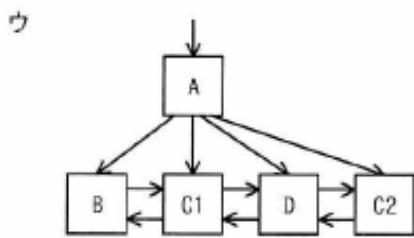
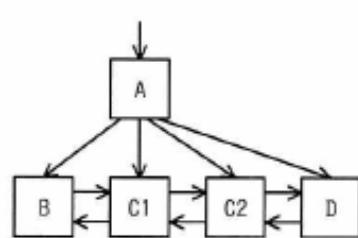
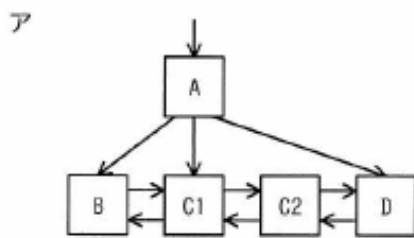
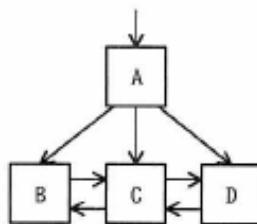
- ア メーカーコードの値が 1001 以外の部品を検索する。
- イ メーカーコードの値が 1001 でも 4001 でもない部品を検索する。
- ウ メーカーコードの値が 4001 以上, 4003 以下の部品を検索する。
- エ メーカーコードの値が NULL 以外の部品を検索する。

問 13：正解（ウ）

B + 木インデックスに関する問題。B + 木インデックスを使った場合、等号や不等号、BETWEEN を使った範囲指定などで期待通りの効果を得られるが、否定系 (< >, !=, NOT IN など) や NULL を指定した場合には索引検索にならず期待していた効果が得られない。そのあたりの知識が問われている。

- ア : B + 木インデックスを作成した列に否定系を使うと、インデックスは適用されない。
イ : 上記同様、否定系を使っているのでインデックスは適用されない。
ウ : BETWEEN を使った範囲指定 (BETWEEN 4001 AND 4003) ではインデックスが適用されるため性能改善が期待できる。これが正解。
エ : NULL を使う場合 (IS NULL や IS NOT NULL) もインデックスは適用されない。

問3 関係データベースのテーブルにレコードを1件追加したところ、インデックスとして使う、B⁺木のリーフノードCがノードC1とC2に分割された。ノード分割後のB⁺木構造はどれか。ここで、矢印はノードへのポインタとする。また、中間ノードAには十分な空きがあるものとする。



問3：正解（イ）

索引（インデックス）に関する問題。B⁺木インデックスについて問われている。B⁺木インデックスは、多分木（各ノードが複数の子を持つことができる木構造）であり、すべてのリーフノードの深さが一定であるという性質を持つ。加えて、リーフノード同士がポインタで接続されており、連結リストのように順序を保持しているため、範囲検索が効率的に行える。今回は、リーフノードの一つ“C”が“C1”と“C2”に分割された時の“分割後の構造”について問われている。選択肢を順番に見ていく。

- ア：リーフノード“C”が分割されると、「中間ノードAには十分な空きがあるものとする」ということなので、中間ノードAの分割は発生せず、中間ノードAのポインタ構造が変更される。具体的には、中間ノードAからのポインタは“C”から、新しい“C1”と“C2”的両方を指すように修整される。したがって正しくない。
- イ：B⁺木の特性を満たし、分割後の構造が適切に維持されている。これが正しい。
- ウ：分割後のリーフノードの順序関係は維持されなければならず、“C1”的次は必ず“C2”になる。したがって正しくない。
- エ：深さが一定になっていない。したがって正しくない。

問15 B^+ 木インデックスとビットマップインデックスを比較した説明のうち、適切なものはどれか。

- ア AND 操作や OR 操作だけで行える検索は、 B^+ 木インデックスの方が有効である。
- イ BETWEEN を用いた範囲指定検索は、ビットマップインデックスの方が有効である。
- ウ NOT を用いた否定検索は、 B^+ 木インデックスの方が有効である。
- エ 少数の異なる値をもつ列への検索は、ビットマップインデックスの方が有効である。

問42 ハッシュインデックスの特徴に関する記述として、適切なものはどれか。

- ア B 木インデックスと比較して、不等号の条件検索が困難である。
- イ B 木インデックスと比較して、ワイルドカード式の検索が容易である。
- ウ インデックスノードが木構造になっており、複数のノードを経由してレコードへアクセスする。
- エ レコードの追加や削除が多くなっても、インデックスの再構成の必要がない。

問 15：正解（工）

インデックス管理に関する問題。B⁺木インデックスとビットマップインデックスの特徴が問われている。選択肢を順番にチェックしていく。

- ア：(WHERE句の) AND や OR 操作だけで行える検索が有利なのはビットマップインデックスの方である。したがって誤り。
- イ：範囲指定検索では、シーケンシャル処理（順次アクセス）の高速性が求められる。B⁺木の場合、シーケンスセットを持っているので、それが可能。したがって誤り。
- ウ：ある属性に対して NOT を用いた場合、ビットマップインデックスの方が有効である。したがって誤り。
- エ：正しい。

問 42：正解ア

ハッシュインデックスとは、ハッシュ関数を利用して、キー値に一致する行をダイレクトに検索できるようにしたインデックスのことである。B木インデックスと比較して、キー値に一致する検索は高速であるが、逐次処理や不等号の条件検索、曖昧検索（ワイルドカード式検索）などは困難である。

- イ：ワイルドカード式の検索は困難である。
- ウ：木構造にはなっていない。
- エ：再構成は必要である。

問4 転置インデックスに関する記述として、適切なものはどれか。

- ア SQL 関数を評価した結果の値をインデックスとして使用する。
- イ 最上位のノードから、実データへのポインタを格納したリーフノードへと至るポインタをインデックスとして使用する。
- ウ テキストに含まれる単語に対して、その単語を含むテキストへのポインタをインデックスとして使用する。
- エ ヒープ領域を使用せずに実データを物理的に並べ替えたデータをインデックスとして使用する。

問4：正解（ウ）

索引（インデックス）に関する問題。転置インデックスについて問われている。転置インデックスは、検索エンジンや全文検索システムなど、大量のテキストデータを扱うシステムで使用される。単語をキーとして、その単語を含む文書のリストを保持することで、全文検索を高速化する。したがって、選択肢ウの説明になる。

- ア：関数インデックス（参照 ➤ 用語集）の説明になる
- イ：B+木インデックス（参照 ➤ 本書の第4章）の説明になる
- ウ：これが転置インデックス（参照 ➤ 用語集）の説明である
- エ：クラスタ化インデックス（参照 ➤ 用語集）の説明になる

4.5 分散データベース

平成 28 年度・D B・午前Ⅱ

問18 分散データベースシステムの目標の一つである“移動に対する透過性”の説明として、適切なものはどれか。

- ア 運用の都合や性能向上の目的で表の格納サイトが変更されても、利用者にこの変更を意識させないで利用可能にする機能のことである。
- イ データベースが通信網を介して物理的に分散配置されていても、利用者にこの分散状況を意識させないで利用可能にする機能のことである。
- ウ 一つの表が複数のサイトに重複して格納されていても、利用者にこれを意識させないで利用可能にする機能のことである。
- エ 一つの表が複数のサイトに分割して格納されていても、利用者にこれを意識させないで利用可能にする機能のことである。

平成 23 年度・D B・午前Ⅱ

問20 分散データベースにおける“複製に対する透過性”の説明として、適切なものはどれか。

- ア それぞれのサーバの DBMS が異種であっても、プログラムは DBMS の相違を意識する必要がない。
- イ 一つの表が複数のサーバに分割されて配置されていても、プログラムは分割された配置を意識する必要がない。
- ウ 表が別のサーバに移動されても、プログラムは表が配置されたサーバを意識する必要がない。
- エ 複数のサーバに一つの表が重複して存在しても、プログラムは表の重複を意識する必要がない。

問 18：正解（ア）

分散データベースにおける“透過性”についての頻出問題である。今年度は“移動に対する透過性”が出題されている。解答に当たっては選択肢を順番に見ていく。

- ア：移動に対する透過性（正しい）
- イ：分散に対する透過性（誤り）
- ウ：重複に対する透過性（誤り）
- エ：分割に対する透過性（誤り）

問 20：正解工

NEW

参照 第4章 重要キーワード「6 分散データベース」

分散データベースに関する問題。データベースの透過性のうち、今回は“複製に対する透過性”が問われている。複製が問われるのは過去のDB午前問題では平成14年まで遡っても出題されていないが、移動に対する透過性（H15）、分割に対する透過性（H19）など、違う切り口では何度も出題されている。ちなみに複製に対する透過性とは、分散されたサイトでデータを重複して持っていても利用者は意識することがない特徴のこと。選択肢ではエになる。

- ア：「DBMSからの独立」に関する記述。誤り。
- イ：「分割からの独立（透過性）」に関する記述。誤り。
- ウ：「移動からの独立（透過性）」に関する記述。誤り。
- エ：正しい。

問36 分散データベースシステムにおける“分割に対する透過性”を説明したものはどれか。

- ア データの格納サイトが変更されても、ユーザのアプリケーションや操作法に影響がないこと
- イ 同一のデータが複数のサイトに格納されていても、ユーザはそれを意識せずに利用できること
- ウ 一つの表が複数のサイトに分割されて格納されていても、ユーザはそれを意識せずに利用できること
- エ ユーザがデータベースの位置を意識せずに利用できること

問12 分散データベースシステムにおける問合せでは、サイト間にまたがる結合演算の最適化が重要である。この最適化の方法として、適切なものはどれか。

- | | |
|-----------|-----------|
| ア LRU 法 | イ セミジョイン法 |
| ウ ソートマージ法 | エ 楽観的方法 |

問36：正解ウ

分散データベースにおける“透過性”についての頻出問題である。今年度は、分割に対する透過性が出題されている。解答に当たっては選択肢を順番に見ていく。

参照 ➤ 「4.3.1 分散データベース機能」

ア：移動に対する透過性（誤り）

イ：重複に対する透過性（誤り）

ウ：分割に対する透過性（正解）

エ：位置に対する透過性（誤り）

問12：正解イ

分散データベースにおける分散問合せ処理に関する問題（参照 ➤ 4.3.2 分散問合せ処理）。分散問合せ処理の方式には、選択肢にあるように、セミジョイン法やソートマージ法、ほかに入れ子ループ法などがある。このうち、「結合演算の最適化」を行って通信料を削減させる目的を持っているのは、セミジョイン方式である。したがって正解はイとなる。

なお LRU 法（参照 ➤ 用語集）は、キャッシュメモリのアクセス管理の方法などに使われるもので、楽観的方法（参照 ➤ 4.1.3 排他制御（同時実行制御））はデッドロックの検出方法の説明である。

問15 関係データベースにおいて、タプル数 n の表二つに対する結合操作を、入れ子ループ法によって実行する場合の計算量はどれか。

- ア $O(\log n)$ イ $O(n)$ ウ $O(n \log n)$ エ $O(n^2)$

問17 分散型データベースで結合演算を行うとき、通信負荷を最も小さくすることができる手法はどれか。ここで、データベースは異なるコンピュータ上に格納されて、かつ結合演算を行う表の行数が、双方で大きく異なるものとする。

- ア 入れ子ループ法 イ インデックスジョイン法
ウ セミジョイン法 エ マージジョイン法

問 15：正解（工）

分散データベースに関する問題（参照▶第4章：WEB 掲載）。入れ子ループ法で等結合を実行する場合、次の要領で処理が実行される。

- ①二つのテーブルをそれぞれ、X、Yとする。
- ②X から 1 行取り出す。（これを“行 X1”とする）
- ③結合処理
 - ③ – 1. Y で、行 A の結合列と等しい値を持つ行を 1 つずつ取り出す。（これを“行 Y1”とする）
 - ③ – 2. “行 X1”と“行 Y1”を結合する。
 - ③ – 3. ③ – 1. に戻って、Y の次の 1 行を処理する。
- ④Y の全ての行の処理が終われば、②に戻り X から次の 1 行を取り出して③の結合処理を行う。

処理②④では X テーブルを全件処理するので n 回のアクセスが発生する。処理③では、X テーブル 1 件ごとに毎回、Y テーブルを全件処理するので、計算量は $n \times n = n^2$ となる。よって、正解は工である。

問 17：正解（ウ）

分散型データベースに関する問題。分散型データベースにおける分散問合せ処理方式には、「入れ子ループ法」、「セミジョイン法」、「マージジョイン法」などがある。このうち、通信負荷を軽減するために考えられたのが、セミジョイン方式である。セミジョイン方式は準結合法ともいい、別々のサイトに存在する関係 R、S のうち、片方の関係（例えば関係 R）から結合の対象となる列のみを取り出し、もう片方のサイト（関係 S）に送信する。したがって、正解はウとなる。

問18 分散型 DBMSにおいて、二つのデータベースサイトの表で結合を行う場合、どちらか一方の表をもう一方のデータベースサイトに送る必要がある。その際、表の結合に必要な列値だけを送り、結合に成功した結果を元のデータベースサイトに転送して、最終的な結合を行う方式はどれか。

- | | |
|---------------|------------|
| ア 入れ子ループ法 | イ セミジョイン法 |
| ウ ハッシュセミジョイン法 | エ マージジョイン法 |

問11 表の結合演算アルゴリズムのうち、等結合だけに適用できるものはどれか。

- | | |
|-----------|---------|
| ア 入れ子ループ法 | イ 索引結合法 |
| ウ ソートマージ法 | エ ハッシュ法 |

問 18：正解（イ）

過去問題（平成18年・DB午前 問39）

参照 第4章 重要キーワード「6 分散データベース」

問題文で記述されている「表の結合に必要な列値だけを送り」、「結合に成功した結果を元のデータベースサイトに転送」する方法はセミジョイン法である。よって正解は（イ）になる。

ア：入れ子ループ法（参照→本書第4章）

ウ：ハッシュセミジョイン法（参照→本書第4章）

エ：マージジョイン法（参照→本書第4章）

問 11：正解（エ）

表の結合演算アルゴリズムに関する問題。選択肢のうち、等結合だけにしか適用できないのはハッシュ法である。これは覚えておこう。ハッシュ法では、結合する列に対してハッシュ関数を用いてハッシュ値を求め、それが同じものを結合する。そのため、等結合にしか使えない。したがって正解はエになる。他の結合は、等結合以外にも使用できる。

問12 分散データベースのトランザクションが複数のサブトランザクションに分割され、複数のサイトで実行されるとき、トランザクションのコミット制御に関する記述のうち、適切なものはどれか。

- ア 2相コミットでは、サブトランザクションが実行される全てのサイトからコミット了承応答が主サイトに届いても、主サイトはサブトランザクションごとにコミット又はロールバックの異なる指示を出す場合がある。
- イ 2相コミットを用いても、サブトランザクションが実行されるサイトに主サイトの指示が届かず、サブトランザクションをコミットすべきかロールバックすべきか分からぬ場合がある。
- ウ 2相コミットを用いると、サブトランザクションがロールバックされてもトランザクションがコミットされる場合がある。
- エ 集中型データベースのコミット制御である1相コミットで、分散データベースを構成する個々のサイトが独自にコミットを行っても、サイト間のデータベースの一貫性は保証できる。

問14 データベース更新における2相コミットに関する記述のうち、適切なものはどれか。

- ア 2相コミットは、トランザクションの処理途中のデータを他のトランザクションから参照できなくなる制御方式のことである。
- イ 2相コミットを行うためには、同時に更新しようとする分散データベースの全てが更新可能かどうかを判断するためのやり取りが必要である。
- ウ 2相コミットを採用している場合、ロールバックは発生しない。
- エ 2相コミットを使えば、通信異常が発生しても、トランザクションをコミットさせることができる。

問 12：正解（イ）

分散データベースに関する問題（参照▶第4章：WEB掲載）。分散データベースのコミット制御は、ACID特性のうち“原子性”（Atomicity）や“一貫性”（Consistency）を確保するために必要な機能である。そのため、たとえデータベースが分散されていたとしても、トランザクションは完全に実行されるか、まったく実行されないかのどちらかでなければならない。それをイメージしながら選択肢を順番に見ていく。

ア：これだと原子性が保障されない。

イ：主サイトからの指示が（サブトランザクションが実行されるサイトに）届かない場合は、サブトランザクション側ではセキュアな状態のままコミットすべきかロールバックすべきか判断がつかない。したがって正しい。

ウ：これだと原子性、一貫性が保障されない。

エ：分散されたサブトランザクションを個々にコミットすると原子性、一貫性が保障されない。

問 14：正解（イ）

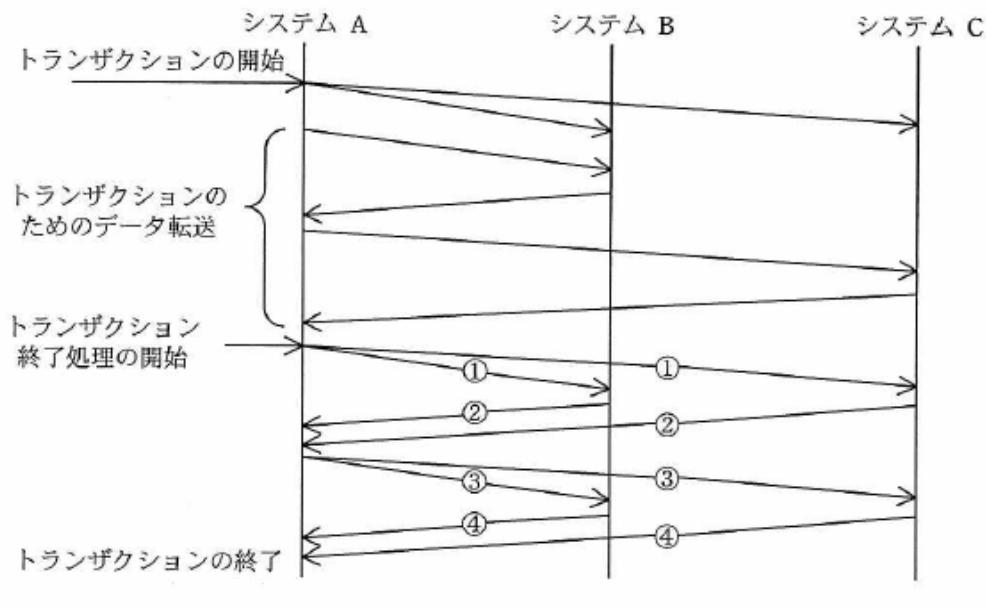
分散データベースに関する問題（参照▶第4章）。2相コミット制御について問われている。2相コミットでは、同時に更新しようとする分散データベースの全てが更新可能になった時にコミットし、ひとつでも更新に失敗すると全ての更新を取り消す。したがって正解はイになる。

ア：排他制御（専有ロック）の説明。

ウ：分散されたサブトランザクション（分散データベースのどこかひとつのトランザクション）で異常が発生した場合、他のサブトランザクション（他の分散サイト等）はロールバックして更新前の状態に戻す。

エ：分散されたサブトランザクションの個々の更新完了を受け取ることができないため、コミットはできない。

問12 図は、分散システムにおける 2 相コミットプロトコルの正常処理の流れを表している。③の動作はどれか。



ア アンロック実行指示

イ コミット可否問合せ

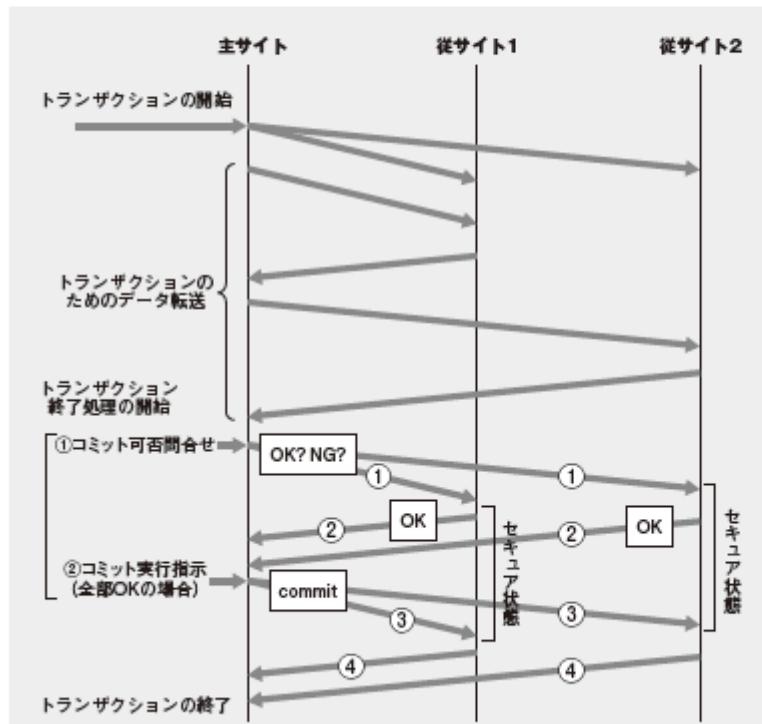
ウ コミット実行指示

エ ログ取得指示

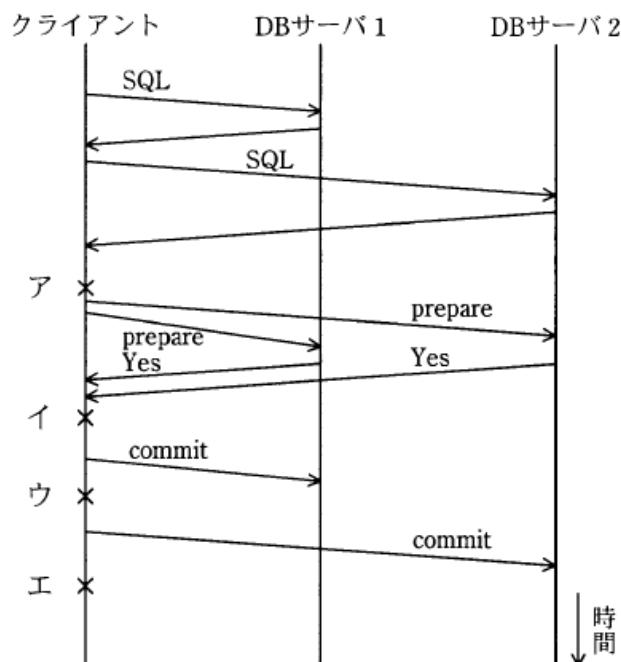
問 12：正解（ウ）

NEW (ただし、類似問題多数)
参照 「4.7 分散データベース」

2相コミット処理に関する問題。一連の流れはこの図のようになる。2相ということで、1回目に“コミット可否問合せ”を実施し、すべてOKが返ってきた段階で2回目の“コミット実行”の指示を出す。したがって、ここで問われている③はウになる。



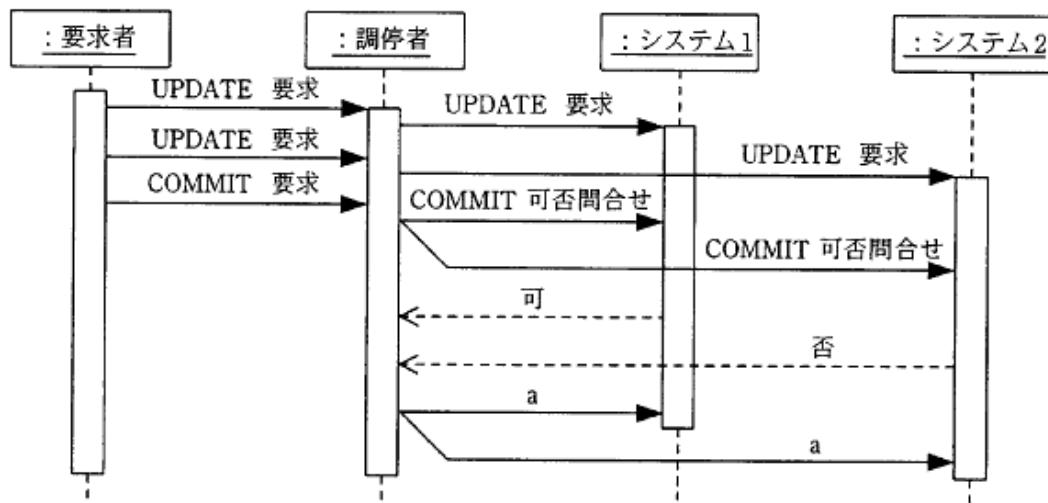
問37 2 相コミットプロトコルを使用した分散データベースにおいて、クライアント障害が発生した場合、各データベースサーバ（DB サーバ）はコミットすべきかアボートすべきか判断不能（ロック状態）になることがある。DB サーバ 1, 2 のどちらもロック状態になる箇所はどこか。



問37：正解イ

2相コミットメントプロトコルでは、更新が終了した時点で、クライアント側からコミット可能かどうかを問い合わせる(prepare)。DBサーバ側では、コミット可能な場合、Yesを返した後、クライアントからのcommitまたはabortの実行指示を待つ。この間をセキュアな状態という。問題文では、「コミットすべきかアボートすべきか判断不能(ロック状態)になる」箇所を問われているので、セキュア状態からコミット指示待ちのイが正解になる。

問38 分散データベースにおいて図のようなコマンドシーケンスがあった。調停者がシーケンス a で発行したコマンドはどれか。ここで、コマンドシーケンスの記述に UML のシーケンス図の記法を用いる。



- ア COMMIT の実行要求
ウ 判定レコードの書出し要求

- イ ROLLBACK の実行要求
エ ログ書出しの実行要求

問 38：正解イ

分散データベースの基本的な問題。調停者は、システム1、システム2の両方から COMMIT “可”が戻ってきたときに初めて、COMMITの実行要求を出す。今回はシステム2から“否”が戻ってきてるので、システム1、システム2に対して、ROLLBACKの実行要求を出さなくてはいけない。よって正解はイになる。

参照 ➤ 「4.3.3 分散トランザクション制御」

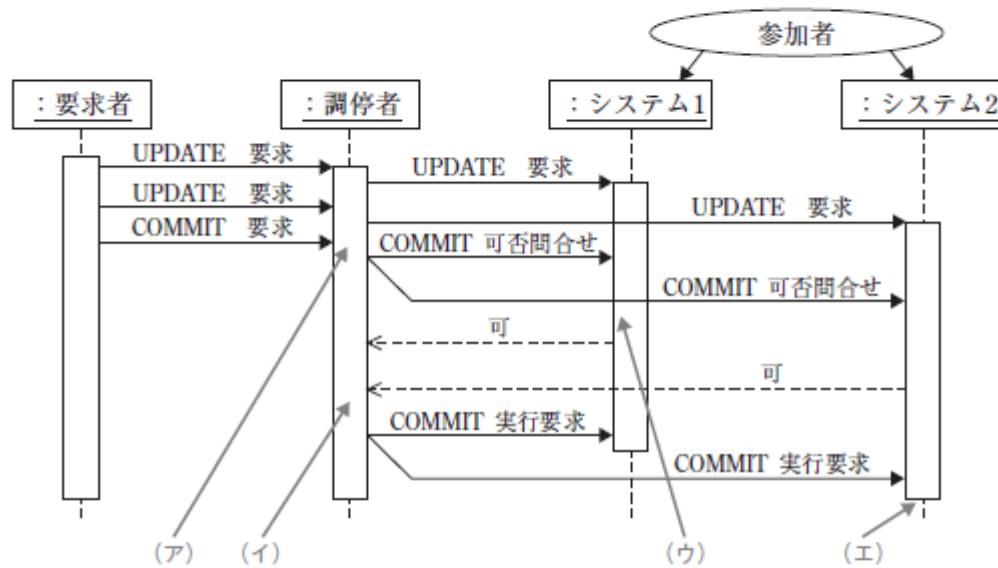
問13 2 相コミットで分散トランザクションの原子性を保証する場合、ネットワーク障害の発生によって参加者のトランザクションが、コミットすべきかロールバックすべきかを判断できなくなることがある。このような状況を発生させるネットワーク障害に関する説明として、適切なものはどれか。

- ア 調停者のトランザクションが、コミット又はロールバック可否の問合せを参加者に送る直前に障害になった。
- イ 調停者のトランザクションが、コミット又はロールバックの決定を参加者に送る直前に障害になった。
- ウ 調停者のトランザクションに、コミット又はロールバック可否の応答を参加者が返す直前に障害になった。
- エ 調停者のトランザクションに、コミット又はロールバックの完了を参加者が返す直前に障害になった。

問 13：正解（イ）

NEW（但し、類似問題多数）
参照 第4章 重要キーワード「6 分散データベース」

分散トランザクションの原子性とは、分散配置されていようがいまいが、完全に実行されるか、まったく実行されないか、それを保証する性質のことをいう。具体的には、2相コミットを用いて、コミットすべきか（完全に実行すべきか）、ロールバックすべきか（まったく実行しないか）を決めることがある。一連の流れはこの図のようになる。



この図をイメージして、問題文のようにネットワーク障害が発生して、参加者のトランザクションが、コミットすべきかロールバックすべきかを判断できなくなるケースを、選択肢から探せばいい。

- ア：ここで障害になった場合、ロールバックする。
- イ：これが正解。参加者が、コミット／ロールバックの実行要求を待っている段階になる。
ここで障害が発生したら、参加者のトランザクションは判断できなくなる。
- ウ：参加者は、まだ COMMIT 可能か不可能かの返事ができないので、一定時間経過後にロールバックする。
- エ：判断どころか、コミットかロールバックかが完了している。

問15 分散データベースシステムにおいて、複数のデータベースサイトを更新する場合に用いられる 2 相コミットの処理手順のうち、適切なものはどれか。

- ア 主サイトが各データベースサイトにコミット準備要求を発行した場合、各データベースサイトは、準備ができていない場合だけ応答を返す。
- イ 主サイトは、各データベースサイトにコミットを発行し、コミットが失敗した場合には、再度コミットを発行する。
- ウ 主サイトは、各データベースサイトのロックに成功した後、コミットを発行し、各データベースサイトをアンロックする。
- エ 主サイトは、コミットが可能であることを各データベースサイトに確認した後、コミットを発行する。

問 15：正解（工）

分散データベースの基本的な問題。2相コミットの処理手順が問われている。選択肢を順番に見ていくばれば良い。

- ア：各データベースサイトは、コミット準備要求に対して、準備ができている場合には“可”や“OK”を、準備ができていない場合には“否”や“NG”を返す。つまり、いずれの状態でも応答を返すため、誤り。
- イ：各データベースサイトに、2フェーズ目でコミットを発行したあとに、コミット失敗を発見しても2相コミットではどうしようもない。よって誤り。
- ウ：ロックは関係ない。ロックに成功したからコミットを発行するのではない。よって誤り。
- エ：正しい。

問 1 分散型データベースシステムにおいては、一貫性・可用性・分断耐性の三つの特性のうち、同時には最大二つまでしか満たすことができないとする理論はどれか。

ア BASE 特性

イ CAP 定理

ウ アムダールの法則

エ ベイズの定理

問1：正解（イ）

分散型データベースに関する問題。問題文の説明は CAP 定理のものなので、選択肢イが正解になる。CAP 定理とは、次の3つの性質の頭文字をとって名付けられた概念である（それぞれの意味をざっくりと説明すると下記のようになる）。このうち3つは同時に満たすことはできず、最大で2つまでしか満たすことしかできないという理論である。Brewer 氏が発表した理論なので、ブリュワーの定理とも言われている。

- ①一貫性（Consistency）全てのサーバで同じ結果になる性質
- ②可用性（Availability）いずれかのサーバで処理が継続できる性質
- ③分断耐性（Partition-tolerance）一時的にネットワークが分断しても、いずれ元の状態に戻せる性質

ア：BASE 特性（[参照 ➤ 用語集](#)）

ウ：アムダールの法則（[参照 ➤ 用語集](#)）

エ：ベイズの定理（[参照 ➤ 用語集](#)）

問 1 CAP 定理に関する記述として、適切なものはどれか。

- ア システムの可用性は基本的に高く、サービスは利用可能であるが、整合性については厳密ではない。しかし、最終的には整合性が取れた状態となる。
- イ トランザクション処理は、データの整合性を保証するので、実行結果が矛盾した状態になることはない。
- ウ 複数のトランザクションを並列に処理したときの実行結果と、直列で逐次処理したときの実行結果は一致する。
- エ 分散システムにおいて、整合性、可用性、分断耐性の三つを同時に満たすことはできない。

問1：正解（エ）

分散型データベースにおける CAP 定理に関する問題。CAP 定理とは、次の3つの性質の頭文字をとった名付けられた理論である。Brewer 氏が発表した理論なので、ブリューワーの定理ともいわれている。このうち3つは同時に満たすことはできず、最大で2つまでしか満たすことしかできないとしている。

- ①一貫性（Consistency）：同じデータ（单一、かつ最新の確定データ）を得られること。
平成31年午前Ⅱ問1では“一貫性”と訳していたが、今回は“整合性”と訳している。
- ②可用性（Availability）：単一障害点（そこで障害が発生したらシステム全体が停止してしまう機器等）が存在しないこと。どこかに障害が発生しても、常にデータにアクセスできること。
- ③分断耐性（Partition-Tolerance）：複数のグループに分散保管し、そのグループ間に通信障害などが発生しても、システムを利用できること（全面停止しないこと）。

これらの基本的な知識をもとに、選択肢を順番に見ていく。選択肢を見ると、すべてデータベースの持つ特性に関する説明になっていることがわかるだろう。

- ア：BASE 特性に関する説明。
- イ：ACID 特性の一貫性（Consistency）に関する説明。
- ウ：ACID 特性の独立性（Isolation）及び直列化可能性（Serializability）に関する説明。
- エ：これが CAP 定理に関する説明になる。正解。

問2 大文字のアルファベットで始まる膨大な数のデータを、規則に従って複数のノードに割り当てる。このようにあらかじめ定めた規則に従って、複数のノードにデータを分散して割り当てる方法はどれか。

〔規則〕

- ・データの先頭文字が A～G の場合はノード 1 に格納する。
- ・データの先頭文字が H～N の場合はノード 2 に格納する。
- ・データの先頭文字が O～Z の場合はノード 3 に格納する。

ア 2 相コミットプロトコル
ウ シャーディング

イ コンシスティントハッシング
エ レプリケーション

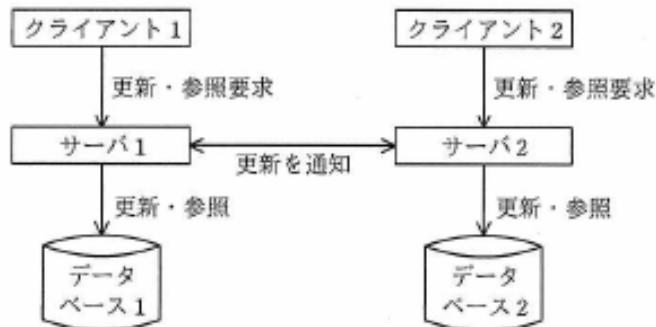
問2：正解（ウ）

分散型のデータベースに関する問題。シャーディングについて問われている。シャーディングとは、大量のデータを、複数のノードに分散して保存する技術のことである。問題文のように、ある規則に従ってシャードと呼ばれる小さなサブセット（これをチャunkという）に分割する。シャーディングのメリットは、応答時間の短縮や障害時の信頼性向上、拡張性に優れているなどがある。分散型ではあるものの、仮にノードの1台が故障しても他のノードは利用可能なため障害は局所的になる。また、データが増えてきた時にはスケールアウト（サーバの増設）で対応できる。

- ア：2相コミットプロトコル（→本書の第4章を参照）。分散型データベースの同期をとるアルゴリズムの一つ。
イ：コンシスティントハッシング（→用語集）。シャーディングのデータ分割アルゴリズムの一つ。
エ：レプリケーション（→用語集）。データベース間で同期をとる技術の一つ。

問1 図のデータベース1, 2は互いのデータの複製をもつ冗長構成である。クライアントからの更新・参照要求を受けたデータベースサーバ（以下、サーバという）は直下のデータベースを更新・参照し、他方のサーバにデータ更新を通知する。通知を受けたサーバは直下のデータベースに更新を反映する。

サーバ1, 2間のネットワークが分断し、データ更新を通知できなくなったとき、CAP定理で重視する特性（C, A, P）に対するサーバの挙動のうち、適切な組合せはどれか。

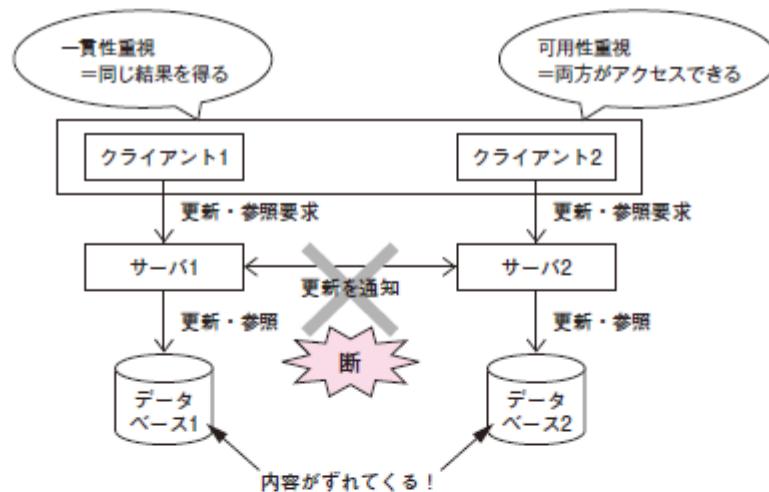


	CAP定理の特性C及びPを重視する場合	CAP定理の特性A及びPを重視する場合
ア	一方のサーバは停止し、もう一方のサーバは動作し続ける。	両方のサーバが停止する。
イ	一方のサーバは停止し、もう一方のサーバは動作し続ける。	両方のサーバが動作し続ける。
ウ	両方のサーバが動作し続ける。	一方のサーバは停止し、もう一方のサーバは動作し続ける。
エ	両方のサーバが動作し続ける。	両方のサーバが停止する。

問1：正解（イ）

分散型データベースにおける CAP 定理に関する問題。CAP 定理とは、次の 3 つの性質の頭文字をとって名付けられた理論である。Brewer 氏が発表した理論なので、ブリューワーの定理ともいわれている。このうち 3 つは同時に満たすことはできず、最大で 2 つまでしか満たすことしかできないとしている。

- ①一貫性（Consistency）：同じデータ（单一、かつ最新の確定データ）を得られること
- ②可用性（Availability）：単一障害点（そこで障害が発生したらシステム全体が停止してしまう機器等）が存在しないこと。どこかに障害が発生しても、常にデータにアクセスできること
- ③分断耐性（Partition-Tolerance）：複数のグループに分散保管し、そのグループ間に通信障害などが発生しても、システムを利用できること（全面停止しないこと）



問題文の図の下にある選択肢の表を見る限り、左側の「CAP 定理の特性 C 及び P を重視する場合」も、右側の「CAP 定理の特性 A 及び P を重視する場合」も、どちらにも“P”があるので、分断耐性は重視することになっている。ということは、上図のようにサーバ1とサーバ2間のネットワークが分断して両サーバで同期をとることが不可能になったとしても、システムを全面停止しないということになる。そして、そのような状態でも一貫性（C）か可用性（A）を確保して利用できなければならないというわけだ。

まず、左側の一貫性（C）を重視する場合、A（可用性）には目をつぶることができるので「一方のサーバは停止し、もう一方のサーバは動作し続ける」ようにできる。どちらかのクライアントはアクセスできなくなるが（可用性は損なわれるが）、その代わり片側のサーバを停止することによって一貫性は保持できる。

また、A（可用性）を重視する場合、一貫性（C）には目をつぶことができるので「両方のサーバが動作し続ける」ようにできる。ネットワークが復旧するまでは、クライアント1、クライアント2で、異なる結果を得ることになるが（一貫性は損なわれるが）、可用性は保持できる。

以上より、正解はイになる。

令和6年度・D B・午前Ⅱ

問17 スタースキーマでモデル化し、一定期間内に発生した取引などを分析対象データとして格納するテーブルはどれか。

ア ディメンションテーブル

イ デシジョンテーブル

ウ ハッシュテーブル

エ ファクトテーブル

問 17：正解（工）

データウェアハウスに関する問題。この説明は“分析対象データ”という記述から、ファクトテーブルの説明だと判断できる。ファクトテーブルは、データウェアハウスの分析のための最小単位のデータであり、多くの場合、業務系のトランザクションデータから作成される。時系列に蓄積されたデータで、“生データ”と呼ぶことも多い。このファクトテーブルの詳細度が情報分析の最も詳細な単位になる。

ア：ディメンジョンテーブルとは、分析軸のテーブルを意味する。ファクトテーブルを売上実績データとした場合だと、商品テーブル、店舗テーブル、年月テーブルなどがディメンジョンテーブルになる。

イ：デシジョンテーブルは、次のような決定表のこと。

エンティティタイプ“出動”の属性値の決定に関するデシジョンテーブル

月例 Y: 条件が真、N: 条件が偽、-: 条件が無関係

X：値が決まる。-：値が決まらない。

平成 17 年午後 II 問 1 より

ウ：ハッシュテーブルは、キーと対応する値のペア単位でデータを格納するテーブルのことである。ハッシュ関数の特性を利用していている。

問13 顧客、商品、注文、販売店という四つのテーブルをスタースキーマでモデル化した場合、ファクトテーブルとなるものはどれか。

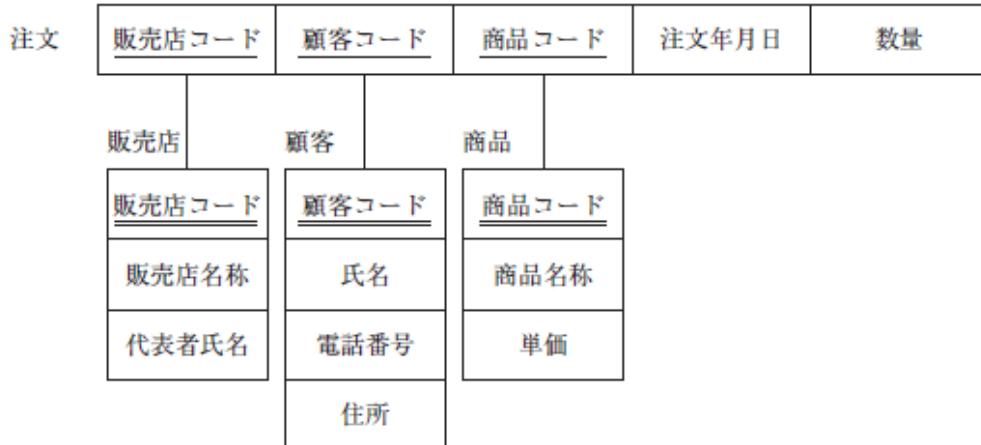
- ア 顧客（顧客コード、氏名、電話番号、住所）
- イ 商品（商品コード、商品名称、単価）
- ウ 注文（販売店コード、顧客コード、商品コード、注文年月日、数量）
- エ 販売店（販売店コード、販売店名称、代表者氏名）

問19 業務系のデータベースから抽出したデータをデータウェアハウスに格納するために、整合されたデータ属性やコード体系などに合うように変換及び修正を行う処理は何か。

- ア クラスタリング
- イ スライシング
- ウ ダイシング
- エ データクレンジング

問13：正解ウ

スタースキーマに関する問題。スタースキーマは、データウェアハウスで使われるスキーマで、分析対象の明細データのファクトテーブルを中心に分析軸（ディメンション）を周囲に配置する。選択肢ア～エのテーブル間の関連をチェックすると、中心に位置するファクトテーブルが明確になる。



この図を見れば一目瞭然、ファクトテーブルは注文テーブルになる。よって正解はウになる。

問19：正解（エ）

データウェアハウスに関する問題。整合されたデータ属性やコード体系に合うように変換及び修正を行う処理は、データクレンジング処理という。したがって正解は（エ）になる。

ア：クラスタリング（[参照](#) ➤ 用語集）

イ：スライシング（[参照](#) ➤ 用語集）

ウ：ダイシング（[参照](#) ➤ 用語集）

問17 OLAP によって、商品の販売状況分析を商品軸、販売チャネル軸、時間軸、顧客タイプ軸で行う。データ集計の観点を、商品、販売チャネルごとから、商品、顧客タイプごとに切り替える操作はどれか。

ア ダイス

イ データクレンジング

ウ ドリルダウン

エ ロールアップ

問13 販売データの分析において、売上額の見方を商品分類ごとから、曜日ごとや販売担当者ごとに変えて見る操作はどれか。

ア スライス

イ ダイス

ウ ドリルダウン

エ ロールアップ

問 17：正解（ア）

データウェアハウスに関する問題。OLAP (Online Analytical Processing) とは、オンライン分析処理のこと、多次元データベースを使って様々な角度（この問題だと、商品軸、販売チャネル軸、時間軸、顧客タイプ軸）から分析処理を行う処理やシステムのことを指している。

分析処理を行う時に、多くの集計項目から 2 つの項目を取り出して縦軸・横軸に指定し、二次元の表を作成するが（これを、“スライス”もしくは“スライシング”という）、この時に縦軸と横軸を入れ替えながら、すなわちデータ集計の観点を切り替えながら操作を進めていくことを“ダイス”もしくは“ダイシング”という。したがってこの説明は“ダイス”的説明になるため、アが正解になる。

イ：データクレンジング（参照 ➤ 用語集）。

ウ：ドリルダウン（参照 ➤ 用語集）。

エ：ロールアップ（参照 ➤ 用語集）。

問 13：正解イ

販売データの分析手法に関する問題。一般的に、この機能は「スライスアンドダイス機能」と説明される場合が多いので、ア、イで迷ったかもしれない。スライスアンドダイスを分けて説明する場合、スライスは「切り口」、ダイスが「（スライスしたものを）変える」ことなので、正解はイになる。

ア：スライス（参照 ➤ 用語集）

イ：ダイス（参照 ➤ 用語集）

ウ：ドリルダウン（参照 ➤ 用語集）

エ：ロールアップ（参照 ➤ 用語集）

問18 データウェアハウスのメタデータに関する記述のうち、データリネージュはどれか。

- ア 誰がどのデータを見てよいかを示す情報であり、適切なアクセス制御を目的として設定される。
- イ データが誰によって作られ管理されているかを示す情報であり、データ構造やデータ辞書を見ても意味が分からぬときの問合せ先を表す。
- ウ データがどこから発生し、どのような変換及び加工を経て、現在の形になったかを示す情報であり、データの生成源の特定又は障害時の影響調査に利用できる。
- エ データ構造がどのように定義されているかを示す情報であり、Web サイトなどに公開して検索できるようにする。

問 18：正解（ウ）

データウェアハウスに関する問題。メタデータのうちのデータリネージについて問われている。

一般的に「メタ」という用語は“一段階上位の視点や概念”を意味する。例えば、メタ認知というと、「自分が何を知っているのかを知っていること」、「自分の認知についての認知」という感じである。メタデータも同じで、「データについてのデータ（データを管理するためのデータ）」、すなわちどんなデータなのかを示す情報の集合体のことになる。データの属性（カラム名、型、更新日など）をデータとして持つ情報だ。

それを踏まえてメタデータの一つであるデータリネージについて説明する。データリネージとは、データがどこから来て、どのような処理を経て今の形になったのかを示す「データの流れ」を追跡するための情報であり、障害調査や信頼性確認に不可欠なメタデータになる。したがって、選択肢ウの説明になる。

なお、他の選択肢のいずれも、メタデータに関する説明になっている。

ア：メタデータの中でも「アクセス権限や利用者管理」に関する情報は、セキュリティに関するメタデータになる。セキュリティメタデータということもある。

イ：データの作成者や管理者は基本的なメタデータに含まれる。

エ：この説明は、構造化されたメタデータを一元的に公開・検索できるようにした仕組みのデータカタログの説明に近い。特に「Webサイトなどで公開・検索」という点が、典型的なデータカタログの利用イメージになる。

問18 OLAP (OnLine Analytical Processing) の操作に関する説明のうち、適切なものはどれか。

- ア 集計単位をより大きくする操作をロールアップという。
- イ 集計単位をより小さくする操作をスライスアンドダイスという。
- ウ 分析軸を入れ替えずにデータの切り口を変えることをダイシングという。
- エ 分析軸を入れ替えてデータの切り口を変えることをスライシングという。

問14 データマイニングの説明として、適切なものはどれか。

- ア 大量のデータを高速に検索するための並行的アクセス手法
- イ 大量のデータを統計的、数学的な手法で分析し、法則や因果関係を引き出す技術
- ウ 販売実績などの時系列データを大量に蓄積したデータベースの保存手法
- エ ユーザの利用目的に合わせて、部門別のデータベースを作成する技術

問13 データマイニングの説明として、適切なものはどれか。

- ア 大量のデータを分析し、単なる検索だけでは分からない隠れた規則や相関関係を見つけ出すこと
- イ データウェアハウスに格納されたデータの一部を、特定の用途や部門用に切り出して、データベースに格納すること
- ウ データ処理の対象となる情報を基に規定した、データの構造、意味及び操作の枠組みのこと
- エ 複数の属性項目をデータベースに格納し、異なる属性項目の組合せによる様々な分析を短時間で行うこと

問 18：正解ア

完全 NEW
参照 ➔ なし

OLAPに関する問題。OLAP(OnLine Analytical Processing)とはDWHシステムなどで行われる処理のことで、オンライン分析処理とか多次元分析処理などともいわれている。選択肢に出てくるような“ロールアップ”や“ドリルダウン”，“スライシング”，“ダイシング”（二つまとめて“スライスアンドダイス”ということもある）など、大容量のデータ分析に向く処理ができるのが特徴。適切な物を一つ選ぶ問題なので、選択肢を順番に見ていく。

- ア：正しい。集計単位をより大きくする操作をロールアップという。
イ：集計単位をより小さくする処理はドリルダウンという。誤り。
ウ：分析軸を入れ替えずにデータの切り口を変える処理はスライシングやスライスという。誤り。
エ：分析軸を入れ替えてデータの切り口を変える処理はダイシングやダイスという。誤り。

問 14：正解イ

データマイニング（[参照 ➔ 用語集](#)）に関する問題。選択肢は変化しているものの、データマイニングに関する問題は頻繁に出題されている。選択肢を順番に見ていく。

- ア：データを高速に検索する手法ではない
イ：正解
ウ：DWH（データウェアハウス、[参照 ➔ 用語集](#)）の説明
エ：データマート（[参照 ➔ 用語集](#)）の説明

問 13：正解ア

データマイニング（[参照 ➔ 用語集](#)）に関する問題。データマイニングツールとは、膨大な原始データから経営やマーケティングにとって必要な傾向、パターン、相関関係を導き出すためのツールである。選択肢のア以外は、いずれもデータウェアハウスに関連する説明である。

※データウェアハウス関連用語に、データマートやBIツールなども含めている

問18 データマイニングに関する説明として、適切なものはどれか。

- ア 基幹業務のデータベースとは別に作成され、更新処理をしない時系列データの分析を主目的とする。
- イ 個人別データ、部門別データ、サマリデータなど、分析者の目的別に切り出され、カスタマイズされたデータを分析する。
- ウ スライシング、ダイシング、ドリルダウンなどのインタラクティブな操作によって多次元分析を行い、意思決定を支援する。
- エ ニューラルネットワークや統計解析などの手法を使って、大量に蓄積されているデータから、顧客購買行動の法則などを探し出す。

問14 データマイニングツールに関する記述として、最も適切なものはどれか。

- ア 企業内で発生する情報を主題ごとに時系列で蓄積することによって、既存の情報システムだけでは得られない情報を提供する。
- イ 集計データを迅速かつ容易に表示するなど、利用者に対して様々な情報分析機能を提供する。
- ウ 大量に蓄積されたデータに対して統計処理などを行い、法則性の発見を支援する。
- エ 利用者が情報を利用するための目的別データベースであり、あらかじめ集計処理などを施しておくことによって検索時間を短縮する。

問18：正解（エ）

過去問題（平成17年度・DB午前 問13）

データマイニングとは、統計解析などの手法によって、大量に蓄積されているデータから、顧客購買行動などの法則性を見出すことである。よって、正解はエである。

ア：データウェアハウスの説明である。

イ：データマートの説明である。

ウ：意思決定支援システムで用いられる多次元解析のことである。

※選択肢ウは、いわゆるBIツールのこと。

問14：正解ウ

データマイニングツールとは、膨大な原始データから経営やマーケティングにとって必要な傾向、パターン、相関関係を導き出すためのツールである。選択肢のウ以外は、すべてデータウェアハウスに関連するものである。

※データウェアハウス関連用語に、データマートやBIツールなども含めている

その他

平成 20 年度・D B・午前

問29 データベースの論理モデルに関する記述のうち、適切なものはどれか。

- ア 階層モデルは、多対多のレコード関係を表現するのに適している。
- イ 関係モデルでは、子レコードはただ一つの親レコードに属する。
- ウ ネットワークモデルは、行と列からなる表で表現できる。
- エ ポイズ・コッド正規形は、関係モデルで使用される形式である。

平成 17 年度・D B・午前

問22 関係データベースにおける定義域に関する記述のうち、適切なものはどれか。

- ア 定義域が異なる属性同士の比較は、本質的には意味がない。
- イ 定義域は、単一の基本データ型又はユーザ定義型でなければならない。
- ウ 定義域は、ユーザ定義のスキーマである。
- エ 一つの属性は、複数の定義域上に定義できなければならない。

問 29：正解工

データベースの論理モデルに関する問題。選択肢を順番に見ていく。

ア：「多対多のレコード関係を表現するのに適している」のは、ネットワークモデルで、階層モデルでは表現できない。よって誤りである

イ：親子関係という概念は、階層モデルやネットワークモデルのものであり、RDBMS で親子関係を実装するのは可能だが、関係モデルには、本来、親子関係の概念はない。よって誤りである

ウ：「行と列からなる表」は関係モデルである。よって誤りである。

エ：正解である

参照 ➤ 「6.3 論理データモデル」

問 22：正解（ア）

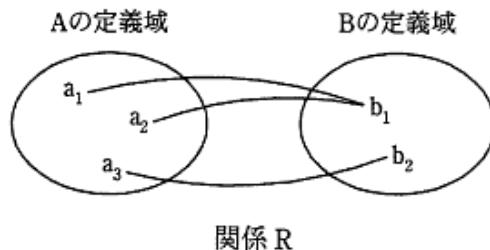
ア：定義域が異なる属性だと、“男性”と“30 歳”的ように全く比較できないか、比較できる属性どうしでもイコールにはならないわけだから、「本質的に意味が無い」と言わればそれは正しい。なのでこれが正解。

イ：単一じゃなくてもいい。CREATE DOMAIN で複雑な条件も定義可能

ウ：基本のデータ型もある。すべてがユーザ定義されているわけじゃない。

エ：複数の定義域上に定義できなくても何ら問題は無い。

問1 図のような関係 R (A, B)において、属性 A の定義域の要素は $\{a_1, a_2, a_3\}$ 、属性 B の定義域の要素は $\{b_1, b_2\}$ である。 a_1 と b_1 を結ぶ線は、 (a_1, b_1) のように、関係 R の要素を表している。この関係 R の要素を表す語として、適切なものはどれか。



ア 組

イ 属性

ウ ドメイン

エ 列

問10 関係データモデルにおいて属性 A, B を考える。属性 A のドメイン（定義域）は m 個の要素から成る集合であり、属性 B のドメインは n 個の要素から成る集合であるとする。このとき、関係 R を $R (A, B)$ とすると、R には最大何個のタプルがあるか。

ア $2^m 2^n$ イ $(m+n)^2$ ウ $m+n$ エ mn

問 1：正解ア

関係モデルに関する問題（参照 ➤ 第4章 3-(1) 関係モデル）。基礎用語（ドメイン、属性、組（タブル）、列）の意味が問われている。問題文で問われているものは、“関係 R の要素”であり、“(a₁, b₁) のよう”なものなので、選択肢アの組が正解になる。データベーススペシャリスト試験の受験者には、タブルというほうがわかりやすいかもしない。

問 10：正解工

過去問題（平成 17 年・DB 午前 問 30 など）
参照 ➤ 第4章 重要キーワード「3 論理データモデル」

関係データモデルに関する問題。ドメインやタブルなどの基本用語の意味が問われている。ドメインを属性集合、タブルを行もしくはインスタンスだと考えれば良い。それでもわかりにくければ、次のように、実際に簡単なデータを使ってシミュレーションしてみれば良い。

<具体例>

属性 A のドメインは 3 個 (= m)。1, 2, 3

属性 B のドメインは 4 個 (= n)。A, B, C, D

関係 R (A, B) … “最大のタブルの数”なので全組合せ。直積になる。全部で 12 個

(1, A), (1, B), (1, C), (1, D),

(2, A), (2, B), (2, C), (2, D),

(3, A), (3, B), (3, C), (3, D)

ア : $m = 3$, $n = 2$ で計算してみると, $8 \times 16 = 128$ 。誤り。

イ : 同じく 49 になる。誤り。

ウ : 同じく 7 になる。誤り。

エ : mn 。これが直積を表している。 $4 \times 3 = 12$ なので答えも合う。正しい。

※上記の選択肢のアの“n=2”は、“n=4”的誤り。他はあってる。

問45 一つの表に大量のデータを格納するとき、並列処理のために異なったディスクにデータを分割格納することがある。このような方式のうちキーレンジ分割方式に関する説明はどれか。

- ア 主キーと外部キーの参照関係を保持し、関数従属性に従って異なった表に分割格納する。
- イ データの発生した順に格納するディスクを変え、ディスクごとのデータ量が均等になるように分割格納する。
- ウ 分割に使用するキーの値にハッシュ関数を適用し、その値に割り当てられたディスクに分割格納する。
- エ 分割に使用するキーの値をあらかじめ決めておき、その値に割り当てられたディスクに分割格納する。

問13 ハッシュ方式によるデータ格納方法の説明はどれか。

- ア レコードの特定のデータ項目の値が論理的に関連したレコードを、同一ブロック又はできる限り隣接したブロックに格納する。
- イ レコードの特定のデータ項目の値に対応した子レコード同士を、ポインタで鎖状に連結して格納する。
- ウ レコードの特定のデータ項目の値の順序を保持して、中間ノードとリーフノードの平衡木構造のブロックを作り、リーフブロックにレコード格納位置へのポインタを格納する。
- エ レコードの特定のデータ項目の値を引数とした関数の結果に従って決められたレコード格納場所に格納する。

問45：正解工

キーレンジ分割方式とは、「分割に使用するキーの値をあらかじめ決めておき、その値に割り当てられたディスクに分割格納する」方式である。したがって、エが正解になる。

- ア：水平分割方式の説明。
- イ：格納順分割方式の説明。
- ウ：ハッシュ分割方式の説明。

問13：正解（工）

データ格納方法に関する問題。ハッシュ方式について問われている。ハッシュ方式は、レコードの特定のデータ項目の値からハッシュ値を算出し、そのハッシュ値をレコードの格納場所とする方式である。ハッシュ値の一意性（ハッシュ関数の衝突回避性）を利用している。したがって正解はエになる。

- ア：クラスタリングの説明
- イ：リスト方式の説明
- ウ：B+木方式の説明

問19 文献検索システム、データ検索システムなどの情報検索システムを評価する尺度

として用いられる再現率 (recall ratio) と精度 (precision ratio) の組合せとして、適切なものはどれか。ここで、a, b, c は次の件数を示す。

a : 薩積された全てのデータのうち、質問に適合する件数

b : 検索結果のデータのうち、質問に適合する件数

c : 検索結果のデータの件数

	再現率	精度
ア	$\frac{a}{b}$	$\frac{b}{c}$
イ	$\frac{b}{a}$	$\frac{b}{c}$
ウ	$\frac{b}{a}$	$\frac{c}{b}$
エ	$\frac{c}{b}$	$\frac{a}{b}$

問21 関係データベースとオブジェクト指向データベースを比較したとき、オブジェクト指向データベースの特徴として、適切なものはどれか。

ア 実世界の情報をモデル化したクラス階層を表現でき、このクラス階層を使うことによって、データと操作を分離して扱うことができる。

イ データと手続がカプセル化され一体として扱われる所以、構造的に複雑で、動作を含む対象を扱うことができる。

ウ データの操作とリレーションが数学的に定義されており、プログラム言語とデータ操作言語との独立性を保つことができる。

エ リレーションが論理的なデータ構造として定義されており、非手続的な操作言語でデータ操作を行うことができる。

問 19：正解（イ）

情報検索システムの評価尺度として用いられる再現率と精度に関する問題。問題文の a, b, c について例を挙げて説明すると次のようになる。

- ①蓄積されているすべてのデータ = 1万件 (a の一部)
- ②(本当なら) 質問に適合する件数 = 100 件 (a)
- ③検索されたデータの件数 (c) = 90 件
- ④検索されたデータのうち、質問に適合する件数 80 件 (b)
- ⑤検索されたデータのうち、質問に適合しない件数 10 件 (c - b)

ここで、再現率とは、情報検索（この問題では質問）を行った場合に、蓄積されたすべてのデータのうち、その質問に本来適合しているはずのデータ件数のうち、どれだけ実際の検索でヒットしたかを表す率である。今回のケースだと、本来適合するはずのデータ件数は 100 件 (a) で、実際に検索されたデータ件数のうち、質問に適合していたのは 80 件 (b) なので、再現率は 80 % になる (b / a)。この段階で、選択肢イもしくはウが正しい。

一方、精度とは、検索結果がどれだけ正しかったのかを表す尺度（字のままの意味）になる。その計算式は、実際に検索されたデータ件数 (c) のうち、正しく検索されたデータ (b) の割合なので、今回の精度は約 89% ($80 \div 90$) になる (b / c)。したがって選択肢アと選択肢イが正しい。

以上より、両方とも正しいのは選択肢イになる。

問 21：正解イ

オブジェクト指向データベース（参照用語集）の特徴に関する頻出問題。選択肢を順番にチェックし消去法も使いながら解答を導く。

ア：クラスの階層は表現できるが、データと操作（手続）をカプセル化して一体的に扱うのがオブジェクト指向データベースの特徴。よって不正解

イ：正解

ウ：関係データベースの特徴である

エ：関係データベースの特徴である

問41 ストアドプロシージャの利点はどれか。

- ア 応用プログラムからネットワークを介して DBMS にアクセスする場合、両者間の通信量を減少させる。
- イ 応用プログラムからの一連の要求を一括して処理することによって、DBMS 内のアクセスパスの数を減少させる。
- ウ 応用プログラムからの一連の要求を一括して処理することによって、DBMS 内の必要バッファ数を減少させる。
- エ データが格納されているディスク装置への I/O 回数を減少させる。

令和 6 年度・D B・午前Ⅱ

問2 クライアントサーバシステムにおけるストアドプロシージャに関する記述のうち、
適切でないものはどれか。

- ア 機密性が高いデータに対する処理を特定のプロシージャ呼出しに限定することによって、セキュリティを向上させることができる。
- イ システム全体に共通な処理をプロシージャとして格納しておくことによって、処理の標準化を行うことができる。
- ウ データベースへのアクセスを細かい単位でプロシージャ化することによって、処理性能（スループット）を向上させることができる。
- エ 複数の SQL 文から成る手続を 1 回のプロシージャ呼出しで実行することによって、クライアントとサーバの間の通信回数を減らすことができる。

問41：正解ア

ストアドプロシージャ（参照 ➤ 用語集）の説明はアになる。“通信料を減少させる”というキーワードがポイント。その他の選択肢にあるような効果は得られない。

問2：正解（ウ）

ストアドプロシージャ（参照 ➤ 用語集）に関する問題。ストアドプロシージャとは、サーバ側のデータベースに格納（ストア）しておく一連の処理（プロシージャ）を記述したモジュールのこと。ストアドプロシージャを用いることで、クライアントサーバ間の通信量を減らすことができる。また、データベースによっては、事前コンパイル・最適化されるため、実行速度が向上する場合がある。適切でないものが問われているので、選択肢を順番に見ていく。

ア：ストアドプロシージャを使用すれば、ユーザーが直接SQLを組み立てて実行することを制限し、予め正しく作成された処理を、当該プロシージャを通してのみデータを操作することになる。これにより、不正アクセスやSQLインジェクションのリスクを低減できるため、セキュリティ向上につながる。適切である。

イ：複数のアプリケーションやモジュールで共通して利用される処理をプロシージャ化することで、当該処理の標準化を行うことができる。適切である。

ウ：ストアドプロシージャを細かく分割しそうすると、本来1回で良い呼び出し回数でも、細かく分割すると複数回になる。その場合、プロシージャの呼び出し回数が増えるため、ネットワークのオーバーヘッドやコンテキスト切り替えの負担も増える。それゆえ、かえって処理性能が低下する可能性がある。したがって適切ではないため、これが正解になる。

エ：選択肢ウの逆で、クライアントとサーバ間のネットワーク通信の回数を減らすことになる。この記述は適切である。

問38 TP モニタの配下で動作するアプリケーションプログラムからデータベースをアクセスするときの説明として、適切なものはどれか。

- ア SQL の検索結果が“対象データなし”であったときは、DBMS がロールバックをする。
- イ TP モニタへのコミットとロールバックの指示はアプリケーションプログラムが行い、DBMS への指示は TP モニタが行う。
- ウ アプリケーションプログラムは、TP モニタ配下のほかのアプリケーションプログラムのコミット、又はロールバックの発行を確認して、自分のトランザクションをコミットするかロールバックするか判断する。
- エ トランザクションの開始をアプリケーションプログラム側から指示することができず、TP モニタが開始時点を決定する。

問17 W3C で勧告されている、Indexed Database API に関する記述として、適切なものはどれか。

- ア Java のアプリケーションプログラムからデータベースにアクセスするための標準的な API が定義されている。
- イ SQL 文をホストプログラムに埋め込むための API が定義されている。
- ウ Web ブラウザ用のストレージの機能として、トランザクション処理の API が定義されている。
- エ データベースに対する一連の手続きを DBMS に格納し、呼び出す API が定義されている。

問38：正解イ

TPモニタ(Transaction Processing Monitor)とは、オンライントランザクション処理を実現するためのソフトウェアで、トランザクション(一つの処理単位)実行におけるデータの整合性を保証する。TPモニタを利用する場合、アプリケーションがTPモニタに対してコミットとロールバックの指示を行い、DBMSに対しては、TPモニタが指示を出す。よって正解はイとなる。

問17：正解（ウ）

データベース応用分野の問題。Indexed Database APIについて問われている。Indexed Database APIは、Webブラウザからデータベースを利用する時の標準APIになる。W3Cにより標準化仕様が策定され、多くのブラウザに組み込まれている。Webブラウザが単にデータをローカルに保存できるだけではなく、トランザクション処理も可能になる。したがって、正解はウになる。

ア：Java → DB の API は、JDBC (Java Database Connectivity) になる。

イ：ホストプログラム→DBで使うSQL文は、一般的に埋め込みSQLと呼ばれている。

エ：データベースに対する一連の手続きをDBMSに格納して利用する機能はストアドプロシージャになる。

問16 ビッグデータの処理に使用される CEP（複合イベント処理）に関する記述として、適切なものはどれか。

- ア 多次元データベースを構築することによって、集計及び分析を行う方式である。
- イ データ更新時に更新前のデータを保持することによって、同時実行制御を行う方式である。
- ウ 分散データベースシステムにおけるトランザクションを実現する方式である。
- エ 連続して発生するデータに対し、あらかじめ規定した条件に合致する場合に実行される処理を実装する方式である。

問19 ビッグデータの処理に使用される CEP（複合イベント処理）に関する記述として、適切なものはどれか。

- ア ストリームデータをメモリ上に展開し、あらかじめ設定した条件に合致した場合に対応するアクションを実行する。
- イ ビジネスの結果を表す数値データをファクトテーブル、そのデータの解析に必要なデータを次元テーブルに格納して処理する。
- ウ 分散データベースにおいて、関係する全てのプロセスでコミットが可能かを判定する相と、各プロセスで実際のコミットを行う相の二つを経てコミット処理を実行する。
- エ 並列分散環境にある多数のサーバによって、分散ファイルシステムに蓄積された大量データをパッチ処理する。

問 16：正解（工）

ビッグデータに関する問題。CEP (Complex Event Processing: 複合イベント処理) について問われている。

CEP とは、大量のビッグデータを “リアルタイム” に処理するために、有用なデータのみを選別して処理する技術の総称である。具体的には、“ストリームデータ” をあらかじめ設定した条件（処理条件や分析シナリオなど）に合致した場合に対応するアクションを実行する。したがって正解はエになる。データを一旦ディスクに蓄積し、それを分析するストック型に対して、フロー型やストリーム型と言われることもある。金融のアルゴリズム取引、クレジットカードの不正利用の検知、ウイルスや不正アクセスの検知などで “リアルタイム性” が求められる処理で使われている。

- ア : DWH (ストック型) で利用される方式
- イ : RDBMS で利用される方式
- ウ : 分散データベース (ストック型) 処理の説明

問 19：正解（ア）

ビッグデータに関する問題。CEP (Complex Event Processing: 複合イベント処理) について問われている。

CEP とは、大量のビッグデータを “リアルタイム” に処理するために、有用なデータのみを選別して処理する技術の総称である。具体的には、“ストリームデータ” を “メモリ上” に展開し、あらかじめ設定した条件（処理条件や分析シナリオなど）に合致した場合に対応するアクションを実行する。データをいったんディスクに蓄積し、それを分析するストック型に対して、フロー型やストリーム型と言われることもある。したがって正解はアになる。金融のアルゴリズム取引、クレジットカードの不正利用の検知、ウイルスや不正アクセスの検知などで “リアルタイム性” が求められる処理で使われている。

- イ : DWH (ストック型) の説明
- ウ : 2 相コミット処理の説明
- エ : 分散処理技術に関する説明。問題文の中に蓄積という表現があることからストック型だと判断できる。

問18 ビッグデータ処理基盤に利用され、オープンソースソフトウェアの一つであるApache Spark の特徴はどれか。

- ア MapReduce の考え方に基づいたバッチ処理に特化している。
- イ RDD (Resilient Distributed Dataset) と呼ばれるデータ集合に対して変換を行う。
- ウ パブリッシュ／サブスクライブ (Publish/Subscribe) 型のメッセージングモデルを採用している。
- エ マスターノードをもたないキーバリューストアである。

問18 データレイクの特徴はどれか。

- ア 大量のデータを分析し、単なる検索だけでは分からない隠れた規則や相関関係を見つけ出す。
- イ データウェアハウスに格納されたデータから特定の用途に必要なデータだけを取り出し、構築する。
- ウ データウェアハウスやデータマートからデータを取り出し、多次元分析を行う。
- エ 必要に応じて加工するために、データを発生したままの形で格納して蓄積する。

問 18：正解（イ）

データベース応用分野の問題。ビッグデータ処理基盤の Apache Spark について問われている。Apache は、代表的な Web サーバの OSS（オープンソースソフトウェア）である。開発管理しているのは Apache Software Foundation で、そこでは（この問題で問われている）Apache Spark などの開発も進められている。

そのうちのひとつ Apache Spark は、ビッグデータを処理するためのフレームワークになる。メモリを効果的に利用して分散処理を行うことで高速処理を実現するが、それを可能にしているのが、データを抽象的に扱う RDD（Resilient Distributed Dataset）と呼ばれるデータ構造の仕組みになる。したがって、正解はイになる。

ア：「MapReduce の考え方に基づいた…」というのは Apache Hadoop になる。

ウ：「Publish/Subscribe 型のメッセージングモデル」は、MQTT（Message Queuing Telemetry Transport）の説明になる。

エ：「マスタノードをもたない」、「（分散型）KVS」は Apache Cassandra の説明になる

問 18：正解（エ）

ビッグデータに関する問題。データレイクの特徴について問われている。データレイクとは、様々なところ（Web サイトや SNS など）から収集した様々なデータを、構造化されているデータも非構造化データも「そのままの形で」まとめて保存する一元化されたりポジトリのことである。いったんデータレイクに貯めておき、ビッグデータ分析や機械学習などで利用するときに、必要な形で取り出して使用する。したがって、選択肢エが正解になる。

ア：データマイニングの説明。

イ：データマートの説明。

ウ：OLAP（Online Analytical Processing）の説明。

問17 機械学習を用いたビッグデータ分析において使用される Jupyter Lab の説明はどれか。

- ア 定期的に実行するタスクを制御するための、ワークフローを管理するツールである。
- イ データ分析を行う際に使用する、対話型の開発環境である。
- ウ 並列分散処理を行うバッチシステムである。
- エ マスターノードをもたない分散データベースシステムである。

問 17：正解（イ）

ビッグデータ分析に関する問題。Jupyter Labについて問われている。Jupyter Labは、ブラウザ上で動作する対話型の統合開発環境（IDE）である。開発に必要な様々な機能がオープンソースとして提供されている。今や、データ分析には欠かせないツールになっている。したがって、選択肢イが正解になる。

ア：ジョブ管理システムのジョブスケジューラの説明。

ウ：ジョブ管理システムの説明。

エ：P2P型分散データベースの説明。

問18 インターネット販売などの巨大な取引データを高速に処理するために、NoSQL と呼ばれるデータ処理方式が使われることがある。その特性のうち結果整合性（Eventual Consistency）の説明として、適切なものはどれか。

- ア データを複製し、複数サイトに分散して保持するとき、コンシスティントハッシング手法によって、時間帯別に格納先を固定する。
- イ 複数のクライアントからの更新要求が衝突する場合、ロック機構によってどちらかを待たせることで整合性を保つ。
- ウ 分散した複製サイト間で更新内容を厳密に同期させずに、同期の一時的な遅れを許容する。
- エ 分散した複製サイト間で更新内容を整合させるために、2 相ロック方式を採用する。

問18：正解（ウ）

NoSQLは、RDBMSの持つ限界を打破するために登場した概念になる。問題文にあるように、SNSサイトやインターネット販売など“大量のデータ”を高速処理するために考え出されたもので、広義には“RDBMS (+SQL) 以外のデータベース”としても使用される言葉である。今は「Not only SQL」と定義されている。代表的なものに、key – valueストアやドキュメント指向データベースがある。

また、NoSQLでは、RDBMSのACID特性に対して、次に示すような“BASE”という概念を用いている。いずれも、しっかりとACID特性に対して、ゆるい一貫性になっているのが特徴である。

BASE の内訳	おおよその意味
Basically Available	原則、可用性を保持する
Soft-state	柔らかい状態
Eventual Consistency	結果的に整合性を保持する

問題文で問われている“結果整合性(Eventual Consistency)”は、BASEの一つで、複数サイト間で更新内容を厳密には同期させずに、同期の一時的な遅れを許容するという特徴を持つ。更新内容は、(即座に反映されずに多少のタイムラグはあっても)結果的に整合性が取れていればいいだろうという緩い一貫性を示している。したがって正解はウになる。

ア：コンシスティントハッシング手法（参照 ➡ 用語集）の説明

イ：ロック機構を持っているのは既存のRDBMSになる。

エ：2相ロック方式も既存のRDBMSになる。

問1 BASE 特性を満たし、次の特徴をもつ NoSQL データベースシステムに関する記述のうち、適切なものはどれか。

[NoSQL データベースシステムの特徴]

- ・ネットワーク上に分散した複数のノードから構成される。
- ・一つのノードでデータを更新した後、他の全てのノードにその更新を反映する。

ア クライアントからの更新要求を2相コミットによって全てのノードに反映する。

イ データの更新結果は、システムに障害がなければ、いつかは全てのノードに反映される。

ウ 同一の主キーの値による同時の参照要求に対し、全てのノードは同じ結果を返す。

エ ノード間のネットワークが分断されると、クライアントからの処理要求を受け付けなくなる。

問16 NoSQL のうち、データ構造はキーに対して一つの値をもつデータモデルであり、データ間は疎結合で分散して配置しやすい特徴をもつものはどれか。

ア キーバリューストア

ウ 文書データストア

イ グラフデータベース

エ ワイドカラムデータストア

問 1：正解（イ）

NoSQL データベースに関する問題。BASE 特性とは、DBMS に持たせる特性の一つ。RDBMS の ACID 特性に対する概念として用いられることが多く、ゆるい一貫性（常に一貫性を保っている必要はなく、一時的には整合性の無い状態になっていても、結果的に整合性が保てる）になっているのが特徴である。この特性によって、高可用性とスケーラビリティを維持しながら、大量の非構造化データを扱うことができる。

BASE の内訳	およそその意味
Basically Available	原則、可用性を保持する
Soft-state	柔らかい状態（常に厳密な状態、すなわち整合性や一貫性を保っている必要はない）
Eventual Consistency	結果的に整合性を保持する

ア：2 相コミットは、すべてのデータベースの完全な一貫性を保証するもので、原則、RDBMS の特徴になる。

イ：「いつかは全てのノードに反映される」のが BASE 特性であり、NoSQL データベースに関する記述になる。これが正解。

ウ：この説明も完全な一貫性の説明。BASE 特性を満たす NoSQL データベースの場合、タイミングによっては異なる結果になる。

エ：この説明も完全な一貫性を確保するための機能。

問 16：正解（ア）

NoSQL に関する問題。NoSQL に分類されるデータベースの特徴について問われている。問題文中の「キーに対して一つの値を持つ」、「データ間は疎結合」という特徴があるのは、キーバリューストア（KVS：Key-Value Store）（型データベース）の説明になる。構造が非常にシンプルなので、大量データの取扱いに向いている。したがって正解はア。

イ：グラフデータベースも NoSQL に分類されるデータベースの一つ。“ノード”，“リレーションシップ”，“プロパティ”の 3 要素によってノード間の関係性を表現する。

ウ：NoSQL に分類されるデータベースの一つ。ドキュメント型ともいう。1 件分のデータを“ドキュメント”と呼び、個々のドキュメントのデータ構造は自由であって、データを追加する都度変えることができる。KVS のバリューの部分が“ドキュメント”になっているイメージ。

エ：NoSQL に分類されるデータベースの一つ。1 件分のデータ部分（KVS のバリューの部分相当）が、任意の数の列と値になっている構造。

memo

著者：三好康之

株式会社エムズネット代表。大阪を主要拠点に活動する IT コンサルタント。本業の傍ら、大手SIベンダの SE に対して、資格取得講座や階層教育を担当している。高度区分において脅威の合格率を誇る。保有資格は、情報処理技術者試験全区分制覇（累計 42 個、内高度系 29 個）をはじめ、IT コーディネータ、中小企業診断士、技術士（経営工学）、販売士 1 級 など多数。JAPAN MENSA 会員。代表的な著書に、『勝ち残りSEの分岐点』、『IT エンジニアのための【業務知識】がわかる本』、『情報処理教科書プロジェクトマネージャ』（以上翔泳社）他多数。

また、「自分らしい働き方」を追求し続け現在に至る。具体的には、プライベートの充実を再優先にしながら、生活の絶対的安定を確保する働き方を模索している。趣味は、ひとりでも多くの IT エンジニアに、楽に、コストをかけずに“資格＝武器”を持ってもらうための研究。

HP : <https://msnet.jp>

mail : miyoshi@msnet.jp

facebook（個人）: <https://www.facebook.com/miyoshijp>

Blog : アメーバ認定 公式ブログ（ビジネス部門） 絶賛公開中！
自分らしい働き方 <https://ameblo.jp/yasuyukimiyoshi/>
※読者登録をしていただくと喜びます。

YouTube : <https://www.youtube.com/user/msnetmiyomiyo>

情報処理技術者試験対策のコンテンツを公開中

※読者登録をしていただくと喜びます。