

# Chapter 11 データベース

## 11-1 DBMSと関係データベース〔解答・解説〕

問 1 ア

〔解説〕関係Zは、レコードのうち学部コードがBである2つを抽出し、さらに"氏名"、"学部コード"、"学部名"の3つの属性を取り出した結果である。表から選択した行(レコード/タプル)を取り出す操作を「選択」、表から選択した列(属性)を取り出す操作を「射影」といい、関係Zを得るために行われた関係演算はこの2つになる。

問 2 ウ

問 3 イ

〔解説〕データベースのインデックスはデータベースのレコードを効率よくアクセスするための索引であり、目的のデータをすばやく探し出すことができる。

問 4 ウ

〔解説〕ア キーを指定することで挿入・検索・削除が効率的に行える木構造の一種。  
イ 全文検索を行う対象となる文書群から単語の位置情報を格納するための索引構造  
エ 検索キーごとにビットマップを用意し、ビットマップ上のビットの有無によりデータの格納位置を管理するインデックス。

問 5 エ

〔解説〕ア ダンプファイルは、データベース移行のときなどに用いられるデータベースの内容をそのまま書き出したファイル  
イ チェックポイントファイルは、コミットされたトランザクションをデータベースに書き出すときに、その情報を記録したファイル  
ウ バックアップファイルは、媒体障害の発生に備えてデータベースのコピーを保存したファイル

問 6 ウ

〔解説〕再編成とは、データベースへのデータの追加や削除を繰り返すことが原因で、格納効率やアクセス効率が低下した場合、データベースの性能修復を行うことである。

問 7 ア

〔解説〕イ データベースダンプは、データベースの内容をファイルに書き出して保存すること  
ウ バックアップは、データの複製を作成し障害時に元の状態に復旧できるように備えること  
エ ロールバックは、トランザクションの途中、プログラムのバグなどでアプリケーションが強制終了した場合に、更新前ログを用いてデータベースをトランザクション開始直前の状態に戻す処理

問 8 エ

問 9 エ

- 〔解説〕ア CLUSTERING(クラスタリング)は、複数のコンピュータを結合し、葡萄の房(クラスタ)のようにひとまとまりとしてふるまわせる技術。SQL の 1 つである CLUSTER 句は表をクラスタ構成にする文。
- イ DISTINCT は、関係演算において重複する行を取り除くことを指定する字句。
- ウ NOT NULL は、列の値として空値(NULL)を認めない制約。

問 10 エ

- 〔解説〕ア 階層型データベースの説明
- イ ネットワーク型データベースの説明
- ウ オブジェクト指向データベースの説明

問 11 エ

- 〔解説〕DBMS はクエリ（問い合わせ）を受け付けると、  
パーサーによる構文解析→クエリオプティマイザによるクエリ最適化→コード生成  
の順で処理される。

問 12 ア

- 〔解説〕イ ビューに対してビューを定義することができる
- ウ 基底表（元になる表）を定義してからビューの定義を行う
- エ 一つの基底表に対して複数のビューを定義することができる

問 13 イ

- 〔解説〕ア 2つの表が共通して持つ属性(列)を基準に結合を行い、新しい表をつくりだす操作
- ウ 表の中から特定の行を取り出す操作
- エ 2つの表のいずれかに属する行を合わせて、新しい表をつくりだす操作

問 14 ア

- 〔解説〕オプティマイザとは、SQL 文の処理手順の煩雑さや優先度などを考慮して、最も効率の良いデータベースへのアクセス経路を選択し、実行プランを決定する RDBMS の機能である。

問 15 エ

- 〔解説〕ア 企業内に散らばっているデータの有効活用を目的に、過去から現在までの基幹系データベースや外部データベースを統合し、効果的な戦略的意思決定を支援するためのデータベース
- イ データベースやシステム開発においてデータの属性・性質などデータに関する情報の定義を保存するための辞書
- ウ データの流れに着目して、その対象となる業務のデータの流れと処理の関係をわかりやすく図式化する構造化分析の手法

問 16 イ

- 〔解説〕ア 3次元であるサイコロを振って見えている面が変わるように、縦横の項目を変えて多次元データのまったく違う面を表にする機能
- ウ 詳細に展開する操作（例 :四半期集計データから月集計データに移る）
- エ 集計レベルを高くする操作を指します。（例 :集計単位を1カ月→1年）別名：ドリルアップ

問 17 ウ

- 〔解説〕 ア 企業の様々な活動を介して得られた大量のデータを目的別に整理・統合して蓄積したもの  
イ データの属性とそのデータ項目を使っているファイルとの関係を管理するもの  
エ データに関するデータのこと（音声ならアーティスト名、アルバム名、ジャンルなど）

問 18 イ

- 〔解説〕 データマイニングとは企業に大量に蓄積されるデータを分析し、その中に潜む因果関係などを探し出す技術。「ビールを買う客は一緒に紙オムツを買うことが多い」「雨の日は肉の売上が良い」など。

問 19 イ

- 〔解説〕 データディクショナリとは、データの意味、他のデータとの関係、期限、用途、フォーマットなどのデータに関する情報を管理したものである。

問 20 エ

- 〔解説〕 候補キー：表のある行を一意に特定することができる属性、または属性の組合せで主キーの候補となるキー。1つの表に複数の候補キーがある場合には、その中で項目の組合せが最小でその表の性質にあったものを主キーとする。
- ア 主キーの属性では値が null であることは許されないが、候補キーでは null 値が許される。候補キーに NOT NULL 制約を加えたものが主キー制約となる。
- イ インデックスの説明です。候補キーとインデックスは関係ない。
- ウ 外部キーの説明。

問 21 ア

- 〔解説〕 イ 射影は、表の中から特定の列を取り出す操作。
- ウ 選択は、表の中から条件に合致した行を取り出す操作。
- エ 挿入は、表に対して特定の行を差し込むする操作。表への列の挿入は SQL の ADD 句で行われるため操作名としては「追加」が適当。

問 22 ウ

- 〔解説〕 スキーマ(schema)とは、データの内容、データの論理構造、記憶形式や編成などデータベースの構造を記述したもの。これを決める DBMS の機能は「定義機能」が適切。
- ア 機密保護機能は、ユーザ認証、アクセス権限、利用ログの記録、暗号化などによってデータを安全に保つ機能。
- イ 障害回復機能は、ロールバック、ロールフォワード、チェックポイント、更新前／更新後ログの記録などによってデータベースを障害から発生する機能。
- エ 保全機能は、排他制御や参照制約、表制約などによってデータの完全性を保つ機能。

## 11-2 主キーと外部キー（解答・解説）

問 1 ウ

- 〔解説〕 問題文中の表定義における参照制約とは、"在庫"表のレコードがもつ製品番号を主キーとする"製品"表のレコードが存在していなければならないということ。これは"製品"表に存在しない製品を、"在庫"表で指定することはできないことを意味している。

問 2 ウ

- 〔解説〕 ア PRIMARY KEY区やFOREIGN KEY句を用いて指定する  
 イ CHECK句を用いて指定する  
 エ 整合性制約の一種でCREATE ASSERTION句で指定

問 3 エ

- 〔解説〕 ア 商品番号、数量、金額が一意に決まらないので主キーとして不適切。  
 イ 顧客番号は、注文番号に関数従属しているので不適切。  
 ウ 顧客番号は、注文番号に関数従属しているので不適切。

問 4 ア

- 〔解説〕 イ 削除しようとするレコードの商品コードが、注文表に使用されていないか参照される  
 ウ 変更しようとするレコードの商品コードが、注文表に使用されていないか参照される  
 エ 追加しようとするレコードの商品コード及び顧客コードが商品表及び顧客表にあるか参照される

問 5 ウ

- 〔解説〕 ア、イ、エでは、顧客ごとの年間購入金額は求まらない。

問 6 エ

- 〔解説〕 プログラマは社員であるため、1つ以上の"プログラム言語"に加えて"社員番号"と"氏名"の属性を記録しなくてはならない。したがって"社員番号"、"氏名"、"プログラム言語"の属性をもつプログラマ表は次のようになる。

プログラマ表

社員番号	氏名	プログラム言語	プログラム言語	プログラム言語
0001	情報太郎	C	Java	PHP
0002	東京次郎	C		

この状態では、"プログラム言語"が繰り返し項目になっているため第1正規化を行って重複項目を排除する。

プログラマ表

社員番号	氏名	プログラム言語
0001	情報太郎	C
0001	情報太郎	Java
0001	情報太郎	PHP
0002	東京次郎	C

この状態で1つのレコードを特定するには"社員番号"と"プログラム言語"が必要なのでこの2つを組み合わせた複合キーが主キーになる。

プログラマ表

社員番号	氏名	プログラム言語
0001	情報太郎	C
0001	情報太郎	Java
0001	情報太郎	PHP
0002	東京次郎	C

さらに"氏名"は"社員番号"に関数従属しているので、"氏名"列を別表に分離する第2正規化を行う。このように順を追って完成した適切なデータベース設計が「エ」の表である。

問 7 エ

- 〔解説〕 ア "銀行"表と"口座"表への対応関係は「1対多」  
イ "口座"表中に"銀行"表への参照制約を課した外部キーがある  
ウ "口座"表と"顧客"表への対応関係は「多対1」

問 8 イ

- 〔解説〕 外部キーをもたない建物 → 建物番号だけを外部キーとする部屋 → 建物番号と部屋番号を外部キーとする資産の順に入力するのが適切である。

問 9 ア

- 〔解説〕 外部キー制約は、関係データベースの2つのテーブルの間の参照整合性を保つための制約で、SQLでは設定する属性に対して「FOREIGN KEY 句」を指定する。  
正規化された表同士を外部キーで結合することでデータの一貫性・正当性を保つことができる。

問 10 ア

- 〔解説〕 イ 配送日「2016-8-30」の部品IDは「2346」と「1342」の2種類ある。よって、配送日は部品IDを関数的に決定していない  
ウ 部署ID「500」の部品IDは「2346」と「1342」の2種類ある。よって、部署IDは部品IDを関数的に決定していない  
エ 部署名「研究開発部」の配送先は「名古屋工場」と「川崎事業所」の2種類ある。よって、部署名は配送先を関数的に決定していない

問 11 エ

- 〔解説〕 ア 受注商品と顧客が関連つけられている記録があるのは当月分だけで、前月～3カ月前についてはそれぞれの受注合計額のしかデータがないので出力できない  
イ 商品マスタファイルは顧客別には集計されていないので出力できない  
ウ 商品マスタファイルに受注した商品と担当者を関連付ける項目(担当者コード)がないので担当者別に出力することはできない

## 11-3 正規化〔解答・解説〕

問 1 イ

- 〔解説〕 データの正規化とは、データの冗長性を排除して一貫性と整合性を図る手法である。

問 2 ウ

- 〔解説〕 ア 第1正規形の目的である  
イ 正規化を進めるほど。整合性制約は増加する  
エ 正規化とロック待ちは関係ない

問 3 ウ

- 〔解説〕 属性Aの値が決まれば属性Bの値が一意に決まるとき、BはAに関数従属であるという。属性Dは属性Aに対して、100→東京都、200→大阪府、300→北海道、400→東京都のように関数従属している。

#### 問 4 エ

〔解説〕 a が決まれば b, c, d, e が決まるのだから, 表1 {a, b, c, d, e} が定義できる。  
b が決まれば f, g が決まるのだから, 表2 {b, f, g} が定義できる。  
b, c が決まれば h が決まるのだから, 表3 {b, c, h} が定義できる。  
※下線部は主キー

#### 問 5 ウ

〔解説〕 注文数量は, 注文番号と商品番号から一意に決まり, 商品名は商品番号から一意に決まる。

#### 問 6 イ

〔解説〕 第1正規化では繰返し項目をなくするためにそれらを複数のレコードに分割する。

1つの紙に複数の商品が記載された伝票をイメージすると、

(伝票番号 1234, ..., 商品コード 111, ...)

(伝票番号 1234, ..., 商品コード 222, ...)

というように、同じ伝票番号を持つレコードが複数存在することになるので、表中のレコードを一意に特定するためには「伝票番号」+「商品コード」の複合主キーにしなければならない。

↓

第2正規化では主キーの一部によって一意に決まる項目を別表に移す。表を見ると「伝票番号」から「日付」「顧客コード」「顧客名」「住所」の4つ、「商品コード」から「単位」「単価」の2つが一意に定まることがわかるのでこれらの関係を別表に分離する。

↓

第3正規化では主キー以外の項目によって一意に決まる項目を別表に移す。主キー以外の項目をみると、「顧客コード」から「顧客名」「住所」が一意に定める事がわかる為この関係を別表に分離する。

[第2正規化後]

伝票番号	日付	顧客コード	顧客名	住所
伝票番号	商品コード	数量		
商品コード	単位	単価		

→

[第3正規化後]

伝票番号	日付	顧客コード		
顧客コード	顧客名	住所		
伝票番号	商品コード	数量		
商品コード	単位	単価		

## 11-4 SQLでデータベースを操作する〔解答・解説〕

#### 問 1 ウ

〔解説〕 列を取り出しているので、射影である。

#### 問 2 ウ

〔解説〕 R表からA1, A2, A3の列を取り出すのは射影, A4 = 'a'の行を取り出すのは選択である。

#### 問 3 ア

〔解説〕 次の操作が行われている場合、ビューは更新できない

- ・ GROUP BY句が使用されている
- ・ DISTINCT句が使用されている
- ・ 結合が行われている

問 4 ア

〔解説〕 副問合せで職務がプログラマである社員の人数をカウントし、それが5人未満の部署コードをとりだせばよい。  
 ※エは、その部署にプログラマである社員が1人もいない場合、抽出できない。

問 5 ウ

〔解説〕 L I K E 句は、特定の文字列を含む項目を選択する場合に用いる。「%」は、ワイルドカードとよび、「%の部分に任意の長さの任意の文字が存在してもよい」ことを表す。  
 「氏名 L I K E '%三%」と指定すると名前の中に「三」があるデータを取り出すことができる。

問 6 イ

〔解説〕 ア 商品番号が"N P 2 0 0"は4個ありレコード数は2なので、平均値として2が得られる  
 イ "出庫記録"は4行からなっているので、4が得られる  
 ウ "出庫記録"のレコードの中で"数量"列の最大値である3が得られる  
 エ 日付が"2 0 1 5 - 1 0 - 1 1"であるレコードは3,4行目なので、2つの行の"数量"の値を合計した3が得られる

問 7 エ

〔解説〕 ア 販売金額の平均が4 0 0 0万円以上、又はいずれかの期の販売金額が3 0 0 0万円以上となる  
 イ 販売金額の合計が4 0 0 0万円以上、かつすべての期の販売金額が3 0 0 0万円以上となる  
 ウ いずれかの期の販売金額が4 0 0 0万円より大きい、かついずれかの期の販売金額が3 0 0 0万円以上となる

問 8 ア

〔解説〕 A表よりS\_\_NENRE Iが23であるレコードであることがわかるが、BUSHOとSHAINを結合させるための条件が必要となる。

問 9 エ

〔解説〕 表Bは、SQL の SELECT コマンドによって、表Aから「部署コード、社員コード、名前」を取り出した表である。表Bは、「部署コード」の順番で同じ部署コードの場合は、社員コードの順番に表示されているので、「ORDER BY 部署コード, 社員コード」を指定する。

問 10 エ

問 11 ア

〔解説〕 ビュー表は次のようになり、アの処理を行うと、商品コードS 0 0 1の売値－仕入値が3 0,0 0 0となり、現れなくなる。

収益商品

商品コード	品名	型式	売値	仕入値
S 0 0 1	T	T 2 0 0 3	1 5 0,0 0 0	1 0 0,0 0 0
S 0 0 5	R	R 2 0 0 3	1 4 0,0 0 0	8 0,0 0 0

問 12 ア

〔解説〕 イ R EXCEPT S は、R表に存在し、S表に存在しない行から成る表(差集合)を返す演算  
 ウ R UNION S は、2つの表の合わせた集合から重複を除去した表(和集合)で返す演算  
 エ R INTERSECT S は、R表とS表に共通する行から成る表(共通集合)を返す演算

問 13 イ

〔解説〕最初に、どの SQL 文にも共通している 2 つの SELECT 文から得られる中間表を考える。  
それぞれ以下の結果となる。

SELECT 部門コード, '第 1 期' AS 期, 第 1 期売上 AS 売上  
FROM 部門別売上

部門コード	期	売上
D01	第 1 期	1,000
D02	第 1 期	2,000
D03	第 1 期	3,000

SELECT 部門コード, '第 2 期' AS 期, 第 2 期売上 AS 売上  
FROM 部門別売上

部門コード	期	売上
D01	第 2 期	4,000
D02	第 2 期	5,000
D03	第 2 期	8,000

2 つの中間表を(ア)INTERSECT (共通)、(イ)UNION (和)、(ウ)CROSS JOIN (直積)、  
(エ)INNER JOIN (内部結合)を行うとそれぞれ以下ようになる

- ア INTERSECT (共通) は 2 つの関係に共通集合を得る演算で、共通する行はないので"結果なし"となる
- イ UNION (和) は 2 つの関係の和集合を得る演算で、1 つ目の関係に 2 つ目の関係が足される感じになるので、設問の問合せ結果と同じになる。
- ウ CROSS JOIN (直積) は、2 つの関係に存在する行のすべての組み合わせを得る演算で、どちらの関係も 3 行ずつあるので、結果表は「 $3 \times 3 = 9$  行」で構成される表となるので誤り。
- エ INNER JOIN (内部結合) は、2 つの関係を共通する属性で結び付ける演算で、結合相手が存在する行だけが残る。2 つの関係を部門コードで結合すると、結果表は以下になるので誤り。

問 14 ア

- 〔解説〕イ スキーマは、データベースの構造などを定義したもの
- ウ テーブルは、属性と組みでデータを格納する表のこと
- エ ビューは、実表の一部、または複数の実表から関係演算(選択、射影、結合など)によって得られた結果を 1 つの表に見せかけた仮想の表



問 15 ウ

〔解説〕 GROUP BY 句でグループ化した後にグループを絞るには HAVING 句を使用する。本問では、学生ごとに全科目の平均を算出したいので、「GROUP BY 学生番号」で学生番号ごとにグルーピングした後、点数の平均が 80 以上の学生を「HAVING AVG(点数) >= 80」で選択することになる。

ア 科目ごとにグループ化しているので、科目ごとの平均点を集計してしまう

イ、エ GROUP BY 句の後に WHERE 句を記述しているので構文エラー

## 11-5 トランザクション管理と排他制御〔解答・解説〕

問 1 ウ

問 2 エ

問 3 イ

- 〔解説〕
- ア ジョブ 3 が起動したとき、ジョブ 1 が資源 A を共有ロックしている状態。ジョブ 3 は資源 A に対して占有ロックをかけたいのでジョブ 1 が資源 A のロック解除するのを待つことになる
  - ウ ジョブ 3 が起動したとき、ジョブ 2 が資源 A を共有ロックしている状態。ジョブ 3 は資源 A に対して占有ロックをかけたいのでジョブ 2 が資源 A のロック解除するのを待つことになる。さらにジョブ 2 も資源 B に対して占有ロックをかけている状態なので、ロック解除を待つ必要がある
  - エ ジョブ 1 が起動したとき、ジョブ 3 が資源 A を占有ロックしている状態。資源 A に共有ロックをかけるにはジョブ 3 がロック解除するのを待つ必要がある

問 4 ア

〔解説〕セマフォとは、共有資源に対する排他制御を行うフラグ変数である。

問 5 イ

- 〔解説〕
- ア 共有ロックが掛けられているのは行 a なので、それとは別の行 b に対する専有ロックは獲得可能
  - イ 正しい。共有ロックが掛けられている行 a は表 A の一部なので、表 A 全体に対する専有ロックは獲得できない
  - ウ 共有ロックが掛けられている資源に対して、他のトランザクションが獲得可能なのは共有ロックのみ
  - エ A 表には専有ロックが掛けられているので、A 表の一部である行 a に対するロックは獲得できない

問 6 イ

- 〔解説〕
- ア 一度コミットした内容は失われない（持続性）
  - ウ データベースの整合性は保たれる（一貫性）
  - エ 更新途中で障害が発生した場合、更新前の状態に戻される（原子性）

問 7 ア

- 〔解説〕
- イ 独立性の説明
  - ウ 原子性説明
  - エ 永続性説明

## 問 8 イ

- 〔解説〕ア トランザクション3が起動したとき、トランザクション1が資源Aを共有ロックしている状態。トランザクション3は資源Aに対して専有ロックをかけたいのでトランザクション1が資源Aのロックを解除するのを待つことになる。
- ウ トランザクション3が起動したとき、トランザクション2が資源Aを共有ロックしている状態。トランザクション3は資源Aに対して専有ロックをかけたいので、トランザクション2が資源Aのロックを解除するのを待つことになる。さらにトランザクション2も資源Bに対して専有ロックをかけている状態なので、こちらのロック解除も待つ必要がある。
- エ トランザクション1が起動したとき、トランザクション3が資源Aを専有ロックしている状態。資源Aに共有ロックをかけるにはトランザクション3がロック解除するのを待つ必要がある。

## 問 9 ア

- 〔解説〕クライアントから命令を送信するだけで処理が実行できるので、ネットワークの通信量を削減できる。

## 問 10 ウ

- 〔解説〕ロールフォワード：前進復帰と訳され、データベースシステムに障害が起こったとき、更新後ログを使用して今まで処理したトランザクションを再現しシステム障害直前までデータベース情報を復帰させる処理

システム障害からの復帰後はチェックポイントの状態から処理が再開される。再開直後はチェックポイント以降にコミットされたトランザクションがデータベースに反映されていないため、まず前進復帰でデータベースへの反映を行う。したがって前進復帰の対象は「最後のチェックポイントからシステム障害発生直前の間」にコミットされたトランザクション「T4とT5」になる。

## 問 11 エ

- 〔解説〕デッドロック：共有資源を使用する2つ以上のプロセスが、互いに相手プロセスの必要とする資源を排他的に使用していて、互いのプロセスが相手が使用している資源の解放を待っている状態に陥ってしまうことをいう

デッドロックを考慮して流れを見て行く

1：③でトランザクションAがテーブルaをロックする

2：④でトランザクションBがテーブルbをロックする

3：⑤でトランザクションAがテーブルbをロックしようとするが、既にトランザクションBにロックされているのでトランザクションAは待ち状態となる

4：⑥でトランザクションBがテーブルaをロックしようとするが、既にトランザクションAにロックされているのでトランザクションBは待ち状態となる。

この状態ではトランザクションAはテーブルbのロック解放を待ち、トランザクションBはテーブルaのロック解放を待っているので、双方のトランザクションの進行がストップしてしまう。したがってデッドロックが成立するのは⑥の時点。

## 問 12 エ

問 13 イ

〔解説〕 プロセス A を基準として他のプロセスを見て行く

[プロセス B]

占有順序がプロセス A と同じなのでデッドロックは発生しない。

プロセス A が資源 X, Y, Z を使用している間、プロセス B は資源の解放を待ち、プロセス A の実行終了後に処理を開始する。

[プロセス C]

占有順序がプロセス A と異なるので、以下の順序の場合にデッドロックが発生する。

- ① プロセス A が資源 X を占有
- ② プロセス C が資源 Z を占有
- ③ プロセス A が資源 Y を占有
- ④ プロセス C は資源 X の解放待ち&プロセス A は資源 Z の解放待ち
- ⑤ デッドロックの発生

[プロセス D]

占有順序がプロセス A と異なるので、以下の順序の場合にデッドロックが発生します。

- ① プロセス A が資源 X を占有
- ② プロセス D が資源 Z を占有
- ③ プロセス A が資源 Y を占有
- ④ プロセス D は資源 Y の解放待ち&プロセス A は資源 Z の解放待ち
- ⑤ デッドロックの発生

したがって、プロセス A とデッドロックが発生する可能性のあるプロセスは「C, D」。

問 14 ア

〔解説〕 イ 正規化によって実現される性質

ウ UNIQUE 制約によって実現される性質

エ 3 層スキーマモデルによって実現される性質

問 15 ウ

〔解説〕 [a について]

表単位のロックと行単位のロックを比較しているので、ロックの競合が起こりやすいのは、より広い範囲をロックする表単位

[b について]

メモリ使用領域が多く必要になるのは、トランザクションの同時実行数が多い行単位

問 16 イ

〔解説〕 ア アボート処理が完了するとアボート済状態に遷移する。

イ 正しい。コミット処理中からアボート処理中に遷移することはあるが、その逆はない。コミットは全ての処理が正常に終了したときの処理である。

ウ コミット処理中に不具合が起こるとアボート処理中に遷移する。

エ コミット処理が完了するとコミット済状態に遷移する。

問 17 ウ

〔解説〕 デッドロック：共有資源を使用する 2 つ以上のトランザクションが、互いに相手トランザクションが必要とする資源を排他的に使用していて、互いのトランザクションが相手を使用して  
いる資源の解放を待っている状態

## 11-6 データベースの障害管理（解答・解説）

問 1 エ

問 2 エ

〔解説〕アはチェックポイントファイル，イはミラーリング，ウはバックアップファイルの説明である。

問 3 エ

問 4 エ

〔解説〕ア トランザクション障害からの回復方法

イ システム障害のうち、障害発生直前のチェックポイント後にコミットされたトランザクションに対する回復方法

ウ システム障害のうち、障害発生直前のチェックポイント後にコミットされていないトランザクションに対する回復方法

問 5 イ

〔解説〕ア システム障害に備えてトランザクション処理の途中に設定するのはチェックポイントである

ウ デッドロック発生時は、通常、両方のトランザクションをキャンセルし、ロールバックする

エ ジャーナルファイルの更新前情報を使用してロールバック処理を行う

問 6 ア

〔解説〕イ ロールフォワードの説明である

ウ 最新のバックアップファイルと更新後ログが使用される

エ ロールバックの説明である

問 7 エ

〔解説〕ロールバックは、トランザクションの途中、プログラムのバグなどで異常終了した場合に、更新前ログを用いてデータベースをトランザクション開始直前の状態に戻す処理。