

ジュニア・アナリスト

模擬試験 簡易解説

内容

解答一覧	1
問 1.....	2
問 2.....	3
問 3.....	4
問 4	5
問 5.....	6
問 6.....	7
問 7.....	8
問 8.....	9
問 9.....	10
問 10.....	11
問 11.....	12
問 12.....	13
問 13.....	14
問 14.....	15
問 15.....	16
問 16.....	17
問 17.....	18
問 18.....	19
問 19.....	20
問 20.....	21
問 21.....	22
問 22.....	23
問 23.....	24
問 24.....	25
問 25.....	26
問 26.....	27

解答一覽

問 1 :	1.3
問 2 :	2.1
問 3 :	3.1
問 4 :	4.2
問 5 :	5.3
問 6 :	6.2
問 7 :	7.3
問 8 :	8.1
問 9 :	9.3
問 10 :	10.1
問 11 :	11.1
問 12 :	12.3
問 13 :	13.3
問 14 :	14.3
問 15 :	15.3
問 16 :	16.1
問 17 :	17.4
問 18 :	18.4
問 19 :	真
問 20 :	20.3
問 21 :	21.3
問 22 :	22.2
問 23 :	23.2
問 24 :	24.4
問 25 :	25.3
問 26 :	26.2

問 1

トピック：トランザクション設計(強い実体を持つ 2 者間の) 1-N の関係

1.1 : 誤答

理由： 実装したい前提と逆の実装。

これは**品種 (PetBreed)** が **1 つのペット (Pet)** に属し、数多くの品種がペットに属する実装となります。

1.2 : 誤答

理由： この実装ではペット (Pet) が**複数の品種 (PetBreed)** に属し、1 つの品種に複数のペットが属する N-N の関係が実装されています。

1.3 : 正答

理由： 実装したい前提通りの内容。

ペット (Pet) と **品種 (PetBreed)** がトランザクションとして定義され、特定の品種が登録されていない状態でもペットを登録することができ、**強い実体を持つ 2 つのトランザクション間の 1-N の関係**です。

1.4 : 誤答

理由： 「1.3」が正答であるため。

問 2

トピック：トランザクション設計 N-N の関係

2.1： 正答

理由： 実装したい前提通りの内容。

ペット (Pet) と食べ物 (Food) がトランザクションとして定義され、食べ物には複数のペットを紐づけられるようにレベルが追加され、ペットの外部参照が実装されている。
これにより 1 種類の食べ物を複数のペットが食べる可能性を満たす。

2.2： 誤答

理由： 弱い実態における 1-N の関係が実装されている。

この実装では、1 つの食べ物に複数のペットを紐づけられるが、ペット (Pet) を管理するトランザクションの定義が無いため、各食べ物で記録されたペットが同一のペットであることを保証できない。また、ペットは食べ物を登録するときのみ登録ができ、単体で存在することができない。

2.3： 誤答

理由： 強い実態を持つ 2 つのトランザクション間の 1-N の関係を実装。

そのため、ペット (Pet) は 1 つの食べ物 (Food) に属し、数多くのペットが同じ食べ物に属する可能性がある実装。

2.4： 誤答

理由： 「2.1」が正答であるため。

問 3

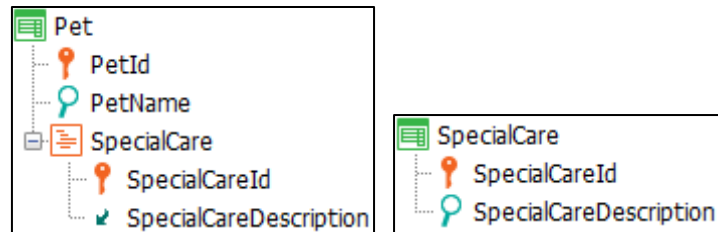
トピック：トランザクション設計（弱い実体間の）1-N の関係

3.1： 正答

理由： 示されたトランザクション設計を適切に説明している。
この場合、ケアガイドライン（SpecialCare）はペット（Pet）が登録されていなければ登録することはできず、他のペットと同じ内容を登録していても、それは同一のデータではない。

3.2： 誤答

理由： この説明は下図のような N-N の関係を説明している。



3.3： 誤答

理由： GeneXus における適切な説明ではありません。
外部参照が可能なトランザクションがない場合でもレベルを定義することができます。

3.4： 誤答

理由： 「3.1」が正答であるため。

問 4

トピック：候補キー

4.1： 誤答

理由： 外部参照キーの [Null 許容] を Yes に変更した場合、
医療記録 (PetMedicalRecord) にペット (Pet) を指定せずに
登録を行うことができるようになるが、前提として記載された
挙動は実装できない。

4.2： 正答

理由： 医療記録 (PetMedicalRecord) の PetId に対し一意のインデックス
(Unique Index) を作成した場合、医療記録において 1 つの PetId は
1 つの医療記録にしか指定することができなくなります。
そのため、前提における「ペットには医療記録が 1 つだけ」と
「医療記録はそのペットだけのもの」を満たします。

4.3： 誤答

理由： 医療記録 (PetMedicalRecord) を PetMedicalRecordId と PetId が
複合キーとなるように定義していますが、前提を満たしません。
この場合、1 つの PetMedicalRecordId に対し、複数の PetId を
指定することができ、かつ、同じ PetId の値を異なる PetMedicalRecordId
で指定することができる状況となります。

4.4： 誤答

理由： 「4.2」が正答であるため。

問 5

トピック：正規化

5.1： 誤答

理由： 次の 2 つの理由により正しい構造ではありません。

- ・ ServicePetsQty はグローバル式のため、物理項目は生成されない
- ・ Service トランザクションの第 2 レベル Pet に対するテーブルがない

5.2： 誤答

理由： 次の理由により正しい構造ではありません。

- ・ ServicePetsQty はグローバル式のため、物理項目は生成されない

5.3： 正答

理由： GeneXus が正規化し、生成する正しい構造です。

5.4： 誤答

理由： 「5.3」が正答であるため。

問 6

トピック：トランザクション設計からのベーステーブルと拡張テーブル

6.1： 誤答

理由： 直接 N-1 の関係となるテーブルのみのため。

6.2： 正答

理由： ベーステーブルとなる SERVICEPET テーブルから直接または間接的に N-1 の関係となるすべてのテーブルが記載されているため。

6.3： 誤答

理由： 間接的に N-1 の関係となるテーブルはすべて拡張テーブルとなるため、COUNTRY の記述も必要なため。

6.4： 誤答

理由： ベーステーブル自身の記述のみとなり、拡張テーブルの記載がないため

問 7

トピック：参照整合性とインデックス

7.1： 誤答

理由： [Null 許容] を Yes としても GeneXus は参照整合性チェックを維持するため。

7.2： 誤答

理由： 実装を「一意のインデックス」に変更した場合、品種（PetBreedId）の入力は必須となる。また、ペット（Pet）において同じ品種を複数のデータで登録できなくなるため。

7.3： 正答

理由： [Null 許容] を Yes として設定した場合、値を指定しない場合には参照整合性チェックが行われず（必須ではなくなる）、値を指定した場合には参照整合性チェックを行うため、適切な実装である。

7.4： 誤答

理由： 「7.3」が正答であるため。

問 8

トピック：インデックス（主キー、外部キー、ユーザー、テンポラリー - キー候補）

8.1： 正答

理由： **SERVICEPET** は Service トランザクションの**第 2 レベル**に基づき生成されるテーブルであり、**ServiceId** と **PetId** の**複合主キー**を持つ。このテーブルに対し、GeneXus が生成する正しいインデックスとして**主キーインデックス**となる IServicePet、**外部キーインデックス**となる IServicePet1 および IServicePet2 が定義されているため。

8.2： 誤答

理由： このインデックスとなるケースは**外部参照キーが無い**トランザクションで、ServiceId と PetId による**複合主キー**が定義されている場合のみとなるため、提示されたトランザクション設計に基づくものでない。

8.3： 誤答

理由： このインデックスとなるケースは**1 項目の主キー**で定義されたトランザクションで、**外部参照キーが 1 つの場合のみ**となるため、提示されたトランザクション設計に基づくものでない。

8.4： 誤答

理由： 「8.1」が正答であるため。

問 9

トピック：参照整合性とインデックス

9.1： 誤答

理由： GeneXus のトランザクションでは、参照整合性チェックが行われます。
そのため、制御が一切行われずに削除されることはないため。

9.2： 誤答

理由： GeneXus は参照整合性違反となるようなデータ操作は行わない。
しかし、ユーザーが意図しない削除は行われず、このように
あるデータを削除する際に外部参照を行っているデータを
削除することはないため

9.3： 正答

理由： 記載の通りの挙動が GeneXus における参照整合性チェックであるため。

9.4： 誤答

理由： 「9.3」が正答であるため

問 10

トピック：ルールをトリガーするタイミング: AfterInsert

10.1： 正答

理由： ルール内にヘッダー（Service（第1レベル））の項目属性しかないため、トリガーイベント AfterInsert は記載の通りの実行タイミングとなる。
ただし、この場合第2レベルのデータが格納されていないため、グローバル式の値は算出されていなく、ルールの条件が正しく評価されないため前提の要件を満たさない。

10.2： 誤答

理由： 記載された実行タイミングとする場合、トリガーイベントを「on **AfterLevel** Level ServicePetDate」と記述する必要がある。
また、このトリガーイベントを定義した場合、要件を満たす実装である。

10.3： 誤答

理由： 記載された実行タイミングとする場合、トリガーイベント **BeforeInsert** を利用する必要があるため。

10.4： 誤答

理由： トリガーイベントが定義された場合、クライアントでトリガーされない。

問 11

トピック：ルールをトリガーするタイミング

11.1： 正答

理由： 第 1 レベルの保存後となる **AfterInsert** が指定されたルールが実行され、第 2 レベルのすべての行の操作完了後となる **AfterLevel** が指定されたルールが実行され、すべての操作が完了し、**Commit 実行後**に **AfterComplete** が指定されたルールが実行されるため。

11.2： 誤答

理由： **AfterLevel** が指定されたルールは第 2 レベルのすべての行の操作が完了した後に実行されるが、「a)」に記載されたルールは**第 1 レベル**に対する **AfterInsert** となり、この順で実行されることはないため。

11.3： 誤答

理由： **AfterComplete** はトリガーイベントにおけるもっとも遅いタイミングのトリガーとなり、このトリガーイベントより後に実行できるルールはないため。

11.4： 誤答

理由： 一部の例外を除き、ルールのトリガー順序は宣言順ではなく、**GeneXus の判断**により決定されるため。

問 12

トピック：グローバル式

12.1： 誤答

理由： グローバル式は**拡張テーブル**に含まれるすべての項目属性を
利用できるため。

12.2： 誤答

理由： グローバル式における **IF 構造**は **Events** と異なり **ELSE** は
利用できないため。

12.3： 正答

理由： 提示された要件を満たし、**条件を満たした場合とそれ以外の場合で**
意図した値を計算するため。

12.4： 誤答

理由： 「12.3」が正答であるため。

問 13

トピック：インライン式

13.1： 誤答

理由： 登録経過年数を正しく計算できているが、PrintBlock1 を出力する処理が **For Each 構文外**にあるため、**1つの動物（Pet）** だけ出力される実装のため。

13.2： 誤答

理由： 登録経過年数の計算が **For Each 構文外**にあるため、PetAddedDate の**値が取得できず**、正しい実装でないため。

13.3： 正答

理由： 登録経過年数の計算が **For Each 構文内**にあり、意図した値の計算が行え、PrintBlock1 の出力も **For Each 構文内**にあるので、参照した動物（Pet）とともに出力し、**すべての動物**が出力できるため。

13.4： 誤答

理由： 「13.3」が正答であるため。

問 14

トピック：1 つのエンティティに対する複数の参照

14.1： 誤答

理由： 獣医のデータが Vet と VetAssistant の 2 つのトランザクションに分かれてしまうため、同一の獣医を同じデータとして扱えないこと、および一人の獣医を登録する際に **2 度登録**しなければならないため、適切な実装ではありません。

14.2： 誤答

理由： サブタイプグループの**定義が不適切**です。
サブタイプグループを定義する場合、別名参照するためのデータの**主キー項目およびそれに属する項目属性を 1 つのサブタイプグループで定義する必要があるため。**

14.3： 正答

理由： Vet に登録された獣医を**担当と代診の 2 種類の異なる意味合い**でペット（Pet）に**紐づけられる実装のため。**

14.4： 誤答

理由： 「14.3」が正答であるため。

問 15

トピック：1 つのエンティティに対する複数の参照

15.1： 誤答

理由： GeneXus において項目属性が**配置される順序**によって記録可能なデータが**異なることはない**ため。

15.2： 誤答

理由： PetCountry トランザクションを**定義する必要はない**ため。
この実装の場合、ペット（Pet）に紐づける国（Country）は、一度 **PetCountry** でデータとして登録しなければならないため、Country での登録と別に国の登録操作が必要となる。

15.3： 正答

理由： 適切な実装である。
PetCountry **サブタイプグループ**を定義し、サブタイプ項目属性を**直接ペット（Pet）の項目属性として定義**することで、国（Country）で登録されたデータをそのまま利用できる。

15.4： 誤答

理由： 「15.3」が正答であるため。

問 16

トピック：ネストされた For Each に関連したロジック – デカルト積

16.1： 正答

理由： ナビゲーションリストに表示された内容は、**Country** がベーステーブルの For Each があり、その中に **Service** がベーステーブルのネストされた For Each があることが読み取れる。

この場合、この 2 つのベーステーブルには 1-N の関係はないため、**デカルト積**の実装となるため。

16.2： 誤答

理由： **コントロールブレイク**の実装として必要な「**同じベーステーブル**」、「**Order によるグループ項目の指定**」の条件を満たしていないため。

16.3： 誤答

理由： Country と Service には 1-N の関係がないため、**結合**とはならない。

16.4： 誤答

理由： 「16.1」が正答であるため。

問 17

トピック：For Each および暗示的なフィルタ

17.1： 誤答

理由： 1 つ目の For Each のベーステーブルが「**Pet.Service**」となっているため、
同じ **Pet** に関する情報が第 2 レベルの **Service** 登録数分繰り返し表示
されるため、適切ではない。

17.2： 誤答

理由： **PetBreed** をベーステーブルとしたネストされた 2 つ目の For Each が
不適切である。PetBreed は **Pet** の拡張テーブルにあたるため、
1 つの For Each として定義する必要はない。

17.3： 誤答

理由： **PrintBlock2** に配置された **ServiceName**、**ServicePetPrice** は、
Pet の拡張テーブルに含まれない項目属性のため、適切ではない。

17.4： 正答

理由： **Parm** ルールで項目属性を利用した場合、受け取った値は暗示的な
フィルタとして利用されるため、パラメーターで受け取った特定の
Pet に関する情報を表示できる。
また、ネストされた For Each で **Pet.Service** の表示を定義しているため
適切な実装である。

問 18

トピック：ネストされた For Each に関連したロジック – Join

18.1： 誤答

理由： For Each がネストされていないため、すべての品種 (PetBreed) のリストが表示された後、すべてのペット (Pet) のリストが表示され、前提を満たす実装となっていない。

18.2： 誤答

理由： 1 つ目の For Each のベーステーブルが「Pet」となっているため、ペット (Pet) ごとの品種が表示され、続けてネストされた For Each でペットの名前を表示する処理がペットのデータ分だけ繰り返され、各品種に属するペットのリストは出力されない。

18.3： 誤答

理由： 1 つの For Each で実装しているため、ペット (Pet) ごとの出力となり、各品種に属するペットのリストは出力されない。

18.4： 正答

理由： 1 つ目の For Each のベーステーブルが「PetBreed」となっているため、登録されたペットがない場合もすべての品種が表示され、品種ごとに該当するペット (Pet) のリストが表示される。

問 19

トピック：Where 節を持つ For Each

正答： 真

理由： 表示対象となる品種（PetBreedId）が「4」と「7」つまり
ペット（Pet）において4「**か**」7のデータを対象とするため、
Where 節で「**Or**」を利用している。
また、上記条件を満たし「**かつ**」登録された年が2020を対象と
するため、**並行する Where 節**を記述することで「**And**」結合となり、
要件を満たした正しい実装のため。

問 20

トピック：ネストされた For Each に関連したロジック – コントロールブレイク

20.1： 誤答

理由： **PrintBlock2** に配置された **PetName** は、PetBreed の
拡張テーブルに含まれない項目属性のため、適切ではない。

20.2： 誤答

理由： 1 つ目の For Each のベーステーブルが「**PetBreed**」となっているため、
登録されたペットがいない場合もすべての品種が表示され、品種ごとに
該当するペット (Pet) のリストが表示される。

20.3： 正答

理由： すべてのペット (Pet) を品種 (PetBreed) ごとにグループ化し、
ペットが登録されている品種のみを出力するためには、
コントロールブレイクの実装が必要となる。
コントロールブレイクの実装として必要な「同じベーステーブル」、
「Order によるグループ項目の指定」の条件を満たしてるため。

20.4： 誤答

理由： 同じベーステーブルを持つネストされた For Each が定義されているが、
Order 節がないため、適切なグループ化が行われない。

問 21

トピック：When none 節を持つ For Each コマンド

21.1： 誤答

理由： For Each の構文で Where 節の条件に該当するレコードが無かった場合の**処理分岐**に「else」は**利用できない**ため。
これは If 構文における条件分岐時のみ利用できるものである。

21.2： 誤答

理由： For Each の構文において **Unique 節**は、**重複したデータを表示しない項目属性を指定**するために利用するものであり、Where 節の条件に該当するレコードがなかった場合の**処理分岐**に**利用できない**。

21.3： 正答

理由： For Each の構文で Where 節の条件に**該当するレコードが無かった**場合の**処理分岐**を行う場合、**When none 節**を利用する。

21.4： 誤答

理由： 「21.3」が正答であるため。

問 22

トピック：データプロバイダー：ベーステーブルを持つグループ

22.1： 誤答

理由： データプロバイダーでコレクションをロードする場合、From 節でベーステーブルを指定する場合、この定義において **SDTCountryItem** に **From 節**を指定することは**不適切**である。
また、**ベーステーブルが Pet** の場合、前提を満たせない。

22.2： 正答

理由： From 節の定義が正しく、データプロバイダーのプロパティも正しい設定が行われている。
ペットの数を計算するインライン式も Country と Pet が**間接的に 1-N の関係**となるため、問題なく前提を満たす実装である。

22.3： 誤答

理由： データプロバイダーのベーステーブルが **PetBreed** のため、すべての国を表示できない可能性がある。
また、データプロバイダーの「Collection」プロパティが **True** に設定されているため、「コレクション型 SDT」のコレクションを出力する定義となり、不適切な実装となっている。

22.4： 誤答

理由： 「22.2」が正答であるため。

問 23

トピック：ビジネスコンポーネントとしてトランザクションで使用するレコードの挿入

23.1： 誤答

理由： 入力日（PetAddedDate）は Default ルールが定義されているため、挿入された場合、その日の日付になる。

23.2： 正答

理由： ビジネスコンポーネントはトランザクションの画面を利用する場合と同様に参照整合性の制御を行い、定義されているルールも実行される。

23.3： 誤答

理由： ビジネスコンポーネントの場合、参照整合性の制御は行われる。
また、入力日（PetAddedDate）の値もルールに基づき値が入力される。

23.4： 誤答

理由： ビジネスコンポーネントの場合、参照整合性の制御が行われる。

問 24

トピック：ベーステーブル付きの Web パネルにおけるフィルタ

24.1： 誤答

理由： **Grid** の定義にベーステーブルがあるため、Load イベントで **For Each** および **Load コマンド** を記述する必要はない。
また、Grid に変数は配置されていないため、代入は不要である。

24.2： 誤答

理由： **Grid** の定義にベーステーブルがあるため、Load イベントで **For Each** および **Load コマンド** を記述する必要はない。

24.3： 誤答

理由： **Start イベント** は画面表示時の初期処理として **1 度しか実行されない**。
また、**Events** 内で項目属性に値の代入は行えない。

24.4： 正答

理由： ベーステーブルのある **Grid** であるため、フィルタ用の変数の値をグリッドの **Conditions プロパティ** で条件として定義することで要件を満たすことができる。

問 25

トピック：ベーステーブル付きの Web パネルにおける Load イベント

25.1： 誤答

理由： **Grid** の定義にベーステーブルがあるため、Load イベントで **For Each** および **Load コマンド** を記述する必要はない。

25.2： 誤答

理由： ペットの数を利用した条件分岐に **For Each** の **Where** 節で実装しているが、**For Each** を利用し、条件分岐の処理を記述することは適切ではない。

25.3： 正答

理由： **Grid** の定義にベーステーブルがあるため、Load イベントは品種（PetBreed）のデータ 1 件ごとに 1 回実行される。
そのため、**Count** 式で各品種のペットの数を取得することができ、**If** 文による条件分岐で表示するコメントを決定できている。

25.4： 誤答

理由： 「25.3」が正答であるため。

問 26

トピック：ベーステーブルなしの Web パネルにおける Load イベント

26.1： 誤答

理由： For Each のベーステーブルがサービス (Service) のため、サービスに関連付けられたペット (SERVICEPET テーブル) は拡張テーブルとならないため、適切なデータを取得できない。

26.2： 正答

理由： For Each のベーステーブルがサービスに関連付けられたペット (Service.Pet) となっているため、パラメーターで受け取った特定のサービスに紐づくペットの一覧を適切に取得できる。

26.3： 誤答

理由： 外側の For Each のベーステーブルがサービス (Service) のため、サービスに関連付けられたペット (SERVICEPET テーブル) は拡張テーブルとならないため、適切なデータを取得できない。

26.4： 誤答

理由： 「26.2」が正答であるため。