

「GeneXus X Ev1シニアアナリスト」模擬試験

各設問に対して**解答は1つのみ**です。

各設問番号のとなりに表示されている数字はその設問の配点(%)です。

試験時間は180分で、合格に必要な正答率は70%以上です。

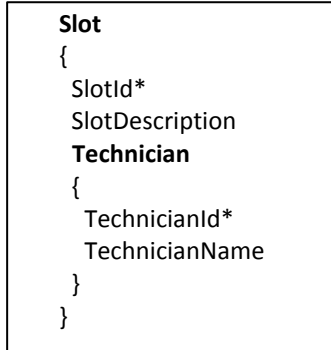
不正解の場合は加点されません(0になります)。

1. [5 %]

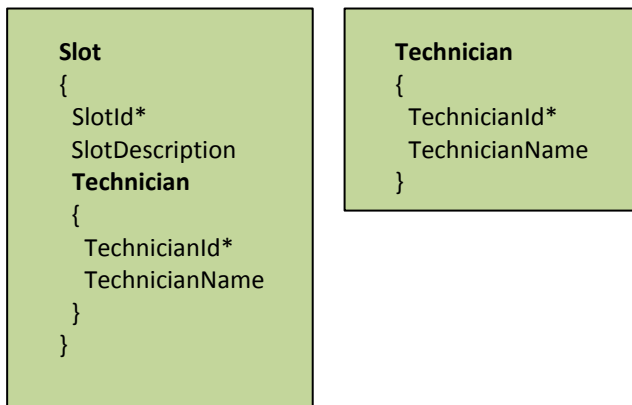
カジノを対象としてGeneXusで開発されたアプリケーションがあります。そこには、スロット(Slot)とその修理を担当する技術者(Technician)を記録するためのトランザクションがあります。

各スロット(Slot)は異なる技術者(Technician)によって修理され、1人の技術者(Technician)が複数のスロット(Slot)を修理することができる場合、この前提に基づく適切なトランザクション設計を選択してください。

a)



b)



c)

```
Slot
{
  SlotId*
  SlotDescription
  TechnicianId
  TechnicianName
}
```

```
Technician
{
  TechnicianId*
  SlotId*
  TechnicianName
}
```

d)

```
Slot
{
  SlotId*
  SlotDescription
  TechnicianId
  TechnicianName
}
```

```
Technician
{
  TechnicianId*
  TechnicianName
}
```

e) いずれも正しくない。

2. [5 %]

カジノを対象としてGeneXusで開発されたアプリケーションには、スロット (Slot) と、スロットタイプ (Type) を記録するトランザクションがあります。

各スロット (Slot) は1つの特定のタイプ (Type) にのみ対応し、同じタイプの多数のスロットが存在する場合、この前提に基づく適切なトランザクション設計を選択してください。

a)

```
Slot
{
  SlotId*
  SlotDescription
  Type
  {
    TypeId*
    TypeDescription
  }
}
```

b)

```
Type
{
  TypeId*
  TypeDescription
  Slot
  {
    SlotId*
    SlotDescription
  }
}
```

```
Slot
{
  SlotId*
  SlotDescription
}
```

c)

```
Type
{
  TypeId*
  TypeDescription
}
```

```
Slot
{
  SlotId*
  SlotDescription
  TypeId
  TypeDescription
}
```

d)

```
Type
{
  TypeId*
  TypeDescription
  SlotId
  SlotDescription
}
```

```
Slot
{
  SlotId*
  SlotDescription
}
```

3. [5 %]

カジノを対象として GeneXus で開発されたアプリケーションで、次のトランザクション構造が意味することを選択してください。

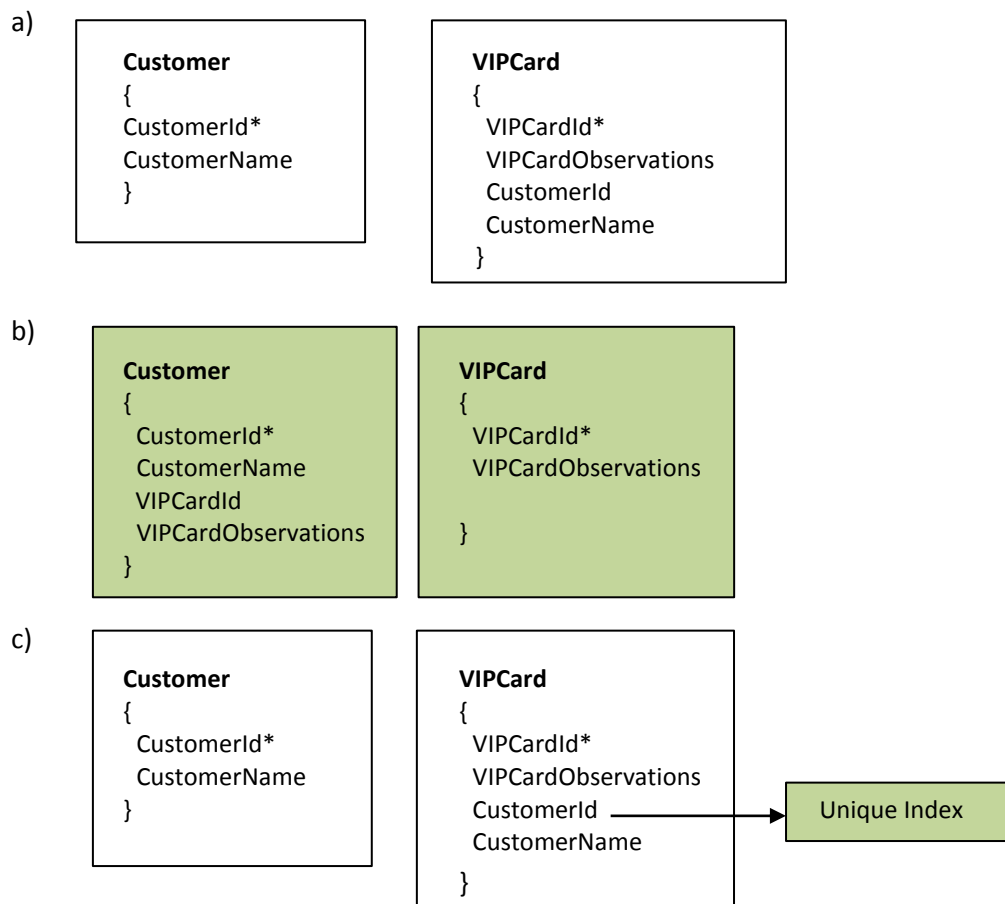
```
Slot
{
  SlotId*
  SlotDescription
  Prize
  {
    PrizeId*
    PrizeDescription
  }
}
```

- a) 各スロット (Slot) に対して複数の賞金 (Prize) があり、各賞金 (Prize) は複数のスロット (Slot) によって提供されることを示す。
- b) 各スロット (Slot) に対して複数の賞金 (Prize) があり、各賞金 (Prize) は 1 つのスロット (Slot) にのみ対応し、ほかのスロット (Slot) には対応しないことを示す (Slot および Prize がそれぞれ独立して存在する)。
- c) 各スロット (Slot) に対して複数の賞金 (Prize) があり、各賞金 (Prize) は 1 つのスロット (Slot) にのみ対応し、ほかのスロットには対応しないことを示す (独立して存在する Slot、およびその Slot に依存する SlotPrize)。
- d) 各スロット (Slot) に対して 1 つの賞金 (Prize) があり、賞金 (Prize) は 1 つのスロット (Slot) にのみ提供されることを示す。

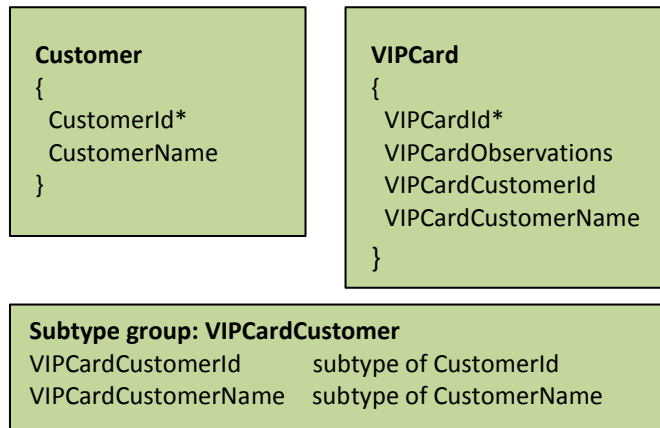
4. [5 %]

カジノを対象として GeneXus で開発されたアプリケーションには、顧客 (Customer) と、顧客に対して発行された VIP カード (VIPCard) を記録するトランザクションがあります。

各顧客 (Customer) は VIP カード (VIPCard) を 1 枚のみ所持しており、各 VIP カード (VIPCard) が 1 人の顧客 (Customer) にのみ帰属する場合、この前提に基づく適切なトランザクション設計を選択してください。



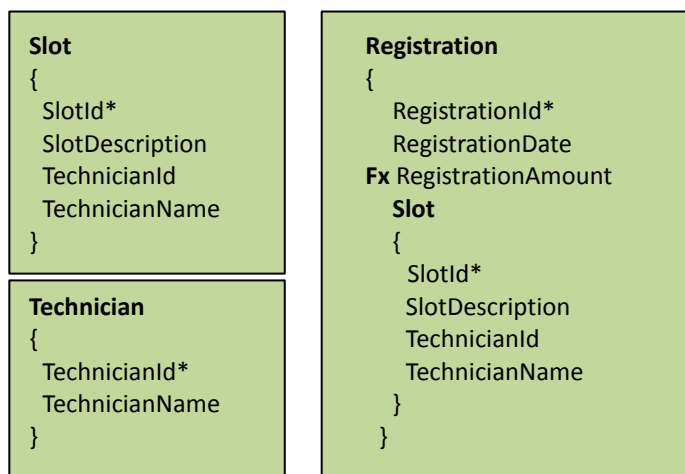
d)



e) いずれも正しくない。

5. [2.5%]

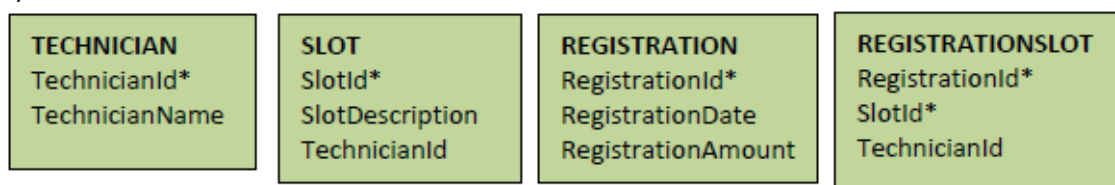
カジノを対象として GeneXus で開発されたアプリケーションで、GeneXus が設計および作成するテーブルの物理的な構造を選択してください。



a)



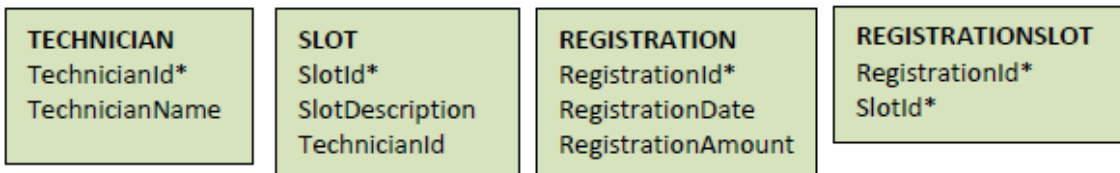
b)



c)



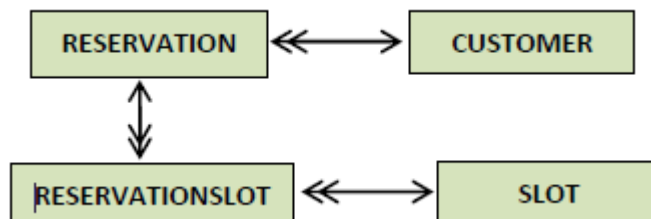
d)



e) いずれも正しくない。

6. [2.5%]

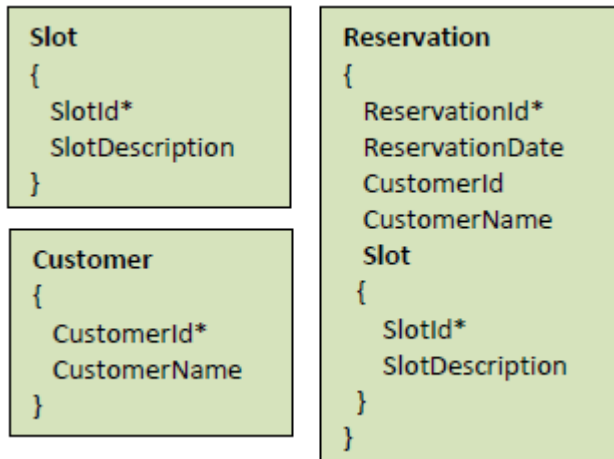
カジノを対象として GeneXus で開発されたアプリケーションで、次の ER 図から、RESERVATION ベーステーブルの拡張テーブルを選択してください。



- a) RESERVATION + RESERVATIONSLOT
- b) RESERVATION + CUSTOMER
- c) RESERVATION + RESERVATIONSLOT + SLOT
- d) いずれも正しくない。

7. [2.5%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図のようにスロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれています。



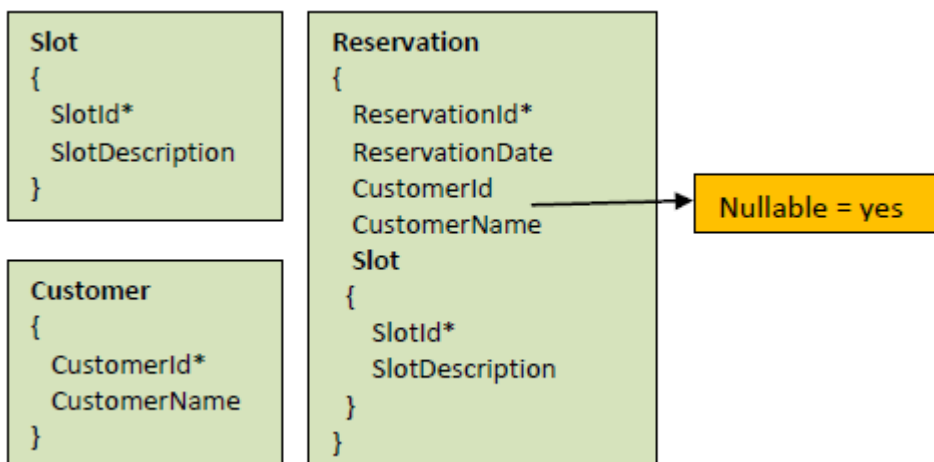
次の中から RESERVATIONSLOT ベーステーブルの拡張テーブルを選択してください。

- a) RESERVATIONSLOT + RESERVATION
- b) RESERVATIONSLOT + SLOT
- c) RESERVATIONSLOT + RESERVATION + SLOT
- d) RESERVATIONSLOT + RESERVATION + SLOT + CUSTOMER

8. [1.5%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図のようにスロット (Slot) およびスロットの予約 (Reservation) を記録する顧客のトランザクションのセットが含まれています。

場合によっては顧客 (CustomerId) を特定せずに予約を行います。



次の中から正しい記述を選択してください。

- a) CustomerIdに null 値の許容が定義されているため、参照整合性は CUSTOMER テーブルに対して制御されない。
- b) CustomerIdに null 値の許容が定義されていても値が指定される場合、参照整合性は CUSTOMER テーブルに対して制御される。
- c) いずれも正しくない。

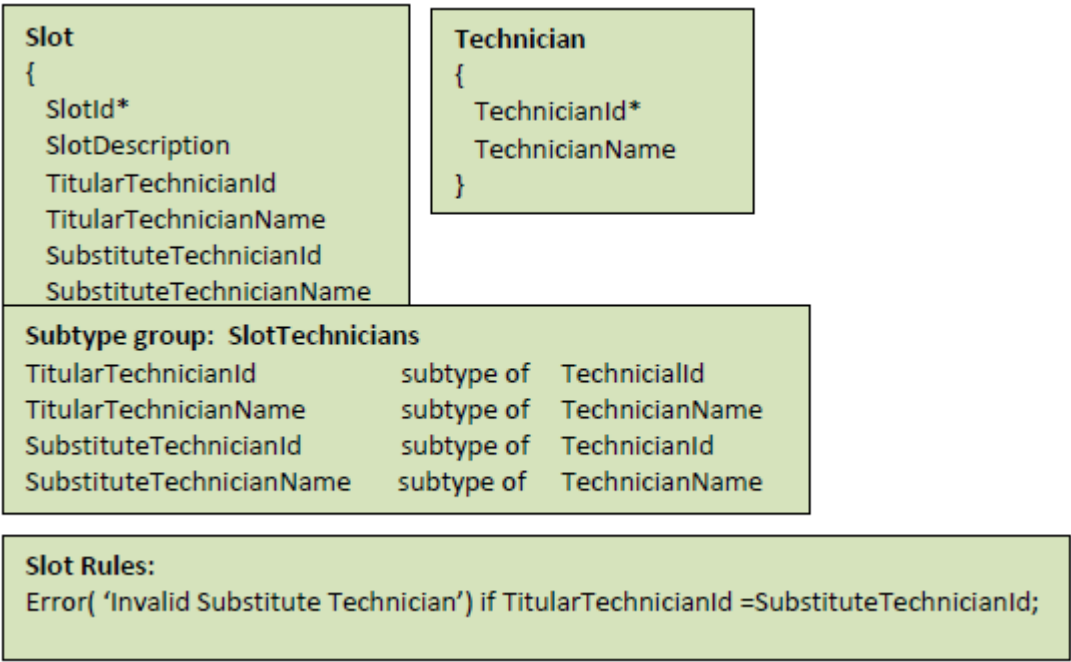
9. [3.5%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロット (Slot) およびその修理を担当する技術者 (Technician) を記録するトランザクションのセットが含まれています。

新しいスロットが記録されるたびに、担当する技術者および代理の技術者に関連付ける必要があります。

このシステムでは、2 名の技術者が異なっていることを確認する必要があります。

上記の要件を満たす次の実装が真（有効）か偽（無効）かを選択してください。



真: ☐ 偽: ☐

10. [3.5%]

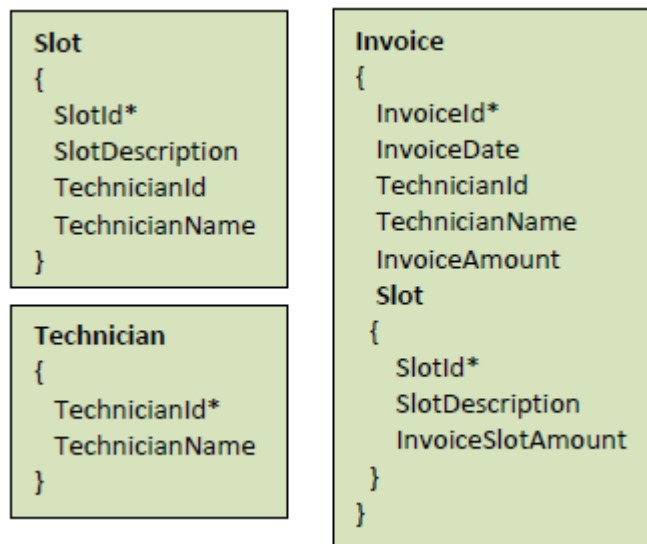
カジノを対象として GeneXus で開発されたアプリケーションには、スロット (Slot) およびその修理を担当する技術者 (Technician) を記録するトランザクションのセットが含まれています。

スロット (Slot) は、1 人の技術者 (Technician) にのみ修理され、各技術者 (Technician) は、修理要請 (Invoice) によって複数のスロット (Slot) を割り当てられます。

技術者 (Technician) が提供するサービスの請求時に、請求書 (Invoice) のヘッダーに記載される技術者 (Technician) と、明細に記載されるスロット (Slot) を実際に担当する技術者 (Technician) が一致していることを検証する必要があります。

次のオプションから、参照される要件の実装を選択してください。

a)



b)

```
Slot
{
  SlotId*
  SlotDescription
  TechnicianId
  TechnicianName
}
```

```
Technician
{
  TechnicianId*
  TechnicianName
}
```

```
Invoice
{
  InvoiceId*
  InvoiceDate
  TechnicianId
  TechnicianName
  InvoiceAmount
  Slot
  {
    SlotId*
    SlotDescription
    TechnicianId
    TechnicianName
    InvoiceSlotAmount
  }
}
```

Invoice Rules:

Error('Invalid Slot') if TechnicianId <> TechnicianId;

c)

```
Slot
{
  SlotId*
  SlotDescription
  TechnicianId
  TechnicianName
}
```

```
Technician
{
  TechnicianId*
  TechnicianName
}
```

```
Invoice
{
  InvoiceId*
  InvoiceDate
  InvoiceTechnicianId
  InvoiceTechnicianName
  InvoiceAmount
  Slot
  {
    SlotId*
    SlotDescription
    TechnicianId
    TechnicianName
    InvoiceSlotAmount
  }
}
```

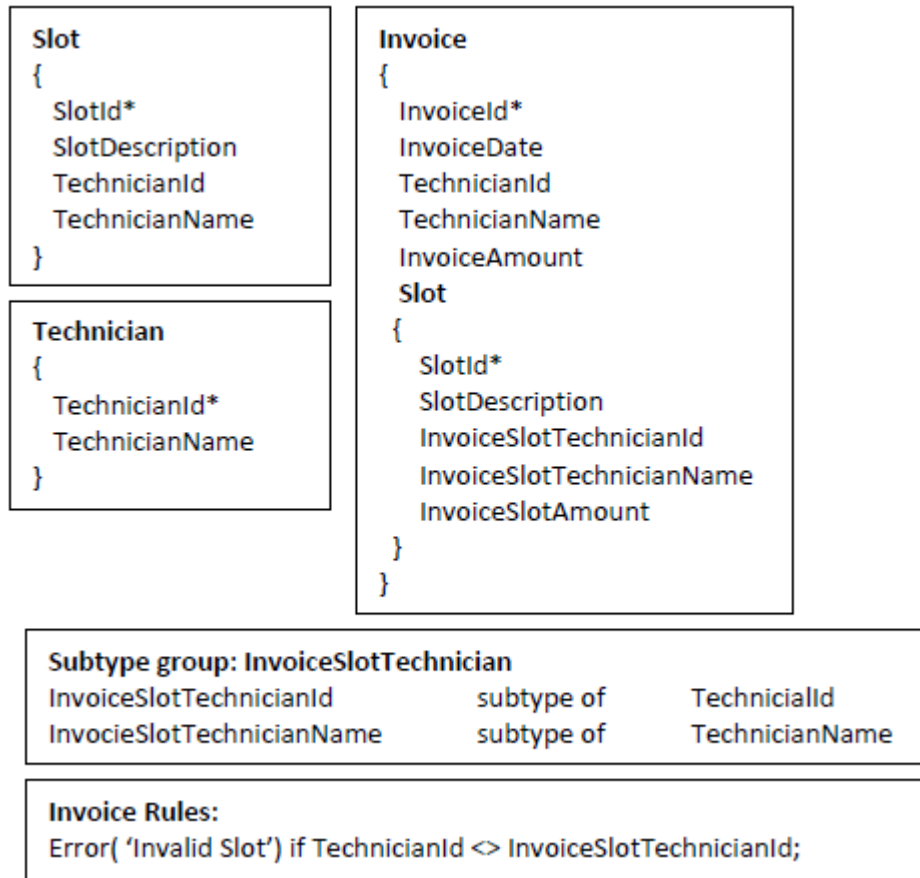
Subtype group: InvoiceTechnician

InvoiceTechnicianId	subtype of	TechnicianId
InvoiceTechnicianName	subtype of	TechnicianName

Invoice Rules:

Error('Invalid Slot') if TechnicianId <> InvoiceTechnicianId;

d)



e) いずれも正しくない。

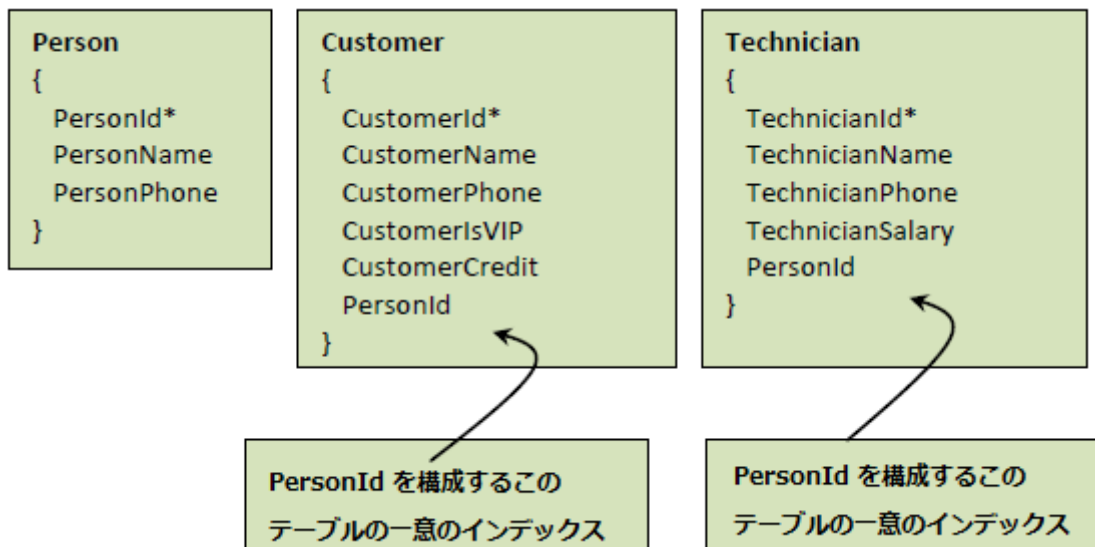
11. [2.5%]

カジノを対象として GeneXus で開発されたアプリケーションでは、修理するスロット (Slot) の技術者 (Technician) に関するデータおよびカジノの顧客 (Customer) に関するデータを記録する必要があります。

技術者 (Technician) と顧客 (Customer) は、共通のデータのセット (名前、電話番号など) がすべて個別に記録されているため、このアイディアでは、一度だけ共通のデータを記録します。

その後、特定のデータのみを記録する必要があります (たとえば、顧客の場合、VIP 顧客でその顧客にカジノから付与されたクレジットがある場合にはその内容が記録に含まれ、技術者の場合、給与が記録されます)。

上記の要件を満たす次の実装が、真 (有効) か偽 (無効) かを選択してください。



真: ☐ 偽: ☐

12. [1.5%]

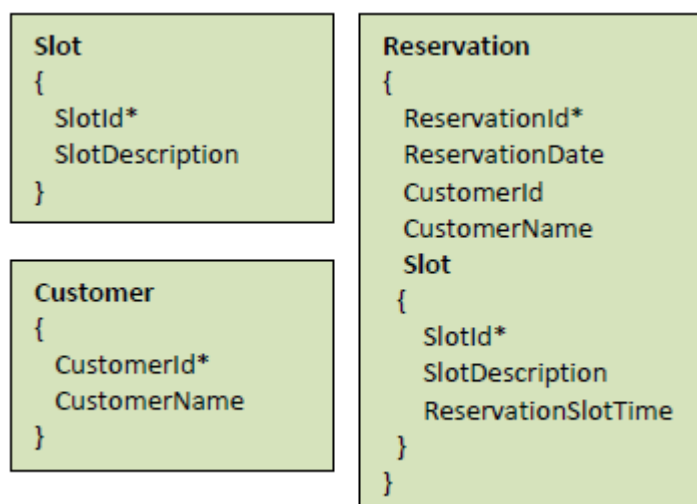
カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、Reservation トランザクションでは次のルールが宣言されています。

今日の日付で予約日を新規挿入する場合：

ReservationDate = Today() on AfterInsert;

上記のルール宣言が GeneXus の定義として適切であり、前提条件を満たしているでしょうか？

真（有効）か偽（無効）かを選択してください。



真: ☐ 偽: ☐

13. [1.5 %]

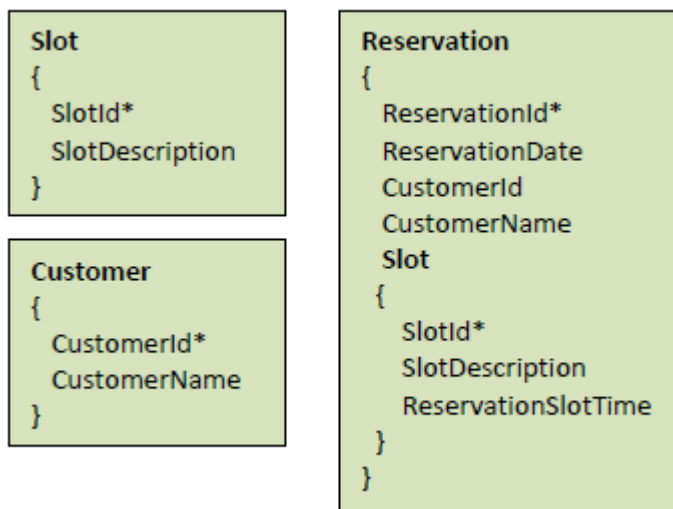
カジノを対象としてGeneXusで開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、Reservation トランザクションでは次のルールが宣言されています。

スロットIDが100より大きい時にスロット時間に60を加算してデータを更新する場合：

ReservationSlotTime += 60 if SlotId >100 on BeforeUpdate;

上記のルール宣言が GeneXus の定義として適切であり、前提条件を満たしているでしょうか？

真（有効）か偽（無効）かを選択してください。



真: ☐ 偽: ☐

14. [1.5 %]

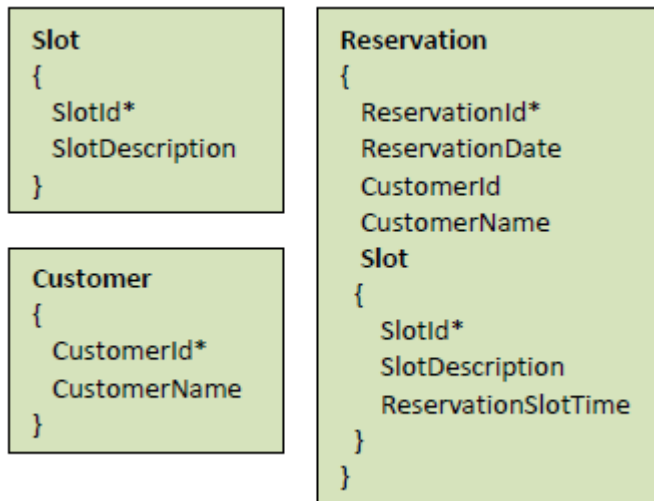
カジノを対象としてGeneXusで開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、Reservation トランザクションでは次のルールが宣言されています。

各明細行に何らかの処理を実行する場合：

Something.call(SlotId) on BeforeComplete;

上記のルール宣言が GeneXus の定義として適切であり、前提条件を満たしているでしょうか？

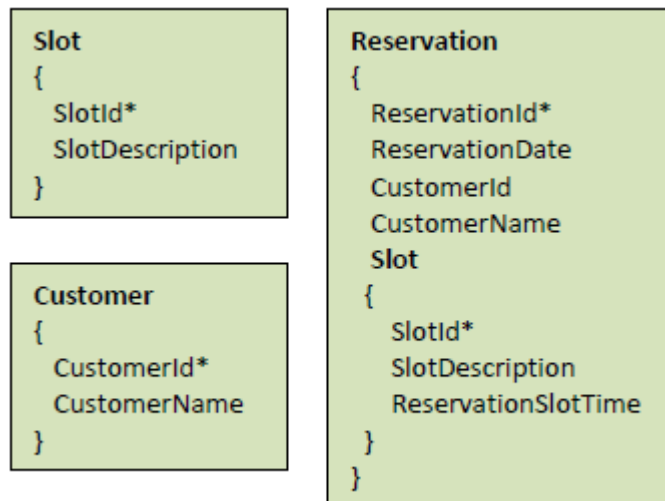
真（有効）か偽（無効）かを選択してください。



真: ☐ 偽: ☐

15. [2%]

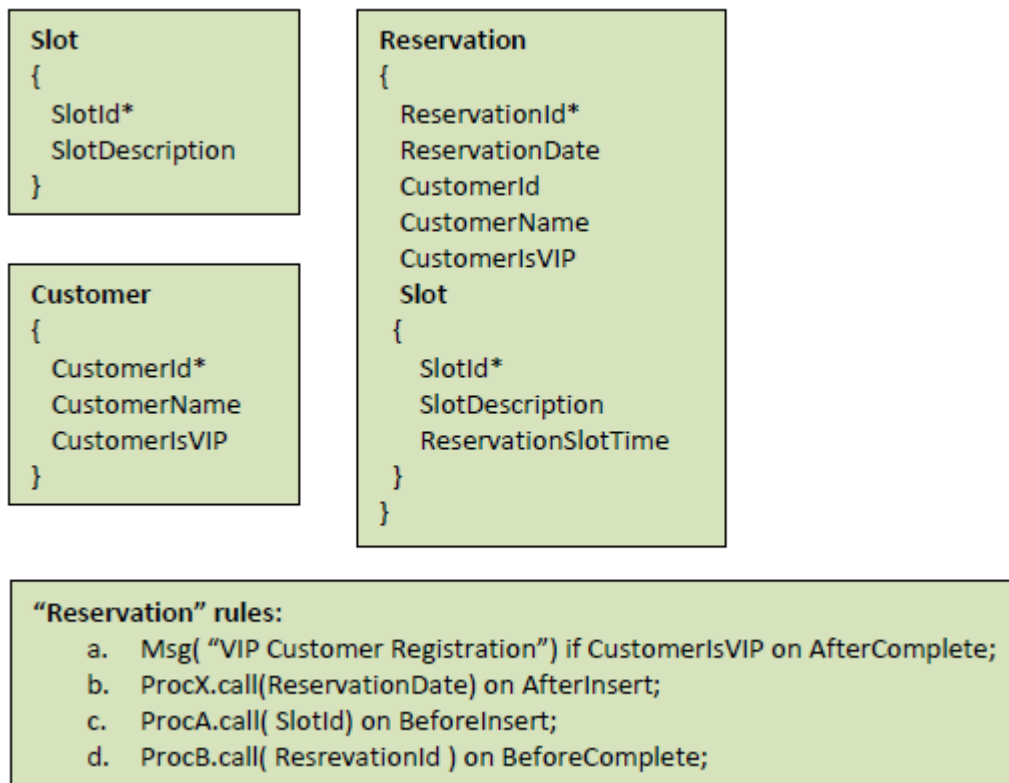
カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれています。Reservation データ (ヘッダーおよび明細) を処理するための、Proc. プロシージャの呼び出しを挿入し、コミット後にプロシージャーを呼び出す場合、次の中から適切な Reservation トランザクションのルールを選択してください。



- a) Proc.call(ReservationId) if Insert on BeforeComplete;
- b) Proc.call(ReservationId) If Insert on Commit;
- c) Proc.call(ReservationId) if Insert on AfterComplete;
- d) Proc.call(ReservationId) on AfterInsert;
- e) いずれも正しくない。

16. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、Reservation トランザクションでは次のルールが宣言されています。



次の中から実行される適切な順序を選択してください。

- a) ルールは、宣言された順序でトリガーされる。
- b) 次の順序で実行される： b. - c. - d. - a.
- c) 次の順序で実行される： c. - b. - d. - a.
- d) いずれも正しくない。

17. [2.5%]

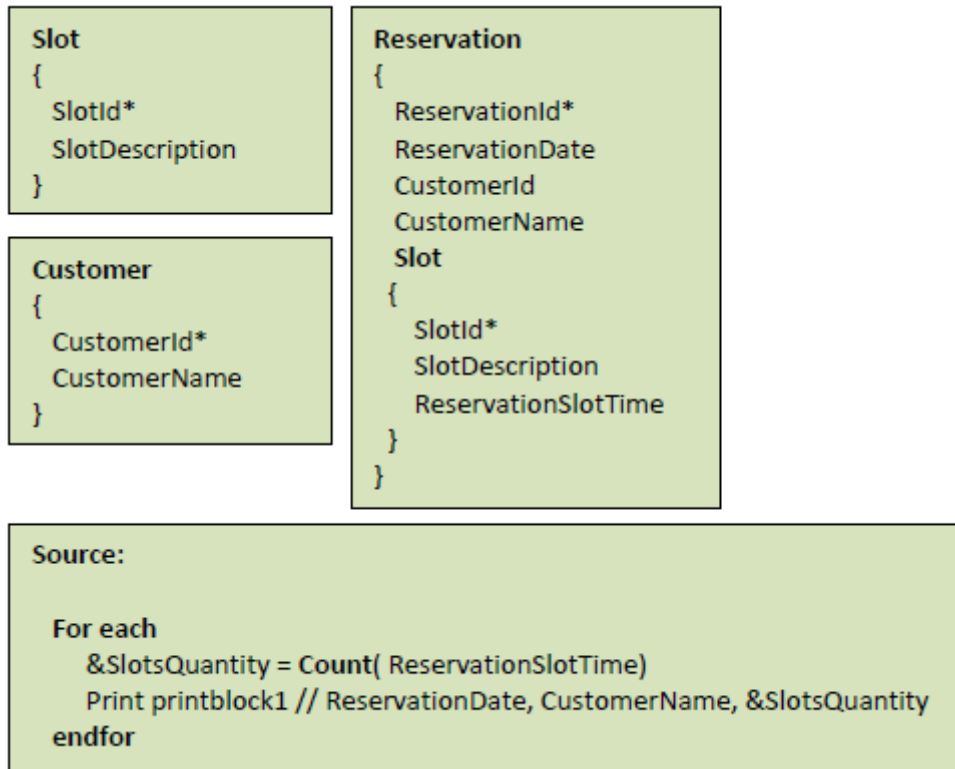
トランザクションのルールでは、物理的に各レベルに属するベーステーブルに属している項目属性のみを更新することができ、拡張テーブルに属している項目属性を更新することはできません。

この内容が真（有効）か偽（無効）かを選択してください。

真： ☐ 偽： ☐

18. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロット (Slot)、顧客 (Customer)、および顧客 (Customer) によるスロットの予約 (Reservation) を記録するトランザクションが含まれており、次のプロシージャーのソースが提供されます。



宣言されたローカル式が属するベーステーブルおよび参照テーブル（レコードのカウントを取得するために参照される）を次の中から選択してください。

- a) ベーステーブル: RESERVATION—参照テーブル: RESERVATIONSLOT
- b) ベーステーブル: RESERVATION—参照テーブル: RESERVATION
- c) ベーステーブル: CUSTOMER—参照テーブル: RESERVATION
- d) ベーステーブル: RESERVATIONSLOT—参照テーブル: RESERVATIONSLOT

19. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロット (Slot)、顧客 (Customer)、および顧客 (Customer) によるスロットの予約 (Reservation) を記録するトランザクションが含まれており、今日の日付で 10 件を超える予約がある顧客のリストを印刷する必要があります。

上記の要件を満たす次の実装が真（有効）か偽（無効）かを選択してください。

Slot { SlotId* SlotDescription }	Reservation { ReservationId* ReservationDate CustomerId CustomerName Slot { SlotId* SlotDescription ReservationSlotTime } }
Customer { CustomerId* CustomerName }	

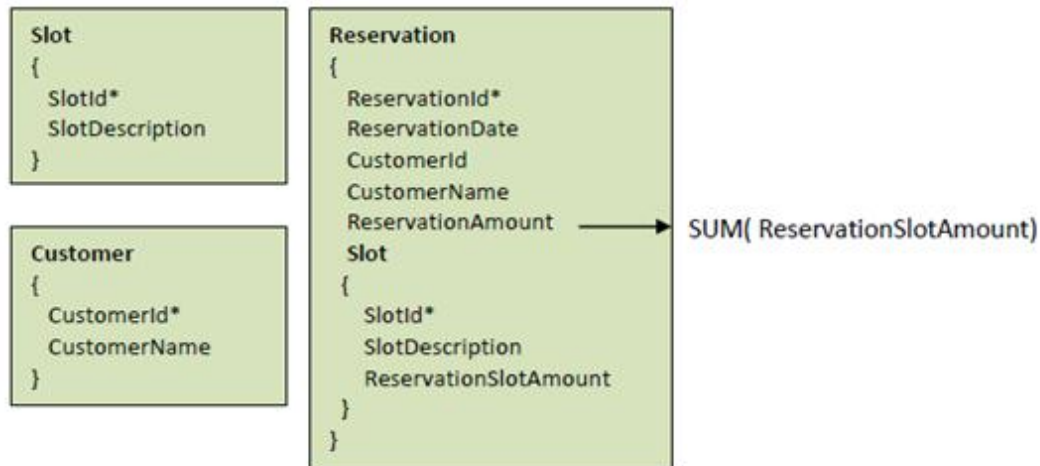
Source:

```
For each  
Where Count( ReservationDate, ReservationDate = Today() ) > 10  
  Print printblock1 // CustomerId, CustomerName  
endfor
```

真: ☐ 偽: ☐

20. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、次のようなネストされた For each のペアがある場合、適切なベーステーブルを選択してください。

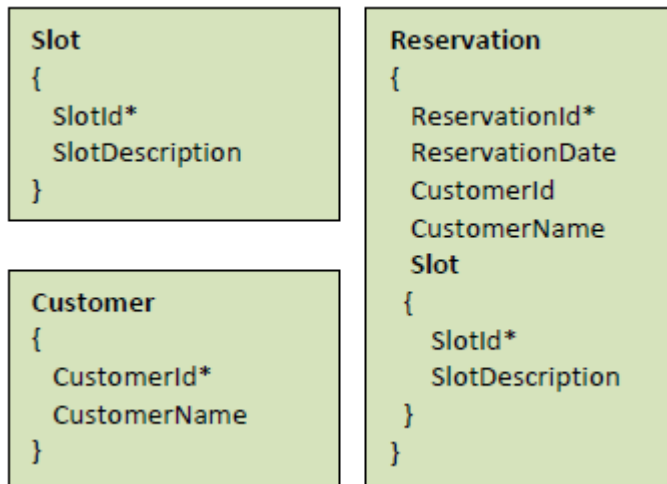


```
For each
  Print printblock1 // ReservationId, ReservationDate, SlotDescription
  For each
    Print printblock2 // CustomerName
  endfor
Endfor
```

- a) 外部 For each: RESERVATIONSLLOT 内部 For each: CUSTOMER
- b) 外部 For each: RESERVATION 内部 For each: CUSTOMER
- c) 外部 For each: SLOT 内部 For each: CUSTOMER
- d) 外部 For each: RESERVATIONSLLOT 内部 For each: RESERVATIONSLLOT

21. [3%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、次のようなプロシージャーが実装されている場合、適切な実行結果を選択してください。

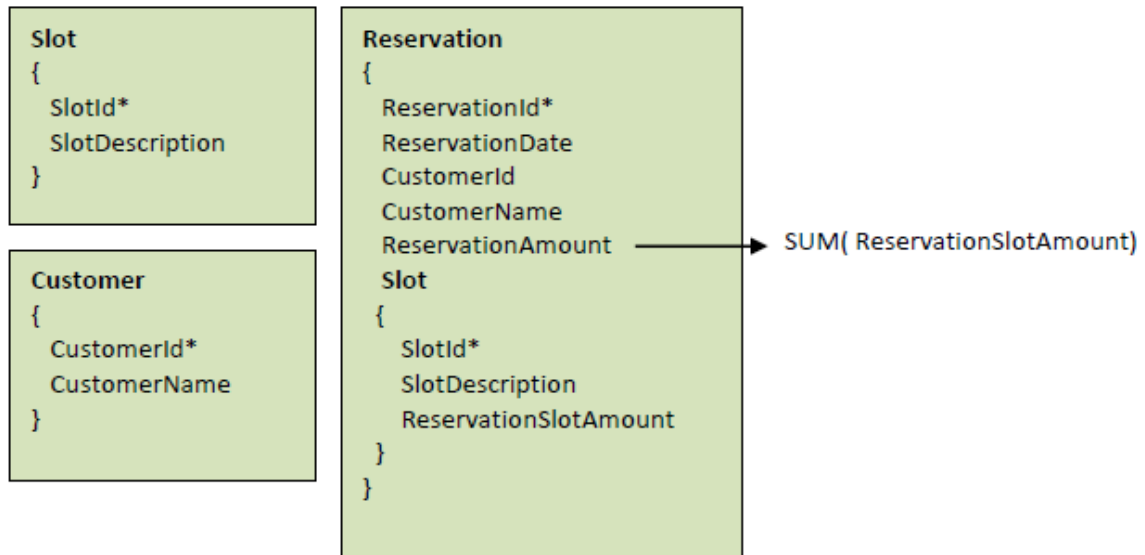


```
For each
  Print printblock1 // CustomerName
  For each
    Print printblock2 // ReservationId, ReservationDate, SlotDescription
  endfor
Endfor
```

- a) デカルト積 : RESERVATIONSLOT テーブルに記録されたすべてのスロットと同様に、抽出条件なしに、Customer テーブル内のすべての顧客がリストされる。つまり、各顧客 (Customer) によって予約 (Reservation) されたすべてのスロット (Slot) が表示される。
- b) Join : CUSTOMER テーブルからすべての顧客 (Customer) がリストされ、各顧客に対して、(自動的に CustomerId によってフィルタリングされた) 顧客の予約に対応する、RESERVATIONSLOT テーブルに記録されているスロットのみが表示される。
- c) いずれも正しくない。

22. [4%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、各顧客のスロットの予約とともにすべての顧客を表示する一覧が必要です（予約がない場合でも、すべての顧客を表示する必要があります）。



この要件を満たす実装を次の中から選択してください。

a)

```
For each order CustomerId
defined by ReservationDate
  Print printblock1 // CustomerId, CustomerName
  For each
    Print printblock2 // ReservationId, ReservationDate, ReservationAmount
  endfor
Endfor
```

b)

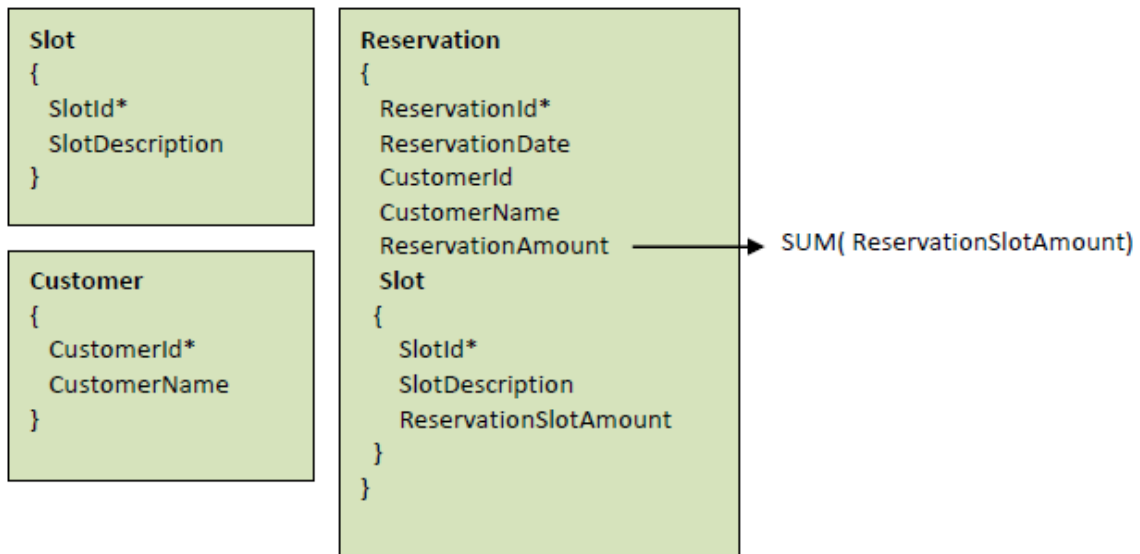
```
For each
  Print printblock1 // CustomerId, CustomerName
  For each
    Print printblock2 // ReservationId, ReservationDate, ReservationAmount
  endfor
Endfor
```

c) いずれも正しくない。

23. [4%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、顧客ごとのスロットの予約一覧を表示する必要があります。このリストには、**予約のある顧客のみを表示する必要があります。**

上記の要件を満たす次の実装が真（有効）か偽（無効）かを選択してください。

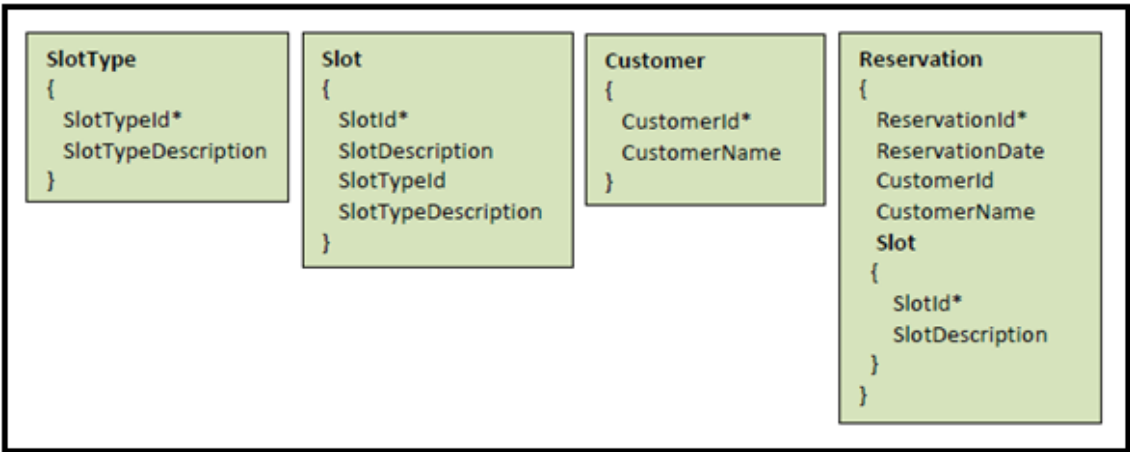


```
For each
defined by ReservationDate
  Print printblock1 // CustomerId, CustomerName
  For each
    Print printblock2 // ReservationId, ReservationDate, ReservationAmount
  endfor
Endfor
```

真: ☐ 偽: ☐

24. [3%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、スロットタイプ (SlotType)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、2 番目の図のようなスロットの予約リストが生成されます。つまり、このリストはスロットタイプごとの顧客別予約一覧であり、予約がない顧客およびスロットタイプは表示されてはいけません。



Slot Type: 1 Wizards		
Customer: 15 Ann Smith		
<u>Reservation Id</u>	<u>Date</u>	<u>Slot Description</u>
.....
.....
.....
Customer: 120 Peter Jones		
<u>Reservation Id</u>	<u>Date</u>	<u>Slot Description</u>
.....
.....
Slot Type: 2 Wand		
Customer: 15 Ann Smith		
<u>Reservation Id</u>	<u>Date</u>	<u>Slot Description</u>
.....

次の中から適切と思われる実装を選択してください。

a)

```
For each order SlotTypeId
  Print TypePB // SlotTypeId, SlotTypeDescription
  For each order CustomerId
    Print CustomerPB // CustomerId, CustomerName
    For each order ReservationId
      Print ReservationPB // ReservationId, ReservationDate, SlotDescription
    Endfor
  Endfor
Endfor
```

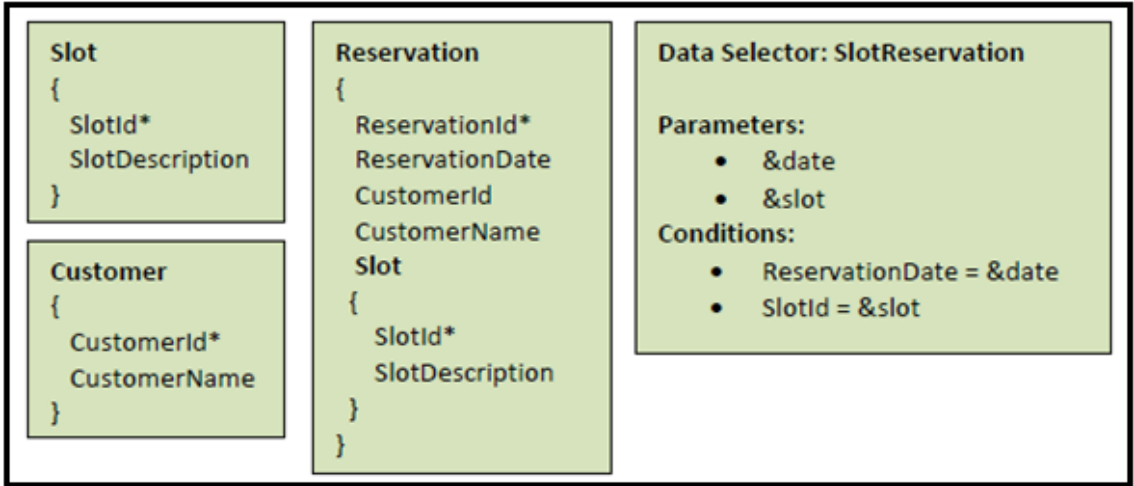
b)

```
For each order SlotTypeId, CustomerId, ReservationId
  Print TypePB // SlotTypeId, SlotTypeDescription
  For each
    Print CustomerPB // CustomerId, CustomerName
    For each
      Print ReservationPB // ReservationId, ReservationDate, SlotDescription
    Endfor
  Endfor
Endfor
```

c) いずれも正しくない。

25. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、特定の日付かつスロットのすべての予約を取得する、頻繁に使用されるクエリのために、次のようなデータセクターが実装されました。これに続き、2 番目の図のようなデータセクターを参照する Web パネルが実装されました。



The diagram shows a Web panel Form and its configuration. The Web panel Form is a table with columns for CustomerName and SlotId. The Grid: Grid1 configuration is as follows:

Grid: Grid1	
ControlName	Grid1
Order	
Conditions	
DataSelector	SlotReservation
Parameters	&Today, 1 ...
Collection	
Rules	None
+ Appearance	
+ Layout	
+ Behavior	

上記を前提とし、次の中から適切と思われるオプションを選択してください。

- a) 実行すると、RESERVATIONSLOT となる Web パネルのベーステーブルに対して重複する顧客名が表示されるが、実装された Web パネルは、スロット ID が 1 で今日日付の予約がある顧客名のグリッドで表示される。
- b) 実行すると、Web パネルのベーステーブルが CUSTOMER となるので、実装された Web パネルは、スロット ID が 1 で今日日付の予約がある顧客名のグリッドで表示され、各顧客名が一度だけ表示される。
- c) いずれも正しくない。

26. [3%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot)、顧客 (Customer)、およびスロットの予約 (Reservation) を記録するトランザクションのセットが含まれており、次の SDT で示すようなカジノの顧客をロードする構造化データのセットが提供されます。また、指定した予約日から各顧客が予約したスロットのリストも取り出します。

上記の要件を満たす次の実装が真（有効）か偽（無効）かを選択してください。

Slot

{

SlotId*

SlotDescription

}

Customer

{

CustomerId*

CustomerName

}

Reservation

{

ReservationId*

ReservationDate

CustomerId

CustomerName

Slot

{

SlotId*

SlotDescription

}

}

Client

Id

Name

SlotsAmount

SlotsReserved

Slot

Description

Data Provider Source:

Rule: parm(&dateFrom);

Client

{

Id = CustomerId

Name = CustomerName

SlotsAmount = Count(SlotId, ReservationDate >= &dateFrom)

SlotsReserved where ReservationDate >= &dateFrom

{

Slot

{

Description = SlotDescription

}

}

}

Output

Client

Collection

True

Collection Name

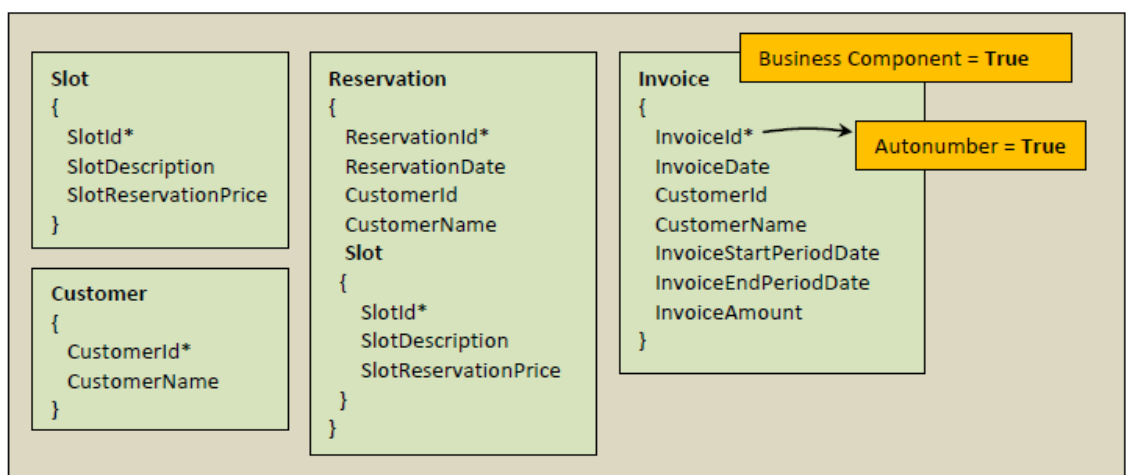
ClientsReservat...

真: ☐ 偽: ☐

27. [3%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、請求目的でスロット(Slot)、顧客(Customer)、およびスロットの予約(Reservation)を記録するトランザクションのセットが含まれており、顧客ごとの請求書が生成され、特定の請求期間にスロットの予約の集計とともに自動的に記録されます。このためには、2 番目の図のような “GetInvoices” データプロバイダーが宣言されます。

上記の前提で以下の内容が真（有効）か偽（無効）か選択してください：“データプロバイダーは、構造化データのロードのみ可能であり、データベースに記録することができないため、別のオブジェクトからデータプロバイダーを呼び出し、下記のように明示的に記録する必要がある。”



Data Provider: GetInvoices

```
Parm( in: &startDate, in: &endDate );

Invoice
{
    InvoiceDate = Today()
    CustomerId
    InvoiceStartPeriodDate = &startDate
    InvoiceEndPeriodDate = &endDate
    InvoiceAmount = Sum( SlotReservationPrice, ReservationDate >= &start and ReservationDate <= &end)
}
```

Output	
Output	Invoice
Collection	True
Collection Name	Invoices

```
For &invoice in GetInvoices( #2011-01-01#, #2011-01-31#)
    &invoice.Save()
    If &invoice.Success()
        Commit
    else
        Rollback
    endif
endfor
```

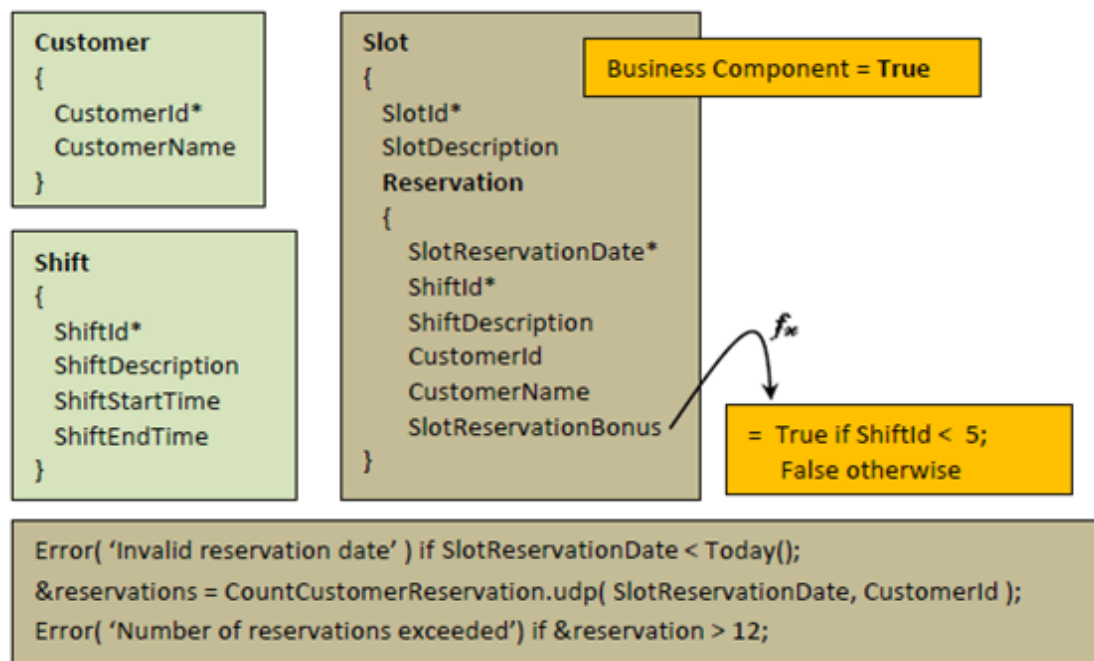
真: ☐ 偽: ☐

28. [3%]

カジノを対象として GeneXus で開発されたアプリケーションに関する設問です。カジノは毎日 24 時間営業しており、顧客のスロットの利用を円滑にするために、1 日を 2 時間ごとの 12 のシフトに分割し、このシフト内で顧客が予約できるようにしています。

このために顧客 (Customer)、シフト (Shift)、および予約 (Reservation) を含むスロット (Slot) トランザクションが定義されています。次の図で示すように、スロット (Slot) には式とルールが定義されています。

新しいスロットである “The Wizard” は、VIP 顧客のリクエストによって購入され、ID 100 でシステムに登録されています。この顧客は、新しいスロットが初めて稼働する日に、シフト 1 と 7 で予約されます。



この要件を満たす実装を次の中から選択してください。

a)

```
// &slot → data type: Slot
// &reservation → data type: Slot.Reservation

&slot.SlotId = 100
&slot.SlotDescription = "The Wizard"

&reservation.SlotReservationDate = &date
&reservation.ShiftId = 1
&reservation.CustomerId = &customerId

&slot.Reservation.Add( &reservation )

&reservation = new ()

&reservation.SlotReservationDate = &date
&reservation.ShiftId = 7
&reservation.CustomerId = &customerId

&slot.Reservation.Add( &reservation )

&slot.Save()
```

b)

```
parm( in: &customerId, in: &date);  
  
// &slot → data type: Slot  
// &reservation → data type: Slot.Reservation  
  
&slot.SlotId = 100  
&slot.SlotDescription = "The Wizard"  
  
&reservation.SlotReservationDate = &date  
&reservation.ShiftId = 1  
&reservation.CustomerId = &customerId  
  
&slot.Reservation.Add( &reservation )  
  
&reservation = new ()  
  
&reservation.SlotReservationDate = &date  
&reservation.ShiftId = 7  
&reservation.CustomerId = &customerId  
  
&slot.Reservation.Add( &reservation )  
  
&slot.Save()  
if &slot.Success()  
    Commit  
else  
    Rollback  
endif
```

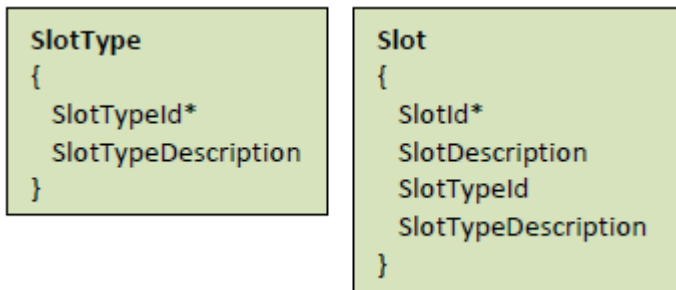
c)

```
parm( in: &customerId, in: &date);  
  
// &slot → data type: Slot  
// &reservation → data type: Slot.Reservation  
  
&slot.SlotId = 100  
&slot.SlotDescription = "The Wizard"  
  
&reservation.SlotReservationDate = &date  
&reservation.ShiftId = 1  
&reservation.CustomerId = &customerId  
  
&slot.Save()  
  
&slot.Reservation.Add( &reservation )  
  
&reservation = new ()  
  
&reservation.SlotReservationDate = &date  
&reservation.ShiftId = 7  
&reservation.CustomerId = &customerId  
  
&slot.Reservation.Add( &reservation )  
  
&slot.Save()  
  
if &slot.Success()  
    Commit  
else  
    Rollback  
endif
```

d) いずれも正しくない。

29. [2.5%]

カジノを対象として GeneXus で開発された Web アプリケーションには、次の図で示すように、スロット (Slot) とスロットタイプ (SlotType) を記録するトランザクションが含まれています。新しいスロットタイプがシステムに登録されるたびに、このタイプのスロットが既に存在するかどうかの結果を表示します。登録されたスロットがないと、スロットタイプが残されないため (同じ論理作業単位を共有します)、すぐにスロットを登録する必要があります。



次の中から適切と思えるオプションを選択してください。

- a) SlotType トランザクションのコミット前に、SlotTypeId の値をパラメーターとして使用し、Slot トランザクションを呼び出す。
- b) SlotType トランザクションのコミット直前に、SlotTypeId の値をパラメーターとして使用し、スロットを記録するプロシーチャーを呼び出す。プロシーチャーの終了時のコミットは無効にする。
- c) 2 つの変数で Web パネルを作成する: 1 つはデータタイプ = “SlotType” のビジネスコンポーネントで、もう 1 つはデータタイプ = “Slot” のビジネスコンポーネントを作成する。フォームにこれら挿入して、2 つのレコード (保存) を実行する関連するイベントの確認ボタンを追加し、エラーが検出されなかった場合には、Commit コマンドを実行する。
- d) a) と b) のみ正しい。
- e) b) と c) のみ正しい。

30. [1.5%]

カジノを対象として GeneXus で開発されたアプリケーションに関する設問です。Web パネルはフリースタイルグリッドで表示するために設計され、修理のために各技術者 (Technician) に割り当てられた複数のスロット (Slot) があります。

適切と思われる実装を選択してください。

```

Technician
{
  TechnicianId*
  TechnicianName
}

```

```

Slot
{
  SlotId*
  SlotDescription
  TechnicianId
  TechnicianName
}

```

Technician:

Slot Id	Slot Description
<input type="text" value="SlotId"/>	<input type="text" value="SlotDescription"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Quantity:

Grid: Grid2

Freestyle Grid: Grid1

a)

```

Event Start
  &SlotsQuantity += 1
EndEvent

```

b)

```

Event Grid1.Load
  &SlotsQuantity += 1
EndEvent

```

c)

```

Event Grid1.Load
  &SlotsQuantity = count( SlotId )
EndEvent

```

d)

```

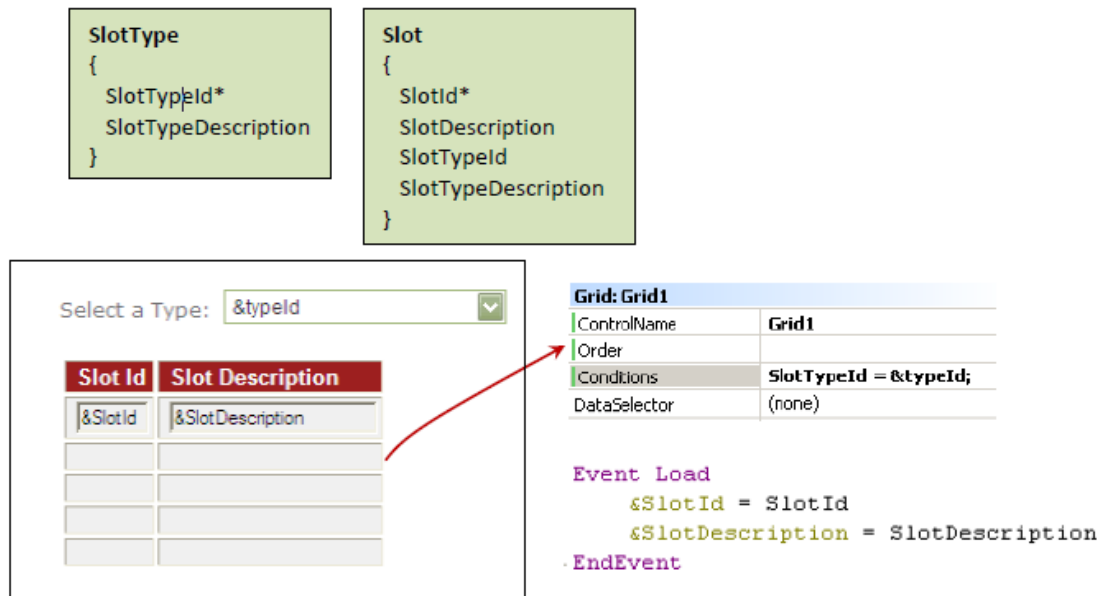
Event Grid1.Refresh
  &SlotsQuantity = count( SlotId )
EndEvent

```

e) いずれも正しくない。

31. [3.5%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロットタイプ (SlotType) を記録する トランザクションおよびスロット (Slot) を記録する別の トランザクションが含まれており、選択されたスロットタイプでグリッドにすべてのスロットを表示できるように Web パネルが設計されています。このためには、下図のような Web パネルが定義されています。



次の中から適切と思えるオプションを選択してください。

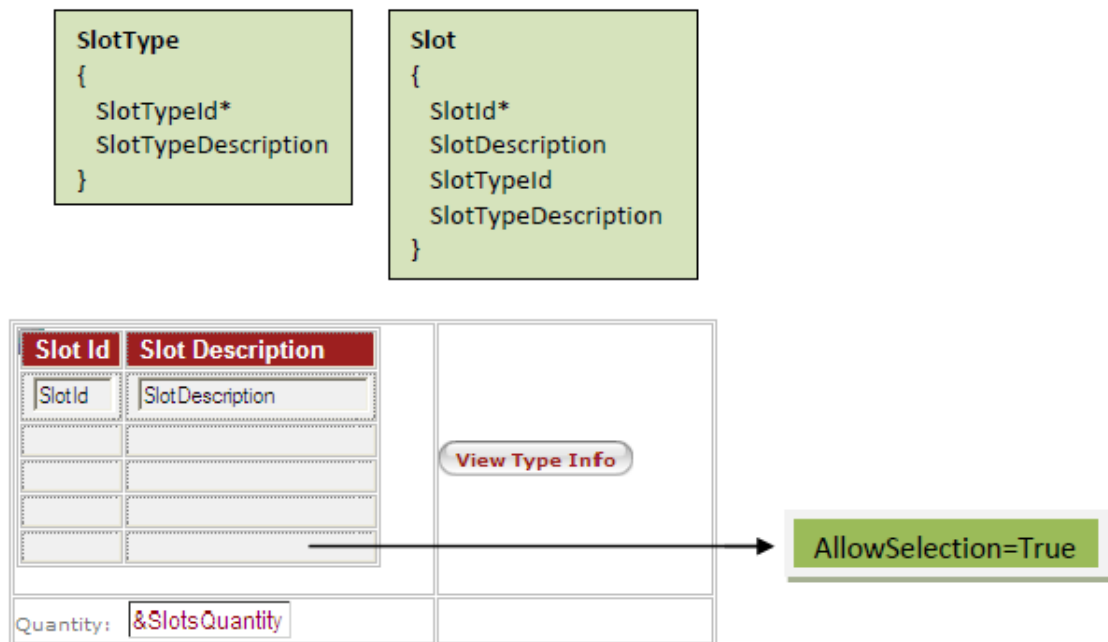
- 条件内に SlotTypeId 項目属性および Load イベントに SlotId と SlotDescription が存在するので、Web パネルが正しく機能する。一度選択されたスロットタイプのスロットごとに Load イベントがトリガーされると、GeneXus が実装されたベーステーブルを推論し、Load イベント内に Load コマンドを配置する必要がない。
- GeneXus は、グリッド内に項目属性が存在しないことからベーステーブルがないことを理解しているため、Web パネルは明細にエラーを表示する。このため、条件の定義または For each コマンドの外側で Load イベントの項目属性を使用することは無意味である。
- Web パネルはエラーを表示しないが、GeneXus には実装されたベーステーブルがないので、誤ってプログラムされ、グリッドに何も表示されない。このため、明示的にロードを定義し、グリッドの各行をロードする Load コマンドを使用するために For each を配置する必要がある。
- いずれも正しくない。

32. [3%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロットタイプ (SlotType) を記録するトランザクションおよびスロット (Slot) を記録する別のトランザクションが含まれており、次の目的で Web パネルが設計されています。

- 1) グリッドにすべてのスロットを表示する。
- 2) スロットタイプ ID (SlotTypeId) が 3、5、および 7 をカウントした結果をグリッドの下に表示する。
- 3) ユーザーがグリッドから行を選択すると、対応するスロットタイプの情報を表示する別の Web パネルを呼び出すことを可能にする。

上記の前提を満たすには、グリッドに不可視の項目属性が追加されない、次のような Web パネルが定義されます。



次の中から適切と思われるオプションを選択してください。

- a) 1)のみが Web パネルの必要条件を満たしている。
- b) 1)と 2)のみが Web パネルの必要条件を満たしている。
- c) 上記の 3 つとも Web パネルの必要条件を満たしている。

33. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、スロット (Slot)、顧客 (Customer)、スロットの予約 (Reservation) を記録するトランザクションが含まれており、次の図で示すような、予約数とともに今日の日付で予約されたすべてのスロットを表示する Web パネルが実装されています。

Slot

{

SlotId*

SlotDescription

}

Customer

{

CustomerId*

CustomerName

}

Reservation

{

ReservationId*

ReservationDate

CustomerId

CustomerName

Slot

{

SlotId*

SlotDescription

}

}

Slot Id	Slot Description	Reservation Amount
&SlotId	&SlotDescription	&Amount

Event Grid1.Load

For each order SlotId

where ReservationDate = &Today

&SlotId = SlotId

&SlotDescription = SlotDescription

&Amount = 0

For each

defined by ReservationDate

&Amount += 1

endfor

Load

endfor

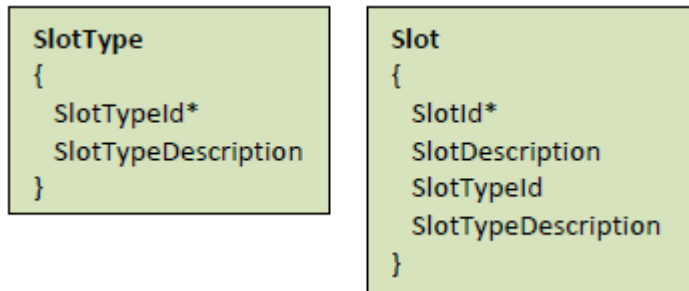
EndEvent

今日の日付で 2 つのスロットのみが予約されている場合、最初のスロットには 5 件の予約があり、2 つめのスロットには 4 件の予約があるとします。Web パネルの実行時にトリガーされる Load イベントの回数を次の中から選択してください。

- a) 9 回
- b) 2 回
- c) 11 回
- d) いずれも正しくない。

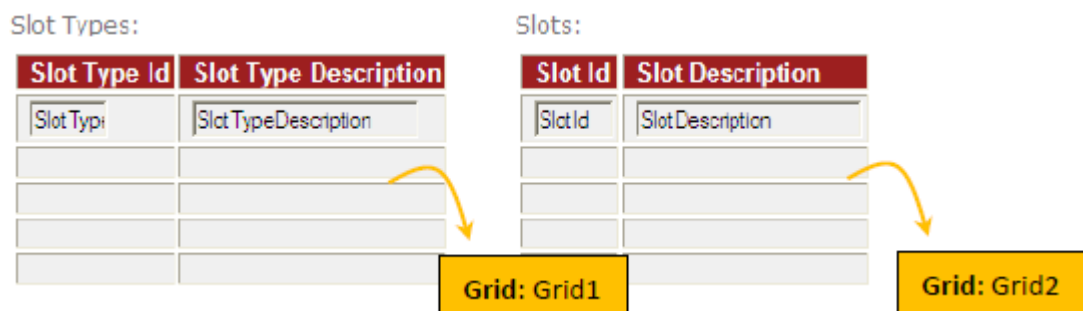
34. [1%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すようにスロット (Slot) およびスロットタイプ (SlotType) を記録するトランザクションが含まれており、次をロードする 2 つの並行グリッドを持つ Web パネルが実装されています： 1 つめのグリッドはスロットタイプを表示し、2 つめのグリッドはスロットを表示する。ユーザーが 1 つめのグリッドで選択したスロットタイプのスロットのみを 2 つめのグリッドに表示する (Allowselection プロパティは有効)。

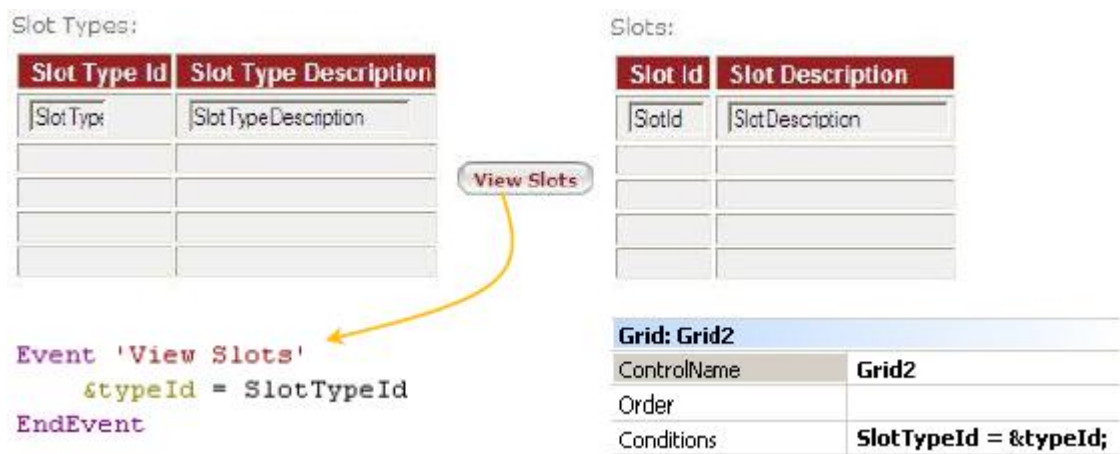


次の中から適切な実装を選択してください。

- a) 両方のグリッドにベーステーブルがあり、相互に関連しているため、GeneXus はユーザーが Grid1 の行を選択すると、関連するスロットが自動的に Grid2 にロードされるようにする。



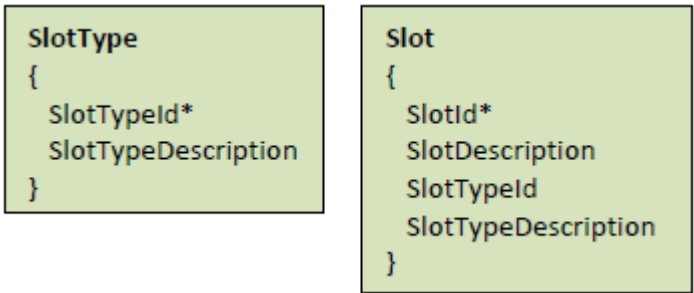
- b) 両方のグリッドにベーステーブルがあり、相互に関連しているが、GeneXus はそれらの関係を決定していないので、実装する必要がある。このため、次を実装するだけで十分であり、画面上に &typeId 変数は必要ない。



- c) 画面上に&typeId 変数が配置された（非表示）場合のみ、オプション b) の実装は正しい。
- d) いずれも正しくない。

35. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot) とスロットタイプ (SlotType) を記録するトランザクションが含まれており、Work With パターンが両方のトランザクションに適用されています。ただし、前提条件として生成される “WWSlotType” の Web パネルには、スロットタイプを削除する機能が不要です。



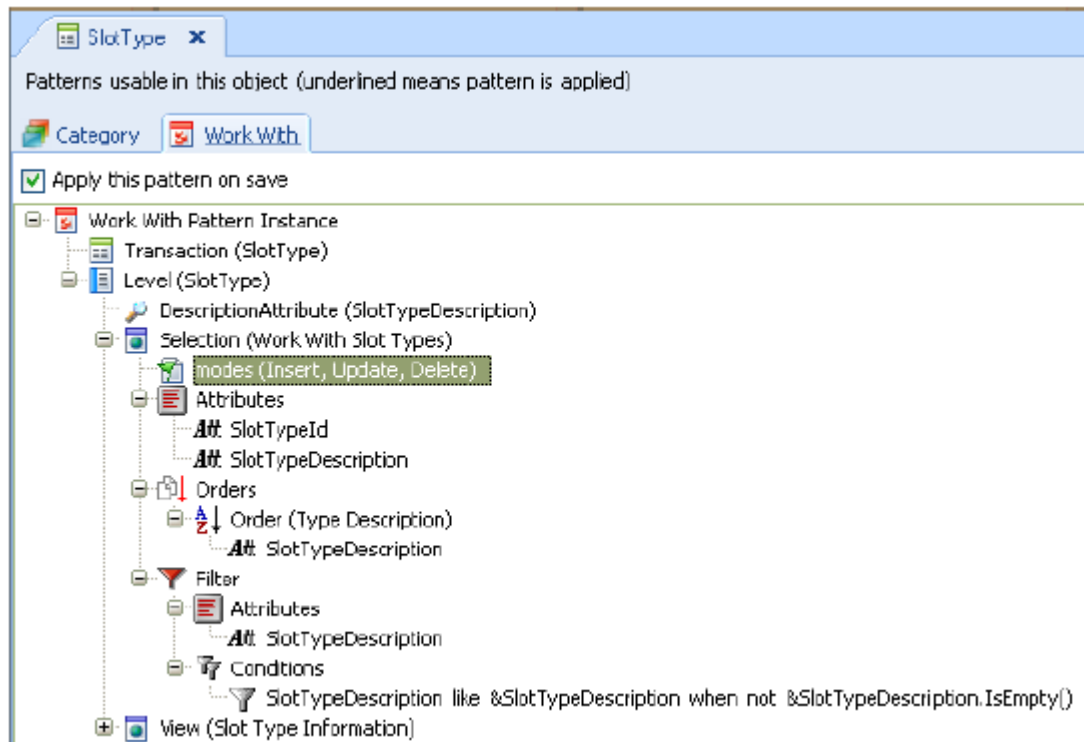
次のオプションから、上記の前提を満たす適切と思われる実装を選択してください。

- a) パターンによって自動的に生成された “WWSlotType” の Web パネルを編集し、それを参照するイベントのプログラムと同様に&Delete 変数の列を削除する。

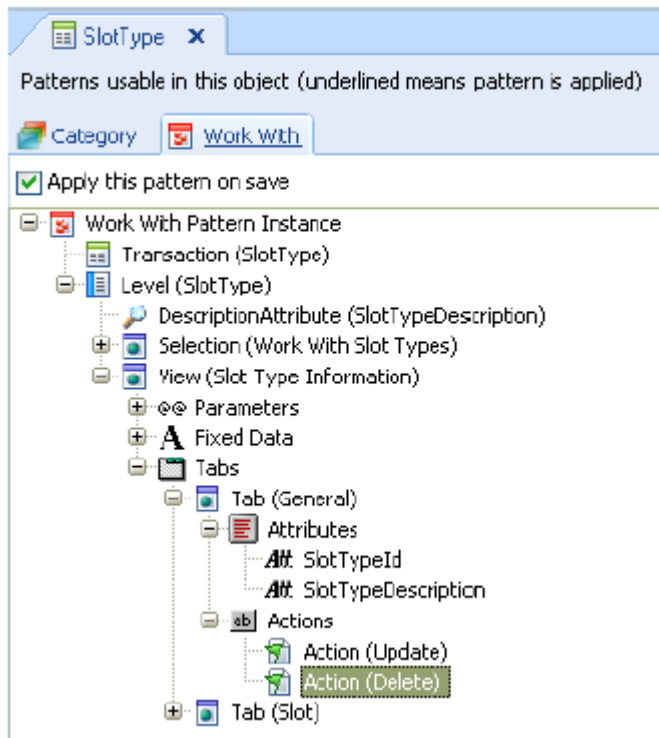
The screenshot shows a web panel titled "Work With Slot Types". It has a "Type Description" field with the value "&SlotTypeDescription". Below this is a table with columns "Type Id" and "Type Description". The first row of the table has "&Update" and "&Delete" in the "Type Id" column, and "SlotType" and "Slot Type Description" in the "Type Description" column. There are four empty rows below the first row. A plus sign icon is visible in the top right corner of the table area.

Type Id	Type Description
&Update &Delete	SlotType Slot Type Description

- b) SlotType トランザクションのパターンおよび次の図に示されている“modes”ノードのタブを編集し、“Delete” プロパティの値を“False”に変更する。



- c) SlotType トランザクションのパターンおよび “View(Slot Type Information)” ノードのタブを編集してから展開し、次の図に示されている、Tab (General) 内の Action (Delete) プロパティを探し、その値を既定から “False “に変更する。



- d) いずれも正しくない。

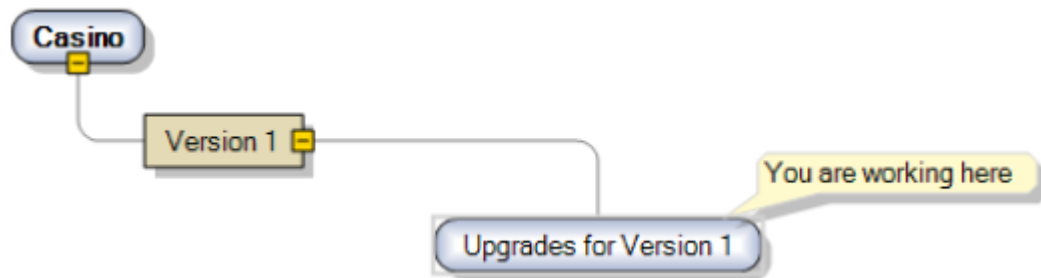
36. [2%]

カジノを対象として GeneXus で開発されたアプリケーションには、次の図で示すように、スロット (Slot) とスロットタイプ (SlotType) を記録するトランザクションがあり、Work With パターンが両方のトランザクションに適用されています。生成された “Work With Slot Types” を次のようにカスタマイズします：

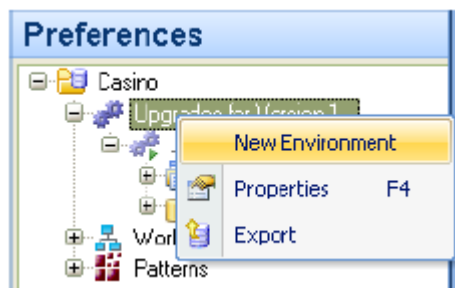
- グリッドの下に MySlotTypeQuery Web パネルを呼び出すボタンを追加する。
- Web パネルは数種類のグラフを実装する動的クエリを表示する。

上記の要件を満たすには、次の図で示すように SlotType トランザクションの Patterns タブに新しいアクションを作成することで十分でしょうか？真（有効）か偽（無効）か選択してください。

- b) Frozen バージョン “Version 1” を作成し、そこからモジュールの開発を開始するために、新しい開発バージョン (“Upgrades for Version 1”) を作成する。



- c) 新しい環境を作成し、それをアクティブとして設定する。その後この環境で新しいモジュールを開発する。



- d) いずれも正しくない。

38. [1%]

カジノを対象として GeneXus で開発されたアプリケーションでは、最終的なプラットフォームはまだ不明なので、.NET 環境と Java 環境が定義されています。

既定の環境は .NET で、そのモジュールのいくつかの開発を続け、すべての定義を Java 環境で更新する必要があります。

上記を前提に次の中から適切なオプションを選択してください。

- a) .NET 環境から Java 環境に移行すると何らかの問題が起きる可能性がある。
- b) すべての変更内容が自動的にすべての環境に反映されるので、何もする必要がない。
- c) Change Defender が使用される。
- d) いずれも正しくない。

模擬試験受験者への試験準備に関するヒント

この模擬試験に含まれるほとんどの主題は、すべてのほかの試験に含まれています。それらは GeneXus に関する基本的な論理なので、ほかの試験はこの試験のバリエーション 1 つと言えるかもしれません。

ただし、この試験には、ほかの試験に含まれる Query オブジェクト、ビルドプロセス、または GXserver に関する主題が含まれていません。