**Download Full Version AZ-203 Exam Dumps(Updated in Feb/2023)**

**QUESTION 4**
You use Azure Table storage to store customer information for an application. The data contains customer details and is partitioned by last name. You need to create a query that returns all customers with the last name Smith. Which code segment should you use?

A. TableQuery.GenerateFilterCondition("PartitionKey", Equals, "Smith")
B. TableQuery.GenerateFilterCondition("LastName", Equals, "Smith")
C. TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "Smith")
D. TableQuery.GenerateFilterCondition("LastName", QueryComparisons.Equal, "Smith")

**Correct Answer:** C
**Explanation:**
Retrieve all entities in a partition. The following code example specifies a filter for entities where 'Smith' is the partition key. This example prints the fields of each entity in the query results to the console.

Construct the query operation for all customer entities where PartitionKey="Smith".

TableQuery<CustomerEntity> query = new
TableQuery<CustomerEntity>().Where(TableQuery.GenerateFilterCondition("PartitionKey", QueryComparisons.Equal, "Smith"));

References:
https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-how-to-use-dotnet


**QUESTION 5**
You are creating a hazard notification system that has a single signaling server which triggers audio and visual alarms to start and stop.

You implement Azure Service Bus to publish alarms. Each alarm controller uses Azure Service Bus to receive alarm signals as part of a transaction. Alarm events must be recorded for audit purposes. Each transaction record must include information about the alarm type that was activated.

You need to implement a reply trail auditing solution.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

A. Assign the value of the hazard message SessionID property to the SequenceNumber property.
B. Assign the value of the hazard message SequenceNumber property to the DeliveryCount property.
C. Assign the value of the hazard message MessageId property to the DeliveryCount property.
D. Assign the value of the hazard message SessionID property to the ReplyToSessionId property.
E. Assign the value of the hazard message MessageId property to the SequenceNumber property.
F. Assign the value of the hazard message MessageId property to the CorrelationId property.

**Correct Answer:** AB


**QUESTION 6**

**AZ-203 Exam Dumps     AZ-203 PDF Dumps     AZ-203 VCE Dumps     AZ-203 Q&As**
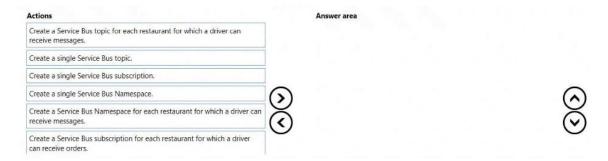**https://www.ensurepass.com/AZ-203.html**

DRAG DROP
You develop software solutions for a mobile delivery service. You are developing a mobile app that users can use to order from a restaurant in their area. The app uses the following workflow:
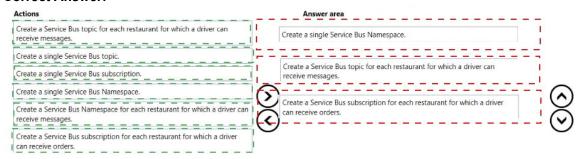
1. A driver selects the restaurants for which they will deliver orders.
2. Orders are sent to all available drivers in an area.
3. Only orders for the selected restaurants will appear for the driver.
4. The first driver to accept an order removes it from the list of available orders.

You need to implement an Azure Service Bus solution.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.



**Correct Answer:**



**QUESTION 7**
HOTSPOT
You are working for a company that designs mobile applications. They maintain a server where player records are assigned to their different games. The tracking system is new and in development.

The application uses Entity Framework to connect to an Azure Database. The database holds a Player table and Game table.

When adding a player, the code should insert a new player record, and add a relationship between an existing game record and the new player record.

The application will call CreatePlayerWithGame with the correct gameIdand the playerId to start the process. (Line numbers are included for reference only.)

```
01. namespace ContosoCradt
02. {
03    public class PlayerDbContext : DbContext
04.    {
05. public PlayerDbContext() : base ("name-dBConnString") { ]
06. public DbSet<Player> Players { get ; set ; }
07. public DbSet<Game> Games { get ; set ]
08. protected override void DmModelCreating{DBModelBuilder modelBuilder}
09. {
10.   modelBuilder.Entity<Player>().MesMany(x => x.Games). WithMany (x => x Players);
11. }
12. }
13.   internal series class dbConfiguration : DbMigrationConfiguration<PlayerDbContext>
14. {
15. public dbConfiguration[] . {AutomaticMigrationsEnabled = true ; }
16. {
17.    public class mp
18.    {
19. public void CreatePlayerWithGame(int playerId, int gameId) => AddPlayer(playerId, GetGame[gameId]);
20. public game GetGame(int gameId)
21. {
22.    using (var db = new PlayerDbContext())
23. {
24.    return db.Games.FirstOrDefault(x => x.GameId == gameId);
25. }
26.    }
27.    public Player AddPlayer (int playerId, Game game)
28.    {
29. using (var db = new PlayerDbContext())
30. {
31.    var player = new Player
32.    {
33.      PlayerId = playerId,
34.      Games = new List <Game> {game },
35. };
36. db.Players.Add(player);
37. db.SaveChanges();
38. return player;
39.    }
40. }
41. public class Player
42. {
43. public int PlayerId { get ; set; }
44. public string PlayerName { get ; set; }
45. public virtual List<Game> Games { get ; set; }
46. }
47. public class Game
48. {
49. public int GameIs { get ; set }
50. public string Title { get ; set; }
51. public string Platform { get ; set; ]
52. public virtual List<Player> Players { get ; set; }
53.    }
54. }
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

|                                                                                          | Yes | No |
|------------------------------------------------------------------------------------------|-----|----|
| The code will successfully insert a player record.                                       | ○   | ○  |
| The code has a bug and will insert an additional copy of the Game record with a new Id.  | ○   | ○  |
| The code has a bug and will insert the wrong gameId value.                                | ○   | ○  |
| There is a valid many-to-many relationship between Players and Games.                     | ○   | ○  |

**Correct Answer:**

|                                                                                          | Yes | No |
|------------------------------------------------------------------------------------------|-----|----|
| The code will successfully insert a player record.                                       | ⊡   | ○  |
| The code has a bug and will insert an additional copy of the Game record with a new Id.  | ○   | ⊡  |
| The code has a bug and will insert the wrong gameId value.                                | ⊡   | ○  |
| There is a valid many-to-many relationship between Players and Games.                     | ○   | ⊡  |

**QUESTION 8**
DRAG DROP
You have an application that provides weather forecasting data to external partners. You use Azure API Management to publish APIs.

You must change the behavior of the API to meet the following requirements:

▪ Support alternative input parameters.
▪ Remove formatting text from responses.
▪ Provide additional context to back-end services.

Which types of policies should you implement? To answer, drag the policy types to the correct scenarios. Each policy type may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content

NOTE: Each correct selection is worth one point.

**Policy types**

| | |
|---|---|
| Inbound | |
| Outbound | |
| Backend | |

**Answer Area**

| Requirement | Policy type |
|---|---|
| Rewrite the request URL to match to the format expected by the web service. | policy type |
| Remove formatting text from responses. | policy type |
| Forward the user ID that is associated with the subscription key for the original request to the back-end service. | policy type |

**Correct Answer:**

**Policy types**

| | |
|---|---|
| Inbound | |
| Outbound | |
| Backend | |

**Answer Area**

| Requirement | Policy type |
|---|---|
| Rewrite the request URL to match to the format expected by the web service. | Outbound |
| Remove formatting text from responses. | Inbound |
| Forward the user ID that is associated with the subscription key for the original request to the back-end service. | Backend |

**QUESTION 9**
Note: This question is part of a series of questions that present the same scenario. Each question in the scries contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result these questions will not appear in the review screen.

Margie's Travel is an international travel and bookings management service. The company is expanding into restaurant bookings. You are tasked with implementing Azure Search for the restaurants listed in their solution.

You create the index in Azure Search.

You need to import the restaurant data into the Azure Search service by using the Azure Search .NET SDK.

Solution:
1. Create a SearchIndexClient object to connect to the search index.
2. Create a DataContainer that contains the documents which must be added.
3. Create a DataSource instance and set its Container property to the DataContamer.
4. Call the Documents.Suggest method of the SearchIndexClient and pass the DataSource.

Does the solution meet the goal?

A. Yes
B. No

**Correct Answer:** B