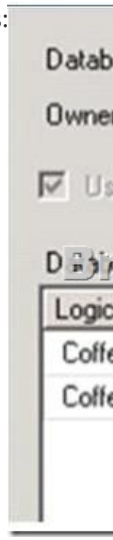
Using Latest Braindump2go 70-464 Exam Dumps Try to Attend 70-464 Exam Pass 100% OR Full Money Back! (156-165)

Do you want to pass Microsoft 70-464 Exam ? If you answered YES, then look no further. Braindump2go offers you the best 70-464 exam questions which cover all core test topics and certification requirements. All REAL questions and answers from Microsoft Exam Center will help you be a 70-464 certified!Vendor: Microsoft Exam Code: 70-464 Exam Name: Developing Microsoft SQL Server 2014 Databases Exam

Case Study 7: Fourth Coffee (Question 156 ~ Question 166)BackgroundCorporate InformationFourth Coffee is global restaurant chain. There are more than 5,000 locations worldwide.Physical LocationsCurrently a server at each location hosts a SQL Server 2012 instance. Each instance contains a database called StoreTransactions that stores all transactions from point of sale and uploads summary batches nightly.Each server belongs to the COFFECORP domain. Local computer accounts access the StoreTransactions database at each store using sysadmin and datareaderwriter roles.Planned changesFourth Coffee has three major initiatives:The FT department must consolidate the point of sales database infrastructure. The marketing department plans to launch a mobile application for micropayments. The finance department wants to deploy an internal tool that will help detect fraud.Initially, the mobile application will allow customers to make micropayments to buy coffee and other items on the company web site. These micropayments may be sent as gifts to other users and redeemed within an hour of ownership transfer. Later versions will generate profiles based on customer activity that will push texts and ads generated by an analytics application.When the consolidation is finished and the mobile application is in production, the micropayments and point of sale transactions will use the same database. Existing EnvironmentExisting Application EnvironmentSome stores have been using several pilot versions of the micropayment application. Each version currently is in a database that is independent from the point of sales systems. Some versions have been used in field tests at local stores, and others are hosted at corporate servers. All pilot versions were developed by using SQL Server 2012.Existing Support InfrastructureThe proposed database for consolidating micropayments and transactions is called CoffeeTransactions. The database is hosted on a SQL Server 2014 Enterprise Edition instance and has the following file structures:



Business RequirementsGeneral Application Solution RequirementsThe database infrastructure must support a phased global rollout of the micropayment application and consolidation.The consolidated micropayment and point of sales database will be into a CoffeeTransactions database. The infrastructure also will include a new CoffeeAnalytics database for reporting on content from CoffeeTransactions.Mobile applications will interact most frequently with the micropayment database for the following activities:- Retrieving the current status of a micropayment;- Modifying the status of the current micropayment;- Canceling the micropayment. The mobile application will need to meet the following requirements:- Communicate with web services that assign a new user to a

micropayment by using a stored procedure named usp_AssignUser.- Update the location of the user by using a stored procedure named usp_AddMobileLocation.The fraud detection service will need to meet the following requirements:- Query the current open micropayments for users who own multiple micropayments by using a stored procedure named usp.LookupConcurrentUsers. - Persist the current user locations by using a stored procedure named usp_Mobilel_ocationSnapshot.- Look at the status of micropayments and mark micropayments for internal investigations.- Move micropayments to dbo.POSException table by using a stored procedure named ups_DetectSuspiciousActivity.- Detect micropayments that are flagged with a StatusId value that is greater than 3 and that occurred within the last minute.The CoffeeAnalytics database will combine imports of the POSTransaction and MobileLocation tables to create a UserActivity table for reports on the trends in activity. Queries against the UserActivity table will include aggregated calculations on all columns that are not used in filters or groupings.Micropayments need to be updated and queried for only a week after their creation by the mobile application or fraud detection services.PerformanceThe most critical performance requirement is keeping the response time for any queries of the POSTransaction table predictable and fast.Web service queries will take a higher priority in performance tuning decisions over the fraud detection agent queries.ScalabilityQueries of the user of a micropayment cannot return while the micropayment is being updated, but can show different users during different stages of the transaction.The fraud detection service frequently will run queries over the micropayments that occur over different time periods that range between 30 seconds and ten minutes.The POSTransaction table must have its structure optimized for hundreds of thousands of active micropayments that are updated frequently.All changes to the POSTransaction table will require testing in order to confirm the expected throughput that will support the first year's performance requirements.Updates of a user's location can tolerate some data loss.Initial testing has determined that the POSTransaction and POSException tables will be migrated to an in-memory optimized table.AvailabilityIn order to minimize disruption at local stores during consolidation, nightly processes will restore the databases to a staging server at corporate headquarters.Technical RequirementsSecurityThe sensitive nature of financial transactions in the store databases requires certification of the COFFECORPAuditors group at corporate that will perform audits of the data. Members of the COFFECORPAuditors group cannot have sysadmin or datawriter access to the database.Compliance requires that the data stewards have access to any restored StoreTransactions database without changing any security settings at a database level.Nightly batch processes are run by the services account in the COFFECORPStoreAgent group and need to be able to restore and verify the schema of the store databases match.No Windows group should have more access to store databases than is necessary.MaintainabilityYou need to anticipate when POSTransaction table will need index maintenance.When the daily maintenance finishes, micropayments that are one week old must be available for queries in UserActivity table but will be queried most frequently within their first week and will require support for in-memory queries for data within first week.The maintenance of the UserActivity table must allow frequent maintenance on the day's most recent activities with minimal impact on the use of disk space and the resources available to queries. The processes that add data to the UserActivity table must be able to update data from any time period, even while maintenance is running.The index maintenance strategy for the UserActivity table must provide the optimal structure for both maintainability and query performance.All micropayments queries must include the most permissive isolation level available for the maximum throughput.In the event of unexpected results, all stored procedures must provide error messages in text message to the calling web service.Any modifications to stored procedures will require the minimal amount of schema changes necessary to increase the performance.PerformanceStress testing of the mobile application on the proposed CoffeeTransactions database uncovered performance bottlenecks. The sys.dm_os_wait_stats Dynamic Management View (DMV) shows high wait_time values for WRTTELOG and PAGEIOLATCHJJP wait types when updating the MobileLocation table. Updates to the MobileLocation table must have minimal impact on physical resources.Supporting InfrastructureThe stored procedure usp_LookupConcurrentUsers has the current implementation:

```
CREATE PROCEDURE usp_LookupConcurrentUsers
AS BEGIN
  --summary table
  CREATE TABLE #POSTransactionTemp (
  POSTransactionId int NOT NULL,
  UserId int NOT NULL,
  StatusID int NOT NULL,
  POSLocation int NOT NULL,
  CreateDate datetime2 NOT NULL,
  Price money
  )
  DECLARE @timewindow datetime2
  SET @timewindow = GETDATE();

  WITH concurrentusers
  AS
  (
  SELECT UserId, COUNT(*) concurrentsessions
  FROM dbo.POSTransaction
  WHERE CreateDate >= dateadd(second,-60, @timewindow )
  GROUP BY UserId
  HAVING COUNT(*) > 1
  )
  INSERT INTO #POSTransactionTemp
  (
POSTransactionId, UserId,
StatusID, POSLocation,
CreateDate, Price
  )

  SELECT d.*
  FROM dbo.POSTransaction d
  JOIN concurrentusers c
  on d.UserID = c.UserId
  WHERE d.CreateDate >= dateadd(second,-60, @timewindow )
  ...
  SELECT * FROM #POSTransactionTemp

  END
```


The current stored procedure for persisting a user location is defined in the following code:

```
CREATE PROCEDURE dbo.usp_MobileLocationSnapsh
AS
BEGIN

INSERT INTO CoffeeAnalytics.dbo.MobileLocatio
SELECT * FROM CoffeeTransactions.dbo.MobileLo

END
```

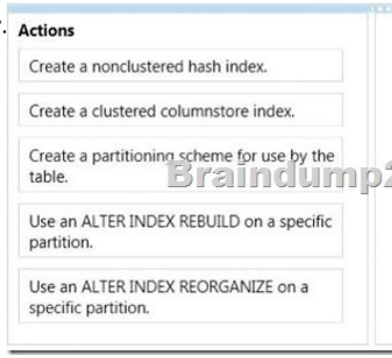

The current stored procedure for managing micropayments needing investigation is defined in the following code:

```
01  CREATE PROCEDURE dbo.
02  WITH NATIVE_COMPILATI
03  AS
04  BEGIN ATOMIC
05  WITH (TRANSACTION ISO
06  LANGUAGE = 'us_englis
07  IF EXISTS(SELECT POST
08  WHERE StatusID >= 4 a
09  GETDATE() ))
10  MERGE dbo.POSExceptio
11  USING (SELECT POSTran
12  POSLocation, CreateDa
13  WHERE StatusID >= 4 a
14  CreateDate = GETDATE(
15  AS source (POSTransac
16  POSLocation, CreateDa
17  ON (target.POSTransac
18  WHEN MATCHED THEN
19  UPDATE SET StatusID =
20  WHEN NOT MATCHED THEN
21  INSERT (POSTransactio
22  POSLocation, CreateDa
23  VALUES (source.POSTra
24  source.UserId, source
25  source.CreateDate, so
26  END
```

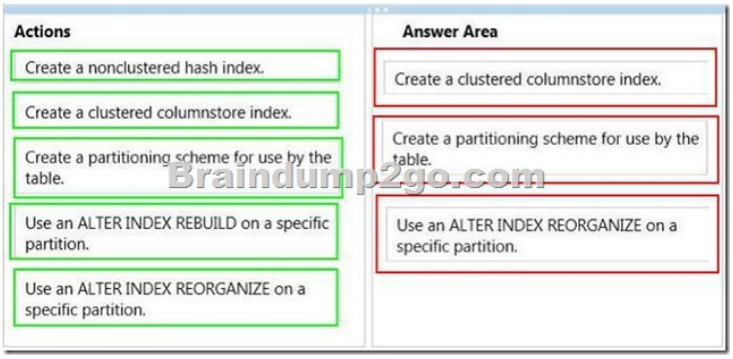
The current table, before implementing any performance enhancements, is defined as follows:

```
CREATE TABLE dbo.POSTransaction (
  POSTransactionId int NOT NULL PRIMARY KEY,
  UserId int NOT NULL,
  POSLocation int NOT NULL,
  StatusCode int NOT NULL,
  CreateDate datetime2 NOT NULL,
  Price money
  )
CREATE INDEX ix_UserID on dbo.POSTransacti
```


QUESTION 156You need to monitor the health of your tables and indexes in order to implement the required index maintenance strategy.What should you do? A. Query system DMVs to monitor avg_chain_length and max_chain_length. Create alerts to notify you when these values converge.B. Create a SQL Agent alert when the File Table: Avg time per file I/O request value is increasing.C. Query system DMVs to monitor total_bucket_count. Create alerts to notify you when this value increases.D. Query system DMVs to monitor total_bucket_count. Create alerts to notify you when this value decreases. Answer: AExplanation:From scenario:- You need to anticipate when POSTransaction table will need index maintenance.- The index maintenance strategy for the UserActivity table must provide the optimal structure for both maintainability and query performance. QUESTION 157Drag and Drop QuestionYou need to design the UserActivity table.Which three steps should you perform in sequence? To answer, move the appropriate three actions from the list of actions to the answer area and arrange them in the correct order.



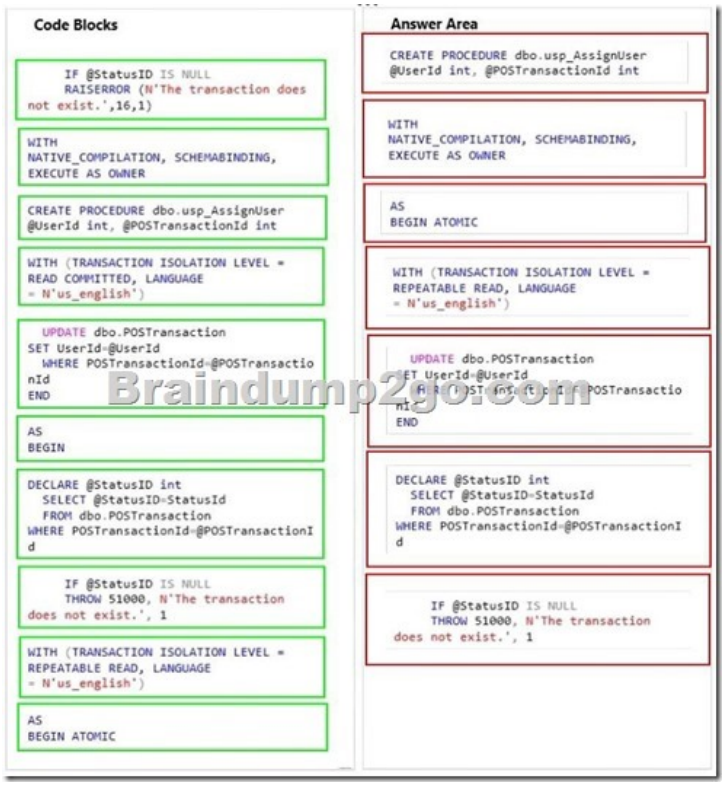
Answer:



QUESTION 158You need to implement security for the restore and audit process. What should you do? A. Grant the COFFECORPAuditors group ALTER ANY CONNECTION and SELECT ALL USER SECURABLES permissions. Grant the COFFECORPStoreAgent group ALTER ANY CONNECTION and IMPERSONATE ANY LOGIN permissions.B. Grant the COFFECORPAuditors group CONNECT ANY DATABASE and IMPERSONATE ANY LOGIN permissions. Grant the COFFECORPStoreAgent group CONNECT ANY DATABASE and SELECT ALL USER SECURABLES permissions.C. Grant the COFFECORPAuditors group ALTER ANY CONNECTION and IMPERSONATE ANY LOGIN permissions. Grant the COFFECORPStoreAgent group ALTER ANY CONNECTION and SELECT ALL USER SECURABLES permissions.D. Grant the COFFECORPAuditors group CONNECT ANY DATABASE and SELECT ALL USER SECURABLES permissions. Grant the COFFECORPStoreAgent group CONNECT ANY DATABASE and IMPERSONATE ANY LOGIN permissions. Answer: A QUESTION 159You need to modify the stored procedure usp_LookupConcurrentUsers.What should you do? A. Add a clustered index to the summary table.B. Add a nonclustered index to the summary table.C. Add a clustered columnstore index to the summary table.D. Use a table variable instead of the summary table. Answer: AExplanation: Scenario: Query the current open micropayments for users who own multiple micropayments by using a stored procedure named usp.LookupConcurrentUsers QUESTION 160Drag and Drop QuestionYou need to create the usp.AssignUser stored procedure.Develop the solution by selecting

and arranging the required code blocks in the correct order. You may not need all of the code blocks.

Answer:



QUESTION 161Drag and Drop QuestionYou need to implement a new version of usp_AddMobileLocation. Develop the solution by selecting and arranging the required code blocks in the correct order. You may not need all of the code blocks.

Answer:

QUESTION 162You need to modify the usp_DetectSuspiciousActivity stored procedure.Which two actions should you perform? Each correct answer presents part of the solution.Choose two.

A. Replace lines 04-06 with the following code:

```
BEGIN ATOMIC WITH
     (
          DELAYED_DURABILITY = ON,
          TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED,
          LANGUAGE = N'English'
     )
```

B. Replace lines 04-06 with the following code:

```
BEGIN ATOMIC WITH
     (
          DELAYED_DURABILITY = ON,
          TRANSACTION ISOLATION LEVEL = REPEATABLE READ
     )
```

C. Change the logic of the stored procedure to use separate UPDATE and INSERT statements.

D. Replace lines 07-09 with the following code:

```
DECLARE @exists BIT = 0
IF EXISTS ( SELECT TOP 1 * FROM POSTransaction (NOLOCK) WHERE StatusID = 4 and CreateDate
>= dateadd(second,-60, GETDATE() ))
```

E. Replace lines 04-06 with the following code:

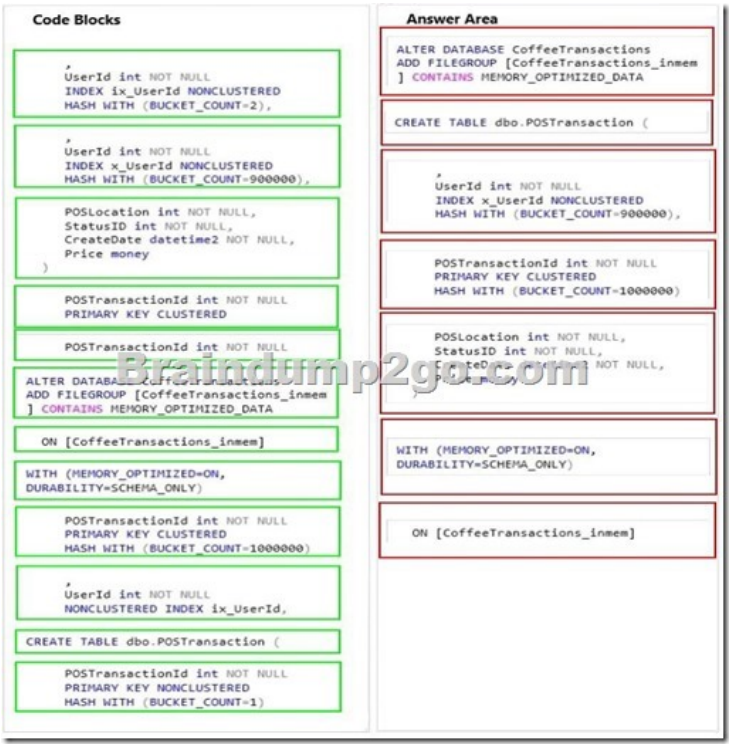
```
BEGIN ATOMIC WITH
     (
          TRANSACTION ISOLATION LEVEL = READ UNCOMMITTED,
          LANGUAGE = N'English'
     )
```

F. Replace lines 07-09 with the following code:

```
DECLARE @exists BIT = 0
SELECT TOP 1 @exists = 1 FROM POSTransaction WHERE StatusID >= 4 and CreateDate >= dateadd
(second,-60, GETDATE() )
IF @exists = 1
```

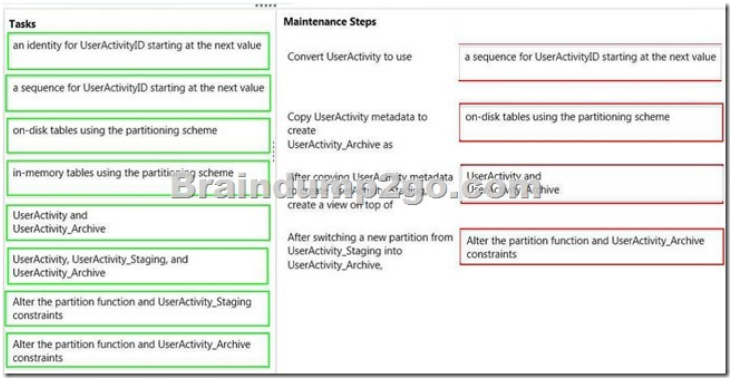

A. Option AB. Option BC. Option CD. Option DE. Option EF. Option F Answer: DEExplanation:Note:- Move micropayments to dbo.POSException table by using a stored procedure named ups_DetectSuspiciousActivity. QUESTION 163Drag and Drop QuestionYou need to redesign the system to meet the scalability requirements of the application.Develop the solution by selecting and arranging the required code blocks in the correct order. You may not need all of the code blocks.

Answer:



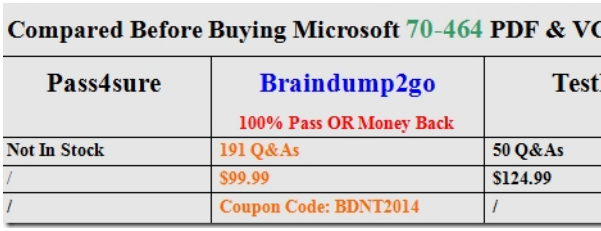
QUESTION 164Drag and Drop QuestionYou need to optimize the index and table structures for POSTransaction.Which task should you use with each maintenance step? To answer, drag the appropriate tasks to the correct maintenance steps. Each task may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

Answer:



QUESTION 165You need to optimize the index structure that is used by the tables that support the fraud detection services.What should you do? A. Add a hashed nonclustered index to CreateDate.B. Add a not hash nonclustered index to CreateDate.C. Add a not hash clustered index on POSTransactionId and CreateDate.D. Add a hashed clustered index on POSTransactionId and CreateDate. Answer: AExplanation: The fraud detection service will need to meet the following requirement (among others):- Detect micropayments that are flagged with a StatusId value that is greater than 3 and that occurred within the last minute.

Braindump2go Promise All 70-464 Questions and Answers are the Latest Updated,we aim to provide latest and guaranteed questions for all certifications.You just need to be braved in trying then we will help you arrange all left things! 100% Pass All Exams you want Or Full Money Back! Do yo want to have a try on passing 70-464?



Compared Before Buying Microsoft 70-464 PDF & VC

Pass4sure	Braindump2go 100% Pass OR Money Back	Testi
Not In Stock	191 Q&As	50 Q&As
/	$99.99	$124.99
/	Coupon Code: BDNT2014	/

http://www.braindump2go.com/70-464.html