70-433 Q&As

TS: Microsoft SQL Server 2008, Database Development

Pass Microsoft 70-433 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.pass4lead.com/70-433.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

QUESTION 1

You administer a database named Contoso running on a Microsoft SQL Server 2008 R2 instance.

You plan to implement custom error handling in your application.

You need to implement custom error handling that meets the following requirements:

The custom message is a reusable user-defined error message. The message returned to the application is an informational message that returns status

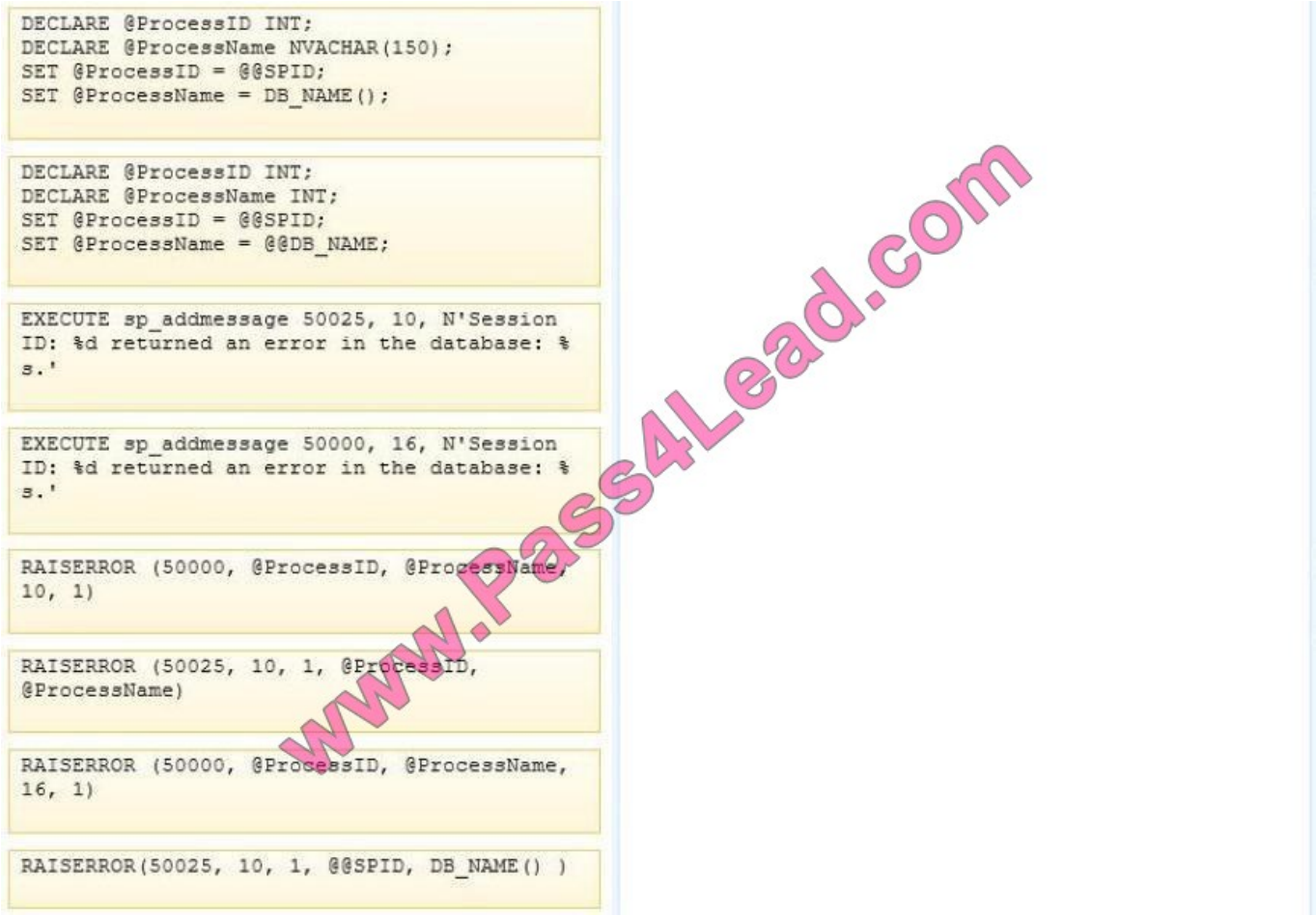
information or error that is not severe.

The custom message returned indicates that an error has occurred in the current database and current session.

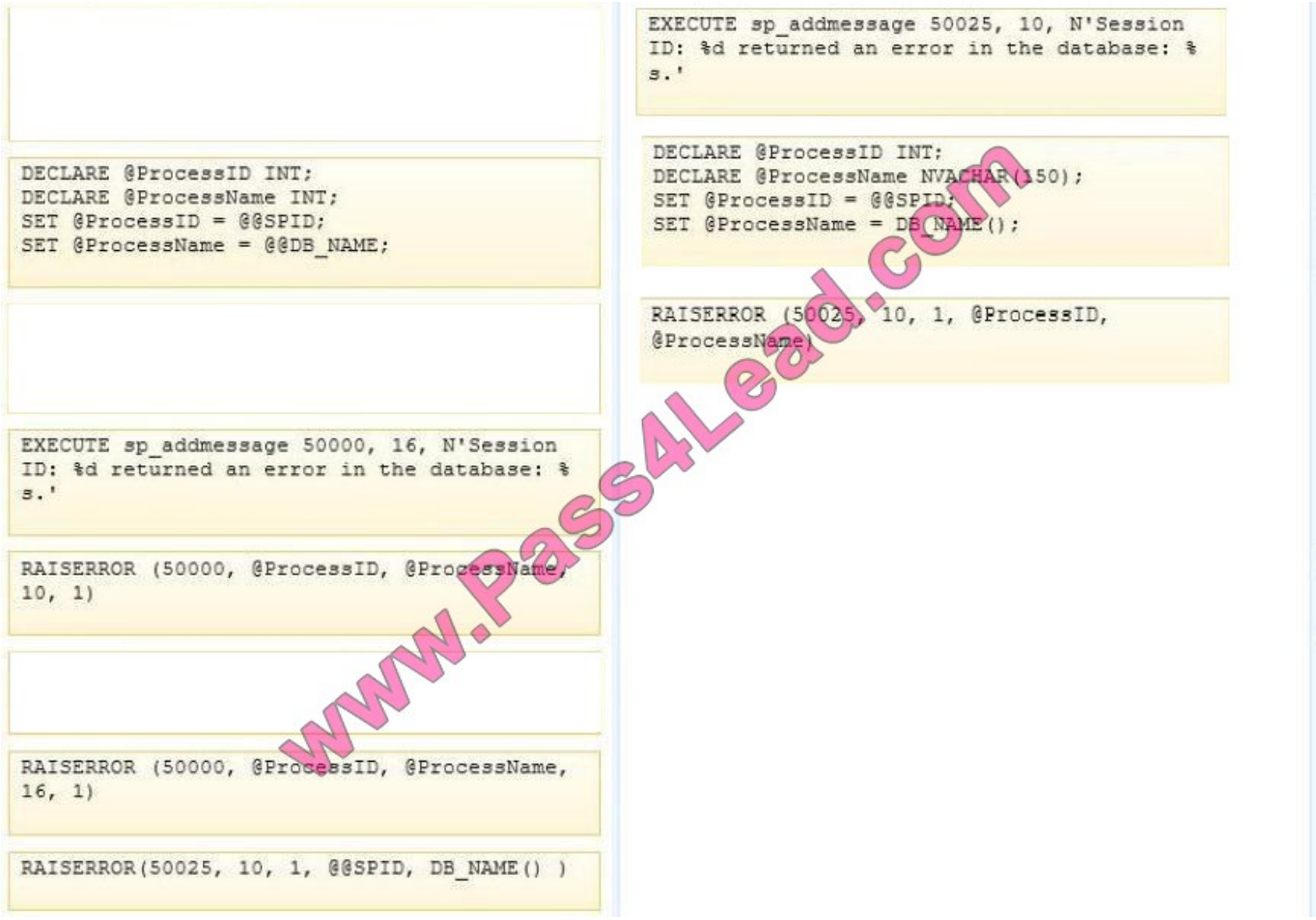
Which three Transact-SQL statements should you use? (To answer, move the appropriate statements from the list of statements to the answer area and arrange

them in the correct order.)

Select and Place:



Correct Answer:



QUESTION 2

You administer a Microsoft SQL Server 2008 R2 instance configured to use Windows Authentication. The database contains a table named CustomerTransaction that has the following definition:

```
CREATE TABLE dbo.CustomerTransaction
(CustomerTransactionId int NOT NULL PRIMARY KEY,
 CustomerID int NOT NULL,
 TransactionAmount money NOT NULL )
```

You define the following table:

```
CREATE TABLE dbo.CustomerWarningLog(
  CustomerWarningLogId int NOT NULL identity PRIMARY KEY,
  CustomerId int NOT NULL,
  Balance money NOT NULL,
  LogTime datetime2(0) NOT NULL,
  LogUserName nvarchar(128) NOT NULL)
```

You need to ensure that the following requirements are met:

An entry is logged in CustomerWarningLog when a customer\\'s account balance is less than 100.00. The TransactionLogUserName is set to the login name of the user who modifies the CustomerTransaction table.

Which Transact-SQL statement or statements should you use?

A.
```
CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateTrigger
ON dbo.CustomerTransaction
FOR INSERT, UPDATE
AS

IF UPDATE(CustomerID) or UPDATE(TransactionAmount)
  WITH TotalCTE AS
    (SELECT CustomerId, SUM(TransactionAmount) AS Balance
    FROM dbo.CustomerTransaction
    WHERE CustomerId in (SELECT CustomerId
      FROM inserted
      UNION ALL
      SELECT CustomerId
      FROM deleted)
    GROUP BY CustomerId)
    INSERT dbo. CustomerWarningLog(
      CustomerId, Balance, LogTime, LogUserName)
    SELECT CustomerId, Balance, SYSDATETIME(), USER_NAME()
    FROM TotalForCustomer
    WHERE TotalCTE.Amount < 100

GO
```

B.
```
CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTrigger
ON dbo.CustomerTransaction
FOR INSERT, UPDATE, DELETE
AS

  WITH TotalCTE AS
  (SELECT CustomerId, SUM(TransactionAmount) AS Balance
  FROM dbo.CustomerTransaction
    WHERE CustomerId in (SELECT CustomerId
      FROM inserted
    )
    GROUP BY CustomerId)
    INSERT dbo. CustomerWarningLog(
      CustomerId, Balance, LogTime, LogUserName)
    SELECT CustomerId, Balance, SYSDATETIME(), SUSER_SNAME()
    FROM TotalForCustomer
    WHERE TotalCTE.Amount <= 100

GO
```

A. B.

C.
```
CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTrigger
ON dbo.CustomerTransaction
FOR INSERT, UPDATE, DELETE
AS

  DECLARE @CustomerId int, @Balance money

  WITH TotalCTE AS
  (SELECT CustomerId, SUM(TransactionAmount) AS Balance
  FROM dbo.CustomerTransaction
  WHERE CustomerId in (SELECT CustomerId
    FROM inserted
    UNION ALL
    SELECT CustomerId
    FROM deleted)
  GROUP BY CustomerId)

  SELECT @CustomerId = CustomerId,
    @Balance = TransactionAmount
  FROM TotalForCustomer
  WHERE TotalCTE.Amount < 100

  IF @CustomerId IS NULL
  INSERT dbo. CustomerWarningLog(
    CustomerId, Balance, LogTime, LogUserName)
  SELECT @CustomerId, @Balance, SYSDATETIME(), USER_NAME()

GO
```

D.
```
CREATE TRIGGER dbo.CustomerTransaction_InsertUpdateDeleteTrigger
ON dbo.CustomerTransaction
FOR INSERT, UPDATE, DELETE
AS

IF UPDATE(CustomerID) or UPDATE(TransactionAmount)
  WITH TotalCTE AS
  (SELECT CustomerId, SUM(TransactionAmount) AS Balance
  FROM dbo.CustomerTransaction
  WHERE CustomerId in (SELECT CustomerId
    FROM inserted
    UNION ALL
    SELECT CustomerId
    FROM deleted)
    GROUP BY CustomerId)
    INSERT dbo. CustomerWarningLog(
      CustomerId, Balance, LogTime, LogUserName)
    SELECT CustomerId, Balance, SYSDATETIME(), SUSER_SNAME()
    FROM TotalForCustomer
    WHERE TotalCTE.Amount < 100

GO
```

C. D.

Correct Answer: D

QUESTION 3

You administer a Microsoft SQL Server 2008 database that contains a table named dbo.[order].

There are no triggers on the table. You plan to create a stored procedure that will have the following parameters:

@ProdId int

@CustId int

You need to ensure that the following requirements are met:

The OrderID and ProdID values of each modified row are captured into a local table variable before data is modified.

The ProdID is modified to @ProdID where CustID is equal to @CustId.

Which Transact-SQL statement should you use?

A.
```
DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);

UPDATE [order]
SET
  ProdID = @CustId
OUTPUT #INSERTED.OrderID, #INSERTED.ProdID INTO @OrderIDs
WHERE
  CustID = @CustId;
```

B.
```
DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);

UPDATE dbo.[order]
SET
  ProdID = @ProdId
OUTPUT INSERTED.OrderID, INSERTED.ProdID INTO @OrderIDs
WHERE
  CustID = @CustId;
```

C.
```
DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);

UPDATE dbo.[order]
SET
  ProdID = @ProdId
OUTPUT DELETED.OrderID, DELETED.ProdID INTO @OrderIDs
WHERE
  CustID = @CustId;
```

D.
```
DECLARE @OrderIDs TABLE (OrderID INT, ProdID INT);

UPDATE dbo.[order]
SET
  ProdID = @ProdId
OUTPUT SELECT d.OrderID, d.ProdID FROM DELETED d INTO @OrderIDs
WHERE
  CustID = @CustId;
```


A. B. C. D.

Correct Answer: A

QUESTION 4

You create and populate two tables by using the following Transact-SQL statements:

CREATE TABLE CurrentStudents (LastName VARCHAR(50), FirstName VARCHAR(50),

Address VARCHAR(100),

Age INT);

INSERT INTO CurrentStudents

VALUES (\\'Fritz\\', \\'David\\', \\'181 Kline Street\\', 14)

,(\\'Reese\\', \\'Paul\\' , \\'4429 South Union\\', 14)

,(\\'Brown\\', \\'Jake\\' , \\'5401 Washington Ave\\',14)

,(\\'Smith\\', \\'Tom\\' , \\'124 Water St\\', 14)

,(\\'Holtz\\', \\'Mary\\' , \\'984 Mass Ct\\', 14)

,(\\'Robbins\\', \\'Jan\\' , \\'4449 Union Ave\\', 14)

,(\\'Larsen\\', \\'Frank\\' , \\'5812 Meadow St\\', 14)

,(\\'Bishop\\', \\'Cathy\\' , \\'14429 Skyhigh Ave\\', 14)

,(\\'Francis\\', \\'Thomas\\' , \\'15401 120th St\\', 14)

CREATE TABLE NewYearRoster(LastName VARCHAR(50),

FirstName VARCHAR(50),

Address VARCHAR(100),

Age INT);

INSERT INTO NewYearRoster

VALUES (\\'Fritz\\', \\'David\\', \\'181 Kline Street\\', 15)

,(\\'Reese\\', \\'Paul\\', \\'1950 Grandview Place\\', 15)

,(\\'Adams\\', \\'Wilbur\\', \\'4231 W. 93rd\\', 15)

,(\\'Adams\\', \\'Norris\\', \\'100 1st Ave\\', 15)

,(\\'Thomas\\', \\'Paul\\', \\'18176 Soundview Dr\\', 15)

,(\\'Linderson\\', \\'Danielle\\', \\'941 W. 37 Ave\\', 15)

,(\\'Moore\\', \\'Joshua\\', \\'2311 10st Ave\\', 15)

,(\\'Dark\\', \\'Shelby\\', \\'1987 Fifth Ave\\', 15)

,(\\'Scharp\\', \\'Mary\\', \\'1902 W. 303rd\\', 15)

,(\\'Morris\\', \\'Walt\\', \\'100 12st St\\', 15);

You run the following MERGE statement to update, insert and delete rows in the CurrentStudents table:

MERGE TOP (3) CurrentStudents AS T

USING NewYearRoster AS S

ON S.LastName = T.LastName AND S.FirstName = T.FirstName WHEN MATCHED AND NOT (T.Age = S.Age OR T.Address = S.Address) THEN UPDATE SET Address = S.Address,

Age = S.Age

WHEN NOT MATCHED BY TARGET THEN

INSERT (LastName, FirstName, Address, Age)

VALUES (S.LastName, S.FirstName, S.Address, S.Age) WHEN NOT MATCHED BY SOURCE THEN

DELETE;

You need to identify the total number of rows that are updated, inserted, and deleted in the CurrentStudent table.

Which total number of rows should you choose?

A. 0

B. 3

C. 6

D. 9

Correct Answer: B

QUESTION 5

You are a developer for a Microsoft SQL Server 2008 R2 database instance used to support a customer service application. You create tables named complaint, customer, and product as follows:

```
CREATE TABLE [dbo].[complaint]
 ([ComplaintID] [int],
  [ProductID] [int],
  [CustomerID] [int],
  [ComplaintDate] [datetime]);

CREATE TABLE [dbo].[customer]
 ([CustomerID] [int],
  [CustomerName] [varchar](100),
  [Address] [varchar](200),
  [City] [varchar](100),
  [State] [varchar](50),
  [ZipCode] [varchar](5));

CREATE TABLE [dbo].[product]
 ([ProductID] [int],
  [ProductName] [varchar](100),
  [SalePrice] [money],
  [ManufacturerName] [varchar](100));
```


You need to write a query to sum the sales made to each customer who has made a complaint by the following entries: Each customer name Each product name The grand total of all sales

Which SQL query should you use?

A.
```
SELECT
  c.CustomerName,
  p.ProductName,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID INNER JOIN
  customer c ON com.CustomerID = c.CustomerID
GROUP BY GROUPING SETS ((c.CustomerName, p.ProductName), ());
```

B.
```
SELECT
  c.CustomerName,
  p.ProductName,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID INNER JOIN
  customer c ON com.CustomerID = c.CustomerID
GROUP BY GROUPING SETS ((c.CustomerName), (p.ProductName), ());
```

C.
```
SELECT
  c.CustomerName,
  COUNT(com.ComplaintID) AS Complaints
FROM
  customer c INNER JOIN
  complaint com ON c.CustomerID = com.CustomerID
WHERE
  COUNT(com.ComplaintID) > 10
GROUP BY
  c.CustomerName;
```

D.
```
SELECT
  c.CustomerName,
  COUNT(com.ComplaintID) AS complaints
FROM
  customer c INNER JOIN
  complaint com ON c.CustomerID = com.CustomerID
GROUP BY
  c.CustomerName
HAVING
  COUNT(com.ComplaintID) > 10;
```

E.
```
SELECT
  c.CustomerName,
  AVG(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID INNER JOIN
  customer c ON com.CustomerID = c.CustomerID
WHERE
```

F.
```
SELECT
  c.CustomerName,
  AVG(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID INNER JOIN
  customer c ON com.CustomerID = c.CustomerID
WHERE
  com.ComplaintDate > '09/01/2011' AND
  AVG(p.SalePrice) >= 500
```

G.
```
SELECT
  p.ProductName,
  DATEPART(mm, com.ComplaintDate) ComplaintMonth,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID
GROUP BY CUBE(p.ProductName, DATEPART(mm, com.ComplaintDate));
```

H.
```
SELECT
  p.ProductName,
  DATEPART(mm, com.ComplaintDate) ComplaintMonth,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID
GROUP BY CUBE;
```

I.
```
SELECT
  p.ProductName,
  DATEPART(mm, com.ComplaintDate) ComplaintMonth,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID
GROUP BY p.ProductName, ComplaintMonth;
```

J.
```
SELECT
  p.ProductName,
  DATEPART(mm, com.ComplaintDate) ComplaintMonth,
  SUM(p.SalePrice) AS Sales
FROM
  product p INNER JOIN
  complaint com ON p.ProductID = com.ProductID
GROUP BY p.ProductName, DATEPART(mm, com.ComplaintDate);
```

A. B. C. D. E. F. G. H. I. J.

Correct Answer: B

To Read the Whole Q&As, please purchase the Complete Version from Our website.

Try our product !

100% Guaranteed Success
100% Money Back Guarantee
365 Days Free Update
Instant Download After Purchase
24x7 Customer Support
Average 99.9% Success Rate
More than 800,000 Satisfied Customers Worldwide
Multi-Platform capabilities - Windows, Mac, Android, iPhone, iPod, iPad, Kindle

We provide exam PDF and VCE of Cisco, Microsoft, IBM, CompTIA, Oracle and other IT Certifications.
You can view Vendor list of All Certification Exams offered:

https://www.pass4lead.com/allproducts

Need Help

Please provide as much detail as possible so we can best assist you.
To update a previously submitted ticket: