

Easy CramBible Lab



70-433

**TS:Microsoft SQL Server 2008,
Database Development**

** Single-user License **

This copy can be only used by yourself for educational purposes.

Web: <http://www.crambible.com/>

E-mail: web@crambible.com

Important Note**Please Read Carefully****Study Tips**

This product will provide you questions and answers along with carefully compiled and written by our experts. Try to understand the concepts behind the questions instead of cramming the questions.

Go through the entire document at least twice so that you make sure that you are not missing anything.

Latest Version

We are constantly reviewing our products. New material is added and old material is revised. Free updates are available for 90 days after the purchase. You should check your member zone at CramBible an update 3-4 days before the scheduled exam date.

Here is the procedure to get the latest version:

1. Go to www.CramBible.com
2. Click on Member zone/Log in
3. The latest versions of all purchased products are download from here. Just click the links.

For most updates, it is enough just to print the new questions at the end of the new version, not the whole document.

Feedback

Feedback on specific questions should be send to web@CramBible.com. You should state: Exam number and version, question number, and login ID.

Our experts will answer your mail promptly.

Copyright

Each pdf file contains a unique serial number associated with your particular name and contact information for security purposes. So if we find out that a particular pdf file is being distributed by you, CramBible reserves the right to take legal action against you according to the International Copyright Laws.

THE TOTAL NUMBER OF QUESTIONS IS 158**QUESTION NO: 1**

You have a user named John. He has SELECT access to the Sales schema. You need to eliminate John's SELECT access rights from the Sales.SalesOrder table without affecting his other permissions. Which Transact-SQL statement should you use?

- A. DROP USER John;
- B. DENY SELECT ON Sales.SalesOrder TO John;
- C. GRANT DELETE ON Sales.SalesOrder TO John;
- D. REVOKE SELECT ON Sales.SalesOrder FROM John;

Answer: B

QUESTION NO: 2

You need to create a column that allows you to create a unique constraint. Which two column definitions should you choose? (Each correct answer presents a complete solution. Choosetwo.)

- A. nvarchar(100) NULL
- B. nvarchar(max) NOT NULL
- C. nvarchar(100) NOT NULL
- D. nvarchar(100) SPARSE NULL

Answer: AC

QUESTION NO: 3

You manage a SQL Server 2008 database that is located at your company's corporate headquarters. The database contains a table named dbo.Sales. You need to create different views of the dbo.Sales table that will be used by each region to insert, update, and delete rows. Each regional office must only be able to insert, update, and delete rows for their respective region. Which view should you create for Region1?

- A. CREATE VIEW dbo.Region1Sales AS SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;
- B. CREATE VIEW dbo.Region1Sales AS SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1 WITH CHECK OPTION;
- C. CREATE VIEW dbo.Region1Sales WITH SCHEMABINDING AS SELECT SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;

D. CREATE VIEW dbo.Region1Sales WITH VIEW_METADATA AS SELECT
SalesID,OrderQty,SalespersonID,RegionID FROM dbo.Sales WHERE RegionID = 1;

Answer: B

QUESTION NO: 4

You administer a SQL Server 2008 database that contains a table name dbo.Sales, which contains the following table definition:

```
CREATE TABLE [dbo].[Sales](  
[SalesID] [int] IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
[OrderDate] [datetime] NOT NULL,  
[CustomerID] [int] NOT NULL,  
[SalesPersonID] [int] NULL,  
[CommentDate] [date] NULL);
```

This table contains millions of orders. You run the following query to determine when sales persons comment in the dbo.Sales table:

```
SELECT SalesID,CustomerID,SalesPersonID,CommentDate FROM dbo.Sales  
WHERE CommentDate IS NOT NULL AND SalesPersonID IS NOT NULL;
```

You discover that this query runs slow. After examining the data, you find only 1% of rows have comment dates and the SalesPersonID is null on 10% of the rows. You need to create an index to optimize the query. The index must conserve disk space while optimizing your query. Which index should you create?

- A. CREATE NONCLUSTERED INDEX idx1 ON dbo.Sales (CustomerID) INCLUDE (CommentDate,SalesPersonID);
- B. CREATE NONCLUSTERED INDEX idx1 ON dbo.Sales (SalesPersonID) INCLUDE (CommentDate,CustomerID);
- C. CREATE NONCLUSTERED INDEX idx1 ON dbo.Sales (CustomerID) INCLUDE(CommentDate) WHERE SalesPersonID IS NOT NULL;
- D. CREATE NONCLUSTERED INDEX idx1 ON dbo.Sales (CommentDate, SalesPersonID) INCLUDE(CustomerID) WHERE CommentDate IS NOT NULL;

Answer: D

QUESTION NO: 5

Your database is 5GB and contains a table named SalesHistory. Sales information is frequently inserted and updated. You discover that excessive page splitting is occurring. You need to reduce the occurrence of page splitting in the SalesHistory table. Which code segment should you use?.

- A. ALTER DATABASE Sales MODIFY FILE
(NAME = Salesdat3, SIZE = 10GB);
- B. ALTER INDEX ALL ON Sales.SalesHistory REBUILD WITH (FILLFACTOR = 60);
- C. EXEC sys.sp_configure 'fill factor (%)', '60';
- D. UPDATE STATISTICS Sales.SalesHistory(Products) WITH
FULLSCAN, NORECOMPUTE;

Answer: B

QUESTION NO: 6

You have a table named dbo.Customers. The table was created by using the following Transact-SQL statement:

```
CREATE TABLE dbo.Customers
(
  CustomerID int IDENTITY(1,1) PRIMARY KEY CLUSTERED,
  AccountNumber nvarchar(25) NOT NULL,
  FirstName nvarchar(50) NOT NULL,
  LastName nvarchar(50) NOT NULL,
  AddressLine1 nvarchar(255) NOT NULL,
  AddressLine2 nvarchar(255) NOT NULL,
  City nvarchar(50) NOT NULL,
  StateProvince nvarchar(50) NOT NULL,
  Country nvarchar(50) NOT NULL,
  PostalCode nvarchar(50) NOT NULL,
  CreateDate datetime NOT NULL DEFAULT(GETDATE()),
  ModifiedDate datetime NOT NULL DEFAULT(GETDATE())
)
```

You create a stored procedure that includes the AccountNumber, Country, and StateProvince columns from the dbo.Customers table. The stored procedure accepts a parameter to filter the output on the AccountNumber column. You need to optimize the performance of the stored procedure. You must not change the existing structure of the table. Which Transact-SQL statement should you use?

- A. CREATE STATISTICS ST_Customer_AccountNumber ON dbo.Customer (AccountNumber) WITH
FULLSCAN;

- B. CREATE CLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber);
- C. CREATE NONCLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber) WHERE AccountNumber = ";
- D. CREATE NONCLUSTERED INDEX IX_Customer_AccountNumber ON dbo.Customer (AccountNumber) INCLUDE (Country, StateProvince);

Answer: D

QUESTION NO: 7

You have a table named Customer. You need to ensure that customer data in the table meets the following requirements: credit limit must be zero unless customer identification has been verified. credit limit must be less than 10,000. Which constraint should you use?

- A. CHECK (CreditLimt BETWEEN 1 AND 10000)
- B. CHECK (Verified = 1 AND CreditLimt BETWEEN 1 AND 10000)
- C. CHECK ((CreditLimt = 0 AND Verified = 0) OR (CreditLimt BETWEEN 1 AND 10000 AND Verified = 1))
- D. CHECK ((CreditLimt = 0 AND Verified = 0) AND (CreditLimt BETWEEN 1 AND 10000 AND Verified = 1))

Answer: C

QUESTION NO: 8

You have a table named AccountsReceivable. The table has no indexes. There are 75,000 rows in the table. You have a partition function named FG_AccountData. The AccountsReceivable table is defined in the following Transact-SQL statement:

```
CREATE TABLE AccountsReceivable (  
column_a INT NOT NULL,  
column_b VARCHAR(20) NULL)  
ON [PRIMARY];
```

You need to move the AccountsReceivable table from the PRIMARY file group to FG_AccountData. Which Transact-SQL statement should you use?

- A. CREATE CLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON [FG_AccountData];
- B. CREATE NONCLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON [FG_AccountData];
- C. CREATE CLUSTERED INDEX idx_AccountsReceivable ON AccountsReceivable(column_a) ON FG_AccountData(column_a);