

# *KillTest*

質量更高 服務更好



## 學習資料

<http://www.killtest.net>

一年免費更新服務

**Exam** : **70-442GB2312**

**Title** : PRO:Design & Optimize  
Data Access by Using MS  
SQL Serv 2005

**Version** : Demo

1. 您是公司的数据库开发人员。有一个将订单项保存到 **OrderItems** 表的过程。如果该项不存在，则应执行插入。如果该项存在，则应执行更新。**OrderItems** 表设计为如下所示。

OrderID (PK, FK)

ProductID (PK, FK)

Quantity

int

int

int

您需要开发一个使用最少量资源的例程。

您应该使用哪个例程？

A. BEGIN TRY

```
UPDATE OrderItems SET Quantity = @Quantity
```

```
WHERE OrderID = @OrderID
```

```
AND ProductID = @ProductID;
```

END TRY

BEGIN CATCH

```
INSERT OrderItems (OrderID, ProductID, Quantity)
```

```
VALUES (@OrderID, @ProductID, @Quantity);
```

END CATCH

B. IF EXISTS (SELECT \* FROM OrderItems

```
WHERE OrderID = @OrderID
```

```
AND ProductID = @ProductID
```

)

```
UPDATE OrderItems SET Quantity = @Quantity
```

```
WHERE OrderID = @OrderID
```

```
AND ProductID = @ProductID;
```

ELSE

```
INSERT OrderItems (OrderID, ProductID, Quantity)
```

```
VALUES (@OrderID, @ProductID, @Quantity);
```

C. IF NOT EXISTS (SELECT \* FROM OrderItems

```
WHERE OrderID = @OrderID
AND ProductID = @ProductID
)
INSERT OrderItems (OrderID, ProductID, Quantity)
VALUES (@OrderID, @ProductID, @Quantity);
ELSE
UPDATE OrderItems SET Quantity = @Quantity
WHERE OrderID = @OrderID
AND ProductID = @ProductID;
D. UPDATE OrderItems SET Quantity = @Quantity
WHERE OrderID = @OrderID AND ProductID = @ProductID;
IF(@@ROWCOUNT = 0)
INSERT OrderItems (OrderID, ProductID, Quantity)
VALUES (@OrderID, @ProductID, @Quantity);
```

**Answer: D**

2. 您是公司的数据库开发人员。在您的数据库中，**Employees** 表包含大约 1,000 行，**Orders** 表包含数百万行。

**Employees** 表设计为如下所示。

EmployeeID (PK)

Firstname

Lastname

...

int

nvarchar(30)

nvarchar(30)

...

**Orders** 表设计为如下所示。

OrderID (PK)

SoldByEmployeeID (FK)

OrderDate

...

int

int

datetime

...

您需要编写一个查询，该查询返回每位员工所下的订单数。

您应该使用哪个查询？

A. SELECT e.Firstname, e.Lastname, COUNT(\*)

FROM Employees AS e

LEFT OUTER JOIN Orders AS o

ON o.SoldByEmployeeID = e.EmployeeID

GROUP BY e.Firstname, e.Lastname

B. SELECT e.Firstname, e.Lastname, COUNT(\*)

FROM Employees AS e

LEFT OUTER JOIN Orders AS o

ON o.SoldByEmployeeID = e.EmployeeID

GROUP BY e.EmployeeID, e.Firstname, e.Lastname

C. SELECT e.Firstname, e.Lastname, COUNT(o.OrderID)

FROM Employees AS e

LEFT OUTER JOIN Orders AS o

ON o.SoldByEmployeeID = e.EmployeeID

GROUP BY e.EmployeeID, e.Firstname, e.Lastname

D. SELECT e.Firstname, e.Lastname, COUNT(o.OrderID)

FROM Employees AS e

LEFT OUTER JOIN Orders AS o

ON o.SoldByEmployeeID = e.EmployeeID

GROUP BY e.Firstname, e.Lastname

**Answer: C**

3. 您是公司的数据库开发人员。现必须优化 **Stockholm** 市订单的处理例程，以获得更好的性能。  
当前例程如以下代码段所示。

```
...  
DECLARE OrderCursor CURSOR FOR  
    SELECT OrderID, City, CustomerID FROM Orders;  
OPEN OrderCursor;  
FETCH NEXT FROM OrderCursor INTO @OrderID, @CustomerID, @City;  
WHILE(@@FETCH_STATUS = 0)  
    BEGIN  
        IF(@City <> 'Stockholm')  
            GOTO Next;  
        <Code for processing order>  
    Next:  
        FETCH NEXT FROM OrderCursor INTO  
            @OrderID, @CustomerID, @City;  
    END
```

...  
您需要优化该例程以获得最佳性能。

您应该怎么办？

- A. 将游标更改为以 **DYNAMIC** 形式声明。
- B. 将游标更改为以 **KEYSET** 形式声明。
- C. 将游标更改为以 **STATIC** 形式声明。
- D. 更改游标中的 **SELECT** 语句以包括 **WHERE City = 'Stockholm'**。
- E. 将 **IF** 语句更改为检查 **@City = 'Stockholm'**，从而避免使用 **GOTO** 语句。

**Answer: D**

4. 您是公司的数据库开发人员。一个名为 **Articles** 的表包含报纸文章，其设计如下所示。

ArticleID (PK)

Title

ArticleXml

CategoryID (FK)

int

nvarchar(50)

xml

int

ArticleXml 列中的 XML 文档如下所示。

```
<article>
  <paragraph>...</paragraph>
  <image>...</image>
  ...
  <image>...</image>
  ...
```

</article>您需要设计一个查询，该查询将按照 CategoryID 值显示图像的总数。

您应该使用哪个查询？

A. SELECT

```
CategoryID
, COUNT(ArticleXml.value('count(/article/image)',
  'INT')) AS NumberOfImages
```

FROM Articles

GROUP BY CategoryID;

B. SELECT

```
CategoryID
, COUNT(ArticleXml.exist('/article/image'))
  AS NumberOfImages
```

FROM Articles

GROUP BY CategoryID;

C. SELECT

```
CategoryID
, SUM(ArticleXml.value('count(/article/image)', 'INT'))
  AS NumberOfImages
```

FROM Articles

GROUP BY CategoryID;

D. SELECT

CategoryID

,SUM(ArticleXml.exist('/article/image')

AS NumberOfImages

FROM Articles

GROUP BY CategoryID;

**Answer: C**

5. 您是公司的数据库开发人员。您负责在公司新的 SQL Server 2005 计算机上将名为 DB1 和 DB2 的两个现有 SQL Server 2000 数据库合并为一个名为 NewDB 的数据库。

以下查询在用于访问 DB1 数据库的旧应用程序中可以正常工作。

```
SELECT * FROM Customer WHERE FaxNumber = NULL
```

但是，该查询在新的 SQL Server 2005 计算机上运行时将不会返回任何行，即使 FaxNumber 列中存在 NULL 值。

您需要解决该问题，以使该查询在 SQL Server 2005 计算机上能正常工作。您需要确保您的解决方案不会影响 DB2 中表和查询的行为。

您应该怎么办？

- A. 使用 ISNULL 函数，而不是 = NULL。
- B. 将 = NULL 更改为 IS NULL。
- C. 将 = NULL 更改为 = 'NULL'。
- D. 在查询前面添加以下命令：SET ANSI\_NULLS ON
- E. 使用 ALTER DATABASE 命令将 ANSI\_NULLS 数据库选项更改为 ON。

**Answer: B**

6. 您是公司的数据库开发人员。用户需要有关在 TotalDue 列中具有最高值的订单的信息。用户希望能够在每次请求信息时返回不同的记录百分比。

您需要在 SQL Server 2005 上生成代码，该代码将允许前端应用程序传递最终用户将为要返回的记录百分比提供的变量。



您应该如何编写代码?

A. CREATE PROCEDURE TopPercentTotalDue

@P int

AS

SELECT TOP(@P) PERCENT

TotalDue, OrderDate

FROM Purchasing.PurchaseOrderHeader

ORDER BY TotalDue DESC

B. CREATE PROCEDURE TopPercentTotalDue

@P int

AS

SELECT TOP(@P) PERCENT

TotalDue, OrderDate

FROM Purchasing.PurchaseOrderHeader

ORDER BY TotalDue ASC

C. CREATE PROCEDURE TopPercentTotalDue

@R int

AS

SET rowcount @R

SELECT TotalDue, OrderDate

FROM Purchasing.PurchaseOrderHeader

ORDER BY TotalDue DESC

D. CREATE PROCEDURE TopPercentTotalDue

@R int

AS

SET rowcount @R

SELECT TotalDue, OrderDate

FROM Purchasing.PurchaseOrderHeader

ORDER BY TotalDue ASC

**Answer: A**

7. 您是公司的数据库开发人员。每天将以 XML 文档的形式发送 **Parts** 数据库的信息，并将该信息放在 **ProductXML** 表的 **ProductInfo** 列中的单一行中。该 XML 文档中的信息的示例如下所示。

```
<Products>
    <Product ID="..." Name="..." Number...">
    <Product ID="..." Name="..." Number...">
    ...
</Products>
```

每天结束时将运行一个批处理作业。这个批处理作业将 XML 文档从 **ProductXML** 表导入 **Product** 表。将为每个 **Product** 元素从 XML 文档中提取产品的 ID、名称和编号，并将这些信息插入 **Product** 表的 **ProductID**、**Name**、和 **Number** 列中。在批处理结束时截断 **ProductXML** 表。

您需要设计从 **ProductXML** 表中检索行以插入 **Product** 表的查询。

您应该使用哪个 Transact-SQL 查询从 **ProductXML** 表中查询数据？

#### A. SELECT

```
ProductInfo.value('/Products/Product)[1]/@ID', 'integer')
    AS ProductID,
ProductInfo.value('/Products/Product)[1]/@Name', 'varchar(50)')
    AS Name,
ProductInfo.value('/Products/Product)[1]/@Number', 'varchar(25)')
    AS ProductNumber
```

FROM ProductXML

#### B. SELECT

```
col.value('@ID', 'integer') AS ProductID,
col.value('@Name', 'varchar(50)') AS Name,
col.value('@Number', 'varchar(25)') AS ProductNumber
```

FROM ProductXML

CROSS APPLY ProductInfo.nodes('/Products/Product') AS x (col)

#### C. SELECT

```
col.query('data(@ID), data(@Name), data(@Number)')
```

FROM ProductXML

CROSS APPLY ProductInfo.nodes('/Products/Product') AS x (col)

D. SELECT

```
ProductInfo.query('
    for $p in /Products/Product
    return fn:concat($p/@ID, $p/@Name, $p/@Number)
') AS Result
```

FROM ProductXML

**Answer: B**

8.您是公司的数据库开发人员。公司正在从 SQL Server 2000 升级到 SQL Server 2005。公司新的编码标准要求符合 ANSI 标准。

您的经理要求您将以下查询转换为使用 ANSI 标准措词。

```
SELECT E.EmployeeID, E.Hiredate, J.Resume
FROM HumanResources.Employee AS E,
HumanResources.JobCandidate AS J
WHERE E.EmployeeID *= J.EmployeeID
```

新查询必须始终生成与现有查询相同的结果集。

您需要编写查询以满足这些要求。

您应该使用哪个 Transact-SQL 查询？

- A. SELECT E.EmployeeID, E.Hiredate, J.Resume  
FROM HumanResources.Employee AS E RIGHT OUTER JOIN  
HumanResources.JobCandidate AS J  
ON E.EmployeeID = J.EmployeeID
- B. SELECT E.EmployeeID, E.Hiredate, J.Resume  
FROM HumanResources.Employee AS E LEFT OUTER JOIN  
HumanResources.JobCandidate AS J  
ON E.EmployeeID = J.EmployeeID
- C. SELECT E.EmployeeID, E.Hiredate, J.Resume  
FROM HumanResources.Employee AS E INNER JOIN  
HumanResources.JobCandidate AS J

ON E.EmployeeID = J.EmployeeID

D. SELECT E.EmployeeID, E.Hiredate, J.Resume

FROM HumanResources.Employee AS E FULL OUTER JOIN

HumanResources.JobCandidate AS J

ON E.EmployeeID = J.EmployeeID

**Answer: B**

9. 您是公司的数据库开发人员。公司最近升级到了 SQL Server 2005。

您正在创建一个查询，以便返回显示指定员工组所下达采购订单总数的信息，按请求订单的供应商分组。

结果集应在其单独一列中显示每位员工的采购订单总数。每个供应商在结果集中都应有自己的行。

您需要创建查询来满足这些要求。

您应该使用哪个 Transact-SQL 命令？

A. CROSS APPLY

B. UNION

C. PIVOT

D. INTERSECT

**Answer: C**

10. 您是公司的数据库开发人员。您正在创建一个参数化存储过程，该存储过程将查询数据并返回一些行，其中包含作为该存储过程输入提供的动词的任何动词时态。查询还应返回同时包含名词的单数和复数形式的行。数据包括在若干 varchar(max) 和 varbinary(max) 列中。

您需要创建存储过程来满足这些要求。

您应该执行哪两项操作？（每个正确答案都仅给出了部分解决方案。请选择两个答案。）

A. 使用 LIKE 运算符。

B. 使用 CONTAINS 谓词。

C. 针对适当的一个或多个列创建全文索引。

D. 针对适当的一个或多个列创建非聚集索引。

**Answer: BC**

11. 您是公司的数据库开发人员。您正在创建一个存储过程，该存储过程将使用 Transact-SQL 游标与存

储过程中的其他语句共享结果集。此游标必须有行顺序不更改的固定成员身份。

通过游标检索的行中的数据应可更新。这些更新在完成后应可供查看。在实现此游标时，不管会对游标的速度产生什么影响，您都需要将内存使用率降到最低。

您需要创建存储过程来满足这些要求。

您应该在 `DECLARE CURSOR` 语句中使用哪个选项？

- A. DYNAMIC
- B. FAST\_FORWARD
- C. KEYSSET
- D. STATIC

**Answer: C**

12. 您是公司的数据库开发人员。您正在更新一个声明和使用动态游标的存储过程。

该存储过程最初需要返回结果集并允许更新结果集。要求已经发生变化，现在数据将为只读。因此，应不允许通过游标进行数据更改。该存储过程中使用的唯一提取命令为 `FETCH NEXT`。

您需要重新编写游标定义以尽可能有效地使用资源并提供最佳性能，同时可满足新的要求。

您应该怎么办？

- A. 更改游标 `DECLARE` 语句以便只使用 `LOCAL` 和 `FORWARD_ONLY` 选项。
- B. 更改游标 `DECLARE` 语句以便只使用 `LOCAL`、`FORWARD_ONLY` 和 `OPTIMISTIC` 选项。
- C. 更改游标 `DECLARE` 语句以便只使用 `LOCAL` 和 `FAST_FORWARD` 选项。
- D. 将 `SCROLL_LOCKS` 选项添加到包含 `LOCAL` 和 `DYNAMIC` 选项的当前 `DECLARE` 语句。

**Answer: C**

13. 您是公司的数据库开发人员。在您的数据库中，`Categories` 表的设计如下所示。

CategoryID (PK)

Name (UQ)

ParentCategoryID (FK)

int

nvarchar(50)

int

您需要编写一个查询，检索直接或间接存在于类别 7 下的所有产品类别。

您应该使用以下哪个查询？

A. WITH c AS (

```
SELECT CategoryID
FROM Categories
WHERE CategoryID = 7
UNION ALL
SELECT cs.CategoryID
FROM Categories AS cs
INNER JOIN c ON c.ParentCategoryID = cs.CategoryID
```

)

SELECT \* FROM Categories AS c2

```
WHERE c2.CategoryID IN (SELECT c.CategoryID FROM c)
```

B. WITH c AS (

```
SELECT CategoryID
FROM Categories
WHERE ParentCategoryID = 7
UNION ALL
SELECT cs.CategoryID
FROM Categories AS cs
INNER JOIN c ON c.ParentCategoryID = cs.CategoryID
```

)

SELECT \* FROM Categories AS c2

```
WHERE c2.CategoryID IN (SELECT c.CategoryID FROM c)
```

C. SELECT \* FROM Categories AS c2

```
WHERE c2.CategoryID IN
```

(

```
SELECT CategoryID
FROM Categories
WHERE ParentCategoryID = 7
UNION ALL
```

```
SELECT cs.CategoryID
FROM Categories AS cs
INNER JOIN Categories AS c
    ON c.ParentCategoryID = cs.CategoryID
)
```

D. SELECT \* FROM Categories AS c2

```
WHERE c2.CategoryID IN
```

```
(
SELECT CategoryID
FROM Categories
WHERE CategoryID = 7
EXCEPT
SELECT cs.CategoryID
FROM Categories AS cs
WHERE ParentCategoryID = 7
)
```

**Answer: B**

14. 您是公司的数据库开发人员。一个 XML 文档存储在 XML 变量 @Article 中。该变量中的 XML 文档可以包含数百个文章元素。XML 文档显示在以下片段中。

```
<articles>
  <article>
    <title>...</title>
    <paragraph>...</paragraph>
    <image filename="...">
      <description>...</description>
    </image>
    <paragraph>...</paragraph>
    <image filename="...">
      <description>...</description>
```

```
</image>
```

```
<paragraph>...</paragraph>
```

```
<paragraph>...</paragraph>
```

```
</article>
```

...

```
</articles>
```

您需要设计一个查询，该查询将采用以下表格格式从 XML 文档中返回所有图像的文件名和说明。

您应该使用哪个查询？

A. SELECT

```
@Article.value('/articles/article/image')[1]/@filename',  
    'NVARCHAR(50)' AS Filename  
,@Article.value('/articles/article/image/description')[1],  
    'NVARCHAR(max)' AS Description
```

B. SELECT

```
col.value('(articles/article/image)[1]/@filename',  
    'NVARCHAR(50)')  
    AS Filename  
,col.value('articles/article/image/description[1]',  
    'NVARCHAR(max)')  
    AS Description
```

```
FROM @Article.nodes('/') AS x (col);
```

C. SELECT

```
col.value('@filename', 'NVARCHAR(50)') AS Filename  
,col.value('description[1]', 'NVARCHAR(max)')  
    AS Description
```

```
FROM @Article.nodes('/articles/article/image') AS x (col) ;
```

D. SELECT

```
col.query('<filename>{@filename}</filename>')  
    AS Filename  
,col.query('description[1]') AS Description
```



FROM @Article.nodes('/articles/article/image') AS x (col) ;

**Answer: C**

15. 您是公司的数据库开发人员。您正在开发一个报表，该报表将返回客户名称和电话号码。客户可能有办公电话号码、移动电话号码和传真号码。报表必须返回每位客户的第一个非空电话号码。

您需要编写查询，该查询为报表返回必需的客户信息。

您应该使用哪一个？

A. NULLIF

B. IF

C. REPLACE

D. COALESCE

**Answer: D**

16. 您是公司的数据库开发人员。您使用以下 Transact-SQL 代码。

```
SELECT e.Resume,j.EmployeeID FROM  
    JobCandidate j, Employee e
```

```
WHERE j.EmployeeID =* e.EmployeeID
```

您需要重新编写此代码以符合 ANSI。

您应该如何重新编写代码？

A. SELECT e.Resume, j.EmployeeID FROM

```
    JobCandidate AS j LEFT OUTER JOIN Employee AS e
```

```
ON j.EmployeeID = e.EmployeeID
```

B. SELECT e.Resume, j.EmployeeID FROM

```
    JobCandidate AS j FULL OUTER JOIN Employee AS e
```

```
ON j.EmployeeID = e.EmployeeID
```

C. SELECT e.Resume, j.EmployeeID FROM

```
    JobCandidate AS j RIGHT OUTER JOIN Employee AS e
```

```
ON j.EmployeeID = e.EmployeeID
```

D. SELECT e.Resume, j.EmployeeID FROM

```
    JobCandidate AS j CROSS JOIN Employee AS e
```

**Answer: C**

17. 您是本地学区的数据库开发人员。您在 `xml` 数据类型的列中维护有关教师和职员的信息。您需要生成一个表格格式报表，其中包含每位员工的 `EmployeeID`、`Title` 和 `YearsOfEmployment` 列的值。此报表应依据 `Title` 按升序排序，并依据 `YearsOfEmployment` 按降序排序。您应该使用什么来检索此数据？

- A. FOR XML AUTO
- B. FOR XML PATH
- C. `xml` 数据列的查询方法
- D. `xml` 数据列的值和节点方法

**Answer: D**

18. 您是公司的数据库开发人员。您正在开发一个存储过程，该存储过程必须接受 `xml` 数据类型的参数，并将数据存储存储在关系表结构中。您需要编写存储过程来满足这些要求。您应该使用哪个功能？

- A. FOR XML AUTO
- B. `xml` 数据类型的节点方法
- C. `xml` 数据类型的查询方法
- D. `xml` 数据类型的修改方法

**Answer: B**

19. 您是公司的数据库开发人员。您的数据库包含一个名为 `Inventory` 的表，该表如下所示。

`ItemID (PK)`

`Item`

`Color`

`Quantity`

`int`

`Varchar(255) NOT NULL`

`varchar(50) NOT NULL`

int

您正在编写一个报表，该报表必须呈现以下输出。

Item	Color	QtySum
-----		
Chair	Blue	101.00
Chair	Red	210.00
Chair	(null)	311.00
Table	Blue	124.00
Table	Red	223.00
Table	(null)	347.00
(null)	(null)	658.00
(null)	Blue	225.00
(null)	Red	433.00

您需要确保报表满足此要求。

您应该使用哪两个关键字？（每个正确答案都仅给出了部分解决方案。请选择两个答案。）

- A. COMPUTE BY
- B. CUBE
- C. ROLLUP
- D. GROUP BY

**Answer: BD**

20. 您是公司的数据库开发人员。您正在开发给定时间段的销售报表。该报表必须基于该时间段销售代表的总销售额列出所有销售代表及他们的位置。如果两个销售代表的总销售额相同，应为他们分配相同的位置。示例输出如下所示。

SalesRepID

Sales

Position

456

123,456

1

765

123,456

1

983

103,762

3

106

101,101

4

您需要创建报表来满足这些要求。

您应该使用哪个运算符？

A. RANK

B. DENSE\_RANK

C. NTILE

D. ROW\_NUMBER

**Answer: A**