0h 57m

Submit Task

1

2

3

Task 3

1. You have written a stored procedure, *insertOrderAndItems*, that inserts data into *dbo.Order*. This table has an auto-incremented/identity column called *Id*. After inserting data into *dbo.Order* you use *@@IDENTITY* to read the value generated for the *Id* column. Then you use the retrieved *Id* to insert related data into the *dbo.OrderItem* table.

   When you call your stored procedure in the testing environment, for example from SQL Server Management Studio, it seems to be working. However, there are problems in the production environment, i.e. sometimes *insertOrderAndItems* uses the wrong *Id* when inserting data into *dbo.OrderItem* table. How can you fix the problem?

   - [ ] A: Replace *@@IDENTITY* with *SCOPE_IDENTITY*.
   - [ ] B: Replace *@@IDENTITY* with *IDENT_CURRENT*.
   - [ ] C: Remove *@@IDENTITY* and use an *OUTPUT* clause with an *INSERT* statement to read the value generated for the *Id column*.
   - [ ] D: Read the value generated for the *Id* column with the following query:

     SELECT TOP(1) Id FROM dbo.Order ORDER BY Id ASC
   - [ ] E: Read the value generated for the *Id* column with the following query:

     SELECT TOP(1) Id FROM dbo.Order ORDER BY Id DESC

---

2. You have a large table with millions of rows. You want to delete all the rows in this table as quickly as possible. How would you do that? Pick the best option.

   - [ ] A: Use TRUNCATE TABLE.
   - [ ] B: Use DELETE.
   - [ ] C: Use DROP TABLE and then recreate it with CREATE TABLE.
   - [ ] D: Write a stored procedure that uses a cursor to iterate through all rows in a table and delete them one by one.