70-451

Microsoft
PRO: Designing Database Solutions and Data Access Using Microsoft SQL Server 2008

Pass4sureofficial.com is a reputable IT certification examination guide, study guides and audio exam provider, we not only ensure that you pass your 70-451 exam in first attempt, but also you can get a high score to acquire Microsoft certification.

If you use pass4sureofficial 70-451 Certification questions and answers, you will experience actual 70-451 exam questions/answers. We know exactly what is needed and have all the exam preparation material required to pass the exam. Our Microsoft exam prep covers over 95% of the questions and answers that may be appeared in your 70-451 exam. Every point from pass4sure 70-451 PDF, 70-451 review will help you take Microsoft 70-451 exam much easier and become Microsoft certified. All the Questions/Answers are taken from real exams.

Here’s what you can expect from the Pass4sureOfficial Microsoft 70-451 course:

* Up-to-Date Microsoft 70-451 questions taken from the real exam.
* 100% correct Microsoft 70-451 answers you simply can't find in other 70-451 courses.
* All of our tests are easy to download. Your file will be saved as a 70-451 PDF.
* Microsoft 70-451 brain dump free content featuring the real 70-451 test questions.

Microsoft 70-451 certification exam is of core importance both in your Professional life and Microsoft certification path. With Microsoft certification you can get a good job easily in the market and get on your path for success. Professionals who passed Microsoft 70-451 exam training are an absolute favorite in the industry. You will pass Microsoft 70-451 certification test and career opportunities will be open for you.
Question: 1
You are the designer of a SQL database on an instance of SQL Server 2008. Your database contains a course table and an instructor table defined as follows:

You want to create a foreign key constraint to associate the two tables and prevent a user from deleting an instructor who is currently assigned to teach a course. Which clause should you include in your foreign key constraint?

A. On delete no action
B. On delete cascade
C. On delete set null
D. On delete set default

Answer: A

For example, in this scenario, you might use the following statement to create the necessary foreign key constraint:

```
alter table course add constraint fk_instructor foreign key (instructorid) references instructor (instructorid) on delete no action;
```

This statement includes the add constraint clause of the alter table statement to add a foreign key constraint to the course table. The references clause specifies the referenced column in the parent table. When you include the on delete no action clause, it ensures that referential integrity is maintained. If a referenced row in the parent table is deleted, then the delete fails if child rows exist. With the given statement, a user could not delete an instructor who was assigned to courses in the course table. You should note that no action is the default for on delete and on update. You should not include the on delete cascade clause in your foreign key constraint. If you did so and a user deleted an instructor that was assigned to teach a course, the delete operation would be performed, and all courses for the instructor would be deleted as well. You should not include the on delete set null clause in your foreign key constraint. If you did so and a user deleted an instructor who was assigned to teach a course, the instructor column would be set to null for all of the instructor's courses in the course table. You should note that if the foreign key consists of multiple columns, all of the columns in the foreign key would be set to null. You should not include the on delete set default clause in your foreign key constraint. If you did so and a user deleted an instructor who was assigned to teach a course, the instructor column would be set to the default value for all of that instructor's courses in the course table. You should note that if the foreign key consists of multiple columns, all of the columns in the foreign key are set to the default value. In addition, you should ensure that each foreign key column has a default definition; otherwise, the column will be assigned a null value.

Question: 2
You manage a database on an instance of SQL Server 2008 at a large educational institution. You have classmate and class detail tables as shown: (click exhibit to view table design)
the user1 user starts a session and executes the following transact-sql: begin transaction update class detail with(tab lock) set classdetail.details = classdetail.details + ' this is a freshman-level class.' From class master inner join class detail on classmaster.classid = classdetail.classid where classmaster.level = 'freshman'; then, user2 starts a session and executes the following transact-sql: begin transaction insert into class master(description, seats, level) values ('calculus i', 25, 'sophomore'), ('accounting iii', 80, 'senior'), ('world history', 30, 'freshman'); delete from dbo.classdetail where cdid = 2; commit transaction what is the result?

A. User1's updates are applied first, and then user2's dml operations are performed.
B. User2's session hangs waiting for user1's update to complete.
C. User1's updates are not performed, but user2's dml operations complete successfully.
D. User2's session immediately returns an error.

Answer: B

In this scenario, user1 begins a transaction and executes an update statement, but does not commit the changes. then, user2 begins a transaction, but user1 currently has locks on the table that prevents user1's actions from being performed. user2's session hangs waiting for user1's session to release the locks. Sql server uses locking to control which transactions have access to which resources at any given time. This prevents one transaction from overwriting another transaction's changes. Sql server determines the locks that will be acquired based on the type of action being performed. however, you can override default locking behavior if necessary using table hints. the option that states user1's updates are applied first and then user2's dml operations are performed is incorrect. user1 does not commit the changes. therefore, the changes are pending and user2's operations are not performed. the option that states user1's updates are not performed but user2's dml operations are performed is incorrect. in this scenario, user1's updates have not been committed and will remain pending, blocking user2's operations from executing. the option that states user2's session immediately returns an error is incorrect. user2's session will wait to obtain the necessary locks to perform the required operations.

Question: 3
You are a database developer on an instance of sql server 2008. you are creating a tracking application. You want to create a column in one of your tables that stores gps latitude and longitude coordinates. which data type should you use for the column?

A. Xml
B. Table
C. Geometry
D. Geography
The geography data type represents data in a round-earth coordinate system. The geography data type is used to represent data as GPS latitude and longitude coordinates. You should not use the xml data type because this data type is used to store XML data, such as XML documents or XML document fragments. You should not use the table data type because this data type is used to store rows of data from a result set. The table data type can be processed later as needed, much like a temporary table. You can also pass table-valued parameters into stored procedures and functions for processing. You should not use the geometry data type. The geometry data type represents data in a Euclidean, or flat, coordinate system.

Question: 4
You are a database developer on an instance of SQL Server 2008. You have a WorkOrder table and a WorkOrderDetail table. The WorkOrder table contains one row for each work order, and the WorkOrderDetail table contains line items of work performed for each work order. You want to create a foreign key constraint to relate the two tables. You want to ensure that if a work order identifier in the WorkOrder table is updated, that the corresponding rows in the WorkOrderDetail table are also updated to maintain referential integrity. What should you do?

A. Use a trigger to enforce referential integrity.
B. Create a check constraint on the WorkOrder table.
C. Include the ON UPDATE CASCADE clause in your foreign key constraint.
D. Include the WITH CHECK clause in your foreign key constraint.

Answer: C

When you include the ON UPDATE CASCADE clause, it ensures that if a referenced row in the parent table is updated, then the corresponding child rows are updated and referential integrity is maintained. This would ensure that if a user updated the identifier of a work order, the corresponding rows in the WorkOrderDetail table would also be updated. You should not use a trigger to enforce referential integrity. Triggers should not be used when constraints can accomplish the same task, foreign key constraints are used to enforce referential integrity. You should not create a check constraint on the WorkOrder table. A check constraint is used to restrict the data allowed for a column to specific values. A check constraint consists of a Boolean expression that evaluates to either true or false. If the expression evaluates to true, the value is allowed for the column, and if the expression evaluates to false, the value is not allowed for the column. Check constraints can be defined at the table level or column level, but only check constraints defined at the table level can use columns other than the constrained column in the constraint expression. You should not include the WITH CHECK clause in your foreign key constraint because this only enables the constraint. By default, foreign key and check constraints are enabled when they are created. You can use the NO CHECK constraint clause of the ALTER TABLE statement to temporarily disable a foreign key or check constraint if you need to insert rows that violate the constraint.

Question: 5
You are a database developer on an instance of SQL Server. You have defined an inline table valued function using the following statement: create function udf_getorderdetails (@or did int)returns table as return (select sh.orderdate, sd.* from salesorderheader sh inner join salesorderdetail sd on sh.salesorderid = sd.salesorderid where sd.salesorderid=@or did); You have a table named SalesOrderHistory that contains a SalesOrderID column. You want to query the SalesOrderHistory table and use the udf_getorderdetails function to return order details. All sales orders in the SalesOrderHistory table and the associated order details should be displayed, even if no details exist for a given order. Which query should you use?
A. Select * from salesorderhistory h cross apply udf_getorderdetails(h.salesorderld) as d
B. Select h.*, d.* from salesorderhistory h outer apply udf_getorderdetails(h.salesorderld) as d 
    order by h.salesorderld; select * from udf_getorderdetails(t.salesorderid)) as t;
C. Select h.*, d.* inner join udf_getorderdetails(h.salesorderld) as d on h.salesorderld = 
    d.salesorderld order by h.salesorderld;

Answer: B

You should use the following query:
select h.*, d.* from salesorderhistory hotter apply udf_getorderdetails (h.salesorderld) as d order 
by h.salesorderld; in this scenario, you wanted to join the results of a table-valued function to 
at least another table. you can accomplish this using the apply operator in the from clause of your query. 
the syntax is as follows: select column list from outer table (cross | outer) apply inner_table 
the inner_table may specify another table or view, but may also specify a function that returns a 
table type. in this scenario, you specified the udf_getorderdetails function as the inner table. for 
each row in the query, the udf_getorderdetails function is called using the salesorderld from the 
outer table. if the function returns rows, these rows are joined with the outer table using union all 
and are available to the query. you should use outer apply to ensure that all rows in the outer 
table, salesorderhistory, are returned even if no order details exist. for rows in the sales order 
history table that are not returned by the function, all order detail columns will be assigned a null 
value. when using the apply operator, the outer table may also use a table-valued function, but it 
cannot pass the function columns to the inner_table as arguments. you should not use the query 
that includes an inner join because this syntax is invalid. this statement will return the following 
error: msg 4104, level 16, state 1, line 12the multi-part identifier "h.salesorderld" could not be 
bound. you should not use the query that uses a sub query because it attempts to reference 
t.salesorderld in the inner query. this statement will generate the following error: msg 4104, level 
16, state 1, line 25the multi-part identifier "t.salesorderld" could not be bound. you should not use 
the query that uses cross apply. a cross apply operator only returns rows that had rows returned 
by the udf_getorderdetails function. in this scenario, you wanted to return all rows from the sales 
order history table, even if they contained no order details. a cross apply would not return rows in 
the salesorderhistory table that did not contain order details.

Question: 6
You manage a database on an instance of sql server 2008 for a large sales organization. You 
have a contact table defined with the following statement: create table contact(contacted int 
primary key,contacttype nvarchar(10),first name nvarchar(30) not null,lastname nvarchar(50) not 
nul, territory nvarchar (20),region nvarchar(10),repid int,initcontact date time default get date()); 
you have a stored procedure that accepts a territory and region as input parameters and queries 
the contact table to return contact information as a table data type. users in the marketing 
department frequently use the stored procedure to obtain information about contacts for a 
specified territory and region. each region contains multiple territories. there are generally a small 
number of contacts within each territory, but each region contains many contacts. the stored 
procedure is performing poorly, and you want to optimize its performance. which action should 
you take?

A. Create a view on the contact table and modify the stored procedure to use the view. 
B. Create a no clustered composite index on the territory and region columns. 
C. Create two no clustered indexes, one on the territory column and one on the region column. 
D. Re-create the stored procedure and include the with schema binding clause.

Answer: B

To optimize the query in the stored procedure, you should create a composite index on the two 
columns. In this scenario, each region contains many contacts, and each territory contains only a
few. Because there are fewer region values than territory values, you should specify the territory column first when creating the index. With the composite index, SQL Server will find the rows that match the territory and region criteria in the WHERE clause with a single index search. The search will be performed only by reading the index pages, and the data pages will be accessed only once to retrieve the result set. You might create the index using the following statement:

```sql
create index ix_contact_territoryregion on contact (territory, region);
```

Using this index would generally offer better performance than if you had created the index with the region column first, because the territory column contains more unique values. You should not create a view on the contact table and modify the stored procedure to use the view. Using a view would not improve performance. The stored procedure would still have to query the view, and the view would access the underlying table. You should not create two no clustered indexes, one on the territory column and one on the region column. In this scenario, a single composite no clustered index would provide better performance. Although SQL Server can use more than one index on a table, it is unlikely that the index on the region column would be used in this scenario because of the low uniqueness of the column's values. You should not re-create the stored procedure including the WITH SCHEMA_BINDING clause because the WITH SCHEMA_BINDING clause is not allowed for a stored procedure. In addition, it has no affect on performance. Attempting to create a stored procedure including with schema binding will generate an error similar to the following: (Message 487, Level 16, State 1, Procedure p, Line 3) an invalid option was specified for the statement "create/alter procedure". Schema binding is used when creating a view or function. You can include the WITH SCHEMA_BINDING clause in a create view statement to ensure that no base tables on which the view is based are dropped or modified in a way that might affect the view's definition. To drop base tables or make such modifications, a user would first need to drop the view, alter the view omitting schema binding, or alter the view to remove any unwanted dependencies. You can also include the WITH SCHEMA_BINDING clause when creating a function to ensure base tables are not changed in such a way that would render the function unusable.

Question: 7
You are a database developer on an instance of SQL Server 2008. You need to create a table to store information for construction projects undertaken by your company. You want to create a column to store a text narrative for each project. You want to minimize storage space, but allow the narrative column to store up to 10000 bytes of data. Which column definition should you use for the narrative column?

A. Narrative text  
B. Narrative varbinary(max)  
C. Narrative varchar(max)  
D. Narrative varchar(10000)

Answer: C

This will create the narrative column as a large-value type that can store up to 2 gb of data. You should not define the narrative column as a text data type because using the text data type should be avoided. A text data type can support 10000 bytes of data, but cannot be indexed. You should use the large-value type VARCHAR(MAX) instead. You should not define the narrative column as a varbinary(max) data type. A varbinary data type is used to store binary data, not text data. You should not define the narrative column using VARCHAR(10000) because this would generate an error. Even though using large-value types allows you to store more than 8060 bytes within a single row, each individual column can be no larger than 8000 bytes.

Question: 8
You are a developer on an instance of SQL Server 2008. You are designing an event table. The event date column must contain a value. In addition, for each event, the event date must be a
Pass4SureOfficial.com Lifetime Membership Features;

- Pass4SureOfficial Lifetime Membership Package includes over 2500 Exams.
- All exams Questions and Answers are included in package.
- All Audio Guides are included free in package.
- All Study Guides are included free in package.
- Lifetime login access.
- Unlimited download, no account expiry, no hidden charges, just one time $99 payment.
- Free updates for Lifetime.
- Free Download Access to All new exams added in future.
- Accurate answers with explanations (if applicable).
- Verified answers researched by industry experts.
- Study Material updated on regular basis.
- Questions, Answers and Study Guides are downloadable in PDF format.
- Audio Exams are downloadable in MP3 format.
- No authorization code required to open exam.
- Portable anywhere.
- 100% success Guarantee.
- Fast, helpful support 24x7.

View list of All exams (Q&A) downloads
http://www.pass4sureofficial.com/allexams.asp

View list of All Study Guides (SG) downloads
http://www.pass4sureofficial.com/study-guides.asp

View list of All Audio Exams (AE) downloads
http://www.pass4sureofficial.com/audio-exams.asp

Download All Exams Samples
http://www.pass4sureofficial.com/samples.asp

To purchase $99 Lifetime Full Access Membership click here
http://www.pass4sureofficial.com/purchase.asp