



70-762^{Q&As}

Developing SQL Databases

Pass Microsoft 70-762 Exam with 100% Guarantee

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

<https://www.passapply.com/70-762.html>

100% Passing Guarantee
100% Money Back Assurance

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

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers





QUESTION 1

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution. Determine whether the solution meets the stated goals.

You have a database that contains a table named Employees. The table stores information about the employees of your company.

You need to implement and enforce the following business rules:

Limit the values that are accepted by the Salary column.

Prevent salaries less than \$15,000 and greater than \$300,000 from being entered.

Determine valid values by using logical expressions.

Do not validate data integrity when running DELETE statements.

Solution: You implement cascading referential integrity constraints on the table.

Does the solution meet the goal?

A. Yes

B. No

Correct Answer: A

References: [https://technet.microsoft.com/en-us/library/ms186973\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms186973(v=sql.105).aspx)

QUESTION 2

Database users report that SELECT statements take a long time to return results. You run the following Transact-SQL statement:

```
SELECT OBJECT_NAME([object_id]) AS [object_name],  
d.equality_columns, d.inequality_columns, d.included_columns  
FROM sys.dm_db_missing_index_details;
```

Object_name	Equality_columns	Inequality_columns	Included_columns
[Users]	[CountryCode]	[UserStatus]	[UserName]

You need to create one nonclustered covering index that contains all of the columns in the above table. You must minimize index key size. Which Transact-SQL statement should you run?

A. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserName);

B. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserStatus) INCLUDE (UserName);



C. CREATE NONCLUSTERED INDEX IX_User ON Users (CountryCode, UserStatus, UserName);

D. CREATE NONCLUSTERED INDEX IX_User ON Users (UserStatus, CountryCode) INCLUDE (UserName);

Correct Answer: D

Use the UserStatus as the first column in the index, as it is an in_equality column. Incorrect Answers:

A: UserStatus is not included.

References: <https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-indexes-with-included-columns>

QUESTION 3

Background

You have a database named HR1 that includes a table named Employee.

You have several read-only, historical reports that contain regularly changing totals. The reports use multiple queries to estimate payroll expenses. The queries run concurrently. Users report that the payroll estimate reports do not always run. You must monitor the database to identify issues that prevent the reports from running.

You plan to deploy the application to a database server that supports other applications. You must minimize the amount of storage that the database requires.

Employee Table

You use the following Transact-SQL statements to create, configure, and populate the Employee table:

```
CREATE TABLE dbo.Employee
  (EmployeeId INT PRIMARY KEY,
  LastName varchar(50),
  FirstName varchar(50),
  DepartmentId int,
  HireDate datetime,
  TerminationDate datetime,
  SupervisorId int,
  CostCenterNumber int,
  EmployeeStatus int,
  EmployeePayRate int)
) GO
CREATE INDEX IX_1 on dbo.Employee (LastName, FirstName, DepartmentId) INCLUDE (HireDate)
CREATE INDEX IX_2 on dbo.Employee (LastName) INCLUDE (EmployeeId, FirstName, DepartmentId)
CREATE INDEX IX_3 on dbo.Employee (LastName, FirstName) INCLUDE (DepartmentId)
CREATE INDEX IX_4 on dbo.Employee (LastName, FirstName) INCLUDE (HireDate, DepartmentId)
GO

INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1001, 'Employee A', 3001001)
INSERT INTO Employee (EmployeeID, LastName, CostCenterNumber) VALUES(1002, 'Employee B', 3001001)
GO
```

Application

You have an application that updates the Employees table. The application calls the following stored procedures simultaneously and asynchronously:



UspA: This stored procedure updates only the EmployeeStatus column.

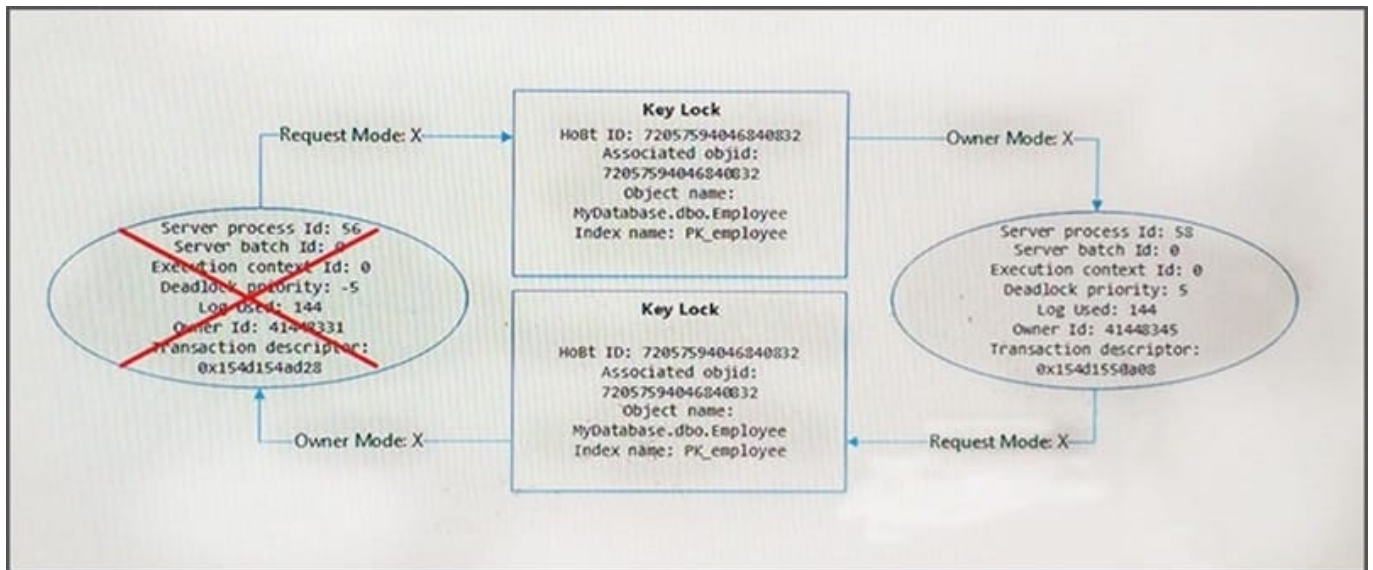
UspB: This stored procedure updates only the EmployeePayRate column.

The application uses views to control access to data. Views must meet the following requirements:

Allow user access to all columns in the tables that the view accesses.

Restrict updates to only the rows that the view returns.

Exhibit



You are analyzing the performance of the database environment. You discover that locks that are held for a long period of time as the reports are generated.

You need to generate the reports more quickly. The database must not use additional resources.

What should you do?

- A. Update all FROM clauses of the DML statements to use the IGNORE_CONSTRAINTS table hint.
- B. Modify the report queries to use the UNION statement to combine the results of two or more queries.
- C. Apply a nonclustered index to all tables used in the report queries.
- D. Update the transaction level of the report query session to READ UNCOMMITTED.

Correct Answer: D

Transactions running at the READ UNCOMMITTED level do not issue shared locks to prevent other transactions from modifying data read by the current transaction. This is the least restrictive of the isolation levels. References: [https://technet.microsoft.com/en-us/library/ms173763\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms173763(v=sql.105).aspx)

QUESTION 4

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not



appear in the review screen.

You need to collect data from the following two sources:

1.

The performance counters of the operating system

2.

Microsoft SQL Server events

You must analyze the two datasets side-by side by using a single tool.

Solution: You use dynamic management views and Data Collector Sets (DCs) in Performance Monitor to collect performance data. You use SQL Server Management Studio (SSMS) to analyze the data.

Does this meet the goal?

A. Yes

B. No

Correct Answer: B

Reference: <https://www.sqlshack.com/dba-guide-sql-server-performance-troubleshooting-part-2-monitoring-utilities/>

QUESTION 5

Note: The question is part of a series of questions that use the same or similar answer choices. An answer choice may be correct for more than one question in the series. Each question is independent of the other question in the series. Information and details provided in a question apply only to that question.

You have a database named DB1. The database does not use a memory-optimized filegroup. The database contains a table named Table1. The table must support the following workloads: You need to add the most efficient index to support the new OLTP workload, while not deteriorating the existing Reporting query performance. What should you do?

Workload	Type	Description
Reporting	Existing	The reporting workload must scan most of the records in the table to aggregate on a number of columns. A clustered columnstore index is already created on the table to support this workload.
OLTP	New	The OLTP workload must support 3,000 transactions per second. Rows are identified by using two columns. The filter is variant on one of the two columns while constant on the other. Only a small number of records with a few columns are returned by the query.

A. Create a clustered index on the table.

B. Create a nonclustered index on the table.

C. Create a nonclustered filtered index on the table.



- D. Create a clustered columnstore index on the table.
- E. Create a nonclustered columnstore index on the table.
- F. Create a hash index on the table.

Correct Answer: C

A filtered index is an optimized nonclustered index, especially suited to cover queries that select from a well-defined subset of data. It uses a filter predicate to index a portion of rows in the table. A well-designed filtered index can improve query performance, reduce index maintenance costs, and reduce index storage costs compared with full-table indexes.

References:[https://technet.microsoft.com/en-us/library/cc280372\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/cc280372(v=sql.105).aspx)

[Latest 70-762 Dumps](#)

[70-762 Study Guide](#)

[70-762 Braindumps](#)