

# Interview Questions

## Navigation

### [Introduction](#)

### [C#, VB.NET, ASP.NET](#)

[50 Important Silverlight Question/Answers](#)

[ASP.NET interview Question](#)

[asp.net interview questions](#)

[ASP.Net Interview Questions and Answers](#)

[C Sharp Interview Questions/Answers Part-1](#)

[C Sharp Interview Questions/Answers Part-2](#)

[CSS Interview Questions](#)

[Difference between ADO and ADO.NET](#)

[Difference between AJAX and Silverlight](#)

[Difference between ASP.NET 3.5 and ASP.NET 4.0](#)

[Difference between CSS and Themes](#)

[Difference between CTS and CLS](#)

[Difference between DataContractSerializer and XMLSerializer](#)

[Difference between DataTable and DataSet](#)

[Difference between DataView and DataTable](#)

[Difference between DTS and SSIS](#)

[Difference between EnableViewState and ViewStateMode properties](#)

[Data Model](#) >

## SQL Server Interview Questions/Answers Part-2

### What is a View in SQL Server?

You can think of a view either as a compiled sql query or a virtual table. As a view represents a virtual table, it does not physically store any data. When you query a view, you actually retrieve the data from the underlying base tables.

### What are the advantages of using views?

Or

### When do you usually use views?

1. Views can be used to implement row level and column level security.
2. Simplify the database schema to the users. You can create a view based on multiple tables which join columns from all these multiple tables so that they look like a single table.
3. Views can be used to present aggregated and summarized data.

### Can you create a view based on other views?

**Yes**, you can create a view based on other views. Usually we create views based on tables, but it also possible to create views based on views.

### Can you update views?

**Yes**, views can be updated. However, updating a view that is based on multiple tables, may not update the underlying tables correctly. To correctly update a view that is based on multiple tables you can make use INSTEAD OF triggers in SQL Server.

### What are indexed views?

Or

### What are materialized views?

A view is a virtual table, it does not contain any physical data. A view is nothing more than compiled SQL query. Every time, we issue a select query against a view, we actually get the data from the underlying base tables and not from the view, as the view itself does not contain

[Difference between Finalize\(\) and Dispose\(\) methods in .NET](#)

[Difference between HttpHandler and HttpModule](#)

[Difference between SOAP and RESTful Webservices](#)

[Difference between STA and MTA](#)

[Difference between Typed DataSet and Untyped DataSet](#)

[Difference between URL and URI](#)

[jQuery Interview Question Answer](#)

[SharePoint Interview question/answers](#)

[Silverlight Interview Questions Answers](#)

[WFC interview question-answer](#)

[What are the Differences between Abstract Class and Interface?](#)

[What are the differences between Structure and Class?](#)

[What is the difference between ref and out parameters?](#)

## Data Model

[Difference between SQL Server 2008 and SQL Server 2012](#)

[Difference between SQL Server 2008 R2 and SQL Server 2012](#)

[SQL Server Interview Questions/Answers Part-1](#)

[SQL Server Interview Questions/Answers Part-2](#)

[What's difference between Repository and Dataset](#)

## Feedback

any data.

When you create an index on a view, the data gets physically stored in the view. So, when we issue a select query against an indexed view, the data is retrieved from the index without having to go to the underlying table, which will make the select statement to work slightly faster. However, the disadvantage is, INSERT, UPDATE and DELETE operations will become a little slow, because every time you insert or delete a row from the underlying table, the view index needs to be updated. Inshort, DML operations will have negative impact on performance. Oracle refers to indexed views as materialized views.

Only the views created with schema binding, can have an Index. Simply adding WITH SCHEMABINDING to the end of the CREATE VIEW statement will accomplish this. However, the effect is that any changes to the underlying tables which will impact the view are not allowed. Since the indexed view is stored physically, any schema changes would impact the schema of the stored results set. Therefore, SQL Server requires that schema binding be used to prevent the view's schema (and therefore the underlying tables) from changing.

The first index for a view must be a UNIQUE CLUSTERED INDEX, after which, it's possible to create non-clustered indexes against the view.

Indexed Views are heavily used in data warehouses and reporting databases that are not highly transactional.

## What are the limitations of a View?

1. You cannot pass parameters to a view.
2. Rules and Defaults cannot be associated with views.
3. The ORDER BY clause is invalid in views unless TOP or FOR XML is also specified.
4. Views cannot be based on temporary tables.

## [b]What are the different types of joins available in sql server?[/b]

There are 3 different types of joins available in sql server, and they are

1. Cross Join
2. Inner Join or Join
3. Outer Join

## Outer Join is again divided into 3 types as shown below.

1. Left Outer Join or Left Join
2. Right Outer Join or Right Join
3. Full Outer Join or Full Join

**MVC**[Common Questions](#)[MVC Interview  
Questions/Answers](#)[What's difference between  
MVC and ASP.NET life cycle](#)**WCF**[WCF Interview Questions and  
Answer](#)[WCF – Interview Questions  
with Answers](#)**Sitemap**

You might have heard about self join, but self join is not a different type of join. A self join means joining a table with itself. We can have an inner self join or outer self join.

**Advantages of stored procedures**

This is a very common sql server interview question. There are several advantages of using stored procedures over adhoc queries, as listed below.

- 1. Better Performance :** Stored procedures are precompiled and hence run much faster than adhoc queries
- 2. Better Security :** Applications making use of dynamically built adhoc sql queries are highly susceptible to sql injection attacks, where as Stored Procedures can avoid SQL injection attacks completely.
- 3. Reduced Network Traffic:** Stored procedures can reduce network traffic to a very great extent when compared with adhoc sql queries. With stored procedures, you only need to send the name of the procedure between client and server. Imagine the amount of network bandwidth that can be saved especially if the stored procedure contains 1000 to 2000 lines of SQL.
- 4. Better Maintainance and Reusability:** Stored procedures can be used any where in the application. It is easier to maintain a stored procedure that is used on several pages as the modifications just need to be changed at one place where the stored procedure is defined. On the other hand, maintaining an adhoc sql query that's used on several pages is tedious and error prone, as we have to make modifications on each and every page.

**Comments**

You do not have permission to add comments.

[Sign in](#) | [Recent Site Activity](#) | [Report Abuse](#) | [Print Page](#) | Powered By [Google Sites](#)