

Stored Procedures vs Functions vs Views vs CTE

Overview

This presentation compares the main SQL Server objects: - Stored Procedures - Functions - Views - Common Table Expressions (CTE)

We compare them by features, performance, complexity, and real use cases.





Feature Comparison

| Feature | Stored Procedure | Function | View | CTE |
|----------------------|------------------------|-----------------|-----------------|------------|
| Returns Value | Optional (Result Sets) | Required | No | No |
| Used in SELECT | ✗ No | ✓ Yes | ✓ Yes | ✓ Yes |
| Modify Data | ✓ Yes | ✗ No | ✗ No | ✗ No |
| Accept Parameters | ✓ Yes | ✓ Yes | ✗ No | ✗ No |
| Multiple Result Sets | ✓ Yes | ✗ No | ✗ No | ✗ No |
| Scope | Database Object | Database Object | Database Object | Query Only |

Time & Space Complexity

| Object | Time Complexity | Space Complexity |
|------------------|-------------------------------|------------------|
| Stored Procedure | Optimized, depends on indexes | Medium |
| Function | Slower inside SELECT | Low |
| View | Same as base query | Very Low |
| CTE | Recomputed every execution | Very Low |

Performance (After First Execution)

| Object | Execution Plan | Performance |
|------------------|-----------------|---|
| Stored Procedure | Cached |  Best |
| Function | Partial caching |  |
| View | Inline query |  |
| CTE | No caching |  |

Stored Procedure

When to Use

- Business logic
- Insert / Update / Delete
- High performance
- Backend APIs

Example

```
CREATE PROCEDURE GetActiveUsers
AS
BEGIN
    SELECT * FROM Users WHERE IsActive = 1;
END;
```

Function

When to Use

- Calculations
- Reusable logic
- Inside SELECT statements

Example

```
CREATE FUNCTION GetUserAge (@BirthDate DATE)
RETURNS INT
AS
BEGIN
```

```
RETURN DATEDIFF(YEAR, @BirthDate, GETDATE());  
END;
```

View

When to Use

- Simplify complex queries
- Security (hide columns)
- Reporting & dashboards

Example

```
CREATE VIEW v_ActiveUsers  
AS  
SELECT Id, Name FROM Users WHERE IsActive = 1;
```

CTE (Common Table Expression)

When to Use

- Temporary results
- Recursive queries
- Improve readability

Example

```
WITH ActiveUsers AS (  
    SELECT Id, Name FROM Users WHERE IsActive = 1  
)  
SELECT * FROM ActiveUsers;
```

Quick Decision Guide

| Scenario | Best Choice |
|------------------------------|------------------|
| Business Logic & Performance | Stored Procedure |
| Calculations in SELECT | Function |

| Scenario | Best Choice |
|-----------------------------|-------------|
| Read-only Data Abstraction | View |
| Temporary / Recursive Query | CTE |

Final Summary

- **Stored Procedure:** Best performance & logic
 - **Function:** Calculations only
 - **View:** Data abstraction
 - **CTE:** Query readability
-

Thank You