

1 Stored procedure

Definition. Stored procedures are precompiled database queries that improve the security, efficiency and usability of database client/server applications.

Benefits. The major benefits of this technology are the substantial performance gains from precompiled execution, development efficiency gains from code reuse and abstraction and the security controls inherent in granting users permissions on specific stored procedures instead of the underlying database tables.

Views. Database views allow you to create "virtual tables" generated on the fly that pulls data from one or more tables. Views are often used to alleviate security concerns by providing users with access to a certain view of a database table without providing access to the underlying table itself.

Content	Views	Stored procedures
Accept parameters	No	Yes
Can be used as a building block in large query	Yes	No
Number of queries	One	Several
Perform modification to tables	No	Yes
Target for Insert, Update, Delete queries	Yes	No

2 Practical part

Syntax of creating a stored procedure:

```
1 CREATE PROCEDURE procedure_name
2   @parameter1 <data type>,
3   @parameter2 <data type>
4 AS
5 BEGIN
6   <procedure_body>
7 END
```

Stored procedure without parameters shows all clients whoses family names are "Mouse"

```
1 CREATE PROCEDURE usp_GetWithoutParameters
2 AS
3 BEGIN
4   SELECT * FROM dbo.Staff
5   WHERE FamilyName='Mouse';
6 END
```

Stored procedure with parameters shows all clients whoses family names match a certain string given by parameter @FaNa.

```
1 CREATE PROCEDURE usp_GetWithParameters
2   @FaNa nvarchar(50)
3 AS
4 BEGIN
5   SELECT * FROM dbo.Staff WHERE FamilyName=@FaNa;
6 END
```

Stored procedure with parameters shows all clients whoses family names contain a specified string.

```
1 CREATE PROCEDURE usp_GetWithLIKE
2   @FaNa nvarchar(50)
3 AS
4 BEGIN
5   SELECT * FROM dbo.Staff WHERE FamilyName LIKE '%' + @FaNa + '%';
6 END
```

This procedure shows all clients whose salary is between @MinSalary and @MaxSalary.

```

1 CREATE PROCEDURE usp_GetWithinRange
2     @MinSalary decimal(8,2),
3     @MaxSalary decimal(8,2)
4 AS
5 BEGIN
6     SELECT * FROM dbo.Staff WHERE Salary BETWEEN @MinSalary AND @MaxSalary;
7 END

```

Default value is specified for @MinSalary so that if a value is not given by user then 0 is automatically assigned to @MinSalary. Thus, the results is the table contains all clients that has salary no more than @MaxSalary.

```

1 CREATE PROCEDURE usp_GetWithMax
2     @MaxSalary decimal(8,2),
3     @MinSalary decimal(8,2)=0
4 AS
5 BEGIN
6     SELECT * FROM dbo.Staff
7     WHERE (@MinSalary IS NULL OR Salary>=@MinSalary)
8     AND (@MaxSalary IS NULL OR Salary<=@MaxSalary);
9 END

```