

# CREATE VIEW

Una Vista è una tabella dinamica analoga ad una query

Viene calcolata attraverso l'esecuzione di una query ed il risultato costituisce però una vera e propria tabella

```
CREATE VIEW NomeVista (col1 , col2, ...) AS  
(  
    SELECT ...  
);
```

# Stored Procedures

Una Stored Procedure è un codice SQL che è possibile salvare e che quindi può essere riutilizzato più volte.

Se si dispone di una query SQL che si scrive più e più volte, conviene salvarla come Stored Procedure.

È inoltre possibile passare dei parametri a una Stored Proc e definire un valore ritornato.

# Stored Procedures

Sono simili alle procedure di altri linguaggi di programmazione in quanto sono in grado di:

- Accettare parametri di input e restituire più valori sotto forma di parametri di output alla procedura o al batch che esegue la chiamata.
- Includere istruzioni di programmazione che eseguono le operazioni nel database, tra cui la chiamata di altre procedure.
- Restituire un valore di stato a una procedura o a un batch che esegue la chiamata per indicare l'esito positivo o negativo (e il motivo dell'esito negativo).

# Stored Procedures

Per creare una Stored Procedure:

```
CREATE PROCEDURE procedure_name @param1 type, @param2 type, ...  
    AS  
    sql_statement ...  
    RETURN value;  
    ...  
GO
```

Per invocare una Stored Procedure:

```
@retval = EXEC procedure_name @param1=value, ...;
```

L'invocazione può avvenire specificando i parametri oppure affidandosi alla loro posizione.

# Demo

Stored Procedures



# Funzioni

- SQL Server ha molte funzioni integrate.
- Vengono utilizzate per manipolare
  - Stringhe (CONCAT, FORMAT ...)
  - Numeri (FLOOR, CEILING, ROUND ...)
  - Date (DATEDIFF, DATEPART ...)
  - Effettuare conversioni (CAST, CONVERT, ...)
  - ... and many more!

# Funzioni

Funzione	Sintassi	Valore restituito	Tipo reso
<a href="#">SYSDATETIME</a>	SYSDATETIME ()	Restituisce il valore <b>datetime2(7)</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La differenza di fuso orario non è inclusa.	datetime2(7)
<a href="#">SYSDATETIMEOFFSET</a>	SYSDATETIMEOFFSET ()	Restituisce il valore <b>datetimeoffset(7)</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La differenza di fuso orario è inclusa.	datetimeoffset(7)
<a href="#">SYSUTCDATETIME</a>	SYSUTCDATETIME ()	Restituisce il valore <b>datetime2(7)</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La data e l'ora vengono restituite in formato ora UTC (Coordinated Universal Time).	datetime2(7)
<a href="#">CURRENT_TIMESTAMP</a>	CURRENT_TIMESTAMP	Restituisce il valore <b>datetime</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La differenza di fuso orario non è inclusa.	datetime
<a href="#">GETDATE</a>	GETDATE ()	Restituisce il valore <b>datetime</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La differenza di fuso orario non è inclusa.	datetime
<a href="#">GETUTCDATE</a>	GETUTCDATE ( )	Restituisce il valore <b>datetime</b> che contiene la data e l'ora del computer in cui è in esecuzione l'istanza di SQL Server. La data e l'ora vengono restituite in formato ora UTC (Coordinated Universal Time).	datetime
<a href="#">DATENAME</a>	DATENAME ( <i>datepart</i> , <i>date</i> )	Restituisce una stringa di caratteri che rappresenta la <i>datepart</i> della data specificata.	nvarchar
<a href="#">DATEPART</a>	DATEPART ( <i>datepart</i> , <i>date</i> )	Restituisce un valore <b>intero</b> che rappresenta il valore <i>datepart</i> dell'argomento <i>date</i> specificato.	int
<a href="#">DAY</a>	DAY ( <i>date</i> )	Restituisce un valore <b>intero</b> che rappresenta la parte del giorno della <i>date</i> specificata.	int
<a href="#">MONTH</a>	MONTH ( <i>date</i> )	Restituisce un valore <b>intero</b> che rappresenta la parte mese di un valore <i>date</i> specificato.	int
<a href="#">YEAR</a>	YEAR ( <i>date</i> )	Restituisce un valore <b>intero</b> che rappresenta la parte dell'anno di <i>date</i> specificata.	int

# Funzioni

<a href="#">DATEFROMPARTS</a>	DATEFROMPARTS ( <i>year, month, day</i> )	Restituisce un valore di tipo <b>date</b> per l'anno, il mese e il giorno specificati.	date
<a href="#">DATETIME2FROMPARTS</a>	DATETIME2FROMPARTS ( <i>year, month, day, hour, minute, seconds, fractions, precision</i> )	Restituisce un valore <b>datetime2</b> per la data e l'ora specificate e con la precisione indicata.	datetime2 ( <i>precision</i> )
<a href="#">DATETIMEFROMPARTS</a>	DATETIMEFROMPARTS ( <i>year, month, day, hour, minute, seconds, milliseconds</i> )	Restituisce un valore di tipo <b>datetime</b> per la data e l'ora specificate.	datetime
<a href="#">DATETIMEOFFSETFROMPARTS</a>	DATETIMEOFFSETFROMPARTS ( <i>year, month, day, hour, minute, seconds, fractions, hour_offset, minute_offset, precision</i> )	Restituisce un valore <b>datetimeoffset</b> per la data e l'ora specificata e con gli offset e la precisione indicati.	datetime ( <i>precision</i> )
<a href="#">SMALLDATETIMEFROMPARTS</a>	SMALLDATETIMEFROMPARTS ( <i>year, month, day, hour, minute</i> )	Restituisce un valore di tipo <b>smalldatetime</b> per la data e l'ora specificate.	smalldatetime
<a href="#">TIMEFROMPARTS</a>	TIMEFROMPARTS ( <i>hour, minute, seconds, fractions, precision</i> )	Restituisce un valore <b>time</b> per l'ora specificate e con la precisione indicata.	time ( <i>precision</i> )
<a href="#">DATEDIFF</a>	DATEDIFF ( <i>datepart, startdate, enddate</i> )	Restituisce il numero di limiti di <i>datepart</i> della data e ora che si sovrappongono tra due date specificate.	int



# Funzioni Utente

Una funzione definita dall'utente è una routine Transact-SQL

- accetta parametri,
- viene effettuata un'azione (ad esempio un calcolo complesso),
- viene restituito il risultato di tale azione sotto forma di valore.

Il valore restituito può essere un valore scalare (singolo) o una tabella.

# Funzioni

Per creare una Funzione Scalare:

```
CREATE FUNCTION [ schema. ] NOME_FUNZIONE  
    ( elenco dei parametri )  
RETURNS tipo_reso  
AS BEGIN  
    --corpo della funzione  
    RETURN espressione_scalare  
END
```

Per creare una Funzione Tabellare:

```
CREATE FUNCTION [ schema. ] NOME_FUNZIONE  
    ( elenco dei parametri )  
RETURNS TABLE  
AS BEGIN  
    --corpo della funzione  
    RETURN comando_SELECT  
END
```

# Funzioni - Parametri

I nomi dei parametri devono iniziare con @ e sono seguiti dal tipo e dalle opzioni che sono dei vincoli. Il nome del parametro deve essere conforme alle regole per gli identificatori.

Poiché i parametri sono locali rispetto alla procedura, è possibile utilizzare gli stessi nomi di parametro in altre procedure.

Una volta definita la funzione per poterla richiamare si può utilizzare la seguente sintassi:

```
SELECT column1, dbo.ufnprocedure_name(param1) AS alias  
FROM table1
```

# Demo

Function

User-defined Function

