

Gestione degli Errori

Il costrutto `TRY CATCH` consente di gestire le eccezioni in SQL Server.

Per utilizzare il costrutto `TRY CATCH`

- inserire un gruppo di istruzioni SQL, che potrebbero causare un'eccezione, in un blocco `BEGIN TRY ... END TRY`
- Quindi si utilizza un blocco `BEGIN CATCH ... END CATCH` immediatamente dopo il blocco `TRY`

Gestione degli Errori

All'interno del blocco CATCH è possibile utilizzare le seguenti funzioni per ottenere informazioni dettagliate sull'errore che si è verificato:

- `ERROR_LINE()`
- `ERROR_MESSAGE()`
- `ERROR_PROCEDURE()`
- `ERROR_NUMBER()`
- `ERROR_SEVERITY()`
- `ERROR_STATE()`

Gestione degli Errori

```
BEGIN TRY

... sql_statements ...

END TRY
BEGIN CATCH

SELECT  ERROR_NUMBER(),ERROR_SEVERITY(),
        ERROR_STATE(),ERROR_PROCEDURE(),ERROR_LINE(),ERROR_MESSAGE();

END CATCH
```

Transazioni

Una transazione è un'unità di lavoro che va trattata come "un tutto". Deve avvenire per intero o per niente.

Un esempio classico è il trasferimento di denaro da un conto bancario a un altro:

- prelevare l'importo dall'account di origine
- quindi depositarlo sull'account di destinazione

L'operazione deve riuscire a pieno. Se ci si ferma a metà strada, i soldi andranno persi ...

```
beginTransaction;  
  
accountB += 100;  
accountA -= 100;  
  
endTransaction;
```

Transazioni

Le transazioni sono caratterizzate da quattro proprietà chiamate proprietà **ACID**. Per superare questo test ACID, una transazione deve essere Atomica, Coerente, Isolata e Durevole.

- *Atomico*: Tutti i passaggi della transazione dovrebbero avere esito positivo o negativo insieme
- *Consistenza*: La transazione porta il database da uno stato stabile a un nuovo stato stabile
- *Isolamento*: Ogni transazione è un'entità indipendente
- *Durevolezza*: i risultati delle transazioni impegnate sono permanenti

Transazioni

Sono possibili solo due esiti di una transizione:

- **COMMIT:** l'intera unità di lavoro è stata completata con successo. Tutte le modifiche applicate ai dati vengono confermate e il database passa con successo ad un nuovo stato 'stabile'
- **ROLLBACK:** uno o più operazioni dell'unità di lavoro sono fallite. Tutte le operazione completate con successo vengono annullate e il database ritorna allo stato 'stabile' iniziale

Transazioni

Le transazioni esplicite iniziano con l'istruzione BEGIN TRANSACTION e terminano con l'istruzione COMMIT o ROLLBACK

```
BEGIN TRANSACTION  
[ transaction_name | @tran_name_variable  
  [ WITH MARK [ 'description' ] ] ]
```

... sql statements ...

```
COMMIT;
```

```
ROLLBACK;
```

Transazioni

Per utilizzare una Transazione in una Stored Procedure:

```
CREATE PROCEDURE procedure_name @param1 type, @param2 type, ...  
AS  
BEGIN  
    BEGIN TRAN  
    BEGIN TRY  
        ... sql_statements ...  
        IF @@ERROR ROLLBACK;  
        COMMIT;  
    END TRY  
    BEGIN CATCH  
        ROLLBACK;  
    END CATCH  
END
```


Demo

Transaction

