--simple procedure to check exception handling

create proc Selectdata(@n1 int , @n2 int , @n3 int output)

AS

begin

begin try

set @n3 = @n1 / @n2

end try

begin catch

select error\_number() as errornumber

,ERROR\_SEVERITY() AS ErrorSeverity

,ERROR\_STATE() AS ErrorState

,ERROR\_PROCEDURE() AS ErrorProcedure

,ERROR\_LINE() AS ErrorLine

,ERROR\_MESSAGE() AS ErrorMessage;

end catch

end

DECLARE @r decimal;

EXEC Selectdata 10, 0, @r output;

PRINT @r;

Exception handling in a stored procedure using tables

CREATE TABLE sales.persons

(

person\_id INT

PRIMARY KEY IDENTITY,

first\_name NVARCHAR(100) NOT NULL,

last\_name NVARCHAR(100) NOT NULL

)

CREATE TABLE sales.deals

(

deal\_id INT

PRIMARY KEY IDENTITY,

person\_id INT NOT NULL,

deal\_note NVARCHAR(100),

FOREIGN KEY(person\_id) REFERENCES sales.persons(

person\_id)

);

insert into

sales.persons(first\_name, last\_name)

values

('John','Doe'),

('Jane','Doe');

insert into

sales.deals(person\_id, deal\_note)

values

(1,'Deal for John Doe');

select \* from sales.persons

select \* from sales.deals

alter PROC usp\_delete\_person(

@person\_id INT

) AS

BEGIN

BEGIN TRY

BEGIN TRANSACTION;

-- delete the person

DELETE FROM sales.persons

WHERE person\_id = @person\_id;

-- if DELETE succeeds, commit the transaction

COMMIT TRANSACTION;

END TRY

BEGIN CATCH

-- report exception

exec usp\_report\_error;

-- Test if the transaction is uncommittable.

IF (XACT\_STATE()) = -1

BEGIN

PRINT N'The transaction is in an uncommittable state.' +

'Rolling back transaction.'

ROLLBACK TRANSACTION;

END;

-- Test if the transaction is committable.

IF (XACT\_STATE()) = 1

BEGIN

PRINT N'The transaction is committable.' +

'Committing transaction.'

COMMIT TRANSACTION;

END;

END CATCH

END;

GO

create proc usp\_report\_error

AS

begin

select error\_number() as errornumber

,ERROR\_SEVERITY() AS ErrorSeverity

,ERROR\_STATE() AS ErrorState

,ERROR\_PROCEDURE() AS ErrorProcedure

,ERROR\_LINE() AS ErrorLine

,ERROR\_MESSAGE() AS ErrorMessage;

end

EXEC usp\_delete\_person 1;

--- RaiseError

EXEC sp\_addmessage

@msgnum = 50005,

@severity = 1,

@msgtext = 'A custom error message';

To verify the insert, you use the following query:

**SELECT**

\*

**FROM**

sys.messages

**WHERE**

message\_id = 50005;

To use this message\_id, you execute the RAISEERROR statement as follows:

DECLARE @MessageText NVARCHAR(100);

SET @MessageText = 'Cannot delete the sales order %s';

RAISERROR(

@MessageText, -- Message text

16, -- severity

1, -- state,

'2001' -- first argument to the message text

) with log

alter proc dividebyZero(@no1 int, @no2 int)

As

begin

declare @res int

set @res =0

Begin try

if @no2= 1

RAISERROR ('Please enter no more than 1',0,1)

Set @res = @no1 / @no2

print 'Result'

end try

begin catch

print error\_number()

print error\_message()

end catch

end