异常

     在程序中，有时候完成一些Transact-SQL会出现错误、异常信息。如果我们想自己处理这些异常信息的话，需要手动捕捉这些信息。那么我们可以利用try catch完成。

TRY…CATCH 构造包括两部分：一个 TRY 块和一个 CATCH 块。如果在 TRY 块中所包含的 Transact-SQL 语句中检测到错误条件，控制将被传递到 CATCH 块（可在此块中处理该错误）。

     CATCH 块处理该异常错误后，控制将被传递到 END CATCH 语句后面的第一个 Transact-SQL 语句。如果 END CATCH 语句是存储过程或触发器中的最后一条语句，控制将返回到调用该存储过程或触发器的代码。将不执行 TRY 块中生成错误的语句后面的 Transact-SQL 语句。

     如果 TRY 块中没有错误，控制将传递到关联的 END CATCH 语句后紧跟的语句。如果 END CATCH 语句是存储过程或触发器中的最后一条语句，控制将传递到调用该存储过程或触发器的语句。

     TRY 块以 BEGIN TRY 语句开头，以 END TRY 语句结尾。在 BEGIN TRY 和 END TRY 语句之间可以指定一个或多个 Transact-SQL 语句。CATCH 块必须紧跟 TRY 块。CATCH 块以 BEGIN CATCH 语句开头，以 END CATCH 语句结尾。在 Transact-SQL 中，每个 TRY 块仅与一个 CATCH 块相关联。

     # 错误函数

TRY...CATCH 使用错误函数来捕获错误信息。  
 ERROR\_NUMBER() 返回错误号。  
 ERROR\_MESSAGE() 返回错误消息的完整文本。此文本包括为任何可替换参数（如长度、对象名称或时间）提供的值。  
 ERROR\_SEVERITY() 返回错误严重性。  
 ERROR\_STATE() 返回错误状态号。  
 ERROR\_LINE() 返回导致错误的例程中的行号。  
 ERROR\_PROCEDURE() 返回出现错误的存储过程或触发器的名称。

     示例

--错误消息存储过程  
if (object\_id('proc\_error\_info') is not null)  
 drop procedure proc\_error\_info  
go  
create proc proc\_error\_info  
as  
 select   
 error\_number() '错误编号',  
 error\_message() '错误消息',  
 error\_severity() '严重性',  
 error\_state() '状态好',  
 error\_line() '错误行号',  
 error\_procedure() '错误对象(存储过程或触发器)名称';  
go

     # 示例：用异常处理错误信息

--简单try catch示例  
begin try  
 select 1 / 0;  
end try  
begin catch  
 exec proc\_error\_info; --调用错误消息存储过程  
end catch  
go

     # 示例：异常能处理的错误信息

--  
--简单try catch示例，无法处理错误  
begin try  
 select \* \* from student;  
end try  
begin catch  
 exec proc\_error\_info;  
end catch  
go  
--  
--简单try catch示例，不处理错误(不存在的表对象)  
begin try  
 select \* from st;  
end try  
begin catch  
 exec proc\_error\_info;  
end catch  
go  
--  
--异常处理，能处理存储过程（触发器）中（不存在表对象）的错误信息  
if (object\_id('proc\_select') is not null)  
 drop procedure proc\_select  
go  
create proc proc\_select  
as  
 select \* from st;  
go  
begin try  
 exec proc\_select;  
end try  
begin catch   
 exec proc\_error\_info;  
end catch  
go

     异常不能处理编译期的错误，如语法错误。以及重编译造成部分名称对象得不到正确解析的时候所出现的错误。

     # 示例：无法提交的事务

--创建临时用表  
if (object\_id('temp\_tab', 'u') is not null)  
 drop table temp\_tab  
go  
create table temp\_tab(  
 id int primary key identity(100000, 1),  
 name varchar(200)  
)  
go  
  
begin try  
 begin tran;  
 --没有createTime字段  
 alter table temp\_tab drop column createTime;  
 commit tran;  
end try  
begin catch  
 exec proc\_error\_info;--显示异常信息  
 if (xact\_state() = -1)  
 begin  
 print '会话具有活动事务，但出现了致使事务被归类为无法提交的事务的错误。'  
 + '会话无法提交事务或回滚到保存点；它只能请求完全回滚事务。'  
 + '会话在回滚事务之前无法执行任何写操作。会话在回滚事务之前只能执行读操作。'  
 + '事务回滚之后，会话便可执行读写操作并可开始新的事务。';  
 end  
 else if (xact\_state() = 0)  
 begin  
 print '会话没有活动事务。';  
 end  
 else if (xact\_state() = 1)  
 begin  
 print '会话具有活动事务。会话可以执行任何操作，包括写入数据和提交事务。';  
 end  
end catch  
go

     # 示例：处理异常日志信息

--  
---异常、错误信息表  
if (object\_id('errorLog', 'U') is not null)  
 drop table errorLog  
go  
create table errorLog(  
 errorLogID int primary key identity(100, 1), --ErrorLog 行的主键。  
 errorTime datetime default getDate(), --发生错误的日期和时间。  
 userName sysname default current\_user, --执行发生错误的批处理的用户。  
 errorNumber int, --发生的错误的错误号。  
 errorSeverity int, --发生的错误的严重性。  
 errorState int, --发生的错误的状态号。  
 errorProcedure nvarchar(126), --发生错误的存储过程或触发器的名称。  
 errorLine int, --发生错误的行号。  
 errorMessage nvarchar(4000)  
)  
go  
--  
--存储过程：添加异常日志信息  
if (object\_id('proc\_add\_exception\_log', 'p') is not null)  
 drop proc proc\_add\_exception\_log  
go  
create proc proc\_add\_exception\_log(@logId int = 0 output)  
as  
begin  
 set nocount on;  
 set @logId = 0;  
 begin try  
 if (error\_number() is null)  
 return;  
   
 if (xact\_state() = -1)  
 begin  
 print '会话具有活动事务，但出现了致使事务被归类为无法提交的事务的错误。'  
 + '会话无法提交事务或回滚到保存点；它只能请求完全回滚事务。'  
 + '会话在回滚事务之前无法执行任何写操作。会话在回滚事务之前只能执行读操作。'  
 + '事务回滚之后，会话便可执行读写操作并可开始新的事务。';  
 end  
 else if (xact\_state() = 0)  
 begin  
 print '会话没有活动事务。';  
 end  
 else if (xact\_state() = 1)  
 begin  
 print '会话具有活动事务。会话可以执行任何操作，包括写入数据和提交事务。';  
 end  
   
 --添加日志信息  
 insert into errorLog values(getDate(),   
 current\_user, error\_number(),   
 error\_severity(), error\_state(),   
 error\_procedure(),   
 error\_line(), error\_message());  
 --设置自增值  
 select @logId = @@identity;  
 end try  
 begin catch  
 print '添加异常日志信息出现错误';  
 exec proc\_error\_info;--显示错误信息  
 return -1;  
 end catch  
end  
go  
--  
---处理异常信息示例  
declare @id int;  
begin try  
 begin tran;  
 --删除带有外键的记录信息  
 delete classes where id = 1;  
 commit tran;  
end try  
begin catch  
 exec proc\_error\_info;--显示错误信息  
 if (xact\_state() <> 0)  
 begin  
 rollback tran;  
 end  
 exec proc\_add\_exception\_log @id output  
end catch  
select \* from errorLog where errorLogID = @id;  
go

http://www.cnblogs.com/hoojo/archive/2011/07/19/2110325.html