```
create database person_info_lab_2
use [person_info_lab_2]

CREATE TABLE PERSON_LOG(
PLOGID                  INT PRIMARY KEY,
PERSONAME               VARCHAR(250),
OPERATION               VARCHAR(50),
UPDATEDDATE             DATETIME
)


--1. Print message like - Error Occur that is: Divide by zero error encountered.

BEGIN TRY
SELECT 1/0;
END TRY
BEGIN CATCH
SELECT 'ERROROCCUR THAT IS::--'+ERROR_MESSAGE() AS ERROR_MSG
END CATCH


--2. Print error message in insert statement using Error_Message () function: Conversion
failed when
--converting datetime from character string.

BEGIN TRY
DECLARE @DATETIME_VALUE VARCHAR(100) ='11/11/2011'
SELECT CONVERT(DATETIME,@DATETIME_VALUE,103) AS  'united kingdom time is::--'
END TRY
BEGIN CATCH
SELECT 'ERROROCCUR THAT IS::--'+ERROR_MESSAGE() AS UK
END CATCH


--3. Create procedure which prints the error message that “The PLogID is already taken.
Try another
--one”.


CREATE PROC PR_PRINT_MASSAGE
@PLOGID INT,
@PLOGNAME VARCHAR(50)
AS
BEGIN
       BEGIN TRY
              INSERT INTO PERSON_LOG VALUES (@PLOGID,@PLOGNAME,'INSERT',GETDATE())
       END TRY

       BEGIN CATCH
       PRINT 'THE PLOGID IS ALREADY TAKEN PLEASE TRY ANOTHER ONCE'
       END CATCH
END

EXEC PR_PRINT_MASSAGE 102,'NISHANT'
SELECT * FROM PERSON_LOG
```

```sql
--4. Create procedure that print the sum of two numbers: take both number as integer &
handle
--exception with all error functions if any one enters string value in numbers otherwise
print result.

       CREATE PROC ADDITION_TWO_NUMBERS
       @NUMBER2 INT,
       @NUMBER1 VARCHAR(2),
       @OUTPUT INT OUTPUT
       AS
       BEGIN
       BEGIN TRY
              SET @OUTPUT =@NUMBER1+@NUMBER2;
       END TRY

       BEGIN CATCH
              SELECT
                     ERROR_NUMBER()              AS     [ERROR_NUMBER],
                     ERROR_MESSAGE()             AS     [ERROR_MESSAGE],
                     ERROR_STATE()        AS     [ERROR_STATE],
                     ERROR_SEVERITY()     AS     [ERROR_SEVERITY],
                     ERROR_LINE()         AS     [ERROR_LINE],
                     ERROR_PROCEDURE()    AS     [ERROR_PROCEDURE];

                     END CATCH
END

DECLARE @RESULT INT ;
EXEC ADDITION_TWO_NUMBERS 3,'A',@RESULT OUTPUT;
PRINT @RESULT;


--5. Throw custom exception using stored procedure which accepts PLogID as input & that
throws
--Error like no plogid is available in database.

CREATE PROC FIND_PLOG_ID
@PLOGID INT
AS
BEGIN
       IF EXISTS(SELECT * FROM PERSON_LOG WHERE PLOGID=@PLOGID)
       PRINT('PLOG ID IS AVAILABLE IN DATABASE')

       ELSE
              THROW 50005,'ERROR!!!! NO PLOGID WITH THIS ID' ,1

END

EXEC FIND_PLOG_ID 1011


--6. Create cursor with name per_cursor which takes PLogID & PersonName as variable and
produce
--combine output with PLogID & Person Name.

DECLARE
       @PLOG_ID INT,
       @PERSONNAME VARCHAR(250);
```

```
DECLARE PERSON_CURSOR CURSOR

FOR SELECT
       PLOGID,
       PERSONAME

       FROM PERSON_LOG;

       OPEN PERSON_CURSOR

       FETCH NEXT FROM PERSON_CURSOR INTO
              @PLOG_ID,
              @PERSONNAME;

              WHILE @@FETCH_STATUS=0
       BEGIN
              PRINT CAST(@PLOG_ID AS VARCHAR) + '--->>' + @PERSONNAME;
              FETCH NEXT FROM PERSON_CURSOR INTO
                     @PLOG_ID,
                     @PERSONNAME;

       END;
       CLOSE PERSON_CURSOR;
       DEALLOCATE PERSON_CURSOR;



--7. Use Table Student (Id, Rno, EnrollmentNo, Name, Branch, University) - Create cursor
that updates
--enrollment column as combination of branch & Roll No. like SOE22CE0001 and so on. (22
is admission year)

CREATE TABLE STUDENT (
              ID                          INT,
              RNO                         INT,
              ENROLLMENTNO  VARCHAR (100),
              NAME                 VARCHAR(50),
              BRANCH               VARCHAR(50),
              UNIVERSITY           VARCHAR(50)
              )
              INSERT INTO STUDENT VALUES (1,001,'COEE','MAHESH','P.HD','HARWARD')
              INSERT INTO STUDENT VALUES (2,002,'COEM','RAMESH','DEGREE','DARSHAN')
              INSERT INTO STUDENT VALUES(3,003,'COEC','SURESH','MASTERS','MARWADI')

              ---------------------------------------------------

              DECLARE @ROLL_NO     INT,
                            @BRANCH              VARCHAR(250),
                            @ENROLLMENT VARCHAR(100);

              DECLARE STUDENT_CURSOR CURSOR
              FOR SELECT
                     RNO,
                     BRANCH
                     FROM STUDENT

              OPEN STUDENT_CURSOR
```

```
            FETCH NEXT FROM STUDENT_CURSOR INTO
                   @ROLL_NO,
                   @BRANCH;

                   WHILE @@FETCH_STATUS=0
                   BEGIN


                   UPDATE STUDENT SET ENROLLMENTNO = ('SOE' +'22' +@BRANCH
+CAST(@ROLL_NO AS VARCHAR) ) WHERE RNO=@ROLL_NO;
                   FETCH NEXT FROM STUDENT_CURSOR INTO
                   @ROLL_NO,
                   @BRANCH;

                   END

                   CLOSE STUDENT_CURSOR;
                   DEALLOCATE STUDENT_CURSOR;


                                 SELECT * FROM STUDENT
```