

## Exception Handling

### Part – A

1. **Handle Divide by Zero Error and Print message like: Error occurs that is - Divide by zero error.**

```
DECLARE @VAL1 INT;  
DECLARE @VAL2 INT;  
BEGIN TRY  
    SET @VAL1=8;  
    SET @VAL2=0; /* ERROR OCCUR HERE */  
    PRINT @VAL1 / @VAL2;  
END TRY  
BEGIN CATCH  
    PRINT 'ERROR OCCUR THAT IS: DIVIDE BY ZERO ERROR.'  
END CATCH
```

2. **Try to convert string to integer and handle the error using try...catch block.**

```
BEGIN TRY  
    SELECT CAST('abc' AS INT) -- try to convert a string to an integer  
END TRY  
BEGIN CATCH  
    SELECT  
        ERROR_NUMBER() AS ErrorNumber,  
        ERROR_MESSAGE() AS ErrorMessage  
END CATCH
```

3. **Create a procedure that prints the sum of two numbers: take both numbers as integer & handle exception with all error functions if any one enters string value in numbers otherwise print result.**

```
CREATE PROCEDURE CalculateSum  
    @num1 INT,  
    @num2 INT  
AS  
BEGIN  
    -- Check if both inputs are integers  
    IF ISNUMERIC(@num1) = 1 AND ISNUMERIC(@num2) = 1  
        PRINT @num1 + @num2;  
END;  
-- Handle the exception at the time of SP execution  
BEGIN TRY  
    EXEC CalculateSum 10, 'ABC'  
END TRY  
BEGIN CATCH  
    SELECT  
        ERROR_NUMBER() AS 'Error Number',  
        ERROR_MESSAGE() AS 'Error Message',
```

```
ERROR_LINE()      AS 'Error Line',  
ERROR_SEVERITY()  AS 'Error Severity',  
ERROR_PROCEDURE() AS 'Error Procedure',  
ERROR_STATE()     AS 'Error State'
```

```
END CATCH
```

4. **Handle a Primary Key Violation while inserting data into customers table and print the error details such as the error message, error number, severity, and state.**

```
BEGIN TRY  
    INSERT INTO customers (Customer_id, Customer_Name, Email)  
    VALUES (1, 'John Doe', 'jd@gmail.com');  
    -- Attempt to insert a duplicate primary key  
    INSERT INTO customers (Customer_id, Customer_Name, Email)  
    VALUES (1, 'Jane Smith', 'js@gmail.com');  
END TRY  
BEGIN CATCH  
    PRINT 'Error Message: ' + ERROR_MESSAGE();  
    PRINT 'Error Number: ' + CAST(ERROR_NUMBER() AS VARCHAR(10));  
    PRINT 'Error Severity: ' + CAST(ERROR_SEVERITY() AS VARCHAR(10));  
    PRINT 'Error State: ' + CAST(ERROR_STATE() AS VARCHAR(10));  
END CATCH;
```

5. **Throw custom exception using stored procedure which accepts Customer\_id as input & that throws Error like no Customer\_id is available in database.**

```
CREATE PROCEDURE SP_CustID  
    @Customer_ID INT  
AS  
BEGIN TRY  
    IF EXISTS (SELECT * FROM Customers WHERE Customer_id = @Customer_ID)  
        PRINT 'Customer ID is available'  
    ELSE  
        THROW 50002, 'No Customer ID is available in Database.', 1;  
END TRY  
BEGIN CATCH  
    PRINT 'Error Message: ' + ERROR_MESSAGE();  
END CATCH
```

#### Part – B

5. **Handle a Foreign Key Violation while inserting data into Orders table and print appropriate error message.**

```
BEGIN TRY  
    -- Attempting to insert an order with a non-existent customer ID  
    INSERT INTO Orders (Order_id, Customer_id, Order_date)  
    VALUES (1, 6, '2023-06-10');  
END TRY  
BEGIN CATCH
```

```
        PRINT 'Error: ' + ERROR_MESSAGE();  
    END CATCH;
```

**6. Throw custom exception that throws error if the data is invalid.**

```
CREATE PROCEDURE SP_Invalid_Data  
    @cid int,  
    @cname varchar(50),  
    @email varchar(50)  
AS  
BEGIN TRY  
    IF @cid < 0  
        THROW 50003, 'Invalid data', 2;  
    ELSE  
        INSERT INTO customers (Customer_id, Customer_Name, Email)  
        VALUES (@cid, @cname, @email);  
END TRY  
BEGIN CATCH  
    PRINT 'Error Message: ' + ERROR_MESSAGE();  
END CATCH
```

– try to insert negative Customer id at the time of SP execution

```
EXEC SP_Invalid_Data -8, 'Nilesh', 'nil_12@gmail.com'
```

**Part – C**

**7. Create a procedure which prints the error message that “The Customer\_id is already taken. Try another one”**

```
CREATE PROCEDURE PR_Customer_Insert  
    @Customer_id INT,  
    @Customer_Name VARCHAR(50),  
    @Email VARCHAR(100)  
AS  
BEGIN TRY  
    INSERT INTO Customers  
    VALUES (@Customer_id, @Customer_Name, @Email)  
END TRY  
BEGIN CATCH  
    PRINT 'The Customer_Id is already taken. Try another one'  
END CATCH
```