

ASSIGNMENT 3

1. Inserting values into 4 tables at once passing values through json using Stored Procedure.

- **USER TABLE**

```
CREATE OR ALTER PROCEDURE [dbo].[SpUserIns]
```

```
@Json NVARCHAR(MAX) OUT,
```

```
@UserId INT
```

```
AS
```

```
BEGIN
```

```
SET NOCOUNT ON;
```

```
If @UserId IS NULL
```

```
BEGIN
```

```
RAISERROR('UserId is required.', 16, 1)
```

```
RETURN
```

```
END
```

```
CREATE TABLE #User --creating temporary table
```

```
(
```

```
Id INT IDENTITY(1,1), --each table has an id
```

```
UserName VARCHAR(50),
```

```
Password VARCHAR(50),
```

```
Hotel NVARCHAR(MAX), --hotel is inside user, branch is inside hotel, and employee  
and customer are inside branch
```

```
Branch NVARCHAR(MAX),
```

```
Employee NVARCHAR(MAX),
```

```
Customer NVARCHAR(MAX)
```

);

INSERT INTO #User

```
(
  UserName,
  Password,
  Hotel,
  Branch,
  Employee,
  Customer
)
```

SELECT

obj.userName, obj.password, obj.hotel, obj.branch, obj.employee, obj.customer --objj is
key,
obj is value, objj is a level, user is key

FROM OPENJSON (@Json)

WITH(

[user] NVARCHAR(MAX) AS JSON

) as objj

Cross apply OPENJSON (objj.[User]) --user vitra jana ko we have no join here, so we do
cross apply WITH(

userName NVARCHAR(MAX),

[password] NVARCHAR(MAX),

hotel NVARCHAR(MAX) AS JSON, --as json because yo json parse

gareko branch NVARCHAR(MAX) AS JSON, employee

NVARCHAR(MAX) AS JSON,

customer NVARCHAR(MAX) AS JSON

) as obj;

CREATE TABLE

#out(id INT

IDENTITY(1,1),

UserId INT --user id preserve

)

INSERT INTO dbo.[User](--the original table

UserName,

[Password]

)

OUTPUT inserted.UserId INTO #out(UserId) --primary key k xa vanne tha hudain; jun table ma insert garira tyo table id; similar to scope identity, but scope identity ma recent PK jun xiryo tyo capture jun multiple value ma kam lagena so output inserted

SELECT

u.UserName,

u.[Password]

from #User as u

LEFT JOIN dbo.[User] AS u1 ON u1.UserName = u.UserName AND u1.Password =
u.Password --join lagako duplicate napathau vanera

WHERE u1.UserId IS NULL

ORDER BY u.Id; --id le order gara

SELECT @Json = (

SELECT

o.UserId AS userId, --user id preserve

```

JSON_QUERY(u.Hotel) AS hotel,
JSON_QUERY(u.Branch) AS branch,
    JSON_QUERY(u.Employee) AS employee,
    JSON_QUERY(u.Customer) AS customer
FROM #Out o

INNER JOIN #User AS u ON u.Id = o.Id

FOR JSON PATH

);

select @json as [user]

END;

```

- **HOTEL TABLE**

```

USE [Swastika_DB]

GO

/***** Object: StoredProcedure [dbo].[SpHotelIns]  Script Date: 6/3/2023 2:16:46 PM
*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE or ALTER PROCEDURE [dbo].[SpHotelIns]

@Json AS NVARCHAR(max) OUTPUT

AS

BEGIN

SET NOCOUNT ON;

```

```
create table #Hotel
```

```
(
```

```
Id int identity (1,1),
```

```
HotelName varchar(50),
```

```
UserId int,
```

```
Branch varchar(MAX),
```

```
Employee varchar(Max),
```

```
Customer varchar(MAX)
```

```
); insert into
```

```
#Hotel
```

```
(
```

```
HotelName,
```

```
UserId,
```

```
Branch,
```

```
Employee,
```

```
Customer
```

```
)
```

```
select oj.hotelName,
```

```
ojj.userId,
```

```
oj.branch,
```

```
oj.Employee,
```

```
oj.Customer
```

```
from openjson (@Json) with(
```

```
hotel nvarchar(max) as json,
```

```
userId int
```

```
) as ojj
```

```
cross apply openjson (ojj.hotel)
```

```
with(
```

```
    hotelName varchar(50),
```

```
    branch nvarchar(max) as Json,
```

```
    employee nvarchar(max) as Json,
```

```
    customer nvarchar(max) as Json
```

```
) as oj;
```

```
select @json as hotel
```

```
create table #out(
```

```
    id int
```

```
identity(1,1),
```

```
    HotelId int
```

```
);
```

```
insert into dbo.Hotel
```

```
(
```

```
    HotelName,
```

```
    UserId
```

```
)
```

```
output inserted.HotelId
```

```
into #out (HotelId)
```

```
select h.HotelName,  
       h.UserId
```

```
from #Hotel as h
```

```
left join dbo.Hotel as h2 on h2.HotelName = h.HotelName
```

```
where h2.HotelId is null
```

```
order by h.id;
```

```
select @json as hotel1
```

```
select @json = ( select
```

```
o.HotelId as hotelId,
```

```
   h.UserId as userId,
```

```
   JSON_QUERY(h.Branch) as branch,
```

```
   JSON_QUERY(h.Employee) as employee,
```

```
   JSON_QUERY(h.Customer) as customer
```

```
from #out o
```

```
inner join #Hotel as h on h.Id = o.Id
```

```
for json path
```

```
);
```

```
select @json as hotel2
```

```
end;
```

- **BRANCH TABLE**

```
use [Swastika_DB]
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[SpBranchIns]  Script Date: 6/4/2023 4:42:31 PM
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE OR ALTER PROCEDURE [dbo].[SpBranchIns]
```

```
    @Json AS NVARCHAR(MAX) OUTPUT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
CREATE TABLE #Branch
```

```
(
```

```
    Id INT IDENTITY (1,1),
```

```
    HotelId INT,
```

```
    UserId INT,
```

```
    BranchName VARCHAR(100),
```

```
    BranchLocation VARCHAR(100),
```

```
    Contact VARCHAR(10),
```

```
    Employee NVARCHAR(MAX),
```

```
    Customer NVARCHAR(MAX)
```

```
);
```



```
INSERT INTO #Branch
```

```
(  
    HotelId,  
    UserId,  
    BranchName,  
    BranchLocation,  
    Contact,  
    Employee,  
    Customer  
)
```

```
SELECT ojj.hotelId,  
ojj.userId,  
oj.branchName,  
oj.branchLocation,  
oj.contact,  
oj.Employee,  
oj.Customer
```

```
FROM OPENJSON(@Json)
```

```
    WITH (branch  
nvarchar(max) as json,  
    hotelId INT,      userId INT  
    ) AS ojj
```

```
CROSS APPLY OPENJSON(ojj.branch)
```

```
        WITH(                branchName
VARCHAR(50),
        branchLocation VARCHAR(50),

        contact VARCHAR(10),
        employee NVARCHAR(max) as json,
        customer NVARCHAR(max) as
json
        ) as oj;
```

```
CREATE TABLE #out
```

```
(
    Id INT IDENTITY(1,1),
    BranchId INT
);
```

```
INSERT INTO dbo.Branch
```

```
(
    HotelId,
    UserId,
    BranchName,
    BranchLocation,
    Contact
```

```
)
```

```
OUTPUT Inserted.BranchId
```

```
INTO #out (BranchId)
```

```
SELECT b.HotelId, b.UserId, b.BranchName, b.BranchLocation, b.Contact
FROM #Branch AS b
LEFT JOIN dbo.Branch AS b2 ON b2.BranchName = b.BranchName WHERE
b2.BranchId IS NULL
ORDER BY b.Id;
```

```
SELECT @Json = (
    SELECT o.BranchId as branchId,
        b.HotelId as hotelId,
        b.UserId as userId,
        JSON_QUERY(b.Employee) as employee,
        JSON_QUERY(b.Customer) as customer
    FROM #out o
    INNER JOIN #Branch AS b ON b.Id = o.Id
    FOR JSON PATH
);

--SELECT @Json;

END;
```

- **EMPLOYEE TABLE**

USE [Swastika_DB]

GO

/***** Object: StoredProcedure [dbo].[SpUserIns] Script Date: 6/4/2023 4:42:18 PM
*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE OR ALTER PROCEDURE [dbo].[SpEmployeeIns]

@Json AS NVARCHAR(MAX) OUTPUT

AS

BEGIN

SET NOCOUNT ON;

CREATE TABLE #Employee

(

Id INT IDENTITY (1,1),

BranchId INT,

UserId INT,

FirstName VARCHAR(50),

LastName VARCHAR(50),

EmployeePost VARCHAR(50),

EmployeeAddress VARCHAR(50),

EmployeeContact VARCHAR(50),

EmployeeSalary VARCHAR(50),

Customer NVARCHAR(MAX)

);

INSERT INTO #Employee (

BranchId,

UserId,

FirstName,

LastName,

EmployeePost,

EmployeeAddress,

EmployeeContact,

EmployeeSalary,

Customer

)

SELECT ojj.branchId, ojj.userId, ojj.FirstName, ojj.LastName, ojj.EmployeePost,
ojj.EmployeeAddress, ojj.EmployeeContact, ojj.EmployeeSalary, ojj.Customer

FROM OPENJSON (@Json)

WITH (

employee nvarchar(max) as json,

customer NVARCHAR(MAX) as

json, branchId INT, userId INT

) AS ojj

CROSS APPLY OPENJSON (ojj.employee)

WITH (

firstName VARCHAR(50),

lastName VARCHAR(50),

employeePost VARCHAR(max),

```
        employeeAddress
VARCHAR(max), employeeContact
VARCHAR(max), employeeSalary
VARCHAR(max)
```

```
) AS oj;
```

```
--CREATE TABLE #out
```

```
--(
--  Id INT IDENTITY(1,1),
--  BranchId INT
--);
```

```
SELECT 1
```

```
INSERT INTO dbo.Employee      (BranchId,  FirstName,  LastName,
    EmployeePost, EmployeeAddress, EmployeeContact, EmployeeSalary, UserId)
```

```
SELECT e.BranchId, e.FirstName, e.LastName, e.EmployeePost, e.EmployeeAddress,
e.EmployeeContact, e.EmployeeSalary, e.UserId
```

```
FROM #Employee AS e
```

```
LEFT JOIN dbo.Employee AS e2 ON e2.FirstName = e.FirstName
```

```
WHERE e2.UserId IS NULL
```

```
ORDER BY e.Id
```

```
SELECT @Json = (
```

```
    SELECT
```

```
        e.UserId as  userId,
```

```
        e.Branchid as branchId,
```

```

        JSON_QUERY(e.Customer) as customer

    from #Employee e

    FOR JSON PATH
    )

END;

```

- **CUSTOMER TABLE**

```

USE [Swastika_DB]

GO

/***** Object: StoredProcedure [dbo].[SpUserIns]  Script Date: 6/4/2023 4:42:18 PM
*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE OR ALTER PROCEDURE [dbo].[SpCustomerIns]

    @Json AS NVARCHAR(MAX) OUTPUT

AS

BEGIN

    SET NOCOUNT ON;

    CREATE TABLE #Customer

    (

```

Id INT IDENTITY (1,1),
BranchId INT,
UserId INT,
FirstName VARCHAR(50),
LastName VARCHAR(100),
PhoneNumber VARCHAR(10),
CustomerAddress VARCHAR(50),
Email VARCHAR(max),

);

INSERT INTO #Customer (BranchId, UserId, FirstName, LastName, PhoneNumber,
CustomerAddress, Email)

SELECT ojj.branchId, ojj.userId, oj.firstName, oj.lastName,
 oj.phoneNumber, oj.customerAddress, oj.email

FROM OPENJSON (@Json)

WITH (

 customer NVARCHAR(MAX) AS JSON,

 branchId INT,

 userId INT

) AS ojj

CROSS APPLY OPENJSON (ojj.customer)

WITH (

 firstName VARCHAR(50),

 lastName VARCHAR(50),

 phoneNumber VARCHAR(50),

 customerAddress VARCHAR(50),

 email VARCHAR(max),


```
        userPersonId INT
    ) AS oj;
```

```
INSERT INTO dbo.CUSTOMER
```

```
(
    FirstName,
    LastName,
    PhoneNumber,
    CustomerAddress,
    Email,
    UserId,
    BranchId)
```

```
SELECT c.FirstName,
c.LastName,

    c.PhoneNumber,
    c.CustomerAddress,
    c.Email,
    c.UserId,
    c.BranchId
```

```
FROM #Customer AS c
```

```
LEFT JOIN dbo.Customer AS c2 ON c2.CustomerAddress = c.CustomerAddress and
c2.FirstName=c.Firstname and c2.LastName=c.Lastname
```

```
WHERE c2.CustomerId IS NULL
```

```
ORDER BY c.Id;
```

END;

- **MAIN SP**

CREATE OR ALTER PROCEDURE [dbo].[SpHotelTsk]

@JSON AS VARCHAR (MAX) OUTPUT

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

BEGIN TRANSACTION

EXEC dbo.SpUserIns @Json = @Json OUTPUT, @UserId = NULL;

select @Json as 'Output From SpUserIns and Input for SpHotelIns'

EXEC dbo.SpHotelIns @Json = @Json OUTPUT;

select @Json as 'Output From SpHotelIns and Input for SpBranchIns'

EXEC dbo.SpBranchIns @Json = @Json OUTPUT;

select @Json as 'Output From SpBranchIns and Input for SpEmployeeIns'

EXEC dbo.SpEmployeeIns @Json = @Json OUTPUT;

select @Json as 'Output From SpEmployeeIns and Input for SpCustomerIns'

EXEC dbo.SpCustomerIns @Json = @Json OUTPUT;

select @Json as 'Output From SpCustomerIns'

select 1/0

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT>0

```
ROLLBACK TRANSACTION;
        THROW;
    END CATCH
END;
```

```
EXEC dbo.SpHotelTsk @Json = '
[
{
    "user": {
        "userName": "Priyaa",
        "password": "678546",
        "hotel": {
            "hotelName": "Annnapurna",
            "branch": {
                "branchName": "Coomfortable Hotel",
                "branchLocation": "Balkkhu",
                "contact": 123356789,
                "employee": [
                    {
                        "firstName": "Jaacky",
                        "lastName": "Joohn",
                        "employeePost": "Engineeer",
                        "employeeAddress": "2934 maiin street",
                        "employeeContact": 7685647031,
                        "employeeSalary": 41008
                    }
                ],
                "customer": [
                    {
                        "firstName": "Supriiya",
                        "lastName": "Bajrachaarya",
```

```
        "phoneNumber": 9087584321,  
        "customerAddress": "5687 jhoebe",  
        "email": "ss@emeil.com"  
    }  
]  
}  
}  
}  
}  
];
```

```
--SELECT json_value(@json, '$.data')
```