

```

/* Authors: Jessica McEwan C3393168, Lena Dahlin C3391146
Task: Assignment 1 Create Package stored procedure
Date Created: 18/03/2023 Last updated: 22/03/2023
*/
-- Drop procedures + type
DROP PROCEDURE IF EXISTS usp_createPackage
GO
DROP TYPE IF EXISTS ServiceList
GO
-- Create the service list table
CREATE TYPE ServiceList AS TABLE(
    serviceID VARCHAR(10),
    quantity INT
)
GO
--Create procedure that creates packages
CREATE PROCEDURE usp_createPackage @PackageName VARCHAR(30),
@ServiceItems ServiceList READONLY,
@Description VARCHAR(100),
@StartDate DATETIME,
@EndDate DATETIME,
@AdvPrice MONEY,
@AdvCurrency VARCHAR(20),
@Employee VARCHAR(30),
@NewPackageID CHAR(10) OUTPUT
AS
BEGIN
    BEGIN TRY
        BEGIN TRANSACTION
        --ERROR CHECKING
        -- Checks that Start date is before the end date
        IF @StartDate > @EndDate
            BEGIN
                RAISERROR ('Start date must be before the end date.', 11, 1) WITH NOWAIT;
            END
        -- Check that service quantities are at least 1
        IF EXISTS (SELECT quantity
            FROM @ServiceItems
            WHERE quantity < 1)
            BEGIN
                RAISERROR ('Service item quantity must be at least 1.', 12, 1) WITH NOWAIT;
            END
        -- Checks that ServiceID exists
        IF NOT EXISTS (SELECT *
            FROM @ServiceItems t
            INNER JOIN ServiceItem s
            ON t.serviceID = s.serviceID)
    
```

```

BEGIN
    RAISERROR ('Service item does not exist', 12, 1) WITH NOWAIT;
END
-- Checks that serviceID is not duplicate
IF EXISTS (SELECT serviceID
FROM @ServiceItems
GROUP BY serviceID
HAVING COUNT(*) > 1)
    BEGIN
        RAISERROR ('Cannot have duplicate service IDs for packages', 12, 1) WITH
NOWAIT;
    END
SET @NewPackageID = CONCAT('P', ABS(CHECKSUM(NEWID()))))
    WHILE EXISTS(SELECT * FROM Package WHERE PackageID = @NewPackageID)
        BEGIN
            SET @NewPackageID = CONCAT('P', ABS(CHECKSUM(NEWID()))))
        END
-- INSERT VALUES
INSERT INTO Package VALUES
(@NewPackageID, @PackageName, @Description, @StartDate, @EndDate,
@AdvPrice, @AdvCurrency, NULL, NULL, NULL, NULL, @Employee);
INSERT INTO PackageServiceItem (packageID, serviceID, quantity)
SELECT @NewPackageID, serviceID, quantity
FROM @ServiceItems
COMMIT TRANSACTION
END TRY
BEGIN CATCH
-- Show errors
SELECT ERROR_MESSAGE() AS ErrorMessage,
ERROR_SEVERITY() AS ErrorSeverity,
ERROR_STATE() AS ErrorState
    ROLLBACK TRANSACTION
END CATCH
END

```