# -- STORED PROCEDURES

## -- usp\_AddTenantToTenancy

CREATE OR ALTER PROCEDURE [dbo].[usp\_AddTenantToTenancy]

    @TenancyID INT,

    @TenantID INT

AS

BEGIN

    SET NOCOUNT ON;

    -- Check is tenant is already added to tenancy

    IF EXISTS (

        SELECT 1

        FROM TenancyTenant

        WHERE TenancyID = @TenancyID AND TenantID = @TenantID

    )

    BEGIN

        -- If so, end with errormessage.

        RAISERROR('Denne lejer er allerede tilføjet til lejemålet', 16, 1);

        RETURN;

    END

    -- if tenant is not already added to tenancy, add tenant

    INSERT INTO TenancyTenant (TenancyID, TenantID)

    VALUES (@TenancyID, @TenantID);

END;

GO

## -- usp\_CreateAddress

CREATE OR ALTER PROCEDURE usp\_CreateAddress

    @Street NVARCHAR(255),

    @Number NVARCHAR(50),

    @FloorNumber NVARCHAR(50),

    @Zipcode NVARCHAR(20),

    @Country NVARCHAR(100),

    @IsStandardized BIT

AS

BEGIN

    SET NOCOUNT ON;

    INSERT INTO Address (Street, Number, FloorNumber, Zipcode, Country, IsStandardized)

    VALUES (@Street, @Number, @FloorNumber, @Zipcode, @Country, @IsStandardized);

    SELECT SCOPE\_IDENTITY() AS NewAddressID;

END;

GO

## -- usp\_CreateTenant

CREATE OR ALTER PROCEDURE [dbo].[usp\_CreateTenant]

    @FirstName NVARCHAR(50),

    @LastName NVARCHAR(50),

    @PhoneNum NVARCHAR(20),

    @Email NVARCHAR(100)

AS

BEGIN

    SET NOCOUNT ON;

    DECLARE @PartyID INT;

    -- Insert into Party table and get the new PartyID

    INSERT INTO Party (PhoneNum, Email, PartyRole)

    VALUES (@PhoneNum, @Email, 'Tenant');

    SET @PartyID = SCOPE\_IDENTITY();

    -- Insert into Tenant table using the new PartyID

    INSERT INTO Tenant (FirstName, LastName, PartyID)

    VALUES (@FirstName, @LastName, @PartyID);

    SELECT SCOPE\_IDENTITY() AS TenantID;

END;

GO

## -- usp\_DeleteAddress

CREATE OR ALTER PROCEDURE usp\_DeleteAddress

    @AddressID INT

AS

BEGIN

    SET NOCOUNT ON;

    DELETE FROM Address

    WHERE AddressID = @AddressID;

END;

GO

## -- usp\_DeleteTenancy

CREATE OR ALTER PROCEDURE [dbo].[usp\_DeleteTenancy]

@TenancyID INT

AS

BEGIN

    BEGIN TRY

        BEGIN TRANSACTION;

        -- Delete related rows in TenancyTenant table first

        DELETE FROM TenancyTenant WHERE TenancyID = @TenancyID;

        -- Delete from Tenancy table

        DELETE FROM Tenancy WHERE TenancyID = @TenancyID;

        COMMIT TRANSACTION;

    END TRY

    BEGIN CATCH

        ROLLBACK TRANSACTION;

        THROW;

    END CATCH

END;

GO

## -- usp\_DeleteTenancyTenant

CREATE OR ALTER PROCEDURE [dbo].[usp\_DeleteTenancyTenant]

    @TenancyID INT,

    @TenantID INT

AS

BEGIN

    SET NOCOUNT ON;

    -- Delete Tenant from Tenancy

    DELETE FROM TenancyTenant

    WHERE TenancyID = @TenancyID AND TenantID = @TenantID;

END;

GO

## -- usp\_DeleteTenant

CREATE OR ALTER PROCEDURE [dbo].[usp\_DeleteTenant]

    @TenantID INT

AS

BEGIN

    SET NOCOUNT ON;

    BEGIN TRY

        BEGIN TRANSACTION;

        -- Delete related rows in TenancyTenant table first

        DELETE FROM TenancyTenant WHERE TenantID = @TenantID;

        -- Delete from Tenant table

        DELETE FROM Tenant WHERE TenantID = @TenantID;

        COMMIT TRANSACTION;

    END TRY

    BEGIN CATCH

        ROLLBACK TRANSACTION;

        THROW;

    END CATCH

END;

GO

## -- usp\_GetAddressByID

CREATE OR ALTER PROCEDURE usp\_GetAddressByID

@AddressID INT

AS

BEGIN

SET NOCOUNT ON;

-- Get address and city with JOIN in City-table

SELECT

a.AddressID,

a.Street,

a.Number,

a.FloorNumber,

a.Zipcode,

a.Country,

a.IsStandardized,

c.City

FROM Address a

JOIN City c

ON a.Zipcode = c.Zipcode -- JOIN på Zipcode

WHERE a.AddressID = @AddressID;

END;

GO

## -- usp\_GetTenancyByID

CREATE OR ALTER PROCEDURE [dbo].[usp\_GetTenancyByID]

@TenancyID INT

AS

BEGIN

SET NOCOUNT ON;

SELECT

t.TenancyID,

t.TenancyStatus,

t.MoveInDate,

t.MoveOutDate,

t.SquareMeter,

t.Rent,

t.Rooms,

t.Bathrooms,

t.PetsAllowed,

t.AddressID,

a.Street,

a.Number,

a.FloorNumber,

a.Zipcode,

a.Country,

a.IsStandardized,

cty.City AS AddressCity, -- Inkluder City fra City-tabellen

t.OrganizationID,

o.OrganizationName,

op.PhoneNum AS OrganizationPhoneNum,

op.Email AS OrganizationEmail,

t.CompanyID,

c.CompanyName,

cp.PhoneNum AS CompanyPhoneNum,

cp.Email AS CompanyEmail,

tp.PartyID,

tn.TenantID,

tn.FirstName AS TenantFirstName,

tn.LastName AS TenantLastName,

tp.PhoneNum AS TenantPhoneNum,

tp.Email AS TenantEmail,

tp.PartyRole,

t.IsDeleted

FROM

Tenancy t

INNER JOIN

Organization o ON t.OrganizationID = o.OrganizationID

INNER JOIN

Party op ON o.PartyID = op.PartyID

LEFT JOIN

Company c ON t.CompanyID = c.CompanyID

LEFT JOIN

Party cp ON c.PartyID = cp.PartyID

LEFT JOIN

Address a ON t.AddressID = a.AddressID

LEFT JOIN

City cty ON a.Zipcode = cty.Zipcode -- JOIN på City-tabellen for at få byoplysninger

LEFT JOIN

TenancyTenant tt ON t.TenancyID = tt.TenancyID

LEFT JOIN

Tenant tn ON tt.TenantID = tn.TenantID

LEFT JOIN

Party tp ON tn.PartyID = tp.PartyID

WHERE

t.TenancyID = @TenancyID

ORDER BY

t.TenancyID, tn.TenantID;

END;

GO

## -- usp\_GetTenantByID

CREATE OR ALTER PROCEDURE [dbo].[usp\_GetTenantByID]

    @TenantID INT

AS

BEGIN

    SET NOCOUNT ON;

    SELECT

        t.TenantID,

        p.PartyID,

        t.FirstName AS TenantFirstName,

        t.LastName AS TenantLastName,

        p.PhoneNum AS TenantPhoneNum,

        p.Email AS TenantEmail,

        p.PartyRole

    FROM

        Tenant t

    INNER JOIN

        Party p ON t.PartyID = p.PartyID

    WHERE

        t.TenantID = @TenantID;

END;

GO

## -- usp\_GetTenantsByTenancyId

CREATE OR ALTER PROCEDURE [dbo].[usp\_GetTenantsByTenancyId]

    @TenancyID INT

AS

BEGIN

    SET NOCOUNT ON;

    SELECT

        t.TenantID,

        p.PartyID,

        t.FirstName AS TenantFirstName,

        t.LastName AS TenantLastName,

        p.PhoneNum AS TenantPhoneNum,

        p.Email AS TenantEmail,

        p.PartyRole

    FROM

        TenancyTenant tt

    INNER JOIN

        Tenant t ON tt.TenantID = t.TenantID

    INNER JOIN

        Party p ON t.PartyID = p.PartyID

    WHERE

        tt.TenancyID = @TenancyID;

END;

GO

## -- usp\_GetUserByUsername

CREATE OR ALTER PROCEDURE [dbo].[usp\_GetUserByUsername]

    @Username NVARCHAR(50)

AS

BEGIN

    SET NOCOUNT ON;

    DECLARE @UserID INT;

    -- Get User ID

    SELECT @UserID = UserID

    FROM ApplicationUser

    WHERE Username = @Username;

    -- If user exist, update ActiveUser status

    IF @UserID IS NOT NULL

    BEGIN

        -- Set all users ActiveUser to FALSE

        UPDATE ApplicationUser

        SET ActiveUser = 0;

        -- Set the chosen user to ActiveUser = TRUE

        UPDATE ApplicationUser

        SET ActiveUser = 1

        WHERE UserID = @UserID;

    END

    SELECT

        UserID,

        Username,

        CONVERT(VARCHAR(MAX), UserPasswordHash, 2) AS UserPasswordHash,

        ActiveUser

    FROM ApplicationUser

    WHERE Username = @Username;

END;

GO

## -- usp\_ReadAllAddresses

CREATE OR ALTER PROCEDURE usp\_ReadAllAddresses

AS

BEGIN

SET NOCOUNT ON;

SELECT

a.AddressID,

a.Street,

a.Number,

a.FloorNumber,

a.Zipcode,

a.Country,

a.IsStandardized,

c.City AS AddressCity -- Inkluder City fra City-tabellen

FROM

Address a

LEFT JOIN

City c ON a.Zipcode = c.Zipcode; -- JOIN på Zipcode

END;

GO

## --usp\_ReadAllTenancies

CREATE OR ALTER PROCEDURE [dbo].[usp\_ReadAllTenancies]

AS

BEGIN

SET NOCOUNT ON;

SELECT

t.TenancyID,

t.TenancyStatus,

t.MoveInDate,

t.MoveOutDate,

t.SquareMeter,

t.Rent,

t.Rooms,

t.Bathrooms,

t.PetsAllowed,

t.AddressID,

a.Street,

a.Number,

a.FloorNumber,

a.Zipcode,

a.Country,

a.IsStandardized,

cty.City AS AddressCity,

t.OrganizationID,

o.OrganizationName,

op.PhoneNum AS OrganizationPhoneNum,

op.Email AS OrganizationEmail,

t.CompanyID,

c.CompanyName,

cp.PhoneNum AS CompanyPhoneNum,

cp.Email AS CompanyEmail,

tp.PartyID,

tn.TenantID,

tn.FirstName AS TenantFirstName,

tn.LastName AS TenantLastName,

tp.PhoneNum AS TenantPhoneNum,

tp.Email AS TenantEmail,

tp.PartyRole,

t.IsDeleted

FROM

Tenancy t

INNER JOIN

Organization o ON t.OrganizationID = o.OrganizationID

INNER JOIN

Party op ON o.PartyID = op.PartyID

LEFT JOIN

Company c ON t.CompanyID = c.CompanyID

LEFT JOIN

Party cp ON c.PartyID = cp.PartyID

LEFT JOIN

Address a ON t.AddressID = a.AddressID

LEFT JOIN

City cty ON a.Zipcode = cty.Zipcode -- Join with City table on Zipcode

LEFT JOIN

TenancyTenant tt ON t.TenancyID = tt.TenancyID

LEFT JOIN

Tenant tn ON tt.TenantID = tn.TenantID

LEFT JOIN

Party tp ON tn.PartyID = tp.PartyID

ORDER BY

t.TenancyID, tn.TenantID;

END;

GO

## -- usp\_ReadAllTenants

CREATE OR ALTER PROCEDURE [dbo].[usp\_ReadAllTenants]

AS

BEGIN

    SET NOCOUNT ON;

    SELECT

        t.TenantID,

        p.PartyID,

        t.FirstName AS TenantFirstName,

        t.LastName AS TenantLastName,

        p.PhoneNum AS TenantPhoneNum,

        p.Email AS TenantEmail,

        p.PartyRole

    FROM

        Tenant t

    INNER JOIN

        Party p ON t.PartyID = p.PartyID;

END;

GO

## -- usp\_UpdateAddress

CREATE OR ALTER PROCEDURE usp\_UpdateAddress

    @AddressID INT,

    @Street NVARCHAR(255),

    @Number NVARCHAR(50),

    @FloorNumber NVARCHAR(50),

    @Zipcode NVARCHAR(20),

    @Country NVARCHAR(100),

    @IsStandardized BIT

AS

BEGIN

    SET NOCOUNT ON;

    UPDATE Address

    SET

        Street = @Street,

        Number = @Number,

        FloorNumber = @FloorNumber,

        Zipcode = @Zipcode,

        Country = @Country,

        IsStandardized = @IsStandardized

    WHERE AddressID = @AddressID;

END;

GO

## -- usp\_UpdateTenancy

CREATE OR ALTER   PROCEDURE [dbo].[usp\_UpdateTenancy]

    @TenancyID INT,

    @TenancyStatus NVARCHAR(50) = NULL,

    @MoveInDate DATETIME = NULL,

    @MoveOutDate DATETIME = NULL,

    @SquareMeter INT = NULL,

    @Rent DECIMAL = NULL,

    @Rooms INT = NULL,

    @Bathrooms INT = NULL,

    @PetsAllowed BIT = NULL,

    @CompanyID INT = NULL,

    @IsDeleted BIT = NULL,

@AddressID INT = NULL,

@OrganizationID INT = NULL

AS

BEGIN

    UPDATE Tenancy

    SET

        TenancyStatus = COALESCE(@TenancyStatus, TenancyStatus),

        MoveInDate = COALESCE(@MoveInDate, MoveInDate),

        MoveOutDate = COALESCE(@MoveOutDate, MoveOutDate),

        SquareMeter = COALESCE(@SquareMeter, SquareMeter),

        Rent = COALESCE(@Rent, Rent),

        Rooms = COALESCE(@Rooms, Rooms),

        Bathrooms = COALESCE(@Bathrooms, Bathrooms),

        PetsAllowed = COALESCE(@PetsAllowed, PetsAllowed),

        CompanyID = COALESCE(@CompanyID, CompanyID),

        IsDeleted = COALESCE(@IsDeleted, IsDeleted),

AddressID = COALESCE(@AddressID, AddressID),

OrganizationID = COALESCE(@OrganizationID, OrganizationID)

    WHERE TenancyID = @TenancyID;

END;

GO

## -- usp\_UpdateTenant

CREATE OR ALTER PROCEDURE [dbo].[usp\_UpdateTenant]

    @TenantID INT,

    @FirstName NVARCHAR(50),

    @LastName NVARCHAR(50),

    @PhoneNum NVARCHAR(20),

    @Email NVARCHAR(100)

AS

BEGIN

    SET NOCOUNT ON;

    DECLARE @PartyID INT;

    -- Get the PartyID associated with the TenantID

    SELECT @PartyID = PartyID FROM Tenant WHERE TenantID = @TenantID;

    -- Update Party table with new PhoneNum and Email

    UPDATE Party

    SET PhoneNum = @PhoneNum,

        Email = @Email

    WHERE PartyID = @PartyID;

    -- Update Tenant table with new FirstName and LastName

    UPDATE Tenant

    SET FirstName = @FirstName,

        LastName = @LastName

    WHERE TenantID = @TenantID;

END;

GO

## -- usp\_ReadAllHistories

CREATE OR ALTER PROCEDURE [dbo].[usp\_ReadAllHistories]

    @TenancyID INT = NULL

AS

BEGIN

    SET NOCOUNT ON;

    SELECT

        h.ChangeID,

        h.ChangeDate,

        h.ChangeOperation,

        h.FieldChanged,

        h.OldValue,

        h.NewValue,

        h.TenancyID,

        h.UserID,

        h.OrganizationID,

        a.AddressID,

        CONCAT(a.Street, ' ', a.Number, ', ', ISNULL(a.FloorNumber + ', ', ''), a.Zipcode, ', ', a.Country) AS FullAddress,

        CASE

            WHEN h.FieldChanged = 'Tenant' THEN

                CASE

                    WHEN h.OldValue IS NOT NULL THEN tOld.FirstName + ' ' + tOld.LastName

                    WHEN h.NewValue IS NOT NULL THEN tNew.FirstName + ' ' + tNew.LastName

                    ELSE NULL

                END

            ELSE NULL

        END AS TenantName

    FROM History h

    LEFT JOIN Tenancy tn ON h.TenancyID = tn.TenancyID

    LEFT JOIN Address a ON tn.AddressID = a.AddressID

    LEFT JOIN Tenant tOld ON h.FieldChanged = 'Tenant' AND TRY\_CAST(h.OldValue AS INT) = tOld.TenantID

    LEFT JOIN Tenant tNew ON h.FieldChanged = 'Tenant' AND TRY\_CAST(h.NewValue AS INT) = tNew.TenantID

    WHERE (@TenancyID IS NULL OR h.TenancyID = @TenancyID)

    ORDER BY h.ChangeDate DESC;

END

GO

## -- usp\_ReadAllAddresses

CREATE OR ALTER PROCEDURE usp\_ReadAllAddresses

AS

BEGIN

SET NOCOUNT ON;

SELECT

a.AddressID,

a.Street,

a.Number,

a.FloorNumber,

a.Zipcode,

a.Country,

a.IsStandardized,

c.City AS AddressCity

FROM

Address a

LEFT JOIN

City c ON a.Zipcode = c.Zipcode; -- JOIN på Zipcode

END;

GO

## -- usp\_CreateTenancy

CREATE OR ALTER PROCEDURE [dbo].[usp\_CreateTenancy]

    @TenancyStatus NVARCHAR(50) = NULL,

    @MoveInDate DATETIME = NULL,

    @MoveOutDate DATETIME = NULL,

    @SquareMeter INT = NULL,

    @Rent DECIMAL(18, 2) = NULL,

    @Rooms INT = NULL,

    @Bathrooms INT = NULL,

    @PetsAllowed BIT = NULL,

    @AddressID INT = NULL,

    @CompanyID INT = NULL,

    @OrganizationID INT = NULL,

    @IsDeleted BIT = 0

AS

BEGIN

    SET NOCOUNT ON;

    -- Insert the tenancy into the Tenancies table

    INSERT INTO Tenancy (

        TenancyStatus,

        MoveInDate,

        MoveOutDate,

        SquareMeter,

        Rent,

        Rooms,

        Bathrooms,

        PetsAllowed,

        AddressID,

        CompanyID,

        OrganizationID,

        IsDeleted

    )

    VALUES (

        @TenancyStatus,

        @MoveInDate,

        @MoveOutDate,

        @SquareMeter,

        @Rent,

        @Rooms,

        @Bathrooms,

        @PetsAllowed,

        @AddressID,

        @CompanyID,

        @OrganizationID,

        @IsDeleted

    );

END;

GO