

Bài 9:

Kiểm tra tài khoản trùng khi thêm nhân viên sử dụng đối tượng *SqlDataReader*

Viết phương thức trên lớp BLL_NhanVien

```
/// <summary>
/// Thực hiện lấy dữ liệu tồn tại của tài khoản nhân viên
/// </summary>
/// <param name="err">Biến lưu lỗi</param>
/// <param name="taiKhoan">tài khoản cần kiểm tra</param>
/// <returns></returns>
public SqlDataReader KiemTraTrungTaiKhoanNhanVien(ref string err,string taiKhoan)
{
    return data.MyExcuteReader("Tên thủ
tục", CommandType.StoredProcedure, ref err,
    new SqlParameter("@TaiKhoan", taiKhoan));
}
```

Thủ tục SQL

```
CREATE PROC SP_NhanVien_KiemTraTrungTaiKhoan
@TaiKhoan VARCHAR(20)
AS
IF EXISTS (SELECT 1 FROM dbo.NhanVien WHERE TaiKhoan=@TaiKhoan)
BEGIN
    SELECT 1 AS result
END
ELSE
BEGIN
    SELECT 0 AS result
END
```

Phương thức kiểm tra

```
private bool KiemTraTaiKhoanTrung(string TaiKhoan)
{
    SqlDataReader _dataReader=bd.KiemTraTrungTaiKhoanNhanVien(ref err,TaiKhoan);
    int result = 0;
    while (_dataReader.Read())
    {
        result = _dataReader.GetInt32(0);
    }
    if(result==1)
    {
        return true;
    }
    return false;
}
```

Sự kiện gói phương thức

```
private void txtTaiKhoan_Leave(object sender, EventArgs e)
```

```
{
    if(!string.IsNullOrEmpty(txtTaiKhoan.Text))
    {
        if(KiemTraTaiKhoanTrung(txtTaiKhoan.Text))
        {
            MessageBox.Show("Tài khoản đã có \nXin vui lòng dùng tài khoản  
khác cho nhân viên " + txtHoTen.Text);
            txtTaiKhoan.ResetText();
            txtTaiKhoan.Focus();
        }
    }
}
```