***// each of these import declarations should be commented –***

***// - what are we importing from each module?***

***// what do each of the imported functions do, in one phrase?***

***//***

using System;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Migrations;

***//***

***// each of the max length constants in what follows should be defined as a constant***

***//***

namespace CPH\_IVT.Data.Migrations

{

public partial class CreateIdentitySchema : Migration

{

protected override void Up(MigrationBuilder migrationBuilder)

{

***// how does Name differ from NormalizedName?***

***// what is a ConcurrencyStamp?***

***//***

migrationBuilder.CreateTable(

name: "AspNetRoles",

columns: table => new

{

Id = table.Column<string>(nullable: false),

Name = table.Column<string>(maxLength: 256, nullable: true),

NormalizedName = table.Column<string>(maxLength: 256, nullable: true),

ConcurrencyStamp = table.Column<string>(nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetRoles", x => x.Id);

});

***// how does Email differ from NormalizedEmail?***

***// what is a SecurityStamp?***

***// what is a ConcurrencyStamp?***

***//***

migrationBuilder.CreateTable(

name: "AspNetUsers",

columns: table => new

{

Id = table.Column<string>(nullable: false),

UserName = table.Column<string>(maxLength: 256, nullable: true),

NormalizedUserName = table.Column<string>(maxLength: 256, nullable: true),

Email = table.Column<string>(maxLength: 256, nullable: true),

NormalizedEmail = table.Column<string>(maxLength: 256, nullable: true),

EmailConfirmed = table.Column<bool>(nullable: false),

PasswordHash = table.Column<string>(nullable: true),

SecurityStamp = table.Column<string>(nullable: true),

ConcurrencyStamp = table.Column<string>(nullable: true),

PhoneNumber = table.Column<string>(nullable: true),

PhoneNumberConfirmed = table.Column<bool>(nullable: false),

TwoFactorEnabled = table.Column<bool>(nullable: false),

LockoutEnd = table.Column<DateTimeOffset>(nullable: true),

LockoutEnabled = table.Column<bool>(nullable: false),

AccessFailedCount = table.Column<int>(nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUsers", x => x.Id);

});

***// what is a Role Claim?***

***// what is a ClaimType? a ClaimValue?***

***//***

migrationBuilder.CreateTable(

name: "AspNetRoleClaims",

columns: table => new

{

Id = table.Column<int>(nullable: false)

.Annotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn),

RoleId = table.Column<string>(nullable: false),

ClaimType = table.Column<string>(nullable: true),

ClaimValue = table.Column<string>(nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetRoleClaims", x => x.Id);

table.ForeignKey(

name: "FK\_AspNetRoleClaims\_AspNetRoles\_RoleId",

column: x => x.RoleId,

principalTable: "AspNetRoles",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

***// what is a User Claim?***

***// what is a ClaimType? a ClaimValue?***

***//***

migrationBuilder.CreateTable(

name: "AspNetUserClaims",

columns: table => new

{

Id = table.Column<int>(nullable: false)

.Annotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn),

UserId = table.Column<string>(nullable: false),

ClaimType = table.Column<string>(nullable: true),

ClaimValue = table.Column<string>(nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserClaims", x => x.Id);

table.ForeignKey(

name: "FK\_AspNetUserClaims\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

***// why do we need to track LoginProviders?***

***// what do we use to generate ProviderKeys?***

***//***

migrationBuilder.CreateTable(

name: "AspNetUserLogins",

columns: table => new

{

LoginProvider = table.Column<string>(maxLength: 128, nullable: false),

ProviderKey = table.Column<string>(maxLength: 128, nullable: false),

ProviderDisplayName = table.Column<string>(nullable: true),

UserId = table.Column<string>(nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserLogins", x => new { x.LoginProvider, x.ProviderKey });

table.ForeignKey(

name: "FK\_AspNetUserLogins\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

name: "AspNetUserRoles",

columns: table => new

{

UserId = table.Column<string>(nullable: false),

RoleId = table.Column<string>(nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserRoles", x => new { x.UserId, x.RoleId });

table.ForeignKey(

name: "FK\_AspNetUserRoles\_AspNetRoles\_RoleId",

column: x => x.RoleId,

principalTable: "AspNetRoles",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

table.ForeignKey(

name: "FK\_AspNetUserRoles\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

***// why do we need User Tokens? Are these cookies for managing web logins?***

***// of what significance are LoginProvider, Name, and Value?***

***//***

migrationBuilder.CreateTable(

name: "AspNetUserTokens",

columns: table => new

{

UserId = table.Column<string>(nullable: false),

LoginProvider = table.Column<string>(maxLength: 128, nullable: false),

Name = table.Column<string>(maxLength: 128, nullable: false),

Value = table.Column<string>(nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserTokens", x => new { x.UserId, x.LoginProvider, x.Name });

table.ForeignKey(

name: "FK\_AspNetUserTokens\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateIndex(

name: "IX\_AspNetRoleClaims\_RoleId",

table: "AspNetRoleClaims",

column: "RoleId");

migrationBuilder.CreateIndex(

name: "RoleNameIndex",

table: "AspNetRoles",

column: "NormalizedName",

unique: true,

filter: "[NormalizedName] IS NOT NULL");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserClaims\_UserId",

table: "AspNetUserClaims",

column: "UserId");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserLogins\_UserId",

table: "AspNetUserLogins",

column: "UserId");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserRoles\_RoleId",

table: "AspNetUserRoles",

column: "RoleId");

migrationBuilder.CreateIndex(

name: "EmailIndex",

table: "AspNetUsers",

column: "NormalizedEmail");

migrationBuilder.CreateIndex(

name: "UserNameIndex",

table: "AspNetUsers",

column: "NormalizedUserName",

unique: true,

filter: "[NormalizedUserName] IS NOT NULL");

}

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(name: "AspNetRoleClaims");

migrationBuilder.DropTable(name: "AspNetUserClaims");

migrationBuilder.DropTable(name: "AspNetUserLogins");

migrationBuilder.DropTable(name: "AspNetUserRoles");

migrationBuilder.DropTable(name: "AspNetUserTokens");

migrationBuilder.DropTable(name: "AspNetRoles");

migrationBuilder.DropTable(name: "AspNetUsers");

}

}

}