```csharp
using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;

namespace Meditech.App.Crm.Models;

public partial class DatabaseContext : DbContext
{
    public DatabaseContext()
    {
    }

    public DatabaseContext(DbContextOptions<DatabaseContext> options)
        : base(options)
    {
    }

    public virtual DbSet<Ambulance> Ambulances { get; set; }

    public virtual DbSet<Appointment> Appointments { get; set; }

    public virtual DbSet<BannerSetting> BannerSettings { get; set; }

    public virtual DbSet<Blog> Blogs { get; set; }

    public virtual DbSet<CareerCategory> CareerCategories { get; set; }

    public virtual DbSet<CheckTheBhytcount> CheckTheBhytcounts { get; set; }

    public virtual DbSet<Comment> Comments { get; set; }

    public virtual DbSet<Customer> Customers { get; set; }

    public virtual DbSet<DanhGiaCldv> DanhGiaCldvs { get; set; }

    public virtual DbSet<DanhSachCongTy> DanhSachCongTies { get; set; }

    public virtual DbSet<DichVuTaiNha> DichVuTaiNhas { get; set; }

    public virtual DbSet<DichVuTaiNhaCategory> DichVuTaiNhaCategories { get; set; }

    public virtual DbSet<DistrictCategory> DistrictCategories { get; set; }

    public virtual DbSet<DmDichVuKhamDoan> DmDichVuKhamDoans { get; set; }

    public virtual DbSet<DoanKhamSucKhoe> DoanKhamSucKhoes { get; set; }

    public virtual DbSet<DoctorManagement> DoctorManagements { get; set; }

    public virtual DbSet<DonHangKhamSucKhoe> DonHangKhamSucKhoes { get; set; }

    public virtual DbSet<DonHangKhamSucKhoeKhac> DonHangKhamSucKhoeKhacs { get; set; }

    public virtual DbSet<Esign> Esigns { get; set; }

    public virtual DbSet<Favorite> Favorites { get; set; }

    public virtual DbSet<GenderCategory> GenderCategories { get; set; }

    public virtual DbSet<Hospital> Hospitals { get; set; }

    public virtual DbSet<HospitalManagement> HospitalManagements { get; set; }

    public virtual DbSet<LikeComment> LikeComments { get; set; }

    public virtual DbSet<NationCategory> NationCategories { get; set; }

    public virtual DbSet<NationalityCategory> NationalityCategories { get; set; }

    public virtual DbSet<Notification> Notifications { get; set; }

    public virtual DbSet<NotificationReceiver> NotificationReceivers { get; set; }

    public virtual DbSet<Partner> Partners { get; set; }

    public virtual DbSet<PatientCompleteOrder> PatientCompleteOrders { get; set; }

    public virtual DbSet<PhanTramKinhDoanh> PhanTramKinhDoanhs { get; set; }

    public virtual DbSet<PhuongAnKinhDoanhKsk> PhuongAnKinhDoanhKsks { get; set; }

    public virtual DbSet<PhuongAnKinhDoanhKskChiTiet> PhuongAnKinhDoanhKskChiTiets { get; set; }

    public virtual DbSet<ProvinceCategory> ProvinceCategories { get; set; }

    public virtual DbSet<RatingRecommendContent> RatingRecommendContents { get; set; }

    public virtual DbSet<RelationshipCategory> RelationshipCategories { get; set; }

    public virtual DbSet<Setting> Settings { get; set; }

    public virtual DbSet<Speciality> Specialities { get; set; }

    public virtual DbSet<ThanhLyHopDongKsk> ThanhLyHopDongKsks { get; set; }

    public virtual DbSet<ThanhLyHopDongKskChiTiet> ThanhLyHopDongKskChiTiets { get; set; }

    public virtual DbSet<ThanhLyHopDongKskChiTietKhac> ThanhLyHopDongKskChiTietKhacs { get; set; }

    public virtual DbSet<ThietLapDanhGiaCldv> ThietLapDanhGiaCldvs { get; set; }

    public virtual DbSet<ThietLapKskDoan> ThietLapKskDoans { get; set; }

    public virtual DbSet<ThuTien> ThuTiens { get; set; }

    public virtual DbSet<TimeSlotCategory> TimeSlotCategories { get; set; }

    public virtual DbSet<WardCategory> WardCategories { get; set; }
    public virtual DbSet<ZaloOaInfo> ZaloOaInfos { get; set; }

    public virtual DbSet<ZaloOaTemplate> ZaloOaTemplates { get; set; }

    public virtual DbSet<ZnsHistory> ZnsHistories { get; set; }

//    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
//#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection str
//        => optionsBuilder.UseNpgsql("Host=dev.meditech.vn;Port=5433;Database=crm;Username=postgres;Password=Meditech2023!");

    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.HasPostgresExtension("uuid-ossp");

        modelBuilder.Entity<Ambulance>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Ambulance_pkey");

            entity.ToTable("Ambulance");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.FullName).HasColumnType("character varying");
            entity.Property(e => e.Latitude).HasColumnType("character varying");
            entity.Property(e => e.Location).HasColumnType("character varying");
            entity.Property(e => e.Longitude).HasColumnType("character varying");
            entity.Property(e => e.PhoneNumber).HasColumnType("character varying");
        });

        modelBuilder.Entity<Appointment>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Appointment_pkey");

            entity.ToTable("Appointment");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.BookingDate).HasColumnType("timestamp without time zone");
            entity.Property(e => e.BookingTime).HasColumnType("character varying");
            entity.Property(e => e.CancelReason).HasColumnType("character varying");
            entity.Property(e => e.EndTime).HasColumnType("timestamp without time zone");
            entity.Property(e => e.IsBhyt).HasColumnName("IsBHYT");
            entity.Property(e => e.MakcbHis)
                .HasMaxLength(50)
                .HasColumnName("makcbHIS");
            entity.Property(e => e.Note).HasColumnType("character varying");
            entity.Property(e => e.StartTime).HasColumnType("timestamp without time zone");
            entity.Property(e => e.Type).HasDefaultValueSql("1");
        });

        modelBuilder.Entity<BannerSetting>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Banner_pkey");

            entity.ToTable("BannerSetting");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.BannerUrl).IsRequired();
            entity.Property(e => e.Name).HasColumnType("character varying");
        });

        modelBuilder.Entity<Blog>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Posts_pkey");

            entity.ToTable("Blog");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.Name).HasColumnType("character varying");
            entity.Property(e => e.ShortDescription).HasColumnType("character varying");
        });

        modelBuilder.Entity<CareerCategory>(entity =>
        {
            entity.HasKey(e => e.CareerId).HasName("CareerCategory_pkey");

            entity.ToTable("CareerCategory");

            entity.Property(e => e.CareerId).HasMaxLength(3);
            entity.Property(e => e.CareerName)
                .IsRequired()
                .HasColumnType("character varying");
        });

        modelBuilder.Entity<CheckTheBhytcount>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("CheckTheBHYTCount_pkey");

            entity.ToTable("CheckTheBHYTCount");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.CreateTime).HasColumnType("timestamp without time zone");
        });

        modelBuilder.Entity<Comment>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Comments_pkey");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.Content).HasColumnType("character varying");
            entity.Property(e => e.ParentId).HasColumnName("parentId");
        });

        modelBuilder.Entity<Customer>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("Customer_pkey");

            entity.ToTable("Customer");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.ChuHoIds)
                .IsRequired()
                .HasColumnType("character varying");
            entity.Property(e => e.Email).HasColumnType("character varying");
            entity.Property(e => e.Hoten).HasColumnType("character varying");
            entity.Property(e => e.MaDanToc).HasColumnType("character varying");
            entity.Property(e => e.MaMqh).HasColumnName("MaMQH");
            entity.Property(e => e.MaNgheNghiep).HasColumnType("character varying");
            entity.Property(e => e.MaPx)
                .HasColumnType("character varying")
                .HasColumnName("MaPX");
            entity.Property(e => e.MaQh)
                .HasColumnType("character varying")
                .HasColumnName("MaQH");
            entity.Property(e => e.MaTt)
                .HasColumnType("character varying")
                .HasColumnName("MaTT");
            entity.Property(e => e.SoBhyt)
                .HasColumnType("character varying")
                .HasColumnName("SoBHYT");
            entity.Property(e => e.SoCccd)
                .HasColumnType("character varying")
                .HasColumnName("SoCCCD");
            entity.Property(e => e.SoDienThoai).HasColumnType("character varying");
            entity.Property(e => e.SoNha).HasColumnType("character varying");
        });

        modelBuilder.Entity<DanhGiaCldv>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("DanhGiaCLDV_pkey");

            entity.ToTable("DanhGiaCLDV");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.GhiChu).HasColumnType("character varying");
            entity.Property(e => e.HoTen).HasColumnType("character varying");
            entity.Property(e => e.SoDienThoai).HasColumnType("character varying");
        });

        modelBuilder.Entity<DanhSachCongTy>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("DanhSachCongTy_pkey");

            entity.ToTable("DanhSachCongTy");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.DiaChi)
                .IsRequired()
                .HasColumnType("character varying");
            entity.Property(e => e.DienThoai).HasColumnType("character varying");
            entity.Property(e => e.Email).HasColumnType("character varying");
            entity.Property(e => e.NguoiLienHe).HasMaxLength(250);
            entity.Property(e => e.TenCongTy)
                .IsRequired()
                .HasMaxLength(500);
        });

        modelBuilder.Entity<DichVuTaiNha>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("DichVuTaiNha_pkey");

            entity.ToTable("DichVuTaiNha");

            entity.HasIndex(e => e.Code, "unique_equip_id").IsUnique();

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.BookingDate).HasColumnType("timestamp without time zone");
            entity.Property(e => e.BookingTime).HasColumnType("timestamp without time zone");
            entity.Property(e => e.CancelReason).HasColumnType("character varying");
            entity.Property(e => e.Code)
                .IsRequired()
                .HasMaxLength(500);
            entity.Property(e => e.CustomerName).HasMaxLength(100);
            entity.Property(e => e.GenderName).HasColumnType("character varying");
            entity.Property(e => e.Latitude).HasColumnType("character varying");
            entity.Property(e => e.Longitude).HasColumnType("character varying");
            entity.Property(e => e.MakcbHis)
                .HasMaxLength(50)
                .HasColumnName("makcbHIS");
            entity.Property(e => e.Note).HasColumnType("character varying");
            entity.Property(e => e.PhoneNumber).HasMaxLength(100);
        });

        modelBuilder.Entity<DichVuTaiNhaCategory>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("DichVuTaiNhaCategory_pkey");

            entity.ToTable("DichVuTaiNhaCategory");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.IconUrl)
                .IsRequired()
                .HasMaxLength(500);
            entity.Property(e => e.KichHoat)
                .IsRequired()
                .HasDefaultValueSql("true");
            entity.Property(e => e.TenDichVu)
                .IsRequired()
                .HasMaxLength(500);
        });

        modelBuilder.Entity<DistrictCategory>(entity =>
        {
            entity.HasKey(e => e.DistrictId).HasName("DistrictCategory_pkey");

            entity.ToTable("DistrictCategory", tb => tb.HasComment("Danh mục quận huyện"));

            entity.Property(e => e.DistrictId).HasMaxLength(4);
            entity.Property(e => e.DistrictName)
                .IsRequired()
                .HasColumnType("character varying");
            entity.Property(e => e.ProvinceId)
                .IsRequired()
                .HasMaxLength(3);
        });
```

```
modelBuilder.Entity<DmDichVuKhamDoan>(entity =>
{
    entity.HasKey(e => e.Id).HasName("DmDichVuKhamDoan_pkey");

    entity.ToTable("DmDichVuKhamDoan");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.TenDichVu)
        .IsRequired()
        .HasMaxLength(500);
});

modelBuilder.Entity<DoanKhamSucKhoe>(entity =>
{
    entity.HasKey(e => e.Id).HasName("DoanKhamSucKhoe_pkey");

    entity.ToTable("DoanKhamSucKhoe");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.DiaChi).HasColumnType("character varying");
    entity.Property(e => e.DienThoai).HasColumnType("character varying");
    entity.Property(e => e.GhiChu).HasColumnType("character varying");
    entity.Property(e => e.MaDoanKham).HasColumnType("character varying");
    entity.Property(e => e.NguoiLienHe)
        .IsRequired()
        .HasColumnType("character varying");
    entity.Property(e => e.NvKinhDoanhId).HasColumnType("character varying");
    entity.Property(e => e.TenDoan)
        .IsRequired()
        .HasColumnType("character varying");
    entity.Property(e => e.VietTat).HasColumnType("character varying");
});

modelBuilder.Entity<DoctorManagement>(entity =>
{
    entity.HasKey(e => e.Id).HasName("DoctorManagement_pkey");

    entity.ToTable("DoctorManagement");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Address).HasColumnType("character varying");
    entity.Property(e => e.AliasName).HasMaxLength(50);
    entity.Property(e => e.Biography1).HasColumnType("character varying");
    entity.Property(e => e.Name).HasColumnType("character varying");
    entity.Property(e => e.NameUnsign).HasColumnType("character varying");
    entity.Property(e => e.PhoneNumber).HasColumnType("character varying");
});

modelBuilder.Entity<DonHangKhamSucKhoe>(entity =>
{
    entity.HasKey(e => e.Id).HasName("DonHangKhamSucKhoe_pkey");

    entity.ToTable("DonHangKhamSucKhoe");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.MaDonHang)
        .IsRequired()
        .HasMaxLength(15);
    entity.Property(e => e.TenDonHang)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<DonHangKhamSucKhoeKhac>(entity =>
{
    entity.HasKey(e => e.Id).HasName("DonHangKhamSucKhoeKhac_pkey");

    entity.ToTable("DonHangKhamSucKhoeKhac");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.ChuyenMon).HasColumnType("character varying");
    entity.Property(e => e.TenVatTu).HasColumnType("character varying");
});

modelBuilder.Entity<Esign>(entity =>
{
    entity.HasKey(e => e.IdentityNo).HasName("Esign_pkey");

    entity.ToTable("Esign");

    entity.Property(e => e.IdentityNo).HasMaxLength(255);
    entity.Property(e => e.Dob).HasColumnType("timestamp without time zone");
    entity.Property(e => e.Name)
        .IsRequired()
        .HasMaxLength(255);
});

modelBuilder.Entity<Favorite>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Favorite_pkey");

    entity.ToTable("Favorite");

    entity.Property(e => e.Id).ValueGeneratedNever();
});

modelBuilder.Entity<GenderCategory>(entity =>
{
    entity.HasKey(e => e.GenderId).HasName("GenderCategory_pkey");

    entity.ToTable("GenderCategory", tb => tb.HasComment("Danh mục giới tính"));

    entity.Property(e => e.GenderId).ValueGeneratedNever();
    entity.Property(e => e.GenderName)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<Hospital>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Hospital_pkey");
    entity.ToTable("Hospital");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Address).HasMaxLength(500);
    entity.Property(e => e.Code)
        .IsRequired()
        .HasMaxLength(200);
    entity.Property(e => e.Name).HasMaxLength(500);
    entity.Property(e => e.Type).HasMaxLength(200);
    entity.Property(e => e.ZnsRatingTemplateId).HasMaxLength(200);
});

modelBuilder.Entity<HospitalManagement>(entity =>
{
    entity.HasKey(e => e.Id).HasName("HospitalMenegement_pkey");

    entity.ToTable("HospitalManagement");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Address).HasColumnType("character varying");
    entity.Property(e => e.Biography1)
        .HasColumnType("character varying")
        .HasColumnName("biography1");
    entity.Property(e => e.Biography2).HasColumnName("biography2");
    entity.Property(e => e.Latitude).HasColumnType("character varying");
    entity.Property(e => e.Longitude).HasColumnType("character varying");
    entity.Property(e => e.MaDkbhyt)
        .HasMaxLength(10)
        .HasColumnName("MaDKBHYT");
    entity.Property(e => e.Name).HasColumnType("character varying");
    entity.Property(e => e.NameUnsign).HasColumnType("character varying");
    entity.Property(e => e.PhoneNumber).HasColumnType("character varying");
    entity.Property(e => e.Representative).HasColumnType("character varying");
    entity.Property(e => e.ZnsRatingTemplateId).HasMaxLength(200);
});

modelBuilder.Entity<LikeComment>(entity =>
{
    entity.HasKey(e => e.Id).HasName("LikeComment_pkey");

    entity.ToTable("LikeComment");

    entity.Property(e => e.Id).ValueGeneratedNever();
});

modelBuilder.Entity<NationCategory>(entity =>
{
    entity.HasKey(e => e.NationId).HasName("EthenicCategory_pkey");

    entity.ToTable("NationCategory");

    entity.Property(e => e.NationId).HasMaxLength(3);
    entity.Property(e => e.NationName)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<NationalityCategory>(entity =>
{
    entity.HasKey(e => e.NationalityId).HasName("NationalityCategory_pkey");

    entity.ToTable("NationalityCategory", tb => tb.HasComment("Danh mục quốc gia"));

    entity.Property(e => e.NationalityId).HasMaxLength(3);
    entity.Property(e => e.NationalityName)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<Notification>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Notification_pkey");

    entity.ToTable("Notification");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Content).HasColumnType("character varying");
    entity.Property(e => e.ShortDescription).HasColumnType("character varying");
    entity.Property(e => e.Subject).HasColumnType("character varying");
});

modelBuilder.Entity<NotificationReceiver>(entity =>
{
    entity.HasKey(e => e.Id).HasName("NotificationReceiver_pkey");

    entity.ToTable("NotificationReceiver");

    entity.Property(e => e.Id).ValueGeneratedNever();

    entity.HasOne(d => d.Notification).WithMany(p => p.NotificationReceivers)
        .HasForeignKey(d => d.NotificationId)
        .OnDelete(DeleteBehavior.ClientSetNull)
        .HasConstraintName("NotificationReceiver_NotificationId_fkey");
});

modelBuilder.Entity<Partner>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Partners_pkey");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Department).HasColumnType("character varying");
    entity.Property(e => e.FullName).HasColumnType("character varying");
    entity.Property(e => e.IsDeleted).HasDefaultValueSql("false");
    entity.Property(e => e.PhoneNumber).HasColumnType("character varying");
    entity.Property(e => e.ReferralCode)
        .IsRequired()
        .HasColumnType("character varying");
    entity.Property(e => e.Workplace).HasColumnType("character varying");
});

modelBuilder.Entity<PatientCompleteOrder>(entity =>
{
    entity.HasKey(e => e.Id).HasName("PatientCompleteOrder_pkey");

    entity.ToTable("PatientCompleteOrder");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.BacSiDieuTri).HasColumnType("character varying");
    entity.Property(e => e.BenhNhanId).HasMaxLength(500);
    entity.Property(e => e.Comment).HasMaxLength(2000);
    entity.Property(e => e.Email).HasMaxLength(200);
    entity.Property(e => e.ExternalId).HasMaxLength(200);
    entity.Property(e => e.MaBenhVien)
        .IsRequired()
        .HasMaxLength(500);
    entity.Property(e => e.MaBhyt)
        .HasMaxLength(500)
        .HasColumnName("MaBHYT");
    entity.Property(e => e.MaKcb).HasMaxLength(500);
    entity.Property(e => e.NoteFromAdmin)
        .HasMaxLength(500)
        .IsFixedLength();
    entity.Property(e => e.SoDienThoai).HasMaxLength(500);
    entity.Property(e => e.TenBenhNhan).HasMaxLength(500);
    entity.Property(e => e.TenBenhVien).HasMaxLength(500);
    entity.Property(e => e.TenHinhThucDieuTri).HasMaxLength(500);
    entity.Property(e => e.ThoiGianDkKham).HasColumnType("timestamp without time zone");
    entity.Property(e => e.ThoiGianKetThucKham).HasColumnType("timestamp without time zone");
    entity.Property(e => e.TongTienThanhToan).HasPrecision(18, 4);
});

modelBuilder.Entity<PhanTramKinhDoanh>(entity =>
{
    entity.HasKey(e => e.Id).HasName("PhanTramKinhDoanh_pkey");

    entity.ToTable("PhanTramKinhDoanh");

    entity.Property(e => e.Id).ValueGeneratedNever();
});

modelBuilder.Entity<PhuongAnKinhDoanhKsk>(entity =>
{
    entity.HasKey(e => e.Id).HasName("PhuongAnKinhDoanhKsk_pkey");

    entity.ToTable("PhuongAnKinhDoanhKsk");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.MaPhuongAn)
        .IsRequired()
        .HasColumnType("character varying");
    entity.Property(e => e.NgayTraCk).HasColumnName("NgayTraCK");
    entity.Property(e => e.NgayTraHs).HasColumnName("NgayTraHS");
    entity.Property(e => e.TenPhuongAn)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<PhuongAnKinhDoanhKskChiTiet>(entity =>
{
    entity.HasKey(e => e.Id).HasName("PhuongAnKinhDoanhKskChiTiet_pkey");

    entity.ToTable("PhuongAnKinhDoanhKskChiTiet");

    entity.Property(e => e.Id).ValueGeneratedNever();
});

modelBuilder.Entity<ProvinceCategory>(entity =>
{
    entity.HasKey(e => e.ProvinceId).HasName("ProvinceCategory_pkey");

    entity.ToTable("ProvinceCategory", tb => tb.HasComment("Danh mục tỉnh thành"));

    entity.Property(e => e.ProvinceId).HasMaxLength(3);
    entity.Property(e => e.ProvinceName)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<RatingRecommendContent>(entity =>
{
    entity.HasKey(e => e.Id).HasName("RatingRecommendContent_pkey");

    entity.ToTable("RatingRecommendContent");

    entity.Property(e => e.Id).UseIdentityAlwaysColumn();
    entity.Property(e => e.Content).HasMaxLength(500);
});

modelBuilder.Entity<RelationshipCategory>(entity =>
{
    entity.HasKey(e => e.Id).HasName("RelationshipCategory_pkey");

    entity.ToTable("RelationshipCategory", tb => tb.HasComment("Danh mục mối quan hệ"));

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Name)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<Setting>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Setting_pkey");

    entity.ToTable("Setting");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.HotLine).HasColumnType("character varying");
    entity.Property(e => e.SdtdvcapCuu)
        .HasMaxLength(150)
        .HasColumnName("SDTDVCapCuu");
    entity.Property(e => e.ThoiGianNhacHen).HasColumnType("character varying");
    entity.Property(e => e.ThongTinDvcapCuu).HasColumnName("ThongTinDVCapCuu");
});

modelBuilder.Entity<Speciality>(entity =>
{
    entity.HasKey(e => e.Id).HasName("Speciality_pkey");

    entity.ToTable("Speciality");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Name).HasColumnType("character varying");
});

modelBuilder.Entity<ThanhLyHopDongKsk>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThanhLyHopDongKsk_pkey");

    entity.ToTable("ThanhLyHopDongKsk");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.ChietKhauHt).HasColumnName("ChietKhauHT");
    entity.Property(e => e.ChietKhauKd).HasColumnName("ChietKhauKD");
    entity.Property(e => e.MaThanhLy)
        .IsRequired()
        .HasColumnType("character varying");
    entity.Property(e => e.NgayTraCk).HasColumnName("NgayTraCK");
    entity.Property(e => e.NgayTraHs).HasColumnName("NgayTraHS");
    entity.Property(e => e.TenThanhLy)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<ThanhLyHopDongKskChiTiet>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThanhLyHopDongKskChiTiet_pkey");

    entity.ToTable("ThanhLyHopDongKskChiTiet");

    entity.Property(e => e.Id).ValueGeneratedNever();
});

modelBuilder.Entity<ThanhLyHopDongKskChiTietKhac>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThanhLyHopDongKskChiTietKhac_pkey");

    entity.ToTable("ThanhLyHopDongKskChiTietKhac");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.TenChiPhi)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<ThietLapDanhGiaCldv>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThietLapDanhGiaCLDV_pkey");

    entity.ToTable("ThietLapDanhGiaCLDV");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.Ten).HasColumnType("character varying");
});

modelBuilder.Entity<ThietLapKskDoan>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThietLapKskDoan_pkey");

    entity.ToTable("ThietLapKskDoan");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.TenThietLap)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<ThuTien>(entity =>
{
    entity.HasKey(e => e.Id).HasName("ThuTien_pkey");

    entity.ToTable("ThuTien");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.NguoiNop).HasColumnType("character varying");
    entity.Property(e => e.NoiDung).HasColumnType("character varying");
    entity.Property(e => e.SoHoaDon).HasMaxLength(15);
    entity.Property(e => e.SoPhieu)
        .IsRequired()
        .HasMaxLength(15);
});

modelBuilder.Entity<TimeSlotCategory>(entity =>
{
    entity.HasKey(e => e.Id).HasName("TimeSlotCategory_pkey");

    entity.ToTable("TimeSlotCategory");

    entity.Property(e => e.Id).ValueGeneratedNever();
    entity.Property(e => e.IsAvailable)
        .IsRequired()
        .HasDefaultValueSql("true");
    entity.Property(e => e.Title)
        .IsRequired()
        .HasColumnType("character varying");
});

modelBuilder.Entity<WardCategory>(entity =>
{
    entity.HasKey(e => e.WardId).HasName("CommuneCategory_pkey");

    entity.ToTable("WardCategory", tb => tb.HasComment("Danh mục phường xã"));

    entity.Property(e => e.WardId).HasMaxLength(5);
    entity.Property(e => e.DistrictId)
```

```
                .IsRequired()
                .HasMaxLength(4);
            entity.Property(e => e.ProvinceId)
                .IsRequired()
                .HasMaxLength(3);
            entity.Property(e => e.WardName)
                .IsRequired()
                .HasColumnType("character varying");
        });

        modelBuilder.Entity<ZaloOaInfo>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("ZaloOaInfo_pkey");

            entity.ToTable("ZaloOaInfo");

            entity.Property(e => e.Id).UseIdentityAlwaysColumn();
            entity.Property(e => e.AccessToken).HasMaxLength(2000);
            entity.Property(e => e.AppId).HasMaxLength(200);
            entity.Property(e => e.Name).HasMaxLength(500);
            entity.Property(e => e.RefreshToken).HasMaxLength(2000);
            entity.Property(e => e.SecretKey).HasMaxLength(200);
        });

        modelBuilder.Entity<ZaloOaTemplate>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("ZaloOaTemplate_pkey");

            entity.ToTable("ZaloOaTemplate");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.Description).HasMaxLength(200);
            entity.Property(e => e.TemplateId)
                .IsRequired()
                .HasMaxLength(50);
        });

        modelBuilder.Entity<ZnsHistory>(entity =>
        {
            entity.HasKey(e => e.Id).HasName("ZnsHistory_pkey");

            entity.ToTable("ZnsHistory");

            entity.Property(e => e.Id).ValueGeneratedNever();
            entity.Property(e => e.ObjectId)
                .IsRequired()
                .HasMaxLength(200);
            entity.Property(e => e.TypeName).HasMaxLength(200);
        });

        OnModelCreatingPartial(modelBuilder);
    }

    partial void OnModelCreatingPartial(ModelBuilder modelBuilder);
}
```