```csharp
// WardrobeOS.Data/AppDbContext.cs
using System;
using Microsoft.EntityFrameworkCore;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace WardrobeOS.Data;

public sealed class AppDbContext : DbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options) : base(options) {}

    // --- DbSets (Этап 1)
    public DbSet<User> Users => Set<User>();
    public DbSet<Avatar> Avatars => Set<Avatar>();
    public DbSet<AvatarCapture> AvatarCaptures => Set<AvatarCapture>();

    public DbSet<Brand> Brands => Set<Brand>();
    public DbSet<Category> Categories => Set<Category>();
    public DbSet<Condition> Conditions => Set<Condition>();
    public DbSet<Fit> Fits => Set<Fit>();
    public DbSet<Color> Colors => Set<Color>();
    public DbSet<Pattern> Patterns => Set<Pattern>();
    public DbSet<Material> Materials => Set<Material>();
    public DbSet<Season> Seasons => Set<Season>();
    public DbSet<Currency> Currencies => Set<Currency>();

    public DbSet<Wardrobe> Wardrobes => Set<Wardrobe>();
    public DbSet<Garment> Garments => Set<Garment>();
    public DbSet<GarmentImage> GarmentImages => Set<GarmentImage>();
    public DbSet<GarmentColor> GarmentColors => Set<GarmentColor>();
    public DbSet<GarmentPattern> GarmentPatterns => Set<GarmentPattern>();
    public DbSet<GarmentMaterial> GarmentMaterials => Set<GarmentMaterial>();
    public DbSet<GarmentSeason> GarmentSeasons => Set<GarmentSeason>();
    public DbSet<GarmentTag> GarmentTags => Set<GarmentTag>();
    public DbSet<GarmentCapture> GarmentCaptures => Set<GarmentCapture>();
    public DbSet<GarmentVector> GarmentVectors => Set<GarmentVector>();
    public DbSet<MediaIngestJob> MediaIngestJobs => Set<MediaIngestJob>();

    // --- DbSets (Этап 2)
    public DbSet<Outfit> Outfits => Set<Outfit>();
    public DbSet<OutfitItem> OutfitItems => Set<OutfitItem>();
    public DbSet<Follow> Follows => Set<Follow>();
    public DbSet<UserOutfitReaction> UserOutfitReactions => Set<UserOutfitReaction>();

    public DbSet<SearchIndexState> SearchIndexStates => Set<SearchIndexState>();
    public DbSet<ContentLabel> ContentLabels => Set<ContentLabel>();
    public DbSet<AuditLog> AuditLogs => Set<AuditLog>();

    // --- DbSets (Этап 3)
    public DbSet<Listing> Listings => Set<Listing>();
    public DbSet<Interest> Interests => Set<Interest>();
    public DbSet<InterestMessage> InterestMessages => Set<InterestMessage>();
    public DbSet<SwapItem> SwapItems => Set<SwapItem>();
    public DbSet<PeerReview> PeerReviews => Set<PeerReview>();

    protected override void OnModelCreating(ModelBuilder b)
    {
        // ---------- USERS
        b.Entity<User>(e =>
        {
            e.ToTable("users");
            e.HasKey(x => x.Id).IsClustered(false);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Email).HasMaxLength(256).IsRequired();
            e.Property(x => x.EmailNorm)
                .HasComputedColumnSql("LOWER([email])", stored: true);
            e.HasIndex(x => x.EmailNorm).IsUnique().HasDatabaseName("UX_users_email_norm");
            e.Property(x => x.Name).HasMaxLength(128);
            e.Property(x => x.Role).HasMaxLength(20).HasDefaultValue("user");
            e.Property(x => x.Status).HasMaxLength(20).HasDefaultValue("active");
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.Property(x => x.RowVersion).IsRowVersion();
            e.HasIndex(x => x.CreatedAt).IsClustered().HasDatabaseName("CI_users_created")
             .IsDescending(true);
        });


        // ---------- AVATAR
        b.Entity<Avatar>(e =>
        {
            e.ToTable("avatars");
            e.HasKey(x => x.UserId);
            e.HasOne(x => x.User).WithOne(x => x.Avatar).HasForeignKey<Avatar>(x => x.UserId).OnDelete(DeleteBehavior.NoAction);
            e.Property(x => x.Sex).HasMaxLength(12);
            e.Property(x => x.SkinTone).HasMaxLength(24);
            e.Property(x => x.HairColor).HasMaxLength(24);
            e.Property(x => x.BodyType).HasMaxLength(24);
            e.Property(x => x.PxPerMm).HasPrecision(9,6);
            e.Property(x => x.UpdatedAt).HasDefaultValueSql("sysutcdatetime()");
        });

        b.Entity<AvatarCapture>(e =>
        {
            e.ToTable("avatar_captures");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.View).HasMaxLength(16).IsRequired();
            e.Property(x => x.Calibrator).HasMaxLength(16);
            e.Property(x => x.PxPerMm).HasPrecision(9,6);
            e.Property(x => x.Device).HasMaxLength(64);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasOne<User>().WithMany().HasForeignKey(x => x.UserId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => new { x.UserId, x.CreatedAt }).HasDatabaseName("IX_avatar_captures_user").IsDescending(true);
        });


        // ---------- LOOKUPS
        // Somnitelno, mozno i potom
        b.Entity<Brand>(e =>
        {
            e.ToTable("brands");
            e.HasKey(x => x.Id);
            e.Property(x => x.Name).HasMaxLength(128).IsRequired();
            e.HasIndex(x => x.Name).IsUnique();
        });

        b.Entity<Category>(e =>
        {
            e.ToTable("categories");
            e.HasKey(x => x.Id);
            e.Property(x => x.Code).HasMaxLength(32).IsRequired();
            e.Property(x => x.Name).HasMaxLength(64).IsRequired();
            e.HasIndex(x => x.Code).IsUnique();
        });

        b.Entity<Condition>(e =>
        {
            e.ToTable("conditions");
            e.HasKey(x => x.Id);
            e.Property(x => x.Code).HasMaxLength(16).IsRequired();
            e.Property(x => x.Name).HasMaxLength(64).IsRequired();
            e.HasIndex(x => x.Code).IsUnique();
        });

        b.Entity<Fit>(e =>
        {
            e.ToTable("fits");
            e.HasKey(x => x.Id);
            e.Property(x => x.Code).HasMaxLength(16).IsRequired();
            e.Property(x => x.Name).HasMaxLength(64).IsRequired();
            e.HasIndex(x => x.Code).IsUnique();
        });

        b.Entity<Color>(e =>
        {
            e.ToTable("colors", tb =>
            {
                tb.HasCheckConstraint("CK_colors_hex", "hex LIKE '#[0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f][0-9A-Fa-f]'");
                tb.HasCheckConstraint("CK_colors_lab_l", "[l] BETWEEN 0 AND 100");
                tb.HasCheckConstraint("CK_colors_lab_a", "[a] BETWEEN -128 AND 127");
                tb.HasCheckConstraint("CK_colors_lab_b", "[b] BETWEEN -128 AND 127");
            });
            e.HasKey(x => x.Id);
            e.Property(x => x.Name).HasMaxLength(32).IsRequired();
            e.HasIndex(x => x.Name).IsUnique();
            e.Property(x => x.Hex).HasColumnName("hex").HasMaxLength(7).IsRequired();
            e.HasIndex(x => x.Hex).IsUnique();
        });

        b.Entity<Pattern>(e =>
        {
            e.ToTable("patterns");
            e.HasKey(x => x.Id);
            e.Property(x => x.Name).HasMaxLength(32).IsRequired();
            e.HasIndex(x => x.Name).IsUnique();
        });

        b.Entity<Material>(e =>
        {
            e.ToTable("materials");
            e.HasKey(x => x.Id);
            e.Property(x => x.Name).HasMaxLength(32).IsRequired();
            e.HasIndex(x => x.Name).IsUnique();
        });

        b.Entity<Season>(e =>
        {
            e.ToTable("seasons");
            e.HasKey(x => x.Code);
            e.Property(x => x.Code).HasMaxLength(16);
            e.Property(x => x.Name).HasMaxLength(32).IsRequired();
        });

        b.Entity<Currency>(e =>
        {
            e.ToTable("currencies");
            e.HasKey(x => x.Code);
            e.Property(x => x.Code).HasMaxLength(3);
            e.Property(x => x.Name).HasMaxLength(64).IsRequired();
        });


        // ---------- WARDROBES
        b.Entity<Wardrobe>(e =>
        {
            e.ToTable("wardrobes");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.HasOne(x => x.Owner).WithMany(x => x.Wardrobes).HasForeignKey(x => x.OwnerId).OnDelete(DeleteBehavior.NoAction);
            e.Property(x => x.Title).HasMaxLength(120).IsRequired();
            e.Property(x => x.Description).HasMaxLength(500);
            e.Property(x => x.Visibility).HasMaxLength(7).HasDefaultValue("private");
            e.Property(x => x.IsDefault).HasDefaultValue(false);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.OwnerId, x.CreatedAt }).IsClustered().IsDescending(true).HasDatabaseName("CI_wardrobes_owner_created");
            e.HasIndex(x => x.OwnerId).HasFilter("[is_default] = 1").IsUnique().HasDatabaseName("UX_wardrobes_default");
        });


        // ---------- GARMENTS
        b.Entity<Garment>(e =>
        {
            e.ToTable("garments", tb =>
            {
                tb.HasCheckConstraint("CK_garments_sale_price",
                    "([is_for_sale] = 0 AND [price_minor] IS NULL) OR ([is_for_sale] = 1 AND [price_minor] IS NOT NULL)");
            });
            e.HasKey(x => x.Id).IsClustered(false);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.HasOne(x => x.Owner).WithMany(x => x.Garments).HasForeignKey(x => x.OwnerId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Wardrobe).WithMany(x => x.Garments).HasForeignKey(x => x.WardrobeId).OnDelete(DeleteBehavior.NoAction);
            e.Property(x => x.Title).HasMaxLength(160).IsRequired();
            e.Property(x => x.Size).HasMaxLength(32);
            e.Property(x => x.Visibility).HasMaxLength(7).HasDefaultValue("private");
            e.Property(x => x.IsForSale).HasDefaultValue(false);
            e.Property(x => x.CurrencyCode).HasMaxLength(3).HasDefaultValue("EUR");
            e.Property(x => x.PxPerMm).HasPrecision(9,6);
            e.Property(x => x.PhashPrefix)
                .HasComputedColumnSql("CONVERT(binary(4), SUBSTRING([phash],(1),(4)))", stored:true);
            e.Property(x => x.IsHidden).HasDefaultValue(false);
            e.Property(x => x.ModerationStatus).HasMaxLength(16).HasDefaultValue("visible");
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.Property(x => x.RowVersion).IsRowVersion();

            e.HasIndex(x => new { x.OwnerId, x.CreatedAt }).IsClustered().IsDescending(true)
                .HasDatabaseName("CI_garments_owner_created");
            e.HasIndex(x => new { x.WardrobeId, x.CreatedAt }).HasDatabaseName("IX_garments_wardrobe").IsDescending(true);
            e.HasIndex(x => x.CategoryId).HasDatabaseName("IX_garments_category");
            e.HasIndex(x => x.BrandId).HasDatabaseName("IX_garments_brand");
            e.HasIndex(x => x.ConditionId).HasDatabaseName("IX_garments_condition");
            e.HasIndex(x => x.Phash).HasDatabaseName("IX_garments_phash");
            e.HasIndex(x => x.Visibility)
                .HasFilter("[visibility] = 'public'")
                .IncludeProperties(x => new { x.OwnerId, x.WardrobeId, x.CategoryId })
                .HasDatabaseName("IX_garments_public");
            e.HasIndex(x => new { x.IsForSale, x.CreatedAt })
                .HasFilter("[is_for_sale] = 1")
                .IncludeProperties(x => new { x.PriceMinor, x.CurrencyCode, x.OwnerId, x.CategoryId })
                .HasDatabaseName("IX_garments_sale");

            e.HasIndex(x => new { x.OwnerId, x.Phash })
                .HasFilter("[phash] IS NOT NULL")
                .IsUnique()
                .HasDatabaseName("UX_garments_owner_phash");
        });

        b.Entity<GarmentImage>(e =>
        {
            e.ToTable("garment_images");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Url).HasMaxLength(512).IsRequired();
            e.Property(x => x.IsPrimary).HasDefaultValue(false);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasOne(x => x.Garment).WithMany(x => x.Images).HasForeignKey(x => x.GarmentId).OnDelete(DeleteBehavior.Cascade);
            e.HasIndex(x => x.GarmentId).HasFilter("[is_primary] = 1").IsUnique().HasDatabaseName("UX_garment_images_primary");
            // (правило "нельзя остаться без primary" реализуется триггером/доменной логикой)
        });

        b.Entity<GarmentColor>(e =>
        {
            e.ToTable("garment_colors");
            e.HasKey(x => new { x.GarmentId, x.ColorId });
            e.Property(x => x.IsPrimary).HasDefaultValue(false);
            e.HasIndex(x => new { x.ColorId, x.GarmentId }).HasDatabaseName("IX_color_to_garments");
        });

        b.Entity<GarmentPattern>(e =>
        {
            e.ToTable("garment_patterns");
            e.HasKey(x => new { x.GarmentId, x.PatternId });
            e.HasIndex(x => new { x.PatternId, x.GarmentId }).HasDatabaseName("IX_pattern_to_garments");
        });

        b.Entity<GarmentMaterial>(e =>
        {
            e.ToTable("garment_materials");
            e.HasKey(x => new { x.GarmentId, x.MaterialId });
            e.HasIndex(x => new { x.MaterialId, x.GarmentId }).HasDatabaseName("IX_material_to_garments");
        });

        b.Entity<GarmentSeason>(e =>

        {
            e.ToTable("garment_seasons");
            e.HasKey(x => new { x.GarmentId, x.SeasonCode });
            e.HasIndex(x => new { x.SeasonCode, x.GarmentId }).HasDatabaseName("IX_season_to_garments");
        });

        b.Entity<GarmentTag>(e =>
        {
            e.ToTable("garment_tags");
            e.HasKey(x => new { x.GarmentId, x.Tag });
            e.Property(x => x.Tag).HasMaxLength(64).IsRequired();
            e.HasIndex(x => new { x.Tag, x.GarmentId }).HasDatabaseName("IX_tag_to_garments");
        });

        b.Entity<GarmentCapture>(e =>
        {
            e.ToTable("garment_captures");
```

```csharp
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.PxPerMm).HasPrecision(9,6);
            e.Property(x => x.Calibrator).HasMaxLength(16);
            e.Property(x => x.Device).HasMaxLength(64);
            e.Property(x => x.Notes).HasMaxLength(256);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
        });

        b.Entity<GarmentVector>(e =>
        {
            e.ToTable("garment_vectors", tb =>
            {
                tb.HasCheckConstraint("CK_garment_vectors_len", "DATALENGTH([vector]) = CONVERT(int, [dims]) * 4");
            });
            e.HasKey(x => x.GarmentId);
            e.Property(x => x.Model).HasMaxLength(64).IsRequired();
            e.Property(x => x.Dims).HasColumnType("smallint");
            e.Property(x => x.IsL2Normalized).HasDefaultValue(true);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
        });

        b.Entity<MediaIngestJob>(e =>
        {
            e.ToTable("media_ingest_jobs");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.TargetType).HasMaxLength(16).IsRequired();
            e.Property(x => x.Status).HasMaxLength(16).HasDefaultValue("queued");
            e.Property(x => x.ErrorMessage).HasMaxLength(1000);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.OwnerId, x.CreatedAt }).HasDatabaseName("IX_ingest_owner_created").IsDescending(true);
        });


        // ---------- OUTFITS (этап 2)
        b.Entity<Outfit>(e =>
        {
            e.ToTable("outfits");
            e.HasKey(x => x.Id).IsClustered(false);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Title).HasMaxLength(160);
            e.Property(x => x.Note).HasMaxLength(1000);
            e.Property(x => x.Visibility).HasMaxLength(7).HasDefaultValue("private");
            e.Property(x => x.IsHidden).HasDefaultValue(false);
            e.Property(x => x.ModerationStatus).HasMaxLength(16).HasDefaultValue("visible");
            e.Property(x => x.LikeCount).HasDefaultValue(0);
            e.Property(x => x.SaveCount).HasDefaultValue(0);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.Property(x => x.RowVersion).IsRowVersion();
            e.HasOne(x => x.Owner).WithMany(x => x.Outfits).HasForeignKey(x => x.OwnerId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Wardrobe).WithMany().HasForeignKey(x => x.WardrobeId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => new { x.OwnerId, x.CreatedAt }).IsClustered().IsDescending(true).HasDatabaseName("CI_outfits_owner_created");
            e.HasIndex(x => x.Visibility)
                .HasFilter("[visibility] = 'public'")
                .IncludeProperties(x => new { x.OwnerId, x.LikeCount, x.SaveCount })
                .HasDatabaseName("IX_outfits_visibility_created");
        });

        b.Entity<OutfitItem>(e =>
        {
            e.ToTable("outfit_items");
            e.HasKey(x => new { x.OutfitId, x.GarmentId });
            e.Property(x => x.RotationDeg).HasPrecision(6,2);
            e.Property(x => x.Scale).HasPrecision(9,4);
            e.Property(x => x.X).HasPrecision(9,4);
            e.Property(x => x.Y).HasPrecision(9,4);
            e.HasOne(x => x.Outfit).WithMany(x => x.Items).HasForeignKey(x => x.OutfitId).OnDelete(DeleteBehavior.Cascade);
            e.HasOne(x => x.Garment).WithMany().HasForeignKey(x => x.GarmentId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => new { x.GarmentId, x.OutfitId }).HasDatabaseName("IX_item_garment_to_outfits");
        });

        b.Entity<Follow>(e =>
        {
            e.ToTable("follows");
            e.HasKey(x => new { x.FollowerId, x.FolloweeId });
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasOne(x => x.Follower).WithMany().HasForeignKey(x => x.FollowerId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Followee).WithMany().HasForeignKey(x => x.FolloweeId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => new { x.FolloweeId, x.CreatedAt }).HasDatabaseName("IX_followee").IsDescending(true);
        });

        b.Entity<UserOutfitReaction>(e =>
        {
            e.ToTable("user_outfit_reactions");
            e.HasKey(x => new { x.UserId, x.OutfitId, x.Type });
            e.Property(x => x.Type).HasMaxLength(8).IsRequired();
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.OutfitId, x.Type }).HasDatabaseName("IX_reactions_outfit");
        });


        // ---------- TECH / MODERATION / AUDIT
        b.Entity<SearchIndexState>(e =>
        {
            e.ToTable("search_index_state");
            e.HasKey(x => new { x.EntityType, x.EntityId, x.IndexName });
            e.Property(x => x.EntityType).HasMaxLength(16).IsRequired();
            e.Property(x => x.IndexName).HasMaxLength(64).IsRequired();
            e.Property(x => x.IndexVersion).HasDefaultValue(1);
            e.Property(x => x.VectorStatus).HasMaxLength(16).HasDefaultValue("pending");
        });

        b.Entity<ContentLabel>(e =>
        {
            e.ToTable("content_labels");
            e.HasKey(x => x.Id);
            e.Property(x => x.TargetType).HasMaxLength(16).IsRequired();
            e.Property(x => x.Label).HasMaxLength(32).IsRequired();
            e.Property(x => x.Score).HasPrecision(5,4).IsRequired();
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.TargetType, x.TargetId }).HasDatabaseName("IX_labels_target");
        });

        b.Entity<AuditLog>(e =>
        {
            e.ToTable("audit_log");
            e.HasKey(x => x.Id);
            e.Property(x => x.Action).HasMaxLength(32).IsRequired();
            e.Property(x => x.Payload);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.EntityType, x.EntityId, x.CreatedAt }).HasDatabaseName("IX_audit_entity").IsDescending(true);
        });


        // ---------- LISTINGS / INTERESTS (этап 3)
        b.Entity<Listing>(e =>
        {
            e.ToTable("listings");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Type).HasMaxLength(12).HasDefaultValue("gift"); // gift|swap|sell
            e.Property(x => x.Note).HasMaxLength(500);
            e.Property(x => x.Status).HasMaxLength(12).HasDefaultValue("active");
            e.Property(x => x.PriceMinor);
            e.Property(x => x.CurrencyCode).HasMaxLength(3).HasDefaultValue("EUR");
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.Property(x => x.UpdatedAt);

            e.HasOne(x => x.Owner).WithMany().HasForeignKey(x => x.OwnerId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Garment).WithMany().HasForeignKey(x => x.GarmentId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Outfit).WithMany().HasForeignKey(x => x.OutfitId).OnDelete(DeleteBehavior.NoAction);

            // XOR check
            e.ToTable(tb => tb.HasCheckConstraint("CK_listings_xor",
                "(CASE WHEN [garment_id] IS NULL THEN 0 ELSE 1 END) + CASE WHEN [outfit_id] IS NULL THEN 0 ELSE 1 END) = 1"));

            e.HasIndex(x => new { x.Status, x.CreatedAt })
             .HasFilter("[status] = 'active'")
             .IncludeProperties(x => new { x.OwnerId, x.Type, x.PriceMinor, x.CurrencyCode })
             .HasDatabaseName("IX_listings_active");
            e.HasIndex(x => new { x.OwnerId, x.CreatedAt }).HasDatabaseName("IX_listings_owner").IsDescending(true);
        });

        b.Entity<Interest>(e =>
        {
            e.ToTable("interests");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Kind).HasMaxLength(12).HasDefaultValue("inquiry");
            e.Property(x => x.OfferCurrency).HasMaxLength(3).HasDefaultValue("EUR");
            e.Property(x => x.Status).HasMaxLength(16).HasDefaultValue("open");
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.Property(x => x.UpdatedAt);

            e.HasOne(x => x.Listing).WithMany().HasForeignKey(x => x.ListingId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Initiator).WithMany().HasForeignKey(x => x.InitiatorId).OnDelete(DeleteBehavior.NoAction);
            e.HasOne(x => x.Owner).WithMany().HasForeignKey(x => x.OwnerId).OnDelete(DeleteBehavior.NoAction);

            e.HasIndex(x => new { x.OwnerId, x.Status, x.CreatedAt }).HasDatabaseName("IX_interests_owner_status").IsDescending(true);
            e.HasIndex(x => new { x.InitiatorId, x.CreatedAt }).HasDatabaseName("IX_interests_initiator").IsDescending(true);
        });

        b.Entity<InterestMessage>(e =>
        {
            e.ToTable("interest_messages");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Body).HasMaxLength(2000).IsRequired();
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasOne(x => x.Interest).WithMany().HasForeignKey(x => x.InterestId).OnDelete(DeleteBehavior.Cascade);
            e.HasOne(x => x.Sender).WithMany().HasForeignKey(x => x.SenderId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => new { x.InterestId, x.CreatedAt }).HasDatabaseName("IX_interest_messages");
        });

        b.Entity<SwapItem>(e =>
        {
            e.ToTable("swap_items");
            e.HasKey(x => new { x.InterestId, x.GarmentId, x.Role });
            e.Property(x => x.Role).HasMaxLength(8).HasDefaultValue("offer"); // offer|request
            e.HasOne(x => x.Garment).WithMany().HasForeignKey(x => x.GarmentId).OnDelete(DeleteBehavior.NoAction);
            e.HasIndex(x => x.InterestId).HasDatabaseName("IX_swap_items_interest");
        });

        b.Entity<PeerReview>(e =>
        {
            e.ToTable("peer_reviews");
            e.HasKey(x => x.Id);
            e.Property(x => x.Id).HasDefaultValueSql("NEWSEQUENTIALID()");
            e.Property(x => x.Rating).HasColumnType("tinyint");
            e.Property(x => x.Comment).HasMaxLength(500);
            e.Property(x => x.CreatedAt).HasDefaultValueSql("sysutcdatetime()");
            e.HasIndex(x => new { x.RevieweeId, x.CreatedAt }).HasDatabaseName("IX_reviews_user").IsDescending(true);
            e.HasIndex(x => new { x.InterestId, x.ReviewerId }).IsUnique().HasDatabaseName("UX_peer_reviews_once");
        });
    }
}
```

```
// ========================= ENTITIES =========================

// Этап 1
```

	KEY	VALUE
1	Id:	Pagrindinis raktas (GUID).
2	Email:	Vartotojo el. paštas (unikalus).
3	Name:	Rodomas vardas (nebūtinas).
4	Role:	Rolė: user|moderator|admin.
5	Status:	Būsena: active|blocked|deleted (soft delete).
6	CreatedAt:	Sukūrimo UTC laikas.
7	EmailNorm:	Normalizuotas el. paštas (lowercase, tik skaitymui).
8	RowVersion:	Konkurencingumo žyma (SQL Server rowversion) optimistinei blokavimų kontrolei.
9	Avatar:	1:1 ryšys su vartotojo avataru.
10	Wardrobes:	1:N vartotojo spintos.
11	Garments:	1:N visos vartotojo drabužių įrašai.
12	Outfits:	1:N vartotojo sukurti deriniai.

```
public class User
{
    public Guid Id { get; set; }                        //Nado
    public string Email { get; set; } = default!;       //Nado
    public string? Name { get; set; }                   //Nado
    public string Role { get; set; } = "user";          //Nado
    public string Status { get; set; } = "active";      //Nado
    public DateTime CreatedAt { get; set; }             //Nado
    public string? EmailNorm { get; private set; } // computed (podumaju 4to eto)
    public byte[] RowVersion { get; set; } = default!;  //4to eto?

    public Avatar? Avatar { get; set; }                 //Sozdadim v 3D?
    public ICollection<Wardrobe> Wardrobes { get; set; } = new List<Wardrobe>();    //Skafy?
    public ICollection<Garment> Garments { get; set; } = new List<Garment>();       //Sety?
    public ICollection<Outfit> Outfits { get; set; } = new List<Outfit>();          //Odezda?
}
```

	KEY	VALUE
1	UserId:	PK ir FK → User.Id.
2	User:	Navigacija į vartotoją.
3	Sex:	Lytis (laisvas tekstas; rekomenduojama male|female|other).
4	HeightCm / WeightKg:	Ūgis / svoris (cm / kg).
5	ChestCm / WaistCm / HipsCm / FootCm / ShoulderCm:	Kūno apimtys (mm arba cm – čia cm).
6	SkinTone / HairColor / BodyType:	Išvaizdos atributai (laisvi).
7	PxPerMm:	Kalibravimas (pikseliai per mm) pagal A4/kortelę.
8	UpdatedAt:	Paskutinio atnaujinimo UTC.

```
public class Avatar
{
    [Key] public Guid UserId { get; set; }
    public User User { get; set; } = default!;

    public string? Sex { get; set; }
    public int HeightCm { get; set; }
    public int WeightKg { get; set; }
    public int? ChestCm { get; set; }
    public int? WaistCm { get; set; }
    public int? HipsCm { get; set; }
    public int? FootCm { get; set; }
    public int? ShoulderCm { get; set; }
    public string? SkinTone { get; set; }
    public string? HairColor { get; set; }
    public string? BodyType { get; set; }
    public decimal? PxPerMm { get; set; }
    public DateTime UpdatedAt { get; set; }
}

public class AvatarCapture
{
    public Guid Id { get; set; }
    public Guid UserId { get; set; }
    public string View { get; set; } = default!;
    public string? Calibrator { get; set; }
    public decimal? PxPerMm { get; set; }
    public string? Device { get; set; }
    public DateTime CreatedAt { get; set; }
}

// Lookups
public class Brand
{
    public int Id { get; set; }
    public string Name { get; set; } = default!;
}
public class Category
{
    public int Id { get; set; }
    public string Code { get; set; } = default!;
    public string Name { get; set; } = default!;
}

public class Condition
{
    public int Id { get; set; }
    public string Code { get; set; } = default!;
    public string Name { get; set; } = default!;
```

```
}

public class Fit
{
	public int Id { get; set; }
	public string Code { get; set; } = default!;
	public string Name { get; set; } = default!;
}

public class Color
{
	public int Id { get; set; }
	public string Name { get; set; } = default!;
	public string Hex { get; set; } = default!;
	public double? L { get; set; }
	public double? A { get; set; }
	public double? B { get; set; }
}

public class Pattern
{
	public int Id { get; set; }
	public string Name { get; set; } = default!;
}

public class Material
{
	public int Id { get; set; }
	public string Name { get; set; } = default!;
}

public class Season
{
	[Key] public string Code { get; set; } = default!;
	public string Name { get; set; } = default!;
}

public class Currency
{
	[Key] public string Code { get; set; } = default!;
	public string Name { get; set; } = default!;
}

public class Wardrobe
{
	public Guid Id { get; set; }
	public Guid OwnerId { get; set; }
	public User Owner { get; set; } = default!;
	public string Title { get; set; } = default!;
	public string? Description { get; set; }
	public string Visibility { get; set; } = "private";
	public bool IsDefault { get; set; }
	public DateTime CreatedAt { get; set; }
	public DateTime? UpdatedAt { get; set; }

	public ICollection<Garment> Garments { get; set; } = new List<Garment>();
}

public class Garment
{
	public Guid Id { get; set; }
	public Guid OwnerId { get; set; }
	public User Owner { get; set; } = default!;
	public Guid WardrobeId { get; set; }
	public Wardrobe Wardrobe { get; set; } = default!;

	public string Title { get; set; } = default!;
	public int? BrandId { get; set; }
	public Brand? Brand { get; set; }
	public int CategoryId { get; set; }
	public Category Category { get; set; } = default!;
	public string? Size { get; set; }
	public int? FitId { get; set; }
	public Fit? Fit { get; set; }
	public int? ConditionId { get; set; }
	public Condition? Condition { get; set; }
	public string Visibility { get; set; } = "private";
	public bool IsForSale { get; set; }
	public int? PriceMinor { get; set; }
	public string? CurrencyCode { get; set; } = "EUR";
	public decimal? PxPerMm { get; set; }
	public byte[]? Phash { get; set; }
	public byte[]? PhashPrefix { get; private set; }   // computed
	public bool IsHidden { get; set; }
	public string ModerationStatus { get; set; } = "visible";
	public DateTime CreatedAt { get; set; }
	public DateTime? UpdatedAt { get; set; }
	public byte[] RowVersion { get; set; } = default!;

	public ICollection<GarmentImage> Images { get; set; } = new List<GarmentImage>();
	public ICollection<GarmentColor> Colors { get; set; } = new List<GarmentColor>();
	public ICollection<GarmentPattern> Patterns { get; set; } = new List<GarmentPattern>();
	public ICollection<GarmentMaterial> Materials { get; set; } = new List<GarmentMaterial>();
	public ICollection<GarmentSeason> Seasons { get; set; } = new List<GarmentSeason>();
	public ICollection<GarmentTag> Tags { get; set; } = new List<GarmentTag>();
}

public class GarmentImage
{
	public Guid Id { get; set; }
	public Guid GarmentId { get; set; }
	public Garment Garment { get; set; } = default!;
	public string Url { get; set; } = default!;
	public int? Width { get; set; }
	public int? Height { get; set; }
```

```csharp
	public bool IsPrimary { get; set; }
	public DateTime CreatedAt { get; set; }
}

public class GarmentColor
{
	public Guid GarmentId { get; set; }
	public int ColorId { get; set; }
	public bool IsPrimary { get; set; }
}

public class GarmentPattern
{
	public Guid GarmentId { get; set; }
	public int PatternId { get; set; }
}

public class GarmentMaterial
{
	public Guid GarmentId { get; set; }
	public int MaterialId { get; set; }
}

public class GarmentSeason
{
	public Guid GarmentId { get; set; }
	public string SeasonCode { get; set; } = default!;
}

public class GarmentTag
{
	public Guid GarmentId { get; set; }
	public string Tag { get; set; } = default!;
}

public class GarmentCapture
{
	public Guid Id { get; set; }
	public Guid GarmentId { get; set; }
	public decimal? PxPerMm { get; set; }
	public string? Calibrator { get; set; }
	public string? Device { get; set; }
	public string? Notes { get; set; }
	public DateTime CreatedAt { get; set; }
}

public class GarmentVector
{
	[Key] public Guid GarmentId { get; set; }
	public string Model { get; set; } = default!;
	public short Dims { get; set; }
	public bool IsL2Normalized { get; set; } = true;
	public byte[] Vector { get; set; } = default!;
	public DateTime CreatedAt { get; set; }
}

public class MediaIngestJob
{
	public Guid Id { get; set; }
	public Guid OwnerId { get; set; }
	public string TargetType { get; set; } = default!; // garment|avatar
	public Guid? TargetId { get; set; }
	public string Status { get; set; } = "queued";
	public string? ErrorMessage { get; set; }
	public DateTime CreatedAt { get; set; }
}

// Этап 2
public class Outfit
{
	public Guid Id { get; set; }
	public Guid OwnerId { get; set; }
	public User Owner { get; set; } = default!;
	public Guid? WardrobeId { get; set; }
	public Wardrobe? Wardrobe { get; set; }

	public string? Title { get; set; }
	public string? Note { get; set; }
	public string Visibility { get; set; } = "private";
	public bool IsHidden { get; set; }
	public string ModerationStatus { get; set; } = "visible";
	public int LikeCount { get; set; }
	public int SaveCount { get; set; }
	public DateTime CreatedAt { get; set; }
	public DateTime? UpdatedAt { get; set; }
	public byte[] RowVersion { get; set; } = default!;

	public ICollection<OutfitItem> Items { get; set; } = new List<OutfitItem>();
}

public class OutfitItem
{
	public Guid OutfitId { get; set; }
	public Outfit Outfit { get; set; } = default!;
	public Guid GarmentId { get; set; }
	public Garment Garment { get; set; } = default!;
	public int? ZIndex { get; set; }
	public decimal? X { get; set; }
	public decimal? Y { get; set; }
	public decimal? Scale { get; set; }
	public decimal? RotationDeg { get; set; }
}

public class Follow
{
    public Guid FollowerId { get; set; }
    public User Follower { get; set; } = default!;
    public Guid FolloweeId { get; set; }
    public User Followee { get; set; } = default!;
    public DateTime CreatedAt { get; set; }
}

public class UserOutfitReaction
{
    public Guid UserId { get; set; }
    public Guid OutfitId { get; set; }
    public string Type { get; set; } = default!; // like|save
    public DateTime CreatedAt { get; set; }
}

// Tech / moderation / audit
public class SearchIndexState
{
    [MaxLength(16)] public string EntityType { get; set; } = default!; // garment|outfit
    public Guid EntityId { get; set; }
    [MaxLength(64)] public string IndexName { get; set; } = default!;
    public int IndexVersion { get; set; } = 1;
    [MaxLength(16)] public string VectorStatus { get; set; } = "pending";
    public DateTime? LastIndexedAt { get; set; }
    public string? ErrorMessage { get; set; }
}
public class ContentLabel
{
    public long Id { get; set; }
    public string TargetType { get; set; } = default!;
    public Guid TargetId { get; set; }
    public string Label { get; set; } = default!;
    public decimal Score { get; set; }
    public DateTime CreatedAt { get; set; }
}
public class AuditLog
{
    public long Id { get; set; }
    public string EntityType { get; set; } = default!;
    public Guid EntityId { get; set; }
    public string Action { get; set; } = default!;
    public Guid? ActorId { get; set; }
    public string? Payload { get; set; }
    public DateTime CreatedAt { get; set; }
}

// Этап 3
public class Listing
{
    public Guid Id { get; set; }
    public Guid OwnerId { get; set; }
    public User Owner { get; set; } = default!;
    public Guid? GarmentId { get; set; }
    public Garment? Garment { get; set; }
    public Guid? OutfitId { get; set; }
    public Outfit? Outfit { get; set; }

    public string Type { get; set; } = "gift"; // gift|swap|sell
    public string? Note { get; set; }
    public string Status { get; set; } = "active"; // active|paused|closed
    public int? PriceMinor { get; set; } // только для sell (деньги вне платформы)
    public string? CurrencyCode { get; set; } = "EUR";
    public DateTime CreatedAt { get; set; }
    public DateTime? UpdatedAt { get; set; }
}

public class Interest
{
    public Guid Id { get; set; }
    public Guid ListingId { get; set; }
    public Listing Listing { get; set; } = default!;
    public Guid InitiatorId { get; set; }
    public User Initiator { get; set; } = default!;
    public Guid OwnerId { get; set; }        // денормализация владельца листинга
    public User Owner { get; set; } = default!;

    public string Kind { get; set; } = "inquiry"; // inquiry|offer|swap_offer
    public int? OfferPriceMinor { get; set; }
    public string? OfferCurrency { get; set; } = "EUR";
    public string Status { get; set; } = "open";  // open|accepted|declined|closed
    public DateTime CreatedAt { get; set; }
    public DateTime? UpdatedAt { get; set; }
}

public class InterestMessage
{
    public Guid Id { get; set; }
    public Guid InterestId { get; set; }
    public Interest Interest { get; set; } = default!;
    public Guid SenderId { get; set; }
    public User Sender { get; set; } = default!;
    public string Body { get; set; } = default!;
    public DateTime CreatedAt { get; set; }
    public DateTime? ReadAt { get; set; }
}

public class SwapItem
{
    public Guid InterestId { get; set; }
    public Interest Interest { get; set; } = default!;
    public Guid GarmentId { get; set; }
    public Garment Garment { get; set; } = default!;
    public string Role { get; set; } = "offer"; // offer|request
}
```

```
public class PeerReview
{
	public Guid Id { get; set; }
	public Guid InterestId { get; set; }
	public Guid ReviewerId { get; set; }
	public Guid RevieweeId { get; set; }
	public byte Rating { get; set; } // 1..5
	public string? Comment { get; set; }
	public DateTime CreatedAt { get; set; }
}
```