Entity Frame Work(scafold-dbcontex)

1.Create database

2.Create table

3.create store procedures:

USE RSA\_GroceryDEV;

SELECT TOP 10 \* from BasketData;

SELECT TOP 10 BasketDataId,Retailer,POSId,OperatorId,TransactionId,TransactionDate,TransactionTotalAmount,TransactionTaxAmount,TransactionTenderType,CreatedDate from BasketData;

--- get proc data from BasketData

create proc usp\_get\_basketData

as

begin

SELECT TOP 10 BasketDataId,Retailer,POSId,OperatorId,TransactionId,TransactionDate,TransactionTotalAmount,TransactionTaxAmount,TransactionTenderType,CreatedDate from BasketData;

end

exec usp\_get\_basketData;

4.Create Models folder in your project

namespace BasketDemo.Models

{

} public class BasketData

{

public int BasketDataId { get; set; }

public string Retailer { get; set; }

public int POSId { get; set; }

public int StoreId { get; set; }

public int OperatorId { get; set; }

public int TransactionId { get; set; }

public DateTime TransactionDate { get; set; }

public DateTime TransactionTime { get; set; }

public int TransactionTotalAmount { get; set; }

public int TransactionTaxAmount { get; set; }

public string TransactionTenderType { get; set; }

public DateTime CreatedDate { get; set; }

}

using System;

using System.Collections.Generic;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata;

namespace BasketProject.Models

{

public partial class RSA\_GroceryDEVContext : DbContext

{

public RSA\_GroceryDEVContext()

{

}

public RSA\_GroceryDEVContext(DbContextOptions<RSA\_GroceryDEVContext> options)

: base(options)

{

}

public virtual DbSet<Address> Addresses { get; set; } = null!;

public virtual DbSet<AddressType> AddressTypes { get; set; } = null!;

public virtual DbSet<AspnetApplication> AspnetApplications { get; set; } = null!;

public virtual DbSet<AspnetMembership> AspnetMemberships { get; set; } = null!;

public virtual DbSet<AspnetPath> AspnetPaths { get; set; } = null!;

public virtual DbSet<AspnetPersonalizationAllUser> AspnetPersonalizationAllUsers { get; set; } = null!;

public virtual DbSet<AspnetPersonalizationPerUser> AspnetPersonalizationPerUsers { get; set; } = null!;

public virtual DbSet<AspnetProfile> AspnetProfiles { get; set; } = null!;

public virtual DbSet<AspnetRole> AspnetRoles { get; set; } = null!;

public virtual DbSet<AspnetSchemaVersion> AspnetSchemaVersions { get; set; } = null!;

public virtual DbSet<AspnetUser> AspnetUsers { get; set; } = null!;

public virtual DbSet<AspnetUsersInRole> AspnetUsersInRoles { get; set; } = null!;

public virtual DbSet<AspnetWebEventEvent> AspnetWebEventEvents { get; set; } = null!;

public virtual DbSet<BasketConsumerId> BasketConsumerIds { get; set; } = null!;

public virtual DbSet<BasketCoupon> BasketCoupons { get; set; } = null!;

public virtual DbSet<BasketCouponUpc> BasketCouponUpcs { get; set; } = null!;

public virtual DbSet<BasketDataJson> BasketDataJsons { get; set; } = null!;

public virtual DbSet<BasketDatum> BasketData { get; set; } = null!;

public virtual DbSet<BasketItem> BasketItems { get; set; } = null!;

public virtual DbSet<BiSpendByUpcAndMember> BiSpendByUpcAndMembers { get; set; } = null!;

public virtual DbSet<BiSpendByUpcAndStore> BiSpendByUpcAndStores { get; set; } = null!;

public virtual DbSet<BiSpendTotalByMember> BiSpendTotalByMembers { get; set; } = null!;

public virtual DbSet<BuyingUnit> BuyingUnits { get; set; } = null!;

public virtual DbSet<Circular> Circulars { get; set; } = null!;

public virtual DbSet<ClientEnterprise> ClientEnterprises { get; set; } = null!;

public virtual DbSet<ClientEnterpriseRoute> ClientEnterpriseRoutes { get; set; } = null!;

public virtual DbSet<ClientJobDetail> ClientJobDetails { get; set; } = null!;

public virtual DbSet<ClientMessage> ClientMessages { get; set; } = null!;

public virtual DbSet<ClientMessageTarget> ClientMessageTargets { get; set; } = null!;

public virtual DbSet<ClientMessagesDeleted> ClientMessagesDeleteds { get; set; } = null!;

public virtual DbSet<ClientStore> ClientStores { get; set; } = null!;

public virtual DbSet<ClientStoreGroup> ClientStoreGroups { get; set; } = null!;

public virtual DbSet<Club> Clubs { get; set; } = null!;

public virtual DbSet<ClubUser> ClubUsers { get; set; } = null!;

public virtual DbSet<ClubsDeleted> ClubsDeleteds { get; set; } = null!;

public virtual DbSet<Company> Companies { get; set; } = null!;

public virtual DbSet<Company1> Companies1 { get; set; } = null!;

public virtual DbSet<CompanyAddress> CompanyAddresses { get; set; } = null!;

public virtual DbSet<CompanyContact> CompanyContacts { get; set; } = null!;

public virtual DbSet<CompanyUser> CompanyUsers { get; set; } = null!;

public virtual DbSet<Contact> Contacts { get; set; } = null!;

public virtual DbSet<ContactMessage> ContactMessages { get; set; } = null!;

public virtual DbSet<ContactType> ContactTypes { get; set; } = null!;

public virtual DbSet<ContactU> ContactUs { get; set; } = null!;

public virtual DbSet<ContentAppId> ContentAppIds { get; set; } = null!;

public virtual DbSet<Country> Countries { get; set; } = null!;

public virtual DbSet<Coupon> Coupons { get; set; } = null!;

public virtual DbSet<CouponRule> CouponRules { get; set; } = null!;

public virtual DbSet<CouponStatus> CouponStatuses { get; set; } = null!;

public virtual DbSet<CouponsOfferNotExist> CouponsOfferNotExists { get; set; } = null!;

public virtual DbSet<CouponsOfferRedemption> CouponsOfferRedemptions { get; set; } = null!;

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

public virtual DbSet<CustomerAddresss> CustomerAddressses { get; set; } = null!;

public virtual DbSet<CustomerComment> CustomerComments { get; set; } = null!;

public virtual DbSet<CustomerContact> CustomerContacts { get; set; } = null!;

public virtual DbSet<CustomerType> CustomerTypes { get; set; } = null!;

public virtual DbSet<CustomerUser> CustomerUsers { get; set; } = null!;

public virtual DbSet<CustomersOld> CustomersOlds { get; set; } = null!;

public virtual DbSet<DataImport> DataImports { get; set; } = null!;

public virtual DbSet<DataImportDetail> DataImportDetails { get; set; } = null!;

public virtual DbSet<DeviceType> DeviceTypes { get; set; } = null!;

public virtual DbSet<DistributorCoupon> DistributorCoupons { get; set; } = null!;

public virtual DbSet<DistributorCouponsOfferRedemption> DistributorCouponsOfferRedemptions { get; set; } = null!;

public virtual DbSet<DistributorCouponsUpdateQueue> DistributorCouponsUpdateQueues { get; set; } = null!;

public virtual DbSet<Exception> Exceptions { get; set; } = null!;

public virtual DbSet<ExistingOffer> ExistingOffers { get; set; } = null!;

public virtual DbSet<Group> Groups { get; set; } = null!;

public virtual DbSet<GroupPage> GroupPages { get; set; } = null!;

public virtual DbSet<IntegrationSetting> IntegrationSettings { get; set; } = null!;

public virtual DbSet<Invoice> Invoices { get; set; } = null!;

public virtual DbSet<InvoiceCoupon> InvoiceCoupons { get; set; } = null!;

public virtual DbSet<InvoiceDetail> InvoiceDetails { get; set; } = null!;

public virtual DbSet<JobIntervalType> JobIntervalTypes { get; set; } = null!;

public virtual DbSet<LmJobHistory> LmJobHistories { get; set; } = null!;

public virtual DbSet<LmReward> LmRewards { get; set; } = null!;

public virtual DbSet<LmRewardDepartment> LmRewardDepartments { get; set; } = null!;

public virtual DbSet<LmRewardGroup> LmRewardGroups { get; set; } = null!;

public virtual DbSet<LmRewardProduct> LmRewardProducts { get; set; } = null!;

public virtual DbSet<LmRewardProductType> LmRewardProductTypes { get; set; } = null!;

public virtual DbSet<LmRewardTarget> LmRewardTargets { get; set; } = null!;

public virtual DbSet<LmRewardTransaction> LmRewardTransactions { get; set; } = null!;

public virtual DbSet<LmRewardType> LmRewardTypes { get; set; } = null!;

public virtual DbSet<LmRewardUpdateQueue> LmRewardUpdateQueues { get; set; } = null!;

public virtual DbSet<LmUserReward> LmUserRewards { get; set; } = null!;

public virtual DbSet<MfgCoupon> MfgCoupons { get; set; } = null!;

public virtual DbSet<MfgCouponsHold> MfgCouponsHolds { get; set; } = null!;

public virtual DbSet<MfgCouponsHoldStore> MfgCouponsHoldStores { get; set; } = null!;

public virtual DbSet<MfgCouponsUpdateQueue> MfgCouponsUpdateQueues { get; set; } = null!;

public virtual DbSet<MyCart> MyCarts { get; set; } = null!;

public virtual DbSet<MyCartProduct> MyCartProducts { get; set; } = null!;

public virtual DbSet<NewsCategory> NewsCategories { get; set; } = null!;

public virtual DbSet<NewsItem> NewsItems { get; set; } = null!;

public virtual DbSet<NewsItemCategory> NewsItemCategories { get; set; } = null!;

public virtual DbSet<NewsTarget> NewsTargets { get; set; } = null!;

public virtual DbSet<Page> Pages { get; set; } = null!;

public virtual DbSet<PageItem> PageItems { get; set; } = null!;

public virtual DbSet<PaymentTerm> PaymentTerms { get; set; } = null!;

public virtual DbSet<Possoftware> Possoftwares { get; set; } = null!;

public virtual DbSet<Posvendor> Posvendors { get; set; } = null!;

public virtual DbSet<Product> Products { get; set; } = null!;

public virtual DbSet<Product1> Products1 { get; set; } = null!;

public virtual DbSet<ProductCategory> ProductCategories { get; set; } = null!;

public virtual DbSet<ProductImport> ProductImports { get; set; } = null!;

public virtual DbSet<ProductMajorCategory> ProductMajorCategories { get; set; } = null!;

public virtual DbSet<ProductsTest> ProductsTests { get; set; } = null!;

public virtual DbSet<Promotion> Promotions { get; set; } = null!;

public virtual DbSet<PushNotificationsQueue> PushNotificationsQueues { get; set; } = null!;

public virtual DbSet<QualifyingProduct> QualifyingProducts { get; set; } = null!;

public virtual DbSet<Redemption> Redemptions { get; set; } = null!;

public virtual DbSet<RedemptionDataJson> RedemptionDataJsons { get; set; } = null!;

public virtual DbSet<RedemptionDistribution> RedemptionDistributions { get; set; } = null!;

public virtual DbSet<RedemptionDistributionJson> RedemptionDistributionJsons { get; set; } = null!;

public virtual DbSet<RsaTransactionHistory> RsaTransactionHistories { get; set; } = null!;

public virtual DbSet<SalesHeader> SalesHeaders { get; set; } = null!;

public virtual DbSet<SalesHeaderDetail> SalesHeaderDetails { get; set; } = null!;

public virtual DbSet<SalesQualifyingProduct> SalesQualifyingProducts { get; set; } = null!;

public virtual DbSet<SalesRedeem> SalesRedeems { get; set; } = null!;

public virtual DbSet<SalesReward> SalesRewards { get; set; } = null!;

public virtual DbSet<SelectedProductIdForWeeklySpecial> SelectedProductIdForWeeklySpecials { get; set; } = null!;

public virtual DbSet<SelectedStoresForCoupon> SelectedStoresForCoupons { get; set; } = null!;

public virtual DbSet<SelectedStoresForIntelligentCoupon> SelectedStoresForIntelligentCoupons { get; set; } = null!;

public virtual DbSet<SellingUnit> SellingUnits { get; set; } = null!;

public virtual DbSet<Ssnews> Ssnews { get; set; } = null!;

public virtual DbSet<SsnewsDepartment> SsnewsDepartments { get; set; } = null!;

public virtual DbSet<SsnewsNcrpromotion> SsnewsNcrpromotions { get; set; } = null!;

public virtual DbSet<SsnewsProduct> SsnewsProducts { get; set; } = null!;

public virtual DbSet<SsnewsProductsDeleted> SsnewsProductsDeleteds { get; set; } = null!;

public virtual DbSet<SsnewsRecurring> SsnewsRecurrings { get; set; } = null!;

public virtual DbSet<SsnewsUserFavoritesDetail> SsnewsUserFavoritesDetails { get; set; } = null!;

public virtual DbSet<SsnewsUserFavoritesList> SsnewsUserFavoritesLists { get; set; } = null!;

public virtual DbSet<SsnewsUsersNcrimpression> SsnewsUsersNcrimpressions { get; set; } = null!;

public virtual DbSet<SsnewsUsersRedeem> SsnewsUsersRedeems { get; set; } = null!;

public virtual DbSet<State> States { get; set; } = null!;

public virtual DbSet<StoreCouponsUpdatequeue> StoreCouponsUpdatequeues { get; set; } = null!;

public virtual DbSet<SubCategory> SubCategories { get; set; } = null!;

public virtual DbSet<TimeZone> TimeZones { get; set; } = null!;

public virtual DbSet<UserAdminStore> UserAdminStores { get; set; } = null!;

public virtual DbSet<UserDetail> UserDetails { get; set; } = null!;

public virtual DbSet<UserDevice> UserDevices { get; set; } = null!;

public virtual DbSet<UserDeviceInfo> UserDeviceInfos { get; set; } = null!;

public virtual DbSet<UserProductUsage> UserProductUsages { get; set; } = null!;

public virtual DbSet<UserProfile> UserProfiles { get; set; } = null!;

public virtual DbSet<UserStatus> UserStatuses { get; set; } = null!;

public virtual DbSet<UserTermsAcceptance> UserTermsAcceptances { get; set; } = null!;

public virtual DbSet<UserToken> UserTokens { get; set; } = null!;

public virtual DbSet<UserType> UserTypes { get; set; } = null!;

public virtual DbSet<UsersCouponsRedeem> UsersCouponsRedeems { get; set; } = null!;

public virtual DbSet<ViewCustomGetClubUser> ViewCustomGetClubUsers { get; set; } = null!;

public virtual DbSet<ViewCustomUserDetail> ViewCustomUserDetails { get; set; } = null!;

public virtual DbSet<VwAspnetApplication> VwAspnetApplications { get; set; } = null!;

public virtual DbSet<VwAspnetMembershipUser> VwAspnetMembershipUsers { get; set; } = null!;

public virtual DbSet<VwAspnetProfile> VwAspnetProfiles { get; set; } = null!;

public virtual DbSet<VwAspnetRole> VwAspnetRoles { get; set; } = null!;

public virtual DbSet<VwAspnetUser> VwAspnetUsers { get; set; } = null!;

public virtual DbSet<VwAspnetUsersInRole> VwAspnetUsersInRoles { get; set; } = null!;

public virtual DbSet<VwAspnetWebPartStatePath> VwAspnetWebPartStatePaths { get; set; } = null!;

public virtual DbSet<VwAspnetWebPartStateShared> VwAspnetWebPartStateShareds { get; set; } = null!;

public virtual DbSet<VwAspnetWebPartStateUser> VwAspnetWebPartStateUsers { get; set; } = null!;

public virtual DbSet<VwBasketCouponItem> VwBasketCouponItems { get; set; } = null!;

public virtual DbSet<VwRedemptionBasketItem> VwRedemptionBasketItems { get; set; } = null!;

public virtual DbSet<VwTestView> VwTestViews { get; set; } = null!;

public virtual DbSet<WebDepartment> WebDepartments { get; set; } = null!;

public virtual DbSet<WebMenu> WebMenus { get; set; } = null!;

public virtual DbSet<WebpagesMembership> WebpagesMemberships { get; set; } = null!;

public virtual DbSet<WebpagesOauthMembership> WebpagesOauthMemberships { get; set; } = null!;

public virtual DbSet<WebpagesRole> WebpagesRoles { get; set; } = null!;

public virtual DbSet<WeeklyAddPdf> WeeklyAddPdfs { get; set; } = null!;

public virtual DbSet<WeeklySpecialImport> WeeklySpecialImports { get; set; } = null!;

public virtual DbSet<WeeklySpecialsExtension> WeeklySpecialsExtensions { get; set; } = null!;

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.

optionsBuilder.UseSqlServer("Server=veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com;Initial Catalog=RSA\_GroceryDEV;MultipleActiveResultSets=true;User ID=Veritragrocerydblogin;Password=otTL5kd0buqb7erEuTqX");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Address>(entity =>

{

entity.ToTable("Address");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.AddressLine1)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.AddressLine2)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.City)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("date");

entity.Property(e => e.ModifiedDate).HasColumnType("date");

entity.Property(e => e.Zip)

.HasMaxLength(6)

.IsUnicode(false)

.HasColumnName("ZIP");

entity.HasOne(d => d.AddressType)

.WithMany(p => p.Addresses)

.HasForeignKey(d => d.AddressTypeId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_Address\_AddressType");

});

modelBuilder.Entity<AddressType>(entity =>

{

entity.ToTable("AddressType");

entity.Property(e => e.AddressType1)

.HasMaxLength(25)

.IsUnicode(false)

.HasColumnName("AddressType");

entity.Property(e => e.CreatedDate).HasColumnType("date");

entity.Property(e => e.ModifiedDate).HasColumnType("date");

});

modelBuilder.Entity<AspnetApplication>(entity =>

{

entity.HasKey(e => e.ApplicationId)

.HasName("PK\_\_aspnet\_A\_\_C93A4C982F785EEC")

.IsClustered(false);

entity.ToTable("aspnet\_Applications");

entity.HasIndex(e => e.LoweredApplicationName, "UQ\_\_aspnet\_A\_\_17477DE4BF736F4D")

.IsUnique();

entity.HasIndex(e => e.ApplicationName, "UQ\_\_aspnet\_A\_\_3091033198E50C55")

.IsUnique();

entity.HasIndex(e => e.LoweredApplicationName, "aspnet\_Applications\_Index")

.IsClustered();

entity.Property(e => e.ApplicationId).HasDefaultValueSql("(newid())");

entity.Property(e => e.ApplicationName).HasMaxLength(256);

entity.Property(e => e.Description).HasMaxLength(256);

entity.Property(e => e.LoweredApplicationName).HasMaxLength(256);

});

modelBuilder.Entity<AspnetMembership>(entity =>

{

entity.HasKey(e => e.UserId)

.HasName("PK\_\_aspnet\_M\_\_1788CC4D9FC5CCD3")

.IsClustered(false);

entity.ToTable("aspnet\_Membership");

entity.HasIndex(e => new { e.ApplicationId, e.LoweredEmail }, "aspnet\_Membership\_index")

.IsClustered();

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

entity.Property(e => e.Comment).HasColumnType("ntext");

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.Email).HasMaxLength(256);

entity.Property(e => e.FailedPasswordAnswerAttemptWindowStart).HasColumnType("datetime");

entity.Property(e => e.FailedPasswordAttemptWindowStart).HasColumnType("datetime");

entity.Property(e => e.LastLockoutDate).HasColumnType("datetime");

entity.Property(e => e.LastLoginDate).HasColumnType("datetime");

entity.Property(e => e.LastPasswordChangedDate).HasColumnType("datetime");

entity.Property(e => e.LoweredEmail).HasMaxLength(256);

entity.Property(e => e.MobilePin)

.HasMaxLength(16)

.HasColumnName("MobilePIN");

entity.Property(e => e.Password).HasMaxLength(128);

entity.Property(e => e.PasswordAnswer).HasMaxLength(128);

entity.Property(e => e.PasswordQuestion).HasMaxLength(256);

entity.Property(e => e.PasswordSalt).HasMaxLength(128);

entity.HasOne(d => d.Application)

.WithMany(p => p.AspnetMemberships)

.HasForeignKey(d => d.ApplicationId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Me\_\_Appli\_\_1F83A428");

entity.HasOne(d => d.User)

.WithOne(p => p.AspnetMembership)

.HasForeignKey<AspnetMembership>(d => d.UserId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Me\_\_UserI\_\_2077C861");

});

modelBuilder.Entity<AspnetPath>(entity =>

{

entity.HasKey(e => e.PathId)

.HasName("PK\_\_aspnet\_P\_\_CD67DC58F77C9FB8")

.IsClustered(false);

entity.ToTable("aspnet\_Paths");

entity.HasIndex(e => new { e.ApplicationId, e.LoweredPath }, "aspnet\_Paths\_index")

.IsUnique()

.IsClustered();

entity.Property(e => e.PathId).HasDefaultValueSql("(newid())");

entity.Property(e => e.LoweredPath).HasMaxLength(256);

entity.Property(e => e.Path).HasMaxLength(256);

entity.HasOne(d => d.Application)

.WithMany(p => p.AspnetPaths)

.HasForeignKey(d => d.ApplicationId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Pa\_\_Appli\_\_216BEC9A");

});

modelBuilder.Entity<AspnetPersonalizationAllUser>(entity =>

{

entity.HasKey(e => e.PathId)

.HasName("PK\_\_aspnet\_P\_\_CD67DC5956E39570");

entity.ToTable("aspnet\_PersonalizationAllUsers");

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

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

entity.Property(e => e.PageSettings).HasColumnType("image");

entity.HasOne(d => d.Path)

.WithOne(p => p.AspnetPersonalizationAllUser)

.HasForeignKey<AspnetPersonalizationAllUser>(d => d.PathId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Pe\_\_PathI\_\_226010D3");

});

modelBuilder.Entity<AspnetPersonalizationPerUser>(entity =>

{

entity.HasKey(e => e.Id)

.HasName("PK\_\_aspnet\_P\_\_3214EC063A762A5C")

.IsClustered(false);

entity.ToTable("aspnet\_PersonalizationPerUser");

entity.HasIndex(e => new { e.PathId, e.UserId }, "aspnet\_PersonalizationPerUser\_index1")

.IsUnique()

.IsClustered();

entity.HasIndex(e => new { e.UserId, e.PathId }, "aspnet\_PersonalizationPerUser\_ncindex2")

.IsUnique();

entity.Property(e => e.Id).HasDefaultValueSql("(newid())");

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

entity.Property(e => e.PageSettings).HasColumnType("image");

entity.HasOne(d => d.Path)

.WithMany(p => p.AspnetPersonalizationPerUsers)

.HasForeignKey(d => d.PathId)

.HasConstraintName("FK\_\_aspnet\_Pe\_\_PathI\_\_2354350C");

entity.HasOne(d => d.User)

.WithMany(p => p.AspnetPersonalizationPerUsers)

.HasForeignKey(d => d.UserId)

.HasConstraintName("FK\_\_aspnet\_Pe\_\_UserI\_\_24485945");

});

modelBuilder.Entity<AspnetProfile>(entity =>

{

entity.HasKey(e => e.UserId)

.HasName("PK\_\_aspnet\_P\_\_1788CC4CA2F6C826");

entity.ToTable("aspnet\_Profile");

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

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

entity.Property(e => e.PropertyNames).HasColumnType("ntext");

entity.Property(e => e.PropertyValuesBinary).HasColumnType("image");

entity.Property(e => e.PropertyValuesString).HasColumnType("ntext");

entity.HasOne(d => d.User)

.WithOne(p => p.AspnetProfile)

.HasForeignKey<AspnetProfile>(d => d.UserId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Pr\_\_UserI\_\_253C7D7E");

});

modelBuilder.Entity<AspnetRole>(entity =>

{

entity.HasKey(e => e.RoleId)

.HasName("PK\_\_aspnet\_R\_\_8AFACE1BC46C69C2")

.IsClustered(false);

entity.ToTable("aspnet\_Roles");

entity.HasIndex(e => new { e.ApplicationId, e.LoweredRoleName }, "aspnet\_Roles\_index1")

.IsUnique()

.IsClustered();

entity.Property(e => e.RoleId).HasDefaultValueSql("(newid())");

entity.Property(e => e.Description).HasMaxLength(256);

entity.Property(e => e.LoweredRoleName).HasMaxLength(256);

entity.Property(e => e.RoleName).HasMaxLength(256);

entity.HasOne(d => d.Application)

.WithMany(p => p.AspnetRoles)

.HasForeignKey(d => d.ApplicationId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Ro\_\_Appli\_\_2630A1B7");

});

modelBuilder.Entity<AspnetSchemaVersion>(entity =>

{

entity.HasKey(e => new { e.Feature, e.CompatibleSchemaVersion })

.HasName("PK\_\_aspnet\_S\_\_5A1E6BC16D82BDB9");

entity.ToTable("aspnet\_SchemaVersions");

entity.Property(e => e.Feature).HasMaxLength(128);

entity.Property(e => e.CompatibleSchemaVersion).HasMaxLength(128);

});

modelBuilder.Entity<AspnetUser>(entity =>

{

entity.HasKey(e => e.UserId)

.HasName("PK\_\_aspnet\_U\_\_1788CC4D418BEB9F")

.IsClustered(false);

entity.ToTable("aspnet\_Users");

entity.HasIndex(e => new { e.ApplicationId, e.LoweredUserName }, "aspnet\_Users\_Index")

.IsUnique()

.IsClustered();

entity.HasIndex(e => new { e.ApplicationId, e.LastActivityDate }, "aspnet\_Users\_Index2");

entity.Property(e => e.UserId).HasDefaultValueSql("(newid())");

entity.Property(e => e.LastActivityDate).HasColumnType("datetime");

entity.Property(e => e.LoweredUserName).HasMaxLength(256);

entity.Property(e => e.MobileAlias).HasMaxLength(16);

entity.Property(e => e.UserName).HasMaxLength(256);

entity.HasOne(d => d.Application)

.WithMany(p => p.AspnetUsers)

.HasForeignKey(d => d.ApplicationId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Us\_\_Appli\_\_2724C5F0");

});

modelBuilder.Entity<AspnetUsersInRole>(entity =>

{

entity.HasKey(e => e.UserId);

entity.ToTable("aspnet\_UsersInRoles");

entity.HasIndex(e => e.RoleId, "aspnet\_UsersInRoles\_index");

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

entity.HasOne(d => d.Role)

.WithMany(p => p.AspnetUsersInRoles)

.HasForeignKey(d => d.RoleId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Us\_\_RoleI\_\_2818EA29");

entity.HasOne(d => d.User)

.WithOne(p => p.AspnetUsersInRole)

.HasForeignKey<AspnetUsersInRole>(d => d.UserId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_\_aspnet\_Us\_\_UserI\_\_290D0E62");

});

modelBuilder.Entity<AspnetWebEventEvent>(entity =>

{

entity.HasKey(e => e.EventId)

.HasName("PK\_\_aspnet\_W\_\_7944C8108D88691E");

entity.ToTable("aspnet\_WebEvent\_Events");

entity.Property(e => e.EventId)

.HasMaxLength(32)

.IsUnicode(false)

.IsFixedLength();

entity.Property(e => e.ApplicationPath).HasMaxLength(256);

entity.Property(e => e.ApplicationVirtualPath).HasMaxLength(256);

entity.Property(e => e.Details).HasColumnType("ntext");

entity.Property(e => e.EventOccurrence).HasColumnType("decimal(19, 0)");

entity.Property(e => e.EventSequence).HasColumnType("decimal(19, 0)");

entity.Property(e => e.EventTime).HasColumnType("datetime");

entity.Property(e => e.EventTimeUtc).HasColumnType("datetime");

entity.Property(e => e.EventType).HasMaxLength(256);

entity.Property(e => e.ExceptionType).HasMaxLength(256);

entity.Property(e => e.MachineName).HasMaxLength(256);

entity.Property(e => e.Message).HasMaxLength(1024);

entity.Property(e => e.RequestUrl).HasMaxLength(1024);

});

modelBuilder.Entity<BasketConsumerId>(entity =>

{

entity.HasKey(e => e.BasketConsumerId1);

entity.HasIndex(e => e.BasketDataId, "idx\_BasketDataID");

entity.HasIndex(e => e.LoyaltyId, "idx\_LoyaltyID");

entity.Property(e => e.BasketConsumerId1).HasColumnName("BasketConsumerId");

entity.Property(e => e.AlternateId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.EmailAddress)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.LoyaltyId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.MobileNumber)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<BasketCoupon>(entity =>

{

entity.HasKey(e => e.BasketCouponId)

.HasName("PK\_BasketCoupons\_1");

entity.HasIndex(e => e.BasketDataId, "idx\_BasketDataID");

entity.HasIndex(e => e.Id, "idx\_ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Id)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.IdType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Source)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Type)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<BasketCouponUpc>(entity =>

{

entity.ToTable("BasketCouponUPC");

entity.Property(e => e.BasketCouponUpcid).HasColumnName("BasketCouponUPCID");

entity.Property(e => e.BasketCouponId).HasColumnName("BasketCouponID");

entity.Property(e => e.BasketCouponPromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("BasketCouponPromotionID");

entity.Property(e => e.BasketDataId).HasColumnName("BasketDataID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Upc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<BasketDataJson>(entity =>

{

entity.ToTable("BasketData\_JSON");

entity.Property(e => e.BasketDataJsonid).HasColumnName("BasketDataJSONID");

entity.Property(e => e.BasketDataId).HasColumnName("BasketDataID");

entity.Property(e => e.BasketJson)

.HasColumnType("text")

.HasColumnName("BasketJSON");

});

modelBuilder.Entity<BasketDatum>(entity =>

{

entity.HasKey(e => e.BasketDataId);

entity.HasIndex(e => e.StoreId, "idx\_StoreID");

entity.HasIndex(e => e.TransactionDate, "idx\_TransactionDate");

entity.Property(e => e.BasketGuid)

.HasColumnName("BasketGUID")

.HasDefaultValueSql("(newid())");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.IsProcessed).HasDefaultValueSql("((0))");

entity.Property(e => e.IsProcessedForMfg).HasDefaultValueSql("((0))");

entity.Property(e => e.IsUpcsplitComplete)

.HasColumnName("IsUPCSplitComplete")

.HasDefaultValueSql("((1))");

entity.Property(e => e.Posid).HasColumnName("POSId");

entity.Property(e => e.Retailer)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.TransactionDate).HasColumnType("date");

entity.Property(e => e.TransactionTenderType)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<BasketItem>(entity =>

{

entity.HasIndex(e => e.BasketDataId, "idx\_BasketDataID");

entity.HasIndex(e => e.Upc, "idx\_UPC");

entity.Property(e => e.CoPrefix)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.FamilyCode1)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.FamilyCode2)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.IdType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.QtyType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SaleType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<BiSpendByUpcAndMember>(entity =>

{

entity.HasKey(e => e.SpendByUpcMemberId);

entity.ToTable("BI\_SPEND\_BY\_UPC\_AND\_MEMBER");

entity.HasIndex(e => e.Upc, "idx\_BI\_SPEND\_BY\_UPC\_AND\_MEMBER");

entity.Property(e => e.SpendByUpcMemberId).HasColumnName("SpendByUpcMemberID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.MemberNumber)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.TotalAmount).HasColumnType("money");

entity.Property(e => e.TransactionDate).HasColumnType("date");

entity.Property(e => e.Upc)

.HasMaxLength(20)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<BiSpendByUpcAndStore>(entity =>

{

entity.HasKey(e => e.SpendByUpcStoreId)

.HasName("PK\_BI\_SPEND\_BY\_SpendByUPCStoreID");

entity.ToTable("BI\_SPEND\_BY\_UPC\_AND\_STORE");

entity.HasIndex(e => e.Upc, "idx\_BI\_SPEND\_BY\_UPC\_AND\_STORE\_UPC");

entity.HasIndex(e => e.StoreId, "idx\_BI\_SPEND\_StoreID");

entity.HasIndex(e => e.DeptId, "idx\_Spend\_DeptID");

entity.Property(e => e.SpendByUpcStoreId).HasColumnName("SpendByUpcStoreID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.TotalAmount).HasColumnType("money");

entity.Property(e => e.TransactionDate).HasColumnType("date");

entity.Property(e => e.Upc)

.HasMaxLength(20)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<BiSpendTotalByMember>(entity =>

{

entity.HasKey(e => e.SpendTotalByMemberId)

.HasName("PK\_SpendTotalByMeberID");

entity.ToTable("BI\_SPEND\_TOTAL\_BY\_MEMBER");

entity.Property(e => e.SpendTotalByMemberId).HasColumnName("SpendTotalByMemberID");

entity.Property(e => e.BasketAmount).HasColumnType("money");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.MemberNumber)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.TransactionDate).HasColumnType("date");

});

modelBuilder.Entity<BuyingUnit>(entity =>

{

entity.Property(e => e.BuyingUnits)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<Circular>(entity =>

{

entity.HasKey(e => e.Cid);

entity.Property(e => e.Cid).HasColumnName("CID");

entity.Property(e => e.CircularId).HasColumnName("circular\_id");

entity.Property(e => e.CircularName)

.HasMaxLength(300)

.IsUnicode(false)

.HasColumnName("circular\_name");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<ClientEnterprise>(entity =>

{

entity.HasKey(e => e.ClientEnterprisesId);

entity.Property(e => e.ClientEnterprisesId).HasColumnName("ClientEnterprisesID");

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PosenterpriseId)

.HasMaxLength(200)

.IsUnicode(false)

.HasColumnName("POSEnterpriseId");

entity.Property(e => e.PosenterpriseSecretKey)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("POSEnterpriseSecretKey");

entity.Property(e => e.Possoftware)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("POSSoftware");

entity.Property(e => e.PosvendorId).HasColumnName("POSVendorId");

});

modelBuilder.Entity<ClientEnterpriseRoute>(entity =>

{

entity.Property(e => e.ClientEnterpriseRouteId).HasColumnName("ClientEnterpriseRouteID");

entity.Property(e => e.ClientEnterprisesId).HasColumnName("ClientEnterprisesID");

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PosenterpriseId)

.HasMaxLength(250)

.IsUnicode(false)

.HasColumnName("POSEnterpriseId");

entity.Property(e => e.PosrouteId)

.HasMaxLength(200)

.IsUnicode(false)

.HasColumnName("POSRouteID");

entity.HasOne(d => d.ClientEnterprises)

.WithMany(p => p.ClientEnterpriseRoutes)

.HasForeignKey(d => d.ClientEnterprisesId)

.HasConstraintName("FK\_ClientEnterpriseRoutes\_ClientEnterprises");

});

modelBuilder.Entity<ClientJobDetail>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.JobStartDate).HasColumnType("datetime");

entity.Property(e => e.LastRunDate).HasColumnType("datetime");

});

modelBuilder.Entity<ClientMessage>(entity =>

{

entity.HasKey(e => e.MessageId);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Details).IsUnicode(false);

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.IsTargetSpecific).HasDefaultValueSql("((0))");

entity.Property(e => e.ScheduledDate).HasColumnType("datetime");

entity.Property(e => e.Title).IsUnicode(false);

});

modelBuilder.Entity<ClientMessageTarget>(entity =>

{

entity.HasKey(e => e.MessageTargetId);

entity.ToTable("ClientMessage\_Targets");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CreatedUserId).HasColumnName("CreatedUserID");

entity.Property(e => e.IsExclude).HasDefaultValueSql("((0))");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedUserId).HasColumnName("ModifiedUserID");

entity.Property(e => e.UserId).HasColumnName("UserID");

entity.HasOne(d => d.Club)

.WithMany(p => p.ClientMessageTargets)

.HasForeignKey(d => d.ClubId)

.HasConstraintName("FK\_ClientMessage\_Targets\_Clubs");

entity.HasOne(d => d.Message)

.WithMany(p => p.ClientMessageTargets)

.HasForeignKey(d => d.MessageId)

.HasConstraintName("FK\_ClientMessage\_Targets\_SSNews");

});

modelBuilder.Entity<ClientMessagesDeleted>(entity =>

{

entity.HasKey(e => e.MessageDeleteId);

entity.ToTable("ClientMessagesDeleted");

entity.Property(e => e.MessageDeleteId).HasColumnName("MessageDeleteID");

entity.Property(e => e.ClientMessageId).HasColumnName("ClientMessageID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.UserId).HasColumnName("UserID");

entity.HasOne(d => d.ClientMessage)

.WithMany(p => p.ClientMessagesDeleteds)

.HasForeignKey(d => d.ClientMessageId)

.HasConstraintName("FK\_ClientMessagesDeleted\_ClientMessages");

});

modelBuilder.Entity<ClientStore>(entity =>

{

entity.Property(e => e.AddressLine1)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.AddressLine2)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.City)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ClientStorePosenterpriseSecretId)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("ClientStorePOSEnterpriseSecretId");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.GlobalStoreId).HasColumnName("GlobalStoreID");

entity.Property(e => e.IsActive).HasDefaultValueSql("((1))");

entity.Property(e => e.Latitude).HasColumnType("decimal(9, 6)");

entity.Property(e => e.Longitude).HasColumnType("decimal(9, 6)");

entity.Property(e => e.OtherPos)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OtherPOS");

entity.Property(e => e.PosrouteId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("POSRouteId");

entity.Property(e => e.PossoftwareId).HasColumnName("POSSoftwareId");

entity.Property(e => e.PossoftwareVersionNumber)

.HasMaxLength(200)

.IsUnicode(false)

.HasColumnName("POSSoftwareVersionNumber");

entity.Property(e => e.PosstoreId)

.HasColumnName("POSStoreId")

.HasDefaultValueSql("((0))");

entity.Property(e => e.Postype).HasColumnName("POStype");

entity.Property(e => e.PosvendorId).HasColumnName("POSVendorId");

entity.Property(e => e.StoreEmail)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.StoreName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.StorePhone)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.StoreProfile).IsUnicode(false);

entity.Property(e => e.StoreTimings)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ZipCode)

.HasMaxLength(20)

.IsUnicode(false);

entity.HasOne(d => d.ClientEnterprises)

.WithMany(p => p.ClientStores)

.HasForeignKey(d => d.ClientEnterprisesId)

.HasConstraintName("FK\_ClientStores\_ClientEnterprises");

});

modelBuilder.Entity<ClientStoreGroup>(entity =>

{

entity.HasNoKey();

entity.HasIndex(e => e.ClientStoreGroupsId, "ci\_azure\_fixup\_dbo\_ClientStoreGroups")

.IsClustered();

entity.Property(e => e.ClientStoreGroupsId).ValueGeneratedOnAdd();

entity.Property(e => e.StoreGroupDescription).IsUnicode(false);

entity.Property(e => e.StoreGroupName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<Club>(entity =>

{

entity.Property(e => e.ClubDetails)

.HasMaxLength(1000)

.IsUnicode(false);

entity.Property(e => e.Createddate)

.HasColumnType("datetime")

.HasColumnName("createddate");

entity.Property(e => e.IsEnableOnSignOn).HasDefaultValueSql("((0))");

entity.Property(e => e.IsMemberIdrequired)

.HasColumnName("IsMemberIDRequired")

.HasDefaultValueSql("((0))");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.Name)

.HasMaxLength(300)

.IsUnicode(false);

entity.Property(e => e.TopicArn)

.HasMaxLength(150)

.IsUnicode(false);

});

modelBuilder.Entity<ClubUser>(entity =>

{

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.ClubMemberId)

.HasMaxLength(300)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.IsSubscribedToTopic).HasDefaultValueSql("((0))");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.HasOne(d => d.Club)

.WithMany(p => p.ClubUsers)

.HasForeignKey(d => d.ClubId)

.HasConstraintName("FK\_ClubUsers\_Clubs1");

});

modelBuilder.Entity<ClubsDeleted>(entity =>

{

entity.ToTable("CLUBS\_DELETED");

entity.Property(e => e.ClubsDeletedId).HasColumnName("ClubsDeletedID");

entity.Property(e => e.ClubId).HasColumnName("ClubID");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

});

modelBuilder.Entity<Company>(entity =>

{

entity.Property(e => e.CompanyEmail)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CompanyLogo)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CompanyName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CompanyPhone)

.HasMaxLength(25)

.IsUnicode(false);

entity.Property(e => e.CompanyProfile)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.CompanyWebSite)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.FederalId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

});

modelBuilder.Entity<Company1>(entity =>

{

entity.HasKey(e => e.CompanyId)

.HasName("PK\_COMPANY");

entity.ToTable("Company");

entity.Property(e => e.BillingAddressId).HasColumnName("Billing\_Address\_ID");

entity.Property(e => e.CompanyMailId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CountryId).HasColumnName("Country\_ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Description)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.FederalId)

.HasMaxLength(10)

.IsUnicode(false)

.HasColumnName("Federal\_ID");

entity.Property(e => e.IsBillingSameAsMailingAddress).HasColumnName("Is\_Billing\_SameAs\_MailingAddress");

entity.Property(e => e.LogoFileUrl)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("LogoFile\_URL");

entity.Property(e => e.MaillingAddressId).HasColumnName("Mailling\_Address\_ID");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PrimaryContactId).HasColumnName("Primary\_Contact\_ID");

entity.Property(e => e.SecondaryContactId).HasColumnName("Secondary\_Contact\_Id");

entity.Property(e => e.WebsiteUrl)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("Website\_URL");

});

modelBuilder.Entity<CompanyAddress>(entity =>

{

entity.HasKey(e => e.RowId);

entity.ToTable("CompanyAddress");

});

modelBuilder.Entity<CompanyContact>(entity =>

{

entity.HasKey(e => e.RowId);

entity.ToTable("CompanyContact");

});

modelBuilder.Entity<CompanyUser>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

});

modelBuilder.Entity<Contact>(entity =>

{

entity.ToTable("Contact");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.ContactTypeId).HasColumnName("ContactTypeID");

entity.Property(e => e.CreatedDate).HasColumnType("date");

entity.Property(e => e.Email)

.HasMaxLength(30)

.IsUnicode(false);

entity.Property(e => e.Fax)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.Phone)

.HasMaxLength(15)

.IsUnicode(false);

entity.HasOne(d => d.ContactType)

.WithMany(p => p.Contacts)

.HasForeignKey(d => d.ContactTypeId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_Contact\_ContactType");

});

modelBuilder.Entity<ContactMessage>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Description).IsUnicode(false);

entity.Property(e => e.Logo)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Title)

.HasMaxLength(200)

.IsUnicode(false);

});

modelBuilder.Entity<ContactType>(entity =>

{

entity.ToTable("ContactType");

entity.Property(e => e.ContactTypeId).HasColumnName("ContactTypeID");

entity.Property(e => e.CreatedDate).HasColumnType("date");

entity.Property(e => e.ModifiedDate).HasColumnType("date");

entity.Property(e => e.TypeName)

.HasMaxLength(15)

.IsUnicode(false);

});

modelBuilder.Entity<ContactU>(entity =>

{

entity.HasKey(e => e.ContactUsId);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Email)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.HowDidYouKnow)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Pos)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("POS");

entity.Property(e => e.Position)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Team)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<ContentAppId>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.AppId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.InstanceName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<Country>(entity =>

{

entity.Property(e => e.CountryName).HasMaxLength(150);

});

modelBuilder.Entity<Coupon>(entity =>

{

entity.Property(e => e.CouponCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DiscountPercentageCount).HasColumnType("money");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.NcrpromotionCode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("NCRPromotionCode");

entity.Property(e => e.NcrpromotionCreatedDate)

.HasColumnType("datetime")

.HasColumnName("NCRPromotionCreatedDate");

entity.Property(e => e.PromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("PromotionID");

entity.Property(e => e.RedeemOn).HasColumnType("datetime");

entity.Property(e => e.UpdatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<CouponRule>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DiscountPercentage).HasColumnType("money");

});

modelBuilder.Entity<CouponStatus>(entity =>

{

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

entity.Property(e => e.CouponStatus1)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("CouponStatus");

});

modelBuilder.Entity<CouponsOfferNotExist>(entity =>

{

entity.HasKey(e => e.MpmId);

entity.ToTable("Coupons\_Offer\_NOT\_EXISTS");

entity.Property(e => e.MpmId).HasColumnName("MPM\_ID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.OfferId).HasColumnName("OfferID");

entity.Property(e => e.OfferIdguid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferIDGUID");

entity.Property(e => e.Retailer)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<CouponsOfferRedemption>(entity =>

{

entity.HasKey(e => e.CouponOfferRedemptionsId);

entity.ToTable("Coupons\_Offer\_Redemptions");

entity.HasIndex(e => e.ClearingStatusId, "idx\_ClearingStatusID");

entity.HasIndex(e => e.CouponOfferId, "idx\_CouponOfferID");

entity.HasIndex(e => e.MemberNumber, "idx\_MemberNumber");

entity.HasIndex(e => e.RedemptionDistributionId, "idx\_RedemptionDistributionId");

entity.Property(e => e.ClearingRecordId).HasColumnName("ClearingRecordID");

entity.Property(e => e.ClearingStatusId)

.HasColumnName("ClearingStatusID")

.HasDefaultValueSql("((1))");

entity.Property(e => e.CouponOfferId).HasColumnName("CouponOfferID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.ErrorCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.MemberNumber)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SentTime).HasColumnType("datetime");

entity.Property(e => e.UpdateTime).HasColumnType("datetime");

});

modelBuilder.Entity<Customer>(entity =>

{

entity.Property(e => e.BillingPercentage).HasColumnType("decimal(18, 2)");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CustomerEmail)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomerGuid).HasColumnName("CustomerGUID");

entity.Property(e => e.CustomerLogo)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomerName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomerPhone)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomerProfile)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.CustomerWebSite)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.FederalId)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.IsPosintegrationEnabled)

.HasColumnName("IsPOSIntegrationEnabled")

.HasDefaultValueSql("((1))");

entity.Property(e => e.MfgcontentAppId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("MFGContentAppId");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.PrimaryColor)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SecondaryColor)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Siccode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("SICCode");

});

modelBuilder.Entity<CustomerAddresss>(entity =>

{

entity.ToTable("CustomerAddresss");

entity.Property(e => e.Id).HasColumnName("ID");

});

modelBuilder.Entity<CustomerComment>(entity =>

{

entity.HasKey(e => e.CommentId)

.HasName("PK\_UserComments");

entity.ToTable("Customer\_Comments");

entity.Property(e => e.CommentDescription).IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<CustomerContact>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.HasOne(d => d.Contact)

.WithMany(p => p.CustomerContacts)

.HasForeignKey(d => d.ContactId)

.HasConstraintName("FK\_CustomerContacts\_Contact");

entity.HasOne(d => d.Customer)

.WithMany(p => p.CustomerContacts)

.HasForeignKey(d => d.CustomerId)

.HasConstraintName("FK\_CustomerContacts\_Customers");

});

modelBuilder.Entity<CustomerType>(entity =>

{

entity.Property(e => e.CustomerDescription)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<CustomerUser>(entity =>

{

entity.HasOne(d => d.ContactType)

.WithMany(p => p.CustomerUsers)

.HasForeignKey(d => d.ContactTypeId)

.HasConstraintName("FK\_CustomerUsers\_ContactType");

});

modelBuilder.Entity<CustomersOld>(entity =>

{

entity.ToTable("CustomersOld");

entity.Property(e => e.DateOfBirth).HasColumnType("date");

entity.Property(e => e.FirstName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Gender)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.LastName)

.HasMaxLength(500)

.IsUnicode(false);

});

modelBuilder.Entity<DataImport>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.FileName)

.HasMaxLength(150)

.IsUnicode(false);

});

modelBuilder.Entity<DataImportDetail>(entity =>

{

entity.Property(e => e.CouponCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CustomerCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductCode)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<DeviceType>(entity =>

{

entity.ToTable("DeviceType");

entity.Property(e => e.DeviceTypeName)

.HasMaxLength(10)

.IsUnicode(false);

});

modelBuilder.Entity<DistributorCoupon>(entity =>

{

entity.ToTable("DISTRIBUTOR\_COUPONS");

entity.Property(e => e.BrandName)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.CouponImage)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.DistCouponsOfferId).HasColumnName("Dist\_Coupons\_Offer\_Id");

entity.Property(e => e.DistributorName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.IsProcessedForNcr).HasColumnName("IsProcessedForNCR");

entity.Property(e => e.OfferActiveDate).HasColumnType("datetime");

entity.Property(e => e.OfferDescription).IsUnicode(false);

entity.Property(e => e.OfferDisclaimer).HasColumnType("text");

entity.Property(e => e.OfferExpiryDate).HasColumnType("datetime");

entity.Property(e => e.OfferFeturedText).IsUnicode(false);

entity.Property(e => e.OfferFinePrint).IsUnicode(false);

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferGUID");

entity.Property(e => e.OfferShutoffDate).HasColumnType("datetime");

entity.Property(e => e.OfferSummary)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.OfferType)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.OfferTypeId).HasColumnName("OfferTypeID");

entity.Property(e => e.OfferUpdateDate).HasColumnType("datetime");

entity.Property(e => e.OfferValue)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasColumnType("text")

.HasColumnName("UPC");

});

modelBuilder.Entity<DistributorCouponsOfferRedemption>(entity =>

{

entity.HasKey(e => e.DistributorCouponsRedemptionsId);

entity.ToTable("DistributorCoupons\_Offer\_Redemptions");

entity.Property(e => e.ClearingRecordId).HasColumnName("ClearingRecordID");

entity.Property(e => e.ClearingStatusId)

.HasColumnName("ClearingStatusID")

.HasDefaultValueSql("((1))");

entity.Property(e => e.CouponOfferId).HasColumnName("CouponOfferID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.ErrorCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.MemberNumber)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SentTime).HasColumnType("datetime");

entity.Property(e => e.UpdateTime).HasColumnType("datetime");

});

modelBuilder.Entity<DistributorCouponsUpdateQueue>(entity =>

{

entity.HasKey(e => e.DistCouponUpdateQueueId)

.HasName("PK\_DIST\_COUPONS\_UpdateQueue");

entity.ToTable("DISTRIBUTOR\_COUPONS\_UpdateQueue");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.DistCouponsOfferId).HasColumnName("Dist\_Coupons\_Offer\_Id");

entity.Property(e => e.OfferIdGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferIdGUID");

});

modelBuilder.Entity<Exception>(entity =>

{

entity.Property(e => e.ExceptionDate).HasPrecision(0);

entity.Property(e => e.ExceptionDetails)

.HasMaxLength(2000)

.IsUnicode(false);

entity.Property(e => e.ExceptionMessage)

.HasMaxLength(500)

.IsUnicode(false);

});

modelBuilder.Entity<ExistingOffer>(entity =>

{

entity.HasNoKey();

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferGUID");

});

modelBuilder.Entity<Group>(entity =>

{

entity.HasKey(e => e.Gid);

entity.Property(e => e.Gid).HasColumnName("GID");

entity.Property(e => e.ActiveDate)

.HasColumnType("datetime")

.HasColumnName("active\_date");

entity.Property(e => e.ActiveDateNum).HasColumnName("active\_date\_num");

entity.Property(e => e.Cid).HasColumnName("CID");

entity.Property(e => e.GroupId).HasColumnName("group\_id");

entity.Property(e => e.GroupName)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("group\_name");

entity.Property(e => e.TerminationDate)

.HasColumnType("datetime")

.HasColumnName("termination\_date");

entity.Property(e => e.TerminationDateNum).HasColumnName("termination\_date\_num");

entity.HasOne(d => d.CidNavigation)

.WithMany(p => p.Groups)

.HasForeignKey(d => d.Cid)

.HasConstraintName("FK\_Groups\_Circulars");

});

modelBuilder.Entity<GroupPage>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.Gid).HasColumnName("GID");

entity.Property(e => e.PageId).HasColumnName("page\_id");

entity.Property(e => e.SortNum).HasColumnName("sort\_num");

entity.HasOne(d => d.GidNavigation)

.WithMany(p => p.GroupPages)

.HasForeignKey(d => d.Gid)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_GroupPages\_Groups");

});

modelBuilder.Entity<IntegrationSetting>(entity =>

{

entity.Property(e => e.CouponExportLocation)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CouponExportUserId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CouponExportUserPassword)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CustomField1)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.CustomField2)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.CustomField3)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.CustomField4)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.CustomField5)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.CustomField6)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.DailySalesFileLocation)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.DailySalesImportUserId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.DailySalesImportUserPassword)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ScheduleTime).HasColumnType("datetime");

});

modelBuilder.Entity<Invoice>(entity =>

{

entity.ToTable("Invoice");

entity.Property(e => e.InvoiceId).HasColumnName("InvoiceID");

entity.Property(e => e.Amount).HasColumnType("decimal(18, 2)");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Description)

.HasMaxLength(200)

.IsUnicode(false);

entity.Property(e => e.InvoiceDate).HasColumnType("datetime");

entity.Property(e => e.InvoiceNumber)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.InvoicePeriodFrom).HasColumnType("datetime");

entity.Property(e => e.InvoicePeriodTo).HasColumnType("datetime");

entity.Property(e => e.Memo)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.SentToDynamicsOn).HasColumnType("datetime");

});

modelBuilder.Entity<InvoiceCoupon>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Modifieddate).HasColumnType("datetime");

});

modelBuilder.Entity<InvoiceDetail>(entity =>

{

entity.Property(e => e.Amount).HasColumnType("decimal(18, 2)");

entity.Property(e => e.InvoiceId).HasColumnName("InvoiceID");

entity.Property(e => e.InvoicePeriodFrom).HasColumnType("datetime");

entity.Property(e => e.InvoicePeriodTo).HasColumnType("datetime");

});

modelBuilder.Entity<JobIntervalType>(entity =>

{

entity.Property(e => e.Description)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<LmJobHistory>(entity =>

{

entity.ToTable("LM\_JOB\_HISTORY");

entity.Property(e => e.LmJobHistoryId).HasColumnName("LM\_Job\_History\_Id");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.LastRunDateTime).HasColumnType("datetime");

entity.Property(e => e.LogfileUrl)

.HasMaxLength(500)

.IsUnicode(false);

});

modelBuilder.Entity<LmReward>(entity =>

{

entity.ToTable("LM\_REWARD");

entity.Property(e => e.LmRewardId).HasColumnName("LM\_REWARD\_ID");

entity.Property(e => e.AdditionalDetails)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.CouponId).HasColumnName("CouponID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.CreatedUserId).HasColumnName("CreatedUserID");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.ImageUrl)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("ImageURL");

entity.Property(e => e.Posdetails)

.HasMaxLength(300)

.IsUnicode(false)

.HasColumnName("POSDetails");

entity.Property(e => e.RewardCouponTypeId).HasColumnName("RewardCouponTypeID");

entity.Property(e => e.RewardDepartmentId).HasColumnName("RewardDepartmentID");

entity.Property(e => e.RewardGroupId).HasColumnName("RewardGroupID");

entity.Property(e => e.RewardMustBeUsedByDate).HasColumnType("datetime");

entity.Property(e => e.RewardQtyAmountMoney).HasColumnType("money");

entity.Property(e => e.RewardTitle)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.RewardTypeId).HasColumnName("RewardTypeID");

entity.Property(e => e.Title)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.ValidFrom).HasColumnType("datetime");

});

modelBuilder.Entity<LmRewardDepartment>(entity =>

{

entity.ToTable("LM\_REWARD\_DEPARTMENTS");

entity.Property(e => e.LmrewardDepartmentId).HasColumnName("LMRewardDepartmentID");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.IsExclude)

.IsRequired()

.HasDefaultValueSql("((1))");

entity.Property(e => e.LmdepartmentExcludeTypeId).HasColumnName("LMDepartmentExcludeTypeID");

entity.Property(e => e.LmrewardId).HasColumnName("LMRewardId");

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UpdateUserId).HasColumnName("UpdateUserID");

});

modelBuilder.Entity<LmRewardGroup>(entity =>

{

entity.ToTable("LM\_REWARD\_GROUPS");

entity.HasIndex(e => e.CouponId, "idx\_CouponID");

entity.HasIndex(e => e.LmrewardId, "idx\_LMRewardID");

entity.HasIndex(e => e.RewardGroupId, "idx\_RewardGroupID");

entity.Property(e => e.LmRewardGroupId).HasColumnName("LM\_REWARD\_GROUP\_ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.LmrewardId).HasColumnName("LMRewardId");

});

modelBuilder.Entity<LmRewardProduct>(entity =>

{

entity.ToTable("LM\_REWARD\_PRODUCT");

entity.HasIndex(e => e.LmRewardId, "idx\_LM\_Reward\_ID");

entity.HasIndex(e => e.ProductId, "idx\_ProductID");

entity.HasIndex(e => e.Upc, "idx\_UPC");

entity.Property(e => e.LmRewardProductId).HasColumnName("LM\_REWARD\_PRODUCT\_ID");

entity.Property(e => e.LmRewardId).HasColumnName("LM\_Reward\_ID");

entity.Property(e => e.ProductId).HasColumnName("ProductID");

entity.Property(e => e.RewardProductTypeId).HasColumnName("RewardProductTypeID");

entity.Property(e => e.Upc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<LmRewardProductType>(entity =>

{

entity.HasKey(e => e.RewardProductTypeId);

entity.ToTable("LM\_REWARD\_PRODUCT\_TYPE");

entity.Property(e => e.RewardProductTypeId).HasColumnName("RewardProductTypeID");

entity.Property(e => e.Description)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RewardProductType)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<LmRewardTarget>(entity =>

{

entity.ToTable("LM\_REWARD\_TARGETS");

entity.Property(e => e.LmrewardTargetId).HasColumnName("LMRewardTargetId");

entity.Property(e => e.ClubId).HasColumnName("ClubID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CreatedUserId).HasColumnName("CreatedUserID");

entity.Property(e => e.IsExclude).HasDefaultValueSql("((0))");

entity.Property(e => e.LmrewardId).HasColumnName("LMRewardId");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedUserId).HasColumnName("ModifiedUserID");

entity.HasOne(d => d.Club)

.WithMany(p => p.LmRewardTargets)

.HasForeignKey(d => d.ClubId)

.HasConstraintName("FK\_LM\_REWARD\_TARGETS\_Clubs");

entity.HasOne(d => d.Lmreward)

.WithMany(p => p.LmRewardTargets)

.HasForeignKey(d => d.LmrewardId)

.HasConstraintName("FK\_LM\_REWARD\_TARGETS\_LM\_REWARD");

});

modelBuilder.Entity<LmRewardTransaction>(entity =>

{

entity.HasKey(e => e.LmRewardTransactionsId);

entity.ToTable("LM\_REWARD\_TRANSACTIONS");

entity.Property(e => e.LmRewardTransactionsId).HasColumnName("LM\_REWARD\_TRANSACTIONS\_ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Lmmessage)

.HasMaxLength(5000)

.IsUnicode(false)

.HasColumnName("LMMessage");

entity.Property(e => e.Lmsubject)

.HasMaxLength(1000)

.IsUnicode(false)

.HasColumnName("LMSubject");

entity.Property(e => e.MemberNumber)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.MessageType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.TransactionType)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<LmRewardType>(entity =>

{

entity.ToTable("LM\_REWARD\_TYPE");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.Description)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.ExcludeDepartmentList)

.HasMaxLength(500)

.IsUnicode(false);

});

modelBuilder.Entity<LmRewardUpdateQueue>(entity =>

{

entity.HasKey(e => e.RewardUpdateQueueId);

entity.ToTable("LM\_Reward\_Update\_Queue");

entity.Property(e => e.RewardUpdateQueueId).HasColumnName("RewardUpdateQueueID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.RewardId).HasColumnName("RewardID");

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UserId).HasColumnName("UserID");

});

modelBuilder.Entity<LmUserReward>(entity =>

{

entity.ToTable("LM\_USER\_REWARD");

entity.HasIndex(e => e.LmrewardId, "idx\_LMRewardID");

entity.HasIndex(e => e.MemberNumber, "idx\_MemberNumber");

entity.HasIndex(e => e.SsnewsId, "idx\_SSNewsID");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.AppliedDate).HasColumnType("datetime");

entity.Property(e => e.LmrewardId).HasColumnName("LMRewardID");

entity.Property(e => e.MemberNumber)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.SsnewsId).HasColumnName("SSNewsID");

entity.Property(e => e.StatusId).HasColumnName("StatusID");

});

modelBuilder.Entity<MfgCoupon>(entity =>

{

entity.ToTable("MFG\_COUPONS");

entity.HasIndex(e => e.CouponOfferId, "idx\_CouponOfferID");

entity.Property(e => e.BrandName)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.CouponImage)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.IsProcessedForNcr).HasColumnName("IsProcessedForNCR");

entity.Property(e => e.OfferActiveDate).HasColumnType("datetime");

entity.Property(e => e.OfferDescription).IsUnicode(false);

entity.Property(e => e.OfferDisclaimer).HasColumnType("text");

entity.Property(e => e.OfferExpiryDate).HasColumnType("datetime");

entity.Property(e => e.OfferFeturedText).IsUnicode(false);

entity.Property(e => e.OfferFinePrint).IsUnicode(false);

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferGUID");

entity.Property(e => e.OfferShutoffDate).HasColumnType("datetime");

entity.Property(e => e.OfferSummary)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.OfferType)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.OfferUpdateDate)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.OfferValue)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SsnewsId).HasColumnName("SSNewsId");

entity.Property(e => e.Upc)

.HasColumnType("text")

.HasColumnName("UPC");

});

modelBuilder.Entity<MfgCouponsHold>(entity =>

{

entity.HasKey(e => e.MfgCouponHoldId);

entity.ToTable("MFG\_COUPONS\_HOLD");

entity.Property(e => e.MfgCouponHoldId).HasColumnName("MfgCouponHoldID");

entity.Property(e => e.Comments).HasColumnType("text");

entity.Property(e => e.CouponOfferId).HasColumnName("CouponOfferID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.HoldDate).HasColumnType("datetime");

entity.Property(e => e.ManucaturerCode)

.HasMaxLength(25)

.IsUnicode(false);

entity.Property(e => e.ManufacturerId)

.HasMaxLength(20)

.IsUnicode(false)

.HasColumnName("ManufacturerID");

entity.Property(e => e.NchclearingHouseId).HasColumnName("NCHClearingHouseID");

entity.Property(e => e.ResolutionComments).HasColumnType("text");

entity.Property(e => e.ResolvedByUserId).HasColumnName("ResolvedByUserID");

entity.Property(e => e.SsnewsId).HasColumnName("SSNewsID");

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UpdateUserId).HasColumnName("UpdateUserID");

});

modelBuilder.Entity<MfgCouponsHoldStore>(entity =>

{

entity.HasKey(e => e.MfgCouponHoldStoreId);

entity.ToTable("MFG\_COUPONS\_HOLD\_STORES");

entity.Property(e => e.MfgCouponHoldStoreId).HasColumnName("MfgCouponHoldStoreID");

entity.Property(e => e.CouponOfferId).HasColumnName("CouponOfferID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.MfgCouponHoldId).HasColumnName("MfgCouponHoldID");

entity.Property(e => e.StatusId).HasColumnName("StatusID");

entity.Property(e => e.StoreId).HasColumnName("StoreID");

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UpdateUserId).HasColumnName("UpdateUserID");

});

modelBuilder.Entity<MfgCouponsUpdateQueue>(entity =>

{

entity.HasKey(e => e.MfgcouponUpdateQueueId);

entity.ToTable("MFG\_COUPONS\_UpdateQueue");

entity.HasIndex(e => e.OfferId, "idx\_OfferID");

entity.Property(e => e.MfgcouponUpdateQueueId).HasColumnName("MFGCouponUpdateQueueId");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.OfferId).HasColumnName("OfferID");

entity.Property(e => e.OfferIdGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferIdGUID");

});

modelBuilder.Entity<MyCart>(entity =>

{

entity.HasKey(e => e.CartId)

.HasName("PK\_MyCartInfo");

entity.ToTable("MyCart");

entity.HasIndex(e => e.OfferId, "idx\_OfferID");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.CategoryId)

.HasMaxLength(10)

.IsFixedLength();

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.RedeemOn).HasColumnType("datetime");

});

modelBuilder.Entity<MyCartProduct>(entity =>

{

entity.ToTable("MyCart\_Products");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.UserAddedItem)

.HasMaxLength(200)

.IsUnicode(false);

});

modelBuilder.Entity<NewsCategory>(entity =>

{

entity.Property(e => e.CategoryName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<NewsItem>(entity =>

{

entity.ToTable("News\_Items");

entity.Property(e => e.NewsItemId)

.ValueGeneratedNever()

.HasColumnName("NewsItemID");

entity.Property(e => e.CreateDatetime).HasColumnType("smalldatetime");

entity.Property(e => e.Details).HasColumnType("text");

entity.Property(e => e.Image).HasMaxLength(250);

entity.Property(e => e.NewsItemCategoryId).HasColumnName("NewsItemCategoryID");

entity.Property(e => e.Title)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.UpdateDateTime).HasColumnType("smalldatetime");

entity.Property(e => e.ValidFrom).HasColumnType("smalldatetime");

entity.Property(e => e.ValidTo).HasColumnType("smalldatetime");

});

modelBuilder.Entity<NewsItemCategory>(entity =>

{

entity.ToTable("News\_Item\_Categories");

entity.Property(e => e.NewsItemCategoryId)

.ValueGeneratedNever()

.HasColumnName("NewsItemCategoryID");

entity.Property(e => e.Description).HasMaxLength(50);

});

modelBuilder.Entity<NewsTarget>(entity =>

{

entity.HasIndex(e => e.ClubId, "idx\_ClubID");

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CreatedUserId).HasColumnName("CreatedUserID");

entity.Property(e => e.IsExclude).HasDefaultValueSql("((0))");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedUserId).HasColumnName("ModifiedUserID");

entity.HasOne(d => d.Club)

.WithMany(p => p.NewsTargets)

.HasForeignKey(d => d.ClubId)

.HasConstraintName("FK\_NewsTargets\_Clubs");

entity.HasOne(d => d.News)

.WithMany(p => p.NewsTargets)

.HasForeignKey(d => d.NewsId)

.HasConstraintName("FK\_NewsTargets\_SSNews");

});

modelBuilder.Entity<Page>(entity =>

{

entity.HasKey(e => e.Pid);

entity.Property(e => e.Pid).HasColumnName("PID");

entity.Property(e => e.Cid).HasColumnName("CID");

entity.Property(e => e.Height).HasColumnName("height");

entity.Property(e => e.ImageFull)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("image\_full");

entity.Property(e => e.ImageMid)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("image\_mid");

entity.Property(e => e.ImageThumb)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("image\_thumb");

entity.Property(e => e.PageId).HasColumnName("page\_id");

entity.Property(e => e.Width).HasColumnName("width");

entity.HasOne(d => d.CidNavigation)

.WithMany(p => p.Pages)

.HasForeignKey(d => d.Cid)

.HasConstraintName("FK\_Pages\_Circulars");

});

modelBuilder.Entity<PageItem>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.AdItemId).HasColumnName("ad\_item\_id");

entity.Property(e => e.DealDetails)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("deal\_details");

entity.Property(e => e.DepartmentId)

.HasMaxLength(100)

.IsUnicode(false)

.HasColumnName("department\_id");

entity.Property(e => e.DepartmentIdInt).HasColumnName("department\_id\_int");

entity.Property(e => e.Height).HasColumnName("height");

entity.Property(e => e.ItemImage)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("item\_image");

entity.Property(e => e.ItemName)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("item\_name");

entity.Property(e => e.ItemPrice)

.HasMaxLength(200)

.IsUnicode(false)

.HasColumnName("item\_price");

entity.Property(e => e.Pid).HasColumnName("PID");

entity.Property(e => e.RecipeId).HasColumnName("recipe\_id");

entity.Property(e => e.RecipeImage)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("recipe\_image");

entity.Property(e => e.RecipeMoreKeyword)

.HasMaxLength(100)

.IsUnicode(false)

.HasColumnName("recipe\_more\_keyword");

entity.Property(e => e.RecipeName)

.HasMaxLength(200)

.IsUnicode(false)

.HasColumnName("recipe\_name");

entity.Property(e => e.Width).HasColumnName("width");

entity.Property(e => e.X).HasColumnName("x");

entity.Property(e => e.Y).HasColumnName("y");

entity.HasOne(d => d.PidNavigation)

.WithMany(p => p.PageItems)

.HasForeignKey(d => d.Pid)

.HasConstraintName("FK\_PageItems\_Pages");

});

modelBuilder.Entity<PaymentTerm>(entity =>

{

entity.HasKey(e => e.PaymentTermsId);

entity.Property(e => e.PaymentTermDescription)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<Possoftware>(entity =>

{

entity.ToTable("POSSoftwares");

entity.Property(e => e.PossoftwareId).HasColumnName("POSSoftwareId");

entity.Property(e => e.PosvendorId).HasColumnName("POSVendorId");

entity.Property(e => e.SoftwareName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<Posvendor>(entity =>

{

entity.ToTable("POSVendors");

entity.Property(e => e.PosvendorId).HasColumnName("POSVendorId");

entity.Property(e => e.Email)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PhoneNumber)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<Product>(entity =>

{

entity.ToTable("Product");

entity.HasIndex(e => e.ProductCode, "IX\_Product");

entity.HasIndex(e => e.ProductCategoryId, "idx\_ProductCategoryId");

entity.HasIndex(e => e.ProductCode, "idx\_ProductCode");

entity.Property(e => e.BarCodeImage)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.BarCodeValue)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.Description)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ProductCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductImage)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.ProductName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SalePrice).HasColumnType("decimal(18, 2)");

entity.Property(e => e.Sku)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("SKU");

entity.HasOne(d => d.ProductCategory)

.WithMany(p => p.Products)

.HasForeignKey(d => d.ProductCategoryId)

.HasConstraintName("FK\_Product\_Company");

});

modelBuilder.Entity<Product1>(entity =>

{

entity.HasKey(e => e.ProductId);

entity.ToTable("Products");

entity.Property(e => e.ProductCode)

.HasMaxLength(10)

.IsUnicode(false);

entity.Property(e => e.ProductName)

.HasMaxLength(50)

.IsUnicode(false);

entity.HasOne(d => d.ProductCategory)

.WithMany(p => p.Product1s)

.HasForeignKey(d => d.ProductCategoryId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("FK\_Products\_ProductCategories");

});

modelBuilder.Entity<ProductCategory>(entity =>

{

entity.HasIndex(e => e.ClientDepartmentId, "idx\_ClientDepartmentID");

entity.HasIndex(e => e.MajorDepartmentId, "idx\_MajorDepartmentID");

entity.Property(e => e.ClientDepartmentId).HasColumnName("ClientDepartmentID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DefaultProductId).HasColumnName("DefaultProductID");

entity.Property(e => e.MajorDepartmentId).HasColumnName("MajorDepartmentID");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ProductCategoryName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<ProductImport>(entity =>

{

entity.HasNoKey();

entity.HasIndex(e => e.SellingUnitId, "ci\_azure\_fixup\_dbo\_ProductImports")

.IsClustered();

entity.Property(e => e.BarCodeImage)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.BarCodeValue)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CompanyName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.Description)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.ProductCategory)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductImage)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.SalePrice).HasColumnType("decimal(18, 2)");

entity.Property(e => e.Sku)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("SKU");

});

modelBuilder.Entity<ProductMajorCategory>(entity =>

{

entity.ToTable("Product\_Major\_Categories");

entity.HasIndex(e => e.ClientMajorDepartmentId, "idx\_ClientMajorDepartmentID");

entity.Property(e => e.ProductMajorCategoryId).HasColumnName("ProductMajorCategoryID");

entity.Property(e => e.CategoryName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ClientMajorDepartmentId).HasColumnName("ClientMajorDepartmentID");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.DepartmentImageName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

});

modelBuilder.Entity<ProductsTest>(entity =>

{

entity.HasNoKey();

entity.ToTable("PRODUCTS\_Test");

entity.Property(e => e.ClientDepartmentId).HasColumnName("ClientDepartmentID");

entity.Property(e => e.ProductCategoryName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ProductName)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<Promotion>(entity =>

{

entity.ToTable("Promotion");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.DiscountPercentageCount).HasColumnType("money");

entity.Property(e => e.NcrpromotionCode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("NCRPromotionCode");

entity.Property(e => e.NcrpromotionCreatedDate)

.HasColumnType("datetime")

.HasColumnName("NCRPromotionCreatedDate");

entity.Property(e => e.PromotionId)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<PushNotificationsQueue>(entity =>

{

entity.ToTable("PushNotificationsQueue");

entity.Property(e => e.AndriodNotificationDate).HasColumnType("datetime");

entity.Property(e => e.AndriodNotificationSent).HasColumnName("andriodNotificationSent");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.IPhoneNotificationDate)

.HasColumnType("datetime")

.HasColumnName("iPhoneNotificationDate");

entity.Property(e => e.IPhoneNotificationSent).HasColumnName("iPhoneNotificationSent");

entity.Property(e => e.PickUpControlNumber)

.HasMaxLength(25)

.IsUnicode(false);

});

modelBuilder.Entity<QualifyingProduct>(entity =>

{

entity.Property(e => e.DirectiveId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.Name)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.Reference)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.Type)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.UnitPrice)

.HasMaxLength(100)

.IsUnicode(false);

});

modelBuilder.Entity<Redemption>(entity =>

{

entity.HasNoKey();

entity.HasIndex(e => e.BasketId, "idx\_BasketID");

entity.HasIndex(e => e.PromotionId, "idx\_PromotionID");

entity.HasIndex(e => e.RedemptionId, "idx\_RedemptionID");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.BasketId).HasColumnName("BasketID");

entity.Property(e => e.Category)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CostAmount).HasColumnType("money");

entity.Property(e => e.PromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("PromotionID");

entity.Property(e => e.RedeemAmount).HasColumnType("money");

entity.Property(e => e.RedemptionDate).HasColumnType("datetime");

entity.Property(e => e.RedemptionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("RedemptionID");

entity.Property(e => e.SaleAmount).HasColumnType("money");

entity.Property(e => e.Title)

.HasMaxLength(200)

.IsUnicode(false);

entity.Property(e => e.TotalAmount).HasColumnType("money");

entity.Property(e => e.TransactionDate)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.TransactionTime)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasMaxLength(20)

.IsUnicode(false)

.HasColumnName("UPC");

entity.Property(e => e.UserId).HasColumnName("UserID");

entity.Property(e => e.UserStoreId).HasColumnName("UserStoreID");

});

modelBuilder.Entity<RedemptionDataJson>(entity =>

{

entity.HasKey(e => e.RedeemOfferJsonId)

.HasName("PK\_RedemptionData\_JSON1");

entity.ToTable("RedemptionData\_JSON");

entity.Property(e => e.CouponOfferGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("CouponOfferGUID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.RedemptionJson).HasColumnType("text");

});

modelBuilder.Entity<RedemptionDistribution>(entity =>

{

entity.ToTable("RedemptionDistribution");

entity.HasIndex(e => e.MemberNumber, "idx\_MemberNumber");

entity.HasIndex(e => e.PromotionId, "idx\_PromotionID");

entity.HasIndex(e => e.RedemptionDate, "idx\_RedemptionDate");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Customer)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.DirectiveId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.DirectiveType)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.DistributionOfferAlias)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.MemberNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PromotionId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.RdjsonId).HasColumnName("RDJsonID");

entity.Property(e => e.RedeemAmount)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.RedeemEnterpriseId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.RedeemId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.RedeemType)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.RedemptionDate).HasColumnType("datetime");

});

modelBuilder.Entity<RedemptionDistributionJson>(entity =>

{

entity.HasKey(e => e.RdjsonId);

entity.ToTable("RedemptionDistribution\_JSON");

entity.Property(e => e.RdjsonId).HasColumnName("RDJsonID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Rdstatus).HasColumnName("RDStatus");

entity.Property(e => e.RedemptionJson).HasColumnType("text");

});

modelBuilder.Entity<RsaTransactionHistory>(entity =>

{

entity.HasKey(e => e.TransactionHistoryId);

entity.ToTable("RSA\_Transaction\_History");

entity.Property(e => e.CreatedDateTime).HasColumnType("datetime");

entity.Property(e => e.Json).IsUnicode(false);

entity.Property(e => e.RewardId).HasColumnName("RewardID");

entity.Property(e => e.TypeId).HasColumnName("TypeID");

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UserId).HasColumnName("UserID");

entity.Property(e => e.VersionId).HasColumnName("VersionID");

});

modelBuilder.Entity<SalesHeader>(entity =>

{

entity.ToTable("SalesHeader");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.MemberNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.TransactionAmount)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.TransactionDate).HasColumnType("datetime");

});

modelBuilder.Entity<SalesHeaderDetail>(entity =>

{

entity.Property(e => e.Amount).HasColumnType("decimal(18, 2)");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DiscountAmount).HasColumnType("decimal(18, 2)");

entity.Property(e => e.NcrpromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("NCRPromotionId");

entity.Property(e => e.ProductUpc)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<SalesQualifyingProduct>(entity =>

{

entity.ToTable("SalesQualifyingProduct");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.OfferItemType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemSno).HasColumnName("RedeemSNO");

entity.Property(e => e.Reference)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.UnitPrice)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<SalesRedeem>(entity =>

{

entity.HasKey(e => e.RedeemSno);

entity.ToTable("SalesRedeem");

entity.Property(e => e.RedeemSno).HasColumnName("RedeemSNO");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Customer)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.OneTimeRedeemCode)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PromotionId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.PromotionName)

.HasMaxLength(200)

.IsUnicode(false);

entity.Property(e => e.RedeemAmount)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.RedeemDateTime).HasColumnType("datetime");

entity.Property(e => e.RedeemEnterpriseId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemLaneNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemStoreNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemTransactionNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemType)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<SalesReward>(entity =>

{

entity.HasKey(e => e.RewardNo)

.HasName("PK\_SalesRewards");

entity.ToTable("SalesReward");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Name)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.OfferItemType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RedeemSno).HasColumnName("RedeemSNO");

entity.Property(e => e.Reference)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.UnitPrice)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<SelectedProductIdForWeeklySpecial>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.SpecialPrice)

.HasMaxLength(50)

.IsUnicode(false);

entity.HasOne(d => d.News)

.WithMany(p => p.SelectedProductIdForWeeklySpecials)

.HasForeignKey(d => d.NewsId)

.HasConstraintName("FK\_SelectedProductIdForWeeklySpecials\_SSNews");

});

modelBuilder.Entity<SelectedStoresForCoupon>(entity =>

{

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.EnterpriseId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.StoreRouteId)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<SelectedStoresForIntelligentCoupon>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.EnterpriseId)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.StoreRouteId)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<SellingUnit>(entity =>

{

entity.Property(e => e.SellingUnits)

.HasMaxLength(20)

.IsUnicode(false);

});

modelBuilder.Entity<Ssnews>(entity =>

{

entity.HasKey(e => e.NewsId);

entity.ToTable("SSNews");

entity.HasIndex(e => e.ExpiresOn, "idx\_ExpiresOn");

entity.HasIndex(e => e.MfgShutOffDate, "idx\_MfgShutOffDate");

entity.HasIndex(e => e.ValidFromDate, "idx\_ValidFromDate");

entity.Property(e => e.NewsId).HasColumnName("NewsID");

entity.Property(e => e.Amount).HasColumnType("money");

entity.Property(e => e.CouponCode)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.CouponLimit).HasDefaultValueSql("((1))");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.CustomerId).HasColumnName("CustomerID");

entity.Property(e => e.Details).IsUnicode(false);

entity.Property(e => e.DiscountAmount).HasColumnType("money");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.ImagePath)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.JsonPageId).HasDefaultValueSql("((0))");

entity.Property(e => e.MfgShutOffDate).HasColumnType("datetime");

entity.Property(e => e.NcrpromotionCode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("NCRPromotionCode");

entity.Property(e => e.NcrpromotionCreatedDate)

.HasColumnType("datetime")

.HasColumnName("NCRPromotionCreatedDate");

entity.Property(e => e.NewsCategoryId).HasColumnName("NewsCategoryID");

entity.Property(e => e.NewsStatusId)

.HasColumnName("NewsStatusID")

.HasDefaultValueSql("((1))");

entity.Property(e => e.OtherDetails)

.HasMaxLength(300)

.IsUnicode(false);

entity.Property(e => e.Puicode)

.HasMaxLength(25)

.IsUnicode(false)

.HasColumnName("PUICode");

entity.Property(e => e.Title)

.HasMaxLength(200)

.IsUnicode(false);

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UpdateUserId).HasColumnName("UpdateUserID");

entity.Property(e => e.ValidFromDate).HasColumnType("datetime");

});

modelBuilder.Entity<SsnewsDepartment>(entity =>

{

entity.ToTable("SSNews\_Departments");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.IsMajorDepartment).HasDefaultValueSql("((0))");

entity.HasOne(d => d.News)

.WithMany(p => p.SsnewsDepartments)

.HasForeignKey(d => d.NewsId)

.HasConstraintName("FK\_SSNews\_Departments\_SSNews");

});

modelBuilder.Entity<SsnewsNcrpromotion>(entity =>

{

entity.ToTable("SSNews\_NCRPromotions");

entity.HasIndex(e => e.NcrpromotionId, "idx\_NCRPromotionID");

entity.HasIndex(e => e.SsnewsId, "idx\_SSNewsID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.EnterpriseId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.NcrpromotionId)

.HasMaxLength(100)

.IsUnicode(false)

.HasColumnName("NCRPromotionId");

entity.Property(e => e.SsnewsId).HasColumnName("SSNewsId");

});

modelBuilder.Entity<SsnewsProduct>(entity =>

{

entity.ToTable("SSNews\_Products");

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.HasIndex(e => e.ProductId, "idx\_ProductID");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CouponProductTypeId)

.HasColumnName("CouponProductTypeID")

.HasDefaultValueSql("((1))");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.NewsId).HasColumnName("NewsID");

});

modelBuilder.Entity<SsnewsProductsDeleted>(entity =>

{

entity.ToTable("SSNews\_Products\_Deleted");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.NewsId).HasColumnName("NewsID");

});

modelBuilder.Entity<SsnewsRecurring>(entity =>

{

entity.HasKey(e => e.RecurringId);

entity.ToTable("SSNewsRecurrings");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.RecurringEndDate).HasColumnType("datetime");

entity.Property(e => e.RecurringStartDate).HasColumnType("datetime");

entity.HasOne(d => d.News)

.WithMany(p => p.SsnewsRecurrings)

.HasForeignKey(d => d.NewsId)

.HasConstraintName("FK\_SSNewsRecurrings\_SSNewsRecurrings");

});

modelBuilder.Entity<SsnewsUserFavoritesDetail>(entity =>

{

entity.HasKey(e => e.UserFavoriteListDetailsId);

entity.ToTable("SSNews\_UserFavorites\_Details");

entity.Property(e => e.UserFavoriteListDetailsId).HasColumnName("UserFavoriteListDetailsID");

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.NewsId).HasColumnName("NewsID");

entity.Property(e => e.StoreId).HasColumnName("StoreID");

entity.Property(e => e.UpdateDate).HasColumnType("datetime");

entity.Property(e => e.UserFavoriteListId).HasColumnName("UserFavoriteListID");

});

modelBuilder.Entity<SsnewsUserFavoritesList>(entity =>

{

entity.HasKey(e => e.UserFavoriteListId);

entity.ToTable("SSNews\_UserFavorites\_List");

entity.Property(e => e.UserFavoriteListId).HasColumnName("UserFavoriteListID");

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.UpdateDate).HasColumnType("datetime");

entity.Property(e => e.UserId).HasColumnName("UserID");

});

modelBuilder.Entity<SsnewsUsersNcrimpression>(entity =>

{

entity.HasKey(e => e.NcrimpressionId);

entity.ToTable("SSNewsUsersNCRImpressions");

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.NcrimpressionId).HasColumnName("NCRImpressionId");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.NcrimpressionCode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("NCRImpressionCode");

entity.Property(e => e.NcrimpressionDate)

.HasColumnType("datetime")

.HasColumnName("NCRImpressionDate");

});

modelBuilder.Entity<SsnewsUsersRedeem>(entity =>

{

entity.ToTable("SSNewsUsersRedeem");

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.RedeemOn).HasColumnType("datetime");

});

modelBuilder.Entity<State>(entity =>

{

entity.Property(e => e.StateName).HasMaxLength(100);

entity.HasOne(d => d.Country)

.WithMany(p => p.States)

.HasForeignKey(d => d.CountryId)

.OnDelete(DeleteBehavior.ClientSetNull)

.HasConstraintName("states$FK\_states\_Country");

});

modelBuilder.Entity<StoreCouponsUpdatequeue>(entity =>

{

entity.HasKey(e => e.StoreCouponUpdateQueueId)

.HasName("PK\_STORE\_COUPONS\_UpdateQueue");

entity.ToTable("STORE\_COUPONS\_UPDATEQUEUE");

entity.Property(e => e.StoreCouponUpdateQueueId).HasColumnName("StoreCouponUpdateQueueID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.Isprocessed).HasDefaultValueSql("((0))");

entity.Property(e => e.OfferId).HasColumnName("OfferID");

});

modelBuilder.Entity<SubCategory>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.SubDepartmentName)

.HasMaxLength(100)

.IsUnicode(false);

});

modelBuilder.Entity<TimeZone>(entity =>

{

entity.HasKey(e => e.TimeZoneId)

.HasName("PK\_timezones\_TimeZoneId");

entity.Property(e => e.BaseUtcOffSet)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.DaylightName)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.DisplayName)

.HasMaxLength(250)

.IsUnicode(false);

entity.Property(e => e.Id)

.HasMaxLength(100)

.IsUnicode(false);

entity.Property(e => e.IphoneTimeZoneName)

.HasMaxLength(50)

.IsFixedLength();

entity.Property(e => e.StandardName)

.HasMaxLength(250)

.IsUnicode(false);

});

modelBuilder.Entity<UserAdminStore>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.HasOne(d => d.User)

.WithMany(p => p.UserAdminStores)

.HasForeignKey(d => d.UserId)

.HasConstraintName("FK\_UserAdminStores\_UserDetails");

});

modelBuilder.Entity<UserDetail>(entity =>

{

entity.HasIndex(e => e.BarCodeValue, "idx\_BarcodeValue");

entity.HasIndex(e => e.IsDeleted, "idx\_IsDeleted");

entity.HasIndex(e => e.SignUpDate, "idx\_SignUpDate");

entity.Property(e => e.AlternateMemberNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.BarCodeImage)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.BarCodeValue)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.CustomField1)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomField2)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.CustomerId).HasColumnName("CustomerID");

entity.Property(e => e.DeviceType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Email)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.FirstName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.LastName)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.Mobile)

.HasMaxLength(15)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.NewTermsAcceptanceRequired).HasDefaultValueSql("((1))");

entity.Property(e => e.QrcodeImage)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("QRCodeImage");

entity.Property(e => e.QrcodeValue)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("QRCodeValue");

entity.Property(e => e.Qtoken)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("QToken");

entity.Property(e => e.SignUpDate).HasColumnType("datetime");

entity.Property(e => e.UserId).HasColumnName("UserID");

entity.Property(e => e.UserName)

.HasMaxLength(150)

.IsUnicode(false);

entity.Property(e => e.XCustomerCode)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("xCustomerCode");

entity.Property(e => e.XUserId).HasColumnName("xUserId");

entity.Property(e => e.ZipCode)

.HasMaxLength(10)

.IsUnicode(false);

});

modelBuilder.Entity<UserDevice>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DeviceId)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

});

modelBuilder.Entity<UserDeviceInfo>(entity =>

{

entity.ToTable("UserDeviceInfo");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.UserDeviceInfoId).HasColumnName("UserDeviceInfoID");

entity.Property(e => e.Awsarn)

.HasMaxLength(500)

.IsUnicode(false)

.HasColumnName("AWSARN");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.DeviceInfo)

.HasMaxLength(500)

.IsUnicode(false);

});

modelBuilder.Entity<UserProductUsage>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<UserProfile>(entity =>

{

entity.HasKey(e => e.UserId)

.HasName("PK\_\_UserProf\_\_1788CC4C18981236");

entity.ToTable("UserProfile");

entity.HasIndex(e => e.UserName, "UQ\_\_UserProf\_\_C9F28456AE5E6A6D")

.IsUnique();

entity.Property(e => e.UserName).HasMaxLength(56);

entity.HasMany(d => d.Roles)

.WithMany(p => p.Users)

.UsingEntity<Dictionary<string, object>>(

"WebpagesUsersInRole",

l => l.HasOne<WebpagesRole>().WithMany().HasForeignKey("RoleId").OnDelete(DeleteBehavior.ClientSetNull).HasConstraintName("fk\_RoleId"),

r => r.HasOne<UserProfile>().WithMany().HasForeignKey("UserId").OnDelete(DeleteBehavior.ClientSetNull).HasConstraintName("fk\_UserId"),

j =>

{

j.HasKey("UserId", "RoleId").HasName("PK\_\_webpages\_\_AF2760AD844932F8");

j.ToTable("webpages\_UsersInRoles");

});

});

modelBuilder.Entity<UserStatus>(entity =>

{

entity.Property(e => e.UserStatusDesc).HasMaxLength(40);

});

modelBuilder.Entity<UserTermsAcceptance>(entity =>

{

entity.Property(e => e.AcceptDateTime).HasColumnType("datetime");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DeviceId).HasColumnName("DeviceID");

entity.Property(e => e.DeviceInfo)

.HasMaxLength(1000)

.IsUnicode(false);

});

modelBuilder.Entity<UserToken>(entity =>

{

entity.HasIndex(e => e.LoginToken, "idx\_LoginToken");

entity.HasIndex(e => e.UserId, "idx\_UserID");

entity.Property(e => e.DeviceInfo).HasMaxLength(500);

entity.Property(e => e.DeviceRegistrationId).HasMaxLength(500);

entity.Property(e => e.LastAccessed).HasPrecision(0);

entity.Property(e => e.LoginToken).HasMaxLength(45);

entity.Property(e => e.MobileModel).HasMaxLength(45);

entity.Property(e => e.MobileOs)

.HasMaxLength(45)

.HasColumnName("MobileOS");

entity.Property(e => e.MobileOsversion)

.HasMaxLength(45)

.HasColumnName("MobileOSVersion");

});

modelBuilder.Entity<UserType>(entity =>

{

entity.Property(e => e.UserType1)

.HasMaxLength(100)

.HasColumnName("UserType");

});

modelBuilder.Entity<UsersCouponsRedeem>(entity =>

{

entity.ToTable("UsersCouponsRedeem");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.RedeemOn).HasColumnType("datetime");

});

modelBuilder.Entity<ViewCustomGetClubUser>(entity =>

{

entity.HasNoKey();

entity.ToView("VIEW\_CUSTOM\_GET\_CLUB\_USERS");

entity.Property(e => e.Clubid).HasColumnName("CLUBID");

entity.Property(e => e.Clubname)

.HasMaxLength(300)

.IsUnicode(false)

.HasColumnName("CLUBNAME");

entity.Property(e => e.Userid).HasColumnName("USERID");

});

modelBuilder.Entity<ViewCustomUserDetail>(entity =>

{

entity.HasNoKey();

entity.ToView("VIEW\_CUSTOM\_USER\_DETAILS");

entity.Property(e => e.Barcodevalue)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("BARCODEVALUE");

entity.Property(e => e.Clientstoreid).HasColumnName("CLIENTSTOREID");

entity.Property(e => e.Email)

.HasMaxLength(150)

.IsUnicode(false)

.HasColumnName("EMAIL");

entity.Property(e => e.Firstname)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("FIRSTNAME");

entity.Property(e => e.Isactive).HasColumnName("ISACTIVE");

entity.Property(e => e.Lastname)

.HasMaxLength(15)

.IsUnicode(false)

.HasColumnName("LASTNAME");

entity.Property(e => e.Mobile)

.HasMaxLength(15)

.IsUnicode(false)

.HasColumnName("MOBILE");

entity.Property(e => e.Name)

.HasMaxLength(66)

.IsUnicode(false)

.HasColumnName("NAME");

entity.Property(e => e.Roleid).HasColumnName("ROLEID");

entity.Property(e => e.Rolename)

.HasMaxLength(256)

.HasColumnName("ROLENAME");

entity.Property(e => e.Userdetailid)

.ValueGeneratedOnAdd()

.HasColumnName("USERDETAILID");

entity.Property(e => e.Userid).HasColumnName("USERID");

entity.Property(e => e.Username)

.HasMaxLength(150)

.IsUnicode(false)

.HasColumnName("USERNAME");

entity.Property(e => e.Usertype)

.HasMaxLength(100)

.HasColumnName("USERTYPE");

entity.Property(e => e.Usertypeid).HasColumnName("USERTYPEID");

entity.Property(e => e.Zipcode)

.HasMaxLength(10)

.IsUnicode(false)

.HasColumnName("ZIPCODE");

});

modelBuilder.Entity<VwAspnetApplication>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_Applications");

entity.Property(e => e.ApplicationName).HasMaxLength(256);

entity.Property(e => e.Description).HasMaxLength(256);

entity.Property(e => e.LoweredApplicationName).HasMaxLength(256);

});

modelBuilder.Entity<VwAspnetMembershipUser>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_MembershipUsers");

entity.Property(e => e.Comment).HasColumnType("ntext");

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.Email).HasMaxLength(256);

entity.Property(e => e.FailedPasswordAnswerAttemptWindowStart).HasColumnType("datetime");

entity.Property(e => e.FailedPasswordAttemptWindowStart).HasColumnType("datetime");

entity.Property(e => e.LastActivityDate).HasColumnType("datetime");

entity.Property(e => e.LastLockoutDate).HasColumnType("datetime");

entity.Property(e => e.LastLoginDate).HasColumnType("datetime");

entity.Property(e => e.LastPasswordChangedDate).HasColumnType("datetime");

entity.Property(e => e.LoweredEmail).HasMaxLength(256);

entity.Property(e => e.MobileAlias).HasMaxLength(16);

entity.Property(e => e.MobilePin)

.HasMaxLength(16)

.HasColumnName("MobilePIN");

entity.Property(e => e.PasswordAnswer).HasMaxLength(128);

entity.Property(e => e.PasswordQuestion).HasMaxLength(256);

entity.Property(e => e.UserName).HasMaxLength(256);

});

modelBuilder.Entity<VwAspnetProfile>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_Profiles");

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<VwAspnetRole>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_Roles");

entity.Property(e => e.Description).HasMaxLength(256);

entity.Property(e => e.LoweredRoleName).HasMaxLength(256);

entity.Property(e => e.RoleName).HasMaxLength(256);

});

modelBuilder.Entity<VwAspnetUser>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_Users");

entity.Property(e => e.LastActivityDate).HasColumnType("datetime");

entity.Property(e => e.LoweredUserName).HasMaxLength(256);

entity.Property(e => e.MobileAlias).HasMaxLength(16);

entity.Property(e => e.UserName).HasMaxLength(256);

});

modelBuilder.Entity<VwAspnetUsersInRole>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_UsersInRoles");

});

modelBuilder.Entity<VwAspnetWebPartStatePath>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_WebPartState\_Paths");

entity.Property(e => e.LoweredPath).HasMaxLength(256);

entity.Property(e => e.Path).HasMaxLength(256);

});

modelBuilder.Entity<VwAspnetWebPartStateShared>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_WebPartState\_Shared");

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<VwAspnetWebPartStateUser>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_aspnet\_WebPartState\_User");

entity.Property(e => e.LastUpdatedDate).HasColumnType("datetime");

});

modelBuilder.Entity<VwBasketCouponItem>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_BasketCouponItem");

entity.Property(e => e.BasketCouponPromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("BasketCouponPromotionID");

entity.Property(e => e.BasketDataId).HasColumnName("BasketDataID");

entity.Property(e => e.BasketItemId).HasColumnName("BasketItemID");

entity.Property(e => e.BasktcouponUpc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("basktcouponUPC");

entity.Property(e => e.DiscountAmount).HasColumnType("money");

entity.Property(e => e.ItemUpc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("ItemUPC");

});

modelBuilder.Entity<VwRedemptionBasketItem>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_RedemptionBasketItems");

entity.Property(e => e.BasketGuid).HasColumnName("BasketGUID");

entity.Property(e => e.ClearingRecordId).HasColumnName("ClearingRecordID");

entity.Property(e => e.ClearingStatusId).HasColumnName("ClearingStatusID");

entity.Property(e => e.CouponOfferId).HasColumnName("CouponOfferID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.GlobalStoreId).HasColumnName("GlobalStoreID");

entity.Property(e => e.IdType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.IsUpcsplitComplete).HasColumnName("IsUPCSplitComplete");

entity.Property(e => e.MemberNumber)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.OfferGuid)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("OfferGUID");

entity.Property(e => e.Posid).HasColumnName("POSId");

entity.Property(e => e.ProductName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Qty).HasColumnName("QTY");

entity.Property(e => e.RedeemId).HasColumnName("RedeemID");

entity.Property(e => e.Retailer)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.RetailerId).HasColumnName("RetailerID");

entity.Property(e => e.TransactionDate).HasColumnType("date");

entity.Property(e => e.TransactionTenderType)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("UPC");

});

modelBuilder.Entity<VwTestView>(entity =>

{

entity.HasNoKey();

entity.ToView("vw\_TestView");

entity.Property(e => e.BasketCouponPromotionId)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("BasketCouponPromotionID");

entity.Property(e => e.BasketDataId).HasColumnName("BasketDataID");

entity.Property(e => e.BasketItemId).HasColumnName("BasketItemID");

entity.Property(e => e.BasktcouponUpc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("basktcouponUPC");

entity.Property(e => e.DiscountAmount).HasColumnType("money");

entity.Property(e => e.ItemUpc)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("ItemUPC");

});

modelBuilder.Entity<WebDepartment>(entity =>

{

entity.ToTable("Web\_Departments");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.WebDepartmentImage)

.HasMaxLength(1000)

.IsUnicode(false);

entity.Property(e => e.WebDepartmentName)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<WebMenu>(entity =>

{

entity.HasKey(e => e.MenuId);

entity.ToTable("WebMenu");

entity.Property(e => e.Description)

.HasMaxLength(200)

.IsUnicode(false);

});

modelBuilder.Entity<WebpagesMembership>(entity =>

{

entity.HasKey(e => e.UserId)

.HasName("PK\_\_webpages\_\_1788CC4C9E755382");

entity.ToTable("webpages\_Membership");

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

entity.Property(e => e.ConfirmationToken).HasMaxLength(128);

entity.Property(e => e.CreateDate).HasColumnType("datetime");

entity.Property(e => e.IsConfirmed).HasDefaultValueSql("((0))");

entity.Property(e => e.LastPasswordFailureDate).HasColumnType("datetime");

entity.Property(e => e.Password).HasMaxLength(128);

entity.Property(e => e.PasswordChangedDate).HasColumnType("datetime");

entity.Property(e => e.PasswordSalt).HasMaxLength(128);

entity.Property(e => e.PasswordVerificationToken).HasMaxLength(128);

entity.Property(e => e.PasswordVerificationTokenExpirationDate).HasColumnType("datetime");

});

modelBuilder.Entity<WebpagesOauthMembership>(entity =>

{

entity.HasKey(e => new { e.Provider, e.ProviderUserId })

.HasName("PK\_\_webpages\_\_F53FC0EDB9564D6E");

entity.ToTable("webpages\_OAuthMembership");

entity.Property(e => e.Provider).HasMaxLength(30);

entity.Property(e => e.ProviderUserId).HasMaxLength(100);

});

modelBuilder.Entity<WebpagesRole>(entity =>

{

entity.HasKey(e => e.RoleId)

.HasName("PK\_\_webpages\_\_8AFACE1A56104F40");

as

entity.ToTable("webpages\_Roles");

entity.HasIndex(e => e.RoleName, "UQ\_\_webpages\_\_8A2B61606445AAC5")

.IsUnique();

entity.Property(e => e.RoleName).HasMaxLength(256);

});

modelBuilder.Entity<WeeklyAddPdf>(entity =>

{

entity.HasKey(e => e.WeeklyAdPdfId);

entity.ToTable("WeeklyAddPDF");

entity.Property(e => e.WeeklyAdPdfId).HasColumnName("WeeklyAdPdfID");

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.Expireson).HasColumnType("datetime");

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.PdfFileName)

.HasMaxLength(1000)

.IsUnicode(false);

entity.Property(e => e.StoreId).HasColumnName("StoreID");

entity.Property(e => e.ValidFromDate).HasColumnType("datetime");

});

modelBuilder.Entity<WeeklySpecialImport>(entity =>

{

entity.Property(e => e.CreatedDate).HasColumnType("datetime");

entity.Property(e => e.DepartmentName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.ExpiresOn).HasColumnType("datetime");

entity.Property(e => e.ImageName)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.ModifiedDate).HasColumnType("datetime");

entity.Property(e => e.Name)

.HasMaxLength(500)

.IsUnicode(false);

entity.Property(e => e.OfferDetails)

.HasMaxLength(1000)

.IsUnicode(false);

entity.Property(e => e.Price)

.HasMaxLength(20)

.IsUnicode(false);

entity.Property(e => e.ProductName)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Upc)

.HasMaxLength(20)

.IsUnicode(false)

.HasColumnName("UPC");

entity.Property(e => e.ValidFromDate).HasColumnType("datetime");

});

modelBuilder.Entity<WeeklySpecialsExtension>(entity =>

{

entity.HasKey(e => e.ExtensionId);

entity.ToTable("WeeklySpecialsExtension");

entity.HasIndex(e => e.NewsId, "idx\_NewsID");

entity.Property(e => e.ExtensionId).HasColumnName("ExtensionID");

entity.Property(e => e.CreateDateTime).HasColumnType("datetime");

entity.Property(e => e.CreateUserId).HasColumnName("CreateUserID");

entity.Property(e => e.NewsId).HasColumnName("NewsID");

entity.Property(e => e.SpecialPrice)

.HasMaxLength(200)

.IsUnicode(false);

entity.Property(e => e.UpdateDateTime).HasColumnType("datetime");

entity.Property(e => e.UpdateUserId).HasColumnName("UpdateUserID");

entity.HasOne(d => d.News)

.WithMany(p => p.WeeklySpecialsExtensions)

.HasForeignKey(d => d.NewsId)

.HasConstraintName("FK\_WeeklySpecialExtension\_SSNews");

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

}

5. run the following commands in NugetPackageConsole.

Scaffold-DbContext "Server=veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com;Initial Catalog=RSA\_GroceryDEV;MultipleActiveResultSets=true;User ID=Veritragrocerydblogin;Password=otTL5kd0buqb7erEuTqX" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

6.appsettingsjson.cs:

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"BasketdbCon": "Server=veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com;Initial Catalog=RSA\_GroceryDEV;MultipleActiveResultSets=true;User ID=Veritragrocerydblogin;Password=otTL5kd0buqb7erEuTqX"

}

}

7.program.cs:

// add connectionstrings: dbcontext

builder.Services.AddDbContext<RSA\_GroceryDEVContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("BasketdbCon")));

builder.Services.AddScoped<IBasketRepository , BasketRepository>();

Controller.cs:

using BasketDemo.Repository;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

namespace BasketDemo.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class BasketController : ControllerBase

{

private readonly IBasketRepository \_basketRepository;

public BasketController(IBasketRepository basketRepository)

{

\_basketRepository = basketRepository;

}

[HttpGet]

[Route("GetAllBaskets")]

public async Task<IActionResult> GetAllBaskets()

{

var result = await \_basketRepository.GetAllBaskets();

return Ok(result);

}

}

}

BasketRepository.cs:

public interface IBasketRepository

{

Task<List<BasketDatum>> GetAllBaskets();

}

using BasketDemo.Models;

using Microsoft.EntityFrameworkCore;

namespace BasketDemo.Repository

{

public class BasketRepository : IBasketRepository

{

private readonly RSA\_GroceryDEVContext \_context;

public BasketRepository(RSA\_GroceryDEVContext context)

{

\_context = context;

}

public async Task<List<BasketDatum>> GetAllBaskets()

{

return await \_context.BasketData.FromSqlRaw<BasketDatum>("usp\_get\_basketData").ToListAsync();

}

}

}

ENTITY FRAME WORK WITHOUT MIGRATION AND SCAFOLD CONTEXT

1.Add appsetting.json with connectionstirngs:

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"BasketDBCon": "Server=veritrarsadevdb.clkbmvhusrdv.us-east-1.rds.amazonaws.com;Initial Catalog=RSA\_GroceryDEV;MultipleActiveResultSets=true;User ID=Veritragrocerydblogin;Password=otTL5kd0buqb7erEuTqX"

}

}

2.Create table class :

namespace BasketProject.Models

{

public class BasketData

{

public int BasketDataId { get; set; }

public string Retailer { get; set; }

public int POSId { get; set; }

public int StoreId { get; set; }

public int OperatorId { get; set; }

public int TransactionId { get; set; }

public DateTime TransactionDate { get; set; }

public TimeSpan TransactionTime { get; set; }

public int TransactionTotalAmount { get; set; }

public int TransactionTaxAmount { get; set; }

public string TransactionTenderType { get; set; }

public DateTime CreatedDate { get; set; }

}

}

RESPONSE:

namespace BasketProject.Models

{

public class ResponseModel

{

public int StatusCode { get; set; }

public string? StatusMessage { get; set; }

public List<BasketData>? Baskets { get; set; }

public PaginationData? PaginationData { get; set; }

public BasketData? Basket { get; set; }

}

public class PaginationData

{

public int TotalRecords { get; set; }

public int TotalPages { get; set; }

}

}

3.Create DB context class:

using BasketProject.Models;

using Microsoft.EntityFrameworkCore;

namespace BasketProject.Data

{

public class BasketDBContext : DbContext

{

public BasketDBContext(DbContextOptions<BasketDBContext> options) : base(options)

{

}

public DbSet<BasketData> BasketData { get; set; }

public virtual ObjectResult<GetSpecialsForRedeem\_Result> GetSpecialsForRedeem(Nullable<int> customerId, Nullable<int> userId)

{

// Create parameters for the stored procedure

var customerIdParameter = customerId.HasValue ?

new ObjectParameter("CustomerId", customerId) :

new ObjectParameter("CustomerId", typeof(int));

var userIdParameter = userId.HasValue ?

new ObjectParameter("UserId", userId) :

new ObjectParameter("UserId", typeof(int));

// Execute the stored procedure using ExecuteFunction

return ((IObjectContextAdapter)this).ObjectContext.ExecuteFunction<GetSpecialsForRedeem\_Result>("GetSpecialsForRedeem", customerIdParameter, userIdParameter);

}

}

}

4.Create Repository folder

using BasketProject.Models;

namespace BasketProject.Repository

{

public class BasketRepository : IBasketRepository

{

private readonly BasketDBContext \_context;

public BasketRepository

(

BasketDBContext context

)

{

\_context = context;

}

public async Task<List<BasketData>> GetAllBaskets()

{

try

{

return await \_context.BasketData.FromSqlRaw<BasketData>("usp\_get\_basketData").ToListAsync();

}

catch(Exception ex)

{

throw new Exception(ex.Message);

}

}

public async Task<List<BasketData>> GetBasketDataById(int id)

{

try

{

var param = new SqlParameter("@BasketDataId", id);

//var basketDetails = await Task.Run(() => \_context.BasketData

//.FromSqlRaw("exec usp\_getBasketDataById @BasketDataId", param).ToListAsync()

//);

var basketDetails = await \_context.BasketData.FromSqlRaw<BasketData>("exec usp\_getBasketDataById @BasketDataId", param).ToListAsync();

if ( basketDetails != null )

{

return basketDetails;

}

return null;

}

catch(Exception ex)

{

throw new Exception(ex.Message);

}

}

public ResponseModel GetBasketPaginationData(int Page, int Limit, string? SearchTerm, string? SortColumn, string? SortDirection)

{

ResponseModel response = new ResponseModel();

try

{

var parameters = new[]

{

new SqlParameter("@Page", SqlDbType.Int) { Value = Page },

new SqlParameter("@Limit", SqlDbType.Int) { Value = Limit },

new SqlParameter("@SearchTerm", SqlDbType.VarChar, 100) { Value = SearchTerm ?? (object)DBNull.Value },

new SqlParameter("@SortColumn", SqlDbType.VarChar, 100) { Value = SortColumn ?? (object)DBNull.Value },

new SqlParameter("@SortDirection", SqlDbType.VarChar, 100) { Value = SortDirection ?? (object)DBNull.Value }

};

// 10 records per page

var result = \_context.BasketData.FromSqlRaw("EXEC usp\_basket\_pagination @Page, @Limit, @SearchTerm, @SortColumn, @SortDirection", parameters).ToList();

// totalrecords

int totalRecords = \_context.BasketData.Count();

// Calculate total pages

int totalPages = (int)Math.Ceiling((double)totalRecords / Limit);

List<BasketData> basketResuts = new List<BasketData>();

if (result != null)

{

foreach(var item in result)

{

basketResuts.Add(item);

}

response.PaginationData = new PaginationData

{

TotalPages = totalPages,

TotalRecords = totalRecords

};

response.Baskets = basketResuts;

response.StatusMessage = "BasketData has been found";

response.StatusCode = 200;

}

else

{

response.PaginationData = null;

response.StatusMessage = "No BasketData found.";

response.StatusCode = 100;

response.Baskets = null;

}

return response;

}catch ( Exception ex )

{

response.StatusCode = 500;

response.StatusMessage = ex.Message;

throw new Exception(response.StatusMessage);

}

}

public async Task<List<BasketData>> GetBasketRecords(int Page, int Limit, string? SearchTerm, string? SortColumn, string? SortDirection)

{

try

{

var parameters = new[]

{

new SqlParameter("@Page", SqlDbType.Int) { Value = Page },

new SqlParameter("@Limit", SqlDbType.Int) { Value = Limit },

new SqlParameter("@SearchTerm", SqlDbType.VarChar, 100) { Value = SearchTerm ?? (object)DBNull.Value },

new SqlParameter("@SortColumn", SqlDbType.VarChar, 100) { Value = SortColumn ?? (object)DBNull.Value },

new SqlParameter("@SortDirection", SqlDbType.VarChar, 100) { Value = SortDirection ?? (object)DBNull.Value }

};

var result = await \_context.BasketData.FromSqlRaw("EXEC usp\_basket\_pagination @Page, @Limit, @SearchTerm, @SortColumn, @SortDirection", parameters).ToListAsync();

// totalrecords

int totalRecords = \_context.BasketData.Count();

// Calculate total pages

int totalPages = (int)Math.Ceiling((double)totalRecords / Limit);

// Calculate total pages

return result;

}

catch (Exception ex)

{

throw ex;

}

}

public async Task<BasketData> UpdateBasketData(int id,BasketData basketData)

{

try

{

var param = new SqlParameter("@BasketDataId", id);

var parameters = new List<SqlParameter>();

parameters.Add(new SqlParameter("@Retailer", basketData.Retailer));

parameters.Add(new SqlParameter("@POSId", basketData.POSId));

parameters.Add(new SqlParameter("@StoreId", basketData.StoreId));

parameters.Add(new SqlParameter("@OperatorId", basketData.OperatorId));

parameters.Add(new SqlParameter("@TransactionId", basketData.TransactionId));

parameters.Add(new SqlParameter("@TransactionDate", basketData.TransactionDate));

parameters.Add(new SqlParameter("@TransactionTime", basketData.TransactionTime));

parameters.Add(new SqlParameter("@TransactionTotalAmount", basketData.TransactionTotalAmount));

parameters.Add(new SqlParameter("@TransactionTaxAmount", basketData.TransactionTaxAmount));

parameters.Add(new SqlParameter("@TransactionTenderType", basketData.TransactionTenderType));

parameters.Add(new SqlParameter("@CreatedDate", basketData.CreatedDate));

// var result = await Task.Run(() => \_context.Database

//.ExecuteSqlRawAsync($"EXEC usp\_update\_basketData {id},@Retailer,@POSId,@StoreId,@OperatorId,@TransactionId,@TransactionDate,@TransactionTime,@TransactionTotalAmount,@TransactionTaxAmount,@TransactionTenderType,@CreatedDate", parameters.ToArray()));

// var result = await \_context.Database

// .ExecuteSqlRawAsync($"EXEC usp\_update\_basketData {id},@Retailer,@POSId,@StoreId,@OperatorId,@TransactionId,@TransactionDate,@TransactionTime,@TransactionTotalAmount,@TransactionTaxAmount,@TransactionTenderType,@CreatedDate", parameters.ToArray());

var result = await \_context.Database.ExecuteSqlInterpolatedAsync($"EXEC usp\_update\_basketData {id},{basketData.Retailer},{basketData.POSId},{basketData.StoreId},{basketData.OperatorId},{basketData.TransactionId},{basketData.TransactionDate},{basketData.TransactionTime},{basketData.TransactionTotalAmount},{basketData.TransactionTaxAmount},{basketData.TransactionTenderType},{basketData.CreatedDate}");

if (result == 1)

{

return Task.FromResult(basketData).Result;

}

return null;

}

catch (Exception ex)

{

throw ex;

}

}

}

}

5.Controller.cs:

using BasketProject.Repository;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

namespace BasketProject.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class BasketController : ControllerBase

{

private readonly IBasketRepository \_basketRepository;

public BasketController(IBasketRepository basketRepository)

{

\_basketRepository = basketRepository;

}

[HttpGet]

[Route("GetAllBaskets")]

public async Task<IActionResult> GetAllBaskets()

{

try

{

var result = await \_basketRepository.GetAllBaskets();

if(result != null)

{

return Ok(result);

}

return NotFound();

}

catch(Exception ex)

{

return StatusCode(StatusCodes.Status500InternalServerError, ex.Message);

}

}

[HttpGet]

[Route("GetBasketPaginationData")]

public IActionResult GetBasketPaginationData(string? SearchTerm, string? SortColumn, string? SortDirection, int Page = 1, int Limit = 10)

{

try

{

//var result = await \_basketRepository.GetBasketRecords(Page, Limit, SearchTerm, SortColumn, SortDirection);

var result = \_basketRepository.GetBasketPaginationData(Page, Limit, SearchTerm,SortColumn, SortDirection);

if(result != null)

{

return Ok(new { data = result.Baskets, TotalRecords = result.PaginationData.TotalRecords, TotalPages = result.PaginationData.TotalPages });

}

return NotFound();

}catch (Exception ex)

{

return StatusCode(StatusCodes.Status500InternalServerError, ex.Message);

}

}

[HttpPut]

[Route("updateBasketData")]

public async Task<IActionResult> UpdateBasketData(int id , BasketData basketData)

{

try

{

var response = new ResponseModel();

var result = await \_basketRepository.UpdateBasketData(id, basketData);

if(result != null)

{

response.StatusCode = 200;

response.StatusMessage = "BasketData has been updated successfully .....";

response.Basket = result;

return Ok(response);

}

return BadRequest();

}catch (Exception ex)

{

return StatusCode(StatusCodes.Status500InternalServerError,ex.Message);

}

}

[HttpGet]

[Route("getbyBasketId")]

public async Task<IActionResult> GetBasketDataById(int id)

{

try

{

var response = new ResponseModel();

var result = await \_basketRepository.GetBasketDataById(id);

if(result != null)

{

response.StatusCode = 200;

response.Basket = result.FirstOrDefault();

return Ok(result);

}

return BadRequest();

}catch (Exception ex)

{

return StatusCode(StatusCodes.Status500InternalServerError, ex.Message);

}

}

[HttpGet]

[Route("getByName")]

// url deploy:https://hb0hemu0z5.execute-api.ap-south-1.amazonaws.com/Test/api/Basket/getByName?Name=sai

public IActionResult GetByName(string Name)

{

return Ok(new {Name});

}

}

}

6.Program.cs

using BasketProject.Data;

using BasketProject.Repository;

using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllers();

// Learn more about configuring Swagger/OpenAPI at https://aka.ms/aspnetcore/swashbuckle

builder.Services.AddEndpointsApiExplorer();

builder.Services.AddSwaggerGen();

// add connection string

builder.Services.AddDbContext<BasketDBContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("BasketDBCon")));

// add repository

builder.Services.AddScoped<IBasketRepository, BasketRepository>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (app.Environment.IsDevelopment())

{

app.UseSwagger();

app.UseSwaggerUI();

}

app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();