

به نام خدا

گزارش فاز اول پروژه پایگاه داده

عنوان پروژه: فروشگاه آنلاین

اعضای گروه:

تینا صداقت ۹۳۳۱۰۴۴

زهرا اطهاری نیکوروان ۹۳۳۱۰۲۷

سامانه فروشگاه:

مواردی که بین همه کالاها مشترک است، در جدول Utility آمده است. این جداول با توجه به مفهوم inheritance پیاده سازی شده اند که جدول Utility به عنوان parent در نظر گرفته شده است:

Utility (UtilityName, UtilityCode, UtilityCompany, UtilityCost, UtilitySale, UtilityNExist, UtilityGroup)

توجه داریم که primary key با یک خط زیر آن ها، و foreign key با رنگ قرمز نشان داده شده اند.

سه جدول HomeUtility و DigitalUtility و SportUtility که زیرگروه های جدول Utility هستند، در ادامه مشخص شده اند:

HomeUtility (UtilityName, UtilityCode, UtilityCompany, UtilityCost, UtilitySale, UtilityNExist, UtilityGroup, produceDate)

DigitalUtility (UtilityName, UtilityCode, UtilityCompany, UtilityCost, UtilitySale, UtilityNExist, UtilityGroup)

SportUtility (UtilityName, UtilityCode, UtilityCompany, UtilityCost, UtilitySale, UtilityNExist, UtilityGroup, UtilityColor)

در جدول Utility، صفتی وجود ندارد که به کمک آن بتوانیم یک صفت دیگر را به طور یکتا مشخص کنیم.

در واقع، وابستگی های تابعی ای که وجود دارد به فرم زیر است:

UtilityCode → UtilityName

UtilityCode →

که می توان از این وابستگی ها صرف نظر کرد؛ زیرا UtilityCode سوپرکلید است.

وابستگی های تابعی دیگری نیز وجود ندارد. بنابراین دارای فرم نرمال BCNF است.

مشتریان:

یکی از مفاهیمی که در این پروژه مطرح است، بحث عوامل انسانی است که در نقش‌های مختلفی نظیر مشتری، عوامل پشتیبانی، کارمند و راننده وسایل نقلیه ظاهر می‌شود. بنابراین، می‌توان یک مفهوم کلی به نام `person` معرفی کرد و نقش‌های مختلف را از آن به ارث برد:

Person (PersonID, PersonName, PersonFamilyName)

مشتریان می‌تواند بدون اشتراک یا دارای اشتراک باشند. مشتریانی که اشتراک دارند، می‌بایست یک `username` منحصر به فرد داشته باشند؛ علاوه بر این، یک اعتبار هم دارند. توجه داریم که این جدول نیز از جدول `person` باید به ارث برده شود:

RegisteredCostumer (CostumerID, PersonName, PersonFamilyName,
CostumerUsername, CostumerCredit, RegisteredDate)

UnRegisteredCostumer (CostumerID, PersonName, PersonFamilyName)

در جدول `RegisteredCostumer`، وابستگی تابعی زیر وجود دارد:

CostumerUsername → CostumerID

این وابستگی باید رفع شود. بنابراین این جدول را به دو جدول زیر تجزیه می‌کنیم:

RegisteredCostumer (CostumerID, PersonName, PersonFamilyName,
CostumerCredit, RegisteredDate)

RegisteredCostumerUsernames (CostumerID, CostumerUsername)

هر مشتری، حداقل یک شماره تلفن و آدرس دارد که برای هر کدام از آن‌ها، یک جدول مجزا در نظر می‌گیریم:

Address (City, Street, Pelak)

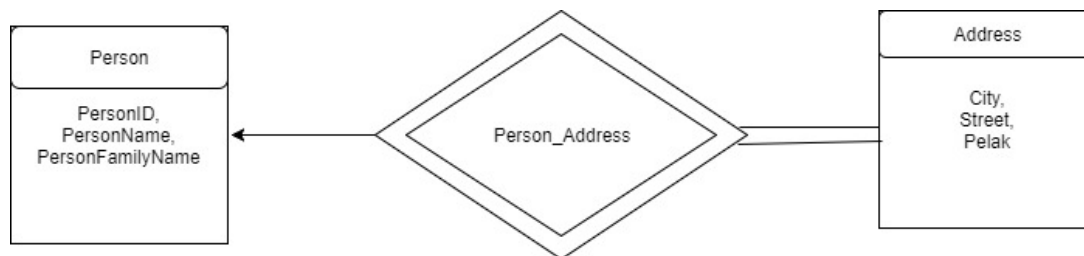
Phone (phoneNumber)

نکته قابل توجه این است که این دو جدول، برای ذخیره آدرس یا شماره تلفن هر فردی استفاده می‌شوند.

این دو از نوع weak entity هستند، زیرا primary key ندارند و discriminator در جدول Address، هر سه صفت و در جدول phone، همان phoneNumber است.

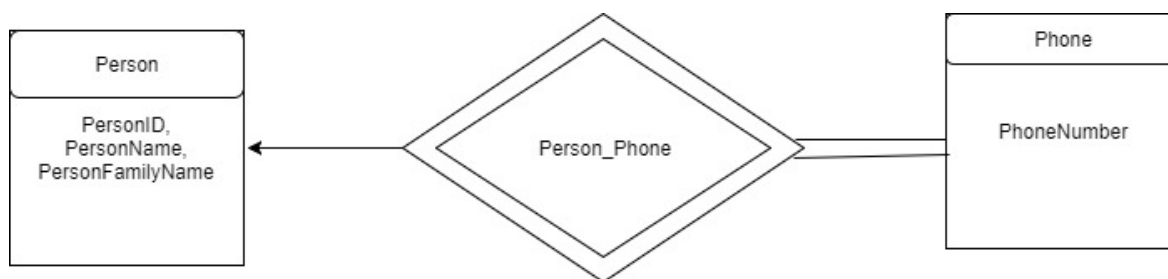
حال باید یک relationship بین جدول Address و جدول Person برقرار کنیم که primary key آن، برابر است با PersonID و City و Street و Pelak.

Person_Address (PersonID, City, Street, Pelak)



همین مسئله برای جدول Phone نیز انجام می‌دهیم. در واقع یک relationship بین جدول Phone و جدول Person برقرار کنیم که primary key آن، برابر است با PersonID و PhoneNumber.

Person_phone (PersonID, PhoneNumber)



جدول سفارش‌های مشتریان به صورت زیر است:

Order (OrderID, CostumerID, UtilityCode)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

جدول خرید به صورت زیر است. به طور خلاصه می توان گفت که این جدول، بیان می کند که هر مشتری، چه سبدی را در چه تاریخی خریداری کرده است. همچنین وضعیت پرداخت، وضعیت سبد خریداری شده و آدرس نیز مشخص است. میزان قیمت و مالیات پرداختی نیز مشخص است:

Bought (BoughtID, **CostumerID**, UtilityTotalCost, UtilityTotalTax, BoughtDateTime, PayStatus, **CostumerCity**, **CostumerStreet**, **CostumerPelak**, BoughtStatus)

BoughtStatus بیان می کند که سبد خریداری شده در کدام مرحله از فرایند ارسال می باشد (قبل از ارسال، در حال ارسال، تحویل داده شده)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

هر سبد خریداری شده، با یک ID منحصر به فرد شناخته می شود. برای این که بدانیم در هر خرید چه اقلامی خریداری شده است، باید از جدول زیر کمک بگیریم:

BoughtDetail (BoughtID, UtilityID, NumberOfUtility)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

برای ذخیره امتیازها و نظرات محصول، از جدول زیر استفاده می کنیم:

Opinion (OpinionID, **CostumerUsername**, **UtilityCode**, Comment, Rank)

با این فرض که هر کاربر برای یک محصول تنها یک بار اجازه دارد که rank وارد کند، وابستگی تابعی زیر در این جدول وجود دارد:

CostumerUsername, UtilityCode → Rank

بنابراین باید این جدول را به صورت زیر بازنویسی می کنیم:

Opinion (**CostumerUsername**, **UtilityCode**, Comment, Rank)

در این حالت، primary key، ترکیب CostumerUsername, UtilityCode می شود. لذا وابستگی تابعی - ای که در بالا ذکر کردیم، مشکل ساز نخواهد شد.

عوامل پشتیبانی:

جدول عوامل پشتیبانی هم از جدول Person به ارث می برند و به صورت زیر است:

Suporters (SupporterID, PersonName, PersonFamilyName, SupproterStatus)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

یک جدول برای رسیدگی به شکایات، دنبال کردن سفارش ها و پاسخگویی آنلاین طراحی کردیم که با توجه به TrackingKind معلوم می شود که کدام یک از این عملیات انجام می شود و به صورت زیر است:

Tracking(TrackingID, TrackingKind, CostumerID, SupporterID, BoughtID, TrackingText, TrackingAnswer, TrackingTime)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

شرکت ها:

در رابطه با شرکت ها، چند موضوع مطرح است: اول این که خود شرکت ها برای ثبت در سیستم نیاز به یک جدول دارند:

Company (CompanyID, CompanyName, RegisteredDate)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

دوم این که هر کدام از شرکت ها، فرد یا افرادی را به عنوان مسئول ارتباط با فروشگاه معرفی می کنند که باید اطلاعات این افراد را نیز در جداول زیر ذخیره کنیم. توجه داریم که مسئول ارتباط نیز از Person به ارث برده می شود:

MrLink (PersonID, PersonName, PersonFamilyName)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

برای فرد مسئول ارتباط، می‌بایست آدرس و شماره تلفن را هم ذخیره کرد که در همان جدول مربوط به تلفن که قبلاً ساختیم، ذخیره می‌کنیم:

Phone (phoneNumber)

در ادامه به یک جدول نیاز داریم که مشخص کند مسئول ارتباط هر شرکت کیست. توجه داریم که هر مسئول رابط، تنها برای یک شرکت فعالیت می‌کند:

Company_Link (CompanyID, LinkID)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

برای ذخیره شماره تلفن‌ها و آدرس‌های شرکت، از جدول‌های زیر استفاده می‌کنیم:

Company_Address (CompanyID, CompanyCity, CompanyStreet, CompanyPelak)

Company_Phone (CompanyID, CompanyPhone)

در این جدول‌ها وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF هستند.

مسئله دیگری که مطرح است، ذخیره اطلاعات کارمندان یک شرکت است:

Employee (CompanyID, EmployeeID, EmployeeName, EmployeeFamilyName)

فرض بر این است که یک کارمند نمی‌تواند عضو دو شرکت باشد؛ بنابراین کلید اصلی در این جدول EmployeeID می‌شود.

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.

برای تایید این که یکی از مشتریان فروشگاه، از کارمندان شرکت هست یا نه، می‌بایست فیلدهای جدول بالا با فیلدهایی که مشتری وارد کرده است مقایسه شود. در صورت یکسان بودن، فرد می‌تواند از تخفیفات مورد نظر استفاده کند.

حمل و نقل:

اطلاعات راننده‌ها در جدول زیر ذخیره می‌شود:

TransportMan (TransportManID, **TransportManName**, **TransportManFamilyName**,
TransportManPelak, TransportManCar, TransportManShenasname,
TransportManStatus)

در این جدول، چند وابستگی تابعی وجود دارد:

TransportManPelak → TransportManCar

فرض را بر این می‌گذاریم که هر راننده تنها یک ماشین دارد. بنابراین TransportManPelak نیز یکتا می‌شود و می‌تواند به عنوان **primary key** استفاده شود. در نتیجه جدول بالا را به صورت زیر بازنویسی می‌کنیم:

TransportMan (**TransportManName**, **TransportManFamilyName**,
TransportManPelak, TransportManCar, TransportManShenasname,
TransportManStatus)

در این جدول نیز همچنان وابستگی تابعی زیر وجود دارد:

TransportManShenasname → TransportManName, TransportManFamilyName

بنابراین باید جدول را تجزیه کنیم:

TransportMan (<u>TransportManPelak</u> , TransportManCar, TransportManShenasname, TransportManStatus)

TransportManIndividual (TransportManName , TransportManFamilyName , <u>TransportManShenasname</u>)

با این که وابستگی تابعی بالا در جدول دوم همچنان وجود دارد، اما چون TransportManShenasname در این جدول سوپرکلید است، مشکلی به وجود نخواهد آمد.

برای ذخیره آدرس و شماره تلفن راننده‌ها، از جداول اصلی مربوط به آدرس و شماره تلفن، استفاده می‌کنیم:

Address (City, Street, Pelak)

Phone (phoneNumber)

جدول زیر تعیین می‌کند که هر راننده، چه سبدی را برده است و کی تحویل داده‌است:

TransportDetail (TransportManPelak, <u>BoughtID</u> , RecievedDate)

در این جدول وابستگی تابعی نداریم. بنابراین دارای فرم نرمال BCNF است.