سوال۲) لیست تمام Entity هادر پایین آمده است اگر attributeای کلید خارجی باشد با رنگ سبز و اگر کلید اصلی باشد با رنگ قرمز نمایش داده شده است چنانچه هم کلید خارجی باشد و هم کلید اصلی باشد با همان رنگ قرمز نمایش داده شده است و مرجع آن در مقابل آن نوشته شده است (از نوشتن محدودیت های هر entity به دلیل تکراری بود و وجود آن ها در کد ایجاد پایگاه داده خودداری شده):

customer

- NationalCode
- FirstName
- LastName
- PhoneNumber
- BirthYear

address

- AddressId
- Name
- Address
- PhoneNumber
- NationalCode **(references to customer)**

food

- FoodId
- Name
- Price
- IsActive

delivery

- NatonalCode
- FrstName
- LastName

- PhoneNumber

sales_receipt

- ReceiptId
- Address
- Date
- UserId (**references to customer**)
- DeliveryId (**references to delivery**)

market

- MarketId
- Name
- IsActive

shopping_receipt

- ReceiptId
- Date
- MarketId (**references to market**)

material

- MaterialId
- Name
- Price
- IsActive

سوال۲) لیست تمام Relationship ها :

- Cus_Add : هر کاربر حداقل یک آدرس دارد و آن آدرس تنها مخصوص همان کاربر است به همین دلیل که رابطه چند به یک است از **کلید خارجی** در آدرس استفاده میکنیم
- Cus_Rec : هر رسید یا بینام است یا دارای نام است به همین سبب ممکن است با کاربر رابطه داشته باشد یا نداشته باشد به همین دلیل هم میتوان از کلید خارجی با قابلیت نال شدن استفاده کرد و هم میتوان این رابطه را به صورت یک جدول به کار برد. در چنین شرایطی باید به آمار توجه کرد که تعداد سفارشات بینام بیشتر است یا سفارشات با نام (اگر سفارشات با نام بیشتر از $\frac{1}{3}$ باشد در این حالت بهتر است از کلید خارجی nullable استفاده کرد.) ما از **کلید خارجی** nullable استفاده کردیم
- Rec_Foo : هر رسید تعدادی غذا دارد و هر غذا میتواند در تعدادی رسید باشد به همین علت لازم است که جدولی داشته باشد ساختار جدول آن به شکل زیر است:

foo_rec

- receiptId (Receipt)
- foodId (Food)

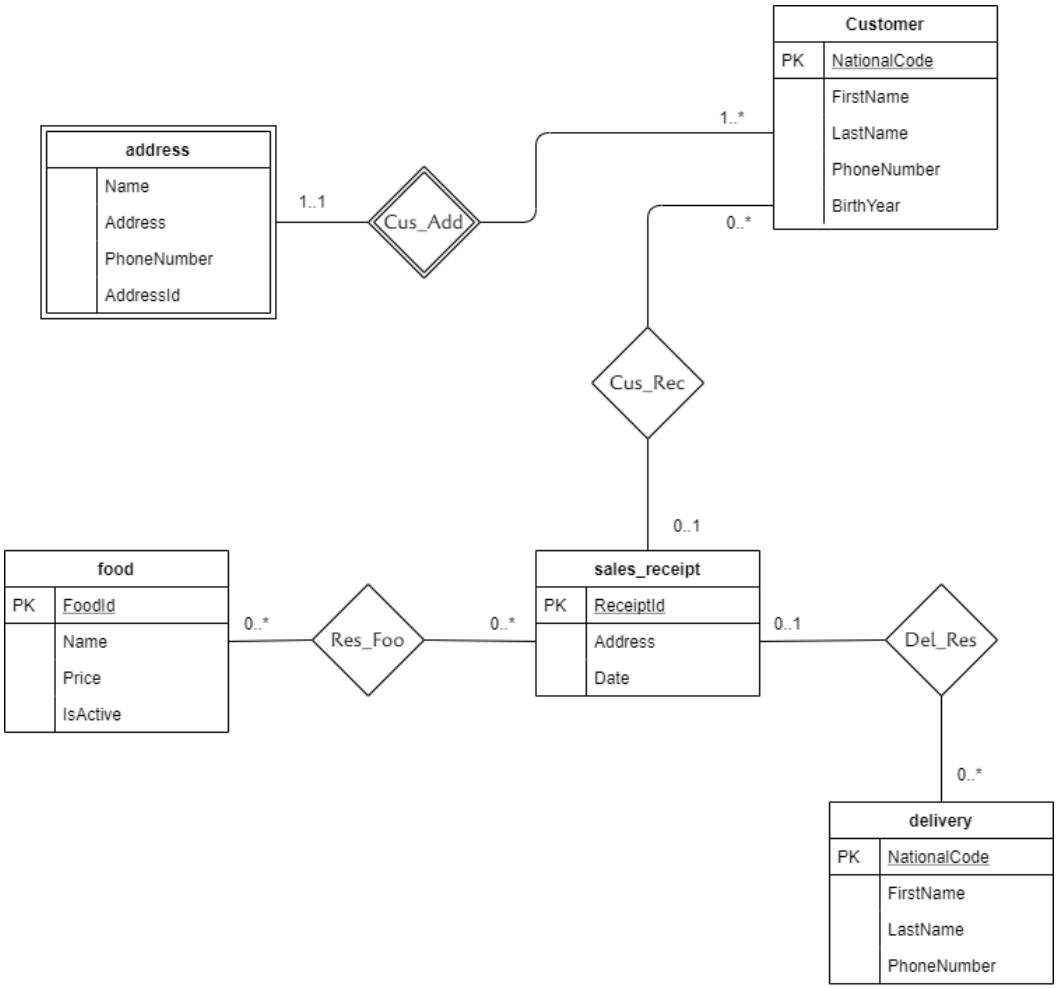
- Del_Res: میدانیم که هر رسید بیرون بر را یک پیک میبرد با توجه به این که رسید میتواند بیرون بر نباشد به همین دلیل بهتر است که از **کلید خارجی** nullable استفاده کنیم.
- Rec_Mar : میدانیم هر رسید متعلق به یک مارکت است به همین دلیل کافیست که یک **کلید خارجی** در هر رسید خرید بگذاریم که به مارکت اشاره میکند.
- Mat_Rec: میدانیم که در یک رسید ممکن است سفارش چندتا از مواد اولیه داده شده باشد به همین دلیل برای پیاده سازی این رابطه از جدول استفاده میکنیم:

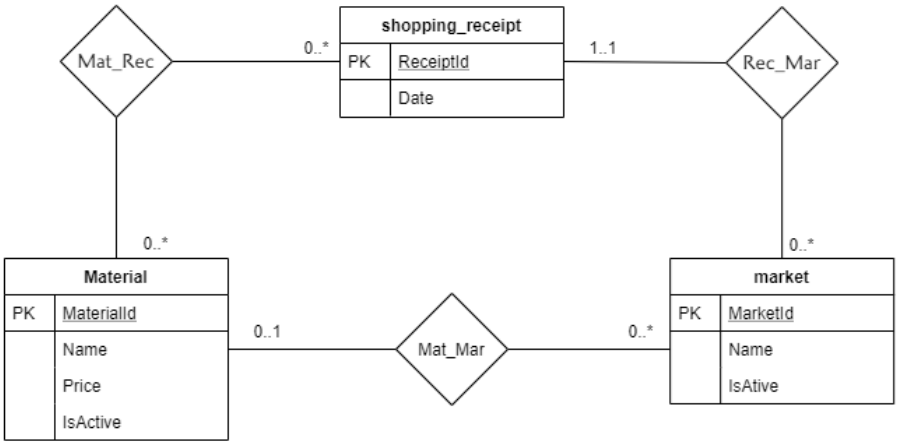
mat_rec

- receiptId (Receipt)
- MaterialId (material)

- Mat_Mar : میدانیم که هر مواداولیه متعلق به یک فروشگاه است بنابراین باید در material **کلید خارجی** ای به آن اشاره کند.

سوال۳) ERD :


Customer
PK NationalCode
FirstName
LastName
PhoneNumber
BirthYear
address
Name
Address
PhoneNumber
AddressId
1..1
Cus_Add
1..*
0..*
Cus_Rec
0..1
food
PK FoodId
Name
Price
IsActive
0..*
Res_Foo
0..*
sales_receipt
PK ReceiptId
Address
Date
0..1
Del_Res
0..*
delivery
PK NationalCode
FirstName
LastName
PhoneNumber


Mat_Rec
0..*
shopping_receipt
PK ReceiptId
Date
1..1
Rec_Mar
0..*
0..*
Material
PK MaterialId
Name
Price
IsActive
0..1
Mat_Mar
0..*
market
PK MarketId
Name
IsAtive

سوال ۵ ،۶و۷ به نظر یک سوال می‌آیند و ضمیمه شده‌اند.

توضیحات مهم :

- در این پروژه چندین مورد آمده بود که میبایستی بعد از تغییر یک ATTRIBUTE در یک جدول نباید مقدار آن در رسیدهای قبلی تغییر کند برای هندل کردن چنین موضوعی از این روش استفاده کردیم که در جدول مذکور یک ستون با نام ISACTIVE ایجاد کردیم به این صورت که چنانچه مقدار آن سطر میخواست تغییر کند به جای آپدیت کردن ویژگی های جدید را به عنوان سطری جدید INSERT میکنیم و در سطر قبلی ISACTIVE را 0 میکنیم. به این ترتیت اطلاعات گذشته را از دست نمیدهیم.
- برای هندل کردن فاکتور بیرون از این روش استفاده کردیم که به جای این که دو ENTITY فاکتور بسازیم با گذاشتن ستونی به نام ADDRESS در جدول فاکتورفروش میتوانستیم آدرس را از کاربر یا از یه کاربر مهمان بگیریم و در خود فاکتور ذخیره کنیم حال اگر فاکتور بیرون بر نبود مقدار ستون آدرس را NULL میگذاریم.
  اگر کاربر ثبت نام کرده بخواهد رسیدی سفارش دهد نام آدرس مدنظرش را (که قبلا سیو کرده است) از او میگیریم و آن آدرس را در ستون آدرس جدول فاکتور فروش ذخیره میکنیم (که اگر در آینده آدرس را عوض کرد دچار مشکل نشود)
- هر مشتری میتواند تعدادی آدرس داشته باشد و هر آدرس متعلق به یک مشتری است از آن جایی که وجود هر آدرس به وجود مشتری وابسته است؛ به همین دلیل آن را WEAK ENTITY میگیریم.
- به جای ذخیره کردن سن کاربر سال تولد او را ذخیره میکنیم.
- برای مواد اولیه و غذا دو جدول مجزا گرفتیم زیرا که هر کدام از مواد اولیه میبایست به یک فروشگاه متعلق باشد و غذاها نیازی نیست که به فروشگاه متعلق باشد. با جدا کردن این دو جدول از همدیگر، تکرار کاهش یافته است.
- همه کلیدهای خارجی NULLABLE هستند و وقتی که رفرنس آن ها پاک شود NULL میشوند. این موضوع تنها برای ساده سازی پیاده سازی بود.