

# به نام خدا

---

طراحی پایگاه داده برای فروشگاه آنلاین (هایپرمارکت)

استاد مربوطه : دکتر شاه بهرامی



## فهرست

1.	معرفی اعضای گروه	4
2.	وظایف اعضا	5
3.	مقدمه	6
4.	تاریخچه	7
5.	هدف و مزایا	7
6.	نیازمندی ها	7
6-1.	برخی از رایج ترین کاربردهای پرسشنامه	7
6-2.	دو شیوه متفاوت استفاده از پرسشنامه در فضاهای آکادمیک و دانشگاهی	8
6-3.	مزایای استفاده از پرسشنامه‌های آنلاین	8
6-4.	روش کار ما	9
6-5.	بررسی سوالات و نتایج آنها	10
7.	نمودار ها	20
7-1.	نمودار ER	20
7-2.	نمودار Relational	21
8.	بررسی و تشریح جداول	22
8-1.	جدول supplier	22
8-2.	جدول staff	23
8-3.	جدول customer	24
8-4.	جدول و روابط محصولات	26
8-5.	جدول product review	27
8-6.	جدول product	28
8-7.	جدول category	29
8-8.	جدول product_meta	30
8-9.	جدول و روابط سفارشات	32
8-10.	جدول discount	34
8-11.	جدول order items	35
8-12.	جدول quantity	36
8-13.	جدول order	37

38	9. نرمال سازی بانک اطلاعاتی .....
38	9-1. دلایل نرمال سازی .....
38	9-2. فرم های نرمال در پایگاه داده .....
40	9-3. معایب نرمال سازی .....
41	9-4. نرمال سازی جداول سامانه .....
41	9-4-1. جدول customer .....
42	9-4-2. جدول staff .....
43	9-4-3. جدول supplier .....
45	9-4-4. جدول order items و order .....
47	10. کد پیاده سازی پایگاه داده .....
51	11. توضیح کد پیاده سازی .....
58	12. کلید اصلی و فرعی .....
61	13. الحاق داده های واقعی .....

## معرفی اعضای گروه

سر گروه :

- نگین بنای شاهانی (960122680031)

اعضای گروه :

- سحر فخریه کاشان (960122680012)
- یاسمن چگینی (960122681001)
- پری ناز ملت دوست (960122680011)

## وظایف اعضا :

در مرحله اول نیازمندیهای پروژه پایگاه داده را تعیین کردیم. برای تعیین نیازمندی های پروژه ابتدا فروشگاه های آنلاین موجود را بررسی کردیم و همچنین یکی از اعضای گروه با پرس و جو از چندین نفر از کارکنان فروشگاه کوروش اطلاعاتی در مورد فروشگاه به دست آورد. سپس به طراحی یک پرسش نامه ی آنلاین بر اساس اطلاعاتی که به دست آورده بودیم ، پرداختیم.

بعد از طراحی پرسش نامه ، لینک پرسش نامه را از طریق کارمندان فروشگاه و شبکه های اجتماعی به صورت رندوم در اختیار افراد قرار دادیم. و از طریق خود وب سایت میزان مشارکت و پرسش نامه های پر شده را زیر نظر داشتیم. سپس با توجه به پاسخ های داده شده و رسم نمودار به تحلیل نیازمندی ها پرداختیم و موجودیت ها و صفاتی که هر موجودیت باید داشته باشد را تعیین کردیم.

برای تک تک موجودیت ها نمودار ER و relational را رسم کردیم تا روابط بین موجودیت ها را بتوانیم به خوبی تعیین کنیم. و برای جلوگیری از افزونگی و رفع مشکلات ساختاری و آنومالی جداول را تا سطح سوم نرمال سازی رساندیم. سپس بر اساس جدول ها ، موجودیت ها و روابط بین آن ها به پیاده سازی یک پایگاه داده در فرمت SQL پرداختیم.

به تمامی جداول داده های با معنی الحاق کردیم تا پایگاه داده ی خود را قبل از ارائه تست کرده باشیم ، بعد از اتمام مراحل پروژه به کمک یکدیگر گزارشی از کارهای انجام شده و مراحل طی شده در پروژه نوشتیم.

نگین بنای شاهانی : بررسی فروشگاه های آنلاین موجود، طراحی پرسش نامه ، پخش پرسش نامه ، تعیین موجودیت ها و صفات ، نرمال سازی تا سطح سوم ، نوشتن گزارش، هماهنگی بین اعضا.

سحر فخریه کاشان : بررسی پرسش نامه، تعیین موجودیت ها و صفات بر اساس پرسشنامه و نیازمندی ها، رسم نمودار های ER و relational برای جداول ، استخراج شبه کد براساس نمودار relational ، نوشتن گزارش.

یاسمن چگینی : پخش پرسش نامه ، پرس و جو از کارمندان افق کوروش، تحلیل پرسش نامه برای طراحی نیازمندی ها ، طراحی کد پایگاه داده، الحاق کردن داده های بامعنی به جداول و نوشتن گزارش.

پری ناز ملت دوست: پخش پرسش نامه ، تحلیل پرسش نامه برای طراحی نیازمندی ها ، پیاده سازی پروژه ی پایگاه داده به زبان SQL ، نوشتن گزارش.

## مقدمه

پایگاه داده مجموعه ای است از داده ها که معمولاً به صورت الکترونیکی ذخیره می شوند. پایگاه داده ها تا حد امکان به صورتی طراحی می شوند که ذخیره و دست یابی به اطلاعات موجود در آن ها آسان باشد. طراحی یک پایگاه داده مناسب از اهمیت زیادی برخوردار است چرا که پایگاه داده ها مقادیر زیادی از اطلاعات را ذخیره و مدیریت می کنند؛ که این کار با استفاده از ابزار های دیگری ممکن نیست یا به سادگی این کار را انجام نمی دهند.

از دلایل دیگر اهمیت پایگاه داده می توان به دقیق بودن آن اشاره کرد به این معنی که درستی اطلاعات موجود در یک پایگاه داده در اغلب موارد تضمین شده است. از طرف دیگر پایگاه داده از روش های مختلفی امنیت داده ها را فراهم می کند و فقط به کاربران مجاز اجازه دسترسی به پایگاه داده را می دهد. پایگاه داده یکپارچگی داده ها را حفظ می کند و دسترسی و تحقیق در داده ها در پایگاه داده بسیار آسان می باشد؛ این کار با استفاده از (Data Query Language) DQL انجام می شود. که امکان جستجو در پایگاه داده و انجام تحقیق ها و محاسبات بر روی آن ها را فراهم می آورد.

تا به اینجا پایگاه داده، کاربرد های آن و اهمیت آن را بررسی کردیم. حال پروژه و اهمیت پایگاه داده در آن را بررسی می کنیم. می دانیم که خرید اینترنتی (Online Shopping) نوعی خرید بوده که از طریق فروشگاه های اینترنتی صورت می گیرد که از جمله مهم ترین فروشگاه های اینترنتی می توان به آمازون (Amazon) و ای بی (eBay) اشاره کرد.

عملکرد فروشگاه های اینترنتی به این صورت است که فروشنده ها لیست و مشخصات کالا های خود را در سایت (سایت فروش اینترنتی) قرار می دهد و خریداران با مراجعه به این سایت و رویت مشخصات و ویژگی های کالا اقدام به ثبت سفارش خرید خود از اجناس فروشگاه می کنند. سفارشات پس از بررسی و تایید نهایی توسط مسئولان سایت به روش های مختلفی به دست ثبت کننده سفارش می رسد، که از جمله ی این روش ها می توان به پست یا باربری اشاره کرد.

هزینه مربوط به سفارش هم به روش های مختلفی قابل پرداخت است که از جمله ی این روش ها می توان به پرداخت در زمان ثبت سفارش و یا پرداخت درب منزل اشاره کرد. امروزه تجارت الکترونیک در سر تا سر دنیا بسیار رواج پیدا کرده است و خرید اینترنتی در جهان روز به روز به صرفه تر و مناسب تر از گذشته شده تا جایی که اغلب مشتریان ترجیح می دهند خرید خود را از اینترنت انجام دهند. طبق بررسی های انجام شده فروشگاه های موفق اینترنتی ابتدا باید اعتماد مشتری را جلب کنند که بهترین قیمت، همراه با کیفیت بالا موجب جلب رضایت مشتری است.

به صورت اجمالی اگر بخواهیم مزیت های فروشگاه اینترنتی نسبت به فروشگاه سنتی را بررسی کنیم باید بگوییم فروشگاه های اینترنتی بر خلاف فروشگاه های سنتی محدود به منطقه یا شهر خاصی نیستند پس از مزیت عدم وجود محدودیت جغرافیایی برخوردارند. از طرف دیگر در فروشگاه های اینترنتی اکثراً محدودیت زمانی وجود ندارد که این محدودیت هم شامل فروشنده و هم خریدار می شود. در این فروشگاه ها هزینه هایی مثل محل فروش و فروشنده و ویتترین مغازه وجود ندارد و کالا به دلیل حذف واسطه ها برای مشتری ارزان تر تمام می شود.

با بررسی موارد گفته شده دریافتیم در فروشگاه های اینترنتی دخالت انسان هرچه کم تر شده و اطلاعات به صورت دستی به ندرت ذخیره می شوند؛ اینجاست که اهمیت وجود یک پایگاه داده مناسب جهت ذخیره میزان فروش، مشخصات خریداران، موجودی انبار و تمام اطلاعات دیگر خودنمایی می کند. در این پروژه به طراحی یک پایگاه داده مناسب برای فروشگاه اینترنتی می پردازیم.

## تاریخچه

تجارت الکترونیک اولین بار در سال 1991 توسط تیم برنرز لی ( John Berners-Lee ) پایه گذاری شد. در سال 1994 بانکداری آنلاین ایجاد شد و شرکت پیتزا هات اولین فروش اینترنتی خود را آغاز کرد. در همان سال بود که شرکت نت اسکپ ارتباط و امنیت لایه انتقال (SSL) را ابداع کرد. خرید اینترنتی به وسیله ی این پروتکل امکان پذیر شد؛ همچنین در همان سال کمپانی آلمانی اینتر شاپ (Inter Shop) اولین سیستم فروش اینترنتی خود را ایجاد کرد. یک سال بعد یعنی در سال 1995 شرکت آمازون و دو سال بعد یعنی در سال 1996 شرکت eBay فعالیت خود را آغاز کردند.

## هدف و مزایا

با توجه به شرایط ایجاد شده و وجود اپیدمی و قرنطینه، فلسفه ی در خانه ماندن و خرید آنلاین بیشتر از هر زمانی دارای اهمیت شده است و خرید مواد غذایی (خرید از هایپر مارکت ها) هم در اولویت همه خرید ها است. برای کم کردن حجم حضور مردم در فروشگاه های بزرگ و جلوگیری از رشد بیماری لازم دیدیم برای یکی از شرکت های بزرگ ایران ( مثلاً افق کوروش ) سامانه ای پیاده سازی کنیم که بتوان از منزل و با امنیت کامل خرید اینترنتی انجام داد، همچنین اطلاعات فروشگاه به صورت دسته بندی شده و مرتب باشد تا کارکنان ها شعبه به اطلاعات محصولات موجود در فروشگاه ها ی دیگر و انبار دسترسی داشته باشند و نیاز های خود را به راحتی برطرف کنند و از حجم کارهای فیزیکی و هماهنگی های مربوطه کاسته شود. تا به نوبه ی خود تلاش لازم را در جهت یاری رساندن در این شرایط انجام دهیم.

## نیازمندی ها

در این پروژه ما از یک پرسشنامه ساده برای جمع آوری و تایید برخی از نیاز مندی ها استفاده کرده ایم تا بتوانیم به این وسیله فرض های اولیه خود را تایید کرده یا آنها را تغییر دهیم. برای این پروژه ما از وبسایت پرس آنلاین برای طراحی پرسشنامه استفاده کرده ایم که به علت محدودیتی که ظرفیت 100 نفره ای که سایت برای ما قرار داده بود پرسشنامه را بیش از این پخش نکردیم.

## برخی از رایج ترین کاربردهای پرسشنامه

کاربردهای پرسشنامه ها، محدود و مشخص نیست و مجموعه گسترده ای از کاربردها را می توان برای پرسشنامه ها تصور کرد. در اینجا صرفاً به برخی از این کاربردها می پردازیم:

برخی پرسشنامه ها، برای سنجش نگرش و ترجیحات مخاطب، در یک زمینه مشخص مورد استفاده قرار می گیرند. به عنوان مثال:

- پرسشنامه برای اطلاع از محبوب ترین برنامه های تلویزیونی یا سایت های اینترنتی
- پرسشنامه برای تشخیص اینکه تغییر بسته بندی و قیمت، تا چه حد تصمیم های خرید مشتری را تغییر می دهد
- پرسشنامه برای انتخاب کتابهایی که هر فرد، برای خرید و مطالعه توصیه می کند

برخی پرسشنامه های تحقیقاتی، با هدف سنجش رابطه بین دو یا چند متغیر مورد استفاده قرار می گیرند. به عنوان مثال:

- سنجش رابطه جنسیت و درآمد
  - سنجش رابطه بین تحصیلات دانشگاهی و احترام به قانون
  - سنجش رابطه بین سن و میزان مطالعه
  - سنجش رابطه بین ماه تولد و ویژگی های شخصیتی یا سلامت
- پرسشنامه هایی که برای سنجش فکت ها (Facts) مورد استفاده قرار می گیرند. به عنوان مثال:
- وقتی شما از یک فروشگاه خرید می کنید، از شما می پرسند که از چه طریق با آن فروشگاه آشنا شده اید.
  - در هنگام مراجعه به یک سمینار، ممکن است تحصیلات شما و سن و جنسیت شما، یا شغل شما، پرسیده و ثبت شود.

## دو شیوه متفاوت استفاده از پرسشنامه در فضاهای آکادمیک و دانشگاهی :

بخشی از تحقیقات دانشگاهی در سراسر جهان، مشخصاً با هدف طراحی و ساخت پرسشنامه انجام می شود. مثلاً یک یا چند محقق، ماه ها و سال ها وقت می گذارند و مثلاً پرسشنامه ای را طراحی می کنند که معتقدند می تواند برای سنجش شادمانی یا سنجش رضایت شغلی یا کیفیت زندگی کاری مورد استفاده قرار بگیرد. بخشی دیگر از تحقیقات دانشگاهی، از پرسشنامه های استاندارد (Standard Questionnaires) استفاده می کنند. در واقع، این محققان، به سراغ دستاوردهای گروه اول می روند و پس از اطمینان از صحت و کارایی و اعتبار و روایی پرسشنامه، از آن برای سنجش یک یا چند ویژگی در جامعه آماری مد نظر خود استفاده می کنند. عمده ی کاربردهای پرسشنامه در تحقیقات دانشگاهی را می توان در یکی از دو گروه فوق طبقه بندی کرد. البته فعالیت های ارزشمند دیگری هم در ارتباط با پرسشنامه ها وجود دارد و آن، محققانی هستند که به صورت دائمی و گسترده، قابل اعتماد بودن و اتکا به پرسشنامه ها را در شرایط مختلف می سنجند.

## مزایای استفاده از پرسشنامه های آنلاین

احتمالاً مهم ترین و بارزترین مزیت استفاده از پرسشنامه های آنلاین صرفه جویی در هزینه، وقت، و نبود مرزهای جغرافیایی و فیزیکی است. در پرسشنامه آنلاین، برخلاف پرسشنامه چاپی، نیازی به چاپ کردن پرسشنامه ها نیست و نیازی نیست که برای رساندن پرسشنامه ها به پاسخ دهندگان هزینه ای صرف شود.



همچنین، این پرسشنامه‌ها می‌توانند مقادیر زیادی داده در زمانی نسبتاً محدود گردآوری کنند. از دیگر مزایای بسیار مهم این نوع پرسشنامه‌ها ارائه‌ی اطلاعات آماری و نتایجی است که توسط برنامه‌های تعبیه‌شده در پایگاه فراهم می‌شود، بنابراین پژوهشگر می‌تواند درست پس از اتمام گردآوری داده به تجزیه و تحلیل آن بپردازد. همان‌طور که اونیل در سال 2004 اشاره می‌کند، شواهدی وجود دارد دال بر اینکه پرسشنامه‌های آنلاین قابلیت گردآوری حجم بیشتری داده نسبت به انواع چاپی را دارند. او همچنین به حداقل رساندن خطاها را مزیت دیگر این ابزار می‌داند. همان‌طور که ذکر شد، در این پرسشنامه‌ها امکان آن وجود دارد که هنگام تکمیل نکردن و یا اشتباه پاسخ دادن به پرسشنامه پیغام‌هایی به پاسخگو ارائه شود که نسبت به رفع خطاها اقدام کند.

## روش کار ما

ما با بررسی فروشگاه‌های اینترنتی متعدد و بررسی ساختار ظاهری آنها و همچنین بررسی تعدادی پایگاه داده مشابه اینترنتی نیازهای اولیه و مشترک این پایگاه داده را تا حد ممکن تشخیص دادیم و در مورد مواردی که می‌توانستند دارای حالات متعدد باشند یا از آنها مطمئن نبودیم از طراحی پرسشنامه استفاده کردیم.

تلاش ما بر این بود تا با کم نگه داشتن تعداد سوالات و کوتاه نگه داشتن صورت آنها افراد بیشتری را تشویق به پاسخ‌گویی به آنها کنیم. پس از طراحی پرسشنامه لینک آن به چندین نفر از کارنان فروشگاه در شعب مختلف داده شد و از آنها درخواست شد که به صورت رندوم لینک پرسشنامه را به مشتریان خود بدهند و از آن‌ها بخواهند که به پرسشنامه پاسخ دهند. و ما هر روز از طریق سایت پرس آنلاین از تعداد افرادی که پرسش‌نامه را دیدند و به آن پاسخ می‌دادند با خبر می‌شدیم.

لینک پرسشنامه : <https://survey.porsline.ir/s/2Z1jlqC>

در مجموع 96 پاسخ دریافت شد. در زیر میزان درصد پاسخ‌دهی و طریقه‌ی استفاده کاربران از لینک را نمایش دادیم:



در ادامه نتایج را تحلیل کرده و از آنها در تشخیص نیازمندی‌ها استفاده کردیم. لازم به ذکر است که به علت تعداد کم پاسخ‌ها تمایل جواب‌ها به یک سمت زیاد مشاهده می‌شود که به احتمال زیاد اگر داده‌های بیشتری در دسترس بود امکان پراکندگی جواب‌ها بیشتر می‌شد.

لینک نتیجه‌گیری نهایی : <https://survey.porsline.ir/r/i6Agyqf>

## صفحه ی اول پرسش نامه :



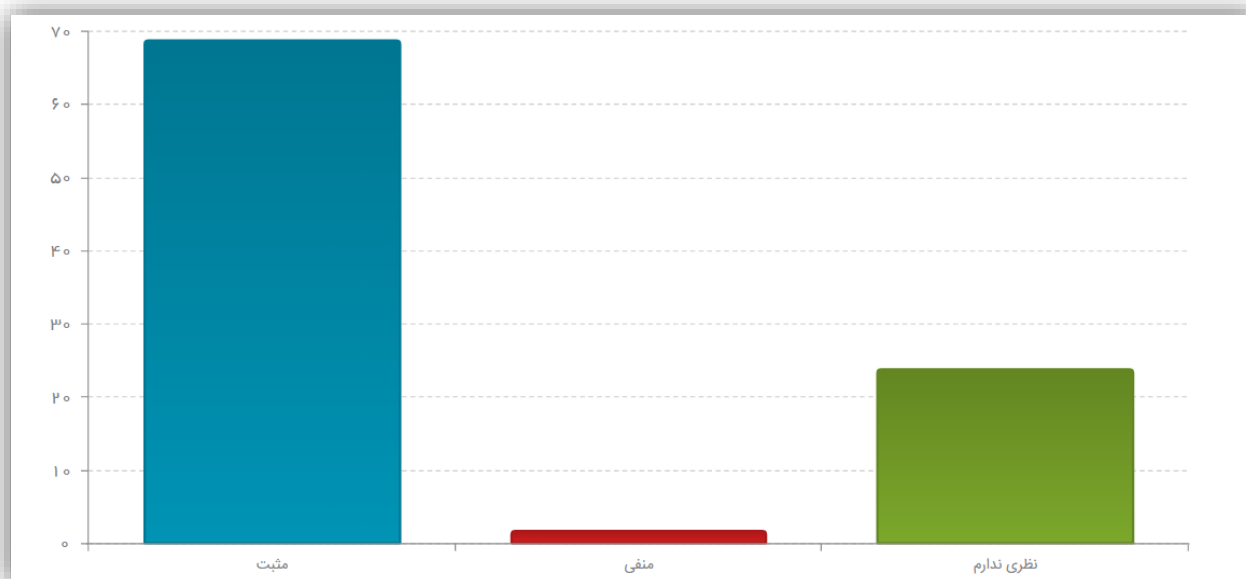
از وقتی که به تکمیل این پرسشنامه اختصاص می دهید، متشکریم. پاسخ دهی به سوالات 10 دقیقه زمان می برد.

احتمالا شما هم تجربه ی خرید کردن از شرکت های بزرگ زنجیره ای را داشته اید. معطل شدن در صف های طولانی خرید ، هزینه ی اضافی برای رفت و آمد، از دست دادن زمان و همچنین سختی پیدا کردن محصول مورد نظر میان قفسه های فروشگاه همه و همه دست به دست دادند که به طراحی یک فروشگاه زنجیره ای آنلاین روی آوریم. تست پیش روی شما برای ارزیابی نیازمندی های این پروژه ی دانشجویی می باشد.

شروع

## بررسی سوالات و نتایج آنها

### 1. اولین واکنش شما نسبت به این نرم افزار چیست؟



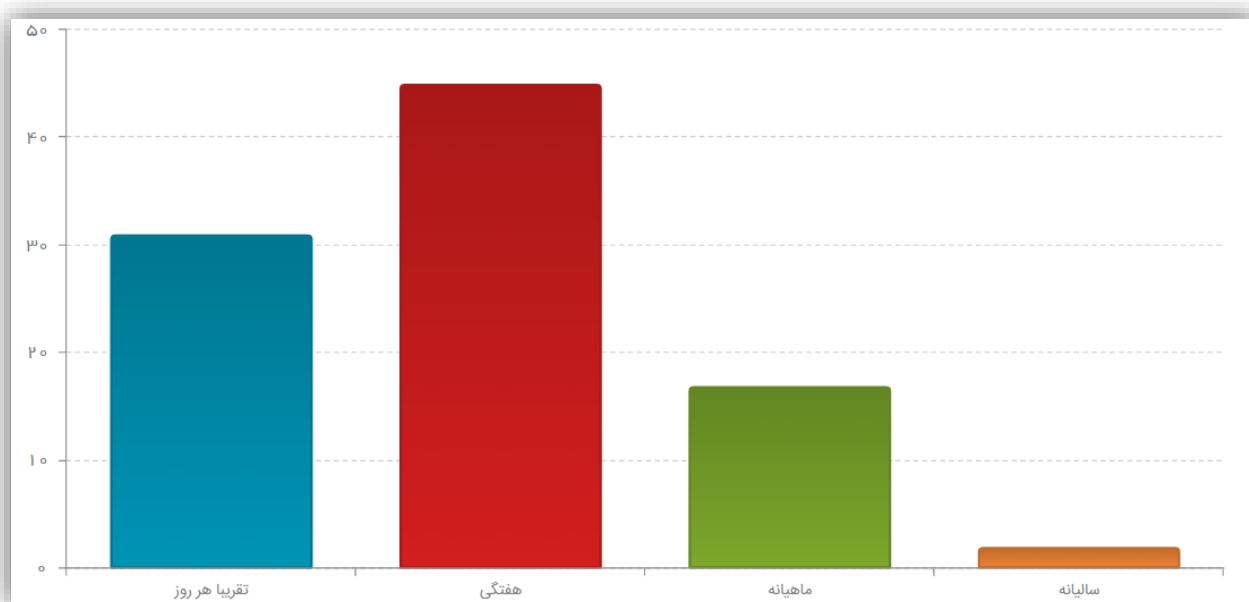
در ابتدا و پیش از رفتن به سراغ سوالات اصلی این سوال قرار داده شده تا بدانیم که آیا طراحی چنین سیستمی نیاز هست یا نه. همانطور که از نتایج مشاهده میشود بیشتر پاسخ ها مثبت بوده که فرض اولیه مبتنی بر نیاز به طراحی چنین سیستمی را تایید میکند.

2. اگر این نرم افزار موجود بود، چقدر احتمال داشت آن را دانلود کنید و استفاده ی مداوم داشته باشید؟



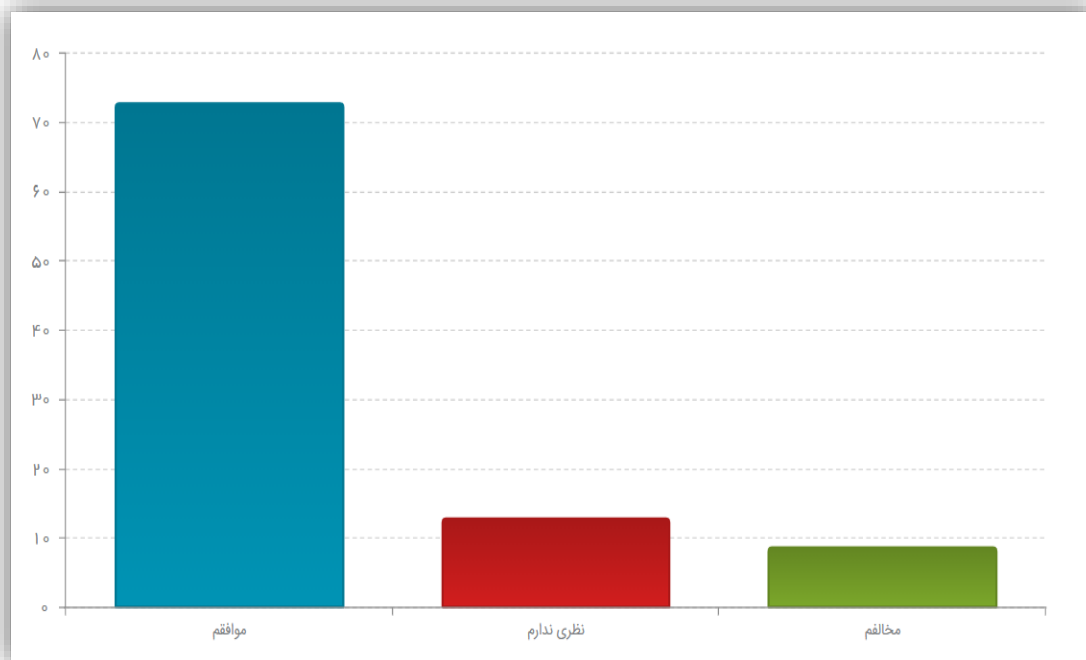
این سوال مبنی بر فرض نیاز به سیستمی برای جایگزینی روش سنتی پرسیده شده و مشاهده میشود که این نرم افزار میتواند کاربرد لازم را داشته باشد و باعث سود آوری برای صاحبان پروژه باشد. در واقع این سوال برای تایید و اطمینان خاطر صاحبان پروژه از این سرمایه گذاری خود است.

3. هر چند وقت یک بار به فروشگاه برای خرید سر می زنید؟



روش های متعدد برای خرید وجود دارد ، از جمله خرید از سوپر مارکت و مغازه های کوچک و فروشگاه بزرگ. امروزه به علت دسترسی راحت به فروشگاه های بزرگ و تنوع جنس در این فروشگاه ها تمایل به این نوع خرید بیشتر شده است. همچنین افراد مختلف ترجیحات مختلفی برای خرید دارند که در تعداد دفعات مراجعه به فروشگاه ها تاثیر گذار است. همانطور که از پاسخ ها مشاهده میشود بیشتر تمایلات به سمت چپ است که نشان دهنده مراجعات متعدد به فروشگاه است. مسلماً این نوع مراجعات میتواند باعث هدر رفتن وقت و حتی هزینه برای مراجعه باشد ، مانند هزینه سوخت و یا تاکسی برای رفتن به فروشگاه و یا وقت لازم برای پیدا کردن اجناس مورد نظر یا صف برای حساب کردن محصول. این نمودار نشان دهنده این است که این نیاز به یک سیستم جایگزین میتواند در زندگی استفاده کنندگان تاثیر گذار باشد.

4. به نظرم خرید آنلاین باعث صرفه جویی در وقت و هزینه ام می شود.



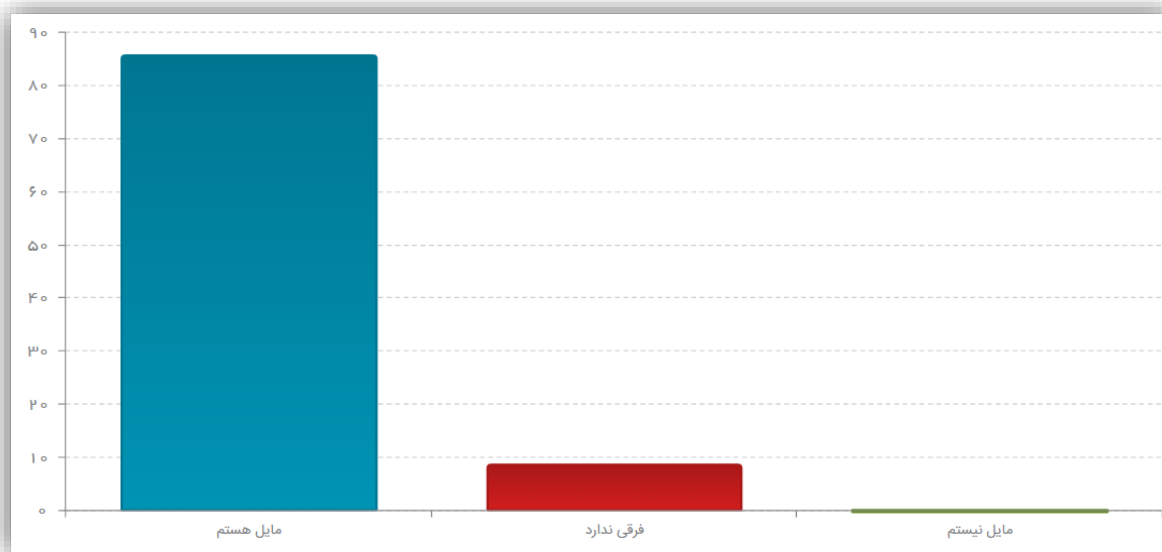
این سوال تایید کننده آن است که مشتریان نیز با فرض اولیه مبنی بر تلف شدن وقت و هزینه در روش سنتی موافق هستند. بیش از 70 درصد رای دهندگان این موضوع را تایید کرده اند.

5. اگر برنامه ای مورد اعتماد وجود داشته باشد، چقدر حاضرید به جای روش سنتی از آن استفاده کنید؟



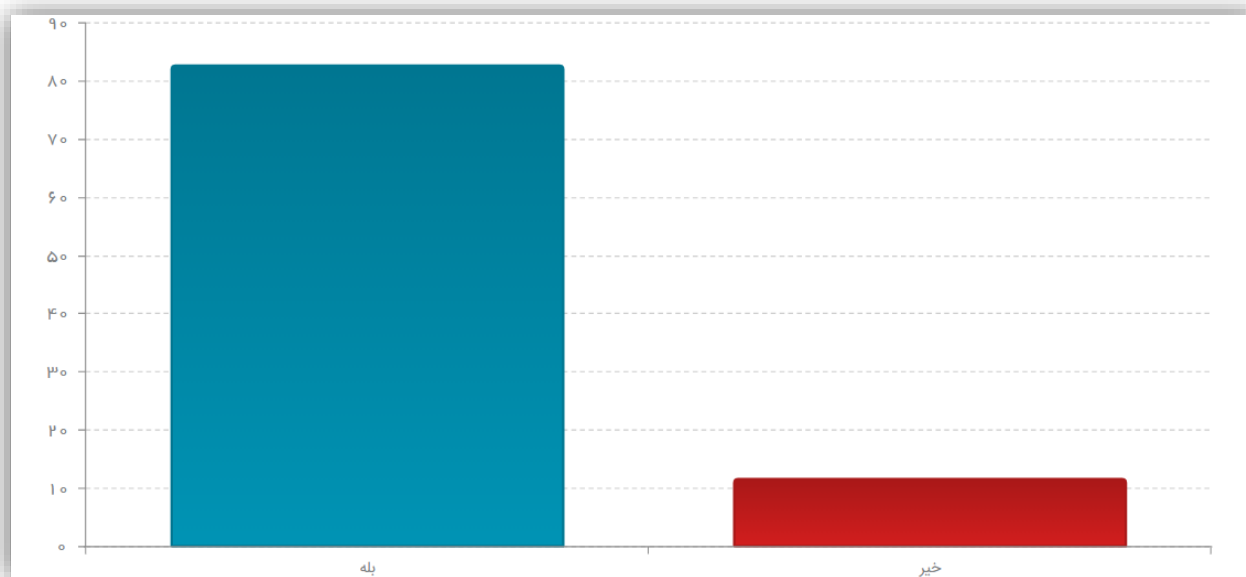
این سوال نشان دهنده آن است که روش جایگزین پیشنهادی که طراحی نرم افزار فروشگاه بود میتواند پاسخ خوبی برای ضعف های روش سنتی باشد.

6. آیا مایل هستید که به هنگام خرید آنلاین جزئیات ریز محصول مورد نظرتان را ببینید؟



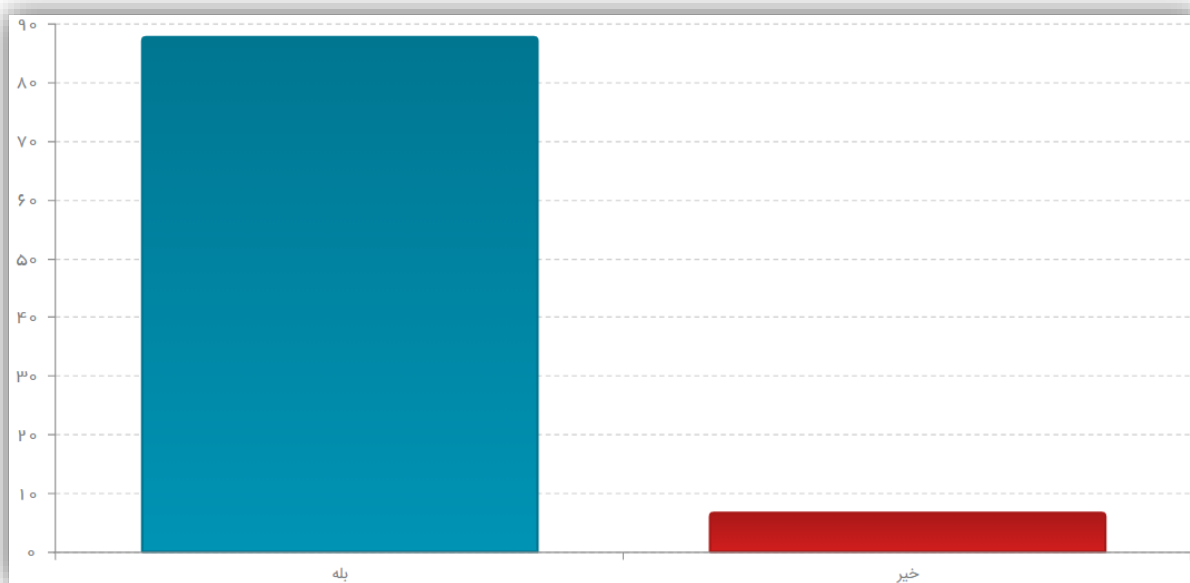
یکی از فرض های اولیه ما برای طراحی این پایگاه داده نیاز به نشان دادن جزئیات محصولات بود. در هنگام خرید اینترنتی ممکن است کاربران صرفاً تمایل به دیدن نام و قیمت و تولید کننده و سایر مشخصات کلی باشند که فقط اطلاعات محصول را تایید میکند و جزئیات برای آنها مهم نباشد. از طرف دیگر ممکن است کاربران به وجود جزئیات مانند ابعاد و وزن و مواد تشکیل دهنده و سایر مشخصات جزئی تمایل داشته باشند. مشاهده میشود که بیشتر افراد به دانستن این جزئیات تمایل داشته اند پس ما یک جدول با نام `product_meta` طراحی کردیم که پاسخگوی این نیاز باشد.

7. آیا ترجیح می دهید که به هنگام خرید موجودی یک محصول در انبار را مشاهده کنید؟



دانستن موضوعی مانند تعداد موجودی یک محصول بستگی به سلیقه کاربران دارد و بر حسب نتایج به دست آمده ما ویژگی number را در جدول quantity تعریف کردیم که در واقع تعداد باقی مانده یک محصول را نشان میدهد.

8. این که بتوانید نظرات بقیه ی کاربران را در مورد محصول مورد نظر را ببینید ، می تواند در تصمیم نهایی شما تاثیر گذار باشد.



بهره بری از نظرات دیگر مشتریان به عنوان یک نیازمندی شناخته شد و بر اساس نتایج به دست آمده و تایید این نیازمندی جدول product\_review ساخته شد.

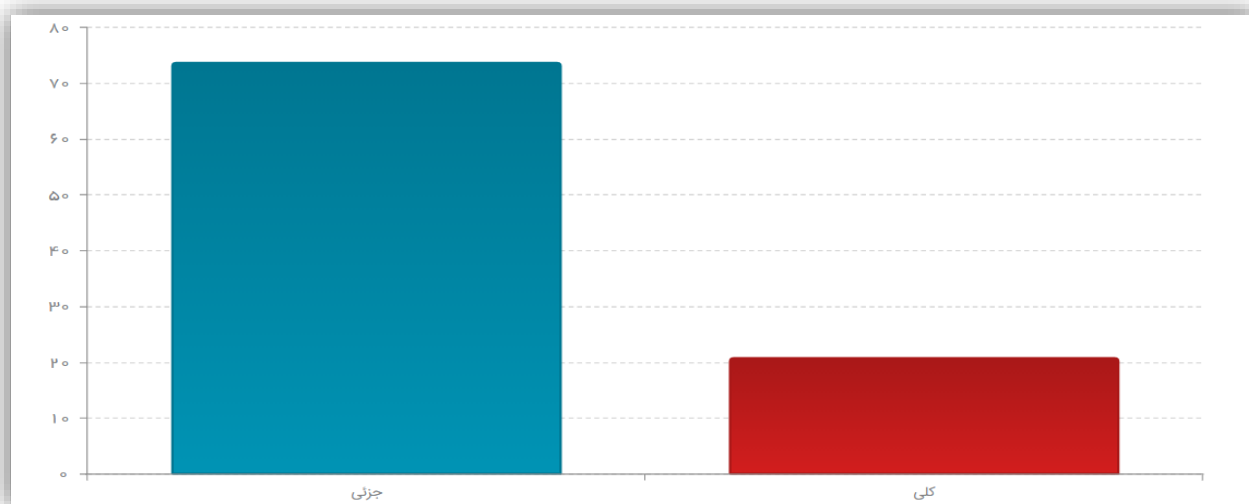
9. اگر موقع خرید بتوانید هشدار های حساسیت و نحوه ی استفاده از محصول را ببینید. چقدر برای شما مفید خواهد بود؟



یکی از ویژگی هایی که میتواند برای یک محصول وجود داشته باشد هشدار ها هستند. هشدار هایی مانند ممنوعیت استفاده برای گروهی از افراد مانند برخی مواد خوراکی که افراد دیابتی یا باردار نباید استفاده کنند و یا حتی موادی که میتوانند باعث حساسیت شوند مانند بادام در افرادی که آلرژی دارند و یا هشدار های استفاده مانند در صورت مشاهده یک عارضه جانبی مصرف قطع یا به

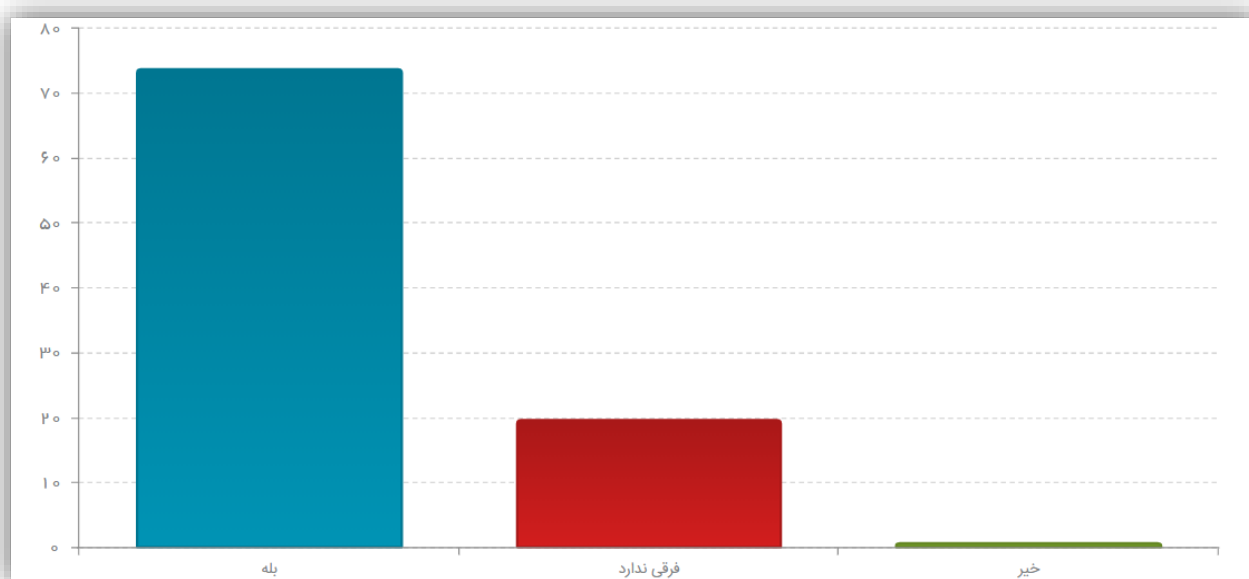
پزشک مراجعه شود و یا هشدار می مانند عدم استفاده در ناحیه چشم یا زخم برای محصولات پوستی می تواند بر خرید افراد تاثیر گذار باشد و با تایید رای دهندگان ما ویژگی warning را در جدول product\_meta قرار دادیم.

#### 10. آیا ترجیح می دهید دسته بندی محصولات به صورت کلی باشد یا جزئی؟



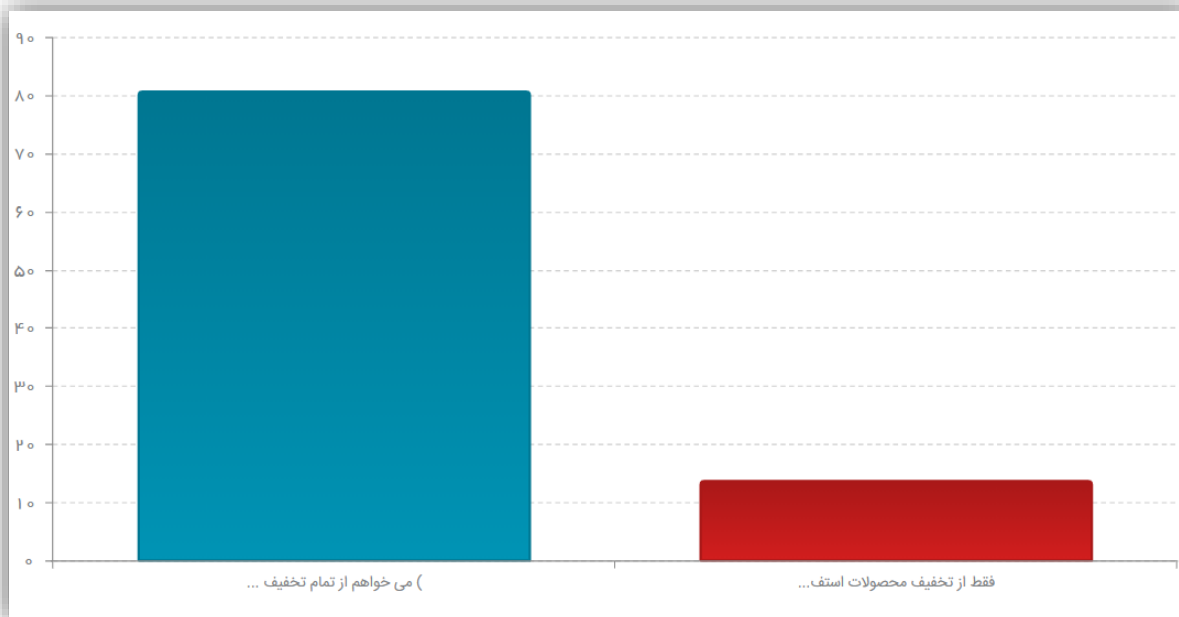
دسته بندی محصولات می تواند به صورت کلی باشد به این گونه که به عنوان مثال ما گروه هایی مانند شیرینی ها و تنقلات و مواد پروتئینی و غیره را داشته باشیم و یا اینکه به صورت جزئی باشد به این گونه که به عنوان مثال ما در گروه خوردنی ها گروه شیرینی جات و سپس گروه شکلات را در آن داشته باشیم. با توجه به نتایج ما دسته بندی را به صورت جزئی قرار دادیم.

#### 11. آیا برای شما مهم است که ابعاد محصول مورد نظر را قبل از خرید بدانید؟



ابعاد یک محصول میتواند در خرید ان تاثیر گذار باشند. مثلا ممکن است شخصی بخواهد شامپو بخرد که برای چندین ماه کافی باشد پس باید ابعاد محصول را بداند تا بتواند محصول مناسب را انتخاب کند. همچنین در مواردی مانند مواد شوینده یا بهداشتی نیاز اطلاع از این موضوع بیشتر میشود. پس بر حسب نتایج ما ویژگی size را در جدول product\_meta قرار دادیم.

**12. آیا دوست دارید تخفیف ها فقط برای محصولات باشن یا مایل هستید که از تخفیف مخصوص کاربران و محصولات خاص اطلاع داشته باشید.**



تخفیف میتواند به دو صورت اعمال شود. حالت اول این است که فقط بر روی هر محصول اعمال شود و حالت دوم این است که انواعی از تخفیف ها وجود داشته باشند که بتوانند بر روی محصول و یا سفارش اعمال شوند. بر حسب ترجیح رای دهندگان ما حالت دوم را به صورت جدول discount قرار دادیم که به سفارش و محصولات و اجزای سفارش متصل است.

**13. برخی از محصولات خاص در یک محدوده ی زمانی خاص در فروشگاه ما به فروش می رسد، چقدر مایل هستید این محصولات را در قسمت پیشنهادی ها مشاهده کنید؟**

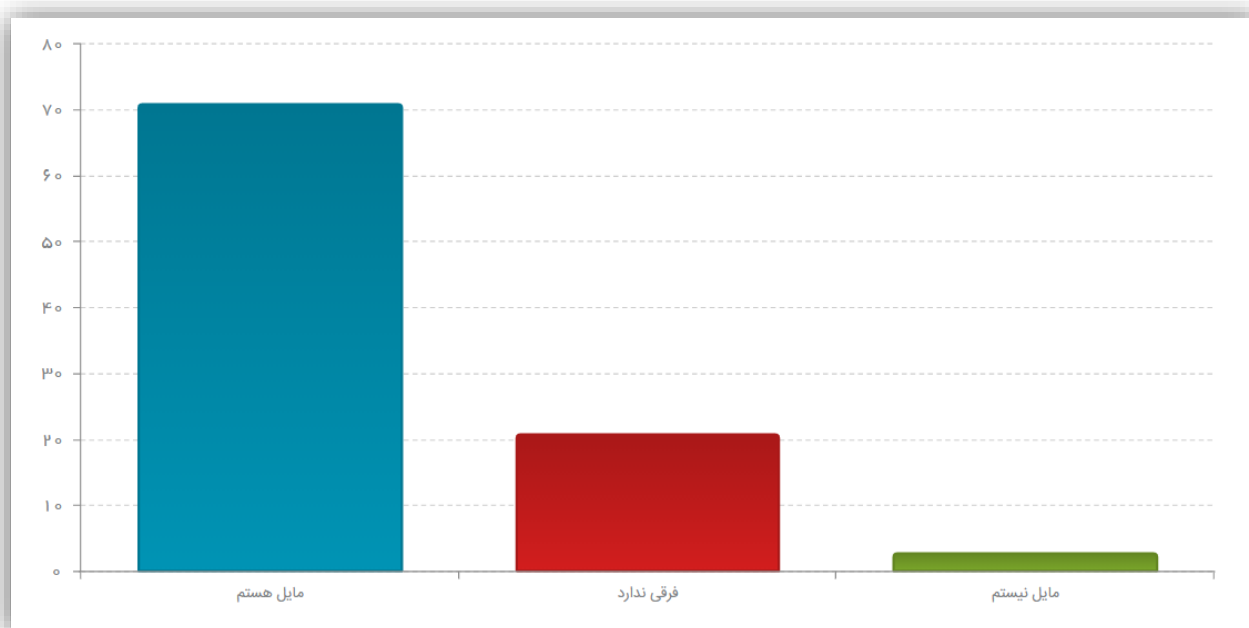




یکی از ویژگی هایی که میتواند به صاحبان یک فروشگاه کمک کند این است که برخی از محصولات جدید را که از موفقیت فروش آنها مطمئن نیستند را به صورت آزمایشی و در یک بازه معین برای فروش بگذارند. همچنین این ویژگی میتواند به تولید کنندگان محصول کمک کند تا از میزان فروش محصولات جدید خود آگاه شوند.

با استفاده از تائیدیه به دست آمده ما 2 ویژگی `startsAt`, `endsAt` را به جدول `product` اضافه کردیم تا این قابلیت را به ما نشان بدهند که محصولات را در بازه ای معین به فروش بگذاریم. همچنین میتوانیم ویژگی `startsAt` را در حالت عادی برای روزی که محصول را به فروش گذاشتیم و `endsAt` را برابر تاریخ انقضاء قرار دهیم تا بازه عادی فروش یک محصول را مشخص کند.

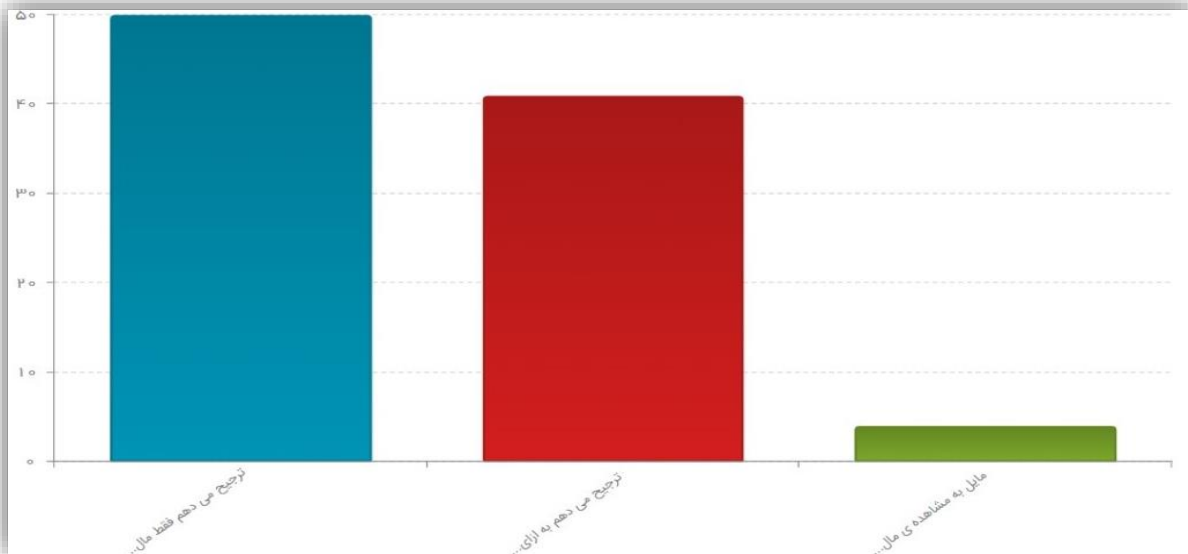
#### 14. آیا ترجیح می دهید در فاکتور مشخصات فروشنده نیز ذکر شود؟



قید شدن مشخصات فروشنده در فاکتور میتواند کمک کند تا در صورت بروز مشکل بتوانیم علت را پیگیری کنیم. با توجه به نتایج ما id فروشنده تایید کننده سفارش را در جدول `order` قرار دادیم.

زیرا فاکتور یک سند حقوقی معتبر است. سندی که می توان با اتکاء به آن علیه صادرکننده فاکتور اقامه دعوی کرد. این ویژگی به مشتریان این اطمینان را می دهد که در صورت عدم انطباق محصول خریداری شده با آنچه در فاکتور ثبت شده است؛ می تواند با پیگیری حقوقی موضوع، به احقاق حقوق از دست رفته خود بپردازند. این ویژگی فاکتور در معاملاتی با مبالغ بالا و یا پرداخت های اقساطی، از اهمیت بیشتری برخوردار است.

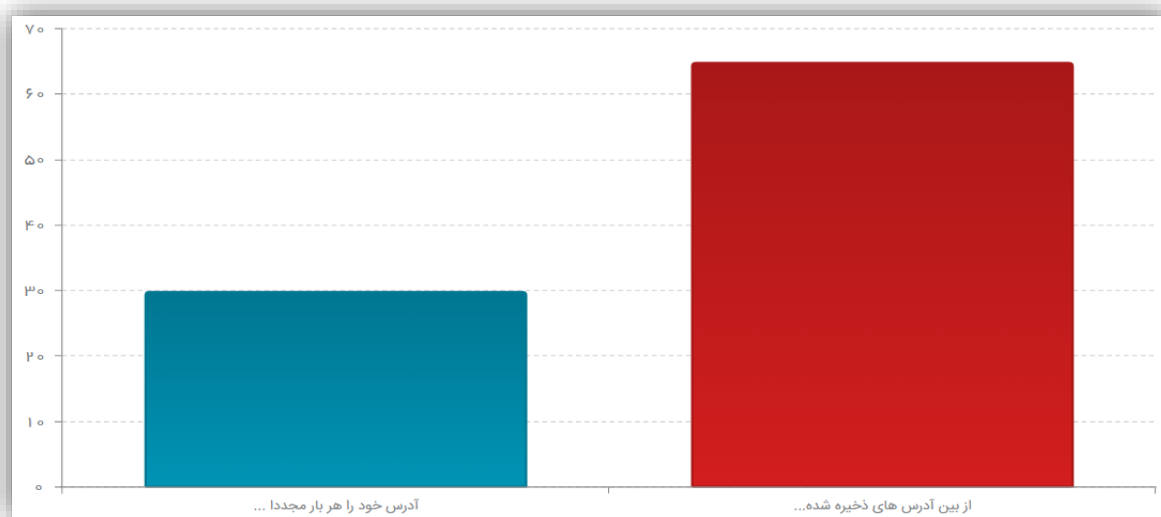
### 15. مالیات مربوط به خرید در سفارش شما چگونه ذکر شود؟



با توجه به نتایج مالیات ها را بر روی فاکتور اعمال کرده ایم تا برای کل محصولات و یکجا اعمال شود و جزئیات اضافه نمایش داده نشود. همچنین این کار کمک میکند تا اطلاعات کمتری را ذخیره کنیم زیرا طبیعتا ذخیره یک ویژگی برای هر محصول سخت تر است تا یک ویژگی برای سفارش.

### 16. آیا مایل هستید که برای هر سفارش آدرس خود را مجددا وارد کنید یا بتوانید از بین آدرس هایی که قبلا در

حساب کاربری شما ذخیره شده است انتخاب کنید؟



با توجه به ترجیح بر داشتن چند آدرس ذخیره شده ما قسمت آدرس را از جدول **product** جدا کرده و در جدول جداگانه ای قرار دادیم تا بتوانیم چندین آدرس را به یک مشتری مربوط کنیم. این کار در واقع با هدف نرمال سازی نیز انجام شده است. زیرا اگر میخواستیم که در جدول **customer** تمامی آدرس ها در فیلدی به نام **address** اضافه کنیم ، باعث میشد که برای بعضی از مشتری ها در یک سطر دو یا چند آدرس داشته باشند ، که این اتفاق اگر در پایگاه داده بیفتد غلط است و نیاز به نرمال سازی دارد.

اگر به ازای هر آدرس یک ستون در نظر می گرفتیم هم غلط بود چون ما نمی دانیم که یک نفر چند آدرس میتواند داشته باشد و تعداد ستون ها نا محدود می شد بنابراین یک جدول جدا برای آدرس های مشتری در نظر گرفتیم که در قسمت های بعدی گزارش مراحل کامل نرمال سازی و همچنین ویژگی جدول ها را توضیح دادیم. همچنین از نتیجه حاصله از این سوال تصمیم بر آن گرفتیم که شرایط یکسانی را برای شماره ی تلفن و ایمیل مشتریان و نیز افرادی که در این سیستم کار میکنند و **staff** نامیده می شوند ، در نظر بگیریم.

**17. اگر بتوانم وضعیت سفارش را مرحله به مرحله ببینم ، چقدر می تواند به افزایش اعتماد شما کمک کند؟ (به عنوان مثال محصول در راه خانه ی شماست یا در انباری دارد بسته بندی می شود)**



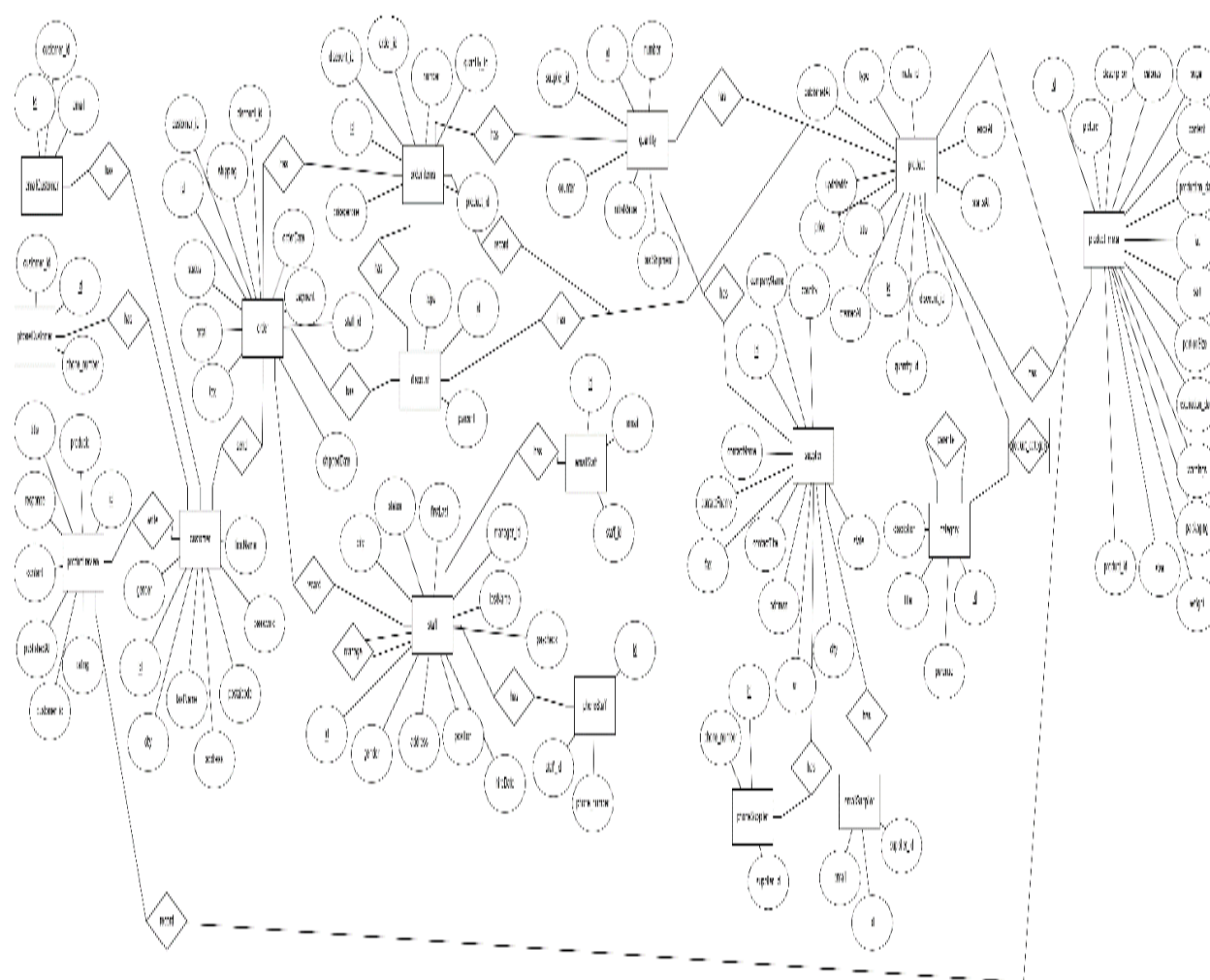
با توجه به نتیجه حاصله ما ویژگی **status** را در جدول **order** قرار دادیم تا بتوانیم حالت های مختلفی را در نظر بگیریم. یعنی به محض این که سفارش از سمت مشتری به ثبت رسید در قسمت **status** به مشتری نشان می دهیم که محصولات در انبار در حال بسته بندی می باشد . و زمانی که محصولات بسته بندی شده ، تحویل باربری داده شد یا زمانی که به درب منزل مشتری رسید می توانیم این وضعیت را در قسمت **status** به مشتری نشان دهیم. با استفاده از این ویژگی برای مسئولین فروشگاه پیگیری یک سفارش آسان تر می شود ، به عنوان مثال زمانی که یک مشکل یا نقض فنی به هنگام رساندن سفارش به مشتری پیش آمد می توانند ببینند که سفارش در چه وضعیتی قرار دارد و راحت تر بتوانند پیگیری کنند و هم مشتری بتواند از وضعیت سفارش خود مطلع شود.

## نمودارها:

ما برای کشیدن نمودار ها از سایت <https://erdplus.com> استفاده کردیم.

ابتدا نمودار ER مربوطه را کشیدیم و براساس آن نمودار relational مورد نظر را execute کردیم و یک شبه کد SQL براساس آن generate کردیم و سپس کد را تکمیل کرده و بسط دادیم.

## نمودار ER

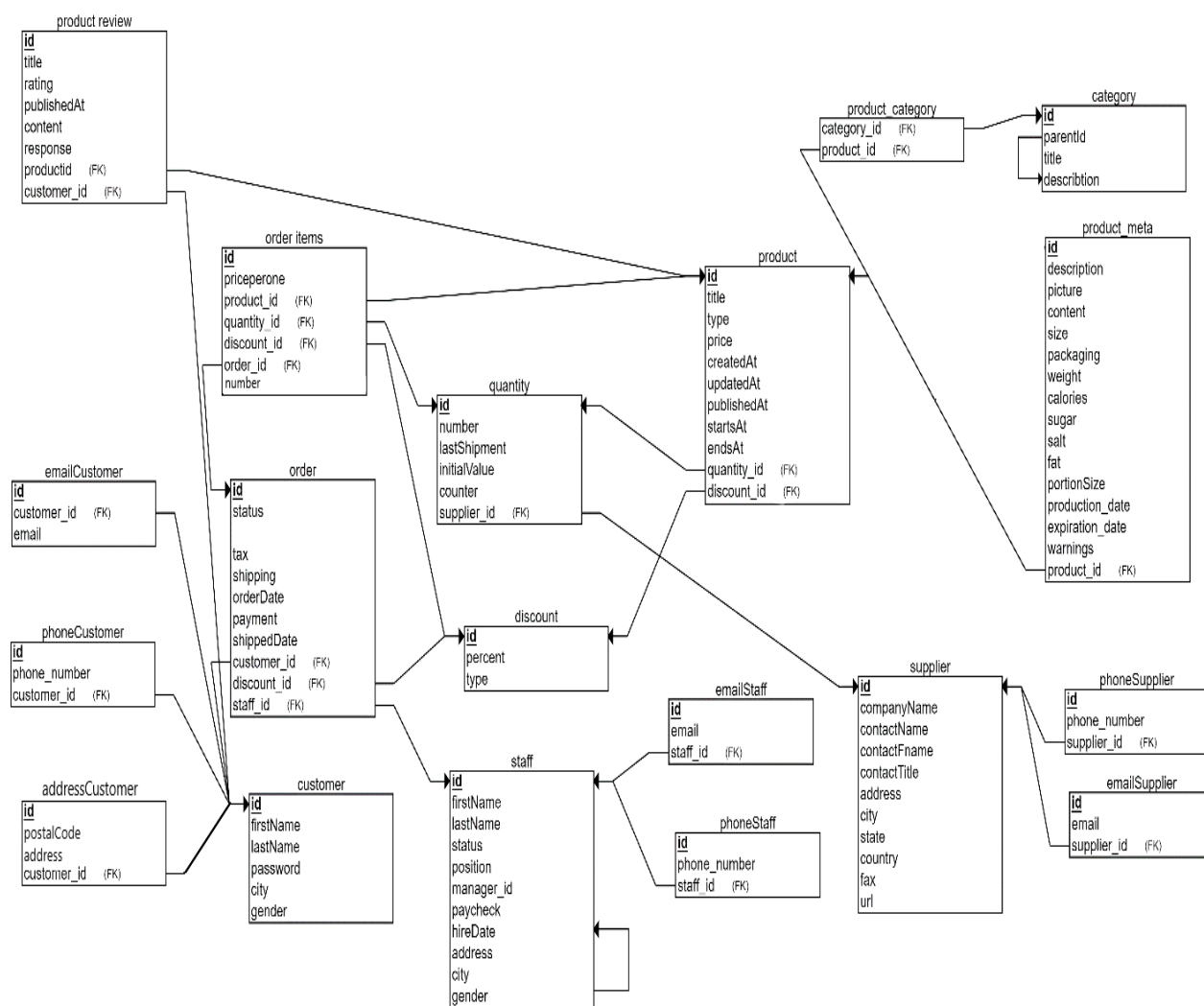


## نمودار Relational

نمودار روابط موجودیت ها یا Entity Relationship Diagram که به اختصار به آن مدل ER نیز می گویند یک نمودار است که Resource های یک نرم افزار یا اشیا یک پایگاه داده را به همراه صفات آن ها و روابط میان آن ها توصیف می کند.

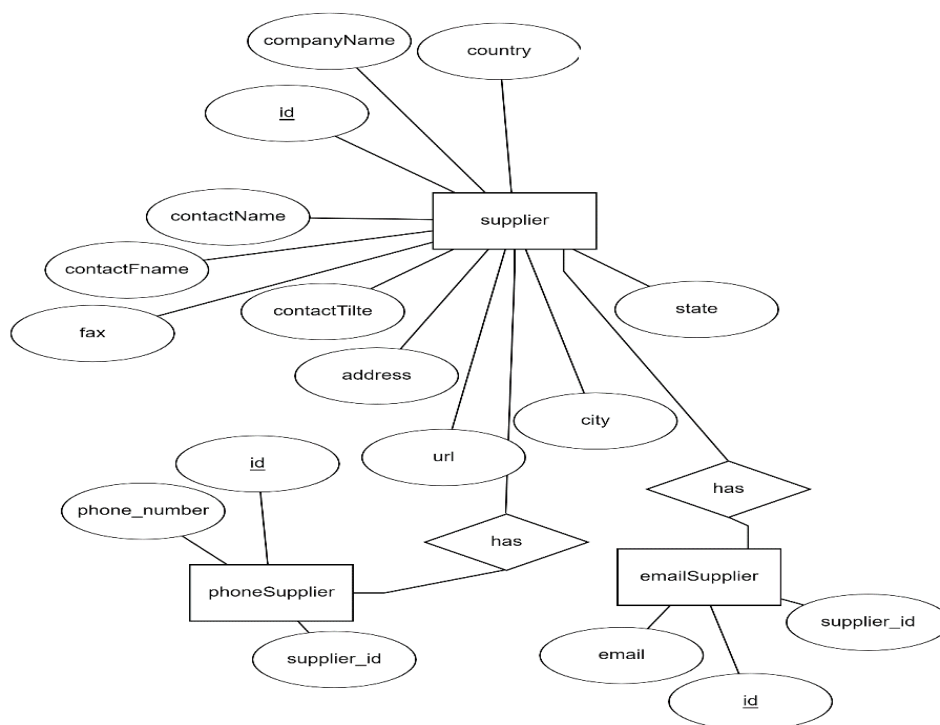
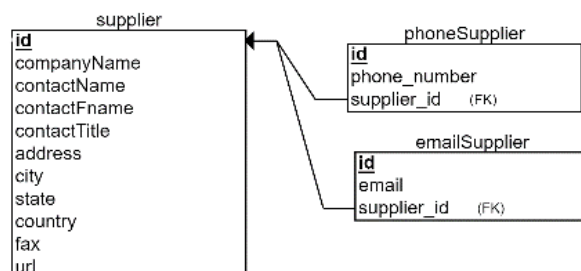
هر نمودار ER پس از تکمیل به راحتی تبدیل به یک پایگاه داده می شود و عملگر های آن نیز در نرم افزار پیاده سازی می شود (روابط میان موجودیت ها یا همان اشیا مدل). پس هر نمودار اطلاعات شماتیک در رابطه با نحوه کارکرد نرم افزار را در اختیار ما قرار می دهد.

در تصویر زیر نمودار Relational فروشگاه آنلاین کوروش دیده میشود:



## بررسی و تشریح جداول

جدول **supplier**:

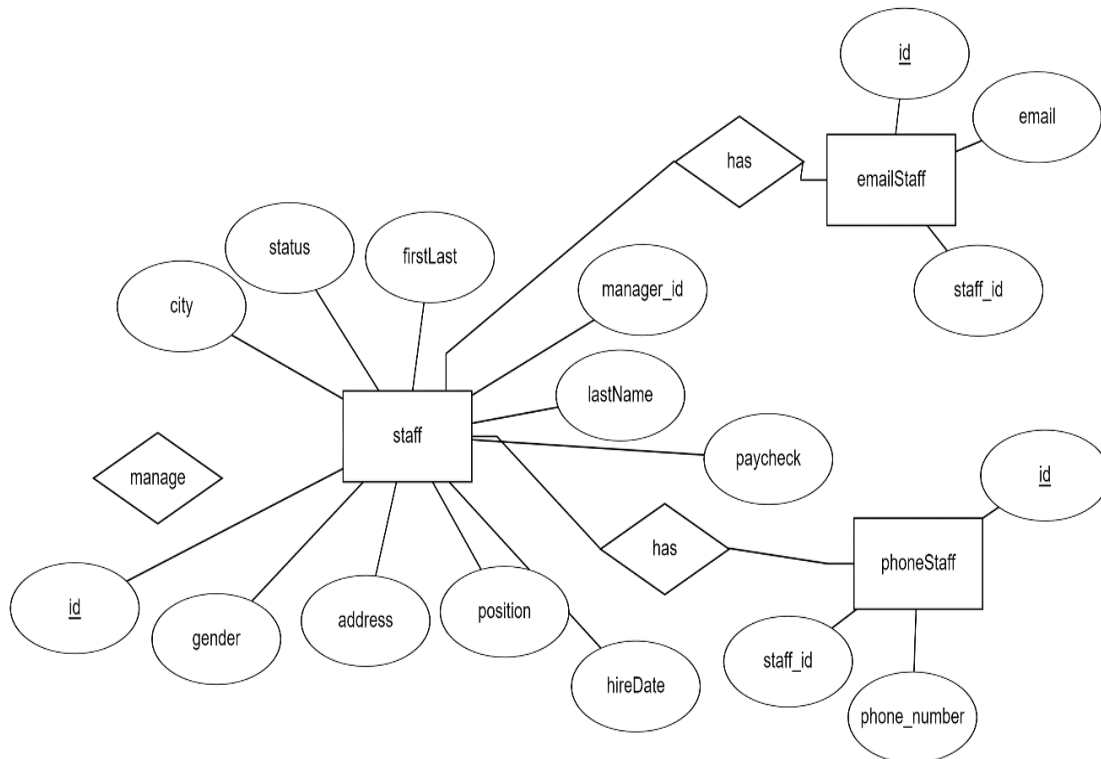
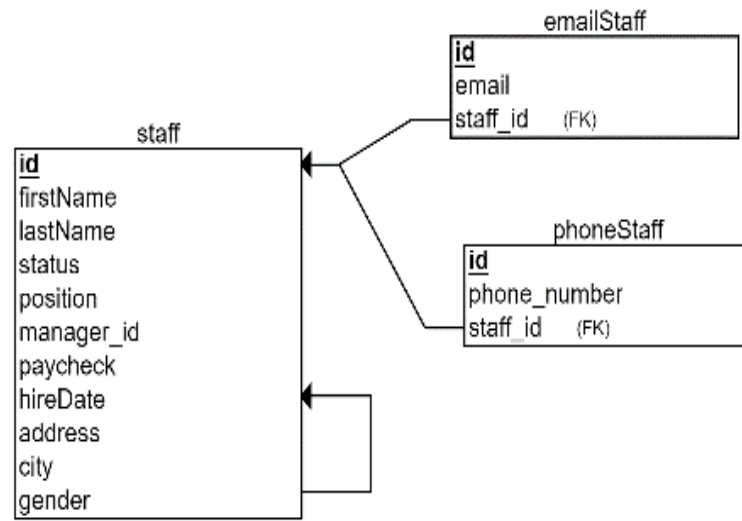


در جدول **supplier** ویژگی‌هایی مانند اسم شرکت، نام و نام خانوادگی رابط، آدرس، شهر، کشور، فکس و... شرکت مورد نظرمان را ثبت کردیم.

در قسمت نرمال سازی به علت وجود **dependency** ها و چند مقداری بودن شماره تلفن و ایمیل **supplier**، احتیاج به آرایه حس میشد اما **sql** این ویژگی را نداشت پس برای آنها جداول جدایی طراحی کردیم.

در نمودار **ER** همان طور که میبینیم **supplier** از طریق رابطه **has** به شماره تلفن و ایمیلش متصل شده است.

جدول staff :



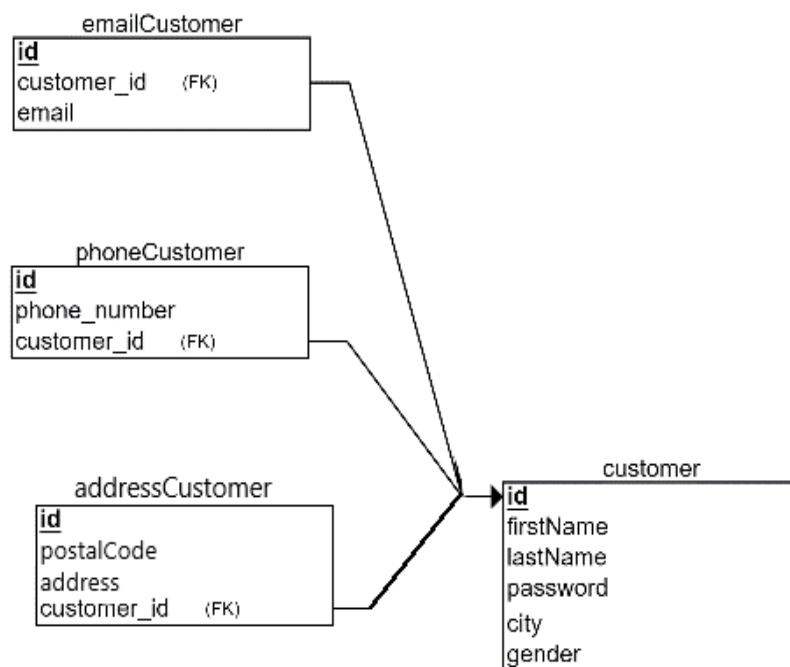
در جدول **staff** ویژگی هایی مانند نام و نام خانوادگی کارمند فروشگاه، درجه کارمند، تاریخ استخدام، آدرس، شهر، جنسیت و... کارمند را ثبت کردیم.

صفت **manager\_id** به خودش رابطه دارد و این جدول به صورت درختی است و یک نفر میتواند علاوه بر کارمند بودن، مدیریت دیگر کارمندان را نیز داشته باشد که در این حالت در صفت **manager\_id** کارکنان این فرد برابر با **id** فرد مسئول قرار خواهد گرفت.

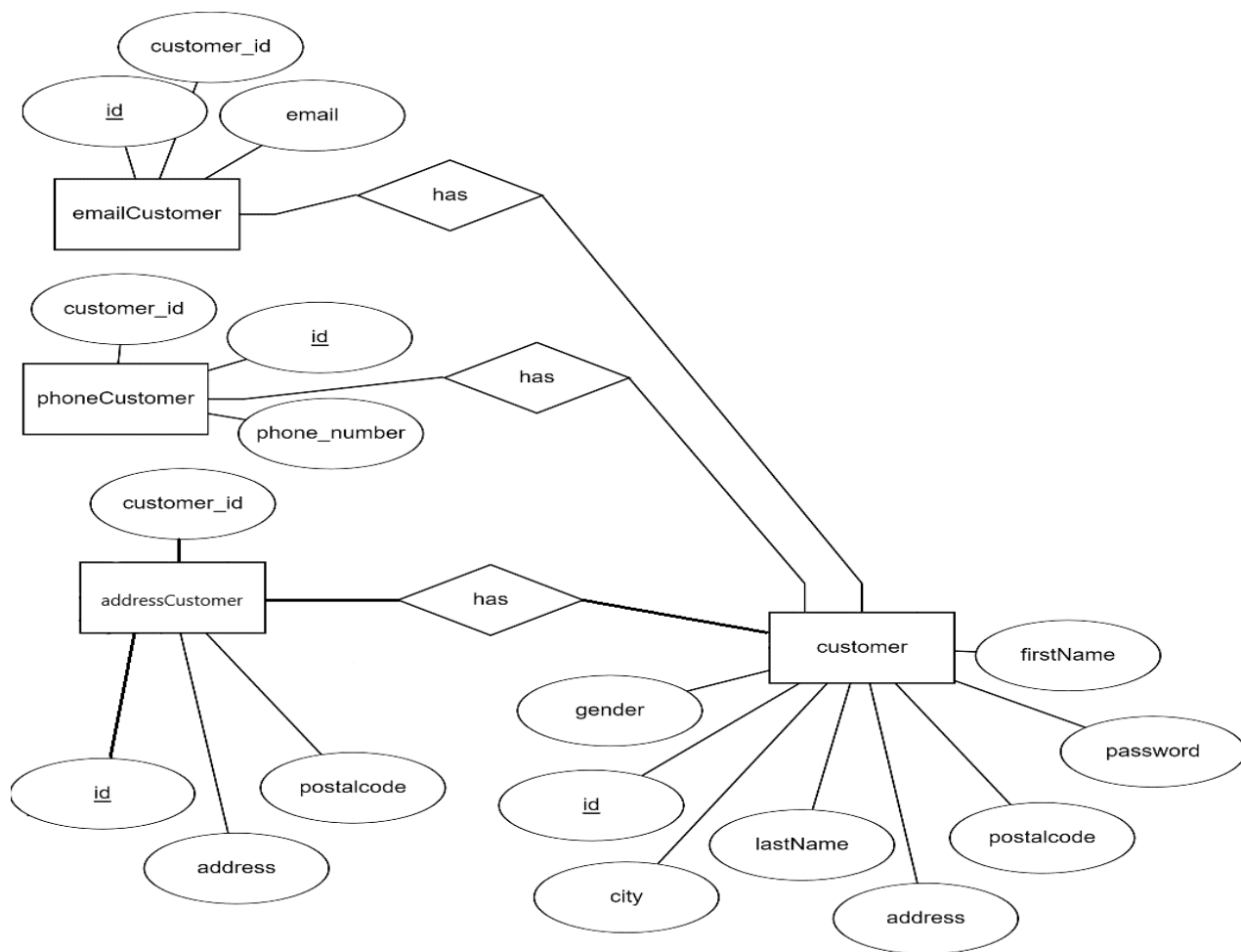
در قسمت نرمال سازی به علت وجود **dependency** ها شماره تلفن و ایمیل **staff** که به صورت چند مقداری هستند و احتیاج به آرایه داشتیم اما **sql** این ویژگی را نداشت برای آن ها جداول جدایی طراحی کردیم.

در نمودار ER همان طور که میبینیم **staff** از طریق رابطه **has** به شماره تلفن و ایمیلش متصل شده است.

#### جدول **customer** :





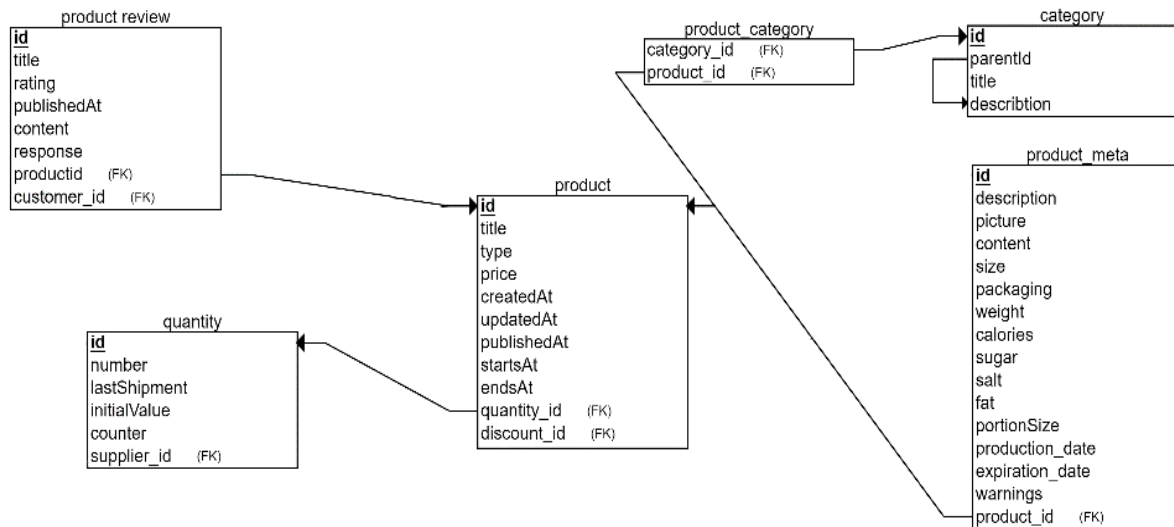


در جدول **customer** ویژگی هایی مانند نام، نام خانوادگی، پسترد، کدپستی، آدرس، شهر و جنسیت مشتری را ثبت کردیم.

در قسمت نرمال سازی به علت وجود **dependency** ها شماره تلفن و ایمیل **customer** که به صورت چند مقداری هستند و احتیاج به آراییه داشتیم اما **sql** این ویژگی را نداشت برای آن ها جداول جدایی طراحی کردیم.

در نمودار **ER** همان طور که میبینیم **customer** از طریق رابطه **has** به شماره تلفن، ایمیل و آدرسش متصل شده است.

## جداول و روابط محصولات :



## روابط:

جدول اصلی موجود در این عکس جدول product است و باقی جداول بنا به پرسشنامه ای که طراحی شد و یا برای نرمال سازی اضافه شده اند که در ادامه به آنها میپردازیم.

محصولات احتیاج به دسته بندی داشتند چرا که اگر بخواهیم در فروشگاه آنلاین به دنبال محصولی بگردیم، وجود دسته بندی به یافتن راحت تر محصول می انجامد. علاوه بر این مساله در پرسش نامه طراحی شده دیدیم که قرار دادن دسته بندی توسط کاربران نیز بسیار پشتیبانی شده و مورد استقبال قرار گرفته است.

پس ما جدول category را به وجود آوردیم و product\_category را به عنوان واسطه این دو جدول ( جدول product و category ) قرار دادیم

در پرسشنامه کاربران ذکر کرده بودند که وجود نظرات در مورد محصولات می تواند باعث شود در مورد محصول مورد نظرشان تجدید نظر کنند پس ما جدول product\_review را قرار دادیم و آن را رابطی بین محصول و کاربر کردیم؛ علاوه بر آن کاربران در پرسش نامه علاقه خود را نسبت به وجود اطلاعات بیشتر در مورد محصول ابراز کرده بودند، چرا که وجود اطلاعات بیشتر از محصول باعث میشد مشتریان متوجه ویژگی های محصول شوند و باعث سهولت در تصمیم گیری آنها برای خرید محصول میشود.

پس ما جدول product\_meta را طراحی کردیم و در آن ویژگی‌های محصولات را قرار دادیم

از طرفی حالتی را در نظر گرفتیم که شاید مشتری بخواهد از یک محصول چند عدد خریداری کند برای این مساله احتیاج به یک شمارنده احساس میشد، پس جدول quantity را طراحی کردیم تا تعداد محصولات خریداری شده را در آن ذخیره کنیم.

جدول product review :

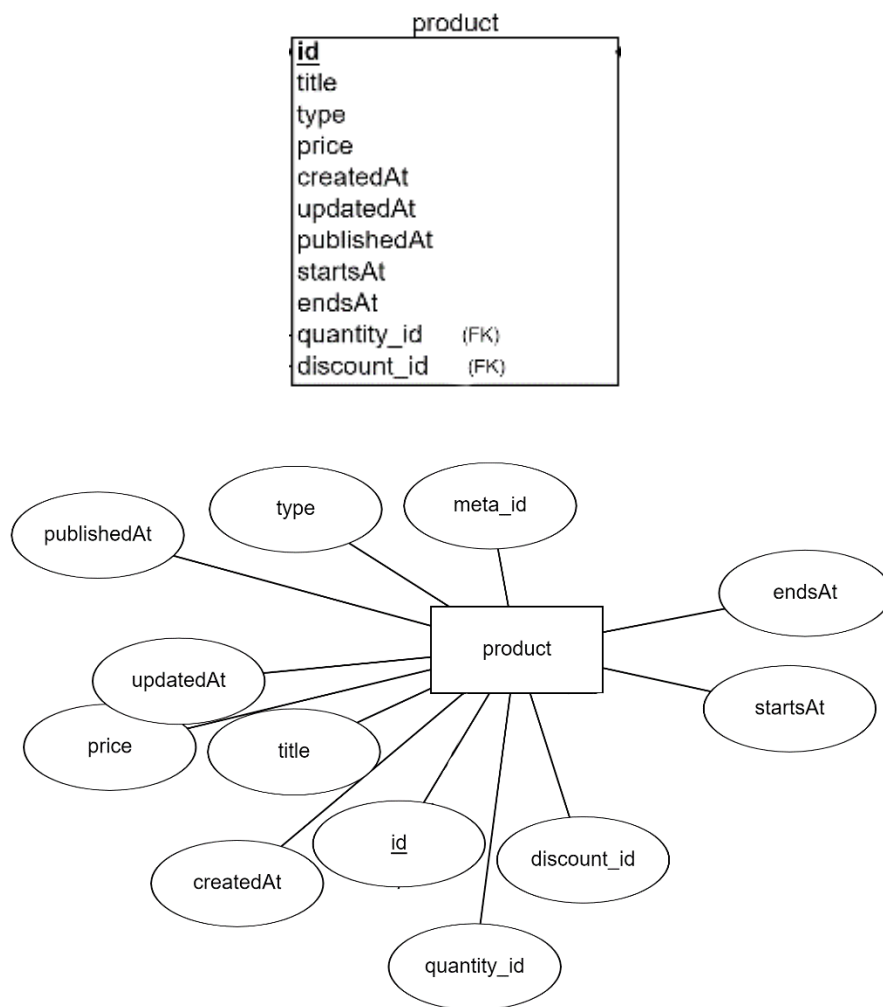
product review	
<u>id</u>	
title	
rating	
publishedAt	
content	
response	
productid	(FK)
customer_id	(FK)



این جدول شامل تیتیر، رتبه، زمان ثبت نظر، محتوای نظر میباشد که در آن کاربر میتواند هنگام ثبت نظر به محصول رتبه دهد که به جدول product از طریق productid و به جدول customer از طریق customer\_id وصل میشود.

در نمودار ER همان طور که میبینیم product review از طریق رابطه write به customer و از طریق record به product متصل شده است.

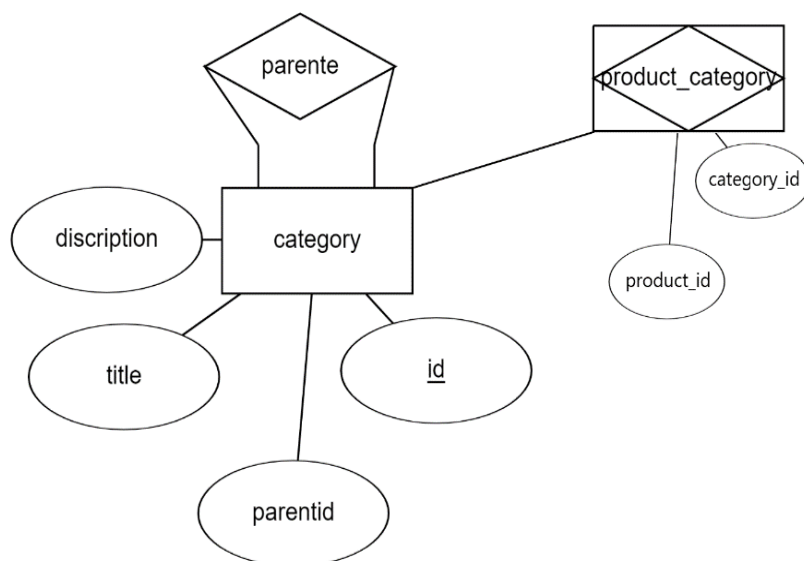
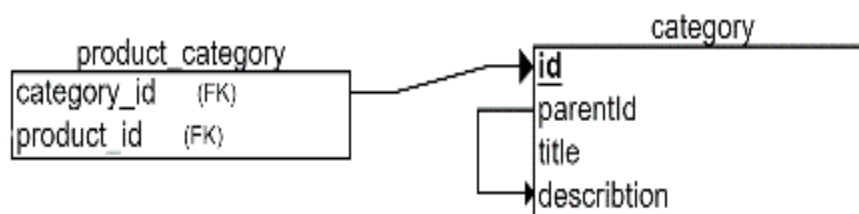
جدول product :



این جدول شامل نام محصول، مدل محصول، قیمت محصول، زمان ساخت و توزیع محصول، تاریخ شروع و پایان می باشد و باقی اطلاعات مانند تعداد خریداری شده از محصول، تخفیف محصول، شرکت فرستنده محصول و... در جداول جداگانه آمده اند و با محصول ارتباط دارند.

در نمودار ER همان طور که میبینیم product از طریق رابطه has به product meta و از طریق record به product review و از طریق product\_category به category متصل شده است که رابطه product\_category خود نیز از نوع موجودیت است.

جدول category :



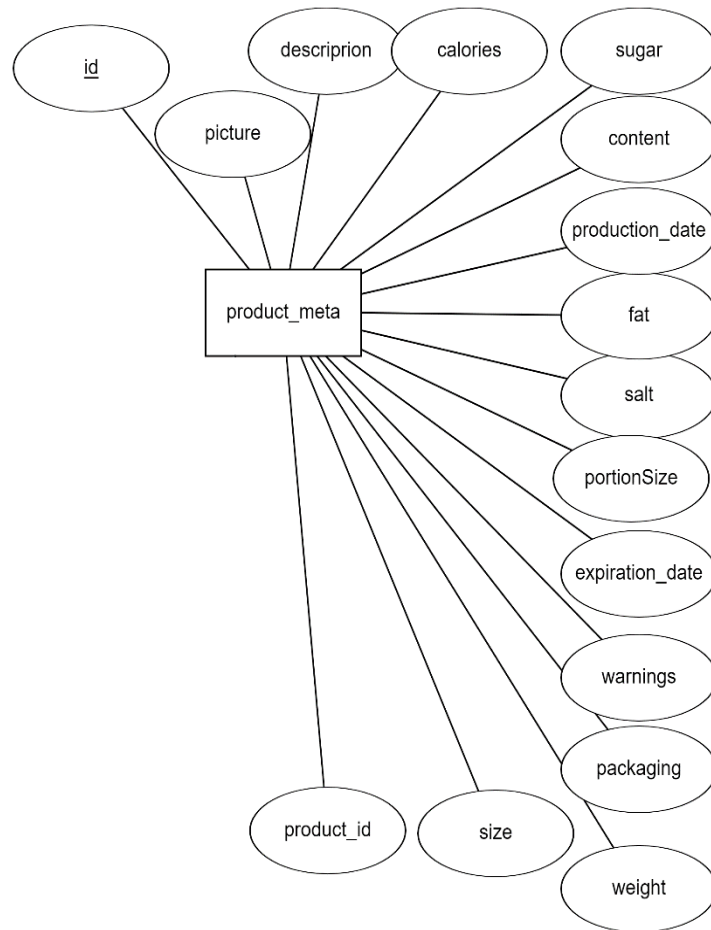
هر محصول دارای دسته بندی است که این دسته بندی می تواند شامل نام دسته و توضیحات این دسته باشد. دسته بندی می تواند به صورت درختی باشد یعنی یک دسته می تواند شامل دسته های دیگری (زیر دسته) نیز باشد، به همین منظور ما یک صفت به اسم `parentId` درست کردیم که این صفت به خودش برمیگردد در این حالت در صفت `parentId` زیر دسته ها برابر با `id` دسته اصلی قرار خواهد گرفت. در نهایت برای اینکه محصولات را به دسته بندی وصل کنیم، چون در حالت عادی `dependency` به وجود می آورد جدولی دیگر در نظر گرفتیم و این دو جدول را به کمک این جدول به همدیگر متصل کردیم.

از طرفی ما در پرسش نامه از کاربر پرسیده بودیم که آیا علاقه مند به دسته بندی کلی است یا جزئی که دسته بندی جزئی رای آورده بود به همین علت جدول را با ساختار تو در تو ساختیم که این حالت را ایجاد کند.

در نمودار ER همان طور که میبینیم `category` از طریق `product_category` به `product` متصل شده است که رابطه `product_category` خود نیز از نوع موجودیت است.

جدول `product_meta` :

product_meta	
<u>id</u>	
description	
picture	
content	
size	
packaging	
weight	
calories	
sugar	
salt	
fat	
portionSize	
production_date	
expiration_date	
warnings	
product_id	(FK)

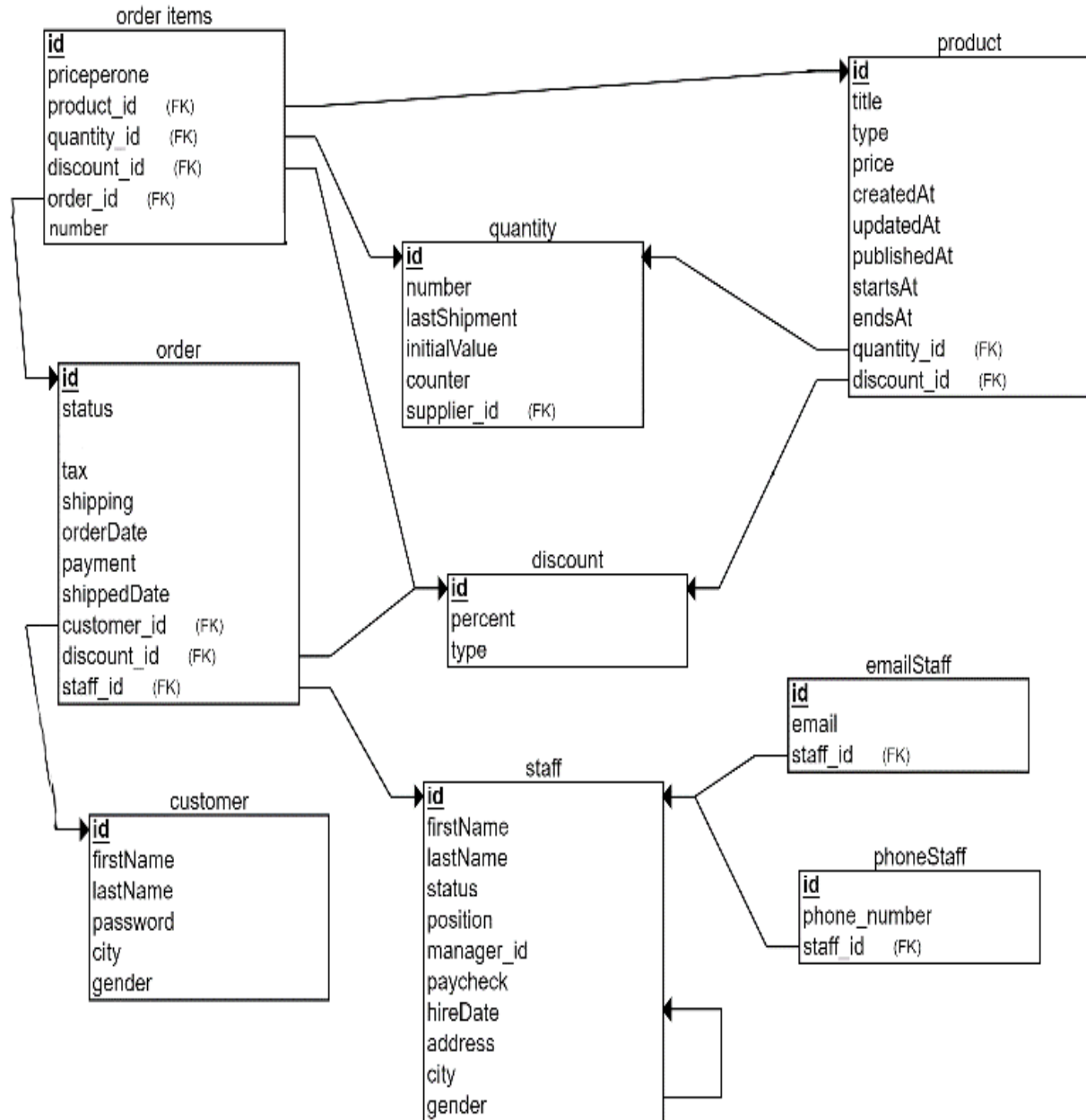


ما محصولاتی داریم مانند مواد خوراکی که جزئیات دارند. این جدول به ما کمک می‌کند تا این جزئیات را ببینیم و لزوماً لازم نیست که همه ستون‌های آن پر باشد (یعنی می‌توانیم ستون null نیز داشته باشیم)

این جدول شامل صفات : سایز، نوع بسته بندی، وزن، کالری، شکر، نمک، چربی، مقدار پروتئین، تاریخ تولید، تاریخ انقضا و هشدارها که این هشدارها برای مثال می‌تواند شامل این باشد که حساسیت‌زا بودن و اینکه چه مقدار باید از آن مصرف کنیم و... که این جدول به جدول محصولات وصل خواهد شد

در نمودار ER همان طور که می‌بینیم product meta از طریق رابطه has به product متصل شده است

جداول و روابط سفارشات :





## روابط:

ابتدا در جدول order احتیاج داشتیم که بدانیم کدام مشتری چه سفارشی داشته است پس این جدول را به جدول customers از طریق customer\_id متصل کردیم

در پرسشنامه کاربران علاقه مند به دیدن نام فروشنده در رسید بودند پس در هر سفارش ما نام کارمندی که محصول را فروخته است را نیز ثبت میکنیم برای این کار جدول staff را به جدول order وصل کردیم.

برای محصولات خریداری شده در نهایت تخفیفی نیز در نظر گرفتیم که می تواند در موارد مختلف تخفیف های متفاوتی باشد

تخفیفات به دو دسته تقسیم میشوند : تخفیفی که مخصوص یک محصول است و یا تخفیف برای کل خرید یعنی حالتی که جشن یا مناسبت خاصی باشد

ما یک تخفیف کلی برای هر خرید در نظر می گیریم پس این جدول را به جدول orders وصل کردیم

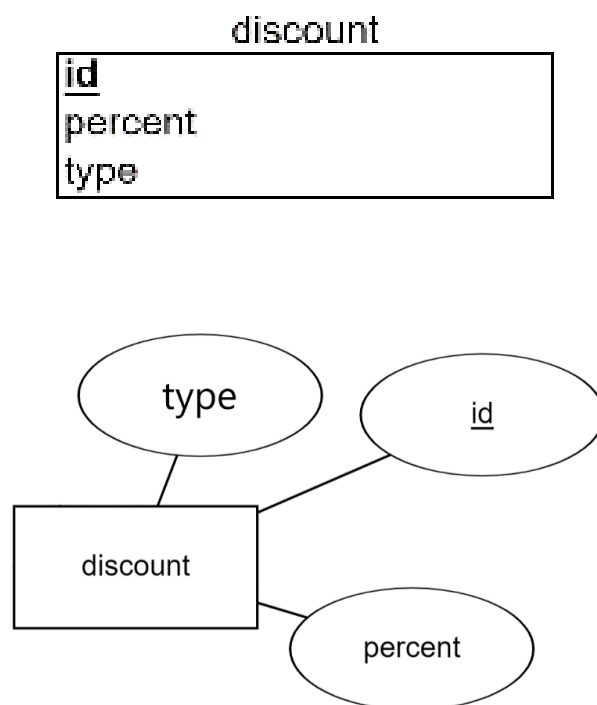
حال اگر شرکتی تعداد خاصی از یک محصول خرید کرد برای آنها نیز تخفیف قائل شویم برای این حالت این جدول را به order items وصل کردیم تا اینکه بدانیم از هر محصول چند عدد خریداری شده و اگر تعداد خریداری شده ما برابر با تعداد مورد نظر ما بود آن محصول شامل تخفیف شود.

ما به یک انبار هم احتیاج داشتیم تا بدانیم چه تعدادی از یک محصول باقی مانده است برای همین جدول orders را به جدول quantity وصل کردیم.

در نهایت ما برای اینکه بدانیم چه محصولاتی خریداری شده اند جدولی با اسم order\_items درست کردیم که در این جدول آی دی محصول خریداری شده، تعداد محصول و تخفیف ثبت میشود و در نهایت در کنار آنها آی دی جدول order میاید که نشان میدهد این اطلاعات برای کدام سفارش است.

بررسی تک تک جداول:

جدول discount :

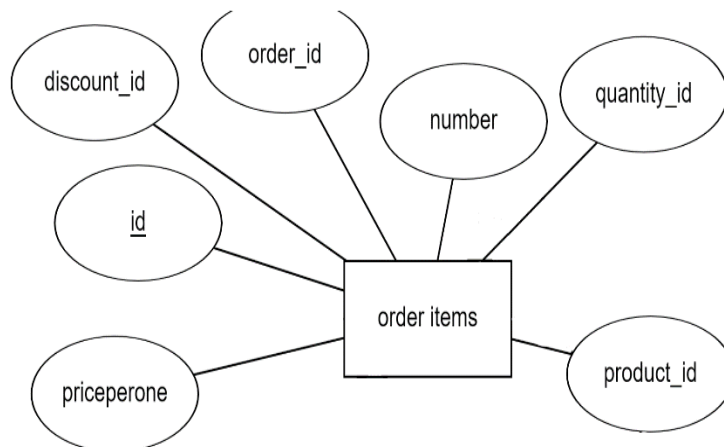


در این جدول ما صفت درصد را داریم که میزان درصد تخفیف را مشخص می‌کند. علاوه بر آن صفت type را داریم که ما می‌توانیم در آن مشخص کنیم که مدل تخفیف به چه صورتی است یعنی تخفیف به صورت درصدی حساب می‌شود یا اینکه اگر فردی از یک محصول تعداد خاصی خرید کرد یک یا چند عدد دیگر نیز جایزه ببرد.

در نمودار ER همان طور که می‌بینیم discount از طریق رابطه has به order , order\_items , product متصل شده است

جدول order items :

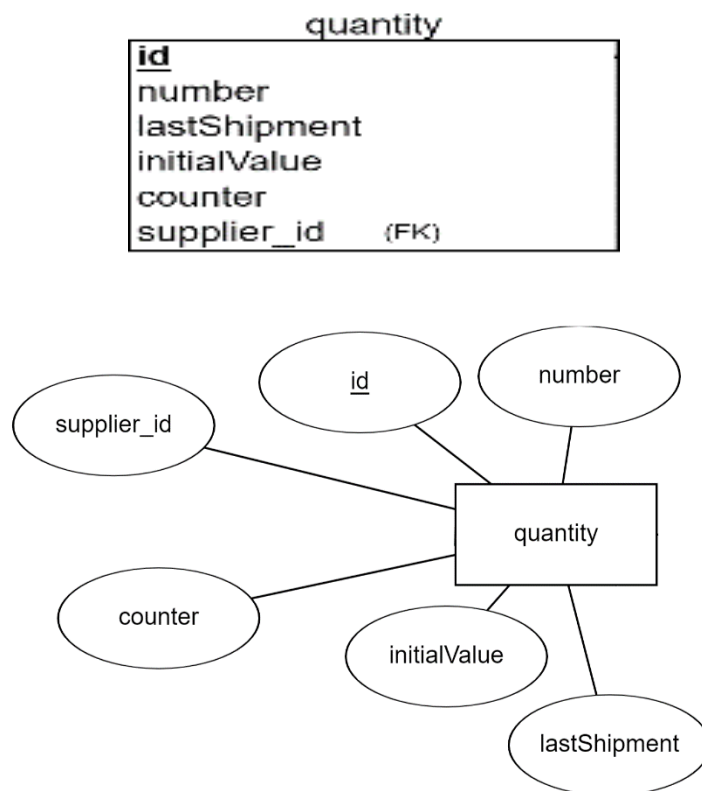
order items	
<u>id</u>	
priceperone	
product_id	(FK)
quantity_id	(FK)
discount_id	(FK)
order_id	(FK)
number	



items دارای یک آیدی محصول است که **product** به آن وصل شده تا بدانیم چه محصولاتی را خریداری کرده ایم و همچنین به **order** وصل شده تا بدانیم از کدام سفارش بوده و قیمت و تخفیف هر یک از محصولاتمان چقدر هست

در نمودار ER همان طور که میبینیم **order items** از طریق رابطه **has** به **quantity, order** و از طریق **record** به **product** متصل شده است.

## جدول quantity :

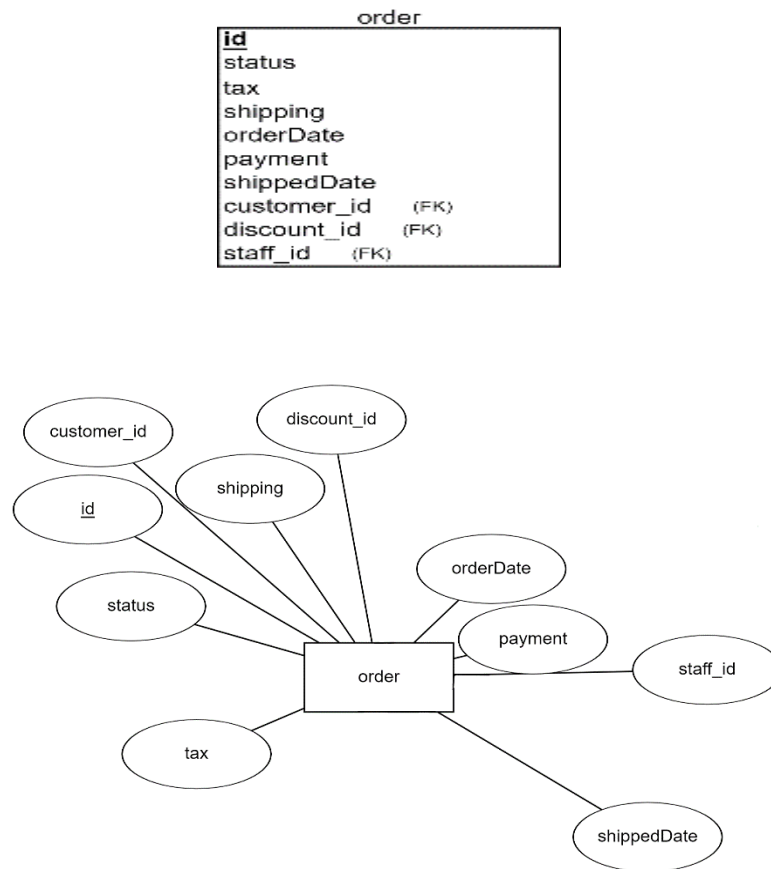


این جدول شامل ویژگی‌های تعداد فعلی آخرین زمانی که محصول برای ما ارسال شده تعداد اولیه محصول خریداری شده تعداد بارهایی که محصول را فروخته ایم می‌باشد

این جدول یک شبه انبار است و در آن تعداد فعلی محصولاتمان ثبت میشود که دارای ویژگی‌هایی مانند این است که آخرین زمانی که محصول برای ما ارسال شده را ثبت می‌کند که اگر محصولی در مدت کمی فروش میرفت با توجه به این صفت متوجه بشویم و دفعه بعد از این محصول بیشتر سفارش بدهیم، در ادامه تعداد اولیه ای که محصول خریداری شده را ثبت میکنیم و در عین حال میشماریم که چند بار از یک محصول تا الان خریداری شده است که علت آن برای حالتی است که برای مثال اگر از محصولی 500 عدد آوردیم و در 10 بار فروش رفته متوجه میشویم که این محصول با تعداد بالا خریداری میشود پس بسته‌های بزرگ تری را در دفعات آینده خریداری میکنیم

در نمودار quantity همان طور که میبینیم product از طریق رابطه has به order items,product متصل شده است .

## جدول order :



با توجه به پرسشنامه که طرح کردیم کاربرانی که این پرسشنامه را پر کردن خواستار این بودند که وضعیت سفارش شان را بدانند به همین دلیل صفت **status** را در جدول قرار دادیم.

ارزش افزوده را در آخر به صورت کلی برای کاربر به نمایش می گذاریم که برای آن متغیر **tax** را قرار دادیم موارد لازم دیگر که باید در سفارشات ثبت میشدند عبارتند از هزینه حمل و انتقال، زمان ثبت سفارش و زمانی که کالا ارسال شده است که این موارد برای سفارش هر فرد به صورت جداگانه ثبت و کنترل میشود.

ما یک صفت به اسم **payment** از نوع **small int** در نظر گرفتیم که در این صفت مشخص می کند که مشتری مبلغ را پرداخت کرده است و یا خیر. نحوه پرداخت که آیا به صورت نقدی باشد و یا کارت به کارت و یا روش های دیگر به ما در این قسمت ربطی ندارد و برای آن چیزی در نظر نگرفتیم.

در نمودار ER همان طور که میبینیم **product** از طریق رابطه **has** به **order items** , **discount** و از طریق **send** به **customer** متصل شده است.

## نرمال سازی بانک اطلاعاتی

نرمال سازی روشی برای طراحی جداول پایگاه داده است و داده ها را به روشی ساماندهی می کند که باعث کاهش افزونگی داده و رفع مشکلات ساختاری و آنومالی شود. هدف از نرمال سازی حذف افزونگی داده و باقی نگهداشتن وابستگی بین داده های مرتبط است. به این طریق اندازه پایگاه داده را کاهش داده و ذخیره منطقی داده را تضمین می کند.

در واقع افزونگی به این معناست یک داده خاص در چند محل مختلف پایگاه ذخیره شود. این امر موجب می شود که این خطر بالقوه به وجود آید که داده ها هر لحظه با هم در تضاد قرار گیرند و استخراج واقعیت از آنها غیرممکن شود به بیان دیگر فرایندی است که بر اساس آن داده ها و اطلاعات در واحدهای منطقی به نام جدول به شکلی توزیع می شود که علاوه بر حفظ موجودیت داده ها از ایجاد پدیده ی افزونگی جلوگیری بعمل می آورد.

به این منظور فرم های نرمال متعددی تعریف و مورد استفاده قرار می گیرد، برای نرمال سازی یک جدول لازم است ابتدا در فرم اول نرمال شده سپس در فرم های بعدی بررسی گردد، در هنگام نرمال سازی جداول تعداد جداول در پایگاه داده افزایش می یابد.

## دلایل نرمال سازی

- ✓ سادگی در نمایش ظاهری
- ✓ سادگی دستورات زبان
- ✓ سادگی اجرای عملیات در پایگاه داده

## فرم های نرمال در پایگاه داده :

تئوری پایگاه داده درجه نرمال سازی جدول را با اصطلاح فرم های نرمال (normal form) شرح می دهد. فرم های نرمال ( یا بطور خلاصه NF ) معیاری برای تعیین درجه نرمال جدول در اختیار می گذارد. فرم های نرمال جداگانه روی هر جدول می توانند بکار بروند.

پایگاه داده زمانی در فرم نرمال n خواهد بود که کل جداول آن در فرم نرمال n باشند. فرم های نرمال عبارتند از:

- (First Normal Form) 1NF ✓
- (Second Normal Form) 2NF ✓
- (Third Normal Form) 3NF ✓
- (Fourth Normal Form) 4NF ✓
- (Boyce/Codd Normal Form) BCNF ✓
- (Fifth Normal Form) 5NF ✓
- (Domain/Key Normal Form) DKNF ✓

فرم نرمال هریک باعث کاهش بیشتر افزونگی و تقسیم جداول به واحدهای کوچکتر می شوند. سه فرم اول نرمال (1NF، 2NF و 3NF) در ابتدا توسط Codd تعریف شد که به طور خلاصه وابستگی صفات خاصه غیر کلید را به کلید الزام می کنند. فرم های چهارم و پنجم (4NF و 5NF) با ارتباطات چند به چند و یک به چند بیت صفات خاصه سروکار دارند. دو فرم دیگر هم وجود دارد که کاملاً با این جریان جور نمی شوند که BCNF و DKNF هستند.

### نرمال سازی 1NF (First Normal Form)

یک جدول در فرم اول نرمال (1NF) است اگر و فقط اگر فاقد گروه داده تکرار شونده باشد. به عبارت دیگر هر ستون در جدول دارای مقدار اتمیک باشد. در مدل رابطه ای هر جدولی حداقل در فرم اول نرمال هست زیرا از الزامات مدل این است که هر جدول شامل دقیقاً یک مقدار برای هر صفت خاصه باشد که اصطلاحاً "فاقد گروه تکرار شونده" گفته می شود.

### نرمال سازی 2NF (Second Normal Form)

یک جدول در فرم دوم نرمال (2NF) است اگر اولاً 1NF باشد و ثانیاً کلیه ستون های غیرکلید با کلید اصلی وابستگی تابعی کامل داشته باشند. ستون Y با ستون X در یک رابطه وابستگی تابعی (functional dependency) دارد اگر و فقط اگر به ازای هر مقدار در X دقیقاً یک مقدار در Y متناظر با آن وجود داشته باشد. که به صورت  $X \rightarrow Y$  نشان داده می شود.

### نرمال سازی 3NF (Third Normal Form)

یک جدول در فرم سوم نرمال (3NF) است اگر اولاً 2NF باشد، ثانیاً کلیه صفات خاصه غیر کلید در جدول با کلید اصلی وابستگی تابعی غیر تعدی داشته باشند. وابستگی تعدی (transitive dependency) یک وابستگی تابعی غیر مستقیم است که در آن  $X \rightarrow Z$  است اگر  $X \rightarrow Y$  و  $Y \rightarrow Z$  باشد.

در فرم سوم نرمال کلیه ستون های جدول مستقیماً توسط کلید اصلی مشخص می شوند. با حذف فیلدهائی که وابستگی مستقیم با کلید ندارند به فرم سوم نرمال می رسیم. برای این کار گروهی از ستون های جدول را که مقدارشان برای بیش از یک رکورد تکرار می شود را در جدول جداگانه ای قرار دهید.

### نرمال سازی 4NF (Forth Normal Form)

یک جدول در فرم چهارم نرمال (4NF) است اگر اولاً 3NF باشد، ثانیاً هیچ ستونی در جدول وابستگی چند مقداری نداشته باشد. وابستگی چندمقداری (multivalued dependency) به این معنی است که حضور رکوردهای معینی در جدول وجود رکوردهای معین دیگری را برساند.

### معایب نرمال سازی

نرمال سازی علاوه بر اینکه که افزونگی داده را کاهش می دهد سبب کاهش اجرای سیستم می شود. درجات بالای نرمال معمولاً جدول بیشتر را می طلبند. برای پاسخ به پرس و جوها گاهی باید کلیه جداول تقسیم شده دوباره با هم الحاق شوند که در کاربردهائی که زمان پاسخ مهم است (نظیر وب) مطلوب نیست.

ما در پروژه ی خود تا سطح سوم نرمال سازی پیش رفتیم و سطح چهارم را لازم ندانستیم . زیرا پایگاه داده ی ما به اندازه ی یک پایگاه داده ی اداره ی مالیات دارای داده های حیاتی نبود و افزونگی در آن یک مشکل حاد محسوب نمی شود ، که تا آخرین سطح بخواهیم نرمال سازی انجام دهیم.



## نرمال سازی جداول سامانه :

1- جدول customer در اول به شکل زیر بود :

id	firstName	lastName	password	city	gender	email	phone	address
1	نگین	شاهانی	*****	لنگرود	1	negin@gmail.com negin2@gmail.com	0902....31 4252....99	آدرس 1 کاربر 1 آدرس 2 کاربر 1
2	علی	تقوی	*****	رشت	0	ali@gmail.com ali2@gmail.com	0902....32 4252....78	آدرس 1 کاربر 2 آدرس 2 کاربر 2

با توجه به جدول بالا مشاهده می کنیم که ستون email و phone و address بیش از یک مقدار را در خود ذخیره کرده اند. زیرا طبق تعریفی که ما برای مشتری قرار دادیم، هر مشتری می تواند یک یا چند شماره تلفن و ایمیل داشته باشد. همچنین به هنگام ثبت سفارش از آنجایی که مشتری می تواند از بین آدرس هایی که از قبل وارد سامانه شده اند ، آدرسی انتخاب کند پس مشتری می تواند آدرس های متفاوتی به سامانه بدهد. اگر بخواهیم جدول را به سطح یک نرمال سازی برسانیم.

اگر در نظر بگیریم که چون در ستون email دو ایمیل وارد شده پس دو سطر به وجود آوریم این کار نیز مشکل را رفع نمیکند زیرا اگر کاربر ایمیل سوم یا چهارمی هم داشت به همان نسبت باید ستون های بیشتری ایجاد شود پس باید نرمال سازی انجام دهیم.

پس علاوه بر جدول customer ، جدول phoneCustomer ، emailCustomer و addressCustomer نیز در نظر می گیریم. در نتیجه :

جدول customer :

id	firstName	lastName	password	city	gender
1	نگین	شاهانی	*****	لنگرود	1
2	علی	تقوی	*****	رشت	0

جدول emailCustomer :

id	Customer_id	email
1	1	negin@gmail.com
2	1	negin2@gmail.com
3	2	ali@gmail.com
4	2	ali2@gmail.com

جدول phoneCustomer :

id	phone_number	Customer_id
1	0902....31	1
2	4252....99	1
3	0902....32	2
4	4252....78	2

جدول addressCustomer :

id	postalCode	address	Customer_id
1	44716	آدرس 1 کاربر 1	1
2	4156	آدرس 2 کاربر 1	1
3	99685	آدرس 1 کاربر 2	2
4	74125	آدرس 2 کاربر 2	2

2- جدول staff در اول به شکل زیر بود :

id	firstName	lastName	status	position	Manager_id	paycheck	hireDate	address	city	gender	email	phone
1	حسن	حسینی	***	حسابدار	1	1200000	2020/29/12	آدرس 1 کاربر 1	رشت	0	has@gmail.com	0902....31
											has1399@gmail.com	4252....99
2	مهسا	حجتی	*****	صندوقدار	1	50000	2020/29/12	آدرس 1 کاربر 2	رشت	1	mahsa@gmail.com	0902....32
											mahsa2@gmail.com	4252....78

با توجه به جدول بالا مشاهده می کنیم که ستون email و phone بیش از یک مقدار را در خود ذخیره کرده اند. زیرا هر یک از کارکنان می توانند یک یا چند شماره تلفن و ایمیل داشته باشند. و اگر در نظر بگیریم که چون در ستون email دو ایمیل وارد شده پس دو سطر به وجود آوریم این کار نیز مشکل را رفع نمیکند زیرا اگر کاربر ایمیل سوم یا چهارمی هم داشت به همان نسبت باید ستون های بیشتری ایجاد شود پس باید نرمال سازی انجام دهیم.

پس در نتیجه خواهیم داشت:

جدول staff :

id	firstName	lastName	status	position	Manager_id	paycheck	hireDate	address	city	gender
1	حسن	حسنى	***	حسابدار	1	1200000	2020/29/12	آدرس 1 کاربر 1	رشت	0
2	مهسا	حجتى	*****	صندوقدار	1	50000	2020/29/12	آدرس 1 کاربر 2	رشت	1

جدول emailStaff :

id	email	staff_id
1	has@gmail.com	1
2	Has1999@gmail.com	1
3	mahsa@gmail.com	2
4	mahsa2@gmail.com	2

جدول phoneStaff :

id	phone_number	Customer_id
1	0902....31	1
2	4252....99	1
3	0902....32	2
4	4252....78	2

3- جدول supplier در اول به شکل زیر بود :

id	companyName	contactName	contactFname	contactTilte	address	city	country	fax	url	email	phone
1	کاله	حسنى	على	حسابدار	آدرس 1	رشت	ایران	111	url1	1@gmail.com 2@gmail.com	0902....11 4252....73
2	اویلا	حجتى	حسن	صندوقدار	آدرس 2	رشت	ایران	112	url2	2a@gmail.com a@gmail.com	0902....00 4252....96

با توجه به جدول بالا مشاهده می کنیم که ستون email و phone بیش از یک مقدار را در خود ذخیره کرده اند. زیرا هر یک از شرکت های طرف قرارداد می توانند یک یا چند شماره تلفن و ایمیل داشته باشند. و اگر در نظر بگیریم که چون در ستون email دو ایمیل وارد شده پس دو سطر به وجود آوریم این کار نیز مشکل را رفع نمیکند زیرا اگر کاربر ایمیل سوم یا چهارمی هم داشت به همان نسبت باید ستون های بیشتری ایجاد شود پس باید نرمال سازی انجام دهیم.

جدول supplier :

id	companyName	contactName	contactFname	contactTilte	address	city	country	fax	url
1	کاله	حسنى	على	حسابدار	آدرس 1	رشت	ایران	111	url1
2	اویلا	حجتى	حسن	صندوقدار	آدرس 2	رشت	ایران	112	url2

جدول phoneSupplier :

id	phone_number	supplier_id
1	0902....11	1
2	4252....73	1
3	0902....00	2
4	4252....96	2

جدول emailSupplier :

id	email	supplier_id
1	1@gmail.com	1
2	2@gmail.com	1
3	2a@gmail.com	2
4	a@gmail.com	2

#### 4- جدول order items و order :

از آن جایی که در سامانه ی ما به دو صورت تخفیف ممکن است اعمال شود ، یکی تخفیف روی تک تک محصولات موجود در یک سفارش و دیگری تخفیف روی کل سفارش بسته به اینکه نوع کاربری مشتری چگونه است.

به عنوان مثال در جدول order داریم:

id	status	Tax	shipping	Order date	payment	shippedDate	Customer_id	discount	Staff-id
1	**	2	12	12/29/2020	0	12/29/2020	1	20%	1
2	**	9	11	12/29/2020	1	12/29/2020	2	15%	1

یا به عنوان مثال در جدول order items داریم:

id	priceperon	Quantity _id	Order-id	discount	Product-id
1	290000	2	1	5%	1
2	120000	5	1	10%	1

برای جلوگیری از افزونگی به جای این که در هر سفارش میزان و نوع هر تخفیف ذکر شود . یک جدول جدا برای تخفیف در نظر می گیریم و نوع و میزان تخفیف ها را در آن جدول مشخص می کنیم .

سپس در دو جدول order items و order فقط از id جدول تخفیف به عنوان کلید خارجی استفاده می کنیم. پس بعد از نرمال سازی خواهیم داشت :

جدول order :

id	status	Tax	shipping	Order date	payment	shippedDate	Customer_id	Discount-id	Staff-id
1	**	2	12	12/29/2020	0	12/29/2020	1	1	1
2	**	9	11	12/29/2020	1	12/29/2020	2	2	1

جدول order items :

id	priceperon	Quantity _id	Order-id	Discount-id	Product-id
1	290000	2	1	3	1
2	120000	5	1	4	1

جدول discount :

id	percent	type
1	20%	کلی
2	15%	کلی
3	5%	جزئی
4	10%	جزئی

با توجه به تحلیل های صورت گرفته همچنین تصمیم گرفتیم تخفیفی که خود شرکت برای هر محصول در نظر گرفته و در جدول product قرار داشت را نیز از آن جدول حذف کرده و وارد جدول discount کنیم. و type آن را شرکتی در نظر بگیریم. سپس در جدول محصولات از id جدول تخفیف به صورت کلید خارجی استفاده می کنیم.

همچنین در مورد دو جدول category و product-category داریم:

مثلا اگر قرار بود در جدول category ، product-id رو مشخص کنیم. چون حالت آرایه ای در sql نداریم . نمی توانیم آن را مشخص کنیم و تکرار به وجود می آید. پس اول تمام کتگوری های موجود در جدول category را مشخص می کنیم. و سپس در جدول product-category مشخص می کنیم که هر محصول با کدام یک از کتگوری ها در ارتباط است.

## کد پیاده سازی پایگاه داده

```
create database test
go
use test
go

CREATE TABLE customer
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    passwordX VARCHAR(50) NOT NULL,
    gender BIT DEFAULT(0)
);

CREATE TABLE phoneCustomer
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    phone_number NCHAR(20) NOT NULL,
    customer_id INT NOT NULL REFERENCES customer(id)
);

CREATE TABLE emailCustomer
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    email VARCHAR(100) NOT NULL,
    customer_id INT NOT NULL REFERENCES customer(id)
);

CREATE TABLE addressCustomer
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    postalCode NCHAR(10) NOT NULL,
    addressX VARCHAR(500) NOT NULL,
    customer_id INT NOT NULL REFERENCES customer(id)
);

CREATE TABLE supplier
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    companyName VARCHAR(100) NOT NULL,
    contactName VARCHAR(100) NOT NULL,
```

```

contactFname VARCHAR(100) NOT NULL,
contactTitle VARCHAR(100) NOT NULL,
addressX VARCHAR(300) NOT NULL,
city VARCHAR(50) NOT NULL,
stateX VARCHAR(100) NOT NULL,
country VARCHAR(100) NOT NULL,
postalcode NCHAR(10) NOT NULL,
fax NCHAR(20) NOT NULL,
url VARCHAR(100) NOT NULL,
);

CREATE TABLE phoneSupplier
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    phone_number NCHAR(20) NOT NULL,
    supplier_id INT NOT NULL REFERENCES supplier(id)
);

CREATE TABLE emailSupplier
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    email VARCHAR(100) NOT NULL,
    supplier_id INT NOT NULL REFERENCES supplier(id)
);

CREATE TABLE staff
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    statusX NCHAR(2) NOT NULL,
    position VARCHAR(100) NOT NULL,
    manager_id INT NOT NULL REFERENCES staff(id),
    paycheck INT NOT NULL,
    hireDate DATETIME NOT NULL,
    addressX VARCHAR(500) NOT NULL,
    city VARCHAR(100) NOT NULL,
    gender BIT DEFAULT(0)
);

CREATE TABLE phoneStaff
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    phone_number NCHAR(20) NOT NULL,
    staff_id INT NOT NULL REFERENCES staff(id)
);

```



```

CREATE TABLE emailStaff
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  email VARCHAR(100) NOT NULL,
  staff_id INT NOT NULL REFERENCES staff(id)
);

CREATE TABLE category
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  parentId INT NOT NULL REFERENCES category(id),
  title VARCHAR(100) NOT NULL,
  description VARCHAR(500) NOT NULL
);

CREATE TABLE quantity
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  number INT NOT NULL,
  lastShipment DATETIME DEFAULT (GETDATE()),
  initialValue INT NOT NULL,
  counterX INT NOT NULL DEFAULT(0),
  supplier_id INT NOT NULL REFERENCES supplier(id)
);

CREATE TABLE discount
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  percentX INT DEFAULT NULL,
  typeX VARCHAR(40) DEFAULT NULL
);

CREATE TABLE product
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  title VARCHAR(100) NOT NULL,
  typeX SMALLINT NOT NULL,
  price INT NOT NULL,
  createdAt DATETIME DEFAULT (GETDATE()),
  updatedAt DATETIME DEFAULT(NULL),
  publishedAt DATETIME DEFAULT(NULL),
  startsAt DATETIME DEFAULT(NULL),
  endsAt DATETIME DEFAULT(NULL),

```

```

discount_id INT NOT NULL REFERENCES discount(id),
quantity_id INT NOT NULL REFERENCES quantity(id));

CREATE TABLE product_category
(
    category_id INT NOT NULL REFERENCES category(id),
    product_id INT NOT NULL REFERENCES product(id)
);

CREATE TABLE product_meta
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    descriptionX VARCHAR(500) DEFAULT(NULL),
    picture VARCHAR(500) DEFAULT(NULL),
    content VARCHAR(500) DEFAULT(NULL),
    size VARCHAR(500) DEFAULT(NULL),
    packaging VARCHAR(500) DEFAULT(NULL),
    weightX INT DEFAULT(NULL),
    calories INT DEFAULT(NULL),
    sugar FLOAT DEFAULT(NULL),
    salt FLOAT DEFAULT(NULL),
    fat FLOAT DEFAULT(NULL),
    portionSize VARCHAR(500) DEFAULT(NULL),
    production_date DATETIME DEFAULT(NULL),
    expiration_date DATETIME DEFAULT(NULL),
    warnings VARCHAR(500) DEFAULT(NULL),
    product_id INT NOT NULL REFERENCES product(id)
);

CREATE TABLE orderX
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    statusX SMALLINT NOT NULL,
    tax FLOAT NOT NULL,
    shipping INT NOT NULL,
    orderDate DATETIME DEFAULT (GETDATE()),
    payment SMALLINT NOT NULL DEFAULT(0),
    shippedDate DATETIME DEFAULT(NULL),
    customer_id INT NOT NULL REFERENCES customer(id),
    discount_id INT NOT NULL REFERENCES discount(id),
    staff_id INT NOT NULL REFERENCES staff(id)
);

```

```
CREATE TABLE order_items
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  priceperone INT NOT NULL,
  quantity_id INT NOT NULL REFERENCES quantity(id),
  order_id INT NOT NULL REFERENCES orderX(id),
  discount_id INT NOT NULL REFERENCES discount(id),
  product_id INT NOT NULL REFERENCES product(id)
);
```

```
CREATE TABLE product_review
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  title VARCHAR(50) NOT NULL,
  rating SMALLINT NOT NULL,
  publishedAt DATETIME DEFAULT (GETDATE()),
  content VARCHAR(500) NOT NULL,
  response VARCHAR(500) DEFAULT(NULL),
  customer_id INT NOT NULL REFERENCES customer(id),
  productid INT NOT NULL REFERENCES product(id)
);
```

## توضیح کد پیاده سازی

جدول اول مربوط به تخفیف است که دارای id و درصد و نوع است. درصد میتواند برای یکی صفر قرار داده شود تا هنگامی که نمیخواهیم تخفیفی قرار دهیم ان را انتخاب کنیم. همچنین نوع میتواند برای مشخص کردن نوعی تخفیف دیگر باشد مانند خرید دو محصول باعث شود که سومی مجانی باشد(مثلا به صورت 2-3 نمایش دهیم) یا اینکه خرید 5 محصول باعث شود 10 درصد تخفیف داشته باشیم(مثلا به صورت 5-10% نمایش دهیم). با کمک این دو میتوانیم انواع مختلفی از تخفیف ها را داشته و استفاده کنیم.

```
CREATE TABLE discount
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  percentX INT DEFAULT NULL,
  typeX VARCHAR(40) DEFAULT NULL
);
```

در جدول مشتریان فقط نام و نام خانوادگی و رمز عبور و جنسیت را ذخیره کرده ایم زیرا ویژگی های ایمیل و تلفن و آدرس که میتوانند دارای چند مقدار باشند را برای نرمال سازی جدا کرده ایم.

```
CREATE TABLE customer
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  firstName VARCHAR(100) NOT NULL,
  lastName VARCHAR(100) NOT NULL,
  passwordX VARCHAR(50) NOT NULL,
  gender BIT DEFAULT(0)
);
```

جدول فروشنده به منظور ذخیره اطلاعات فروشنده ای ایجاد شده است که ما با او در ارتباطیم. این فروشنده لزوماً رئیس شرکت نیست پس ما مقام کاری و اطلاعات او را ذخیره میکنیم تا در صورت نیاز دسترسی داشته باشیم.

```
CREATE TABLE supplier
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  companyName VARCHAR(100) NOT NULL,
  contactName VARCHAR(100) NOT NULL,
  contactFname VARCHAR(100) NOT NULL,
  contactTitle VARCHAR(100) NOT NULL,
  addressX VARCHAR(300) NOT NULL,
  city VARCHAR(50) NOT NULL,
  stateX VARCHAR(100) NOT NULL,
  country VARCHAR(100) NOT NULL,
  postalcode NCHAR(10) NOT NULL,
  fax NCHAR(20) NOT NULL,
  url VARCHAR(100) NOT NULL,
);
```

جدول کارمندان اطلاعاتی تمامی کارمندان را ذخیره میکند و در آن علاوه بر اطلاعات معلوم و پایه status که وضعیت فعلی را نشان میدهد (مثلاً در حال کار است یا مرخصی است و یا شیفت کاری تمام شده و حالت های دیگر) و اعداد ذخیره شده در آن میتواند با توجه به نیاز معنا دهی شوند را داریم. همچنین position سمت کاری و manager\_id رئیسی را نشان میدهد که کارمند مورد نظر زیر نظر او کار میکند.

```

CREATE TABLE staff
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  firstName VARCHAR(100) NOT NULL,
  lastName VARCHAR(100) NOT NULL,
  statusX NCHAR(2) NOT NULL,
  position VARCHAR(100) NOT NULL,
  manager_id INT NOT NULL REFERENCES staff(id),
  paycheck INT NOT NULL,
  hireDate DATETIME NOT NULL,
  addressX VARCHAR(500) NOT NULL,
  city VARCHAR(100) NOT NULL,
  gender BIT DEFAULT(0)
);

```

قسمت دسته بندی دارای یک ستون parentId است که به همین جدول رفرنس دهی شده تا ساختار تو در تو را برای ما ایجاد کند. اگر بخواهیم یک دسته بندی دارای parent نباشد میتوانیم به id خودش رفرنس دهیم تا در واقع parent خودش باشد.

```

CREATE TABLE category
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  parentId INT NOT NULL REFERENCES category(id),
  title VARCHAR(100) NOT NULL,
  description VARCHAR(500) NOT NULL
);

```

برای اجازه دادن به مشتری برای ذخیره سازی چند تلفن از این جدول استفاده کرده ایم که به جدول مشتریان رفرنس دارد.

```

CREATE TABLE phoneCustomer
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  phone_number NCHAR(20) NOT NULL,
  customer_id INT NOT NULL REFERENCES customer(id)
);

```

برای اجازه دادن به مشتری برای ذخیره سازی چند ایمیل از این جدول استفاده کرده ایم که به جدول مشتریان رفرنس دارد.

```
CREATE TABLE emailCustomer
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  email VARCHAR(100) NOT NULL,
  customer_id INT NOT NULL REFERENCES customer(id)
);
```

برای اجازه دادن به مشتری برای ذخیره سازی چند ادرس از این جدول استفاده کرده ایم که به جدول مشتریان رفرنس دارد.

```
CREATE TABLE addressCustomer
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  postalCode NCHAR(10) NOT NULL,
  addressX VARCHAR(500) NOT NULL,
  customer_id INT NOT NULL REFERENCES customer(id)
);
```

برای اجازه دادن به فروشنده برای ذخیره سازی چند تلفن از این جدول استفاده کرده ایم که به جدول فروشنده رفرنس دارد.

```
CREATE TABLE phoneSupplier
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  phone_number NCHAR(20) NOT NULL,
  supplier_id INT NOT NULL REFERENCES supplier(id)
);
```

برای اجازه دادن به فروشنده برای ذخیره سازی چند ایمیل از این جدول استفاده کرده ایم که به جدول فروشنده رفرنس دارد.

```
CREATE TABLE emailSupplier
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  email VARCHAR(100) NOT NULL,
  supplier_id INT NOT NULL REFERENCES supplier(id)
);
```

برای اجازه دادن به کارمند برای ذخیره سازی چند تلفن از این جدول استفاده کرده ایم که به جدول کارمند رفرنس دارد.

```
CREATE TABLE phoneStaff
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    phone_number NCHAR(20) NOT NULL,
    staff_id INT NOT NULL REFERENCES staff(id)
);
```

برای اجازه دادن به کارمند برای ذخیره سازی چند ایمیل از این جدول استفاده کرده ایم که به جدول کارمند رفرنس دارد.

```
CREATE TABLE emailStaff
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    email VARCHAR(100) NOT NULL,
    staff_id INT NOT NULL REFERENCES staff(id)
);
```

برای ثبت اطلاعات کلی سفارش از این جدول استفاده شده. Status نشان دهنده وضعیت فعلی (در حال پردازش یا بسته بندی یا فرستاده شده و غیره) است.

```
CREATE TABLE orderX
(
    id INT NOT NULL PRIMARY KEY IDENTITY,
    statusX SMALLINT NOT NULL,
    tax FLOAT NOT NULL,
    shipping INT NOT NULL,
    orderDate DATETIME DEFAULT (GETDATE()),
    payment SMALLINT NOT NULL DEFAULT(0),
    shippedDate DATETIME DEFAULT(NULL),
    customer_id INT NOT NULL REFERENCES customer(id),
    discount_id INT NOT NULL REFERENCES discount(id),
    staff_id INT NOT NULL REFERENCES staff(id)
);
```

این جدول شبه انبار است. در ابتدای کار وقتی محصول میرسد lastShipment تنظیم میشود و number و initialValue هر دو برابر تعداد دریافتی میشوند اما با هر بار خرید counter یکی بالا میرود و از تعداد number تعدادی کم میشود در واقع counter برای این است که بفهمیم در چند بار خرید تعدادی از محصول رفته.

```

CREATE TABLE quantity
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  number INT NOT NULL,
  lastShipment DATETIME DEFAULT (GETDATE()),
  initialValue INT NOT NULL,
  counterX INT NOT NULL DEFAULT(0),
  supplier_id INT NOT NULL REFERENCES supplier(id)
);

```

ویژگی های اصلی محصول اینجا نمایش داده میشوند type در واقع نوع محصول را نمایش میدهد. createdAt زمانی را نشان میدهد که این محصول ایجاد شده و updatedAt برای زمانی است که اصلاحی انجام پذیرد. publishedAt زمانی است که محصول به نمایش گذاشته شده و startsAt و endsAt برای بازه ای که قرار است به نمایش گذاشته شوند

```

CREATE TABLE product
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  title VARCHAR(100) NOT NULL,
  typeX SMALLINT NOT NULL,
  price INT NOT NULL,
  createdAt DATETIME DEFAULT (GETDATE()),
  updatedAt DATETIME DEFAULT(NULL),
  publishedAt DATETIME DEFAULT(NULL),
  startsAt DATETIME DEFAULT(NULL),
  endsAt DATETIME DEFAULT(NULL),
  discount_id INT NOT NULL REFERENCES discount(id),
  quantity_id INT NOT NULL REFERENCES quantity(id)
);

```

این جدول میانه ای است که رابطه محصول و دسته بندی را برقرار میکند.

```

CREATE TABLE product_category
(
  category_id INT NOT NULL REFERENCES category(id),
  product_id INT NOT NULL REFERENCES product(id)
);

```



توضیحات ریز محصول در آن قرار داده شده و محصول می‌تواند هر یک از این ویژگی‌ها را داشته باشد یا نداشته باشد.

```
CREATE TABLE product_meta
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  descriptionX VARCHAR(500) DEFAULT(NULL),
  picture VARCHAR(500) DEFAULT(NULL),
  content VARCHAR(500) DEFAULT(NULL),
  size VARCHAR(500) DEFAULT(NULL),
  packaging VARCHAR(500) DEFAULT(NULL),
  weightX INT DEFAULT(NULL),
  calories INT DEFAULT(NULL),
  sugar FLOAT DEFAULT(NULL),
  salt FLOAT DEFAULT(NULL),
  fat FLOAT DEFAULT(NULL),
  portionSize VARCHAR(500) DEFAULT(NULL),
  production_date DATETIME DEFAULT(NULL),
  expiration_date DATETIME DEFAULT(NULL),
  warnings VARCHAR(500) DEFAULT(NULL),
  product_id INT NOT NULL REFERENCES product(id)
);
```

برای هر یک از محصولات فاکتور هست و تعداد و محصول را مشخص می‌کند.

```
CREATE TABLE order_items
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  priceperone INT NOT NULL,
  quantity_id INT NOT NULL REFERENCES quantity(id),
  order_id INT NOT NULL REFERENCES orderX(id),
  discount_id INT NOT NULL REFERENCES discount(id),
  product_id INT NOT NULL REFERENCES product(id)
);
```

برای نمایش نظرات کاربران در مورد هر محصول است.

```
CREATE TABLE product_review
(
  id INT NOT NULL PRIMARY KEY IDENTITY,
  title VARCHAR(50) NOT NULL,
  rating SMALLINT NOT NULL,
  publishedAt DATETIME DEFAULT (GETDATE()),
  content VARCHAR(500) NOT NULL,
  response VARCHAR(500) DEFAULT(NULL),
  customer_id INT NOT NULL REFERENCES customer(id),
  productid INT NOT NULL REFERENCES product(id)
);
```

### کلیدهای اصلی و فرعی :

هر جدول باید دارای یک کلید اصلی باشد. کلید اصلی یک مقدار خاص و غیرتهی (non-null) می باشد که یک ردیف را در دیتا بیس شناسایی می کند. یکی از مهمترین موضوعات مورد بحث در دنیای SQL Server تصمیم در مورد نوع داده ای است که بهترین کلید اصلی را می سازد. برای تعریف کلید اصلی در هر جدول از کلمه کلیدی Primary Key بعد از نام ستون استفاده می کنیم.

کلید خارجی (foreign key) صفت خاصه ای در یک جدول است که در جدول دیگر نقش کلید اصلی یا کاندید را بازی کند. کلید خارجی ارتباط بین دو جدول را برقرار می کند. کلید خارجی بر خلاف کلید اصلی می تواند تکراری یا null باشد و ممکن است در یک جدول بیشتر از یک کلید خارجی وجود داشته باشد. جدول شامل کلید خارجی را گاهی جدول فرزند و جدولی که به آن ارجاع دارد را جدول والد می نامند.

معرفی کلید های اصلی پروژه :

هر یک از جداول پروژه حاوی یک کلید اصلی می باشد. تمام صفت های خاصه ی پروژه با نام id را به عنوان کلید اصلی ذخیره کردیم.

معرفی کلیدهای خارجی :

customer\_id در جدول phoneCustomer کلید خارجی است و به id جدول customer رفرنس داده می شود.

customer\_id در جدول emailCustomer کلید خارجی است و به id جدول customer رفرنس داده می شود.

customer\_id در جدول addressCustomer کلید خارجی است و به id جدول customer رفرنس داده می شود.

supplier\_id در جدول phoneSupplier کلید خارجی است و به id جدول supplier رفرنس داده می شود.

supplier\_id در جدول emailSupplier کلید خارجی است و به id جدول supplier رفرنس داده می شود.

staff\_id در جدول phoneStaff کلید خارجی است و به id جدول staff رفرنس داده می شود.

staff\_id در جدول emailStaff کلید خارجی است و به id جدول staff رفرنس داده می شود.

parentId در جدول category کلید خارجی است و به id جدول category رفرنس داده می شود.

supplier\_id در جدول quantity کلید خارجی است و به id جدول supplier رفرنس داده می شود.

discount\_id در جدول product کلید خارجی است و به id جدول discount رفرنس داده می شود.

quantity\_id در جدول product کلید خارجی است و به id جدول quantity رفرنس داده می شود.

category\_id در جدول product\_category کلید خارجی است و به id جدول category  
رفرنس داده می شود.

product\_id در جدول product\_category کلید خارجی است و به id جدول product  
رفرنس داده می شود.

product\_id در جدول product\_meta کلید خارجی است و به id جدول product  
رفرنس داده می شود.

customer\_id در جدول orderX کلید خارجی است و به id جدول customer  
رفرنس داده می شود.

discount\_id در جدول orderX کلید خارجی است و به id جدول discount  
رفرنس داده می شود.

staff\_id در جدول orderX کلید خارجی است و به id جدول staff  
رفرنس داده می شود.

quantity\_id در جدول order\_items کلید خارجی است و به id جدول quantity  
رفرنس داده می شود.

order\_id در جدول order\_items کلید خارجی است و به id جدول orderX  
رفرنس داده می شود.

discount\_id در جدول order\_items کلید خارجی است و به id جدول discount  
رفرنس داده می شود.

product\_id در جدول order\_items کلید خارجی است و به id جدول product  
رفرنس داده می شود.

customer\_id در جدول product\_review کلید خارجی است و به id جدول customer  
رفرنس داده می شود.

discount\_id در جدول product\_review کلید خارجی است و به id جدول product  
 رفرنس داده می شود.

## الحاق داده های واقعی :

جدول customer :

	id	firstName	lastName	passwordX	gender
1	1	Yasaman	Chegini	1234jnv	1
2	2	Sara	Salary	6789kbj	1
3	3	samira	Jam	67dkfjvj	1
4	4	nastaran	Javidan	ksdhfi8kbj	1
5	5	Yasin	Chegini	bcgt64	0
6	6	Bano	Gholami	67ggcd5	1
7	7	Ali	Chegini	jkvf567	0

جدول discount :

	id	percentX	typeX
1	1	15	NULL
2	2	NULL	1_2
3	3	10	NULL
4	4	NULL	2_3
5	5	45	NULL

جدول emailStaff :

	id	email	staff_id
1	1	staff4@gmail.com	4
2	2	staff1@gmail.com	1
3	3	staff2@gmail.com	2
4	4	staff2-2gmail.com	2
5	5	staff3@gmail.com	3
6	6	staff6@gmail.com	6
7	7	staff6-2@gmail.com	6
8	8	staff7@gmail.com	7
9	9	staff8@gmail.com	1
10	10	staff5@gmail.com	5

جدول emailCustomer :

	id	email	customer_id
1	1	nastaran@gmail.com	4
2	2	Yasaman@gmail.com	1
3	3	Sara@gmail.com	2
4	4	Sara2@gmail.com	2
5	5	samira@gmail.com	3
6	6	Bano@gmail.com	6
7	7	Bano2@gmail.com	6
8	8	Ali@gmail.com	7
9	9	yahoooooooo@gmail.com	5
10	10	Yasin@gmail.com	5

جدول emailSupplier :

	id	email	supplier_id
1	1	supplier4@gmail.com	4
2	2	supplier1@gmail.com	1
3	3	supplier2@gmail.com	2
4	4	supplier2-2gmail.com	2
5	5	supplier3@gmail.com	3
6	6	supplier6@gmail.com	6
7	7	supplier6-2@gmail.com	6
8	8	supplier7@gmail.com	7
9	9	supplier8@gmail.com	7
10	10	supplier5@gmail.com	5

جدول orderX :

	id	statusX	tax	shipping	orderDate	payment	shippedDate	customer_id	discount_id	staff_id
1	1	1	7	12000	1905-07-04 00:00:00.000	1	1905-07-03 00:00:00.000	1	1	2
2	2	1	9	50000	1905-07-01 00:00:00.000	1	1905-06-20 00:00:00.000	2	3	3
3	3	2	7	34000	1905-06-21 00:00:00.000	1	1905-06-19 00:00:00.000	3	2	1

جدول order\_items :

	id	priceperone	quantity_id	order_id	discount_id	product_id
1	1	32000	6	1	1	6
2	2	30000	9	1	4	9
3	3	45000	3	2	1	3
4	4	50000	7	2	3	7
5	5	60000	8	2	3	8
6	6	8000	10	3	2	10
7	7	10000	2	3	1	2
8	8	50000	7	3	3	7

جدول phoneStaff :

	id	phone_number	staff_id
1	1	9112366600	3
2	2	9112450022	1
3	3	9115560081	2
4	4	1330007613	5
5	5	9388002123	4
6	6	2133005444	4
7	7	1300354670	6
8	8	4400100032	7
9	9	9114005123	4

جدول phoneCustomer :

	id	phone_number	customer_id
1	1	9112345678	3
2	2	9112223322	1
3	3	9115557781	2
4	4	1333333113	5
5	5	9388882133	4
6	6	2133445678	4
7	7	1332345670	6
8	8	4486994032	5
9	9	9114456123	4

جدول product :

	id	title	typeX	price	createdAt	updatedAt	publishedAt	startsAt	endsAt	discount_id	quantity_id
1	1	MayeZarfShoi	3	20000	1903-11-01 00:00:00.000	NULL	2020-04-27 00:00:00.000	2021-01-12 00:00:00.000	2022-02-26 00:00:00.000	1	1
2	2	Noshabe	5	10000	1901-12-01 00:00:00.000	NULL	2020-03-11 00:00:00.000	2021-01-12 00:00:00.000	2023-01-24 00:00:00.000	1	2
3	3	Berenje	2	45000	1901-04-12 00:00:00.000	NULL	2020-12-16 00:00:00.000	2021-01-11 00:00:00.000	2022-01-12 00:00:00.000	1	3
4	4	Mast	5	12000	1900-12-16 00:00:00.000	NULL	2020-12-01 00:00:00.000	2021-01-10 00:00:00.000	2024-01-09 00:00:00.000	2	4
5	5	Dogh	5	17000	1900-04-18 00:00:00.000	NULL	2020-12-01 00:00:00.000	2021-03-11 00:00:00.000	2024-04-27 00:00:00.000	4	5
6	6	Zaferan	1	32000	1900-12-16 00:00:00.000	NULL	2020-03-29 00:00:00.000	2021-04-27 00:00:00.000	2022-10-08 00:00:00.000	1	6
7	7	Pesteh	1	50000	1900-03-08 00:00:00.000	2020-02-25 00:00:00.000	2021-12-01 00:00:00.000	2021-01-12 00:00:00.000	2023-01-10 00:00:00.000	3	7
8	8	badamHendi	1	60000	1903-10-31 00:00:00.000	NULL	2020-12-01 00:00:00.000	2021-01-20 00:00:00.000	2026-01-10 00:00:00.000	3	8
9	9	alchole	4	30000	1900-04-10 00:00:00.000	NULL	2020-12-01 00:00:00.000	2021-03-11 00:00:00.000	2022-02-01 00:00:00.000	4	9
10	10	dastmalkaghazi	5	8000	1900-05-20 00:00:00.000	NULL	2020-12-01 00:00:00.000	2021-12-01 00:00:00.000	2024-01-16 00:00:00.000	2	10

جدول phoneSupplier :

	id	phone_number	supplier_id
1	1	9112366678	3
2	2	9112455322	1
3	3	9115566781	2
4	4	1333557613	5
5	5	9388882123	4
6	6	2133445444	7
7	7	1332354670	6
8	8	4486100032	7
9	9	9114445123	4

جدول product\_meta :

	id	descriptionX	picture	content	size	packaging	weightX	calories	sugar	salt	fat	portionSize	production_date	expiration_date	warnings	product_id
1	1	Kafe faravan Ba charibari bala	NULL	content1	5*2*12	1	200	NULL	NULL	NULL	NULL	200	2020-07-10 00:00:00.000	2021-06-24 00:00:00.000	NULL	1
2	2	Tame vaghei	NULL	content2	5*5*10	1	900	500	1.2	0.3	0.3	900	2020-06-26 00:00:00.000	2022-03-27 00:00:00.000	NULL	2

جدول product\_review :

	id	title	rating	publishedAt	content	response	customer_id	productid
1	1	Khob	8	2021-01-04 22:22:51.153	content1	response from manager 1	1	2
2	2	Aali	10	2021-01-04 22:22:51.153	content2	response from manager 2	2	1
3	3	Khob	9	2021-01-04 22:22:51.153	content3	response from manager 3	3	4
4	4	pishnahad mishe	10	2021-01-04 22:22:51.153	content4	response from manager 4	4	3
5	5	motevaset	5	2021-01-04 22:22:51.153	content5	response from manager 5	5	6
6	6	Khob	8	2021-01-04 22:22:51.153	content6	response from manager 6	6	5
7	7	Khob	9	2021-01-04 22:22:51.153	content7	response from manager 7	1	3
8	8	narazi	3	2021-01-04 22:22:51.153	content8	response from manager 8	3	5



: جدول product\_category

	category_id	product_id
1	6	1
2	3	2
3	1	3
4	2	4
5	2	5
6	7	6
7	7	7
8	7	8
9	4	9
10	5	10

: جدول staff

	id	firstName	lastName	statusX	position	manager_id	paycheck	hireDate	addressX	city	gender
1	1	Ali	Alavi	1	sandogdar	1	500000	2012-06-24 00:00:00.000	address1	Rasht	0
2	2	Mahdi	Asgari	1	Moaven sandogdar	1	400000	2016-04-04 00:00:00.000	address2	Roudbar	0
3	3	Ali	Asil	2	sandogdar	1	530000	2020-01-13 00:00:00.000	address3	Anzali	0
4	4	Arasto	Aamel	2	Moaven sandogdar	1	340000	2017-04-27 00:00:00.000	address4	Rasht	0
5	5	Milad	Mohammadi	1	sandogdar	1	120000	2014-05-08 00:00:00.000	address5	Langarood	0
6	6	Setayesh	Sodayi	1	sandogdar	1	410000	2018-01-19 00:00:00.000	address6	Rasht	1
7	7	Shima	Fariba	4	Moaven sandogdar	1	210000	2019-01-15 00:00:00.000	address7	Anzali	1

: جدول quantity

	id	number	lastShipment	initialValue	counterX	supplier_id
1	1	20	2021-01-04 22:22:51.093	3	17	1
2	2	45	2021-01-04 22:22:51.093	5	40	2
3	3	10	2021-01-04 22:22:51.093	1	9	3
4	4	32	2021-01-04 22:22:51.093	30	1	4
5	5	80	2021-01-04 22:22:51.093	80	0	5
6	6	30	2021-01-04 22:22:51.093	30	0	6
7	7	80	2021-01-04 22:22:51.093	40	40	7
8	8	20	2021-01-04 22:22:51.093	3	17	1
9	9	20	2021-01-04 22:22:51.093	3	17	1
10	10	20	2021-01-04 22:22:51.093	3	17	1

جدول addressCustomer :

	id	postalCode	addressX	customer_id
1	1	4242425677	Rassht-Janbazan	4
2	2	5678934277	Anzali-Emamstreet	1
3	3	6662256444	Roudbar-Bolvarvahdat	2
4	4	1234598734	Tehran-Enghelabstreet	6
5	5	7228652914	Rassht-Janbazan	7
6	6	7223152914	Rasht-Glsar	3
7	7	7228653231	Rasht-Glsar	5

جدول supplier :

	id	companyName	contactName	contactFname	contactTitle	addressX	city	stateX	country	postalcode	fax	url
1	1	Kaleh	Ali	Alizadeh	Monshi Anbar	address1	Rast	Guilan	Iran	44567654	1334345521	www.kaleh.ir
2	2	Pegah	Amir	Amiriniya	Monshi	address2	Tehran	Tehran	Iran	42367654	1334300521	www.Pegah.ir
3	3	Koka	Mona	Nili	Monshi Modir	address3	Rast	Guilan	Iran	12343214	1330045521	www.Koka.ir
4	4	Tabiaat	MAhdi	Yahyayi	Anbardar	address4	Esfahan	Esfahan	Iran	98706578	1334311521	www.Tabiaat.ir
5	5	Mohsen	Sara	Jahani	Modir Frosh	address5	Yazd	Yazd	Iran	32189657	1334345510	www.Mohsen.ir
6	6	Pepsi	Donya	Abdi	Moaven Modir	address6	Rast	Guilan	Iran	44563907	1334345020	www.Pepsi.ir
7	7	Oila	Yasin	Yasini	Visitor	address7	Shiraz	Fars	Iran	90112657	1334111521	www.Oila.ir

جدول category :

	id	parentId	title	description
1	1	1	MavadeKHoraki	description1
2	2	1	Labaniyat	description1-2
3	3	1	noshidani	description1-3
4	4	4	Behdashti	description4
5	5	4	ZedOfonikonande	description4-1
6	6	4	Shoyande	description4-2
7	7	1	khoshkbar	description1-4