



دانشگاه شهید بهشتی تهران

مستند مدل داده

تهیه و تنظیم:

میثم ولوئیان ۹۱۲۱۳۱۳۲

شیرین طهماسبی ۹۱۱۱۳۰۲۹

درس مربوطه: مهندسی نرم افزار ۲

استاد راهنما: دکتر صادق علی اکبری

گروه آموزشی مهندسی کامپیوتر

پاییز ۱۳۹۴

مدل^۱ یک اصطلاح کلی است که در موارد مختلفی کاربرد دارد. مدل از لحاظ معنایی هم معنایهای مختلفی می‌دهد. اما معنای مدل در مهندسی نرم‌افزار، ارائه یک شمای دیداری و استفاده از تعاریف و اشکال مختلف برای فهماندن موضوعی به مخاطب است. برای مثال ما می‌آئیم برای مثال ما می‌آئیم و برای داده‌های درون دیتابیس خود یک مدل ارائه می‌کنیم. این مدل می‌تواند رابطه‌ای باشد، مثلاً از با استفاده از اشکال مختلف، رابطه‌ی بین موجودیت‌ها را به نمایش درمی‌آوریم. این مدل می‌تواند شی‌گرا نیز باشد. یعنی مثلاً موجودیت‌ها را بصورت شی ببینیم و ارتباط بین اشیا را با استفاده از اشکال قراردادی به نمایش درآوریم.

در این مستند قصد داریم تا با استفاده از یک سری تعاریف قراردادی و اشکال مختلف، داده‌های درون سیستم خود را مدل کنیم. دلیل مدل کردن داده‌ها این است که بدانیم چه داده‌هایی در این سیستم وجود دارد و ارتباط این داده‌ها با یکدیگر چیست. با داشتن اطلاعات درمورد داده‌هایی که در سیستم جریان دارد، ما می‌توانیم به درک درستی از سیستم در دست طراحی برسیم.

مدل داده^۲ را نباید با مدل اطلاعات^۳ اشتباه گرفت. در مدلی که در این مستند قرار است ارائه شود، ما نمی‌خواهیم جریان اطلاعات و یا رابطه‌ی بین اطلاعات مختلف را شناسایی کنیم. بلکه می‌خواهیم داده‌ها را در سیستم مدل کنیم. یعنی این داده‌ها هنوز به اطلاع تبدیل نشده‌اند؛ یا اینکه برای ما معنا و مفهوم خاصی ندارند. همانطور که می‌دانیم اطلاع از جفت name-value بدست می‌آید و داده فقط از valueها تشکیل شده‌است. بنابراین ما فقط داده داریم و نام این داده برای ما اهمیت ندارد. لذا ما با مدل داده سر و کار داریم و نه با مدل اطلاع.

model^۱

Data model^۲

Information model^۳

فهرست مطالب

۲	اهداف مدل‌سازی:	۴
۳	مدل‌سازی داده های فروشگاه دیجی کالای دانشگاه شهید بهشتی	۴
۳,۱	یک نکته مهم	۵
۴	مدل سازی داده ها	۵
۴,۱	مدل سازی داده های مربوط به یوزرها	۵
۴,۱,۱	چندی ارتباط:	۵
۴,۱,۲	توضیح:	۶
۴,۲	مدل سازی داده های مربوط به سبد کالا	۷
۴,۳	مدل سازی داده های مربوط به کالاها(لیست)	۹
۴,۳,۱	چندی ارتباط:	۹
۴,۳,۲	توضیح:	۹
۴,۴	مدل سازی داده های مربوط به کالاها(گوشی ها)	۱۱
۴,۴,۱	چندی ارتباط:	۱۱
۴,۴,۲	توضیح:	۱۱
۴,۵	ارتباط بین موجودیت ها	۱۳
۴,۵,۱	نکات مهم مربوط به ER	۱۴
۴,۵,۲	چندی ارتباط	۱۴
۵	نرمال سازی	۱۷
۵,۱	بررسی 1NF بودن	۱۷
۵,۲	بررسی 2NF بودن	۱۷
۵,۳	بررسی 3NF بودن	۱۸
۵,۴	بررسی BCNF بودن	۱۸

۱ اهداف مدل سازی:

بطور کلی مدل کار را برای انسان راحت تر می کند. برای مثال برای فهم نیازمندی های مشتری، سند نیازمندی ها یک منبع مهم و معتبر است. ولی خواندن مستندات کمی مشکل است و وقت زیادی از انسان می گیرد و در مقابل حجم زیادی که دارد، اطلاعات زیادی به دست نمی آید. از آنجا که اشکال می توانند خیلی گویاتر باشند، لذا ما می آئیم و نیازمندی های خود را بصورت های مختلف، اعم از sequence diagram و یا prototype مدل می کنیم. با اینکار فهم بهتری از نیازمندی ها بدست می آید و برنامه نویسان می فهمند که قرار است چه سناریویی را طراحی بکنند و مشتری از آن ها چه می خواهد. بنابراین مدلسازی فهم بهتری از پروژه به ما می دهد. هرگاه که به درک درستی از پروژه رسیدیم و خواستیم وارد جزئیات شویم، آنگاه به سراغ سند نیازمندی ها می رویم و جزئیات پروتوتایپ را از آنجا بدست می آوریم.

در داده نیز همین مشکل وجود دارد. داده های زیادی در سیستم وجود دارند که با هم رابطه دارند. اگر ما بتوانیم روابط بین داده ها را مدل کنیم، ساختار کلی داده ها را بهتر درک می کنیم و یک دید کلی از داده های درون سیستم بدست می آوریم. بنابراین می آئیم و اشکالی قراردادی را برای مدل کردن داده ها تعریف می کنیم و با استفاده از این اشکال داده های درون سیستم را مدل می کنیم.

۲ مدل سازی داده های فروشگاه دیجی کالای دانشگاه شهید بهشتی

در این مستند، برای مدل سازی داده های درون سیستم، از مدل رابطه ای استفاده کرده ایم. یکی از مزیت های بزرگ مدل رابطه ای نسبت به دیگر مدل سازی ها سادگی آن است. در مدل رابطه ای نیاز به تخصص برای فهم ارتباط بین داده ها وجود ندارد. این مدل از اشکال و مفاهیم ساده ای برای مدلسازی استفاده می کند که فهم آن آسان است و استفاده از این مدل نیز به همین دلیل گسترده و فراوان است.

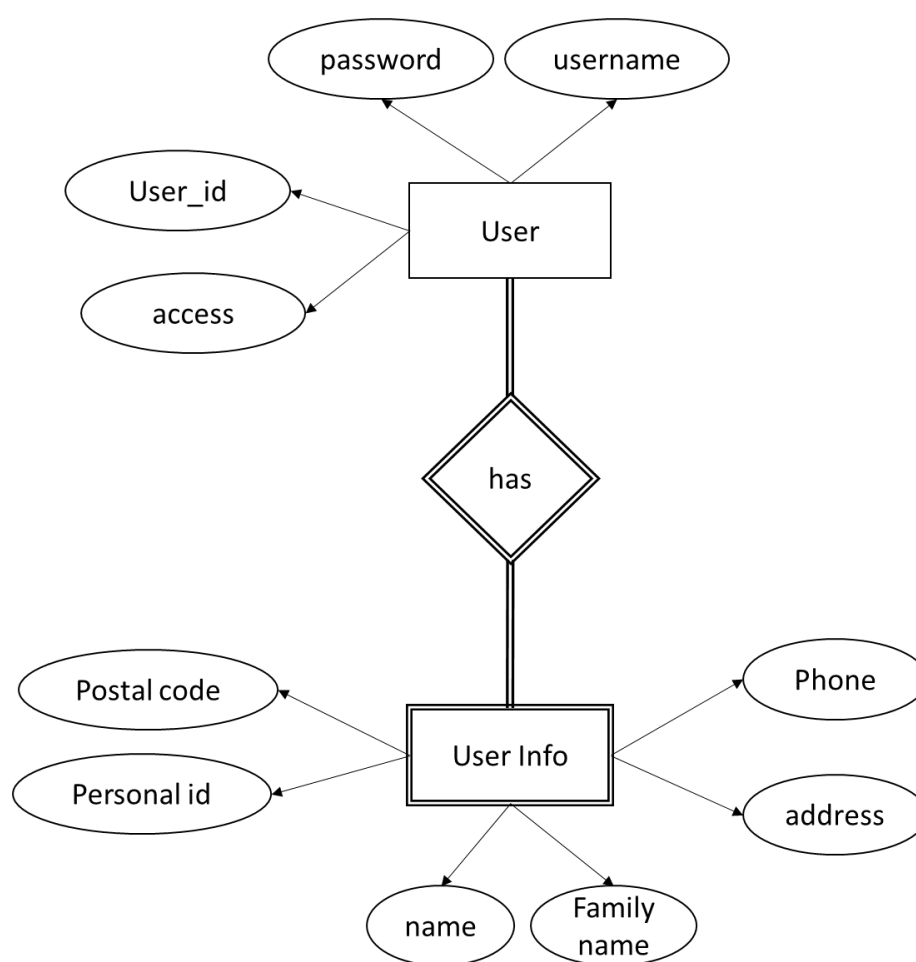
در این مستند قصد نداریم که به معرفی اشکال مدل رابطه ای و یا مفاهیم هر یک از اشکال برای مدلسازی بپردازیم. بلکه فرض ما بر این است که خواننده با مفاهیم هر یک از اشکال آشنایی دارد و درک درستی از موجودیت ها و نحوه ی ارتباط آن ها دارد. همانطور که از نام این مستند پیداست، در این مستند فقط به مدل سازی داده می پردازیم. خواننده محترم می تواند برای یادگیری مدل رابطه ای به دیگر کتب مربوطه مراجعه کند.

۲،۱ یک نکته مهم

سیستم فروشگاه دیجی کالا، بسیار بزرگ و گسترده می باشد. لذا صفحات گنجایش اینکه کل مدل داده را در یک صفحه رسم کنیم، ندارند. لذا مجبوریم برای هر قسمت، مدل سازی را رسم کنیم. در پایان این تکه مدل ها را به مربوط خواهیم کرد و به ذکر جزئیات ارتباط بین آن ها خواهیم پرداخت.

۳ مدل سازی داده ها

۳،۱ مدل سازی داده های مربوط به یوزرها



شکل ۱ مدل سازی داده مربوط به یوزرها

۳،۱،۱ چندی ارتباط:

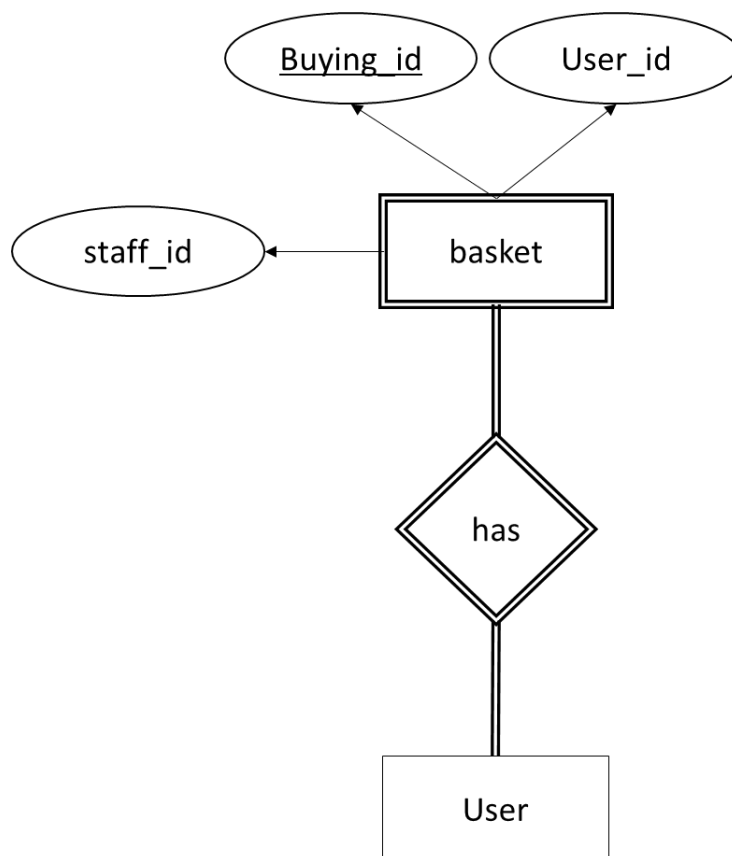
ارتباط بین اطلاعات یوزر و اطلاعات شخصی یک یوزر در اینجا ۱ به ۱ است.

۳،۱،۲ توضیح:

در این مدل هر یوزر اطلاعات شخصی دارد و اطلاعات یوزری. ما آمدیم و این دو را از هم جدا کردیم. اطلاعات شخصی یوزر یک موجودیت ضعیف است و در صورتی معنا دارد که یوزری وجود داشته باشد. فلسفه داشتن اطلاعات شخصی یوزر این است که فروشگاه بتواند اجناس خریداری شده توسط هر یوزر را برای او ارسال کند. بخاطر همین ما شماره تلفن و آدرس یوزر را نیز نگه می‌داریم. گرچه در فرمی که برای ارسال وجود دارد از یوزر می‌خواهیم که آدرس ارسال را وارد کند و اگر چیزی را وارد نکرد، آدرس پیش‌فرض او همان آدرسی خواهد بود که در دیتابیس و سطر مربوط به یوزر مربوطه وجود دارد.

هر یوزر سبد کالا هم دارد که در ادامه به مباحث مربوط به سبد کالا نیز خواهیم پرداخت. سبد کالا یک موجودیت کلی نیست و هر یوزر برای خود سبدي دارد. اینطور به نظر می‌رسد که باید برای هر یوزر یک جدول ایجاد کنیم و کالاهایی را که خریده در آن بگذاریم. اما ما یک سبدکالای کلی ایجاد می‌کنیم و اطلاعات مربوط به هر یوزر را در آن می‌گذاریم. یعنی یک جدول کلی به نام سبد داریم که کالاهای همه یوزرها در آن قرار دارد. اجازه دهید در ادامه به بررسی مفصل این موضوع بپردازیم.

۳,۲ مدل سازی داده های مربوط به سبد کالا



شکل ۲ مدل سازی داده های مربوط به سبد کالا

۳,۳ چندی ارتباط:

ارتباط بین یوزر و سبد خرید او ۱ به ۱ خواهد بود. دلیل ۱ به ۱ بودن این ارتباط در ادامه توضیح داده شده است.

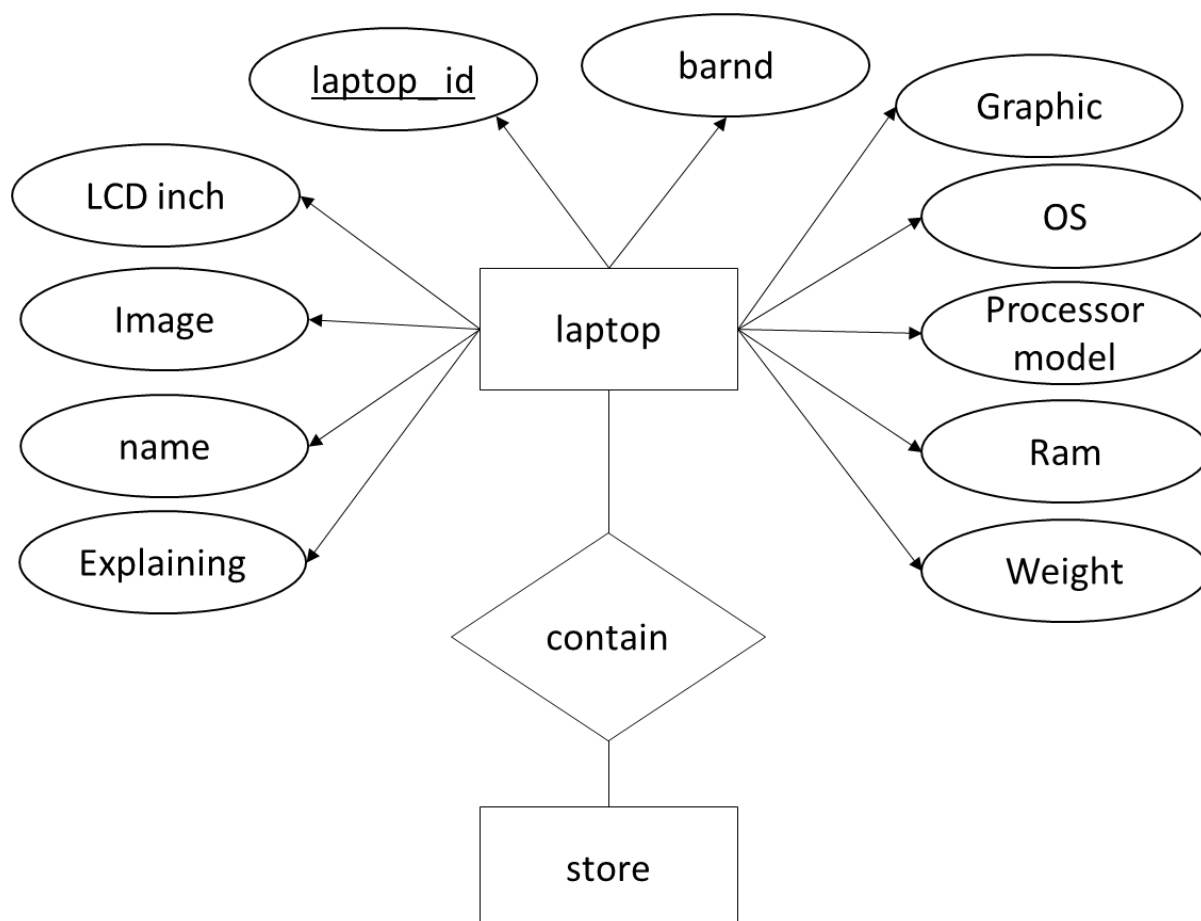
توضیح:

در شکل بالا، هر یوزر سبد کالا دارد و سبد کالا موجودیت ضعیف است. زیرا اگر یک یوزر از بین برود، سبد کالا از بین می رود ولی اطلاعات مربوط به خرید او همچنان باقی می ماند. اینکار ما را قادر می سازد تا بتوانیم گزارش گیری کاملی از سبد کالا داشته باشیم و همچنین در عمل، سبد کالا یکی است و برای همه ی مشتریان و هر مشتری با ذخیره id مربوط به خود در سبد کالا، کالاهای خود را از دیگران جدا می کند. بنابراین اینطور نیست

که یک جدول برای کالاهای هر یوزر ایجاد کنیم. بلکه یک جدول کلی ایجاد می‌کنیم و همه‌ی کالاهای خریداری-
شده را در آن نگه می‌داریم. ولی اینطور به نظر می‌رسد که هر یوزر یک سبد کالا دارد. ارتباط ۱ به ۱ بین یوزر و
سبد کالا این مورد را اثبات می‌کند.

۳،۴ مدل سازی داده های مربوط به کالاها (لپتاپ)

دو نوع کالا داریم. لپتاپ و گوشی. ابتدا نحوه ذخیره داده ها در لپتاپ را مدل می کنیم و سپس به سراغ مدل سازی دیگر کالاها می رویم.



شکل ۳،۳،۳ مدل سازی داده های مربوط به کالاها (لپتاپ)

۳،۴،۱ چندی ارتباط:

ارتباط بین انبار و لپتاپ چند به چند است. اما چون انبار برای ما مهم نیست لذا هیچ مشخصاتی را که مربوط به انبار باشد در دیتابیس ذخیره نمی کنیم. چون ما نمی خواهیم یک سیستم انبارداری بزنیم. پس تنها مشخصات لپتاپ برای ما مهم است و اینکه هر لپتاپ در کدام انبار است برای ما مهم نیست. پس فروشگاه لپتاپ هایی دارد که مشخصات خاص خود را دارند.

۳،۴،۲ توضیح:

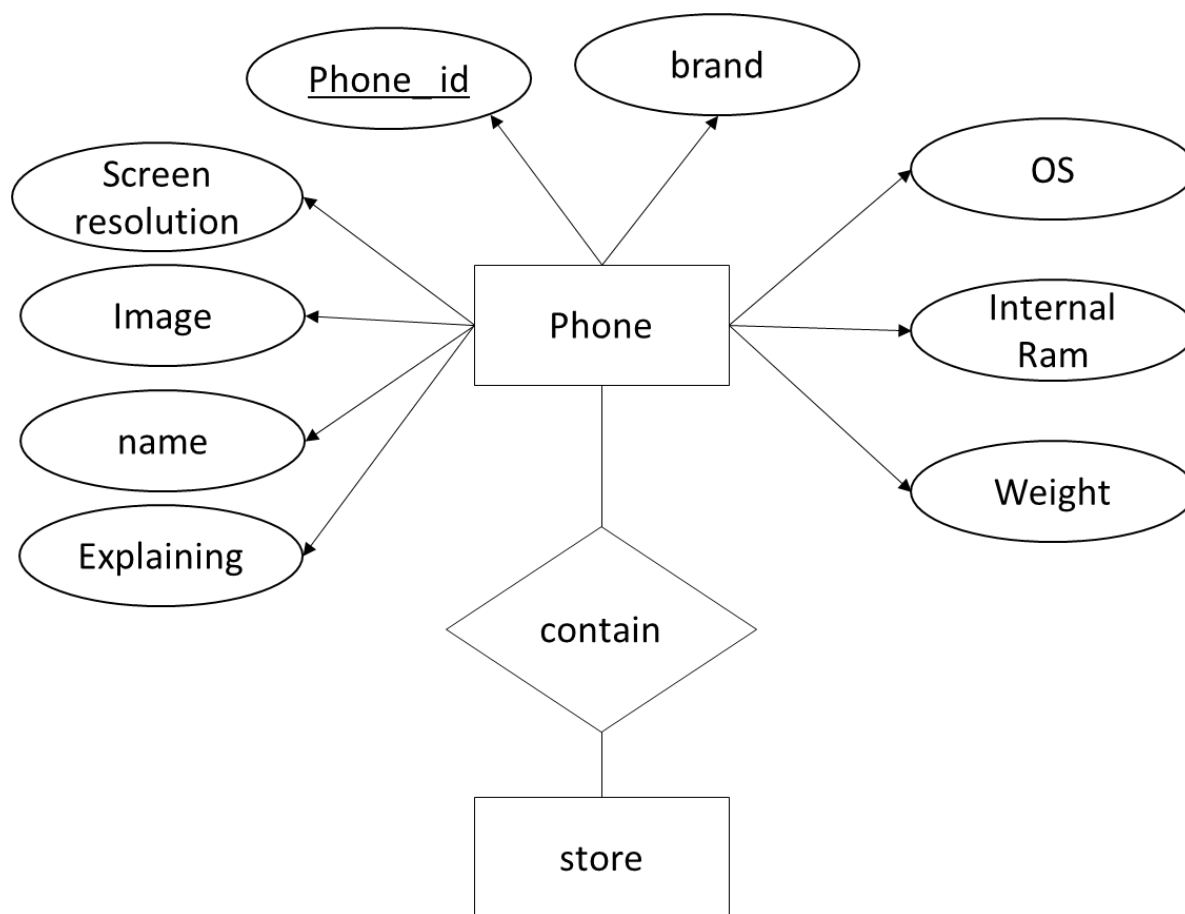
در شکل بالا، مدلی برای ذخیره مشخصات لپتاپ ارائه شده است. توضیح اینکه این مدل می تواند داینامیک باشد. بدین صورت که برای ذخیره مشخصات یک لپتاپ خاص، ما می توانستیم نام-مقدار را ذخیره کنیم. مثلاً

بگوییم رم=۵گیگ. یا بگوییم گرافیک=۲گیگ envidia و... یعنی مشخصات را بصورت نام-مقدار ذخیره کنیم. ولی چون هم وقت پروژه کم بود و هم در مصاحبه و جلساتی که با مشتری داشتیم، نیازمندی ما اینچنین نبود، لذا هر کالا مشخصات خاص خود را دارد. بنابراین مشخصات ثابتی را برای هر کالا در نظر گرفتیم و با توجه به آنها پروژه را پیش بُردیم.

همین اصل برای گوشی‌ها نیز صادق است که در ادامه به توضیح این موضوع نیز خواهیم پرداخت.

۳،۵ مدل سازی داده های مربوط به کالاها(گوشی ها)

مدل سازی را به شکل زیر انجام می دهیم.



شکل ۴ مدل سازی داده های مربوط به کالاها(گوشی ها)

۳،۵،۱

۳،۵،۲ چندی ارتباط:

ارتباط بین انبار و گوشی چند به چند است. اما چون انبار برای ما مهم نیست لذا هیچ مشخصاتی را که مربوط به انبار باشد در دیتابیس ذخیره نمی کنیم. چون ما نمی خواهیم یک سیستم انبارداری بزنیم. پس تنها مشخصات گوشی برای ما مهم است و اینکه هر گوشی در کدام انبار است برای ما مهم نیست. پس فروشگاه گوشی هایی دارد که هریک مشخصات خاص خود را دارند.

۳،۵،۳ توضیح:

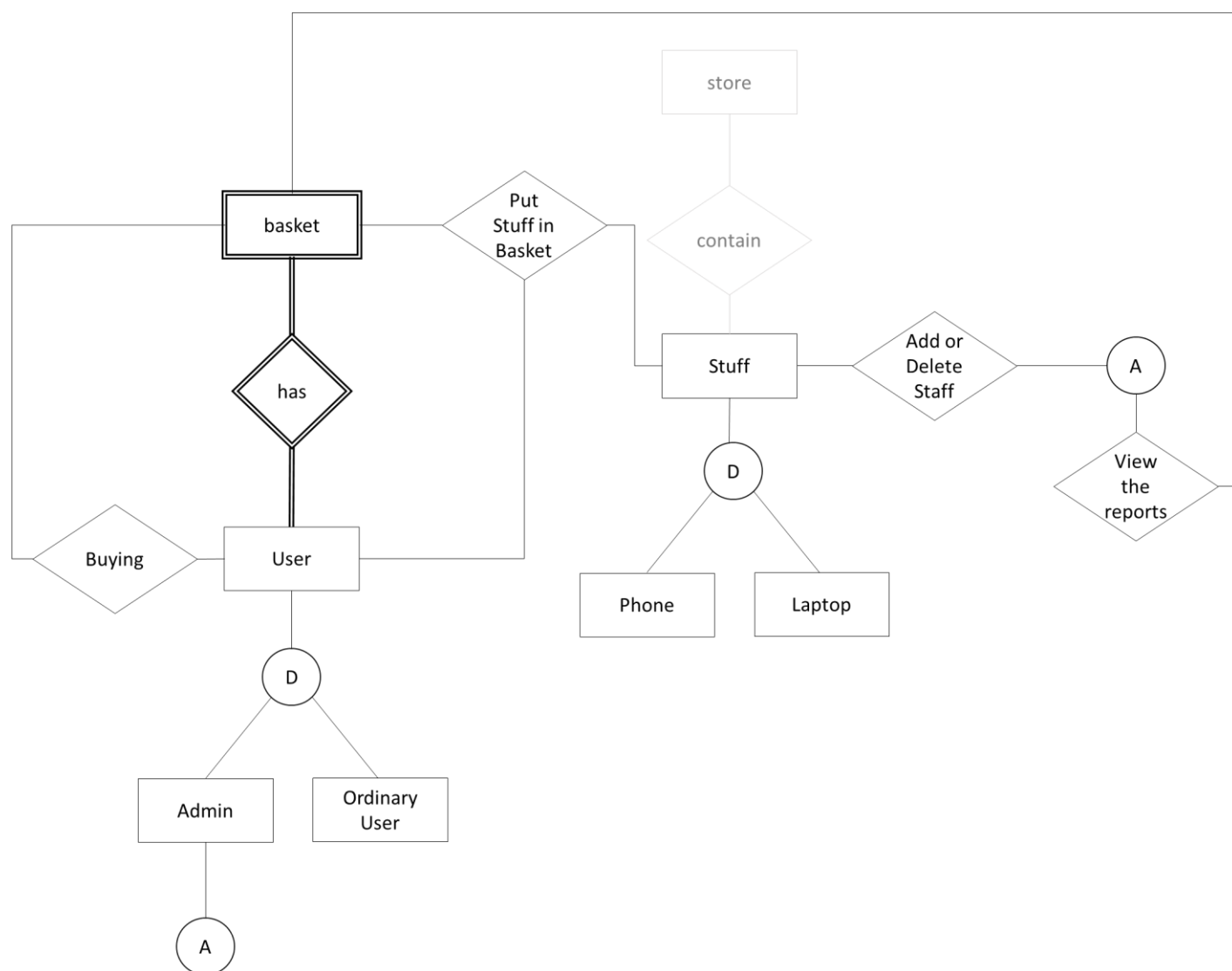
در شکل بالا، مدلی برای ذخیره مشخصات گوشی ارائه شده است. همامطور که در قبل نیز گفته شد، مدل می توانست داینامیک باشد. بدین صورت که برای ذخیره مشخصات یک گوشی خاص، ما می توانستیم نام-مقدار را

ذخیره کنیم. مثلاً بگوییم رم داخلی=۵ گیگ و... یعنی مشخصات را بصورت نام-مقدار ذخیره کنیم. ولی چون هم وقت پروژه کم بود و هم در مصاحبه و جلساتی که با مشتری داشتیم، نیازمندی ما اینچنین نبود، لذا هر کالا مشخصات خاص خود را دارد. بنابراین مشخصات ثابتی را برای هر کالا در نظر گرفتیم و با توجه به آن‌ها پروژه را پیش بُردیم.

در قبل نیز گفتیم که برای لپتاپ‌ها نیز مشخصات خاصی را مدنظر گرفتیم و فرض کردیم که مشخصات دیگری غیر از این مشخصات وجود ندارد.

۳,۶ ارتباط بین موجودیت‌ها

گفتیم که چون صفحات گنجایش اینکه بخواهیم کل داده‌ها را در یک صفحه مدل کنیم ندارند. لذا مدل‌سازی را تکه تکه کردیم و برای هر موجودیت بصورت جدا مدل کردیم. حال می‌خواهیم ارتباط بین مدل‌ها را رسم کنیم. مدلی که رسم خواهد شد، کامل‌تر است. اجازه دهید که تنها موجودیت‌ها و ارتباط بین موجودیت‌ها را مدل کنیم و از ذکر جزئیات به دلیل کمبود فضا صرف نظر کنیم. پس ما در اینجا صفات موجودیت‌ها را بطور کامل بیان نمی‌کنیم (زیرا جزئیات مربوط به صفات هر موجودیت، در صفحات قبلی مورد بررسی قرار گرفت) و تنها تمرکز ما روی ارتباط‌ها و صفت ارتباط است.



شکل ۵ نمایش ارتباط بین موجودیت‌ها

۳،۶،۱ نکات مهم مربوط به ER

چند نکته مهم در تصویر بالا وجود دارد که نیاز به توجه ویژه دارد.

- ۱- ادمین موجودیت جدا نیست. بلکه یک یوزر است. یک یوزر خاص که دسترسی‌های خاصی دارد. ما می‌خواستیم دسترسی‌هایی که یک ادمین دارد را به نمایش بگذاریم. لذا یوزرها را به دو قسمت ادمین و معمولی تقسیم کردیم و دسترسی‌های ادمین را نشان دادیم. نکته اینکه به دلیل کمبود جا مجبور به انتقال قسمتی از ER به سمت راست صفحه شدیم. لذا از حرف A برای انتقال ER استفاده کردیم و ادامه ER را در سمت راست و از ادامه حرف A کشیدیم.
- ۲- قسمتی از شکل که کمرنگ شده مربوط به کار ما نیست و نمی‌بایست در ER ذکر می‌شد. ما این مورد را ذکر کردیم تا تاکید کنیم که سیستم ما یک سیستم فروش است و اصلاً به مباحث مربوط به انبارداری و... مربوط نمی‌شود. لذا برای اینکه بگوییم انبارداری جزء ER ما نیست، این موارد را کمرنگ کردیم.
- ۳- ارتباط بین User و Basket و stuff این است که یک یوزر، کالای خاصی را در سبد خود قرار می‌دهد. در ارتباط buying که مربوط به خرید می‌شود، تنها User و Basket را دخالت دادیم. زیرا برای خرید کالا، یوزر ابتدا باید آن را در سبد خود قرار دهد. یعنی یوزر در صورتی می‌تواند کالایی را خرید کند که آن را در سبد خرید قرار دهد. بنابراین نیازی نیست که برای امر خریدن، Stuff را نیز دخالت دهیم. گرچه دخالت stuff در این ارتباط نیز منطقی به نظر می‌رسد، ولی طبق مفهوم و سناریوی خرید، بهتر است کالا را در این ارتباط دخالت ندهیم. زیرا محتویات سبد خرید، کالا است. بنابراین نیازی به دخالت کالا نیست.
- ۴- همانطور که قبلاً نیز ذکر کردیم، در اینجا به دلیل کمبود جا، صفات موجودیت‌ها را ذکر نکردیم. چرا که صفات هر موجودیت را قبلاً ذکر کرده بودیم.

۳،۶،۲ چندی ارتباط

• ارتباط put stuff in basket

- ارتباط بین یوزر و سبد، ۱ به ۱ است. چون هر یوزر تنها یک سبد دارد.
- ارتباط بین یوزر و کالا، ۱ به چند است. زیرا یک یوزر می‌تواند چندین کالا را در سبد خرید خود قرار دهد.

- ارتباط بین سبد و کالا نیز ۱ به چند است. زیرا در یک سبد می‌تواند چندین کالا قرار گیرد.

• ارتباط buying

○ این ارتباط ۱ به ۱ است و تنها دو موجودیت یوزر و سبد در این ارتباط دخالت دارند. زیرا هر یوزر تنها یک سبد دارد و می‌تواند محتویات سبد خود را خریداری کند. حال ممکن است در این سبد چندین کالا قرار گیرد. یعنی ۱ به ۱ بودن این ارتباط بدین معنا نیست که تنها یک کالا خریداری شود. بلکه بدین معناست که تنها محتویات یک سبد خریداری می‌شود.

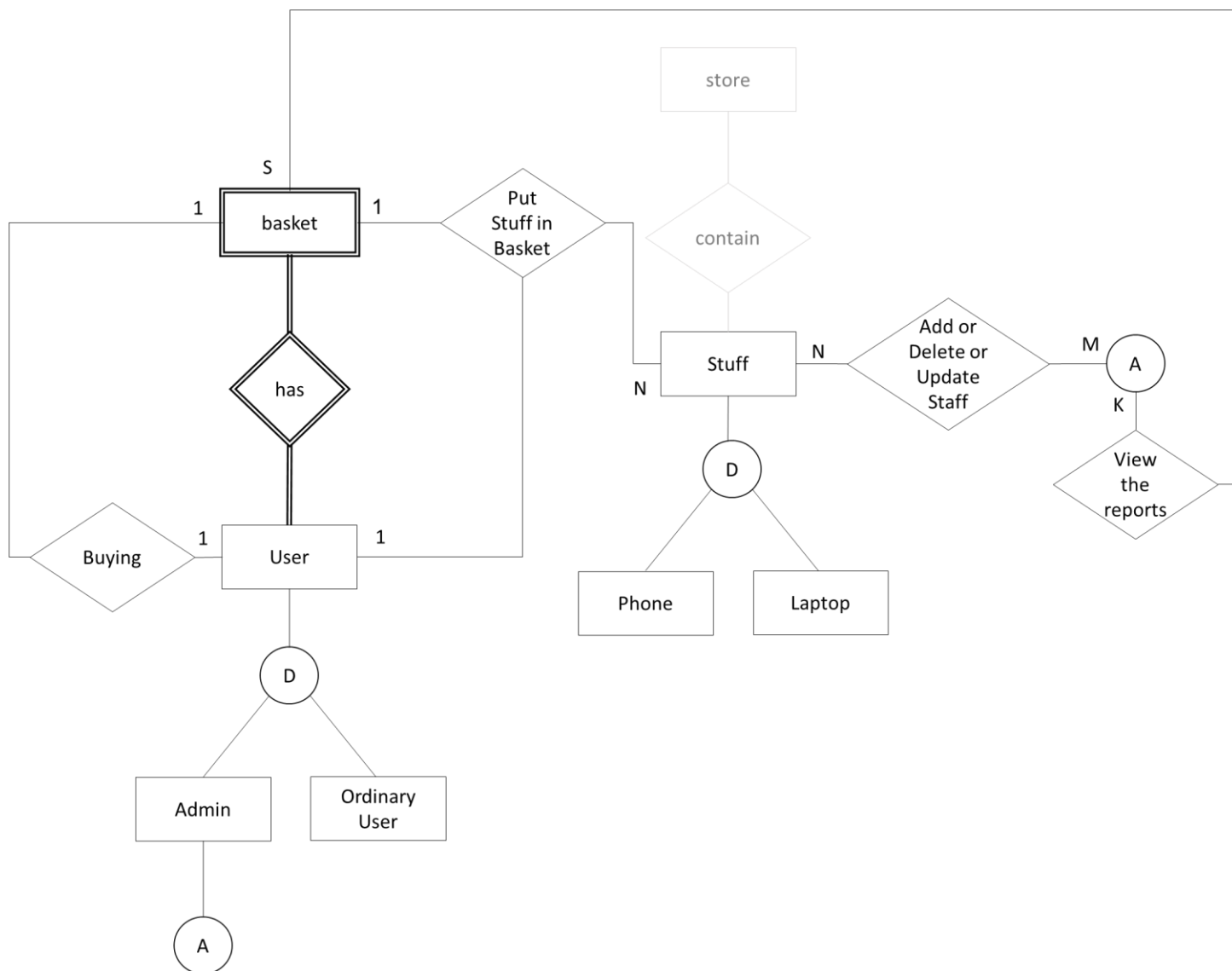
• ارتباط Add or Delete or Update Staff

○ یک کالا می‌تواند توسط چندین ادمین بروز رسانی و یا اضافه و یا حذف شود. یعنی نرم‌افزار می‌تواند چندین ادمین داشته‌باشد. بنابراین این ارتباط چند به چند است.

• ارتباط View the reports

○ چندین ادمین می‌توانند چندین گزارش را از سبد خرید مشتریان مختلف مشاهده کنند. بنابراین این ارتباط نیز چند به چند است.

با توجه به نکات ذکر شده، ER بصورت زیر نمایش داده می‌شود تا چندی ارتباطها بیشتر مشخص باشد.



شکل ۶ نمایش ارتباط بین موجودیت‌ها و چندی هر ارتباط

۴ نرمال سازی

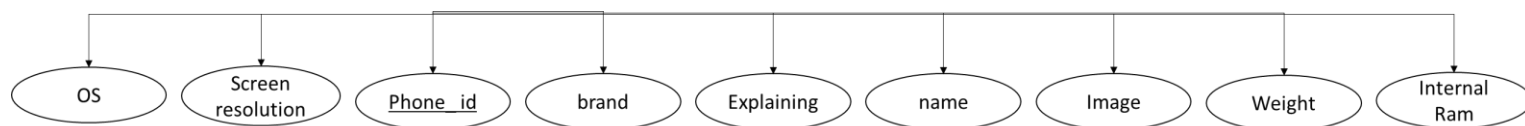
باید ER رسم شده در شکل بالا نرمال باشد. یعنی تست های 1NF، 2NF و 3NF و نهایتاً BCNF را پاس کرده و نرمال باشد. لازم به ذکر است که تست های نرمال سازی را اجرا کردیم و مشکلات را برطرف کردیم تا ER ما بصورت نرمال درآید. در ادامه به بررسی جزئی تری می پردازیم و تک تک تست ها را توضیح می دهیم.

۴,۱ بررسی 1NF بودن

برای اینکه یک جدول فرم نرمال 1NF را داشته باشد باید:

- ۱- Eliminate repeating groups in individual tables
- ۲- Create a separate table for each set of related data
- ۳- Identify each set of related data with a primary key

شکل زیر که مربوط به جدول مشخصات تلفن ها می باشد، نشان می دهد که ER رسم شده، 1NF است. زیرا هیچ گروه تکراری در جداول وجود ندارد. همچنین بغیر از PHONE_ID هیچ کلید دیگری نیز وجود ندارد. پس ER رسم شده 1NF است.



حال می توان همین جدول را برای دیگر جداول دیتابیس رسم کرد و مشاهده می شود که همه ی جداول 1NF هستند. نیازی به جدول جدید هم نیست.

۴,۲ بررسی 2NF بودن

برای اینکه یک جدول فرم نرمال 2NF را داشته باشد باید:

- ۱- Table is in 1NF (First normal form)
- ۲- No non-prime attribute is dependent on the proper subset of any candidate key of table

تمامی جداول ما دو شرط بالا را هم اکنون دارد. پس 2NF نیز هست. یعنی به گونه ای نیست که بخواهیم برای یک کلید (یک سطر) چندین سطر بخاطر رابطه ی یک به چند داشته باشیم.

۴,۳ بررسی 3NF بودن

برای اینکه یک جدول فرم نرمال 3NF را داشته باشد باید:

1- Table must be in 2NF

2- Transitive functional dependency of non-prime attribute on any super key should be removed

تمامی جداول ما دو شرط بالا را هم اکنون دارد. پس 3NF نیز هست.

۴,۴ بررسی BCNF بودن

برای اینکه یک جدول فرم نرمال BCNF را داشته باشد باید:

1- Table must be in 3NF

2- A table complies with BCNF if it is in 3NF and for every functional dependency $X \rightarrow Y$, X should be the super key of the table

تمامی جداول ما دو شرط بالا را هم اکنون دارد. پس BCNF نیز هست.

خیلی وقت‌ها جداولی که طراحی می‌کنیم از همان اول، بصورت نرمال هستند و نیازی به نرمال‌سازی آنچنانی ندارند. گرچه این موارد را در کلیه جداول بررسی کردیم و نرمال بودند و هیچگونه تغییر خاصی اعمال نشد.