



بسته آموزشی :

فن آوری اطلاعات سلامت

گروه هدف :

گروه مدارک پزشکی

مدل داده ها (Data Model)

مقدمه

وقتی کسی در باره کره زمین صحبت می کند، همگان آن را به صورت یک کره تصور می کنیم. کره ساده ترین و نزدیک ترین شی ریاضی است که زمین به آن شباهت دارد. یا وقتی عدد ۳ را مشاهده می کنیم، همگان می دانیم که این نمادی است که گوینده برای بیان مفهوم ذهنی خود آن را به کار برده است. یا وقتی که کلمه کتاب را می بینیم یا می شنویم، اگر چه ممکن است دقیقا همان سابقه ذهنی گوینده یا نویسنده را در باره کتاب مورد نظر وی نداشته باشیم، اما درک می کنیم که وی از چه نوع شیئی صحبت می کند و مسلما تصور نخواهیم کرد که منظور وی درخت، دیوار و ... می باشد.

وقتی که تصمیم به ساختن یک بنا گرفته می شود، نقشه ای ترسیم می شود که کمابیش حاوی اجزای تشکیل دهنده و روابط بین اجزای آن ساختمان می باشد و همین طور برای ساختن هواپیما، اتومبیل و ... چنین طرح هایی ضروری است. ما دردنیایی از

سیستم‌ها زندگی می‌کنیم، بسیاری از سیستم‌ها مانند سیستم اتومبیل قابل مشاهده و لمس است، ما به راحتی اجزای آن را می‌بینیم و با تعمق بیشتر روابط بین اجزا را نیز می‌توانیم کشف کنیم و اگر کسی بتواند اجزا و روابط آنها را به صورت قابل فهم بر روی کاغذ به تصویر بکشد، ما به راحتی خواهیم توانست این سیستم را بشناسیم. اما سیستم‌هایی نیز وجود دارد که تأثیرات عمده‌ای در زندگی انسان‌ها دارند و در اغلب موارد ایجاد آنها همانند ماشین یا هواپیما نبوده و بدون طرح‌های کاملاً ملموس ساخته می‌شوند. به هر حال این سیستم‌ها اعم از سیستم بهداشت، سیستم آموزشی، بازرگانی و غیره در دنیای واقعی وجود دارند و نماد بیرونی آنها در حد نمودارهای سازمانی می‌باشد.

برای بهبود زندگی انسان‌ها، لازم است سیستم‌های مناسبی داشته باشیم. برای ایجاد سیستم‌های جدید و یا برای بهینه‌سازی سیستم‌های موجود بایستی بتوانیم آن سیستم‌ها را به گونه‌ای نشان دهیم که همگان درک مشابهی از اجزا و روابط آنها داشته باشند. در واقع باید بتوانیم مفاهیم ذهنی را، به مفاهیم عینی و قابل مشاهده تبدیل کنیم، مفاهیم ذهنی که معمولاً برای دو فرد انسانی کاملاً مشابه و یکسان نمی‌باشد. در سیستم‌هایی که ترکیبی از عوامل انسانی و فنی و فرایندهای گوناگون می‌باشند، عینی کردن و قابل درک نمودن اجزای سیستم، فرایندها و روابط بین اجزا و به عبارت بهتر مدل‌سازی اهمیت به مراتب بیشتری پیدا می‌کند. در این فصل خوانندگان را با مدل‌سازی داده‌ها آشنا خواهیم کرد و این که به عنوان اولین گام در ایجاد سیستم‌های اطلاعاتی چگونه داده‌ها را به تصویر بکشیم.

تعریف مدل‌سازی داده‌ها

مدل‌سازی داده‌ها، روشی است برای تعیین این که چه داده‌ها و ارتباط‌هایی بایستی در یک پایگاه داده ذخیره شود و در واقع روشی برای مستند نمودن طرح پایگاه داده می‌باشد. در تعریفی دیگر مدل‌سازی داده‌ها روشی است که فهرست، شکل، اندازه، محتوی و قوانین مربوط به عناصر داده‌های مورد استفاده در حیطه یک فرایند کسب و کار را ثبت می‌کند. این حیطه کاری ممکن است به وسعت یک شرکت چند منظوره بین‌المللی یا به محدودیت تحویل جعبه‌ها بر روی عرشه یک کشتی باشد. محصول نهایی چیزی شبیه به یک نقشه است، همراه با همه مستندات پشتیبان که برای تفسیر و واضح‌سازی آن مورد نیاز می‌باشد. مدل‌سازی به شما کمک می‌کند قبل از این که یک پایگاه داده ساخته شود، ساختار و مفهوم داده‌هایی را که ضروری هستند، درک کنید.

تفاوت مدل‌سازی داده و مدل‌سازی فرایند

یک مدل‌سازی فرایند، الگویی از فعالیت‌ها، وظایف و فرایندهای یک سازمان است. فرایندها در یک مدل‌سازی فرایند اغلب در قالب دروندادها و برون‌دادهایشان تعریف می‌شوند. اما یک مدل‌سازی داده‌ها هیچ اقدام یا جریان اطلاعات را منعکس نمی‌نماید، بلکه فقط یک دیدگاه ایستا از داده‌ها در یک مدل‌سازی داده‌ها نشان داده می‌شود. برای مدل‌سازی داده‌ها لازم است که پرسنل یک حیطه کاری، با یک معمار داده‌ها همفکری و تلاش نمایند و برای تعریف اشیای لازم در نقشه، صفات خاصه آنها و ارتباط بین اشیای به توافق برسند.

چه کسانی از مدل‌سازی داده‌ها استفاده می‌کنند؟

همه کسانی که لازم است از داده‌ها استفاده نمایند یا بر داده‌ها مدیریت کنند، بایستی از مدل‌سازی داده‌ها استفاده نمایند. افراد خبره در زمینه موضوع، کاربران نهایی که داده‌ها را تحلیل نموده و گزارشاتی را تهیه می‌نمایند و تهیه‌کنندگان نرم‌افزارها و پایگاه‌های داده در زمره این افراد هستند.

چرا به مدل داده ها نیاز دارید؟

- ۱- یک مدل با تضمین این که الزامات محصول نهایی به طور رضایت بخش تامین خواهد شد، ریسک کلی راکاهش می دهد. با آزمودن یک مدل آزمایشی از محصول نهایی، کاربران مورد نظر می توانند به طور منطقی تعیین کنند که آیا محصول حقیقتاً نیازها و اهداف آنها را تامین می نماید یا نه.
- ۲- یک مدل به پدیدآورندگان محصول کمک می کند تا مجسم کنند که محصول نهایی چگونه با سیستم ها و کارکردها مواجه خواهد شد. اگر یک مدل به صورت مشروح تهیه شود، میزان تلاش مورد نیاز برای تطابق و قابلیت انجام کار می تواند به طور منطقی تخمین زده شود.
- ۳- یک مدل به افراد ذیربط کمک می کند تا درک کنند که چگونه به محصول نهایی پل بزنند و محصول مورد نظر چگونه با وظیفه کاری آنها مرتبط می باشد. مدل به پدیدآورندگان محصول نیز کمک می کند تا مهارت های مورد نیاز مخاطبین را درک کنند و این که کدام نیازهای آموزشی برای استفاده مناسب از محصول وجود دارد.
- ۴- بالاخره یک مدل تضمین می کند که افراد سازنده محصول و آنهایی که محصول مورد نظر را درخواست می کنند، انتظارات یکسانی از نتیجه نهایی کار داشته باشند. با آزمودن مدل، امکان فرصت سوزی به طور قابل ملاحظه ای کاهش می یابد و باور و اعتماد افراد ذینفع نسبت به رضایت بخش بودن محصول نهایی افزایش می یابد.

هدف از مدل سازی داده ها

هدف از مدل سازی داده ها، ایجاد یک مدل دقیق یا نمایش تصویری از نیازهای اطلاعاتی مشتریان و فرایندهای کاری می باشد. یک مدل داده خوب یک نقشه معماری است که ایجاد یک پایگاه داده با ثبات و نرم افزار کاربردی را امکان پذیر می سازد. مدل داده ها به عنوان یک چهارچوب کاری برای ایجاد نرم افزارهای جدید یا ارتقای نرم افزارهای پیشین عمل می کند. هدف از مدل سازی داده ها این نیست که داده ها چگونه ذخیره خواهند شد. این به کسانی مربوط می شود که پایگاه داده را اجرا می کنند. یک مدل داده با نحوه پردازش داده ها نیز سر و کار ندارد. این مورد مز به مدل سازی فرایند مربوط می باشد. بلکه هدف، یک نمایش دقیق از نیازهای داده ای و ارتباطات داده ای در دنیای واقعی می باشد.

مدل داده ها و سازمان ها

مدل سازی به طور گسترده ای در سازمان ها به کار می رود تا در باره مشکلات سازمانی و طراحی راه حل ها درس هایی آموخته شود. مدل سازی در سطوح مختلف انجام می گیرد. در بالاترین سطح، یک سازمان بایستی با تعریف عناصر عمده در محیط خود از قبیل اهداف واحدهای سازمانی، منطقه ای که در آن فعالیت می کند، وقایع مهم، موجودیت ها و فرایندها، حیطة کاری خود را مشخص کند. در بالاترین سطح، مدل های متنی به فراوانی مورد استفاده قرار می گیرند. برای مثال یک سازمان ممکن است اهداف عمده خود و فرایندهای کاری را فهرست نماید. یک نقشه معمولاً برای نشان دادن محل فعالیت سازمان به کار می رود. وقتی که مدیر ارشد حیطة کاری را مشخص نمود، مدل هایی برای هر کدام از عناصر عمده می تواند ساخته شود. اهداف به برنامه های سازمانی تبدیل خواهند شد، واحدهای سازمانی در قالب نمودار سازمانی قرار می گیرند و الی آخر. مدل های کلیدی در سازمان ها عبارتند از مدل داده ها و مدل فرایندها. مدیریت داده ها با مدل داده ها، تجزیه و تحلیل سیستم ها با مدل فرایند سر و کار دارند. فناوری یا آخرین مرحله، مدل ها را به سیستم های عملیاتی تبدیل می کند تا از فعالیت های روزمره سازمان پشتیبانی نماید.

مدل های داده معمولاً به پایگاه های داده ای رابطه ای تبدیل می شوند و مدل های فرایندی به تهیه نرم افزارهای کامپیوتری تبدیل می شوند. بنا بر این شما می توانید مشاهده کنید که مدل سازی داده ها جزئی از یک فعالیت جامع مدل سازی است که

اغلب برای طراحی سیستم های کاری مورد نیاز است. وقتی که یک کسب و کار دچار تحولی بزرگ می شود، مانند اجرای پروژه مهندسی مجدد، بسیاری از ابعاد می توانند تغییر یابند و این فرصت خوبی برای دوباره اندیشی در باره عناصر سازمانی می باشد. از آنجایی که چنین تغییر عمده ای بسیار مخرب و هزینه بردار است، کمتر اتفاق می افتد. احتمال بسیاری وجود دارد که مدل سازی داده ها به عنوان یک فعالیت مستقل یا به عنوان بخشی از مدل سازی فرایند برای ایجاد یک نرم افزار کاربردی جدید برای سازمان انجام شود.

یک مدل به ما کمک می کند تا ببینیم که اکنون داده ها را چگونه مدیریت می کنیم و در گذشته چگونه مدیریت نموده ایم. از دیدگاه پایگاه داده، مدل داده ها یک طرح تفصیلی برای یک پایگاه داده است نه خود پایگاه داده. یک سنگ بنا برای ایجاد نرم افزار است اما یک سنگ بنای مهم از آنچه که وجود دارد. مدل سازی داده ها همچنین برای ایجاد طرح ها برای چیزهای جدید به کار می رود (آنچه که باید باشد). در بیشتر موارد ما مدل سازی می کنیم تا گزینه ها را بیازماییم. تعدادی از راه حل ها می تواند حتی برای مسائل مشابه وجود داشته باشد و همیشه شناختن راه حل درست کار آسانی نیست.

انواع مدل داده ها در یک سازمان

مدل موضوعی (Contextual)

یک مدل داده موضوعی، حیطة مورد نظر را مشخص می کند و موضوعات عمده و روابط بین آنها را نشان می دهد. این سطح، پیچیدگی ها را پنهان می کند و در وهله اول برای جستجوی موضوعاتی مورد استفاده قرار گیرند که بیشترین سهم را دارند و بنا بر این لازم است به طور مداوم شناسایی شده و تعریف شوند و حتی به روش های متفاوت مورد استفاده قرار گیرند.

مدل مفهومی داده ها

بعضی از مدل ها برای مستندسازی ایده های بسیار سطح بالا ساخته می شوند که به نام مدل های مفهومی معروف هستند. این مدل بر تحلیلی از نیازهای مشتری مبتنی است و موارد مهم و مورد علاقه یک سازمان یا کسب و کار را توصیف می کند. مدل مفهومی داده ها یکی از قویترین و موثرترین تکنیک های تحلیلی است که تاکنون برای درک و سازماندهی اطلاعات مورد نیاز برای پشتیبانی از هر سازمان ایجاد شده است. این مدل بر تصویر بزرگ و اهداف استراتژیکی تاکید می کند که سلامت و موفقیت سازمان را تضمین می نمایند. موضوعات داده ای بین مرزهای مالی، فرایندی و سازمانی به اشتراک گذاشته می شوند. در نتیجه این مدل و نرم افزارها و پایگاه های داده ای که بر اساس آنها ساخته می شوند، با افزایش کاربرد و جلوگیری از افزونگی داده ها، نقطه محوری برای حذف اتلاف وقت و هزینه در هدایت فرایندها می باشند.

مدل مفهومی، مدل موضوعی را پالایش می کند و ارتباط با سایر موجودیت ها را که برای درک کامل هر موضوع مورد نیاز است اضافه می کند. هدف از این سطح روشن ساختن ارتباط بین موجودیت های عمده و افزودن جزئیات مورد نیاز است تا تمایز هر موجودیت از موجودیت دیگر را امکان پذیر سازد. مدل های داده اغلب سازمان ها در این سطح آغاز می شود، زیرا شرایط و جو حاکم بر سازمان قبلاً مشخص شده است.

یک مدل مفهومی می تواند بگوید که چگونه یک سیستم کامپیوتری راه اندازی می شود. مدل مفهومی نیازهای داده ای سازمان را به صورت مستقل از نیازهای وظیفه ای سازمان (مانند حجم تراکنش ها، زمان پاسخ دهی سیستم و امنیت داده ها) بیان می کند، و مستقل از خصوصیات نرم افزار و پایگاه داده انتخاب شده می باشد. اما زمانی که سازمان از یک استراتژی منظم ایجاد نرم افزار تبعیت کند، با استفاده از مدل مفهومی داده ها به عنوان یک نقطه شروع، مدل منطقی و سپس مدل های فیزیکی داده ها ایجاد خواهند شد و طراحان پایگاه داده و نرم افزار برای طراحی و اجرای نرم افزارهای سیستم های اطلاعاتی جدید یا ارتقای سیستم های قبلی معمولاً از مدل مفهومی استفاده می کنند.

مدل منطقی داده ها

مدل منطقی با پالایش مدل مفهومی تهیه می شود و نمایانگر روابط منطقی بین داده ها و اطلاعاتی است که در یک سیستم نگهداری می شود. یک مدل منطقی عناصر داده ای را شکل می دهد، آنها را تعریف نموده و استاندارد می کند تا بتوانند برای تمام اهداف سازمانی به اشتراک گذاشته شوند، نه این که فقط برای یک واحد کاری یا یک نقطه نظر خاص در نظر گرفته شوند. یک مدل منطقی در حقیقت طرحی از داده ها در کل یک سازمان بدون در نظر گرفتن نوع کامپیوتر، سیستم عامل، ساختار فایل یا نوع فناوری می باشد. مدل منطقی منظره کل داده های شما را نشان می دهد.

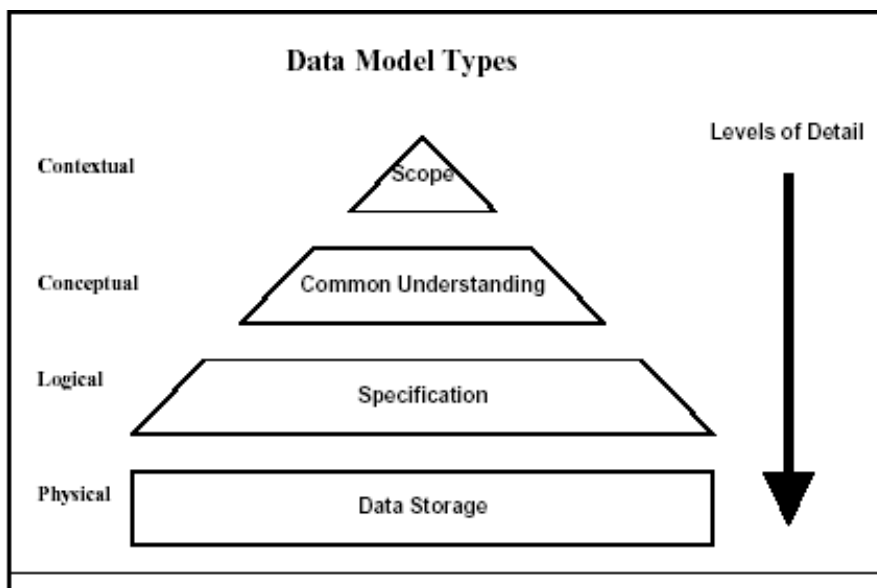
مدل منطقی داده ها با ساختار داده ها سر و کار دارد تا با اصول ثابت مدیریت پایگاه داده مطابقت نماید. این مدل نیازهای داده ای واقعی را که بر روی فرم ها و گزارشات یک سیستم اطلاعاتی وجود دارند، نشان می دهد. مدل منطقی برای یک سازمان، مبتنی بر مدل مفهومی داده ها است اما دارای جزئیاتی بوده و بیان قویتری از ساختارهای داده ای می باشد. مدل منطقی داده ها مشروح ترین سطح از مدل داده هاست. در این مدل، همه پیچیدگی های موجودیت های داده ها و روابط آنها بیان می شود و همه موجودیت ها بایستی به صورت کامل با تمامی خصوصیات خود توصیف شوند. مدل منطقی امکان تحلیل و طراحی بهینه ساختار پایگاه داده را فراهم می کند.

مدل فیزیکی داده ها

مشهورترین مدل داده ها، مدل فیزیکی است که طرح واقعی یک پایگاه داده را مشخص می کند. این مدل بر مبنای مدل منطقی ساخته می شود و مبنای برنامه ای می شود که برای ساختن جداول، دیدها و قیود یکپارچگی به کار می رود (در فصل پنجم در این باره توضیح داده خواهد شد). مدل فیزیکی داده ها محدودیت های فیزیکی را که به وسیله کسب و کار اجباری می شود، از بین می برد (عملکرد تراکنش های خاص، محدودیت های مربوط به حجم داده ها، توزیع فیزیکی داده ها در محل های مختلف و حفاظت امنیتی داده های انتخاب شده) و یک طرح فیزیکی متعادل از پایگاه داده ایجاد می کند که آن محدودیت ها (بعضی اوقات، تعارض ها) را برطرف نماید، بدون این که سازماندهی طبیعی داده ها را قربانی نماید. هر مدل فیزیکی بر مدل منطقی داده ها و با در نظر گرفتن الزامات عملکرد و خصوصیات اجتماعی نرم افزار اجرایی انتخاب شده. ساخته می شود. مدل فیزیکی داده ها حاوی همان جزئیات مدل منطقی است اما تغییراتی یافته است تا نشان دهد چگونه داده ها در یک سیستم اطلاعاتی ذخیره می شوند. ساختارهای تبادل اطلاعات و فرمت ها (پیغام ها) نیز در مدل فیزیکی تعریف می شوند.

چگونه سه مدل را با هم متناسب نماییم؟

این سه نوع مدل به راحتی با یکدیگر مرتبط می شوند و به عنوان دیدگاه های متفاوت از یک فرایند منفرد کسب و کار می توانند پشتیبان یکدیگر باشند. وقتی که نیازها جمع آوری گردید، یک مدل مفهومی طراحی می گردد. این مدل یک دیدگاه کلی در مورد موضوع را فراهم می نماید (موجودیت ها و ارتباط های آنها). گام بعدی طراحی یک مدل منطقی است. مدل منطقی بر داده های مورد نیاز سیستم متمرکز است، مستقل از این چگونگی نرم افزار پایگاه داده و یا چگونگی پردازش داده ها و یا این که چگونه داده ها ذخیره می شوند. وقتی که مدل منطقی کامل گردید، یک یا چند مدل فیزیکی مشروح می تواند تهیه شود. مدل فیزیکی همان مدل منطقی است که مطابق با محدودیت های نرم افزار سیستم مدیریت پایگاه داده انتخاب شده، اجرا می شود. انواع مختلف مدل های داده اهداف متفاوتی دارند و هر چه به سمت پایین می رویم، جزئیات بیشتری را نشان می دهند. شکل زیر ارتباط بین مدل های logical, conceptual, contextual و physical را نشان می دهد.



چگونه مدل، تبادل داده ها را تسهیل می نماید؟

مدل مفهومی وقتی که در بین نرم افزارها و کل سازمان به اشتراک گذاشته می شود، داده هایی را که مشترک هستند و آنهایی را که به اشتراک گذاشته خواهند شد بیان می کند. در نتیجه می تواند استاندارد برای تعریف و ارائه داده ها فراهم می نماید. نرم افزارها و پایگاه های داده ای که بر اساس مدل مفهومی ساخته می شوند، لازم است که داده ها را در یک شکل سازگار یا نقشه ای بر اساس آن مدل ذخیره نمایند تا تبادل داده ها به راحتی بتواند انجام گیرد.

چه کسانی سازنده مدل داده هستند؟

یافتن این که چه کسانی در سازمان شما مهارت ساختن مدل داده ها را دارند، می تواند یک چالش باشد. شرکت های مختلف از عناوین مختلف برای این پست استفاده می کنند و روز به روز این مهارت را به توصیف شغلی عناوین شغلی مختلف اضافه می کنند. عده ای معتقدند که مدل سازی داده ها اغلب یک مجموعه مهارت است تا یک شغل تمام وقت و برخی معتقدند مدل سازی داده ها یک تخصص است که به نظر می رسد همانند مدیریت پایگاه داده ها یا برنامه نویسی عنوان مشخصی داشته باشد. اکنون با چند مثال به مقایسه نحوه کار مدل ساز داده ها با برخی از حرفه ها می پردازیم:

تصویربردار

به فرد یا تیم مدل ساز به عنوان یک عکاس مجله نشنال جیوگرافی^۱ فکر کنید. متناسب با وظیفه محوله، آنها یا عکاس فضای ابری یا نوترون های درون فضا هستند. آنها تصاویری می گیرند که واقعیات را با هر سطح از جزئیات مستند می کند و به بیننده این قدرت را می دهد که پاسخ ها را پیدا کند. تصاویر زیبا، تمام رنگی و ساده امکان باز شناخت واقعیات را به راحتی فراهم می نمایند. بدون تصاویری از زندگی کنونی، درک ما از زندگی و طبیعت چه می تواند باشد؟ شغل یک عکاس مشابه کسی است که ساختارهای داده های را که از قبل وجود دارند، مستند می کند. (بنا بر این ساختارهای "همان طور که هست" نامیده می شوند). هنر عکاسی به سادگی واقعیات زندگی را به صورت کامل منعکس نمی کند، اما یک

چیزی اضافه می کند که به ما در درک موضوع کمک می کند. عکاسان مهارت خود را به کار می گیرند تا جنبه هایی که موضوع را از دیدگاه خاصی می نگرد، برجسته شود و توجه ما را بر چیزی که آنها مهم تلقی کرده اند، متمرکز می کنند. تیم مدل سازی تلاش می کند تا وضعیت موجود داده های سازمان را مستند نماید. داده هایی که به طور مداوم اطلاعات مربوط به عناصر داده ای را به هنگام می کنند، آنها چه هستند، در کجا مورد استفاده قرار می گیرند، آنها چگونه ساخته شده اند، معنی آنها چیست؟ و همه جزئیاتی که به ماهیت فیزیکی آنها اشاره می کنند (مانند نوع داده، اندازه، دقت اندازه گیری، مقادیر پیش فرض و غیره). سطح جزئیات متناسب با مخاطب و هدف مدل فرق می کند.

آرشیستک

حالا به فرد یا تیم مدل سازی به عنوان یک آرشیستک بیندیشید. آنها تلاش می کنند تا نیازهای مالک، استفاده مورد نظر، بودجه، منابع و طرح های آینده یک ساختمان را درک کنند. یک آرشیستک می تواند جزئیات یک موزه و موسسه پژوهشی را ایجاد نماید. طرحی که می تواند از طرح یک نانویی یا کارخانه اتومبیل سازی متفاوت باشد. زیرا هر طرح بایستی پارامترهای متعددی را در نظر بگیرد. سپس وی به ساختمان از دیدگاه یک مشاور و ارزیاب ساختمان نگاه می کنند. وی به عنوان عضوی از تیم باقی می ماند و در صورت بروز مشکلات تغییرات لازم را در نقشه ایجاد می کند.

آرشیستک ساختار داده ها، فرد یا تیمی است که یک مدل داده جدید را طراحی می کند. این ها طرح های "آن طور که باشد" یا وضعیت آینده می باشند. مدل ها به دفعات به مشتری ارائه می شود و بازنگری می شود تا این که گزینه نهایی مناسب به دست آید.

تیم مدل سازی که برای ساختن آینده کار می کند، سعی می کند بهترین راه حل را برای مشکل کسب و کار طراحی نماید. راه حل ، ممکن است به تاکید بر انعطاف پذیری، امنیت، بهبود فرایند، کیفیت داده ها یا همه آنها نیاز داشته باشد. این تیم مدل وضعیت موجود عناصر داده های را برای کشف نارسایی ها و داده های ضروری و جدید مورد استفاده قرار می دهند. تیم ممکن است یک راه حل کاملا جدید ایجاد کند یا یک طرح بازنگری شده را تهیه کند.

مهندس بهره وری

مهندسین بهره وری (در صنعت اتومبیل و غیره) بارها ماشین ها، ابزارها یا سیستم ها را مورد بررسی قرار می دهند تا کیفیت آنها را بهتر سازند. آنها روش هایی برای افزایش کارایی و کاهش مصرف انرژی پیدا می کنند. آنها می خواهند امنیت و قابلیت اطمینان و رضایت بخشی استانداردهای جدید را افزایش دهند و نیز می خواهند نقاط شکست و نارسایی را به پایین ترین سطح ممکن برسانند، در حالی که از بخش های مهم بیشترین پشتیبانی را به عمل می آورند. نمونه های اولیه آنها معمولا طرح های پیشنهادی و راه حل هایی هستند که به یک کمیته جهت بررسی و تصویب ارسال می نمایند.

این مثال، تاکید سازمانی بر یک روش بهبود مدل سازی داده ها را شرح می دهد. داده ها جزو دارایی های سازمان محسوب می شوند. اگر از هر بخش امور مالی سوال کنید، آیا آنها ترجیح نمی دهند که همه دارایی های سازمان بایستی با هزینه کمتر و حتی الامکان تا مدت زمان طولانی نگهداری شوند. تیم مدل ساز داده مانند یک مهندس بهینه سازی عمل می کند و فقط با مدل های فیزیکی داده ها کار خواهد کرد. آنها احتمالا با یک موضوع "وضعیت موجود" شروع خواهند کرد و با یک یا چندین راه حل های پیشنهادی "باید باشد" کار را ادامه خواهند داد.

کتابداران در باره آنچه که نیاز دارید، از شما انواع سوالات را می پرسند. آنها به انواع فهرست ها و قفسه ها سر می کشند و شما را راهنمایی می کنند تا مطالب مورد نظر خور را پیدا کنید. اگر شما یک نوع خاص از داده را جستجو می کنید، تیم مدل ساز داده افرادی هستند که می توانند مناسب باشند. آنها می توانند کتابداران مرجع برای داده های سازمان باشند، مخصوصا اگر هیچ تیم مدیریتی برای شناسایی داده ها وجود نداشته باشد. حداقل این است که آنها همه طرح ها را می شناسند.

مدل داده ها چگونه ترسیم می شود؟

ساختن یک مدل داده نتیجه یک تشریح مساعی بین مشتری، نماینده ای از کاربران نهایی پایگاه داده و یک طراح می باشد. البته این تیم می تواند فقط از مشتریان و طراحان تشکیل شود. ترسیم یک مدل داده یک فرایند تکراری از آزمون و بازنگری است. یک مدل داده، یک سند کاری است که هر چه شما بیشتر در باره نیازهای مشتری و دنیای مشتری فرا می گیرید، تغییر خواهد یافت. نسخه های اولیه شما احتمالا از نسخه نهایی کاملا متفاوت خواهد بود. دیگرام های خود را با یک مداد ترسیم کنید و یک مداد پاک کن بزرگ نیز در دست داشته باشید. برای بهتر شدن کار می توانید از نرم افزارهای مدل داده برای ترسیم و بازنگری استفاده کنید.

دو نوع مدل عمده برای داده ها عبارتند از: مدل ارتباط - موجودیت^۲ و مدل اشیای معنایی^۳. مدل ارتباط - موجودیت در این فصل مورد بررسی قرار می گیرد.

مدل ارتباط - موجودیت

این مدل توسط پیترچن^۴ در سال ۱۹۷۶ معرفی شد. چن در مقاله خود اصول مدل سازی را بیان نمود. این اصول بعدا توسط خود وی و سایرین توسعه و تغییر یافت. امروزه یک مدل E-R استاندارد و یکتا وجود ندارد. توجه داشته باشید که ممکن است در محیط کارتان با علائم و نشانه های مختلفی از E-R رو به رو شوید.

واحدهای ساختمانی مدل داده ها

هدف از یک پایگاه داده ذخیره داده ها در باره چیزهاست. این چیزها می توانند شامل حقایق (مثلا هزینه یک نوع عمل جراحی)، برنامه ها (مثلا اعمال جراحی برنامه ریزی شده برای سه ماهه اول سال)، برآوردها (مثلا پیش بینی تعداد مراجعات بیمه شدگان تامین اجتماعی به درمانگاه) و یا سایر داده های مختلف باشد. یک مدل داده این چیزها و ارتباط آنها را با چیزهای دیگر با استفاده از چهار عنصر موجودیت، صفت، ارتباط و شناسه توصیف می کند.

موجودیت (Entity)

موجودیت، واحد اساسی یک مدل داده می باشد. یک موجودیت چیزی است که لازم است آن را توصیف نماییم و داده هایی در باره آن بایستی نگهداری شود، یا چیزی است که برای کاربران مهم می باشد و می خواهند داده هایی در باره آن جمع آوری کنند. بعضی از موجودیت ها فیزیکی هستند مانند بیمار، پزشک و درمانگاه. برخی دیگر مفهومی و فرایندی هستند مانند ویزیت و تهیه صورت حساب.

PATIENT

یک موجودیت با یک مستطیل نمایش داده می شود و نام موجودیت با حروف بزرگ نوشته می شود. هر موجودیت در یک مدل داده ها نام منحصر بفردی دارد که ما به شکل یک کلمه مفرد می نویسیم. چرا کلمه

1. Entity-Relationship
2. Semantic Object
3. Peter chen

مفرد؟ زیرا ما می خواهیم تاکید کنیم که یک موجودیت مثالی از یک چیز را نشان می دهد. بنا بر این وقتی از واژه Patient برای تعریف یک موجودیت استفاده می شود، به دلیل آن است که هر مورد بیمار و نه بیمار به طور عام را توصیف نماید. شما تعریف موجودیت ها را چگونه شروع خواهید نمود؟ یک روش این است که زیر هر کدام از اسم ها در شرح مسئله خط بکشید. بیشتر اسم ها موجودیت های احتمالی هستند و خط کشیدن، شما را مطمئن می سازد که هیچ موجودیت بالقوه ای را نادیده نگرفته اید. یک شرح مسئله می تواند به این صورت باشد: بیماران از چه خدماتی استفاده می کنند؟ در اینجا واژه های بیماران و خدمات می توانند به عنوان دو موجودیت مستقل در مدل سازی مورد استفاده قرار گیرند. با انتخاب یک موجودیت که به نظر می رسد برای مسئله مورد نظر محوری می باشد، آغاز کنید. اگر شما یک پایگاه داده بیمار را طراحی نموده اید، شما ممکن است با موجودیت بیمار آغاز نمایید. وقتی که موجودیت محوری را انتخاب نمودید، آن را توصیف کنید و سپس به سراغ سایر موجودیت ها بروید.

صفت خاصه (Attribute)

صفت خاصه، یک موجودیت را توصیف می کند. به عبارت دیگر یک موجودیت دارای ویژگی ها یا صفاتی مخصوص به خود می باشد که ما می خواهیم موجودیت را با آن صفات خاص خود ذخیره نماییم و موجودیت ها چیزهایی در محیط هستند که ما می خواهیم در باره آنها اطلاعاتی را نگهداری نماییم. وقتی که یک موجودیت شناسایی شد، گام بعدی تعیین صفات یا داده هایی است که بایستی نگهداری شوند تا موجودیت را به صورت واقعی توصیف نمایند. برای موجودیتی به نام بیمار، صفات می تواند شامل: نام، نام خانوادگی، تاریخ تولد، نام پدر، تاریخ اولین مراجعه، وزن و غیره باشد. یک عنوان صفت در یک مدل داده یکتاست. شما ممکن است لازم داشته باشید که از یک پیشوند برای یکتا سازی عنوان صفت استفاده نمایید (مانند تاریخ عمل یا تاریخ مراجعه به جای تاریخ تنها).

قاعده این است که نام صفات با حروف کوچک در داخل مستطیل و در زیر نام موجودیت نوشته شود. شکل زیر نشان می دهد که موجودیت PATIENT دارای صفاتی از قبیل pt_code ، pt_name و غیره می باشد.

<p>بیمار کد بیمار. نام بیمار. نام خانوادگی بیمار. تاریخ تولد بیمار. مراجعه تاریخ</p>	<p>PATIENT</p> <p>pt_code pt_name pt_family pt_bIRTH_d pt_first_adM</p>
--	---

مشاهده می کنید که پیشوند pt به کرات مورد استفاده قرار گرفته است. امکان دارد که سایر موجودیت های پایگاه داده نیز صفاتی از قبیل name و lname را داشته باشند (مثلا برای موجودیت های پزشک، پرسنل و ...). بنا بر این ما یک پیشوند را اضافه نموده ایم تا یکتایی صفات تضمین گردد. تعریف صفات مستلزم بحث قابل توجهی با مشتری می باشد. شما بایستی هر صفتی را که احتمال می رود برای تصمیم گیری های حال و آینده مورد نیاز باشد، در نظر بگیرید. اما به حاشیه نپردازید و از ذخیره داده های غیر ضروری بپرهیزید. برای مثال

برای توصیف موجودیتی به نام دانشجو، شما معمولا تاریخ تولد را ثبت می کنید اما غیر محتمل است که شما بخواهید اندازه قد را ثبت کنید. اما وقتی که موجودیتی به نام بیمار را توصیف می کنید، ممکن است ثبت اندازه قد ضروری باشد، زیرا این صفت بعضی اوقات به تصمیمات پزشکی مرتبط می باشد.

یک صفت یک مقدار واحد دارد. که احتمالا نمی تواند پوچ یا null باشد و مقادیر متعدد نیز مجاز نمی باشند. اگر نیاز دارید که مقادیر متعددی را ثبت نمایید، علامت آن است که شما یک ارتباط یک به چند دارید و لازم است موجودیت دیگری را تعریف نمایید. بعدا در این باره بیشتر صحبت خواهیم کرد.

نمونه یا رکورد (instance)

یک نمونه، یک بار وقوع موجودیت است (در اینجا یک نمونه بیمار). همان طور که بعداً خواهیم دید، یک رکورد یا سطر نیز همین مفهوم را دارد. یک مدل داده معمولاً کمتر از ۱۰۰ موجودیت دارد. یک پایگاه داده می تواند به راحتی میلیون ها instance (سطر یا رکورد) برای هر موجودیت داشته باشد. تعداد رکوردهای موجود در یک پایگاه داده مالیات کشوری را می توانید تصور کنید. موجودیت ها در داخل کلاس های موجودیت و یا مجموعه ای از موجودیت ها از یک نوع جمع می گردند. یک کلاس موجودیت با یک نمونه (instance) از موجودیت تفاوت دارد. یک کلاس موجودیت فرم حالت عام یا تعریف یک چیز است. مثل یک مشتری. در حالی که یک نمونه از کلاس موجودیت نمایش دهنده یک موجودیت خاص است. مثل مشتری ۱۲۳۴۵. اصطلاحات موجودیت و کلاس اغلب به جای یکدیگر مورد استفاده قرار می گیرند. در واقع تعداد زیادی نمونه از یک موجودیت وجود دارد. به طور مثال در داخل کلاس مشتری، نمونه های زیادی از مشتریان مختلف در داخل پایگاه داده وجود دارد.

برای جلوگیری از سردرگمی، لازم است که هر نمونه در یک موجودیت به طور منحصر بفرد شناسایی شود. تصور کنید در یک شرکت حقوق فردی به نام رضا کریمی را ۱۰ درصد افزایش دهند. آیا می توان جستجویی بر اساس نام انجام داده و این تغییر حقوق را انجام داد؟ مسلم است که افراد زیادی با نام رضا ممکن است در آن شرکت وجود داشته باشند. ممکن است افزایش حقوق را برای نام خانوادگی کریمی انجام دهند. باز ممکن است افراد زیادی با این نام خانوادگی در شرکت (اگر بزرگ باشد) وجود داشته باشند. یا تصور کنید در یک بیمارستان بزرگ بخواهند یک اقدام خاص درمانی برای فردی به نام احمد انجام دهند. احتمال دارد بیمار یا بیماران دیگری با نام احمد در بیمارستان بستری شده باشند. اگر شرکت در طراحی پایگاه داده خود به گونه ای عمل کند که هر کارمند یک شماره منحصر بفرد داشته باشد، و یا پایگاه داده بیمارستان یک مشخصه یکتا و منحصر بفرد برای بیمار در نظر بگیرد و بازایی اطلاعات مربوط به یک فرد خاص از طریق آن مشخصه یکتا صورت گیرد، مشکلات یاد شده و تبعات آنها اتفاق نخواهد افتاد.

شناسه (Identifier)

یک صفت یا مجموعه ای از صفات که به طور منحصر بفرد یک رکورد (instance) از یک موجودیت را مشخص کند، شناسه نامیده می شود. شناسه برای دو مثالی که در بالا اشاره شد، می تواند به صورت یک کد پرسنلی برای یک کارمند و یک شماره منحصر به فرد پرونده برای یک بیمار باشد. بعضی اوقات ممکن است بیش از یک صفت در یک موجودیت منحصر به فرد باشند و یا ترکیبی از صفات به عنوان شناسه یکتا در نظر گرفته شود. تصور کنید جستجوی رکورد یک کارمند به نام رضا کریمی فرزند اکبر (از صفت نام، نام خانوادگی و نام پدر به عنوان شناسه استفاده می شود)، در یک شرکت متوسط فقط یک رکورد را نشان خواهد داد. در مدل داده صفتی که شناسه در نظر گرفته می شود، با علامت * مشخص می گردد. در موجودیت مقابل pt_code به عنوان شناسه مشخص شده است. هر موجودیتی بایستی یک شناسه داشته باشد.

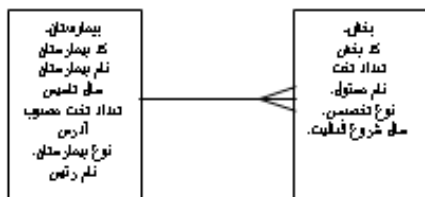
PATIENT
pt_code
pt_name
pt_family
pt_birth_d
pt_first_adm

پس تا اینجا یک موجودیت به نام patient ایجاد کردیم. صفات خاصه آن را که ذاتی بوده و تغییر نخواهند کرد، فهرست نمودیم. سپس یک شناسه منحصر بفرد که خاص هر فرد می باشد، برای آن تعیین نمودیم. به همین ترتیب سایر موجودیت ها (از قبیل پزشک، درمانگاه، صورتحساب، ویزیت و غیره) را تعیین نموده و صفات خاصه و شناسه آنها را نیز مشخص می کنیم. لازم به یادآوری است که نمونه هایی از مدل داده در انتهای فصل آورده شده است."

ارتباط (Relationship)

موجودیت ها مستقل نیستند. در دنیای واقعی، موجودیت ها به موجودیت های دیگر مرتبط می شوند. اگر هیچ ارتباطی بین موجودیت ها نبود، نیازی برای مدل داده ها و همین طور نیازی به پایگاه های داده رابطه ای وجود نمی داشت. یک فایل ساده (یک پایگاه داده منفرد) می توانست کافی باشد.

ارتباط دو سویه بوده و پیوند بین دو موجودیت را توصیف می کند و با یک قوس بین آنها نشان داده می شود. از آنجایی که ارتباط دو سویه است، دو توصیف گر ارتباط برای هر کدام از موجودیت ها دارد. هر توصیف گر ارتباط، درجه ای دارد که بیان می کند چه تعداد رکورد موجودیت دیگر ممکن است به هر رکورد از موجودیت توصیف شده مرتبط شود.



ارتباط یک به چند

یک ارتباط یک به چند با یک خط ارتباط دهنده با علامتی شبیه پای مرغ در سمت انتهایی چند تایی نشان داده می شود. رابطه مقابل را بین یک بیمارستان و بخش در نظر بگیرید. این ارتباط به این صورت خوانده می شود: یک بیمارستان می تواند چندین بخش داشته باشد اما یک بخش فقط به یک بیمارستان تعلق دارد. موجودیت بیمارستان به وسیله کد بیمارستان شناسایی می شود و صفاتی چون نام بیمارستان، سال تاسیس و غیره دارد.

ارتباط یک به چند به فراوانی در موقعیت های کاری وجود دارد. بعضی اوقات در یک شکل درختی اتفاق می افتد. یک شرکت را در نظر بگیرید. این شرکت دارای دواير مختلفی است اما یک دایره فقط متعلق به یک شرکت است. هر دایره دارای چندین واحد است، اما یک واحد فقط به یک دایره تعلق دارد و الی آخر. ما چنین ساختار درختی را در گزارش فروش بر اساس کشور، منطقه، استان و شهر نیز می توانیم مشاهده نماییم.

چرا ما موجودیت دیگر را اضافه می کنیم؟

تصور کنید جدولی مانند جدول زیر برای بیمارستان ها طراحی کرده ایم. در جدول زیر دو موجودیت به نام های بیمارستان و دانشگاه را می بینیم که در یک جدول آورده شده اند و وابستگی بیمارستان را به یک دانشگاه علوم پزشکی نشان می دهد. در نگاه اول به نظر کامل می آید، چرا که صفات لازم برای یک بیمارستان در آن آورده شده است. اما این جدول افزونگی اطلاعات را ایجاد خواهد نمود. همان طور که در شکل می بینید، کد دانشگاه و نام دانشگاه پنج بار تکرار شده است (با کامل شدن جدول این تکرار موارد بیشتر نیز خواهد شد). وقتی که بخواهیم داده هایی را در جدول وارد نماییم، مواردی را حذف کنیم و یا داده هایی را تغییر دهیم، افزونگی مشکلاتی را نشان می دهد این مشکلات به عنوان آنومالی های تغییر نامیده می شوند.

کد بیمارستان	نام بیمارستان	تاریخ تاسیس	تعداد تخت	تخصص	نوع مالکیت	آدرس	کد دانشگاه	نام دانشگاه	تعداد هیات علمی
۱	الزهرا	۱۳۴۰	۲۰۰	زنان	دولتی	تبریز- خیابان ارتش شمالی	۳	تبریز	۵۱۰
۲	اسدابادی	۱۳۴۵	۱۰۰	کودکان	دولتی	تبریز- خیابان بهار	۳	تبریز	۵۱۰
۳	رازی	۱۳۳۰	۳۵۰	روانپزشکی	دولتی	تبریز- جاده ائل گلی	۳	تبریز	۵۱۰
۴	امام	۱۳۳۵	۴۰۰	عمومی	دولتی	تبریز- جاده دانشگاه	۳	تبریز	۵۱۰

۵	شفا	۱۳۵۸	۶۰	عمومی	خصوصی	تبریز-خیابان منجم	۳	تبریز	۵۱۰
۶	سینا	۱۳۳۰	۵۰۰	عمومی	دولتی	تهران-خیابان سی تیر	۱	تهران	۷۰۰
۷	فقیه‌هی	۱۳۳۷	۶۰۰	عمومی	دولتی	شیراز- خیابان زند	۲	شیراز	۴۸۰

آنومالی وارد کردن داده

ما نمی توانیم در باره یک نام دانشگاه موردی را اضافه نماییم، مگر این که قبلا یک بیمارستانی داشته باشیم. در نظر بگیرید وقتی که می خواهیم رکوردی را برای دانشگاه زنجان وارد کنیم و هیچ بیمارستانی برای زنجان نداریم. ما نمی توانیم این مسئله را با وارد کردن null در فیلدهای قبلی نادیده بگیریم. چرا که شناسه ای به نام کد بیمارستان نیز null می گیرد که مجاز نمی باشد (یک شناسه در موجودیت نمی تواند null باشد). اگر ما یک جدول مجزا برای اطلاعات مربوط به دانشگاه داشته باشیم، پس می توانیم به راحتی اطلاعات مربوط به دانشگاه ها را اضافه نماییم بدون این که ضرورتی داشته باشد، حتما اطلاعات بیمارستان های آن را وارد کنیم. این بخصوص زمانی مفید خواهد بود که سایر سازمان ها مثلا وزارت بهداشت به دسترسی به اطلاعات دانشگاه ها نیاز پیدا کند. در جدول بیمارستان بایستی یک ستون برای وارد نمودن کد دانشگاه که در جدول دیگر شناسه است، اضافه شود تا ارتباط بین این دو برقرار شود.

آنومالی حذف داده ها

اگر داده هایی را از یک بیمارستان خاص حذف کنیم، ممکن است اطلاعات مربوط به اعضای هیات علمی دانشگاه مربوطه را نیز از دست بدهیم. برای مثال اگر ما جزئیات بیمارستان فقیه‌هی را حذف کنیم، ما تعداد اعضای هیات علمی دانشگاه شیراز را نیز حذف کرده ایم.

آنومالی های تغییر

تعداد اعضای هیات علمی متغیر است. بسیاری از دانشگاه ها لازم است تعداد آن را هر ماه تغییر دهند. تصور کنید اگر تعداد اعضای هیات علمی دانشگاه تبریز تغییر کند، این تغییر در هر پنج رکورد مربوط به تبریز بایستی اعمال گردد (و در یک جدول کامل نیاز به تغییر این مورد خیلی بیشتر باشد). یک خطر نیز وجود دارد و آن این که فردی فراموش کند تغییرات مربوط به دانشگاه تبریز را در تمامی سطرها انجام دهد. در نتیجه می تواند دو مقدار برای تعداد اعضای هیات علمی یک دانشگاه وجود داشته باشد. پس ملاحظه می کنید که اگر موجودیت دانشگاه از موجودیت بیمارستان جدا شود، نه تنها افزونگی را نخواهیم داشت، بلکه فقط یک تغییر ضروری خواهد بود و هیچ خطری از داده های ناسازگار نیز وجود نخواهد داشت.

ایجاد رابطه یک به چند

همان طور که قبلا ذکر گردید، هر موجودیت به یک جدول در یک پایگاه داده تبدیل می شود، نام موجودیت به نام جدول ، نام صفت به نام ستون و هر شناسه به یک کلید اصلی تبدیل خواهد شد. ارتباط یک به چند با افزودن یک ستون به موجودیت قرار گرفته در انتهای چند تایی خط ارتباط (در اینجا بیمارستان) انجام می گیرد. ستون اضافه شده، حاوی شناسه موجودیت سمت

دیگر خط ارتباطی (در اینجا دانشگاه) خواهد بود. در واقع اگر آن شناسه در موجودیت دیگر ذخیره نشود، هیچ راهی وجود ندارد که دانشگاهی را که بیمارستان آن در لیست آمده، شناسایی گردد. یعنی شناسه ارتباط این دو موجودیت را برقرار می کند.

کد بیمارستان	نام بیمارستان	تاریخ تاسیس	تعداد تخت	تخصص	نوع مالکیت	آدرس	کد دانشگاه
۱	الزهرا	۱۳۴۰	۲۰	زنان	دولتی	تبریز- خیابان ارتش شمالی	۱
۲	اسدابادی	۱۳۴۵	۱۰	کودکان	دولتی	تبریز- خیابان بهار	۱
۳	رازی	۱۳۳۰	۳۵	روانپزش- کی	دولتی	تبریز- جاده ائل گلی	۱
۴	امام	۱۳۳۵	۴۰	عمومی	دولتی	تبریز- جاده دانشگاه	۱
۵	شفا	۱۳۵۸	۶۰	عمومی	خصوصی	تبریز- خیابان منجم	۱
۶	سینا	۱۳۳۰	۵۰	عمومی	دولتی	تهران- خیابان سی تیر	۲
۷	فقیهی	۱۳۳۷	۶۰	عمومی	دولتی	شیراز- خیابان زند	۳

کد دانشگاه	نام دانشگاه	تعداد هیات علمی
۳	تبریز	۵۱۰
۱	تهران	۷۰۰
۲	شیراز	۴۸۰

توجه داشته باشید که ستون کد دانشگاه در هر دو جدول بیمارستان و دانشگاه وجود دارد. در جدول دانشگاه، کد دانشگاه کلید اصلی است و برای رکوردهای این جدول، یکتا و منحصر بفرد می باشد. در جدول بیمارستان، کد دانشگاه کلید خارجی است، زیرا در جدولی دیگر که در رابطه یک به چند در سمت یک قرار دارد، کلید اصلی است. جور کردن کلید اصلی- کلید خارجی روشی است برای ثبت ارتباط یک به چند بین دو جدول. خط رابطه یک به چند در شکل بالا بین دو جدول ترسیم شده است. اگرچه در این مثال کلید اصلی و کلید خارجی (کد دانشگاه) همنام می باشند، اما اجباری در این کار وجود ندارد و نام های متفاوت نیز می توان برای این ها در نظر گرفت. قرار دادن نام های مشابه جهت یادآوری این نکته است که دو جدول با هم رابطه دارند.

همان طور که مشاهده می شود، در جدول بیمارستان، هفت بیمارستان ثبت شده است که پنج بیمارستان مربوط به دانشگاه تبریز، یک بیمارستان مربوط به دانشگاه شیراز و یک بیمارستان نیز مربوط به دانشگاه تهران می باشد. اگرچه هر دانشگاه می تواند بیمارستان های زیادی داشته باشد، اما می توان دانشگاه جدیدی داشت که هنوز برای آن بیمارستانی وجود ندارد (از نظر تئوری). یعنی در اصطلاح مدل سازی داده، بسیاری از موارد می توانند مقادیر صفر یا بیشتر را داشته باشند، اما این مسئله اجباری است که برای هر مقدار در جدول بیمارستان برای فیلد کد دانشگاه، بایستی مقداری نیز برای کد دانشگاه در جدول دانشگاه وجود داشته باشد. این الزام که به نام قید یکپارچگی ارجاعی^۵ معروف است، دقت یک پایگاه داده را حفظ می نماید. کاربرد این قید بدین معنی است که هر کلید خارجی در یک جدول، یک کلید اصلی برابر با آن در همان جدول یا در یک جدول دیگر دارد. در این مثال آن به این معنی است که برای هر مقدار کد دانشگاه در جدول بیمارستان، یک مقدار متناظر در جدول دانشگاه وجود دارد. در نتیجه یک رکورد (سطر) کلید اصلی بایستی قبل از ایجاد رکورد (سطر) کلید خارجی متناظر آن ساخته شود. بنا بر این جزئیات برای یک دانشگاه، بایستی قبل از وارد کردن اطلاعات یک بیمارستان وابسته به آن دانشگاه وارد شود.

چرا کلید خارجی در جدول، در سمت چپ تایی یک ارتباط می باشد؟

زیرا هر رکورد از بیمارستان که دقیقا با یک رکورد از دانشگاه همراه است. قانون این است که بیمارستان بایستی در لیست یک و فقط یک دانشگاه وجود داشته باشد. بنا بر این فیلد کلید خارجی وقتی که در سمت چپ تایی ارتباط قرار دارد یک مقدار دارد. کلید خارجی در سمت یک تایی ارتباط قرار نمی گیرد، زیرا در آن صورت هر رکورد از دانشگاه می تواند با بیش از یک رکورد (چند بیمارستان) در جدول بیمارستان همراه شود و این نشان دهنده یک کلید خارجی چند مقداری می باشد. مدل رابطه ای به خاطر مشکلات پردازشی که ممکن است به وجود آید، از فیلدهای چند مقداری پشتیبانی نمی کند.

نکته: باید به این نکته توجه نماییم که شناسه الزامی نباید کد باشد، مثلا برای بیمارستان، نام بیمارستان نیز می تواند به عنوان شناسه باشد.

روابط چند به چند

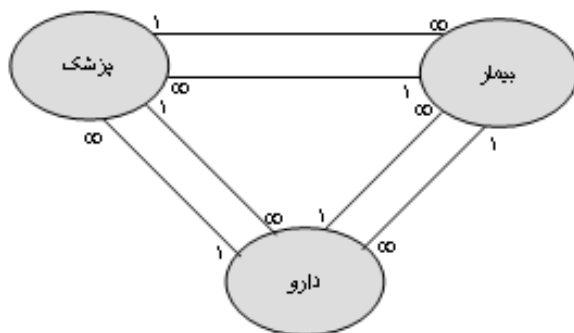
فرض کنیم که در قسمتی از طراحی مدل به این نتیجه رسیده ایم که در سیستم ما چند موجودیت از قرار زیر وجود دارد که عبارتند از:

۱- بیمار

۲- پزشک

۳- دارو

که این موجودیت ها روابطی به صورت زیر دارند:



۱. هر بیمار جهت معالجه با n پزشک ملاقات دارد.
۲. هر پزشک n بیمار را ویزیت می نماید.
۳. هر پزشک برای بیماران خود n دارو را تجویز می کند.
۴. هر دارو توسط n پزشک تجویز می گردد.
۵. هر بیمار جهت مداوا نیاز به n دارو دارد.
۶. هر دارو ممکن است برای n بیمار تجویز گردد.

می توان نتیجه گرفت که این روابط چند به چند هستند.

حال موجودیت بیمار را بررسی می نماییم

موجودیت دارو نیز از قرار زیر است:

دو موجودیت فوق با هم رابطه ای دارند و یک رابطه یک به چند از بیمار به دارو و همچنین یک رابطه یک به چند از دارو به بیمار وجود دارد. همچنین کد بیمار (شناسه موجودیت بیمار) در موجودیت دارو موجود می دارو (شناسه موجودیت دارو) باید در موجودیت بیمار عنوان گردد. در

به صورت جدول صفحه بعد خواهد بود:

کد بیمار	نام	نام خانوادگی	کد دارو
۱	علی	حسینی	M1
۲	زهرا	علوی	M100
۳	امیر	محمدی	M22
۱	علی	حسینی	M100
۴	رضا	رضایی	M52
۱	علی	حسینی	M22
۵	زینب	اکبری	M2

اشکالات جدول

- ۱- بیمار با کد ۱ (علی حسینی) جهت مداوا نیاز به سه نوع دارو (M100, M1, M22) دارد. کد بیمار، شناسه این موجودیت بوده، اما می بینیم که تکرار شده است.
- ۲- کد دارو نیز با توجه به تعریف صفات خاصه، مشکل دارد. اطلاعات تکراری بسیار زیاد است و اطلاعاتی مانند نام و نام خانوادگی که برای هر دارو عنوان شده، افزونگی اطلاعات ایجاد نموده است.

بیمار
* کد بیمار.
نام بیمار.
نام خانوادگی بیمار.
تاریخ تولد بیمار.
شماره تشخیصیه
چشمیت.
وضیعت، تامل

دارو
* کد دارو.
نام دارو
شرکت تولید کننده
تاریخ مصرف
قیمت دارو
توضیحات

باشد و به همین ترتیب کد نتیجه وضعیت جدول بیمار

حال جدول (موجودیت) دارو را نیز مورد بررسی قرار می دهیم.

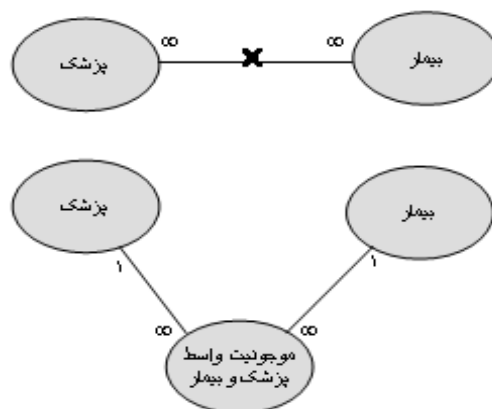
کد دارو	نام دارو	شرکت تولید کننده	کد بیمار
M1	a	کیمی دارو		۱
M2	b	رازی		۲
M3	c	رازی		۴
M1	a	کیمی دارو		۳
M100	d	ایران دارو		۱
M22	f	ایران دارو		۵
M1	a	کیمی دارو		۳
M22	f	سینا		۱
M100	d	رازی		۳

تمام مشکلات موجود برای موجودیت بیمار برای موجودیت دارو نیز صادق است. حال اگر موجودیت پزشک را هم وارد نماییم، برای پزشک هم این مشکلات صادق است. از آنجایی که ارتباطی میان بیمار و پزشک وجود دارد، فیلد شناسه پزشک هم در موجودیت بیمار راه پیدا می کند. در نتیجه با پیاده سازی سیستم با شرایط فوق این مشکلات به وجود می آید.

- افزونگی اطلاعات پیش می آید.
- کلید اصلی تکراری می شود.
- حافظه زیادی اشغال می شود.
- سرعت جستجو کاهش می یابد.

• تعاریف اولیه در تحلیل سیستم و اطلاعات ذاتی بیمار، پزشک، دارو و ... زیر سوال می رود.

به این نتیجه می رسیم که رابطه های چند به چند هر چند که در منطق طراحی به آن می رسیم، ولی در عمل قابل پیاده سازی نیستند و نباید وجود داشته باشند.



هر رابطه چند به چند همان طور که در شکل بالا می بینید، به دو رابطه یک به چند با یک جدول واسط تبدیل می شود و فیلدهای شناسه آنها در جدول واسط قرار می گیرد.

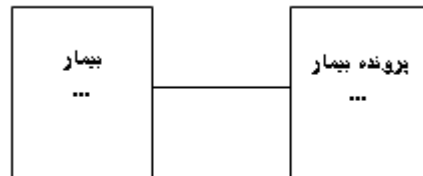
موجودیت واسط



پس به این ترتیب مشکلات مربوط به ارتباط ها برطرف گردید. یعنی ما جدول واسطی ایجاد کردیم که حاوی کلیدهای اصلی هر دو موجودیت بیمار و پزشک و احتمالا فیلدهایی اضافه تر می باشد. برای موجودیت های دارو و بیمار نیز می توان به همین روش موجودیت واسط بیمار- دارو ایجاد نمود.

ارتباط یک به یک

رابطه بیمار و پرونده بیمار را در نظر بگیرید.



رابطه بین دو موجودیت فوق به این صورت خوانده می شود.

➤ هر بیمار باید یک پرونده داشته باشد.

➤ هر پرونده باید برای یک بیمار باشد.

اگر برای هر کدام از موجودیت های فوق شناسه ای مانند شماره پرونده بیمار را در نظر بگیریم، از آنجایی که ارتباط بین دو موجودیت از طریق این شناسه (کلید) صورت می گیرد، لذا رابطه یک به یک بین این موجودیت ها وجود دارد. عده ای از نویسندگان معتقد هستند که در صورت حتمی بودن ارتباط (یعنی موردی وجود نداشته باشد که برای بیمار پرونده ای تشکیل نشود) بهتر است این دو موجودیت با هم ادغام شده و به یک موجودیت تبدیل شوند. اما اگر مواردی پیش بیاید که پرونده ای تشکیل نشود، در صورت ادغام دو موجودیت، افزونگی اطلاعات پیش می آید. چرا که تعدادی از فیلدهای مربوط به مشخصات پرونده خالی خواهد ماند.

موارد استفاده دیگر برای رابطه یک به یک، کدهایی است که به بعضی از اقلام اختصاص می یابد.

فرض کنیم اطلاعات دانشجویان در جدولی به شکل زیر ثبت شده است:

تلفن	آدرس	شرح وضعیت		وضعیت تحصیلی	نام خانوادگی	نام	کد دانشجو
		علت	تاریخ				
				اخراج			
				انصراف			
				در حال تحصیل			

در مورد فیلد وضعیت تحصیلی، دانشجو در یکی از وضعیت های زیر قرار می گیرد: اخراجی، در حال تحصیل یا انصرافی و ... فیلد شرح وضعیت برای خیلی از دانشجویان تکمیل نمی شود. بنا بر این:

۱. فضای زیادی از حافظه اشغال می شود.
۲. سرعت جستجو پایین می آید.
۳. قانون مربوط به صفات خاصه رعایت نمی شود، زیرا اطلاعاتی پر می شود که ذاتی نیستند.

برای رفع چنین مشکلاتی می توان وضعیت تحصیلی را به جداول دیگری انتقال داد. جداول مورد نظر می تواند به صورت زیر باشد:

جدول انصراف

کد دانشجو	تاریخ	علت

جدول اخراج

کد دانشجو	تاریخ	علت

این جداول از طریق کد دانشجو می توانند با جدول اصلی مرتبط شوند. از آنجایی که فیلد کد در این جداول به عنوان (شناسه) کلید اصلی تعریف می شوند و ارتباط جداول از طریق کلیدهای اصلی صورت می گیرد، لذا رابطه یک به یک محسوب می شود.

این نوع رابطه می تواند یک به صفر یا یک به یک باشد. مفهوم آن این است که می توانیم دانشجویی داشته باشیم که اخراج نشده باشد. می توانیم دانشجویی داشته باشیم که انصراف نداده باشد. اما در هر صورت دانشجویی با یک کد دانشجویی وجود دارد.

کیفیت مدل داده ها

دو معیار برای قضاوت در باره کیفیت مدل داده وجود دارد. بایستی خوب شکل داده شده باشد و از دقت بالایی برخوردار باشد.

۱- مدل داده خوب

یک مدل داده خوب به وضوح اطلاعات را به مشتری انتقال می دهد. شکل خوب به معنی آن است که قوانین ساختن مدل رعایت شده باشد. هیچ ابهامی وجود نداشته باشد. همه موجودیت ها نامگذاری شده باشند و همه موجودیت ها دارای شناسه باشند. اگر شناسه ها فراموش شده باشند، مشتری ممکن است استنباط نادرستی در باره مدل داده داشته باشد. همه ارتباط ها با استفاده از علامت های مناسب بایستی ثبت شوند. شرح ارتباط ها در هر جایی که امکان سر در گمی وجود داشته باشد، درج می شود. همه صفات خاصه یک موجودیت فهرست می شوند، زیرا صفات فراموش شده دو نوع مشکل به وجود می آورند. اول آن که نامشخص است که چه داده هایی در باره هر نمونه ذخیره خواهد شد. دوم، داده های مدل ممکن است بعضی از رابطه ها را از دست بدهند. صفات خاصه ای که می توانند مقادیر متعددی داشته باشند، باید به یک موجودیت دیگر تبدیل شوند. فقط با فهرست کردن همه صفات خاصه یک موجودیت است که موجودیت های ضروری که باید اضافه شوند، شناخته می شوند.

در یک مدل خوب، نام همه صفات ها با معنی و یکتا می باشد. نام موجودیت ها، شناسه ها، صفات و رابطه ها بایستی برای مشتری با معنی باشد، زیرا هدف از آنها توصیف دنیای مشتری است. تقریباً در تمامی موارد آنها اسامی هستند که مشتری به طور روزمره از آنها استفاده می کند. در انتخاب واژه ها مواظب باشید، زیرا آنها برای انتقال مفاهیم مهم می باشند. آزمون دقیق برای جامعیت یک مدل، عبارت است از این که مشتری بتواند مدل داده را برای کاربران بالقوه بخواند. جهت جلوگیری از سردرگمی، لازم است که اسامی یکتا باشند.

۲- مدل داده دقیق

علاقمندان موسیقی خیلی دوست دارند که یک سیستم استریو دقیق و واضح داشته باشند که با حداقل تغییر و یا بدون هر گونه تغییر در پخش صدا، کاملاً عملکرد اصلی را داشته باشد. یک مدل داده زمانی یک تصویر دقیق است که دنیایی را که در نظر دارد نمایش دهد، صادقانه توصیف نماید. در یک مدل دقیق، همه رابطه ها ثبت می شود و درجه دقت، بالا می باشد. هیچ تغییر یا مصالحه ای وجود ندارد. اگر رابطه در دنیای خارج به صورت چند به چند است، پس همان نیز بایستی در مدل نشان داده شود. مدلی که دو معیار بالا را داشته باشد، کامل، قابل فهم، دقیق و از نظر کلمات و عبارات درج شده صحیح می باشد.

خصوصیات یک مدل داده خوب

همه قواعد ساختاری رعایت شده باشد
هیچ ابهامی وجود نداشته باشد
همه موجودیت ها نامگذاری شده باشند
هر موجودیتی یک شناسه داشته باشد
همه ارتباط ها با استفاده از علامت های صحیح نشان داده شوند.
شرح رابطه ها برای جلوگیری از درک نادرست نوشته استفاده شود
همه صفات خاصه یک موجودیت فهرست شود
نام همه صفات خاصه با معنی و یکتا باشد

نرمال سازی داده ها

پس از تهیه مدل داده ها بایستی داده ها در جداول مرتبط سازماندهی شوند. این عمل نرمال سازی نامیده می شود. به عبارتی نرمال سازی، فرایند سازماندهی داده ها در جداول است. یک جدول نرمال نوعی فایل است که شرایط زیر را دارد:

➤ هیچ دو ردیف جدول با یکدیگر یکسان نیستند.

➤ ترتیب ردیف های جدول مهم نیست.

➤ ترتیب ستون های جدول مهم نیست.

➤ هر ستون دارای نامی یگانه است. به عبارت دیگر اسامی فیلدها تکراری نیست.

با استفاده از نرمال سازی، داده های زاید و قابل محاسبه حذف می شوند و داده های باقیمانده طوری دسته بندی می شوند که بتوان از طریق ارتباط دادن دسته ها کلیه اطلاعات مورد نیاز را به دست آورد.

فرض کنید یک بیمار سه بار به یک درمانگاه مراجعه کرده باشد. در یک پایگاه داده تک جدولی شما باید این اطلاعات را دو بار وارد کنید. چنانچه آدرس بیمار تغییر کند، لازم است آدرس وی را در هر سه مورد تغییر دهید و در غیر این صورت داده های شما ناسازگار خواهند بود. با وارد کردن اطلاعات مربوط به آدرس در یک جدول مجزا و ارتباط دادن آن با رکورد مراجعه نه تنها داده های اضافی و بعضا متداخل را حذف کرده اید، بلکه تغییرات را فقط در یک محل ثبت می کنید. شکل زیر مثالی از نمایش این جداول را نشان می دهد. توجه کنید دو جدول جداگانه برای بیماران و مراجعات ایجاد شده است. در جدول مراجعات هر بیمار با یک شماره مشخص می شود. بیماری به نام حسینی سه مراجعه دارد، بیمار دیگری به نام اکبری یک مراجعه دارد. می توان با در اختیار داشتن تنها یک آدرس از آقای حسینی می توان تمامی مراجعات وی را پیگیری نمود.

جدول بیمار

شماره پرونده	نام	نام خانوادگی	محل سکونت
۱۱۲۳	علی	حسینی	تهران
۱۱۴۵	حسین	اکبری	تبریز
۱۱۳۴	زهرا	کریمی	رشت

جدول مراجعه

شماره پرونده	تاریخ مراجعه	نام پزشک
۱۱۲۳	۸۲/۱۱/۱۲	رضایی
۱۱۲۳	۸۲/۰۱/۲۳	دیبا
۱۱۴۵	۸۲/۰۷/۱۲	کرمی
۱۱۲۳	۸۱/۰۲/۲۴	رضایی

برای نرمال سازی داده ها چندین قاعده وجود دارد. این قواعد فرم های اول، دوم و سوم نامیده می شوند (البته قواعد چهار و پنج نرمال سازی نیز وجود دارد که در این کتاب به آنها نخواهیم پرداخت). به عبارتی می توان گفت که عمل نرمال سازی معمولا در سه مرحله انجام می شود:

حالت اول نرمال ($FNF^{\hat{}}$) - در این حالت یک ستون نمی تواند شامل مقادیر چند گانه باشد.

مثلا در جدول یا موجودیت بیمار که حاوی مشخصات ذاتی بیمار است، نمی توان اطلاعاتی در باره ویزیت مانند تاریخ ویزیت، پزشک معالج و تشخیص را قرار داد، زیرا اگر بیماری چندین مراجعه داشته باشد، به ازای هر مراجعه یک بار مشخصات بیمار تکرار می شود.

حالت دوم نرمال (SNF^{\vee}) - در این حالت هر ستون غیر کلید باید به یک کلید کامل و نه جزئی از آن وابسته باشد. برای مثال اگر ترکیبی از فیلدهای کد ویزیت و کد بیمار را به عنوان کلید به کار می برید، کلیه ستون های جدول باید تنها برای ترکیب یک ویزیت و یک بیمار با یکدیگر به کار رود. بنا بر این فیلد آدرس بیمار به این جدول تعلق ندارد، زیرا فقط به کد بیمار وابسته است. پس در این مرحله صفت هایی مشخص می شوند که به کل کلید وابسته نیستند و باید از جدول حذف شوند. اگر جدولی نرمال سطح دوم باشد، حتما نرمال سطح اول نیز هست.

فرم نرمال سوم (TNF^{\wedge}) - بسیار مشابه فرم نرمال دوم است. در این فرم کلیه ستون های غیر کلید نباید به هیچ ستون غیر کلید دیگر وابسته باشند. در این مرحله کلیه فیلدهای های غیر کلیدی را که از فیلدهای های غیر کلیدی دیگر قابل دسترسی و یا محاسبه هستند، حذف می کنند. برای مثال اگر جدولی با فیلد کد بیمار دارید، با داشتن کد بیمار می توان مشخصات وی را از جدول بیمار به دست آورد. بنا بر این فیلد های نام بیمار، تاریخ تولد بیمار و ... از موجودیت های ویزیت و مراجعه حذف می شوند. یا مثلا اگر شما فیلدهای تاریخ بستری و مدت بستری را دارید، فیلد تاریخ ترخیص از دو فیلد دیگر قابل محاسبه است و لذا باید حذف شود.

برای آنکه جدولی TNF باشد، باید حتما FNF و SNF باشد.

هدف اصلی از مراحل نرمال سازی، کم کردن تعداد فیلدها و در نتیجه کاهش اندازه رکوردها در جدول می باشد. البته این مسئله به افزایش تعداد جداول و طولانی شدن زمان دسترسی به اطلاعات منجر می شود.

نکاتی در مورد مدل سازی

ایجاد یک مدل داده با دقت بالا زمان بر است. مشتری باید مسئله را به طور کامل توضیح دهد تا این که طراح پایگاه داده بتواند نیازهای مشتری را درک کند و آنها را به یک مدل داده ترجمه نماید. مدل سازی شبیه نمونه سازی است. طراح پایگاه داده، مدل پایگاه داده را به عنوان نتیجه تعامل با مشتری و به صورت گام به گام تهیه می کند. بعضی از مسائل در مراحل پیشرفت ایجاد مدل به فراوانی رخ می دهند و نکات زیر به شما کمک می کنند تا بسیاری از مسائل رایج را که با آنها مواجه می شوید، حل نمایید.

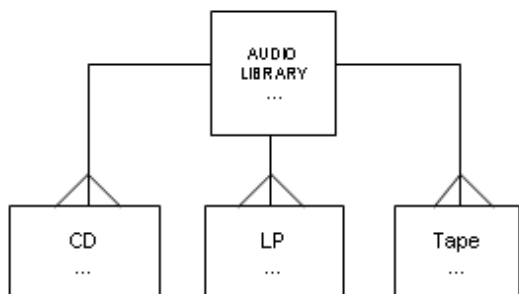
افت و خیز یک مدل

انتظار داشته باشید که مدل داده شما مرتبا گسترده و یا فشرده شود. اولین مدل داده در همان زمان که شما مرزهای نرم افزار را توسعه می دهید، گسترش خواهد یافت. شما درمی یابید که بعضی از صفات و موجودیت ها تکامل خواهد یافت و بعضی ارتباط های یک به چند به ارتباط های چند به چند تبدیل خواهند شد. مدل رشد خواهد کرد، زیرا شما موجودیت ها، صفات و ارتباط ها

را اضافه خواهید نمود تا استثناها را نیز پوشش دهند. مدل شما رشد خواهد کرد زیرا شما پیچیدگی دنیای واقعی را نشان می دهید. سعی نکنید که رشد آن را محدود نمایید.

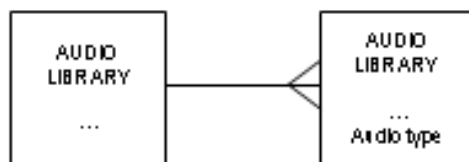
اجازه دهید مدل داده ها تا آنجایی که لازم است بزرگ شود. به عنوان یک قانون ثابت، انتظار داشته باشید مدل داده نهایی شما به اندازه تقریباً دو تا سه برابر تعداد موجودیت های مدل داده اولیه رشد کند.

وقتی که ساختارها را کلی می کنید، انتظار داشته باشید مدل داده شما منقبض شود. مدل داده خلاصه شده زیر را در نظر بگیرید:



این قطعه از مدل در مدل هایی که توسط مدل سازهای تازه کار تهیه می شود، متداول است. این مدل می تواند با شناخت این نکته که CD، LP و Tape همه انواع وسایل ضبط صدا هستند، خلاصه تر شود. برای نشان دادن همان داده ها، یک موجودیت عام تر یعنی ضبط صدا می تواند استفاده شود. البته آن ممکن است نیاز به یک صفت بیشتر داشته باشد تا انواع مختلف وسایل ضبط صدا را از هم متمایز نماید (به شکل نگاه کنید).

همان طور که شما موجودیت های جدید را کشف می کنید، مدل شما بزرگتر خواهد شد. وقتی که ساختارها را عام تر می کنید، مدل شما کوچکتر خواهد شد.



نکاتی در باره شناسه

اگر هیچ شناسه ساده و واضحی وجود نداشته باشد، خود یک شناسه بسازید. ساده ترین آنها یک کد بدون معنی می باشد (مثلاً personnumber یا Ordernumber). اگر خود یک شناسه را ایجاد کنید، می توانید منحصر بفرد بودن آن را نیز تضمین نمایید.

بر روی یک شناسه زیاد کار نکنید. بدترین موردی که می تواند وجود داشته باشد، یک کد محصول مثلاً ۲۲ کاراکتري است که می تواند به وسیله یک کارخانه تولیدی به کار برده شود. به این صورت که نه تنها به طور منحصر بفرد یک آیتم را شناسایی کند، بلکه رنگ، فرایند تولید آن و نوع ماده به کار رفته در ساختن آن را نیز مشخص نماید. رنگ، فرایند تولید و نوع ماده همه جزو صفات هستند. این کد محصول پیچیده بوده و اشتباه در ورود داده ها بسیار بالا بود، لذا افراد بسیار کمی می توانند نحوه تبدیل آن را به خاطر بسپارند.

یک شناسه باید فقط یک کار را انجام دهد، یعنی به طور منحصر بفرد هر رکورد از موجودیت را شناسایی نماید. بعضی اوقات که سعی می شود آن را به صورت دوتایی یا حتی سه گانه ایجاد کنند، این اقدام فقط کار مشتری را زیاد می کند.

وضعیت و ترتیب

هیچ ترتیبی در مدل داده وجود ندارد. موجودیت ها می توانند در هر جایی از طرح قرار داده شوند. شما معمولا موجودیت محوری را نزدیک به مرکز مدل قرار می دهید، فقط به خاطر این که تمایل دارید با موجودیت های مهم آغاز کنید (مثلا زمانی که سیستم اطلاعاتی دانشجو را مدل سازی می کنید کار را با موجودیت دانشجو شروع کنید). آنچه که واقعا مهم است، این است که همه موجودیت های مرتبط را شناسایی کنید.

صفات نیز هیچ ترتیبی ندارند. شما ممکن است که تمایل داشته باشید که آنها را همان طوری که شناسایی می شوند، فهرست نمایید. ایده خوبی است که برای قابل خوانا شدن، بعضی از صفات را به ترتیب فهرست نمایید. برای مثال نام، نام خانوادگی معمولا با هم هستند. این ضروری نیست، اما سرعت آزمون کامل بودن یک مدل داده را تسریع می نماید (مگر در مواردی بسیار نادر که چون امکان آن ضعیف است، صحبتی از آن به میان نمی آوریم). رکورد ها نیز چنین فرض می شوند که هیچ ترتیبی ندارند، هیچ رکورد اول، دومی یا آخری وجود ندارد. اگر لازم است که یک ترتیب خاص را مشخص نمایید، یک صفت خاصه ایجاد کنید تا ترتیب را بتوانید ثبت کنید. برای مثال: اگر رتبه بندی اعمال جراحی بالقوه بایستی ثبت شوند، یک صفت `op-rank` را برای ذخیره کردن این داده ها در نظر بگیرید. این بدان معنی نیست که رکورد ها به ترتیب `projrank` ذخیره می شوند، بلکه آن به شما اجازه می دهد که عبارت `order by` در `SQL` را برای گزارش کردن اعمال جراحی به ترتیب رتبه به کار ببرید.

اگر نیاز دارید که جزئیات یک نظم تقدم و تاخر را ذخیره نمایید (یعنی یک ترتیبی از رکورد ها وجود دارد و یک نیاز برای دانستن رکورد های پیشین و پسین وجود دارد) پس یک ارتباط بازگشتی را به کار ببرید. شکل:



صفات خاصه و تداوم

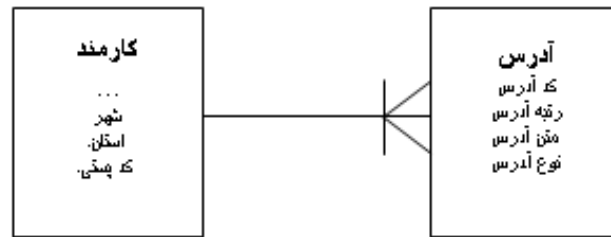
صفات خاصه بایستی ثابت باشند، یعنی همان مفهوم را برای هر رکورد نگهداری نمایند. یک مثال از یک صفت بی ثبات می تواند `financial status` باشد که حاوی نسبت قیمت به درآمد یا میزان بازگشت سرمایه باشد. یک صفت دیگر یعنی `financial status code` مورد نیاز است تا مشخص کند کدام معنی برای یک رکورد خاص به کار می رود. کد می تواند ۱ برای صفت نسبت قیمت به درآمد و ۲ برای صفت بازگشت سرمایه باشد. اگر یک صفت معنی صفت دیگری را مشخص کند، شما عدم ثبات را دارید. در این صورت نوشتن پرس و جوهای `SQL` بی نهایت چالش انگیز خواهد بود، زیرا داده های بی ثبات پیچیدگی پرس و جو را افزایش می دهند. بهتر است که صفات مجزایی برای نسبت قیمت به درآمد و میزان بازگشت سرمایه ایجاد نمایید.

نکاتی در مورد نام ها و آدرس ها

یک صفت خاصه کوچکترین جز داده هاست که به احتمال زیاد یک پرس و جو را تشکیل خواهد داد. اگر یک صفت بخش هایی داشته باشد (مثلا نام شخص) تعیین کنید که آیا این ها می توانند بخشی از یک پرس و جو را تشکیل دهند. اگر چنین است پس آنها را صفات مجزا در نظر بگیرید و آزمون پرس و جو را دوباره به کار ببرید. هنگام آزمون پرس و جو برای نام شخص ، شما معمولا تصمیم می گیرید که سه صفت خاصه را ایجاد کنید، نام کوچک، نام مستعار و نام خانوادگی.

در مورد عناوین دو نوع عنوان وجود دارد: عناوین پیشوندی (مثلا Mr، Ms، Mrs، Dr) و عناوین پسوند (مانند Jr و II). این ها مخصوصا اگر نام را به صفات مجزا تقسیم کرده باشید، بایستی صفات مجزایی باشند. البته این روش بهتر است و بعدا می توانیم گزارشات را راحت تر و متنوع تر داشته باشیم.

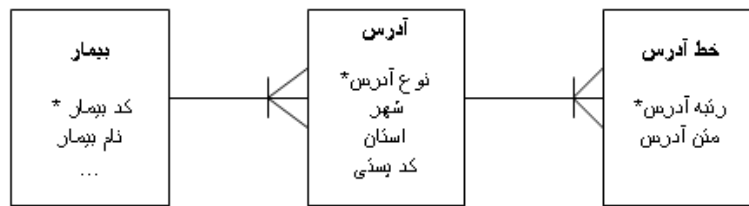
آدرس ها به نظر می رسد که به خاطر متغیر بودنشان نسبت به نام ها توجه بیشتری را بطلبند ، آدرس های کاری به پیچیدگی گرایش بیشتری دارند و آدرس های خارجی نیز پیچیدگی بیشتری را دارند. قطعه مدل داده در شکل زیر در بسیاری از موارد کاربرد دارد. افراد به عنوان بیمار یا کارمند معمولا آدرس های مختلف دارند، یک آدرس منزل، یک آدرس محل کار و احتمالا یک آدرس موقت برای شهر محل کار. از آنجایی که وجود آدرس های چدگانه و متغیر بودن آدرس ها مشکلاتی را برای پایگاه داده ها ایجاد می کنند، لذا از روش زیر برای جلوگیری از پاره ای از مشکلات می توان استفاده نمود.



ویژگی مشترک آدرس ها عبارتند از: شهر، ایالت، کد پستی و کشور. بعضی از این ها می توانند Null باشند. برای مثال کشور سنگاپور ایالت ندارد. در ایالات متحده شهر و ایالت می تواند از zipcode ها استخراج شود، اما این مسئله ضرورتا برای سایر کشورها درست نمی باشد. بعلاوه حتی اگر این برای هر کشور درست باشد، شما به یک قانون استنباط متفاوت برای هر کشور نیاز خواهید داشت که می تواند فراتر از حیطه بعضی از سیستم های پایگاه داده رابطه ها باشد. اگر یک تضمین مداوم وجود داشته باشد که هر آدرس در پایگاه داده برای یک کشور خواهد بود، پس احتمال کاهش افزونگی را با فقط با ذخیره کد پستی امتحان نمایید. مشاهده می کنید که مشکل داشتن آدرس های متعدد با یک ارتباط یک به چند نشان داده می شود. یک خط آدرس یک متن رشته ای است که به عنوان یک خط آدرس ظاهر می شود. اغلب یک توالی مجموعه وجود دارد که در آن این ها نشان داده می شوند. صفت خاصه رتبه آدرس، این ترتیب را ثبت می کند. در این حالت می توان در موجودیت آدرس برای یک بیمار یا کارمند، چندین آدرس را همراه با صفات خاصه مناسب دیگر از قبیل نوع آدرس، رتبه آدرس و غیره وارد نمود.

از آنجایی که نام شهر، استان و کد پستی بیشتر از متن آدرس برای بازیابی مورد نیاز می باشند، لذا می توان رشته یا متن آدرس را همراه با رتبه آدرس در یک موجودیت دیگر قرار داد. به این ترتیب می توان در تهیه گزارشات، آدرس افراد را به صورت کامل فهرست نمود. در عین حال ایجاد ستون های مجزا برای نام کوچک، پلاک و غیره وجود ندارد. شناسه نوع آدرس برای تمایز بین انواع آدرس ها به کار می رود. این قطعه مدل داده در موقعیت هایی که یک حرفه دارای آدرس های پستی و تحویل متفاوتی است، به کار می رود.

مدل داده خلاصه زیر نشان می دهد که چگونه می توان آدرس های متعدد و متغیر را مدل سازی نمود.

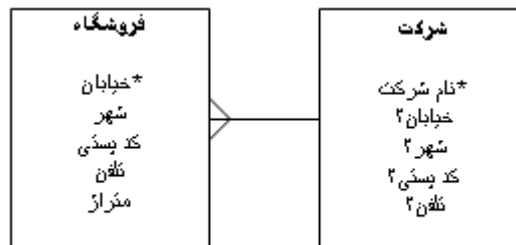


وقتی که داده ها را مدل سازی می کنید، می توانید از یک میانبر برای نام ها و آدرس ها استفاده کنید؟ اگر همه اجزای نام و آدرس را برای یک مدل داده بنویسید، زمان بر خواهد بود. از طرف دیگر باعث شلوغی مدل می شود و چیزی به دقت آن نیز نمی افزاید. توصیه عملی ما این است که این دو کار را انجام دهید.

ابتدا یک خط مشی برای نام ها و آدرس ها ایجاد کنید (مثلا همه نام ها به صورت سه قسمتی نام، نام مستعار و نام خانوادگی) ذخیره خواهند شد. دوم این که وقتی آنها از خط مشی تبعیت می کنند، شکل های کوتاه شده از صفات را برای نام ها و آدرس ها به کار ببرید. پس، از حالا به بعد ما نام و آدرس را به عنوان صفت هایی به کار خواهیم برد، به این صورت که وقتی پایگاه داده ایجاد می شود، بخش های این صفت ها ستون های مجزایی خواهند شد.

موجودیت های تک رکوردی

درباره ایجاد یک موجودیت با رکورد منفرد نگران نباشید. قطعه مدل داده در شکل زیر که یک فروشگاه منفرد را توصیف می کند، در نظر بگیرید. به خاطر این که مدل داده فقط یک شرکت را توصیف می کند، موجودیت شرکت فقط یک رکورد خواهد داشت. وجود چنین موجودیت تک رکوردی اجازه می دهد که حقایقی در باره شرکت نگهداری شود. بعلاوه در توسعه آن به صورت واقعی، انعطاف لازم فراهم می شود. در این حالت حتی اگر دو فروشگاه ترکیب شوند، مدل داده ها نیازی به اصلاح نخواهد داشت.



کلمات با ارزش

کلمات بسیار مهم هستند. آنها تمام آن چیزی هستند که ما برای فهماندن منظور خود به دیگران در اختیار داریم. اجازه دهید که مشتری کلمات را انتخاب کند. زیرا مدل داده به وی تعلق دارد و دنیای او را توصیف می کند. اگر شما کلمات خود را به مشتری تحمیل کنید، احتمالاً درک نادرست ایجاد می کنید.

مترادف ها

کلمات مترادف کلماتی هستند که معنی یکسان دارند. برای مثال: *task*, *assignment* و *project* ممکن است در دنیای واقعی به یک چیز اطلاق گردند. یکی از این واژه ها که ممکن است استفاده رایج تری داشته باشد، باید برای نامیدن یک موجودیت انتخاب شود. لازم است به مشتریان نام های رسمی موجودیت گفته شود و آنها تشویق شوند که یک واژه را برای توصیف آن بپذیرند یا می توان دید های منطقی از پایگاه داده ایجاد نمود که مشتریان را قادر نماید تا واژه های خود را به کار برند (در مورد

دیدها در فصل پنجم توضیح داده خواهد شد). مترادفها یک مسئله فنی نیستند، بلکه از این جهت که توافق با مشتری در باره یک واژه ضروری می باشد، یک مسئله اجتماعی تلقی می شوند.

واژه های هم آوا

اینها کلماتی هستند که در ظاهر یکسان هستند اما معانی متفاوتی دارند. مثلا، تاریخ فروش در کسب و کار می تواند معانی متعدد داشته باشد: برای پرسنل فروش، روزی است که مشتری سفارش خرید می دهد. برای وکیل، روزی است که قرارداد امضا می شود و برای بخش تولید، آن روزی است که مشتری کالا را تحویل می گیرد. در این مثال به طور منطقی راه حل آشکاری وجود دارد. تاریخ فروش را دوباره تعریف کنید تا واژه های مجزایی برای هر حوزه داشته باشید. تاریخ سفارش برای بخش فروش، تاریخ عقد قرارداد برای بخش حقوقی و تاریخ تحویل برای بخش تولید. وقتی که مشتریان متوجه نباشند که از یک کلمه برای موجودیت های مختلف یا صفت های مختلف استفاده می شود، کلمات هم آوا باعث اغتشاش واقعی می شوند، این کلمات را با پرسیدن سوالات زیاد از طیفی از مشتریان پیگیری و سپس به کار ببرید.

شکار استثناها

به شکار استثناها بروید. از مشتری چنین سوالاتی را بپرسید:

- آیا همیشه این چنین است؟
- آیا موقعیت های دیگری می تواند وجود داشته باشد که این بتواند یک ارتباط چند به چند باشد؟
- آیا استثنایی در آن وجود دارد؟
- آیا احتمال تغییر شی ها در آینده وجود دارد؟

همیشه برای استثناها آماده باشید و آنها را در مثال هایی جستجو کنید که مشتری استفاده می کند. مدل داده ها را دوباره طراحی کنید تا با لحاظ شدن استثناها دقت مدل را افزایش دهد.

نامگذاری ارتباط ها

نامگذاری ارتباط ها یک مدل داده ها را شلوغ می کند. در بسیاری از موارد مفهوم ارتباط می تواند به صورت صحیح استنباط شود. برچسب های توضیحی را فقط وقتی به کار ببرید که احتمال ابهام وجود دارد.

حفظ تناسب مدل داده ها

زمانی که شما موجودیت ها و صفاتی را به مدل داده ها اضافه می کنید، دقت را حفظ کنید. قبل از آن که جزئیات شناسه و صفات را کامل نکرده اید، موجودیت های دیگر را اضافه نکنید. با حفظ شکل و تناسب مدل داده ها از ابهام جلوگیری کنید. ابهامات به کرات موجب درک نادرست می شوند.

موجودیت های استفاده شده

آیا شما یک مدل داده آماده را می خرید؟ در بسیاری از موارد پاسخ مثبت است، زیرا یک مدل داده که قبلا مورد استفاده قرار گرفته است، ممکن است که دقت بالاتری را نسبت به یک مارک جدید داشته باشد. یک مدل داده استفاده شده بارها بررسی و بازنگری ها شده و می تواند نمایش دقیق تری از دنیای واقعی باشد.

شناسه های با معنی

وقتی که بعضی از صفت های موجودیت بتواند از مقدار شناسه استنباط گردد، نشان دهنده با معنی بودن شناسه است. (مثلا کد B78 برای یک کالا، کارمند را مطلع می سازد که سیاه رنگ است). بسیاری از مدیران داده عقیده دارند که شناسه ها یا کدهای با معنی مشکلاتی را ایجاد می کنند. هر چند که جلوگیری از شناسه های معنی دار به عنوان یک اقدام خوب در مدیریت داده ها پذیرفته شده است، اما آنها هنوز به طور وسیع مورد استفاده قرار می گیرند. بعد از یک شرح کوتاه از بعضی مثال ها در باره شناسه های با معنی، این بخش مزایا و معایب شناسه های معنی دار را مورد بحث قرار می دهد.

فاکتور شماره 98dec0001 یک شناسه با معنی است. پنج کاراکتر اول نشان می دهد که چه زمان فاکتور ایجاد شده است. چهار رقم انتهایی شماره فاکتور یک معنی دارد. ۰۰۰۱ نشان می دهد که آن اولین فاکتوری است که در دسامبر فرستاده شده است. بعلاوه آدرس الکترونیکی مانند mvdpas@bizzo.nl یک مفهوم دارد.

وقتی جنسیت را کدگذاری می کنیم، بعضی از سیستم ها، کدهای f و m و بعضی دیگر کدهای ۱ و ۲ را به کار می برند. کدهای f و m با معنی هستند اما کدهای ۱.۲ چنین نیستند و مسئولیت را به عهده کاربر می گذارد تا معانی مربوطه را به خاطر بیاورد.

کدهای f و m از حقیقتی مشتق می شوند که نشان دهنده آن است که آنها اولین حروف واژه های male و female می باشند. کدهای ۱ و ۲ از هیچ واقعیتی سرچشمه نمی گیرند و بنا بر این مفهومی ندارند. در اینجا با درج سند سازمان بین المللی استاندارد در باره استاندارد تبادل اطلاعات برای نشان دادن جنسیت انسانی (ISO 5218) مفهوم فوق را دوباره تکرار می کنیم.

"اهمیتی ندارد که یک حقیقت که مذکر است کد ۱ را و مونث کد ۲ را بگیرد. این استاندارد در اثر تلاش قابل توجه کشورهای عضو ایجاد شده و مطلقا به معنی اهمیت، رتبه بندی یا سایر مبناهایی که بتواند تبعیض جنسی را نشان دهد، نمی باشد."

در تعیین این که آیا یک شناسه با معنی است یا نه، فرد می تواند به روش و روندی که مقادیر جدید آن شناسه ایجاد می شوند، نگاه کند. اگر ایجاد شناسه بر اساس ویژگی هایی اتفاق افتد که در واقعیت ظاهر می شود، شناسه با معنی است. به این دلیل است که شناسه فاکتور 98dec0002 که شناسه اختصاص داده شده به دومین فاکتور تهیه شده در دسامبر ۹۸ می باشد، با معنی است. آن همچنین دلیل معنا دار بودن یک آدرس e-Mail می باشد. شناسه های های با معنی آشکارا مزایا و معایبی دارند و در این قسمت مورد بررسی قرار می گیرند.

مزایا و معایب شناسه های با معنی

مزایا	معایب
قابل شناخت و قابل به یاد آوری	تمام شدن شناسه
سادگی مدیریت	تغییرات در دنیای واقعی
	از دست رفتن معنا

مزایای شناسه های با معنی

قابلیت شناخت و قابلیت به یاد آوری

مردم اغلب اهمیت شناسه های با معنی را شناخته و به یاد می آورند. اگر برای مثال فردی یک e-Mail از rwatson@terry.uga.edu دریافت کند، این فرد احتمالا نام فرستنده و محل آن را می تواند استنباط نماید. اگر یک کارخانه برای مثال دو رقم انتهایی کد وسایل تولیدی خود را برای رنگ در نظر بگیرد، مثلا ۰۸ برای رنگ قهوه ای، کارکنان سریعا می توانند رنگ یک وسیله بسته بندی شده را از روی کد آن تشخیص دهد. مشتریانی هم که سفارش می دهند و احتمالا طرح را به خاطر ندارند، سریعا می توانند یک وسیله بسته بندی شده را به خاطر بیاورند.

سهولت اجرا

با استفاده از شناسه های با معنی، متمرکز نمودن مدیریت نسبتا آسان می شود. برای مثال، مدیریت آدرس های الکترونیکی می تواند در سطح دامنه (domain) انجام گیرد. به این ترتیب هر وقت که آدرس الکترونیکی جدیدی ایجاد شود، می توان از همسان بودن با سایر domain ها جلوگیری کرد. مثال دوم EAN^۱ یا بارکد فراورده های اروپایی است. صدور EAN می تواند در هر کشور انجام گیرد، زیرا هر کشور یک کد منحصر به فرد در EAN دارد. وقتی که یک کالای جدید تهیه می شود، یک کشور فقط مجبور است کدی را صادر کند که قبلا در آن کشور صادر نشده است. این روش که کشورها قراردادی امضا کنند که ابتدا اعلام کنند، یک کد EAN قبلا صادر شده است یا نه، زمان بر و خطا پذیر است. از طرف دیگر ایجاد یک پایگاه داده مرکزی از کدهای EAN یک راه حل می باشد. اگر هر صادر کننده کدهای EAN بتواند به چنین پایگاه داده های دسترسی یابد، نظم بین کشورها ایجاد می شود و از دوباره کاری جلوگیری می شود.

معایب شناسه های با معنی

به اتمام رسیدن شناسه

تصور کنید یک شرکت با یک شناسه کالا به صورت شش رقمی، تصمیم می گیرد که از سه رقم اول برای مشخص کردن منحصر بفرد گروه های تولیدی استفاده کند. در این گروه سه رقم باقی مانده برای مشخص کردن کالا به کار می رود. در نتیجه برای هر گروه تولید، فقط یک هزار جنس می توانند به این ترتیب کدگذاری شوند. حتی وقتی که بعضی از اعداد در یک گروه تولید خاص یا در کل گروه های تولید مورد استفاده قرار نگیرند، این مشکل باقی می ماند. در نظر بگیرید وقتی که گروه تولید ۰۱۰ شماره هایش به اتمام رسیده و کدهایش توسط کد گروه دیگر تولید مثلا ۹۴۰ تکمیل می شود. در ظاهر امر یک کد ثابت و سریع است، اما باعث مشکلاتی می شود، زیرا در این حالت یک گروه تولید کد شناسه را از گروه دیگر دریافت نموده و همه مجبورند به خاطر داشته باشند که یک استثنا وجود دارد. وقتی که شناسه های موجود به علت تمام شدن بایستی تغییر یابند، کدهای چاپی بر روی برچسب ها، بسته ها، قفسه ها و کاتالوگ ها دوباره باید انجام شوند. انجام دوباره یک فرایند هزینه بر است و بایستی از آن جلوگیری نمود. تمام شدن شناسه ها تنها مشکل شناسه های با معنی نیست. این مسئله می تواند برای شناسه های بدون معنی نیز رخ دهد. مشکل در اینجاست که سیستم های شناسه بامعنی گرایش بیشتری به تمام شدن دارند. برای جلوگیری از ته کشیدن شناسه و نگهداری شناسه های با معنی، بعضی از طراحان طول فیلد شناسه را افزایش می دهند. (مثلا چهار رقمی برای گروه تولید مثال قبل). در نتیجه فضای دیسک زیادی مورد نیاز خواهد بود تا شناسه های طولانی تر را ذخیره نماید.

تغییرات در دنیای واقعی

مشکل دوم شناسه های با معنی آن است که واقعیتی که آنها نشان می دهند، در حال تغییر است. مثلا یک شرکت یا دانشگاه نام خود را تغییر می دهد. مثلا دانشکده بازرگانی در دانشگاه جورجیا نام خود را به دانشکده بازرگانی Terry تغییر داد و در نتیجه آدرس های الکترونیکی نیز تغییر یافتند. فقط در صورتی که بخش با معنی شناسه به ندرت تغییر کند یا اصلا تغییر نکند، می تواند در طی یک زمان طولانی با معنی باقی بماند.

شناسه های بی معنی از تاثیر تغییرات فوق جلوگیری می کنند. در اغلب سیستم های حسابداری تلفن ها از یک شماره حساب برای مشخص ساختن یک مشتری استفاده می کنند نه یک شماره تلفن. این بدان معنی است که یک مشتری می تواند شماره تلفن خود را تغییر دهد، بدون این که باعث بروز هر گونه مشکل در کد شود. زیرا شماره تلفن، شناسه منحصر بفرد برای هر مشتری نمی باشد.

یک شناسه با معنی می تواند با معنی بودن خود را از دست بدهد

مشکلی را در نظر بگیرید که با شناسه ای اتفاق افتد که هم رنگ و هم نوع ماده یک قلم جنس را ثبت می کند. هر وقت هم پوشانی بین خصوصیات اتفاق بیفتد، چه اتفاقی می افتد. همیشه کارکنان و مشتریانی وجود خواهند داشت که معنی این شناسه های نه چندان با معنی و ما به ازای نادرست آنها را ندانند.

راه حل - شناسه های بی معنی

بیشتر مردم در ابتدا تمایل دارند که سعی کنند شناسه های با معنی ایجاد کنند. یک حسی وجود دارد که در صورت استفاده از یک شماره ساده برای مشخص ساختن یک محصول، چیزی در این میان گم می شود. اما نه تنها چیزی از دست نمی رود، بلکه چیزی هم به دست می آید. برای مشخص ساختن یک موجودیت به صورت منحصر بفرد، شناسه های بدون معنی منظور خاص خود را به خوبی بیان می کنند. برای توصیف ویژگی های موجودیت صفت خاصه مورد استفاده قرار می گیرند (مثلا رنگ، نوع جنس و ..). لذا به جای گنجاندن صفات خاصه در کد شناسه، می توان آنها را به صورت مجزا در موجودیت تعریف نمود. ایجاد یک تمایز آشکار بین نقش شناسه ها و صفات خاصه، مشکلات اجرایی کمتری برای حال و آینده ایجاد می کند.

هفت عادت مدل سازان موثر

اغلب، شکاف بزرگی بین عملکرد مدل سازان معمولی و متوسط وجود دارد. کسب یک بصیرت در مورد خصوصیات مدل سازان ماهر می تواند توانایی مدل سازی داده ها را در شما افزایش دهد.

۱- Immerse (غرق شدن)

با غرق کردن خودتان در محیط کار مربوطه دریابید که مشتری چه می خواهد. مقداری وقت را برای همراهی با مشتری و شرکت کردن در کار روزانه وی اختصاص دهید. با تجربه کردن مشکل توسط خودتان، درک وسیع تری از نیازهای مشتری خواهید داشت. مشاهده کنید، سوال پرسید، منعکس کنید و با تعداد بیشتری از مردم (مثلا مدیران، پرسنل عملیاتی، مشتریان و تامین کنندگان) صحبت کنید. هر قدر در باره مشکل بیشتر یاد بگیرید، برای ساختن مدل داده دقیق تر، بیشتر و بهتر مجهز می شوید.

۲- Challenge (چالش)

با مفروضات موجود چالش داشته باشید. استثناها را شکار کنید. محدوده مدل داده ها را بررسی کنید. سعی کنید که در باره مشکل از زوایای مختلف فکر کنید (مثلا چقدر احتمال دارد که یک مدیر با آن مشکل مواجه شود). جهت تشویق به جستجوی راه حل های رفع موانع، یک جلسه بارش افکار با مشتری برگزار کنید.

۳- Generalize (کلی کردن)

جهت ساده کردن مدل داده ها، هر موقع که امکان داشته باشد، تعداد موجودیت ها را با استفاده از ساختارهای کلی سازی کاهش دهید. معمولا درک مدل داده ساده آسانتر بوده و در اجرا هزینه کمتری خواهد داشت. جهت قضاوت مناسب در باره مشابهت موجودیت های متفاوت، مدل سازهای خبره می توانند ماورای تفاوت های سطحی را ببینند.

۴- Test (آزمودن)

مدل داده را با خواندن آن برای خودتان و برای افرادی که با مسئله آشنایی دارند، بیازمایید. هر دو جهت هر ارتباط را تست کنید. (یک بخش فقط به یک بیمارستان تعلق دارد). یک نمونه اولیه ایجاد کنید تا مشتری بتواند به صورت تجربی آن را آزمایش کند. آزمایش کردن بسیار مهم است، زیرا بازسازی یک مدل داده بسیار ارزان تر تمام می شود، اما برای تعمیر سیستمی که بر پایه یک مدل داده غیر صحیح ایجاد می شود، هزینه زیادی صرف می شود.

۵- Limit (محدود کردن)

برای زمان تهیه مدل و مرزهای آن، محدودیت های موجهی فراهم کنید. اجازه ندهید که مرحله مدل سازی تا ابد ادامه یابد. مرزهای پروژه را زودتر کشف کنید و آنها را علامت گذاری کنید. مگر این که دلایل اجباری برای گسترش پروژه وجود داشته باشد. معمولا از آنجایی که گفتن "بلی" آسانتر از گفتن "نه" می باشد، لذا به پروژه های بسیاری اجازه داده می شود که گسترش پیدا کنند. اما با طولانی شدن زمان اجرا، با وعده های بسیار و افزایش طول عمر پروژه، مشتری را آزار می دهید. موجودیت های محوری و صفت های اصلی را که اغلب مسائل را حل خواهند نمود، تعیین کنید و مدل داده ها را به این موجودیت های اصلی محدود نمایید. به محدودیت های زمانی و بودجه ای توجه داشته باشید.

۶- Integrate (یکپارچه نمودن)

گامی به عقب بگذارید و با دقت بنگرید که تا چه حد پروژه شما با معماری اطلاعات سازمان مناسبت دارد. تا جایی که امکان دارد با سیستم های موجود ادغام نموده و از دوباره کاری جلوگیری کنید. تا چه حد مدل داده شما با مدل داده های شرکت و مدل های سایر پروژه ها تناسب دارد و آیا شما بخشی از یک مدل داده موجود را به کار می برید؟ یک مدل ساز ماهر در عین حال هم جزئیات یک پروژه مدل داده و هم تصویر بزرگ منابع داده ای شرکت را بایستی بتواند ببیند.

۷- Complete (کامل کردن)

یک مدل ساز خوب، مدل های داده را نامشخص و تعریف نشده رها نمی کند. همه موجودیت ها، صفات خاصه و ارتباط ها به طور دقیق تعریف می شوند، ابهامات برطرف می شوند و بر روی استثناها کار می شود. از آنجایی که ارزش کامل یک مدل داده فقط وقتی درک می شود که سیستم کامل می شود، مدل ساز داده بایستی تا زمانی که سیستم اجرا می شود، کماکان با پروژه درگیر باشد تا بتواند در کل مراحل تداوم ایجاد نموده و حل مشکلات سیستم را تضمین نماید.

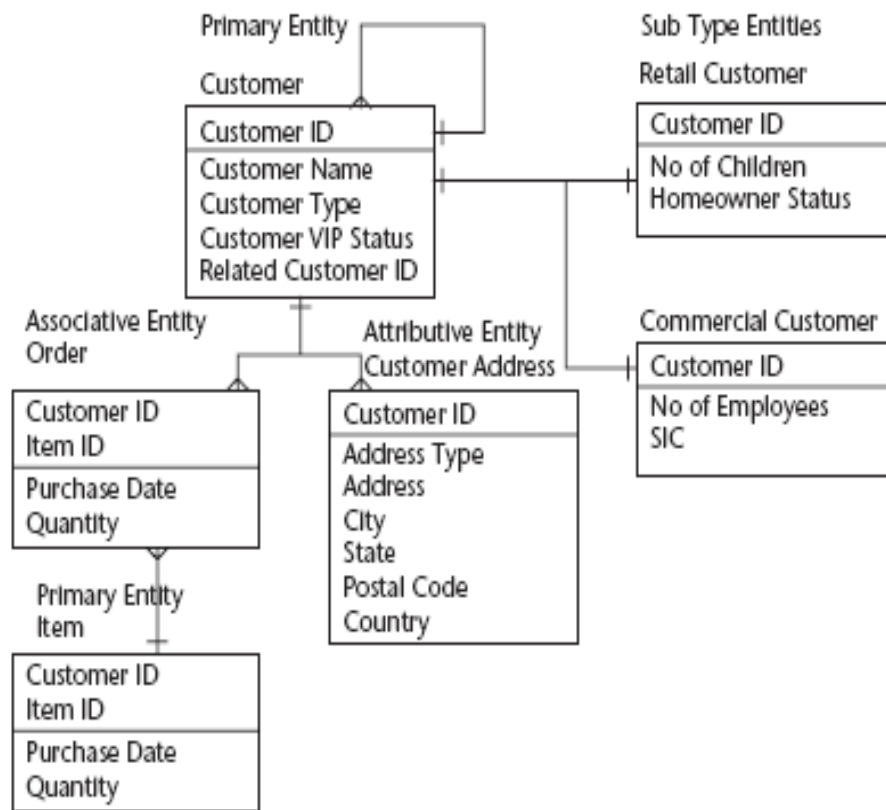
خلاصه

مدل سازی داده اولاً روشی برای ساختن الگوی داده ها و رابطه های آن و یک نمایش گرافیکی از یک پایگاه داده است. ثانياً روش برقراری ارتباط در یک طرح پایگاه داده می باشد. هدف از تهیه مدل داده ها، شناسایی حقایقی است که باید در یک پایگاه داده ذخیره شود. ساختن یک مدل داده، یک مشارکت بین مشتری، نماینده کاربران نهایی پایگاه داده و یک طراح می باشد. واحدهای ساختمانی یک مدل داده شامل موجودیت، صفات خاصه، شناسه و رابطه ها می باشد. یک مدل داده خوب، که به

معنی رعایت شدن قوانین ساختاری می باشد، به طور شفاف اطلاعات را به مشتری انتقال می دهد. یک مدل داده دقیق دنیای واقعی را به طور صادقانه منعکس می کند. یک مدل داده، یک تصویر در حال تکامل است. هر تغییری می بایست یک گام در بهبود کیفیت آن باشد. کیفیت مدل داده ها می تواند فقط با درک زمینه ای که مدل در آن زمینه استفاده خواهد شد، مشخص می گردد. انتظار داشته باشید که یک مدل داده گسترده و فشرده شود. یک مدل داده هیچ ترتیبی ندارد. اگر ترتیبی مورد نیاز باشد، یک صفت خاصه برای آن ایجاد کنید. یک صفت خاصه بایستی مفهوم یکسان برای هر رکورد داشته باشد. موثرترین مدل سازها از هفت عادت برخوردار هستند که عبارتند از: غرق شدن در محیط مورد نظر، چالش، کلی کردن، آزمودن، محدود کردن، یکپارچه سازی و کامل نمودن مدل داده ها.

نمونه ای از مدل های داده در بخش بازرگانی و بهداشت و درمان در صفحه بعد نشان داده شده است:

نمونه ای از یک مدل داده ها در حیطه بازرگانی



نمونه مدل داده ها در زمینه هزینه مراقبت بهداشتی

