بسم الله الرّحمن الرّحیم

پایگاه داده ها

(Database)

۲ واحد

دانشکده شهید منتظری مشهد

مدرس: قربانعلی عربی

(ali-arabi.ir)

۳-۱٤- درس پایگاه داده‌ها		نظری	عملی
	تعداد واحد	۲	-
نوع درس: تخصصی	تعداد ساعت	۳۲	-
پیش‌نیاز: -	هم نیاز: -		

هدف کلی درس: آشنایی با مفاهیم تخصصی و معماری چند سطحی سیستم پایگاه داده و انواع پایگاه‌های داده, آشنایی با داده‌های حجیم, معرفی مدل‌های داده خصوصاً مدل رابطه‌ای و نحوه ترسیم نمودار ER و سطوح نرمال‌سازی پایگاه داده و آشنایی با زبان SQL استاندارد.

الف- سرفصل آموزشی و رئوس مطالب

ردیف	رئوس محتوا	زمان یادگیری (ساعت)	
		نظری	عملی
۱	مقدمه‌ای بر فایل‌ها عناصر و اجزای فایل، مشکلات فایل، نسل‌های ذخیره و بازیابی اطلاعات، داده‌های حجیم	٤	-
۲	تعریف پایگاه داده‌ها عناصر تخصصی پایگاه داده، ویژگی‌های سخت‌افزار، معرفی انواع نرم‌افزار، انواع کاربر، ویژگی‌های داده, انواع پایگاه‌های داده و کاربردهای آن‌ها DBMS و RDBMS و ORDBMS	٤	-
۳	معماری پایگاه داده‌ها معماری کلاینت- سرور، معماری ANSI/SPARCو... دید داخلی، دید ادراکی، دید خارجی – ارتباطات بین دیدها- زبان میزبان – زبان فرعی داده- مدیر پایگاه داده، وظایف مدیر پایگاه داده، دیکشنری داده‌ها	٤	-
٤	سیستم مدیریت پایگاه داده وظایف سیستم مدیریت پایگاه داده، ارتباط سیستم مدیریت پایگاه داده و سطوح معماری پایگاه داده	۲	-
٥	روند اجرای درخواست کاربر در سیستم نحوه ارتباط، نحوه اجرای درخواست	۲	-
٦	انواع روش‌های مدل‌سازی داده توصیف و تشریح مدل‌های داده سلسله‌مراتبی، شبکه‌ای، رابطه‌ای ER، رابطه‌ای-شیءگرا با مزایا و معایب آن‌ها	٤	-
۷	مدل داده رابطه‌ای رابطه، ویژگی، تاپل، بسط، کاردینالیتی مفاهیم موجودیت، موجودیت ضعیف، فرا موجودیت، مفاهیم رابطه، رابطه یک‌به‌یک، رابطه یک به چند، رابطه چند به چند مفاهیم ویژگی (صفت)، صفت کلید تخصصی ، صفت کلید خارجی، صفت استنتاجی، صفت چندگانه، تعریف جامعیت – قواعد جامعیت در مدل داده رابطه‌ای	٤	-
۸	پیاده‌سازی عملیات روی رابطه‌ها زبان SQLاستاندارد و شرح دستورات تخصصی تعریف داده، دستکاری داده و مدیریت داده، ایجاد پرس‌وجوهای نمونه‌ای روی پایگاه داده	٤	-
۹	نرمال‌سازی	٤	-

	هدف از نرمال‌سازی، فرم اول نرمال، فرم دوم نرمال، فرم سوم نرمال		
جمع		۳۲	-

ب- مهارت های عمومی و تخصصی مورد انتظار

فراگیر پس از گذراندن درس باید قادر باشد ساختار پایگاه داده رابطه‌ای و انواع آن‌ها را تعریف نماید، تعریف و کاربرد دادههای حجیم را توضیح دهد، روش‌ها و مدل‌های طراحی ساختار پایگاه داده را بشناسد، بتواند از دستورات SQLاستاندارد استفاده نماید و با مفهوم نرمال‌سازی آشنا شود.

ج - منبع درسی (حداقل سه مورد منبع فارسی و خارجی)

عنوان منبع	مؤلف	مترجم	ناشر	سال انتشار
مقدمه‌ای بر پایگاه داده‌ها	روحانی رانکوهی		جلوه	۱۳۹۰
مفاهیم سیستم‌های پایگاه داده	آبراهام سیلبرشاتس- هنری اف کورت-اس سودارشان	عین الله جعفرنژاد قمی	علوم رایانه	۱۳۹۲
بانک اطلاعاتی	مصطفی حق جو		دانشگاه علم و صنعت	۱۳۹۲

د - استانداردهای آموزشی (شرایط آموزشی و یادگیری مطلوب درس)

مساحت تجهیزات و وسایل موردنیاز درس

کلاس تئوری، تخته وایت برد، ویدئو پروژکتور

ویژگی‌های مدرس (مدرک تحصیلی، مرتبه علمی، سوابق تحصیلی و تجربی):

کارشناسی و کارشناسی ارشد نرم‌افزار + ۳ سال سابقه کاری در حوزه بانک‌های اطلاعاتی

روش تدریس و ارائه درس (سخنران، مباحثه‌ای، تمرین و تکرار، کارگاه، آزمایشگاه، پروژه‌ای، پژوهش، گروه، مطالعه موردی و...):

سخنران، مباحثه‌ای، تمرین، پروژه‌ای

روش سنجش و ارزشیابی درس (پرسش‌های شفاهی، حل مسئله، آزمون کتبی، عملکردی- آزمون شناسایی (عیب‌یابی- رفع عیب و ...) انجام کار در محیط‌های شبیه‌سازی‌شده، تولید نمونه کار (انواع دست‌ساخته‌ها) پرسش‌های عملی و انشایی، مشاهده رفتار (مسئولیت‌پذیری، رعایت اخلاق حرفه‌ای و ...) پوشه مجموعه کار، ارائه مقالات و طرح‌ها گزارش فعالیت‌های تحقیقاتی، خود سنجی و ...):

پرسش‌های شفاهی، حل مسئله، آزمون کتبی

۱– مقدمه ای بر فایل ها

سال‌ها پیش و قبل از تولید مفهومی به نام پایگاه داده، برای ذخیره کردن اطلاعات از سیستم فایلینگ استفاده می‌شد. در این سیستم، اطلاعات وارد شده در برنامه، درون یک یا چند فایل نوشته می‌شد و برنامه‌نویس مجبور بود همه الگوریتم‌های موردنیاز برای مرتب‌سازی یا جستجوی داده‌ها را درون نرم‌افزار پیاده‌سازی کند.

هر فایل شامل مجموعه‌ای از داده های مرتبط به هم است. مانند داده‌های مربوط به دانشجویان یک دانشگاه.

۱–۱– عناصر و اجزای فایل

داده‌های مربوط به هر یک از اجزای فایل، یک رکورد نام دارد. به عنوان مثال، در یک دانشگاه داده‌های مربوط به هر دانشجو تشکیل یک رکورد را می‌دهند. لذا می‌توان گفت که هر فایل، مجموعه‌ای از چند رکورد است. اگر باز هم دقیق تر به فایل دانشجویان دانشگاه پرداخته شود، مشاهده می‌شود که هر دانشجو ممکن است چند قلم داده داشته باشد: مثل نام دانشجو، تعداد واحدهایی که گذرانده، نمره هر درس و.... به هر یک از اجزای یک رکورد، فیلد گفته می شود. لذا می‌توان گفت که هر رکورد مجموعه‌ای از چند فیلد است.

داده ها ممکن است به ۴ روش در فایل ذخیره شده سپس بازیابی شوند:

۱- داده‌ها کاراکتر به کاراکتر در فایل نوشته شده، سپس کاراکتر به کاراکتر از فایل خوانده شوند.
۲- داده‌ها به صورت رشته‌ای از کاراکترها در فایل نوشته شده، به صورت رشته‌ای از کاراکترها دستیابی شوند.
۳- داده‌ها در حین نوشتن بر روی فایل، با فرمت خاصی نوشته شده، سپس با همان فرمت خوانده شوند(عددی، کاراکتری، رشته ای).
۴- داده ها به شکل ساختمان یا ساختار(رکورد) بر روی فایل نوشته شده، سپس به صورت ساختمان از فایل خوانده شوند.

داده‌ها ممکن است در فایل به دو صورت ذخیره شوند:

۱- اسکی یا متن
۲- باینری

این دو روش در موارد زیر با یکدیگر تفاوت دارند:

۱- نحوه ذخیره شدن اعداد بر روی دیسک
۲- تعیین انتهای خط
۳- تعیین انتهای فایل

تفاوت متنی و باینری:

۱- در فایل متنی، اعداد به صورت رشته‌ای از کاراکترها ذخیره می شوند ولی در فایل باینری اعداد به همان صورتی که در حافظه قرار می گیرند بر روی دیسک ذخیره می شوند.
۲- درفایل متنی، کاراکتری که پایان خط را مشخص می کند، در حین ذخیره شدن بر روی دیسک، باید به کاراکترهای CR/LF تبدیل شود و در حین خوانده شدن، عکس این عمل باید صورت گیرد، یعنی کاراکترهای

CR/LF باید به کاراکترهای تعیین کننده پایان خط تبدیل شوند و بدیهی است که این تبدیلات مستلزم وقت است، لذا دسترسی به اطلاعات موجود در فایل های متنی کندتر از فایل‌های باینری است.

۲-۱- مشکلات فایل

برای درک دلایلی که منجر به کنار گذاشتن سیستم فایلینگ و عرضه سیستم‌های مدیریت پایگاه داده شد، ابتدا باید معایب فایلینگ را بررسی کنیم:

الف) در سیستم فایلینگ، برنامه‌نویس مجبور است برای هر کاربرد خاص، فایلهای مجزایی را طراحی و پیاده‌سازی کند و از آنجا که تمام عملیات ذخیره‌سازی و بازیابی توسط کدهای برنامه کنترل می‌شود، هر تغییر کوچکی در برنامه مستلزم بازنویسی کدهای نرم‌افزار است. به این ترتیب تولید یک نرم‌افزار به حجم بالایی از کدنویسی نیاز دارد.

ب) سیستم دچار افزونگی و داده‌های تکراری(Redundancy) در ذخیره‌سازی اطلاعات می‌شود؛ به این معنی که مثلاً نشانی دانشجو یک‌بار در فایل‌های نرم‌افزار «اداره رفاه»، و بار دیگر در فایل‌های نرم‌افزار «اداره آموزش»، ثبت می‌شود و همین مسأله «تکرار در ذخیره‌سازی» را به دنبال خواهد داشت.

ج) جدا بودن سیستم‌های ذخیره‌سازی اطلاعات از هم، مشکل دیگری را به دنبال دارد که آن را ناسازگاری داده-ها(Conflict) می‌نامیم. برای مثال اگر فردی در سیستم پرسنلی با نام «امیر محمودی» و در سیستم آموزشی به اسم «امیر محمودی دهکردی» معرفی شده باشد، داده‌های مربوط به این فرد در دو زیرسیستم، ناسازگار تلقی خواهند شد.

د) به دلیل وجود فایل‌های متعدد، اعمال استاندارد در شیوه ذخیره‌سازی داده‌ها بسیار دشوار است و به عنوان مثال ممکن است برای ذخیره‌سازی نام خانوادگی در فایلهای مختلف، از اندازه‌های متفاوتی استفاده شود.

و) دشواری در به‌روزرسانی(Update) داده‌ها

با وجود این که هنوز هم برای تولید برنامه‌های کوچک از سیستم فایلینگ استفاده می‌شود، اما تولید نرم‌افزارهای یکپارچه که همه نیازهای یک سازمان را رفع کند و برای مثال شامل سیستم پرسنلی، مالی و آموزشی یک دانشگاه باشد، نیازمند استفاده از پایگاه داده است. بهره‌گیری از یک «سیستم مدیریت پایگاه داده» باعث ایجاد تمرکز در ذخیره‌سازی اطلاعات می‌شود و در صورت طراحی صحیح، افزونگی و ناسازگاری اطلاعات را از بین می‌برد. به عنوان مثال، مشخصات پرسنلی تنها یک‌بار و با رعایت استانداردی مشخص، ذخیره خواهد شد. علاوه بر این، ذخیره‌سازی و بازیابی اطلاعات بر عهده این سیستم گذاشته می‌شود و برنامه‌نویس مجبور نیست مانند روش فایلینگ، برای ایجاد یا تغییر فایل‌ها، اضافه کردن رکورد و جستجو و مرتب‌سازی، کدهای اضافی بنویسد.

۳-۱- نسل های ذخیره و بازیابی اطلاعات

الف) نسل اول – فایلهای ساده ترتیبی

در این نسل رسانه خارجی معمولاً نوار بوده است. این نسل را میتوان نسل بی‌نرم‌افزار واسط نیز نامید. مشخصات کلی این نسل عبارتند از:

۱- ساختار فایلها ترتیبی است.

۲- ساختار فیزیکی همان ساختار منطقی فایل است.

۳- تنها روش پردازش فایلها، پردازش یکجا یا دسته‌ای(Batch Processing) است.

۴- نرم افزار عملیات ورودی/خروجی را انجام می‌دهد. نرم افزار واسطی برای مدیریت پردازش فایلها وجود ندارد.

۵- طراحی ساختار فیزیکی فایلها هم، برعهده کاربر است.

۶- هرگونه تغییر در ساختار داده‌ها و یا رسانه‌های ذخیره‌سازی سبب بروز تغییر در برنامه و بازنویسی و کامپایل آن می‌شود.

۷- داده ها برای کاربرد خاصی طراحی و سازماندهی می‌شوند.

۸- اشتراک داده‌ها(Data Sharing) مطرح نیست.

۹- تکرار در ذخیره‌سازی داده‌ها در بالاترین حد است.

۱۰- برای انجام عملیات بهنگام‌سازی، الزاماً فایل دیگری ایجاد و تغییرات را در آن وارد کرده، نسخه قدیمی را به عنوان «فایل پدر» نگهداری می‌کنند و به این دلیل نسخه های متعددی از یک فایل نگهداری می‌شوند.

ب) نسل دوم – شیوه های دستیابی(AM)

این نسل را باید نسل شیوه‌های دستیابی(Access Methods) نامید. مهمترین ویژگی این نسل را باید پیدایش نرم افزارهای موسوم به «شیوه های دستیابی» و همچنین ایجاد رسانه های با دستیابی مستقیم (یعنی دیسک) دانست. نرم افزار شیوه دستیابی، نرم افزاری است که به جنبه‌های فیزیکی محیط ذخیره‌سازی و عملیات در این محیط می‌پردازد. به نحوی که دیگر برنامه کاربر نیازی به پرداختن به این جنبه ها را ندارد. مشخصات این نسل عبارتند از:

۱- نرم افزار واسط برای ایجاد فایلها با ساختارهای گوناگون بین برنامه های کاربردی و محیط ذخیره‌سازی وجود دارد.

۲- امکان دستیابی ترتیبی و مستقیم به رکوردها (نه فیلدها) وجود دارد.

۳- پردازش در محیط‌های بلادرنگ(Real Time) و برخط(On Line) بسته به نوع سیستم عامل میتواند انجام شود.

۴- ساختار فیزیکی و ساختار منطقی فایلها از یکدیگر جدا هستند ولی نه تا حدی که برنامه های کاربردی از محیط فیزیکی ذخیره‌سازی مستقل شوند.

۵- تغییر در رسانه‌های ذخیره‌سازی بر روی برنامه‌های کاربردی تاثیر چندان ندارد.

۶- هنوز امکان بازیابی براساس چندین کلید وجود ندارد.

۷- ایمنی و حفاظت داده ها مطرح بوده ولی روشهای تامین امنیت و حفاظت ابتدایی هستند.

۸- داده ها همچنان برای کاربردهای خاص طراحی و ذخیره‌سازی می‌شوند.

۹- تکرار ذخیره‌سازی هنوز در حد نسبتاً بالایی وجود دارد.

۱۰- برای پیاده‌سازی فایل با ارتباط خاصی بین انواع رکوردها (مثلاً ارتباط سلسله مراتبی) خود برنامه‌ساز باید ارتباطات را در برنامه‌اش بسازد.

ج) نسل سوم – سیستم مدیریت داده‌ها(Data Management System)

در این نسل نرم افزاری کاملتر از نرم افزار دستیابی به عنوان واسط بین برنامه های کاربردی و فایلهای محیط فیزیکی طراحی و ایجاد شد. در این نسل دریافتند که می توان برنامه های کاربردی را در قبال رشد فایلها(File Growth) مثلاً افزودن یک فیلد به یک نوع رکورد از یک فایل مصون نگاه داشت. تا قبل از این نسل برنامه های کاربردی فقط در قبال تغییرات سخت افزاری و رشد کمی فایلها (یعنی افزایش حجم داده های فایل) مصون بودند.

مشخصات کلی این نسل عبارتند از:

۱- نرم افزار نسبتاً پیچیده‌ای به نام سیستم مدیریت داده‌ها واسط بین برنامه کاربردی و محیط فیزیکی ذخیره‌سازی است.
۲- فایلهای منطقی متعددی می‌توانند از داده های فیزیکی مشترک بهره‌برداری کنند و این فایلها می‌توانند به هم مرتبط باشند.
۳- میزان تکرار ذخیره‌سازی کاهش یافته است.
۴- داده‌های مشترک در کاربردهای متنوع به کار می‌روند.
۵- صحت داده‌های ذخیره‌شده تا حدی تامین می‌شود.
۶- نشانی‌دهی به داده ها در سطح فیلد یا گروهی از فیلدها امکان‌پذیر است.
۷- تسهیلاتی برای پردازش فایلها پیش‌بینی شده است.
۸- بازیابی به کمک چند کلید(Multi Key Retrieval) امکان‌پذیر است.
۹- ترکیبی از انواع ساختارهای فایل به کار گرفته می‌شود.

د) نسل چهارم – سیستم مدیریت پایگاه داده(Data Base Management System)

این نسل از اواخر دهه ۶۰ آغاز شد و هم‌اکنون نیز ادامه دارد. مهمترین خصیصه این نسل مستقل‌شدن برنامه‌های کاربردی(Application Program) از جنبه‌ها و خصوصیات محیط فیزیکی ذخیره‌سازی است که اصطلاحاً به آن استقلال داده‌یی فیزیکی(Physical Data Independence) می‌گویند. در ادامه این بحث، ویژگیهایDBMS را شرح می‌دهیم.

در اینجا فقط بعضی از ویژگی‌های اصلی آن را به صورت خلاصه بیان می‌کنیم.

۱- کاربران در یک محیط انتزاعی(Abstractive) و مبتنی بر یک ساختار داده‌یی تجریدی کار می‌کنند. بدین ترتیب برنامه‌های کاربردی از داده‌های محیط فیزیکی کاملاً مستقل می‌شوند. اصطلاح داده‌افزار(Data Ware) نیز ناظر به همین جنبه انتزاعی می‌باشد.

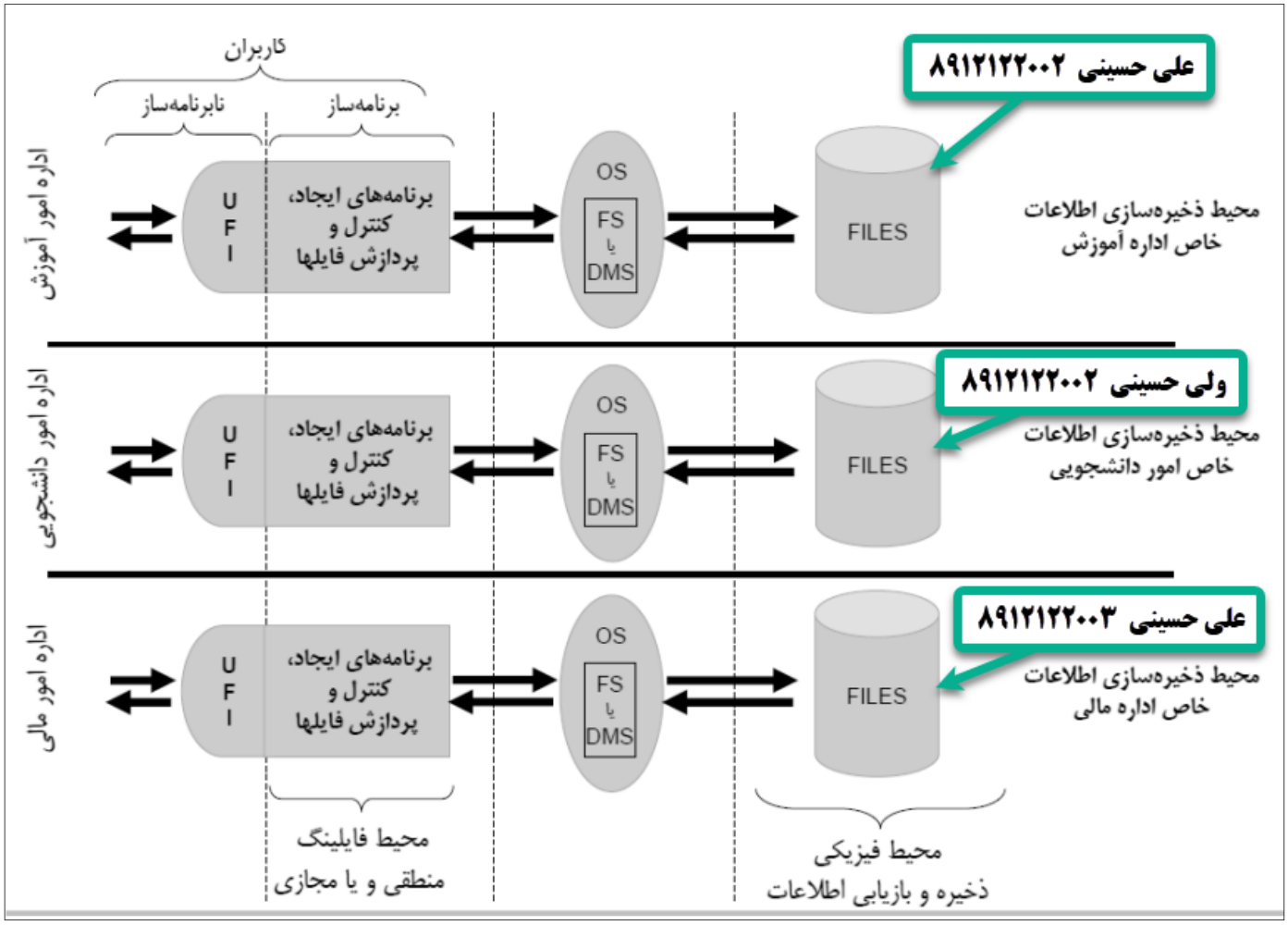
۲- امکان کنترل متمرکز روی تمام داده های عملیاتی وجود دارد و یکی از مزایای این کنترل متمرکز کاهش میزان افزونگی(Redundancy) در ذخیره‌سازی داده‌هاست.

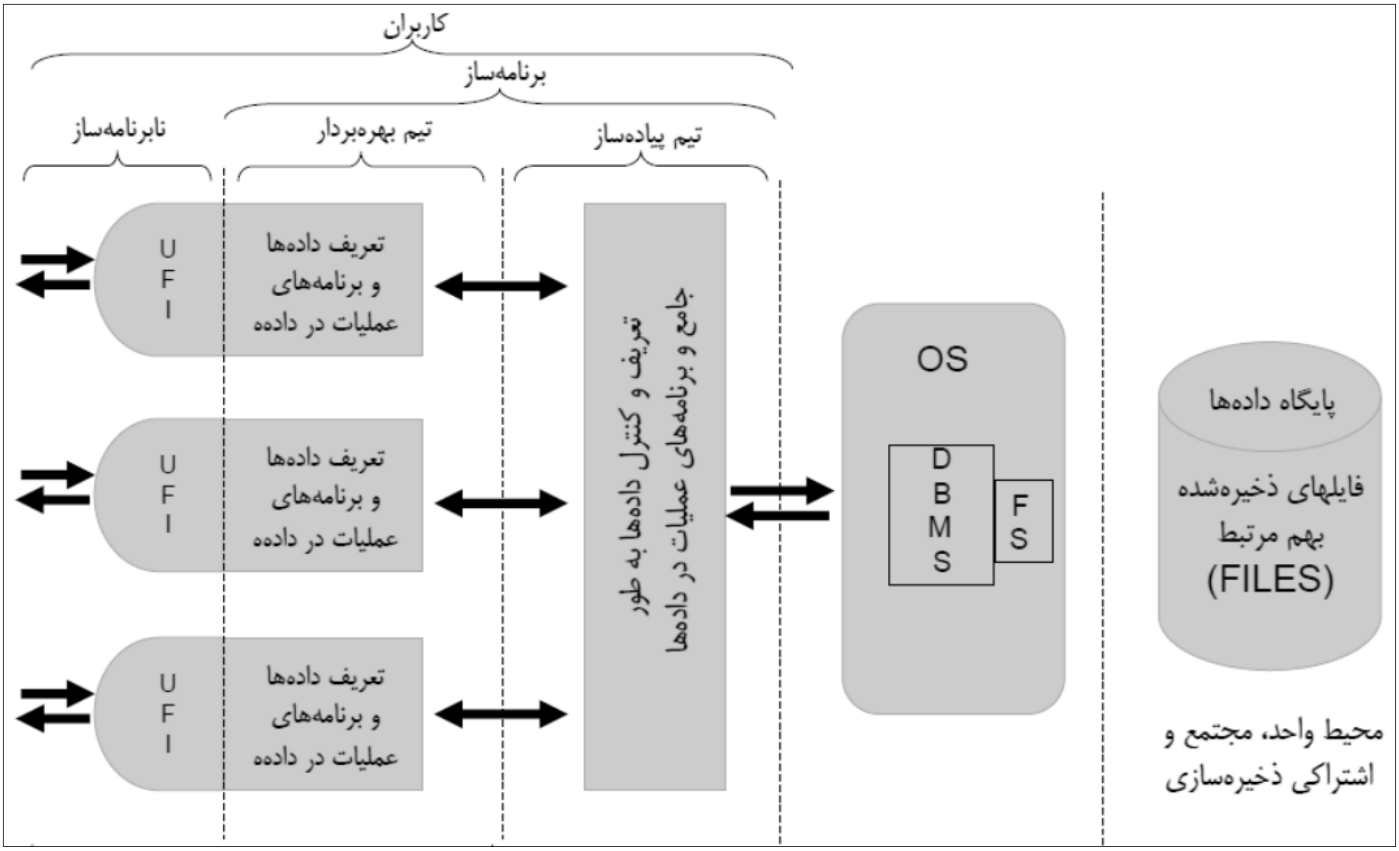
۳- نرم افزار پیچیده و جامع موسوم به سیستم مدیریت بانک اطلاعاتی(DBMS)، واسط بین برنامه های کاربران و محیط داخلی و فیزیکی ذخیره‌سازی است. این نرم افزار امکان می‌دهد تا کاربران در یک محیط انتزاعی کار کنند و در عین حال به داده های ذخیره شده دستیابی داشته و عملیات موردنظر خود را انجام دهند.

۴- سرعت دستیابی به داده ها بالا می‌باشد. ایمنی داده ها زیاد است و امکان استفاده اشتراکی از داده ها وجود دارد.

۵- در این نسل مفهوم چند سطحی بودن ساختار داده‌یی و معماری چند سطحی ذخیره‌سازی مطرح و به تدریج بسط یافت. این سطوح از پائین به بالا عبارتند از: ساختار فیزیکی بانک، ساختار داخلی بانک، ساختار ادراکی بانک و ساختار خارجی بانک

۶- رشدپذیری یکی دیگر از ویژگی‌های این سیستم است. این رشدپذیری به خاطر وجود معماری چند سطحی و نیز استقلال برنامه های کاربردی از ساختار ذخیره‌سازی و استراتژی دستیابی تامین می‌شود. در واقع باید گفت اصلی‌ترین تفاوت این نسل با نسلهای قبلی وجود حصاری نفوذناپذیر به نام سیستم مدیریت بانک اطلاعاتی یا DBMS است که هرگونه دستیابی به داده ها می‌بایست از طریق آن انجام شود.





ه) نسل پنجم – بانک معرفت(Knowledge Base) یا پایگاه شناخت(پایگاه دانش)

در این تکنولوژی، ضمن استفاده از تکنولوژی بانکهای اطلاعاتی با بهره گیری از منطق صوری(Formal Logic)، سیستمهای خبره(Expert System)، مفاهیم هوش مصنوعی(AI = Artificial Intelligence) و پردازش زبان طبیعی(Natural Language) سیستمی طراحی و ایجاد می‌شود که قادر به استنتاج منطقی از داده‌های ذخیره‌شده است.

تعریف Knowledge Base: مجموعه‌ای از واقعیتهای ساده و قواعد عام که نشان‌دهنده بخشی از جهان واقعی(Real World) باشد. منظور از جهان واقعی محیطی است عینی که در رابطه با آن می‌خواهیم اطلاعاتی داشته باشیم و گاه به آن محیط عملیاتی نیز گفته میشود. تفاوت اساسی بین بانکهای اطلاعاتی و بانکهای معرفت در این است که بانک معرفت حاوی مجموعه‌ای است از واقعیت‌های ساده و قواعد عام که به طور صریح بیان شده باشند، همراه با تعداد نسبتاً کمی از قواعد عام که به طور ضمنی بیان می‌شوند. از خصوصیات اساسی این نسل وجود سیستمی به نام «سیستم بانک معرفت» یا KBS مخفف Knowledge Base System است که خود مجموعه‌ای است از امکانات نرم افزاری و سخت‌افزاری که مسئولیت ذخیره‌سازی معرفت، تامین امنیت(Security) و جامعیت(Integrity) بانک و نیز تامین نیازهای کاربران را بر عهده دارد.

اضافه کردن بانک دانش(Knowledge Base) و قدرت استنتاج منطقی به بانک، با نامهای مختلفی مثل بانک اطلاعات پویا، بانک اطلاعات خبره، بانک اطلاعات استنتاجی(Deductive)، بانک اطلاعات بازگشتی(Recursive)، بانک اطلاعات استنباطی(Referential) و غیره مطرح می‌شود.

مثلاً در بانک اطلاعات دانشگاه:

مدرک آقای امیری دکتری است

تخصص آقای امیری شبکه‌های کامپیوتری است

آقای امیری می‌تواند درس شبکه‌های کامپیوتری را تدریس کند

بنابراین با استفاده از این روش، اطلاعات جدیدی مرتباً کشف شده و بر داده‌های بانک افزوده می‌گردد. همچنین میتوان بسیاری از اطلاعات را ذخیره نکرد و در موقع لزوم آنها را استنتاج نمود.

تذکر: نوع جدید بانکهای اطلاعاتی، بانکهای اطلاعاتی شیءگرا(OODB مخفف Object Oriented Data Base) می‌باشد.

۴-۱- داده های حجیم

کلان داده یا Big Data به معنای میزان عظیمی از داده‌های ساختاربندی شده و نشده است که این پتانسیل را دارد که به شرکت‌ها کمک کند تا عملیات‌های خود را، بهبود بخشیده و تصمیمات سریعتر و هوشمندانه‌تری اتخاذ نمایند. تعداد این داده ها به قدری است که پردازش آنها به وسیله دیتابیس‌های سنتی و نرم افزارهای موجود، دشوار است.

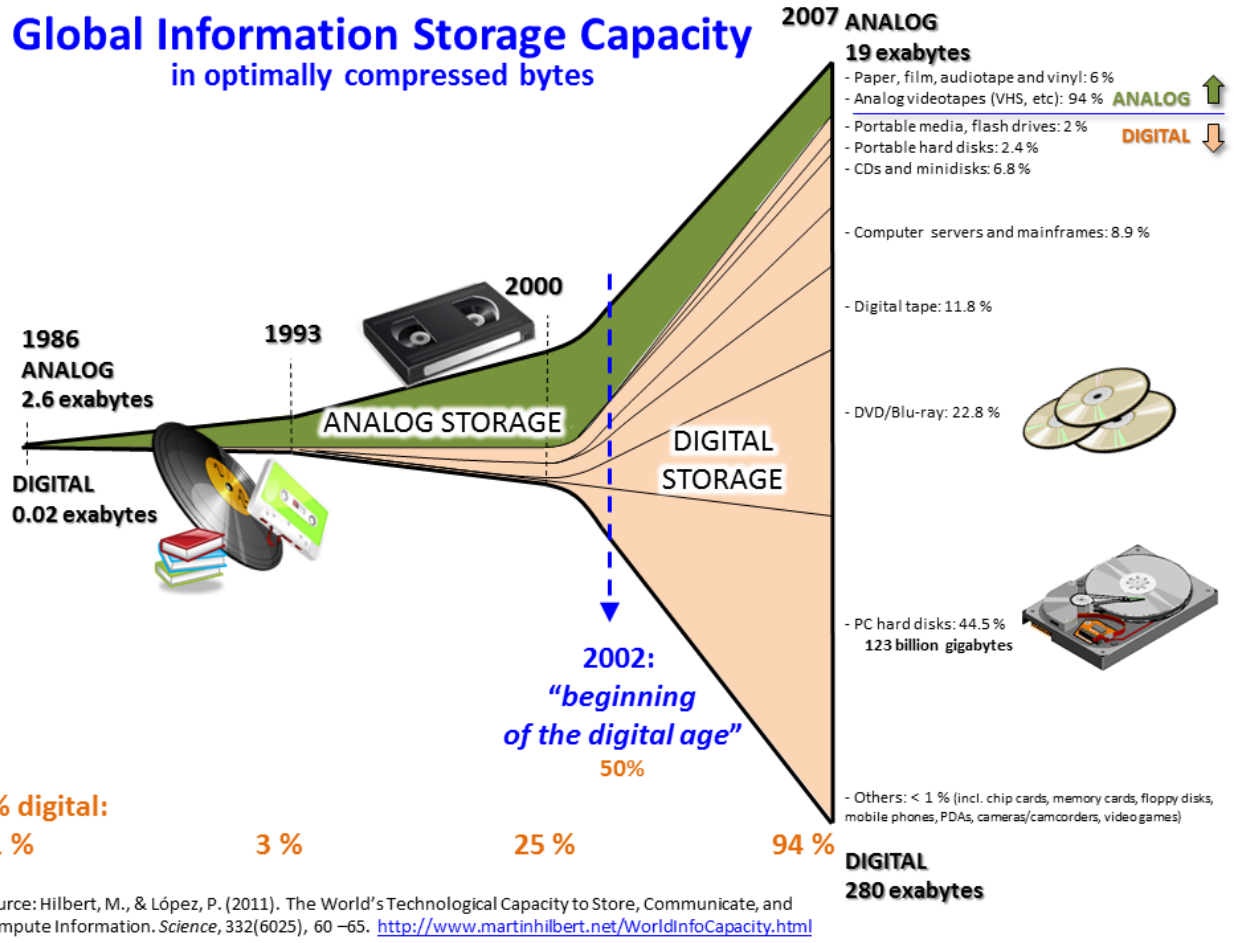
در اکثر سازمان‌ها میزان داده‌ها خیلی بزرگ است یا با سرعت زیادی رشد می‌کند و ظرفیت پردازش فعلی سازمان‌ها را، رد کرده است. در حال حاضر تمام کسب و کارهای بزرگ داخل و خارج از ایران با این مفهوم درگیر هستند.

بیگ دیتا با سه ویژگی اصلی شناخته می‌شود: حجم، سرعت، تنوع

این ویژگی‌ها تنها مربوط به داده‌های بیگ دیتا نیست. بلکه به فناوری ذخیره‌سازی و پردازش این اطلاعات نیز اطلاق می‌شود. این فناوری شامل ابزارها و فرایندهایی است که می‌بایست داده‌های بی ساختار خیلی بزرگ را، فراخوانی نمایند. حجم داده‌ها در بررسی بیگ دیتا دارای اهمیت است زیرا داده‌های بیگ دیتا شامل انواع داده‌های اشتباه، پردازش نشده، صحیح، پردازش شده و... خواهند بود. سرعت دریافت اطلاعات با توجه به همزمانی استفاده از اینترنت و ذخیره داده‌ها، بسیار بالا است. تنوع داده‌های بیگ دیتا بسیار بالاست، زیرا شامل بازه‌ی بزرگی از انواع داده مانند صدا، تصویر، متن، فیلم و... هستند.

این داده ها از تراکنشهای online، Email ها، ویدئوها، صوتها، کلیک کردن‌ها ، log ها و ارسال‌ها، درخواست‌های جستجو، یادداشت‌های درست، تعاملات شبکه های اجتماعی، داده های علمی، سنسورها و تلفنهای همراه و برنامه های کاربردی آنها و... تولید می‌شوند. آنها بر روی پایگاه داده ها که به شکل حجیم رشد می‌کنند ، ذخیره می‌شوند و ضبط، شکل دهی، ذخیره سازی، مدیریت، به اشتراک گذاری، تحلیل و نمایش آنها از طریق ابزارهای نوعی نرم افزار پایگاه داده ها، دشوار می‌شود.

پنج اگزا بایت (۱۰ به توان ۱۸ بایت) دیتا تا سال ۲۰۰۳ به وسیله انسان به وجود آمده است. امروزه این مقدار اطلاعات در دو روز تولید میشود. در سال ۲۰۱۲ داده های دنیای دیجیتال به ۲,۷۲ زتا بایت(۱۰ به توان ۲۱ بایت) توسعه پیدا کرد. این مقدار هر دو سال، دو برابر شود و به حدود ۸ زتا بایت داده در سال ۲۰۱۵ رسید.



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html

۲– تعریف پایگاه داده‌ها

پایگاه داده را می‌توان یکی از انواع سیستم‌های ذخیره و بازیابی اطلاعات، محسوب کرد. در این نوع سیستم‌ها به کاربر اجازه داده می‌شود داده‌های موردنظر خود را ذخیره نموده و ضمن انجام پردازش روی آنها، عملیات بازیابی، یعنی استفاده از اطلاعات ذخیره شده را انجام دهد.

پایگاه داده:

- مجموعه‌ای از داده‌های ذخیره‌شده و پایا،
- به‌صورت مجتمع(یک‌پارچه) (نه لزوما فیزیکی، بلکه حداقل به طور منطقی)،
- به هم مرتبط،
- با کمترین افزونگی،
- تحت مدیریت یک سیستم کنترل متمرکز،
- مورد استفاده یک یا چند کاربر از یک یا بیش از یک سیستم کاربردی،
- به طور همزمان
- و اشتراکی.

با توجه به مشکلات سیستم فایلینگ می‌توان پایگاه داده را به این صورت تعریف کرد: مجموعه‌ای سازمان‌یافته و بدون افزونگی از داده‌های مرتبط با هم که در چارچوب یک مدل داده‌ای و تحت کنترل یک سیستم متمرکز، قرار دارند.

مزایای روش پایگاهی

- ✓ کنترل داده های اضافی(Control of data redundancy)
- ✓ پایداری داده ها(Data consistency)
- ✓ استخراج اطلاعات بیشتر از داده ها.
- ✓ اشتراک داده ها(Sharing of data)
- ✓ بهبود یکپارچگی داده ها(Improved data integrity)
- ✓ بهبود امنیت داده ها(Improved security)
- ✓ اعمال استانداردها(Enforcement of standards)

معایب روش های پایگاهی:

- ⊗ پیچیدگی(Complexity)
- ⊗ اندازه(Size)
- ⊗ قیمت نرم افزار(Cost of DBMS)
- ⊗ هزینه های تبدیل(Cost of conversion)
- ⊗ تاثیر ناگوارتر خرابی سیستم (Higher impact of a failure)

۱-۲- عناصر تخصصی پایگاه داده:

سخت افزار، نرم افزار، کاربر، داده ها

۲-۱-۱- ویژگی‌های سخت افزار:

- سخت‌افزار ذخیره‌سازی یا رسانه‌های ذخیره‌سازی داده‌ها که معمولا از دیسک‌های سریع با ظرفیت بالا استفاده می‌شود.
- سخت‌افزار پردازشی یعنی همان کامپیوتر یا ماشین. ماشین‌های خاصی برای محیط‌های بانک اطلاعاتی نیز طراحی و تولید شده‌اند. این ماشین‌ها از نظر معماری، حافظه اصلی و سایر اجزا از ویژگی‌های خاصی برخوردارند.
- سخت‌افزار ارتباطی
  - جهت برقراری ارتباط‌های راه دور و نزدیک
  - معماری متمرکز، مشتری خدمتگزار، توزیع شده، موازی، چند پایگاهی، موبایل و ...

۲-۱-۲- انواع نرم‌افزار:

- نرم‌افزار کاربردی، نرم افزاری است که کاربر(برنامه ساز) باید برای تماس با سیستم بانک اطلاعاتی آماده کند.
- نرم‌افزار سیستمی، بین بانک اطلاعاتی فیزیکی که داده ها بصورت فیزیکی در آن ذخیره می‌شوند و کاربران سیستم، لایه‌ای از نرم‌افزار موسوم به مدیر بانک اطلاعاتی قرار دارد. سیستم مدیریت بانک اطلاعاتی نرم‌افزاری است که به کاربران امکان می‌دهد که پایگاه را از دید خود تعریف کنند، به پایگاه خود دستیابی داشته باشند، با پایگاه خود کار کنند و روی آن کنترل داشته باشند.

۲-۱-۳- انواع کاربر:

- مدیر داده ها(Data Administrator)، شخصی است که کنترل مرکزی داده ها را در سازمان به عهده دارد. این فرد لازم است مفهوم داده ها را درک کند و نیاز موسسه به داده ها را در سطح مدیریت عالی قرار دهد. مدیر داده ها تصمیم میگیرد که چه داده هایی از همان اول در بانک اطلاعاتی قرار گیرد و پس از ذخیره آنها، سیاستهایی را برای دستیابی به آنها تنظیم کند. توجه کنید که مسئول داده ها یک مدیر است نه یک نفر فنی.
- مدیر بانک اطلاعاتی(DBA)، که یک فرد حرفه‌ای در تکنولوژی اطلاعات(IT) است، وظایفی مثل مراقبت و نگهداری و تهیه نسخه پشتیبان، تعریف کاربران، تعریف سطوح دسترسی، تعریف ساختارهای داده، تعریف قیود داده و مسئولیت طراحی و تصمیم‌گیری برای کلیه موارد یک سیستم بانک اطلاعاتی را دارد.
- کاربر برنامه نویس(DBP)، که تصمیمات مدیر را پیاده‌سازی می‌کند. این افراد مسئول ساختن برنامه‌هایی هستند که از یک طرف به بانک اطلاعات متصل است و از طرف دیگر به کاربر نهایی یا همان اپراتور.
- کاربران نهایی، که از طریق پایانه‌ها به سیستم دسترسی دارند و از طریق برنامه‌های تهیه‌شده، داده‌ها را در حیطه نظارت DBMS دستکاری می‌نمایند.

تنها کسانی که می‌توانند دور از چشم DBMS به داده ها دسترسی داشته باشند، مدیر و برنامه‌سازان مجاز بانک اطلاعاتی هستند.

۲-۱-۴- ویژگی‌های داده:

أ) داده(Data): ANSI دو تعریف برای داده ارائه کرده است:

- نمایش واقعیات(Facts)، پدیده‌ها، مفاهیم یا معلومات به صورتی صوری و مناسب برای برقرای ارتباط، تفسیر یا پردازش توسط انسان یا امکانات خودکار.
- هر نمایشی، اعم از کاراکتری یا کمیت‌های قیاسی(آنالوگ)، که به آن معنایی منتسب است و یا باید منتسب شود و به طور کلی ما عملیاتی را روی داده یا اقلام داده‌یی انجام می‌دهیم تا در مورد یک موجودیت اطلاعاتی تهیه کنیم.

✓ از نظر ساختاری، داده عبارت است از مقادیر صفات خاصه انواع موجودیتها.

✓ بوده‌های خام که معنای اندکی دارند، مگر اینکه به صورت منطقی سازمان‌دهی شده باشند.

✓ داده ارزش‌های واقعی هستند که از طریق مشاهده و تحقیق بدست می‌آیند و به عبارت دیگر داده نمودی از وقایع، معلومات، رخدادها، پدیده‌ها و مفاهیم می‌باشند.

ب) اطلاع (Information): ANSI اطلاع را اینگونه تعریف کرده است:

- معنایی که انسان به داده منتسب می‌کند، از طریق قراردادهای شناخته شده‌ای که در نمایش داده به کار رفته‌اند.

✓ میتوان گفت از پردازش داده‌ها، اطلاعات حاصل میشود و یا داده پس از آنکه مورد تفسیر قرار گرفت تبدیل به اطلاع میشود.

✓ داده سازمان یافته‌ای که شناختی را منتقل می‌کند.

✓ داده‌ای که توسط یک فرد یا سازمان برای تصمیم‌گیری به کار می‌رود.

تذکر: در بعضی از کتابها کلمات «اطلاعات» و «داده‌ها» به جای هم به کار می‌روند. بعضی از نویسندگان، داده‌ها را همان چیزهایی می‌دانند که در بانک اطلاعاتی ذخیره می‌شوند و « اطلاعات» را به معنای آن داده‌ها از دید کاربر می‌دانند.

ج) داده‌های عملیاتی مورد نیاز کاربر(Operational Data): داده‌هایی است که کاربر بطور روزانه با آنها سر و کار دارد.

د) داده‌های پایدار(مانا): داده های بانک اطلاعاتی، داده های پایدار و با ثبات هستند. منظور از پایداری این است که نوع داده های بانک اطلاعاتی با داده های ناپایداری مثل داده های ورودی، داده های خروجی، دستورات کنترلی، صف‌ها، بلوکهای کنترل نرم‌افزار و نتایج میانی که ماهیت آنها گذرا است، تفاوت دارد. به این دلیل می‌گوئیم داده‌های بانک اطلاعاتی پایدار است که وقتی داده ها توسط سیستم مدیریت بانک اطلاعاتی برای ورود به بانک پذیرفته شد، فقط در صورتی می‌تواند حذف شود که درخواستی به سیستم مدیریت بانک اطلاعاتی ارسال شود و با اثرات جانبی ناشی از اجرای برنامه حذف نخواهد شد. با توجه به این پایداری، تعریف دقیق‌تر بانک اطلاعاتی به صورت زیر خواهد بود: بانک اطلاعاتی مجموعه‌ای از داده‌های پایدار است که توسط سیستم‌های کاربردی موجود در موسسه‌ای مورد استفاده قرار می‌گیرد. مثلاً در موسسه دانشگاه داده‌های مربوط به دانشجویان ذخیره می‌شود.

- تذکر: در کتاب‌های قدیمی به جای واژه «داده‌های پایدار» از «داده‌های عملیاتی» استفاده می‌شد.
- سرمایه‌های یک سازمان: داده‌ها(داده افزار)، نرم‌افزار، سخت‌افزار، امکانات مالی، نیروهای تخصصی.

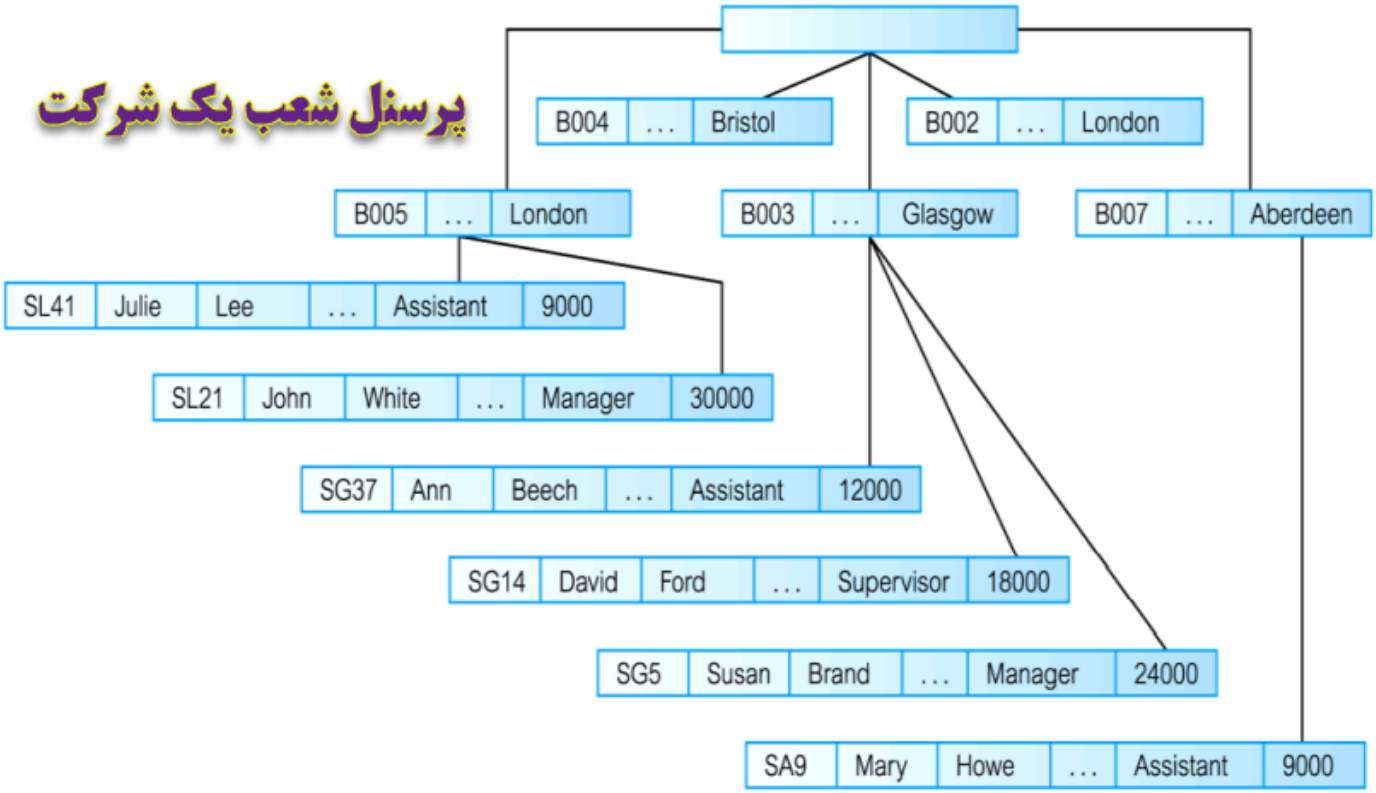
۲-۲- انواع مدل‌های پایگاه داده و کاربردهای آن‌ها:

سلسله مراتبی(Hierarchical) و شبکه‌ای(Network)

و رابطه‌ای(Relational) و رابطه‌ای شیءگرا(Object Relational) و شیءگرا(Object-Oriented)

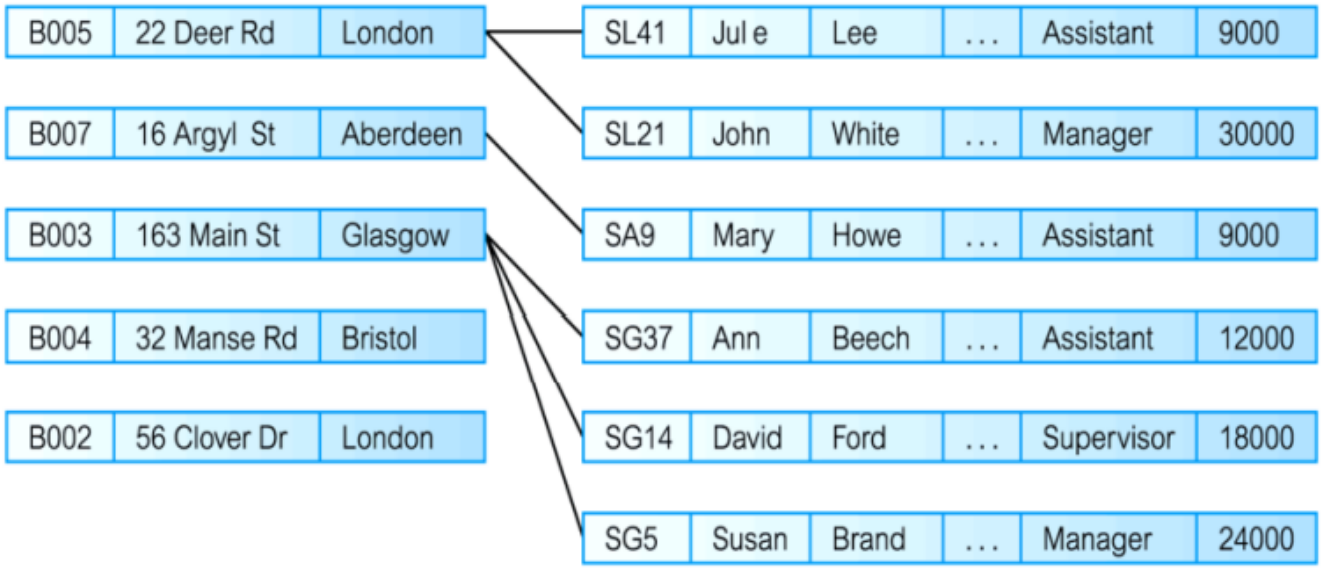
الف) مدل سلسله مراتبی(Hierarchical):

قدیمی ترین مدل برای طراحی پایگاه داده، مدل سلسله مراتبی(hierarchical) است، که در اوایل دهه ۶۰ توسط IBM برای سازماندهی دنیای تجارت به شکل سلسله مراتبی پیشنهاد شد. این مدل اجازه تکرار اطلاعات را توسط ارتباطات والد/فرزند می‌دهد؛ یعنی هر گره بعنوان والد در هر سطح، می‌تواند تعدادی گره وابسته یا فرزند داشته باشد. هر گره فرزند تنها دارای یک گره والد است. چون داده به صورت یک درختواره سازماندهی می شود برای داده هایی که ماهیت سلسله مراتبی دارند مناسب است. ساختار درختی انعطاف پذیر نیست. ارتباط تنها به صورت "تعلق دارد" یا "شامل می شود" کد می‌شوند.



ب) مدل داده شبکه ای(Network):

در مقایسه با مدل سلسله مراتبی که ساختمان‌های داده‌ای به صورت درختی از رکوردها سازماندهی می‌شود و هر رکورد آن یک والد و چند فرزند دارد، مدل شبکه اجازه رکوردهائی با چند والد و چند فرزند را می‌دهد. عملیات ذخیره و بازیابی پیچیده‌تر از مدل سلسله مراتبی است. مدل شبکه انعطاف پذیری بیشتری نسبت به سلسله مراتبی دارد.



ج) مدل رابطه‌ای(Relational DBMS):

در این مدل، داده‌ها بصورت رکوردهای مرتبط سازماندهی می‌شوند و بانک اطلاعات بصورت مجموعه‌ای از رابطه‌ها طراحی می‌شود.

Branch

branchNo	street	city	postCode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

علل موفقیت مدل رابطه‌ای عبارتند از:

۱- سادگی

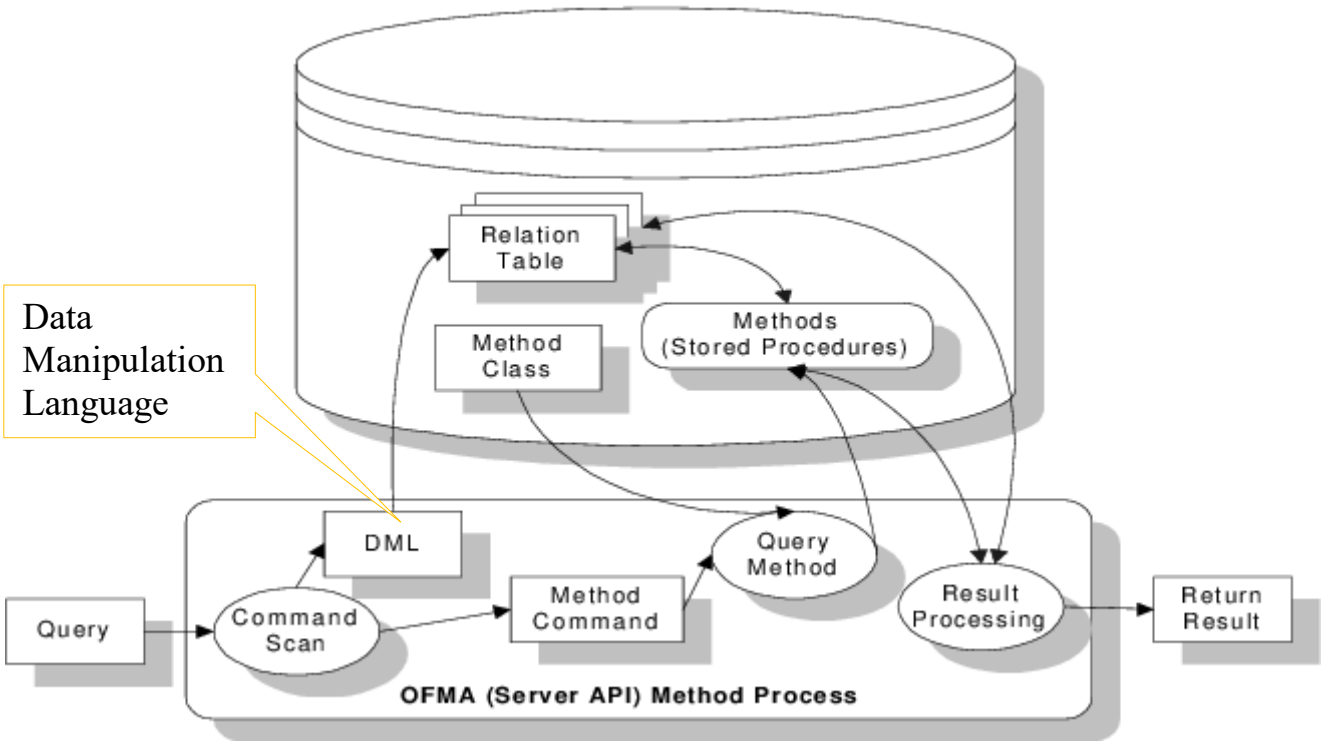
۲- پشتوانه‌ی تئوریک قوی

Staff

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

د) مدل شیء–رابطه‌ای (Object-Relational DBMS)

مانند پایگاه داده رابطه‌ای کار می‌کند ولی همراه با امکانات شیءگرایی. در این نوع پایگاه داده، می‌توان با استفاده از روابط بین داده‌ها به راحتی نسبت به جمع‌آوری سوابق مرتبط با یک شیء اقدام نمود. ولی در RDBMS سنتی، برای جمع‌آوری اطلاعات نیاز به JOIN کردن است.

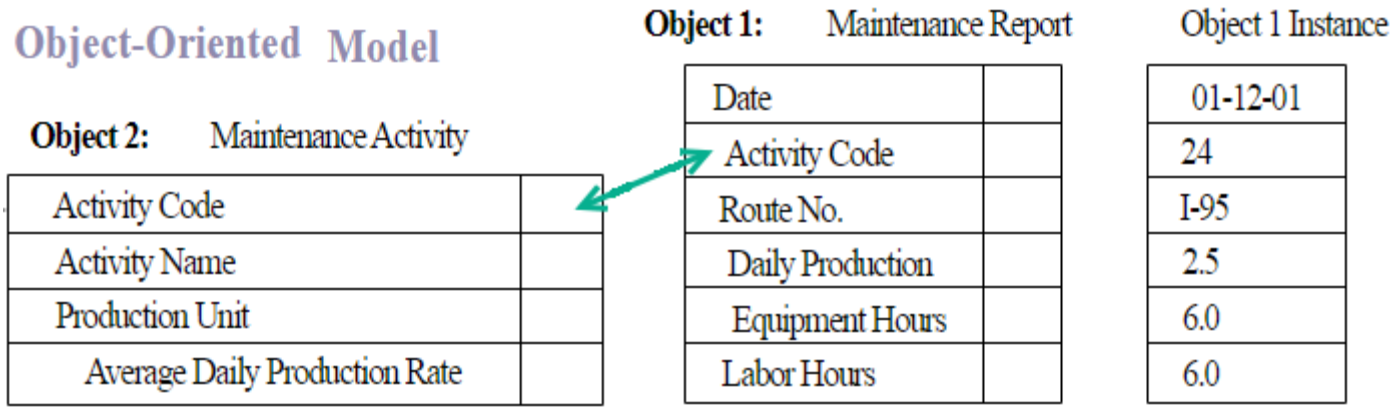


ه) مدل شیءگرا(Object-Oriented DBMS)

ساختار برنامه نویسی شیء گرا (object oriented) را می‌توان به طور مستقیم و بدون تبدیل نمودن به سایر فرمت‌ها، در پایگاه داده‌ها مورد استفاده قرار داد. این وضعیت به دلیل مفاهیم مالکیت(ownership) در سیستم چند بعدی، رخ می‌دهد. در برنامه شیء گرا (OO)، یک شیء خاص "مالک" سایر اشیاء در حافظه است، مثلا علی مالک نشانی خود می‌باشد. در صورتی که مفهوم مالکیت در پایگاه داده رابطه‌ای وجود ندارد. مدلسازی و ساخت اطلاعات بصورت اشیاء را پشتیبانی می‌کند. مواردی چون کلاس، زیرکلاس، وراثت و متد و ... .

مزایای OODBMS:

- طراح می‌تواند ساختار اشیا و رفتار آنها را مشخص کند(متدها)
- تعامل بهتر با زبان‌های شی‌گرا مانند Java و C++
- تعریف انواع داده‌ها و داده‌های تعریف شده توسط کاربر(User-defined)
- کپسوله‌سازی عملیات و متدهای تعریف شده توسط کاربر



۳– معماری پایگاه داده‌ها

یکی از مباحث مهم و کلیدی در پایگاه داده، اجزا تشکیل دهنده و پیکربندی یا طرز ترکیب اجزا سیستم و چگونگی تعامل اجزا با یکدیگر است. در معماری پایگاه داده ها حداقل یک پایگاه داده، یک سیستم مدیریت پایگاه داده، یک سیستم عامل، یک کامپیوتر با دستگاه های جانبی و تعدادی برنامه کاربردی و کاربر وجود دارند.

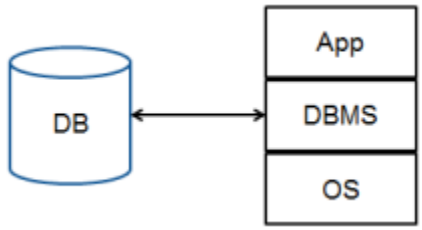
معماری پایگاه داده بستگی به دو عنصر اصلی سخت افزار و نرم افزار مدیریت پایگاه داده دارد، البته عوامل دیگری هم در این طراحی دخالت دارند از جمله:

موقعیت جغرافیایی کاربران، نیازهای کاربران، حجم داده ها، موقعیت مکانی داده ها

در اساس دو نوع معماری برای سیستم پایگاهی وجود دارد:

۳-۱- معماری متمرکز:

در این معماری یک پایگاه داده ها روی یک سیستم کامپیوتری و بدون ارتباط با سیستم کامپیوتری دیگر، ایجاد می‌شود. برای کاربردهای کوچک مناسب است.

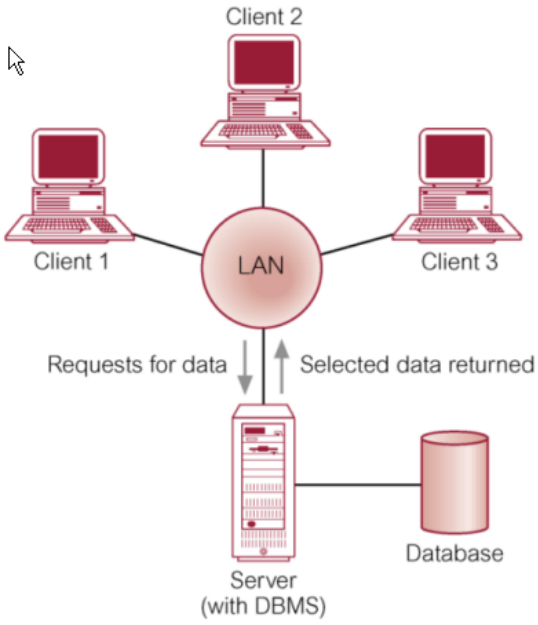


زمانی می‌گوییم سیستم متمرکز است که روی یک سیستم باشد.

۳-۲- معماری نامتمرکز:

خود دارای انواع مختلفی می باشد:

- معماری مشتری- خدمتگزار(Client-Server)
- معماری توزیع شده
- معماری چند پایگاهی
- معماری با پردازش موازی
- و ..



۳-۲-۱- معماری مشتری- خدمتگزار(Client-Server):

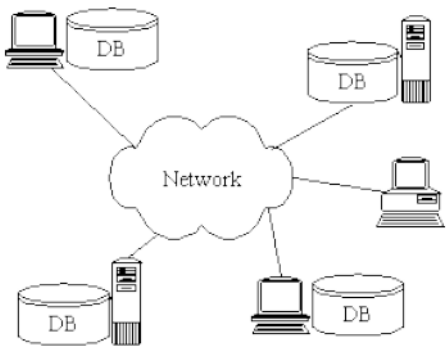
هر معماری که در آن قسمتی از پردازش را یک برنامه، سیستم یا ماشین انجام دهد و انجام قسمت دیگر از پردازش را از برنامه دیگر بخواهد، معماری مشتری خدمتگزار خواهد بود.

در واقع وظایف سیستم به دو دسته تقسیم خواهد شد:

۱.دسته ای که انجام آن بر عهده سرور خواهد بود.

۲ .دسته ای که انجام آن توسط مشتری خواهد بود.

۳-۲-۲- معماری توزیع شده:

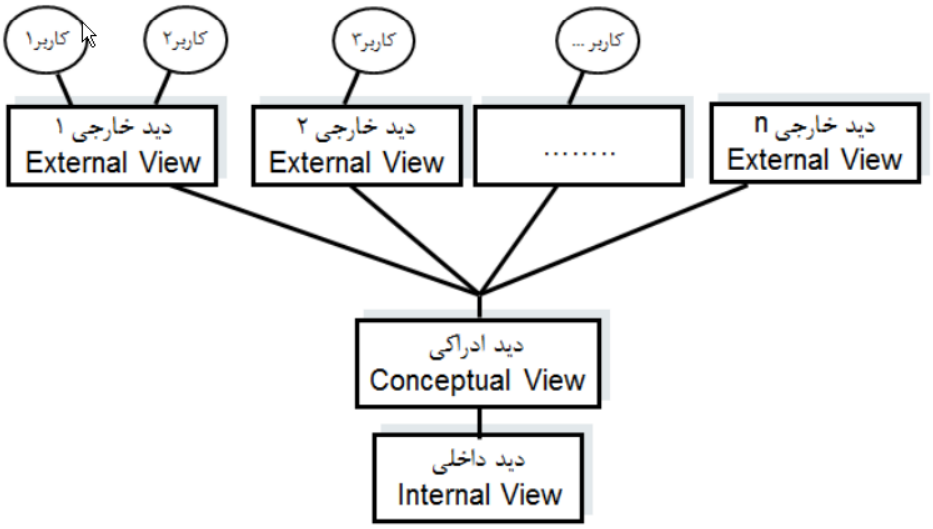


مجموعه ای از داده های ذخیره شده، که منطقا به یک سیستم تعلق دارند، ولی در کامپیوترهای مختلف که در یک یا بیش از یک شبکه توزیع شده اند، قرار گرفته‌اند. می‌توان گفت در این معماری تعدادی پایگاه داده‌ی ذخیره شده روی کامپیوترهای مختلف داریم که از نظر کاربران، پایگاه واحدی هستند. به بیان دیگر، این معماری مجموعه ای است از چند پایگاه داده منطقا یکپارچه(مجتمع)، ولی به طور فیزیکی توزیع شده روی یک شبکه کامپیوتری.

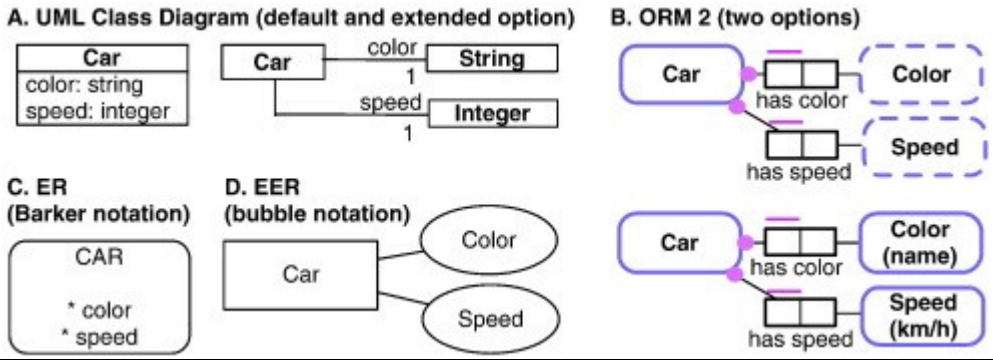
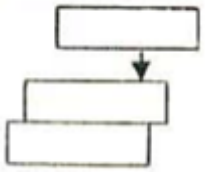
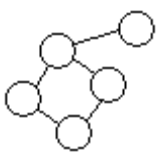
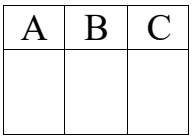
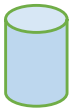
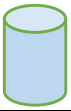
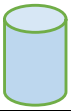
۳-۳- معماری ANSI/SPARC

سیستم های مدیریت پایگاه داده دارای معماری‌های یکسانی نیستند. معماری سه سطحیANSI/SPARC یکی از استانداردهایی است که امروزه اساس اکثر سیستم های مدیریت پایگاه داده را شکل می دهد. هدف معماری سه سطحی این

است که امکاناتی را فراهم کند تا کاربران بتوانند با دیدگاه‌های شخصی خود به داده مورد نیاز دسترسی پیدا کنند. یعنی هر کاربری بتواند به داده مشترک توسط یک زبان دسترسی پیدا کند اما دید خاص خود را داشته باشد.



از طرف دیگر فاصله بین سطح داخلی از سطح خارجی دلالت بر این دارد که کاربر نیازی به دانستن جزئیات فیزکی داده ذخیره شده در پایگاه داده ندارد. این تفکیک سطح، اجازه تغییر ساختار ذخیره سازی پایگاه داده را بدون تاثیر روی دیدهای کاربران می دهد. لازمه این کار ، استقلال سطوح از همدیگر است به نحوی که تغییرات روی یک سطح روی بقیه تاثیر نگذارد.

تصویر خارجی	کاربر ۱ ..... کاربر ۲ ..... کاربر n	دیدهای کاربران مختلف (Views)
تصویر ادراکی عام	نمودارهای ER و EER و UML و ORM و ... 	کل بانک بدون توجه به مدل خاص
تصویر ادراکی خاص	مدل سلسله مراتبی  مدل شبکه‌ای  مدل رابطه‌ای  مدل شیءگرا	کل بانک در قالب مدل انتخابی
تصویر داخلی یا فیزیکی	 .....  	کل بانک روی رسانه‌ها

اهداف معماری سه سطحی:

- هر کاربر باید بتواند به داده‌های بانک اطلاعاتی دسترسی داشته باشد. با این توجه که هر کاربر دید خاصی از داده‌ها دارد.

- دید هر کاربر باید مصون از دید کاربران دیگر باشد.
- نیازی نیست که کاربر از جزئیات ذخیره سازی فیزیکی پایگاه داده مطلع باشد.
- کاربر نباید بتواند با جزئیات ذخیره سازی داده سروکار داشته باشد.
- مدیر پایگاه داده(DBA) باید بتواند ساختار ذخیره سازی پایگاه داده را، بدون تاثیر بر دید کاربران، تغییر دهد.
- مدیر پایگاه داده باید بتواند ساختار مفهومی پایگاه داده را، بدون تاثیر بر کاربران، تغییر دهد
- تغییرات در جنبه های فیزیکی ذخیره سازی نباید روی ساختار داخلی پایگاه داده تاثیر بگذارد.

طبق مدل پیشنهادی ANSI (ارائه شده در سال ۱۹۷۵) معماری سیستم بانک اطلاعاتی از اجزای زیر تشکیل شده است:

الف) دید داخلی یا فیزیکی(Internal View)

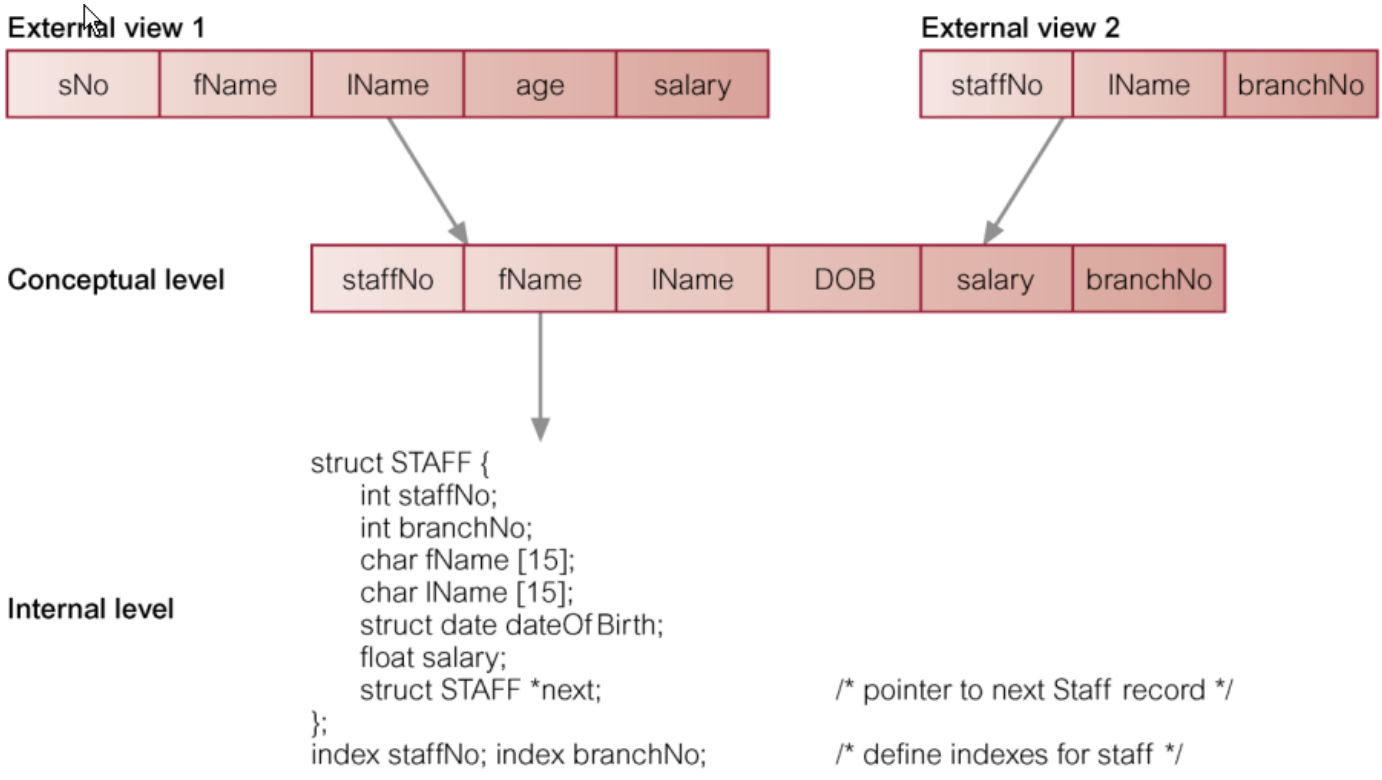
ب) دید ادراکی یا مفهومی(Conceptual View)

ج) دید خارجی(External View)

د) تبدیلات بین سطوح(Transformation یا Mappings)

ه) زبان میزبان(HL یا Host Language)

و) زبان فرعی داده‌یی(DSL یا Data Sub Language)



الف) دید داخلی

سطح داخلی درگیر چگونگی نمایش فیزیکی پایگاه داده روی سیستم کامپیوتری است و شرح می دهد چگونه داده واقعا در پایگاه داده و سخت افزار ذخیره می شود.

سطح داخلی، دید طراح پایگاه داده از محیط فیزیکی ذخیره سازی و درواقع فایل های محیط فیزیکی است که توسط شمای داخلی(internal schema) توصیف می‌شود. شمای داخلی نحوه نمایش فیزیکی داده هایی را که در شمای ادراکی شرح داده شده را مشخص می کند. انواع مختلف رکوردها، فیلدهای داده، فایل ها، نحوه نمایش رکوردها در فایل، استراتژی دستیابی، شاخص‌ها و تخصیص فضای ذخیره‌سازی برای داده‌ها، محل رکورد، فشردگی داده‌ای و تکنیک‌های رمزگذاری داده‌ها و چگونگی ترتیب رکوردها در فایل توسط شمای داخلی تشریح می شوند.

جزئیات تبدیل به منبع ذخیره سازی در معماری سه سطحی بیان نمی شود و از این سطح به پائین در اختیار DBMS نیست و به عهده سیستم عامل و درایورهای دستگاه ذخیره سازی است.

- سطح داخلی نزدیکترین سطح به رسانه ذخیره‌سازی فیزیکی است.
- مباحث این سطح مربوط به درس ذخیره و بازیابی اطلاعات است

ب) دید ادراکی (مفهومی):

دید طراح بانک از داده های ذخیره در بانک است. دید طراح، دیدی است جامع دیدهای همه کاربران و در عین حال متفاوت با هر یک از دیدها. در این سطح ساختار داده‌ها و ارتباط موجودیتها و صفات خاصه، امنیت و جامعیت داده‌ها و اطلاعات معنایی داده ها مطرح می‌گردد.

داده های دنیای واقعی آنطور که واقعا هستند توسط طراح پایگاده داده مدل می شوند. برای تعریف سطح ادراکی از یک ساختار یا مدل داده استفاده می شود که شمای ادراکی(conceptual schema) نامیده می شود. علاوه بر این رویه های شناسائی و قیدهای جامعیت را نیز در بر می‌گیرد. برای کسب استقلال داده، شمای ادراکی تنها درگیر معنی داده است و جنبه های نمایش داده، سازماندهی فیزیکی و استراتژی های دستیابی، نادیده گرفته می شود.

شمای خارجی از شمای ادراکی مشتق می شود و اگر مدل داده در هر دو سطح یکسان نباشد، سیستم پایگاه داده را دوساختاری می نامند.

مثال: در محیط عملیاتی دانشگاه، موجودیت‌های درس و دانشجو را در نظر می‌گیریم. فرض می‌کنیم ساختار بانک به صورت جدولی باشد. طراح برای هر نوع موجودیت یک جدول طراحی می‌کند و برای نمایش ارتباط بین موجودیت‌ها نیز یک جدول دیگر. مثلاً به صورت زیر:

جدول دانشجو (STTAB)			
شماره دانشجویی	نام	نام خانوادگی	سال ورود

جدول درس (COTAB)			
شماره درس	نام درس	تعداد واحد	ماهیت درس

جدول دانشجو-درس (STCOTAB)			
شماره دانشجو	شماره درس	ترم	نمره ترم

ج) دید خارجی

دید کاربر از داده های ذخیره شده در پایگاه داده است. منظور از دید کاربر(user view) قسمتی از پایگاه داده است که کاربر با آن سروکار دارد. یعنی مجموعه ای از صفات خاصه موجودیت هائی است که در اختیار کاربر قرار داده می شود. هر کاربر دیدگاه های خاص خود را از پایگاه داده می تواند داشته باشد.

دید هر کاربر باید تعریف شود. به تعریف و شرح دید کاربر شمای خارجی(external schema) می‌گویند. برای تعریف شمای خارجی از یک مدل داده استفاده می شود که معمولا همان است که در سطح ادراکی بکار رفته است.

مثال: کاربر A1 میتواند دید زیر را روی جدول STTAB داشته باشد:

جدول دانشجو (STTAB1)	
شماره دانشجویی	سال ورود

یا کاربر B2 دید زیر را میتواند روی جدول STCOTAB داشته باشد:

جدول دانشجو-درس (STCOTAB1)	
شماره دانشجویی	شماره درس

- در دید خارجی می‌توان شرط هایی قرار داد تا سطرهایی (یا ستونهایی) از جدول مبنا در دید کاربر قرار گیرد.
- تنها لایه‌ای که کاربران با آن سر و کار دارند لایه خارجی است. لایه‌های دیگر به مدیر و برنامه سازان بانک مربوط می‌شود.
- طبق قانون «پنهان سازی اطلاعات» که می‌گوید «به هر کس به همان اندازه اطلاعات بده که نیاز دارد و نه بیشتر»؛ در لایه خارجی دیدهای مختلف کاربران مطرح است تا هر کدام بخشی از بانک را که نیاز دارند ببینند.
- سطح خارجی نزدیکترین سطح به کاربران است.

د) تبدیلات بین سطوح

در شکل استاندارد ANSI دو تبدیل وجود دارد: تبدیل ادراکی/داخلی و تبدیل خارجی/ادراکی.

عمل تبدیل را با توجه به آنچه که تبدیل می‌یابد میتوان به سه دسته زیر تقسیم کرد.

۱- **تبدیل داده‌ها:** یعنی تبدیل داده‌های تعریف شده در سطح خارجی به داده های تعریف شده سطح ادراکی و بالاخره به داده های تعریف شده در سطح داخلی و نیز مسیر برعکس.
۲- **تبدیل احکام:** یعنی تبدیل حکم عمل کننده در سطح خارجی به حکم عمل کننده در سطح ادراکی و بالاخره به حکم یا احکامی در سطح داخلی
۳- **تبدیل ساختار:** یعنی تبدیل ساختار سطح خارجی به ساختار سطح ادراکی. مثلاً اگر ساختار داده یی در سطح ادراکی سلسله مراتبی و در سطح خارجی جدولی باشد میبایست تبدیل ساختار سلسله مراتبی به جدولی

و برعکس را داشته باشیم به این سیستم ها دو ساختاری میگویند. البته اغلب سیستم های موجود یک ساختاری هستند.

تبدیلات بین سطوح:

تبدیل ادراکی/ داخلی: مثلاً اگر طراح بانک تعدادی جدول را طراحی کرده باشد، در تبدیل ادراکی به داخلی، برای هر جدول می‌توان فایلی تعریف کرد بصورتی که هر سطر جدول رکوردی از این فایل باشد. تغییرات در سطح داخلی بانک، ممکن است همیشه بروز کند. اینگونه تغییرات نباید در دید ادارکی تاثیر داشته باشد. در تبدیل ادراکی/داخلی از سیستم عامل نیز کمک گرفته میشود.

تبدیل خارجی/ادراکی: در واقع این تبدیل مکانیسمی برای برقراری تناظر بین دیدهای خارجی مختلف و دید واحد ادراکی است. یک دید مشخص از یک کاربر خاص، بخشی است از دید واحد ادراکی و از نظر انواع موجودیتها، صفات خاصه هر موجودیت، نوع صفت و غیره لزوماً همان نیست که در دید ادراکی از نظر طراح وجود دارد. مثلاً ممکن است یک صفت خاصه از یک موجودیت، از دید یک کاربر ترکیبی از چند صفت خاصه از سطح ادراکی باشد. همچنین یک کاربر میتواند چندین دید داشته باشد. تبدیل خارجی/ادراکی مسائل فوق را حل میکند.

ه) زبان میزبان(HL) و زبان فرعی داده‌یی (Data Sub Language)

منظور از زبان میزبان یکی از زبانهای سطح بالای برنامه‌سازی مثل پایتون، Go، C++، C#، Java، Visual Basic ، Visual C، Rust، می‌باشد.

زبان DSL زبانی سطح بالاتر است، که میهمان یک زبان سطح بالا مثل Visual C می‌شود. هر مدل داده‌یی خاص (مثل سلسله مراتبی، شبکه‌ای، رابطه‌ای) زبان فرعی خاص خود را دارد.

تعداد احکام این زبانها معمولاً کم است.

برای هر سطح از معماری دستوراتی وجود دارد موسوم به:

- زبان فرعی داده‌یی خارجی
- زبان فرعی داده‌یی ادراکی
- زبان فرعی داده‌یی داخلی

احکام زبان DSL را میتوان به سه دسته زیر تقسیم کرد:

۱- احکام تعریف داده ها (Data Definition Language = DDL)
- ایجاد، حذف یا تغییر در جدول

۲- احکام کار با (پردازش) داده ها (Data Manipulation Language = DML)
- درج، حذف و ویرایش رکوردها

۳- احکام کنترلی (Data Control Language = DCL)
- دادن یا گرفتن مجوز انجام کاری خاص(GRANT, REVOKE)
- امکان تعیین نوع استراتژی های دستیابی، تعریف شاخص ها

به طور کلی دو دسته زبان داده‌یی وجود دارد یکی زبان داده یی نامستقل یا ادغام شده (Embedded)و دیگری زبان داده یی مستقل. در نوع نامستقل DSL حتماً باید میهمان یک زبان سطح بالا باشد مثل SQL که در دلفی یا ویژوال سی استفاده میشود یا Btrieve که زبان فرعی داده ای برای C یا پاسکال است. در نوع مستقل DSL نیازی به زبان میزبان ندارد مثل Access و FoxPro

البته SQL هم به صورت مستقل و هم به صورت نامستقل وجود دارد.

چند نکته در رابطه با DSL :

- هر چه جنبه انتزاعی بودن سطح خارجی و ادراکی بانک قویتر باشد، DSL منتزع از مفاهیم فایل‌پردازی خواهد بود.
- هر DSL فقط برای یک مدل داده یی مشخص طراحی میشود.
- مجموعه احکام DSL باید حداقل باشد.
- اصل وحدت عملگرها باید در DSL رعایت شود؛ یعنی برای انجام یک عمل مشخص در سطوح خارجی و ادراکی، حکم واحدی وجود داشته باشد.
- اصل وحدت عملگرها در یک سطح مشخص نیز مطرح است مثلاً هم برای درج داده ها و هم برای درج ارتباط بین آنها می‌بایست یک حکم واحد وجود داشته باشد.
- DSLباید داده های مختلف کاربران (مثل بردارها، ماتریسها، رشته ها و ...) را نیز بپذیرد.

تذکر: DSL یک زبان بیانی(Declarative) است که در آن کاربر می‌گوید چه می‌خواهد؛ ولی رویه انجام کار را بیان نمی‌کند، برعکس C و جاوا که رویه‌ای(Procedural) هستند و کاربر باید رویه انجام کار را بیان کند.

لغتنامه داده‌ها و کاتالوگ سیستم:

لغتنامه داده‌ها(Data Dictionary) شبیه لغتنامه‌های معمولی، تمامی اسامی استفاده شده در سیستم و معنای آنها را در بر می‌گیرد. در مرحله طراحی بانک اطلاعات هرگاه طراح برای مفهومی نامی انتخاب می‌کند، باید آن را در لغتنامه داده‌ها همراه با معنای آن و فرمت آن وارد کند.

این اسامی شامل تمامی نام‌های جداول، شیء‌ها، صفت‌ها و غیره است. در بانک‌های جدید نرم‌افزار ویژه‌ای برای کار با لغتنامه‌ها وجود دارد که به کمک آن میتوان اسامی را وارد یا جستجو کرد. این نرم‌افزارها از اشتباهاتی نظیر وارد کردن یک نام با دو معنای مختلف(Homonym) و یا دو نام برای یک مفهوم(Synonym) جلوگیری می‌کنند.

علاوه بر اسامی داده‌ها اطلاعات دیگری باید در مورد بانک نگهداری شود مثل:

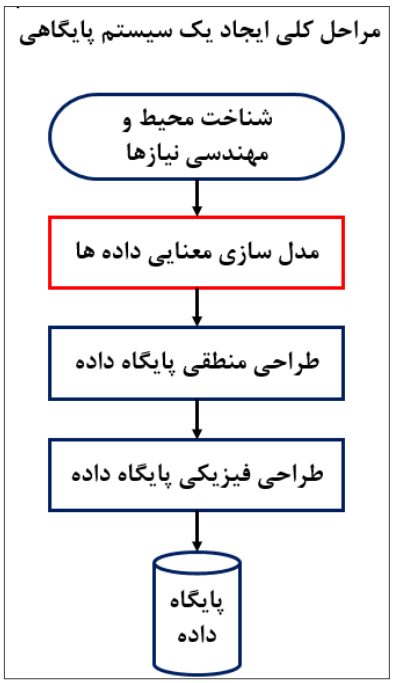
- اطلاعات مربوط به حق دستیابی افراد به داده‌های مختلف
- تاریخ ایجاد و یا تغییر داده‌ها
- تعداد نسخه‌های هر پرونده
- اندازه هر جدول یا شیء و غیره.

این‌گونه اطلاعات در کاتالوگ سیستم(System Catalog) نگهداری می‌شوند. سیستمDBMS به طور خودکار و به کمک کاربران اطلاعات موجود در کاتالوگ سیستم را همواره به روز نگه میدارد.

به اطلاعات موجود در دیکشنری داده‌ها اصطلاحاً فراداده یا دادگان(Meta Data) می‌گویند که به معنی داده در مورد داده است.

۴– انواع روش‌های مدل‌سازی داده(Data Modeling)

در طراحی یک بانک اطلاعاتی ابتدا می بایست مراحلی چون امکان سنجی، بررسی نیازها و محدودیت‌ها، بررسی سیستم دستی موجود و غیره صورت گیرد که این مراحل در درس تجزیه و تحلیل سیستم ها مطرح می‌گردد.



پس از انجام مراحل فوق، طراح بانک به کمک نمودارهایی، شمای کلی بانک را مستقل از مدل بانک(جدولی، شبکه‌ای، سلسله مراتبی) و نیز مستقل از جنبه‌های برنامه نویسی ترسیم می‌کند.

مدل سازی معنایی داده ها:

عبارتست از ارائه مدلی از محیط عملیاتی به کمک مفاهیمی مستقل از موضوعات مربوط به نمایش منطقی و فیزیکی داده‌ها. مدل سازی معنایی را در بعضی از کتب، طراحی ادراکی یا مفهومی(conceptual) می‌نامند. ارائه یک مدل کلی از پایگاه داده‌یی که می-خواهیم ایجاد کنیم، در بالاترین سطح انتزاعی و با توجه به معنایی که کاربر برای داده‌ها قائل است(یعنی وارد جزئیات نمی‌شویم).

برای این کار مدل‌های مختلفی از جمله مدل دودویی(Binary)، مدل داده‌ای معنایی(Semantic Data)، مدل ER، مدل ER گسترش یافته(EER)، مدل شیءگرا (OR) و زبان مدل‌سازی یکپارچه(UML=Unified Modeling Language) و غیره وجود دارد. ما در این مجال ER و سپس EER را که معروف‌تر از بقیه هستند، شرح می‌دهیم.

این مدل‌ها اصولاً ابزار طراحی هستند و نه پیاده‌سازی.

نمودار ER(Entity Relationship Diagram) :

اولین طرح پایگاه‌داده‌ها و مدل کلی آن در بالاترین سطح انتزاع می باشد. نموداری است که در آن سه مفهوم اساسی یعنی (موجودیت، صفت، ارتباط) نمایش داده می‌شود.

۱– موجودیت(Entity):

آنچه که می‌خواهیم از یک محیط مشخص، در موردش اطلاع داشته باشیم. مثل محیط دانشکده. موجودیت‌هایی مثل استاد، دانشجو، برنامه، کلاس، کارمند و ... .

برای تشخیص موجودیت‌ها در یک محیط باید توجه داشته باشیم که موجودیت دارای خصوصیات زیر است:

- هر نوع موجودیت یک نام دارد.
- هر نوع موجودیت یک معنا دارد.
- هر نوع موجودیت نمونه‌هایی دارد.
- هر نوع موجودیت مجموعه ای از صفات دارد.
- ارتباط‌هایی با نوع‌های دیگر دارد.
- عدم وابستگی و یا وابستگی به یک نوع دیگر

نمونه موجودیت:

تمام نمونه‌های مشخص از هر نوع موجودیت در یک محیط، مجموعه‌ای به نام نمونه‌های موجودیت را تشکیل می‌دهند.

- نمونه موجودیت‌های خودرو: پیکان، پژو، سمند، بنز و ...
- نمونه موجودیت‌های دانشجو: کریمی، احمدی، اکبری و ...
- نمونه موجودیت‌های درس: پایگاه داده، شبکه، سیستم عامل و ...

حالات یک موجودیت:

- موجودیت مستقل(قوی): موجودیتی است که مستقل از هر موجودیت دیگر و به خودی خود، در یک محیط مشخص مطرح باشد. مثل کارمند، استاد، دانشجو.
- موجودیت ضعیف یا وابسته: موجودیتی است که وجودش وابسته به یک نوع موجودیت دیگر است. در واقع هر نمونه موجود از یک موجودیت ضعیف به یکی از نمونه‌های یک موجودیت قوی وابسته است.
  - بعنوان مثال، فرض کنید کارمند و فیش حقوقی دو موجودیت در یک سیستم پرسنلی باشند. پس اطلاعات تعدادی کارمند و نیز تعدادی فیش حقوقی(بعنوان نمونه‌های دو موجودیت) در سیستم ذخیره شده‌اند. اگر یک کارمند را از سیستم حذف کنیم، وجود فیش‌های حقوقی مربوط به او دیگر مفهومی ندارد و منطقا باید حذف شوند.
  - موجودیت ضعیف از خود شناسه ندارد. بلکه یک صفت ممیزه یا جداساز(Discriminator) دارد.

۲– صفت:

هر نوع موجودیتی خصیصه‌هایی(ویژگی‌هایی) دارد که حالت یا وضع آن موجودیت را توصیف می کند. مثل موجودیت دانشجو که صفات آن عبارتند از: شماره دانشجویی، نام، رشته و ... .

- هر صفت یک نام دارد
- هر صفت یک معنا دارد.
- هر صفت دامنه(میدان) دارد که نوع و طیف مقادیر و معنای آن را مشخص می‌کند. مثل نمره که نوع آن اعشاری و مجموعه مقادیر مجاز آن از ۰ تا ۲۰

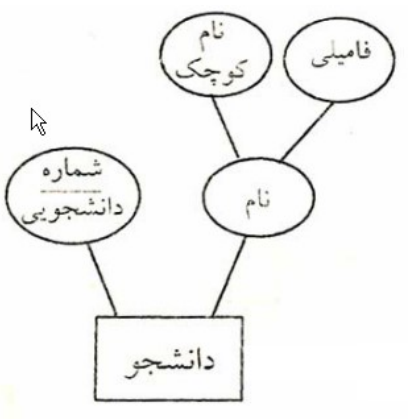
۲-۱) محدودیت‌های یا قیود یک صفت(Constraint):

- محدودیت‌های دامنه‌ای:
  - نوع و مقدار
- محدودیت‌های یکتایی مقدار(uniqueness)

- مثل کد ملی و شماره دانشجویی
- محدودیت هیچ مقدار ناپذیری(null able)
  - مثل صفتی که همیشه باید مقادیرش معلوم باشد.
- محدودیت بین صفات
  - مثل انواع وابستگی‌ها

۲-۲) صفت ساده و مرکب:

- صفت ساده: صفتی است که مقدار آن از لحاظ معنایی تجزیه شدنی نباشد، به این معنا که اگر مقدار آن را به اجزائی تجزیه کنیم، مقادیر هر جز فاقد معنا باشد.
- صفت مرکب: صفتی که از چند صفت ساده تشکیل شده است و تجزیه شدنی است. مانند آدرس که می تواند شامل نام کشور، استان، شهر، منطقه و ... باشد.
  - در بانک اطلاعات رابطه‌ای امکان تعریف صفات مرکب وجود ندارد.(توضیحات بیشتر در ادامه بحث)



۲-۳) صفت تک مقداری و چندمقداری:

- صفت تک مقداری: صفتی است که حداکثر یک مقدار از میان مقادیر را برای یک نمونه از یک نوع موجودیت می‌گیرد. به بیانی دیگر مقدار آن صفت برای نوع موجودیت مورد نظر، یک مقدار مشخص از میدان مقادیر مربوط به آن صفت می باشد.
  - مثل شماره دانشجویی، تاریخ تولد، نام پدر
- صفت چند مقداری: صفتی است که می‌تواند بیش از یک مقدار از میان مقادیر را برای یک نمونه از موجودیت در بر گیرد.
  - مثل صفت مدرک تحصیلی ممکن است برای یک شخص، کاردانی، کارشناسی و کارشناسی ارشد باشد.
  - توجه: در بانک اطلاعات رابطه‌ای امکان تعریف صفات چندمقداری وجود ندارد.

۲-۴) صفت کلیدی(شناسا):

صفتی که باید یکتایی مقدار داشته باشد و به‌عبارت دیگر مقدار این صفت بین نمونه‌های مختلف تکرار نمی‌شود.

- مثل شماره دانشجویی برای موجودیت دانشجو یک صفت کلید است.
- در مدل‌سازی معنایی، برای هر موجودیت باید یک شناسه مشخص کرد. اما یک موجودیت ممکن است بیش از یک شناسه داشته باشد. مثلا کد ملی نیز یک شناسه دیگر برای موجودیت دانشجو است. در چنین حالاتی، یکی از شناسه ها را باید به‌عنوان شناسه اول(اصلی) مشخص کرد. صفت دیگر نیز می‌تواند به عنوان شناسه دوم(فرعی) معرفی شود

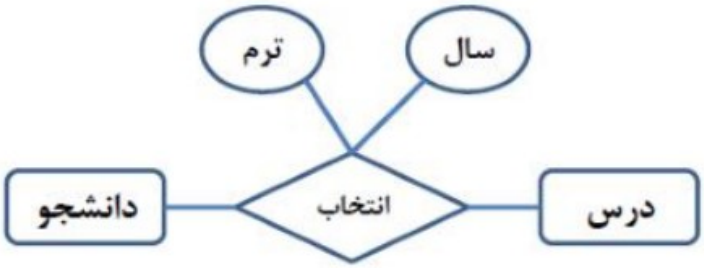
۲-۵) صفت مجازی، مشتق(Derived):

صفتی که مقدارش در پایگاه داده ذخیره نشده باشد و وجود خارجی ندارد، ولی از روی دیگر صفات قابل محاسبه(قابل اشتقاق) است. مانند سن و معدل(مرتباً با گذراندن دروس بیشتر عوض می‌شود).

۳– ارتباط(Relationship):

موجودیت‌های هر محیط عملیاتی با هم ارتباطاتی دارند. ارتباط، وابستگی بین چند موجودیت را نشان می دهد. یک ارتباط، یک وابستگی معنی‌دار بین دو یا چند نوع موجودیت مختلف است.

- ارتباطات معادل افعال یا مفاهیمی نظیر خریدکردن، تعمیرکردن، عضوبودن، رئیس یک سازمان بودن هستند.
- نکات مربوط به ارتباط:
  - هر نوع ارتباط یک نام دارد.
  - هر نوع ارتباط شرکت‌کنندگانی دارد. تعداد شرکت‌کنندگان را درجه(n) می‌گویند.
  - می‌تواند صفت‌هایی داشته باشد



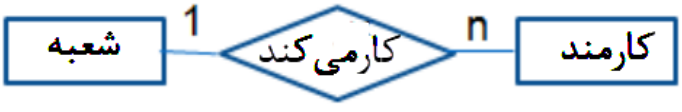
۳-۱) چندی ارتباط:

چندی ارتباط، نوع تناظر بین نمونه‌های دو موجودیت حاضر در ارتباط را نشان می‌دهد و می‌تواند(1:1) یا (1:N) یا (M:N) باشد.

- ارتباط یک به یک: هر نمونه از موجودیت اول می تواند فقط با یک نمونه از موجودیت دوم ارتباط داشته باشد. هر نمونه از موجودیت دوم نیز فقط می‌تواند با یک نمونه از موجودیت اول ارتباط داشته باشد. در واقع همان مفهوم تناظر یک به یک در نظریه مجموعه ها است
  - بعنوان مثال، اگر درجه ارتباط مدیریت بین موجودیت‌های استاد و گروه آموزشی(۱:۱) باشد، مفهوم آن به صورت شکل زیر خواهد بود.

ارتباط یک به یک

- ارتباط یک به چند: هر نمونه از موجودیت اول می‌تواند با چند نمونه از موجودیت دوم ارتباط داشته باشد. اما هر نمونه از موجودیت دوم فقط می‌تواند با یک نمونه از موجودیت اول ارتباط داشته باشد.
  - اگر درجه ارتباط بین موجودیت‌های کارمند و شعبه، یک به چند باشد، مفهوم ارتباط چنین است. هر شعبه می تواند چندین کارمند داشته باشد. اما هر کارمند می‌تواند فقط در یک شعبه کار کند.

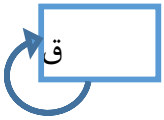


ارتباط یک به چند

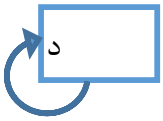
- ارتباط چند به چند: هر نمونه از موجودیت اول می‌تواند با چند نمونه از موجودیت دوم ارتباط داشته باشد. هر نمونه از موجودیت دوم نیز می‌تواند با چند نمونه از موجودیت اول ارتباط داشته باشد.
  - حال اگر درجه ارتباط بین موجودیت‌های دانشجو و درس چند به چند باشد، مفهوم ارتباط چنین است. یعنی هر دانشجو می‌تواند چند درس را بگیرد. هر درس هم می‌تواند توسط چند دانشجو گرفته شود.

ارتباط چند به چند

⊕ نکته: ممکن است یک موجودیت با خودش ارتباط داشته باشد.



این بدین معناست که «یک قطعه از قطعه یا قطعات دیگر ساخته شده است».



«یک درس پیش نیازهایی دارد».

۳-۲) درجه ارتباط:

برای هریک از موجودیت‌های شرکت کننده در یک ارتباط می‌توان یک حد مشخص کرد. این حد نشان دهنده حداقل و حداکثر تعداد نمونه های آن موجودیت است که می‌توانند در ارتباط شرکت کنند.

بعنوان مثال، حدود نشان داده شده در شکل زیر برای دو موجودیت دانشجو و درس بیانگر مفاهیم زیر هستند:

- هر دانشجو حداقل ۱ و حداکثر ۵ درس را می تواند انتخاب کند.
- هر درس می تواند توسط هیچ دانشجویی انتخاب نشود و یا حداکثر توسط ۲۰ دانشجو انتخاب شود.

ارتباط چند به چند

تعداد موجودیت‌های شرکت کننده در ارتباط	درجه	
	لاتین	فارسی
۱	Unary	یگانی
۲	Binary	دوگانی
۳	Ternary	سه گانی
...	...	...
N	n-ary	چند گانی

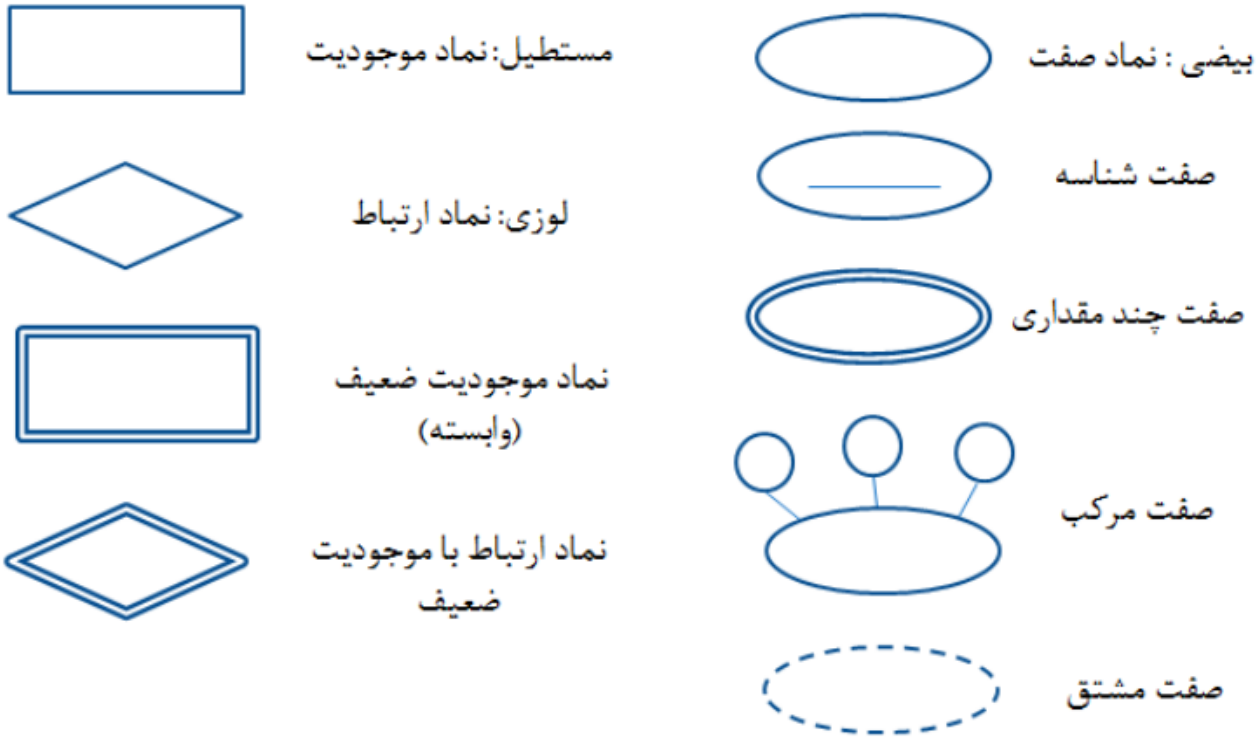
۳-۳) انواع مشارکت در ارتباط:

موجودیتهایی که در یک ارتباط شرکت دارند، نوع ارتباط‌شان میتواند اجباری یا اختیاری باشد. اگر مقدارحداقل حد یک موجودیت در یک ارتباط صفر(۰) باشد، به این معنی است که نمونه ای از این موجودیت می تواند وجود داشته باشد که اصلا در ارتباط شرکت نکند. در این حالت شرکت موجودیت را اختیاری می گوییم. در غیر اینصورت، شرکت موجودیت در ارتباط اجباری است. اگر شرکت موجودیت دانشجو در ارتباط انتخاب درس اجباری و شرکت درس در این رابطه اختیاری باشد، این دو مفهوم به این شکل در نمودار ER نشان داده می‌شوند.

مشارکت الزامی

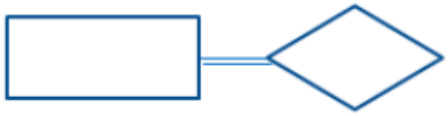
اگر در دانشگاهی قانونی وجود داشته باشد که «هر استاد حداقل باید یک درس را تدریس کند» آنگاه ارتباط استاد با گروه درسی اجباری می‌شود.

۴– نمادهای نمودار ER:



نماد مشارکت در ارتباط

نماد مشارکت الزامی در ارتباط

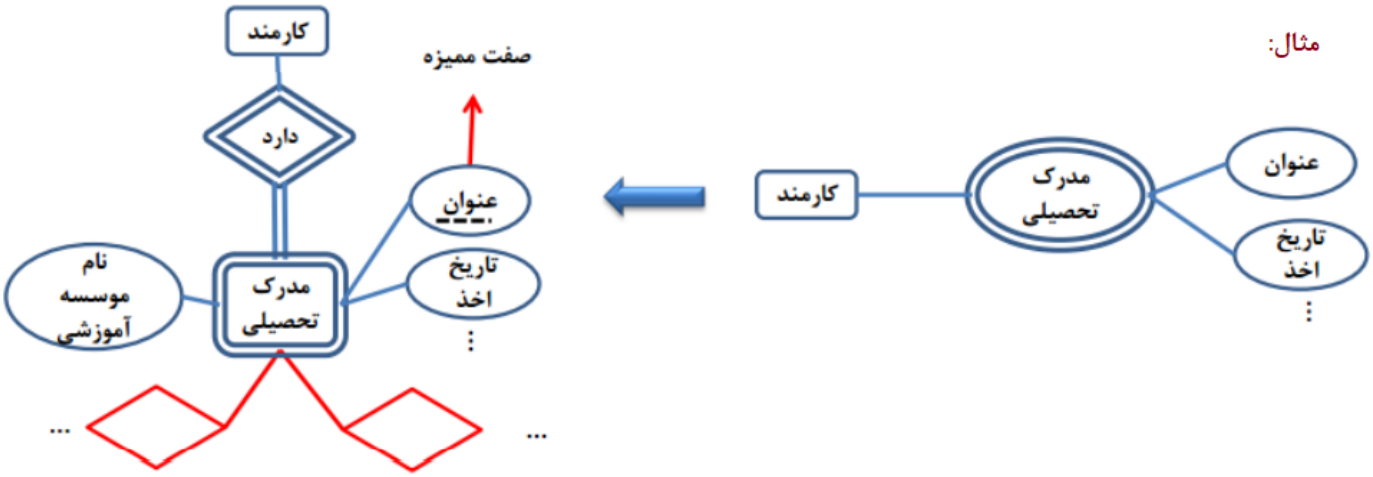


چندی ارتباط

مثال: عضو خانواده به عنوان یک موجودیت ضعیف

شماره کارمند	نام عضو خانواده
۱۰۰	گلی سلی قلی
۲۰۰	ناجی تاجی سلی

مثال:



تمرین ۱: ترسیم نمودار ER کتابخانه

می‌خواهیم در یک کتابخانه، کتبی در پایگاه داده ثبت شود؛

- اطلاعات کتب مثل شماره کتاب، سال نشر، عنوان، نویسنده یا نویسندگان، قیمت، موضوع
- اطلاعات دانشجویان شامل شماره دانشجویی، نام، آدرس
- دانشجویان می‌توانند کتاب‌هایی را به کتابخانه اهدا کنند.
- اطلاعات متصدیان کتابخانه شامل کد کارمندی، تلفن، تاریخ استخدام. معمولا اطلاعات اعضای خانواده متصدی نیز در کنار اطلاعات فوق، نگه‌داری می‌شود.
- هر دانشجویی می‌تواند یک یا چند کتاب را به امانت بگیرد؛ تاریخ تحویل و تاریخ برگشت کتاب باید نگه‌داری شود.
- عملیات ثبت‌نام دانشجو توسط متصدی انجام خواهد شد و هر متصدی موظف به ثبت چند دانشجو خواهد بود.

تمرین ۲: مدل‌سازی معنایی بانک

می‌خواهیم برای یک سیستم بانکی بسیار ساده یک پایگاه داده طراحی کنیم.

بعد از مصاحبه و بررسی و تحلیل این سیستم مشخصات آن به شرح زیر بدست آمده است:

- هر بانک شعبی دارد. هر شعبه در شهر خاص قرار دارد و با یک نام منحصر بفرد شناخته می شود. بانک دارایی های هر شعبه را زیر نظر دارد.
- مشتری های هر بانک با شماره مشتری خود شناخته می شوند. بانک برای هر مشتری نام و شهر و خیابانی که در آن زندگی می کند را ذخیره می کند. هر مشتری حساب هایی دارد و می تواند از بانک وام دریافت کند. به هر مشتری یک بانکدار اختصاص می یابد که می تواند بانکدار شخصی آن مشتری یا مسئول پرداخت وام به او باشد.
- کارمندان بانک با شماره کارمندی خود شناسایی می شوند. مدیریت بانک، اطلاعاتی همچون نام، شماره تلفن، حقوق، نام وابستگان و شماره کارمندی هر کارمند را ذخیره می کند. به علاوه برای هر کارمند بانک، تاریخ شروع به کار و مدت خدمت هر کارمند را نیز نگهداری می کند. از هر کارمند چندین شماره تلفن نگه داری می شود.

- هر بانک دو نوع حساب دارد: حساب جاری و حساب پس انداز. هر حساب می تواند چند صاحب حساب داشته باشد و هر مشتری می تواند چندین حساب داشته باشد. هر حساب شماره منحصر بفردی دارد. بانک موجودی هر حساب و تاریخ آخرین دسترسی صاحب حساب به آن را نگه می دارد. هر حساب پس انداز نرخ بهره ای دارد. برای حساب جاری میزان چک های بی محل ذخیره می شود.
- هر وام می تواند به چند مشتری داده شود ولی در یک شعبه خاص صادر می شود. هر وام یک شماره یکتا دارد. بانک مبلغ هر وام و میزان اقساط آن را نگه می دارد. شماره اقساط وام بین همه وام ها منحصربفرد نیست بلکه برای هر وام، شماره اقساط‌ش منحصربفرد است. تاریخ و مبلغ هر قسط وام نیز در بانک ثبت می شود.

تمرین ۳: نمودار ER رستوران

یک رستوران قصد تولید یک سیستم اطلاعاتی برای انجام فعالیت‌های خود دارد. بدین منظور احتیاج به استفاده از پایگاه داده برای تامین نیاز داده‌ای خود دارد. گروه تحلیلگر نیازهای محیط را اینگونه شناسایی کرده‌اند:

- رستوران شعبه‌های مختلفی دارد. شعب تهران زیر نظر مدیر شعبه‌ی اصلی اداره می‌شود. شهب شهرستان هر کدام مدیر مستقلی دارند که تحت نظر مدیر شعبه‌ی اصلی فعالیت می‌کنند.
- رستوران با تامین کنندگان مواد غذایی مختلفی قرارداد دارد که محموله های مختلفی را بر حسب نیاز، هر روز یا هر چند روز یکبار به شعب مختلف تحویل می‌دهند.
- هر شعبه چندین آشپز، پیشخدمت، کارگر و راننده دارد که با توجه به ساعات کاریشان دستمزد ماهانه می گیرند.
- غذاهای رستوران که نوع و میزان مواد غذایی هر کدام مشخص ،است با قیمتهای مشترک توسط مشتریان خریداری می شود.
- به جز مشتریان مشترک و موردی چندین سازمان با شعب مختلف رستوران قرارداد ماهیانه دارند و غذاهای مشخصی را هر روز دریافت میکنند.
- شکایات مشتریان از هر نوع(غذا، سرویس و ...) ثبت شده و در اختیار مدیریت کل رستوران قرار میگیرد و از سوی او نمره منفی به شعبه تعلق میگیرد.
- هر مشتری میتواند پس از صرف غذا امتیازی بین ۰ تا ۱۰ به غذا و سرویس رستوران بدهد.

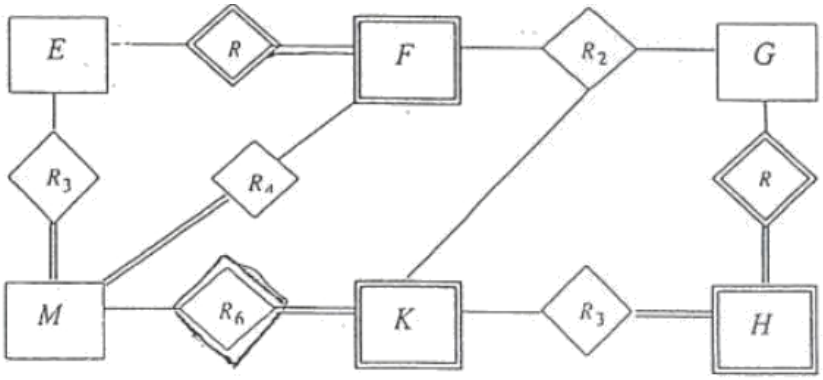
تمرین ۴: نمودار ER شرکت بیمه خودرو

یک شرکت بیمه ی خودرو قصد دارد اطلاعات مربوط به بیمه شدگان و نوع خودرو آنها را نگهداری کند:

- بیمه شده می‌تواند حقیقی یا حقوقی باشد.
- خودرو نیز انواع مختلفی مانند سواری، باری و موتور دارد.

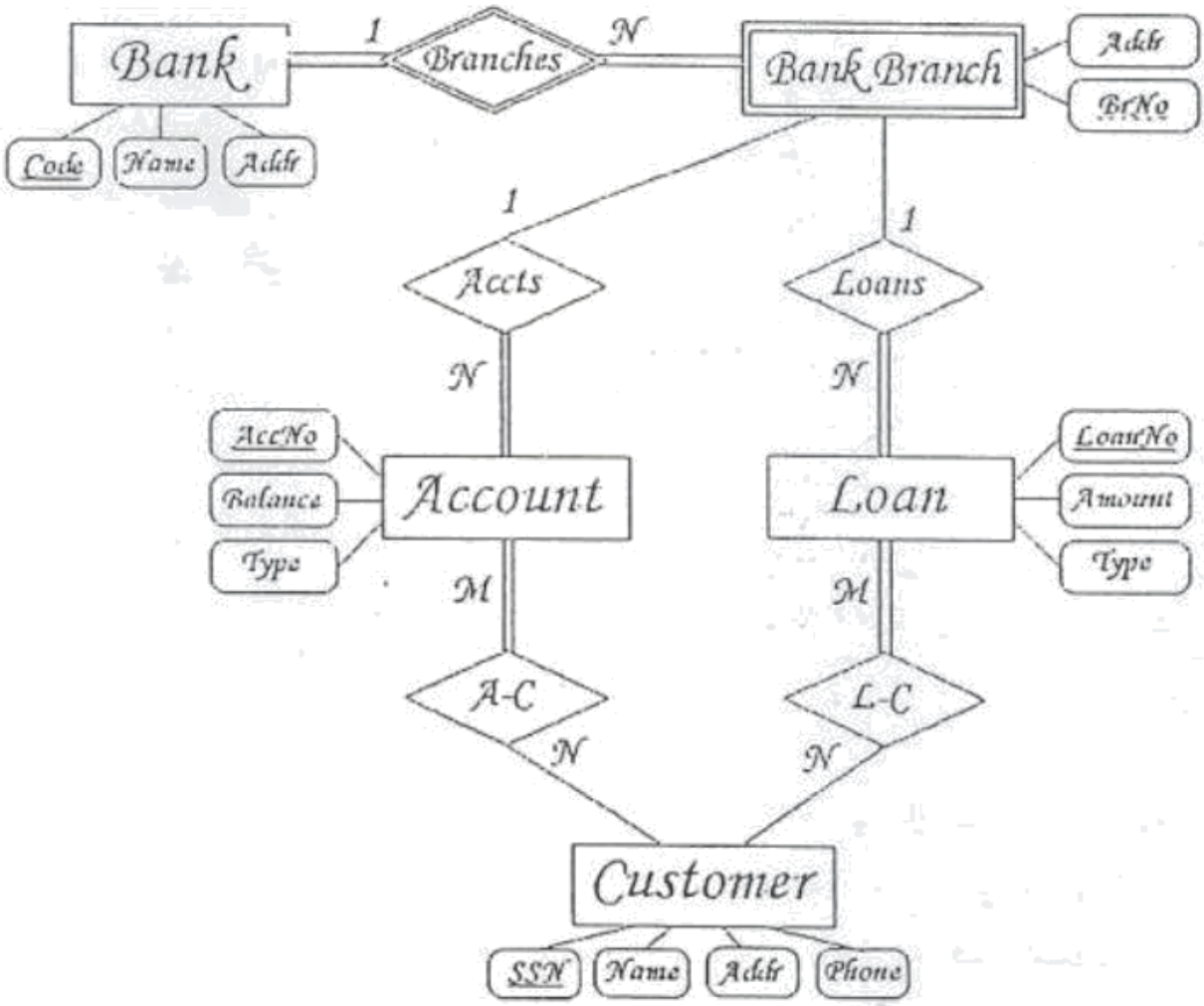
تمرین ۵:

در نمودار زیر چند نوع موجودیت ضعیف و چند نوع موجودیت قوی وجود دارد؟



تمرین ۶:

مدل سازی زیر را در نظر بگیرید: موجودیت های غیر ضعیف را نام ببرید. موجودیت های کارمندان بانک، جایزه و قرعه کشی را با ارتباطات رایج به این نمودار اضافه کنید. فرض های خود را بنویسید.

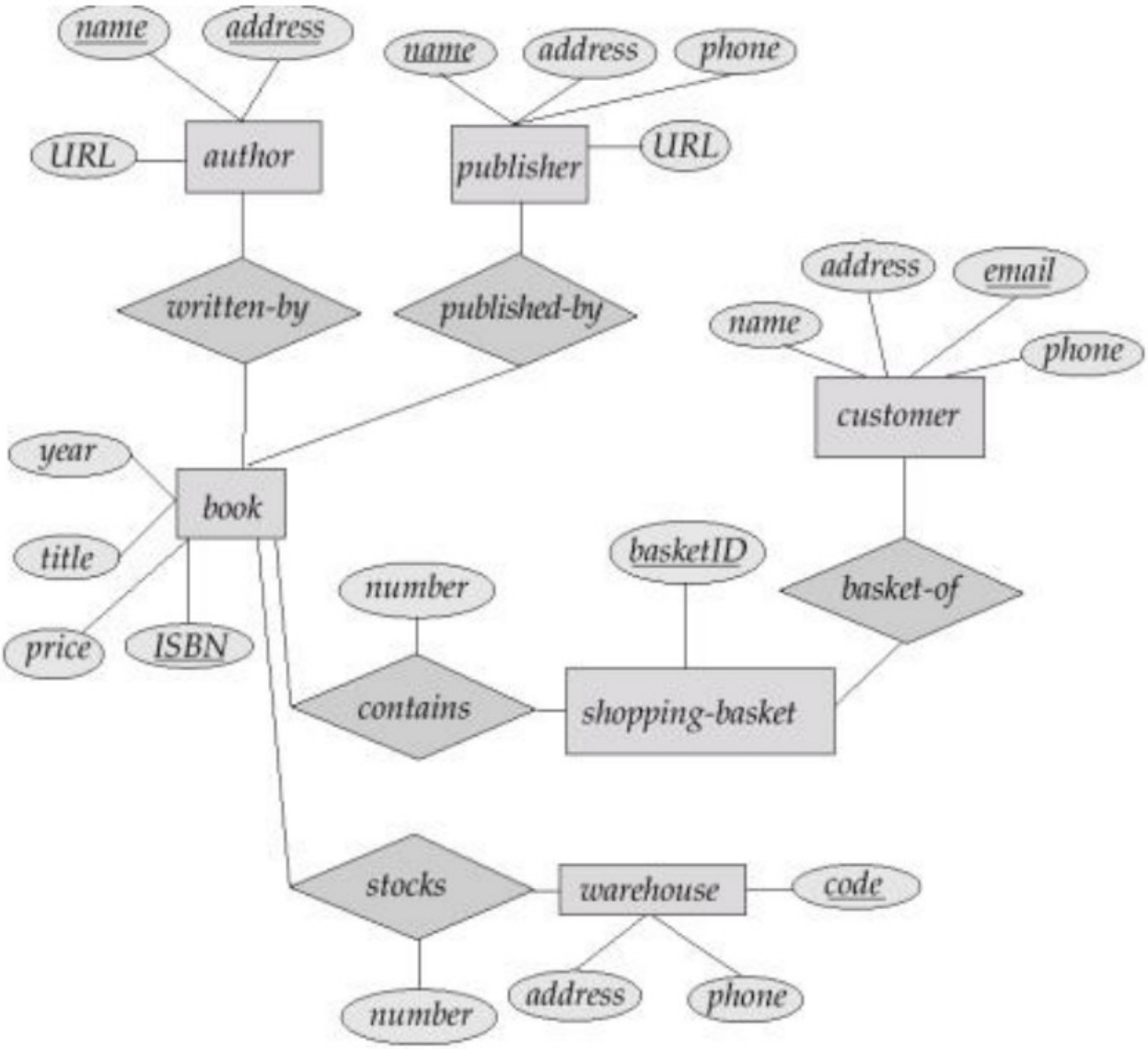


تمرین ۷:

شکل زیر را در نظر بگیرید که یک کتابفروشی online را مدل می‌کند.

۱- در نظر بگیرید که کتابفروشی می‌خواهد کاست موسیقی و دیسک فشرده را به مجموعه خود اضافه کند. موسیقی‌های یکسان ممکن است در شکل کاست ها یا دیسکهای فشرده، با قیمت های متفاوت ارائه شوند. نمودارER را برای مدل کردن این موارد اضافی تغییر دهید، از اثرات آن روی basket-shopping صرفنظر شود.

۲- حالا نمودارER را برای عمومیت دادن به منظور مدل کردن حالتی که basket-shopping ممکن است شامل هر ترکیبی از کتابها، کاستهای موسیقی، یا دیسکهای فشرده باشد، تغییر دهید.

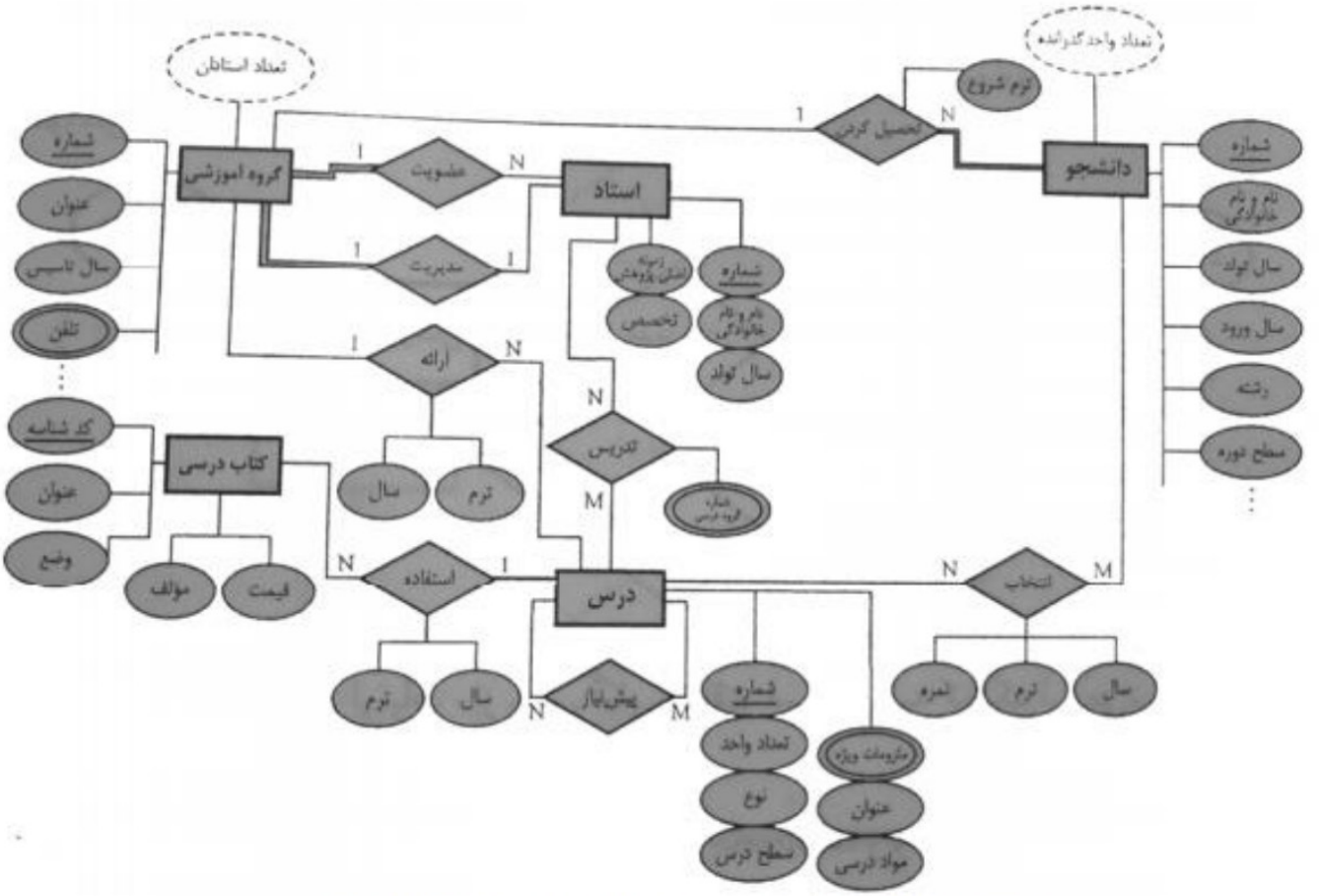


تمرین ۸:

برای یک سیستم فروشگاه با مشخصات زیر، یک نمودار ER طراحی کنید:

- هریک از اجناس فروشگاه یک کد مشخصه، نام و قیمت دارد.(توجه شود که اجناس یک نمونه، کد مشخصه یکسانی دارند)

- هر مشتری می‌تواند در هر بار خرید، از برخی اجناس تعدادی را خریداری کند. به ازای هر خرید یک فاکتور صادر می‌شود که دارای شماره و تاریخ است.
- فروشگاه تعدادی مشتری ثابت و مشخص دارد که هر مشتری دارای کد اشتراک، نام، نشانی و تلفن می‌باشد.
- جزئیات اطلاعات مشتریان هم در سیستم نگه‌داری می‌شود. مشتری می‌تواند حقیقی یا حقوقی باشد. برای مشتری حقیقی شماره، نام و نام خانوادگی و برای مشتریان حقوقی، شماره مشتری، نام و نوع شغل ثبت می‌شود. شماره مشتری برای مشتریان حقیقی و حقوقی در همه شعب یکتاست.



بخشی از نمودار ER (بسیار ساده شده) پایگاه داده آموزش دانشکده

نمودار ER گسترش یافته یا EER(Extended or Enhanced):

نمودار ER کاستی‌هایی دارد که در EER رفع شده است.

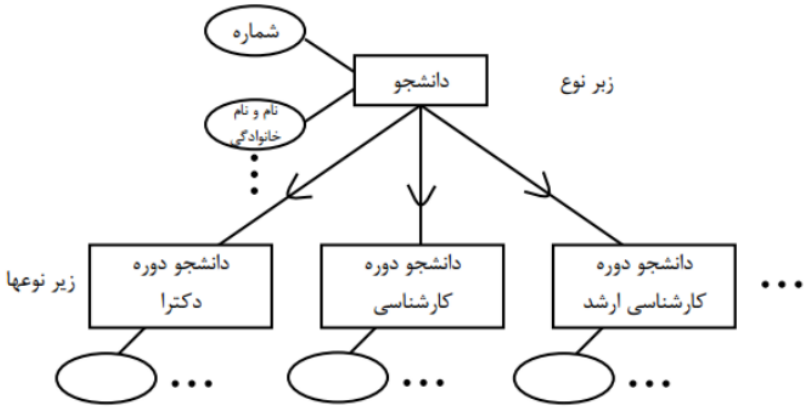
تعدادی ارتباط دیگر + ER = EER :

۱- ارتباط "IS A" یا "گونه ایست" یا "هست یک"
- تکنیک تخصیص یا ویژه نمایی(Specialization)
- تکنیک تعمیم(Generalization)

۲- ارتباط "IS A PART OF"
- تکنیک تجزیه
- تکنیک ترکیب

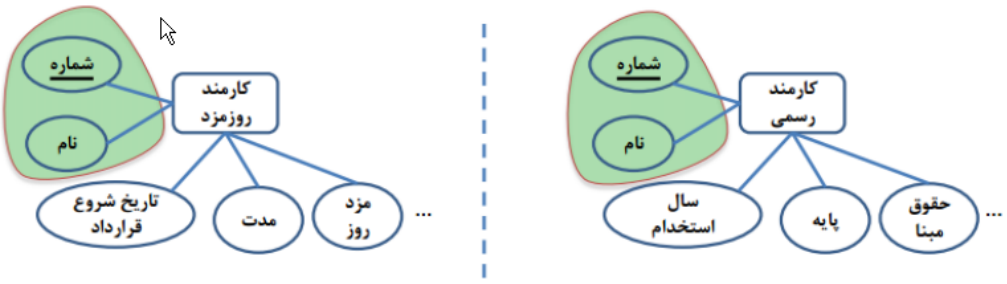
۳- ارتباط با ارتباط
- تکنیک تجمیع(Aggregation)

- تخصیص عبارتست از بازشناسی گونه‌های خاص یک شی بر اساس یک ضابطه‌ی مشخص. می‌گوییم بین هر زیرنوع موجودیت و زبرنوع آن ارتباط "IS A" وجود دارد. به بیان دیگر هر زیرنوع گونه‌ی خاصی است از حداقل یک زبرنوع. زبرنوع مجموعه‌ای از صفات دارد که بین تمام زیرنوع‌های آن مشترک است و بنابراین هر زیرنوع، صفات زبرنوع خود را به ارث می برد. به علاوه هر زیرنوع، صفات خاص خود را دارد.

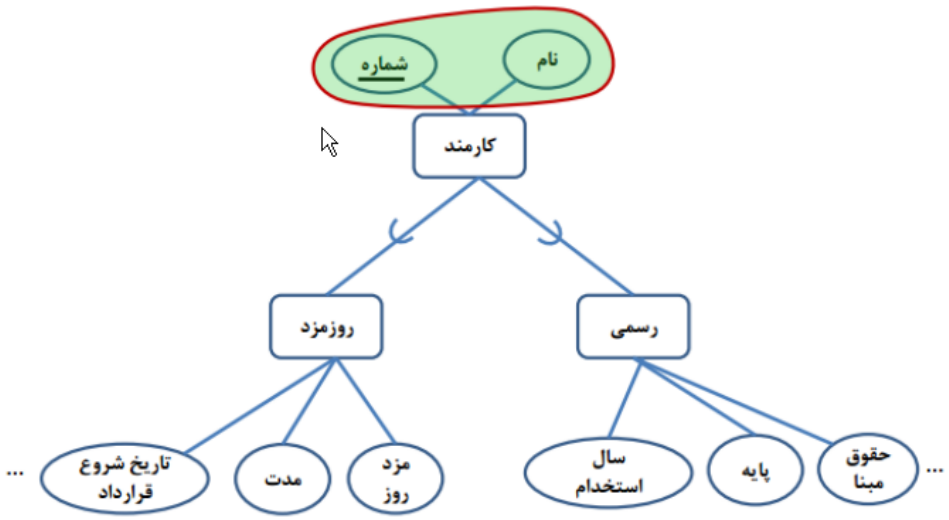


مثال تخصیص

- تعمیم، عکس عمل تخصیص است، به این معنا که با داشتن حداقل دو زیرنوع خاص، صفات مشترک بین آنها(ازجمله شناسه) را در یک مجموعه صفات برای یک زبرنوع موجودیت در نظر می‌گیریم. طبعاً این مجموعه صفات مشترک جدا شده را برای هر زیرنوع تکرار نمی‌کنیم. از این خاصیت زمانی استفاده خواهیم کرد که در سیستم، چندین موجودیت داشته باشیم که دارای صفات مشترکی باشند. در این حالت یک زبرنوع ایجاد کرده و صفات مشترک را برای آن در نظر می‌گیریم و بقیه‌ی موجودیت‌ها را به عنوان زیرنوع آن در نظر می‌گیریم. در این زیرنوع‌ها دیگر صفات مشترک لحاظ نخواهد شد.



یک نوع موجودیت(کارمند) در سطح انتزاعی بالاتر دیده می‌شود:



شرایط تعمیم:

- داشتن شناسه مشترک(یعنی از یک دامنه).
- وجود حداقل دو نوع زیرنوع.
- هرچه صفات مشترک بیشتر باشند، تعمیم توجیه پذیرتر است.

۵– مدل داده رابطه ای

پس از پایگاه داده‌های سلسله مراتبی و شبکه‌ای، که هر یک دارای ضعف‌هایی بودند، متخصصان در جستجوی مدلی بودندکه دارای ساختار داده‌ای با انتزاع قوی باشد. مدل رابطه‌ای در سال ۱۹۷۶ توسط E.F.Codd ابداع گردید.

ساختار داده‌ای آن بر اساس یک مفهوم ریاضی به نام رابطه(Relation) استوار است.

اساسا پایگاه داده‌ی رابطه‌ای، مجموعه‌ای است از تعدادی جدول.

مفاهیم ساختاری جدول عبارتند از: جدول، سطر، ستون

تعریف دامنه یا میدان(Domain):

مجموعه تمام مقادیر یک صفت را دامنه گوییم.

مثال۱: دامنه اعداد صحیح یعنی مجموعه {... ,-1, 0, 1, 2, ...} و صفتی که دامنه‌اش اعداد صحیح است می‌تواند هر یک از این مقادیر را در خود جای دهد.

مثال۲: میدان نام تهیه‌کننده‌ها و شهرها به صورت زیر است:

D_{sname}= {آلومین، پولادین، ایران قطعه، فن‌آوران}, D_{city} = {شیراز، تهران، تبریز}

تذکر : میدان‌های یک رابطه لزوماً از یکدیگر مجزا نیستند، یعنی دو ستون می‌توانند دارای یک میدان یکسان باشند، مثل نام فرد و نام پدر و استان محل تولد و استان محل سکونت.

تعریف رابطه:

زیر مجموعه ای از ضرب دکارتی چند دامنه است.

مثال۱: اگر داشته باشیم D2:Integer و D1:String ، آن گاه هر مجموعه‌ای که عضوهایش زوج های مرتب (D1,D2) باشند یک رابطه است. ساده ترین راه نمایش و پیاده سازی رابطه، جدول دو بعدی است.

D1:String	D2: integer
Ali	10
Reza	20
⋮	⋮

مثال۲: ضرب دکارتی

$$\{1,2,3\} \times \{4,5\} = \{(1,4), (1,5), (2,4), (2,5), (3,4), (3,5)\}$$

آنگاه $R = \{(1,5), (2,4), (3,4)\}$ یک زیرمجموعه و بنابراین یک رابطه است.

مثال۳:

$$\{1,2,3\} \times \{k, m\} \times \{A, B\} = \{(1, k, A)\}, \{1, k, B), (2, k, A), (2, k, B), (3, k, A), (3, k, B), (1, m, A), \ldots\}$$

ضرب دکارتی ۳ مجموعه فوق مجموعه‌ای از ۳ تائی‌های مرتب را می‌دهد. هر زیر مجموعه‌ای از حاصلضرب فوق، یک رابطه است. از آنجا که رابطه مذکور از ۳تائی‌های مرتب تشکیل شده است، درجه این رابطه ۳ می‌باشد. پس **درجه رابطه** تعداد صفات خاصه رابطه است.

مثال۴:

{S1, S2} × {پولادین، فن‌آوران} × {شیراز، تبریز، تهران} =

{(S1, تبریز ,پولادین), (S1, تهران ,پولادین), (S1, شیراز ,فن‌آوران), (S1, تبریز ,فن‌آوران), (S1, تهران ,فن‌آوران)
...}

تعداد ستونهای جدول همان درجه رابطه است.

تعداد سطرهای رابطه در یک لحظه از حیات آن، **کاردینالیتی رابطه** نام دارد و در طول حیات رابطه متغیر است.

S#	Sname	City
S1	فن‌آوران	تهران
S1	فن‌آوران	تبریز
S1	فن‌آوران	شیراز
S1	پولادین	تهران
S1	پولادین	تبریز
.	.	.

رابطه از دو مجموعه عنوان(Heading) و پیکر(Body) تشکیل یافته است. مجموعه عنوان مجموعه اسامی صفات خاصه است و مجموعه پیکر، مجموعه‌ای است متغیر در زمان از سطرها.

H_s = {S#, Sname, City}

B_s = {(S1, تهران ,فن‌آوران), (S2, تبریز ,ایران قطعه), (S3,تبریز , پولادین)}

زوج مرتب یا تاپل(Tuple):

هر سطر یک جدول، معادل یک تاپل است. هر یک از ستون های یک جدول نیز معرف یک صفت است. در مدل رابطه‌ای صفت‌ها از دامنه‌های ساده(تجزیه ناپذیر) تشکیل می‌شوند و دامنه‌های تودرتو مجاز نیستند.

تناظر بین اجزای دو مفهوم رابطه و جدول:

اجزاء مفهوم رابطه	اجزاء مفهوم جدول
رابطه	جدول
تاپل	سطر(رکورد)
صفت	ستون(فیلد)
میدان	مجموعه مقادیر ستون
درجه	تعداد ستون ها
کاردینالیتی	تعداد سطرها

ویژگی های رابطه:

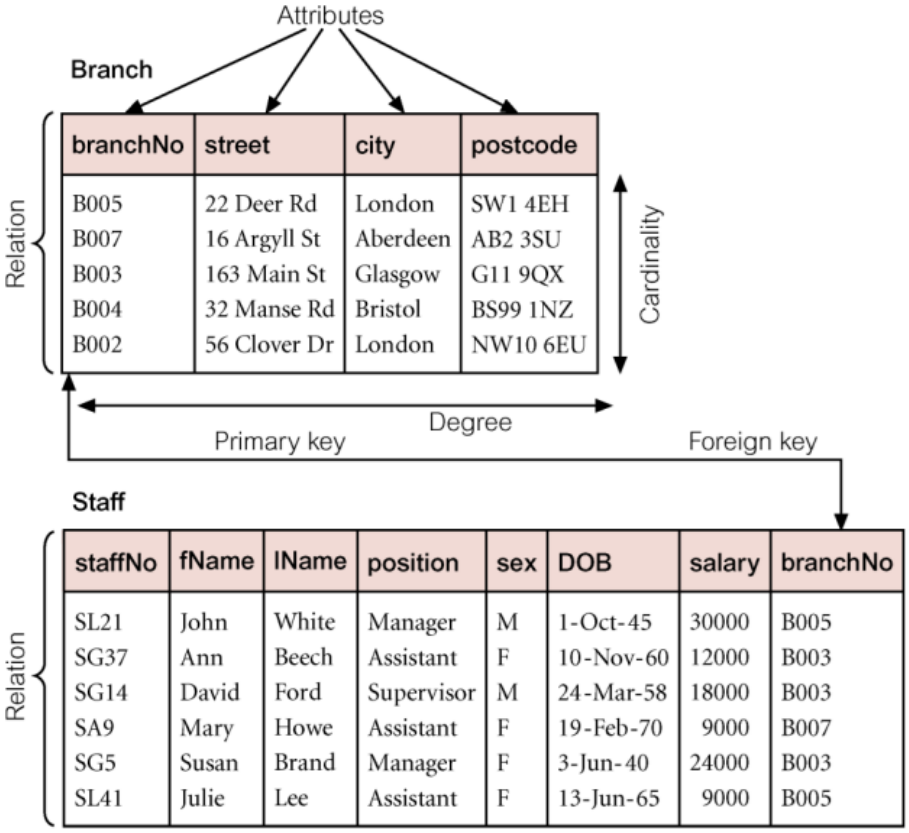
- هر تاپل باید منحصر به فرد باشد. (از خواص مجموعه‌ها در ریاضیات)
- ترتیب صفات خاصه اهمیتی ندارد.
- ترتیب تاپل‌ها اهمیتی ندارد.
- هر سلول رابطه شامل فقط یک مقدار است. بنابراین در بانک اطلاعاتی رابطه‌ای نمی‌توان مثلاً جدولی به صورت زیر تعریف کرد که فیلدی از آن مرکب بوده و از فیلدهای ساده‌تری تشکیل شده باشد. به عبارت دیگر از تقاطع هر سطر و ستون باید یک مقدار بدست آید.

نام		تاریخ تولد			معدل
نام کوچک	فامیل	روز	ماه	سال	

مقدار اتومیک(atomic): مقداری است ساده که قابل تجزیه به مقادیر دیگر نباشد، به بیان دیگر از یک میدان ساده برگرفته شود. مثلاً در شکل فوق نام و تاریخ اتومیک نیست ولی معدل اتومیک است.

رابطه‌ای که همه مقادیر صفات خاصه آن اتومیک باشند، به رابطه نرمال شده(Normalized) موسوم است. در واقع اصطلاح رابطه در بانک رابطه‌ای، همیشه به رابطه نرمال اطلاق می‌شود.

مثال:



انواع کلید در مدل رابطه ای:

۱- ابر کلید(Super Key):

هر صفت خاصه یا ترکیبی از صفات در رابطه که یکتایی مقدار(تکرار نشدن یا Unique) در گستره رابطه داشته باشد.

ابر کلید در عمل کاربردی ندارد و در برخی از گونه های نرمال کاربرد دارد.

مثلاً «شماره دانشجویی» و «نام دانشجو ـ شماره دانشجویی» هر دو ابر کلید هستند.

۲- کلید کاندید(Candidate Key):

هر صفت خاصه یا ترکیبی از صفات در یک رابطه که اولاً یکتایی مقدار و ثانیاً کاهش ناپذیری یا کمینگی داشته باشد.

- تفاوت ابرکلید با کلیدکاندید در کاهش ناپذیر بودن کلید کاندید است، در حالی که ابر کلید می تواند کاهش پذیر باشد.
- کلید کاهش ناپذیر، کلیدی است که اگر هر جزء آنرا حذف کنیم دیگر کلید نباشد.
- هر رابطه می تواند چند کلید کاندید داشته باشد.
- وقتی یک کلید، بیش از یک صفت داشته باشد، به آن کلید مرکب می گوییم.
- رابطه ای که در آن کلیه صفات با هم کلید کاندید شوند، را رابطه تمام کلید(All-Key) گوییم.

مثلاً «شماره دانشجویی» کلید کاندید است ولی «نام دانشجو ـ شماره دانشجویی» کلید کاندید نیست.

نکته : در هر رابطه، حداقل یک کلید کاندید وجود دارد. زیرا در بدترین حالت، مجموعه عنوان، کلید کاندید رابطه است.

۳- کلید اصلی(Primary Key):

یکی از کلیدهای کاندید که برای شناسایی تاپل ها به طور یکتا درون رابطه توسط طراح پایگاه داده انتخاب شود.

ضابطه‌های انتخاب کلید اصلی:

- شناسه رایج برای آن موجودیت در محیط عملیاتی باشد.
- طول کوتاهتر داشته باشد.
- مقدارش حتی الامکان تغییر نکند.

مثال: در یک اداره برای کارمندان هم کدملی و هم کدکارمندی کلید کاندید خواهد بود، ولی به دلیل کوتاه‌تربودن کدکارمندی و نیز شناسایی افراد در اداره با کدکارمندی مرسوم‌تر است، بنابراین کدکارمندی به عنوان کلید اصلی انتخاب می‌شود.

۴- کلید بدیل یا فرعی(Alternative Key):

کلیدهای کاندیدی که به عنوان کلید اصلی انتخاب نمی‌شوند.

کلید فرعی در عمل پیاده سازی نشده است.

۵- کلید خارجی(Foreign Key):

دو رابطه R1 و R2 را در نظر می‌گیریم. صفت(یا ترکیبی از صفات) درR2 که در این رابطه کلید نیست اما در R1 کلید اصلی است، یک کلید خارجی برای رابطهR2 نامیده می شود.

- کلید خارجی برای برقراری ارتباط بین موجودیت‌ها(جداول) به کار می‌رود.
- کلید خارجی را با نقطه چین زیر صفت یک رابطه نشان می دهیم.

نکته: از بین تمام کلیدهای نامبرده شده، کلید خارجی تنها کلیدی است که می تواند هیچ مقدار(Null able) باشد.

تراکنش(Transaction):

هر برنامه‌ای که توسط کاربر در محیط بانک اطلاعاتی اجرا میشود تراکنش نام دارد. تراکنش یک واحد منطقی از کار است و معمولاً شامل چندین عمل بانک اطلاعاتی است. تفاوت اصلی یک تراکنش با یک برنامه معمولی در محیط غیربانکی این است که تراکنش همواره به DBMS تسلیم می‌شود و DBMS در اعمال هرگونه کنترل و حتی به تعویق انداختن و ساقط کردن آن آزادی عمل دارد.

BEGIN TRANSACION READ A; A = A-500; WRITE A; READ B; B = B+500; WRITE B; END TRANSACION	Transaction started. At this stage A = 10000 and B = 14000 (**Current consistent state)** At the end of the transaction, A = 9500 and B = 14500(**New consistent state)**

شرط سازگاری داده‌ها این است که حاصل A+B ثابت بماند.

امنیت (Security):

امنیت به معنای محافظت در برابر خطراتی از قبیل آتش‌سوزی و سرقت و نیز جلوگیری از دستیابی غیرمجاز به داده-هاست. راه‌های مختلفی برای جلوگیری از دستیابی غیرمجاز به داده‌ها مثل استفاده از رمز عبور(Password) وجود دارد ولی همواره ممکن است افرادی پیدا شوند و این رمزها را بگشایند.

جامعیت(integrity):

صحت، دقت و سازگاری و اعتبار داده‌های ذخیره‌شده در پایگاه در تمام لحظات و پیروی از مقررات سیستم. در واقع جامعیت مجموعه ای است از امکانات برای انجام کنترل داده‌های ذخیره شده درپایگاه داده.

مثال:

- یکسری کنترل‌ها وجود دارد که دانشجوی مشروط شده نمی تواند بیشتر از ۱۴ واحد اخذ کند.
- موجودی واقعی حسابهای بانکی نباید منفی باشد و یا شخص نتواند بیش از موجودی خود از حساب برداشت کند.

در بانک اطلاعاتی آنچه در درجه اول اهمیت قرار دارد داده است نه برنامه. داده‌های بانک اطلاعاتی را مانا(Persistent) می‌نامند زیرا برنامه‌ها می‌آیند و می‌روند اما داده‌ها می‌مانند. مثلاً برنامه‌ای که پولی را به حسابی می‌ریزد یا برداشت می‌کند آن‌قدرها مهم نیست. مهم این است که موجودی حسابها اشتباه نشود.

قوانین جامعیت در مدل رابطه‌ای:

- جامعیت دامنه‌ای: یعنی تمام صفات درتمامی رابطه ها ازنوع دامنه خود باشند. مثلا ۷۴/۱ به عنوان شماره دانشجویی که عدد صحیح است، پذیرفته نشود.
- جامعیت درون رابطه‌ای: یعنی هر رابطه به تنهایی صحیح باشد. مثلا عضو تکراری نداشته باشد وکلیدهایش درست باشند و کلیدها دارای مقدار تهی یا تکراری نباشد.
- جامعیت ارجاعی: یعنی کلید خارجی درست تعریف شده باشد. مثلا کلید خارجی دریک رابطه حتما در رابطه دیگر کلید باشد و مقداری که به کلید خارجی داده می‌شود، در جدول دیگر وجود داشته باشد.

تضمین جامعیت بانک اطلاعاتی:

آقای جیم‌گری(Jim Gray) در سال ۱۹۸۱ ثابت کرد که چهار کنترل زیر لازم است روی تمامی تراکنش‌ها در بانک اطلاعات اعمال گردد تا صحت و جامعیت آن تضمین شود این کنترلها به خواص ACID معروفند.

A	C	I	D
Atomicity	Consistency	Isolation	Durability
غیر قابل تجزیه	صحت	انزوا(مجزا یا عایق)	پایائی(پایداری یا ماندگاری)

۱- غیرقابل تجزیه (Atomicity): این خاصیت به همه یا هیچ موسوم است. منظور این است که یا تمام دستورات یک تراکنش باید اجراء شود یا هیچکدام از آنها نباید اجراء شود. مثلاً تراکنشی می‌خواهد مبلغی را از حسابی به حساب دیگر منتقل کند. فرض کنید بخش اول کار (برداشت پول) در یک ماشین و بخش دوم کار (واریز پول) در ماشینی دیگر اجراء می‌شود. حال در نظر بگیرید پس از انجام بخش اول (برداشت پول) ارتباط با ماشین دوم ناگهان قطع شود. بدیهی است که در این حالت باید پول برداشت شده دوباره به همان حساب اول بازگردانده شود.

۲- صحت(Consistency): این خاصیت به این صورت بیان می‌گردد که: هر تراکنش اگر به تنهایی اجراء شود بانک اطلاعات را از حالتی صحیح به حالت صحیح دیگری منتقل می‌کند. یعنی این تراکنش ممکن است دو نوع پایان داشته باشد:

الف) پایان موفق که آنرا انجام(Commit) می‌نامند.

ب) پایان ناموفق که آنرا سقوط(Abort) می‌نامند. در این حالت Rollback باید انجام شود.

۳- انزوا(Isolation): در بانک اطلاعاتی ممکن است تراکنش‌های همروند وجود داشته باشد(مثل Multitasking در سیستم عامل Windows که چند برنامه همزمان اجراء می‌شوند). بر طبق خاصیت انزوا همروندی تراکنش‌ها باید کنترل شود تا اثر مخرب بر روی هم نداشته باشند به عبارتی دیگر اثر تراکنشهای همروند روی یکدیگر چنان است که گویا هر کدام در انزوا انجام می‌شود.

// A is 10000, B is 14000
BEGIN TRANSACION1
READ A; // A is 10000
A = A-500;
	BEGIN TRANSACION2
	READ A;
	READ B;
	Tax = 0.10*A+0.05*B;
		کدام حالت روی می‌دهد؟
		a) Tax = 950+725 = 1675
		b) Tax = 1000+700 = 1700
		c) Tax = 1000+725 = 1725

WRITE A; // 9500
READ B; //B is 14000
B = B+500;
WRITE B; //14500
END TRANSACION1
// A is 9500, B is 14500
	WRITE Tax;
	END TRANSACION2
		d) Tax = 950+700 = 1650

این کنترل توسط بخشی از DBMS به نام واحد کنترل همروندی(Concurrency Control) انجام می‌شود. کتاب دیت مفهوم Isolate را به صورت زیر بیان میکند: Isolate یعنی به هنگام‌سازی حاصل از تراکنش T1 توسط تراکنش دیگری مثل T2 قابل مشاهده نیست. مگر اینکه عمل COMMIT را اجرا کند. COMMIT موجب می‌شود تا به‌هنگام‌سازی‌هایی که توسط یک تراکنش انجام شده، توسط تراکنش‌های دیگر قابل رؤیت باشد. اگر تراکنش ROLLBACK را اجراءکند تمام به‌هنگام‌سازی‌هایی که انجام شده از بین می‌روند.

۴- پایائی(Durability): براساس این خاصیت تراکنش‌هایی که به مرحله انجام(Commit) برسند اثرشان ماندنی است و هرگز به طور تصادفی از بین نمی‌رود. مثلاً اگر مبلغی به حسابی واریز شود و تراکنش مربوطه انجام یافته اعلام شود، حتی در صورت وقوع آتش‌سوزی در آن شعبه بانک، مشتری نباید متضرر شود، یعنی مثلاً عمل واریز قبل از اعلام انجام موفق، باید در جای دیگری نیز ثبت شده باشد.

تذکر ۱: دو عمل یکپارچگی و پایائی توسط واحدی از DBMS به نام واحد مدیریت بازگرد(Recovery Management) انجام می‌گیرد.

تذکر ۲: در تراکنش‌ها باید تضمین شود که اجرای ناپیوسته(Interleaved) مجموعه‌ای از تراکنش‌های همزمان، معمولاً به صورت سریال انجام شود. یعنی اجرای سریال آنها همان نتایجی را دارد، که همان تراکنش‌ها به ترتیب نامشخصی اجراء می‌شوند.

۶– طراحی منطقی پایگاه داده:

به فرایند ساخت یک مدل از اطلاعات محیط عملیاتی بر اساس یک مدل داده‌ای مشخص ولی مستقل از هر DBMS بخصوص و سایر ملاحظات فیزیکی، طراحی منطقی گفته می شود. طراحی منطقی پایگاه داده، مدل مفهومی را به یک مدل داده منطقی می‌نگارد. این مدل تحت تاثیر مدل داده‌ای مورد استفاده در پایگاه داده است(مثل مدل داده‌ای رابطه‌ای). در ادامه حالت‌های مختلف تبدیل نمودار ER به رابطه شرح داده می شود.

۱– نگاشت ارتباط های N:M :

شرایط این حالت عبارتند از:

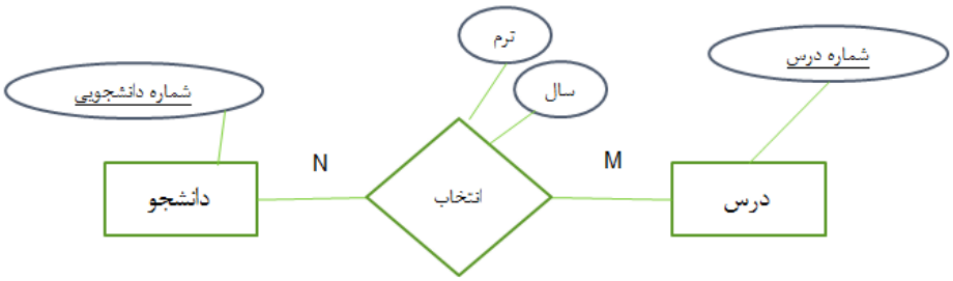
- وجود حداقل ۲ موجودیت
- موجودیت ها باید مستقل باشند
- ارتباط از نوع N:M

روش طراحی به این صورت است که برای هر یک از موجودیت‌ها، یک جدول در نظر بگیریم و یک جدول هم ارتباط بین موجودیت‌ها را نشان دهد. در واقع با وجود n موجودیت، حداقل n+1 جدول طراحی می‌شود.

کلید کاندید جدول مربوط به ارتباط موجودیت‌ها، کلید کاندید موجودیت های مستقل می شود.

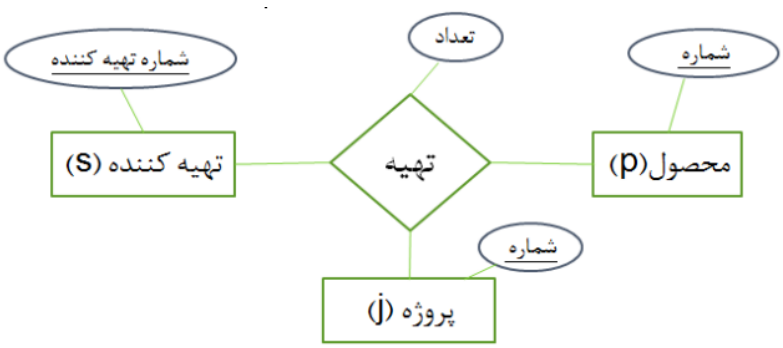
در این حالت طراحی، مشارکت الزامی یا غیرالزامی موجودیت‌ها هیچ تاثیری در طرز طراحی ندارد.

مثال ۱:



- Student (Stid, Stname, tel, address, …)
- Course (Coid, Coname, unit, …)
- StC (Stid, Coid, term, year, …)

مثال ۲:



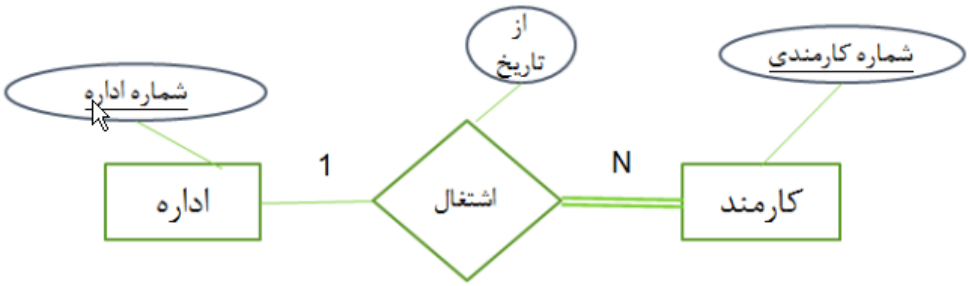
۲– نگاشت ارتباط های 1:N :

شرایط این حالت عبارتند از:

- وجود ۲ موجودیت
- موجودیت‌ها باید مستقل باشند
- ارتباط از نوع 1:N

یک جدول برای موجودیت طرف ۱ و جدول دوم برای موجودیت طرفN . برای نمایش ارتباط کافیست کلید کاندید موجودیت طرف یک، به عنوان کلید خارجی در موجودیت طرف N قرار گیرد. این کلید خارجی جز تشکیل دهنده کلید کاندید رابطه طرف N قرار نمی‌گیرد.

مثال:



- Office (Officeid, name, address, …)
- Employee (Eid, Ename, tel, address, …, Officeid, date)

توجه : مشارکت الزامی در شکل بالا نشان می‌دهد که هیچ کارمندی نیست که در اداره‌ای اشتغال نداشته باشد. این مسئله باید در هنگام تعریف در DBMS با استفاده از قید Not Null لحاظ شود. به عبارتی محدودیت الزامی بودن به طور مستقیم در رابطه‌ها و جدول‌ها دیده نمی‌شود و در شِما آورده می‌شود.

۳– نگاشت ارتباط های 1:1 :

شرایط این حالت عبارتند از:

- وجود ۲ موجودیت
- موجودیت‌ها باید مستقل باشند
- ارتباط از نوع 1:1

برای طراحی، بسته به وضعیت مشارکت، عمل می‌کنیم:

۱- اگر مشارکت هر دو موجودیت غیرالزامی باشد، نیاز به سه جدول داریم و مشابه حالت اول عمل می‌کنیم.

۲- اگر یکی از طرفین الزامی و دیگری غیر الزامی باشد، طراحی با دو جدول انجام می‌شود و کلید کاندید طرف غیر الزامی، به عنوان کلید خارجی در موجودیت دارای مشارکت الزامی قرار می گیرد و جزو کلید کاندید آن نمی‌شود.

۳- اگر مشارکت هر دو موجودیت الزامی باشد، فقط یک جدول می‌تواند کافی باشد. کلید کاندید این جدول، کلید کاندید هر یک از موجودیت‌ها می‌تواند باشد.

مثال:

- Teacher (Tid, Tname, tel, address, …)
- Department (Deptid, Deptname, …, Tid)

اگر مشارکت هر دو طرف الزامی باشد آنگاه یک جدول با دو کلید کاندید به شکل زیر خواهیم داشت:

- Dept_Teach (Deptid, …, Tid, …)

اگر مشارکت دو طرف غیرالزامی باشد آنگاه سه جدول به شکل زیر خواهیم داشت:

- Teacher (Tid, Tname, tel, address, …)
- Department (Deptid, Deptname, …, Tid)
- Dept_Teach (Deptid, Tid)

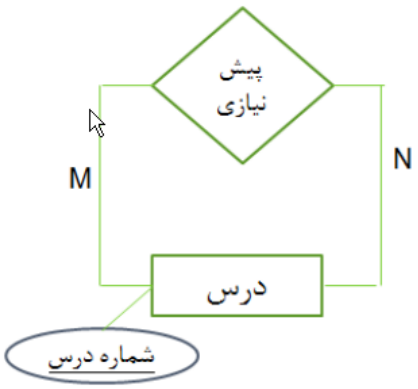
۴– ارتباط با خود (N:M):

شرایط این حالت عبارتند از:

- وجود ۱ موجودیت
- موجودیت‌ها باید مستقل باشند
- ارتباط از نوع N:M

برای طراحی نیاز به دو جدول داریم، یکی برای موجودیت و دیگری برای ارتباط. کلید کاندید موجودیت مستقل، دو بار در جدول ارتباط به عنوان کلید خارجی ظاهر می‌شود و ترکیب آن دو، کلید کاندید جدول ارتباط می‌شوند. نام یکی از دو صفت کلید خارجی را باید تغییر دهیم. اگر ارتباط موجودیت با خودش، صفتی هم داشته باشد، در جدول ارتباط قرار می‌گیرد.

مثال:



- Course (Coid, …)
- PreCo (Coid, Copreid)

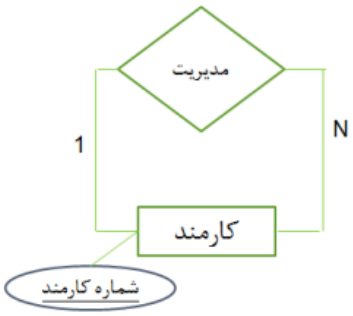
۵– ارتباط با خود (1:N):

شرایط این حالت عبارتند از:

- وجود ۱ موجودیت
- موجودیت‌ها باید مستقل باشند
- ارتباط از نوع 1:N

برای طراحی فقط یک رابطه کافیست. کلید کاندید جدول با یک اسم دیگر به عنوان کلید خارجی در رابطه تکرار می شود. اگر مشارکت موجودیت غیرالزامی و درصد مشارکت هم ضعیف باشد، می توان دو جدول مثل حالت چهار طراحی کرد. این طرز حالت پنجم، باعث ایجاد حلقه ارجاع می‌شود و رابطه به خودش رجوع می‌کند.

مثال:



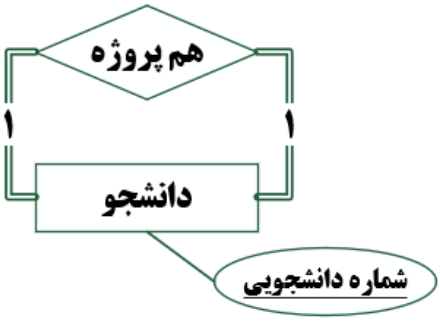
- Emp (Eid, …, Emid)

توجه: دامنه مقادیر Eid و Emid یکی است. این مثالی است از حالتی که رابطه مرجع و رجوع کننده در تعریف کلید خارجی یکی است. اگر این رابطه را با داده‌های اداره پر کنیم، سطر مربوط به کارمند مدیر، در بخش Emid مقدار null می‌گیرد.

۶– ارتباط با خود (1:1):

شرایط این حالت عبارتند از:

- وجود ۱ موجودیت
- موجودیت‌ها باید مستقل باشند
- ارتباط از نوع 1:1



برای طراحی، بسته به وضعیت مشارکت، به یک یا دو جدول نیاز داریم.

۱- اگر مشارکت موجودیت الزامی باشد، طراحی با یک جدول انجام می‌شود و به دو طرح می‌توان این جدول را طراحی نمود.

۲- اگر مشارکت موجودیت غیر الزامی باشد، و سیستم به دفعات زیاد نیاز به شماره دانشجویی دو فرد هم پروژه داشته باشد و مشارکت در رابطه هم پروژگی زیاد نباشد، طراحی با دو جدول انجام می‌شود.

مثال: فرض کنید در دانشگاه، هر دانشجو الزاما باید با یک دانشجوی دیگر هم پروژه باشد:

- Student (Stid, …, CoprojStid)

توجه: صفت Stid و CoprojStid هم دامنه هستند و یکی تغییر نام داده شده است. در حالت مشارکت الزامی اگر ملاحظات دیگری مطرح نباشد، مثل اینکه درجه رابطه بزرگ شود و یا کارایی سیستم در عملیات بازیابی کاهش یابد به شکل زیر هم می‌توان طراحی را انجام داد:

- Student (Stid, …, CoprojStid, …)

برای مشارکت غیر الزامی هم می توان از دو جدول زیر استفاده کرد:

- Student (Stid, Stname, address, …)
- StuCoproj (stid, CoprojStid)

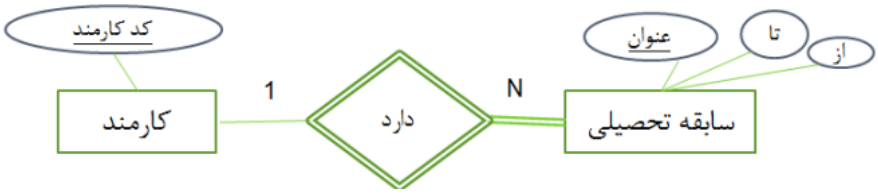
۷- ارتباط با موجودیت ضعیف:

برای طراحی، یک جدول برای موجودیت قوی در نظر گرفته می‌شود و یک جدول نیز برای موجودیت ضعیف و خود ارتباط. کلید کاندید موجودیت قوی به عنوان کلید خارجی صفت ممیزه موجودیت ضعیف ترکیب می‌شود.

توجه: صفت ممیزه یا کلید جزئی، به طور سراسری یکتا نیست، بلکه در بین نمونه‌هایی که با موجودیت قوی ارتباط دارند، یکتاست.

نکته: نوع موجودیت ضعیف را گاهی با یک صفت مرکب یا چند مقداری هم نشان می‌دهند. انتخاب نوع نمایش بر عهده طراح پایگاه داده است. یک معیار انتخاب این است که اگر صفات آن زیاد باشد، از موجودیت ضعیف استفاده شود. ولی در مواردی که موجودیت ضعیف به طور مستقل در ارتباط‌های دیگر شرکت داشته باشد، نمایش آن با صفت مرکب یا چند مقداری صحیح نیست.

مثال:



در این مدل، عنوان صفت ممیزه است. زیرا هر کارمند سابقه تحصیلی تکراری ندارد.

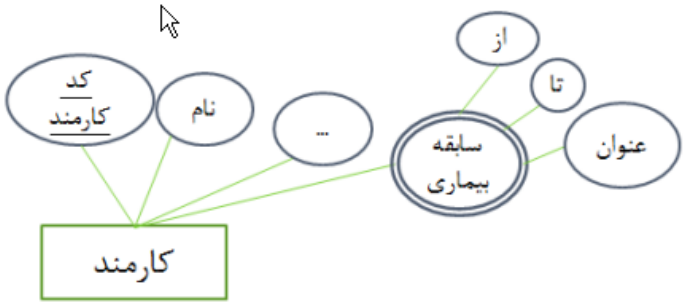
- Emp (Eid, …)
- EmpEdu (Eid, Edutitle, from, to)

۸- موجودیتی که صفت چند مقداری داشته باشد:

برای طراحی، یک رابطه برای موجودیت قوی درنظرگرفته می‌شود و یک رابطه برای صفت چند مقداری. کلید کاندید موجودیت به عنوان کلید خارجی به صفت چند مقداری در رابطه‌اش، افزوده می شود. در حالت کلی اگر موجودیتی n صفت ساده یا مرکب تک مقداری و m صفت چند مقداری داشته باشد، به m+1 رابطه نیاز داریم.

نکته: در مدل سازی، موجودیت ضعیف به صفت چندمقداری ارجحیت دارد ولی روش عمومی طراحی آنها یکی است.

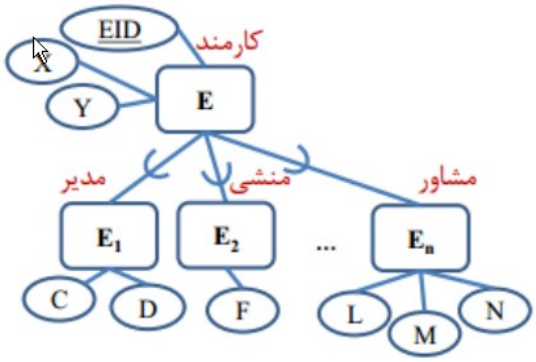
مثال:



واحد درمان یک اداره، سابقه بیماری‌های کارمندان را نگهداری می‌کند. هر کارمند ممکن است سابقه بیماری‌های مختلفی داشته باشد و حتی به یک بیماری چندین بار مبتلا شده باشد.

- Emp (Eid, Ename, …)
- Illness (Eid, title, from, to)

۹– وجود ارتباط IS-A بین دو موجودیت:



نوع موجودیت E، n زیر نوع دارد.

۴ تکنیک برای طراحی وجود دارد:

تکنیک اول) طراحی با n+1 رابطه. یک رابطه برای زبرنوع و یک رابطه برای هریک اززیرنوع‌ها.

- E (EID, X, Y)
- E1 (E, C, D)
- E2 (E, F)
- …
- En (E, L, M, N)

مزیت: شرط خاصی از نظر نوع تخصیص ندارد(تکنیک‌های دیگری که مطرح می‌شود، همگی برای شرایط خاص هستند).

عیب: اگر بخواهیم در مورد یک زیرنوع، اطلاعات کامل به دست آوریم، باید Join کنیم.

تکنیک دوم) طراحی با n رابطه. برای زبرنوع، رابطه‌ای طراحی نمی‌کنیم. بنابراین صفات مشترک باید در رابطه نمایشگر هر زیرنوع وجود داشته باشد.

- E1 (EID, X, Y, C, D)
- E2 (EID, X, Y, F)
- …
- En (EID, X, Y, L, M, N)

مزیت: برای به دست آوردن اطلاعات کامل زیرنوع‌ها نیازی به Join نیست.

تکنیک سوم) طراحی فقط با یک رابطه با استفاده از صفت نمایشگر نوع زیرنوع‌ها

- E (EID, X, Y, C, D, F, L, M, N, TYPE)

  100 x1 y1 c1 d1 مدیر

  200 x2 y2 f2 منشی

  300 x3 y3 l3 m3 n3 مشاور

مزیت: برای به دست آوردن اطلاعات کامل زیرنوع‌ها نیازی به Join نیست.

عیب: مقدار null زیاد دارد و درجه رابطه زیاد است.

تکنیک چهارم) طراحی فقط با یک رابطه. با استفاده از آرایه بیتی. هر بیت نمایشگر نوع یک زیر نوع. در واقع برای نمایش هر نمونه موجودیت، بسته به اینکه در مجموعه نمونه‌های کدام زیرنوع باشد، بیت مربوطه‌اش را ۱ می کنیم.

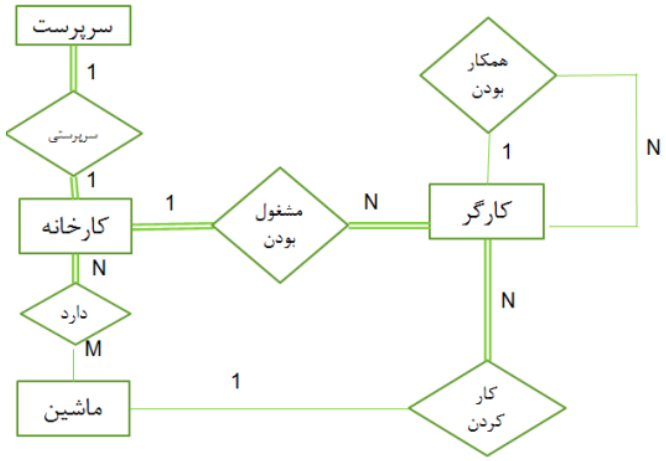
- E (EID, X, Y, C, D, F, L, M, N, TB1, TB2, …, TB3)

  100 x1 y1 c1 d1 1 0 0

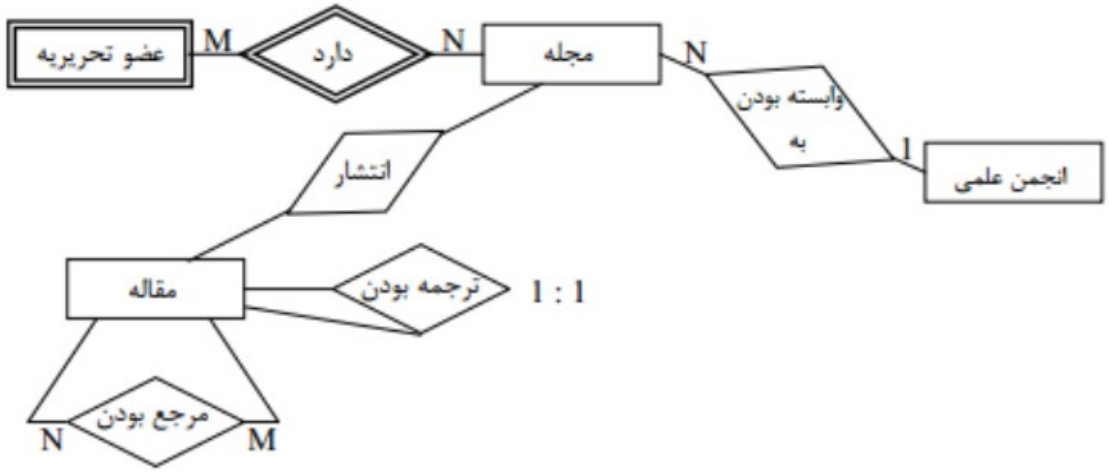
  200 x2 y2 f2 0 1 0

  300 x3 y3 l3 m3 n3 0 0 1

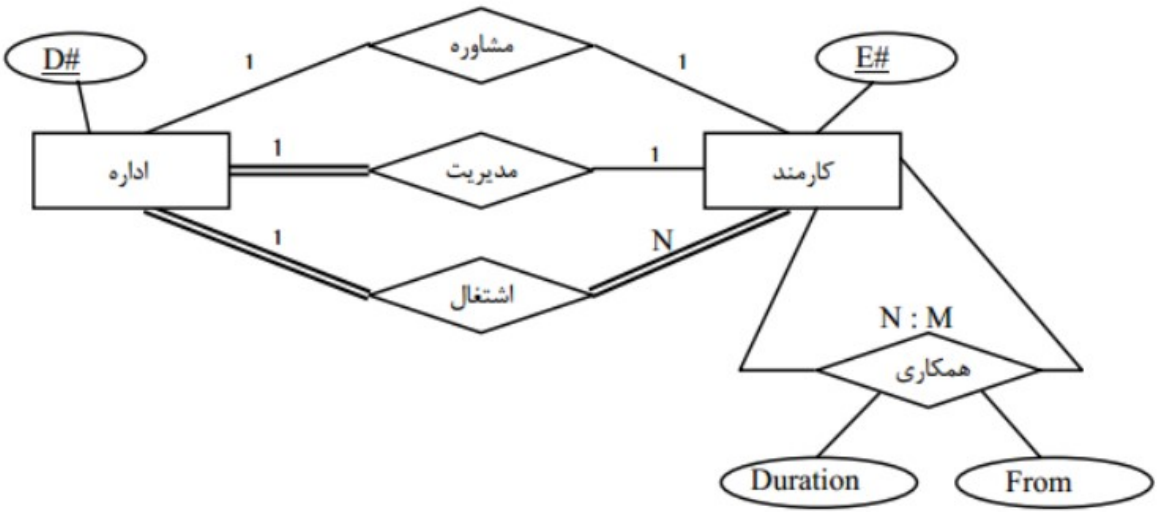
تمرین ۱: با توجه به نمودار ER داده شده، پایگاه رابطه ای مربوطه را طراحی کنید.



تمرین ۲: با استفاده از روش طراحی بالا به پایین نمودارEER زیر را به مدل رابطه ای تبدیل کنید. کلید اصلی موجودیت‌ها عبارتند از: مجله : عنوان، انجمن علمی: عنوان، مقاله: عنوان و کد مقاله، کلید صفت ممیزه عضو تحریریه: نام



تمرین ۳: با توجه به نمودار ER داده شده، پایگاه رابطه ای مربوطه را طراحی کنید.



۷– نرمال سازی:

پس از شناسایی موجودیت‌ها و رسم نمودار ER، صفات مورد نیاز و ارتباط بین موجودیت‌ها، جدول‌های (روابط) مورد نیاز باید طراحی شوند. یک تصور نادرست این است که هرچه صفات جدول بیشتر باشد، از اکمال و جامعیت بالاتری برخوردار است، حال آن که گزینش درست صفات، از بروز برخی مشکلات در بانکهای اطلاعاتی اجتناب می‌کند. نارسایی‌ها در طراحی و یک طراحی نامناسب، باعث ایجاد پدیدهٔ «آنومالی» می‌گردد.

آنومالی، در سه محور قابل طرح و بررسی است:

الف) انجام‌ناپذیری یکی از عملیات در بانک

ب) بروز تبعات نامطلوب در پی انجام یک عملیات مبنایی

ج) فزونکاری برای انجام یک عملیات مبنایی

قبل از شرح موارد آنومالی، چند مفهوم مقدماتی دیگر ذکر می‌شود، سپس در قالب یک مثال آنومالی و راههای رفع آن گفته خواهد شد.

وابستگی تابعی(Functional Dependency)

شبیه تعریف تابع در ریاضیات، دررابطه R، صفت y با صفت x وابستگی تابعی دارد اگر به ازای هر مقدار x، تنها یک مقدار y وجود داشته باشد.

وابستگی تابعیy با x را به صورت $x \longrightarrow y$ نشان می‌دهیم.

در رابطه R1، وابستگی تابعی $A \longrightarrow B$ و $B \longrightarrow C$ را بررسی می‌کنیم:

R1

A	B	C
a_1	b_1	c_1
a_2	b_2	c_3

a_1	b_1	c_2
a_3	b_4	c_2
a_5	b_1	c_1

وابستگی $A \rightarrow B$ برقرار است. زیرا به ازای هر مقدار از A، دو مقدار متفاوت از B وجود ندارد. اما وابستگی $B \rightarrow C$ برقرار نیست. زیرا به ازای b_1، دو مقدار c_1 و c_2 در R وجود دارد.

وابستگی تابعی بیان کننده قواعد محیط عملیاتی است. این قواعد را می‌توان در نمودار «وابستگی تابعی» یا نمودار FD نمایش داد.

فرض کنید قواعد زیر در یک محیط عملیاتی برقرار است. نمودار FD متناظر را رسم می‌کنیم. یک جدول با مقادیر دلخواه متناظر و با نام FIRST ایجاد می‌کنیم.

قاعدهٔ ۱: هر ناشر، تعدادی کتاب منتشر می‌کند.

قاعدهٔ ۲: هر ناشر از یک کتاب، شمارگان مشخصی منتشر می‌کند.

قاعدهٔ ۳: هر ناشر در یک شهر دفتر دارد.

قاعدهٔ ۴: هر ناشر دارای یک رتبه صنفی است.

قاعدهٔ ۵: ناشران یک شهر دارای یک رتبه صنفی هستند.

با توجه به قواعد بالا داریم:

P#	Book#	Qty	City	Grade
شماره ناشر	شماره کتاب	شمارگان	شهر	رتبه صنفی ناشر

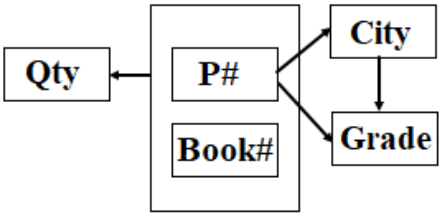
1) $P\# \nrightarrow Book\#$

2) $(P\# , Book\#) \rightarrow Qty$

3) $P\# \rightarrow City$

4) $P\# \rightarrow Grade$

5) $City \rightarrow Grade$



P#	Book#	Qty	City	Grade
P_1	b_1	1000	c_1	40
P_2	b_4	3000	c_2	60
P_3	b_3	2000	c_1	40
P_1	b_2	3000	c_1	40
P_4	b_3	4000	c_3	50
P_4	b_5	3500	c_3	50

نکته: تمام صفات یک موجودیت با صفت کلیدآن، وابستگی تابعی دارند.

مسئله: اگر $A \rightarrow B$ و $B \rightarrow C$ برقرار باشد، آیا $A \rightarrow C$ برقرار است؟

پاسخ: اگر $A \rightarrow C$ برقرار نباشد، در این صورت داریم: $a_1 \ldots c_2$, $a_1 \ldots c_1$

چهار حالت زیر قابل تصور است:

حالت اول	حالت دوم	حالت سوم	حالت چهارم
a_1 b_1 c_1	a_1 b_2 c_1	a_1 b_1 c_1	a_1 b_2 c_1
a_1 b_1 c_2	a_1 b_2 c_2	a_1 b_2 c_2	a_1 b_1 c_2

حالت اول و دوم ناقض فرض $B \rightarrow C$ و حالت سوم و چهارم ناقض فرض $A \rightarrow B$ است.

در نتیجه طبق برهان خلف داریم: $A \rightarrow C$

وابستگی تابعی کامل(Fully Functional Dependency)

صفتy با صفت x وابستگی تابعی کامل دارد $(x \Rightarrow y)$ اگر:

۱) y با x وابستگی تابعی داشته باشد.

۲) y با هیچ زیرمجموعه x وابستگی تابعی نداشته باشد.

مسئله: با توجه به R2 تحقیق کنید، آیا وابستگی تابعی کامل بین C و (A, B) برقرار است؟

R2

A	B	C
a_1	b_1	c_1
a_2	b_1	c_3
a_1	b_2	c_2

پاسخ:

وابستگی تابعی بین C و (A, B) برقرار است. $(A, B) \rightarrow C$

$A \rightarrow C$ برقرار نیست زیرا $((a_1, c_1), (a_1, c_2))$

$B \rightarrow C$ برقرار نیست زیرا $((b_1, c_1), (b_1, c_3))$

پس $(A, B) \Rightarrow C$

نکته: اگر x صفت ساده باشد و وابستگی تابعی بین x و y برقرار باشد$(x \rightarrow y)$، وابستگی تابعی کامل$(x \Rightarrow y)$ همواره برقرار است.

شرح آنومالی‌ها:

همانطورکه گفتیم آنومالی سه وجه دارد. اینک با درنظرگرفتن رابطه زیر توضیح می‌دهیم.

P#	Book#	Qty	City	Grade
P1	b_1	1000	c_1	40
P2	b_4	3000	c_2	60
P3	b_3	2000	c_1	40
P1	b_2	3000	c_1	40
P4	b_3	4000	c_3	50
P4	b_5	3500	c_3	50

۱- درج: در آنومالی ناشی از درج، تمام یا بخشی از کلید اصلی تعریف‌نشده و نامعین است. طبق یکی از قواعد عام بانک اطلاعاتی، مقدار کلید اصلی باید کاملا مشخص باشد. فرض کنید می‌خواهیم رکورد $<P_{10}, , 1000, c_4, 10>$ را درج کنیم. این درج امکان‌پذیر نیست زیرا مشخص نیست ناشر چه کتابی را منتشر کرده است.

۲- حذف: در آنومالی نوع دوم پس از انجام یک عمل، عوارض نامطلوب داریم.

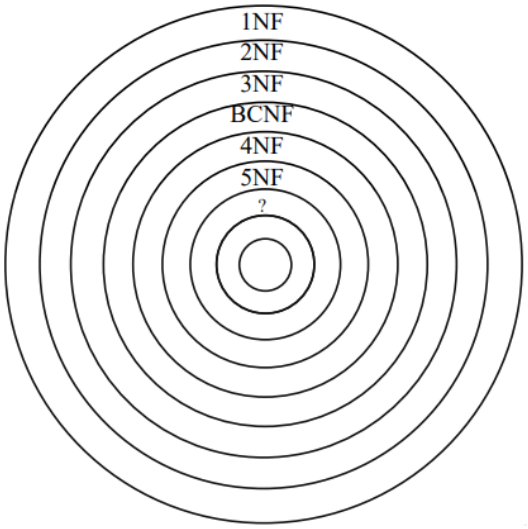
فرض کنید می‌خواهیم رکورد $<P_2, b_4, 3000>$ را حذف کنیم. این حذف اگرچه شدنی است اما رتبه صنفی ناشران شهر c_2 (مقدار ۶۰) ناخواسته حذف می‌شود (دقت کنید فعلاً تنها یک ناشر ساکن شهر c_2 است).

۳- به‌هنگام‌سازی: این نوع آنومالی، موجب فزونکاری می‌شود. درفزونکاری به ازای یک عمل مبنایی چندین عملیات صورت می‌گیرد. فرض کنید قرار است رتبه صنفی ناشران ساکن شهر c_3 از ۵۰ به ۷۰ تغییرکند. این عمل موجب به‌هنگام‌سازی منتشرشونده (فزون کاری) در سیستم می‌شود، یعنی در بیش از یک سطر جدول باید تغییر صورت گیرد.

همانطورکه مشاهده می‌شود، رابطه فوق دارای آنومالی است. در یک بیان غیردقیق، علت آنومالی‌ها، «اختلاط اطلاعاتی» است. به این معنی که اطلاعات نشر و اطلاعات پایه‌ای ناشر در یک رابطه(جدول) با هم جمع شده است. برای ایجاد یک طراحی مناسب از رابطه‌ها باید آنها را از لحاظ سطوح نرمال بررسی کنیم و در صورت نرمال نبودن در آن سطح، آن رابطه را با «تجزیه» اصلاح نماییم. در این صورت از بروز آنومالی جلوگیری می‌شود. اینک مناسب است سطوح نرمال رابطه را با بیان دقیق‌تر مورد بررسی قرار دهیم.

سطوح نرمال:

روابط از سطح غیرنرمال تا سطح نرمال ۵ قابل تبیین و بررسی است. سطوح نرمال عبارتند از:



۱- سطح نرمال اول 1NF (First Normal Form)

۲- سطح نرمال دوم 2NF

۳- سطح نرمال سوم 3NF

۴- سطح نرمال BCNF (به احترام نام Boyce و Codd)

۵- سطح نرمال چهارم4NF

۶- سطح نرمال پنجم 5NF

۱– سطح نرمال اول:

رابطه فوق را در نظر بگیرید. تمام فیلدها به نحوی هستند که مقادیر آنها به صورت منطقی قابل تقسیم نیست. به هر یک از این فیلدها، یک فیلد تک‌مقداری گوییم. اگر تمام فیلدها تک‌مقداری باشد، رابطه در سطح نرمال اول قرار دارد.

> رابطهٔ R در سطح نرمال اول(1NF) است، اگر:
> تمام فیلدهای آن در هر سطرجدول، تک‌مقداری باشد.

مسئله ۱: رابطه کارمند را در نظر بگیرید، آیا این رابطه در سطح 1NF است؟

کد پرسنلی	میزان تحصیلات	شماره شناسنامه	نام
۱۱۰	دیپلم	۲۴۳	علی علوی
۱۴۰	کارشناسی	۲۷۱۹	احمد احمدی
۱۳۰	کاردانی	۵۹۳	رضا رضوی
۱۷۰	کارشناسی	۹۰۹	اسماعیل اسماعیلی
۱۹۰	کارشناسی	۷۱۴	جواد جوادی

پاسخ: فیلد نام، قابل تجزیه به دو فیلد نام و نام خانوادگی است. اگر بخواهیم نام‌خانوادگی کارمندان را از این رابطه به دست آوریم، فیلد نام را تجزیه کرده‌ایم؛ بنابراین رابطه غیرنرمال است و در سطح 1NF قرار ندارد. شکل نرمال 1NF آن به صورت زیر است:

کد پرسنلی	میزان تحصیلات	شماره شناسنامه	نام خانوادگی	نام
۱۱۰	دیپلم	۲۴۳	علوی	علی

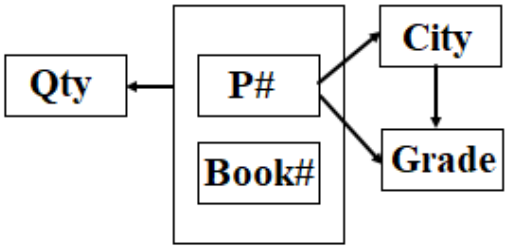
مسئله ۲: رابطه کتاب‌های موجود در یک کتابخانه را در نظر بگیرید، آیا این رابطه در 1NF قرار دارد؟

شماره کتاب	نام کتاب	اطلاعات نشر(ناشر – سال انتشار)
۱۲۰۳	برنامه نویسی C	نقش‌سیمرغ - ۱۳۸۸
۱۲۰۴	SQL Server	عابد – ۱۳۸۹
۱۲۰۶	Data Base	برگ زیتون - ۱۳۸۹

۲– سطح نرمال دوم:

درقسمت‌های قبل با تعریف وابستگی تابعی کامل آشنا شدیم. رابطه وقتی در سطح 1NF باقی می‌ماند که وابستگی تابعی کامل بین حداقل یک فیلد با کلید اصلی نقض شود.

به عنوان مثال نمودار مقابل را در نظر بگیرید:



کلید اصلی در این رابطه، صفت مرکب (P#, Book#) است. از آن-جایی که تمام فیلدها، با کلید وابستگی تابعی دارند، پس

(P#, Book#) → City

برای برقراری وابستگی تابعی کامل، City نباید به P# و Book# وابستگی تابعی داشته باشد.

اما همانطورکه در نمودار FD قابل مشاهده است، P# → City

بنابراین وابستگی تابعی کامل بین City و (P#, Book#) نقض شده است.

> رابطهٔ R در سطح نرمال دوم(2NF) است، اگر:
> اولاً: 1NF باشد.
> ثانیاً: تمام صفات غیرکلید، با کلید اصلی وابستگی تابعی کامل داشته باشند.

برای رفع آنومالی و افزایش سطح نرمال رابطه زیر، باید آنرا تجزیه کنیم.

FIRST (P#, Book#, Qty, City, Grade) → SECOND (P#, City, Grade), PB (P#, Book#, Qty)

FIRST

P#	Book#	Qty	City	Grade
P1	b_1	1000	c_1	40
P2	b_4	3000	c_2	60
P3	b_3	2000	c_1	40
P1	b_2	3000	c_1	40
P4	b_3	4000	c_3	50
P4	b_5	3500	c_3	50

→

SECOND

P#	City	Grade
P1	c_1	40
P2	c_2	60
P3	c_1	40
P4	c_3	50

,

PB

P#	Book#	Qty
P1	b_1	1000
P2	b_4	3000
P3	b_3	2000
P1	b_2	3000
P4	b_3	4000
P4	b_5	3500

ملاک تجزیهٔ رابطه:

تجزیه رابطه R به رابطه‌های R1 و R2 باید به نحوی باشد که پیوند دو رابطه R1 و R2 رابطه R را ایجاد کند و تاپلی (رکورد یا سطری) کم یا زیاد نشود. از طرف دیگر تجزیه R باید وابستگی‌های تابعی را حفظ کند.

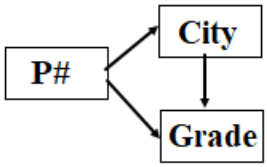
سوال: آیا معیارهای تجزیه مناسب در تجزیه FIRST لحاظ شده است؟

جواب: رابطه SECOND، در سطح دوم نرمال است. زیرا اولاً: 1NF است و ثانیاً: وابستگی تابعی کامل نقض نشده است. اما این رابطه نیز دارای آنومالی می‌باشد.

۳– سطح نرمال سوم:

رابطه SECOND را در نظر می‌گیریم. نمودار FD آن به صورت زیر است:

1. P# → City
2. City → Grade
3. P# → Grade



همان‌طورکه که قبلا دیدیم:

P# → City, City → Grade ==> P# → Grade

یعنی Grade وابستگی تابعی با P# از طریق City دارد. علت بروز آنومالی‌هایSECOND نیز همین عامل یعنی وابستگی تابعی باواسطه است.

> رابطهٔ R در سطح نرمال سوم(3NF) است، اگر:
> اولاً: 2NF باشد.
> ثانیاً: هر صفت غیرکلید، با کلید اصلی، وابستگی تابعی بی‌واسطه داشته باشد.

برای رفع آنومالی‌های آن، باید این رابطه تجزیه شود. بنابراین SECOND را به رابطه‌های PC و CG تجزیه می‌کنیم:

SECOND (P#, City, Grade) → PC(P#,City), CG (City, Grade)

SECOND

P#	City	Grade
P1	c_1	40
P2	c_2	60
P3	c_1	40
P4	c_3	50

→

PC

P#	City
P1	c_1
P2	c_2
P3	c_1
P1	c_3

,

CG

City	Grade
c_1	40
c_2	60
c_3	50

تمرین ۱: آیا می‌توان از وابستگی تابعی A → C عبارت (A, B) → C را نتیجه گرفت؟

تمرین ۲: رابطه Z را درنظر بگیرید و آن را به دو رابطهٔ X(A, B) وY(A, C) تجزیه کنید.آیا پیوند(JOIN) آنها رابطهٔ Z را ایجاد می‌کند؟

Z

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_1	c_3
a_3	b_3	c_4

تمرین ۳: در رابطهٔ R(A, B, C, D) وابستگی‌های تابعی C → D و A → C وجود دارد و (A, B) کلید است. سطح نرمال رابطه را مشخص کنید.

موفق و موید و سربلند باشید