۳: یادآوری تعریف داده

کلمه data ریشه لاتین دارد و در اصل از کلمه ای به معنای " دادن" مشتق می شود و مفرد آن datum است. در متون معمولا به صورت جمع به کار می رود.

- داده عبارتست از نمایش ذخیره شده اشیاء فیزیکی، چیزهای مجرد، بوده ها ، رویداد ها یا موجودیتهای دیگر قابل مشاهده که در تصمیم گیری به کار می آیند

۴ : یادآوری تعریف اطلاع

اطلاع (در لاتین : انفورماتیا) را آگاهی ، دانسته و خبر می‌گویند . این مفهوم بسته به زمینه مورد استفاده معنای خاص خود را دارد . مثلاً در دانش سایبرنتیک ، عامل (فاکتور) کیفی نشان دهنده وضعیت یک سیستم است که احتمالاً از خود سیستم به سیستم دیگری داده می‌شود .

در اصل کلمه "انفورماتیا" به معنای مجموعه بوده‌ها یا شده‌هاست که امکان می‌دهند تا عیب یا نقصی[1] آشکار شود و در پی آن عیبهای دیگری عیان شوند .

در متون کلاسیک دانش و فن کامپیوتر ، تعریف دقیق و جامع از مفهوم اطلاع ارائه نشده است (منابع زیادی مورد بررسی قرارگرفته‌اند و در بسیاری از آنها اساساً این مفهوم مطرح نشده است) . برخی از تعریفهای ارائه شده چنین‌اند :

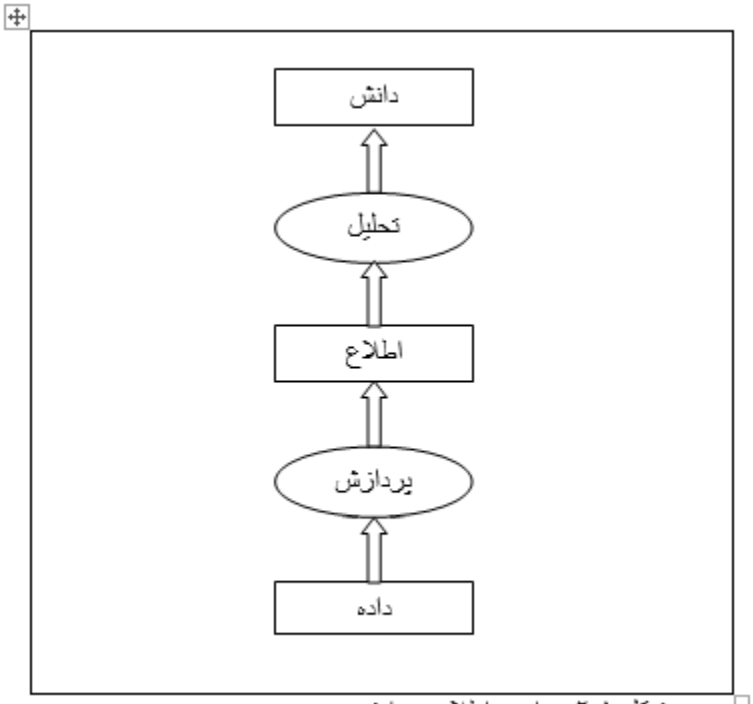
- اطلاع به داده‌ای اطلاق می‌شود که توسط یک فرد یا سازمان برای تصمیم‌گیری[2] به کار می‌رود [STAN 89] .
- اطلاع ، داده پردازش شده است [LIPS 92] .
- اطلاع عبارتست از داده آراسته به صورتی معنا دار [ROB 93] .
- اطلاع عبارتست از داده سازمان یافته که شناختی را منتقل می‌کند . به بیان دیگر اطلاع ، داده‌ای است که سازمان یافته است تا چیزی به ما بگوید [MODE 92] . و عبارات کم و بیش مشابه که به نظر چندان دقیق نمی‌رسند . اما ببینم ANSI در این مورد چه می‌گوید . برای این مفهوم ، ANSI تعریف زیر را ارائه کرده است :
- معنایی که انسان به داده منتسب می‌کند ، از طریق قراردادهای شناخته شده‌ای که در نمایش داده به کار می‌روند [روحا ۷۸- الف] .

در انفورماتیک می‌توان گفت که اطلاع مجموعه داده‌هایی است که در تصمیم‌گیری به کار می‌روند و اساساً کمیتی است نسبی و وابسته به وضعیت مشخص ، زمان مشخص و نیز خود شخص (یا سیستم) تصمیم گیرنده . در واقع عامل تصمیم‌گیرنده با تفسیر داده‌ها در یک وضعیت مشخص (در متن یک مسئله[3] مشخص) ، اطلاعات لازم برای تصمیم‌گیری را به دست می‌آورد .

۵: اشاره ای به مفهوم دانش

تعریف ما از دانش از معنایی که این اصطلاح در متون مربوط به سیستم های مدیریت پایگاه دانش دارد ناشی می شود: دانش عبارت است از نمایش نمادین جنبه هایی از بخشی از جهان واقع (Real world) (جهان مورد نظر Universe of discourse (UoD) یا محیط مطرح). آن گاه با توجه به این تعریف، یک "تکه دانش" (Piece) را به مثابه مدلی از جنبه ای از بخشی از جهان واقع می بیند. جهان واقع می تواند جهان واقعی یا غیر واقعی مثل آینده، جهان تخیلی و ... باشد. سپس مثال هایی از تکه دانش ارائه می کند از جمله:

- "الف" با "ب" ازدواج کرده است.
- "الف" کارمند شرکت "ش" است.
- کارمند "ج" فکر میکند که زندگی خوبی دارد.



در شکل ۱-۲ : داده ، اطلاع و دانش

۷- پایگاه داده‌ها در اساس چیزی بیش از مجموعه‌ای از اطلاعات نیست که مدت زمان طولانی ، حتی چندین سال ، می‌تواند وجود داشته باشد . در بیان عام ، اصطلاح پایگاه داده‌ها به مجموعه‌ای از داده‌ها اطلاق می‌شود که توسط یک سیستم مدیریت پایگاه داده‌ها ، مدیریت می‌شود [ULLM 02] .

۸- پایگاه داده‌ها مجموعه‌ای است از داده‌ها ، نوعاً نشان دهندهٔ (توصیفگر) فعالیتهای یک یا چند سازمان بهم مرتبط [RAMA 98] .

و جملاتی از این دست ... .

ما در اینجا تعریفی ارائه می‌کنیم که به نظر می‌رسد جامع تعاریف موجود در متون آکادمیک و تکنیک باشد . البته فعلاً نمی‌خواهیم در ظرایف بحث وارد بشویم . در واقع در تعریف پایگاه داده‌ها ابتدا باید محیطی که این مفهوم در آن مطرح می‌شود را مشخص کنیم . آیا در محیطی انتزاعی[۱] (که خود می‌تواند سطوحی داشته باشد) این مفهوم را تعریف می‌کنیم؟ آیا در محیط فایلینگ منطقی هستیم یا در محیط فایلینگ فیزیکی؟ مثلاً اگر در محیط فایلینگ منطقی باشیم ، این تعریف که : پایگاه داده‌ها مجموعه‌ای از فایلهای منطقی بهم مرتبط است ، طبعاً تا حدی پذیرفتنی است . اما خواهیم دید که آنچه که اساسی است ، **درک این مفهوم در محیط انتزاعی** است که در گفتارهای سوم و چهارم به آن خواهیم پرداخت .

۱-۱ : تعریف پایگاه داده‌ها

مجموعه‌ای است از داده‌های ذخیره شده[۲] و پایا ، به صورت مجتمع (یکپارچه)[۳] (نه لزوماً همیشه بطور فیزیکی ، بلکه حداقل بطور منطقی) ، بهم مرتبط ، حتی‌الامکان با کمترین افزونگی[۴]، (دارای یک ساختار منطقی مبتنی بر یک مدل داده[۵] ، توصیف شده بویژه در چارچوب همان مدل داده و نیز دارای معماری خاص) ، تحت مدیریت یک سیستم کنترل متمرکز[۶] ، مورد استفاده یک یا چند کاربر از یک (یا بیش از یک) "سیستم کاربردی" ، بطور همروند[۷] و اشتراکی[۸].

با توجه به این تعریف می‌توان دریافت که از دیدگاه تخصصی هر "مجموعه‌ای از فایلها" لزوماً پایگاه داده‌ها نیست .

در تعریف ارائه شده مفاهیمی وجود دارندکه باید به درستی درک شوند . برای درک برخی از این مفاهیم مثالی مقدماتی قید می‌کنیم . (توضیح صفت "پایا" را در گفتار سوم خواهیم دید) .

■ مثال ۱ (مقدماتی)

یک محیط عملیاتی یا بخشی از جهان واقع موسوم به خرد جهان واقع[1] (یا محیط مطرح) را درنظر می‌گیریم : **دانشگاه** . همه ما کم و بیش با این محیط آشنایی داریم .

می‌دانیم که هر محیط می‌تواند از تعدادی زیرمحیط تشکیل بشود . در دانشگاه هم قسمتهای مختلفی وجود دارد . هر قسمتی نیازهای اطلاعاتی و پردازشی خاص خود را دارد و در هر قسمتی ، مجموعه‌ای از **نوع موجودیت[2]ها** (یا به تعبیری **نوعِ شیء**) وجود دارند . (فعلاً مفهوم نوع موجودیت را در همان حدی مطرح می‌کنیم که در درس ذخیره و بازیابی اطلاعات دیدیم [روحا ۸۶ - الف] . در گفتار سوم این مفهوم را به گستردگی بررسی می‌کنیم) .

قسمتهای مختلف یک محیط می‌توانند نوع موجودیتهای مشترک داشته باشد . ما در این مثال مقدماتی سه قسمت از دانشگاه راکه در همه آنها نوع موجودیت دانشجو مطرح است ، در نظر می‌گیریم :

- اداره امور آموزش
- اداره امور فارغ التحصیلان
- اداره امور دانشجویی

توجه داریم که در هریک از این سه قسمت به اطلاعاتی در مورد نوع موجودیت دانشجو نیاز است ، البته حجم (کمیت) و صورت (فرم) و نیز در مواردی جنبه‌های کیفی دیگر اطلاعات می‌توانند در این سه قسمت متفاوت باشند . می‌خواهیم فعالیتهای این سه قسمت را "کامپیوتری" کنیم . برای این منظور باید یک یا چند "سیستم کاربردی[3]" طراحی و تولید کنیم .

برای ایجاد این سیستم (ها) در اساس دو رهیافت (رویکرد) یا مشی، وجود دارد :

- رهیافت (مشی) سنتی[4] یا فایلینگ (ناپایگاهی[5])
- رهیافت (مشی) پایگاهی[6]

مقدمه

داده‌هاي ذخيره شـده در پايگاه اصـطلاحاً به **داده‌هاي عملياتي** موسـومند: داده‌هاي عملياتي داده‌هايي هسـتند كه كاربران، روزانه با آنها سروكار دارند. اين داده‌ها **پايا** هستند، البته نه به اين معنا كه تغيير نمي كنند. منظور از پايا يعني اولاً: داده‌ها بعد از اجراي برنامه كاربر، در سيستم ماندگار هستند و ثانياً تا زماني كه كاربر مجاز، درخواست تغيير يا حذف آنها را به سيستم ندهد، بي تغيير مي‌مانند.

داده‌هاي ذخيره شـده در پايگاه داده‌ها، با داده‌هاي ورودي/خروجي تفاوت دارند. داده‌هاي ورودي به نحوي در پايگاه ذخيره مي‌شوند ولي لزوماً هر داده ذخيره شده در سيستم پايگاهي، داده ورودي نيست و نيز هر داده عرضه شده در خروجي به كاربر، ذخيره شده در پايگاه داده‌ها نيست، ممكن است مثلاً با محاسبه اي توليد و در اختيار كاربر گذاشته شود(به مطلب شـماره ۵ از قسـمت ۲-۲-۱ مراجعه شـود). بطور كلي مي‌توان گفت كه داده‌هاي ذخيره شـده در پايگاه داده‌ها با هر داده ماهيتاً **ناپايا** فرق دارند [DATE 03]

داده‌هاي ذخيره شدني در پايگاه داده‌ها ابتدا بايد در **بالاترين سطح انتزاع**، مدلسازي معنايي شوند.

مدلسازي معنايي داده‌ها يعني ارائه مدلي از داده‌هاي محيط عملياتي **(Operational environment)** (يا خرد جهان واقع: بخشي از يك محيط واقعي شامل واقعيات يا: بوده‌ها، شده‌ها، هست‌ها) به كمك مجموعه‌اي از **مفاهيم انتزاعي(** يعني مفاهيمي مستقل از جنبه‌هاي مربوط به نمايش منطقي و نمايش فيزيكي داده‌ها) و باتوجه به **معنايي** كه كاربر براي داده‌ها قائل است. به

مدلسازی معنایی گاه **طراحی ادراکی**[۱] (مفهومی) گفته می‌شود [CONN 05 و ELMA 03]، اما ما به دلیلی که در بحث معماری پایگاه داده‌ها (گفتار پنجم) خواهیم دید، این عبارت را برای مدلسازی به کار نمی‌بریم. در واقع، مرحله مدل‌سازی معنایی داده‌ها را می‌توان **مرحله پیش‌طراحی**[۲] نامید.

برای مدلسازی معنایی داده‌ها روشهایی وجود دارد. از روشهای کلاسیک رایج در مدلسازی معنایی داده‌ها روش موسوم به موجودیت - ارتباط (ER[۳]) یا مدل موجودیت - ارتباط است [ELMA 03، DATE 03، CONN 05 و SILB 06] که توسط چن ابداع شد و موضوع بحث ما همین روش است. البته روشهای دیگری هم اخیراً مطرح شده‌اند از جمله روش موسوم به "زبان مدلسازی یگانه[۴]" و روش موسوم به "تکنیک مدلسازی شیء[۵]" [GARD 99]. در روش "زبان مدلسازی یگانه"، خصوصیات هر شیء، ارتباط بین اشیاء، رفتار اشیاء و ... به کمک نمودارهایی، از جمله نمودار رده[۶] نمایش داده می‌شوند. بدیهی است برای استفاده از این روشها، شناخت کافی از مفاهیم شیءگرایی لازم است. اما باید توجه داشته باشیم که این روش عمدتاً برای استفاده در طراحی سیستمهای نرم‌افزاری ایجاد شده است، هر چند بخشی از فرایند طراحی نرم افزار، طراحی پایگاه داده‌هایی است که مورد دستیابی واحدهای نرم افزاری قرار می‌گیرند (پس می‌توان مدلسازی داده‌ها را با این روش هم انجام داد) (به قسمت ۵ از این گفتار مراجعه شود).

کنجکاوی ۱: روشهای دیگر مدلسازی معنایی کدامند؟

۲: مدلسازی با روش ER

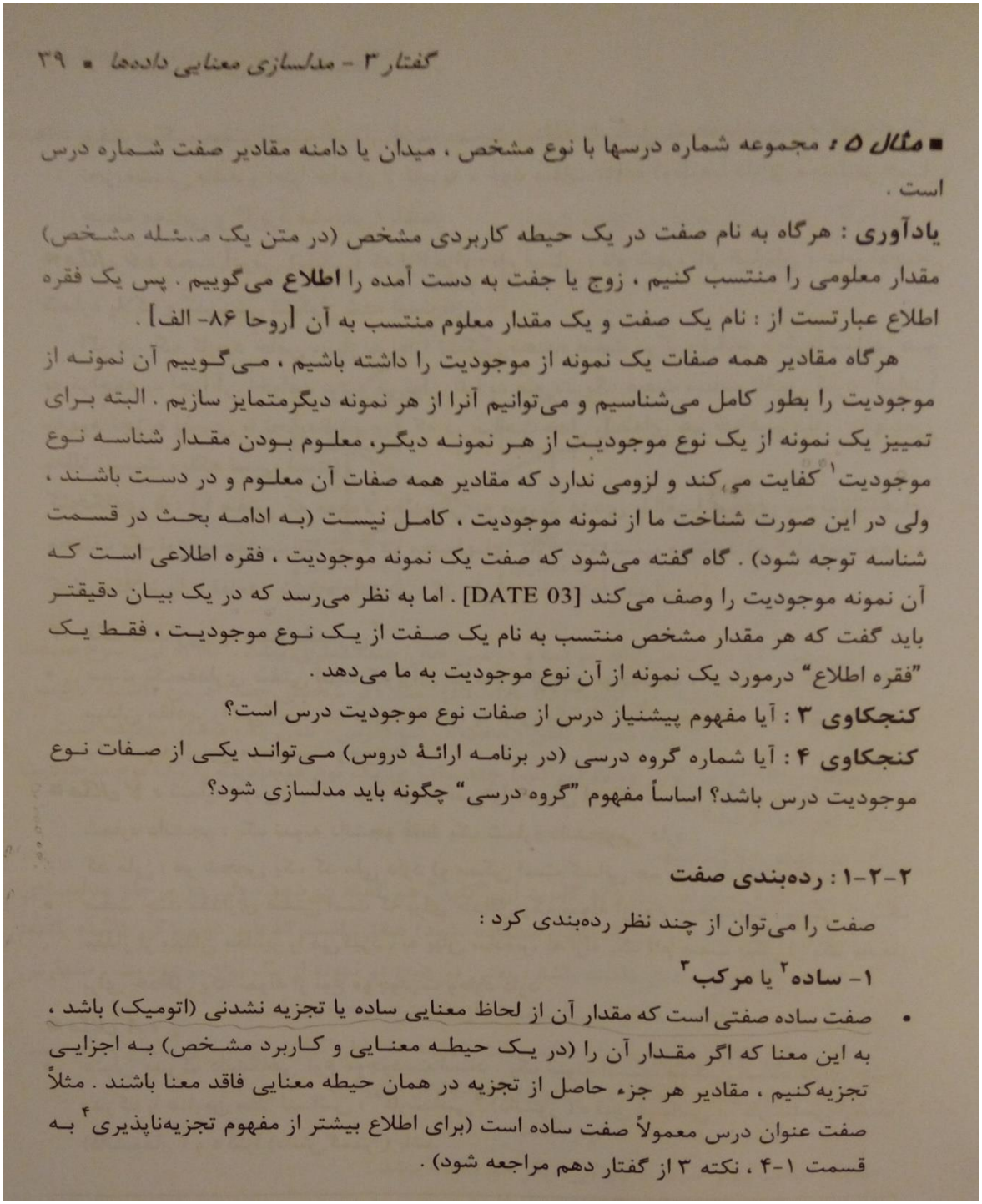
در روش ER، سه مفهوم معنایی[۷] وجود دارد و داده‌های هر محیطی به کمک همین سه مفهوم نمایش داده می‌شوند:

- نوع موجودیت (و شاید درست تر باشد بگوییم نوع موجود یا "هست").
- صفت[۸] (خصیصه)

۲-۱- نوع موجودیت

هر آنچه که درباره آن اطلاعات در سیستم نرم افزاری ذخیره می شود را **موجودیت** گویند. مانند دانشجو، کالا، درس، استاد و ...

توجه : در مدلسازی معنایی داده ها با روش ER، گاه تشخیص نوع موجودیت های یک محیط دشوار است ، زیرا این کار یعنی تشخیص دقیق نوع موجودیت ها ، ظاهرا کاری است ساده نما ، در عمل پیچیدگی هایی دارد.



گفتار ۳ - مدلسازی معنایی داده‌ها ■ ۳۹

■ **مثال ۵ :** مجموعه شماره درسها با نوع مشخص ، میدان یا دامنه مقادیر صفت شماره درس است .

یادآوری : هرگاه به نام صفت در یک حیطه کاربردی مشخص (در متن یک مسئله مشخص) مقدار معلومی را منتسب کنیم ، زوج یا جفت به دست آمده را **اطلاع** می‌گوییم . پس یک فقره اطلاع عبارتست از : نام یک صفت و یک مقدار معلوم منتسب به آن [روحا ۸۶- الف] .

هرگاه مقادیر همه صفات یک نمونه از موجودیت را داشته باشیم ، می‌گوییم آن نمونه از موجودیت را بطور کامل می‌شناسیم و می‌توانیم آنرا از هر نمونه دیگرمتمایز سازیم . البته برای تمییز یک نمونه از یک نوع موجودیت از هر نمونه دیگر، معلوم بودن مقدار شناسه نوع موجودیت[1] کفایت می‌کند و لزومی ندارد که مقادیر همه صفات آن معلوم و در دست باشند ، ولی در این صورت شناخت ما از نمونه موجودیت ، کامل نیست (به ادامه بحث در قسمت شناسه توجه شود) . گاه گفته می‌شود که صفت یک نمونه موجودیت ، فقره اطلاعی است که آن نمونه موجودیت را وصف می‌کند [DATE 03] . اما به نظر می‌رسد که در یک بیان دقیقتر باید گفت که هر مقدار مشخص منتسب به نام یک صفت از یک نوع موجودیت ، فقط یک "فقره اطلاع" درمورد یک نمونه از آن نوع موجودیت به ما می‌دهد .

کنجکاوی ۳ : آیا مفهوم پیشنیاز درس از صفات نوع موجودیت درس است؟

کنجکاوی ۴ : آیا شماره گروه درسی (در برنامه ارائهٔ دروس) می‌تواند یکی از صفات نوع موجودیت درس باشد؟ اساساً مفهوم "گروه درسی" چگونه باید مدلسازی شود؟

۲-۲-۱ : رده‌بندی صفت

صفت را می‌توان از چند نظر رده‌بندی کرد :

۱- ساده[2] یا مرکب[3]

- صفت ساده صفتی است که مقدار آن از لحاظ معنایی ساده یا تجزیه نشدنی (اتومیک) باشد ، به این معنا که اگر مقدار آن را (در یک حیطه معنایی و کاربرد مشخص) به اجزایی تجزیه‌کنیم ، مقادیر هر جزء حاصل از تجزیه در همان حیطه معنایی فاقد معنا باشند . مثلاً صفت عنوان درس معمولاً صفت ساده است (برای اطلاع بیشتر از مفهوم تجزیه‌ناپذیری[4] به قسمت ۱-۴ ، نکته ۳ از گفتار دهم مراجعه شود) .

- صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه‌ای که تجزیه‌شدنی باشد و اجزا حاصل از تجزیه ، خود صفات ساده (وطبعاً دارای معنا در همان حیطه معنایی و کاربرد مشخص) باشند .

■ ***مثال ۶ :*** صفت آدرس (نشانی) که از اجزاء : نام استان ، نام شهر، نام خیابان ، نام کوچه ، شماره پلاک و کدپستی تشکیل شده است .

اگر در یک کاربرد خاص ، نیاز به اجزا تشکیل دهنده صفت مرکب نباشد ، طبعاً نیازی هم به مراجعه به اجزا آن نخواهد بود و می‌توان آنرا به صورت یک صفت ساده تلقی کرد (اساساً مفهوم تجزیه پذیری یا تجزیه‌ناپذیری ، که در مباحث مدل رابطه‌ای هم خواهیم دید ، مفهومی مطلق نیست ، بلکه نسبی است) .

کنجکاوی ۵ : آیا صفتی که معمولاً مقادیرش به صورت عددی یا الفبا عددی به نمونه‌های مختلف یک نوع موجودیت یا شیء در محیطهای کاری ، منتسب می‌شود (و گاه موسوم به کد) ، مثلاً شماره قطعه ، شماره دانشجو ، کد ملی و ... صفت ساده است؟

۲- تک مقداری[1] یا چند مقداری[2]

- صفت تک‌مقداری صفتی است که برای هر نمونه از نوع موجودیت ، حداکثر یک مقدار از میدان مقادیر را می‌گیرد ؛ به بیان ساده‌تر، به ازای یک نام صفت حداکثر یک مقدار برای یک نمونه از نوع موجودیت وجود دارد .

■ ***مثال ۷ :*** شماره درس : یک نمونه درس نمی‌تواند بیش از یک شماره داشته باشد .
شماره دانشجو : یک نمونه دانشجو فقط یک شماره دانشجویی دارد .
کد ملی : هر شخص یک کد ملی دارد (و ممکن است کسانی هم هنوز نداشته باشند) .

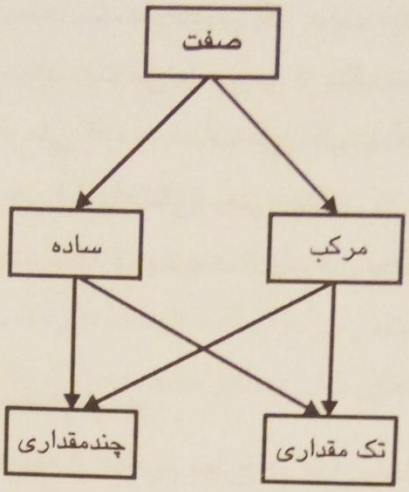
- صفت چند مقداری صفتی است که برای حداقل یک نمونه از نوع موجودیت ، بیش از یک مقدار از میدان مقادیر را می‌گیرد . به بیان ساده‌تر، به ازاء یک نام صفت بیش از یک مقدار برای حداقل یک نمونه از نوع موجودیت وجود دارد .

■ ***مثال ۸ :***

- عنوان مدرک دانشگاهی نوع موجودیت استاد : یک نمونه استاد ممکن است دارای چند مدرک با عناوین مثلاً لیسانس : "کارشناسی" (دانشور*) ، فوق لیسانس : "کارشناسی ارشد" (دانشمدار*) و دکترا (دانش‌گستر*) باشد .

- شماره تلفن نوع موجودیت دانشکده : یک نمونه دانشکده می‌تواند چند شماره تلفن داشته باشد .

نکته ۳ : به نمودار ساده زیر توجه شود :



می‌بینیم که صفت می‌تواند مرکب و درعین حال چندمقداری باشد . گاه به این نوع صفت اصطلاحاً صفت پیچیده[1] می‌گویند [ELMA 03] . مثلاً اگر شخصی چند آدرس داشته باشد و در هر آدرس ، بیش از یک شماره تلفن ؛ صفت مرکب آدرس- تلفن یک صفت پیچیده است .

کنجکاوی ۶ : آیا می‌توان صفت (مرکب) چند مقداری یک نوع موجودیت را یک نوع موجودیت دیگر در نظر گرفت؟

۳- شناسه یا ناشناسه

- صفت شناسه موجودیت (که گاه به آن "کلید" هم گفته می‌شود ، ولی در مرحله مدلسازی معنایی بهتر است از این اصطلاح استفاده نشود) صفتی است که باید دو ویژگی داشته باشد:

الف- یکتایی[2] مقدار داشته باشد : یعنی در هیچ دو نمونه از یک نوع موجودیت ، مقدارش یکسان نباشد . بنابراین عامل تمییز دونمونه از یک نوع موجودیت است .

از ویژگی (الف) نتیجه می‌شود که صفت شناسه ، از نظر کاربر، شناسای[3] نوع موجودیت است و متمایز کننده نمونه‌های آن نوع موجودیت از یکدیگر .

ب- مقادیرش همیشه **معلوم** باشند .

صفت شناسه بهتر است دو ویژگی دیگر هم داشته باشد :

پ: طول نمایش مقادیرش حتی الامکان کوتاه باشد.
ت: مقدار حتی الامکان تغییر نکند .
اما این دو ویژگی، ویژگی های ذاتی صفت شناسه نیستند.
هر نوع موجودیت اگر(ضعیف نباشد) حداقل یک صفت شناسه دارد، حتی اگر جز صفات طبیعی آن نباشد.پس نوع موجودیت می تواند بیش از یک شناسه داشته باشد، مثلا شماره کارگزینی و کد ملی کارمند شناسه های کارمند هستند.

٤ -هیچمقدار پذیر یا هیچمقدار ناپذیر
هیچمقدار چیست؟ این مفهوم از مفاهیم مطرح در مدل رابطه ای است. در اینجا صرفنظر از مطالب نظری و گاه ظریف بیان شده درباره این مفهوم، فعلاً به کوتاهی آن را معرفی می کنیم :
هیچمقدار یعنی مقدار ناشناخته به هر دلیل اعم از اینکه ناموجود، غیر قابل اعمال یا تعریف نشده باشد. گذشته ازجزئیات، می توان گفت معادل است. با همان مفهوم " داده نهست " و یا به تسامح " اطلاع نهست "{ روحا ٨٦- الف} .
با این مفهوم در عمل بسیار برخورد می کنیم.زیرا همیشه ممکن است مقدار حداقل یک صفت برای از نمونه های یک نوع موجودیت ، ناشناخته باشد.
 مثال ٩:
* شماره کارگزینی یک نمونه کارمند ممکن است هنوز، ازطرف اداره کارگزینی، (به هر دلیلی) تعیین نشده باشد.
*شماره تلفن یک نمونه استاد در دست نباشد.
*یک نمونه درس ممکن است به ملزومات ویژه ای نیاز نداشته باشد.
*نام استاد یک درس در برنامه درسی ترممکن است هنوز اعلام نشده باشد.
* نام سخنران یکی از سخنرانی ها(مثلا در یک سمینار یا کنفرانس) بعداً اعلام می شود.

حال می گوییم اگر مقدار یک صفت در یک یا بیش از یک نمونه از یک نوع موجودیت ، برابر با "هیچمقدار" باشد، آن .صفت هیچمقدارپذیر است
صفت شناسه نوع موجودیت نمی تواند هیچمقدار پذیر باشد ،زیرا عامل تمییز نمونه های نوع موجودیت از یکدیگر است و .عامل تمییز، خود نمیتواند ناشناخته باشد
.برای اطلاع بیشتر از مفهوم هیجمقدار، به قسمت ٨-٢-٢-١-١ از گفتار دهم رجوع شود
Detived یا مشتق (Base attribute یا مبنا Real واقعی) (Stored)ذخیره شده -٥
- صفت ذخیره شده صفتی است که مقادیرش در پایگاه داده ها ذخیره شده باشند و البته میتواند هیچمقدار هم داشته . باشد ، اگر شناسه نباشد
- صفت مشتق صفتی است که مقادیرش در پایگاه داده ها ذخیره شده نباشد بلکه حاصل یک پردازش روی فقره .هایی از داده های ذخیره شده باشند. مثلا از یک محاسبه بدست آیند
:مثال ١٠
- صفت معدل دانشجو در یک ترم
- صفت کل ساعات اضافه تدریس یک استاد در یک ترم
- صفت میانگین نمرات دانشجویان در یک درس

.نیز میگوئیم Calculated atribute یا صفت محاسبه شده Vitual atribute به صفت مشتق گاه صفت مجازی
نکته ٤: توجه داشته باشیم که صرف اینکه صفتی محاسبه شدنی باشد ، به این معنا نیست که همیشه باید بصورت صفت .واقعی درنظر بگیرد
مدلساز باید پس از تعیین مجموعه صفات هرنوع موجودیت، وضع هر صفت را ،با توجه به رده بندی صفات، مشخص .کند

۲-۳ :نوع ارتباط

نوع ارتباط یا بستگی مفهومی است بسیار مهم در مدل سازی مفهومی داده. بین نوع موجودیت ها، معمولا ارتباط نوع (ارتباط هایی) برقرار است.

۲-۳-۱ :تعریف نوع ارتباط

نوع ارتباط عبارت است از اندرکنش (تعامل) بین n نوع موجودیت=<n) ۱ (و ماهیتا نوعی بستگی بین موجودیت ها است. و نیز می توان گفت که نوع ارتباط عملی است که بین نوع موجودیت ها جاری بوده، هست و یا خواهد بود.

هر نوع ارتباط یک معنای مشخص دارد و یا یک نام بیان می شود.

مثال ۱۱: نوع موجودیت های دانشجو و درس را در نظر میگیریم. بین این نوع موجودیت ها چند نوع ارتباط وجود دارد از جمله:

- دانشجو درس را انتخاب کی کند
- دانجشو درس را حذف می کند
- دانشجو درس را قبول می شود

۲-۳-۲ : نمودار ER

دراینجا ، قبل از آنکه بحث درباره مفهوم مهم نوع ارتباط را ادامه دهیم ، با مفهوم نمودار ER۱ آشنا می شویم :

نموداری است که درآن سه مفهوم اساسی مدل ER یعنی نوع موجودیت ، صفت و نوع ارتباط نمایش داده می شوند .

درواقع این نمودار ، امکانی است برای نمایش مدلسازی و اولین طرح پایگاه داده ها در بالاترین سطح انتزاع . برای رسم این نمودار

به نمادهایی نیاز داریم . این نمادها (موسوم به نمادهای چن ۲.) درجدول ۳-۱ دیده می شوند (گاه از نمادهای دیگری موسوم به

نمادهای Rein۸۵ Crows foot , و IDEFIX هم استفاده می شود ROB]۰۰[ ) .

هرنمودار ER پاسخگوی مجموعه ای مشخص از نیازهای کاربران است . (نیازهایی که در مرحله مطالعه ، تحلیل

وشناخت محیط برآورد شده اند) . بنابراین متناسب با تغییر یا رشد نیازهای کاربران ، نمودار ER هم تغییر یا رشد می کند ، هر

چند بهتر است مدلسازی حتی الامکان از ابتدا به صورت جامع انجام شود . حال به کمک این نمادها بحث را با مثالهای لازم ادامه

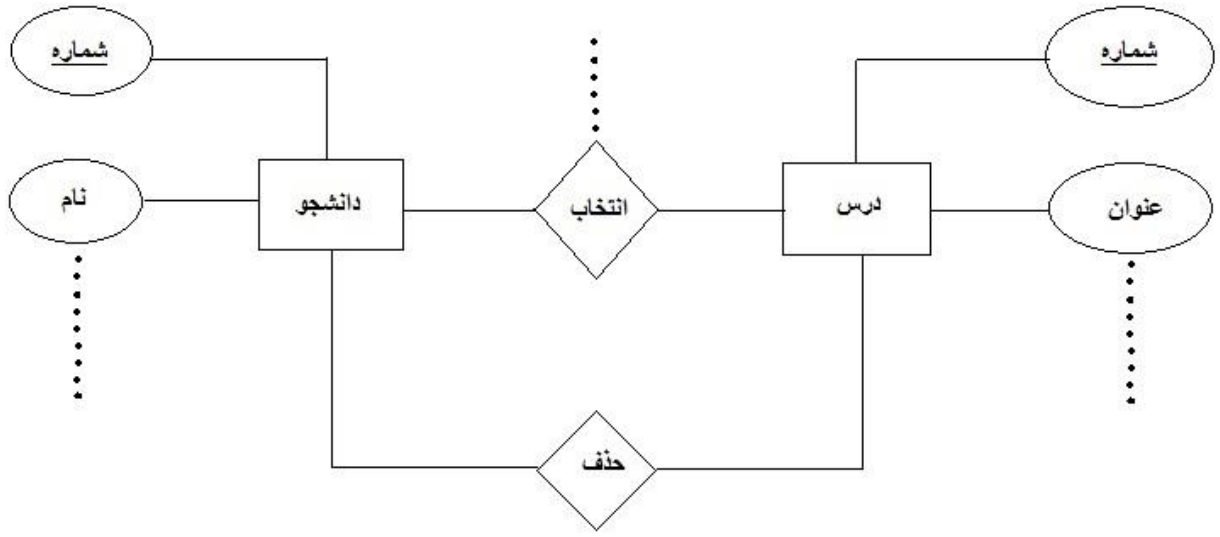
می دهیم :

- مثال ۱۲: درنمودار ER شکل ۳-۲ دونوع ارتباط بین موجودیتهای دانشجو و درس را می بینیم و چنین می خوانیم :

- " دانشجو درس را انتخاب می کند . "
- " درس توسط دانشجو انتخاب می شود . "
- " دانشجو درس را حذف می کند . "

- “درس توسط دانشجو حذف می شود . “

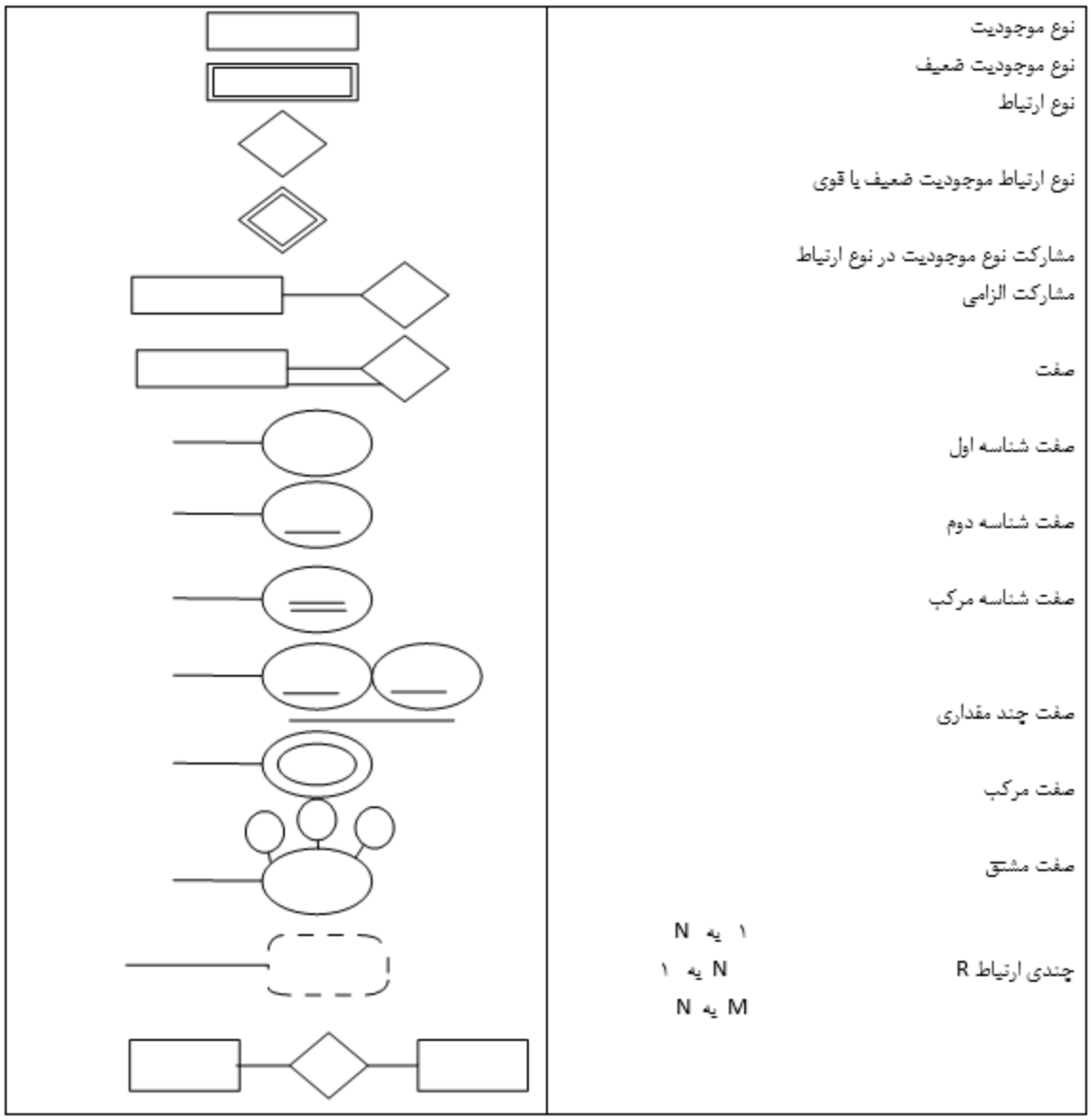
ضمنا زیر صفات شناسه خط می کشیم .



در همین مثال که بین مثلا دو نوع موجودیت، ممکن است بیش از یک نوع ارتباط وجود داشته باشد. پس اگر N تعداد نوع ارتباط بین دو نوع موجودیت باشد< داریم : $0 \leq N$.

جدول ۳ – ۱ : نمادهای * رسم نمودار ER :

- توجه: فقط اشکال تدریس شده در کلاس در سوالات امتحانی خواسته می شوند.



۲- ۳- ۳ : خصوصیات نوع ارتباط

۲- ۳- ۳- ۱ : خصوصیات کلی

الف : هر نوع ارتباط یک **نام** دارد : معمولا کلمه یا عبارتی فعلی و نه اسمی (در متون خارجی گاه چنین است و گاه نوع ارتباط را با یک کوته نوشت که حاصل ترکیب حرف اول نام نوع موجودیتها است ، نشان می دهند) .

ب : هر نوع ارتباط یک **معنای مشخص** دارد و این معنا با معنای هر نوع ارتباط دیگر متفاوت است . معنای نوع ارتباط " انتخاب " (کردن و شدن) با معنای نوع ارتباط " حذف " (کردن و شدن) طبعا فرق دارد .

پ : هر نوع ارتباط **نمونه** هایی دارد . مثلا " دانشجو (s) درس (c) را انتخاب کرده است " یک نمونه از ارتباط " انتخاب " است .

نوع ارتباط خصوصیات دیگری هم دارد که در ادامه خواهیم دید .

۲- ۳- ۳- ۲ : وضع مشارکت در ارتباط

نوع موجودیتهایی که بین آنها یک نوع ارتباط برقرار است ، **شرکت کنندگان** در آن نوع ارتباط نام دارند . مثلا در نوع ارتباط انتخاب : دو نوع موجودیت مشارکت دارند ، هر یک را شرکت کننده می گوییم .
مشارکت یک نوع موجودیت در یک نوع ارتباط ممکن است **الزامی** (کامل) یا **غیر الزامی** (نا کامل) باشد .
مشارکت نوع موجودیت E در نوع ارتباط R را الزامی می گوییم اگر تمام نمونه های E در R شرکت کنند . در غیر اینصورت مشارکت غیر الزامی است .
الزامی بودن مشارکت در نوع ارتباط ، از محدودیتهای معنایی محیط است و خواهیم دید که در طرز طراحی پایگاه داده ها تاثیر دارد.
. مثال ۱۳ : مشارکت نوع موجودیت دانشجو در نوع ارتباط " انتخاب " الزامی اسن . توجه داریم که منظور ما در اینجا از کلمه دانشجو ، دانشجوی فعال است و نه دانشجوی به هر دلیل ثبت نام نکرده . حال اگر دانشجویان " مرخصی گرفته " یا " غایب بی اطلاع " در ترم را

- **مثال ۱۴:** مشارکت نوع موجودیت دانشجو و نوع موجودیت درس در نوع ارتباط "حذف" الزامی نیست، زیرا لزوماً همه دانشجویان درس (درس هایی) را حذف نمی کنند و نیز ممکن است درس یا درس هایی اصلاً توسط دانشجویان حذف نشود.

اگر فرض کنیم که مشارکت دانشجو و درس در نوع ارتباط "انتخاب" الزامی باشد، نمایش آن به صورتی است که در شکل ۳-۳ دیده می شود.

شکل ۳-۳ الزامی

گفتیم که در هر نوع ارتباط، حداقل یک نوع موجودیت مشارکت دارد. هر نوع موجودیت شرکت کننده در یک نوع ارتباط، یک نقش مشخص را ایفا می کند که معمولاً با یک عبارت فعلی بیان می شود. این نقش در واقع همان تاثیر عملی است که انجام می شود.
از نظر تئوریک ممکن است یک نوع موجودیت، اصلاً در هیچ نوع ارتباطی مشارکت نداشته باشد، نوع موجودیت **"ایزوله"** باشد، در عمل اما معمولاً چنین نیست.

- **مثال ۱۵:** نقش نوع موجودیت دانشجو در نوع ارتباط انتخاب: "انتخاب کردن" است و نقش موجودیت درس در این نوع ارتباط: "انتخاب شدن". معمولاً با توجه به معنای یک نوع ارتباط به آسانی می توان به نقش شرکت کنندگان در آن پی برد.

۳-۳-۳-۲: صفت نوع ارتباط

نوع ارتباط می تواند صفت (صفاتی) داشته باشد. به عنوان مثال:
نمونه ارتباط زیر را در نظر می گیریم (از مثال ۱۴):
"دانشجو "S" درس "C" را انتخاب کرده است. این نمونه ارتباط، اطلاعاتی در مورد یک واقعیت به ما می دهد. اما معلوم نیست که دانشجو "S" درس "C" را در کدام ترم - سال آموزشی

انتخاب کرده است . در واقع گاه لازم می‌شود از نمونه‌های یک نوع ارتباط ، اطلاعات بیشتری به دست آوریم ، به بیان دیگر در مورد واقعیتی که یک نمونه ارتباط نمایشگر آن است ، اطلاعات دیگری هم داشته باشیم (علاوه بر شناسه شرکت‌کنندگان در نوع ارتباط) . برای پاسخ دادن به این نیاز ، باید برای نوع ارتباط ، صفت یا صفات لازم را در نظر بگیریم . گاه به صفت نوع ارتباط، صفت توصیفی[1] می‌گویند .

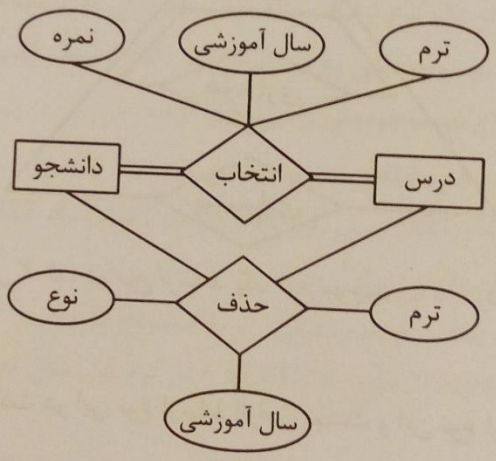
■ ***مثال ۱۶ :*** برای نوع ارتباط "انتخاب" ، مثلاً می‌توان صفات ترم ، سال آموزشی و نمره و برای نوع ارتباط "حذف" ، صفات ترم ، سال آموزشی و نوع حذف را در نظر گرفت (شکل ۳-۴) .

در این مثال صفات ترم و سال دو صفت جدا در نظر گرفته شده‌اند .

کنجکاوی ۱۰ : اگر دو صفت ترم و سال را یک صفت بگیریم مثلاً به صورت YYT (که در آن YY : سال تحصیلی و T : ترم است) ، آیا کار درستی است؟

کنجکاوی ۱۱ : آیا دو نوع ارتباط می‌توانند مجموعه صفات یکسان داشته باشند؟

کنجکاوی ۱۲ : آیا صفت نوع ارتباط می‌تواند ساده چندمقداری و یا مرکب چندمقداری باشد؟

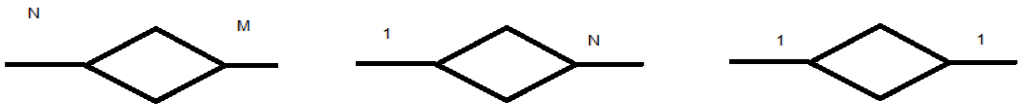


شکل ۳-۴ : نوع ارتباط می‌تواند صفت (صفات) داشته باشد

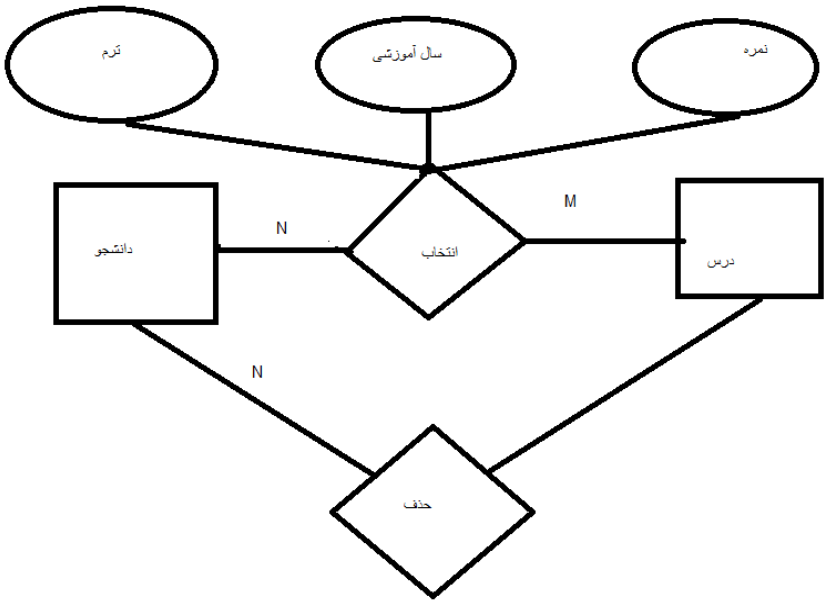
۲-۳-۳-۴: درجه[2] نوع ارتباط

تعداد شرکت‌کنندگان در یک نوع ارتباط را **درجه** آن گوییم و اصطلاحات زیر را داریم :

۵-۳-۳-۲ : چندی (multiplicity) یا ماهیت (nature) نوع ارتباط
چندی یک نوع ارتباط مثلا بین دو نوع موجودیت که به آن ماهیت نوع ارتباط یا نسبت کاردینالیتی elma 03 هم می گویند عبارتست از چگونگی تناظر بین دو مجموعه نمونه های آن دو نوع موجودیت . می دانیم که سه گونه تناظر داریم : تناظر یک به یک
One to one تناظر یک به چند one to many و تناظر چند به چند many to many چندی نوع ارتباط را چنین نمایش می دهیم :



به بیان دیگر اگر دو نوع موجودیت e , f را در نظر بگیریم در اینصورت :
در نوع ارتباط با چندی یک به یک e به f یک نمونه از e حداکثر با یک نمونه از f ارتباط دارد و برعکس .
در نوع ارتباط با چندی یک به چند از سوی e به f یک نمونه e به n نمونه از f (n>1 و در صورت مشارکت غیر الزامی n=0 ارتباط دارد ولی یک نمونه از f حداکثر با یک نمونه از e ارتباط دارد .
در نوع ارتباط با چندی چند به چند یک نمونه از e با n نمونه از f(n>1) ارتیباط دارد .و بر عکس .
مثال ۲۰:
چندی نوع ارتباط انتخاب در حالت کلی n:m است
چندی نوع ارتباط حذف تک درس معمولا 1=n است یعنی یک دانشجو یک درس را حذف می کند ولی یک درس ممکن است توسط چند دانشجو حذف شود .



گفتار ٤

آشنایی با ساختار داده‌ها

۱ : پایگاه داده‌ها در سطح انتزاعی مبتنی بر مدل داده

در گفتار سوم دیدیم که مدلسازی معنایی داده‌ها ، نمایشی است از پایگاه داده‌ها در یک محیط انتزاعی . این محیط فراتر است از محیط نمایش منطقی داده‌ها بر اساس یک مدل داده ، و طبعاً فراتر از محیط فایلینگ منطقی و فایلینگ فیزیکی . اما مفهوم محیط انتزاعی خود می‌تواند سطوحی داشته باشد . سطح پایین‌تر از سطح مدلسازی معنایی ، سطح طراحی منطقی است که با استفاده از یک ساختار داده ، (بخشی از مدل داده[1]) انجام می‌شود .

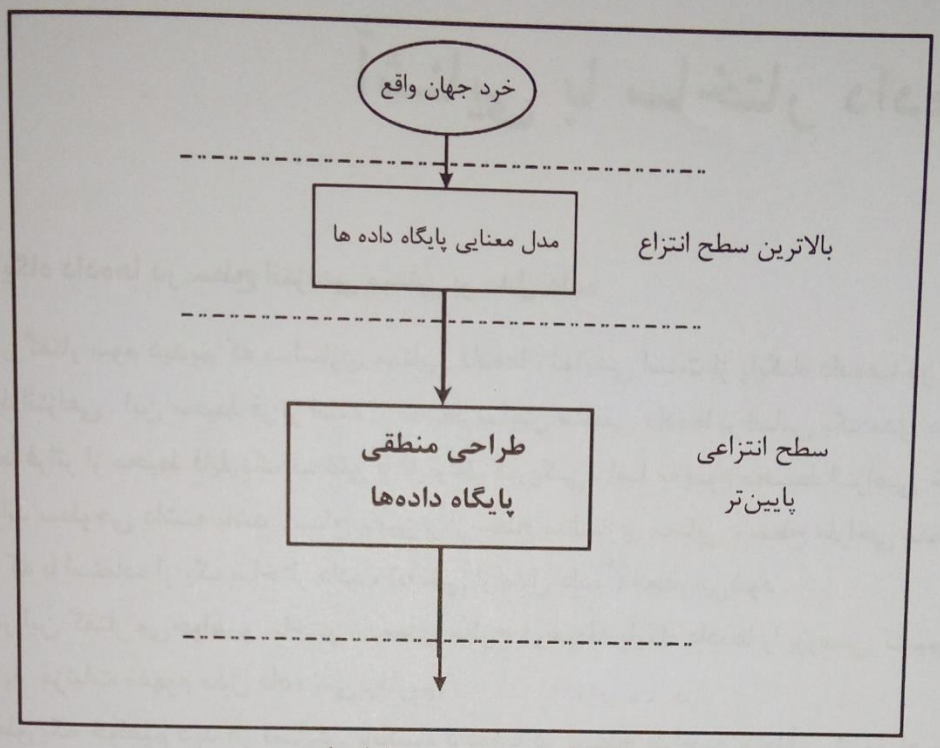
در این گفتار می‌خواهیم ساختار داده‌های مطرح درحیطه پایگاه داده‌ها را بررسی کنیم (و البته به جزئیات مفهوم مدل داده نمی‌پردازیم) .

بطوریکه خواهیم دید در نمایش پایگاه داده‌ها در سطح طراحی منطقی از مفاهیمی استفاده می‌شود (و یا بهتر است بگوییم باید استفاده شود) که مستقل از مفاهیم محیط فایلینگ پایگاه‌داده‌ها هستند . به بیان دیگر در این سطح از انتزاع به جنبه‌های مربوط به نمایش[2] نوع موجودیتها و نوع ارتباط‌های بین آنها، در چارچوب فایلینگ منطقی و جنبه‌های مربوط به پیاده‌سازی[3] فایلها در محیط فیزیکی ذخیره‌سازی نمی‌پردازیم . در شکل ۴-۱ سطوح مختلف محیط انتزاعی پایگاه داده‌ها دیده می‌شود .

همانطور که درمدلسازی داده‌ها نیاز به امکانی برای نمایش واقعیات داریم ، برای طراحی منطقی پایگاه داده‌ها هم امکان خاصی لازم است : یک **مدل داده** ، که شامل یک **ساختار داده** است (دیرتر خواهیم دید که ساختار داده در واقع بخشی از مفهوم گسترده‌تر مدل داده

1- Data model
2- Presentation
3- Implementation

است) . فعلاً با قدری تسامح می‌گوییم که همین ساختار داده از یک مدل داده است که تأمین کننده این سطح از انتزاع پایگاه داده‌ها و نیز چارچوب طراحی منطقی آن است . به بیان دیگر، و به شرحی که خواهیم دید ، طراحی منطقی پایگاه داده‌ها با استفاده از مفاهیم اساسی یک مدل داده و در چارچوب ساختار داده آن مدل انجام می‌شود .



شکل ۴-۱: سطوح محیط انتزاعی

در واقع پایگاه داده‌ها در سطح انتزاعی مبتنی بر مدل داده، خود یک **ساخت انتزاعی**[1] است . **ساختار داده (یا همان بخش ساختاری مدل داده) امکانی است برای نمایش نوع موجودیت‌ها و ارتباط‌های بین آنها در سطح طراحی منطقی پایگاه داده‌ها** .

نکته ۱ : هر ساختار داده حداقل یک **عنصر ساختاری اساسی**[2] دارد . عنصر ساختاری اساسی عنصری است که به کمک آن نوع موجودیت یا نوع ارتباط یا هر دو نمایش داده می‌شوند .

بنابراین می‌توان گفت که پایگاه داده‌ها در سطح انتزاعیِ مبتنی بر مدل داده (و پایین‌تر از سطح مدلسازی معنایی) ، مجموعه‌ای است از نمونه‌های متمایز عنصر

(عناصر) ساختاری اساسی یک ساختار داده (از یک مدل داده مشخص) . این مجموعه می‌تواند به همان معنای مجموعه ریاضی یا یک گردایه[1] باشد .

در این بحث فعلاً مفهوم بسیار مهم مدل داده را رها می‌کنیم و مطلب را با همان مفهوم ساختار داده ادامه می‌دهیم .

۲ : ساختار داده‌ها

در درس "ساختمان داده‌ها" با گونه‌هایی از ساختار داده‌ها آشنا شده‌ایم . اما در حیطه پایگاه داده‌ها مفهوم ساختار داده گونه‌های خاص خود را دارد . در عمل گونه‌های کلاسیک موجود ساختار داده عبارتند از :

- **ساختار داده رابطه‌ای[2]**
- **ساختار داده سلسله مراتبی[3]**
- **ساختار داده شبکه‌ای[4]**

توجه : به نظر می‌رسد دانشجوی درس پایگاه داده‌ها باید با اصول کلی (و نه جزئیات) مدل داده سلسله مراتبی و مدل داده شبکه‌ای هم تا حدی آشنا شود ، زیرا در حال حاضر درصد قابل توجهی از سیستمهای پایگاهی موجود در دنیا آنهایی هستند که با DBMSهای سلسله مراتبی و شبکه‌ای طراحی و ایجاد شده بودند . این سیستمها را اصطلاحاً **سیستمهای بازمانده[5]** **(گذشته)** می‌گویند . در حال حاضر یکی از مشکلات مهم در حیطه داده‌داری - داده‌پردازی ، چگونگی بهره‌برداری کارا از این سیستمها است ، بویژه مسئله تعامل سیستمهای پایگاهی جدید با آن سیستمها ، بسیار مطرح است . هنوز حدود ۸۵ درصد داده‌های عملیاتی سازمانها روی این سیستمهای سلسله مراتبی و شبکه‌ای و حتی روی سیستمهای فایلینگ ذخیره شده‌اند [DATE 03] . این آمار بسیار جالب توجه است (البته اینک ، سال ۱۳۹۰ ، این آمار لابد کمتر شده است) . بنابراین آشنایی با مفاهیم پایه‌ای این سیستمها ، مخصوصاً ساختار داده آنها ، می‌تواند سودمند باشد .

ساختار داده‌های دیگری (حداقل از دیدگاه نظری) هم وجود داشته و یا دارند - که مورد نظر ما نیستند - از جمله :

- ساختار داده جعبه‌ای[۱]
- ساختار داده هایپرگرافی[۲]
- "ساختار داده" لیستهای وارون[۳]

نکته ۲ : ساختار داده دیگری که می‌توان در حیطه پایگاه داده‌ها مطرح کرد ، ساختار داده شیء[۴] است (از مدل داده شیء) هر چند در مورد وجود این مدل ، حداقل به آن معنا که در تئوری مدل رابطه‌ای مطرح است ، بین مؤلفین اتفاق نظر نیست . ضمناً ساختار داده شیء - رابطه‌ای[۵] هم مطرح است (برای اطلاع بیشتر در این‌باره می‌توان از جمله به [روحا ۸۸] مراجعه کرد) .

۲-۱ : آشنایی با ساختار دادهٔ جدولی[۶] (دیدگاه کاربردی)

در این قسمت با ساختار داده جدولی (در تئوری : رابطه‌ای) آشنا می‌شویم . منظور از دیدگاه کاربردی یعنی اصلاً به مفاهیم نظری مدل داده رابطه‌ای نمی‌پردازیم (این مفاهیم را درگفتار دهم خواهیم دید) . فعلاً ببینیم این ساختار از دیدگاه کاربردی چگونه است؟

۲-۱-۱ : عنصر ساختاری

رابطه مفهومی است ریاضی که در بحث مفاهیم مبنایی مدل رابطه‌ای خواهیم دید . اما از دید کاربر، **رابطه نمایش جدولی دارد** ، به بیان دیگر **جدول صرفاً امکانی است ساده شده برای نمایش مفهوم رابطه** . پایگاه داده‌های جدولی مجموعه‌ای است از تعدادی **نوع جدول** . هر نوع جدول یک نام و حداقل یک ستون دارد با نام و نوع داده مشخص . پایگاه داده‌های جدولی از نظر محتوای داده‌ای مجموعه‌ای است از نمونه‌های متمایز **نوع سطرها** (حداقل یک نوع سطر). نمونه‌های متمایز هر نوع سطر در یک جدول جای دارند ، نوع سطر را نوع جدول مربوطه مشخص می‌کند) .

با توجه به آنچه گفته شد ، در این بحث فقط به ساختار داده جدولی می‌پردازیم . مفاهیم ساختار جدولی عبارتند از :

- **نوع جدول**
- **سطر**
- **ستون**

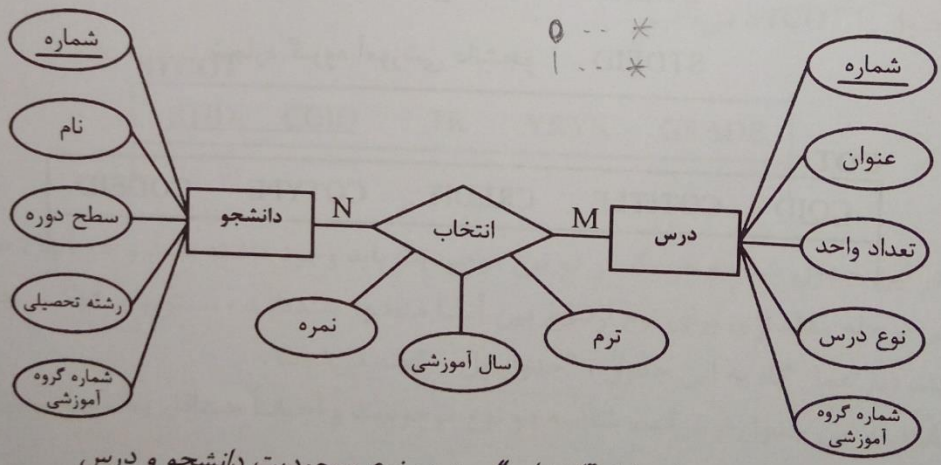
گفتیم که هر ساختار داده ، حداقل یک عنصر ساختاری اساسی دارد . در این ساختار **تنها عنصر ساختاری اساسی همین مفهوم نوع جدول** است . معنای این مطلب این است که آنچه را در مدلسازی با نوع موجودیتها ، صفات و نوع ارتباط‌ها نمایش دادیم ، در مرحله طراحی منطقی با ساختار جدولی فقط با مفهوم نوع جدول نمایش می‌دهیم . به‌این نکته مهم باز هم خواهیم پرداخت .

۲-۱-۲ : طراحی پایگاه داده‌های جدولی[1] (مثالهای مقدماتی)

■ ***مثال ۱ :*** ارتباط با چندی M : N

در این مثال قسمتی از نمودار ER مثال ۲۴ از گفتار سوم ، البته بسیار ساده شده ، را در نظر می‌گیریم (شکل ۴-۲) .

توجه داریم که به خاطر ساده شدن مثال فقط دو نوع موجودیت دانشجو و درس و نوع، ارتباط "انتخاب" بین این دو نوع موجودیت را در نظر گرفته‌ایم . در حالیکه در نمودار ER مثال در پیوستار ۱ نوع موجودیتهای دیگر از جمله نوع موجودیت ضعیف "گروه درسی" ، نوع موجودیت استاد و نوع موجودیت گروه آموزشی و ... را هم منظور کرده‌ایم . بنابراین در این گفتار ، نمودار بسیار ساده شکل ۴-۲ مبنای بحث است . در این مثال همه صفات نوع موجودیت‌های دانشجو و درس را در نظر نگرفته‌ایم (بعلاوه در یک "سیستم واقعی"، صفات هر یک از این نوع موجودیت‌ها خیلی بیشتر است و گاه بیش از یک نوع جدول برای نمایش هر یک از آنها ، لازم است) .



شکل ۴-۲ : نوع ارتباط "انتخاب" بین دو نوع موجودیت دانشجو و درس

می‌خواهیم بر اساس این مدلسازی ، پایگاه داده‌های جدولی را طراحی کنیم . برای این منظور :

- هر نوع موجودیت مستقل شرکت‌کننده در ارتباط با چندی M : N را با یک نوع جدول نمایش می‌دهیم .

بنابراین در این مثال یک نوع جدول برای نوع موجودیت دانشجو و یک نوع جدول برای نوع موجودیت درس داریم . صفات هر نوع موجودیت با ستونهای جدول نشان داده می‌شوند (فعلاً همه صفات را ساده و تک‌مقداری فرض می‌کنیم) :

STT

STID	STNAME	STLEV	STMJR	STDEID

معنای صفات به شرح زیر است :

نام صفت	معنا
STID	شماره دانشجویی
STNAME	نام دانشجو
STLEV	سطح دوره تحصیلی
STMJR	رشته تحصیلی
STDEID	شماره گروه آموزشی دانشجو

COT

COID	COTITLE	CREDIT	COTYPE	CODEID

در اینجا نیز همه صفات نوع موجودیت درس را در نظر نگرفته‌ایم . معنای صفات به شرح زیر است :

نام صفت	معنا
COID	شماره درس
COTITLE	عنوان درس
CREDIT	تعداد واحد
COTYPE	نوع درس
CODEID	شماره گروه آموزشی ارائه کننده درس

هر سطر از جدول STT یک نمونه متمایز از نوع موجودیت دانشجو و هر سطر از جدول COT ، یک نمونه متمایز از نوع موجودیت درس را نشان می‌دهد . هر سطر از هر جدول ، یک مقدار مشخص شناسه (کلید) دارد که در هیچ سطر دیگری وجود ندارد . در اینجا ستون STID ، شناسه سطرهای جدول STT و ستون COID شناسه سطرهای جدول COT است . حال باید نوع ارتباط بین این دو نوع موجودیت را نمایش دهیم ، برای این منظور :

- هر نوع ارتباط M : N بین دو نوع موجودیت مستقل را با یک نوع جدول نمایش می‌دهیم . بنابراین در این مثال جدول سومی لازم است تا نوع ارتباط "انتخاب" را نشان دهد . این جدول را STCOT می‌نامیم .

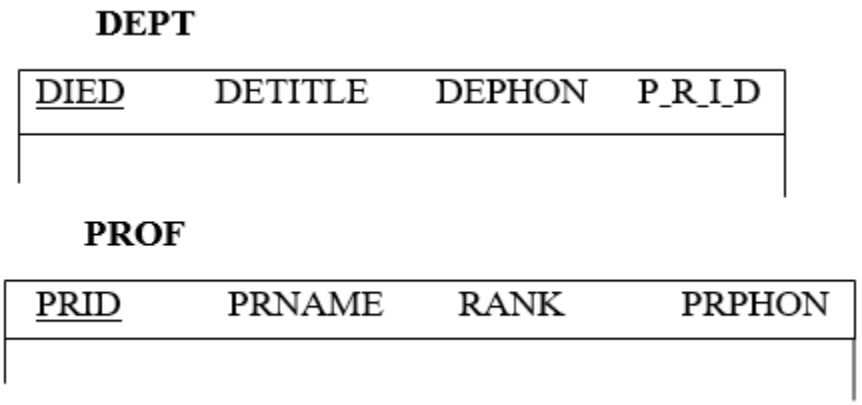
STCOT

STID	COID	TR	YRYR	GRADE

در این جدول شناسه هر یک از دو نوع موجودیت باید وجود داشته باشد و جز آنها ، صفاتی که در مرحله مدلسازی برای نوع ارتباط بین آنها منظور شده‌اند ، ستونهای دیگر جدول هستند (در عمل گاه به این جدول ، "جدول پل[1]" گفته می‌شود) .

شناسه این جدول از ترکیب شناسه دو نوع موجودیت و احیاناً حداقل یک صفت از نوع ارتباط بین آنها تشکیل می‌شود .

مشارکت غیرالزامی را در جدول نمایشگر نوع موجودیت با مشارکت الزامی تکرار می کنیم تا شماره مدیر گروه را در این جدول داشته باشیم. (برای جزئیات به گفتار چهاردهم مراجعه شود).



توجه داریم که طراحی جدولها در مثالهای ۲ و ۳ مستقل از یکدیگر انجام شده است.
توجه : این حالات و برخی حالات دیگر ؛ در طراحی پایگاه داده های رابطه ای ، بسیار مهم هستند . در واقع طرح ، در تبدیل نمودار ER به "مجموعه ای از رابطه ها"، باید طرز تبدیل حالات گوناگون در نمودار ER به طراحی منطقی پایگاه داده های رابطه ای را به درستی بداند.
ما در جای خود (گفتار چهاردهم) به این موضوع بسیارمهم به تفصیل خواهیم پرداخت.

۲-۱-۳ : شِما ی پایگاه داده های جدولی (ساده شده)
در اینجا برای اینکه با ساختار داده جدولی و نیز پایگاه جدولی بیشتر آشنا شویم ، جدول های طراحی شده در مثال یک را تعریف می کنیم.
شمای پایگاه داده ها (در بیان مقدماتی) عبارت است از تعریف (توصیف) ساختهای منطقی طراحی شده ونوعی "برنامه" است شامل دستورات " تعریف داده ها " و "کنترل داده ها " ، و دستورات "پردازش داده ها " (به تسامح : پردازش داده ها که گاه دستکاری داده ها گفته می شود) در آن وجود ندارد.

برای نوشتن شما هریک از جدول ها را باید تعریف و به سیستم مدیریت معرفی کنیم (برای اطلاع بیشتر در مورد شِما به گفتار پنجم مراجعه شود) . به عبارت دیگر نام جدول ، مشخصات هر ستون و ... را باید بدهیم. بدیهی است به یک دستور(حکم) تعریف (ایجاد) جدول نیاز داریم.ما از امکانات زبانی استفاده می کنیم موسوم به زبان SQL که زبان "استانده " سیستم های رابطه ای است. فعلا بعضی دستورات این زبان را به کار میبریم و با امکانات آن در حد مقدمات آشنا می شویم (درقسمت ۲-۱-۵ از همین گفتار) . چون دستورات این زبان ، حداقل در سطح برنامه سازی مقدماتی و

غیرتخصصی ، از سادگی خاصی برخوردار است (بنابراین می توان در این بحث آنها را به کار برد).در گفتار دوازدهم جنبه های این زبان را به تفصیل بررسی خواهیم کرد.
نکته ۳ : یکی از اهداف تکنولوژی پایگاه داده ها جداسازی برنامه ها تعریف و کنترل پایگاه جداسازی برنامه های تعریف و کنترل پایگاه داده ها (شِما) از برنامه های برزش داده های ذخیره شده در پایگاه داده ها است . در مبحث معماری پایگاه داده ها (گفتار پنجم) ، در قسمت زبان ، خواهیم دید که دستورهای تعریف و کنترل داده ها و دستورهای برزش داده ها ، هر یک زیر مجموعه ای از یک زبان را تشکیل می دهند که بطور جداگانه و نه در یک برنامه ، استفاده می شوند.

شِمای (ساده شده) پایگاه داده های جدولی مثال ۱ چنین است:

```
CREATE   TABLE   STT
(STID   CHAR(۸)   NOT NULL ,
STNAME      CHAR(۲۵) ,
STLEV        CHAR(۱۲) ,
STMJR        CHAR(۲۰) ,
STDEID       CHAR(٤) ,
PRIMARY   KEY   STID ;

CREATE    TABLE  COT
(COID         CHAR(٦)   NOT   NULL ,
COTITLE     CHAR, (۱٦)
```

4-1-2 : عملیات در پایگاه داده های جدولی
حال فرض می کنیم کاربر می خواهد در پایگاه جدولی خود عملیاتی انجام دهد :همان چهار عمل اصلی را که می شناسیم :

- بازیابی
- درج
- حذف
- به هنگام سازی

```
INSERT
INTO STT
VALUES ('7412077' , 'ROHANI' , 'bs' , 'Comp' , 'D222') ;
```

و نیز می‌توان سطر ناقص هم درج کرد . در درج سطر ناقص ، باید مقدار ستون کلید را به سیستم داد .

کنجکاوی ۳ : اگر بخواهیم تعدادی سطر در یک جدول درج کنیم ، چه باید کرد؟

ب : حذف

- **حذف تک سطر**

■ ***مثال ۱۲ :*** حذف یک نمونه درس

```
DELETE
FROM COT
WHERE COID = 'COM777' ;
```

با کلاز[1] WHERE شرط یا شرایط مورد نظر را به سیستم می‌دهیم .

کنجکاوی ۴ : اگر در جدول STCOT سطرهایی با همین شماره درس وجود داشته باشد ، با حذف درس COM777 از جدول COT چه پیش می‌آید؟

- **حذف چند سطر (حذف مجموعه‌ای)**

■ ***مثال ۱۳ :*** حذف تعدادی دانشجو

```
DELETE
FROM STT
WHERE STLEV = 'ms'
```

با این دستور مشخصات دانشجویان دروه "کارشناسی ارشد" از جدول STT حذف می‌شود .

کنجکاوی ۵ : اگر در جدول STCOT سطرهایی با شماره دانشجویی دانشجویان دوره کارشناسی ارشد وجود داشته باشند ، چه پیش می‌آید؟

پ : بهنگام‌سازی

- **بهنگام‌سازی تک سطر :**

■ ***مثال ۱۴ :*** تغییر شماره یک درس

```
UPDATE COT
SET COID = 'Com303'
WHERE COID = 'Com202' ;
```

با این دستور شماره درس Com202 ، عوض می‌شود .

کنجکاوی ۶ : اگر در جدول STCOT سطرهایی با شماره درس Com202 وجود داشته باشند چه پیش می‌آید؟ چه باید کرد؟

- **بهنگام‌سازی چند سطر :**

■ ***مثال ۱۵ :*** تغییر واحد درسهای آزمایشگاهی

```
UPDATE COT
SET CREDIT = '1'
WHERE COTYPE = 'Lab.'
```

با این دستور تعداد واحد تمام درسهای آزمایشگاهی (عملی) ، ۱ (یک) می‌شود .

۲-۱-۵ : آشنایی با SQL (مقدماتی)

در این قسمت ، زبان "SQL استانده" را به طور مقدماتی مطالعه می‌کنیم بی آنکه وارد جزئیات آن بشویم (در گفتار **دوازدهم** امکانات این زبان را با تفصیل خواهیم دید) . نگارش[۱] (گرداک)های متعددی از این زبان وجود دارد ، در واقع هر **سیستم جدولی** نگارش خاص خود را دارد .

اولین نگارش این زبان به عنوان قسمتی از سیستم R[۲]، در اواخر دهه ۱۹۸۰ میلادی در مرکز پژوهش سن خوزه وابسته به شرکت آی.بی.ام طراحی شد . این نگارش صورت تغییر یافته زبان SEQUEL2 است که توسط چمبرلین[۳] در سال ۱۹۷۶ میلادی طراحی شد . زبان اخیر خود بر اساس نگارشی به نام SQUAR که بایس[۴] در سال ۱۹۷۶ میلادی طراحی کرده بود ، ایجاد شد .

در سال ۱۹۸۶ مؤسسه استانده‌های ملی آمریکا (ANSI) و سازمان استانده‌های بین‌المللی (ISO) نگارش استانده این زبان را ارائه کردند . درسال ۱۹۸۹ نگارش استانده دیگری به نام ANSI/ISO SQL-89 ، موسوم به SQL1 ، عرضه شد . نگارش استانده دیگر این زبان SQL-92 یا SQL2 است که در سال ۱۹۹۲ عرضه شد . در حال حاضر آخرین نگارش به SQL-2003 موسوم است و تمام جنبه‌های SQL2 و نیز بعض از مفاهیم شیء - رابطه‌ای را در بردارد . شرکت آی.بی.ام در سال ۱۹۸۷ نگارش خاص خود را عرضه کرد .

در نگارش SQL 1 ، در اساس امکانات زیر وجود دارد :

- دستورات تعریف و کنترل داده ها
- دستورات برزش داده ها
- دستورات مجاز شماری (به طور مختصر)

۱-۵-۱-۲ : دستورات تعریف داده ها

این دستورات تحت نام کلی دستورات تعریف **شما** ۳ مطرح اند.

به جای INTEGER ، DECIMAL و CHARACTER می‌توان به ترتیب INT ، DEC و CHAR هم نوشت .

مقدار پیش‌نهاده (پیشفرض) length ، عدد یک ، مقدار پیش‌نهاده scale ، عدد صفر و مقدار پیش‌نهاده precision ، همان است که در ماشین مورد استفاده کاربر پیاده‌سازی شده است و چنین است در مورد SMALLINT و INTEGER .

همانطور که دیده می‌شود در این نگارش نوع داده‌ها محدودند به آنهایی که در کاربردهای معمولی داده‌داری - داده‌پردازی مورد نیازند .

- **دستور ایجاد جدول**

این دستور ، گذشته از جزئیات ، چنین است :

```
CREATE TABLE    name
 (Column        definition)
    "              "
    '              '
    '              '
    '              '

                          )
UNIQUE            (column (s)) ;
-  -  -  -  -  -
-  -  -  -  -  -
-  -  -  -  -  -
```

با اجرای این دستور، جدولی ایجاد می‌شود موسوم به **جدول مبنا** (پایه) که **پایا** است تا زمانی که به درخواست کاربر مجاز حذف شود (برای جزئیات ، به گفتار دوازدهم مراجعه شود).

■ ***مثال ۱۶ :*** تعریف نوع جدول دانشجو

```
CREATE TABLE STT
(STID CHAR(8) NOT NULL ,
STNAME CHAR(25) ,
STLEV CHAR(12) ,
STMJR CHAR(20) ,
STDEID CHAR(4))
UNIQUE (STID) ;
```

با اجرای دستور ایجاد جدول ، مشخصات جدول درکاتالوگ سیستم مدیریت وارد می‌شود . فایل متناظر با جدول طبعاً در بدو ایجاد خالی است . کاربر می‌تواند داده‌های مورد نظرش را در جدول ایجاد شده وارد کند که خواهیم دید .

.دستورSELECT

شکل کلی این دستور چنین است :

```
SELECT  [all distinct]item(s)-list
From  table(s)-name
[Where condition(s)]
[order by clumn(s)]
Group by column(s)]
[having condition(s)]
```

در بعضی نگارش ها امکان ORDER BY column(s) نیز وجود دارد.

تنها دستور بازیابی در این زبان همین دستور select است که امکانات گسترده ای دارد.

پایگاه داده های آموزشی ،از جمله جدولهای STCOT، COT، STT بخشی از امکانات این دستور را بررسی کنیم.

ما در ادامه بحث

۱-۲-۵-۱-۲ : بازیابی یک جدول

مثال۱۸ : در این مثال عملکرد کلی دستور

SELECT را نشان می دهیم:

Q: شماره و نام دانشجویان دوره کارشناسی را بدهید.

```
SELECT STID  ,STNAME
FROM STT
WHERE STLEV= bs;
```

جواب این پرسش ،جدولی است دو ستونی حاوی شماره و نام دانشجویان دوره" کارشناسی ارشد"

از همین مثال ساده بر می آید که حاصل اجرای دستور

SELECT در SQL

در حالت کلی ،زیر مجموعه افقی –عمودی از یک جدول است .ضمنا می توان نام هر ستون را همراه با نام جدول مربوطه نوشت و

مثلاSTIDبه جای فقط STT.STID

در این صورت نام جدول را اصطلاحا شناسگر ستون می گوییم .

این کار برای وضوح بیشتر برنامه است و البته در مواردی الزامی است که خواهیم دید.

Item(s)-list بعد از کلاس select

می توان نام ستون های مورد نظر و یا یک عبارت اسکالر نوشت مثال مورد اخیر را در ادامه بحث خواهیم دید.

به جدول مبنای BOOK نام دیگر B را داده‌ایم و نام ستون BKPRICE در جدول جواب PR است . می‌توان به جای AS ، فقط یک فاصله (بلانک) بین دو نام جدول گذاشت .

- **توابع جمعی[1] (گروهی)**

در این زبان تعدادی تابع وجود دارد (به گفتارهای یازدهم و دوازدهم مراجعه شود) . از جمله این توابع ، توابع جمعی هستند به شرح زیر :

- COUNT : تعداد مقادیر اسکالر در یک ستون را به دست می‌دهد ،
- SUM : حاصل جمع مقادیر اسکالر در یک ستون را به دست می‌دهد ،
- AVG : میانگین مقادیر اسکالر در یک ستون را به دست می‌دهد ،
- MAX : مقدار بیشینه (مقدار ماکزیمم) اسکالر در یک ستون را به دست می‌دهد ،
- MIN : مقدار کمینه (مقدار می‌نیمم) اسکالر در یک ستون را به دست می‌دهد .

شکل کلی این توابع چنین است [DATE 97] :

```
Function-name := COUNT(*)
                | { AVG | MAX | MIN | SUM | COUNT}
                ([ALL | DISTINCT] Column-name | Scalar-expression)
```

این توابع در متن ساختار SELECT و نه به تنهایی ، به کار می‌روند .

در توابع SUM و AVG ، نوع مقادیر ستون باید عددی باشد . در صورت قید DISTINCT ، مقادیر تکراری ستون منظور نمی‌شوند . در عوض با ALL ، همه مقادیر ستون منظور می‌شوند و البته ALL نیازی به تصریح ندارد و گزینه پیش‌نهاده است .

در تابع COUNT ، گزینه DISTINCT باید قید شود . تابع COUNT(*) صورت خاصی از تابع COUNT است و نوشتن گزینه DISTINCT در آن مجاز نیست . این تابع تمام سطرهای جدول را می‌شمارد ، حتی اگر سطر تکراری وجود داشته باشد . پس این تابع ، تابع سطرشمار است .

در توابع MAX و MIN قید گزینه DISTINCT بی‌معناست و سیستم آنرا نادیده می‌گیرد . حال عملکرد این توابع را با مثالهایی بررسی می‌کنیم .

■ مثال ۲۵ :

Q : کل تعداد درس‌ها را بدهید .

```
SELECT COUNT(*) AS CN
    FROM COT ;
```

1- Aggregate functions

کنجکاوی ۱۰ : آیا این پرسش به طرز دیگری هم نوشته می‌شود؟

■ ***مثال ۲۶ :***

Q : تعداد درسهایی که دانشجو با شماره '78110555' در ترم اول سال ۸۵ آنها قبول شده است ، را بدهید .

```
SELECT COUNT(DISTINCT COID)
    FROM STCOT
    WHERE STID = '78110555' AND GRADE ≥ 10 AND TR = '1'
    AND YRYR = '85' ;
```

■ ***مثال ۲۷ :***

Q : تعداد دانشجویان ثبت نام کرده در ترم اول ۷۹-۷۸ را بدهید .

```
SELECT COUNT (DISTINCT STID)
  FROM STCOT
      WHERE TR = '1' AND YRYR = '78-79' ;
```

کنجکاوی ۱۱ : آیا در پاسخ به این پرسش ، می‌توان از جدول دیگری استفاده کرد؟

■ ***مثال ۲۸ :***

Q : مقدار بیشینه و مقدار کمینه نمره در درس COM222 در ترم دوم ۸۰-۷۹ چیست؟

```
SELECT MIN (GRADE) , MAX (GRADE)
  FROM STCOT
    WHERE TR = '2' AND YRYR = '79-80' AND COID = 'COM 222' ;
```

در برخی از نگارش‌های SQL توابع دیگری هم وجود دارد ، از جمله توابع زیر :

- DATE و TIME
- NEXT-DAY
- MONTHS-BETWEEN
- SYSDATE
- SIN
- SINH
- TAN
- TANH
- LOG
- MOD
- RPAD
- REPLACE
- LENGTH
- USER
- NEW-TIME
- CEIL
- FLOOR
- COS
- COSH
- EXP
- LN
- POWER
- LPAD
- TRANSLATE
- INSTR

مطالعه این توابع را به خواننده وا می‌گذاریم .

- **LIKE و NOTLIKE**

با این دو امکان می‌توان داده‌های مورد نظر را با دادن یک رشته کاراکتری به عنوان نشانوند جستجو و بیان شرایط مورد نظر ، بازیابی کرد . برای درک مطلب مثالهایی قید می‌شود .

جدول زیر را در نظر بگیریم :

PROF | **PRID** , PRNAME , PRRANK , ... , PRDEID |

■ **مثال ۲۹ :**

Q : مشخصات استادانی را بدهید که نام آنها مختوم به کاراکتر Y باشد .

```
SELECT   PROF.*
  FROM   PROF
       WHERE PRNAME LIKE '%Y' ;
```

■ **مثال ۳۰ :**

Q : مشخصات استادانی را بدهید که نام آنها با AR شروع شده باشد .

```
SELECT   PROF.*
  FROM   PROF
       WHERE PRNAME LIKE 'AR%' ;
```

■ **مثال ۳۱ :**

Q : مشخصات استادانی را بدهید که در نام آنها رشته کاراکتری ZAD وجود داشته باشد .

```
SELECT   PROF.*
  FROM   PROF
    WHERE  PRNAME LIKE '%ZAD%' ;
```

■ **مثال ۳۲ :**

Q : مشخصات استادانی را بدهید که نام آنها هشت کاراکتری بوده و کاراکتر سوم و چهارم در نام آنها BA باشد .

```
SELECT   PROF.*
    FROM PROF
       WHERE PRNAME LIKE '--BA ----' ;
```

می‌توان صورت نایی همین پرسشها را هم مطرح کرد و آنها را با NOT LIKE نوشت.

- **آزمون وجود هیچمقدار در یک ستون**

می‌توان با امکان IS NULL وجود هیچمقدار در یک ستون را تست کرد .

■ ***مثال ۳۳ :***

Q : شماره و نام دانشجویانی را بدهید که نمره آنها در SOC333 در ترم اول ۸۷-۸۶ هنوز اعلام نشده است .

```
SELECT STID , STNAME
   FROM  STCOT
      WHERE COID = 'SOC333' AND TR = '1' AND YRYR  = '86-87'
      AND GRADE IS NULL ;
```

صورت نایی این امکان هم وجود دارد . اگر بنویسیم :

```
Column-name IS NOT NULL
```

معادل است با :

```
NOT (Column-name IS NULL)
```

- **گروه‌بندی : GROUP BY**

با این امکان می‌توان سطرهای جدول را بر حسب مقادیر یک یا بیش از یک ستون گروه‌بندی کرد به نحوی که درهر گروه ، مقدار آن ستون(ها) یکسان باشد .

■ ***مثال ۳۴ :*** شماره هر درس انتخاب شده در ترم دوم سال ۷۹-۷۸ و میانگین نمرات دانشجویان در آن را بدهید.

```
SELECT  COID , AVG(GRADE) AS AVGGR
      FROM  STCOT
            WHERE TR = '2'
            AND YRYR = '78-79'
            GROUP BY  COID ;
```

این دستور جدول STCOT را بر حسب مقادیر ستون COID **منطقاً** گروه‌بندی می‌کند ، به نحوی که در هر گروه مقدار ستون COID یکسان است . سپس میانگین نمرات هر درس محاسبه می‌شود . به ستون ذکر شده در کلاز GROUP BY ، ستون گروه‌بندی[1] گفته می‌شود. جدول نتیجه گروه‌بندی، جدول گروه‌بندی شده[2] نام دارد.

کنجکاوی ۱۲ : در این جمله ، کلمه منطقاً یعنی چه؟

1- Grouping column
2- Grouped table