

به نام آنکه جان را فکرت آموخت

چراغ دل به نور جان برافروخت

پایگاه داده ها (بانک اطلاعاتی) - UML

نوع درس : تئوری

دوره : کارشناسی

رشته : مهندسی کامپیوتر

تعداد واحد : ۳

مدرس : علیرضا عظیمی

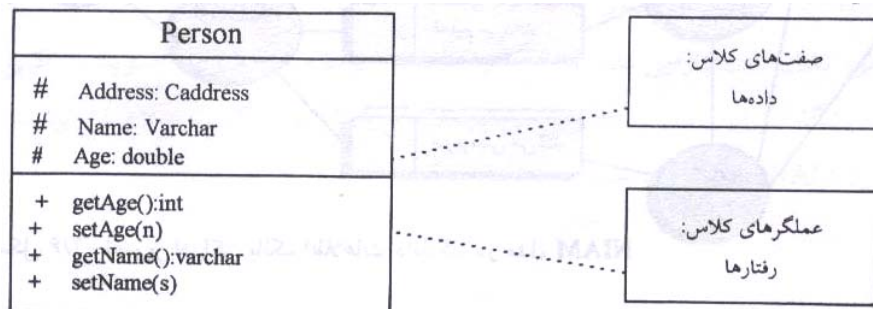
2/3- مدل UML

UML (Unified Modeling Language) یک استاندارد مدلسازی است که ترکیبی از روشهای مدلسازی قبلی است. این زبان برای مشخص سازی (*specifying*)، بصری سازی (*visualizing*)، ساختار بندی (*constructing*) و مستند سازی (*documenting*) سیستم های نرم افزاری به کار می رود و قابل استفاده در تمامی فرآیندهای چرخه حیات سیستم است. UML ترکیبی از مفاهیم مدلسازی فرآیند کاری (جریان کاری)، مدلسازی اشیاء و مولفه های سیستم است. اولین نسخه UML در اکتبر 1995 ارائه گردید. در این بخش به کاربرد UML در بانک اطلاعات می پردازیم. مقدمات UML دانسته فرض می شود.

2/3-1 مدل کردن کلاس ها

مدل کلاسها در UML ابزار اصلی برای بیان ساختار منطقی یک سیستم نرم افزاری است که نیازمندیهای داده ها و رفتار اشیاء را بیان می کند. یک کلاس موجودیت منطقی اصلی در UML است که داده ها و رفتارها را بیان می کند و یک کلاس یک الگو یا مدل از اشیاء و نمونه هایی است که در زمان اجرا ایجاد می شوند. وقتی که یک مدل منطقی مانند سلسله مراتب ساختاری در UML می سازیم، به طور صریح با کلاسها سرو کار داریم. وقتی با دیاگرام های پویا مانند دیاگرام توالی و همکاری کار می کنیم، به اشیاء یا نمونه هایی از کلاسها و فعل و انفعالات بین آنها در زمان اجرا سرو کار داریم.

اصول پنهان سازی داده و بسته بندی در کلاسها رعایت می شود. یک کلاس شامل یک سری عناصر داده ها است که خودش مسئول آنهاست. دسترسی به این عناصر درونی باید از طریق واسط کلاس یا رفتارهای تعریف شده توسط کلاس انجام گردد. شکل زیر کلاس Person را نمایش می دهد:



رفتار در مدل کلاسها با استفاده از عملیاتی که برای کلاس تعریف شده اند، مشخص می گردد. عملیات ممکن است از خارج کلاس قابل رویت باشند، برای فرزندان قابل رویت باشند (حفاظت شده) و یا به طور کلی مخفی باشند. با ترکیب کردن داده های پنهان با یک واسط دسترسی عمومی و دستکاری داده به صورت حفاظت شده یا پنهان، طراح کلاس می تواند واحدهای ساختاری را که قابلیت نگهداری بالایی دارند، ایجاد نماید.

کلاسها و اشیا به روشهای زیر در UML ارتباط برقرار می کنند:

ارتباط (association) یک ارتباط بین دو کلاس است که نشان می دهد حداقل یک طرف ارتباط چیزهایی از طرف مقابل می داند و به نوعی آن را استفاده یا دستکاری

می کند. این رابطه می تواند تابعی (برای من کاری انجام بده) یا ساختاری (برای من چیزی باش) باشد.

اجتماع (aggregation) نوعی از ارتباط است که نشان دهنده تجمع یک کلاس از کلاس های دیگری است.

ترکیب (composition) نوعی قویتر از اجتماع است که نشان می دهد یک کلاس عملاً ترکیبی از دیگران است.

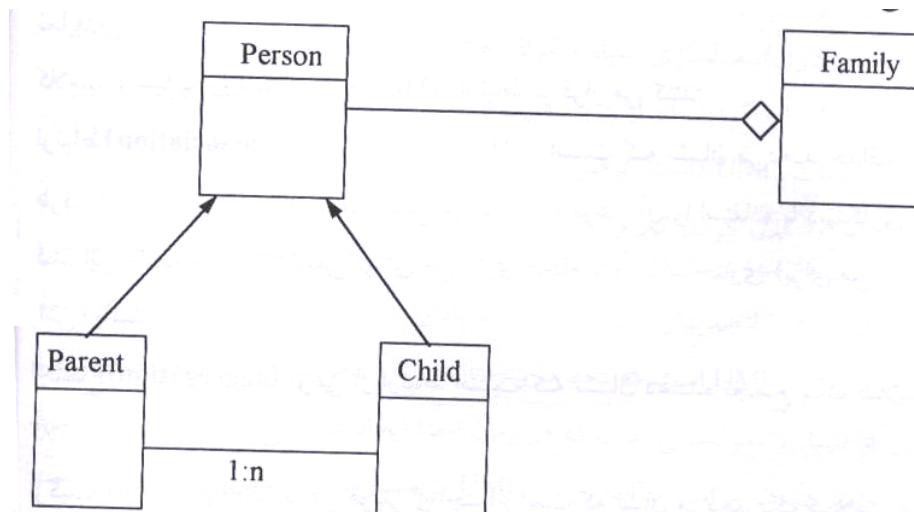
وراثت (inheritance) در مدل کلاس ها قابلیتی فراهم می نماید که از طریق آن می توان رفتارهای مشترک را در کلاس های عمومی قرارداد و آنها را به عنوان اجداد کلاس های دیگر معرفی نمود. وراثت ابزاری برای مدیریت استفاده مجدد و پیچیدگی است.

ناوبری (navigation) ارتباط یک شیء یا شیء دیگری در زمان اجرا است. یک شیء نوعی پیوست (یک اشاره گر یا شناسه شیء) دارد که توسط آن می توان آن شیء را احضار نمود یا موقعیت آن را تعیین کرد.

2-3-2- پشتیبانی از مدل رابطه ای

جداول و ستونها : جدول مجموعه ای است از یک یا چند ستون که هر کدام نامی یکتا دارند. هر ستون از یک نوع داده اولیه معین مانند عدد، متن یا داده دودویی تعریف می شود.

در شکل زیر چهار جدول دیده می شود که با ارث بری، ارتباط و اجتماع با هم پیوند خورده اند.



رفتار : جداول، علاوه بر صفتها، می توانند دارای رفتار نیز باشند.

رفتار تخصیص یافته به هر جدول معمولاً بر اساس قوانین منطقی و کار آن است.

انواع رفتار عبارتند از :

محدودیتها (constraint) می توانند روی ستونها به شکل الزامات یکتایی، قبول جامعیت رابطه ای با جداول و سطرهای دیگر، مقادیر مجاز و یا انواع داده ها اعمال شوند.

تریگرها (trigger) رفتارهای اضافی را، که می توانند به پدیده تخصیص داده شوند ، فراهم می آورند و معمولاً برای حفظ جامعیت داده ای قبل یا بعد از به روزرسانی، حذف یا اضافه کردن، به کار می روند.

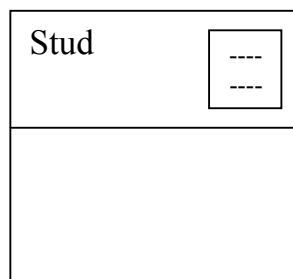
رویه های ذخیره شده بانک اطلاعات (*stored procedure*) ابزاری برای گسترش کارایی بانک اطلاعات از طریق زبانهای دیگر است.

ارتباط و شناسه :

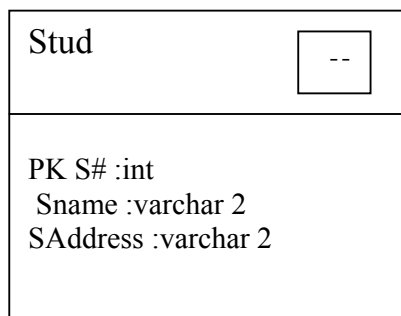
کلید اصلی یک جدول مقدار یکتایی را برای یک سطر خاص در اختیار قرار می دهد. دو نوع کلید اصلی وجود دارد: اولی کلید معنی دار است که از ستونهای داده های که معنایش را در دامنه کاری بیان می کنند، تشکیل شده و دومی شناسه انتزاعی یکتا مانند مقدار شمارنده که معنای کاری ندارد ولی بطور یکتا یک سطر را مشخص می کند.

یک جدول ممکن است ستونهایی داشته باشد که به کلید اصلی جدول دیگری نگاشت شده باشد. این ارتباط بین جداول، کلید خارجی را تعریف می کند و به یک ارتباط ساختاری یا تناظری بین جداول، کلید خارجی را تعریف می کند و به یک ارتباط ساختاری یا تناظر بین دو جدول اشاره می نماید.

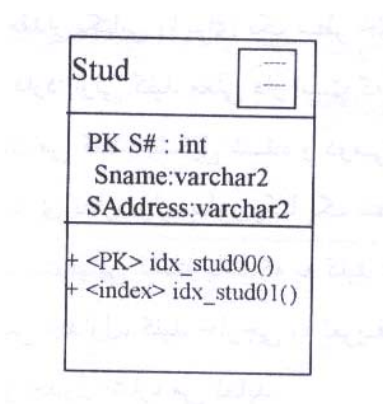
یک جدول در مدل UML ، کلاسی با قالب <table> است و همانطور که در شکل زیر دیده می شود، با آیکون جدول در گوشه راست بالای آن نمایش داده می شود.



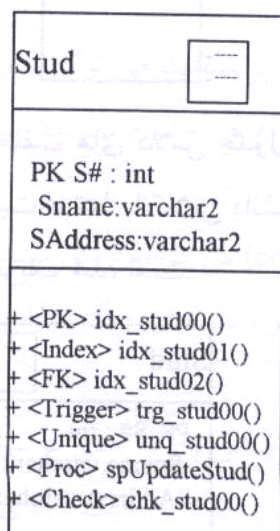
ستونهای جدول به عنوان صفت های کلاس جدول تعریف می شوند . به عنوان مثال شکل زیر بیانگر چند خصوصیت مرتبط با کلاس دانشجو است . در این مثال یک شناسه شی به عنوان کلید اصلی تعریف شده است .



رفتارها به عنوان عملیات قالب بندی شده (***stereotyped operations***) بیان می گردند. شکل زیر جدول بالا را با یک کلید اصلی و شاخص نشان می دهد که هر دو توسط عملیات قالب بندی شده تعریف گشته اند:



PK روی ستون S# کلید اصلی منطقی را تعریف می کند در حالی که <PK> idx_stud00 بیان کننده محدودیتها و رفتار مربوط به پیاده سازی کلید اصلی است. حال می توانیم رفتارهای اضافی مانند تریگرها، شروط و رویه های ذخیره شده را تعریف کنیم.



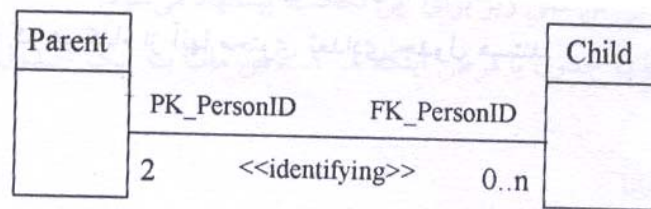
شکل بالا بیانگر رفتارهای زیر است :

1. قید کلید اصلی (**PK**)
2. قید کلید خارجی (**FK**)
3. قید شاخص (**Index**)
4. یک تریگر (**Trigger**)
5. قید یکتایی (**Unique**)
6. یک رویه ذخیره شده (**Proc**) که رسماً بخشی از پروفایل داده نیست.
7. اعتبار سنجی (**Check**)

با استفاده از روش فوق، مدلسازی داده ای پیچیده و تعریف رفتارها ممکن می گردد. علاوه بر این UML روشی برای نمایش ارتباط بین پدیده ها به صورت زیر در اختیار قرار می دهد.

مدل داده ها در UML ارتباط را به عنوان یک وابستگی بین دو جدول تعریف می کند و آن را به صورت یک رابطه تناظر دارای قالب نشان می دهد که شامل مجموعه ای از کلید های اصلی و خارجی است. پروفایل داده نیازمند این است که یک ارتباط همواره یک پدر و یک فرزند داشته باشد که پدر یک کلید اصلی را تعریف می کند و فرزند کلید خارجی را براساس تمام یا بخشی از کلید اصلی پدر معرفی می کند. اگر کلید خارجی فرزند شامل تمامی عناصر کلید اصلی پدر باشد، این ارتباط **identifying** و اگر فقط شامل برخی از عناصر کلید اصلی **non-identifying** نامیده می شود.

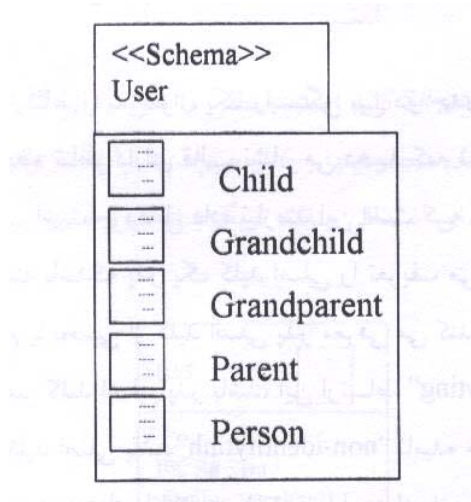
ارتباط می تواند شامل محدودیتهای اندازه ای باشد و با استفاده از جفتهای وابسته PK-FK به عنوان قوانین تناظر مدل گردد. شکل زیر مدلسازی این نوع ارتباط را در UML نشان می دهد.



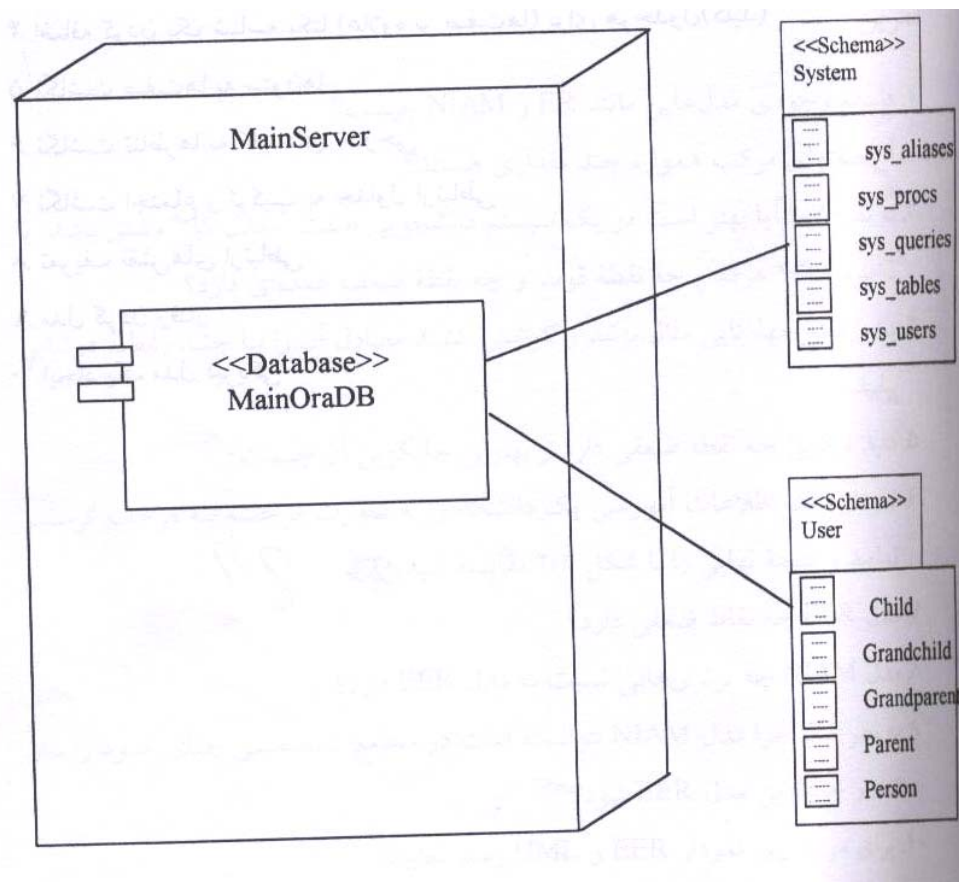
3-2/3 مدل فیزیکی

زبان UML مکانیزمهایی برای بیان ساختار فیزیکی کلی بانک اطلاعات، محتویات و مکان استقرار آن نیز دارد. برای بیان بانک اطلاعات فیزیکی در UML از مولفه های **(component)** قالب دار استفاده می شود. یک مولفه، یک پدیده گسسته و قابل استقرار در مدل را نمایش می دهد. در مدل فیزیکی یک مولفه می تواند به یک بخش

فیزیکی سخت افزار (یک نود در **UML**) نگاشت شود. برای نمایش شمای بانک اطلاعات از قالب **<schema>** در یک بسته (**package**) استفاده می شود. یک جدول می تواند برای تعیین قلمرو و مکان در یک بانک اطلاعات در **<schema>** قرار داده شود. شکل زیر یک نمونه را نشان می دهد:



یک نود بخشی فیزیکی از سخت افزار است (مانند یک سرور یونیکس) که مولفه ها بر روی آن مستقر می شوند. در مثال شکل زیر مولفه بانک اطلاعات به دو شمای منطقی نگاشت شده که هر کدام از آنها محتوی تعدادی جدول هستند.



2/3-4 نگاشت از UML به مدل شیئی رابطه ای

پس از طراحی بانک اطلاعات در UML می توان آنرا به مدل شیئی رابطه ای نگاشت کرد و پیاده سازی نمود.

مراحل کلی زیر برای این نگاشت پیشنهاد می شود:

1. نگاشت کلاس ها به جدول با فرض اینکه هر کلاس مانا به یک جدول رابطه ای نگاشت می شود.

2. شناسائی اشیاء مانا

3. انتخاب یک استراتژی وراثت
4. اضافه کردن یک شناسه یکتا (علاوه بر صفت ها) برای هر جدول (کلید)
5. نگاشت صفت ها به ستون ها
6. نگاشت تناظرها به کلیدهای خارجی
7. نگاشت اجتماع و ترکیب به جداول ارتباطی
8. تعریف نقش های ارتباطی
9. مدل کردن رفتار
10. ایجاد یک مدل فیزیکی