

تکنیک های مدل سازی

Modeling Techniques

قبل از پرداختن به مدل ها و مدل سازی ها بهتر است یک بار مفاهیم متدلوژی را مرور کنیم :

متدلوژی ها

* رهنمون های جامعی هستند در جهت انجام و تکمیل تمام فعالیت های

SDLC (Software Development Life Cycle)

* مجموعه ای از مدل ها ، ابزار و تکنیک ها هستند .

مدل ها Models

* نمایش یک "نمود" مهم از دنیای واقعی ، نه مشابه اجسام واقعی .

* تجرید استفاده شده برای مجزا کردن ، خارج از "نمود" .

* نمودارها و جدول ها .

* برنامه ریزی پروژه و کمک در بودجه بندی .

بعضی از مدل های مورد استفاده در توسعه سیستم ها عبارتند از :
DFD,ERD,FlowChart,Use Case Diagrams,Class Diagrams .
و بعضی از مدل های مورد استفاده در مدیریت فرایند توسعه عبارتند از :
PERT Charts,Gantt chart organ, Hierarchy chart .

ابزار Tools

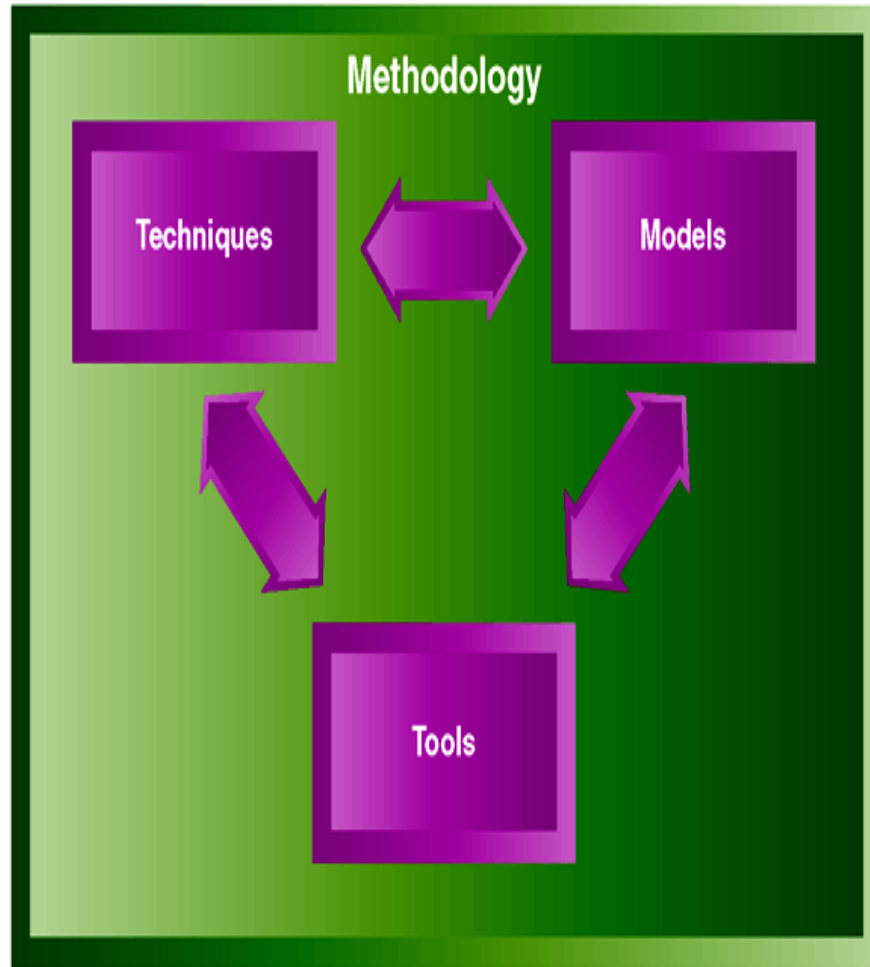
یک ابزار ، یک نرم افزار پشتیبان است که کمک می کند به ایجاد مدل ها
یا دیگر اجزای مورد نیاز در پروژه ، مانند :
smart editors , C-S help , debugging tools ,
CASE (Computer-Aided System Engineering) Tools .

تکنیک ها Technique

تکنیک مجموعه ای از رهنمون ها است که کمک می کند تا یک تحلیل گر فعالیت مربوط به توسعه سیستم را تکمیل کند از جمله :

Object Oriented analysis , Data modeling , Relational database design , Structured analysis and design .

ار تباط میان اجزای
متدلوژی



مدل سازی چیست ؟

یک مدل ، یک انتزاع یا تقریبی است که برای شبیه سازی واقعیات استفاده می شود . تمرکز مدل سازی روی این است که سیستم ، ”چه کاری“ انجام می دهد نه ”چگونه“ آن را انجام می دهد.

سه قسمت مدل سازی عبارتند از :

- مدل سازی فرایندی (DFD / Activity diagrams) .
- مدل سازی داده ای (ERD / Class diagrams) .
- مدل سازی رفتاری

(State Transition Diagrams ‘STD’ or Statechart Diagram , Decision Table , Decision Trees , Pseudo code , Structured English) .

دو خط مشی شناخته شده برای توسعه سیستم ها وجود دارد :
توسعه ساخت یافته

- نگرش فرایند گرا
- تکنیک تحلیل و طراحی ساخت یافته (SADT) که اولین تلاش های صورت گرفته برای رسیدن به نگرش بهبود کیفیت نرم افزار بود .
- برنامه نویسی بالا به پایین (که برنامه های پیچیده را به سلسله مازول ها تقسیم می کرد) .
- طراحی ساخت یافته
- جداول ساخت یافته
- آنالیز ساخت یافته
- نمودارهای جریان داده ها (DFD)
- نمودار ارتباط موجودیت ها (ERD)

مدل سازی

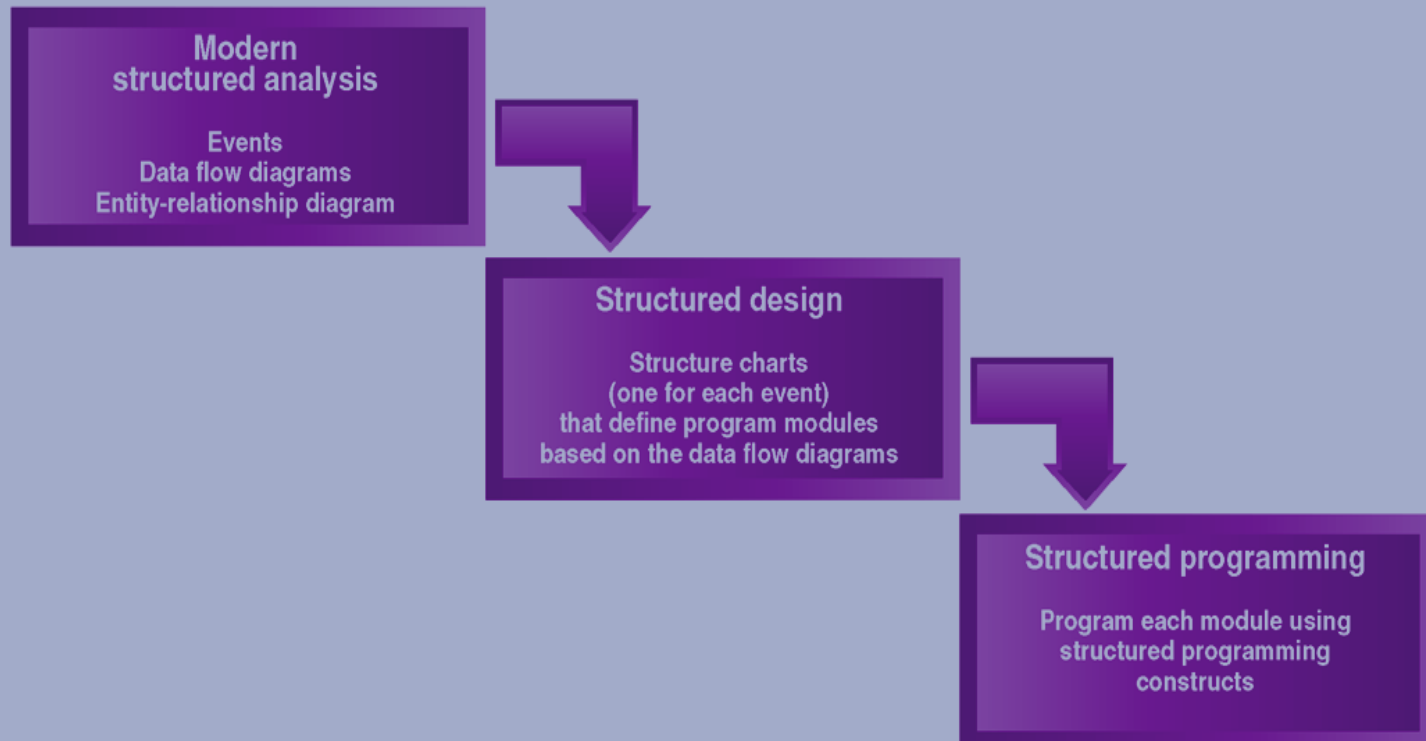
نگرش شی گرا

- نگرش شیء گرا چشم انداز یک سیستم اطلاعاتی است به عنوان مجموعه ای از تعاملات اشیا که برای به انجام رساندن وظایف با هم کار می کنند .
- شیء عنصری است در سیستم کامپیوتر که می تواند به پیام ها واکنش دهد .
- طراحی شی گرا یعنی تعریف تمام انواع اشیا لازم برای برقراری ارتباط با افراد و دیگر وسایل سیستم و تصحیح تعریف هر نوع شی به نحوی که با یک زبان یا محیط معین پیاده سازی شود .
- برنامه نویسی شی گرا یعنی نوشتن دستورات در یک زبان برنامه نویسی برای تعریف اینکه هر نوع شی چه کاری انجام دهد ، شامل پیام هایی که اشیا می فرستند یا دریافت می کنند .

اما به عنوان یک نگاه اجمالی در مورد دو نگرش گفته شده می توان گفت :

- آنالیز ساخت یافته منجر می شود به طراحی ساخت یافته و متعاقبا برنامه نویسی ساخت یافته . موارد زیر از اهداف تکنیک های ساخت یافته است :
- * رسیدن به کیفیت بالا در برنامه های با رفتار قابل پیش بینی .
 - * ایجاد برنامه هایی که به راحتی قابل اصلاح اند .
 - * ساده کردن برنامه ها و فرایند توسعه برنامه .
 - * رسیدن به قدرت پیش گویی و کنترل بالاتر در فرایند توسعه .
 - * سرعت بالاتر در فرایند توسعه .
 - * هزینه کمتر در فرایند توسعه .

How structured analysis leads to structured design and to structured programming.



شی گرا چیست ؟

- ساختن بنیادی اشیا ، که ترکیب می کند ساختمان داده ها و رفتار را در یک موجودیت واحد .
- راه سازمان دهی نرم افزار به عنوان مجموعه ای از اشیا مجزا که به هم می پیوندند در هر دو ساختمان داده ها و رفتار .
- پس در شی گرا ارتباط و اتصال میان داده و رفتار بسیار قوی است در حالیکه این اتصال در ساخت یافته بسیار آزادانه و ضعیف است .

هر کدام از دو نگرش گفته شده تکنیک های خاص خود را دارند .
مدل سازی فرایندی ، مدل سازی داده ای و مدل سازی رفتاری از جمله
تکنیک های مورد استفاده در هر دو نگرش هستند که نوع عملکرد این
تکنیک ها و پیاده سازی آنها در هر یک از این دو متفاوت است .
تکنیک های ساخت یافته عبارتند از:
مدل سازی فرایندی (DFD) ، مدل سازی داده ای (ERD) و مدل سازی
رفتاری (ELH) .
و تکنیک های شی گرا هم عبارتند از:

**Use case diagrams , Class diagrams ,
Interaction diagrams .**

ما در اینجا به تکنیک های مدل سازی ساخت یافته می پردازیم .

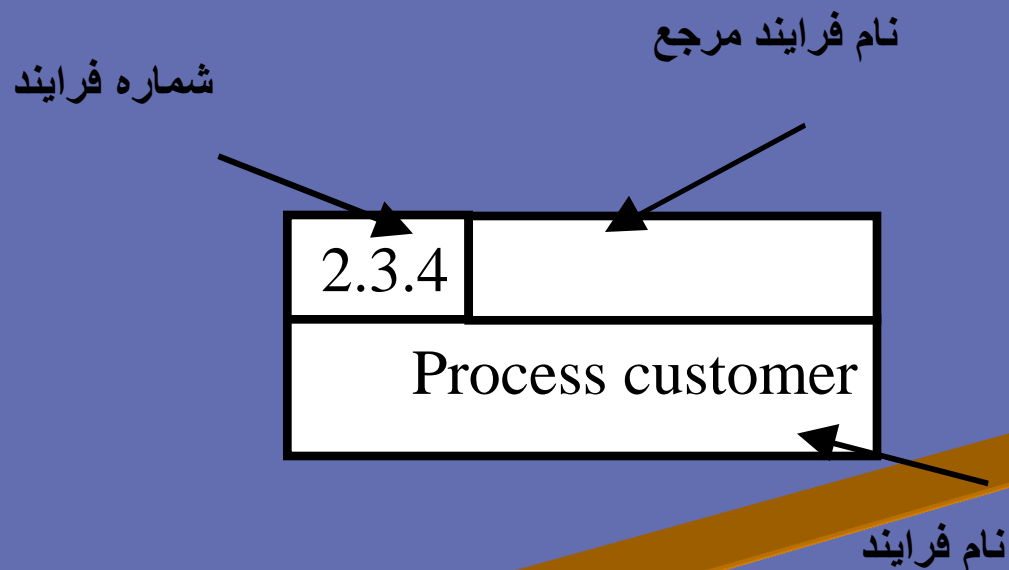
- یک DFD حرکت داده ها را در سرتاسر یک سیستم اطلاعاتی نشان می دهد ولی منطق برنامه یا مراحل پردازش را نشان نمی دهد .
- مجموعه ای از DFD ها یک مدل منطقی را بوجود می آورند که نشان می دهد سیستم چه کاری انجام می دهد (نه چگونه آن را انجام می دهد) .

اجزای DFD

DFD ها از چهار علامت پایه ای استفاده می کنند برای نمایش فرایندها ، جریان داده ها ، گروه های داده ای و موجودیت ها .

فرایند Process

داده ورودی را دریافت می کند و خروجی را تولید می کند که در شکل یا محتویات ویا هر دو فرق می کند .



جریان داده ها Data flow

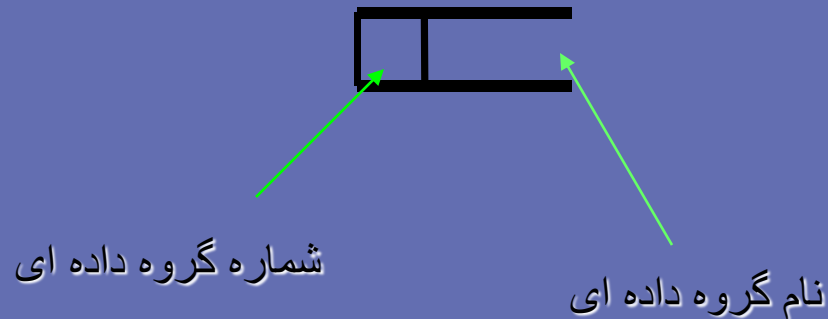
یک یا چند آیتم داده ای را نشان می دهد که به شکل زیر است :



پیکان یک جهت در مواردی است که داده ای را فقط وارد یا خارج کنیم
و پیکان دو طرفه برای مواردی است که هم داده را می خوانیم و هم مجددا
بازنویسی می کنیم .

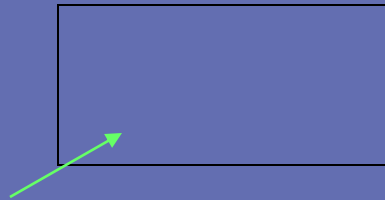
گروه های داده ای Data store

داده هایی را نشان می دهد که توسط سیستم ذخیره شده است . مشخصات فیزیکی یک گروه داده ای مهم نیست زیرا ما با یک مدل منطقی سرو کار داریم . یک گروه داده ای به شکل زیر نمایش داده می شود :



موجودیت Entity

با یک مستطیل نمایش داده می شود که نام موجودیت در داخل آن می آید و از آن به عنوان یک منبع و مرجع اصلی استفاده می شود .



نام موجودیت

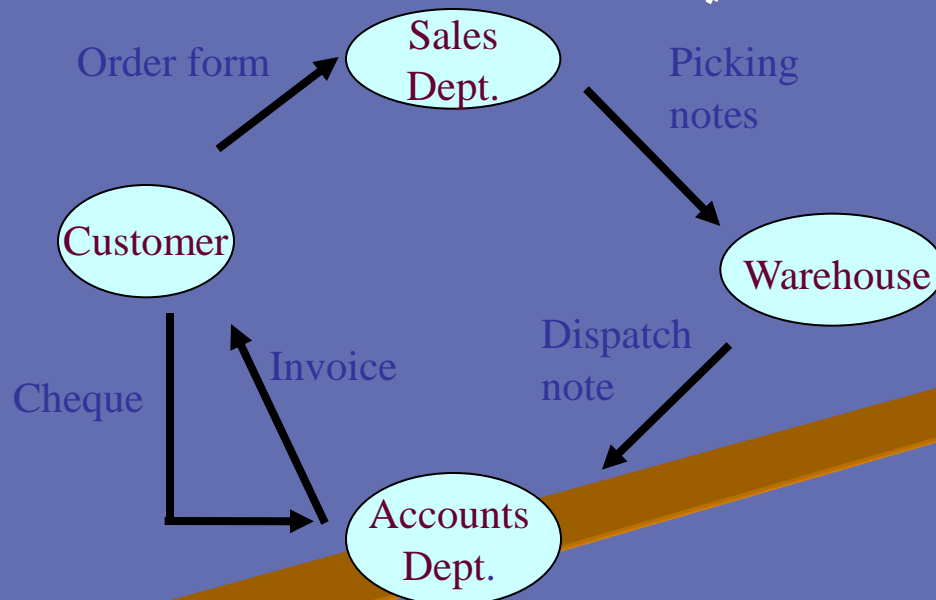
قواعد DFD ها

- 1 داده ها به طور مستقیم بین فرایندها جریان پیدا نمی کنند .
- 2 داده ها به طور مستقیم بین گروه های داده ای جریان پیدا نمی کنند .
- 3 داده ها نمی توانند به طور مستقیم از Data store به موجودیت ها یا بر عکس انتقال یابند .

در ادامه روشی برای تهیه یک DFD آمده است که توسط Geoff Cutts ارائه شده است . نکاتی هم علاوه بر این روش ذکر شده است .

: Method by Geoff Cutts

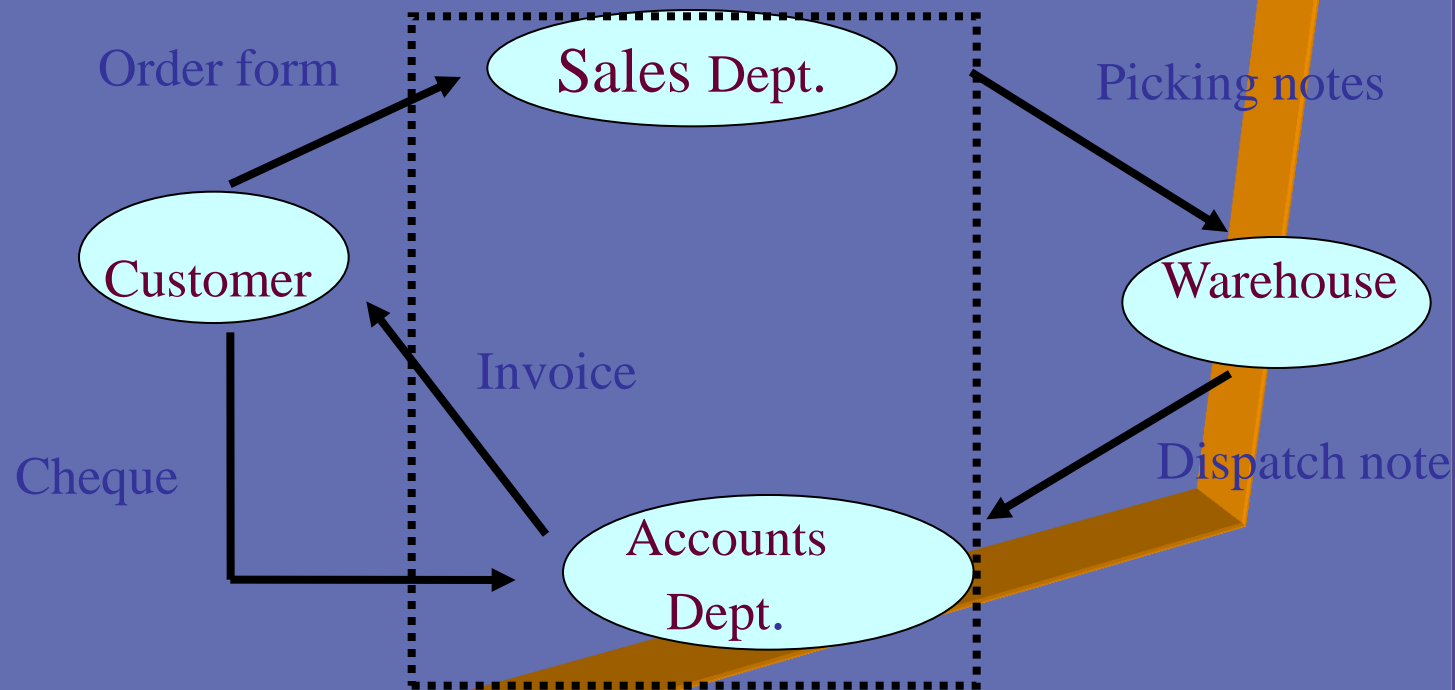
1 ایجاد کنید یک نمودار ایجاد اسناد فیزیکی را . از بیضی برای نمایش یک شخص یا قسمت استفاده کنید .



مدل سازی فرایندی DFD

19

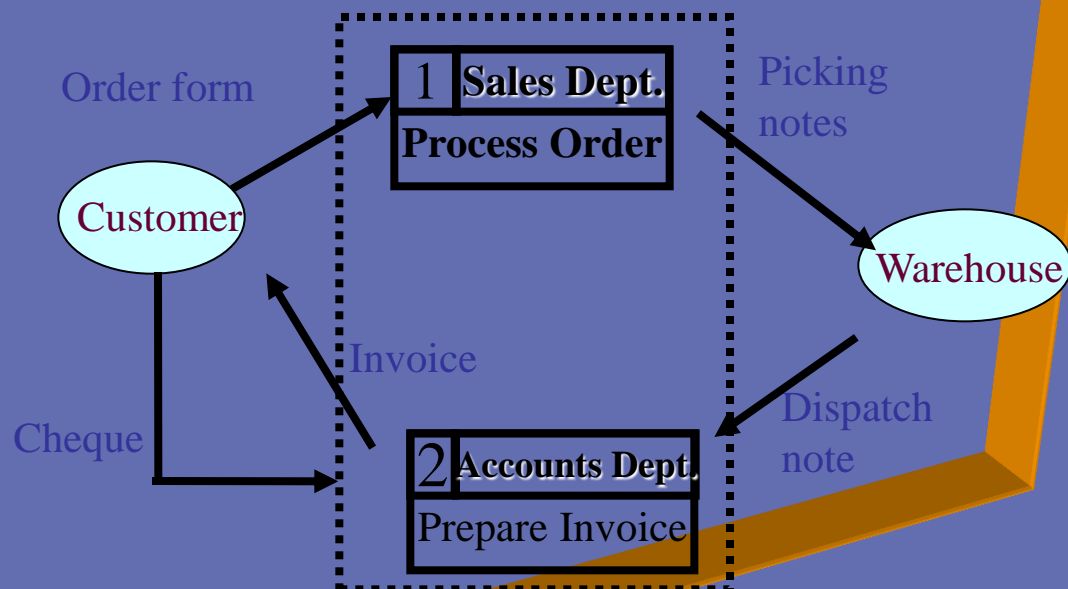
2 محدوده سیستم را مشخص کنید . با نقطه چین محدوده نشان دهید . هر چیزی که خارج از نقطه چین باشد ، یک موجودیت خارجی است .



مدل سازی فرایندی DFD

20

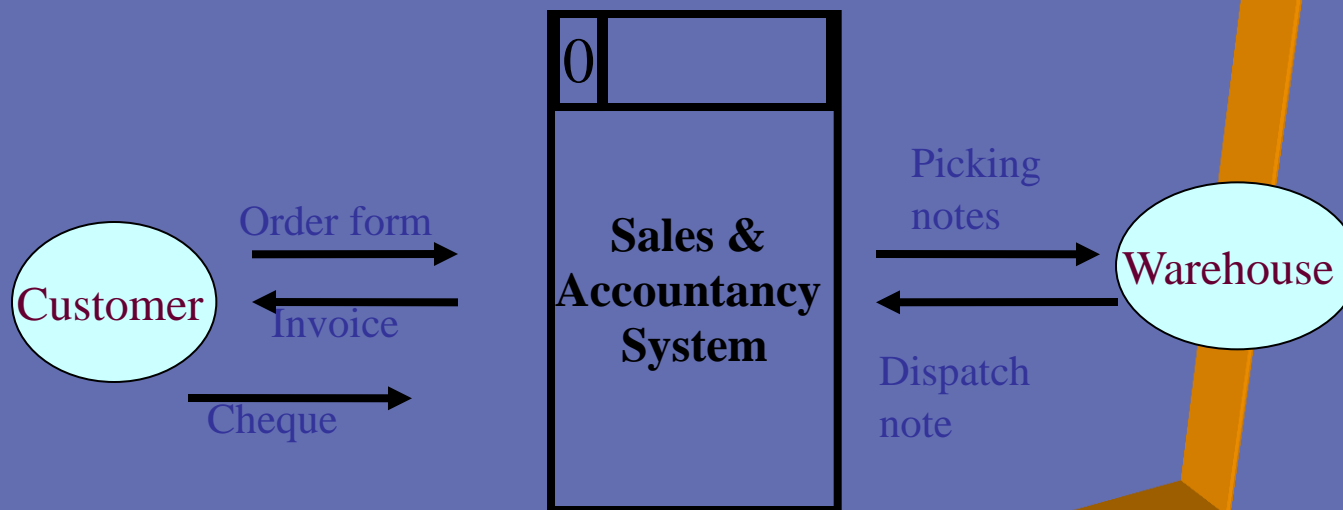
3 ایجاد کنید نمودار جریان داده های فیزیکی کنونی را . بیضی های درون محدوده را تبدیل به فرایند کنید .



مدل سازی فرایندی DFD

21

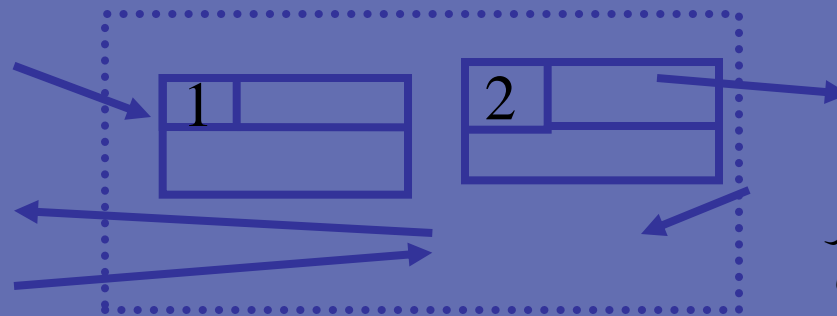
4 ایجاد کنید نمودار سطح صفر را که ارتباطات بیرونی سیستم را نشان می دهد و در واقع شمای کلی سیستم است .



- در موقع ایجاد سطح صفر باید به نکاتی توجه کرد :
- * دیاگرام سطح صفر را طوری بکشید که در یک صفحه جا شود .
- * برای نامگذاری فرایند ها در سطح صفر از سیستم اطلاعاتی استفاده کنید .
- * در نامگذاری از نام های یکتا استفاده کنید .
- * خطوط را از روی هم عبور ندهید .
- * یک نام اختصاصی و یک شماره ارجاع برای هر فرایند درج کنید .
- * به دست بیاورید ورودی و فیدبک کاربر را .

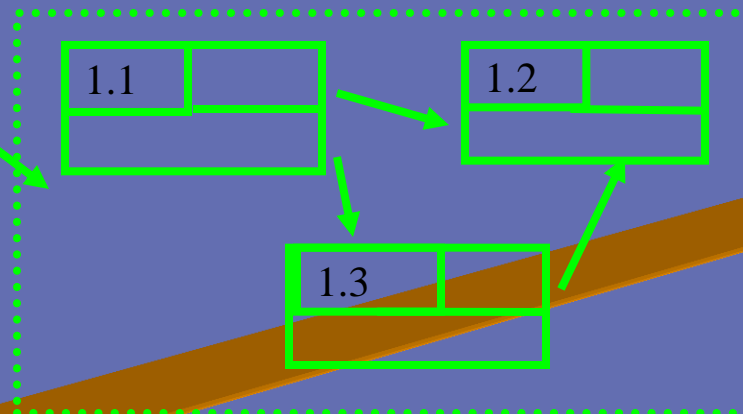
5 ایجاد کنید نمودارهای سطوح پایین تر را . هر فرایند می تواند بسط پیدا کند و تجزیه شود به سطوح پایین تر ، DFD های سطوح پایین تر اطلاعات جزئی تر و دقیق تری را شامل می شوند .

Level 1



پس از رسیدگی های بیشتر
فرایند 1 ممکن است تبدیل
شود به :

Level 2 –
Reference ID is
1.1, 1.2

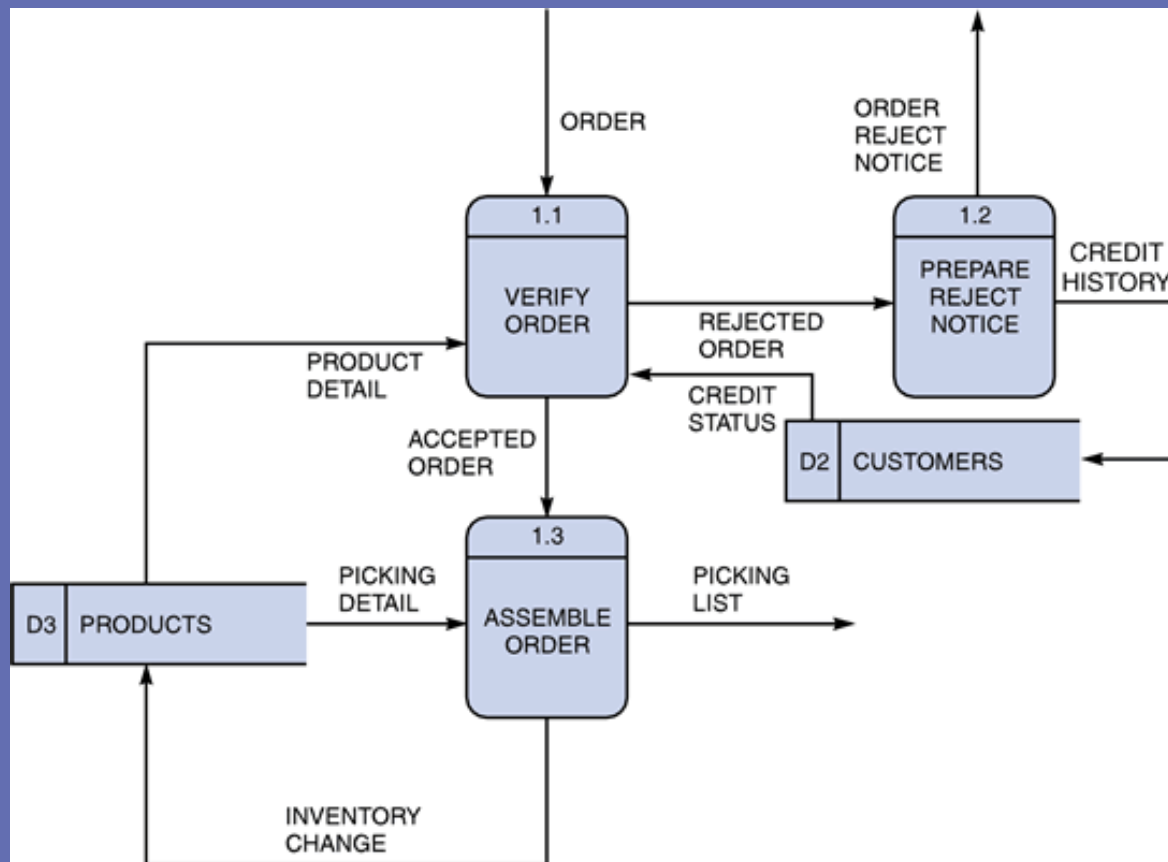


مطمئن شوید که
ورودی ها
و خروجی ها با هم
متناسب
هستند .

مدل سازی فرایندی DFD

24

مثالی از DFD سطوح پایین تر :



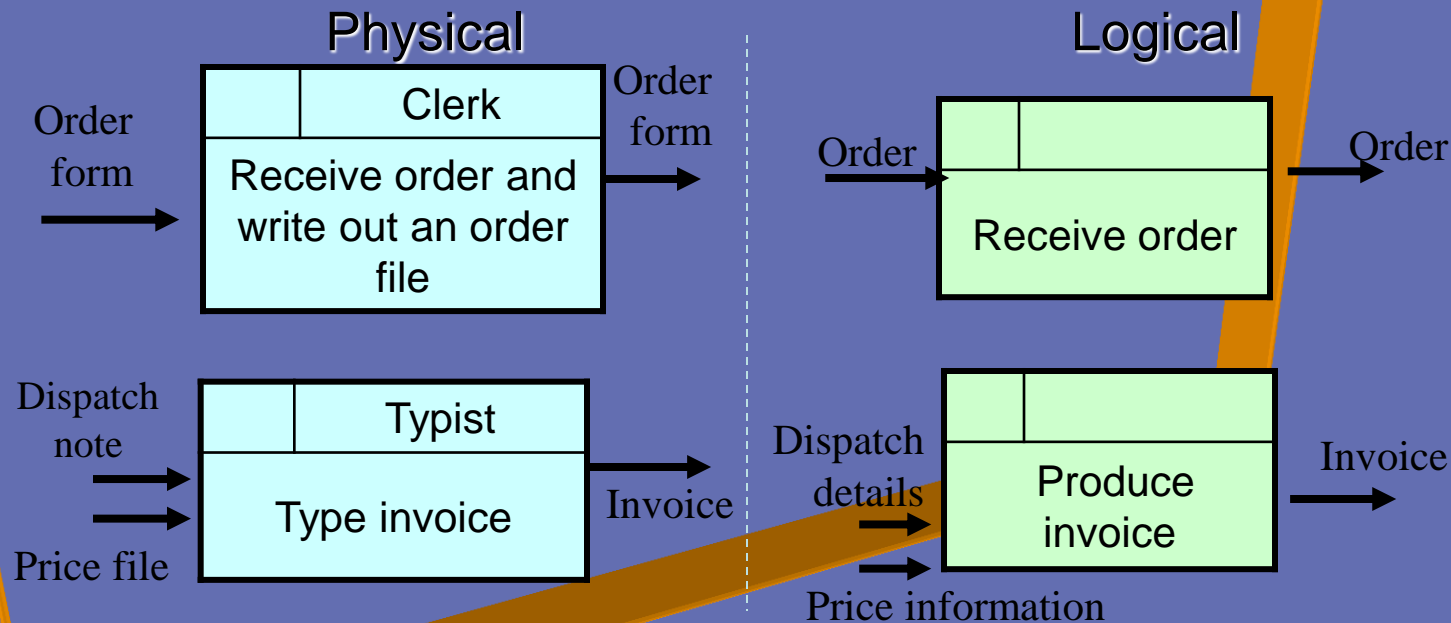
6 مطمئن شوید که به همه جریان داده ها نامی داده شده است . یک DFD کامل نمی شود مگر اینکه همه توابع و جریان داده ها نام با معنی داشته باشند .

7 هر جا که گروه های داده ای نیاز است آنها را اضافه کنید .

از نگاهی دیگر می توان گفت DFD دو نوع دانست :

DFD فیزیکی : که نشان می دهد اشیا چگونه روی می دهند .

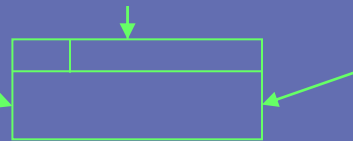
DFD منطقی : که از یک DFD فیزیکی بیرون کشیده می شود و نشان می دهد سیستم چه کاری انجام می دهد .



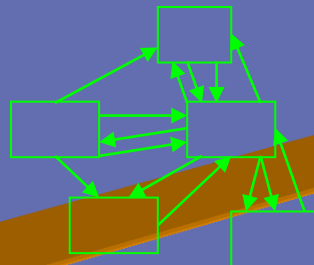
در انتهای این بخش ذکر چند نکته ضروری است :

- * در رسم سطوح پایین تر باید به هم سطحی فرایند ها و توازن بین آنها توجه کرد .
- * یک سیستم ممکن است مجموعه ای از DFD ها با سطوح 0 تا N داشته باشد (3 یا 4 سطح رایج است) .
- * هر سطح حداکثر 7 فرایند می تواند داشته باشد .
- * فرایندهای سطوح پایین تر که دیگر قابل تجزیه نیستند ، توابع اولیه نامیده می شوند .
- * یک فرایند باید خروجی داشته باشد :

بدون خروجی
این فرایند نیست



- * اگر فقط در فایل ها بنویسیم ، بزرگ شدن فایل ایجاد مشکل می کند .
- * از ارتباطات پیچیده باید جلوگیری کرد .



مدل ER (ERD) توسط آقای Peter Chen در سال 1976 معرفی شد و اکنون یکی از اجزای CASE است.

عناصر و المان های کلیدی در ERD عبارتند از:

- موجودیت Entity

به تمام چیزهای قابل تشخیص در محیط کاری کاربر گفته می شود. در پایگاه های داده ها هر جدول یک Entity است.

موجودیت ضعیف: در پایگاه داده ها نمی تواند وجود داشته باشد مگر اینکه نوع دیگری از موجودیت موجود باشد.



PAT-PRESCRIPTION

موجودیت وابسته: وجود آنها بسته به وجود موجودیت های دیگر است.



DORM-OCCUPANT

- صفات ها Attributes

خصوصیات یک موجودیت را توصیف می کنند . در پایگاه داده ها هر یک از ستون های جداول یک صفت است . هر سطر در جداول یک نمونه از آن موجودیت را با صفاتش نشان می دهد . صفت ها چند نوع اند :

1 صفت های ساده : قابل تجزیه نیستند (نام ، جنس ، وضعیت تاهل و ...) .

2 صفت های مرکب : می توانند تجزیه شوند (تاریخ تولد ، آدرس و ...) .

3 صفت های مشتق : که از روی دیگر صفات می توان آنها را به دست آورد (سن و معدل و ...) .

Entity Type (Table)

Student Table

StudentID	First Name	Surname	Date-of-Birth
S1	John	Doe	May 28, 1980
S2	Mary	Moore	April 4th, 1979
...

Attributes
(Columns)

Entity
Instances
(Rows)

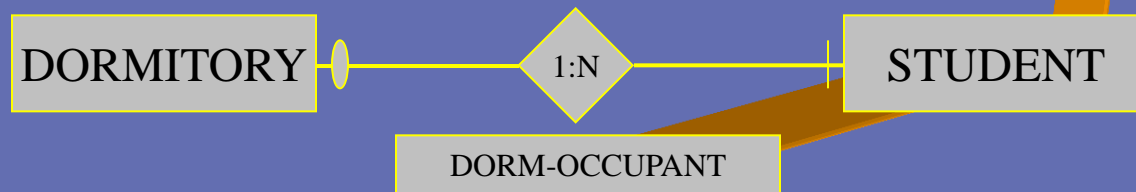
- ارتباط Relationship

راهی است که نشان می دهد موجودیت ها چگونه با هم تعامل دارند . در پایگاه داده ها ارتباط اشاره دارد به اتصالات بین جداول از طریق کلید خارجی .

- درجه ارتباط Cardinality

تعداد ارتباط های یک موجودیت با دیگر موجودیت ها است .

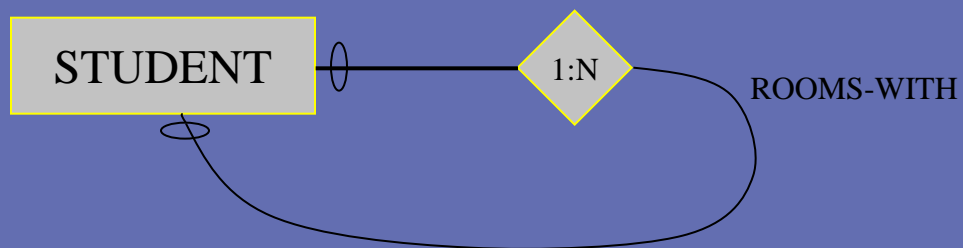
شکل زیر مینیم و ماکزیمم درجه ارتباط را نشان می دهد :



مدل سازی داده ای ERD

32

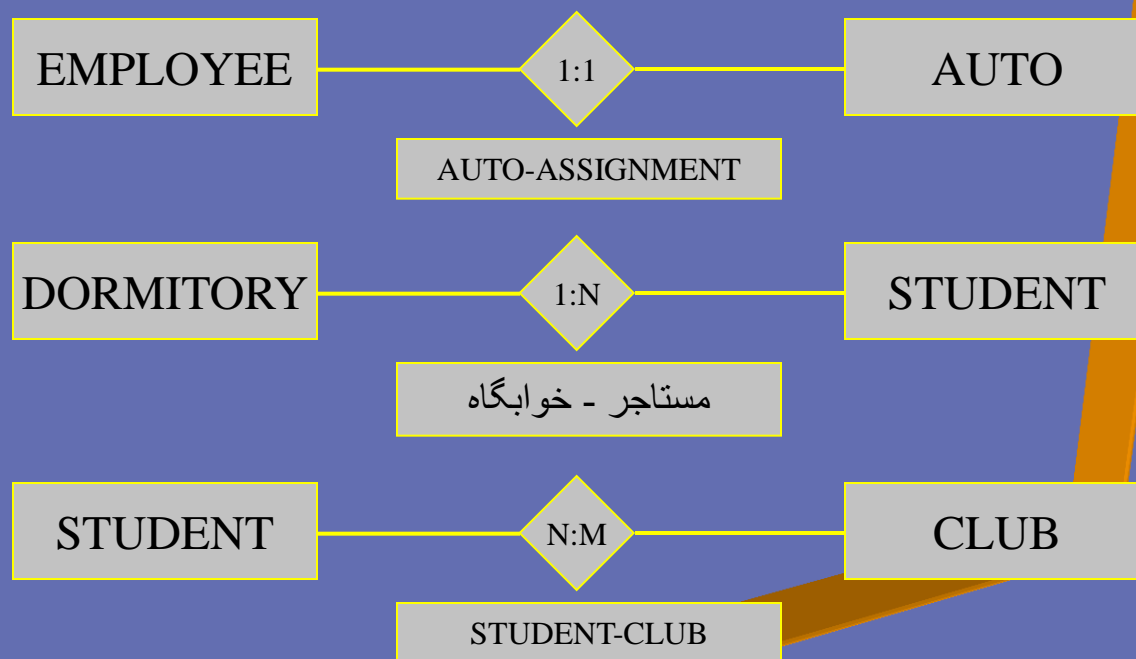
نوعی از ارتباط ، ارتباط بازگشتی است :



مدل سازی داده ای ERD

33

نوعی از ارتباط ها ارتباط های دودویی هستند که سه نوع این ارتباط در مثال زیر مشخص شده است :



نمادهای ERD

EntityType

Entity

—————d

One-to-Zero-Or-One Relationship (Optionality).

—————||

Mandatory One-to-One Relationship

—————α

One-to-Zero-Or-More Relationship

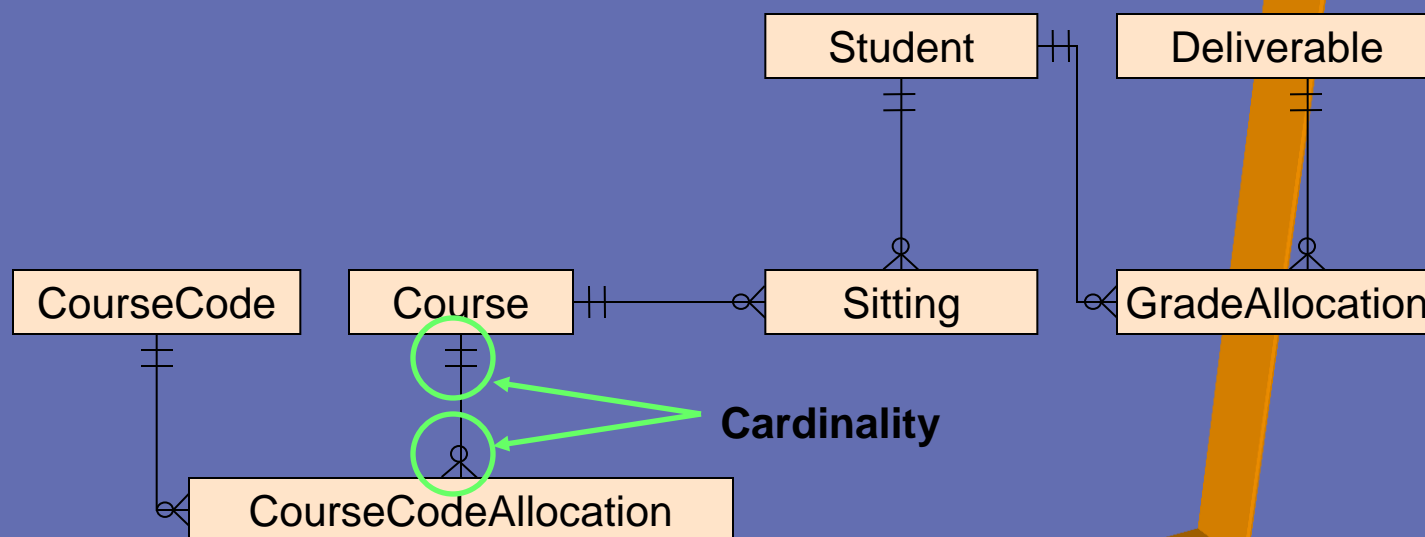
—————K

One-to-One-Or-More Relationship

مدل سازی داده ای ERD

35

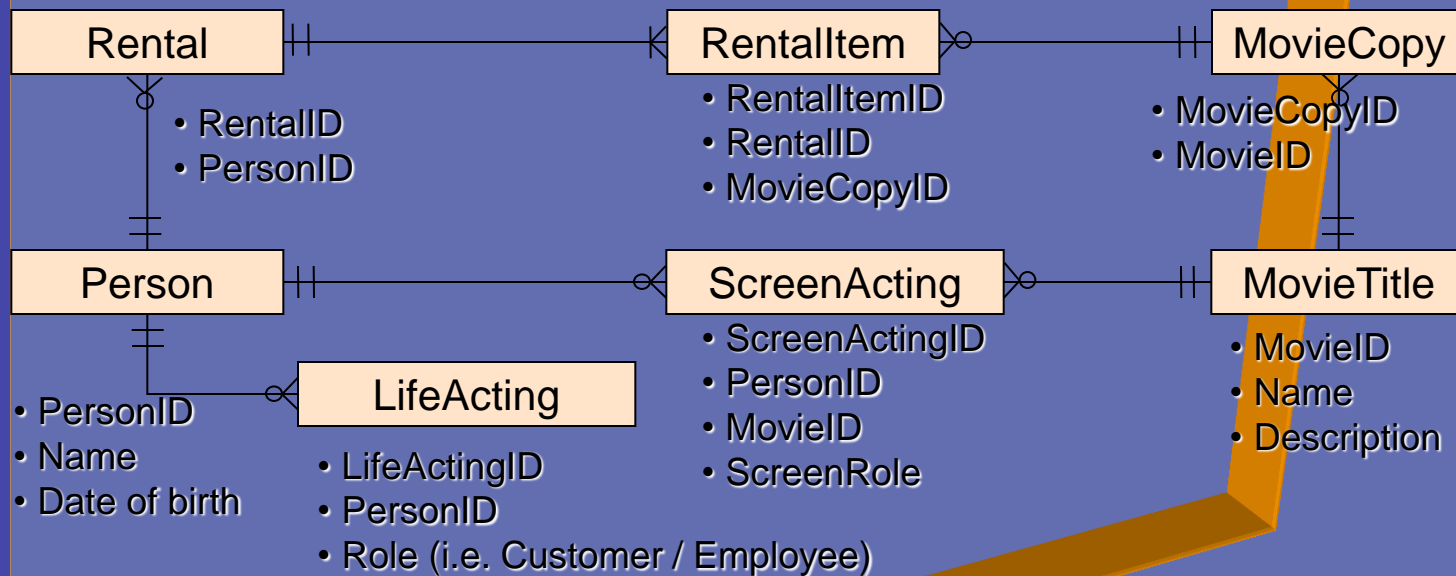
مثال زیر این نمادها را بهتر نشان می دهد :



مدل سازی داده ای ERD

36

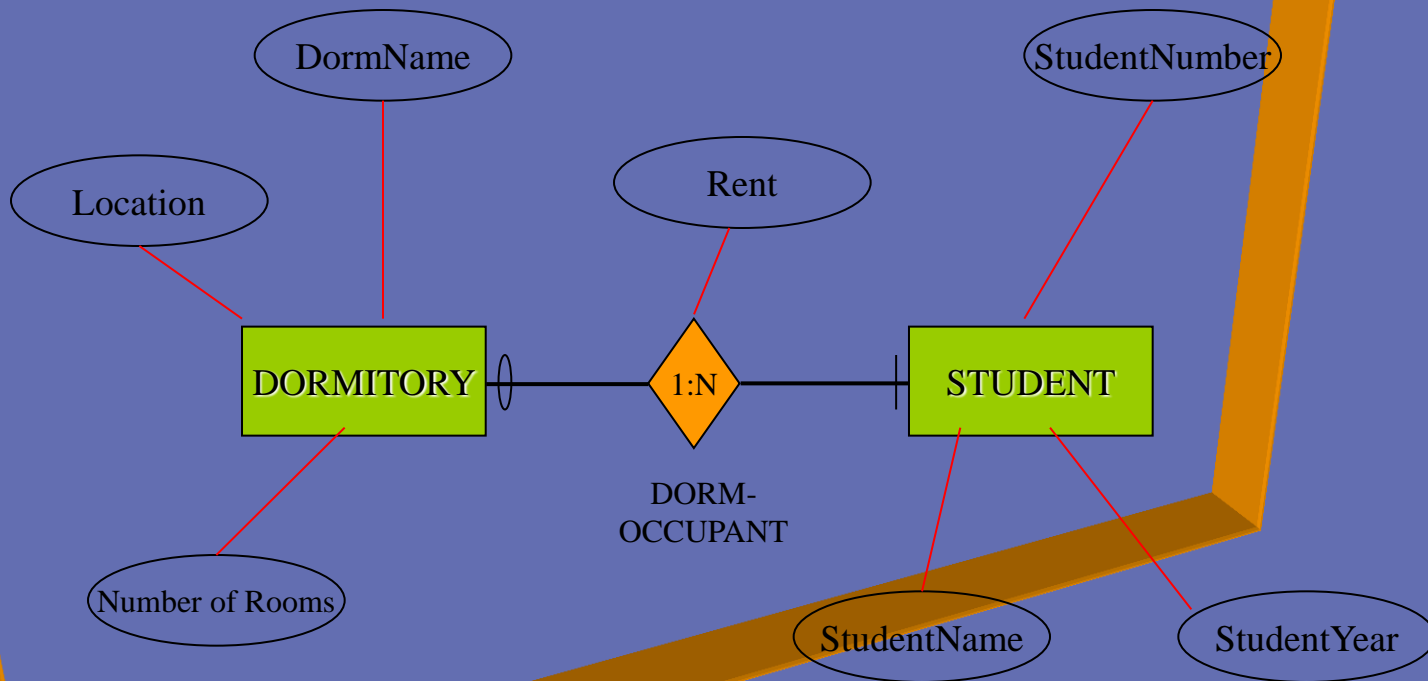
در مثال زیر هم نوع ارتباط موجودیت ها با هم همراه با صفات هر یک از موجودیت ها آمده است : (مدل سازی یک ویدئو کلوپ)



مدل سازی داده ای ERD

37

به عنوان آخرین مثال نمودار یک ERD برای دو موجودیت در زیر آمده است : (معمولا ERD ها را به این شکل نمایش می دهند ، که در آن صفات به صورت یک بیضی نشان داده می شوند و آنها را کنار موجودیت قرار نمی دهند)



سومین تکنیک مدل سازی ، ELH ها هستند که البته در حال حاضر به عنوان یک تکنیک مدل سازی رایج نیست و می توان گفت منسوخ شده است . در این تکنیک تغییراتی را که یک موجودیت در کل سیستم پذیرفته است را توصیف می کند . در اینجا هم بیش از این به این تکنیک پرداخته نمی شود .