دانشگاه فنی و حرفه ای
دانشکده فنی دکتر شریعتی

جزوه درس پایگاه داده ها

مقطع کاردانی

مدرس:

استاد عزیزی

سال تحصیلی ۹۶-۱۳۹۵

فهرست مطالب

فصل اول: مقدمه و مفاهیم اساسی ۳

۱-۱- مقدمه ۴

۱-۲- سیستم های فایلی ۴

۱-۳- روش پایگاهی (بانک اطلاعاتی) ۶

۱-۴- معایب روش های پایگاهی ۷

۱-۵- سیستم مدیریت پایگاه داده (DBMS) ۷

۱-۶- معماری بانک اطلاعاتی ۸

۱-۷- معماری سیستم بانک اطلاعاتی ANSI /Sparc ۹

۱-۸- مدل داده ای ۱۱

۱-۹- نسل های بانک های اطلاعاتی ۱۱

۱-۱۰- تراکنش (transaction) ۱۳

1-11- خواص ACID ۱۴

۱-۱۲- مراحل طراحی بانک اطلاعات ۱۵

۱-۱۳- استقلال داده ای ۱۵

۱-۱۴- عناصر اصلی پایگاه داده ۱۶

فصل دوم: مدلسازی معنایی داده ها ۱۷

۲-۱- انواع معماری سیستم پایگاه داده ۱۷

۲-۲- مراحل کلی ایجاد یک سیستم پایگاهی ۲۰

۲-۳- مدل سازی معنایی داده ۲۰

2-4- نمودار ER ۲۱

۲-۵- صفت ۲۲

۲-۶- ارتباط ۲۴

2-7- نمادهای نمودار ER ۲۶

2-8- مثال اول ترسیم نمودار ER ۲۷

۲-۹- موجودیت ضعیف ۲۸

۲-۱۰- نکات راهنمای تدوین نمودار ER ۳۰

۲-۱۱- ER گسترش یافته ۳۱

۲-۱۲- تخصیص ۳۲

۲-۱۳- تعمیم ۳۳

۲-۱۴- نمودار ER آموزش دانشگاه ۳۵

۲-۱۵- مثال ۲ مدلسازی معنایی ۳۵

۲-۱۶- مثال ۳ ۳۷

۲-۱۷- مثال ۴ ۳۷

۲-۱۸- مثال ۵ ۳۸

۲-۱۹- مثال ۶ ۳۸

۲-۲۰- مثال ۷ ۳۹

۲-۲۱- مثال ۸ ۴۰

فصل سوم : مدل رابطه ای **٤١**
۳-۱- تاریخچه رابطه ۴۱
۳-۲- تعریف دامنه ۴۱
3-3- تعریف رابطه ۴۲
۳-۴- زوج مرتب (تاپل) ۴۲
۳-۵- ویژگی های رابطه ۴۲
۳-۶- تناظر بین اجزا دو مفهوم رابطه و جدول ۴۳
۳-۷- مثالی از نمایش رابطه های کارمند و شعبه ۴۳
۳-۸- کلید در مدل رابطه ای ۴۴
۳-۹- تعریف هایی از جامعیت (integrity) ۴۶
۳-۱۰- طراحی منطقی پایگاه داده ۴۶
3-11- مثال ۱ ..... ۵۷
3-12- یادداشت های فصل ۵۹

فصل چهارم : جبر رابطه ای **٦٠**
۴-۱- معرفی جبر رابطه ای ۶۰
۴-۲- کاربرد جبر رابطه ای ۶۰
۴-۳- هشت عملگر جبر رابطه ای ۶۰
۴-۴- تصویری از عملگرهای جبر رابطه ای ۶۱

فصل پنجم : SQL **٦٧**
۵-۱- زبان میزبان و زبان فرعی داده ۶۷
5-2- دستورات SQL ۶۷

فصل ششم: نرمالسازی **٧٨**
۶-۱- مقدمه ۷۸
۶-۲- سطوح نرمال سازی ۷۸
6-3- وابستگی تابعی (Functional Dependency) ۷۹
۶-۴- فرم نرمال 1NF ۸۱

منابع درس :

بارم بندی درس:

زمان امتحان میانترم:

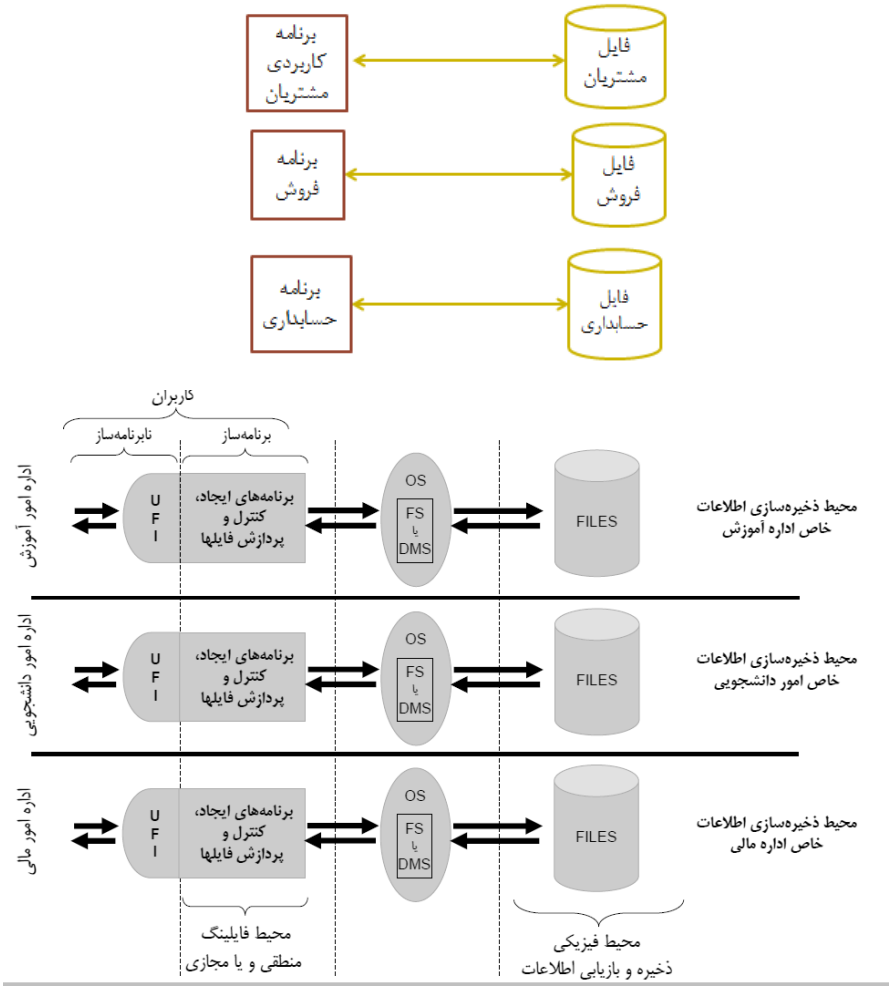
فصل اول: مقدمه و مفاهیم اساسی

۱-۱- مقدمه

اصطلاح **سیستم مدیریت پایگاه داده ها** در معنای عام، یکی از سیستم های ذخیره و بازیابی اطلاعات است. پس از طراحی نسل اولیه سیستم های ذخیره و بازیابی اطلاعات، برای تقویت مکانیزم ها و الگوریتم های مرتبط با ذخیره، بازیابی و پردازش داده ها و به منظور تسهیل در انجام امور فوق سیستم های مدیریت پایگاه داده ها شکل گرفتند.
تکنولوژی پایگاه داده ها از اواسط دهه ۶۰ میلادی و به منظور توسعه سیستم های فایلی ایجاد شدند.

۱-۲- سیستم های فایلی

عبارت است از مجموعه ای از برنامه های کاربردی که سرویس هایی برای کاربران ارائه می دهند (ایجاد گزارش ها و ...).
در این سیستم هر برنامه داده های مربوط به خود را تعریف و مدیریت می کند.



۱-۲-۱- مراحل کار در روش فایلی

مراحل زیر معمولا برای هر زیر محیط به طور جداگانه انجام می شود:

۱) مطالعه و شناخت محیط

۲) انجام مهندسی نیازها

۳) طراحی تعدادی فایل برای ذخیره سازی اطلاعات

   ۱) این فایل ها معمولا به صورت منفرد در سطح سیستم هستند.

   ۲) تعیین ساختار فایل

   ۳) تعیین نحوه دسترسی

   ۴) تعیین اندازه فایل

   ۵) مشخص کردن نحوه ارتباط با فایل های دیگر

   ۶) مشخص کردن عملیات مجاز در فایل ها

۴) برنامه مورد درخواست با یک زبان برنامه نویسی و با توجه به فایل های طراحی شده، تهیه میشود.

۵) طراحی مجموعه ای از تست ها جهت بررسی هر زیر سیستم

۱-۲-۲- معایب روش فایلی

- جدایی داده ها در برنامه های کاربردی مختلف > **عدم وجود یک سیستم کنترل متمرکز**
- تکرار در ذخیره سازی داده ها (افزونگی)
- وابستگی برنامه کاربردی به محیط ذخیره سازی داده ها (فایل ها).
- ناسازگاری فرمت فایل ها در زبان های برنامه سازی مختلف.
- مصرف نابهینه سخت افزار.
- عدم امکان اعمال ضوابط حفظ امنیت داده ها.
- در صورت خرابی نرم افزار یا سخت افزار امکان بازیافت اطلاعات ضعیف یا نا ممکن است.
- دسترسی به فایل داده ها محدود به یک کاربر در هر زمان است.

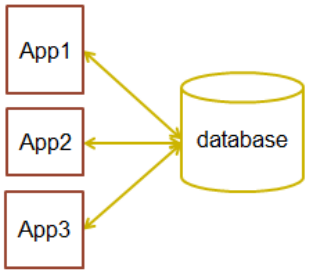
۱-۲-۳- علت معایب روش فایلی

- هیچ کنترلی برای دست یابی و کار روی داده ها به جز آن چه که برنامه کاربردی اجازه می دهد، وجود ندارد.
- داده ها در متن برنامه های کاربردی تعریف شده اند، نه به صورت جداگانه و مستقل.

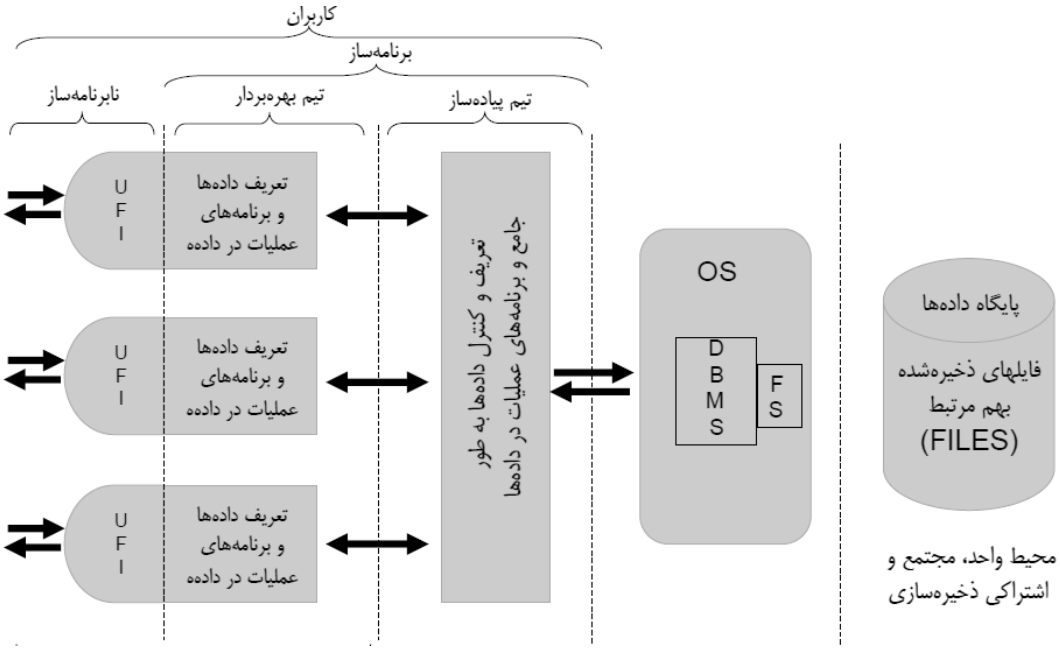
۱-۳- روش پایگاهی (بانک اطلاعاتی)

تعریف پایگاه داده (DB):

به مجموعه ای از داده های به هم مرتبط پایگاه داده می گویند که لزوما این داده ها از نظر فیزیکی در یک مکان قرار ندارند ولی با هم ارتباط منطقی دارند.



۱-۳-۱- شمایی از روش پایگاهی



۱-۳-۲- 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).

۱-۴- 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)

۱-۵- سیستم مدیریت پایگاه داده (DBMS)

در یک تعریف ساده، سیستمی است که به کاربران امکان می دهد عملیات مورد نظرشان را (مانند تعریف داده ها، بازیابی داده ها، ذخیره سازی داده ها) انجام دهند.

DBMS که نرم افزاری پیچیده است میهمان یک سیستم عامل است و از امکانات سیستم عامل در انجام وظایفش بهره می برد.

۱-۵-۱- امکانات DBMS

- Data definition language (DDL) بانک اطلاعاتی را تعریف می کند و به کاربران امکان می دهد که نوع (Type) و ساختار (Structure) داده ها را تعیین و محدودیت (Constraint) هایی را که باید روی داده ها اعمال کرد مشخص کنند.
- Data manipulation language (DML) به کاربران امکان اضافه (Insert) حذف (Delete) و بازیابی (Retrieve) داده ها را از بانک اطلاعاتی فراهم می نماید.

DBMS دسترسی کنترل شده به پایگاه داده را با در نظر گرفتن موارد زیر فراهم می نماید:

- **A security system** (امنیت سیستم)
- **An integrity system** (یکپارچگی سیستم)
- **A concurrency control system** (کنترل همزمانی سیستم)
- **A recovery control system** (کنترل ترمیم سیستم)
- **A user-accessible catalog**(کاتالوگ قابل دسترسی برای کاربر)

۱-۵-۲- کاتالوگ

انواع داده

- داده هایی که کاربر وارد می کند
- داده هایی که سیستم ایجاد می کند
  - ❖ متا دیتا = دادگان= فراداده = داده هایی درباره داده= داده در مورد داده

در این سیستم اطلاعاتی مربوط به حق دستیابی افراد به داده های مختلف، تاریخ ایجاد و به روز رسانی انها، تعداد نسخه های هر پرونده و ترتیب زمانی انها، ایجاد کننده پرونده ها و اندازه هر جدول یا شی و ...، همچنین در بردارنده لغتنامه داده ها می باشد که به این اطلاعات اصطلاحا دادگان می گویند.

لغت نامه داده ها زیر مجموعه کاتا لوگ سیستم است و به دلیل کاربرد از هم جدا شده اند وبرای کار با آن نرم افزار طراحی شده است.

کاربر می تواند محتوای کاتالوگ را به طور ضمنی تغییر دهد.

SYSTABLE:

نام جدول	ایجاد کننده	تاریخ ایجاد	تعداد ستون ها	کلید اصلی	....
S	x	D1	4	S#	
P	x	D2	5	P#	

۱-۵-۳- لغت نامه داده ها

در یک سیستم بانک اطلاعاتی اسامی زیادی مورد استفاده قرار می گیرد و از آن جایی که افراد زیاد و متفاوتی در یک مجموعه بانک اطلاعاتی درگیر هستند، مرجعی برای ایجاد یکنواختی و هماهنگی در نام داده ها و معنای آن ها ضروری است. این مرجع لغتنامه داده نام دارد.

شبیه لغتنامه های معمولی تمامی اسامی استفاده شده در سیستم و معنای ان ها را در برمی گیرد. (مانند جدول یا شی یا صفت و ...)

از اشتباهاتی شبیه واردکردن یک نام با دو معنای مختلف (homonym) و یا دو نام برای یک مفهوم (synonym) جلوگیری می کند.

در یک سیستم بانک اطلاعاتی اسامی زیادی مورد استفاده قرار می گیرد و از آن جایی که افراد زیاد و متفاوتی در یک مجموعه بانک اطلاعاتی درگیر هستند، مرجعی برای ایجاد یکنواختی و هماهنگی در نام داده ها و معنای آن ها ضروری است. این مرجع لغتنامه داده نام دارد.
شبیه لغتنامه های معمولی تمامی اسامی استفاده شده در سیستم و معنای ان ها را در برمی گیرد. (مانند جدول یا شی یا صفت و ...)
از اشتباهاتی شبیه واردکردن یک نام با دو معنای مختلف (homonym) و یا دو نام برای یک مفهوم (synonym) جلوگیری می کند.

۱-۶- معماری بانک اطلاعاتی

معماری سه سطحی ANSI/SPARC استانداردی برای طراحی سیستم های پایگاه داده است. کاربران مختلف در سطوح مختلف توسط یک زبان با سیستم تعامل می کنند.

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

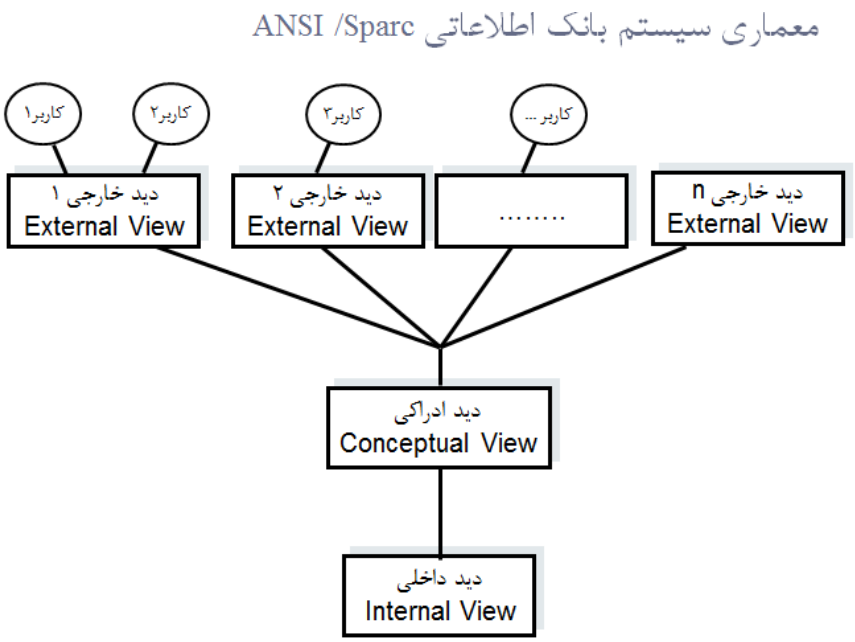
هدف معماری سه سطحی این است که امکاناتی را فراهم کند تا کاربران بتوانند با دیدگاه های شخصی خود به داده موردنیاز دسترسی پیدا کنند. یعنی هر کاربری بتواند به داده مشترک دسترسی پیدا کند اما دید خاص خود را داشته باشد.

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.

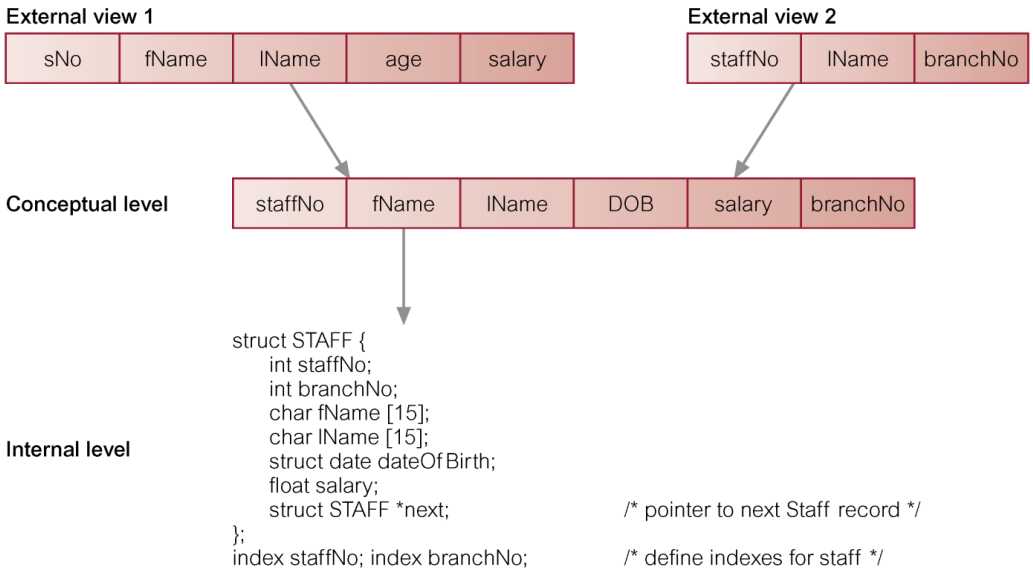
۱-٦-۱- اهداف معماری سه سطحی

- هر کاربر باید بتواند به داده های بانک اطلاعاتی دسترسی داشته باشد. با این توجه که هر کاربردید خاصی از داده ها 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.
- تغییرات در جنبه های فیزیکی ذخیره سازی نباید روی ساختار داخلی پایگاه داده تاثیر بگذارد.
- مدیر پایگاه داده (DBA) باید بتواند ساختار مفهومی پایگاه داده را، بدون تاثیر بر کاربران، تغییر دهد.

۱-۷- معماری سیستم بانک اطلاعاتی ANSI /Sparc



۱-۷-۱- تفاوت های بین سه سطح معماری



۱-۷-۲- سطح خارجی

سطح 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) می گویند. برای تعریف شمای خارجی از یک مدل داده استفاده می شود که معمولا همان است که در سطح ادراکی بکار رفته است.

۱-۷-۳- سطح ادراکی

سطح ادراکی کل داده های که در پایگاه داده ذخیره می شوند و ارتباط مابین آنها را شرح می دهد. یعنی داده هائی درباره انواع موجودیت ها و ارتباط آنها در محیط عملیاتی را توصیف می کند.

سطح ادراکی دید طراح پایگاه داده از داده های ذخیره شده در پایگاه داده است. 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. شمای ادراکی کلیه داده ها و ارتباط بین آنها را توصیف می کند. علاوه بر این رویه های شناسائی و قیدهای جامعیت را نیز دربر می گیرد.

برای کسب استقلال داده، شمای ادراکی تنها درگیر معنی داده است و جنبه های نمایش داده، سازماندهی فیزیکی و استراتژی های دستیابی ندیده 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.

۱-۷-٤- سطح 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 می شود. 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، شاخص ها 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.

۱-۸- مدل داده ای

مجموعه ای منسجم از مفاهیم برای توصیف داده ها، روابط بین آن ها، و محدودیت های آن ها در یک سازمان.

- مدل داده شامل قسمت های زیر است:
  - ❖ قسمت ساختاری (structural part)
  - ❖ قسمت عملیاتی (manipulative part)
  - ❖ مجموعه ای از قوانین یکپارچگی (set of integrity rules)
- هدف (Purpose):
  - ❖ نمایش داده ها به روشی قابل فهم

۱-۹- نسل های بانک های اطلاعاتی

- نسل اول:
  - ❖ سلسله مراتبی (Hierarchical)
  - ❖ شبکه ای(Network)
- نسل دوم:
  - ❖ رابطه ای (Relational)
- نسل سوم:
  - ❖ رابطه ای شیءگرا (Object Relational)
  - ❖ شیءگرا (Object-Oriented)

۱-۹-۱- مدل سلسله مراتبی

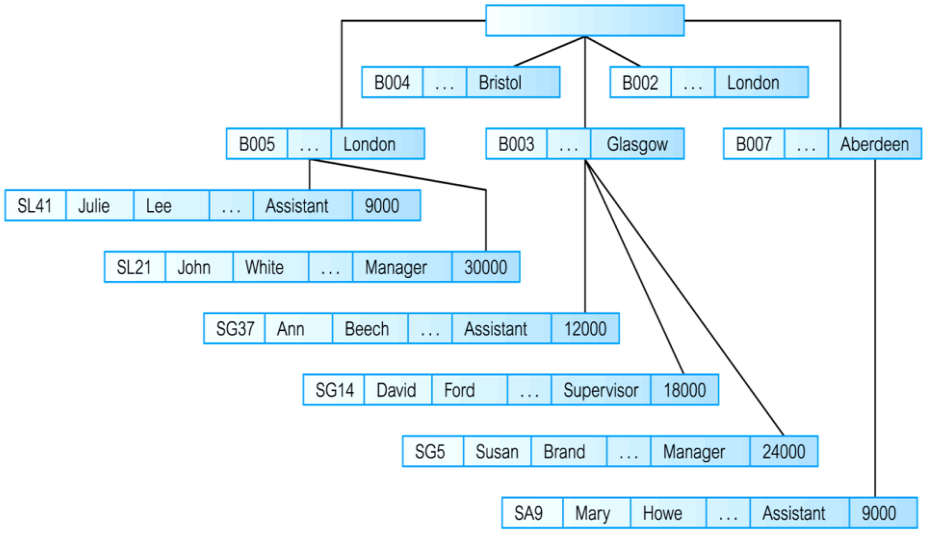
قدیمی ترین مدل برای طراحی پایگاه داده مدل سلسله مراتبی (hierarchical model)است، که در اوایل دهه ۶۰ توسط IBMبرای سازماندهی دنیای تجارت به شکل سلسله مراتبی پیشنهاد شد.

مدل اجازه تکرار اطلاعات را توسط ارتباطات والد/فرزند می دهد؛ یعنی هر گره در هر سطح می تواند تعدادی گره وابسته یا فرزند داشته باشد که بعنوان والد آنها محسوب می شود. 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.

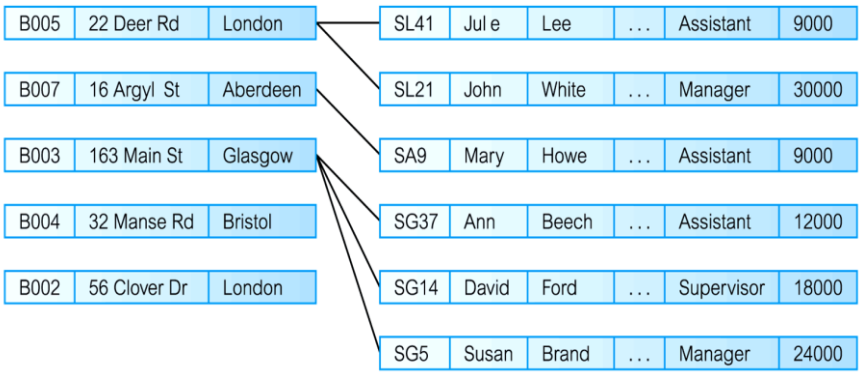


۱-۹-۲- مدل داده شبکه ای

درمقایسه با مدل سلسله مراتبی که ساختمان های داده ای به صورت درختی از رکوردها سازماندهی می شود و هر رکورد آن یک والد و چند فرزند دارد، مدل شبکه اجازه رکوردهائی با چند والد و چند فرزند را می دهد.

عملیات ذخیره و بازیابی پیچیده تر از مدل سلسله مراتبی است.

مدل شبکه انعطاف پذیری بیشتری نسبت به سلسله مراتبی دارد.



۱-۹-۳- مدل رابطه‌ای

- در مدل رابطه‌ای داده‌ها بصورت رکوردهای مرتبط سازماندهی می‌شوند و بانک اطلاعات بصورت مجموعه‌ای از رابطه‌ها طراحی می‌شود.
- علل موفقیت مدل رابطه‌ای:
  - ❖ سادگی
  - ❖ پشتوانه‌ی تئوریک قوی

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

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

۱-۱۰- تراکنش (transaction)

- هر گونه برنامه‌ای که توسط کاربر در محیط بانک اطلاعات اجرا می‌شود تراکنش نام دارد.
- تراکنش همواره به DBMS تسلیم شده و DBMS در اعمال هرگونه کنترل، به تعویق انداختن و یا ساقط کردن آن تصمیم‌گیری می‌کند.
  - ❖ هدف اصلی از این کنترل‌ها حفظ جامعیت و صحت بانک اطلاعات است.

- داده‌های بانک اطلاعات را مانا (persistent) می‌نامند زیرا برنامه‌ها می‌آیند و می‌روند اما داده‌ها می‌مانند یعنی مثل برخی از داده های محیط برنامه نویسی (متغیرها) نیست و در واقع داده های دیتابیس بنا به درخواست کاربر مجاز تغییر میکنند اما با پایان رفتن برنامه از بین نمی روند.
- مثال:
  - ❖ یک حساب بانکی به طور متناوب به آن پول واریز و برداشت می شود و مهم است که موجودی حساب در هر لحظه درست باشد.

۱-۱۱- خواص ACID

۱-۱۱-۱- یکپارچگی

- به معنی "همه" یا "هیچ"
- منظور از یکپارچگی این است که یا تمامی دستورالعمل‌های یک تراکنش باید اجرا شود و یا هیچکدام از آنها.
- مثال: تراکنشی برای انتقال مبلغی پول از حسابی به حساب دیگر.
  این تراکنش شامل دو بخش است.
  - ❖ بخش اول، پول را از حساب اول برداشت می‌کند.
  - ❖ بخش دوم، همان پول را به حساب دوم واریز می‌کند.
  - ❖ فرض کنیم بخش اول تراکنش اجرا شود اما ناگهان ارتباط با ماشین دوم قطع گردد و بخش دوم قابل انجام نباشد، بدیهی است که باید پول را به حساب اول بازگرداند تا صحت و جامعیت اطلاعات حفظ شود.
  - ❖ این عمل معادل این است که بگوییم هیچ عملی انجام نشده است.

۱-۱۱-۲- همخوانی

- این خاصیت می‌گوید که تراکنش باید تمامی قوانین جامعیت بانک اطلاعات را رعایت کند. (فرض می‌شود که تراکنش یک برنامه صحیح است.)
- مثال: در برنامه انتقال پول اگر مبلغ برداشت شده با مبلغ واریز شده به حساب دیگر برابر نباشد تراکنش غلط است. (چنین مواردی توسط نظام مدیریت بانک اطلاعات به طور اتوماتیک قابل کنترل نیست)

■ تراکنش ممکن است دو نوع پایان داشته باشد:
- ❖ پایان ناموفق : abort
- ❖ پایان موفق : commit > تثبیت

۱-۱۱-۳- انزوا

- اثر تراکنش‌های همروند روی یکدیگر چنان است که گویا هر کدام در انزوا هستند. (همروندی تراکنش‌ها کنترل می‌شود تا اثر مخرب روی هم نداشته باشند.)
- این عمل توسط بخشی از DBMS به نام واحد کنترل همروندی (concurrency control) انجام می‌شود.

۱-۱۱-٤- پایایی

- تراکنش‌هایی که به مرحله انجام (commit) برسند، اثرشان ماندنی است و هرگز بطور تصادفی از بین نمی‌رود.
  - ❖ مثال : اگر مبلغی به حسابی واریز شد و تراکنش مربوطه commit اعلام شد، این عمل باید در جایی دیگر نیز ثبت شده باشد.
- دو عمل یکپارچگی و پایایی توسط واحدی از DBMS به نام واحد مدیریت بازگرد (recovery management) انجام می‌گیرد.

۱-۱۲- مراحل طراحی بانک اطلاعات

۱. بررسی و تجزیه و تحلیل
   - ❖ امکان سنجی، شناخت نیازها و ...
۲. طراحی ادراکی عام
   - ❖ طراحی شماتیک مانند مدل ER
۳. طراحی ادراکی خاص
   - ❖ انتخاب مدل، تبدیل شماتیک به ساختار مدل و ...
۴. طراحی فیزیکی

۱-۱۳- استقلال داده ای

سیستم های قدیمی وابسته به داده بودند به این معنی که روش سازماندهی داده در دستگاه جانبی و روش دسترسی به آن توسط برنامه و در منطق آن ساخته می شدند. در چنین سیستمی تغییر در ساختار دخیره سازی یا استراتژی دستیابی بدون تاثیر روی برنامه غیرممکن است.

یکی از مزایای سیستم پایگاه داده استقلال داده است. استقلال داده به معنی مصونیت برنامه ها از تغییرات ساختار ذخیره سازی و استراتژی دستیابی است. ساختار ذخیره سازی بدون تاثیر روی دیدهای کاربران است می تواند تغییر کند. پایگاه داده و برنامه های کاربری هم می توانند مستقل از یکدیگر تغییر کنند.

دو نوع استقلال داده وجود دارد:

استقلال فیزیکی داده= نشان می دهد تا چه اندازه شمای داخلی می تواند بدون تاثیر روی برنامه های کاربردی تغییر کند. مثلا اگر تغییری در ذخیره سازی داده ها داده شود (نوع دیسک عوض شود) برنامه های کاربردی هیچ تغییری نمی کنند.

استقلال منطقی داده = نشان می دهد تا چه اندازه شمای ادراکی می تواند بدون تاثیر روی برنامه کاربردی تغییر کند.

۱-۱۴- عناصر اصلی پایگاه داده

- سخت افزار
- نرم افزار
- کاربر
- داده

۱-۱٤-۱- سخت افزار

- سخت افزار ذخیر ه سازی

منظور همان رسانه های ذخیره سازی است که معمولا برای ذخیره سازی داده ها از دیسک های سریع با ظرفیت بالا استفاده می شود.

- سخت افزار پردازشی

منظور همان کامپیوتر یا ماشین است .ماشین های خاص برای محیط های بانک اطلاعاتی نیز طراحی و تولید شده اند. این ماشین ها از نظر معماری، حافظه اصلی و سایر اجزا از ویژگیهای خاصی برخوردارند.

- سخت افزار ارتباطی
  - برای برقراری ارتباط های راه دور و نزدیک
  - معماری متمرکز، مشتری خدمتگذار، توزیع شده، موازی،چند پایگاهی، موبایل و ...

۱-۱٤-۲- نرم افزار

- نرم افزار کاربردی

نرم افزاری است که کاربر(برنامه ساز)باید برای تماس با سیستم بانک اطلاعاتی آماده کند.

- نرم افزار سیستمی

بین بانک اطلاعاتی فیزیکی که داده ها بصورت فیزیکی در آن ذخیره می شوند و کاربران سیستم، لایه ای از نرم افزار موسوم به مدیر بانک اطلاعاتی قرار دارد .سیستم مدیریت بانک اطلاعاتی نرم افزاری است که به کاربران امکان می دهد که پایگاه از دید خود را تعریف کنند و به پایگاه خود دستیابی داشته باشند، با پایگاه خود کار کنند و روی آن کنترل داشته باشند.

۱-۱٤-۳- کاربران پایگاه داده

کاربران پایگاه داده به سه دسته تقسیم می شوند:

- مدیر بانک اطلاعاتی (DBA)
  - ❖ که یک فرد حرفه ای 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
  - ❖ مسئولیت طراحی و تصمیم‌گیری برای کلیه موارد یک سیستم بانک اطلاعاتی را دارد.
- کاربر برنامه نویس DBP
  - ❖ این گروه 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 پایانه ها به سیستم دسترسی دارند و از طریق برنامه‌های تهیه شده داده‌ها را در حیطه نظارت DBMS دستکاری می‌نمایند.

فصل دوم: مدلسازی معنایی داده ها

۲-۱- انواع معماری سیستم پایگاه داده

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

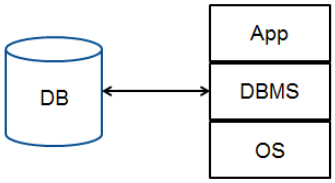
در اساس دو نوع معماری برای سیستم پایگاهی وجود دارد:

معماری متمرکز و معماری نامتمرکز

۲-۱-۱- معماری متمرکز

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

زمانی می گوییم سیستم متمرکز است که روی یک سیستم باشد.



۲-۱-۲- معماری نامتمرکز

معماری نامتمرکز خود دارای انواع مختلفی می باشد:

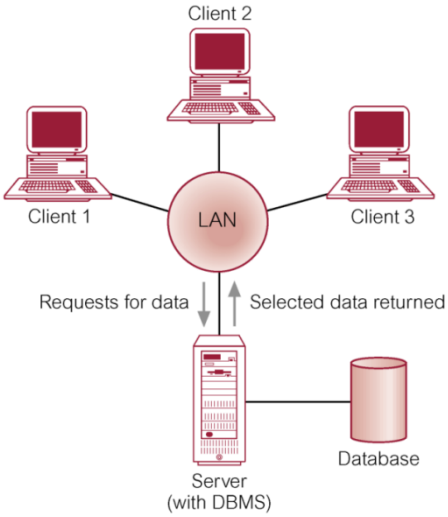
- معماری مشتری-خدمتگزار
- معماری توزیع شده
- معماری چند پایگاهی
- معماری با پردازش موازی
- و ...

۲-۱-۳- معماری مشتری-خدمتگزار

- تعریفی از معماری مشتری-خدمتگزار:
- 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.

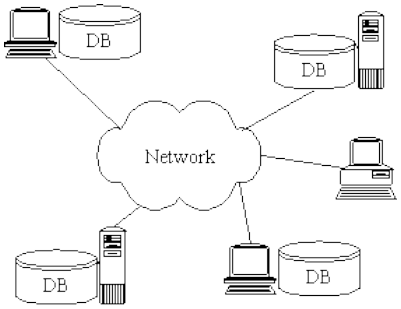
۲. دسته ای که انجام آن توسط مشتری خواهد بود.



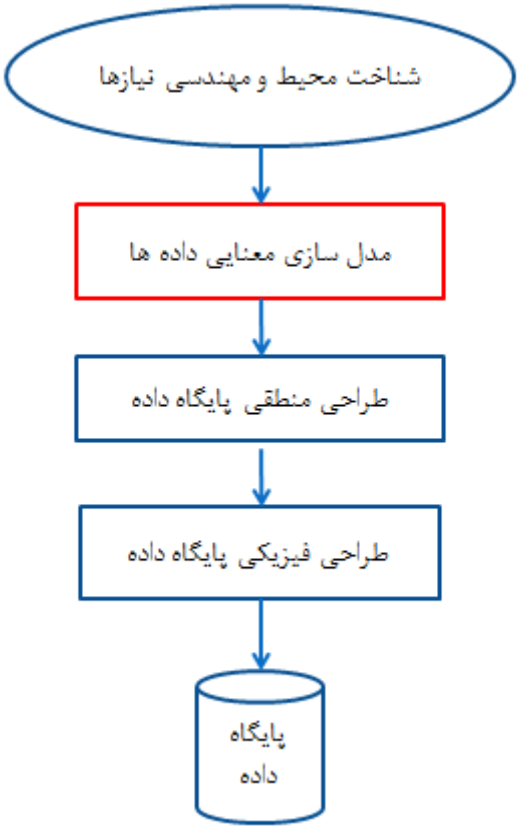
۲-۱-٤- معماری توزیع شده

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 گرفته اند.

میتوان گفت در این معماری تعدادی پایگاه داده های 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.



۲-۲- مراحل کلی ایجاد یک سیستم پایگاهی



۲-۳- مدل سازی معنایی داده

تعریف مدل سازی معنایی داده ها:

عبارتست از ارائه مدلی از محیط عملیاتی به کمک مفاهیمی مستقل از موضوعات مربوط به نمایش منطقی و فیزیکی داده ها. مدل سازی معنایی را در بعضی کتب طراحی ادراکی یا مفهومی یا (conceptual design) می نامند.

ارائه یک مدل کلی از آنچه که می خواهیم ایجاد کنیم، از پایگاه داده در بالاترین سطح انتزاعی و با توجه به معنایی که کاربر برای داده ها قائل است.

انتزاعی ، ذهنی ، مجرد (معنا) جدا از جنبه های نمایش منطقی و جدا از جنبه های مربوط به سطح پیاده سازی. (یعنی وارد جزئیات نمی شویم.)

۲-۳-۱- روش های رایج در مدل سازی

برای مدلسازی معنایی داده ، روش هایی داریم :

- روش موجودیت- ارتباط (ER)
- روش موجودیت- ارتباط توسعه یافته (EER)
- روش UML (فقط خاص مدلسازی معنایی داده نیست و برای طراحی سیستم های نرم افزاری استفاده می شود)

۲-۴- نمودار ER

اولین طرح پایگاه داده ها و مدل کلی آن در بالاترین سطح انتزاع می باشد. نموداری است کـه در آن سـه مفهـوم اساسـی یعنـی (موجودیـت، صفت، ارتباط) نمایش داده می شود.

۲-٤-۱- موجودیت

تعریف موجودیت: مفهوم کلی آنچه که می خواهیم در موردش اطلاع داشته باشیم {از یک محیط مشخص} مثال: محیط دانشکده بعضی از موجودیت ها می شوند استاد ، دانشجو ، برنامه ، کلاس ، کارمند و...

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

- هر نوع موجودیتی یک **نام** دارد.
- هر نوع موجودیتی یک **معنا** دارد.
- هر نوع موجودیتی **نمونه** هایی دارد.
- هر نوع موجودیتی مجموعه ای از **صفات** دارد.
- **ارتباط** هایی با نوع های دیگر دارد.
- عدم وابستگی و یا وابستگی به یک نوع دیگر

۲-٤-۲- نمونه موجودیت

- تمام نمونه های مشخص از هر نوع موجودیت در یک محیط، مجموعه ای به نام نمونه های موجودیت را تشکیل می دهند.
- نمونه موجودیت های خودرو : پیکان، پژو، سمند، بنز و ...
- نمونه موجودیت های دانشجو: کریمی، احمدی، اکبری

- نمونه موجودیت های درس: پایگاه داده، شبکه، سیستم عامل

۲-٤-۳- حالات یک موجودیت

موجودیت مستقل (قوی)، موجودیتی است که مستقل از هر موجودیت دیگر و به خودی خود، در یک محیط مشخص مطرح باشد.
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 نمونه های 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.

۲-۵- صفت

- **تعریف صفت:** هر نوع موجودیتی خصیصه هایی (ویژگی هایی) دارد که حالت یا وضع آن موجودیت را توصیف می کند.
- مثال: دانشجو = موجودیت ، صفات آن: شماره دانشجویی ، نام، رشته
- 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.
  - مثال: نمره، نوع: اعشاری– مجموعه مقادیر مجاز: ۰ تا ۲۰

۲-۵-۱- محدودیت های یک صفت (Constraint)

- محدودیت های دامنه ای:
  - نوع و مقدار
- محدودیت های یکتایی مقدار (uniqueness)
  - مثال: کد ملی
- محدودیت هیچ مقدار ناپذیری

- مثال: صفتی که همیشه باید مقادیرش معلوم باشد.
- محدودیت بین صفات
  - مثال انواع وابستگی ها (موضوع فصل آخر)

۲-۵-۲- صفت ساده و مرکب

صفت ساده: صفتی است که مقدار آن از لحاظ معنایی تجزیه نشدنی می باشد، به این معنا که اگر مقدار آن را به اجزائی تجزیه کنیم، مقادیر هر جز فاقد معنا می 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.

توجه : در بانک اطلاعات رابطه ای امکان تعریف صفات مرکب وجود ندارد. (توضیحات بیشتر در فصل بعد)

۲-۵-۳- صفت تک مقداری و چند مقداری

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

۲-۵-٤- صفت کلیدی (شناسا)

صفتی که باید یکتایی مقدار داشته باشد و بعبارت دیگر مقدار این صفت بین نمونه های مختلف تکرار نمیشود. بعنوان مثال، شماره دانشجویی برای موجودیت دانشجو یک صفت کلید است.

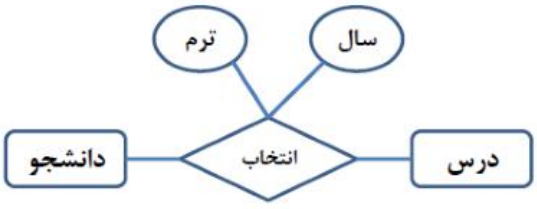
در مدلسازی معنایی، برای هر موجودیت باید یک شناسه مشخص کرد. اما یک موجودیت ممکن است بیش از یک شناسه داشته باشد. مثلا، کد ملی نیز یک شناسه دیگر برای موجودیت دانشجو است. در چنین حالاتی، یکی از شناسه ها را باید بعنوان شناسه اول (اصلی) مشخص کرد. صفت دیگر نیز می تواند بعنوان شناسه دوم (فرعی) معرفی شود.

۲-۵-۵- صفت مجازی، مشتق (Derived)

صفت ذخیره شده صفتی از یک موجودیت است که مقادیرش برای نمونه های آن موجودیت در پایگاه داده ها ذخیره شود.
صفتی که مقدارش در پایگاه داده ذخیره نشده باشد. این صفت وجود خارجی ندارد. ولی از روی دیگر صفات قابل محاسبه (قابل اشتقاق) است. مانند سن

۲-۶- ارتباط

- موجودیت های هر محیط عملیاتی با هم ارتباطاتی دارند. ارتباط وابستگی بین چند موجودیت را نشان می دهد. یک (relationship) یک وابستگی معنی دار بین دو یا چند نوع موجودیت مختلف است.
- ارتباطات معادل افعال یا مفاهیمی نظیر خرید کردن، تعمیر کردن، عضو بودن، رئیس یک سازمان بودن هستند.
- نکات مربوط به ارتباط:
  - هر نوع ارتباط یک **نام** دارد.
  - هر نوع ارتباط **شرکت کنندگانی** دارد تعداد شرکت کنندگان را در درجه (n) می گویند.
  - می تواند صفت هایی داشته باشد.

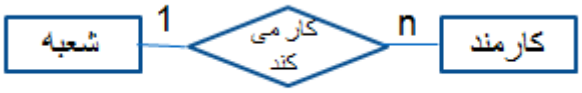


۲-٦-۱- چندی ارتباط

- چندی ارتباط یا بعبارتی نوع تناظر بین نمونه های دو موجودیت حاضر در ارتباط را نشان می دهد و مـی توانـد (1:1) ، (1:n) و (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 می تواند با یک نمونه از موجودیت اول ارتباط داشته باشد. در واقع همان مفهوم تناظر یک به یک در نظریه مجموعه ها است.
  - مثال: بعنوان مثال، اگر درجه ارتباط بین موجودیت های دانشجو و درس (۱:۱) باشد مفهوم آن به شکل زیر خواهد بود.

ارتباط یک به یک

- ارتباط (1:n) : 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.
- اما اگر درجه ارتباط بین موجودیتهای دانشجو و درس یک به چند باشد، مفهوم ارتباط چنین است:



ارتباط یک به چند

- هر دانشجو می تواند چند درس را بگیرد. اما هر درس میتواند فقط توسط یک دانشجو گرفته شود.
- 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(مطابق شکل زیر)، 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

۲-٦-۲- 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 می توانند 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.

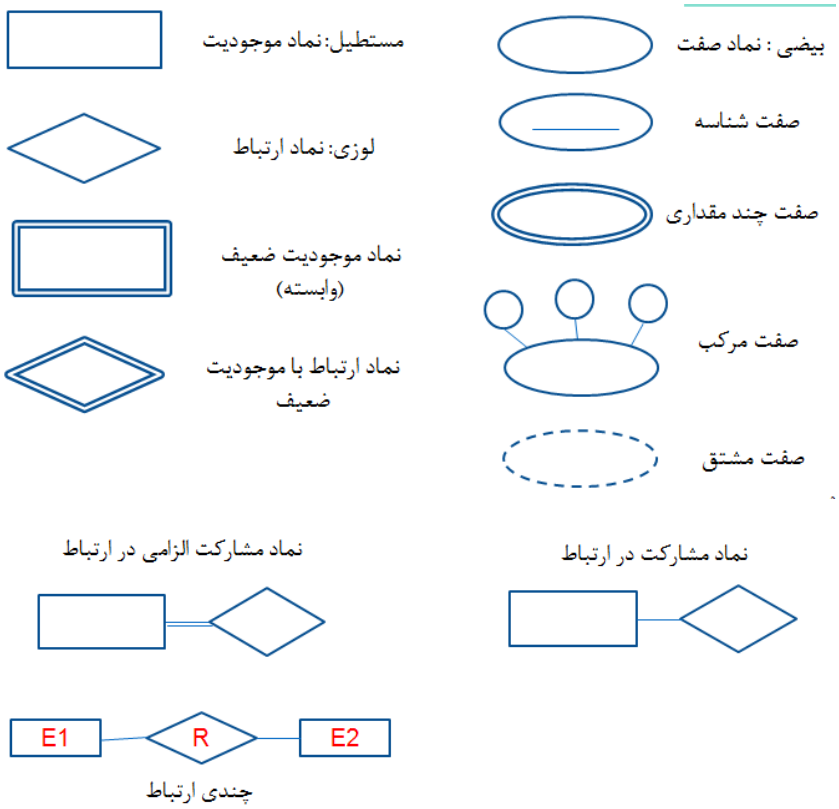
تعداد موجودیت‌های شرکت کننده در ارتباط	درجه	
	لاتین	فارسی
۱	Unary	یگانی
۲	Binary	دوگانی
۳	Ternary	سه گانی
...	...	...
N	n-ary	چند گانی

۲-۶-۳- 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 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 شکل زیر در نمودار این دو موضوع به شکل زیر در نمودار ER نشان داده می شوند.

مشارکت الزامی

۲-۷- نمادهای نمودار ER



۲-۸- مثال اول ترسیم نمودار ER

میخواهیم در یک کتابخانه ، کتبی در پایگاه داده ثبت شود، معمولا اطلاعاتی مثل شماره کتاب، سال نشر، عنوان، نویسنده یا نویسندگان، قیمت، موضوع نگه داری می شود.

اطلاعات دانشجویان هم در سیستم ذخیره می کنیم که این اطلاعات شامل ، شماره دانشجویی، نام، آدرس خواهد بود.

دانشجویان میتوانند کتاب هایی را به کتابخانه اهدا کنند.

اطلاعات متصدی کتابخانه نیز دخیره می گردد، معمولا اطلاعات اعضای خانواده متصدی نیز در کنار اطلاعاتی مثل کد کارمندی، تلفن، تاریخ استخدام نگه داری می شود.

هر دانشجویی می تواند یک یا چند کتاب را به امانت بگیرد، تاریخ تحویل و تاریخ برگشت کتاب باید نگه داری شود.

عملیات ثبت نام دانشجو توسط متصدی انجام خواهد شد و هر متصدی موظف به ثبت چند دانشجو خواهد بود.

پاسخ:

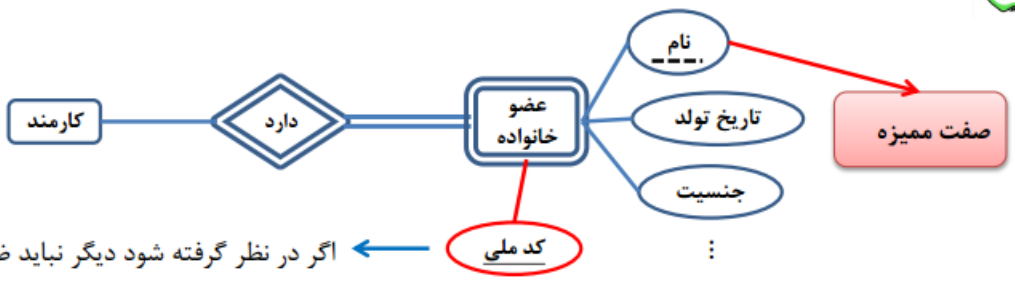
۲-۹- موجودیت ضعیف

نوع موجودیت F را ضعیف نوع موجودیت E گوییم هرگاه F با E «وابستگی وجودی» داشته باشد. (یعنی اگر E در مدلسازی مطرح نشود، F هم مطرح نباشد). علاوه‌براین نوع موجودیت ضعیف از خود شناسه ندارد.

- **طرز نمایش:**

- **تاکید:** قوی و ضعیف بودن نسبی است.
- نوع ضعیف از خود شناسه ندارد. بلکه از خود حداقل یک **صفت ممیزه-جداساز** (Discriminator) دارد.
- **صفت ممیزه (کلید جزئی):**
  - صفتی که یکتایی مقدار دارد اما نه در تمام نمونه های نوع ضعیف بلکه در بین مجموعه تمام نوع ضعیف‌های وابسته به یک نمونه از نوع موجودیت قوی (به صورت نسبی یکتاست).
  - در عمل اگر یک نوع موجودیت وابستگی وجودی به نوع موجودیت دیگر داشته باشد و از خود شناسه داشته باشد دیگر ضعیف دیده نمی‌شود.

مثال عضو خانواده به عنوان یک موجودیت ضعیف



: اگر در نظر گرفته شود دیگر نباید ضعیف دیده شود.

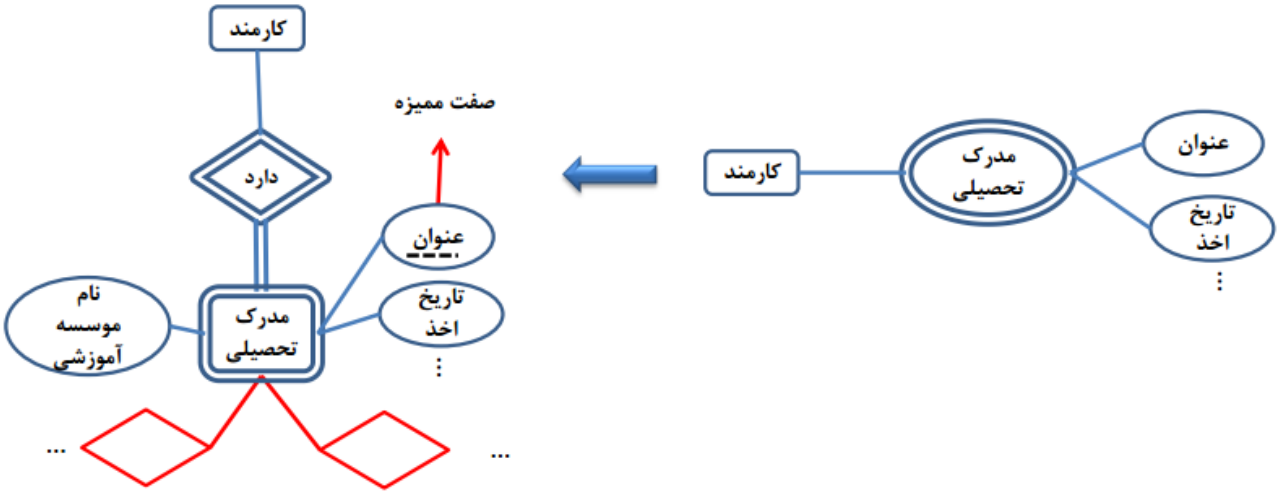
شماره کارمند	نام
۱۰۰	گلی سلی قلی
۲۰۰	ناجی تاجی سلی

- به ارتباط قوی-ضعیف، **ارتباط شناسا** (Identifying Relation) گویند.

- مشارکت نوع ضعیف در ارتباط شناسا الزامی است.

- چندی ارتباط معمولا 1:N (در حالت خاص 1:1

- صفت چند مقداری (به خصوص مرکب) را همیشه می‌توان با مفهوم نوع موجودیت ضعیف مدل کرد (نمایش داد) اما عکس این تکنیک توصیه نمی‌شود.
- **دلیل:** انعطاف پذیری مدل را از نظر گسترش پذیری کاهش می‌دهد، زیرا نوع ضعیف می‌تواند خود نوع ارتباط‌هایی داشته باشد با دیگر نوع موجودیت‌ها، اما وجود ارتباط با صفت معنا ندارد.



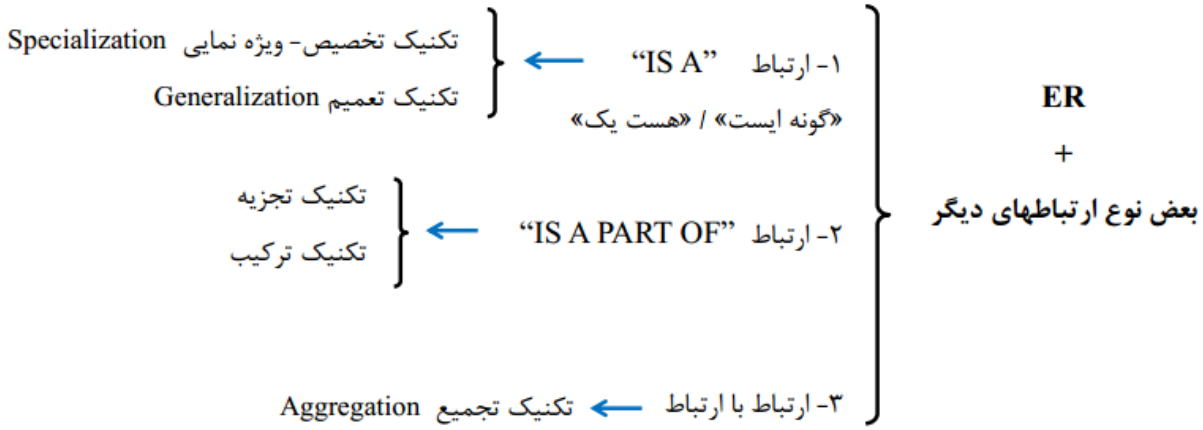
۲-۱۰- نکات راهنمای تدوین نمودار ER

- مشکل تصمیم‌گیری در مورد اینکه یک مفهوم، نوع موجودیت در نظر گرفته شود یا صفت یا نوع ارتباط باید در یک فرآیند تدریجی در مدلسازی معنایی داده‌ها اصلاح شود.
  - اگر یک مفهوم، صفت به نظر آید، آنرا صفت می‌گیریم، اما اگر به نوع موجودیت دیگری ارجاع داشته باشد، آنرا به یک نوع ارتباط در نظر می‌گیریم.
  - اگر یک (چند) صفت در چند نوع موجودیت، مشترک باشند، آنرا به عنوان یک نوع موجودیت مستقل منظور می‌کنیم.
  - اگر یک نوع موجودیت، تنها یک صفت داشته باشد و تنها با یک نوع موجودیت دیگر مرتبط باشد، آن را صفت در نظر می‌گیریم.
  - اگر مجموعه‌ای از صفات مستقلا قابل شناسایی نباشند، آنرا به صورت نوع موجودیت ضعیف در نظر می‌گیریم.

۲-۱۱- ERگسترش یافته

❑ **EER**: Extended ER یا Enhanced ER

❑ ER مبنایی کمداشت هایی داشت در نمایش بعض نوع ارتباطها (که بعدا در حیطه شی‌گرایی مطرح شد)

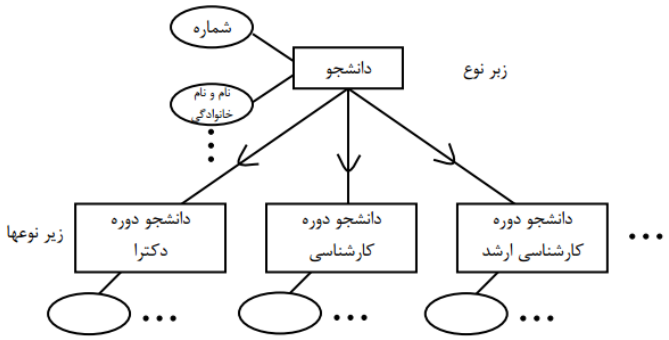


۲-۱۲- تخصیص

۲-۱۳- تعمیم

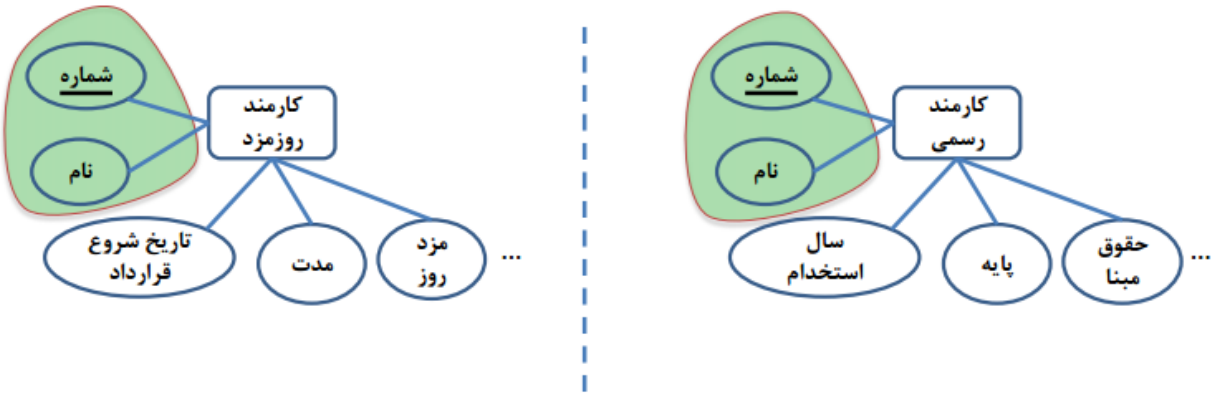
- تخصیص عبارتست از بازشناسی گونه های خاص یک شی براساس یک ضابطه ی مشخص. می گوییم بین هر زیر نوع موجودیت و زبر نوع آن ارتباط "is-a" وجود دارد. به بیان دیگر هر زیر نوع گونـه ی خاصـی اسـت از حداقل یک زبر نوع.
- زبر نوع مجموعه ای از صفات دارد که بین تمام زیر نوع های آن مشترک است و بنابر این ، هر زیر نوع، صفات زبر نوع خود را به ارث می برد. به علاوه هر زیر نوع ، صفات خاص خود را دارد.

تعمیم ، عکس عمل تخصیص است ، به این معنا که با داشتن زیـر نوعهـای خـاص ، صـفات مشـترک بـین آنهـا (ازجمله شناسه) را در یک مجموعه صفات برای یک زبر نوع موجودیت در نظر می گیریم، طبعاً این مجموعـه صـفات مشترک جدا شده را ، دیگر برای هر زیر نوع تکرار نمی کنیم. از این خاصیت زمانی استفاده خواهیم کرد که در سیسـتم چنـدین موجودیـت داشـته باشـیم کـه دارای صـفات مشترکی باشند در این حالت یک زبر نوع ایجاد کرده و صـفات مشـترک را بـرای آن در نظـر مـی گیـریم و بقیـه ی موجودیت ها را به عنوان زیر نوع آن در نظر می گیریم. در این زیر نوع ها دیگر صفات مشترک لحاظ نخواهد شد.



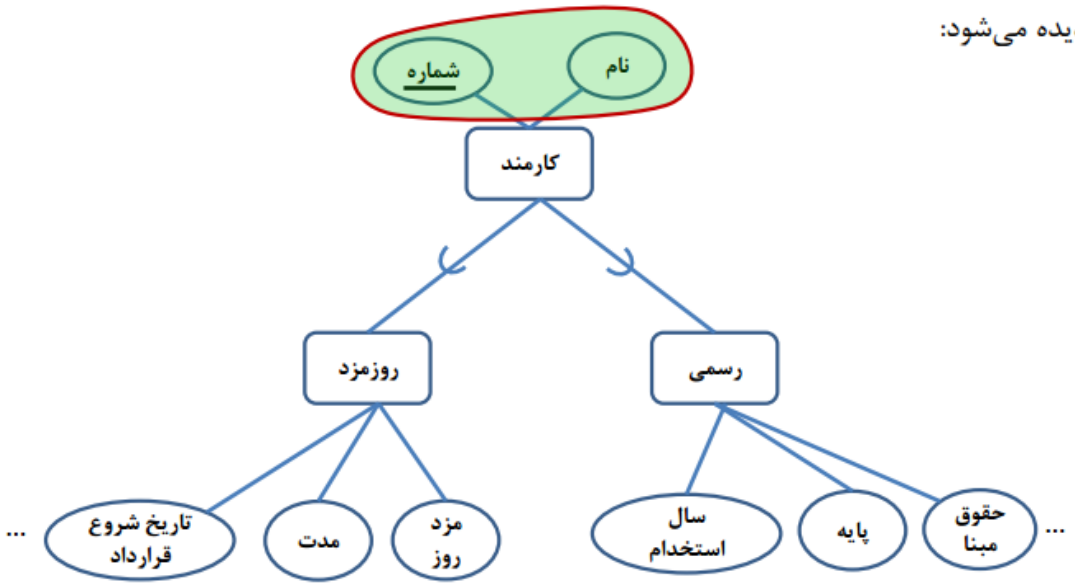
شکل ۱۲-۲: مثال تخصیص

❑ تعمیم عبارت است از تشخیص یک نوع موجودیت جدید از روی [با داشتن] $n \geq 2$ نوع موجودیت از پیش دیده که ماهیتا از یک نوع باشند. (احیانا به منظور ادغام ERD های جدا)



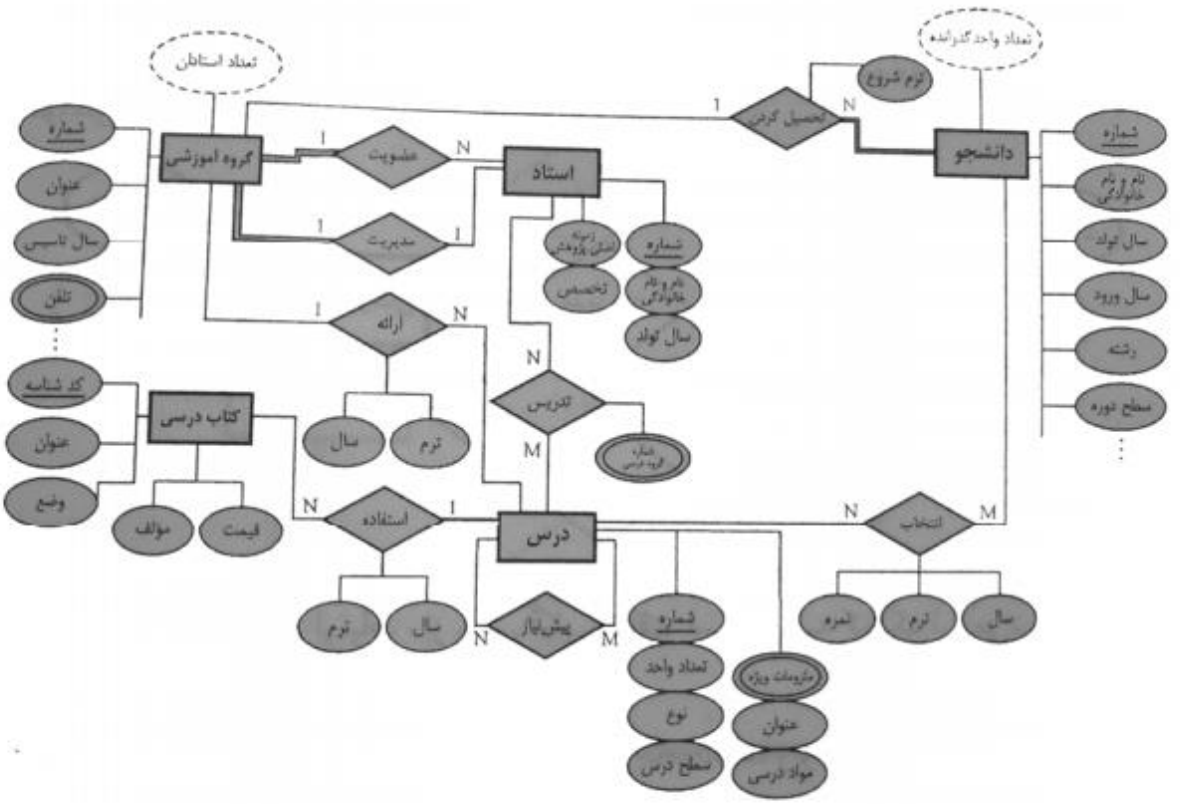
- یک نوع موجودیت (کارمند) در سطح انتزاعی

بالاتر دیده می‌شود:



- **شرایط تعمیم:**
  - داشتن شناسه مشترک [یعنی از یک دامنه]
  - حداقل وجود دو نوع زیرنوع
  - هرچه صفات مشترک بیشتر، تعمیم توجیه پذیرتر است [شرطِ لازم نیست ولی شرطِ ارجحیت است].

۲-۱۴- نمودار ER آموزش دانشگاه



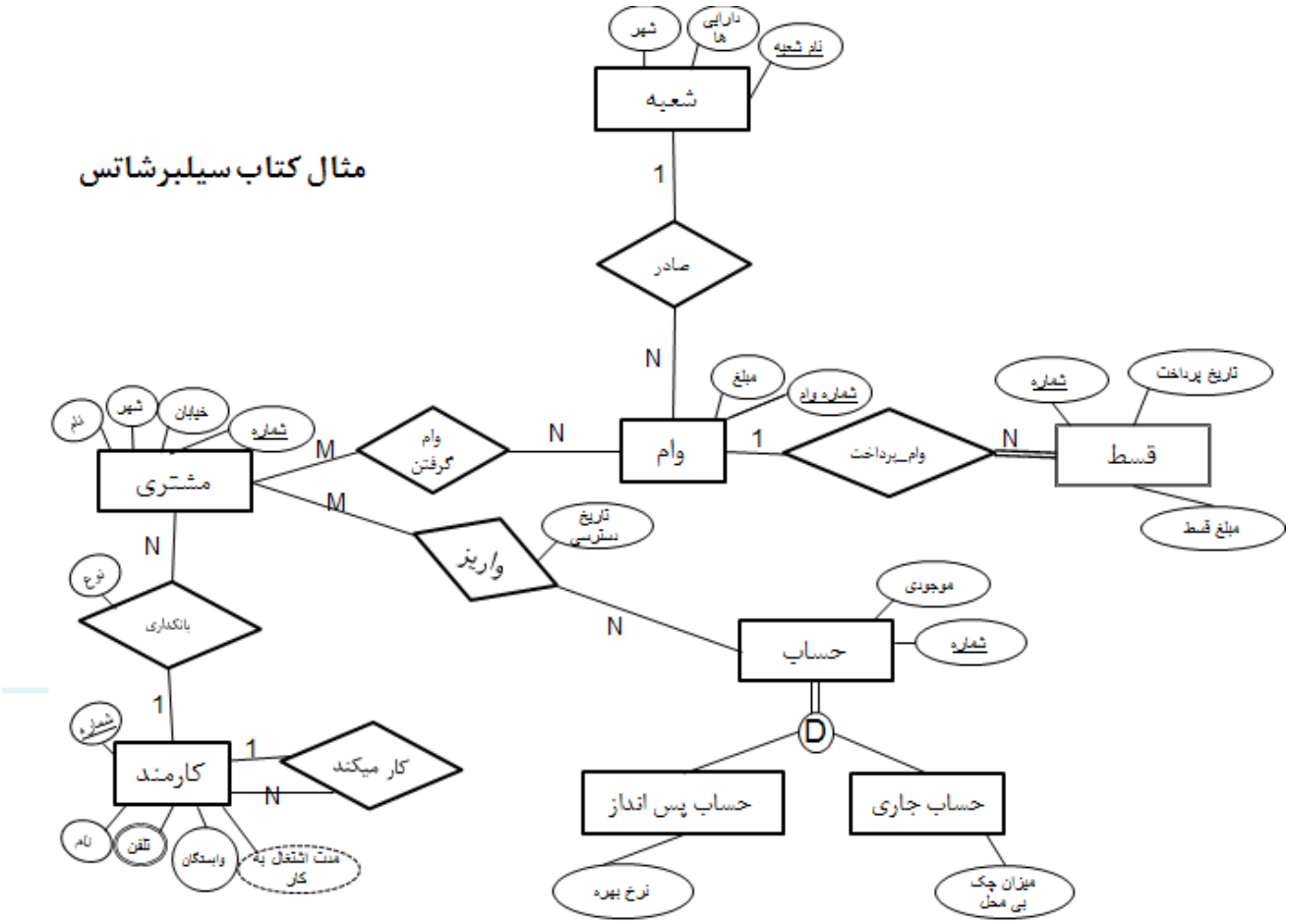
شکل ۸-۲: بخشی از نمودار ER (بسیار ساده شده) پایگاه داده آموزش دانشکده

۲-۱۵- مثال ۲ مدلسازی معنایی

می خواهیم 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 امده 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.



۲-۱۶- مثال ۳

۴- [سوال ۱۸ فصل ۳] 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 می‌شوند. شعب شهرستان هر کدام مدیر مستقلی دارند که تحت نظر مدیر شعبه‌ی اصلی فعالیت می‌کنند.
- رستوران با تامین‌کنندگان مواد غذایی مختلفی قرارداد دارد که محموله‌های 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 می‌گیرند.
- غذاهای رستوران که نوع و میزان مواد غذایی هر کدام مشخص است، با قیمت‌های مشترک توسط مشتریان خریداری می‌شود.
- 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.

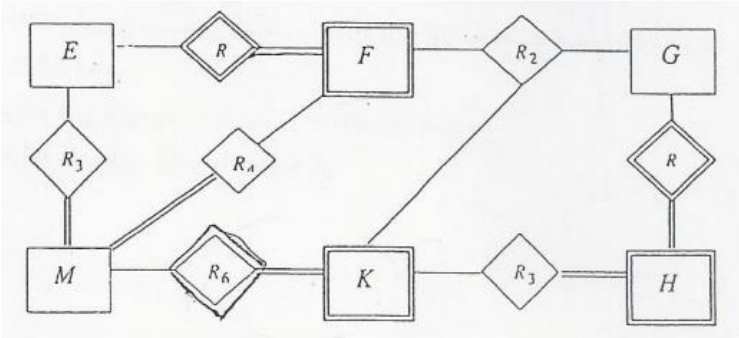
یک مدل‌سازی معنایی داده (با استفاده از مدل [E]ER) از محیط فوق ارایه دهید. اگر احتیاج به فرض دیگری دارید آن را ذکر کنید.

۲-۱۷- مثال ۴

یک شرکت بیمه ی خودرو قصد دارد اطلاعات مربوط به بیمه شدگان و نوع خودرو آنها را نگهداری کند. بیمه شده می تواند حقیقی یا حقوقی باشد. خودرو نیز انواع مختلفی مانند سواری، باری و موتور دارد. یکبار با استفاده از مدل ER و یکبار با استفاده از مدل EERداده های مورد نیاز را مدل سازی کنید.

۲-۱۸- مثال ۵

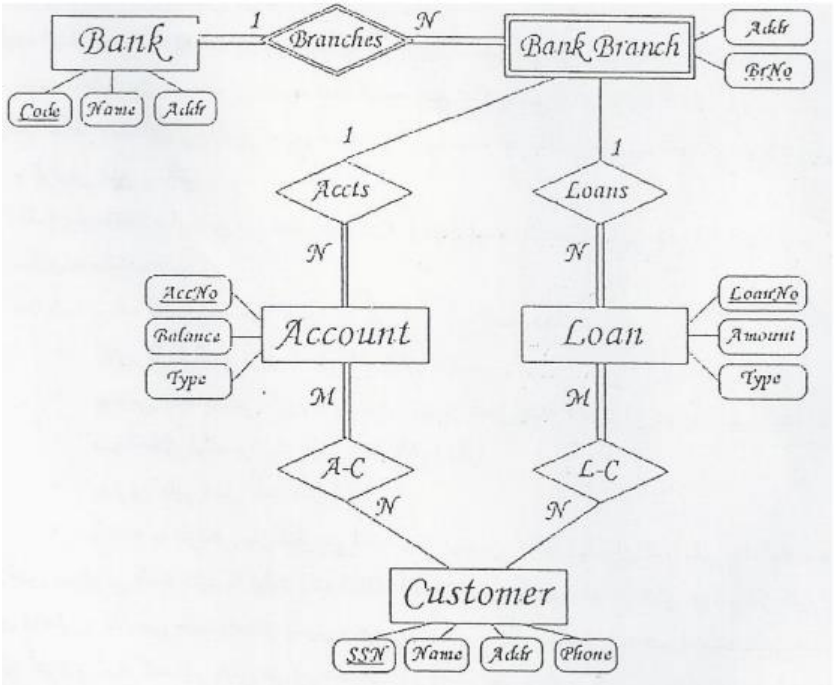
در نمودار زیر چند نوع موجودیت ضعیف و چند نوع موجودیت قوی وجود دارد؟



۲-۱۹- مثال 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.

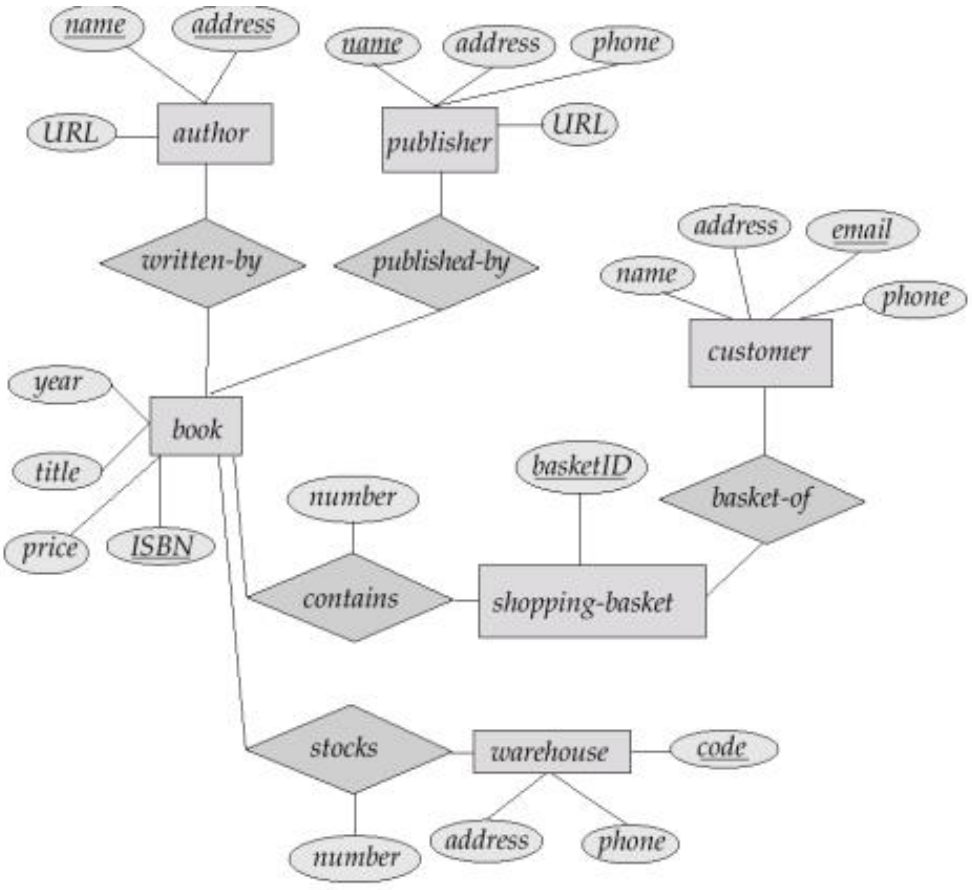


۲-۲۰- مثال ۷

تمرین: شکل ۲-۱۹را در نظر بگیرید که یک کتابفروشی onlineرا مدل می کند.

۱. در نظر بگیرید که کتابفروشی می خواهد کاست موسیقی و دیسک فشرده را به مجموعه خود اضافه کند. موسیقی های یکسان ممکن است در شکل کاست ها یا دیسکهای فشرده، با قیمت های متفاوت ارائـه شـوند. نمودار ERرا برای مدل کردن این مـوارد اضـافی تغییـر دهیـد، از اثـرات آن روی shopping-basket صرفنظر شود.

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



شکل ۱۹-۲: مثالی از کتاب silberschots online bookstore

۲-۲۱- مثال 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 می تواند در هر بار خرید از برخی اجناس تعدادی را خریداری کند. به ازای هر خرید یک فاکتور صادر می شود 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 نگه داری می شود. مشتری می تواند حقیقی یا حقوقی باشد. برای مشتری حقیقی شماره، نام و نام خانوادگی و برای مشتریان حقوقی، شماره مشتری، نام و نوع شغل ثبت می شود. شماره مشتری برای مشتریان حقیقی و حقوقی در همه شعب یکتاست.

فصل سوم : مدل رابطه ای

۳-۱- تاریخچه رابطه

پس از پایگاه داده های سلسله مراتبی و شبکه ای، که هر یک دارای ضعف هایی بودند، متخصصان در جستجوی مدلی بودندکه دارای ساختار داده ای با انتزاع قوی باشد. مدل رابطه ای در سال ۱۹۷۰ توسط E.F.Codd ابداع گردید.

این مدل دارای ساختار داده ای با انتزاع قوی بوده و اساسا ساختار داده ای در آن بر اساس یک مفهوم ریاضی به نام رابطه (Relation) استوار است.

- اساسا پایگاه داده ی رابطه ای مجموعه ای است از تعدادی جدول.
  - مفاهیم ساختاری جدول عبارتند از:
    - جدول
    - سطر
    - ستون

۳-۱-۱- تعریف رابطه

- با فرض وجود n میدان D1,D2,D3,…Dn [نه لزوما متمایز] رابطه R از دو قسمت تشکیل می شود:
- ۱) سرآیند: مجموعه ای از n عنصر به صورت $<A_i : D_i>$ که در آن A_i نام صفت است. (حتما متمایز) و D_i نام میدان صفت است (لزوما متمایز نیست) ، مثال:
- $\{ A_1:D_1\},\{A_2:D_2\},\{....\}$
- نحوه نمایش سرآینده معمولا به این صورت است.
- $R(H)=H_R=H=\{A_1,A_2,A_3,\ldots A_n\}$
- ۲) پیکره یا بدنه: مجموعه ای از m تاپل T به نحوی که هر T خود مجموعه ای n تایی به صورت $<A_i : V_i>$ دارد. V_i مقدار آن صفت است.
- مثال : {<علی: نام>، <۱۲۰۱: شماره دانشجویی>}

۳-۲- تعریف دامنه

- مجموعه تمام مقادیر یک صفت را دامنه گوییم. مثلا دامنه اعداد صحیح یعنی مجموعه {…,-1,0,1,2,…} و صفتی که دامنه اش اعداد صحیح است می تواند هر یک از این مقادیر را تا حد گنجایش کامپیوتر مربوطه در خود جای دهد.

۳-۳- تعریف رابطه

زیر مجموعه ای از ضرب دکارتی چند دامنه است. مثلا اگر داشته باشیم D1:Integer و D2:String آن گاه هر مجموعه ای که عضوهایش زوج های مرتب (D1,D2) باشند یک رابطه است. ساده ترین راه نمایش و پیاده سازی رابطه، جدول دو بعدی است. اسلاید صفحه بعد نمونه ای از یک رابطه را نشان می دهد.

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

۳-۴- زوج مرتب (تاپل)

ارتباط مجموعه ای از مقادیر در یک رابطه است. هر سطر یک جدول معادل یک تاپل است.

هر یک از ستون های یک جدول نیز معرف یک صفت است. در مدل رابطه ای صفت ها از دامنه های ساده (تجزیه ناپذیر) تعریف می شوند و دامنه های تودرتو مجاز نیستند.

۳-۵- ویژگی های رابطه

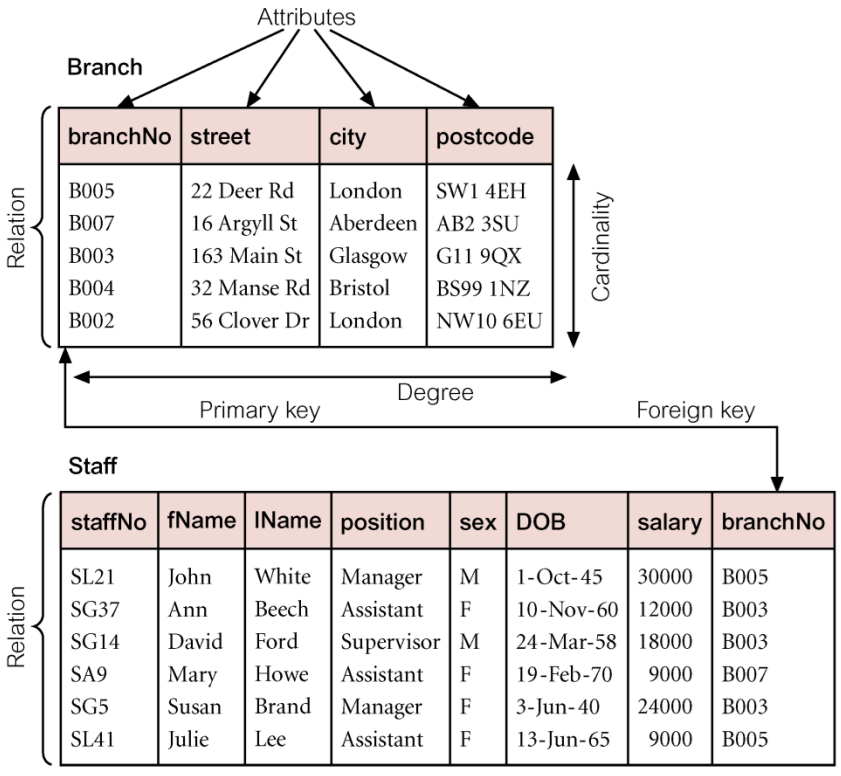
- هر تاپل باید منحصر به فرد باشد.
- ترتیب صفات خاصه اهمیتی ندارد.
- هر سلول رابطه شامل فقط یک مقدار است.
- ترتیب تاپل ها اهمیتی ندارد.

۳-۶- تناظر بین اجزا دو مفهوم رابطه و جدول

اجزاء مفهوم رابطه	اجزاء مفهوم جدولی
رابطه	جدول
تاپل	سطر
صفت	ستون
میدان	مجموعه مقادیر ستون
درجه	تعداد ستون‌ها
کاردینالیتی	تعداد سطرها

۳-۷- مثالی از نمایش رابطه های کارمند و شعبه

- رابطه:
- کاردینالیتی:
- درجه:
- صفات رابطه ها:
- کلید خارجی:



۳-۸- کلید در مدل رابطه ای

- کلید انواع مختلفی دارد که عبارتند از:
- ابر کلید (Super Key)
- کلید کاندید (Candidate Key)
- کلید اصلی (Primary Key)
- کلید فرعی (Alternative Key)
- کلید خارجی (Foreign Key)

۳-۸-۱- ابر کلید

تعریف: هر صفت خاصه یا ترکیبی از صفات در رابطه که یکتایی مقدار (تکرار نشدن) در گستره رابطه داشته باشد. ابر کلید ممکن است حاوی صفات اضافی باشد که برای شناسایی یکتایی صفت ضروری نباشند. به همین دلیل ابر کلید در عمل کاربردی ندارد و در برخی از گونه های نرمال کاربرد دارد.

۳-۸-۲- کلید کاندید

- تعریف: هر صفت خاصه یا ترکیبی از صفات در یک رابطه که دو خاصیت زیر را داشته باشد کلید کاندید رابطه است:
- ۱- یکتایی مقدار
- ۲- کاهش ناپذیری یا کمینگی
- *(توجه کنید که تفاوت ابرکلید یا کلیدکاندید در کاهش ناپذیر بودن کلید کاندید است در حالی که ابر کلید می تواند کاهش پذیر یا کاهش ناپذیر باشد)
- توجه :کلید کاهش ناپذیر، کلیدی است که اگر هر جزء آنراحذف کنیم دیگر کلید نباشد.
- نکته : هر رابطه می تواند چند کلید کاندید داشته باشد.
- وقتی یک کلید، بیش از یک صفت داشته باشد، به ان **کلید مرکب** می گوییم.
- رابطه ای که در آن کلیه صفات با هم کلید کاندید شوند، را **رابطه تمام کلید** گوییم.

۳-۸-۳- کلید اصلی

- تعریف: یکی از کلیدهای کاندید که برای شناسایی تاپل ها به طور یکتا درون رابطه توسط طراح پایگاه داده انتخاب شود را کلید اصلی گوییم.
- 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 کد ملی و هم کد کارمندی کلید کاندید خواهد بود ولی به دلیل کوتاه تر بودن کد کارمندی و نیز شناسایی افراد در اداره با کد کارمندی مرسوم تر است، بنابراین کد کارمندی به عنوان کلید اصلی انتخاب می شود.

۳-۸-٤- کلید بدیل یا فرعی

- کلیدهای کاندیدی که به عنوان کلید اصلی انتخاب نمی شوند، کلید بدیل یا فرعی نام دارند. کلید فرعی در عمل پیاده سازی نشده است.
- کلید خارجی
- تعریف:دو رابطه R1و R2 را در نظر میگیریم. صفتی (یا ترکیبی از صفات) در R1که در این رابطه کلید نیست اما در R2 کلید اصلی است، یک کلید خارجی برای رابطه R1نامیده می شود.
- کلید خارجی برای برقراری ارتباط بین موجودیتها (جداول) به‌کار می‌رود.
- کلید خارجی را با نقطه چین زیر صفت یک رابطه نشان می دهیم.
- توجه: از بین تمام کلید های نامبرده شده، کلید خارجی تنها کلیدی است که می تواند هیچ مقدار باشد.

۳-۸-۵- مثال کلید

۳-۹- تعریف هایی از جامعیت (integrity)

تعریف : صحت، دقت و سازگاری و اعتبار داده های ذخیره شده در پایگاه در تمام لحظات. در واقع جامعیت مجموعه ای است از امکانات برای انجام کنترل داده های ذخیره شده درپایگاه داده .

مثال:یکسری کنترل ها وجود دارد که دانشجوی مشروط شده نمی تواند بیشتر از ۱۴ واحد اخذ کند.

در برخی مراجع دو مفهوم جامعیت و سازگاری(Consistency) مترادف یکدیگر بیان شده اند.

۳-۹-۱- قوانین جامعیت

قوانین جامعیت در مدل رابطه ای:

۱. **جامعیت دامنه ای**: یعنی تمام صفات درتمامی رابطه ها ازنوع دامنه خود باشند (مثال ۷۴/۱به عنوان شماره دانشجویی که عدد صحیح است پذیرفته نشود).

۲. **جامعیت درون رابطه ای**:یعنی هر رابطه به تنهایی صحیح باشد. مثلا عضو تکراری نداشته باشد وکلیدهایش درست باشند، کلیدها دارای مقدار تهی یا تکراری نباشد.

۳. **جامعیت ارجاعی**: یعنی کلید خارجی درست تعریف شده باشد مثلا کلید خارجی دریک رابطه حتما دررابطه دیگر کلید باشد و مقداری که به کلید خارجی داده می شود در جدول دیگر وجود داشته باشد.

۳-۱۰- طراحی منطقی پایگاه داده

به فرایند ساخت یک مدل از اطلاعات محیط عملیاتی بر اساس یک مدل داده ای 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 می نگارد.این مدل تحت تاثیر مدل داده ای مورد استفاده در پایگاه داده است (مثل مدل داده ای رابطه ای)

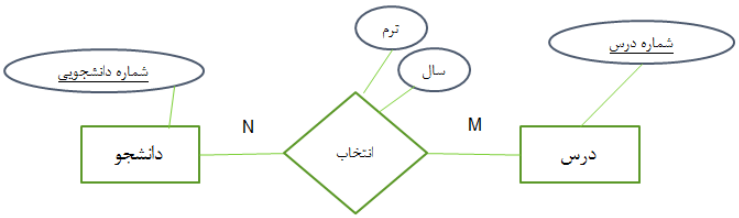
در ادامه حالت های مختلف تبدیل نمودار ER به رابطه شرح داده می شود.

۳-۱۰-۱- حالت اول: نگاشت ارتباط های M:N

شرایط این حالت عبارتند از:

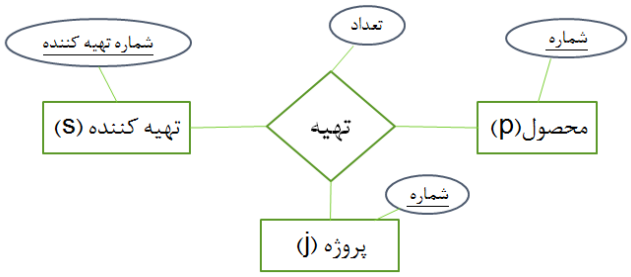
- تعداد موجودیت ها n>=2
- نوع موجودیت ها = مستقل
- ارتباط از نوع M:N
  - روش طراحی به این صورت است که برای هر یک از موجودیت ها، یک جدول در نظر بگیریم و یک جدول هم ارتباط بین موجودیت ها را نشان دهد. در واقع با وجود n موجودیت، در این حالت حداقل n+1 جدول طراحی می شود.
  - کلید کاندید؟
- کلید کاندید جدول مربوط به ارتباط موجودیت ها، کلید کاندید موجودیت های مستقل می شود.
  - در این حالت طراحی، مشارکت الزامی یا غیر الزامی موجودیت ها هیچ تاثیری در طرز طراحی ندارد.

۳-۱۰-۲- مثال حالت اول



- Student(Stid,sname,tel,address,….)
- Course(Coid,cname,unit,….)
- StC(Stid, Coid,term,year,….)

۳-۱۰-۳- مثال دیگری از حالت اول



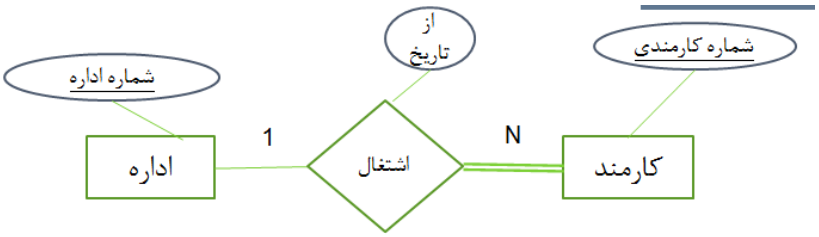
۳-۱۰-٤- حالت دوم: نگاشت ارتباط های 1:N

شرایط این حالت عبارتند از:

- تعداد موجودیت ها n=2
- نوع موجودیت ها مستقل
- ارتباط 1:N

- برای طراحی دو ارتباط کافیست. یک جدول برای موجودیت طرف یک و جدول دوم برای موجودیت طرف N و خود ارتباط. 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 نمی گیرد.
- در حالت کلی می توان گفت برای n=1 و یا n=2 به n جدول نیاز داریم.

۳-۱۰-۵- مثال حالت دوم



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

توجه : مشارکت الزامی در شکل بالا نشان می دهد که هیچ کارمندی نیست که در اداره ای اشتغال نداشته باشد. این مسئله باید در هنگـام تعریـف در DBMS با استفاده از قید not null لحاظ شود. به عبارتی محدودیت الزامی بودن به طور مستقیم در رابطه ها و جدول ها دیده نمـی شـود و در شِما آورده می شود.

۳-۱۰-٦- حالت سوم: نگاشت ارتباط های 1:1

شرایط این حالت عبارتند از:

- تعداد موجودیت ها n=2
- نوع موجودیت ها مستقل
- ارتباط 1:1

برای طراحی بسته به وضعیت مشارکت، عمل می کنیم:

۱. اگر مشارکت هر دو موجودیت غیر الزامی باشد، نیاز به سه جدول داریم و مشابه حالت اول عمل می کنیم.

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 جز کلید کاندیدی آن نمی شود.

۳. ۱. اگر مشارکت هر دو موجودیت الزامی باشد، فقط یک جدول می تواند کافی باشد.کلید کاندید این جدول، کلید کاندید هر یک از موجودیت ها می تواند باشد.

۳-۱۰-۷- مثال حالت سوم

- Department(Deptid,name,….,Tid)
- Teacher(Tid,name,tel,address,….)

- اگر مشارکت هر دو طرف الزامی باشد آن گاه یک جدول با دو کلید کاندید به شکل زیر خواهیم داشت:
- Dept_Teach(Deptid,…, Tid,…)
- اگر مشارکت دو طرف غیر الزامی باشد آن گاه سه جدول به شکل زیر خواهیم داشت:

- Teacher(Tid,name,tel,address,….)
- Department(Deptid,name,….,Tid)
- Dept_Teach(Deptid, Tid)

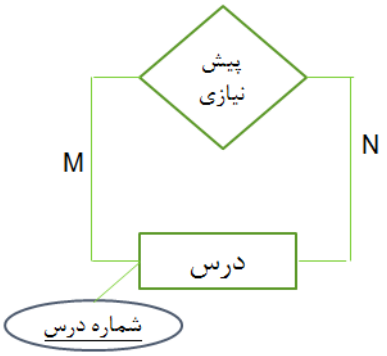
۳-۱۰-۸- حالت چهارم

شرایط این حالت عبارتند از:

- تعداد موجودیت ها n=1
- نوع موجودیت ها مستقل
- ارتباط M:N

برای طراحی نیاز به دو جدول داریم: یکی برای موجودیت و دیگری برای ارتباط. کلید کاندید موجودیت مستقل، دوبار در جدول ارتباط به عنوان کلید خارجی ظاهر می شود و ترکیب هر دو کلید کاندید جدول ارتباط می شوند. نام یکی از دو صفت کلید خارجی را هم باید تغییر دهیم. اگر ارتباط موجودیت با خودش، صفتی هم داشته باشد، در جدول ارتباط قرار می گیرد.

۳-۱۰-۹- مثال حالت چهارم



- Course(coid,…)
- PreCo(coid,copreid)

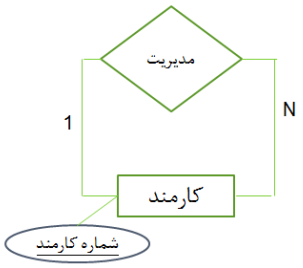
۳-۱۰-۱۰- حالت پنجم

شرایط این حالت عبارتند از:

- تعداد موجودیت ها n=1
- نوع موجودیت ها مستقل
- ارتباط 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 غیر الزامی و درصد مشارکت هم ضعیف باشد، می توان دو جدول مثل حالت چهار طراحی کرد. این طرز حالت پنجم، باعث ایجاد حلقه ارجاع می شود و رابطه به خودش رجوع می کند.

۳-۱۰-۱۱- مثال حالت پنجم



- 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 داده های اداره پر کنیم، سطر مربوط به کارمند مدیر، در بخش emid مقدار null می گیرد.

۳-۱۰-۱۲- حالت ششم

شرایط این حالت عبارتند از:

- تعداد موجودیت ها n=1
- نوع موجودیت ها مستقل
- ارتباط 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 می شود و به دو طرح می توان این جدول را طراحی نمود، در مثال صفحه بعد این دو طرح نشان داده شده است.

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 می شود.

۳-۱۰-۱۳- مثال حالت ششم

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

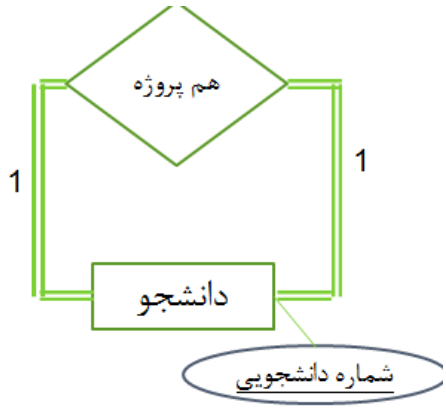
- student(stid,…,CoprojStid)

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 می توان طراحی را انجام داد:

- student(stid,…,CoprojStid, ….)

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

- student(stid, name, address , …)
- StuCoproj(stid, CoprojStid)



۳-۱۰-۱٤- حالت هفتم

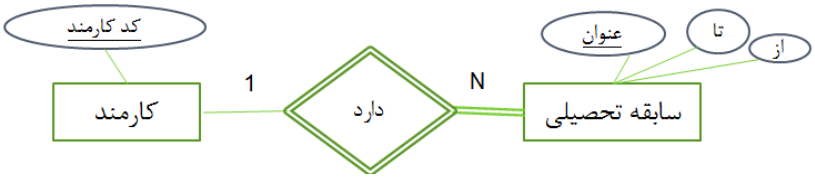
- نوع ارتباط با موجودیت ضعیف

برای طراحی یک جدول برای موجودیت قوی در نظر گرفته می شود 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 نمونه هایی که با موجودیت قوی ارتباط دارند، یکتاست.

نکته: نوع موجودیت ضعیف را گاهی با یک صفت مرکب یا چند مقداری هم نشان می دهند. 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 ارتباط های دیگر شرکت داشته باشد، نمایش ان با صفت مرکب یا چند مقداری صحیح نیست.

۳-۱۰-۱۵- مثال حالت هفتم



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

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

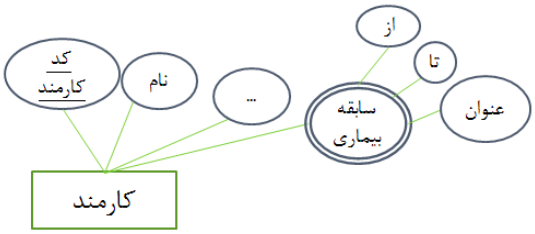
۳-۱۰-۱۶- حالت هشتم

- نوع موجودیتی که صفت چند مقداری داشته باشد.

برای طراحی یک رابطه برای موجودیت قوی در نظر گرفته می شود و یک رابطه صفت چند مقداری. کلید کاندید موجودیت به عنوان کلید خارجی به صفت چند مقداری در رابطه اش، 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 چند مقداری ارجحیت دارد ولی روش عمومی طراحی آن ها یکی است.

۳-۱۰-۱۷- مثال حالت هشتم



واحد درمان یک اداره سابقه بیماری های کارمندان را نگه داری می کند. هر کارمند ممکن است سابقه بیماری های مختلفی داشته باشد و حتی به یک بیماری چندین بار مبتلا شده باشد.

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

۳-۱۰-۱۸- حالت نهم

- وجود ارتباط IS-A بین دو نوع موجودیت.
- چهار تکنیک دارد:

۱- فرض: نوع موجودیت E، n زیرنوع دارد.

n+1 رابطه طراحی می‌کنیم. یک رابطه برای زیرنوع و یک رابطه برای هر یک از زیرنوع‌ها.

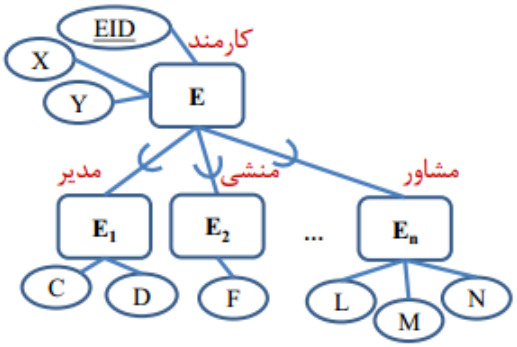
E (EID, X, Y)

E1 (EID, A, B)

E2 (EID, F)

…

En (EID, L, M, N)



حالت نهم – تکنیک ۱

- مزیت این تکنیک: شرط خاصی از نظر نوع تخصیص ندارد (تکنیک‌های دیگری که مطرح می‌شود، همگی برای شرایط خاص هستند).
- عیب این تکنیک: اگر بخواهیم در مورد یک زیرنوع، اطلاعات کامل به دست آوریم، باید JOIN کنیم.

حالت نهم- تکنیک ۲

۲- طراحی با n رابطه: برای **زیرنوع**، رابطه‌ای طراحی نمی‌کنیم. بنابراین صفات مشترک باید در رابطه نمایشگر هر **زیرنوع** وجود داشته باشد.

- شرط لازم: باید تخصیص کامل باشد. اگر نباشد، بخشی از داده‌های محیط قابل نمایش نیستند.

E1 (EID, X, Y, A, B)

E2 (EID, X, Y, F)

…

En (EID, X, Y, L, M, N)

- مزیت نسبت به تکنیک اول: برای به دست آوردن اطلاعات کامل زیرنوع‌ها نیازی به JOIN نیست.
- نکته: در این تکنیک، لزوماً افزونگی پیش نمی‌آید. اگر تخصیص هم‌پوشا باشد میزانی افزونگی پیش می‌آید.

حالت نهم- تکنیک ۳

۳- طراحی فقط با یک رابطه، با استفاده از صفت نمایشگر نوع **زیرنوع‌ها**

- شرط استفاده از این تکنیک: تخصیص مجزا باشد؛ یعنی یک نمونه کارمند، جزء نمونه‌های حداکثر یک **زیرنوع** باشد.

E (EID,	X,	Y,	A,	B,	F,	L,	M,	N,	TYPE)
100	x1	y1	a1	b1	?	?	?	?	مدیر
200	x2	y2	?	?	?	l2	m2	n2	مشاور

- مزیت این تکنیک: برای به دست آوردن اطلاعات کامل زیرنوع‌ها نیازی به JOIN نیست.
- عیب این تکنیک: هیچمقدار (Null) زیاد دارد و درجه رابطه زیاد است.

حالت نهم- تکنیک ۴

۴- طراحی فقط با یک رابطه، با استفاده از آرایه بیتی؛ هر بیت نمایشگر نوع یک **زیرنوع**. در واقع برای نمایش هر نمونه موجودیت، بسته به اینکه در مجموعه نمونه‌های کدام زیرنوع باشد، بیت مربوطه‌اش را ۱ می‌کنیم.

❑ شرط استفاده از این تکنیک: وقتی تخصیص هم‌پوشا باشد (سایر شرایط همانها که در تکنیک ۳ گفته شد).

آرایه بیتی

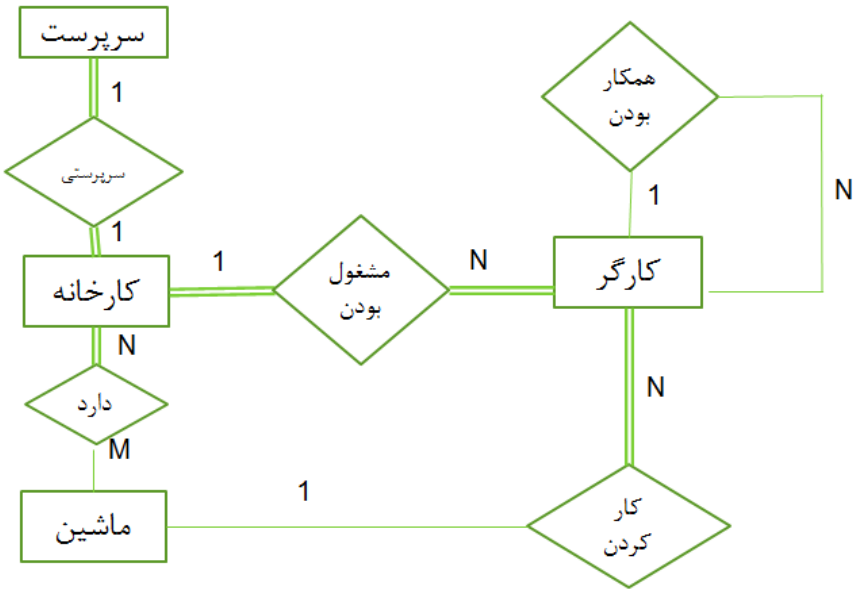
E (EID, X, Y, A, B, F, L, M, N, TB1, TB2, …,TBn)

مدیر منشی مشاور

100	x1	y1	1	0	0
200	x2	y2	0	1	0

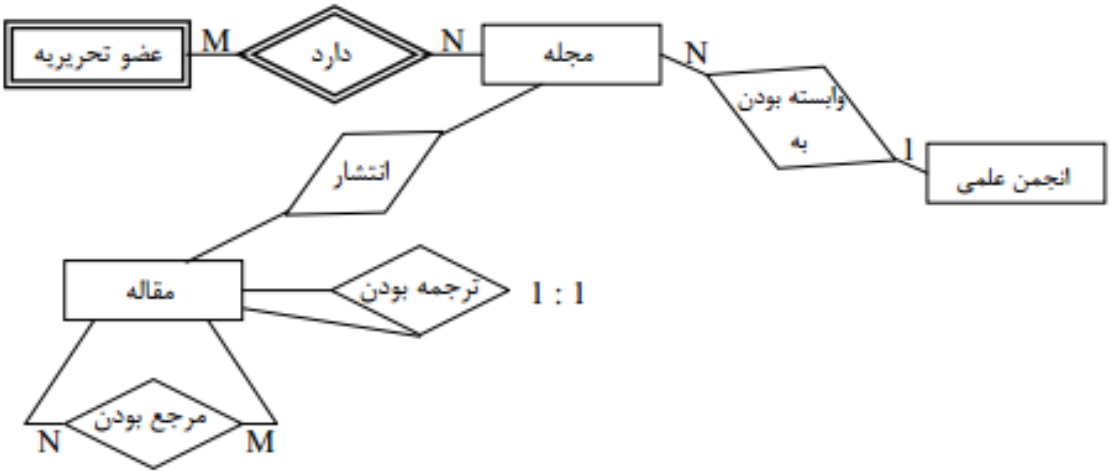
۳-۱۱- مثال ۱

با توجه به نمودار er داده شده، پایگاه رابطه ای مربوطه را طراحی کنید.



۳-۱۱-۱- مثال ۲

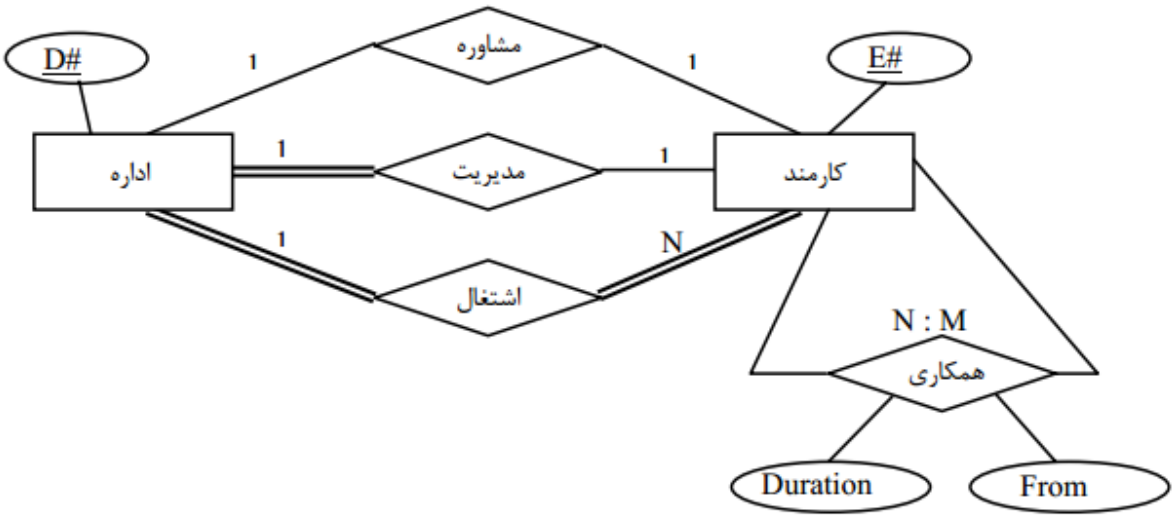
با استفاده از روش طراحی بالا به پایین نمودار EERزیر را با مدل رابطه ای تبدیل کنید.



کلید اصلی موجودیت ها عبارتند از: مجله : عنوان، انجمن علمی: عنوان، مقاله: عنوان و کد مقاله، کلید صفت ممیزه عضو تحریریه: نام

۳-۱۱-۲- مثال ۳

با توجه به نمودار er داده شده، پایگاه رابطه ای مربوطه را طراحی کنید.



۳-۱۲- یادداشت های فصل