به نام خدا

جزوه

پایگاه داده ها

تهیه و تدوین : سخی محمد محمدی
آموزشکده فنی دختران بیرجند

فصل اول

سیستم های بانک اطلاعاتی

۱-۱- پایگاه دادها

در سیستم های کامپیوتری با دو مفهوم داده و اطلاع سروکار داریم . **داده** حقایق خام وبدون معنی وتفسیر است مثل : 75/19 و **اطلاعات** معنا وتفسیر داده هاست به عنوان مثال: معدل=75/19

داده های ذخیره شده در پایگاه داده می تواند توسط کاربر یا برنامه خوانده شده و به دلخواه تفسیر شوند . در نتیجه هم مجموعه ای از داده ها هستند و هم مجموعه ای از اطلاعات پس می توان آنها را پایگاه داده ها یا بانک اطلاعاتی نامگذاری کرد.

پایگاه داده مجموعه ای از رکوردهای ذخیره شده به صورت سازماندهی شده شامل فایل ها و جداول....به طوری که عملیات زیر روی آنها قابل اجرا باشد:

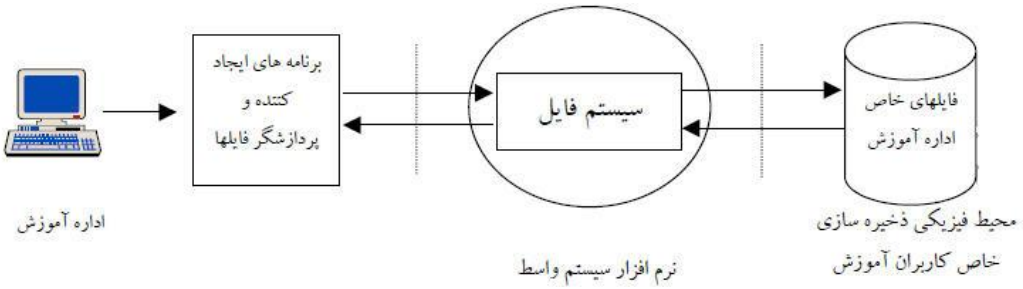
۱- ساخت،اصلاح،حذف فایل ها یا جداول

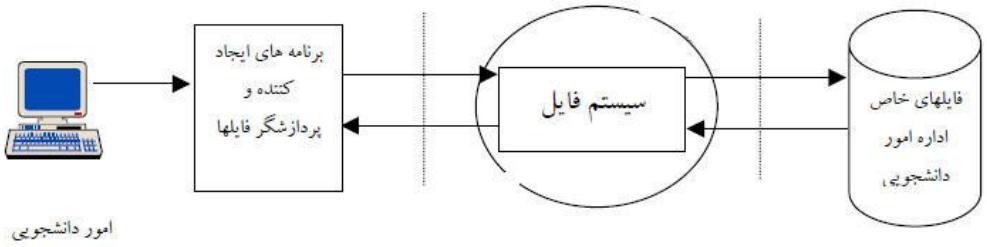
۲- اضافه و حذف،ویرایش، بازیابی رکوردها

۱-۲- دو روش برای مدیریت رکوردها ی اطلاعات

۱)روش فایلینگ (فایل پردازی):

رکوردها در داخل فایل هایی ذخیره شده و با استفاده از امکانات زبان برنامه نویسی می توان آنها را مدیریت کرد .برای مدیریت هر فایل باید برنامه جداگانه(بخشی از برنامه)وجود داشته باشد. این برنامه خودشان ساختار داده ها را تعیین کرده، ایجاد می کنند و سپس روش هاییی برای دسترسی به داده ها استفاده می کنند که وابسته به نحوه سازماندهی داده هاست. به عنوان مثال در محیط دانشگاه ممکن است دو سیستم مختلف در اداره آموزش و امور دانشجویی استفاده شود که هر کدام مسئول ایجاد ساختارهای خاص خود و مدیریت داده های آنها است.





معایب این شیوه عبارت است از:

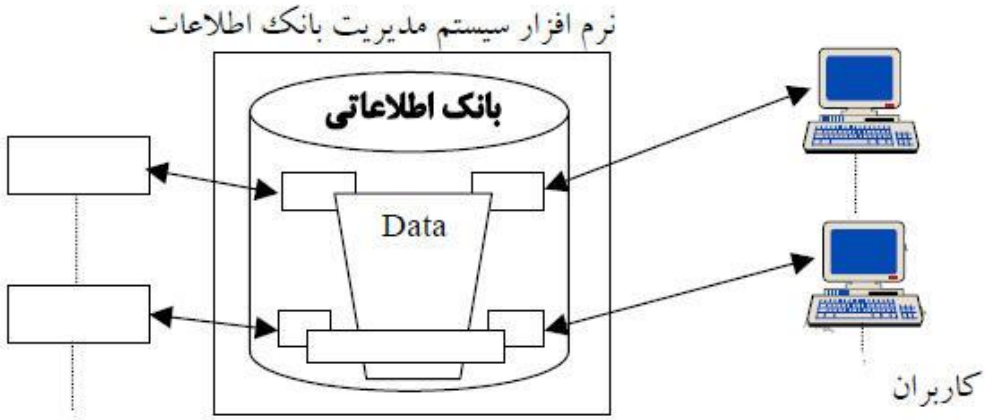
*نسخه های مختلفی از اطلاعات یکسان در فایل های مختلف ذخیره می شود.

*برنامه ها وابستگی شدیدی به داده ها دارند،تغییر ساختار داده ها منجر به تغییر برنامه می شود و تغییر برنامه ، ساختار فایل را تغییر می دهد.

*داده ها دارای استقلال نیستند و ساختار داده ها را دستورات برنامه تعیین می کند.

*امنیت پائین است و امکان کنترل دسترسی متمرکز و یکپارچه به داده ها وجود ندارد.

۲)روش پایگاه داده(Data base)



در این روش اطلاعات به صورت منسجم و سازگار ذخیره شده و می توانند به صورت اشتراکی توسط کاربران مختلف استفاده شوند. و هیچ کدام از عیب های روش فایلینگ را ندارند.

۱-۳- سیستم بانک اطلاعاتی

یک سیستم بانک اطلاعاتی از اجزای زیر تشکیل شده است.

۱)سخت افزار ۲)کاربران

۳)نرم افزار ۴)داده ها

۱-۳-۱- داده ها

منظور از داده ها همان داده های فیلدهای ذخیره شده است، فیلد ذخیره شده، کوچکترین واحد داده دارای معنا و مفهوم بوده و دارای انواع داده مشخص می باشند. رکورد ذخیره شده از فیلدهای ذخیره شده تشکیل شده است و

مشخصات مربوط به یک نمونه از واقعیت ها یا موجودیت ها از جهان واقعی را نگه می دارد. مجموعه ای از رکوردهای ذخیره شده را فایل ذخیره شده می گویند. فایل ذخیره شده، مشخصات نمونه های مختلف موجودیت را نگهداری می کند. به علاوه اطلاعاتی در مورد داده های ذخیره شده مثل ساختاررکورد، ساختار فایل ، طول رکوردها و ... در بانک اطلاعاتی نگهداری می شوند که شبه داده ها (Meta-Data) گفته می شوند.

بر اساس این تعاریف، بانک اطلاعاتی ذخیره شده عبارت از مجموعه ای از فایل های ذخیره شده است.

۱-۳-۲-نرم افزار

اصلی ترین نرم افزار در سیستم بانک اطلاعاتی، سیستم مدیریت پایگاه داده (یا DBMS -DataBase Management System) است. این نرم افزار امکانات زیر را فراهم می کند.

۱-ایجاد بانک اطلاعاتی و حذف آن و عملیات دیگر

۲-ایجاد جداول (فایل هاس اطلاعاتی)،تعیین ساختار آن ها،اصلاح و حذف آن ها

۳-ورود داده ها،بازیابی،ویرایش و حذف آن ها

علاوه بر امکانات اساسی فوق هرDBMS می تواند دارای امکانات متنوع دیگری نیز باشد از جمله :

۱-کپی،تغییر نام،پشتیبان گیری و بازیابی پشتیبان،ترمیم بانک اطلاعاتی،صدور/ورود اطلاعات

۲-امکان ارتباط بین جداول،ایجاد دیدگاه(QUERY) ،جستجو در جدول و خالی کردن جدول و ...

۳-حذف گروهی،یا به طور کلی عملیات گروهی روی داده ها

۴-امکانات دیگری مثل ساخت گزارش های دلخواه،برنامه نویسی برای مدیریت داده ها و طراحی فرم

چند نوع از DBMS ها عبارتند از Access,SQL Server,Oracle، DB2 ، Paradox،MySQL و...

۱-۳-۳- کاربران

چند دسته کاربر در یک سیسم بانک اطلاعاتی کار می کند:

۱-مدیر داده ها (Data Administrator) (DA) : فردی خبره در سازمان است که وظیفه تعیین چارچوب اطلاعاتی و جزئیات داده های ذخیره شده را تعیین می کند. این فرد ممکن است تخصصی در زمینه IT نداشته باشد ولی در مورد اطلاعات سازمان و روند فعالیت ها تسلط کافی دارد.

۲- مدیر بانک اطلاعاتی(Database Administrator) (DBA) یک فرد با تخصص در زمینه IT است که وظیفه تعیین ساختار بانک اطلاعاتی ومدیریت آنرا برعهده دارد. این کاربر زیر نظر مدیر داده ها اقدام به ایجاد بانک اطلاعاتی و تعیین نوع داده ها نموده و دسترسی کاربران به بانک اطلاعاتی را با توجه به ملاحظات امنیتی و سیاست های مدیریت سازمان کنترل می کند، ضمن اینکه خط مشی تهیه پشتیبان از بانک اطلاعات و ترمیم آن را تعیین کرده و موظف به نگهداری اطلاعات و رفع اشکلات احتمالی در حداقل زمان ممکن است.

۳-برنامه نویسان کاربردی: این کاربران با استفاده از زبان های برنامه سازی، برنامه های کاربردی را برای کار با بانک اطلاعاتی و مدیریت داده ها ایجاد می کنند و برای این کار از زبان های برنامه نویسی مثل زبان ویژوال بیسیک، دلفی، سی شارپ و ... استفاده می کنند. ،این زبان ها اغلب دارای امکاناتی برای ارتباط با بانک اطلاعاتی و مدیریت آن هستند و گاهی نیز ممکن است برنامه نویسان کتابخانه های لازم را برای ارتباط با بانک اطلاعاتی ایجاد کنند.

٤-کاربران نهایی: کاربرانی هستند که وظیفه ثبت اطلاعات و مدیریت آن ها را بر عهده دارند مثل کارمند آموزش، دانشجویان یا کارمند بایگانی. ارتباط این کاربران با بانک اطلاعاتی به دو شکل است:

۱-استفاده از برنامه کاربردی نوشته شده توسط گروه قبل(برنامه کاربردی درون خطی Online)

۲-استفاده از امکانات DBMS (برنامه های کاربردی پیش ساخته(Built-in)

برای استفاده این دسته از کاربران، DBMS ها دو دسته امکانات را فراهم می کند :

- رابط های منو گرا(فرم گرا): استفاده از پنجره ها و منوها
- رابط های دستور گرا: گرفتن دستورات با یک زبان پرس و جو مثل SQL

۱-۳-٤- سخت افزار

در یک سیستم بانک اطلاعاتی سخت افزارهای مورد استفاده عبارتند از:

- پردازنده ها
- حافظه های اصلی
- رسانه های ذخیره سازی

۱-٤- مزایای شیوه بانک اطلاعاتی

شیوه بانک اطلاعاتی مزایای زیادی دارد که مهمترین انها عبارتند از:

۱-اشتراک داده ها :

به عنوان مثال بخش های مختلف آموزشکده می توند از اطلاعات دانشجویان به صورت مشترک استفاده کنند.و حتی برنامه های کاربردی بعدی هم می توانند از داده های موجود استفاده کند.

۲-کاهش افزونگی داده ها:

داده ها تکرار نمی شوند و هر داده ای فقط یک بار در بانک اطلاعاتی نوشته می شود به عنوان مثال ما دو مجموعه از مشخصات دانشجویان را نخواهیم داشت

۳-اجتناب از ناسازگاری داده ها:

ناسازگاری یعنی اینکه یعنی واقعیت یکسانی به چند شکل در سیستم وجود داشته باشد.به عنوان مثال برای یک دانشجو در یک سیستم دو معدل و یا دو آدرس متفاوت ثبت شده باشد.

*کاهش افزونگی باعث کاهش ناسازگاری هم میشود.

به عنوان مثال دیگری مثلا دانشجویی که وجود ندارد برای او انتخاب واحد شده است در صورتی که کاربر شماره دانشجویی را اشتباه وارد کند و یا در صورتی که دانشجویی حذف شود و انتخاب واحد او حذف نشود این مشکلات پیدا خواهد شد.

در شیوه بانک اطلاعاتی، این مشکلات با بهنگام سازی انتشاری حل می شود.

٤-حمایت از تراکنش ها(Transactions):

تراکنش مجموعه ای از چند دستور است که به عنوان یک واحد تلقی شده یا همه ی آن ها باید اجرا شوند یا هیچکدام اجرا نشوند. مثل برداشت از حساب A و واریز به حساب B. بانک اطلاعاتی با حمایت از تراکنش ها ،درستی داده ها و عملیات و بعدها ترمیم بانک اطلاعاتی در صورت وجود مشکل، اجرای تراکنش ها را به شکل درست تضمین می کند.

٥-اعمال جامعیت یا یکپارچگی داده ها(Integration):

در شیوه بانک اطلاعاتی، برای اعمال جامعیت مجموعه ای از قوانین یا قواعد یا محدودیت ها در مورد ستون های جدول ،سطرهای جدول و یا چند جدول تعریف می شود این قوانین درستی و سازگاری داده ها را تضمین می کند. بانک اطلاعاتی می تواند امکان تعریف این قوانین و در نظر گرفتن آنها هنگام درج اطلاعات و تغییرات بعدی جامعیت داده ها تضمین کند.

٦-اعمال محدودیتهای امنیتی:

شیوه بانک اطلاعاتی، امکان تعیین کاربران و تعیین اختیارات آنها در مورد بانک اطلاعاتی،جداول، ستونهای جدول و حتی اجازه ی درج ،ویرایش و یا حذف هر کدام از ستون ها فراهم می کند.

٧-متعادل نمودن نیازهای متضاد:

در صورت استفاده از یک بانک اطلاعاتی مشترک در همه بخش های سازمان، کاربران ملزم به کوتاه آمدن از خواسته های غیر واقعی و پذیرفتن یک بانک اطلاعاتی متعادل و قابل استفاده برای تمام بخش های سازمان می شوند.

٨-اعمال استانداردها:

در صورت استفاده از بانک اطلاعاتی راحت تر می توان استانداردهایی مثل نحوه نمایش داده ها و دیگر استانداردهای سازمانی، منطقه ای، ملی و بین المللی را اعمال کرد و رعایت این استانداردها، تبادل اطلاعات بین سیستم های مشابه را راحت تر می کند.

٩-استقلال داده ها:

در شیوه فایلینگ، تغییرات برنامه می تواند تغییرات ساختار داده ها را درپی داشته باشد، ضمن اینکه تغییر ساختار داده ها جز با تغییر دستورات برنامه امکانپذیر نیست . بعبارتی دیگر دستورات برنامه هستند که ساختار داده ها و شیوه دستیابی به آنها را تعیین می کنند.

در شیوه بانک اطلاعاتی، دو نوع استقلال داده ها وجود دارد: استقلال منطقی و استقلال فیزیکی.

یک نرم افزار بانک اطلاعاتی برای ساخت بانک و جداول آن با مفاهیمی سرو کار دارد مثل فایل ها،شاخص ها،بلوک ها،رکوردها و... ولی این پیچیدگی ها را از دید کاربر مخفی کرده و به عنوان مثال می تواند بانک اطلاعاتی را به شکل رابطه ای یعنی بصورت جداول،ستون های جدول و روابط بین جداول ارائه دهد.

همچنین برنامه های کاربردی با این جداول کار کرده و دید سطح بالایی نسبت به بانک اطلاعی دارند و اگر ساختارهای فیزیکی بانک اطلاعاتی تغییر کند(به عنوان مثال بانک اطلاعاتی از یک نرم افزار مثل Access به نرم افزار دیگری مثل SQL Server منتقل شود) نیازی به تغییر دستورات برنامه کاربردی نیست.

از طرفی دیگر برنامه های کاربردی هر کدام می توانند دیدگاه خودشان را نسبت به بانک اطلاعاتی داشته و بخشی از آن را استفاده کنند. به عنوان مثال یک برنامه کاربردی ممکن است شماره دانشجویی را به صورت کاراکتری در نظر بگیرد در صورتی که برنامه کاربردی دیگری شماره دانشجویی را به صورت عددی صحیح یا اعشاری ببیند و با آن کار کند.بر عکس نیز ممکن است مدیر بانک اطلاعاتی ستون هایی از جدول را تغییر دهد مثلا ستون را اضافه یا حذف کرده یا تغییر دهد که نیازی به تغییر برنامه های کاربردی نباشد و یا تنها برنامه های کاربردی خاصی تغییر کنند.

فصل دوم

معماری سیستم بانک اطلاعاتی

معماری یک سیستم یعنی چارچوب برای سیستم بعبارتی دیگر معماری یعنی اجزای کلی سیستم و ارتباط بین آنها .معماری یک سیستم یک دید کلی نسبت به سیستم فراهم کرده و به عنوان یک استاندارد می تواند توسط سازندگان آن سیستم استفاده گردد.

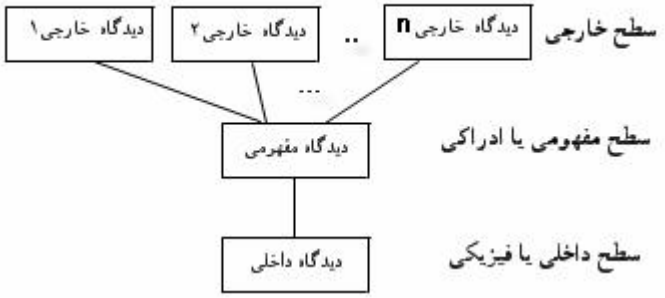
۲-۱- معماری ANSI/SPARC

یکی از معماری های در سیستم‌های بانک اطلاعاتی معماری ANSI/SPARC است در این معماری سیستم بانک اطلاعاتی به سه سطح تقسیم می شود:

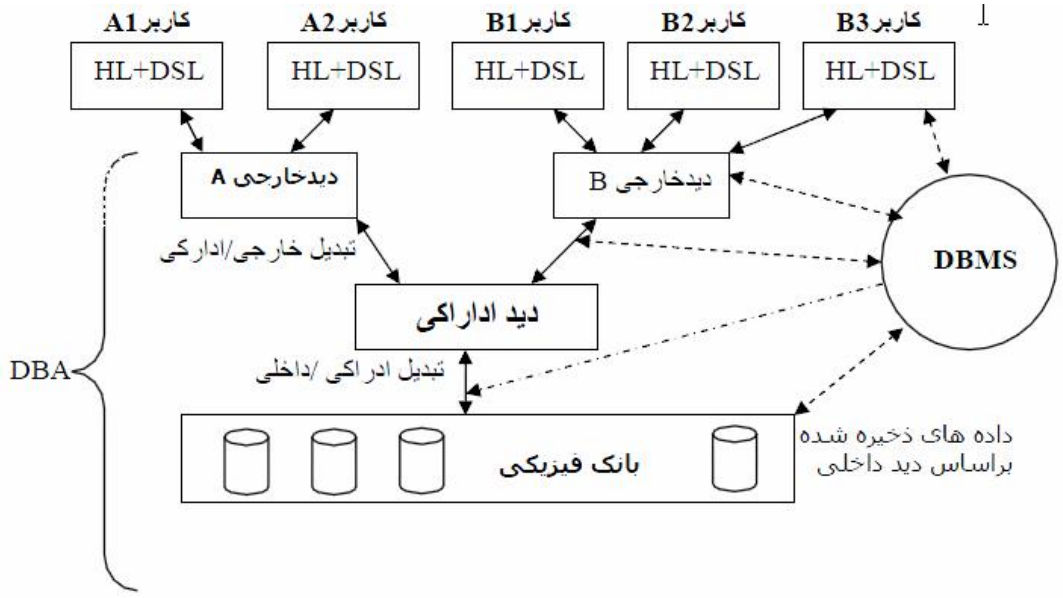
سطح داخلی(فیزیکی):سطحی که به مفاهیمی مثل رکورد ذخیره شده ،بلوکها،فایل ها و... و تکنیک هایی برای فشرده سازی و رمزگذاری داده ها در سطح سیستم فایل مربوط می شود. نرم افزار مدیریت بانک اطلاعاتی با این سطح سطح سرو کار دارد در حالی که این سطح از دید کاربران و مدیر بانک اطلاعاتی مخفی است.

سطح خارجی: سطحی که شامل دیدگاه های کاربران نسبت به بانک اطلاعاتی است هر کاربر و یا برنامه کاربردی با توجه به وظایف و اختیاراتی که دارد تنها به بخشی از بانک اطلاعاتی وبصورت محدود دسترسی داشته و برداشت خاصی نسبت به اطلاعات دارد.هر کاربر می‌تواند از یک زبان برنامه نویسی خاص مثل Delphi یا VB یا C# یا ... و از یک زبان پرس و جو مثل SQL برای کار با بانک اطلاعاتی استفاده نمایداستفاده کند. گاهی اوقات محدودیت‌های زبان برنامه نویسی باعث این تفاوت دیدها می شود. بعنوان مثال ممکن است یک کاربر یک فیلد عددی را به شکل متنی دیده و استفاده کند.

سطح مفهومی:شامل اجتماع دیدگاه کاربران نسبت به بانک اطلاعاتی است که همان دیدگاه مدیر بانک اطلاعاتی است.در این دیدگاه تمام جداول بانک اطلاعاتی و تمام ستون ها و سطرهای آن به همان شکلی که وجود دارند دیده می شوند.بین سطح فیزیکی و سطح مفهومی باید یک نگاشت از بانک اطلاعاتی ارایه شده برای کاربران به فایل های ساختاری که روی رسانه ذخیره سازی ایجاد شده است و نگاشتی هم بین دیدگاه‌های خارجی و مفهومی وجود داشته باشد.



شکل کامل تر مربوط به سطوح بانک اطلاعاتی و نگاشت های بین سطوح در ذیل آمده است:



همانگونه که شکل فوق نشان داده شده است، هر کاربر از طریق یک زبان میزبان(Host Language) (زبان برنامه نویسی مثل .....,Delphi,VB)ویک زیر زبان داده ها (Data Sub-Language) یا بطور خلاصه DSL مخصوص برای دسترسی به بانک اطلاعاتی استفاده می کند. منظور از DSL مجموعه امکانات و دستوراتی است که در یک زبان برنامه نویسی برای ارتباط با بانک اطلاعاتی و دستکاری داده ها وجود دارد.

DSL شامل دو مجموعه از دستورات است

DDL : Data Definition Language زبان تعریف داده ها شامل دستوراتی برای ایجاد بانک اطلاعاتی، تعیین ساختار جداول و اصلاح ساختارها .

DML: Data Manipulation Language زبان دستکاری داده ها شامل دستوراتی برای درج، حذف و ویرایش داده ها و بازیابی آنها.

چند کاربر یا برنامه کاربردی ممکن است دیدگاه خارجی یکسانی نسبت به بانک اطلاعاتی داشته باشند.

برای هر دیدگاه خارجی باید شمای مخصوص آن ایجاد شود.

دیدگاه مفهومی اجتماع دیدگاه خارجی است و از اجتماع شماهای خارجی شمای مفهومی بدست می اید.

در سطح داخلی بانک اطلاعاتی ذخیره شده را داریم با ساختار مخصوص خودش.

ساختار داده ها در هر سطح با استفاده از شماهای مخصوص آن سطح نشان داده می شود. بین دیدگاه های خارجی و مفهومی و نیز بین دیدگاه مفهومی و دیدگاه داخلی نگاشت صورت گیرد. منظور از نگاشت (mapping) تبدیل داده ها، دستورات و ساختارها از یک سطح به سطحی دیگر است. این تبدیلات در زمان ایجاد بانک اطلاعاتی و شکل گرفتن دیدگاه و همچنین در زمان دسترسی به بانک اطلاعاتی و خواندن یا نوشتن داده ها باید انجام شود.

DBA وظیفه تهیه و تدوین شماهای خارجی، شمای مفهومی، نگاشت بین این دیدگاه ها و نظارت بر کیفیت ذخیره سازی بانک اطلاعاتی را بر عهده دارد.

۲-۲- وظایف DBMS

DBMS در تشکیل تمام دیدگاه ها و نگاشت بین دیدگاه ها نقش اساسی را داشته و امکانات لازم را برای آن ها فراهم می نماید. برخی از وظایف DBMS عبارتند از:

- تعریف داده ها

سیستم مدیریت بانک اطلاعاتی قادر به قبول تعاریف داده ها (شمای خارجی، شمای مفهومی، شمای داخلی و تمام نگاشت های متاظر) به شکل منبع بوده و آنها را به شکل مقصد مناسب تبدیل می کند. بخشی از DBMS به نام پردازنده DDL می باشد. **دستورات DDL** بر اساس شماها و نگاشت ها می باشند به پردازشگر DDL داده شده تا بر اساس آنها تعاریف و اشیایی تولید کرده و به عنوان شبه داده ها ذخیره کند.

- دستکاری داده ها

DBMS باید قادر باشد که درخواست های بازیابی، به هنگام رسانی یا حذف داده های موجود در بانک اطلاعاتی و همچنین اضافه کردن داده های جدیدی را به بانک اطلاعاتی انجام دهد. برای این کار دارای بخشی به نام پردازنده DML است. **دستورات DML** دو دسته هستند :

دستورات DML برنامه ریزی شده که در زبان طراحی بانک اطلاعاتی مشخص هستند و DBMS سعی می کند ساختار بانک اطلاعاتی را طوری تعیین که این دستورات پرس و جو بهتر اجرا شوند.

دستورات DML برنامه ریزی نشده بر اساس نیازهای پیش آمده اجرا می شوند. این در خواست ها می توانند به یک زبان پرس و جو مثل SQL نوشته شوند.

- بهینه سازی و اجرا

تمام درخواست های DML که توسط پردازشگر DML کمپایل می شود و در خواست های کمپایل شده بهینه سازی می شود تا سریع تر اجرا شوند. پرس و جوهای بهینه شده توسط بخشی به نام مدیر زمان اجرا،اجرا می شوند.در تمام این مراحل از شبه داده ها استفاده می شوند.

- جامعیت و امنیت داده ها

DBMS باید بر درخواست های کاربران نظارت داشته باشد و از هرگونه تلاشی که محدودیت های جامعیت و امنیت تعریف شده توسط DBA را برهم می زند جلوگیری کند.

- ترمیم و سازگاری داده ها

DBMS یا بعضی از نرم افزارهای مرتبط (مدیر تراکنش یا ناظر پردازش تراکنش) باید کنترل های خاصی را برای ترمیم و سازگاری داده ها فراهم کنند.

- فرهنگ داده ها :

در بانک اطلاعاتی دو نوع داده ذخیره می شود: ۱-داده ۲-شبه داده

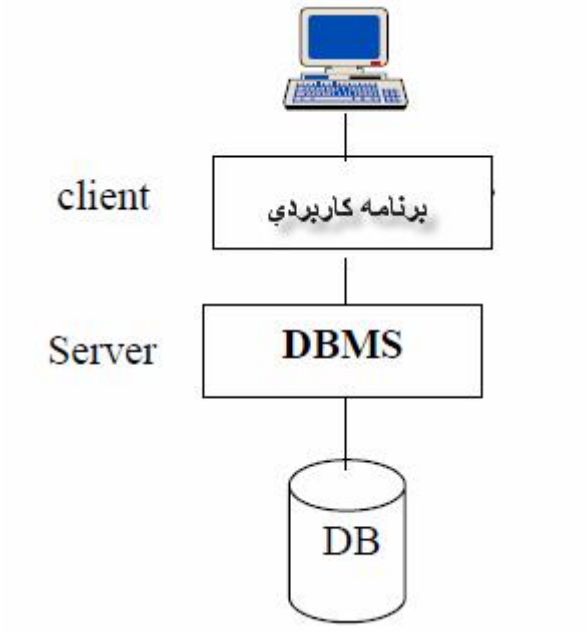
شبه داده ها : اطلاعاتی در مورد داده ها مثل مشخصات بانک اطلاعاتی،ساختار جداول و... را مشخص می کنند. این اطلاعات عبارتند از:

- تمام دیدگا ههای خارجی و ادراکی
- تمام نگاشت ها
- موجودیت ها (جداول)، صفات خاصه آنها و نوع داده هر فیلد
- ارتباط بین موجودیت ها و شاخص ها
- تاریخ ایجاد بانک اطلاعاتی، تاریخ اصلاح و دسترسی به آن

به کاتالوگ بانک اطلاعاتی، راهنما یا فرهنگ داده ها (Data Dictionary) نیز گفته می شود. کاتالوگ معمولا" بصورت یک بانک اطلاعاتی و یا مجموعه ای از جداول اطلاعاتی پیاده سازی و نگهداری می شود.

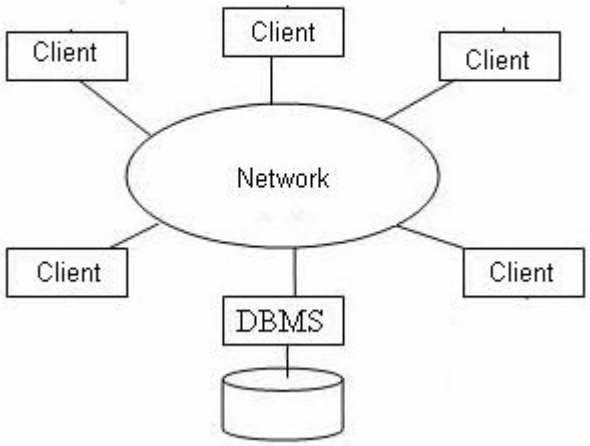
۲-٤ - معماری Client/Server

غالبا DBMS ها به صورت Client/Server هستند که شامل یک برنامه سرویس دهنده و چندین برنامه سرویس گیرنده می باشند.کاربران از طریق برنامه های سرویس گیرنده درخواست هایی را به سرویس دهنده ارسال می کنند و برنامه ای بنام مدیر ارتباطات داده ای این درخواست ها را گرفته و پس از اجرا توسط بخش های دیگر پاسخ را به سرویس گیرنده ها می فرستد. برنامه ها و نرم افزارهای سرویس گیرنده و سرویس دهنده می توانند روی یک سیستم یا روی سیستم های مختلفی قرار گرفته باشند.در هر صورت روش ارتباطی آنها یکسان می باشد. در این معماری بانک اطلاعاتی بر روی کامپیوتر سرویس دهنده قراردارد.



۲-۵- پردازش توزیعی

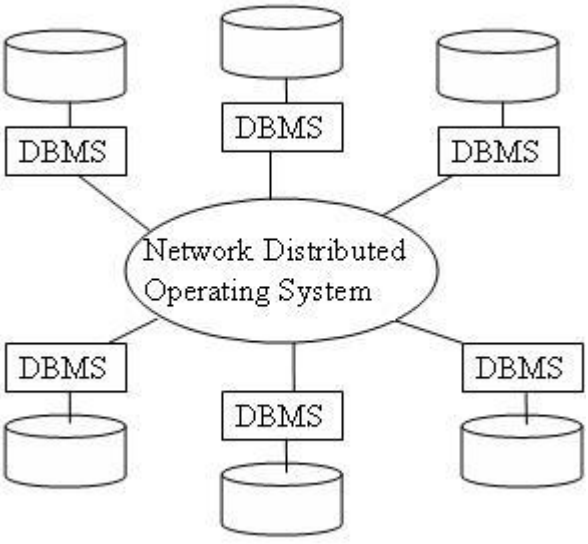
در سیستم های مشتری/خدمتگزار ممکن است چندین ماشین مشتری قادر به دستیابی به ماشین سرویس دهنده باشند. بنابر این چندین سیستم مشتری مجزا می توانند بصورت مشترک از یک بانک اطلاعاتی منفرد استفاده کنند.



یک ماشین سرویس دهنده، چند ماشین سرویس گیرنده

سیستم های دیگری نیز هستند که بر خلاف سیستم های مشتری/خدمتگزار بانک اطلاعاتی در کل شبکه پخش شده است. یعنی هر سیستم دارای یک بانک اطلاعاتی محلی با یک برنامه برای مدیریت بانک اطلاعاتی و چندین برنامه کاربردی بعنوان سرویس گیرنده دارد. برنامه های کاربردی بیشتر وقتشان را با داده های محلی کار می کنندو اگر به اطلاعات قرارگرفته در یک سیستم راه دور نیاز داشتند درخواستی به سرویس دهنده محل اطلاعات فرستاده و پاسخ را دریافت می کنند. در این سیستم ها ترافیک شبکه به شدت پائین آمده و دسترسی به داده های محلی سریع است، پردازش در کل شبکه توزیع شده است ضمن اینکه امکان استفاده اشتراکی از اطلاعات نیز وجود دارد.

روش کار سیستم طوری است که اولا: تمام اطلاعات در کل شبکه با هم همگام بود و ناسازگاری نداشته باشند. ثانیا: کاربر متوجه پراکندگی داده ها نشده و بدون دانستن محل قرارگیری آنها بتواند به آنها دسترسی پیدا کند.



هر ماشین هم سرویس دهنده و سرویس گیرنده

فصل سوم

بانک های اطلاعاتی رابطه ای

۳-۱- خصوصیات بانک اطلاعاتی رابطه ای

سیستم های رابطه ای بر پایه یک اصل رسمی یا نظریه ای به نام مدل رابطه ای داده ها، استوار هستند.
این سیستم ها از جنبه های زیر رابطه ای می باشند.

۱- جنبه ساختاری

داده های بانک اطلاعاتی فقط به صورت جدول در اختیار کاربر قرار میگیرند. این جداول ساختار منطقی هستند نه ساختار فیزیکی ، یعنی اینکه در سطح فیزیکی سیستم آزاد است که داده ها را به هر شکلی که می خواهد ذخیره کند ولی در سطح منطقی تبدیل به جدول ها می شوند. این مدل یک مفهوم انتزاعی از داده ها را ارائه می دهد و کاربر درگیر جزئیات ذخیره سازی نمی شود.

۲- جنبه جامعیت

هر نوع قانون برای تضمین جامعیت و یکپارچگی داده ها بر روی جداول تعریف می شود.

۳- جنبه دستکاری

تمامی عملیات قابل اجرا روی داده ها به صورت عملکردهایی روی جدول جداول اجرا شده و نتیجه آنها نیز به شکل جدول می باشد.

این عملکردها عبارتند از:

الف- محدودیت (Restriction) : عملی که باعث بازیابی بعضی سطرهای جدول می شوند. یعنی سطرهایی که شرط خاصی را دارا باشند.

ب- تصویر(Projection) : عملی که باعث بازیابی تعدادی از ستون های جدول می شود.یعنی ستون مورد نظر ما.

ج- الحاق (Join) : عملی که بر روی دو یا چند جدول بر اساس ستون های مشترک آنها اجرا شده و یک جدول شامل ستون های همه آنها تولید می کند.

مثال: با در نظر گرفتن دو جدول برای دانشجویان (students) و رشته ها (Fields) بصورت زیر عملکردهای مختلف را اجرا می کنیم

Students

id	Name	Avg	Fid
100	ali	17	1
101	reza	15	2
102	sara	19	2

Fields

fid	Title
1	Computer
2	Accounting
3	Graphics

الف – عملکرد محدودیت :

Students WHERE Avg > 16

نتیجه :

Id	Name	Avg	Fid
100	ali	17	1
102	sara	19	2

ب- عملکرد تصویر

Students OVER Name , Avg

نتیجه :

Name	Avg
ali	17
reza	15
sara	19

ج- عملکرد الحاق

Fields and Students OVER Fid

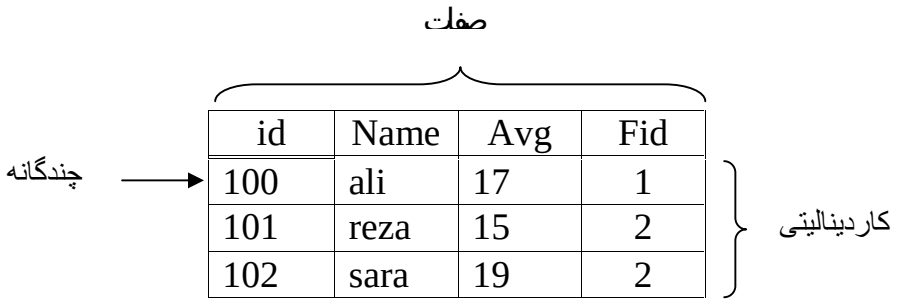
نتیجه :

Fid	Title	Id	Name	Avg
1	computer	100	ali	17
2	accountry	101	Reza	15
2	accountry	102	Sara	19

گاهی اوقات برای اجرای یک عمل نیاز به چند عمل کوچکتر است نتیجه هر کدام از این اعمال میانی هم به صورت جدول است.جدول هایی که به طور موقت ایجاد شده و از بین میرود و آخرین جدول (نتیجه آخرین عمل)به عنوان نتیجه نشان داده می شود.

جنبه های فوق را خصوصیات غیر رسمی بانک اطلاعاتی نیز می گویند.

۳-۲- معرفی چند اصطلاح رسمی



اصطلاحات رسمی جداول رابطه ای	اصطلاحات غیر رسمی در جداول رابطه ای
چندگانه	سطر
صفت	ستون یا فیلد
کلید اصلی	یک کلید منحصر به فرد
رابطه	جدول
کاردینالیتی	تعداد سطرها
درجه	تعداد ستون ها
دامنه	نوع داده یک ستون

چندگانه: همان سطح های جدول را می گویند. که عبارت از چند مقدار است.مثلا چندگانه<reza,15,2,101>

صفات: هر صفت عبارت است از چند مقدار ثبت شده برای یک خصوصیت است.مثلا صفت sara به عنوان نام یک دانشجو معادل ستون های جدول.

درجه: تعداد صفات (تعداد ستون های جدول)

کاردینالیتی: تعداد چندگانه ها; تعداد سطرهای جدول

دامنه: محدوده ای از داده هایی که می توان برای یک صفت در نظر گرفت.به عنوان مثال برای صفت Avg دامنه آن اعداد اعشاری بین 0 و 120 است.

رابطه: به مجموع مفاهیم فوق رابطه می گویند.اگر هر کدام از چندگانه های جدول را یک گزاره ی درست در نظر بگیریم رابطه مجموعه ای از گزاره های درست است.

۳-۳- خصوصیات رسمی بانک های اطلاعاتی رابطه ای:

در یک رابطه (جدول):

۱- ترتیب قرار گرفتن چندگانه ها اهمیتی ندارد.

۲- ترتیب قرار گرفتن چندگانه ها اهمیتی ندارد.

۳- یک رابطه هیچ گاه چندگانه تکراری ندارد.اگر برای یک جدول کلید اصلی در نظر بگیریم هیچ گاه چندگانه تکراری نخواهیم داشت.

۴- هر چندگانه برای هر صفت فقط یک مقدار است.

۳-٤- کلید ها در جدول

بعضی از ستون های جدول مقادیر منحصر به فرد دارند و یا ممکن است ترکیبی از چند ستون جدول مقدار منحصر به فرد داشته باشد که به این ستون یا ترکیب این ستون ها ،ستون های کلیدی گفته می شود. انواع کلیدها در جدول عبارت است از:

کلیدهای کاندید(CK-Condidate Key)

ستون ها یا ترکیبی از چند ستون که مقدارشان منحصر به فرد است.مثل شماره دانشجویی، کد ملی و کد داوطلب در جدول داوطلبان کنکور.

کلید اصلی(PK-Primary Key)

یکی از کلیدهای کاندید که به عنوان مبنای دسترسی به اطلاعات استفاده میشود.مثل شماره دانشجویی در جدول شماره دانشجویان یا کد داوطلب در جدول داوطلبان کنکور . کلید اصلی ستونی غیر داده ای است. یعنی جزء مشخصات اصلی نیست که بخواهد در آینده تغییر کند . بلکه مشخصه ای است که به هر موجودیت اضافه شده و هیچگاه نیاز به تغییر آن نیست.

کلیدهای جانشین (Alternative Key)

کلیدهای کاندیدی که کلید اصلی نیستند مثل کد ملی در جدول دانشجویان

کلید خارجی (FK –Foreign Key)

ستون هایی از یک جدول که متناظر با کلید اصلی در جدولی دیگر باشند مثل کد رشته در جدول دانشجویان که متناظر با ستون کدرشته یعنی کلید اصلی در جدول است.

فصل چهارم

مدلسازی داده ها و طراحی بانک اطلاعاتی

٤-١- مدلسازی داده ها

یکی از فعالیت های مهم تحلیل و طراحی سیستم ها، تحلیل داده ها و طراحی بانک اطلاعاتی می باشد. تحلیل داده بعد از شناسایی عملکرد سیستم، فرآیندها و جریان داده ها در سیستم انجام می شود.

قبل از طراحی بانک اطلاعاتی باید داده های محیط تجزیه و تحلیل شده داده هایی که باید نگهداری شوند تعیین شده و ساختارهای مناسب برای نگهداری آنها طراحی گردند. برای مدلسازی داده ها از نمودارER استفاده می شود. این نمودار موجودیت ها، خصوصیات و رابطه بین موجودیت ها را به صورت تصویری و قابل فهم نمایش می دهد.

در یک محیط عملیاتی اطلاعاتی که باید ذخیره شوند عبارتند از:

1-نهادها(Entity-موجودیت): هر چیزی از دنیای واقعی که باید راجع به آن اطلاعاتی ثبت شود مثل کتاب،دانشجو . هر محیط عملیاتی دارای مجموعه بزرگی از موجودیت ها از جمله افراد، نقش ها، واقعه ها ، فعالیت ها، بخش ها، گزارشات و ... است که با توجه به وظایف و عملکردهای سیستم باید بخشی از انها انتخاب گردد.

2-صفات(Attribute-ویژگی):ویژگی هایی از موجودیت ها که باید برای آنها اطلاعاتی ثبت شود (مثل نام و نام خانوادگی شخصی،کد و عنوان و مولف کتاب ،شماره ی دانشجویی ورشته و آدرس دانشجو)

3-رابطه ها(Relation):ارتباط بین نهادها به عنوان مثال رابطه بین دانشجو ودرس،کتاب و ناشر و....

در هر محیط واقعی نهادهای خیلی زیادی وجود دارد و نهادها هم صفات متنوعی دارند بین نهادها هم ممکن است روابط مختلفی بر قرار باشد.

وظیفه ی تحلیلگران:بررسی نهادها و تعیین نهادهای لازم و صفات لازم می باشد به طوریک همه اطلاعات مورد نیاز ذخیره شده وسیستم بتواند وظایف مورد نظر را انجام دهد و قابلیت توسعه نیز تا حدودی داشته باشد.

نتایج تحلیل باید به صورت نمودارهایER نمایش داده شود.(ERD-Entity-Relation-Diagram)

این نمودارها میتوانند برای ارائه نتایج تحلیل به افراد گروه طراحی،افراد خبره،(کار فرما)و نیز نگهداری به عنوان مستندات سیستم جهت استفاده آتی(به منظور توسعه سیستم)مورد بهره برداری قرار گیرند.

٤-٢- اجزای نمودار ER

٤-٢-١- موجودیت

موجودیت ها به صورت مستطیل نشان داده می شوند و دو دسته اند: ۱- موجودیت قوی ۲-موجودیت ضعیف

موجودیت ضعیف وجودش بستگی به موجودیت دیگر دارد به عنوان مثال:کارمند یک موجودیت قوی است ولی اعضای خانواده کارمند موجودیت ضعیف هستند.

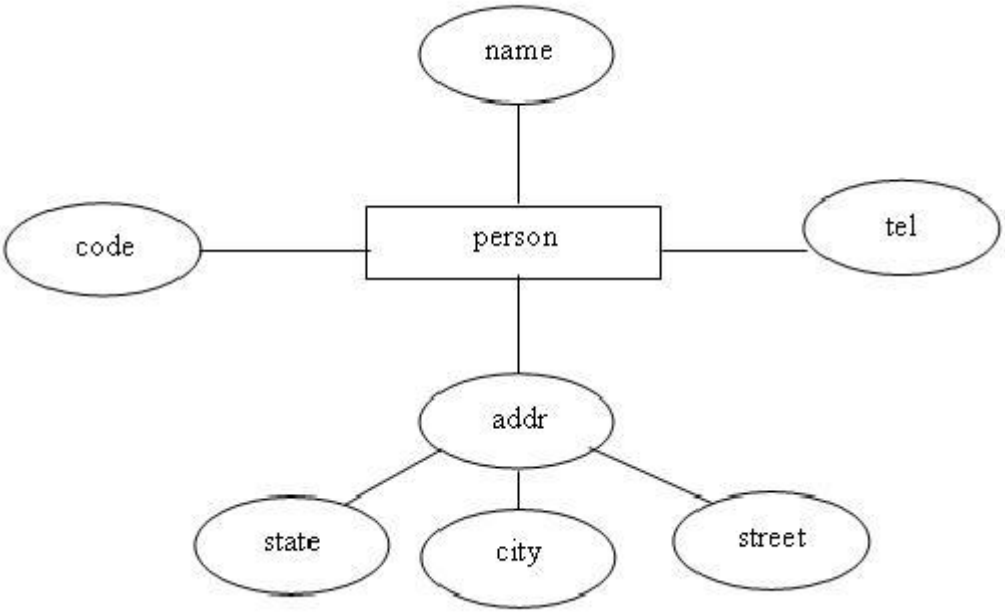
موجودیت قوی	موجودیت ضعیف

۲-**صفات خاصه:**ویژگی ها یا صفات موجودیت ها و روابط

صفات خاصه کلید: یک یا چند صفت خاصه که برای یک موجودیت منحصربه فرد است مثل شماره ی دانشجویی برای دانشجو،کد ملی برای شخص،کد درس برای درس

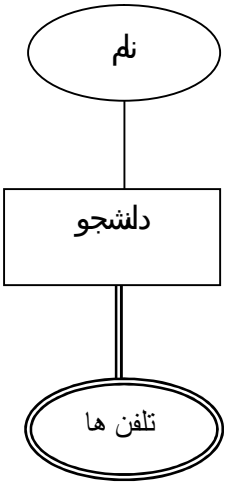
صفات خاصه ساده/مرکب: صفت ساده از یک بخش تشکیل شده و صفت مرکب از بیش از یک بخش.صفت ساده مثل:نام،سال تولد،نام خیابان،

صفت مرکب:آدرس که از استان، شهر و نشانی تشکیل شده یا تاریخ تولد که از سال ، ماه و روز تشکیل شده است.



٤-۲-۲- صفات خاصه

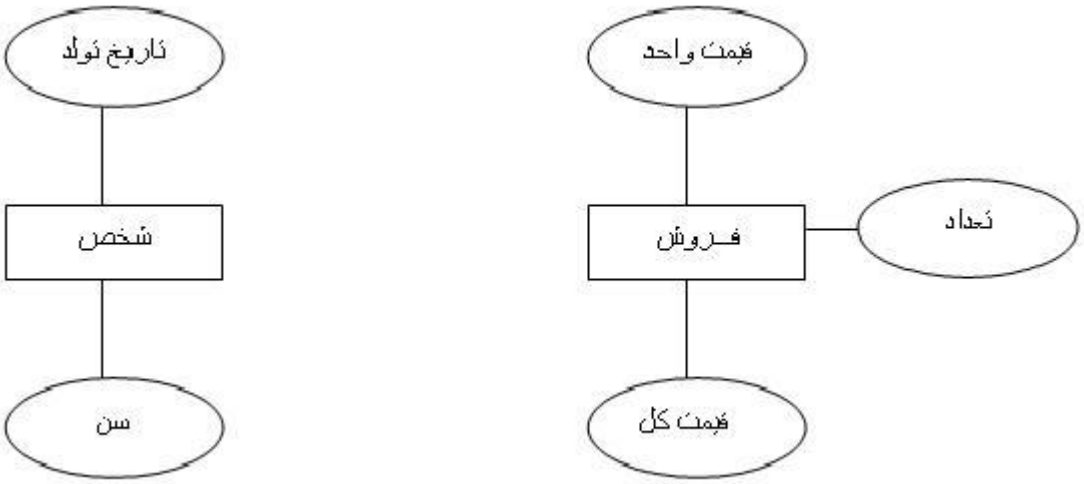
صفت خاصه تک مقداری یا چند مقداری: صفت تک مقداری همیشه یک مقدار دارد مثل :نام دانشجو،نام استاد.صفت چند مقداری گاهی اوقات می تواند بیش از یک مقدار داشته باشد.مثل:فعالیت علمی دانشجویا مدرک استاد



صفت خاصه تک مقداری

صفت خاصه چند مقداری

صفت خاصه مشتق(محاسبه شدنی): صفاتی که بر اساس صفات دیگر قابل محاسبه باشد.مثل:سن شخص که بر اساس تاریخ تولد و تاریخ جاری قابل محاسبه است.یا قیمت کل در جدول فروش که بر اساس قیمت واحد و تعداد قابل محاسبه است.



صفت خاصه هیچ مقدارپذیر:صفت خاصه ای که می تواند مقدار نداشته باشد،می تواند مقدار پوچ(null)بگیرد.بعبارتی صفتی که مقداردهی آن اجباری نیست.

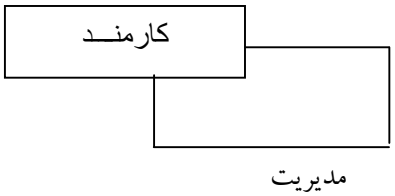
٤-۲-۳- رابطه ها

ارتباط بین نهادها (موجودیت ها)را رابطه گوییم. رابطه هایی را برای سیستم در نظر می گیریم که برای ما اهمیت داشته و باید آن ارتباط در سیستم ثبت شود به عنوان مثال بری نهاد دانشجو و نهاد درس رابطه ی انتخاب اهمیت دارد ولی رابطه ی مطالعه اهمیتی ندارد و نباید در نظر گرفته شود. یک رابطه دارای ویژگی هایی می باشد:

الف- درجه رابطه:

تعداد نهادهایی که در یک رابطه شرکت دارند و می تواند 1 و2و3ویا بیشتر باشد.

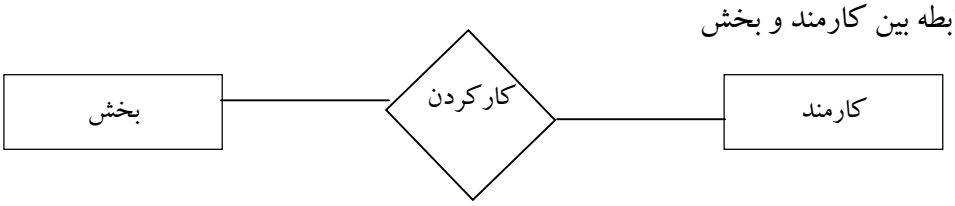
- **رابطه درجه یک**



یعنی یک کارمند میتواند مدیر یک یا چند کارمند دیگر باشد.

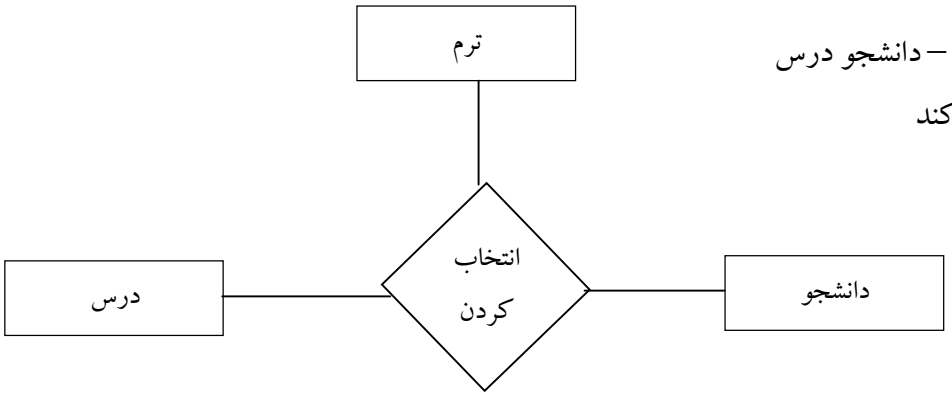
- **رابطه درجه دو**

رابطه بین دانشجو و درس یا رابطه بین کارمند و بخش



- **رابطه درجه سه**

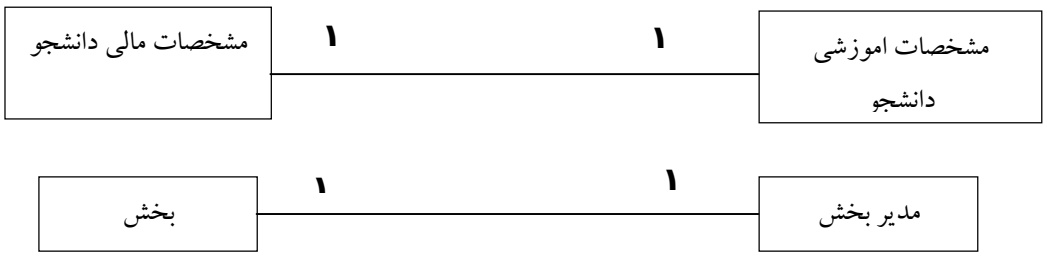
رابطه بین دانشجو و درس و ترم – دانشجو درس را در ترم مشخصی انتخاب می کند



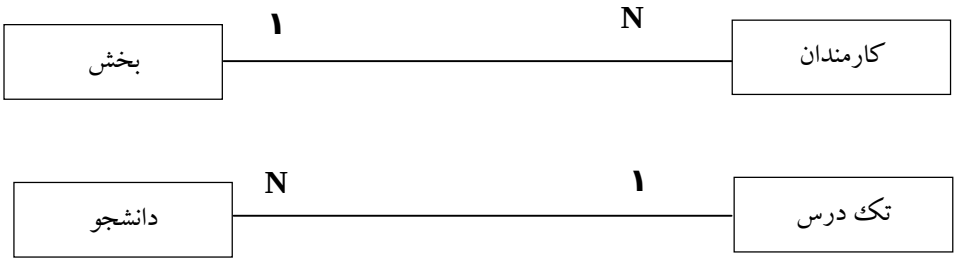
ب- کاردینالیتی رابطه:

تعداد نمونه هایی از موجودیت که در یک رابطه می تواند شرکت کند و شکل های مختلف ان عبارت است از :

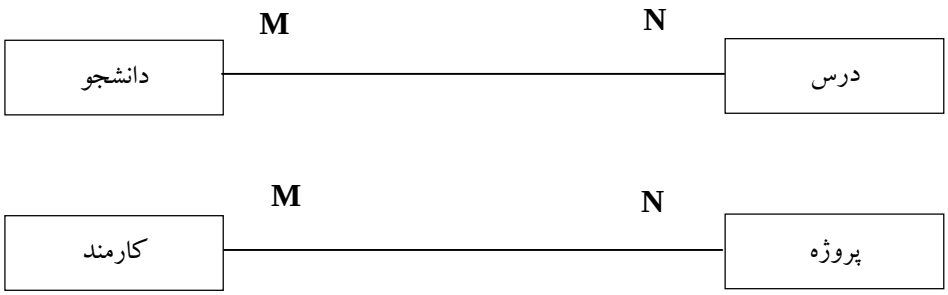
- **رابطه یک به یک (1:1)**



- **رابطه یک به چند (1 : N)**



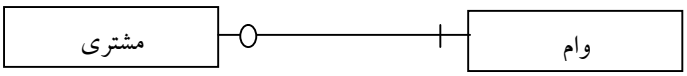
- **رابطه چند به چند (M : N)**



ج- الزام مشارکت (Modality)

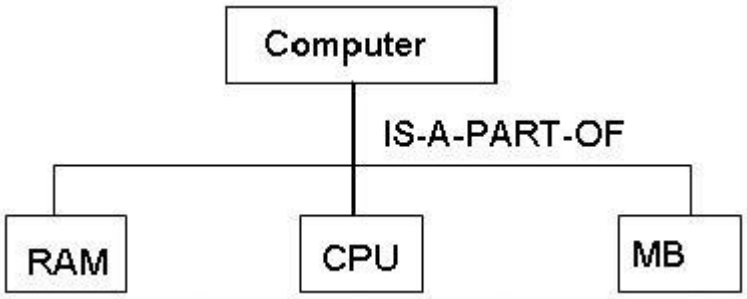
الزام مشارکت به دو صورت کامل (۱) و ناکامل(۰) می باشد . در مشارکت کامل تمام نمونه های یک موجودیت باید در رابطه شرکت داشته باشند و در مشارکت ناکامل الزامی به شرکت همه نمونه های موجودیت در ارتباط وجود ندارد.

به عنوان مثال در رابطه بین وام و مشتریان بانک ، موجودیت مشتری مشارکت ناکامل دارد، چون همه مشتریان ممکن است وام نگرفته باشند. ولی مشارکت وام در این رابطه الزامی است، چون هر وام مربوط به یک مشتری است و وامی بدون مشتری وجود ندارد.

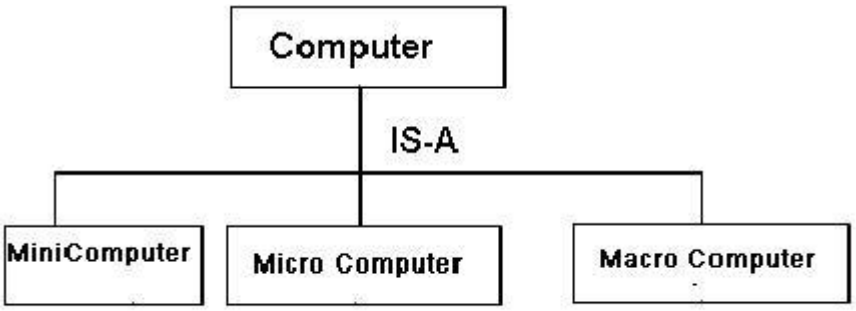


٤-۲-۲- موارد اضافه شده به نمودار ER:

رابطه تجمیع (aggregation):اگریک موجودیت از موجودیت های دیگری تشکیل شده باشد بین آنها رابطه ی تجمیع وجود دارد.یعنی یک کامپیوتر از اجزای MB،CPU، RAM تشکیل شده است.



رابطه ی تخصیص(specialization):اگر یک موجودیت دارای انواع خاصی هم باشد.



مثال ۱: سیستم درمانگاه

موجودیت ها و خواص آنها (صفات خاصه)

- پزشک (کد پزشکی،نام،نظام پزشکی،آدرس،تلفن،مدرک،.....)
- پرسنل (کد پرسنلی،نام،مدرک تحصیلی،مهارت ها،.....)
- بیمار (کد ملی،نام،نوع بیماری،آدرس،تلفن)

- بیمه ها (کد بیمه،عنوان)

- بخش ها (کد بخش،عنوان بخش)

مثال ۲ : سیستم آموزش

الف- موجودیت ها و خواص آنها

۱- دروس(کد درس، عنوان درس، تعداد واحد(عملی- تئوری) ، تعداد واحد اموزشی)

۲- دانشجویان (شماره ی دانشجویی، نام دانشجو شامل نام و نام خانوادگی، رشته ی تحصیلی، سال ورود، معدل ، امکانات مورد نیاز(یک دانشجو می تواند هیچ یا چند امکان را انتخاب می کند))

۳- کارکنان (کد پرسنلی، نام، نام خانوادگی،بخش، نوع کارمند(اداری – استاد))

۴- اساتید (کد پرسنلی ، نام، رشته تحصیلی، مدرک)

۵- بخش ها(کد بخش، عنوان بخش)

۶- رشته ها (کدرشته، عنوان رشته ، تعداد دانشجو)

ب- رابطه ها

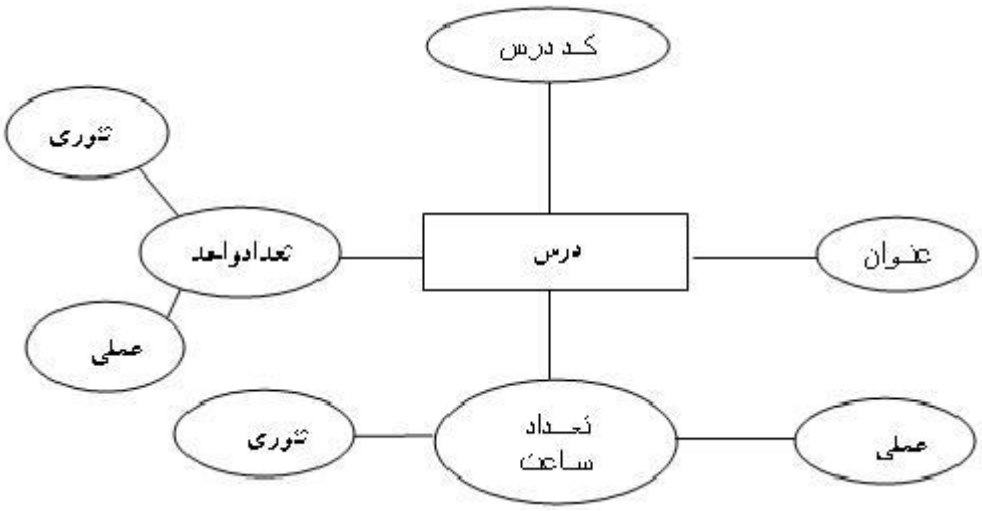
۱- دانشجو میتواند درس خاصی را با استاد خاصی انتخاب کند.

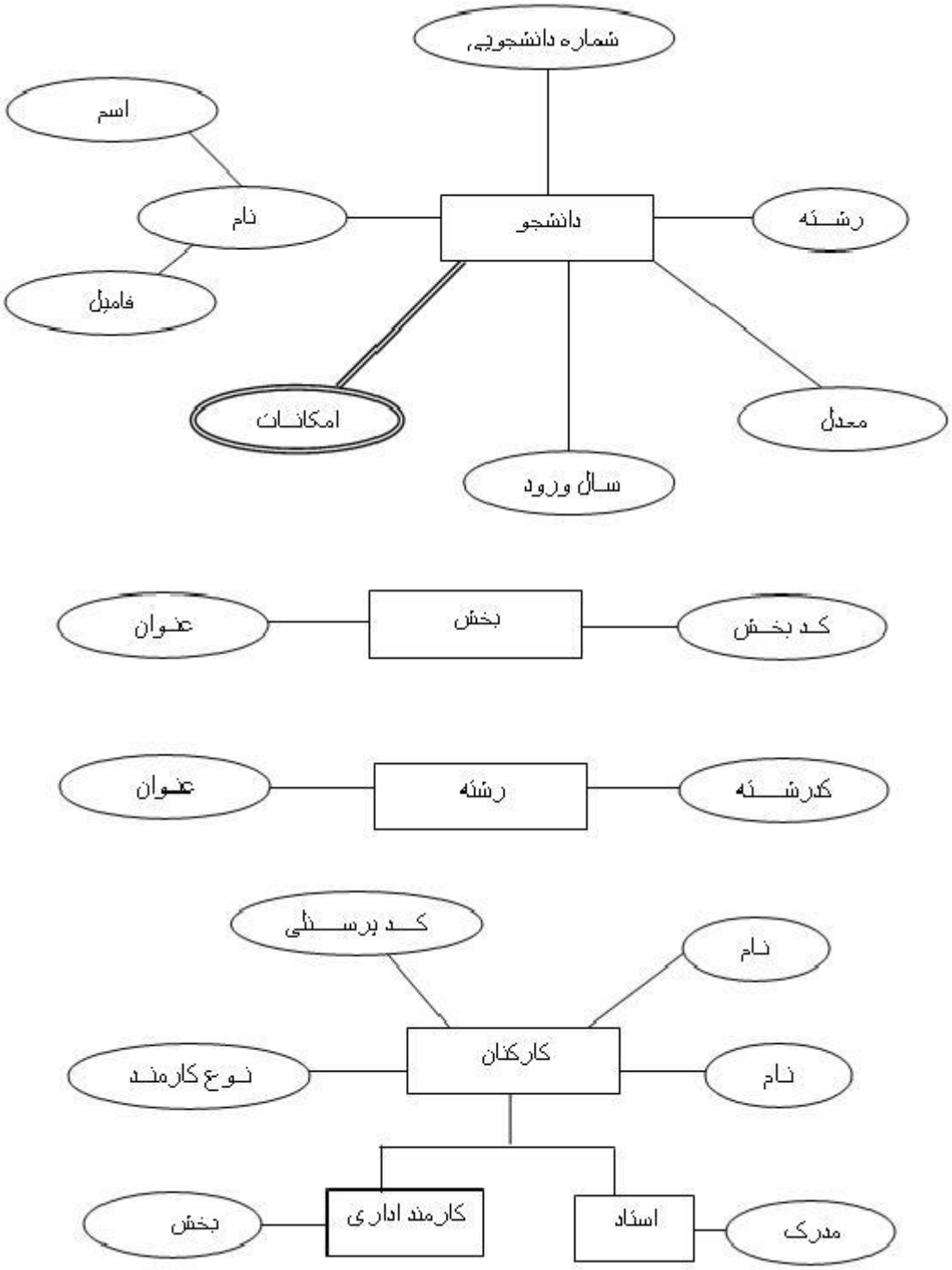
۲- استاد خاصی درس خاصی را ارائه می کند.

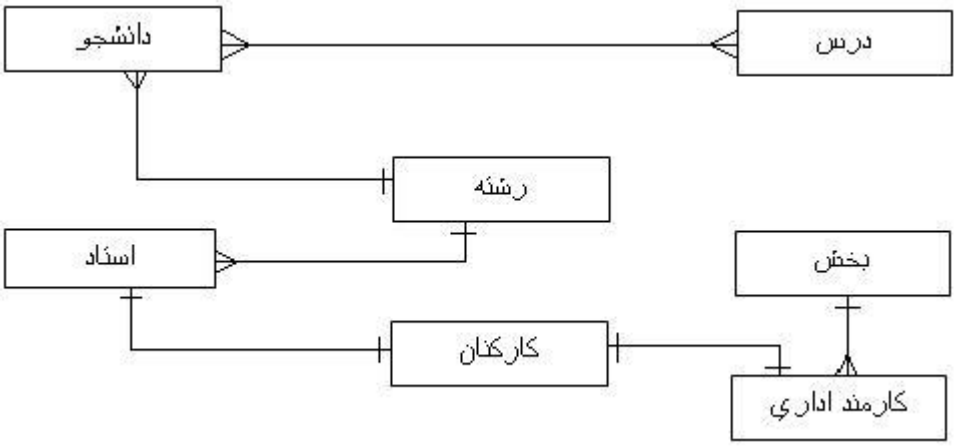
۳- کارمند خاصی در بخش خاصی کار می کند.

۴- استاد خاصی مربوط به رشته مشخصی است.

ج- رسم نمودارER




شماره دانشجویی
اسم
نام
فامیل
دانشجو
رشته
امکانات
معدل
سال ورود
عنوان
بخش
کد بخش
عنوان
رشته
کدرشته
کد پرسنلی
نام
کارکنان
نوع کارمند
نام
بخش
کارمند اداری
استاد
مدرک



٤-۳- تبدیل نمودار ER به بانک اطلاعاتی

برای تبدیل نمودار ER به بانک اطلاعاتی قواعد زیر را استفاده می کنیم :

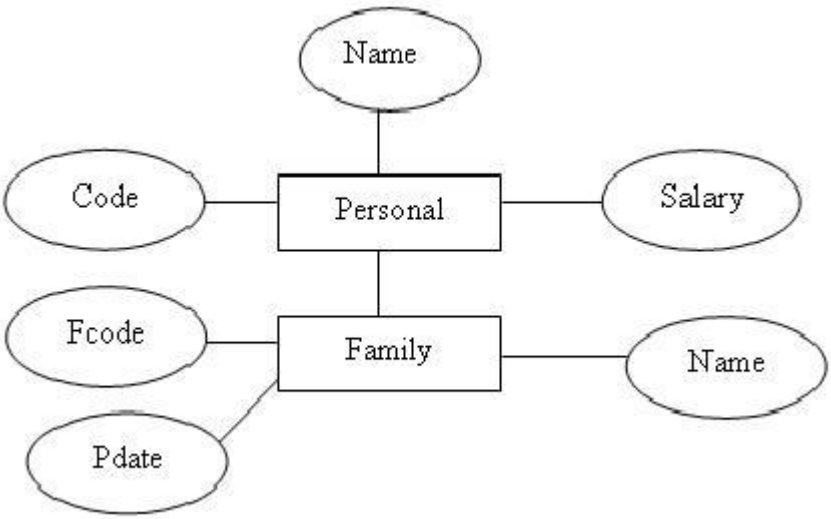
قاعده اول:

هر نهاد قوی به یک جدول تبدیل می شود که دارای ستون هایی برای هر کدام از صفات خاصه بوده و هر بخش از صفات خاصه مرکب بصورت یک ستون جداگانه پیاده سازی می گردد.

قاعده دوم:

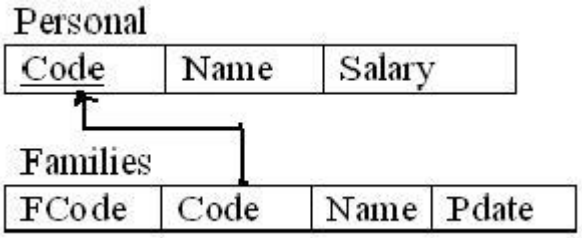
هر موجودیت ضعیف تبدیل به یک جدول شده و این جدول دارای یک کلید خارجی متناظر با کلید اصلی در جدول مربوط به موجودیت قوی است.

مثال:



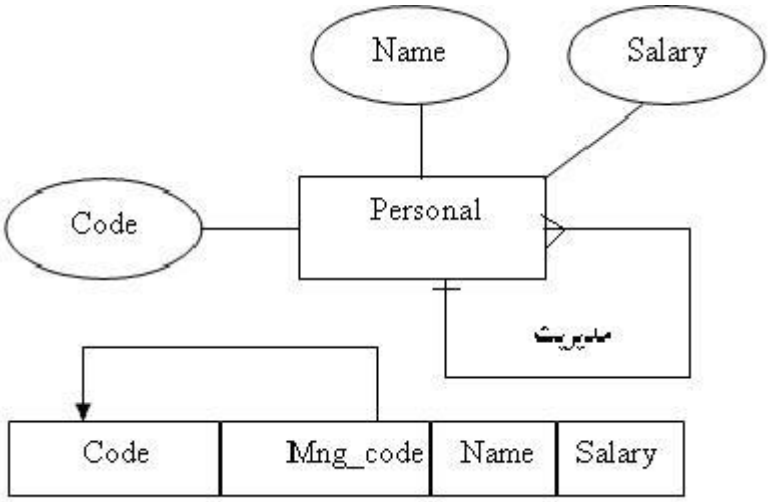
Personal: کارمند **Family:** اعضای خانواده کارمند

Family یک نهاد ضعیف است چون وجودش وابسته به نهاد Personal می باشد.یعنی اگر کارمندی وجود نداشته باشد اعضای خانواده ی او هم نمی توانند وجود داشته باشند. پیاده سازی آن بصورت زیر خواهد بود:



قاعده سوم:

برای یک رابطه ی درجه یک (رابطه ای که بین بک موجودیت و خودش است)، یک ستون اضافی به عنوان کلید خارجی در آن ایجاد کرده که در ارتباط با کلید اصلی جدول می باشد.

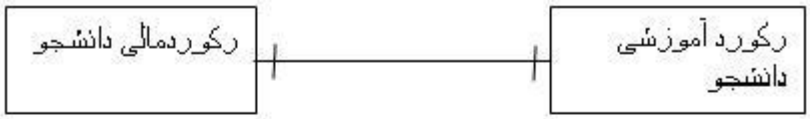


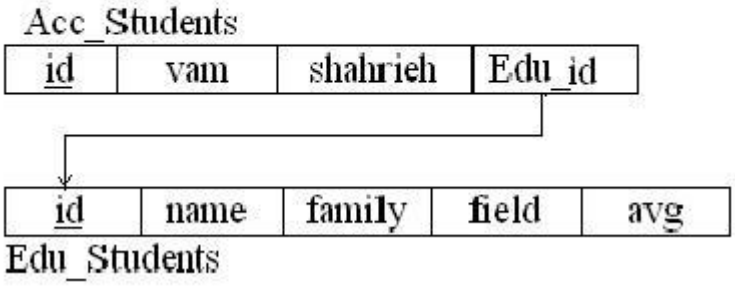
مثال:

Code	Mng_Code	Name	Salary
1	0	ali	200
2	1	reza	150
3	1	sara	120
4	3	hadi	110
5	3	zahra	150

قاعده چهارم:

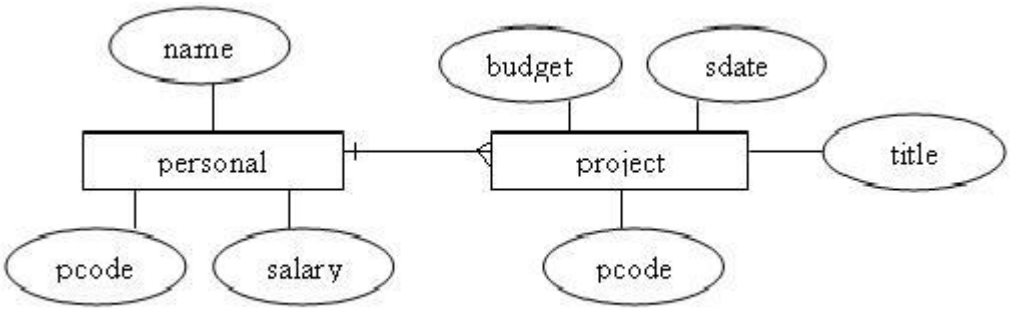
برای یک رابطه درجه ی ۲ و با کاردینالیتی 1:1 برای هر نهاد یک جدول ایجاد کرده و کلید اصلی آن دو جدول با هم متناظر خواهد بود.

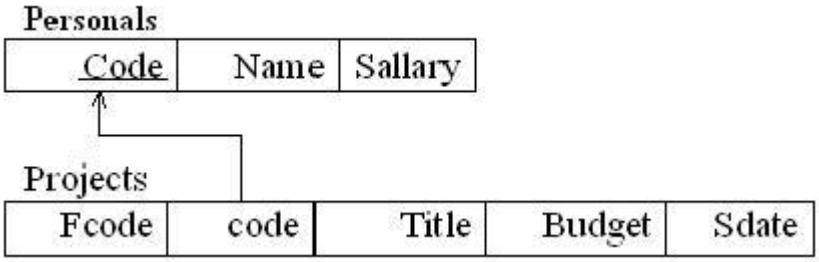




قاعده پنجم:

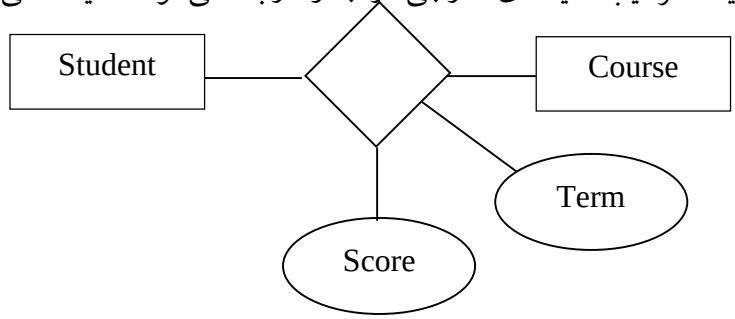
برای یک رابطه درجه۲ باکاردینالیتی1:N نهادطرف یک تبدیل به یک جدول می شود و نهاد طرف چند(N) تبدیل به یک جدول با کلید خارجی متناظر با کلید اصلی جدول دیگری می شود.به عنوان مثال برای نهادهای کارمند و پروژه ، به شرط اینکه بر روی هر پروژه فقط یک کارمند کار کند و هر کارمندی بتواند چند پروژه را انجام دهد، خواهیم داشت:



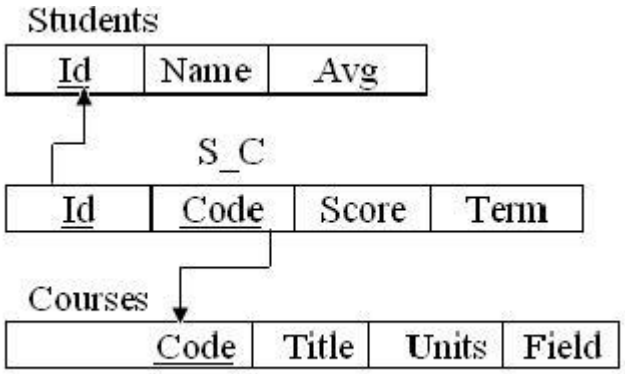


قاعده ششم:

در رابطه ی درجه ی ۲ با کاردینالیتی M:N (چند به چند) هر کدام از نهادها به صورت یک جدول پیاده سازی شده و خود رابطه نیز به صورت یک جدول جداگانه ایجاد می شود که دو کلید خارجی متناظر با کلیدهای اصلی در دو جدول نهاد دارد. ضمن اینکه ترکیب کلیدهای خارجی در جدول رابطه می تواند کلید اصلی آن باشد.



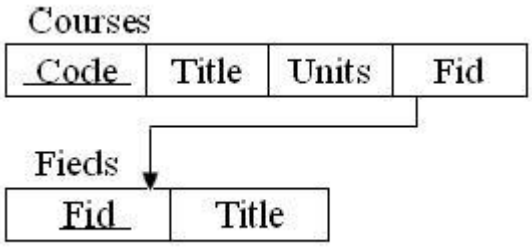
مثال: رابطه دانشجو با درس که هر دانشجو می تواند چند درس را انتخاب کرده و هر درس می تواند توسط چند دانشجو انتخاب شود.



قاعده هفتم:

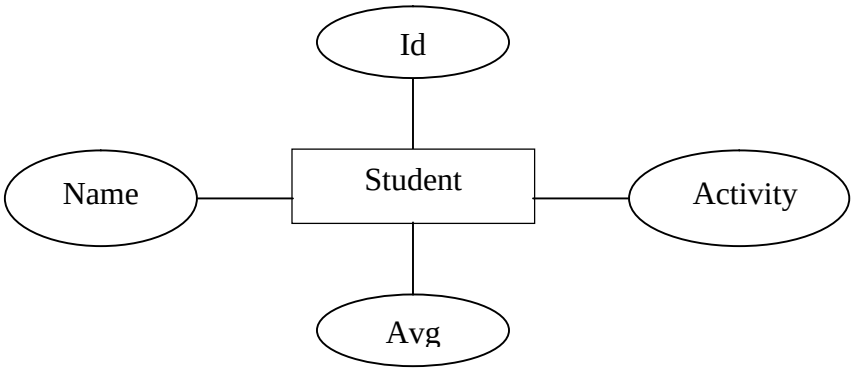
برای صفاتی که می توانند مقدار آنها از این چند مقدار معین انتخاب شوند.

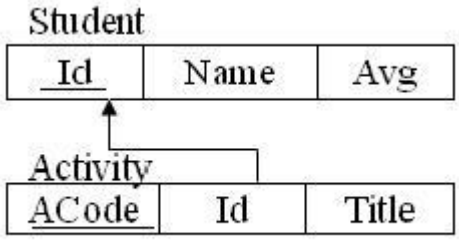
مثل رشته در جدول درس ها یا جدول دانشجو و یا استان محل سکونت در جدول دانشجویان و غیره. یک جدول برای نگهداری مقادیر قابل انتخاب ایجاد می کنیم مثلا جدول رشته ها ودر جدولی که دارای آن صفت است ،صفت به صورت کلید خارجی متناظر با کلید اصلی این جدول پیاده سازی می شود.



قاعده هشتم:

برای صفات چند مقداری مثل صفت فعالیت علمی دانشجو یا مدارک استاد یک جدول جداگانه ای برای نگهداری فعالیت ها و مقادیر آن صفت ایجاد می کنیم که دارای یک کلید خارجی مرتبط با کلید اصلی درجدول نهاد است.





مثال :جدول دانشجویان و فعالیت های آنها

Students

Id	Name	Avg
100	Ali	17
101	Sara	14

Activities

ACode	Id	Title
1	100	رتبه اول مسابقات ورزشي
2	101	رتبه اول مسابقات علمي
3	100	برگزاري سمينار آموزشي

قاعده نهم:

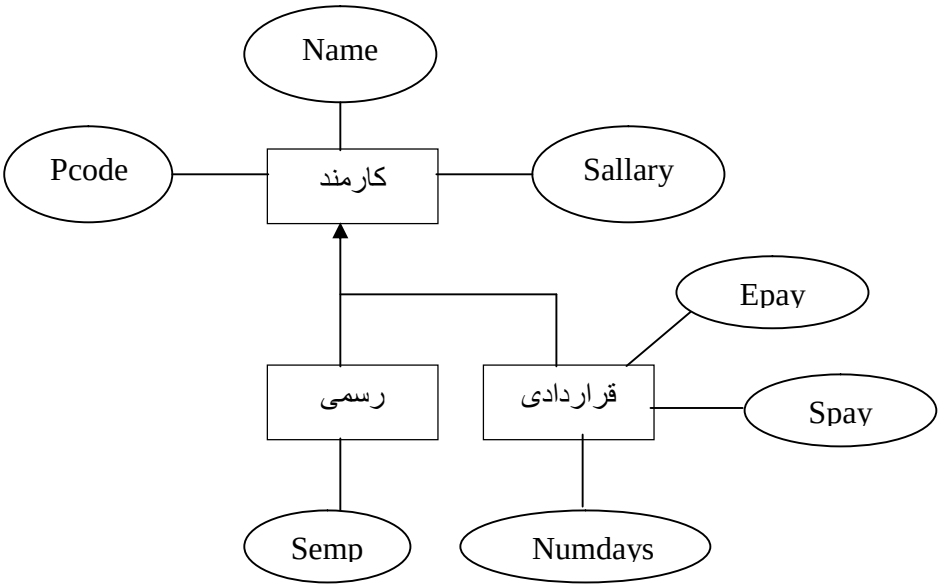
یک رابطه تخصیص به صورت دو جدول که جدول اصلی دارای اطلاعات نهاد اصلی(پایـه) و جـدول مـشتق دارای اطلاعات نهادهای خاص(مشتق) می باشد. در جدول مشتق یک ستون به عنوان کلید خارجی مرتبط با کلید اصلی در جدول پایه خواهد بود و بازای هر سطر در جدول مشتق باید یک سطر در جدول پایه وجود داشته باشد.

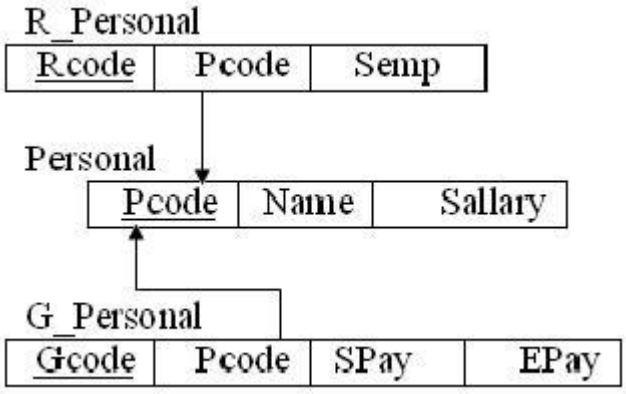
Spay: شروع قرارداد

Epay: پایان قرارداد

Semp: شروع استخدام

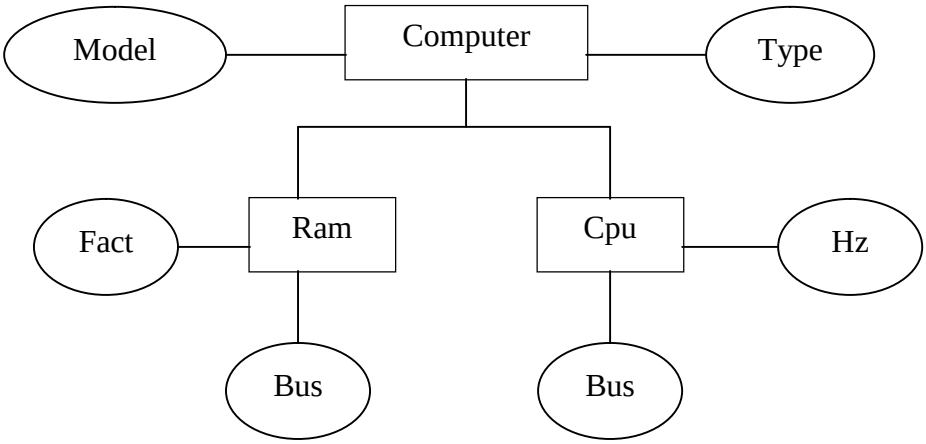
Numday: تعداد روزهای کارکرد

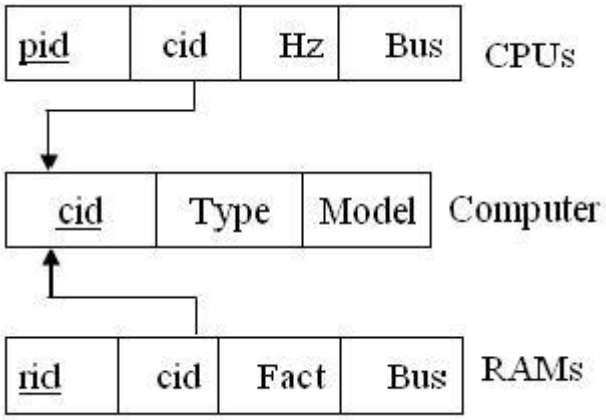




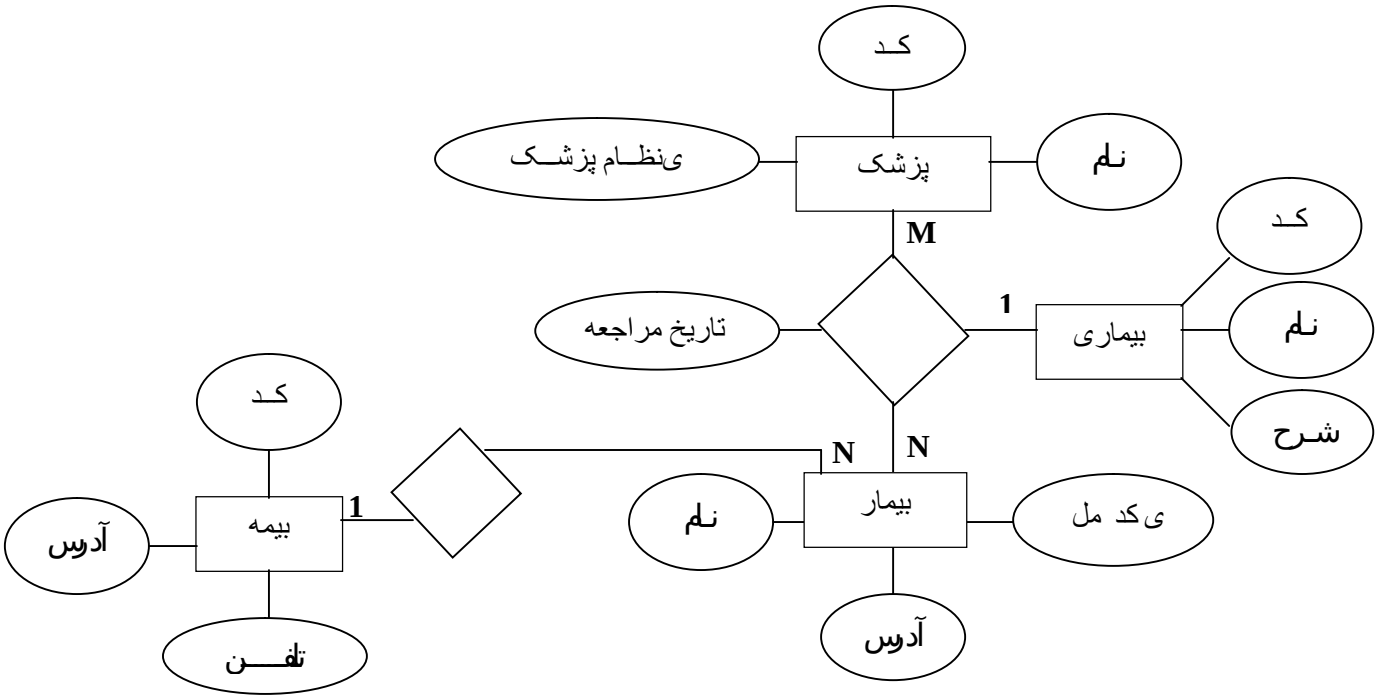
قاعده دهم :

رابطه تجمیع مثل رابطه بین کامپیوتر و قطعات آن. یک جدول ایجاد می کنیم که دارای ستونهایی برای صفات خاصه نهاد اصلی است و یک جدول برای هرکدام از نهادهایی جزئی که دارای ستونهایی برای نگهداری مشخصات آنها بعلاوه یک ستون بعنوان کلید خارجی مرتبط با کلید اصلی در جدول اصلی هستند.





مثال:نمودار زیر را به بانک اطلاعاتی تبدیل کنید.



جدول ها :

۱- جدول پزشک ها

Doctors

ردیف	نام ستون	نوع داده
۱	Dcode	عددی
۲	Nezam	عددی
۳	Name	کاراکتری

۲- جدول بیمه ها

Insurance

ردیف	نام ستون	نوع داده
۱	Icode	عددی
۲	Addr	متنی
۳	Tel	عددی

۳- جدول بیماران

Bimaran

ردیف	نام ستون	نوع داده
۱	Bcode	عددی
۲	Name	کاراکتری
۳	Addr	کاراکتری
۴	Icode	عددی

۴- جدول بیماریها

Bymary

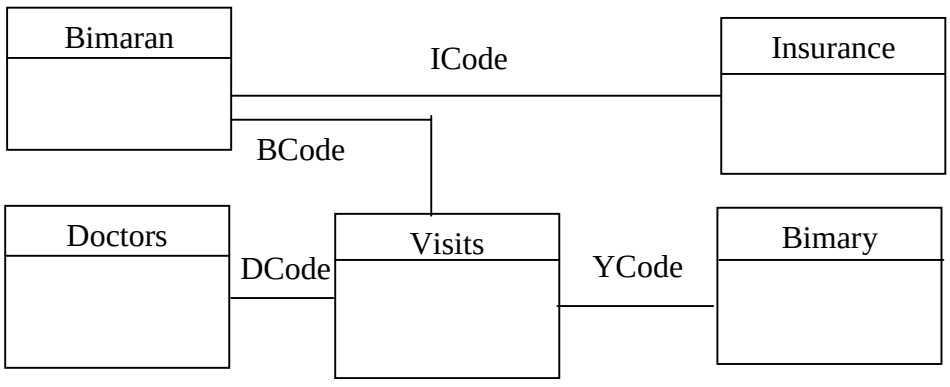
ردیف	نام ستون	نوع داده
۱	Ycode	عددی
۲	Name	کاراکتری
۳	Sharh	کاراکتری

۵- جدول رابطه – پزشک ، بیمار ، بیماری

Vizits

ردیف	ستون	نوع داده
۱	Dcode	عددی
۲	Bcode	عددی
۳	Ycode	عددی
۴	Vdate	تاریخ

رابطه بین جداول بصورت زیر است :



فصل پنجم

زبان پرس و جوی ساختیافته SQL

برای تعریف بانک اطلاعاتی،اجزای آن و دستکاری داده ها هر DBMS دستورات را به زبان خاصی دریافت می کند.به این زبانDSL(Data Sub Language) می گویند

دستورات DSL شامل دو بخش است:

۱) زبان تعریف داده ها (DDL) :شامل دستوراتی برای ساخت بانک اطلاعاتی ،جدول،دیدگاه ها و اصطلاح بانک اطلاعاتی و جداول و...به عبارتی این دستورات بخشی از بانک اطلاعاتی به نام شبه داده ها یا همان کاتالوگ را تعیین می کند.

۲) زبان دستکاری داده ها (DML) : شامل دستوراتی که داده ها را در باتک اطلاعاتی ذخیره کرده و آنها را دستکاری می کند.

۵-۱- زبان SQL

نام این زبان سرنام عبارت Structured Query Language یعنی زبان پرس و جوی ساختیافته است که یک زبان استاندارد برای کار با بانک اطلاعاتی است و توسط بیش تر DBMS ها شناسایی شده و قابل استفاده است. این زبان نسخه ها و لهجه های متفاوت دارد و هر DBMS ممکن است لهجه خاصی از SQL را استفاده کند.

برای اجرای دستورات SQL می توان از نرم افزاری جانبی DBMS مثلQuery Analyzer در Sql Server استفاده کرد و یا در محیط های برنامه نویسی با استفاده از امکانات زبان برنامه نویسی فرامینی به شکل SQL به DBMS فرستاد. زبان های برنامه نویسی بانک اطلاعاتی مثل VBو C#میتوانند مستقیما از برنامه نویس دستورات SQL گرفته و یا اینکه با فراهم کردن کامپوننت هایی دستورات را به شکل فراخوانی متدهای خاص گرفته و آن ها را به SQL تبدیل کند و نهایتا دستور SQL را برای اجرا به سیستم مدیریت بانک اطلاعاتی بفرستند.

۵-۲- دستورات DML

در SQL مهمترین این دستورات عبارتند از:

۱- Select بازیابی اطلاعات

۲- Insert درج سطر جدید

۳- Update ویرایش رکوردهای دلخواه

۴- Delet حذف رکوردهای دلخواه

۵-۲-۱- دستور Select

Select لیست ستون ها From نام جداول

Where محدودیت

لیست ستون ها می تواند ستاره (*) یعنی همه ی ستون ها باشد.

محدودیت می تواند هر شرط ساده یا شرط ترکیبی با استفاده از عملگر های and,or,not و... باشد.

عملگرهای مقایسه ای :

= مساوی <> نامساوی < بزرگتر

=> بزرگتر یا مساوی > کوچکتر =< کوچکتر یا مساوی

در ذیل مفاهیم و تکنیک هایی برای بازیابی اطلاعات مورد نیاز را در قالب مثال هایی ارائه می کنیم:

جدول Studs

Id	Name	Avg	Fid
100	Ali	17	1
101	Reza	18	1
102	Ahmad	15	2
103	Hadi	16	1

مثال ۱: بازیابی کل سطرهای وستون های جدول studs

```
Select * from studs
```

وقتی نمی خواهیم محدودیت ایجاد کنیم نیازی به نوشتن where نمی باشد.

مثال ۱: بازیابی شماره دانشجویی و نام همه دانشجویان (عملکرد تصویر)

```
Select id ,name from studs
```

در عمل تصویر نیازی به نوشتن where نیست ومحدودیت روی سطرها ایجاد نمی شود.

مثال۳: بازیابی دانشجویانی که کد رشته آنها 1 است. (عملکرد محدودیت)

```
Select * from studs where fid=1
```

خروجی :

Id	Name	Avg	Fid
100	Ali	17	1
101	Reza	18	1
103	Hadi	16	1

مثال ۴: بازیابی نام ومعدل دانشجویان رشته 1. (ترکیب عملکرد تصویر و محدودیت)

```
Select Name,Avg from studs where fid= 1
```

خروجی:

Name	Avg
Ali	17
Reza	18
Hadi	16

مثال۵: بازیابی نام ومعدل دانشجویان رشته 1 با معدل بالاتر از 17.

```
Select Name,Avg from studs
Where  fid=1 and avg>17;
```

در آخر دستورات بالا از(;) استفاده شود . در صورتی که یک دستور باشد نوشتن ;الزامی نیست.

مرتب سازی خروجی

برای مرتب کردن نتایج پرس و جو از بخش Order By به صورت زیر استفاده می شود:

select ستون ها from جدول where شرط

order by ستون 1, ستون 2,.... Asc|Desc

Asc : صعودی (پیش فرض) Desc: نزولی

مثال۶: لیست دانشجویان به صورت مرتب شده براساس نام نمایش دهید.

```
Select * from studs order by name
```

عملگر Like

برای مقایسه یک عبارت با بخشی از فیلد متنی از Like استفاده می شود:

مثال ۷: بازیابی لیست دانشجویانی که نام آنها شامل کلمه ali است.

```
Select  *  from studs where name like '%ali% '
```

% مشخص کننده هیچ یا چند کاراکتر است.

عملگر Between

برای بررسی قرارداشتن مقدار یک ستون در بین دو مقدار معین

مثال ۸: اسامی دانشجویانی را بازیابی کنید که معدل آنها بین 15 و 17 است.

```
Select name from studs where avg>=15 and avg<=17
```

یا

```
Select name from studs where avg between 15 and 17
```

عملگر In

بررسی می کند مقدار یک ستون در بین چند مقدار معین وجود دارد یا خیر

مثال ۹: اسامی دانشجویان رشته های 1و2و3 را نمایش دهید.

```
select name from studs where fid=1 or fid=2 or fid=3
```

یا

```
Select name from studs where fid in ( 1 , 2 , 3 );
```

جدول رشته ها بصورت زیر را درنظر بگیرید:

fields

fid	title
1	Computer
2	Graphics
3	Mechanics

مثال ۱۰: لیست رشته ها و دانشجویان هر رشته را نمایش دهید.(عملکرد الحاق)

```
Select fids.*,studs.* from  fids,studs
Where filds.fid=studs.fid
```

نتیجه این دستور بازیابی تمامی ستونها از دو جدول تعیین شده خواهد بود . و بازای هر سطر از جدول دانشجویان مشخصات رشته تحصیلی آن را نیز نمایش خواهد داد.

fid	title	id	name	avg	Fid
1	computer	100	Ali	17	1
1	computer	101	Reza	18	1
1	computer	103	Hadi	16	1
2	graphics	102	ahmad	15	2

برای الحاق یک شرط لازم است که ستون مشترک در دو جدول را با هم مساوی قرار دهد.

۵-۲-۲- پرس وجوهای تجمعی

در SQL دو نوع تابع وجود دارد :

۱- توابع اسکالر: روی یک ستون (بدون تعیین ستون) اجرا شده و روی یک مقدارعمل میکند.مثل توابع ریاضی،رشته ای،تاریخ و غیره

۲- توابع تجمعی: روی یک ستون از جدول اجرا شده و بر روی چند سطر عمل می کند.

توابع تجمعی عبارتند از

count : محاسبه تعداد سطرها sum : محاسبه مجموع مقادیر یک ستون

avg : محاسبه میانگین مقادیر یک ستون min، max : محاسبه حداکثر یاحداقل مقادیر یک ستون

مثال۱- تعداد سطر های جدولstuds را برمی گرداند.

```
Select count (*)  from  studs
```

خروجی:

Count(*)
4

جدول فروش رابصورت زیر درنظر بگیرید:

cp

cid	pid	qty
100	10	150
101	10	160
100	3	130
101	4	100

مثال۲- محاسبه ی مجموع مقادیر درجدول فروش

```
Select sum(qty) from cp
```

مثال۳- مجموع فروش کالاهای شماره ی 10 را پیدا کنید

```
Select sum(qty) From cp Where pid=10 ;
```

مثال۴- میانگین کل دانشجویان یک دانشگاه

```
Select Avg(avgr) from studs
```

مثال ۵- بالاترین معدل بین دانشجویان رشته کامپیوتروکد 2

```
Select max(avgr) from studs where fid=2
```

خروجی:

max(avgr)
18

جدول دانشجویان ، دروس و انتخاب درس را بصورت زیر در نظر بگیرید:

studs

Id	Name	Avg	Fid
100	Ali	17	2
101	Reza	18	1
102	Ahmad	15	2
103	Hadi	16	1

جدول courses

code	title	units
10	Db	2
11	Web	2
12	physics	3

Sabtenam

id	code	score
100	10	16
101	12	18
102	11	10
102	10	12

Code: کد درس id : شماره دانشجویی score: نمره

مثال۶- نمایش بالاترین معدل ها به تفکیک رشته

```
Select fid , Max (avgr)
From studs
Group by fid
```

Fid	Max(Avgr)
1	18
2	17

مثال ۷ : نامگذاری ستون ها بصورت دلخواه

```
Select fid , Max (avgr)  MG
From studs
Group by fid
```

Fid	MG
1	18
2	17

مثال۸- بالاترین معدل های بالای 15 را به تفکیک رشته نمایش دهید.

```
Select Fid , Max (avgr)
From studs
Group by Fid
Having Max (avgr) > 15;
```

*** از HAVING برای اعمال محدودیت روی نتیجه پرس و جوی تجمعی استفاده می شود**

مثال ۹- بالاترین معدل ها را به تفکیک رشته برای دانشجویان با شماره دانشجویی بزرگتر از 3 نمایش دهید.

```
Select Fid , Max (avgr)
From studs
Where id > 3
Group by Fid;
```

۵-۲-۳- نوشتن SELECT های تودرتو

بیشتر اوقات برای انجام الحاق به جای روش قبلی می توان از select های تودرتو استفاده کرد که باعث خوانایی بیشتر دستور و نیز افزایش سرعت اجرای آن می شود.

مثال ۱- لیست دروس انتخاب شده توسط ali را نمایش دهید.

```
Select sabtnam.*
From  sabtnam , studs
Where sabtnam .id = studs.id     and    studs . name= 'ali';
```

برای نوشتن پرس وجویی معدل فوق می توانیم بنویسیم:

```
Select  *
From sabtnam
Where id in (
                select id
                from studs
                where name = 'ali' ) )
```

مثال ۲- کد و عنوان دروس انتخاب شده توسط دانشجویان کامپیوتر باکد 2 را نمایش دهید.

```
Select code, title
From course
Where code in (select code
                  from sabtnam
                  where id in (select id
                                  from studs
                                  where Fid = 2)) ;
```

خروجی :

code	title
10	**DB**
11	**Web**

۵-۲-۳- دستور INSERT

```
Insert into  tablename(  column1, column2,… )
                Values ( value1,value2,…  );
```

مقادیر از نظر تعداد و نوع داده باید متناظر با ستونها باشند. اگر قصد مقداردهی تمام ستون های جدول را داریم لازم نیست نام ستونها ذکر شود و لازم است مقادیر به ترتیبی که ستونهای جدول هستند نوشته شوند.

```
Insert into  tablename      Values ( value1,value2,… );
```

با فرض اینکه id ازنوع identity (خود افزایشی)است. می توان برای آن مقدار صفر برای آن تعیین کرده، یا مقدار آنرا ذکر نکنیم:

```
Insert into studs  values( 0 , 'ali' , 17 );
Insert into studs( name , avgr )   values( 'reza' ,  17 ) ;
```

*** فیلدهای خود افزایشی در هنگام درج مقداری برای آنها تعیین شده یا با صفر مقداردهی می شوند.**

۵-۲-٤- دستورDELETE

حذف یک یا چند سطر از جدول

```
DELETE FROM tablename
WHERE   عبارت شرطی
```

مثال۱: حذف دانشجویی با شماره ی دانشجویی 10

```
Delete from studs where id=10
```

مثال۲- حذف دانشجویان با معدل کمتر از 10

```
Delete from studs where avgr>10
```

مثال۳- حذف دانشجویانی که هیچ درسی را انتخاب نکرده اند

```
Delete from studs
Where  not id in (  Select  id
                    from   sabtnam ) ;
```

۵-۲-۵- دستور UPDATE

بروز رسانی یک یا چند سطر از جدول

```
Update   نام جدول   set  1ستون=1عبارت ,2ستون=2عبارت ,.....
where    عبارت شرطی
```

مثال۱- اصلاح مشخصات دانشجویی با شماره ی دانشجویی 100

```
Update studs set  fid=2 , name='sara' , avgr=17.5
Where id=100
```

مثال۲- به معدل دانشجویان رشته 2 ، یک نمره اضافه کند

```
Update studs set avgr=avgr + 1
Where fid=2;
```

عبارت محاسباتی می تواند براساس مقادیرثابت،نام ستونها وبا کمک عملگرهای محاسباتی ساخته شود. و برای ساخت آن مشابه مثال های بخش select استفاده می شود.

۵-۳- دستورات DDL

دستورات دستوراتی هستند روی ساختار بانک اطلاعاتی عمل می کنند. و عبارتند از:

۵-۳-۱- دستور create

ایجاد اشیای بانک اطلاعاتی شامل جدول،دیدگاه،بانک اطلاعاتی، ایندکس و.....

ساخت بانک اطلاعاتی :

```
create database test
```
ایجاد بانک اطلاعاتی test با تنظیمات پیش فرض

```
Create Table
```
نام جدول (مشخصات ستون ها)

مثال: ایجاد جدول دانشجویان

```
Create table studs( id      int primary key,
                    name    varchar(50) ,
                    avgr    float ,
                    fid     int ) ;
```

قواعد جامعیت (constraint) مروبط به ستون های جدول :

۱- unique منحصر به فرد

۲-primary key کلید اصلی

۳- default تعداد پیش فرض

۴- check تعیین قانون برای بررسی

۵- not null نتواند مقدار هیچ بگیرد

```
Create table studs(id int primary key,
                   Name char(100) not nnll,
                   Avgr flaot default 0,
                   Fid int check fid<100 );
```

ساخت شاخص :

اگر قرار است بعدا بر اساس ستونهایی از جدول جستجو انجام شود بهتر است برای آنها شاخص ایجاد کنیم تا سرعت انجام جستجو بیشتر شود.

Create index نام شاخص on نام جدول(نام ستون ها)

مثال: ایجاد یک شاخص بر اساس ستون name برای جدول studs به صورت صعودی

Create index name-ind on studs(name) Asc

ساخت دیدگاه:

دیدگاه پنجره ای است بر روی پایگاه داده ها و می تواند شامل ستون ها یا سطرهایی از یک یا چند جدول باشد. دیدگاه ماشبه جدول عمل کرده و اغلب عملیات قابل اجرا روی جداول پایه ، روی جداول دیدگاه نیز اجرا می شوند.

Create view نام دیدگاه AS دستورپرس وجو ;

مثال: یک دیدگاه بر روی جدول دانشجویان ایجاد کنید که شامل نام و معدل دانشجویان رشته ۲ باشد.

Create view vstud_2 AS Select name,avg from studs where fid=2

٥-٣-٢- دستور DROP

برای حذف کردن - بانک اطلاعاتی - جدول - دیدگاه - شاخص استفاده می شود.

DROP database test;
DROP table studs;

٥-٣-٣- دستور ALTER

اصلاح ساختار - جدول - شاخص - دیدگاه

مثال۱: اضافه کردن ستون city با نوع داده ای کاراکتری جدول studs

ALTER table studs ADD city varchar(50) ;

مثال ۲- حذف ستون avgr از جدول studs

ALTER table studs Drop column avgr;

مثال ۳- تغییر نوع داده ستون avgr از float به int

Alter table studs Alter column avgr float int;

٥-٣-٤- دستورات دیگر

دستور Use : باز کردن بانک اطلاعاتی

Use نام بانک اطلاعاتی

دستور Truncate :خالی کردن جدول

Truncate نام جدول; تمام سطرهای جدول را حذف می کند

Truncate studs;

دستورBackup: پشتیبان گیری

دستور Restore: بازیابی پشتیبان

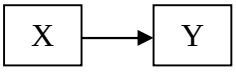
فصل ششم

نرمال سازی پایگاه داده

یکی از مهمترین فعالیت ها در طراحی بانک اطلاعاتی، نرمال سازی آنها است.در فصل های قبلی قواعدی برای تبدیل نمودار ER به بانک اطلاعاتی رابطه ای ارائه شد که هدف پیاده سازی یک بانک اطلاعاتی نرمال بود، ولی رعایت آن قواعد به تنهایی کافی نبوده و پس از ایجاد بانک اطلاعاتی باید بررسی های روی آن انجام شده واشکالات (آنامولی ها) آن برطرف گردیده و نرمال سازی صورت گیرد. برای درک نرمال سازی نیاز به آشنایی با وابستگی های تابعی می باشد.

٦-١- وابستگی تابعی

FD سرنام Functional Dependency است یعنی وابستگی تابعی. در یک جدول بین ستونها وابستگی تابعی وجود دارد، به عبارتی بعضی ستونها به بعضی دیگر وایسته هستند و یا بعضی ستونها ، ستونهایی دیگری را به صورت تابعی تعیین میکنند.



Y به طور تابعی به X وابسته است. با X تعیین کننده Y است.

مثال: جدول انتخاب واحد دانشجویان را در نظر بگیرید.

id	name	avgr	code	Title	score
100	**ali**	**17**	**1**	**DB**	**19**
101	**reza**	**18**	**2**	**Web**	**18**
100	**ali**	**17**	**2**	**Web**	**17**

در جدول انتخاب واحد وابستگی هایی تابعی از جمله موارد زیر را داریم:

{id} → { name , avgr}
{id,name} → { avgr }
{ code} → { title }
{id , code } → {mark}
{id , code , name , title }→ {mark , avgr }

بعضی از وابستگی ها کاهش ناپذیر هستند مثل وابستگی :

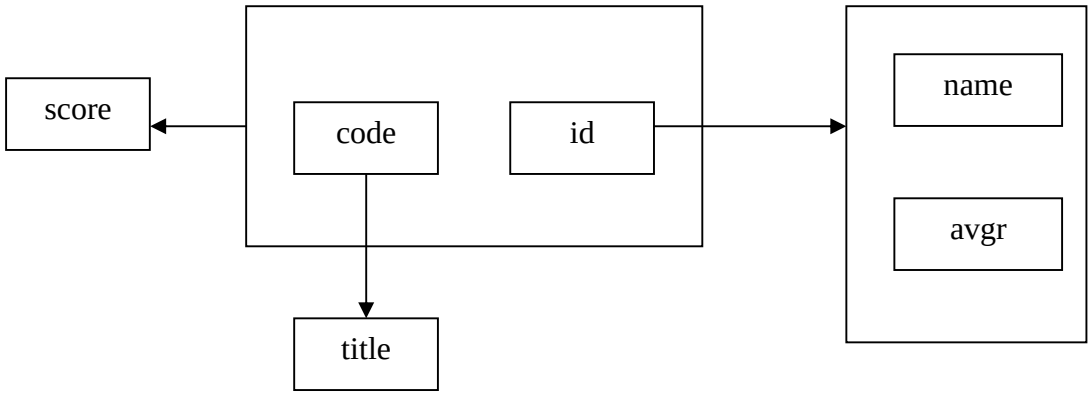
{id} → { name , avgr}

{ code} → { title }
{id , code } → {mark}

وابستگی تابعی کاهش ناپذیز:اگر در یک وابستگی تابعی نتوانیم ستون های سمت چپ را از آنچه که هست کمتر کنیم.

٦-۲- نمودار وابستگی تابعی

برای نمایش واضح تر وابستگی های تابعی کاهش ناپذیر از نمودار وابستگی تابعی (FDD) استفاده می شود. در این نمودار فیلدهابصورت جعبه ها و وابستگی ها با فلش نشان داده می شوند.



مثال: نمودار FD را برای جدول زیر رسم کنید.

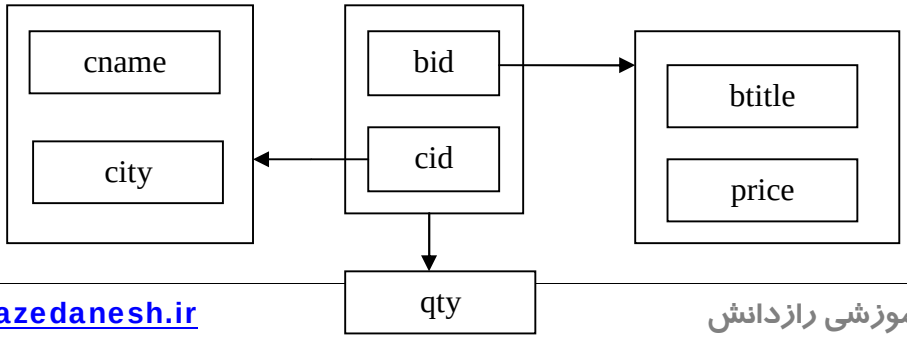
Sale(bid , btitle , price , cid , cname , city , qty);

bid:کد کتاب btitle:عنوان کتاب
price:قیمت واحد cid:کد مشتری
cname:نام مشتری city:شهر محل سکونت مشتری qty:تعداد

وابستگی ها تابعی عبارتند از:

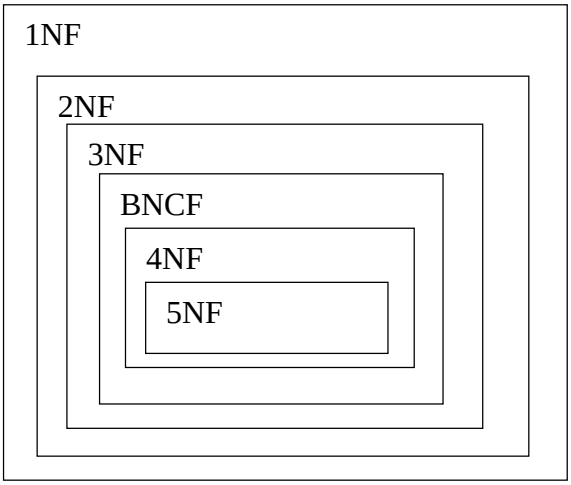
bid→ { btitle , price }
cid→ { cname, city }
{ bid , cid } → { qty }

و نمودار وابستگی تابعی آن بصورت زیر رسم می شود.



٦-٣- شکل های نرمال

بانک های اطلاعاتی از نظر نرمال بودن در یکی از شکل های زیر قرار می گیرند.



1-شکل نرمال اول(1NF)
2-شکل نرمال دوم(2NF)
3-شکل نرمال سوم(3NF)
4-شکل نرمال BC (BCNF)
5-شکل نرمال چهارم(4NF)
6-شکل نرمال پنجم(5NF)

هر بانک اطلاعاتی به شکل نرمال kNF به صورت های نرمال قبل از k نیز خواهد بود.
به عنوان مثال: یک بانک اطلاعاتی به شکل نرمال سوم به اشکال نرمال دوم و اول هم خواهد بود.

٦-٣-١- شکل نرمال اول(1NF)

اولین و ساده ترین سطح نرمال سازی است. اگر جدول بر اساس اصول پایگاه داده رابطه ای ایجاد شود و در هر چند گانه برای هر صفت تنها یک مقدار واحد را ذخیره کند به شکل نرمال اول است.

به عنوان مثال: اگر قصد داریم برای یک دانشجو نام استان،نام شهرو آدرس او را جداگانه ثبت کنیم باید برای هر کدام یک ستون مجزا تعریف شود تا بعدا بر اساس هر کدام از این ستون ها به طور جداگانه بتوانیم پرس و جوهای اجرا کنیم.
اگر طراح برای نگهداری استان شهر و آدرس به طور کلی یک ستون تعریف کند بعدا اجرای یک سری از پرس و جوهای وجود نخواهد داشت و جدول ها به شکل نرمال اول نخواهد بود.

مثال 1:جدول غیر نرمال

id	name	Address
100	Ali	Tehran , Karaj
100	Sara	Khorasan, Birjand

برای نرمال کردن جدول فوق دو ستون مجزا برای نگهداری شهر و استان در نظر می گیریم:

id	name	state	City
100	ali	Tehran	Karaj
100	sara	Tehran	Karaj

٦-٣-٢- شکل نرمال دوم

یک جدول به شکل نرمال دوم است اگر و فقط اگر به شکل نرمال اول بوده و هر صفت غیر کلیدی آن به صورت کاهش نا پذیر به کلید اصلی وابسته باشد.

مثال ١ :

در جدول ثبت نام داریم:

sabtname (id , name , avgr , code , title , score)

وابستگی های کاهش نا پذیر:

id → {name , avgr }
code → title
{ id , code }→ score

کلید اصلی id , code می باشد.

کلید اصل id وcode است در حالی که نام ومعدل به آن وابستگی کاهش ناپذیر تدارند بلکه به بخشی از آن (id) وابستگی دارند.همچنین title به بخشی از آن (code)وابستگی کاهش ناپذیر دارد. در نتیجه می گوییم جدول به شکل نرمال دوم نیست و به همین خاطر یک سری مشکلاتی برای درج،حذف یا اصلاح داده ها خواهد شد.

id	name	avgr	code	title	Score
100	Ali	18	1	Db	18
101	Reza	17	2	Web	17

100	Ali	18	2	web	16

اولین مشکل: افزونگی داده هاست یعنی اینکه اطلاعات تکراری مثل نام ومعدل دانشجو و عنوان درس در آن چندین بار تکرار میشود.

مشکل دوم: در هنگام درج پیش می آید این است که امکان ثبت مشخصات دانشجو وجود ندارد مگر اینکه درسی را انتخاب کرده باشد یعنی اینکه ثبت یک واقعیت وابسته شده به ثبت واقعیت دیگری است.

مشکل سوم: هنگام حذف پیش می آید که با حذف انتخاب واحد یک دانشجو مشخصات دانشجو هم حذف می شود.یعنی حذف یک واقعیت(انتخاب درس) واقعیتی دیگری(دانشجو را) حذف می کند.

مشکل چهارم: در هنگام بروز رسانی داده ها پیش می آید که برای تغییر نام یک دانشجو باید تمام سطرهای انتخاب واحد دانشجو تغییر کند یعنی نام دانشجو را در چند سطر تغییر دهیم که این کار هم زمان بر است هم ممکن است ناسازگاری داده هاپیش آید یعنی بعضی سطرها تغییر کند و بعضی تغییر نکند و در نتیجه یک واقعیت به چند شکل ثبت شده باشد.

برای نرمال کردن جدول: باید جدول شکسته شود(به دو یا چند جدول) و این تجزیه باید به صورت بدون نقصان باشد(تجزیه بدون نقصان)یعنی اینکه با الحاق بخش های جدول بتوان جدول اولیه را بدست آورد.

Sabtenam(id , name , avgr , code , title , score)

برای تبدیل به شکل نرمال دوم و درنتیجه رفع مشکلات فوق،با توجه به وابستگی :

Id→ { name , avgr }

جدول را به دو جدول زیر تجزیه می کنیم:

stud (id , name , avgr)
sabt2(id , code , title , score)

جدول sabt2 نیز با توجه به وابستگی :

code → { title }

به دو جدول زیر تجزیه می شود:

courses (code , title)
sabt3 (id , code , score)

که در جدول sabt3، فیلد id کلیدخارجی متناظر با کلید اصلی جدول studs و فیلد code کلید خارجی متناظر با کلید اصلی جدول courses است.

مثال ۲ :

در جدول فروش داریم که {bid , cid} کلید اصلی است:

sale(bid , btitle , price , cid , cname , city , qty)

وابستگی های کاهش ناپذیر عبارتند از:

{bid , cid } → { qty }
bid → { btitle , price }
cid → { cname , city }

چون bid به تنهایی تعیین کننده {btitle , price}است پس بصورت کاهش ناپذیر به کلید اصلی وابسته نیستند و جدول به شکل دوم نرمال نمی باشد.

برای تبدیل به شکل دوم نرمال، جدول sale را به دو جدول زیر تجزیه می کنیم:

books(bid , btitle , price)
sale2 (bid , cid , cname , qty)

و چون در جدول sale2 ستون cid به تنهایی نعیین کننده {cname,city}است پس slae2 را به دو جدول زیر تجزیه می کنیم :

customers (cid , cname , city)
sale3 (bid , cid , qty)

٦-٣-٣- شکل نرمال سوم

یک رابطه به شکل نرمال سوم است (3NF)اگر و فقط اگر به شکل نرمال دوم بوده و هر صفت غیر کلیدی به صورت غیر متعدی به کلید اصلی وابسته باشد به عبارت دیگر هر دو ستون از جدول به جز کلید اصلی از یکدیگر مستقل باشند.اگر جدولی به شکل نرمال سوم نباشد مشکلات قبلی (4 مشکل) را خواهد داشت.

مثال :جدول اطلعات کتابها و ناشرین آنها را درنظربگیرید:

pid: کد ناشر pname: نام ناشر pcity: شهر ناشر

bid: کد کتاب و کلید اصلی جدول

books(bid , btitle , price , pid , pname , pcity)

وابستگی های کاهش ناپذیر عبارتند از :

bid → { btitle , price , pname , pcity)
pid → { pname , pcity)

چون bid کلید اصلی است در حالی که pname و pcity به ستونی غیر از آن یعنی pid وابسته اند.در نتیجه به شکل نرمال سوم نیست.

برای تبدیل به شکل سوم نرمال، باید آن را به دو جدول زیر تجزیه کنیم:

books(bid , btitle , price , pid)
pubs(pid , pname , pcity)

که جدول books دارای یک کلید خارجی متناظر با کلید اصلی در جدول ناشرین خواهد بود.