پایگاه داده ها

Database - SQL

سیستم های مدیریت پایگاه داده

سیستم مدیریت پایگاه داده ها(database management system) به اختصار DBMS، یک نرم افزار است که با هدف مدیریت پایگاه دادهها طراحی شده است، به گونه ای که کاربر درگیر مسائل مربوط به ذخیره و بازیابی و شاخص بندی داده ها نمیشود و بر روی طراحی منطقی پایگاه تمرکز می نماید.این بسته به سازمان ها اجازه می دهد تا به راحتی پایگاه های داده را برای کاربردهای مختلف به وسیله ی مدیر پایگاه داده به اختصار (DBAs) و متخصصان دیگر توسعه دهد. سیستم مدیریت پایگاه داده اجازه می دهد تا برنامه های مختلف کاربردی کاربر به طور همزمان به یک پایگاه داده دسترسی داشته باشد

سیستم های مدیریت پایگاه داده

سیستم های مدیریت پایگاه داده ممکن است از مدل های مختلف پایگاه داده جهت سهولت در توصیف و حمایت از برنامه های کاربردی، استفاده کنند. یک سیستم مدیریت پایگاه داده امکان کنترل برای دسترسی به داده، اجرای تمامیت داده ها، مدیریت کنترل همزمانی و بازیابی پایگاه داده، پس از شکست و بازگرداندن آن از فایل های پشتیبان و همچنین حفظ امنیت پایگاه داده را فراهم می آورد.

سیستم های مدیریت پایگاه داده

۱. مایکروسافت اکسس(Microsoft Access) : یکی از اجزای مایکروسافت آفیس است که برای ایجاد پایگاه داده های رابطه ای مورد استفاده قرار میگیرد .این نرم افزار پایگاه داده جت را با یک واسط کاربری گرافیکی و ابزاری جهت تولید نرم افزار ترکیب نمود.

نسخه ی ۱,۰ این نرم افزار در سال ۱۹۹۲ میلادی همراه با مایکروسافت ویندوز پا به عرصه ی وجود نهاد، در این نسخه این امکان فراهم شد تا بسته های پایگاه داده ی جداگانه بتوانند از طریق تکنولوژی اتصال پایگاه داده ی شی گرا (ODBC) با یکدیگر ارتباط برقرار کنند.

نسخه ی ۲,۰ اکسس در سال ۱۹۹۴ وارد بازار شد. یکی از ویژگیهای مهم این نسخه افزوده شدن موتور پایگاه داده ی جت (jet database engine) بود که باعث شد اجرای پرسوجو ها به صورت محسوسی سریعتر شود. نسخه ی کنونی اکسس، نسخه ی ۲۰۱۳ میباشد.

سیستم های مدیریت پایگاه داده

۲. اوراکل : شرکت اوراکل (Oracle Corporation) یکی از بزرگترین شرکت های نرم افزاری در آمریکا و جهان است. این شرکت در سال ۱۹۷۷ میلادی با نام Relational Software Incorporatedیا RSIشروع به کار کرد. در ویرایش ۳ نرم افزار، نام شرکت از RSI به اوراکل تغییر کرد. محصول اصلی آن نرم افزارپایگاه داده های اوراکل است. این شرکت پر قدرتترین شرکت در زمینه ی سامانه مدیریت پایگاه داده ها است. در سال ۲۰۱۱ درآمد اوراکل از فروش محصولات و خدمات مختلف به رقم ۴۰.۲ میلیارد دلار رسید.

سیستم های مدیریت پایگاه داده

۳. **MySQL** : یک سامانه مدیریت پایگاه داده ها متن باز است، که توسط شرکت اوراکل توسعه، توزیع، و پشتیبانی میشود.سرور MySQL به چندین کاربر اجازه استفاده همزمان از داده ها را میدهد.

۴. **Database 2 - DB2**: به خانواده ای از محصولات شرکت IBMدر زمینه ی سیستم های مدیریت پایگاه داده ها اطلاق میشود.

سیستم های مدیریت پایگاه داده

۵. PostgreSQL : یک سامانه مدیریت پایگاه داده های - شی رابطه ای است که برای سکوهای مختلفی از جمله لینوکس، Free BSD، ویندوز، و مک اواس موجود است. PostgreSQL توسط گروه توسعه سراسری PostgreSQL توسعه داده میشود، که شامل تعداد زیادی از افراد داوطلب است. PostgreSQL بخش اعظم استاندارد SQL ۲۰۰۸ را پیاده سازی میکند و دارای افزونه های بسیاری است که توسط دیگران ایجاد شده است.

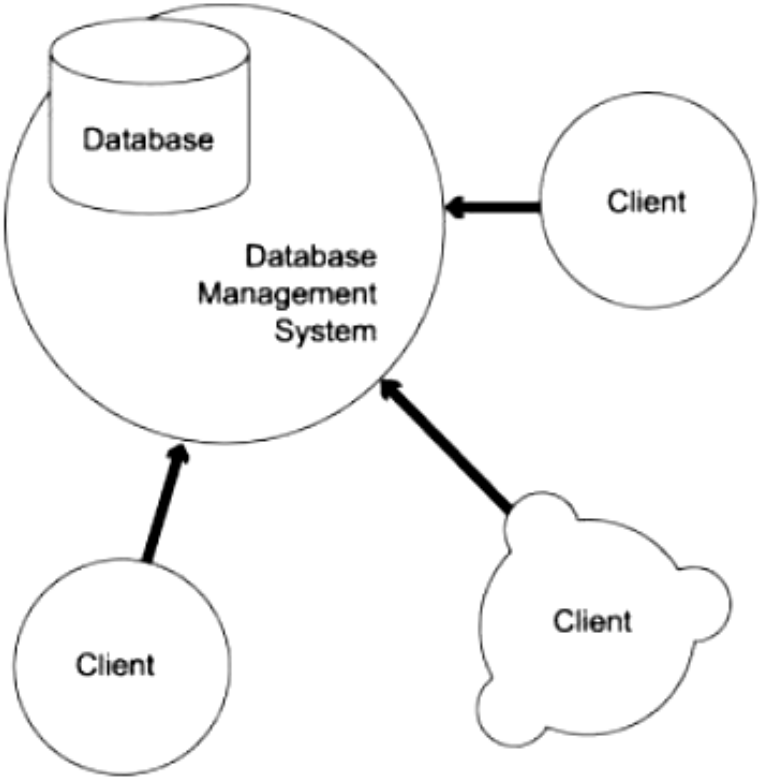
سیستم های مدیریت پایگاه داده

۶. **Microsoft SQL Server** :یک نرم افزار سیستم مدیریت بانک های اطلاعاتی است که توسط شرکت مایکروسافت توسعه داده میشود. برخی از ویژگی های این سیستم مدیریت پایگاه داده ها به این شرح است:

- بانک اطلاعاتی رابطه ای
- امکان از trigger, View, Stored procedure
- پشتیبانی از XML
- بسیار قدرتمند و بدون محدودیت حجم و تعداد رکورد
- پشتیبانی از FullText Searchبرای سرعت در بازیابی اطلاعات و استفاده از زبان طبیعی در جستجوها

سیستم های مدیریت پایگاه داده

جدیدترین نسخه ی سرور SQLسرور SQL2008میباشد در سال ۲۰۰۸ پیشنهاد و عرضه گردید. اهداف SQL Server 2008ایجاد و مدیریت دادهها به شکل هماهنگی، سازماندهی و محافظت به شکل اتوماتیک میباشد و به صورت client/serverعمل می کند



سیستم های مدیریت پایگاه داده

به همراه SQLسرور ، ابزارهای گرافیکی نیز وجود دارد از جمله :

- SQL Server Management Studio: (SSMS) محیطی برای انجام تمام عملیات روی بانک اطلاعاتی و اجرای دستورات SQL

- Configuration Manager: محیطی برای انجام پیکر بندی های مختلف، محیطی و ...

SSMS اجازه می دهد که همزمان به یک یا چند سرور SQLموجود در شبکه متصل شده و بانک های اطلاعاتی روی آنها را مدیریت کنید. در پنجره Object Explorerآن لیست سرورها و اجزای داخل آن ها نمایش داده می شود. از جمله پوشه های databasesکه حاوی بانک های اطلاعاتی است

سیستم های مدیریت پایگاه داده

هربانک اطلاعاتی دارای اشیائی است،ایجاد بانک اطلاعاتی برای برآورده کردن اهداف تجاری مستلزم ایجاد و کارکردن با آن اشیا است که به اختصار معمول ترین آنها را توضیح میدهیم:

- table: حاوی تمام اطلاعات موجود در بانک اطلاعاتی است.
- view: یک جدول مجازی است که محتویات آن توسط یک تقاضا مشخص میشود وممکن است از چند جدول به دست بیاید.
- stored proceduer: مجموعه ای از دستورات sql است که کاربر یک بار آنها را می نویسد و بارها و بارها آن را فراخوانی و اجرا می کند.

- roles :کاربرانی را با نیاز های دستیابی یکسان دسته بندی می کند.

- Rules :روش استانداری برای محدود کردن مقادیر در یک ستون هستند در واقع مقرراتی را به ستون ها اعمال می کند.

- Defaults :وقتی مقادیری برای ستون ها در نظر گرفته نمی شود پیش فرض ها را برای ستون ها در نظر میگیرد.

پایگاه داده ها

Database - SQL

پایگاه داده

فایل های اطلاعاتی که به نوعی به هم مرتبط هستند، تشکیل یک بانک اطلاعاتی را مـی دهنـد .فایـل شـامل مجموعـه ای از رکوردها می باشد و رکورد مجموعه ای از فیلدهای به هم مرتبط است و فیلد کوچکترین جزء یک بانک اطلاعـاتی مـی باشـد. مثلا در بانک اطلاعاتی دانشگاه چندین فایل وجود دارد مانند فایل دانشجویان که شامل چندین رکورد است، هر رکورد شـامل اطلاعات یک دانشجو می باشد که از چند فیلد مانند شماره دانشجویی، نام، آدرس، معدل و.... تشکیل شده است.

داده : نمایش پدیده هاومفاهیم به صورت صوری و مناسب برای برقراری ارتباط یا پردازش.

اطلاع: داده پردازش شده می باشد.

شناخت: نمایش نمادین جنبه هایی از بخشی از جهان واقع می باشد.به عبـارتی نـوعی اطـلاع سطح بالاتر است.

پایگاه داده: مجموعه ای از داده های ذخیره شده و پایا به صورت مجتمع و بهم مرتبط، با کمترین افزونگی،تحت مدیریت یک سیستم کنترل متمرکز، مورد استفاده یک یا چند کاربر به صورت همزمان و اشتراکی.

روش های ایجاد سیستم های کاربردی

۱- روش فایلینگ(ناپایگاهی)

۲- روش پایگاهی

۱- روش فایلینگ(ناپایگاهی)

در روش فایلینگ(سنتی)، نیازهای اطلاعاتی و پردازشی هر قسمت از محیط برآورده می شوند. مراحل اولیه طراحی و تولید برای هر قسمت به طور کلاسیک انجام شده و بعد از طراحی، مشخصات هر سیستم همراه با وظایف آنها مشخص می شود. در این روش، برای ایجاد محیط ذخیره سازی اطلاعات از یک سیستم فایل (FS) و برای برنامه سازی از یک زبان سطح بالا استفاده می شود و در نهایت برای هر قسمت، یک سیستم کاربردی ایجاد می شود.

روش های ایجاد سیستم های کاربردی

معایب روش فایلینگ

۱- عدم وجود محیط مجتمع ذخیره سازی

۲- عدم وجود سیستم کنترل متمرکز

۳- عدم وجود ضوابط ایمنی کارا

۴- عدم امکان اشتراکی شدن داده ها

۵- تکرار در ذخیره سازی اطلاعات

۶- مصرف نامناسب امکانات سخت افزاری و نرم افزاری

۷- وابسته بودن برنامه های کاربردی به محیط ذخیره سازی داده ها

۸- حجم زیاد برنامه سازی

روش های ایجاد سیستم های کاربردی

۲- روش پایگاهی

در این روش نیازهای اطلاعاتی تمامی قسمتها مورد مطالعه قرار می گیرد تا بتوان یک سیستم یکپارچه (integrated) طراحی کرد. داده های سازمان مدلسازی معنایی (SDM) می شوند و مشخصات سیستم یکپارچه تعیین می شود. برای سیستم مدیریت متمرکز از یک یا چند DBMS استفاده می شود. طراحی پایگاه داده ها در سطوح لازم انجام می شودو کاربران هر قسمت، پایگاه داده های خود را تعریف می کنند و با آن کار می کنند.

در واقع در روش پایگاهی یک محیط ذخیره سازی واحد، مجتمع واشتراکی، تحت کنترل متمرکز وجود دارد که کاربران براساس نیاز خاص خود، پایگاه خود را تعریف کرده و هر کاربر تصور می کند که پایگاه خود را دارد.

روش های ایجاد سیستم های کاربردی

- ✓ کاربران در روش پایگاهی بطور همزمان از سیستم استفاده می کنند.
- ✓ در روش پایگاهی نسبت به داده های ذخیره شده، تنوع و کثرت دید وجود دارد.
- ✓ در روش پایگاهی نسبت به روش فایلینگ، حجم برنامه ها کمتر و برنامه سازی آسانتر است.
- ✓ پایگاه داده ها بر حسب تعداد رکوردهای آن، به دسته های کوچک، متوسط، بزرگ (LDB) و خیلی بزرگ (VLDB) تقسیم می شوند.

نسل های ذخیره و بازیابی اطلاعات

پنج نسل ذخیره و بازیابی اطلاعات عبارتند از:

۱- نسل فایلهای ساده ترتیبی

۲- نسل AM

۳- نسل DMS

۴- نسل DBMS

۵- نسل KBS

نسل های ذخیره و بازیابی اطلاعات

۱- نسل فایلهای ساده ترتیبی

در این نسل از نوار به عنوان رسانه خارجی استفاده می شد. ساختار فایلها ترتیبی بود و ساختار فیزیکی و منطقی فایل یکسان بود. تنها روش پردازش فایلها، یکجا بود و نرم افزار واسطی برای مدیریت پردازش فایلها وجود نداشت امکان اشتراکی کردن داده ها وجود نداشت و تکرار در ذخیره سازی داده ها در بالاترین حد بود. هر نوع تغییر در ساختار داده ها یا رسانه ذخیره سازی موجب بروز تغییر در برنامه می شد و نسخه های متعددی از یک فایل نگهداری می شد.

۲- نسل AM

در این نسل یعنی شیوه های دستیابی (Access Methods)، نرم افزاری به نام شیوه های دستیابی ایجاد شد و برنامه کاربر را از پرداختن به جنبه های فیزیکی محیط ذخیره سازی مستقل کرد. رسانه این نسل دیسک بود و امکان دسترسی ترتیبی و مستقیم به رکوردها (نه فیلدها) وجود داشت و تا حدودی ساختار فیزیکی و منطقی فایلها از یکدیگر جدا شدند.

۳- نسل DMS

در این نسل یعنی نسل سیستم مدیریت داده ها ، نرم افزار کاملتری نسبت به نرم افزار دستیابی ایجاد شد و واسط بین برنامه های کاربردی و فایلهای محیط فیزیکی شد. در این نسل:

۱- از داده های اشتراکی در برنامه استفاده شد.

۲- آدرس دهی به داده ها در سطح فیلد ممکن شد.

۳- امکان بازیابی براساس چند کلید مهیاگشت.

۴- میزان افزونگی کاهش یافت.

۵- برنامه های کاربردی در قبال رشد فایلها مصون شدند.

۴- نسل DBMS

در نسل چهارم، برنامه های کاربردی از ویژگی های محیط فیزیکی ذخیره سازی مستقل شدند. کاهش افزونگی در ذخیره سازی داده ها، افزایش سرعت دستیابی به داده ها، بالا رفتن امنیت، امکان استفاده اشتراکی از داده ها و وجود یک محیط انتزاعی (Abstractive) از دیگر ویژگی های این نسل می باشد.

۵- نسل KBS

در این نسل یعنی نسل بانک معرفت(شناخت) به کمک مفاهیم هوش مصنوعی و سیستم های خبره، سیستمی طراحی شد که قادر به استنتاج منطقی از داده های ذخیره شده می باشد. در نسل پنجم سیستم بانک معرفت (KBS) ایجاد شد که مسئولیت ذخیره سازی شناخت، تامین جامعیت و امنیت بانک و تامین نیازهای کاربران را بر عهده دارد.

عناصر محیط پایگاه داده ها

عناصر چهارگانه محیط پایگاه داده ها عبارتند از:

۱- **نرم افزار**(DBMS- نرم افزار شبکه- برنامه های کاربردی - رویه های ذخیره شده)

۲- **سخت افزار** (ذخیره سازی- ارتباطی- پردازشگر)

۳- **کاربر**(موردی (نامنظم)- همیشگی (منظم))

۴- داده

عناصر محیط پایگاه داده ها

- ✓ به هر استفاده کننده از سیستم پایگاه داده ها، کاربر می گویند.
- ✓ کاربر منظم یا نامنظم خود بر دو نوع برنامه ساز و نابرنامه ساز تقسیم می شوند. کاربر نابرنامه ساز از طریق منو، هدایت می شود. کاربر برنامه ساز می تواند سیستمی یا کاربردی باشد.
- ✓ برنامه ساز کاربردی، برنامه های بهره برداری از پایگاه داده ها را می نویسد و برنامه ساز سیستم، برنامه های ایجاد و کنترل پایگاه داده ها را می نویسد.
- ✓ یک نوع کاربر نیز به نام کاربر پایانی (End-user) وجود دارد که به هر دو نوع کاربر برنامه ساز کاربردی و کاربر نابرنامه ساز گفته می شود.

ساختار داده ای

یک مدل داده ای شامل یک ساختار داده (DS) است. در واقع طراحی منطقی پایگاه داده ها به کمک مفاهیم اساسی یک مدل داده ای و در چار چوب ساختار داده ای آن مدل انجام می گیرد. ساختار داده ای امکانی است برای نمایش داده های موجودیت ها و انواع ارتباطات بین آنها.

عناصر تشکیل دهنده هر مدل

۱- ساختار داده ای

۲- امکانات عملیات در پایگاه داده ها

۳- امکانات کنترل جامعیت پایگاه داده ها

ساختار داده ای

انواع ساختارهای داده ای

مدل های داده ای بر پایه رکورد بر سه نوع می باشند:

۱- رابطه ای (RDS)

۲- سلسله مراتبی (HDS)

۳- شبکه ای (NDS)

مدلهای داده ای بر پایه شیء عبارتند از: (ER , Semantic , Functional)

مفهوم ساختار داده ای بخشی از مفهوم مدل داده ای است.

ساختار رابطه ای

ساختار رابطه ای (RDS) ، دارای ویژگی های زیر می باشد:

۱- مبنای تئوریک قوی دارد. (تامین کننده محیط انتزاعی به طور کامل)

۲- مسطح بودن محیط (مانند یک فایل ترتیبی ساده)

۳- دارای نمایش ساده از نظر کاربر

۴- دارای فقط یک عنصر ساختاری اساسی (جدول)

۵- امکان نمایش ارتباطات 1:1, 1:N, N:M

۶- ساده بودن منطق و دستور بازیابی

۷- دارای رویه پاسخگوی قرینه برای پرسشهای قرینه

ساختار سلسله مراتبی

ساختار سلسله مراتبی(HDS) ، قدیمی ترین ساختار داده ای برای طراحی منطقی پایگاه داده ها می باشد که دو عنصر اساسی دارد: نوع رکورد و نوع پیوند پدر– فرزندی (PCL).

بین هر دو رکورد پشت سرهم در یک مسیر درخت، پیوند پدر فرزندی وجود دارد که ارتباط 1:N را نمایش می دهد. در این ساختار، هر رکورد فرزند، تنها یک رکورد پدر دارد، یعنی در یک نوع PCL شرکت دارد.

ویژگی های ساختار سلسله مراتبی عبارتند از:

۱- مبنای ریاضی ندارد.

۲- مناسب برای ارتباط 1 : N .

۳- سادگی نمایش ساختار رابطه ای را ندارد.

۴- نمایش ارتباط N : M در آن مشکل است.

ساختار سلسله مراتبی

۵- جستجو حتماً باید از ریشه انجام شود.

۶- نداشتن تقارن ساختار جدولی.

۷- داشتن تعدادی محدودیت جامعیت.

۸- مشکل بودن دستور بازیابی در آن نسبت به ساختار جدولی.

۹- نمایش ارتباط با درجه بیشتر از دو.

پایگاه داده سلسله مراتبی، مجموعه ای منظم از نمونه های یک یا چند نوع سلسله مراتب می باشد.

در HDS ، برای نمایش ارتباط چند به چند بین دو نوع موجودیت، می توان:

۱- دو نوع سلسله مراتب جدا طراحی کرد.

۲- دو نوع سلسله مراتب به هم مرتب طراحی کرد.

۳- یک نوع سلسله مراتب طراحی کرد.

ساختار شبکه ای

ساختار شبکه ای (پلکس) توسط گروه DBGT پیشنهاد شد و اولین سیستم مدیریت پایگاه داده ای شبکه ای IDMS نام دارد. این سیستم گاهی به نام کوداسیل نامیده می شود. شبکه، نوعی گراف جهت دار است. در ساختار شبکه هر گره فرزند می تواند بیش از یک گره پدر داشته باشد، بنابراین گسترش یافته ساختار سلسله مراتبی است.

ساختار شبکه ای از دو عنصر تشکیل شده است:

۱- نوع رکورد

۲- نوع مجموعه (مجموعه کوداسیلی)

ساختار شبکه ای

مجموعه کوداسیلی از سه جزء تشکیل شده است:

۱- نام مجموعه

۲- یک نوع رکورد مالک

۳- یک نوع رکورد عضو

درج مالک، بدون عضو، ممکن نمی باشد و درج یک عضو بدون وجود مالک ممکن نمی باشد.

با حذف مالک، اعضای آن نیز حذف می شوند.

ویژگی های ساختار شبکه ای

۱- تقارن دارد.

۲- مبنای ریاضی ندارد.

۳- سادگی ساختار رابطه ای را ندارد.

۴- عدم آنومالی در عملیات ذخیره سازی.

۵- فقط مخصوص نمایش ارتباطات 1:N نیست و هر نوع ارتباطی را می تواند نمایش دهد.

۶- امکان ناسازگاری داده ها در آن کمتر از سلسله مراتبی است.

۷- قواعد جامعیت ذاتی دارد.

۸- فزونکاری (به علت ایجاد و اصلاح اشاره گرها) دارد.

۹- دستور بازیابی پیچیده تری دارد.

۱۰- رویه جستجوی رکورد در آن نسبت به ساختارهای دیگر پیچیده تر است.

۱۱- اصل وحدت عملگر در یک عمل واحد مانند درج رعایت نمی شود.

ساختار شبکه ای

در NDS ، نوع رکورد مالک یک نوع مجموعه، می تواند عضو یا مالک در نوع مجموعه دیگر باشد.

در NDS اگر n نوع موجودیت و احیانا دارای صفات چند مقداری، در یک نوع ارتباط شرکت داشته باشند، حداقل n مجموعه کوداسیلی برای طراحی پایگاه داده شبکه ای (NDB) لازم است.

در نمونه مجموعه مدل شبکه ای بر خلاف مجموعه ریاضی:

۱- حداقل یک عنصر وجود دارد.

۲- یک عنصر وجود دارد که از نظر نوع با عناصر دیگر فرق دارد.

۳- بین بعضی از عناصر نوعی نظم وجود دارد.

ساختار شبکه ای

- ✓ اگر n نوع موجودیت در یک نوع ارتباط شرکت داشته باشند و این نوع ارتباط a صفت داشته باشد، رکورد پیوند دهنده در NDS ، دارای n+a فیلد در سطح پیاده سازی است.
- ✓ در NDS ، صفات نوع ارتباط R با چندی 1:N بین دو نوع موجودیت E و F با فیلدهایی در نوع رکورد **عضو** نمایش داده می شوند.
- ✓ فقط در RDS ، مفهوم نظم مطرح نیست.
- ✓ ساختار سلسله مراتبی تقارن ندارد، اما ساختار شبکه ای دارای تقارن است.
- ✓ تعداد عناصر ساختاری اساسی در RDS برابر 1 و در HDS و NDS برابر 2 می باشد.
- ✓ فقط در RDS ، جستجو در پایگاه داده ها به صورت اتوماتیک(خودکار) است.
- ✓ با توجه به سه نوع ساختار رابطه ای، سلسله مراتبی و شبکه ای، سه دسته DBMS به ترتیب زمان پیدایش آنها به نام های HDBMS , NDBMS و RDBMS وجود دارد. بعد از آنها OODBMS است.

معماری پایگاه داده ها

معماری استاندارد پایگاه داده ها که توسط ANSI پیشنهاد شد، یک معماری سه سطحی می باشد. این سطوح عبارتند از:

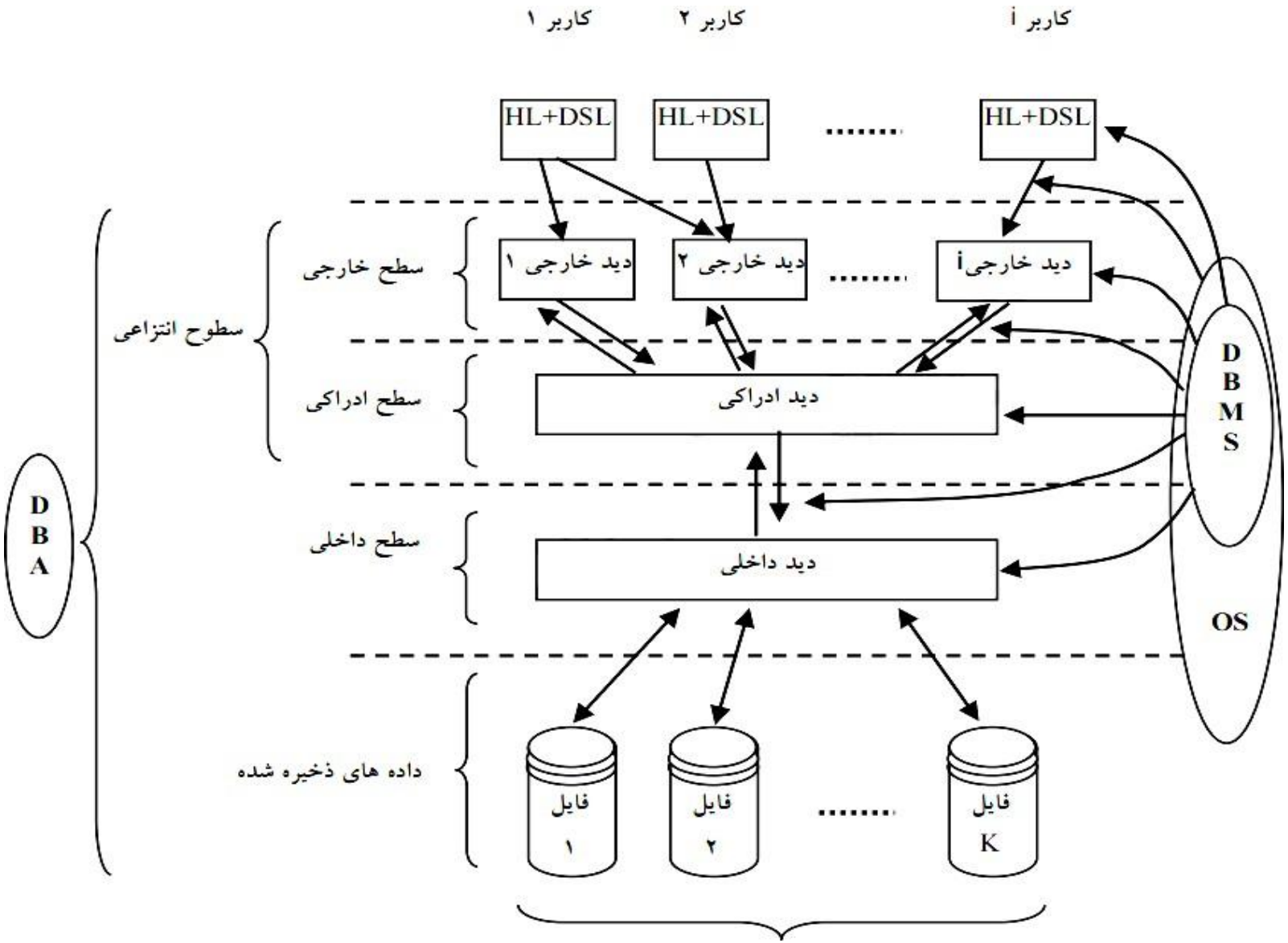
۱- سطح خارجی (External Level)

۲- سطح ادراکی یا مفهومی (Conceptual Level)

۳- سطح داخلی (Internal Level)

تذکر: به دو سطح خارجی و ادراکی، سطوح انتزاعی می گویند.

معماری پایگاه داده ها



اجزاء معماری پایگاه داده ها

۱- دید ادارکی (Conceptual View)

دید، پنجره ای است که از آن کاربر می تواند محدوده پایگاه خود را ببیند و خارج از این محدوده، چیزی نمی بیند. دید ادارکی، دید طراح نسبت به داده های ذخیره شده در پایگاه است که در برگیرنده تمام نیازهای کاربران در محیط عملیاتی می باشد. دید ادراکی چون در یک محیط انتزاعی مطرح است، بر یک ساختار داده ای مشخص بنا شده است.

شمای ادراکی:

به توصیف و شرح دید ادارکی، شمای ادراکی می گویند. در واقع شمای ادراکی برنامه ای است شامل دستورات تعریف داده ها و کنترل داده ها (نه دستورات عملیات بر روی داده ها).

اجزاء معماری پایگاه داده ها

۲- دید خارجی (External View)

دید خارجی، دید کاربر نسبت به داده های ذخیره شده در پایگاه داده ها می باشد. این دید:

۱- یک دید جامع نمی باشد.

۲- روی دید ادراکی طراحی و تعریف می شود.

۳- در سطح انتزاعی مطرح است.

۴- مبتنی بر همان ساختار داده ای می باشد که دید ادراکی بر اساس آن طراحی می شود.

شمای خارجی:

به وصف یا تعریف دید خارجی، شمای خارجی می گویند. در واقع شمای خارجی برنامه ای است حاوی دستورات تعریف داده ها و گاه کنترل داده ها.

اجزاء معماری پایگاه داده ها

۳- دید داخلی (Internal View)

دید DBMS و طراح پایگاه داده ها نسبت به داده های ذخیره شده را دید داخلی می گویند.

شمای داخلی:

به تعریف دید داخلی، شمای داخلی می گویند. در واقع نوعی برنامه است که توسط خود DBMS تولید می شود. انواع رکوردها در شمای داخلی تعریف می شوند و شامل دستورهای لازم جهت ایجاد فایلها و کنترل آنها می باشد.

۴-کاربر

هر استفاده کننده از پایگاه داده ها را کاربر می نامند. کاربر می تواند موردی یا همیشگی باشد.

اجزاء معماری پایگاه داده ها

۵- زبان میزبان (HL : Host Language)

زبان میزبان می تواند یکی از زبانهای برنامه سازی چون پاسکال ، C ، ایدا ، ... باشد.

۶- زبان داده ای فرعی (DSL : Data Sub Language)

دستورهای این زبان به سه قسمت تقسیم می شود:

۱- دستورات تعریف داده ها (DDL)

۲- دستورات کنترل داده ها (DCL)

۳- دستورات عملیات روی داده ها (DML)

سیستم مدیریت پایگاه داده DBMS

این سیستم یکی از نرم افزارهای واسط بین محیط فیزیکی ذخیره و بازیابی و محیط منطقی برنامه سازی مـی باشـد. DBMS به برنامه ساز امکان می دهد تا پایگاه داده های خود را تعریف کرده و در آن عملیات خود را انجام دهد.

عوامل موثر در دسته بندی DBMS ها عبارتند از:

قیمت، نوع کاربرد، سیستم فایل ، نوع و ماهیت DSL، نوع معماری، نوع ساختار داده‌ای، محیط سخت افزاری و محـیط سیستم عامل.

سیستم مدیریت پایگاه داده DBMS

نرم افزار DBMS از نمای بیرونی از دو واحد تشکیل شده است:

۱- واحد پردازشگر پرسش‌ها و برنامه‌های کاربردی

۲- واحد ایجاد و مدیریت داده‌های ذخیره شده

دلایل ایجاد سربار(فزونکاری) در DBMS عبارتند از:

۱- لزوم اعمال قواعد جامعیت

۲- لزوم انجام تبدیلات بین سطوح

۳- لزوم اعمال ضوابط ایمنی

مدیریت پایگاه داده DBA

مدیر پایگاه داده ها فردی است متخصص در پایگاه داده ها، با مسئولیت علمی و فنی که همراه با یک تیم تخصصی کار می کند.

وظایف تیم DBA عبارت است از:

الف – مشارکت در :

۱- تفهیم نقش داده به مدیریت سازمان

۲- تفهیم مزایای تکنولوژی پایگاه داده ها

۳- تصمیم گیری در مورد استفاده یا عدم استفاده از تکنولوژی پایگاه داده ها

۴- انتخاب DBMS و پیکربندی سخت افزاری و نرم افزاری لازم

مدیریت پایگاه داده DBA

ب – تصمیم گیری در مورد:

۱- تعیین معماری سیستم پایگاه داده ها

۲- انتخاب و انتساب اعضاء تیمهای اجرایی

۳- زبان برنامه سازی مورد نیاز

۴- مشخصات ساختار سطح داخلی پایگاه داده ها و تعیین ساختار فایلهای مناسب

۵- نوشتن شمای داخلی (طراحی فیزیکی)

۶- چگونگی رشد دادن پایگاه داده ها

۷- چگونگی سازماندهی مجدد پایگاه داده ها

۸- چگونگی ترمیم پایگاه داده ها

مدیریت پایگاه داده DBA

ج – طراحی :

۱- سطح ادراکی پایگاه داده ها (طراحی منطقی)

۲- واسطهای کاربری

۳- برنامه های کاربردی

د– نظارت بر :

۱- تعیین دیدهای خارجی و نوشتن شماهای خارجی

۲- وارد کردن داده ها

۳- تهیه مستندات لازم

۴- عملیات انجام شونده در پایگاه داده ها

استقلال داده ای

وابسته نبودن برنامه های کاربردی به داده های ذخیره شده را استقلال داده ای می نامند که مهمترین اهداف تکنولوژی پایگاه داده ها می باشد. به عبارتی استقلال داده ای عبارت است از تاثیر ناپذیری برنامه های کاربردی در سطح خارجی در قبال رشد پایگاه داده ها و تغییر در ساختار داده های عملیاتی است.

۱- استقلال داده ای فیزیکی

مصونیت دیدهای کاربران و برنامه های کاربردی در قبال تغییرات در سطح داخلی – فیزیکی.

۲- استقلال داده ای منطقی

مصونیت دیدهای کاربران و برنامه های کاربردی در قبال تغییرات در سطح ادراکی.

پایگاه داده

در DBMS های جدید،استقلال داده ای فیزیکی کاملا تامین است، ولی استقلال داده ای منطقی کاملا تامین نیست.

تغییر در سطح ادراکی یعنی تغییر در طراحی منطقی پایگاه داده ها و تغییر در شمای ادراکی، که این تغییر دارای دو وجه است: رشد پایگاه در سطح ادراکی و سازماندهی مجدد پایگاه در سطح ادراکی.

از مزایای مهم تکنولوژی پایگاه داده ها،استقلال داده ای است که لازمه تامین آن، معتبر ماندن شمای خارجی پس از تغییرات در شمای ادراکی و شمای داخلی است.

کاتالوگ سیستم

کاتالوگ سیستم حاوی داده هایی است در مورد داده های ذخیره شده در پایگاه داده های کاربر. به این داده های ذخیره شده، متا داده می گویند.

پایگاه داده

محتویات کاتالوگ در سیستم های مختلف یکسان نیست ولی بطور کلی شامل اطلاعات زیر می باشد:

۱- شماهای خارجی، ادراکی، داخلی

۲- ضوابط کنترل ایمنی داده ها

۳- مشخصات پیکربندی سخت افزاری سیستم

۴- شرح سازمان فیزیکی داده های ذخیره شده

۵- مشخصات کاربران و حقوق دستیابی آنها به داده ها

۶- مشخصات برنامه های کاربردی

۷- مشخصات پایانه های متصل به سیستم

۸- قواعد جامعیت

۹- ارتباط بین برنامه های کاربردی و داده های ذخیره شده

۱۰- توابع تعریف شده توسط کاربران

تراکنش (TRANSACTION)

تراکنش به برنامه ای گفته می شود که یک کاربر در محیط بانک اطلاعاتی اجرا می کند. پایان یک تراکنش یا موفق (commit) است و یا ناموفق (abort).

DBMS بر روی هر تراکنش کنترل هایی را انجام می دهد تا جامعیت بانک اطلاعاتی تضمین شود. این کنترلها به ACID معروف می باشند که به ترتیب معرف Atomicity و Consistency و Isolation و Durability می باشند.

۱- یکپارچگی (Atomicity)

به این معنی است که یا تمام دستورات یک تراکنش انجام می شود یا هیچکدام از دستورات اجرا نمی شوند. این خاصیت به همه یا هیچ موسوم است.(مثلا تراکنش انتقال مبلغی از یک حساب به حساب دیگر)

۲- همخوانی (Consistency)

یعنی هر تراکنش اگر به تنهایی اجرا شود بانک را از حالتی صحیح به حالتی صحیح دیگر منتقل می کند.

۳- انزوا (Isolation)

یعنی اثر تراکنش های همروند روی یکدیگر چنان باشد که ظاهرا هر کدام به طور مجزا و در انزوا انجام می شوند.

۴- پایائی (Durability)

به این معنی است که اثر تراکنش هایی که به مرحله انجام (commit) می رسند ماندنی است و به طور تصادفی از بین نمی رود. مثلا در تراکنش انتقال پول از حسابی به حساب دیگر ، بعد از واریز مبلغ تحت هیچ شرایطی (همچون آتش سوزی) اثر عمل انجام شده از بین نمی رود.

تراکنش های هم روند

تراکنش ها می توانند اصل سریالیتی را رعایت نکنند و به طور هم روند اجرا شوند. به طـور نمونـه دو تـراکنش A و B را در نظر بگیرید که A دو عمل a1 و a2 و B دو عمل b1 و b2 را انجام می دهند. این اعمال در اجرای هم روند می توانند به ترتیب زیر انجام گیرند:

زمان	تراکنش A	تراکنش B
t_1	a1	-
t_2	-	b1
t_3	a2	-
t_4		b2

معماری سیستم های پایگاه داده

۱-معماری متمرکز

یک پایگاه داده‌ها روی یک سیستم کامپیوتری ایجاد می‌شود و به سیستم کامپیوتری دیگری ارتباط ندارد.

۲- معماری مشتری خدمتگزار

در معماری مشتری خدمتگزار (Client-Server) قسمتی از پردازش توسط یک ماشین و قسمتی دیگر توسط ماشین دیگر انجام می‌شود. یعنی مسئولیت ها بطور منطقی تقسیم شده است. داده‌ها در سایت client ذخیره می‌شوند و برنامه‌های کاربردی در سایت server اجرا می‌شوند.

معماری سیستم های پایگاه داده

انواع معماری مشتری خدمتگزار(C/S DB) :

الف- معماری چند مشتری/یک خدمتگزار (MC/S DB)

ب- معماری یک مشتری/چند خدمتگزار(C/MS DB)

ج- معماری چند مشتری/چند خدمتگزار(MC/MS DB)

مزایای معماری مشتری- خدمتگزار نسبت به معماری متمرکز

الف- تقسیم پردازش

ب- اشتراک داده‌ها

ج- کاهش ترافیک شبکه

د- استقلال ایستگاههای کاری

معماری توزیع شده

این معماری از ترکیب دو تکنولوژی پایگاه داده‌ها و شبکه معماری توزیع شده حاصل می‌شود. پایگاه داده‌های توزیع شده (DDB) یعنی مجموعه‌ای از چند پایگاه دادهٔ بهم مرتبط و توزیع شده روی یک شبکه که از نظر کاربران، پایگاه واحدی است.

مزایای معماری توزیع شده	معایب معماری توزیع شده
۱- اشتراک داده‌ها	۱- هزینه بالا
۲- کاهش هزینه ارتباطات	۲- مصرف حافظه بیشتر
۳- دستیابی بهتر به داده‌ها	۳- پیچیدگی پیاده سازی
۴- تسهیل گسترش سیستم	۴- پیچیدگی طراحی سیستم
۵- استفاده از پایگاه داده‌های از قبل موجود	۵ – وجود تهدیدهای بالقوه برای امنیت سیستم
۶- سازگاری با سازمانهای جدید	

معماری توزیع شده

۴-معماری با پردازش موازی

نوع گسترش یافته معماری توزیع شده است که برای دستیابی پذیری بالا طراحی می‌شود. سیستم‌های موازی قادر به اجرای موازی تعداد زیادی تراکنش در ثانیه می‌باشند.

مدل های ممکن برای معماری موازی

الف- معماری با حافظه اصلی مشترک

ب- معماری با دیسک مشترک

ج- معماری بی اجزا مشترک

مزایای معماری موازی با دیسک های مشترک

الف- کاهش مصرف حافظه جانبی

ب- تسهیل گسترش سیستم

ج- تسهیل تحمل خرابی

فصل دوم

مدل ER

مفاهیم اساسی در مدل ER

نوع موجودیت (Entity type)

به مفهوم کلی شیء، نوع موجودیت می گویند. در واقع هر چیزی که بخواهیم در مورد ان اطلاع داشته باشیم اعم از اینکه وجود فیزیکی یا ذهنی داشته باشد. بطور نمونه انواع موجودیت ها در محیط عملیاتی دانشکده عبارتند از: دانشجو، درس، استاد، کارمند، گروه آموزشی و ... .

انواع موجودیت

۱- موجودیت قوی (مستقل)

موجودیتی که مستقل از هر نوع موجودیت دیگر، در یک محیط مشخص مطرح باشد. مانند موجودیت درس در محیط عملیاتی دانشگاه.

۲- موجودیت ضعیف (وابسته)

موجودیتی که وجودش وابسته به یک نوع موجودیت دیگر است، اما شناسه ندارد. مانند موجودیت عضو خانواده برای موجودیت کارمند یا موجودیت اثر منتشره برای موجودیت استاد.

صفت

ویژگی یک نوع موجودیت را صفت موجودیت می گویند. هر موجودیت دارای مجموعه ای از صفات است.

تقسیم بندی صفات

۱- ساده (Single) یا مرکب (Composition)

صفتی که مقدار آن از لحاظ معنایی اتومیک یا تجزیه نشدنی باشد را صفت ساده می گویند(مانند صفت درس) و صفتی که از چند صفت ساده تشکیل شده باشد را مرکب می گویند.

۲- تک مقداری یا چند مقداری

صفت تک مقداری صفتی است که به ازای یک نام صفت، حداکثر یک مقدار برای یک نمونه از موجودیت، داشته باشیم. مانند شماره دانشجویی که نمی تواند بیش از یک مقدار داشته باشد و اگر بتوان به ازاء یک نام صفت، چند مقدار برای یک نمونه از موجودیت داشته باشیم، آن صفت را چند مقداری می گوییم. مانند مدرک دانشگاهی برای موجودیت استاد که می تواند دارای مقادیری برای لیسانس، فوق لیسانس و دکتری باشد. یا شماره تلفن برای موجودیت دانشکده که می تواند چند مقدار داشته باشد.

۳- ذخیره شده (Stored) یا مشتق (Derived)

صفتی که مقادیرش در پایگاه داده ها ذخیره شده باشد صفت ذخیره شده نام دارد. صفتی که مقادیرش در پایگاه ذخیره نشده باشد و از طریق محاسبه روی داده های ذخیره شده بدست آید را صفت مشتق می گویند.

صفت معدل دانشجو یک صفت مشتق است چون مستقیماً در پایگاه ذخیره نشده است و از طریق یکسری محاسبه ها روی داده های موجود در پایگاه داده ها، حاصل می شود.

۴- شناسه (کلید)

صفت شناسه موجودیت، صفتی است که دارای دو ویژگی یکتایی مقدار و کوتاه بودن طول مقادیر می باشد.

۵- مقدار هیچ پذیر

مقدار یک صفت برای برخی از نمونه های یک نوع موجودیت می تواند تعریف نشده (مقدار هیچ) باشد. مانند شماره تلفن یک نمونه کارمند که در پایگاه داده موجود نیست، یا نام استاد یک درس در ترم جاری که تا حالا اعلام نشده است، یا نمره دانشجویان که تا پایان ترم مقداری ندارد.

نوع ارتباط

نوع ارتباط یعنی تعامل (interaction) بین دو یا بیش از دو نوع موجودیت (و یا بین یک نوع موجودیت با خودش) که دارای یک معنای مشخص است و با یک نام بیان می شود. نوع ارتباط حالت خاصی از نوع موجودیت است.

به طور مثال، دو نوع ارتباط حذف و انتخاب بین موجودیتهای دانشجو و درس وجود دارد:

۱- دانشجو درس را حذف می کند.

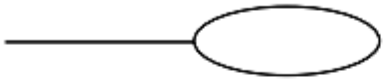
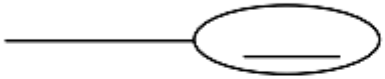
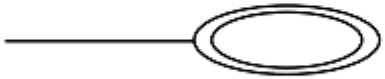
۲- دانشجو درس را انتخاب می کند.

نمودار ER

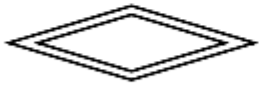
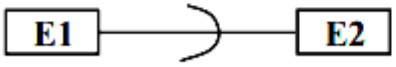
این نمودار مدل کلی پایگاه داده ها در بالاترین سطح انتزاع است که سه مفهوم اساسی مدل ER (نوع موجودیت، صفت و نـوع ارتباط) توسط نمادهایی، نمایش داده می شود.

معنا	نماد
موجودیت	
نوع ارتباط	
مشارکت موجودیت در ارتباط	

نمودار ER

معنا	نماد
صفت	
صفت شناسه	
صفت چند مقداری	
صفت مرکب	

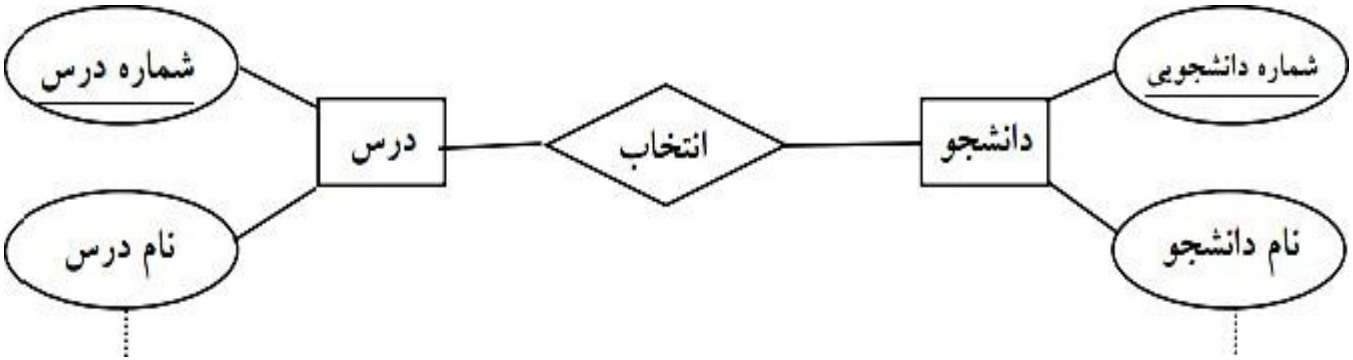
نمودار ER

معنا	نماد
صفت مشتق	
مشارکت الزامی	
موجودیت ضعیف(وابسته)	
نوع ارتباط با موجودیت ضعیف	
ارتباط " گونه ای اســت از ... " E2 IS-A E1	

نمودار ER

مثال

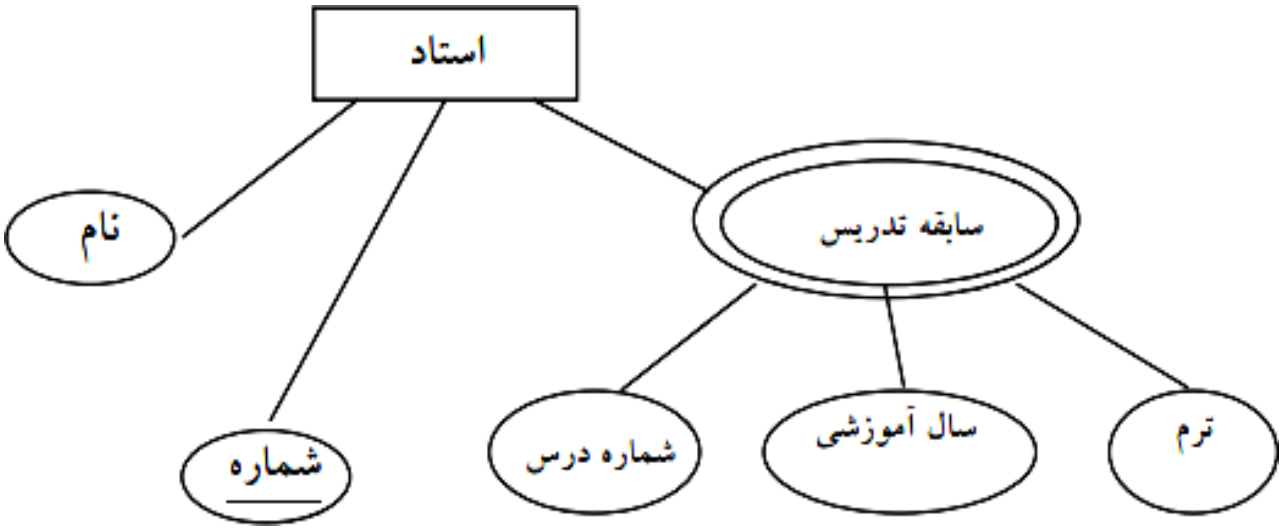
نمودار ER زیر نوع ارتباط انتخاب بین موجودیتهای دانشجو و درس را مشخص می کند:



نمودار ER

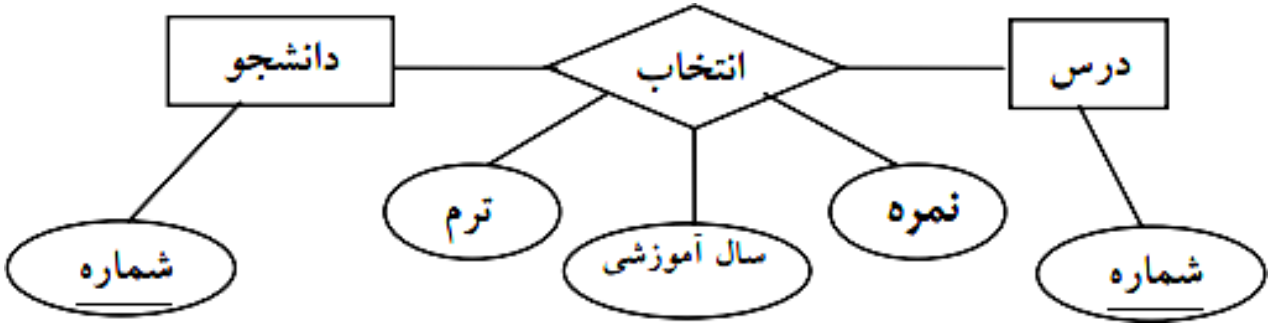
مثال

صفت سابقه تدریس برای استاد، چند مقداری است:



مثال

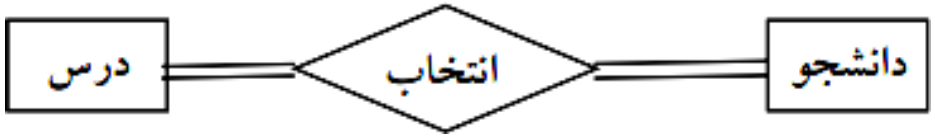
نوع ارتباط می تواند دارای صفت باشد. صفت نوع ارتباط ، می تواند تک مقداری یا چند مقداری باشد. در شکل زیـر، انتخـاب، صفت نوع ارتباط با سه صفت "سال آموزشی، ترم و نمره" می باشد.



انواع مشارکت

مشارکت بر دو نوع است: ۱- الزامی (کامل) ۲- غیر الزامی (ناکامل)

مشارکت یک موجودیت در یک ارتباط را الزامی گویند، اگر همه نمونه های آن موجودیت در آن ارتباط شرکت کنند، در غیر اینصورت مشارکت غیر الزامی است.



مشارکت بین دانشجو و درس در رابطه حذف الزامی نمی باشد چون لزوماً همه دانشجویان درسی را حذف نمی کنند.

نوع ارتباط نیز خود نوعی موجودیت است (موجودیت ضعیف)، بنابراین می تواند دارای صفاتی باشد که معمولاً فاقد صفت کلید است. مثلاً ارتباط انتخاب دارای صفاتی چون ترم، نمره و سال آموزشی می باشد.

ارتباط یک نوع موجودیت ضعیف با یک نوع موجودیت قوی، دارای صفت نمی باشد.

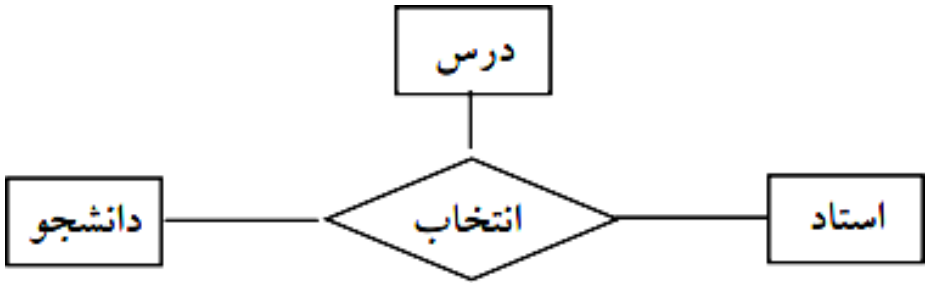
در مشارکت نوع موجودیت ضعیف در نوع ارتباط شناسا، مشارکت در نوع الزامی است.

درجه نوع ارتباط

تعداد شرکت کنندگان در یک ارتباط را درجه(چندی) آن ارتباط گویند.

مثال

ارتباط زیر از درجه سه گانی (Ternary) است:



چندی نوع ارتباط R ، همان نرخ کاردینالیتی R است.

چندی نوع ارتباط در وضع مشارکت نوع موجودیت ها در ارتباط تاثیر ندارد.

انواع تناظر

انواع تناظر عبارتند از:

۱- یک به یک (1:1) ۲- یک به چند (1:N) ۳- چند به چند (N:M)

در ارتباط یک به یک موجودیت R1 با R2 ، یک نمونه از R1 حداکثر با یک نمونه از R2 ارتباط دارد و برعکس.

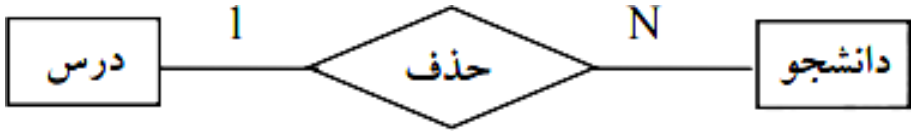
در ارتباط یک به چند R1 با R2 ، یک نمونه از R1 با تعدادی از نمونه های R2 ارتباط دارد ولی یک نمونه از R2 حداکثر با یک نمونه از R1 ارتباط دارد.

در ارتباط چند به چند R1 با R2 ، یک نمونه از R1 با تعدادی از نمونه های R2 ارتباط دارد و برعکس.

نمودار ER

مثال

تناظر بین موجودیت های زیر N : 1 است، یعنی یک دانشجو یک درس را حذف می کند ولی یک درس ممکن است توسط چند دانشجو حذف شود.



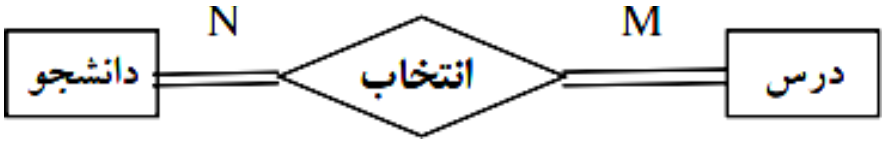
مثال

ارتباط بین موجودیت های زیر 1:1 است. یعنی یک استاد می تواند فقط مدیر یک گروه آموزشی باشد و هر گروه آموزشی فقط یک مدیر دارد.

نمودار ER

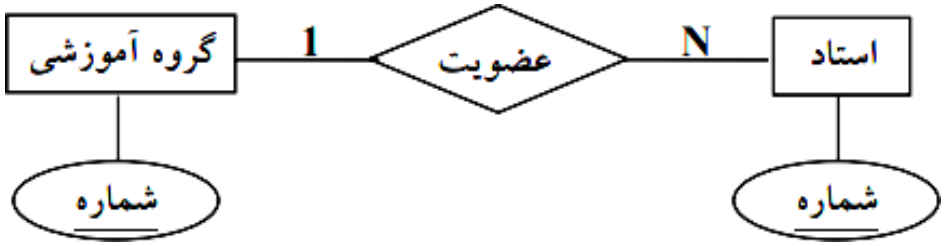
مثال

ارتباط بین موجودیت های زیر M: N است، یعنی یک درس توسط چند دانشجو می تواند انتخاب شود و یک دانشجو می تواند چند درس را انتخاب کند.



مثال

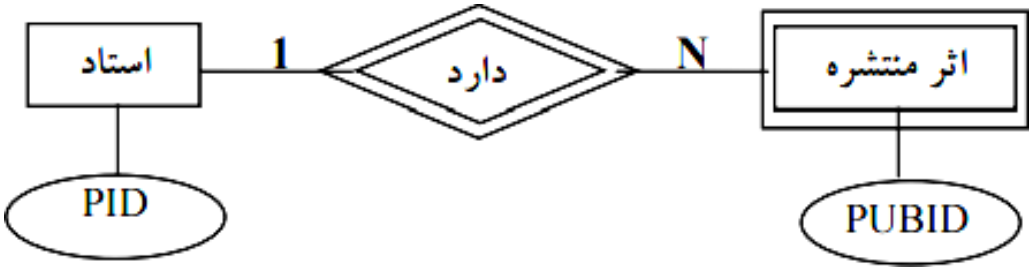
ارتباط بین گروه آموزشی و استاد 1:N است. یعنی در هر گروه اموزشی، N استاد عضویت دارند و هر استاد فقط در یک گروه آموزشی عضو می باشد:



نمودار ER

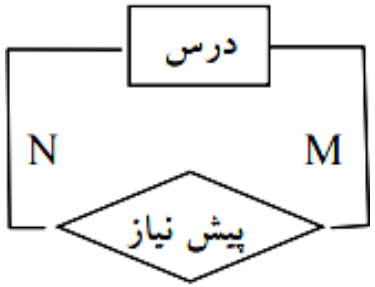
مثال

ارتباط بین استاد و اثر منتشره 1:N است. یعنی هر استاد می تواند N اثر داشته باشد و هر اثر متعلق به یک استاد است. اثر منتشره یک موجودیت ضعیف است.



مثال

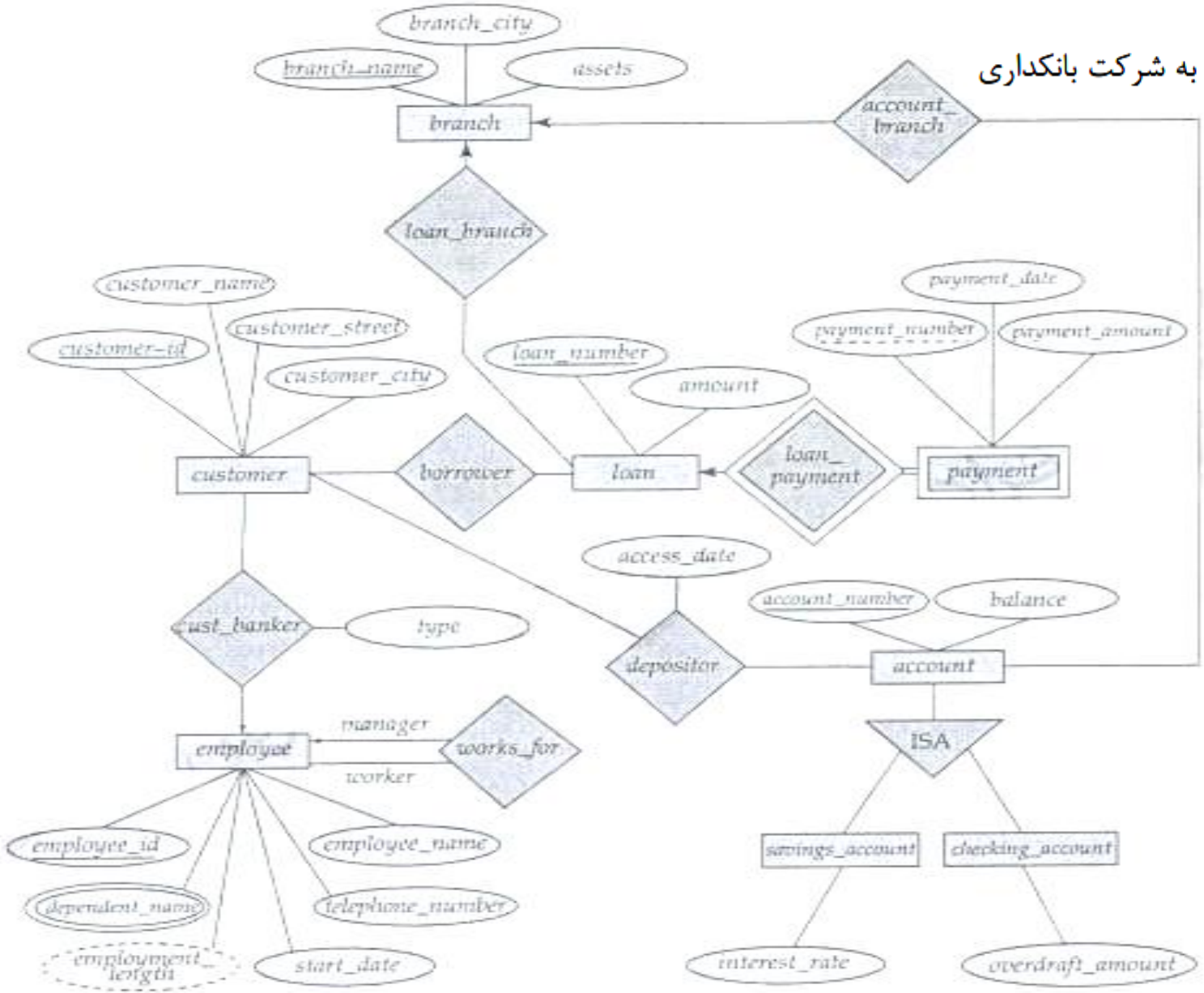
ارتباط پیشنیازی بین موجودیت درس N:M است :



نمودار ER

مثال

نمودار ER مربوط به شرکت بانکداری



دام های پیوندی (Connection traps)

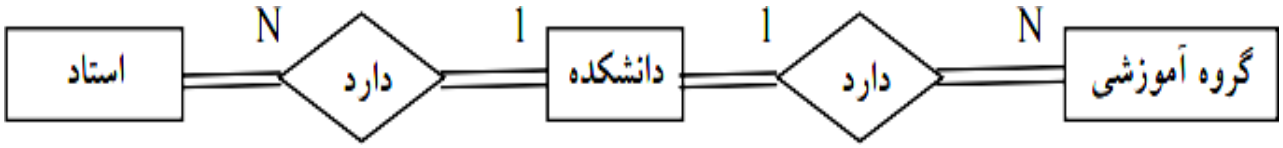
یکی از معایب مدلسازی به روش ER ، دامهای پیوندی است که از درک نادرست معنای ارتباطات ناشی می شود. دام های پیوندی رایج عبارتند از:

۱- دام یک چندی(چند شاخه) (Fan trap)

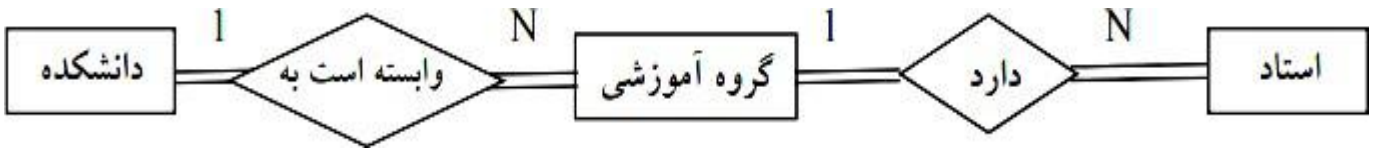
۲- دام شکاف(گسل) (Chasm trap)

دام یک – چندی

این دام وقتی ایجاد می شود که ارتباطی بین چند نوع موجودیت وجود داشته باشد، اما مسیر ارتباطی مبهم باشد.

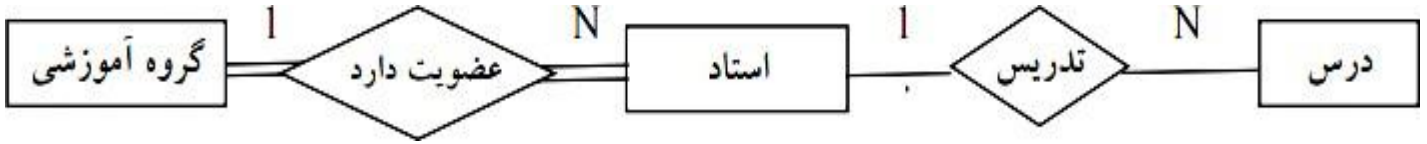


مدل سازی را به صورت زیر باید اصلاح کرد تا چنین دامی ایجاد نشود:



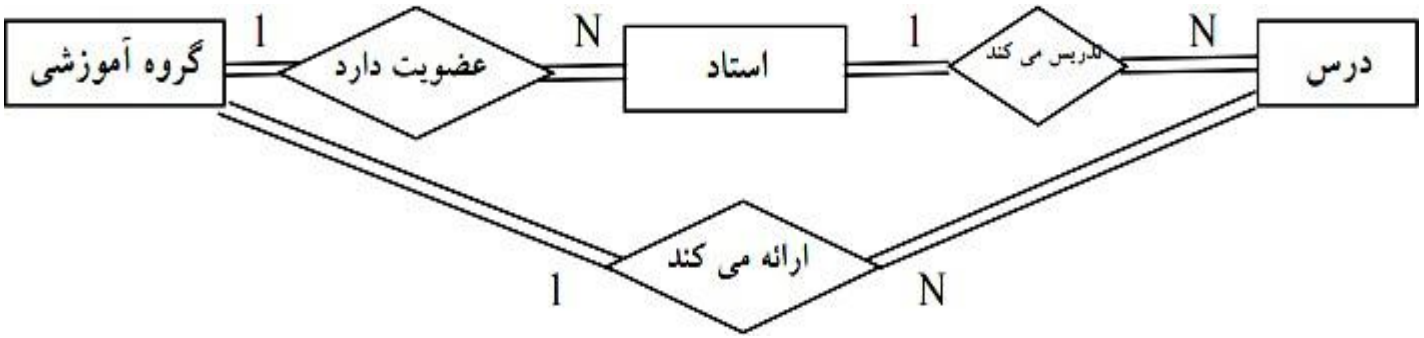
دام شکاف

این دام وقتی ایجاد می شود که مشارکت یک موجودیت در یک ارتباط الزامی نباشد. نمودار زیر معرف این دام است. در این حالت نمی توان به سئوال "درس پایگاه داده ها در کدام گروه آموزشی ارائه می شود" پاسخ داد. (با فرض اینکه استادی این درس را ارائه ندهد.)



مدل ER

مدل سازی را به صورت زیر باید اصلاح کرد تا چنین دامی ایجاد نشود:

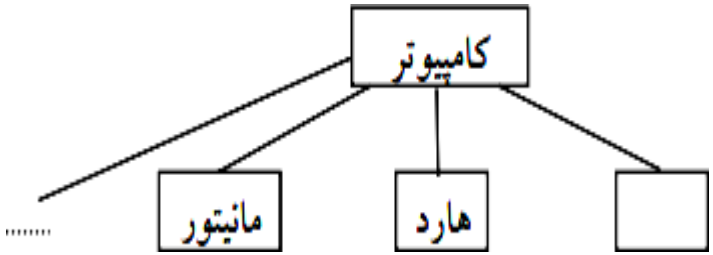


روش EER

روش EER ، روشی است که در آن می توان از مفاهیم زیر که در مدلسازی شیء گرا وجود دارد، استفاده کرد.

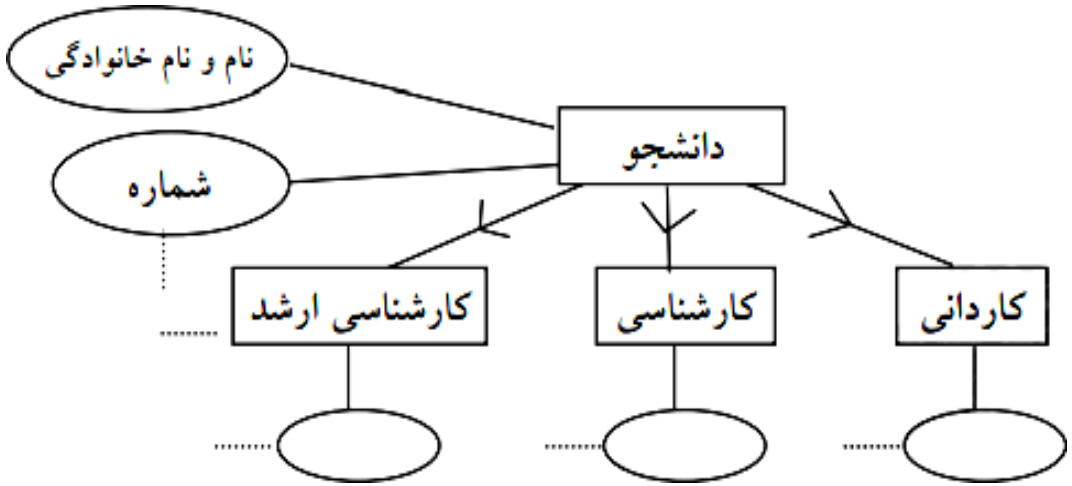
۱- تجزیه و ترکیب

جداسازی یک شیء کل به اجزاء تشکیل دهنده آن را تجزیه و عکس عمل تجزیه را ترکیب می نامند. شکل زیر مثالی از تجزیه و ترکیب است:



۲- تخصیص و تعمیم

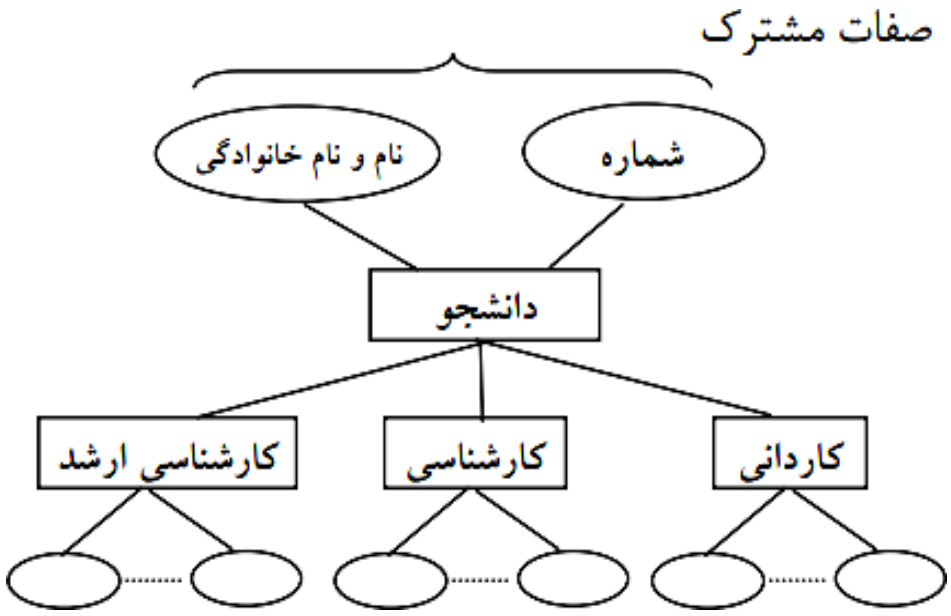
تخصیص یعنی مشخص کردن انواع خاص یک موجودیت است. مثالی از تخصیص برای موجودیت دانشجو:



روش EER

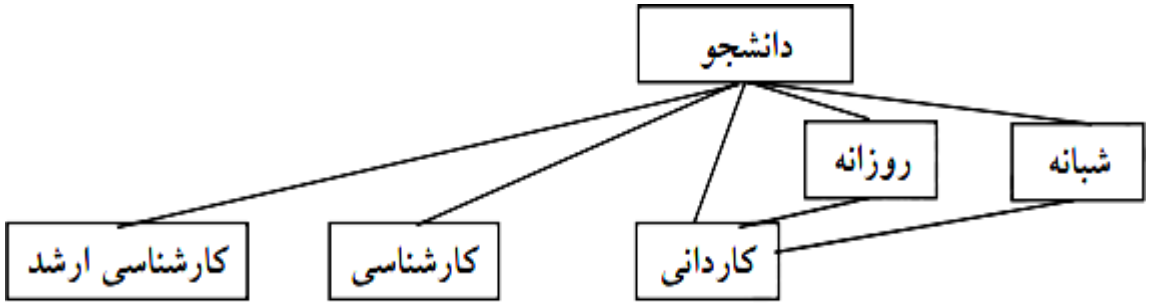
موجودیت دانشجو دارای سه زیر نوع "دانشجوی کاردانی، کارشناسی و کارشناسی ارشد" می باشد که صفاتی چون شماره و نام در همه آنها وجود دارد.(صفات مشترک)

تعمیم عکس عمل تخصیص است. تعمیم یعنی تشخیص حداقل صفت شناسه مشترک بین حداقل دو نوع موجودیت و انتساب آن صفت به یک نوع موجودیت جدید است. مثال زیر معرف تعمیم است.



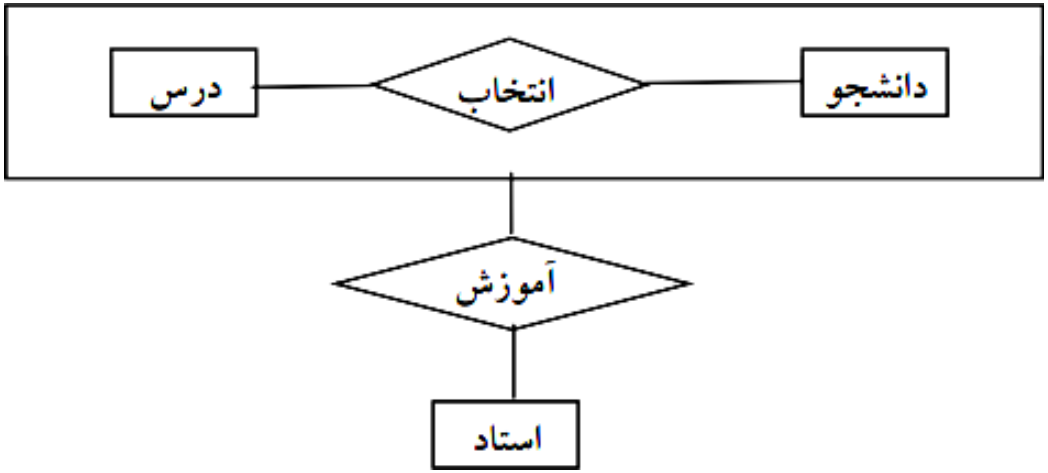
۳- وراثت چندگانه (Multiple inheritance)

یک زیر نوع می تواند، متعلق به چند موجودیت باشد. در مثال زیر، نوع دانشجوی کاردانی هم متعلق به موجودیت دانشجو می باشد و هم متعلق به دانشجوی روزانه و هم دانشجوی شبانه:



۴- تجمیع

می توان دو یا بیش از دو موجودیت مرتبط را به صورت یک موجودیت واحد در نظر گرفت. زمانی از تجمیع استفاده می کنیم که بخواهیم ارتباطی را بین ارتباط ها بیان کنیم. در مثال زیر تجمیع نمایش داده شده است.



فصل سوم

مدل رابطه ای

مدل رابطه ای

تعریف رابطه از نظر کاد

رابطه R تعریف شده روی n مجموعه S_1 تا S_n، زیر مجموعه ای از ضرب کارتزین آنها می باشد.

مثال

رابطه Student (S1,S2,S3) روی سه مجموعه زیر تعریف شده است:

S1 = {شماره} , S2 = {نام} , S3 = {نمره}

که نمایش جدولی آن به صورت زیر است:

S1	S2	S3
120	Ali	14
198	Reza	20

مدل رابطه ای

میدان

میدان، مجموعه ای است نامدار از مقادیر همنوع که یک یا بیش از یک صفت از آن مقدار می گیرند. از نظر کاد، مفهوم میدان،گسترش یافته مفهوم نوع داده است. مزایای میدان عبارتند از:

۱- امکانی برای کنترل مقداری پرسش ها

۲- امکانی برای کنترل معنایی پرسش ها

۳- امکانی برای تسریع پاسخدهی به برخی از پرسش ها

۴- امکانی برای ساده تر شدن شمای پایگاه داده ها.

مدل رابطه ای

تعریف رابطه از نظر دیت

با فرض وجود n میدان D1 , D2 , ... , Dn نه لزوماً متمایز، رابطه از دو قسمت تشکیل شده است:

۱- عنوان (Heading) : مجموعه اسامی صفات خاصه

۲- بدنه (Body) : مجموعه ای از تاپل ها

مثال

در رابطه R که در زیر نشان داده شده، مجموعه عنوان، مجموعه بدنه، درجه و کاردینالیتی رابطه را بیان کنید.

A	B	C
a1	b1	c1
a2	b2	c2

حل: رابطه R (A,B,C) دارای ۳ صفت خاصه و۲ سطر است، بنابراین درجه رابطه ۳ و کاردینالیتی آن ۲ است.

مجموعه عنوان رابطه R : HR = {A,B,C}

و مجموعه بدنه رابطه R : BR = {{ a1 , b1 , c1 } , {a2 , b2 , c2 } }

مدل رابطه ای

خواص رابطه

یک رابطه دارای خواص زیر است:

۱- تاپل تکراری ندارد.

یک مجموعه دارای عناصر تکراری نمی باشد و چون تاپلها، عناصر مجموعه پیکر هستند، تاپل تکراری در رابطه وجود ندارد.

۲- تاپلها نظم ندارند.

عناصر مجموعه دارای نظم نمی باشند و چون تاپلها، عناصر مجموعه پیکر هستند، نظم ندارند.

۳- صفات رابطه نظم ندارند (از چپ به راست).

عناصر مجموعه دارای نظم نمی باشند و چون صفات، عناصر مجموعه عنوان هستند، دارای نظم نمی باشند.

مدل رابطه ای

۴- مقادیر تمام صفات، تجزیه نشدنی(اتومیک) هستند.

صفت اتومیک، صفتی است که اگر آن را به اجزایی تجزیه کنیم، اجزای حاصل بی معنا باشند. البته تجزیه ناپذیری مفهومی نسبی است و بستگی به کاربردهای خاص دارد. مثلاً صفت تاریخ می تواند از سه جزء سال، ماه و روز تشکیل شود که این اجزاء ممکن است در یک کاربرد با معنا و در کاربردی دیگر بی معنا باشند.

صفات رابطه نظم ندارند، یعنی به صفات رابطه از طریق نام آنها دسترسی می شود، در حالیکه ستونهای جدول نظم دارند، یعنی به ستونهای جدول از طریق مکان آنها دسترسی می شود.

به دلیل خاصیت تک مقداری بودن صفات رابطه، مدل رابطه ای در نمایش داده های پیچیده مشکل دارد.

مدل رابطه ای

انواع کلید

کلید کاندید(C.K)	هر زیر مجموعه از مجموعه عنوان که دارای خاصیت یکتایی مقدار و کاهش ناپذیری باشد.
کلید اصلی(P.K)	یکی از کلیدهای کاندید که توسط طراح پایگاه داده انتخاب می شود.
کلید بدیل(A.K)	هر کلید کاندید غیر از کلید اصلی.
کلید خارجی (F.K)	با فرض وجود دو رابطه R1 , R2 ، هر زیر مجموعه از صفات R2 که در R1 کلید کاندید باشد، کلید خارجی R2 است. (R1 وR2 لزوما متمایز نیستند)
سوپر کلید(S.K)	هر ترکیبی از اسامی صفات رابطه که در هیچ دو تاپل، مقدار یکسانی نداشته باشد.

نکاتی در رابطه با کلید کاندید

۱- کلید کاندید می تواند ساده یا مرکب باشد.

۲- رابطه ممکن است بیش از یک کلید کاندید داشته باشد.

۳- کلید کاندید امکانی است برای ارجاع به یک تاپل در رابطه .

۴- کلیدهای کاندید یک رابطه ممکن است صفت مشترک داشته باشند.

۵- رابطه ای که کلید کاندید آن از ترکیب تمام صفات رابطه حاصل می شود، تمام کلید (ALL KEY) نام دارد.

۶- در شمای ادراکی، کلیدهای کاندید باید معرفی شوند.

۷- کلید کاندید می توان هیچمقدار داشته باشد.

۸- هر رابطه ای حتماً کلید اصلی دارد، چون هر رابطه حداقل یک کلید کاندید دارد.

۹- اگر تعداد کلیدهای کاندید رابطه R برابر N باشد، تعداد کلیدهای بدیل برابر N-1 است.

۱۰- حداکثر تعداد کلیدهای کاندید ساده یک رابطه درجه n برابر n است.

مدل رابطه ای

۱۱- حداکثر تعداد کلیدهای کاندید یک رابطه درجه n برابر $C^{n}_{\left\lceil \frac{n}{2} \right\rceil}$ است.

۱۲- یک رابطه با درجه n ، حداکثر دارای $\frac{n!}{m! \times (n-m)!}$ کلید کاندید m صفتی است. $(m \geq 1)$

۱۳- حداکثر تعداد کلیدهای کاندید N صفتی دوبدو و ناهمپوشای یک رابطه درجه n برابر $\left\lfloor \frac{n}{N} \right\rfloor$ است.

۱۴- کلید کاندید کاهش ناپذیر است. یعنی اگر یکی از عناصر کلید حذف شود، باقیمانده لزوماً کلید کاندید نیست.

نکاتی در رابطه با کلید خارجی

۱- کلید خارجی می تواند مقدار تکراری داشته باشد.

۲- کلید خارجی می تواند مقدار تهی (Null) داشته باشد.

۳- کلید خارجی برای نمایش ارتباطات بین انواع موجودیت ها بکار می رود.

۴- کلید خارجی یک رابطه، می تواند با نام دیگر، کلید کاندید در همان رابطه باشد.

۵- کلید خارجی یک رابطه، می تواند با نام دیگر، کلید کاندید در رابطه ای غیر از آن رابطه باشد.

۶- کلید خارجی یک رابطه، می تواند با هر نام، کلید کاندید در هر تعداد رابطه باشد.

۷- تنها امکان نمایش ارتباط بین دو موجودیت، کلید خارجی نیست بلکه یک صفت مشترک نیز می تواند یک ارتباط ایجاد کند.

مدل رابطه ای

۸- از معایب کلید خارجی می توان بروز افزونگی و فزونکاری سیستم به خاطر کنترل جامعیت را نام برد.

۹- در کلید خارجی، با افزایش افزونگی، کار لازم برای کنترل جامعیت افزایش می یابد.

۱۰- کلید خارجی و کلید بدیل ممکن است در رابطه ای وجود نداشته باشند.

۱۱- در رابطه نشان دهنده نوع ارتباط با چندی 1:N ،کلید خارجی لزوما جزء تشکیل دهنده کلید کاندید نیست.

۱۲- در یک رابطه با درجه n و با یک کلید کاندید ساده، حداکثر n کلید خارجی ساده می تواند وجود داشته باشد.

۱۳- تعداد کلید خارجی یک رابطه می تواند صفر باشد.

۱۴- رابطه با درجه n، حداکثر 2^{n-1} کلید خارجی دارد.

نکاتی در رابطه با سوپرکلید

۱- هر کلید کاندید، یک سوپر کلید است.

۲- هر سوپر کلید(ابر کلید)، شامل حداقل یک کلید کاندید است.

۳- سوپر کلید دارای خاصیت یکتایی مقدار است.

۴- همه صفات با سوپر کلید وابستگی تابعی دارند.

۵- سوپر کلید، کاهش پذیر است.

۶- سوپر کلید می توان هیچمقدار داشته باشد.

۷- اگر H_R عنوان رابطه R ، $G \subset H_R$ و t_i و t_j دو تاپل دلخواه متمایز از R باشند و $t_i(G) \neq t_j(G)$ ، در اینصورت G سوپر کلید R است.

۸- رابطه با درجه n، حداکثر $2^n - 1$ سوپر کلید دارد.

۹- رابطه با درجه n، با دو کلید کاندید ساده، دارای $3 \times 2^{n-2}$ سوپر کلید است.

۱۰- یک رابطه با درجه n و تعداد k کلید کاندید ساده،دارای $2^n - 2^{n-k}$ سوپر کلید دارد.

مدل رابطه ای

مثال

در رابطه R(A,B,C,D,E,F,G) ، صفات A و (B,D) کلیدهای کاندید هستند. چند سوپر کلید را نام ببرید.

حل:

CBDFG , BDEF , ABCD

مثال

کلید های کاندید و خارجی را در رابطه های زیر مشخص کنید.

ST (شماره گروه آموزشی، رشته تحصیلی،سطوح دوره تحصیلی ، نام دانشجو ، شماره دانشجویی)

CT (شماره گروه آموزشی ارائه کننده درس ، نوع درس ، تعداد واحد ، عنوان درس ، شماره درس)

SCT (..... شماره درس ، شماره دانشجویی)

مدل رابطه ای

حل: **کلید کاندید**:

شماره دانشجویی در ST و شماره درس در CT و ترکیب شماره درس و شماره دانشجو در SCT.

تذکر: اگر عنوان هیچ دو درس یکسان نباشد آنگاه می توان عنوان درس را نیز در CT کلید کاندید در نظر گرفت.

کلید خارجی:

شماره دانشجویی در SCT کلید خارجی است، چون همین صفت در ST کلید اصلی است.

شماره درس در SCT کلید خارجی است، چون همین صفت در CT کلید اصلی است.

مدل رابطه ای

مثال

بانک تهیه کننده- قطعه با سه جدول s,p,sp مفروض است:

S (s# , sname , status , city)
P (p# , pname , color , weight , city)
SP(s# , p# , qty)

حل:

کلید اصلی : صفت s# در رابطه s و صفت p# در رابطه p و ترکیب صفات s#,p# در رابطه sp.

کلید خارجی: صفت s# در رابطه sp کلید خارجی است چون در رابطه s کلید اصلی است.

صفت p# در رابطه sp کلید خارجی است، چون در رابطه p کلید اصلی است.

مثال

در دو رابطه زیر کلیدهای کاندید و خارجی را تعیین کنید.

DEPT(شماره دپارتمان , نام دپارتمان , تلفن , شماره مدیر دپارتمان)

PROF (شماره مدرس , نام مدرس , شماره دپارتمان مدرس)

حل:

کلید های کاندید: شماره دپارتمان در DEPT و شماره مدرس در PROF .

کلیدهای خارجی: شماره مدیر دپارتمان در DEPT کلید خارجی است چون در PROF کلید اصلی است.

شماره دپارتمان مدرس در PROF کلید خارجی است چون در DEPT کلید اصلی است.

لازم به ذکر است که مدیر دپارتمان خود یک مدرس است.

مدل رابطه ای

قواعد جامعیت

جامعیت پایگاه داده ها یعنی صحت، دقت و سازگاری داده های ذخیره شده در پایگاه در تمام لحظات. بروز عواملی چون اشتباه در ورود اطلاعات، اشتباه در برنامه های کاربردی، وجود افزونگی کنترل نشده و خرابی های سخت افزار و نرم افزاری موجب نقض جامعیت می شوند. **قواعد جامعیت بر ۳ نوع است:**

۱- قاعده میدانی

قاعده مشخص کننده مقادیر مجاز یک میدان (مثلاً مقادیر میدان نمره اعداد از ۰ تا ۲۰ است)

۲- قواعد خاص

قواعد جامعیت خاص(کاربردی)، قواعدی هستند که توسط کاربر، مجاز تعریف می شوند. DBMS به کاربر امکان تعریف این قواعد جامعیت را می دهد. مجموعه قواعد خاص یک محیط عملیاتی، باید مورد تایید مدیر داده ها(DA) برسد و سپس DBA آنها را در طراحی و پیاده سازی منظور نماید.

مدل رابطه ای

قواعد خاص بر سه نوع می باشند:

الف- قاعده صفتی : قاعده بیان کننده نوع صفت

مثلاً صفت نام دانشجویی از نوع کاراکتر است.

ب- قاعده رابطه ای : قاعده بیان کننده مقادیر مجاز یک متغیر رابطه ای

مثلاً در رابطه درس، درس عملی از گروه آموزشی خاصی نمی تواند بیشتر از دو واحد داشته باشد.

ج- قاعده پایگاهی : قاعده ناظر به چند متغیر رابطه ای مرتبط با هم.

مثلاً در رابطه های دانشجو،درس، مدرس و دانشجو– درس– مدرس این محدودیت وجود داشته باشد که "مدرس با مرتبه فوق لیسانس به بالا نباید درسی از دوره کاردانی را تدریس کند."

تذکر: در سیستم های موجود برای معرفی این قواعد از مکانیسم اظهار (Assertion) استفاده می شود.

۳- قواعد عام

قواعد عام (متا قواعد)، قواعدی که توسط هر سیستم رابطه ای در پایگاه رابطه ای اعمال می شوند و بـه داده هـای خـاص وابسته نیستند و بر دو نوع می باشند:

الف- **قاعده جامعیت موجودیتی:** هیچ جزء تشکیل دهنده کلید اصلی نباید تهی باشد.

ب- **قاعده جامعیت ارجاعی** : مقدار کلید خارجی یک رابطه نمی تواند در رابطه مرجع وجود نداشته باشد.

مدل رابطه ای

راههای اعمال قواعد جامعیت

۱- معرفی کلید اصلی

۲- معرفی کلید خارجی

۳- معرفی میدان و مقادیر آن

۴- معرفی وابستگی های تابعی

۵- اعلام هیچ مقدار ناپذیری صفت

۶- اعلان محدودیت ها در شمای پایگاه

فصل چهارم

زبان رابطه ای SQL

زبان رابطه ای SQL

اصولاً هر RDBMS دارای نسخه SQL خاص خود است. ویژگی های SQL عبارت است از:

۱- زبانی غیر رویه ای

۲- دارای عملگرهای بسیار قوی

۳- تامین کننده استقلال داده ای

۴- دارای اکمال ساختاری

۵- شامل تمام داده های استاندارد

۶- قابل استفاده هم به صورت مستقل و هم به صورت ادغام شده

زبان رابطه ای SQL

انواع عملگرها در SQL

توسط عملگرها، عمل خاصی را انجام می دهیم. انواع عملگرها عبارتند از:

۱- محاسباتی (+,-,*,/,%)

۲- رابطه ای (=,>,<,<=,>=, !=)

۳- منطقی(AND,OR,NOT)

۴- بیتی (^,|,&,~)

۵- انتساب(=)

۶- ویژه (IN,BETWEEN,ALL,ANY,LIKE)

دستورهای SQL

دستورات در SQL را می توان به سه دسته زیر تقسیم بندی کرد:

۱- **DML** : (SELECT , INSERT , UPDATE , DELETE)

۲- **DDL** : (CREATE , ALTER , DROP)

۳- **DCL** : (GRANT , REVOKE)

ایجاد جدول (CREATE TABLE)

مثال

ایجاد جدول دانشجو (S) با فیلد های شماره دانشجویی(کلید) و نام دانشجو:

```
CREATE TABLE S
 (
   ID      Char (10)   PRIMARY KEY ,
   Name   Char (20),
 );
```

حذف جدول (DROP TABLE)

برای حذف یک جدول از دستور DROP TABLE ، استفاده می شود. وقتی جدولی حذف شود، فایل متناظر جدول حذف شده و تعریف جدول از کاتالوگ خارج می شود. برای حذف جدول S ، از دستور زیر استفاده می کنیم:

```
DROP TABLE S;
```

درج (INSERT)

از این دستور برای اضافه کردن رکورد به جداول بانک اطلاعاتی استفاده می شود.

مثال

درج اطلاعات یک دانشجو به جدول STUDENT:

```
INSERT INTO S (ID,Name)
VALUES ('110' , 'Ali' );
```

بهنگام سازی (UPDATE)

برای ویرایش رکوردهای جدول از دستور UPDATE استفاده می کنیم.

مثال

شماره دانشجویی 100 را به 200 تغییر دهید.

```
UPDATE S
SET ID=200
WHERE ID=100;
```

بازیابی (SELECT)

مثال

بازیابی شماره و نام دانشجویان از جدول S :

```
SELECT  ID, Name
FROM    S;
```

مثال

بازیابی شماره دانشجویانی که نام آنها ALI باشد از جدول S :

```
SELECT  ID
FROM    S
WHERE   Name='ALI';
```

زبان رابطه ای SQL

مثال

بازیابی شماره و نام دانشجویانی که نام آنها ALI باشد:

```
SELECT  *
FROM    S
WHERE   Name='ALI';
```

مثال

بازیابی نام دانشجویان بدون نمایش نام های تکراری:

```
SELECT  DISTINCT  SNAME
FROM  S;
```

توابع جمعی (Aggregate Functions)

توابع جمعی در قسمت جبر رابطه ای توضیح داده شد. این توابع عبارتند از:

COUNT , MAX , MIN , SUM , AVG

مثال

پرس و جویی که تعداد دانشجویان را بر می گرداند:

```
SELECT  COUNT(*)
FROM    S;
```

مرتب سازی رکوردها

توسط امکان ORDER BY می توان جدول جواب را برحسب یک یا بیش از یک ستون به صورت صعودی (ASC) یا نزولی (DESC) مرتب کرد. (ASC پیش فرض است).

```
SELECT *
FROM S
ORDER BY ID DESC;
```

تست مقدار تهی فیلد

توسط این امکان می توان وجود مقدار هیچ در یک ستون را بررسی کرد.

مثال

شماره دانشجویانی را بدهید که نمره آنها در درس C1 هنوز اعلام نشده است.

```
SELECT SID
FROM SC
WHERE CID = 'C1' AND GRADE IS NULL;
```

عملگر LIKE

توسط این امکان در SQL می توان عمل بازیابی را بر اساس نشانوند جستجوی کاراکتر با شرایط مورد نظر انجام داد. به عبارتی مشخص می کند که آیا رشته ای در قسمتی از فیلد قرار دارد یا خیر.

تذکر: به جای کاراکتر – می تواند یک کاراکترو به جای کاراکتر % می تواند تعدادی کاراکتر قرار گیرد.

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۴ حرفی است و حرف دوم در نام آنها A باشد، مانند SARA.

```
SELECT *
FROM   ST
WHERE  SNAME  LIKE  '-A--';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها با کاراکتر A شروع شود.

```
SELECT *
FROM   ST
WHERE  SNAME LIKE  'A%';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۳ حرفی است و حرف اول، یکی از کاراکترهای A,B,C,D باشد.

```
SELECT  *
FROM    ST
WHERE  SNAME  LIKE  '[A-D]--';
```

مثال

مشخصات دانشجویانی را بدهید که نام آنها ۳ حرفی است و حرف اول یکی از کاراکترهای A,B,C,D نباشد:

```
SELECT  *
FROM    ST
WHERE  SNAME  LIKE  '[^A-D]--';
```

عملگر UNION

مثال

با فرض اینکه اطلاعات دانشجویان در بانک S و اطلاعات اساتید در بانک T قرار دارد، برای مشخص کردن نام دانشجویان و اساتید از دستور زیر استفاده می کنیم:

```
(SELECT NAME   FROM S )
UNION
(SELECT TNAME FROM T );
```

عملگر BETWEEN

توسط این امکان می توان وجود یک مقدار را در یک محدوده بررسی کرد.

مثال

شماره دانشجویانی را بدهید که نمره آنها در درس C1 بین ۱۰ تا ۱۲ باشد.

```
SELECT SC.SID
FROM   SC
WHERE CID='C1' AND GRADE BETWEEN 10 AND 12;
```

گروه بندی اطلاعات

توسط امکان گروه بندی، می توان سطرهای یک جدول را برحسب مقادیر یک ستون گروه بندی کرد به نحوی که در هر گروه مقدار آن ستون یکسان باشد.

مثال

گروه بندی جدول R بر حسب ستون Y :

```
SELECT *
FROM   R
GROUP BY Y;
```

X	Y	Z
X1	Y1	Z1
X1	Y2	Z2
X2	Y1	Z2
X3	Y1	Z3
X3	Y3	Z4

⇒

X	Y	Z
X1	Y1	Z1
X2	Y1	Z2
X3	Y1	Z3
X1	Y2	Z2
X3	Y3	Z4

مثال

گروه بندی جدول دانشجو – درس برحسب مقادیر ستون شماره CID :

```
SELECT  SID , AVG(GRADE)
FROM    SC
GROUP BY CID;
```

استفاده از HAVING در دستور SELECT

اگر بخواهیم از توابع جمعی در شرط WHERE استفاده کنیم، پیام خطا صادر می شود و باید از HAVING به جای WHERE استفاده کرد.

بنابراین دستور زیر نادرست است:

```
SELECT   S# , AVG(QTY)
FROM     SP
WHERE    AVG(QTY) > 100
GROUP    BY  S#;
```

دستور صحیح به صورت زیر می باشد:

```
SELECT  S#,AVG(QTY)
FROM    SP
GROUP BY  S#
HAVING  AVG(QTY) > 100;
```

پیوند رابطه ها

برای پاسخ به بعضی از پرسش ها نیاز است که به بیش از یک جدول مراجعه شود. در این حالت از پیوند باید استفاده شود.

مثال

نام دانشجویانی را بدهید که درس شماره C3 را انتخاب کرده اند.

```
SELECT  ST.SNAME
FROM    ST , SC
WHERE  ST.SID = SC.SID   AND   SC.CID = ' C3  ';
```

مثال

با فرض وجود سه جدول T1,T2,T3 از یک پایگاه داده ها:

۱- پیوند ضربدری(Cross join) دو جدول T1 و T2:

```
SELECT  *
FROM    T1,T2;
```

۲- پیوند ضربدری دو جدول T1 و T2 با شرط θ:

```
SELECT  *
FROM    T1,T2
WHERE   θ;
```

۳- پیوند دو جدول T1 و T2 روی فیلد مشترک F :

```
SELECT  *
FROM    T1,T2
WHERE   T1.F = T2.F;
```

که می توان به صوت زیر نیز نوشت:

```
SELECT  A.*
FROM    T1  A,T2  B
WHERE   A.F = B.F;
```

۴- پیوند سه جدول T1 و T2 و T3 .

ارتباط بین دو جدول T1 و T2 از طریق فیلد مشترک F1 و ارتباط بین دو جدول T2 و T3 از طریق فیلد مشترک F2 برقرار می شود. (X,Y,Z,W فیلدهای موجود در سه جدول هستند.) (θ شرط معمولی است.)

```
SELECT   X,Y,Z,W
FROM     T1  A,T2  B, T3  C
WHERE    A.F1=B.F1  AND  B.F2 = C.F2  AND  θ ;
```

پیوند درونی و بیرونی

۱- پیوند درونی

در مواقعی که تعداد شرط ها زیاد می شود، برای ساده شدن تشخیص شرط پیوند با شرط معمولی ، از پیوند درونی (INNER JOIN) استفاده می شود. در این نوع پیوند، شرط معمولی در قسمت WHERE و شرط الحاق در قسمت INNER JOIN مـی اید.

```
SELECT  X,Y,Z,W   FROM  T1  A
INNER   JOIN  T2  B  ON  A.F1= B.F1
INNER   JOIN  T3  C  ON  B.F2 = C.F2
WHERE  شرط معمولی ;
```

۲- پیوند بیرونی

در این نوع پیوند، کلیه رکوردهای موجود در یک جدول، حتی رکوردهایی که در جدول دیگر وجود ندارند، ارزیابی شده و در خروجی نمایش داده می شود. پیوند بیرونی دارای سه نوع "چپ، راست و کامل" می باشد.

```
SELECT  *
FROM  T1
LEFT OUTER JOIN  T2
ON  شرط پیوند ;
```

مثال

دو جدول M و N مفروض هستند. حاصل اجرای دستور داده شده را بدست آورید؟

```
SELECT  *
FROM  N
LEFT  OUTER  JOIN  M
ON  N.B=M.B;
```

N

A	B	C
a1	b1	c1
a2	b4	c2
a3	b2	c3

M

B	D
b1	d1
b2	d2
b2	d5

پرسش های تودرتو (Nested Query)

پرسش هایی که تعدادی پرسش فرعی (درونی) در درون خود دارند را پرسش تودرتو می نامند. در واقع یک دستور SELECT که درون یک دستور SELECT دیگر نوشته شود. در هنگام اجرا ابتدا پرسش داخلی اجرا می شود.

مثال

نام دانشجویانی را بدهید که درس C2 را گرفته اند.

حل:

```
SELECT   SNAME
FROM     ST
WHERE    SID  IN  ( SELECT   SID
                    FROM     SC
                    WHERE    CID='C2' );
```

مثال

نام دانشجویان هم رشته با دانشجوی شماره 110 را بدهید.

حل:

```
SELECT   SNAME
FROM     ST
WHERE    SMJR = ( SELECT    SMJR
                  FROM      ST
                  WHERE     SID = ' 110 ' );
```

مثال

نام افرادی را بدهید که وزن آنها از بیشترین وزن موجود در پایگاه کمتر باشد.(جدول T با دو فیلد NAME,WEIGHT مشخصات افراد شامل نام و وزن آنها را نگهداری می کند.)

```
SELECT   T
WHERE    WEIGHT < ( SELECT   MAX(WEIGHT)
                    FROM   T );
```

پایگاه داده "تهیه کننده – قطعه"

پایگاه داده ای تهیه کننده و قطعه دارای سه جدول است:

S (S# , SNAME , STAUAS , CITY)
P (P# , PNAME , COLOR , WEIGHT , CITY)
SP (S# , P# , QTY)

زبان رابطه ای SQL

S

S#	SNAME	STATUS	CITY
S1	Sn1	20	C1
S2	Sn2	10	C2
S3	Sn3	30	C2
S4	Sn4	20	C1
S5	Sn5	30	C3

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Pn1	RED	12	C1
P2	Pn2	YELLOW	17	C2
P3	Pn3	BLUE	17	C4
P4	Pn3	GREEN	14	C1
P5	Pn5	BLUE	12	C2
P6	Pn6	BLACK	19	C1

SP

S#	P#	QTY
S1	P1	100
S1	P4	200
S2	P1	300
S2	P2	400
S3	P6	500
S4	P3	500
S5	P2	800
S5	P4	700
S5	P6	200

زبان رابطه ای SQL

- نام تهیه کنندگانی را بیابید که قطعه P2 را تهیه می کنند.

```
SELECT   SNAME
FROM   S
WHERE   S#   IN ( SELECT   S#
                  FROM   SP
                  WHERE   P# = 'P2'
                    );
```

با توجه به رابطه ها، نتیجه SELECT داخلی برابر است با:

S#
S2
S5

و نتیجه SELECT خارجی برابر است با:

SNAME
Sn2
Sn5

نام تهیه کنندگانی را بیابید که اقلاً یک قطعه به رنگ RED تهیه می کنند.

```
SELECT   SNAME
FROM     S
WHERE    S#  IN ( SELECT  S#
                  FROM     SP
                 WHERE   P#  IN ( SELECT  P#
                                  FROM     P
                                  WHERE   COLOR = 'RED'
                                 )
               );
```

شماره قطعاتی را بیابید که توسط بیش از یک تهیه کننده، تهیه شده باشد.

```
SELECT P#
FROM SP
GROUP BY P#
HAVING COUNT(*) > 1;
```

رابطه SP بعد از گروه بندی روی P# :

S#	P#	QTY
S1	P1	100
S2	P1	300
S2	P2	400
S5	P2	800
S4	P3	500
S1	P4	200
S5	P4	700
S3	P6	500

حداکثر مقدار تهیه شده از هر قطعه را بیابید.

```
SELECT   P# , MAX(QTY)
FROM   SP
GROUP BY   P#;
```

جدول جواب به صورت زیر است:

P#	
P1	300
P2	800
P3	500
P4	700
P6	500

شماره قطعاتی را بیابید که یا وزن آنها بیشتر از 16 باشد یا توسط S2 تهیه شده است یا هر دو شرط را دارد.

```
SELECT  P#  FROM  P   WHERE  WEIGHT > 16
UNION
SELECT  P#  FROM  SP  WHERE  S# = 'S2' ;
```

نتیجه برابر است با :

P#
P1
P2
P3

فصل پنجم

وابستگی ها

انواع وابستگی ها

انواع وابستگی ها عبارتند از:

۱- وابستگی تابعی(FD)

۲- وابستگی تابعی کامل(FFD)

۳- وابستگی با واسطه

۴- وابستگی تابعی چند مقداری(MVD)

۵- وابستگی پیوندی(JD)

FD : Function Dependency

MVD : Multi Valued Dependency

FFD: Full Function Dependency

JD : Join Dependency

وابستگی تابعی

رابطه R (A , B , ...) را در نظر بگیرید. می گوییم B با A وابستگی تابعی(FD) دارد و نشان می دهیم A→B ، اگر و فقط اگر در هر مقدار ممکن از متغیر رابطه R ، به هر مقدار A فقط یک مقدار B متناظر باشد.

در رابطه R(A,B,C) که در زیر آورده شده است، FD ها را مشخص کنید.

A	B	C
A1	B2	C1
A1	B1	C2
A2	B1	C3
A2	B2	C1

حل: کلید رابطه (A,B) است، بنابراین وابستگی (A,B)→C وجود دارد. همچنین مشخص است که وابستگی تابعی C→B نیز در رابطه وجود دارد، یعنی با معلوم بودن مقدار C می توان مقدار B را مشخص کرد و جواب یکتا است. مثلا مقدار متناظر با C1 همواره B2 است. دقت شود که وابستگی B→C وجود ندارد، چون مثلا مقدار متناظر با B1 یکتا نیست.

وابستگی تابعی کامل (FFD)

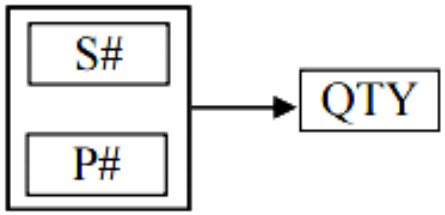
اگر X و Y دو زیر مجموعه از مجموعه عنوان رابطه R باشند، می گوییم Y با X وابستگی تابعی کامل دارد و نشان می دهیم X⇒Y ، اگر و فقط اگر Y با X وابستگی تابعی (FD) داشته باشد ولی با هیچ زیر مجموعه از X وابستگی تابعی نداشته باشد. بدیهی است اگر سمت چپ FD صفت ساده باشد، وابستگی FFD خواهد بود.

وابستگی ها

مثال

در رابطه SP (S# , P# , QTY) ، صفت QTY با صفت مرکب(S# , P#) وابستگی تابعی کامل دارد، یعنی وابستگی تابعی (S# , P#)→QTY وجود دارد، در حالیکه هیچ کدام از وابستگی های S#→QTY و P#→QTY برقرار نمی باشد.

S#	P#	QTY
S1	P1	100
S2	P2	400
S3	P6	100
S4	P2	300
S4	P5	400



وابستگی با واسطه

رابطه R (A,B,C) مفروض است. اگر صفت B با صفت A ، FD داشته باشد $(A \rightarrow B)$ و صفت C نیز با صفت B , FD داشته باشد $(B \rightarrow C)$، ولی A با B، FD نداشته باشد، می گوییم C باA ، وابستگی با واسطه دارد. برای از بین بردن این وابستگی، رابطه را به دو رابطه زیر تجزیه می کنیم:

R1 (A,B)
R2 (B,C)

وابستگی ها

قواعد استنتاج آرمسترانگ

با فرض اینکه A , B , C , D زیر مجموعه هایی از صفات رابطه R باشند، قواعد زیر برقرارند:

انعکاسی	اگر $B \subseteq A$ آنگاه $A \rightarrow B$
تعدی (تراگذاری)	اگر $A \rightarrow B$ و $B \rightarrow C$ آنگاه $A \rightarrow C$
افزایش	اگر $A \rightarrow B$ آنگاه $AC \rightarrow BC$
تجزیه	اگر $A \rightarrow BC$ آنگاه $A \rightarrow B$ و $A \rightarrow C$
اجتماع	اگر $A \rightarrow B$ و $A \rightarrow C$ آنگاه $A \rightarrow BC$
ترکیب	اگر $C \rightarrow D$ و $A \rightarrow B$ آنگاه $AC \rightarrow BD$
شبه تعدی	اگر $CB \rightarrow D$ و $A \rightarrow B$ آنگاه $AC \rightarrow D$

وابستگی ها

مثال

$R = \{ S\#, CITY, STATUS \}$

$F = \{ S\# \rightarrow CITY, CITY \rightarrow STATUS, S\# \rightarrow STATUS \}$

حل:

$S\# \rightarrow CITY$

$CITY \rightarrow STATUS$

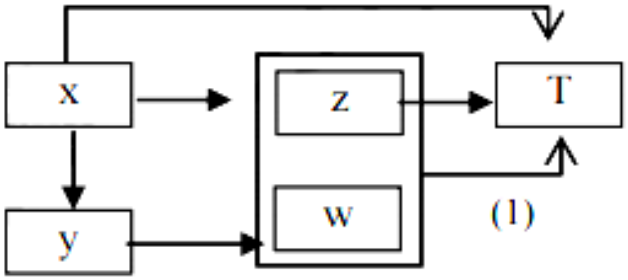
$\Rightarrow S\# \rightarrow STATUS$

$F = \{ S\# \rightarrow CITY, CITY \rightarrow STATUS \}$

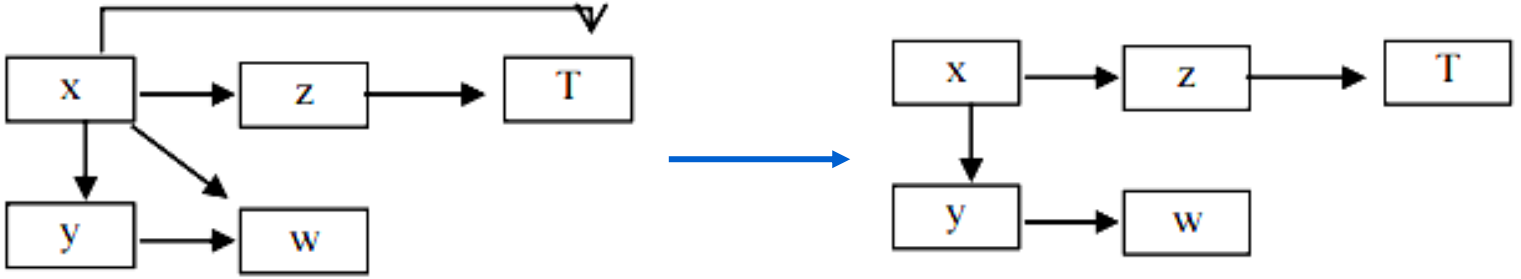
وابستگی ها

مثال

نمودار حاصل پس از حذف وابستگی های نمودار زیر را بدست آورید.



حل:



وابستگی ها

پیدا کردن کلید کاندید

مثال

کلید کاندید رابطه زیر را بدست آورید.

$R = (A , B, C , D , E , F, G)$

$F = \{AF \rightarrow BE , FC \rightarrow DE , F \rightarrow CD , D \rightarrow E , C \rightarrow A \}$

وابستگی ها

حل:

$AF \rightarrow BE \quad \Rightarrow \quad AF \rightarrow B \, , \, AF \rightarrow E$

$FC \rightarrow DE \quad \Rightarrow \quad FC \rightarrow D \, , \, FC \rightarrow E$

$F \rightarrow CD \quad \Rightarrow \quad F \rightarrow C \, , \, F \rightarrow D$

$F \rightarrow C \, , \, FC \rightarrow D \quad \Rightarrow \quad F \rightarrow D$

$F \rightarrow C \, , \, FC \rightarrow E \quad \Rightarrow \quad F \rightarrow E$

$F \rightarrow C \, , \, C \rightarrow A \quad \Rightarrow \quad F \rightarrow A$

$F \rightarrow A \, , \, AF \rightarrow B \quad \Rightarrow \quad F \rightarrow B$

$F \rightarrow A \, , \, AF \rightarrow E \quad \Rightarrow \quad F \rightarrow E$

وابستگی ها

مثال

کلید کاندید رابطه زیر را بدست آورید؟

$$R = (S, T , U , V , W)$$

$$F = \{ S \rightarrow T , V \rightarrow SW , T \rightarrow U \}$$

وابستگی چند مقداری (MVD)

در رابطه(R (X , Y , Z با صفات ساده یا مرکب X , Y , Z می گوییم که Y با X وابستگی تابعی چند مقداری دارد و نمـایش می دهیم $X \rightarrow\rightarrow Y$ ، اگر به یک مقدار X ، مجموعه ای از مقادیر Y متناظر باشد.

R

X	Y	Z
X1	Y1 Y2 Y3	Z1
X1	Y1 Y2 Y3	Z2

وابستگی ها

قواعد آرمسترانگ در مورد وابستگی چند مقداری

در رابطه $R(A, B, C, \ldots)$:

۱- اگر $A \to B$ ، آنگاه $A \to\to B$

۲- اگر $A \subset B$ ، آنگاه $A \to\to B$

۳- اگر $A \to B$ ، آنگاه $AC \to\to BC$

۴- اگر $A \to\to B$ ، آنگاه $A \to\to R(H) - B - A$

۵- اگر $A \to\to B$ و $A \to\to C$ ، آنگاه $A \to\to BC$

۶- اگر $A \to\to B$ و $B \to\to C$ ، آنگاه $A \to\to C-B$

۷- اگر $A \to\to B$ و $A \to\to C$ ، آنگاه $C \to\to B \cap A$

۸- اگر $A \to\to B$ و $A \to\to C$ ، آنگاه $A \to\to (C - B)$ ، $A \to\to (B - C)$

۹- اگر $B \subset C$ ، $D,A \to\to B$ آنگاه $(A,D) \subset (B,C)$

وابستگی ها

وابستگی پیوندی (JD)

رابطه R وابستگی پیوندی به n پرتوش دارد، اگر و فقط اگر R حاصل پیوند n پرتوش باشد و نه کمتر. این وابستگی را به صورت $R = JD^*(R_1, R_2, \ldots, R_n)$ نمایش می دهیم که $R_1 \ldots R_n$ پرتوهای رابطه R می باشند.

مثال

رابطه SPJ (SP , PJ , JS) ، وابستگی پیوندی به ۳ پرتوش دارد و به صورت SPJ = JD * (SP , PJ , JS) نمایش داده می شود.

در واقع اگر پرتوهای این رابطه را یک بار روی صفات P# , S# و بار دیگر روی صفات خاصه P# , J# بدست آوریم و آنها را با یکدیگر JOIN کنیم و سپس نتیجه را با رابطه حاصل از پرتو روی صفات S# و J# و , Join کنیم، حاصل همان SPJ خواهد بود و هیچ سطری اضافه یا کم نخواهد شد.

وابستگی ها

SP

S#	P#	J#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

⇒

PJ

SPJ

S#	P#
S1	P1
S1	P2
S2	P1

P#	J#
P1	J2
P2	J1
P1	J1

⇒

S#	P#	J#
S1	P1	J2
S1	P1	J1
S1	P2	J1
S2	P2	J2
S2	P1	J1

وابستگی ها

S#	P#	J#
S1	P1	J2
S1	P1	J1
S1	P2	J1
S2	P2	J2
S2	P1	J1

J#	S#
J2	S1
J1	S1
J1	S2

S#	P#	J#
S1	P1	J2
S1	P2	J1
S2	P1	J1
S1	P1	J1

فصل ششم

نرمال سازی

نرمال سازی

آنومالی

دشواری و وضع غیرعادی را آنومالی می گویند. مثلاً وقتی که سطری را حذف می کنیم و پی آمد آن اطلاعات ناخواسته ای نیز حذف شود. یا مقدار صفتی را برای یک سطر تغییر می دهیم در حالیکه در سطرهای دیگر نیز امکان دارد نیازبه تغییر داشته باشد. در واقع آنومالی در عملیات ذخیره سازی به هر یک از سه حالت زیر گفته می شود:

۱- بروز پیامد بد، بعد از انجام یک عمل

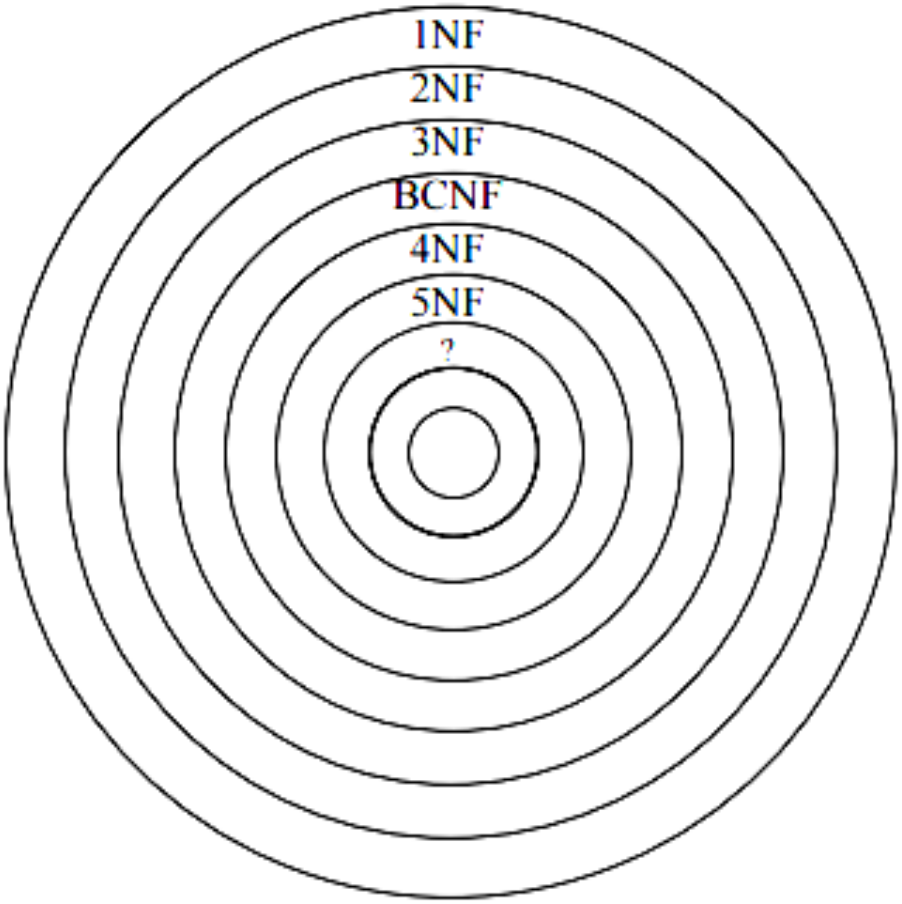
۲- عدم امکان انجام یک عمل

۳- بروز اضافه کاری در انجام یک عمل

S#	STATUS	CITY	P#	QTY
S1	20	C2	P1	300
S1	20	C2	P2	200
S2	10	C3	P1	300
S3	10	C3	P2	200
S4	20	C2	P5	400

نرمال سازی

صورت های نرمال (NORMAL FORMS)



تعریف صورتهای نرمال

1NF	هر صفت خاصه در هر تاپل، تک مقداری باشد.
2NF	هر صفت خاصه غیرکلید با کلید اصلی، وابستگی تابعی کامل داشته باشند.
3NF	هر صفت خاصه غیر کلید با کلید اصلی، وابستگی تابعی بی واسطه داشته باشد.
BCNF	هر دترمینان، کلید کاندید باشد.
4NF	وابستگی تابعی چند مقداری وجود نداشته باشد.
5NF	تمام وابستگی های پیوندی، ناشی از کلیدهای کاندید باشد.

نرمال سازی

مزایای نرمالترسازی

۱- کاهش بعضی از آنومالی ها

۲- کاهش بعضی از انواع افزونگی

۳- تسهیل اعمال بعضی از قواعد جامعیت

۴- ارائه یک طرح بهتر و واضح تر با کمترین اختلاط اطلاعات

معایب نرمالترسازی

۱- بروز فزونکاری در سیستم در عمل بازیابی

۲- ایجاد نوعی افزونگی

۳- زمانگیر بودن فرایند نرمالترسازی

۴- مشکل شدن تصمیم گیری ها در تعدد تجزیه ها در مواردی

نرمال سازی

رابطه 1NF

هر رابطه نرمالی1NF است، اما رابطه1NF ای که2NF نیست دارای آنومالی هایی می باشد. به طور نمونه جدول FIRST را در نظر می گیریم:

FIREST (S# , P# , STATUS , CITY , QTY)

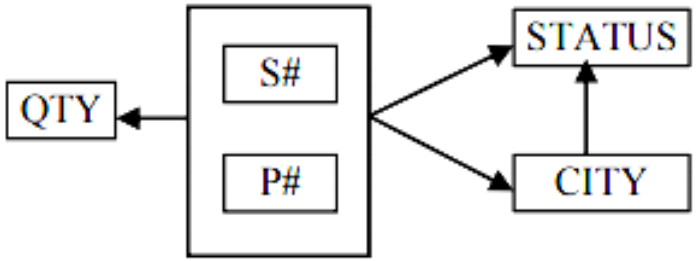
کلید اصلی این رابطه، صفت مرکب (S#, P#) است و فرض کرده ایم که وابستگی CITY → STATUS وجود داشته باشد. یعنی وضعیت یک تهیه کننده از طریق شهر او، تعیین می شود. قبلاً دیدیم که این رابطه در درج، حذف و بهنگام سازی دارای آنومالی می باشد که برای رفع این آنومالی ها باید آن را به دو رابطه تجزیه کرد.

SP (S# , P# , QTY) و SECOND (S# , STATUS , CITY)

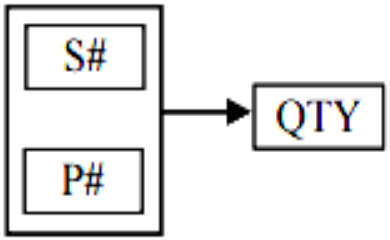
نرمال سازی

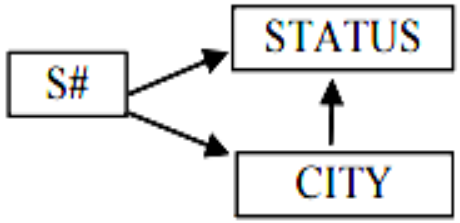
با بررسی رابطه های SECOND وSP مشخص می شود که آنومالی های موجود درFIRST در آنها وجود ندارند، یعنی در سطح نرمالتری از FIRST قرار دارند.

قبل از تجزیه:



بعد از تجزیه:





رابطه 2NF

رابطه های حاصل از تجزیه FIRST هر دو در صورت دوم نرمال قرار دارند. رابطه SECOND هنوز دارای آنومالی هایی می باشد و علت این مشکلات در این است که وابستگی STATUS به S# از یک سو کامل است و از سوی دیگر STATUS از طریق CITY نیز با S# وابستگی دارد.

SC (S# , CITY)
CS (CITY , STATUS)

رابطه 3NF

ابتدا FIRST را به دو رابطه SECOND و SP تجزیه کردیم. رابطه SECOND در صورت 3NF نبود، چون صفت غیر کلید STATUS با کلید اصلی S# ، وابستگی باواسطه داشت و این برخلاف تعریف صورت 3NF است. به همین علت آن را به دو رابطه تجزیه کردیم. اما رابطه SP در صورت سوم نرمال است. چون صفت غیر کلید QTY با کلید اصلی (S# , P#) وابستگی باواسطه ندارد.

نرمال سازی

مثال

رابطه R را در نظر می گیریم:

R (SID , CID , SNAME , GRADE , SMJR , SDEID)

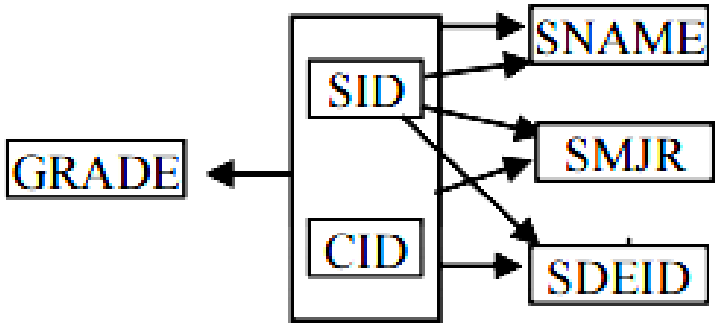
قواعد جامعیت این رابطه عبارتند از:

۱- هر دانشجو یک نام دارد.

۲- هر دانشجو در یک درس یک نمره دارد.

۳- هر دانشجو در یک رشته تحصیل می کند.

۴- هر رشته تحصیلی در یک گروه آموزشی وجود دارد.



R1 (SID , SNAME , SMJR , SDEID) , **R2** (SID , CID , GRADE)

R3 (SID , SNAME , SMJR) , **R4** (SMJR , SDEID)

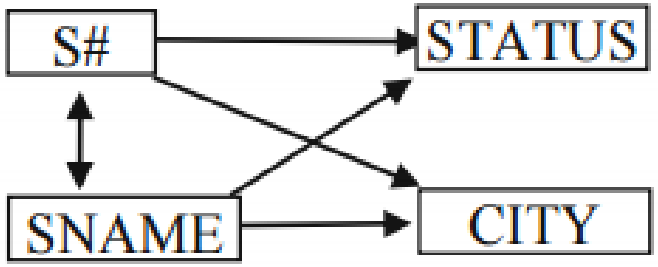
رابطه BCNF

رابطه R در سطح BCNF است اگر و فقط اگر هر دترمینان آن، کلید کاندید باشد.(هر صفت ای که صفت دیگر با آن وابستگی تابعی کامل داشته باشد دترمینان نام دارد. یعنی در وابستگی $A \rightarrow B$ ، صفتA دترمینان است.)

هر رابطه3NF، BCNF نیست. اگر رابطه تنها یک کلید کاندید داشته باشد و در مجموعه وابستگی های تابعی کاهش ناپذیر آن، وابستگی تابعی دیگری غیر از وابستگی های ناشی از کلید کاندید وجود نداشته باشد آنگاه رابطه BCNF هم هست. اگر رابطه3NF بیش از یک کلید کاندید داشته باشد و کلیدهای کاندید صفت مشترک نداشته باشند رابطه BCNF هم هست و اگر صفت مشترک داشته باشند ممکن است BCNF نباشد.

مثال

رابطه **S** (S# , SNAME , STATUS , CITY) با دو کلید کاندید S# , SNAME در فرم BCNF است. چون صفات S# , SNAME علاوه بر اینکه دترمینان های رابطه هستند، کلیدهای کاندید آن نیز می باشند.



رابطه 4NF

در رابطهBCNF نیز احتمال وجود آنومالی خواهد بود. در این حالت رابطه را باید نرمالتر کرد.

مثال

رابطه غیرنرمال R1 (A,B,C) مفروض است:

A	B	C
A1	B1 B2	C1 C2
A2	B1	C1 C3 C4

نرمال سازی

A	B	C
A1	B1	C1
A1	B1	C2
A1	B2	C1
A1	B2	C2
A2	B1	C1
A2	B1	C3
A2	B1	C4

⇒

(A2 , B3 , C1)
(A2 , B3 , C2)
(A2 , B3 , C3)

⇒

R1

A	B
A1	B1
A1	B2
A2	B1

و

R2

A	C
A1	C1
A1	C2
A2	C1
A2	C3
A2	C4