جزوه درس پایگاه داده ها

Database
Managment

بابک آشفته یزدی

پایگاه داده:

(تعاریف اولیه)

① Data **(داده):**

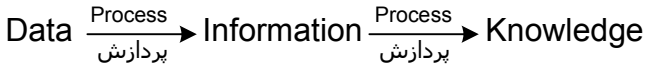
اطلاعاتی که بصورت عام معنی خاصی ندارد
ورودی بانک اطلاعاتی

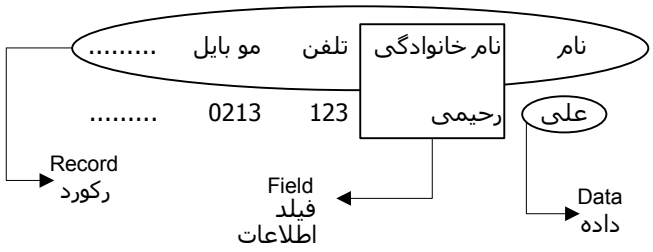
② Information **(اطلاعات):**

به داده پردازش شده گویند.
معنای داده از دید کاربر

③ Knowledge **(دانش):**

به اطلاعات پردازش شده گفته می شود.
به مجموعه عظیمی (بزرگی) از اطلاعات دانش گفته میشود(پایگاه دانش) مانند سیستم خبره پزشکی یا کتابخانه





(جزئیات بانک اطلاعاتی)

① Field **(فیلد):**

به کوچکترین عنصر بانک اطلاعاتی گفته میشود
به هر ستون داده ها گفته میشود

② Record **(رکورد):**

به مجموعه ای از فیلد های مرتبط با هم رکورد گویند.
به هر سطر داده ها گفته میشود

③ Table **(جدول):**

به مجموعه ای از رکورد ها گفته میشود.
جدول به نوعی فایل می باشد.

④ Database **(بانک اطلاعاتی):**

به مجموعه ای از جداول مرتبط با هم بانک اطلاعاتی گفته میشود.

⑤ Super key - Primary key **(کلید اولیه):**

فیلد های منحصر به فرد (یکتا) Unique

⑥ Combination key **(کلید ترکیبی):**

اگر بیش از یک کلید با هم ترکیب شود کلید جدید ترکیبی می باشد.

⑦ Forign key **(کلید خارجی):**

فیلدی که در جدول خود کلید اولیه بوده، وقتی در جدول دیگر فراخوانی میشود کلید خارجی گویند.

در این حالت کلید منحصر به فرد نیست
حداقل دردو جا مورد استفاده قرار می گیرد.

مثال کتابخانه

Library

جدول کتاب

Book Table

Primary key

کد کتاب	نام کتاب	مولف	قطع	ناشر	قیمت	تیراژ .....
1	پاسکال	اکبری	وزیری	ققنوس	123	500 .....

جدول اعضاء

member Table

Primary key

کد عضویت	نام	نام خانوادگی	نام پدر	ش ش	رشته تحصیلی
1	علی	اکبری	مراد	23	ریاضی .....

جدول ارتباط

Relation Table

Forign key

کد عضویت	کد کتاب	تاریخ خروج کتاب	تاریخ برگشت .....

فیلد اطلاعات ثابت در جداول دارای کلید اولیه یکبار ثبت می شود.
فیلد اطلاعات متغییر در جدول رابطه درج میگردد

انتخاب واحد در دانشگاه

جدول اساتید

Primary key

کد استاد ،نام،نام خانوادگی،تلفن همراه،وضعیت تاهل،رشته تحصیلی،کد رشته تحصیلی،.....

جدول دروس

Primary key

کد درس،نام درس،تعداد واحد،مبلغ شهریه

جدول دانشجویان

Primary key

کد دانشجو ،نام،نام خانوادگی،تلفن همراه،وضعیت تاهل،مدرک تحصیلی،کد رشته تحصیلی،سال ورودی،تاریخ تولد-آدرس.....

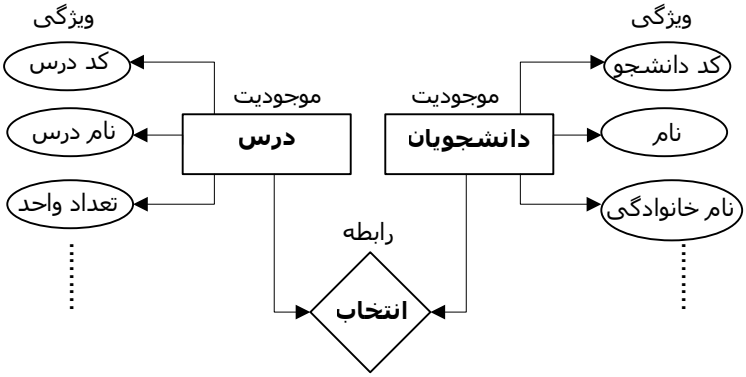
جدول ارتباط

Relation Table

Forign key

کد دانشجو،کد درس،کد استاد،ترم تحصیلی،تاریخ امتحان،ساعت تشکیل کلاس،شماره کلاس.....

(نمودار ER)



1 موجودیت 2 ویژگی،خصوصیات 3 رابطه

طراحی سیستم ارتباطی

1 Single user سیستم تک کاربره

2 Multi user سیستم چند کاربره

کاربران بانک اطلاعاتی

1 برنامه نویسان کاربردی:
تجزیه و تحلیل و برنامه نویسی بانک اطلاعاتی

2 کاربران نهایی:
استفاده کننده ها Users

3 مدیران بانک اطلاعاتی: DBA
Database Administrator

انواع سیستمها

1 DBMS سیستم مدیریت پایگاه داده
Database Management System

یک نرم افزار سیستم است که کاربران را قادر به تعریف ،ایجاد،نگهداری و مدیریت یک بانک اطلاعاتی می کندو امکان دسترسی کنترل شده به این بانک اطلاعاتی را فراهم می آورد.

2 Filing system سیستمهای فایلینگ

-اطلاعات بصورت فایلهای متعدد بر روی سیستم ذخیره می شود ،دسترسی اطلاعات در این حالت بصورت ترتیبی است
- سرعت کند است
- تکرار داده زیاد است
- ناسازگاری داده ها درون فایلها:مثل قرار دادن فیلم در فایل TXT
- پرس و جو ثابت است (پیشرفته نیست)و المانهای زیادی را در بر نمی گیرد.
- وابستگی به محیط پیاده سازی و زبان برنامه نویسی دارد.

دو خصوصیت(مزیت)بانک اطلاعاتی نسبت به فایلینگ

1 مجتمع بودن: Normalize
جامع بودن داده ها یعنی داده تکراری در بانک اطلاعاتی وجود ندارد.

2 اشتراکی بودن داده ها:
به اشتراک گذاشتن اطلاعات بانک برای چندین کاربر

روشهای دستیابی به اطلاعات

1 ترتیبی:
سیستم فایلینگ

2 تصادفی(مستقیم):
هارد دیسک،فلش .....

مدیریت پایگاه داده ها
DBMS

1 DDL زبان تعریف داده ها
Data definition language

- تعریف پایگاه داده،ایجاد پایگاه داده
Create

2 DML زبان دستکاری داده ها
Data Manipulation language

- وارد کردن،ویرایش،جستجو،به هنگام سازی،بازیابی اطلاعات و کلا هرگونه تغییر در بانک داده
Select-Delete-Insert-Update

دیدگاه
View

- دیدگاه زیر مجموعه ای از داده های بانک اطلاعاتی را به نمایش میگذارد.

مزایای دیدگاه
View benefits

- کاهش پیچیدگی دیدن داده ها طوریکه مد نظر کاربران است.
- فراهم آوردن یک سطح از امنیت بخاطر عدم دسترسی مستقیم به پایگاه داده ها.
- امکان داشتن تصویری ثابت و سازگار که مستقل از تغییرات پایگاه داده ها است.

نقش ها در بانک اطلاعاتی

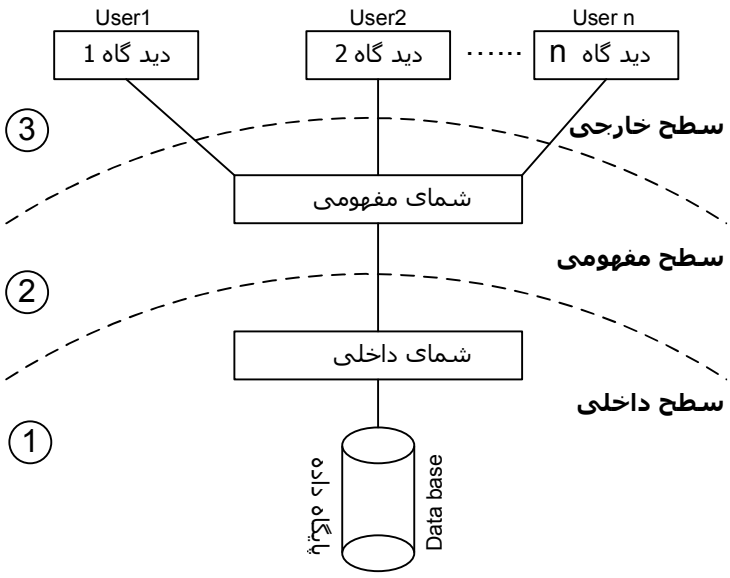
1 DA (مدیر داده ها): Data Administration

- مسئول مدیریت منابع داده شامل طراحی،توسعه،نگهداری استانداردها ،خط مشی ها،طراحی مفهومی و منطقی پایگاه داده
(تجزیه و تحلیل)

2 DBA (مدیر بانک اطلاعاتی): Database Administration

- مسئول تحقق فیزیکی پایگاه داده شامل طراحی و پیاده سازی فیزیکی ،کنترل یکپارچگی وامنیت ،نگهداری سیستم عامل و اطمینان از کارایی بانک اطلاعاتی برای کاربران
(پیاده سازی و اجراء)

معماری سه سطحی
Ansi-Spare



- تشریح کلی پایگاه داده را شمای بانک اطلاعاتی گویند.

1 سطح داخلی

- همان سطح فیزیکی است که به چگونگی ذخیره داده در بانک اطلاعاتی اشاره می کند و پایین ترین سطح انتزاع است.مثلا جزئیات فیلد که چه نوعی است ،چند کاراکتر است و.....

2 سطح مفهومی

- اجتماع دیدگاه کاربران از بانک اطلاعاتی است و تشریح می کند که چه داده هایی در بانک اطلاعاتی ذخیره شده اند و رابطه بین آنها چیست (رابطه بین جداول مختلف)

3 سطح خارجی

- همان دیدی است که کاربر از بانک اطلاعاتی دارد و بالاترین سطح انتزاع است.

رابطه بین جداول
Relation type

1 One to One (رابطه یک به یک) 1 - 1

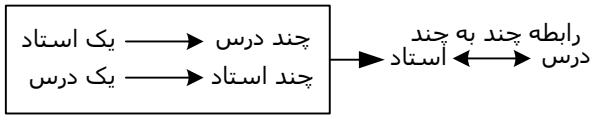
- یک فیلد از یک جدول با یک فیلد از جدول دیگر رابطه دارد.
- مثال:رابطه کد ملی کارمند با خودش یا شخص با کد عضویت

2 One to Many (رابطه یک به چند) 1 -∞

- در یک جدول ثبت شده و چندین بار در جداول دیگر تکرار میشود.
- مثال:رابطه یک شخص به چندین شماره حساب

3 Many to Many(رابطه چند به چند) ∞ - ∞

- چند فیلد در یک جدول با چند فیلد در جدول دیگر ارتباط دارند.
- مثال:چند استاد چند درس را تدریس مکنند.



مدل داده و اجزاء آن
Data Model

- مجموعه یکپارچه ای از مفاهیم برای تشریح داده ها،روابط آنها و محدودیت بر روی داده ها در یک سازمان است.

1 بخش ساختاری:

- شامل قوانینی که بر اساس آن بانک اطلاعاتی ساخته می شود.

2 بخش اعمال تعریف شده:

- بطور کلی اعمالی(تغییرات) که بر روی داده ها تعریف(انجام) میگردد را شامل میشود.

3 بخش یکپارچه سازی و تضمین آن:

- مجموعه ای از قوانین یکپارچه سازی که تضمین می کند که داده ها دقیق هستند.

انواع مدل داده
Data Model

1 مدل داده شی گرا:

2 مدل داده مبتنی بر رکورد:

- مدل داده رابطه ای ، مدل داده شبکه ای ، مدل داده سلسله مراتبی

Table regard
مدل داده رابطه ای

- رابطه فیزیکی (بین جداول از طریق فیلدها) Relationship

- هر رابطه دارای یک درجه است (1 ∞)

DBMS Services
خدمات سیستم مدیریت پایگاه داده ها

تراکنش:

- مجموعه ای از دستورات، دستکاری و یا تعریف داده ها را شامل می شودکه DBMS تضمین میکند که یا همه دستورات و یا هیچکدام آن اجراء شوند.

1 ذخیره ،بازیابی و بهنگام سازی داده ها

2 کاتالوگ قابل دسترس برای کاربر

3 پشتیبانی از تراکنش

- جلوگیری از بروز نا سازگاری در بانک اطلاعاتی

4 کنترل همزمانی

- تضمین به هنگام سازی در بانک اطلاعاتی وقتی چندین کاربر بطور همزمان به یک مجموعه از داده ها دسترسی دارند.

5 سرویسهای ترمیم

- امکان ترمیم Recovery پایگاه داده ها در هنگام بروز اشکال یا از دست دادن داده ها در اثر هر عاملی

6 سرویسهای اعتبار سنجی

- فقط کاربران مجاز قادر به دستیابی اطلاعات هستند.

7 پشتیبانی از ارتباطات داده ای

- امکان تبادل داده با نرم افزارهای ارتباطی

8 سرویسهای یکپارچگی

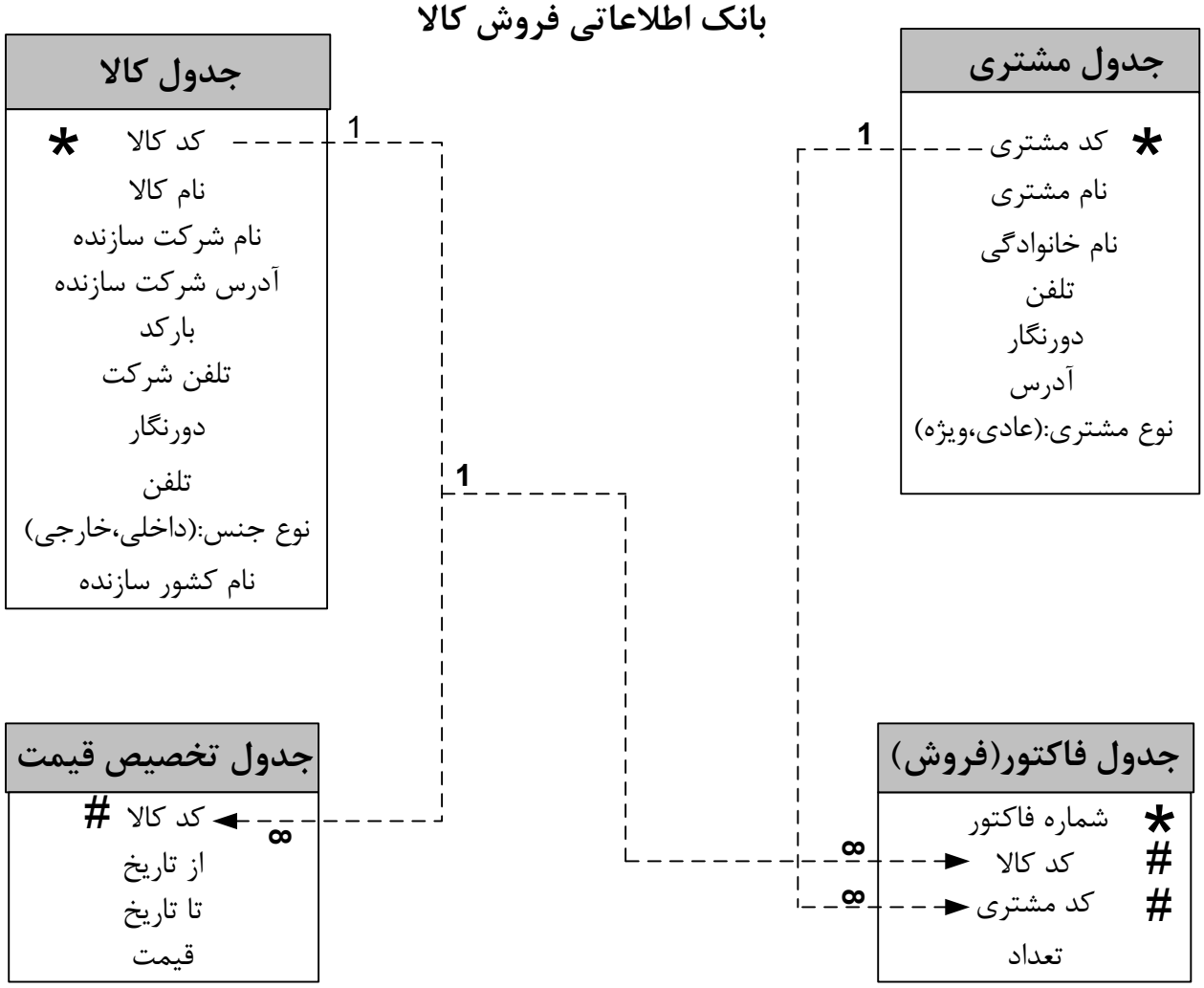
- هم داده و هم تغییرات آنها ازقوانین خاصی پیروی کنند.

9 امکان استقلال داده ها

- عدم اثر داده ها بر روی یکدیگر

10 فراهم آوردن نرم افزارهای کمکی کاربر بانک اطلاعاتی

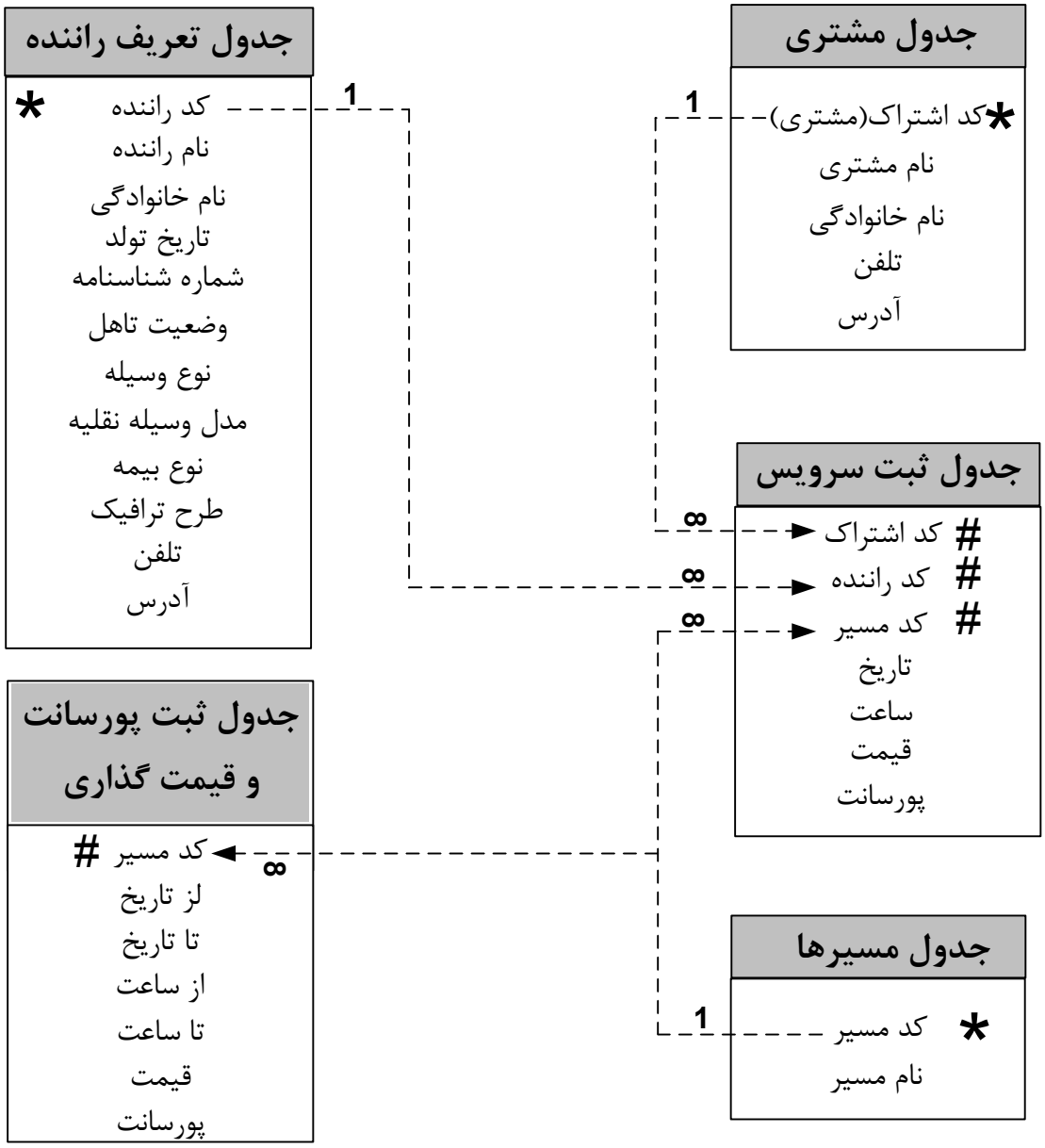
بانک اطلاعاتی فروش کالا



* Primary key

Forign key

بانک اطلاعاتی تاکسی تلفنی



---

* Primary key
# Forign key

عناصر پایگاه داده ها

1 سخت افزار

- سخت افزار ذخیره سازی داده ها(حافظه جانبی و اصلی)
- سخت افزار پردازشگر(کامپیوتر)
- سخت افزار ارتباط (امکانات محلی-امکانات شبکه ای)

2 نرم افزار

- سیستم مدیریت پایگاه داده ها
- برنامه های کاربردی

3 کاربر

4 داده

مدل سازی معانی داده ها

اجزاء نمودار ER

1 موجودیت

2 صفت یا ویژگی هر موجودیت

3 نوع ارتباط

1

موجودیت

هدف موجودیت:نوع موجودیت عبارت است از مفهوم کلی شی و بطور کلی هر آنچه می خواهیم در موردش اطلاع داشته باشیم و شناخت خود را در موردش افزایش دهیم ،هر نوع موجودیت از نظر کاربر نام و معنای مشخص دارد.

ضابطه در تشخیص یک نوع موجودیت:

- یک نوع موجودیت از یک محیط معمولا نمونه های متمایز از یکدیگر دارد.
- یک نوع موجودیت معمولا بیش از یک صفت دارد و کاربر به مجموعه ای از اطلاعات در مورد آن نیاز دارد.
- معمولا حالت فاعلیت یا مفعولیت دارد

موجودیت
- قوی (مستقل)
  - موجودیتی است که مستقل از هر نوع موجودیت دیگر و به خودی خود در یک محیط مشخص مطرح باشد.
- ضعیف (وابسته)
  - موجودیتی است که وجودش وابسته به یک نوع موجودیت دیگر است به گونه ای که اگر موجودیت قوی حذف شود آن موجودیت نیز حذف میشود.

2

صفت یا ویژگی هر موجودیت

صفت ساده یا مرکب:

- صفت ساده صفتی است که مقدار آن از لحاظ معنایی ساده یا تجزیه نشدنی باشد.به این معنا که اگر مقدار آن را به اجزاء تجزیه کنیم مقادیر حاصله فاقد معنی باشد.

- صفت مرکب صفتی است که از چند صفت ساده تشکیل شده باشد به گونه ای که تجزیه شدنی باشد و اجزاء حاصل از تجزیه خود صفات ساده باشند.

صفت تک مقداری:

- صفتی است که برای یک نمونه از یک نوع موجودیت حداکثر یک مقدار از دامنه مقادیر را می گیرد مانند شماره درس-شماره دانشجویی

صفت شناسه:

- صفتی است که دو ویژگی داشته باشد الف) یکتایی مقدار داشته باشد یعنی در هیچ دو نمونه از یک نوع موجودیت مقدارش یکسان نباشد ب)حدالمکان طول مقادیرش کوتاه باشد.

صفت چند مقداری:

- صفتی است که برای بعضی یا همه نمونه های نوع موجودیت بیش از یک مقدار از دامنه مقادیر را می گیرد مانند مدرک اسناد-شماره تلفن دانشگاه

صفت هیچ مقدار پذیر یا ناپذیر:

- اگر فیلدی مقدار تهی را بپذیرد آن صفت هیچ مقدار پذیر است و اگر صفتی تهی را نپذیرد هیچ مقدار ناپذیر است.

صفت ذخیره شده واقعی یا مشتق:

- صفت واقعی صفتی است که مقادیرش در پایگاه داده ها ذخیره شده باشد و البته می تواند هیچ مقدار نیز داشته باشد اگر شناسه نباشد.

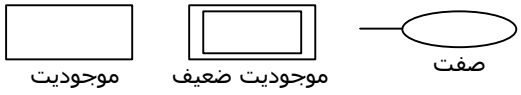
- صفت مشتق صفتی است که مقادیرش در پایگاه داده ها ذخیره شده نباشد بلکه حاصل یک پردازش روی داده های ذخیره شده باشد مانند معدل دانشجو

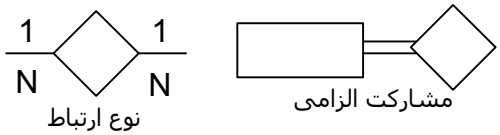
تمرین:محیط عملیاتی کتابخانه را صفاتش را استخراج کرده وبرای هر کدام دو مثال بزنید.

3

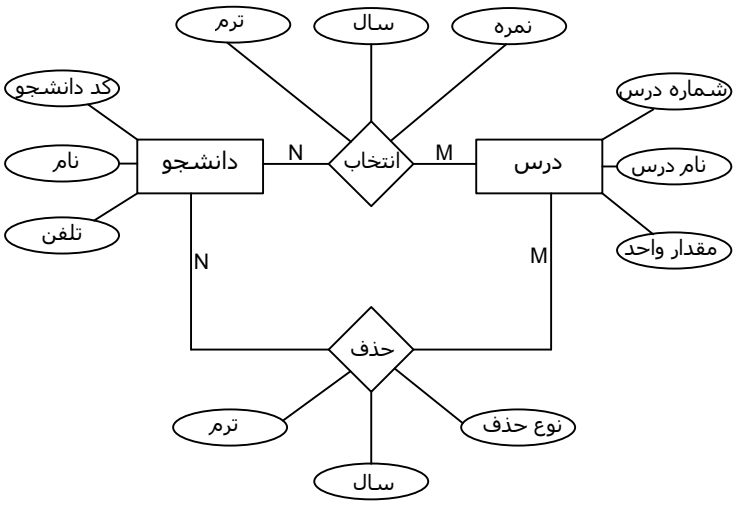
ارتباط

نوع ارتباط عبارتست از تعامل بین دو یا بیش از دو نوع موجودیت.





نمودارERاولین طرح پایگاه داده ها و معدل کلی آن است در بالاترین سطح انتزاع،هر نمودار ER پاسخگوی مجموعه مشخص از نیازهای کاربران است.



خصوصیات نوع ارتباط

الف) هر ارتباط یک نام دارد

ب) هر نوع ارتباط یک معنی مشخص داردو این معنا با معانی هر نوع ارتباط دیگر متفاوت است.

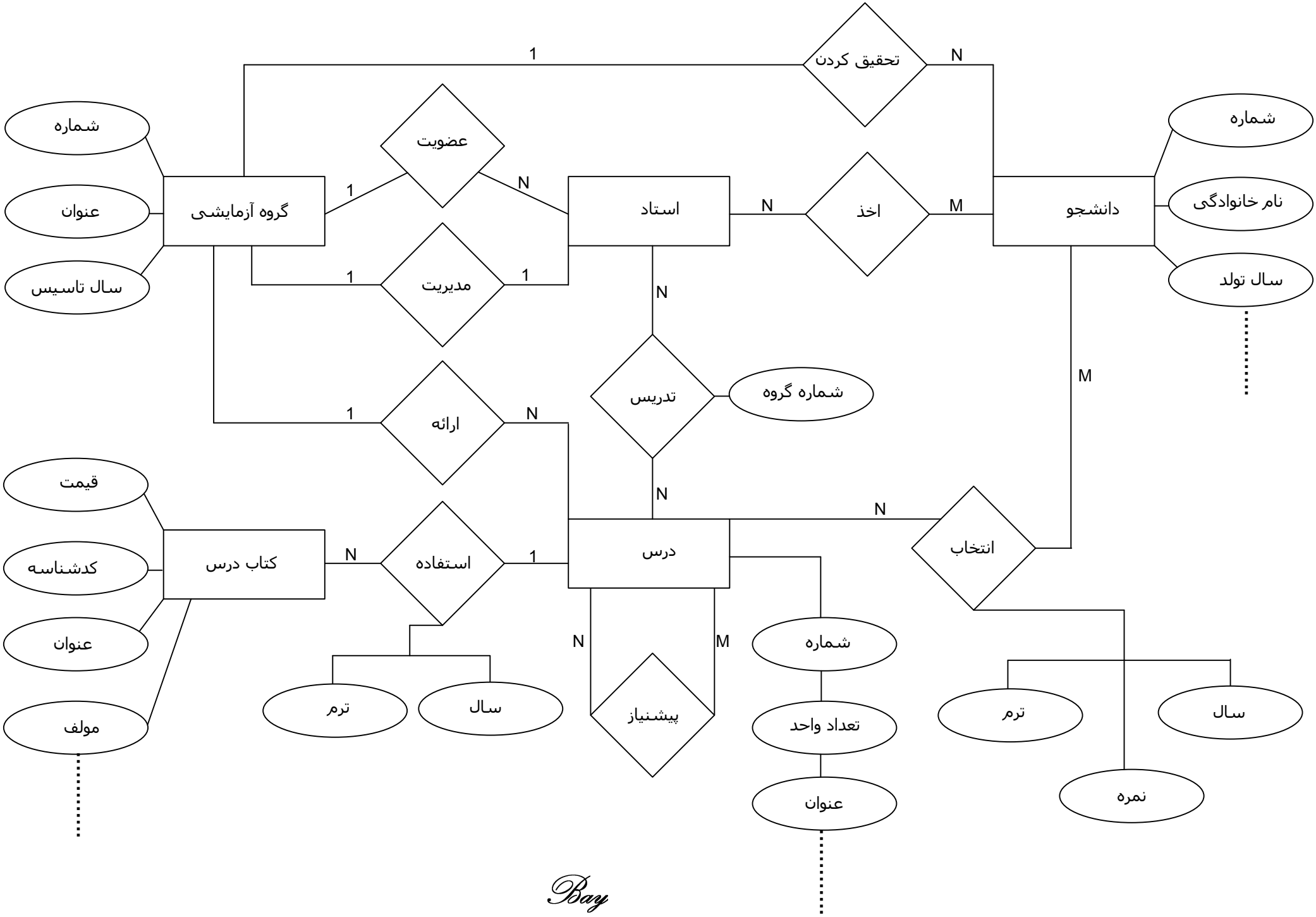
ج) هر نوع ارتباط نمونه هایی دارد

وضع مشارکت در ارتباط

1-الزامی (کامل)

2- غیر الزامی (غیر کامل)

مشارکت یک نوع موجودیت در یک نوع ارتباط را الزامی می گوییم اگر تمام نمونه های آن نوع موجودیت در آن نوع ارتباط شرکت کند در غیر اینصورت مشارکت غیر الزامی است.


تحقیق کردن
1
N
شماره
عنوان
سال تاسیس
گروه آزمایشی
عضویت
1
N
مدیریت
1
1
استاد
N
اخذ
M
دانشجو
شماره
نام خانوادگی
سال تولد
N
تدریس
شماره گروه
M
1
ارائه
N
N
قیمت
کدشناسه
عنوان
مولف
کتاب درس
N
استفاده
1
درس
N
انتخاب
N
ترم
سال
N
پیشنیاز
M
شماره
تعداد واحد
عنوان
ترم
سال
نمره
Bay
www.freebay.ir

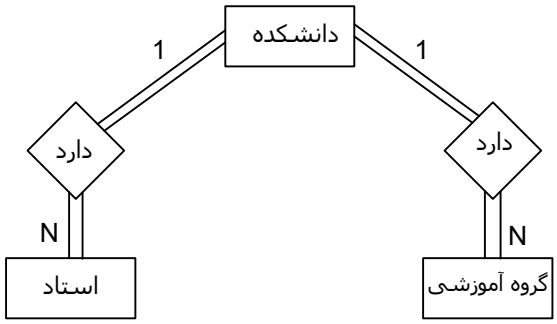
مشکلات روش ER

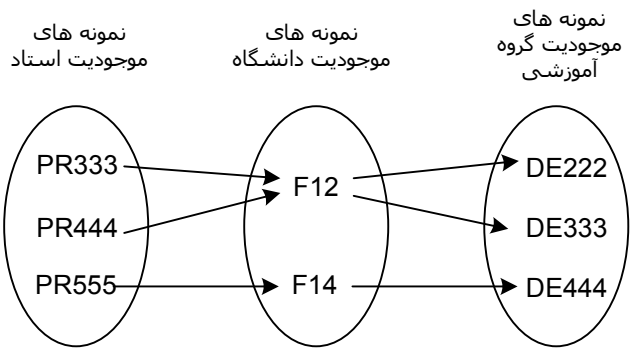
دام یک چندی (چند شاخه ای – چتری)

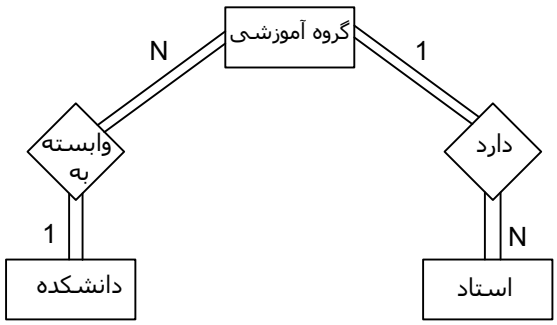
دام شکاف

دام یک چندی

این نوع دام وقتی ایجاد میشود که ارتباطی بین چند نوع موجودیت وجود داشته باشد ولی مسیر ارتباطی بین بعضی نمونه های یک موجودیت مبهم باشد.

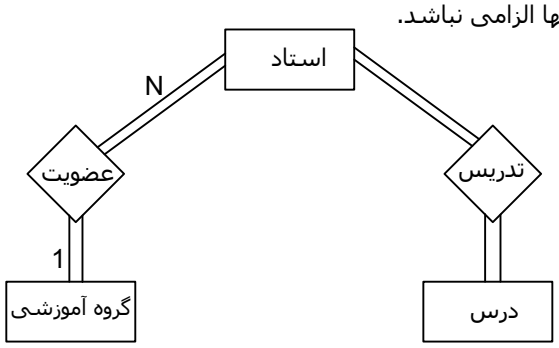


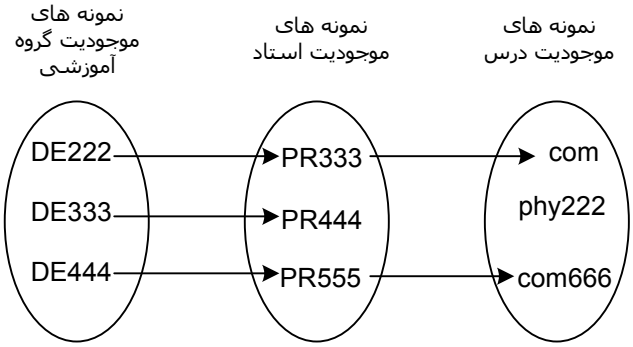


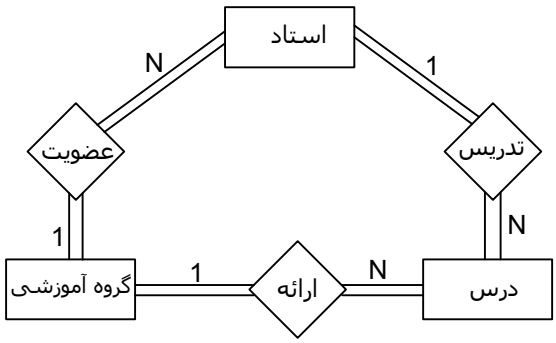


دام شکاف

این نوع دام وقتی ایجاد میشود که درمدل سازی انجام شده وجود ارتباط بین انواع موجودیتها نشان داده شده ولی چنین ارتباطی بین نمونه های از یک نوع موجودیت و دیگر انواع موجو دیتها وجود ندارد ،در واقع این دام وقتی ایجاد میشود که مشارکت یک نوع موجودیت در یک ارتباط بین انواع موجودیتها الزامی نباشد.







کلیدها در نمودار رابطه ای

1 فوق کلید(سوپر کلید):

- یک صفت یا مجموعه ای از صفات که به صورت منحصر به فردیک چند تایی را در یک رابطه مشخص میکند.

2 کلید کاندید:

- یک فوق کلید که هیچ زیر مجموعه ای از آن در آن رابطه یک فوق کلید نیست و دارای دو خاصیت زیر است.

1-منحصر بفرد بودن 2-غیر قابل کاهش بودن

3 کلید اصلی:

- کلید کاندیدی که انتخاب میشود تا بصورت منحصر به فرد چند تائیهای رابطه را مشخص نماید.

4 کلید جایگزین:

- کلیدهای کاندیدی که بعنوان کلید جایگزین انتخاب نشده اند.

5 کلید خارجی:

- یک صفت و یا مجموعه ای از صفات در یک رابطه که در یک رابطه دیگر کلید کاندید(اصلی هستند)

رده بندی سیستمهای مدیریت پایگاه داده ها

1 از نظر ساختار داده ای

- رابطه ای
- سلسله مراتبی
- شبکه ای

2 از نظر محیط سخت افزاری

- وابسته به یک سیستم خاص
- نا وابسته به یک محیط خاص

3 از نظر رده کامپیوتر

- خاص کامپیوترهای کوچک
- خاص کامپیوترهای متوسط
- خاص کامپیوترهای بزرگ

(4) از نظر محیط سیستم عامل

- وابسته به یک سیستم عامل خاص
- اجرا شونده در محیط چند سیستم عامل

(5) از نظر نوع معماری سیستم پایگاه داده ها

- با توانایی ایجاد پایگاه داده متمرکز
- با توانایی ایجاد پایگاه داده نامتمرکز

(6) از نظر نوع معماری مشتری-خدمتگذار

- با توانایی ایجاد معماری چند مشتری – یک خدمتگذار
- با توانایی ایجاد معماری چند مشتری – چند خدمتگذار

(7) از نظر زبان

- سیستم دارای SQL
- سیستم فاقد SQL

(8) از نظر نوع زبان داده های فرعی

I.DSL
E.DSL
E/I.DSL

(9) از نظر ماهیت زبان داده ای فرعی

- با زبان رویه ای
- با زبان نارویه ای

(10) از نظر سیستم فایل

- خودکفا
- وابسته به سیستم فایل محیط سیستم عامل

(11) از نظر نوع کاربرد

- تک منظوره
- چند منظوره

(12) از نظر قیمت

(13) از نظر واسط کاربر

- با واسط زمانی
- با واسط غیر زمانی
- با هر دو واسط

(14) از نظر رفتار در قبال رویدادها

- سیستم فعال
- سیستم غیر فعال

(15) از نظر متدولوژی زبان پایگاهی

- با متدلوژی شی گرا
- بدون متدلوژی شی گرا

اجزاء سیستم مدیریت پایگاه داده ها

(1) DBMS نمای بیرونی

- واحد پردازشگر پرسشها و برنامه های کاربردی
- واحد ایجاد و مدیریت داده های ذخیره شده

(2) DBMS نمای درونی

- لایه هسته (موتور پایگاه داده ها)
- لایه مدیریت محیط پایگاه داده ها
- لایه تسهیلات نرم افزاری (ابزارها)

مدیر پایگاه داده ها:

- فردیست متخصص(با دانش و تجربه در مدیریت)و آشنا با دانش و تکنولوژی پایگاه داده ها

وظائف مدیر پایگاه داده ها

1-مدل سازی معانی داده ها(رسم نمودار ER)
2-طراحی واسطهای کاربری
3- تصمیم گیری در انتخاب و انتصاب اعضاء تیمهای اجرائی
4- مطالعه دقیق و همه جانبه محیط عملیاتی و براورد خواسته ها و نیازهای کاربران
5- تست پایگاه داده ها با داده های واقعی در حجم واقعی
6- تلاش در جهت ارتقاء سطح دانش و فن اعضاء تیم و کاربران

سیستمهای پایگاه داده ها

(1) تک کاربری
(2) چند کاربری

مزایای سیستم تک کاربری

1- هربخش از سازمان داده های خود را نگهداری و پردازش میکند و در نتیجه سیستم کامپیوتر مرکزی میتواند به کارهای دیگر بپردازد.
2- با استفاده از کامپیوترهای شخصی حجم داده های سیستم مرکزی کاهش می یابد.
3- پایگاه داده ای که بر روی کامپیوتر شخصی ایجاد میشود کوچک و مدل سازی طراحی و پیاده سازی آن ساده است
4- کار با سیستمهای تک کاربری و برنامه سازی در محیط آنها ساده و نیاز به تخصص چندان ندارد.

معایب سیستم تک کاربری

1- وجود تعدادی سیستم پایگاه داده های کوچک و پراکنده در یک سازمان منجر به بروز افزونگی و ناسازگاری داده ها و تا حدی نا امنی آنها میشود.

2- خود سیستم نمیتواند کارا و قوی باشد در نتیجه بسیاری از خدماتی که یک سیستم قوی میتواند ارائه کندارائه نمیشود.

3-امکانات تولید نسخه های پشتیبان و ترمیم پایگاه داده ها معمولا کم است.

4- اعمال مجموعه ای از استانداد ها در کل سازمان نا ممکن است.

مزایای سیستم چند کاربری

1- اشتراک داده ها 2- کاهش حد المکان افزودگی
3- اجتناب از نا سازگاری داده ای 4-امکان اعمال استانداردها
5- حفظ محرمانگی داده ها 6- کاهش حجم برنامه ها
7- تنوع کاربران

معایب سیستم چند کاربری

1- هزینه بالای نرم افزار 2-هزینه بالای سخت افزار
3- هزینه بیشتر برای برنامه سازی
4- پیچیده بودن سیستم و نیاز به تخصص بیشتر
5-تاثیرات گسترده ترخرابیها و دشواری بیشتر ترمیم آنها

شرایط استفاده از تکنولوژی پایگاه داده ها

1. ایجاد یک سیستم یکپارچه اطلاعاتی در سازمان مد نظر باشد
2. حجم داده های سازمان زیاد بوده و مرتبا بصورت پویا رشد یابد
3. تغییرات مداوم در داده های ذخیره شده زیاد باشد و استقلال داده ای مورد نظر باشد.
4. میزان داده های مشترک بین برنامه های کاربردی زیاد باشد.
5. وجود ارتباطات پیچیده بین داده های سازمان
6. نیاز به سیستم داده کاوی و کشف دانش در سازمان

استقلال داده ای

یعنی وابسته نبودن برنامه های کاربردی به داده های ذخیره شده یا به بیان دیگر مصونیت دیدهای کاربران در سطح خارجی و برنامه های کاربردی آنها در قبال تغییراتی که در سطح ادراکی (مفهومی)و سطح داخلی پایگاه داده ها بروز میکند.

1. استقلال فیزیکی
- عبارتست از مصونیت دیدهای کاربران و برنامه های کاربردی در قبال تغییرات در سطح داخلی –فیزیکی پایگاه داده ها

2. استقلال منطقی
- عبارتست از مصونیت دیدهای کاربران و برنامه های کاربردی در قبال تغییرات در سطح ادراکی(مفهومی) پایگاه داده ها

معماری پایگاه داده ها

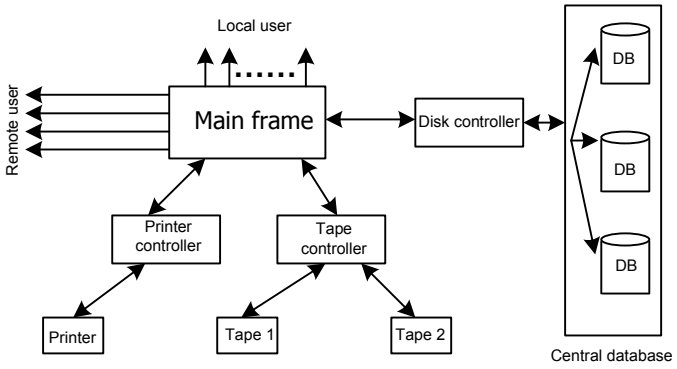
1- معماری متمرکز 2-معماری مشتری –خدمتگزار 3-معماری توزیع شده 4- معماری با پردازش موازی 5-معماری چند پایگاهی

منظور از معماری پایگاه دادها پیکره بندی یا طرز ترکیب اجزاء سیستمی است که در آن حداقل یک سیستم عامل-یک کامپیوتر با دستگاههای جانبی و تعدادی کاربر وجود دارد و خدمات پایگاهی به کاربران ارائه میکند.

1 معماری متمرکز

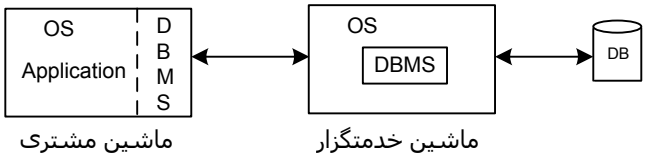
-در این معماری، پایگاه داده ها روی یک سیستم کامپیوتری بدون ارتباط با سیستم کامپیوتری دیگر ایجاد میشود.سخت افزار این سیستم میتواند یک کامپیوتر شخصی ،متوسط و یا بزرگ باشدو طبعا قدرت توانایی و کارائی سیستم نیز متفاوت است.

نکته:البته سیستم معماری متمرکز روی کامپیوترهای متوسط و بویژه بزرگ متصل به تعداد زیادی پایانه میتواند سیستم توانمندی باشد.



2 معماری مشتری خدمتگذار

- هر معماری که در آن قسمتی از پردازش را یک برنامه،سیستم و یا ماشین انجام دهد و انجام قسمت دیگری از پردازش را از برنامه،سیستم و یا ماشین دیگر بخواهد معماری مشتری خدمتگزار گفته میشود.



DBMS در ماشین مشتری الزامی نیست

دسته بندی ازلحاظ تعداد مشتری و خدمتگذار به سه حالت تقسیم میشود

1- چند مشتری – یک خدمتگذار

2- یک مشتری – چند خدمتگذار

3- چند مشتری – چند خدمتگذار

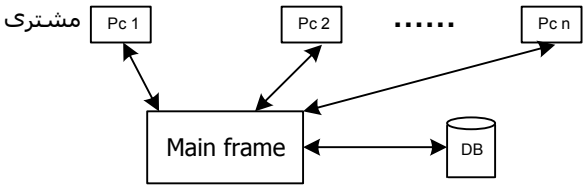
دسته بندی از لحاظ پیکره بندی سخت افزاری

1- معماری حول کامپیوتر بزرگ

2- معماری حول شبکه

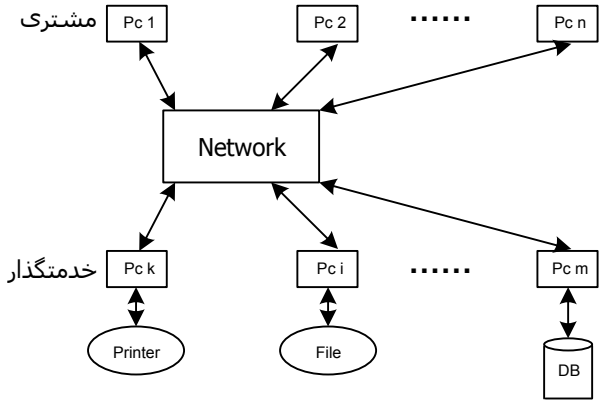
1 معماری حول کامپیوتر بزرگ

-در این طرح ماشین خدمتگذار یک کامپیوتر بزرگ است و پایگاه داده ها روی همین کامپیوتر ایجاد و مدیریت میشود و تعدادی کامپیوتر شخصی از خدمات پایگاهی این کامپیوتر بزرگ استفاده میکنند.



2 معماری حول شبکه

-در این طرح تعدادی کامپیوتر شخصی بعنوان خدمتگذار و تعدادی دیگر بعنوان مشتری از طریق شبکه بهم متصلند.



مزایای معماری مشتری - خدمتگذار

1- تقسیم پردازش

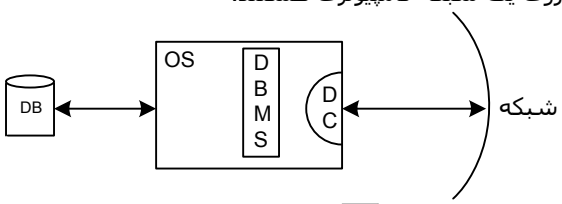
2- کاهش ترافیک شبکه ای

3- استقلال ایستگاههای کار

4- اشتراک داده ها

3 معماری توزیع شده

- مجموعه ای از داده های ذخیره شده که منطقا به یک سیستم تعلق دارند ولی در کامپیوترهای مختلف یا بیش از یک شبکه توزیع شده اند ،بعبارت دیگر در این معماری تعدادی پایگاه داده های ذخیره شده روی کامپیوتر های مختلف داریم و از نظر کاربران پایگاه واحدی هستند .و به بیان دیگر مجموعه ای است از چند پایگاه داده که منطقا بهم مرتبط و توزیع شده روی یک شبکه کامپیوتری هستند.



هر یک کامپیوتر در شبکه

- پایگاه داده
- سیستم مدیریت پایگاه داده ها
- مدیر انتقال داده ها

مزایای معماری توزیع شده

1-کارایی بیشتر در پردازش داده ها به ویژه در پایگاه داده های بزرگ

2- دستیابی بهتر به داده ها

3- اشتراک داده ها

4- افزایش پردازش موازی

5-کاهش هزینه ارتباطات

6- تسهیل گسترش سیستم

معایب معماری توزیع شده

1- پیچیدگی طراحی سیستم

2- پیچیدگی پیاده سازی

3-هزینه بیشتر

4-مصرف حافظه بیشتر

(4) معماری با پردازش موازی

این معماری با ساخت و گسترش ماشینهای موازی برای ایجاد پایگاه داده های خیلی بزرگ بکار میرود،گونه گسترش یافته معماری توزیع شده است و برای تامین کارائی و دستیابی پذیری بالا و گسترش پذیری سریع طراحی میشود.

این معماری بر اساس 2 طرح طراحی میشود.

1- چند پردازنده قوی (5-6)

2- تعدادی پردازنده ضعیف (100-150)

سه بخش اصلی طراحی موازی

1- مدیر تماسهای اجرائی کاربران

2-مدیر درخواستها

3- مدیر داده ها

چهار مدل مختلف در طراحی موازی

1- معماری با حافظه مشترک

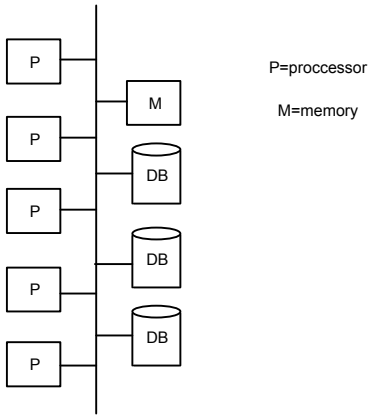
2- معماری با دیسک مشترک

3- معماری بی اجزاء مشترک

4- معماری سلسله مراتبی

(4-1) معماری با حافظه مشترک

در این طرح پردازنده ها به حافظه مشترک دسترسی دارند وضعیت این طرح این است که ارتباط بین پردازنده ها بطور کارا انجام میشود ،همچنین داده های ذخیره شده در حافظه در اختیار همه پردازنده هاقرار میگیرد،عیب این معماری در این است که نمیتوان بیش از 32 یا 64 پردازنده داشت . زیرا احتمال بروز تنگنا در Bus های حافظه یا شبکه ارتباطی افزایش میابد.



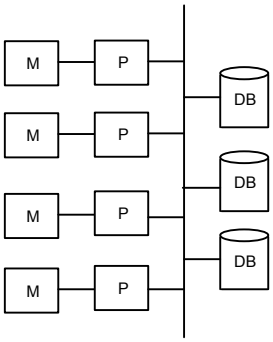
(4-2) معماری با دیسک مشترک

در این طرح تمام پردازنده ها به تمام دیسکها از طریق شبکه ارتباطی دسترسی دارند،هر پردازنده حافظه اختصاصی خود را دارد و مزیت آن

1-عدم بروز تنگنا

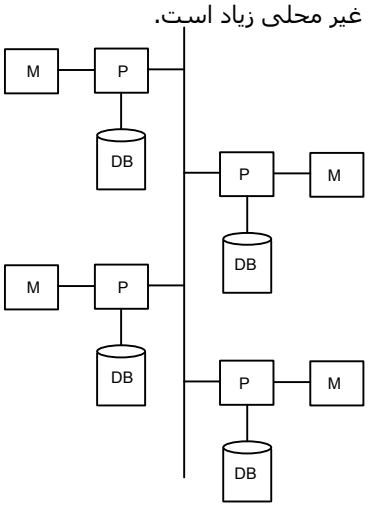
2-تسهیل تحمل خرابی

و عیب آن دشواری در گسترش سیستم است زیرا با افزایش دیسکها و پردازنده ها در ارتباط بین اجزاء تنگنا ایجاد میشودو سرعت ارتباط بین آنها کاهش می یابد.



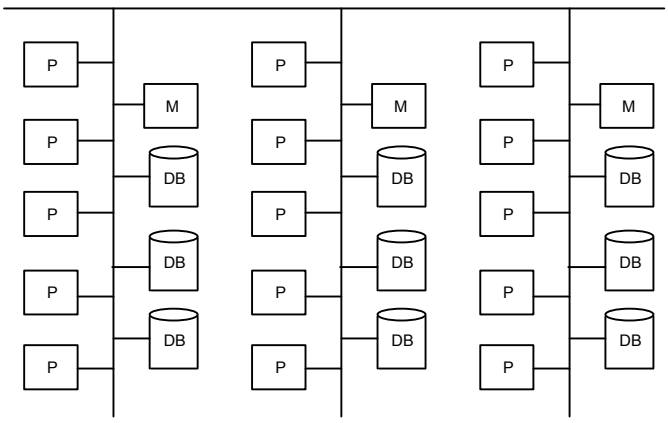
(4-3) معماری بی اجزاء مشترک

در این طرح هرماشین،پردازنده-حافظه و دیسک خود را دارد،یک شبکه ارتباطی با سرعت بالا این ماشینها را بهم مرتبط مینماید هر ماشین نوعی خدمتگذار پایگاهی است،مزیت آن تسهیل گسترش و عیب آن هزینه ارتباط و دستبابی های غیر محلی زیاد است.



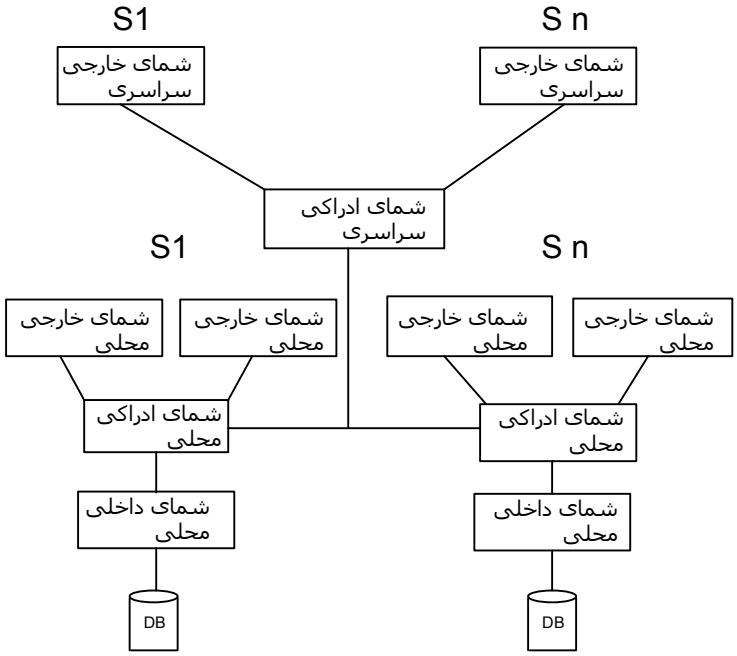
(4-4) معماری سلسله مراتبی

در این طرح ویژگیهای سه طرح پیش را ترکیب کرده و در بالاترین سطح سیستم تعدادی گره با شبکه ارتباطی بهم مرتبطند و اجزاء مشترک ندارند ،پس در این سطح معماری بی اشتراک داریم ،هر گره خود میتواند تعداد کمی پردازنده با حافظه مشترک داشته باشد و یا یک سیستم با دیسکهای مشترک داشته باشد.



معماری چند پایگاهی ⑤

این معماری نوعی معماری توزیع شده است که در آن گره ها خود مختاری کامل دارند،در این معماری معمولا چند سیستم با معماری توزیع شده از قبل وجود دارد و هر سیستم روی عملیات محلی خود کنترل کامل دارد.برای آنکه کاربران بتوانند نیازهای اطلاعاتی خود را تامین کنند بی آنکه مستقیما با اجزای معماری مرتبط باشند به یک لایه نرم افزاری خاصی نیاز است،این لایه نرم افزاری امکان میدهد تا در هر سیستم مدیر پایگاه داده هه روی داده های پایگاه خود کنترل کامل داشته باشد و نیازی به کنترل متمرکز نباشد.



دستورات SQL

ساخت جدول

```
create TABLE stt (
        stid char(10) not null PRIMARY KEY ,
        stname char (10)  ,
        stfamily char (10)  ,
        stdeg char (10)  ,
        stmaj char (10)  ,
        stdeid char (10) ,
      unique (stid) )
```

حذف جدول

```
DROP table stt
```

جدول دانشجو Stt table

	Column Name	Data Type	Length	Allow Nulls
🔑	stid	char	10	
	stname	char	10	✓
	stdeg	char	10	✓
	stmaj	char	10	✓
	stdeid	char	10	✓

جدول درس cot table

	Column Name	Data Type	Length	Allow Nulls
🔑	coid	char	10	
	cotitle	char	10	✓
	credit	int	4	✓
	cotype	char	10	✓
	codeid	char	10	✓

جدول نمره Stcot table

	Column Name	Data Type	Length	Allow Nulls
	stid	char	10	✓
	coid	char	10	✓
	tr	char	10	✓
	[year]	char	10	✓
	grade	int	4	✓

stt

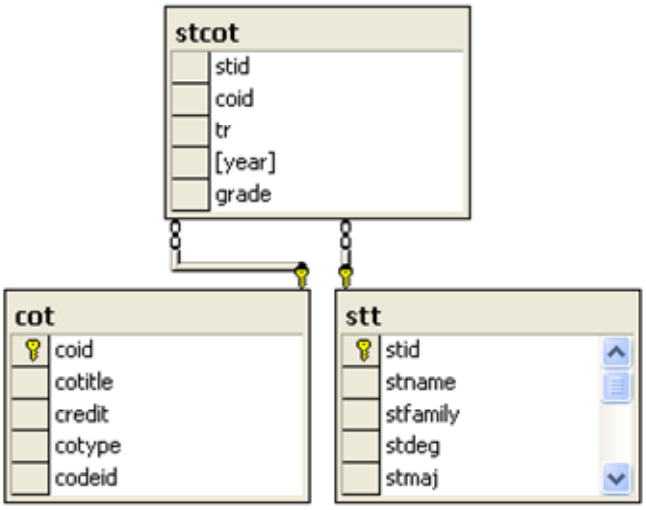
stid	stname	stdeg	stmaj	stdeid
76010222	stn2	ms	phys.	d333
76010444	stn3	ms	comp.	d777
77120333	stn1	bs	math.	d222
78110555	stn4	bs	hist.	d444
78120666	stn5	ms	comp.	d111

cot

coid	cotitle	credit	cotype	codeid
che777	chem lab	2	p	d777
com111	soft.sng.	3	t	d111
com190	data lab	1	p	d111
mat333	eng.math	4	t	d222
phy222	gen.phys.	4	t	d333

stcot

stid	coid	tr	year	grade
77120333	com111	1	77-78	11
77120333	com190	2	78-79	13
76010222	phy222	2	76-77	9
76010444	com190	1	77-78	6
78110555	com111	2	78-79	12
78110555	mat333	2	78-79	14



شماره و نام دانشجویان دوره کارشناسی را بدست آورید.

```
select stid,stname
from stt
where stdeg='ms'
```

	stid	stname
1	76010222	stn2
2	76010444	stn3
3	78120666	stn5

شماره و تعداد واحد تمام درسها را مرتب شده بصورت نزولی بر حسب مقادیر تعداد واحد بازیابی کنید.

```
select coid,credit
from cot
order by credit desc
```

	coid	credit
1	mat333	4
2	phy222	4
3	com111	3
4	che777	2
5	com190	1

شماره هر دانشجو و معدل اورا در ترم دوم 78-79 بدست آورید.

```
select stid,avg(grade)
from stcot
where tr='2' and year='78-79'
group by stid
```

	stid	(No column name)
1	77120333	13
2	78110555	13

کل تعداد درسها را بدست آورید.

```
select count(*)
from cot
```

	(No column name)
1	5

تعداد دانشجویان ثبت نام کرده در ترم دوم 78-79 را بدهید

```
select count(distinct stid)
from stcot
where tr='2' and year='78-79'
```

	(No column name)
1	2

بالاترین و پایین ترین نمره در ترم دوم سال 79-80 چیست

```
select max(grade),min(grade)
from stcot
where tr='2' and year='78-79'
```

	(No column name)	(No column name)
1	14	12

بالاترین و پایین ترین نمره در ترم دوم سال 79-80 در درس com222 چیست

```
select max(grade),min(grade)
from stcot
where tr='2' and year='78-79'and coid='com222'
```

	(No column name)	(No column name)
1	NULL	NULL

بالاترین و پایین ترین نمره با ذکر کد دانشجو در ترم دوم سال 77-78 را بدست آورید


```
select max(grade),min(grade)
from stcot
where tr='2' and year='78-79'
group by stid
```

	stid	(No column name)	(No column name)
1	77120333	13	13
2	78110555	14	12

مشخصات استادانی را بدهید که نام آنها با ma شروع میشود

```
select *
from prof
where prname like 'ma%'
```

مشخصات استادانی را بدهید که درنام آنها رشته کاراکتریzadوجود داشته باشد

```
select *
from prof
where prname like '%zad%'
```

مشخصات استادانی را بدهید که درنام آنها 8 کاراکتری بوده و کاراکتر سوم و چهارم آن ba باشد

```
select *
from prof
where prname like '--ba----'
```

شماره دانشجویان در ترم دوم 78-79 که در درس com111 را بدهید

```
select stid
from stcot
where tr='2' and year='78-79'
group by stid
```

	stid
1	78110555

شماره دانشجویان در ترم دوم 78-79 که در درسcom111 نمرهآنها هنوز اعلام نشده است را بیابید

```
select stid
from stcot
where coid='com111'and tr='2' and grade is null
```

	stid

شماره دانشجویانی را بدهید که یا دوره کارشناسی باشند یا درس com111 را ثبت نام کرده باشند.

```
select stid
from stt where stdeg='ms'
union
select stid
from stcot where
coid='com111'
```

	stid
1	76010222
2	76010444
3	77120333
4	78110555
5	78120666

دانشجویان گروه آموزشی d222 را بر حسب سطح دوره تحصیلی گروه بندی کنید.

```
select stdeid
from stt
where  stdeid='d222'
group by stdeid
```

	stdeid
1	d222

شماره درسهایی را بدهید که در ترم دوم 78-79 کمتر از 10 دانشجو در آن ثبت نام کرده اند.

```
select coid
from stcot where tr='2' and
year='78-79'
group by coid
having count(*)<10
```

	coid
1	com111
2	com190
3	mat333

درس شماره com111 را برای دانشجو با شماره 78110555 حذف کنید.

```
Delete
From stcot
Where stid='78110555' and coid='com111'
```

درسهای دانشجو با شماره 78110555 را در ترم دوم سال 78-79 حذف کنید

```
Delete
From stcot
Where stid='78110555' and tr='2' and
year='78-79'
```

گروه آموزشی با شماره d333 را حذف کنید.

```
Delete
From cot
Where codeid='d333'
```

اطلاعات زیر را درج کنید. (78110888,com888,2,78-79,12)

```
insert into stcot
('stid','coid','tr','year', grade)
values('78110888','com888','2','78-79',12)
```

تعداد دانشجویانی را بیابید که در درس com111 ترم 2 سال 78-79 نمره نیاورده اند.

```
select count(*)
from stcot
where tr='2' and year='78-79' and grade
between 0 and 10
```

	(No column name)
1	0

لیست دانشجویانی را استخراج کنید که در ترم اول 78-79 درس 3 واحدی انتخاب کرده اند.

```
select *
from stt
where stid in
(select stid from stcot where tr='1' and
year='78-79' and
coid in
(select coid from cot where credit='3'))
```

	stid	stname	stdeg	stmaj	stdeid

شماره دانشجویانی بدهید که نمره آنها در درس com190 در ترم دوم 78-79 بین 13 الی 19 باشد

```
select stid
from stcot where tr='2' and year='78-79'and
coid='com190'
and grade between 13 and 19
```

	stid
1	77120333

نام دانشجویانی را بدهید که درس com111 را انتخاب کرده اند.

```
select stt.stname
from stt,stcot
where stt.stid=stcot.stid and
stcot.coid='com111'
```

	stname
1	stn1
2	stn4

```
select stname
from stt where stid in
(select stid from stcot
where coid='com111')
```

نام دانشجویانی را بدهید که حداقل یک درس آزمایشگاهی انتخاب کرده باشند.

```
select stt.stname
from stt where stid in
(select stcot.stid
from stcot where coid in
(select cot.coid
from cot where cotype='p'))
```

	stname
1	stn3
2	stn1

عنوان کتابهایی را بدهید که قیمت آنها از بالاترین قیمت موجود در پایگاه داده ها کمتر باشد

```
Select booktitle
From book
Where bkprice<
(select max(bkprice) from book)
```

تغییر pr777 به دانشیار.

```
Update prof
Set rank='daneshyar'
Where prid='pr777'
```

تعداد واحد درسهای عملی را یک واحد افزایش دهید.

```
update cot
set credit=credit+1
where cotype='p'
```

	coid	cotitle	credit	cotype	codeid
	che777	chem lab	3	p	d777
	com111	soft.sng.	3	t	d111
	com190	data lab	2	p	d111
	mat333	eng.math	4	t	d222
	phy222	gen.phys.	4	t	d333

شماره دانشجویی 78110555 را به 78110777 تغییر دهید.

```
update stt
set stid='78110777'
where stid='78110555'
update stcot
set stid='78110777'
where stid='78110555'
```

```
select first, last, city
   from empinfo
   where first LIKE 'Er%'
```

first	last	city
Eric	Edwards	San Diego
Erica	Williams	Show Low

```
select first, last
   from empinfo
    where last LIKE '%s'

select * from empinfo
    where first = 'Eric'


select first, last, city
from empinfo

select last, city, age
from empinfo
where age > 30;

select first, last, city, state
from empinfo
where first LIKE 'J%'

select * from empinfo

select first, last
from empinfo
where last LIKE '%s'

select first, last, age
from empinfo
where last LIKE '%illia%'

select *
from empinfo
where first = 'Eric'

select first,last,city
from empinfo
where city <>'Payson'
```

first	last	city
Eric	Edwards	San Diego
Mary Ann	Edwards	Phoenix
Ginger	Howell	Cottonwood
Sebastian	Smith	Gila Bend
Gus	Gray	Bagdad
Mary Ann	May	Tucson
Erica	Williams	Show Low
Leroy	Brown	Pinetop
Elroy	Cleaver	Globe

```
select first, last
from empinfo  where last LIKE '%ay'

select *
from empinfo
where first = 'Mary'
```

Empinfo Table

id	first	last	age	city	state
99980	John	Jones	45	Payson	Arizona
99982	Mary	Jones	25	Payson	Arizona
88232	Eric	Edwards	32	San Diego	California
88233	Mary Ann	Edwards	32	Phoenix	Arizona
98002	Ginger	Howell	42	Cottonwood	Arizona
92001	Sebastian	Smith	23	Gila Bend	Arizona
22322	Gus	Gray	35	Bagdad	Arizona
32326	Mary Ann	May	52	Tucson	Arizona
32327	Erica	Williams	60	Show Low	Arizona
32380	Leroy	Brown	22	Pinetop	Arizona
32382	Elroy	Cleaver	22	Globe	Arizona

```
select "column1"
  [,"column2",etc]
  from "tablename"
  [where "condition"]
          [] = optional
```

=Equal
>Greater than
<Less than
>=Greater than or equal
<=Less than or equal
<>Not equal

Inserting into a Table

```
insert into "tablename"
  (first_column,...last_column)
    values (first_value,...last_value)
```

```
INSERT INTO empinfo
(id, [first], [last], age, city, state)
VALUES (23456, ' babak', 'ashofteh',
 '35', ' tehran', ' karaj')
```

id	first	last	age	city	state
99980	John	Jones	45	Payson	Arizona
99982	Mary	Jones	25	Payson	Arizona
88232	Eric	Edwards	32	San Diego	California
88233	Mary Ann	Edwards	32	Phoenix	Arizona
98002	Ginger	Howell	42	Cottonwood	Arizona
92001	Sebastian	Smith	23	Gila Bend	Arizona
22322	Gus	Gray	35	Bagdad	Arizona
32326	Mary Ann	May	52	Tucson	Arizona
32327	Erica	Williams	60	Show Low	Arizona
32380	Leroy	Brown	22	Pinetop	Arizona
32382	Elroy	Cleaver	22	Globe	Arizona
23456	babak	ashofteh	35	tehran	karaj

Column Name	Data Type	Length	Allow Nulls
id	int	4	✓
[first]	char	10	✓
[last]	char	10	✓
age	char	10	✓
city	char	10	✓
state	char	10	✓

Updating Records

```
update "tablename"
set "columnname" =
    "newvalue"
 [,"nextcolumn" =
   "newvalue2"...]
where "columnname"
  OPERATOR "value"
 [and|or "column"
  OPERATOR "value"]
[] = optional
```

```
update empinfo
  set first='hassan'
  where id=23456
```

id	first	last	age	city	state
99980	John	Jones	45	Payson	Arizona
99982	Mary	Jones	25	Payson	Arizona
88232	Eric	Edwards	32	San Diego	California
88233	Mary Ann	Edwards	32	Phoenix	Arizona
98002	Ginger	Howell	42	Cottonwood	Arizona
92001	Sebastian	Smith	23	Gila Bend	Arizona
22322	Gus	Gray	35	Bagdad	Arizona
32326	Mary Ann	May	52	Tucson	Arizona
32327	Erica	Williams	60	Show Low	Arizona
32380	Leroy	Brown	22	Pinetop	Arizona
32382	Elroy	Cleaver	22	Globe	Arizona
23456	hassan	ashofteh	35	tehran	karaj

Empinfo Table

id	first	last	age	city	state
99980	John	Jones	45	Payson	Arizona
99982	Mary	Jones	25	Payson	Arizona
88232	Eric	Edwards	32	San Diego	California
88233	Mary Ann	Edwards	32	Phoenix	Arizona
98002	Ginger	Howell	42	Cottonwood	Arizona
92001	Sebastian	Smith	23	Gila Bend	Arizona
22322	Gus	Gray	35	Bagdad	Arizona
32326	Mary Ann	May	52	Tucson	Arizona
32327	Erica	Williams	60	Show Low	Arizona
32380	Leroy	Brown	22	Pinetop	Arizona
32382	Elroy	Cleaver	22	Globe	Arizona

```
select *
from empinfo
where first LIKE '%Mary%'
```

id	first	last	age	city	state
99982	Mary	Jones	25	Payson	Arizona
88233	Mary Ann	Edwards	32	Phoenix	Arizona
32326	Mary Ann	May	52	Tucson	Arizona

Creating Tables

The **create table** statement is used to create a new table.

Here is the format of a simple **create table** statement:

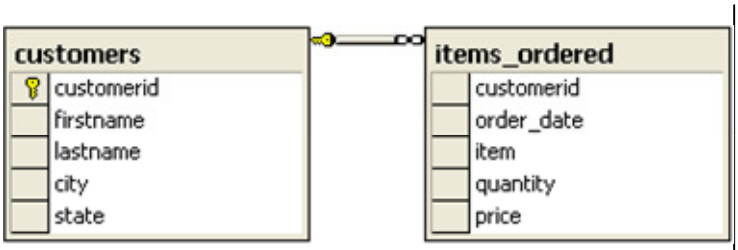
```
create table "tablename"
("column1" "data type",
 "column2" "data type",
 "column3" "data type");
```

create table if you were to use optional constraints:

```
create table "tablename"
("column1" "data type"
          [constraint],
 "column2" "data type"
          [constraint],
 "column3" "data type"
         [constraint]);
 [ ] = optional
```

```
create table employee
(first varchar(15),
 last varchar(20),
 age number(3),
 address varchar(30),
 city varchar(20),
    state varchar(20))
```

char(size) Fixed-length character string. Size is specified in parenthesis. Max 256bytes.
varchar(size) Variable-length character string. Max size is specified in parenthesis.
number(size) Number value with a max number of column digits specified in parenthesis.
date Date value
number(size,d) Number value with a maximum number of digits of "size" total, with a maximum number of "d" digits to the right of the decimal.



customers Table

customerid	firstname	lastname	city	state
10101	John	Gray	Lynden	Washington
10298	Leroy	Brown	Pinetop	Arizona
10299	Elroy	Keller	Snoqualmie	Washington
10315	Lisa	Jones	Oshkosh	Washington
10325	Ginger	Schultz	Pocatello	Idaho
10329	Kelly	Mendoza	Kailua	Hawaii
10330	Shawn	Dalton	Cannon Beach	Oregon
10338	Michael	Howell	Tillamook	Oregon
10339	Anthony	Sanchez	Winslow	Arizona
10408	Elroy	Cleaver	Globe	Arizona
10410	Mary Ann	Howell	Charleston	South Carolina
10413	Donald	Davids	Gila Bend	Arizona
10419	Linda	Sakahara	Nogales	Arizona
10429	Sarah	Graham	Greensboro	North Carolina
10438	Kevin	Smith	Durango	Colorado
10439	Conrad	Giles	Telluride	Colorado
10449	Isabela	Moore	Yuma	Arizona

Item_ordered Table

customerid	order_date	item	quantity	price
10330	30-Jun-1999	Pogo stick	1	28
10101	30-Jun-1999	Raft	1	58
10298	01-Jul-1999	Skateboard	1	33
10101	01-Jul-1999	Life Vest	4	125
10299	06-Jul-1999	Parachute	1	1250
10339	27-Jul-1999	Umbrella	1	4.5
10449	13-Aug-1999	Unicycle	1	180.79
10439	14-Aug-1999	Ski Poles	2	25.5
10101	18-Aug-1999	Rain Coat	1	18.3
10449	01-Sep-1999	Snow Shoes	1	45
10439	18-Sep-1999	Tent	1	88
10298	19-Sep-1999	Lantern	2	29
10410	28-Oct-1999	Sleeping Bag	1	89.22
10438	01-Nov-1999	Umbrella	1	6.75
10438	02-Nov-1999	Pillow	1	8.5
10298	01-Dec-1999	Helmet	1	22
10449	15-Dec-1999	Bicycle	1	380.5
10449	22-Dec-1999	Canoe	1	280
10101	30-Dec-1999	Hoola Hoop	3	14.75
10330	01-Jan-2000	Flashlight	4	28
10101	02-Jan-2000	Lantern	1	16
10299	18-Jan-2000	Inflatable Mattress	1	38
10438	18-Jan-2000	Tent	1	79.99
10413	19-Jan-2000	Lawnchair	4	32
10410	30-Jan-2000	Unicycle	1	192.5
10315	2-Feb-2000	Compass	1	8
10449	29-Feb-2000	Flashlight	1	4.5
10101	08-Mar-2000	Sleeping Bag	2	88.7
10298	18-Mar-2000	Pocket Knife	1	22.38
10449	19-Mar-2000	Canoe paddle	2	40
10298	01-Apr-2000	Ear Muffs	1	12.5
10330	19-Apr-2000	Shovel	1	16.75


```
update phone_book
  set last_name = 'Smith', prefix=555,
suffix=9292
  where last_name = 'Jones'

update employee
  set age = age+1
  where first_name='Mary' and
last_name='Williams'
```

Deleting Records

```
delete from "tablename"

where "columnname"
  OPERATOR "value"
[and|or "column"
  OPERATOR "value"]
[ ] = optional
```

```
delete from employee
  where lastname = 'May'

delete from employee
  where firstname = 'Mike' or firstname =
'Eric'

delete from myemployees_ts0211
 where lastname = 'Weber-Williams'

delete from myemployees_ts0211
  where salary > 70000
```

```
delete from empinfo
where last='yazdi' or id=23456

DELETE FROM empinfo
WHERE ([last] = 'yazdi')OR(id = 23456)
```

Drop a Table

```
drop table "tablename"

drop table myemployees_ts0211
```

customers Table

Column Name	Data Type	Length	Allow Nulls
customerid	int	4	
firstname	char	10	✓
lastname	char	10	✓
city	char	25	✓
state	char	25	✓

Item_ordered Table

Column Name	Data Type	Length	Allow Nulls
customerid	int	4	✓
order_date	char	20	✓
item	char	20	✓
quantity	bigint	8	✓
price	real	4	✓

```
SELECT customerid, order_date, item
FROM items_ordered
WHERE item LIKE 's%'
```

	customerid	order_date	item
1	10298	01-Jul-1999	Skateboard
2	10439	14-Aug-1999	Ski Poles
3	10449	01-Sep-1999	Snow Shoes
4	10410	28-Oct-1999	Sleeping Bag
5	10101	08-Mar-2000	Sleeping Bag
6	10330	19-Apr-2000	Shovel

```
SELECT DISTINCT item
FROM items_ordered
```

	item
1	Bicycle
2	Canoe
3	Canoe paddle
4	Compass
5	Ear Muffs
6	Flashlight
7	Helmet
8	Hoola Hoop
9	Inflatable Mattress
10	Lantern
11	Lawnchair
12	Life Vest
13	Parachute
14	Pillow
15	Pocket Knife
16	Pogo stick
17	Raft
18	Rain Coat
19	Shovel
20	Skateboard
21	Ski Poles
22	Sleeping Bag
23	Snow Shoes
24	Tent
25	Umbrella
26	Unicycle

Aggregate Functions

MIN returns the smallest value in a given column
MAX returns the largest value in a given column
SUM returns the sum of the numeric values in a given column
AVG returns the average value of a given column
COUNT returns the total number of values in a given column
COUNT(*) returns the number of rows in a table

SELECT Statement

```
SELECT [ALL | DISTINCT] column1[,column2]

FROM table1[,table2]

[WHERE "conditions"]

[GROUP BY "column-list"]

[HAVING "conditions]

[ORDER BY "column-list" [ASC | DESC] ]


SELECT name, age, salary
FROM employee
WHERE age > 5o

SELECT name, title, dept
FROM employee
WHERE title LIKE 'Pro%
```

ALL and **DISTINCT** are keywords used to select either ALL (default) or the "distinct" or unique records in your query results. If you would like to retrieve just the unique records in specified columns, you can use the "DISTINCT" keyword. DISTINCT will discard the duplicate records for the columns you specified after the "SELECT" statement: For example

```
SELECT DISTINCT age
FROM employee_info
```

```
SELECT customerid, item, price
FROM items_ordered
WHERE customerid=10449
```

	customerid	item	price
1	10449	Unicycle	180.78999
2	10449	Snow Shoes	45.0
3	10449	Bicycle	380.5
4	10449	Canoe	280.0
5	10449	Flashlight	4.5
6	10449	Canoe paddle	40.0

```
SELECT * FROM items_ordered
WHERE item = 'Tent'
```

	customerid	order_date	item	quantity	price
1	10439	18-Sep-1999	Tent	1	88.0
2	10438	18-Jan-2000	Tent	1	79.989998

```
SELECT state, count(state)
FROM customers
GROUP BY state
```

	state	(No column name)
1	Arizona	6
2	Colorado	2
3	Hawaii	1
4	Idaho	1
5	North Carolina	1
6	Oregon	2
7	South Carolina	1
8	Washington	3

```
SELECT item, max(price), min(price)
FROM items_ordered
GROUP BY item
```

	item	(No column name)	(No column name)
1	Bicycle	380.5	380.5
2	Canoe	280.0	280.0
3	Canoe paddle	40.0	40.0
4	Compass	8.0	8.0
5	Ear Muffs	12.5	12.5
6	Flashlight	28.0	4.5
7	Helmet	22.0	22.0
8	Hoola Hoop	14.75	14.75
9	Inflatable Mattress	38.0	38.0
10	Lantern	29.0	16.0
11	Lawnchair	32.0	32.0
12	Life Vest	125.0	125.0
13	Parachute	1250.0	1250.0
14	Pillow	8.5	8.5
15	Pocket Knife	22.379999	22.379999
16	Pogo stick	28.0	28.0
17	Raft	58.0	58.0
18	Rain Coat	18.299999	18.299999
19	Shovel	16.75	16.75
20	Skateboard	33.0	33.0
21	Ski Poles	25.5	25.5
22	Sleeping Bag	89.220001	88.699997
23	Snow Shoes	45.0	45.0
24	Tent	88.0	79.989998
25	Umbrella	6.75	4.5
26	Unicycle	192.5	180.78999

```
SELECT avg (price)
FROM items_ordered
```

	(No column name)
1	102.06656211614609

```
SELECT avg (price)
FROM items_ordered
where customerid=10101
```

	(No column name)
1	53.458332697550453

```
SELECT count (*)
FROM customers
```

	(No column name)
1	17

```
SELECT max(price)
FROM items_ordered
```

```
SELECT avg(price)
FROM items_ordered
where order_date like '%sep%'
```

	(No column name)
1	54.0

```
SELECT min(price)
FROM items_ordered
WHERE item = 'Tent'
```

	(No column name)
1	79.989998

GROUP BY clause

The GROUP BY clause will gather all of the rows together that contain data in the specified column(s) and will allow aggregate functions to be performed on the one or more columns. This can best be explained by an example:

GROUP BY clause syntax:

```
SELECT column1,
SUM(column2)
FROM "list-of-tables"
GROUP BY "column-list"
```

```
SELECT quantity, max(price)
FROM items_ordered
GROUP BY quantity
```

	quantity	(No column name)
1	1	1250.0
2	2	88.699997
3	3	14.75
4	4	125.0

```
SELECT customerid, count(customerid),
sum(price)
FROM items_ordered
GROUP BY customerid
HAVING count(customerid) > 1
```

	customerid	(No column name)	(No column name)
1	10101	6	320.74999618530273
2	10298	5	118.8799991607666
3	10299	2	1288.0
4	10330	3	72.75
5	10410	2	281.72000122070312
6	10438	3	95.239997863769531
7	10439	2	113.5
8	10449	6	930.78999328613281

ORDER BY clause

ORDER BY is an optional clause which will allow you to display the results of your query in a sorted order (either ascending order or descending order) based on the columns that you specify to order by

ORDER BY clause syntax:

```
SELECT column1, SUM(column2)
FROM "list-of-tables"
ORDER BY "column-list" [ASC | DESC]

[ ] = optional

ASC = Ascending Order - default
DESC = Descending Order
```

```
SELECT lastname, firstname,
city
FROM customers
ORDER BY lastname
```

	lastname	firstname	city
1	Brown	Leroy	Pinetop
2	Cleaver	Elroy	Globe
3	Dalton	Shawn	Cannon Beach
4	Davids	Donald	Gila Bend
5	Giles	Conrad	Telluride
6	Graham	Sarah	Greensboro
7	Gray	John	Lynden
8	Howell	Michael	Tillamook
9	Howell	Mary Ann	Charleston
10	Jones	Lisa	Oshkosh
11	Keller	Elroy	Snoqualmie
12	Mendoza	Kelly	Kailua
13	Moore	Isabela	Yuma
14	Sakahara	Linda	Nogales
15	Sanchez	Anthony	Winslow
16	Schultz	Ginger	Pocatello
17	Smith	Kevin	Durango

33

```
SELECT customerid, count(customerid),
sum(price)
FROM items_ordered
GROUP BY customerid
```

	customerid	(No column name)	(No column name)
1	10101	6	320.74999618530273
2	10298	5	118.8799991607666
3	10299	2	1288.0
4	10315	1	8.0
5	10330	3	72.75
6	10339	1	4.5
7	10410	2	281.72000122070312
8	10413	1	32.0
9	10438	3	95.239997863769531
10	10439	2	113.5
11	10449	6	930.78999328613281

HAVING clause

The HAVING clause allows you to specify conditions on the rows for each group - in other words, which rows should be selected will be based on the conditions you specify. The HAVING clause should follow the GROUP BY clause if you are going to use it

HAVING clause syntax:

```
SELECT column1,
SUM(column2)
FROM "list-of-tables"
GROUP BY "column-list"
HAVING "condition"
```

```
SELECT state,
count(state)
FROM customers
GROUP BY state
HAVING count(state) > 1
```

	state	(No column name)
1	Arizona	6
2	Colorado	2
3	Oregon	2
4	Washington	3

```
SELECT item, max(price),
min(price)
FROM items_ordered
GROUP BY item
HAVING max(price) > 190.00
```

	item	(No column name)	(No column name)
1	Bicycle	380.5	380.5
2	Canoe	280.0	280.0
3	Parachute	1250.0	1250.0
4	Unicycle	192.5	180.78999

```
SELECT item, price
FROM items_ordered
WHERE (item LIKE 'S%') OR (item LIKE 'P%')
OR (item LIKE 'F%')
order by item
```

	item	price
1	Flashlight	28.0
2	Flashlight	4.5
3	Parachute	1250.0
4	Pillow	8.5
5	Pocket Knife	22.379999
6	Pogo stick	28.0
7	Shovel	16.75
8	Skateboard	33.0
9	Ski Poles	25.5
10	Sleeping Bag	89.220001
11	Sleeping Bag	88.699997
12	Snow Shoes	45.0

IN NOT IN and BETWEEN NOT BETWEEN Conditional Operators

```
SELECT order_date, item, price
FROM items_ordered
WHERE price BETWEEN 40 AND 80
order by item
```

	order_date	item	price
1	19-Mar-2000	Canoe paddle	40.0
2	30-Jun-1999	Raft	58.0
3	01-Sep-1999	Snow Shoes	45.0
4	18-Jan-2000	Tent	79.989998

```
SELECT firstname, city, state
FROM customers
WHERE state not IN ('Arizona', 'Washington',
'Oklahoma', 'Colorado', 'Hawaii')
```

	firstname	city	state
1	Ginger	Pocatello	Idaho
2	Shawn	Cannon Beach	Oregon
3	Michael	Tillamook	Oregon
4	Mary Ann	Charleston	South Carolina
5	Sarah	Greensboro	North Carolina

Mathematical Operators

+addition
-subtraction
***multiplication**
/division
%modulo


```
SELECT lastname, firstname, city
FROM customers
ORDER BY lastname DESC

SELECT item, price
FROM items_ordered
WHERE price > 10.00
ORDER BY price ASC
```

	item	price
1	Ear Muffs	12.5
2	Hoola Hoop	14.75
3	Lantern	16.0
4	Shovel	16.75
5	Rain Coat	18.299999
6	Helmet	22.0
7	Pocket Knife	22.379999
8	Ski Poles	25.5
9	Pogo stick	28.0
10	Flashlight	28.0
11	Lantern	29.0
12	Lawnchair	32.0
13	Skateboard	33.0
14	Inflatable Mattress	38.0
15	Canoe paddle	40.0
16	Snow Shoes	45.0
17	Raft	58.0
18	Tent	79.989998
19	Tent	88.0
20	Sleeping Bag	88.699997
21	Sleeping Bag	89.220001
22	Life Vest	125.0
23	Unicycle	180.78999
24	Unicycle	192.5
25	Canoe	280.0
26	Bicycle	380.5
27	Parachute	1250.0

Combining conditions and Boolean Operators

```
SELECT customerid, order_date, item
FROM items_ordered
WHERE (item <> 'Snow shoes') AND
(order_date not like '%1999')
```

	customerid	order_date	item
1	10330	01-Jan-2000	Flashlight
2	10101	02-Jan-2000	Lantern
3	10299	18-Jan-2000	Inflatable Mattress
4	10438	18-Jan-2000	Tent
5	10413	19-Jan-2000	Lawnchair
6	10410	30-Jan-2000	Unicycle
7	10315	2-Feb-2000	Compass
8	10449	29-Feb-2000	Flashlight
9	10101	08-Mar-2000	Sleeping Bag
10	10298	18-Mar-2000	Pocket Knife
11	10449	19-Mar-2000	Canoe paddle
12	10298	01-Apr-2000	Ear Muffs
13	10330	19-Apr-2000	Shovel

```
SELECT customers.customerid,
customers.firstname, customers.state,
items_ordered.item
FROM customers, items_ordered
WHERE customers.customerid =
items_ordered.customerid
ORDER BY customers.state DESC
```

	customerid	firstname	state	item
1	10101	John	Washington	Raft
2	10101	John	Washington	Life Vest
3	10299	Elroy	Washington	Parachute
4	10101	John	Washington	Rain Coat
5	10101	John	Washington	Hoola Hoop
6	10101	John	Washington	Lantern
7	10299	Elroy	Washington	Inflatable Mattress
8	10315	Lisa	Washington	Compass
9	10101	John	Washington	Sleeping Bag
10	10410	Mary Ann	South Carolina	Unicycle
11	10410	Mary Ann	South Carolina	Sleeping Bag
12	10330	Shawn	Oregon	Pogo stick
13	10330	Shawn	Oregon	Shovel
14	10330	Shawn	Oregon	Flashlight
15	10438	Kevin	Colorado	Tent
16	10439	Conrad	Colorado	Ski Poles
17	10438	Kevin	Colorado	Umbrella
18	10438	Kevin	Colorado	Pillow
19	10439	Conrad	Colorado	Tent
20	10298	Leroy	Arizona	Lantern
21	10449	Isabela	Arizona	Snow Shoes
22	10298	Leroy	Arizona	Helmet
23	10449	Isabela	Arizona	Unicycle
24	10339	Anthony	Arizona	Umbrella
25	10298	Leroy	Arizona	Skateboard
26	10413	Donald	Arizona	Lawnchair
27	10449	Isabela	Arizona	Bicycle
28	10449	Isabela	Arizona	Canoe
29	10298	Leroy	Arizona	Ear Muffs
30	10298	Leroy	Arizona	Pocket Knife
31	10449	Isabela	Arizona	Canoe paddle
32	10449	Isabela	Arizona	Flashlight

ABS(x) returns the absolute value of x
SIGN(x) returns the sign of input x as -1, 0, or 1 (negative, zero, or positive respectively)
MOD(x,y) modulo - returns the integer remainder of x divided by y (same as x%y)
FLOOR(x) returns the largest integer value that is less than or equal to x
CEILING(x) or
CEIL(x) returns the smallest integer value that is greater than or equal to x
POWER(x,y) returns the value of x raised to the power of y
ROUND(x) returns the value of x rounded to the nearest whole integer \
ROUND(x,d) returns the value of x rounded to the number of decimal places specified by the value d
SQRT(x) returns the square-root value of x

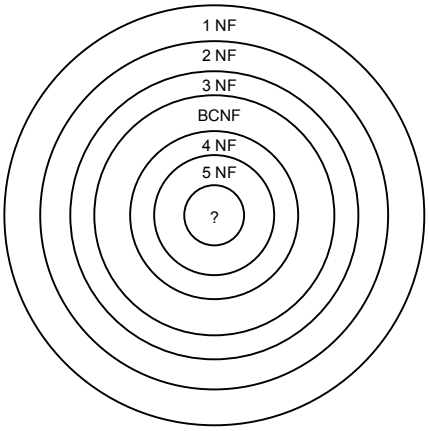
```
select item, sum(price)/sum(quantity)
from items_ordered
group by item
```

	item	(No column name)
1	Bicycle	380.5
2	Canoe	280.0
3	Canoe paddle	20.0
4	Compass	8.0
5	Ear Muffs	12.5
6	Flashlight	6.5
7	Helmet	22.0
8	Hoola Hoop	4.916666666666667
9	Inflatable Mattress	38.0
10	Lantern	15.0
11	Lawnchair	8.0
12	Life Vest	31.25
13	Parachute	1250.0
14	Pillow	8.5
15	Pocket Knife	22.379999160766602
16	Pogo stick	28.0
17	Raft	58.0
18	Rain Coat	18.299999237060547
19	Shovel	16.75
20	Skateboard	33.0
21	Ski Poles	12.75
22	Sleeping Bag	59.306666056315102
23	Snow Shoes	45.0
24	Tent	83.994998931884766
25	Umbrella	5.625
26	Unicycle	186.64499664306641

Table Joins, a must

```
SELECT customers.customerid,
customers.firstname, customers.lastname,
items_ordered.order_date,
items_ordered.item, items_ordered.price
FROM customers, items_ordered
WHERE customers.customerid =
items_ordered.customerid
```

نرمال سازی:



وابستگی تابعی

فرض میکنیم R یک متغییر رابطه ای و A و B دو زیر مجموعه دلخواه از عنوان Rباشند می گوییم B با A وابستگی تابعی دارد و چنین نمایش می دهیم $A \longrightarrow B$ اگر و فقط اگر در هر مقدار از متغییر رابطه ای R به هر مقدار A فقط یک مقدار B متناظر باشد به بیان دیگر اگر مقدار A در 2 یا بیش از 2 رکورد از R یکسان باشد مقدار نیز چنین است.

STTCOT TABLE

STID	COID	STNAME	GRADE	STMJR	STDEID
222	MAT222	STN1	12	COMP	D111
222	COM111	STN1	8	COMP	D111
222	COM120	STN1	13	COMP	D111
111	MAT222	STN2	17	MATH	D222
111	MAT777	STN2	11	MATH	D222
333	COM777	STN3	10	PHYS	D333
444	COM444	STN4	14	STAT	D444
444	SAT666	STN4	15	STAT	D444
666	COM500	STN5	11	COMP	D111

ترکیب دو فیلد کلید اصلی میشود و هر فیلدی به کلید اصلی وابستگی تابعی دارد (STID,COID)

$STID \xrightarrow{\times} COID$ $COID \xrightarrow{\times} STNAME$

رابطه تابعی دارد $STID \xrightarrow{\checkmark} STNAME$ $COID \xrightarrow{\times} GRADE$

$STID \xrightarrow{\times} GRADE$ $COID \xrightarrow{\times} STDEID$

$STID \xrightarrow{\checkmark} STMJR$ $(STID,COID) \xrightarrow{\checkmark} STDEID$

$STID \xrightarrow{\checkmark} STDEID$ $(STID,COID) \xrightarrow{\checkmark} GRADE$

$COID \xrightarrow{\times} STMJR$

وابستگی تابعی کامل

اگر x و y دو زیر مجموعه از مجموعه عنوان رابطه A باشد می گوییم x با y وابستگی تابعی کامل دارد و چنین نشان می دهیم $x \Longrightarrow y$ اگر و فقط اگر x با y وابستگی داشته باشد ولی با هیچ زیر مجموعه از x وابستگی تابعی نداشته باشد.

$(STID,COID) \xrightarrow{\times} STNAME$

وابستگی تابعی کامل ندارد
- $(STID,COID) \xrightarrow{\checkmark} STNAME$
- $STID \xrightarrow{\checkmark} STNAME$
- $COID \xrightarrow{\times} STNAME$

$(STID,COID) \xrightarrow{\times} STDEID$

وابستگی تابعی کامل ندارد
- $(STID,COID) \xrightarrow{\checkmark} STDEID$
- $STID \xrightarrow{\checkmark} STDEID$
- $COID \xrightarrow{\times} STDEID$

$(STID,COID) \xrightarrow{\checkmark} GRADE$

وابستگی تابعی کامل دارد
- $(STID,COID) \xrightarrow{\checkmark} GRADE$
- $STID \xrightarrow{\times} GRADE$
- $COID \xrightarrow{\times} GRADE$

$STID \xrightarrow{\checkmark} STDEID$

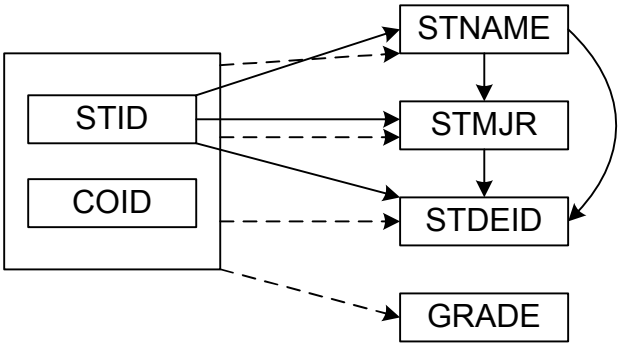
$STID \xrightarrow{\checkmark} STDEID$

در این حالت فقط وابستگی تابعی را بررسی میکنیم که در این مثال وابستگی تابعی کامل دارد

وابستگی تابعی کامل دارد
- $(STID,COID) \xrightarrow{\checkmark} STMJR$
- $STID \xrightarrow{\checkmark} STMJR$
- $COID \xrightarrow{\times} STMJR$

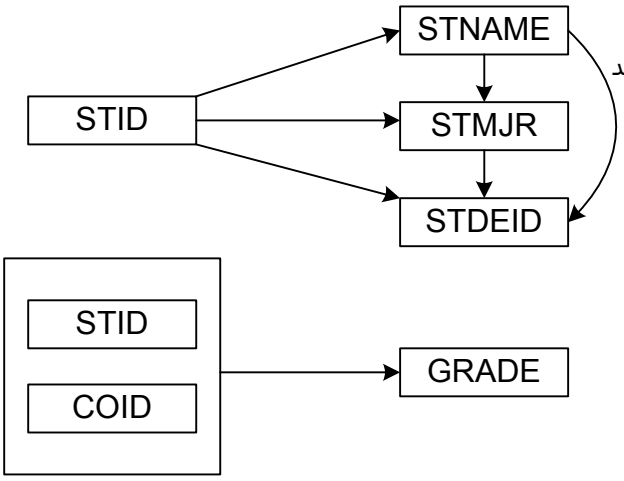
$(STID,COID) \xrightarrow{\times} STMJR$

STID,COID باهم کلید اصلی هستند



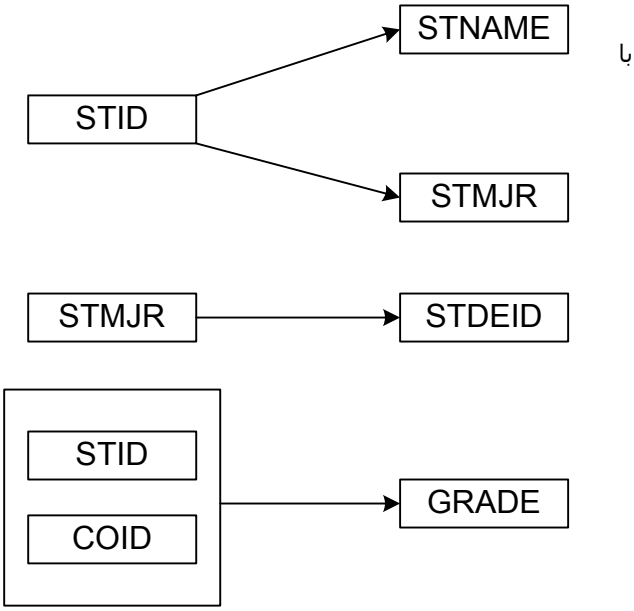
رابطه ای 1NF است اگر هر صفت خاصه آن در هر رکورد تک مقداری باشد،به بیان دیگر صفت چند مقداری نداشته باشد (یعنی فیلد مرکب وجود نداشته باشد مانند فیلد آدرس که میشود به فیلد های کوچکتر مانند استان،شهر،خیابان،کوچه،پلاک.... تجزیه کرد.)

STID ✓→ STNAME
STID ✗→ GRADE
STID ✓→ STMJR
STID ✓→ STDEID
STNAME ✓→ STDEID
COID ✗→ STNAME
COID ✗→ GRADE
COID ✗→ STDEID
COID ✗→ STMJR
STMJR ✓→ STMJR



رابطه ای 2NF است اگر اولا 1NF باشد و ثانیا تمام صفات غیر کلید با کلید اصلی وابستگی تابعی کامل داشته باشند.
یک فیلد2NF حتما 1NFنیز هست.

(STID,COID) ⟹✓ GRADE
(STID,COID) ⟹✗ STNAME
(STID,COID) ⟹✗ STDEID
(STID,COID) ⟹✗ STMJR

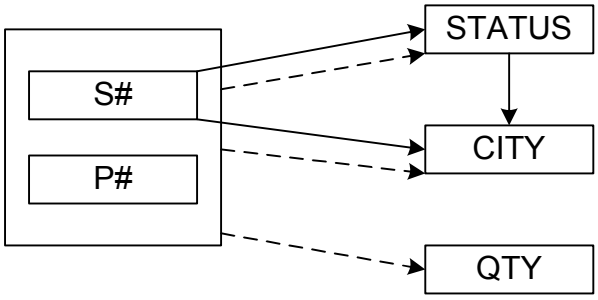


رابطه ای 3NF است اگر اولا 2NF باشد و ثانیاهر صفت غیر کلید با کلید اصلی وابستگی تابعی بی واسطه داشته باشد.

S#	STATUS	CITY	P#	QTY
S1	20	C2	P1	300
S1	20	C2	P2	200
S1	20	C2	P3	400
S2	10	C3	P1	100
S2	10	C3	P2	200
S3	10	C3	P2	200
S4	20	C2	P2	200
S4	20	C2	P4	300

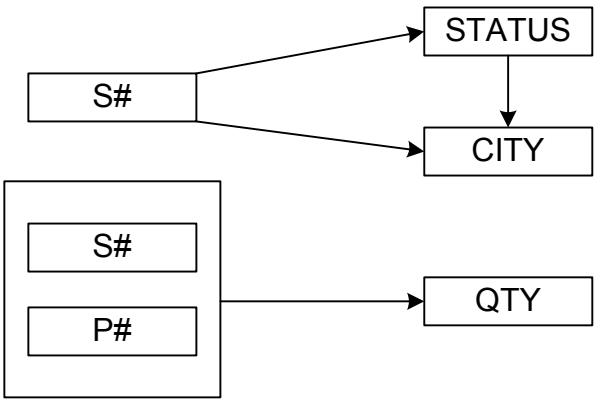
S# ✗→ QTY
S# ✓→ CITY
S# ✓→ STATUS
P# ✗→ QTY
P# ✗→ CITY
P# ✗→ STATUS
(S#,P#) ✓⇒ QTY
(S#,P#) ✗⇒ CITY
(S#,P#) ✗⇒ STATUS

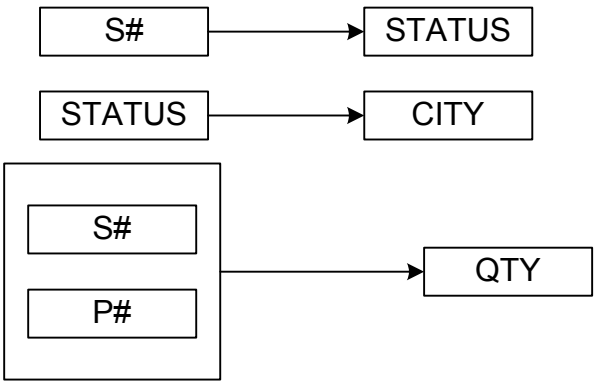
STATUS ✓→ CITY
STATUS ✗→ QTY
CITY ✗→ QTY



1NF

2NF





3NF

① Domain integrity (جامعیت دامنه ای):

یعنی تمام صفات در تمامی رابطه ها از نوع دامنه خود باشد(مثلا 74.1به عنوان کد دانشجویی که عدد صحیح مثبتی است پذیرفته نشود)

② Intra-relation integrity (جامعیت درون رابطه ای):

یعنی هر رابطه به تنهایی صحیح باشد ،مثلا عضو تکراری نداشته باشد و کلیدهایش درست باشند و کلید ها دارای مقادیر تهی(null)نباشند.

③ refrential integrity (جامعیت ارجاع):

یعنی کلید خارجی درست تعریف شده باشد و مقداری که به کلید خارجی داده می شود در جدول دیگر وجود داشته باشد.

نرم افزار DBMS همواره خاصیت های ذکر شده بالا را کنترل میکند

جداول نمونه

Stud(s#,sname,city,avg,clg#)
(شماره دانشکده،معدل کل،شهر محل تولد،نام،شماره دانشجو) دانشجو

Prof(pname,office,esp,degree,clg#)
(شماره دانشکده،مدرک تحصیلی،تخصص،دفتر کار،نام استاد) استاد

Crs(c#,cname,unit,clg#)
(شماره دانشکده ارائه دهنده،تعداد واحد،نام،شماره درس)درس

Sec(sec#,c#,s#,term,pname,score)
(نمره،نام استاد،ترم،شماره دانشجو،شماره درس،شماره گروه)گروه درس

Clg(clg#,clgname,city,pname)
(نام رئیس،نام شهر،نام دانشکده،شماره دانشکده)دانشکده

رئیس دانشکده ازبین اساتید انتخاب میشود.

برای ارتباط بین دو جدول باید یک فیلد کلید اصلی در یک جدول و در دیگری فیلد کلید خارجی وجود داشته باشد.

(Stud table)

S#	sname	city	avg	Clg#
71133848	محمدی	تهران	17.24	10
72130502	وکیلی	اصفهان	14.06	10
72203305	علینقی زاده	مشهد	16.42	1
73120504	کمانی	یزد	17.56	4
73166801	احمدی	کرمان	15.44	5
74182532	جوادی	تهران	16.08	5
74209836	حسین زاده	تبریز	12.2	6

(Prof table)

pname	office	esp	degree	Clg#
میرشمسی	4	کامپیوتر	فوق لیسانس	10
ابوطالبی	3	مواد	دکتری	6
قربانی	12	کامپیوتر	دکتری	10
اشرفی زاده	8	شیمی	دکتری	5
هاشمی اصل	10	کامپیوتر	فوق لیسانس	10
جلالی	5	برق	دکتری	7
نقره کار	3	معماری	دکتری	11
حسنی	2	ریاضی	دکتری	1
جاهد مطلق	1	کامپیوتر	دکتری	10
ذاکر	4	فیزیک	دکتری	2
مفتون	1	زبان	دکتری	3
صادقیان	3	صنایع	دکتری	4

(Crs table)

C#	cname	unit	Clg#
10172	شبیه سازی	3	10
10174	مدار منطقی	3	10
12100	معارف 1	2	12
12564	ریاضی عمومی 1	4	1
51516	شیمی آلی	3	5
71203	کنترل خطی	3	7

(Sec table)

Sec#	C#	S#	term	pname	score
1724	10172	71133848	761	هاشمی اصل	14.5
1516	51516	74182532	752	اشرفی زاده	17
1747	10174	71133848	752	میرشمسی	15.75
1747	10174	72130502	752	میرشمسی	12.5
1748	10172	72203305	761	قربانی	16.25

(Clg table)

Clg#	clgname	city	pname
1	ریاضی	تهران	حسنی
10	کامپیوتر	تهران	جاهد مطلق
11	معماری	یزد	نقره کار
12	معارف	تهران	خاتمی
2	فیزیک	مشهد	ذاکر
3	زبان	مشهد	مفتون
4	صنایع	تهران	صادقیان
5	شیمی	تهران	اشرفی زاده
6	مواد	تبریز	ابوطالبی
7	برق	تهران	جلالی

Relational algebra

جبر رابطه ای

نوع داده در جبر رابطه ای فقط و فقط رابطه است.
عملگرها در جبر رابطه ای به چهار دسته تقسیم میشوند.

1- عملگر های ساده شامل گزینش(انتخاب سطر) و پرتو(انتخاب ستون)

σ =select گزینش $\prod$ =project پرتو

2- عملگرهای مجموعه ای

$\cup$ اجتماع
$\cap$ اشتراک
$-$ تفاضل

3- عملگرهای پیوند join

$\times$ ضرب دکارتی
$\bowtie$ Natural join پیوند طبیعی
$\times_\theta$ Theta join پیوند شرطی
$\ltimes$ Semi join نیم پیوند

4- عملگرهای دیگر

ρ نامگذاری
$\leftarrow$ جایگزینی
$\div$ تقسیم

عملگرها یا اصلیند و یا اضافی ،هر عملگر اضافی را می توان با ترکیب عملگرهای اصلی جایگزین کرد بنابر این عملگرهای اضافی حالت کمکی و ساده کردن کار را دارند.

گزینش	پرتو	اجتماع	اشتراک	تفاضل	تقسیم	ضرب دکارتی	پیوند شرطی	پیوند طبیعی	نیم پیوند	نامگذاری	جایگزینی
σ	Π	$\cup$	$\cap$	-	$\div$	$\times$	$\times_{\theta}$	∞	$\propto$	ρ	$\leftarrow$
اصلی	اصلی	اصلی	اضافی	اصلی	اضافی	اصلی	اضافی	اضافی	اضافی	اضافی	اصلی
انتخاب سطر	انتخاب ستون	جدول باید همتا باشد	جدول باید همتا باشد	جدول باید همتا باشد	برای پرس و جوهای دارای شرط همه	زمان و حافظه زیادی می گیرد(گران است)	زیر مجموعه ضرب دکارتی		استفاده در بانک اطلاعات نا متمرکز است	تغییر نام جدول	کپی کردن جدول

مثال:تمام ستونهای جدول دانشجو که شهر آنها کلمه یزد را نشان میدهد؟

$\sigma_{City=\text{" یزد "} \wedge clg\#=4}(stud)$

S#	sname	city	avg	Clg#
73120504	کمانی	یزد	17.56	4

σ (نام جدول)
شرط بر روی سطرها
یا=$\vee$ و=$\wedge$

مثال:ستونهای شماره دانشجو ،نام دانشجو،کد دانشکده از جدول دانشجو ؟

$\Pi_{S\#,sname,clg\#}(stud)$

S#	sname	Clg#
71133848	محمدی	10
72130502	وکیلی	10
72203305	علینقی زاده	1
73120504	کمانی	4
73166801	احمدی	5
74182532	جوادی	5
74209836	حسین زاده	6

Π (نام جدول)
انتخاب ستونهای جدول
بدون هیچگونه شرط

مثال:شهر های مختلف محل تولد دانشجویان ؟

$\Pi_{city}(stud)$

city
تهران
اصفهان
مشهد
یزد
کرمان
تهران
تبریز

مثال:ستونهای شماره دانشجویی،نام،کد دانشکده و میانگین دانشجویانی که معدل آنها بالای 15 است.؟

$\Pi_{S\#,sname,avg,clg\#}(\sigma_{Avg>15}(stud))$

OR

$\sigma_{Avg>15}(\Pi_{S\#,sname,avg,clg\#}(stud))$

S#	sname	avg	Clg#
71133848	محمدی	17.24	10
72203305	علینقی زاده	16.42	1
73120504	کمانی	17.56	4
73166801	احمدی	15.44	5
74182532	جوادی	16.08	5

در عملگرهای مجموعه ای (اجتماع،اشتراک و تفاضل)ورودی هر کدام دو رابطه و خروجی آنها یک رابطه است.روابط ورودی باید همتا باشند(Same arity) یعنی:
- تعداد صفتهای دورابطه (ستونهای دو جدول) مساوی باشند.
- صفتها به ترتیب دارای دامنه های یکسان باشند.

مثال:لیست نام همه افرادیکه در دانشکده هستند.؟

$\Pi_{sname}(stud) \cup \Pi_{pname}(prof)$

sname	pname	pname
محمدی	میرشمسی	حسنی
وکیلی	ابوطالبی	جاهد مطلق
علینقی زاده	قربانی	ذاکر
کمانی	اشرفی زاده	مفتون
احمدی	هاشمی اصل	صادقیان
جوادی	جلالی	
حسین زاده	نقره کار	

مثال:لیست نام اساتیدی که رئیس دانشکده نیستند.؟

$\prod_{pname}(prof) - \prod_{pname}(clg)$

pname
میرشمسی
قربانی
هاشمی اصل
خاتمی

مثال:لیست اسامی دانشجویان و اساتید همنام.؟

$\prod_{sname}(stud) \cap \prod_{pname}(prof)$

نکاتی در ارتباط با عملگر های مجموعه ای

- این عملگر ها را معمولا نمیتوان با با عملگرهای دیگر جایگزین کرد.مثلا آنچه از طریق اجتماع بدست می آوریم نمیتوان از طریق دیگر بدست آورد.

- عملگرهای مجموعه ای به تنهایی ازتوان محاسباتی کافی برخوردار نیستند مثلا نمی توانیم نام و مدرک تحصیلی اساتیدی که مسئولیت ندارند لیست کنیم زیرا در جدول دانشکده مدرک تحصیلی ذکرنشده ،اصولا در جبر رابطه ای توان محاسباتی در ترکیب انواع عملگر ها نهفته است.

ترتیب جداول ورودی فقط در تفاضل مهم است.

$x-y <> y-x \quad x \cup y = y \cup x \quad x \cap y = y \cap x$

(عملگرهای پیوند)

این عملگر ها بسیار قدرتمند و پرکار هستند و بعضا بسیار گران تمام می شوند(زمان و فضای زیادی می گیرند)

(1) (ضرب دکارتی):

ضرب دکارتی دو جدول جدولیست که ستونهایش همه جمع همه ستونهای دو جدول و سطر هایش تمام ترکیبیهای ممکن سطر های آن دو جدول است(ضرب سطرهای دو جدول)،
اگر دو جدول سطونهای همنام داشته باشند در خروجی از نقطه گذاری برای تشخیص آنها استفاده میشود#Crs.clg ╳ #Clg.clg

مثال: Crs ✕ Clg

Crs✕ Clg=(6,4)(10,4)=(60,8)

C#	cname	unit	Crs.clg#	Clg.clg#	clgname	city	pname
71203	کنترل خطی	3	7	1	ریاضی	تهران	حسنی
71203	کنترل خطی	3	7	2	فیزیک	مشهد	ذاکر
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
71203	کنترل خطی	3	7	12	معارف	تهران	خاتمی
10172	شبیه سازی	3	10	1	ریاضی	تهران	حسنی
10172	شبیه سازی	3	10	2	فیزیک	مشهد	ذاکر
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10172	شبیه سازی	3	10	12	معارف	تهران	خاتمی
10174	مدار منطقی	3	10	1	ریاضی	تهران	حسنی
10174	مدار منطقی	3	10	2	فیزیک	مشهد	ذاکر
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
10174	مدار منطقی	3	10	12	معارف	تهران	خاتمی
12100	معارف 1	2	12	1	ریاضی	تهران	حسنی
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(2) (پیوند شرطی): $\times_{\theta}$ Theta join

- این عملگر زیر مجموعه ای از ضرب دکارتی است که شرط θ روی سطرهای آن اعمال شده باشد ،ستونهای خروجی معادل ستونهای ضرب دکارتی است.

مثال:نام و شماره دروسی که توسط استاد قربانی ارائه شده است.

نام و شماره دروس در جدول درس امده است ،این جدول با جدول استاد ارتباط ندارد(هیچکام کلید خارجی در دیگری ندارد)اما با جدول گروه درس ارتباط دارد پس میتوان از پیوند شرطی این دو جدول استفاده کرد.

$\prod_{cname,crs.c\#} (crs \times sec)_{Pname=" \text{قربانی} " \wedge crs.c\#=sec.c\#}$

C#	cname
10172	شبیه سازی

(3) (پیوند طبیعی): Natural join ∞ فیلد مشترک مهم است

همانند پیوند شرطی است با تفاوتهای زیر

1- خود به خود شرط تساوی روی همه ستونهای همنام دو جدول اعمال میشود.یعنی فقط سطرهایی از دو جدول انتخاب میشوند که همه ستونهای همنام آن دو جدول مقادیر مساوی داشته باشند.در صورتیکه دو جدول ستون همنام نداشته باشند نتیجه پیوند طبیعی معادل ضرب دکارتی است.

2- ستونهای تکراری فقط یکبار در خروجی ظاهر میشوند ،از آنجا که فقط مقادیر مساوی آنها انتخاب میشود نیازی به تکرار ستون یا نقطه گذاری مثلا (sec.c#) نیست

3- با توجه به روشهای مرتب کردن و شاخص گذاری نرم افزارهایی شبیه SQL نتیجه پیوند طبیعی عملگر گرانی نیست و زمان و فضای آن قابل قیاس با ضرب دکارتی نمی باشد.

مثال:نام و شماره دروسی که توسط استاد قربانی ارائه شده است.

$(\prod_{C\#} \sigma_{Pname=" \text{قربانی} "} (sec)) \infty (\prod_{C\#,cname} (crs))$

مثال:مشخصات کامل روسای دانشکده ها

$(\prod_{pname} (clg)) \infty Prof$

مثال:خروجی دستور زیر چیست؟ Stud ∞ Clg

جواب:کسانی که شهر دانشکده آنها با شهر تولد آنها یکی باشد.

دو صفت مشترک Clg ∞ Prof تطبیق داده میشود.در پیوند طبیعی مساوی بودن همه صفتهای مشترک بررسی میشود

مثال:خروجی دستور زیر چیست؟ Clg ∞ Prof

هر یک از دو جدول یک کلید خارجی در دیگری دارد ،بنابراین در پیوند طبیعی آنها شرط تساوی هر ستون رعایت میشود.چون ستون pnameدر جدول Clg مربوط به رئیس دانشکده است پس دستور فوق مشخصات کامل دانشکده ها و روسای آنها را می دهد.

مثال:نام دانشجویانی که در دروس دانشکده های دیگر نیفتاده اند؟

به هیچ وجه نمی توان از دستور زیر استفاده نمود زیرا وجود ستونهای همنام #Clg در دو جدول Crs,Stud باعث حذف دروس دانشکده های دیگر میشود.

$\prod_{.....} \sigma_{.....} (stud \infty crs \infty sec)$

دستور زیر نیز غلط است پس سئوال جواب ندارد

$\prod_{sname} \sigma_{Score>10} ((stud \times crs) \infty sec)_{Stud.clg \neq crs.clg\#}$

④ **(نیم پیوند):** $\propto$ Semi- join

همانند پیوند طبیعی است با این تفاوت که فقط ستونهای جدول اول را می دهد.

مثال :خروجی دستور زیر چیست؟

$$\sigma_{Clg\#=1 \wedge Term=771}(crs \propto sec)$$

دستور فوق غلط است زیرا ستون term مربوط به جدول crs نیست و پس از نیم پیوند حذف میشود ،پس نمیتوان در شرط گنجاند

یکی از دستورات صحیح$(\sigma_{Clg\#=1}(crs)) \propto (\sigma_{Term=771}(sec))$

نکاتی در مورد نیم پیوند:

- ممکن است تعداد سطرهای خروجی به مراتب کمتر از پیوند طبیعی باشد زیرا با کنار رفتن چند ستون ،سطر های تکراری پدید می آیند و حذف می شوند.
-برخلاف سایر عملگر ها،ترتیب جداول ورودی در نیم پیوند مهم است زیرا همیشه ستونهای جداول اول را می دهد. $(x \propto y \neq y \propto x)$

کاربرد نیم پیوند در بانکهای نامتمرکز است اگر جدول X در سایت الف و جدول Y در سایت ب ذخیره شده باشد و دستور $x \propto y$ را در سایت الف صادر کنیم از انتقال داده های جدول Y از سایت ب به سایت الف پرهیز می کند.

عملگرهای دیگر

نامگذاری: ρ_b^a

با این نامگذاری نام b روی جدول a گذاشته میشود.

ارزش عملیاتی آن اینست که بدون ذخیره سازی مجدد یک جدول میتوان از آن دو بار استفاده کرد.(اشاره گر جدید تعریف میشود)

$$Prof \underset{Prof.office=p.office \wedge Prof.pname \neq p.pname}{\times} (\rho_p(\prod_{Pname,office}(prof)))$$

نکته:همه اساتید با خودشان هم اتاق هستند

جایگزینی: $\longleftarrow$

جدول حاصل از دستورات ذخیره می گردد تا در ادامه مورد استفاده قرار بگیرد.

مثال:مشخصات دروسی که توسط استاد قربانی ارائه شده است.

$$Temp \longleftarrow \prod_{C\#}\sigma_{Pname=\text{"قربانی"}}(sec)$$

$$Temp \infty\infty Crs$$

تقسیم: $\div$

ابتدا بخشی که شامل همه است را پیدا میکنیم (مقسوم علیه)سپس بخش دیگر را برآن تقسیم می کنیم ،نکته اینکه باید فیلد مشترک بین مقسوم و مقسوم علیه وجود داشته باشد.

مثال:دانشجویانی که همه درسهای استاد تبریزی را گرفته اند.

$$Temp \longleftarrow \prod_{C\#}\sigma_{Pname=\text{"تبریزی"}}(sec)$$

$$stud \div temp$$

دستور $stud \div temp$ غلط است زیرا صفت C# در stud وجود ندارد.

پاسخ صحیح به شکل زیر است.

$$\prod_{C\#,sname,s\#}(stud \infty\infty sec) \div temp$$

$$\frac{C\#,sname,s\#}{C\#} = sname,s\#$$

مثال:درسهایی که توسط همه دانشکده ها ارائه میشوند.

ابتدا همه دانشکده ها $Temp \longleftarrow \prod_{Clg\#}(Clg)$

$Crs \div Temp$