

## امنیت و جامعیت

### امنیت

امنیت یا Security حفاظت از داده ها در مقابل خطراتی از قبیل آتش سوزی و نیز جلوگیری از دستیابی غیر مجاز انسان ها به داده ها می باشد .  
تامین امنیت داده ها بسیار مشکل بوده و امنیت 100% را هیچ گاه نمی توان انتظار داشت ولی راه هایی وجود دارد که تا حد امکان امنیت را تامین نمود .  
مثلا در یک کاربرد مانند بانک ها اگر داده های خیلی پرارزشی داریم می توان آنها را در گاوصندوق ها نگهداری نمود تا از آتش سوزی در امان باشند و یا اینکه نسخه های مختلف این داده ها را در جاهی مختلف و شعب متفاوت نگهداری نمود ، تا اگر در جایی مورد دستبرد قرار گرفت در جای دیگر قابل بازیابی باشد .  
بخش دیگر مساله امنیت یا Security این است که افراد غیر مجاز نتوانند به داده های دسترسی پیدا کنند .

### افراد غیر مجاز عبارتند از :

- افراد خارج سیستم
- افراد داخل سیستم : یک دانشجو بخشی از کاربران یک بانک اطلاعات است ولی دستیابی ایشان به داده ها با یک استاد و یک مسئول آموزش متفاوت می باشد . دانشجو حق تغییر ستون نمره را ندارد . حتی یک دانشجو نباید به نمره های دیگر دانشجویان دسترسی داشته باشد .

### جامعیت یا Integrity

جامعیت به معنای صحت داده ها و پیروی از مقررات در پردازش داده ها می باشد .  
مثلا اگر ستونی به نام ستون نمره دانشجو داریم و این ستون عددی بین صفر تا بیست می باشد ، در این صورت عدد 22 در اینجا معنا ندارد .  
بنابراین با مکانیزم هایی می توان به DBMS گفت که این ستون عددی بین 0 تا 20 می باشد و اگر داده دیگری کسی وارد کرد مورد پذیرش قرار نمی گیرد .  
یکی از بخش های جامعیت همخوانی یا Consistency می باشد :  
مثلا ممکن است در یک بانک موجودی یک حساب درست باشد یعنی منفی نباشد اما زمانی که به یک شعبه رفته و پولی از حساب برداشت می شود . مبلغی از حساب کسر می گردد ، بلافاصله شعبه ای که در آن هستید موجودی جدید را نشان می دهد ولی اگر در همین لحظه در یک شهرستان با کارت عابر بان به حساب خود دسترسی پیدا کنید ، آیا موجودی قبلی را نشان می دهد و یا اینکه بلافاصله این موجودی به روز شده است ؟  
این مفهوم همخوانی یا Consistency نام دارد .  
باید تمام نسخه های داده ها هر جا یک داده ای تکرار شده همه یک چیز را نشان دهند و با هم همخوانی داشته باشند .

## تراکنش (Transaction)



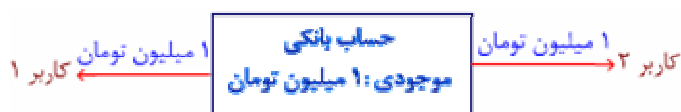
تراکنش (Transaction)

برنامه در محیط بانک اطلاعات



سوال

وقتی DBMS درخواست های کاربران را دریافت می کند چه عملی انجام می دهد؟  
DBMS کنترل می کند که آیا این درخواست ها مجاز می باشد یا خیر.



DBMS باید کنترل هایی را روی تراکنش ها انجام دهد تا جامعیت زیر سوال نرود.



سوال چه کنترل هایی را روی تراکنش ها انجام دهد تا جامعیت زیر سوال نرود؟

آقای جیم گری (Jim Gary) در سال ۱۹۸۱ با استفاده از روش های ریاضی ثابت نمود که ۴ کنترل لازم است روی تمامی تراکنش ها در بانک اطلاعات اعمال گردد تا صحت و



جامعیت آن تضمین گردد.

تا قبل از این زمان هر کس کنترل هایی را برای سیستم خود به صورت شخصی تعریف می نمود.  
این ۴ خواص به خواص ACID شهرت دارند و عبارتند از:

- یکپارچگی (Atomicity)
- هم خوانی (Consistency)
- انزوا (Isolation)
- پایایی (Durability)

یکپارچگی (Atomicity)

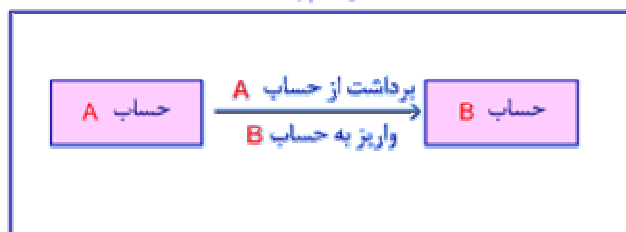


یکپارچگی (Atomicity) یعنی همه یا هیچ

یک تراکنش یک برنامه است و برنامه مجموعه ای از دستور ها می باشد.


باید اطمینان حاصل شود که یا همه دستور ها انجام می گردد یا هیچ یک انجام نمی گردد.

سیستم بانک



سوال آیا در صورت عدم توانایی انجام یک برنامه می توان کارهای انجام شده را باز گرداند؟

بله- در محیط های **DBMS** همواره می توان این عمل را انجام داد .  
از این مفهوم به عنوان بازگرد (**Roll Back**) یاد می گردد.

 **نکته** روش های گوناگونی برای عمل باز گرد (**Roll Back**) وجود دارد مانند:

- **روش کارنامه ای ( Log Based )**

بدین صورت که از تمام کارهای انجام شده در سیستم اطلاعاتی را در کارنامه ذخیره می نمایند به طوری که همواره بتوان وضعیت قبلی سیستم را ایجاد کرد .

- **روش سایه ( Shadow Based )**

بدین صورت که اگر بخواهیم به یک **Page** از دیسک دسترسی داشته باشیم آن را کپی کرده وبه **Page** های اصلی دسترسی نداشته و آن ها را تغییر نمی دهیم و تمام تغییرات روی کپی ها انجام می شود . در صورت انجام تراکنش با موفقیت کپی روی **Page** اصلی قرار داده می شود .

اما در صورتی که تراکنش به مرحله **Commit** نرسید همان **Page** قبلی نگه داشته می شود .  
یک تراکنش دو نوع پایان دارد :

- پایان ناموفق (**Abort**) یا سقوط

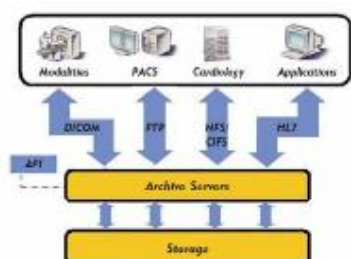
- پایان موفق (**Commit**) یا انجام

اگر تراکنش به پایان موفق برسد کارش انجام شده است و اگر به پایان ناموفق برسد کاری انجام نداده زیرا از عمل بازگرد استفاده می شود .

## هم خوانی (Consistency)

این خاصیت بر مبنای یک فرض می باشد و تلاش DBMS و DBA بر این است که این فرض برقرار باشد.

این فرض این چنین است :



اگر تراکنشی به مرحله **Commit** رسید کارهایی را که این تراکنش به تنهایی انجام داده جامعیت سیستم را بر هم نزند.

اگر تراکنشی به پایان ناموفق رسید کاری را انجام نداده است.

پایان ناموفق یا **Abort** مانند این است که این تراکنش هرگز شروع نشده است.

بنابراین

اگر تراکنشی به پایان موفق رسید باید کنترل شود که جامعیت را بر هم نزند.



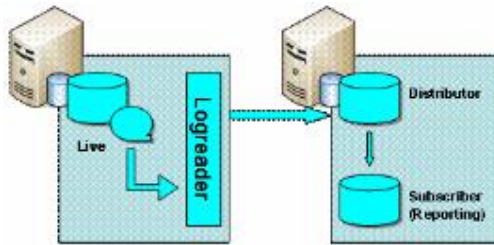
نمره ۲۲ یا ویروس

با مکانیزم های مختلفی می توان این کنترل را انجام داد اما هنوز **DBMS** ها به جواب

۱۰۰٪ نرسیده اند. اما می توان قیود جامعیتی یا برای سیستم تعریف کرد.

## انزوا (Isolation)

زمانی مشکل ایجاد می شود که چند برنامه به طور هم زمان روی داده هایی کار کنند.



نمره دانشجو ← تراکنش مسئول آموزش  
تراکنش استاد



سوال کدام تراکنش مقدم تر است ؟

آیا ترتیبی وجود دارد ؟

در اینجا ترتیب اهمیت دارد و همواره سیستم باید بداند که از کدام نسخه داده ها که در حال حاضر در سیستم وجود دارد استفاده نموده است. این مفهوم انزوا (Isolation) نام دارد .



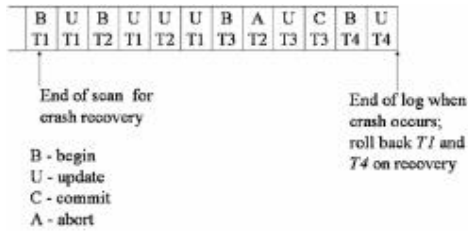
طبق این تعریف اثر تراکنش های همروند روی یکدیگر چنان است که گویا هر کدام در انزوا انجام می شود .

بدین معنا که اگر تراکنشی در حال حاضر روی داده هایی کار می کند تراکنش های همروند دیگر تاثیر مخرب روی این تراکنش ندارند .

انجام این مفهوم کنترل همروندی (Concurrency Control) نام دارد .

**پایایی (Durability)**

آقای جیم گری ثابت کرد این آخرین خاصیتی است که اگر همه تراکنش ها داشته باشند جامعیت سیستم زیر سوال نمی رود .



**نکته**

اگر تراکنشی تغییری را در سیستم انجام داد و تراکنش **Commit** شد از آن لحظه تغییراتی که تراکنش در سیستم انجام داده است نباید دستخوش حوادث شود. تنها چیزهایی که می توانند داده ها را دستخوش تغییر نمایند یا اثری روی بانک های اطلاعات **DBMS** ها داشته باشند تراکنش ها هستند.

### خاصیت پایایی (Durability)

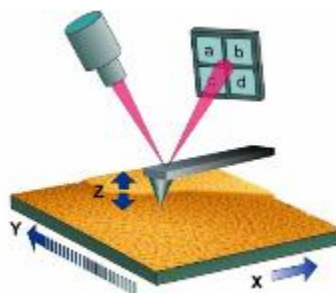
تاثیر یک تراکنش روی بانک های اطلاعاتی فقط و فقط توسط تراکنش های دیگر قابل تغییر می باشد.

## تصویر ادراکی بانک اطلاعات ( Database Schema )

فرض کنید در مدل رابطه ای هستیم یعنی مکانیزم انتزاعی جدول به هم پیوسته می باشد.

دانشجو  
stud

شماره دانشجو	نام دانشجو	شهر محل تولد	معدل کل	شماره دانشکده
s #	sname	city	avg	clg#




ساختاری که نشان دهنده وجود جدول بوده و داده ای در آن نیست Database Schema نام دارد .  
انواع Schema عبارتند از :

- تصویر خارجی ( External Schema )
- تصویر ادراکی ( Conceptual Schema )
- تصویر فیزیکی ( Physical Schema )

### استقلال داده ها ( Data Dependency )

زمانی که در برنامه نویسی معمولی پرونده ای باز می گردد و برنامه ای برای خواندن و استخراج اطلاعات از این پرونده نوشته می شود برنامه به پرونده وابسته می باشد .  
بدین معنی که

اگر تغییری در پرونده انجام شود روی برنامه تاثیر می گذارد و حتماً برنامه نیز باید اصلاح گردد .

 **نکته** به طور کلی تغییر برنامه ها کار بسیار سختی می باشد .  
بنابراین

باید روشی اتخاذ نمود که نیاز به تغییر برنامه ها نباشد .  
در سیستم بانک اطلاعات اگر پرونده ها تغییر نماید فیلدهای جداول را بر اساس پرونده جدید پر می کند .

بنابراین  
برنامه کاربر دستخوش تغییر نمی گردد .

 **نکته**  
کاربران تنها زمانی مجبور به تغییر برنامه خود می باشند که دیدهای خود را تغییر دهند .

## زبان های برنامه سازی بانک اطلاعات

بانک اطلاعات برای بهتر در ارتباط قرار دادن کاربران مختلف با **DBMS** از یک **Interface** استفاده می کند که بسیار قدرتمند می باشد و این چهره واسط یک زبان می باشد .  
زبان مشترکی بین کاربران و **DBMS** وجود دارد که کاربران بتوانند نیاز های خود را به راحتی به **DBMS** در خواست نمایند .



این زبان با زبان های متداول برنامه سازی متفاوت است و عمده ترین تفاوت آن عبارت است از :  
● در زبان های برنامه سازی معمولی که از آن ها به زبان های برنامه سازی روالی (**Procedural**) یاد می گردد روال کار را برنامه ساز تعیین می نماید .

زبان های بانک اطلاعات نیازی به این مسائل ندارند به همین دلیل به این زبان ها زبان های بیانی (**Declarative**) گفته میشود و کاربران تنها نیازهای خود را بیان می کنند و نیازی به بیان روال انجام کار ندارند

● در زبان های بیانی (**Declarative**) حلقه تکرار (**Loop**) نداریم .



مهم ترین زبان بیانی **SQL** می باشد .

در **SQL3** همه توان های روالی و بیانی با هم وجود دارد .

زیرا یک از مشکلات زبان های بیانی این است که در صورت نیاز نمی توان کارهای زبان های روالی را انجام داد و زبان های بیانی هم قدرت کافی ندارند به همین جهت نمی توان همه اطلاعات موجود در بانک اطلاعات را استخراج نمود .





تعریف

در گذشته این زبان ها را به صورت توکار در زبان های برنامه سازی استفاده می شد و برای دسترسی به بانک اطلاعات با فراخوانی SQL اعمال مورد نیاز انجام می گردید . سپس نتیجه خروجی SQL در جداول قرار داده شده و در اختیار زبان برنامه سازی مورد نظر قرار داده می شد .

این مفهوم زبان توکار ( Embedded Language ) نام دارد .

یکی از اشکالات Embedding عبارت است از اینکه

طبیعت زبان های روالی با زبان های بیانی هم خوانی ندارد که به این مشکل نا همگونی مانع ( Impedance Mismatch ) گفته می شود .

زبان های بیانی شامل دو بخش هستند :

- Data Declaration Language یا DDL ← شما ها
- Data Manipulation Language یا DML ← داده ها

### طراحی بانک اطلاعات

ابزار متفاوتی در اختیار است که بتوان بانک اطلاعات را طراحی کرد .



سوال آیا طراحی بانک اطلاعات برای پیاده سازی خاصی است ؟

- اگر پاسخ منفی باشد با ابزار هایی مانند UML و ER و EER بانک اطلاعات طراحی می گردد تا دیگران از آن استفاده کرده و آن را پیاده سازی کنند .
- اگر پاسخ مثبت باشد بانک اطلاعات در همان مدل خاص پیاده سازی می شود



معمولاً بانک اطلاعات به عنوان بخشی از سیستم نرم افزاری مطرح می باشد و باید تمام مراحل طراحی نرم افزار از قبیل:

- بررسی و تجزیه و تحلیل
  - طراحی نرم افزار
- را پشت سر گذاشته باشیم .