

پایگاه داده‌ها



فصل اول: بانک اطلاعات چیست؟

آیا به نظر شما بانک اطلاعات سیستمی است برای ذخیره داده ها و ایندکس گذاری روی رکوردها که جستجو را تسهیل می کند؟ آیا بانک اطلاعات صرفاً به داده هایی گفته می شود که به صورت منطقی در کنار هم و با ارتباطی خاص قرار گرفته اند؟ اگر پاسخ شما به این سوالات مثبت باشد یا اگر مواردی از این قبیل را به عنوان وظایف اصلی یک سیستم مدیریت بانک اطلاعات (DBMS) در نظر می گیرید، در اشتباه هستید.

تا قبل از سال ۱۹۷۰ که دهه انقلابی به نام انقلاب بانک اطلاعات بود، سیستمهایی توسعه یافته بودند که کارهای بالا را به خوبی انجام می دادند و شما احتمالاً با آنها در درس ذخیره و بازیابی اطلاعات آشنا شده اید. مانند بسته محبوب btrieve که یک سیستم فایلینگ قوی در سیستم عامل unix یا novell بود. اما کجای کار مشکل داشت و این سیستمها چه امکاناتی را پشتیبانی نمی کردند که باعث شد ناکارآمد جلوه کنند و موجودی به نام بانک اطلاعات پا به عرصه وجود بگذارد.

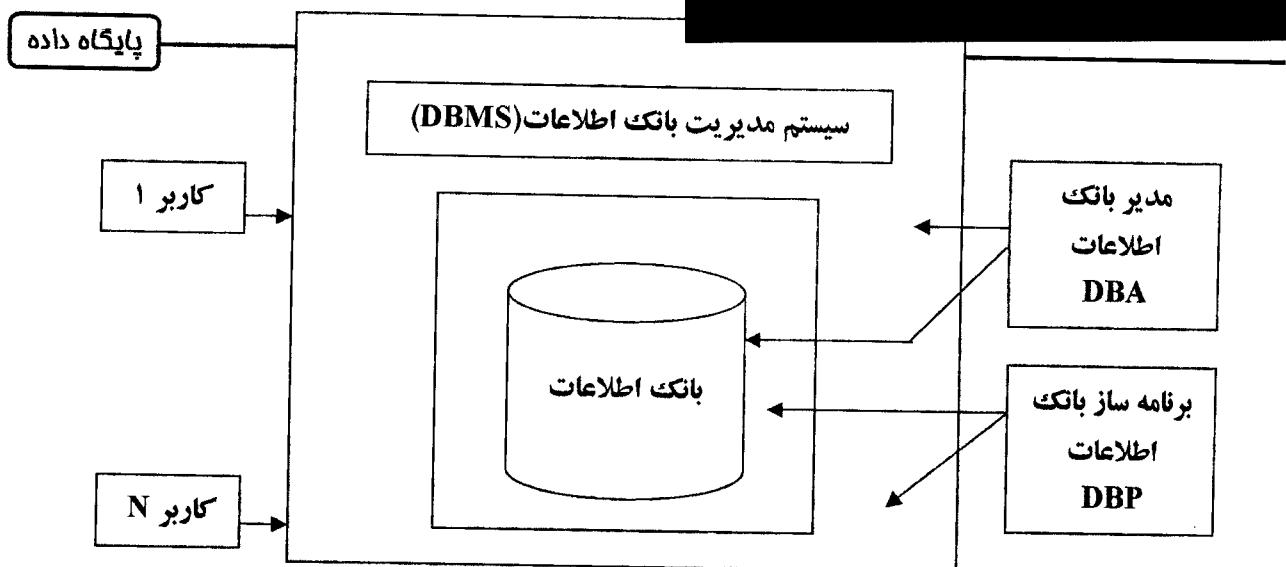
در سیستمهای فایلینگ هر کاربر می توانست روی کل پرونده مسلط شود و از اطلاعات آن استفاده کند. این مساله ۲ مشکل بزرگ را به همراه داشت: اول اینکه در سیستمهای چند کاربره، کاربران دیگری که با یک پرونده کار داشتند بایست منتظر می ماندند تا کاربر اول کار خود را تمام کند و پرونده را آزاد کند و این بازده و سرعت را پایین می آورد. از طرف دیگر تسلط کاربر بر کل پرونده یعنی قربانی شدن امنیت. این مشکلات و مشکلات دیگری مانند افزونگی بیش از حد داده ها باعث شدند تا سیستمهای مدیریت بانک اطلاعات (DBMS) ها و مفاهیم بانک اطلاعات به موازات آن رشد پیدا کنند. در واقع DBMS برنامه ای است که همه پرونده ها را در اختیار خود می گیرد و همه کاربران بدون استثناء می بایست برای دسترسی به داده ها در خواستهای خود را از طریق DBMS بدهند و ایشان اگر صلاح دیدند به درخواستها پاسخ مثبت می دهند.

به این ترتیب بسیاری از مشکلات حل شد. امنیت به سطح قابل قبولی رسید. زیرا DBMS مفت به کسی اطلاعات نمی دهد!!! تا کاربر اجازه دسترسی به یک جدول را نداشته باشد حصار بلند به نام DBMS به او اجازه نمی دهد که از داده ها استفاده کند. از طرفی دیگر کاربران متعددی می توانند روی داده ها به طور همزمان کار کنند و DBMS یکپارچگی و صحت داده ها را در بانک اطلاعات تضمین خواهد کرد. با ظهور DBMS بسیاری از وظایف از دوش برنامه نویس برداشته شد. به عنوان مثال کنترل ورود صحیح اطلاعات حفظ یکپارچگی و البته کلیه عملیات مربوط به دستکاری داده ها بر دوش DBMS گذاشته شده است. زبان های برنامه نویسی امروزی نیز با قرار دادن کامپونتهای مختص بانک اطلاعاتی به این امر کمک کرده اند. به طوریکه با چند خط برنامه نویسی در محیطی مثل Delphi می توان یک پروژه بانک اطلاعاتی را توسعه داد.

DBMS از کجا می داند که به چه کسی باید چقدر اطلاعات بدهد و به چه اطلاعات ندهد؟ یا اصلاً بانک اطلاعات چگونه روی DBMS قرار می گیرد؟ اینجاست که با نیروی انسانی درگیر هستیم:

مدیر بانک اطلاعات (DBA): مسئول طراحی و تصمیم گیری برای کلیه موارد بالا، مدیر بانک اطلاعات است. در بانک های توزیع شده (بانک هایی که از لحاظ فیزیکی در یک محل قرار ندارند) ممکن است یک تیم مسئول مدیریت بانک اطلاعات باشد.

برنامه ساز بانک اطلاعات (DBP): این گروه افراد مسئول ساختن برنامه هایی هستند که از یک طرف به بانک اطلاعات متصل است و از طرف دیگر به کاربر نهایی یا همان اپراتور. در واقع این افراد تصمیمات مدیر را پیاده سازی می کنند. کاربران کسانی هستند که از طریق این برنامه ها داده ها را در حیطه نظارت DBMS دستکاری می نمایند.



معماری بانک اطلاعات:

بعد از سال ها بحث پیرامون اینکه بانک اطلاعات چه هست و چه نیست، سرانجام کمیته ANSI/SPARC معماری ۳ لایه را ارائه داد که بعد ها یک لایه به آن افزودند و ما معماری ۴ لایه را بررسی می کنیم. این معماری قابل تطبیق با اکثر مدل های بانک اطلاعات که بعدا بررسی خواهیم کرد، می باشد.

لایه اول لایه تصویر خارجی است. تصویر خارجی مربوط به دید کاربران است. یعنی اینکه هر کاربر چه قسمتهایی از بانک اطلاعات را اجازه دارد ببیند و چه کارهایی روی آن قسمتها می تواند انجام دهد.

اصل اول بانک اطلاعات: این اصل می گوید به هر کس همان مقدار اطلاعات بده که لازم دارد نه بیشتر. در زندگی روزمره هم رعایت این اصل بسیار مفید است. همیشه اطلاعات زیادی به کسی دادن باعث دردسر است. بنابراین در این لایه قسمتهایی از بانک اطلاعات که هر کاربر می بیند مورد نظر است.

لایه دوم لایه تصویر ادراکی عام است. تصویر ادراکی عام یعنی طراحی بانک اطلاعات بدون وابستگی به مدل خاص و پیاده سازی فیزیکی خاص. این لایه را کاربر نهایی نمی بیند. (بسیار مهم)

فقط مدیر بانک و برنامه نویس هستند که این لایه به دردشان می خورد.

لایه سوم لایه تصویر ادراکی خاص یا همان مدل منطقی است. یعنی اینکه داده ها به صورت منطقی چگونه کنار هم قرار می گیرند. مدل های مرسوم جدول، درخت گراف و مانند این هاست.

لایه چهارم لایه تصویر فیزیکی است. یعنی اینکه داده های به صورت فیزیکی چگونه روی دیسک یا هر رسانه ذخیره و بازیابی دیگری قرار می گیرند.

انواع مدل های منطقی:

۱) مدل های قدیمی: شامل مدل شبکه ای و مدل سلسله مراتبی است. مدل شبکه ای داده ها را به صورت رکورد هایی می بیند که با هم

تشکیل یک گراف می دهند و مدل سلسله مراتبی داده ها را رکورد های تشکیل دهنده یک درخت می بیند.

۲) مدل های جدید: شامل مدل جدولی است. داده ها را به صورت رکورد هایی می بیند که تشکیل یک جدول یا رابطه می دهند.



۳) مدل های جدید: مدل هایی مانند شیء گرا، شیء رابطه ای و منطقی و استنتاجی که برای پوشش دادن ضعف های مدل قدرتمند رابطه ای عرضه شده اند. از عمده ترین این ضعف ها می توان به بحث ذخیره و بازیابی multimedia یا چند رسانه ای اشاره کرد.

استقلال داده ها:

همانطور که مشاهده کردید، مدل های منطقی گوناگونی برای ذخیره سازی داده ها وجود دارد. به عنوان مثال در مدل رابطه ای داده ها مستقل از چه نوعی باشند در جداول ذخیره می شوند. بدیهی است که ذخیره سازی فیزیکی داده ها روی دیسک یا هر رسانه دیگر به شکل جدول نخواهد بود و استقرار فیزیکی داده ها از دید کاربر مخفی است. هر تغییری که در نوع ذخیره سازی داده ها به وجود آید مثلا اگر به جای دیسک از نوار استفاده شود، کاربر این تغییر را احساس نخواهد کرد چون مثل قبل او فقط جداول را می بیند. این نوع استقلال فیزیکی داده هاست. در مقابل همانطور که قبلا هم اشاره شد، تصویر ادراکی عام از دید کاربر مخفی است. به عنوان مثال اگر به جدولی ستونی اضافه شود یا به Schema یک جدول اضافه شود برنامه های قبلی باز هم به درستی کار خواهند کرد و نیازی به دستکاری ندارند. حتی دیدهای سابق (view) به کار قبلی خود ادامه می دهند. این استقلال استقلال منطقی داده ها نامیده می شود.

دو ابزار مهم در بانک اطلاعات:

از جمله ابزارهایی که در سیستمهای فایلینگ وجود ندارد اما بانک اطلاعات آنها را پشتیبانی می کند، لغتنامه داده و کاتالوگ سیستم است. لغتنامه داده: در یک سیستم بانک اطلاعات اسامی زیادی مورد استفاده قرار می گیرند. از آنجا که افراد زیادی درگیر با بانک اطلاعات هستند به خصوص اگر بانک اطلاعات به صورت توزیع شده. در نقاط مختلف باشد، برای ایجاد یکنواختی و هماهنگی مرجعی به نام لغتنامه داده (data dictionary) پدید آمده است. این لغتنامه مانند یک لغتنامه معمولی شامل تمامی اسامی و معانی آنها در بانک است. نرم افزارهای بسیار پیشرفته ای نیز وجود دارند که از استفاده اسامی مختلف برای یک معنای مختلف برای یک اسم جلوگیری می کنند. فرض کنید در یک جدول از Account-number به عنوان نام فیلد شماره حساب استفاده شده است. نرم افزار اجازه نمی دهد در جای دیگری از این سیستم از نام Account# برای شماره حساب استفاده شود.

کاتالوگ سیستم: اضافه بر آنچه گفته شد، اندازه جداول، ساینز فایلها، لایه خارجی بانک، زمان و تاریخ ایجاد هر یک از جداول، نام ایجاد کننده و ... همه در کاتالوگ سیستم ذخیره می شود.

تراکنش (Transaction):

یکی از ویژگیهای فوق العاده مهم بانک اطلاعات است که سیستم های فایلینگ مثل btrieve فاقد آن هستند.

$$\text{Transaction} = \text{Query} + \text{Update}$$

این ساده ترین تعریف برای تراکنش است. فرق اساسی تراکنش با سایر برنامه ها این است که اجرای تراکنش بر گردن DBMS گذاشته می شود و DBMS مسؤل اجرا یا ساقط کردن و یا به تعویق انداختن تراکنش است. سوالی که اینجا مطرح می شود این است که چرا مجموعه دستورات کاربر باید به تعویق انداخته شود؟ پاسخ این است که همه این وضعیت ها برای این اتفاق می افتد که جامعیت و صحت داده ها حفظ شود.



تراکنش یا به طور کامل انجام می شود و یا ساقط می شود و اصلا انجام نمی شود. مثال معروف برای تراکنش این است که یک تراکنش می خواهد از حسابی پولی برداشت کند و همان پول را به حساب دیگری واریز کند. یک برنامه معمولی این کار را در ۲ مرحله مجزا انجام می دهد یعنی ابتدا عمل برداشت را انجام می دهد و سپس به حساب دیگری واریز می کند. فرض کنید که در اثنای این کار یعنی درست بعد از اینکه از حساب اول پول برداشت شده، مشکلی در سیستم رخ بدهد و برنامه از ادامه کار خود باز بماند چه اتفاقی می افتد؟ مبلغی پول از حسابی برداشت شده و گم شده است. مفهوم تراکنش می گوید یا هر ۲ عمل برداشت و واریز انجام شود و یا هیچ کدام انجام نشود. بنابراین اگر در حین انجام کار به دلیلی مثل قطع برق یا خرابی دیسک و ... تراکنش به مرحله انجام نرسد، کلیه اعمال انجام شده تا آن لحظه به حالت قبل از انجام تراکنش بر میگردد. و تراکنش بار دیگر از نو انجام خواهد شد. آنقدر این حلقه تکرار می شود تا تراکنش با موفقیت به انجام برسد.

نکته: در تئوری بانک اطلاعات اصالت با داده است نه برنامه. بدین مفهوم که برنامه ها می آیند و می روند. اما مهم این است که داده ها صحیح باشند و کلیه بحث های مربوط به تراکنش مثل همروندی و بازگرد (دو واحد مهم در DBMS) ناشی از همین اصل مهم است. به همین خاطر است که در بانک اطلاعات داده را مانا (Persistant) می گویند.

ثابت می شود که ۴ کنترل ACID برای تضمین صحت و جامعیت اطلاعات کفایت می کند که هر یک را جداگانه شرح می دهیم:

یکپارچگی (Atomicity): بدین مفهوم است که امکان ندارد بخشی از دستورات اجرا شود و بخش دیگر باقی بماند. یا همه دستورات یک تراکنش اجرا می شود و یا هیچ کدام از دستورات اجرا نمی شود. بهترین مثال همان مثال برداشت از یک حساب و واریز به حساب دیگر است. همخوانی (Consistency): بدین مفهوم است که در دستورات تراکنش باید قوانین جامعیت رعایت شود و نتیجه هر تراکنش صحیح باشد.

مثال: تراکنش زیر مربوط به برداشت ۵۰۰۰۰ تومان از یک حساب و واریز آن به حساب دیگر است. خاصیت همخوانی در این مثال به این مفهوم است که جمع مبلغ برداشتی و واریزی باشد صفر باشد. یا به عبارت دیگر مجموع موجودی حسابهای A و B باید ثابت باشد. البته همانطور که مشاهده می شود این کنترل بیشتر به عهده طراح تراکنش است تا DBMS.

T_i: Read(A)
 A:=A-50000
 Write(A)
 Read(B)
 B:= B + 50000
 Write(B)

انزوا (Isolation): به این معناست که هر تراکنشی از تراکنش های دیگر مستقل است و اجرای تراکنش های مختلف نباید روی هم اثری داشته باشد. به عبارت دیگر هر تراکنش طوری اجرا می شود که انگار بقیه تراکنش ها اصلا وجود ندارند.

پایایی (Durability): این خاصیت باعث می شود که اگر تراکنش تا مرحله انجام یا Commit رسید، کاربر مطمئن باشد که همه تغییرات اعمال شده و هیچ نقصانی مانند قطع برق یا مشکل سخت افزاری باعث از بین رفتن اطلاعات نخواهد شد. در واقع در این مرحله تغییرات در جایی ثبت شده و اگر پایان تراکنش با موفقیت اعلام نشود، اطلاعات کاربر در جایی ثبت شده است. راجع به پیاده سازی دو خاصیت اخیر بعدا بیشتر بحث خواهد شد.



پرسش های ۴ گزینه ای:

۱) بحث های امنیتی کاربران مربوط به کدام لایه معماری بانک اطلاعات است؟ (مولف)

- الف) لایه خارجی
- ب) لایه منطقی
- ج) لایه فیزیکی
- د) لایه تصویر ادراکی عام

۲) کدامیک از وظایف اصلی DBMS محسوب نمی شود؟ (کامپیوتر ۷۲)

- الف) امکان پردازش زبان طبیعی برای کار با پایگاه داده
- ب) امکان کار با داده ها به کمک یک DSL (Data Sub Language)
- ج) تامین امنیت
- د) تامین جامعیت

۳) مزایای یک پایگاه داده نسبت به فایل های متعارف چیست؟ (مهندسی کامپیوتر ۷۲)

- الف) کنترل حساب شده مقدار افزونگی داده در پایگاه داده
- ب) اطمینان از صحت داده ها (Data validation) کمتر مورد نیاز خواهد بود.
- ج) دستیابی مشترک به داده ها
- د) ۱ و ۳

۴) کدامیک جزء وظایف DBA نیست؟ (مهندسی کامپیوتر ۷۲)

- الف) کنترل معناها در لغتنامه داده
- ب) نظارت بر عملکرد پایگاه داده
- ج) تهیه رویه و استراتژی تهیه Back-up و نحوه احیاء (recovery) پایگاه داده
- د) تهیه Schema برای پایگاه داده

۵) تصویر ادراکی یعنی: (مولف)

- الف) دید کاربر از بخشی از پایگاه داده
- ب) پایگاه داده از دید کاربر
- ج) دید DBP از پایگاه داده
- د) دید DBA از پایگاه داده

۶) قابلیت ایندکس گذاری روی کل متن مربوط به کدام لایه معماری بانک اطلاعات است؟ (مولف)

- الف) تصویر خارجی
- ب) طراحی منطقی
- ج) تصویر ادراکی عام
- د) تصویر فیزیکی

۷) کدامیک از وظایف DBMS نیست؟ (مولف)

- الف) کنترل همزمانی
- ب) پردازش پرس و جو
- ج) نرمال سازی پایگاه داده
- د) اعتبار



۸) کدام گزینه از وظایف DBA نیست؟ (مولف)

الف) تغییر شما (Schema) پایگاه داده

ج) کنترل همزمانی

ب) تغییر نوع داده ها

د) تعیین نحوه ذخیره سازی

۹) جمله زیر به کدام خاصیت تراکنش مربوط می شود؟ (مولف)

"حداقل موجودی حساب ها ۵۰۰۰ تومان است."

الف) A

ب) C

ج) I

د) D

۱۰) از مدل و بستر فیزیکی مستقل است و بر اساس مدلی خاص انجام می شود. (مولف)

الف) طراحی ادراکی - طراحی منطقی

ج) طراحی فیزیکی - طراحی منطقی

ب) طراحی ادراکی - طراحی منطقی

د) طراحی فیزیکی - طراحی منطقی

۱۱) مفهوم داده مانا چیست؟

الف) پس از خرابی دیسک می ماند.

ج) پس از اتمام برنامه های مربوطه می ماند.

ب) پس از خرابی حافظه اصلی می ماند.

د) پس از سقوط تراکنش می ماند.

۱۲) کدام مورد شامل استقلال داده ها نمی شود؟ (مولف)

الف) اضافه کردن جدول به Schema

ج) اضافه کردن view

ب) حذف جدول از Schema

د) اضافه کردن ستون به جدول