

درس ۱ پایگاه داده پیشرفته

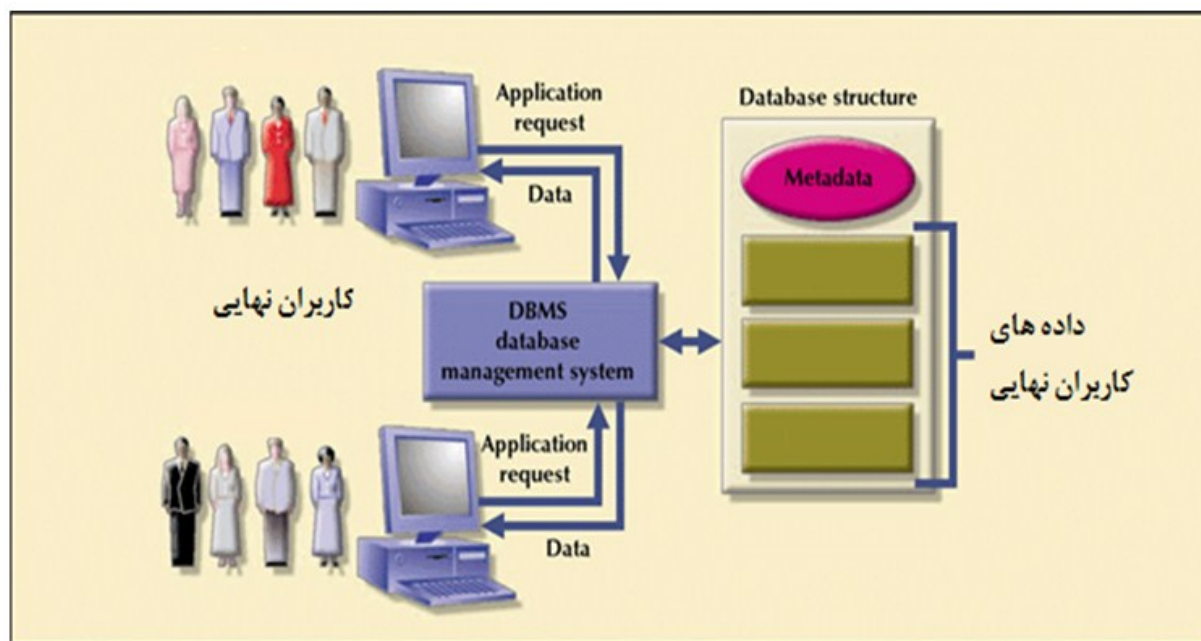
مقدمه و مرور بر پایگاه داده ها و مفاهیم اولیه

تعریف پایگاه داده ها (۱)

- پایگاه داده ها، مجموعه ای است از
- داده های ذخیره شده و پایا
- بصورت مجتمع
- مبتنی بر یک ساختار بطور صوری
- با حداقل افزونگی
- تحت مدیریت یک سیستم کنترل متمرکز بوده و
- مورد استفاده یک یا چند کاربر
- بطور اشتراکی و همزمان

تعریف پایگاه داده ها (۲)

- پایگاه داده، مجموعه ای از داده ها به صورت مجتمع و اشتراکی است که شامل
- داده های کاربران نهایی (واقعیات خام) و
- فراداده ها (داده هایی در مورد داده ها) است.
- سیستم مدیریت بانک اطلاعاتی **DBMS**
- شامل مجموعه ای از برنامه هاست که ساختار پایگاه داده ها را مدیریت کرده و دسترسی به
- داده های ذخیره شده در پایگاه داده ها را کنترل می کند.
- تامین امنیت، جامعیت، همزمانی



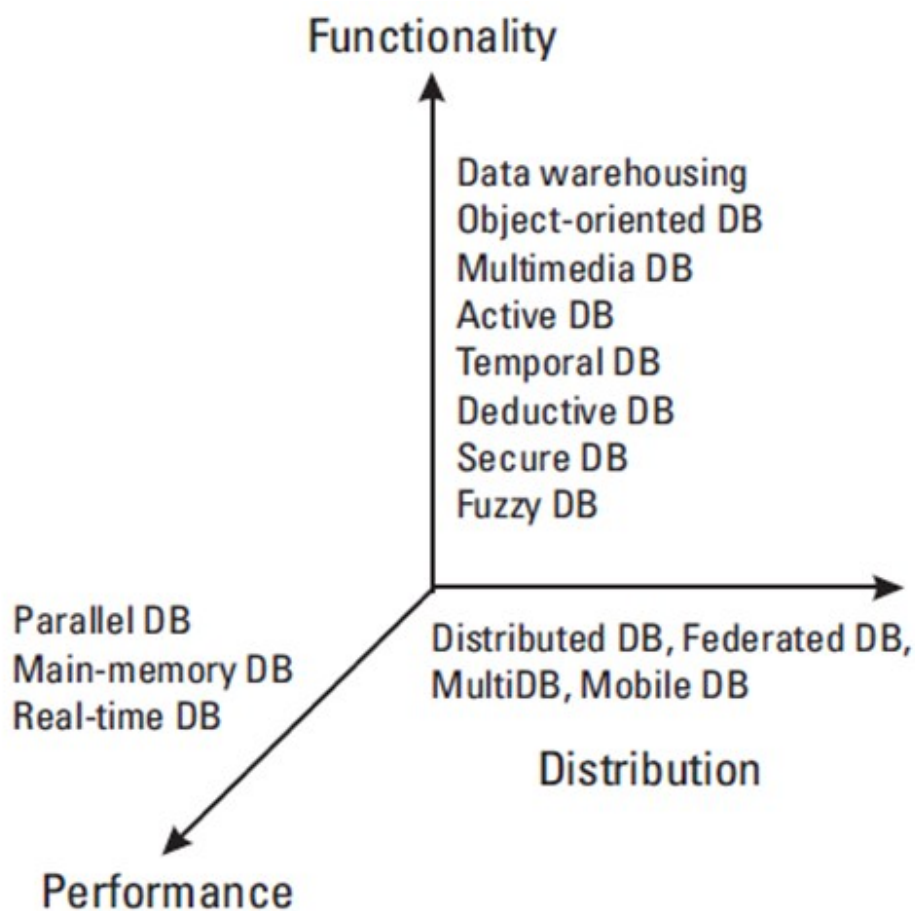
دلایل پایگاه داده ها

- امکان مدلسازی داده های عملیاتی بر اساس سمانتیک آنها
- وحدت ذخیره سازی کل داده های محیط عملیاتی
- اشتراکی شدن داده ها
- کاهش میزان افزونگی (نامطلوب)
- نامطلوب (ضعف طراحی) نرمال سازی
- افزونگی مطلوب آگاهانه
- تأمین جامعیت داده ها (عام و خاص)
- امکان اعمال ضوابط دقیق ایمنی
- امکان ترمیم داده ها
- تأمین استقلال داده ها
- عدم ناسازگاری در داده ها
- دستیابی همزمان
- سیر تحولات در پایگاه داده ها

1960

First DB products (DBOM, IMS, IDS, Total, IDMS)

- 1970
 - Codasyl standards
 - Relational model
 - RDBMS prototypes
 - Relational theoretical works
 - Three-level architecture (ANSI and Codasyl)
 - E/R model
 - First relational market products
- 1980
 - Distributed DBs
 - CASE tools
 - spreading the applications of conceptual modeling and relaunching DB design methodologies.
 - SQL standard (ANSI, ISO)
 - Object-oriented DB manifesto
- 1990
 - Third-generation DB manifesto
 - Client/server architecture (two-tier architecture)
 - First object DB products
 - Reference models (ISO/ANSI)
 - SQL 92
 - ODMG consortium (OO standards)
 - Data warehouses
 - SQL: 1999 (previously SQL3)
- 2000 Three-tier architecture
 - Object relational model
 - Databases and the World Wide Web
 - Mobile DBs
 - SQL/MM

Research and Market Trends 

برخی اصطلاحات

امنیت

محافظت داده ها در برابر خطرات مختلف و جلوگیری از دستیابی غیر مجاز

کاربران غیر مجاز و خطاهای عمدی

جامعیت

صحت داده ها و کنترل صحت، دقت و سازگاری داده ها و پیروی از مقررات سیستم.

کاربران مجاز و خطاهای سهوی

برخی اصطلاحات

ترمیم

- اطمینان از عدم خرابی داده های ماندگار(روی دیسک) بر اثر اشکالات احتمالی ایجاد شده در حین اجرای برنامه و بازسازی بانک اطلاعاتی که احتمالاً در اثر بروز اشکال در سیستم ناقص شده است.
- کنترل همروندی
- فعالیتی است در جهت هماهنگ کردن اعمال فرایند هایی که به طور موازی اجرا شده اند و به داده های مشترک دسترسی دارند و احتمالاً تداخل بالقوه دارند.

انواع سیستم های مدیریت پایگاه داده ها

- از نظر تعداد کاربران
- تک کاربره desktop
- چند کاربره

Work Group ■

Enterprise ■

- از نظر مکان ذخیره سازی داده ها
- Centralized
- Distributed
- کاربرد
- OLTP
- OLAP

انواع سیستم های مدیریت پایگاه داده ها چند کاربره

- Workgroup:
- در صورتی که یک پایگاه داده چند کاربری تعداد کمی از کاربران را پشتیبانی کند (بطور مثال کم تر از ۵۰)
- Enterprise:
- اگر پایگاه داده برای یک سازمان بزرگ استفاده شود و تعداد کاربران زیادی (بیش تر از ۵۰ و معمولاً ۱۰۰) داشته باشد

تقسیم بندی از نظر مکان ذخیره سازی داده ها

- Centralized
- داده ها فقط در یک سایت قرار دارند.
- Distributed

داده ها در چندین سایت توزیع شده ولی کاربر احساس توزیع شدگی داده ها را نخواهد داشت.

تقسیم بندی از نظر نوع کاربرد

Transactional (or production):

On-Line Transactional Processing (OLTP)

هدف، انجام درخواستها و پاسخگویی به تراکنشهای کاربران است که در آن پردازش تراکنش

ها مشتمل بر انجام عملیات روزانه مانند خرید و فروش و عملیات بانکی و مانند آن می باشد.

دادههای مورد استفاده در این تراکنشها دادههای بهروز، جاری و با جزئیات است.

Data warehouse:

On-Line Analytical Processing(OLAP) پایگاه داده تحلیلی برخط

پایگاه داده تحلیلی بستر مناسبی فراهم می آورد که داده ها به منظور پاسخگویی به پرسش های

تحلیلی به صورت بایگانی شده، سر جمع شده و سازمان یافته، ذخیره شوند.

سرویس دهنده های پردازش تحلیلی

ROLAP (پردازش تحلیلی برخط رابطه ای)

Relational OLAP

MOLAP (پردازش تحلیلی برخط چندبعدی)

Multi-dimensional OLAP

HOLAP (پردازش تحلیلی برخط ترکیبی)

Hybrid OLAP

یک سرویس دهنده ROLAP، از نوع توسعه یافته ای از سیستم های مدیریت پایگاه های داده رابطه ای

استفاده می کند.

پردازش تحلیلی بر خط رابطه ای براساس نوع ارتباط جدول واقعیت با جداول بعد به اشکال مختلفی

مدل می شوند.

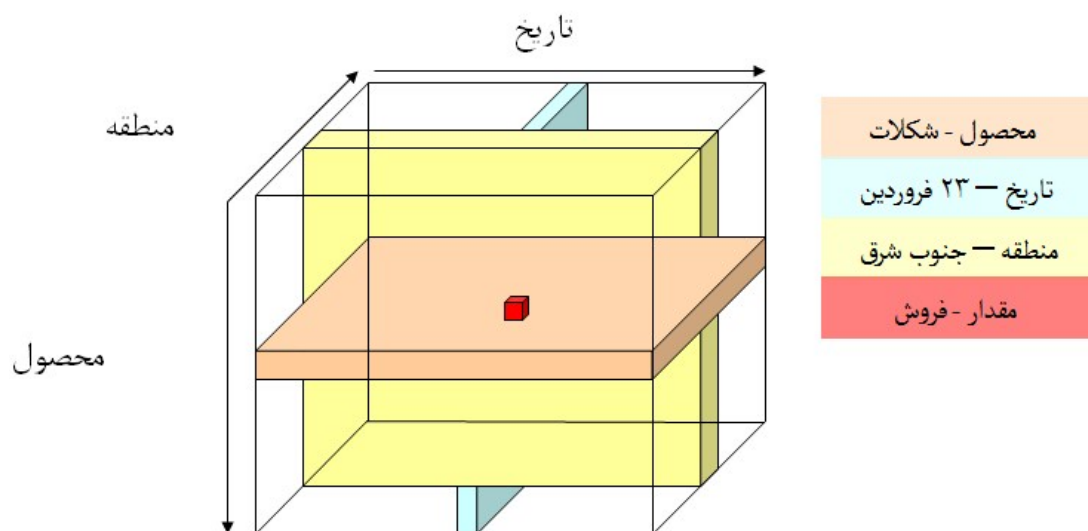
یک پایگاه داده تحلیلی چندبعدی داده را به شکل یک مکعب داده می بیند.

یک مکعب داده مانند فروش اجازه می دهد که داده ها در ابعاد مختلف مدل شوند و از دیدگاه های

مختلف مورد بررسی قرار گیرند.

- HOLAP نیز ROLAP و MOLAP را بایکدیگر ترکیب می کند.
- به عنوان مثال از ROLAP برای داده‌های مربوط به سابقه و تاریخچه استفاده می شود، در حالی که، داده‌هایی که به تناوب مورد دسترسی هستند، در یک MOLAP جداگانه نگهداری می شوند.

مفهوم مکعب داده



چه مقدار شکلات در منطقه جنوب شرق در تاریخ ۲۳ فروردین به فروش رسیده است؟

درس ۲

پایگاه داده پیشرفته

مفهوم تراکنش (بخش اول)

مفهوم تراکنش در پایگاه داده ها

□ تعریف تراکنش

□ تراکنش، واحد برنامه نویسی است که شامل یکسری عملیات مرتبط برای دسترسی و تغییر اطلاعات یک بانک اطلاعاتی است که در **جهان واقعی** در حکم یک عمل واحد تلقی می شوند.

□ تراکنش عبارتست از واحد سازگار و قابل اطمینان یک پردازش مشخص در پایگاه داده (OZSU)

□ تراکنش واحد اجرای برنامه ای است که به داده‌ها ی ذخیره شده دستیابی دارد.

□ مثال: انتقال مقدار ۵۰ دلار از حساب A به حساب B

1. **read(A)**
2. $A := A - 50$
3. **write(A)**
4. **read(B)**
5. $B := B + 50$
6. **write(B)**

□ هر تراکنش معادل یک رخداد در دنیای واقعی است.

□ تراکنش همواره به DBMS تسلیم می شود و DBMS در اعمال هرگونه کنترل و حتی به تعویق انداختن و ساقط کردن آن آزادی عمل دارد.

□ واحد کار DBMS تراکنش است.

□ هر تراکنش شامل مجموعه ای از عملیات است که با دستور شروع تراکنش (begin transaction) آغاز و با یک عمل **commit** و یا **undo** پایان می پذیرد.

□ تراکنش ممکن است بارها اجرا شود ولی هر بار اجرا یک تراکنش محسوب می شود.

نکات مهم در باره تراکنش

□ طراحی صحیح correctness

□ برنامه نویس باید عملیات اجرایی یک تراکنش را بصورت واحد و یکپارچه طراحی کند و این به خود DBMS ربطی ندارد.

□ خواندن اطلاعات

□ هر مورد اطلاعاتی مورد نیاز یک تراکنش باید فقط یک بار خوانده شود.

□ نوشتن اطلاعات

□ هر مورد اطلاعاتی مورد عمل در تراکنش در صورت تغییر فقط یکبار نوشته شود.

ویژگیهای تراکنش

- جیم گیری در سال ۱۹۸۱ ثابت کرد چهار کنترل ACID لازم است روی تمام تراکنش ها انجام شود تا صحت و جامعیت آن تضمین شود.
- ۱-ویژگی (تجزیه ناپذیری) اتمی بودن Atomicity
- تراکنش یا به تمامی اجرا می شود یا اصلا اجرا نمی شود.
- ۲-سازگاری(همخوانی) consistency
- تراکنش ها سازگاری و جامعیت پایگاه داده را حفظ می کنند.
- تراکنش برنامه ای است که هر بار اجرای کامل آن پایگاه داده را از یک حالت سازگاریه حالت سازگار دیگر در می آورد.(در صورت رسیدن به حالت اجرای کامل)
- ۳-ویژگی جداسازی(انزوا) Isolation
- تراکنش ها از یکدیگر مجزا هستند یعنی اثر مخرب روی یکدیگر ندارند.
- تراکنش ها مستقل از یکدیگر اجرا می شوند.
- این خاصیت ایجاب می کند هر تراکنش در تمام لحظات پایگاه داده را در حالت سازگار ببیند.
- ۴-ویژگی های پایداری Durability
- پس از آنکه تراکنش انجام(Commit) شد اثر آن را در بانک باقی می ماند حتی اگر سیستم اندکی بعد از کار بیفتد.

مثال تراکنش و خصوصیات تراکنش

- خاصیت سازگاری
 - مجموع مقادیر A و B پس از اجرای تراکنش تغییر نمی کند
1. **read(A)**
 2. $A := A - 50$
 3. **write(A)** اتمی بودن
 4. **read(B)** اگر تراکنش پس از مرحله ۳ و قبل از مرحله ۶ متوقف گردد، سیستم
 5. $B := B + 50$ تضمین می کند که تغییرات در بانک ثبت نگردند.
 6. **write(B)**

پایداری

پس از آنکه اجرای تراکنش مورد تایید قرار گرفت و تراکنش کامل گردید ، این تغییرات در بانک پایدار خواهند بود.

مفهوم Isolation

<input type="checkbox"/> T ₁	T ₂
1. read (A)	
2. $A := A - 50$	
3. write (A)	
	read(A), read(B), print(A+B)
4. read (B)	
5. $B := B + 50$	
6. write (B)	

جداسازی

اگر بین مراحل ۳ و ۶ یک تراکنش دیگر اجازه دستیابی به تغییرات در بانک را داشته باشد باعث ناسازگاری در بانک خواهد گردید.

جداسازی تراکنش ها می تواند با اجرای ترتیبی آنها تضمین شود ولی ما نیاز به اجرای همزمان تراکنش ها داریم

حالت های اجرای تراکنش

Active (فعال)

وقتی تراکنش شروع به اجرا می کند یا در حال اجراست

Partially committed

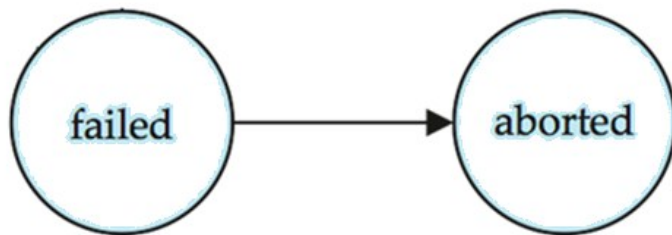
پس از اجرای آخرین دستور تراکنش به این حالت می رود.

Failed

پس از اینکه سیستم تشخیص داد تراکنش به هر دلیلی امکان ادامه اجرای عادی ندارد.

حالت های اجرای تراکنش (حالات پایانی)

Aborted (سقوط)



- ❑ در این حالت در حین اجرای تراکنش اشکالی پیش آمده است که منجر به توقف اجرای آن شده است.
- ❑ در صورت بروز اشکال در اجرای یک تراکنش، برای حفظ یکپارچگی اطلاعات، اثرات احتمالی بخشی از تراکنش که اجرا شده روی بانک اطلاعاتی باید خنثی شود. به این حالت برگشت Rollback گفته می شود.
- ❑ دوگزینه پس از ساقط شدن یک تراکنش مطرح است

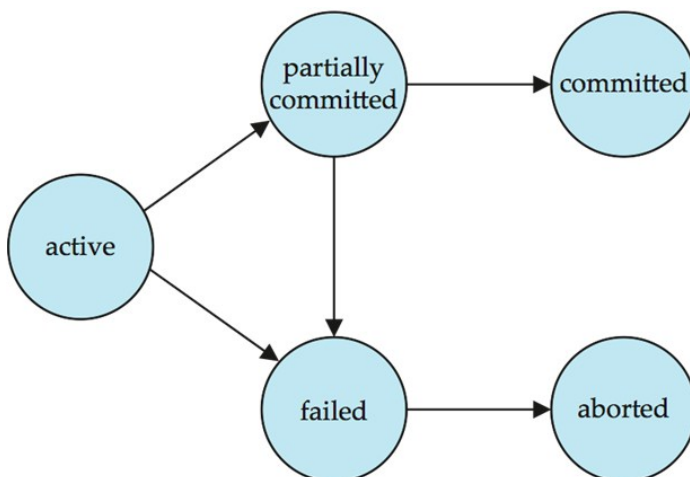
❑ restart the transaction

- can be done only if no internal logical error

❑ kill the transaction

❑ Committed(انجام)

- ❑ حالتی است که عملیات تراکنش بطور کامل موفقیت آمیز انجام شده و اثر آن نیز ثبت شده است پس از انجام تراکنش خنثی کردن تغییرات احتمالی تراکنش روی بانک غیر ممکن است.



دسته بندی مدل های تراکنش ها

- ❑ از نظر مدت زمان اجرا

❑ Short life Transaction (یا برخط)

- تراکنش زمان اجرا و پاسخ دهنی بسیار کوتاه دارد و معمولاً به بخش کوچکی از پایگاه داده دسترسی دارد.

- تراکنش عملیات بانکی، رزرو جا در خطوط هوایی

❑ Long life Transaction های طولانی

- تراکنش زمان اجرا و پاسخ دهنی طولانی دارد و معمولاً به بخش بزرگی از پایگاه داده دسترسی دارد.

■ تراکنش های برنامه های CAD/CAM

■ پرس و جو های پیچیده

■ برنامه های پردازش صوت و تصویر

□ از نظر ترتیب خواندن /نوشتن

□ تراکنش عمومی

■ دستورهای خواندن/نوشتن از ترتیب خاصی پیروی نمی کنند.

□ تراکنش های دو مرحله ای two step

■ تمام عمل های خواندن قبل از عمل های نوشتن اجرا می شوند.

□ تراکنش های محدود شده Restricted

■ اگر هر داده ای باید قبل از به روز رسانی خوانده شده باشد تراکنش را محدود شده گویند

□ تراکنش نوشتن کور Blind Write

■ تراکنشی که بی آنکه داده ای را بخواند آنرا بنویسد .

□ از نظر تعداد سطوح

□ Flat Transaction

■ تراکنش مسطح، تراکنشی است که یک نقطه آغاز و یک نقطه پایانی دارد و می

تواند هر تعداد عمل ساده باشد.

□ Nested Transaction

■ تراکنش های تودر تو نوعی از تراکنش که از تعدادی زیر تراکنش به هم مرتبط

تشکیل شده است.

■ به عبارتی تراکنش اصلی درخت یا سلسله مراتبی از تراکنش ها را تشکیل می

دهد.

■ برای عملیات کاربردهای پیچیده و بروز رسانی انبوه

Oracle and Transactions

□ Oracle is transaction oriented; that is, Oracle uses transactions to ensure data integrity.

□ A transaction is a series of SQL data manipulation statements that does a logical unit of work.

- For example, two UPDATE statements might credit one bank account and debit another.
- If your program fails in the middle of a transaction, Oracle detects the error and rolls back the transaction
- Thus, the database is restored to its former state automatically.
- Oracle and Transactions
- You use the COMMIT, ROLLBACK, SAVEPOINT, and SET TRANSACTION commands to control transactions.
- COMMIT makes permanent any database changes made during the current transaction.
- ROLLBACK ends the current transaction and undoes any changes made since the transaction began.
- SAVEPOINT marks the current point in the processing of a transaction.
- SET TRANSACTION
- sets transaction properties such as read/write access and isolation level.

اجرای همروند تراکنش ها

- محیط سیستم پایگاه داده یک محیط چند کاربری است و ممکن است تعدادی تراکنش بخواهند به داده مشترک دسترسی پیدا کنند.
- در یک سیستم پایگاه داده امکان اجرای همزمان/همروند تراکنش ها وجود دارد.
- همروندی یعنی اجرای غیر متوالی دو یا چند تراکنش.
- سیستم باید به دو یا چند تراکنش امکان دهد تا به داده مشترک به طور همزمان دسترسی داشته باشند.
- بدیهی است که همروندی تراکنش ها بایستی کنترل شود و عملهای تراکنش هایی که به طور غیر متوالی اجرا می شوند و به داده مشترک دسترسی دارند هماهنگ شوند.
- مزایای همروندی دو یا چند تراکنش
- افزایش بهره وری CPU
- هر تراکنش شامل چندین مرحله است برخی I/O را درگیر می کنند برخی CPU ، امکان موازی سازی دستورات تراکنش ها و موازی سازی تراکنش ها کارایی را بالا می برد.
- افزایش گذر دهی throughput
- تعداد تراکنش های اجرا و تمام شده در واحد زمان بیشتر شده

کاهش میانگین زمان پاسخ دهی

تراکنش های کوتاه نیازی به منتظر ماندن تا پایان تراکنش های بلند نیستند.

مثال، اجرای همزمان تراکنش ها

در اجرای همزمان، دو تراکنش می توانند به چندین حالت اجرا شوند.

مثال:

T_1	T_2
read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit

اجرای همزمان تراکنش ها (حالت ۱)

نه داده ای گم می شود و سازگاری برقرار است.

T_1	T_2	T_1	T_2
read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit	read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit

اجرای همزمان تراکنش ها (حالت ۲)

ممکن است داده ای گم شود و یکپارچگی نقض شود

T_1	T_2	T_1	T_2
read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit	read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit

کنترل همروندی تعبیه می شود تا مانع از بروز مشکل شود

کنترل همروندی

عدم کنترل همروندی باعث ایجاد سه پیامد منفی است:

Lost Update Problem

این مشکل موقعی بروز می کند که تراکنشی بلافاصله بعد از تراکنش دیگری که مقداری را برای داده ای نوشته است بخواند مقدار جدیدی برای آن داده بنویسد.

مشکل خواندن داده تثبیت نشده The Uncommitted dependency Problem

تراکنشی بخواند مقدار موقتا تغییر یافته توسط تراکنش دیگر را بخواند

The Inconsistency Analysis Problem تحلیل ناسازگار

مشکل بروزسانی گم شده Lost Update Problem

فرض کنیم تراکنش A مقدار داده t را در لحظه t1 می خواند

تراکنش B همان مقدار را در لحظه t2 می خواند.

تراکنش A مقدار داده t را در لحظه t3 تغییر می دهد.

Transaction A	time	Transaction B
Read (t)	t1	
	t2	Read (t)
Update t	t3	
	t4	Update t

- تراکنش B همان مقدار t را که در لحظه t2 خوانده در لحظه t4 تغییر می دهد.
- در لحظه t4 مقدار تغییر داده شده t در لحظه t3 گم می شود.

The Uncommitted dependency Problem

Transaction A	x=50	Transaction B
	x=15	Update x
Read(x)		
		Rollback

Transaction A	Time	Transaction B
		Update x
Update x		
		Rollback

The Inconsistency Analysis Problem

- این مشکل وقتی بروز می کند که تراکنش T1 مقدار دوداده نظیر A, B را بروز رسانی می کند و تراکنش T2 یکی از این دوداده مثل A را پس از بروز رسانی می خواند و مقدار داده دیگر B را پیش از بروز رسانی .
- لذا نتیجه T2 نادرست است.

تحلیل ناسازگار The Inconsistency Analysis Problem

Transaction T ₁	A=40, B=50, C=30 SUM=120	Transaction T ₂
Read (A)	A=40	
Read (B)	B=50	
	C=30	Read (C)
	C=20	C=C-10
	C=20	Write(C)
	A=40	Read A
	A=50	A=A+10
	A=50	Write (A)
		Commit
Read (C)	C=20, SUM=110	

کنترل همروندی

□ بخشی از DBMS است که مانع از بروز مشکلات ناشی از همروندی می شود. به عبارت دیگر امکان تاثیر تراکنش ها را برهم از بین می برد.

□ کنترل همروندی با استفاده از زمانبندی (برنامه ریزی) انجام می شود.

□ برنامه ریزی Scheduling

■ هر DBMS برای اجرای کنترل همروندی یک زمانبند ایجاد می کند که به این زمانبند طور پویا ایجاد شده و تراکنش ها در آن وارد و اجرا می شوند.

زمانبند

□ زمانبند

□ خطی (ترتیبی) Serial

■ تراکنش ها پشت سرهم

اجرا می شوند.

T_1	T_2
read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit

□ همروند Concurrent

■ تراکنش ها تماما با هم

اجرا می شوند.

□ زمانبند همروند

□ ترتیب پذیر

Serializable

□ ترتیب ناپذیر non

serializable

همروندی موقعی معتبر است که

ترتیب پذیر باشند.

A serial schedule in which T_1 is followed by T_2 :

زمانبند های معادل Equivalent Schedule

■ زمانبند A معادل B است اگر روی یک پایگاه داده واحد جواب یکسانی را تولید کنند.

T_1	T_2	T_1	T_2
read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit	read (A) $A := A - 50$ write (A) read (B) $B := B + 50$ write (B) commit	read (A) $temp := A * 0.1$ $A := A - temp$ write (A) read (B) $B := B + temp$ write (B) commit

زمانبند ترتیب پذیر serializable

- زمانبند ای که خروجی آن معادل اجرای یک زمانبند پی در پی (ترتیبی یا خطی) باشد.
- نتیجه حاصل از اجرای همروند چند تراکنش باید دقیقا شبیه اجرای تک تک آنها به طور مجزا باشد.
- در اجرای همروند تراکنش ها مکانیزمی شبیه زمانبند ترتیبی خواهیم داشت.
- برای اجرای n تراکنش n! زمانبند خطی امکان پذیر است.
- زمانبند های معادل ترتیب پذیر

conflict serializable

view serializable

ترتیب پذیری

ترتیب پذیری بیان صحت کنترل همروندی در سیستم های پایگاه داده هاست.

مزایای ترتیب پذیری:

- سیستم پایگاه داده ای که که اجراهایش ترتیب پذیر باشد فهمش ساده تر است. از دید کاربر چنین سیستم همانند یک پردازشگر پی در پی تراکنش ها به نظر میرسد.
- یک سیستم پایگاه داده که اجراهای پی در پی پذیر را تولید می کند از تداخلهای تغییرات گم شده و تحلیل های ناسازگار جلوگیری می کند.

جمع آوری : سراج جمالی

براساس جزوه های دکتر معصومی