नोट - यह दस्तावेज़ केवल न्यू सिलेबस के छात्रों के लिए है

न्यू सिलेबस के छात्रों को प्रश्न बैंक के साथ इस Topic का अध्ययन करना होगा

New added topics according to new syllabus (for sem 2 Msc)

Unit 4

Transaction Processing

Transaction, operations का समूह होता है जो कि केवल एक लॉजिकल ऑपरेशन की तरह treat होता है।

उदाहरण के लिये:-जब हम अपने बैंक अकाउंट से पैसे निकालते है या बैंक अकाउंट में पैसे जमा करवाते है।

ACID Properties:-Transaction की चार properties होती है जिसे हम ACID Properties कहते है।

1:-A-Atomicity
2:-C-Consistency
3:-I-Isolation
4:-D-Durability

1:-Atomicity:-जब Transaction एक ही स्टेप में पूर्ण हो जाता है तब transaction atomic होता है।

2:-Consistency:-जब transaction होता है तो डेटाबेस एक state से दूसरे स्टेट में consistence होता है।

3:-Isolation:-जब दो या उससे अधिक transaction एक साथ execute होते है तो एक transaction दूसरे transaction को प्रभावित नही करता है।

4:-Durability:-जब एक transaction पूरी तरह से complete हो जाता है तो जो परिवर्तन होते है वह permanently सिस्टम में रहते है।

अर्थात transaction durable होना चाहिए।

Types of Transaction:-

Transaction दो प्रकार के होते है:-

1:-Implicit Transaction
2:-Explicit Transaction

1:-Implicit Transaction:- Implicit Transaction में, SQL डेटाबेस इंजन एक नए transaction को शुरू कर देता है जब वर्तमान Transaction committed या rollback होता है।
Implicit transaction को on या off करने के लिए इस स्टेटमेंट का प्रयोग करते है:-
【SET IMPLICIT_TRANSACTIONS ON/OFF 】

प्रत्येक transaction को end करने के लिए COMMIT TRANSACTION या ROLLBACK TRANSACTION स्टेटमेंट का प्रयोग किया जाता है।

2:-Explicit Transaction:-एक explicit transaction में दोनों starting तथा ending transaction को स्पष्ट रूप से define किया जाता है।
Explicit transaction को शुरू करने के लिए BEGIN TRANSACTION का प्रयोग करते है तथा transaction को कम्पलीट करने के लिए COMMIT TRANSACTION या ROLLBACK TRANSACTION का प्रयोग करते है।

"Implicit तथा Explicit में main difference यह है कि Implicit transaction ऑटोमैटिक होता है तथा Explicit transaction यूजर डिफाइंड होता है।"

Locking technique

lock based protocol in dbms

इस प्रकार के प्रोटोकॉल में, कोई भी लेन-देन तब तक डेटा को पढ़ या लिख नहीं सकता है जब तक कि वह उस पर एक उपयुक्त lock प्राप्त नहीं करता है। lock दो प्रकार के होते हैं:

- **Shared lock:**

इसे read only lock के रूप में भी जाना जाता है। एक shared lock में, डेटा आइटम केवल transaction द्वारा पढ़ा जा सकता है।
इसे transaction के बीच share किया जा सकता है क्योंकि जब transaction lock होता है, तो यह डेटा आइटम पर डेटा को अपडेट नहीं कर सकता है।

- **Exclusive lock:**

Exclusive lock में, डेटा आइटम दोनों को पढ़ा जा सकता है और साथ ही लेनदेन द्वारा लिखा (write) जा सकता है।
यह lock exclusive है, और इस lock में, multiple transactions एक साथ एक ही डेटा को modify नहीं करते हैं।

There are four types of lock protocols available:

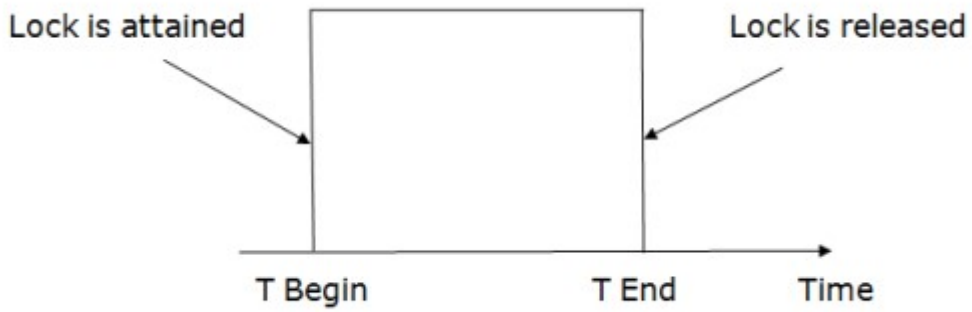
1. simplistic lock protocol

यह लेन-देन करते समय डेटा को lock करने का सबसे सरल तरीका है। simplistic lock-आधारित प्रोटोकॉल सभी लेनदेन को delete या insert या अपडेट करने से पहले डेटा पर lock प्राप्त करने की अनुमति देते हैं। यह लेनदेन को पूरा करने के बाद डेटा आइटम को unlock करेगा।

2. Pre-claiming lock protocol

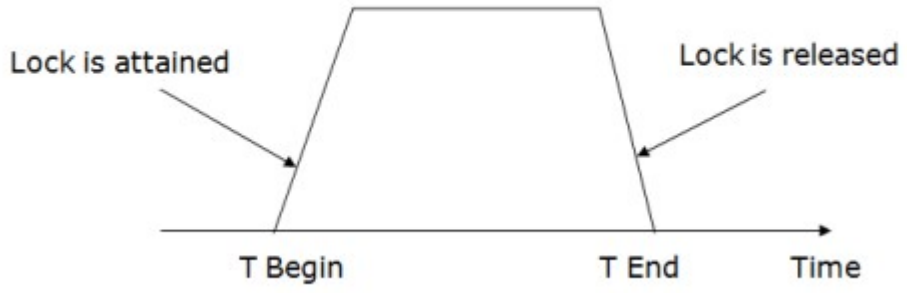
- Pre-claiming lock protocol उन सभी डेटा वस्तुओं को सूचीबद्ध करने के लिए लेनदेन का मूल्यांकन करता है जिन पर उन्हें lock की आवश्यकता होती है।
- लेनदेन के निष्पादन की शुरुआत करने से पहले, यह उन सभी data items पर सभी lock के लिए DBMS को अनुरोध करता है।
- यदि सभी locks दिए गए हैं, तो यह प्रोटोकॉल लेनदेन को शुरू करने की अनुमति देता है। जब लेनदेन पूरा हो जाता है तो यह सभी locks को रिलीज़ करता है।

- यदि सभी locks प्रदान नहीं किए जाते हैं, तो यह प्रोटोकॉल लेनदेन को rollback करने की अनुमति देता है और तब तक इंतजार करता है जब तक कि सभी locks प्रदान नहीं किए जाते हैं।



3. Two phase locking (2PL)

- Two phase locking protocol लेनदेन के निष्पादन चरण को तीन भागों में विभाजित करता है।
- पहले भाग में, जब लेन-देन का निष्पादन शुरू होता है, तो उसे उस lock की अनुमति लेनी होती है, जिसकी उसे आवश्यकता होती है।
- दूसरे भाग में, लेनदेन सभी locks को प्राप्त करता है। जैसे ही लेनदेन अपना पहला lock release करता है तीसरे चरण को शुरू किया जाता है।
- तीसरे चरण में, लेनदेन किसी नए lock की मांग नहीं कर सकता है। यह केवल अधिग्रहित lock को release करता है।



There are two phases of 2PL:

Glowing phase: growing phase में, लेनदेन द्वारा data item पर एक नया lock प्राप्त किया जा सकता है, लेकिन कोई भी release नहीं किया जा सकता है।

Shrinking phase : shrinking phase में, लेनदेन द्वारा रखे गए मौजूदा lock को release किया जा सकता है, लेकिन कोई नया lock हासिल नहीं किया जा सकता है। नीचे दिए गए उदाहरण में, यदि lock conversion की अनुमति है तो निम्न चरण हो सकता है:

1. Growing phase में lock (S(a) से X(a)) के upgrading की अनुमति है।
2. Shrinking phase में (X (a) से S (a)) lock का downgrade होना चाहिए।

उदाहरण:

	T1	T2
0	LOCK-S(A)	
1		LOCK-S(A)
2	LOCK-X(B)	
3	——	——
4	UNLOCK(A)	
5		LOCK-X(C)
6	UNLOCK(B)	
7		UNLOCK(A)
8		UNLOCK(C)
9	——	——

निम्नलिखित तरीके से पता चलता है कि 2-PL के साथ कैसे unlocking और locking काम करते हैं।

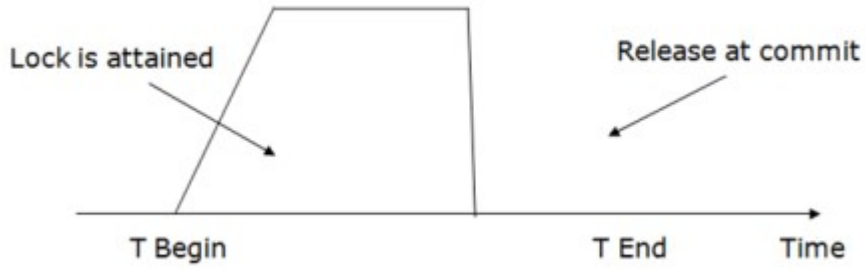
Transaction T1:

- **Growing phase:** चरण 1-3
- **Shrinking phase:** चरण 5-7
- **Lock point:** 3 पर

Transaction T2:

- **Growing phase:** चरण 2-6
- **Shrinking phase:** चरण 8-9
- **Lock point:** 6 पर

4. Strict Two-phase locking (Strict-2PL) in hindi
   - Strict -2PL का पहला चरण 2PL के समान है। पहले चरण में, सभी locks को प्राप्त करने के बाद, लेन-देन सामान्य रूप से निष्पादित होता है।
   - 2PL और strict 2PL के बीच एकमात्र अंतर यह है कि Strict-2PL उपयोग करने के बाद एक Lock release नहीं करता है।
   - संपूर्ण लेनदेन करने के लिए strict -2PL प्रतीक्षा करता है, और फिर यह एक समय में सभी locks release करता है।
   - Strict -2PL प्रोटोकॉल में lock release का shrinking phase नहीं है।



इसमें cascading abort नहीं है जैसा कि 2PL करता है।

Time based ordering Protocol

Timestamp

- Timestamp ordering protocol का उपयोग उनके टाइमस्टैम्प के आधार पर लेनदेन के लिए किया जाता है। लेन-देन का क्रम लेन-देन निर्माण के ascending order के अलावा और कुछ नहीं है।
- पुराने लेन-देन की प्राथमिकता अधिक है, इसलिए यह पहले निष्पादित होता है। लेनदेन के टाइमस्टैम्प को निर्धारित करने के लिए, यह प्रोटोकॉल system time या logical counter का उपयोग करता है।
- Lock based protocol का उपयोग निष्पादन समय पर लेनदेन के बीच परस्पर विरोधी जोड़े के बीच ऑर्डर को manage करने के लिए किया जाता है। लेकिन लेन-देन बनते ही timestamp based protocols काम करना शुरू कर देते हैं।
- मान लेते हैं कि दो लेनदेन T1 और T2 हैं। मान लीजिए कि लेनदेन T1 ने 007 बार सिस्टम में प्रवेश किया है और लेनदेन T2 ने 009 बार सिस्टम में प्रवेश

किया है। T1 की उच्च प्राथमिकता है, इसलिए यह पहले निष्पादित होता है क्योंकि यह सिस्टम में पहले दर्ज किया गया है।

- Timestamp ordering protocol एक डेटा पर अंतिम 'read' और 'write' ऑपरेशन के Timestamp को भी बनाए रखता है।

Basic timestamp ordering protocol work as follows:

1. जब भी कोई लेनदेन Ti **(Read) X** जारी करता है, तो निम्न स्थिति की जाँच करें :

- यदि W_TS (X)> TS (Ti) तो ऑपरेशन अस्वीकार कर दिया गया है।
- यदि W_TS (X) <= TS (Ti) तो ऑपरेशन निष्पादित होता है।
- सभी डेटा आइटम के timestamp अपडेट किए जाते हैं।

2. जब भी कोई लेनदेन Ti **write(X)** ऑपरेशन **जारी** करता है तो निम्न स्थिति की जांच करें :

- यदि TS (Ti) <R_TS (X) है तो ऑपरेशन अस्वीकार कर दिया जाता है।
- यदि TS (Ti) <W_TS (X) तो ऑपरेशन को अस्वीकार कर दिया जाता है और Ti को rollback किया जाता है अन्यथा ऑपरेशन निष्पादित हो जाता है।

Where,

TS(TI) लेन-देन Ti के टाइमस्टैम्प को दर्शाता है।
R_TS (X) डेटा-आइटम X के read timestamp को दर्शाता है।
W_TS (X) डेटा-आइटम X के write timestamp को दर्शाता है।

Advantages and Disadvantages of Timestamp Ordering protocol in hindi:

- TO protocol क्रमबद्धता (serializability) सुनिश्चित करता है:

Image: Precedence Graph for TS ordering

- Ts प्रोटोकॉल deadlock से Freedom सुनिश्चित करता है, जिसका अर्थ है कि कोई भी लेन-देन कभी भी इंतजार नहीं करता है।
- लेकिन schedule, recover नहीं हो सकता है और cascade-free भी नहीं हो सकता है।

Validation Technique

alidation based protocol को Optimistic Concurrency Control Technique भी कहते है. इस प्रोटोकॉल का प्रयोग DBMS (डेटाबेस मैनेजमेंट सिस्टम) में transactions में concurrency को avoid करने के लिए किया जाता है.
इसे optimistic कहते है क्योंकि इसमें बहुत ही कम interference होता है इसलिए transactions को execute करते समय इसको check करने की कोई आवश्यकता नहीं है।

इस तकनीक में, transaction को execute करने के दौरान कोई checking नहीं की जाती है.

validation based protocol तीन phase वाला प्रोटोकॉल है. अर्थात् इसमें transaction तीन phases में execute होता है.

1. Read phase
2. Validation Phase
3. Write Phase

1:- Read phase:- इस फेज में, transaction T को read तथा execute किया जाता है. इसका प्रयोग विभिन्न data items की values को read करने तथा उन्हें temporary local variables में स्टोर करने के लिए किया जाता है.
यह वास्तविक database में बदलाव किये बिना temporary variables में सभी write ऑपरेशन को perform कर सकता है.

2:- validation phase:-

इस फेज में, temporary variable value को वास्तविक data के विपरीत validate किया जाता है. validate यह check करने के लिए किया जाता है कि कहीं temporary variable वैल्यू serializability को violate तो नहीं कर रही है.

3:- Write Phase:-

यदि transaction का validation वेलिडेट हो जाता है, तो उसके बाद temporary values को database में write किया जाता है अन्यथा transaction वापस roll back हो जाता है.

यहाँ प्रत्येक phase के अलग-अलग timestamps होते हैं.

Start(Ti): यह उस time को contain किये रहता है जब Ti इसके execution को शुरू करता है.
Validation (T_i): यह उस time को contain किये रहता है जब Ti इसके read phase को पूरा कर लेता है और validation को शुरू करता है.
Finish(Ti): यह उस time को contain किये रहता है जब Ti इसके write phase को पूरा कर लेता है.
validation based protocol बहुत उपयोगी है और यदि conflicts होने की संभावना बहुत कम है तो यह concurrency की अधिकतम degree प्रदान करता है. ऐसा इसलिए होता है क्योंकि Serializability order पहले से निर्धारित नहीं होता है इसे validation phase में निर्धारित किया जाता है.

अतः यह ऐसे transaction को contain किये रहता है जिनके roll back होने की सम्भावना बहुत ही कम होती है.

Recovery concept in dbms

Log Based Recovery in DBMS in Hindi

- Log, records का एक sequence है। प्रत्येक लेन-देन का Log कुछ stable storage में रखा जाता है ताकि यदि कोई विफलता होती है, तो उसे वहां से पुनर्प्राप्त किया जा सके।
- यदि डेटाबेस पर कोई operation किया जाता है, तो उसे log में दर्ज किया जाएगा।
- लेकिन log को संग्रहीत करने की प्रक्रिया को डेटाबेस में वास्तविक लेन-देन लागू होने से पहले किया जाना चाहिए।

मान लेते हैं कि एक छात्र के शहर को modify करने के लिए एक लेन-देन है। इस लेन-देन के लिए निम्नलिखित log लिखे गए हैं।

- जब लेनदेन शुरू किया जाता है, तो यह 'start' log लिखता है।
  1. <Tn, start>
- जब लेन-देन शहर को 'Noida' से 'Bangalore' तक modify करता है, तो फ़ाइल में एक और log लिखा जाता है।
  1. <Tn, City, 'Noida', 'Bangalore' >
- जब लेन-देन समाप्त हो जाता है, तो यह लेन-देन के अंत को indicate करने के लिए एक और log लिखता है।
  1. <Tn, Commit>

डेटाबेस को modify करने के लिए दो दृष्टिकोण हैं:

1. Deferred database modification:

- आस्थगित संशोधन तकनीक तब होती है जब लेन-देन डेटाबेस को modify नहीं करता है जब तक कि यह commit न हो।
- इस विधि में, सभी log बनाए जाते हैं और stable storage में संग्रहीत किए जाते हैं, और लेन-देन के शुरू होने पर डेटाबेस को अपडेट किया जाता है।

2. Immediate database modification:

- तत्काल संशोधन तकनीक तब होती है जब डेटाबेस संशोधन तब हो जब लेन-देन अभी भी सक्रिय होता है।
- इस तकनीक में, डेटाबेस को हर operation के तुरंत बाद modify किया जाता है। यह एक actual database modification का अनुसरण करता है।

Recovery using log records

जब सिस्टम crash हो जाता है, तो सिस्टम लॉग को यह पता लगाने के लिए पूछता है कि कौन से लेन-देन को undone करने की आवश्यकता है और जिसे फिर से करने की आवश्यकता है।

1. यदि लॉग में रिकॉर्ड <Ti, start> और <Ti, commit> या <Ti, commit > है, तो लेन-देन Ti को फिर से करने की आवश्यकता है।

यदि लॉग में रिकॉर्ड <T n , start> होता है, लेकिन रिकॉर्ड <t >, commit > या <Ti, abort> नहीं है, तो लेन-देन Ti को undone करने की आवश्यकता है।

Database Security

डेटाबेस सुरक्षा कवर और डेटाबेस के सभी पहलुओं और घटकों पर सुरक्षा को लागू करती है। यह भी शामिल है:

- डेटाबेस में डेटा संग्रहीत
- डेटाबेस सर्वर
- डेटाबेस प्रबंधन प्रणाली (DBMS)
- अन्य डेटाबेस वर्कफ़्लो अनुप्रयोग

डेटाबेस सुरक्षा आमतौर पर डेटाबेस प्रशासक और या अन्य सूचना सुरक्षा पेशेवर द्वारा नियोजित, कार्यान्वित और अनुरक्षित की जाती है।

डेटाबेस सुरक्षा का विश्लेषण और कार्यान्वित करने के कुछ तरीकों में शामिल हैं:

- मजबूत और मल्टीफ़ॉर्मर एक्सेस और डेटा प्रबंधन नियंत्रण को लागू करके अनधिकृत पहुंच और उपयोग को प्रतिबंधित करना

- लोड / तनाव परीक्षण और डेटाबेस के क्षमता परीक्षण यह सुनिश्चित करने के लिए कि यह वितरित वितरण सेवा (DDoS) के हमले या उपयोगकर्ता अधिभार में दुर्घटनाग्रस्त न हो

- डेटाबेस सर्वर और चोरी और प्राकृतिक आपदाओं से बैकअप उपकरण की भौतिक सुरक्षा

- किसी भी ज्ञात या अज्ञात कमजोरियों के लिए मौजूदा प्रणाली की समीक्षा करना और उन्हें कम करने के लिए रोड मैप / योजना को परिभाषित करना और लागू करना

Access control Database security

डेटाबेस सिस्टम में एक्सेस कंट्रोल की आपूर्ति करने का सामान्य तरीका डेटाबेस के भीतर विशेषाधिकार प्रदान करने और बदलने के लिए निर्भर है। एक विशेषाधिकार उपयोगकर्ता को कुछ डेटाबेस ऑब्जेक्ट बनाने या एक्सेस करने या कुछ विशिष्ट डीबीएमएस उपयोगिताओं को चलाने की अनुमति देता है। विशेषाधिकार उन कार्यों के लिए आवश्यक कार्यों को प्राप्त करने के लिए उपयोगकर्ताओं को दिए जाते हैं।

डेटाबेस विभिन्न प्रकार के अभिगम नियंत्रण प्रदान करता है:

- विवेकाधीन अभिगम नियंत्रण (Discretionary Access Control) (DAC)
- अनिवार्य अभिगम नियंत्रण (Mandatory Access Control)(मैक)