

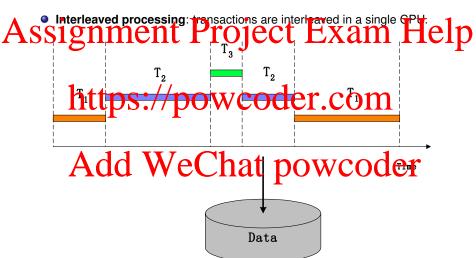
Assignment Project Exam Help

https://cpawcadaricam

Add WeChat powcoder



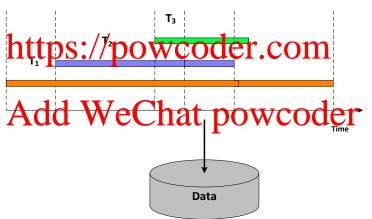
Concurrent Transactions





Concurrent Transactions

Assignment Project Exam Help





Concurrent Transactions

Assignment-Project-Exam-Help

- Increase throughput (average number of completed transactions)
- For example, while one transaction is waiting for an object to be transaction from distriction to be good from distriction to be done in parallel with CPU activity).
- --> Reduce latency (average time to complete a transaction)
- For example, internetive execution of a short transaction with a brig transaction usually allows the slowly real of the completed more quickly.
- But the DBMS has to guarantee that the interleaving of transactions does not lead to inconsistencies, i.e., concurrency control.



Why is Concurrency Control Needed?

Assignment Project Exam Help

Concurrency control is needed for preventing the following problems:

https://powcoder.com

The dirty read problem

Add De Chat powcoder

The phantom read problem



(1) - The Lost Update Problem

Assign Busic dut (E). To ject Exam Help

```
T<sub>1</sub>: SELECT balance FROM ACCOUNT WHERE name='Bob';
T<sub>2</sub>: SELECT balance FROM ACCOUNT WHERE name='Bob';
T<sub>1</sub>: UPDATE ACCOUNT SET balance=balance=100 WHERE name='Bob';
T<sub>2</sub>: OPDATE ACCOUNT ET balance=balance=500 WHERE name='Bob';
T<sub>2</sub>: COMMIT;
```

Steps -	T1 XX7	T_2
AC	Cead(BVV CC	lnat pov
2		read(B)
3	write(B) (B:=B-100)	
4	commit	
5		write(B) (B:=B+500)
6		commit

/\$te/s	(Bob)
before 1	\$200
after 2	\$200
after 4	\$100
after 6	\$700



(1) - The Lost Update Problem

Assign Help (E). For this account while Alice deposits the property of the saccount (E). For this account (E) acco

```
T1: SELECT balance FROM ACCOUNT WHERE name='Bob';
T2: SELECT balance FROM ACCOUNT WHERE name='Bob';
T1: UPDATE ACCOUNT SET balance=balance=100 WHERE name='Bob';
T2: UPDATE ACCOUNT ET balance=balance=500 WHERE name='Bob';
T2: UPDATE ACCOUNT ET balance=balance=500 WHERE name='Bob';
T2: COMMIT;
```

Steps	T1 XX 7	T_2		
¹ /A C	read(EVV e	hat nov	XXE (S)	(Bob)
2		read(B)	before 1	\$200
3	write(B) (B:=B-100)		after 2	\$200
4	commit		after 4	\$100
5		write(B) (B:=B+500)	after 6	\$700
6		commit		

• Answer: Bob's balance should be \$600. The update by T_1 is lost!



(1) - The Lost Update Problem

SSOCOTA VIET WE tankact engroda ethe same object, and one transaction position overwrite the value of the object which has already been updated by another transaction (write-write conflicts).

https://powcoder.com



• write(B) by T_2 overwrites B, and the update by T_1 is lost.



(2) - The Dirty Read Problem

Assign the Buschult (E). To ject Exam Help

```
T1: SELECT balance FROM ACCOUNT WHERE name='Bob';

T1: UPDATE ACCOUNT SET balance=balance=100 WHERE name='Bob';

T2: SELECT balance FROM ACCOUNT WHERE name='Bob';

T3: UPDATE ACCOUNT LET balance=balance=500 WHERE name='Bob';

T4: COMMIT;
```

Steps -	T1 XX7	T_2	5
AC	ead(B) (B:=B-100)	lnat po	W
3	WITE(D) (B:=B-100)	read(B)	а
4	abort	,	a
5		write(B) (B:=B+500)	a
6		commit	a

Steps	B(Bob)
before 4	\$200
after 1	\$200
after 2	\$100
after 4	\$200
after 6	\$600



(2) - The Dirty Read Problem

Assignituable action (E). To ject Exam Help

```
T1: SELECT balance FROM ACCOUNT WHERE name='Bob';
T1: UPDATE ACCOUNT SET balance=balance-100 WHERE name='Bob';
T2: SELECT balance FROM ACCOUNT WHERE name='Bob';
T1 AFORT SET BALANCE BALANCE
```

Steps	T ₁	<i>T</i> ₂	Steps	B(Bob)
$^{1}\Delta$	read(B)	that poy	7/eim	ASE 00
3	MICCITAL (b:=1-100)	read(B)	atter 1	\$200
4	abort	1044(D)	after 2	\$100
5		write(B) (B:=B+500)	after 4	\$200
6		commit	after 6	\$600

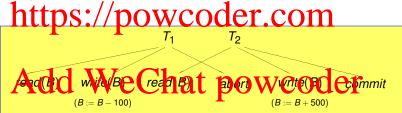
Answer: Bob's balance should be \$700 since T₁ was not completed.



(2) - The Dirty Read Problem

Socrary the population pould read the value of an object that his population but has not yet committee (write lead populates).

Example:



• T_1 fails and must change the value of B back to \$200; but T_2 has read the uncommitted (\cong dirty) value of B (\$100).



(3) - The Unrepeatable Read Problem

A S **Example:** Bob checks his a count (T_2) twice (takes time to decide we the S State with the S to with the S to with the S to with the S to S to with the S to S to with the S to S to

```
T1: SELECT balance FROM ACCOUNT WHERE name='Bob';
T2: SELECT balance FROM ACCOUNT WHERE name='Bob';
T3: WPDATE ACCOUNT SET balance=balance=b00 WHERE name='Bob';
T1: SELECT balance FROM ACCOUNT WHERE name='Bob';
```

Steps	T ₁	J_2	<u>~</u>	
$^{1}\!Ac$	ead(L)	read		hat
2				
3		write	(B) (B:=B-500)
4		com	nit	
5	read(B)			

Steps	B(Bob)		
	\$500	er	
after 2	\$500		
after 3	\$0		
after 4	\$0		
after 5	\$0		



(3) - The Unrepeatable Read Problem

Example: Bob checks his account (T₁) twice (takes time to decide whether State with that v 5100) Mile A ce with rates 1500 from Bob saccount [T₂]. C 1 T1: SELECT balance FROM ACCOUNT WHERE name='Bob'; T2: SELECT balance FROM ACCOUNT WHERE name='Bob'; T2: UPDATE ACCOUNT SET balance=balance-500 WHERE name='Bob';

her Size		der com
_	_	

Steps	<i>T</i> ₁	<i>T</i> ₂
1	read(B)	7-014
² A C	la v	Vree nat
3 -	,	write(B) (B:=B-500)
4		commit
5	read(B)	

Steps	B(Bob)	
before 1	\$500	
):(ft):\/ 2 / (\$ 700	r
after 3	\$0	Г
after 4	\$0	
after 5	\$0	

 Answer: Bob received two different account balances \$500 and \$0, even though he hasn't withdrawn any money yet.



(3) - The Unrepeatable Read Problem

Assignment Project Fxam Help another transaction but is still in progress (could issue two read for the object, or a write after reading the object) (read-write conflicts).

• Exhtetps://powcoder.com

Add We Chat Powcoder read(B) read(B) write(B) commit read(B) ... (B = 500) (B = 500) (B = 0)



(4) - The Phantom Read Problem

Example: A query is subtilized for finding all customers whose account S balance are less than \$3.0 (I_1) with Alice is opening an exact unit with the balance \$200 (T_2).

Assume that only Bob (B) has an account whose balance is less than \$300 before Alice (A) opens his new account.

7, NEUCOSE, VROM DONWHEGO GET, COM

 T_2 : INSERT INTO ACCOUNT(id, name, balance) VALUES(99, 'Alice', 250);

 T_2 : COMMIT;

 T_1 : SELECT name FROM ACCOUNT WHERE balance<300;

Steps(V	e	
1	read(R)			
2			write	(R)
3			comi	mit
4	read(R)			

prog V	Anth Le sun
before 1	$R = \{B\}$
after 1	$R = \{B\}$
after 2	$R = \{A, B\}$
after 4	$R = \{A, B\}$



(4) - The Phantom Read Problem

Example: A query is substituted for finding all cutstomers whose act of nt. Statemers whose act of nt. The balance (T_1) the balance (T_2) .

Assume that only Bob (B) has an account whose balance is less than \$300 before Alice (A) opens his new account.

 T_1 Filic have FROM A) out WHIRE bellances 0; C T_2 : INSERT INTO According (id, name, balance) VALUES (99, 'Alice', 250);

 T_2 : COMMIT;

T₁: SELECT name FROM ACCOUNT WHERE balance<300;

Steps	σ V		nat	PIPPS XX	/ Burry resput
1	read(R)		Itt	before 1	$R = \{B\}$
2		write(R)		after 1	$R = \{B\}$
3		commit		after 2	$R = \{A, B\}$
_4	read(R)			after 4	$R = \{A, B\}$

 Answer: T₁ reads Account based on the condition balance<300 twice but gets two different results {B} and {A, B}.



(4) - The Phantom Read Problem

Assignment Project Fxam Help conditions of another transaction so that, by the same search condition, the

• Exhttps://powcoder.com

transaction obtains different results at different times.

Add WeChat powcoder read(R) write(R) commit read(R) (R = {B}) (insert A) (R = {A,B})