

A decorative graphic on the left side of the slide, consisting of a network of light blue lines and circles, resembling a circuit board or a neural network, set against a dark blue background.

# CONTROL DE CONCURRENCIA

PRIMERA PARTE

# ÍNDICE

- **Introducción**

- Lock binario, shared lock
- Two phase locking
- Control de concurrencia por timestamp
- Control de concurrencia por multiversion
- Niveles Aislamiento en SQL

# INTRODUCCIÓN

- Existen 3 problemas básicos de control de concurrencia
  - Modificación perdida ( lost update)
  - Falsa Modificación ( o dirty read)
  - Falsa Sumarizacion ( non repeatable read)

# MODIFICACIÓN PERDIDA

- Transacción 1

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-

# MODIFICACIÓN PERDIDA

- Transacción 1

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Valor X disco	Valor X – T1	Valor X – T2
0	1	-	-
1	1	1	-

# MODIFICACIÓN PERDIDA

- Transacción 1

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Oper.	Valor X ( disco )	Valor X - T1	Valor X - T2
0	1	-	-
1	1	1	-
2	1	3	-

# MODIFICACIÓN PERDIDA

- Transacción 1

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Oper.	Valor X ( disco )	Valor X - T1	Valor X - T2
0	1	-	-
1	1	1	-
2	1	3	-
3	1	3	1

# MODIFICACIÓN PERDIDA

- Transacción 1

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Oper.	Valor X ( disco )	Valor X - T1	Valor X - T2
0	1	-	-
1	1	1	-
2	1	3	-
3	1	3	1
4	3	3	1



# MODIFICACIÓN PERDIDA

- Transacción 1
- Transacción 2

1)  $r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

3)  $r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Opera.	Valor X ( disco )	Valor X - T1	Valor X - T2
0	1	-	-
1	1	1	-
2	1	3	-
3	1	3	1
4	3	3	1
5	3	3	6
6	6	3	6

Es correcto el resultado?

# MODIFICACIÓN PERDIDA

- Transacción 1
  - Transacción 2
- 1)  $r1(x)$
  - 2)  $X = x + 2$
  - 3)  $r2(x)$
  - 4)  $w1(x)$
  - 5)  $X = x + 5$
  - 6)  $w2(x)$

Opera.	Valor X ( disco )	Valor X - T1	Valor X - T2
0	1	-	-
1	1	1	-
2	1	3	-
3	1	3	1
4	3	3	1
5	3	3	6
6	6	3	6

Es correcto el resultado?

Originalmente X valía 1 y le sumé 2 y luego 5

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

8)  $a1$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

8)  $a1$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-
1	1	1	-

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

8)  $a1$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-
1	1	3	-
2	1	3	-
3	3	3	

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

8)  $a1$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

Operación	Valor X - disco	Valor X - T1	Valor X - T2
0	1	-	-
1	1	3	-
2	1	3	-
3	3	3	
4	3	3	3
5	3	3	8
8	8	3	8

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

8) **a1**

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-
1	1	3	-
2	1	3	-
3	3	3	
4	3	3	3
5	3	3	8
8	8	3	8

# FALSA MODIFICACIÓN

Transacción 1

1)  $r1(x)$

2)  $x = x + 2$

3)  $w1(x)$

• Transacción 2

4)  $r2(x)$

5)  $x = x + 5$

6)  $w2(x)$

7)  $c2$

8) **al**

Operación	Valor X - disco	Valor X – T1	Valor X – T2
0	1	-	-
1	1	3	-
2	1	3	-
3	3	3	
4	3	3	3
5	3	3	8
8	8	3	8

Es correcto el resultado ?



# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

## 7) $r^2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$
[illegible]

# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r_1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

## 7) $r^2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$
[illegible]

# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r_1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

7)  $r_2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$
[illegible]

# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r_1(y)$

$$3) y = y - 2$$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

7)  $r_2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$
[illegible]

# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

## 7) $r^2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$
[illegible]

# FALSA SUMARIZACIÓN

# 1) $r_1(x)$

## 2) $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

## 5) $w_1(x)$

## 6) $r_2(x)$

## 7) $r^2(y)$

8)  $z = x + y$

9)  $w_2(z)$

$$10)w1(y)$$

Op	Valor X - disco	Valor X – T1	Valor X – T2	Valor Y - disco	Valor Y – T1	Valor Y – T2	Valor Z – T2
0	10	-	-	20			
1	10	10	-	20	-	-	-
2	10	10	-	20	20	-	-
3	10	10	-	20	18	-	-
4	10	13	-	20	18	-	-
5	13	13	-	20	18	-	-

# FALSA SUMARIZACIÓN

1)  $r1(x)$

2)  $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

5)  $w1(x)$

6)  $r2(x)$

7)  $r2(y)$

8)  $z = x + y$

9)  $w2(z)$

10)  $w1(y)$

Op	Valor X - disco	Valor X – T1	Valor X – T2	Valor Y - disco	Valor Y – T1	Valor Y – T2	Valor Z – T2
0	10	-	-	20			
1	10	10	-	20	-	-	-
2	10	10	-	20	20	-	-
3	10	10	-	20	18	-	-
4	10	13	-	20	18	-	-
5	13	13	-	20	18	-	-
6	13	13	13	20	18	-	-
7	13	13	13	20	18	20	-

# FALSA SUMARIZACIÓN

1)  $r1(x)$

2)  $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

5)  $w1(x)$

6)  $r2(x)$

7)  $r2(y)$

8)  $z = x + y$

9)  $w2(z)$

10)  $w1(y)$

Op	Valor X - disco	Valor X – T1	Valor X – T2	Valor Y - disco	Valor Y – T1	Valor Y – T2	Valor Z – T2
0	10	-	-	20			
1	10	10	-	20	-	-	-
2	10	10	-	20	20	-	-
3	10	10	-	20	18	-	-
4	10	13	-	20	18	-	-
5	13	13	-	20	18	-	-
6	13	13	13	20	18	-	-
7	13	13	13	20	18	20	-
8	13	13	13	20	18	20	33



# FALSA SUMARIZACIÓN

1)  $r1(x)$

2)  $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

5)  $w1(x)$

6)  $r2(x)$

7)  $r2(y)$

8)  $z = x + y$

9)  $w2(z)$

10)  $w1(y)$

Op	Valor X - disco	Valor X – T1	Valor X – T2	Valor Y - disco	Valor Y – T1	Valor Y – T2	Valor Z
0	10	-	-	20	-	-	-
1	10	10	-	20	-	-	-
2	10	10	-	20	20	-	-
3	10	10	-	20	18	-	-
4	10	13	-	20	18	-	-
5	13	13	-	20	18	-	-
6	13	13	13	20	18	-	-
7	13	13	13	20	18	20	-
8	13	13	13	20	18	20	33
9	13	13	13	20	18	20	33
10	13	13	13	18	18	20	33

Cuánto valen X  
de T1 ? Y de  
Cuáles son los  
posibles de

# FALSA SUMARIZACIÓN

1)  $r1(x)$

2)  $r1(y)$

3)  $y = y - 2$

4)  $x = x + 3$

5)  $w1(x)$

6)  $r2(x)$

7)  $r2(y)$

8)  $z = x + y$

9)  $w2(z)$

10)  $w1(y)$

Op	Valor X - disco	Valor X - T1	Valor X - T2	Valor Y - disco	Valor Y - T1	Valor Y - T2	Valor Z
0	10	-	-	20	-	-	-
1	10	10	-	20	-	-	-
2	10	10	-	20	20	-	-
3	10	10	-	20	18	-	-
4	10	13	-	20	18	-	-
5	13	13	-	20	18	-	-
6	13	13	13	20	18	-	-
7	13	13	13	20	18	20	-
8	13	13	13	20	18	20	33
9	13	13	13	20	18	20	33
10	13	13	13	18	18	20	33

Cuánto valen X  
de T1 ? Y de  
T2 ?  
Cuáles son los  
posibles de

# INTRODUCCIÓN

- Hasta ahora estuvimos trabajando sobre schedules y que características tenían que tener esos schedules para que no se produjeran problemas de concurrencia
- Armamos los schedules conociendo TODAS las operaciones que tenía una transacción por adelantado..

# INTRODUCCIÓN

- Eso no es posible en una “base de datos real”
- Para resolver esto se proponen distintos “tipos de schedulers” , cuál es la característica saliente que tienen que tener?
- Cuando el **scheduler** recibe una operación puede tomar 3 decisiones
  - Rechazarla
  - Procesarla
  - Demorarla
- Existen dos tipos básicos de schedulers, los que funcionan en base a locks y los que no

# ÍNDICE

- Introducción
- **Lock binario, shared lock**
- Two phase locking
- Control de concurrencia por timestamp
- Control de concurrencia por multiversion
- Niveles Aislamiento en SQL

# LOCK BINARIO – SHARED LOCK

- Estos dos mecanismos utilizan el concepto de “lock”, es decir asocian una variable a cada “data ítem” para indicar si el mismo esta disponible o no
- El lockeo binario es el modelo más sencillo. Un data ítem simplemente puede estar lockeado o no lockeado
- La notación es
  - $l_i(x)$  :  $T_i$  lockea el data ítem  $x$
  - $u_i(x)$  :  $T_i$  libera el lock sobre el data ítem  $x$
- Cuando una transacción lee o escribe un ítem solicita un lock sobre el mismo
- Que problema tiene esta aproximación?

## LOCK BINARIO – SHARED LOCK ( CONT)

- ▶ Para resolver la limitación del lockeo binario el shared lock o lockeo ternario permite que el lock sea exclusivo o compartido ( shared)
- ▶ La notación es
  - ▶  $rl_i(x)$  :  $T_i$  lockea el data ítem  $x$  en formato compartido
  - ▶  $wl_i(x)$  :  $T_i$  lockea el data ítem  $x$  en formato exclusivo
- ▶ Existe la posibilidad de que una transacción efectúe un “upgrade” de un lock para pasar de un lock de lectura a uno de escritura
  - ▶ La notación es  $ul_i(x)$

## LOCK BINARIO – SHARED LOCK ( CONT)

► Son posibles las siguientes situaciones?

1.  $rl1(x), rl2(x)$

2.  $rl1(x), wl2(x)$

3.  $wl1(x), rl2(x)$

4.  $wl1(x), wl2(x)$

5.  $rl1(x), wl1(x)$



# RESUMIENDO TIPOS DE LOCKEO

Ti   Tj	Read Lock (x)	Write Lock (x)
Read lock (x)		
Write Lock (x)		

# RESUMIENDO TIPOS DE LOCKEO

Ti   Tj	Read Lock (x)	Write Lock (x)
Read lock (x)	OK	CONFLICTO
Write Lock (x)	CONFLICTO	CONFLICTO

# MODIFICACIÓN PERDIDA

- Transacción 1

1) Write lock(x)

$r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3) Write lock(x)

$r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Lock x
0	-

# MODIFICACIÓN PERDIDA

- Transacción 1

1) Write lock(x)

$r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3) Write lock(x)

$r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Lock x
0	-
1	T1 - write

# MODIFICACIÓN PERDIDA

- Transacción 1

1) Write lock(x)

$r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3) Write lock(x)

$r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Lock x
0	-
1	T1 - write
2	T1 - write

# MODIFICACIÓN PERDIDA

- Transacción 1

1) Write lock(x)

$r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3) **Write lock(x)**

$r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Lock x
0	-
1	T1 - write
2	T1 - write
3	

# MODIFICACIÓN PERDIDA

- Transacción 1

1) Write lock(x)

$r1(x)$

2)  $X = x + 2$

4)  $w1(x)$

- Transacción 2

3) Write lock(x)

$r2(x)$

5)  $X = x + 5$

6)  $w2(x)$

Operación	Lock x
0	-
1	T1 – write
2	T1 - write
3	No puede procesarse

Por qué no puede procesarse la operación 3?

Porque existe un conflicto en los locks T2 quiere lockear para escribir y T1 ya tiene un lock de escritura

# CONTROL DE CONCURRENCIA

- Esta presentación fue armada utilizando, además de material propio, material de
  - "Concurrency Control and Recovery in Database Systems" de Bernstein, Hadzilacos y Goodman



A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a neural network, extending from the top to the bottom.

# CONTROL DE CONCURRENCIA

MUCHAS GRACIAS!