



Apunte Certamen 3

Marcelo PAZ
Base de Datos

18 de diciembre de 2023



1. Importante

- **Dependencia Funcional:** Sea r una relación y $X, Y \subseteq R$ entonces $X \rightarrow Y$ es una dependencia funcional de r si y solo si para cada par de tuplas t_1 y t_2 de r si $t_1[X] = t_2[X]$ entonces $t_1[Y] = t_2[Y]$. **Video explicativo**

Ejercicio:

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_3	c_2	d_3
a_3	b_3	c_2	d_4

DF trivial $A \rightarrow A$

1. $A \rightarrow C$

A	C
a_1	c_1
a_1	c_1
a_2	c_2
a_2	c_2
a_3	c_2

2. $C \rightarrow A$

C	A
c_1	a_1
c_1	a_1
c_2	a_2
c_2	a_2
c_2	a_3

3. $AB \rightarrow D$

AB	D
a_1b_1	d_1
a_1b_2	d_2
a_2b_2	d_2
a_2b_3	d_3
a_3b_3	d_4

- **Leyes de Armstrong S+:** Sean $A, B, C, D \subseteq R$ y r una relación.

1. *Reflexibilidad:* Si $B \subset A$ / $A \rightarrow B$.
2. *Aumentatividad:* $A \rightarrow B$ / C $AC \rightarrow BC$.
3. *Descomposición:* $A \rightarrow BC$ $A \rightarrow B \wedge A \rightarrow C$.
4. *Composición:* $A \rightarrow B \wedge C \rightarrow D$ $AC \rightarrow BD$.
5. *Autodeterminación:* $A \rightarrow A$.
6. **Transitividad:* $A \rightarrow B \wedge B \rightarrow C$ $A \rightarrow C$.
7. *Pseudotransitividad:* $A \rightarrow B \wedge CB \rightarrow D$ $AC \rightarrow D$.



- **Super Clave:** Un atributo que determina al resto. Es aquella que es determinante para todos los atributos.
- **Clave Candidata:** Sea r una relación y $X \subseteq R$ es una clave candidata de r si y solo si $X \rightarrow R$ y X es minimal. **Video algoritmo alternativo**
- **Forma Normal:** Es una propiedad que satisface un cierto número de restricciones para tener una buena optimización. **Video explicativo**

1. **1FN:** Todos los atributos son atómicos y posee alguna clave candidata.

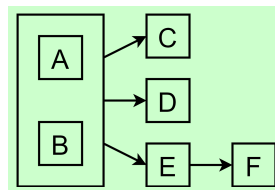
Nombre
Marcelo Paz

Nombre	Apellido
Marcelo	Paz

2. **2FN:** Está en 1FN y todos los atributos no primos dependen de la clave candidata (los atributos depende de su clave).

Sea $R(A, B, C, D, E, F)$ con DFs:

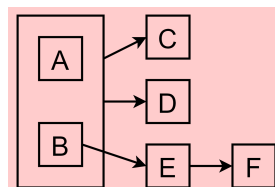
- $AB \rightarrow C$
- $AB \rightarrow D$
- $AB \rightarrow E$
- $E \rightarrow F$



R está en 2FN porque C , D y E dependen de la clave candidata AB y F depende transitivamente de AB .

Sea $R(A, B, C, D, E, F)$ con DFs:

- $AB \rightarrow C$
- $AB \rightarrow D$
- $B \rightarrow E$
- $E \rightarrow F$



R NO está en 2FN porque E depende de B .

Descomposición a 2FN:

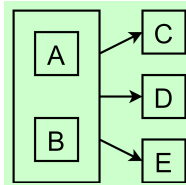
- $R_1(B, E, F)$
- $R_2(A, B, C, D)$



3. **3FN**: Está en 2FN y cada atributo no clave depende de manera no transitiva de atributos clave.

Sea $R(A, B, C, D, E)$ con DFs:

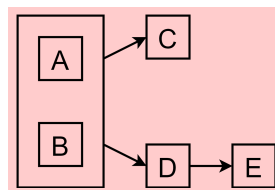
- $AB \rightarrow C$
- $AB \rightarrow D$
- $AB \rightarrow E$



R está en 3FN porque C , D y E dependen de la clave candidata AB .

Sea $R(A, B, C, D, E)$ con DFs:

- $AB \rightarrow C$
- $AB \rightarrow D$
- $D \rightarrow E$



R NO está en 3FN porque E depende transitivamente de AB .

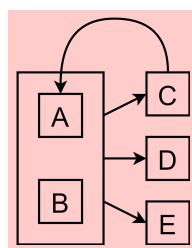
Descomposición a 3FN:

- $R_1(D, E)$
- $R_2(A, B, C, D)$

4. **FNBC**: Está en 3FN y existen dos o más llaves candidatas para la relación, éstas son llaves compuestas (con más de un atributo) y su intersección no es vacía.

Sea $R(A, B, C, D, E)$ con DFs:

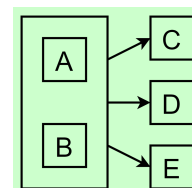
- $AB \rightarrow CDE$
- $C \rightarrow A$



R NO está en FNBC porque C depende de A y A no es clave.

Sea $R(A, B, C, D, E)$ con DFs:

- $AB \rightarrow CDE$



R está en FNBC porque AB es clave.



- **Transacciones:** Las transacciones son el fundamento de la ejecución concurrente y la recuperación de un fallo del sistema en un SGBD
- **SGBD:** Sistemas Gestores de Base de Datos son la opción perfecta para crear, gestionar y administrar las bases de datos que poseas.

Desde el punto de vista de SGBD, una transacción es una serie de operaciones de escritura y lectura a la BD.

Propiedades:

1. La ejecución de cada transacción es atómica: o se realizan todas las acciones o no se realiza ninguna.
2. Consistencia: las transacciones deben preservar la consistencia de la BD.
3. Aislamiento: las transacciones están aisladas, o protegidas, de los efectos de otras transacciones.
4. Durabilidad: si la transacción finaliza exitosamente, sus efectos deben persistir incluso si se produce una caída del sistema.

Acciones:

- $R_t(O)$ denota lectura del objeto O por transacción t .
 - $W_t(O)$ denota escritura de O por t .
 - **Commit(comprometer)** indica que la transacción está terminada satisfactoriamente.
 - **Abort(abortar)** indica que la transacción está terminada y deshechos los cambios realizados en la BD.
- **Plan de ejecución:** es una lista de acciones (lectura, escritura, comprometer o abortar) de un conjunto de transacciones.

Tipo de plan:

- **Plan serial(Secuenciable):** es aquel en el que las transacciones son ejecutadas de inicio a fin, una por una.

Serial: $T_1 \rightarrow T_2$	
T_1	T_2
$R(A)$ $W(A)$ $R(C)$ $W(C)$ $commit_{T_1}$	
	$R(B)$ $W(B)$ $commit_{T_2}$



Conflictos:

- **Escritura-Lectura (WR):** Lectura sucia. T_2 lee un objeto que fue modificado por T_1 , pero que aún no ha sido comprometido.

Sea: $A = 1000$ y $B = 500$.

T_1 transfiere 100 desde A a B.

T_2 incrementa A y B en 10 %.

WR		$T_1 \rightarrow T_2$		$T_2 \rightarrow T_1$	
T_1	T_2	T_1	T_2	T_1	T_2
$R(A), A = 1000$ $W(A), A = 900$		$R(A), A = 1000$ $W(A), A = 900$ $R(B), B = 500$ $W(B), B = 600$ $commit_{T_1}$			$R(A), A = 1000$ $W(A), A = 1100$ $R(B), B = 500$ $W(B), B = 550$ $commit_{T_2}$
	$R(A), A = 900$ $W(A), A = 990$ $R(B), B = 500$ $W(B), B = 550$ $commit_{T_2}$		$R(A), A = 900$ $W(A), A = 990$ $R(B), B = 600$ $W(B), B = 660$ $commit_{T_2}$	$R(A), A = 1100$ $W(A), A = 1000$ $R(B), B = 550$ $W(B), B = 650$ $commit_{T_1}$	
$R(B), B = 550$ $W(B), B = 650$ $commit_{T_1}$					
$A = 990$ y $B = 650$.		$A = 990$ y $B = 660$.		$A = 1000$ y $B = 650$.	

El problema es que T_2 lee el valor de A modificado por T_1 pero no comprometido. Mostrándonos resultados diferentes.

- **Lectura-Escritura(RW):** T_2 modifica el valor de un objeto A que es leído por una T_1 , mientras T_1 está aún en progreso.

Considere las transacciones T_1 y T_2 , y sea A el número de copias disponibles de un libro.

T_1 lee $A = 1$.

T_2 lee $A = 1$, resta 1 y compromete ($A=0$)

T_1 trata de restar 1 a A y recibe un error, A ya no tiene el valor 1

- **Escritura-Escritura(WW):** T_2 sobrescribe el valor de un objeto A que ya ha sido modificado por una transacción T_1 , mientras T_1 todavía está en progreso.

Suponga que los salario de Pedro y Juan deben ser iguales:

T_1 asigna salarios iguales a 2000.

T_2 asigna salarios iguales a 1000.

T_1	T_2
$W(Juan), Salario = 2000$	$W(Pedro), Salario = 1000$
$W(Pedro), Salario = 2000$ $commit_{T_1}$	$W(Juan), Salario = 1000$ $commit_{T_2}$
$P_{salario} \neq J_{salario}$	



- **Plan con Abort:** Cuando una transacción aborta, todas las acciones son deshechas, y la BD vuelve al estado inicial.

Considere las transacciones T_1 y T_2 :

T_1 transfiere 100 desde la cuenta A a la B.

T_2 incrementa A y B por el 10 %.

Supongamos los valores iniciales $A = 1000$ y $B = 500$

T_1	T_2
$R(A), A = 1000$ $W(A), A = 900$ $abort_{T_1}$	$R(A), A = 900$ $W(A), A = 990$ $R(B), B = 500$ $W(B), B = 550$ $commit_{T_2}$

T_2 ha leído un valor de A que no debería haber estado nunca ahí.

Como T_2 ya comprometió sus acciones no pueden ser deshechas

- **Plan recuperable:** las transacciones se comprometen sólo después de que se comprometan a su vez todas las transacciones que hayan leído sus cambios.
- **Planes conflictos equivalentes:** Dos acciones están en conflicto si operan sobre el mismo objeto y al menos una de ellas es escritura.

Podemos cambiar el orden de acciones que no son conflictivas sin alterar el efecto del plan sobre la BD

Todo plan conflicto equivalente es serializable.

Considere el siguiente plan:

$$R_{T_1}(x), R_{T_2}(y), W_{T_3}(x), R_{T_2}(x), R_{T_1}(y)$$

T_1	T_2	T_3
$R(x)$	$R(y)$	
	$R(x)$	$W(x)$
$R(y)$		

 \Rightarrow

T_1	T_2	T_3
$R(x)$		
$R(y)$		
	$R(y)$	$W(x)$
	$R(x)$	

1. Las lecturas de T_1 pueden ser ejecutadas en el siguiente orden:
 $R_{T_1}(x), R_{T_1}(y), R_{T_2}(y), W_{T_3}(x), R_{T_2}(x)$
2. La lectura de T_2 se puede reordenar de la siguiente forma:
 $R_{T_1}(x), R_{T_2}(y), W_{T_3}(x), R_{T_2}(x), R_{T_1}(y)$
3. El plan es conflicto equivalente a la ejecución serial $T_1-T_3-T_2$, y como consecuencia el plan es serializable.



- **Control de Concurrency Basado en Bloqueos:** Un SGBD solo debe permitir la ejecución de planes que son serializables y recuperables.

Tipos de bloqueos:

- **Bloqueo compartido** ($S_T(O)$): Un bloqueo compartido sobre un objeto O permite que una transacción T lea O pero no lo modifique.
- **Bloqueo exclusivo** ($X_T(O)$): Un bloqueo exclusivo sobre un objeto O permite que una transacción T lea y modifique O .

Protocolos:

- **2PL estricto:** El protocolo 2PL estricto permite que solo planes seguros sean ejecutados.

Reglas:

1. Si una transacción T quiere leer (respectivamente, modificar) un objeto, primero solicita un bloqueo compartido (respectivamente, exclusivo) sobre el objeto.
2. Todos los bloqueos concedidos a una transacción se liberan cuando la transacción se completa.

T_1	T_2
$X(A)$ $R(A)$ $W(A)$ $X(B)$ $R(B)$ $W(B)$ $commit_{T_1}$	
	$X(B)$ $R(B)$ $W(B)$ $commit_{T_2}$

T_1	T_2
$S(A)$ $R(A)$	$S(A)$ $R(A)$ $X(B)$ $R(B)$ $W(B)$ $commit_{T_2}$
$X(C)$ $R(C)$ $W(C)$ $commit_{T_1}$	

T_1	T_2
$R(A)$ $W(A)$	$R(A)$ $W(A)$ $R(B)$ $W(B)$ $commit_{T_2}$
$R(B)$ $W(B)$ $commit_{T_1}$	



- **2PL:** En 2PL una transacción puede entregar sus candados antes del final de la transacción (antes de commit o abort).

Reglas:

1. Si una transacción T quiere leer (respectivamente, modificar) un objeto, primero solicita un bloqueo compartido (respectivamente, exclusivo) sobre el objeto.
2. Una transacción no puede pedir candados adicionales una vez que empieza a devolver sus candados.

T_1	T_2
$X(A)$ $R(A)$ $W(A)$ $X(B)$ $liberar X(A)$	$X(A)$ $R(A)$ $W(A)$ $X(C)$ $W(C)$ $liberar X(A)$ $liberar X(C)$ $commit_{T_2}$
$W(B)$ $liberar X(B)$ $commit_{T_1}$	

- **Interbloqueos:** Es un ciclo en el cual dos o más transacción esperan por un bloqueo que la otra tiene.

T_1	T_2
$X(A)$ $R(A)$ $W(A)$ $Solicita X(B)$	$X(B)$ $R(B)$ $W(B)$ $Solicita X(A)$

Prevención: Se le otorga a cada transacción un **nivel de prioridad**, y se le otorga el bloqueo al que tenga mayor prioridad.



- **Recuperación de una BD:**

- **LOG:** es un historial de las acciones ejecutadas por el SGBD.

Operaciones que se registran en el LOG:

- Actualizaciones a páginas: updates.
- Commit: una transacción compromete sus cambios una vez que “commit” es registrado en el LOG y éste ha sido almacenado en disco
- Abort
- End: finalización real de la transacción, se finalizan operaciones extras como borrar archivos, etc.
- Recuperación: se registran todas las actualizaciones que son deshechas debido a abortos de transacciones o fallas del sistema.

- **ARIES:** Algoritmo de recuperación de una BD.

Es invocado después de una caída del sistema.

Fases:

1. Análisis: identificación de “páginas sucias”, i.e. páginas con datos que están en el buffer y no han sido escritos en disco y la identificación de las transacciones activas.
2. Rehacer: repetir todas las acciones, partiendo desde el último punto de revisión (checkpoint) del LOG hasta dejar la BD en el mismo estado antes de la caída.
3. Deshacer: deshacer todas las acciones de transacciones que no se comprometieron. De esta manera, la BD refleja solo las acciones de transacciones comprometidas.

Ejemplo:

LSN		LOG
10	-	update: T_1 escribe $P5$
20	-	update: T_2 escribe $P3$
30	-	T_1 abort
40,45	-	Deshacer T_1 LSN 10, T_1 end
50	-	update: T_3 escribe $P1$
60	-	update: T_2 escribe $P5$
	X	CRASH, RESTART
70	-	Deshacer T_2 LSN 60
80,85	-	Deshacer T_3 LSN 50, T_3 end
	X	CRASH, RESTART
90,95	-	Deshacer T_2 LSN 20, T_2 end



2. Ejercicios

Certamen 2 2022

1. Considere la relación $R(A, B, C, D, E, F)$ con DFs:

- $E \rightarrow F$ (1)
- $A \rightarrow E$ (2)
- $EC \rightarrow A$ (3)
- $ABC \rightarrow D$ (4)

A) Genere las posibles claves candidatas para R. Desarrolle paso a paso nombrando los axiomas de Armstrong. TIP: Existen 2 claves candidatas.

Para la 1° clave candidata comenzamos de ABC :

(4)	$ABC \rightarrow D$	Dado
(5)	$ABC \rightarrow A$	Reflexibilidad (4)
(6)	$ABC \rightarrow B$	Reflexibilidad (4)
(7)	$ABC \rightarrow C$	Reflexibilidad (4)
(8)	$ABC \rightarrow E$	Transitividad de (5) \wedge (2)
(8)	$ABC \rightarrow F$	Transitividad de (8) \wedge (1)

Por lo tanto ABC es una clave candidata pues obtenemos $ABCDEF$.

Para la 2° clave candidata comenzamos de EC :

(3)	$EC \rightarrow A$	Dado
(5)	$EC \rightarrow E$	Reflexibilidad (3)
(6)	$EC \rightarrow C$	Reflexibilidad (3)
(7)	$EC \rightarrow F$	Transitividad de (5) \wedge (1)
(8)	$BCE \rightarrow AB$	Aumentatividad en B
(9)	$BCE \rightarrow A$	Descomposición de (8)
(10)	$BCE \rightarrow B$	Descomposición de (8)
(11)	$BCE \rightarrow C$	Reflexibilidad de (8)
(12)	$BCE \rightarrow ABC$	Union de (8) \wedge (11)
(13)	$BCE \rightarrow D$	Transitividad de (12) \wedge (4)

Por lo tanto BCE es una clave candidata pues obtenemos $ABCDEF$.



B) ¿En cuál forma normal se encuentra R y por qué no cumple con la siguiente forma normal? Justifique.

Respuesta: R se encuentra en 1FN pues todos los atributos son atómicos. R no cumple con 2FN, porque hay DFs que dependen parcialmente de la clave candidata $E \rightarrow F$.

C) Si R no está en FNBC descomponga R paulatinamente hasta dejar un conjunto de relaciones en FNBC (si es posible). Justifique cada paso.

Respuesta:

$R_1(E, F)$

- $E \rightarrow F$

* R_1 está en 3FN y FNBC porque E es super clave.

$R_2(A, B, C, D, E)$

- $A \rightarrow E$
- $EC \rightarrow A$
- $ABC \rightarrow D$

* R_2 está en 3FN pero no en FNBC porque ni A ni EC son super clave.

Descomposición R_2

- $R_{21}(A, E)$
 - $A \rightarrow E$

* R_{21} está en 3FN y FNBC porque A es super clave.

- $R_{22}(A, B, C, D)$
 - $ABC \rightarrow D$

* R_{22} está en 3FN y FNBC porque ABC es super clave.

-
2. Considere la relación $R(A, B, C, D)$, donde el conjunto AB es clave candidata para R.

A) ¿Qué puede concluir a raíz de que AB es clave candidata para R? TIP: nombre las dependencias funcionales que se cumplen en R.

Respuesta: DFs que se cumplen en R:

- $AB \rightarrow C$
- $AB \rightarrow D$

Estas DFs satisfacen R con AB como clave candidata.



B) Agregue una dependencia funcional de tal manera que R no cumpla con la 2FN.

Respuesta:

- $A \rightarrow C$
- $A \rightarrow D$
- $B \rightarrow C$
- $B \rightarrow D$

3. Considere el siguiente plan:

$$S = \{R_3(z), R_3(y), W_2(y), R_2(z), R_3(x), R_1(x), W_1(x), W_2(x)\}$$

A) Demuestre que S es conflicto equivalente a un plan serial.

Respuesta:

$$S = R_3(z), R_3(y), W_2(y), R_2(z), R_3(x), R_1(x), W_1(x), W_2(x)$$

$$R_3(z), R_3(y), R_3(x), W_2(y), R_2(z), R_1(x), W_1(x), W_2(x)$$

$$R_3(z), R_3(y), R_3(x), R_1(x), W_1(x), W_2(y), R_2(z), W_2(x)$$

Original

T_1	T_2	T_3
		$R(z)$
		$R(y)$
	$W(y)$	
	$R(z)$	
		$R(x)$
$R(x)$		
$W(x)$		
	$W(x)$	

Plan Serial

T_1	T_2	T_3
		$R(z)$
		$R(y)$
		$R(x)$
	$W(y)$	
	$R(z)$	
$R(x)$		
$W(x)$		
	$W(x)$	

T_1	T_2	T_3
		$R(z)$
		$R(y)$
		$R(x)$
$R(x)$		
$W(x)$		
	$W(y)$	
	$R(z)$	
	$W(x)$	



B) Muestre la ejecución del plan bajo el protocolo 2PL estricto.

Respuesta: 2PL Estricto

T_1	T_2	T_3
$X(x)$ $R(x)$ $W(x)$ $Commit_{T_1}$	$X(y)$ Esperar T_3 $W(y)$ $S(z)$ $R(z)$ $X(x)$ $W(x)$ $Commit_{T_2}$	$S(z)$ $R(z)$ $S(y)$ $R(y)$ $S(x)$ $R(x)$ $Commit_{T_3}$

C) Muestre la ejecución del plan bajo el protocolo 2PL.

Respuesta: 2PL

T_1	T_2	T_3
$X(x)$ $R(x)$ $W(x)$ $Commit_{T_1}$ y Liberar " x "	$X(y)$ $W(y)$ $S(z)$ $R(z)$ $X(x)$ $W(x)$ $Commit_{T_2}$	$S(z)$ $R(z)$ $S(y)$ $R(y)$ $S(x)$ Liberar " z " y " y " $R(x)$ $Commit_{T_3}$ y Liberar " x "



4. Considere el siguiente historial:

LSN		LOG
10	-	update: T_2 escribe P_5
20	-	update: T_3 escribe P_1
30	-	update: T_1 escribe P_4
40	-	update: T_2 escribe P_6
50	-	T_2 commit
60	-	update: T_1 escribe P_4
70	-	T_2 end
80	-	update: T_3 escribe P_2
90	-	update: T_1 escribe P_6
100	-	update: T_3 escribe P_7
	X	CRASH, RESTART

Complete el LOG con la etapa de restauración de la base de datos.

LSN		LOG
110	-	Deshacer T_3 LSN 100
120	-	Deshacer T_1 LSN 90
130	-	Deshacer T_3 LSN 80
140	-	Deshacer T_1 LSN 60
150	-	Deshacer T_1 LSN 30
160	-	T_1 end
170	-	Deshacer T_3 LSN 20
180	-	T_3 end



Guía Transacciones y Recuperación de Base de Datos

Recuperación de BDs

1. Complete el historial con las acciones efectuadas en la etapa de Deshacer.

LSN		LOG
10	-	update: T_1 escribe P_2
20	-	update: T_1 escribe P_1
30	-	update: T_2 escribe P_5
40	-	update: T_3 escribe P_3
50	-	T_3 commit
60	-	update: T_2 escribe P_5
70	-	update: T_2 escribe P_3
80	-	T_2 abort
90	-	update: T_1 escribe P_4
100	-	T_3 end
	X	CRASH, RESTART
110	-	Deshacer T_1 LSN 90
120	-	Deshacer T_2 LSN 70
130	-	Deshacer T_2 LSN 60
140	-	Deshacer T_2 LSN 30
150	-	T_2 end
160	-	Deshacer T_1 LSN 20
170	-	Deshacer T_1 LSN 10
180	-	T_1 end

2. Complete el historial con las acciones efectuadas en la etapa de Deshacer.

LSN		LOG
10	-	update: T_3 escribe P_2
20	-	update: T_1 escribe P_1
30	-	update: T_3 escribe P_3
40	-	T_3 abort
50	-	update: T_1 escribe P_5
60	-	update: T_2 escribe P_4
70	-	T_1 abort
80	-	update: T_2 escribe P_6
90	-	Deshacer T_3 LSN 30
100	-	Deshacer T_3 LSN 10
110	-	T_3 end
	X	CRASH, RESTART
120	-	Deshacer T_2 LSN 80
130	-	Deshacer T_1 LSN 50
140	-	Deshacer T_2 LSN 60
150	-	T_2 end
160	-	Deshacer T_1 LSN 20
170	-	T_1 end

* Parece que el **abort** tiene preferencia en el Deshacer LSN 130.