CC3201-1
BASES DE DATOS
OTOÑO 2021

Clase 11: Transacciones, ACID y Respaldos

Aidan Hogan aidhog@gmail.com

### Una cuenta bancaria ...

#### Ingreso

cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Deposito iniciál	2020-01-21	20:02:02	300000	300000	TRCXGU8JSHD
7873698669	C0°0°L Designs	2020-02-06	09:15:33	50000	325000	TRCCIA2J8A0

#### **Gasto**

cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

#### Cuenta

<u>número</u>	rut	tipo	saldo_clp	saldo_usd	
7873698669	32.000.273	3-K Estacional	225000	344,42	
Cliente					
rut	nombre	fono	dirección	6 00	do 10
32.000.273-K	Kelvin	+56976698463	3 Campo de	Hielo Sur, De	pto 273

Divisa		
<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,0001533
USD	CLP	652,2750000

# Una cuenta bancaria ... integridad

10	~	re	-	$\sim$
	v	re	. >	t J
•	~	•		•

cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Deposito iniciál	2020-01-21	20:02:02	300000	300000	TRCXGU8JSHD
7873698669	C0°0°L Designs	2020-02-06	09:15:33	50000	325000	TRCCIA2J8A0

#### **Gasto**

cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

#### Cuenta

número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,42

#### Cliente

rut	nombre	fono	dirección
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

#### **Divisa**

<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,0001533
USD	CLP	652,2750000

### Restricciones sobre varias tablas

#### Ingreso

cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Deposito iniciál	2020-01-21	20:02:02	300000	300000	TRCXGU8JSHD
7873698669	C0°0°L Designs	2020-02-06	09:15:33	50000	325000	TRCCIA2J8A0

#### Gasto

cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

Cuenta				
número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

### CREATE TABLE Cuenta (

número INTEGER PRIMARY KEY, rut VARCHAR (12) NOT NULL, tipo VARCHAR (12) NOT NULL, saldo\_clp BIGINT NOT NULL, saldo\_usd FLOAT NOT NULL, CHECK (



¿A. P. A.? ...

INSERT INTO Ingreso ...

INSERT INTO Gasto ...

UPDATE Cuenta

SET saldo\_clp ...

¿A. S.? ...

( SELECT SUM(monto) FROM Ingreso WHERE cuenta=número ) - ( SELECT SUM(monto) FROM Gasto WHERE cuenta=número )

= saldo\_clp ) )

# **TRANSACCIONES**

### Transacciones

Una transacción es

un conjunto de operaciones que

se ejecutan de manera atómica

(es decir, como si fuera una sola operación)

# Vacaciones ...



# Transacciones: START TRANSACTION/COMMIT

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD
7873698669	Noruega	2020-02-12	02:14:20	400000	-175000	TRCLK9K24KS

C	u	e	n	t	a
	_				

número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	-175000	-268,29

#### **Cliente**

rut	nombre	fono	dirección
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

#### Divisa

<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,001533
USD	CLP	652,275000

#### START TRANSACTION;

INSERT INTO Gasto VALUES

```
(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');

UPDATE Cuenta SET saldo_clp=-175000, saldo_usd=-268.29 WHERE número=7873698669;

COMMIT;
```

START TRANSACTION (o a veces BEGIN) inicia la transacción COMMIT realiza/guarda los cambios

# Transacciones (por defecto)

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD
7873698669	Noruega	2020-02-12	02:14:20	400000	-175000	TRCLK9K24KS

Cuenta				
número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	-175000	-268,29
Cliente				

1013090009	32.000.273	-N Estacional	-175000 -200,29	CLP	USD	0.0
Cliente				USD	CLP	652,2
rut	nombre	fono	dirección			
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273			

Divisa		
<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,001533
USD	CLP	652,275000

```
INSERT INTO Gasto VALUES
```

```
(7873698669, 'Noruega', '2020-02-12', '02:14:20',400000,-175000, 'TRCLK9K24KS'); -- COMMIT;
```

```
UPDATE Cuenta SET saldo_clp=-175000, saldo_usd=-268.29 WHERE número=7873698669;
-- COMMIT;
```

Si no hay una transacción explícita, por defecto, Postgres hace un COMMIT después de cada sentencia (pero se puede cambiar la configuración)

### Transacciones: ROLLBACK

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

Cuenta						
número	rut		tipo	saldo_clp	saldo_usd	
7873698669	32.000.273	B-K	Estacional	225000	344,94	
Cliente						
rut	nombre	for	10	dirección		
32.000.273-K	Kelvin	+5	6976698463	Campo de	Hielo Sur, Dept	o 273

Divisa		
<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,001533
USD	CLP	652,275000

#### START TRANSACTION;

INSERT INTO Gasto VALUES

(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');

ROLLBACK;

ROLLBACK deshace/borra los cambios desde el inicio de la transacción

### Transacciones: SAVEPOINT

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD
7873698669	Noruega	2020-02-12	02:14:20	400000	-175000	TRCLK9K24KS

Cuenta					
número	rut	tipo	saldo_clp	saldo_usd	
7873698669	32.000.273	-K Estacional	-175000	-268,29	
Cliente					
rut	nombre	fono	dirección		
32.000.273-K	Kelvin	+56976698463	Campo de	Hielo Sur, Depto	273

Divisa		
<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,001533
USD	CLP	652,275000

#### START TRANSACTION:

COMMIT:

```
INSERT INTO Gasto VALUES
    (7873698669,'Noruega','2020-02-12','02:14:20',400000,-175000,'TRCLK9K24KS');
UPDATE Cuenta SET saldo_clp=-175000, saldo_usd=-268.25 WHERE número=7873698669;
SAVEPOINT CompraNoruega;
INSERT INTO Gasto VALUES
    (7873698669,'BOSE','2020-02-12',200000,-375000,'TRCASD8PNAK');
ROLLBACK TO SAVEPOINT CompraNoruega;
```

ROLLBACK puede deshacer/borrar los cambios desde un punto específico con SAVEPOINT

### Una transacción con valores dinámicos

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

		$\boldsymbol{\alpha}$	n	•	7
C	u	C		L	а
_		_		_	

número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

#### **Cliente**

rut	nombre	fono	dirección
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

# Divisa d2 valor CLP USD 0.001533

```
USD CLP 652,275000
```

#### START TRANSACTION;

COMMIT:

```
INSERT INTO Gasto VALUES
   (7873698669,'Noruega','2020-02-12','02:14:20',400000,-175000,'TRCLK9K24KS');
UPDATE Cuenta SET saldo_clp=(saldo_clp-400000) WHERE número=7873698669;
UPDATE Cuenta SET saldo_usd=(A.saldo_clp/valor)
   FROM ( SELECT valor FROM Divisa WHERE d1='USD' AND d2='CLP') T,
        ( SELECT saldo_clp FROM Cuenta WHERE número=7873698669 ) A
   WHERE número=7873698669;
```

### ¿Valor final de saldo usd en Cuenta?

### Una transacción con valores dinámicos

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD
7873698669	Noruega	2020-02-12	02:14:20	400000	-175000	TRCLK9K24KS

Cuenta				
número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	-175000	-268,28
Cliente				

Cliente			
rut	nombre	fono	dirección
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

```
        Divisa
        d2
        valor

        CLP
        USD
        0,001533

        USD
        CLP
        652,275000
```

#### START TRANSACTION;

```
INSERT INTO Gasto VALUES
     (7873698669,'Noruega','2020-02-12','02:14:20',400000,-175000,'TRCLK9K24KS');
UPDATE Cuenta SET saldo_clp=(saldo_clp-400000) WHERE número=7873698669;
UPDATE Cuenta SET saldo_usd=(A.saldo_clp/valor)
    FROM ( SELECT valor FROM Divisa WHERE d1='USD' AND d2='CLP') T,
          ( SELECT saldo_clp FROM Cuenta WHERE número=7873698669 ) A
    WHERE número=7873698669;
COMMIT;
```

¿Valor final de saldo\_usd en Cuenta?

-268, 28 (Se lee el valor actual de la misma transacción)

### Una transacción con CHECK

Gasto						
cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Electricidad	2020-02-02	20:00:01	8200	291800	TRCJASJDA9A
7873698669	Calefacción	2020-02-02	20:00:02	600	291200	TRC81KAQWAS
7873698669	Moviestar	2020-02-02	20:00:03	16200	275000	TRCK8J7JA8D
7873698669	Cajero	2020-02-08	16:05:02	100000	225000	TRCPM8A45AD

	10	nt	
Cı		HL	C

número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

#### **Cliente**

rut	nombre	fono	dirección
32.000.273-K	Kelvin	+56976698463	Campo de Hielo Sur, Depto 273

#### Divisa

<u>d1</u>	<u>d2</u>	valor
CLP	USD	0,001533
USD	CLP	652,275000

#### START TRANSACTION;

INSERT INTO Gasto VALUES

(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');

UPDATE Cuenta SET saldo\_clp=(saldo\_clp-400000) WHERE número=7873698669;

UPDATE Cuenta SET saldo usd=(A.saldo clp/valor)

FROM ( SELECT valor FROM Divisa WHERE d1='USD' AND d2='CLP') T,

( SELECT saldo\_clp FROM Cuenta WHERE número=7873698669 ) A

WHERE número=7873698669;

¡No!

¿Funciona?

COMMIT;

```
CREATE TABLE Cuenta ( ...

CHECK ( saldo_clp = saldo_usd * (SELECT valor FROM Divisa WHERE d1='USD' AND d2='CLP') ) )
```

# Transacciones / Restricciones: IMMEDIATE

Cuenta				
número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

```
CREATE TABLE Cuenta (
...,
CONSTRAINT Cuenta_PK
PRIMARY KEY (número)
)
```

```
START TRANSACTION;
INSERT INTO Cuenta VALUES
(7873698669,'32.000.273-K','Estacional',
-175000,-268.29);
DELETE FROM Cuenta
WHERE número=7873698669 AND saldo_clp=225000;
COMMIT;
```

Por defecto, se aplica la restricción inmediatamente después de cada sentencia

# Transacciones / Restricciones: DEFERRABLE

Cuenta				
número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

### Cuenta

número	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	-175000	-268,29

```
CREATE TABLE Cuenta (
...,
CONSTRAINT Cuenta_PK
PRIMARY KEY (número)
DEFERRABLE
)
```

```
START TRANSACTION;
```

SET CONSTRAINT Cuenta\_PK DEFERRED;

INSERT INTO Cuenta

(7873698669, '32.000.273-K', 'Estacional',

-175000,-268.29);

DELETE FROM Cuenta

WHERE número=7873698669 AND saldo\_clp=225000;

COMMIT;

DEFERRABLE define una restricción que se puede diferir hasta un COMMIT

DEFERRED difiere la restricción hasta el COMMIT en la transacción actual

# Transacciones / Restricciones: DEFERRABLE

Cuenta				
<u>número</u>	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94

#### Cuenta

<u>número</u> rut		tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	-175000	-268,29

```
CREATE TABLE Cuenta (
...,
CONSTRAINT Cuenta_PK
PRIMARY KEY (número)
DEFERRABLE INITIALLY DEFERRED
)
```

DEFERRABLE INITIALLY DEFERRED

define una restricción que sea diferida por

defecto hasta un COMMIT

```
START TRANSACTION;
```

DELETE FROM Cuenta

WHERE número=7873698669 AND saldo\_clp=225000;
COMMIT;

Atomicidad, Coherencia, Aislamiento, Durabilidad (*Atomicity, Consistency, Isolation, Durability*)

### LAS GARANTÍAS DE ACID

# No hay un solo usuario ...



... hay que tener cuidado con la concurrencia

### Una cuenta con varios usuarios





N° Cuenta : 7873698669

Saldo (CLP) : 225000

Límite de crédito : 200000

Disponible : 425000



```
CREATE TABLE Cuenta ( ..., CHECK ( saldo_clp > -200000 ) )
```

COMPRA(Islas de Caimán, 300000)

COMPRA(Noruega, 400000)

```
UPDATE Cuenta ...
```

```
INSERT INTO Gasto ...
UPDATE Cuenta ...
```

# Caos



### Esta vez con transacciones ...





N° Cuenta : 7873698669

Saldo (CLP) : 225000

Límite de crédito : 200000

Disponible : 425000



```
CREATE TABLE Cuenta ( ..., CHECK ( saldo_clp > -200000 ) )
```

COMPRA(Islas de Caimán, 300000)

COMPRA(Noruega, 400000)

```
START TRANSACTION

INSERT INTO Gasto ...

UPDATE Cuenta ...

COMMIT;
```

```
START TRANSACTION

INSERT INTO Gasto ...

UPDATE Cuenta ...

COMMIT;
```

### Garantías de ACID

- Atomicidad:
  - La ejecución de cada transacción es atómica:
    - Se realizan todas las acciones o no se realiza ninguna
- Coherencia:
  - Cada transacción debe preservar la integridad
    - La base de datos satisface todas las restricciones después de una transacción
- Aislamiento (Isolation):
  - Una transacción no puede afectar otra
- Durabilidad:
  - Una vez que haya un COMMIT, la base de datos debe guardar los cambios

### **ACID:** Atomicidad

```
CREATE TABLE Balance (
    cuenta BIGINT PRIMARY KEY,
    total_gasto BIGINT,
    total_ingreso BIGINT,
    saldo BIGINT,
    CHECK (total_ingreso - total_gasto = saldo)
)
```

```
START TRANSACTION
   UPDATE Balance SET saldo=saldo-10 WHERE Cuenta=7873698669 ;
   UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;
COMMIT;
```

### **Atomicidad**

No se puede actualizar el saldo sin actualizar el gasto directamente después. (Si alguna actualización falla, ambas fallan.)

### ACID: Coherencia (Consistencia)

```
CREATE TABLE Balance (
    cuenta BIGINT PRIMARY KEY,
    total_gasto BIGINT,
    total_ingreso BIGINT,
    saldo BIGINT,
    CHECK (total_ingreso - total_gasto = saldo)
)
```

```
START TRANSACTION
    UPDATE Balance SET saldo=saldo-100 WHERE Cuenta=7873698669 ;
    UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;
COMMIT;
```



### Coherencia

Si el resultado de la transacción no satisface todas las restricciones, fallará.

# ACID: Aislamiento (Isolation)

```
CREATE TABLE Balance (
   cuenta BIGINT PRIMARY KEY,
   total_gasto BIGINT,
   total_ingreso BIGINT,
   saldo BIGINT,
   CHECK (total_ingreso - total_gasto = saldo)
START TRANSACTION
                                          START TRANSACTION
                                 T1
 UPDATE Balance
                                            UPDATE Balance
    SET saldo=saldo-10
                                              SET saldo=saldo+100
   WHERE Cuenta=7873698669 :(1)
                                              WHERE Cuenta=7873698669;
 UPDATE Balance
                                            UPDATE Balance
    SET total_gasto=total_gasto+100
                                              SET total_ingreso=total_ingreso+100
                                              WHERE Cuenta=7873698669 ; (5)
    WHERE Cuenta=7873698669 ;(3)
                                          COMMIT; (6)
COMMIT;
                   (4) ROLLBACK;
```

### **Aislamiento**

Una transacción no puede interferir con otra transacción.

En (4), hay que tener cuidado con el ROLLBACK: no se puede restaurar el valor de saldo antes del paso (1) porque el valor ya fue cambiado por (2).

### ACID: Durabilidad

```
CREATE TABLE Balance (
    cuenta BIGINT PRIMARY KEY,
    total_gasto BIGINT,
    total_ingreso BIGINT,
    saldo BIGINT,
    CHECK (total_ingreso - total_gasto = saldo)
)
```

```
START TRANSACTION
   UPDATE Balance SET saldo=saldo-10 WHERE Cuenta=7873698669 ;
   UPDATE Balance SET total_gasto=total_gasto+10 WHERE Cuenta=7873698669 ;
   COMMIT;
```

### Durabilidad

Una vez que haya un **COMMIT** exitoso, se persisten los cambios.

(Normalmente la persistencia aquí significa en el disco duro. Sin persistencia, en el caso de que la máquina falle y toda la evidencia de los cambios esté en la memoria principal, el sistema de base de datos olvidará los cambios silenciosamente.)

### ¿Cuándo no tenemos ACID?

### • Atomicidad:

Vina transacción se ejecuta solo a medias pero afecta el estado de la base de datos

### Coherencia:

- X Al ejecutar la transacción, la base de datos no satisface las restricciones de integridad
- Aislamiento (Isolation):
  - El resultado final de dos transacciones no es equivalente a ejecutar cada transacción en serie

### Durabilidad:

La base de datos se actualiza momentáneamente y luego vuelve al estado anterior.

# ACID implementado por muchos sistemas



# ACID implementado por muchos sistemas

### Registros

- Se escriben todas las operaciones al disco duro
- Permiten recordar y deshacer/rehacer operaciones

### Bloqueos

- Cuando una transacción actualice x (sea una fila, una tabla, etc.), adquiere un bloqueo exclusivo sobre x tal que tenga acceso exclusivo a x
- Cuando una transacción lea x (sea una fila, una tabla, etc.), adquiere un bloqueo no exclusivo sobre x tal que otras transacciones puedan leer x pero no puedan actualizar x

# Entonces con las garantías de ACID ...



# ... todo está tranquilo.

\* asumiendo que no hay errores de usuario, hardware o software

# RESPALDOS

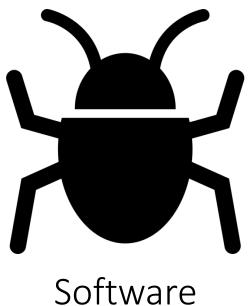
# Tipos de errores



Humano



Hardware



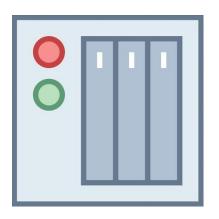
# Tipos de respaldo



Nivel del sistema de base de datos

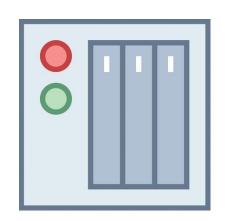


Nivel del sistema de archivos



Nivel del hardware





RESPALDOS EXTERNOS

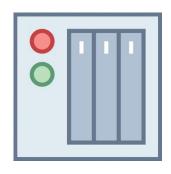
# Respaldos Externos: Sistema de Archivos



Usar métodos estándares de respaldar archivos

- ✓ Algo simple
- Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- Mantener una historia de copias ante errores de usuario o software puede ocupar mucho espacio
- En general, habría que detener el sistema de base de datos para hacer un respaldo "coherente"
- × No podemos consultar los respaldos

# Respaldos Externos: Hardware



- Replicar el disco (p.ej., RAID-1) o la máquina
- ✓ Protección ante errores de hardware
- ✓ Podemos consultar el respaldo (más lecturas)
- ✓ No es necesario desactivar el sistema de base de datos
- Mantener la réplica actualizada puede tener un costo (en particular en el caso de usar otra máquina)
- En sí, no provee protección ante errores humanos o de software (solo respaldará el estado actual)
- × Existe el costo de comprar y mantener el hardware adicional



RESPALDOS INTERNOS

# Respaldos Internos: Completos



 Respaldar todos los datos dentro del sistema de base de datos cada vez (por ejemplo, cada noche)



- ✓ No hay que parar el sistema de base de datos
- ✓ Más fácil de cargar de nuevo (que las opciones que siguen)
- Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- Mantener una historia de copias completas puede ocupar mucho espacio
- × No podemos consultar los respaldos

# Respaldos Internos: Diferencial



 Respaldar todos los datos que hayan cambiado desde el último respaldo completo

1	Δ(2,1)	Δ(3,1)	Δ(4,1)	5	Δ(6,5)	
---	--------	--------	--------	---	--------	--

- ✓ No hay que detener el sistema de base de datos
- Puede mantener una historia de copias en menos espacio, pero el espacio usado aumenta en función del último respaldo completo
- La carga del respaldo será un poco más costosa que en el caso del respaldo completo
- Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- × No podemos consultar los respaldos

# Respaldos Internos: Incremental



 Respaldar todos los datos que han cambiado desde el último respaldo completo o incremental

1	Δ(2,1)	Δ(3,2)	Δ(4,3)	5	Δ(6,5)	
---	--------	--------	--------	---	--------	--

- ✓ No hay que detener el sistema de base de datos
- ✓ Puede mantener una historia de copias en menos espacio, y el espacio no aumentará en función del último respaldo completo
- Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- La carga del respaldo será un poco más costosa que en el caso del respaldo completo
- × No podemos consultar los respaldos

# Respaldos Internos: Registros



 Respaldar el registro de transacciones del sistema de base de datos (al nivel de sistema de archivos)



- ✓ No hay que detener el sistema de base de datos
- ✓ Se pueden repetir las transacciones desde un punto particular, o hasta un punto particular, o saltando una transacción
- ✓ No hay que mantener copias históricas (contiene la historia)
- Solo hay protección ante errores de hardware si se usa otro disco para respaldar la información
- La carga del respaldo será significativamente más costosa que en los otros casos
- × No podemos consultar los respaldos directamente

# Respaldos Internos: En la práctica



- No está en el estándar de SQL
  - Hay que revisar la documentación del sistema de base de datos particular
- Ejemplos de respaldos completos

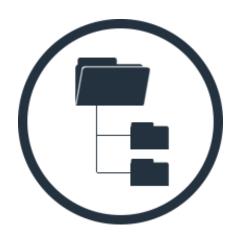
```
MySQL: shell> mysqldump --databases cc3201 > dump.msql

Postgres: shell> pg_dump cc3201 > dump.psql

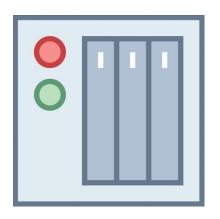
SQL Server shell> BACKUP DATABASE cc3201 TO DISK = 'dump.sqls'
```



Nivel del sistema de base de datos



Nivel del sistema de archivos



Nivel del hardware



Hemos terminado con bases de datos relacionales

# Preguntas?

