

CC3201-1

BASES DE DATOS

PRIMAVERA 2016

Clase 11: Integridad, Transacciones, ACID (I)

Aidan Hogan

aidhog@gmail.com

Un programador *freelance* abre una cuenta



Banco de Chilly

Y (por supuesto) hay una base de datos

Ingreso

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

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Divisa

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

Cliente

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

La base de datos: integridad

Ingreso

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

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

<u>número</u>	<u>rut</u>	<u>tipo</u>	<u>saldo_clp</u>	<u>saldo_usd</u>
7873698669	32.000.273-K	Estacional	225000	344,42

Divisa

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

Cliente

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

Modelo Relacional: Restricciones

Restricciones (*de integridad*):

son **restricciones** formales

que imponemos a **un esquema**

que todas sus instancias

deben satisfacer

CC3201-1
BASES DE DATOS
PRIMAVERA 2016

Clase 2: Modelo Relacional

Aidan Hogan
aidhog@gmail.com

(Integrity Constraints)

RESTRICCIONES DE INTEGRIDAD

Capítulo 5.7 Database Management Systems,
Ramakrishnan / Gehrke (Third Edition)

Restricciones básicas: llaves, nulos, domino

Cuenta

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

```
CREATE TABLE Cuenta (  
  número BIGINT PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL,  
  saldo_usd DOUBLE PRECISION  
  NOT NULL
```

```
INSERT INTO Cuenta VALUES  
  (7873698669, '28.923.123-7', 'Estacional', 1000, 1.53)
```

```
UPDATE Cuenta SET tipo=NULL WHERE número=7873698669
```

```
INSERT INTO Cuenta VALUES  
  (7273697679, '28.923.0123-7', 'Estacional', 1000, 1.53)
```

Restricciones básicas: valores por defecto

Cuenta				
<u>número</u>	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94
7273697679	28.923.123-7	Estacional	0	0,00

Banco de Chile

```
CREATE TABLE Cuenta (  
  número BIGINT PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL  
    DEFAULT 0,  
  saldo_usd DOUBLE PRECISION  
    NOT NULL DEFAULT 0  
)
```

```
INSERT INTO Cuenta (número, rut, tipo)  
VALUES (7273697679, '28.923.123-7', 'Estacional')
```

Restricciones de unicidad

Cuenta				
<u>número</u>	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94
7273697679	28.923.123-7	Estacional	0	0,00

```
CREATE TABLE Cuenta (  
  número INTEGER PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL  
    DEFAULT 0,  
  saldo_usd FLOAT NOT NULL  
    DEFAULT 0,  
  UNIQUE (rut,tipo)  
)
```

```
INSERT INTO Cuenta (número, rut, tipo)  
VALUES (8079766582, '28.923.123-7', 'Estacional')
```

La llave primaria implica una restricción de unicidad. La unicidad representa una llave candidata: se pueden tener varias llaves candidatas pero una sola llave primaria.

Nombrar (y borrar) restricciones

Cuenta				
<u>número</u>	rut	tipo	saldo_clp	saldo_usd
7873698669	32.000.273-K	Estacional	225000	344,94
7273697679	28.923.123-7	Estacional	0	0,00

Banco de Chile

```
CREATE TABLE Cuenta (  
  número INTEGER,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL  
    DEFAULT 0,  
  saldo_usd FLOAT NOT NULL  
    DEFAULT 0,  
  CONSTRAINT Cuenta_uni_rt UNIQUE (rut,tipo),  
  CONSTRAINT Cuenta_pk PRIMARY KEY (número) )
```

```
ALTER TABLE Cuenta  
DROP CONSTRAINT Cuenta_uni_rt
```

*Más fácil cambiar restricciones
posteriormente.*

*Si hay una violación, el mensaje de error
será más intuitiva si las restricciones
tienen nombres intuitivos.*

Crear dominios: VARCHAR

Ingreso

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

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

```
CREATE DOMAIN tr_str VARCHAR (12)  
CHECK ( VALUE LIKE 'TRC%' );
```

```
CREATE TABLE Ingreso (  
... ,  
id tr_str PRIMARY KEY );
```

```
CREATE TABLE Gasto (  
... ,  
id tr_str PRIMARY KEY );
```

```
UPDATE Ingreso SET id='XJ1' WHERE id='TRCXGU8JSHD'
```

```
UPDATE Gasto SET id='8AS' WHERE id='TRCPM8A45AD'
```

Crear dominios: INTEGER

Ingreso

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

Cuenta

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

```
CREATE DOMAIN c_no BIGINT  
CHECK ( VALUE > 999999999  
AND VALUE <= 9999999999);
```

```
CREATE TABLE Ingreso (  
cuenta c_no, ... );
```

```
CREATE TABLE Cuenta (  
número c_no PRIMARY KEY, ... );
```

```
UPDATE Ingreso SET cuenta=65537 WHERE id='TRCXGU8JSHD'
```

```
UPDATE Cuenta SET número=691 WHERE número=7873698669
```

Dominios: compatibles con el tipo base

Ingreso

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

Cuenta

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

```
CREATE DOMAIN c_no BIGINT
CHECK ( VALUE > 999999999
AND VALUE <= 9999999999);
```

```
CREATE TABLE Ingreso (
cuenta c_no, ... );
```

```
CREATE TABLE Cuenta (
número c_no PRIMARY KEY, ... );
```

```
SELECT rut FROM Cuenta
WHERE número > saldo_clp
```

rut

32.000.273-K

Se puede comparar valores del dominio con otros valores como fuera del tipo base

Tipos: son distintos a otros tipos

Ingreso

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

Cuenta

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

```
CREATE TYPE c_no_t UNDER BIGINT;
```

```
CREATE TABLE Ingreso (
  cuenta c_no_t, ... );
```

```
CREATE TABLE Cuenta (
  número c_no_t PRIMARY KEY, ... );
```

```
SELECT rut FROM Cuenta
WHERE número > saldo_clp
```



No se puede comparar valores del nuevo tipo con valores de otros tipos (solo entre sí).

Tipos: son distintos a otros tipos

Ingreso

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

Cuenta

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

```
CREATE TYPE c_no_t UNDER BIGINT;
```

```
CREATE TABLE Ingreso (
  cuenta c_no_t, ... );
```

```
CREATE TABLE Cuenta (
  número c_no_t PRIMARY KEY, ... );
```

```
SELECT número, (número+1) AS siguiente
FROM Cuenta
```



No se puede usar funciones del tipo base con valores del nuevo tipo.



WARNING

Tipos son estándares (en SQL)

Pero Postgres solo soporte tipos “complejos”

Restricciones sobre varias columnas

Cuenta

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

Banco de Chile

```
CREATE TABLE Cuenta (  
  número BIGINT PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL,  
  saldo_usd DOUBLE PRECISION NOT NULL,  
  CHECK ( ROUND ( (saldo_clp/saldo_usd)::NUMERIC - 652.275 , 1 ) = 0 )  
)
```

```
INSERT INTO Cuenta VALUES  
(7273697679, '28.923.123-7',  
'Estacional', 100, 0.99)
```



Restricciones sobre varias tablas

Cuenta

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

Divisa

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

```
CREATE TABLE Cuenta (  
  número BIGINT PRIMARY KEY,  
  rut VARCHAR (12) NOT NULL,  
  tipo VARCHAR (12) NOT NULL,  
  saldo_clp BIGINT NOT NULL,  
  saldo_usd DOUBLE PRECISION NOT NULL,  
  CHECK ( ROUND((saldo_clp/saldo_usd)::NUMERIC -  
    ( SELECT valor FROM Divisa  
      WHERE d1='USD' AND d2='CLP' ) , 1 ) = 0  
)
```

```
INSERT INTO Cuenta VALUES  
(7273697679, '28.923.123-7',  
'Estacional', 100, 0.99)
```





WARNING

Postgres no permite consultas anidadas en CHECK.

Restricciones de llaves externas

Ingreso

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

Cuenta

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

```
CREATE TABLE Ingreso (  
  cuenta BIGINT REFERENCES Cuenta(número),  
  comentario VARCHAR (255),  
  fecha DATE NOT NULL,  
  hora TIME NOT NULL,  
  monto BIGINT NOT NULL,  
  saldo INT NOT NULL,  
  id VARCHAR (12) PRIMARY KEY  
)
```

```
INSERT INTO Ingreso VALUES  
  (7273697679, ... , ...)
```

*Cada **cuenta** en **Ingreso** tiene que estar en **Cuenta.número**.*

Restricciones sobre varias tablas (!)

Ingreso

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

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

```
CREATE TABLE Ingreso (  
  cuenta INTEGER,  
  ...,  
  CHECK(  
    ( SELECT COUNT(*) FROM Ingreso I  
      WHERE I.fecha=fecha AND I.cuenta=cuenta ) +  
    ( SELECT COUNT(*) FROM Gasto G  
      WHERE G.fecha=fecha AND G.cuenta=cuenta ) < 1000  
  )  
)
```

¿Algún problema aquí? ...

¿Por qué la ponemos en Ingreso cuando involucra Gasto igualmente? Por ejemplo, si agregáramos la milésima tupla (con la misma cuenta y fecha) a Gasto, ¿no tendríamos una violación!

¿Alguna solución?

Duplicar la restricción en Gasto o ...



WARNING

Postgres no permite consultas anidadas en CHECK.

Asertos: Restricciones independientes

Ingreso

cuenta	comentario	fecha	hora	monto	saldo	<u>id</u>
7873698669	Deposito inicial	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

```
CREATE ASSERTION MaxTransferenciasDiarias
CHECK (
  ( SELECT MAX(num)
    FROM
      ( SELECT COUNT(*) AS num
        FROM
          ( SELECT * FROM Ingreso
            UNION
              SELECT * FROM Gasto ) Trans
          GROUP BY fecha, cuenta ) TransC ) ) < 1000 )
```

Rechazará alguna operación en el esquema que violaría la restricción

La restricción no depende de ni una tabla ni la otra.

... pero puede ser más costosa/compleja así.

¡Garantizar integridad con restricciones!

Ingreso

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

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Divisa

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

Cliente

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

Restricciones sobre varias tablas (!!)

Ingreso

cuenta	comentario	fecha	hora	monto	saldo	id
7873698669	Deposito inicial	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 (  
    ( SELECT SUM(monto) FROM Ingreso WHERE cuenta=número )  
    - ( SELECT SUM(monto) FROM Gasto WHERE cuenta=número )  
    = saldo_clp ) )
```



¿A. P. A.? ...

INSERT INTO Ingreso ...

INSERT INTO Gasto ...

UPDATE Cuenta
SET saldo_clp ...

¿A. S.? ...

TRANSACCIONES

Transacciones

Una **transacción** es

un **conjunto de operaciones** que

se ejecutan de **manera atómica**

(es decir, **como fueran una sola operación**)

Queda menos de una semana
antes del día de San Valentín ...



... y Kelvin quiere organizar una
vacación para él y su polola

Transacciones: START TRANSACTION/COMMIT

Gasto

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Divisa

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

Cliente

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

```
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

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Divisa

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

Cliente

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

```
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

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Cliente

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

Divisa

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

```
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

<u>cuenta</u>	<u>comentario</u>	<u>fecha</u>	<u>hora</u>	<u>monto</u>	<u>saldo</u>	<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

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

Divisa

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

Cliente

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

```
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=-268.25 WHERE número=7873698669;  
SAVEPOINT CompraNoruega;  
INSERT INTO Gasto VALUES  
  (7873698669, 'BOSE', '2020-02-12', '20:00:01', 200000, -375000, 'TRCASD8PNAK');  
ROLLBACK TO SAVEPOINT CompraNoruega;  
COMMIT;
```

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

Transacciones / Restricciones: IMMEDIATE

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

Banco de Chile

```
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
)
```

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

DEFERRABLE INITIALLY DEFERRED define una restricción que sea diferido por defecto hasta un **COMMIT**



WARNING

Restricciones diferibles son estándares

Pero Postgres no permite diferir
restricciones usando ni CHECK ni NOT NULL

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

Cuenta

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

Divisa

d1	d2	valor
CLP	USD	0,0001533
USD	CLP	652,2750000

Cliente

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

```
START TRANSACTION;
INSERT INTO Gasto VALUES
(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');
SELECT (saldo_clp - 400000) AS saldo_actual
FROM Cuenta WHERE número=7873698669;
UPDATE Cuenta SET saldo_clp=A.saldo_clp, 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;
COMMIT;
```



¿Valor final de `saldo_clp` en Cuenta?

¡22500! Leemos un valor de Gasto que no hemos guardado

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

Cuenta

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

Divisa

d1	d2	valor
CLP	USD	0,0001533
USD	CLP	652,2750000

Cliente

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

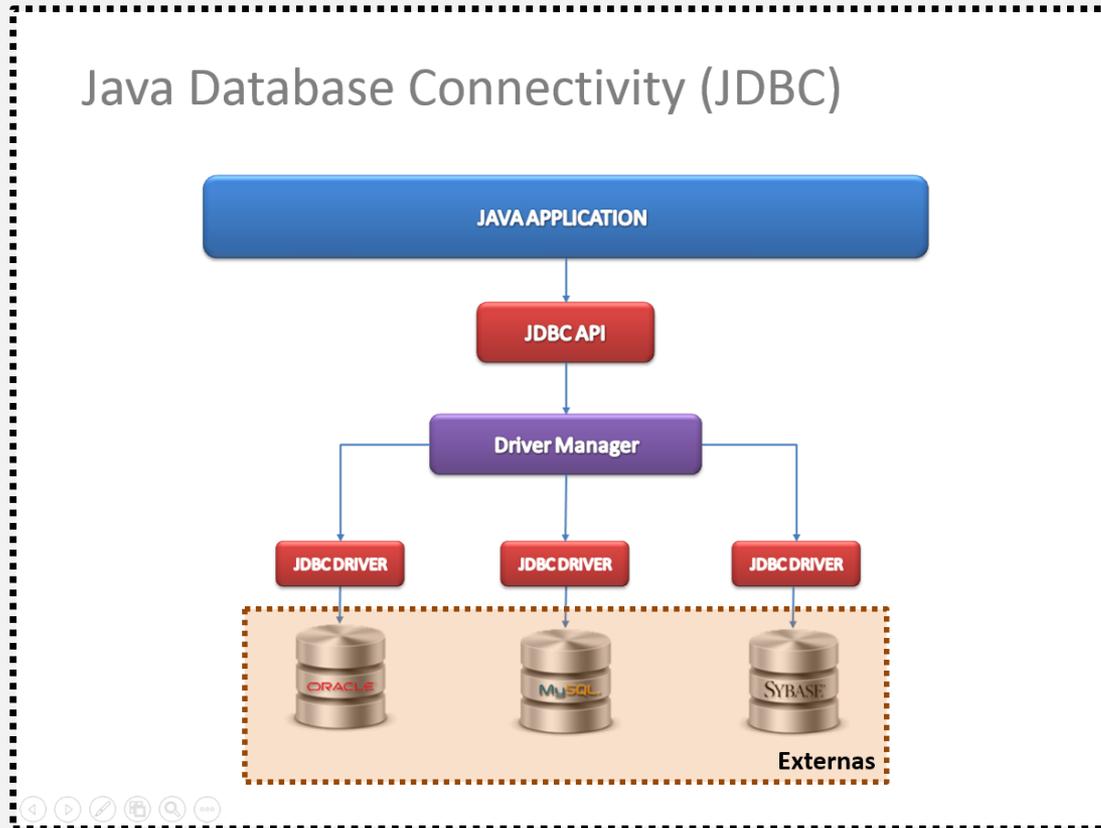
```

START TRANSACTION;
INSERT INTO Gasto VALUES
(7873698669, 'Noruega', '2020-02-12', '02:14:20', 400000, -175000, 'TRCLK9K24KS');
UPDATE Cuenta SET saldo_clp=saldo_actual
FROM ( SELECT (saldo_clp - 400000) AS saldo_actual
      FROM Cuenta WHERE número=7873698669 ) A;
UPDATE Cuenta SET saldo_clp=A.saldo_clp, 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;
COMMIT;

```



Si parece demasiado complejo, se puede usar acceso programático con transacciones ...



... veremos más en el lab.

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

LAS GARANTIAS DE ACID

Capítulo 8.1, Database Management Systems,
Ramakrishnan / Gehrke (Third Edition)

No hay un solo usuario ...



... hay que tener cuidado con la concurrencia

Una cuenta con varios usuarios



Banco de Chilly

N° Cuenta : 7873698669
Saldo (CLP) : 225000
Límite de crédito : 200000
Disponible : 425000



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

COMPRA(Islas de Caimán,300000)

```
INSERT INTO Gasto ...
```

```
UPDATE Cuenta ...
```

COMPRA(Noruega,400000)

```
INSERT INTO Gasto ...
```

```
UPDATE Cuenta ...
```

¿Qué será en resultado final? ...

Caos



Esta vez con transacciones ...



Banco de Chilly

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;
```

¿Qué será en resultado final?

Se rechazará una transacción.

Garantías de ACID

- **Atomicidad:**
 - La ejecución de cada transacción es atómica:
 - Se realizan todas las acciones o no se realiza ninguna
- **Consistencia:**
 - Cada transacción debe preservar la integridad
 - La base de datos satisfacen 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 persistir los cambios

ACID: Un ejemplo más limpio

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

Usaré restricciones con CHECK porque dan ejemplos más claros pero es importante tener en cuenta que postgres no soporte CHECKs diferibles.

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;
```

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

ACID: 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;
```



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 T1  
  UPDATE Balance  
    SET saldo=saldo-10 (1)  
    WHERE Cuenta=7873698669 ;  
  UPDATE Balance  
    SET total_gasto=total_gasto+100  
    WHERE Cuenta=7873698669 ; (3) ❌  
COMMIT; (4) ROLLBACK;
```

```
START TRANSACTION T2  
  UPDATE Balance  
    SET saldo=saldo+100 (2)  
    WHERE Cuenta=7873698669 ;  
  UPDATE Balance  
    SET total_ingreso=total_ingreso+100  
    WHERE Cuenta=7873698669 ; (5)  
COMMIT; (6)
```

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;
```



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 falla y toda la evidencia de los cambios está en memoria principal, el sistema de base de datos **olvidará los cambios silenciosamente**.)

Entonces con las garantías de ACID ...



... todo está tranquilo.

LA PROXIMA VEZ, CONTINUAREMOS CON LA PREGUNTA:
**¿CÓMO IMPLEMENTA UN SISTEMA DE
B. DE D. LAS GARANTÍAS DE ACID?**

Capítulo 8.2+, Database Management Systems,
Ramakrishnan / Gehrke (Third Edition)

Preguntas?

