

CC3201-1

BASES DE DATOS

OTOÑO 2018

Clase 7: Actualizaciones, Restricciones,  
Formas Normales

Aidan Hogan

[aidhog@gmail.com](mailto:aidhog@gmail.com)

# Las preguntas de hoy

| Planeta  |       |       |      |          |         |      |        |
|----------|-------|-------|------|----------|---------|------|--------|
| nombre   | dist  | radio | grav | días     | años    | temp | anillo |
| Mercurio | 0,39  | 0,38  | 2,8  | 58,646   | 0,241   | 440  | false  |
| Venus    | 0,72  | 0,95  | 8,9  | -243,019 | 0,615   | 730  | false  |
| Tierra   | 1,00  | 1,00  | 9,8  | 0,997    | 1,000   | 288  | false  |
| Marte    | 1,52  | 0,53  | 3,7  | 1,026    | 1,880   | 186  | false  |
| Júpiter  | 5,20  | 10,97 | 22,9 | 0,414    | 11,862  | 152  | true   |
| Saturno  | 9,54  | 9,14  | 9,1  | 0,444    | 29,447  | 134  | true   |
| Urano    | 19,19 | 3,98  | 7,8  | -0,719   | 84,017  | 76   | true   |
| Neptuno  | 30,07 | 3,86  | 11,0 | 0,671    | 164,791 | 53   | true   |

| Satélite  |         |                    |      |
|-----------|---------|--------------------|------|
| nombre    | planeta | descubridor        | año  |
| Luna      | Tierra  | ⊥                  | ⊥    |
| Ganímedes | Júpiter | Galileo Galilei    | 1610 |
| Calisto   | Júpiter | Galileo Galilei    | 1610 |
| Europa    | Júpiter | Galileo Galilei    | 1610 |
| Ío        | Júpiter | Galileo Galilei    | 1610 |
| Titán     | Saturno | Christiaan Huygens | 1655 |
| Tritón    | Neptuno | William Lassell    | 1846 |

| Aterrizaje    |          |      |      |
|---------------|----------|------|------|
| nave          | planeta  | país | año  |
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |
| Viking 1      | Marte    | EEUU | 1976 |
| Beagle 2      | Marte    | ESA  | 2003 |
| Galileo       | Júpiter  | EEUU | 2003 |

*¿Pero cómo se puede crear y actualizar las tablas?*

*¿Y cómo se puede saber si es un buen diseño relacional o no?*

# SQL: GESTIONAR Y CREAR TABLAS

# SQL: Esquema

SistemaSolar

```
CREATE SCHEMA SistemaSolar; -- crear una agrupación de tablas
```

*¿Para que sirven los esquemas?*

Podemos configurar agrupaciones de tablas usando esquemas ...

# SQL: Privilegios de Esquema

SistemaSolar



```
CREATE SCHEMA SistemaSolar; -- crear una agrupación de tablas  
GRANT USAGE ON SCHEMA SistemaSolar TO narmstrong; -- sólo lectura  
GRANT ALL PRIVILEGES ON SCHEMA SistemaSolar TO csagan; -- todo
```

# SQL: Crear tablas

SistemaSolar

Aterrizaje

| nave | planeta | país | año |
|------|---------|------|-----|
|------|---------|------|-----|

```
CREATE SCHEMA SistemaSolar; -- crear una agrupación de tablas
CREATE TABLE SistemaSolar.Aterrizaje (
  nave VARCHAR (255),
  planeta VARCHAR (255),
  país VARCHAR (255),
  año SMALLINT
);
```

# SQL: Borrar tablas

SistemaSolar

```
CREATE SCHEMA SistemaSolar; -- crear una agrupación de tablas
CREATE TABLE SistemaSolar.Aterrizaje (
  nave VARCHAR (255),
  planeta VARCHAR (255),
  país VARCHAR (255),
  año SMALLINT
);
DROP TABLE SistemaSolar.Aterrizaje;
```

*¿Hay que poner el esquema cada vez?*

...

# SQL: Camino de esquema

SistemaSolar

```
CREATE SCHEMA SistemaSolar; -- crear una agrupación de tablas
SHOW search_path; -- public
ALTER USER csagan SET search_path TO SistemaSolar,public;
CREATE TABLE Aterrizaje ( ... );
DROP TABLE Aterrizaje;
```

*Seleccionará el primer esquema en el camino.  
P. ej., si hay SistemaSolar.Aterrizaje y public.Aterrizaje,  
leerá de la primera tabla*



# SQL: ACTUALIZAR TABLAS

# SQL: Insertar tuplas

SistemaSolar

## Aterrizaje

| nave      | planeta  | país | año  |
|-----------|----------|------|------|
| Messenger | Mercurio | EEUU | 2015 |

...

```
CREATE TABLE Aterrizaje ( ... );
```

```
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);
```

# SQL: Insertar tuplas

SistemaSolar

## Aterrizaje

| nave      | planeta  | país | año  |
|-----------|----------|------|------|
| Messenger | Mercurio | EEUU | 2015 |
| Venera 3  | Venus    | URRS | 1966 |

...

```
CREATE TABLE Aterrizaje ( ... );
```

```
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);
```

```
INSERT INTO Aterrizaje VALUES ('Venera 3','Venus','URRS',1966);
```

# SQL: Insertar tuplas

SistemaSolar

## Aterrizaje

| nave      | planeta  | país | año  |
|-----------|----------|------|------|
| Messenger | Mercurio | EEUU | 2015 |
| Venera 3  | Venus    | URRS | 1966 |
| Pioneer   | Venus    | EEUU | 1978 |

...

```
CREATE TABLE Aterrizaje ( ... );  
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);  
INSERT INTO Aterrizaje VALUES ('Venera 3','Venus','URRS',1966);  
INSERT INTO Aterrizaje VALUES ('Pioneer','Venus','EEUU',1978);
```

# SQL: Insertar tuplas

SistemaSolar

## Aterrizaje

| nave      | planeta  | país | año  |
|-----------|----------|------|------|
| Messenger | Mercurio | EEUU | 2015 |
| Venera 3  | Venus    | URRS | 1966 |
| Pioneer   | Venus    | EEUU | 1978 |

...

```
CREATE TABLE Aterrizaje ( ... );  
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);  
INSERT INTO Aterrizaje VALUES ('Venera 3','Venus','URRS',1966);  
INSERT INTO Aterrizaje VALUES ('Pioneer','Venus','EEUU',1978);  
INSERT INTO Aterrizaje VALUES ('Mars 2 lander','Marte','URRS','1971'); -- error
```

# SQL: Insertar tuplas

SistemaSolar

## Aterrizaje

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | EEUU | ⊥    |

...

```
CREATE TABLE Aterrizaje ( ... );  
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);  
INSERT INTO Aterrizaje VALUES ('Venera 3','Venus','URRS',1966);  
INSERT INTO Aterrizaje VALUES ('Pioneer','Venus','EEUU',1978);  
INSERT INTO Aterrizaje VALUES ('Mars 2 lander','Marte','URRS','1971'); -- error  
INSERT INTO Aterrizaje (país,nave,planeta) VALUES ('EEUU','Mars 2 lander','Marte');
```

# SQL: Insertar tuplas

## SistemaSolar

AterrizajeEEUU

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | EEUU | ⊥    |

Aterrizaje

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | EEUU | ⊥    |

```
...
CREATE TABLE Aterrizaje ( ... );
INSERT INTO Aterrizaje VALUES ('Messenger','Mercurio','EEUU',2015);
INSERT INTO Aterrizaje VALUES ('Venera 3','Venus','URRS',1966);
INSERT INTO Aterrizaje VALUES ('Pioneer','Venus','EEUU',1978);
INSERT INTO Aterrizaje VALUES ('Mars 2 lander','Marte','URRS','1971'); -- error
INSERT INTO Aterrizaje (país,nave,planeta) VALUES ('EEUU','Mars 2 lander','Marte');
CREATE TABLE AterrizajeEEUU ( ... ); -- misma definición que Aterrizaje
INSERT INTO AterrizajeEEUU ( SELECT * FROM Aterrizaje WHERE país='EEUU' );
```

# SQL: Editar tuplas

## SistemaSolar

AterrizajeEEUU

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | EEUU | ⊥    |

Aterrizaje

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |

...

```
UPDATE Aterrizaje SET año=1971,país=URRS WHERE nave='Mars 2 lander';
```



# SQL: Borrar tuplas

## SistemaSolar

**AterrizajeEEUU**

| nave      | planeta  | país | año  |
|-----------|----------|------|------|
| Messenger | Mercurio | EEUU | 2015 |
| Pioneer   | Venus    | EEUU | 1978 |

**Aterrizaje**

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |

...

```
DELETE FROM AterrizajeEEUU WHERE año IS NULL;
```

# SQL: Borrar columnas

## SistemaSolar

AterrizajeEEUU

| nave      | planeta  | año  |
|-----------|----------|------|
| Messenger | Mercurio | 2015 |
| Pioneer   | Venus    | 1978 |

Aterrizaje

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |

...

```
ALTER TABLE AterrizajeEEUU DROP COLUMN país;
```

# SQL: Crear columnas

## SistemaSolar

**AterrizajeEEUU**

| nave      | planeta  | año  | despegue |
|-----------|----------|------|----------|
| Messenger | Mercurio | 2015 | ⊥        |
| Pioneer   | Venus    | 1978 | ⊥        |

**Aterrizaje**

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |

...

```
ALTER TABLE AterrizajeEEUU ADD COLUMN despegue DATE;
```

# SQL: Modificar columnas

## SistemaSolar

**AterrizajeEEUU**

| nave      | planeta  | año  | despegue |
|-----------|----------|------|----------|
| Messenger | Mercurio | 2015 | ⊥        |
| Pioneer   | Venus    | 1978 | ⊥        |

**Aterrizaje**

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |

...

```
ALTER TABLE AterrizajeEEUU ALTER COLUMN despegue VARCHAR(255);
```

# Postgres: Cargar datos

## SistemaSolar

### AterrizajeEEUU

| nave      | planeta  | año  | despegue |
|-----------|----------|------|----------|
| Messenger | Mercurio | 2015 | ⊥        |
| Pioneer   | Venus    | 1978 | ⊥        |

### Aterrizaje

| nave          | planeta  | país | año  |
|---------------|----------|------|------|
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Messenger     | Mercurio | EEUU | 2015 |
| Venera 3      | Venus    | URRS | 1966 |
| Pioneer       | Venus    | EEUU | 1978 |
| Mars 2 lander | Marte    | URRS | 1971 |
| Viking 1      | Marte    | EEUU | 1976 |
| Beagle 2      | Marte    | ESA  | 2003 |
| Galileo       | Júpiter  | EEUU | 2003 |

...

```
COPY Aterrizaje FROM '/home/csagan/aterrizaje.tsv' DELIMITER E'\t';
```

¿Algún problema aquí?

...

*Específico de Postgres*

*Concatena los datos*

*(Integrity Constraints)*

# SQL: RESTRICCIONES

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           |

### 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  |



# 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

# Restricciones básicas: llaves, nulos, domino

| Cuenta     |              |            |           |           |
|------------|--------------|------------|-----------|-----------|
| número     | 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     |              |            |           |           |
|------------|--------------|------------|-----------|-----------|
| número     | 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     |              |            |           |           |
|------------|--------------|------------|-----------|-----------|
| número     | 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     |              |            |           |           |
|------------|--------------|------------|-----------|-----------|
| número     | 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.*

# Restricciones de llaves foráneas

## 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 de llaves compuestas

| Divisa    |           |             |
|-----------|-----------|-------------|
| <u>d1</u> | <u>d2</u> | valor       |
| CLP       | USD       | 0,0001533   |
| USD       | CLP       | 652,2750000 |
| CLP       | EUR       | 0,0001498   |
| EUR       | CLP       | 735,9700000 |

| Cambio       |                |                |           |
|--------------|----------------|----------------|-----------|
| <u>id</u>    | <u>D.venta</u> | <u>D.compa</u> | monto     |
| CA0121312393 | CLP            | USD            | 100000,00 |
| CA0121312393 | CLP            | EUR            | 20000,00  |
| CA2134812341 | EUR            | CLP            | 4815,16   |

Banco de Chile

```
CREATE TABLE Divisa (  
  d1 VARCHAR (3),  
  d2 VARCHAR (3),  
  valor DOUBLE PRECISION,  
  PRIMARY KEY (d1, d2)  
)
```

```
CREATE TABLE Cambio (  
  id VARCHAR (12),  
  venta VARCHAR (3),  
  compra VARCHAR (3),  
  monto DOUBLE PRECISION,  
  FOREIGN KEY (venta, compra) REFERENCES Divisa (d1, d2),  
  PRIMARY KEY (id, venta, compra)  
)
```

# Restricciones sobre varias columnas

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

## Divisa

| <u>d1</u> | <u>d2</u> | valor       |
|-----------|-----------|-------------|
| 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)
```



# Restricciones sobre varias tablas (!)

## 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 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 ...*

# Asertos: Restricciones independientes

## 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 |

```
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 |



# WARNING

Postgres no permite consultas anidadas en CHECK  
ni asertos

(son caros!)

DEFINIR DOMINIOS Y TIPOS

# Crear dominios: VARCHAR

## 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 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> | 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 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> | 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 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 domino con otros valores como fuera del tipo base*

# Tipos: son distintos a otros tipos

## 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 |

## Cuenta

| <u>número</u> | rut          | tipo       | saldo_clp | saldo_usd |
|---------------|--------------|------------|-----------|-----------|
| 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 pueden comparar valores del nuevo tipo con valores de otros tipos (solo entre sí).*

# Tipos: son distintos a otros tipos

## 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 |

## Cuenta

| <u>número</u> | rut          | tipo       | saldo_clp | saldo_usd |
|---------------|--------------|------------|-----------|-----------|
| 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 pueden usar funciones del tipo base  
con valores del nuevo tipo.*



# WARNING

Tipos son estándares (en SQL)

Pero Postgres solo soporte tipos compuestos ...

# Tipos: un tipo compuesto

| Cuenta     |              |            |                          |
|------------|--------------|------------|--------------------------|
| número     | rut          | tipo       | saldo                    |
| 7873698669 | 32.000.273-K | Estacional | (clp:225000, usd:344,94) |

Banco de Chile

```
CREATE TYPE valor AS (  
  clp    BIGINT,  
  usd    DOUBLE PRECISION  
);  
CREATE TABLE Cuenta AS (  
  número ... ,  
  saldo valor  
);  
INSERT INTO Cuenta VALUES (7873698669, '32.000.273-K', 'Estacional', ROW(225000, 344.94));
```

# Tipos: un tipo compuesto

| Cuenta     |              |            |                          |
|------------|--------------|------------|--------------------------|
| número     | rut          | tipo       | saldo                    |
| 7873698669 | 32.000.273-K | Estacional | (clp:225000, usd:344,94) |

Banco de Chile

...  
`SELECT saldo.usd FROM Cuenta;`

| Cuenta    |
|-----------|
| saldo.usd |
| 344,49    |

# FORMAS NORMALES

# Todo bien

## 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 |



*¿Pero si un cliente puede tener varios números de teléfono?*

...



# UNF: Forma No Normalizada *(UnNormalised Form)*

## Cliente

| <u>rut</u>   | nombre  | fono                      | dirección                       |
|--------------|---------|---------------------------|---------------------------------|
| 32.000.273-K | Kelvin  | +56976698463              | Campo de Hielo Sur, Depto 273   |
| 12.491.671-K | Rankine | +56991324842,+56223491234 | Campo de Hielo Norte, Depto 502 |

## UNF:

*Varias multiplicidades de valores en una columna de la tabla*



Banco de Chile

# 1NF: Primera Forma Normal *(First Normal Form)*

## Cliente

| <u>rut</u>   | nombre  | <u>fono</u>  | <u>dirección</u>                |
|--------------|---------|--------------|---------------------------------|
| 32.000.273-K | Kelvin  | +56976698463 | Campo de Hielo Sur, Depto 273   |
| 12.491.671-K | Rankine | +56991324842 | Campo de Hielo Norte, Depto 502 |
| 12.491.671-K | Rankine | +56223491234 | Campo de Hielo Norte, Depto 502 |

**1NF:**

*Un valor en cada celda de la tabla*



Banco de Chile

# ¿Todo bien con sola la 1NF?

## 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   |
| 12.491.671-K | Rankine       | +56991324842 | Campo de Hielo Norte, Depto 502 |
| 12.491.671-K | Rankine       | +56223491234 | Campo de Hielo Norte, Depto 502 |

*¿Algún problema aquí? ...*



Banco de Chile

# ¿Todo bien con sola la 1NF? No

## 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   |
| 12.491.671-K | Rankine       | +56991324842 | Campo de Hielo Norte, Depto 502 |
| 12.491.671-K | Rankine       | +56223491234 | Campo de Hielo Norte, Depto 502 |

Soluciones con nulos  
no cuentan aquí.



*¿Algún problema aquí? ...*

**Redundancia**

*¿Pero por qué es un problema? ¿Sólo espacio? ...*

**Anomalía de actualización:**

*Por ejemplo, se puede actualizar la dirección de Rankine en un lugar sin actualizar todos los valores*

**Anomalía de inserción:**

*No podemos insertar un nuevo cliente a la tabla hasta tengamos un número de teléfono*

**Anomalía de borrado:**

*Si el número de teléfono ahora está invalido, tendremos que borrar la fila entera con la dirección, etc.*

# ¿Todo bien con sola la 1NF? No

## Cliente

| <u>rut</u>   | nombre  | <u>fono</u>  | <u>dirección</u>                |
|--------------|---------|--------------|---------------------------------|
| 32.000.273-K | Kelvin  | +56976698463 | Campo de Hielo Sur, Depto 273   |
| 12.491.671-K | Rankine | +56991324842 | Campo de Hielo Norte, Depto 502 |
| 12.491.671-K | Rankine | +56223491234 | Campo de Hielo Norte, Depto 502 |

*¿La solución? ...*

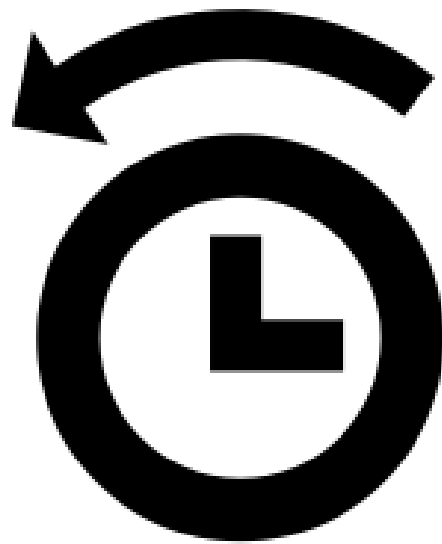
*Crear otra tabla con rut y fono*

*¿Pero cómo podemos definir el problema aquí? ...*

...



Banco de Chile





# Modelo Relacional: Restricciones (Llaves)

Un conjunto de atributos de una relación

forma una llave candidata

si es una súper llave

*y no hay un subconjunto propio de esos atributos*

*que es una súper llave*



# Modelo Relacional: Restricciones (Llaves)

Persona(rut, nombre, fecha-de-nacimiento, madre-rut, padre-rut)

*¿Hay otra llave candidata?*

*Probablemente ...*

Persona(rut, nombre, fecha-de-nacimiento, madre-rut, padre-rut)

*... o puede ser ...*

Persona(rut, nombre, fecha-de-nacimiento, madre-rut, padre-rut)

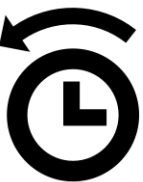
*(si no tenemos un tipo como Gengis Kan)*





# Modelo Relacional: Restricciones (Llaves)

Un atributo es primo  
si está en alguna llave candidata



# Modelo Relacional:

## Restricciones (Dependencias funcionales)

Dado una relación

y dos conjuntos de atributos  $X, Y$

$X$  determina funcionalmente  $Y$

si y solo si

cada valor de  $X$  en la relación

tiene asociado un solo valor de  $Y$



# Modelo Relacional:

## Restricciones (Dependencias funcionales)

### Cervezas

| <u>nombre</u>      | tipo    | grados | ciudad-origen |
|--------------------|---------|--------|---------------|
| Austral Lager      | Lager   | 4,6    | Punta Arenas  |
| Austral Yagan      | Ale     | 5,0    | Punta Arenas  |
| Austral Pale Ale   | Ale     | 5,0    | Punta Arenas  |
| Kuntsmann Torobayo | Ale     | 5,1    | Valdivia      |
| Kross 5            | Ale     | 7,2    | Curacaví      |
| Kross Golden       | Ale     | 5,3    | Curacaví      |
| Kross Pilsner      | Pilsner | 4,9    | Curacaví      |

*¿Hay una **dependencia funcional** aquí?*

$\{\underline{\text{nombre}}\} \rightarrow \{\text{tipo}, \text{grados}, \text{ciudad-origen}\}$

$\{\underline{\text{nombre}}\} \rightarrow \{\text{tipo}, \underline{\text{nombre}}\}$

$\{\text{grados}\} \rightarrow \{\text{grados}\}$

$\{\text{grados}\} \rightarrow \{\text{tipo}, \text{ciudad-origen}\}$

(al menos aquí)

...





# ¿Todo bien con sola la 1NF? No

## Cliente

| <u>rut</u>   | nombre  | <u>fono</u>  | <u>dirección</u>                |
|--------------|---------|--------------|---------------------------------|
| 32.000.273-K | Kelvin  | +56976698463 | Campo de Hielo Sur, Depto 273   |
| 12.491.671-K | Rankine | +56991324842 | Campo de Hielo Norte, Depto 502 |
| 12.491.671-K | Rankine | +56223491234 | Campo de Hielo Norte, Depto 502 |

¿La solución? ...

Crear otra tabla con rut y fono

¿Pero cómo podemos definir el problema aquí? ...

{rut} → {nombre, dirección}

... pero rut es sólo parte de una llave candidata



# 2NF: Segunda Forma Normal *(Second Normal Form)*

## Cliente

| <u>rut</u>   | nombre  | dirección                       |
|--------------|---------|---------------------------------|
| 32.000.273-K | Kelvin  | Campo de Hielo Sur, Depto 273   |
| 12.491.671-K | Rankine | Campo de Hielo Norte, Depto 502 |

## Contacto

| <u>rut</u>   | fono         |
|--------------|--------------|
| 32.000.273-K | +56976698463 |
| 12.491.671-K | +56991324842 |
| 12.491.671-K | +56223491234 |



## 2NF: Satisface 1NF y ...

*No existe:*

$$A \rightarrow \{\mathbf{b}\}$$

*tal que:*

*A sea un subconjunto propio de una llave candidata  
y **b** sea un atributo no primo.*

# 2NF: Segunda Forma Normal

*(Se asume que haya una persona por cuenta)*

## Transferencia

| <u>cuenta_origen</u> | <u>número</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>nombre_destino</u> | <u>monto</u> |
|----------------------|---------------|-----------------------|--------------------|-----------------------|--------------|
| 7873698669           | 0000047       | 1849123812            | 12.491.671-K       | Rankine               | 41920        |
| 7873698669           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 10000        |
| 8273697679           | 0000047       | 7873698669            | 32.000.273-K       | Kelvin                | 529000       |
| 8273697679           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 51920        |

*¿Está en la 2NF? ...*

*¡Sí!*





# ¿Todo bien con sola la 2NF?

*(Se asume que haya una persona por cuenta)*

## Transferencia

| <u>cuenta_origen</u> | <u>número</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>nombre_destino</u> | <u>monto</u> |
|----------------------|---------------|-----------------------|--------------------|-----------------------|--------------|
| 7873698669           | 0000047       | 1849123812            | 12.491.671-K       | Rankine               | 41920        |
| 7873698669           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 10000        |
| 8273697679           | 0000047       | 7873698669            | 32.000.273-K       | Kelvin                | 529000       |
| 8273697679           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 51920        |

*¿Está en la 2NF? ...*

*¡Sí!*

*¿Se pueden tener anomalías? ...*





# ¿Todo bien con sola la 2NF? No

(Se asume que haya una persona por cuenta)

## Transferencia

| <u>cuenta_origen</u> | <u>número</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>nombre_destino</u> | <u>monto</u> |
|----------------------|---------------|-----------------------|--------------------|-----------------------|--------------|
| 7873698669           | 0000047       | 1849123812            | 12.491.671-K       | Rankine               | 41920        |
| 7873698669           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 10000        |
| 8273697679           | 0000047       | 7873698669            | 32.000.273-K       | Kelvin                | 529000       |
| 8273697679           | 0000048       | 1849123812            | 12.491.671-K       | Rankine               | 51920        |

¿Está en la 2NF? ...

¡Sí!

¿Se pueden tener anomalías? ...

¡Sí! Por ejemplo ...

## Transferencia

| ... | <u>cuenta_destina</u> | ... | <u>nombre_destino</u> | <u>monto</u> |
|-----|-----------------------|-----|-----------------------|--------------|
| ... | 1849123812            | ... | Rankine               | 41920        |
| ... | ...                   | ... | ...                   | ...          |
| ... | 1849123812            | ... | Kelvin                | 10000        |

¿Dónde está el problema? ...

{cuenta\_destina} → {rut\_destino, nombre\_destino}

.. pero el atributo cuenta\_destina no es primo



# 3NF: Tercera Forma Normal *(Third Normal Form)*

*(Se asume que haya una persona por cuenta)*

## Transferencia

| <u>cuenta_origen</u> | <u>número</u> | <u>cuenta_destina</u> | <u>monto</u> |
|----------------------|---------------|-----------------------|--------------|
| 7873698669           | 0000047       | 1849123812            | 41920        |
| 7873698669           | 0000048       | 1849123812            | 10000        |
| 8273697679           | 0000047       | 7873698669            | 529000       |
| 8273697679           | 0000048       | 1849123812            | 51920        |

## Destino

| <u>cuenta</u> | <u>rut</u>   | <u>nombre</u> |
|---------------|--------------|---------------|
| 1849123812    | 12.491.671-K | Rankine       |
| 7873698669    | 32.000.273-K | Kelvin        |



**3NF: Satisface la 2NF y ...**

*No existe:*

$$X \rightarrow Y \rightarrow \{ z \}$$

*tal que :*

*z sea no primo*

$$y \ Y \nrightarrow X \ y \ z \notin Y$$

# ¿Todo bien con sola la 3NF?

(Se asume que haya una persona por cuenta)

## TransferenciaTotal

| <u>cuenta_origen</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>total</u> |
|----------------------|-----------------------|--------------------|--------------|
| 7873698669           | 1849123812            | 12.491.671-K       | 51920        |
| 8273697679           | 7873698669            | 32.000.273-K       | 529000       |
| 8273697679           | 1849123812            | 12.491.671-K       | 51920        |

¿Está en la 3NF? ...

*¡No! Porque existe*

$\{ \text{cuenta\_origen}, \text{cuenta\_destina} \} \rightarrow \{ \text{cuenta\_destina} \} \rightarrow \{ \text{rut\_destino} \}$   
donde  $\{ \text{cuenta\_destina} \} \nrightarrow \{ \text{cuenta\_origen}, \text{cuenta\_destina} \}$  y  
 $\text{rut\_destino} \notin \{ \text{cuenta\_destina} \}$  y  $\text{rut\_destino}$  no es primo



# ¿Todo bien con sola la 3NF?

(Se asume que haya una persona por cuenta  
y haya una cuenta por persona)

## TransferenciaTotal

| <u>cuenta_origen</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>total</u> |
|----------------------|-----------------------|--------------------|--------------|
| 7873698669           | 1849123812            | 12.491.671-K       | 51920        |
| 8273697679           | 7873698669            | 32.000.273-K       | 529000       |
| 8273697679           | 1849123812            | 12.491.671-K       | 51920        |

¿Está en la 3NF ahora? ...

¡Sí!

{cuenta\_origen, cuenta\_destina} → {cuenta\_destina} → { rut\_destino }  
no cuenta ahora porque rut\_destino es un atributo primo  
(dado que {cuenta\_origen, rut\_destino} es una llave candidata)

¿Hay una posibilidad de anomalías? ...





# ¿Todo bien con sola la 3NF? No

(Se asume que haya una persona por cuenta  
y haya una cuenta por persona)

## TransferenciaTotal

| <u>cuenta_origen</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>total</u> |
|----------------------|-----------------------|--------------------|--------------|
| 7873698669           | 1849123812            | 12.491.671-K       | 51920        |
| 8273697679           | 7873698669            | 32.000.273-K       | 529000       |
| 8273697679           | 1849123812            | 12.491.671-K       | 51920        |

¿Está en la 3NF ahora? ...

¡Sí!

{cuenta\_origen, cuenta\_destina} → {cuenta\_destina} → {rut\_destino}  
no cuenta ahora porque rut\_destino es un atributo primo  
dado que {cuenta\_origen, rut\_destino} es una llave candidata

¿Hay una posibilidad de anomalías? ...

¡Sí! Por ejemplo ...

## TransferenciaTotal

| <u>cuenta_origen</u> | <u>cuenta_destina</u> | <u>rut_destino</u> | <u>total</u> |
|----------------------|-----------------------|--------------------|--------------|
| 7873698669           | 1849123812            | 12.491.671-K       | 51920        |
| ...                  | ...                   | ...                | ...          |
| 8273697679           | 1849123812            | 24.482.054-9       | 51920        |



# BCNF: Forma Normal de Boyce–Codd

(Se asume que haya una persona por cuenta  
y haya una cuenta por persona)

## TransferenciaTotal

| <u>cuenta_origen</u> | <u>cuenta_destina</u> | total  |
|----------------------|-----------------------|--------|
| 7873698669           | 1849123812            | 51920  |
| 8273697679           | 7873698669            | 529000 |
| 8273697679           | 1849123812            | 41920  |

## Cuenta

| <u>cuenta</u> | rut          |
|---------------|--------------|
| 1849123812    | 12.491.671-K |
| 7873698669    | 32.000.273-K |



**BCNF:** Satisface la 1NF y ...

Para cada:

$$X \rightarrow Y$$

$X$  es una súper llave

$$\text{o } Y \subseteq X$$

# ¿Todo bien con la BCNF?

(Se asume que un cliente tenga un ejecutivo por banco y dos bancos no puedan tener el mismo ejecutivo)

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>banco</u>   | <u>ejecutivo_rut</u> |
|--------------------|----------------|----------------------|
| 17.123.031-K       | Santander      | 10.213.412-7         |
| 17.123.031-K       | Banco de Chile | 14.123.521-9         |
| 17.123.031-K       | BICE           | 15.823.914-6         |
| 19.423.923-9       | Santander      | 10.213.412-7         |
| 19.423.923-9       | Itaú           | 16.124.123-8         |

¿Está en la BCNF? ...

*¡No! Porque existe*

$\{ \text{ejecutivo\_rut} \} \rightarrow \{ \text{banco} \}$

*... donde  $\{ \text{ejecutivo\_rut} \}$  no es una súper llave y*

$\{ \text{banco} \} \not\subseteq \{ \text{ejecutivo\_rut} \}$

¿Un ejemplo de una anomalía?

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>banco</u> | <u>ejecutivo_rut</u> |
|--------------------|--------------|----------------------|
| 17.123.031-K       | Santander    | 10.213.412-7         |
| ...                | ...          | ...                  |
| 19.423.923-9       | BICE         | 10.213.412-7         |



# ¿Todo bien con la BCNF?

*(Se asume que un cliente tenga un ejecutivo por banco y dos bancos no puedan tener el mismo ejecutivo)*

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>banco</u>   | <u>ejecutivo_rut</u> |
|--------------------|----------------|----------------------|
| 17.123.031-K       | Santander      | 10.213.412-7         |
| 17.123.031-K       | Banco de Chile | 14.123.521-9         |
| 17.123.031-K       | BICE           | 15.823.914-6         |
| 19.423.923-9       | Santander      | 10.213.412-7         |
| 19.423.923-9       | Itaú           | 16.124.123-8         |

*¿Cómo podemos rediseñar el modelo para obtener la BCNF?*





# ¿Todo bien con la BCNF?

*(Se asume que un cliente tenga un ejecutivo por banco y dos bancos no puedan tener el mismo ejecutivo)*

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>ejecutivo_rut</u> |
|--------------------|----------------------|
| 17.123.031-K       | 10.213.412-7         |
| 17.123.031-K       | 14.123.521-9         |
| 17.123.031-K       | 15.823.914-6         |
| 19.423.923-9       | 10.213.412-7         |
| 19.423.923-9       | 16.124.123-8         |

## BancoDeEjecutivo

| <u>ejecutivo_rut</u> | <u>banco</u>   |
|----------------------|----------------|
| 10.213.412-7         | Santander      |
| 14.123.521-9         | Banco de Chile |
| 15.823.914-6         | BICE           |
| 11.435.213-4         | Santander      |
| 16.124.123-8         | Itaú           |

*¿Cómo podemos rediseñar el modelo para obtener la BCNF?*

***Dos tablas: EjecutivoDeCliente y BancoDeEjecutivo***

*¿Algún problema aquí? ...*

**Todavía hay anomalías posibles ...**

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>ejecutivo_rut</u> |
|--------------------|----------------------|
| 17.123.031-K       | 10.213.412-7         |
| 17.123.031-K       | 14.123.521-9         |
| 17.123.031-K       | 15.823.914-6         |
| 19.423.923-9       | 10.213.412-7         |
| 19.423.923-9       | 16.124.123-8         |

## BancoDeEjecutivo

| <u>ejecutivo_rut</u> | <u>banco</u> |
|----------------------|--------------|
| 10.213.412-7         | Santander    |
| 14.123.521-9         | Santander    |
| 15.823.914-6         | BICE         |
| 11.435.213-4         | Santander    |
| 16.124.123-8         | Itaú         |



# ¿Todo bien con la BCNF?

*(Se asume que un cliente tenga un ejecutivo por banco y dos bancos no puedan tener el mismo ejecutivo)*

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>ejecutivo_rut</u> |
|--------------------|----------------------|
| 17.123.031-K       | 10.213.412-7         |
| 17.123.031-K       | 14.123.521-9         |
| 17.123.031-K       | 15.823.914-6         |
| 19.423.923-9       | 10.213.412-7         |
| 19.423.923-9       | 16.124.123-8         |

## BancoDeEjecutivo

| <u>ejecutivo_rut</u> | <u>banco</u>   |
|----------------------|----------------|
| 10.213.412-7         | Santander      |
| 14.123.521-9         | Banco de Chile |
| 15.823.914-6         | BICE           |
| 11.435.213-4         | Santander      |
| 16.124.123-8         | Itaú           |

*¿Cómo podemos rediseñar el modelo para obtener la BCNF?*

***Dos tablas: EjecutivoDeCliente y BancoDeEjecutivo***

*¿Una alternativa? ...*

**Usar una llave foránea con la 3NF (la BCNF no siempre es mejor)**

## EjecutivoDeCliente

| <u>cliente_rut</u> | <u>B.banco</u> | <u>B.ejecutivo_rut</u> |
|--------------------|----------------|------------------------|
| 17.123.031-K       | Santander      | 10.213.412-7           |
| 17.123.031-K       | B. de Chile    | 14.123.521-9           |
| 17.123.031-K       | BICE           | 15.823.914-6           |
| 19.423.923-9       | Santander      | 10.213.412-7           |
| 19.423.923-9       | Itaú           | 16.124.123-8           |

## BancoDeEjecutivo

| <u>ejecutivo_rut</u> | <u>banco</u> |
|----------------------|--------------|
| 10.213.412-7         | Santander    |
| 14.123.521-9         | B. de Chile  |
| 15.823.914-6         | BICE         |
| 11.435.213-4         | Santander    |
| 16.124.123-8         | Itaú         |



# ¿Todo bien con la BCNF?

*¿Hay formas normales más fuertes que BCNF? ...*



**BCNF:** Satisface la 1NF y ...

*Para cada:*

$$X \rightarrow Y$$

*$X$  es una súper llave*

$$\text{o } Y \subseteq X$$

# ¿Todo bien con la BCNF?

*¿Hay formas normales más fuertes que la BCNF? ...*

***¡Sí, hay más formas normales! La 4NF, 5NF, 6NF, ...***  
*Pero las anomalías posibles son más y más raras ...*  
*... entonces pararemos aquí con BCNF. 😊*



**BCNF:** Satisface la 1NF y ...

*Para cada:*

$$X \rightarrow Y$$

*X es una súper llave*

$$\text{o } Y \subseteq X$$



# Formas Normales

*¿Qué piensan ustedes?*

*¿Son útiles las formas normales?*

*¿Siempre hay que obtener (al menos) la BCNF?*



**BCNF:** Satisface la 1NF y ...

*Para cada:*

$$X \rightarrow Y$$

*$X$  es una súper llave*

$$\text{o } Y \subseteq X$$

# A veces, los nulos no son tan malos

| Satélite  |         |                    |      |
|-----------|---------|--------------------|------|
| nombre    | planeta | descubridor        | año  |
| Luna      | Tierra  | ⊥                  | ⊥    |
| Ganímedes | Júpiter | Galileo Galilei    | 1610 |
| Calisto   | Júpiter | Galileo Galilei    | 1610 |
| Europa    | Júpiter | Galileo Galilei    | 1610 |
| Ío        | Júpiter | Galileo Galilei    | 1610 |
| Titán     | Saturno | Christiaan Huygens | 1655 |
| Tritón    | Neptuno | William Lassell    | 1846 |

# Calendario

- Lab: 17 de abril (gestionar tablas)
- Proyecto: 23 de abril
- Proyecto: 25 de abril
- Clase: 30 de abril (optimización con Sebastián)
- Lab: 2 de mayo (optimización)
- ...



Preguntas?

