

CC3201-1

BASES DE DATOS

OTOÑO 2017

**Clase 6: Actualizaciones, Restricciones,
Formas Normales**

Aidan Hogan

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 VALUES ( 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

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

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

```
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 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

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



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

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 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

<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

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 puede usar funciones del tipo base
con valores del nuevo tipo.*



WARNING

Tipos son estándares (en SQL)

Pero Postgres solo soporte tipos “complejos”

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

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

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? ...

Redundancia



Banco de Chile

1NF: Primera Forma Normal (First Normal Form)

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? ...

Anomalía de actualización:

Si quiero actualizar la dirección de Rankine, tendré que actualizar varias tuplas (y la base de datos será inconsistente entre las actualizaciones)



Banco de Chile

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

Ciente

<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? ...

Anomalía de inserción:

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

Sí, se puede usar un valor nulo o un valor vacío, pero este tipo de solución no cuenta aquí ...

si es la única solución, todavía será considerado una anomalía



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

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? ...

Anomalía de borrado:

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

Sí, se puede reemplazar el valor con un nulo o un valor vacío, pero este tipo de solución no cuenta aquí ... si es la única solución, todavía será considerado una anomalía



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

¿La solución? ...

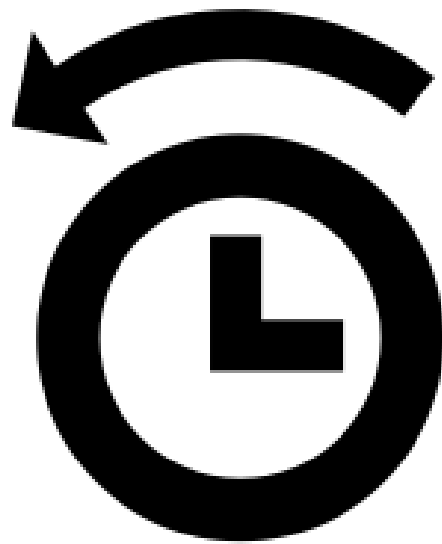
Crear otra tabla con rut y fono

¿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

nombre	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)

...





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

Ciente

<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? ...

$\{\underline{\text{rut}}\} \rightarrow \{\text{nombre}, \text{dirección}\}$

... 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 *(Second Normal Form)*

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 2NF? ...

¡Sí!

¿Se puede tener anomalías? ...

¡Sí!

(p.ej. podríamos tener varios **nombre_destinos** para el mismo **rut_destino**)

¿Dónde está el problema? ...

{cuenta_destino} → {rut_destino, nombre_destino}

Pero **cuenta_destino** no es primo



3NF: Tercera Forma Normal (Third Normal Form)

(Se asume que haya una persona por cuenta destino)

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 2NF y ...

No existe:

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

tal que :

z sea no primo

$$y \ Y \not\rightarrow X \ y \ z \notin Y$$

3NF: Tercera Forma Normal (Third Normal Form)

(Se asume que cada cuenta tenga sólo una persona)

Transferencia Total

<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 3NF? ...

¡Sí!

¿Por qué? ...

$\{ \text{cuenta_origen}, \text{cuenta_destina} \} \rightarrow \{ \text{cuenta_destina} \} \rightarrow \{ \text{rut_destino} \}$

*... pero **rut_destino** es un atributo primo ...*

$\{ \text{cuenta_origen}, \text{rut_destino} \}$ es una llave candidata

¿Hay una posibilidad de anomalías? ...

¡Sí! La misma cuenta destina con dos ruts diferentes.



BCNF: Forma Normal de Boyce–Codd

(Se asume que cada cuenta tenga sólo una persona)

TransferenciaTotal

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

Destino

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



BCNF: Satisface 1NF y ...

Para cada:

$$X \rightarrow Y$$

X es una súper llave

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

Otras Formas Normales

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

***¡Sí!** 4NF, 5NF, 6NF, ...*

*Pero las anomalías posibles son más y más raras ...
... entonces pararemos aquí con BCNF. 😊*



BCNF: Satisface 1NF y ...

Para cada:

$X \rightarrow Y$

X es una súper llave

o $Y \subseteq X$

Otras Formas Normales

¿Qué piensan ustedes?
¿Son útiles las formas normales?
¿Siempre hay que tener (al menos) BCNF?



BCNF: Satisface 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

LA PROXIMA VEZ, CONTINUAREMOS CON:
OPTIMIZACIÓN DE CONSULTAS

Calendario

- Censo: 19 de abril
- Auxiliar: 21 de abril (SQL)
- Clase: 24 de abril (optimización con Sebastián)
- Lab: 26 de abril (optimización)
- Auxiliar: 28 de abril (preparación pa'l control)
- Feriado: 1 de mayo
- Lab: 3 de mayo (programación)
- Control: 5 de mayo

Preguntas?

