

CC3201

**BASES DE DATOS
PRIMAVERA 2025**

Clase 9: SQL III

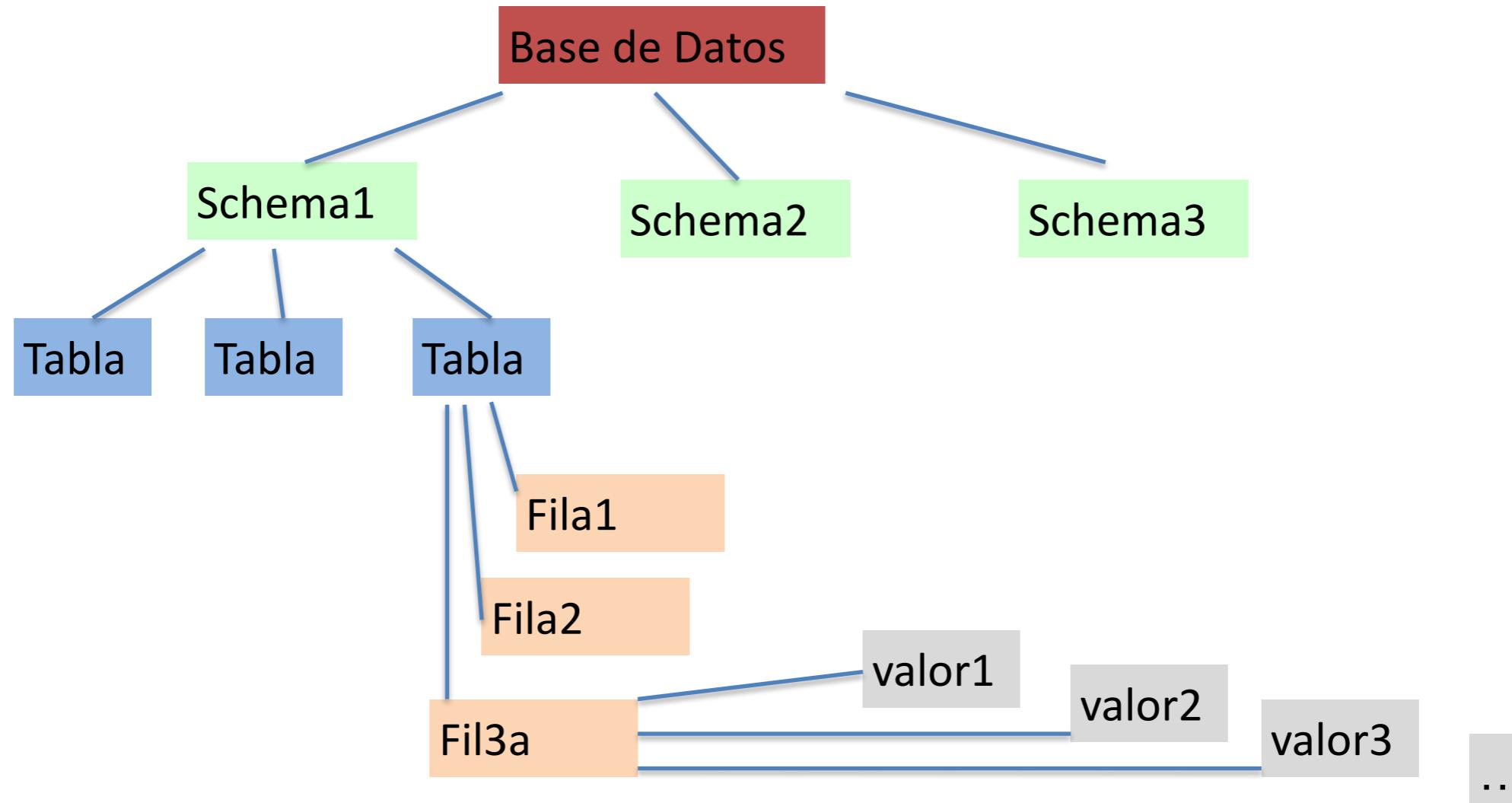
{Claudio Gutierrez, Cristián Salazar,
Eduardo Godoy}

SQL: GESTIONAR Y CREAR TABLAS

Láminas de Aidan Hogan

Capítulo 3.1.1 | Ramakrishnan / Gehrke

SQL: Jerarquías de estructura de una BD



Una base de datos es un conjunto de esquemas

Un esquema es un conjunto de 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: 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: ACTUALIZAR TABLAS

Capítulo 3.1.1 | Ramakrishnan / Gehrke

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: Actualizar tuplas de otra tabla

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	1971

...

```
UPDATE Aterrizaje A
SET país=AE.país
FROM AterrizajeEEUU AE
WHERE A.nave=AE.nave AND A.planeta=AE.planeta;
```

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	EEUU	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	EEUU	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	EEUU	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	EEUU	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	EEUU	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';
```

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

cuenta	comentario	fecha	hora	monto	saldo	<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

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

Cuenta

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

Cliente

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

Divisa

d1	d2	valor
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

Restricciones básicas: llaves, nulos, domino

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 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>	<u>rut</u>	<u>tipo</u>	<u>saldo_clp</u>	<u>saldo_usd</u>
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

```
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 intuitivo si las restricciones tienen
nombres intuitivos.*

Restricciones de llaves foráneas

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

Cuenta

número	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>	<u>valor</u>
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>	<u>monto</u>
CA0121312393	CLP	USD	100000,00
CA0121312393	CLP	EUR	20000,00
CA2134812341	EUR	CLP	4815,16

```
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 más complejas

¡Use con cautela! ¡trate de no usarlas!