

---

---

# Bases de datos clase 18: Insert y update

Leonardo Bravo Illanes  
Escuela de Informática y Telecomunicaciones  
Universidad Diego Portales

---

---

# Noticias

- Actualización para el Avance 2 del proyecto y la Evaluación de este
- Querys SQL de las interacciones (No insert , no update, no delete)
  - Modelo de datos en FN1, FN2 y FN3
- << Entrega el 27 de Mayo >>

## Proyecto

---

- Pauta de Evaluación [ [ODP](#) | [PDF](#) ] - Actualizado el 18-may-2022
- Pre-Evaluación - Desde el 14 Marzo [Link](#) (Entrar con el correo de la UDP)
- Avance 1 - 8 de Abril [ [ODP](#) | [PDF](#) ]
- Avance 2 - Semana del 2 de Mayo [ [ODP](#) | [PDF](#) ] - Actualizado el 18-may-2022
- Presentación Final - Semana del 30 de Mayo [ [ODP](#) | [PDF](#) ]

# Noticias

## A mitad de Camino

Semana	Observaciones	Tarea individual, etc)	Evaluaciones grupales (ej: entrega informe grupal, etc.)	Laboratorios (Entregas calificadas)
7 marzo	Introducción y características de las BDD			
14 marzo	Modelamiento E-R Parte 1 y 2			
21 marzo	Modelamiento E-R Parte 3 y Ejercicio con Draw.io			
28 marzo	Algebra Relacional Parte 1 y 2			
4 abril	Algebra Relacional Parte 3 y SQL		Entrega 1 Proyecto	
11 abril	SQL parte 2 – Viernes 15 feriado			
18 abril	SQL parte 3 – Repaso Solemne			
25 abril	Semana de Solemnes	Solemne 1		
2 mayo	5-6 Receso			
9 mayo	Normalización parte 1 y 2			
16 mayo	Normalización parte 3 y Ejercicios de Normalización			
23 mayo	Aspectos de diseño de BDD 1 y Confidencialidad		Entrega 2 Proyecto	Entrega Lab 1
30 mayo	Presentación de Proyectos			
6 junio	Optimización de Consultas 1 y 2		Presentación Proyecto	Entrega Lab 2
13 junio	Ejercicios pre-solemne2 (1 y 2)			
20 junio	Martes 21 feriado – Semana de Solemnes 2	Solemne 2		
27 junio	Ejercicios examen (1 y 2)		Carga inicial de la base de datos y consultas	Entrega Lab 3
04 julio - 15 julio	Semana de Exámenes		Exámenes	
	The end			

# Ejercicio

En base al modelo de Entidad-Relación del proyecto

- 1) Cree el modelo-relacional en FN1 (20 coins)
- 2) Transforme el modelo-relacional a FN2 (10 coins)
- 3) Transforme el modelo-relacional a FN3 (10 coins)



Coins - Plazo máximo Miércoles 25 - 23:59

# Clase - Taller

- 1) Crear DB
- 2) Insertar datos
- 3) Actualizar datos



# Clase - Taller

- 1) Crear DB
- 2) Insertar datos
- 3) Actualizar datos

# Instalar PostgreSQL

docker pull postgres

```
leo@dell:~$ docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
214ca5fb9032: Pull complete
e6930973d723: Pull complete
aea7c534f4e1: Pull complete
d0ab8814f736: Pull complete
648cc138980a: Pull complete
7804b894301c: Pull complete
cfce56252c3f: Pull complete
8cce7305e3b6: Pull complete
8e979d981f07: Pull complete
4b0a5f0b050c: Pull complete
a6bc1be6e5b0: Pull complete
d115610a4c3b: Pull complete
bf74ca3879b4: Pull complete
Digest: sha256:117e7b9287612505575ac11db1cf81742eea6fd5cd8b2ce26e40f366b1f74e25
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

# Corriendo contenedor

```
docker run --workdir /bdd --volume $(pwd):/bdd --name docker-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
```

```
leo@dell:~/Documentos/UDP2/Clases_BDD/bdd_18_insert_update$ docker run --name docker-postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
0ea04001edf6772bbaa724575aac159afc4c5f4935cabdf89ab17fd49be9672b
```

```
docker ps
```

```
leo@dell:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0ea04001edf6	postgres	"docker-entrypoint.s..."	3 minutes ago	Up 3 minutes	5432/tcp	docker-postgres



# Ejecutando contenedor modo bash

```
docker exec -it --user root docker-postgres bash
```

```
leo@dell:~/Documentos/UDP2/Clases_BDD/bdd_18_insert_update$ docker exec -it --user root docker-postgres4 bash
root@5d6b9d5a4653:/bdd#
```

```
apt update && apt install vim nano less
```

```
root@5d6b9d5a4653:/bdd# apt update && apt install vim nano less
0% [Working]
Hit:1 http://deb.debian.org/debian bullseye InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 http://apt.postgresql.org/pub/repos/apt bullseye-pgdg InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
3 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

# Iniciando como usuario postgres

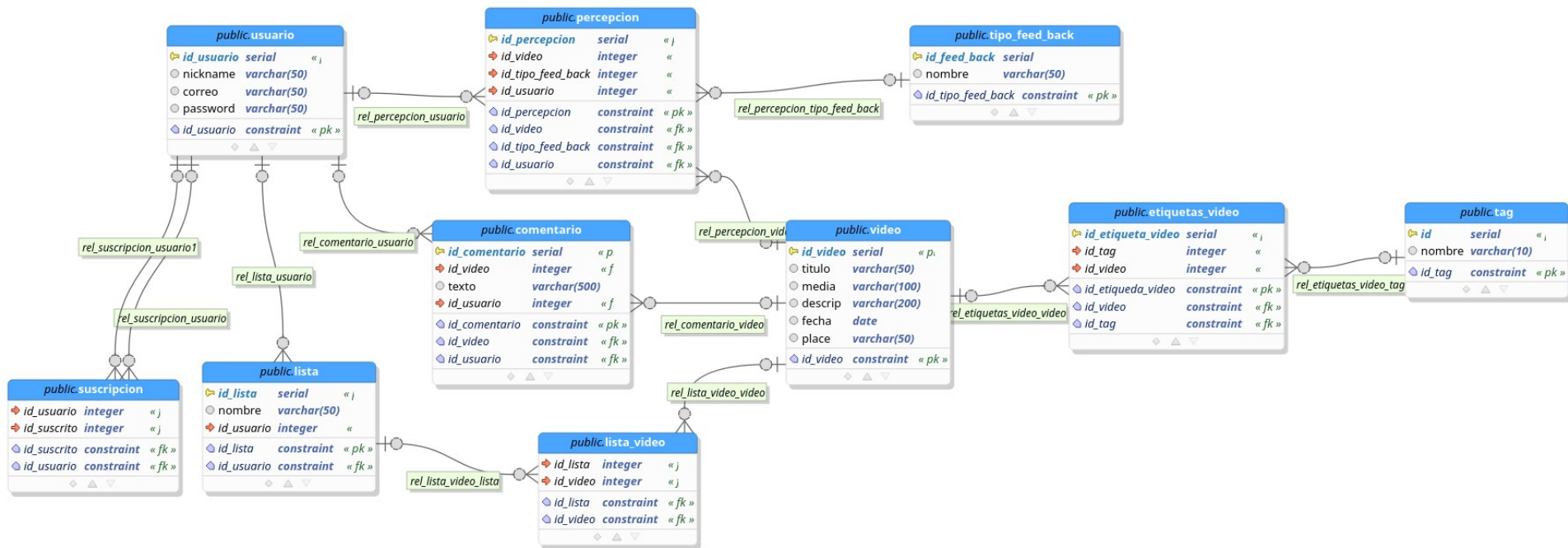
su postgres

```
Processing triggers for libc-bin (2.15-1)
root@5d6b9d5a4653:/bdd# su postgres
postgres@5d6b9d5a4653:/bdd$
```

ls -ltr

```
postgres@5d6b9d5a4653:/bdd$ ls -ltr
total 1992
-rw-rw-r-- 1 1000 1000 12273 May 24 03:23 miTube.dbm
-rw-rw-r-- 1 1000 1000 7353 May 24 03:24 mitube.sql
-rw-rw-r-- 1 1000 1000 207547 May 24 03:27 mitube.png
-rwxrwxr-x 1 1000 1000 135 May 24 03:42 run_docker.sh
-rwxrwxr-x 1 1000 1000 47 May 24 03:46 remove_exited_containers.sh
-rw-rw-r-- 1 1000 1000 1800334 May 24 03:50 bdd_18_insert_update.odp
postgres@5d6b9d5a4653:/bdd$
```

# MiTube



# Iniciando como usuario postgres

less mitube.sql

```
-- Database generated with pgModeler (PostgreSQL Database Modeler).
-- pgModeler version: 0.9.4
-- PostgreSQL version: 13.0
-- Project Site: pgmodeler.io
-- Model Author: ---

-- Database creation must be performed outside a multi lined SQL file.
-- These commands were put in this file only as a convenience.
--
-- object: "miTube" | type: DATABASE --
-- DROP DATABASE IF EXISTS "miTube";
CREATE DATABASE "miTube";
-- ddl-end --

-- object: public.video | type: TABLE --
-- DROP TABLE IF EXISTS public.video CASCADE;
CREATE TABLE public.video (
    id_video serial NOT NULL,
    titulo varchar(50),
    media varchar(100),
    descrip varchar(200),
    fecha date,
    place varchar(50),
    CONSTRAINT id_video PRIMARY KEY (id_video)
);
-- ddl-end --
ALTER TABLE public.video OWNER TO postgres;
```

# Crear DB para MiTube

```
createdb mitube
```

```
psql -U postgres -d mitube -f mitube.sql
```

```
CREATE DATABASE mitube;
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  username VARCHAR(50) NOT NULL,
  password VARCHAR(50) NOT NULL,
  email VARCHAR(50) NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE users ADD COLUMN last_login TIMESTAMP;
CREATE TABLE videos (
  id SERIAL PRIMARY KEY,
  title VARCHAR(100) NOT NULL,
  description VARCHAR(200) NOT NULL,
  url VARCHAR(200) NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE videos ADD COLUMN views INTEGER;
CREATE TABLE comments (
  id SERIAL PRIMARY KEY,
  video_id INTEGER NOT NULL,
  user_id INTEGER NOT NULL,
  text VARCHAR(200) NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE comments ADD COLUMN parent_id INTEGER;
CREATE TABLE likes (
  id SERIAL PRIMARY KEY,
  video_id INTEGER NOT NULL,
  user_id INTEGER NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE likes ADD COLUMN is_liked BOOLEAN;
CREATE TABLE playlists (
  id SERIAL PRIMARY KEY,
  title VARCHAR(50) NOT NULL,
  description VARCHAR(100) NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE playlists ADD COLUMN video_id INTEGER;
CREATE TABLE subscriptions (
  id SERIAL PRIMARY KEY,
  channel_id INTEGER NOT NULL,
  user_id INTEGER NOT NULL,
  created_at TIMESTAMP NOT NULL
);
ALTER TABLE subscriptions ADD COLUMN is_subscribed BOOLEAN;
```

# En la consola de postgres

```
psql -U postgres -d mitube
```

```
\d
```

```
test1=# \d
```

List of relations			
Schema	Name	Type	Owner
public	comentario	table	postgres
public	comentario_id_comentario_seq	sequence	postgres
public	etiquetas_video	table	postgres
public	etiquetas_video_id_etiqueta_video_seq	sequence	postgres
public	lista	table	postgres
public	lista_id_lista_seq	sequence	postgres
public	lista_video	table	postgres
public	percepcion	table	postgres
public	percepcion_id_percepcion_seq	sequence	postgres
public	suscripcion	table	postgres
public	tag	table	postgres
public	tag_id_seq	sequence	postgres
public	tipo_feed_back	table	postgres
public	tipo_feed_back_id_feed_back_seq	sequence	postgres
public	usuario	table	postgres
public	usuario_id_usuario_seq	sequence	postgres
public	video	table	postgres
public	video_id_video_seq	sequence	postgres
(18 rows)			

# En la consola de postgres

```
psql -U postgres -d mitube
```

```
\d
```

```
test1=# \d
```

List of relations			
Schema	Name	Type	Owner
public	comentario	table	postgres
public	comentario_id_comentario_seq	sequence	postgres
public	etiquetas_video	table	postgres
public	etiquetas_video_id_etiqueta_video_seq	sequence	postgres
public	lista	table	postgres
public	lista_id_lista_seq	sequence	postgres
public	lista_video	table	postgres
public	percepcion	table	postgres
public	percepcion_id_percepcion_seq	sequence	postgres
public	suscripcion	table	postgres
public	tag	table	postgres
public	tag_id_seq	sequence	postgres
public	tipo_feed_back	table	postgres
public	tipo_feed_back_id_feed_back_seq	sequence	postgres
public	usuario	table	postgres
public	usuario_id_usuario_seq	sequence	postgres
public	video	table	postgres
public	video_id_video_seq	sequence	postgres
(18 rows)			

# En la consola de postgres

```
psql -U postgres -d mitube
```

```
\d
```

```
test1=# \d
```

List of relations			
Schema	Name	Type	Owner
public	comentario	table	postgres
public	comentario_id_comentario_seq	sequence	postgres
public	etiquetas_video	table	postgres
public	etiquetas_video_id_etiqueta_video_seq	sequence	postgres
public	lista	table	postgres
public	lista_id_lista_seq	sequence	postgres
public	lista_video	table	postgres
public	percepcion	table	postgres
public	percepcion_id_percepcion_seq	sequence	postgres
public	suscripcion	table	postgres
public	tag	table	postgres
public	tag_id_seq	sequence	postgres
public	tipo_feed_back	table	postgres
public	tipo_feed_back_id_feed_back_seq	sequence	postgres
public	usuario	table	postgres
public	usuario_id_usuario_seq	sequence	postgres
public	video	table	postgres
public	video_id_video_seq	sequence	postgres
(18 rows)			



# MiTube - Tabla usuario

\d usuario

```
test1=# \d usuario
```

Table "public.usuario"				
Column	Type	Collation	Nullable	Default
id_usuario	integer		not null	nextval('usuario_id_usuario_seq'::regclass)
nickname	character varying(50)			
correo	character varying(50)			
password	character varying(50)			

Indexes:

"id\_usuario" PRIMARY KEY, btree (id\_usuario)

Referenced by:

TABLE "suscripcion" CONSTRAINT "id\_suscrito" FOREIGN KEY (id\_suscrito) REFERENCES usuario(id\_usuario)

TABLE "percepcion" CONSTRAINT "id\_usuario" FOREIGN KEY (id\_usuario) REFERENCES usuario(id\_usuario)

TABLE "comentario" CONSTRAINT "id\_usuario" FOREIGN KEY (id\_usuario) REFERENCES usuario(id\_usuario)

TABLE "lista" CONSTRAINT "id\_usuario" FOREIGN KEY (id\_usuario) REFERENCES usuario(id\_usuario)

TABLE "suscripcion" CONSTRAINT "id\_usuario" FOREIGN KEY (id\_usuario) REFERENCES usuario(id\_usuario)

# Insertando un Usuario

```
INSERT INTO usuario (nickname, correo, password)  
values ('FakeNick', 'fake@mail.com', 'bm9sYXNlLjEK');
```

```
test1=# INSERT INTO usuario (nickname, correo, password) values ('FakeNick', 'fake@mail.com', 'bm9sYXNlLjEK');  
INSERT 0 1
```

# Insertando un video

```
INSERT INTO video (titulo, media, descrip)
```


```
Values (
```

```
'Arch Enemy - Sunset Over The Empire',
```

```
'https://youtu.be/E3mOvCMG24I',
```

```
'New single "Sunset Over The Empire" out everywhere!');
```

```
test2=# INSERT INTO video (titulo, media, descrip) values ('Arch Enemy - Sunset Over The Empire', 'https://youtu.be/E3mOvCMG24I', 'New single "Sunset Over The Empire" out everywhere!')
test2=# ;
INSERT 0 1
```



```
test2=# select * from video;
```

id_video	titulo	media	descrip	fecha	place
1	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-24	

(1 row)

# MiTube - Tabla Video

\d video

```
test1=# \d video
```

Table "public.video"				
Column	Type	Collation	Nullable	Default
id_video	integer		not null	nextval('video_id_video_seq'::regclass)
titulo	character varying(50)			
media	character varying(100)			
descrip	character varying(200)			
fecha	date			
place	character varying(50)			

Indexes:

"id\_video" PRIMARY KEY, btree (id\_video)

Referenced by:

TABLE "percepcion" CONSTRAINT "id\_video" FOREIGN KEY (id\_video) REFERENCES video(id\_video)

TABLE "comentario" CONSTRAINT "id\_video" FOREIGN KEY (id\_video) REFERENCES video(id\_video)

TABLE "lista\_video" CONSTRAINT "id\_video" FOREIGN KEY (id\_video) REFERENCES video(id\_video)

TABLE "etiquetas\_video" CONSTRAINT "id\_video" FOREIGN KEY (id\_video) REFERENCES video(id\_video)

# Insertando un video

```
INSERT INTO video (titulo, media, descrip)
  Values (
    'Arch Enemy - Sunset Over The Empire',
    'https://youtu.be/E3mOvCMG24I',
    'New single "Sunset Over The Empire" out everywhere!');
```

```
test2=# INSERT INTO video (titulo, media, descrip) values ('Arch Enemy - Sunset Over The Empire', 'https://youtu.be/E3mOvCMG24I', 'New single "Sunset Over The Empire" out everywhere!')
test2=# ;
INSERT 0 1
```

# Insertando un video

```
INSERT INTO video (titulo, media, descrip, fecha)
values (
    'Arch Enemy - Sunset Over The Empire',
    'https://youtu.be/E3mOvCMG24I',
    'New single "Sunset Over The Empire" out everywhere!',
    '20220525');
```

```
est2=# INSERT INTO video (titulo, media, descrip, fecha) values ('Arch Enemy - Sunset Over The Empire', 'https://youtu.be/E3mOvCMG24I', 'New single "Sunset Over The Empire" out everywhere!', '20220525');
INSERT 0 1
est2=# select * from video;
```

id video	titulo	media	descrip	fecha	place
1	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-24	
2	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-25	

```
2 rows)
```

# Insertando un video

```
INSERT INTO video (titulo, media, descrip, fecha)
values (
    'Arch Enemy - Sunset Over The Empire',
    'https://youtu.be/E3mOvCMG24I',
    'New single "Sunset Over The Empire" out everywhere!',
    '2022-05-25');
```

```
test2=# INSERT INTO video (titulo, media, descrip, fecha) values ('Arch Enemy - Sunset Over The Empire', 'https://youtu.be/E3mOvCMG24I', 'New single "Sunset Over The Empire" out everywhere!', '2022-05-25');
INSERT 0 1
test2=# select * from video;
```

id_video	titulo	media	descrip	fecha	place
1	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-24	
2	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-25	
3	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-25	

(3 rows)

# Postgres date

<< PostgreSQL INSERT statement, and we will use the YYYYMMDD format to INSERT >>

<https://www.postgresql.org/docs/9.0/functions-datetime.html>



# Actualizando un video

```
UPDATE video
  set titulo='Hans Zimmer performs INCEPTION "Time"',
      media='https://youtu.be/xdYYN-4ttDg',
      descrip='at Hollywood in Vienna 2018'
 where id_video=2;
```

```
test2=# UPDATE video set titulo='Hans Zimmer performs INCEPTION "Time"', media='https://youtu.be/xdYYN-4ttDg', descrip='at Hollywood in Vienna 2018' where id_video=2;
```

```
UPDATE 1
```

```
test2=# select * from video;
```

id_video	titulo	media	descrip	fecha	place
1	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-24	
3	Arch Enemy - Sunset Over The Empire	https://youtu.be/E3mOvCMG24I	New single "Sunset Over The Empire" out everywhere!	2022-05-25	
2	Hans Zimmer performs INCEPTION "Time"	https://youtu.be/xdYYN-4ttDg	at Hollywood in Vienna 2018	2022-05-25	

```
(3 rows)
```

# Links referencias

- <https://www.postgresql.org/docs/9.5/sql-insert.html>
- <https://www.postgresql.org/docs/9.1/sql-update.html>

**Material Disponible en:**

[https://gitlab.com/l30bravo/db\\_udp](https://gitlab.com/l30bravo/db_udp)

**Muchas  
gracias!**

**Correo:**

leonardo.bravo@mail.udp.cl

**Grupo de WhatsApp**

