

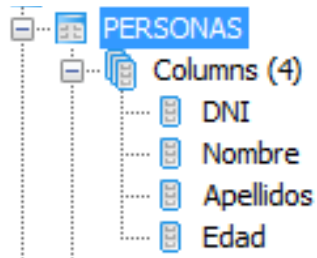
MANIPULAR BD POSTGRESQL

1- DEFINE las siguientes tablas a través de la interfaz gráfica en Windows e INSERTA datos en ella

1.1 Previamente necesitarás conocer los tipos de datos que tiene Psql.
Disponibles en el siguiente enlace: <http://www.postgresql.org.ar/trac/wiki/datatype.html>

1.2 Crear Tabla PERSONAS:

- DNI: Debe incluir 9 caracteres alfanuméricos. Será PK
- Nombre: caracteres
- Apellidos: caracteres
- Edad: entero menor de 99



1.3 Pega aquí el pantallazo de la/s sentencia/s SQL resultante/s de la configuración que has hecho gráficamente.

```
CREATE TABLE PERSONAS(  
    DNI varchar(10),  
    Nombre varchar(15),  
    Apellidos(20),  
    Edad int  
    PRIMARY KEY(DNI)  
    CHECK(Edad BETWEEN 0 AND 100)  
);
```

1.4 Busca dónde introducir 2 filas de datos y pega aquí el pantallazo

```
INSERT INTO public.personas(  
    dni, nombre, apellidos, edad)  
VALUES ('72745715X', 'Iker', 'Gomez', 22);  
  
INSERT INTO public.personas(  
    dni, nombre, apellidos, edad)  
VALUES ('72216789B', 'Jon', 'Sanchez', 21);
```

2- LEE los siguientes comandos, RECUERDA o BUSCA otros comandos SQL que vayas a necesitar, CREA las tablas por la terminal en Ubuntu e INSERTA datos

2.1 COMANDOS:

Previo a ejecutar comandos sql hay que conectarse a la consola:

sudo su postgres

psql

1) Comando: **\h**

Este comando sirve para ver la ayuda con respecto a la sintaxis de nuestras consultas SQL, por ejemplo: **\h INSERT**

Al utilizar este comando nos mostrará cómo debemos escribir la sentencia "insert", así como parámetros y el orden correspondiente:

2) Comando: **\dt**

Mostrará la lista de las tablas de la base de datos que tengamos seleccionada.

3) Comando: **\?**

Este comando nos mostrará una lista de todos los comandos que podemos usar en la consola interactiva de postgresql.

Para desplazarnos por la lista ocupamos las flechas de nuestro teclado y para salir solo presionamos "q".

4) Comando: **\conninfo**

Este comando nos mostrará la información de nuestra conexión activa.

5) Iniciar la consola: **psql -U user -W -h host database**

En el caso de ya estar iniciada la consola con el usuario de B.D. (en nuestro caso postgres) sólo haría falta hacer psql

6) Comando: **\l**

Lista las bases de datos

7) Comando: **\c DB**

Seleccionar una base de datos o cambiar de BD

8) Comando: **\d TABLA**

Para ver la información de la estructura de una tabla (=DESC)

- 9) Vaciar una tabla:

TRUNCATE TABLE *table* **RESTART IDENTITY**

Con este comando borramos el contenido de una tabla y reiniciamos su índice sino agregamos RESTART IDENTITY nuestros índices no serán reiniciados y seguirán según el último registro

- 10) Crear una base de datos:

CREATE DATABASE *basename* **ENCODING = 'UTF8'**

- 11) Borrar o eliminar una base de datos:

DROP DATABASE *basename*;

- 12) Borrar o eliminar una tabla:

DROP TABLE *tablename*;

- 13) Uso de comillas:

SELECT "column" FROM "table" WHERE "column" = 'value';

Generalmente utilizar comillas dobles (") para columnas y comillas simples (') para valores. No es una regla pero es necesario cuando utilizamos nombres reservados, por ejemplo:

SELECT **to** FROM table;

En este caso tenemos un campo llamado "to", esto nos dará un error de sintaxis, por lo tanto tendremos que usar comillas dobles:

SELECT "**to**" FROM table;

- 14) Comando:\q

Salir del cliente psql:

2.2 Crea la siguiente tabla:

Tabla: ANIMALES

- Id: campo entero que se autoincrementa. Será PK
- Nombre: caracteres. Longitud 10
- Fecha_Nacimiento. Fecha
- Dueño: caracteres. Longitud 30
- Dirección. Caracteres. Longitud 30

```
CREATE TABLE ANIMALES
(
    id SERIAL PRIMARY KEY,
    nombre varchar(10),
    fechanac date,
    dueño varchar(30),
    direccion varchar(30)
);
```

2.3 Introduce 2 filas de datos.

```
INSERT INTO public.animales(
    nombre, fechanac, "dueño", direccion)
VALUES ('Freya', '2020/6/20', 'yo', 'mi casa');

INSERT INTO public.animales(
    nombre, fechanac, "dueño", direccion)
VALUES ('Goku', '2019/9/16', 'tu', 'tu casa');
```

2.4 Saca la descripción de la tabla para ver de qué tipo son sus columnas

```
CREATE TABLE public.personas
(
    dni character varying(10) NOT NULL,
    nombre character varying(15),
    apellidos character varying(20),
    edad integer,
    CONSTRAINT personas_pkey PRIMARY KEY (dni),
    CONSTRAINT personas_edad_check CHECK (edad >= 0 AND edad <= 100)
);
WITH (
    OIDS=FALSE
);
ALTER TABLE public.personas
    OWNER TO postgres;
```

```
CREATE TABLE public.animales
(
    id integer NOT NULL DEFAULT nextval('animales_id_seq'::regclass),
    nombre character varying(10),
    fechanac date,
    "dueño" character varying(30),
    direccion character varying(30),
    CONSTRAINT animales_pkey PRIMARY KEY (id)
);
WITH (
    OIDS=FALSE
);
ALTER TABLE public.animales
    OWNER TO postgres;
```

2.5Lista los valores introducidos en la tabla.

	dni character varying(10)	nombre character varying(15)	apellidos character varying(20)	edad integer
1	72745715X	Iker	Gomez	22
2	72216789B	Jon	Sanchez	21

	id integer	nombre character varying(10)	fechanac date	dueño character varying(30)	direccion character varying(30)
1	1	Freya	2020-06-20	yo	mi casa
2	2	Goku	2019-09-16	tu	tu casa