



Laboratorio 1

Profesor: Leonardo Bravo

Ayudante: Marcelo Fuenzalida

Alumno: Juan Pablo Schiaffino

Correo: juan.schiaffino_l@mail.udp.cl

Sección: 1

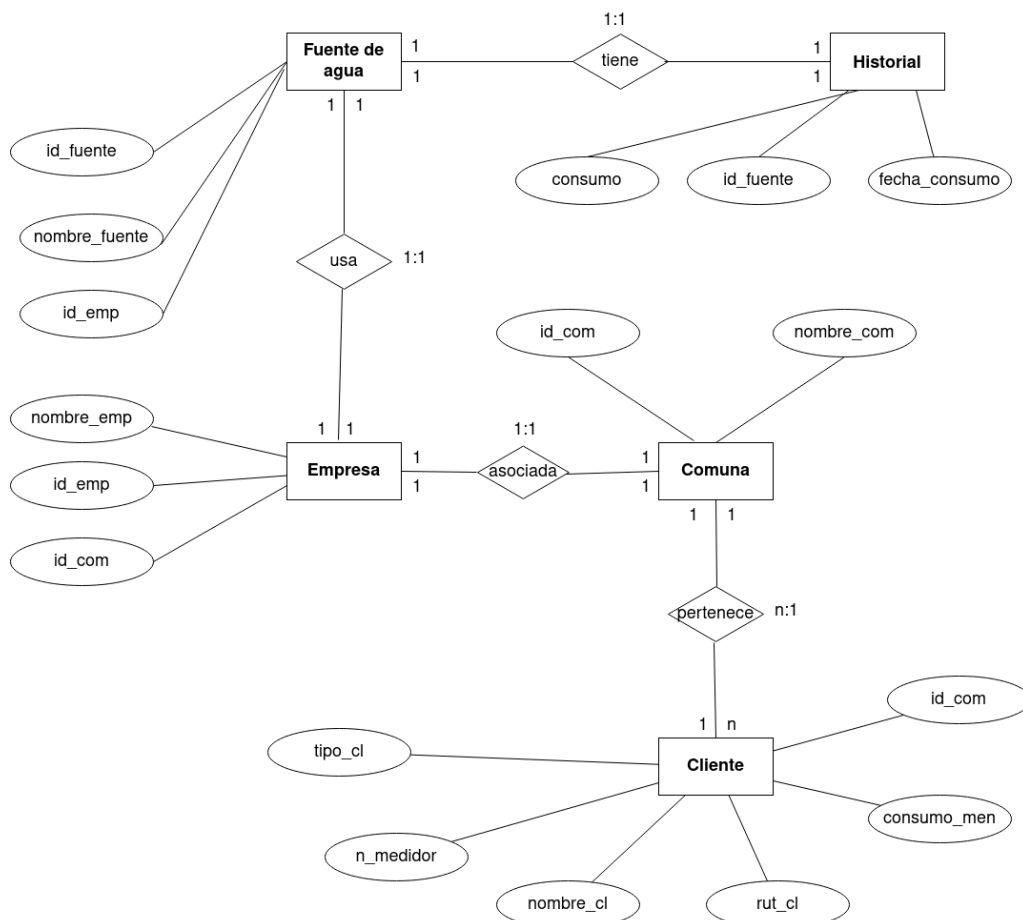
Fecha: 26/05/2022

Introducción

Para este laboratorio se nos solicita crear un sistema de base de datos para llevar control sobre el consumo de agua de los clientes de Santiago. Para realizar esto tendremos que formular el modelo entidad relación identificando las entidades y atributos respectivos del problema.

Desarrollo

Lo primero de todo es construir el modelo entidad relación en base al problema planteado.



A continuación adjunto pantallazos acerca de cómo se creó la base de datos junto a las entidades con sus respectivos atributos.

Se crea la base de datos con el nombre de "lab1"

```
[postgres@arco ~]$ createdb lab1;  
[postgres@arco ~]$
```

Luego ejecutamos el siguiente comando desde la consola de postgres.

```
[postgres@arco ~]$ psql -U postgres -d lab1 -f lab1.sql;  
DROP TABLE  
DROP TABLE  
DROP TABLE  
DROP TABLE  
DROP TABLE  
CREATE TABLE  
ALTER TABLE  
CREATE TABLE  
ALTER TABLE  
CREATE TABLE  
ALTER TABLE  
CREATE TABLE  
ALTER TABLE  
CREATE TABLE  
ALTER TABLE  
INSERT 0 1  
INSERT 0 1  
INSERT 0 1  
INSERT 0 1  
INSERT 0 1  
INSERT 0 1
```

Uno de los parámetros principales a destacar aquí es que le damos la opción "-f" indicando el archivo lab1.sql que básicamente es un código que nos creará las tablas dentro de la base de datos creada previamente, además de desde ahí ya hacer las tres inserciones solicitadas por entidad.

```
2 /* [Parte dos */  
1  
>>3 DROP TABLE IF EXISTS Cliente CASCADE;  
1 DROP TABLE IF EXISTS Historial CASCADE;  
2 DROP TABLE IF EXISTS Fuente_de_Agua CASCADE;  
3 DROP TABLE IF EXISTS Empresa CASCADE;  
4 DROP TABLE IF EXISTS Comuna CASCADE;  
5
```

Se eliminan las tablas en caso de existir, esto está básicamente porque como constantemente editaba la base de datos, ahorrarme el trabajo de eliminar desde adentro con el comando de "DROP DATABASE filename;"

Ahora se procederá a crear las tablas respectivas dentro del archivo, adjunto las imagenes correspondientes a la entidad además de detallar en caso de existir llave primaria y/o llave foránea.

- Entidad: Comuna
Atributos: nombre_com, id_com
Llave primaria (PK): id_com

```
8 CREATE TABLE public.Comuna(  
7     nombre_com name NOT NULL,  
6     id_com integer NOT NULL,  
5     CONSTRAINT comuna_pkey PRIMARY KEY (id_com)  
4 );  
3  
2 ALTER TABLE IF EXISTS public.Comuna  
1     OWNER to postgres;
```

- Entidad: Cliente
Atributos: tipo_cl, n_medidor, nombre_cl, rut_cl, consumo_men, id_com
Llave primaria (PK): n_medidor
Llave foránea: id_com

```
1 CREATE TABLE public.Cliente(  
2     tipo_cl text NOT NULL,  
3     n_medidor integer NOT NULL,  
4     nombre_cl name NOT NULL,  
5     rut_cl integer NOT NULL,  
6     consumo_men integer NOT NULL,  
7     id_com integer NOT NULL,  
8     CONSTRAINT cliente_pkey PRIMARY KEY (n_medidor),  
9     CONSTRAINT fk_comuna FOREIGN KEY (id_com)  
10    REFERENCES public.Comuna (id_com) MATCH SIMPLE  
11    ON UPDATE NO ACTION  
12    ON DELETE NO ACTION  
13 );  
14  
15 ALTER TABLE IF EXISTS public.Cliente  
16     OWNER to postgres;  
17
```

- Entidad: Empresa
Atributos: id_emp, nombre_emp, id_com
Llave primaria (PK): id_emp
Llave foránea: id_com

```
9 CREATE TABLE public.Empresa(  
10     id_emp integer NOT NULL,  
11     nombre_emp name NOT NULL,  
12     id_com integer NOT NULL,  
13     CONSTRAINT empresa_pkey PRIMARY KEY (id_emp),  
14     CONSTRAINT fk_comuna FOREIGN KEY (id_com)  
15         REFERENCES public.Comuna (id_com) MATCH SIMPLE  
16         ON UPDATE NO ACTION  
17         ON DELETE NO ACTION  
18 );  
19  
20 ALTER TABLE IF EXISTS public.Empresa  
21     OWNER to postgres;  
22
```

- Entidad: Fuente_de_Agua
Atributos: id_fuente, nombre_fuente, id_emp
Llave primaria (PK): id_fuente
Llave foránea: id_emp

```
8 CREATE TABLE public.Fuente_de_Agua(  
9     id_fuente integer NOT NULL,  
10     nombre_fuente name NOT NULL,  
11     id_emp integer NOT NULL,  
12     CONSTRAINT fuente_de_agua_pkey PRIMARY KEY (id_fuente),  
13     CONSTRAINT fk_empresa FOREIGN KEY (id_emp)  
14         REFERENCES public.Empresa (id_emp) MATCH SIMPLE  
15         ON UPDATE NO ACTION  
16         ON DELETE NO ACTION  
17 );  
18  
19 ALTER TABLE IF EXISTS public.Fuente_de_Agua  
20     OWNER to postgres;  
21
```

- Entidad: Historial

Atributos: fecha_consumo, consumo, id_fuente

Llave primaria (PK): fecha_consumo

Llave foránea: id_fuente

```
10 CREATE TABLE public.Historial(  
11     fecha_consumo date NOT NULL,  
12     consumo integer NOT NULL,  
13     id_fuente integer NOT NULL,  
14     CONSTRAINT historial_pkey PRIMARY KEY (fecha_consumo),  
15     CONSTRAINT fk_fuente_de_agua FOREIGN KEY (id_fuente)  
16         REFERENCES public.Fuente_de_Agua (id_fuente) MATCH SIMPLE  
17         ON UPDATE NO ACTION  
18         ON DELETE NO ACTION  
19 );  
20  
21 ALTER TABLE IF EXISTS public.Historial  
22     OWNER to postgres;  
23
```

Una vez creadas las tablas, procedemos a la inserción de tres registros en cada una de ellas. Se adjuntan las fotografías correspondientes.

- Comuna

```
12 INSERT INTO public.Comuna(  
11     nombre_com, id_com)  
10     VALUES ('Providencia', 7500000);  
9  
8 INSERT INTO public.Comuna(  
7     nombre_com, id_com)  
6     VALUES ('QuintaNormal', 8500000);  
5  
4 INSERT INTO public.Comuna(  
3     nombre_com, id_com)  
2     VALUES ('Ñuñoa', 7750000);
```

- Cliente

```
10 INSERT INTO public.Cliente(  
9     tipo_cl, n_medidor, nombre_cl, rut_cl, consumo_men, id_com)  
~ 8     VALUES ('Particular', 1, 'Pablito', 201234567, 100, 7500000);  
7  
6 INSERT INTO public.Cliente(  
5     tipo_cl, n_medidor, nombre_cl, rut_cl, consumo_men, id_com)  
4     VALUES ('Particular', 2, 'Leopoldo', 207654321, 2500, 7750000);  
3  
2 INSERT INTO public.Cliente(  
1     tipo_cl, n_medidor, nombre_cl, rut_cl, consumo_men, id_com)  
~ 106     VALUES ('Restauran', 3, 'Cristobalin', 203211234, 1800, 8500000);
```

- Empresa

```
12 INSERT INTO public.Empresa(  
13     id_emp, nombre_emp, id_com)  
14     VALUES (777, 'LosSanguches', 7500000);  
15  
16 INSERT INTO public.Empresa(  
17     id_emp, nombre_emp, id_com)  
18     VALUES (666, 'Biblioteca', 7750000);  
19  
20 INSERT INTO public.Empresa(  
21     id_emp, nombre_emp, id_com)  
22     VALUES (555, 'Liquidos', 8500000);  
23
```

- Fuente_de_Agua

```
13 INSERT INTO public.Fuente_de_Agua(  
14     id_fuente, nombre_fuente, id_emp)  
15     VALUES (212, 'FuentesLiq', 555);  
16  
17 INSERT INTO public.Fuente_de_Agua(  
18     id_fuente, nombre_fuente, id_emp)  
19     VALUES (323, 'FuentesSan', 777);  
20  
21 INSERT INTO public.Fuente_de_Agua(  
22     id_fuente, nombre_fuente, id_emp)  
23     VALUES (434, 'FuentesBib', 666);  
24
```

- Historial

```
14 INSERT INTO public.Historial(  
15     fecha_consumo, consumo, id_fuente)  
16     VALUES ('23/12/2022', 1000, 323);  
17  
18 INSERT INTO public.Historial(  
19     fecha_consumo, consumo, id_fuente)  
20     VALUES ('11/10/2021', 2000, 434);  
21  
22 INSERT INTO public.Historial(  
23     fecha_consumo, consumo, id_fuente)  
24     VALUES ('16/6/2023', 5000, 212);  
25
```

Ya creadas los registros procedemos en el apartado de análisis a realizar las Querys para los requisitos solicitados en el problema.



Análisis

Por la nueva normativa una persona particular no puede superar 150 m3 al mes y una empresa no puede superar los 1000 m3 al mes, identifique la query SQL para:

- Empresas que no cumplen con la nueva normativa
SELECT nombre_cl FROM cliente WHERE consumo_men > 1000 and tipo_cl != 'Particular';

```
lab1=# SELECT nombre_cl FROM cliente WHERE consumo_men > 1000 and tipo_cl != 'Particular';
 nombre_cl
-----
Cristobalin
(1 row)
```

- Comuna que más consume agua
- Proveedores de agua con menor reserva

Conclusión

Para cerrar este laboratorio podemos decir que fue realizado con éxito ya que el modelo entidad relación nos permitió hacer las relaciones de manera óptima, además de poder introducir las tablas previamente mencionadas con sus llaves primarias y foráneas en caso de que tuviesen, y para las queries solicitadas también podemos decir que fueron consultadas con éxito arrojando los parámetros buscados.

enlace a proyecto github:

<https://github.com/juanpabloslopez/lab1bd>