

# Actividad 31 - Postgresql

## Ejercicio 1

1. Crear una base de datos llamada **call\_list**.

```
CREATE DATABASE call_list;
```

```
\c call_list;
```

2. En la base de datos recién creada, crear la tabla **users** con los campos id (clave primaria), first\_name, email.

```
call_list=# CREATE TABLE Users (id SERIAL PRIMARY KEY,  
call_list=# first_name VARCHAR(15),  
call_list=# email VARCHAR(30)  
call_list=# );
```

3. Ingresar un usuario, llamado **Carlos** (el resto de los datos deben inventarlos).

```
call_list=# INSERT INTO users (first_name, email) VALUES ('Carlos',  
call_list=# 'carlos@gmail.com');
```

4. Ingresar un usuario, llamada **Laura** (el resto de los datos deben inventarlos).

```
call_list=# INSERT INTO users (first_name, email) VALUES ('Laura',  
call_list=# 'laura@gmail.com');
```

5. Crear una tabla llamada **calls** con los campos id (clave primaria), phone, date, user\_id (foreign key relacionado a users).

```
call_list=# CREATE TABLE calls (id SERIAL PRIMARY KEY,
```

```
call_list(# phone VARCHAR(11),
call_list(# date DATE,
call_list(# user_id INTEGER REFERENCES users(id)
call_list(# );
```

6. Agregar a la tabla **users** el campo **last\_name**.

```
call_list=# ALTER TABLE users ADD COLUMN last_name VARCHAR(20);
```

7. Ingresar el apellido del usuario Carlos.

```
call_list=# UPDATE users set last_name = 'Diaz' where id=1;
```

8. Ingresar el apellido del usuario Laura.

```
call_list=# UPDATE users set last_name = 'Perez' where id=2;
```

9. Ingresar 6 llamadas asociadas al usuario Laura.

```
call_list=# INSERT INTO calls (phone, date, user_id) VALUES
    ('+5683616606', '01-04-18', 2), ('+5683616605', '02-04-18', 2),
    ('+5683616604', '03-04-18', 2), ('+5683616605', '04-04-18', 2),
    ('+5683616607', '10-04-18', 2), ('+5683616604', '05-04-18', 2);
```

10. Ingresar 4 llamadas asociadas al usuario Carlos.

```
call_list=# INSERT INTO calls (phone, date, user_id) VALUES
    ('+5683616606', '01-06-18', 1), ('+5683616605', '02-06-18', 1),
    ('+5683616604', '03-07-18', 1), ('+5683616605', '04-05-18', 1);
```

11. Crear un nuevo usuario.

```
call_list=# INSERT INTO users (first_name, email, last_name) VALUES
    ('Patricia', 'patricia@gmail.com', 'Vera');
```

12. Seleccionar la cantidad de llamados de cada uno de los

usuarios (nombre de usuario y cantidad de llamadas).

```
call_list=# SELECT first_name, COUNT(phone) FROM users INNER JOIN  
calls on (user_id = calls.user_id) GROUP BY first_name;
```

13. Seleccionar los llamados del usuario llamado Carlos ordenados por fecha en orden descendente.

```
call_list=# SELECT * FROM calls WHERE user_id = 1 order by date DESC;
```

### **Nuevos cambios solicitados por cliente.**

"Necesito agregar a la base una tabla de auditoría que registre el motivo del borrado de una llamada y el usuario que lo efectuó."

```
call_list=# CREATE TABLE audit ( id SERIAL PRIMARY KEY,  
call_list(# reason VARCHAR(30),  
call_list(# call_id INTEGER REFERENCES calls(id),  
call_list(# user_id INTEGER REFERENCES users(id)  
call_list(# );
```

14. A partir de la base anterior, agregar este requerimiento y modelar la base de datos (agregar print de pantalla [utilizar <https://www.draw.io/>]).