



Universidad de  
los Andes  
Colombia

Educación  
Continua  
Vicerrectoría Académica

# Diseña tu mundo y explora los demás

*Ejercicio de clase: Creación de tablas SQL*

## Contexto:

Vamos a diseñar mundos y a representarlos en bases de datos. Hoy cada persona va a diseñar un pedazo de un mundo ficticio. Una ciudad y sus habitantes, un sistema escolar con sus estudiantes, un zoológico, un museo, un gimnasio, un videojuego con personajes fantásticos, un planeta en el universo. Luego todos nos vamos a conectar a los mundos de otros y vamos a interactuar con ese mundo.



Fotos de [Josh Hild](#), [Pixabay](#), [Lance Martin](#), [Brett Sayles](#), [Pixabay](#).



Universidad de  
los Andes  
Colombia

Educación  
Continua  
Vicerrectoría Académica

## Ejercicio

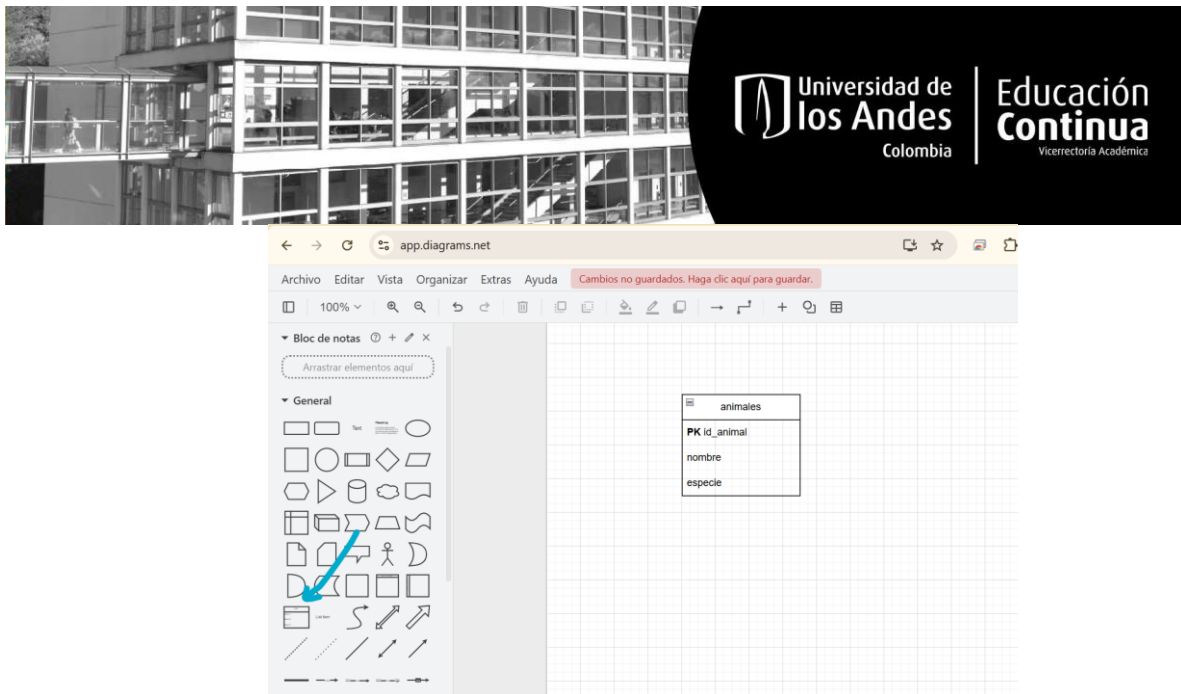
Primera parte: creación del mundo y la misión (10 minutos) .....	2
Segunda parte: modelado del mundo (20 minutos) .....	2
Tercera parte: creamos un servicio de base de datos (10 minutos) .....	3
Cuarta parte: Creación de tablas (20 min) .....	7
Quinta parte: explorar e interactuar con el mundo (15 minutos) .....	8

### Primera parte: creación del mundo y la misión (10 minutos)

1. Escoge un **contexto** en el que ocurrirá tu misión (una ciudad y sus habitantes, un sistema escolar con sus estudiantes, un zoológico, un museo, un gimnasio, un videojuego con personajes fantásticos, un planeta en el universo)
2. Escoge un **rol para la persona que se va a conectar a tu base de datos**: el vigilante del museo, el administrador del restaurante, el rey del castillo.
3. Piensa en una **misión de escritura**: “tienes que registrarte como el nuevo instructor del gimnasio y registrar una nueva máquina.”
4. Piensa en una **misión de consulta**: “tu misión es ubicar la máquina más costosa del gimnasio”.

### Segunda parte: modelado del mundo (20 minutos)

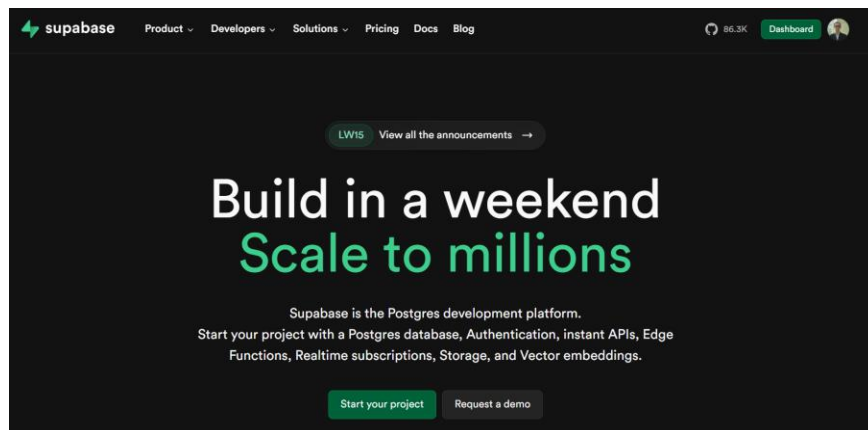
5. Piensa en **2 o máximo 3 entidades** para tu mundo. Piensa que sean relevantes para las misiones que le vas a asignar a tus usuarios.
6. Piensa en **los atributos** de estas entidades.
7. Piensa en las **relaciones** entre las entidades. Sus llaves foráneas y primarias.
8. Estructura el diagrama entidad relación. Utiliza cualquier herramienta que puedas compartir. Si quieres puedes usar <https://draw.io/> o cualquier otra (incluso power point). Procura tener claridad sobre los atributos y las llaves (¡puedes preguntar al/la profesor/a si tienes dudas!)



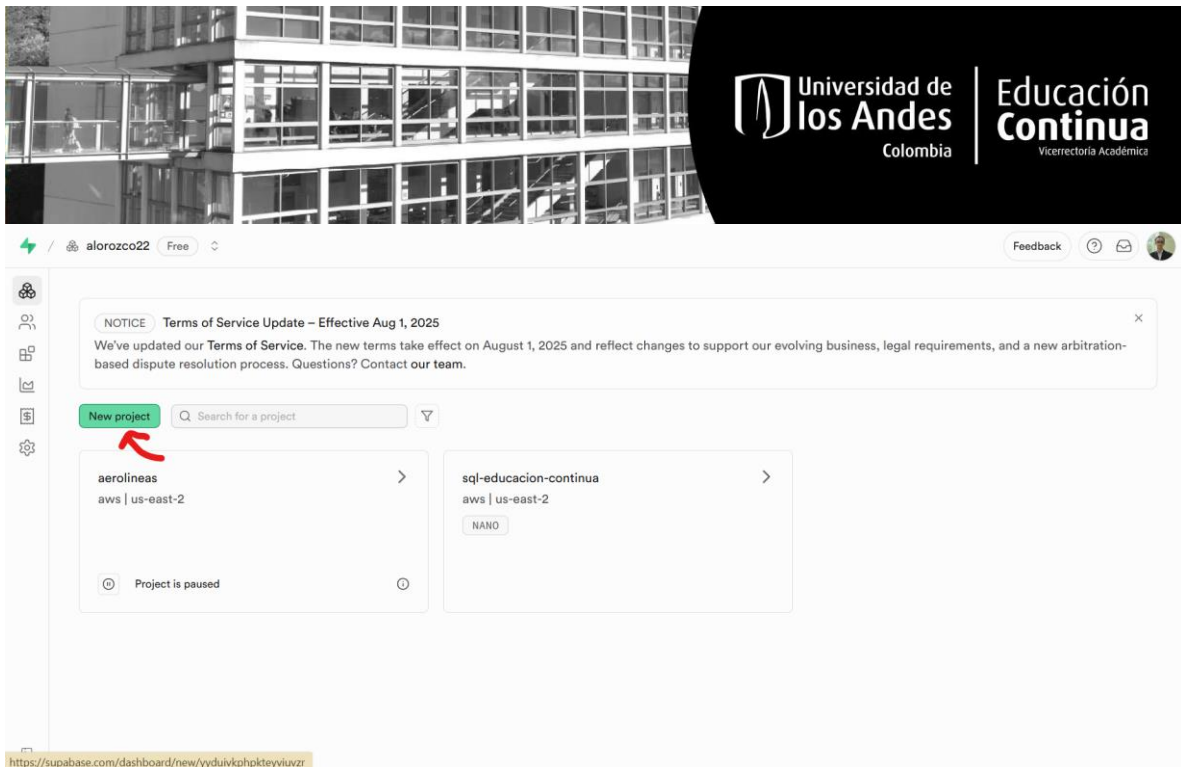
9. Guarda tu modelo, lo vamos a compartir después.

### Tercera parte: creamos un servicio de base de datos (10 minutos)

10. Para este proyecto vamos a crear una cuenta en <https://supabase.com/>. Ingresa al enlace y crea una cuenta.



11. Creamos un proyecto dando clic en Start your project, o bien en New Project en la plataforma de administrador:



12. Indicamos el nombre del proyecto y la contraseña. Piensen una **contraseña segura y anótenla**, la vamos a necesitar después.

The image shows the 'Create a new project' form in the Supabase dashboard. The form is titled 'Create a new project' and includes a sub-header: 'Your project will have its own dedicated instance and full Postgres database. An API will be set up so you can easily interact with your new database.' The form fields are: 'Organization' (dropdown menu showing 'alorozco22' and 'Free'), 'Project name' (text input field), 'Database Password' (text input field with a hint 'Type in a strong password' and a note 'This is the password to your Postgres database, so it must be strong and hard to guess. Generate a password'), and 'Region' (dropdown menu showing 'East US (North Virginia)' with a note 'Select the region closest to your users for the best performance.'). At the bottom of the form, there is a link 'SECURITY OPTIONS >'. The top of the dashboard shows the user 'alorozco22' and the 'New project' button.





Las demás opciones las podemos dejar como están y creamos el proyecto:

SECURITY OPTIONS ▾

What connections do you plan to use?

- ☒ Data API + Connection String  
Connect to Postgres via autogenerated HTTP APIs or the Postgres protocol
- ☐ Only Connection String  
Use Postgres without the autogenerated APIs

Data API Configuration

- ☒ Use public schema for Data API Default  
Query all tables in the public schema
- ☐ Use dedicated API schema for Data API  
Query allowlisted tables in a dedicated api schema

These settings can be changed after the project is created via the project's settings

ADVANCED CONFIGURATION ▾

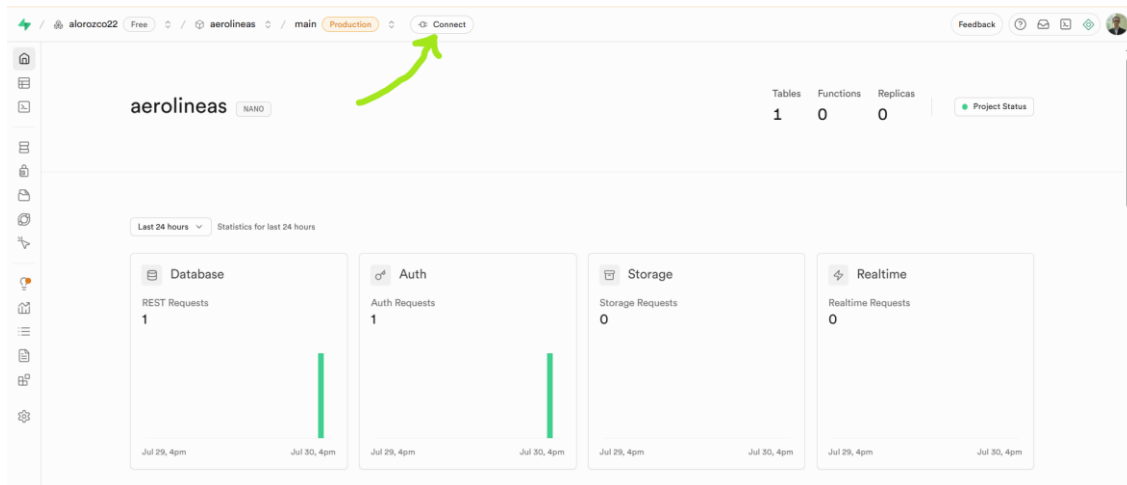
Postgres Type

- ☒ Postgres Default  
Recommended for production workloads
- ☐ Postgres with OrioleDB Alpha  
Not recommended for production workloads

These settings cannot be changed after the project is created

Cancel Create new project

13. Para conectarnos a la base de datos, vamos a la opción Conect:



14. Copiamos la cadena de conexión correspondiente a Session Pooler:



### Transaction pooler

Shared Pooler

Ideal for stateless applications like serverless functions where each interaction with Postgres is brief and isolated.

```
postgres://postgres.djorhrqoxrkfzftrh:[YOUR-PASSWORD]@
```

Does not support PREPARE statements

> View parameters

- Suitable for a large number of connected clients
- Pre-warmed connection pool to Postgres
- IPv4 compatible  
Transaction pooler connections are IPv4 proxied for free.

### Session pooler

Shared Pooler

Only recommended as an alternative to Direct Connection, when connecting via an IPv4 network.

```
postgres://postgres.djorhrqoxrkfzftrh:[YOUR-PASSWORD]@
```

> View parameters

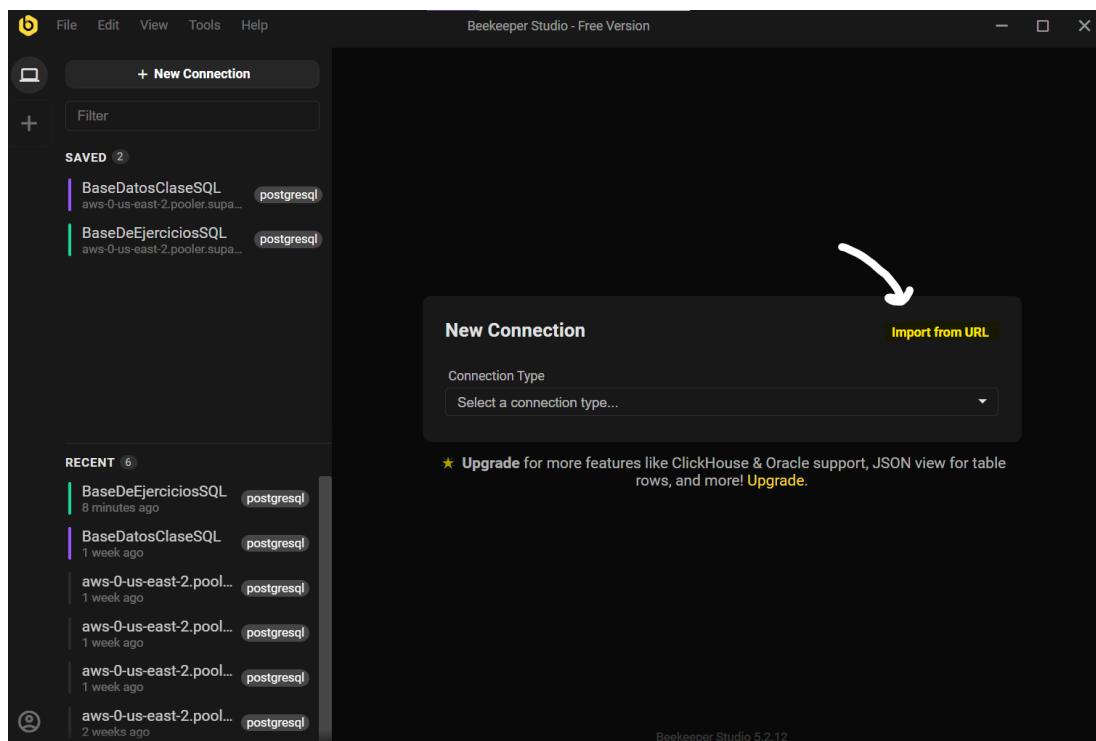
- IPv4 compatible  
Session pooler connections are IPv4 proxied for free
- Only use on a IPv4 network  
Use Direct Connection if connecting via an IPv6 network

Reset your database password

You may reset your database password in your project's [Database Settings](#)

**Importante:** reemplazar [YOUR-PASSWORD] con la contraseña de su base de datos.

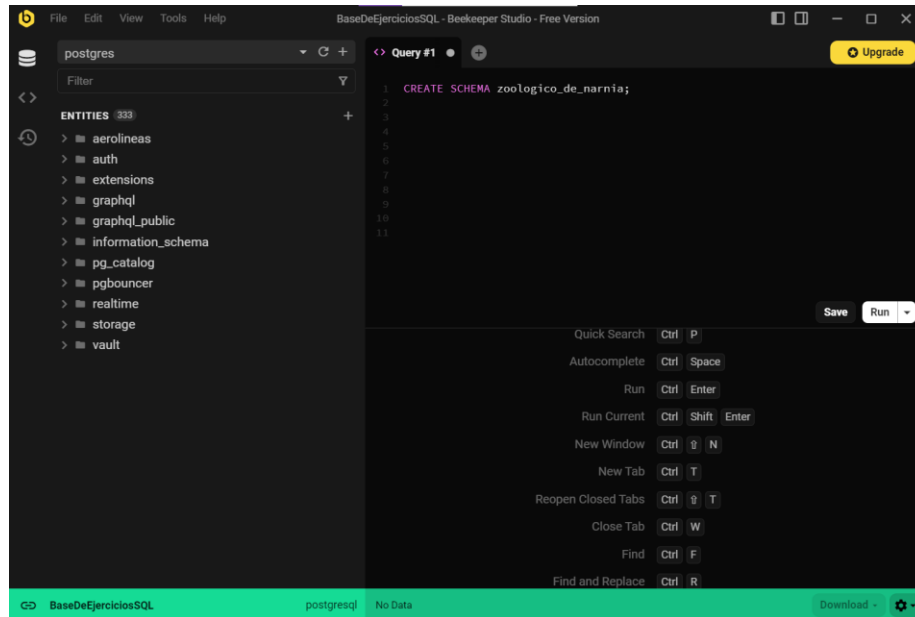
15. Conectarse a su base de datos con Beekeeper:



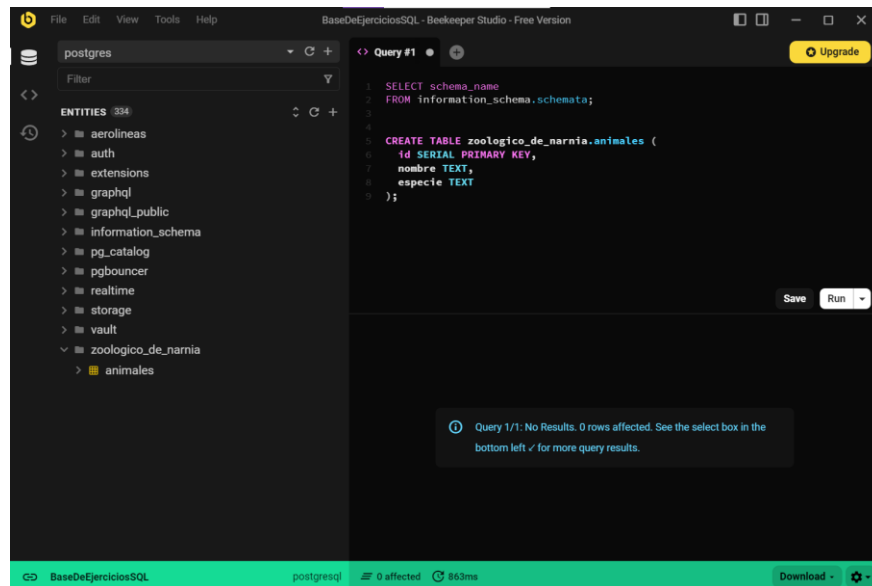


## Cuarta parte: Creación de tablas (20 min)

16. Una vez conectados, vamos a crear un **Schema** (que en este caso corresponde como a una carpeta de tablas en postgres).



17. Al crear el schema no va a aparecer en la parte izquierda hasta que creamos una tabla dentro del schema. **Importante:** al crear las tablas hay que referenciarlas dentro del schema. Por ejemplo: **zoologico\_de\_narnia.animales**



18. Finalmente crear las tablas con algunas restricciones.

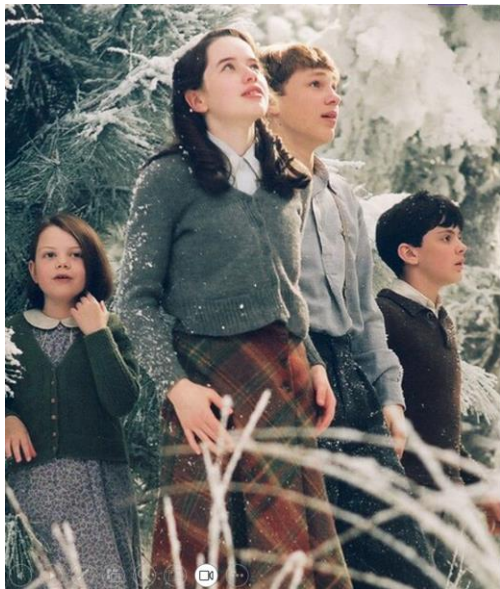
19. Insertar algunas filas en las todas las tablas para iniciar el mundo.

```
<> Query #1 • + Upgrade
1 INSERT INTO zoologico_de_narnia.animales (nombre, especie)
2 VALUES
3 ('Aslan', 'León'),
4 ('Reepicheep', 'Ratón'),
5 ('Trufflehunter', 'Tejón'),
6 ('Glenstorm', 'Centauro'),
7 ('Fledge', 'Caballo alado');
```

Save Run

## Quinta parte: explorar e interactuar con el mundo (15 minutos)

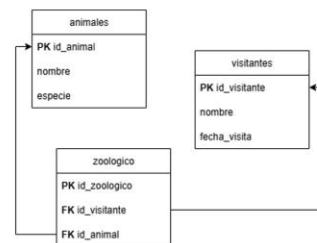
20. Compartimos nuestra misión. Usemos la plantilla de invitación para compartirla por el chat. Recuerden poner la cadena de conexión:



### El zoológico de Narnia

**Misión 1:** Tienes que registrar un nuevo animal en el zoológico, y si quieres registrarte como visitante del zoológico.

**Misión 2:** ubica el animal cuyo nombre termina con lan.



21. Conectémonos a mundos de nuestros compañeros y completemos un par de misiones.