

IIC2413 – Bases de Datos

Guía Ayudantía 1

1. Objetivo

- Mostrar como subir archivos al servidor
- Mostrar como crear una tabla y añadir una tupla a su base de datos
- Responder dudas

2. Ayudantía

2.1. Conexión inicial con el servidor y subir archivos

Lo primero que deben hacer es mostrar la wiki del curso, en la sección de **Tutorial del servidor y Base de Datos Postgresql**

Posteriormente, deben mencionar las diversas maneras que existen para poder conectarse al servidor, es decir, la existencia de putty y la posibilidad de utilizar directamente la consola tanto en Ubuntu como en Linux. Una vez hecho esto, deben conectarse al **Grupo999** cuya contraseña es grupo999, para esto pueden usar `ssh grupo999@codd.ing.puc.cl` en su terminal(en el caso de tener ubuntu o linux). Luego, deben mostrar con el comando `ls` que existe solamente la carpeta **Sites** al interior de su usuario, e indicar que aquello que deben hacer es **tener una carpeta para cada entrega** y mostrar como se crean dentro de la consola utilizando `mkdir nombreCarpeta`. OJO, deben dar énfasis que el formato es **EntregaX**.

Lo que deberían hacer a continuación es mostrar a través de Filezilla y Github como subir el archivo al servidor, para esto mostrar como se subiría un archivo como el enunciado de la entrega 1 o este mismo archivo.

Recuerden que deben recomendar github para el trabajo en grupo, dado que así evitan que se sobrescriban los archivos, para esto pueden crear un repositorio privado. Además, recomienden el uso de `.gitignore` para aquellos archivos del estilo `.DS_STORE` u otros.

2.1.1. Usando git clone

Este método **NO SE RECOMIENDA**, dado que se crea una carpeta del repositorio al interior del servidor y no se podría entonces seguir el formato especificado. La idea es que muestren lo que sucede y luego usen el comando `rm -R nombre_carpeta` para borrarlo

2.1.2. Usando git init

Este método es el indicado para poder usar git, para usarlo deben poner el **git init** y decir que aún no está conectado al repositorio privado que uno tiene, es por esto que deben conectarlo.

Para realizar la conexión, lo que deben hacer es usar el comando **git remote add origin direction**, donde direction es la dirección de su repositorio.

Recuerden que deben hacer **git pull origin master** en el servidor para actualizar la información que poseen al interior de este. Para setear que siempre se traigan los cambios de la rama master se debe hacer **git branch --set-upstream-to=origin/master master**.

2.2. Conectarse a Postgress y crear una tabla

Como ya explicaron como conectarse al servidor y subir archivos, ahora solo queda conectarse a la base de datos. Para realizar esto, deben mostrar que el comando a utilizar es **psql**, luego les preguntará su contraseña, la que sigue siendo grupoXX donde XX es el número del grupo.

Una vez conectados, deberían explicar algunos de los comandos que podrían ser útiles para el ramo como

- **\ dt** Sirve para poder ver todas las tablas existentes dentro de la base de datos
- **\ du** Sirve para poder ver a todos los usuarios existentes en el servidor
- **\ q** Se usa para salir del servidor
- **\ d table_name** Sirve para mostrar la información de una determinada tabla(sus columnas y eso)

Una vez hecho eso, deben mostrar como se crea una tabla e insertar valores, para esto usar los comandos **CREATE TABLE table_name(atr1 type1, atr2 type2);** e **INSERT INTO table_name VALUES(value1, value2)**

Además, pueden mostrar como se cambia la clave, que sería utilizando el comando **passwd**, y usar una clave de prueba como **123grupo999**

3. Preguntas frecuentes

1. ¿Cuál es el puerto de conexión con Filezilla?

El puerto 22

2. ¿Las tablas se crean en la carpeta del grupo?

No, se crean al interior de postgresql

3. ¿Cómo salgo del servidor?

Una de las manera comunes para salir del servidor es escribir exit en la consola

4. ¿Cómo hago un Join?

Para esta entrega no es necesario, pero el comando es **SELECT * from tabla1, tabla2;**

5. **¿Es necesario el ;?**
Al interior de postgres si lo es
6. **¿Cómo defino un PRIMARY KEY?**
Indicándolo al crear una tabla, es decir **CREATE TABLE table_name(atr1 type1 PRIMARY KEY, atr2 type2);**
7. **¿Si me equivoco al crear las tablas que hago?**
Usando el comando **DROP TABLE nombre_tabla**
8. **¿Puedo usar powershell?**
Si, powershell tiene ssh, que es lo necesario para conectarse al servidor
9. **¿Puedo crear tablas con filezilla?**
No, filezilla es solo para poder añadir archivos, para crear tablas deben conectarse a postgres a través del servidor
10. **No sé si soy un grupo con número par o impar, ¿dónde se puede ver?**
Se puede ver en la siguiente planilla o link https://docs.google.com/spreadsheets/d/1_wBBmV8DM2YLOBsur130nA8mLQx1_12ULE1MvBFBF7Y/edit?usp=sharing
11. **¿Cómo cambio de grupo par a impar?**
Debes encontrar un grupo que quiera cambiar de número también y mandar un mail a arpincheira@uc.cl informando el cambio
12. **¿A que se refieren en el enunciado cuando hablan de esquema?**
Nos referimos a la notación del tipo **Personajes(nombre, apellido, edad).**, lo ideal es que pongan el tipo de los atributos, para recibir un buen feedback, aunque esto último no será evaluado en esta entrega.
13. **En la parte del esquema, si una relación tiene más de una key posible (por ej, id y rut) al escribirlo tenemos que subrayar ambas o escogemos una arbitraria como principal?**
Si tu consideras que hay dos Primary Keys, debes subrayar ambas. Pero si ambas cumplen el objetivo de una primary key, entonces basta con escoger aquella que sea más fácil de implementar (por ejemplo ints sobre strings), ya que tener ambas es redundante.
14. **¿Se pueden agregar atributos adicionales a cada esquema?**
Se puede, pero se debe cumplir con los mínimos que se exigen por enunciado.
15. **Cuando hago git pull desde el servidor, a veces me se me abre nano pidiendo que escriba un mensaje para el 'merge' ¿Por qué sucede?**
Lo que pasa es que tienes cambios sin committear en el repo local y estas haciendo git pull para traer cambios del repositorio remoto.

Para evitar que esto suceda, lo que debes hacer es asegurarte que no tienes ningún archivo sin commitear (es decir, ningún archivo en rojo al hacer git status) y luego hacer git pull de forma limpia. Si te vuelve a suceder, basta con escribir el mensaje del merge (Ctrl+O) y luego git push.

16. **¿Pueden existir instancias dentro una tabla que tengan atributos con un valor nulo o todas deben tener un valor distinto de cero?**

La idea es que su esquema evite a toda costa que existan valores NULL dentro de sus tablas.

17. **Cuando en la terminal se me pide escribir la contraseña del servidor la consola no me deja escribir**

lo que probablemente sucede es que efectivamente estás escribiendo, pero por configuración del servidor, al momento de escribir no te aparece un 'registro visual' de lo escrito. Es por esto que te recomiendo que escribas tu contraseña igual e intentes ingresarla, aunque no la puedas ver.

18. **¿Al crear una tabla en la base de datos, tengo que distinguir cual es mi PRIMARY KEY?**

En esta primera entrega la idea es que se familiaricen con la base de datos, por lo que no es estrictamente necesario que denoten las primary Keys, de todas formas, para tratar de adelantar trabajo, sería ideal que probaran creando la tabla lo más realista posible e indicaran de antemano cual es su primary key.

19. **¿Es necesario crear la tabla en la base de datos con la importación de un .csv o podemos directamente crearla desde la terminal con los comandos de PSQL (CREATE TABLE)?**

Ambas maneras son válidas.

20. **¿Qué nombre debe tener el archivo pdf de la entrega?**

El archivo no debe tener ningún nombre en específico, mientras esté en la ruta solicitada, todo bien.

21. **¿Dónde se sube el pdf con la entrega?**

En pdf lo deben subir al servidor, dentro de una carpeta denominada Entrega1, en el caso de no estar creada esta carpeta, ustedes la pueden crear a través de la consola o importarla desde sus computadores usando filezilla o github.