Laboratorio 6 - Inyecciones SQL

Profesor: Sebastián Ferrada

P0. Conéctese a postgres para ver las tablas de este taller. En **transparencia** encontrará las remuneraciones brutas de todos los empleados de la Universidad desde enero de 2015. Escriba una consulta SQL para obtener los salarios mensuales de todos los empleados que tengan un apellido paterno elegido por usted, ordenados por monto.

En la tabla **notas** usted encontrará las notas finales de un curso de usted y sus compa˜neros (obviamente no tienen nada que ver con la realidad... *¿o sí?*). Escriba una consulta SQL que obtenga **su** nota final del ramo.

P1. En una celda de Colab, escriba un código que pida un apellido usando `input()`. Una vez que se ingrese, genere una query que entregue los diez sueldos más altos de empleados con ese apellido. Usted debe correr el código y verificar que funcione correctamente. Su código debe concatenar directamente el input con el string de consulta, no utilice sentencias precompiladas.

P2. Piense que usted es un simple usuario, por lo tanto, usted no tiene privilegios en la base de datos (es decir, usted no conoce el esquema ni los detalles de conexión); usted tampoco tiene acceso a leer o modificar el código del script. Todo su poder se basa en la capacidad de escribir cuando el programa le pide un apellido (En la vida real, esta situación puede darse a través de un formulario HTML donde usted no puede ver el código del servidor). Usted debe ejecutar la celda e ingresar un ataque de SQL injection para subir su nota registrada en notas.

Hints:

- **No** cambie nada dentro del código. Solo ingrese un SQL injection en la consola cuando le pidan un apellido.
- Si la celda arroja una excepción, no significa que su ataque fue infructuoso. ¿Acaso esperaba un mensaje de felicitaciones por hackear la base de datos?
- Revise en otra celda, usando el sql magic si su ataque fue efectivo.
- La base de datos requiere que la nota esté entre 1,0 y 7,0

P3. En otra celda, escriba la misma funcionalidad, pero esta vez, de forma segura ante SQL injections. Pruebe que al re-ingresar la inyección del problema anterior, esta vez su nota no cambia.