



IIC2115 – Programación como Herramienta para la Ingeniería (I/2020)

Laboratorio 4 - Bases de datos

Objetivos

- Aplicar los contenidos de bases de datos para crear un modelo de datos. Además, analizar y realizar consultas en base a la información almacenada.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L04**.
- **Entrega:** Domingo 31 de Mayo a las 23:59 hrs.
- **Formato de entrega:**
 - Archivo python notebook (**L04.ipynb**) con su código. Utilice múltiples celdas de texto y código para facilitar la revisión de su laboratorio.
 - Archivo python (**L04.py**) con el mismo código entregado en el notebook.
 - Todos los archivos deben estar ubicados en la carpeta **L04**. No se debe subir ningún otro archivo a la carpeta. Los archivos **.ipynb** y **.py** deben contener la misma solución. No incluya las bases de datos en esta entrega.
- **Descuentos:** El descuento por atraso se realizará de acuerdo a lo definido en el programa del curso. Además de esto, tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Laboratorios con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

- Si su laboratorio es entregado fuera de plazo, tiene hasta el **Lunes 1 de Junio a las 23:59 hrs** para responder el formulario de **entregas fuera de plazo** disponible en el Syllabus.
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.
- El uso de librerías externas que sean estructurales en la solución de los problemas no podrán ser utilizadas. Solo se podrán utilizar las que han sido aprobadas en las *issues* de GitHub.

Introducción: En búsqueda de una herramienta urbana

El Gobierno busca contar con una herramienta que le ayude a comprender la estructura y distribución del uso del suelo en las comunas de Chile. El uso del suelo hace referencia a cómo se utiliza el espacio en los distintos territorios. Con el objetivo de agilizar la creación de la anhelada herramienta, el Gobierno ha creado un concurso en el que espera participen los mejores programadores del país. Cada participante debe inscribirse mostrando la herramienta que ha creado y luego se decidirá quién es el ganador en base a la calidad de dicha herramienta. Para el desarrollo del concurso, el Gobierno ha dispuesto de una serie de datos que ha recopilado por medio de SII y que serán entregados a cada participante.

Usted ha visto el anuncio y ha decidido participar. Para ello, deberá comprender cómo son los datos que el Gobierno ha dispuesto y cumplir una serie de misiones. Cada misión agregará una funcionalidad extra a la herramienta y se espera que al postular esta herramienta cubra todas las misiones.

Los datos del SII

El Gobierno ha dispuesto de una serie de datos urbanos en dos tablas, las que se encuentran en un archivo SQLite llamado “**datos_sii.db**”. A continuación, se define cada tabla y se describen sus respectivas columnas. Previo a entender las tablas, es necesario saber que el SII maneja la información por medio de Manzanas, Predios, Líneas de construcción y Roles.

- **Manzana:** unidad espacial definida por un polígono. Toda comuna de Chile se encuentra subdividida en manzanas.
- **Predio:** unidad espacial contenida dentro de una manzana. Esta se encuentra definida por un código o rol único. Esta puede estar edificada o no.
- **Línea de construcción:** corresponde a una edificación o construcción. Esta se encuentra contenida dentro de un predio. Dentro de un predio puede existir más de una línea de construcción.

- **Rol:** código que maneja el SII para identificar un predio en específico. Esta compuesto por el código de la manzana seguido del código del predio. Por ejemplo, si estamos hablando del predio 5 de la manzana 23, su rol corresponde a 23-5. Este código es único dentro de una comuna, por lo que para una comuna en específico no existen dos roles iguales.

TABLA 1: Información Roles no agrícolas

Esta tabla se encuentra dentro del archivo “datos_sii.db” con el nombre “Predio” y contiene la información de todos los predios para las comunas de Chile. Las columnas de cada registro son las siguientes:

1. **comuna:** código SII de la Comuna
2. **manzana:** número de Manzana
3. **predio:** número de Predio
4. **direccion:** dirección o nombre del predio
5. **avaluo_fiscal:** avalúo fiscal total
6. **contribucion_semestral:** contribución semestral (con aseo)
7. **destino:** código de destino principal. Este corresponde al tipo de uso de suelo utilizado principalmente en ese predio.
8. **avaluo_exento:** avalúo exento de la propiedad
9. **superficie_m2:** superficie total del terreno de la propiedad (sin decimales, en m2)

TABLA 2: Información de terrenos y construcciones no agrícolas

Esta tabla se encuentra dentro del archivo “datos_sii.db” con el nombre “Linea” y contiene la información de todas las líneas de construcción para las comunas de Chile. Las columnas de cada registro son las siguientes:

1. **comuna:** código SII de la Comuna
2. **manzana:** número de Manzana
3. **predio:** número de Predio
4. **linea_construccion:** número correlativo de la línea de construcción. Comienza desde 1 en adelante.

5. **material_estructural:** código del material estructural de la línea de construcción.
6. **calidad_construccion:** código de calidad de la línea de construcción
7. **ano_construccion:** año de la línea de construcción
8. **superficie_m2:** superficie de la línea de construcción (sin decimales, en m2 o m3 según tipo de construcción)
9. **destino:** código de destino de la línea de construcción. Este corresponde al tipo de uso de suelo con que fue categorizada la línea de construcción.
10. **condicion:** código de condición especial de la línea de construcción

Diccionarios para tablas 1 y 2

Al visualizar los datos de las tablas 1 y 2, notaran que algunas columnas almacenan la información mediante códigos. Por ejemplo, en la TABLA 2, la columna “calidad_construccion” maneja números de 1 a 5 donde cada número tiene un significado. Para saber el significado de cada número usted cuenta con un archivo “.csv” con el nombre de la columna (“calidad_construccion.csv” en este ejemplo). En él, se almacena un diccionario para conocer los significados de cada columna de este tipo.

Los archivos “.csv” de diccionarios disponibles son los siguientes:

- comuna.csv
- destino.csv
- material_estructural.csv
- calidad_construccion.csv
- condicion.csv

Prepare su entorno de trabajo

Este apartado te guiará en dejar todo listo antes de comenzar. Descargue la carpeta “datosL04” desde <https://bit.ly/2T08n0q>. El link abrirá una carpeta de Dropbox web, presione “Direct Download” para descargar todo su contenido. La carpeta “datosL04” contiene la base de datos con las tablas descritas y los archivos “.csv” mencionados. Además, junto a la carpeta “datosL04” vendrá un archivo de nombre “git.gitignore”. Una vez descargado, deben hacer lo siguiente:

- Agrega el archivo “git.gitignore” a tu repositorio privado (junto a tus carpetas de laboratorio). Una vez hecho esto, al abrir tu repositorio deberías ver las carpetas L01, L02, L03 y este nuevo archivo “git.gitignore”.
- Ahora vamos a renombrar el archivo “git.gitignore”. Esto se hará diferente dependiendo el sistema operativo:
 - **Windows:** Renombra el archivo “git.gitignore” a “.gitignore.” (un punto delante y uno al final). Si te dice que ya existe, simplemente reemplázalo. El archivo quedará como “.gitignore” (sin el punto al final, eso queremos).
 - **macOS:** Presiona las tres teclas “**command+shift+.**” (todas juntas, la última es la tecla punto). Luego, renombra el archivo “git.gitignore” a “.gitignore” (acepta cualquier ventana). Si te dice que ya existe, simplemente reemplázalo.
- Finalmente *commitea* tus cambios como de costumbre con: `git add *`, `git commit -m "agregando el gitignore"`, `git push`. Lo que acabas de hacer fue decirle a GitHub que no quieres que se suban las bases de datos, así no tendrás errores al *commitear* tu trabajo. Si eres usuario macOS, puedes volver a apretar la combinación de teclas para dejar todo como antes (**command+shift+.**), ya no veras el archivo “.gitignore”.
- Ahora crea la carpeta “L04” dentro de tu repositorio. Dentro de la carpeta “L04” guarda la carpeta “datosL04” que descargaste del link. Ahora puedes crear tu Jupyter Notebook dentro de la carpeta “L04” (no lo crees dentro de “datosL04” porque esta carpeta no se va a sincronizar con GitHub) y comenzar a trabajar. CONSEJO: Le recomendamos dejar una copia de la carpeta “datosL04” en su computador. Si mientras desarrollan el laboratorio hacen algo mal, simplemente la pueden volver a reemplazar.

Misiones para la herramienta

Para completar requerimiento solicitado por el Gobierno, usted deberá crear una herramienta que sea capaz de realizar las tareas descritas en cada una de las misiones siguientes:

- M1. **Conociendo los datos y depurando:** Su primera misión será inspeccionar las bases de datos del gobierno y revisar los tipos de datos. Para ello conéctese a la base de datos creando una conexión:

```
connection = sqlite3.connect('datosL04/datos_sii.db')
```

- (a) Averigüe como visualizar el *schema* de una base de datos mediante SQL. Utilícelo para conocer los tipos de datos que posee cada columna de la base de datos (0.1 puntos)
- (b) Visualice dos filas de cada tabla mediante consultas SQL. (0.1 puntos)
- (c) Ahora que ha visto los datos y los tipos de datos. Modifique las tablas mediante SQL para cambiar el tipo de dato que no corresponda y que le facilite el posterior desarrollo de su herramienta. Indique sus cambios en una celda de texto. Además defina cuáles van a ser sus **PK** por tabla. (0.3 puntos)

M2. Integrando los diccionarios: Ya tenemos las tablas en el formato correcto, ahora toca incorporar los diccionarios a nuestras bases de datos.

- (a) Para cada archivo de diccionario “.csv” cree una tabla dentro de la base de datos mediante el uso de SQL. Sea cuidadoso en el tipo de dato que escoge en cada columna e indique cuáles serán sus **PK**. (0.3 puntos)
- (b) Lea los datos de los archivos “.csv” y cárguelos en las tablas correspondientes mediante SQL. (0.3 puntos)

M3. Verificando consistencia: Realice 5 verificaciones (una por celda) mediante consultas en SQL para asegurarse que la base de datos es consistente. De encontrar inconsistencias, depúrelas bajo algún criterio o elimínelas de la base de datos. Un ejemplo es asegurarse que los datos de la columna “superficie_m2” sea mayor que 0. Enuncie las verificaciones que va a realizar en una celda de texto. (0.5 puntos)

M4. Realizando algunas consultas (nivel fácil): Confeccione una única consulta en SQL que retorne una tabla con lo pedido en los siguientes ítem. Entiéndase como una tabla a una lista de tuplas con datos de solo las columnas pedidas en cada ítem.

- (a) Una tabla con el código comuna, manzana, predio y superficie para todos los predios que sean mayores a 700 m^2 . (0.3 puntos)
- (b) Una tabla con el nombre de la comuna y el promedio del avalúo exento (no considere los avalúo exento igual a cero dentro del promedio) por comuna. (0.3 puntos)
- (c) Una tabla con el nombre de la comuna y el número de líneas de construcción que sean con destino “Habitacional”, tengan superficie par y se hayan construido posterior al 2000. (0.4 puntos)

M5. Realizando algunas consultas (nivel difícil): Confeccione una única consulta en SQL que retorne lo pedido en los siguientes ítem:

- (a) Para una comuna en específico (a elección) identifique, mediante rol y número de línea de construcción, las líneas de construcción de destino “Habitacional” que compartan predio con líneas de construcción de destino “Oficina”, que pertenezcan a predios que tengan al menos 3 líneas de construcción y todas sean posteriores a 1980. (0.7 puntos)
- (b) Identifique todos los predios por nombre de comuna y rol que tengan al menos 2 construcciones con código de destino diferente (basta con que hayan 2 códigos de destino diferentes entre las líneas de construcción), una de las líneas de construcción tenga una superficie mayor a 200 m^2 y otra tenga una superficie menor a 100 m^2 . (0.7 puntos)

M6. Completando la tabla comunas: Ahora que ya ha practicado haciendo consultas. Su objetivo será completar la tabla de comunas e incorporar las siguientes columnas usando exclusivamente SQL:

- (a) avaluo_promedio: promedio de avalúo exento por comuna sin considerar los ceros. (0.4 puntos)
- (b) n_lineas_de_calidad: conteo de lineas de construcción por comuna que sean de calidad “Superior” y el avalúo exento del predio al que pertenecen sea mayor que el promedio de avalúo exento de la comuna. (0.8 puntos)

M7. Creando una función genérica: Su última misión consiste en crear una función en Python que reciba el nombre de una comuna y un código de destino (ambos como texto). A partir de ello, y usando una única consulta en SQL, retorne una tabla que incluya Manzana, Predio y número de líneas de construcción del destino ingresado para la comuna ingresada. Su respuesta debe estar ordenada de mayor a menor por el número de líneas de construcción encontrado. (0.8 puntos)

Corrección

Para la corrección de este laboratorio, se revisarán las consultas construidas para responder las diferentes misiones propuestas y la estructura de como utiliza Python en ellas. Además, se revisará el diseño y construcción de las tablas de su base de datos.

Política de Integridad Académica

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los

demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.