



IIC2115 – Programación como Herramienta para la Ingeniería (II/2019)

Laboratorio 4 - Bases de datos

Objetivos

- Entender la estructura y funcionamiento de bases de datos en SQL.
- Aprender a confeccionar consultas SQL para extraer información de una base de datos.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L04**.
- **Entrega:** domingo 20 de octubre a las 23:59 hrs.
- **Formato de entrega:** archivo python notebook (**.ipynb**) y archivo python (**.py**) con la solución de este enunciado. Los archivos deben estar ubicados en la carpeta **L04**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea. Los archivos **ipynb** y **py** deben contener la misma solución.
- **Descuentos:** se descontará 0.5 puntos por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Si su laboratorio es entregado fuera de plazo, tiene un hasta el **lunes 21 de octubre 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.

Introducción

El estudio de los viajes de Santiago

El constante aumento del número de habitantes y la tasa de motorización en la ciudad de Santiago esta generando un sin fin de problemas. Uno de los más grandes esta relacionado con la movilidad que enfrentan los capitalinos. Los altos niveles de congestión que enfrentan los habitantes a la hora de viajar ha llegado a niveles históricos.

Dada esta problemática, el Ministerio de Transportes y Telecomunicaciones (MTT) realiza cada cierto tiempo la Encuesta Origen Destino de Viajes 2012 de Santiago (EOD-Santiago). Esta corresponde a la base de datos que contiene el patrón de viajes de los habitantes de la ciudad, información fundamental para la formulación de planes y proyectos de transporte. De este modo, poder comprender de mejor forma la ciudad y planificar adecuadamente el futuro de Santiago.

Lamentablemente, la última encuesta realizada fue el año 2012. Siete años después, el MTT desea realizar un nuevo estudio utilizando esta información. Para esto, ha recurrido a los alumnos del ramo Programación como herramienta para la ingeniería, con el objetivo que puedan desarrollar los objetivos planteados.

EOD-Santiago 2012

La EOD cuenta con una serie de Tablas que incluyen información relevante de viajes realizados en la ciudad de Santiago. También existe información relevante respecto de los hogares y personas asociadas a cada viaje. En la lista de a continuación, se incluye el nombre de cada Tabla en la base de datos y una breve descripción de su contenido.

Base de datos y Tablas de la EOD

La base de datos de la EOD se encuentra en el archivo **eod.db** disponible en la carpeta L04. Este archivo de base de datos contiene la siguientes tablas:

1. **Hogares:** En esta tabla se almacena toda la información referente a los hogares que participan en la encuesta. Entiéndase por hogar, al conjunto de personas que viven en un misma vivienda. Se puede encontrar información asociada a localización, ingreso del hogar, número de vehículos, bicicletas, etc.
2. **Vehículos:** En esta tabla se describen los vehículos presentes en los hogares de la encuesta. Incorpora información respecto al tipo de vehículo, marca, año, etc.

3. **Personas:** En esta tabla se almacena toda la información referente a las personas que participan de la encuesta. Se puede encontrar información asociada al hogar al que pertenecen, sexo, ingreso, ocupación, lugar de trabajo o estudios, etc.
4. **EdadPersonas:** Esta es una pequeña tabla que solo contiene información de la edad de las personas que participan en la encuesta.
5. **Viajes:** En esta tabla se almacena toda la información referente a las viajes encuestados por la EOD. Cada viaje es único y es realizado por una única persona (si dos personas viajan juntas, entonces habrán dos viajes). Se puede conocer la persona que realiza el viaje, su origen, destino, propósito, etc.
6. **DistanciaViaje:** Esta es una pequeña tabla que solo contiene información asociada a la distancia de cada viaje y si este fue imputado o no.
7. **ViajesDifusion:** En esta tabla se almacena toda la información referente al modo general utilizado en el viaje.
8. **Etapas:** En esta tabla se almacena toda la información referente a las etapas de los viajes encuestados por la EOD. Los viajes, presentes en la tabla Viajes, están compuestos por una o más etapas de viaje. Estas corresponden a las distintas fases realizadas por la personas para realizar un viaje. Se puede conocer el origen y destino de la etapa, así como también el tiempo de viaje, modo empleado, entre otras.

Tablas de parámetros

El detalle de las columnas de cada tabla de la EOD la puede consultar dentro de la base de datos. Para entender el significado de algunas columnas, son necesarias una serie de tablas de parámetros que se dispone en la carpeta de este laboratorio. Por ejemplo, dentro la tabla **Viajes** existe una columna **ActividadDestino** que representa la actividad que desarrolla en el destino del viaje. Dentro de los datos de esta columna, solo hay enteros 1 y 8. Para saber que significa cada número, debemos buscar el archivo **ActividadDestino.csv** dentro de las tablas de parámetros y encontraremos los significados de cada valor.

Misiones del MTT

Para completar requerimiento solicitado por el MTT usted deberá implementar una serie de misiones descritas más abajo. Sea ordenado y claro. Además sea riguroso en el uso de SQLite y Python según se indica en cada misión.

Reglas de las misiones

Las misiones deberán ser respondidas utilizando dos tipos de herramientas. Una de ellas, será por medio de **SQLite**, es decir, deberá utilizar únicamente consultas de base de datos para llegar a la solución. La otra será **Python**. En esta última, deberá acceder a la información mediante SQLite y luego deberá responder a la pregunta utilizando lógica con Python.

#Ejemplo de solución con SQLite

var = #consulta con SQLite que resuelve la pregunta

#Ejemplo de solución con Python

var = #consulta con SQLite que obtiene datos

#Lógica con Python

resp = #respuesta encontrada

Misiones

Cuando entregue su tarea procure **NO SUBIR LA BASE DE DATOS A GITHUB**

- M1. Dado que has aceptado el trabajo impuesto por MTT, su primera misión será comprender correctamente los datos, es decir, saber qué columnas contiene cada tabla. Para ello debes utilizar la consulta `pragma table_info(NombreTabla)`. Investigue acerca de esta función para facilitar su uso. Estudie las columnas de cada tabla, le será de ayuda para misiones futuras. También deberá construir una consulta para determinar el número de filas de cada tabla. El objetivo de esta misión será entregar una lista de listas. Cada Sub-lista debe ser `['nombre tabla', nFilas, nColumns]` (0.4 puntos)
- M2. Es hora de realizar la primera consulta a los datos. Construya una única consulta en **SQLite** que entregue todos los hogares que posean 3 o más habitantes. (0.2 puntos)
- M3. Siempre que se trabajan con datos de fuentes externas, es necesario realizar una verificación y depuración de los datos. Su nueva misión será realizar consultas en las tablas individuales y detectar posibles errores o inconsistencias. Por ejemplo, una distancia o tiempo negativo, entre otras. Su misión será reportar 3 inconsistencias. Para ello, entregue una celda de texto **ordenada** donde identifique, la tabla, columna que posee la inconsistencia y describa **brevemente** cuál fue la inconsistencia detectada. (0.5 punto)
- M4. Al finalizar la primera misión, se ha percatado que la base de datos fue cargada con un pequeño error. Todos los tipos de datos de las tablas son TEXT. Esto no está bien, pues a hay otros de carácter

numérico (entero o decimal). Su misión será investigar cómo es posible cambiar el tipo de dato de una columna y corregirlas. Su criterio es esencial para responder esta pregunta. (0.5 puntos)

M5. Ahora que estas familiarizado con los datos. Toca realizar algunas consultas de la información de la EOD, para cada una de ellas use **exclusivamente SQLite**:

(a) Construya una única consulta que entregue a todas las personas que participaron de la EOD, sean hombres y que posean licencia de conducir. Cree una nueva tabla de nombre **HombresConductores** con la respuesta de esta pregunta. (0.25 puntos)

(b) Tomando como base la respuesta obtenida de la parte (a). Construya una nueva consulta que entregue las personas que sean mayores de 30 y menores de 45. (0.25 puntos)

M6. Es hora de entender los viajes que se realizan. En esta pregunta debe utilizar tanto **Python** como **SQLite** para responder. Defina una clase **Persona** que posea un entero de “rut” (id de la persona en la EOD), entero de “edad”, texto de “sexo” (‘M’ o ‘F’) y texto del nombre de la “comuna de residencia”. Luego, construya una función que reciba como parámetros “distancia de viaje” y “modo”. Esta debe retornar una lista de objetos Persona, que hayan realizado algún viaje en el modo señalado y de una distancia Euclidiana mayor o igual a la indicada. (0.4 puntos)

M7. Redefina (en otra celda posterior), una nueva versión de la clase Persona, esta vez, debe incluir (además de los campos anteriores) los campos: personas en el hogar, bicicletas en el hogar y vehículos en el hogar. Luego, construya una función que reciba como parámetros “sexo” (‘M’ o ‘F’) y “Edad”. Esta debe retornar una lista de todas las Persona con todos los campos cubiertos. Solo puede utilizar **Python** para convertir los datos obtenidos de la consulta en objetos Persona y construir la lista de respuesta. (0.5 punto)

M8. Vuela a redefinir por última vez (en otra celda posterior), una nueva versión de la clase Persona, esta vez, debe incluir (además de los campos de la clase Persona de la misión 6) los campos: número total de viajes y distancia total recorrida. Luego, construya una función que reciba como parámetro una “comuna”. Retorne una lista de todas las personas que viven en esa comuna y los objetos Persona deben tener todos los campos cubiertos. Solo puede utilizar **Python** para convertir los datos obtenidos de la consulta en objetos Persona y construir la lista de respuesta. (0.6 puntos)

M9. Construya una consulta que le permita conocer a nivel comunal, el número de viajes que se generan en hora punta mañana (6 am - 9 am) y en hora punta tarde (5pm - 8pm) por distintos propósitos. Para ello defina una función que reciba el nombre de comuna y un propósito de viaje. Esta debe retornar

una lista de listas donde cada Sub-lista es ['comuna', nViajesAM, nViajesPM]. Solo puede utilizar **Python** para convertir los datos obtenidos de la consulta. (0.7 puntos)

M10. Construya un ranking que muestre los 10 pares OD de comunas, con mayor cantidad de viajes. Para eso, debe determinar la cantidad de viajes que se realizan entre comunas. Muestre las 10 que más presentan viajes (investigue el uso de `limit`). Esta debe retornar una lista de listas donde cada Sub-lista es ['comunaOrigen', 'comunaDestino', nViajes]. Solo puede utilizar **Python** para convertir los datos obtenidos de la consulta. (0.7 puntos)

M11. Construya una consulta que retorne un ranking de los Hogares que más utilizan transporte público en días laborales. Se define uso de transporte público para un hogar, como la cantidad de total de kilómetros recorridos exclusivamente en transporte público por sus habitantes. Es importante revisar el detalle de las etapas de cada viaje dado que un viaje puede ser realizado en más de un modo. Liste los ids de los 10 hogares más altos en el ranking y la cantidad de kilómetros que recorren en la encuesta. Esta debe retornar una lista de listas donde cada Sub-lista es [idHogar, kmTPub]. Solo puede utilizar **Python** para convertir los datos obtenidos de la consulta. (1 punto)

Corrección

Para la corrección de este laboratorio, se revisarán las respuestas entregadas para cada una de las misiones. Utilice múltiples celdas para facilitar su corrección. Sea ordenado y claro en presentar sus respuestas.

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.