Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 – Programación como Herramienta para la Ingeniería (II/2022)

Laboratorio 4

Aspectos generales

- **Objetivo:** evaluar individualmente el aprendizaje sobre Bases de Datos mediante el uso del lenguaje SQL, mediante una Base de Datos con información de encuestas de salud mental en el campo de computación.
- **Lugar de entrega:** jueves 1 de Diciembre a las 23:59 hrs. en repositorio privado.
- **Formato de entrega: ÚNICAMENTE** el archivo Python Notebook (**L4.ipynb**) con la solución del laboratorio. El archivo debe estar ubicado en la carpeta **L4**. Es requerimiento de formato el utilizar múltiples celdas de texto y código para la construcción de la solución. Laboratorios que no cumplan el formato de entrega tendrán un descuento de 0,2 ptos de la nota final.
- **Entregas atrasadas:** El descuento por atraso se realizará de acuerdo a lo definido en el programa del curso. Si su laboratorio es entregado fuera de plazo, tiene hasta el **viernes 2 de Diciembre a las 11:59 AM** para responder el formulario de **entregas fuera de plazo** disponible en el Syllabus.
- **Issues:** Las discusiones en las *issues* del Syllabus que sean relevantes para el desarrollo del laboratorio, serán destacadas y se considerarán como parte de este enunciado. Así mismo, el uso de librerías externas que solucionen aspectos fundamental del laboratorio no podrán ser utilizadas. Solo se podrán utilizar las que han sido aprobadas en las *issues*, previa consulta de los estudiantes.
- **Laboratorios con errores de sintaxis y/o que generen excepciones en todas las ejecuciones serán calificados con nota 1.0.**

Introducción

Con el fin practicar los contenidos de Bases de Datos y SQL, en este ejercicio deberá leer una base de datos y crear diferentes consultas referentes a un encuesta sobre salud mental en relación a trabajos relacionados a tecnología. Esto incluye crear nuevas relaciones, modificar sus filas y especialmente, hacer consultas sobre sus datos.

En este laboratorio, tendremos dos grandes misiones, conformadas por múltiples desafíos. Cada misión cuenta con su propio **Bonus**. La primera misión se relaciona a explorar elementos generales de las preguntas. La segunda, se relaciona con un análisis más profundo de las edades y género de las personas encuestadas.

Es obligatorio entregar su Laboratorio dividido en diferentes celdas de código e incluir celdas de texto explicando sus consultas y cómo logran los objetivos.

Datos

En el archivo `salud_mental_Tech.zip` hay un archivo `mental_health.sqlite` que alberga tres relaciones. La primera es `Survey`, que contiene información sobre las encuestas que se han hecho. La segunda relación, `Question`, corresponde a información de las preguntas que se han hecho en las encuestas. Por último, `Answer` guardan información de en las respuestas que se han hecho a las preguntas de cada encuesta.
Sus esquemas son:

```
Survey (PRIMARY KEY INT SurveyID, TEXT Description)
```

```
Question (PRIMARY KEY QuestionID, TEXT QuestionText)
```

```
Answer (FOREIGN KEY SurveyID, INT UserID, FOREIGN KEY QuestionID, TEXT AnswerText, PRIMARY
    KEY (SurveyID, UserID, QuestionID))
```

A menos que se indique lo contrario, no deberá modificar el esquema de las tablas ni crear tablas nuevas.

1. Misión 1: Exploración General (5 pts)

En las primeras misiones vamos a hacer consultas sobre todas la preguntas. El objetivo es observar el comportamiento de las respuestas entregadas. Recuerde usar celdas de texto para explicar las consultas y facilitar la corrección.

1.1. Desafío 1

Para comenzar, revisaremos la cantidad de personas que contestaron cada encuesta. El resultado debe incluir la descripción de las encuestas, el año y el total de respuestas. Ordene por año.

1.2. Desafío 2

Para explorar las preguntas que existen, busque todas las preguntas que contengan la palabra "Describe". Muestre el ID de la pregunta y su texto. Hint: busque en internet sobre el comando `LIKE` de SQL.

1.3. Desafío 3

Siguiendo nuestro análisis, queremos tener más información respecto a la encuesta y tener una noción de si las preguntas eran abiertas o de alternativas. Para eso, encuentre para cada pregunta, la cantidad de respuestas distintas que tiene. El resultado debe mostrar el texto de la pregunta y la cantidad de respuestas distintas.

Hint: revise el uso del comando `DISTINCT`.

1.4. Desafío 4

Avanzando en la misma línea, ahora cuente para cada pregunta y cada respuesta, la cantidad de registros que hay. Luego, muestre la tabla resultante con columnas para el texto de la pregunta, el texto de la respuesta y la cantidad de registros.

1.5. Desafío 5

Ahora que ya hemos contado para cada pregunta y respuesta sus ocurrencias, queremos obtener sólo la respuesta más frecuente para cada pregunta. Filtre sus resultados para mostrar sólo las preguntas con ID en esta lista: [1, 2, 3, 5, 6, 20, 52, 89, 118]. La tabla final debe mostrar el texto de la pregunta, el texto de la respuesta y la frecuencia asociada.

BONUS: SQL + Pandas y Graficar Resultados (0.5 pts)

Busque en internet cómo ejecutar Queries SQL usando `Pandas`. Similar al desafío 4, encuentre para la pregunta de ID igual a 6 la frecuencia de respuesta que tiene cada alternativa. La tabla final debe mostrar el texto de la pregunta, el texto de la respuesta y la frecuencia. Debe obtener el `DataFrame` asociado a esa consulta.

Para finalizar, debe hacer un gráfico de barras que muestre el texto de la pregunta como título, el texto de las respuestas como categorías y que los datos sean la frecuencia. El gráfico lo puede hacer con `Pandas` o `Matplotlib`.

2. Misión 2: Análisis Edades y Género (6 pts)

Para estas misiones nos vamos a enfocar en la pregunta que pide la edad para nuestro análisis. A medida que avancemos, vamos a relacionarla con la pregunta: *"What is your gender?"* (¿Cuál es tu género?) para estudiar el perfil de las personas que respondieron la encuesta. Recuerde usar celdas de texto para explicar las consultas y facilitar la corrección.

2.1. Desafío 1

Al trabajar con encuestas, es muy común que existan valores inválidos. Se le pide crear una consulta SQL que elimine las filas de las respuestas que tengan valores inválidos para la edad, por ejemplo: valores negativos. Determine de forma personal y describa su criterio de eliminación. Para finalizar, muestre la tabla resultante e indique cuántos registros se perdieron.

2.2. Desafío 2

Busque en internet sobre el comando `LIKE` de SQL. Use este comando en una Query para encontrar el ID de la pregunta que pide la edad de la persona encuestada. Finalmente, indique cuál es el ID de esta pregunta. Haga lo mismo para la pregunta del género. Las preguntas están en inglés así que use *age* para la edad y *gender* para el género.

2.3. Desafío 3

Para comenzar, obtendremos la edad promedio de todas las personas encuestadas. Muestre este valor acompañado del texto de la pregunta. Segundo, busque el promedio de edad de las personas encuestadas de acuerdo al año en que se hizo la encuesta. Deben mostrar el texto de la pregunta, más el año, acompañado por el promedio asociado. **Hint:** recuerde revisar la lista de funciones de agregación de SQLite.

2.4. Desafío 4

Respecto al género de las personas encuestadas, muestre la tabla con todas las respuestas únicas que existen usando el comando `DISTINCT`. Luego, encuentre las 7 respuestas más frecuentes a esta pregunta. Muestre esta tabla resultante y arme una lista con estos valores.

2.5. Desafío 5

Utilizando la lista de géneros del desafío anterior, encuentre el promedio de edad para cada uno de los géneros. La tabla resultante debe tener una columna que indique el género de nombre "Genero" y una columna llamada "Edad_Promedio"para el promedio de edad pedido.

2.6. Desafío 6

Utilizando sólo una consulta, cuente la cantidad de personas encuestadas para los siguientes rangos de edad:

- Menor a 25 años
- Entre 26 y 35 años
- Entre 36 y 45 años
- Entre 46 y 55 años
- Mayor a 55 años

Dada la complejidad de esta consulta, se recomienda usar múltiples consultas anidadas para simplificar la obtención de cada rango. El resultado final debe tener una columna para cada rango de edad que debe contar con un nombre explicativo.

BONUS: Estandarización de Datos (0.5 pts)

Como observamos en el desafío 4, existen múltiples formas de escribir un mismo género, incluyendo mayúsculas y versiones con guión (Ej: *Non binary*, *non-binary*). El objetivo de este desafío es asegurar que las respuestas a esa pregunta sean parte de la siguiente lista: "female", "nonbinary", "male", "other", "not answered", siendo la última opción la asociada al valor `-1`.

Para lograrlo, cree una nueva tabla con el mismo esquema de la tabla `Answers`. Debe insertar en esta nueva tabla los registros de la pregunta de género pero modifique las respuestas para que coincidan con la lista mencionada anteriormente. Use librerías como `CSV` o `Pandas` para cambiar los valores y ese `Sqlite3` para crear la tabla nueva e ingresar las nuevas respuestas. Finalmente, repita la consulta del desafío 5.

OJO: cuidado con el manejo de caracteres especiales como espacios, signos de interrogación, etc. Si no se pasan correctamente, pueden generar errores de sintaxis en la inserción de datos.

Corrección

Es importante que deje todas las celdas de su trabajo ejecutadas antes de subir el archivo si es que es posible, en caso contrario, indicarlo en una celda de texto. Para la corrección de este laboratorio, se revisarán los procedimientos desarrollados para responder las diferentes misiones propuestas y la estructura de como utiliza el módulo `SQLite3` y el lenguaje `SQL`. Dado lo abierto de las misiones, se espera que las respuestas incluyan análisis y visualizaciones del resultado de las consultas que permitan justificar las decisiones tomadas..

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

"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.