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

IIC2115 – Programación como Herramienta para la Ingeniería (II/2017)

Actividad 10

Objetivos

- Modelar y consultar una base de datos mediante la utilización del lenguaje SQL.

Entrega

- **Lenguaje a utilizar:** Python/SQL
- **Lugar:** GitHub
- **Hora:** 16:55

Introducción

Desde un lejano y culto país, llegó una oferta de trabajo para Hernán. Su labor consistiría en realizar investigaciones sobre el mercado de libros en algunas ciudades, en diferentes situaciones. Lamentablemente, este empleo no se ajustaba a los deseos de tu ayudante, así que te recomendó a ti como un buen candidato para el trabajo.

Instrucciones

Parte 1: modelar base de datos

Antes de empezar a trabajar, necesitas modelar la base de datos sobre la que trabajarás. Para hacerlo, dispones de la siguiente información:

- Las ciudades del país lejano poseen un **nombre** y una **cantidad de habitantes**.
- Las personas tienen un **nombre**, **dirección de correo electrónico** (única para cada persona, pero no representa una llave primaria) y **una ciudad en la que viven**.
- Los libros cuentan con un **título**, un **género** y un **precio de venta**.

Además, las personas **publican libros** y también pueden **recomendarlos**. En sus recomendaciones, otorgan un puntaje de 1 a 10, donde 1 es *poco recomendado* y 10 es *muy recomendado*.

Con el objeto de simplificar las consultas de la parte siguiente, **debes** poblar tu base de datos con la siguiente información:

- Felipe vive en Hanslandia, un pueblo con 9000 habitantes, su mail es fnquinteros@uc.cl. Hasta la fecha ha publicado 2 libros. Ambos son de terror, tienen un precio de venta de $9.500 y se llaman: "20 llamadas perdidas de mamá" y "Tenemos que hablar"
- Daniela vive en Apalachicola, una ciudad con 2.000 habitantes, su mail es diflores@uc.cl. Hasta la fecha ha publicado 1 libro de comedia titulado "Un Verano sin Vicky" y tiene un precio de venta de $5.000. Ella ha recomendado con 10 puntos leer "20 llamadas perdidas de mamá"
- Hugo vive en Apalachicola, su mail es hanavarrete@uc.cl. No ha publicado ningún libro, pero si ha recomendado con 8 puntos leer "20 llamadas perdidas de mamá" y con 1 punto "Tenemos que hablar".
- Patricio vive en Apalachicola, su mail es pcerdam@uc.cl. Ha publicado un libro de terror llamado "Cariño, se duplicaron mis suegras" y tiene un costo de $10.000. Hasta ahora ha recomendado todos los libros mencionados (incluso el suyo) con puntaje 10.

Para esta parte, contarás con dos funciones en el `main.py` `crear_base_de_datos` y `poblar_base_de_datos` que deberás completar de forma tal que, al momento de ejecutarse, se creen las tablas y se llenen con la información solicitada.

Para obtener el puntaje completo en esta parte, debes asegurarte de modelar correctamente las llaves primarias, foráneas y atributos únicos de cada tabla. También tendrás revisar que no existan datos replicados, por lo que conviene usar una o más tablas para conectar las entidades. Finalmente, no olvides poblar las tablas con los datos solicitados.

Parte 2: consultas

Ahora que tienes tu base de datos modelada y poblada, el Ministerio te solicitó completar una serie de funciones entregadas en el `main.py` las cuales entregan una información específica dado una situación:

1. **Situación 1:** Bastían quiere saber si vale la pena comprar un determinado libro de sus amigos ayudantes y por ello necesita saber cuál es el puntaje de recomendación de cada uno.
   - **Esta situación debe ser respondida utilizando únicamente con una consulta SQL (puedes usar consultas anidadas) o solo podrá optar a la mitad del puntaje de este item.**
   - **función:** `puntaje_libro`
   - **Input:** título de un libro `(string)`
   - **Output:** puntaje promedio de recomendación `(integer)`
2. **Situación 2:** El ministerio necesita un conteo de los libros por cada ciudad y género. Para eso, necesitan que esta función responda a la pregunta: ¿cuántos son los libros publicados por género y por ciudad?
   - **Esta situación puede utilizar Python y SQL para responderla**
   - **función:** `conteo_libros`
   - **Input:** ninguno `(None)`
   - **Output:** diccionario donde la `key` es el nombre de cada ciudad y la `value` es otro diccionario. Este otro diccionario tiene de `key` el nombre de cada género y de `value` la cantidad de libros publicados en dicha ciudad y que tienen dicho género `(dict)`

```
{
    ciudad_1 :{
        genero_1: 4,
        genero_2: 1,
    },
    ciudad_2:{
        genero_1: 4,
        genero_2: 1,
    }
}
```

3. **Situación 3:** el gran líder Hans ha decretado que todas las ganancias generadas por la venta de libros deben ser repartidas en partes iguales para todos los habitantes de la ciudad donde habita el autor de un libro. Considerando que la acción de publicar un libro convierte a la persona en el autor de dicho libro y con solo publicarlo se venden 10000 copias de éste, el ministerio busca saber cuál es la ciudad que ha generado mayores ingresos por persona para un género específico.

    - **Esta situación puede utilizar Python y SQL para responderla. Se dará puntaje adicional por la función que utilice únicamente SQL.**
    - **función:** `ciudad_con_mas_dinero`
    - **Input:** género de un libro `(string)`
    - **Output:** ciudad que generó más ganancias por persona con los libros cuyo género es el indicado en el input y sus autores corresponden a habitantes de dicha ciudad `(string)`

4. **Situación 4:** finalmente, el ministerio siempre busca tener contactos con diversos autores, y por ello necesitan encontrar a quien haya publicado el libro más caro para una determinada ciduad.

    - **Esta situación debe ser respondida utilizando únicamente una consulta SQL (puedes usar consultas anidadas) o solo podrás optar a la mitad del puntaje de este item.**
    - **función:** `autor_libro_mas_caro`
    - **Input:** nombre de una ciudad `(string)`
    - **Output:** e-mail del autor cuyo libro es el más caro en la ciudad solicitida `(string)`

Para conseguir el puntaje completo, debes tener las 4 funciones implementadas y cada función del `main.py` debe ser capaz de ejecutarse sin ningún problema. No olvide que las funciones deben **retornar lo solicitado**, no deben hacer `print del resultado`

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.