



IIC2115 – Programación como Herramienta para la Ingeniería (I/2023)

## Ejercicios Capítulo 4a

### Aspectos generales

- **Objetivos:** aplicar los contenidos de bases de datos relacionales para modelar entidades y sus relaciones, y poblarlas con datos.
- **Lugar de entrega:** lunes 07 de noviembre a las 22:00 hrs. en repositorio privado.
- **Formato de entrega:** archivo Python Notebook (**C4a.ipynb**) con el avance logrado durante la sesión. El archivo debe estar ubicado en la carpeta **C4a**. Utilice múltiples celdas de texto y código para facilitar el trabajo del cuerpo docente. SOLO SUBA EL ARCHIVO INDICADO, NO SUBA LOS DATOS.

### Introducción

Con el fin practicar los contenidos de bases de datos relacionales, en este ejercicio deberá leer una base de datos y expandir el esquema creando tablas que modelen las entidades faltantes un conjunto de datos referentes a un encuesta sobre salud mental en relación a trabajos relacionados a tecnología. Luego, deberá poblar esas tablas con datos presentes en un archivo `.csv`.

### Descripción de los datos

En el archivo `data.zip` hay tres archivos. El primero es el archivos `salud_mental_tech.db`, la base de datos con la relación `Survey`, que contiene información sobre las encuestas que se han hecho. Su esquema es: `Survey (PRIMARY KEY INT SurveyID, TEXT Description)`.

Por otro lado, tenemos dos archivos `.csv` que corresponden a las tablas `Question` y `Answer`. Los datos de `Question.csv` corresponde a información de las preguntas que se han hecho en las encuestas. Mientras que los datos de `Answer.csv` guardan información de en las respuestas que se han hecho a las preguntas de cada encuesta.

# 1 Misiones

## 1.1 Misión 1: Leer la base de datos

Utilice la librería `sqlite3` para conectarse y leer la base de datos `salud_mental_tech.db`. Utilice las siguientes funciones para revisar las tablas que contiene actualmente y revisar su contenido. Luego, explique en sus palabras cómo hace su tarea cada función.

```
def list_tables(cursor):
    sql_query = "SELECT name FROM sqlite_master WHERE type='table';"
    cursor.execute(sql_query)
    tables = cursor.fetchall()

    for t in tables:
        t_name = t[0]
        sql_query = f"SELECT * FROM {t_name};"
        cursor.execute(sql_query)
        c_names = [description[0] for description in cursor.description]
        print(t_name)
        print(c_names, type(c_names[0]))
        print("\n=====")

def get_table_contents(cursor, t_name, limit=0):
    if limit:
        sql_query = f"SELECT * FROM {t_name} LIMIT {limit};"
    else:
        sql_query = f"SELECT * FROM {t_name}"

    print(t_name)
    cursor.execute(sql_query)
```

```

c_names = [description[0] for description in cursor.description]
print(c_names)

for r in cursor.fetchall():
    print(r)
print("\n=====\n")

```

## 1.2 Misión 2: Question.csv

Para comenzar, debe leer el archivo **Question.csv** para revisar su contenido, puede usar las librerías **pandas**, **csv** o la que estime conveniente. Al conocer las columnas que contiene, ahora debe crear una nueva tabla para estos datos en nuestra base de datos. Considere cuidadosamente qué tipo de dato usar y qué columna será la llave primaria.

Luego, debe insertar los valores en esta nueva tabla. Se recomienda usar la librería **csv**, pero puede explorar más opciones. Al terminar, muestre que se ha creado la tabla e imprima sólo 5 filas de su contenido.

## 1.3 Misión 3: Answer.csv

Para el archivo **Answer**, deberá hacer un proceso similar que el de la misión 2. Deber revisar el contenido para determinar sus columnas y los tipos de datos de cada una. Pero, en este caso, además de llave primaria, tenemos dos casos de llaves foráneas. Indique cuáles son y considere esto al crear la nueva tabla.

Además, en este caso, debemos hacer una llave primaria de múltiples columnas. Busque cómo generar una llave primaria de varias columnas y cuáles columnas se deben usar. Al terminar, muestre que se ha creado la tabla y 5 filas de su contenido.