

CuartoParcial

March 13, 2024

Before you turn this problem in, make sure everything runs as expected. First, **restart the kernel** (in the menubar, select Kernel→Restart) and then **run all cells** (in the menubar, select Cell→Run All).

Make sure you fill in any place that says YOUR CODE HERE or “YOUR ANSWER HERE”, as well as your name and collaborators below:

```
[ ]: NAME = ""  
COLLABORATORS = ""
```

Cuarto Parcial

Profesor: M. en C. Miguel Angel Pérez León

Ayudante: Diana Laura Rojas Cristino

Materia: Manejo de Datos

```
[ ]: from macti.evaluation import Quizz  
quizz = Quizz('1', 'ManejoDatos', 'CuartoParcial')
```

1 (14 Puntos) Cuarto Parcial

El examen consiste en elegir las preguntas que desees responder para aprobar (en el mejor de los casos el examen completo). Con respecto a las secciones teóricas, es necesario contestarlas con texto y lenguaje matemático, mientras que para las secciones prácticas es obligatorio escribir el código en *python* para realizar las pruebas respectivas.

En las secciones que involucren código, no hay evaluaciones parciales, ya que el algoritmo debe devolver exactamente lo que se pide, de otra forma la respuesta es incorrecta.

En cada pregunta debes ser tan explícito como la calificación que desees obtener. En caso que tu justificación o desarrollo esté incompleto, también lo será la calificación asignada.

1.1 (8 Puntos) *MySQL*

1.1.1 (0.1 Puntos)

Para establecer la conexión a la base de datos son necesarios algunos datos como:

- usuario: manejo

- host (servidor): hpc-matematicas-z.fciencias.unam.mx
- contraseña: prometeo
- base: ManejoDatos

Completa los datos faltante en la siguiente celda y valida que la conexión se haya establecido.

```
[ ]: !pip install mysql-connector-python

# se importa la biblioteca necesaria
import mysql.connector
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
# asignamos las credenciales de conexion y se establece la conexion
host=''
user=''
password=''
database=''
con = None
# YOUR CODE HERE
raise NotImplementedError()
print(con.is_connected())
```

```
[ ]: conectada = con.is_connected()
quizz.eval_numeric('1', conectada)
```

IMPORTANTE: toda consulta tiene que ser dinámica, lo que significa que no debe tener datos fijos. Por ejemplo, para buscar al alumno con el promedio más alto, no esta permitido realizar la consulta estática del id del alumno con el promedio más alto y usarlo como restricción en la consulta.

Ingresa a la base *ManejoDatos* en el servidor del curso con el usuario *manejo* y contraseña *prometeo* y realiza las siguientes consultas:

1.1.2 (0.15 Puntos)

1. Mostrar cuenta, nombre y el id_carrera de cada uno de los alumnos.

```
[ ]: sql = ''
# YOUR CODE HERE
raise NotImplementedError()
```

```
[ ]: data = pd.read_sql(sql, con)
print(data)
datos = [str(data)]
quizz.eval_numeric('2', datos)
```

1.1.3 (0.25 Puntos)

2. Mostrar cuenta, nombre y el `id_carrera` de cada uno de los alumnos ordenados por `id_carrera`.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()

[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('3', datos)
```

1.1.4 (0.5 Puntos)

3. Muestra el `id_carrera` y cuantos alumnos hay inscritos por carrera.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()

[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('4', datos)
```

1.1.5 (0.75 Puntos)

4. Muestra el nombre y la calificación más alta de cada alumno.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()

[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('5', datos)
```

1.1.6 (0.75 Puntos)

5. Muestra el `nombre_curso` y el promedio de cada curso.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()
```

```
[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('6', datos)
```

1.1.7 (1 Punto)

6. Muestra el nombre_curso y el promedio de la materia de programacion 1.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()
```

```
[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('7', datos)
```

1.1.8 (1 Punto)

7. Muestra el nombre_carrera y cuantos alumnos tiene inscritos.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()
```

```
[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('8', datos)
```

1.1.9 (1.5 Puntos)

8. Muestra cuenta, nombre y promedio de cada uno de los alumnos.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()
```

```
[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('9', datos)
```

1.1.10 (2 Puntos)

9. Muestra el nombre y el promedio más alto.

```
[ ]: sql = ''
      # YOUR CODE HERE
      raise NotImplementedError()

[ ]: data = pd.read_sql(sql, con)
      print(data)
      datos = [str(data)]
      quizz.eval_numeric('10', datos)
```

2 (4 Puntos) Estándares (****XML****, ****JSON****)

2.1 (2 Puntos)

Dada la base **FCiencias** genera un documento tipo llamado creaJSON.json de tipo *JSON* usando *Python* y el conector *MySQL*, que contenga la siguiente información por cada alumno:

- Nombre del alumno.
- Nombre del tutor.
- Nombre de la especialidad.

Tu documento debe generarse usndo las tablas de **ALUMNOS**, **TUTORES** y **ESPECIALIDADES** y debe verse igual al siguiente documento.

IMPORTANTE: si tu documento se genera haciendo uso de tablas diferenres a las permitidas, se te restaran puntos.

```
[ ]: # se importa la biblioteca necesaria
import mysql.connector
import json

# asignamos las credenciales de conexion y se establece la conexion
con = mysql.connector.connect(host="hpc-matematicas-z.fciencias.unam.mx",
                              user="manejo",
                              password="prometeo",
                              database="FCiencias")

# funcion que nos permite consultar la base de datos
def creaJSON():
    # YOUR CODE HERE
    raise NotImplementedError()

# se llama a la funcion para crear JSON
creaJSON()
```

2.2 (2 Puntos) *Python* + *MySQL* + *JSON*

Tomando como base el documento generado en la pregunta anterior (*CreaJSON.json*), ahora tienes que leerlo con *Python* y escribir los datos en la tabla **ALUMNOSYTUTORES**, esta tabla tiene el siguiente esquema.

Una vez que hayas ejecutado tu código, puedes consultar la tabla ALUMNOSYTUTORES para validar tu resultado.

Es importante aclarar que **el usuario *manejo* (mediante el cual realizaras INSERT a la base de datos) es un usuario compartido con todos los integrantes del curso**, así que cuando realices tus pruebas fíjate en el número de folio antes y después de ejecutar tu celda para realmente estar seguro que tu código funciona.

```
[ ]: # importamos el modulo json y mysql
import json
import mysql.connector

# se abre la conexion a la bd
con = mysql.connector.connect(host="hpc-matematicas-z.fciencias.unam.mx",
    user="manejo",
    password="prometeo",
    database="FCiencias")

cursor = con.cursor()

# YOUR CODE HERE
raise NotImplementedError()

cursor.close()
con.close()
```

3 Referencias

- Bases de Datos
- Estándares