Python and SQL Questions

# Python Questions

## 1. Regular Expressions

Write a Python program using regular expressions to find all adjectives in the following sentence: 'The quick brown fox jumps over the lazy dog.'

import nltk

from nltk import pos\_tag

from nltk.tokenize import word\_tokenize

# Download the necessary NLTK data

nltk.download('punkt')

nltk.download('averaged\_perceptron\_tagger')

sentence = 'The quick brown fox jumps over the lazy dog.'

# Tokenize the sentence

words = word\_tokenize(sentence)

# Tag the parts of speech

tagged\_words = pos\_tag(words)

# Initialize an empty list to store adjectives

adjectives = []

# Loop through tagged words and identify adjectives (JJ)

for word, pos in tagged\_words:

if pos == 'JJ':

adjectives.append(word)

print("Adjectives in the sentence:", adjectives)

Adjectives in the sentence: ['quick', 'lazy']

## 2. Lists

Write a Python function that takes a list of numbers and returns a new list containing only the unique elements of the original list.

def unique\_elements(input\_list):

return list(set(input\_list))

numbers = [1, 2, 3, 4, 4, 5, 6, 6, 7, 8, 8, 9]

unique\_numbers = unique\_elements(numbers)

print(unique\_numbers)

## 3. File Handling

Write a Python script to read a file 'data.txt' and print its content. Then, append the line 'End of the file' to it and save the changes.

# Reading the file

with open('data.txt', 'r') as file:

content = file.read()

print("Original Content:")

print(content)

# Appending to the file

with open('data.txt', 'a') as file:

file.write('\nEnd of the file')

print("File Updated Successfully!")

## 4. Exception Handling

Write a Python program to handle an exception thrown by the following code block: division = 10 / 0. Catch the exception and print an error message.

try:

division = 10 / 0

except ZeroDivisionError:

print("Error: Division by zero!")

# SQL Questions

## 1. Joins

Write an SQL query to select all employees' names and their department names. Use the tables 'employees' and 'departments', where 'department\_id' is a foreign key in 'employees' table that references 'id' in 'departments' table. Assume both tables have 'name' columns.

SELECT e.name AS employee\_name, d.name AS department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.id;

## 2. Subquery

Write an SQL query to find the names of employees who earn more than the average salary in their department. Use the tables 'employees' (with columns 'name', 'salary', and 'department\_id') and 'departments' (with columns 'id', 'name').

SELECT e.name

FROM employees e

JOIN (

SELECT department\_id, AVG(salary) AS avg\_salary

FROM employees

GROUP BY department\_id

) AS dept\_avg ON e.department\_id = dept\_avg.department\_id

WHERE e.salary > dept\_avg.avg\_salary;

## 3. Joins (Different Context)

Write an SQL query to find all products sold in the year 2020. Use the 'orders' table (with columns 'id', 'product\_id', 'order\_date') and the 'products' table (with columns 'id', 'name'). Assume the 'product\_id' in 'orders' is a foreign key that references 'id' in 'products' table.

SELECT p.name AS product\_name

FROM orders o

JOIN products p ON o.product\_id = p.id

WHERE YEAR(o.order\_date) = 2020;