**Python**

**Q1. Explain the concept of list comprehensions in Python and provide an example of how they can be used.**

List comprehensions are a concise way to create lists in Python. They allow you to generate a new list by applying an expression to each item in an iterable (like a list or a range) while optionally filtering items using a condition.

**Example:**

# Generate a list of squares of even numbers from 0 to 9

squares\_of\_even\_numbers = [x\*\*2 for x in range(10) if x % 2 == 0]

print(squares\_of\_even\_numbers) # Output: [0, 4, 16, 36, 64]

**Q2. Python program to generate even numbers.**

# Generate even numbers from 0 to 20

even\_numbers = [x for x in range(21) if x % 2 == 0]

print(even\_numbers) # Output: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

**Q3. Explain the purpose and usage of Python's context managers using the with statement. Provide an example.**

Context managers provide a way to allocate and release resources precisely when you want to. The with statement simplifies exception handling by encapsulating common preparation and cleanup tasks in so-called context manager objects.

**Example:**

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

data = file.read()

# File is automatically closed after the block under 'with' is executed.

**Q4. Write a regular expression to match and extract all email addresses from a given text string.**

Here's a simple regular expression to match email addresses:

import re

text = "Please contact us at support@example.com or sales@example.org."

emails = re.findall(r'[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}', text)

print(emails) # Output: ['support@example.com', 'sales@example.org']

**Q5. What is the purpose of the if name == "main": block in Python scripts?**

This block checks whether the script is being run as the main module or being imported by another module. If the script is run directly, the code inside this block will execute, which allows you to run some test code or script functionality without it executing when the file is imported.

**Example:**

def main():

print("This script is being run directly.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**SQL**

**Q1. Explain the difference between a SQL database and a SQL table.**

* A SQL database is a collection of data organized in a structured way. It contains one or more tables along with metadata, schemas, and relationships among the data.
* A SQL table is a specific structure within a database that organizes data into rows and columns. Each row represents a record, and each column represents a field within that record.

**Q2. Different Commands in SQL?**

Common SQL commands include:

* **DDL (Data Definition Language)**:
  + CREATE, ALTER, DROP (for schema definitions).
* **DML (Data Manipulation Language)**:
  + SELECT, INSERT, UPDATE, DELETE (for data manipulation).
* **DCL (Data Control Language)**:
  + GRANT, REVOKE (for permission control).
* **TCL (Transaction Control Language)**:
  + COMMIT, ROLLBACK (for transaction management).

**Q3. What is the purpose of the SQL WHERE clause, and how is it used in a SELECT statement?**

The SQL WHERE clause is used to filter records that meet a specified condition. It allows you to retrieve only those rows from a table that satisfy certain criteria.

**Example:**

SELECT \* FROM Customers WHERE Country = 'USA';

This query retrieves all customers located in the USA.

**Q5. What is the difference between SQL's GROUP BY and HAVING clauses? Provide examples of when you would use each.**

* **GROUP BY**: This clause is used to group rows that have the same values in specified columns into summary rows. It is often used with aggregate functions.

**Example:**

SELECT Country, COUNT(\*) FROM Customers GROUP BY Country;

This will count the number of customers in each country.

* **HAVING**: This clause is like the WHERE clause but is used to filter groups created by GROUP BY. You can use it to filter results based on aggregate functions.

**Example:**

SELECT Country, COUNT(\*) FROM Customers GROUP BY Country HAVING COUNT(\*) > 5;

This query will show countries with more than five customers.

**Data Science**

**Q1. Discuss the role of regular expressions in text extraction and provide an example where they can be applied effectively.**

Regular expressions (regex) are used in text extraction to identify patterns within text. They allow for searching, matching, and manipulating strings based on specific patterns, making them very useful for tasks such as validation, extraction, and replacement.

**Example**: Extracting date formats from a text string.

import re

text = "Important dates are 2022-01-15, 2023-04-20, and 2021-12-31."

dates = re.findall(r'\d{4}-\d{2}-\d{2}', text)

print(dates) # Output: ['2022-01-15', '2023-04-20', '2021-12-31']

**Q2. How does named entity recognition contribute to text extraction? Provide an example.**

Named Entity Recognition (NER) identifies and classifies key entities in text into predefined categories such as names, organizations, locations, dates, etc. It contributes significantly to text extraction by enabling more structured data extraction from unstructured text.

**Example**: Using an NER library like spaCy to identify entities.

import spacy

nlp = spacy.load("en\_core\_web\_sm")

text = "Apple Inc. was founded in Cupertino, California on April 1, 1976."

doc = nlp(text)

for ent in doc.ents:

print(ent.text, ent.label\_) # Output: Apple Inc. ORG, Cupertino GPE, California GPE, April 1, 1976 DATE

**Q3. Explain the difference between rule-based and statistical approaches in text extraction.**

* **Rule-Based Approaches**: These rely on predefined rules and patterns to extract information. They are often more interpretable but lack flexibility; changes in the input format require rule adjustments. Examples include using regular expressions for specific patterns.
* **Statistical Approaches**: These involve machine learning models trained on labeled data. They adapt to variations in data and improve accuracy as they learn from more examples. However, they may require more data and are less interpretable. An example is using NER models trained on large corpora for identifying entities.