Questions

# Python Try-Except Questions

1. What is the purpose of the try-except clause in Python?

To handle exceptions or errors that may occur during the execution of code. It allows you to gracefully handle errors without crashing the program, by providing a mechanism to catch and respond to exceptions.

2. How can you catch specific exceptions in Python? Provide an example.

You can catch specific exceptions in Python by specifying the exception type after the except keyword.

try:

x = 10 / 0

except ZeroDivisionError:

print("Error: Division by zero occurred")

3. What is the use of the `else` block in try-except statements?

The else block in try-except statements is used to execute code when no exceptions occur within the try block. It is optional and provides a way to distinguish the code that should run only if the try block executes successfully.

try:

x = 10 / 2

except ZeroDivisionError:

print("Error: Division by zero occurred")

else:

print("No error occurred, result:", x)

4. Explain how to use the `finally` block in try-except statements.

The finally block in try-except statements is used to execute cleanup code regardless of whether an exception occurs or not. It is optional and is typically used to release external resources like files or network connections.

try:

f = open("example.txt", "r")

except FileNotFoundError:

print("File not found")

finally:

f.close() # This will always execute, even if an exception occurs

SQL Joins Questions

1. What are the different types of JOINs available in SQL?

INNER JOIN

LEFT JOIN (or LEFT OUTER JOIN)

RIGHT JOIN (or RIGHT OUTER JOIN)

FULL JOIN (or FULL OUTER JOIN)

CROSS JOIN

2. How would you join two tables using an INNER JOIN and what result does it return?

SELECT column1, column2

FROM table1

INNER JOIN table2 ON table1.column = table2.column;

This query will return only the rows from both table1 and table2 that have matching values in the specified column(s).

3. Explain the difference between LEFT JOIN and RIGHT JOIN with examples.

LEFT JOIN: All rows from the left table are preserved, regardless of whether there is a match in the right table. If there's no match, NULL values are returned for columns from the right table.

SELECT \*

FROM table1

LEFT JOIN table2 ON table1.column = table2.column;

RIGHT JOIN: All rows from the right table are preserved, regardless of whether there is a match in the left table. If there's no match, NULL values are returned for columns from the left table.

SELECT \*

FROM table1

RIGHT JOIN table2 ON table1.column = table2.column;