**Introduction to Python**

1. **What are the key features of Python?**
   * Easy to learn and read.
   * Interpreted and dynamically typed.
   * Supports object-oriented, procedural, and functional programming.
   * Extensive libraries and community support.
2. **Briefly explain Python's history and its growth in popularity.**
   * Python was created by Guido van Rossum in 1991.
   * Its simplicity and versatility have made it a popular choice for web development, data analysis, AI, and more.
3. **How do you install Python on your system?**
   * Download from the official Python website (python.org).
   * Install and add Python to your system's PATH.
4. **What are the differences between running a Python program using an interpreter and an IDE?**
   * **Interpreter**: Runs Python code line-by-line in a terminal or REPL.
   * **IDE**: Provides features like debugging, code suggestions, and project management.
5. **Why is indentation important in Python?**
   * Python uses indentation to define code blocks instead of braces {}. Improper indentation causes errors.
6. **Can you write and explain a simple "Hello, World!" program in Python?**

python

Copy code

print("Hello, World!")

* + The print() function outputs text to the screen.

**2. Data Types and Variables**

1. **What are the different data types in Python?**
   * Numeric (int, float, complex), Sequence (str, list, tuple), Mapping (dict), Set (set), Boolean (bool).
2. **How is a string different from a list?**
   * Strings are immutable and store text, while lists are mutable and store sequences of items.
3. **What is the difference between a tuple and a list?**
   * Tuples are immutable, while lists are mutable.
4. **How do you perform type conversion in Python? Provide examples.**
   * Use int(), float(), str(), etc.  
     Example: int("10") converts the string "10" to the integer 10.
5. **Can you explain dictionaries and sets with examples?**
   * **Dictionary**: Key-value pairs. Example: {'name': 'John', 'age': 30}.
   * **Set**: Unordered collection of unique elements. Example: {1, 2, 3}.
6. **What is the purpose of the complex data type in Python?**
   * Represents complex numbers. Example: 3+4j.

**3. Basic Input/Output**

1. **How do you take input from a user in Python?**
   * Use input(). Example: name = input("Enter your name: ").
2. **What are the ways to format output in Python?**
   * Using f-strings: f"Hello, {name}".
   * Using .format(): "Hello, {}".format(name).
3. **What does the print() function do, and how can you customize its behavior?**
   * Outputs data. Customize with sep and end parameters. Example: print("Hello", "World", sep="-").

**4. Operators**

1. **List all the arithmetic operators in Python with examples.**
   * +, -, \*, /, %, //, \*\*. Example: 5 + 2 = 7.
2. **What is the difference between == and is in Python?**
   * == checks value equality, is checks object identity.
3. **How do logical operators work in Python?**
   * and, or, not. Example: True and False returns False.
4. **Can you explain bitwise operators with examples?**
   * &, |, ^, ~, <<, >>. Example: 5 & 3 = 1.
5. **What are identity and membership operators in Python?**
   * **Identity**: is, is not. Example: x is y.
   * **Membership**: in, not in. Example: "a" in "apple".

**5. Control Flow**

1. **Explain how if, elif, and else work in Python with examples.**

python

Copy code

x = 10

if x > 0:

print("Positive")

elif x == 0:

print("Zero")

else:

print("Negative")

1. **How do you write a while loop? Provide an example.**

python

Copy code

count = 0

while count < 5:

print(count)

count += 1

1. **What is the difference between break, continue, and pass statements in loops?**
   * break: Exits loop.
   * continue: Skips to next iteration.
   * pass: Placeholder, does nothing.
2. **How does a for loop work in Python? Can it iterate over a dictionary?**
   * Yes, it can. Example:

python

Copy code

for key, value in my\_dict.items():

print(key, value)

**6. Functions**

1. **How do you define a function in Python?**
   * Use def. Example:

def greet(name):

return f"Hello, {name}"

1. **What are default and keyword arguments in Python? Provide examples.**
   * Default: Predefined values. Example: def greet(name="Guest").
   * Keyword: Pass by name. Example: greet(name="John").
2. **Explain the difference between \*args and \*\*kwargs.**
   * \*args: Tuple of arguments.
   * \*\*kwargs: Dictionary of keyword arguments.
3. **What is a lambda function? Write an example.**
   * Anonymous function. Example: square = lambda x: x\*\*2.
4. **What is the scope of a variable? Explain local and global variables with examples.**
   * Local: Defined inside a function.
   * Global: Defined outside all functions.

**7. Modules and Packages**

1. **How do you import a module in Python?**
   * Use import. Example: import math.
2. **What is the difference between import module and from module import?**
   * import module imports entire module.
   * from module import imports specific components.
3. **Can you explain how to create a custom module?**
   * Save Python code in a .py file and import it.
4. **Name a few commonly used standard libraries in Python and their purposes.**
   * math (mathematics), datetime (date and time), random (random numbers).
5. **How do you install packages using pip?**
   * Use pip install <package\_name>.

**Introduction to Python**

1. **Why is Python considered an interpreted language?**
   * Python code is executed line-by-line by the Python interpreter, rather than being compiled into machine code first.
2. **What is PEP 8, and why is it important?**
   * PEP 8 is the style guide for Python, promoting readability and consistency in Python code.
3. **What are some popular Python IDEs?**
   * PyCharm, VS Code, Jupyter Notebook, and Spyder.

**2. Data Types and Variables**

1. **What is the difference between mutable and immutable types in Python?**
   * Mutable types (e.g., lists, dictionaries) can be changed after creation, while immutable types (e.g., strings, tuples) cannot.
2. **What is Python's None type?**
   * It represents the absence of a value or a null value, often used as a default function return.
3. **How are variables dynamically typed in Python?**
   * Variables in Python do not require explicit type declaration; the type is assigned during runtime based on the value.

**3. Basic Input/Output**

1. **How can you read a multi-line input from the user?**
   * Use a loop:

python

Copy code

lines = []

while True:

line = input()

if line:

lines.append(line)

else:

break

1. **How can you handle special characters in strings for output?**
   * Use escape sequences like \n for new lines or raw strings (prefix with r).

**4. Operators**

1. **What is the difference between / and // in Python?**
   * / performs floating-point division, while // performs floor division (returns the integer part).
2. **What is operator precedence, and why is it important?**
   * Operator precedence determines the order in which operations are executed. For example, multiplication has higher precedence than addition.
3. **How does Python handle division by zero?**
   * It raises a ZeroDivisionError.

**5. Control Flow**

1. **What is the purpose of an else block in a loop?**
   * The else block executes if the loop completes normally (no break was encountered).
2. **How does the range() function work in loops?**
   * It generates a sequence of numbers. Example: range(5) produces 0, 1, 2, 3, 4.
3. **Can you write a Python program to check if a number is prime?**

python

Copy code

def is\_prime(n):

if n < 2:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

**6. Functions**

1. **What is the purpose of the return statement in Python?**
   * It exits a function and optionally returns a value to the caller.
2. **What are higher-order functions? Provide an example.**
   * Functions that take other functions as arguments or return them. Example:

python

Copy code

def apply(func, x):

return func(x)

print(apply(lambda x: x\*\*2, 5)) # Outputs: 25

1. **What is recursion? Provide a simple example.**
   * A function calling itself. Example:

python

Copy code

def factorial(n):

if n == 0:

return 1

return n \* factorial(n - 1)

**7. Modules and Packages**

1. **What is the difference between sys.path and PYTHONPATH?**
   * sys.path is the list of directories Python searches for modules at runtime.
   * PYTHONPATH is an environment variable used to add directories to sys.path.
2. **What are the benefits of using modules?**
   * Code reusability, better organization, and reduced redundancy.
3. **How do you explore the contents of a module?**
   * Use the dir() function. Example: dir(math) lists all functions and constants in the math module.
4. **What is the role of the \_\_init\_\_.py file in Python packages?**
   * It indicates that a directory is a package and can contain initialization code for the package.

**General Questions**

1. **What is the difference between a shallow copy and a deep copy?**
   * A shallow copy creates a new object but references the original nested objects.
   * A deep copy creates a new object and recursively copies all nested objects. Use the copy module for deep copies.
2. **How does Python manage memory for objects?**
   * Python uses reference counting and garbage collection to manage memory.
3. **What is the Global Interpreter Lock (GIL)?**
   * GIL is a mutex that allows only one thread to execute Python bytecode at a time, ensuring thread safety in CPython.

**POSTGRESQL**

**Day 1: PostgreSQL Basics**

1. **What is a database?**
   * A database is an organized collection of data stored electronically, enabling easy access, management, and updating.
   * Types include relational databases (e.g., PostgreSQL, MySQL) and NoSQL databases (e.g., MongoDB, Redis).
2. **What is PostgreSQL, and what are its key features?**
   * PostgreSQL is an open-source, relational database management system (RDBMS) known for advanced features like ACID compliance, extensibility, JSON support, and full-text search.
3. **How do you create and connect to a PostgreSQL database?**
   * Command to create: CREATE DATABASE mydb;
   * To connect: \c mydb (in psql) or use a connection string in applications.
4. **How do you rename or drop a database in PostgreSQL?**
   * Rename: ALTER DATABASE old\_name RENAME TO new\_name;
   * Drop: DROP DATABASE mydb;

**Day 2: Data Types in PostgreSQL**

1. **What is the difference between CHAR, VARCHAR, and TEXT in PostgreSQL?**
   * CHAR(n): Fixed-length, padded with spaces.
   * VARCHAR(n): Variable-length with a defined limit.
   * TEXT: Variable-length with no limit.
2. **How do you convert data types in PostgreSQL?**
   * Using CAST. Example:

sql

Copy code

SELECT CAST(42 AS TEXT);

1. **What is a UUID, and where is it used?**
   * A UUID (Universally Unique Identifier) is a 128-bit identifier commonly used for generating unique keys.

**Day 3: Querying & Filtering Data**

1. **How do you fetch unique records using SELECT DISTINCT?**
   * Example:

sql

Copy code

SELECT DISTINCT column\_name FROM table\_name;

1. **What is the difference between LIKE and ILIKE?**
   * LIKE is case-sensitive; ILIKE is case-insensitive.
2. **How do you paginate query results in PostgreSQL?**
   * Using LIMIT and OFFSET:

sql

Copy code

SELECT \* FROM table\_name LIMIT 10 OFFSET 20;

**Day 4: Managing Tables**

1. **How do you create a table with an auto-incrementing column?**

sql

Copy code

CREATE TABLE my\_table (

id SERIAL PRIMARY KEY,

name TEXT

);

1. **How do you add and drop a column in PostgreSQL?**
   * Add: ALTER TABLE table\_name ADD COLUMN column\_name data\_type;
   * Drop: ALTER TABLE table\_name DROP COLUMN column\_name;
2. **How do you import a CSV file into a PostgreSQL table?**
   * Use COPY:

sql

Copy code

COPY table\_name FROM '/path/to/file.csv' DELIMITER ',' CSV HEADER;

**Day 5: Modifying Data**

1. **What is the difference between INSERT and UPSERT?**
   * INSERT adds new rows, while UPSERT updates existing rows if there’s a conflict or inserts if no conflict exists.  
     Example:

sql

Copy code

INSERT INTO my\_table (id, name)

VALUES (1, 'Alice')

ON CONFLICT (id) DO UPDATE SET name = EXCLUDED.name;

1. **How do you handle NULL values in queries?**
   * Use IS NULL or IS NOT NULL.  
     Example:

sql

Copy code

SELECT \* FROM table\_name WHERE column\_name IS NULL;

1. **What does COALESCE do in PostgreSQL?**
   * It returns the first non-NULL value from a list.  
     Example:

sql

Copy code

SELECT COALESCE(column1, 'default\_value') AS result FROM table\_name;

**Day 6: Aggregate Functions and Grouping**

1. **How do aggregate functions handle NULL values?**
   * NULL values are ignored unless explicitly handled.
2. **How do you filter grouped data using HAVING?**

sql

Copy code

SELECT department, COUNT(\*)

FROM employees

GROUP BY department

HAVING COUNT(\*) > 10;

**Day 7: PostgreSQL Join Operations**

1. **What is the difference between INNER JOIN and LEFT JOIN?**
   * INNER JOIN: Returns matching rows from both tables.
   * LEFT JOIN: Returns all rows from the left table and matching rows from the right table.
2. **When would you use UNION vs. UNION ALL?**
   * UNION removes duplicates, while UNION ALL retains them.
3. **How do you write a Cartesian product query?**

sql

Copy code

SELECT \* FROM table1 CROSS JOIN table2;

**Day 8: Control Flow and Transactions**

1. **What is the difference between COMMIT and ROLLBACK?**
   * COMMIT: Saves changes made during a transaction.
   * ROLLBACK: Undoes changes made during a transaction.
2. **What is the purpose of constraints in PostgreSQL?**
   * Constraints enforce rules on data integrity, e.g., NOT NULL, PRIMARY KEY, FOREIGN KEY.

**Revised Day 9: Advanced SQL Queries and View Operations**

1. **How does DISTINCT differ from GROUP BY?**
   * DISTINCT filters unique rows.
   * GROUP BY groups rows based on a column, often used with aggregate functions.
2. **How do you create or replace a view?**

sql

Copy code

CREATE OR REPLACE VIEW view\_name AS

SELECT column\_name FROM table\_name WHERE condition;

**Revised Day 10: Advanced Topics and File Interactions**

1. **How do you calculate the size of a PostgreSQL database?**

sql

Copy code

SELECT pg\_size\_pretty(pg\_database\_size('database\_name'));

1. **How do you export data from PostgreSQL to a CSV file?**

sql

Copy code

COPY table\_name TO '/path/to/file.csv' DELIMITER ',' CSV HEADER;

1. **What is the difference between CURRENT\_DATE and CURRENT\_TIMESTAMP?**
   * CURRENT\_DATE: Returns the current date.
   * CURRENT\_TIMESTAMP: Returns the current date and time.
2. **Question:**  
   "Can you explain how PostgreSQL manages transactions internally to ensure ACID compliance?"

**Twist:**  
"What happens if there's a power failure during a transaction? How does PostgreSQL recover?"

**Answer:**  
PostgreSQL uses a Write-Ahead Log (WAL) to ensure durability. During a power failure, upon restart, PostgreSQL replays the WAL to recover any committed transactions and discard incomplete ones. This ensures atomicity and consistency.

**Data Types**

1. **Question:**  
   "You are asked to store hierarchical data, like an organizational structure, in PostgreSQL. Which data type or approach would you use?"

**Twist:**  
"How would you optimize queries on this hierarchical data?"

**Answer:**

* + Use the JSONB or ARRAY data type for flexibility.
  + To optimize queries, create a GIN index on JSONB fields or use recursive CTEs for better traversal of hierarchical relationships.

**Querying & Filtering Data**

1. **Question:**  
   "How would you fetch the second highest salary from an employee table?"

**Twist:**  
"What if two employees have the same salary, but you only want one result?"

**Answer:**

sql

Copy code

SELECT MAX(salary)

FROM employees

WHERE salary < (SELECT MAX(salary) FROM employees);

* + To limit duplicates: Add DISTINCT to the inner query or LIMIT 1 after ordering results.

**Advanced Queries**

1. **Question:**  
   "How can you efficiently find duplicate rows in a table based on multiple columns?"

**Twist:**  
"What if the table contains millions of rows?"

**Answer:**

sql

Copy code

SELECT column1, column2, COUNT(\*)

FROM table\_name

GROUP BY column1, column2

HAVING COUNT(\*) > 1;

* + For large datasets, use indexes on the grouped columns to optimize performance.

**Joins and Complex Data Manipulation**

1. **Question:**  
   "What is the difference between INNER JOIN and CROSS JOIN?"

**Twist:**  
"How would you simulate a CROSS JOIN using UNION?"

**Answer:**

* + CROSS JOIN produces the Cartesian product.
  + To simulate it:

sql

Copy code

SELECT column1, column2 FROM table1

UNION

SELECT column1, column2 FROM table2;

**Optimization**

1. **Question:**  
   "What are the steps you take to optimize a slow query?"

**Twist:**  
"The query uses multiple joins and aggregate functions. How do you debug performance issues?"

**Answer:**

* + Steps:
    1. **Analyze the execution plan** using EXPLAIN or EXPLAIN ANALYZE.
    2. Add appropriate indexes (B-Tree for equality, GIN/GIN for text or JSONB).
    3. Reduce unnecessary columns in SELECT.
    4. Use materialized views for aggregates.
    5. Apply partitioning for large tables.

**Constraints and Transactions**

1. **Question:**  
   "Can you explain the difference between a primary key and a unique constraint?"

**Twist:**  
"What happens if you try to insert a NULL value into a column with a unique constraint but no primary key?"

**Answer:**

* + Both enforce uniqueness, but the primary key also disallows NULL values.
  + A column with a unique constraint can accept multiple NULLs since NULL != NULL by SQL standards.

**Aggregate Functions**

1. **Question:**  
   "How does PostgreSQL handle NULLs in aggregate functions?"

**Twist:**  
"What is the result of this query: SELECT AVG(NULL);?"

**Answer:**  
Aggregate functions ignore NULLs.

* + AVG(NULL) results in NULL because there are no values to calculate.

**File Interactions and Data Import/Export**

1. **Question:**  
   "How do you import data from a CSV file into a PostgreSQL table?"

**Twist:**  
"What happens if the CSV contains duplicate rows or invalid data types?"

**Answer:**

* + Use COPY or \copy.
  + To handle duplicates, use ON CONFLICT or import into a temporary table and filter before insertion.
  + For invalid data types, pre-validate the file using tools like Python or create a staging table.

**Practical Scenario**

1. **Question:**  
   "You are given a table with millions of rows and are asked to add a new column with a default value. What approach would you take?"

**Twist:**  
"How do you ensure minimal downtime for this operation?"

**Answer:**

* + Use ALTER TABLE to add the column without a default value first, which is faster:

sql

Copy code

ALTER TABLE table\_name ADD COLUMN new\_column data\_type;

* + Then backfill values in chunks to avoid locking:

UPDATE table\_name SET new\_column = default\_value WHERE condition LIMIT batch\_size;

**Twisted Queries**

1. **Question:**  
   "How can you find the third highest salary from an employee table?"

**Twist:**  
"What if there are duplicate salaries, and you want the third distinct highest salary?"

**Answer:**

sql

Copy code

SELECT DISTINCT salary

FROM employees

ORDER BY salary DESC

OFFSET 2 LIMIT 1;

* + Use DISTINCT to filter unique salaries, ORDER BY for descending order, and OFFSET to skip the top two.

1. **Question:**  
   "How do you find the employees who earn more than the average salary of their department?"

**Twist:**  
"What if two departments have the same average salary?"

**Answer:**

sql

Copy code

SELECT e.employee\_id, e.name

FROM employees e

JOIN (

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

FROM employees

GROUP BY department\_id

) d ON e.department\_id = d.department\_id

WHERE e.salary > d.avg\_salary;

* + To handle duplicate averages, this query focuses only on matching department and individual comparisons.

**Indexing and Optimization**

1. **Question:**  
   "What is the difference between a B-Tree index and a GIN index?"

**Twist:**  
"If you have a JSONB column and a text column in the same query, which index would you use, and why?"

**Answer:**

* + **B-Tree Index:** Optimized for equality and range queries (e.g., =, <, >).
  + **GIN Index:** Optimized for complex data types like JSONB and arrays.
  + Use a **GIN index** for the JSONB column and a **B-Tree index** for the text column. For combined queries, analyze with EXPLAIN to decide.

1. **Question:**  
   "Why is a full table scan sometimes faster than using an index?"

**Answer:**

* + Full table scans can be faster if:
    - The table is small.
    - The query retrieves most rows, making index lookups redundant.
    - Index maintenance overhead outweighs its benefits.
  + PostgreSQL dynamically chooses based on cost estimates.

**Joins**

1. **Question:**  
   "What happens if you use a LEFT JOIN but reference columns from the right table in the WHERE clause?"

**Twist:**  
"How would you fix it to return all rows from the left table, even if there's no match?"

**Answer:**

* + Referencing a column from the right table in the WHERE clause turns the LEFT JOIN into an INNER JOIN.
  + Fix it using the ON condition or move filters into the JOIN clause:

sql

Copy code

SELECT \*

FROM table1 t1

LEFT JOIN table2 t2

ON t1.id = t2.id AND t2.status = 'active';

**Data Manipulation**

1. **Question:**  
   "How would you update a table with millions of rows without locking the table?"

**Twist:**  
"What if other transactions need to read or write during the update?"

**Answer:**

* + Use **batch updates** with a WHERE condition and LIMIT:

sql

Copy code

DO $$

BEGIN

LOOP

UPDATE table\_name

SET column\_name = new\_value

WHERE condition

LIMIT 1000;

EXIT WHEN NOT FOUND;

END LOOP;

END $$;

* + Add NOWAIT for concurrent transactions or use VACUUM to minimize locking impact.

**Constraints**

1. **Question:**  
   "What is the difference between a FOREIGN KEY constraint and a UNIQUE constraint?"

**Twist:**  
"Can a foreign key column have duplicate values or NULLs?"

**Answer:**

* + A FOREIGN KEY enforces referential integrity but allows duplicates and NULLs unless explicitly combined with a UNIQUE or NOT NULL constraint.
  + A UNIQUE constraint ensures all values in a column are unique but doesn’t establish relationships with other tables.

**Aggregate Functions**

1. **Question:**  
   "How do aggregate functions handle NULL values?"

**Twist:**  
"How would you count the number of NULL values in a column?"

**Answer:**

* + Aggregate functions like SUM, AVG, MAX ignore NULLs, but COUNT(\*) includes them.
  + To count NULLs explicitly:

sql

Copy code

SELECT COUNT(\*) - COUNT(column\_name) AS null\_count

FROM table\_name;

**Error Scenarios**

1. **Question:**  
   "What happens if you try to insert data violating a unique constraint?"

**Twist:**  
"How can you modify the query to handle duplicates gracefully?"

**Answer:**

* + PostgreSQL throws an error.
  + Use ON CONFLICT:

sql

Copy code

INSERT INTO table\_name (id, value)

VALUES (1, 'new\_value')

ON CONFLICT (id)

DO UPDATE SET value = EXCLUDED.value;

**Complex Scenarios**

1. **Question:**  
   "You have a large table with a composite primary key. How would you partition this table for better performance?"

**Twist:**  
"What type of queries would benefit most from this partitioning?"

**Answer:**

* + Partition by range or hash using a relevant column:

sql

Copy code

CREATE TABLE partitioned\_table

PARTITION BY RANGE (date\_column);

* + Queries filtering on the partition key (date\_column) benefit most, as PostgreSQL prunes irrelevant partitions.