PYTHON MCQ

1️⃣ What will be the output of the following code?

```
x = [1, 2, 3]
print(x * 2)
```

A) [2, 4, 6]
B) [1, 2, 3, 1, 2, 3]
C) [1, 2, 3, 2, 4, 6]
D) Error
✅ **Answer:** B
➡️ *List multiplication replicates elements, it doesn't multiply values.*

2️⃣ What is the data type of the variable x after executing x = (1,)?

A) tuple
B) list
C) int
D) set
✅ **Answer:** A
➡️ *A trailing comma defines a single-element tuple.*

3️⃣ What will print(0.1 + 0.2 == 0.3) output?

A) True
B) False
C) 0.3
D) Error
✅ **Answer:** B
➡️ *Floating-point arithmetic precision issue makes this False.*

4️⃣ What is the output of:

```
a = [1, 2, 3]
b = a
b.append(4)
print(a)
```

A) [1, 2, 3]
B) [1, 2, 3, 4]
C) [4]
D) Error
✅ **Answer:** B
➡️ *Lists are mutable and both a and b refer to the same object.*

5️⃣ What is the output of:

```
print("hello"[-1])
```

A) h
B) o
C) hello
D) Error
✅ **Answer:** B
➡️ *Negative indexing starts from the end.*

6️⃣ Which of the following statements is true?

A) Python is compiled only
B) Python is interpreted only
C) Python is both compiled and interpreted
D) Python is neither compiled nor interpreted

✅ **Answer:** C
➡️ *Python code is compiled to bytecode, then interpreted by the Python Virtual Machine.*

7️⃣ What does list(range(0, 10, 2)) return?

A) [1, 3, 5, 7, 9]
B) [0, 2, 4, 6, 8]
C) [0, 2, 4, 6, 8, 10]
D) range(0, 10, 2)
✅ **Answer:** B

8️⃣ What is the result of:

print(type({}))
A) <class 'set'>
B) <class 'dict'>
C) <class 'list'>
D) <class 'tuple'>
✅ **Answer:** B
➡️ *Empty braces {} create a dictionary, not a set.*

9️⃣ Which of the following will NOT produce an error?

A) int('xyz')
B) int(10.5)
C) int('10a')
D) int('10.5')
✅ **Answer:** B
➡️ *Only numeric float → int conversion works.*

🔟 What is the output of:

a = [1, 2, 3]
print(a[::-1])
A) [3, 2, 1]
B) [1, 2, 3]
C) [2, 1, 3]
D) Error
✅ **Answer:** A
➡️ *Slice [::-1] reverses the list.*

1️⃣1️⃣ Which method is used to add an element at a specific position in a list?

A) add()
B) append()
C) insert()
D) extend()
✅ **Answer:** C
➡️ *insert(pos, value) adds at a given index.*

1️⃣2️⃣ What will be the output of:

x = "Python"
print(x[1:4])
A) yth
B) Pyt
C) ytho
D) yto
✅ **Answer:** A
➡️ *Slicing excludes the end index.*

1️⃣3️⃣ Which of the following is immutable?

A) list
B) dict
C) tuple
D) set
✅ **Answer:** C

14️⃣ What will be the output of:

```
print(bool([]))
```

A) True
B) False
C) None
D) Error
✅ **Answer:** B
➡️ *Empty containers are considered False.*

15️⃣ What will len("Hello\nWorld") return?

A) 10
B) 11
C) 12
D) 9
✅ **Answer:** B
➡️ *\n counts as one character.*

16️⃣ What is the result of:

```
print(2 ** 3 ** 2)
```

A) 512
B) 64
C) 16
D) 8
✅ **Answer:** A
➡️ *Exponentiation is right-associative → 2^(3^2) = 2^9 = 512.*

17️⃣ Which of the following can be used to create a dictionary?

A) {}
B) dict()
C) {"a":1, "b":2}
D) All of the above
✅ **Answer:** D

18️⃣ What will be the output of:

```
a = [1, 2, 3]
a[1:3] = [5, 6, 7]
print(a)
```

A) [1, 5, 6, 7]
B) [1, 5, 6, 7, 3]
C) [5, 6, 7]
D) [1, 5, 6, 7, 2, 3]
✅ **Answer:** A
➡️ *Slicing replaces elements 1 and 2 with the new list.*

19️⃣ Which of these is a valid variable name in Python?

A) 1value
B) value_1
C) value-1
D) @value
✅ **Answer:** B

2️⃣0️⃣ What is the output of:

print('abc' * 2)
A) 'aabbcc'
B) 'abcabc'
C) 'abc2'
D) Error
✅ **Answer:** B

2️⃣1️⃣ What does id() function do in Python?

A) Returns the memory address of an object
B) Returns the name of a variable
C) Returns object's hash value
D) Deletes an object
✅ **Answer:** A

2️⃣2️⃣ What will set([1,2,2,3,3]) return?

A) {1,2,2,3,3}
B) {1,2,3}
C) [1,2,3]
D) Error
✅ **Answer:** B

2️⃣3️⃣ What will be the output of:

x = [1, 2, 3]
print(x.pop())
A) 1
B) 2
C) 3
D) Error
✅ **Answer:** C

2️⃣4️⃣ What will print(type(lambda x: x**2)) output?

A) <class 'lambda'>
B) <class 'function'>
C) <class 'method'>
D) Error
✅ **Answer:** B

2️⃣5️⃣ What will be printed:

print(bool("False"))
A) True
B) False
C) None
D) Error
✅ **Answer:** A
➡️ *Non-empty strings are always True.*

2️⃣6️⃣ What will be the output of:

x = [10, 20, 30]
y = x.copy()
y.append(40)
print(x)
A) [10, 20, 30, 40]
B) [10, 20, 30]
C) [40]

D) Error
✅ **Answer:** B
➡️ *copy() creates a shallow copy; appending to y does not affect x.*

2️⃣7️⃣ Which function returns the number of items in an object?

A) size()
B) length()
C) count()
D) len()
✅ **Answer:** D

2️⃣8️⃣ What will be the output of:

print({1, 2, 3} & {2, 3, 4})
A) {1,2,3,4}
B) {2,3}
C) {1,4}
D) Error
✅ **Answer:** B
➡️ *& performs set intersection.*

2️⃣9️⃣ What is the output of:

print([i for i in range(5) if i % 2 == 0])
A) [0, 1, 2, 3, 4]
B) [1, 3]
C) [0, 2, 4]
D) [2, 4]
✅ **Answer:** C

3️⃣0️⃣ Which statement is true about Python strings?

A) Strings are mutable
B) Strings are immutable
C) Strings are numeric arrays
D) Strings must be ASCII
✅ **Answer:** B

3️⃣1️⃣ What will the following code print?

a = [1, 2]
b = [1, 2]
print(a is b)
A) True
B) False
C) [1, 2]
D) Error
✅ **Answer:** B
➡️ *is checks identity, not equality.*

3️⃣2️⃣ What is the output of:

print("abc".upper())
A) abc
B) Abc
C) ABC
D) Error
✅ **Answer:** C

3️⃣3️⃣ What does strip() method do?

A) Removes spaces from both ends
B) Removes characters from start

C) Removes only newline
D) Removes all spaces in string
✅ **Answer:** A

3️⃣4️⃣ What is the output of:

```
print(sum([1, 2, 3], 10))
```

A) 6
B) 16
C) 10
D) Error
✅ **Answer:** B
➡️ *The second argument is the start value (10 + 1 + 2 + 3).*

3️⃣5️⃣ What will print(bool(0)) return?

A) True
B) False
C) None
D) Error
✅ **Answer:** B

3️⃣6️⃣ What will the following print?

```
a = [1, 2, 3]
print(a[3:])
```

A) []
B) [3]
C) Error
D) [1, 2, 3]
✅ **Answer:** A
➡️ *Slicing beyond the length gives an empty list.*

3️⃣7️⃣ What will be printed?

```
def f(x=[]):
    x.append(1)
    return x
print(f(), f(), f())
```

A) [1] [1] [1]
B) [1] [1,1] [1,1,1]
C) [1,1,1] [1,1,1] [1,1,1]
D) Error
✅ **Answer:** B
➡️ *Default list persists across calls.*

3️⃣8️⃣ What is the output of:

```
print(all([True, 1, 3]))
```

A) True
B) False
C) Error
D) None
✅ **Answer:** A
➡️ *All elements are truthy.*

3️⃣9️⃣ What will any([0, '', False]) return?

A) True
B) False
C) None
D) Error

✅ **Answer:** B
➡️ *All elements are falsy.*

4️⃣0️⃣ What is the output of:

```
x = [1, 2, 3]
print(x[1:])
```

A) [2, 3]
B) [1]
C) [1, 2]
D) [3]
✅ **Answer:** A

4️⃣1️⃣ What will print(list("abc")) output?

A) ['abc']
B) ['a', 'b', 'c']
C) ['a b c']
D) Error
✅ **Answer:** B

4️⃣2️⃣ What will be the output of:

```
a = [1, 2, 3]
b = a[:]
b.append(4)
print(len(a), len(b))
```

A) 3 3
B) 3 4
C) 4 4
D) Error
✅ **Answer:** B
➡️ *Slicing creates a shallow copy.*

4️⃣3️⃣ What will print(type((1))) output?

A) <class 'tuple'>
B) <class 'int'>
C) <class 'list'>
D) <class 'set'>
✅ **Answer:** B
➡️ *Parentheses alone don't create tuples without a comma.*

4️⃣4️⃣ What is the output of:

```
print('A' < 'a')
```

A) True
B) False
C) Error
D) None
✅ **Answer:** A
➡️ *ASCII value of uppercase 'A' < lowercase 'a'.*

4️⃣5️⃣ What will the following print?

```
for i in range(3):
    print(i, end=',')
```

A) 0,1,2,
B) 012
C) 0,1,2
D) Error

✅ **Answer:** A
➡️ *end=',' replaces newline.*

4️⃣6️⃣ What is the output of:

print([i**2 for i in range(4)])
A) [1, 4, 9, 16]
B) [0, 1, 4, 9]
C) [0, 1, 4, 9, 16]
D) Error
✅ **Answer:** B

4️⃣7️⃣ What is the result of:

print("Hello {0}!".format("World"))
A) Hello 0!
B) Hello {World}!
C) Hello World!
D) Error
✅ **Answer:** C

4️⃣8️⃣ What is the output of:

print(list(map(lambda x: x+1, [1,2,3])))
A) [2,3,4]
B) [1,2,3]
C) [1,3,5]
D) Error
✅ **Answer:** A

4️⃣9️⃣ What will reversed([1,2,3]) return?

A) [3,2,1]
B) reversed object
C) (3,2,1)
D) None
✅ **Answer:** B
➡️ *It returns a reverse iterator, not a list.*

5️⃣0️⃣ What is the output of:

a = [1, 2, 3]
print(a.clear())
A) []
B) None
C) Error
D) [1, 2, 3]
✅ **Answer:** B
➡️ *clear() empties list in place and returns None.*

SQL SUBJECTIVE

1️⃣ **Find the Second Highest Salary**
Table: Employee(emp_id, name, salary)
Query:

```
SELECT MAX(salary) AS SecondHighest
FROM Employee
WHERE salary < (SELECT MAX(salary) FROM Employee);
```

2️⃣ **Find Employees Who Earn More Than Their Manager**
Table: Employee(emp_id, name, manager_id, salary)
Query:
```
SELECT e.name
FROM Employee e
JOIN Employee m ON e.manager_id = m.emp_id
WHERE e.salary > m.salary;
```
3️⃣ **Find Duplicate Email IDs**
Table: Users(id, email)
Query:
```
SELECT email, COUNT(*) AS count
FROM Users
GROUP BY email
HAVING COUNT(*) > 1;
```
4️⃣ **Display the nth Highest Salary (e.g., 3rd highest)**
Table: Employee(emp_id, name, salary)
Query:
```
SELECT DISTINCT salary
FROM Employee e1
WHERE 3 = (
  SELECT COUNT(DISTINCT salary)
  FROM Employee e2
  WHERE e2.salary >= e1.salary
);
```
5️⃣ **Find Departments with Average Salary > 50000**
Table: Employee(emp_id, name, dept_id, salary)
Query:
```
SELECT dept_id, AVG(salary) AS avg_salary
FROM Employee
GROUP BY dept_id
HAVING AVG(salary) > 50000;
```
6️⃣ **Retrieve Employees Who Joined in the Last 6 Months**
Table: Employee(emp_id, name, join_date)
Query:
```
SELECT *
FROM Employee
WHERE join_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH);
```
7️⃣ **Get Total Salary Paid per Department**
Table: Employee(emp_id, name, dept_id, salary)
Query:
```
SELECT dept_id, SUM(salary) AS total_salary
FROM Employee
GROUP BY dept_id;
```
8️⃣ **Find All Employees Without a Manager**
Table: Employee(emp_id, name, manager_id)
Query:
```
SELECT name
FROM Employee
WHERE manager_id IS NULL;
```
9️⃣ **Find Customers Who Have Never Placed an Order**

Tables:
Customers(cust_id, cust_name)
Orders(order_id, cust_id)
Query:

```
SELECT c.cust_name
FROM Customers c
LEFT JOIN Orders o ON c.cust_id = o.cust_id
WHERE o.cust_id IS NULL;
```

🔟 **Find Top 3 Highest Paid Employees per Department**
Table: Employee(emp_id, name, dept_id, salary)
Query:

```
SELECT *
FROM (
  SELECT emp_id, name, dept_id, salary,
        RANK() OVER (PARTITION BY dept_id ORDER BY salary DESC) AS rnk
  FROM Employee
) t
WHERE rnk <= 3;
```

1️⃣1️⃣ **Find Departments Having More Than 5 Employees**
Query:

```
SELECT dept_id, COUNT(*) AS num_employees
FROM Employee
GROUP BY dept_id
HAVING COUNT(*) > 5;
```

1️⃣2️⃣ **Find Employees Who Have the Same Salary**
Query:

```
SELECT salary, COUNT(*) AS num_people
FROM Employee
GROUP BY salary
HAVING COUNT(*) > 1;
```

1️⃣3️⃣ **Find the Employee with the Longest Name**
Query:

```
SELECT name
FROM Employee
ORDER BY LENGTH(name) DESC
LIMIT 1;
```

1️⃣4️⃣ **Retrieve the 3 Most Recent Orders**
Table: Orders(order_id, order_date)
Query:

```
SELECT *
FROM Orders
ORDER BY order_date DESC
LIMIT 3;
```

1️⃣5️⃣ **Find Average Salary of Employees per Manager**
Query:

```
SELECT manager_id, AVG(salary) AS avg_salary
FROM Employee
WHERE manager_id IS NOT NULL
GROUP BY manager_id;
```

1️⃣6️⃣ **Get the Name of Employees Whose Salary Is Above the Company Average**
Query:

```
SELECT name, salary
```

```
FROM Employee
WHERE salary > (SELECT AVG(salary) FROM Employee);
```

17️⃣ Find Departments with No Employees

Tables: Department(dept_id, dept_name), Employee(emp_id, dept_id)

Query:

```
SELECT d.dept_name
FROM Department d
LEFT JOIN Employee e ON d.dept_id = e.dept_id
WHERE e.dept_id IS NULL;
```

18️⃣ Find the Total Number of Orders per Customer

Tables: Orders(order_id, cust_id)

Query:

```
SELECT cust_id, COUNT(order_id) AS total_orders
FROM Orders
GROUP BY cust_id;
```

19️⃣ Get the Product with the Highest Price per Category

Tables: Products(prod_id, category, price)

Query:

```
SELECT *
FROM (
  SELECT prod_id, category, price,
       RANK() OVER (PARTITION BY category ORDER BY price DESC) AS rnk
  FROM Products
) t
WHERE rnk = 1;
```

20️⃣ Retrieve Employees Hired in the Same Year

Query:

```
SELECT YEAR(join_date) AS hire_year, COUNT(*) AS num_employees
FROM Employee
GROUP BY YEAR(join_date);
```

21️⃣ Get Customer Who Spent the Most Money

Tables: Orders(order_id, cust_id, amount)

Query:

```
SELECT cust_id, SUM(amount) AS total_spent
FROM Orders
GROUP BY cust_id
ORDER BY total_spent DESC
LIMIT 1;
```

22️⃣ Find the Difference Between Highest and Lowest Salary in Each Department

Query:

```
SELECT dept_id, MAX(salary) - MIN(salary) AS salary_diff
FROM Employee
GROUP BY dept_id;
```

23️⃣ Retrieve Employees Who Share the Same Department and Salary

Query:

```
SELECT dept_id, salary, COUNT(*) AS count
FROM Employee
GROUP BY dept_id, salary
HAVING COUNT(*) > 1;
```

24️⃣ Find Customers Who Ordered All Products

Tables:
Orders(order_id, cust_id, prod_id)
Products(prod_id)
Query:

```
SELECT cust_id
FROM Orders
GROUP BY cust_id
HAVING COUNT(DISTINCT prod_id) = (SELECT COUNT(*) FROM Products);
```

25 **Retrieve the Longest Tenured Employee per Department**
Query:

```
SELECT *
FROM (
  SELECT emp_id, name, dept_id, join_date,
        RANK() OVER (PARTITION BY dept_id ORDER BY join_date ASC) AS rnk
  FROM Employee
) t
WHERE rnk = 1;
```

26 **Find Products Never Ordered**
Tables: Products(prod_id), Orders(order_id, prod_id)
Query:

```
SELECT p.prod_id
FROM Products p
LEFT JOIN Orders o ON p.prod_id = o.prod_id
WHERE o.prod_id IS NULL;
```

27 **Retrieve Departments with the Highest Average Salary**
Query:

```
SELECT dept_id, AVG(salary) AS avg_salary
FROM Employee
GROUP BY dept_id
ORDER BY avg_salary DESC
LIMIT 1;
```

28 **Get Employees Working Under More Than One Manager**
Query:

```
SELECT emp_id
FROM Employee
GROUP BY emp_id
HAVING COUNT(DISTINCT manager_id) > 1;
```

29 **Find the Cumulative Sum of Salaries by Department**
Query:

```
SELECT emp_id, dept_id, salary,
      SUM(salary) OVER (PARTITION BY dept_id ORDER BY emp_id) AS cumulative_salary
FROM Employee;
```

30 **Find the Month with Maximum Sales**
Tables: Sales(sale_id, sale_date, amount)
Query:

```
SELECT MONTH(sale_date) AS month, SUM(amount) AS total_sales
FROM Sales
GROUP BY MONTH(sale_date)
ORDER BY total_sales DESC
LIMIT 1;
```

PYTHON SUBJECTIVE CODE

1️⃣ **Problem: Find All Prime Numbers in a Range**

Input:

10 30

Output:

11 13 17 19 23 29

Solution:

```
def primes_in_range(a, b):
    for n in range(a, b+1):
        if n > 1:
            for i in range(2, int(n**0.5)+1):
                if n % i == 0:
                    break
            else:
                print(n, end=" ")
a, b = map(int, input().split())
primes_in_range(a, b)
```

2️⃣ **Problem: Check if a String is Pangram**

(A pangram contains every letter of the alphabet at least once.)

Input:

The quick brown fox jumps over the lazy dog

Output:

Pangram

Solution:

```
import string
s = input().lower()
print("Pangram" if set(string.ascii_lowercase) <= set(s) else "Not Pangram")
```

3️⃣ **Problem: Find the Frequency of Each Word in a Sentence**

Input:

hello world hello python

Output:

hello:2 world:1 python:1

Solution:

```
from collections import Counter
words = input().split()
for word, count in Counter(words).items():
    print(f"{word}:{count}", end=" ")
```

4️⃣ **Problem: Find Intersection of Two Lists**

Input:

1 2 3 4

3 4 5 6

Output:

3 4

Solution:

```
a = set(map(int, input().split()))
b = set(map(int, input().split()))
print(*sorted(a & b))
```

5️⃣ **Problem: Find All Pairs with Given Sum**

Input:

6
1 2 3 4 5 6
7

Output:

(1,6) (2,5) (3,4)

Solution:

```
n = int(input())
arr = list(map(int, input().split()))
target = int(input())
for i in range(n):
    for j in range(i+1, n):
        if arr[i] + arr[j] == target:
            print(f"({arr[i]},{arr[j]})", end=" ")
```

6️⃣ **Problem: Find the Longest Word in a Sentence**

Input:

Python makes coding interesting

Output:

interesting

Solution:

```
s = input().split()
print(max(s, key=len))
```

7️⃣ **Problem: Print Pascal's Triangle**

Input:

5

Output:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Solution:

```
def pascal(n):
    for i in range(n):
        num = 1
        for j in range(i+1):
            print(num, end=" ")
            num = num * (i - j) // (j + 1)
        print()
n = int(input())
pascal(n)
```

8️⃣ **Problem: Find GCD of Two Numbers**

Input:

24 36

Output:

12

Solution:

```
import math
a, b = map(int, input().split())
print(math.gcd(a, b))
```

9️⃣ **Problem: Count the Number of Words and Characters**

Input:
I love programming
Output:
Words: 3 Characters: 18
Solution:
```
s = input()
print("Words:", len(s.split()), "Characters:", len(s))
```
🔟 **Problem: Sort Dictionary by Values**
Input:
a 3 b 1 c 2
Output:
{'b': 1, 'c': 2, 'a': 3}
Solution:
```
data = input().split()
d = {data[i]: int(data[i+1]) for i in range(0, len(data), 2)}
print(dict(sorted(d.items(), key=lambda x: x[1])))
```
1️⃣1️⃣ **Problem: Count Unique Words in a Sentence**
Input:
Python is easy and Python is powerful
Output:
5
Solution:
```
s = input().lower().split()
print(len(set(s)))
```
1️⃣2️⃣ **Problem: Convert a List into a Dictionary**
Input:
a b c
1 2 3
Output:
{'a': 1, 'b': 2, 'c': 3}
Solution:
```
keys = input().split()
values = list(map(int, input().split()))
print(dict(zip(keys, values)))
```
1️⃣3️⃣ **Problem: Find the Missing Number in Sequence**
Input:
1 2 4 5 6
Output:
3
Solution:
```
arr = list(map(int, input().split()))
for i in range(min(arr), max(arr)):
    if i not in arr:
        print(i)
        break
```
1️⃣4️⃣ **Problem: Find All Duplicates in a List**
Input:
1 2 3 2 4 1 5
Output:
1 2
Solution:
```
arr = list(map(int, input().split()))
```

```
dup = [x for x in set(arr) if arr.count(x) > 1]
print(*sorted(dup))
```

15️⃣ Problem: Check if a String is Anagram

Input:

listen
silent

Output:

Anagram

Solution:

```
s1 = input()
s2 = input()
print("Anagram" if sorted(s1) == sorted(s2) else "Not Anagram")
```

16️⃣ Problem: Find the Sum of Diagonals of a Square Matrix

Input:

3
1 2 3
4 5 6
7 8 9

Output:

Sum: 30

Solution:

```
n = int(input())
mat = [list(map(int, input().split())) for _ in range(n)]
s = sum(mat[i][i] + mat[i][n-1-i] for i in range(n))
if n % 2 != 0:
    s -= mat[n//2][n//2]
print("Sum:", s)
```

17️⃣ Problem: Count Capital Letters in String

Input:

HelloWorld

Output:

2

Solution:

```
s = input()
print(sum(1 for c in s if c.isupper()))
```

18️⃣ Problem: Find All Armstrong Numbers in a Range

Input:

1 500

Output:

1 153 370 371 407

Solution:

```
a, b = map(int, input().split())
for num in range(a, b+1):
    power = len(str(num))
    if sum(int(d)**power for d in str(num)) == num:
        print(num, end=" ")
```

19️⃣ Problem: Rotate Array by K Positions

Input:

5
1 2 3 4 5
2

Output:

4 5 1 2 3
Solution:
```
n = int(input())
arr = list(map(int, input().split()))
k = int(input()) % n
print(*(arr[-k:] + arr[:-k]))
```
20 Problem: Find the Most Frequent Character
Input:
mississippi
Output:
i
Solution:
```
from collections import Counter
s = input()
print(Counter(s).most_common(1)[0][0])
```
21 Problem: Find Sum of Elements Above Main Diagonal
Input:
3
1 2 3
4 5 6
7 8 9
Output:
11
Solution:
```
n = int(input())
mat = [list(map(int, input().split())) for _ in range(n)]
print(sum(mat[i][j] for i in range(n) for j in range(i+1, n)))
```
22 Problem: Merge and Sort Two Lists
Input:
1 3 5
2 4 6
Output:
1 2 3 4 5 6
Solution:
```
a = list(map(int, input().split()))
b = list(map(int, input().split()))
print(*sorted(a + b))
```
23 Problem: Remove All Punctuation from String
Input:
Hello, world! Welcome.
Output:
Hello world Welcome
Solution:
```
import string
s = input()
print(''.join(ch for ch in s if ch not in string.punctuation))
```
24 Problem: Find the Sum of Numbers in a String
Input:
abc12xyz5
Output:
17
Solution:

```
import re
s = input()
nums = map(int, re.findall(r'\d+', s))
print(sum(nums))
```

25 Problem: Convert Decimal to Binary Without Built-in

Input:

10

Output:

1010

Solution:

```
n = int(input())
res = ''
while n > 0:
    res = str(n % 2) + res
    n //= 2
print(res)
```

26 Problem: Sort Words Alphabetically

Input:

banana apple orange

Output:

apple banana orange

Solution:

```
s = input().split()
print(*sorted(s))
```

27 Problem: Remove Duplicate Characters from a String

Input:

programming

Output:

progamin

Solution:

```
s = input()
res = ''
for ch in s:
    if ch not in res:
        res += ch
print(res)
```

28 Problem: Check Substring Occurrence Count

Input:

abababa
aba

Output:

3

Solution:

```
s = input()
sub = input()
count = 0
for i in range(len(s) - len(sub) + 1):
    if s[i:i+len(sub)] == sub:
        count += 1
print(count)
```

29 Problem: Print All Substrings of a String

Input:

abc
Output:
a ab abc b bc c
Solution:

```
s = input()
for i in range(len(s)):
    for j in range(i+1, len(s)+1):
        print(s[i:j], end=" ")
```

3️⃣0️⃣ **Problem: Find All Unique Permutations of a String**
Input:
abc
Output:
abc acb bac bca cab cba
Solution:

```
from itertools import permutations
s = input()
for p in sorted(set([''.join(x) for x in permutations(s)])):
    print(p, end=" ")
```

SQL MCQ

1️⃣ **Find the Second Highest Salary**
Table: Employee(emp_id, name, salary)
Query:

```
SELECT MAX(salary) AS SecondHighest
FROM Employee
WHERE salary < (SELECT MAX(salary) FROM Employee);
```

2️⃣ **Find Employees Who Earn More Than Their Manager**
Table: Employee(emp_id, name, manager_id, salary)
Query:

```
SELECT e.name
FROM Employee e
JOIN Employee m ON e.manager_id = m.emp_id
WHERE e.salary > m.salary;
```

3️⃣ **Find Duplicate Email IDs**
Table: Users(id, email)
Query:

```
SELECT email, COUNT(*) AS count
FROM Users
GROUP BY email
HAVING COUNT(*) > 1;
```

4️⃣ **Display the nth Highest Salary (e.g., 3rd highest)**
Table: Employee(emp_id, name, salary)
Query:

```
SELECT DISTINCT salary
FROM Employee e1
WHERE 3 = (
  SELECT COUNT(DISTINCT salary)
  FROM Employee e2
  WHERE e2.salary >= e1.salary
);
```

5️⃣ **Find Departments with Average Salary > 50000**

Table: Employee(emp_id, name, dept_id, salary)
Query:
```
SELECT dept_id, AVG(salary) AS avg_salary
FROM Employee
GROUP BY dept_id
HAVING AVG(salary) > 50000;
```
6️⃣ **Retrieve Employees Who Joined in the Last 6 Months**
Table: Employee(emp_id, name, join_date)
Query:
```
SELECT *
FROM Employee
WHERE join_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH);
```
7️⃣ **Get Total Salary Paid per Department**
Table: Employee(emp_id, name, dept_id, salary)
Query:
```
SELECT dept_id, SUM(salary) AS total_salary
FROM Employee
GROUP BY dept_id;
```
8️⃣ **Find All Employees Without a Manager**
Table: Employee(emp_id, name, manager_id)
Query:
```
SELECT name
FROM Employee
WHERE manager_id IS NULL;
```
9️⃣ **Find Customers Who Have Never Placed an Order**
Tables:
Customers(cust_id, cust_name)
Orders(order_id, cust_id)
Query:
```
SELECT c.cust_name
FROM Customers c
LEFT JOIN Orders o ON c.cust_id = o.cust_id
WHERE o.cust_id IS NULL;
```
🔟 **Find Top 3 Highest Paid Employees per Department**
Table: Employee(emp_id, name, dept_id, salary)
Query:
```
SELECT *
FROM (
  SELECT emp_id, name, dept_id, salary,
        RANK() OVER (PARTITION BY dept_id ORDER BY salary DESC) AS rnk
  FROM Employee
) t
WHERE rnk <= 3;
```
1️⃣1️⃣ **Find Departments Having More Than 5 Employees**
Query:
```
SELECT dept_id, COUNT(*) AS num_employees
FROM Employee
GROUP BY dept_id
HAVING COUNT(*) > 5;
```
1️⃣2️⃣ **Find Employees Who Have the Same Salary**
Query:
```
SELECT salary, COUNT(*) AS num_people
```

```
FROM Employee
GROUP BY salary
HAVING COUNT(*) > 1;
```

13️⃣ Find the Employee with the Longest Name

Query:

```
SELECT name
FROM Employee
ORDER BY LENGTH(name) DESC
LIMIT 1;
```

14️⃣ Retrieve the 3 Most Recent Orders

Table: Orders(order_id, order_date)

Query:

```
SELECT *
FROM Orders
ORDER BY order_date DESC
LIMIT 3;
```

15️⃣ Find Average Salary of Employees per Manager

Query:

```
SELECT manager_id, AVG(salary) AS avg_salary
FROM Employee
WHERE manager_id IS NOT NULL
GROUP BY manager_id;
```

16️⃣ Get the Name of Employees Whose Salary Is Above the Company Average

Query:

```
SELECT name, salary
FROM Employee
WHERE salary > (SELECT AVG(salary) FROM Employee);
```

17️⃣ Find Departments with No Employees

Tables: Department(dept_id, dept_name), Employee(emp_id, dept_id)

Query:

```
SELECT d.dept_name
FROM Department d
LEFT JOIN Employee e ON d.dept_id = e.dept_id
WHERE e.dept_id IS NULL;
```

18️⃣ Find the Total Number of Orders per Customer

Tables: Orders(order_id, cust_id)

Query:

```
SELECT cust_id, COUNT(order_id) AS total_orders
FROM Orders
GROUP BY cust_id;
```

19️⃣ Get the Product with the Highest Price per Category

Tables: Products(prod_id, category, price)

Query:

```
SELECT *
FROM (
  SELECT prod_id, category, price,
       RANK() OVER (PARTITION BY category ORDER BY price DESC) AS rnk
  FROM Products
) t
WHERE rnk = 1;
```

20️⃣ Retrieve Employees Hired in the Same Year

Query:

```
SELECT YEAR(join_date) AS hire_year, COUNT(*) AS num_employees
FROM Employee
GROUP BY YEAR(join_date);
```

21️⃣ Get Customer Who Spent the Most Money

Tables: Orders(order_id, cust_id, amount)

Query:

```
SELECT cust_id, SUM(amount) AS total_spent
FROM Orders
GROUP BY cust_id
ORDER BY total_spent DESC
LIMIT 1;
```

22️⃣ Find the Difference Between Highest and Lowest Salary in Each Department

Query:

```
SELECT dept_id, MAX(salary) - MIN(salary) AS salary_diff
FROM Employee
GROUP BY dept_id;
```

23️⃣ Retrieve Employees Who Share the Same Department and Salary

Query:

```
SELECT dept_id, salary, COUNT(*) AS count
FROM Employee
GROUP BY dept_id, salary
HAVING COUNT(*) > 1;
```

24️⃣ Find Customers Who Ordered All Products

Tables:

Orders(order_id, cust_id, prod_id)

Products(prod_id)

Query:

```
SELECT cust_id
FROM Orders
GROUP BY cust_id
HAVING COUNT(DISTINCT prod_id) = (SELECT COUNT(*) FROM Products);
```

25️⃣ Retrieve the Longest Tenured Employee per Department

Query:

```
SELECT *
FROM (
  SELECT emp_id, name, dept_id, join_date,
        RANK() OVER (PARTITION BY dept_id ORDER BY join_date ASC) AS rnk
  FROM Employee
) t
WHERE rnk = 1;
```

26️⃣ Find Products Never Ordered

Tables: Products(prod_id), Orders(order_id, prod_id)

Query:

```
SELECT p.prod_id
FROM Products p
LEFT JOIN Orders o ON p.prod_id = o.prod_id
WHERE o.prod_id IS NULL;
```

27️⃣ Retrieve Departments with the Highest Average Salary

Query:

```
SELECT dept_id, AVG(salary) AS avg_salary
FROM Employee
GROUP BY dept_id
```

```
ORDER BY avg_salary DESC
LIMIT 1;
```

28 Get Employees Working Under More Than One Manager

Query:

```
SELECT emp_id
FROM Employee
GROUP BY emp_id
HAVING COUNT(DISTINCT manager_id) > 1;
```

29 Find the Cumulative Sum of Salaries by Department

Query:

```
SELECT emp_id, dept_id, salary,
    SUM(salary) OVER (PARTITION BY dept_id ORDER BY emp_id) AS cumulative_salary
FROM Employee;
```

30 Find the Month with Maximum Sales

Tables: Sales(sale_id, sale_date, amount)

Query:

```
SELECT MONTH(sale_date) AS month, SUM(amount) AS total_sales
FROM Sales
GROUP BY MONTH(sale_date)
ORDER BY total_sales DESC
LIMIT 1;
```

1 Problem: Find All Prime Numbers in a Range

Input:

```
10 30
```

Output:

```
11 13 17 19 23 29
```

Solution:

```
def primes_in_range(a, b):
    for n in range(a, b+1):
        if n > 1:
            for i in range(2, int(n**0.5)+1):
                if n % i == 0:
                    break
            else:
                print(n, end=" ")
a, b = map(int, input().split())
primes_in_range(a, b)
```

2 Problem: Check if a String is Pangram

(A pangram contains every letter of the alphabet at least once.)

Input:

```
The quick brown fox jumps over the lazy dog
```

Output:

```
Pangram
```

Solution:

```
import string
s = input().lower()
print("Pangram" if set(string.ascii_lowercase) <= set(s) else "Not Pangram")
```

3 Problem: Find the Frequency of Each Word in a Sentence

Input:

```
hello world hello python
```

Output:

```
hello:2 world:1 python:1
```

Solution:

```
from collections import Counter
words = input().split()
for word, count in Counter(words).items():
    print(f"{word}:{count}", end=" ")
```

4 **Problem: Find Intersection of Two Lists**

Input:

1 2 3 4
3 4 5 6

Output:

3 4

Solution:

```
a = set(map(int, input().split()))
b = set(map(int, input().split()))
print(*sorted(a & b))
```

5 **Problem: Find All Pairs with Given Sum**

Input:

6
1 2 3 4 5 6
7

Output:

(1,6) (2,5) (3,4)

Solution:

```
n = int(input())
arr = list(map(int, input().split()))
target = int(input())
for i in range(n):
    for j in range(i+1, n):
        if arr[i] + arr[j] == target:
            print(f"({arr[i]},{arr[j]})", end=" ")
```

6 **Problem: Find the Longest Word in a Sentence**

Input:

Python makes coding interesting

Output:

interesting

Solution:

```
s = input().split()
print(max(s, key=len))
```

7 **Problem: Print Pascal's Triangle**

Input:

5

Output:

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Solution:

```
def pascal(n):
    for i in range(n):
        num = 1
        for j in range(i+1):
```

```
            print(num, end=" ")
            num = num * (i - j) // (j + 1)
        print()
n = int(input())
pascal(n)
```

8️⃣ **Problem: Find GCD of Two Numbers**

Input:

24 36

Output:

12

Solution:

```
import math
a, b = map(int, input().split())
print(math.gcd(a, b))
```

9️⃣ **Problem: Count the Number of Words and Characters**

Input:

I love programming

Output:

Words: 3 Characters: 18

Solution:

```
s = input()
print("Words:", len(s.split()), "Characters:", len(s))
```

🔟 **Problem: Sort Dictionary by Values**

Input:

a 3 b 1 c 2

Output:

{'b': 1, 'c': 2, 'a': 3}

Solution:

```
data = input().split()
d = {data[i]: int(data[i+1]) for i in range(0, len(data), 2)}
print(dict(sorted(d.items(), key=lambda x: x[1])))
```

1️⃣1️⃣ **Problem: Count Unique Words in a Sentence**

Input:

Python is easy and Python is powerful

Output:

5

Solution:

```
s = input().lower().split()
print(len(set(s)))
```

1️⃣2️⃣ **Problem: Convert a List into a Dictionary**

Input:

a b c

1 2 3

Output:

{'a': 1, 'b': 2, 'c': 3}

Solution:

```
keys = input().split()
values = list(map(int, input().split()))
print(dict(zip(keys, values)))
```

1️⃣3️⃣ **Problem: Find the Missing Number in Sequence**

Input:

1 2 4 5 6

Output:
3
Solution:
```
arr = list(map(int, input().split()))
for i in range(min(arr), max(arr)):
    if i not in arr:
        print(i)
        break
```
14 Problem: Find All Duplicates in a List
Input:
1 2 3 2 4 1 5
Output:
1 2
Solution:
```
arr = list(map(int, input().split()))
dup = [x for x in set(arr) if arr.count(x) > 1]
print(*sorted(dup))
```
15 Problem: Check if a String is Anagram
Input:
listen
silent
Output:
Anagram
Solution:
```
s1 = input()
s2 = input()
print("Anagram" if sorted(s1) == sorted(s2) else "Not Anagram")
```
16 Problem: Find the Sum of Diagonals of a Square Matrix
Input:
3
1 2 3
4 5 6
7 8 9
Output:
Sum: 30
Solution:
```
n = int(input())
mat = [list(map(int, input().split())) for _ in range(n)]
s = sum(mat[i][i] + mat[i][n-1-i] for i in range(n))
if n % 2 != 0:
    s -= mat[n//2][n//2]
print("Sum:", s)
```
17 Problem: Count Capital Letters in String
Input:
HelloWorld
Output:
2
Solution:
```
s = input()
print(sum(1 for c in s if c.isupper()))
```
18 Problem: Find All Armstrong Numbers in a Range
Input:

1 500
Output:
1 153 370 371 407
Solution:
```
a, b = map(int, input().split())
for num in range(a, b+1):
    power = len(str(num))
    if sum(int(d)**power for d in str(num)) == num:
        print(num, end=" ")
```
19 Problem: Rotate Array by K Positions
Input:
5
1 2 3 4 5
2
Output:
4 5 1 2 3
Solution:
```
n = int(input())
arr = list(map(int, input().split()))
k = int(input()) % n
print(*(arr[-k:] + arr[:-k]))
```
20 Problem: Find the Most Frequent Character
Input:
mississippi
Output:
i
Solution:
```
from collections import Counter
s = input()
print(Counter(s).most_common(1)[0][0])
```
21 Problem: Find Sum of Elements Above Main Diagonal
Input:
3
1 2 3
4 5 6
7 8 9
Output:
11
Solution:
```
n = int(input())
mat = [list(map(int, input().split())) for _ in range(n)]
print(sum(mat[i][j] for i in range(n) for j in range(i+1, n)))
```
22 Problem: Merge and Sort Two Lists
Input:
1 3 5
2 4 6
Output:
1 2 3 4 5 6
Solution:
```
a = list(map(int, input().split()))
b = list(map(int, input().split()))
print(*sorted(a + b))
```

23️⃣ Problem: Remove All Punctuation from String
Input:
Hello, world! Welcome.
Output:
Hello world Welcome
Solution:
```
import string
s = input()
print(''.join(ch for ch in s if ch not in string.punctuation))
```
24️⃣ Problem: Find the Sum of Numbers in a String
Input:
abc12xyz5
Output:
17
Solution:
```
import re
s = input()
nums = map(int, re.findall(r'\d+', s))
print(sum(nums))
```
25️⃣ Problem: Convert Decimal to Binary Without Built-in
Input:
10
Output:
1010
Solution:
```
n = int(input())
res = ''
while n > 0:
    res = str(n % 2) + res
    n //= 2
print(res)
```
26️⃣ Problem: Sort Words Alphabetically
Input:
banana apple orange
Output:
apple banana orange
Solution:
```
s = input().split()
print(*sorted(s))
```
27️⃣ Problem: Remove Duplicate Characters from a String
Input:
programming
Output:
progamin
Solution:
```
s = input()
res = ''
for ch in s:
    if ch not in res:
        res += ch
print(res)
```
28️⃣ Problem: Check Substring Occurrence Count

Input:
abababa
aba
Output:
3
Solution:
```
s = input()
sub = input()
count = 0
for i in range(len(s) - len(sub) + 1):
    if s[i:i+len(sub)] == sub:
        count += 1
print(count)
```
2️⃣9️⃣ **Problem: Print All Substrings of a String**
Input:
abc
Output:
a ab abc b bc c
Solution:
```
s = input()
for i in range(len(s)):
    for j in range(i+1, len(s)+1):
        print(s[i:j], end=" ")
```
3️⃣0️⃣ **Problem: Find All Unique Permutations of a String**
Input:
abc
Output:
abc acb bac bca cab cba
Solution:
```
from itertools import permutations
s = input()
for p in sorted(set([''.join(x) for x in permutations(s)])):
    print(p, end=" ")
```
1️⃣ **What does SQL stand for?**
A) Structured Query Language
B) Simple Query Language
C) Sequential Query Logic
D) Structured Question Language
✅ **Answer:** A
➡️ *SQL stands for Structured Query Language.*
2️⃣ **Which command is used to remove a table from a database?**
A) DELETE
B) DROP
C) REMOVE
D) ERASE
✅ **Answer:** B
➡️ *DROP TABLE table_name; removes the table structure completely.*
3️⃣ **Which of the following is not a type of SQL command?**
A) DDL
B) DML
C) DCL
D) DQLL

✅ **Answer:** D
➡️ *There's no DQLL; valid ones are DDL, DML, DCL, TCL, DQL.*

4️⃣ **Which SQL statement is used to retrieve data from a table?**
A) GET
B) SELECT
C) EXTRACT
D) FETCH
✅ **Answer:** B

5️⃣ **Which clause is used to filter records?**
A) ORDER BY
B) GROUP BY
C) WHERE
D) DISTINCT
✅ **Answer:** C

6️⃣ **Which keyword is used to sort the result set?**
A) ORDER
B) ORDER BY
C) SORT
D) SORT BY
✅ **Answer:** B

7️⃣ **Which function returns the number of rows in a table?**
A) COUNT()
B) SUM()
C) MAX()
D) LENGTH()
✅ **Answer:** A

8️⃣ **What is the default sort order of the ORDER BY clause?**
A) Descending
B) Random
C) Ascending
D) No order
✅ **Answer:** C

9️⃣ **Which operator is used to test for NULL values?**
A) =
B) IS
C) ==
D) EQUAL
✅ **Answer:** B
➡️ *Use IS NULL or IS NOT NULL.*

🔟 **What will SELECT * FROM Employees WHERE salary BETWEEN 3000 AND 5000; return?**
A) All employees
B) Employees with salary < 3000
C) Employees with salary 3000–5000
D) Error
✅ **Answer:** C

1️⃣1️⃣ **Which of the following is a DDL command?**
A) SELECT
B) INSERT
C) UPDATE
D) CREATE
✅ **Answer:** D
➡️ *DDL → CREATE, ALTER, DROP, TRUNCATE.*

12️⃣ Which SQL statement is used to modify data?
A) MODIFY
B) UPDATE
C) CHANGE
D) ALTER
✅ **Answer:** B
13️⃣ Which command removes all rows from a table but keeps the structure?
A) DROP
B) DELETE
C) TRUNCATE
D) ERASE
✅ **Answer:** C
14️⃣ Which keyword is used to combine rows from two or more tables?
A) UNION
B) COMBINE
C) GROUP
D) JOIN
✅ **Answer:** D
15️⃣ What does the GROUP BY clause do?
A) Filters rows
B) Sorts results
C) Groups rows with same values
D) Deletes duplicates
✅ **Answer:** C
16️⃣ Which constraint uniquely identifies each row in a table?
A) FOREIGN KEY
B) UNIQUE
C) PRIMARY KEY
D) CHECK
✅ **Answer:** C
17️⃣ Which command is used to add a new column?
A) ADD COLUMN
B) INSERT COLUMN
C) ALTER TABLE … ADD
D) MODIFY TABLE
✅ **Answer:** C
18️⃣ What will this query do?
SELECT DISTINCT city FROM customers;
A) Shows all cities
B) Shows duplicate cities
C) Shows unique cities only
D) Shows cities in uppercase
✅ **Answer:** C
19️⃣ Which keyword is used to remove duplicate rows in a query?
A) REMOVE
B) UNIQUE
C) DISTINCT
D) CLEAN
✅ **Answer:** C
20️⃣ Which of the following constraints allows only a specific range of values?
A) DEFAULT
B) CHECK

C) PRIMARY KEY
D) FOREIGN KEY
✅ **Answer:** B
2️⃣1️⃣ **What is the result of:**
SELECT COUNT(*) FROM employee WHERE dept_id IS NULL;
A) Counts all rows
B) Counts rows with NULL dept_id
C) Returns 0 always
D) Error
✅ **Answer:** B
2️⃣2️⃣ **Which function gives the current date?**
A) CURDATE()
B) SYSDATE()
C) NOW()
D) All of the above
✅ **Answer:** D
2️⃣3️⃣ **What is the purpose of a foreign key?**
A) To identify a row uniquely
B) To ensure referential integrity
C) To auto-increment a column
D) To store large text
✅ **Answer:** B
2️⃣4️⃣ **Which clause is used to filter grouped data?**
A) WHERE
B) HAVING
C) ORDER BY
D) FILTER
✅ **Answer:** B
➡️ *HAVING works with aggregate functions.*
2️⃣5️⃣ **What does SELECT COUNT(DISTINCT dept_id) return?**
A) Number of departments
B) Total employees
C) Sum of salaries
D) Maximum dept_id
✅ **Answer:** A
2️⃣6️⃣ **Which of the following commands commits a transaction?**
A) SAVE
B) COMMIT
C) APPLY
D) END
✅ **Answer:** B
2️⃣7️⃣ **What is a composite key?**
A) A foreign key
B) A key made of multiple columns
C) A key that can’t be NULL
D) A temporary key
✅ **Answer:** B
2️⃣8️⃣ **What is a view?**
A) A duplicate table
B) A virtual table based on query
C) A permanent copy

D) A backup table
✅ **Answer:** B
29️⃣ Which of the following statements is true?
A) A table can have multiple primary keys
B) A table can have multiple foreign keys
C) A table cannot have constraints
D) Foreign keys must be unique
✅ **Answer:** B
30️⃣ What is normalization?
A) Process of duplicating data
B) Process of minimizing redundancy
C) Process of deleting tables
D) Process of indexing
✅ **Answer:** B
31️⃣ Which normal form removes partial dependency?
A) 1NF
B) 2NF
C) 3NF
D) BCNF
✅ **Answer:** B
32️⃣ Which normal form removes transitive dependency?
A) 2NF
B) 3NF
C) BCNF
D) 4NF
✅ **Answer:** B
33️⃣ What does the LIKE operator do?
A) Compares exact match
B) Pattern matching with wildcards
C) Checks NULL values
D) Filters numbers
✅ **Answer:** B
34️⃣ What symbol represents a single character in LIKE pattern?
A) %
B) _
C) ?
D) #
✅ **Answer:** B
35️⃣ Which operator is used to combine multiple conditions?
A) AND / OR
B) BETWEEN
C) IS
D) EXISTS
✅ **Answer:** A
36️⃣ What is the use of IN operator?
A) Check range of numbers
B) Check if value exists in a list
C) Check NULL
D) Compare two tables
✅ **Answer:** B
37️⃣ Which index is automatically created with a primary key?

A) Non-clustered
B) Clustered
C) Hash
D) Bitmap
✅ **Answer:** B
38️⃣ Which SQL command is used to change data type of a column?
A) MODIFY
B) CHANGE
C) ALTER TABLE … MODIFY
D) UPDATE
✅ **Answer:** C
39️⃣ Which of the following is used to give a temporary name to a column?
A) ALIAS
B) TEMP
C) AS
D) NAME
✅ **Answer:** C
40️⃣ What will the query do?
SELECT name FROM employee WHERE name LIKE 'S%';
A) Names ending with S
B) Names starting with S
C) Names containing S
D) Error
✅ **Answer:** B
41️⃣ What is a subquery?
A) A query within another query
B) A stored procedure
C) A join operation
D) A transaction
✅ **Answer:** A
42️⃣ Which type of join returns all records from both tables?
A) INNER JOIN
B) LEFT JOIN
C) RIGHT JOIN
D) FULL OUTER JOIN
✅ **Answer:** D
43️⃣ Which join returns only matching records?
A) INNER JOIN
B) LEFT JOIN
C) RIGHT JOIN
D) FULL JOIN
✅ **Answer:** A
44️⃣ What is the purpose of the UNION operator?
A) Combine rows and remove duplicates
B) Combine columns
C) Join tables horizontally
D) Return intersection
✅ **Answer:** A
45️⃣ What does the EXISTS operator do?
A) Checks if subquery returns any rows
B) Checks for NULL
C) Deletes rows

D) Creates temporary tables
✅ **Answer:** A
4️⃣6️⃣ What is a transaction in SQL?
A) A single logical unit of work
B) A permanent table
C) A stored procedure
D) A rollback command
✅ **Answer:** A
4️⃣7️⃣ What command undoes changes made in a transaction?
A) CANCEL
B) DELETE
C) ROLLBACK
D) RESET
✅ **Answer:** C
4️⃣8️⃣ Which SQL clause restricts the number of rows returned?
A) RESTRICT
B) LIMIT
C) TOP
D) Both B and C
✅ **Answer:** D
➡️ *MySQL uses LIMIT, SQL Server uses TOP.*
4️⃣9️⃣ What does ACID stand for in databases?
A) Atomicity, Consistency, Isolation, Durability
B) Accuracy, Control, Integrity, Data
C) Access, Cache, Index, Durability
D) None of the above
✅ **Answer:** A
5️⃣0️⃣ Which of the following is true about foreign keys?
A) Can have NULL values
B) Must reference a primary key
C) Can enforce referential integrity
D) All of the above
✅ **Answer:** D