First and last name .....

Question 1/30 *(1 p.)*

Which SQL clause filters rows **post-aggregation**?

A. Where
B. Filter
C. Having
D. All of the above

Question 2/30 *(1 p.)*

When you are combining two tables together - one called the **parent** and the other called the **child** - and you want to include only the information from the parent that has matching information in the child, **which type of join** are you using?

A. **Left** Join
B. **Inner** Join
C. **Right** Join

Question 3/30 *(1 p.)*

Consider the following SQL query, which is designed to retrieve the **name** and **salary** of **employees** who earn **more than the average salary** in their respective **departments.**

```
SELECT e.name, e.salary
FROM employees e
WHERE e.salary > (
    SELECT AVG(e.salary)
    FROM employees
    WHERE department = e.department
);
```

A. The query will return the **correct results**, showing the employees who earn more than the average salary in their department
B. The query will fail due to **ambiguous column references** since both the **inner** and **outer queries** refer to the **employees** table
C. The query will fail because the **AVG function cannot** be used **with** the greater-than operator (>)
D. The query will return **incorrect results** because the **department** field is **not included** in the **GROUP BY** clause.

Question 4/30 *(1 p.)*

```
ls -l | grep source | sort | wc -l
```

Which of the following **statements are true** based on the Linux command? **Check all that apply.**

A. Four separate processes are **spawned**
B. Four **pipes** are created
C. All processes can process **data in parallel**
D. All processes **run in parallel**

Question 5/30 *(1 p.)*

Which of the following commands will create a new git branch and switch to it?

A. git **branch** <branch_name>
B. git **checkout** <branch_name>
C. git **branch -b** <branch_name>
D. git **checkout -b** <branch_name>
E. git **checkout branch -b** <branch_name>

Question 6/30 *(2 p.)*

You are working on a feature in a new branch and have just committed your changes. You find out that the main branch has been updated. What command can you use to **get those updates into your branch without merging** ?

A. git **pull origin** main
B. git **merge** main
C. git **rebase** main
D. git **fetch** main

Question 7/30 *(1 p.)*

Regarding **pull requests**, which of the following statements are true? **Check all that apply**

A. Pull requests must **compare two branches**
B. Pull requests can be made from the **Git CLI**
C. **git pull** is the command to execute a pull request

Question 8/30 *(1 p.)*

The following sequence of commands is valid for downloading code from GitHub, making edits and uploading changes back to the remote repository.

```
git clone <repository-url>

cd <repository-name>

touch a.txt

git add a.txt

git commit -m "adds a.txt"

git push
```

A. True
B. False

Question 9/30 *(1 p.)*

In **GitHub**, how can you automatically **close an issue** using a **pull request**?

A. Add "**Closes #issue_number**" in the pull request description
B. Use the command `**git pull** --close <issue_number>` in the terminal
C. Use the command `**git close** <issue_number>` in the terminal
D. Send an **email** to GitHub support

Question 10/30 *(1 p.)*

What is used to separate multiple **parameters** in a URL?

A. **?**
B. **&**
C. |

Question 11/30 *(1 p.)*

Which **HTTP** method is often used to submit form data?

A. **PUT**
B. **GET**
C. **POST**
D. **DELETE**

Question 12/30 *(1 p.)*

In which **architecture** model are messages written to a topic or queue and **not deleted after readers have read them**?

A. **Producer-Consumer** Model
B. **Publisher-Subscriber** Model

Question 13/30 *(1 p.)*

Rank the following from **slowest to fastest** in terms of performance.

A. SSD, HDD, memory
B. HDD, SSD, memory
C. Memory, SSD, HDD
D. Memory, HDD, SSD
E. SSD, memory, HDD

Question 14/30 *(1 p.)*

A **container** is a virtualization / emulation of a physical hardware computer, and a **VM (Virtual Machine)** is the virtualization of an OS (Operating System)

A. True
B. False

Question 15/30 *(2 p.)*

Which of the following code snippets safely retrieves the value from a **nested python dictionary without risk of a KeyError**?

A.
```
value = data.get('key1').get('key2').get('key3')
```

B.
```
value = data['key1']['key2']['key3']
```

C.
```
value = data.get('key1', {}).get('key2', {}).get('key3', None)
```

D.
```
try:
    value = data['key1']['key2']['key3']
except KeyError as exc:
    raise KeyError("Key could not be found")
```

E.
```
value = data->"key1"->"key2"->"key3"
```

Question 16/30 *(2 p.)*

Which of the following code snippets correctly creates a dictionary that maps user IDs (**from 1 to n**) to user objects?

A.
```
{user.id: user for user in users}
```

B.
```
{i: f"user_{i}" for i in range(1, n+1)}
```

C.
```
{i: User(i) for i in range(n)}
```

D.
```
{i: User(i) for i in range(1, n+1)}
```

Question 17/30 *(2 p.)*

```
nested_list = [[1, 2], [3, 4], [5, 6]]
```

Flatten the list and **double each element if it is even**, and otherwise leave the element as-is.

A.
```
[2 * x if x % 2 == 0 for x in sublist for sublist in nested_list]
```

B.
```
[2 * x if x % 2 == 0 else x for sublist in nested_list for x in sublist
```

C.
```
[2 * x for x in sublist if x % 2 == 0 for sublist in nested_list]
```

D.
```
[2 * x for sublist in nested_list for x in sublist if x % 2 == 0]
```

Question 18/30 *(1 p.)*

Consider the following Python code:

```
class MyClass:
    def __init__(self, value):
        self.value = value

class Wrapper:
    def create(self, value):
        return MyClass(value)

obj = Wrapper()
x = MyClass(5)
y = obj.create(5)
```

Which of the following statements are correct?

A. **y** is an instance of **Wrapper**, while **x** is an instance of **MyClass**

B. **y** is an instance of MyClass created inside Wrapper, making it a **subclass of Wrapper**

C. Both **x** and **y** are instances of **MyClass**

Question 19/30 *(1 p.)*

Consider the following python code:

```
def my_generator():
    for i in range(5):
        if i % 2 == 0:
            yield i

gen = my_generator()
next(gen)
next(gen)
third_call = next(gen)
```

What will be the value of the variable **third_call** after executing the above code?

A. 0

B. 1

C. 2

D. 4

E. None

F. **StopIteration** Exception

Question 20/30 *(1 p.)*

You have a series of **python statements** that might **raise** different **exceptions**. You want to **handle each exception** type with specific logic. What's the best way to structure the exception handling?

A. Use a **single except** block with multiple **if statements**

B. Use a single except block that takes in a **tuple of exceptions**

C. **Nested try**-except blocks

D. Use **multiple except** blocks

E. Use **multiple except** blocks within a **context manager**

Question 21/30 *(1 p.)*

Given the following Python code and **assuming Python v3.5+**

```
def multiply(x: int, y: float) -> float:
    return x * y
```

Select all options below that are true

A. During **runtime**, the function will always raise an **error** if **x** is **not an integer** or if **y** is **not a float.**
B. **x** can be specified as an **integer OR** a **float** by using **typing.Union** in the source code.
C. Changing the function **return type to int** will cause a **runtime error.**
D. The **int** and **float types** are **ignored** by the python runtime **during execution.**

Question 22/30

You have a **CPU-bound** task that you want to parallelize across multiple cores. You decide to use the **ThreadPoolExecutor.** What could be a potential issue with this approach?

A. The ThreadPoolExecutor **cannot** run **more** than **one thread** concurrently
B. The Global Interpreter Lock (**GIL**) may **limit** the **parallelism** of CPU-bound tasks.
C. The ThreadPoolExecutor **can only** be used for **I/O-bound** tasks
D. The ThreadPoolExecutor **cannot** run tasks that include **loops**

Question 23/30 *(1 p.)*

JSON is most closely related to which Python built-in data structure?

A. Dictionary
B. Tuple
C. List
D. String

Question 24/30 *(1 p.)*

You notice that some entries in a **numerical column are encoded as strings**. What potential issue might this lead to during analysis?

A. This will automatically create **outliers** in the data.
B. This will always lead to a **biased sample**.
C. This will lead to **incorrect relationships** between variables
D. This will **prevent** the correct application of **mathematical operations** on the column.

Question 25/30 *(1 p.)*

In a CSV file, a field (column) has a comma in it. How should we handle it?

A. Enclose the field (column) with **double quotes**
B. **No action needed**, the CSV format automatically accounts for this

Question 26/30 *(1 p.)*

Which of the following facts are true about **DataFrames** in general? **Check all that apply.**

A. All **data** must be of the **same data type**
B. All values in a **column** must be of the **same data type**
C. All values in a **row** (or record) must be of the **same data type**
D. There can only be **one column** with a **specific datatype** in a DataFrame
E. The primary column (**index**) of the DataFrame needs to have a **single type**, but all **other columns** can be **free-form**.
F. **Columns** in a DataFrame cannot have complex data types. They **must be primitive types** like int, float, str, etc.

Question 27/30 *(1 p.)*

You are working on **multiple Python projects**, and each project requires **different package versions**. What's the best practice to handle these dependencies?

A. **pip install** <package_name> for all packages, and find a package version that's the lowest version that supports all projects by trial and error.
B. Store each project in a **separate directory** on the filesystem. This will enforce isolation
C. Use **conda** or a virtual environment to manually create a separate env for each project
D. Open the projects separately in an **IDE** like VSCode or Pycharm to execute code.

Question 28/30 *(1 p.)*

Data lakes designed in the **modern** era have the following feature: Data is stored in **object store**; tools that can process and analyze the data in place, such as **Serverless SQL**, are used.

A. True
B. False

Question 29/30 *(1 p.)*

A system must handle large amounts of rapidly changing data, like session information or real-time analytics, where the value associated for sessions, or an ID change frequently. **Which database should you use?**

A. NoSQL **Document** Store (ex. **Mongo**)
B. NoSQL **Graph Database (**ex. **Neo4J)**
C. Cloud **Object Store (**ex. **S3)**
D. NoSQL **Key-Value** Store (ex. **Redis**)
E. **Relational tables** with spatial indexing (ex. **Postgres**)
F. **Distributed** File System (ex. **Hadoop**)

Question 30/30 *(1 p.)*

A dataset has a zero statistical variance. Which statement is true regarding its observations?

A. All observations have **distinct values**
B. All observations have the **same value**
C. Most observations are **close to the mean.**
D. Observations are **evenly distributed on both sides of the mean**