

3. GAC017 Computing III: Data Science and Web Apps Starter Quiz · 课前小测

GAC Computing

Name: _____ Score: _____

1. Where must a database password live?
 - ☐ In the JavaScript, minified
 - ☐ On the server only
 - ☐ In an HTML comment
 - ☐ In the CSS
2. Which query is safe from SQL injection?
 - ☐ "SELECT * FROM students WHERE name = '" + name + "'"
 - ☐ "SELECT * FROM students WHERE name = ?" with the value passed separately
 - ☐ Either, if the input is short
 - ☐ Either, if the site has few visitors
3. Students take many courses and courses have many students. How many tables?
 - ☐ One
 - ☐ Two
 - ☐ Three — the relationship needs its own table
 - ☐ One per student
4. You want only groups whose COUNT(*) exceeds 20. Which keyword filters them?
 - ☐ WHERE
 - ☐ HAVING
 - ☐ ORDER BY
 - ☐ LIMIT
5. A student exists but has no enrolments. What should the API return?
 - ☐ 404, because there is nothing
 - ☐ 200 with an empty list
 - ☐ 500, an error
 - ☐ Nothing at all
6. A request can reach your server without your page ever being loaded.
 - ☐ True
 - ☐ False

7. Which values must be treated as untrusted? Choose all that apply.

- ☐ Form fields
- ☐ URL parameters
- ☐ Uploaded files
- ☐ Values your own code computed

Tick every correct option.

8. Dropping rows with missing values is a neutral choice that needs no explanation.

- ☐ True
- ☐ False

9. What text format is normally used to send data between browser and server?

10. Returning the database's error message helps the visitor fix their input.

- ☐ True
- ☐ False

GAC Computing

1. **On the server only**

Front-end code is downloaded to every visitor's machine by definition. Minifying hides nothing.

2. **"SELECT * FROM students WHERE name = ?" with the value passed separately**

A parameterised query sends the value separately, so input can never become instruction.

3. **Three — the relationship needs its own table**

A many-to-many relationship cannot live in either table. Its rows are (student, course) pairs.

4. **HAVING**

WHERE filters rows before grouping; the count only exists after the grouping has happened.

5. **200 with an empty list**

The student exists, so the request succeeded. 404 is for a student who does not exist.

6. **True**

Which is exactly why the server must check every rule itself, whatever the browser already checked.

7. **Form fields; URL parameters; Uploaded files**

Everything that arrived from outside. The one attacker only needs you to have forgotten once.

8. **False**

It changes who your result describes. Write down the decision and who was removed.

9. **JSON**

Both sides can parse it, and it is readable when you are debugging.

10. **False**

It tells an attacker your table names and an ordinary visitor nothing useful. Log it; return a plain message.