📘 MCQs: Tracing – Traces vs Spans, Multi-run Traces

**1. What does tracing mean in Agentic AI?**

A) Measuring only model accuracy  
B) Recording execution flow of agents, tools, and reasoning steps  
C) Storing user prompts only  
D) Counting tokens

**Answer:** B  
**Explanation:** Tracing is about **logging the full execution flow** (inputs, outputs, tool calls, reasoning).

**2. What is a trace?**

A) A single token sequence  
B) The entire record of one agent run or session  
C) Only the tool call history  
D) Only errors

**Answer:** B  
**Explanation:** A trace captures the **complete journey** of an execution.

**3. What is a span in tracing?**

A) The overall execution log  
B) A sub-section of a trace representing a smaller task or event  
C) A retry loop  
D) A system message

**Answer:** B  
**Explanation:** A span is a **unit of work** inside a trace (e.g., tool call, model step).

**4. How do traces and spans relate?**

A) Traces contain spans  
B) Spans contain traces  
C) They are the same  
D) They never overlap

**Answer:** A  
**Explanation:** A **trace is made of multiple spans** organized hierarchically.

**5. Example of a span?**

A) The entire multi-turn conversation  
B) A single API call made during reasoning  
C) A retry fallback system  
D) A log of token counts

**Answer:** B  
**Explanation:** A span is a **smaller step** within the larger trace.

**6. Why is tracing useful?**

A) For debugging, monitoring, and analyzing agent behavior  
B) For token reduction  
C) For temperature adjustment  
D) For guardrail enforcement

**Answer:** A  
**Explanation:** Tracing makes agent **execution transparent** and easier to debug.

**7. What is a multi-run trace?**

A) A trace of a single span  
B) A combined trace that spans multiple related executions  
C) A retry loop trace  
D) A log of errors only

**Answer:** B  
**Explanation:** Multi-run traces link **several executions** for long workflows.

**8. When are multi-run traces most useful?**

A) For static single-step Q&A  
B) For complex workflows with multiple sequential runs  
C) For adjusting hyperparameters  
D) For token counting

**Answer:** B  
**Explanation:** Multi-run traces help in **tracking long-running agent processes**.

**9. In tracing, a parent-child relationship exists between:**

A) Tokens and prompts  
B) Traces and spans  
C) Errors and retries  
D) Context and system messages

**Answer:** B  
**Explanation:** **Spans nest inside traces**, forming a parent-child hierarchy.

**10. Which tool is often used to visualize traces and spans?**

A) Token counter  
B) OpenTelemetry or similar observability frameworks  
C) Retry logger  
D) Guardrails

**Answer:** B  
**Explanation:** Observability tools like **OpenTelemetry** visualize traces.

**11. If a span fails, what happens to the trace?**

A) Entire trace is deleted  
B) Trace still exists, but shows the error inside the span  
C) Trace completes successfully anyway  
D) Span is hidden

**Answer:** B  
**Explanation:** A trace includes **both successful and failed spans**.

**12. How does tracing help with error handling?**

A) It prevents all errors  
B) It provides detailed logs of where and why errors occurred  
C) It disables retries  
D) It resets execution

**Answer:** B  
**Explanation:** Tracing gives **step-level insight** into failures.

**13. Which is true about multi-run traces?**

A) They merge multiple independent traces randomly  
B) They provide continuity across sessions or runs of an agent  
C) They only apply to tool errors  
D) They reduce token usage

**Answer:** B  
**Explanation:** Multi-run traces give a **big-picture view** of multi-step tasks.

**14. What type of data can a span record?**

A) Start/end time, inputs, outputs, errors  
B) Only token counts  
C) Only retry logs  
D) Only system messages

**Answer:** A  
**Explanation:** Spans capture **execution metadata**.

**15. Which tracing feature is critical for performance monitoring?**

A) Retry loops  
B) Latency measurement inside spans  
C) Guardrails  
D) Temperature sampling

**Answer:** B  
**Explanation:** Spans track **latency per step**, aiding performance tuning.

**16. Which is a practical example of multi-run tracing?**

A) Tracking a chatbot’s single reply  
B) Tracking a cargo assistant’s workflow across multiple API calls and retries  
C) Token sampling analysis  
D) Guardrail enforcement logs

**Answer:** B  
**Explanation:** Multi-run tracing is essential for **multi-step agent pipelines**.

**17. Why are spans more granular than traces?**

A) They capture small steps like tool calls within larger workflows  
B) They only show failures  
C) They represent entire agent runs  
D) They ignore inputs

**Answer:** A  
**Explanation:** Spans are **fine-grained execution units**.

**18. In a multi-run trace, how are runs usually linked?**

A) By common session IDs or correlation IDs  
B) By token usage logs  
C) By guardrails  
D) By retry loops

**Answer:** A  
**Explanation:** **Session/correlation IDs** link multiple runs together.

**19. What is the best practice when using tracing in production?**

A) Disable tracing to save tokens  
B) Use structured, hierarchical spans with IDs and timestamps  
C) Log everything as plain text  
D) Only log errors

**Answer:** B  
**Explanation:** Structured spans make debugging and observability **easier and scalable**.

**20. Why are traces and spans important in Agentic AI?**

A) They ensure safety guardrails  
B) They make the reasoning process **observable, debuggable, and auditable**  
C) They reduce temperature  
D) They limit creativity

**Answer:** B  
**Explanation:** Traces + spans = **visibility** into how agents reason and act.