### Command Specifications for Memory Structure Management

#### General Notes:

- The `execute\_command` function processes various commands to manage the memory and interactions within the task management system.

- Each command has a unique role in subscribing, retrieving, querying, codifying, triggering actions, and managing content or triggers.

- The `anchor` parameter is used to identify where the result of the operation should be written or stored.

### Commands:

1. \*\*SUBSCRIBE\*\*

- \*\*Function\*\*: Registers interest in an event type with the event broker.

- \*\*Parameters\*\*:

- `event`: The event to subscribe to.

- `anchor`: The anchor point for the subscription.

- \*\*Example\*\*:

```python

instruction = {

'command': 'SUBSCRIBE',

'event': 'user\_login',

'anchor': 'subscription\_1'

}

```

2. \*\*RETRIEVE\*\*

- \*\*Function\*\*: Retrieves data from a specified URL, optionally compresses it, and stores it in the memory.

- \*\*Parameters\*\*:

- `url`: The URL to retrieve data from.

- `compress`: Boolean indicating whether to compress the retrieved data.

- `anchor`: The anchor point for storing the retrieved data.

- \*\*Example\*\*:

```python

instruction = {

'command': 'RETRIEVE',

'url': 'http://example.com/data.json',

'compress': True,

'anchor': 'data\_1'

}

```

3. \*\*RAG\_QUERY\*\*

- \*\*Function\*\*: Performs a retrieval-augmented generation query using the LLM.

- \*\*Parameters\*\*:

- `prompt`: The query prompt for the RAG operation.

- `anchor`: The anchor point for storing the result.

- \*\*Example\*\*:

```python

instruction = {

'command': 'RAG\_QUERY',

'prompt': 'What is the capital of France?',

'anchor': 'result\_1'

}

```

4. \*\*LLM\_QUERY\*\*

- \*\*Function\*\*: Queries the LLM with a given prompt, providing context from a specified scope.

- \*\*Parameters\*\*:

- `prompt`: The query prompt for the LLM.

- `scope`: The scope of memory to provide as context.

- `model` (optional): The specific LLM model to use.

- `anchor`: The anchor point for storing the result.

- \*\*Example\*\*:

```python

instruction = {

'command': 'LLM\_QUERY',

'prompt': 'Summarize the document.',

'scope': 'document\_1',

'model': 'gpt-3.5-turbo',

'anchor': 'summary\_1'

}

```

5. \*\*CODIFY\*\*

- \*\*Function\*\*: Converts a given prompt into executable code and stores it.

- \*\*Parameters\*\*:

- `prompt`: The query prompt to be converted into code.

- `anchor`: The anchor point for storing the generated code.

- \*\*Example\*\*:

```python

instruction = {

'command': 'CODIFY',

'prompt': 'Write a function to add two numbers.',

'anchor': 'code\_1'

}

```

6. \*\*TRIGGER\_ACTIONS\*\*

- \*\*Function\*\*: Sets up triggers for specific events or content.

- \*\*Sub-Commands\*\*:

- `LLM\_CLASSIFY`

- \*\*Parameters\*\*:

- `scope`: The scope of memory to monitor for classification.

- `anchor`: The anchor point for storing the result.

- \*\*Example\*\*:

```python

instruction = {

'command': 'LLM\_CLASSIFY',

'scope': 'user\_input\_1',

'anchor': 'classification\_1'

}

```

- `UNSUBSCRIBE`

- \*\*Parameters\*\*:

- `event`: The event to unsubscribe from.

- `anchor`: The anchor point for the unsubscription.

- \*\*Example\*\*:

```python

instruction = {

'command': 'UNSUBSCRIBE',

'event': 'user\_login',

'anchor': 'subscription\_1'

}

```

- `REMOVE\_CONTENT\_TRIGGER`

- \*\*Parameters\*\*:

- `scope`: The scope of memory to remove the content trigger from.

- \*\*Example\*\*:

```python

instruction = {

'command': 'REMOVE\_CONTENT\_TRIGGER',

'scope': 'user\_input\_1'

}

```

- `COMPRESS`

- \*\*Parameters\*\*:

- `prompt`: The query prompt for compression.

- `anchor`: The anchor point for storing the compressed data.

- `datalimit`: The limit for the amount of data to compress.

- \*\*Example\*\*:

```python

instruction = {

'command': 'COMPRESS',

'prompt': 'Compress this text.',

'anchor': 'compressed\_data\_1',

'datalimit': 1024

}

```

### Additional Commands:

7. \*\*REMOVE\_CODE\_TRIGGER\*\*

- \*\*Function\*\*: Removes a code trigger for a specific event.

- \*\*Parameters\*\*:

- `event`: The event to remove the code trigger from.

- \*\*Example\*\*:

```python

instruction = {

'command': 'REMOVE\_CODE\_TRIGGER',

'event': 'user\_login'

}

```

8. \*\*REMOVE\_CODE\_CONTENT\_TRIGGER\*\*

- \*\*Function\*\*: Removes a code content trigger for specific items in the scope.

- \*\*Parameters\*\*:

- `scope`: The scope of memory to remove the code content trigger from.

- \*\*Example\*\*:

```python

instruction = {

'command': 'REMOVE\_CODE\_CONTENT\_TRIGGER',

'scope': 'user\_input\_1'

}

```

9. \*\*REMOVE\_CLASSIFY\_TRIGGER\*\*

- \*\*Function\*\*: Removes a classification trigger for specific items in the scope.

- \*\*Parameters\*\*:

- `scope`: The scope of memory to remove the classification trigger from.

- \*\*Example\*\*:

```python

instruction = {

'command': 'REMOVE\_CLASSIFY\_TRIGGER',

'scope': 'user\_input\_1'

}

```