

~\Desktop\workflow.json

```

1  {
2    "name": "My workflow",
3    "nodes": [
4      {
5        "parameters": {
6          "options": {}
7        },
8        "id": "9722d5a3-5403-4368-ac78-6656d2a30a3b",
9        "name": "OpenAI Chat Model",
10       "type": "@n8n/n8n-nodes-langchain.lmChatOpenAi",
11       "typeVersion": 1,
12       "position": [
13         -740,
14         260
15     ]
16   },
17   {
18     "parameters": {
19       "jsonMode": "expressionData",
20       "jsonData": "={{ $json.data || $json.text || $json.concatenated_data }}",
21       "options": {
22         "metadata": {
23           "metadataValues": [
24             {
25               "name": "=file_id",
26               "value": "={{ $('Set File ID').first().json.file_id }}"
27             },
28             {
29               "name": "file_title",
30               "value": "={{ $('Set File ID').first().json.file_title }}"
31             }
32           ]
33         }
34       }
35     },
36     "id": "b928576a-d809-49b0-8031-ed57027fadb",
37     "name": "Default Data Loader",
38     "type": "@n8n/n8n-nodes-langchain.documentDefaultDataLoader",
39     "typeVersion": 1,
40     "position": [
41       1000,
42       1040
43   ]
44 },
45 {
46   "parameters": {
47     "model": "text-embedding-3-small",
48     "options": {}
49   },
50   "id": "21572c59-0136-407b-8353-9413347cc1a7",
51   "name": "Embeddings OpenAI1",

```

```
52     "type": "@n8n/n8n-nodes-langchain.embeddingsOpenAi",
53     "typeVersion": 1,
54     "position": [
55         760,
56         1040
57     ]
58 },
59 {
60     "parameters": {
61         "content": "## Agent Tools for RAG",
62         "height": 528.85546469693,
63         "width": 583.4552380860637,
64         "color": 4
65     },
66     "id": "06e2066d-37da-43d6-ae7a-5ea9dae56fc0",
67     "name": "Sticky Note",
68     "type": "n8n-nodes-base.stickyNote",
69     "typeVersion": 1,
70     "position": [
71         0,
72         -100
73     ]
74 },
75 {
76     "parameters": {
77         "content": "## Tool to Add a Google Drive File to Vector DB",
78         "height": 867,
79         "width": 3073,
80         "color": 5
81     },
82     "id": "b6c9df15-28c6-4870-aa21-2e689b787417",
83     "name": "Sticky Note1",
84     "type": "n8n-nodes-base.stickyNote",
85     "typeVersion": 1,
86     "position": [
87         -1740,
88         440
89     ]
90 },
91 {
92     "parameters": {
93         "operation": "download",
94         "fileId": {
95             "__rl": true,
96             "value": "={{ $('Set File ID').item.json.file_id }}",
97             "mode": "id"
98         },
99         "options": {
100             "googleFileConversion": {
101                 "conversion": {
102                     "docsToFormat": "text/plain"
103                 }
104             }
105         }
106     }
```

```

106     },
107     "id": "e0f086c6-11f5-459c-b994-5e4332fd9072",
108     "name": "Download File",
109     "type": "n8n-nodes-base.googleDrive",
110     "typeVersion": 3,
111     "position": [
112         -660,
113         720
114     ],
115     "executeOnce": true
116 },
117 {
118     "parameters": {
119         "pollTimes": {
120             "item": [
121                 {
122                     "mode": "everyMinute"
123                 }
124             ]
125         },
126         "triggerOn": "specificFolder",
127         "folderToWatch": {
128             "__rl": true,
129             "value": "1HtaIIK3kWwjbwhsmEbtJ-upalxVn3Ek9",
130             "mode": "list",
131             "cachedResultName": "n8n Documents",
132             "cachedResultUrl":
133                 "https://drive.google.com/drive/folders/1HtaIIK3kWwjbwhsmEbtJ-upalxVn3Ek9"
134         },
135         "event": "fileCreated",
136         "options": {}
137     },
138     "id": "f3eb94bf-9bfd-4a9f-a51d-9e5a8016b4cd",
139     "name": "File Created",
140     "type": "n8n-nodes-base.googleDriveTrigger",
141     "typeVersion": 1,
142     "position": [
143         -1680,
144         560
145     ],
146     {
147         "parameters": {
148             "pollTimes": {
149                 "item": [
150                     {
151                         "mode": "everyMinute"
152                     }
153                 ]
154             },
155             "triggerOn": "specificFolder",
156             "folderToWatch": {
157                 "__rl": true,
158                 "value": "1HtaIIK3kWwjbwhsmEbtJ-upalxVn3Ek9",

```

```

159         "mode": "list",
160         "cachedResultName": "n8n Documents",
161         "cachedResultUrl":
"https://drive.google.com/drive/folders/1HtaIIK3kWWjbwhsmEbtJ-upalxVn3Ek9"
162     },
163     "event": "fileUpdated",
164     "options": {}
165 },
166 "id": "dd257dfd-c069-47e1-a47c-14b723ca4fa2",
167 "name": "File Updated",
168 "type": "n8n-nodes-base.googleDriveTrigger",
169 "typeVersion": 1,
170 "position": [
171     -1680,
172     720
173 ],
174 },
175 {
176     "parameters": {
177         "operation": "text",
178         "options": {}
179     },
180     "id": "608d35f0-404c-4aaa-8fb5-b4a2d1023ad1",
181     "name": "Extract Document Text",
182     "type": "n8n-nodes-base.extractFromFile",
183     "typeVersion": 1,
184     "position": [
185         200,
186         1040
187     ],
188     "alwaysOutputData": true
189 },
190 {
191     "parameters": {},
192     "id": "f39f0524-849c-4143-935e-71a6fe2de445",
193     "name": "Postgres Chat Memory",
194     "type": "@n8n/n8n-nodes-langchain.memoryPostgresChat",
195     "typeVersion": 1,
196     "position": [
197         -600,
198         260
199     ],
200     "notesInFlow": false
201 },
202 {
203     "parameters": {
204         "operation": "delete",
205         "tableId": "documents",
206         "filterType": "string",
207         "filterString": "=metadata->>file_id=like.*{{ $json.file_id }}"
208     },
209     "id": "454f0f1c-23ec-476e-a435-9839aba1d3e7",
210     "name": "Delete Old Doc Rows",
211     "type": "n8n-nodes-base.supabase",

```

```
212     "typeVersion": 1,
213     "position": [
214         -1140,
215         560
216     ],
217     "alwaysOutputData": true
218 },
219 {
220     "parameters": {
221         "assignments": {
222             "assignments": [
223                 {
224                     "id": "10646eae-ae46-4327-a4dc-9987c2d76173",
225                     "name": "file_id",
226                     "value": "={{ $json.id }}",
227                     "type": "string"
228                 },
229                 {
230                     "id": "f4536df5-d0b1-4392-bf17-b8137fb31a44",
231                     "name": "file_type",
232                     "value": "={{ $json.mimeType }}",
233                     "type": "string"
234                 },
235                 {
236                     "id": "77d782de-169d-4a46-8a8e-a3831c04d90f",
237                     "name": "file_title",
238                     "value": "={{ $json.name }}",
239                     "type": "string"
240                 },
241                 {
242                     "id": "9bde4d7f-e4f3-4ebd-9338-dce1350f9eab",
243                     "name": "file_url",
244                     "value": "={{ $json.webViewLink }}",
245                     "type": "string"
246                 }
247             ]
248         },
249         "options": {}
250     },
251     "id": "e6aecdd1-67a3-4cd3-9e53-f1cbafb66c",
252     "name": "Set File ID",
253     "type": "n8n-nodes-base.set",
254     "typeVersion": 3.4,
255     "position": [
256         -1320,
257         720
258     ]
259 },
260 {
261     "parameters": {
262         "content": "## RAG AI Agent with Chat Interface",
263         "height": 464.8027193303974,
264         "width": 1035.6381264595484
265     },
```

```
266     "id": "d40da6c5-d379-462f-a39c-a92eec338e8c",
267     "name": "Sticky Note2",
268     "type": "n8n-nodes-base.stickyNote",
269     "typeVersion": 1,
270     "position": [
271         -1040,
272         -40
273     ]
274 },
275 {
276     "parameters": {
277         "options": {}
278     },
279     "id": "d0cadd9b-2f86-4517-947c-2d4333ca2431",
280     "name": "Respond to Webhook",
281     "type": "n8n-nodes-base.respondToWebhook",
282     "typeVersion": 1.1,
283     "position": [
284         -180,
285         40
286     ]
287 },
288 {
289     "parameters": {
290         "assignments": {
291             "assignments": [
292                 {
293                     "id": "9a9a245e-f1a1-4282-bb02-a81ffe629f0f",
294                     "name": "chatInput",
295                     "value": "={{ $json?.chatInput || $json.body.chatInput }}",
296                     "type": "string"
297                 },
298                 {
299                     "id": "b80831d8-c653-4203-8706-adedfdb98f77",
300                     "name": "sessionId",
301                     "value": "={{ $json?.sessionId || $json.body.sessionId }}",
302                     "type": "string"
303                 }
304             ]
305         },
306         "options": {}
307     },
308     "id": "d7cb4634-5909-4e11-a861-116bd6bf846b",
309     "name": "Edit Fields",
310     "type": "n8n-nodes-base.set",
311     "typeVersion": 3.4,
312     "position": [
313         -740,
314         40
315     ]
316 },
317 {
318     "parameters": {
319         "public": true,
```

```
320     "options": {}
321   },
322   "id": "db0dce94-7899-4d78-b277-8c509a9e4dad",
323   "name": "When chat message received",
324   "type": "@n8n/n8n-nodes-langchain.chatTrigger",
325   "typeVersion": 1.1,
326   "position": [
327     -1000,
328     40
329   ],
330   "webhookId": "e104e40e-6134-4825-a6f0-8a646d882662"
331 },
332 {
333   "parameters": {
334     "httpMethod": "POST",
335     "path": "bf4dd093-bb02-472c-9454-7ab9af97bd1d",
336     "authentication": "headerAuth",
337     "responseMode": "responseNode",
338     "options": {}
339   },
340   "id": "23e4c31e-f05b-4a35-a90d-379ca5c3365f",
341   "name": "Webhook",
342   "type": "n8n-nodes-base.webhook",
343   "typeVersion": 2,
344   "position": [
345     -1000,
346     240
347   ],
348   "webhookId": "bf4dd093-bb02-472c-9454-7ab9af97bd1d"
349 },
350 {
351   "parameters": {
352     "operation": "pdf",
353     "options": {}
354   },
355   "id": "36d51402-3ead-4044-8c53-a24e0340a9dc",
356   "name": "Extract PDF Text",
357   "type": "n8n-nodes-base.extractFromFile",
358   "typeVersion": 1,
359   "position": [
360     200,
361     480
362   ]
363 },
364 {
365   "parameters": {
366     "aggregate": "aggregateAllItemData",
367     "options": {}
368   },
369   "id": "2f573d33-12aa-4f4f-acea-2ad370a201a8",
370   "name": "Aggregate",
371   "type": "n8n-nodes-base.aggregate",
372   "typeVersion": 1,
373   "position": [
```

```

374         240,
375         660
376     ]
377 },
378 {
379     "parameters": {},
380     "id": "59c8fbc6-012b-4237-98e6-5330398e37fb",
381     "name": "Character Text Splitter",
382     "type": "@n8n/n8n-nodes-langchain.textSplitterCharacterTextSplitter",
383     "typeVersion": 1,
384     "position": [
385         900,
386         1160
387     ]
388 },
389 {
390     "parameters": {
391         "fieldsToSummarize": {
392             "values": [
393                 {
394                     "aggregation": "concatenate",
395                     "field": "data"
396                 }
397             ]
398         },
399         "options": {}
400     },
401     "id": "e433e9e4-0318-4cef-95a7-19706f0523ae",
402     "name": "Summarize",
403     "type": "n8n-nodes-base.summarize",
404     "typeVersion": 1,
405     "position": [
406         440,
407         740
408     ]
409 },
410 {
411     "parameters": {
412         "promptType": "define",
413         "text": "={{ $json.chatInput }}",
414         "options": {
415             "systemMessage": "You are a personal assistant who helps answer questions from
a corpus of documents. The documents are either text based (Txt, docs, extracted PDFs,
etc.) or tabular data (CSVs or Excel documents).\n\nYou are given tools to perform RAG in
the 'documents' table, look up the documents available in your knowledge base in the
'document_metadata' table, extract all the text from a given document, and query the
tabular files with SQL in the 'document_rows' table.\n\nAlways start by performing RAG
unless the question requires a SQL query for tabular data (fetching a sum, finding a max,
something a RAG lookup would be unreliable for). If RAG doesn't help, then look at the
documents that are available to you, find a few that you think would contain the answer,
and then analyze those.\n\nAlways tell the user if you didn't find the answer. Don't make
something up just to please them."
416         }
417     },
418     "id": "558a8ac3-f0ac-45d4-9170-cde7cd61b3d3",

```

```

419     "name": "RAG AI Agent",
420     "type": "@n8n/n8n-nodes-langchain.agent",
421     "typeVersion": 1.6,
422     "position": [
423         -520,
424         40
425     ]
426 },
427 {
428     "parameters": {
429         "rules": {
430             "values": [
431                 {
432                     "conditions": {
433                         "options": {
434                             "caseSensitive": true,
435                             "leftValue": "",
436                             "typeValidation": "strict",
437                             "version": 1
438                         },
439                         "conditions": [
440                             {
441                                 "leftValue": "={{ $('Set File ID').item.json.file_type }}",
442                                 "rightValue": "application/pdf",
443                                 "operator": {
444                                     "type": "string",
445                                     "operation": "equals"
446                                 }
447                             }
448                         ],
449                         "combinator": "and"
450                     }
451                 },
452                 {
453                     "conditions": {
454                         "options": {
455                             "caseSensitive": true,
456                             "leftValue": "",
457                             "typeValidation": "strict",
458                             "version": 1
459                         },
460                         "conditions": [
461                             {
462                                 "id": "2ae7faa7-a936-4621-a680-60c512163034",
463                                 "leftValue": "={{ $('Set File ID').item.json.file_type }}",
464                                 "rightValue": "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet",
465                                 "operator": {
466                                     "type": "string",
467                                     "operation": "equals",
468                                     "name": "filter.operator.equals"
469                                 }
470                             }
471                         ],

```

```
472         "combinator": "and"
473     }
474 },
475 {
476     "conditions": {
477         "options": {
478             "caseSensitive": true,
479             "leftValue": "",
480             "typeValidation": "strict",
481             "version": 1
482         },
483         "conditions": [
484             {
485                 "id": "fc193b06-363b-4699-a97d-e5a850138b0e",
486                 "leftValue": "={{ $('Set File ID').item.json.file_type }}",
487                 "rightValue": "application/vnd.google-apps.spreadsheet",
488                 "operator": {
489                     "type": "string",
490                     "operation": "equals",
491                     "name": "filter.operator.equals"
492                 }
493             }
494         ],
495         "combinator": "and"
496     }
497 },
498 {
499     "conditions": {
500         "options": {
501             "caseSensitive": true,
502             "leftValue": "",
503             "typeValidation": "strict",
504             "version": 1
505         },
506         "conditions": [
507             {
508                 "id": "b69f5605-0179-4b02-9a32-e34bb085f82d",
509                 "leftValue": "={{ $('Set File ID').item.json.file_type }}",
510                 "rightValue": "application/vnd.google-apps.document",
511                 "operator": {
512                     "type": "string",
513                     "operation": "equals",
514                     "name": "filter.operator.equals"
515                 }
516             }
517         ],
518         "combinator": "and"
519     }
520 }
521 ]
522 },
523 "options": {
524     "fallbackOutput": 3
525 }
```

```

526     },
527     "id": "60d528e1-c78b-4a6a-ace9-58967e5c599d",
528     "name": "Switch",
529     "type": "n8n-nodes-base.switch",
530     "typeVersion": 3,
531     "position": [
532         -460,
533         700
534     ]
535 },
536 {
537     "parameters": {
538         "mode": "insert",
539         "tableName": {
540             "__rl": true,
541             "value": "documents",
542             "mode": "list",
543             "cachedResultName": "documents"
544         },
545         "options": {
546             "queryName": "match_documents"
547         }
548     },
549     "id": "4f3e1794-09cc-4443-a70f-d6c69a6264d3",
550     "name": "Insert into Supabase Vectorstore",
551     "type": "@n8n/n8n-nodes-langchain.vectorStoreSupabase",
552     "typeVersion": 1,
553     "position": [
554         920,
555         820
556     ]
557 },
558 {
559     "parameters": {
560         "operation": "xlsx",
561         "options": {}
562     },
563     "id": "dbfbe212-bbd1-449a-a146-35e3a80fb073",
564     "name": "Extract from Excel",
565     "type": "n8n-nodes-base.extractFromFile",
566     "typeVersion": 1,
567     "position": [
568         20,
569         660
570     ]
571 },
572 {
573     "parameters": {
574         "assignments": {
575             "assignments": [
576                 {
577                     "id": "f422e2e0-381c-46ea-8f38-3f58c501d8b9",
578                     "name": "schema",

```

```

579         "value": "={{ $('Extract from Excel').isExecuted ? $('Extract from
Excel').first().json.keys().toJsonString() : $('Extract from
CSV').first().json.keys().toJsonString() }}",
580         "type": "string"
581     },
582     {
583         "id": "bb07c71e-5b60-4795-864c-cc3845b6bc46",
584         "name": "data",
585         "value": "={{ $json.concatenated_data }}",
586         "type": "string"
587     }
588 ]
589 },
590 "options": {}
591 },
592 "type": "n8n-nodes-base.set",
593 "typeVersion": 3.4,
594 "position": [
595     880,
596     600
597 ],
598 "id": "fa26aaf6-39d1-49d9-84e2-4da528220e99",
599 "name": "Set Schema"
600 },
601 {
602     "parameters": {
603         "options": {}
604     },
605     "type": "n8n-nodes-base.extractFromFile",
606     "typeVersion": 1,
607     "position": [
608         20,
609         840
610     ],
611     "id": "c09b6dc4-67ef-4a6a-bd08-c09f10e525b0",
612     "name": "Extract from CSV"
613 },
614 {
615     "parameters": {
616         "content": "## Run Each Node Once to Set Up Database Tables",
617         "height": 300,
618         "width": 680,
619         "color": 3
620     },
621     "type": "n8n-nodes-base.stickyNote",
622     "position": [
623         -1740,
624         120
625     ],
626     "typeVersion": 1,
627     "id": "24f04905-46ab-48ad-ae72-9679c6b3cdfc",
628     "name": "Sticky Note3"
629 },
630 {

```

```

631     "parameters": {
632         "operation": "executeQuery",
633         "query": "CREATE TABLE document_metadata (\n    id TEXT PRIMARY KEY,\n    title
TEXT,\n    url TEXT,\n    created_at TIMESTAMP DEFAULT NOW(),\n    schema TEXT\n);",
634         "options": {}
635     },
636     "type": "n8n-nodes-base.postgres",
637     "typeVersion": 2.5,
638     "position": [
639         -1480,
640         220
641     ],
642     "id": "b93782eb-3593-4b36-b3c2-f59ddfb71e7a",
643     "name": "Create Document Metadata Table"
644 },
645 {
646     "parameters": {
647         "operation": "executeQuery",
648         "query": "CREATE TABLE document_rows (\n    id SERIAL PRIMARY KEY,\n
dataset_id TEXT REFERENCES document_metadata(id),\n    row_data JSONB -- Store the
actual row data\n);",
649         "options": {}
650     },
651     "type": "n8n-nodes-base.postgres",
652     "typeVersion": 2.5,
653     "position": [
654         -1260,
655         220
656     ],
657     "id": "1f0d2621-4f07-4321-a540-4065742368a8",
658     "name": "Create Document Rows Table (for Tabular Data)"
659 },
660 {
661     "parameters": {
662         "descriptionType": "manual",
663         "toolDescription": "Use this tool to fetch all available documents, including the
table schema if the file is a CSV or Excel file.",
664         "operation": "select",
665         "schema": {
666             "__rl": true,
667             "mode": "list",
668             "value": "public"
669         },
670         "table": {
671             "__rl": true,
672             "value": "document_metadata",
673             "mode": "list",
674             "cachedResultName": "document_metadata"
675         },
676         "returnAll": true,
677         "options": {}
678     },
679     "type": "n8n-nodes-base.postgresTool",
680     "typeVersion": 2.5,

```

```

681     "position": [
682         -460,
683         260
684     ],
685     "id": "747080c8-ad6e-43eb-941d-d1fda239394f",
686     "name": "List Documents"
687 },
688 {
689     "parameters": {
690         "descriptionType": "manual",
691         "toolDescription": "Given a file ID, fetches the text from the document.",
692         "operation": "executeQuery",
693         "query": "SELECT \n      string_agg(content, ' ') as document_text\nFROM\n documents\n WHERE metadata->>'file_id' = $1\nGROUP BY metadata->>'file_id';",
694         "options": {
695             "queryReplacement": "={{ $fromAI('file_id') }}"
696         }
697     },
698     "type": "n8n-nodes-base.postgresTool",
699     "typeVersion": 2.5,
700     "position": [
701         -320,
702         260
703     ],
704     "id": "f6673a78-47dd-44d9-85b5-c5b18c185946",
705     "name": "Get File Contents"
706 },
707 {
708     "parameters": {
709         "descriptionType": "manual",
710         "toolDescription": "Run a SQL query - use this to query from the document_rows\n table once you know the file ID you are querying. dataset_id is the file_id and you are\n always using the row_data for filtering, which is a jsonb field that has all the keys\n from the file schema given in the document_metadata table.\n\nExample query:\n\nSELECT\n AVG((row_data->>'revenue')::numeric)\nFROM document_rows\nWHERE dataset_id =\n '123';\n\nExample query 2:\n\nSELECT \n      row_data->>'category' as category,\n SUM((row_data->>'sales')::numeric) as total_sales\nFROM dataset_rows\nWHERE dataset_id =\n '123'\nGROUP BY row_data->>'category';",
711         "operation": "executeQuery",
712         "query": "{{ $fromAI('sql_query') }}",
713         "options": {}
714     },
715     "type": "n8n-nodes-base.postgresTool",
716     "typeVersion": 2.5,
717     "position": [
718         -160,
719         260
720     ],
721     "id": "5abefc0c-356f-429b-8afe-6c662b7d6753",
722     "name": "Query Document Rows"
723 },
724 {
725     "parameters": {
726         "mode": "retrieve-as-tool",
727         "toolName": "documents",

```

```

728     "toolDescription": "Use RAG to look up information in the knowledgebase.",
729     "tableName": {
730         "__rl": true,
731         "value": "documents",
732         "mode": "list",
733         "cachedResultName": "documents"
734     },
735     "options": {
736         "queryName": "match_documents"
737     }
738 },
739 "type": "@n8n/n8n-nodes-langchain.vectorStoreSupabase",
740 "typeVersion": 1,
741 "position": [
742     160,
743     40
744 ],
745 "id": "866fd439-c755-47b2-b8fe-a4e21fa10bee",
746 "name": "Supabase Vector Store1"
747 },
748 {
749     "parameters": {
750         "options": {}
751     },
752     "type": "@n8n/n8n-nodes-langchain.embeddingsOpenAi",
753     "typeVersion": 1.2,
754     "position": [
755         260,
756         220
757     ],
758     "id": "63c31f78-f359-40ba-b3a5-5599beb02aac",
759     "name": "Embeddings OpenAI2"
760 },
761 {
762     "parameters": {
763         "options": {
764             "reset": false
765         }
766     },
767     "type": "n8n-nodes-base.splitInBatches",
768     "typeVersion": 3,
769     "position": [
770         -1500,
771         560
772     ],
773     "id": "42b7c387-190e-48ab-9043-31a21b8b6d91",
774     "name": "Loop Over Items"
775 },
776 {
777     "parameters": {
778         "operation": "executeQuery",
779         "query": "-- Enable the pgvector extension to work with embedding vectors\ncreate
extension vector;\n\n-- Create a table to store your documents\ncreate table documents
(\n  id bigserial primary key,\n  content text, -- corresponds to Document.pageContent\n
```

```

metadata jsonb, -- corresponds to Document.metadata\n embedding vector(1536) -- 1536
works for OpenAI embeddings, change if needed\n);\n\n-- Create a function to search for
documents\ncreate function match_documents (\n query_embedding vector(1536),\n
match_count int default null,\n filter jsonb DEFAULT '{}'\n) returns table (\n id
bigint,\n content text,\n metadata jsonb,\n similarity float\n)\nlanguage plpgsql\nas
$$\n#variable_conflict use_column\nbegin\n return query\n select\n id,\n
content,\n metadata,\n 1 - (documents.embedding <=> query_embedding) as
similarity\n from documents\n where metadata @> filter\n order by documents.embedding
<=> query_embedding\n limit match_count;\nend;\n$$";

```

```

780     "options": {}
781 },
782 "type": "n8n-nodes-base.postgres",
783 "typeVersion": 2.5,
784 "position": [
785     -1680,
786     220
787 ],
788 "id": "d7c8285c-3880-4a6f-84ba-64d79b082809",
789 "name": "Create Documents Table and Match Function"
790 },
791 {
792     "parameters": {
793         "operation": "delete",
794         "tableId": "document_rows",
795         "filters": {
796             "conditions": [
797                 {
798                     "keyName": "dataset_id",
799                     "condition": "eq",
800                     "keyValue": "={{ $('Set File ID').item.json.file_id }}"
801                 }
802             ]
803         }
804     },
805     "type": "n8n-nodes-base.supabase",
806     "typeVersion": 1,
807     "position": [
808         -980,
809         720
810     ],
811     "id": "eeb17ade-4255-4bfc-93d5-9947669f7d0b",
812     "name": "Delete Old Data Rows",
813     "alwaysOutputData": true,
814     "executeOnce": true
815 },
816 {
817     "parameters": {
818         "operation": "upsert",
819         "schema": {
820             "__rl": true,
821             "mode": "list",
822             "value": "public"
823         },
824         "table": {
825             "__rl": true,

```

```
826     "value": "document_metadata",
827     "mode": "list",
828     "cachedResultName": "document_metadata"
829 },
830 "columns": {
831     "mappingMode": "defineBelow",
832     "value": {
833         "id": "={ { $('Set File ID').item.json.file_id } }",
834         "title": "={ { $('Set File ID').item.json.file_title } }",
835         "url": "={ { $('Set File ID').item.json.file_url } }"
836     },
837     "matchingColumns": [
838         "id"
839     ],
840     "schema": [
841         {
842             "id": "id",
843             "displayName": "id",
844             "required": true,
845             "defaultMatch": true,
846             "display": true,
847             "type": "string",
848             "canBeUsedToMatch": true,
849             "removed": false
850         },
851         {
852             "id": "title",
853             "displayName": "title",
854             "required": false,
855             "defaultMatch": false,
856             "display": true,
857             "type": "string",
858             "canBeUsedToMatch": false
859         },
860         {
861             "id": "url",
862             "displayName": "url",
863             "required": false,
864             "defaultMatch": false,
865             "display": true,
866             "type": "string",
867             "canBeUsedToMatch": false,
868             "removed": false
869         },
870         {
871             "id": "created_at",
872             "displayName": "created_at",
873             "required": false,
874             "defaultMatch": false,
875             "display": true,
876             "type": "dateTime",
877             "canBeUsedToMatch": false
878         },
879         {
```

```

880         "id": "schema",
881         "displayName": "schema",
882         "required": false,
883         "defaultMatch": false,
884         "display": true,
885         "type": "string",
886         "canBeUsedToMatch": false,
887         "removed": true
888     }
889 ],
890     "attemptToConvertTypes": false,
891     "convertFieldsToString": false
892 },
893     "options": {}
894 },
895     "type": "n8n-nodes-base.postgres",
896     "typeVersion": 2.5,
897     "position": [
898         -820,
899         580
900     ],
901     "id": "38f39b17-45cd-4bc3-9cb5-c68f5bdf4c2",
902     "name": "Insert Document Metadata",
903     "executeOnce": true
904 },
905 {
906     "parameters": {
907         "schema": {
908             "__rl": true,
909             "mode": "list",
910             "value": "public"
911         },
912         "table": {
913             "__rl": true,
914             "value": "document_rows",
915             "mode": "list",
916             "cachedResultName": "document_rows"
917         },
918         "columns": {
919             "mappingMode": "defineBelow",
920             "value": {
921                 "dataset_id": "={{ $('Set File ID').item.json.file_id }}",
922                 "row_data": "={{ $json.toJsonString().replaceAll(/'/g, '\\''\\') }}"
923             },
924             "matchingColumns": [
925                 "id"
926             ],
927             "schema": [
928                 {
929                     "id": "id",
930                     "displayName": "id",
931                     "required": false,
932                     "defaultMatch": true,
933                     "display": true,

```

```

934         "type": "number",
935         "canBeUsedToMatch": true,
936         "removed": true
937     },
938     {
939         "id": "dataset_id",
940         "displayName": "dataset_id",
941         "required": false,
942         "defaultMatch": false,
943         "display": true,
944         "type": "string",
945         "canBeUsedToMatch": true
946     },
947     {
948         "id": "row_data",
949         "displayName": "row_data",
950         "required": false,
951         "defaultMatch": false,
952         "display": true,
953         "type": "object",
954         "canBeUsedToMatch": true
955     }
956 ],
957 "attemptToConvertTypes": false,
958 "convertFieldsToString": false
959 },
960 "options": {}
961 },
962 "type": "n8n-nodes-base.postgres",
963 "typeVersion": 2.5,
964 "position": [
965     240,
966     840
967 ],
968 "id": "3d666271-f707-4e0b-8abc-57fbcd86f3f6",
969 "name": "Insert Table Rows"
970 },
971 {
972     "parameters": {
973         "operation": "upsert",
974         "schema": {
975             "__rl": true,
976             "mode": "list",
977             "value": "public"
978         },
979         "table": {
980             "__rl": true,
981             "value": "document_metadata",
982             "mode": "list",
983             "cachedResultName": "document_metadata"
984         },
985         "columns": {
986             "mappingMode": "defineBelow",
987             "value": {

```

```
988     "id": "={{ $( 'Set File ID' ).item.json.file_id }}",
989     "schema": "={{ $json.schema }}"
990 },
991 "matchingColumns": [
992     "id"
993 ],
994 "schema": [
995     {
996         "id": "id",
997         "displayName": "id",
998         "required": true,
999         "defaultMatch": true,
1000         "display": true,
1001         "type": "string",
1002         "canBeUsedToMatch": true,
1003         "removed": false
1004     },
1005     {
1006         "id": "title",
1007         "displayName": "title",
1008         "required": false,
1009         "defaultMatch": false,
1010         "display": true,
1011         "type": "string",
1012         "canBeUsedToMatch": false,
1013         "removed": true
1014     },
1015     {
1016         "id": "url",
1017         "displayName": "url",
1018         "required": false,
1019         "defaultMatch": false,
1020         "display": true,
1021         "type": "string",
1022         "canBeUsedToMatch": false,
1023         "removed": true
1024     },
1025     {
1026         "id": "created_at",
1027         "displayName": "created_at",
1028         "required": false,
1029         "defaultMatch": false,
1030         "display": true,
1031         "type": "dateTime",
1032         "canBeUsedToMatch": false
1033     },
1034     {
1035         "id": "schema",
1036         "displayName": "schema",
1037         "required": false,
1038         "defaultMatch": false,
1039         "display": true,
1040         "type": "string",
1041         "canBeUsedToMatch": false,
```

```

1042         "removed": false
1043     }
1044 ],
1045     "attemptToConvertTypes": false,
1046     "convertFieldsToString": false
1047 },
1048     "options": {}
1049 },
1050     "type": "n8n-nodes-base.postgres",
1051     "typeVersion": 2.5,
1052     "position": [
1053         1100,
1054         600
1055     ],
1056     "id": "d9239289-80b1-47aa-956c-e20826671797",
1057     "name": "Update Schema for Document Metadata"
1058 },
1059 {
1060     "parameters": {
1061         "content": "## 🚀 Ultimate n8n Agentic RAG Template\n\n## What is this?\nThis template provides a complete implementation of an **Agentic RAG (Retrieval Augmented Generation)** system in n8n that can be extended easily for your specific use case and knowledge base. Unlike standard RAG which only performs simple lookups, this agent can reason about your knowledge base, self-improve retrieval, and dynamically switch between different tools based on the specific question.\n\n## Why Agentic RAG?\nStandard RAG has significant limitations:\n- Poor analysis of numerical/tabular data\n- Missing context due to document chunking\n- Inability to connect information across documents\n- No dynamic tool selection based on question type\n\n## What makes this template powerful:\n- **Intelligent tool selection**: Switches between RAG lookups, SQL queries, or full document retrieval based on the question\n- **Complete document context**: Accesses entire documents when needed instead of just chunks\n- **Accurate numerical analysis**: Uses SQL for precise calculations on spreadsheet/tabular data\n- **Cross-document insights**: Connects information across your entire knowledge base\n- **Multi-file processing**: Handles multiple documents in a single workflow loop\n- **Efficient storage**: Uses JSONB in Supabase to store tabular data without creating new tables for each CSV\n\n## Getting Started\n1. Run the table creation nodes first to set up your database tables in Supabase\n2. Upload your documents through Google Drive (or swap out for a different file storage solution)\n3. The agent will process them automatically (chunking text, storing tabular data in Supabase)\n4. Start asking questions that leverage the agent's multiple reasoning approaches\n\n## Customization\nThis template provides a solid foundation that you can extend by:\n- Tuning the system prompt for your specific use case\n- Adding document metadata like summaries\n- Implementing more advanced RAG techniques\n- Optimizing for larger knowledge bases\n\n---\n\nI do intend on making a local version of this agent very soon!",
1062         "height": 1320,
1063         "width": 540,
1064         "color": 6
1065     },
1066     "type": "n8n-nodes-base.stickyNote",
1067     "position": [
1068         -2320,
1069         -20
1070     ],
1071     "typeVersion": 1,
1072     "id": "28a8a713-e8b6-4aae-a8af-4bd05bffffd4c",
1073     "name": "Sticky Note9"

```

```
1074     }
1075   ],
1076   "pinData": {
1077     "File Created": [
1078       {
1079         "json": {
1080           "kind": "drive#file",
1081           "copyRequiresWriterPermission": false,
1082           "writersCanShare": true,
1083           "viewedByMe": true,
1084           "mimeType": "application/vnd.google-apps.spreadsheet",
1085           "exportLinks": {
1086             "application/x-vnd.oasis.opendocument.spreadsheet":
1087             "https://docs.google.com/spreadsheets/export?id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1088             AWlM&exportFormat=ods",
1089             "text/tab-separated-values": "https://docs.google.com/spreadsheets/export?
1090             id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=tsv",
1091             "application/pdf": "https://docs.google.com/spreadsheets/export?
1092             id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=pdf",
1093             "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet":
1094             "https://docs.google.com/spreadsheets/export?id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1095             AWlM&exportFormat=xlsx",
1096             "text/csv": "https://docs.google.com/spreadsheets/export?
1097             id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=csv",
1098             "application/zip": "https://docs.google.com/spreadsheets/export?
1099             id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=zip",
1100             "application/vnd.oasis.opendocument.spreadsheet":
1101             "https://docs.google.com/spreadsheets/export?id=1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1102             AWlM&exportFormat=ods"
1103           },
1104           "parents": [
1105             "1HtaIIK3kwwjbwhsmEbtJ-upalxVn3Ek9"
1106           ],
1107           "thumbnailLink": "https://lh3.googleusercontent.com/drive-
1108           storage/AJQWtB025toxt6RwVPjSmlzZEda-lk4VxRy3z0bFM0tcvKSX8m0uyYJAOMHYjIxSVnl6-
1109           GgWT9QJFzn6oAaUF0PdnWPB89gUcAo5c97K0Nb1vYNFjk5jE0pro51Aa5PYpA=s220",
1110           "iconLink": "https://drive-
1111           thirdparty.googleusercontent.com/16/type/application/vnd.google-apps.spreadsheet",
1112           "shared": false,
1113           "lastModifyingUser": {
1114             "displayName": "Cole Medin",
1115             "kind": "drive#user",
1116             "me": true,
1117             "permissionId": "09805982089636049437",
1118             "emailAddress": "colemedin@ottomotor.ai",
1119             "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjXYQW9sb7WW3wnC-
1120             1AVobRmOW6jJVl-c6A8a1Vmfg22B_1lKKPYS1Ko9UEZvIV23wtXqnJ0dgC56hiJN_z6e2d9S2MdiFDTpg7cyD-
1121             gXf1LICu2REL4QH54-MvYmKecDcm4MgnZYSms-X1KpD6seI1pkEehGcRg0NIXn2wGCPicXUZ30K8SF-
1122             o9PxCr_0NRyR93hQn8hkX3M1nBuHIRL4Fvu5MjOvxRJQkqaF9zf0JVyB1LHdV-Ypr87v_B_7hDaJ8Jor2z73LA-
1123             rKq_mcU6Ta46Hsm83AnTY3iOJE9WxH2Cd9N795co4COaYL5hHEADdPq_B-OfVcbYNk1kf2qT4Josi_-
1124             cu1uRGUFnhpBpD5Cu5uy2qD-J8GrMgkKdRkr1S7Z9ZD6bYyb3r3j6s4NHpE_iswHBjH-WfD2fg590TiCs6ciAPbx-
1125             5769BYDF6Z2hVsCVq0G5bgS3JfbCTVnK8mNCh_4gzyRyfzj380R0L5Fngv1i8y-HluGNw2k37cyrYu4HunK-
1126             jMZKy_NsmOTWLTiTeYU_VoDvH-egUftryYVVCdUn1SEpSjDBYjpLmQqCk-Kg3jK-0DGmClklbFw_svgvWUqy-
1127             oOWMHPh5fNTLdmPYK6w0Zcft5GXar9bahLAK9IT-epMIFTwnxhQu2yT0cP1I186OoWWUxCapElN_AEJD-
1128             hibEMLYrHlMjUuhq205dClKGEvXw9D2ejYVpY0xVQ1ZxpKV-QGDTkniAjlWVCRwdaDrghH3ieKgy5LXj357oCXUw-
1129             RC-5niqp5MiybN0bN7KUE1zPSJjMDSjIkKcUMd_uAx_hlFkhx_jD0i0IMs3BLQ3OEOLw1IoD2Lo16jBstCU-
```

```
cZFCQrikP7d-nabEX8yDTLfyI9e8sDPp2yNA4pBJfKUm9-AHOAFWZ1BzXJN97RIiTQyKdExCKSz3kPf2lrQyTD-
VwYkZ75-ZkksHvXoz3kWo3MrAjlG156MxCrOkd88teMhDHq0E=s64"
1107     },
1108     "owners": [
1109         {
1110             "displayName": "Cole Medin",
1111             "kind": "drive#user",
1112             "me": true,
1113             "permissionId": "09805982089636049437",
1114             "emailAddress": "colemedin@ottomotor.ai",
1115             "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjXYQW9sb7WW3wnC-
1AVobRmOW6jJVl-c6A8a1Vmfg22B_1lKKPYS1Ko9UEZvIV23wtXqnJ0dgC56hiJN_z6e2d9S2MdiFDTpg7cyD-
gXf1LICu2REL4QH54-MvYmKecDcm4MgnZYSms-X1KpD6seI1pkEehGcRg0NIXn2wGCPicXUZ30K8SF-
o9PxCr_0NRyR93hQn8hkX3M1nBuHIRL4Fvu5MjOvxRJQkqaF9zf0JVyB1LHdV-Ypr87v_B_7hDaJ8Jor2z73LA-
rKq_mcU6Ta46Hsm83ANty3i0JE9WxH2Cd9N795co4C0aYL5hHEAdDpQ_B-OfVcbYNk1kf2qT4Josi_-
cu1uRGUFnbpD5Cu5uy2qD-J8GrMgkKdRKR1S7Z9ZD6bYyb3r3j6s4NHpE_iswHBjh-WfD2fg590TiCs6ciAPbx-
5769BYDF6Z2hVsCVq0G5bgS3JfbCTVNk8mNCh_4gzyRyfzj380R0L5Fngv1i8y-HluGNw2k37cyrYu4HunK-
jMZKy_NsmOTWLITeyU_VoDvH-egUftryYVVCdUn1SEpSjDBYjpLmQqCk-Kg3jK-0DGmClkblFw_svgvWUqy-
oOWMHHPH5fNTLdmPYK6w0Zcft5GXar9bahLAK9IT-epMIFTwnxhQu2yT0cP1I186OoWWUxCapE1N_AEJD-
hibEMLyrHlmJUuhq205dClKEvXw9D2ejYvY0xVQ1ZxpKV-QGDTkniAjlWVCRwdaDrgbH3ieKgy5LXj357oCXUw-
RC-5niqp5MiybN0bN7KUe1zPSJjMDSjIkKcUMD_uAx_hlFkhx_jD0i0IMs3BLQ30EOLw1IoD2Lol6jBstCU-
cZFCQrikP7d-nabEX8yDTLfyI9e8sDPp2yNA4pBJfKUm9-AHOAFWZ1BzXJN97RIiTQyKdExCKSz3kPf2lrQyTD-
VwYkZ75-ZkksHvXoz3kWo3MrAjlG156MxCrOkd88teMhDHq0E=s64"
1116         }
1117     ],
1118     "webViewLink": "https://docs.google.com/spreadsheets/d/1py8boCnHGKiZ4EUJ3jA-
P8rNDVxYArwoOJsg0xXeAWlM/edit?usp=drivesdk",
1119     "size": "1024",
1120     "viewersCanCopyContent": true,
1121     "permissions": [
1122         {
1123             "id": "09805982089636049437",
1124             "displayName": "Cole Medin",
1125             "type": "user",
1126             "kind": "drive#permission",
1127             "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjU-
zfOspR2RPV0ODNiUMTUKvYDBBCZV-dZusxjov3wKmsVu93G9IfBKLBtQef-ZfzxVCqoc8HAVOEJyVPWiFax-
dpQqjp5Bmqnsiestzam2U55u4Y7m3qsZGuIwTYYCQ4-TI6w2ActqiVwzXSrFt8hF5q5UYn1zzHvR8OG1fQH-
hoxUgOpsSwsylfLh2gRj49gNFuJ0tQZFIT37R0-dsKLjV0DqBDUpqHgP4mnA02dUgy3D9BjshwFHQY-
IdWnK1xGTTrunpzeHBrIE306eUhHcXaN2D0P8vAdpFRjTKyJMCDVp5ZVMn79MSZWUVbN9TGD_u4DpbXFV-
B27hi4MjS8mHblF5tJLSKtLs-UW4itcrt-QnUMeHkIEpcly8vsGcui0myNH1ldsJ79Hh0sSPHY-
9vxxvwSpXBOn_9F15_8080m2de0duT2RwGeM1EXXjE0P0456Y4BStBV1uRa3VceZ213MgXkbF8BZG_f0-
DjNrVkw7xqdOHBarD087vRhEawPlawPTFw8UX7KGRsv9jIyRuLmvcSINIw2nEuZOXYKFeHR6IfIRIuTk-
0F7E0qk8WftQlvVx-a--ELays2G5R1oVqrFst2mb06U9wQz9Uhz3SDsIh60B7S0e86x0hkdfS11bKmxSqPx-d-
kcYuW88etuwSYnoZEQVAs1Vyu_XHYxfMw_nFZ0PTH_LMc9DYINUILu1RvuMPq_bfkusoIr4EEyZTGtQVY-
4qf01eEgRCG1xGBIN6_6dzyWGZoHdayfSXyDoFfwE2vcWBz30qFeImwis1u67VnbWCyFC0vb8rEScWo1-
9Xpyxw81umlclTlOXA8buq7nnD8IgVETwn7islB-mzsjnQ0trjDCOJ4T4KdzYYpM0iWdwSbhu95sDfa2-
GBenSM6_shlwm88V56XEyDZ7t0K52rtgjX7tI9qQg_5uAsQSmUQWwHXs1qLw_0Lm8HJm7NTJey6_un2pCzhM-
eajxl3Gw=s64",
1128             "emailAddress": "colemedin@ottomotor.ai",
1129             "role": "owner",
1130             "deleted": false,
1131             "pendingOwner": false
1132         }
1133     ],
1134     "hasThumbnail": true,
1135     "spaces": [
```

```
1136     "drive"
1137 ],
1138 "id": "1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwoOJsg0xXeAWlM",
1139 "name": "Test Sheet #1",
1140 "starred": false,
1141 "trashed": false,
1142 "explicitlyTrashed": false,
1143 "createdTime": "2025-02-23T17:00:50.000Z",
1144 "modifiedTime": "2025-02-23T17:43:45.286Z",
1145 "modifiedByMeTime": "2025-02-23T17:43:45.286Z",
1146 "viewedByMeTime": "2025-02-23T18:21:12.774Z",
1147 "quotaBytesUsed": "1024",
1148 "version": "17",
1149 "ownedByMe": true,
1150 "isAppAuthorized": false,
1151 "capabilities": {
1152     "canChangeViewersCanCopyContent": true,
1153     "canEdit": true,
1154     "canCopy": true,
1155     "canComment": true,
1156     "canAddChildren": false,
1157     "canDelete": true,
1158     "canDownload": true,
1159     "canListChildren": false,
1160     "canRemoveChildren": false,
1161     "canRename": true,
1162     "canTrash": true,
1163     "canReadRevisions": true,
1164     "canChangeCopyRequiresWriterPermission": true,
1165     "canMoveItemIntoTeamDrive": true,
1166     "canUntrash": true,
1167     "canModifyContent": true,
1168     "canMoveItemOutOfDrive": true,
1169     "canAddMyDriveParent": false,
1170     "canRemoveMyDriveParent": true,
1171     "canMoveItemWithinDrive": true,
1172     "canShare": true,
1173     "canMoveChildrenWithinDrive": false,
1174     "canModifyContentRestriction": true,
1175     "canChangeSecurityUpdateEnabled": false,
1176     "canAcceptOwnership": false,
1177     "canReadLabels": false,
1178     "canModifyLabels": false,
1179     "canModifyEditorContentRestriction": true,
1180     "canModifyOwnerContentRestriction": true,
1181     "canRemoveContentRestriction": false,
1182     "canDisableInheritedPermissions": false,
1183     "canEnableInheritedPermissions": true
1184 },
1185 "thumbnailVersion": "3",
1186 "modifiedByMe": true,
1187 "permissionIds": [
1188     "09805982089636049437"
1189 ],
```

```
1190     "linkShareMetadata": {
1191       "securityUpdateEligible": false,
1192       "securityUpdateEnabled": true
1193     },
1194     "inheritedPermissionsDisabled": false
1195   }
1196 },
1197 {
1198   "json": {
1199     "kind": "drive#file",
1200     "copyRequiresWriterPermission": false,
1201     "writersCanShare": true,
1202     "viewedByMe": true,
1203     "mimeType": "application/vnd.google-apps.spreadsheet",
1204     "exportLinks": {
1205       "application/x-vnd.oasis.opendocument.spreadsheet":
1206       "https://docs.google.com/spreadsheets/export?id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1207       AWlM&exportFormat=ods",
1208       "text/tab-separated-values": "https://docs.google.com/spreadsheets/export?
1209       id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=tsv",
1210       "application/pdf": "https://docs.google.com/spreadsheets/export?
1211       id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=pdf",
1212       "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet":
1213       "https://docs.google.com/spreadsheets/export?id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1214       AWlM&exportFormat=xlsx",
1215       "text/csv": "https://docs.google.com/spreadsheets/export?
1216       id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=csv",
1217       "application/zip": "https://docs.google.com/spreadsheets/export?
1218       id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXeAWlM&exportFormat=zip",
1219       "application/vnd.oasis.opendocument.spreadsheet":
1220       "https://docs.google.com/spreadsheets/export?id=1py8boCnHGKiZ4EUJ3jAP8rNDVxYArwo0Jsg0xXe-
1221       AWlM&exportFormat=ods"
1222     },
1223     "parents": [
1224       "1HtaIIK3kWwjbwhsmEbtJ-upalxVn3Ek9"
1225     ],
1226     "thumbnailLink": "https://lh3.googleusercontent.com/drive-
1227     storage/AJQWtB025toxt6RwVPjSmlzZEda-lk4VxRy3z0bFM0tcvKSX8m0uyYJAOMHYjIxSVnl6-
1228     GgWT9QJFzn6oAaUFOPdnWPB89gUcAo5c97K0Nb1vYNFjk5jE0pro51Aa5PYpA=s220",
1229     "iconLink": "https://drive-
1230     thirdparty.googleusercontent.com/16/type/application/vnd.google-apps.spreadsheet",
1231     "shared": false,
1232     "lastModifyingUser": {
1233       "displayName": "Cole Medin",
1234       "kind": "drive#user",
1235       "me": true,
1236       "permissionId": "09805982089636049437",
1237       "emailAddress": "colemedin@ottomotor.ai",
1238       "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjXYQW9sb7WW3wnC-
1239       1AVobRmOW6jJV1-c6A8a1Vmfg22B_1lKKPYS1Ko9UEZvIV23wtXqnJ0dgC56hiJN_z6e2d9S2MdiFDTPg7cyD-
1240       gXf1LICu2REL4QH54-MvYmKecDcm4MgnZYSms-X1KpD6seI1pkEehGcRg0NIXn2wGCPicXUZ30K8SF-
1241       o9PxCr_0NRYR93hQn8hkX3M1nBuHIRL4Fvu5MjOvxRJQkqaF9zf0JVyB1LHdV-Ypr87v_B_7hDaJ8Jor2z73LA-
1242       rKq_mcU6Ta46Hsm83AntY3i0JE9WxH2Cd9N795co4COaYL5hHEAdDpQ_B-OfVcbYnk1kf2qT4Josi_-
1243       cu1uRGUFnhpBpD5Cu5uy2qD-J8GrMgkKdRKR1S7Z9ZD6bYyb3r3j6s4NHpE_iswHBjH-WfD2fg590TiCs6ciAPbx-
1244       5769BYDF6Z2hVsCVq0G5bgS3JfbCTVNk8mNch_4gzyRyFzj380R0L5Fngv1i8y-HluGNw2k37cyrYu4HunK-
1245       jMZKy_NsmOTWLITeyU_VoDVH-egUftryYVVCdUn1SEpSjDByjpLmQqCk-Kg3jK-0DGmClkb1Fw_svgvWUqy-
```

```
oOWMHPh5fNTLdmPYK6w0Zcft5GXar9bahLAK9IT-epMIFTwnxhQu2yT0cP1I186OoWWUxCapElN_AEJD-
hibEMLYrHlMjUuhq205dClKGEvXw9D2ejYVpY0xVQ1ZxpKV-QGDTkniAjlWVCRwdaDrgbH3ieKgy5LXj357oCXUw-
RC-5niqp5MiybN0bN7KUE1zPSJjMDSjIkKcUMd_uAx_hlFkhx_jD0i0IMs3BLQ30EOLw1IoD2Lo16jBstCU-
cZFCQrikP7d-nabEX8yDTLfyi9e8sDPp2yNA4pBJfKUm9-AHOAFWZ1BzXJN97RIiTQyKdExCKSz3kPf2lrQyTD-
VwYkZ75-ZkksHvXoz3kWo3MrAjG156MxCrOkd88teMhDHq0E=s64"
1226     },
1227     "owners": [
1228         {
1229             "displayName": "Cole Medin",
1230             "kind": "drive#user",
1231             "me": true,
1232             "permissionId": "09805982089636049437",
1233             "emailAddress": "colemedin@ottomotor.ai",
1234             "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjXYQW9sb7WW3wnC-
1AVobRmOW6jJV1-c6A8a1Vmfg22B_1lKKPYSlKo9UEZvIV23wtXqnJ0dgC56hiJN_z6e2d9S2MdiFDTpg7cyD-
gXf1LICu2REL4QH54-MvYmKecDcm4MgnZYSms-X1KpD6seI1pkEehGcRg0NIXn2wGCPicXUZ30K8SF-
o9PxCr_0NRyR93hQn8hkX3M11nBuHIRL4Fvu5MjOvxRJQkqaf9zf0JVyB1LHdV-Ypr87v_B_7hDaJ8Jor2z73LA-
rKq_mcU6Ta46Hsm83AntY3iOJE9WxH2Cd9N795co4CoaYL5hHEADdPq_B-OfVcbYnK1kf2qT4Josi_-
cu1uRGUFnhpBpD5Cu5uy2qD-J8GrMgkKdRkr1S7Z9ZD6bYyb3r3j6s4NHpE_iswHBjH-WfD2fg590TiCs6ciAPbx-
5769BYDF6Z2hVsCVq0G5bgS3JfbCTVnK8mNCh_4gzyRyFzj380R0L5Fngv1i8y-HluGNw2k37cyrYu4HunK-
jMZKy_NsmOTWLITeyU_VoDvH-egUftryYVVCdUn1SEpSjDBYjpLmQqCk-Kg3jK-0DGmClkblFw_svgvWUqy-
oOWMHPh5fNTLdmPYK6w0Zcft5GXar9bahLAK9IT-epMIFTwnxhQu2yT0cP1I186OoWWUxCapElN_AEJD-
hibEMLYrHlMjUuhq205dClKGEvXw9D2ejYVpY0xVQ1ZxpKV-QGDTkniAjlWVCRwdaDrgbH3ieKgy5LXj357oCXUw-
RC-5niqp5MiybN0bN7KUE1zPSJjMDSjIkKcUMd_uAx_hlFkhx_jD0i0IMs3BLQ30EOLw1IoD2Lo16jBstCU-
cZFCQrikP7d-nabEX8yDTLfyi9e8sDPp2yNA4pBJfKUm9-AHOAFWZ1BzXJN97RIiTQyKdExCKSz3kPf2lrQyTD-
VwYkZ75-ZkksHvXoz3kWo3MrAjG156MxCrOkd88teMhDHq0E=s64"
1235         }
1236     ],
1237     "webViewLink": "https://docs.google.com/spreadsheets/d/1py8boCNhGKiZ4EUJ3jA-
P8rNDVxYArwoOJsg0xXeAWlM/edit?usp=drivesdk",
1238     "size": "1024",
1239     "viewersCanCopyContent": true,
1240     "permissions": [
1241         {
1242             "id": "09805982089636049437",
1243             "displayName": "Cole Medin",
1244             "type": "user",
1245             "kind": "drive#permission",
1246             "photoLink": "https://lh3.googleusercontent.com/a-/ALV-UjU-
zf0spr2RPV0ODNiUMTUKvYDBBCZV-dZusxjov3wKmsVu93G9IfBKLbtQef-ZfzxVCqoc8HAVOEJyVPWiFax-
dpQqjp5Bmqnsiestzam2U55u4Y7m3qsZgGuIwTYCYCQ4-TI6w2ActqiVwzXSrFt8hF5q5UYn1zzHvR80G1fQH-
hoxUgOpsSwsylfLh2grJ49gNFuJ0tQZFIT37R0-dsKLjV0DqBDUpqHgP4mnA02dUgy3D9BjshwFHQY-
IdWnK1xGTrunpzeHBrIE306eUhhcXaN2D0P8vAdpFRjTKyJMCdVp5ZVMn79MSZwUVbN9TGD_u4DpbXFV-
B27hi4MjS8mHblF5tJLSKtLs-UW4itcrt-QnUMeHkIEpCly8vsGcuiOmyNH1ldsJ79HhOsSPhY-
9vxxvSpXBOn_9F15_8080m2de0duT2RewGeM1EXXjE0P0456Y4BStBV1uRa3VceZ213MgXkbF8BZG_f0-
DjNrVkw7xqdOHBArD087vRhEawPlawPTFw8UX7kGRsV9jIyRuLmvcSINIw2nEuZOXYKFeHR6IfIRIuTk-
0F7E0qk8WftQ1vVx-a--ELays2G5R1oVqrFst2mb06U9wQz9UhZ3SDsIh60B7S0e86x0hkdfS11bKmxSqPxx-d-
kcYuW88etuwSYnoZEQVAs1Vyu_XHYxfMw_nFZ0PTH_LMC9DYINUlLu1RvuMPq_bfkusoIr4EEyZTGtQVY-
4qf01eEgRCG1xGBIN6_6dzyWGZoHdayfSXyDoFfwE2vcWBz30qFeImwis1u67VnbWCyfc0vb8rEScwo1-
9Xpyxw81umlCTLOXA8buq7nnD8IgVETwn7is1B-mzsJnQ0trjDCOJ4T4KdzYYpM0iWdwSbhu95sDfa2-
GBeNSM6_shlwwm88V56XEyDZ7t0K52rtgjX7tI9qQg_5uAsQSmUQWwHXs1qLw_0Lm8HJm7NTJey6_uN2pCzhM-
eajxl3Gw=s64",
1247             "emailAddress": "colemedin@ottomotor.ai",
1248             "role": "owner",
1249             "deleted": false,
1250             "pendingOwner": false
1251         }
    ]
}
```

```
1252 ],
1253 "hasThumbnail": true,
1254 "spaces": [
1255   "drive"
1256 ],
1257 "id": "1py8boCNhGKiZ4EUJ3jAP8rNDVxYArwoOJsg0xXeAWlM",
1258 "name": "Test Sheet #1",
1259 "starred": false,
1260 "trashed": false,
1261 "explicitlyTrashed": false,
1262 "createdTime": "2025-02-23T17:00:50.000Z",
1263 "modifiedTime": "2025-02-23T17:43:45.286Z",
1264 "modifiedByMeTime": "2025-02-23T17:43:45.286Z",
1265 "viewedByMeTime": "2025-02-23T18:21:12.774Z",
1266 "quotaBytesUsed": "1024",
1267 "version": "17",
1268 "ownedByMe": true,
1269 "isAppAuthorized": false,
1270 "capabilities": {
1271   "canChangeViewersCanCopyContent": true,
1272   "canEdit": true,
1273   "canCopy": true,
1274   "canComment": true,
1275   "canAddChildren": false,
1276   "canDelete": true,
1277   "canDownload": true,
1278   "canListChildren": false,
1279   "canRemoveChildren": false,
1280   "canRename": true,
1281   "canTrash": true,
1282   "canReadRevisions": true,
1283   "canChangeCopyRequiresWriterPermission": true,
1284   "canMoveItemIntoTeamDrive": true,
1285   "canUntrash": true,
1286   "canModifyContent": true,
1287   "canMoveItemOutOfDrive": true,
1288   "canAddMyDriveParent": false,
1289   "canRemoveMyDriveParent": true,
1290   "canMoveItemWithinDrive": true,
1291   "canShare": true,
1292   "canMoveChildrenWithinDrive": false,
1293   "canModifyContentRestriction": true,
1294   "canChangeSecurityUpdateEnabled": false,
1295   "canAcceptOwnership": false,
1296   "canReadLabels": false,
1297   "canModifyLabels": false,
1298   "canModifyEditorContentRestriction": true,
1299   "canModifyOwnerContentRestriction": true,
1300   "canRemoveContentRestriction": false,
1301   "canDisableInheritedPermissions": false,
1302   "canEnableInheritedPermissions": true
1303 },
1304 "thumbnailVersion": "3",
1305 "modifiedByMe": true,
```

```
1306     "permissionIds": [  
1307         "09805982089636049437"  
1308     ],  
1309     "linkShareMetadata": {  
1310         "securityUpdateEligible": false,  
1311         "securityUpdateEnabled": true  
1312     },  
1313     "inheritedPermissionsDisabled": false  
1314 }  
1315 }  
1316 ]  
1317 },  
1318 "connections": {  
1319     "OpenAI Chat Model": {  
1320         "ai_languageModel": [  
1321             [  
1322                 {  
1323                     "node": "RAG AI Agent",  
1324                     "type": "ai_languageModel",  
1325                     "index": 0  
1326                 }  
1327             ]  
1328         }  
1329     },  
1330     "Download File": {  
1331         "main": [  
1332             [  
1333                 {  
1334                     "node": "Switch",  
1335                     "type": "main",  
1336                     "index": 0  
1337                 }  
1338             ]  
1339         ]  
1340     },  
1341     "File Created": {  
1342         "main": [  
1343             [  
1344                 {  
1345                     "node": "Loop Over Items",  
1346                     "type": "main",  
1347                     "index": 0  
1348                 }  
1349             ]  
1350         ]  
1351     },  
1352     "Extract Document Text": {  
1353         "main": [  
1354             [  
1355                 {  
1356                     "node": "Insert into Supabase Vectorstore",  
1357                     "type": "main",  
1358                     "index": 0  
1359                 }  
1360             ]  
1361         ]  
1362     }  
1363 }
```

```
1360     ]
1361   ]
1362 },
1363 "Embeddings OpenAI1": {
1364   "ai_embedding": [
1365     [
1366       {
1367         "node": "Insert into Supabase Vectorstore",
1368         "type": "ai_embedding",
1369         "index": 0
1370       }
1371     ]
1372   ]
1373 },
1374 "Default Data Loader": {
1375   "ai_document": [
1376     [
1377       {
1378         "node": "Insert into Supabase Vectorstore",
1379         "type": "ai_document",
1380         "index": 0
1381       }
1382     ]
1383   ]
1384 },
1385 "Postgres Chat Memory": {
1386   "ai_memory": [
1387     [
1388       {
1389         "node": "RAG AI Agent",
1390         "type": "ai_memory",
1391         "index": 0
1392       }
1393     ]
1394   ]
1395 },
1396 "Delete Old Doc Rows": {
1397   "main": [
1398     [
1399       {
1400         "node": "Delete Old Data Rows",
1401         "type": "main",
1402         "index": 0
1403       }
1404     ]
1405   ]
1406 },
1407 "Set File ID": {
1408   "main": [
1409     [
1410       {
1411         "node": "Delete Old Doc Rows",
1412         "type": "main",
1413         "index": 0
```

```
1414     }
1415   ]
1416 ]
1417 },
1418 "File Updated": {
1419   "main": [
1420     [
1421       {
1422         "node": "Loop Over Items",
1423         "type": "main",
1424         "index": 0
1425       }
1426     ]
1427   ]
1428 },
1429 "Edit Fields": {
1430   "main": [
1431     [
1432       {
1433         "node": "RAG AI Agent",
1434         "type": "main",
1435         "index": 0
1436       }
1437     ]
1438   ]
1439 },
1440 "When chat message received": {
1441   "main": [
1442     [
1443       {
1444         "node": "Edit Fields",
1445         "type": "main",
1446         "index": 0
1447       }
1448     ]
1449   ]
1450 },
1451 "Webhook": {
1452   "main": [
1453     [
1454       {
1455         "node": "Edit Fields",
1456         "type": "main",
1457         "index": 0
1458       }
1459     ]
1460   ]
1461 },
1462 "Extract PDF Text": {
1463   "main": [
1464     [
1465       {
1466         "node": "Insert into Supabase Vectorstore",
1467         "type": "main",
```

```
1468         "index": 0
1469     }
1470 ]
1471 ]
1472 },
1473 "Aggregate": {
1474     "main": [
1475         [
1476             {
1477                 "node": "Summarize",
1478                 "type": "main",
1479                 "index": 0
1480             }
1481         ]
1482     ]
1483 },
1484 "Character Text Splitter": {
1485     "ai_textSplitter": [
1486         [
1487             {
1488                 "node": "Default Data Loader",
1489                 "type": "ai_textSplitter",
1490                 "index": 0
1491             }
1492         ]
1493     ]
1494 },
1495 "Summarize": {
1496     "main": [
1497         [
1498             {
1499                 "node": "Set Schema",
1500                 "type": "main",
1501                 "index": 0
1502             },
1503             {
1504                 "node": "Insert into Supabase Vectorstore",
1505                 "type": "main",
1506                 "index": 0
1507             }
1508         ]
1509     ]
1510 },
1511 "RAG AI Agent": {
1512     "main": [
1513         [
1514             {
1515                 "node": "Respond to Webhook",
1516                 "type": "main",
1517                 "index": 0
1518             }
1519         ]
1520     ]
1521 },
```

```
1522     "Switch": {
1523       "main": [
1524         [
1525           {
1526             "node": "Extract PDF Text",
1527             "type": "main",
1528             "index": 0
1529           }
1530         ],
1531         [
1532           {
1533             "node": "Extract from Excel",
1534             "type": "main",
1535             "index": 0
1536           }
1537         ],
1538         [
1539           {
1540             "node": "Extract from CSV",
1541             "type": "main",
1542             "index": 0
1543           }
1544         ],
1545         [
1546           {
1547             "node": "Extract Document Text",
1548             "type": "main",
1549             "index": 0
1550           }
1551         ]
1552       ]
1553     },
1554     "Extract from Excel": {
1555       "main": [
1556         [
1557           {
1558             "node": "Aggregate",
1559             "type": "main",
1560             "index": 0
1561           },
1562           {
1563             "node": "Insert Table Rows",
1564             "type": "main",
1565             "index": 0
1566           }
1567         ]
1568       ]
1569     },
1570     "Set Schema": {
1571       "main": [
1572         [
1573           {
1574             "node": "Update Schema for Document Metadata",
1575             "type": "main",
```

```
1576         "index": 0
1577     }
1578 ]
1579 ]
1580 },
1581 "Extract from CSV": {
1582     "main": [
1583         [
1584             {
1585                 "node": "Aggregate",
1586                 "type": "main",
1587                 "index": 0
1588             },
1589             {
1590                 "node": "Insert Table Rows",
1591                 "type": "main",
1592                 "index": 0
1593             }
1594         ]
1595     ]
1596 },
1597 "Create Document Metadata Table": {
1598     "main": [
1599         [
1600             {
1601                 "node": "Create Document Rows Table (for Tabular Data)",
1602                 "type": "main",
1603                 "index": 0
1604             }
1605         ]
1606     ]
1607 },
1608 "List Documents": {
1609     "ai_tool": [
1610         [
1611             {
1612                 "node": "RAG AI Agent",
1613                 "type": "ai_tool",
1614                 "index": 0
1615             }
1616         ]
1617     ]
1618 },
1619 "Get File Contents": {
1620     "ai_tool": [
1621         [
1622             {
1623                 "node": "RAG AI Agent",
1624                 "type": "ai_tool",
1625                 "index": 0
1626             }
1627         ]
1628     ]
1629 },
```

```
1630     "Query Document Rows": {
1631       "ai_tool": [
1632         [
1633           {
1634             "node": "RAG AI Agent",
1635             "type": "ai_tool",
1636             "index": 0
1637           }
1638         ]
1639       ],
1640     },
1641     "Supabase Vector Store1": {
1642       "ai_tool": [
1643         [
1644           {
1645             "node": "RAG AI Agent",
1646             "type": "ai_tool",
1647             "index": 0
1648           }
1649         ]
1650       ],
1651     },
1652     "Embeddings OpenAI2": {
1653       "ai_embedding": [
1654         [
1655           {
1656             "node": "Supabase Vector Store1",
1657             "type": "ai_embedding",
1658             "index": 0
1659           }
1660         ]
1661       ],
1662     },
1663     "Loop Over Items": {
1664       "main": [
1665         [],
1666         [
1667           {
1668             "node": "Set File ID",
1669             "type": "main",
1670             "index": 0
1671           }
1672         ]
1673       ],
1674     },
1675     "Insert into Supabase Vectorstore": {
1676       "main": [
1677         [
1678           {
1679             "node": "Loop Over Items",
1680             "type": "main",
1681             "index": 0
1682           }
1683         ]
1684       ]
1685     }
1686   ]
1687 }
```

```
1684     ]
1685   },
1686   "Create Documents Table and Match Function": {
1687     "main": [
1688       [
1689         {
1690           "node": "Create Document Metadata Table",
1691           "type": "main",
1692           "index": 0
1693         }
1694       ]
1695     ]
1696   },
1697   "Delete Old Data Rows": {
1698     "main": [
1699       [
1700         {
1701           "node": "Insert Document Metadata",
1702           "type": "main",
1703           "index": 0
1704         }
1705       ]
1706     ]
1707   },
1708   "Insert Document Metadata": {
1709     "main": [
1710       [
1711         {
1712           "node": "Download File",
1713           "type": "main",
1714           "index": 0
1715         }
1716       ]
1717     ]
1718   },
1719 },
1720 "active": false,
1721 "settings": {
1722   "executionOrder": "v1"
1723 },
1724 "versionId": "",
1725 "meta": {
1726   "instanceId": "51fc3d48a90a282b53eea2c2349b43eb49d341b4b82397ed1a3f093824973a13"
1727 },
1728 "tags": []
1729 }
```