

# **Information Management**

## **MIS 381N**

---

**ONLINE LEARNING BEHAVIOR & PATTERNS  
WITH  
SNOWFLAKE**

---

**Team ANS: Abhiroop Kumar, Nikhil Kumar, Simoni Dalal**

UNIVERSITY OF TEXAS AT AUSTIN | MCCOMBS SCHOOL OF BUSINESS

---

# Table of Contents

<b>3) Data Source</b> Brief Description	<b>4) Bronze Layer</b> Importing Raw Data	<b>5) Silver Layer</b> Detail & Data Load	
<b>6) STAR Schema</b> Silver Layer Architecture	<b>7) Gold Layer</b> Answering 3 Questions	<b>8) Streamlit Dashboard</b> Graphs and Visualizations	<b>10) Incremental Loads</b> Before vs After & it's Effect
<b>12) AI SQL Functions</b> Enriching Data with AI	<b>14) Cortex Search</b> Using Bronze Layer	<b>16) Cortex Analyst</b> Using Silver Layer	

# Brief Intro About Data

1

**Dataset Overview:** Kaggle dataset provides metadata on DataCamp courses and learning tracks, useful for analyzing learning patterns, course structures, and optimizing user learning journeys.

2

## Main Files:

- courses.csv: Core dataset with 23 columns (e.g., title, difficulty, technology, instructors).
- all\_tracks.csv: Overall info on learning tracks with 18 columns (track title, course count).

3

## Mapping Files:

- technology\_mapping.csv: Maps technology IDs to names.
- topic\_mapping.csv: Maps topic IDs to subject areas.



# Bronze Layer

## Bronze Schema Creation

Created a BRONZE schema as the container for raw CSV data, preserving original data format with VARCHAR columns.

## Data Loading Infrastructure

Established staging area and file format definitions to facilitate CSV file ingestion into the database.

## Data Import Execution

Used COPY INTO commands to load data from staged CSV files into corresponding BRONZE tables efficiently.

## Validation & Foundation

Checked for errors, row counts, and column structure to ensure data accuracy for next stages.



# Silver Layer - Key Steps



## Create Schema

Set up a new SILVER schema.

## Design ERD

Used STAR/Snowflake architecture.

## Build Tables

17 tables  
[2 Fact, 10 Dimension,  
4 Bridge, 1 Audit]  
+ 13 sequences

## Surrogate Keys

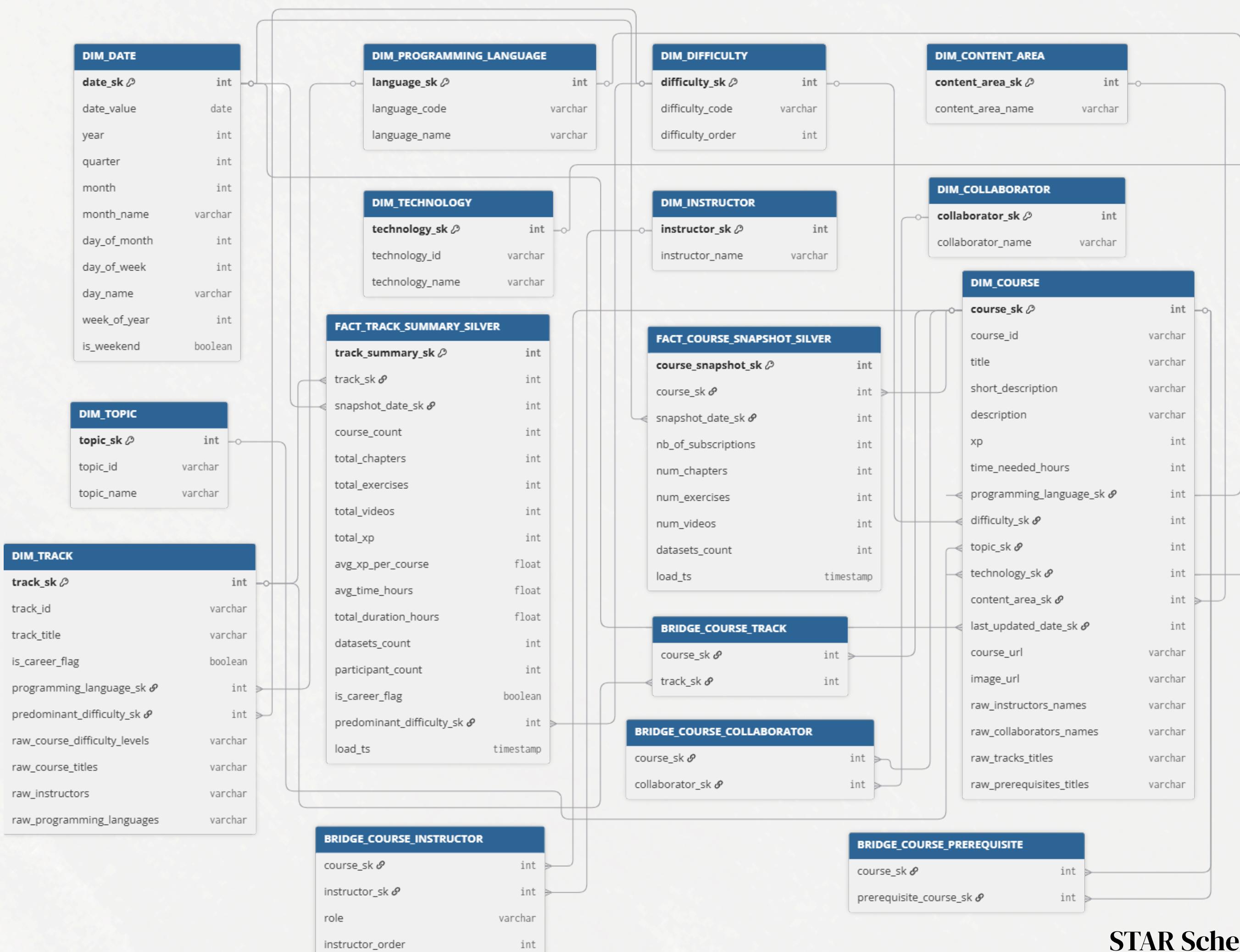
Replaced Bronze PKs with SKs for ID differentiation.

## Populate & Transform

Inserted data using queries and functions (e.g., TO\_DATE, TRIM, LOWER).

## Validate

Ran checks for clean, accurate data insertion.





# Gold Layer

Created a new GOLD schema with 3 dynamic tables to answer key business questions, ensuring near real-time data freshness from SILVER. Added an Audit Table for log tracking.

Use Case	Business Question	Required Silver Tables
1 - Programming Language Instructional Effort	Identify which programming languages require the most instructional effort (time, chapters, exercises, videos) across all courses.	FACT_COURSE_SNAPSHOT_SILVER, DIM_PROGRAMMING_LANGUAGE, DIM_DIFFICULTY, DIM.Course.
2 - Track-Level Summary	For each career or skill track, calculate total learning content (time, chapters, exercises, videos) and number of courses per track.	BRIDGE_COURSE_TRACK, DIM_TRACK, FACT_COURSE_SNAPSHOT_SILVER, DIM.Course.
3 - Course Difficulty Distribution & Curriculum Depth	Analyze course catalog distribution across difficulty levels (Beginner, Intermediate, Advanced) and identify which level contributes the most learning content (chapters, exercises, videos).	FACT_COURSE_SNAPSHOT_SILVER, DIM.Course, DIM_DIFFICULTY.

# Streamlit Dashboard

Our approach blends strategic insight, creative execution, and data-driven optimization - tailored specifically for growing businesses.

## Key Steps

Create a new stage in GOLD schema

Create a Streamlit python script that visualizes Gold table result-set data

Upload the Python file to the new stage

In SnowSQL, create a Streamlit object pointing to the stage & script

The Streamlit dashboard is ready!



# Visualization - 3 Use Cases

## By Track

**DataCamp Learning Content – Gold Layer Dashboard**

Visualizations built from GOLD tables: `G_LANGUAGE_INSTRUCTIONAL EFFORT`, `G_TRACK_CONTENT_SUMMARY`, `G_DIFFICULTY_CONTENT_SUMMARY`.

By Programming Language [By Track](#) By Difficulty

**Content Summary by Track**

Filter tracks by type:

All  Career Tracks  Skill Tracks

TRACK_TITLE	IS_CAREER_FLAG	COURSE_COUNT	TOTAL_TIME_HOURS	TOTAL
Associate Data Scientist in Python	✓	23	90	
Machine Learning Scientist in Python	✓	21	85	
Data Analyst in Python	✓	9	36	
Associate AI Engineer for Data Scientists	✓	9	32	
Python Data Fundamentals	✗	7	28	
Associate AI Engineer for Developers	✓	9	26	
Supervised Machine Learning in Python	✗	6	25	
Developing AI Applications	✗	8	21	
Natural Language Processing in Python	✗	5	20	
Time Series in Python	✗	5	20	

**Total learning hours per track**

**Number of courses per track**

## By Programming Language

**DataCamp Learning Content – Gold Layer Dashboard**

Visualizations built from GOLD tables: `G_LANGUAGE_INSTRUCTIONAL EFFORT`, `G_TRACK_CONTENT_SUMMARY`, `G_DIFFICULTY_CONTENT_SUMMARY`.

By Programming Language [By Track](#) By Difficulty

**Instructional Effort by Programming Language**

LANGUAGE_NAME	COURSE_COUNT	TOTAL_TIME_HOURS	TOTAL CHAPTERS	TOTAL_EXERCISES	TOTAL
Python	102	381	380	4,708	
R	17	32	56	641	
Sql	1	3	3	34	

**Total learning hours per programming language**

**Total chapters per programming language**

**LANGUAGE\_NAME**

## By Difficulty

**DataCamp Learning Content – Gold Layer Dashboard**

Visualizations built from GOLD tables: `G_LANGUAGE_INSTRUCTIONAL EFFORT`, `G_TRACK_CONTENT_SUMMARY`, `G_DIFFICULTY_CONTENT_SUMMARY`.

By Programming Language [By Track](#) [By Difficulty](#)

**Course Distribution and Content by Difficulty**

DIFFICULTY_CODE	DIFFICULTY_ORDER	COURSE_COUNT	TOTAL_TIME_HOURS	TOTAL CHAPTERS	TOTAL
Intermediate	2	3	10	11	
Advanced	3	1	4	5	

**Course count by difficulty level**

**DIFFICULTY\_CODE**

**Total learning hours by difficulty level**

**DIFFICULTY\_CODE**

# Incremental Load Workflow

**Pre-load validation**  
Silver & Gold  
Audit Log SPs

## Bronze Layer

- Create 4 new CSVs
- Create New Bronze stage
- Create 4 snowpipes
- Upload CSVs → Stage
- Auto-ingest via Snowpipe
- Validate Bronze loads

## Silver Layer

- SP1 → Update BTs
- SP2 → Update DTs
- SP3 → Update FTs
- Wrapper SP → Run all 3
- Validate Silver loads

## Gold Layer

- No manual step required
- Gold dynamic tables auto-refresh

## Incremental Data Verification

Call the Audit Log stored procedures again and view the audit tables. Validate the updated count of rows and change in aggregate values across all the tables.

# Incremental load

## SILVER\_LOAD\_AUDIT

#	AUDIT_ID	LOAD_TS	# SILVER_DIM_COURSE_ROW_COUNT	# SILVER_FACT_COURSE_ROWS	# SILVER_DIM_TRACK_ROW_COUNT	# SILVER_FACT_TRACK_ROWS
	1   2	1/12/2025   1/12/2025	116   120	116   236	115   117	115   232
1	1	2025-12-02 22:41:05.575	116	116	115	115
2	2	2025-12-02 22:54:01.515	120	236	117	232

## GOLD\_LOAD\_AUDIT

#	AUDIT_ID	LOAD_TS	# GOLD_LANGUAGE_TOTAL_COURSES	# GOLD_TRACK_TOTAL_COURSES	# GOLD_DIFFICULTY_TOTAL_SUBS
	1   2	1/12/2025   1/12/2025	116   120	214   219	16083645   16126245
1	1	2025-12-02 22:44:33.567	116	214	16083645
2	2	2025-12-02 22:54:03.508	120	219	16126245

# AI SQL Functions

## AI in Snowflake

### Use Case 1: AI\_COMPLETE

Predicts course ***title*** using external LLM  
(claude-3-7-sonnet)

Added column ***ai\_predicted\_title***, filled via  
AI\_COMPLETE



- 34% of titles have perfect similarity (1.0)
- 55%  $\geq 0.90 \rightarrow$  very high accuracy; only 1 value < 0.40

### Use Case 2: AI\_SIMILARITY

Calculates semantic similarity between ***title*** and  
***ai\_predicted\_title***

Added column ***ai\_title\_similarity***, updated using  
AI\_SIMILARITY

# AI SQL Columns - Bronze Layer

ID	A_TITLE	A_AI_PREDICTED_TITLE	A_AI_TITLE_SIMILARITY	A_DESCRIPTION	A_SHORT_DESCRIPTION
1	Hyperparameter Tuning in Python	Hyperparameter Tuning in Python	1	Building powerful machine learning models depends heavily on tuning hyperparameters.	Learn techniques for automating this process.
2	Manipulating Time Series Data in Python	Manipulating Time Series Data in Python	1	In this course you'll learn the basics of manipulating time series data.	In this course you'll learn how to work with time series data.
3	Dimensionality Reduction in Python	Dimensionality Reduction in Python	1	High-dimensional datasets can be overwhelming and leave you feeling lost.	Understand the concept of dimensionality reduction.
4	Machine Learning with PySpark	Machine Learning with PySpark	1	Spark is a powerful, general purpose tool for working with Big Data.	Learn how to make predictions using PySpark.
5	Time Series Analysis in Python	Time Series Analysis in Python	1	From stock prices to climate data, time series data are found everywhere.	In this four-hour course, learn how to analyze time series data.
6	Introduction to Python	Introduction to Python	1	Python is a general-purpose programming language that is easy to learn.	Master the basics of data science with Python.
7	Extreme Gradient Boosting with XGBoost	Extreme Gradient Boosting with XGBoost	1	Do you know the basics of supervised learning and want to take your skills to the next level?	Learn the fundamentals of XGBoost.
8	Intermediate Python	Intermediate Python	1	Learning Python is crucial for any aspiring data science practitioner.	Level up your data science skills with Intermediate Python.
9	Machine Learning for Finance in Python	Machine Learning for Finance in Python	1	Time series data is all around us; some examples are the weather and financial markets.	Learn to model and predict financial data using Python.
10	Data Manipulation with pandas	Data Manipulation with pandas	1	pandas is the world's most popular Python library, used for data manipulation.	Learn how to import and manipulate data with pandas.
11	Advanced Deep Learning with Keras	Advanced Deep Learning with Keras	1	This course shows you how to solve a variety of problems using deep learning.	Learn how to develop deep learning models.
12	Web Scraping in Python	Web Scraping in Python	1	The ability to build tools capable of retrieving and parsing information from websites.	Learn to retrieve and parse web data.
13	Biomedical Image Analysis in Python	Biomedical Image Analysis in Python	1	The field of biomedical imaging has exploded in recent years.	Learn the fundamentals of biomedical image analysis.
14	Model Validation in Python	Model Validation in Python	1	Machine learning models are easier to implement now more than ever.	Learn the basics of model validation.
15	Explainable AI in Python	Explainable AI in Python	1	Explainable AI is critical for data scientists and machine learning practitioners.	Gain the essential skills for explainable AI.
16	Working with Categorical Data in Python	Working with Categorical Data in Python	1	Being able to understand, use, and summarize non-numeric data is key to effective data science.	Learn how to manipulate categorical data.
17	Reinforcement Learning from Human Feedback	Reinforcement Learning from Human Feedback	1	Combine the efficiency of Generative AI with the understanding of Reinforcement Learning.	Learn how to make Generative AI better.
18	Introduction to Data Science	Introduction to Data Science	1	Data is an exciting and growing field that plays a significant role in our daily lives.	Gain an introduction to data science.

# Implementation of Cortex Search



**Why:** Avoid repetitive SELECT queries for searching relevant details; use Cortex Search for faster, smarter lookups.



**Setup:** Create 2 Cortex Search Services in BRONZE schema for tables COURSES and ALL\_TRACKS.



**Search Focus:** Select an appropriate column as the search focus

**COURSES**

Focus on TITLE column.

**ALL\_TRACKS**

Focus on TRACK\_TITLE column.



**Search Filters:** Include relevant columns as search filter parameters.



**Execution:** Run search services for related course/track titles.



**Validation:** Confirm search results return accurate and relevant values.

# Cortex Search Results

## COURSES\_CORTEX\_SEARCH

“python beginner”

```
[  
  {  
    "@scores": {  
      "cosine_similarity": 0.62192774,  
      "text_match": 1.480712100000000e-07  
    },  
    "DIFFICULTY_LEVEL": "1",  
    "PROGRAMMING_LANGUAGE": "python",  
    "TITLE": "Introduction to Python"  
  },  
  {  
    "@scores": {  
      "cosine_similarity": 0.6165951,  
      "text_match": 1.480712100000000e-07  
    },  
    "DIFFICULTY_LEVEL": "1",  
    "PROGRAMMING_LANGUAGE": "python",  
    "TITLE": "Introduction to Python for Developers"  
  },  
  {  
    "@scores": {  
      "cosine_similarity": 0.5640888,  
      "text_match": 1.480712100000000e-07  
    },  
    "DIFFICULTY_LEVEL": "1",  
    "PROGRAMMING_LANGUAGE": "python",  
    "TITLE": "Intermediate Python"  
  },  
]
```

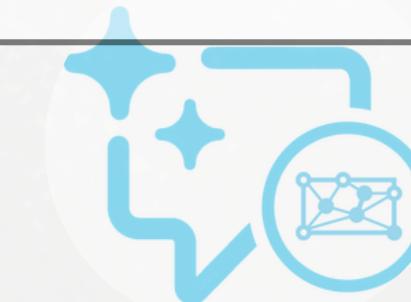
## TRACKS\_CORTEX\_SEARCH

“machine learning”

```
[  
  {  
    "@scores": {  
      "cosine_similarity": 0.5353421,  
      "text_match": 1.6448731  
    },  
    "COURSE_COUNT": "12",  
    "IS_CAREER": "Yes",  
    "TRACK_TITLE": "Machine Learning Engineer"  
  },  
  {  
    "@scores": {  
      "cosine_similarity": 0.51772237,  
      "text_match": 1.6448731  
    },  
    "COURSE_COUNT": "6",  
    "IS_CAREER": "No",  
    "TRACK_TITLE": "Supervised Machine Learning in Python"  
  },  
  {  
    "@scores": {  
      "cosine_similarity": 0.49985087,  
      "text_match": 1.6448731  
    },  
    "COURSE_COUNT": "21",  
    "IS_CAREER": "Yes",  
    "TRACK_TITLE": "Machine Learning Scientist in Python"  
  },  
]
```

# Cortex Analyst - Answer to Complex Queries

- Download YAML template
- Add silver schema tables + relationships
- Upload YAML to stage
- Open Cortex Analyst
- Select semantic model
- Open YAML
- Use chatbot to query the data



ans\_datacamp\_silver\_model

Which are the most difficult courses?

Cortex Analyst

This is our interpretation of your question:

Which are the most difficult courses?

COURSE_ID	TITLE	DIFFICULTY_CODE
114	ARIMA Models in Python	Advanced
107	CI/CD for Machine Learning	Advanced
115	Case Study: Building Software in Python	Advanced
82	Deep Learning for Images with PyTorch	Advanced
88	Deep Learning for Text with PyTorch	Advanced
80	Deep Reinforcement Learning in Python	Advanced
110	Efficient AI Model Training with PyTorch	Advanced
89	Ensemble Methods in Python	Advanced

Enter prompt

Run

---

# Meet the Team



**Abhiroop Kumar**



**Nikhil Kumar**



**Simoni Dalal**

# The End

THANK YOU

DROPPING THE TABLES & SIGNING OFF!