AI IDE(Snowflake) - Analysis

Exploratory Tools Analysis

June 24, 2025

Overview

Following points were considered for this exercise :

- Snowflake Integration
- AI-Assisted Development and Auto-Complete
- Schema/Context awareness
- Collaboration Features
- User Experience
- Existing preferences/already available or licensed tools

IDE with LLM integration

IDE	LLM
VS Code + Tabnine	Claude 3.5 Sonnet, Claude 3.7 Sonnet, Gemini 2.0 Flash, GPT-4o and Tabnine Protected
VS Code + Github Copilot	Claude Sonnet 4, Claude 3.7 Sonnet, Claude 3.7 Sonnet (thinking), Claude 3.5 Sonnet, Gemini 2.0 Flash, Gemini 2.5 Pro, GPT 4.1, GPT 4o, o1, o3-mini, o4-mini
Coginiti Pro	OpenAI, Anthropic and Gemini models
HEX	OpenAI model with future Anthropic integration
Navicat	OpenAI, Gemini, DeepSeek and Ollama
DataGrip+Github Copilot / AI Assistant	Claude Sonnet 4, Claude 3.7 Sonnet, Claude 3.7 Sonnet (thinking), Claude 3.5 Sonnet, Gemini 2.0 Flash, Gemini 2.5 Pro, GPT 4.1, GPT 4o, o1, o3-mini, o4-mini

Eval Details

IDE	Platform	Key Features	AI Integration	Pricing
Visual Studio Code	Desktop	Multiple extensions for snowflake with auto completion and GIT integration	GitHub Copilot, other Opensource extensions like Continue	Free
Snowflake Copilot	Web-based	Native AI assistant within SF, secure and complaint	Uses cortex, Mistral models	Included with the subscription
Hex	Web-based	Unified platform where each cell can execute python, SQL and visualizations	New prompts can be added to every cell, the tool is called Magic	Paid & Free Trial
Navicat	Desktop	Comprehensive Database management tool with advanced data modelling and visualizations	AI assistant with support for ChatGPT, Deepseek, Gemini and others	Paid and Free Trial (14 days)
Coginiti Pro	Desktop	Advanced SQL editor with introspection and schema management	In built AI Assistant	Paid & Free Trial
DataGrip	Desktop	Multiple extensions for snowflake with auto completion and GIT	In Built AI Assistant and Github Copilot	Paid & Free Trial

Other IDE's also available	Leading LLM's for SQL Generation
Cursor AI and DBeaver	• Claude Sonnet 4, 3.7 • GPT 4.1, 4o • Gemini 2.5 Pro

Criteria Breakdown

Category	Criteria	Example
Code Generation	SQL from existing SQLs/ flowchart	Using the existing data pipeline, IDE should understand and generate similar SQL for other tables
	Natural Language to SQL Conversion	Input: Show total sales pers region IDE should Generate: SELECT region, sum(sales) ....
	Answer Data modelling questions	IDE should help suggest optimal schemas, relationships, overall design strategies based on user questions
	Refactor/Lint existing SQL for better performance	Improve existing slow SQL and improve performance (CHNLPTNRIT_DB.CHNLPTNRIT_PXP_STG.PXP_BOOKINGS_DATA_LOAD_VW)
	suggestions for Enhancements to existing SQL's by using natural language	Improve existing queries and add additional logic
	Generate Java Script using set of SQLs/flowchart / Natural Language	Generate Java Script code for testing on subset data for overrides
	Git Integration and version control	Commit, push, pull code directly from the IDE, track history, review differences, revert changes
Testing	Generate test cases using code/ data flow	Test with empty tables, test negative or invalid revenue amounts, test boundary cases like end of month (sales opportunities)
	Generate SQL for test cases	SELECT ..... WHERE revenue < 0
Debugging	Code errors	IDE should highlight the missing parenthesis and should suggest necessary corrections
	Summaries to Details comparison	Compare the summary data to transaction level details for accuracy
	Extraction comparison to sources	Should highlight source to loaded table and highlight the mismatches
	Data issues	Should detect nulls, duplicates or give query
	Code reviews	IDE should suggest best practices, issues with code.
Documentation	Generate Data Flow Charts	Visualize data transformation paths(source -> staging -> aggregated -> reporting) (CPI data)
	Schema Modeling	Generate ER diagrams representing data tables, relationships, keys

Criteria 1 & 5 (Code Generation): SQL from Existing SQL or Flowchart

Test Goal: Evaluate whether the IDE with it's LLM assistant can adapt an existing SQL query or visual flowchart to a new use case with minimal user input.

IDE	Support Level	SQL Output Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Full	100 %	Given an Input prompt with existing SQL or flowchart, generates a new query bases on Input.	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Full	100 %	Given an Input prompt with existing SQL or flowchart, generates a new query bases on Input.	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Partial	100 %	Generates SQL with given Input prompt with existing SQL	Cannot attach flowcharts as an input
Hex (GPT 4o)	Partial	100 %	Generates SQL with given Input prompt with existing SQL	Cannot attach flowcharts as an input
Navicat (GPT 4o & Gemini 2.0 Flash)	Partial	100 %	Generates SQL with given Input prompt with existing SQL	Cannot attach flowcharts as an input
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Full	100 %	Generates SQL with given Input prompt with existing SQL	Cannot attach flowcharts as an input

Criteria 2: Natural Language to SQL

Test Goal: Evaluate whether the IDE with it's LLM assistant can accurately convert a plain English business question into a valid SQL query (eg: show the top 5 customers by total revenue in 1995)

IDE	Support Level	SQL Output Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅Full	100%	Given an Input prompt generates SQL.	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅Full	100%	Given an Input prompt generates SQL.	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Given an Input prompt generates SQL.	
Hex (GPT 4o)	✅Full	100%	Given an Input prompt generates SQL.	
Navicat (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Generates SQL with given Input prompt	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅Full	100%	Generates SQL with given Input prompt	Generates SQL with given Input prompt

Criteria 3: Answer Data Modelling Questions

Test Goal: Understand entity relationships, keys and constraints and answer data modelling-related questions.

IDE	Support Level	Output Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅Full	100%	Schema-aware assistant can explain key structures	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅Full	100%	Schema-aware assistant can explain key structures	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Schema-aware assistant can explain key structures	
Hex (GPT 4o)	❌None	0%		Not accurate when compared, strictly for SQL generation does not support regular questions
Navicat (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Schema-aware assistant can explain key structures	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅Full	100%	Schema-aware assistant can explain key structures	

Criteria 4 & 5: SQL Refactoring & Linting for Performance

Test Goal: Suggest improvements for readability, performance and best practices.

IDE	Support Level	SQL Output Accuracy and performance	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅ Full	100%	Refactors current SQL and explains the improvements made	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅ Full	100%	Refactors current SQL and explains the improvements made with additional suggestions	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	✅ Full	100%	Refactors the query and provides improvements and Explanations	
Hex (GPT 4o)	✅ Full	50%	Refactors the query and provides with improvements	Does not explain the improvements made and fails to answer follow up questions
Navicat (GPT 4o & Gemini 2.0 Flash)	✅ Full	100%	Refactors the query and provides with improvements	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅ Full	100%	Refactors current SQL and explains the improvements made with additional suggestions	

Criteria 6: Genrate JavaScript using set of SQLs/ Flowchart

Test Goal: Understand entity relationships, keys and constraints and answer data modelling-related questions.

IDE	Support Level	Code generation	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Full	100%	Generates relevant Java Script with GPT-4o and Sonnet 3.7	Gemini 2.0 Flash fails to understand the same prompt
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Full	100%	Generates relevant Java Script with GPT 4.1, Sonnet 4.0 and Gemini 2.5 Pro	Gemini 2.0 Flash fails here again to understand the same prompt
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Full	100%	Generates relevant Java Script	Gemini 2.0 Flash fails here again to understand the same prompt
Hex (GPT 4o)	Full	100%	Generates relevant Java Script	
Navicat (GPT 4o & Gemini 2.0 Flash)	Full	100%	Generates relevant Java Script	Gemini 2.0 Flash fails here again to understand the same prompt
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Full	100%	Generates relevant Java Script	

Criteria 7: GIT Integration

Test Goal: Assess whether the IDE offers built-in Git Integration.

IDE	Support Level	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅ Full	Built-in Git UI: Init,commit, diff, braches, merge, push/pull	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅ Full	Built-in Git UI: Init,commit, diff, braches, merge, push/pull	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	❌ None		No built –in Git
Hex (GPT 4o)	✅ Full	Git export to repository	Not available in trial version
Navicat (GPT 4o & Gemini 2.0 Flash)	❌ None		No built –in Git
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅ Full	Built-in Git UI: Init,commit, diff, braches, merge, push/pull	

Criteria 1 & 2 (Testing) : Generate test cases and SQLs using SQL/ Data Flow

Test Goal: Understand entity relationships, keys and constraints and answer data modelling-related questions.

IDE	Support Level	Output Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅Full	100%	Directly creates and generates multiple test cases	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅Full	100%	In Agent mode generates test cases in a SQL worksheet	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Generates multiple test cases	
Hex (GPT 4o)	✅Full	100%	Multiple test cases are given for the SQL query	
Navicat (GPT 4o & Gemini 2.0 Flash)	✅Full	100%	Generates multiple test cases	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅Full	100%	Generates multiple test cases	

Criteria 1 (Debugging): Debugging and Error Detection

Test Goal: Detect and fix SQL syntax errors, query logic like invalid joins or NULL filters

IDE	Support Level	Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Full	100%	Suggests new query within the workspace	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Full	100%	Fixes and compares the query with github diff view, making it easy to understand	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Full	100%	Lists out errors and makes necessary corrections	
Hex (GPT 4o)	Partial	40%		Only generates a new fixed query no explanation and struggles with complex queries
Navicat (GPT 4o & Gemini 2.0 Flash)	Full	100%	Lists out errors and makes necessary corrections	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Full	100%	Gives suggestions to make necessary corrections within the current worksheet	

Criteria 2: Summaries vs Detail Comparison

Test Goal:

IDE	Support Level	Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	✅ Full	100%	Generates SQL query for comparison	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	✅ Full	100%	Generates SQL query for comparison	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	✅ Full	100%	Uses Analog Catalog for reusability	
Hex (GPT 4o)	✅ Full	100%	Generates SQL query for comparison	
Navicat (GPT 4o & Gemini 2.0 Flash)	✅ Full	100%	Visual query builder makes it easy to visualize the query flow	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	✅ Full	100%	Is schema-aware, generates correct SQL query	

Criteria 3: Extraction comparison to sources

Test Goal: Compare extracted/ staging data against source table, detect mismatches, suggest improvements

IDE	Support Level	Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Full	100%	Generates queries for validating source and destination tables	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Full	100%	Generates queries for validating source and destination tables	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Full	100%	Generates queries for validating source and destination tables	
Hex (GPT 4o)	Full	100%	Generates queries for validating source and destination tables	
Navicat (GPT 4o & Gemini 2.0 Flash)	Full	100%	Generates queries for validating source and destination tables and comes with data synchronization tool which can compare 2 tables	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Full	100%	Generates queries for validating source and destination tables	

Criteria 4: Data issues

Test Goal: Detect anomalies such as missing values, duplicates, outliers

IDE	Support Level	Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Partial	30%	Generates queries for validation but does not perform comparision	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Partial	30%	Generates queries for validation but does not perform comparision	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Partial	30%	Generates queries for validation but does not perform comparision	
Hex (GPT 4o)	Partial	30%	Generates queries for validation but does not perform comparision	
Navicat (GPT 4o & Gemini 2.0 Flash)	Partial	30%	Generates queries for validation but does not perform comparision	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Partial	30%	Generates queries for validation but does not perform comparision	

Criteria 5: Code reviews

Test Goal: Effective code review to ensure reliability, maintainability and performance.

IDE	Support Level	Accuracy	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Full	100%	Performs code review with suggestions	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Full	100%	Performs code review with suggestions	
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Full	100%	Performs code review with suggestions	
Hex (GPT 4o)	Full	100%	Performs code review with suggestions	
Navicat (GPT 4o & Gemini 2.0 Flash)	Full	100%	Performs code review with suggestions	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Full	100%	Performs code review with suggestions	

Criteria 1: Generate Data Flow Charts and Schema Model

Test Goal: Evaluate whether the IDE or AI assistant can generate ER diagrams and Data flow charts

IDE	Support Level	How it supports the task	Limitations
VS Code + Tabnine (GPT-4o & Sonnet 3.5 & Gemini 2.0 Flash)	Partial	It can generate basic charts in SQL	
VS Code + Git Copilot (GPT 4.1 & Sonnet 4.0 & Gemini 2.5 Pro)	Partial	It can generate ER diagrams to understand the schema and Key relationships using mermaid extension in agent mode	Claude models performs better than other models
Coginiti Pro (GPT 4o & Gemini 2.0 Flash)	Partial	It can generate basic charts in SQL	
Hex (GPT 4o)	None		Cannot generate data flow charts but can generate regular visualization charts
Navicat (GPT 4o & Gemini 2.0 Flash)	Partial	It can generate basic charts in SQL	
DataGrip+Github Copilot / AI Assistant (GPT 4.1 & Sonnet 4.0)	Partial	It can generate ER diagrams to understand the schema and Key relationships using mermaid extension in agent mode also has built-in support for ER diagrams	

IDE	Pros	Cons
VS Code Extensions: Github Copilot Copilot,Snowflake, Mermaid LLM: Claude Sonnet 4/ GPT4o, GPT4.1	1. Full git integration 2. String support for LLM-driven SQL generation, refactoring, testing, debugging 3. Mermaid support for visual schema within markdown	1. Diagram rendering requires plugins 2. No built-in data quality tools
VS Code Extension: Tabnine, Snowflake, Mermaid LLM: Sonnet 3.7/ GPT4o	1. Full git integration 2. String support for LLM-driven SQL generation, refactoring, testing, debugging 3. Mermaid support for visual schema within markdown	1. Diagram rendering requires plugins 2. No built-in data quality tools
Coginiti Pro LLM: GPT 4o	1. Good for schema modelling 2. Data exploration charts, pivoting & filtering like excel 3. Supports Cogs for code reusability	1. No GIT integration 2. Do not accept flowcharts as an input
Navicat LLM: GPT 4o	1. Only tool with visual Query builder 2. Built in Visualization tool	1. No GIT integration 2. Do not accept flowcharts as an input
Hex LLM: GPT 4o	1. Collaborative notebooks 2. Git integration on Enterprise plan 3. Built-in visual charts	1. 1. Does not support follow up questions, only for SQL. 2. No Debugging features
DataGrip Extension: Github Copilot/ AI assistant, Mermaid LLM: Claude Sonnet 4/ GPT4o, GPT4.1	1. Rich Schema modelling, ER-diagrams and git integration 2. Live syntax validations, better UI for data exploration and can attach database schema 3. String support for LLM-driven SQL generation, refactoring, testing, debugging 4. Mermaid support for visual schema within markdown	1. Heavier on memory 2. Schema attached has a limitation for tables 3. No built-in data quality tools 4. Flowcharts can't be attached to AI Assistant as input, but Github copilot can.

Conclusion

Best All-Around IDE Setup:

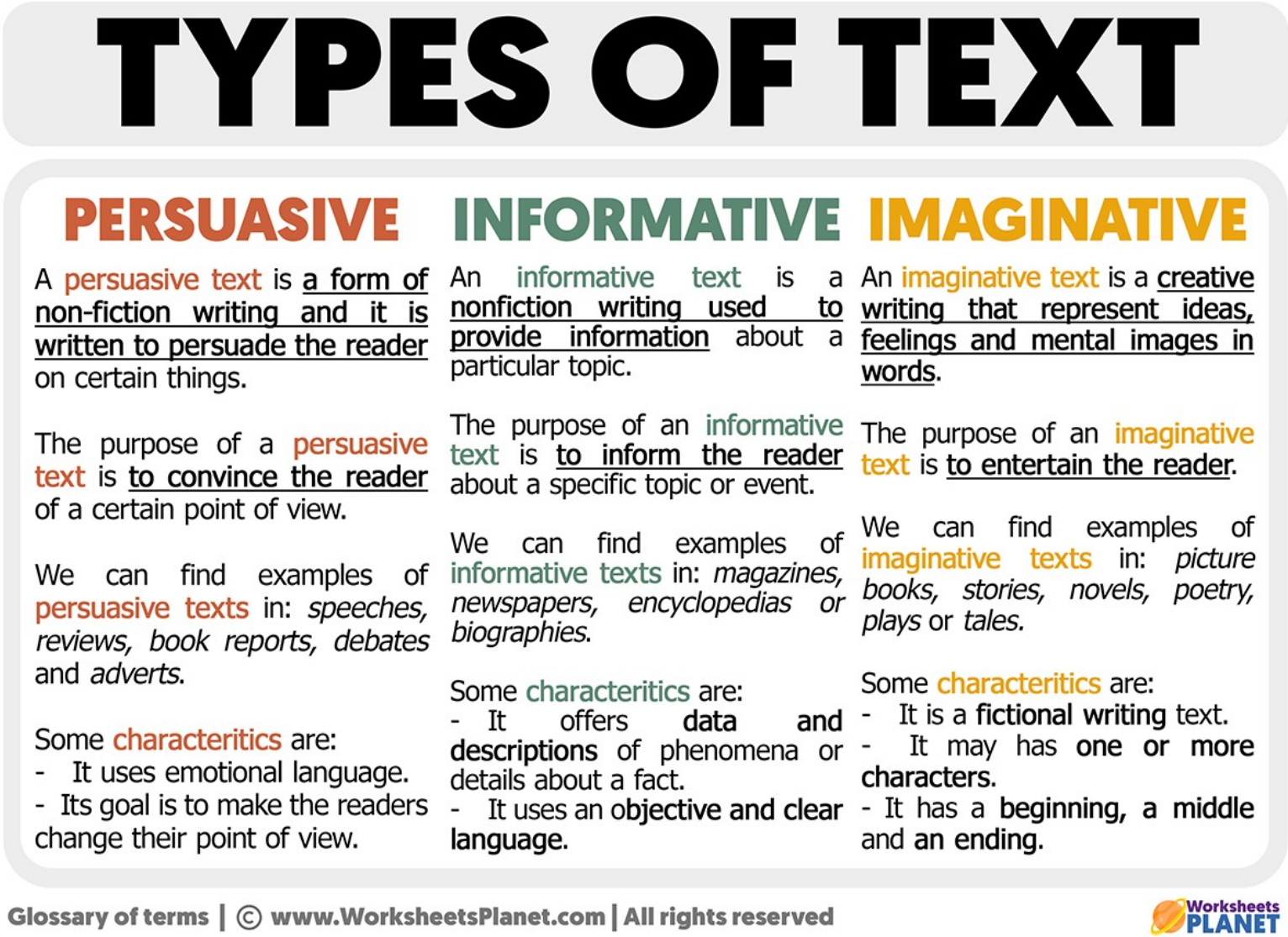
- Visual Studio Code paired with GitHub Copilot and DataGrip with Github Copilot/ AI Assistant are the most comprehensive solutions for Snowflake development involving AI driven workflows. Selecting either of these solutions will deliver a significant and immediate uplift in productivity, code quality, and operational efficiency.
- IDE: VS Code and DataGrip
- LLMs: Claude Sonnet 4 & GPT4.1/4o
- Extensions:
  - VS Code: Snowflake, Mermaid, Github Copilot
  - DataGrip: Mermaid, Github Copilot/ AI Assistant


CISCO

Bak Up

Model Name	Why Use It	Pros	Cons
OpenAI (text-embedding-3-small)	Best accuracy, highly semantic	Accurate, multilingual, handles complex logic	Paid, requires internet
sentence-transformers	Good open-source embeddings	Free, runs locally, many models to choose	Slightly lower semantic accuracy
bge-small-en / e5-base	Optimized for search	Very fast and compact	Limited multi-language
Cohere, Gemini	Alternative commercial options	Scalable, high quality	May need API keys
LLaMA / Mistral	Needs adapter/fine-tuning for embedding use	Unified with same model	Complex setup, resource-heavy

Bak Up



Bak Up

MLA In- Text Citation

10. Indirect source

When a writer's or speaker's quoted words appear in a source written by someone else, begin the citation with the abbreviation "qtd. in."

"When lion sightings become common," says Fjelline, "trouble often follows" (qtd. in Robinson 30).

Column 1	Column 2
Braedon Haas: Dui a tincidunt quisque tempus.	**Shamar Brennan**: Aenean nec viverra a orci.
Monserrat Banks: Fringilla dolor ultricies aliquam dolor.	Avery Potts: Nam nec pretium aenean faucibus augue sed.
Janiah Ho: Condimentum ultricies elementum consequat sit tempus aliquet.	Alina Ashley: Natoque hendrerit nec tellus rhoncus enim natoque hendrerit.
Monica Ferguson: Eget ultricies in vel sed cras.	Nelson Fitzpatrick: Vulputate ante vulputate pretium quam in curabitur.
Landon Kirby: Fringilla maecenas donec maecenas pede.	Dominique Drake: Consequat eget dolor duis eu magnis tincidunt quis.
Deja Lucas: Et commodo vulputate quam eget dolor	Magdalena Howell: Quisque felis ridiculus

IDE Suitability

Category	Criteria	Cogniti Pro	Hex	DataGrip	VS Code	Navicat
Code Generation	SQL from existing SQLs/ flowchart	✓	✓	✓	✓	✓
	Natural Language to SQL Conversion	✓	✓	✓	✓	✓
	Answer Data modelling questions	✓	✓	✓	✓	✓
	Refactor/Lint existing SQL for better performance	✓	✓	✓	✓	✓
	Generate Java Script using set of SQLs/flowchart / Natural Language	✓	✓	✓	✓	✓
	Git Integration and version control	✗	✓	✓	✓	✗
Testing	Generate test cases using code/ data flow		✗			✓
	Generate SQL for test cases	✓	✗	✓	✓	✓
Debugging	Code errors					
	Summaries to Details comparison	✓	✗	✓	✓	✓
	Extraction comparison to sources	✓	✓			✓
	Data issues	✓	✓			✓
	Code reviews	✓	✓			✓
Documentation	Generate Data Flow Charts	✗	✓			
	Schema Modeling	✗	✗			

✓ Partial

Eval Details

IDE	Platform	Key Features	AI Integration	Pricing
Visual Studio Code	Desktop	Multiple extensions for snowflake with auto completion and GIT integration	GitHub Copilot, other Opensource extensions like Continue	Free
Snowflake Copilot	Web-based	Native AI assistant within SF, secure and complaint	Uses cortex, Mistral models	Included with the subscription
Hex	Web-based	Unified platform where each cell can execute python, SQL and visualizations	New prompts can be added to every cell, the tool is called Magic	Paid & Free Trial
Navicat	Desktop	Comprehensive Database management tool with advanced data modelling and visualizations	AI assistant with support for ChatGPT, Deepseek, Gemini and others	Paid and Free Trial (14 days)
Coginiti Pro	Desktop	Advanced SQL editor with introspection and schema management	In built AI Assistant	Paid & Free Trial

Other IDE's also available	Leading LLM's for SQL Generation
DataGrip, Cursor AI and DBeaver	• Claude Opus 4 • Arctic-Text2SQL-R1 (32B and 14B) • GPT 4.1 • Gemini 2.5 Pro • Qwen3 (when fine-tuned, performs better than GPT 4o)

Eval Summary

Experience Level	Recommendations	Comments
Mid Level to Senior	Continue with their current preferred tool, cross referencing Claude Opus 4 independently	Next steps in DevOps process also needs to be considered to reduce overall timelines
Entry to Junior	Should use the IDE's finalized part of this analysis	Risks over-reliance and reduced critical thinking

Column %	Coca-Cola	Diet Coke	Coke Zero	Pepsi	Diet Pepsi	Pepsi Max	New column
25 to 29	12%	12%	12%	12%	12%	12%	NaN
30 to 34	10%	10%	10%	10%	9% ▼	10%	NaN
35 to 39	11%	11%	12%	11%	12%	11% ▼	NaN
40 to 44	12%	12%	12%	12%	12%	11%	NaN
45 to 49	8%	8%	8%	8%	8%	7% ▼	NaN
50 to 54	12%	12%	11%	12%	12%	13% ▲	NaN
55 to 64	16%	16%	15%	16%	17%	16%	NaN
65 or more	7%	7%	7%	7%	7%	7%	NaN
NET	100%	100%	100%	100%	100%	100%	NaN
New row	NaN	NaN	NaN	NaN	NaN	NaN	NaN