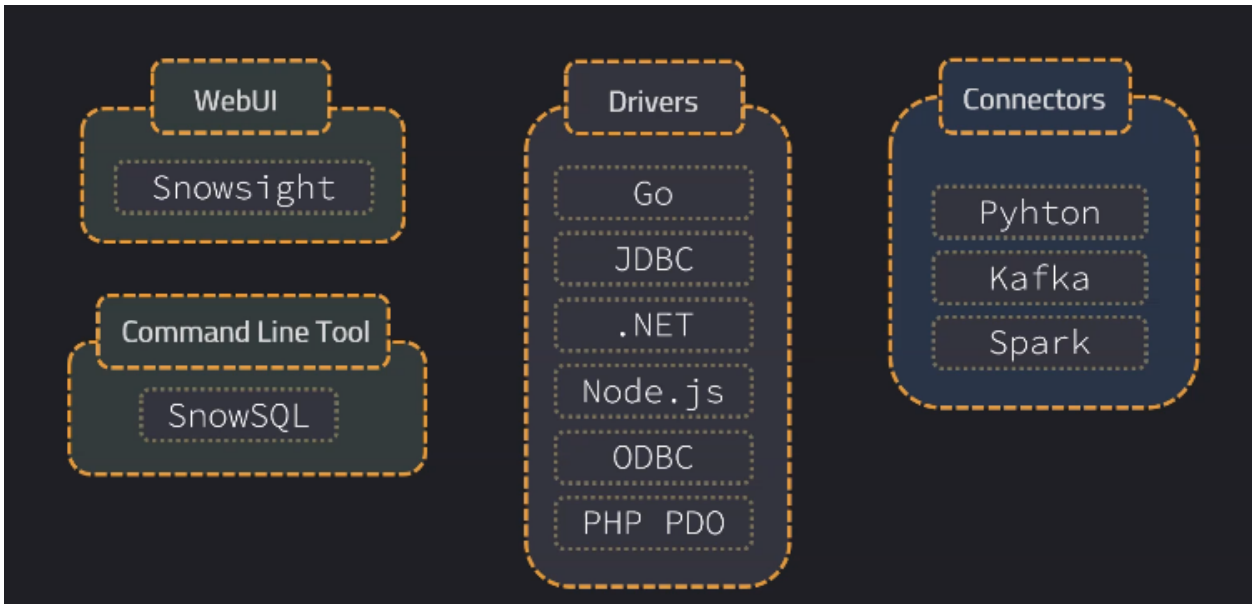
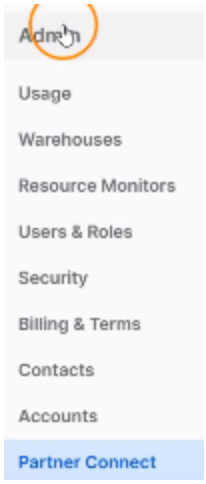
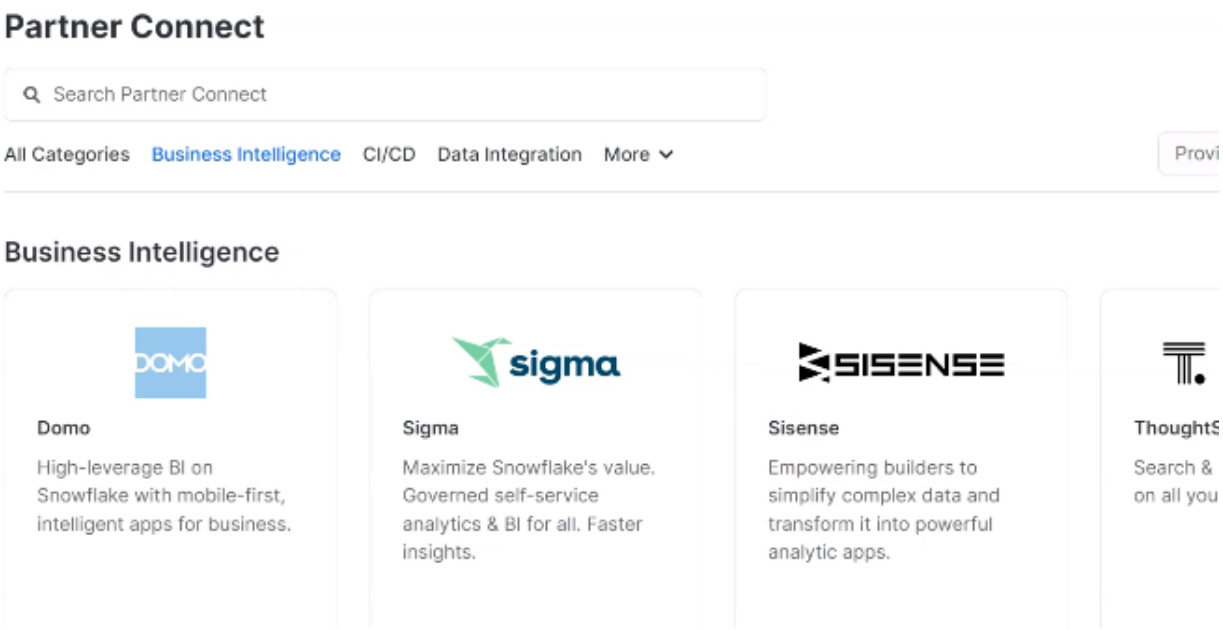
Connectors, Drivers & partner connect

1. TO write SQL…we can use SnowSight WebUI…and also we can install snowsql tool to run sql queries on CLI and can connect to snowflake
2. 

3. We can also connect to snowflake partners and use them

4. 
Admin
Usage
Warehouses
Resource Monitors
Users & Roles
Security
Billing & Terms
Contacts
Accounts
Partner Connect


Partner Connect
Search Partner Connect
All Categories
Business Intelligence
CI/CD
Data Integration
More
Business Intelligence
Domo
High-leverage BI on Snowflake with mobile-first, intelligent apps for business.
Sigma
Maximize Snowflake's value. Governed self-service analytics & BI for all. Faster insights.
Sisense
Empowering builders to simplify complex data and transform it into powerful analytic apps.

5. **Snowflake connectors** are software components that allow you to connect to Snowflake from a variety of applications and tools. Connectors can be used to load data into Snowflake, transform data in Snowflake, and query data in Snowflake.

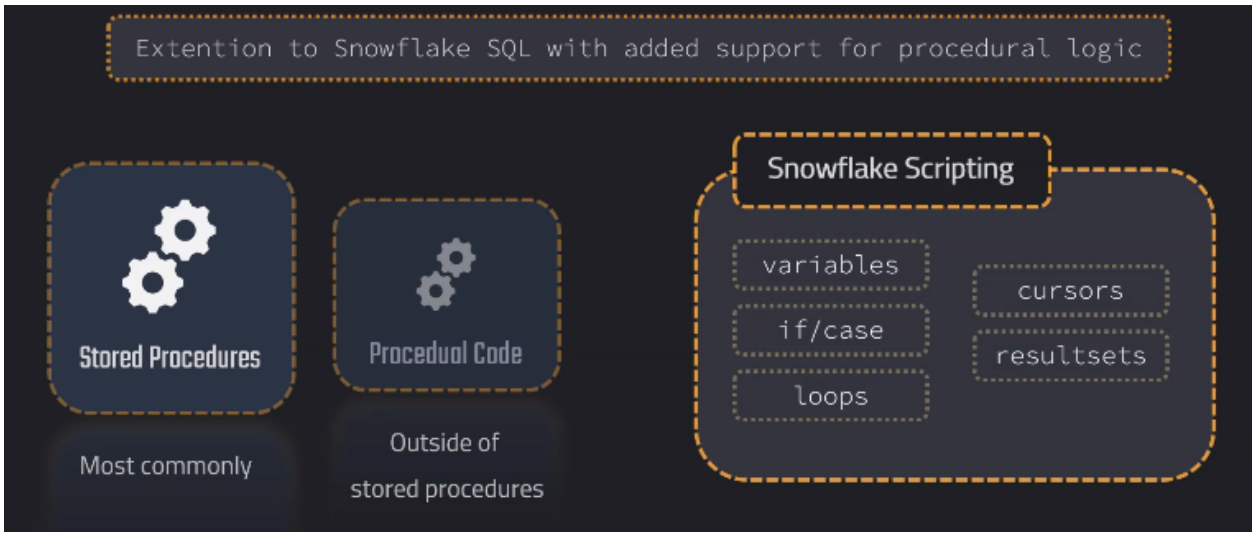
   Snowflake drivers are low-level software components that allow you to interact with the Snowflake database using a specific programming language. Drivers are typically used to develop custom applications that interact with Snowflake.

   Snowflake Partner Connect is a program that allows Snowflake partners to integrate their products and services with Snowflake. Partner Connect makes it easy for Snowflake customers to find and use certified Snowflake partner solutions.

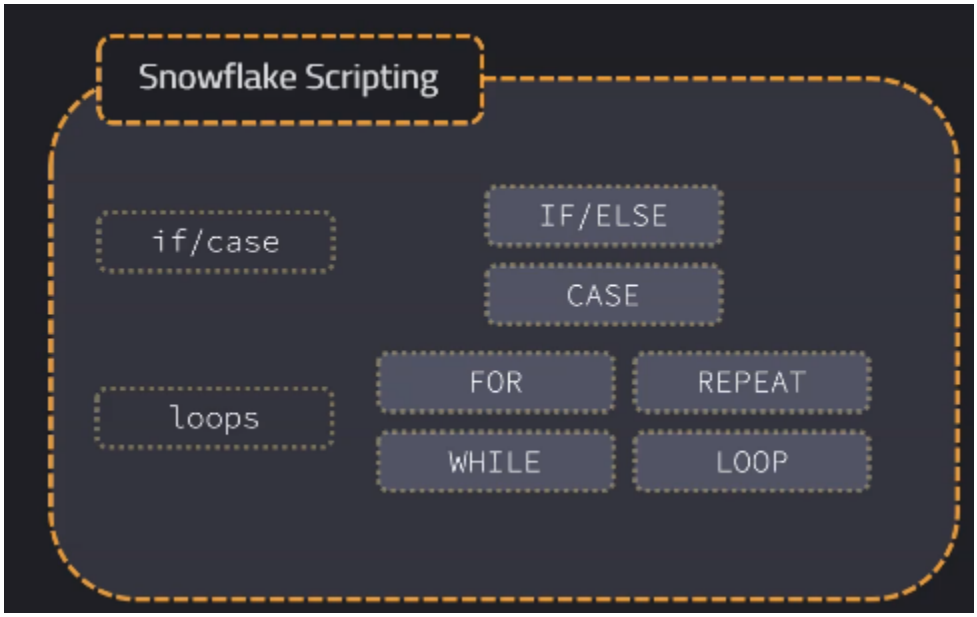
6. Here are some examples of how connectors, drivers, and Partner Connect are used in Snowflake:

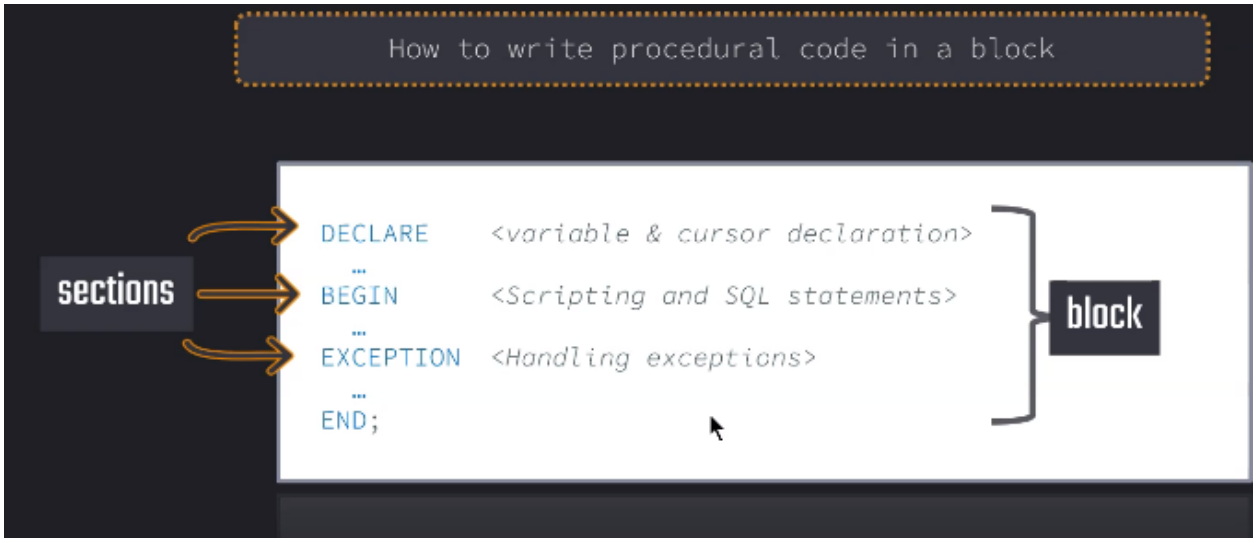
   - **A company could use a connector to load data from its CRM system into Snowflake.**
   - **A data scientist could use a driver to write a custom Python script to query data in Snowflake.**
   - **A business intelligence analyst could use a Partner Connect solution to generate reports from Snowflake data.**

Snowflake scripting

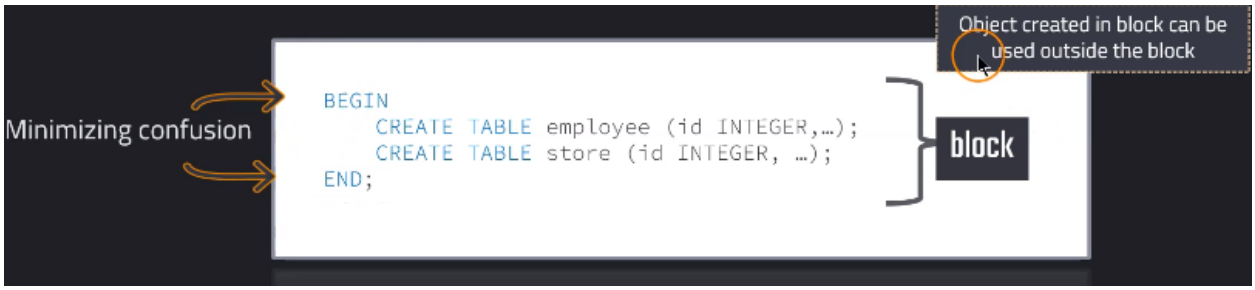
1. 

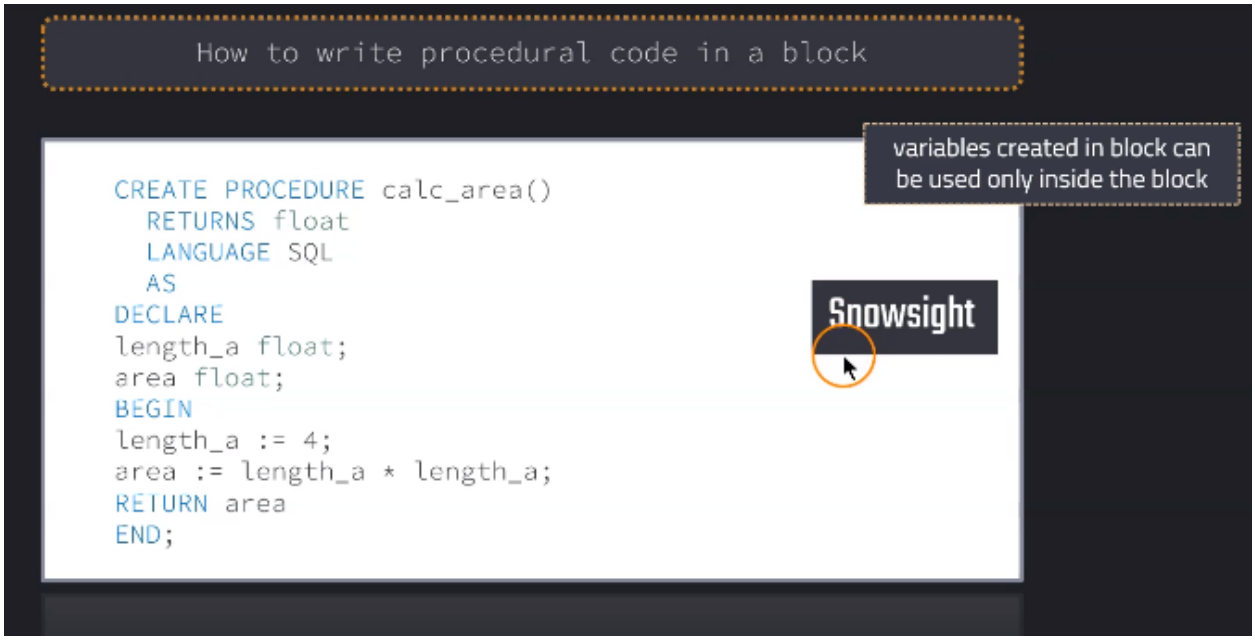
2. We can also use if/case and loops



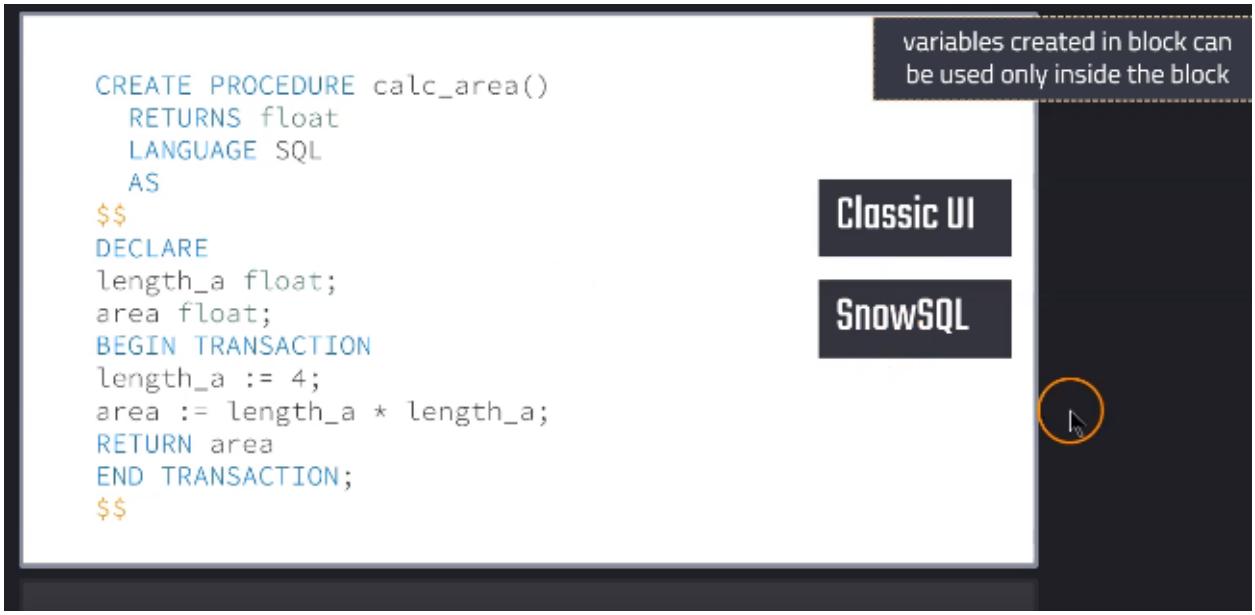
3. 

4. So we have the declare section in which we can declare variables and cursors and then we have the begin section This is where the scripting and the SQL statements start, and then we have the exception section in which we can optionally handle some exceptions.
5. Declare and exception are optional
6. We can create tables(objects) using begin and can use this objects later

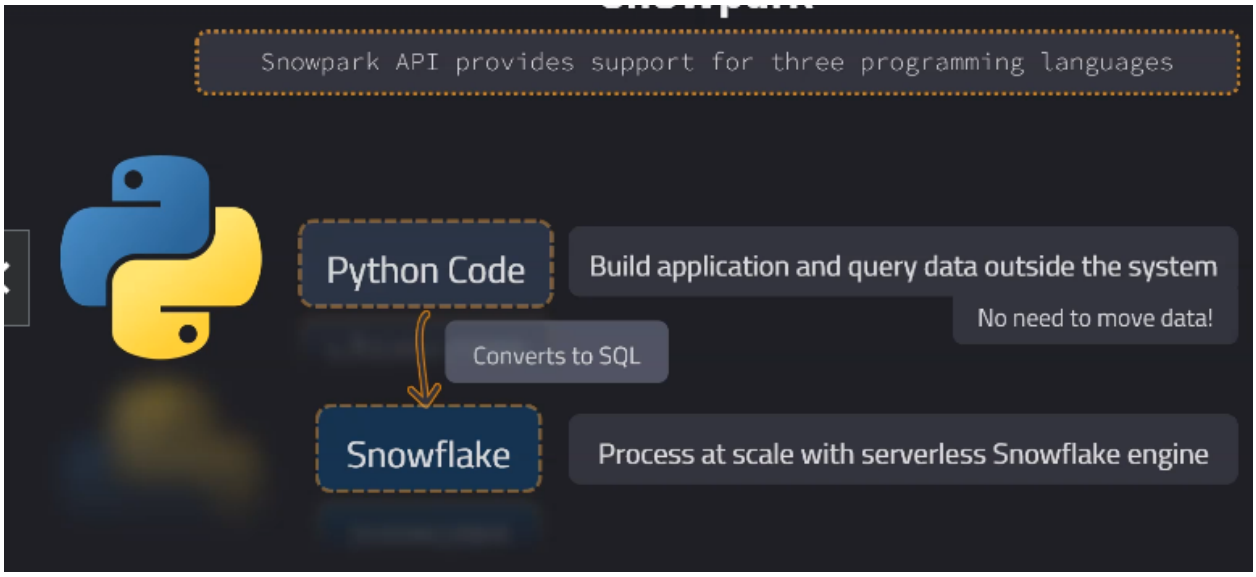
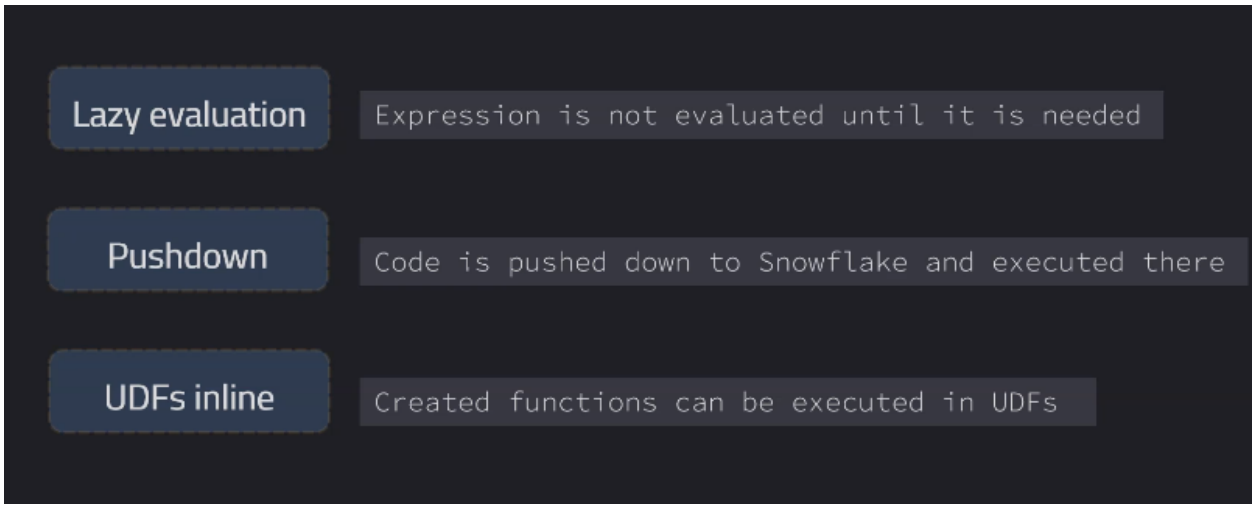


7. 

8. But if we are using classical UI and SnowSQl we use $$



SnowPark

1. 
Snowpark API provides support for three programming languages
Python Code
Build application and query data outside the system
No need to move data!
Converts to SQL
Snowflake
Process at scale with serverless Snowflake engine
2. 
Lazy evaluation
Expression is not evaluated until it is needed
Pushdown
Code is pushed down to Snowflake and executed there
UDFs inline
Created functions can be executed in UDFs

1. Snowpark is a framework for developing and running applications that process data in Snowflake using Python, Java, or Scala. Snowpark provides a high-level API for working with Snowflake data, making it easy to develop and run data processing applications without writing SQL.

   Snowpark is built on top of the Snowflake Spark Connector, which allows Snowpark to push computations to Snowflake, taking advantage of Snowflake's powerful compute engine. This can significantly improve the performance of data processing applications.

2. Snowpark can be used to develop a wide variety of applications, including:

   - **Data pipelines:** Snowpark can be used to develop data pipelines that load, transform, and analyze data in Snowflake.
   - **Machine learning:** Snowpark can be used to develop machine learning applications that train and deploy machine learning models in Snowflake.
   - **Data science:** Snowpark can be used to develop data science applications that explore and analyze data in Snowflake.

3. Here is an example of a simple Snowpark application that queries data from a Snowflake table:

Python

```python
import snowpark

# Create a Snowpark Spark session
session = snowpark.Session.builder.getOrCreate()

# Load the data from the Snowflake table
df = session.read.table("my_table")

# Filter the data
df = df.filter(df["column_name"] > 10)

# Show the results
df.show()
```