SYSTEM SNOWFLAKE

projekt zrealizowany w ramach przedmiotu
Hurtownie Danych i Przetwarzanie Analityczne
na Politechnice Poznańskiej
we współpracy z Santander Bank Polska

grupa projektowa:
Maryna Kotok
Inga Spychała
Monika Wnuck-Lipińska

opiekun:
Hubert Półtorak

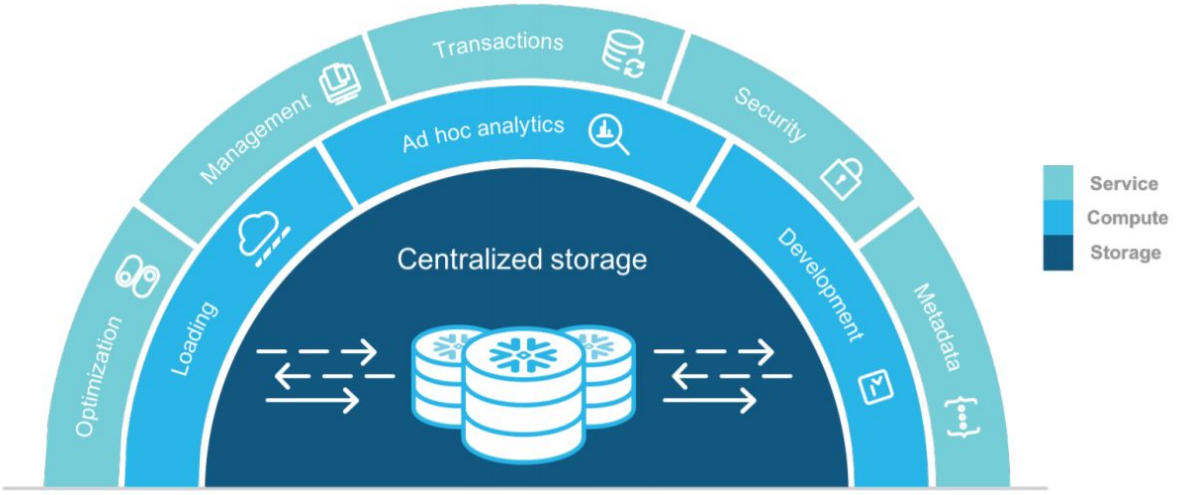
Poznań, 14.06.2019 r.

SPIS TREŚCI

1. System Snowflake

1.1. Architektura

System Snowflake to nowatorski projekt, który fizycznie oddziela, ale logicznie integruje trzy części - pamięć, obliczenia i usługi, dostarczając potrzebne zasoby dokładnie wtedy, gdy są potrzebne. Tego typu architektura nazywana jest multi-cluster/shared data i składa się z komponentów, które zostały zaprezentowane na rysunku 1.1.1.



Rysunek 1.1.1. Architektura Snowflake

Pamięć (Centralized storage) jest trwałą warstwą pamięci dla danych, przechowywanych w systemie Snowflake. Znajduje się w skalowalnej i chmurowej usłudze do przechowywania danych, np. Amazon S3, która zapewnia replikację danych, skalowanie i dostępność bez potrzeby zarządzania przez klientów. Snowflake optymalizuje i przechowuje dane w formacie kolumnowym, w warstwie pamięci zorganizowanej w bazie danych określonej przez użytkownika.

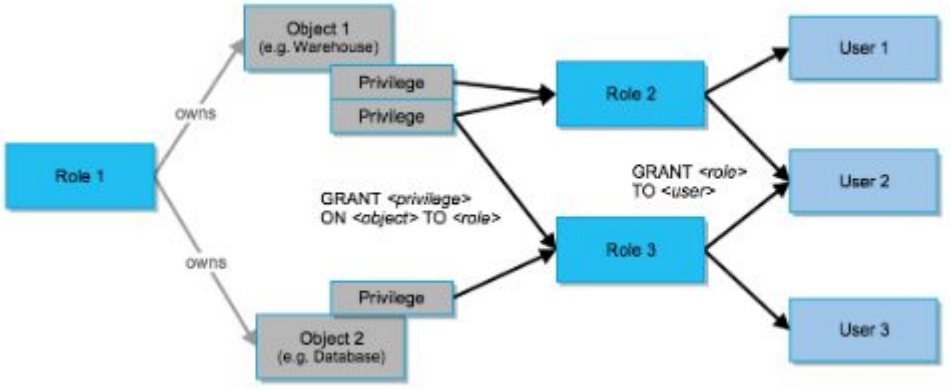
Zasoby obliczeniowe (Compute) - zbiór niezależnych zasobów obliczeniowych, które wykonują zadania, dotyczące przetwarzania danych, wymagane do zapytań. Aby przydzielić zasoby obliczeniowe do zadań (ładowanie, transformacja i zapytania), użytkownicy tworzą wirtualne hurtownie danych. Takie hurtownie mają możliwość uzyskania dostępu do dowolnej bazy danych w warstwie pamięci, do której został udzielony dostęp i mogą być dynamicznie tworzone, zmieniane i usuwane, gdy zasoby potrzebują zmian. Kiedy wirtualne hurtownie danych wykonują zapytania, automatycznie buforują one dane z warstwy pamięci. Taka hybrydowa architektura łączy ujednoliconą pamięć architektury shared-disk z korzyściami wynikającymi z architektury shared-nothing.

Usługi (Services) - zbiór usług systemowych, które są odpowiedzialne za obsługę infrastruktury, bezpieczeństwa, metadanych i za optymalizację w całym systemie Snowflake. Należą do nich: metadane, bezpieczeństwo, kontrola dostępu i infrastruktura. Usługi w tej warstwie płynnie komunikują się z aplikacjami klienckimi, aby koordynować przetwarzanie zapytań i zwracanie wyników. Warstwa usług zachowuje metadane o danych

przechowywanych w Snowflake, co umożliwia nowym hurtowniom danych natychmiastowy dostęp do danych.

1.2. Uprawnienia

W modelu Snowflake dostęp do zabezpieczonych obiektów jest udzielany za pomocą uprawnień przypisanych do ról, które z kolei są przypisane do innych ról lub użytkowników. Podejście Snowflake do kontroli dostępu łączy aspekty modeli DAC oraz RBAC.



Rysunek 1.2.1. Podejście Snowflake do kontroli dostępu

System został zaprojektowany w sposób zapewniający większą kontrolę i elastyczność.

Na koncie Snowflake dostępne są cztery predefiniowane role systemowe: ACCOUNTADMIN, SECURITYADMIN, SYSADMIN i PUBLIC. Użytkownicy z odpowiednim dostępem mogą edytować role zdefiniowane przez system, a także tworzyć role niestandardowe. Wszyscy użytkownicy Snowflake mają domyślnie przypisaną rolę PUBLIC, która pozwala na zalogowanie się do systemu i podstawowy dostęp do obiektów. Później mogą zostać przypisane uprawnienia kontroli dostępu, które określają, kto może uzyskać dostęp i wykonywać operacje na konkretnych obiektach w Snowflake.

Role standardowe:

- *ACCOUNTADMIN (Administrator konta)* - w jej skład wchodzą predefiniowane role systemowe - SYSADMIN i SECURITYADMIN. Jest to rola najwyższego poziomu w systemie i powinna być przyznawana tylko ograniczonej liczbie użytkowników na koncie.

- *SECURITYADMIN (Administrator bezpieczeństwa)* - rola przy pomocy, której możliwe jest tworzenie, monitorowanie i zarządzanie na poziomie użytkowników i ról. Dokładniej posiada uprawnienia do:

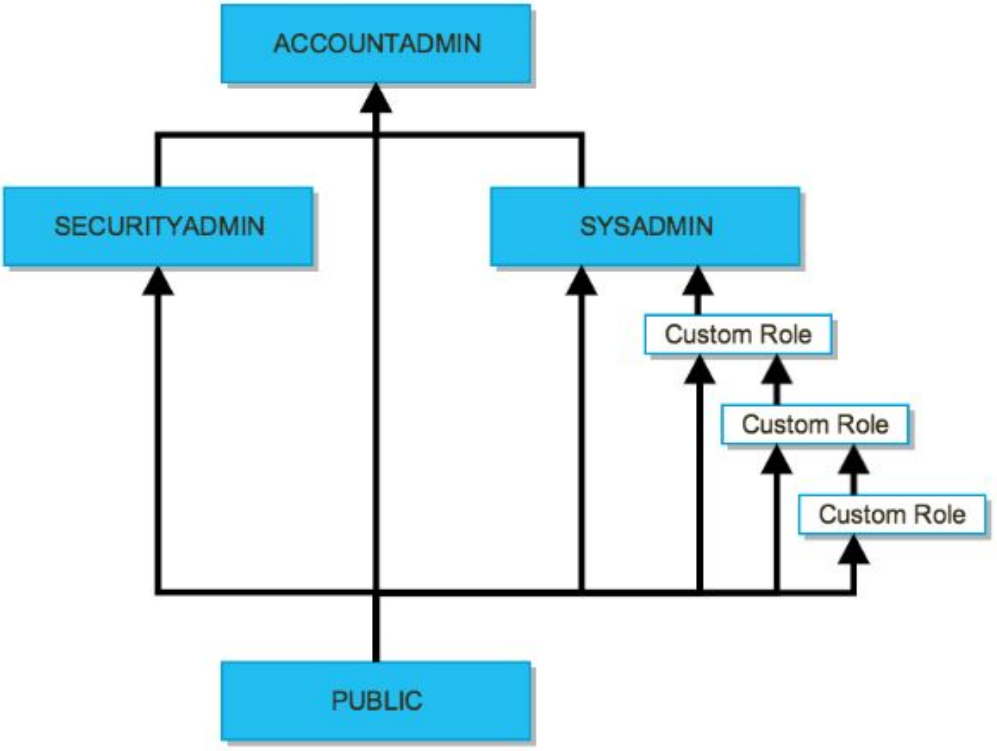
  - tworzenia użytkowników i ról na koncie (i nadawania uprawnień innym rolom),

  - modyfikacji i monitorowania dowolnego użytkownika, roli lub sesji,

- modyfikacji i przydzielania roli, w tym odebranie jej.

- *SYSADMIN (Administrator systemu)* - rola, która ma uprawnienia do tworzenia hurtowni i baz danych (a także innych obiektów) na koncie. Jeśli zgodnie z zaleceniami zostanie utworzona hierarchia ról, która ostatecznie przypisuje wszystkie role niestandardowe do roli SYSADMIN, ta rola będzie również miała możliwość nadawania uprawnień do hurtowni, baz danych i innych obiektów innym rolom.

- *PUBLIC* - pseudo-rola, która jest automatycznie przyznawana każdemu użytkownikowi i każdej roli na koncie. Rola PUBLIC może posiadać zabezpieczane obiekty, tak jak każda inna rola. Obiekty będące jej własnością są z definicji dostępne dla każdego innego użytkownika i roli na koncie. Zazwyczaj jest ona używana w przypadkach, gdy nie jest potrzebna jawna kontrola dostępu, a wszyscy użytkownicy są traktowani jako równi w zakresie swoich praw dostępu.



Rysunek 1.2.2. Hierarchia ról i dziedziczenie uprawnień

Istnieje możliwość definiowania własnych ról. Role niestandardowe mogą być tworzone zarówno przez role SECURITYADMIN, jak i przez każdą rolę, do której przyznano uprawnienie CREATE ROLE. Domyślnie nowo utworzona rola nie jest przypisana do żadnego użytkownika ani nie jest przyznawana żadnej innej roli.

Podczas tworzenia ról, które będą służyć jako właściciele obiektów w systemie, zalecane jest utworzenie hierarchii ról niestandardowych, z najwyższą rolą niestandardową

przypisaną do roli systemowej SYSADMIN. W ten sposób administratorzy systemu będą mogli zarządzać wszystkimi hurtowniami i wszystkimi bazami danych, zachowując zarządzanie użytkownikami i rolami ograniczonymi do użytkowników, którym przyznano rolę SECURITYADMIN lub ACCOUNTADMIN i odwrotnie.

Uprawnienia są zarządzane za pomocą SQL (polecenia GRANT, REVOKE) lub interfejsu webowego. Interfejs webowy nie obsługuje wszystkich zadań związanych z użytkownikami, ale zapewnia wygodny kreator do tworzenia użytkowników i wykonywania najczęstszych czynności, takich jak resetowanie hasła użytkownika.

1.3. DDL, DCL, DML, DQL w porównaniu z Teradata

Data Definition Language (DDL)

Język DDL składa się z poleceń SQL, których można użyć do zdefiniowania schematu bazy danych. DDL umożliwia dodawanie, modyfikowanie i usuwanie struktur logicznych, zawierających dane lub umożliwiających użytkownikom dostęp do danych, na przykład bazy danych, tabele, klucze, widoki.

W tabeli 1.3.1 przedstawione zostały polecenia DDL dostępne w systemach Snowflake i Teradata.

Tabela 1.3.1. Polecenia języka DDL

Polecenie	Opis	Snowflake	Teradata
CREATE	Tworzy nowy obiekt określonego typu.	✓	✓
ALTER	Modyfikuje metadane poziomu konta lub obiektu bazy danych lub parametry sesji.	✓	✓
DROP	Usuwa określony obiekt z systemu.	✓	✓
SHOW	Wyświetla listę istniejących obiektów dla określonego typu obiektu. Dane wyjściowe zawierają metadane dla obiektów.	✓	✓
DESCRIBE	Opisuje szczegóły dla określonego obiektu.	✓	
COMMENT	Dodaje komentarz lub zastępuje istniejący komentarz do istniejącego obiektu.	✓	

USE	Określa rolę, magazyn, bazę danych lub schemat do użycia w bieżącej sesji.	✓	
HELP	Wyświetla atrybuty określonego obiektu. W przypadku ONLINE wyświetlana jest pomoc dla dowolnej instrukcji SQL lub polecenia narzędzia klienta.		✓
RENAME	Zmienia nazwę bieżącego istniejącego obiektu.		✓
MODIFY	Zmienia parametry określonego obiektu.		✓
REPLACE	Tworzy lub zastępuje nowy obiekt określonego typu.		✓
FLUSH	Opróżnia jedną, kilka lub wszystkie pamięci podręczne DBQL lub pamięci podręczne zarządzania obciążeniem na dysk.		✓
DELETE	Usuwa wszystkie tabele danych, widoki, wyzwalacze, procedury SQL, makra i pliki zainstalowane przez użytkownika (UIF) z bazy danych lub użytkownika.		✓
SET	Ustawia bieżący obiekt dla sesji.		✓
BEGIN	Rozpoczyna jawną współbieżną operację izolowanego obciążenia na tabeli izolowanej obciążenia.		✓
END	Kończy jawną współbieżną operację izolowanego obciążenia dla określonej wartości grupy obciążenia.		✓

Data Manipulation Language (DML)

DML pozwala na dodawanie, modyfikowanie i usuwanie danych. Tabela 1.3.2 opisuje dostępne polecenia DML w systemach Snowflake i Teradata.

Tabela 1.3.2. Polecenia języka DML

Polecenie	Opis	Snowflake	Teradata
INSERT	Aktualizuje tabelę, wstawiając jeden lub więcej wierszy do tabeli. Wartości wstawione do każdej kolumny w tabeli mogą być jawnie określone lub wyniki zapytania.	✓	✓
UPDATE	Aktualizuje określone wiersze w tabeli docelowej o nowe wartości.	✓	✓
DELETE	Usuwa dane z tabeli za pomocą opcjonalnej klauzuli WHERE i / lub dodatkowych tabel. To polecenie nie usuwa historii ładowania plików zewnętrznych.	✓	✓
INSERT (multi-table)	Aktualizuje wiele tabel, wstawiając do tabel jeden lub więcej wierszy z wartościami kolumn (z zapytania). Obsługuje wstawki bezwarunkowe i warunkowe.	✓	
MERGE	Wstawia, aktualizuje i usuwa wartości w tabeli na podstawie wartości w drugiej tabeli lub podzapytaniu. Może to być przydatne, jeśli druga tabela jest dziennikiem zmian zawierającym nowe wiersze, zmodyfikowane wiersze w tabeli.	✓	
TRUNCATE	Usuwa wszystkie wiersze z tabeli, ale pozostawia tabelę nietkniętą (w tym wszystkie przywileje i ograniczenia w tabeli). Usuwa również metadane obciążenia dla tabeli.	✓	
COPY INTO	Ładuje / rozładowuje dane z / do plików pomostowych do / z istniejącej tabeli. Pliki muszą być już umieszczone w jednej z następujących lokalizacji	✓	

PUT	Przesyła pliki danych z lokalnego katalogu / folderu na komputerze klienckim do jednego z etapów Snowflake.	✓	
GET	Pobiera pliki danych z jednego z etapów Snowflake do lokalnego katalogu / folderu na komputerze klienta.	✓	
LIST	Zwraca listę plików, które zostały zaaranżowane (tj. Przesłane z lokalnego systemu plików lub rozładowane z tabeli) w jednym z etapów Snowflake.	✓	
REMOVE	Usuwa pliki, które zostały przeniesione (tj. Przesłane z lokalnego systemu plików lub wyładowane z tabeli) w jednym z etapów wewnętrznych Snowflake.	✓	
EXECUTE	Wykonuje przygotowaną instrukcję o nazwie nazwa_wyrażenia. Wartości parametrów są zdefiniowane w klauzuli USING.		✓

Data Query Language (DQL)

Główną instrukcją DQL jest SELECT, która pobiera potrzebne dane. SHOW pobiera informacje o metadanych. W języku DQL Snowflake i Teradata zawierają następujące polecenia (Tabela 1.3.3.):

Tabela 1.3.3. Polecenie SELECT języka DQL

Polecenie	**Opis**	**Snowflake**	**Teradata**
WITH	Klauzula WITH jest klauzulą opcjonalną, która poprzedza klauzulę SELECT w instrukcji. Do aliasów klauzuli WITH można odwoływać się w klauzuli FROM. Klauzula WITH definiuje jedno lub więcej podzapytań lub wyrażeń.	✓	✓
SELECT	SELECT może być używany zarówno w instrukcji, jak i w klauzuli w instrukcji SELECT. Jako instrukcja instrukcja SELECT jest najczęściej wykonywaną instrukcją SQL; wysyła zapytanie do bazy danych i pobiera zestaw wierszy. Jako klauzula	✓	✓

	SELECT definiuje zestaw kolumn zwracanych przez zapytanie.		
FROM	Określa tabele, widoki lub funkcje tabeli, które mają być używane w instrukcji SELECT.	✓	✓
AT | BEFORE	Klauzula AT lub BEFORE jest używana do podróży w czasie Snowflake. W zapytaniu jest ona określona w klauzuli FROM bezpośrednio po nazwie tabeli i określa punkt w przeszłości, z którego żądane są dane historyczne dla obiektu. Słowo kluczowe AT określa, że żądanie zawiera wszelkie zmiany wprowadzone przez instrukcję lub transakcję ze znacznikiem czasu równym określonemu parametrowi. Słowo kluczowe BEFORE określa, że żądanie odnosi się do punktu bezpośrednio poprzedzającego określony parametr.	✓	
JOIN	Klauzula JOIN jest klauzulą klauzuli FROM. Operacja JOIN łączy wiersze z dwóch tabel (lub innych źródeł, takich jak widoki lub funkcje tabel) w celu utworzenia nowego połączonego wiersza, który może zostać użyty w zapytaniu.	✓	✓
UNPIVOT	Obraca tabelę, przekształcając kolumny w wierszu. UNPIVOT to operator relacyjny, który akceptuje dwie kolumny (z tabeli lub podzapytania) wraz z listą kolumn i generuje wiersz dla każdej kolumny określonej na liście.	✓	
PIVOT	Obraca tabelę, zmieniając unikalne wartości z jednej kolumny w wyrażeniu wejściowym na wiele kolumn i agregując wyniki tam, gdzie jest to wymagane, na wszystkich pozostałych wartościach kolumn. W zapytaniu jest określona w klauzuli FROM po nazwie tabeli lub podzapytaniu.	✓	
VALUES	W instrukcji SELECT klauzula VALUES klauzuli FROM umożliwia określenie zestawu stałych, które mają zostać użyte do utworzenia skończonego zestawu wierszy.	✓	

SAMPLE	Zwraca podzbiór wierszy próbkowanych losowo z określonej tabeli.	✓	✓
WHERE	Klauzula WHERE filtruje wynik klauzuli FROM.	✓	✓
GROUP BY	Grupuje wiersze z tymi samymi wyrażeniami grupowymi i oblicza funkcje agregujące dla grupy wynikowej. Wyrażenie GROUP BY może być nazwą kolumny, liczbą odwołującą się do pozycji na liście SELECT lub ogólnym wyrażeniem.	✓	✓
HAVING	Filtuje wiersze produkowane przez GROUP BY, które nie spełniają predykatu.	✓	✓
ORDER BY	Określa kolejność wierszy tabeli wyników z listy SELECT.	✓	✓
LIMIT	Ogranicza maksymalną liczbę wierszy zwracanych przez instrukcję lub podzapytanie. Obsługiwane są zarówno LIMIT (składnia Postgres), jak i FETCH (składnia ANSI) i dają ten sam wynik.	✓	
QUALIFY	Klauzula warunkowa w instrukcji SELECT, która filtruje wyniki poprzednio obliczonej uporządkowanej funkcji analitycznej zgodnie z warunkami wyszukiwania określonymi przez użytkownika. Używając QUALIFY, możesz wybrać konkretne rekordy z tabeli.		✓
DISTINCT	DISTINCT można dodać w SELECT, aby zwrócić unikatowe wartości, eliminując powtarzające się wartości.		✓

Data Control Language (DCL)

DCL służy do przyznawania i odwoływania uprawnień do baz danych i ich zawartości. DCL dotyczy bezpieczeństwa. W języku DCL Snowflake i Teradata zawierają następujące polecenia (Tabela 1.3.4.)

Tabela 1.3.4. Polecenia języka DCL

Polecenie	Opis	Snowflake	Teradata
REVOKE	Usuwa uprawnienie z roli lub udziału.	✓	✓

GRANT	Przydziela rolę lub uprawnienie dostępu do obiektu zabezpieczonego roli.	✓	✓

1.4. Integracja z innymi narzędziami zewnętrznymi

1.4.1. Narzędzia ETL

System Snowflake pozwala na integrację z narzędziami ETL, które służą do:

- pozyskanie danych ze źródeł zewnętrznych (na przykład pobieranie danych z Amazon S3),
- znalezienie, wyczyszczenie i poprawienie błędów danych,
- redukcja do pojedynczych metryk / wymiarów / katalogów,
- agregacja do wymaganych szczegółów,
- załadowanie danych do systemu docelowego / pamięci masowej (do bazy danych).

Snowflake eliminuje potrzebę długotrwałego i czasami pracochłonnego wykonywania procesów - pobrania, transformacji, ładowania, udostępniając dane wewnętrznym i zewnętrznym partnerom poprzez udostępnianie danych i hurtowni danych Snowflake.

1.4.2. Narzędzia do wizualizacji

Snowflake umożliwia również połączenie stworzonej w systemie hurtowni z narzędziami do wizualizacji danych, np. Tableau.

Snowflake nie wspiera interaktywnego podejścia do analizy i dlatego nie zawiera zbyt zaawansowanych narzędzi wewnętrznych lub funkcji służących do wizualizacji. Do integracji Snowflake z Tableau należy znać następujące informacje:

1) Nazwa serwera.
2) Metoda Uwierzytelnienia: nazwa użytkownika i hasło, SAML IdP lub Oauth.
3) Poświadczenia logowania zależą od wybranej metody uwierzytelniania i mogą obejmować następujące elementy: nazwa użytkownika i hasło, URL serwera SAML IdP, nazwa użytkownika.
4) (Opcjonalnie) Początkowa instrukcja SQL uruchamiana za każdym razem, gdy zostaje nawiązane połączenie.

1.4.3. Snow SQL (CLI Client)

SnowSQL jest klientem linii poleceń nowej generacji do łączenia się ze Snowflake w celu wykonywania zapytań SQL i wykonywania wszystkich operacji DDL i DML, w tym ładowania danych do tabel bazy danych i wyładowywania danych. Oprogramowanie dostarczone jest formie do pobrania w specjalnych wersjach dostosowanych dla następujących platform: Microsoft Windows, Mac OS, Linux.

Do ustawienie polaczenia z instancją Snowflake zalecane jest ustawienie danych uwierzytelniających połączenia w sekcji ‘connection’ w pliku konfiguracyjnym ‘./snowsql/config’. Poniższa grafika (Rysunek 1.4.1) przedstawia przykładowe dane konieczne do wprowadzenia w pliku konfiguracyjnym klienta SnowSQL. Hasło podawane jest jako jawny ciąg znaków.

```
accountname = xy12345.eu-central-1
username = jsmith
password = xxxxxxxxxxxxxxxxxxx
dbname = mydb
schemaname = public
warehousename = mywh
```

Rysunek 1.4.1. Przykładowe dane konfiguracyjne połączenia z pliku ‘config’

1.5. Licencje a model kosztowy

Koszt usługi Snowflake zależy od licencji systemu oraz faktycznego wykorzystania, które zmienia się w zależności od indywidualnej aplikacji. Większość klientów zdobywa doświadczenie dzięki Snowflake On Demand. Umożliwia to opracowanie i przetestowanie obciążenia aplikacji oraz zapewnienie rzeczywistego doświadczenia w celu oszacowania kosztu miesięcznego. Po zrozumieniu obciążenia aplikacji można dokonać zakupu odpowiedniej wielkości.

W tabeli 1.5.1. zaprezentowano informacje o kosztach związanych z prowadzeniem konta w systemie Snowflake z uwzględnieniem lokalizacji maszyny, na której postawiony jest system oraz wyszczególnieniem wykorzystywanych usług. W tabeli 1.5.2 przedstawiono koszty związane z wykorzystaniem pamięci.

Tabela 1.5.1. Ceny kredytów. System Snowflake (ceny w USD)

Licencja	Dublin	Frankfurt	Usługi
Standard	2,5	2,7	Kompletna hurtownia danych SQL
			Współdzielenie danych
			Wsparcie w godzinach pracy od poniedziałku do piątku
			Jednodniowa historia
			Zawsze włączone szyfrowanie dla danych podczas transmisji oraz spoczynku
			Dedykowane klientowi wirtualne hurtownie
Premier	2,8	3	Wsparcie 24 x 365

(Standard +)			Szybszy czas reakcji
			SLA z refundacją za opóźnienia
Enterprise (Premier +)	3,7	4	Hurtownia Multi-Cluster
			do 90 dni utrzymanej historii
			Uwierzytelnianie federacyjne
			Coroczna aktualizacja szyfrowania dla zaszyfrowanych danych
			Dziennik kontroli (H2 2017)
			Replikacja między regionami (H2 2017)
			AWS PrivateLink dostępne za dodatkową opłatę
Enterprise for Sensitive Data Enterprise +)	5	5,4	Wsparcie HIPAA
			Zgodność z PCI
			Szyfrowanie danych wszędzie
			Ulepszona polityka bezpieczeństwa
			Klucze szyfrowania zarządzane przez klienta
			Zawiera opcję dla AWS PrivateLink
Virtual Private Snowflake (VPS)	cena do indywidualnego ustalenia		Wirtualne serwery dedykowane dla klienta zawsze, kiedy klucz szyfrowania znajduje się w pamięci
			Dedykowany dla klienta magazyn metadanych
			Dodatkowa widoczność operacyjna

Tabela 1.5.2. Koszty wykorzystania pamięci w systemie Snowflake (ceny w USD)

STORAGE	Dublin TB / Month	Frankfurt TB / Month	
On-Demand Storage	40	45	Miesięczna płatność bieżąca
Capacity Storage	23	24,5	Płatność z góry

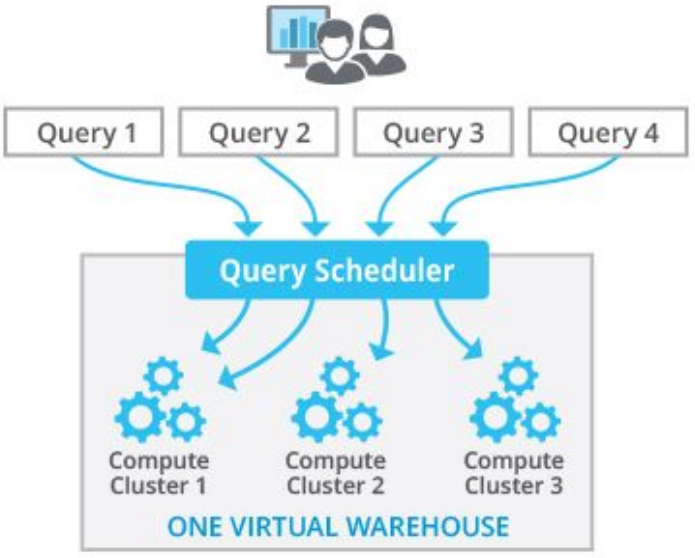
Wycena Snowflake opiera się na rzeczywistym wykorzystaniu Storage i Virtual Warehouses (Compute) architektury systemu, bez opłat za koszty związane z warstwą Services (szczegółowy opis architektury znajduje się w punkcie 1.1).

Storage. Wszyscy klienci są obciążeni miesięczną opłatą za dane przechowywane w Snowflake. Koszt przechowywania jest mierzony przy użyciu średniej ilości przechowywanych danych miesięcznych ze wszystkich danych klienta przechowywanych w Snowflake, po kompresji.

Virtual Warehouse. Klienci płacą za Virtual Warehouse za pomocą Kredytów Snowflake tylko wtedy, gdy jest on uruchomiony. Gdy Virtual Warehouse nie działa, czyli jest ustawiony w tryb uśpienia, nie zużywa żadnych kredytów Snowflake. Kredyty są zużywane w tempie zależnym od wielkości działającej hurtowni.

1.6. Skalowanie maszyn pod kątem zwiększenia mocy obliczeniowych i związane z tym koszty

System Snowflake pozwala na natychmiastowe zmienianie rozmiaru oraz zatrzymywanie i wznawianie pracy wirtualnych hurtowni danych. Hurtownia reprezentuje liczbę fizycznych węzłów, które użytkownik może wykorzystać do wykonywania zadań (np. zapytań). Snowflake wspiera alokację (statyczną i dynamiczną) większej ilości zasobów, poprzez określenie dodatkowych klastrów dla hurtowni. Dla licencji typu Enterprise oraz Enterprise for Sensitive Data możliwe jest ręczne dostosowanie ilości klastrów, natomiast dla niższych edycji skalowanie oparte jest na zmianie rozmiaru hurtowni.



Rysunek 1.6.1. Wizualizacja działania wieloklastrowej hurtowni danych

Wszystkie klastry obliczeniowe w hurtowni są tego samego rozmiaru. Użytkownik Snowflake w edycji typu Enterprise lub Enterprise for Sensitive Data może wybrać jeden z dwóch trybów działania dla swojej wieloklastrowej hurtowni:

1. Maximized - przy starcie hurtowni, Snowflake zawsze startuje wszystkie klastry, w celu zapewnienia, że maksymalna liczba zasobów jest dostępna w czasie działania hurtowni
2. Auto Scaling - Snowflake startuje i zatrzymuje klastry według zapotrzebowania, w celu dynamicznego zarządzania obciążenia w hurtowni.

Zasoby skalują się liniowo, a rozmiar hurtowni można zmieniać nawet podczas jej działania. W przypadku zmiany rozmiaru hurtowni wieloklastrowej, nowy rozmiar jest zastosowany do wszystkich klastrów hurtowni, łącznie z aktywnymi.Tabela 1.6.1 przedstawia zestawienie dostępnych rozmiarów hurtowni oraz kosztów związanych z jej użytkowaniem.

Tabela 1.6.2. Koszty wykorzystania pamięci w systemie Snowflake (ceny w USD)

Rozmiar hurtowni	**Serwery / Klastry**	**Kredyty / Godzina**
X-Small	1	1
Small	2	2
Medium	4	4
Large	8	8
X-Large	16	16
2X-Large	32	32
3X-Large	64	64
4X-Large	128	128

Liczba serwerów w każdym klastrze jest określona przez rozmiar hurtowni:

- całkowita liczba serwerów: (rozmiar hurtowni) * (maksymalna liczba klastrów)
  = maksymalna liczba kredytów wykorzystanych w hurtowni w ciągu pełnej godziny użytkowania.

1.7. Zalety i wady systemu Snowflake

Tablica 1.7.1. Wady i zalety systemu Snowflake

Zalety	Wady
Możliwość dynamicznego połączenia zestawu zasobów, w celu obsłużenia wielu różnych scenariuszy użytkowania, z właściwą równowagą IO, pamięci, procesora itp. (Workloads z różnymi zapytaniami i wzorcami dostępu w pojedynczej obsłudze)	Snowflake musi ulepszać swoje części przestrzenne, ponieważ nie ma zbyt wiele funkcji pod względem zapytań geoprzestrzennych.
Obsługa wszystkich danych różnych typów w jednym systemie.	Wsparcie Snowflake czasami może być trudne do osiągnięcia.
Wsparcie dla wszystkich przypadków użycia z dynamiczną elastycznością. Zasoby można dodawać, odrzucać i zmieniać w każdej chwili, nie zakłócając przetwarzania.	Wydajność może być problemem, jeśli w systemie jest jednocześnie wielu użytkowników.
Łatwość w eksploatacji dzięki samozarządzającej usłudze sprzętu i oprogramowania oraz automatycznej adaptacji.	Uboga dokumentacja.
Automatyzacja takich działań jak: ciągła ochrona danych, kopiowanie do klonowania, dystrybucja danych, ładowanie danych, dynamiczna optymalizacja zapytań, skalowalne obliczenie,	Bardzo ograniczona liczba zakładek – zapisane zapytania, co wymaga przechowywania kodu w innym miejscu i ponownego wykorzystywania istniejących zapytań.
Nielimitowana ilość pamięci i moc obliczeniowa.	Mało rozwinięte funkcje tworzenia wykresów.
Snowflake to baza danych masowego równoległego przetwarzania (MPP), w pełni relacyjna, zgodna z ACID i przetwarza standardowy SQL bez tłumaczenia i symulacji.	Narzędzia do raportowania nie są zbyt rozbudowane , co stwarza potrzebę połączenia z innymi narzędziami, takimi jak Tableau.

Obsługa generowanych maszynowo i dostarczanych w półstrukturalnych formatach danych, takie jak JSON, AVRO, XML itd.	
Możliwość odpytywania danych za pomocą rozszerzeń do SQL, dzięki czemu zapytania relacyjne mogą łączyć dostęp do danych strukturalnych i półstrukturalnych w jednym zapytaniu.	
Konfiguracja obliczeniowa jest określana niezależnie od ilości danych w systemie.	
Możliwość skalowania, w celu obsługi większej liczby użytkowników i nakładów pracy bez wpływu na wydajność.	
Płatny jako usługa oparta na wykorzystaniu.	

2. Profilowanie danych w systemie Snowflake

2.1. Cel profilowania

Profilowanie danych w bazie pozwala na szybszą analizę zawartości kolumn oraz lepsze zrozumienie danych, bez konieczności przeglądania każdej wartości. Ułatwia również przeszukiwanie kolumn w celu wyszukania wartości odstających.

Przeprowadzenie profilowania w systemie Snowflake ma również na celu przygotowanie podstawy do porównania efektywności przetwarzania tych samych danych na różnych platformach oraz zestawienie możliwości i ograniczeń poszczególnych systemów.

2.2. Dane wejściowe

Zanonimizowane dane, które służą jako podstawa profilowania udostępnione zostały na potrzeby projektu na platformie AWS. Są one zapisane w postaci plików .dat.

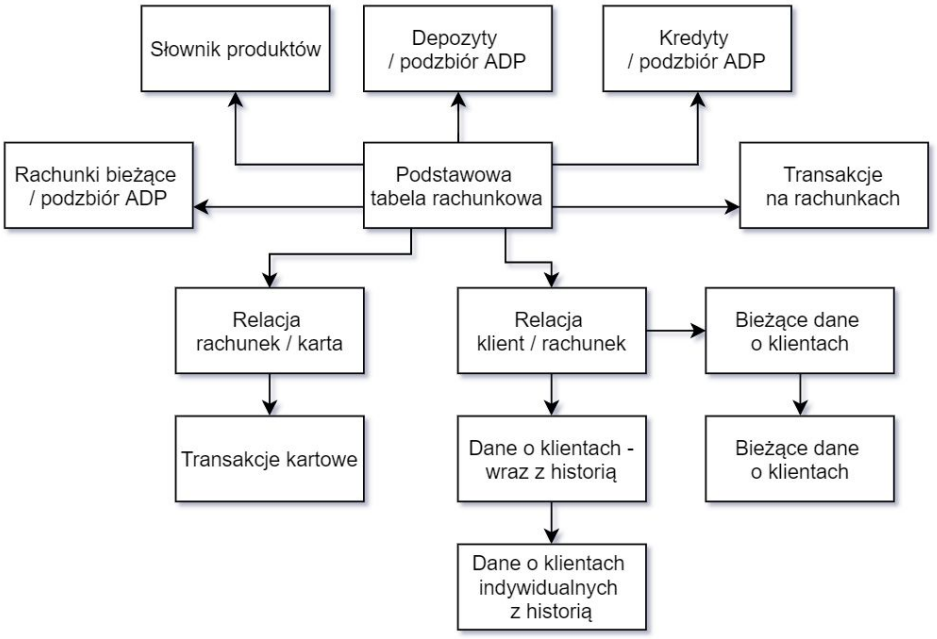
2.3. Utworzenie struktury DWH na Snowflake

Hurtownia danych w systemie Snowflake została utworzona na podstawie dostarczonego schematu istniejącej już hurtowni w obecnie wykorzystywanym systemie.

Głównym zadaniem było przepisanie otrzymanych poleceń tworzących strukturę w systemie Teradata na odpowiadające im polecenia akceptowane przez system Snowflake. Wiązało się to z usunięciem wielu ograniczeń oraz opcji.

Jedynym ograniczeniem integralnościowym, w pełni wspieranym i kontrolowanym przez Snowflake jest NOT NULL. Inne można wykorzystywać, ale nie są one egzekwowane przez to narzędzie. System nie wspiera indeksów ani partycjonowania tabel - wszelkie operacje, mające na celu przyspieszenie wykonywanych operacji na danych są zaimplementowane wewnętrznie w systemie. Zakładane jest również, że dane wgrywane do systemu są już odpowiednio oczyszczone i przetransformowane.

Przed wczytaniem danych do tabel niezbędna była także zmiana typów niektórych kolumn, a także utworzenie nowego typu danych - Snowflake nie obsługuje plików z rozszerzeniem .dat. Aby z takich korzystać, należy zadeklarować własny typ ze wskazaniem odpowiedniego separatora oraz kodowania znaków.



Rysunek 2.3.1. Uproszczony schemat hurtowni danych

2.4. Integracja z S3 i załadowanie danych

W celu załadowania danych z AWS S3, stworzono scenę do wystawienia danych. Połączenie tworzone jest za pomocą parametrów: adresu url, wskazującego na dane znajdujące się w S3 oraz niezbędnych danych uwierzytelniających: AWS_KEY_ID oraz AWS_SECRET_KEY

Po utworzeniu połączenia dane zostały załadowane do wcześniej utworzonych tabel przy użyciu polecenia COPY INTO ze wskazaniem na definicję formatu pliku źródłowego.

2.5. Opis procedury profilowania

Profilowanie wykonano za pomocą przygotowanego w tym celu skryptu w języku python, załączonym w postaci pliku jupyter notebook. Program umożliwia połączenie się z hurtownią założoną w systemie Snowflake oraz obliczenie parametrów profilowania danych.

Program został napisany w oparciu o wykorzystanie funkcjonalności dostępnych w pakiecie snowflake.connector (https://github.com/snowflakedb/snowflake-connector-python). Połączenie ze Snowflake tworzone jest w oparciu o niezbędne dane uwierzytelniająca:

- nazwę konta w Snowflake
- nazwę użytkownika
- hasło

Dane uwierzytelniające program odczytuje z zewnętrznego pliku tekstowego.

Program udostępnia dwie główne funkcje: pierwsza pozwala na profilowanie danych, pochodzących ze wskazanej tabeli z uwzględnieniem wyłącznie podanych kolumn, natomiast druga umożliwia przeprowadzenie profilowania dla całej bazy lub hurtowni:

- dataProfilingforTable(table, columns, save, database) - w przypadku niepodania żadnej kolumny, funkcja wykonuje profilowanie dla całej tabeli,
- dataProfiling(database, tables, columns, save) - stanowi rozszerzenie funkcji dataProfilingforTable, pozwala na podanie listy tabel do profilowania oraz, w przypadku nie podania żadnej wartości w miejsce listy tabel i kolumn, na przeprowadzenie profilowania dla wszystkich danych znajdujących się we wskazanej bazie.

  Przykładowe wywołania funkcji:

  - dataProfiling(‘db’) - profilowanie dla całej bazy ‘db’ z zapisywaniem wyników do tabeli,
  - dataProfiling(‘db’, [‘table1’], [‘’]) - profilowanie dla wszystkich kolumn tabeli ‘table1’ z bazy ‘db’ z zapisywaniem wyników do tabeli wynikowej,
  - dataProfiling(‘db’, [‘table1’,’table2’], [‘col1,col2’,’col3’], save = False) - profilowanie dla column ‘col1’ i ‘col2’ z tabeli ‘table1’ oraz dla kolumny ‘col3’ z tabeli ‘table2’ z bazy ‘db’ bez zapisywania wyników.

Dane w tabelach są analizowane przez zebranie statystyk, których użycie jest zależne od typu danych kolumny.

		Typ danych w kolumnie		
Statystyka	Opis	Tekstowy	Liczbowy	Inny
Count	Całkowita liczba wierszy	✓	✓	✓
Nulls	Liczba wierszy o wartości NULL	✓	✓	✓
Empties	Liczba wierszy o długości 0	✓		
Blanks	Liczba wierszy o długości 0 lub składających się wyłącznie z białych znaków	✓		
Distinct	Liczba unikalnych wartości	✓	✓	✓
Average length	Średnia długość	✓		
Minimal length	Najmniejsza długość	✓		
Maximal length	Największa długość	✓		
Minimum	Minimalna wartość		✓	
Maximum	Maksymalna wartość		✓	
Sum	Suma wartości (z pominięciem wartości NULL)		✓	
Standard deviation	Standardowe odchylenie obliczone na podstawie wartości (z pominięciem wartości NULL)		✓	
Mean	Średnia z wartości (z pominięciem wartości NULL)		✓	

Profilowanie danych dla bazy danych oparto na wyznaczeniu statystyk dla wszystkich kolumn z tabel znajdujących się w bazie. Dla kolumny sprawdzany jest jej typ, a następnie tworzone są zapytania SQL, pozwalające na wyznaczenie odpowiednich dla typu danych

statystyk. Zapytania wysyłane są do Snowflake poprzez nawiązane połączenie, a otrzymane odpowiedzi służą następnie do stworzenia polecenia wstawiania (Insert). Wyniki profilowania dla każdej kolumny zapisywane są w tabeli DATA_PROFILING w Snowflake, pozwala to na późniejsze prezentowanie wyników profilowania i przeprowadzenie analizy porównawczej.

2.6. Prezentacja wyników profilowania poprzez integrację z Tableau

Integracja systemu Snowflake z Tableau jest procesem bardzo prostym w obsłudze. Wystarczy podać adres serwera oraz dane uwierzytelniające, a wszystkie tabele zostaną pobrane do programu.

Na rysunkach 2.6.1. oraz 2.6.2. zostały przedstawione przykładowe tablice z wynikami profilowania. Na pierwszej z nich została umieszczona tabela z wartościami obliczonych statystyk dla wybranych tabel i kolumn. Na drugiej natomiast znajdują się dwa wykresy: pierwszy, znany jako wykres tornado, przedstawiający liczbę rekordów oraz liczbę unikalnych wartości w danej tabeli. Drugi wykres - kropkowy - wizualizuję liczbę wartości pustych dla danych tabel.

TABLE_NAME: PRODUCT_BZWBK

COULUMN_NAME: (All)

Profiling results

TABLE_NAME	COULUMN_NAME	AVGLEN	BLANKS	COUNT	DISTINCTS	EMPTIES	MAXIMUM	MAXLEN	MEAN	MINIMUM	MINLEN	NULLS	STANDARD..	SUM
PRODUCT_BZ..	START_DTE			2 837	547							0		
	SRCE_SYS			2 837	3		5 003		5 002	5 001		0	1	14 191 225
	SRCE_PROD_ID	3	0	2 837	439	0		3			3	0		
	SRCE_PROD_CDE	20	0	2 837	2 837	0		20			20	0		
	PURPOSE_CDE	5	0	2 837	369	0		5			5	1 304		
	PROD_SUMM_DESCR	12	0	2 837	9	0		18			7	0		
	PROD_STA_CDE	4	0	2 837	2	0		4			4	0		
	PROD_NAME	33	0	2 837	1 191	0		98			7	0		
	PROD_GRP_DESCR	16	0	2 837	23	0		32			10	0		
	PROD_DTL_GRP_DESCR	21	0	2 837	155	0		49			3	0		
	PROD_DTL_DESCR	11	0	2 837	15	0		41			8	0		
	PROD_CDE			2 837	2 837		17 287		12 474	8 056		0	3 025	35 388 478
	PROD_CAT_DESCR	50	0	2 837	3	0		50			50	0		
	MATURITY_GRP_CDE		0	2 837	0	0						2 837		
	LOAD_TIME			2 837	1							0		
	LOAD_LAST_ACTION	1	0	2 837	1	0		1			1	0		
	LOAD_DTE			2 837	1							0		
	ISO_CRNCY_CDE	3	0	2 837	19	0		3			3	25		
	INTEREST_PLAN_NO	6	0	2 837	171	0		6			6	0		
	END_DTE			2 837	0							2 837		
	CHARGE_PLAN_NO	6	0	2 837	295	0		6			6	1 855		

Rysunek 2.6.1. Przykładowa tabela z wynikami profilowania

Rysunek 2.6.2. Przykładowe wykresy z wynikami profilowania

3. Analiza systemu pod kątem wydajności

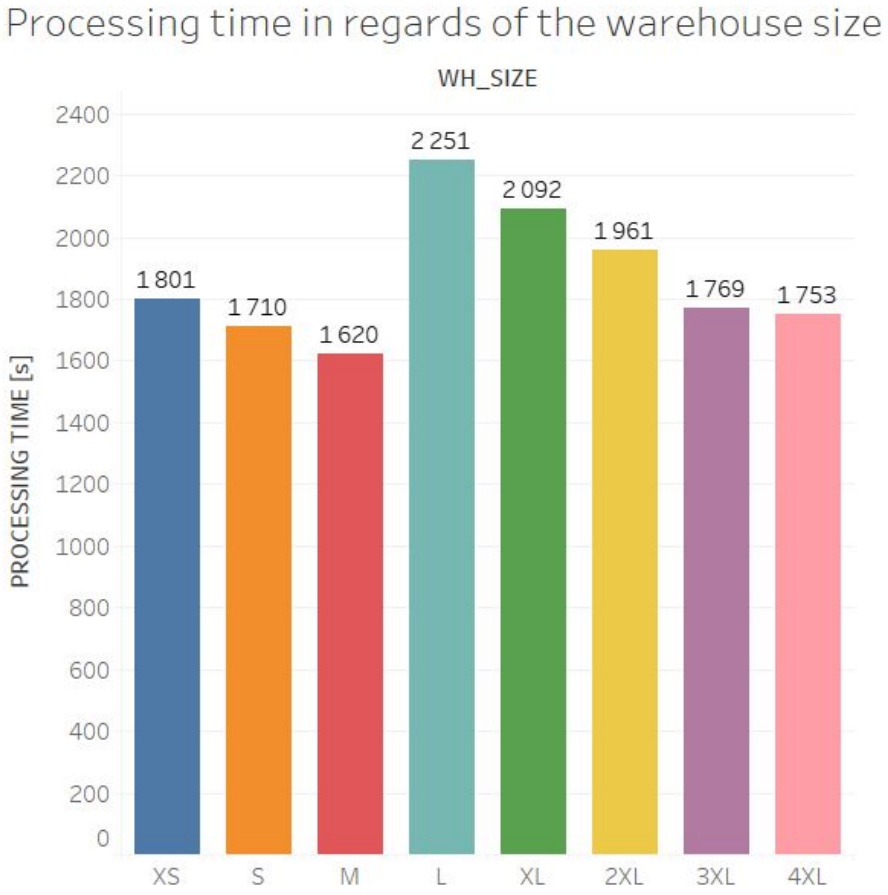
3.1. Opis podejścia do testów wydajnościowych dla hurtowni na wszystkich rozmiarach środowiska

W celu porównania wydajności profilowania na różnych rozmiarach środowiska, przeprowadzono profilowanie dla całej bazy danych dla każdego z dostępnych rozmiarów, bez zapisu wyników profilowania. Czas przetwarzania dla poszczególnych wielkości środowiska zapisano w stworzonej na potrzeby zestawienia w tabeli PROFILING_PER_SIZE w Snowflake.

3.2. Integracja z Tableau - wyniki

Wyniki testów wydajnościowych zostały zapisane w specjalnie utworzonej w tym celu tabeli. Tableau pozwala na bardzo prostą i szybką wizualizację takich danych.

Na wykresie widoczny jest czas przetwarzania zapytania w zależności od rozmiaru wykorzystywanej hurtowni danych.



3.3. Wnioski

Według dokumentacji systemu Snowflake, rozmiar wykorzystywanej hurtowni danych jest odwrotnie proporcjonalny do czasu przetwarzania danego zapytania. Oznacza to, że czas przetwarzania na każdym kolejnym rozmiarze powinien być dwukrotnie krótszy od czasu przetwarzania na hurtowni o rozmiarze mniejszym, co wiązałoby się z tym, że przetwarzanie danego zapytania generuje stały koszt (w postaci wykorzystanych kredytów), niezależnie od mocy obliczeniowej. Nie jest to jednak zgodne z otrzymanymi wynikami testów wydajnościowych.

Podejrzewaną przyczyną jest zbyt mały rozmiar danych - możliwe, że każda operacja obciążona jest pewnym minimalnym czasem, który jest potrzebny do przesłania zapytania i pobranych danych między chmurą, w której przechowywane są dane a klientem. Kolejną z możliwych przyczyn jest fakt, że stosowane zapytania nie są zapytaniami złożonymi. Wszystkie testy przeprowadzono na instancjach testowych Snowflake na licencji trial, co również mogło mieć wpływ na uzyskane wyniki.

4. Słownik pojęć

Amazon S3 - internetowy nośnik danych firmy Amazon. Ma prosty w obsłudze interfejs WWW, który umożliwia dostęp do przechowywanych danych i zarządzanie nimi.

Shared-disk - rozproszona architektura obliczeniowa, w której wszystkie dyski są dostępne ze wszystkich węzłów klastra.

Shared-nothing - rozproszona architektura obliczeniowa, w której każde żądanie aktualizacji danych jest spełnione przez pojedynczy węzeł. Węzły nie współdzielą pamięci.

DAC (*Dyskretna kontrola dostępu)* - Każdy obiekt ma właściciela, który z kolei może udzielić dostępu do tego obiektu.

RBAC (Kontrola dostępu oparta na rolach) - Uprawnienia dostępu są przypisane do ról, które z kolei są przypisywane użytkownikom.

ETL (Extract, Transform and Load) - narzędzie wspomagające proces pozyskania danych dla baz danych, szczególnie dla hurtowni danych. Proces składa się z trzech etapów:

1) Pobranie (Extract): dane są pobierane z jednorodnych lub heterogenicznych zbiorów danych,
2) Przekształcenie (Transform): dane są przekształcane w odpowiednie formaty do przechowywania,
3) Ładowanie (Load): dane są ostatecznie ładowane do docelowej bazy danych lub hurtowni danych.

DQL - Data Query Language - instrukcje, za pomocą których możliwe jest pozyskiwanie danych z bazy. Najważniejszym poleceniem jest SELECT.

DML - Data Manipulation Language - instrukcje manipulacji danymi. Można do nich zaliczyć polecenia takie jak INSERT, UPDATE, DELETE. Najważniejszą cechą tych instrukcji jest fakt, że za ich pomocą możliwe jest manipulowanie danymi w obiektach, np. w tablicach.

DDL - Data Definition Language - instrukcje definiujące. Do tej grupy zaliczane są polecenia takie jak CREATE, ALTER, DROP. Za pomocą instrukcji DDL nie manipulujemy bezpośrednio danymi, a ich strukturą. Możliwe jest definiowanie atrybutów, ich modyfikacja pod kątem formatu lub usuwanie obiektów.

DCL - Data Control Language - Instrukcje sterujące uprawnieniami w bazie danych / serwerze. Za ich pomocą możliwe jest nadawanie uprawnień do obiektów, przypisywanie ról, zmiana haseł itp. Najważniejsze polecenia to GRANT, DENY, REVOKE.