

Snowflake 필기(11~15)

CHAPTER 11. Resource Monitors'

우린 Resource Monitor를 통해 비용을 관리하고, WH 사용으로 인한 크레딧도 관리가능 또한 WH의 크레딧 사용 한도를 둘 수도 있음 --->

Snowflake UI에서 확인가능하며, AccountAdmin Role 통해서만 확인할 수 있음

Resource는 오직 AccountAdmin Role 일때만 만들 수 있지만, Monitor & Modify 권한 이 있는 역할은 또한 수정이 가능하다

※materialized View는 리소스 모니터에 계산이 안된다고 함.... ?

리소스 모니터의 할당

1. 계정 수준에서 단일 모니터를 설정하여 계정의 모든 WH에 대한 크레딧 사용을 제어할 수 있음.
2. 하나 이상의 WH에 모니터를 할당하여 각 창고의 크레딧 사용량을 제어할 수 있음.
3. **계정 수준 아래의 리소스 모니터 하나에만 WH를 할당도 가능**

Resource Monitor의 매개변수

1. 크레딧 할당량(Credit Quota) : 지정한 기간 간격 동안 모니터에 할당되는 크레딧 수를 지정. 이 수에는 제한이 없음
 2. Monitor level : 리소스 모니터를 사용하여 전체 계정의 크레딧 사용량을 모니터링할 것인지 아니면 지정하는 WH에 대해 모니터링을 할지 지정.
 3. Schedule : 모니터가 모니터링을 시작할때와 크레딧이 0으로 초기화되는 시기 결정. 일반적으로 크레딧은 매월 초에 0으로 초기화 됨
 4. Actions : 트리거의 일정으로 지정한 기간 내에 크레딧이 임계값에 도달했을 때, 알림을 보내는 역할을 할 수 있음. 알림을 받으려면 Account Administrator가 웹 인터페이스를 통해 명시적으로 알림을 사용하도록 하고 이메일도 적어야함. 3가지 Actions
- Notify (send notification) : 아무런 동작은 취하지 않지만 이메일과 Web UI로 경고 알림을 보냄
 - Notify & Suspend (suspend warehouse) : 알림을 보내고, 모니터에 지정된 모든 WH의 임무를 마무리한 후 일시 중단함.
 - Notify & Suspend Immediately (kill query) : 알림을 보내고, 모니터에 지정된 모든 WH의 동작을 즉시 멈춤

만약 WH가 일시 중단된 상태에서 다시 사용하려는 경우 다음 옵션을 사용

1. 모니터의 credit quota 상향
2. 일시 중단되는 동작의 임계값이 증가함. (즉, 퍼센티지를 말하는 듯)
3. Warehouse가 더 이상 Monitor에 할당 되지 않도록 설정
4. 모니터를 삭제
5. 다음 기간(주기)가 있는 경우 그 시작 날짜에 다시 시작됨

chapter 11 예상문제

Can Resource Monitors be replicated? : True. In 2022, supported

Can a Warehouse be assigned to more than one Resource Monitor?

: NO. three options (1 monitor-1WH, 1M-many WH, account- all monitor)

For which activities does Snowflake have **administration settings** to help with resource consumption -> actions

1. Help control costs associated with unexpected warehouse credit usage
2. Manage access to Snowflake for specific users
3. Manage the availability of the product

What can the resource monitor associated with a Warehouse do when it reaches (or is about to) hit the limit? 1,2,3

1. Suspend the Warehouse
2. Send notification alert
3. Kill the query that is running
4. Delete the Snowflake account

What is the property name from the Resource Monitors that lets you specify whether you want to control the credit usage of your entire account or a specific set of warehouses? 2

- | | |
|-----------------|------------------|
| 1. Credit Quota | 2. Monitor Level |
| 3. Schedule | 4. Notification |

What actions can a Resource Monitor take when it hits the limit?

-> Notify , Notify & suspend, Notify & suspend Immediately

Which properties of a Resource Monitor can be modified? 1,2,3,4

- | | |
|-----------------|------------------|
| 1. Credit Quota | 2. Monitor Level |
| 3. Schedule | 4. Actions |
- > all parameters can be modified

CHAPTER 12 Cache & Query Performance

캐시는 쿼리의 속도를 향상 시킬뿐만 아니라, 비용을 최적화한다

일단 총 3가지의 캐시가 존재한다 (메타데이터 캐시, 쿼리 결과 캐시, 웨어하우스 캐시)
주로 Fastdata Access & Response Time을 제공하기 위해 사용

1. 메타데이터 캐시(MetaData Cache)

- 메타데이터 캐시는 **서비스 계층 전역**에서 유지되며, 오브젝트 정보 및 통계를 포함
- 마이크로파티션에서 사용하는 유형의 메타데이터가 저장된다.
- 메타데이터는 64일 동안 지속되며, WH를 사용하지 않으므로 컴퓨팅 비용은 없다.
- 이 캐시는 MIN, MAX, COUNT와 같은 연산에 대해 수행을 돕는다.
- 그래서 만약 엄청 많은 행들이 있는 테이블을 COUNT 해보면 얼마 안걸림

2. 쿼리 결과 캐시 (Query Result Cache)

- 같은 쿼리를 다시 실행할 때의 시간이 감소함
- 쿼리의 결과를 24시간동안 저장함
- 오직 변경되지 않은 데이터에 대한 쿼리에서만 작동함.
- 캐시를 통한 쿼리 결과 반환은 WH를 사용하지 않음 -> 컴퓨팅 비용 발생 X
- 기본적으로 기록탭에서 다른 사람의 쿼리 결과를 볼 수 없다. 하지만, 같은 **역할이 부여된 사용자**라면, 동일한 쿼리를 수행했을 때 캐시를 사용할 수 있도록 결과를 저장한다.
- 즉 쿼리 결과 캐시는 역할 수준에 저장되는 것 같다.
- 또한 현재 세션에 ALTER를 사용하여 쿼리 결과 캐시를 사용하지 않도록 할 수 있다.

ALTER SESSION SET USE_CACHED_RESULT = FALSE

- 이 캐시는 서비스 레이어에서 비용 부과라서 큰 비용지출이 아님

3. 웨어하우스 캐시(Warehouse Cache)

- 쿼리를 통해 가져온 **테이블**을 **웨어하우스**에 저장하는 것이다.
- WH가 일시 중단되면 캐시의 값들의 잃는다.
- 이를 통해 컴파일 시간/ 대기 시간 / 실행 시간 등 많은 시간을 절약한다.
- 쿼리 결과 캐시는 완전히 똑같은 쿼리에서 작동하는 것이기 때문에 테이블 단위에서는 작동하지 않는다,
- History에서 결과 중 Bytes Scanned 영역이 있다. 이중 파란색은 DW에서 데이터가 재사용된 것을 나타내며, 노란색은 Storage 층에서 가져온 데이터의 양을 의미한다.
- 현재 UI에서는 숫자만 나옴;
- WH에서 데이터를 가져올 때 Bytes Scanned 영역에 찍히는 것 같다.

※만약 COUNT를 찍으면 메타데이터 캐시에서 읽으니 스캔한 byte가 없을 것이다.

Snowflake 성능 향상 방법.

1. Use dedicated Virtual Warehouses (전용 가상 창고 사용) : 즉 내가 수행하려는 작업에 맞는 WH를 사용한다.
2. Scale UP/DOWN for workloads that are known : 만약 특정 시간에 작업량이 증가된다는 것을 알고 있으면 그때 스케일업 한다. 추후에 스케일 다운함.
3. Multi-Warehouses for unknown workloads : 예측할 수 없는 작업에 대해서는 스케일 아웃 & 인을 이용한다.
4. 캐시 사용을 극대화 하도록 시도한다.
5. 클러스터 키 : 큰 테이블에 클러스터 키를 두어 필터링 작업에 성능을 높인다.

chapter 12 예상문제

What are the different caching mechanisms available in Snowflake?

-> metadata cache, query result cache, warehouse cache

A query executed a couple of hours ago, which took more than 5 minutes to run, is executed again, and it returned the results in less than a second. What might have happened?

-> Snowflake used the persisted query result from the query result cache.

Are Snowflake caches automatically invalidated if the underlying data changes?

- True : 데이터가 바뀌면 그 바뀐 데이터에 대한 캐시는 쓸모없어진다. (무효화된다)

What command will you execute if you want to disable the query cache?

-> ALTER SESSION SET USE_CACHED_RESULT = FALSE

Which type of data incur Snowflake storage cost? 1, 2, 4

1. Data Stored in permanent tables.

2. Data Stored in temporal tables.

3. **Cache results.**

4. Data retained for Fail-Safe & Time-Travel.

-> 캐시 결과는 아직 안배웠는데 스토리지 비용에 포함되지 않는다.

Which cache runs for 24 hours?

-> query result cache (Result Cache)

May the warehouse cache be reset if a running warehouse is suspended and resumes? -> true

Does the warehouse cache size change with the warehouse size?

-> true, the larger WH has more WH cache

To improve the performance, which of the below techniques can be used in Snowflake? 1,2,3,4,5

1. Cluster Keys
2. Multi-Warehouses
3. Maximize the cache use
4. Increasing the Warehouse Size
5. **Dedicated (전용) Warehouses**

Optimizing query를 테스트 할때 alter session set use_cached_result=false를 사용

웨어하우스 캐시는 중단 후 즉각적으로 사라짐.

CHAPTER 13 Time Travel, Fail-Safe & Zero-Copy Cloning

위의 이것들은 일조의 복구 시스템이라고 생각하면된다.

첫째로, Time Travel은 함수를 통해 쉽게 복구가 된다. 둘째로 Fail-safe는 snowflake가 유지기간동안 제공해주는 것이다.



- 즉 일반적인 쿼리들을 실행하면 Time Travel SQL을 **자동으로** 실행해준다.
- Fail-Safe는 사용자가 관리할 수 없다

1. Time Travel : 데이터가 바뀌거나 삭제된 데이터에 대한 지점에 복구 가능

- 테이블을 Drop 하면 기간내에 undrop으로 복구가능 (스키마, 데이터베이스 포함)
- 설정 기간 내 어느 시점에서든 과거 데이터에 액세스
- 하지만 이미 동일한 이름의 개체가 존재하는 상황에서 undrop을 하면 에러 발생
- 이럴 때 사용하면 좋음

- 실수 또는 고의로 삭제된 데이터 관련 개체 복원
- 과거 주요 시점의 데이터 복제 및 백업
- 과거 데이터에 대한 쿼리

- Time Travel 기본 값은 1일이며, Enterprise 이상시 90일까지 된다. 하지만 이는 더 많은 저장 공간의 비용을 초래한다.

★또한 과거 데이터에 대한 쿼리도 가능하다.

- 즉 Time Travel에 있는 과거 데이터에 대해 쿼리가 가능
- 방법은 총 3가지가 있다.

○ TimeStamp :

```
SELECT *  
FROM my_table  
AT(timestamp => 'Mon, 01 May 2021 08:00:00 -0700'::timestamp_tz);
```

○ offset : 초단위 인듯

```
SELECT *  
FROM my_table  
AT(offset => -60*15);
```

○ query statement ID : History 탭에서 확인 가능

```
SELECT *  
FROM my_table  
BEFORE(STATEMENT => '8e5d0ca9-005e-44e6-b858-a8f5b37c5726');
```

2. Fail Safe

- Fail Safe는 시스템 장애 또는 기타 치명적인 이벤트 발생시의 데이터를 보호하는 역할
- Time Travel 기간이 끝난 데이터는 즉시 Fail-safe가 시작되며, 이또한 마찬가지로 추가적인 공간을 사용하기 때문에 추가 비용이 발생한다
- 이 Fail-safe는 오직 영속테이블에만 적용된다.--> 7일 적용 (바꿀 수 없음)
- 또한 복구는 스노우플레이크 지원팀에 요청을 해야만 가능하다.

3. Zero-Copy Cloning

- 이를 사용하면 테이블, 스키마 또는 데이터베이스의 스냅샷을 만들 수 있음
- 복제된 개체는 독립적이며 원본에 영향이 없음. (개발 업무에 적합)
- Cloning은 데이터를 복제하는 것이 아닌 마이크로 파티션의 메타데이터를 복제한다.
- 따라서 Zero-Copy Cloning은 비용을 소비 하지 않음.
- 하지만 일부 복제된 데이터를 수정할 경우, 마이크로 파티션을 재생성해야하므로 비용이 많이 들기 때문에 스토리지를 소비한다. 즉, 읽기 작업 같은 것은 비용이 들지 않는다.
- **클론에 데이터를 수정하면 수정한 블록에 대한 반영사항만이 스토리지 비용으로 추가된다.**
- 복제된 개체가 생성되는 순간 그 개체만의 생명주기가 작동한다.

고려사항

- 권한은 복제 안됨
- Named internal Stage는 복제 안되지만 External Stage는 clone 됨
- 테이블 Stage는 복제 됨
- 파이프는 복제되지만 Named internal stage를 참조하면 안됨. external은 됨
- 데이터 history는 복제 안됨
- 복제할 개체에 따라 몇몇의 권한들이 필요하다.

	Table	Pipe	Stream	Task	Other objects
Needed Privilege	SELECT	OWNERSHIP	OWNERSHIP	OWNERSHIP	USAGE

※Time-Travel 기간을 변경하면 날짜에 맞게 Fail-safe에 들어가지 않은 데이터들은 유효기간인 연장되거나 축소 될수 있다.

즉 10/10일 retention time =10 days인 테이블을 20으로 바꾸고 10월 12일날 확인하면 Time-travel은 10/1일 10/2일 의 데이터를 가지고 있다.

chapter 13 예상문제

What type of data incur in Snowflake Storage costs? 1,2,4,6

1. Data stored in permanent tables
2. Fail-Safe data
3. Zero-copy cloning data
4. Time travel data
5. Data stored in cloud storage like AWS
6. Data stored in temporary tables

여기서 Zero-copy coning data라는 것은 수정되기 전의 클로닝된 데이터를 말함

What should be the first option to restore data into a table? 1

1. Time-Travel
2. Fail-Safe
3. Zero-Copy Cloning

Which Snowflake edition allows only one day of Time Travel? 1

1. Standard
2. Enterprise
3. Business Critical

Which Snowflake edition (and above) allows until 90 days of Time Travel? 2

1. Standard
2. Enterprise
3. Business Critical

Do you need to contact Snowflake Support to retrieve data from Fail-Safe? 1

1. True
2. False

Is the Fail-Safe period for temporary and transient tables 0? true

->++ temporary and transient tables can be 0-1 days for time travel

Select the statements that are true about Fail-Safe: 2, 3 , 5

1. There is no difference between Time-Travel and Fail-Safe
2. The data retention period for a permanent table with 30 days of Time-Travel is 37days
3. Fail-Safe provides a non-configurable 7-day period
4. Fail-Safe provides a non-configurable 90-day period
5. Fail-Safe ensures that historical data is protected in the event of a system failure or other catastrophic events

A point-in-time snapshot of data that users can update is called... 3

1. Time-Travel
2. Fail-Safe
3. Zero-Copy Cloning

Which service does Snowflake use to provide the Zero-Copy cloning functionality? 3

- 1. SSD Cache of the Virtual Warehouses
- 2. Cache
- 3. Metadata from the service layer

What would happen if we executed the following command? 4

```
CREATE OR REPLACE TABLE newTable CLONE table1;
```

- 1. “newTable” is created with all the data from “table1”
- 2. “newTable” is created, and Snowflake internally executes a batch job to copy all the data from “table1”
- 3. “newTable” is created, and Snowflake internally executes a pipe to copy all the data from “table1”
- 4. Snowflake creates a new entry in the metadata store to keep track of the new clone. The existing micro-partitions of “table1” are mapped to the new table.

CHAPTER 14 Roles in Snowflake

Role이라는 것은 스노우플레이크에서 보안과 매우 관련있는 특징이다,

액세스 관리 키 개념에 대해. 다음 두 모델이 결합된 형태를 가짐

- Discretionary Access Control (DAC) : 각 객체는 소유주가 있으며, 각 소유자는 해당 객체에 대한 접근 권한을 부여할 수 있음
- Role-Based Access Control (RBAC) : 권한은 역할에 할당되며, 역할은 다시 사용자에게 부여됨.

여기서

- 사용자란 사람또는 프로그램을 의미
- 역할이란 권한을 부여할 수 있는 개체로 사용자에게 역할 부여 가능. 역할은 계정 수준의 오브젝트이며, 다른 역할에게 부여하여 역할의 계층 구조를 만들 수 있음. (상속)
- 보안 가능한 개체 (Securable Object) : 접근 권한을 부여할 수 있는 개체를 의미. 즉, 테이블, 데이터베이스, 또는 스키마 같은 것들이 해당됨.
- 권한(Privilege) : 각 개체에 대한 정의된 수준의 접근 권한을 의미. 상세한 권한 하나를 의미한다.

※권한은 역할에 할당되고, 역할은 사용자에게 할당된다.

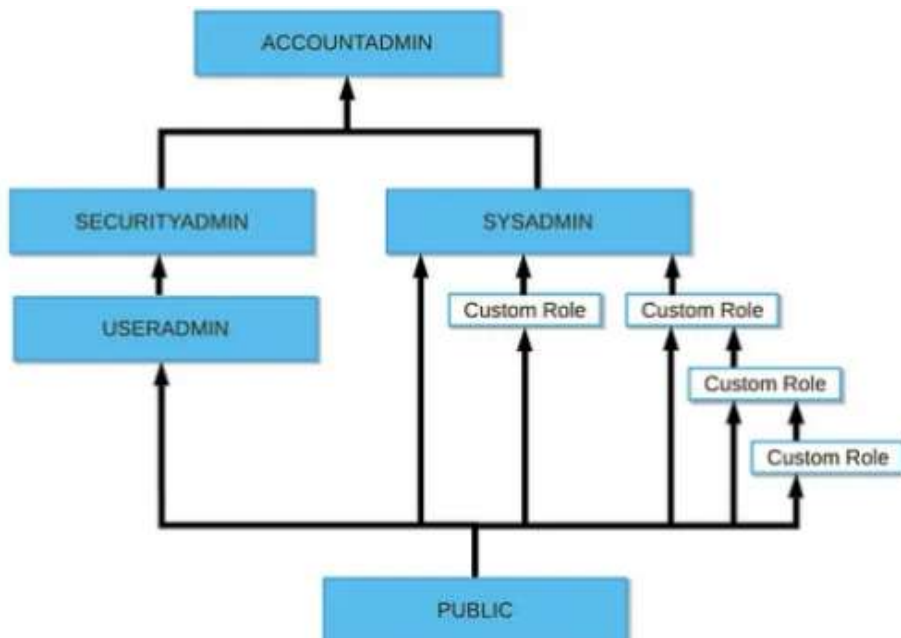
Snowflake는 기본적으로 5가지의 역할이 주어지고, 커스텀으로 만들 수 있다,

	Role	Account panel	Notifications	Create Shares	Network Policies	Use
	ACCOUNTADMIN	Yes	Yes	Yes	Yes	Top-level role.
	SECURITYADMIN	Yes	No	No	Yes	Manage users & roles & Network Policies
	SYSADMIN	No	No	No	No	Manage objects
	USERADMIN	No	No	No	No	Manage users & roles
	PUBLIC	No	No	No	No	Lowest Role.
	CUSTOM	No	No	No	No	Depends on the assigned privileges

- 추가로 계정을 생성하거나 관리하는 ORGADMIN이 있는데 시스템 역할의 계층 구조에 포함되지 않는다.

1. AccountAdmin : SYSADMIN과 SECURITYADMIN 시스템 정의 역할을 포함함. 최상위 역할이며, 많은 사용자에게 부여하면 안됨.
 - AccountAdmin 사용자는 Snowflake UI에서 Account 및 Notification섹션에 액세스.\
 - **Network Policies 와 Shares**에 대한 **CREATE, ALTER, DROP** 가능
2. SECURITYADMIN : 사용자와 역할을 관리하는 역할이다.
 - UI에서 Account 탭에 액세스할 수 있으나, Usage&Billing 부분을 볼 수는 없음
3. SYSADMIN : 웨어하우스를 만들거나 데이터베이스, 다른 오브젝트를 생성할 수 있는 역할이다.
4. USERADMIN : 이 역할은 오직 사용자 및 역할 관리에만 사용된다. SecurityAdmin보다 적은 권한을 가진 관리자 역할과 같다.
5. PUBLIC : 계정 내의 모든 사용자와 모든 역할에게 부여되는 역할. 자신의 개체를 소유할 수 있으나 가장 낮은 역할이라 모든 사용자에게 이용 가능
6. Custom Roles : UserAdmin 이상의 역할에 의해 생성되어, 생성된 역할에 권한을 부여하여 사용.

역할의 캡슐화. (상속을 의미) --> "Grant role <child_role> to role <parent role>"
 예를 들어 ACCOUNTADMIN은 SYSADMIN과 SECURITY를 상속하는데 이때 권한을 캡슐화하여 사용한다는 것이다.



- 사용자를 만들 때 역할을 지정해야하는데, Default_role을 지정하더라도 실질적으로 역할을 유저에게 부여해야 사용할 수 있다.
 - 역할에 대한 모든 권한을 확인하고 싶으면 -> Show GRANTS to role <ROLE>
 - 해당 역할이 부여된 사용자들을 확인하고 싶으면 -> Show GRANTS of role <ROLE>
- + 고급: 소유권을 특정 권한을 DBA에게 주어 SYSADMIN이 사용자의 데이터베이스를 볼 수 없도록 할 수도 있다.

chapter 14 예상문제

Which answers are true about roles in Snowflake? 2,3

1. Snowflake users have a limit on the number of roles that they can assume
2. **Snowflake users can have one or more roles**
3. **Only a role can be active for a particular session**
4. Privileges can be directly assigned to users

Which of the following object types are stored within a schema? 1,2,3

1. Tables
2. Views
3. File Formats
4. Roles

Which of the following roles are the default ones in Snowflake? 1,2,4,5

1. ACCOUNTADMIN
2. SECURITYADMIN
3. VIEWER
4. USERADMIN
5. SYSADMIN
6. NETWORKADMIN

Which command will you run to list all users and roles to which a role has been granted? 2

1. SHOW GRANTS TO ROLE <ROLE>
2. **SHOW GRANTS OF ROLE <ROLE>**
3. SHOW GRANTS IN ROLE <ROLE>

Which roles can create, alter or drop network policies? 1, 2

1. ACCOUNTADMIN
2. SECURITYADMIN
3. SYSADMIN
4. USERADMIN

Which roles can create shares and resource monitors? 1

- 1. ACCOUNTADMIN
- 2. SECURITYADMIN
- 3. SYSADMIN
- 4. USERADMIN

Can worksheets of the Snowflake UI have a different role, warehouse, and database? 1

- 1. True
- 2. False

CHAPTER 15 Data Sharing

- 다른 회사와 특정 테이블만 공유하고 싶은 경우.. -> Snowflake Secure Data Sharing
- 보안 데이터 공유는 계정 내 DB에서 선택한 객체를 다른 Snowflake 계정과 공유할 수 있음. 이때, 공유된 객체는 수정 불가하며 오직 읽기전용임.
- 다음과 같은 객체 공유 가능

테이블	외부 테이블	보안 뷰	보안된 구체화된 뷰	보안된 UDFs
-----	--------	------	------------	----------

- 공유는 DB를 공유하는데 필요한 모든 정보를 갖고 있는 또다른 객체이다.
 - 공유 객체는 다음을 포함
 - 1. 공유할 DB와 해당 객체를 포함하는 스키마에 접근 권한을 부여하는 권한들
 - 2. 우리가 공유하려는 객체에 접근할 수 있도록 하는 권한
 - 3. 데이터베이스와 해당 객체를 공유할 대상 계정들.
- ※ share 오브젝트를 생성 후 이 오브젝트에 권한을 부여한다.

Producers & Consumers : 제공하는자와 제공받는자.

Producers는 Share를 생성하고 다른 계정이 사용할 수 있도록하는 계정을 의미함. 이 생산자는 공유하고자하는 데이터의 저장 공간 비용을 지불함.

Consumer는 공유를 받는 계정들.

- 실제 데이터는 계정들간 복사되거나 전송되지 않으므로 공유된 데이터는 최신상태를 유지
- 생산자가 저장 비용을 지불하고 있어서 비용은 없음.
- 공유된 데이터로 복제 생성이나 Time-Travel 불가
- 소비자가 공유를 받으면 그 공유로 데이터베이스를 만들어야 모든 공유 객체에 액세스 가능
- 기본적으로 데이터를 공유하려면 같은 지역에 있어야하지만, Snowflake는 데이터베이스 복제를 허용하여 서로 다른 지역 및 서로 다른 클라우드 플랫폼에서 안전하게 공유 가능

소비자에게는 2가지 종류가 있음

- 1. Full account : 기존 snowflake 계정과 데이터를 공유시 적용. 소비자는 실행한 쿼리에 대한 비용은 자신이 지불. 하지만 저장공간은 생산자가 지불
- 2. Reader account : Snowflake 계정을 가지고 있지 않은 계정과 공유할 때 사용됨.
 - snowflake 계정 없이도 공유 가능.
 - 각 Reader의 계정은 생산자 계정에 속하고, 당연히 생산자가 모두 비용 부담
 - Reader account는 오직 생산자 계정으로부터만 데이터 소비

INBOUND & OUTBOUND Shares

1. OUTBOUND Share : 데이터를 공유하려는 자신(생산자)이 만든 Share 객체
 2. INBOUND share : 다른 계정(생산자)이 생성하여 자신(소비자)와 공유한 Share 객체
- 인바운드 공유가 본인이 사용한 공유 객체가 되는 것이다.

chapter 15 예상문제.

What is the minimum Snowflake edition needed for the Data Sharing capability?

1. Standard 1,2,3
2. Enterprise
3. Business Critical

Which database objects can be shared using Snowflake Secure Data Sharing?

1. Tables 1,2,3,4,5
2. External tables
3. Secure views
4. Secure materialized views
5. Secure UDFs
6. Users

How can a producer share a table with a consumer located in a different region? 2

1. This is not a problem; producers and consumers can be in different regions
2. Replicate your account to another region and create a share from that region.
3. Create a script to replicate your data in the consumer account.

Which Snowflake role can set up a Snowflake Share? 4

1. SECURITYADMIN
2. SYSADMIN
3. DATASHARINGADMIN
4. ACCOUNTADMIN
5. PUBLIC

What are the two types of data consumer accounts available in Snowflake?

1. Shared Account
2. Reader Account
3. Public Account
4. Full Account

Is shared data always immediately available to Consumer Accounts? 1

1. True
2. False