Snowflake 필기(6~10)

CHAPTER 06. Tables & Views

테이블은 다른 SQL 언어에서와 같이 DB에서 데이터를 포함하는 오브젝트이다.
테이블에는 4가지 종류의 타입이 있고 조금씩 다름

Type of Table	Time Travel	Fail Safe	Cloning	Create views
Permanent	0-90 days.	Si	Permanent Transient Temporary	Yes
Transient	0-1 days.	No	Transient Temporary	Yes
Temporary	0-1 days.	No	Transient Temporary	Yes
External	No	No	No	Yes

Transient Table은 각 **세션 이후에도 유지**해야 하는 일시적인 데이터에 사용
Temporary table은 비영속적인 데이터를 저장하거나 세션일 때만 사용된다.
- 따라서 다른 세션이나 다른 사용자에게는 노출되지 않는다.
- 세션이 끝나면 완전히 데이터는 제거된다.

External table은 외부의 DB를 파일 형태로 가진것이므로 오직 읽기 연산밖에 안됨'
- 파일 수준의 메타데이터만을 포함한다.

※Time Travel은 Enterprise 이상에서 제공하는 옵션이다.

뷰는 테이블 쿼리의 결과를 보여주는 역할을 한다.
뷰는 거의 어디서든 사용가능하다.
뷰는 크게 두가지 타입으로 나뉘는데, 하나는 Non-Materialized(normally view), 다른 하나는 Materialized. 이 둘은 보안 또는 비보안적인 상태일 수 있다..

Type of View	Edition you can create them	Do they store data?	Can you perform clustering?	Can you share them?	Are stream supported?
Regular	All editions	No	No	No	Yes
Materialized	Enterprise	Yes	Yes	No	No
Secure	All editions, but you need Enterprise for secure materialized views.	Regular no Materialized yes	Regular no Materialized yes	Yes	Yes

일반적인 뷰는 결과를 저장하지 않기 때문에, 성능도 구체화된 뷰보다 느리다.
- 데이터를 저장하지 않으므로 클러스터링도 불가하다
- 만약 보안이 적용되어 있다면 share가 가능하다. 보통은 안된다.
- 2022년 3월 스트림을 지원한다. (일반뷰와 Secure view에)

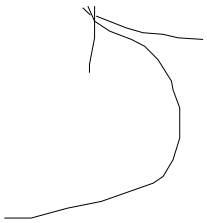
Materialized view
- 일반적으로 테이블처럼 동작한다
- 결과를 빠르게 액세스할 수 있도록 저장되나 저장공간과 적극적인 유지 관리가 필요하여 추가적인 비용이 발생한다
- 또한 Enterprise 이상 에디션에서만 사용가능하다
- 또한 기본 테이블에 대한 변경 사항은 Materialized view 자동적으로 전파[illegible] 클러스터 키 가능
  (즉, 열을 추가하면 그 열은 나타나지 않고, 열을 삭제하면 에러를 일으킴)

--〉 그래서 Snowflake가 쿼리를 수행하기전 업데이트하거나 수동 업데이트해야함
- 위와 비슷한 말로, 스트림이 지원되지 않는다.
- 앞서 정보를 테이블처럼 유지하므로 저장 비용과, 자동적이고 투명하게 관리되므로 컴퓨팅 비용이 발생한다

※스트림(stream)이란, Snowflake에서 지원하는 기능 중 하나로 데이터 테이블에 대한 변경 사항을 말합니다. 테이블에 추가, 수정, 삭제 등의 DML 작업이 발생하면 이를 스트림이 추적하고 기록합니다.
즉, 해당 뷰가 이런 스트림 객체를 통해 테이블의 변경 사항을 참조하거나 반영할 수 있다는 의미입니다.

Materialized view 사용하기 좋은 조건
1. 쿼리 결과가 (기본 테이블 기준으로) 적은 수의 행/열을 반환하는 경우
2. 쿼리 결과가 반정형 데이터 또는 집계 분석에 상당한 처리 시간을 요구하는 경우
3. 쿼리가 외부의 테이블에서 수행하는 경우, 특히 더 낮은 성능을 보이는 외부에서
4. 테이블이 자주 바뀌지 않는 뷰일 때 사용

보안 뷰 (Secure view)
: 개인정보나 보안을 위해 내부 데이터를 숨기는 용도로 사용
- 오직 권한이 있는 자만 표시됨
- 뷰가 어떻게 생성되었는지 알수 없음.
- 일반 뷰와 Materialized view는 secure view가 될 수 있음
- 보안이 적용되면 바로 공유 가능
- 이 공유 가능이라는 것은 뷰 자체를 공유한다는 것이아니라 접근의 의미로써 공유가 가능하다는 뜻이다.



chapter 06 예상 문제

What types of tables are available in Snowflake?

-〉 Permanent, Temporary, Transient, External

What types of views are available in Snowflake?

-〉 regular(Non-Materialized), Materialized view, Secure view

We need to temporarily store intermediate data, which an ETL process will only use. We don't need the data outside the ETL process. If you want to optimize storage cost, what type of table will you create to store this data?

-〉 Temprorary view. Because it needs during the ETL session.

Can you use streams to track changes in materialized view?

-〉 NO

In which of the below scenarios will you use an external table?

1. You have data on the cloud providers, but the data cannot be copied or moved to any other location due to compliance regulations.
2. You have a high volume of data on the cloud providers, but we only need some of the data in Snowflake.
3. You have data on the cloud providers that need to be updated by Snowflake.
4. You have XML data on the cloud provider

Solution: 1, 2. The third answer is incorrect, as external tables can only read data. They also don't support XML.

Select two limitations with materialized views:

1. Time Travel is not supported.
2. We cannot define Cluster Keys.
3. Streams cannot track changes.

Solution ; 1,3. Materialized view stores data, so it can define Cluster key

Can temporary tables be created with a clustering key defined?

Yes. Because it stores data inside.

CHAPTER 07. Stages & Storage Integration

스테이지는 파일들이 스노우플레이크에 이동하기 전의 위치를 나타내는 것이다.

External Stages(일반 스테이지)는 Snowflake 바깥에 저장되어 있는 데이터 파일을 참조
명명된 이름은 데이터베이스 오브젝트이다.
이 외부 스테이지들은 위의 3가지 클라우드 공급자에 대해서만 지원된다.

Internal stages Snowflake 안에 데이터 파일을 저장
- 유저(개인) 별로 기본적인 파일저장을 위해 할당된 Snowflake 스테이지가 있으며, 해당 스테이지가 속한 사용자 외의 누구도 액세스 할 수 없음
-이 개인 스테이지는 From절을 이용해 복사작업을 수행할 수 있음
- Snowflake의 기본적인 UI로는 개인 스테이지를 확인하거나 삭제/변경 불가능하며 CLI를 통해서만 접근이 가능
- 테이블로 복사한 데이터는 그 뒤 스토우플레이크에 남게 되며 클라우드 제공자에게 가져오지 않는다.

PUT file://c:/myData/myCSV.csv @~;
COPY INTO MYTABLE FROM @~staged file_format=(format_name = 'my_csv_format');
에서 첫 번째 명령어는 로컬의 파일을 내부스테이지에 집어넣는 것이고, ("@~")
두 번째 명령어는 사용자 내부 스테이지에서 mytable이라는 테이블로 데이터를 복사한다.
여기서 Staged file_format의 인수로서 압축할지 말지 결정할 수 있음

각 테이블에는 기본적으로 파일 저장을 위한 Snowflake stage가 할당되어 있음
이 스테이지는 여러 사용자가 파일에 액세스할 수 있어야하고 파일이 단 하나의 테이블에 있으면 편리한 옵션이 된다.

PUT file://c:/myData/myCSV.csv @%myTable
이 명령어는 자신 로컬에 있는 파일을 mytable의 table stage로 복사하는 것이다.
즉 ("@%")를 사용하고 이또한 CLI에서 확인할 수 있다.

Named Internal Stage : Snowflake에서 관리하는 클라우드 스토리지 위치
Named Stage는 외부와 내부 스테이지를 모두 가리키는 데이터베이스 객체
만약 로드시 데이터 파일을 단독으로 사용하거나 하나의 테이블에 로드할 것이면, 사용자 스테이지나 테이블 스테이지를 사용하는 것이 더 적합
Named Internal Stages는 **영구적** 또는 **일시적**일 수 있다.

Stage Metadata :

Snowflake는 외부/내부 스테이직 파일들에 대한 메타데이터를 자동적으로 생성한다.

이 메타데이터는 다음의 가상적인 열에 저장된다.

●METADATA$FILENAME →현재 행이 속한 스테이지 데이터 파일의 이름입니다. 이는 스테이지 내의 데이터 파일 경로를 포함합니다

●METADATA$FILE_ROW_NUMBER → 각 스테이지 데이터 파일 내의 레코드(record) 별로 해당 레코드가 위치한 행 번호입니다.

```
select metadata filename, metadata$file_row_number, t.$1, t.$2 from
@mystage1 (file_format => myformat) t;
```

이 명령어는 메타데이터를 쿼리하는 방법의 예제입니다.

STORAGE INTEGRATION

이는 외부 클라우드 저장소에 대한 ID 또는 IAM 개체를 저장하는 Snowflake 오브젝트임.

이는 또한 선택적으로 허용되거나 차단된 스토리지의 위치(외부 클라우드)의 집합도 포함함

- Storage Integration은 사용자가 데이터를 로드하거나 언로드할 때 자격증명을 안해도 되게하도록 하는 **계정수준**의 오브젝트이다.
- 또한 여러 스테이지를 지원한다.

```
create storage integration <name>
  type = external_stage
  storage_provider = s3
  storage_aws_role_arn = 'arn:aws:iam::001234567890:role/myrole'
  enabled = true
  storage_allowed_locations = ('s3://mybucket1/path1/',
's3://mybucket2/path2/';
```

이 명령어는 Storage Integration을 생성하는 명령이다.

chapter 07 예상 문제

What are the different stages available in Snowflake?

-〉 User, Table, Named Internal, Named External

Which are the two metadata columns for staged files?

-〉 METADATA$FILENAME , METADATA$FILE_ROW_NUMBER

Which character identifies a table stage?

-〉4. “@%”

Which character identifies a user stage?

1. “@~”

When staging uncompressed files in a Snowflake stage, are the files automatic ally compressed using gzip unless compression is explicitly disabled?

1. True

Which Snowflake object stores a generated identity and access management (I AM) entity for your external cloud storage, along with an optional set of allowe d or blocked storage locations (Amazon S3, Google Cloud Storage, or Microsoft Azure)?

-〉 Storage Integration Object

Can a single storage integration support multiple external stages?

-〉 Yes

You have two types of named stages, one is an external stage, and the other one is an internal stage. Will external stages always require a cloud storage pr ovider?

-〉 Sure. External Stages must be created by Three Cloud Provider

CHAPTER 08. Data Loading – with snowpipe

보통 우리가 데이터를 Snowflake에 로드할때 다음 방식을 따른다
원본 시스템 -〉 스노우플레이크 스테이지 -〉 스노우플레이크 테이블
원본 데이터가 있는 시스템에서 스테이지로 전송하고, 스테이지에 전송하면 이를 복사함.

그런데 이 복사하는 방법이 2가지가 있음
1. Bulk Load(대량 로드) : 데이터를 대량으로 한번에 로드하는 방식
2. Continuous Load (연속 로드)

Bulk Load
- 이미 스테이지에 있는 파일들로부터 데이터를 배치 단위(여러 개의 파일을 하나로 묶어서)로 Snowflake 테이블로 로드하는 방식.
- 이를 위해 COPY 명령어 사용 (WH 크레딧 소모)
- COPY 명령어는 데이터 로딩 중 데이터 변환 지원(열 재정렬, 열 생략, 형 변환 등)
- COPY INTO 명령어는 외부 스테이지든 사용자/내부 스테이지든 모든 스테이지에 동작하여 GZIP 압축 효율 같은 요소들이 영향을 받아 테이블에 로드함.

데이터 로드 시 수행될 수 있는 변환
1. Truncate Columns (열 잘라내기) : 로드할 데이터의 각 열을 지정된 길이로 잘라냄. 즉, 텍스트 열의 일부 내용을 지정된 길이로 제한
2. Omit columns (열 생략) : 로드할 때 특정 열을 제외(생략)하여 로드
3. Reorder columns(열 재정렬) : 로드할 때 열의 순서를 변경.
4. Cast (자료형변환) : 로드할 데이터의 자료형을 대상 테이블에 맞게 변환. 예를 들어 숫자형을 문자형으로 변환 등

LOAD 와 UNLOAD
스테이지에서 테이블로 가면 LOAD, 테이블에서 스테이지로 가면 UNLOAD라 한다.

고려사항
- 데이터를 복사할 테이블 이름, 파일이 위치한 스테이지, 복사할 파일 또는 파일 패턴, 그리고 파일 형식을 지정해야 합니다
- 다음 64일 동안은 동일한 파일을 복사(COPY)할 수 없습니다. 이 경우, "FORCE=True" 명령을 지정하지 않으면 파일을 다시 복사할 수 없습니다.
- 로컬 드라이브에서 파일을 로드하거나 언로드할 수 없습니다.
- 일부 변환 작업인 Flatten, Join, Group by, Filters 또는 Aggregations와 같은 작업은 지원되지 않습니다.
- Snowflake UI를 사용하여 로드할 수 있는 파일 크기는 50MB로 제한됩니다. 그러나 SnowSQL을 사용하면 더 큰 파일도 복사할 수 있습니다.
- 입력 데이터를 세분화된 경로로 구성하면 로드 성능을 향상시킬 수 있습니다.
----〉세분화된 경로란 데이터를 디렉토리 구조로 구분하는 것을 의미합니다. 예를 들어, 데이터를 연도별로, 월별로, 일별로 디렉토리를 생성하여 구분하는 방법입니다.

에러 발생 시 오류 처리 옵션 : 일부 파일 로드 시 에러 발생 가능.
○ABORT_STATEMENT : 데이터 파일에 오류가 있는 경우 로드 작업 중단
○CONTINUE : 데이터 파일 오류가 발생해도 계속 로드 (레코드 단위)
○SKIP_FILE : 데이터 파일 오류가 있으면 건너뛰기
○SKIP_FILE_num : 파일 내 오류 레코드 수가 지정된 수와 동일하거나 그 수가 초과하면 해당 파일을 건너뜀
○SKIP_FILE_num% : 파일 내 오류 레코드의 비율이 지정된 비율을 넘으면 스킵
※validation_mode =Return_ALL_ERRORS;을 통해 오류가 생기는 레코드를 확인할 수 있는데, 실제 COPY 명령은 작동되지 않는다,

COPY INTO 명령어에서 시험에서 자주 출제되는 기타 옵션들은 다음과 같습니다:
- pattern = 〈pattern〉 → 패턴 매칭을 사용하여 스테이지에서 파일을 테이블로 로드
=〉 예를 들어, 'pattern = *.csv' 옵션을 사용하면 스테이지에 있는 모든 .csv 확장자를 가진 파일들을 테이블로 로드하게 됩니다. 이렇게 파일 이름의 패턴을 지정하면 특정 유형의 파일만 선택적으로 로드할 수 있어서 데이터 로딩 작업을 효율적으로 관리할 수 있습니다.

- FORCE = TRUE → 테이블로 파일을 복사한 후, 파일의 메타데이터 때문에 그 파일들은 다음 64일 동안은 다시 복사할 수 없습니다. 만약 이 옵션이 true로 설정되면, 이전에 로드된 파일이더라도 변경되지 않았다면 모든 파일들을 로드합니다.

- PURGE = TRUE → 데이터 로드 작업이 성공적으로 완료된 후 스테이지에서 데이터 파일을 자동으로 삭제할지 여부를 지정합니다. 만약 파일 삭제 작업이 어떤 이유로 실패해도 오류가 반환되지 않습니다. 파일 삭제 작업의 성공 여부를 확인하는 훌륭한 방법은 "LIST stage" 명령어를 사용하여 스테이지에서 파일들의 목록을 확인하는 것입니다.

- MAX_FILE_SIZE → 이 옵션을 사용하여 데이터를 언로드(Unload)할 때 각 파일의 최대 크기를 지정할 수 있습니다.

CONTINUOUS LOAD (지속적인 로드)

데이터의 소량(마이크로 배치)을 로드하고, 이를 점차적(단계적)으로 분석에 활용할 수 있도록 하는 방법. 이를 위한 다양한 방식이 있음

1. Snowpipe : 가장 간단하고 인기 있음
2. Snowflake Connector for Kafka : 아파치 Kafka 토픽으로부터 데이터를 읽어와 Snowflake 테이블로 로드.
3. Third-Party Data Integration Tools (타사 데이터 통합 도구) : 다른 지원되는 통합 도구로 데이터 로드

※Kafka는 분산형 스트리밍 플랫폼으로, 대용량의 실시간 데이터 스트림을 처리하는 데 사용되는 오픈 소스 솔루션

Snowpipe

- 파일이 어떤 스테이지에 있는 데이터를 로드할 수 있도록 허용한다.
- 이는 적은 양의 반복적인 데이터가 자주 발생할 때 사용하며, 데이터를 지속적으로 로드하는데 사용한다.(마이크로 배치)
- 서버리스 방식이며, **WH를 사용하지 않는다.**
- 스트리밍 또는 Near Real-time 데이터에 사용
- 일반적으로 가장 오래전에 스테이징된 파일부터 처리하나, **보장되는 순서는 없다.**

Snowpipe가 스테이지의 새로운 파일을 감지하는 방법

1. 클라우드 메시징을 이용한 Snowpipe 자동화
- Amazon SQS를 통해 스테이지에 이벤트 발생시 트리거 설정 가능
- SQS를 이용하면 파일 누락에 대한 정보를 저장

2. Snowpipe REST 엔드포인트 호출
- 클라이언트 애플리케이션에서 파이프 객체의 이름과 데이터 파일명 목록을 포함하여 공개 REST 엔드포인트를 호출.
- JSON 키 쌍 인증을 통해 API의 함수를 통해 요청한다.

추가 고려사항

- 14일 동안의 메타데이터 (COPY INTO의 경우 64일)를 가지고 있습니다. 이 기간 동안에는 동일한 파일을 다시 복사할 수 없습니다.
- 기본적으로 Snowpipe는 파일 로딩 중에 오류가 발생하면 "SKIP_FILE"을 수행합니다. 반면에 COPY INTO 명령의 기본 ON_ERROR 옵션은 "ABORT_STATEMENT"입니다.
- snowpipe를 중지하는 명령을 하려면 계정 수준이므로 Accountadmin이면 됨.

chapter 08 예상 문제

What are the usual data-loading steps in Snowflake?

-〉 Source System – Snowflake Stage – Snowflake Table

What key concepts will need to be considered while loading data into Snowflake?

1. Stage
2. File Format
3. Transformation
4. Region of your Snowflake account
5. Error validation

1,2,3,5

Using COPY INTO command, you can unload data from a table into which locations?

1. Named internal stage (or table/user stage).
2. Named external stage that references an external location (Amazon S3, Google Cloud Storage, or Microsoft Azure).
3. An external location like Amazon S3 or Azure.
4. Local Drive

-〉 1,2,3, it can unload any stages and Cloud Service provider

After how many days does the COPY INTO load metadata expire?

-〉 64 DAYS

While loading data through the COPY command, you can transform the data. Which of the below transformations are allowed? 1,2,4,5

1. Truncate columns
2. Omit columns
3. Filters
4. Reorder columns
5. Cast
6. Aggregate

After a successful load into a Snowflake table with the COPY INTO command, what option will you specify to delete the stage files? 3

1. DELETE = TRUE
2. REMOVE = TRUE
3. PURGE = TRUE --〉 테이블로 로드하면 스테이지의 파일은 자동 삭제

4. TRUNCATE = TRUE

In which of the below scenarios is Snowpipe recommended to load data? 1
1. We have a small volume of frequent data
2. We have a huge volume of data generated as part of a batch schedule
3. In both of the previous scenarios

Is Snowpipe Serverless? -〉 true

After how many days does the load history of Snowpipe expire? -〉 14 days
※Must be requested from Snowflake via a REST endpoint, SQL table function, or ACCOUNT_USAGE view.

Can Snowpipe load a file with the same name if it has been modified later?
-〉 False . This is because of the Snowpipe metadata. Changing the name doe sn't modify this metadata, so it won't be copied.

Does Snowpipe guarantee that files are loaded in the same order they are staged? -〉 FALSE. it' doen't guarantee the same order.

Which of the below APIs are Snowpipe REST APIs?
three endpoints : **insertFiles, insertReport, loadHistoryScan**

Which data-loading method requires a user-specified warehouse to execute COPY statements? 1
1. Bulk Data Load
2. Snowpipe
3. Both

Snowpipe doesn't use WH

CHAPTER 09. PUT & GET commands

PUT 명령어를 통해 로컬에 있는 파일을 업로드하여 INTERNAL 스테이지로 이동 시킬 수 있음. 이 명령어는 외부스테이에서는 작동하지 않음.
또한 SnowSQL을 통해서만 가능.
또한 ,COPY INTO를 통해 내부 스테이지에서 테이블로 이동

GET COMMAND
- Snowflake 안에 있는 스테이지에서 데이터를 클라이언트 컴퓨터의 디렉토리/폴더로 다운로드하는 명령어.
- 테이블이 아닌 스테이지에서 데이터를 다운.
- 만약 스테이지에 파일이 없으면 테이블에서 스테이지로 언로드해야함.

chapter 09 예상 문제

COPY INTO나 LIST SHOW 같은 경우는 모두 Snowflake UI에서 실행가능하나, GET, PUT은 오직 Snowsql에서만 가능하다는 것을 알아야한다.

Which of the following commands cannot be executed from the Snowflake UI? 3,5
1. SHOW
2. LIST
3. GET
4. COPY INTO
5. PUT

Which command will we use to download the files from the stage/location loaded through the COPY INTO 〈LOCATION〉 command?
1. GET
2. PUT
3. UNLOAD
4. INSERT INTO

Does GET support downloading files from external stages?
-〉 GEnerally False. But use the JDBC, its possible

When data is staged to a Snowflake internal staging area using the PUT command, is the data automatically encrypted?
-〉 True

CHAPTER 10. Data Warehouses

Data Warehouse는 컴퓨팅 자원의 클러스터이다.
DW는 CPU, 메모리, 임시 스토리지를 DML 연산을 수행하기 위해 제공
당연히 WH가 가동중이면 Snowflake 크레딧을 소모한다.
전에 말햇듯 웨어하우스 사용 시 첫 1분은 고정비용이고, 그후 초단위로 요금 지불

WH sizes는 당연히 크기에 따라 쿼리 실행에 영향을 줌
쿼리 동시성이 있는 경우 웨어하우스 멀티클러스터링을 통해 관리 가능

또한 비용은 그 멀티 클러스터가 작동된 시간에 따라 비용이 부과됨
스케일은 원래 클러스터의 크기에 맞는 클러스터가 추가로 생기는 거임

스케일 업 : 파워를 늘려 하나의 쿼리를 실행하는데 더 빠르게 함
스케일 아웃 : 클러스터를 늘려 동시에 여러 쿼리를 실행하게 함

최대/최소 클러스터 수에 따라 두가지 모드가 있음.
1. Maximize -〉 거의 항상 지정한 최대의 클러스터 수를 유지함
- 보통 최대 클러스터 수와 최소 클러스터 수를 같게 해서 작동
2. Auto-Scale -〉 필요에 따라 클러스터를 늘이고 없애는 역할을 함.
이는 WH Scaling Policy에 따라 결정됨.
- 최대/최소 클러스터의 수를 다르게 해야함

WH Scaling Policy : 멀티 클러스터 웨어하우스를 만들때는 이 스케일링 정책을 반드시 지정해야함. 이로써 크레딧을 절약하는 데 도움이 됨.
1. Standard Policy : 크레딧을 절약하기 보다 클러스터의 수를 늘리는데 우선순위
2. Economy Policy : 크레딧을 아끼는데 집중된 정책

Auto Suspend & Auto Resume
Snowflake는 WH가 비활성화일 경우 자동 중단됨.
시간을 지정하여 자동 중단 시기를 결정할 수 있음

또한 중단된 상태에서 쿼리 또는 DML과 같이 WH가 필요하면 자동적으로 WH를 재개함.
이는 디폴트 세팅임.

chapter 10 예상 문제

How is query processing done in Snowflake? 3
1. AWS EMR with Spark
2. AWS EC2 with Spark
3. Virtual Warehouses

You have two virtual warehouses in your Snowflake account. If one of them updates the data in the storage layer, when will the other one see it? 1
1. Immediately
2. After an average time of 5 seconds
3. After the sync process

Can you resize the warehouse once you have selected the size?
-〉 True

If you want a dedicated virtual warehouse, which is the lowest Snowflake edition you should opt for? 1
1. Standard
2. Enterprise
3. Business Critical
4. Virtual Private Snowflake

f you want a multi-cluster warehouse, which is the lowest Snowflake edition you should opt for? 2
1. Standard
2. Enterprise
3. Business Critical
4. Virtual Private Snowflake

Queries in Snowflake are getting queued in the warehouses and delaying the ETL processes of the company. What are the possible solution options you can think of, considering we have the Snowflake Enterprise addition? 1,2
1. Resize the warehouse -〉Scale up
2. Use multi-cluster warehouse -〉Scale out
3. Set auto-resize parameter to TRUE
4. Contact Snowflake support to increase the size of the warehouse

A warehouse ran for 62 seconds, and it was suspended. After some time, it r an for another 20 seconds. For how many seconds will you be billed?
-〉 122 Seconds

Can two different virtual warehouses from the same account access the same data imultaneously without any contention issues?
-〉 True

Can virtual warehouses be resized while they are running?
-〉 True

A medium (M) warehouse has auto-suspend configured after 15 minutes. You have noticed that all of the queries that run on this warehouse finish within a minute. What will you do to optimize compute costs? 2
1. Delete the warehouse after a minute
2. Reduce the auto-suspend time to 1 minute
3. Use another data-warehouse

What happens to incoming queries when a warehouse does not have enough resources to process them? 2
1. Queries are aborted
2. Queries are queued and executed when the warehouse has resources
3. Snowflake resizes the warehouse

Which function returns the name of the warehouse of the current session? 3
1. ACTIVE_WAREHOUSE()
2. RUNNING_WAREHOUSE()
3. CURRENT_WAREHOUSE()
4. WAREHOUSE()