Snowflake 필기(16~20)

CHAPTER 16 StoredProcedures&UDFs

- 저장 프로시저와 UDF는 다른 프로그래밍 언어의 함수와 비슷.
- JavaScript, SQL, java, python으로 Snowflake를 확장함

Stored Procedures
- 이를 사용하면 Snowflake SQL을 JavaScript와 결합하여 분기 및 반복과 같은 프로그래밍 구조를 포함할 수 있음.
- Snowpark를 이용하면 Python, Java, Scalar 에서도 작성가능
- 저장 프로시저는 단일 값 또는 아무것도 반환하지 않음.
- 반환된 값은 SQL 문에서 직접 사용 불가함.

User-Defined Functions(UDFs)
- Snowflake 내장 함수로 수행할 수 없는 작업 수행 가능
- SQL, JavaScript, Java, python 으로 사용 가능

Stored Procedures V/S UDFs
- UDFs는 입력 행마다 하나의 출력 행을 반환
- 반환된 행은 단일 열/값으로 구성
- 반드시 반환값이 있어야함
- 반환된 값은 SQL 문에서 직접 사용할 수 있음

Name	Programming languages that are supported	Return	Can you use it directly from a SQL Statement?	Typical use cases
Stored Procedures	JavaScript SQL Java, Python & Scala using Snowpark	0 - 1 value	No	Typical queries (select) + DML (insert, update...) Administrative Tasks.
UDF	JavaScript SQL **Java** **Python**	One output row for each input row	Yes	If you need a function that can be called as part of a SQL statement and that **must** return a value that will be used in the statement.
UDTF	Same as UDF.	0 - Several rows	Yes	Same case as UDF but returning multiple rows.

UDTFs 는 User-Defined Table Functions 로 UDF와 거의 동일.
- 하지만 다중 행을 반환할 수 있다는 것이 유일한 차이점.

```
---- Function definition ----
create function t()
    returns table(msg varchar)
    as
    $$
        select 'Hello'
---- Calling the function ----
select msg
    from table(t())
    order by msg;

---- Result ----
+-------+
| MSG   |
|-------|
| Hello |
| World |
+-------+
```

chapter 16 예상문제

Which Snowflake object returns a set of rows instead of a single, scalar value, and canbe accessed in the FROM clause of a query? 2

1. UDF 2. UDTF 3. Stored procedure

Do UDFs support both SQL & JavaScript in Snowflake?

1. True 2. False 1

Are UDFs, UDTFs, and Stored Procedures account or schema-level objects in Snowflake? 1

1. Schema level 2. Account-level

CHAPTER 17 Task & Transactions

- task는 snowflake 환경 내 실행되는 예약 가능한 스크립트임.
- 이벤트 소스는 task를 트리거할 수 없지만, task는 일정에 따라 실행됨.
- **task는 소유자의 권한에 따라 실행되며, 단일 SQL문과 프로시저를 실행 가능함.**
- Snowflake는 일정이 있는 작업의 인스턴스가 지정된 시간에 한 개만 실행되도록 함.
- 즉 어떤 작업이 5분마다 실행되는데, 그 작업이 5분 넘게 실행중이면 그 작업은 다시 실행되지 않고 스킵됨.
- 또한 작업의 최대 지속 시간은 기본적으로 60분으로, 넘으면 자동을 종료함.
- CRON 표현식과 시간대를 지정하여 예약 작업을 주기적으로 실행할 수 있음.
- 기본적으로 작업을 만들면 일시 중단되고, Alter를 통해 task를 resume해야함.

- 사용자는 루트 작업을 시작으로 종속성을 가지는 트리형 작업구조를 가질 수 있음
- 오직 부모 작업 노드가 완료된 후에 자식 노드가 실행 가능

고려사항

1. 각 작업은 최대 100개의 하위 작업을 가질 수 있음
2. 작업 트리에는 루트 작업을 포함하여 최대 1000개 까지의 노드 가능
3. 하위 작업에는 이전 작업만 가지고 있었음. (즉, 오직 이전 레벨의 노드에서만 받아서 처리), 하지만 22년 9월부터 DAG 지원 (자신의 위에 있는 노드라면 누구든 자신에게 연결 가능)
4. DAG 시 스케줄링은 루트 노트에 지정한다.

부모 노드가 실행되고 그 자식 노드가 동시에 실행다고 하더라도 태스크는 오직 하나의 태스크만 실행되고 나머지는 대기시간을 갖게 된다. 즉, Queing time을 갖게 된다,.

```
CREATE TASK <newTask> AFTER <rootTask>;

ALTER TASK <newTask> ADD AFTER <rootTask>;
```

- 이 구문을 통해 자식 노드를 생성할 수 있다.

만약 이 트리구조에서 Root 노드를 제거하면 다음과 같게 된다.

1. 자식태스크가 독립된 태스크가 되거나
2. 자식 태스크가 루트 태스크가 된다.

태스크 소유권을 가진 역할이 삭제되면, 그 역할을 삭제한 역할에게 소유권이 이전된다.

작업의 기록을 보려면 스노우플레이크 함수를 통해 지정된 날짜 범위의 작업 사용 기록을 볼 수 있다.

- 그 기록들은 Information schema에 들어가 있고 다음 권한들이 필요하다
- AccountAdmin role, ownership of a task, global MONITOR_EXECUTION.

```
SELECT *
FROM table(information_schema.task_history())
ORDER BY scheduled_time;
```

Serverless Tasks

- 만약 5분짜리 작업을 돌린다고 하면 WH의 Auto_suspend에 의해 10분을 사용(낭비)하게된다.
- 그래서 21년 9월 서버리스 task을 발표함
- 서버리스라고 하지만 이는 Snowflake가 관리하는 모델에 의해 task가 실행된다.
- 이 리소스는 기본적인 컴퓨팅 작업보다 1.5배 비싸지만, 사용된만큼 비용지불이라 절감될 수 있다.
- 그 task가 필요한 리소스에 따라 Snowflake가 자동으로 스케일업/다운을 한다.

트랜잭션은 내가 데이터베이스에서 알던 것과 동일하다

- 여기서는 ACID 규정을 준수한다.

고려 사항

1. 트랜잭션이 세션에서 실행 중이다 세션 연결이 끊어지든가 중단되면, 이 트랜잭션이 다른 트랜잭션에서 동일한 테이블을 잠그지 못하도록 차단하고 5분 동안 유휴 상태이면 자동으로 중단되어 롤백된다.
   - 이 트랜잭션이 같은 테이블을 수정하지 못하게 차단하지 않고 4시간이 지나면 자동으로 중단되고 롤백됨
   - 실행 중인 트랜잭션을 명시적으로 중단하려면 트랜잭션을 시작한 사용자나 계정관리자가 SYSTEM%ABORT_TRANSACTION을 호출 한다.
   - 기본적으로 트랜잭션의 커밋 또는 롤백을 막으면, 그 트랜잭션이 보유한 리소스에 대한 잠금과 함께 트랜잭션이 분리된 상태로 남는다.
2. 각 트랜잭션은 독립된 범위를 가짐
3. **Snowflake는 중첩된 트랜잭션을 지원하지 않지만, 중첩된 프로시저는 지원한다.**

chapter 17 예상문제

What object will you use to schedule a **merge** statement in Snowflake so that it runs every hour? 1

1. Task
2. Stream
3. Pipe
4. Table

Select the true statements about Snowflake tasks: 1,3

1. A task can execute a single SQL Statement
2. A task can execute multiple SQL Statements
3. A task can execute a call to a Stored Procedure
4. A task can execute a function

A task is still being executed before the next scheduled task. What is going to happen with the new scheduled task? 2

1. Snowflake will abort it
2. Snowflake will skip it
3. Snowflake will wait for the previous task to complete

Which roles or privileges are required to view TASK_HISTORY? 1 2 5

1. AccountAdmin
2. MONITOR EXECUTION privilege
3. SysAdmin
4. SecurityAdmin
5. Task owner (OWNERSHIP privilege)

What will happen to the child task if you remove ALL its predecessors? 2, 3

1. The child task is removed from the system
2. The child task may become the root task
3. The child task may become a standalone task

A task went into a loop. How long will the task run before Snowflake finishes it? 2

1. 15 minutes
2. 30 minutes
3. 60 minutes
4. 4 hours

the default time of task duration is 60 min

The owner of a task (the one with the OWNERSHIP privilege) is deleted. What will happen to the task? 1

1. The task will belong to the role that dropped the owner's role
2. The task is deleted
3. The task is suspended

If a transaction disconnects and goes into a detached state, which cannot be committed or rolled back, how long will Snowflake take to abort the transaction? 3

1. 15 minutes
2. 60 minutes
3. 4 hours
4. 12 hours

Does Snowflake support Nested Transactions? 2

1. True
2. False

+

CHAPTER 18 Streams

스트림은 Snowflake object로서 DML(삽입,업데이트,삭제)에 대한 변경사항과 각 변경사항에 대한 메타데이터를 기록한다.

- 이 기록된 데이터를 통해 다양한 작업 수행 가능
- 스트림은 테이블의 데이터를 저장하는 것이 아닌 저장한 위치(store offset)을 저장함.
- 스트림 생성 시 그 테이블에 대한 현재 트랜잭션 버전으로 해당 생성 시점(**offset**)으로 초기화하여 각 모든 행에 대한 스냅샷(time-travel)을 논리적으로 생성함. 그 후 변경사항이 기록
- 행에 대한 변경사항을 저장하는 또 다른 객체라고 생각하자.
- 처음 생성되었을 때 테이블의 스냅샷과 확인하려고 하는 시간대의 스탭샷과 비교하여 스트림 내용을 출력한다.
- 스트림 생성 시 원본 테이블에 숨겨진 열이 여러개 추가됨
- 스트림 생성 시 테이블 속성에 Change_tracking이 true로 된다.

스트림에는 3가지 종류가 있다.

1. Standard : 모든 DML 작업에 대한 변경사항을 저장 (insert, update, delete). 또한 테이블, 디렉토리 테이블, 그리고 뷰에 대한 스트림 생성이 허용됨
2. Append Only : 행 삽입에 대해서만 추적함. 또한 테이블, 디렉토리 테이블, 그리고 뷰에 대한 스트림 생성이 허용됨
3. Insert Only : 행 삽입에 대해서만 추적함 (Insert와 다른 점은 이것은 오직 외부 테이블에 대해서만 스트림 생성이 허용됨.)

스트림에는 공통적으로 다음 column이 포함되어 있다.

1. METADATA$ACTION : DML 연산들을 나타낸다. (Insert,delete)
2. METADATA$ISUPDATE : 그 작업이 UPDATE 문의 일부인지 나타냄
3. METADATA$ROW_ID : 해당 행의 고유하고 불변한 ID를 나타낸다

※ UPDATE는 DELETE/INSERT로 나뉘어서 저장된다.

추가로 기억해야할 함수

SYSTEM$STREAM_HAS_DATA : 스트림에 CDC(Change Data Capture) 레코드가 포함되어 있는지의 여부를 나타낸다.

- 한번 테이블이 드롭되면 스트림이 비활성화 돼서 에러를 일으키는 것 같음

chapter 18 예상문제

Which object in Snowflake stores DML change made to tables and metadata about each change?

1. Tables 2. Pipes 3. Table Streams 4. Account Streams

What statements are true about streams?

1. A Stream itself does NOT contain any table data.
2. A stream only stores the offset for the source table
3. The hidden columns used by a stream consume storage

Which are the additional columns that the streams create? 1,3,4

1. METADATA$ACTION
2. METADATA$ISREAD
3. METADATA$ISUPDATE
4. METADATA$ROW_ID
5. METADATA$COLUMN_ID

What different types of streams exist in Snowflake? 1,2,4

1. Standard
2. Append-only
3. Update-only
4. Insert-only

Do streams ensure exactly-once semantics for new or changed data in a table? 1

1. True 2. False

Are Insert-only streams only supported on external tables?

1. True 2. False

A stream has been created on a table. A row is inserted in the table, and it's updated later. Will the stream capture both events?

1. True 2. False

Can we use streams in materialized views?

1. True 2. False

Can multiple streams be created for the same table?

1. True 2. False

CHAPTER 19 File Formats & Sequences

파일 포맷은 오브젝트로서 스노우플레이크로 데이터를 로드하기 위해 필요한 포맷에 대한 정보를 나타내고 저장한다.

- 각 매개변수를 달리해서 파일 포맷을 정할 수 있다(File's delimiter나 Skip_hedder 등)
- Snowflake는 기본적으로 Structured data와 Semi-Structured data 모두 지원함. (정형 비정형)
- 즉 정형에는 CSV 파일같은 것이 있다. (로드/언로드가 가장 빠름)
- 비정형에는 JSON, Parquet, XML, Avro, ORC가 있다.
- 비정형 데이터는 최대크기가 16MB로 제한 되어 있고, **JSON 표기법을 통해 쿼리 가능**

Format	Type	Load	Unload	Binary format
CSV	Structured	Yes	Yes	No
JSON	Semi-structured	Yes	Yes	No
Parquet	Semi-structured	Yes	Yes	Yes
XML	Semi-structured	Yes	No	No
AVRO	Semi-structured	Yes	No	Yes
ORC	Semi-structured	Yes	No	Yes

- FLATTEN 함수를 통해 반정형 데이터를 관계형 표현식으로 변환할 수 있음.
- FLATTEN을 통해 Variant, Object, or Array column과 같은 열들을 측면뷰로 변환 (String화 된 뷰처럼 나타냄)

- 기본적으로 반정형 데이터는 VARIANT 데이터 타입으로 들어간다. 바이너리 포맷이 되면 BINARY 데이터 타입도 되는 듯. CSV는 VARCHAR이나 STRING 타입

SEQUENCES

시퀀스는 세션 및 문장 전반에 걸쳐 고유한 숫자 값을 생성한다.

- 시퀀스를 통해 Primary key를 만들거나 고유한 값을 갖는 열을 생성할 수 있다.
- 시퀀스를 생성하기 위해서는 시작 값과 Interval을 정해야한다.
- 또한 쿼리 표현식을 통해 시퀀스도 사용가능하다. (**'nextval'** 함수 사용)
- 시퀀스 PEOPLE_SEQ 생성 후 쿼리 삽입시 PEOPLE_SEQ.nextval 하면 다음 숫자가 들어가게 된다.
- 혹은 테이블 생성 시 컬럼명을 지정할 때 Default PEOPLE_SEQ.nextval하면 모든 행에 시퀀스에 맞게 숫자가 삽입된다.

chapter 19 예상문제

Does Snowflake allow only the load of structure data? 2

1. True 2. False

Which of the following file formats are supported by Snowflake? 1,2,4

1. CSV 2. XML 3. TXT 4. Avro

How will you store JSON data in Snowflake? 1,3

1. Using a column of the JSON type
2. Using a column of the VARCHAR type
3. Using a column of the VARIANT type
4. Using a column of the NULL type

Which of the following object types are stored within a schema? 1,2,3,4,7

1. Tables 2. Views 3. File Formats 4. UDFs
5. Roles 6. Users 7. Sequences

You have the following data in a variant column from the table "myTable". How can you query the favorite technology that Gonzalo uses? 2

```
{
  "name": "Gonzalo",
  "country": "Spain",
  "favouriteTechnology": Snowflake,
  "hobbies": [
    {
      "name": "soccer",
      "since": "2000",
    },{
      "name": "music",
      "since": "2005",
    },{
      "name": "technology",
      "since": "1996",
    }
  ]
}
```

1. SELECT favouriteTechnology FROM myTable;
2. SELECT src:favouriteTechnology FROM myTable;
3. SELECT src:$favouriteTechnology FROM myTable;
4. SELECT CONVERT_JSON(src:favouriteTechnology) FROM myTable;

Which table function allows you to convert semi-structured data to a relational representation?

1. FLATTEN 2. CHECK_JSON 3. PARSE_JSON

CHAPTER 20 Ecosystem

Snowflake는 비즈니스 파트너와 함께 평가 계정을 쉽게 만들고 통합혀여 타사 도구와 서비스를 이용가능함. 파트너에는 2가지 유형이 있음

1. Technology partners : 솔루션을 Snowflake와 통합하여 데이터를 신속하게 확보하고 Snowflake로 빠르게 가져옴 (Software, driver, interface 제공)
2. Solution partners : 신뢰가능하고 검증된 전문가와 서비스. 컨설팅 파트너와 같음

파트너들은 자세히는 6가지로 분류됨.

1. Data Integratipon
2. ML & Data Science\
3. Security & Governance
4. Business Intelligence
5. SQL Editors
6. Programming Interfaces

Snowflake 규정준수(Compliance)
전부는 아니지만 다음과 같은 규정에 대해 준수함

1. HITRUST/HIPAA
2. ISO/IEC 27001
3. FedRAMP Moderate
4. PCI-DSS
5. SOC 2 Type II
6. SOC 1 Type II
7. GxP

SNOWFLAKE DATA MARKETPLACE에서 데이터셋을 구매하거나 팔수 있음
- 마켓플레이스에서 3가지 유형의 리스팅 사용 가능
1.Free Listing : Standard Listing. 일반적이고 집겨된 또는 고객별이 아닌 데이터를 제공하는데 적합
2. Personalized Listing : 프리미엄 데이터로서, 공급자에게 특정 데이터 셋을 **요청** 가능
3. Paid Listing : 공급자로서, 소비자들에게 자신의 리스팅을 사용할 수 있게 함.
※ 또, Private Listing 이라는 것이 있지만, 이는 마켓 플레시으셍서 사용할 수 없지만, 다른 계정과 공유하여 사용하고 싶을 때 사용가능

만약 데이터셋을 올린다고 한다면 그 데이터 셋이 다음을 충족해야만 한다.

1. Fresh data (최신)
2. Non-static data (비정적)
3. Real data (실제_
4. Compliant data (규정준수)
5. Legally Distributable data (합법적 배포가능)

Snowflake는 Column 수준의 보안을 제공한다 (Enterprise editioN 이상)

1. Dynamic Data masking : 쿼리 시 데이터를 마스킹하는 열 수준 보안 기능. 역할이 낮은 사용자를 위해 열 내용의 일부분을 숨길 수 있음
2. External Tokenization : 민감한 데이터를 snowflake에 로드하기 전에 토근화할 수 있음. 마스킹 정책을 통해 토큰을 해제할수도 있음. 비밀번호나 민감한 데이터에 유용

chapter 20 에상문제
Which certifications are compliant with Snowflake?

1. HIPAA
2. PCI-DSS
3. FedRAMP

Can Snowflake perform transformations after loading files staged by partner software (ELT)? 1

Column-level security in Snowflake allows the application of a masking policy to a column within a table or view. Which two features are related to column-level security? 1 3

1. Dynamic Data Masking
2. Lock Databases
3. External Tokenization

What is common between Fivetran, Informatica, Stitch, and Talend?

1. They are Snowflake programming interfaces partners
2. They are Snowflake data integration partners
3. They are Snowflake security partners