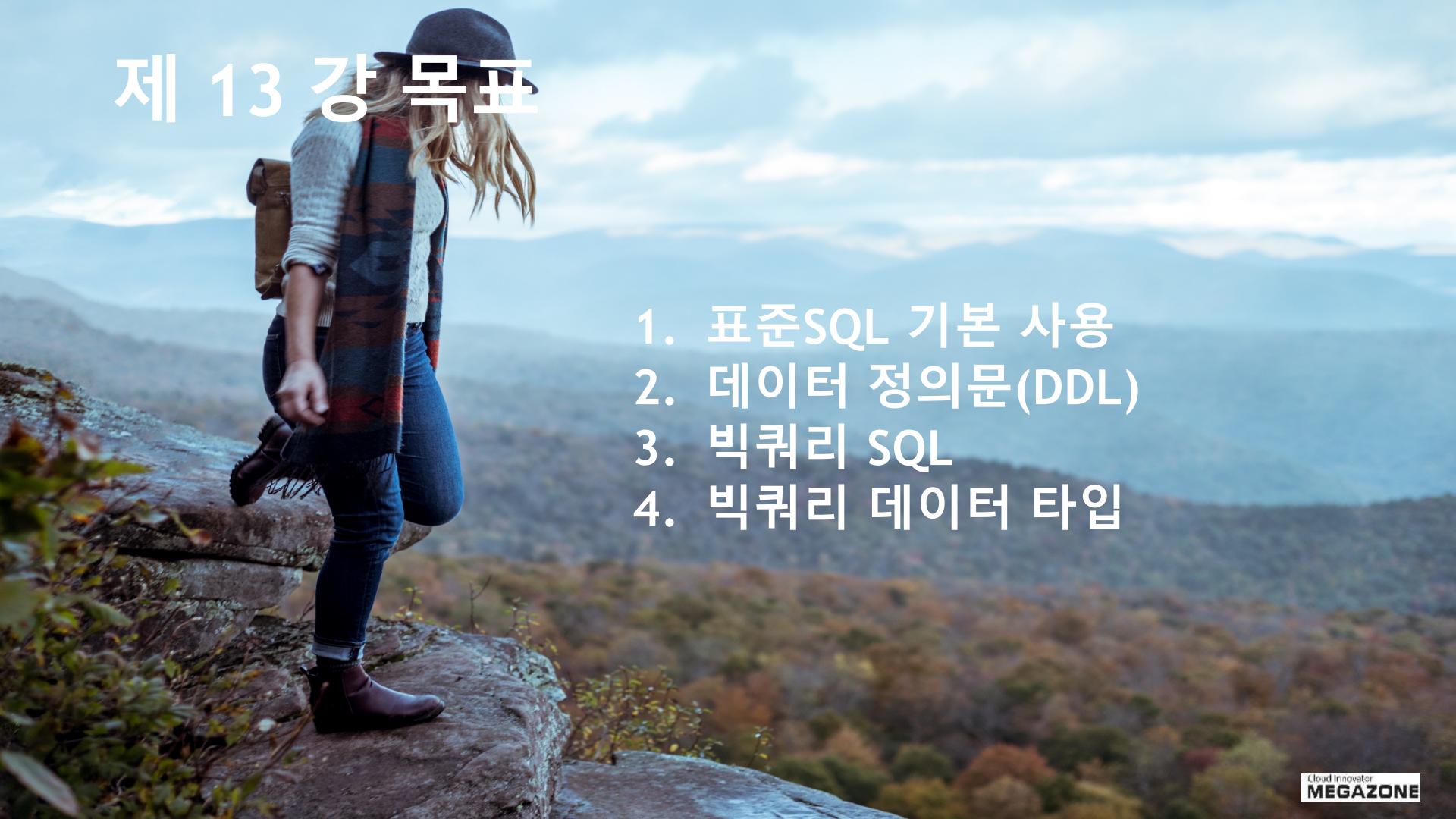




# 빅쿼리를 활용한 빅데이터 분석 (13)

서진호

# 제 13 강 목표

- 
- A photograph of a woman from behind, standing on a rocky outcrop. She is wearing a grey sweater, blue jeans, and a patterned scarf, and has a backpack. She is looking out over a misty, mountainous landscape under a cloudy sky.
1. 표준SQL 기본 사용
  2. 데이터 정의문(DDL)
  3. 빅쿼리 SQL
  4. 빅쿼리 데이터 타입

# 구조화된 쿼리 언어(SQL)

- SQL은 프로그래밍에 사용하며 관계형 데이터베이스 관리 시스템(RDBMS)에 보관된 데이터를 관리하기 위해 설계된 도메인별 언어
- 분산형 또는 인메모리 형태의 관계형 데이터 스트림 관리 시스템(RDSMS)의 스트림 처리용으로 엔터티와 변수 간의 관계를 통합하는 정형(Structured) 데이터에도 사용된다. Ex) Windows 함수
- SQL은 1986년 ANSI(American National Standards Institute)의 표준이 되었고 1987년에는 ISO(International Organization for Standardization)의 표준이 되었다.
- 현재 빅쿼리에서는 ANSI:2011 표준 SQL 지원



# SQL 문법의 구성 예

```
UPDATE clause {UPDATE country  
SET clause   {SET population =  $\overbrace{\text{population} + 1}$   
WHERE clause  {WHERE  $\underbrace{\text{name} = \overbrace{\text{'USA'}}}$ ;  
                           ^ predicate}}}} statement
```

- 문(Statement)
- 절(Clauses)
- 표현식(Expressions)
- 술어(Predicates)
- 쿼리(Queries)
- SQL 문장 끝은 세미콜론(";" ) 문 종결함.



발췌: <https://en.wikipedia.org/wiki/SQL>

# 구글 표준 SQL

- 데이터 쿼리 언어(DQL)문 또는 쿼리문
- 절차적 언어 문(저장 프로시저, Stored Procedure)
- 데이터 정의 언어(DDL)문
- 데이터 조작 언어(DML)문
- 데이터 제어 언어(DCL)문
- 트랜잭션 제어 언어(TCL) 문
- 기타 문



# 데이터 정의 언어(DDL)

- 데이터 정의 언어(DDL) 문을 사용하면 표준(Standard) SQL 쿼리 구문으로 BigQuery 리소스를 만들고 수정할 수 있음.
- DDL 명령어를 사용하여 테이블, 테이블 클론, 테이블 스냅샷, 뷰, 사용자 정의 함수(UDF), 행 수준 액세스 정책과 같은 리소스를 생성, 변경, 삭제 할 수 있음.
- 필수권한: `bigquery.jobs.create`
- IAM 역할:  
`bigquery.user`, `bigquery.jobUser`, `bigquery.admin`



# DDL - CREATE SCHEMA 문

- SCHEMA: 테이블, 뷰, 기타 리소스의 논리적 모음
- 빅쿼리에서는 Dataset 과 동일한 개념으로 사용함.
- Schema 는 빅쿼리 테이블 스키마를 나타내지 않음.

```
CREATE SCHEMA [ IF NOT EXISTS ]  
'project_name.dataset_name' [DEFAULT  
COLLATE collate specification]  
[OPTIONS(schema option list)]
```



# DDL - CREATE TABLE 문

CREATE [ OR REPLACE ] [ TEMP | TEMPORARY ]  
TABLE [ IF NOT EXISTS ]

table name

[(  
column[, ...]  
)]

[DEFAULT COLLATE collate specification]  
[PARTITION BY partition expression] [CLUSTER  
BY clustering column list]  
[OPTIONS(table option list)]  
[AS query\_statement]



# DDL - CREATE TABLE 세부정보

- CREATE 문 하나만 허용
- 중복 열 이름은 허용되지 않음
- LIKE와 as query\_statement 절이 모두 있는 경우, 쿼리 문의 열 목록은 LIKE 절에서 참조하는 테이블의 열과 일치해야 함.
- 세부 사항
  - 쿼리 결과에서 수집 시간으로 파티션을 나눈 테이블을 만들 수 없음
  - 대신 CREATE TABLE DDL 문을 사용하여 테이블을 만든 후 INSERT DML 문을 사용하여 데이터를 삽입함.
  - OR REPLACE 한정자를 사용하여 테이블을 다른 종류의 파티션 나누기로 대체할 수 없음.
  - 대신 테이블을 DROP한 후 CREATE TABLE ... AS SELECT ... 문을 사용하여 다시 생성함.



# DDL - CREATE TABLE ↴ Column

```
column :=  
    column_name column_schema  
  
column_schema :=  
{  
    simple_type [NOT NULL]  
    | STRUCT<field_list> [NOT NULL]  
    | ARRAY<array_element_schema>  
}  
[OPTIONS(column_option_list)]  
  
field_list :=  
    field_name column_schema [, ...]  
  
array_element_schema :=  
{ simple_type | STRUCT<field_list> }  
[NOT NULL]  
  
simple_type :=  
{ data_type | STRING COLLATE collate_specification }
```



# 실습1 – CREATE SHEMA 문

