

코멘토 실무PT 클래스

개발자 없이 시작하는 데이터 분석 - 비개발자를 위한 SQL

Chapter.

1. 지난시간

1주차

2주차

3주차

4주차

5주차

6주차

22.10.30(일) 20:00~23:00

4주차

고객의 첫 주문이 알고
싶어요!

01 3주차 세션 회고 및 과제 피드백

02 조건문(CASE), UNION, EXCEPT

03 현업에서 자주 사용하는 함수 설명
- 내재함수 (TYPE, DESCRIBE, EXPLAIN)
- 날짜함수 (날짜 변환(e.g. 요일추출), 날짜 연산)
- 문자함수 (숫자를 문자로)
- 수학함수 (통계, 제공)

04 고급함수 맛보기
- WINDOW 분석함수 (리텐션, 코호트)
- 정규표현식(휴대폰번호 추출)
- UDF(사용자 정의 함수)

05 실시간 Q&A

과제 함수 & WINDOW 실습
- 유저의 첫 구매를 구하고, 주차별 리텐션율을 구해보기

1주차

2주차

3주차

4주차

5주차

6주차

22.11.06(일) 20:00~23:00

5주차

대시보드란

무엇인가요

01 4주차 세션 회고 및 과제 피드백

02 대시보드의 이해
- 대시보드의 목적, 종류

오늘은 문제 푸는 시간을 가져보도록 하겠습니다.

- 시각화 기초, 목적 별 차트 사용 방법

04 시각화 데이터 적용

- 구글 스튜디오(or 구글 시트)에서 대시보드 만들기 실습

05 실시간 Q&A

과제 구글 시트로 간단한 대시보드 제작하기

- (일/주/월)별 지표 트렌드
- (시간/요일)별 지표 집계

Chapter.

1. 과제피드백

Q1. 정류소별로 가장 마지막에 빌려진
자전거의 대여시각을 구하는 쿼리를 작성.

1번 정답.

-- Q1. 정류소별로 가장 마지막에 빌려진 자전거의 대여시각 을 구하는 쿼리를 작성.

```
WITH result_t AS (
```

```
SELECT
```

```
  r.rent_station_id,
```

```
  r.rent_at,
```

```
  ROW_NUMBER() OVER(PARTITION BY r.rent_station_id ORDER BY r.rent_at DESC) AS rn
```

```
FROM rental_history r
```

```
)
```

```
SELECT *
```

```
FROM result_t
```

```
WHERE rn = 1 -- 가장 마지막에 빌렸을때의 대여시각
```

Q2. “요일별” 대여횟수를 구하는 쿼리를 구해주세요.
요일 은 월,화,수,목,금,토,일 같이 한글로 표현해주세요.

2번 정답.

-- Q2. “요일별” 대여횟수를 구하는 쿼리를 구해주세요.
-- 요일 은 월,화,수,목,금,토,일 같이 한글로 표현해주세요.

```
WITH result_t as (  
SELECT  
  st.name AS station_name,  
  rh.rent_at -- 2015-09-01(월), 2015-09-02(화), 2015-09-06(금), .....  
FROM rental_history rh -- 대여이력  
JOIN station st  
  ON rh.rent_station_id = st.station_id  
)
```

```
-- 요일을 구하는 함수 : https://www.sqlite.org/lang_datefunc.html  
SELECT  
CASE  
  WHEN strftime('%w',rent_at) = '0' then "0일"  
  WHEN strftime('%w',rent_at) = '1' then "1월"  
  WHEN strftime('%w',rent_at) = '2' then "2화"  
  WHEN strftime('%w',rent_at) = '3' then "3수"  
  WHEN strftime('%w',rent_at) = '4' then "4목"  
  WHEN strftime('%w',rent_at) = '5' then "5금"  
  WHEN strftime('%w',rent_at) = '6' then "6토"  
END AS day,  
COUNT(*) AS rent_count  
FROM result_t
```

	day	rent_count
1	0월	322697
2	1화	362275
3	2수	372368
4	3목	356135
5	4금	362229
6	5토	370835
7	6일	353461

데일리 퀴즈

<https://solvesql.com/problems/olympic-cities/>

```
SELECT
  year,
  UPPER(substring(city,1,3)) as city
FROM games
WHERE year >= 2000
ORDER BY 1 DESC
```

	year	city
1	2016	RIO
2	2014	SOC
3	2012	LON
4	2010	VAN
5	2008	BEI
6	2006	TOR
7	2004	ATH
8	2002	SAL
9	2000	SYD

데일리 퀴즈

	Region	Furniture	Office Supplies	Technology
1	Central	124	314	108
2	East	156	354	158
3	South	89	204	83
4	West	195	400	177

<https://solvesql.com/problems/characteristics-of-orders/>

SELECT

region AS "Region",

COUNT(DISTINCT CASE WHEN category= 'Furniture' THEN order_id END) AS Furniture,

COUNT(DISTINCT CASE WHEN category= 'Office Supplies' THEN order_id END) AS "Office
Supplies",

COUNT(DISTINCT CASE WHEN category= 'Technology' THEN order_id END) AS Technology

FROM records

GROUP BY 1

ORDER BY 1

Chapter.

에러Errors를 읽는 방법

에러Error의 종류

모든 에러를 다 다루기는 어렵습니다.

우리가 보통 접하는 에러들의 종류들을 짧게 설명드립니다.

제일 많이 접하는 오류는 “**문법 오류Syntax**”입니다.

말그대로 틀린 문법을 적은 경우 입니다. 함수를 잘못쓰거나, 오타자를 내는 경우 발생합니다.

이런 경우 올바르게 문법을 사용하면 해결됩니다.

문법오류 예시

- near "SELECT": syntax error

SELECT 근처에서 오류가 발생했다고 알려준다.

```
3
4 WITH result_t as (
5 SELECT
6     st.name AS station_name,
7     rh.rent_at
8 FROM rental_history rh
9     JOIN station st
10    ON rh.rent_station_id = st.station_id
11
12
13 -- 요일을 구하는 함수 : https://www.sqlite.org/lang_datef
14 SELECT
15     CASE
```

▶ 실행

SQL 포매팅



SQL 쿼리 수행 중 오류가 발생했습니다!

near "SELECT": syntax error

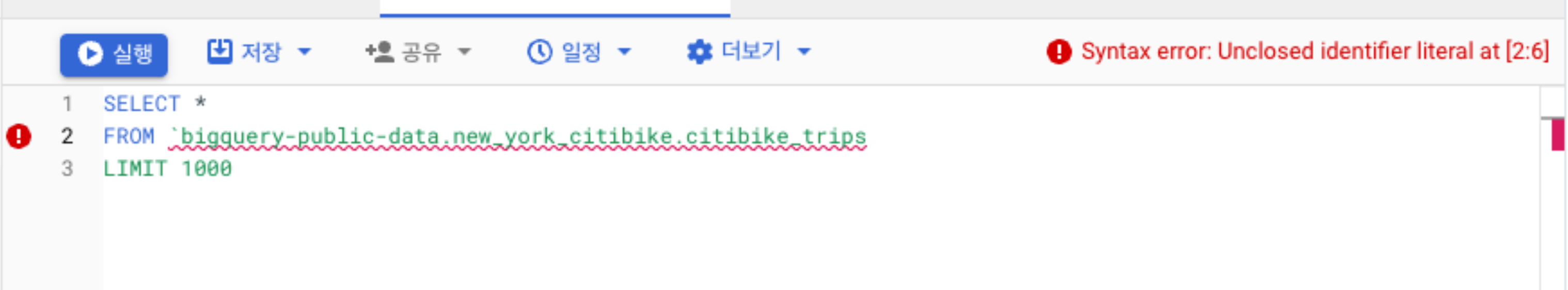
문법오류 예시

빅쿼리는 친절하게 어느 부분에서 오류가 발생했는지 상세하게 알려준다.

“Syntax error: Unclosed identifier literal at [2:6]”

2행에서 `가 닫혀지지 않아서 생긴 문제.

From `bigqur~~ trips`



The screenshot shows the Google Cloud BigQuery console interface. At the top, there is a toolbar with buttons for '실행' (Run), '저장' (Save), '공유' (Share), '일정' (Schedule), and '더보기' (More). A red error message is displayed at the top right: 'Syntax error: Unclosed identifier literal at [2:6]'. Below the toolbar, the SQL query is displayed in a text area:

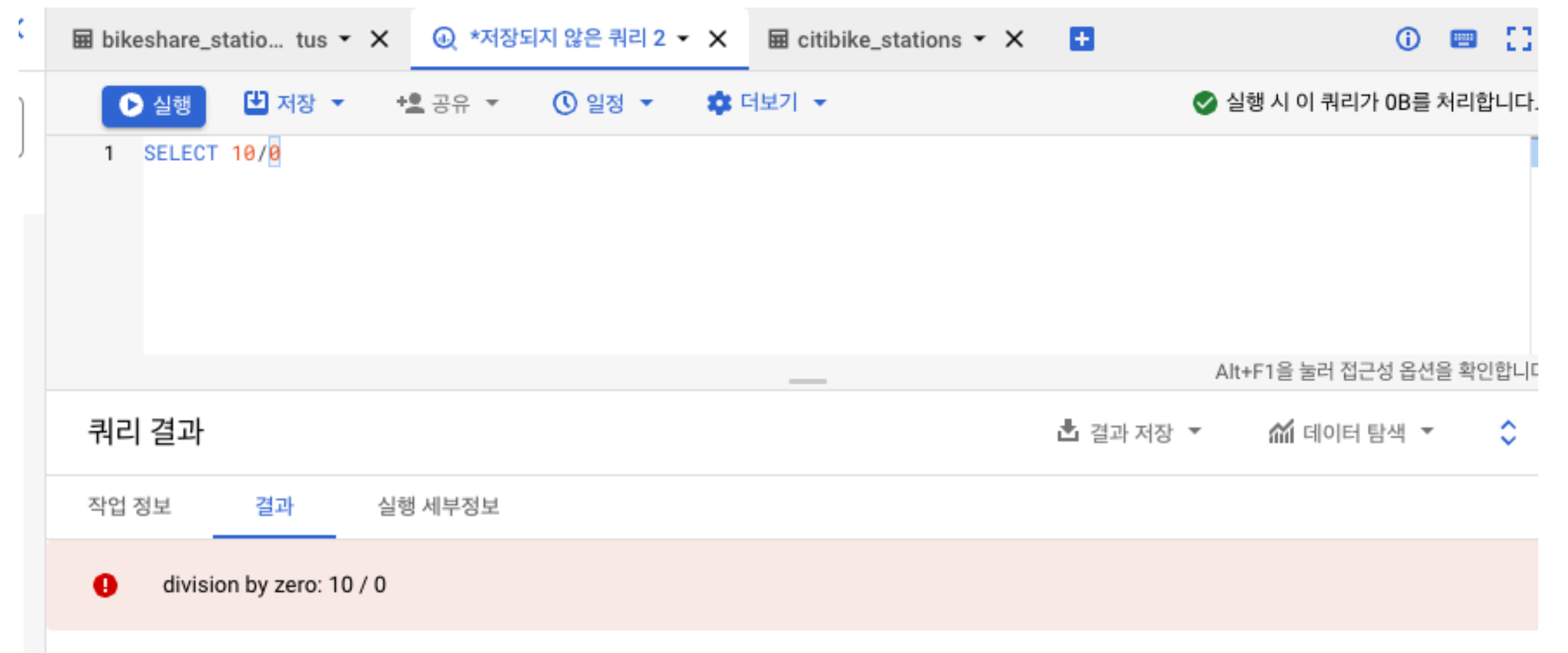
```
1 SELECT *
2 FROM `bigquery-public-data.new_york_citibike.citibike_trips`
3 LIMIT 1000
```

The error is highlighted with a red squiggly line under the backtick at the end of the table name in line 2.

연산오류 예시

division by zero :

0으로 나누는 연산은 문법적으로는 문제는 없으나
결과값을 내놓을 수 없기 때문에 오류가 발생.



SQL Exception 예외처리.

빅쿼리의 경우 SAFE를 붙이면 “어느정도” 해결됩니다.
다만, 남발하면 안됩니다.

문제가 생겼는데, “예외”로 봐주자.



The screenshot shows a BigQuery query editor with the following SQL code:

```
1 SELECT  
2 | SAFE_DIVIDE(10, 0)
```

Below the query editor, the results are displayed under the heading "쿼리 결과". There are three tabs: "작업 정보", "결과" (selected), and "JSON". The results table has three columns: "행" (Row), "f0_" (Function 0), and an empty column. The first row shows the value "1" in the "행" column and "null" in the "f0_" column.

행	f0_	
1	null	

Chapter.

문제 은행

먼저, 남아있는 문제를 같이 풀어봅시다.
<https://solvesql.com/problems/>

코딩테스트 문제를 가져와봤습니다.
같이 풀어보실까요.

1. 변환함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/conversion_functions?hl=ko

2. 날짜함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/date_functions?hl=ko

3. 수학함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/mathematical_functions?hl=ko

4. 문자열 함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/string_functions?hl=ko

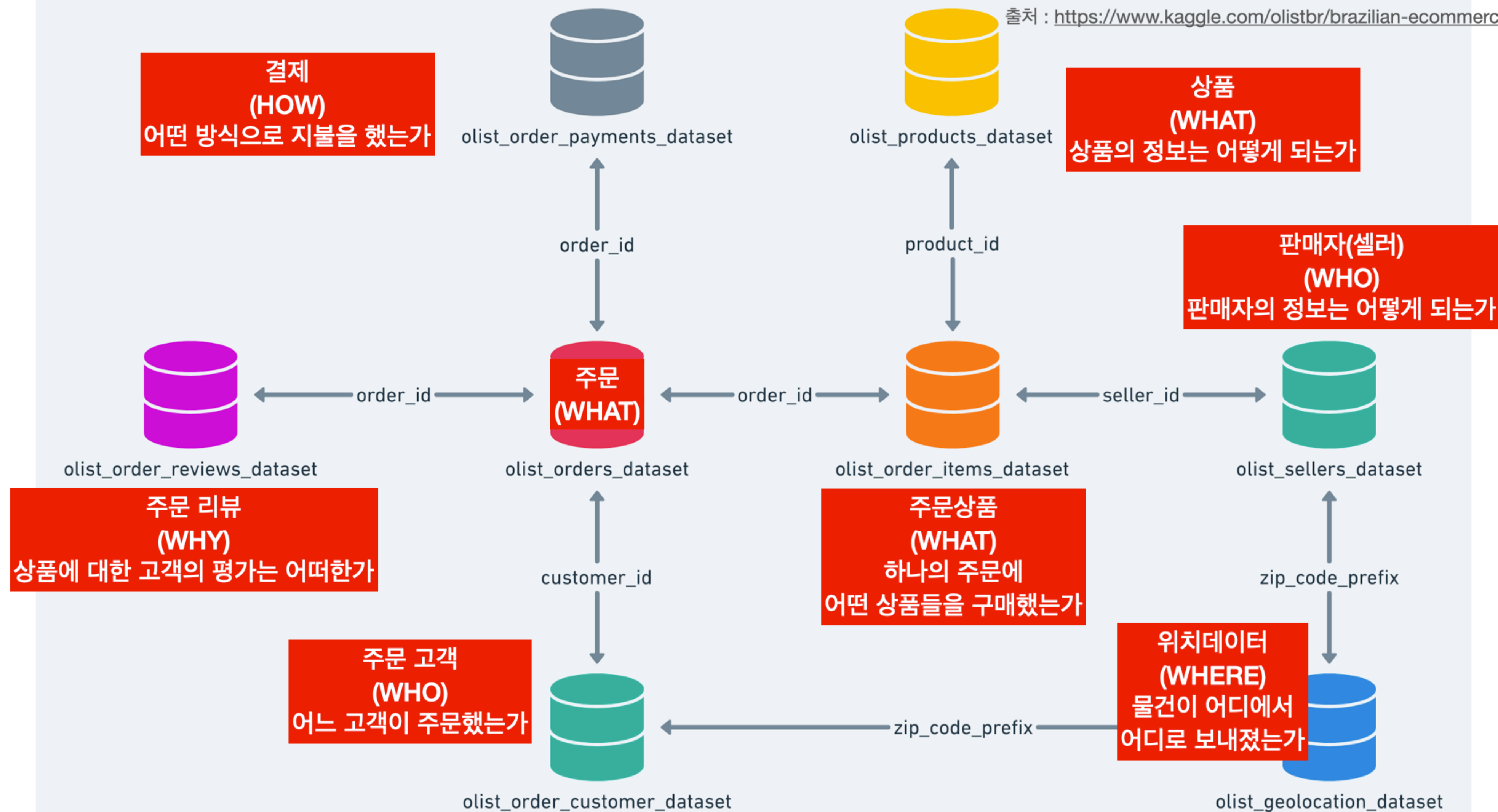
탐색 함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/navigation_functions?hl=ko

윈도우 함수 : <https://cloud.google.com/bigquery/docs/reference/standard-sql/window-function-calls?hl=ko>

번호 지정 함수 : https://cloud.google.com/bigquery/docs/reference/standard-sql/numbering_functions

Q1. 정류소별로 가장 마지막에서 “2번째”로 빌려진
자전거의 대여시각을 구하는 쿼리를 작성.
(HINT : ROW_NUMBER() 를 사용합니다.)

출처 : <https://www.kaggle.com/olistbr/brazilian-ecommerce>



해커랭크 풀어봅시다.

[https://www.hackerrank.com/challenges/revising-the-select-
query/problem?isFullScreen=true](https://www.hackerrank.com/challenges/revising-the-select-query/problem?isFullScreen=true)

5주차 강의를 모두 들으시느라 고생하셨습니다!

궁금한 내용은 언제든지 문의주세요. :-)

이제 마지막 한 주차가 남았습니다!!

마지막까지 같이 나아가요!