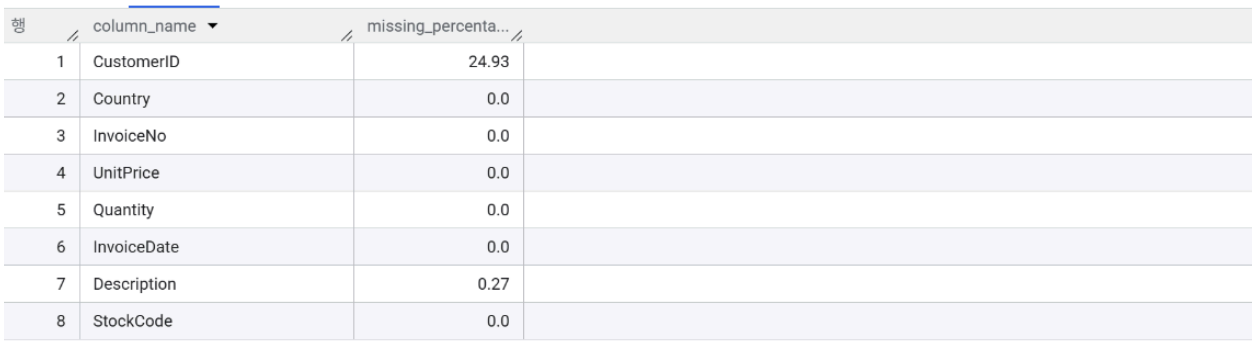
2025.07.08_고객 세그먼테이션_PROJECT

11-4. 데이터 전처리(1): 결측치 제거

1. 컬럼별 결측치 비율

```
select *
from
(select
  'InvoiceNo'as column_name,round(sum(case when InvoiceNo IS NULL THEN 1 E
from eng-origin-464902-s3.modulabs_project.data union all
select
    'StockCode'as column_name,round(sum(case when StockCode IS NULL THEN
from eng-origin-464902-s3.modulabs_project.data union all
select
    'Description'as column_name,round(sum(case when Description IS NULL THE
from eng-origin-464902-s3.modulabs_project.data union all
select
   'Quantity'as column_name,round(sum(case when Quantity IS NULL THEN 1 ELS
   from eng-origin-464902-s3.modulabs_project.data union all
select
    'InvoiceDate'as column_name,round(sum(case when InvoiceDate IS NULL THE
  from eng-origin-464902-s3.modulabs_project.data union all
select
    'UnitPrice'as column_name,round(sum(case when UnitPrice IS NULL THEN 1 E
  from eng-origin-464902-s3.modulabs_project.data union all
select
    'CustomerID'as column_name,round(sum(case when CustomerID IS NULL THE
  from eng-origin-464902-s3.modulabs_project.data union all
select
    'Country'as column_name,round(sum(case when Country IS NULL THEN 1 ELS
  from eng-origin-464902-s3.modulabs_project.data
) AS column_data;
```



행	column_name	missing_percenta...
1	CustomerID	24.93
2	Country	0.0
3	InvoiceNo	0.0
4	UnitPrice	0.0
5	Quantity	0.0
6	InvoiceDate	0.0
7	Description	0.27
8	StockCode	0.0

2. 결측치 처리

결측치를 처리하기 위한 대표적인 방법은 결측치가 존재하는 **행을 삭제**하거나, **다른 값들의 평균, 중앙값, 최빈값 등으로 대체**

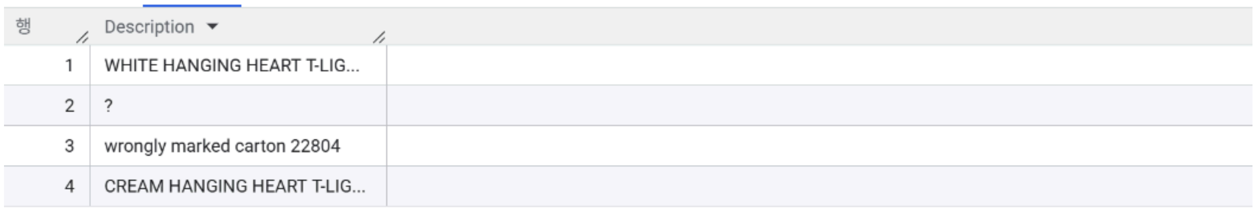
1) `CustomerID` (24.93%)

→고객 클러스터링에 매우 중요한 컬럼이나, 25%의 누락 값을 다른 값으로 대체하는 것은 분석에 높은 편향 및 노이즈 발생 가능

→ 누락된 `CustomerID`가 있는 행을 제거

2) `Description` (0.27%)

```
select distinct Description
from eng-origin-464902-s3.modulabs_project.data_2
where StockCode = '85123A';
```

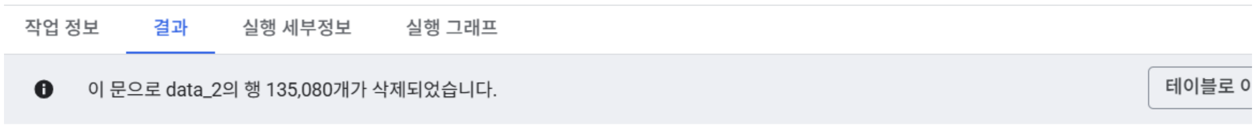


행	Description
1	WHITE HANGING HEART T-LIG...
2	?
3	wrongly marked carton 22804
4	CREAM HANGING HEART T-LIG...

→ 일관성 고려, 결측치 비율이 매우 작으므로 삭제

3. 결측치 삭제하기

```
DELETE from eng-origin-464902-s3.modulabs_project.data
where Description is null
or CustomerID  is null;
```



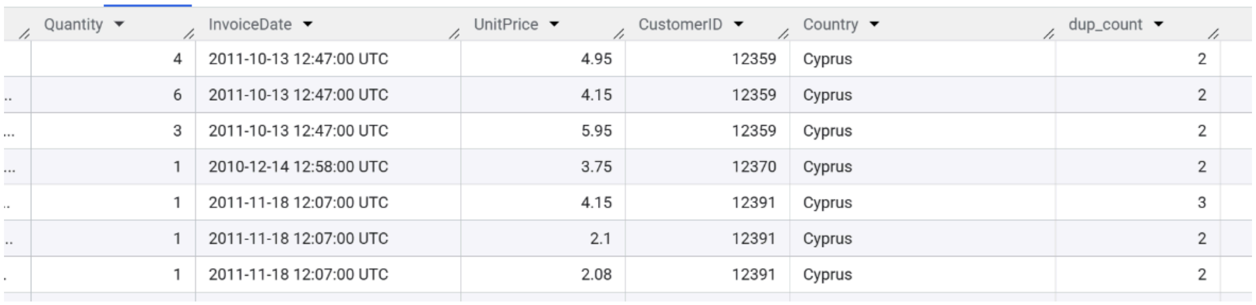
작업 정보 결과 실행 세부정보 실행 그래프

이 문으로 data_2의 행 135,080개가 삭제되었습니다.

11-5. 데이터 전처리(2): 중복값 처리

1. 중복값 확인

```
select *,
  COUNT(*) AS dup_count
from eng-origin-464902-s3.modulabs_project.data
group by InvoiceNo,StockCode, Description,Quantity, InvoiceDate, UnitPrice, Cus
having count(*) > 1;
```



Quantity	InvoiceDate	UnitPrice	CustomerID	Country	dup_count
4	2011-10-13 12:47:00 UTC	4.95	12359	Cyprus	2
6	2011-10-13 12:47:00 UTC	4.15	12359	Cyprus	2
3	2011-10-13 12:47:00 UTC	5.95	12359	Cyprus	2
1	2010-12-14 12:58:00 UTC	3.75	12370	Cyprus	2
1	2011-11-18 12:07:00 UTC	4.15	12391	Cyprus	3
1	2011-11-18 12:07:00 UTC	2.1	12391	Cyprus	2
1	2011-11-18 12:07:00 UTC	2.08	12391	Cyprus	2

2. 중복값 없는 데이터로 교체

```
CREATE OR REPLACE TABLE eng-origin-464902-s3.modulabs_project.data AS
SELECT DISTINCT *
```

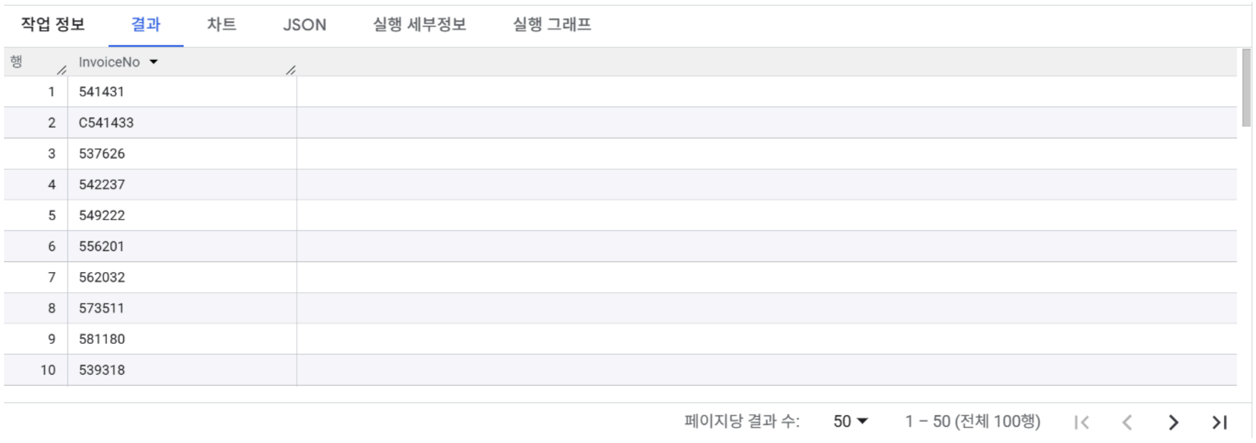
```
FROM eng-origin-464902-s3.modulabs_project.data;
```

11-6. 데이터 전처리(3): 오류값 처리

1. `InvoiceNo` 살펴보기

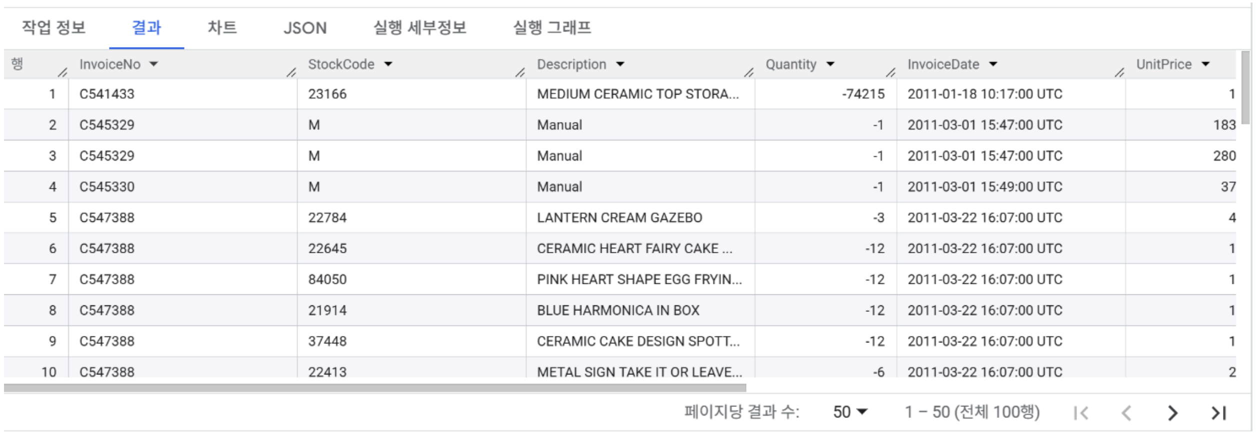
고유(unique)한 `InvoiceNo` 의 개수를 출력해 보세요.

```
select distinct InvoiceNo
FROM eng-origin-464902-s3.modulabs_project.data
limit 100;

select *
FROM eng-origin-464902-s3.modulabs_project.data
where InvoiceNo like "C%"
limit 100;

SELECT ROUND(SUM(CASE WHEN Quantity < 0 then 1 else 0 end)/ count(*)*100
FROM eng-origin-464902-s3.modulabs_project.data;
```



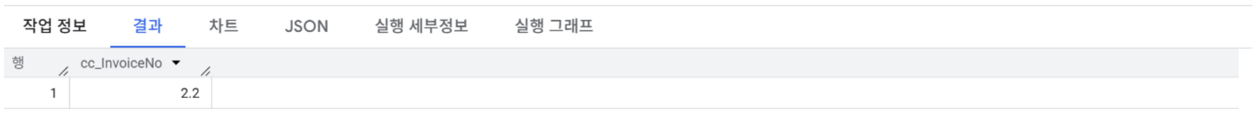
행	InvoiceNo
1	541431
2	C541433
3	537626
4	542237
5	549222
6	556201
7	562032
8	573511
9	581180
10	539318

페이지당 결과 수: 50 1 – 50 (전체 100행)


작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice
1	C541433	23166	MEDIUM CERAMIC TOP STORA...	-74215	2011-01-18 10:17:00 UTC	1
2	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	183
3	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	280
4	C545330	M	Manual	-1	2011-03-01 15:49:00 UTC	37
5	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4
6	C547388	22645	CERAMIC HEART FAIRY CAKE ...	-12	2011-03-22 16:07:00 UTC	1
7	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1
8	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1
9	C547388	37448	CERAMIC CAKE DESIGN SPOTT...	-12	2011-03-22 16:07:00 UTC	1
10	C547388	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2

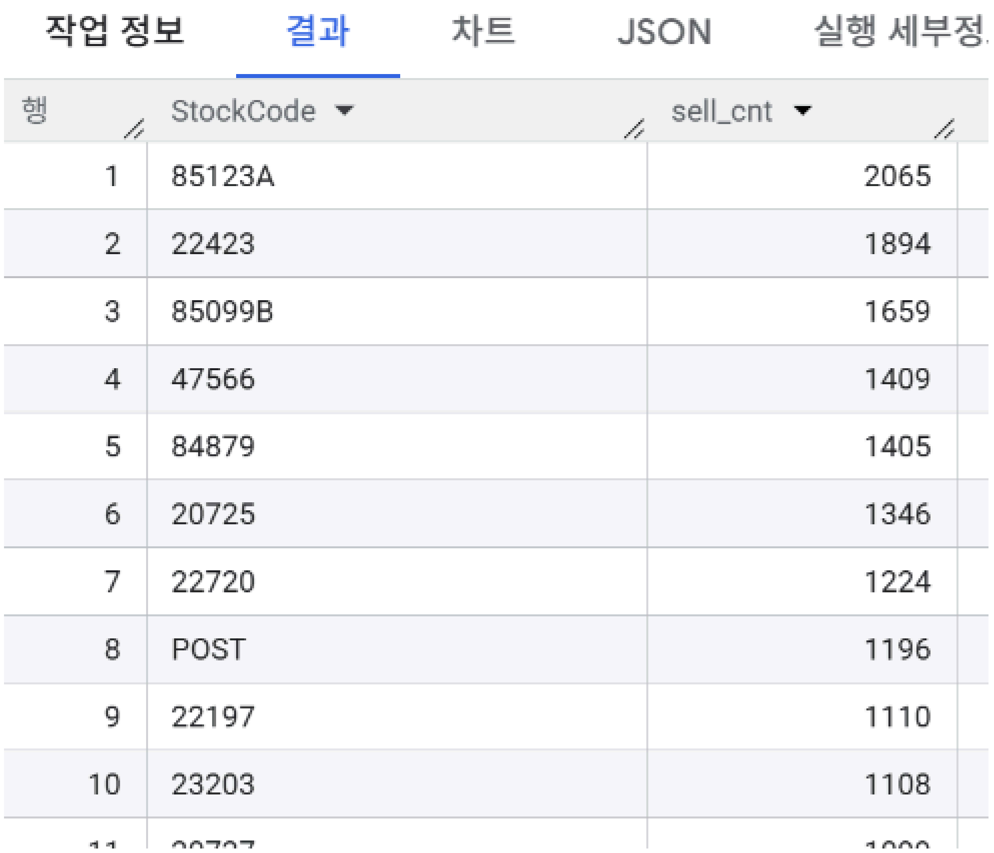
페이지당 결과 수: 50 1 – 50 (전체 100행)


작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

행	cc_InvoiceNo
1	2.2

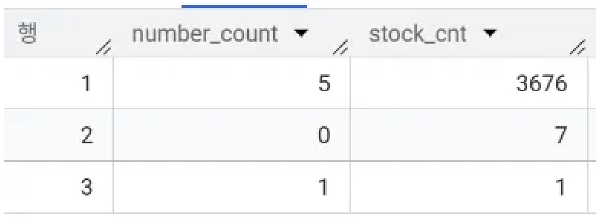
2. `StockCode` 살펴보기

```
select StockCode, count(*) as sell_cnt
from eng-origin-464902-s3.modulabs_project.data
group by 1
order by sell_cnt desc;
```

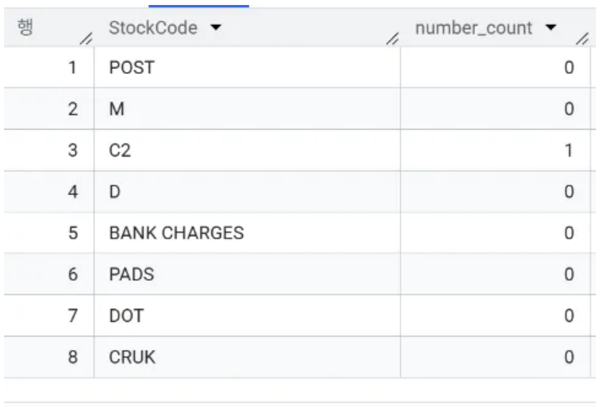


행	StockCode	sell_cnt
1	85123A	2065
2	22423	1894
3	85099B	1659
4	47566	1409
5	84879	1405
6	20725	1346
7	22720	1224
8	POST	1196
9	22197	1110
10	23203	1108

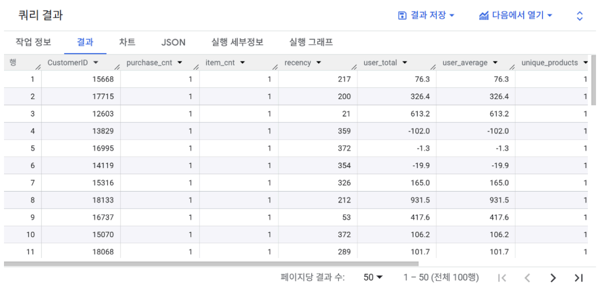
```
WITH UniqueStockCodes AS (
  SELECT DISTINCT StockCode
  FROM eng-origin-464902-s3.modulabs_project.data
)

select length(StockCode) - length(regexp_replace(StockCode, r'[0-9]','')) as num
  count(*) as stock_cnt
from UniqueStockCodes
group by number_count
ORDER BY stock_cnt DESC;

SELECT DISTINCT StockCode, number_count
FROM (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS
  FROM eng-origin-464902-s3.modulabs_project.data
)
WHERE number_count between 0 and 1;
```



행	number_count	stock_cnt
1	5	3676
2	0	7
3	1	1



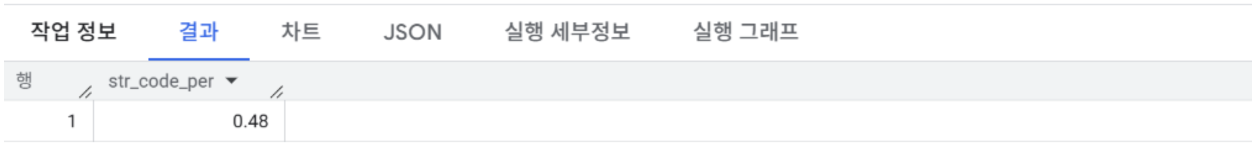
행	StockCode	number_count
1	POST	0
2	M	0
3	C2	1
4	D	0
5	BANK CHARGES	0
6	PADS	0
7	DOT	0
8	CRUK	0



행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products
1	15668	1	1	217	76.3	76.3	1
2	17715	1	1	200	326.4	326.4	1
3	12603	1	1	21	613.2	613.2	1
4	13829	1	1	359	-102.0	-102.0	1
5	16995	1	1	372	-1.3	-1.3	1
6	14119	1	1	354	-19.9	-19.9	1
7	15316	1	1	326	165.0	165.0	1
8	18133	1	1	212	931.5	931.5	1
9	16737	1	1	53	417.6	417.6	1
10	15070	1	1	372	106.2	106.2	1
11	18068	1	1	289	101.7	101.7	1

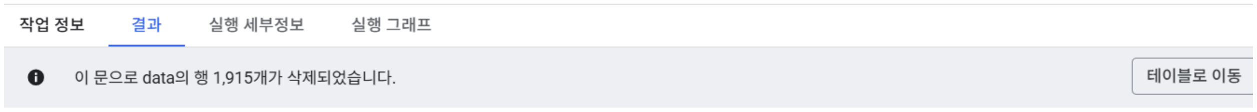
페이지당 결과 수: 50 1 – 50 (전체 100행)

```
SELECT round(sum(case when StockCode in ('POST', 'M', 'C2','D','BANK CHARG
FROM eng-origin-464902-s3.modulabs_project.data;
```



행	str_code_per
1	0.48

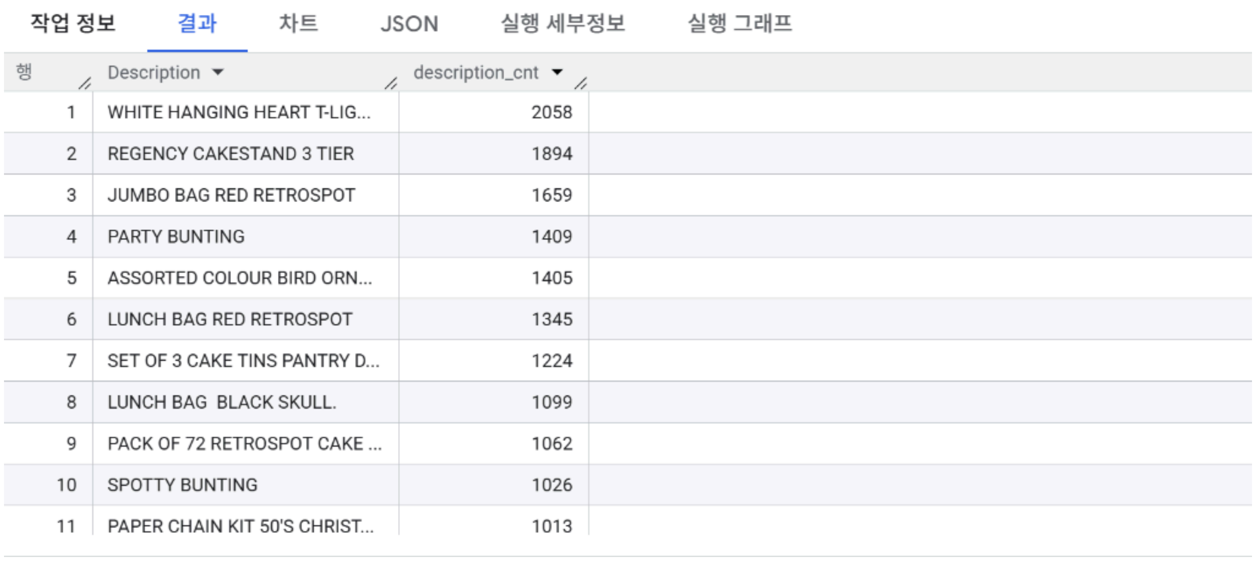
```
DELETE FROM eng-origin-464902-s3.modulabs_project.data
WHERE StockCode IN (
  SELECT DISTINCT StockCode
  FROM  (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS
  FROM eng-origin-464902-s3.modulabs_project.data)
  where number_count between 0 and 1);
```


작업 정보 결과 실행 세부정보 실행 그래프

이 문으로 data의 행 1,915개가 삭제되었습니다.

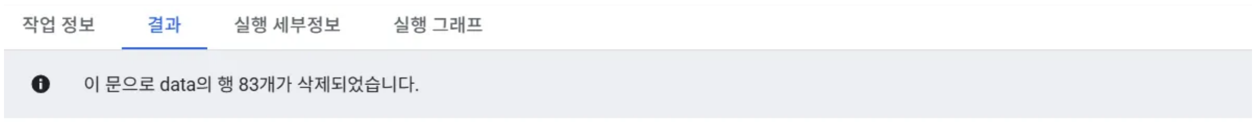
3. Description 살펴보기

```
SELECT Description, COUNT(*) AS description_cnt
FROM eng-origin-464902-s3.modulabs_project.data
group by Description
order by description_cnt desc
limit 30;
```



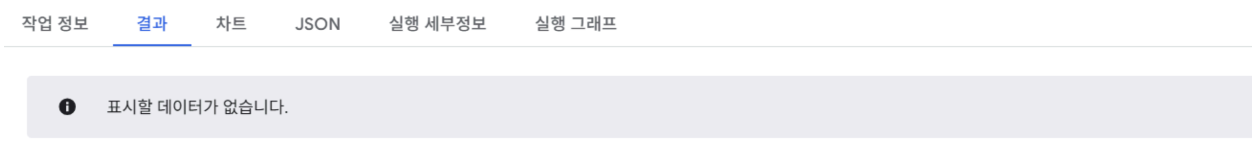
행	Description	description_cnt
1	WHITE HANGING HEART T-LIG...	2058
2	REGENCY CAKESTAND 3 TIER	1894
3	JUMBO BAG RED RETROSPOT	1659
4	PARTY BUNTING	1409
5	ASSORTED COLOUR BIRD ORN...	1405
6	LUNCH BAG RED RETROSPOT	1345
7	SET OF 3 CAKE TINS PANTRY D...	1224
8	LUNCH BAG BLACK SKULL.	1099
9	PACK OF 72 RETROSPOT CAKE ...	1062
10	SPOTTY BUNTING	1026
11	PAPER CHAIN KIT 50'S CHRIST...	1013

```
DELETE
FROM eng-origin-464902-s3.modulabs_project.data
WHERE Description in ('High Resolution Image','Next Day Carriage');
```



이 문으로 data의 행 83개가 삭제되었습니다.

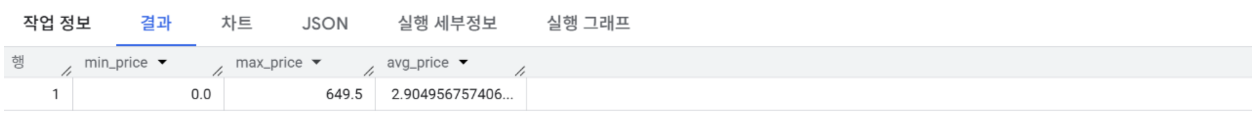
```
CREATE OR REPLACE TABLE eng-origin-464902-s3.modulabs_project.data AS
SELECT
  * EXCEPT (Description),
  upper(Description) AS Description
FROM eng-origin-464902-s3.modulabs_project.data;

SELECT DISTINCT Description
FROM eng-origin-464902-s3.modulabs_project.data
WHERE REGEXP_CONTAINS(Description, r'[a-z]');
```



표시할 데이터가 없습니다.

4. UnitPrice 살펴보기

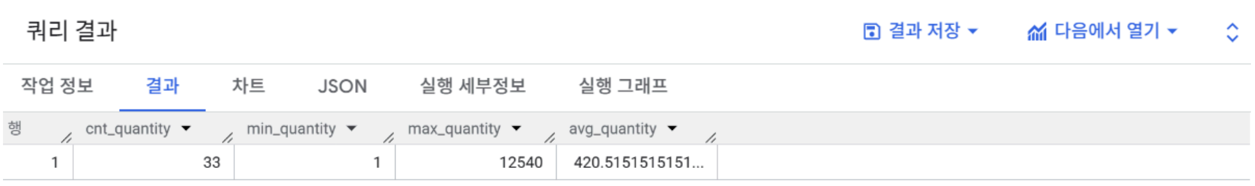
```
SELECT
min(UnitPrice) AS min_price
,max(UnitPrice) AS max_price
,avg(UnitPrice) AS avg_price
FROM eng-origin-464902-s3.modulabs_project.data;
```



행	min_price	max_price	avg_price
1	0.0	649.5	2.904956757406...

단가가 0원인 거래의 개수, 구매 수량(Quantity)의 최솟값, 최댓값, 평균

```
SELECT count(Quantity) AS cnt_quantity,MIN(Quantity) AS min_quantity, MAX(Qu
FROM eng-origin-464902-s3.modulabs_project.data
WHERE UnitPrice = 0;
```



행	cnt_quantity	min_quantity	max_quantity	avg_quantity
1	33	1	12540	420.5151515151...

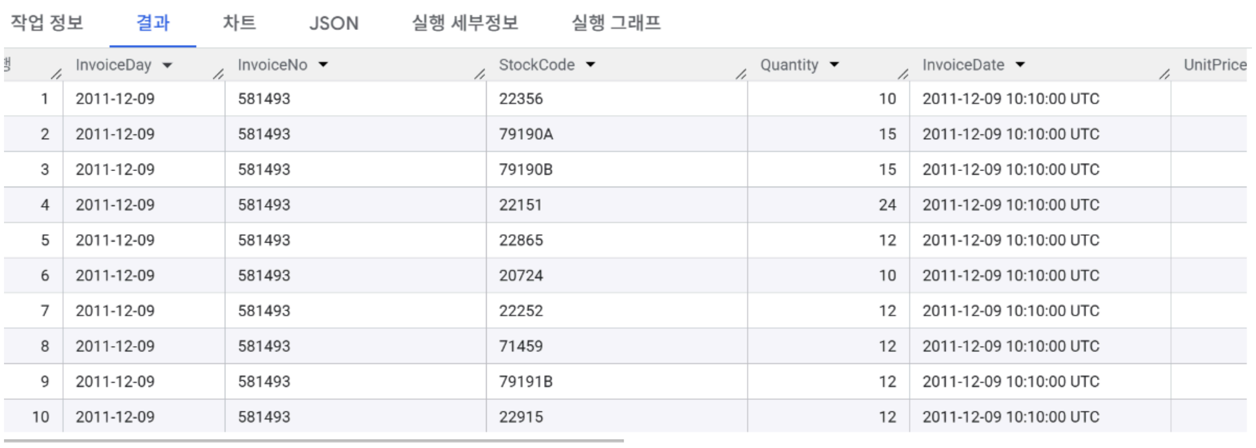
UnitPrice가 '0'이 아닌 값만 남겨서 저장

```
CREATE OR REPLACE TABLE eng-origin-464902-s3.modulabs_project.data AS
SELECT *
FROM eng-origin-464902-s3.modulabs_project.data
WHERE UnitPrice != 0;
```

11-7. RFM 스코어

1. Recency

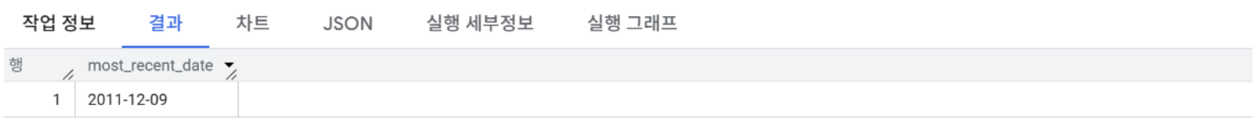
```
select date(InvoiceDate) AS InvoiceDay,*
FROM eng-origin-464902-s3.modulabs_project.data
order by InvoiceDay desc;
```



작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

행	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice
1	2011-12-09	581493	22356	10	2011-12-09 10:10:00 UTC	
2	2011-12-09	581493	79190A	15	2011-12-09 10:10:00 UTC	
3	2011-12-09	581493	79190B	15	2011-12-09 10:10:00 UTC	
4	2011-12-09	581493	22151	24	2011-12-09 10:10:00 UTC	
5	2011-12-09	581493	22865	12	2011-12-09 10:10:00 UTC	
6	2011-12-09	581493	20724	10	2011-12-09 10:10:00 UTC	
7	2011-12-09	581493	22252	12	2011-12-09 10:10:00 UTC	
8	2011-12-09	581493	71459	12	2011-12-09 10:10:00 UTC	
9	2011-12-09	581493	79191B	12	2011-12-09 10:10:00 UTC	
10	2011-12-09	581493	22915	12	2011-12-09 10:10:00 UTC	

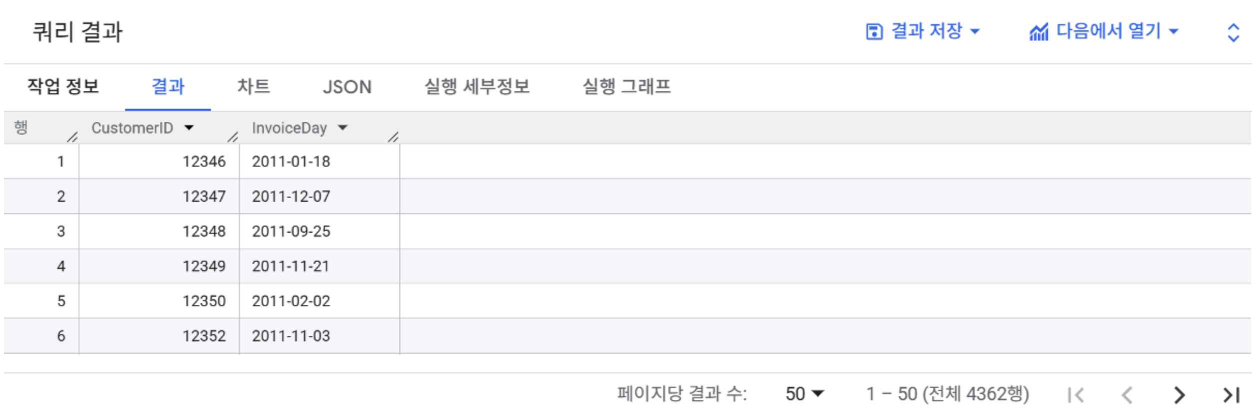
```
select max(InvoiceDay) as most_recent_date
from (select date(InvoiceDate) AS InvoiceDay, *
FROM eng-origin-464902-s3.modulabs_project.data
order by InvoiceDay desc);
```



작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

행	most_recent_date
1	2011-12-09

```
--유저 별로 가장 큰 InvoiceDay
select CustomerID,
MAX(DATE(InvoiceDate)) AS InvoiceDay
FROM eng-origin-464902-s3.modulabs_project.data
GROUP BY 1;
```



쿼리 결과

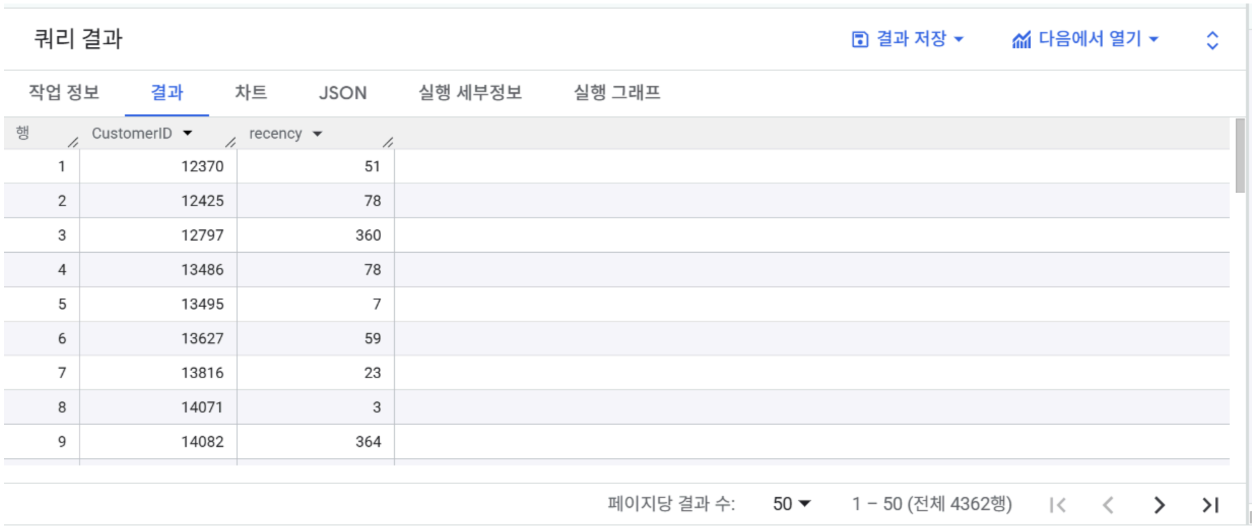
결과 저장 / 다음에서 열기

작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

행	CustomerID	InvoiceDay
1	12346	2011-01-18
2	12347	2011-12-07
3	12348	2011-09-25
4	12349	2011-11-21
5	12350	2011-02-02
6	12352	2011-11-03

페이지당 결과 수: 50　1 – 50 (전체 4362행)

```
--가장 최근 구매 일자(most_recent_date)와 유저별 마지막 구매일(InvoiceDay)간의 차이

SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM eng-origin-464902-s3.modulabs_project.data
  GROUP BY CustomerID
);
```



쿼리 결과

결과 저장 / 다음에서 열기

작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행 그래프

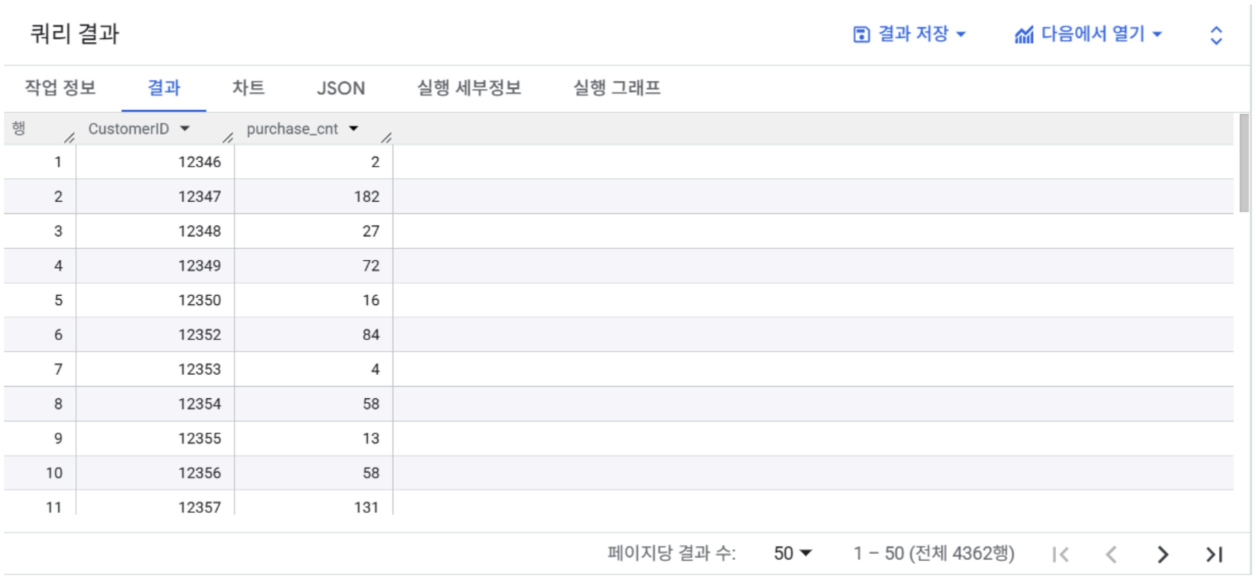
행	CustomerID	recency
1	12370	51
2	12425	78
3	12797	360
4	13486	78
5	13495	7
6	13627	59
7	13816	23
8	14071	3
9	14082	364

페이지당 결과 수: 50　1 – 50 (전체 4362행)

```
CREATE OR REPLACE TABLE eng-origin-464902-s3.modulabs_project.user_r AS

SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM eng-origin-464902-s3.modulabs_project.data
  GROUP BY CustomerID
);
```

2. Frequency

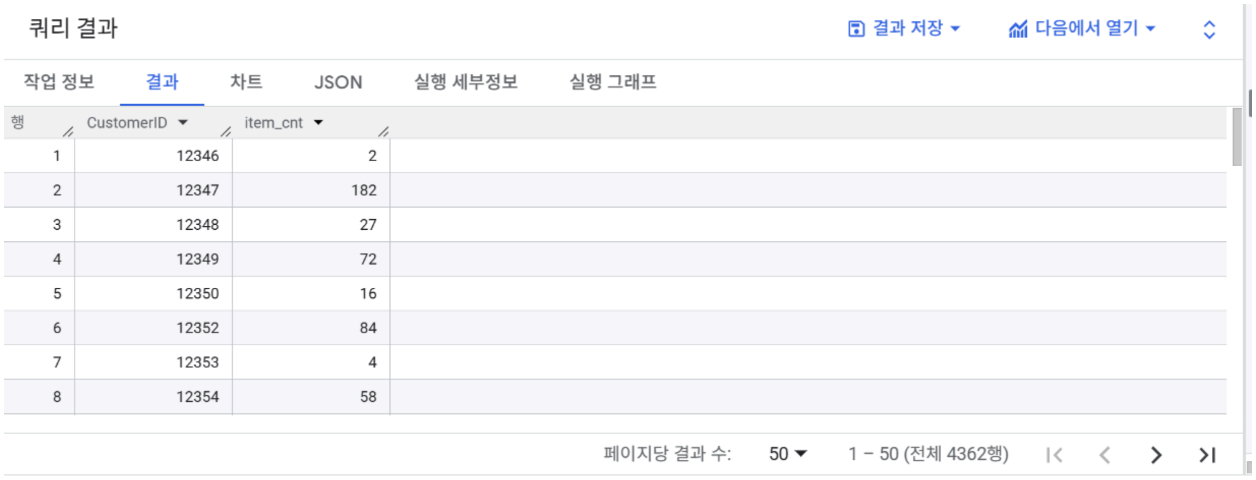
```
--1. 전체 거래 건수 계산
SELECT
  CustomerID,
  COUNT(InvoiceNo) AS purchase_cnt
FROM eng-origin-464902-s3.modulabs_project.data
GROUP BY 1;
```



행	CustomerID	purchase_cnt
1	12346	2
2	12347	182
3	12348	27
4	12349	72
5	12350	16
6	12352	84
7	12353	4
8	12354	58
9	12355	13
10	12356	58
11	12357	131

페이지당 결과 수: 50 1 – 50 (전체 4362행)

```
--2. 구매한 아이템의 총 수량 계산
SELECT
  CustomerID,
  COUNT(Quantity) AS item_cnt
FROM eng-origin-464902-s3.modulabs_project.data
GROUP BY 1;
```



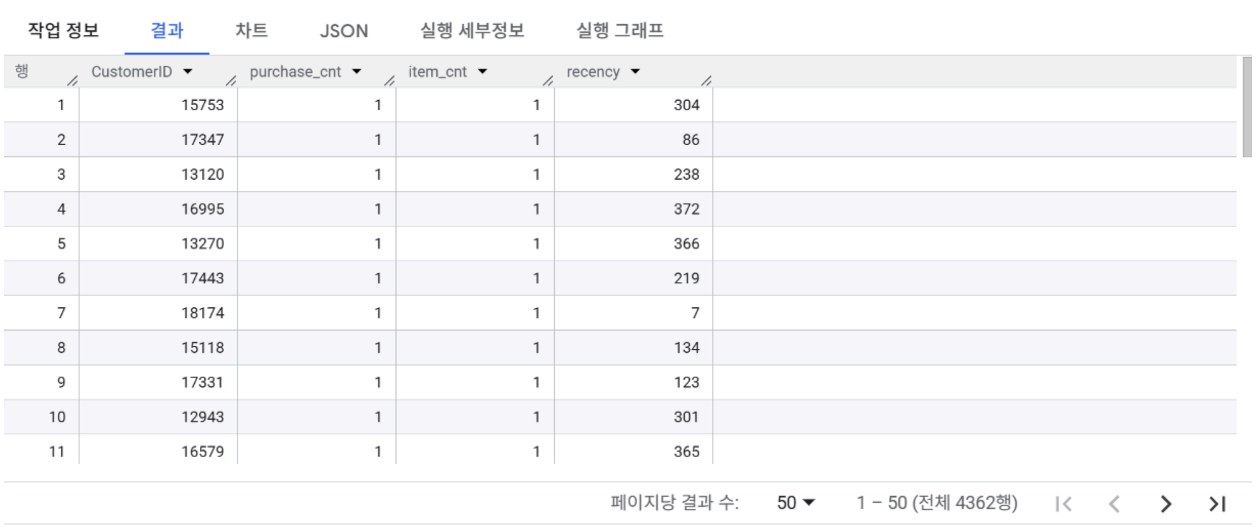
행	CustomerID	item_cnt
1	12346	2
2	12347	182
3	12348	27
4	12349	72
5	12350	16
6	12352	84
7	12353	4
8	12354	58

페이지당 결과 수: 50 1 – 50 (전체 4362행)

```
--테이블 합치기
create or replace table eng-origin-464902-s3.modulabs_project.uer_rf as

with purchase_cnt as (SELECT
  CustomerID,
  COUNT(InvoiceNo) AS purchase_cnt
FROM eng-origin-464902-s3.modulabs_project.data
GROUP BY 1),

item_cnt as (SELECT
  CustomerID,
  COUNT(Quantity) AS item_cnt
FROM eng-origin-464902-s3.modulabs_project.data
GROUP BY 1)

select
```

```
pc.CustomerID,
pc.purchase_cnt,
ic.item_cnt,
ur.recency
from purchase_cnt as pc
join item_cnt as ic
on pc.CustomerID = ic.CustomerID
JOIN eng-origin-464902-s3.modulabs_project.user_r AS ur
on ic.CustomerID = ur.CustomerID;
```

```
SELECT *
FROM eng-origin-464902-s3.modulabs_project.uer_rf
```

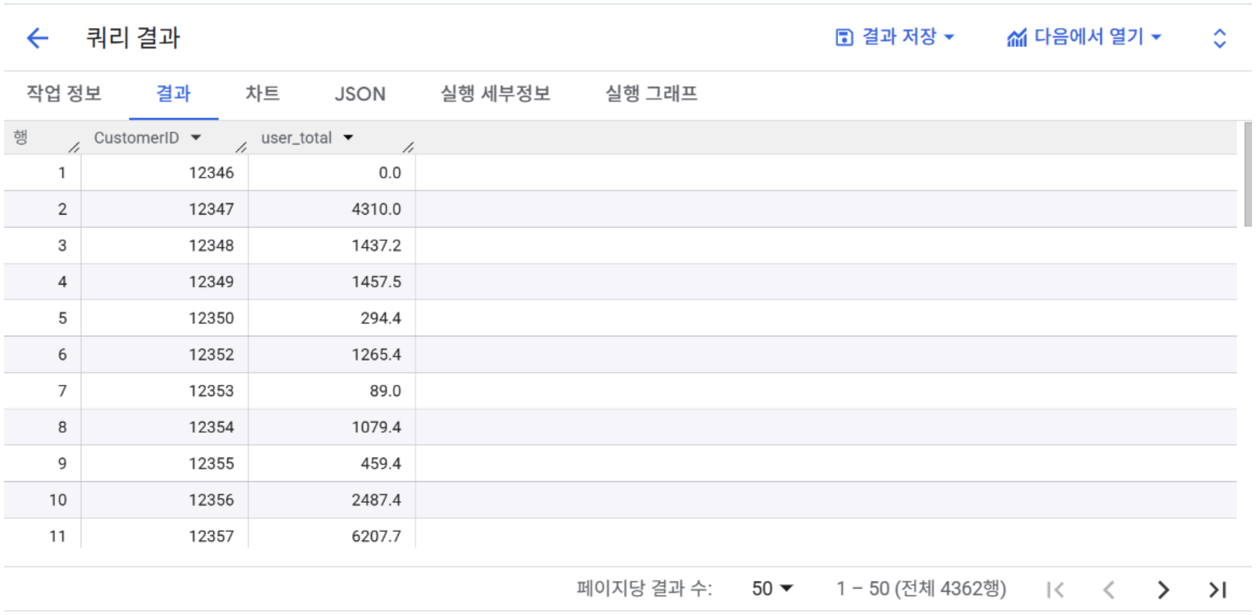


행	CustomerID	purchase_cnt	item_cnt	recency
1	15753	1	1	304
2	17347	1	1	86
3	13120	1	1	238
4	16995	1	1	372
5	13270	1	1	366
6	17443	1	1	219
7	18174	1	1	7
8	15118	1	1	134
9	17331	1	1	123
10	12943	1	1	301
11	16579	1	1	365

페이지당 결과 수: 50 1 – 50 (전체 4362행)

3. Monetary

```
--1. 고객별 총 지출액 계산
SELECT
CustomerID,
round(sum(Quantity*UnitPrice),1) AS user_total
FROM eng-origin-464902-s3.modulabs_project.data
group by 1
```



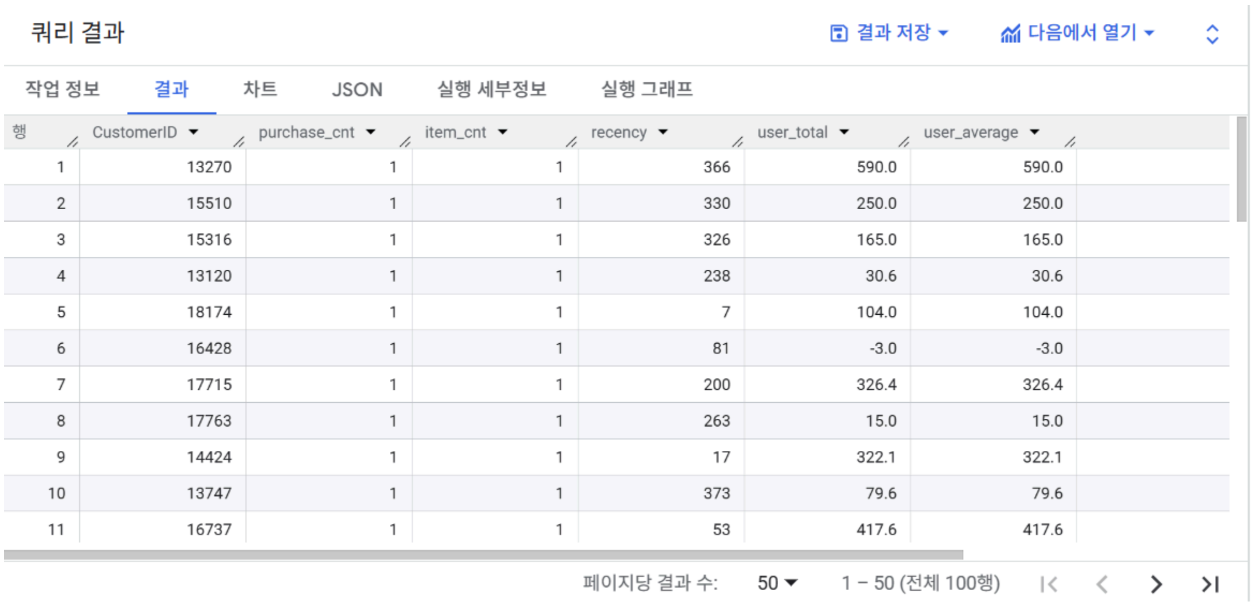
행	CustomerID	user_total
1	12346	0.0
2	12347	4310.0
3	12348	1437.2
4	12349	1457.5
5	12350	294.4
6	12352	1265.4
7	12353	89.0
8	12354	1079.4
9	12355	459.4
10	12356	2487.4
11	12357	6207.7

페이지당 결과 수: 50 1 – 50 (전체 4362행)

```
--2. 고객별 평균 거래 금액 계산
CREATE OR REPLACE TABLE `eng-origin-464902-s3.modulabs_project.user_rfm
SELECT
  rf.CustomerID AS CustomerID,
  rf.purchase_cnt,
  rf.item_cnt,
  rf.recency,
  ut.user_total,
  (ut.user_total/rf.purchase_cnt) AS user_average
FROM `eng-origin-464902-s3.modulabs_project.uer_rf` rf
LEFT JOIN (SELECT
 CustomerID,
 round(sum(Quantity*UnitPrice),1) AS user_total
FROM eng-origin-464902-s3.modulabs_project.data
group by CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;
```

```
select *
from `eng-origin-464902-s3.modulabs_project.user_rfm`
```

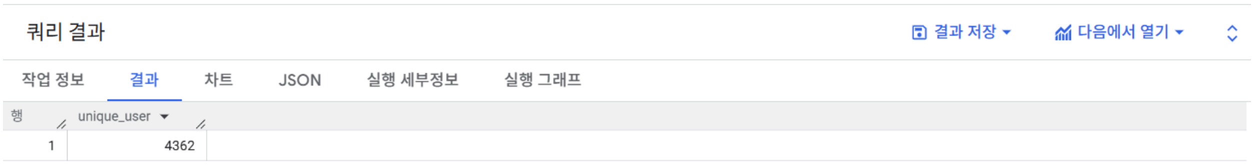
```
limit 100;
```


쿼리 결과

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	13270	1	1	366	590.0	590.0
2	15510	1	1	330	250.0	250.0
3	15316	1	1	326	165.0	165.0
4	13120	1	1	238	30.6	30.6
5	18174	1	1	7	104.0	104.0
6	16428	1	1	81	-3.0	-3.0
7	17715	1	1	200	326.4	326.4
8	17763	1	1	263	15.0	15.0
9	14424	1	1	17	322.1	322.1
10	13747	1	1	373	79.6	79.6
11	16737	1	1	53	417.6	417.6

페이지당 결과 수: 50 1 – 50 (전체 100행)

```
select COUNT(distinct CustomerID) AS unique_user
from `eng-origin-464902-s3.modulabs_project.user_rfm`
```

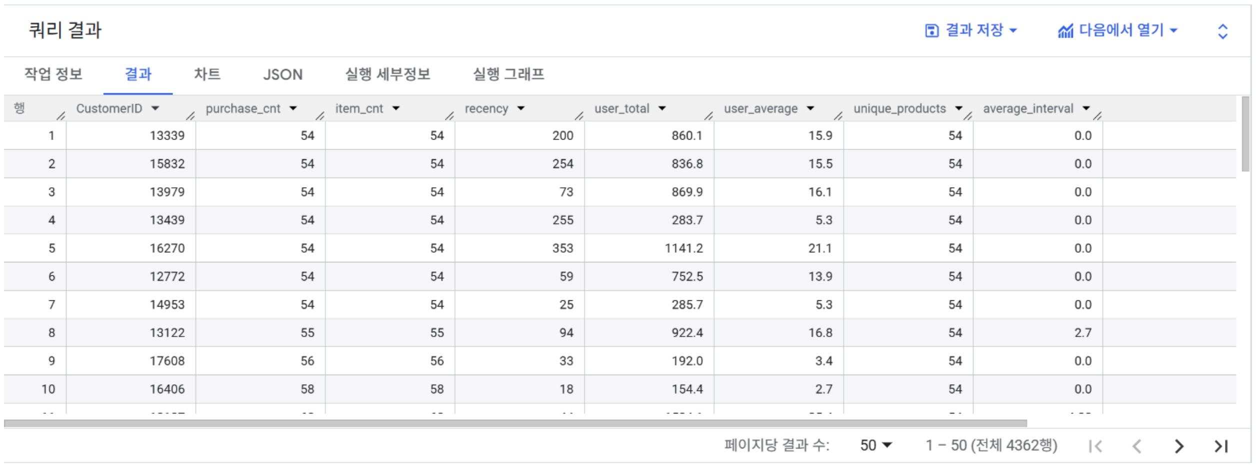

쿼리 결과

행	unique_user
1	4362

11-8. 추가 Feature 추출

```
--1. 구매하는 제품의 다양성

CREATE OR REPLACE TABLE eng-origin-464902-s3.modulabs_project.user_data

with unique_products as (
  SELECT
  CustomerID,
  COUNT(DISTINCT StockCode) AS unique_products
  FROM eng-origin-464902-s3.modulabs_project.data
  GROUP BY CustomerID
)

select ur.*, up.* EXCEPT (CustomerID)
from `eng-origin-464902-s3.modulabs_project.user_rfm` ur
join unique_products up
on ur.CustomerID = up.CustomerID
```

```
-- --2. 평균 구매 주기
create or replace table eng-origin-464902-s3.modulabs_project.user_data as

with purchase_interval as(
SELECT CustomerID,
CASE WHEN ROUND(AVG(interval_),2) is null then 0 else round(avg(interval_),2)
FROM (
  SELECT CustomerID,
  DATE_DIFF(InvoiceDate, lag(InvoiceDate) over(partition by CustomerID ORDER
  FROM eng-origin-464902-s3.modulabs_project.data
  WHERE CustomerID IS NOT NULL
  )
GROUP BY CustomerID
)

select ur.*, pi.* EXCEPT (CustomerID)
from eng-origin-464902-s3.modulabs_project.user_data ur
join purchase_interval pi
on ur.CustomerID = pi.CustomerID;
```


쿼리 결과

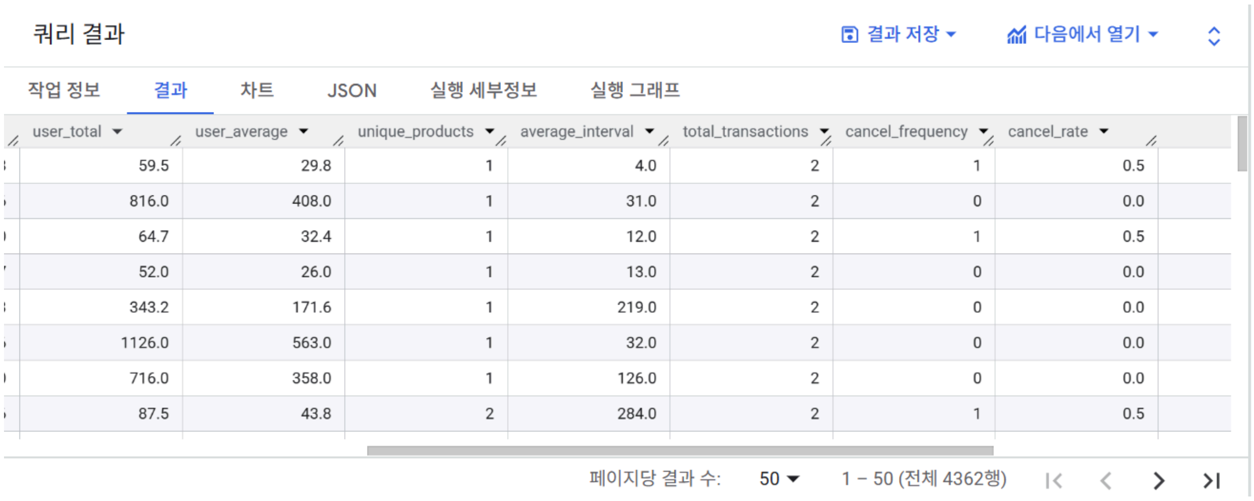
결과 저장 · 다음에서 열기

작업 정보 · 결과 · 차트 · JSON · 실행 세부정보 · 실행 그래프

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products	average_interval
1	13339	54	54	200	860.1	15.9	54	0.0
2	15832	54	54	254	836.8	15.5	54	0.0
3	13979	54	54	73	869.9	16.1	54	0.0
4	13439	54	54	255	283.7	5.3	54	0.0
5	16270	54	54	353	1141.2	21.1	54	0.0
6	12772	54	54	59	752.5	13.9	54	0.0
7	14953	54	54	25	285.7	5.3	54	0.0
8	13122	55	55	94	922.4	16.8	54	2.7
9	17608	56	56	33	192.0	3.4	54	0.0
10	16406	58	58	18	154.4	2.7	54	0.0

페이지당 결과 수: 50 · 1 – 50 (전체 4362행)

```
--3. 구매 취소 경향성
create or replace table eng-origin-464902-s3.modulabs_project.user_data as

with TransactionInfo as (
  select
  CustomerID
  ,COUNT(InvoiceNo) AS total_transactions
  ,SUM(CASE WHEN InvoiceNo like "C%" then 1 else 0 end) AS cancel_frequency
  from eng-origin-464902-s3.modulabs_project.data
  group by 1
)

SELECT ur.*
,t.* except(CustomerID)
,round(safe_divide(t.cancel_frequency, t.total_transactions),2) AS cancel_rate
from eng-origin-464902-s3.modulabs_project.user_data ur
left join TransactionInfo t
on ur.CustomerID = t.CustomerID;
```


쿼리 결과

결과 저장 · 다음에서 열기

작업 정보 · 결과 · 차트 · JSON · 실행 세부정보 · 실행 그래프

user_total	user_average	unique_products	average_interval	total_transactions	cancel_frequency	cancel_rate
59.5	29.8	1	4.0	2	1	0.5
816.0	408.0	1	31.0	2	0	0.0
64.7	32.4	1	12.0	2	1	0.5
52.0	26.0	1	13.0	2	0	0.0
343.2	171.6	1	219.0	2	0	0.0
1126.0	563.0	1	32.0	2	0	0.0
716.0	358.0	1	126.0	2	0	0.0
87.5	43.8	2	284.0	2	1	0.5

페이지당 결과 수: 50 · 1 – 50 (전체 4362행)