강상현_고객을 세그먼테이션하자 [프로젝트]

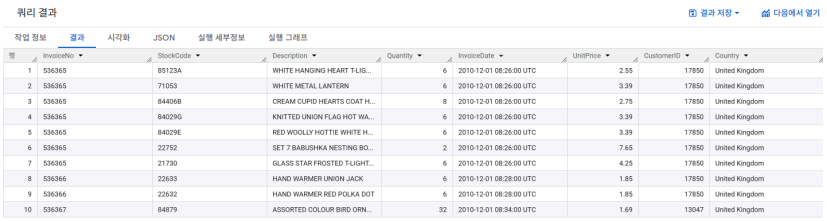
11-2. 데이터 불러오기

데이터 살펴보기

- **테이블에 있는 10개의 행만 출력하기**

```
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
LIMIT 10
```

[결과 이미지를 넣어주세요]

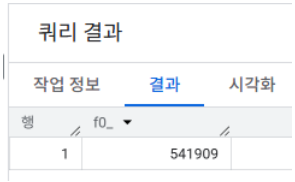


행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIG...	6	2010-12-01 08:26:00 UTC	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00 UTC	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT H...	8	2010-12-01 08:26:00 UTC	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WA...	6	2010-12-01 08:26:00 UTC	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE H...	6	2010-12-01 08:26:00 UTC	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BO...	2	2010-12-01 08:26:00 UTC	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT...	6	2010-12-01 08:26:00 UTC	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00 UTC	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00 UTC	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORN...	32	2010-12-01 08:34:00 UTC	1.69	13047	United Kingdom

- **전체 데이터는 몇 행으로 구성되어 있는지 확인하기**

```
SELECT COUNT(*)
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]

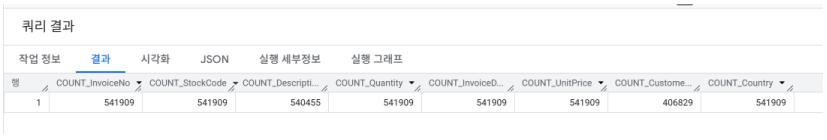


데이터 수 세기

- **COUNT 함수를 사용해서, 각 컬럼별 데이터 포인트의 수를 세어 보기**

```
SELECT COUNT(InvoiceNo) AS COUNT_InvoiceNo,
    COUNT(StockCode) AS COUNT_StockCode,
    COUNT(Description) AS COUNT_Description,
    COUNT(Quantity) AS COUNT_Quantity,
    COUNT(InvoiceDate) AS COUNT_InvoiceDate,
    COUNT(UnitPrice) AS COUNT_UnitPrice,
    COUNT(CustomerID) AS COUNT_CustomerID,
    COUNT(Country) AS COUNT_Country,
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]



행	COUNT_InvoiceNo	COUNT_StockCode	COUNT_Descripti...	COUNT_Quantity	COUNT_InvoiceD...	COUNT_UnitPrice	COUNT_Custome...	COUNT_Country
1	541909	541909	540455	541909	541909	541909	406829	541909

11-4. 데이터 전처리 방법(1): 결측치 제거

컬럼 별 누락된 값의 비율 계산

- **각 컬럼 별 누락된 값의 비율을 계산**
  - **각 컬럼에 대해서 누락 값을 계산한 후, 계산된 누락 값을 UNION ALL을 통해 합치기**

```
SELECT
    'InvoiceNo' AS column_name,
    ROUND(SUM(CASE WHEN InvoiceNo IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
UNION ALL
SELECT
    'StockCode' AS column_name,
    ROUND(SUM(CASE WHEN StockCode IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
UNION ALL
SELECT
    'Description' AS column_name,
    ROUND(SUM(CASE WHEN Description IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`

UNION ALL
SELECT
    'Quantity' AS column_name,
    ROUND(SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`

UNION ALL
SELECT
    'InvoiceDate' AS column_name,
    ROUND(SUM(CASE WHEN InvoiceDate IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`

UNION ALL
SELECT
    'UnitPrice' AS column_name,
    ROUND(SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`

UNION ALL
SELECT
    'CustomerID' AS column_name,
    ROUND(SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`

UNION ALL
SELECT
    'Country' AS column_name,
    ROUND(SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100,2) AS missing_percentage
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]

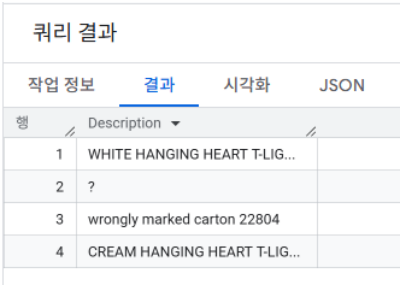
행	column_name	missing_percenta...
1	InvoiceNo	0.0
2	StockCode	0.0
3	Description	0.27
4	Quantity	0.0
5	InvoiceDate	0.0
6	UnitPrice	0.0
7	CustomerID	24.93
8	Country	0.0

결측치 처리 전략

- `StockCode = '85123A'` 의 `Description` 을 추출하는 쿼리문을 작성하기

```
SELECT DISTINCT Description
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE StockCode = '85123A'
```

[결과 이미지를 넣어주세요]



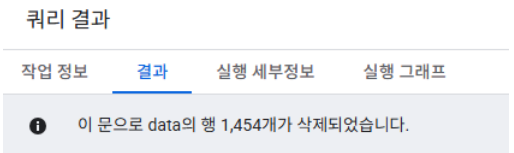
행	Description
1	WHITE HANGING HEART T-LIG...
2	?
3	wrongly marked carton 22804
4	CREAM HANGING HEART T-LIG...

결측치 처리

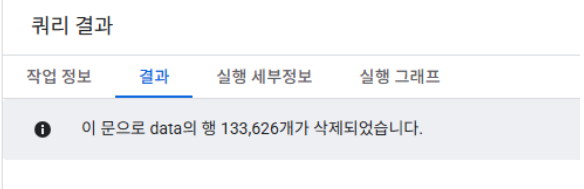
- **DELETE 구문을 사용하며, WHERE 절을 통해 데이터를 제거할 조건을 제시**

```
DELETE
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE Description IS NULL
OR CustomerID IS NULL
```

[결과 이미지를 넣어주세요]



>Description IS NULL 제거 값



>CustomerID IS NULL 제거 값

최종 135,080개가 제거 됨

11-5. 데이터 전처리(2): 중복값 처리

중복값 확인

- **중복된 행의 수를 세어보기**
  - **8개의 컬럼에 그룹 함수를 적용한 후, COUNT가 1보다 큰 데이터를 세어보기**

```
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country
HAVING COUNT(*) > 1
```

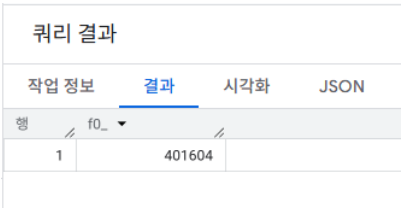
[결과 이미지를 넣어주세요]

중복값 처리

- **중복값을 제거하는 쿼리문 작성하기**
  - **CREATE OR REPLACE TABLE 구문을 활용하여 모든 컬럼(*)을 DISTINCT 한 데이터로 업데이트**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.data` AS
SELECT DISTINCT *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]



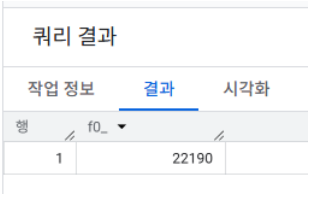
11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

- **고유(unique)한 InvoiceNo 의 개수를 출력하기**

```
SELECT COUNT(DISTINCT InvoiceNo)
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]



- **고유한 InvoiceNo 를 앞에서부터 100개를 출력하기**

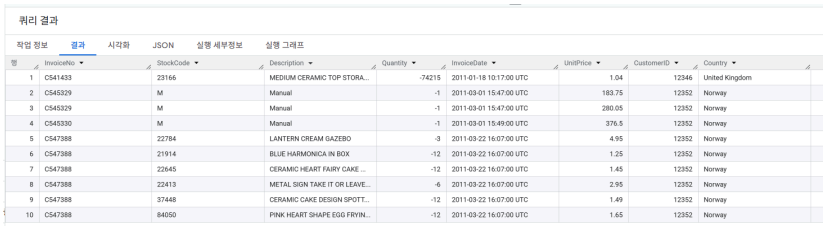
```
SELECT DISTINCT InvoiceNo
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
LIMIT 100
```

[결과 이미지를 넣어주세요]

- **InvoiceNo 가 'C'로 시작하는 행을 필터링 할 수 있는 쿼리문을 작성하기 (100행까지만 출력)**

```
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE InvoiceNo LIKE 'C%'
LIMIT 100;
```

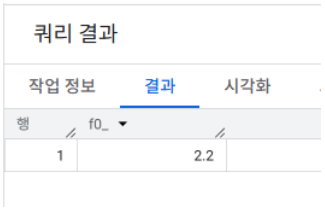
[결과 이미지를 넣어주세요]



- **구매 건 상태가 Canceled 인 데이터의 비율(%) - 소수점 첫번째 자리까지**

```
SELECT ROUND(SUM(CASE WHEN Quantity < 0 THEN 1 ELSE 0 END) / COUNT(*) * 100, 1)
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]

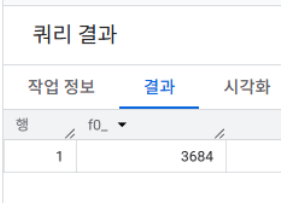


StockCode 살펴보기

- **고유한 StockCode의 개수를 출력하기**

```
SELECT COUNT(DISTINCT StockCode)
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
```

[결과 이미지를 넣어주세요]



- **어떤 제품이 가장 많이 판매되었는지 보기 위하여 StockCode 별 등장 빈도를 출력하기**
  - 상위 10개의 제품들을 출력하기

```
SELECT StockCode, COUNT(*) AS sell_cnt
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY 1
ORDER BY sell_cnt DESC
LIMIT 10
```

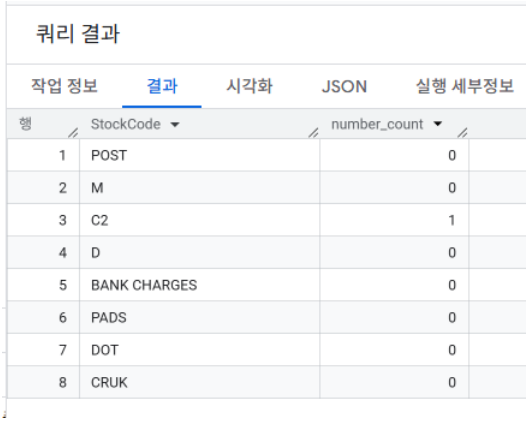
[결과 이미지를 넣어주세요]

- **StockCode의 컬럼에 있던 값 중에서 숫자를 제외한 문자만 남기고 문자가 몇 자리 수 인지 세고**

- **숫자가 0~1개인 값**들에는 어떤 코드들이 들어가 있는지 출력하기

```
SELECT DISTINCT StockCode, number_count
FROM (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
)
WHERE number_count IN(0,1);
```

[결과 이미지를 넣어주세요]



쿼리 결과

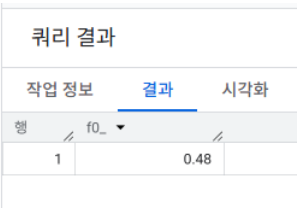
작업 정보 | 결과 | 시각화 | JSON | 실행 세부정보

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

- `StockCode`**의 컬럼에 있던 값 중에서 숫자를 제외한 문자만 남기고 문자가 몇 자리 수 인지 세고**
  - **숫자가 0~1개인 값들을 가지고 있는 데이터 수는 전체 데이터 수 대비 몇 퍼센트**인지 구하기 (소수점 두 번째 자리까지)

```
SELECT ROUND(SUM(CASE WHEN number_count IN (0,1) THEN 1 ELSE 0 END) / COUNT(*) * 100 ,2)
FROM (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
) ;
```

[결과 이미지를 넣어주세요]



쿼리 결과

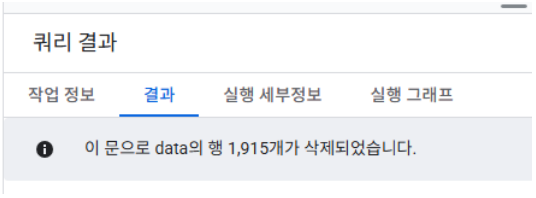
작업 정보 | 결과 | 시각화

행	f0_
1	0.48

- **제품과 관련되지 않은 거래 기록을 제거하기**

```
DELETE FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE StockCode IN (
  SELECT DISTINCT StockCode
  FROM (
    SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
  )
  WHERE number_count IN(0,1)
);
```

[결과 이미지를 넣어주세요]



Description 살펴보기

- **고유한 Description 별 출현 빈도를 계산하고 상위 30개를 출력하기**

```
SELECT Description, COUNT(*) AS description_cnt
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY Description
ORDER BY description_cnt DESC
LIMIT 30
```

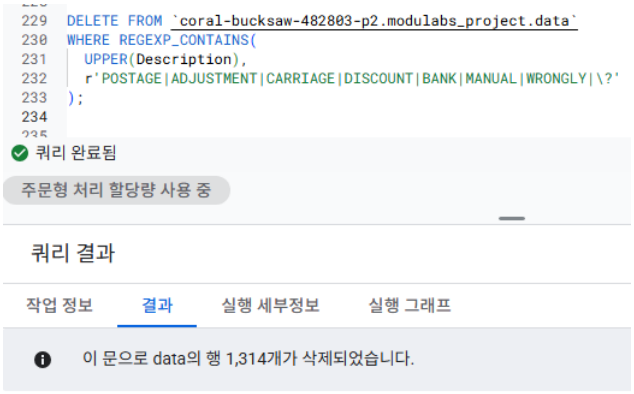
[결과 이미지를 넣어주세요]

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

- **서비스 관련 정보를 포함하는 행들을 제거하기**

```
DELETE
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE REGEXP_CONTAINS(
  UPPER(Description),
  r'POSTAGE|ADJUSTMENT|CARRIAGE|DISCOUNT|BANK|MANUAL|WRONGLY|\?'
);
```

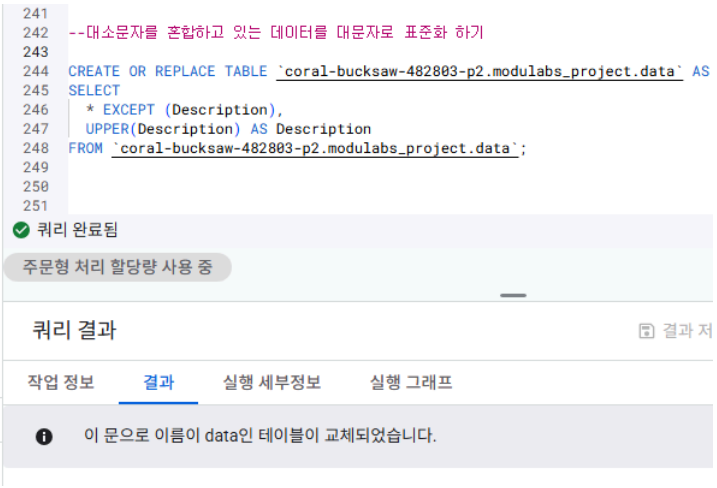
[결과 이미지를 넣어주세요]



- **대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.data` AS
SELECT
  * EXCEPT (Description),
  UPPER(Description) AS Description
FROM `coral-bucksaw-482803-p2.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]



UnitPrice 살펴보기

- **UnitPrice의 최솟값, 최댓값, 평균을 구하기**

```
SELECT MIN(UnitPrice) AS min_price, MAX(UnitPrice) AS max_price, AVG(UnitPrice) AS avg_price
FROM `coral-bucksaw-482803-p2.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

- **단가가 0원인 거래의 개수, 구매 수량(`Quantity`)의 최솟값, 최댓값, 평균 구하기**

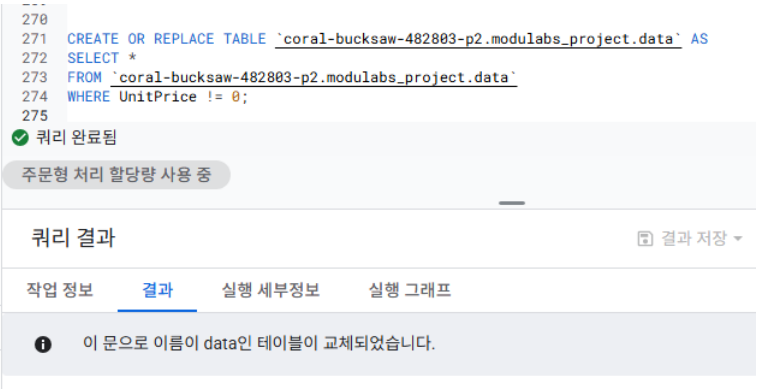
```
SELECT COUNT(Quantity) AS cnt_quantity, MIN(Quantity) AS min_quantity, MAX(Quantity) AS max_quantity, AVG(Quantity) AS avg_quantity
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE UnitPrice = 0;
```

[결과 이미지를 넣어주세요]

- **`UnitPrice = 0` 를 제거하고 일관된 데이터셋을 유지하기**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.data` AS
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
WHERE UnitPrice != 0;
```

[결과 이미지를 넣어주세요]



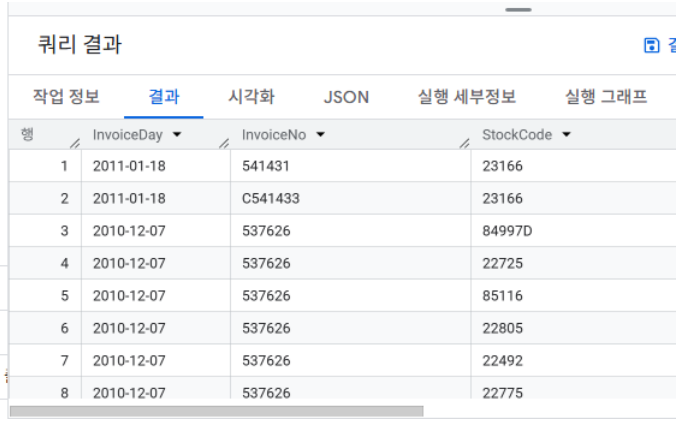
11-7. RFM 스코어

Recency

- **`InvoiceDate` 컬럼을 연월일 자료형으로 변경하기**

```
SELECT DATE(InvoiceDate) AS InvoiceDay, *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

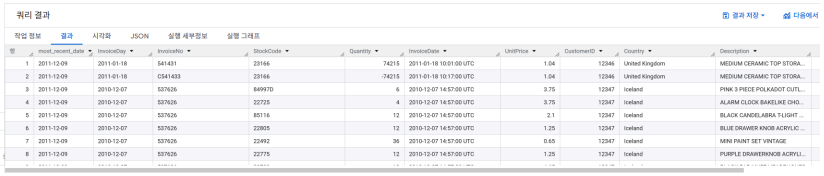


행	InvoiceDay	InvoiceNo	StockCode
1	2011-01-18	541431	23166
2	2011-01-18	C541433	23166
3	2010-12-07	537626	84997D
4	2010-12-07	537626	22725
5	2010-12-07	537626	85116
6	2010-12-07	537626	22805
7	2010-12-07	537626	22492
8	2010-12-07	537626	22775

- **가장 최근 구매 일자를 MAX() 함수로 찾아보기**

```
SELECT
   DATE(MAX(InvoiceDate) OVER()) AS most_recent_date,
  DATE(InvoiceDate) AS InvoiceDay, *
FROM `coral-bucksaw-482803-p2.modulabs_project.data`;
```

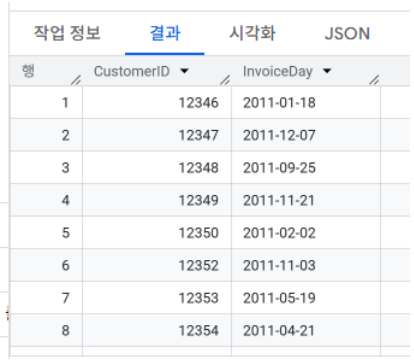
[결과 이미지를 넣어주세요]



- **유저 별로 가장 큰 InvoiceDay를 찾아서 가장 최근 구매일로 저장하기**

```
SELECT
   CustomerID,
   MAX(DATE(InvoiceDate)) AS InvoiceDay
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]



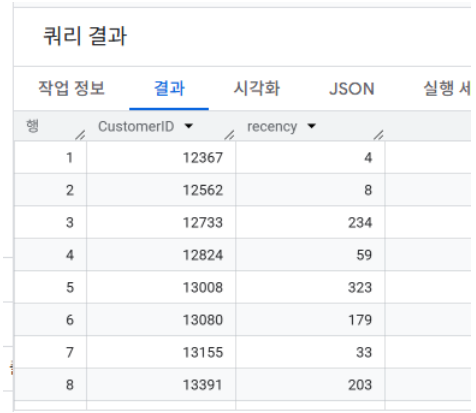
행	CustomerID	InvoiceDay
1	12346	2011-01-18
2	12347	2011-12-07
3	12348	2011-09-25
4	12349	2011-11-21
5	12350	2011-02-02
6	12352	2011-11-03
7	12353	2011-05-19
8	12354	2011-04-21

- **가장 최근 일자(most_recent_date)와 유저별 마지막 구매일(InvoiceDay)간의 차이를 계산하기**

```
SELECT
  CustomerID,
  EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
  SELECT
```

```
  CustomerID,
  MAX(DATE(InvoiceDate)) AS InvoiceDay
 FROM project_name.modulabs_project.data
 GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]



쿼리 결과

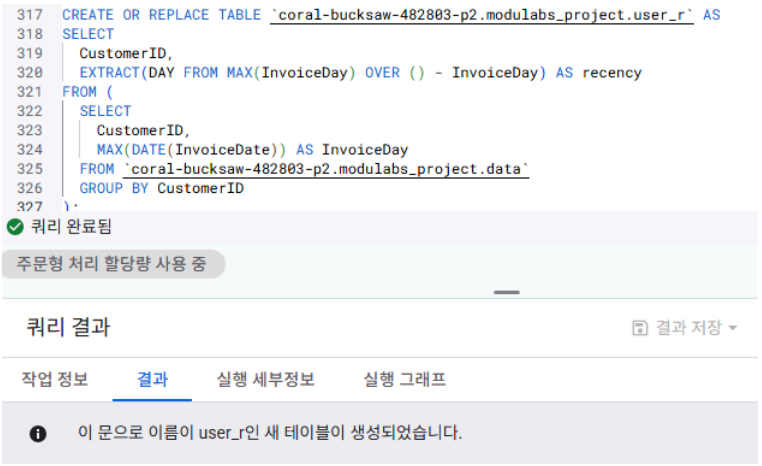
작업 정보 | 결과 | 시각화 | JSON | 실행 서

행	CustomerID	recency
1	12367	4
2	12562	8
3	12733	234
4	12824	59
5	13008	323
6	13080	179
7	13155	33
8	13391	203

- **최종 데이터 셋에 필요한 데이터들을 각각 정제해서 이어붙이고 지금까지의 결과를 `user_r` 이라는 이름의 테이블로 저장하기**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.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 `coral-bucksaw-482803-p2.modulabs_project.data`
  GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]



```
317 CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.user_r` AS
318 SELECT
319   CustomerID,
320   EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
321 FROM (
322   SELECT
323     CustomerID,
324     MAX(DATE(InvoiceDate)) AS InvoiceDay
325   FROM `coral-bucksaw-482803-p2.modulabs_project.data`
326   GROUP BY CustomerID
327 );
```

쿼리 완료됨

주문형 처리 할당량 사용 중

쿼리 결과 — 결과 저장

작업 정보 | 결과 | 실행 세부정보 | 실행 그래프

이 문으로 이름이 user_r인 새 테이블이 생성되었습니다.

user_r 쿼리 다음에서 열기 공유

스키마 세부정보 미리보기 테이블 탐색기 프리뷰

행	CustomerID	recency
1	16705	0
2	17490	0
3	18102	0
4	16626	0
5	16558	0
6	17428	0
7	13069	0
8	12985	0
9	17315	0
10	14051	0
11	12518	0
12	17364	0
13	14422	0
14	12662	0
15	13777	0

Frequency

- **고객마다 고유한 InvoiceNo의 수를 세어보기**

```
SELECT
  CustomerID,
  COUNT(InvoiceNo) AS purchase_cnt
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY CustomerID ;
```

[결과 이미지를 넣어주세요]

쿼리 결과

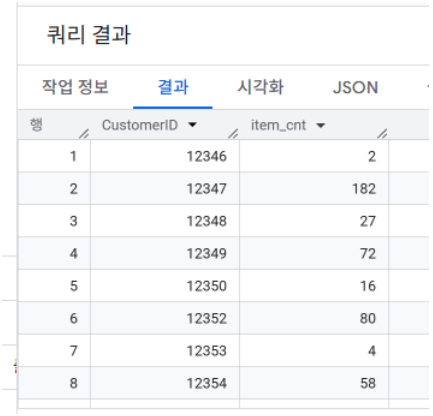
작업 정보 결과 시각화 JSON 실행 시

행	CustomerID	purchase_cnt
1	12346	2
2	12347	182
3	12348	27
4	12349	72
5	12350	16
6	12352	80
7	12353	4
8	12354	58

- **각 고객 별로 구매한 아이템의 총 수량 더하기**

```
SELECT
  CustomerID,
  COUNT(Quantity) AS item_cnt
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]



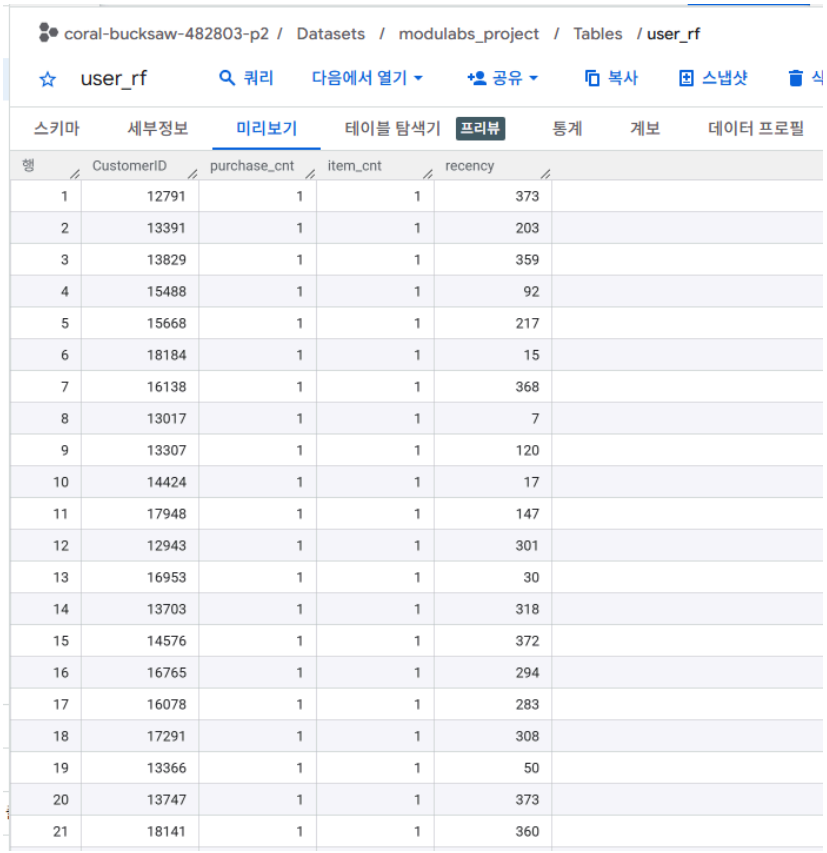
행	CustomerID	item_cnt
1	12346	2
2	12347	182
3	12348	27
4	12349	72
5	12350	16
6	12352	80
7	12353	4
8	12354	58

- **전체 거래 건수 계산와 구매한 아이템의 총 수량 계산의 결과를 합쳐서 `user_rf` 라는 이름의 테이블에 저장하기**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.user_rf` AS

-- (1) 전체 거래 건수 계산
WITH purchase_cnt AS (
  SELECT
    CustomerID,
    COUNT(InvoiceNo) AS purchase_cnt
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
  GROUP BY CustomerID
),

-- (2) 구매한 아이템 총 수량 계산
item_cnt AS (
 SELECT
    CustomerID,
    COUNT(Quantity) AS item_cnt
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY CustomerID
)

-- 기존의 user_r에 (1)과 (2)를 통합
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 `coral-bucksaw-482803-p2.modulabs_project.user_r` AS ur
  ON pc.CustomerID = ur.CustomerID;
```

[결과 이미지를 넣어주세요]

```sql
-- 기존의 user_r에 (1)과 (2)를 통합
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 `coral-bucksaw-482803-p2.modulabs_project.user_r` AS ur
  ON pc.CustomerID = ur.CustomerID;
```

행	CustomerID	purchase_cnt	item_cnt	recency
1	12791	1	1	373
2	13391	1	1	203
3	13829	1	1	359
4	15488	1	1	92
5	15668	1	1	217
6	18184	1	1	15
7	16138	1	1	368
8	13017	1	1	7
9	13307	1	1	120
10	14424	1	1	17
11	17948	1	1	147
12	12943	1	1	301
13	16953	1	1	30
14	13703	1	1	318
15	14576	1	1	372
16	16765	1	1	294
17	16078	1	1	283
18	17291	1	1	308
19	13366	1	1	50
20	13747	1	1	373
21	18141	1	1	360

Monetary

- **고객별 총 지출액 계산 (소수점 첫째 자리에서 반올림)**

```sql
SELECT
  CustomerID,
  ROUND(SUM(UnitPrice), 1) AS user_total
FROM `coral-bucksaw-482803-p2.modulabs_project.data`
GROUP BY CustomerID;
```

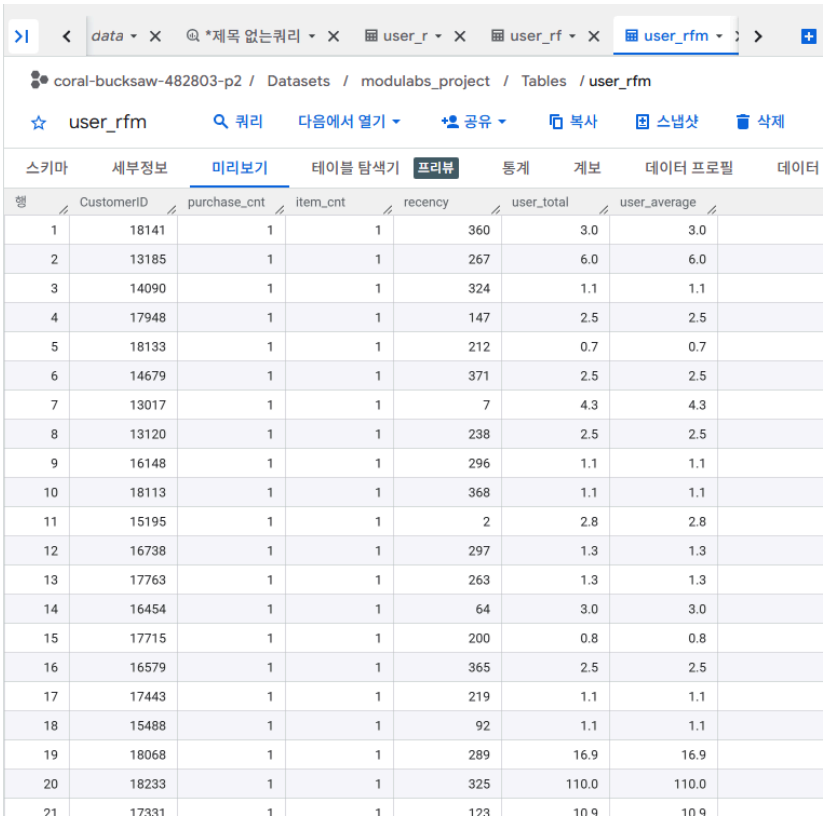
[결과 이미지를 넣어주세요]

행	CustomerID	user_total
1	12346	2.1
2	12347	481.2
3	12348	18.7
4	12349	305.1
5	12350	25.3
6	12352	324.7
7	12353	24.3
8	12354	261.2

- **고객별 평균 거래 금액 계산**
  - **고객별 평균 거래 금액을 구하기 위해 1)** `data` **테이블을** `user_rf` **테이블과 조인(LEFT JOIN) 한 후, 2)** `purchase_cnt` **로 나누어서 3)** `user_rfm` **테이블로 저장하기**

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.user_rfm` AS
SELECT
  rf.CustomerID AS CustomerID,
  rf.purchase_cnt,
  rf.item_cnt,
  rf.recency,
  ut.user_total,
  ROUND(ut.user_total / rf.purchase_cnt, 1 ) AS user_average
FROM `coral-bucksaw-482803-p2.modulabs_project.user_rf` AS rf
LEFT JOIN (
  SELECT CustomerID, ROUND(SUM(UnitPrice), 1) AS user_total
  FROM `coral-bucksaw-482803-p2.modulabs_project.data`
  GROUP BY CustomerID
) AS ut
ON rf.CustomerID = ut.CustomerID;
```

[결과 이미지를 넣어주세요]



행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	18141	1	1	360	3.0	3.0
2	13185	1	1	267	6.0	6.0
3	14090	1	1	324	1.1	1.1
4	17948	1	1	147	2.5	2.5
5	18133	1	1	212	0.7	0.7
6	14679	1	1	371	2.5	2.5
7	13017	1	1	7	4.3	4.3
8	13120	1	1	238	2.5	2.5
9	16148	1	1	296	1.1	1.1
10	18113	1	1	368	1.1	1.1
11	15195	1	1	2	2.8	2.8
12	16738	1	1	297	1.3	1.3
13	17763	1	1	263	1.3	1.3
14	16454	1	1	64	3.0	3.0
15	17715	1	1	200	0.8	0.8
16	16579	1	1	365	2.5	2.5
17	17443	1	1	219	1.1	1.1
18	15488	1	1	92	1.1	1.1
19	18068	1	1	289	16.9	16.9
20	18233	1	1	325	110.0	110.0
21	17331	1	1	123	10.9	10.9

RFM 통합 테이블 출력하기

- **최종 user_rfm 테이블을 출력하기**

```
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.user_rfm`
```

[결과 이미지를 넣어주세요]

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	18141	1	1	360	3.0	3.0
2	13185	1	1	267	6.0	6.0
3	14090	1	1	324	1.1	1.1
4	17948	1	1	147	2.5	2.5
5	18133	1	1	212	0.7	0.7
6	14679	1	1	371	2.5	2.5
7	13017	1	1	7	4.3	4.3
8	13120	1	1	238	2.5	2.5
9	16148	1	1	296	1.1	1.1
10	18113	1	1	368	1.1	1.1
11	15195	1	1	2	2.8	2.8
12	16738	1	1	297	1.3	1.3

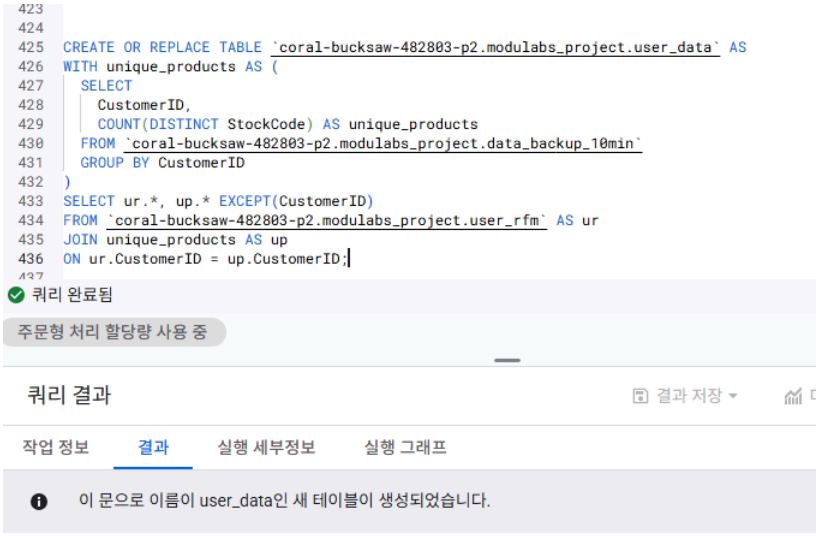
11-8. 추가 Feature 추출

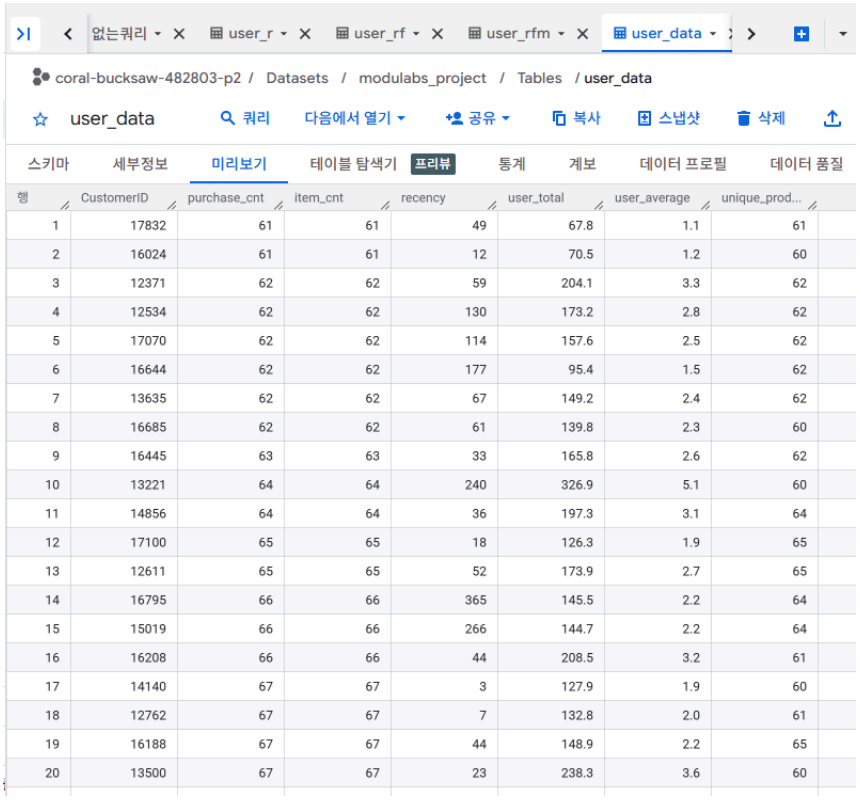
1. 구매하는 제품의 다양성

- **1) 고객 별로 구매한 상품들의 고유한 수를 계산하기**
  2) `user_rfm` **테이블과 결과를 합치기**
  3) `user_data` **라는 이름의 테이블에 저장하기**

```
CREATE OR REPLACE TABLE project_name.modulabs_project.user_data AS
WITH unique_products AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS unique_products
  FROM project_name.modulabs_project.data
  GROUP BY CustomerID
)
SELECT ur.*, up.* EXCEPT (CustomerID)
FROM project_name.modulabs_project.user_rfm AS ur
JOIN unique_products AS up
ON ur.CustomerID = up.CustomerID;
```

[결과 이미지를 넣어주세요]





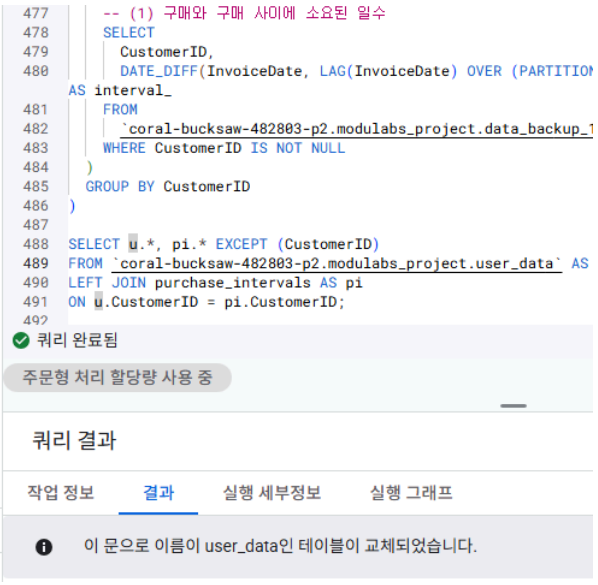
행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_prod...
1	17832	61	61	49	67.8	1.1	61
2	16024	61	61	12	70.5	1.2	60
3	12371	62	62	59	204.1	3.3	62
4	12534	62	62	130	173.2	2.8	62
5	17070	62	62	114	157.6	2.5	62
6	16644	62	62	177	95.4	1.5	62
7	13635	62	62	67	149.2	2.4	62
8	16685	62	62	61	139.8	2.3	60
9	16445	63	63	33	165.8	2.6	62
10	13221	64	64	240	326.9	5.1	60
11	14856	64	64	36	197.3	3.1	64
12	17100	65	65	18	126.3	1.9	65
13	12611	65	65	52	173.9	2.7	65
14	16795	66	66	365	145.5	2.2	64
15	15019	66	66	266	144.7	2.2	64
16	16208	66	66	44	208.5	3.2	61
17	14140	67	67	3	127.9	1.9	60
18	12762	67	67	7	132.8	2.0	61
19	16188	67	67	44	148.9	2.2	65
20	13500	67	67	23	238.3	3.6	60

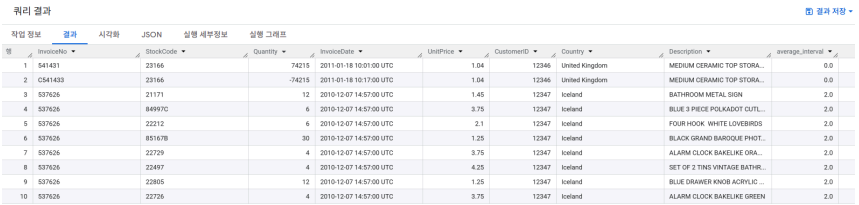
2. 평균 구매 주기

- **고객들의 쇼핑 패턴을 이해하는 것을 목표 (고객 별 재방문 주기 살펴보기)**
  - **균 구매 소요 일수를 계산하고, 그 결과를 `user_data` 에 통합**

```
CREATE OR REPLACE TABLE coral-bucksaw-482803-p2.modulabs_project.user_data AS
WITH purchase_intervals AS (
  -- (2) 고객 별 구매와 구매 사이의 평균 소요 일수
  SELECT
    CustomerID,
    CASE WHEN ROUND(AVG(interval_), 2) IS NULL THEN 0 ELSE ROUND(AVG(interval_), 2) END AS average_interval
  FROM (
    -- (1) 구매와 구매 사이에 소요된 일수
    SELECT
      CustomerID,
      DATE_DIFF(InvoiceDate, LAG(InvoiceDate) OVER (PARTITION BY CustomerID ORDER BY InvoiceDate), DAY) AS interval_
    FROM
      `coral-bucksaw-482803-p2.modulabs_project.data_backup_10min`
    WHERE CustomerID IS NOT NULL
  )
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM coral-bucksaw-482803-p2.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

[결과 이미지를 넣어주세요]





3. 구매 취소 경향성

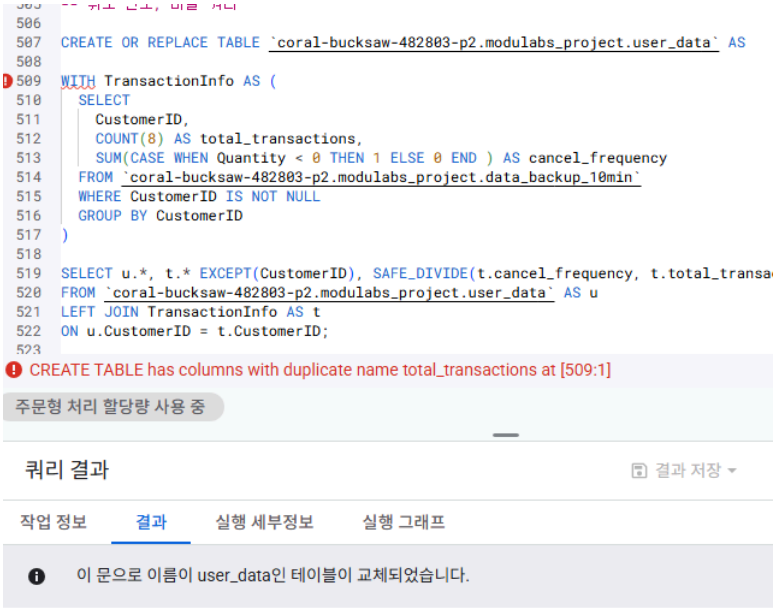
- **고객의 취소 패턴 파악하기**
  1) 취소 빈도(cancel_frequency) : 고객 별로 취소한 거래의 총 횟수

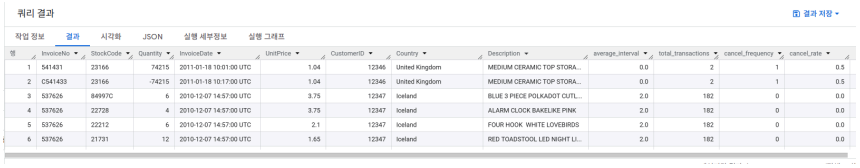
2) 취소 비율(cancel_rate) : 각 고객이 한 모든 거래 중에서 취소를 한 거래의 비율

- **취소 빈도와 취소 비율을 계산하고 그 결과를 `user_data`에 통합하기**
  (취소 비율은 소수점 두번째 자리)

```
CREATE OR REPLACE TABLE `coral-bucksaw-482803-p2.modulabs_project.user_data` AS

WITH TransactionInfo AS (
  SELECT
    CustomerID,
    COUNT(8) AS total_transactions,
    SUM(CASE WHEN Quantity < 0 THEN 1 ELSE 0 END ) AS cancel_frequency
  FROM `coral-bucksaw-482803-p2.modulabs_project.data_backup_10min`
  WHERE CustomerID IS NOT NULL
  GROUP BY CustomerID
)

SELECT u.*, t.* EXCEPT(CustomerID), SAFE_DIVIDE(t.cancel_frequency, t.total_transactions ) AS cancel_rate
FROM `coral-bucksaw-482803-p2.modulabs_project.user_data` AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID;
```

[결과 이미지를 넣어주세요]

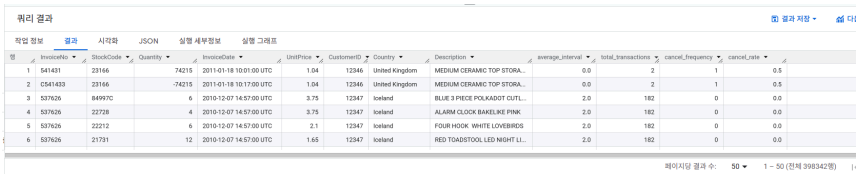




- **다양한 컬럼들을 활용하여 고객의 구매 패턴과 선호도를 보다 심층적으로 이해할 수 있도록 최종적으로 `user_data`를 출력하기**

```
SELECT *
FROM `coral-bucksaw-482803-p2.modulabs_project.user_data`
```

[결과 이미지를 넣어주세요]



중간에 데이터 한번 날려 먹어서 진행이 안되는 것일 수도 있어요... 마지막에 안맞네요ㅠ

회고

[회고 내용을 작성해주세요]

Keep :

Problem : 백업테이블 없이 CREATE OR REPLACE TABLE 하다가 테이블 이름 잘 못 적어서 기본 데이터인 data 테이블을 날려먹음

Try : 10분전 데이블 새 테이블로 복사해서 시도