고객을 세그먼테이션하자 [프로젝트]

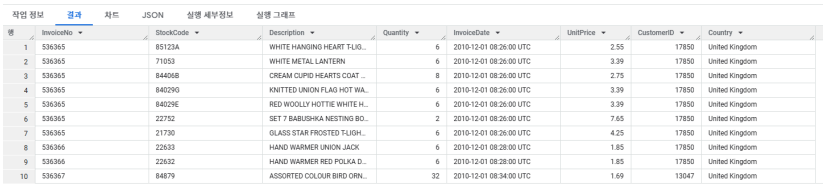
11-2. 데이터 불러오기

데이터 살펴보기

- **테이블에 있는 10개의 행만 출력하기**

```
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.data
LIMIT 10
```

[결과 이미지]

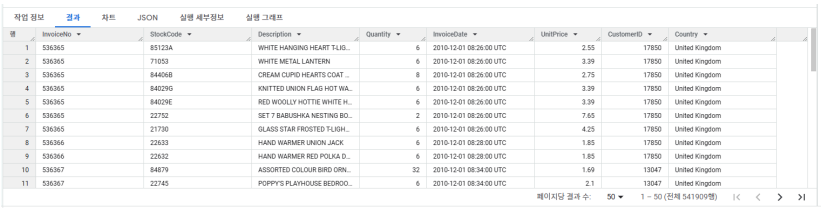


행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIG...	6	2010-12-01 08:26:00 UTC	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00 UTC	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT ...	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-LIGH...	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 D...	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 *
FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]



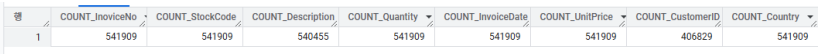
행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIG...	6	2010-12-01 08:26:00 UTC	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00 UTC	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT ...	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-LIGH...	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 D...	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
11	536367	22745	POPPY'S PLAYHOUSE BEDROO...	6	2010-12-01 08:34:00 UTC	2.1	13047	United Kingdom

데이터 수 세기

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

```
SELECT COUNT(InvoiceNo) AS COUNT_InoviceNo, 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 possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]



행	COUNT_InoviceNo	COUNT_StockCode	COUNT_Description	COUNT_Quantity	COUNT_InvoiceDate	COUNT_UnitPrice	COUNT_CustomerID	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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

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

결측치 처리 전략

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

```
SELECT DISTINCT Description
FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE StockCode = '85123A'
```

[결과 이미지]

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

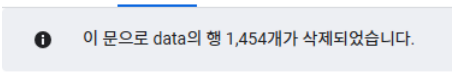
결측치 처리

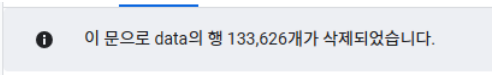
- **DELETE 구문을 사용하며, WHERE 절을 통해 데이터를 제거할 조건을 제시**

```
DELETE FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE Description IS NULL
OR CustomerID IS NULL
```

[결과 이미지]

처음에 실수로 AND로 1,454를 먼저 삭제하고, 그 다음 OR로 나머지를 삭제함. (총 135,085행 삭제)





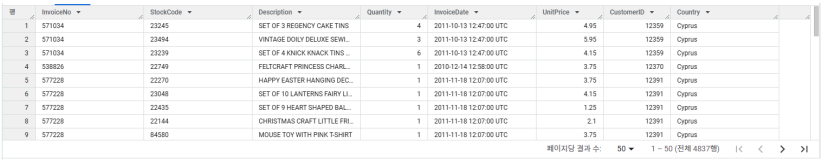
11-5. 데이터 전처리(2): 중복값 처리

중복값 확인

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

```
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1,2,3,4,5,6,7,8
HAVING COUNT(*) > 1
```

[결과 이미지]

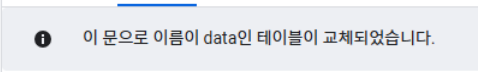


중복값 처리

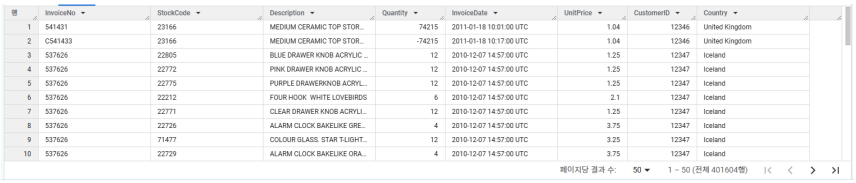
- **중복값을 제거하는 쿼리문 작성하기**
  - **CREATE OR REPLACE TABLE 구문을 활용하여 모든 컬럼(*)을 DISTINCT 한 데이터로 업데이트**

```
CREATE OR REPLACE TABLE `possible-cocoa-456102-m9.modulabs_project.data` AS
SELECT DISTINCT *
FROM `possible-cocoa-456102-m9.modulabs_project.data`
```

[결과 이미지]



행	cnt_rows
1	401604



11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

- **고유(unique)한 InvoiceNo의 개수를 출력하기**

```
SELECT
  'InvoiceNo' AS column_name,
   COUNT(*) AS UniqueInvoiceNo FROM (
     SELECT InvoiceNo
     FROM possible-cocoa-456102-m9.modulabs_project.data
     GROUP By InvoiceNo)
```

[결과 이미지]

행	column_name	UniqueInvoiceNo
1	InvoiceNo	22190

- **고유한 InvoiceNo를 앞에서부터 100개를 출력하기**

```
SELECT InvoiceNo
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP By 1
ORDER BY 1
LIMIT 100
```

[결과 이미지]

행	InvoiceNo
1	536365
2	536366
3	536367
4	536368
5	536369
6	536370
7	536371
8	536372
9	536373
10	536374

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE InvoiceNo LIKE 'C%'
LIMIT 100
```

[결과 이미지]

행	InvoiceNo	StockCode	Description	Quantity	Invo
1	C541433	23166	MEDIUM CERAMIC TOP STOR...	-74215	201
2	C545329	M	Manual	-1	201
3	C545329	M	Manual	-1	201
4	C545330	M	Manual	-1	201
5	C547388	22413	METAL SIGN TAKE IT OR LEAV...	-6	201
6	C547388	22701	PINK DOG BOWL	-6	201
7	C547388	21914	BLUE HARMONICA IN BOX	-12	201
8	C547388	84050	PINK HEART SHAPE EGG FRYI...	-12	201
9	C547388	22784	LANTERN CREAM GAZEBO	-3	201

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT ROUND(SUM(CASE WHEN InvoiceNo LIKE 'C%' THEN 1 ELSE 0 END) / COUNT(*) * 100, 1) AS CanceledRatio
FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

행	CanceledRatio
1	2.2

`StockCode` 살펴보기

- **고유한 `StockCode` 의 개수를 출력하기**

```
SELECT
  'StockCode' AS column_name,
   COUNT(*) AS UniqueStockCode FROM (
     SELECT StockCode
     FROM possible-cocoa-456102-m9.modulabs_project.data
     GROUP By StockCode)
```

[결과 이미지]

행	column_name	UniqueStockCode
1	StockCode	3684

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

```
SELECT StockCode, COUNT(*) AS sell_cnt
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
ORDER BY sell_cnt DESC
LIMIT 10
```

[결과 이미지]

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

- `StockCode`**의 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 possible-cocoa-456102-m9.modulabs_project.data
)
WHERE number_count < 2
```

[결과 이미지]

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
  'StringStockCode' AS column_name,
  ROUND(stringcnt / allcnt * 100 , 2)  AS Ratio FROM (
    SELECT
      COUNT(*) AS allcnt,
      SUM(CASE WHEN
      LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) < 2 THEN 1 ELSE 0 END) AS stringcnt
  FROM possible-cocoa-456102-m9.modulabs_project.data )
```

[결과 이미지]

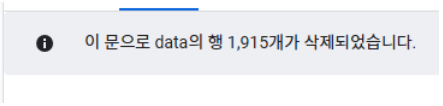
행	column_name	Ratio
1	StringStockCode	0.48

- **제품과 관련되지 않은 거래 기록을 제거하기**

```
DELETE
FROM possible-cocoa-456102-m9.modulabs_project.data
```

```
WHERE StockCode IN (
  SELECT DISTINCT StockCode
  FROM ( SELECT StockCode,
    FROM possible-cocoa-456102-m9.modulabs_project.data
    WHERE LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) < 2 ))
```

[결과 이미지]

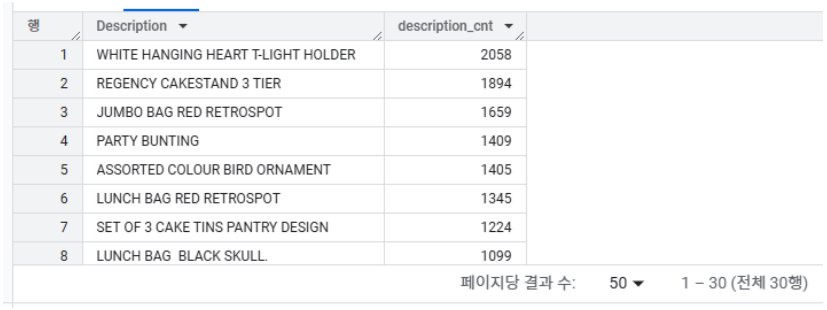


Description 살펴보기

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

```
SELECT Description, COUNT(*) AS description_cnt
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
ORDER BY 2 DESC
LIMIT 30
```

[결과 이미지]

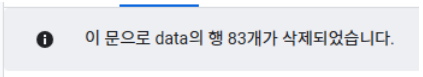


행	Description	description_cnt
1	WHITE HANGING HEART T-LIGHT HOLDER	2058
2	REGENCY CAKESTAND 3 TIER	1894
3	JUMBO BAG RED RETROSPOT	1659
4	PARTY BUNTING	1409
5	ASSORTED COLOUR BIRD ORNAMENT	1405
6	LUNCH BAG RED RETROSPOT	1345
7	SET OF 3 CAKE TINS PANTRY DESIGN	1224
8	LUNCH BAG BLACK SKULL.	1099

- **서비스 관련 정보를 포함하는 행들을 제거하기**

```
DELETE
FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE Description LIKE 'Next Day Carriage'
OR Description LIKE 'High Resolution Image'
```

[결과 이미지]



- **대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기**

```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.modulabs_project.data AS
SELECT
  * EXCEPT (Description),
 REGEXP_REPLACE(Description, r'[a-z]', r'[A-Z]') AS Description
FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

이 문으로 이름이 data인 테이블이 교체되었습니다.

```
188 SELECT DISTINCT Description
189 FROM possible-cocoa-456102-m9.modulabs_project.data
190 WHERE REGEXP_CONTAINS(Description, r'[a-z]')
191
192
```

쿼리 결과

작업 정보 | 결과 | 차트 | JSON | 실행 세부정보 | 실행

표시할 데이터가 없습니다.

UnitPrice 살펴보기

- **UnitPrice의 최솟값, 최댓값, 평균을 구하기**

```
SELECT
MIN(UnitPrice) OVER () AS min_price,
MAX(UnitPrice) OVER () AS max_price,
AVG(UnitPrice) OVER () AS avg_price
FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

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 구하기**

```
SELECT
 COUNT(*) OVER()  AS cnt_quantity,
 MIN(Quantity) OVER () AS min_quantity,
 MAX(Quantity) OVER () AS max_quantity,
 MAX(Quantity) OVER () AS avg_quantity
FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE UnitPrice = 0
```

[결과 이미지]

행	cnt_quantity	min_quantity	max_quantity	avg_quantity
1	33	1	12540	12540

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

```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.modulabs_project.data AS
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.data
WHERE UnitPrice != 0
```

[결과 이미지]

이 문으로 이름이 data인 테이블이 교체되었습니다.

행	min_price	max_price	avg_price
1	0.03	649.5	2.905196672447...

11-7. RFM 스코어

Recency

- `InvoiceDate` 컬럼을 연월일 자료형으로 변경하기

```
SELECT DATE(InvoiceDate) AS InvoiceDay, * EXCEPT (InvoiceDate)
FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

행	InvoiceDay	InvoiceNo	StockCode	Quantity	UnitPrice	Cu
1	2011-01-18	541431	23166	74215	1.04	
2	2011-01-18	C541433	23166	-74215	1.04	
3	2010-12-07	537626	21171	12	1.45	
4	2010-12-07	537626	22494	12	1.25	
5	2010-12-07	537626	20782	6	5.49	

- **가장 최근 구매 일자를 MAX() 함수로 찾아보기**

```
SELECT MAX(DATE(InvoiceDate)) OVER () AS most_recent_date,
 DATE(InvoiceDate) AS InvoiceDay,
 * EXCEPT (InvoiceDate)
 FROM possible-cocoa-456102-m9.modulabs_project.data
```

[결과 이미지]

행	most_recent_date	InvoiceDay	InvoiceNo	StockCode	Quantity	Uni
1	2011-12-09	2011-09-25	568145	23326	96	
2	2011-12-09	2011-08-19	563852	84816	1	
3	2011-12-09	2011-03-03	545475	21080	96	
4	2011-12-09	2011-03-03	545475	21086	216	
5	2011-12-09	2011-03-03	545475	22045	200	

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

```
SELECT DISTINCT CustomerID,
MAX(DATE(InvoiceDate)) OVER (PARTITION BY CustomerID) AS InvoiceDay
FROM possible-cocoa-456102-m9.modulabs_project.data
ORDER BY 2 DESC
```

[결과 이미지]

행	CustomerID	InvoiceDay
1	16558	2011-12-09
2	12713	2011-12-09
3	17428	2011-12-09
4	14051	2011-12-09
5	17315	2011-12-09
6	12433	2011-12-09

페이지당 결과 수: 50 1 – 50 (전체 4362행)

- **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 이미지]

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

```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.modulabs_project.data
 GROUP BY CustomerID
);
```

[결과 이미지]

행	CustomerID	recency
9	18102	0
10	17315	0
11	14446	0
12	17754	0
13	14397	0
14	16954	0

페이지당 결과 수: 50 1 – 50 (전체 4362행)

Frequency

- **고객마다 고유한 InvoiceNo의 수를 세어보기**

```
SELECT
 CustomerID,
 COUNT(DISTINCT InvoiceNo) AS purchase_cnt
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
```

[결과 이미지]

행	CustomerID	purchase_cnt
1	12346	2
2	12347	7
3	12348	4
4	12349	1
5	12350	1
6	12352	8

페이지당 결과 수: 50 1 – 50 (전체 4362행)

- **각 고객 별로 구매한 아이템의 총 수량 더하기**

```
SELECT
 CustomerID,
 SUM(Quantity) AS item_cnt
```

```
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
```

[결과 이미지]

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

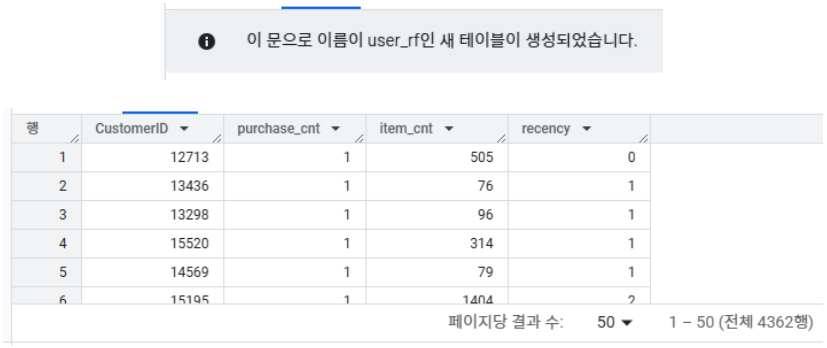
```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.modulabs_project.user_rf AS

-- (1) 전체 거래 건수 계산
WITH purchase_cnt AS (
 SELECT
  CustomerID,
  COUNT(DISTINCT InvoiceNo) AS purchase_cnt
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
),

-- (2) 구매한 아이템 총 수량 계산
item_cnt AS (
SELECT
  CustomerID,
  SUM(Quantity) AS item_cnt
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1 )

-- 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 possible-cocoa-456102-m9.modulabs_project.user_r AS ur
  ON pc.CustomerID = ur.CustomerID;
```

[결과 이미지]



Monetary

- **고객별 총 지출액 계산 (소수점 첫째 자리에서 반올림)**

```
SELECT
    CustomerID,
    ROUND(SUM(UnitPrice * Quantity), 1) AS user_total
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
```

[결과 이미지]

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

페이지당 결과 수: 50 1 – 50 (전체 4362행)

- **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 possible-cocoa-456102-m9.modulabs_project.user_rfm AS
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 possible-cocoa-456102-m9.modulabs_project.user_rf rf
LEFT JOIN (
SELECT
  CustomerID,
  ROUND(SUM(UnitPrice * Quantity), 1) AS user_total
FROM possible-cocoa-456102-m9.modulabs_project.data
GROUP BY 1
) ut
ON rf.CustomerID = ut.CustomerID;
```

[결과 이미지]

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	14911	242	76823	1	128768.2	532.1
2	12748	217	23516	0	29820.0	137.4193548387...
3	17841	169	22613	1	39861.5	235.8668639053...
4	14606	125	5932	1	11486.6	91.89280000000...
5	15311	118	37673	0	59284.2	502.4084745762...
6	13089	118	30742	2	57322.1	485.7805084745...
7	12971	88	9204	3	10933.8	124.2477272727...

페이지당 결과 수: 50 1 – 50 (전체 4362행)

RFM 통합 테이블 출력하기

- **최종 user_rfm 테이블을 출력하기**

```
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.user_rfm
ORDER BY 2 DESC
```

[결과 이미지]

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	14911	242	76823	1	128768.2	532.1
2	12748	217	23516	0	29820.0	137.4193548387...
3	17841	169	22613	1	39861.5	235.8668639053...
4	14606	125	5932	1	11486.6	91.89280000000...
5	15311	118	37673	0	59284.2	502.4084745762...
6	13089	118	30742	2	57322.1	485.7805084745...
7	12971	88	9204	3	10933.8	124.2477272727...

페이지당 결과 수: 50 1 – 50 (전체 4362행)

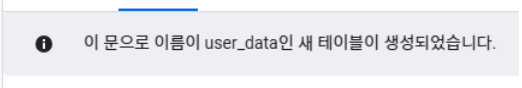
11-8. 추가 Feature 추출

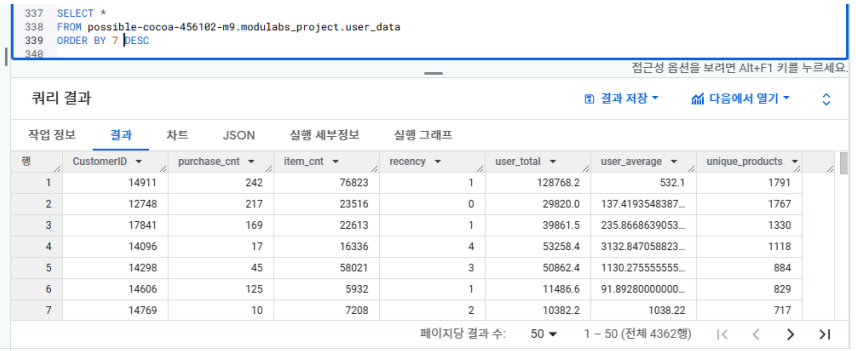
1. 구매하는 제품의 다양성

- **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 possible-cocoa-456102-m9.modulabs_project.user_data AS
WITH unique_products AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS unique_products
  FROM possible-cocoa-456102-m9.modulabs_project.data
  GROUP BY CustomerID
)
SELECT ur.*, up.* EXCEPT (CustomerID)
FROM possible-cocoa-456102-m9.modulabs_project.user_rfm AS ur
JOIN unique_products AS up
ON ur.CustomerID = up.CustomerID;
```

[결과 이미지]





2. 평균 구매 주기

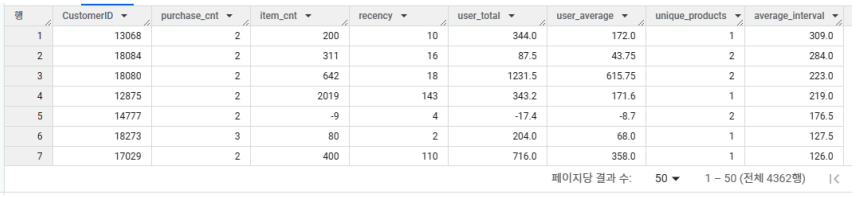
- **고객들의 쇼핑 패턴을 이해하는 것을 목표 (고객 별 재방문 주기 살펴보기)**
  - **균 구매 소요 일수를 계산하고, 그 결과를** `user_data` **에 통합**

```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.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
      possible-cocoa-456102-m9.modulabs_project.data
    WHERE CustomerID IS NOT NULL
  )
```

```
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM possible-cocoa-456102-m9.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

[결과 이미지]



행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products	average_interval
1	13068	2	200	10	344.0	172.0	1	309.0
2	18084	2	311	16	87.5	43.75	2	284.0
3	18080	2	642	18	1231.5	615.75	2	223.0
4	12875	2	2019	143	343.2	171.6	1	219.0
5	14777	2	-9	4	-17.4	-8.7	2	176.5
6	18273	3	80	2	204.0	68.0	1	127.5
7	17029	2	400	110	716.0	358.0	1	126.0

페이지당 결과 수: 50 1 – 50 (전체 4362행)

3. 구매 취소 경향성

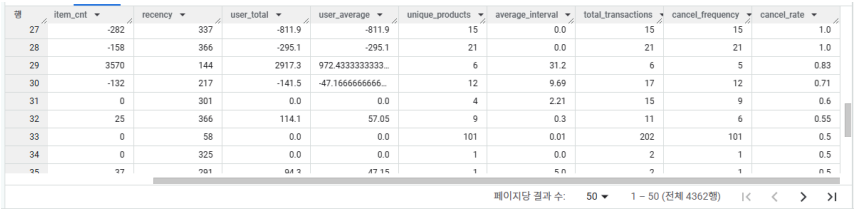
- **고객의 취소 패턴 파악하기**
  1) 취소 빈도(cancel_frequency) : 고객 별로 취소한 거래의 총 횟수
  2) 취소 비율(cancel_rate) : 각 고객이 한 모든 거래 중에서 취소를 한 거래의 비율
  - **취소 빈도와 취소 비율을 계산하고 그 결과를 `user_data` 에 통합하기**
    (취소 비율은 소수점 두번째 자리)

```
CREATE OR REPLACE TABLE possible-cocoa-456102-m9.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 possible-cocoa-456102-m9.modulabs_project.data
  GROUP BY 1
)

SELECT u.*, t.* EXCEPT(CustomerID), ROUND( t.cancel_frequency / t.total_transaction , 2) AS cancel_rate
FROM `possible-cocoa-456102-m9.modulabs_project.user_data` AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID
```

[결과 이미지]



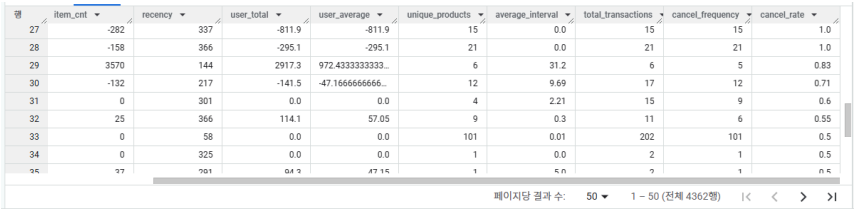
행	item_cnt	recency	user_total	user_average	unique_products	average_interval	total_transactions	cancel_frequency	cancel_rate
27	-282	337	-811.9	-811.9	15	0.0	15	15	1.0
28	-158	366	-295.1	-295.1	21	0.0	21	21	1.0
29	3570	144	2917.3	972.4333333333...	6	31.2	6	5	0.83
30	-132	217	-141.5	-47.1666666666...	12	9.69	17	12	0.71
31	0	301	0.0	0.0	4	2.21	15	9	0.6
32	25	366	114.1	57.05	9	0.3	11	6	0.55
33	0	58	0.0	0.0	101	0.01	202	101	0.5
34	0	325	0.0	0.0	1	0.0	2	1	0.5

페이지당 결과 수: 50 1 – 50 (전체 4362행)

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

```
SELECT *
FROM possible-cocoa-456102-m9.modulabs_project.user_data
ORDER BY cancel_rate DESC
```

[결과 이미지]



행	item_cnt	recency	user_total	user_average	unique_products	average_interval	total_transactions	cancel_frequency	cancel_rate
27	-282	337	-811.9	-811.9	15	0.0	15	15	1.0
28	-158	366	-295.1	-295.1	21	0.0	21	21	1.0
29	3570	144	2917.3	972.4333333333...	6	31.2	6	5	0.83
30	-132	217	-141.5	-47.1666666666...	12	9.69	17	12	0.71
31	0	301	0.0	0.0	4	2.21	15	9	0.6
32	25	366	114.1	57.05	9	0.3	11	6	0.55
33	0	58	0.0	0.0	101	0.01	202	101	0.5
34	0	325	0.0	0.0	1	0.0	2	1	0.5
35	37	281	94.3	47.15	1	5.0	2	1	0.5

페이지당 결과 수: 50 1 – 50 (전체 4362행)

회고

[회고 내용]

Keep : 집중해서 열심히 했습니다.

Problem : 어제와 동일하게 초반에 너무 겁을 먹었는데, 겁먹지 말자!

Try : 시간 여유가 있었다면 가이드를 확인하지 않고 혼자 해본 후에, 가이드를 따랐으면 좋았을 것 같습니다.