고객을 세그먼테이션하자 [프로젝트] - 송현준

11-2. 데이터 불러오기

데이터 살펴보기

- **테이블에 있는 10개의 행만 출력하기**

```
SELECT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
LIMIT 10;
```

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(*) `rows`
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	rows
1	541909

데이터 수 세기

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

```
SELECT COUNT(InvoiceNo) cnt_InvoiceNo, COUNT(StockCode) cnt_StockCode, COUNT(Description) cnt_Description, COUNT(Quantity) cnt_Quantity, COUNT(InvoiceDate) cnt_InvoiceDate, COUNT(UnitPrice) cnt_UnitPrice, COUNT(CustomerID) cnt_CustomerID, COUNT(Country) cnt_Country
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	cnt_InvoiceNo	cnt_StockCode	cnt_Description	cnt_Quantity	cnt_InvoiceDate	cnt_UnitPrice	cnt_CustomerID	cnt_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(IF(InvoiceNo IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
```

```
UNION ALL
SELECT 'StockCode' AS column_name, ROUND(SUM(IF(StockCode IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'Description' AS column_name, ROUND(SUM(IF(Description IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'Quantity' AS column_name, ROUND(SUM(IF(Quantity IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'InvoiceDate' AS column_name, ROUND(SUM(IF(InvoiceDate IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'UnitPrice' AS column_name, ROUND(SUM(IF(UnitPrice IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'CustomerID' AS column_name, ROUND(SUM(IF(CustomerID IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
UNION ALL
SELECT 'Country' AS column_name, ROUND(SUM(IF(Country IS NULL,1,0)) / COUNT(*) * 100, 2) AS missing_percentage
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	column_name	missing_percentage
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 project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE StockCode = '85123A'
ORDER BY Description;
```

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

결측치 처리

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

```
DELETE FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE Description IS NULL OR CustomerID IS NULL;
```

이 문으로 data의 행 135,080개가 삭제되었습니다.

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 COUNT(*) `중복된 행`
FROM(SELECT *,COUNT(*) cnt
    FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
    GROUP BY GROUPING SETS((InvoiceNo,StockCode,Description,Quantity,InvoiceDate,UnitPrice,CustomerID,Country))
    HAVING cnt > 1);
```

행	중복된 행
1	4837

중복값 처리

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

```
CREATE OR REPLACE TABLE project-c3ddfe61-a8f3-4972-b02.modulabs_project.data AS
SELECT DISTINCT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

이 문으로 이름이 data인 테이블이 교체되었습니다.

11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

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

```
SELECT COUNT(DISTINCT InvoiceNo)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	f0_
1	22190

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

```
SELECT DISTINCT InvoiceNo
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
LIMIT 100;
```

행	InvoiceNo
1	541431
2	C541433
3	537626
4	542237
5	549222
6	556201
7	562032
8	573511
9	581180
10	539318
11	541998
12	548955

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE InvoiceNo LIKE 'C%'
LIMIT 100;
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	C541433	23166	MEDIUM CERAMIC TOP STORA...	-74215	2011-01-18 10:17:00 UTC	1.04	12346	United Kingdom
2	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	280.05	12352	Norway
3	C545329	M	Manual	-1	2011-03-01 15:47:00 UTC	183.75	12352	Norway
4	C545330	M	Manual	-1	2011-03-01 15:49:00 UTC	376.5	12352	Norway
5	C547388	22701	PINK DOG BOWL	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
6	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1.25	12352	Norway
7	C547388	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
8	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1.65	12352	Norway
9	C547388	37448	CERAMIC CAKE DESIGN SPOTT...	-12	2011-03-22 16:07:00 UTC	1.49	12352	Norway
10	C547388	22645	CERAMIC HEART FAIRY CAKE ...	-12	2011-03-22 16:07:00 UTC	1.45	12352	Norway
11	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4.95	12352	Norway

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT ROUND(SUM(IF(InvoiceNo LIKE 'C%',1,0))/COUNT(InvoiceNo)*100,1)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	f0_
1	2.2

StockCode 살펴보기

- **고유한 `StockCode`의 개수를 출력하기**

```
SELECT COUNT(DISTINCT StockCode)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	f0_
1	3684

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

```
SELECT StockCode, COUNT(*) AS sell_cnt
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
GROUP BY StockCode
ORDER BY sell_cnt DESC
LIMIT 10;
```

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 project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
)
WHERE number_count <= 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

- `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(IF(StockCode IN (
 SELECT DISTINCT StockCode
 FROM (
  SELECT StockCode,
   LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
 )
 WHERE number_count <= 1),1,0))/COUNT(*)*100,2)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	f0_
1	0.48

- **제품과 관련되지 않은 거래 기록을 제거하기**

```
DELETE FROM project-c3ddfe61-a8f3-4972-b02.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 project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
 )
 WHERE number_count <= 1);
```

이 문으로 data의 행 1,915개가 삭제되었습니다.

Description 살펴보기

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

```
SELECT Description, COUNT(*) AS description_cnt
FROM project-c3ddfe61-a8f3-4972-b02.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

페이지당 결과 수: 50 1 – 30 (전체 30행)

- **서비스 관련 정보를 포함하는 행들을 제거하기**

```
DELETE FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE Description = 'High Resolution Image' OR Description = 'Next Day Carriage';
```

이 문으로 data의 행 83개가 삭제되었습니다.

- **대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기**

```
CREATE OR REPLACE TABLE project-c3ddfe61-a8f3-4972-b02.modulabs_project.data AS
SELECT
  * EXCEPT (Description),
  UPPER(Description) AS Description
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

이 문으로 이름이 data인 테이블이 교체되었습니다.

UnitPrice 살펴보기

- **UnitPrice의 최솟값, 최댓값, 평균을 구하기**

```
SELECT MIN(UnitPrice) AS min_price, MAX(UnitPrice) AS max_price, ROUND(AVG(UnitPrice),2) AS avg_price
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	min_price	max_price	avg_price
1	0.0	649.5	2.9

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

```
SELECT SUM(IF(UnitPrice=0,1,0)) AS cnt_quantity, MIN(Quantity) AS min_quantity, MAX(Quantity) AS max_quantity, ROUND(AVG
(Quantity),2) AS avg_quantity
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE UnitPrice=0;
```

행	cnt_quantity	min_quantity	max_quantity	avg_quantity
1	33	1	12540	420.52

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

```
CREATE OR REPLACE TABLE project-c3ddfe61-a8f3-4972-b02.modulabs_project.data AS
SELECT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
WHERE UnitPrice != 0;
```

이 문으로 이름이 data인 테이블이 교체되었습니다.

11-7. RFM 스코어

Recency

- **InvoiceDate 컬럼을 연월일 자료형으로 변경하기**

```
SELECT DATE(InvoiceDate) AS InvoiceDay, *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
1	2011-01-18	541431	23166	74215	2011-01-18 10:01:00 UTC	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORAGE J...
2	2011-01-18	C541433	23166	-74215	2011-01-18 10:17:00 UTC	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORAGE J...
3	2010-12-07	537626	22773	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	GREEN DRAWER KNOB ACRYLIC ED...
4	2010-12-07	537626	22772	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	PINK DRAWER KNOB ACRYLIC EDW...
5	2010-12-07	537626	21731	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	RED TOADSTOOL LED NIGHT LIGHT
6	2010-12-07	537626	22771	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	CLEAR DRAWER KNOB ACRYLIC ED...
7	2010-12-07	537626	22494	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	EMERGENCY FIRST AID TIN
8	2010-12-07	537626	22195	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	LARGE HEART MEASURING SPOONS
9	2010-12-07	537626	84997D	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	PINK 3 PIECE POLKADOT CUTLERY ...
10	2010-12-07	537626	84997C	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	BLUE 3 PIECE POLKADOT CUTLERY ...
11	2010-12-07	537626	85116	12	2010-12-07 14:57:00 UTC	2.1	12347	Iceland	BLACK CANDELABRA T-LIGHT HOLD...

- **가장 최근 구매 일자를 MAX() 함수로 찾아보기**

```
SELECT
  DATE(MAX(InvoiceDate) OVER()) AS most_recent_date,
  DATE(InvoiceDate) AS InvoiceDay,
  *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data;
```

행	most_recent_date	InvoiceDay	InvoiceNo	StockCode	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Description
1	2011-12-09	2011-01-18	541431	23166	74215	2011-01-18 10:01:00 UTC	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...
2	2011-12-09	2011-01-18	C541433	23166	-74215	2011-01-18 10:17:00 UTC	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...
3	2011-12-09	2010-12-07	537626	22773	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	GREEN DRAWER KNOB ACRYLI...
4	2011-12-09	2010-12-07	537626	22772	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	PINK DRAWER KNOB ACRYLIC ...
5	2011-12-09	2010-12-07	537626	21731	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	RED TOADSTOOL LED NIGHT LI...
6	2011-12-09	2010-12-07	537626	22771	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	CLEAR DRAWER KNOB ACRYLI...
7	2011-12-09	2010-12-07	537626	22494	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	EMERGENCY FIRST AID TIN
8	2011-12-09	2010-12-07	537626	22195	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	LARGE HEART MEASURING SP...
9	2011-12-09	2010-12-07	537626	84997D	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	PINK 3 PIECE POLKADOT CUTL...
10	2011-12-09	2010-12-07	537626	84997C	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	BLUE 3 PIECE POLKADOT CUTL...
11	2011-12-09	2010-12-07	537626	85116	12	2010-12-07 14:57:00 UTC	2.1	12347	Iceland	BLACK CANDELABRA T-LIGHT ...

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

```
SELECT
  CustomerID,
  DATE(MAX(InvoiceDate)) AS InvoiceDay
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
GROUP BY CustomerID
ORDER BY CustomerID;
```

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
9	12355	2011-05-09
10	12356	2011-11-17
11	12357	2011-11-06

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,
    DATE(MAX(InvoiceDate)) AS InvoiceDay
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID
  ORDER BY CustomerID
);
```

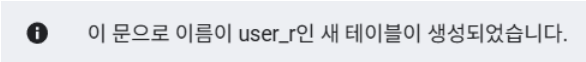
행	CustomerID	recency
1	12378	129
2	12402	323
3	12514	267
4	12689	25
5	12694	70
6	12708	29
7	12720	2
8	12747	2
9	12913	4
10	12976	211
11	13060	253

페이지당 결과 수: 50

1 – 50 (전체 4362행)

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

```
CREATE OR REPLACE TABLE project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_r AS(
  SELECT
    CustomerID,
    EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
  FROM (
    SELECT
      CustomerID,
      DATE(MAX(InvoiceDate)) AS InvoiceDay
    FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
    GROUP BY CustomerID
    ORDER BY CustomerID
  )
);
```



Frequency

- **고객마다 고유한 InvoiceNo의 수를 세어보기**

```
SELECT
  CustomerID,
  COUNT(InvoiceNo) AS purchase_cnt
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
GROUP BY CustomerID;
```

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

- **각 고객 별로 구매한 아이템의 총 수량 더하기**

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

```
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
GROUP BY CustomerID;
```

행	CustomerID	item_cnt
1	12346	0
2	12347	2458
3	12348	2332
4	12349	630
5	12350	196
6	12352	463
7	12353	20
8	12354	530
9	12355	240
10	12356	1573
11	12357	2708

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

```
CREATE OR REPLACE TABLE project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_rf AS

WITH purchase_cnt AS(
  SELECT
    CustomerID,
    COUNT(InvoiceNo) AS purchase_cnt
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID
),
item_cnt AS(
  SELECT
    CustomerID,
    SUM(Quantity) AS item_cnt
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID
)

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 project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_r AS ur
  ON pc.CustomerID = ur.CustomerID;
```

이 문으로 이름이 user_rf인 새 테이블이 생성되었습니다.

Monetary

- **고객별 총 지출액 계산 (소수점 첫째 자리에서 반올림)**

```
SELECT
  CustomerID,
  ROUND(SUM(Quantity*UnitPrice),1) AS user_total
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
GROUP BY CustomerID;
```

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

- **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 project-c3ddfe61-a8f3-4972-b02.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,2) AS user_average
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_rf rf
LEFT JOIN (
  SELECT
    CustomerID,
    ROUND(SUM(Quantity*UnitPrice),1) AS user_total
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;
```

ⓘ 이 문으로 이름이 user_rfm인 새 테이블이 생성되었습니다.

RFM 통합 테이블 출력하기

- **최종 user_rfm 테이블을 출력하기**

```
SELECT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_rfm;
```

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	17428	343	9435	0	17078.5	49.79
2	13777	217	12807	0	25758.1	118.7
3	12423	118	1312	0	1624.1	13.76
4	15804	273	2513	0	3848.5	14.1
5	13113	278	2596	0	10523.7	37.86
6	16626	184	2670	0	4379.7	23.8
7	14397	95	1852	0	2556.7	26.91
8	14446	276	856	0	1005.6	3.64
9	12985	78	1413	0	1215.6	15.58
10	17389	223	7442	0	31317.5	140.44
11	17364	409	2671	0	4437.2	10.85
12	13069	469	5454	0	3713.1	7.92

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-c3ddfe61-a8f3-4972-b02.modulabs_project.user_data AS
WITH unique_products AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT StockCode) AS unique_products
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID
)
SELECT ur.*, up.* EXCEPT (CustomerID)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_rfm AS ur
JOIN unique_products AS up
ON ur.CustomerID = up.CustomerID;
```

ⓘ 이 문으로 이름이 user_data인 새 테이블이 생성되었습니다.

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 project-c3ddfe61-a8f3-4972-b02.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
      project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
    WHERE CustomerID IS NOT NULL
  )
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

ℹ 이 문으로 이름이 user_data인 테이블이 교체되었습니다.

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 project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_data AS

WITH TransactionInfo AS (
  SELECT
    CustomerID,
    COUNT(InvoiceNo) total_transactions,
    COUNT(IF(InvoiceNo LIKE 'C%',1,NULL)) cancel_frequency
  FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.data
  GROUP BY CustomerID)

SELECT u.*, t.* EXCEPT(CustomerID), t.cancel_frequency/t.total_transactions*100 cancel_rate
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_data AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID=t.CustomerID;
```

ℹ 이 문으로 이름이 user_data인 테이블이 교체되었습니다.

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

```
SELECT *
FROM project-c3ddfe61-a8f3-4972-b02.modulabs_project.user_data;
```

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products	average_interval	total_transactions	cancel_frequency	cancel_rate
1	16765	1	4	294	34.0	34.0	1	0.0	1	0	0.0
2	16148	1	72	296	76.3	76.3	1	0.0	1	0	0.0
3	15510	1	2	330	250.0	250.0	1	0.0	1	0	0.0
4	14576	1	12	372	35.4	35.4	1	0.0	1	0	0.0
5	14424	1	48	17	322.1	322.1	1	0.0	1	0	0.0
6	13135	1	4300	196	3096.0	3096.0	1	0.0	1	0	0.0
7	15195	1	1404	2	3861.0	3861.0	1	0.0	1	0	0.0
8	13841	1	100	252	85.0	85.0	1	0.0	1	0	0.0
9	17986	1	10	56	20.8	20.8	1	0.0	1	0	0.0
10	13302	1	5	155	63.8	63.8	1	0.0	1	0	0.0
11	13017	1	48	7	204.0	204.0	1	0.0	1	0	0.0
12	12603	1	56	21	613.2	613.2	1	0.0	1	0	0.0
13	18133	1	1350	212	931.5	931.5	1	0.0	1	0	0.0
14	15070	1	36	372	106.2	106.2	1	0.0	1	0	0.0

페이지당 결과 수: 50 1 – 50 (전체 4362행)

회고

[회고 내용을 작성해주세요]

Keep :

- 다양한 방법을 사용하고 각 시도의 오류를 보면서 어떤 실수가 있었는지 확인하며 해결해나갔다.
- CTE 내에서의 계산, 메인 SELECT 절에서의 계산 등 같은 결과라도 방법을 달리해보면 적용해봄.

Problem :

-)와 같은 실수 때문에 시간이 오래 걸림.
- 원본 데이터를 유지시키는 것이 중요한데, 편하다는 이유로 하나의 테이블에서만 계속 확장해나가려고 함.
    - 데이터를 덮는 실수를 하진 않았지만 분기점 별로 테이블을 저장하는 것이 중요하다고 생각.

Try :

- 데이터를 변형해야 할 땐, 같은 테이블에 덮어씌우는 것이 아닌 새로운 테이블 만들어서 저장하기.
- 보는 사람이 이해하기 쉽도록 만든 열에 적당한 컬럼 이름 부여하기