

# 고객을 세그먼테이션하자 [프로젝트] 정범준

## 11-2. 데이터 불러오기

### 데이터 살펴보기

- 테이블에 있는 10개의 행만 출력하기

```
SELECT *  
FROM fleet-bongo-479201-u9.modulabs_project.data  
LIMIT 10
```

행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGH...	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 CLIPID 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 TLIGHT ...	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 fleet-bongo-479201-u9.modulabs_project.data
```

행	f0_
1	541909

### 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 fleet-bongo-479201-u9.modulabs_project.data
```

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 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 fleet-bongo-479201-u9.modulabs_project.data
```

UNION ALL

```

SELECT ROUND(SUM(CASE WHEN InvoiceNo IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN StockCode IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN Description IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN InvoiceDate IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
, ROUND(SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2) AS missing_percentage
FROM fleet-bongo-479201-u9.modulabs_project.data

```

행	COUNT_InvoiceNo	COUNT_StockCode	COUNT_Descripti...	COUNT_Quantity	COUNT_InvoiceD...	COUNT_UnitPrice	COUNT_Custome...	COUNT_Country
1	541909.0	541909.0	540455.0	541909.0	541909.0	541909.0	406829.0	541909.0
2	0.0	0.0	0.27	0.0	0.0	0.0	24.93	0.0

## 결측치 처리 전략

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

```

SELECT DISTINCT Description
FROM fleet-bongo-479201-u9.modulabs_project.data
WHERE StockCode = "85123A"

```

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

## 결측치 처리

- DELETE 구문을 사용하여, WHERE 절을 통해 데이터를 제거할 조건을 제시

```
DELETE
FROM fleet-bongo-479201-u9.modulabs_project.data
WHERE (CustomerID IS NULL) OR (Description 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 InvoiceNo, StockCode, Description, Quantity, InvoiceDate, Unit
Price, CustomerID, Country, COUNT(*)
FROM fleet-bongo-479201-u9.modulabs_project.data
GROUP BY InvoiceNo, StockCode, Description, Quantity, InvoiceDate, U
```

nitPrice, CustomerID, Country  
HAVING COUNT(\*) > 1

번호	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	FO
1	571034	23239	SET OF 4 KNUCK KNACK TINS P...	6	2011-10-13 12:47:00 UTC	4.15	12359	Cyprus	2
2	571034	23494	VINTAGE DOLLY DELUXE SEWIN...	3	2011-10-13 12:47:00 UTC	5.95	12359	Cyprus	2
3	571034	23245	SET OF 3 REGENCY CAKE TINS	4	2011-10-13 12:47:00 UTC	4.95	12359	Cyprus	2
4	538826	22749	FELTCRAFT PRINCESSES CHARLO...	1	2010-10-14 12:58:00 UTC	3.75	12370	Cyprus	2
5	577228	84580	MOUSE TOY WITH PINK T-SHIRT	1	2011-11-18 12:07:00 UTC	3.75	12391	Cyprus	2
6	577228	22270	HAPPY EASTER HANGING DEC...	1	2011-11-18 12:07:00 UTC	3.75	12391	Cyprus	2
7	577228	22435	SET OF 9 HEART SHAPED BALL...	1	2011-11-18 12:07:00 UTC	1.25	12391	Cyprus	2
8	577228	22144	CHRISTMAS CRAFT LITTLE FRI...	1	2011-11-18 12:07:00 UTC	2.1	12391	Cyprus	2
9	577228	23156	SET OF 5 MINI GROCERY MAGN...	1	2011-11-18 12:07:00 UTC	2.08	12391	Cyprus	2
10	577228	23048	SET OF 10 LANTERNS FAIRY LI...	1	2011-11-18 12:07:00 UTC	4.15	12391	Cyprus	3
11	539419	48138	DOORMAT UNION FLAG	10	2010-12-17 14:10:00 UTC	6.75	12431	Australia	2
12	538174	22326	ROUND SNACK BOXES SET OF 4...	12	2010-12-10 09:35:00 UTC	2.95	12471	Germany	2
13	555162	22704	WRAP RED APPLES	25	2011-06-01 10:15:00 UTC	0.42	12473	Germany	2
14	576607	22066	PAPER CHAIN KIT 50'S CHRIST...	6	2011-11-18 15:42:00 UTC	2.95	12474	Germany	2
15	576607	22940	FELTCRAFT CHRISTMAS FAIRY	8	2011-11-18 15:42:00 UTC	4.25	12474	Germany	2
16	553731	23208	LUNCH BAG VINTAGE LEAF DE...	10	2011-05-19 08:26:00 UTC	1.65	12481	Germany	2
17	564734	72131	COLUMBIAN CANDLE RECTANG...	1	2011-08-28 13:32:00 UTC	1.95	12484	Spain	2
18	564734	47580	TEA TIME DES TEA COSY	1	2011-08-28 13:32:00 UTC	2.55	12484	Spain	2
19	574550	23243	SET OF TEA COFFEE SUGAR TL...	2	2011-11-04 15:18:00 UTC	4.95	12484	Spain	2
20	575692	21818	GLITTER HEART DECORATION	1	2011-11-10 16:27:00 UTC	0.39	12508	France	2
21	569527	85123A	WHITE HANGING HEART LIGH...	1	2011-10-04 14:53:00 UTC	2.95	12512	Israel	2
22	569527	21992	VINTAGE PAISLEY STATIONERY...	1	2011-10-04 14:53:00 UTC	1.25	12512	Israel	2
23	569527	22759	SET OF 3 NOTEBOOKS IN PARC...	1	2011-10-04 14:53:00 UTC	1.65	12512	Israel	2
24	555383	22668	PINK BABY BUNTING	5	2011-06-02 15:13:00 UTC	2.95	12517	Germany	2
25	555383	21828	EIGHT PIECE SNAKE SET	2	2011-06-02 15:13:00 UTC	1.25	12517	Germany	2
26	575886	21355	TOAST ITS - I LOVE YOU	1	2011-11-11 13:57:00 UTC	1.25	12517	Germany	2
27	575886	21354	TOAST ITS - BEST MUM	1	2011-11-11 13:57:00 UTC	1.25	12517	Germany	2
28	575886	22059	CERAMIC STRAWBERRY DESIG...	1	2011-11-11 13:57:00 UTC	1.49	12517	Germany	2
29	570007	20676	RED RETROSPOT BOWL	8	2011-10-07 09:26:00 UTC	1.25	12519	Germany	2
30	559645	22553	PLASTERS IN TIN SKULLS	2	2011-07-11 13:35:00 UTC	1.65	12556	Spain	2

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 4837행)

## 중복값 처리

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

CREATE OR REPLACE TABLE fleet-bongo-479201-u9.modulabs\_project.data1 AS(

```
SELECT DISTINCT InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country
FROM fleet-bongo-479201-u9.modulabs_project.data
)
```



이 문으로 이름이 data1인 새 테이블이 생성되었습니다.

## 11-6. 데이터 전처리(3): 오류값 처리

### InvoiceNo 살펴보기

- 고유(unique)한 InvoiceNo의 개수를 출력하기

```
SELECT COUNT(DISTINCT InvoiceNo)
FROM fleet-bongo-479201-u9.modulabs_project.data1
```

행	f0_
1	22190

- 고유한 InvoiceNo를 앞에서부터 100개를 출력하기

```
SELECT DISTINCT InvoiceNo
FROM fleet-bongo-479201-u9.modulabs_project.data1
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
13	568172
14	577609
15	543037
16	544156
17	545323
18	545332
19	546869
20	547390
21	567505
22	568699
23	574275
24	C545329
25	C545330
26	C547388
27	553900
28	550911
29	552449
30	541430

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 100행)

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

```
SELECT *
FROM fleet-bongo-479201-u9.modulabs_project.data1
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	37448	CERAMIC CAKE DESIGN SPOTT...	-12	2011-03-22 16:07:00 UTC	1.49	12352	Norway
6	C547388	22645	CERAMIC HEART FAIRY CAKE M...	-12	2011-03-22 16:07:00 UTC	1.45	12352	Norway
7	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1.65	12352	Norway
8	C547388	22701	PINK DOG BOWL	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
9	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1.25	12352	Norway
10	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4.95	12352	Norway
11	C547388	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
12	C549955	22839	3 TIER CAKE TIN GREEN AND C...	-2	2011-04-13 13:38:00 UTC	14.95	12359	Cyprus
13	C549955	22666	RECIPE BOX PANTRY YELLOW D...	-2	2011-04-13 13:38:00 UTC	2.95	12359	Cyprus
14	C580165	22720	SET OF 3 CAKE TINS PANTRY D...	-1	2011-12-02 11:21:00 UTC	4.95	12359	Cyprus
15	C580165	22797	CHEST OF DRAWERS GINGHAM...	-2	2011-12-02 11:21:00 UTC	16.95	12359	Cyprus
16	C580165	22826	LOVE SEAT ANTIQUE WHITE ME...	-1	2011-12-02 11:21:00 UTC	42.5	12359	Cyprus
17	C580165	23245	SET OF 3 REGENCY CAKE TINS	-2	2011-12-02 11:21:00 UTC	4.95	12359	Cyprus
18	C544902	22273	FELTCRAFT DOLL MOLLY	-1	2011-02-24 13:05:00 UTC	2.95	12362	Belgium
19	C544902	22629	SPACEBOY LUNCH BOX	-1	2011-02-24 13:05:00 UTC	1.95	12362	Belgium
20	C563752	22659	LUNCH BOX I LOVE LONDON	-6	2011-08-19 10:38:00 UTC	1.95	12362	Belgium
21	C563752	22630	DOLLY GIRL LUNCH BOX	-1	2011-08-19 10:38:00 UTC	1.95	12362	Belgium
22	C563752	22891	TEA FOR ONE POLKADOT	-1	2011-08-19 10:38:00 UTC	4.25	12362	Belgium
23	C579178	22942	CHRISTMAS LIGHTS 10 SANTAS	-2	2011-11-28 14:55:00 UTC	8.5	12362	Belgium
24	C579178	22941	CHRISTMAS LIGHTS 10 RENDE...	-2	2011-11-28 14:55:00 UTC	8.5	12362	Belgium
25	C579178	22943	CHRISTMAS LIGHTS 10 VINTAG...	-3	2011-11-28 14:55:00 UTC	4.95	12362	Belgium
26	C544577	M	Manual	-1	2011-02-21 14:02:00 UTC	320.69	12365	Cyprus
27	C581071	85099C	JUMBO BAG BAROQUE BLACK ...	-1	2011-12-07 11:27:00 UTC	2.08	12375	Finland
28	C559193	22659	LUNCH BOX I LOVE LONDON	-1	2011-07-07 10:26:00 UTC	1.95	12379	Belgium
29	C572532	22960	JAM MAKING SET WITH JARS	-1	2011-10-24 16:03:00 UTC	4.25	12380	Belgium
30	C565050	851598	WHITE TEA,COFFEE,SUGAR JARS	-1	2011-08-31 10:00:00 UTC	1.95	12381	Norway

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)
FROM fleet-bongo-479201-u9.modulabs_project.data1
```

행	f0_
1	2.2

## StockCode 살펴보기

- 고유한 **StockCode** 의 개수를 출력하기

```
SELECT COUNT(DISTINCT StockCode)
FROM fleet-bongo-479201-u9.modulabs_project.data1
```

행	f0_
1	3684

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

```
SELECT StockCode, COUNT(*) AS cnt
FROM fleet-bongo-479201-u9.modulabs_project.data1
GROUP BY StockCode
ORDER BY cnt DESC
LIMIT 10
```

행	StockCode	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 수인지 세고
  - 숫자가 0~1개인 값들에는 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 fleet-bongo-479201-u9.modulabs_project.data1
)
WHERE number_count IN (0, 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 수인지 세고
  - 숫자가 0~1개인 값을 가지고 있는 데이터 수는 전체 데이터 수 대비 몇 퍼센트인지 구하기 (소수점 두 번째 자리까지)

```

SELECT
  ROUND(
    SUM(CASE WHEN (LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', ''))) IN (0, 1) THEN 1 ELSE 0 END)
    /
    COUNT(*)
    *100, 2)
FROM fleet-bongo-479201-u9.modulabs_project.data1
  
```

행	f0_
1	0.48

- 제품과 관련되지 않은 거래 기록을 제거하기

## DELETE

```
FROM fleet-bongo-479201-u9.modulabs_project.data1  
WHERE (LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCod  
e, r'[0-9]', ''))) IN (0, 1)
```



이 문으로 data1의 행 1,915개가 삭제되었습니다.

## Description 살펴보기

- 고유한 Description 별 출현 빈도를 계산하고 상위 30개를 출력하기

```
SELECT Description, COUNT(*) AS description_cnt  
FROM fleet-bongo-479201-u9.modulabs_project.data1  
GROUP BY Description  
LIMIT 30
```

행	Description	description_cnt
1	MEDIUM CERAMIC TOP STORA...	208
2	LARGE HEART MEASURING SP...	226
3	BATHROOM METAL SIGN	60
4	ALARM CLOCK BAKELIKE PINK	636
5	CLEAR DRAWER KNOB ACRYLIC...	331
6	GREEN DRAWER KNOB ACRYLI...	149
7	RED TOASTTOOL LED NIGHT LI...	539
8	ALARM CLOCK BAKELIKE GREEN	819
9	SET OF 2 TINS VINTAGE BATH...	59
10	FOUR HOOK WHITE LOVEBIRDS	265
11	SET/3 DECOUPAGE STACKING ...	54
12	PINK 3 PIECE POLKADOT CUTL...	103
13	BLACK CANDLESTICK FLIGHT ...	40
14	RED DRAWER KNOB ACRYLIC E...	101
15	PURPLE DRAWERKNOB ACRYLIC...	151
16	BLUE DRAWER KNOB ACRYLIC ...	124
17	PINK DRAWER KNOB ACRYLIC E...	173
18	MINI PAINT SET VINTAGE	335
19	ALARM CLOCK BAKELIKE RED	917
20	EMERGENCY FIRST AID TIN	126
21	BLACK EAR MUFF HEADPHONES	18
22	ALARM CLOCK BAKELIKE CHOC...	339
23	BOX OF 4 ASSORTED COLOUR T...	75
24	RED 3 PIECE RETROSPOT CUTL...	100
25	BOOM BOX SPEAKER BOYS	56
26	COLOUR GLASS STAR FLIGHT ...	247
27	AIRLINE BAG VINTAGE JET SET...	96
28	3D DOG PICTURE PLAYING CAR...	61
29	CAMOUFLAGE EAR MUFF HEAD...	17
30	BLUE 3 PIECE POLKA DOT CUTL...	97

페이지당 결과 수: 50 ▾ 1 ~ 30 (전체 309명)

- 서비스 관련 정보를 포함하는 행들을 제거하기

```
DELETE  
FROM fleet-bongo-479201-u9.modulabs_project.data1  
WHERE Description LIKE "Next%" OR  
Description LIKE "High%"
```

 이 문으로 data1의 행 83개가 삭제되었습니다.

- 대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기

```
CREATE OR REPLACE TABLE fleet-bongo-479201-u9.modulabs_projec  
t.data2 AS (  
    SELECT  
        * EXCEPT (Description),  
        UPPER>Description) AS Description  
    FROM fleet-bongo-479201-u9.modulabs_project.data1  
)
```

 이 문으로 이름이 data2인 새 테이블이 생성되었습니다.

## UnitPrice 살펴보기

- UnitPrice 의 최솟값, 최댓값, 평균을 구하기

```
SELECT MIN(UnitPrice) AS min_price, MAX(UnitPrice) AS max_price, A  
VG(UnitPrice) AS avg_price
```

```
FROM fleet-bongo-479201-u9.modulabs_project.data2
```

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(*), MIN(Quantity), MAX(Quantity), AVG(Quantity)
FROM fleet-bongo-479201-u9.modulabs_project.data2
WHERE UnitPrice = 0
```

행	f0_	f1_	f2_	f3_
1	33	1	12540	420.5151515151...

- UnitPrice = 0** 를 제거하고 일관된 데이터셋을 유지하기

```
CREATE OR REPLACE TABLE fleet-bongo-479201-u9.modulabs_project.data3 AS (
    SELECT *
    FROM fleet-bongo-479201-u9.modulabs_project.data2
    WHERE UnitPrice > 0
)
```

이 문으로 이름이 data3인 새 테이블이 생성되었습니다.

## 11-7. RFM 스코어

### Recency

- InvoiceDate 컬럼을 연월일 자료형으로 변경하기

```
SELECT DATE(InvoiceDate) AS InvoiceDay, *
FROM fleet-bongo-479201-u9.modulabs_project.data3
```

번호	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 STORA...
2	2011-01-18	C541433	23166	74215	2011-01-18 10:17:00 UTC	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...
3	2010-12-07	537626	22492	36	2010-12-07 14:57:00 UTC	0.65	12347	Iceland	MINI PAINT SET VINTAGE
4	2010-12-07	537626	84969	6	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	BOX OF 6 ASSORTED COLOUR T...
5	2010-12-07	537626	22497	4	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	SET OF 2 TINS VINTAGE BATHR...
6	2010-12-07	537626	84558A	24	2010-12-07 14:57:00 UTC	2.95	12347	Iceland	3D DOG PICTURE PLAYING CAR...
7	2010-12-07	537626	22773	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	GREEN DRAWER KNOB ACRYLI...
8	2010-12-07	537626	85167B	30	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	BLACK GRAND BAROQUE PHOT...
9	2010-12-07	537626	22195	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	LARGE HEART MEASURING SP...
10	2010-12-07	537626	22725	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE CHO...
11	2010-12-07	537626	22212	6	2010-12-07 14:57:00 UTC	2.1	12347	Iceland	FOUR HOOK WHITE LOVEBIRDS
12	2010-12-07	537626	21731	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	RED TOASTISTool LED NIGHT L...
13	2010-12-07	537626	22726	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE GREEN
14	2010-12-07	537626	22729	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE ORA...
15	2010-12-07	537626	20782	6	2010-12-07 14:57:00 UTC	5.49	12347	Iceland	CAMOUFLAGE EAR MUFF HEAD...
16	2010-12-07	537626	22728	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE PINK
17	2010-12-07	537626	22805	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	BLUE DRAWER KNOB ACRYLIC ...
18	2010-12-07	537626	21064	6	2010-12-07 14:57:00 UTC	5.95	12347	Iceland	BOOM BOX SPEAKER BOYS
19	2010-12-07	537626	71477	12	2010-12-07 14:57:00 UTC	3.25	12347	Iceland	COLOUR GLASS STAR FLIGHT ...
20	2010-12-07	537626	22774	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	RED DRAWER KNOB ACRYLIC E...
21	2010-12-07	537626	84997C	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	BLUE 3 PIECE POLKA DOT CUTL...
22	2010-12-07	537626	22771	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	CLEAR DRAWER KNOB ACRYLIC...
23	2010-12-07	537626	22494	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	EMERGENCY FIRST AID TIN
24	2010-12-07	537626	20780	12	2010-12-07 14:57:00 UTC	4.65	12347	Iceland	BLACK EAR MUFF HEADPHONES
25	2010-12-07	537626	22772	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	PINK DRAWER KNOB ACRYLIC E...
26	2010-12-07	537626	84997B	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	RED 3 PIECE RETROSPOT CUTL...
27	2010-12-07	537626	85232D	3	2010-12-07 14:57:00 UTC	4.95	12347	Iceland	SET/3 DECOUPAGE STACKING ...
28	2010-12-07	537626	22727	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE RED
29	2010-12-07	537626	22375	4	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	AIRLINE BAG VINTAGE JET SET...
30	2010-12-07	537626	21171	12	2010-12-07 14:57:00 UTC	1.45	12347	Iceland	BATHROOM METAL SIGN

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 399573행)

- 가장 최근 구매 일자를 MAX() 함수로 찾아보기

```
SELECT
    DATE(MAX(InvoiceDate) OVER()) AS most_recent_date,
    DATE(InvoiceDate) AS InvoiceDay,
    *
FROM fleet-bongo-479201-u9.modulabs_project.data3
```

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	22492	36	2010-12-07 14:57:00 UTC	0.65	12347	Iceland	MINI PAINT SET VINTAGE
4	2011-12-09	2010-12-07	537626	84969	6	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	BOX OF 6 ASSORTED COLOUR T...
5	2011-12-09	2010-12-07	537626	22497	4	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	SET OF 2 TINS VINTAGE BATHR...
6	2011-12-09	2010-12-07	537626	84558A	24	2010-12-07 14:57:00 UTC	2.95	12347	Iceland	3D DOG PICTURE PLAYING CAR...
7	2011-12-09	2010-12-07	537626	22773	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	GREEN DRAWER KNOB ACRYLIC...
8	2011-12-09	2010-12-07	537626	85167B	30	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	BLACK GRAND BAROQUE PHOT...
9	2011-12-09	2010-12-07	537626	22195	12	2010-12-07 14:57:00 UTC	1.65	12347	Iceland	LARGE HEART MEASURING SP...
10	2011-12-09	2010-12-07	537626	22725	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE CHO...
11	2011-12-09	2010-12-07	537626	22212	6	2010-12-07 14:57:00 UTC	2.1	12347	Iceland	FOUR HOOK WHITE LOVEBIRDS
12	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...
13	2011-12-09	2010-12-07	537626	22726	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE GREEN
14	2011-12-09	2010-12-07	537626	22729	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE ORA...
15	2011-12-09	2010-12-07	537626	20782	6	2010-12-07 14:57:00 UTC	5.49	12347	Iceland	CAMOUFLAGE EAR MUFF HEAD...
16	2011-12-09	2010-12-07	537626	22728	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE PINK
17	2011-12-09	2010-12-07	537626	22805	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	BLUE DRAWER KNOB ACRYLIC ...
18	2011-12-09	2010-12-07	537626	21064	6	2010-12-07 14:57:00 UTC	5.95	12347	Iceland	BOOM BOX SPEAKER BOYS
19	2011-12-09	2010-12-07	537626	71477	12	2010-12-07 14:57:00 UTC	3.25	12347	Iceland	COLOUR GLASS STAR TLIGHT ...
20	2011-12-09	2010-12-07	537626	22774	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	RED DRAWER KNOB ACRYLIC E...
21	2011-12-09	2010-12-07	537626	84997C	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	BLUE 3 PIECE POLKA DOT CUTL...
22	2011-12-09	2010-12-07	537626	22771	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	CLEAR DRAWER KNOB ACRYLIC...
23	2011-12-09	2010-12-07	537626	23494	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	EMERGENCY FIRST AID TIN
24	2011-12-09	2010-12-07	537626	20780	12	2010-12-07 14:57:00 UTC	4.65	12347	Iceland	BLACK EAR MUFT HEADPHONES
25	2011-12-09	2010-12-07	537626	22772	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland	PINK DRAWER KNOB ACRYLIC E...
26	2011-12-09	2010-12-07	537626	84997B	6	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	RED 3 PIECE RETROSPOT CUTL...
27	2011-12-09	2010-12-07	537626	85232D	3	2010-12-07 14:57:00 UTC	4.95	12347	Iceland	SET/3 DECOUPAGE STACKING ...
28	2011-12-09	2010-12-07	537626	22727	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland	ALARM CLOCK BAKELIKE RED
29	2011-12-09	2010-12-07	537626	22375	4	2010-12-07 14:57:00 UTC	4.25	12347	Iceland	AIRLINE BAG VINTAGE JET SET...
30	2011-12-09	2010-12-07	537626	21171	12	2010-12-07 14:57:00 UTC	1.45	12347	Iceland	BATHROOM METAL SIGN

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 3995738쪽)

## • 유저 별로 가장 큰 InvoiceDay를 찾아서 가장 최근 구매일로 저장하기

```
SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
FROM fleet-bongo-479201-u9.modulabs_project.data3
GROUP 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
12	12358	2011-12-08
13	12359	2011-12-02
14	12360	2011-10-18
15	12361	2011-02-25
16	12362	2011-12-06
17	12363	2011-08-22
18	12364	2011-12-02
19	12365	2011-02-21
20	12367	2011-12-05
21	12370	2011-10-19
22	12371	2011-10-11
23	12372	2011-09-29
24	12373	2011-02-01
25	12374	2011-11-14
26	12375	2011-12-07
27	12377	2011-01-28
28	12378	2011-08-02
29	12379	2011-09-19
30	12380	2011-11-18

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 4362쪽)

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

```

SELECT
CustomerID,
EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recen
cy
FROM (
    SELECT
        CustomerID,
        MAX(DATE(InvoiceDate)) AS InvoiceDay
    FROM fleet-bongo-479201-u9.modulabs_project.data3
    GROUP BY CustomerID
)

```

행	CustomerID	recency
1	12522	39
2	12672	329
3	12834	282
4	13249	92
5	13322	17
6	13365	8
7	13642	215
8	13811	183
9	14083	4
10	14162	173
11	14286	23
12	14354	5
13	14606	1
14	14652	77
15	14661	26
16	14810	40
17	14869	3
18	14911	1
19	14958	147
20	15006	51
21	15051	23
22	15069	179
23	15277	46
24	15344	0
25	15392	4
26	15512	155
27	15714	11
28	15925	194
29	16054	145
30	16419	112

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 4362행)

- 최종 데이터셋에 필요한 데이터들을 각각 정제해서 이어붙이고 지금까지의 결과를 `user_r`이라는 이름의 테이블로 저장하기

```

CREATE OR REPLACE TABLE fleet-bongo-479201-u9.modulabs_projec
t.user_r AS (

```

```
SELECT
CustomerID,
EXTRACT(DAY FROM MAX(InvoiceDay) OVER () - InvoiceDay) AS recency
FROM (
    SELECT
        CustomerID,
        MAX(DATE(InvoiceDate)) AS InvoiceDay
    FROM fleet-bongo-479201-u9.modulabs_project.data3
    GROUP BY CustomerID
)
)
```



이 문으로 이름이 user\_r인 새 테이블이 생성되었습니다.

## Frequency

- 고객마다 고유한 InvoiceNo의 수를 세어보기

```
SELECT
CustomerID,
COUNT(DISTINCT InvoiceNo) AS purchase_cnt
FROM fleet-bongo-479201-u9.modulabs_project.data3
GROUP BY CustomerID
```

행	CustomerID	purchase_cnt
1	12346	2
2	12347	7
3	12348	4
4	12349	1
5	12350	1
6	12352	8
7	12353	1
8	12354	1
9	12355	1
10	12356	3
11	12357	1
12	12358	2
13	12359	6
14	12360	3
15	12361	1
16	12362	13
17	12363	2
18	12364	4
19	12365	1
20	12367	1
21	12370	4
22	12371	1
23	12372	3
24	12373	1
25	12374	1
26	12375	3
27	12377	2
28	12378	1
29	12379	3
30	12380	5

페이지당 결과 수: 50 ▼ 1 ~ 50 (전체 4362행)

- 각 고객 별로 구매한 아이템의 총 수량 더하기

```

SELECT
    CustomerID,
    SUM(Quantity) AS item_cnt
FROM fleet-bongo-479201-u9.modulabs_project.data3
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
12	12358	242
13	12359	1599
14	12360	1156
15	12361	90
16	12362	2180
17	12363	408
18	12364	1499
19	12365	173
20	12367	172
21	12370	2349
22	12371	582
23	12372	788
24	12373	196
25	12374	339
26	12375	175
27	12377	942
28	12378	2829
29	12379	401
30	12380	1109

페이지당 결과 수: 50 ▼ 1 ~ 50 (전체 4362행)

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

```

CREATE OR REPLACE TABLE fleet-bongo-479201-u9.modulabs_project.user_rf AS (
    -- (1) 전체 거래 건수 계산
    WITH purchase_cnt AS (
        SELECT
            CustomerID,
            COUNT(DISTINCT InvoiceNo) AS purchase_cnt
        FROM fleet-bongo-479201-u9.modulabs_project.data3
        GROUP BY CustomerID
    ),
    -- (2) 구매한 아이템 총 수량 계산
    item_cnt AS (
        SELECT
            CustomerID,
            SUM(Quantity) AS item_cnt
        FROM fleet-bongo-479201-u9.modulabs_project.data3
        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 fleet-bongo-479201-u9.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(UnitPrice), 1) AS user_total
FROM fleet-bongo-479201-u9.modulabs_project.data3
GROUP BY CustomerID
```

행	CustomerID	user_total
1	12346	21.1
2	12347	481.2
3	12348	18.7
4	12349	305.1
5	12350	25.3
6	12352	330.5
7	12353	24.3
8	12354	261.2
9	12355	54.7
10	12356	170.9
11	12357	438.7
12	12358	77.2
13	12359	2210.1
14	12360	337.9
15	12361	18.4
16	12362	930.3
17	12363	53.2
18	12364	102.4
19	12365	56.6
20	12367	17.2
21	12370	423.9
22	12371	204.1
23	12372	105.1
24	12373	24.2
25	12374	99.3
26	12375	39.6
27	12377	169.3
28	12378	656.4
29	12379	86.6
30	12380	286.6

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 4362행)

- 고객별 평균 거래 금액 계산

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

```

CREATE OR REPLACE TABLE fleet-bongo-479201-u9.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 fleet-bongo-479201-u9.modulabs_project.user_rf AS rf
    LEFT JOIN (
        -- 고객 별 총 지출액
        SELECT
            CustomerID,
            ROUND(SUM(UnitPrice), 1) AS user_total
        FROM fleet-bongo-479201-u9.modulabs_project.data3
        GROUP BY CustomerID
    ) AS ut
    ON rf.CustomerID = ut.CustomerID
)

```

#두 고객의 총 지출액이 똑같이 높더라도

#어떤 고객은 평균 거래 금액이 높은, 한 번에 많이 구매하는 고객일 수 있고

#어떤 고객은 평균 거래 금액이 낮은, 여러번에 걸쳐 구매하는 고객일 수 있다.

#고객의 행동 패턴에 따라 고객 분류도 달라져야하므로 (마케팅 전략 다각도 접근)

#총 지출액만 계산하는 것에서 그치지 않고 평균 거래 금액도 계산해 새로운 테이블에 넣어준다.



이 문으로 이름이 `user_rfm`인 새 테이블이 생성되었습니다.

#두 고객의 총 지출액이 똑같이 높더라도

#어떤 고객은 평균 거래 금액이 높은, 한 번에 많이 구매하는 고객일 수 있고

#어떤 고객은 평균 거래 금액이 낮은, 여러번에 걸쳐 구매하는 고객일 수 있다.

#고객의 행동 패턴에 따라 고객 분류도 달라져야하므로 (마케팅 전략 다각도 접근)

#총 지출액만 계산하는 것에서 그치지 않고 평균 거래 금액도 계산해 새로운 테이블에 넣어준다.

## RFM 통합 테이블 출력하기

- 최종 user\_rfm 테이블을 출력하기

```
SELECT *
FROM fleet-bongo-479201-u9.modulabs_project.user_rfm
```

번호	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	12713	1	505	0	77.1	77.1
2	14569	1	79	1	47.0	47.0
3	13436	1	76	1	70.0	70.0
4	15520	1	314	1	31.0	31.0
5	13298	1	96	1	7.5	7.5
6	14204	1	72	2	99.2	99.2
7	15195	1	1404	2	2.8	2.8
8	15471	1	256	2	146.4	146.4
9	12478	1	233	3	118.4	118.4
10	12650	1	250	3	20.7	20.7
11	12442	1	181	3	12.1	12.1
12	15318	1	642	3	19.3	19.3
13	16569	1	93	3	13.0	13.0
14	14578	1	240	3	58.8	58.8
15	17914	1	457	3	152.2	152.2
16	16528	1	171	3	54.8	54.8
17	15992	1	17	3	18.3	18.3
18	18015	1	157	4	58.7	58.7
19	14219	1	78	4	16.0	16.0
20	16597	1	184	4	11.3	11.3
21	13790	1	748	4	34.7	34.7
22	12367	1	172	4	17.2	17.2
23	15097	1	170	4	56.7	56.7
24	17383	1	148	4	88.8	88.8
25	17936	1	257	5	173.9	173.9
26	16988	1	43	5	41.8	41.8
27	15773	1	311	5	13.3	13.3
28	13153	1	418	5	19.2	19.2
29	16535	1	356	5	93.8	93.8
30	15539	1	410	7	55.1	55.1

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 fleet-bongo-479201-u9.modulabs_project.us  
er_data AS (
```

```
    WITH unique_products AS (  
        SELECT  
            CustomerID,  
            COUNT(DISTINCT StockCode) AS unique_products  
        FROM fleet-bongo-479201-u9.modulabs_project.data3  
        GROUP BY CustomerID  
    )  
    SELECT ur.*, up.* EXCEPT (CustomerID)  
    FROM fleet-bongo-479201-u9.modulabs_project.user_rfm AS ur  
    JOIN unique_products AS up  
    ON ur.CustomerID = up.CustomerID
```

```
)
```

#한 가지 제품군에서만 구매하는 고객과  
#다양한 제품군을 구매하는 고객 역시 마케팅 전략이 매우 달라질 수 있다.

#고객의 구매 내역을 살폈을 때,  
#어떤 제품군을 주로 구매하는지 혹은 한 번도 구매하지 않았는지를  
#해석하는 것 또한 중요한 마케팅 재료가 될 수 있다.



이 문으로 이름이 `user_data`인 새 테이블이 생성되었습니다.

#한 가지 제품군에서만 구매하는 고객과  
#다양한 제품군을 구매하는 고객 역시 마케팅 전략이 매우 달라질 수 있다.

#고객의 구매 내역을 살펴볼 때,  
#어떤 제품군을 주로 구매하는지 혹은 한 번도 구매하지 않았는지를  
#해석하는 것 또한 중요한 마케팅 재료가 될 수 있다.

## 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 fleet-bongo-479201-u9.modulabs_project.us
er_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 Cust
                omerID ORDER BY InvoiceDate), DAY) AS interval_
            FROM
                fleet-bongo-479201-u9.modulabs_project.data3
            WHERE CustomerID IS NOT NULL
        )
        GROUP BY CustomerID
    )
```

```
)
```

```
SELECT u.*, pi.* EXCEPT (CustomerID)
FROM fleet-bongo-479201-u9.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID
```

```
)
```



이 문으로 이름이 user\_data인 테이블이 교체되었습니다.

### 3. 구매 취소 경향성

- 고객의 취소 패턴 파악하기

- 취소 빈도(cancel\_frequency) : 고객 별로 취소한 거래의 총 횟수
- 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 fleet-bongo-479201-u9.modulabs_project.us
er_data AS (
```

```
WITH TransactionInfo AS (
    SELECT
        CustomerID,
        COUNT(DISTINCT InvoiceNo) AS total_transactions,
        COUNT(DISTINCT CASE WHEN Quantity < 0 THEN InvoiceNo ELSE
NULL END) AS cancel_frequency
    FROM fleet-bongo-479201-u9.modulabs_project.data3
    GROUP BY CustomerID
)
```

```

SELECT u.* , t.* EXCEPT(CustomerID), ROUND((cancel_frequency / total_transactions) * 100, 2) AS cancel_rate
FROM fleet-bongo-479201-u9.modulabs_project.user_data AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID
)

```

**ⓘ 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 fleet-bongo-479201-u9.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	17443	1	504	219	1.1	1.1	1	0.0	1	0	0.0
2	14589	1	5	371	23.9	23.9	3	0.0	1	0	0.0
3	14124	1	1618	84	1.8	1.8	4	0.0	1	0	0.0
4	15063	1	56	213	42.8	42.8	5	0.0	1	0	0.0
5	17022	1	108	31	6.7	6.7	7	0.0	1	0	0.0
6	18191	1	140	261	28.1	28.1	7	0.0	1	0	0.0
7	15803	1	290	51	14.0	14.0	8	0.0	1	0	0.0
8	13638	1	93	15	23.9	23.9	8	0.0	1	0	0.0
9	13751	1	152	288	31.8	31.8	9	0.0	1	0	0.0
10	16457	1	140	218	24.0	24.0	9	0.0	1	0	0.0
11	14569	1	79	1	47.0	47.0	10	0.0	1	0	0.0
12	16181	1	152	73	26.5	26.5	11	0.0	1	0	0.0
13	15678	1	184	52	24.5	24.5	12	0.0	1	0	0.0
14	13897	1	200	50	33.1	33.1	17	0.0	1	0	0.0
15	16339	1	21	284	89.3	89.3	17	0.0	1	0	0.0
16	16207	1	140	25	80.9	80.9	19	0.0	1	0	0.0
17	12790	1	164	192	67.9	67.9	20	0.0	1	0	0.0
18	16072	1	140	288	193.2	193.2	23	0.0	1	0	0.0
19	15175	1	454	71	17.0	17.0	23	0.0	1	0	0.0
20	14647	1	91	201	67.6	67.6	24	0.0	1	0	0.0
21	13974	1	249	49	61.5	61.5	28	0.0	1	0	0.0
22	17201	1	231	53	75.9	75.9	38	0.0	1	0	0.0
23	12638	1	573	33	59.8	59.8	38	0.0	1	0	0.0
24	15076	1	571	172	149.3	149.3	49	0.0	1	0	0.0
25	12784	1	278	15	133.1	133.1	53	0.0	1	0	0.0
26	13339	1	326	200	282.7	282.7	54	0.0	1	0	0.0
27	15832	1	465	254	124.8	124.8	54	0.0	1	0	0.0
28	15019	1	150	266	144.7	144.7	64	0.0	1	0	0.0
29	12611	1	846	52	173.9	173.9	65	0.0	1	0	0.0
30	15721	1	388	11	174.4	174.4	80	0.0	1	0	0.0

페이지당 결과 수: 50 ▾ 1 ~ 50 (전체 4362행)

## 회고

[회고 내용을 작성해주세요]

Keep : 이 프로젝트를 하기 전에 그동안 공부했던 것들을 쭉 복습하고 왔다. 크게 도움이 됐다.

Problem : 프로젝트를 하면서 이 노션 페이지를 동시에 채워어야했는데 따로따로 해서 시간이 낭비됐다.

Try : 다음에 하는 프로젝트는 동시에 채워야겠다.