고객을 세그먼테이션하자! [프로젝트] - 김진욱

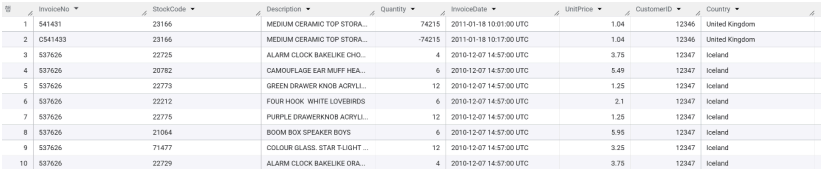
11-2. 데이터 불러오기

데이터 살펴보기

- **테이블에 있는 10개의 행만 출력하기**

```
SELECT *
FROM `massive-house-470203-k0.modulabs_project.data`
LIMIT 10;
```

[결과 이미지를 넣어주세요]

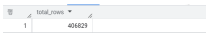


행	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	541431	23166	MEDIUM CERAMIC TOP STORA...	74215	2011-01-18 10:01:00 UTC	1.04	12346	United Kingdom
2	C541433	23166	MEDIUM CERAMIC TOP STORA...	-74215	2011-01-18 10:17:00 UTC	1.04	12346	United Kingdom
3	537626	22725	ALARM CLOCK BAKELIKE CHO...	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland
4	537626	20782	CAMOUFLAGE EAR MUFF HEA...	6	2010-12-07 14:57:00 UTC	5.49	12347	Iceland
5	537626	22773	GREEN DRAWER KNOB ACRYLI...	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland
6	537626	22212	FOUR HOOK WHITE LOVEBIRDS	6	2010-12-07 14:57:00 UTC	2.1	12347	Iceland
7	537626	22775	PURPLE DRAWERKNOB ACRYLI...	12	2010-12-07 14:57:00 UTC	1.25	12347	Iceland
8	537626	21064	BOOM BOX SPEAKER BOYS	6	2010-12-07 14:57:00 UTC	5.95	12347	Iceland
9	537626	71477	COLOUR GLASS. STAR T-LIGHT ...	12	2010-12-07 14:57:00 UTC	3.25	12347	Iceland
10	537626	22729	ALARM CLOCK BAKELIKE ORA...	4	2010-12-07 14:57:00 UTC	3.75	12347	Iceland

- **전체 데이터는 몇 행으로 구성되어 있는지 확인하기**

```
# SELECT COUNT(*) AS total_rows
FROM `massive-house-470203-k0.modulabs_project.data`;[[YOUR QUERY]]
```

[결과 이미지를 넣어주세요]



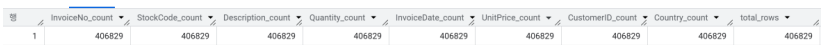
행	total_rows
1	406829

데이터 수 세기

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

```
SELECT
  COUNT(InvoiceNo)   AS InvoiceNo_count,
  COUNT(StockCode)   AS StockCode_count,
  COUNT(Description) AS Description_count,
  COUNT(Quantity)    AS Quantity_count,
  COUNT(InvoiceDate) AS InvoiceDate_count,
  COUNT(UnitPrice)   AS UnitPrice_count,
  COUNT(CustomerID)  AS CustomerID_count,
  COUNT(Country)     AS Country_count,
  COUNT(*)           AS total_rows
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]



행	InvoiceNo_count	StockCode_count	Description_count	Quantity_count	InvoiceDate_count	UnitPrice_count	CustomerID_count	Country_count	total_rows
1	406829	406829	406829	406829	406829	406829	406829	406829	406829

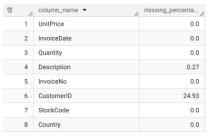
11-4. 데이터 전처리 방법(1): 결측치 제거

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 project_name.modulabs_project.data

UNION ALL
SELECT 'StockCode',
      ROUND(SUM(CASE WHEN StockCode IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'Description',
      ROUND(SUM(CASE WHEN Description IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'Quantity',
      ROUND(SUM(CASE WHEN Quantity IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'InvoiceDate',
      ROUND(SUM(CASE WHEN InvoiceDate IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'UnitPrice',
      ROUND(SUM(CASE WHEN UnitPrice IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'CustomerID',
      ROUND(SUM(CASE WHEN CustomerID IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data

UNION ALL
SELECT 'Country',
      ROUND(SUM(CASE WHEN Country IS NULL THEN 1 ELSE 0 END) / COUNT(*) * 100, 2)
FROM project_name.modulabs_project.data;
```

[결과 이미지를 넣어주세요]



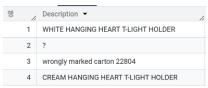
행	column_name	missing_percenta...
1	UnitPrice	0.0
2	InvoiceDate	0.0
3	Quantity	0.0
4	Description	0.27
5	InvoiceNo	0.0
6	CustomerID	24.93
7	StockCode	0.0
8	Country	0.0

결측치 처리 전략

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

```
SELECT DISTINCT Description
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE StockCode = '85123A';
```

[결과 이미지를 넣어주세요]



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

결측치 처리

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

```
DELETE FROM massive-house-470203-k0.modulabs_project.data
WHERE CustomerID IS NULL
  OR Description IS NULL;
```

[결과 이미지를 넣어주세요]

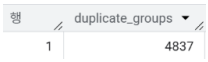
11-5. 데이터 전처리(2): 중복값 처리

중복값 확인

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

```
SELECT COUNT(*) AS duplicate_groups
FROM (
  SELECT
    InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country,
    COUNT(*) AS cnt
  FROM `massive-house-470203-k0.modulabs_project.data`
  GROUP BY 1,2,3,4,5,6,7,8
  HAVING COUNT(*) > 1
);
```

[결과 이미지를 넣어주세요]



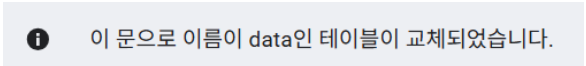
행	duplicate_groups
1	4837

중복값 처리

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

```
CREATE OR REPLACE TABLE `massive-house-470203-k0.modulabs_project.data` AS
SELECT DISTINCT *
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]



열 개수	401,604

11-6. 데이터 전처리(3): 오류값 처리

InvoiceNo 살펴보기

- **고유(unique)한 InvoiceNo의 개수를 출력하기**

```
SELECT
  COUNT(DISTINCT InvoiceNo) AS unique_invoice_no_count
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

행	unique_invoice_no_count
1	22190

- **고유한 InvoiceNo를 앞에서부터 100개를 출력하기**

```
SELECT DISTINCT InvoiceNo
FROM `massive-house-470203-k0.modulabs_project.data`
LIMIT 100;
```

[결과 이미지를 넣어주세요]

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	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT *
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE InvoiceNo LIKE 'C%'
LIMIT 100;
```

[결과 이미지를 넣어주세요]

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	22413	METAL SIGN TAKE IT OR LEAVE...	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
7	C547388	22784	LANTERN CREAM GAZEBO	-3	2011-03-22 16:07:00 UTC	4.95	12352	Norway
8	C547388	22645	CERAMIC HEART FAIRY CAKE ...	-12	2011-03-22 16:07:00 UTC	1.45	12352	Norway
9	C547388	21914	BLUE HARMONICA IN BOX	-12	2011-03-22 16:07:00 UTC	1.25	12352	Norway
10	C547388	22701	PINK DOG BOWL	-6	2011-03-22 16:07:00 UTC	2.95	12352	Norway
11	C547388	84050	PINK HEART SHAPE EGG FRYIN...	-12	2011-03-22 16:07:00 UTC	1.65	12352	Norway
12	C549955	22666	RECIPE BOX PANTRY YELLOW ...	-2	2011-04-13 13:38:00 UTC	2.95	12359	Cyprus

페이지당 결과 수: 50 1 – 50 (전체 100행)

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

```
SELECT
  ROUND(
    SUM(CASE WHEN InvoiceNo LIKE 'C%' THEN 1 ELSE 0 END) / COUNT(*) * 100,
    1
  ) AS canceled_ratio_percentage
FROM project_name.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

행	canceled_ratio_percentage
1	2.2

StockCode 살펴보기

- **고유한 StockCode의 개수를 출력하기**

```
SELECT
  COUNT(DISTINCT StockCode) AS unique_stockcode_count
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

행	unique_stockcode_count
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 `massive-house-470203-k0.modulabs_project.data`
GROUP BY StockCode
ORDER BY sell_cnt DESC
LIMIT 10;
```

[결과 이미지를 넣어주세요]

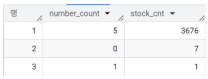
행	StockCode	sell_cnt
1	85123A	2065
2	22423	1894
3	85099B	1659
4	47566	1409
5	84879	1405
6	20725	1346
7	22720	1224
8	22197	1110
9	23203	1108
10	20727	1099

- **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

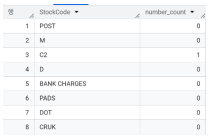
```
WITH UniqueStockCodes AS (
  SELECT DISTINCT StockCode
  FROM massive-house-470203-k0.modulabs_project.data
)
SELECT
  LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count,
  COUNT(*) AS stock_cnt
FROM UniqueStockCodes
GROUP BY number_count
ORDER BY stock_cnt DESC;

-- 숫자가 0~1개인 값들에 어떤 코드가 들어가 있는지 확인
SELECT DISTINCT StockCode, number_count
FROM (
  SELECT
    StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
```

```
  FROM `massive-house-470203-k0.modulabs_project.data`
)
WHERE number_count <= 1;
```

[결과 이미지를 넣어주세요]



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

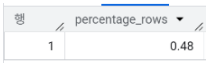


행	StockCode	number_count
1	POST	0
2	M	0
3	C2	1
4	D	0
5	BANK CHARGES	0
6	PADS	0
7	DOT	0
8	CRUK	0

- `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 DISTINCT StockCode, number_count
FROM (
  SELECT StockCode,
    LENGTH(StockCode) - LENGTH(REGEXP_REPLACE(StockCode, r'[0-9]', '')) AS number_count
  FROM project_name.modulabs_project.data
)
WHERE # [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

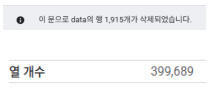


행	percentage_rows
1	0.48

- **제품과 관련되지 않은 거래 기록을 제거하기**

```
DELETE FROM massive-house-470203-k0.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 massive-house-470203-k0.modulabs_project.data
  ) AS t
  WHERE number_count <= 1
);
```

[결과 이미지를 넣어주세요]



이 문으로 data에 행 1,915개가 삭제되었습니다.

열 개수	399,689

`Description` 살펴보기

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

```
SELECT
  Description,
  COUNT(*) AS description_cnt
```

```
FROM `massive-house-470203-k0.modulabs_project.data`
GROUP BY Description
ORDER BY description_cnt DESC
LIMIT 30;
```

[결과 이미지를 넣어주세요]

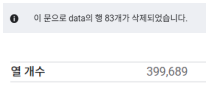
행	Description	description_cnt
1	WHITE HANGING HEART T-LIG...	2058
2	REGENCY CAKESTAND 3 TIER	1894
3	JUMBO BAG RED RETROSPOT	1659
4	PARTY BUNTING	1409
5	ASSORTED COLOUR BIRD ORN...	1405
6	LUNCH BAG RED RETROSPOT	1345
7	SET OF 3 CAKE TINS PANTRY D...	1224
8	LUNCH BAG BLACK SKULL.	1099
9	PACK OF 72 RETROSPOT CAKE ...	1062
10	SPOTTY BUNTING	1026
11	PAPER CHAIN KIT 50'S CHRIST...	1013
12	LUNCH BAG SPACEBOY DESIGN	1006

페이지당 결과 수: 50 1 – 30 (전체 30행)

- **서비스 관련 정보를 포함하는 행들을 제거하기**

```
DELETE FROM massive-house-470203-k0.modulabs_project.data
WHERE Description IN ('Next Day Carriage', 'High Resolution Image');
```

[결과 이미지를 넣어주세요]



이 문으로 data의 행 83개가 삭제되었습니다.

열 개수	399,689

- **대소문자를 혼합하고 있는 데이터를 대문자로 표준화 하기**

```
CREATE OR REPLACE TABLE `massive-house-470203-k0.modulabs_project.data` AS
SELECT
  * REPLACE(UPPER(Description) AS Description)
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

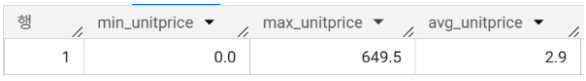
이 문으로 이름이 data인 테이블이 교체되었습니다.

UnitPrice 살펴보기

- **UnitPrice의 최솟값, 최댓값, 평균을 구하기**

```
SELECT
  MIN(UnitPrice) AS min_unitprice,
  MAX(UnitPrice) AS max_unitprice,
  ROUND(AVG(UnitPrice), 2) AS avg_unitprice
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

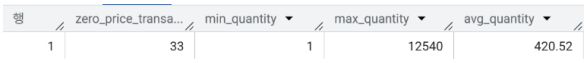


행	min_unitprice	max_unitprice	avg_unitprice
1	0.0	649.5	2.9

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

```
SELECT
  COUNT(*) AS zero_price_transactions,        -- 단가 0원 거래 개수
  MIN(Quantity) AS min_quantity,              -- 구매 수량 최솟값
  MAX(Quantity) AS max_quantity,              -- 구매 수량 최댓값
  ROUND(AVG(Quantity), 2) AS avg_quantity     -- 구매 수량 평균 (소수점 둘째 자리까지)
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE UnitPrice = 0;
```

[결과 이미지를 넣어주세요]

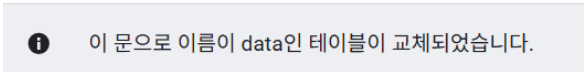


행	zero_price_transa...	min_quantity	max_quantity	avg_quantity
1	33	1	12540	420.52

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

```
DELETE FROM `massive-house-470203-k0.modulabs_project.data`
WHERE UnitPrice = 0;

CREATE OR REPLACE TABLE `massive-house-470203-k0.modulabs_project.data` AS
SELECT
  * EXCEPT(InvoiceDate),
  DATE(InvoiceDate) AS InvoiceDate
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]



11-7. RFM 스코어

Recency

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

```
SELECT
  DATE(InvoiceDate) AS InvoiceDay,
  *
FROM `massive-house-470203-k0.modulabs_project.data`;
```

[결과 이미지를 넣어주세요]

행	InvoiceDay	InvoiceNo	StockCode	Quantity	UnitPrice	CustomerID	Country	Description	InvoiceDate
1	2010-12-01	536370	10002	48	0.85	12583	France	INFLATABLE POLITICAL GLOBE	2010-12-01
2	2010-12-01	536382	10002	12	0.85	16098	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-01
3	2010-12-03	536863	10002	1	0.85	17967	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-03
4	2010-12-05	537047	10002	1	0.85	13069	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-05
5	2010-12-06	537227	10002	24	0.85	17677	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-06
6	2010-12-08	537770	10002	12	0.85	15529	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-08
7	2010-12-09	538069	10002	8	0.85	16795	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-09
8	2010-12-09	538086	10002	10	0.85	12872	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-09
9	2010-12-09	538093	10002	12	0.85	12682	France	INFLATABLE POLITICAL GLOBE	2010-12-09
10	2010-12-09	538167	10002	12	0.85	14713	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-09
11	2010-12-10	538196	10002	36	0.85	12731	France	INFLATABLE POLITICAL GLOBE	2010-12-10
12	2010-12-10	538255	10002	12	0.85	14911	EIRE	INFLATABLE POLITICAL GLOBE	2010-12-10
13	2010-12-13	538593	10002	24	0.85	16701	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-13
14	2010-12-14	538853	10002	4	0.85	16805	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-14
15	2010-12-14	538890	10002	3	0.85	12867	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-14
16	2010-12-16	539322	10002	5	0.85	14713	United Kingdom	INFLATABLE POLITICAL GLOBE	2010-12-16

페이지당 결과 수: 50 1 – 50 (전체 399573행)

- **가장 최근 구매 일자를 MAX() 함수로 찾아보기**

```
SELECT
MAX(DATE(InvoiceDate)) OVER () AS most_recent_date,
DATE(InvoiceDate) AS InvoiceDay,
*
FROM massive-house-470203-k0.modulabs_project.data;
```

[결과 이미지를 넣어주세요]

행	most_recent_date	InvoiceDay	InvoiceNo	StockCode	Quantity	UnitPrice	CustomerID	Country	Description	InvoiceDate
1	2011-12-09	2011-01-06	540372	15034	168	0.14	13081	United Kingdom	PAPER POCKET TRAVELING FAN	2011-01-06
2	2011-12-09	2011-10-19	571909	16008	6	0.12	15006	United Kingdom	SMALL FOLDING SCISSOR(POI...	2011-10-19
3	2011-12-09	2011-11-01	573777	16008	96	0.12	13726	United Kingdom	SMALL FOLDING SCISSOR(POI...	2011-11-01
4	2011-12-09	2011-12-07	581125	16011	25	0.21	14087	United Kingdom	ANIMAL STICKERS	2011-12-07
5	2011-12-09	2011-06-16	557057	16218	80	0.06	16843	United Kingdom	CARTOON PENCIL SHARPENERS	2011-06-16
6	2011-12-09	2011-06-15	556915	16235	1	0.21	15036	United Kingdom	RECYCLED PENCIL WITH RABBI...	2011-06-15
7	2011-12-09	2011-07-28	561625	16235	60	0.21	16843	United Kingdom	RECYCLED PENCIL WITH RABBI...	2011-07-28
8	2011-12-09	2011-08-17	563522	16238	28	0.21	17975	United Kingdom	PARTY TIME PENCIL ERASERS	2011-08-17
9	2011-12-09	2011-08-30	564764	16254	1	1.63	14096	United Kingdom	TRANSPARENT ACRYLIC TAPE ...	2011-08-30
10	2011-12-09	2010-12-15	539041	17012B	24	0.5	15456	United Kingdom	ORIGAMI JASMINE INCENSE/C...	2010-12-15
11	2011-12-09	2011-04-20	550785	17084J	25	0.21	15517	United Kingdom	LOVE POTION MASALA INCENSE	2011-04-20
12	2011-12-09	2011-04-19	550621	17091A	12	0.38	13263	United Kingdom	LAVENDER INCENSE IN TIN	2011-04-19
13	2011-12-09	2011-04-10	549542	17129F	48	0.64	15311	United Kingdom	BLUE GLASS GEMS IN BAG	2011-04-10

페이지당 결과 수: 50 1 – 50 (전체 399573행)

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

```
SELECT
  CustomerID,
  MAX(DATE(InvoiceDate)) AS InvoiceDay   -- InvoiceDate에서 날짜만 추출 후 최댓값(최근 일자)
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE CustomerID IS NOT NULL
GROUP BY CustomerID;
```

[결과 이미지를 넣어주세요]

행	CustomerID	InvoiceDay
1	12583	2011-12-07
2	16098	2011-09-13
3	17967	2010-12-03
4	13069	2011-12-09
5	17677	2011-12-08
6	15529	2011-11-17
7	16795	2010-12-09
8	12872	2011-01-17
9	12682	2011-12-06
10	14713	2011-11-30

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

```
SELECT
  CustomerID,
  DATE_DIFF(MAX(InvoiceDay) OVER (), InvoiceDay, DAY) AS recency
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM `massive-house-470203-k0.modulabs_project.data`
  WHERE CustomerID IS NOT NULL
  GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]

행	CustomerID	recency
1	17711	10
2	14352	157
3	13418	11
4	12933	24
5	15128	60
6	15537	163
7	17252	116
8	14081	267
9	18050	359
10	15721	11
11	15858	15
12	15230	239
13	17754	0
14	17370	72
15	14546	4
16	17346	3

페이지당 결과 수: 50

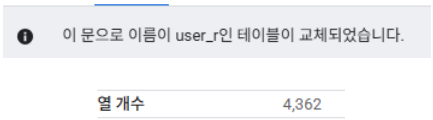
1 – 50 (전체 4362행)

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

```
CREATE OR REPLACE TABLE `massive-house-470203-k0.modulabs_project.user_r` AS
SELECT
  CustomerID,
  DATE_DIFF(MAX(InvoiceDay) OVER (), InvoiceDay, DAY) AS recency
```

```
FROM (
  SELECT
    CustomerID,
    MAX(DATE(InvoiceDate)) AS InvoiceDay
  FROM `massive-house-470203-k0.modulabs_project.data`
  WHERE CustomerID IS NOT NULL
  GROUP BY CustomerID
);
```

[결과 이미지를 넣어주세요]



Frequency

- **고객마다 고유한 InvoiceNo의 수를 세어보기**

```
SELECT
  CustomerID,
  COUNT(DISTINCT InvoiceNo) AS unique_invoice_count
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE CustomerID IS NOT NULL
GROUP BY CustomerID
ORDER BY unique_invoice_count DESC;
```

[결과 이미지를 넣어주세요]

행	CustomerID	unique_invoice_c...
1	14911	242
2	12748	217
3	17841	169
4	14606	125
5	13089	118
6	15311	118
7	12971	88
8	13408	75
9	14646	73
10	16029	66

- **각 고객 별로 구매한 아이템의 총 수량 더하기**

```
SELECT
  CustomerID,
  SUM(Quantity) AS total_quantity
FROM `massive-house-470203-k0.modulabs_project.data`
WHERE CustomerID IS NOT NULL
GROUP BY CustomerID
ORDER BY total_quantity DESC;
```

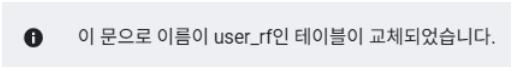
[결과 이미지를 넣어주세요]

행	CustomerID	total_quantity
1	14646	196556
2	12415	76946
3	14911	76823
4	17450	69021
5	18102	64124
6	17511	63014
7	13694	61904
8	14298	58021
9	14156	56896
10	16684	49391

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

```
CREATE OR REPLACE TABLE project_name.modulabs_project.user_rf AS

-- (1) 전체 거래 건수 계산
WITH purchase_cnt AS (
  # [[YOUR QUERY]]
),

-- (2) 구매한 아이템 총 수량 계산
item_cnt AS (
  # [[YOUR QUERY]]
)

-- 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 project_name.modulabs_project.user_r AS ur
  ON pc.CustomerID = ur.CustomerID;
```

[결과 이미지를 넣어주세요]



행	CustomerID	purchase_cnt	item_cnt	recency
1	12583	17	4978	2
2	16098	7	613	87
3	17967	1	73	371
4	13069	27	5454	0
5	17677	43	9722	1
6	15529	12	3447	22
7	16795	1	239	365
8	12872	2	319	326
9	12682	31	5485	3
10	14713	12	1611	9

페이지당 결과 수: 50 1 – 50 (전체 4362행)

Monetary

- **고객별 총 지출액 계산 (소수점 첫째 자리에서 반올림)**

```
SELECT
  CustomerID,
  # [[YOUR QUERY]] AS user_total
FROM project_name.modulabs_project.data
# [[YOUR QUERY]];
```

[결과 이미지를 넣어주세요]

행	CustomerID	total_spend
1	14646	278778.0
2	18102	259657.3
3	17450	189575.5
4	14911	128768.2
5	12415	123638.2
6	14156	113685.8
7	17511	88138.2
8	16684	65920.1
9	13694	62961.5
10	16029	60369.9

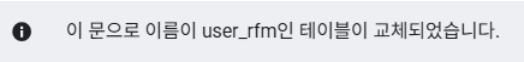
페이지당 결과 수: 50
1 – 50 (전체 4362행)

- **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 `massive-house-470203-k0.modulabs_project.user_rfm` AS
SELECT
  rf.CustomerID AS CustomerID,
  rf.purchase_cnt,
```

```
  rf.item_cnt,
  rf.recency,
  ut.user_total,
  ROUND(SAFE_DIVIDE(ut.user_total, rf.purchase_cnt), 2) AS user_average
FROM `massive-house-470203-k0.modulabs_project.user_rf` rf
LEFT JOIN (
  -- 고객별 총 지출액
  SELECT
    CustomerID,
    SUM(Quantity * UnitPrice) AS user_total
  FROM `massive-house-470203-k0.modulabs_project.data`
  WHERE CustomerID IS NOT NULL
  GROUP BY CustomerID
) ut
ON rf.CustomerID = ut.CustomerID;
```

[결과 이미지를 넣어주세요]



RFM 통합 테이블 출력하기

- **최종 user_rfm 테이블을 출력하기**

```
SELECT *
FROM massive-house-470203-k0.modulabs_project.user_rfm;
```

[결과 이미지를 넣어주세요]

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average
1	12713	1	505	0	794.5500000000...	794.55
2	15520	1	314	1	343.4999999999...	343.5
3	14569	1	79	1	227.3899999999...	227.39
4	13436	1	76	1	196.89	196.89
5	13298	1	96	1	360.0	360.0
6	14204	1	72	2	150.6099999999...	150.61
7	15195	1	1404	2	3861.0	3861.0
8	15471	1	256	2	454.4800000000...	454.48
9	16528	1	171	3	244.41	244.41
10	14578	1	240	3	168.6300000000...	168.63

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 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]

이 문으로 이름이 user_data인 테이블이 교체되었습니다.

행	CustomerID	purchase_cnt	item_cnt	recency	user_total	user_average	unique_products	average_interval
1	17331	1	16	123	175.2	175.2	1	0.0
2	17986	1	10	56	20.8	20.8	1	0.0
3	17752	1	192	359	80.64	80.64	1	0.0
4	18113	1	72	368	76.32000000...	76.32	1	0.0
5	15524	1	4	24	440.0	440.0	1	0.0
6	16061	1	-1	269	-29.95	-29.95	1	0.0
7	12603	1	56	21	613.1999999...	613.2	1	0.0
8	17443	1	504	219	534.24	534.24	1	0.0
9	13270	1	200	366	590.0	590.0	1	0.0
10	17347	1	216	86	228.96	228.96	1	0.0

2. 평균 구매 주기

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

```
CREATE OR REPLACE TABLE project_name.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_name.modulabs_project.data
    WHERE CustomerID IS NOT NULL
  )
  GROUP BY CustomerID
)

SELECT u.*, pi.* EXCEPT (CustomerID)
FROM project_name.modulabs_project.user_data AS u
LEFT JOIN purchase_intervals AS pi
ON u.CustomerID = pi.CustomerID;
```

[결과 이미지를 넣어주세요]

이 문으로 이름이 user_data인 테이블이 교체되었습니다.

행	InvoiceNo	StockC...	Quantity	UnitPrice	Custo...	Country	Description	InvoiceDate	average_interval
1	541431	23166	74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...	2011-01-18	0.0
2	C541433	23166	-74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...	2011-01-18	0.0
3	537626	22195	12	1.65	12347	Iceland	LARGE HEART MEASURING SP...	2010-12-07	2.02
4	537626	22805	12	1.25	12347	Iceland	BLUE DRAWER KNOB ACRYLIC ...	2010-12-07	2.02
5	537626	71477	12	3.25	12347	Iceland	COLOUR GLASS. STAR T-LIGHT ...	2010-12-07	2.02
6	537626	85167B	30	1.25	12347	Iceland	BLACK GRAND BAROQUE PHOT...	2010-12-07	2.02
7	537626	22726	4	3.75	12347	Iceland	ALARM CLOCK BAKELIKE GREEN	2010-12-07	2.02
8	537626	22375	4	4.25	12347	Iceland	AIRLINE BAG VINTAGE JET SET...	2010-12-07	2.02
9	537626	22492	36	0.65	12347	Iceland	MINI PAINT SET VINTAGE	2010-12-07	2.02
10	537626	22725	4	3.75	12347	Iceland	ALARM CLOCK BAKELIKE CHO...	2010-12-07	2.02

페이지당 결과 수: 50 1 – 50 (전체 399572행)

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)

```
-- 구매취소 경향성
WITH TransactionInfo AS (
  SELECT
    CustomerID,
    COUNT(DISTINCT InvoiceNo) AS total_transactions,
    COUNT(DISTINCT IF(STARTS_WITH(InvoiceNo, 'C'), InvoiceNo, NULL)) AS cancel_frequency
  FROM massive-house-470203-k0.modulabs_project.data
  GROUP BY CustomerID
)

SELECT
  u.*,
  t.* EXCEPT(CustomerID),
  ROUND(IFNULL(t.cancel_frequency, 0) / NULLIF(t.total_transactions, 0), 2) AS cancel_rate
FROM massive-house-470203-k0.modulabs_project.data AS u
LEFT JOIN TransactionInfo AS t
ON u.CustomerID = t.CustomerID;
```

[결과 이미지를 넣어주세요]

작업 정보 결과 시각화 JSON 실행 세부정보 실행 그래프

행	InvoiceNo	StockCode	Quantity	UnitPrice	Custome...	Country	Description	InvoiceDate	average_interval	total_transactions	cancel_frequency	cancel_rate
1	541431	23166	74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TO...	2011-01-18	0.0	2	1	0.5
2	C541433	23166	-74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TO...	2011-01-18	0.0	2	1	0.5
3	537626	22195	12	1.65	12347	Iceland	LARGE HEART MEASU...	2010-12-07	2.02	7	0	0.0
4	537626	22805	12	1.25	12347	Iceland	BLUE DRAWER KNOB ...	2010-12-07	2.02	7	0	0.0
5	537626	71477	12	3.25	12347	Iceland	COLOUR GLASS. STAR ...	2010-12-07	2.02	7	0	0.0
6	537626	85167B	30	1.25	12347	Iceland	BLACK GRAND BAROQ...	2010-12-07	2.02	7	0	0.0
7	537626	22726	4	3.75	12347	Iceland	ALARM CLOCK BAKELI...	2010-12-07	2.02	7	0	0.0
8	537626	22375	4	4.25	12347	Iceland	AIRLINE BAG VINTAGE...	2010-12-07	2.02	7	0	0.0
9	537626	22492	36	0.65	12347	Iceland	MINI PAINT SET VINTA...	2010-12-07	2.02	7	0	0.0
10	537626	22725	4	3.75	12347	Iceland	ALARM CLOCK BAKELI...	2010-12-07	2.02	7	0	0.0

페이지당 결과 수: 50 1 – 50 (전체 399572행)

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

```
-- 최종 user_data 출력
SELECT *
FROM massive-house-470203-k0.modulabs_project.data
```

[결과 이미지를 넣어주세요]

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

행	InvoiceNo	StockCode	Quantity	UnitPrice	CustomerID	Country	Description	InvoiceDate	average_interval
1	541431	23166	74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...	2011-01-18	0.0
2	C541433	23166	-74215	1.04	12346	United Kingdom	MEDIUM CERAMIC TOP STORA...	2011-01-18	0.0
3	537626	22195	12	1.65	12347	Iceland	LARGE HEART MEASURING SP...	2010-12-07	2.02
4	537626	22805	12	1.25	12347	Iceland	BLUE DRAWER KNOB ACRYLIC ...	2010-12-07	2.02
5	537626	71477	12	3.25	12347	Iceland	COLOUR GLASS. STAR T-LIGHT ...	2010-12-07	2.02
6	537626	85167B	30	1.25	12347	Iceland	BLACK GRAND BAROQUE PHOT...	2010-12-07	2.02
7	537626	22726	4	3.75	12347	Iceland	ALARM CLOCK BAKELIKE GREEN	2010-12-07	2.02
8	537626	22375	4	4.25	12347	Iceland	AIRLINE BAG VINTAGE JET SET...	2010-12-07	2.02
9	537626	22492	36	0.65	12347	Iceland	MINI PAINT SET VINTAGE	2010-12-07	2.02
10	537626	22725	4	3.75	12347	Iceland	ALARM CLOCK BAKELIKE CHO...	2010-12-07	2.02

페이지당 결과 수: 50 1 – 50 (전체 399572행)

회고

[회고 내용을 작성해주세요]

Keep : 데이터 이상치 제거 이후 별도저장 방식 좋지 않음

Problem : average_interval 구할때 컬럼명 중복되는 오류가 있었음, 하루에 복수 주문할 경우 기준 필요

Try : 취소 이후를 제품별로 추가 분석 필요